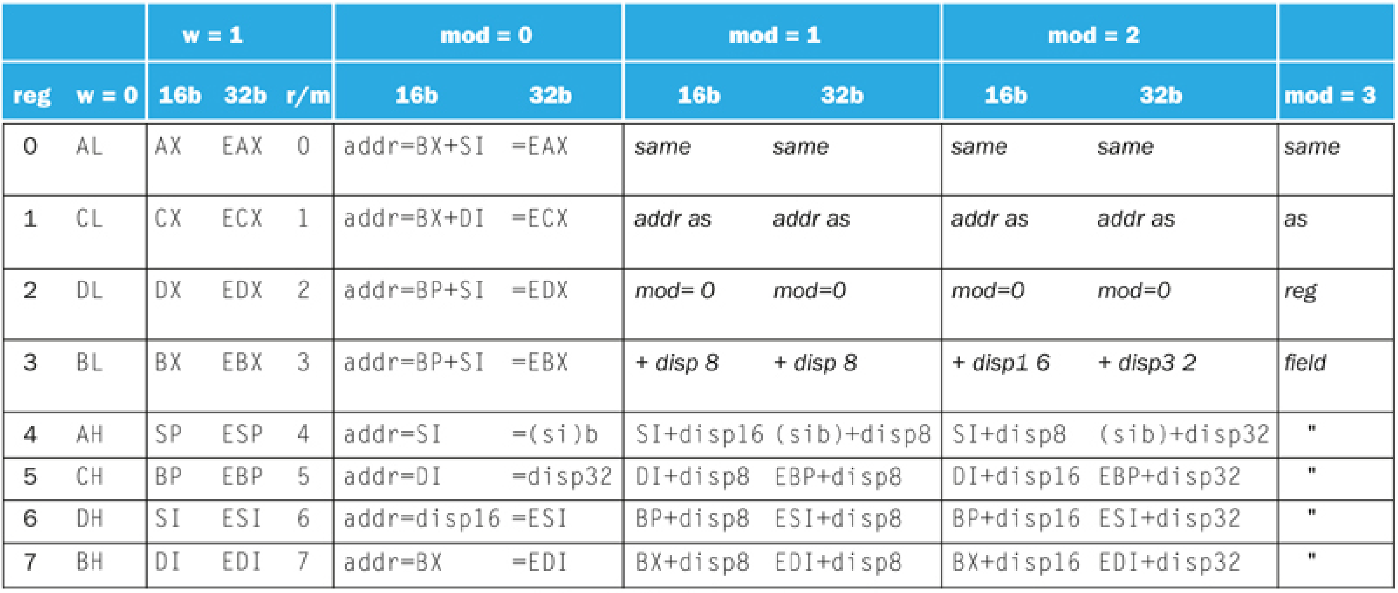

| | | w = 1 | | | mod = 0 | | mod = 1 | | mod = 2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| reg | w = 0 | 16b | 32b | r/m | 16b | 32b | 16b | 32b | 16b | 32b | mod = 3 |
| 0 | AL | AX | EAX | 0 | addr=BX+SI | =EAX | *same* | *same* | *same* | *same* | *same* |
| 1 | CL | CX | ECX | 1 | addr=BX+DI | =ECX | *addr as* | *addr as* | *addr as* | *addr as* | *as* |
| 2 | DL | DX | EDX | 2 | addr=BP+SI | =EDX | *mod= 0* | *mod=0* | *mod=0* | *mod=0* | *reg* |
| 3 | BL | BX | EBX | 3 | addr=BP+SI | =EBX | *+ disp 8* | *+ disp 8* | *+ disp1 6* | *+ disp3 2* | *field* |
| 4 | AH | SP | ESP | 4 | addr=SI | =(si)b | SI+disp16 | (sib)+disp8 | SI+disp8 | (sib)+disp32 | " |
| 5 | CH | BP | EBP | 5 | addr=DI | =disp32 | DI+disp8 | EBP+disp8 | DI+disp16 | EBP+disp32 | " |
| 6 | DH | SI | ESI | 6 | addr=disp16 | =ESI | BP+disp8 | ESI+disp8 | BP+disp16 | ESI+disp32 | " |
| 7 | BH | DI | EDI | 7 | addr=BX | =EDI | BX+disp8 | EDI+disp8 | BX+disp16 | EDI+disp32 | " |

**FIGURE E.38 The encoding of the first address specifier of the 80x86, *mod, reg, r/m*.** The first four columns show the encoding of the 3-bit reg field, which depends on the w bit from the opcode and whether the machine is in 16- or 32-bit mode. The remaining columns explain the mod and r/m fields. The meaning of the 3-bit r/m field depends on the value in the 2-bit mod field and the address size. Basically, the registers used in the address calculation are listed in the sixth and seventh columns, under mod = 0, with mod = 1 adding an 8-bit displacement and mod = 2 adding a 16- or 32-bit displacement, depending on the address mode. The exceptions are r/m = 6 when mod = 1 or mod = 2 in 16-bit mode selects BP plus the displacement; r/m = 5 when mod = 1 or mod = 2 in 32-bit mode selects EBP plus displacement; and r/m = 4 in 32-bit mode when mod ¦3 (sib) means use the scaled index mode shown in Figure E.39. When mod = 3, the r/m field indicates a register, using the same encoding as the reg field combined with the w bit.