
APPENDIX B

Instruction Set Summary

The instruction set summary contains a complete alphabetical listing of the entire 8086–Pentium 4 instruction set. The coprocessor and MMX instructions are listed in Chapter 14 and are not repeated in this appendix. The SIMD instructions appear at the end of this appendix after the main instruction set summary.

Each instruction entry lists the mnemonic opcode plus a brief description of the purpose of the instruction. Also listed is the binary machine language coding of each instruction and any other data required to form the instruction, such as the displacement or immediate data. Listed to the right of each binary machine language version of the instruction are the flag bits and any change that might occur for the instruction. The flags are described in the following manner: A blank indicates no effect or change; a ? indicates a change with an unpredictable outcome; a * indicates a change with a predictable outcome; a 1 indicates the flag is set; and a 0 indicates that the flag is cleared. If the flag bits ODITSZAPC are not illustrated with an instruction, the instruction does not modify any of these flags.

Before the instruction listing begins, some information about the bit settings in binary machine language versions of the instructions is presented. Table B-1 lists the modifier bits, coded as OO in the instruction listing.

Table B-2 lists the memory-addressing modes available using a register field coding of mmm. This table applies to all versions of the microprocessor, as long as the operating mode is 16 bits.

Table B-3 lists the register selections provided by the rrr field in an instruction. This table includes the register selections for 8-, 16-, and 32-bit registers.

Table B-4 lists the segment register bit assignment (rrr) found with the MOV, PUSH, and POP instructions.

TABLE B-1 The modifier bits, coded as oo in the instruction listing.

oo	Function
00	If mmm = 110, a displacement follows the opcode; otherwise no displacement is used.
01	An 8-bit signed displacement follows the opcode.
10	A 16- or 32-bit signed displacement follows the opcode.
11	mmm specifies a register instead of an addressing mode.

TABLE B–2 The 16-bit register/memory (mmm) field description.

<i>mmm</i>	16-Bit
000	DS:[BX+SI]
001	DS:[BX+DI]
010	SS:[BP+SI]
011	SS:[BP+DI]
100	DS:[SI]
101	DS:[DI]
110	SS:[BP]
111	DS:[BX]

TABLE B–3 The register (rrr) field.

<i>rrr</i>	<i>W=0</i>	<i>W=1 (16-Bit)</i>	<i>W=1 (32-Bit)</i>
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

TABLE B–4 Register field assignments (rrr) for the segment registers.

<i>rrr</i>	<i>Segment Register</i>
000	ES
001	CS
010	SS
011	DS
100	FS
101	GS

When the 80386–Core2 are used, some of the definitions provided in Table B–1 through B–3 change. See Tables B–5 and B–6 for these changes as they apply to the 80386–Core2 microprocessors.

TABLE B–5 Index registers specified by rrr when the 80386–Core2 are operated in 32-bit mode.

<i>rrr</i>	<i>Index Register</i>
000	DS:[EAX]
001	DS:[ECX]
010	DS:[EDX]
011	DS:[EBX]
100	(see Table B–6)
101	SS:[EBP]
110	DS:[ESI]
111	DS:[EDI]

TABLE B–6 Possible combinations of oo, mmm, and rrr for the 80386–Core2 microprocessors using 32-bit addressing.

<i>oo</i>	<i>mmm</i>	<i>rrr</i> (<i>Base in Scaled-Index Byte</i>)	<i>Addressing Mode</i>
00	000	—	DS:[EAX]
00	001	—	DS:[ECX]
00	010	—	DS:[EDX]
00	011	—	DS:[EBX]
00	100	000	DS:[EAX+scaled index]
00	100	001	DS:[ECX+scaled index]
00	100	010	DS:[EDX+scaled index]
00	100	011	DS:[EBX+scaled index]
00	100	100	SS:[ESP+scaled index]
00	100	101	DS:[disp32+scaled index]
00	100	110	DS:[ESI+scaled index]
00	100	111	DS:[EDI+scaled index]
00	101	—	DS:disp32
00	110	—	DS:[ESI]
00	111	—	DS:[EDI]
01	000	—	DS:[EAX+disp8]
01	001	—	DS:[ECX+disp8]
01	010	—	DS:[EDX+disp8]
01	011	—	DS:[EBX+disp8]
01	100	000	DS:[EAX+scaled index+disp8]
01	100	001	DS:[ECX+scaled index+disp8]
01	100	010	DS:[EDX+scaled index+disp8]
01	100	011	DS:[EBX+scaled index+disp8]
01	100	100	SS:[ESP+scaled index+disp8]
01	100	101	SS:[EBP+scaled index+disp8]
01	100	110	DS:[ESI+scaled index+disp8]
01	100	111	DS:[EDI+scaled index+disp8]
01	101	—	SS:[EBP+disp8]
01	110	—	DS:[ESI+disp8]
01	111	—	DS:[EDI+disp8]
10	000	—	DS:[EAX+disp32]
10	001	—	DS:[ECX+disp32]
10	010	—	DS:[EDX+disp32]
10	011	—	DS:[EBX+disp32]
10	100	000	DS:[EAX+scaled index+disp32]
10	100	001	DS:[ECX+scaled index+disp32]
10	100	010	DS:[EDX+scaled index+disp32]
10	100	011	DS:[EBX+scaled index+disp32]
10	100	100	SS:[ESP+scaled index+disp32]
10	100	101	SS:[EBP+scaled index+disp32]
10	100	110	DS:[ESI+scaled index+disp32]
10	100	111	DS:[EDI+scaled index+disp32]
10	101	—	SS:[EBP+disp32]
10	110	—	DS:[ESI+disp32]
10	111	—	DS:[EDI+disp32]

Note: disp8 = 8-bit displacement and disp32 = 32-bit displacement.

In order to use the scaled-index addressing modes listed in Table B–6, code oo and mmm in the second byte of the opcode. The scaled-index byte is usually the third byte and contains three fields. The leftmost two bits determine the scaling factor (00 = $\times 1$, 01 = $\times 2$, 10 = $\times 4$, or 11 = $\times 8$). The next three bits toward the right contain the scaled-index register number (this is obtained from Table B–5). The rightmost three bits are from the mmm field listed in Table B–6. For example, the MOV AL,[EBX+2*ECX] instruction has a scaled-index byte of 01 001 011, where 01 = X_2 , 001 = ECX, and 011 = EBX.

Some instructions are prefixed to change the default segment or to override the instruction mode. Table B–7 lists the segment and instruction mode override prefixes with append at the beginning of an instruction if they are used to form the instruction. For example, the MOV AL,ES:[BX] instruction used the extra segment because of the override prefix ES:.

In the 8086 and 8088 microprocessors, the effective address calculation required additional clocks that are added to the times in the instruction set summary. These additional times are listed in Table B–8. No such times are added to the 80286–Core2. Note that the instruction set summary does not include clock times for the Pentium Pro through the Core2. Intel has not released these times and has decided that the RDTSC instruction can be used to have the microprocessor count the number of clocks required for a given application. Even though the timings do not appear for these new microprocessors, they are very similar to the Pentium, which can be used as a guide.

TABLE B–7 Override prefixes.

Prefix Byte	Purpose
26H	ES: segment override
2EH	CS: segment override
36H	SS: segment override
3EH	DS: segment override
64H	FS: segment override
65H	GS: segment override
66H	Memory operand instruction mode override
67H	Register operand instruction mode override

TABLE B–8 Effective address calculations for the 8086 and 8088 microprocessors.

Type	Clocks	Example Instruction
Base or index	5	MOV CL,[DI]
Displacement	3	MOV AL,DATA1
Base plus index	7	MOV AL,[BP+SI]
Displacement plus base or index	9	MOV DH,[DI+20H]
Base plus index plus displacement	11	MOV CL,[BX+DI+2]
Segment override	ea + 2	MOV AL,ED:[DI]

INSTRUCTION SET SUMMARY

AAA	ASCII adjust AL after addition								
00110111		O	D	I	T	S	Z	A	P C
Example		?		?	?	*	?	*	*
		Microprocessor				Clocks			
AAA		8086				8			
		8088				8			
		80286				3			
		80386				4			
		80486				3			
		Pentium–Core2							
AAD	ASCII adjust AX before division								
11010101 00001010		O	D	I	T	S	Z	A	P C
Example		?		?	?	*	*	?	*
		Microprocessor				Clocks			
AAD		8086				60			
		8088				60			
		80286				14			
		80386				19			
		80486				14			
		Pentium–Core2				10			
AAM	ASCII adjust AX after multiplication								
11010100 00001010		O	D	I	T	S	Z	A	P C
Example		?		?	?	*	*	?	*
		Microprocessor				Clocks			
AAM		8086				83			
		8088				83			
		80286				16			
		80386				17			
		80486				15			
		Pentium–Core2				18			

AAS ASCII adjust AL after subtraction	
00111111	O D I T S Z A P C ? ? * ? *
Example	Microprocessor Clocks
AAS	8086 8
	8088 8
	80286 3
	80386 4
	80486 3
	Pentium–Core2 3
ADC Addition with carry	
000100dw oorrrmmm disp	O D I T S Z A P C * * * * *
Format	Examples
ADC reg,reg	ADC AX,BX ADC AL,BL ADC EAX,EBX ADC CX,SI ADC ESI,EDI
	8086 3
	8088 3
	80286 3
	80386 3
	80486 1
	Pentium–Core2 1 or 3
ADC mem,reg	ADC DATAY,AL ADC LIST,SI ADC DATA2[DI],CL ADC [EAX],BL ADC [EBX+2*ECX],EDX
	8086 16 + ea
	8088 24 + ea
	80286 7
	80386 7
	80486 3
	Pentium–Core2 1 or 3

ADC reg,mem	ADC BL,DATA1 ADC SI,LIST1 ADC CL,DATA2[SI] ADC CX,[ESI] ADC ESI,[2*ECX]	8086	9 + ea
		8088	13 + ea
		80286	7
		80386	6
		80486	2
		Pentium–Core2	1 or 2
100000sw oo010mmm disp data			
Format	Examples	Microprocessor	Clocks
ADC reg,imm	ADC CX,3 ADC DI,1AH ADC DL,34H ADC EAX,12345 ADC CX,1234H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1 or 3
ADC mem,imm	ADC DATA4,33 ADC LIST,'A' ADC DATA3[DI],2 ADC BYTE PTR[EBX],3 ADC WORD PTR[DI],669H	8086	17 + ea
		8088	23 + ea
		80286	7
		80386	7
		80486	3
		Pentium–Core2	1 or 3
ADC acc,imm	ADC AX,3 ADC AL,1AH ADC AH,34H ADC EAX,2 ADC AL,'Z'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1

ADD mem,imm	ADD DATA4,33 ADD LIST,'A' ADD DATA3[DI],2 ADD BYTE PTR[EBX],3 ADD WORD PTR[DI],669H	8086	17 + ea	
		8088	23 + ea	
		80286	7	
		80386	7	
		80486	3	
		Pentium–Core2	1 or 3	
ADD acc,imm	ADD AX,3 ADD AL,1AH ADD AH,34H ADD EAX,2 ADD AL,'Z'	8086	4	
		8088	4	
		80286	3	
		80386	2	
		80486	1	
		Pentium–Core2	1	
AND Logical AND				
Format		O D I T S Z A P C 0 * * ? * 0		
Examples		Microprocessor		
AND reg,reg	AND CX,BX AND DL,BL AND ECX,EBX AND BP,SI AND EDX,EDI	8086	3	
		8088	3	
		80286	2	
		80386	2	
		80486	1	
		Pentium–Core2	1 or 3	
AND mem,reg	AND BIT,AL AND LIST,DI AND DATAZ[BX],CL AND [EAX],BL AND [ESI+4*ECX],EDX	8086	16 + ea	
		8088	24 + ea	
		80286	7	
		80386	7	
		80486	3	
		Pentium–Core2	1 or 3	

AND reg,mem	AND BL,DATAW AND SI,LIST AND CL,DATAQ[SI] AND CX,[EAX] AND ESI,[ECX+43H]	8086	9 + ea
		8088	13 + ea
		80286	7
		80386	6
		80486	2
		Pentium–Core2	1 or 2
100000sw oo100mmm disp data			
Format	Examples	Microprocessor	Clocks
AND reg,imm	AND BP,1 AND DI,10H AND DL,34H AND EBP,1345H AND SP,1834H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1 or 3
AND mem,imm	AND DATA4,33 AND LIST,'A' AND DATA3[DI],2 AND BYTE PTR[EBX],3 AND DWORD PTR[DI],66H	8086	17 + ea
		8088	23 + ea
		80286	7
		80386	7
		80486	3
		Pentium–Core2	1 or 3
AND acc,imm	AND AX,3 AND AL,1AH AND AH,34H AND EAX,2 AND AL,'r'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1

ARPL Adjust requested privilege level		O	D	I	T	S	Z	A	P	C
		*								
Format	Examples	Microprocessor					Clocks			
ARPL reg,reg	ARPL AX,BX	8086	—							
	ARPL BX,SI	8088	—							
	ARPL AX,DX	80286	10							
	ARPL BX,AX	80386	20							
	ARPL SI,DI	80486	9							
		Pentium–Core2	7							
ARPL mem,reg	ARPL DATA,Y,AX	8086	—							
	ARPL LIST,DI	8088	—							
	ARPL DATA3[DI],CX	80286	11							
	ARPL [EBX],AX	80386	21							
	ARPL [EDX+4*ECX],BP	80486	9							
		Pentium–Core2	7							
BOUND Check array against boundary										
Format	Examples	Microprocessor					Clocks			
BOUND reg,mem	BOUND AX,BETS	8086	—							
	BOUND BP,LISTG	8088	—							
	BOUND CX,DATAX	80286	13							
	BOUND BX,[DI]	80386	10							
	BOUND SI,[BX+2]	80486	7							
		Pentium–Core2	8							

BSF Bit scan forward										
		O	D	I	T	S	Z	A	P	C
Format	Examples	?	?	*	?	?	?	?	?	?
BSF reg,reg	BSF AX,BX BSF BX,SI BSF EAX,EDX BSF EBX,EAX BSF SI,DI	8086		—						
		8088		—						
		80286		—						
		80386		10 + 3n						
		80486		6–42						
		Pentium–Core2		6–42						
BSF reg,mem	BSF AX,DATAY BSF SI,LIST BSF CX,DATA3[DI] BSF EAX,[EBX] BSF EBP,[EDX+4*ECX]	8086		—						
		8088		—						
		80286		—						
		80386		10 + 3n						
		80486		7–43						
		Pentium–Core2		6–43						
BSR Bit scan reverse										
		O	D	I	T	S	Z	A	P	C
Format	Examples	?	?	*	?	?	?	?	?	?
BSR reg,reg	BSR AX,BX BSR BX,SI BSR EAX,EDX BSR EBX,EAX BSR SI,DI	8086		—						
		8088		—						
		80286		—						
		80386		10 + 3n						
		80486		6–103						
		Pentium–Core2		7–71						

BSR reg,mem	BSR AX,DATAY	8086	—
	BSR SI,LIST	8088	—
	BSR CX,DATA3[DI]	80286	—
	BSR EAX,[EBX]	80386	10 + 3n
	BSR EBP,[EDX+4*ECX]	80486	7–104
		Pentium–Core2	7–72
BSWAP Byte swap			
00001111 11001rrr			
Format	Examples	Microprocessor	Clocks
BSWAP reg32	BSWAP EAX	8086	—
	BSWAP EBX	8088	—
	BSWAP EDX	80286	—
	BSWAP ECX	80386	—
	BSWAP ESI	80486	1
		Pentium–Core2	1
BT Bit test			
00001111 10111010 oo100mmm disp data			
O D I T S Z A P C * Format Examples Microprocessor Clocks			
BT reg,imm8	BT AX,2	8086	—
	BT CX,4	8088	—
	BT BP,10H	80286	—
	BT CX,8	80386	3
	BT BX,2	80486	3
		Pentium–Core2	4

BT mem,imm8	BT DATA1,2 BT LIST,2 BT DATA2[DI],3 BT [EAX],1 BT FROG,6	8086	—
		8088	—
		80286	—
		80386	6
		80486	3
		Pentium–Core2	4
00001111 10100011 disp			
Format	Examples	Microprocessor	Clocks
BT reg,reg	BT AX,CX BT CX,DX BT BP,AX BT SI,CX BT EAX,EBX	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
		Pentium–Core2	4 or 9
BT mem,reg	BT DATA4,AX BT LIST,BX BT DATA3[DI],CX BT [EBX],DX BT [DI],DI	8086	—
		8088	—
		80286	—
		80386	12
		80486	8
		Pentium–Core2	4 or 9

BTC		Bit test and complement					
		00001111 10111010 oo111mmm disp data					
Format	Examples	Microprocessor			Clocks		
BTC reg,imm8	BTC AX,2	8086	—				
	BTC CX,4	8088	—				
	BTC BP,10H	80286	—				
	BTC CX,8	80386	6				
	BTC BX,2	80486	6				
		Pentium–Core2	7 or 8				
BTC mem,imm8	BTC DATA1,2	8086	—				
	BTC LIST,2	8088	—				
	BTC DATA2[DI],3	80286	—				
	BTC [EAX],1	80386	7 or 8				
	BTC FROG,6	80486	8				
		Pentium–Core2	8				
00001111 10111011 disp		Microprocessor			Clocks		
Format	Examples						
BTC reg,reg	BTC AX,CX	8086	—				
	BTC CX,DX	8088	—				
	BTC BP,AX	80286	—				
	BTC SI,CX	80386	6				
	BTC EAX,EBX	80486	6				
		Pentium–Core2	7 or 13				
BTC mem,reg	BTC DATA4,AX	8086	—				
	BTC LIST,BX	8088	—				
	BTC DATA3[DI],CX	80286	—				
	BTC [EBX],DX	80386	13				
	BTC [DI],DI	80486	13				
		Pentium–Core2	7 or 13				

BTR		Bit test and reset						
		0 D I T S Z A P C *						
Format	Examples	Microprocessor				Clocks		
BTR reg,imm8	BTR AX,2	8086	—					
	BTR CX,4	8088	—					
	BTR BP,10H	80286	—					
	BTR CX,8	80386	6					
	BTR BX,2	80486	6					
		Pentium–Core2	7 or 8					
BTR mem,imm8	BTR DATA1,2	8086	—					
	BTR LIST,2	8088	—					
	BTR DATA2[DI],3	80286	—					
	BTR [EAX],1	80386	8					
	BTR FROG,6	80486	8					
		Pentium–Core2	7 or 8					
00001111 10110011 disp		Microprocessor				Clocks		
Format	Examples	Microprocessor				Clocks		
BTR reg,reg	BTR AX,CX	8086	—					
	BTR CX,DX	8088	—					
	BTR BP,AX	80286	—					
	BTR SI,CX	80386	6					
	BTR EAX,EBX	80486	6					
		Pentium–Core2	7 or 13					
BTR mem,reg	BTR DATA4,AX	8086	—					
	BTR LIST,BX	8088	—					
	BTR DATA3[DI],CX	80286	—					
	BTR [EBX],DX	80386	13					
	BTR [DI],DI	80486	13					
	BTC [DI],DI	Pentium–Core2	7 or 13					

BTS		Bit test and set						
		0 D I T S Z A P C *						
Format	Examples	Microprocessor				Clocks		
BTS reg,imm8	BTS AX,2	8086	—					
	BTS CX,4	8088	—					
	BTS BP,10H	80286	—					
	BTS CX,8	80386	6					
	BTS BX,2	80486	6					
		Pentium–Core2	7 or 8					
BTS mem,imm8	BTS DATA1,2	8086	—					
	BTS LIST,2	8088	—					
	BTS DATA2[DI],3	80286	—					
	BTS [EAX],1	80386	8					
	BTS FROG,6	80486	8					
		Pentium–Core2	7 or 8					
00001111 101101011 disp		Microprocessor				Clocks		
Format	Examples	Microprocessor				Clocks		
BTS reg,reg	BTS AX,CX	8086	—					
	BTS CX,DX	8088	—					
	BTS BP,AX	80286	—					
	BTS SI,CX	80386	6					
	BTS EAX,EBX	80486	6					
		Pentium–Core2	7 or 13					
BTS mem,reg	BTS DATA4,AX	8086	—					
	BTS LIST,BX	8088	—					
	BTS DATA3[DI],CX	80286	—					
	BTS [EBX],DX	80386	13					
	BTS [DI],DI	80486	13					
		Pentium–Core2	7 or 13					

CALL Call procedure (subroutine)			
Format	Examples	Microprocessor	Clocks
CALL label (near)	CALL FOR_FUN	8086	19
	CALL HOME	8088	23
	CALL ET	80286	7
	CALL WAITING	80386	3
	CALL SOMEONE	80486	3
		Pentium–Core2	1
10011010 disp			
Format	Examples	Microprocessor	Clocks
CALL label (far)	CALL FAR PTR DATES	8086	28
	CALL WHAT	8088	36
	CALL WHERE	80286	13
	CALL FARCE	80386	17
	CALL WHOM	80486	18
		Pentium–Core2	4
11111111 oo0010mmm			
Format	Examples	Microprocessor	Clocks
CALL reg (near)	CALL AX	8086	16
	CALL BX	8088	20
	CALL CX	80286	7
	CALL DI	80386	7
	CALL SI	80486	5
		Pentium–Core2	2

CALL mem (near)	CALL ADDRESS CALL NEAR PTR [DI] CALL DATA1 CALL FROG CALL ME_NOW	8086	21 + ea
		8088	29 + ea
		80286	11
		80386	10
		80486	5
		Pentium–Core2	2
11111111 oo011mmm			
Format	Examples	Microprocessor	Clocks
CALL mem (far)	CALL FAR_LIST[SI] CALL FROM_HERE CALL TO_THERE CALL SIXX CALL OCT	8086	16
		8088	20
		80286	7
		80386	7
		80486	5
		Pentium–Core2	2
CBW	Convert byte to word (AL \Rightarrow AX)		
10011000			
Example		Microprocessor	Clocks
CBW		8086	2
		8088	2
		80286	2
		80386	3
		80486	3
		Pentium–Core2	3

CDQ	Convert doubleword to quadword (EAX \Rightarrow EDX:EAX)						
11010100 00001010	Example					Microprocessor	Clocks
CDQ		8086	—				
		8088	—				
		80286	—				
		80386	2				
		80486	2				
		Pentium–Core2	2				
CLC	Clear carry flag						
11111000		O	D	I	T	S	Z
Example		0				A	P
CLC		8086	—				
		8088	—				
		80286	—				
		80386	—				
		80486	—				
		Pentium–Core2	—				
CLD	Clear direction flag						
11111100		O	D	I	T	S	Z
Example		0				A	P
CLD		8086	—				
		8088	—				
		80286	—				
		80386	—				
		80486	—				
		Pentium–Core2	—				

CLI	Clear interrupt flag						
11111010		O	D	I	T	S	Z A P C
Example		0					
CLI		Microprocessor	Clocks				
		8086	2				
		8088	2				
		80286	3				
		80386	3				
		80486	5				
		Pentium–Core2	7				
CLTS	Clear task switched flag (CR0)						
00001111 00000110		O	D	I	T	S	Z A P C
Example		0					
CLTS		Microprocessor	Clocks				
		8086	—				
		8088	—				
		80286	2				
		80386	5				
		80486	7				
		Pentium–Core2	10				
CMC	Complement carry flag						
10011000		O	D	I	T	S	Z A P C
Example		0					*
CMC		Microprocessor	Clocks				
		8086	2				
		8088	2				
		80286	2				
		80386	2				
		80486	2				
		Pentium–Core2	2				

CMOVcondition		Conditional move									
Format	Examples			Microprocessor	Clocks						
CMOVcc reg,mem	CMOVNZ AX,FROG CMOVC EAX,[EDI] CMOVNC BX,DATA1 CMOVP EBX,WAITING CMOVNE DI,[SI]	8086	—								
		8088	—								
		80286	—								
		80386	—								
		80486	—								
		Pentium–Core2	—								
Condition Codes											
Codes	Mnemonic	Flag	Description								
0000	CMOVO	O = 1	Move if overflow								
0001	CMOVNO	O = 0	Move if no overflow								
0010	CMOVB	C = 1	Move if below								
0011	CMOVAE	C = 0	Move if above or equal								
0100	CMOVE	Z = 1	Move if equal/zero								
0101	CMOVNE	Z = 0	Move if not equal/zero								
0110	CMOVBE	C = 1 + Z = 1	Move if below or equal								
0111	CMOVA	C = 0 • Z = 0	Move if above								
1000	CMOVS	S = 1	Move if sign								
1001	CMOVNS	S = 0	Move if no sign								
1010	CMOVP	P = 1	Move if parity								
1011	CMOVNP	P = 0	Move if no parity								
1100	CMOVL	S • O	Move if less than								
1101	CMOVGE	S = 0	Move if greater than or equal								
1110	CMOVLE	Z = 1 + S • O	Move if less than or equal								
1111	CMOVG	Z = 0 + S = O	Move if greater than								
CMP		Compare									
001110dw oorrrmmm disp			O	D	I	T	S	Z	A	P	C
Format		Examples		*	*	*	*	*	*	*	*
CMP reg,reg		CMP AX,BX CMP AL,BL CMP EAX,EBX CMP CX,SI CMP ESI,EDI		8086	—	3					
				8088	—	3					
				80286	—	2					
				80386	—	2					
				80486	—	1					
				Pentium–Core2	—	1 or 2					

CMP mem,reg	CMP DATAY,AL CMP LIST,SI CMP DATA6[DI],CL CMP [EAX],CL CMP [EDX+4*ECX],EBX	8086	9 + ea
		8088	13 + ea
		80286	7
		80386	5
		80486	2
		Pentium–Core2	1 or 2
CMP reg,mem	CMP BL,DATA2 CMP SI,LIST3 CMP CL,DATA2[DI] CMP CX,[EDI] CMP ESI,[ECX+200H]	8086	9 + ea
		8088	13 + ea
		80286	6
		80386	6
		80486	2
		Pentium–Core2	1 or 2
100000sw oo111mmm disp data			
Format	Examples	Microprocessor	Clocks
CMP reg,imm	CMP CX,3 CMP DI,1AH CMP DL,34H CMP EDX,1345H CMP CX,1834H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1 or 2
CMP mem,imm	CMP DATAS,3 CMP BYTE PTR[EDI],1AH CMP DADDY,34H CMP LIST,'A' CMP TOAD,1834H	8086	10 + ea
		8088	14 + ea
		80286	6
		80386	5
		80486	2
		Pentium–Core2	1 or 2

0001111w data		Format	Examples	Microprocessor	Clocks		
CMP acc,imm	CMP AX,3		CMP AL,1AH	8086	4		
	CMP AH,34H		CMP EAX,1345H	8088	4		
	CMP AL,'Y'			80286	3		
				80386	2		
				80486	1		
				Pentium–Core2	1		
CMPS Compare strings							
1010011w							
		O D I T S Z A P C					
		* * * * *					
Format	Examples	Microprocessor		Clocks			
CMPSB CMPSW CMPSD	CMPSB		8086	32			
	CMPSW		8088	30			
	CMPSD		80286	8			
	CMPSB DATA1,DATA2		80386	10			
	REPE CMPSB		80486	8			
	REPNE CMPSW		Pentium–Core2	5			
CMPXCHG Compare and exchange							
00001111 1011000w 11rrrrrr		O D I T S Z A P C					
		* * * * *					
Format	Examples	Microprocessor		Clocks			
CMPXCHG reg,reg	CMPXCHG EAX,EBX		8086	—			
	CMPXCHG ECX,EDX		8088	—			
			80286	—			
			80386	—			
			80486	6			
			Pentium–Core2	6			

0001111w data		Format	Examples	Microprocessor	Clocks	
CMPXCHG mem,reg	CMPXCHG DATAD,EAX CMPXCHG DATA2,EDI	8086	—			
		8088	—			
		80286	—			
		80386	—			
		80486	7			
		Pentium–Core2	6			
CMPXCHG8B Compare and exchange 8 bytes						
00001111 11000111 oorrrmmm		O D I T S Z A P C				
Format	Examples	*				
		8086	—			
		8088	—			
		80286	—			
		80386	—			
		80486	—			
CPUID CPU identification code		Pentium–Core2				
00001111 10100010		10				
Example		Microprocessor		Clocks		
CPUID		8086	—	—		
		8088	—	—		
		80286	—	—		
		80386	—	—		
		80486	—	—		
		Pentium–Core2	14	—		

CWD	Convert word to doubleword (AX \Rightarrow DX:AX)			
10011000 Example		Microprocessor	Clocks	
CWD	8086	5		
	8088	5		
	80286	2		
	80386	2		
	80486	3		
	Pentium–Core2	2		
CWDE	Convert word to extended doubleword (AX \Rightarrow EAX)			
10011000 Example		Microprocessor	Clocks	
CWDE	8086	—		
	8088	—		
	80286	—		
	80386	3		
	80486	3		
	Pentium–Core2	3		
DAA	Decimal adjust AL after addition			
00100111 Example	O D I T S Z A P C ? * * * * *	Microprocessor	Clocks	
DAA	8086	4		
	8088	4		
	80286	3		
	80386	4		
	80486	2		
	Pentium–Core2	3		

DAS Decimal adjust AL after subtraction	
00101111	O D I T S Z A P C ? * * * * *
Example	Microprocessor Clocks
DAS	8086 4
	8088 4
	80286 3
	80386 4
	80486 2
	Pentium–Core2 3
DEC Decrement	
1111111w oo001mmm disp	O D I T S Z A P C * * * * *
Format	Examples
DEC reg8	DEC BL DEC BH DEC CL DEC DH DEC AH
	8086 3
	8088 3
	80286 2
	80386 2
	80486 1
	Pentium–Core2 1 or 3
DEC mem	DEC DATAY DEC LIST DEC DATA6[DI] DEC BYTE PTR [BX] DEC WORD PTR [EBX]
	8086 15 + ea
	8088 23 + ea
	80286 7
	80386 6
	80486 3
	Pentium–Core2 1 or 3

01001rrr		Format	Examples	Microprocessor	Clocks	
DEC reg16 DEC reg32	DEC CX			8086	3	
	DEC DI			8088	3	
	DEC EDX			80286	2	
	DEC ECX			80386	2	
	DEC BP			80486	1	
				Pentium–Core2	1	
DIV Divide						
1111011w oo110mmm disp						
O D I T S Z A P C						
Format Examples		Microprocessor		Clocks		
DIV reg	DIV BL			8086	162	
	DIV BH			8088	162	
	DIV ECX			80286	22	
	DIV DH			80386	38	
	DIV CX			80486	40	
				Pentium–Core2	17–41	
DIV mem	DIV DATAY			8086	168	
	DIV LIST			8088	176	
	DIV DATA6[DI]			80286	25	
	DIV BYTE PTR [BX]			80386	41	
	DIV WORD PTR [EBX]			80486	40	
				Pentium–Core2	17–41	

ENTER Create a stack frame			
Format	Examples	Microprocessor	Clocks
ENTER imm,0	ENTER 4,0 ENTER 8,0 ENTER 100,0 ENTER 200,0 ENTER 1024,0	8086 8088 80286 80386 80486	— — 11 10 14
		Pentium–Core2	11
ENTER imm,1	ENTER 4,1 ENTER 10,1	8086 8088 80286 80386 80486	— — 12 15 17
		Pentium–Core2	15
ENTER imm,imm	ENTER 3,6 ENTER 100,3	8086 8088 80286 80386 80486	— — 12 15 17
		Pentium–Core2	15 + 2n
ESC Escape (obsolete—see coprocessor)			

HLT	Halt						
11110100					Microprocessor	Clocks	
Example							
HLT		8086	2				
		8088	2				
		80286	2				
		80386	5				
		80486	4				
		Pentium–Core2	varies				
IDIV	Integer (signed) division						
1111011w oo111mmm disp					O D I T S Z A P C		
					? ? ? ? ? ?		
Format	Examples					Microprocessor	Clocks
IDIV reg	IDIV BL IDIV BH IDIV ECX IDIV DH IDIV CX	8086	184				
		8088	184				
		80286	25				
		80386	43				
		80486	43				
		Pentium–Core2	22–46				
IDIV mem	IDIV DATAY IDIV LIST IDIV DATA6[DI] IDIV BYTE PTR [BX] IDIV WORD PTR [EBX]	8086	190				
		8088	194				
		80286	28				
		80386	46				
		80486	44				
		Pentium–Core2	22–46				

IMUL Integer (signed) multiplication												
1111011w oo101mmm disp			O	D	I	T	S	Z	A	P	C	
Format	Examples		Microprocessor				Clocks				*	
IMUL reg	IMUL BL IMUL CX IMUL ECX IMUL DH IMUL AL		8086				154					
			8088				154					
			80286				21					
			80386				38					
			80486				42					
			Pentium–Core2				10–11					
IMUL mem	IMUL DATAY IMUL LIST IMUL DATA6[DI] IMUL BYTE PTR [BX] IMUL WORD PTR [EBX]		8086				160					
			8088				164					
			80286				24					
			80386				41					
			80486				42					
			Pentium–Core2				10–11					
011010s1 oorrrmmm disp data			Microprocessor				Clocks					
Format	Examples		Microprocessor				Clocks					
IMUL reg,imm	IMUL CX,16 IMUL DI,100 IMUL EDX,20		8086				—					
			8088				—					
			80286				21					
			80386				38					
			80486				42					
			Pentium–Core2				10					
IMUL reg,reg,imm	IMUL DX,AX,2 IMUL CX,DX,3 IMUL BX,AX,33		8086				—					
			8088				—					
			80286				21					
			80386				38					
			80486				42					
			Pentium–Core2				10					

IMUL reg,mem,imm	IMUL CX,DATAY,99	8086	—	
		8088	—	
		80286	24	
		80386	38	
		80486	42	
		Pentium–Core2	10	
00001111 10101111 oorrrmmm disp				
Format	Examples	Microprocessor	Clocks	
IMUL reg,reg	IMUL CX,DX IMUL DI,BX IMUL EDX,EBX	8086	—	
		8088	—	
		80286	—	
		80386	38	
		80486	42	
		Pentium–Core2	10	
IMUL reg,mem	IMUL DX,DATAY IMUL CX,LIST IMUL ECX,DATA6[DI]	8086	—	
		8088	—	
		80286	—	
		80386	41	
		80486	42	
		Pentium–Core2	10	
IN Input data from port				
1110010w port#				
Format	Examples	Microprocessor	Clocks	
IN acc,pt	IN AL,12H IN AX,12H IN AL,0FFH IN AX,0A0H IN EAX,10H	8086	10	
		8088	14	
		80286	5	
		80386	12	
		80486	14	
		Pentium–Core2	7	

1110110w		Format	Examples	Microprocessor	Clocks	
IN acc,DX			IN AL,DX	8086	8	
			IN AX,DX	8088	12	
			IN EAX,DX	80286	5	
				80386	13	
				80486	14	
				Pentium–Core2	7	
INC		Increment				
1111111w oo000mmmm disp						
Format	Examples					
INC reg8	INC BL INC BH INC AL INC AH INC DH		O	D	I	
			*	*	T	
			*	*	S	
			*	*	Z	
			*	*	A	
			*	*	P	
INC mem	INC DATA3 INC LIST INC COUNT INC BYTE PTR [DI] INC WORD PTR [ECX]		8086	3	15 + ea	
			8088	3	23 + ea	
			80286	2	7	
			80386	2	6	
			80486	1	3	
			Pentium–Core2	1 or 3		
INC reg16 INC reg32	INC CX INC DX INC BP INC ECX INC ESP		8086	3		
			8088	3		
			80286	2		
			80386	2		
			80486	1		
			Pentium–Core2	1		

INS Input string from port			
0110110w			
Format	Examples	Microprocessor	Clocks
INSB INSW INSD	INSB	8086	—
	INSW	8088	—
	INSD	80286	5
	INS DATA2	80386	15
	REP INSB	80486	17
		Pentium–Core2	9
INT Interrupt			
11001101 type			
Format	Examples	Microprocessor	Clocks
INT type	INT12H	8086	51
	INT15H	8088	71
	INT 21H	80286	23
	INT 2FH	80386	37
	INT 10H	80486	30
		Pentium–Core2	16–82
INT 3 Interrupt 3			
11001100			
Example		Microprocessor	Clocks
INT 3		8086	52
		8088	72
		80286	23
		80386	33
		80486	26
		Pentium–Core2	13–56

INTO	Interrupt on overflow				
11001110 Example		Microprocessor	Clocks		
INTO	8086	53			
	8088	73			
	80286	24			
	80386	35			
	80486	28			
	Pentium–Core2	13–56			
INVD	Invalidate data cache				
00001111 00001000 Example		Microprocessor	Clocks		
INTVD	8086	—			
	8088	—			
	80286	—			
	80386	—			
	80486	4			
	Pentium–Core2	15			
IRET/IRET D	Return from interrupt				
11001101 data	O D I T S Z A P C * * * * * * * * *				
Format	Examples	Microprocessor	Clocks		
IRET IRETD	IRET IRETD IRET 100	8086	32		
		8088	44		
		80286	17		
		80386	22		
		80486	15		
		Pentium–Core2	8–27		

Jcondition Conditional jump			
Format	Examples	Microprocessor	Clocks
Jcnd label (8-bit disp)	JA ABOVE JB BELOW JG GREATER JE EQUAL JZ ZERO	8086 8088 80286 80386 80486	16/4 16/4 7/3 7/3 3/1
		Pentium–Core2	1
00001111 1000cccc disp			
Format	Examples	Microprocessor	Clocks
Jcnd label (16-bit disp)	JNE NOT_MORE JLE LESS_OR_SO	8086 8088 80286 80386 80486	— — — 7/3 3/1
		Pentium–Core2	1
Condition Codes	Mnemonic	Flag	Description
0000	JO	O = 1	Jump if overflow
0001	JNO	O = 0	Jump if no overflow
0010	JB/NAE	C = 1	Jump if below
0011	JAE/JNB	C = 0	Jump if above or equal
0100	JE/JZ	Z = 1	Jump if equal/zero
0101	JNE/JNZ	Z = 0	Jump if not equal/zero
0110	JBE/JNA	C = 1 + Z = 1	Jump if below or equal
0111	JA/JNBE	C = 0 • Z = 0	Jump if above
1000	JS	S = 1	Jump if sign
1001	JNS	S = 0	Jump if no sign
1010	JP/JPE	P = 1	Jump if parity
1011	JNP/JPO	P = 0	Jump if no parity
1100	JL/JNGE	S • O	Jump if less than
1101	JGE/JNL	S = 0	Jump if greater than or equal
1110	JLE/JNG	Z = 1 + S • O	Jump if less than or equal
1111	JG/JNLE	Z = 0 + S = O	Jump if greater than

JCXZ/JECXZ Jump if CX (ECX) equals zero			
Format	Examples	Microprocessor	Clocks
JCXZ label JECXZ label	JCXZ ABOVE	8086	18/6
	JCXZ BELOW	8088	18/6
	JECXZ GREATER	80286	8/4
	JECXZ EQUAL	80386	9/5
	JCXZ NEXT	80486	8/5
		Pentium–Core2	6/5
JMP Jump			
Format	Examples	Microprocessor	Clocks
JMP label (short)	JMP SHORT UP	8086	15
	JMP SHORT DOWN	8088	15
	JMP SHORT OVER	80286	7
	JMP SHORT CIRCUIT	80386	7
	JMP SHORT JOKE	80486	3
		Pentium–Core2	1
Format	Examples	Microprocessor	Clocks
JMP label (near)	JMP VERS	8086	15
	JMP FROG	8088	15
	JMP UNDER	80286	7
	JMP NEAR PTR OVER	80386	7
		80486	3
		Pentium–Core2	1

11101010 disp		Format	Examples	Microprocessor	Clocks
JMP label (far)	JMP NOT_MORE		JMP UNDER JMP AGAIN JMP FAR PTR THERE	8086	15
	JMP UNDER			8088	15
	JMP AGAIN			80286	11
	JMP FAR PTR THERE			80386	12
				80486	17
				Pentium-Core2	3
11111111 oo100mmm		Format	Examples	Microprocessor	Clocks
JMP reg (near)	JMP AX		JMP EAX JMP CX JMP DX	8086	11
	JMP EAX			8088	11
	JMP CX			80286	7
	JMP DX			80386	7
				80486	3
				Pentium-Core2	2
JMP mem (near)	JMP VERS		JMP FROG JMP CS:UNDER JMP DATA1[DI+2]	8086	18 + ea
	JMP FROG			8088	18 + ea
	JMP CS:UNDER			80286	11
	JMP DATA1[DI+2]			80386	10
				80486	5
				Pentium-Core2	4
11111111 oo101mmm		Format	Examples	Microprocessor	Clocks
JMP mem (far)	JMP WAY_OFF		JMP TABLE JMP UP JMP OUT_OF_HERE	8086	24 + ea
	JMP TABLE			8088	24 + ea
	JMP UP			80286	15
	JMP OUT_OF_HERE			80386	12
				80486	13
				Pentium-Core2	4

LAHF Load AH from flags											
10011111 Example					Microprocessor	Clocks					
LAHF					8086	4					
					8088						
					80286						
					80386						
					80486						
					Pentium–Core2						
LAR Load access rights byte											
00001111 00000010 oorrrmmm disp					O D I T S Z A P C						
					*						
Format		Examples			Microprocessor	Clocks					
LAR reg,reg		LAR AX,BX LAR CX,DX LAR ECX,EDX			8086	—					
					8088						
					80286						
					80386						
					80486						
					Pentium–Core2						
LAR reg,mem		LAR CX,DATA1 LAR AX,LIST3 LAR ECX,TOAD			8086	—					
					8088						
					80286						
					80386						
					80486						
					Pentium–Core2						

LDS Load far pointer to DS and register			
11000101 oorrrmmm			
Format	Examples	Microprocessor	Clocks
LDS reg,mem	LDS DI,DATA3	8086	16 + ea
	LDS SI,LIST2	8088	24 + ea
	LDS BX,ARRAY_PTR	80286	7
	LDS CX,PNTR	80386	7
		80486	6
		Pentium–Core2	4
LEA Load effective address			
10001101 oorrrmmm disp			
Format	Examples	Microprocessor	Clocks
LEA reg,mem	LEA DI,DATA3	8086	2 + ea
	LEA SI,LIST2	8088	2 + ea
	LEA BX,ARRAY_PTR	80286	3
	LEA CX,PNTR	80386	2
		80486	2
		Pentium–Core2	1
LEAVE Leave high-level procedure			
11001001			
Example		Microprocessor	Clocks
LEAVE		8086	—
		8088	—
		80286	5
		80386	4
		80486	5
		Pentium–Core2	3

LES Load far pointer to ES and register			
11000100 oorrrmmm			
Format	Examples	Microprocessor	Clocks
LES reg,mem	LES DI,DATA3	8086	16 + ea
	LES SI,LIST2	8088	24 + ea
	LES BX,ARRAY_PTR	80286	7
	LES CX,PNTR	80386	7
		80486	6
		Pentium–Core2	4
LFS Load far pointer to FS and register			
00001111 10110100 oorrrmmm disp			
Format	Examples	Microprocessor	Clocks
LFS reg,mem	LFS DI,DATA3	8086	—
	LFS SI,LIST2	8088	—
	LFS BX,ARRAY_PTR	80286	—
	LFS CX,PNTR	80386	7
		80486	6
		Pentium–Core2	4
LGDT Load global descriptor table			
00001111 00000001 oo010mmm disp			
Format	Examples	Microprocessor	Clocks
LGDT mem64	LGDT DESCRIPT	8086	—
	LGDT TABLED	8088	—
		80286	11
		80386	11
		80486	11
		Pentium–Core2	6

LGS Load far pointer to GS and register			
Format	Examples	Microprocessor	Clocks
LGS reg,mem	LGS DI,DATA3 LGS SI,LIST2 LGS BX,ARRAY_PTR LGS CX,PNTR	8086 8088 80286 80386 80486 Pentium–Core2	— — — 7 6 4
LIDT Load interrupt descriptor table			
Format	Examples	Microprocessor	Clocks
LIDT mem64	LIDT DATA3 LIDT LIST2	8086 8088 80286 80386 80486 Pentium–Core2	— — 12 11 11 6
LLDT Load local descriptor table			
Format	Examples	Microprocessor	Clocks
LLDT reg	LLDT BX LLDT DX LLDT CX	8086 8088 80286 80386 80486 Pentium–Core2	— — 17 20 11 9

LLDT mem	LLDT DATA1 LLDT LIST3 LLDT TOAD	8086	—	
		8088	—	
		80286	19	
		80386	24	
		80486	11	
		Pentium–Core2	9	
LMSW Load machine status word (80286 only)				
00001111 00000001 oo110mmm disp				
Format	Examples	Microprocessor	Clocks	
LMSW reg	LMSW BX LMSW DX LMSW CX	8086	—	
		8088	—	
		80286	3	
		80386	10	
		80486	2	
		Pentium–Core2	8	
LMSW mem	LMSW DATA1 LMSW LIST3 LMSW TOAD	8086	—	
		8088	—	
		80286	6	
		80386	13	
		80486	3	
		Pentium–Core2	8	

LOCK Lock the bus			
11110000 Format	Examples	Microprocessor	Clocks
LOCK:inst	LOCK:XCHG AX,BX LOCK:ADD AL,3	8086	2
		8088	3
		80286	0
		80386	0
		80486	1
		Pentium–Core2	1
LODS Load string operand			
1010110w Format	Examples	Microprocessor	Clocks
LODSB LODSW LODSD	LODSB LODSW LODSD LODS DATA3	8086	12
		8088	15
		80286	5
		80386	5
		80486	5
		Pentium–Core2	2
LOOP/LOOPD Loop until CX = 0 or ECX = 0			
11100010 disp Format	Examples	Microprocessor	Clocks
LOOP label LOOPD label	LOOP NEXT LOOP BACK LOOPD LOOPS	8086	17/5
		8088	17/5
		80286	8/4
		80386	11
		80486	7/6
		Pentium–Core2	5/6

LOOPE/LOOPED Loop while equal				
Format	Examples	Microprocessor	Clocks	
LOOPE label LOOPED label LOOPZ label LOOPZD label	LOOPE AGAIN LOOPED UNTIL LOOPZ ZORRO LOOPZD WOW	8086	18/6	
		8088	18/6	
		80286	8/4	
		80386	11	
		80486	9/6	
		Pentium–Core2	7/8	
LOOPNE/LOOPNED Loop while not equal				
Format	Examples	Microprocessor	Clocks	
LOOPNE label LOOPNED label LOOPNZ label LOOPNZD label	LOOPNE FORWARD LOOPNED UPS LOOPNZ TRY AGAIN LOOPNZD WOO	8086	19/5	
		8088	19/5	
		80286	8/4	
		80386	11	
		80486	9/6	
		Pentium–Core2	7/8	
LSL Load segment limit				
00001111 00000011 oorrrmmm disp		O D I T S Z A P C	*	
Format	Examples	Microprocessor	Clocks	
LSL reg,reg	LSL AX,BX LSL CX,BX LSL EDX,EAX	8086	—	
		8088	—	
		80286	14	
		80386	25	
		80486	10	
		Pentium–Core2	8	

LSL reg,mem	LSL AX,LIMIT LSL EAX,NUM	8086	—	
		8088	—	
		80286	16	
		80386	26	
		80486	10	
		Pentium–Core2	8	
LSS Load far pointer to SS and register				
00001111 10110010 oorrrmmm disp				
Format	Examples	Microprocessor	Clocks	
LSS reg,mem	LSS DI,DATA1 LSS SP,STACK_TOP LSS CX,ARRAY	8086	—	
		8088	—	
		80286	—	
		80386	7	
		80486	6	
		Pentium–Core2	4	
LTR Load task register				
00001111 00000000 oo001mmm disp				
Format	Examples	Microprocessor	Clocks	
LTR reg	LTR AX LTR CX LTR DX	8086	—	
		8088	—	
		80286	17	
		80386	23	
		80486	20	
		Pentium–Core2	10	

LTR mem16	LTR TASK LTR NUM	8086	—	
		8088	—	
		80286	19	
		80386	27	
		80486	20	
		Pentium–Core2	10	
MOV Move data				
100010dw oorrrmmm disp				
Format	Examples	Microprocessor	Clocks	
MOV reg,reg	MOV CL,CH MOV BH,CL MOV CX,DX MOV EAX,EBP MOV ESP,ESI	8086	2	
		8088	2	
		80286	2	
		80386	2	
		80486	1	
		Pentium–Core2	1	
MOV mem,reg	MOV DATA7,DL MOV NUMB,CX MOV TEMP,EBX MOV [ECX],BL MOV [DI],DH	8086	9 + ea	
		8088	13 + ea	
		80286	3	
		80386	2	
		80486	1	
		Pentium–Core2	1	
MOV reg,mem	MOV DL,DATA8 MOV DX,NUMB MOV EBX,TEMP+3 MOV CH,TEMP[EDI] MOV CL,DATA2	8086	10 + ea	
		8088	12 + ea	
		80286	5	
		80386	4	
		80486	1	
		Pentium–Core2	1	

1100011w oo000mmm disp data		Microprocessor	Clocks
Format	Examples		
MOV mem,imm	MOV DATAF,23H MOV LIST,12H MOV BYTE PTR [DI],2 MOV NUMB,234H MOV DWORD PTR[ECX],1	8086	10 + ea
		8088	14 + ea
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1
1011wrrr data		Microprocessor	Clocks
Format	Examples		
MOV reg,imm	MOV BX,22H MOV CX,12H MOV CL,2 MOV ECX,123456H MOV DI,100	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1
101000dw disp		Microprocessor	Clocks
Format	Examples		
MOV mem,acc	MOV DATAF,AL MOV LIST,AX MOV NUMB,EAX	8086	10
		8088	14
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1
MOV acc,mem	MOV AL,DATAE MOV AX,LIST MOV EAX,LUTE	8086	10
		8088	14
		80286	5
		80386	4
		80486	1
		Pentium–Core2	1

100011d0 oosssmmm disp		Microprocessor	Clocks
Format	Examples		
MOV seg,reg	MOV SS,AX MOV DS,DX MOV ES,CX MOV FS,BX MOV GS,AX	8086 8088 80286 80386 80486 Pentium–Core2	2 2 2 2 1 1
MOV seg,mem	MOV SS,STACK_TOP MOV DS,DATAS MOV ES,TEMP1	8086 8088 80286 80386 80486 Pentium–Core2	8 + ea 12 + ea 2 2 1 2 or 3
MOV reg,seg	MOV BX,DS MOV CX,FS MOV CX,ES	8086 8088 80286 80386 80486 Pentium–Core2	2 2 2 2 1 1
MOV mem,seg	MOV DATA2,CS MOV TEMP,DS MOV NUMB1,SS MOV TEMP2,GS	8086 8088 80286 80386 80486 Pentium–Core2	9 + ea 13 + ea 3 2 1 1

00001111 001000d0 11rrrrmmmm		Format	Examples	Microprocessor	Clocks
MOV reg,cr			MOV EBX,CR0	8086	—
			MOV ECX,CR2	8088	—
			MOV EBX,CR3	80286	—
				80386	6
				80486	4
				Pentium–Core2	4
MOV cr,reg			MOV CR0,EAX	8086	—
			MOV CR1,EBX	8088	—
			MOV CR3,EDX	80286	—
				80386	10
				80486	4
				Pentium–Core2	12–46
00001111 001000d1 11rrrrmmmm		Format	Examples	Microprocessor	Clocks
MOV reg,dr			MOV EBX,DR6	8086	—
			MOV ECX,DR7	8088	—
			MOV EBX,DR1	80286	—
				80386	22
				80486	10
				Pentium–Core2	11
MOV dr,reg			MOV DR0,EAX	8086	—
			MOV DR1,EBX	8088	—
			MOV DR3,EDX	80286	—
				80386	22
				80486	11
				Pentium–Core2	11

00001111 001001d0 11rrrrmmmm		Microprocessor	Clocks
Format	Examples		
MOV reg,tr	MOV EBX,TR6 MOV ECX,TR7	8086	—
		8088	—
		80286	—
		80386	12
		80486	4
		Pentium–Core2	11
MOV tr,reg	MOV TR6,EAX MOV TR7,EBX	8086	—
		8088	—
		80286	—
		80386	12
		80486	6
		Pentium–Core2	11
MOVS Move string data			
1010010w		Microprocessor	Clocks
Format	Examples		
MOVSB MOVSW MOVSD	MOVSB MOVSW MOVSD MOVS DATA1,DATA2	8086	18
		8088	26
		80286	5
		80386	7
		80486	7
		Pentium–Core2	4

MOVSX Move with sign extend			
Format	Examples	Microprocessor	Clocks
MOVSX reg,reg	MOVSX BX,AL MOVSX EAX,DX	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
		Pentium–Core2	3
MOVSX reg,mem	MOVSX AX,DATA34 MOVSX EAX,NUMB	8086	—
		8088	—
		80286	—
		80386	6
		80486	3
		Pentium–Core2	3
MOVZX Move with zero extend			
Format	Examples	Microprocessor	Clocks
MOVZX reg,reg	MOVZX BX,AL MOVZX EAX,DX	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
		Pentium–Core2	3
MOVZX reg,mem	MOVZX AX,DATA34 MOVZX EAX,NUMB	8086	—
		8088	—
		80286	—
		80386	6
		80486	3
		Pentium–Core2	3

MUL		Multiply										
1111011w oo100mmm disp		O	D	I	T	S	Z	A	P	C		
Format	Examples	Microprocessor					Clocks					
MUL reg	MUL BL MUL CX MUL EDX	8086				*	?	?	?	*		
		8088				*	?	?	?	*		
		80286				*	?	?	?	*		
		80386				*	?	?	?	*		
		80486				*	?	?	?	*		
		Pentium–Core2		10 or 11								
MUL mem	MUL DATA9 MUL WORD PTR [ESI]	8086				*	?	?	?	*		
		8088				*	?	?	?	*		
		80286				*	?	?	?	*		
		80386				*	?	?	?	*		
		80486				*	?	?	?	*		
		Pentium–Core2		11								
NEG		Negate										
1111011w oo011mmm disp		O	D	I	T	S	Z	A	P	C		
Format	Examples	Microprocessor					Clocks					
NEG reg	NEG BL NEG CX NEG EDI	8086				*	?	?	?	*		
		8088				*	?	?	?	*		
		80286				*	?	?	?	*		
		80386				*	?	?	?	*		
		80486				*	?	?	?	*		
		Pentium–Core2		1 or 3								

NEG mem	NEG DATA9 NEG WORD PTR [ESI]	8086	16 + ea	
		8088	24 + ea	
		80286	7	
		80386	6	
		80486	3	
		Pentium–Core2	1 or 3	
NOP No operation				
10010000 Example		Microprocessor	Clocks	
NOP		8086	3	
		8088	3	
		80286	3	
		80386	3	
		80486	3	
		Pentium–Core2	1	
NOT One's complement				
1111011w oo010mmm disp Format Examples		Microprocessor	Clocks	
NOT reg	NOT BL NOT CX NOT EDI	8086	3	
		8088	3	
		80286	2	
		80386	2	
		80486	1	
		Pentium–Core2	1 or 3	
NOT mem	NOT DATA9 NOT WORD PTR [ESI]	8086	16 + ea	
		8088	24 + ea	
		80286	7	
		80386	6	
		80486	3	
		Pentium–Core2	1 or 3	

OR mem,imm	OR DATAS,3 OR BYTE PTR[EDI],1AH OR DADDY,34H OR LIST,'A' OR TOAD,1834H	8086	17 + ea
		8088	25 + ea
		80286	7
		80386	7
		80486	3
		Pentium–Core2	1 or 3
0000110w data			
Format	Examples	Microprocessor	Clocks
OR acc,imm	OR AX,3 OR AL,1AH OR AH,34H OR EAX,1345H OR AL,'Y'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1
OUT	Output data to port		
1110011w port#			
Format	Examples	Microprocessor	Clocks
OUT pt,acc	OUT 12H,AL OUT 12H,AX OUT 0FFH,AL OUT 0A0H,AX OUT 10H,EAX	8086	10
		8088	14
		80286	3
		80386	10
		80486	10
		Pentium–Core2	12–26
1110111w			
Format	Examples	Microprocessor	Clocks
OUT DX,acc	OUT DX,AL OUT DX,AX OUT DX,EAX	8086	8
		8088	12
		80286	3
		80386	11
		80486	10
		Pentium–Core2	12–26

OUTS Output string to port				
0110111w Format		Examples	Microprocessor	Clocks
OUTSB OUTSW OUTSD		OUTSB	8086	—
		OUTSW	8088	—
		OUTSD	80286	5
		OUTS DATA2	80386	14
		REP OUTSB	80486	10
			Pentium–Core2	13–27
POP Pop data from stack				
01011rrr Format		Examples	Microprocessor	Clocks
POP reg		POP CX	8086	8
		POP AX	8088	12
		POP EDI	80286	5
			80386	4
			80486	1
			Pentium–Core2	1
10001111 oo000mmm disp Format				
Examples		Microprocessor	Clocks	
POP mem		POP DATA1	8086	17 + ea
		POP LISTS	8088	25 + ea
		POP NUMBS	80286	5
			80386	5
			80486	4
			Pentium–Core2	3

00sss111		Format	Examples	Microprocessor	Clocks
POP seg	POP DS POP ES POP SS			8086 8088 80286 80386 80486 Pentium–Core2	8 12 5 7 3 3
00001111 10sss001					
Format		Examples		Microprocessor	Clocks
POP seg	POP FS POP GS			8086 8088 80286 80386 80486 Pentium–Core2	— — — 7 3 3
POPA/POPAD Pop all registers from stack					
01100001		Example		Microprocessor	Clocks
POPA POPAD				8086 8088 80286 80386 80486 Pentium–Core2	— — 19 24 9 5

POPF/POPDF		Pop flags from stack									
10010000		O	D	I	T	S	Z	A	P	C	
Example		Microprocessor									
POPF POPDF	8086	8									
	8088	12									
	80286	5									
	80386	5									
	80486	6									
	Pentium–Core2	4 or 6									
PUSH		Push data onto stack									
01010rrr		Format									
		Examples									
PUSH reg	PUSH CX	8086	11								
	PUSH AX	8088	15								
	PUSH EDI	80286	3								
		80386	2								
		80486	1								
		Pentium–Core2	1								
11111111 oo110mmm disp											
Format		Examples									
		Micropocessor									
PUSH mem		PUSH DATA1	8086	16 + ea							
		PUSH LISTS	8088	24 + ea							
		PUSH NUMBS	80286	5							
			80386	5							
			80486	4							
			Pentium–Core2	1 or 2							

00ss110		Format	Examples	Microprocessor	Clocks
PUSH seg	PUSH ES PUSH CS PUSH DS			8086	10
				8088	14
				80286	3
				80386	2
				80486	3
				Pentium–Core2	1
00001111 10sss000					
Format	Examples	Microprocessor		Clocks	
PUSH seg	PUSH FS PUSH GS	8086	—	8088	—
		80286	—	80386	2
		80486	3	Pentium–Core2	1
011010s0 data					
Format	Examples	Microprocessor		Clocks	
PUSH imm	PUSH 2000H PUSH 53220 PUSHW 10H PUSH ',' PUSHD 100000H	8086	—	8088	—
		80286	3	80386	2
		80486	1	Pentium–Core2	1

PUSHA/PUSHAD Push all registers onto stack			
01100000 Example		Microprocessor	Clocks
PUSHA PUSHAD	8086	—	
	8088	—	
	80286	17	
	80386	18	
	80486	11	
	Pentium–Core2	5	
PUSHF/PUSHFD Push flags onto stack			
10011100 Example		Microprocessor	Clocks
PUSHF PUSHFD	8086	10	
	8088	14	
	80286	3	
	80386	4	
	80486	3	
	Pentium–Core2	3 or 4	
RCL/RCR/ROL/ROR Rotate			
1101000w ooTTTmmm disp	O D I T S Z A P C	*	*
Format	Examples	Microprocessor	Clocks
ROL reg,1 ROR reg,1	ROL CL,1	8086	2
	ROL DX,1	8088	2
	ROR CH,1	80286	2
	ROR SI,1	80386	3
		80486	3
		Pentium–Core2	1 or 3

RCL reg,1 RCR reg,1	RCL CL,1 RCL SI,1 RCR AH,1 RCR EBX,1	8086	2
		8088	2
		80286	2
		80386	9
		80486	3
		Pentium–Core2	1 or 3
ROL mem,1 ROR mem,1	ROL DATA1,1 ROL LIST,1 ROR DATA2[DI],1 ROR BYTE PTR [EAX],1	8086	15 + ea
		8088	23 + ea
		80286	7
		80386	7
		80486	4
		Pentium–Core2	1 or 3
RCL mem,1 RCR mem,1	RCL DATA1,1 RCL LIST,1 RCR DATA2[SI],1 RCR WORD PTR [ESI],1	8086	15 + ea
		8088	23 + ea
		80286	7
		80386	10
		80486	4
		Pentium–Core2	1 or 3
1101001w ooTTTmmmm disp			
Format	Examples	Microprocessor	Clocks
ROL reg,CL ROR reg,CL	ROL CH,CL ROL DX,CL ROR AL,CL ROR ESI,CL	8086	8 + 4n
		8088	8 + 4n
		80286	5 + n
		80386	3
		80486	3
		Pentium–Core2	4

RCL reg,CL RCR reg,CL	RCL CH,CL RCL SI,CL RCR AH,CL RCR EBX,CL	8086	8 + 4n
		8088	8 + 4n
		80286	5 + n
		80386	9
		80486	3
		Pentium–Core2	7–27
ROL mem,CL ROR mem,CL	ROL DATA1,CL ROL LIST,CL ROR DATA2[DI],CL ROR BYTE PTR [EAX],CL	8086	20 + 4n
		8088	28 + 4n
		80286	8 + n
		80386	7
		80486	4
		Pentium–Core2	4
RCL mem,CL RCR mem,CL	RCL DATA1,CL RCL LIST,CL RCR DATA2[SI],CL RCR WORD PTR [ESI],CL	8086	20 + 4n
		8088	28 + 4n
		80286	8 + n
		80386	10
		80486	9
		Pentium–Core2	9–26
1100000w ooTTTmmmm disp data			
Format	Examples	Microprocessor	Clocks
ROL reg,imm ROR reg,imm	ROL CH,4 ROL DX,5 ROR AL,2 ROR ESI,14	8086	—
		8088	—
		80286	5 + n
		80386	3
		80486	2
		Pentium–Core2	1 or 3

RCL reg,imm RCR reg,imm	RCL CL,2 RCL SI,12 RCR AH,5 RCR EBX,18	8086	—	
		8088	—	
		80286	5 + n	
		80386	9	
		80486	8	
		Pentium–Core2	8–27	
ROL mem,imm ROR mem,imm	ROL DATA1,4 ROL LIST,3 ROR DATA2[DI],7 ROR BYTE PTR [EAX],11	8086	—	
		8088	—	
		80286	8 + n	
		80386	7	
		80486	4	
		Pentium–Core2	1 or 3	
RCL mem,imm RCR mem,imm	RCL DATA1,5 RCL LIST,3 RCR DATA2[SI],9 RCR WORD PTR [ESI],8	8086	—	
		8088	—	
		80286	8 + n	
		80386	10	
		80486	9	
		Pentium–Core2	8–27	
RDMSR Read model specific register				
00001111 00110010 Example		Microprocessor	Clocks	
RDMSR		8086	—	
		8088	—	
		80286	—	
		80386	—	
		80486	—	
		Pentium–Core2	20–24	

REP Repeat prefix			
11110011 1010010w			
Format	Examples	Microprocessor	Clocks
REP MOVS	REP MOVS REP MOVSW REP MOVSD REP MOVS DATA1,DATA2	8086	9 + 17n
		8088	9 + 25n
		80286	5 + 4n
		80386	8 + 4n
		80486	12 + 3n
		Pentium–Core2	13 + n
11110011 1010101w			
Format	Examples	Microprocessor	Clocks
REP STOS	REP STOSB REP STOSW REP STOSD REP STOS ARRAY	8086	9 + 10n
		8088	9 + 14n
		80286	4 + 3n
		80386	5 + 5n
		80486	7 + 4n
		Pentium–Core2	9 + n
11110011 0110110w			
Format	Examples	Microprocessor	Clocks
REP INS	REP INSB REP INSW REP INSD REP INS ARRAY	8086	—
		8088	—
		80286	5 + 4n
		80386	12 + 5n
		80486	17 + 5n
		Pentium–Core2	25 + 3n

11110011 0110111w		Format	Examples	Microprocessor	Clocks	
REP OUTS	REP OUTSB REP OUTSW REP OUTSD REP OUTS ARRAY			8086	—	
				8088	—	
				80286	5 + 4n	
				80386	12 + 5n	
				80486	17 + 5n	
				Pentium–Core2	25 + 4n	
REPE/REPNE		Repeat conditional				
11110011 1010011w		Format	Examples	Microprocessor	Clocks	
REPE CMPS	REPE CMPSB REPE CMPSW REPE CMPSD REPE CMPS DATA1,DATA2			8086	9 + 22n	
				8088	9 + 30n	
				80286	5 + 9n	
				80386	5 + 9n	
				80486	7 + 7n	
				Pentium–Core2	9 + 4n	
11110011 1010111w		Format	Examples	Microprocessor	Clocks	
REPE SCAS	REPE SCASB REPE SCASW REPE SCASD REPE SCAS ARRAY			8086	9 + 15n	
				8088	9 + 19n	
				80286	5 + 8n	
				80386	5 + 8n	
				80486	7 + 5n	
				Pentium–Core2	9 + 4n	

11110010 1010011w		Format	Examples	Microprocessor	Clocks	
REPNE CMPS	REPNE CMPSB REPNE CMPSW REPNE CMPSD REPNE CMPS ARRAY,LIST	8086	9 + 22n			
		8088	9 + 30n			
		80286	5 + 9n			
		80386	5 + 9n			
		80486	7 + 7n			
		Pentium–Core2	8 + 4n			
11110010 101011w		Format	Examples	Microprocessor	Clocks	
REPNE SCAS	REPNE SCASB REPNE SCASW REPNE SCASD REPNE SCAS ARRAY	8086	9 + 15n			
		8088	9 + 19N			
		80286	5 + 8n			
		80386	5 + 8n			
		80486	7 + 5n			
		Pentium–Core2	9 + 4n			
RET		Return from procedure				
11000011		Example	Microprocessor	Clocks		
RET (near)		8086	16			
		8088	20			
		80286	11			
		80386	10			
		80486	5			
		Pentium–Core2	2			

11000010 data		Format	Examples	Microprocessor	Clocks
RET imm (near)	RET 4 RET 100H			8086	20
				8088	24
				80286	11
				80386	10
				80486	5
				Pentium–Core2	3
11001011		Example		Microprocessor	Clocks
RET (far)				8086	26
				8088	34
				80286	15
				80386	18
				80486	13
				Pentium–Core2	4–23
11001010 data		Format	Examples	Microprocessor	Clocks
RET imm (far)	RET 4 RET 100H			8086	25
				8088	33
				80286	11
				80386	10
				80486	5
				Pentium–Core2	4–23

RSM Resume from system management mode			
00001111 10101010	O D I T S Z A P C * * * * * * * * *		
Example	Microprocessor Clocks		
RSM	8086 —		
	8088 —		
	80286 —		
	80386 —		
	80486 —		
	Pentium–Core2 83		
SAHF Store AH into flags			
10011110	O D I T S Z A P C * * * * * * * * *		
Example	Microprocessor Clocks		
SAHF	8086 4		
	8088 4		
	80286 2		
	80386 3		
	80486 2		
	Pentium–Core2 2		
SAL/SAR/SHL/SHR Shift			
1101000w ooTTTmmm disp	O D I T S Z A P C * * * ? * *		
TTT = 100 = SHL/SAL , TTT = 101 = SHR, and TTT = 111 = SAR			
Format	Examples	Microprocessor	Clocks
SAL reg,1	SAL CL,1	8086	2
SHL reg,1	SHL DX,1	8088	2
SHR reg,1	SAR CH,1	80286	2
SAR reg,1	SHR SI,1	80386	3
		80486	3
		Pentium–Core2	1 or 3

SAL mem,1 SHL mem,1 SHR mem,1 SAR mem,1	SAL DATA1,1 SHL BYTE PTR [DI],1 SAR NUMB,1 SHR WORD PTR[EDI],1	8086 8088 80286 80386 80486 Pentium–Core2	15 + ea 23 + ea 7 7 4 1 or 3
1101001w ooTTTmmmm disp			
Format	Examples	Microprocessor	Clocks
SAL reg,CL SHL reg,CL SAR reg,CL SHR reg,CL	SAL CH,CL SHL DX,CL SAR AL,CL SHR ESI,CL	8086 8088 80286 80386 80486 Pentium–Core2	8 + 4n 8 + 4n 5 + n 3 3 4
SAL mem,CL SHL mem,CL SAR mem,CL SHR mem,CL	SAL DATAU,CL SHL BYTE PTR [ESI],CL SAR NUMB,CL SHR TEMP,CL	8086 8088 80286 80386 80486 Pentium–Core2	20 + 4n 28 + 4n 8 + n 7 4 4
1100000w ooTTTmmmm disp data			
Format	Examples	Microprocessor	Clocks
SAL reg,imm SHL reg,imm SAR reg,imm SHR reg,imm	SAL CH,4 SHL DX,10 SAR AL,2 SHR ESI,23	8086 8088 80286 80386 80486 Pentium–Core2	— — 5 + n 3 2 1 or 3

SAL mem,imm SHL mem,imm SAR mem,imm SHR mem,imm	SAL DATAU,3 SHL BYTE PTR [ESI],15 SAR NUMB,3 SHR TEMP,5	8086 8088 80286 80386 80486 Pentium–Core2	— — 8 + n 7 4 1 or 3
SBB Subtract with borrow			
Format	Examples	Microprocessor	Clocks
SBB reg,reg	SBB CL,DL SBB AX,DX SBB CH,CL SBB EAX,EBX SBB ESI,EDI	8086 8088 80286 80386 80486 Pentium–Core2	3 3 2 2 1 1 or 2
SBB mem,reg	SBB DATAJ,CL SBB BYTES,CX SBB NUMBS,ECX SBB [EAX],CX	8086 8088 80286 80386 80486 Pentium–Core2	16 + ea 24 + ea 7 6 3 1 or 3
SBB reg,mem	SBB CL,DATAL SBB CX,BYTES SBB ECX,NUMBS SBB DX,[EBX+EDI]	8086 8088 80286 80386 80486 Pentium–Core2	9 + ea 13 + ea 7 7 2 1 or 2

100000sw oo011mmm disp data		Format	Examples	Microprocessor	Clocks	
SBB reg,imm	SBB CX,3 SBB DI,1AH SBB DL,34H SBB EDX,1345H SBB CX,1834H	8086	4			
		8088	4			
		80286	3			
		80386	2			
		80486	1			
		Pentium–Core2	1 or 3			
		8086	17 + ea			
SBB mem,imm	SBB DATAS,3 SBB BYTE PTR[EDI],1AH SBB DADDY,34H SBB LIST,'A' SBB TOAD,1834H	8088	25 + ea			
		80286	7			
		80386	7			
		80486	3			
		Pentium–Core2	1 or 3			
0001110w data						
Format	Examples	Microprocessor	Clocks			
SBB acc,imm	SBB AX,3 SBB AL,1AH SBB AH,34H SBB EAX,1345H SBB AL,'Y'	8086	4			
		8088	4			
		80286	3			
		80386	2			
		80486	1			
		Pentium–Core2	1			
SCAS Scan string						
1010111w		O D I T S Z A P C	*	*	*	
Format	Examples	Microprocessor	Clocks			
SCASB SCASW SCASD	SCASB SCASW SCASD SCAS DATAF REP SCASB	8086	15			
		8088	19			
		80286	7			
		80386	7			
		80486	6			
		Pentium–Core2	4			

SETcondition		Conditional set	
Format	Examples	Microprocessor	Clocks
SETcnd reg8	SETA BL SETB CH SETG DL SETE BH SETZ AL	8086 8088 80286 80386 80486	— — — 4 3
		Pentium–Core2	1 or 2
SETcnd mem8	SETE DATAK SETAE LESS_OR_SO	8086 8088 80286 80386 80486	— — — 5 3
		Pentium–Core2	1 or 2
Condition			
Codes	Mnemonic	Flag	Description
0000	SETO	O = 1	Set if overflow
0001	SETNO	O = 0	Set if no overflow
0010	SETB/SETAE	C = 1	Set if below
0011	SETAE/SETNB	C = 0	Set if above or equal
0100	SETE/SETZ	Z = 1	Set if equal/zero
0101	SETNE/SETNZ	Z = 0	Set if not equal/zero
0110	SETBE/SETNA	C = 1 + Z = 1	Set if below or equal
0111	SETA/SETNBE	C = 0 • Z = 0	Set if above
1000	SETS	S = 1	Set if sign
1001	SETNS	S = 0	Set if no sign
1010	SETP/SETPE	P = 1	Set if parity
1011	SETNP/SETPO	P = 0	Set if no parity
1100	SETL/SETNGE	S • O	Set if less than
1101	SETGE/SETNL	S = 0	Set if greater than or equal
1110	SETLE/SETNG	Z = 1 + S • O	Set if less than or equal
1111	SETG/SETNLE	Z = 0 + S = O	Set if greater than

SGDT/SIDT/SLDT Store descriptor table registers			
Format	Examples	Microprocessor	Clocks
SGDT mem	SGDT MEMORY SGDT GLOBAL	8086	—
		8088	—
		80286	11
		80386	9
		80486	10
		Pentium–Core2	4
00001111 00000001 oo001mmm disp			
Format	Examples	Microprocessor	Clocks
SIDT mem	SIDT DATAS SIDT INTERRUPT	8086	—
		8088	—
		80286	12
		80386	9
		80486	10
		Pentium–Core2	4
00001111 00000000 oo000mmm disp			
Format	Examples	Microprocessor	Clocks
SLDT reg	SLDT CX SLDT DX	8086	—
		8088	—
		80286	2
		80386	2
		80486	2
		Pentium–Core2	2
SLDT mem	SLDT NUMBS SLDT LOCALS	8086	—
		8088	—
		80286	3
		80386	2
		80486	3
		Pentium–Core2	2

SHLD/SHRD Double precision shift						
		00001111 10100100 oorrrmmm disp data	O D I T	S Z A P C		
Format	Examples	Microprocessor			Clocks	
SHLD reg,reg,imm	SHLD AX,CX,10 SHLD DX,BX,8 SHLD CX,DX,2	8086	—	—	—	
		8088	—	—	—	
		80286	—	—	—	
		80386	3	—	—	
		80486	2	—	—	
		Pentium–Core2	4	—	—	
SHLD mem,reg,imm	SHLD DATAQ,CX,8	8086	—	—	—	
		8088	—	—	—	
		80286	—	—	—	
		80386	7	—	—	
		80486	3	—	—	
		Pentium–Core2	4	—	—	
		00001111 10101100 oorrrmmm disp data	Microprocessor			
Format	Examples	Microprocessor			Clocks	
SHRD reg,reg,imm	SHRD CX,DX,2	8086	—	—	—	
		8088	—	—	—	
		80286	—	—	—	
		80386	3	—	—	
		80486	2	—	—	
		Pentium–Core2	4	—	—	
SHRD mem,reg,imm	SHRD DATAZ,DX,4	8086	—	—	—	
		8088	—	—	—	
		80286	—	—	—	
		80386	7	—	—	
		80486	2	—	—	
		Pentium–Core2	4	—	—	

00001111 10100101 oorrrmmm disp		Format	Examples	Microprocessor	Clocks
SHLD reg,reg,CL	SHLD BX,DX,CL	8086	—		
		8088	—		
		80286	—		
		80386	3		
		80486	3		
		Pentium–Core2	4 or 5		
SHLD mem,reg,CL	SHLD DATAZ,DX,CL	8086	—		
		8088	—		
		80286	—		
		80386	7		
		80486	3		
		Pentium–Core2	4 or 5		
00001111 10101101 oorrrmmm disp		Format	Examples	Microprocessor	Clocks
SHRD reg,reg,CL	SHRD AX,DX,CL	8086	—		
		8088	—		
		80286	—		
		80386	3		
		80486	3		
		Pentium–Core2	4 or 5		
SHRD mem,reg,CL	SHRD DATAZ,DX,CL	8086	—		
		8088	—		
		80286	—		
		80386	7		
		80486	3		
		Pentium–Core2	4 or 5		

SMSW Store machine status word (80286)				
Format	Examples	Microprocessor	Clocks	
SMSW reg	SMSW AX SMSW DX SMSW BP	8086	—	
		8088	—	
		80286	2	
		80386	10	
		80486	2	
		Pentium–Core2	4	
SMSW mem	SMSW DATAQ	8086	—	
		8088	—	
		80286	3	
		80386	3	
		80486	3	
		Pentium–Core2	4	
STC Set carry flag				
11111001		O D I T S Z A P C		
Example		1		
		Microprocessor	Clocks	
STC		8086	2	
		8088	2	
		80286	2	
		80386	2	
		80486	2	
		Pentium–Core2	2	

STD	Set direction flag						
11111101		O	D	I	T	S	Z A P C
Example		1				Microprocessor	Clocks
STD		8086				2	
		8088				2	
		80286				2	
		80386				2	
		80486				2	
		Pentium–Core2				2	
STI	Set interrupt flag						
11111011		O	D	I	T	S	Z A P C
Example		1				Microprocessor	Clocks
STI		8086				2	
		8088				2	
		80286				2	
		80386				3	
		80486				5	
		Pentium–Core2				7	
STOS	Store string data						
1010101w							
Format	Examples			Microprocessor		Clocks	
STOSB	STOSB			8086		11	
STOSW	STOSW			8088		15	
STOSD	STOSD			80286		3	
	STOS DATA_LIST			80386		40	
	REP STOSB			80486		5	
				Pentium–Core2		3	

STR Store task register			
Format	Examples	Microprocessor	Clocks
STR reg	STR AX	8086	—
	STR DX	8088	—
	STR BP	80286	2
		80386	2
		80486	2
		Pentium–Core2	2
STR mem	STR DATA3	8086	—
		8088	—
		80286	2
		80386	2
		80486	2
		Pentium–Core2	2
SUB Subtract			
Format		O D I T S Z A P C * * * * * *	
SUB reg,reg	SUB CL,DL	8086	3
	SUB AX,DX	8088	3
	SUB CH,CL	80286	2
	SUB EAX,EBX	80386	2
	SUB ESI,EDI	80486	1
		Pentium–Core2	1 or 2

SUB mem,reg	SUB DATAJ,CL SUB BYTES,CX SUB NUMBS,ECX SUB [EAX],CX	8086 8088 80286 80386 80486 Pentium–Core2	16 + ea 24 + ea 7 6 3 1 or 3
SUB reg,mem	SUB CL,DATAL SUB CX,BYTES SUB ECX,NUMBS SUB DX,[EBX+EDI]	8086 8088 80286 80386 80486 Pentium–Core2	9 + ea 13 + ea 7 7 2 1 or 2
100000sw oo101mmm disp data			
Format	Examples	Microprocessor	Clocks
SUB reg,imm	SUB CX,3 SUB DI,1AH SUB DL,34H SUB EDX,1345H SUB CX,1834H	8086 8088 80286 80386 80486 Pentium–Core2	4 4 3 2 1 1 or 3
SUB mem,imm	SUB DATAS,3 SUB BYTE PTR[EDI],1AH SUB DADDY,34H SUB LIST;A' SUB TOAD,1834H	8086 8088 80286 80386 80486 Pentium–Core2	17 + ea 25 + ea 7 7 3 1 or 3

0010110w data										
Format	Examples	Microprocessor			Clocks					
SUB acc,imm	SUB AL,3 SUB AX,1AH SUB EAX,34H	8086	4							
		8088	4							
		80286	3							
		80386	2							
		80486	1							
		Pentium–Core2	1							
TEST Test operands (logical compare)										
1000001w oorrrmmm disp										
		O	D	I	T	S	Z	A	P	C
		0		*	*	?	*	0		
Format	Examples	Microprocessor			Clocks					
TEST reg,reg	TEST CL,DL TEST BX,DX TEST DH,CL TEST EBP,EBX TEST EAX,EDI	8086	5							
		8088	5							
		80286	2							
		80386	2							
		80486	1							
		Pentium–Core2	1 or 2							
TEST mem,reg reg,mem	TEST DATAJ,CL TEST BYTES,CX TEST NUMBS,ECX TEST [EAX],CX TEST CL,POPS	8086	9 + ea							
		8088	13 + ea							
		80286	6							
		80386	5							
		80486	2							
		Pentium–Core2	1 or 2							

1111011sw oo000mmmm disp data		Microprocessor	Clocks
Format	Examples		
TEST reg,imm	TEST BX,3 TEST DI,1AH TEST DH,44H TEST EDX,1AB345H TEST SI,1834H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1 or 2
TEST mem,imm	TEST DATAS,3 TEST BYTE PTR[EDI],1AH TEST DADDY,34H TEST LIST,'A' TEST TOAD,1834H	8086	11 + ea
		8088	11 + ea
		80286	6
		80386	5
		80486	2
		Pentium–Core2	1 or 2
1010100w data		Microprocessor	Clocks
Format	Examples		
TEST acc,imm	TEST AL,3 TEST AX,1AH TEST EAX,34H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1

VERR/VERW Verify read/write		O	D	I	T	S	Z	A	P	C
*										
Format	Examples	Microprocessor					Clocks			
VERR reg	VERR CX VERR DX VERR DI	8086					—			
		8088					—			
		80286					14			
		80386					10			
		80486					11			
		Pentium–Core2					7			
VERR mem	VERR DATAJ VERR TESTB	8086					—			
		8088					—			
		80286					16			
		80386					11			
		80486					11			
		Pentium–Core2					7			
00001111 00000000 oo101mmm disp		Microprocessor					Clocks			
Format	Examples	Microprocessor					Clocks			
VERW reg	VERW CX VERW DX VERW DI	8086					—			
		8088					—			
		80286					14			
		80386					15			
		80486					11			
		Pentium–Core2					7			
VERW mem	VERW DATAJ VERW TESTB	8086					—			
		8088					—			
		80286					16			
		80386					16			
		80486					11			
		Pentium–Core2					7			

WAIT	Wait for coprocessor		
10011011 Example		Microprocessor	Clocks
WAIT	8086	4	
FWAIT	8088	4	
	80286	3	
	80386	6	
	80486	6	
	Pentium–Core2	1	
WBINVD	Write-back cache invalidate data cache		
00001111 00001001 Example		Microprocessor	Clocks
WBINVD	8086	—	
	8088	—	
	80286	—	
	80386	—	
	80486	5	
	Pentium–Core2	2000+	
WRMSR	Write to model specific register		
00001111 00110000 Example		Microprocessor	Clocks
WRMSR	8086	—	
	8088	—	
	80286	—	
	80386	—	
	80486	—	
	Pentium–Core2	30–45	

XADD Exchange and add		O	D	I	T	S	Z	A	P	C
*	*	*	*	*	*	*	*	*	*	*
Format	Examples	Microprocessor					Clocks			
XADD reg,reg	XADD EBX,ECX	8086	—							
	XADD EDX,EAX	8088	—							
	XADD EDI,EBP	80286	—							
		80386	—							
		80486	3							
		Pentium–Core2	3 or 4							
00001111 1100000w oorrrmmm disp		Microprocessor					Clocks			
Format	Examples	Microprocessor					Clocks			
XADD mem,reg	XADD DATA5,ECX	8086	—							
	XADD [EBX],EAX	8088	—							
	XADD [ECX+4],EBP	80286	—							
		80386	—							
		80486	4							
		Pentium–Core2	3 or 4							
XCHG Exchange		Microprocessor					Clocks			
1000011w oorrrmmm		Microprocessor					Clocks			
Format	Examples	Microprocessor					Clocks			
XCHG reg,reg	XCHG CL,DL	8086	4							
	XCHG BX,DX	8088	4							
	XCHG DH,CL	80286	3							
	XCHG EBP,EBX	80386	3							
	XCHG EAX,EDI	80486	3							
		Pentium–Core2	3							

XCHG mem,reg reg,mem	XCHG DATAJ,CL XCHG BYTES,CX XCHG NUMBS,ECX XCHG [EAX],CX XCHG CL,POPS	8086	17 + ea
		8088	25 + ea
		80286	5
		80386	5
		80486	5
		Pentium–Core2	3
10010reg Format		Examples	Microprocessor Clocks
XCHG acc,reg reg,acc	XCHG BX,AX XCHG AX,DI XCHG DH,AL XCHG EDX,EAX XCHG SI,AX	8086	3
		8088	3
		80286	3
		80386	3
		80486	3
		Pentium–Core2	2
XLAT	Translate		
11010111 Example		Microprocessor	Clocks
XLAT		8086	11
		8088	11
		80286	5
		80386	3
		80486	4
		Pentium–Core2	4

XOR		Exclusive-OR								
Format	Examples	O 0	D *	I *	T ?	S *	Z 0	A 0	P 0	C 0
		Microprocessor					Clocks			
XOR reg,reg	XOR CL,DL XOR AX,DX XOR CH,CL XOR EAX,EBX XOR ESI,EDI	8086							3	
		8088							3	
		80286							2	
		80386							2	
		80486							1	
		Pentium–Core2					1 or 2			
XOR mem,reg	XOR DATAJ,CL XOR BYTES,CX XOR NUMBS,ECX XOR [EAX],CX	8086							16 + ea	
		8088							24 + ea	
		80286							7	
		80386							6	
		80486							3	
		Pentium–Core2					1 or 3			
XOR reg,mem	XOR CL,DATAL XOR CX,BYTES XOR ECX,NUMBS XOR DX,[EBX+EDI]	8086							9 + ea	
		8088							13 + ea	
		80286							7	
		80386							7	
		80486							2	
		Pentium–Core2					1 or 2			
Format	Examples	Microprocessor					Clocks			
XOR reg,imm	XOR CX,3 XOR DI,1AH XOR DL,34H XOR EDX,1345H XOR CX,1834H	8086							4	
		8088							4	
		80286							3	
		80386							2	
		80486							1	
		Pentium–Core2					1 or 3			

XOR mem,imm	XOR DATAS,3 XOR BYTE PTR[EDI],1AH XOR DADDY,34H XOR LIST,'A' XOR TOAD,1834H	8086	17 + ea
		8088	25 + ea
		80286	7
		80386	7
		80486	3
		Pentium–Core2	1 or 3
0010101w data Format		Examples	Microprocessor Clocks
XOR acc,imm	XOR AL,3 XOR AX,1AH XOR EAX,34H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
		Pentium–Core2	1

SIMD INSTRUCTION SET SUMMARY

The SIMD (single-instruction, multiple data) instructions add a new dimension to the use of the microprocessor for performing multimedia and other operations. The XMM registers are numbered from XMM_0 to XMM_7 and are each 128 bits in width. Data formats stored in the XMM registers and used by the SIMD instructions appear in Figure B-1.

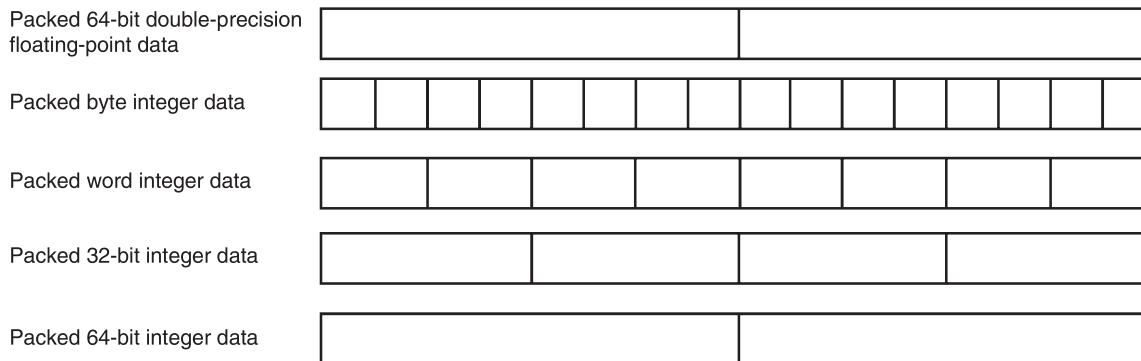


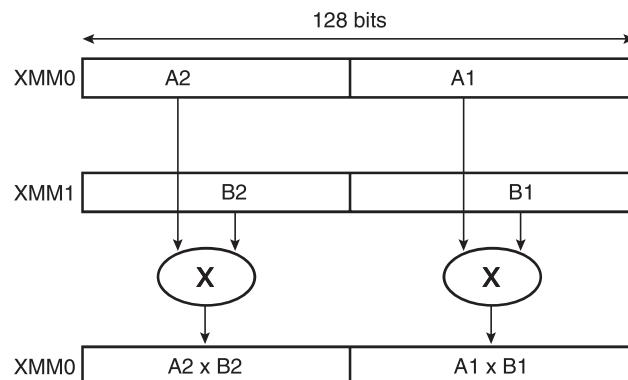
FIGURE B-1 Data formats for the 128-bit-wide XMM registers in the Pentium III and Pentium 4 microprocessors.

Data stored in the memory must be stored as 16-byte-long data in a series of memory locations accessed by using the OWORD PTR override when addressed by an instruction. The OWORD PTR override is used to address an octalword of data or 16 bytes. The SIMD instructions allow operations on packed and scalar double-precision floating-point numbers. The operation of both forms is illustrated in Figure B–2, which shows both packed and scalar multiplication. Notice that scalar only copies the leftmost double-precision number into the destination register and does not use the leftmost number in the source. The scalar instructions are meant to be compatible with the floating-point coprocessor instructions.

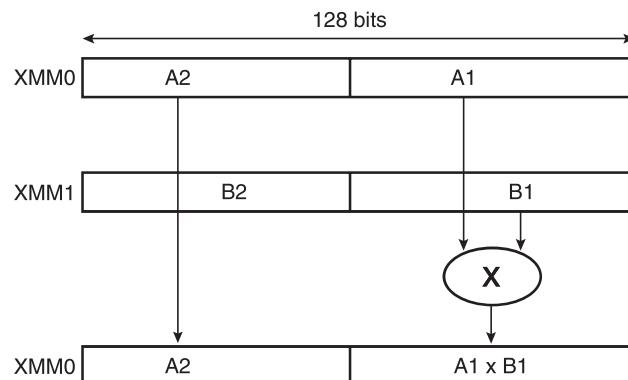
This section of the appendix details many of the SIMD instructions and provides examples of their usage.

FIGURE B–2 Packed and scalar double-precision floating-point operation.

Packed double-precision multiplication MULPD XMM0, XMM1



Packed double-precision multiplication MULSD XMM0, XMM1



DATA MOVEMENT INSTRUCTIONS

MOVAPD	Move aligned packed double-precision data, data must be aligned on 16-byte boundaries
Examples	
	MOVAPD XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVAPD OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVUPD Move unaligned packed double-precision data	
Examples	
	MOVUPD XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVUPD OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVSD	Move scalar packed double-precision data to low quadword
Examples	
	MOVSD XMM0, DWORD DATA3 ;copies DATA3 to XMM0 MOVSD DWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVHPD	Move packed double-precision data to high quadword
Examples	
	MOVHPD XMM0, DWORD DATA3 ;copies DATA3 to XMM0 MOVHPD DWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVLPD	Move packed double-precision data into low quadword
Examples	
	MOVLPD XMM0, DWORD DATA3 ;copies DATA3 to XMM0 MOVLPD DWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVMSKPD	Move packed double-precision mask
Examples	
	MOVMSKPD EAX, XMM1 ;copies 2 sign bits to general-purpose register

MOVAPS	Move 4 aligned packed single-precision data, data must be aligned on 16-byte boundaries
Examples	
	MOVAPS XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVAPS OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVUPS Move 4 unaligned packed single-precision data	
Examples	
	MOVUPS XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVUPS OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVLPS	Move 2 packed single-precision numbers to low-order quadword
Examples	
	MOVLPS XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVLPS OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVHPS	Move packed single-precision numbers to high-order quadword
Examples	
	MOVHPS XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVHPS OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVAPD	Move aligned packed double-precision data, data must be aligned on 16-byte boundaries
Examples	
	MOVAPD XMM0, OWORD DATA3 ;copies DATA3 to XMM0 MOVAPD OWORD PTR DATA4, XMM2 ;copies XMM4 to DATA4
MOVLHPS	Move 2 packed single-precision numbers from the low-order quadword to the high-order quadword
Examples	
	MOVLHPS XMM0, XMM1 ;copies XMM1 low to XMM0 high MOVLHPS XMM3, XMM2 ;copies XMM2 low to XMM3 high

MOVHLPS	Move 2 packed single-precision numbers from high-order quadword to low-order quadword
Examples	
MOVHLPS XMM0, XMM2	;copies high XMM2 to low XMM0
MOVHLPS XMM4, XMM5	;copies high XMM5 to low XMM4
MOVMSKPS Move 4-sign bits of 4 packed single-precision numbers to general-purpose register	
Examples	
MOVMSKPS EBX, XMM0	;copies sign bits of XMM0 to EBX
MOVMSKPS EDX, XMM2	;copies sign bits of XMM2 to EDX

ARITHMETIC INSTRUCTIONS

ADDPD	Adds packed double-precision data
Examples	
ADDPD XMM0, OWORD DATA3	;adds DATA3 to XMM0
ADDPD XMM2, XMM3	;adds XMM3 to XMM2
ADDSD Adds scalar double-precision data	
Examples	
ADDSD XMM0, OWORD DATA3	;adds DATA3 to XMM0
ADDSD XMM4, XMM2	;adds XMM2 to XMM4
ADDPS Adds 2 packed single-precision numbers	
Examples	
ADDPS XMM0, QWORD DATA3	;adds DATA3 to XMM0
ADDPS XMM3, XMM2	;adds XMM2 to XMM3

ADDLS	Adds scalar single-precision data
Examples	
ADDLS XMM0, DWORD DATA3	;adds DATA3 to XMM0
ADDLS XMM7, XMM2	;adds XMM2 to XMM7
SUBPD Subtracts packed double-precision data	
Examples	
SUBPD XMM0, OWORD DATA3	;subtracts DATA3 from XMM0
SUBPD XMM2, XMM3	;subtracts XMM3 from XMM2
SUBSD Subtracts scalar double-precision data	
Examples	
SUBSD XMM0, OWORD DATA3	;subtracts DATA3 from XMM0
SUBSD XMM4, XMM2	;subtracts XMM2 from XMM4
SUBPS Subtracts 2 packed single-precision numbers	
Examples	
SUBPS XMM0, QWORD DATA3	;subtracts DATA3 from XMM0
SUBPS XMM3, XMM2	;subtracts XMM2 from XMM3
SUBLS Subtracts scalar single-precision data	
Examples	
SUBLS XMM0, DWORD DATA3	;subtracts DATA3 from XMM0
SUBLS XMM7, XMM2	;subtracts XMM2 from XMM7
MULPD Multiplies packed double-precision data	
Examples	
MULPD XMM0, OWORD DATA3	;multiplies DATA3 times XMM0
MULPD XMM3, XMM2	;multiplies XMM2 times XMM3

MULSD	Multiplies scalar double-precision data
Examples	
MULSD XMM0, OWORD DATA3	;multiplies DATA3 times XMM0
MULSD XMM3, XMM6	;multiplies XMM6 times XMM3
MULPS Multiplies 2 packed single-precision numbers	
Examples	
MULPS XMM0, QWORD DATA3	;multiplies DATA3 times XMM0
MULPS XMM0, XMM2	;multiplies XMM2 times XMM0
MULSS Multiplies a single-precision number	
Examples	
MULSS XMM0, DWORD DATA3	;multiplies DATA3 times XMM0
MULSS XMM1, XMM2	;multiplies XMM2 times XMM1
DIVPD Divides packed double-precision data	
Examples	
DIVPD XMM0, OWORD DATA3	;divides XMM0 by DATA3
DIVPD XMM3, XMM2	;divides XMM3 by XMM2
DIVSD Divides scalar double-precision data	
Examples	
DIVSD XMM0, OWORD DATA3	;divides XMM0 by DATA3
DIVSD XMM3, XMM6	;divides XMM3 by XMM6
DIVPS Divides 2 packed single-precision numbers	
Examples	
DIVPS XMM0, QWORD DATA3	;divides XMM0 by DATA3
DIVPS XMM0, XMM2	;divides XMM0 by XMM2

DIVSS	Divides a single-precision number
Examples	
DIVSS XMM0, DWORD DATA3	;divides XMM0 by DATA3
DIVSS XMM1, XMM2	;divides XMM1 by XMM2
SQRTPD Finds the square root of packed double-precision data	
Examples	
SQRTPD XMM0, OWORD DATA3	;finds square root of DATA3, result to XMM0
SQRTPD XMM3, XMM2	;finds square root of XMM2, result to XMM3
SQRTSD Finds the square root of scalar double-precision data	
Examples	
SQRTSD XMM0, OWORD DATA3	;finds square root of DATA3, result to XMM0
SQRTSD XMM3, XMM6	;finds square root of XMM6, result to XMM3
SQRTPS Finds the square root of 2 packed single-precision numbers	
Examples	
SQRTPS XMM0, QWORD DATA3	;finds square root of DATA3, result to XMM0
SQRTPS XMM0, XMM2	;finds square root of XMM2, result to XMM0
SQRTSS Finds the square root of a single-precision number	
Examples	
SQRTSS XMM0, DWORD DATA3	;finds the square root of DATA3, result to XMM0
SQRTSS XMM1, XMM2	;finds the square root of XMM2, result to XMM1
RCPSS Finds the reciprocal of a packed single-precision number	
Examples	
RCPSS XMM0, OWORD DATA3	;finds the reciprocal of DATA3, result to XMM0
RCPSS XMM3, XMM2	;finds the reciprocal of XMM2, result to XMM3

RCPSS	Finds the reciprocal of a single-precision number
Examples	
RCPSS XMM0, OWORD DATA3	;finds the reciprocal of DATA3, result to XMM0
RCPSS XMM3, XMM6	;finds the reciprocal of XMM6, result to XMM3
RSQRTPS Finds reciprocals of packed single-precision data	
Examples	
RSQRTPS XMM0, OWORD DATA3	;finds reciprocal of square root of DATA3
RSQRTPS XMM3, XMM2	;finds reciprocal of square root of XMM2
RSQRTSS Finds the reciprocal of square root of a scalar single-precision number	
Examples	
RSQRTSS XMM0, OWORD DATA3	;finds reciprocal of square root of DATA3
RSQRTSS XMM3, XMM6	;finds reciprocal of square root of XMM6
MAXPD Compares and returns the maximum packed double-precision floating-point number	
Examples	
MAXPD XMM0, OWORD DATA3	;compares numbers in DATA3, largest to XMM0
MAXPD XMM3, XMM2	;compares numbers in XMM2, largest to XMM3
MAXSD Compares scalar double-precision data and returns the largest	
Examples	
MAXSD XMM0, OWORD DATA3	;compares numbers in DATA3, largest to XMM0
MAXSD XMM3, XMM6	;compares numbers in XMM6, largest to XMM3
MAXPS Compares and returns the largest packed single-precision number	
Examples	
MAXPS XMM0, QWORD DATA3	;compares numbers in DATA3, largest to XMM0
MAXPS XMM0, XMM2	;compares numbers in XMM2, largest to XMM0

MAXSS	Compares scalar single-precision numbers and returns the largest
Examples	
MAXSS XMM0, DWORD DATA3	;compares numbers in DATA3, largest to XMM0
MAXSS XMM1, XMM2	;compares numbers in XMM2, largest to XMM1
MINPD Compares and returns the minimum packed double-precision floating-point number	
Examples	
MINPD XMM0, OWORD DATA3	;compares numbers in DATA3, least to XMM0
MINPD XMM3, XMM2	;compares numbers in XMM2, least to XMM3
MINSD	Compares scalar double-precision data and returns the smallest
Examples	
MINSD XMM0, OWORD DATA3	;compares numbers in DATA3, least to XMM0
MINSD XMM3, XMM6	;compares numbers in XMM6, least to XMM3
MINPS	Compares and returns the smallest packed single-precision number
Examples	
MINPS XMM0, QWORD DATA3	;compares numbers in DATA3, least to XMM0
MINPS XMM0, XMM2	;compares numbers in XMM2, least to XMM0
MINSS	Compares scalar single-precision numbers and returns the smallest
Examples	
MINSS XMM0, DWORD DATA3	;compares numbers in DATA3, least to XMM0
MINSS XMM1, XMM2	;compares numbers in XMM2, least to XMM1

LOGIC INSTRUCTIONS

ANDPD	ANDs packed double-precision data
Examples	
ANDPD XMM0, OWORD DATA3	;ands DATA3 to XMM0
ANDPD XMM2, XMM3	;ands XMM3 to XMM2
ANDNPD NANDs packed double-precision data	
Examples	
ANDNPD XMM0, OWORD DATA3	;Nands DATA3 to XMM0
ANDNPD XMM4, XMM2	;Nands XMM2 to XMM4
ANDPS ANDs 2 packed single-precision data	
Examples	
ANDPS XMM0, QWORD DATA3	;ands DATA3 to XMM0
ANDPS XMM3, XMM2	;ands XMM2 to XMM3
ANDNPS NANDs 2 packed single-precision data	
Examples	
ANDNPS XMM0, DWORD DATA3	;Nands DATA3 to XMM0
ANDNPS XMM7, XMM2	;Nands XMM2 to XMM7
ORPD ORs packed double-precision data	
Examples	
ORPD XMM0, OWORD DATA3	;ors DATA3 to XMM0
ORPD XMM2, XMM3	;ors XMM3 to XMM2
ORPS ORs 2 packed single-precision numbers	
Examples	
ORPS XMM0, OWORD DATA3	;ors DATA3 to XMM0
ORPS XMM3, XMM2	;ors XMM2 to XMM3

XORPD	Exclusive-ORs packed double-precision data	
Examples		
XORPD	XMM0, OWORD DATA3	;exclusive-ors DATA3 to XMM0
XORPD	XMM2, XMM3	;exclusive-ors XMM3 to XMM2
XORPS	Exclusive-ORs packed double-precision data	
Examples		
XORPS	XMM0, OWORD DATA3	;exclusive-ors DATA3 to XMM0
XORPS	XMM2, XMM3	;exclusive-ors XMM3 to XMM2

COMPARISON INSTRUCTIONS

CMPPD	Compares packed double-precision numbers	
Examples		
CMPPD	XMM0, OWORD DATA3	;compares DATA3 with XMM0
CMPPD	XMM2, XMM3	;compares XMM3 with XMM2
CMPSD	Compares scalar double-precision data	
Examples		
CMPSD	XMM0, QWORD DATA3	;compares DATA3 with XMM0
CMPSD	XMM3, XMM2	;compares XMM2 with XMM3
CMPISD	Compares scalar double-precision data and sets EFAGS	
Examples		
CMPISD	XMM0, OWORD DATA3	;compares DATA3 with XMM0
CMPISD	XMM2, XMM3	;compares XMM3 with XMM2

UCOMISD	Compares scalar unordered double-precision numbers and changes EFLAGS	
Examples		
UCOMISD	XMM0, QWORD DATA3	;compares DATA3 with XMM0
UCOMISD	XMM3, XMM2	;compares XMM2 with XMM3
CMPSS	Compares packed single-precision data	
Examples		
CMPSS	XMM0, OWORD DATA3	;compares DATA3 with XMM0
CMPSS	XMM2, XMM3	;compares XMM3 with XMM2
CMPSS	Compares 2 packed single-precision numbers	
Examples		
CMPSS	XMM0, QWORD DATA3	;compares DATA3 with XMM0
CMPSS	XMM3, XMM2	;compares XMM2 with XMM3
COMISS	Compares scalar single-precision data and changes EFLAGS	
Examples		
COMISS	XMM0, OWORD DATA3	;compares DATA3 with XMM0
COMISS	XMM2, XMM3	;compares XMM3 with XMM2
UCOMISS	Compares unordered single-precision numbers and changes EFLAGS	
Examples		
UCOMISS	XMM0, QWORD DATA3	;compares DATA3 with XMM0
UCOMISS	XMM3, XMM2	;compares XMM2 with XMM3

DATA CONVERSION INSTRUCTIONS

SHUFPD	Shuffles packed double-precision numbers
Examples	
SHUFPD XMM0, OWORD DATA3	;shuffles DATA3 with XMM0
SHUFPD XMM2, XMM2	;swaps upper and lower quadword in XMM2
UNPCKHPD Unpacks the upper double-precision number	
Examples	
UNPCKHPD XMM0, OWORD DATA3	;unpacks DATA3 into XMM0
UNPCKHPD XMM3, XMM2	;unpacks XMM2 into XMM3
UNPCKLPD Unpacks the lower double-precision number	
Examples	
UNPCKLPD XMM0, OWORD DATA3	;unpacks DATA3 into XMM0
UNPCKLPD XMM3, XMM2	;unpacks XMM2 into XMM3
SHUFPS Shuffles packed single-precision numbers	
Examples	
SHUFPS XMM0, QWORD DATA3	;shuffles DATA3 with XMM0
SHUFPS XMM2, XMM2	;swaps upper and lower quadword in XMM2
UNPCKHPS Unpacks the lower double-precision number	
Examples	
UNPCKHPS XMM0, QWORD DATA3	;unpacks DATA3 into XMM0
UNPCKHPS XMM3, XMM2	;unpacks XMM2 into XMM3
UNPCKLPSD Unpacks the lower double-precision number	
Examples	
UNPCKLPSD XMM0, QWORD DATA3	;unpacks DATA3 into XMM0
UNPCKLPSD XMM3, XMM2	;unpacks XMM2 into XMM3

APPENDIX C

Flag-Bit Changes

This appendix shows only the instructions that actually change the flag bits. Any instruction not listed does not affect any of the flag bits.

<i>Instruction</i>	<i>O</i>	<i>D</i>	<i>I</i>	<i>T</i>	<i>S</i>	<i>Z</i>	<i>A</i>	<i>P</i>	<i>C</i>
AAA	?				?	?	*	?	*
?				*	*	?	*	?	
AAM	?				*	*	?	*	?
AAS	?				?	?	*	?	*
ADC	*				*	*	*	*	*
ADD	*				*	*	*	*	*
AND	0				*	*	?	*	0
ARPL					*				
BSF					*				
BSR					*				
BT									*
BTC									*
BTR									*
BTS									*
CLC									0
CLD		0							
CLI			0						
CMC									*
CMP	*				*	*	*	*	*
CMPS	*				*	*	*	*	*
CMPXCHG	*				*	*	*	*	*
CMPXCHG8B					*				
DAA	?				*	*	*	*	*
DAS	?				*	*	*	*	*
DEC	*				*	*	*	*	*
DIV	?				?	?	?	?	?
IDIV	?				?	?	?	?	?
IMUL	*				?	?	?	?	*
INC	*				*	*	*	*	*

<i>Instruction</i>	<i>O</i>	<i>D</i>	<i>I</i>	<i>T</i>	<i>S</i>	<i>Z</i>	<i>A</i>	<i>P</i>	<i>C</i>
IRET	*	*	*	*	*	*	*	*	*
LAR						*			
LSL						*			
MUL	*				?	?	?	?	*
NEG	*				*	*	*	*	*
OR	0				*	*	?	*	0
POPF	*	*	*	*	*	*	*	*	*
RCL/RCR	*								*
REPE/REPNE						*			
ROL/ROR	*								*
SAHF					*	*	*	*	*
SAL/SAR	*				*	*	?	*	*
SHL/SHR	*				*	*	?	*	*
SBB	*				*	*	*	*	*
SCAS	*				*	*	*	*	*
SHLD/SHRD	?				*	*	?	*	*
STC									1
STD		1							
STI			1						
SUB	*				*	*	*	*	*
TEST	0				*	*	?	*	0
VERR/VERW					*				
XADD	*				*	*	*	*	*
XOR	0				*	*	?	*	0