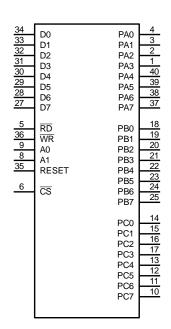
8	3 2 55	Adresleme
A_1	A_0	Port
0	0	PortA
0	1	PortB
1	0	PortC, Status
1	1	Kontrol

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 ($B_2B_1B_1$	$(0)_2$ nu	umara	lı pini	ni seç.						

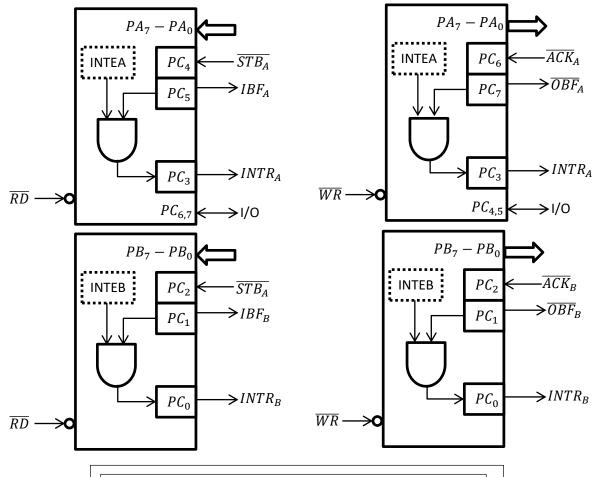
S/R: Seçilen pinde Set (1) veya Reset (0) oluştur.

_		8255 Basit I/O (Mod 0) Kontrol Yazmacı												
D_6	D_1	D_0												
0	0	PA	PC_U	0	PB	PC_L								
PA PB PC _U PC _L Port Yönü														
0	0	0	Çı	kış										
1	1	1	G	iriş										
	0	0 0	0 0 PA	$\begin{array}{c cccc} 0 & 0 & PA & PC_U \\ \hline PB & PC_U & PC_L & Port \\ \hline 0 & 0 & 0 & \varsigma_1 \\ \hline \end{array}$	$0 0 PA PC_U 0$	$egin{array}{ c c c c c c c c c c c c c c c c c c c$								



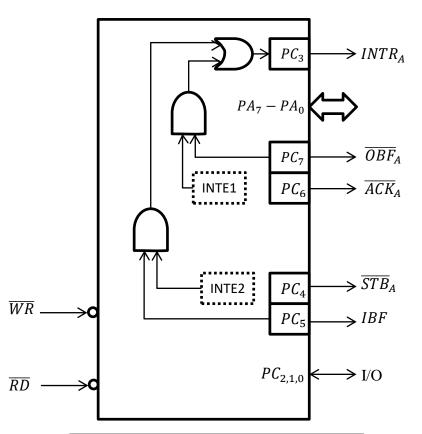
	8255 Mod 1 (Grup A) Kontrol Yazmacı										
	D_7	D_6	D	5	D_4	D	3	D_2	D_1	D_0	
	1	0	1	L	PA	PC	6,7	X	X	X	
	PA	PC_6	,7	Р	ort Yċ	inü					
Г	0	0			Çıkış	•					
	1	1			Giriş	;					

8255 Mod 1 (Grup B) Kontrol Yazmacı													
D_6	D_5	D_4	D_3	D_2	D_1	D_0							
X	X	X	X	1	PB	X							
Port	Yönü	i											
Ç	ıkış												
G	iriş												
	D ₆ X Port Ç	D ₆ D ₅ X X	D6 D5 D4 X X X Port Yönü Çıkış	D6 D5 D4 D3 X X X X Port Yönü Çıkış	D6 D5 D4 D3 D2 X X X X 1 Port Yönü Çıkış	D6 D5 D4 D3 D2 D1 X X X X 1 PB Port Yönü Çıkış							



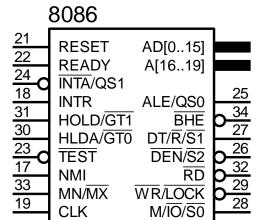
			825	5 Mo	d 1 Input	Status				
D_7	D_6	D_5		D_4	D_3	D_2	D_1	D_0		
I/O	1/0	IBF	$_{4} \mid IN$	TE_A	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$		
					• • • •					
			825	5 Mod	d 1 Outpu	t Status				
D_7	$\mid D_7 \mid D_6 \mid D_5 \mid D_4 \mid D_3 \mid D_2 \mid D_1 \mid D_0$									
\overline{OBF}_{i}	₄ IN	TE_A	I/O	1/0	$INTR_A$	$INTE_B$	\overline{OBF}_{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	255	Mod 2	(Gru	р А) ŀ	Contro	ol Yaz	macı
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1 1 <i>X X</i>				X	X	$PC_{2,1,0}$
PC_2	,1,0	Port '	Yönü				
0		Çıl	κιş				
1		Gi	riş				
				_			

	8255 Mod 2 Status												
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	μP → 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	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_6 D_5 D_4 D_3 D_2 D_1 D_0											
EH IR RTS ER SBRK RXE DTR TXE											

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

		825	1 Sta	tus Ya	zmacı		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY

D[0..7] RESET CLK C/D WR RD

DTR DSR RTS CTS

TxEMPTY

TxC

RxD

 \overline{RxC}

RxRDY

SYNDET

9

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

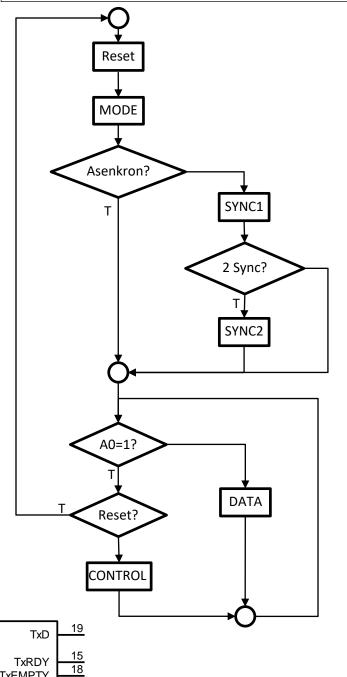
82	51 Mo	d Yazm	acı (Asenkı	ron)	
D_6	D_5	D_4	D_3	D_2	D_1	D_0
S_1	EP	PEN	L_2	L_1	B_2	B_1
S_1	Stop	biti say	ISI			
0	Ir	nvalid				
1	1 s	top biti				
0	1.5	stop bit	i			
1	2 s	top biti				
	Parity					
Oc	ld parit	У				
Eve	en pari	ty				
V F	arity e	nable				
	D ₆ S ₁ 0 1 0 1 Eve	D6 D5 S1 EP S1 Stop 0 Ir 1 1 s 0 1.5 1 2 s Parity Odd parit Even parit	D6D5D4S1EPPENS1Stop biti say0Invalid11 stop biti01.5 stop biti12 stop bitiParityOdd parityEven parity	D6D5D4D3S1EPPENL2S1Stop biti sayısı0Invalid11 stop biti01.5 stop biti12 stop biti12 stop biti Parity Odd parity Even parity	D6D5D4D3D2S1EPPENL2L1S1Stop biti sayısıOInvalid11 stop biti01.5 stop biti12 stop bitiParityOdd parityEven parity	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$

L_2	L_1	Data bit sayısı
0	0	5
0	1	6
1	0	7
1	1	8

Parity yok Parity var

B_2	B_1	Baud rate factor
0	0	Senkron mod
0	1	1
1	0	16
1	1	64

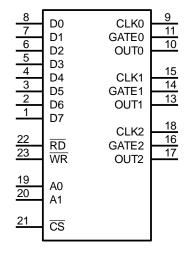
		8254	4 Status	5			
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	Null Count	RW_1	RW_0	M_2	M_1	M_0	BCD

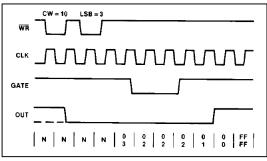


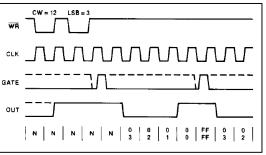
	82	254 Adresleme
A_1	A_0	Yazmaç
0	0	Counter0, Status0
0	1	Counter1, Status1
1	0	Counter2, Status2
1	1	Control

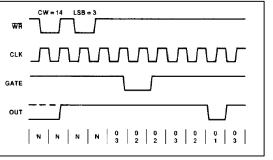
		8	254	Kontro	ΙY	azr	nacı		
D_7	D_6		5	D_4	D	3	D_2	D_1	D_0
SC_1	SC_0	RI	N_1	RW_0	M	1 ₂	M_1	M_0	BCD
SC_1	SC_0	, 5	SC -	Select	Co	unt	ter		
0	0			Count	er0)			
0	1			Count	er1				
1	0			Count	er2				
1	1	R	ead	Back C	om	ıma	and		
M_2	M_1	M) I	M – Mo	d				
0	0	0		Mod 0					
0	0	1		Mod 1					
Χ	1	0		Mod 2					
Χ	1	1		Mod 3					
1	0	0		Mod 4					
1	0	1		Mod 5					
RW_1	L RV	V_0		RW –	Rea	ad/	Writ	е	
0	()	Со	unter L	atc	h (Comn	nand	
0		L			LS	b			
1	()			M:	Sb			
1		L	(Önce LS	b, s	sor	ra M	Sb	
BCD)		Say	yma					
0			Bir	nary					
1	Bir	ary	Coc	led Dec	ima	al			
		8	3254	l Read ነ	Bac	ck (comn	nand	

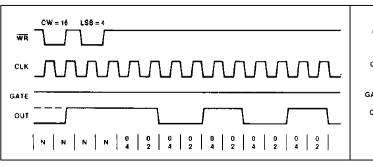
		825	4 Read Bac	k Comm	and		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	\overline{COUNT}	<u>STATUS</u>	CNT_2	CNT_1	CNT_0	0
COUN	$\overline{T} =$	0 : Sayn	na değeri tu	ıt			
\overline{STAT}	$\overline{US} =$: 0 : Duru	ım tut				
CNT2	= 1	: Sayı	cı 2 için tut				
CNT1	= 1	: Sayı	cı 1 için tut				
CNT0	= 1	: Sayı	cı0 için tut				

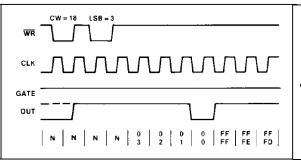


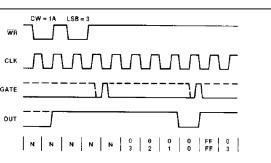












200	ADD destin	ADD destination, source			ODITSZAPC
ADD	Addition				X X X X X X X X X X X X X X X X X X X
Operands	nds	Clocks	Transfers	Bytes	Coding Example
register, register	er	3	-	2	ADD CX, DX
register, memory	ory.	9+EA	1	2-4	ADD DI, [BX]
memory, register	ter	16+EA	2	2-4	ADD TEMP, CL
register, immediate	diate	4	1	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			ODITSZAPIC
AND	Logical and				
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, register	ster	3	1	7	AND AL, BL
register, memory	nory	9+EA	1	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
accumulator immediate	immediate	4		2-3	AND AX. 01010000B

accumulator, immediate	immediate	4	-	2-3	2-3 AND AX, 01010000B
	CALL target				ODITSZAPC
CALL	Call a procedure	dure			Sabin
Operands	ands	Clocks	Transfers	Bytes	Coding Example
near-proc		19	1	3	CALL NEAR_PROC
far-proc		28	2	5	CALL FAR_PROC
memptr16		21+EA	2	2-4	CALL PROC_TABLE[SI]
regptr16		16	1	2	CALL AX

CIC	CLC (no operands) Clear carry flag	rands) flag				Flags ODITSZAPC
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
no operands		2		1	CLC	
į	CLI (no operands)	rands)				ODITSZAPC
3	Clear interrupt flag	upt flag				riags 0
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
no operands		2		1	CLI	
9	CMP destin	CMP destination, source				ODITSZAPC
	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
Oper	Operands	Clocks	Transfers	Bytes		Coding Example
20401000		,		,	70 70 70 70	3

į	CLI (no operands)	ands)				ZSTIGO	T S Z A
3	Clear interrupt flag	upt flag				LIGRS 0	
Operands	ands	Clocks	Transfers	Bytes	3	Coding Example	e
no operands		2	-	1	CLI		

9	CMP destina	CMP destination, source			AZSTIGO
	Compare de	Compare destination to source	ource		X X X X Sapil
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, register	ster	3	-	2	CMP BX, CX
register, memory	nory	9+EA	1	2-4	СМР 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
accumulator, immediate	immediate	4		2-3	CMP AL. 00010000B

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

mem16		(150-168)+EA	П	2-4	DIV TABLE [SI]
3	IN accumulator, port	ator, port			ODITSZAPIC
Ē	Input byte or word	or word			LIGES
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
accumulator, immed8	immed8	10	1	2	IN AL, OFFEAH
accumulator, DX	DX	∞	1	Т	IN AX, DX
J.	INC destination	tion			ODITSZAPIC
ر <u>اع</u>	Increment by 1	oy 1			X X X X X X X X X X X X X X X X X X X
Oper	Operands	Clocks	Transfers	Bytes	Coding Example

2	INC destination	tion			AZZILZODITSZAP	ZAP
2	Increment by 1	oy 1			X X X CSp.L	XXXX
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
reg16		2	-	1	INC CX	
reg8		ю		2	INC BL	
, ao mo m		1515	,	7-0	INC ALDIACIOLERY]	

į	INT interrupt-type	ot-type			ODITSZAPIC
2	Interrupt				00 8891
Operands	ands	Clocks	Transfers	Bytes	Coding Example
immed8 (type=3)	e=3)	52	2	1	INT 3
immed8 (type≠3	e≠3)	51	5	2	INT 67

1301	IRET (no operands)	erands)				ODITSZAPC
Į.	Interrupt return	turn			LI LINES L L	rrrrrr
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	ple
no operands		24	3	1	IRET	
)r	JC short-label	e			Flags	ODITSZAPC
2	Jump if carry	>			0005	
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	ple
short-label		16 or 4		7	JC CARRY-SET	

21/31	JE/JZ short-label	label			ODITSZAPC
JE/JF	Jump if equ	Jump if equal / Jump if zero	10		ridgs
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
short-label		16 or 4	-	2	JZ ZERO
IMP	JMP target				ODITSZAPC
	Jump				1985
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
chort labol		11		,	Taous and

short-label		16 or 4	-	2	2 JZ ZERO					
9	JMP target					-	\circ	DITSZA	SZ	₹
Z Z	Jump					ridgs				
Oper	Operands	Clocks	Transfers	Bytes	8	Coding Example	xam	ple		
short-label		15	-	2	JMP SHORT					
near-label		15		က	JMP WITHIN_SEGMENT	N_SEG	MEN	⊨		
far-label		15		2	JMP FAR_LABEL	4BEL				
memptr16		18+EA	1	2-4	JMP [BX]					
regptr16		11		2	JMP CX					
memntr37		24+FA	2	D-C	IMP FAR [RX+123H]	X+123	Ξ			

	LAHF (no operands)	perands)				ODITSZAP
TAHE	Load AH from flags	m flags				Flags
Oper	Operands	Clocks	Transfers	Bytes	9	Coding Example
no operands		4	-	1	LAHF	

	, =::::::::::::::::::::::::::::::::::::				110000000000000000000000000000000000000
LAHF	LAHF (no operands) Load AH from flags	ກ flags			Flags OD I T S Z A P C
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
no operands		4	-	1	LAHF
-	LEA destina	LEA destination, source			ODITSZAPC
LEA	Load effective address	ve address			ridgs
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
reg16, mem16	91	2+EA	-	2-4	LEA BX, [BP+DI]
acci	LOOP short-label	-label			ODITSZAPC
1001	Loop				- 1dg5
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
short-label		17/5	-	2	LOOP AGAIN
1	MUL source				ODITSZAPC
MOL	Multiplication	Multiplication, unsigned			x nnnn x
oper Oper	Operands	Clocks	Transfers	Bytes	Coding Example
		1			

900	LOOP short-label	label			ū]]]	Ξ	ODITSZ
Ď	Loop				_	ridgs		
Operands	ands	Clocks	Transfers	Bytes	Codi	Coding Example	nple	au
short-label		17/5		2	LOOP AGAIN			
	MUI source						Ę	ODITSZ

	MUL source				AZSTIGO
MOL	Multiplicati	Multiplication, unsigned			nnn x
Operands	ands	Clocks	Transfers	Bytes	Coding Example
reg8		70-77	,	2	MUL BL
reg16		118-133	1	7	MUL CX
mem8		(76-83)+EA	1	2-4	MUL MONTH[SI]
mem16		(124-139)+EA	н	2-4	MUL BAUD_RATE

202	MOV destin	MOV destination, source			ODITSZAPIC
202	Move				Flags
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
memory, accumulator	umulator	10	1	က	MOV ARRAY[SI], AL
accumulator, memory	memory	10	1	æ	MOV AX, TEMP_RESULT
register, register	ster	2	,	7	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	ediate	4		2-3	MOV CL, 2
memory, immediate	nediate	10+EA	1	3-6	MOV MASK[BX+SI], 2CH
seg-reg, reg16	9	2	,	7	MOV ES, CX
seg-reg, mem16	116	8+EA	1	2-4	MOV DS, SEGMENT_BASE
reg16, seg-reg	20	2	i	7	MOV BP, SS
memory. seg-reg	-reg	9+EA	_	2-4	MOV DATA2. CS

	,				
ő	OR destination, source	ion, source			ODITSZAPC
5	Logical inclusive or	isive or			0 x n x x 0 cSpl
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, register	iter	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
0+0:10000000000000000000000000000000000	o-cipo-	47.7	·	0	וויכנו עם מס

5	OUI port, a	OUI port, accumulator			Class ODI I SZAPIC	ر
3	Output byte or word	e or word			capin	
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
immed8, accumulator	umulator	10	1	2	OUT 44, AX	
DX, accumulator	ator	∞	1	⊣	OUT DX, AL	
						l
G	POP destination	ation			ODITSZAPIC	O
5	Pop word off stack	ff stack			ridgs	
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
register		8	1	1	POP DX	
seg-reg (CS illegal)	legal)	∞	7	н	POP DS	
, 4000000		17,50	,	,	DODDANACTED	

memory		17+EA	2	2-4	2-4 POP PARAMETER
13110	PUSH source	e e			ODITSZAPIC
ב	Push word onto stack	onto stack			LIGES
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
register		11	1	1	PUSH SI
seg-reg (CS illegal)	legal)	10	П	⊣	PUSH ES

PUSHE	PUSHF (no operands)	operands)			Flags	ODITSZAPC
:	Push flags onto stack	nto stack			0	
Operands	ands	Clocks	Transfers	Bytes	Coding Example	xample
no operand		10	1	1	PUSHF	

	Operands	no operand	2	J.	Operands	register, 1	register, CL	memory, 1	memory, CL
י מטון וותפט סוונס פתבוי	ands		RCL destina	Rotate left i	spue				
35450	Clocks	10	RCL destination, count	Rotate left through carry	Clocks	7	8+4*bit	15+EA	20+EA+4*bit
•	Transfers	1			Transfers		,	2	2
	Bytes	1			Bytes	2	7	2-4	2-4
	Coding Example	PUSHF	ODITSZAPIC	× × × × × × × × × × × × × × × × × × ×	Coding Example	RCL CX, 1	RCL AL, CL	RCL ALPHA, 1	RCL [BP+2], CL

	DET continue	order ace				
130	מוחוולה ושע	rei optional-pop-value				Elags ODI I 34AFC
1	Return from	Return from procedure				- 1dg2
Operands	ands	Clocks	Transfers	Bytes	S	Coding Example
(intra-segment, no pop)	nt, no pop)	8	1	1	RET	
(intra-segment, pop)	nt, pop)	12	1	ĸ	RET 4	
(inter-segment, no pop)	nt, no pop)	18	2	1	RET	
(inter-segment, pop)	nt, pop)	17	2	က	RET 2	

Ö	ROL destina	ROL destination, count			ODITSZAPC
2	Rotate left				X Sapil
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
I) homom		30+FA+1+bi+	,	,	כ אום וא וסם

G	ROR destina	ROR destination, count			ODITISZAPC
Ş	Rotate right				×
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, 1		7	-	2	ROR AL, 1
register, CL		8+4*bit	,	2	ROR BX, CL
memory, 1		15+EA	2	2-4	ROR PORT_STATUS, 1
memory, CL		20+EA+4*bit	2	2-4	ROR CMD_WORD, CL

Hellioly, CL		20TEAT4 DIL	7	+ -7	2-4 NON CIVID_WOND, CL
ENVS	SAHF (no operands)	perands)			ODITSZAPC
L	Store AH into flags	to flags			LI GBP LI
Oper	Operands	Clocks	Transfers	Bytes	Coding Example
no operand		4	,	1	SAHF
1113/143	SAL/SHL de	SAL/SHL destination, count	t		ODITSZAPC
SAL/ STIL	Shift arithm	Shift arithmetic left/Shift logical left	ogical left		x X X X X X X X X X X X X X X X X X X X
	7				1

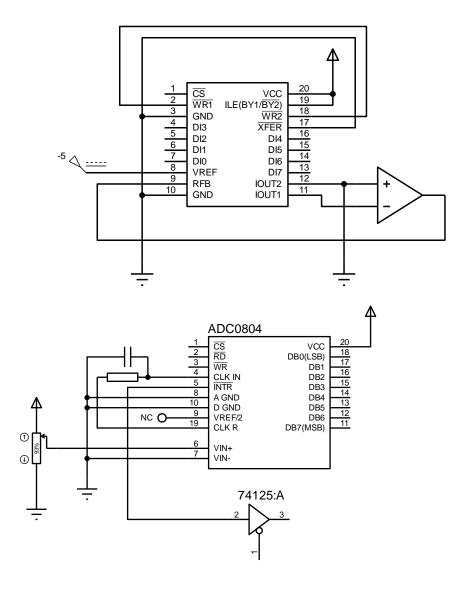
)		
Operands	ands	Clocks	Transfers	Bytes	Coding Example
register, 1		2		2	SAL AL, 1
register, CL		8+4*bit	,	2	SHL DI, CL
memory, 1		15+EA	2	2-4	SHL [BX], 1
memory, CL		20+EA+4*bit	2	2-4	SAL STORE_COUNT, CL
į	STC (no operands)	erands)			ODITSZAPIC
3IC	Set carry flag	ag.			ridgs
Onerands	ands	Clocks	Transfers Rytes	Rytes	Coding Example

	CTC (no operands)	randel				
STC	Set carry flag	lanus) B				Flags CD 1 1 2 2 7 1
Oper	Operands	Clocks	Transfers	Bytes	ŏ	Coding Example
no operand		2	-	1	STC	

STI	STI (no operand)	STI (no operand)			Flags OD I T S Z A P	А
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
no operand		2		1	STI	
	SUB destina	SUB destination, source			ODITSZAP	ΑР
SOB	Subtraction				riags ×	× × ×
Oper	Operands	Clocks	Transfers	Bytes	Coding Example	
register, register	ster	3		2	SUB CX, BX	
register, memory	nory	9+EA	1	2-4	SUB DX, MATH_TOTAL[SI]	
memory, register	gister	16+EA	2	2-4	SUB [BP+2], CL	
accumulator, immediate	, immediate	4	1	2-3	SUB AL, 10	
register, immediate	nediate	4		3-4	SUB SI, 5280	
memory, immediate	mediate	17+FA	2	3-6	SUB [RP] 1000	

XOX XOR	destinat	XOR destination, source			ODITSZAPC
	Logical exclusive or	sive or			0 x n x x 0 c Spil
Operands		Clocks	Transfers	Bytes	Coding Example
register, register		3		2	XOR CX, BX
register, memory		9+EA	1	2-4	XOR CL, MASK_BYTE
memory, register		16+EA	2	2-4	XOR ALPHA[SI], DX
accumulator, immediate	ediate	4		2-3	XOR AL, 01000010B
register, immediate	e	4		3-4	XOR SI, 00C2H
memory, immediate	te	17+EA	2	3-6	XOR RETURN_CODE, 0D2H

		INPUT							OUT	PUTS				IED UT
E1	ENABLE E2	E3	С	B	А	<u></u> Y 0	<u>¥1</u>	<u> </u>	<u> </u>	<u>¥4</u>	Y 5	<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	Х	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>¥1</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	\overline{Y3}
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>





				825	9 <i>ICW</i>	1		
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	Х	Х	1	LTIM	0	SNGL	IC ₄
LTIM	A	Çıkla	ma					
0	Kena	ır teti	kleme	غ				
1	Sevi	e 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					

			- 1	8259	ICW ₂	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	Х	Х	Х
$\overline{(A_7A_6}A$	$\overline{_5A_4A}$	3000) ₂ IRO	için l	kesme	isteğ	ji adre	esi

			8259	ICW:	3 SGN	IL=0 is	se (M	aster)	
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		ļ	Açıkla	ma					
0	IF	R_i 'ye s	slave	bağlı (değil				
1	IF	R_i 'ye s	slave	bağlı					

		825	9 <i>IC</i> V	V_3 SC	NL=0	ise (S	lave)	
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
$\overline{(ID_2ID_2)}$	$\overline{ID_0}$	Slav	e 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	SFNM	BUF	M/S	AEOI	μP		
BUF	M/S	Buff	ered	– Maste	r/Slave	2				
0	Х		Nor	n-buffere	d					
1	0		Buffered slave							
1	1		Buffe	red mas	ter					
AFOI=	1 otom	atik k	esme	sonland	ırma	_				

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

			8	3259 (OCW.	1		
A_0	D_7	D_6	D_5	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	Açıkl	ama						
0	Mask	reset						
1	Mask	set						

			8	259 <i>(</i>	OCW ₂	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

			825	9 <i>OC</i>	$\overline{W_3}$			
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

