x86 Instruction Set Reference

LOOP/LOOPcc

Loop According to ECX Counter

Opcode	Mnemonic	Description
E2 cb	LOOP rel8	Decrement count; jump short if count != 0.
E1 cb	LOOPE rel8	Decrement count; jump short if count != 0 and ZF=1.
E1 cb	LOOPZ rel8	Decrement count; jump short if count != 0 and ZF=1.
E0 cb	LOOPNE rel8	Decrement count; jump short if count != 0 and ZF=0.
E0 cb	LOOPNZ rel8	Decrement count; jump short if count != 0 and ZF=0.

Description

Performs a loop operation using the ECX or CX register as a counter. Each time the LOOP instruction is executed, the count register is decremented, then checked for 0. If the count is 0, the loop is terminated and program execution continues with the instruction following the LOOP instruction. If the count is not zero, a near jump is performed to the destination (target) operand, which is presumably the instruction at the beginning of the loop. If the address-size attribute is 32 bits, the ECX register is used as the count register; otherwise the CX register is used.

The target instruction is specified with a relative offset (a signed offset relative to the current value of the instruction pointer in the EIP register). This offset is generally specified as a label in assembly code, but at the machine code level, it is encoded as a signed, 8-bit immediate value, which is added to the instruction pointer. Offsets of -128 to +127 are allowed with this instruction.

Some forms of the loop instruction (LOOPcc) also accept the ZF flag as a condition for terminating the loop before the count reaches zero. With these forms of the instruction, a condition code (cc) is associated with each instruction to indicate the condition being tested for. Here, the LOOPcc instruction itself does not affect the state of the ZF flag; the ZF flag is changed by other instructions in the loop.

Operation

```
if(AddressSize == 32) Count = ECX;
else Count = CX; //AddressSize == 16

Count = Count - 1;

switch(Instruction) {
        case LOOPE:
        case LOOPZ:
                if(ZF == 1 && Count != 0) BranchCond = 1;
                else BranchCond = 0;
                break;
        case LOOPNE:
        case LOOPNZ:
                if(ZF == 0 && Count != 0) BranchCond = 1;
                else BranchCond = 0;
                break;
        default: //LOOP
                if(Count != 0) BranchCond = 1;
                else BranchCond = 0;
                break;
}
if(BranchCond == 1) {
        EIP = EIP + SignExtend(Destination);
        if(OperandSize == 16) EIP = EIP & 0xFFFF;
        else /*OperandSize == 32*/ if(EIP < CS.Base || EIP < CS.Limit) Exception(GP);
}
else ResumeExecution(); //Terminate loop and continue program execution at EIP
```

Flags affected

None.

Protected Mode Exceptions

#GP(0)	If the offset being jumped to is beyond the limits of the CS segment.

Real-Address Mode Exceptions

#GP	If the offset being jumped to is beyond the limits of the CS segment or is outside of the effective address space from 0 to FFFFH. This condition can occur if a 32-bit address size override prefix is used.

Virtual-8086 Mode Exceptions

Same exceptions as in Real Address Mode

Instruction	Latency	Throughput	Execution Unit
CPUID	0F3n/0F2n/069n	0F3n/0F2n/069n	0F2n
LOOP	8	1.5	ALU