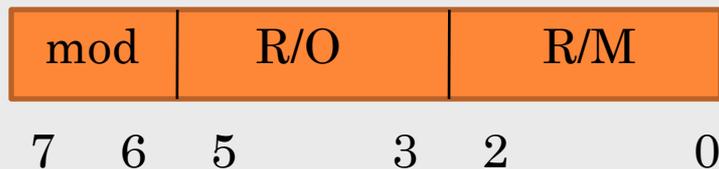


ФОРМАТ КОМАНДЫ ПРОЦЕССОРА INTEL С АРХИТЕКТУРОЙ IA-32

ОБЩИЕ СВЕДЕНИЯ О ФОРМАТЕ КОМАНДЫ

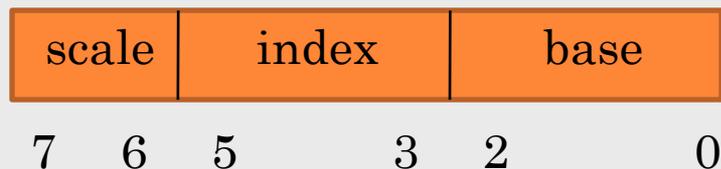
- Префиксы (0-5 байт)
 - Блокировки
 - Замены сегмента
 - Замены длины операнда и длины адреса
 - Повторений
- Код команды (1-2 байта)
 - Может включать в себя специальные поля регистра, размера операнда, размера непосредственного значения, направления выполнения команды, кода условия
- Mod R/M (0-1) байт
 - Определяет способ адресации нерегистрового операнда, регистр или дополнительный код операции



ОБЩИЕ СВЕДЕНИЯ О ФОРМАТЕ КОМАНДЫ

□ SIB – Scale-Index-Base

- Дополнительное поле для адресации в 32-х разрядном режиме. Хранит коэффициент масштаба, номера индексного и базового регистров



- Непосредственное значение (0-4 байта)
- Смещение (0-4 байта). В командах дальних переходов хранится 2 поля – смещение и селектор сегмента



ПРЕФИКСЫ

Значение префикса	Название префикса	Влияние префикса на выполнение команды
F0h	LOCK	Префикс блокировки шины. Используется в многопроцессорных конфигурациях, которые работают по принципу «чтение-модификация-запись». При выполнении команды с префиксом LOCK процессор получает монополярный доступ к шине, не позволяя другим процессорам обратиться к ней.
F2h	REPNE/ REPNZ	Данный префикс формируется в машинной команде, если в программе перед инструкцией была использована префиксная команда REPNE/REPNZ. Префикс означает, что команда будет повторяться до тех пор, пока ее операнды не равны.
F3h	REP/REP E/REPZ	Данный префикс формируется в машинной команде, если в программе перед инструкцией была использована префиксная команда REP/REPЕ/REPZ. Префикс означает, что команда будет повторяться до тех пор, пока ее операнды равны или пока регистр CL/CX/ECX не примет значение 0.

ПРЕФИКСЫ

Значение префикса	Название префикса	Влияние префикса на выполнение команды
2Eh	CS:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром CS: <code>mov ax, word ptr CS:some_variable</code>
36h	SS:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром SS: <code>mov ax, word ptr SS:some_variable</code>
3Eh	DS:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром DS: <code>mov ax, word ptr DS:some_variable</code>

ПРИМЕР ФОРМИРОВАНИЯ ПРЕФИКСА ПОВТОРЕНИЯ

```
1      0000                                     .model tiny
2      0000                                     .code
3                                             org 100h
4      0100  8B 1E 0111r      main: mov bx, x
5      0104  8B 0E 0113r      mov cx, n
6      0108  B8 0001          mov ax, 1
7      010B  33 D2          xor dx,dx
8      010D  F3> F7 E3      rep mul bx
9      0110  C3              ret
10     0111  0002          x      dw          2
11     0113  0007          n      dw          7
12
```



ПРЕФИКСЫ

Значение префикса	Название префикса	Влияние префикса на выполнение команды
26h	ES:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром ES: <code>mov ax, word ptr ES:some_variable</code>
64h	FS:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром FS: <code>mov ax, word ptr FS:some_variable</code>
65h	GS:	Префикс формируется, если перед операндом инструкции, который находится в памяти, явно указана замена сегмента на сегмент, адресованный регистром GS: <code>mov ax, word ptr GS:some_variable</code>

ПРИМЕР ФОРМИРОВАНИЯ ПРЕФИКСА ЗАМЕНЫ СЕКМЕНТА

```
1      0000                                .model tiny
2      0000                                .code
3                                          org 100h
4      0100  33 DB                          main: xor bx, bx
5      0102  8B 87 0115r                     mov ax, arr[bx]
6      0106  8B 87 0117r                     mov ax, DS:arr+2[bx]
7      010A  26: 8B 87 0117r                 mov ax, ES:arr+2[bx]
8      010F  2E: 8B 87 0117r                 mov ax, CS:arr+2[bx]
9      0114  C3                               ret
10     0115  0A*(0000)                       arr dw 10 dup(0)
11                                          end main
```



ПРЕФИКСЫ

Значение префикса	Название префикса	Влияние префикса на выполнение команды
66h	OS	<p><i>Operand Size</i> – префикс замены размера операнда. Если операнд находится в 16-разрядном сегменте и указан данный префикс, то размер операнда будет считаться равным 32-м разрядам.</p> <p>Если операнд находится в 32-разрядном сегменте и указан данный префикс, то размер операнда будет считаться равным 16-ти разрядам.</p> <p>Если данный префикс не указан, то размер операнда совпадает с разрядностью сегмента.</p>
67h	AS	<p><i>Address Size</i> – префикс замены размера эффективного адреса.</p> <p>Если операнд находится в 16-разрядном сегменте и указан данный префикс, то для обращения к этому операнду будет рассчитан 32-х разрядный эффективный адрес.</p> <p>Если операнд находится в 32-разрядном сегменте и указан данный префикс, то для обращения к этому операнду будет рассчитан 16-ти разрядный эффективный адрес.</p> <p>Если данный префикс не указан, то размер эффективного адреса совпадает с разрядностью сегмента.</p>

ПРИМЕР ФОРМИРОВАНИЯ ПРЕФИКСА OS

```
1          0000                                .model      tiny
2                                                .386
3          0000                                .code
.....
4                                                org 100h
5          0100  66| A1 0105r                main: mov  eax, x
6          0104  C3                          ret
7          0105  00000064                    x      dd   100
8                                                end main
```



ПРИМЕР ФОРМИРОВАНИЯ ПРЕФИКСА AS

```
1          0000          .model      small
2
3          .386
4          ; сегмент кода
5          CODES SEGMENT PARA USE16 'CODE'
6          ASSUME CS:CODES,
7          DS:DATAS,
8          ES:DATAS
9
10         0000  66| 33 DB      main:  xor ebx,ebx
11         0003  67| 8B 83 00000000r mov ax,word ptr arr[ebx]
12         000A  C3
13         000B          CODES ENDS
14
15         ; сегмент стека
16         .stack      100h
17
18         ; сегмент данных
19         DATAS SEGMENT PARA USE32 'DATA'
20         arr dw      100 dup(0)
21         DATAS ENDS
22         end main
```



Поля кода операции

Название специального поля	Значение
поле длины регистра (w)	$w = 0$ – команда работает с байтами; $w = 1$ – команда работает со словами или двойными словами в зависимости от разрядности сегмента и наличия префикса OS
поле размера непосредственного операнда (s)	$s = 0$ – непосредственный операнд указан полностью; $s = 1$ – в команде указан только младший байт 2-хбайтного или 4-хбайтного операнда, причем этот байт должен рассматриваться как число со знаком. Перед обработкой данный операнд должен быть расширен до нормальной длины
поле направления операции (d)	$d = 0$ – код источника указан в поле R/O, а приемника – в R/M; $d = 1$ – код источника указан в поле R/M, а приемника – в R/O; Наличие этого поля позволяет всегда хранить операнд, адресованный регистром, в поле R/O байта Mod R/M, а второй операнд, который может быть адресован любым способом – в поле R/M. При этом неважно, на каком месте в команде будут находиться данные операнды



Поля кода операции

Название специального поля	Значение
поле регистра (<i>reg</i>)	3-хбитное поле, которое позволяет указать код регистра, с которым работает данная команда. Если код регистра не указан в поле <i>reg</i> , то возможно, что он указан в поле R/O байта Mod R/M. Коды регистров приведены ниже
поле условия (<i>cond</i>)	4-хбитное поле, которое используется в командах, которые выполняются в зависимости от определенных условий, например, в командах условного перехода. Коды условий приведены ниже.



КОДЫ РЕГИСТРОВ

Код	Регистры	Код	Регистры
000	AL/AX/EAX/ST(0)/MM0	100	AH/SP/ESP/ST(4)/MM 4
001	CL/CX/ECX/ST(1)/MM1	101	CH/BP/EBP/ST(5)/MM 5
010	DL/DX/EDX/ST(2)/MM2	110	DH/SI/ESI/ST(6)/MM6
011	BL/BX/EBX/ST(3)/MM3	111	BH/DI/EDI/ST(7)/MM7



КОДЫ УСЛОВИЙ

Код	Услови е	Значения флагов	Код	Условие	Значения флагов
0000	O	OF = 1	1000	S	SF = 1
0001	NO	OF = 0	1001	NS	SF = 0
0010	C/B/N AE	CF = 1	1010	P/PE	PF = 1
0011	NC/NB /AE	CF = 0	1011	NP/PO	PF = 0
0100	E/Z	ZF = 1	1100	L/NGE	SF <> OF
0101	NE/NZ	ZF = 0	1101	NL/GE	SF = OF
0110	BE/NA	CF = 1 и ZF = 1	1110	LE/NG	ZF = 1 и SF <> OF
0111	NBE/A	CF = 0 и ZF = 0	1111	NLE/G	ZF = 0 и SF = OF



ФОРМИРОВАНИЕ ПОЛЯ REG

Поле регистра *reg*. В качестве примера рассмотрим команду уменьшения регистра на 1 – DEC. Для уменьшения значения 16-ти и 32-х битных регистров используется команда следующего формата (условно обозначим *48reg*):

7	6	5	4	3	2	1	0
0	1	0	0	1	reg		

Тогда команда уменьшения на 1 значения регистра VX/EBX будет иметь код *4Bh*

7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	1

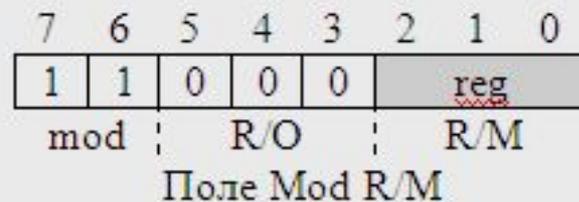
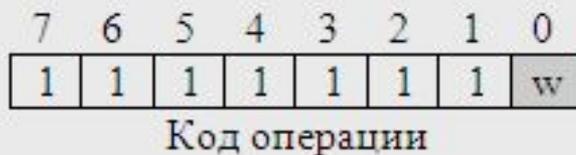
а команда уменьшения на 1 значения регистра SP/ESP будет иметь код *4Ch*:

7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	0

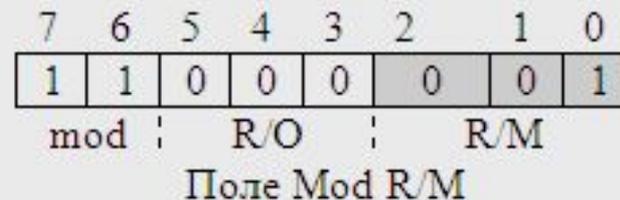
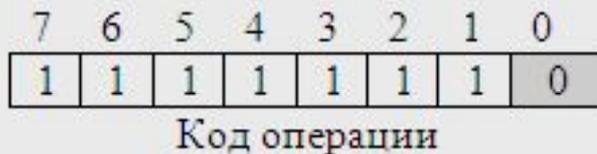
Какой конкретно регистр будет использован – 16-ти или 32-х разрядный, будет определено в зависимости от разрядности сегмента и наличия префикса OS.

ФОРМИРОВАНИЕ ПОЛЯ W

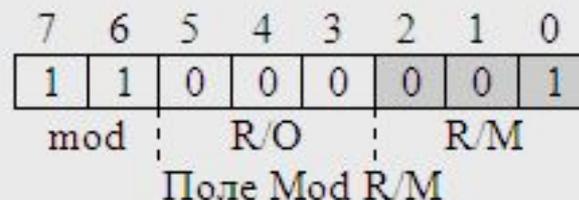
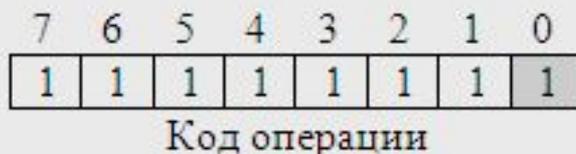
Формирование поля длины регистра w . Рассмотрим команду увеличения на 1 значения произвольного регистра INC. Данная команда имеет следующий формат:



Тогда команда увеличения регистра CL будет иметь код FE C1 и вид (поле $w = 0$, поле R/M = 001):



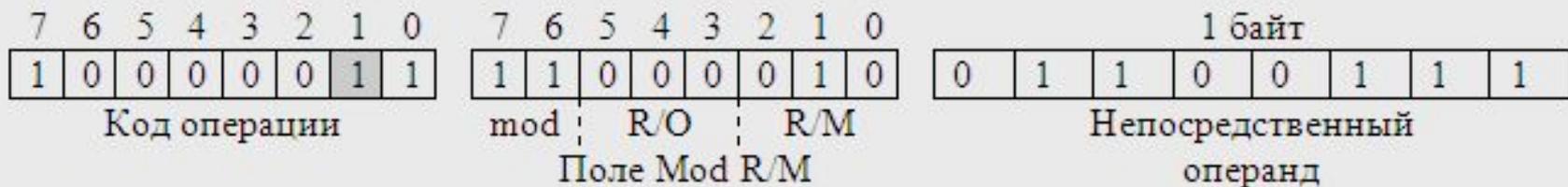
а команда увеличения регистра CX – код FF C1 и вид (поле $w = 1$, поле R/M = 001):



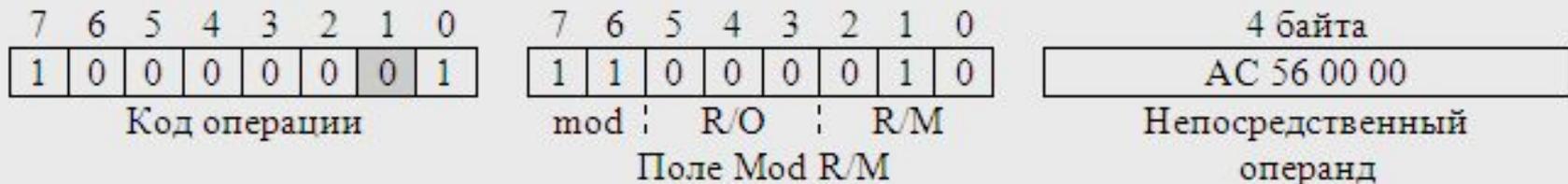
ФОРМИРОВАНИЕ ПОЛЯ S

6	83 C2 67	add <u>edx</u> , 67h
7	81 C2 000056ACr	add <u>edx</u> , 56ACh

Обратите внимание, что первая команда сложения `add edx, 67h` занимает всего 3 байта – непосредственный операнд удалось поместить в 1 байт и, следовательно, бит $s = 1$. В бинарном выражении это будет выглядеть следующим образом:



Во второй команде сложения поместить двухбайтный операнд в один байт невозможно, поэтому бит $s = 0$:



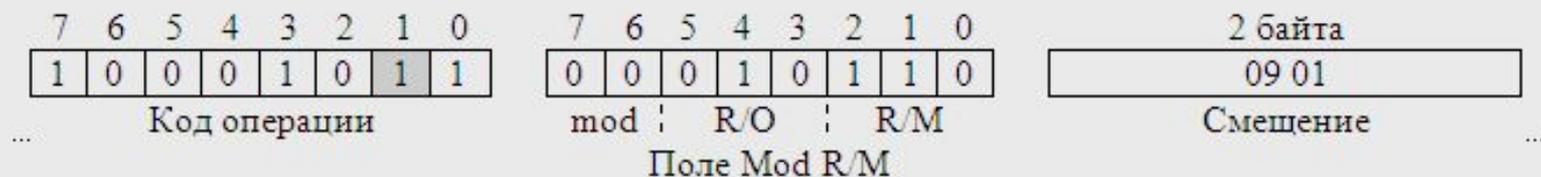
ФОРМИРОВАНИЕ ПОЛЯ D

```
1      0000                                .model tiny
2      0000                                .code
3                                          org 100h
4      0100                                main:
5      0100  8B 16 0109r                    mov dx, x
6      0104  89 16 0109r                    mov x, dx
7      0108  C3                               ret
8      0109  0017                            x      dw      17h
9                                          end main
```

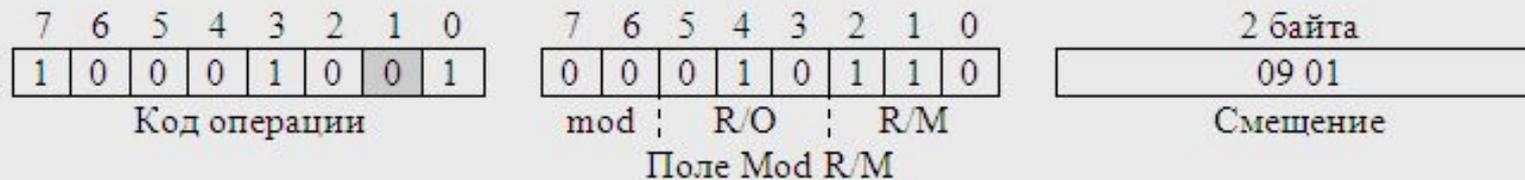


ФОРМИРОВАНИЕ ПОЛЯ D

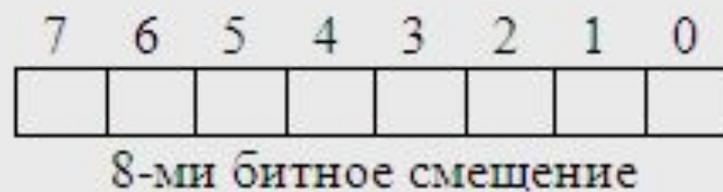
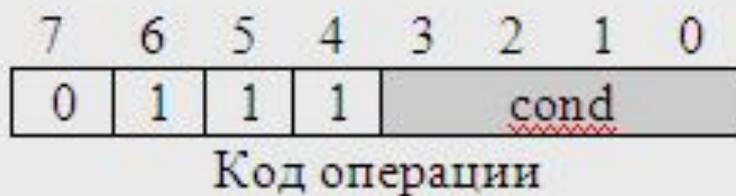
MOV dx, x



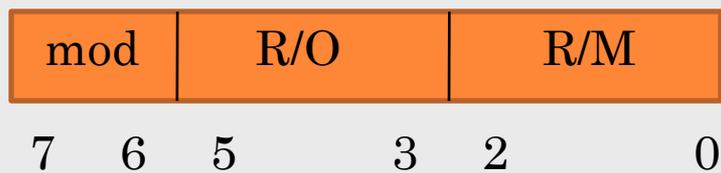
MOV x, dx



Поле кода условий в командах условного перехода



ПОЛЕ MOD R/M



ЗНАЧЕНИЕ ПОЛЯ MOD

Значение поля mod	Комментарий
00	Используется адресация без смещения: - при использовании 32-х разрядных регистров – косвенная; - при использовании 16-ти разрядных регистров – косвенная или по базе с индексированием и без смещения
01	Используется адресация с 8-ми битным смещением
10	Используется адресация с 16-ти или 32-х битным смещением
11	Используется регистровая адресация и поле R/M содержит номер регистра



ЗНАЧЕНИЕ ПОЛЯ R/O

- Продолжение кода операции
- Номер регистра

Код	Регистры	Код	Регистры
000	AL/AX/EAX/ST(0)/MM0	100	AH/SP/ESP/ST(4)/MM4
001	CL/CX/ECX/ST(1)/MM1	101	CH/BP/EBP/ST(5)/MM5
010	DL/DX/EDX/ST(2)/MM2	110	DH/SI/ESI/ST(6)/MM6
011	BL/BX/EBX/ST(3)/MM3	111	BH/DI/EDI/ST(7)/MM7

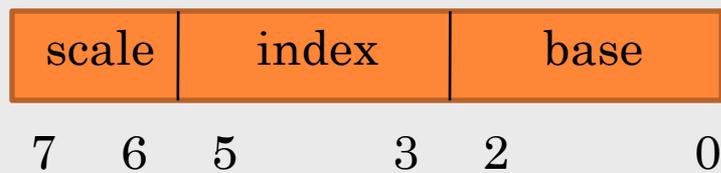


ЗНАЧЕНИЕ ПОЛЯ R/M

Значение	Значение для 16-ти разрядных регистров	Значение для 32-х разрядных регистров
000	[BX + SI]	[EAX]
001	[BX + DI]	[ECX]
010	[BP + SI]	[EDX]
011	[BP + DI]	[EBX]
100	[SI]	Адресация задана через SIB
101	[DI]	[EBP], но если mod = 00, то после поля mod R/M находится 32-х битное смещение, т.е. используется прямая адресация
110	[BP], но если mod = 00, то после поля mod R/M находится 16-ти битное смещение, т.е. используется прямая адресация	[ESI]
111	[BX]	[EDI]



ПОЛЕ SIB



ЗНАЧЕНИЕ ПОЛЯ S (SCALE)

Значение S	Коэффициент масштаба
00	1 или нет
01	2
10	4
11	8



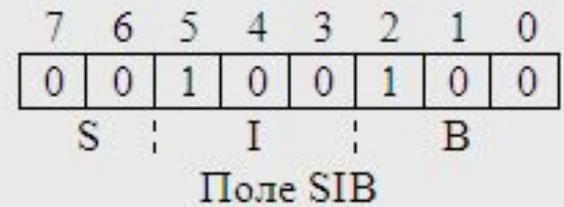
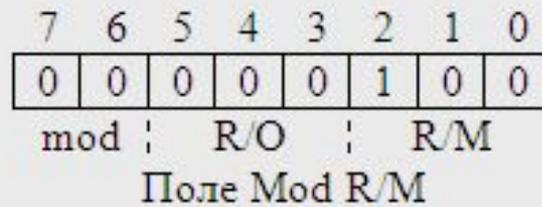
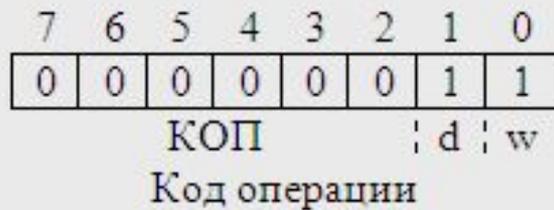
ЗНАЧЕНИЕ ПОЛЕЙ I (INDEX) и B (BASE)

Значение	Index	Base
000		EAX
001		ECX
010		EDX
011		EBX
100	Нет индекса	ESP
101	EBP	если mod = 01 или 10, то EBP, если mod = 00, то базы нет
110		ESI
111		EDI



ФОРМИРОВАНИЕ ПОЛЯ SIB

ADD EAX,[ESP]



НЕПОСРЕДСТВЕННОЕ ЗНАЧЕНИЕ

- необязательное поле команды, которое может занимать 0, 1, 2 или 4 байта. Размер непосредственного операнда определен несколькими факторами:
- разрядностью программного сегмента;
- наличием в команде префикса OS = 66h;
- значением бита s, который позволяет сохранять в тексте программы только младший байт 16-ти или 32-х битного непосредственного операнда, если его значение попадает в диапазон для короткого целого числа со знаком



СМЕЩЕНИЕ

- Если в команде используется смещение, то поле `mod` должно принимать значения 01 или 10. В случае, если `mod = 01` используется 8-ми битное смещение, а при `mod = 10` – 16-ти или 32-х разрядное смещение.
- Если в команде используется прямая адресация, то требуется, чтобы поле `mod = 00`, а поле `R/M` принимало значение 110 для указания 16-ти разрядного смещения в 16-ти разрядных сегментах, и 101 – для 32-х разрядного смещения в 32-х разрядных сегментах.
- Использование 32-х или 16-ти разрядного смещения зависит от разрядности сегмента и наличия префикса `AS = 67h`. В случае, когда разрядность сегмента кода не совпадает с разрядностью сегмента данных, команда обязательно формируется с префиксом `AS`, а разрядность смещения устанавливается по разрядности сегмента кода.
- При использовании индексной адресации через 32-х разрядные регистры без указания регистра базы обязательно формируется 32-х разрядное смещение, даже если оно не было указано.
- При адресации через 32-х разрядные регистры с использованием регистра базы размер смещения может быть оптимизирован (хранится в виде одного байта), если его значение находится в диапазоне `-128..+127`.

