Assembly Language Lecture Series:

x86-64 Rotate Instructions

Sensei RL Uy, College of Computer Studies,
De La Salle University, Manila, Philippines

Copyright Notice

x86-64 Rotate Instructions

1. **ROL**
   rotate left
2. **RCL**
   rotate left thru carry
3. **ROR**
   rotate right
4. **RCR**
   rotate right thru carry

x86-64 Rotate Instructions: ROL

ROL (Rotate Left)

Syntax: ROL dst, count
dst ← dst (ROL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

x86-64 Rotate Instructions: ROL

ROL (Rotate Left)

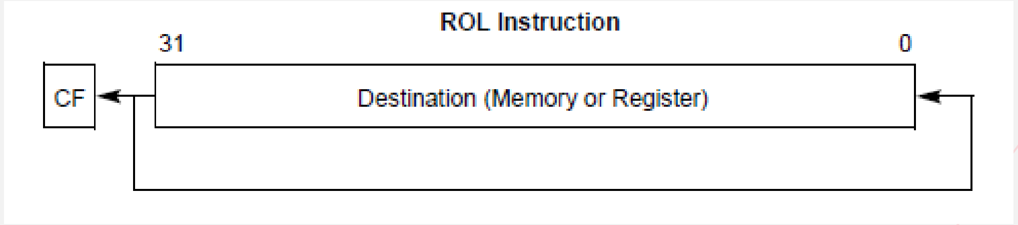
Syntax: ROL dst, count
dst ← dst (ROL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
MOV RAX, 0x1234_5678_8765_4321
ROL RAX, 32
```

1. **What will RAX contain after execution?**
2. **What will CF contain after execution?**



x86-64 Rotate Instructions: ROL

ROL (Rotate Left)

Syntax: ROL dst, count
dst ← dst (ROL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

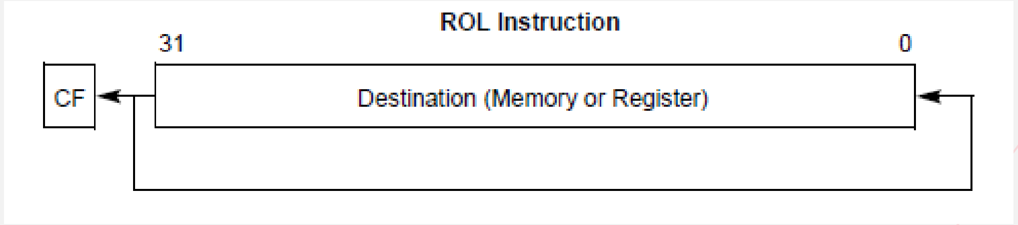
Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
MOV RAX, 0x1234_5678_8765_4321
ROL RAX, 32
```

1. What will RAX contain after execution?
2. What will CF contain after execution?

RAX = 8765_4321_1234_5678
CF = 0



x86-64 Rotate Instructions: RCL

RCL (Rotate Left thru Carry)

Syntax: RCL dst, count
dst ← dst (RCL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

x86-64 Rotate Instructions: RCL

RCL (Rotate Left thru Carry)

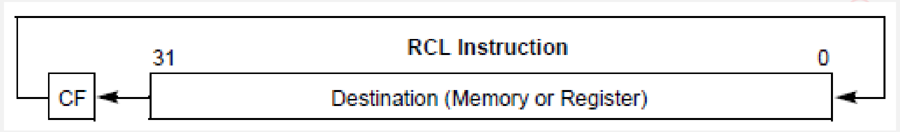
Syntax: RCL dst, count
dst ← dst (RCL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
CLC
MOV RAX, 0x1234_5678_8765_4321
RCL RAX, 4
```

1. **What will RAX contain after execution?**
2. **What will CF contain after execution?**



x86-64 Rotate Instructions: RCL

RCL (Rotate Left thru Carry)

Syntax: RCL dst, count
dst ← dst (RCL) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

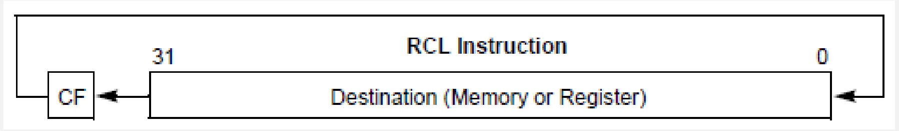
Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the CF bit (after the rotate) and the MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
CLC
MOV RAX, 0x1234_5678_8765_4321
RCL RAX, 4
```

1. **What will RAX contain after execution?**
2. **What will CF contain after execution?**

RAX=2345_6788_7654_3210
CF=1



x86-64 Rotate Instructions: ROR

ROR (Rotate Right)

Syntax: ROR dst, count
dst ← dst (ROR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF= exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF,SF,ZF,AF – no change
*all status flags no change: if count is 0

x86-64 Rotate Instructions: ROR

ROR (Rotate Right)

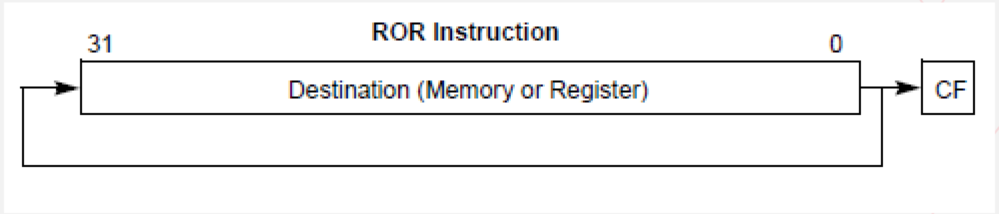
Syntax: ROR dst, count
dst ← dst (ROR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF= exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF,SF,ZF,AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
MOV RAX, 0x1234_5678_8765_4321
ROR RAX, 32
```

1. **What will RAX contain after execution?**
2. **What will CF contain after execution?**



x86-64 Rotate Instructions: ROR

ROR (Rotate Right)

Syntax: ROR dst, count
dst ← dst (ROR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

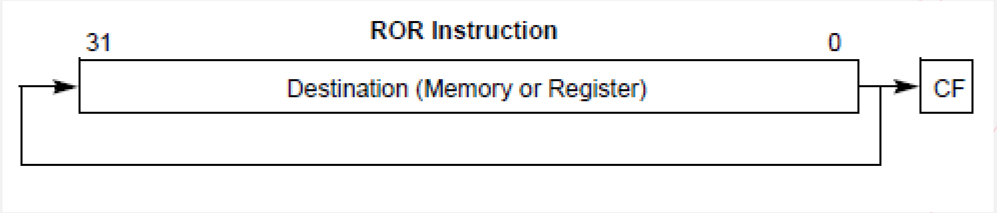
Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF= exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF,SF,ZF,AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
MOV RAX, 0x1234_5678_8765_4321
ROR RAX, 32
```

1. What will RAX contain after execution?
2. What will CF contain after execution?

RAX=8765_4321_1234_5678
CF=1



x86-64 Rotate Instructions: RCR

RCR (Rotate Right thru Carry)

Syntax: RCR dst, count
dst ← dst (RCR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

x86-64 Rotate Instructions: RCR

RCR (Rotate Right thru Carry)

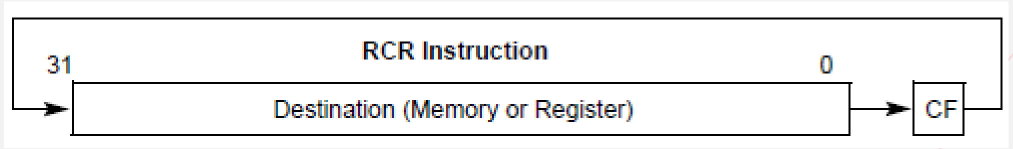
Syntax: RCR dst, count
dst ← dst (RCR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
CLC
MOV RAX, 0x1234_5678_8765_4321
RCR RAX, 4
```

1. **What will RAX contain after execution?**
2. **What will CF contain after execution?**



x86-64 Rotate Instructions: RCR

RCR (Rotate Right thru Carry)

Syntax: RCR dst, count
dst ← dst (RCR) count
*dst = r/m
*count = 1, CL or imm8
*count is masked to 5 bits (32-bit)
*count is masked to 6 bits (64-bit)

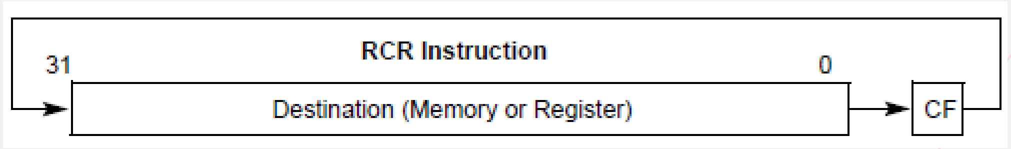
Flags affected:
*CF receives a copy of the bit that was rotated from one end to the other.
*OF = exclusive-OR of the two MSb of the result (for 1-bit shift) else undefined
*PF, SF, ZF, AF – no change
*all status flags no change: if count is 0

Example:

```
section .text
CLC
MOV RAX, 0x1234_5678_8765_4321
RCR RAX, 4
```

1. What will RAX contain after execution?
2. What will CF contain after execution?

RAX=2123_4567_8876_5432
CF=0


RCR Instruction
31
0
Destination (Memory or Register)
CF