”-“表示被更改的内容，”+“表示更改后的内容

i386手册勘误

- 17.2.1 ModR/M and SIB Bytes中的Table 17-3:

```
@@ -?,2 +?,2 @@
 disp8[EDX]           010   42    4A    52    5A    62    6A    72    7A
-disp8[EPX]           011   43    4B    53    5B    63    6B    73    7B
+disp8[EBX]           011   43    4B    53    5B    63    6B    73    7B
```

- 17.2.1 ModR/M and SIB Bytes中的Table 17-4:

```
@@ -?,2 +?,2 @@
    Base =                  0     1     2     3     4     5     6     7
-   r32                     EAX   ECX   EDX   EBX   ESP   EBP   ESI   EDI
+   r32                     EAX   ECX   EDX   EBX   ESP   [*]   ESI   EDI
@@ -?,2 +?,2 @@
 [ECX*2]              001    48    49    4A    4B    4C    4D    4E    4F
-[ECX*2]              010    50    51    52    53    54    55    56    57
+[EDX*2]              010    50    51    52    53    54    55    56    57
@@ -?,2 +?,2 @@
 [EDX*4]              010    90    91    92    93    94    95    96    97
-[EBX*4]              011    98    89    9A    9B    9C    9D    9E    9F
+[EBX*4]              011    98    99    9A    9B    9C    9D    9E    9F
@@ -?,2 +?,2 @@
 NOTES:
-  [*] means a disp32 with no base if MOD is 00, [ESP] otherwise. This provides the follo
+  [*] means a disp32 with no base if MOD is 00. Otherwise, [*] means disp8[EBP] or disp3
```

- 17.2.2.11 Instruction Set Detail中的DEC -- Decrement by 1

```
@@ -?,2 +?,2 @@
 FF /1     DEC r/m16          2/6      Decrement r/m word by 1
-          DEC r/m32          2/6      Decrement r/m dword by 1
+FF /1     DEC r/m32          2/6      Decrement r/m dword by 1
```

- 17.2.2.11 Instruction Set Detail中的INC -- Increment by 1

```
@@ -?,2 +?,2 @@
 FF  /0      INC r/m16                    Increment r/m word by 1
-FF  /6      INC r/m32                    Increment r/m dword by 1
+FF  /0      INC r/m32                    Increment r/m dword by 1
```

- 17.2.2.11 Instruction Set Detail中的Jcc -- Jump if Condition is Met

```
@@ -?,2 +?,2 @@
 72  cb         JB rel8           7+m,3    Jump short if below (CF=1)
-76  cb         JBE rel8          7+m,3    Jump short if below or (CF=1 or ZF=1)
+76  cb         JBE rel8          7+m,3    Jump short if below or equal (CF=1 or ZF=1)
@@ -?,2 +?,2 @@
 7C  cb         JL rel8           7+m,3    Jump short if less (SF!=OF)
-7E  cb         JLE rel8          7+m,3    Jump short if less or equal (ZF=1 and SF!=OF)
+7E  cb         JLE rel8          7+m,3    Jump short if less or equal (ZF=1 or SF!=OF)
@@ -?,2 +?,2 @@
 0F  8C cw/cd   JL rel16/32       7+m,3    Jump near if less (SF!=OF)
-0F  8E cw/cd   JLE rel16/32      7+m,3    Jump near if less or equal (ZF=1 and SF!=OF)
+0F  8E cw/cd   JLE rel16/32      7+m,3    Jump near if less or equal (ZF=1 or SF!=OF)
```

- 17.2.2.11 Instruction Set Detail中的MOV -- Move Data

```
@@ -?,14 +?,14 @@
 8C  /r       MOV r/m16,Sreg    2/2           Move segment register to r/m word
-8D  /r       MOV Sreg,r/m16    2/5,pm=18/19  Move r/m word to segment register
+8E  /r       MOV Sreg,r/m16    2/5,pm=18/19  Move r/m word to segment register
 A0           MOV AL,moffs8     4             Move byte at (seg:offset) to AL
 A1           MOV AX,moffs16    4             Move word at (seg:offset) to AX
 A1           MOV EAX,moffs32   4             Move dword at (seg:offset) to EAX
 A2           MOV moffs8,AL     2             Move AL to (seg:offset)
 A3           MOV moffs16,AX    2             Move AX to (seg:offset)
 A3           MOV moffs32,EAX   2             Move EAX to (seg:offset)
-B0 + rb      MOV reg8,imm8     2             Move immediate byte to register
-B8 + rw      MOV reg16,imm16   2             Move immediate word to register
-B8 + rd      MOV reg32,imm32   2             Move immediate dword to register
-Ciiiiii      MOV r/m8,imm8     2/2           Move immediate byte to r/m byte
-C7           MOV r/m16,imm16   2/2           Move immediate word to r/m word
-C7           MOV r/m32,imm32   2/2           Move immediate dword to r/m dword
+B0 + rb ib   MOV reg8,imm8     2             Move immediate byte to register
+B8 + rw iw   MOV reg16,imm16   2             Move immediate word to register
+B8 + rd id   MOV reg32,imm32   2             Move immediate dword to register
+C6 ib        MOV r/m8,imm8     2/2           Move immediate byte to r/m byte
+C7 iw        MOV r/m16,imm16   2/2           Move immediate word to r/m word
+C7 id        MOV r/m32,imm32   2/2           Move immediate dword to r/m dword
```

- 17.2.2.11 Instruction Set Detail中的MUL -- Unsigned Multiplication of AL or AX

```
@@ -?,2 +?,2 @@
 Flags Affected
-OF and CF as described above; SF, ZF, AF, PF, and CF are undefined
+OF and CF as described above; SF, ZF, AF, PF are undefined
```

- 17.2.2.11 Instruction Set Detail中的OR -- Logical Inclusive OR

```
@@ -?,6 +?,6 @@
 08  /r       OR r/m8,r8        2/6       OR byte register to r/m byte
 09  /r       OR r/m16,r16      2/6       OR word register to r/m word
 09  /r       OR r/m32,r32      2/6       OR dword register to r/m dword
-0A  /r       OR r8,r/m8        2/7       OR byte register to r/m byte
-0B  /r       OR r16,r/m16      2/7       OR word register to r/m word
-0B  /r       OR r32,r/m32      2/7       OR dword register to r/m dword
+0A  /r       OR r8,r/m8        2/7       OR r/m byte to byte register
+0B  /r       OR r16,r/m16      2/7       OR r/m word to word register
+0B  /r       OR r32,r/m32      2/7       OR r/m dword to dword register
```

- 17.2.2.11 Instruction Set Detail中的PUSH -- Push Operand onto the Stack

```
@@ -?,3 +?,3 @@
 FF   /6    PUSH m32      5        Push memory dword
-50 + /r    PUSH r16      2        Push register word
-50 + /r    PUSH r32      2        Push register dword
+50 + rw    PUSH r16      2        Push register word
+50 + rd    PUSH r32      2        Push register dword
```

- 17.2.2.11 Instruction Set Detail中的REP/REPE/REPZ/REPNE/REPNZ -- Repeat Following String Operation

```
@@ -?,13 +?,13 @@
    service pending interrupts (if any);
    perform primitive string instruction;
    CountReg <- CountReg - 1;
    IF primitive operation is CMPB, CMPW, SCAB, or SCAW
    THEN
-      IF (instruction is REP/REPE/REPZ) AND (ZF=1)
+      IF (instruction is REP/REPE/REPZ) AND (ZF=0)
       THEN exit WHILE loop
       ELSE
-         IF (instruction is REPNZ or REPNE) AND (ZF=0)
+         IF (instruction is REPNZ or REPNE) AND (ZF=1)
          THEN exit WHILE loop;
          FI;
       FI;
    FI;
```

- 17.2.2.11 Instruction Set Detail中的SBB -- Integer Subtraction with Borrow

```
@@ -?,6 +?,6 @@
 18  /r       SBB r/m8,r8       2/6     Subtract with borrow byte register from r/m byte
 19  /r       SBB r/m16,r16     2/6     Subtract with borrow word register from r/m word
 19  /r       SBB r/m32,r32     2/6     Subtract with borrow dword register from r/m dwor
-1A  /r       SBB r8,r/m8       2/7     Subtract with borrow byte register from r/m byte
-1B  /r       SBB r16,r/m16     2/7     Subtract with borrow word register from r/m word
-1B  /r       SBB r32,r/m32     2/7     Subtract with borrow dword register from r/m dwor
+1A  /r       SBB r8,r/m8       2/7     Subtract with borrow r/m byte from byte register
+1B  /r       SBB r16,r/m16     2/7     Subtract with borrow r/m word from word register
+1B  /r       SBB r32,r/m32     2/7     Subtract with borrow r/m dword from dword registe
```

- 17.2.2.11 Instruction Set Detail中的SETcc - Byte Set on Condition

```
@@ -?,2 +?,2 @@
 0F  94   SETE r/m8    4/5     Set byte if equal (ZF=1)
-0F  9F   SETG r/m8    4/5     Set byte if greater (ZF=0 or SF=OF)
+0F  9F   SETG r/m8    4/5     Set byte if greater (ZF=0 and SF=OF)
@@ -?,3 +?,3 @@
 0F  9C   SETLE r/m8   4/5     Set byte if less (SF!=OF)
-0F  9E   SETLE r/m8   4/5     Set byte if less or equal (ZF=1 and SF!=OF)
-0F  96   SETNA r/m8   4/5     Set byte if not above (CF=1)
+0F  9E   SETLE r/m8   4/5     Set byte if less or equal (ZF=1 or SF!=OF)
+0F  96   SETNA r/m8   4/5     Set byte if not above (CF=1 or ZF=1)
@@ -?,2 +?,2 @@
 0F  9D   SETNL r/m8   4/5     Set byte if not less (SF=OF)
-0F  9F   SETNLE r/m8  4/5     Set byte if not less or equal (ZF=1 and SF!=OF)
+0F  9F   SETNLE r/m8  4/5     Set byte if not less or equal (ZF=0 and SF=OF)
```

- 17.2.2.11 Instruction Set Detail中的SHLD -- Double Precision Shift Left

```
@@ -?,2 +?,2 @@
 Flags Affected
-OF, SF, ZF, PF, and CF as described above; AF and OF are undefined
+SF, ZF, PF, and CF as described above; AF and OF are undefined
```

- 17.2.2.11 Instruction Set Detail中的SHLR -- Double Precision Shift Right

```
@@ -?,2 +?,2 @@
 Flags Affected
-OF, SF, ZF, PF, and CF as described above; AF and OF are undefined
+SF, ZF, PF, and CF as described above; AF and OF are undefined
```

- 17.2.2.11 Instruction Set Detail中的SUB - Integer Subtraction