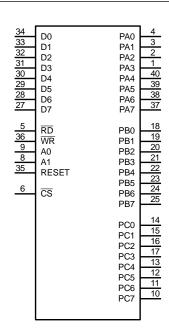
8255 Adresleme											
A_1	A_0	Port									
0	0	PortA									
0	1	PortB									
1	0	PortC, Status									
1	1	Kontrol									

82	8255 Bit Set Reset Mod Kontrol Yazmacı													
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0														
0 X X X B ₂ B ₁ B ₀ S/R														
Port C'nin $(B_2B_1B_0)_2$ numaralı pinini seç.														
S/R: 9	Seçiler	n pind	e Set (1) vey	a Rese	et (0) d	oluştur.							

8	8255 Basit I/O (Mod 0) Kontrol Yazmacı													
D_7	D_6	D_1	D_0											
1	0	0	PA	PC_U	0	PB	PC_L							
PA	PB													
0	0													
1	1													



8	255 N	/lod	1 (G	rup	A) I	Kon	trol Y	azma	CI
D_7	D_6	$D_{!}$	5 D	4	D	3	D_2	D_1	D_0
1	0	1	P	Α	PC.	6,7	X	X	X
PA	PC ₆	5,7	Port	Υċ	inü				
0	0		Ç	ıkış	}				
1	1		G	iriş	}				

 D_6

1/0 1/0

 $\overline{OBF_A}$

 D_5

 $\overline{IBF_A}$

 $INTE_A$

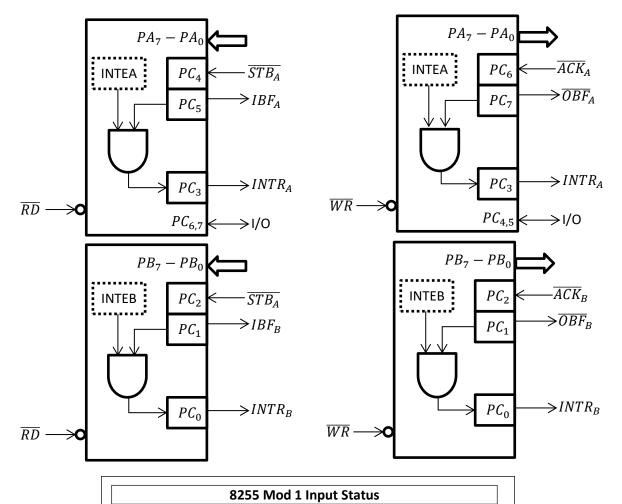
 D_4

 $INTE_A$

1/0

 D_4

82	8255 Mod 1 (Grup B) Kontrol Yazmacı													
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0							
1	X	X	X	X	1	PB	X							
PB	Port	Yönü	i											
0	Ç	ıkış												
1	G	iriş												



 D_3

 $INTR_A$

8255 Mod 1 Output Status

 D_3

I/O $INTR_A$ $INTE_B$

 D_2

 $INTE_{B}$

 D_1

 \overline{OBF}_{B}

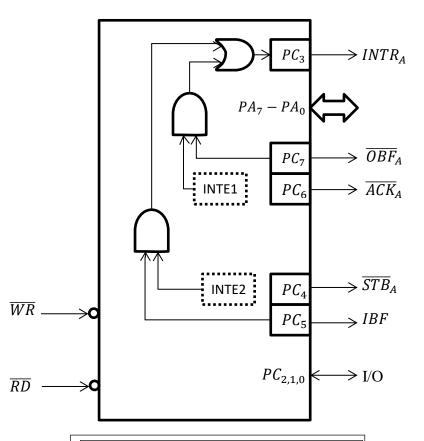
 D_0

 D_0

 $\overline{INTR_B}$

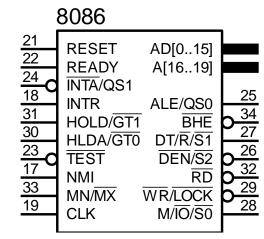
 IBF_B $INTR_B$

8086 Flags															
D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Χ	Х	Х	Х	0	D	I	Т	S	Z	Х	Α	Х	Р	Х	С



8	8255 Mod 2 (Grup A) Kontrol Yazmacı													
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0							
1	1	X	X	X	X	X	$PC_{2,1,0}$							
PC_2	,1,0	Port '	Yönü]										
0		Çıl	κιş											
1		Gi	riş											

		8255	5 Mod 2 S	tatus			
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
\overline{OBF}_{A}	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	Х	Х	Х



	8251 Adresleme												
C/\overline{D}	\overline{RD}	\overline{WR}	Yazmaç										
0	0	1	Data → μP										
0	1	0	<i>μP</i> → Data										
1	0	1	Status → μP										
1	1	0	$\mu P \rightarrow Mode$, Control, Sync										

	8251 Mod Yazmacı (Senkron)												
D_7	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0												
SCS	SCS ESD EP PEN L_2 L_1 0 0												

SCS: Sync karakter sayısı. 0: 2 sync, 1: 1 sync

ESD: External sync detect. 0: SYNDET output, 1: SYNDET input.

	8251 Kontrol Yazmacı												
D_7	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0												
EΗ	EH IR RTS ER SBRK RXE DTR TXE												

IR: Internal reset. ER: Clear error bits. SBRK: Break transmit, forcing TxD low.

	8251 Status Yazmacı											
D_7	$egin{array}{ c c c c c c c c c c c c c c c c c c c$											
DSR	DSR SYNDET FE OE PE TxE RxRDY TxRDY											

D[0..7] RESET CLK

C/D WR RD

DTR DSR RTS CTS 15 18

9

TxRDY

TxC

RxD

 \overline{RxC}

RxRDY

SYNDET

TxEMPTY

FE: Framing error. OE: Overrun error. PE: Parity error.

	u i i i i	ig ciro	1. OL. C	VCITO		J1.1 L	. 1 411
	82	51 Mo	d Yazm	acı (A	senkr	on)	
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S_2	S_1	EP	PEN	L_2	L_1	B_2	B_1
S_2	S_1	Stop	biti say	ISI			
0	0	Ir	nvalid				
0	1	1 s	top biti				
1	0	1.5	stop bit	:i			
1	1	2 s	top biti				
EP	F	Parity					
0	Od	d parit	.у				
1	Eve	n pari	ty				
PE	V P	arity e	nable				
0		Parity	yok				
1		Parity	var				
L_2	L_1	Data	bit say	ISI			
0	0		5				
0	1		6				
1	0		7				
1	1		8				
B_2	B_1	Baud	d rate f	actor			

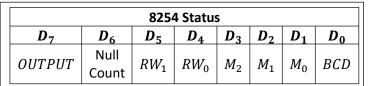
Senkron mod

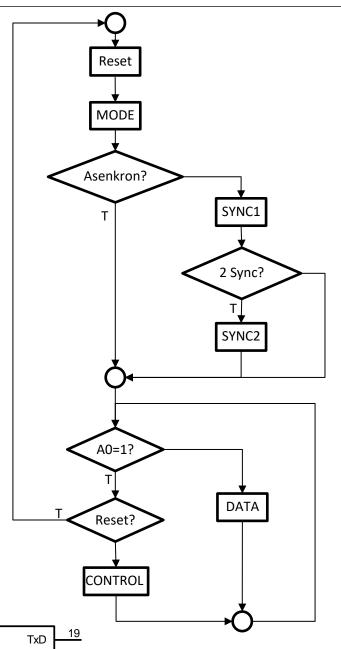
16

0

1

1



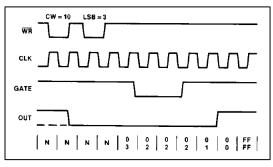


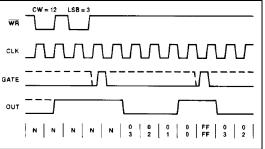
	82	254 Adresleme
A_1	A_0	Yazmaç
0	0	Counter0, Status0
0	1	Counter1, Status1
1	0	Counter2, Status2
1	1	Control

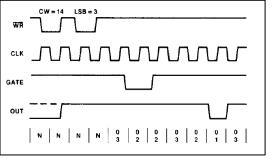
	_		8	254	Kor	ntro	ΙY	azr	nacı			
D_7		D_6	L) ₅	D	4	D	3	D_2		D_1	D_0
SC_1	Ŀ	SC_0	R	W_1	RV	V_0	M	12	M_1		M_0	BCD
SC_1		$\overline{SC_0}$		SC -	Sel	ect (Со	unt	ter			
0		0			Co	unte	erC)				
0		1			Co	unte	er1					
1		0			Co	unte	er2					
1		1	R	ead	Bac	k Co	om	ıma	and			
M_2	1	M_1	M) I	M –	Мо	d					
0		0	0		Mc	od 0						
0		0	1		Mc	d 1						
Χ		1	0		Mc	d 2						
Χ		1	1		Mc	od 3						
1		0	0		Mc	d 4						
1		0	1		Mc	od 5						
RW	1	RV	V_0		RW	/ — F	Rea	ad/	'Wri	te	•	
0		С		Со	unte	er La	atc	h (Comi	m	and	
0		1					LS	b				
1		C)				M	Sb				
1		1	-	(Önce	e LS	b, :	sor	ra N	15	Sb	
BCL)			Say	yma							
0				Bir	nary							
1		Bin	ary	Coc	led I	Deci	m	al				
			8	3254	1 Re	ad E	3ac	ck (Com	m	and	

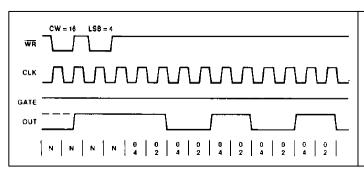
		825	4 Read Bac	k Comm	and		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	COUNT	<u>STATUS</u>	CNT_2	CNT_1	CNT_0	0
\overline{COUI}	$\overline{VT} =$	0 : Sayn	na değeri tu	ıt			
\overline{STAT}	$\overline{US} =$	= 0 : Duru	ım tut				
CNT2	2 = 1	: Sayı	cı 2 için tut				
CNT1	l = 1	: Sayı	cı 1 için tut				
CNT() = 1	: Sayı	cı0 için tut				

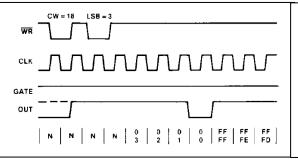
8 D0 D1 D2 D3 D4 D5 D6 D7 WR MR MR MD A0 A1 CS	CLK0 GATE0 OUT0 CLK1 GATE1 OUT1 CLK2 GATE2 OUT2	9 11 10 15 14 13 18 16 17
--	---	---

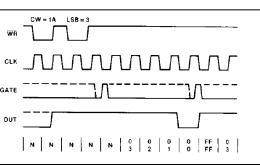












000	ADD destina	ADD destination, source			ODITSZAPC
900	Addition				X X X X X X X X X X X X X X X X X X X
Operands	spu	Clocks	Transfers	Bytes	Coding Example
register, register	er	3	-	2	ADD CX, DX
register, memory	ory.	9+EA	1	2-4	ADD DI, [BX]
memory, register	ter	16+EA	2	2-4	ADD TEMP, CL
register, immediate	diate	4		3-4	ADD CL, 2
memory, immediate	ediate	17+EA	2	3-6	ADD ALPHA, 2
accumulator, immediate	nmediate	4	-	2-3	ADD AX, 200

	AND destina	AND destination, source			ODITSZAPC
AND	Logical and				O x n x x O sapil
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, register	iter	3	-	2	AND AL, BL
register, memory	nory	9+EA	н	2-4	AND CX, FLAG_WORD
memory, register	ster	16+EA	2	2-4	AND ASCII [DI], AL
register, immediate	ediate	4	,	3-4	ND CX, 0F0H
memory, immediate	nediate	17+EA	2	3-6	AND BETA, 01H
adaile a constitue de la constitue de	inches on the	•		,	000000000000000000000000000000000000000

accumulator, immediate	te 4	-	2-3	AND AX, 01010000B
CALL target	CALL target			Hags ODITSZAPC
Operands	Clocks	Transfers	Bytes	Coding Example
near-proc	19	1	က	CALL NEAR_PROC
far-proc	28	2	2	CALL FAR_PROC
memptr16	21+EA	2	2-4	CALL PROC_TABLE[SI]
regptr16	16	1	2	CALL AX

SIC	CLC (no operands) Clear carry flag	rands) flag				Flags OD ITSZAPC
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
no operands		2		T	CLC	
ē	CLI (no operands)	ands)				ODITSZAPC
3	Clear interrupt flag	upt flag				0
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
no operands		2	-	1	CLI	
9	CMP destin	CMP destination, source				ODITSZAPC
<u> </u>	Compare de	Compare destination to source	ource			x x x x x x x
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
register, register	ster	3		2	CMP BX. CX	X

Flags OD T S Z	Bytes Coding Example	T) (C[1
	Transfers Byt	- 1
erands) rupt flag	Clocks	2
CLI (no operands) Clear interrupt flag	Operands	no operands

ç	CMP destina	CMP destination, source			Ido	ODITSZA
N N	Compare de	Compare destination to source	ource		× X X X X X X X X X X X X X X X X X X X	×
Operands	ands	Clocks	Transfers	Bytes	Coding Example	e
register, register	ster	3	-	2	CMP BX, CX	
register, memory	nory	9+EA	Н	2-4	CMP DH, ALPHA	
memory, register	ster	9+EA	1	2-4	CMP [BX+2], SI	
register, immediate	ediate	4	,	3-4	CMP BL, 02H	
memory, immediate	nediate	10+EA	1	3-6	CMP TABLE[BX+2000], 3420H	3420H
accumulator immediate	immediate	_		2-3	CMP AL DODGOOD	

à	DIV source				ODITSZAPC
Š	Division, unsigned	signed			n n n n cap L
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
reg8		06-08	-	2	DIV CL
reg16		144-162	1	7	DIV BX
mem8		(86-96)+EA	7	2-4	DIV ALPHA
mem16		(150-168)+EA	н	2-4	DIV TABLE [SI]

IN accul	IN accumulator, port			ODITISZAPIC
	Input byte or word			ridgs
Operands	Clocks	Transfers	Bytes	Coding Example
accumulator, immed8	10	1	2	IN AL, OFFEAH
accumulator, DX	8	1	⊣	IN AX, DX

JNI	INC destination	tion			STIDO	ITSZAPC
ו	Increment by 1	by 1			×	××××
Operands	ands	Clocks	Transfers	Bytes	Coding Example	
reg16		7	-	1	INC CX	
reg8		က	,	2	INC BL	
memory		15+EA	2	2-4	INC ALPHA[DI+BX]	

INI	INT interrupt-type Interrupt	t-type			Flags OD T	0D T S Z A P C
Operands	spu	Clocks	Transfers	Bytes	Coding Example	100
immed8 (type=3)	=3)	52	2	П	INT 3	
immed8 (type≠3	≠3)	51	2	2	INT 67	

immed8 (type≠3)	e≠3)	51	5	2	INT 67
IRET	IRET (no operands)	erands)			Flags ODITSZAPC
	Interrupt return	turn			
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
no operands		24	3	1	IRET
<u> </u>	JC short-label	el			CODITSAPIC
יל	Jump if carry	>			L dgs
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
short-lahel		16 or 4		C	IC CARRY-SET

21/31	JE/JZ short-label	label			ODITSZAPC
JE/ JE	Jump if equ	Jump if equal / Jump if zero	ero		capin
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
short-label		16 or 4	-	2	JZ ZERO
QVI	JMP target				ODITSZAPC
HAIC	Jump				ridgo
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
lodel troda		11		,	Tachoat

	IMP target					T C 7 A P
ΔM	יייו יייופרי				Flags	,
	Jump				003	
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	ole
short-label		15	-	2	JIMP SHORT	
near-label		15	•	က	JMP WITHIN_SEGMENT	_
far-label		15	1	2	JMP FAR_LABEL	
memptr16		18+EA	1	2-4	JMP [BX]	
regptr16		11	1	2	JMP CX	
memptr32		24+EA	2	2-4	JMP FAR [BX+123H]	

LAHF	LAHF (no operands) Load AH from flags	perands) im flags				Flags
Oper	Operands	Clocks	Transfers	Bytes	Coc	Coding Example
no operands		7	-	1	LAHF	

AH S	F (no og	LAHF (no operands)			Flags ODITSZAPC
Operands		Clocks	Transfers	Bytes	Coding Example
		4		1	LAHF
FA dec	tina	IFA destination source			ODITISZAPIC
oad ef	fecti	Load effective address			_
Operands		Clocks	Transfers	Bytes	Coding Example
reg16, mem16		2+EA		2-4	LEA BX, [BP+DI]
LOOP short-label	iort	-label			ODITSZAPC
Loop					riago
Operands		Clocks	Transfers	Bytes	Coding Example
	Ī				

	200				
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
short-label		17/5	-	2	LOOP AGAIN
12	MUL source				ODITSZAPC
5	Multiplicati	Multiplication, unsigned			x n n n n x cap L
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
reg8		<i>LL-01</i>		2	MUL BL
reg16		118-133		2	MULCX
mem8		(76-83)+EA	1	2-4	MUL MONTH[SI]
mem16		(124-139)+EA	1	2-4	MUL BAUD_RATE

MOV	MOV destin	MOV destination, source			Flags
	Move)
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
memory, accumulator	umulator	10	1	က	MOV ARRAY[SI], AL
accumulator, memory	, memory	10	1	33	MOV AX, TEMP_RESULT
register, register	ster	2	,	2	MOV AX, CX
register, memory	nory	8+EA	1	2-4	MOV BP, STACK_TOP
memory, register	ister	9+EA	1	2-4	MOV COUNT[DI], CX
register, immediate	nediate	4	,	2-3	MOV CL, 2
memory, immediate	nediate	10+EA	1	3-6	MOV MASK[BX+SI], 2CH
seg-reg, reg16	9	2	,	2	MOV ES, CX
seg-reg, mem16	116	8+EA	1	2-4	MOV DS, SEGMENT_BASE
reg16, seg-reg	ρ0 20	2	,	2	MOV BP, SS
memory, seg-reg	reg	9+EA	Т	2-4	MOV DATA2, CS

Ĉ	OR destination, source	ion, source			ODITSZAPIC
5	Logical inclusive or	usive or			O X N X X O S S P I J
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, register	ster	3		2	OR AL, BL
register, memory	nory	9+EA	1	2-4	OR DX, PORT_ID[DI]
memory, register	ster	16+EA	2	2-4	OR FLAG_BYTE, CL
accumulator, immediate	immediate	4	,	2-3	OR AL, 01101100B
register, immediate	ediate	4	,	3-4	OR CX, 01H
otcibo casa: , a oca o ca	0,000	17.64	C	0 0	וויייסטון מסירי אפן פס

TUO	OUT port, accumulate	OUT port, accumulator			Flags OD I TSZAP C	O
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
immed8, accumulator	umulator	10	Н	2	OUT 44, AX	
DX, accumulator	ator	8	1	1	OUT DX, AL	
90	POP destination	ation			ODITSZAPIC	C
5	Pon word off stack	ff stack			Tags	

	Pop word on stack	II Stack			
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register		8	1	1	POP DX
seg-reg (CS illegal)	legal)	8	н	н	POP DS
memory		17+EA	2	2-4	POP PARAMETER
3	PUSH source	a			ODITSZAPC
בנסב	Push word onto stack	onto stack			capin

000	es Coding Example	PUSH SI	PUSH ES	4 PUSH RETURN_CODE[SI]	
	Transfers Bytes	1 1	1 1	2 2-4	
Push word onto stack	ls Clocks	11	1) 10	16+EA	
	Operands	register	seg-reg (CS illegal)	memory	

PUSHF (no operands)	Push flags onto stack	ds Clocks Transfers Bytes Coding Example	10 1 PUSHF	
PUSHF (no o	Push flags or	Operands		
PUSHF		Oper	no operand	

130	RCL destination, count	tion, count			ODITSZAPC
J.	Rotate left t	Rotate left through carry			X capil
Operands	spue	Clocks	Transfers	Bytes	Coding Example
register, 1		7	-	2	RCL CX, 1
register, CL		8+4*bit	1	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+2], CL

PET	RET optiona	RET optional-pop-value			ODITSZAPC
1	Return from	Return from procedure			- 1883
Operands	ands	Clocks	Transfers	Bytes	Coding Example
(intra-segment, no pop)	nt, no pop)	8	1	1	RET
(intra-segment, pop)	nt, pop)	12	1	e	RET 4
(inter-segment, no pop)	nt, no pop)	18	2	1	RET
(inter-segment, pop)	nt, pop)	17	2	e	RET 2

Ğ	ROL destina	ROL destination, count			ODITSZAPC
Ž	Rotate left				X X X
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, 1		2		2	ROL BX, 1
register, CL		8+4*bit		2	ROL DI, CL
memory, 1		15+EA	2	2-4	ROL FLAG_BYTE[DI], 1
memory Cl		20+FA+4*hit	2	2-4	ROI AIPHA CI

R ROR destination, count Coperands Clocks 1 2 2 1 8+4*bit 15+EA	Transfers - 2	Bytes 2 2 2 2-4	Flags ODI TS Z A P C N
20+EA+4*bit	bit 2	2-4	ROR CMD_WORD, CL

SAHF	SAHF (no operands)	perands)			Flags OD ITSZAPC
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
no operand		4		1	SAHF
SAL/SHL	SAL/SHL de	SAL/SHL destination, count	nt		Flags
	Shift arithm	Shift arithmetic left/Shift logical left	logical left		×
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
register, 1		2	•	2	SAL AL, 1
register, CL		8+4*bit		2	SHL DI, CL
memory, 1		15+EA	2	2-4	SHL [BX], 1
Č			(0

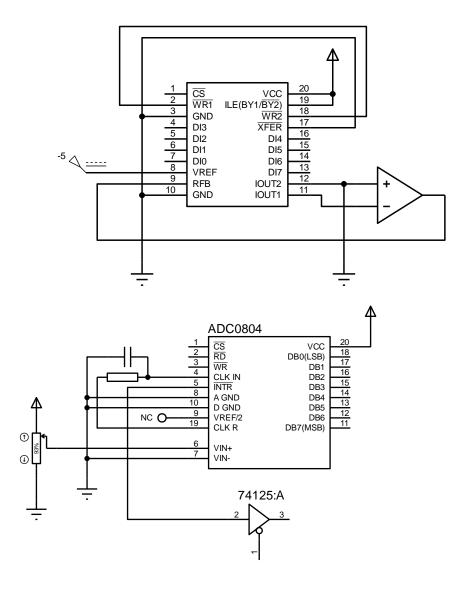
memory, 1		15+EA	2	2-4	2-4 SHL [BX], 1
memory, CL		20+EA+4*bit	2	2-4	SAL STORE_COUNT, CL
į	STC (no operands)	erands)			ODITSZAPC
אר	Set carry flag	38			ridgs 1
Operands	ands	Clocks	Transfers Bytes	Bytes	Coding Example
no operand		2		1	STC

STC	Set carry flag	(B				Flags - 1
Operands	ands	Clocks	Transfers Bytes	Bytes	9	Coding Example
no operand		2		1	STC	
E	STI (no operand)	rand)				ODITSZAPIC
=	Set interrup	Set interrupt enable flag				ridgs 1

Operations	CIOCKS	Hallsters	הארכי	COULING EVAILIBLE
no operand	2	-	1	STI
	SUB destination, source			ODITSZAPC
Subtraction				Tiags ×
Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	SUB CX, BX
register, memory	9+EA	1	2-4	SUB DX, MATH_TOTAL[SI]
memory, register	16+EA	2	2-4	SUB [BP+2], CL
accumulator, immediate	4	,	2-3	SUB AL, 10
register, immediate	4	,	3-4	SUB SI, 5280
memory, immediate	17+EA	2	3-6	SUB [BP], 1000

XOR destin	XOR destination, source			ODITSZAPC
Logical exclusive or	lusive or			O X N X X O S S P I J
Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	XOR CX, BX
register, memory	9+EA	П	2-4	XOR CL, MASK_BYTE
memory, register	16+EA	2	2-4	XOR ALPHA[SI], DX
accumulator, immediate	4		2-3	XOR AL, 01000010B
register, immediate	4		3-4	XOR SI, 00C2H
memory, immediate	17+EA	2	3-6	XOR RETURN_CODE, 0D2H

		INPUTS	5				ekoder Fonksiyon Tablosu OUTPUTS								
	ENABLE		9	ELEC	Γ		0017013								
E1	<u>E2</u>	E3	С	В	А	<u>¥0</u>	<u>¥1</u>	<u>¥2</u>	<u>¥3</u>	<u>¥4</u>	<u>¥5</u>	<u>¥6</u>	<u>77</u>	SELECTED	
L	Х	Х	Х	Х	Х	Н	Н	Н	Н	Н	Н	Н	Н	NONE	
Χ	Х	Н	Х	Χ	Х	Н	Н	Н	Н	Н	Н	Н	Н	NONE	
Χ	Н	Х	Х	Х	Х	Н	Н	Н	Н	Н	Н	Н	Н	NONE	
Н	L	L	L	L	L	L	Н	Н	Н	Н	Н	Н	Н	<u> 70</u>	
Н	L	L	L	L	Н	Н	L	Н	Н	Н	Н	Н	Н	<u> </u>	
Н	L	L	L	Н	L	Н	Н	L	Н	Н	Н	Н	Н	<u>¥2</u>	
Н	L	L	L	Н	Н	Н	Н	Н	L	Н	Н	Н	Н	 <u>¥3</u>	
Н	L	L	Н	L	L	Н	Н	Н	Н	L	Н	Н	Н	<u>¥4</u>	
Н	L	L	Н	L	Н	Н	Н	Н	Н	Н	L	Н	Н	<u> 75</u>	
Н	L	L	Н	Н	L	Н	Н	Н	Н	Н	Н	L	Н	<u>¥6</u>	
Н	L	L	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	L	<u> 77</u>	





		8259 <i>ICW</i> ₁												
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
0	Χ	Х	Х	1	LTIM	0	SNGL	IC_4						
LTIM		Açıklama												
0	Kena	ar teti	kleme	غ										
1	Sevi	ye tet	iklem	e										
SNGL		Açık	lama											
0	Kask	at ba	ğlı 82	59'lar										
1	Tek	8259												
IC ₄		Açıkla	ma											
0	<i>IC</i> ₄ kı	ıllanıl	maya	cak										
1	<i>IC</i> ₄ kı	ıllanıl	acak											

			8259 <i>ICW</i> ₂										
A_0		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
1		A_7	A_6	A_5	A_4	A_3	Χ	Х	Х				
$\overline{(A_7A_7)}$	$(A_7A_6A_5A_4A_3000)_2$ IRO için kesme isteği adresi												

	_	8259 ICW ₃ SGNL=0 ise (Master)												
A_0		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0					
1		S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0					
S_i		-	Açıkla	ma										
0	Ħ	R_i 'ye s	slave	bağlı (
1	II	R_i 'ye s	slave	bağlı										

		8259 <i>ICW</i> ₃ SGNL=0 ise (Slave)												
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
1	0	0	0	0	0	ID_2	ID_1	ID_0						
$\overline{(ID_2II}$	$(ID_2ID_1ID_0)_2$ Slave ID													

				8259	ICW ₄					
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	0	0	0 0 SFNM BUF M/S AEOI μP							
BUF M/S Buffered – Master/Slave										
0	Χ		Nor	n-buffere	d					
1	0		Buff	ered slav	/e					
1	1		Buffered master							
AEOI=	1 otom	atik k	esme	sonland	ırma	_				

AEOI=1 otomatik kesme sonlandırma μP =1 8086 için SFNM=0, BUF=0, M/S=0 kullanılacak

		8259 OCW ₁										
A_0	D_7	$D_7 D_6$		D_4	D_3	D_2	D_1	D_0				
1	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0				
M_i	Açıkla	ma										
0	Mask r	eset										
1	Mask s	et										

		8259 OCW ₂											
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0					
0	R	SL	EOI	0	0	L_2	L_1	L_0					

		8259 OCW ₃												
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
0	0	ESMM	SMM	0	1	Р	RR	RIS						

