

2.6.5. Reprezentarea instrucțiunilor mașină

O instrucțiune mașină x86 reprezintă o secvență de 1 până la 15 octeți, care prin valorile lor specifică o operație de executat, operanzii asupra cărora va fi aplicată, precum și modificatori suplimentari care controlează modul în care aceasta va fi executată.

O instrucțiune mașină x86 are maximum doi operanzi. Pentru cele mai multe dintre instrucțiuni, cei doi operanzi poartă numele de *sursă*, respectiv *destinație*. Dintre cei doi operanzi, maximum unul se poate afla în memoria RAM. Celălalt se află fie într-un registru al **EU**, fie este o constantă întreagă. Astfel, o instrucțiune are forma:

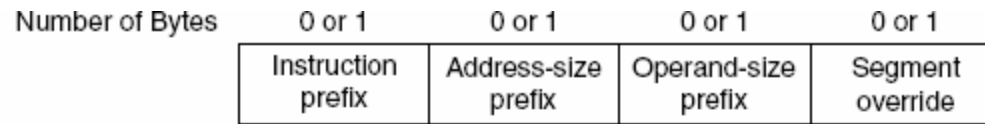
numeinstrucțiune destinație, sursă

Formatul intern al unei instrucțiuni este variabil, el putând ocupa între 1 și 15 octeți, având următoarea formă generală de reprezentare (*Instructions byte-codes from OllyDbg*):

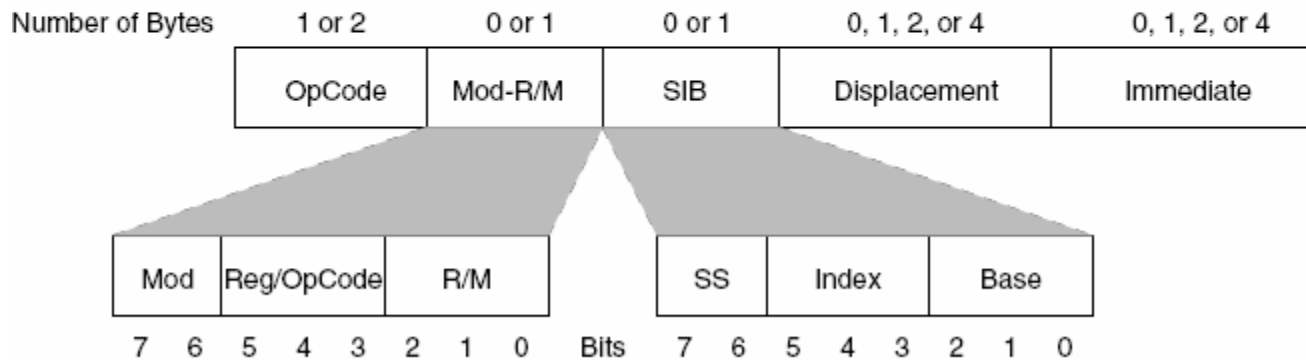
[prefixe] + cod + [ModR/M] + [SIB] + [deplasament] + [imediat]

Prefixele controlează modul în care o instrucțiune se execută. Acestea sunt opționale (0 până la maximum 4) și ocupă câte un octet fiecare. De exemplu, acestea pot solicita execuția repetată (în buclă) a instrucțiunii curente sau pot bloca magistrala de adrese pe parcursul execuției pentru a nu permite accesul concurent la operanzi și rezultate.

Operația care se va efectua este identificată prin intermediul a 1 sau 2 octeți de *cod* (opcode), aceștia fiind singurii octeți obligatoriu prezenți, indiferent de instrucțiune.

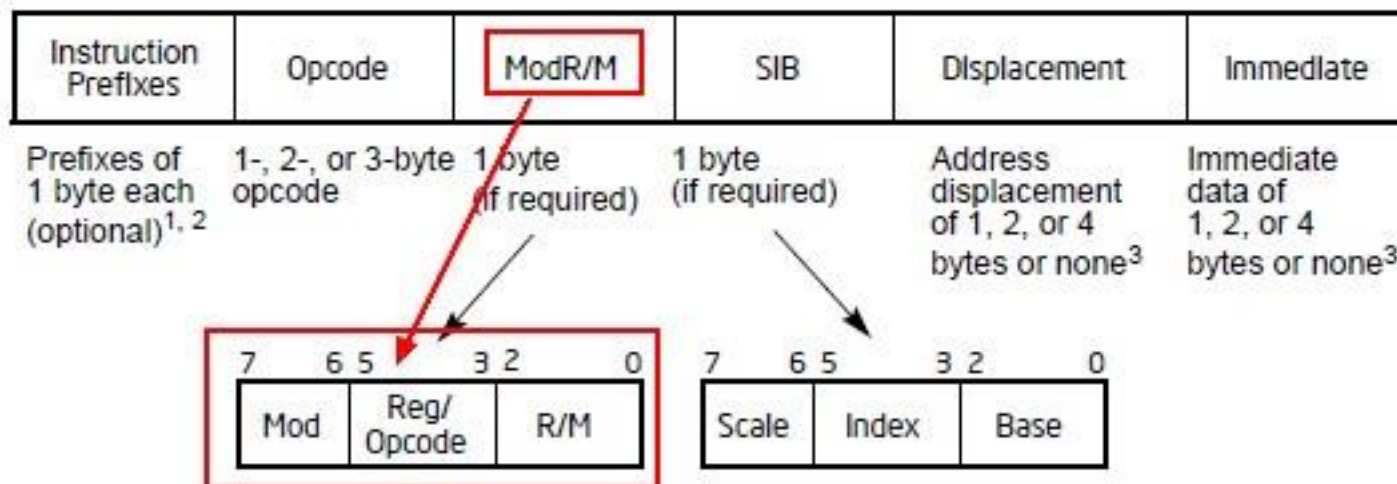


(a) Optional instruction prefixes



(b) General instruction format

Although the diagram seems to imply that instructions can be up to 16 bytes long, in actuality the x86 will not allow instructions greater than 15 bytes in length.



Octetul *ModR/M* (mod registru/memorie) specifică pentru unele dintre instrucțiuni natura și locul operanzilor (registru, memorie). Acesta permite specificarea fie a unui registru, fie a unei locații de memorie a cărei adresă este exprimată prin intermediul unui offset (<http://datacadamia.com/intel/modrm>)

Pentru cazuri mai complexe de adresare decât cele codificabile direct prin ModR/M, combinarea acestuia cu octetul SIB (Scale – Index – Base) permite următoarea formulă generală de definire a unui offset:

$$\text{offset} = [\text{bază}] + [\text{index} \times \text{scală}] + [\text{constantă}]$$

(SIB) (deplasament + imediat)

unde pentru bază și index vor fi folosite valorile a doi regiștri iar scală este 1, 2, 4 sau 8. Regiștrii permiși ca bază sau / și index sunt: EAX, EBX, ECX, EDX, EBP, ESI, EDI. Registrul ESP este disponibil ca bază însă nu poate fi folosit cu rol de index. (http://www.c-jump.com/CIS77/CPU/x86/lecture.html#X77_0100_sib_byte_layout)

Majoritatea instrucțiunilor folosesc pentru reprezentare fie numai campul de cod, fie cod urmat de ModR/M.

Deplasament (displacement) apare în cazul unor forme de adresare particulare (operanzi din memorie) și urmează direct după ModR/M sau SIB, când SIB este prezent. Acest câmp poate fi codificat fie pe octet, fie pe cuvânt, fie pe dublu cuvânt (32 biți).

The most common addressing mode, and the one that's easiest to understand, is the *displacement-only* (or **direct**) addressing mode. The displacement-only addressing mode consists of a 32-bit constant that specifies the address of the target location. The displacement-only addressing mode **is perfect for accessing simple scalar variables**. Intel named this the displacement-only addressing mode because a 32-bit constant (displacement) follows the MOV opcode in memory. On the 80x86 processors, this displacement is an offset from the beginning of memory (that is, address zero).

Displacement mode, the **operand's offset** is contained as part of the instruction as an 8-, 16-, or 32-bit displacement. The displacement addressing mode is found on few machines because, as mentioned earlier, it leads to long instructions. In the case of the x86, the displacement value can be as long as 32 bits, making for a 6-byte instruction. **Displacement addressing can be useful for referencing global variables.**

Ca și consecință a imposibilității prezenței mai multor câmpuri de ModR/M, SIB și deplasament într-o instrucțiune, arhitectura 80x86 NU permite codificarea a două adrese de memorie în aceeași instrucțiune.

Valoare imediată oferă posibilitatea definirii unui operand ca fiind o constantă numerică pe 1, 2 sau 4 octeți. Când este prezent, acest câmp apare întotdeauna la sfârșitul instrucțiunii.