



B

**IA MMX™
Instruction Formats
and Encodings**





APPENDIX B

IA MMX™ INSTRUCTION FORMATS AND ENCODINGS

B.1. INSTRUCTION FORMATS

All MMX instructions, except the EMMS instruction, use the same format similar as the two-byte Intel Architecture integer operations. Details of subfield encodings within these formats are presented below.

Table B-1. Encoding of Granularity of Data (gg) Field

gg	Granularity of Data
00	packed bytes
01	packed words
10	packed doublewords
11	quadword

Table B-2. Encoding of 32-bit General Purpose (reg) Field for Register-to-Register Operations

reg Field	Register Selected
000	EAX
001	ECX
010	EDX
011	EBX
100	ESP
101	EBP
110	ESI
111	EDI

NOTE: For register-to-register operations, the decoding of integer registers is independent of processor mode. For register-to-memory operations, the effective address is calculated based on the processor mode in effect.

Table B-3. Encoding of 64-bit MMX™ Register (mmxreg) Field

mmxreg Field	MMX Register Selected
000	mm0
001	mm1
010	mm2
011	mm3
100	mm4
101	mm5
110	mm6
111	mm7

For more details, see Table 25-2, Table 25-3, and Appendix F of the *Pentium® Processor Family Developer's Manual*.

B.2. INSTRUCTION ENCODINGS AND DATATYPE CROSS-REFERENCE

For each MMX instruction, Table B-4 lists instruction encodings and the datatypes supported—byte (B), word (W), doubleword (DW), and quadword (QW).

O	= output
I	= input
S	= signed saturation
U	= unsigned saturation
n/a	= not applicable

Figure B-1. Key to Codes for Datatype Cross-Reference

Table B-4. IA MMX™ Instruction Formats and Encodings

Instruction	Format	B	W	DW	QW
EMMS - Empty MMX state	0000 1111:01110111	n/a	n/a	n/a	n/a
MOVD - Move doubleword		N	N	Y	N
reg to mmxreg	0000 1111:01101110: 11 mmxreg reg				
reg from mmxreg	0000 1111:01111110: 11 mmxreg reg				
mem to mmxreg	0000 1111:01101110: mod mmxreg r/m				
mem from mmxreg	0000 1111:01111110: mod mmxreg r/m				
MOVQ - Move quadword		N	N	N	Y
mmxreg2 to mmxreg1	0000 1111:01101111: 11 mmxreg1 mmxreg2				
mmxreg2 from mmxreg1	0000 1111:01111111: 11 mmxreg1 mmxreg2				
mem to mmxreg	0000 1111:01101111: mod mmxreg r/m				
mem from mmxreg	0000 1111:01111111: mod mmxreg r/m				

Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PACKSSDW ¹ - Pack dword to word data (signed with saturation)		n/a	O	I	n/a
mmxreg2 to mmxreg1	0000 1111:01101011: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:01101011: mod mmxreg r/m				
PACKSSWB ¹ - Pack word to byte data (signed with saturation)		O	I	n/a	n/a
mmxreg2 to mmxreg1	0000 1111:01100011: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:01100011: mod mmxreg r/m				
PACKUSWB ¹ - Pack word to byte data (unsigned with saturation)		O	I	n/a	n/a
mmxreg2 to mmxreg1	0000 1111:01100111: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:01100111: mod mmxreg r/m				
PADD - Add with wrap-around		Y	Y	Y	N
mmxreg2 to mmxreg1	0000 1111: 111111gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111: 111111gg: mod mmxreg r/m				

Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PADDQ - Add signed with saturation		Y	Y	N	N
mmxreg2 to mmxreg1	0000 1111: 111011gg: 11 mmxreg1 mmxreg2				
memory to reg	0000 1111: 111011gg: mod reg r/m				
PADDUS - Add unsigned with saturation		Y	Y	N	N
mmxreg2 to mmxreg1	0000 1111: 110111gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111: 110111gg: mod mmxreg r/m				
PAND - Bitwise And		N	N	N	Y
mmxreg2 to mmxreg1	0000 1111:11011011: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11011011: mod mmxreg r/m				
PANDN - Bitwise AndNot		N	N	N	Y
mmxreg2 to mmxreg1	0000 1111:11011111: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11011111: mod mmxreg r/m				
PCMPEQ - Packed compare for equality		Y	Y	Y	N
mmxreg2 with mmxreg1	0000 1111:011101gg: 11 mmxreg1 mmxreg2				
memory with mmxreg	0000 1111:011101gg: mod mmxreg r/m				

Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PCMPGT - Packed compare greater (signed)		Y	Y	Y	N
mmxreg2 with mmxreg1	0000 1111:011001gg: 11 mmxreg1 mmxreg2				
memory with mmxreg	0000 1111:011001gg: mod mmxreg r/m				
PMADD - Packed multiply add		n/a	I	O	n/a
mmxreg2 to mmxreg1	0000 1111:11110101: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11110101: mod mmxreg r/m				
PMULH - Packed multiplication		N	Y	N	N
mmxreg2 to mmxreg1	0000 1111:11100101: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11100101: mod mmxreg r/m				
PMULL - Packed multiplication		N	Y	N	N
mmxreg2 to mmxreg1	0000 1111:11010101: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11010101: mod mmxreg r/m				
POR - Bitwise Or		N	N	N	Y
mmxreg2 to mmxreg1	0000 1111:11101011: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11101011: mod mmxreg r/m				

Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PSLL ² - Packed shift left logical		N	Y	Y	Y
mmxreg2 by mmxreg1	0000 1111:111100gg: 11 mmxreg1 mmxreg2				
mmxreg by memory	0000 1111:111100gg: 11 mmxreg r/m				
mmxreg by immediate	0000 1111:011100gg: 11 110 mmxreg: imm8 data				
PSRA ² - Packed shift right arithmetic		N	Y	Y	N
mmxreg2 by mmxreg1	0000 1111:1111000gg: 11 mmxreg1 mmxreg2				
mmxreg by memory	0000 1111:1111000gg: 11 mmxreg r/m				
mmxreg by immediate	0000 1111:011100gg: 11 100 mmxreg: imm8 data				
PSRL ² - Packed shift right logical		N	Y	Y	Y
mmxreg2 by mmxreg1	0000 1111:110100gg: 11 mmxreg1 mmxreg2				
mmxreg by memory	0000 1111:110100gg: 11 mmxreg r/m				
mmxreg by immediate	0000 1111:011100gg: 11 010 mmxreg: imm8 data				
PSUB - Subtract with wrap-around		Y	Y	Y	N
mmxreg2 to mmxreg1	0000 1111:111110gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:111110gg: mod mmxreg r/m				

Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PSUBS - Subtract signed with saturation		Y	Y	N	N
mmxreg2 to mmxreg1	0000 1111:111010gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:111010gg: mod mmxreg r/m				
PSUBUS - Subtract unsigned with saturation		Y	Y	N	N
mmxreg2 to mmxreg1	0000 1111:1110110gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:1110110gg: mod mmxreg r/m				
PUNPCKH - Unpack high data to next larger type		Y	Y	Y	N
mmxreg2 to mmxreg1	0000 1111:011010gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:011010gg: mod mmxreg r/m				
PUNPCKL - Unpack low data to next larger type		Y	Y	Y	N
mmxreg2 to mmxreg1	0000 1111:011000gg: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:011000gg: mod mmxreg r/m				



Table B-4. IA MMX™ Instruction Formats and Encodings (Contd.)

Instruction	Format	B	W	DW	QW
PXOR - Bitwise Xor		N	N	N	Y
mmxreg2 to mmxreg1	0000 1111:11101111: 11 mmxreg1 mmxreg2				
memory to mmxreg	0000 1111:11101111: mod mmxreg r/m				

NOTE:

1. The PACK instructions perform saturation from signed packed data of one type to signed or unsigned data of the next smaller type.
2. The format of shift instructions has one additional format to support shifting by immediate shift-counts. The shift operations are not supported equally for all data types.



