D

IA MMX™ Instruction Set Opcode Map

APPENDIX D
IA MMX™ INSTRUCTION SET OPCODE MAP

The detailed encodings of the Intel Architecture MMX™ instructions are listed in the shaded boxes of the Opcode Map tables below. All MMX instructions, except the EMMS instruction, use the same format as the two-byte Intel Architecture integer operations.

All blanks in the Opcode Map are reserved and should not be used. Do not depend on the operation of unspecified opcodes. 0F0Bh or 0FB9h should be used when deliberately generating an illegal opcode exception.

Key to Abbreviations

Operands are identified by a two-character code of the form Zz. The first character, an uppercase letter, specifies the addressing method; the second character, a lowercase letter, specifies the type of operand. For opcodes with two operands, the left code refers to the destination operand and the right code refers to the source operand. All MMX instructions, except the EMMS instruction, reference and operate on two operands.

Codes for Addressing Method

C The reg field of the ModR/M byte selects a control register; e.g., MOV (0F20, 0F22).

D The reg field of the ModR/M byte selects a debug register; e.g., MOV (0F21, 0F23).

E A ModR/M byte follows the opcode and specifies the operand. The operand is either a general register or a memory address. If it is a memory address, the address is computed from a segment register and any of the following values: a base register, an index register, a scaling factor, a displacement.

G The reg field of the ModR/M byte selects a general register; e.g., AX(000).

I Immediate data. The value of the operand is encoded in subsequent bytes of the instruction.

M The ModR/M byte may refer only to memory; e.g., LSS, LFS, LGS, CMPXCHG8B.

P The reg field of the ModR/M byte selects a packed quadword MMX register.

Q A ModR/M byte follows the opcode and specifies the operand. The operand is either an MMX register or a memory address. If it is a memory address, the address is computed

from a segment register and any of the following values: a base register, an index register, a scaling factor, a displacement.

R The mod field of the ModR/M byte may refer only to a general register; e.g., MOV (0F20-0F24, 0F26).

Codes for Operand Type

b Byte (regardless of operand size attribute).

d Doubleword (regardless of operand size attribute).

p 32-bit or 48-bit pointer, depending on operand size attribute.

q Quardword (regardless of operand size attribute).

s Six-byte pseudo-descriptor.

v Word or doubleword, depending on operand size attribute.

w Word (regardless of operand size attribute).

Register Codes

When an operand is a specific register encoded in the opcode, the register is identified by its name, for example: AX, CL, or ESI. The name of the register indicates whether the register is 32-bits, 16-bits, or 8-bits wide. A register identifier of the form eXX is used when the width of the register depends on the oeprand size attribute; for example, eAX indicates that the AX register is used when the operand size attribute is 16 and the EAX register is used when the operand size attribute is 32.

Table D-1. Opcode Map (First Byte is 0FH)

	0	1	2	3	4	5	6	7
0	GRP 6		LAR Gv, Ew	LSL Gv, Ew			CLTS	
1								
2	MOV Rd, Cd	MOV Rd, Dd	MOV Cd, Rd	MOV Dd, Rd				
3	WRMSR	RDTSC	RDMSR					
4								
5								
6	PUNPCKLBW Pq, Qd	PUNPCKLWD Pq, Qd	PUNPCKLDQ Pq, Qd	PACKSSWB Pq, Qq	PCMPGTB Pq, Qq	PCMPGTW Pq, Qq	PCMPGTD Pq, Qq	PACKUSWB Pq, Qq
7		Grp A			PCMPEQB Pq, Qq	PCMPEQW Pq, Qq	PCMPEQD Pq, Qq	EMMS
		PSHIMW	PSHIMD	PSHIMQ				
8	Long-displacement jump on condition (Jv)							
	JO	JNO	JB	JNB	JZ	JNZ	JBE	JNBE
9	Byte Set on condition (Eb)							
	SETO	SETNO	SETB	SETNB	SETZ	SETNZ	SETBE	SETNBE
A	PUSH FS	POP FS	CPUID	BT Ev, Gv	SHLD Ev, Gv, Ib	SHLD Ev, Gv, CL		
B	CMPXCH Eb, Gb	CMPXCH Ev, Gv	LSS Mp	BTR Ev, Gv	LFS Mp	LGS Mp	MOVZX Gv, Eb	MOVZX Gv, Ew
C	XADD Eb, Gb	XADD Ev, Gv						GRP 9
D		PSRLW Pq, Qq	PSRLD Pq, Qq	PSRLQ Pq, Qq		PMULLW Pq, Qq		
E		PSRAW Pq, Qq	PSRAD Pq, Qq			PMULHW Pq, Qq		
F		PSLLW Pq, Qq	PSLLD Pq, Qq	PSLLQ Pq, Qq		PMADDWD Pq, Qq		

Table D-1. Opcode Map (First Byte is 0FH) (Contd)

	8	9	A	B	C	D	E	F
0	INVD	WB INVD		Illegal opcode				
1								
2								
3								
4								
5								
6	PUNPCKHBW Pq, Qq	PUNPCKHWD Pq, Qq	PUNPCKHDQ Pq, Qq	PACKSSDW Pq, Qq			MOVD Pq, Ed	MOVQ Pq, Qq
7							MOVD Ed, Pd	MOVQ Qq, Pq
8	Long-displacement jump on condition (Jv)							
	JS	JNS	JP	JNP	JL	JNL	JLE	JNLE
	Byte Set on condition (jv)							
9	SETS Eb	SETNS Eb	SETP Eb	SETNP Eb	SETL Eb	SETNL Eb	SETLE Eb	SETNLE Eb
A	PUSH GS	POP GS	RSM	BTS Ev, Gv	SHRD Ev, Gv, Ib	SHRD Ev, Gv, CL		IMUL Gv, Ev
B		Illegal opcode	GRP 8 Ev, Ib	BTC Ev, Gv	BSF Gv, Ev	BSR Gv, Ev	MOVSX Gv, Eb	MOVSX Gv, Ew
C	BSWAP EAX	BSWAP ECX	BSWAP EDX	BSWAP EBX	BSWAP ESP	BSWAP EBP	BSWAP ESI	BSWAP EDI
D	PSUBUSB Pq, Qq	PSUBUSW Pq, Qq		PAND Pq, Qq	PADDUSB Pq, Qq	PADDUSW Pq, Qq		PANDN Pq, Qq
E	PSUBSB Pq, Qq	PSUBSW Pq, Qq		POR Pq, Qq	PADDSB Pq, Qq	PADDSW Pq, Qq		PXOR Pq, Qq
F	PSUBB Pq, Qq	PSUBW Pq, Qq	PSUBD Pq, Qq		PADDB Pq, Qq	PADDW Pq, Qq	PADDD Pq, Qq	

Table D-2. Opcodes Determined by Bits 5, 4, 3 of Mod R/M Byte

mod	nnn	R/M

Group	000	001	010	011	100	101	110	111
1	ADD	OR	ADC	SBB	AND	SUB	XOR	CMP
2	ROL	ROR	RCL	RCR	SHL SAL	SHR		SAR
3	TEST Ib/Iv		NOT	NEG	MUL AL/eAX	IMUL AL/eAX	DIV AL/eAX	IDIV AL/eAX
4	INC Eb	DEC Eb						
5	INC Ev	DEC Ev	CALL Ev	CALL Ep	JMP Ev	JMP Ep	PUSH Pv	
6	SLDT Ew	STR Ew	LLDT Ew	LTR Ew	VERR Ew	VERW Ew		
7	SGDT Ms	SIDT Ms	LGDT Ms	LIDT Ms	SMSW Ew		LMSW Ew	INVLPG
8					BT	BTS	BTR	BTC
9		CMPXCH 8BMq						
A			PSRL Pq, Ib		PSRA Pq, Ib		PSLL Pq, Ib	