

APPENDIX A

The 8088 Instruction Set

The 8088 instruction set is summarized in the series of tables that makes up the bulk of this appendix. Examples of typical assembler-language statements are provided for each instruction listed. The time required to execute each instruction is indicated by the number of clocks specified. If "+EA" appears in this column, it indicates that time must be spent calculating the effective address of an operand that is located in main memory. This time depends on the addressing mode used to access the operand and can be obtained from Table A-1. The way each instruction affects the various 8088 flags is indicated in the box at the upper right corner of each table. The flag codes are explained in Table A-2.

The information provided here has been extracted from the *8086/8087/8088 Macro Assembly Language Reference Manual* and is reprinted with permission from Intel Corp.

Table A-1. Effective-Address Calculation Time

EA Components	Clocks*
Displacement Only	6
Base or Index Only	5
Displacement +	9
Base or Index	
Base	(BX, BP, SI, DI) BP + DI, BX + SI
+	
Index	BP + SI, BX + DI
Displacement	BP + DI + DISP BX + SI + DISP
Base	
+	BP + SI + DISP
Index	BX + DI + DISP
	12

* Add 2 clocks for segment override.

Table A-2. Key to Flag Codes

Code	Meaning
1	unconditionally set
0	unconditionally cleared
X	altered to reflect operation result
U	undefined (mask it out)
R	replaced from memory (e.g., SAHF)
b	[blank] unaffected

INSTRUCTION-SET TABLES

AAA	AAA (no operands) ASCII adjust for addition			OD ITSZAPC Flags U UUXUX	
	Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)		4	—	1	AAA

AAD		AAD (no operands) ASCII adjust for division		Flags OD ITSZAPC U XXUXU	
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	60	—	2	AAD	

AAM		AAM (no operands) ASCII adjust for multiply		Flags OD ITSZAPC U XXUXU	
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	83	--	1	AAM	

AAS		AAS (no operands) ASCII adjust for subtraction		OD ITSZAP C Flags U UUXUX	
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	4	—	1	AAS	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

ADC		ADC destination, source Add with carry			OD ITSZAPC Flags X XXXXX	
Operands	Clocks	Transfers*	Bytes	Coding Example		
register, register	3	—	2	ADC AX, SI		
register, memory	9+EA	—	2-4	ADC DX, BETA [SI]		
memory, register	16+EA	2	2-4	ADC ALPHA [BX] [SI], DI		
register, immediate	4	—	3-4	ADC BX, 256		
memory, immediate	17+EA	2	3-6	ADC GAMMA, 30H		
accumulator, immediate	4	—	2-3	ADC AL, 5		

ADD	ADD destination, source Addition				Flags	OD ITSZAP C X XXXXX
	Operands	Clocks	Transfers*	Bytes	Coding Example	
register, register	3	—	2	ADD CX, DX		
register, memory	9+EA	1	2-4	ADD DI, [BX], ALPHA		
memory, register	16+EA	2	2-4	ADD TEMP, CL		
register, immediate	4	—	3-4	ADD CL, 2		
memory, immediate	17+EA	2	3-6	ADD ALPHA, 2		
accumulator, immediate	4	—	2-3	ADD AX, 200		

AND	AND destination, source				Flags	OD ITSZAPC 0 XXUX0
	Logical AND					
Operands	Clocks	Transfers*	Bytes	Coding Example		
register, register	3	—	2	AND AL, BL		
register, memory	9+EA	1	2-4	AND CX, FLAG_WORD		
memory, register	16+EA	2	2-4	AND ASCII [DI], AL		
register, immediate	4	—	3-4	AND CX0, FOH		
memory, immediate	17+EA	2	3-6	AND BETA, 01H		
accumulator, immediate	4	—	2-3	AND AX, 01010000B		

CALL	CALL target		Flags	OD ITSZAPC
	Call a procedure			
Operands	Clocks	Transfers*	Bytes	Coding Examples
near-proc	19	1	3	CALL NEAR_PROC
far-proc	28	2	5	CALL FAR_PROC
mempr 16	21+EA	2	2-4	CALL PROC_TABLE [SI]
regpr 16	16	1	2	CALL AX
mempr 32	37+EA	4	2-4	CALL [BX], TASK [SI]

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

CBW		CBW (no operands) Convert byte to word		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	CBW	
CLC		CLC (no operands) Clear carry flag		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	CLC	
CLD		CLD (no operands) Clear direction flag		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	CLD	
CLI		CLI (no operands) Clear interrupt flag		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	CLI	
CMC		CMC (no operands) Complement carry flag		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	CMC	
CMP		CMP destination, source Compare destination to source		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
register, register register, memory memory, register register, immediate memory, immediate accumulator, immediate		3 9+EA 9+EA 4 10+EA 4	— 1 1 — 1 —	2-4 2-4 2-4 3-4 3-6 2-3	CMP BX, CX CMP DH, ALPHA CMP BP+2, SI CMP BL, 02H CMP [BX], RADAR [DI], 3420H CMP AL, 00010000B	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

CMPS		CMPS dest-string, source-string Compare string		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
dest-string, source-string (repeat) dest-string, source-string		22 9+22/ rep	2 2/rep	1	CMPS BUFF1, BUFF2 REPE CMPS ID, KEY	
CWD		CWD (no operands) Convert word to doubleword		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		5	—	1	CWD	
DAA		DAA (no operands) Decimal adjust for addition		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		4	—	1	DAA	
DAS		DAS (no operands) Decimal adjust for subtraction		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		4	—	1	DAS	
DEC		DEC destination Decrement by 1		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
reg16 reg8 memory		2 3 15+EA	— — 2	1 2 2-4	DEC AX DEC AL DEC ARRAY [SI]	
DIV		DIV source Division, unsigned		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
reg8 reg16 mem8 mem16		80-90 144-162 [86-96] (150-168) +EA	— — 1 1	2 2 2-4 2-4	DIV CL DIV BX DIV ALPHA DIV TABLE [SI]	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

ESC	ESC external-opcode,source Escape	Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes
Immediate, memory Immediate, register	8+EA 2	1 —	2-4 2
			ESC 6, ARRAY [SI] ESC 20, AL

HLT	HLT (no operands) Halt	Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes
(no operands)	2	—	1
			HLT

IDIV	IDIV source Integer division	Flags	OD ITSZAPC U UUUUU
Operands	Clocks	Transfers*	Bytes
reg8 reg16 mem8 mem16	101-112 165-184 (107-118) +EA (171-190) +EA	— — 1 1	2 2 2-4 2-4
			IDIV BL IDIV CX IDIV DIVISOR BYTE [SI] IDIV [BX], DIVT- SOR_WORD

IMUL	IMUL source Integer multiplication	Flags	OD ITSZAPC X UUUUX
Operands	Clocks	Transfers*	Bytes
reg8 reg16 mem8 mem16	80-98 128-154 (86-104) +EA (134-160) +EA	— — 1 1	2 2 2-4 2-4
			IMUL CL IMUL BX IMUL RATE_BYTE IMUL RATE_WORD [BP] [DI]

IN	IN accumulator, port Input byte or word	Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes
accumulator, imm8 accumulator, DX	10 8	1 1	2 1
			IN AL, OFFEAH IN AX, DX

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

INC	INC destination Increment by 1	Flags	OD ITSZAPC X XXXX
Operands	Clocks	Transfers*	Bytes
reg16 reg8 memory	2 3 15+EA	— — 2	1 2 2-4
			INC CX INC BL INC ALPHA [DI] [BX]

INT	INT interrupt-type Interrupt	Flags	OD ITSZAPC 00
Operands	Clocks	Transfers*	Bytes
imm8 (type=3) imm8 (type=3)	52 51	5 5	1 2
			INT 3 INT 67

INTR	INTR (external maskable in- terrupt) Interrupt if INTR and IF = 1	Flags	OD ITSZAPC 00
Operands	Clocks	Transfers*	Bytes
(no operands)	61	7	N/A
			N/A

INTO	INTO (no operands) Interrupt if overflow	Flags	OD ITSZAPC 00
Operands	Clocks	Transfers*	Bytes
(no operands)	53 or 4	5	1
			INTO

IRET	IRET (no operands) Interrupt Return	Flags	OD ITSZAPC RRRRRRRR
Operands	Clocks	Transfers*	Bytes
(no operands)	24	3	1
			IRET

JA/JNBE	JA/JNBE short-label Jump if above/Jump if not below not equal	Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes
short-label	16 or 4	—	2
			JA ABOVE

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

JAE/JNB		JAE/JNB short-label Jump if above or equal/Jump if not below		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JAE ABOVE_EQUAL	

JB/JNAE		JB/JNAE short-label Jump if below/Jump if not above nor equal		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JB BELOW	

JBE/JNA		JBE/JNA short-label Jump if below or equal/Jump if not above		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JNA NOT_ABOVE	

JC		JC short-label Jump if carry		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JC CARRY_SET	

JCXZ		JCXZ short-label Jump if CX is zero		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	18 or 6	—	2	JCXZ COUNT_DONE	

JE/JZ		JE/JZ short-label Jump if equal/Jump if zero		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JZ ZERO	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

JG/JNLE		JG/JNLE short-label Jump if greater/Jump if not less nor equal		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JG GREATER	

JGE/JNL		JGE/JNL short-label Jump if greater or equal/ Jump if not less		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JGE GREATER_EQUAL	

JL/JNGE		JL/JNGE short-label Jump if less/Jump if not greater nor equal		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JL LESS	

JLE/JNG		JLE/JNG short-label Jump if less or equal/Jump if not greater		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JNG NOT_GREATER	

JMP		JMP target Jump		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	15	—	2	JMP SHORT	
near-label	15	—	3	JMP WITHIN_SEGMENT	
far-label	15	—	5	JMP FAR_LABEL	
mem16	18+EA	1	2-4	JMP [BX],TARGET	
reg16	11	—	2	JMP CX	
mem32	24+EA	2	2-4	JMP OTHER_SEG [SI]	

JNC		JNC short-label Jump if not carry		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
short-label	16 or 4	—	2	JNC NOT_CARRY	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

JNE / JNZ		JNE / JNZ short-label Jump if not equal / Jump if not zero		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JNE NOT EQUAL	

JNO		JNO short-label Jump if not overflow		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JNO NO OVERFLOW	

JNP / JPO		JNP / JPO short-label Jump if not parity / Jump if parity odd		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JPO ODD PARITY	

JNS		JNS short-label Jump if not sign		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JNS POSITIVE	

JO		JO short-label Jump if overflow		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JO SIGNED_OVERFLOW	

JP / JPE		JP / JPE short-label Jump if parity / Jump if parity even		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JPE EVEN_PARITY	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

JS		JS short-label Jump if sign		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		16 or 4	—	2	JS NEGATIVE	

LAHF		LAHF (no operands) Load AH from flags		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		4	—	1	LAHF	

LDS		LDS destination, source Load pointer using DS		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
reg16, mem32		16 + EA	2	2-4	LDS S1, DATA, SEG [DI]	

LOCK		LOCK (no operands) Lock bus		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
(no operands)		2	—	1	LOCK XCHG FLAG, AL	

LODS		LODS source-string Load string		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
source-string (repeat) source-string		12 9 + 13/ rep	1 1/rep	1	LODS CUSTOMER NAME REP LODS NAME	

LOOP		LOOP short-label Loop		Flags		OD ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example	
short-label		17/5	—	2	LOOP AGAIN	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

LOOPE/ LOOPZ	LOOPE/LOOPZ short-label Loop if equal/Loop if zero	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
short-label	18 or 6 — 2	LOOPE AGAIN	

LOOPNE/ LOOPNZ	LOOPNE/LOOPNZ short-label Loop if not equal/Loop if not zero	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
short-label	19 or 5 — 2	LOOPNE AGAIN	

LEA	LEA destination, source Load effective address	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
reg16, mem16	2+EA — 2-4	LEA BX, [BP] [DI]	

LES	LES destination, source Load pointer using ES	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
reg16, mem32	16+EA 2 2-4	LES DI, [BX], TEXT_BUFFER	

NMI	NMI (external nonmaskable interrupt) if NMI = 1	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
(no operands)	50 5 N/A	N/A	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

MOV	MOV destination, source Move	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
memory, accumulator accumulator, memory register, register register, memory memory, register register, immediate segment, register segment, register register, segment memory, segment	10 10 2 8+EA 9+EA 4 10+EA 2 8+EA 2 9+EA	1 1 — 1 1 — 1 — 1 1	3 3 2 2-4 2-4 2-3 3-6 2 2-4 2 2-4
			MOV ARRAY [SI], AL MOV AX, TEMP_RESULT MOV AX, CX MOV BP, STACK_TOP MOV COUNT [DI], CX MOV CL, 2 MOV MASK [BX] [SI], 2 CH MOV ES, CX MOV DS, SEGMENT_BASE MOV BP, SS MOV [BX], SEG_SAVE_CS

MOVS	MOVS dest-string, source-string Move string	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
dest-string, source-string (repeat) dest-string, source-string	18 9+17/ rep	2 2/rep 1	MOV LINE_EDIT_DATA REP MOV SCREEN_BUFFER

MOVSB/ MOVSW	MOVSB/MOVSW (no operands) Move string (byte/word)	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
(no operands) (repeat) (no operands)	18 9+17/ rep	2 2/rep 1	MOVSB REP MOVSW

MUL	MUL source Multiplication, unsigned	Flags	OD ITSZAPC
Operands	Clocks Transfers* Bytes	Coding Example	
reg8 reg16 mem8 mem16	70-77 118-133 (76-83) +EA (124-139) +EA	— — 1 1	2 2 2-4 2-4
			MUL BL MUL CX MUL MONTH [SI] MUL BAUD_RATE

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

NEG	NEG destination Negate			Flags	
				OD ITSZAPC	X XXXX1*
Operands		Clocks	Transfers*	Bytes	Coding Example
register		3	—	2	NEG AL
memory		16+EA	2	2-4	NEG MULTIPLIER

* 0 if destination = 0

NOP		NOP (no operands) No Operation		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	3	—	1	NOP	

NOT		NOT destination Logical NOT		Flags OD ITSZAPC	
Operands	Clocks	Transfers*	Bytes	Coding Example	
register memory	3 16+EA	— 2	2 2-4	NOT AX NOT CHARACTER	

OR	OR destination, source Logical inclusive OR	Flags			OD ITSZAPC 0 XXUX0
	Operands	Clocks	Transfers*	Bytes	Coding Example
register, register register, memory memory, register accumulator, immediate register, immediate memory, immediate	3 9+EA 16+EA 4 4 17+EA	— 1 2 — — 2	2 2-4 2-4 2-3 3-4 3-6	OR AL, BL OR DX, PORT ID [DI] OR FLAG BYTE, CL OR AL, 0110110B OR CX, 01FH OR [BX] CMD WORD, 0CFH	

OUT		OUT port, accumulator Output byte or word		Flags OD ITSZAPC	
Operands		Clocks	Transfers*	Bytes	Coding Example
immed8, accumulator DX, accumulator		10 8	1 1	2 1	OUT 44, AX OUT DX, AL

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

POP	POP destination Pop word off stack		Flags		OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
register seg:reg (CS illegal) memory	8 17+EA	1 1 2	1 1 2-4	POP DX POP DS POP PARAMETER	

POPF		POPF (no operands) Pop flags off stack		Flags OD ITSZAPC RRRRRRRRRR	
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	8	1	1	POPF	

PUSH		PUSH source Push word onto stack	Flags OD ITSZAPC		
Operands		Clocks	Transfers*	Bytes	Coding Example
register	11	1	1	1	PUSH SI
seg:reg (CS legal)	10	1	1	1	PUSH ES
memory	16+EA	2	2-4		PUSH RETURN CODE [SI]

PUSHF		PUSHF (no operands) Push flags onto stack		Flags	OD ITSZAPC
Operands	Clocks	Transfers*	Bytes	Coding Example	
(no operands)	10	1	1	PUSHF	

RCL	RCL destination, count Rotate left through carry			Flags	
				OD	ITSZAPC
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	RCL CX, 1	
register, CL	8+4/bit	—	2	RCL AL, CL	
memory, 1	15+EA	2	2-4	RCL ALPHA, 1	
memory, CL	20+EA+ 4/bit	2	2-4	RCL [BP], PARAM, CL	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

RCR		RCR destination, count Rotate right through carry		Flags	
				OD ITSSZAPC X X	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	RCR BX, 1	
register, CL	8+4/bil	—	2	RCR BL, CL	
memory, 1	15+EA	2	2-4	RCR [BX], STATUS, 1	
memory, CL	20+EA+ 4/bil	2	2-4	RCR ARRAY [DI], CL	

REP		REP (no operands) Repeat string operation		Flags	
				OD ITSSZAPC	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	REP MOVSB, DEST, SRCE	

REPE/REPZ		REPE/REPZ (no operands) Repeat string operation while equal/while zero		Flags	
				OD ITSSZAPC	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	REPE CMPSB, DATA, KEY	

REPNE/ REPNZ		REPNE/REPNZ (no operands) Repeat string operation while not equal/not zero		Flags	
				OD ITSSZAPC	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	REPNE SCASB, INPUT LINE	

RET		RET optional-pop-value Return from procedure		Flags	
				OD ITSSZAPC	
Operands		Clocks	Transfers*	Bytes	Coding Example
(intra-segment, no pop)	8	1	1	RET	
(intra-segment, pop)	12	1	3	RET 4	
(inter-segment, no pop)	18	2	1	RET	
(inter-segment, pop)	17	2	3	RET 2	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

ROL		ROL destination, count Rotate left		Flags	
				OD ITSSZAPC X X	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	ROL BX, 1	
register, CL	8+4/bil	—	2	ROL DI, CL	
memory, 1	15+EA	2	2-4	ROL FLAG BYTE [DI], 1	
memory, CL	20+EA+ 4/bil	2	2-4	ROL ALPHA, CL	

ROR		ROR destination, count Rotate right		Flags	
				OD ITSSZAPC X X	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	ROR AL, 1	
register, CL	8+4/bil	—	2	ROR BX, CL	
memory, 1	15+EA	2	2-4	ROR PORT STATUS, 1	
memory, CL	20+EA+ 4/bil	2	2-4	ROR CMD WORD, CL	

SAHF		SAHF (no operands) Store AH into flags		Flags	
				OD ITSSZAPC RRRRR	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	4	—	1	SAHF	

SAL/SHL		SAL/SHL destination, count Shift arithmetic left/Shift logical left		Flags	
				OD ITSSZAPC X X	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	SAL AL, 1	
register, CL	8+4/bil	—	2	SHL DI, CL	
memory, 1	15+EA	2	2-4	SHL [BX], OVERDRAW, 1	
memory, CL	20+EA+ 4/bil	2	2-4	SAL STORE_COUNT, CL	

SAR		SAR destination, source Shift arithmetic right		Flags	
				OD ITSSZAPC XXUXX	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	2	SAR DX, 1	
register, CL	8+4/bil	—	2	SAR DI, CL	
memory, 1	15+EA	2	2-4	SAR N BLOCKS, 1	
memory, CL	20+EA+ 4/bil	2	2-4	SAR N BLOCKS, CL	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

SBB		SBB destination, source Subtract with borrow		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, register	3	—	1	2	SBB BX, CX
register, memory	9+EA	—	2	2-4	SBB DI, [BX], PAYMENT
memory, register	16+EA	—	2	2-4	SBB BALANCE, AX
accumulator, register	4	—	—	2-3	SBB AX, 2
register, immediate	4	—	—	3-4	SBB CL, 1
memory, immediate	17+EA	2	—	3-6	SBB COUNT [SI], 10

SCAS		SCAS dest-string Scan string		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
dest-string	15	1	1	1	SCAS INPUT_LINE
(repeat) dest-string	9+15/ rep	1/rep	1	1	REPNE SCAS BUFFER

SHR		SHR destination, count Shift logical right		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, 1	2	—	—	2	SHR SI, 1
register, CL	8+4/bit	—	2	2	SHR SI, CL
memory, 1	15+EA	2	2-4	2-4	SHR ID BYTE [SI] [BX], 1
memory, CL	20+EA+ 4/bit	2	2-4	2-4	SHR INPUT WORD, CL

SINGLE STEP		SINGLE STEP (Trap flag in- terrupt) if TF = 1		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	50	5	N/A	N/A	

STC		STC (no operands) Set carry flag		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	1	STC

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

STD		STD (no operands) Set direction flag		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	1	STD

STI		STI (no operands) Set interrupt enable flag		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	—	1	1	STI

STOS		STOS dest-string Store byte or word string		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
dest-string	11	1	1	1	STOS PRINT LINE
(repeat) dest-string	9+10/ rep	1/rep	1	1	REP STOS DISPLAY

SUB		SUB destination, source Subtraction		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, register	3	—	2	2	SUB CX, BX
register, memory	9+EA	1	2-4	2-4	SUB DX, MATH TOTAL [SI]
memory, register	16+EA	2	2-4	2-4	SUB BP+2, CL
accumulator, immediate	4	—	2-3	2-3	SUB AL, 10
register, immediate	4	—	3-4	3-4	SUB SI, 5280
memory, immediate	17+EA	2	3-6	3-6	SUB [BP], BALANCE, 1000

TEST		TEST destination, source Test or nondestructive logical AND		Flags	
Operands		Clocks	Transfers*	Bytes	Coding Example
register, register	3	—	2	2	TEST SI, DI
register, memory	9+EA	1	2-4	2-4	TEST SI, END_COUNT
accumulator, immediate	4	—	2-3	2-3	TEST AL, 00100000B
register, immediate	5	—	3-4	3-4	TEST BX, 0004H
memory, immediate	11+EA	—	3-6	3-6	TEST RETURN CODE, 01H

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

WAIT (no operands)		Flags	
Wait while <u>TEST</u> pin not asserted		OD ITSZAPC	
Operands	Clocks	Transfers*	Bytes
(no operands)	3+5n	—	1
		Coding Example	
		WAIT	

XCHG destination,source Exchange		Flags	
		OD ITSZAPC	
Operands	Clocks	Transfers*	Bytes
accumulator, reg16	3	—	1
memory, register	17+EA	2	2-4
register, register	4	—	2
		Coding Example	
		XCHG AX, BX XCHG SEMAPHORE, AX XCHG AL, BL	

XLAT source-table Translate		Flags	
		OD ITSZAPC	
operands	Clocks	Transfers*	Bytes
source-table	11	1	1
		Coding Example	
		XLAT ASCII_TAB	

XOR destination,source Logical exclusive OR		Flags	
		OD ITSZAPC Flags 0 XXUX0	
Operands	Clocks	Transfers*	Bytes
register, register	3	—	2
register, memory	9+EA	1	2-4
memory, register	16+EA	2	2-4
accumulator, immediate	4	—	2-3
register, immediate	4	—	3-4
memory, immediate	17+EA	2	3-6
		Coding Example	
		XOR CX, BX XOR CL, MASK BYTE XOR ALPHA [SI], DX XOR AL, 01000010B XOR SI, 00C2H XOR RETURN CODE, 0D2H	

* For the 8086, add four clocks for each 16-bit word transfer with an odd address. For the 8088, add four clocks for each 16-bit word transfer.

APPENDIX B

References

1. The IBM Personal Computer Technical Reference Manual
2. The IBM Personal Computer Disk Operating System Manual
3. The IBM Personal Computer Macro Assembler Manual
4. Intel IAPX 86 Book
5. Intel 8086 Family User's Manual
6. Intel 8086/8087/8088 Macro Assembly Language Reference Manual for 8086-Based Development Systems
7. Intel product specifications for the following devices:
 - A. 8255 Programmable Peripheral Interface
 - B. 8259 Interrupt Controller
 - C. 8237 DMA Controller
 - D. 8253 Timer
8. Motorola product specification for the 6845 CRT Controller