24.24.1 Охарактеризувани алгоритм визначення байтів modr/mта Sib на другому перегляді компілятора Асемблера.

Структура байта Modr/mпри 16-розрядній адресації.

Поле mod - 2 біта	Поле reg - 3 біта	Поле r/m - 3 біта

Поле mod використовується для визначення зміщення в команді, а поле r/m - для визначення регістрів адрес, вміст яких використовується для формування ефективної адреси. Поле reg призначене для задання регісра даних, який використовується в команді або є частиною коду операції.

Розглянемо поле mod:

- При mod=00 зміщення в команді відсутнє, а ефективна адреса формується по вмісту регістру (регістрів) адрес, які задаються полем r/m.
- При mod=01 зміщення в команді однобайтне, яке при формуванні ефективної адреси знаково розширюється до двох байт з послідуючим додаванням до вмісту регістру (регістрів) адрес, які задаються полем r/m.
- При mod=10 зміщення в команді двохбайтне і додається при формуванні ефективної адреси до вмісту регістру (регістрів) адрес, які задаються полем r/m.
- При mod=11 адреса пам'яті не задається, а поле r/m задає код регістра даних.

В полі r/m регістри адрес або їх можлива комбінація задається слідуючим чином:

Таблиця 5.1

Код в полі r/m	Регістри адрес	Сегментний регістр, який використовується по замовчуванню
000	BX+SI	DS
001	BX+DI	DS
010	BP+SI	SS
011	BP+DI	SS
100	SI	DS
101	DI	DS
110	BP	SS
111	BX	DS

З поданої таблиці випливають наступні висновки, які справедливі і для сучасних мікропроцесорів сімейства при 16-розрядній адресації:

1. При 16-розрядній адресації тільки чотири регістра -BX,BP,SI та DI можуть використовуватись як регістри адрес.
2. Для формування багатокомпонентної адреси використовуються тільки обмежений набір пар регістрів - (BX,SI), (BX,DI), (BP,SI) (BP,DI).
3. Із реалізації випадає один із широко вживаних режимів - режим прямої адресаціїї, тобто режим коли зміщення в команді і є зміщенням в сегменті.

Відносно п.3 інженери фірми Intel вимушені були прийняте наступне вирішення - режим прямої адресації ввести при mod=00 та r/m=110, тобто, не дивлячись на те, що mod=00 зміщення в команді задавати двохбайтним, якщо r/m=110. При цьому регістр BP не використовується. Це дуже нагадує "латку" в програмах. Але ця "латка" досить продумана. З апаратної реалізації випадає режим посередньої регістрової адресації з використанням регістра BP як регістра адреси, але використання цього режиму, в стратегічному призначенні регістра BP як базового регістра структур даних стека, малоймовірне. В крайньому випадку можна використати режим при mod=01 та нульовим байтом зміщення в команді, що і реалізовано трансляторами програм на мові Асемблера.

Інтерпретація полів байта mod-r/m в режимі 32-розрядної адресації відрізняється від режиму 16-розрядної адресації. Головна відмінність – в інтерпретації процесором поля r/m (див. Табл 5.2)

Таблиця 5.2

Код в полі r/m	Регістри адрес	Сегментний регістр, який використовується по замовчуванню
000	EAX	DS
001	ECX	DS
010	EDX	DS
011	EBX	DS
100	Наявність байта SIB	--
101	EBP*	SS
110	ESI	DS
111	EDI	DS

Байт SIB має наступну структуру

Поле множника – 2 біта	Поле індексного регістра - 3 біта	Поле базового регістра - 3 біта

В полі індексного регістра може вказуватись будь який регістр за виключенням ESP. При формуванні ефективної адреси вміст регістру зсувається вліво на кількість розрядів, які вказані в полі множника. Тим самим фактично при формуванні ефективної адреси відбувається множення вмісту індексного регістру на 2, 4 або 8, що позбавляє програміста додаткових дій при адресації елементів масивів слів, подвійних слів та квадрослів.

В полі базового регістра може використовуватись любий РЗП, включаючи і ESP.

Таким чином можливість формування ефективної адреси в 32-розрядному режимі значно ширша, порівнюючи з 16-розрядним режимом адрес. Тому 32-розрядний режим за допомогою префіксу зміни розрядності адрес може використовуватись і в реальному режимі при наступному уточненні – **старші 16 розрядів ефективної адреси в реальному режимі ігноруються.**

Окрім іншої інтерпретації поля r/m по іншому інтерпретується поле mod

- При mod=01 зміщення в команді однобайтне, яке при формуванні ефективної адреси знаково розширюється до чотирьох байт з послідуючим додаванням до вмісту регістру (регістрів) адрес, які задаються полем r/m та sib.
- При mod=10 зміщення в команді чотирьохбайтне і додається при формуванні ефективної адреси до вмісту регістру (регістрів) адрес, які задаються полем r/m та sib.

Приклади адресації в 32-розрядному режимі.

```
Mov     ax,Value[ecx*4+edx]
Mov     edx,[edx*8]
Mov     al,[eax+esp]
```

Як і у випадку 16-розрядних адрес, режим використання регістру EBP при mod=0 відсутній. При коді 101 в полі r/m або в полі базового регістру байта SIB встановлюється режим використання 4-х байтного зміщення в команд.