x86 Instruction Set Reference

RCL/RCR/ROL/ROR

Rotate

Opcode	Mnemonic	Description
D0 /2	RCL r/m8, 1	Rotate 9 bits (CF, r/m8) left once.
D2 /2	RCL r/m8, CL	Rotate 9 bits (CF, r/m8) left CL times.
C0 /2 ib	RCL r/m8, imm8	Rotate 9 bits (CF, r/m8) left imm8 times.
D1 /2	RCL r/m16, 1	Rotate 17 bits (CF, r/m16) left once.
D3 /2	RCL r/m16, CL	Rotate 17 bits (CF, r/m16) left CL times.
C1 /2 ib	RCL r/m16, imm8	Rotate 17 bits (CF, r/m16) left imm8 times.
D1 /2	RCL r/m32, 1	Rotate 33 bits (CF, r/m32) left once.
D3 /2	RCL r/m32, CL	Rotate 33 bits (CF, r/m32) left CL times.
C1 /2 ib	RCL r/m32,i mm8	Rotate 33 bits (CF, r/m32) left imm8 times.
D0 /3	RCR r/m8, 1	Rotate 9 bits (CF, r/m8) right once.
D2 /3	RCR r/m8, CL	Rotate 9 bits (CF, r/m8) right CL times.
C0 /3 ib	RCR r/m8, imm8	Rotate 9 bits (CF, r/m8) right imm8 times.
D1 /3	RCR r/m16, 1	Rotate 17 bits (CF, r/m16) right once.
D3 /3	RCR r/m16, CL	Rotate 17 bits (CF, r/m16) right CL times.
C1 /3 ib	RCR r/m16, imm8	Rotate 17 bits (CF, r/m16) right imm8 times.
D1 /3	RCR r/m32, 1	Rotate 33 bits (CF, r/m32) right once.
D3 /3	RCR r/m32, CL	Rotate 33 bits (CF, r/m32) right CL times.
C1 /3 ib	RCR r/m32, imm8	Rotate 33 bits (CF, r/m32) right imm8 times.
D0 /0 ROL r/m8, 1	Rotate 8 bits r/m8 left once.	
D2 /0 ROL r/m8, CL	Rotate 8 bits r/m8 left CL times.	
C0 /0 ib ROL r/m8, imm8	Rotate 8 bits r/m8 left imm8 times.	
D1 /0 ROL r/m16, 1	Rotate 16 bits r/m16 left once.	
D3 /0 ROL r/m16, CL	Rotate 16 bits r/m16 left CL times.	
C1 /0 ib ROL r/m16, imm8	Rotate 16 bits r/m16 left imm8 times.	
D1 /0 ROL r/m32, 1	Rotate 32 bits r/m32 left once.	
D3 /0 ROL r/m32, CL	Rotate 32 bits r/m32 left CL times.	
C1 /0 ib ROL r/m32, imm8	Rotate 32 bits r/m32 left imm8 times.	
D0 /1	ROR r/m8, 1	Rotate 8 bits r/m8 right once.
D2 /1	ROR r/m8, CL	Rotate 8 bits r/m8 right CL times.
C0 /1 ib	ROR r/m8, imm8	Rotate 8 bits r/m16 right imm8 times.
D1 /1	ROR r/m16, 1	Rotate 16 bits r/m16 right once.
D3 /1	ROR r/m16, CL	Rotate 16 bits r/m16 right CL times.
C1 /1 ib	ROR r/m16, imm8	Rotate 16 bits r/m16 right imm8 times.
D1 /1	ROR r/m32, 1	Rotate 32 bits r/m32 right once.
D3 /1	ROR r/m32, CL	Rotate 32 bits r/m32 right CL times.
C1 /1 ib	ROR r/m32, imm8	Rotate 32 bits r/m32 right imm8 times.

Description

Shifts (rotates) the bits of the first operand (destination operand) the number of bit positions specified in the second operand (count operand) and stores the result in the destination operand.

The destination operand can be a register or a memory location; the count operand is an unsigned integer that can be an immediate or a value in the CL register. The processor restricts the count to a number between 0 and 31 by masking all the bits in the count operand except the 5 leastsignificant bits.

The rotate left (ROL) and rotate through carry left (RCL) instructions shift all the bits toward more-significant bit positions, except for the most-significant bit, which is rotated to the leastsignificant bit location (see Figure 7-11 in the IA-32 Intel Architecture Software Developer's Manual, Volume 1). The rotate right (ROR) and rotate through carry right (RCR) instructions shift all the bits toward less significant bit positions, except for the least-significant bit, which is rotated to the most-significant bit location (see Figure 7-11 in the IA-32 Intel Architecture Software Developer's Manual, Volume 1).

The RCL and RCR instructions include the CF flag in the rotation. The RCL instruction shifts the CF flag into the least-significant bit and shifts the most-significant bit into the CF flag (see Figure 7-11 in the IA-32 Intel Architecture Software Developer's Manual, Volume 1). The RCR instruction shifts the CF flag into the most-significant bit and shifts the least-significant bit into the CF flag (see Figure 7-11 in the IA-32 Intel Architecture Software Developer's Manual, Volume 1). For the ROL and ROR instructions, the original value of the CF flag is not a part of the result, but the CF flag receives a copy of the bit that was shifted from one end to the other.

The OF flag is defined only for the 1-bit rotates; it is undefined in all other cases (except that a zero-bit rotate does nothing, that is affects no flags). For left rotates, the OF flag is set to the exclusive OR of the CF bit (after the rotate) and the most-significant bit of the result. For right rotates, the OF flag is set to the exclusive OR of the two most-significant bits of the result.

Operation

```
switch(Instructions) {
        case RCL:
        case RCR:
                //RCL and RCR instructions
                switch(OperandSize) {
                        case 8:
                                TemporaryCount = (Count & 0x1F) % 9;
                                break;
                        case 16:
                                TemporaryCount = (Count & 0x1F) % 17;
                                break;
                        case 32:
                                TemporaryCount = Count & 0x1F;
                                break;
                }
                if(Instruction == RCL) {
```

```
                        //RCL instruction operation
                        while(TemporaryCount != 0) {
                                TemporaryCF = MSB(Destination);
                                Destination = (Destination << 1) + CF;
                                CF = TemporaryCF;
                                TemporaryCount = TemporaryCount - 1;
                        }
                        if(Count == 1) OF = MSB(Destination) ^ CF;
                        else OF = Undefined;
                }
                else {
                        //RCR instruction operation
                        if(Count == 1) OF = MSB(Destination) ^ CF;
                        else OF = Undefined;
                        while(TemporaryCount != 0) {
                                TemporaryCF = LSB(Destination);
                                Destination = (Destination >> 1) + (CF << Size);
                                CF = TemporaryCF;
                                TemporaryCount = TemporaryCount - 1;
                        }
                }
                break;
        case ROL:
        case ROR:
                //ROL and ROR instructions
                switch(OperandSize) {
                        case 8:
                                TemporaryCount = Count % 8;
                                break;
                        case 16:
                                TemporaryCount = Count % 16;
                                break;
                        case 32:
                                TemporaryCount = Count % 32;
                                break;
                }
                if(Instruction == ROL) {
                        //ROL instruction operation
                        while(TemporaryCount != 0) {
                                TemporaryCF = MSB(Destination);
                                Destination = (Destination << 1) + TemporaryCF;
                                TemporaryCount = TemporaryCount - 1;
                        }
                        CF = LSB(Destination);
                        if(Count == 1) OF = MSB(Destination) ^ CF;
                        else OF = Undefined;
                }
                else {
                        //ROR instruction operation
                        while(TemporaryCount != 0) {
                                TemporaryCF = LSB(Destination);
                                Destination = (Destination >> 1) + (TemporaryCF << Size);
                                TemporaryCount = TemporaryCount - 1;
                        }
                        CF = MSB(Destination);
                        if(Count == 1) OF = MSB(Destination) ^ SMSB(Destination); //SMSB: bit next to high-order bit
                        else OF = Undefined;
                }
                break;
}
```

Flags affected

The CF flag contains the value of the bit shifted into it. The OF flag is affected only for singlebit rotates (see "Description" above); it is undefined for multi-bit rotates. The SF, ZF, AF, and PF flags are not affected.

IA-32 Architecture Compatibility

The 8086 does not mask the rotation count. However, all other IA-32 processors (starting with the Intel 286 processor) do mask the rotation count to 5 bits, resulting in a maximum count of 31. This masking is done in all operating modes (including the virtual-8086 mode) to reduce the maximum execution time of the instructions.

Instruction	Latency	Throughput	Execution Unit
CPUID	0F3n/0F2n	0F3n/0F2n	0F2n
RCL/RCR reg	6/4	1/1	-
ROL/ROR	1/4	0.5/1	-