Table 5-3. 8096 Memory Addressing Options Identified by the EA Abbreviations in Tables 5-4, 5-5, and 5-6

Morman         Size of Earling	Memory	Segment	-					
None   Si		Register	Register	Index Register	16-Bit Unsigned	8-Bit High-order Bit Extended	None	Language Operand Mnemonic
CS, SS or ES)				B	×	×	×	
CS, SS or ES)         BX         DI         X		şo	amov.	ā	*	×	×	
SS   Nome   Nome   X		(Alternate)		Sı	×	×	×	
None	2///	.S. 55 or ES.	×a	10	×	×	×	
SS	Метогу			None	×	×	*	
SS	aguera	\$0	None	Mone	*			
CS, DS or ES)  SS SP None SI  ES None DI  CS PC None None  CS PC None  DS DX None CSA.  These columns contribute to OEA.		S.		īS	×	×	×	
SS   SP		(Alternate)	40	jo.	×	×	×	
SS         SP         None           DS         None         SI           ES         None         DI           CS         PC         None           CS         PC         None           DS         DX         None           These columns contribute to DEA.         These columns contribute to EA.	7	SS, DS or ESI		None	×	×		
DS         None         SI           ES         None         Di           CS         PC         None           CS         PC         None           DS         DX         None           These columns contribute to DEA.         These columns contribute to EA.	Stack	SS	SP	None				
ES         None         DI         X           CS         PC         None         X           CS         PC         None         X           DS         DX         None         X           These columns contribute to OEA.         These columns contribute to EA.	String	SO	None	ıs				
CS         PC         None         X           CS         PC         None         X           DS         DX         None         None           These columns contribute to OEA.         These columns contribute to EA.	Data	ES	None	а				
CS         PC         None         X           DS         DX         None         These columns contribute to OEA.           These columns contribute to EA.	fruction	s	2	None				
DS DX None These columns contribute to OEA.  These columns contribute to EA.	ranch	so	PC	None		×		
	O Data	SO	X	None				
			Ţ	se columns co	intribute to OEA			This column
The state of the s		ı	Ŧ	se columns co	entribute to EA.			to be provided
	Sha	Shaded row applies to EA and LABEL.	to EA and LAB	ä	X These ar	X These are displacements that can be used to compute	at can be use	d to compute

## The following abbreviations are used in Tables 5-4 and 5-5:

Accumulator, high-order byte Accumulator, low-order byte

The value of register AL high-order bit (0 or 1) extended to a byte (0016 or FF16) Accumulator, both bytes

The value of register AH high-order bit (0 or 1) extended to a 16-bit word (000016 or FFF16). The destination is a byte operand (used only by the Assembler)

B register, high-order byte

B register, low-order byte

Program memory direct address, used in Branch addressing option shown in Tables 5-1 and 5-2. The source is a byte operand (used only by the Assembler)

B register, both bytes Carry status

C register, high-order byte

C register, low-order byte Code Segment register C register, both bytes

Data memory address operands identified in Table 5-3 Eight bits of immediate data

16 bits of immediate data

D register, high-order byte Destination Index register An 8-bit or 16-bit signed displacement An 8-bit signed displacement

D register, Jow-order byte
Data Segment register
D register, both bytes
Effective data memory address using any of the memory addressing options identified in Table 5-2

Extra Segment register

Status flag set to 1

Increment/decrement selector for string operations; increment if D is 0, decrement if D is 1 Direct data memory address, as identified in Table 5-2 A number between 0 and 7 Status flag reset to 0 

Offset data memory address used to compute EA: EA =OEA + [DS] • 16 Program Counter

I/O port addressed by DX register contents; port number can range from 0 through 65,536 A label identifying an I/O port number in the range 0 through 25510 Any one of the eight byte registers: AH, AL, BH, BL, CH, CL, DH, or DL Any RB register as a destination Any RB register as a source

Any one of the eight 16-bit registers: AX, BX, CX, DX, SP, BP, SI, or DI Any RW register as a destination

Any RW register as a source Label identifying a 16-bit value loaded into the CS Segment register to execute a segment jump

Status Flags register 

Any one of the Segment registers CS, DS, ES, or SS Stack Segment register

The destination is a word operand (used only by the Assembler)
The source is a word operand (used only by the Assembler)
Contents of the memory location addressed by the contents of the location enclosed in the double Data on the right-hand side of the arrow is moved to the location on the left-hand side of the arrow Contents of locations on each side of —— are exchanged The twos complement of the value under the — The contents of the location enclosed in the brackets Any number in the range 0 through 25510 Status flag modified to reflect result Status flag modified, but undefined Not equal to brackets =1 | 1\*

## INSTRUCTION EXECUTION TIMES AND CODES

Table 5-5 lists instructions in alphabetical order, showing object codes and execution times, for the 8086 and the 8088, expressed in whole clock cycles. Execution time is the time required from beginning execution of an instruction that is in the queue to beginning execution of the next instruction in the queue. The time required to place an instruction from memory into the queue (instruction fetch time) is not shown in the table; because of queuing, instruction fetch time and thus has no effect on overall timing, except as specifically noted in the table.

Instruction object codes are represented as two hexadecimal digits for instruction bytes without variations.

Instruction object codes are represented as eight binary digits for instruction bytes with variations for the instruction.

The following notation is used in Tables 5-4 and 5-5:

11 causes bbb to select a register, using the 3-bit code given below for reg. in bit position 0 a=0 specifies 1 data byte; a=1 specifies 2 data bytes in bit position 1 a=0 specifies 2 data bytes: a=1 specifies 1 data byte represents three binary digits identifying a destination register (see reg.) represents two hexadecimal digit memory displacement two binary digits identifying a segment register: two DISP bytes = 10, or 00 with bbb = 110 three bits choosing addressing mode:
000 EA = (8X) + (5I) + DISP
001 EA = (8X) + (DI) + DISP
001 EA = (8P) + (5I) + DISP
011 EA = (8P) + (DI) + DISP
100 EA = (5I) + DISP
101 EA = (BP) + DISP
110 EA = (BP) + DISP
111 EA = (BP) + DISP
111 EA = (BX) + DISP 00 = ES 01 = CS 10 = SS 11 = DS three binary digits identifying a register: indicate an optional object code byte two bits choosing address length: no DISP = 00 one bit choosing length: one DISP byte = 01 DISP ddd rr qqq aa

ēg

000 = AX 001 = CX 010 = DX 100 = BX 101 = BP 110 = SI 111 = DI

represents three binary digits identifying a source register (see reg) represents four hexadecimal digit memory address one bit choosing shift length: sss PPOO

0 count =

1 count = (CL) "don't care" bit

represents two hexadecimal data digits represents four hexadecimal data digits

one bit where z XOR (ZF) = 1 terminates loop 

Includes up to eight clock cycles of overhead on each transfer due to queue maintenance. For conditional jumps, the lesser figure is when the test fails (no jump taken). Execution time is less than or equal to instruction fetch time.

Effective Address calculation and extra clock cycles:

_		Extra Clock Periods		
	bbb	EA	8086(1)	8088(2)
	000 000 000 000 001 001 011 011 110 111	(BX) + (SI) (BX) + (SI) + DISPB (BX) + (SI) + DISP16 (BX) + (DI) (BX) + (DI) + DISP16 (BX) + (DI) + DISP16 (BP) + (SI) (BP) + (SI) (BP) + (SI) (BP) + (DI) (DI) (DI) (DI) (DI) (DI) (DI) (DI)	7 112 8 8 12 12 12 12 12 12 12 12 12 12 12 12 12	7 111 112 112 113 114 115 115 115 115 115 115 115 115 115
	(1) Add 8 16-bi (2) Add 8 16-bi Substitute	<ol> <li>Add another 4 clock cycles for each         <ul> <li>16-bit operand or an odd address boundary.</li> </ul> </li> <li>Add anoter 4 clock cycles for each         <ul> <li>16-bit operand.</li> </ul> </li> <li>Substitute the clock cycles shown above wherever EA appears in Tables 5-4 and 5-5.</li> </ol>	in Tables 5-4	and 5-5.

1	۰	٦	7
4	ú	ė	Ĺ
3	•	١,	,

·	т-	_	_	_	_	_		_	_					
the ES register (RB) $\leftarrow$ [EA] $\rightarrow$ 100 ADDR to size one by 100 ADDR to a decreased by 100 ADDR to 1										A3+8	dddbbbee A8 {92IQ][92IQ]	R00A0,88	VOM	Primar
segment register contents, oreates the offective data memory addresses [RW] = [EA], [ES] — [EA+2] — [EA+2] — [EA] bits of test from the memory word addressed by DADDR into register RW. Load 15 bits of data from the next sequential memory word into					-					43+9ſ	C4 assasbbb	яаааа, мя	\$37	Primary Memory R
the DS register [RW] — OEA Losd into RW the 16-bit address displacement which, when added to the				ĺ						S+E∀	dddsasse G8 [qSlQ][qSlQ]	RODAO,WR	rev	Reference
[SA+3] → [KA], [A3] → [WR] string the Market of the Marke										A3+81	C5 8888sbbb (DISP)[DISP]	AGGAG,WA	\$07	,
F+TRO9 for Not ontents are output of NO port PORIS. [FPO9] [FDM]. [FDM]										8	ΕE	[XQ],XA	TUO	
in the XX ant ni [HA] → [LL], [PORT+1] → [HA] → [L + TRO9] (LL), [FORT+1] → [TRO9] (LT + TRO9) (LT +										Οţ	K) //	TRO9,XA	TUO	
TRO9 noq O\l ot JA retriger most stab to eryd eno tuqtuO  [JA] — [XG9] bled al aserbbe esoriw noq O\l erit of JA retriger most stab to eryd eno tuqtuO										8	L 33	[xa]'¬∀	TUO	
si searbes eachw thop OVI ent most state seviesen HA. near the IVO port whose address is the ligher. [AI — [TRO]										Ot	E9 AA	TRO¶,1A	TUO	\$/0
receives data from I/O port POT+1 [AL] — (PDX+1] [AL] — (PDX+1] [AL] — (PDX+1) (And 16 bits of data into AX. AL receives data from I/O port whose address is										8	ED .	[XQ],XA	NI	
register [AR] — [HA], [HA] (— [AR] —										Ot -	EP AA	TЯ0Ч,ХА	NI	
JA othi TRO9 that O\I mort stat to etyd eno bead [JA] → [JA] XG ent ni bled at seatbas seartw that O\I mort stat to etyd eno JA othi baod										8	EC 1	(DX)	Ni	
[JA] → [JA]	Ц	L	Ļ	L	L	╙	L	Ц		οι	E4 XX	T809,⊥A	NI	<u> </u>
Operation Performed	2	d	٧	_	:m)#	18	١,	а	0	Clack Cycles	Object Code	(s)bnsteqO	pinomenM	Туре

Table 5-4. A Summary of 8086 and 8088 Instructions (Continued)

(A3) — [WA1]  [A4] — [WA1]  [A5] — [WA2]  [B7] — [WA2]  [B7] — [A3]  [B8] — [B8]  [	D	d	_	2	_	_		3 0	#+EA 9+EA 10 10 10 10 10 10 10	BB seaddbbb BB	ROGAG,WR RR,ROGAG HRAJ,IA HRAJ,IA HALJBBAJ TA,JBBAJ XA,JBBAJ RA,JBBAJ RR,RGGAG	Mnemonic MOV MOV MOV MOV MOV	Primary Memory Reference (Continued)
With the Second Part of the state of the Second Part of the Second Part of the Second Part of Se									A3+6 O1 O1 O1 O1 O1	(92)[0](92)[0]  [92)[0](92)[0]  [92)[0](92)[0]  [92)[0](92)[0]  DD99 FA  DD99 SA  DD99 SA  BE saChrobb  OddnObe 38	89,RODAQ W8,RODAD 128AJ,IA 138AJ,XA 138AJ XA,I38AJ XA,I38AJ	MOW AOM AOM	Memory Reference (Continued)
WRI — [RR]  Jefore the dess byte from register RB in the memory byte addressed by [A]  Store the dess byte town the provide and the memory byte addressed by [A]  Jefore the febit date word throw register RW in the memory word and by DADBA by DABBA by LABBA by LABBA into register or in LBBA by LABBA care the 8-bit date memory word directly addressed by LABBA by LABBA care the 8-bit contents of register Ab into the date memory byte directly addressed by LABBA									01 01 01 01 01	ddesesebbe [qgi][qgi] ddesesebbe 89 esesebbe Ggi][qgi] DD99 FA DD99 SA DD99 SA BE seOmbbb	MA, FODAQ JAA, LABEL JAA, JABAJ XA, JABAJ YA, JABAJ	MOW AOM	Memory Reference (Continued)
BBI → [A3] Alto beseabbe sixed women entities in BR is selected state of the date of the date of the date of the date to force it.  [WH] → [AM] ← [MM] ← [MM] ← [MM] Both which is selected to the date which is selected to the bod is selected by LABE   [AA] → [AA] ← [AA]									01 01 01 01 01	[92IO][92IO] [92IO][92IO] [92IO][92IO] [92IO][92IO] OD99 FA  DD99 SA  DD99 & A  DD99 & SA  DD99 & SA  DD99 & SA  SEE SEE SEE SEE SEE SEE SEE SEE SEE SE	MA, FODAQ JAA, LABEL JAA, JABAJ XA, JABAJ YA, JABAJ	MOW AOM	Memory Reference (Continued)
(MRI) — [A3]  Store ha for the determinant brown step is 16 - bit determinant by I MoDAD with a float a float a float a float and a float a float and a float a float and a float a fl									or or or A3+8	fq8tq](q8tq) ODqq 0A ODqq rA ODqq sA ODqq sA ODqq sA ODqq sA ODqq sA	138AJ,XA 138AJ, 1A,138AJ XA,138AJ	MOV MOV	Memory Reference (Continued)
Store the f 6-bit date word from register RW in the memory word add by DADDR  [AJ] — [EA]  Load the date memory byte directly addressed by LABEL into register [AX] — [EA]  [AX] — [EA]  Load the f 6-bit date memory word directly addressed by LABEL into rown for the 8-bit contents of register AL into the date memory byte direct addressed by LABEL  [EA] — [A]  ADDRESSED TO SEGMENT OF SEGMENT AT INTO The date memory byte direct addressed by LABEL  [SR] — [EA]  Load into Segment register SR the contents of the 16-bit memory word or diessed by DADBR  [ABEL] — [AB]  Load Into Segment register SR the contents of the 16-bit memory word or the 8-bit memory word or 16-bit memory location or 16-bit word or 16-bit memory location or 16-bit word or 16-bit wor									or or or A3+8	[089][018] DD99 DA DD99 FA DD99 SA SPOQ BE 880Tbbb DD99 SB S90[089]	138AJ,XA 138AJ, 1A,138AJ XA,138AJ	MOV MOV	Memory Reference (Continued)
ABCAL by DADDA by the disterent by DADDA by DAD									01 01 10	DD99 fA DD99 SA DD99 SA DD99 SA DD99 SA DD99 SA SSECTION SECTION SECTI	J38AJ. JA.J38AJ XA.J38AJ RGGAG.RS	AOM AOM	Memory Reference (Continued)
[AB] — [LA] to date memory byte directly addressed by LABL into the date the date of the date of the date into the date of the date memory word directly addressed by LABL into in contents of the date of the date into the date memory byte directly addressed by LABL    [AB] — [AB] — [AB]  Store the 8-bit contents of register AL into the date memory byte directly addressed by LABL    [AB] — [AB] — [AB]  Load into Segment register SR the contents of the 18-bit memory work of the 18-bit of the ABBL    [AB] — [AB] — [AB] → [AB]    ModGal with the 18-bit date of Segment register SR in the 18-bit memory work of the 28-bit memory work of the 28-bit memory work of the 28-bit memory and the 28-bit memory locations the contents of Septembra of Segment register SR in the 18-bit memory locations and store the contents of Segment register SR in the 18-bit memory locations.									01 01 10	DD99 FA DD99 SA A3 PPQQ BE sa0mbb block BE sa0mbb Boddnobs Boddnobs Boddnobs Boddnobs Boddnobs	J38AJ. JA.J38AJ XA.J38AJ RGGAG.RS	AOM AOM	memory reference (continued)
LAX   — [KA] — [KA] best-vit data memory word directly addressed by LABEL into to Load the 16-bit data memory word directly addressed by LABEL [KA] — [KA] Store the 8-bit contents of register AL into the data memory byte direct A[KA] — [KA] = 3 store the state memory word or Store the 16-bit contents of register AS into the data memory word or Store the 16-bit memory word or load into Sugment register SR the contents of the 16-bit memory word or into Sugment register SR the contents of the 16-bit memory word [KA] — [KB] = 16-bit memory word into Sugment of Store the contents of Store the contents of Sequent register SR in the 16-bit memory locations the contents of Sequent register SR in the Store the contents of Sequent register SR in the Store the contents of Sequent register SR in the 16-bit memory locations.									01 01 10	DD99 SA  PD099 &A  BE self[G2][92][92]  Squmbb  Squmbb	JA,1384.J XA,1384.J RGGAG,R2	VOM	selected to continued to
Load the 16-bit data memory word binectly addressed by LABEL into to XX  [EA] — [A2]  [EA] — [A2]  Store the 8-bit contents of register AL into the data memory byta direct dereaded by LABEL  Growth 16 - [AX]  Store the 16-bit contents of register AX into the data memory word of the 16-bit contents of register AX into the data memory word of the 16-bit memory word of the 16-bit memory word into Segment register SR the contents of the 16-bit memory word into Segment register SR the contents of the 16-bit memory word into Segment register SR the contents of the 16-bit memory word into Segment register SR the contents of Segment register SR in International Store the contents of Segment register SR in International Store the contents of Segment register SR in International Store the contents of Segment register SR in International Store the contents of Segment register SR in International Store the contents of Segment register SR in International Store the contents of Segment register SR in International Store Store the contents of Segment register SR in International Store Store Store International Segment register SR in International Segment Segment register SR in International Segment register SR in International Segment Segment Segment Register SR in International Segment Segment Register SR in International Segment Segment Register SR international Segment Register SR interna									10 10	DD99 SA  PD099 &A  BE self[G2][92][92]  Squmbb  Squmbb	JA,1384.J XA,1384.J RGGAG,R2	VOM	Memory Reference (Continued)
XA    EA] — [AI]  Store the 8-bit contents of register AL into the data memory byte direct dreesed by LABEL  (EA) — [AZ]  Store the 16-bit contents of register AZ into the data memory word or stored the 16-bit contents of the stored by LABEL  (SR) — [AR]  Load into Segment register SR the contents of the 16-bit memory word or dreesed by DADDR  (EA) — [SR]  (EA) — [SR]  (EA) — [SR]  Store the contents of Segment register SR in the 16-bit memory locations the contents of the stored by Store the contents of Segment register SR in the Store the contents of Section of Section (Stored Stored Stor									10 8+EA	A3 PPGG BE saOrrbbb [DISP][DISP]	XA,136AJ RGGAG,R2	VOM	memory Reference (Continued)
[EA] — [AI] Store the 8-bit contents of register AL into the data memory byte direct divesced by LABEL  [AI] — [AI]  Store the 16-bit contents of register AX into the data memory word of addressed by LABEL  [SR] ← [EA]  Load into Sugment register SR the contents of the 18-bit memory word into Sugment register SR the contents of the 18-bit memory word into Sugment register SR the contents of the 18-bit memory words.									10 8+EA	A3 PPGG BE saOrrbbb [DISP][DISP]	XA,136AJ RGGAG,R2	VOM	Memory Reference (Continued)
beside any time 8-bit continues of register AL into the 8-bit source of the 8-bit denserod by LABEL denserod by LABEL [EA] $-$ [AZ] $-$ [									10 8+EA	A3 PPGG BE saOrrbbb [DISP][DISP]	XA,136AJ RGGAG,R2	VOM	memory Hererence (Continued
directed by LABEL Store that $(EK) - [KX] = [KX] - [KX]$ Store tha 16-bit contents of register KX into the data memory word of backesed by LABEL [SR] $\leftarrow$ [EA] Load into Segment register SR the contents of the 16-bit memory work of the search by DADDR [EK] $\leftarrow$ [SR] $\leftarrow$ [SR] Store the contents of Segment register SR in the 16-bit memory locations the contents of Segment register SR in the 16-bit memory locations.									B+E∀	BE seOmbbb [DiSP]	FIGGAG,R2	AOW	memory nererence (Contin
[KA] — [KA] folion on the folion of the first state of the folion of folion of the folion of folio									B+E∀	BE seOmbbb [DiSP]	FIGGAG,R2	AOW	Memory Heterence (Co.
descended by LABEL [R2] $\rightarrow$ [R2] $\rightarrow$ [R2] included by Degment register SR the contents of the 16-bit memory we MODAD by the SR [R2] $\rightarrow$ [R2] $\rightarrow$ [R2] included by Degment register SR in the 16-bit memory locations the contents of Sequence 18-bit memory locations.										[DISP][DISP]			Memory Hererence
SP] → [RA] boad into Segment register SR the contents of the 16-bit memory wo interest by PADDR  RA] → [SR]  Store the contents of Segment register SR in the 16-bit memory locati										[DISP][DISP]			memory Reference
Load into Segment register SA the contents of the 1-bit memory wo disessab W DADDA [AA] — [SR] Store the contents of Segment register SR in the 16-bit memory locati										[DISP][DISP]			wemary Heter
of resect by DADDR [EA] — [SR] in the 16-bit memory locations the contents of Segment register SR in the 16-bit memory locations the contents of Segment register SR in the 16-bit memory locations.									∀3+6	8C esOmbbb	88,800 <u>A</u> 0	VOM	Meinory no
EA) — [SR) Store the contents of Segment register SR in the 18-bit memory locet:									¥3+6		R8,800A0	VOM	Memory
Store the contents of Segment register SR in the 16-bit memory locati						1	ŀ	ı			1		3
		ŀ		Ш		-	ŀ	1	1				1 3
		- 1	1 1						1	1	1 1		1 ~
[BH] ←→ [BH]	1 1	- 1	1 1	Ιl	- 1		1		17+EA	dddgeres 86	ROOAO,8R	хсне	Ì
Exchange a byte of data between register RB and the data memory in	Н	٠.	Н	1	ı	- [	1	1	Į.	[4810][4810]	1 1		lŝ
AddAd yd besserbbs	Ιl		П	П	l	- 1	1		1	1	1		١.
	Ιl		П	П	- 1	-1	١		A3+C1		ядама, мя	хсне	1
	Ιl	- 1	1	П	. 1		ŀ	1	I	[Jein[Jein]	1		l
	Н	- 1	П	П	ſ	-1	ſ	1	1 "	40	1 1	TAJX	1
	H	- 1	П	1	. 1	- 1	- 1	Т					ı
stnetnoo X8 ritiw stnetnoo AA laitini gnim	ŀΙ		П	Н	1	- 1	1	Т	1			1	ı
	Н		Н	Н	1	- 1	1	1			1		l
	П		Ш	Ш	- 1	- 1					1		ı
·	ı		Ш	П	IJ	- 1	1		]		1		,
			Ш	Н		- 1	-1	Т			1		
		- 1	Ш	Н		-	1	ı			1		ı
	Exchange a byte of data between register RB and the data incomery is addressed by DADDR addressed by DADDR [RW] [RA] and the data memory is EXI by DADDR addressed by DADDR [AL] [RM] [AL]	Exchange a byte of data between register RB and the data memory is addressed by DADDR  [RW] [EA]  Exchange 16 bits of data between register RW and the data memory is addressed by DADDR  [AL] [AL] + [BX]]  Load into AL the data byte stored in the memory location addressed b	In younger stab ents bas 8A setsliger neewted stab to etyd so genericzen procession for add by documents and by the WR]  [A3] [WR]  In younger stab ent bas WR setsliger neewted state to state the water sets and readers and reader	Exchange a byte of data between register RB and rive data memory ly floribate addressed by DADDA  [RW] [EA]  Exchange 15 bits of data between register RW and the data memory list addressed by DADDA  (AL) [EX]  Load into AL the data byte stored in the memory location addressed by	Exchange a byte of data between register RB and the data memory ly sidesesed by DADDA  [RW] [EA]  Exchange 15 bits of data between register RW and the data memory list addressed by DADDH  [AL] [AL]  Load into AL the data byte stored in the memory location addressed b	Fxchange a byte of data between register RB and the data memory ly florible addressed by DADDA  [May] [EA]  Exchange 15 bits of data between register RW and the data memory list addressed by DADDA  [A]  [A]  [A]  [A]  [A]  [A]  [A]	Profinence a byte of data between register RB and the data memory ly floresced by DADDA addressed by DADDA by the office of data between register RW has addressed by DADDA byte stored in the memory location addressed by the CBX].  [AL] — [LA] + [BX]]  [Load into AL the data byte stored in the memory location addressed by the stored in the memory location and the stored in the memory location addressed by the stored in the memory location addressed by the stored in the memory location and the stored in the location and the stored in the store	Profinence a byte of data between register RB and the data memory ly fromers addressed by DADDA  [May] [EA]  Exchange 15 bits of data between register RW and the data memory list addressed by DADDA  [A]  [A]  [A]  [A]  [A]  [A]  [A]	Exchange a byte of data between register RB and the batta between the data mannery is addressed by DADDA  [RW] [EA]  Exchange 15 bits of data between register RW and the data memory is dedressed by DADDA  [LAL] [LAL] [LAL]  Load into AL the data byte stored in the memory location addressed by	Pathere of data between register RB and the data between register RB and the data memory ly NDDDR  A∃+∇Γ  [RW] → → [RA]  Exchange 15 bits of data between register RW and the data memory lite data memory lite (BX)]  I Γ  Load into AL the data byte stored in the memory location addressed b	[4][1][4][2]   [4][4][4]   [4][4][4][5]   [4][4][4][5][5][5][5][5][5][5][5][5][5][5][5][5]	MODACI (PSIGI)   Fractions are to tast by eavle stable or to tast between the Brita bases when the Brita bases of the Brita ba	Exchange a byte of data between register RB and the data memory lightlight [Dig9] [Dig

				891	ENJE	ns,								ě
bелтотье <b>Я</b> лойвтедО	2	d	٧	Z	S	ī	1	a	0	Clock Cycles	Object Code	Operand(s)	lainamen M	Туре
[BB] ← [EA] + [BB] + [C]	х	x	X	x	įχ	L.	_	L	X	A3+6	dddbbbse S1	ACCAC,8R	ADC	
Add the contents of the data byte addressed by DADDR, plus the Carry status.	H				1	l			1		[dSiq][dSiq]			l
8A netaigen of						l								
$[BM] \leftarrow [EV] + [BM] + [C]$	[x	x	х	х	x	l	П	1	x	∀3+6	dddbbbss £1	RW, DADDR	DŒ	1
Add the contents of the 16-bit data word addressed by DADDR, plus the Cer-		l					l				[4810][4810]	l .		
WR 1stuc, to register PI	I^	^	^	^	l^			Į į	l^	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	44400000 01	000000	Juv	l
[EA] [EA] + [RB] + [C]	[v	νI	v	Į,	l۷	l		1	x	43+91	dddesess Of [98IQ][98IQ]	89,900A0	oa∀	l
Add the 8-bit contents of register RB, plus the Cerry status, to the data memory byte addressed by DADDR											(Jeig][Jeig:			l
$[EV] \leftarrow [EV] + [BM] + [C]$	×	х	х	х	×		ı		×	A3+81	dddasaes FI	WA,AGGAG	ODA	l
Add the 16-bit contents of register RW, plus the Carry status, to the date	Ш						i	1 '			[DISh][DISh]			
RQQAQ yd bassaibba biuw				l		ı						l		Ē
[88] ← [EA] + [88]	x	×	x	х	Ιx	l			x	∀3+6	dddbbbss 50	<b>900</b> A0,8Я	QQA	3
8R retaiger of ROGAG yd besserbbe etyd stab eith fo stenetroo erit bbA	^	^ <b> </b>	^	l^	<b> </b> ^	l	ı	'	1	1	[PISP][DISP]		ų.	õ
[WR] + [A3] → [WR]	I۷I	ایا	¥	v	l,		1	]	×	∀3+6	fazinifazini	ROGAG,WR	aa∀	(Memory Operate)
WR retaigen of RGGAD vd besserbts brow rid-81 for the contents of the $A=A$ (AB) + [AB] + [AB] $A=A$	<b>*</b>	^	^	١*	*				X	16+EA	[98jg][98jg] dddsssss 00	BR.ROOAO	₫₫¥	3
yd besenbas eityd ynomen eiter AB to the data memory byte addressed by	ارا	υl	U	l,	l,	ĺ		1	J	M3401	[DISESSED OO	חחיים	200	
AGDAD		l			l						I must law.			
[W8] + [A3] → [A3]	Ι×	×	X	×	×			]	Ιx	A3+81	dddasasa 10	WR,ROGAO	QQA	ě
beaserbbs brow yromem sisb arth of WR reteigen to atnetnop fid-81 arth bbA	1				П		l				[DISP][DISP]			\$
ыр рАрок				l										Ä
[88] → [88]	0	x	n	х	x	l	ı		0	∀3+6	dddbbbes SS	яаала.вя	αν¥	]
besserbbs ayyd yromem steb ent fittiw 8R tetsigat to stratfroot tid-8 ent GMA	ı						ı			l	[4810][4810]			Secondary Memory Reference
8A in the result in RB	ľ	ſ٦	'''	1	^				ľ	73.0	444444	addva ma	CINT	7
IRW] ← [EA] AND [RW]	اما	٧l	0	×	v			١ ١	اها	∀3+6	dddbbbss 82	AGGAG,WR	ανΑ	阜
-bs brow yromem steb ent thiw WH with the date memory word ad-	П										[4810][4810]			ğ
dressed by DADDR. Store the negalt in RW  [EA] — [EA] AND [RB]	ľ	×۱	"	×	ľ×				0	A3+91	dddesass OS	BR.ROOAG	QNA	Š
bessenbbs sited ynomem sists entit this BR issues to structure side. 8 ent GNA	۱		_	١					۱	10101	[018b][D18b]			
by DADDR. Store the result in the addressed data memory byte	П					1			1					
[EA] ← [EA] AND [MM]	0	x	n	×	×	l		Į I	0	16+EA	dddeeses 12	WR,RODAG	<b>GNA</b>	l
-be brow ynomem stab ent fittiw WR refraiger to strietnes tid-81 and dNA	ł			İ	l	İ					[ASIQ][ASIQ]			
brow yromem steb bearerbbs ent in the subtanged data memory word	"	'''	^	"	<u> </u>			1	"	12.0	14000000			
(R8) – (EA)	ΙxΙ	ŀχΙ	x	x	lχ				x	¥3+6	dddbbbes AS	яаада,ая	CWP	
Subtract the contents of the data memory byte addressed by DADDR from the	l			1		1		1	1	·	(ASIG)(ASIG)			
contents of register RB. Discard the result, but adjust status flage [RW] - [EA]	^	<b>*</b>	^	١^	١^	ı			×	¥3+6	38 sedddbbb	POGAG,WR	СМР	
FIGURE 16-bit contents of the data memory word addressed by DADDR	IJ	IJ	J	L	J			1	L	۷۵۱۸	[DISE][DISE]			

Table 5-4. A Summary of 8086 and 8088 Instructions (Continued)

(beunitino)) anothautent 8808 bins 6808 to gramming A  $\,\text{P-G}$  eldeT

bemiched noisseqO	Г		_	88	ŝuţ	81S				Clock Cycles	eboO toeldO	(s)bnsraq0	oinomen M	γpe
	-	_	_	z	•	+	Ī	a	0					,
[EA] – [RB] Subtract the 8-bit contents of register RB from the data memory byte ad-	1 <sub>×</sub> 1	x	x	x	×		1	П	х	9+€∀	dddseses 88 [92][92][	89,800A0	смь	
dressed by DADDR. Discard the result, but adjust status flags	H			l		ŀ	l	H			( ioid)( ioid)			l
[WR] – [A3]	۱×۱	X	×	×	×	1	1		х	<b>∀</b> 3+6	dddesass 65	WA, A d d A d	CMP	l
Subtract the 16-bit contents of register RW from the data memory word ad-				1				l l			[DISh][DISh]			l
dressed by DADDA. Discard the solut, but adjust status flags	ı			1			l	Ιl						i
f - [A3] [A3]	ı	X	х	×	×		l	ı	x	15+EA	effffff	Radac	DEC	
Decrement the contents of the memory location addressed by DADDR. De-				l			1				ddd f OOes	]		l
pending on the prior definition of DADDR, an 8-bit or a 16-bit memory loca-	1			l			l	Ιl			[DISE][DISE]	l		l
tion may be decremented  flow may be decremented	I"I	11	Γ''	۱"	ا"	1	l	Ιl	"	¥27(30 38)	44401100 83	23040 X4	All	<u>B</u>
[AX] ← [AX]/[EA] vomem att to stratnos tid-8 att vid XA sersions to stratnos tid-8 t att objuito.	ا۱	_			6		ı		n	¥3+( 96-98)	F6 as 1 (Obbb	ROGAG,XA	DIA	ΙĒ
Divide the 16-bit contents of register XX by the 8-bit contents of the memory byte addressed by DADDR. Store the integer quotient in AL and the remainder	Н			1			l	l l			[45][0][45][0]			ă
of the executed by Decipier is greater than FF16, execute a "divide by O" interrupt	1			1			l							(Continued)
[DX] $[XX]$ $[XX]$ $[XX]$	<u>ا</u> ۱۰	U	lυ	u	lu	1	l		n	(160-168)+EA	dddoff BB 73	AGGAG,XG	DIA	€
yd (rebro-wol) XA bris frebro-dgirl} XC arstalger to atmestree £1-5€ erit ebivid	ŀ. l				L		l	Ιl			[DISP][DISP]			Ιš
the 16-bit contents of the memory word addressed by DADDR. Store the in-	Ιl				1	1	l							불
eger quotient in AX and the remainder in DX. If the quotient is greater than	IJ			1		1	l	l l						Ž
FFFF16, execute a "divide by O" interrupt	П						l	Ιl				1		18
	n	n	n	l II	n	1	l	Ιİ	n	A3+8(811-701)	ddd11188 87	.RGGAG,XA	IDIA	Memory Reference (Memory Operate)
Divide the 16-bit contents of register AX by the 8-bit contents of the memory	Ιl			i	1	1	l	Ιl			[AS(0][AS(0]	l		=
byte addressed by DADDR, treating both contents as signed binary numbers:	[ ]					1	l							į
Store the quotient, as a signed binary number, in AL. Store the remainder, as	Ιl						l							Ιŝ
esel to .a.f.17 rish research in the quotient is greater than 7F16, or less	H						l							3
[DX] (EX] ← [DX] (EX]/(EA]	l٩l	п	I'n	ļη	ľ	l	l		n	A3+(081-(171)	dddfffes 74	ADQAQ,XQ	AIGI :	Į₹
bivide the 32-bit contents of register DX (high-order) AA bne (self-order)	ارا		Ĺ	_	Ĺ	I	l	l		V3+4001-41413	[DISE][DISE]	ugakatya	AIGI	Įž
the 16-bit contents of the memory word addressed by MODAU treat both	H			1	i i	l		[		i	[ local[ loca]	l		₹
contents as signed binary numbers. Store the quotient, as a signed binary				1			l							3
II. HA ii. Store the remainder, as an unaigned binary number, in HA ii.	Ιl				l	1						l		Secondary
the quotient is greater than 7FFF16, or less than -800016, execute a "divide	Ιl				l	l						l		ĕ
by O" interrupt	ľΙ		l	l					- 1					စ္တ
$[A\exists] \cdot [A] \rightarrow [XA]$	lx!	n	n,	n	n	ı		1	×	(86-104)+EA	dddfOfas 84	AL, DADOR	ואחר	l
Multiply the 8-bit contents of register AL by the contents of the memory byte	H		l			1		H			[4810][4810]			l
addressed by DADDR. Treat both numbers as signed binary numbers. Store	łΙ	l	İ			1		i						1
XA ni fouborg fid-8f ent	1^		Ι	١	Ι	ı	ļ		^.	11.(001 701)	44400 23	444		Į
[A3] - [XA] — [XA] X4 Notices to steeped tid 85 act visibility	I۷I	0	'n	n	١٨		1		v	(134-160)+EA	dddf0188 77	ADDAG,XA	ואַחר	ı
Multiply the 16-bit contents of register AX by the 16-bit contents of the memory word addressed by DADA. Treat both numbers as signed binary	H		ı	J			]				[4810][4810]			
the second bases of the control of the second secon	H		ı				l	Ш						ı
word)	ΙI		ı	1		ı	1					1		1

1	ū	ľ	
		١	
١	ľ	ď	١
ľ	7	•	

	5	۵	₽	+	s	+	+	┿	10	<del>                                     </del>		<del></del>	
[EA] + 1 Increment the contents of the memory location addressed by DADDR. De panding on the prior definition of DADDR, an 8-bit or a 16-bit memory local panding on the prior definition of DADDR, an 8-bit or a 16-bit memory local		×	x	×	×				×	A3+&1	81111111 dd000aa [98][98][	ROGAO	JNI:
tion may be incremented [AX] — [AL] • [EA] • [AL] • [EA] • [AL] • [EA] • [AL] • [EA] • [AL] • [EA] • [AL] •	×	n	n	٥	٦	י			×	A3+(£8-91)	ddd001se 87 [92][92][93]	ROOAQ,JA	J∪M
XA rif Job product in XX [A2] (A3] - [XA] - [XA] [XQ] (A3] - [XA] [XQ] (A3] - [XA] [XQ] (A1] (A2] (A2] (A2] (A2] (A3] (A3] (A3] (A3] (A3] (A3] (A3] (A3	×	n	n	n	(	,			×	(124-138)+EA	43 (012P)	7.3	אחר
numbers. Store the 32-bit product in DX Ihigh-order word) and AX (low-orde word) [EA] — [EA] Twos complement the contents of the addressed memory location. Depend ing on the prior definition of DADDR, an 8-bit or 16-bit memory location maing on the prior definition of DADDR, an 8-bit or 16-bit memory location mains on the prior definition of DADDR, and 8-bit or 16-bit memory location mains on the prior definition of DADDR, and 8-bit or 16-bit memory location mains of the prior definition of DADDR, and 8-bit or 16-bit memory location.	×	×	×	×	×				×	A3+8f	ettottit dditose (92IQ][92K]	#₫₫ <b>A</b> ₫	NEG
be twos complemented EA] — NOT [EA]  Case complement the contents of the addressed memory location. Depending  ons to complement the contents of the addressed memory location. Depending  on the prior definition of DADDA, as 8-bit or 16-bit memory location may be										A3+81	8f101111 ddd0108a [98][][98][]	AGGAG	TON
ones complemented [RB] $\sim$ [EA] OR [RB] OR (RB) with the data memory byte addressed by the 8-bit contents of register RB with the data memory byte addressed by	×	×	n	x	×				×	∀3+6	dddbbbss AO [qgl0](q2l0]	наола, вн	но
DADDR. Store the result in RB [RW] [RW] OF IDPR. Store the result in RW with the data memory word addressed OR that 16-bit contents of register RW with the data memory word addressed by DADR. Store the result in RW	×	×	n	×	×				×	9+E∀	dddbbbaa 80 [98][98]	AGGAG,WA	ЯO
[EA] or [RA] OR [RB] OB the 8-bit contents of register RB with the data memory byte addressed by DADDR. Store the result in the data memory byte	×	×	n	×	×				×	16+EA	dddasses 80 [98(0][98(0]	8 <b>9,800A</b> 0	яo
(A3) → [A3] MOR (RW)  MA 16-bit contents of register PM with the data memory word soldsesed by DADDB.  Stoop the result in the data memory word	×	×	n	×	×				×	A3+81	dddsses 60 [9810][9810]	WA,AddAq	ЯO

(beunitno) and substance of 808 bases are Submined A. A.-B. eldeT

or DADDR may address a word:  [EA1]	MA-20+EA 101000ve Add000be Add	emary Operatel (Continued)
[I+A3]	A3+E1 DADDR.N 110100ve N=115+EA	emory Operate) (Continued)
	ASP N=115+EA N=15+EA	emory Op
	ddd100ee (98/0][98/0]	8
X Rotate the contents of the data memory location addressed by DADDR left.  Move the left most bit into the Carry status. If N = 1, then rotate one bit positions.  Ton. If N = CL, then register CL contents provides the number of bit positions.  Depending on prior definition, DADDR may address a byte:  C = [EA]  C = [EA]	ROL DADDR,N 110100ve N>1 68000bbb 4N+20+EA Districtions∀1	Secondary Memory Reference (Memory Operate) (Continued)

	r
	7
Ļ	r.

bemoties notated	Г		_		SM1			_	┪	Clock Cycles	Object Code	(a) bneneqO	binomenM	γpe
	5	ď	٧	2	s	1		a c	តា					7
As AOL, but rotate right:	×							,	, l	A3+61 1=N	6v001011 ddd100ee [98!0][98!0]	N,ROOAG		
Shift the contents of the data memory location addressed by DADAN left. Wow if $t = t$ is the content of the data of the solution of the solu											( IDIO) ( IDIO)		<b>7∀\$</b>	
tion, if $N=CL$ , then register CL contents provides the number of bit positions. Depending on prior definition, DADDR may address a byte:														
C ← [EA] ← O														đ
or DADDR may address a word:			1											Conun
(C)														erate) (
(EV+1) ——-0														Memory Reference (Memory Operate) (Continued
:ngis stegedorg bre stight stids and "LAS aA	×	X	n	×	x			3	×	N≖1 15+EA;	84001011	И,ЯОДАД	ЯAS	e uvjeni
										N>1 4N+20+EA	ddd f 1 fes [9810][9810]			i di di C
(EA)														3
(EV)														
D ← (I+A3)					l									a province y
(RB) ← (FA) – (C)	×	×	×	X	×				×	43+6	ddd bbbse A1	ROOAG, BR	ed BS	١.
Subtract the contents of the data byte addressed by DADBA from the con- tents of 8-bit register RB, using twos complement arithmetic. Decrement the											(4\$10)[4\$10]			
result in RB if the Carry status was initially set  [RW] — [RW] = [EA] = [C]  Subtract the contents of the 16-bit data word addressed by DADDR from the	x	×	x	x	x			,	×	9+€∀	dddbbes 81 (92(0)[92(0)]	AGGAG,WA	88\$	
contents of the 16-bit register RW, using twos complement arithmetic. Decrement the result in RW If the Cerry status was initially set			ı	ì			П		1					ŀ

Table 5-4. A Summary of 8086 and 8088 Instructions (Continued)

Operation Performed	o	ď	٧	z	s	Ŧ	Ī	a	0	Clock Cycles	Object Code	Operandial	oinomenM	Typ
Subtract the contents of 8-bit register RB from the data byte addressed DADDR, using twos complement arithmetic. Decrement the result in da	×	x	×	×	×				x	A3+81	dddaaaee 81 [92id][92id]	88,800A0	88S	
Subtract the contents of 16-bit register RW from the 16-bit data word a	x	x	×	×	×				x	43+8ſ	dddesess 6 [ [93IO][93IO]	wa,яddAg	88\$	
unesseu of whom the framework of the control of the	x	×	n	×	×				×			и,яаала	THS	
:High Hids but, JAS aA			ı	1	1	1			x	;A3+&f f=W f <m< td=""><td>8v001011 dd10,188</td><td>N,ROQAQ</td><td>инs</td><td>tinued</td></m<>	8v001011 dd10,188	N,ROQAQ	инs	tinued
0 - [EA] - C										<del>¢</del> И+50+Ε₩	[48]Q][48]Q]			te) (Con
(FEA)														Secondary Memory Reference (Memory Operate) (Continued
(I+A3)														ence (M
Subtract the contents of the data memory byte addressed by DADDR from t	x	x	x	х	×				x	9+E∀	dddbbbee AS (9810)[9810]	Я <b>ФФФФ</b>	ens	y Refer
[RW] — [RW] — [EA] Subtract the contents of the 16-bit data memory word addressed by DADE	×	x	x	×	×				×	43+6	28 seddbbbb [qqbbb	яаама, мя	ens	Memor
Subtract the contents of 8-bit register RB from the data memory byte a Subtract the contents of 8-bit register RB from the data memory byte a	x	х	×	×	×				x	A3+81	dddssaes 82 (DISP)(DISP)	BR,RODAO	aus	condary
	x	×	×	x	x				x	18+EA	dddssess 82 [93(0)[93(0)]	WA,AGGAG	Ens	ş
JUAD yd besserbbe neitecol ynomem steb ant to atnatnoc tid-8 ant DNA	0	x	n	x	×				0	8+E <b>∀</b>	dddgeres 48 [92i0][93i0]	89,900A0	1\$31	
wint the contents of 8-oit register his. Discard the result, out adjust state flags appropriately														
	DADDR, using twos complement enthimetic. Decrement the result in dismemory if the Carry status was initially set memory if the Carry status was initially set Subtract the contents of 16-bit register RW from the 16-bit data word a in data memory if the Carry status was initially set.  This is an alternate mnemonic for SAL  As SAL, but shift right:  Or   [EA]  Or   [EA]  (RW) - [EA]  Subtract the contents of the data memory byte addressed by DADDR from the contents of 8-bit register RB, using twos complement arithmetic contents of the 16-bit register RW, using twos complement arithmetic contents of 18-bit register RB, using twos complement arithmetic contents of 18-bit register RB, using twos complement arithmetic contents of 18-bit register RW, using twos complement arithmetic contents of 18-bit register RW, using twos complement arithmetic contents of 18-bit register RW, using twos complement arithmetic software the contents of 18-bit register RW, using twos complement arithmetic contents of 18-bit register RW, using twos complement arithmetic software the contents of 18-bit register RW, using twos complement arithmetic contents of 18-bit register RW, using twos complement arithmetic software the contents of 18-bit register RW from the data memory byte a Subtract the contents of 18-bit register RW from the data memory byte a Subtract the contents of 18-bit register RW from the data memory byte a Subtract the contents of 18-bit register RW from the data memory byte a Subtract for contents of 18-bit register RW from the data memory byte a Subtract for contents of 18-bit register RW from the data memory byte a Subtract for contents of 18-bit register RW from the data memory byte a Subtract for contents of 18-bit register RW from the data memory byte a Subtract for contents of 18-bit register RW from the data memory byte a Subtract for contents of the data memory for complement arithmetic for the formatic register RW from the data memory byte a subtract RW from the formatic for the formatic for the formatic for the	DADDR, using twos complement airthmetic. Decrement the result in dismemory if the Carry status was initially set  Subtract the contents of 16-bit register RW from the 16-bit data word a reased by DADDR, using two complement airthmetic. Decrement the rea in data memory if the Carry status was initially set  This is an alternate mnemoric for SAL  X  As SAL, but shift right:  O    [EA]   [RM]   [EA]     [CA+1]     [C	The Manual of the contents of B-bit register RW trom the determent the result in distributed to the contents of 16-bit register RW from the 16-bit data word a Subtract the contents of 16-bit register RW from the 16-bit data word in the result in data memory if the Carry status was initially set.  X X As SAL, but shift right:  10	The Management of the contents of 16-bit register RW from the 16-bit data word initially set    EA] — [EA] — [RW] — [C]  Subtract the contents of 16-bit register RW from the 16-bit data word a decreased by DADDR, using twos complement antihmetic. Decrement the residue to the test of 16-bit register RW from the Data triplet of SAL.    This is an alternate memoring to the SAL   This is an alternate memoring to the SAL   C	ModDB, using twos complement sithmetic. Decrement the result in dismination and the result in dismination by the Carry status was initially set adversarily the Carry status was initially set decrement the contents of 16-bit tegister RW from the 16-bit data word in data memory if the Carry status was initially set in data memory if the Carry status was initially set.  X X X X X X X X X X X X X X X X X X X	Module the central in distingtion of the central in distingtion of the result in distingtion of the central in distingtion of the central indication of the data memory byte addressed by DADDR from the contents of the data memory byte addressed by DADDR from the central indication of the data memory byte addressed by DADDR from the central indication of the data memory word addressed by DADDR from the data memory word addressed by DADDR, using twos complement arithmetic subtract the contents of the 16-bit register RW, using twos complement arithmetic subtract the contents of the 16-bit register RW, using twos complement arithmetic subtract the contents of the 16-bit register RW, using twos complement arithmetic subtract the contents of the 16-bit register RW from the data memory byte a Subtract the contents of the 16-bit register RW from the data memory byte a Subtract the contents of the data memory word addressed by DADDR, using twos complement arithmetic subtract the contents of the data memory word addressed by DADDR, using twos complement arithmetic subtract the contents of the data memory word addressed by DADDR.  X X X X X X X X X X X X X X X X X X X	A X X X X X X X X X X X X X X X X X X X	Manual of the Cerry status was initially set remember the result in displayed memory if the Cerry status was initially set Subtract the contents of 16-bit register RW from the 16-bit data word and data memory if the Cerry status was initially set in data memory if the Cerry status was initially set in data memory if the Cerry status was initially set in data memory if the Cerry status was initially set in data memory if the Cerry status was initially set in data memory if the Cerry status was initially set in the set in a sitemate memory if the Cerry status was initially set of the set in the cerry status was initially set in the set in set in the set	DADDR, using twose complement anithmetic. Decrement the result in distinct the secult in distinct the secult in distinct the certy status was initially set  Subtract the contents of 16-bit register RW from the 16-bit data word a fine set of the contents of 16-bit register RW from the 16-bit data word a fine set of the carry status was initially set.  X X X X X X X X X X X X X X X X X X X	The memory if the Cerry status was initially set a Subtract the contents of 16-bit data word the result in diality set a Subtract the contents of 16-bit register RW from the 16-bit data word a Subtract the contents of 16-bit register RW from the 16-bit data word a initially set in the sea of 16-bit register RW from the 16-bit data word a in the Cerry status was initially set in the set of 16-bit register RW initially set in the set of 16-bit register RW initially set in the set of 16-bit register RW initially set in the set of 16-bit register RW in the set of 16-bit register RW in the set of 16-bit register RW in the date memory word addressed by DADDR from the contents of 16-bit register RW, using twos complement arithmetic contents of 16-bit register RW, using twos complement arithmetic subtract the contents of 16-bit register RW, using twos complement arithmetic contents of 16-bit register RW, using twos complement arithmetic subtract the contents of 16-bit register RW, using twos complement arithmetic subtract the contents of 16-bit register RW from the date memory byte a Subtract the contents of 16-bit register RW from the date memory byte a Subtract the contents of 16-bit register RW from the date memory byte a Subtract the contents of 16-bit register RW from the date memory byte a Subtract the contents of 16-bit register RW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte a Subtract RRW from the date memory byte and the Subtract RRW from the date memory and dates RRW from the date memory and dates RRW from the date memory and dates RRW from the	DADDB, using twose complement antitumetic. Decrement the result in distributed to decrement the result in distributed to decrement the result in distributed to decrement the contents of 16-bit register RW from the 16-bit data word to descreed by DADDB, using twose complement arithmetic. Decrement the result in data memory if the Carry status was initially set.  X X X X X X X X X X X X X X X X X X X	dddbbb under the result in discussion of the part of the result in discussion of the result in the r	WRJ-RIGIDS A STATE OF CONTRACT	B88 decomplement influency of the Carry status was initially set.    A

Demotron Performed

Object Code

ø
1
ю

			_											
another 16-bit unsigned data value that is loaded into the CS segment register register [PC] — [EA]  Jump inclinect in current segment. The 16-bit contents of the data memory word addressed by DADDR is loaded into PC										18+E∀⊷	ddd00 f sa 17 [9210][9210]	яаана	9ML	
purde as an 8-bit or 1 6-bit signed binary number, as needed, by the assembler [PC] — DATA16, ICS] — DATA716, ICS] — TAAA16, ICS] — DATA316, ICS] — BRANCH is a label which becomes a 16-bit unraigned data value which is loaded into PC. SEGM is a label which becomes										91	EA PPQQ PPQQ	ввуисн, зеем	qML	Jump
[PC] — [PC] + DISP Jump direct to program memory location identified by label BRANCH. The displacement DISP which must be added to the Program Counter will be com-	L									gı	18010111 [92 0] 92 0	ВВУИСН	dWr	
8R retaigen tid-8 ofni 8ATAQ eryd stab etaibemmi enth baoJ 8tATAQ — [WR] WR retaigen tid-81 ofni 8tATAQ brow stab tid-81 etaibemmi ent baoJ										٠.٢	AAAA PPP L LOL	81ATAQ,WR	NOW	
8ATAQ — [8A]	l					l				.0	AA PPPOLLOL	8ATAQ,88	VOM	Į
ROGAQ vd - (A3) -31ATAQ (A3) -31ATAQ blow eseb itd-81 easib mem erit bed late memory word ROGAQ vd basearbbs										10+E∀	(DISP)(DISP) YYYY	.ROGAG 81ATAG	VOM	Immediate
8ATAQ $\rightarrow$ (A3) beaseabbs eryd vromem stab erir oini 8ATAQ eryd stab stabenmi erit baod eryd yn ar ar ar ar ar ar ar ar ar ar ar ar ar										10+EA	CG 88000PP	,RGGAG 8ATAG	VOM	
bessed by DADAl. Store the result in the addressed data memory part [A3] ROX [WR] $\rightarrow$ [EA] HOX [WR] $\rightarrow$ [EA] HOX [WR] $\rightarrow$ [EX] Exclusive OR the 16-bit contents of register RW with the data memory word states bedressed by DADAL Store after result in the sedressed by AGDAL Store of the result in the sedressed by AGDAL Store of the result in the sedressed by EADAL Store of the result in th	0	x	n	×	×				0	43+9ι	dddsssss 16 [q2l0](q2l0]	WA,800A0	нох	(Me
word addressed by DADDR. Store the result in RW word addressed by DADDR. Store the result in RB with the data memory byte ad- Exclusive OR the 8-bit contents of register RB with the data memory byte ad-	o	x	ր	×	×				0	A3+81	(DISE)[DISE) 30 seceppe	89,900A0	яох	ondary n
dressed by VADDR. Store the result in RB [RW] — [RW] XOR [E.A.] Exclusive OR the 16-bit contents of register RW with the 16-bit data memory	0	x	n	x	x				0	9+EA	33 99ddbbbb [95][Disp]	ROGAG,WR	яох	Memory erate) (C
sphrorpriately (A3) ROX [88] $\rightarrow$ (R8) HOX (R8) $\rightarrow$ RB with the data memory byte advances by the 8-bit contents of register R8 with the data memory byte advanced to the BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Ababa BAD Report of the Abab BA	0	×	n	x	×				0	<b>∀∃+</b> 6	32 sedddbbb [98][01SP]	90 <b>0</b> A0,8Я	яох	Secondary Memory Reference (Memory Operate) (Continued)
[EA] AND (RW) MW. DADGM with AND IRW: AND the 16-bit contents of the date memory word addressed by DADDR with the contents of 16-bit register RW. Discerd the result, but adjust status flags	0	×	n	×	×				0	∀3+6	ddd genes 38 [9810][9810]	WA,AGGAG	TS3T	- 6
<u> </u>	1	T.	1	1_	1.	1.	1	1	_		· — —			-

Table 5-4. A Summary of 8088 and 8088 Instructions (Continued)

	bermothe9 notranego	Г			96	en	1815				Glack Cycles	Object Code	(a)bristeqQ	hiermaniki	18
	POLICE IN LONG MACH	э	ď	٧	Z	1		ij	a	٥	points vorio	enno stelato	Jelmin idria	DILI POLICIO DE LA CALCADA DE	ΨŅ
	[PC] — [EA], [CS] — [EA+2]  Lump indiversints o new segment. The 1 8-bit contents of the data mem word addressed by DADDR is toaded into PC. The next sequential 9-bit demony word's contents is toaded, into the CS segment register.										54+Ε∀	ddd f O fee 44 (4810][4810]	S⊃'HODD#O	4ML	Jump (Cont.)
	FC  — RWI   PROPRIES IN CONTRICT IN CONT										ıı	FF 11100reg	WA	9ML	Jum
	[[SP]] ~ [PC], [SP] ~ [SP] ~2, [PC] ~ [PC] + DISP	L	L		$\perp$	$\perp$	$\perp$		L	L	••61	E8 DISh DISh	BHFACH	כערר	L
	Call a subroutine in the current program segment using direct addressing [[SP]] $\leftarrow$ [CS], [SP] $\leftarrow$ [SP] $\rightarrow$ 2, [[SP]] $\leftarrow$ [PC], [SP] $\leftarrow$ [SP] $\rightarrow$ 2, [PC]										82	DD99 DD99 A6	BSANCH,	CALL	
	[CS] — DATA16 [CS] — DATA6 direct address: Soling direct address:							-							
Ásur	BMANCH and SEGM are labels that become different 16-bit data words: the are loaded into PC and CS, respectively [FP] — [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] — [PC] [PC] [PC] [PC] [PC] [PC] [PC] [PC]										21+E∀**	dddO1Oaa 33	900A0	11¥3	
	reserbbs toentioning misurant progress managed the profit of entitly of the section of the secti											[DISE][DISE]	NOON'S	2740	
[A3	(SE)] ← [CC2] [SE] ← [SS] − 2' ([SE]] ← [EC]' [SE] ← [SE] − 2' [EC] ← [E										37+EA"	FF 8e011bbb	DADDR,CS	C⊁LL	ā
	[CS] — [EA+2]  Call a subroutine in a different program segment using indirect address.											(asio)(asio)			Subroutine Call and Return
	The address of the subroutine called is stored in the 16-bit data mem word address of the schools. The new CS register contents is stored in														e ar
	[2b] ← [bC]*[2b] ← [2b−5]* [bC] ← [bM] uext sedneutjej bto8tem wework word										91	FF 11010re9	WA	CALL	) e
l	Call a subroutine whose address is contained in register RW. [PC] ← [[SP]], [SP] + 2							1			8	C3		T38	brout
	Hetum from a subroutine in the current segment $(SP)$ . $(SP)$ ← $(SP)$ +2					l					12.	CB.	cs	T3A	မှု
	Hetum trom a subroutine in another segment [PC] ← [[SP]], [SP] → [SP] → [SP]					l	1				••L1	C2 YYYY	31ATAO	T3A	
-sib	Between a subrougher in the current segment and the distribution of S of thermostic														
	$[PC] \leftarrow [[SP]]_{+}[SP] \leftarrow [SP] + 2$ , $[CS] \leftarrow [[SP]]_{+}[SP] \rightarrow [SP] + 2 + DATA16$ Hetum from a subroutine in enother segment and act an immediate displa										<b></b> 8ι	CA YYYY	81ATAG,89	138	
	q2 of from					ŀ			l	ĺ					
				1		Т	Т	1	1	ŀ					1

	ı
L	r

bernrotte¶ goldsregO	Н	_	_	<u> </u>		_	_	_	-	Cłock Cycles	eboO tosidO	(s)bnsreqO	DinomenM	툊
	э	ď	٧	z	s	1	ī	a	0					3
SATAG + (JA) (JA)	х	x	x	x	х		L		х	.7	AA #0	8ATAG,JA	00¥	
tateigat JA ant of stab atsebammi fid-8 bbA 8 rATAG + [XA] → [XA]	х	X	×	×	×				х	,7	05 YYYY	91ATAQ,XA	₫₫¥	
reteiger XA ant ot siste etistemmi tid-81 bbA 8ATAC + (89] — [89]			l		l				х	.17	YY bbb00011 08	8ATAQ.BR	aa∀	
retziger 8R ent of stab etsibernmi tid-8 bbA			l					ļ						
	x	x	×	×	x				x		PPP0001118	81ATAG,WR	ad∧	
Tetaiger WR erft of state for the FW register (EA) → (EA) → (AE)	Х	х	×	х	×				x	17+EA	ddd000es 08	DADDR	aa∧	
RODAD yd beesenbbs efyd ynornem steb ertt of steb eteibemmi fid-8 bbA											(DIS6)[DIS6] AA	BATAO		l
3tATAC + [A3] [A3] SOUND of the person o	x	x	x	х	x				x	17+EA	DISP][DISP] YYYY	,RGGAG 81ATAG	aa∀	
AddAd yd besseibba brow ynomen atab edta to the deta memory word seditessed by DADAA (JA) $\rightarrow$ (JA) $\rightarrow$ (JA)	×	×	×	×	X				×	.*	从 <b>排</b>	8ATAG,JA	DŒ∀	
Add 8-bit immediate data, plus carry, to the AL register [AX] $+$ DATA16 $+$ [C]	Н		ı	i					×	•	AAAA GL	81ATAG,XA	DG∀	
Add 16-bit immediate data, plus carry, to the AX register			L	L	l.,	l								
(R8) ← (R8) + DATA8 + [C]	x	х	x	×	x				x		AN PPPOLOIL OB	8ATAG.8	VDC	8
Add 8-bit immediate data, plus carry, to the RB register [RW] + DATA16 + [C]	×	×	ľ	ľ	ľ				Х	-7	PPPOLOIL 18	3 LATAQ,WR	DOA	Operate
Add 16-bit immediate data, plus carry, to the RW register			ĺ.		l						AAAA			وُ
[S] + 8ATAC + [A3] — [A3] where study are specially stribegraph stribegraph and appropriate the stribegraph in the stribegraph and appropriate the stribegraph in the	x	х	×	x	×	İ			x	∀3+71	vv (92)0)(92)(0)	RATAC	oa∀	
bAd 8-bit immediate date, plus carry, to the date memory byte addressed by HODAD	$ \hat{\ } $	l^	l^		ļ^					'3''2'	YY [4810][4810]	8ATAQ 800AQ	500	(m)meguate
$[EA] \leftarrow [EA] + DATA16 + [C]$ Add $16$ -bit immediate dats, plus certy, to the data memory word addressed	,	×	×		×				x	¥3+∠1	(DISP)[DISP] YYYY	.RODAD 91ATAD	DOA	
ROGAQ vd	ľ	^	ľ	^	^	ļ			۳	.*	74 77	8ATAG,JA	QNA	
BATAD GNA [JA] → [JA] S-Lit date dete with AL register contents	ارا	U	ľ	Ļ	l <sub>x</sub>			П	0	_		מעושמיזע	200	l
81ATAQ QNA [XA] → [XA]	0	х	n	×	×				٥	.1	26 YYYY	81ATAG,XA	₫ <i>M</i> A	
stnetnoo reteiger XA rive step estatermin id-8 f GNA	ľ	l^	Ι''	1^	1^				ľ	.7	YY bbboottf 08	BATAQ.8R	_ dwa	
8ATAD DNA (8R) → [8R] Antendrate deter with RP register contents	۱	l,	L	l	×				,	_	I I PPPPPPP	חשושמים ו	ann	l
81ATAG GNA [WR] [WR]	o	x	'n	x	x			1	0	.7		8 ratad,wa	₫N₩	
etnotnoo toteigot WA ritiw etab etabommi tid 8 t GNA 8ATAQ GNA [A3] — [A3]	٥	×	ľ	×	×				0	17+EA	VVVV ddd001aa 08	8,ADDAG	₫N₩	ı
yd besserbbs efyd yromem atab to strefnon drive atab efeibernmi tid-8 QMA			1	١	l			1			YY [98IQ][98IQ]			ı
R00AQ	ľ	^	Ι"	ľ	ľ	ı			ľ	A3+7†	ddd001as 18	,800Aa	₫NA	
at ATAO GNA [A3] → [A3]  -bs brow gromem etsb tid-81 to streams with atse when tid-81 GNA	۱	.,	L	U	Ļ				٥	551.11	YYYY [q2iQ][q2iQ]	81ATAG	- AND	l
AUGAG vd bessenb			1							I				

Table 5-4. A Summary of 8088 and 8088 Instructions (Continued)

рептотье¶ по!инеqО	<u> </u>	٩	٧	٠	S S	1181		ľ		٩	Clock Cycles	Object Code	(s)pustedO	Minemonic	Тyре
[AL] - DATA8 Suptrect 8-bit immediate date from AL register contents. Discard result, but	Н	-	-	+	+	+	1	1	+	×	٠.5	3C AA	8ATAQ,JA	CMP	L
egelt autste reujbe ATATAQ — [XA]	×	×	×	×	×	1	ĺ		- [	×	.7	3D XXXX	BIATAG,XA	CMP	
Subtreet 16-bit immediate date from AX register contents. Discard result, but adjust status flags:						-		1							
[RB] — DATAB Subtract 8-bit immediate data from RB register contents. Discard result, but	x	×	×	x	,	ĸ			:	×	٠.7	AA PPPLLLLL 08	8ATAQ,88	сиь	
ageit autste teuibs	^	<b> </b> ^	^	^	ľ			1		^	.,	1400001	8 LATAQ,WPI	arts	
[RW] - DATA16 Subtract 16-bit immediate data from RW register contents. Discard result,	Ļ	v	Ļ	ļ	Ι,			1	Ι,	×	.0	PPPLLLL	DIWING'AAU	CWI	
sgelf eufsiz fizijbe tud 8ATAG - [A3] baasathha atvd vromam steh to stretono mont steh atsibeoomi int.2 nestdu2	×	×	×	×	þ		-		:	×	10+EA	YY (YY) 80 8e11166 YY (PISP) YY	,800A0 8ATA0	CIVID	
Subtreat 8-bit immediate data from contents of data memory byte addressed Dy DADOR, Discater result, but adjuet statue flags [EA] — DATAT8	×	×	ľ	×	`	1				×	A3+01	r r relatricial	,A00A0	CIMB	<u>.</u>
** Subtrect 16-bit immediate data from contents of 16-bit data memory worn subtreesed by DADDR. Discard result, but adjust status flags		l.,				$\ $					1	[DISE][DISE] 사시[사시] 81 1 1 PPP			Immediate Operate (Continued)
8ATAO 90 (JA) → (JA)	o	х	n	x	>	۲Ì			ŀ	٥	.0	W 30	8ATAG,JA	90	្តិ
sinerinon negister dets with AL register contents $i$ 04-8 RO $i$ 04- $i$ 1	٥	×	١	×	×	4		l		٥	.t	OD AAAA	81ATAQ,XA	80	perat
atnernoo nagiste AXA driw atse sainten mita-81 RO 8ATAO RO [88] —- [88]	٥١	×	n	\×		۲		١	ľ	٥	.7	YY bbb10011 08	8ATAQ,8R	90	ata o
stnetnoo retsiges 88 ithiw stab etsiberimi tid-8 80 81 ATAO RO [WR] [WR]	0	x	n	x	ľ	\		١		٥	.0	PPP10011 18	81ATAG,WR	RÓ	300
etnetnoo reteiger WR ritiw steb etsibernmi bid-81 RO 8 ATAO RO [A3] — [A3]	H				ı	1		l	1	٩	A3+71	80 88001PPP AAAA	,AGGAG	яO	3
bessetbbs elvy thomen steb to strenth out the sets by Byte addressed by RODAC		l			l.			l				YY [4210][4210]	8ATA0		ļ
81ATAO RO [A∃] → [A∃]	0	x	n	×	×	۲		l	ŀ	ا ه	17+EA	ddd100as 18 [9840][9810]	ROGAG BIATAG	ЯO	l
PR 16-bit Immediate data with contants of 16-bit data mamony word ad dresaed by DADDA [A1] — [A1] — [A2]	×	ľ	ľ	*	]'	×		l	ľ	×	•7	1C AA	8ATAQ,JA	888	
Subtract 8-bit immediate signed binary data from AL register contents using the Southub and the Configuration of the Companies of the Companie	U	l	Ļ					l	1				ov i vatan	550	
ilveerint [D] – 81ATAQ – [XA] → [XA]	×	x	х	×	×	4			1	×	.17	YYYY at	81ATAG,XA	888	
Subtrect 16-bit immediste signed binary data from AX register contients using twos complament arithmetic. If the Carry status was originally 1 decre- ment the result															

ъента1•94 поізвтедО	Г	_		50	snı	812	_			Clock Cycles	eboO toeidO	Operand(s)	Mnemonic	ξ
	5	4	₹	z	8	ī	Ī	a	0					E
[RB] — [RB] — DATAR — [C]  Subtract 8-bit immediate signed binary data from RB register contents using  Subtract 8-bit immediate signed binary data from RB register contents using  the Cery status was originally 1 decrement	×	x	x	x	×				х	•Þ	AA PPPILO11 08	8ATA0,89	888	
the result [RW] = DATA16 [C] Subtract 16-bit immediate signed binary data from RW register contents subtract 16-bit immediate signed binary data from RW register contents using twos complement arithmetic. If the Carry status was originally 1 decre-	x	x	×	x	×				×		18000001 18000001	lə f АТАQ,WЯ	985	
ment the result [EA] — DATA8 – [C] $S=100$ Contents of data memory $S=100$ Contents of data memory	x	x	×	×	x				x	17+EA	80 8801 Tobb YY (92)	ROGAG 8ATAG	88S	
Subtract 16-bit immediate signed binary data from contents of 16-bit data	×	×	×	×	×				x	۱۲+EA	rs00000r ddd 1 10ss	,AGÖAG ƏTATAQ	88\$	
memory word addressed by DADA using twos complement entithmetic. If the Carry atsitus was originally 1 decrement the result [AL] — (BAL) — BATAR SALI — (LAI) — (LAI) — (LAI) — Charts immediate signed binary data from AL register contents	×	×	x	x	×				x	.9	(DISP)(DISP)YY (YY)	8ATAG,JA	ens	(Continued)
the memory of the properties of the memory of the state o	x	×	x	×	×				x	- +7	SD AAAA	ê1ATAG,XA	e∩s	
camplement enthmetic (RR) — [RR] — DATA8 Subtract the 8-bit immediate signed binary data from RB register contents	×	x	x	x	×				×	-4	W bbbroiii 08	8ATAQ,8₽	ens	mmediate Operate
using twos complement anthmetic [WR] → [WR] – [WR] – [WR] = DATAT 6 Subtract the 16-bit immediate signed binary data from RW register contents	×	x	x	x	x				x	.₹	AAAA PPPlOIII 18	atad,wa	aus	   
using twos complement sithmetic [EA] — [EA] — DATAB Subtract the 8-bit immediate signed binary data from the contents of the data	x	×	×	×	×				×	A∃+71	dddf0fae 08 YY [q2l0][q2l0]	,ADDAQ 8ATAQ	<b>9∩</b> S	
memory byte addressed by DADA using twos complement arithmetic $B = A + A + A + A + A + A + A + A + A + A$	x	x	×	x	×			5	x	<b>A3</b> +√1	1000001 88101bbb (VY)YY (YY)		ens	
withmestic AMD IAP. Description and AL register contents. Discard the result but a MU IAP. Betting a suit but but a suits a suits a suits but but but but but but but but but but	0	×	n	×	×					۰.7	YY 8A	8ATAQ,JA	TEST	
		Γ				Γ	T	Γ						Γ

Table 5-4. A Summary of 8088 and 8088 Instructions (Continued)

bernrofted notrareq0	ŀ	٦1	٠,	_	800	_	-	T,	ľ	0	Сюск Сусівв	Object Code.	(s)bnsreqQ	ымпетопіс	Š
31ATAC CNA	[XA]	-+	-+	$\overline{}$	x z	-	+-	Ļ	-	0	.7	YYYY 6A	81ATAQ,XA	T83T	L
and the 16-bit immediate data ond XA bagiater contents. Diseast drivers taulib f b diseast starld flags AATAO OVA	1ud (8A)	٥	×	n	x	×				0	₽9	YY bbb0001f 89	8ATAQ,88	TEST	
th 3-bit immediate dats and RB register contents. Discard the result but the staff such stage of the content and the contents the content and	iрв				^	l^				۱	+3	-PP-00011 Z3	ATAN WR	1931	
AND DAFATE. Discend the result to protect to the result of the result indicated the result.	INA	۱	Ţ	_	¥1	Ļ				0	.9	AAAA EV 11000999	atatad,wR	1831	١
t adjust status flags AND DAAB ID the B-bit immediate data and the contents of the data memory location	(EA)	٥	×	n	×	×				0	11+EA	F6 88000bbb YY [920][0 SP] YY	, ROGAG SATAG	TEST	nueaj
sgafur statis satis should the result but salpast affart affacts and ADADA by BATATA GNA 8.1ATAA GNA 9.1ATAA GNA 9	[EA]	٥	x	٥	X	×				0	- <b>∀3</b> +≀≀	[DISP][DISP]YYYY	AGGAG 81ATAG	1831	Immediate Operate  Continued
seel sursts taying the result but adjust status flegs but adjust status flegs $AVAD\ ROX\ [AI]$		٥	×	n	x	×				0	·b	34 44	8ATAQ,JA	яох	opera
arinetines AL register contents 6 FO FO Selection of IATAQ ROX [XA] → 6 FATAQ ROX [XA]	[XA]	0	×	n	x	×				0	.17	36 777	81ATAG,XA	нох	Square
stnetnoo netsigen XA ritiw stsb etsibenmii tid-81 RO evisuic RATAD ROX [BR] — stestnoo session BE 48 etsibeneni std 8 RO evisuic	· [88]	0	×	ņ	x	×		l		0	.7	AA PPPOLILI 08	8ATAG,89	яох	
sheshnor arsiger BR Athwasts immediate asset with TB BO eviaulc	[MA]	0	×	n	x	×		l		0	.7	PPPOLITIES AVYY	aratad,wa	яох	l
— [EA] XOR DATAS	[EV]	۰	×	n	x	×		l		0	17+EA	80 1096 VY (48/U) 44	,AGGAG BATAG	яох	
HODAQ we be be a possible of the 16-bit data memory (A Describer 14-bit data memory into-21 and 16-bit data memory (A Society SO evilable PRO evilable by MODAQ we bessen but a debase by	• [Á3]	0	×	n	x	×				0	A3+√I	640010ee 18 [98](98] 7777	.RGGAG 81ATAG	нох	
← [CX] ¬ 1 II. [CX] ≠ 0 II/ett [bC] ← [bC] + DI2bB		4	4		Ц	L	1	L		_	** ( 10 d	ES DISb	84SIQ	4007	L
crement CX register and branch if CX contents are not 0 $\leftarrow$ [CX] $-1$ If [CX] $\neq$ 0 and [Z] $=$ 1 then [PC] + DISPB	(CX)			i				l			**81 no 8	E1 DISP	84SIQ	34001	1
crement CX register and branch if CX contents is not 0 and Z status is 1 → [CX] —1 If [CX] ≠ 0 and [X] = 0 then [PC] ← [PC] + DISP8	[CX]	İ						١			+•er ₁o ē	EO DIZE	892IO	LOOPNE	COLMINION
crement CX register and branch if CX contents is not 0 and 2 status is 0 a LOOPUE  LOOPE  LOOPE	99S										91 10 4	-d\$IO ₹₹	89210 89210 89210	Z4001 ZN4001	Dialog Cit
. (PC] + DiSP8   PC		-	- 1			-	1	ı	ı		0.10.6	1010.44	2 1014	. AL	١.

4	
Ļ	
Ġ	

Second   Control   Contr	Display   Disp	Operation Performed (s) Object Code Cycles Statuses	nomen M	8
1	Second   Cold			ء ا
Second   Fig.   Second   Sec	Second Head   Second Head		3AL	L
Second   S	Beach   10d   10			
18   18   18   18   18   18   18   18	18   18   18   18   18   18   18   18		ar i	
1   10   10   10   10   10   10   10	1		- 1	
18   18   18   18   18   18   18   18	1   1   1   1   1   1   1   1   1   1		38r	l
Search   1   1   1   1   1   1   1   1   1	Display			
100   100	10   10   10   10   10   10   10   10		CXX	
19   19   19   19   19   19   19   19	## Second Residue Second Remark it 2 is 7 to 1879  ## Second Remark it 2 is 0 or the 5 and 0 eletuees are the same and the	O si salos XXX et 1 i d'acanch if the CX register contents is 0	ı	l
### Proof   1998   1908	### Opicial Proof   1998   1999   199		3r	
100 DISPR 7 Or 16 S and 0 statuses are the same ame ame ame ame ame ame ame ame ame	1912   1912   1913		- ]	
10 DISPR 7 OF 16.1  10 DISPR 7 OF 16.1  11 DISPR 7 OF 16.1  12 DISPR 7 OF 16.1  13 DISPR 7 OF 16.1  14 DISPR 7 OF 16.1  15 DISPR 7 OF 16.1  16 DISPR 7 OF 16.1  17 DISPR 7 OF 16.1  18 DISPR 7 OF 16.1  18 DISPR 7 OF 16.1  19 DISPR 7 OF 16.1  19 DISPR 7 OF 16.1  19 DISPR 7 OF 16.1  19 DISPR 7 OF 16.1  19 DISPR 7 OF 16.1  20 DIS	100 DISPR 7 OF 16 Property of 16 Pro		26	l
Part   Part	### Sping of the Same of the Same of Spinguages are the same  #### Spinguages of the Same of Spinguages are the Same of Spinguages are the same of Spinguages of Spinguage		301	
Post   Post	## Single Picks   10-10-10-10-10-10-10-10-10-10-10-10-10-1		JDC	
1	### Part of Part of Statuses differ the S and O statuses differ the S and O statuses differ the S and O statuses differ the S and O statuses differ the S and O statuses differ the S and O statuses differ the S and O status and O status and O status and O status and O status and O status and O status and O status and O s and O status and O s and O status and O s and O status and O s and O status and O s		J. 1	9
Dispers	See JGE   See			ē
Dispers	See JGE   See		370	907
Dispers	See JGE   See			S
Dispers	See JGE   See		ANL	٥
Dispers	See JGE   See		3AML	i
Dispers	Disps   4 or 16"   Breach if 9 is 1		8Nr	3
Dispers	See JGE   See		BBNC S	2
Dispers	See JGE   See		anc 9	Ş
Dispers	See JGE   See		Ş.,	9
Dispers	See JGE   See		DNC S	a
Blease   B	See JG   S			l a
Disparage   70 Disparage   4 or 16".	100 DISP8 70 DISP 4 or 16". Branch if Pic) + DISP8  100 DISP8 70 DISP 4 or 16". Branch if Dis 0  100 DISP8 70 DISP 4 or 16". Branch if Dis 0  100 DISP8 70 DISP 4 or 16". Branch if Dis 0  100 DISP8 70 DISP8 4 or 16". Branch if Dis 0  100 DISP8 70 DISP8 70 TISP8  100 DISP8 70 DISP8 70 TISP8  100 DISP8 70 TI			
Branch if P is 0   Branch if P is 1	Branch if O is O is O is O is O is O is O is O			
Brench if P is 1   Brench if P is 1   Brench if P is 1	Jun Dispe 78 Dispe 4 or 16".  Jun Dispe 79 Dispe 4 or 16".  Jun Dispe 4 or 16".  See Jul Branch if 9 is 0  See Jul Branch if 9 is 0  See Jul Branch if 9 is 0  See Jul Branch if 9 is 0  See Jul Branch if 9 is 0  See Jul Branch if 9 is 0  See Jul Branch if 9 is 1  See Jul Branch			
Branch if P is 0   Branch if P is 0   Branch if P is 0   Branch if P is 0   Branch if P is 0   Branch if D is 7   Branch if D is 7   Branch if D is 7   Branch if D is 7   Branch if D is 7   Branch if D is 7   Branch if D is 7   Branch if P is 1   Branch if P	MSP8 79 DISP8 4 or 16**  DISP8 70 DISP8 4 or 16**  DISP8 70 DISP8 4 or 16**  AND DISP8 4 or 16**  Branch if P is 0  Branch if P is 0  Branch if P is 0  Branch if P is 0  Branch if P is 0  Branch if P is 0  Branch if P is 0		dNL.	
Branch if 9 is 0   Branch if 9 is 0   Branch if 9 is 0   Branch if 9 is 1   Branch if 0	O 12 PB 8 Sench if P is 0 O O O O O O O O O O O O O O O O O O			
Brench if P is 1   Brench if P is 1   Brench if P is 1   Brench if D is 1   Brench if D is 1   Brench if D is 1   Brench if D is 1   Brench if D is 1   Brench if D is 1   Brench if D is 2   Brench if D is 3   Brench if	DISP8	2 DISD8 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	SNC	
OISP8 70 DISP8 4 or 16**   PC] — [PC] — [PC] + DISP8  Branch if P is 1  Branch if P is 1  Branch if P is 1	U DISP8 70 DISP8 4 or 16"   PC] ← [PC] ← [PC] + DISP8  Branch if 0 is 1  Branch if P is 1  Branch if P is 1			
t a0 G ranch if O is 7 A DISP8 4 or 16**   PC] ← [PC] + DISP8   Branch if P is 1	1 Branch if O is 1 Branch if Pis 1 Branch if Pis 1 Branch if Pis 1			
Th DISP8 7A DISP 4.0r 16**   Branch if P is 1	19 OISP8 7A DISP 4 or 16**   Branch if Pis] + DISP8		OF	
r ei 9 li rhansa	Branch if P is 1		aı	
			JC	
41 998 1 [ ] [ ] [ ] 1   1   1   3480   1 340   1 3			∌ar′	

Table 5-4. A Summary of 8088 and 8088 Instructions (Continued)

from A rea . From rea				36	sn	ete				1		(-,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		8
Operation Performed	9	đ	٧	Ż	S	1	Ī	a	0	Clock Cycles	oboD taeidO	(s) pusted (s)	SIUOWAVIA	Түрө
9/NL ee2 898/G + (291 → (291			L							gt x0.ft	4SIG8T	84SIQ DISP8	O-II	(Cont.)
[bC] ← [bC] + DiSb8	]						1			gi 10 t	TOUOT			Ĉ
3f <del>8</del> 9\$			Γ	T	T	T	Γ	Г	Г			DISP8	Ζſ	800
(ABD) [ABS] Herister 16 any 88 maris of a										ۍ.	asabbb f f A8	289,089	VOM	
Move the contents of any RB register to any RB register [RWD] $\leftarrow$ [RWD]	1									٠.٢	sssppp11 88	SW9,GW9	VOM	•
Move the contents of any RW register to any RW register $[SR] \leftarrow [RWS]$										۲۰	82811Û [ 38	WAR	VOM	Register Register Move
Move the contents of any RW register to any Segment register														ate.
[RWD] [SR] how Segment register to any RW register	-			l			l			.z	BC 110mddd	AS,WA	VOM	Regi
[WA] ← → [XA]	1									3.	1001006	WR,XA	хсне	ě.
Exchange the contents of AX and any RW register (RB) $\leftarrow$ $\rightarrow$ (RB)	l									•₽	gengenii 88	ลя,ลя	хсне	egi et
Exchange the contents of any two RB registers [RW] — — [RW]	Ĺ		l		1		ı	ĺ		ا ۳۰	gengent! 78	WA,WA	энэх	,
Exchange the contents of any two RW registers											Ba Garri			
$[lg]] \rightarrow [lg] \rightarrow [lg] \rightarrow [lg] \rightarrow [lg] \rightarrow [lg] \rightarrow [lg] \rightarrow [lg]$	×	×	×	x	×	1	L	Q/I	×	22	94	\$8'08	CMPS	
Compare the data bytes addressed by the SI and DI Index registers using string data addressing*	ı						1						·	
$z \pm [iq] \rightarrow [iq] + [is] \rightarrow [is] \rightarrow [is] / [iq] - [is]$	×	×	×	×	×		l	Q/I	х	22	. TA	SW, GW	CMPS	! !
Compare the 16-bit data words addressed by the Si and Di Index registers using string data addressing*	ı		l		1		l							1
1 ∓ [IS] → [IS] [[IS]] → [TV]	ı						l	ķ/I		12	⊃∀	\$8'G8	Sacı	호
Move a data byte from the location addressed by the SI Index register to the BL register is in a state and the second party of		ĺ		1			1	ı						Block Transfer and Search
i ∓ ( S  →  S ); ( S ) → (XV)	١				ľ		ı	la/I		71	₫¥	MD'M2	\$601	2
Move a date word from the 16-bit location addressed by the SI Index register.  The AX register using string date addressing	1	-					l							5
$\mathbf{l} + [IGI] \rightarrow [IGI] + \mathbf{l} + [ISI] \rightarrow [ISI] \rightarrow [IGI]$	١.		-				l	Q/I		81	₽∀	\$8'08	SVOM	Į₫Į
Move a data byte from the locations addressed by the SI Index register to the extre segment location addressed by the DI register rising string data address-	Į.	ļ			ŀ	ı								얁
•Bui	1						ı						3,1011	
[[01]] $\rightarrow$ [[81]], [81] $\rightarrow$ [	1		l					a/ı		81	SA	SW, GW	SAOW	
the extra segment location addressed by the DI index register using string	ı		l	Ĺ		П	l	ĺ				•		11
*gniaeerbbs steb	l		Ì			ı	l	1						
* For these instructions, the default destination segment register cannot be overtiden.	ľ	1	1	1	1	1	1		1			1	1	1

ř	Ξ,	
•	•	

bermohet notinedO	3	d	٧		-	818 T	'''	3 10	믝:	Clock Cycles	eboO toeidO	(s)bna•eqO	ainamenM	Type
Repeat the next sequential instruction (which must be a Block Transfer and		_	Н	L	_	Н	, (	-	_	+5 ber (cop	21001111	N	93 <i>P</i> I	⊢
Search instruction) until CX contents decrements to 0. Decrement CX con-	'						-							٩
tents on each repeat, if the next instruction is CMPB, CMPW, SCAB, or						Ιl		1	1					(Continued)
SCAW then repeat until CX contents decrements to 0 or 2 status does not equal N			ı					1	1					ĮŽ
r ± [[D]], [D(] ← [DI] ± 1	×	x	x	X.	х		c	1/10	x	15	∃A	S8'08	SCAS	
Compere AL register contents with the extra segment data byte addressed by the DI Index register using string data addressing									1					Transfer and Search
(AX) ~ [(DI)], (DI) ← [DI) ± 2	х	X	х	X	Х	Н	c	7HZ	×	91	∃₩	WD,WS	SCAS	Š
Compare AX register contents with the extra segment 16-bit data ord ad-		İ			l		- [	1						ġ
dressed by the DI Index register using string data addressing $\Gamma = \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot \Gamma \cdot $	×	x	X	х	ľ		١	1/1	^	11	AA	S8'08	SOTS	] ē
Store the AL register contents in the extra segment data memory byte ad-	IJΙ	J	Į, l	v	l,			ال.	ĮΨ		u.	00'00	2010	3
gnisserbbe etsb gnirte gnisu retsiger xebni 10 eft yd beseerb						Ιl	- [							🖺
S = [0] - [AX] - [0]	X:	x	x	х	x	l	c	/	x	11	8.4	SW, GW	SOTE	Black
Store the AX register contents in the extra segment 16-bit data memory word budgesed by the DI Index register using string data addressing		ı				ll	- [		1	ļ				L
	٣	-	$\stackrel{\circ}{+}$	_	Ĥ	Н	+	+	+			555 444	557	┝╌
[RBD] — [RBD] + [RBS] + [C] Add the B-bit contents of register RBS, plus the B-bit contents of register RBS.	$ \mathbf{x} $	×	χ	x	l,	H	1	Ι,	×۱	.ε	seapppll ZI	287,087	oa∀	ı
[BMD] ← [BMD] + [BMS] + [C]	x	X	x	X	×	Ιl	- [	1	×	ε	13 lldddsss	RWD,RWS	⊃aA	ı
Add the 16-bit contents of register RWS, plus the Cerry status, to register						l	- 1	1	1	Į				ı
OWR [CBR] + [CBR] + [CBR]	×	×	X	X	ľ	]	-	ľ	×۱	3.	sssbbb11 20	289,089	₫₫¥	ı
GBR netsigen of SBR netsigen to athering fid-8 ent bbA					l	Н		Ι.			·			ľ
[UMD] → [UMD] + [UMD]	×	x	×	x	×	] ]	1	1:	×	3.	03 11 dddsss	SWB, GWR	ad∧	ă
OWR letsiger of SWR referents of the bld for and bbd bdd [den] → [den]	0	x	n	X	×		-	- [ '	۱ ۵	.ε	ssabbb [	\$89,089	QNA	윷
GBR retaiger Afriw GBR netaiger to extremo bid-8 ant QMA					l				₋│	_				ğ
[SWR] UNA [GWR] → IGWR]	0	x	n	х	×	Ιl	-	- [	۰	3.	23 11 dddses	SWF,GWF	<b>GNA</b>	Register Operate
DAM 16-bit contents of register RWS with register RWD  [ [ΛΔ] → [HA]	1				١.	l		1	ı	۶.	86		CBM	
HA offit jid ngis JA brieftx3						Ιİ								Register -
(288) – (089)	x	х	х	х	x			- [:	x l	.8.	seebbbil AE	289,089	смь	ŝ.
Subtract the contents of register RBD from register RBS. Discerd the result, but adjust status flags	۱ ا				l	П		ı						۱_
[BMD] - [BMS]	×	X	х	X	×	Ιſ		1	×	.ε	3B 11dddsss	RWD,RWS	CWE	ı
Subtract the contents of register RWD from register RWS. Discard the result,					l			ı						ı
spelt status flegs [atxa] → [vol	١,	.				П		1		3	66.		GWO	ı
[dtxA] → [x0].				•			- 1	- 1	- 1	9	20	I	CWD	•

Table 5-4. A Summary of 8086 and 8088 Instructions (Continued)

hamushad anitoranft	Г			84	en	181	s				Ohio 1 topidi)	(-)beeneage	~;somes M	8
Demotred notrared	э	d	٧	z	9	ŀ	ı Ti	a	0	Clock Cycles	Object Code	(s)pusied(s)	pinomenia	Туре
[S8R]/(XA] → [XA]	n	n,	n	ln	ľ	1	$\perp$		n	06-08	FG 11110355	SBF	VIG	
Divide the 16-bit contents of AX by the 8-bit contents of RBS. Store the in			1	l		1	1							
teger quotient in AL and the remainder in AH. If the quotient is greater that		ı		1		1	1	1	1					l
FF <sub>T</sub> 6, execute a "divide by O" interrupt [DX] (AX] ← [DX] (BX]/[RWS]	u	l٩	۱''	m	ľ	'	1		n	291-441	28801111 TH	SWA	NIG	
Divide the 32-bit contents of registers DX (high-order) and AX (low-order) b	Ĺ	1	Ļ	٦	١,	Т	1	ı						
en ent bins XA ni trientoup regetni ent erot? SWR to atrientoo tid-81 ent		ı	J	1	1	ŀ	1	1	1					
mainder in DX. If the quotient is greater than FFFF <sub>16</sub> , execute a "divide by 0"	l	ı	1	1			1		1					ı
tqurnetni	1	ı		1	ŀ	Ι	1	ı						
[S89]/(XA) → (XA)	n	n	'n	'n	ħ	ı	1	1	n	211-101	F6 11111 35s	888	VIGI	
Divide the 16-bit contents of register AX by the 8-bit contents of RBS, treat	l	ı	1	Ł	1	ı	1	ı	1					ľ
ing both contents as signed binary numbers. Store the quotient, as a signed binary number is		ı		1	ļ	1	1	1	1					ı
binary number, in AL. Store the remainder, as an unsigned binary number, in the guorier, in LA. If the guorier	ı	ı		1			1		1					*
is greater than 7F16, or less than -8016, execute a "divide by O" interrup	l	ı		1	1		1		1					Register - Register Operate (Continued)
$[SWR]/[XA] [XQ] \rightarrow [XA] [XQ]$	n	n	U	lu	In	١	1	ı	lu.	181-391	F7 1111885	SWA	IDIA	I ∰
Divide the 32-bit contents of register DX (high-order) and AX (low-order) by		1		1					1					Š.
the 16-bit contents of RWS. Treat both contents as signed binary numbers		ı		1			1	1	1					8
Store the quotient, as a signed binary number, in AX. Store the remainder, as		ı		1			1		1	·			]	3
an unsigned binary number, in AH. If the quotient is greater than 7FFF 16, o	ı	ı		1		1		1	1			·		유
less than -800016, execute a "divide by O" interrupt	ľ	Ι	ł''	Ι	Τ.	1		Ì.	<b> </b> ^	00 00				Ę
$[AX] \leftarrow [AL] \cdot [ARS]$	x	ŀ'n	l٦	l۵	l٦	d	1	1	x	86-08	F6 11101585	\$88	INNE	탈
Multiply the 8-bit contents of register AL by the contents of RBS. Treat both		l	i	ı		1	1	1	1					3
XA in toubors as signed binary numbers. Store the 1-5-bit product in XA (IXA) → [XA] × (IXA) • (IXA)	×	n	111	L	1"	Ί.	1	1	×	128-164	#82 TOT IT TH	SWR	IMINE	î
Multiply the 16-bit contents of register AX by the 16-bit contents of RWS	١.,	٦	1_	L	١,		Ι.	1	L					8
Treat both numbers as signed binary numbers. Store the 32-bit product in O)	ı	ı	ł		Ł	İ	1	l	Į			ļ		- gi
(brow rebto-wol) XA bns (brow rebto-dgid)	l	l	ı		ı	1	1	ļ	1					
[S89] • [JA] → [XA]	×	U	l٩	ļn	١'n	1	1	1	X	71-0L	F6 11 100ses	SSA	JUM	
Multiply the 8-bit contents of register AL by the contents of RBS. Treat both	•	l	l		1	١	Т	1	1					
numbers as unsigned binary numbers. Store the 16-bit product in AX		l	1			ı	Т	1	1					
[SWA] • [XA] → [XA] [Xd]	×	n.	ln	l٦	ŀ۵	ı	1	1	lχ	118-133	ses00111 74	SWR	MUL	
Multiply the 16-bit contents of register AX by the 16-bit contents of RWS	ı	l	l	l	ŀ	ı	1	ļ	1					
Treat both numbers as unsigned binary numbers. Store the 32-bit product in	1	l	1		L	1	1	1	1				1	ĺ
XG (high-order word) and AX (low-order word)  (299) — (1999)	ľ.	^	l''	۱^	ገ^	1			I٥	••	agabbbit AO	200 000	- au	ı
[BRR] → [GBR] → [GBR]	١,	ĺv	ľ	I٧	٦		1		6	3.	asabbbii AO	\$89,089	80	ı
. OR the 8-bit contents of register RBS with register RBD (RWD) ← [RWD] OR [RWS]	О	×	l'n	١×	Ι×	4	-	L	٥	3.	seabbbir 80	RWD,RWS	80	1
		1	I .	I	н.	. 1	- 1		1 .	1				

œ
-
•
10

Continued of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement of the series of the complement	bemiotre9 noitsieg0	ш	_	_	_	_	٠.,	•	_	ч	Clock Cycles	Object Code	(a) bnereqO	Mnemonic	Į₹
Bass Gubrand Harmann, and Sak Harbeller (1 the Carry status was objected the result in		9	đ	٧	z	8	Ī	Ţ	a	0					J
The firm of the conversion of		×	×	×	×	×	1			×	.ε	essbbbff Af	288,088	888	
AAA  A	anthmetic. If the Carry status was originally 1 decrement the result [RWD] — [RWD] — [RWS] — [C]	x	×	×	×	×				x	3-	sssppp     gl	SWA,0WA	888	ĕ
AAA  A	ment stithmetic. If the Carry status was originally 1 decrement the result	×	x	×	×	ľ				х	•€	sssbbbii AS	28 <b>4,08</b> A	ans	mtinue
AAA  A	Subtract the 8-bit contents of register RBS from RBD using twos complement					ĺ.,									(C)
AAA  A	[GWR] → [GWR]	x	x	x	х	×				x	3.	SE 11dddsss	SWA, GWA	aus	Opera
AAA  A	nent sittematins inem	٥	X	n	×	×				0	.E	ganganii 48	\$89,089	1831	jister'
AAA  A	AND the 8-bit contents of register d and register RBS. Discard the result, but	l			l	l									3
AAA  A	[SWA] DNA (DWA)	o	x	n	x	x				o	3.	gengenii 68	SWA, GWA	1831	gister -
AAA  A		0	x	IJ	X	×				0	3.	39 11 dddses	\$89,089	яох	Į,
AAA  A		٥	x	n	×	×	1			0	3.	31 11dddass	SW6,0W9	яох	
DAA  MAA  MAA  MAA  MAA  MAA  MAA  MAA		Ĥ	<u>'''</u>	î	Ï	Ë	t	┞	Н	Η		26			┝
MAA  MAA  MAA  MAA  MAA  MAA  MAA  MAA		Ų	n	Ļ	_	  n			l			16			
AAA  AAA  AAA  BAA  AAA  BAA  AAA  AAA	to generate an unpacked decimal quotient. (See accompanying text for	'n	x	n	×	×				n	09	∀0 <b>⊆</b> 0		ανν	l
DEC RB FE 11001ddd 3* X X X X X X X X X X X X X X X X X X		٥	×	'n	×	×		l		n	83	A0 40		MAA	•
DEC RB FE 11001ddd 3** X X X X X X X X X X X X X X X X X X		ľ	П	ľ	ľ	ľ				l <sup>n</sup>	.7	3E		SAA	į
DEC RB FE 11001ddd 3** X X X X X X X X X X X X X X X X X X				l	L	L		1							ۄۣٙ
DEC RB FE 11001ddd 3** X X X X X X X X X X X X X X X X X X		×	×	×	×	×				n	٠,4	72		AAG	1
DEC RB FE 110014dd 3* X X X X X X X X X X X X X X X X X X	is a packed decimel number. {See accompanying text for details}	l				ı									3
88 retgiger of register 8 -bit contents of register 8		Ļ	Ų	Ļ	l,	l,				ا؞ا		.17		CHG	
			x	×	×	×				x	3.	EE J JOOJ 999	87	DEC	
	[HW] → [RW] → I	1	x	×	×	×		ı		×	۶.	DDD10010	WA	DEC	l

3 Instructions (Continued)	380	18 F	pue	_		_	_	еши	ung	A 4-6 eldeT	L	<u> </u>		L
bermorheq notineqO	٥	a.	٧	Z	_	T T	<u>                                     </u>	a	0	Clock Cycles	eboO toeidO	(e)bnateqO	pinomen M	ŀ
r+ (88) → (88)	Щ	х	×.	-	-	+	L	_	Х	3.	FE 11000ddd	88	INC	Ļ
Increment the 8-bit contents of register RB   HR   HR   +1		х	ľ×	×	ľ×				×	٥.	PPP00010	WA	INC	ĺ
WR reteigen to atnetnop tid-81 ant tnemenont-			l	l	1	1	l							ı
f + [88] → [88] 98 parameters to stock on tid-8 add transferror sow.	x	χ	lχ	Ιx	l <sub>x</sub>			П	x	3.	PPPLLOLL 94	. 88	DEN	ı
Twos complement the 8-bit contents of register RB $\Gamma$ + $\Gamma$ WM] $\rightarrow$ $\Gamma$	x	x	×	x	×				×	3•	F7 11011444	WA	NEG	l
WR iegister N = 16-bit contents of register RM   N = 187] → (BR)								1		3.	Pbboror 84	87	TON	۱
Ones complement the 8-bit contents of register RB				l							DDD01011 C1	au.		۱
[WA] → [WA]				l						3.	PPPOLOLI ZE	WA	TON	
Ones complement the 16-bit contents of register RW Rotate lett through Carry the 8-bit contents of RB register, or the 16-bit	x		1		1				×		PPPOLOLI O^OOLOLI	N'88	BCF	۱
contents of RNV register, as illustrated for memory operate	X								X		PPPOLOLI LAGGLOLI	N,WA	HCL	۱
Rotate right through Cerry the 8-bit contents of RB register, or the 16-bit	X			l	1	1			×		bbbrrorr ovgorore	NBA,	หวม	
	x		l	l		1			х		bberrorr ivoororr	N,WA	มวย	۱
Rotate left the 8-bit contents of RB register, or the 16-bit contents of RW	х		-	l	1	1	1		x	.Z l=N	PPP00011 0^001011	N'8H	ТОН	۱
register as illustrated for memory operate	x		ĺ	l					×	8+N+ L <n< td=""><td>PPPOOOLL IAGOROLI</td><td>N,WA</td><td>TOB</td><td>۱</td></n<>	PPPOOOLL IAGOROLI	N,WA	TOB	۱
WR to stream or tid-81 are the register, or the 16-bit contents of the WR.	x					1			Ŷ		PPPLOOFF 0400FOFF	N,8A	ROR	۱
will be stratuo, and all earlies of BB is stratuo; side and the left the WR to stratuo; side and the stratuc; side and the stratuc;	Ş	^	000	l^	×	1	1		×		bbbrootti 0-001011	N,WA N RR	ROR	۱
Shift left the 8-bit contents of RB register, or the 16-bit contents of RW register, as illustrated for memory operate	X	X	Ľ	ľ	ľ				×		PPPOOLLL I^OOLOLU	N,BPI N,WPR	T∀S T∀S	ĺ
Shift right the 8-bit contents of register RB, or the 16-bit contents of register	X	X	ņ	×	l.X		ı	١.	X		PPPCALLUL OAGOLOLI	N,8A	RAS	l
RW, as illustrated for memory operate	Х	×	J.	×	×	1	l	L	×		bbbittiff tvoorotf	N,WR	AAS	l
TAS 998	×	×	n	×	į×	1	L	L	×			N,8A	THS	ı
JAS 998	×	×	n	x	ĺχ	1	ı	ı	x			N'AAH	THS	ı
	x	x	'n	١ŵ	lû	1	ı	П	l×	.Z I=N	PPP10111 0^001011	N,8A	BHS	ı
MW, as illustrated for memory operate	$\overline{}$	V	'n	\_	₩	+	✝	Н	┥		PPPLOLIL IAOOLOLI	N,WA	SHS	t
[EA] ← [[SP]], [SP] ← [SP] + 2. Load the 16-bit Stack word, addressed using Stack addressing, into the 16-			ı		i	1	ı	ŀ		17+EA	8F 88000bbb [DISP][DISP]	PADDR	404	ı
2 yd 98 triemeroni. SIOAO yd besserbab brow yromen steb fid			ı		1	1	ı	1	1				1	ı
S + [48] → [48]] ([48]) → [48] → [48]			l	1	Ł	1	1	1	ł	8	PPP11010	WFI	40a	ı
Load the 16-bit Stack word, addressed using Stack addressing, into the			١	١	1	1	1	1	l	8	1117000	88	909	ı
specified 16-bit register. Increment SP by 2.					1	1	1			_				ı
(SEH) — [(SP)] (SP) — [SP] + 2	×	×	l×	١,	ŀ×	l,	Į,	$ _{\mathbf{x}}$	×	ંક	- 06		ad0d	Į
Load the 16-bit Stack word, addressed using Stack addressing, into the Status Flags register			1	1	ı	ı	L	l						ı
$[SP] \leftarrow [SP] - 2$ , $[SP] \rightarrow [EA]$		Ш	ı	1	ı	ı	1	1		43+91	dddOllee 33	RODAG	H\$Nd	l
		li	1	ı	1	1	1	Į			[4810][4810]			l

·	l			80	RUI	atS						I	l .	L "_
bemrohre4 naitereqQ	э	d	٧	z	_	•	ľ	a	0	Clock Cycles	eboO toe(dO	(s)bns•eqO	oinomenM	Түрө
[SP] — [SP] $-2$ , [(SP)] — [RW or SR] Store the contents of the specified 16-bit register in the 16-bit Stack word addressed using Stack addressing. Decrement SP by 2 [SP] — [SP]] — [SPR] — [SPR] — [SPR] — [SPR] when some sing stack word addressed using Stack addressing. Decrement SP by $\Sigma$										01 11	01010 011 <del>11</del> 000	ЭС 2Ш ВМ	H2∪9 H2∪9 ∓H2U9	Stack (Cont.)
Execute a software interrupt and vector through table entry V Execute a software interrupt and vector through table entry Ne O status is 1, execute a software interrupt and vector through table entry 10 $\pm$						0 0 0	0 0	ı		52 F3 4 Of 53	CE CD AA CC	£ A	INT INT INTO INET	Interrupts
[C] ← 0 Clear Carry status	٥	_	L	L	L	L	L	L		٦.	81		сгс	L
D Clear Decrement/Increment select								0		٠.٢	D4		сгв	
(i) — 0 Clea <u>r in</u> terrupt enable status, disabling all interrupts							0			٠.	A4		CIL	
[C] ← [C] Complement Carry status	x									3,	. 64		смс	
ewollot as register HA of again telenset										.7	46		. ∃HÆ⊐	
2 Z O V O D I C S V B BK LICO														,
Transfar MH register contents to status flags as follows:	×	×	x	×	×					٠,٠	36		∃H <b>A</b> \$	Status
S Z P P C Cegister														
Set Carry etatus to 1	ı						ı			.z	64		STC	
$\uparrow \to \uparrow \to \uparrow \to \downarrow $ Set Decrement's ratue to $\uparrow$					l	ĺ		ι		.z	GH		are	
$t \sim 0.0$							ı			٦.	84		iT2	
Endational da Remanda (1 de entera oragio admissar voc			l	ı	1		ı			1 1				1

(beunitno2) anottourianl 8808 and 8808 to gramma? A. A-& sldsT.

Performed ?— [EA]  7 — [EA]  The contents of the data memory location addressed by DADDR is read out of a memory and placed on the data bus; however, it is not input to the CPU Halt CPU Halt Gustantee the CPU bus control during execution of the next sequential innertuction accesses the segment register SR. See Table 20-1 for allowed memory reference instruction accesses the segment register. SR. See Table 20-1 for allowed memory ment identified by Segment register SR. See Table 20-1 for allowed memory reference instructions.  CPU entstructions  The restruction of the segment register SR. See Table 20-1 for allowed memory reference instructions.	L	d	4	. 2	Z S	3 .		a	0	3. 3+2u 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5.	Object Code  11011xxx abaxxxbbb Ft Ft F0 O01reg110SP] 98	#ddadd #gdadd	HET LOCK SEG	Other Type