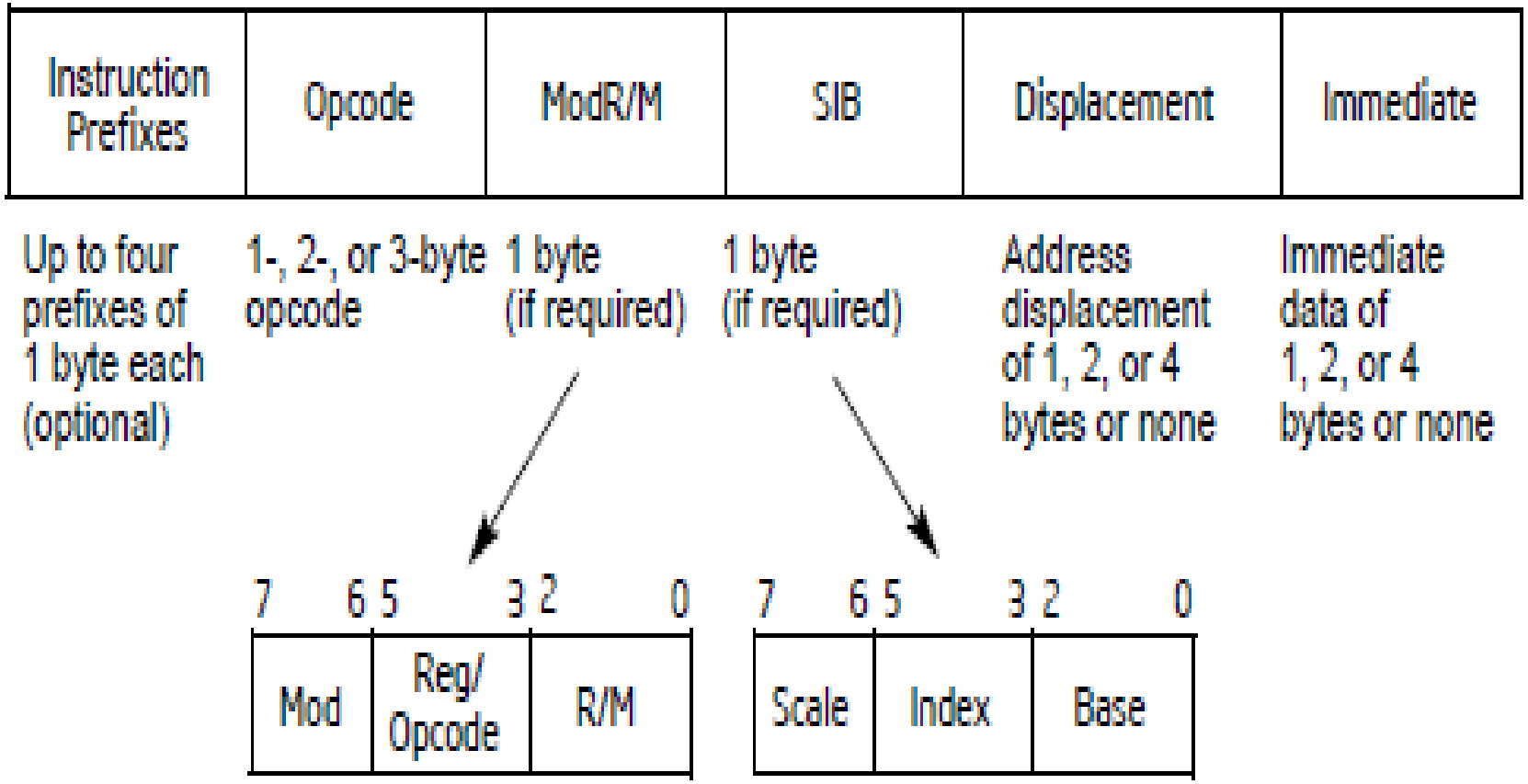
Система инструкции за микропроцесори IA-32

Формат на инструкциите

Инструкциите за Intel 64 и IA-32 архитектури се кодират с подмножество формати, показани на следващия слайд. Инструкциите се състоят от:

- Опционален префикс на инструкцията /в произволен ред/,
- Първичен операционен код /opcode/ (до 3 байта),
- Спецификатор за формат на адресация (ако се изисква), състоящ се от ModR/M байт
- SIB (Scale-Index-Base) байт,
- Отместване /displacement/ (ако се изисква),
- Непосредсвено данново поле (ако се изисква).

Формат на инструкциите



Префикси на инструкциите

Префиксът се ползва да модифицира изпълнението на една инструкция – онази, която следва префикса.

Прехфиксите на инструкциите се делят на 4 групи:

• Група 1 — Lock и repeat префикси:

• LOCK (F0H) въздействува на операцията, осигуряваща изключително използване на споделена памет в мултипроцесорно обкръжение

• repeat префикси (REPNE/REPNZ or REP or REPE/REPZ). Repeat префикс се прилага само на стрингови и i/o инструкции. Префиксите Repeat (F2H, F3H) предизвикват инструкцията да се повтори за всеки елемент на стринга.

• Група 2 — Смяна на сегмента /Segment override prefixes/. Той се ползва за смяна на сегмента по подразбиране Прилага се за всеки сегментен регистър.

Префикси на инструкциите

• Група 3 – Префикс за смяна на дължината на операнда /Operand-size override prefix/. Код 66H (66H е също задължителен за някои инструкции). Този префикс позволява една програма да превключва между 16- и 32-b дължина на операнда. Префиксът се ползва за избор на дължина на операнда, която не е установена по подразбиране.

• Група 4 - 67H—Префикс за смяна на дължината на адреса. /Address-size override prefix/. Този префикс позволява програмите да превключват между 16- и 32-b адресиране. Едната дължина се подразбира. Префиксът превключва към нестандартна дължина. Използването на този префикс и/или други недефинирани opcodes, когато операндите за инструкцията не са разположени в запазената област от паметта, може да предизвика непредсказуемо обкръжение.

За всяка инструкция може да се ползва само 1 префикс от всяка група в произволен ред.

Операционни кодове

- Първичният код на операцията (КОП) може да бъде дълъг 1, 2, или 3 В. По-малки полета могат да се дефинират вътре в първичния КОП. Такива полета дефинират посока на операцията, дължина на отместването, кодиране на регистър, условни кодове или разширение на знака. Кодиращите полета в КОП варират в зависимост от класа на операцията.

ModR/M и SIB байт

Много инструкции, които сочат операнд в паметта, имат байт специфициращ адресен формат (ModR/M байт), следващ след първичния КОП. ModR/M байтът съдържа 3 полета информация:

- *mod полето* се комбинира с r/m полето, за да се формират 32 възможни стойности: 8 регистъра и 24 методи на адресация.
- *reg/opcode* полето указва номер на регистър или допълнителни 3 бита информация от КОП. Смисълът на reg/opcode полето е указан в първичния КОП..
- *r/m* полето указва регистър като операнд или може да да се комбинира с mod полето за декодиране на метода на адресация. При някои инструкции се използва известна комбинация от *mod поле* и *r/m поле,* за да се изрази информацията на opcode.

ModR/M и SIB байт

Сигурното декодиране на ModR/M байта изисква втори адресен байт (SIB байт). Базово-индексен и мащаб-индексен формати при 32-b адресиране изискават SIB байт.

SIB байтът включва следните полета:

- *scale* полето указва мащаба.
- *index* полето указва номер на индексен регистър.
- *base* полето указва номер на базов регистър.

Отместване и непосредствени байтове

- Някои адресни форми включват отместване, незабавно следващо ModR/M байта (или SIB байт, ако присъствува). Ако отместването изисква, то може да бъде 1,2,4 байта.
- Ако инструкцията указва непосредствен операнд, операндът винаги следва някакви байтове за отместване. Непосредственият операнд може да бъде 1,2, 4 байта.

Пример

```
03 C3     ADD AX,BX

_____________________________________
0 0 0 0  0 0  1  1  1 1  0 0 0  0 1 1
------------  |  |  ---  -----  -----
    ADD op    | Word |    AX     BX
              Dest   |     |      |
                  Reg-Reg  |      |
                   mode    |      |
                       destination source
```

```
66 B8 00 00 00 00 00 00 12 00     MOV EAX,12H

_____________________________________________________________________
0 1 1 0 0 1 1 0  1 0 1 1  1  0 0 0   0000 0000 0000 0000 0000 0000 0001 001
--------------  -------  |  -----   ------------------------------------
 32bit prefix    MOV op  Word  AX          immediate data
```

Полета при Pentium инструкции

0 - 3 bytes	1 - 2 bytes	0 - 1 bytes	0 - 1 bytes	0, 1, 2 or 8 bytes
Prefix	Opcode D W	Operand MOD REG R/M	S I B	Displacement / Immediate Data

2EH	CSEG
3EH	DSEG
36H	SSEG
26H	ESEG
64H	FSEG
65H	GSEG
66H	32bit mode
B8H	16bit mode
F0H	Lock
F3H	REP

00	Memory
01	Memory+d8
10	Mem+d32/d16
11	Register

	EAX	ECX	EDX	EBX	ESP		ESI	EDI
[EAX]	00	01	02	03	04	05	06	07
[ECX]	08	09	0A	0B	0C	0D	0E	0F
[EDX]	10	11	12	13	14	15	16	17
[EBX]	18	19	1A	1B	1C	1D	1E	1F
	20	21	22	23	24	25	26	27
[EBP]	28	29	2A	2B	2C	2D	2E	2F
[ESI]	30	31	32	33	34	35	36	37
[EDI]	38	39	3A	3B	3C	3D	3E	3F

0	REG is source
1	REG is destination

	W=0	W=1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

R/M	MOD=00	MOD=01	MOD=10
000	(BX+SI)	(BX+SI+d8)	(BX+SI+d16)
001	(BX+DI)	(BX+DI+d8)	(BX+DI+d16)
010	(BP+SI)	(BP+SI+d8)	(BP+SI+d16)
011	(BP+DI)	(BP+DI+d8)	(BP+DI+d16)
100	(SI)	(SI+d8)	(SI+d16)
101	(DI)	(DI+d8)	(DI+d16)
110	direct	(BP+d8)	(BP+d16)
111	(BX)	(BX+d8)	(BX+d16)

d8 - a byte of data

d16 - a 2 byte word

Pentium instruction code fields

Инструкции за обща употреба

Инструкциите за обща употреба позволяват операции за базово прехвърляне на данни, аритметични, логически операции, операции по предаване на управлението, стрингови операции, които програмистите ползват общо за писане на приложен и системен софтуер за Intel 64 and IA-32 процесори.

Те оперират над данни разположени в паметта, в регистрите с общо предназначение (EAX, EBX, ECX, EDX, EDI, ESI, EBP и ESP) и в EFLAGS регистър.

Те също така оперират с адресна информация в паметта, регистрите с общо предназначение и сегментните регистри (CS, DS, SS, ES, FS и GS).

Инструкции за обща употреба

Тази група инструкции включва:

- За прехвърляне на данни. Инструкциите за прехвърляне на данни между паметта и регистрите с общо предназначение и сегментните регистри. Те също извършват специфични операции като условно прехвърляне на данни, достъп до стека и преобразуване на данни.
- За двоична целочислена аритметика. Тези инструкции извършват базисни двоични целочислени пресмятания върху байт, дума или двойна дума цели числа, разположени в паметта и/или в регистрите с общо предназначение.
- За десетична аритметика. Тези инструкции извършват десетична аритметика върху двоично кодирани десетични (BCD) данни.

Инструкции за обща употреба

- За логически операции. Тези инструкции извършват базовите AND, OR, XOR, и NOT логически операции на байт, дума и двойна дума.
- За изместване и ротация. Тези инструкции изместват и ротират наляво и надясно битовете в операнди дума и двойна дума.
- За битови и байтови операции. Тези инструкции тестват и модифицират индивидуални битове в операнди с размер дума и двойна дума. Байтовите инструкции установяват стойност на байтов операнд, за да индицират състоянието на флаговете в EFLAGS регистъра.
- За управление на програмата. Тези инструкции извършват преход, условен преход, цикъл за извикване и връщане от подрограма за управление на хода на програмата.

Инструкции за обща употреба

- За стрингови операции. Тези инструкции оперират над стрингове от байтове, позволявайки те да бъдат премествани в паметта..
- За I/O инструкции. Тези инструкции прехвърлят данни между процесорните I/O портове и регистър или паметта.
- За управление на флаговете. Тези инструкции боравят с флаговете от EFLAGS регистъра.
- За операции със сегментните регистри. Тези инструкции позволяват да се зареждат в сегментните регистри далечни указатели.
- Enter и Leave инструкции. Тези инструкции осъществяват поддръжка чрез машинния език на извикване на процедури от блоково-структурните езици. Това са инструкциите ENTER – вход на процедура за език от високо ниво и LEAVE – изход на процедура от високо ниво.
- Смесени инструкции. Тези инструкции съдържат инструкции за зареждане на ефективен адрес, изпълнение на празна операция, и извличане на информация за идентификация на процесора.

X87 FPU инструкции

x87 FPU инструкциите се изпълняват от процесорното x87 FPU. Те оперират с операнди с плаваща запетая, целочислени и двоично-десетични операнди.

Те се делят на следните подгрупи:

- За прехвърляне на данни. Те преместват стойности с плаваща запетая, целочислени, и BCD между паметта и x87 FPU регистрите. Те също така правят условни премествания на операнди с плаваща запетая.
- За базови аритметични операции. Базовите аритметични инструкции извършват базови аритметични операции над целочислени и операнди с плаваща запетая.
- Инструкции за сравнение. Те изследват и проверяват операнди с плаваща точка и целочислени такива.
- x87 FPU трансцедентни инструкции. Те изпълняват базови тригонометрични операции над операнди с плаваща точка.

X87 FPU инструкции

- за зареждане на x87 FPU с константа. Те зареждат общи константи като π, в x87 регистрите с плаваща точка.
- за управление на FPU. Те оперират върху x87 FPU регистровия стек и записват и възстановяват състоянието на x87 FPU.

X87 FPU и управление на SIMD състоянието

За управление на състоянието в IA-32 архитектурата са въведени 2 инструкции при фамилия процесори Pentium II:

- FXSAVE за съхраняване x87 FPU и SIMD състоянието
- FXRSTOR за възстановяване на x87 FPU и SIMD състоянието

Отначало тези инструкции са оперирали само върху x87 FPU (и MMX) регистрите, за да съхраняват и възстановяват бързо x87 FPU и MMX състоянието. С въвеждането на SSE разширението в процесорна фамилия Pentium III, тези инструкции са разширени също така да съхраняват и възстановяват състоянието на XMM и MXCSR регистрите.

Intel 64 архитектурата също поддържа тези инструкции.

MMX инструкции

- 4 разширения са въведени в IA-32 архитектурата за извършване на single-instruction multiple-data (SIMD) операции. Тези разширения включват **MMX** технологията, **SSE** разширенията, **SSE2** разширенията, и **SSE3** разширенията.
- MMX инструкциите оперират над пакетиран байт, дума, двойна дума, четворна дума за представяне на целочислени операнди в паметта, MMX регистрите и/или в регистрите с общо предназначение.
- MMX инструкциите се изпълняват от процесорите Intel 64 и IA-32 , които поддържат MMX технологията. Поддръжката на тази технология може да бъде диагностицирана чрез инструкцията CPUID.

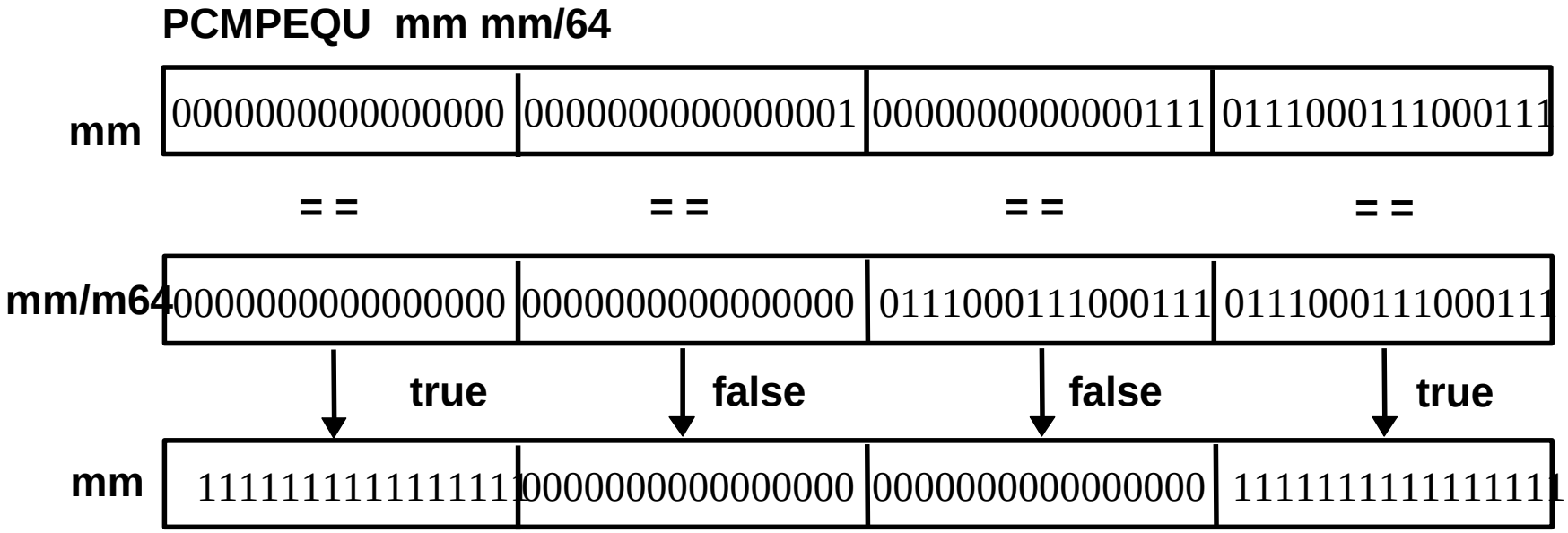
MMX инструкции

MMX инструкциите се делят на следните групи:

- За прехвърляне на данни/data transfer/. Те прехвърлят двойна и четворна дума операнди между MMX регистрите и между MMX регистрите и паметта.
- За преобразуване /Conversion/. Те пакетират и разпакетират байтове, думи и двойни думи.
- За пакетна аритметика /packed arithmetic/. Тези инструкции извършват пакетирана целочислена аритметика върхуцели числа, представени в пакетирани байтове, думи, двойни думи.
- За сравнение /Comparison/. Те сравняват пакетирани байтове, думи и двойни думи.

MMX инструкции

- Логически /Logical/. Те извършват AND, AND NOT, OR, XOR операции над четворни думи.
- За изместване и ротация /shift and rotate/. Те изместват и ротират пакетирани байтове, думи или двойни думи или четворни думи в 64-b операнди.
- За управление на състоянието /state management instructions/. Инструкцията EMMS изтрива MMX състоянието от MMX регистрите.

Пример за MMX инструкция

PCMPEQB Compare packed bytes for equal
/Сравняване на пакетирани байтове за еднаквост/