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

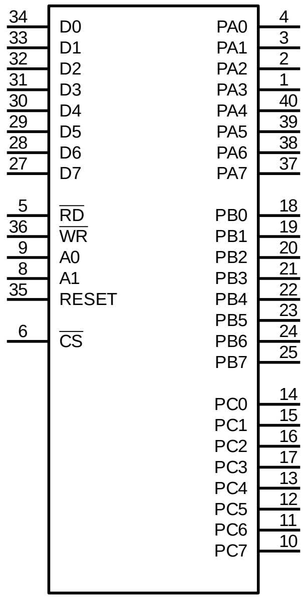
8255 Bit Set Reset Mod Kontrol Yazmacı							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	B_2	B_1	B_0	S/R

Port C'nin $(B_2B_1B_0)_2$ numaralı pinini seç.

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

8255 Basit I/O (Mod 0) Kontrol Yazmacı							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	PA	PC_U	0	PB	PC_L

PA	PB	PC_U	PC_L	Port Yönü
0	0	0	0	Çıkış
1	1	1	1	Giriş



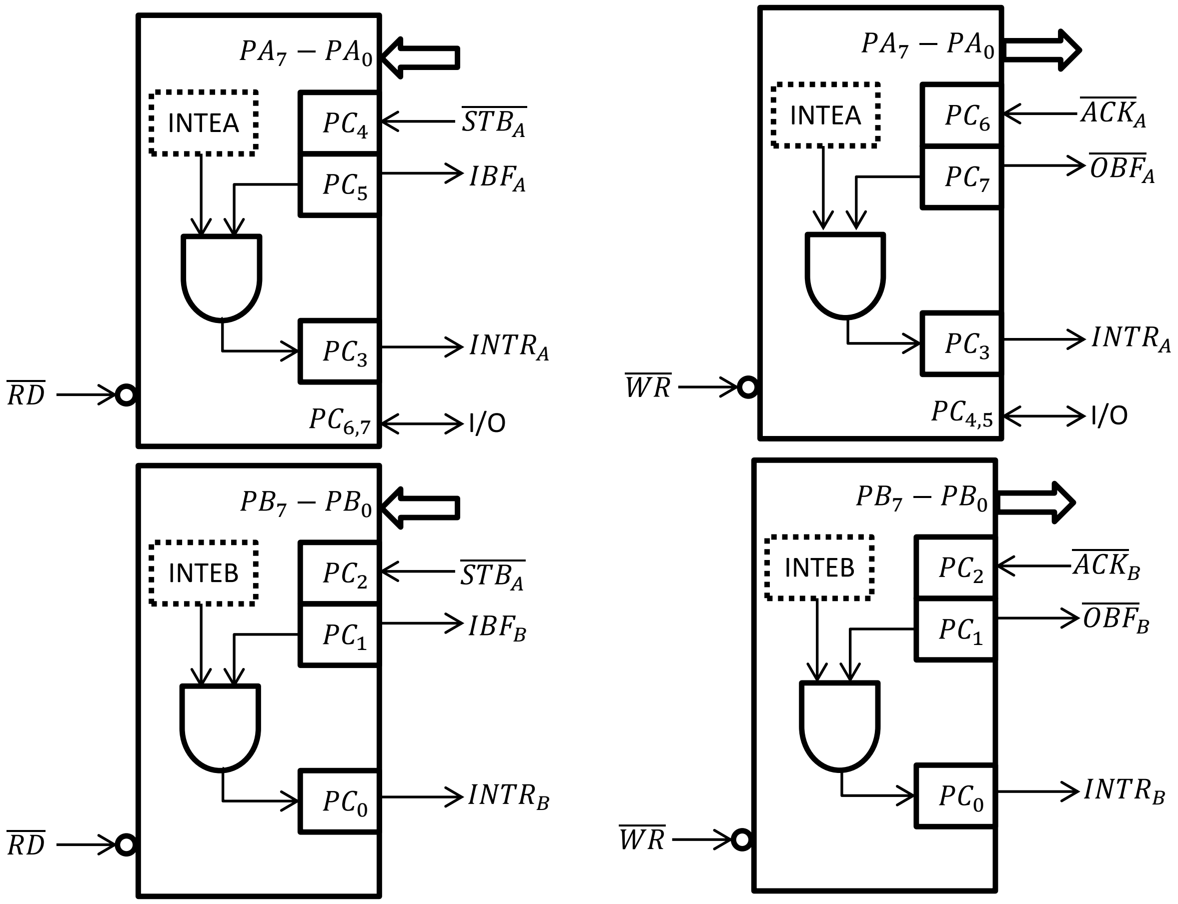
8086 Flags															
D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
X	X	X	X	O	D	I	T	S	Z	X	A	X	P	X	C

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	PA	$PC_{6,7}$	X	X	X

PA	$PC_{6,7}$	Port Yönü
0	0	Çıkış
1	1	Giriş

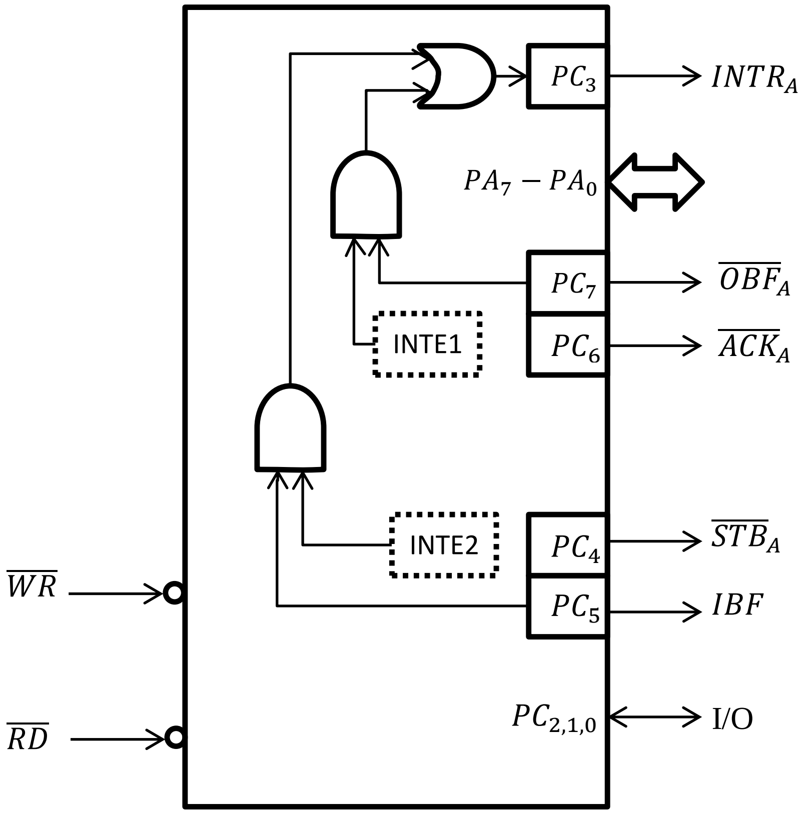
8255 Mod 1 (Grup B) Kontrol Yazmacı							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	X	X	X	X	1	PB	X

PB	Port Yönü
0	Çıkış
1	Giriş



8255 Mod 1 Input Status							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$

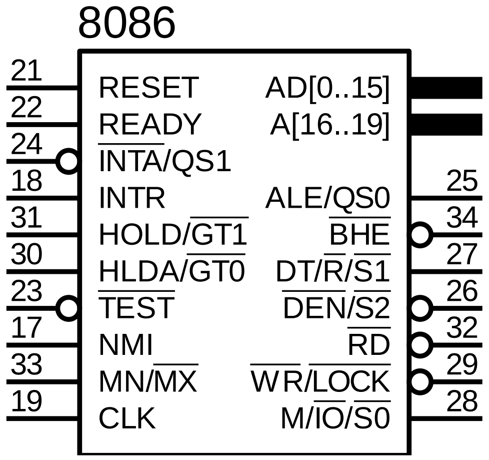
8255 Mod 1 Output Status							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\overline{OBF_A}$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF_B}$	$INTR_B$



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	Çıkış
1	Giriş

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	$INTE_2$	$INTR_A$	X	X	X



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	μP→Mode, Control, Sync

8251 Mod Yazmacı (Senkron)

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.

8251 Status Yazmacı

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
DSR	$SYNDET$	FE	OE	PE	TxE	$RxRDY$	$TxRDY$

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

8251 Mod Yazmacı (Asenkron)

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

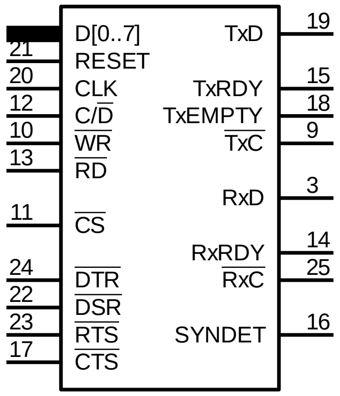
S_2	S_1	Stop biti sayısı
0	0	Invalid
0	1	1 stop biti
1	0	1.5 stop biti
1	1	2 stop biti

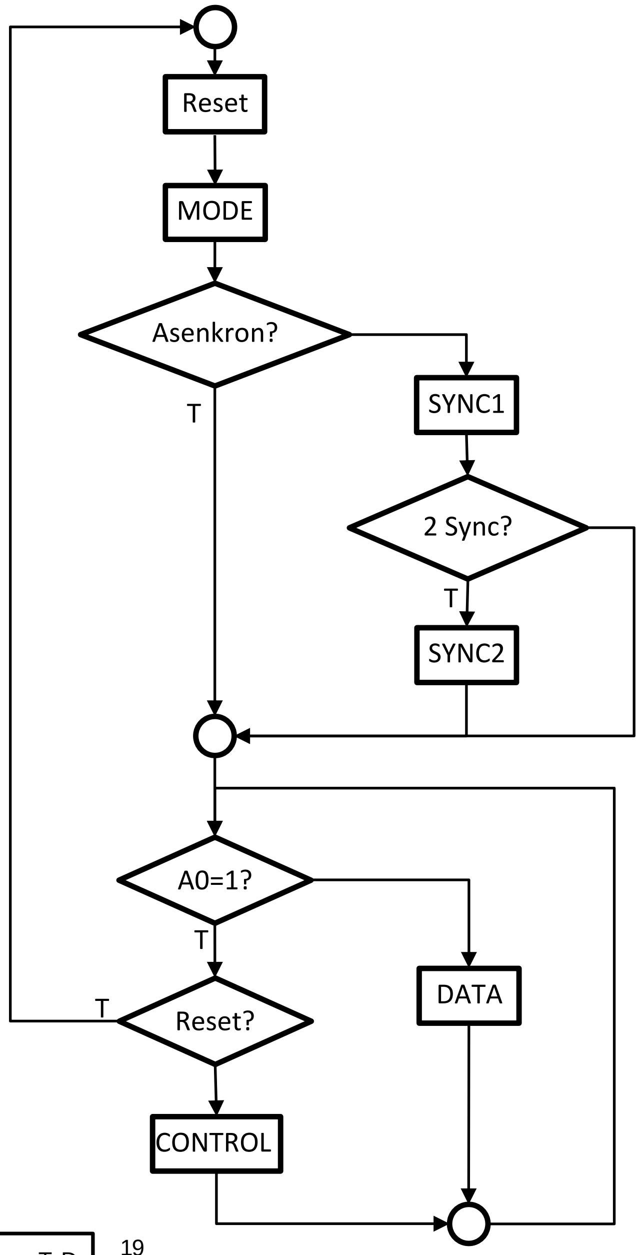
EP	Parity
0	Odd parity
1	Even parity

PEN	Parity enable
0	Parity yok
1	Parity var

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

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



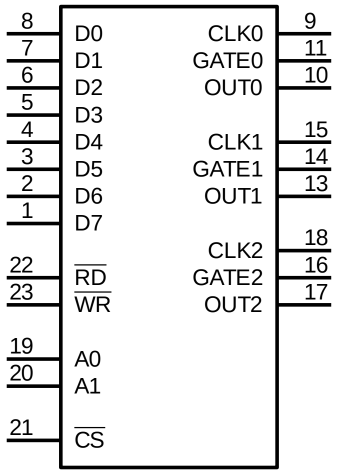


8254 Status

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

8254 Adresleme

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



8254 Kontrol Yazmacı

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC_1	SC_0	RW_1	RW_0	M_2	M_1	M_0	BCD

SC_1	SC_0	SC – Select Counter
0	0	Counter0
0	1	Counter1
1	0	Counter2
1	1	Read Back Command

M_2	M_1	M_0	M – Mod
0	0	0	Mod 0
0	0	1	Mod 1
X	1	0	Mod 2
X	1	1	Mod 3
1	0	0	Mod 4
1	0	1	Mod 5

RW_1	RW_0	RW – Read/Write
0	0	Counter Latch Command
0	1	LSb
1	0	MSb
1	1	Önce LSb, sonra MSb

BCD	Sayma
0	Binary
1	Binary Coded Decimal

8254 Read Back Command

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT_2	CNT_1	CNT_0	0

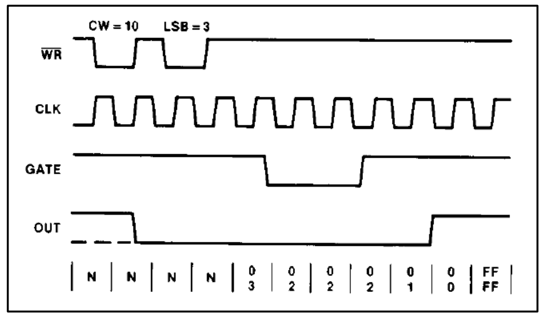
$\overline{COUNT} = 0$: Sayma değeri tut

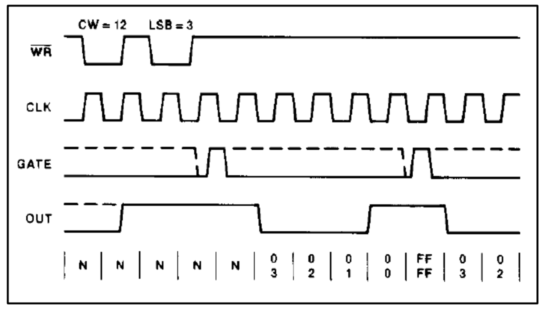
$\overline{STATUS} = 0$: Durum tut

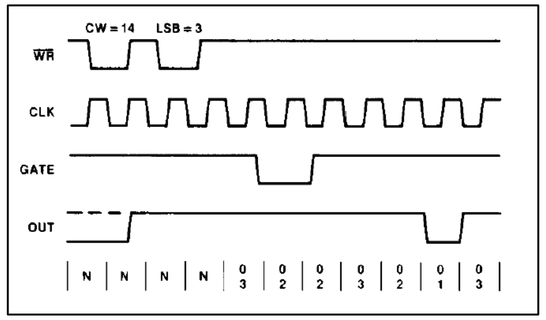
$CNT2 = 1$: Sayıcı 2 için tut

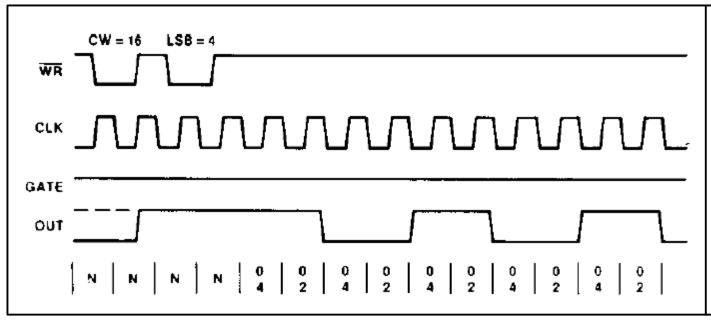
$CNT1 = 1$: Sayıcı 1 için tut

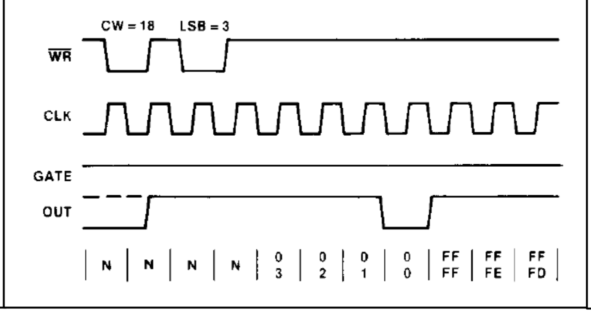
$CNT0 = 1$: Sayıcı0 için tut

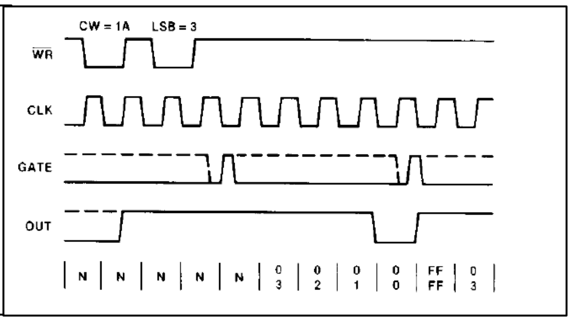












ADD	ADD destination, source Addition	Flags	O	D	I	T	S	Z	A	P	C
			x				x	x	x	x	x

Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	ADD CX, DX
register, memory	9+EA	1	2-4	ADD DI, [BX]
memory, register	16+EA	2	2-4	ADD TEMP, CL
register, immediate	4	-	3-4	ADD CL, 2
memory, immediate	17+EA	2	3-6	ADD ALPHA, 2
accumulator, immediate	4	-	2-3	ADD AX, 200

AND	AND destination, source Logical and	Flags	O	D	I	T	S	Z	A	P	C
			0				x	x	u	x	0

Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	AND AL, BL
register, memory	9+EA	1	2-4	AND CX, FLAG_WORD
memory, register	16+EA	2	2-4	AND ASCII [DI], AL
register, immediate	4	-	3-4	ND CX, 0F0H
memory, immediate	17+EA	2	3-6	AND BETA, 01H
accumulator, immediate	4	-	2-3	AND AX, 01010000B

CALL	CALL target Call a procedure	Flags	O	D	I	T	S	Z	A	P	C

Operands	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
memptr32	37+EA	4	2-4	CALL FAR PTR [BX]

CLC	CLC (no operands) Clear carry flag	Flags	O	D	I	T	S	Z	A	P	C
											0

Operands	Clocks	Transfers	Bytes	Coding Example
no operands	2	-	1	CLC

CLI	CLI (no operands) Clear interrupt flag	Flags	O	D	I	T	S	Z	A	P	C
					0						

Operands	Clocks	Transfers	Bytes	Coding Example
no operands	2	-	1	CLI

CMP	CMP destination, source Compare destination to source	Flags	O	D	I	T	S	Z	A	P	C
			x				x	x	x	x	x

Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	CMP BX, CX
register, memory	9+EA	1	2-4	CMP DH, ALPHA
memory, register	9+EA	1	2-4	CMP [BX+2], SI
register, immediate	4	-	3-4	CMP BL, 02H
memory, immediate	10+EA	1	3-6	CMP TABLE[BX+2000], 3420H
accumulator, immediate	4	-	2-3	CMP AL, 00010000B

DIV	DIV source Division, unsigned	Flags	O	D	I	T	S	Z	A	P	C
			u				u	u	u	u	u

Operands	Clocks	Transfers	Bytes	Coding Example
reg8	80-90	-	2	DIV CL
reg16	144-162	-	2	DIV BX
mem8	(86-96)+EA	1	2-4	DIV ALPHA
mem16	(150-168)+EA	1	2-4	DIV TABLE [SI]

IN	IN accumulator, port Input byte or word	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
accumulator, immed8	10	1	2	IN AL, 0FFEAH
accumulator, DX	8	1	1	IN AX, DX

INC	INC destination Increment by 1	Flags	O	D	I	T	S	Z	A	P	C
			x				x	x	x	x	x

Operands	Clocks	Transfers	Bytes	Coding Example
reg16	2	-	1	INC CX
reg8	3	-	2	INC BL
memory	15+EA	2	2-4	INC ALPHA[DI+BX]

INT	INT interrupt-type Interrupt	Flags	O	D	I	T	S	Z	A	P	C
					0	0					

Operands	Clocks	Transfers	Bytes	Coding Example
immed8 (type=3)	52	5	1	INT 3
immed8 (type≠3)	51	5	2	INT 67

IRET	IRET (no operands) Interrupt return	Flags	O	D	I	T	S	Z	A	P	C
			r	r	r	r	r	r	r	r	r

Operands	Clocks	Transfers	Bytes	Coding Example
no operands	24	3	1	IRET

JC	JC short-label Jump if carry	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
short-label	16 or 4	-	2	JC CARRY-SET

JE/JZ	JE/JZ short-label Jump if equal / Jump if zero	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
short-label	16 or 4	-	2	JZ ZERO

JMP	JMP target Jump	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
short-label	15	-	2	JMP SHORT
near-label	15	-	3	JMP WITHIN_SEGMENT
far-label	15	-	5	JMP FAR_LABEL
memptr16	18+EA	1	2-4	JMP [BX]
regptr16	11	-	2	JMP CX
memptr32	24+EA	2	2-4	JMP FAR [BX+123H]

LAHF	LAHF (no operands) Load AH from flags	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
no operands	4	-	1	LAHF

LEA	LEA destination, source Load effective address	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
reg16, mem16	2+EA	-	2-4	LEA BX, [BP+DI]

LOOP	LOOP short-label Loop	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
short-label	17/5	-	2	LOOP AGAIN

MUL	MUL source Multiplication, unsigned	Flags	O	D	I	T	S	Z	A	P	C
			x				u	u	u	u	x

Operands	Clocks	Transfers	Bytes	Coding Example
reg8	70-77	-	2	MUL BL
reg16	118-133	-	2	MUL CX
mem8	(76-83)+EA	1	2-4	MUL MONTH[SI]
mem16	(124-139)+EA	1	2-4	MUL BAUD_RATE

MOV	MOV destination, source Move	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
memory, accumulator	10	1	3	MOV ARRAY[SI], AL
accumulator, memory	10	1	3	MOV AX, TEMP_RESULT
register, register	2	-	2	MOV AX, CX
register, memory	8+EA	1	2-4	MOV BP, STACK_TOP
memory, register	9+EA	1	2-4	MOV COUNT[DI], CX
register, immediate	4	-	2-3	MOV CL, 2
memory, immediate	10+EA	1	3-6	MOV MASK[BX+SI], 2CH
seg-reg, reg16	2	-	2	MOV ES, CX
seg-reg, mem16	8+EA	1	2-4	MOV DS, SEGMENT_BASE
reg16, seg-reg	2	-	2	MOV BP, SS
memory, seg-reg	9+EA	1	2-4	MOV DATA2, CS

OR	OR destination, source Logical inclusive or	Flags	O	D	I	T	S	Z	A	P	C
			0				x	x	u	x	0

Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	OR AL, BL
register, memory	9+EA	1	2-4	OR DX, PORT_ID[DI]
memory, register	16+EA	2	2-4	OR FLAG_BYTE, CL
accumulator, immediate	4	-	2-3	OR AL, 01101100B
register, immediate	4	-	3-4	OR CX, 01H
memory, immediate	17+EA	2	3-6	OR [BX+123H], 10CFH

OUT	OUT port, accumulator Output byte or word	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
immed8, accumulator	10	1	2	OUT 44, AX
DX, accumulator	8	1	1	OUT DX, AL

POP	POP destination Pop word off stack	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
register	8	1	1	POP DX
seg-reg (CS illegal)	8	1	1	POP DS
memory	17+EA	2	2-4	POP PARAMETER

PUSH	PUSH source Push word onto stack	Flags	O	D	I	T	S	Z	A	P	C

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

PUSHF	PUSHF (no operands) Push flags onto stack	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
no operand	10	1	1	PUSHF

RCL	RCL destination, count Rotate left through carry	Flags	O	D	I	T	S	Z	A	P	C
			x								x

Operands	Clocks	Transfers	Bytes	Coding Example
register, 1	2	-	2	RCL CX, 1
register, CL	8+4*bit	-	2	RCL AL, CL
memory, 1	15+EA	2	2-4	RCL ALPHA, 1
memory, CL	20+EA+4*bit	2	2-4	RCL [BP+2], CL

RET	RET optional-pop-value Return from procedure	Flags	O	D	I	T	S	Z	A	P	C

Operands	Clocks	Transfers	Bytes	Coding Example
(intra-segment, no pop)	8	1	1	RET
(intra-segment, pop)	12	1	3	RET 4
(inter-segment, no pop)	18	2	1	RET
(inter-segment, pop)	17	2	3	RET 2

ROL	ROL destination, count Rotate left	Flags	O	D	I	T	S	Z	A	P	C
			x								x

Operands	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+EA+4*bit	2	2-4	ROL ALPHA, CL

ROR	ROR destination, count Rotate right	Flags	O	D	I	T	S	Z	A	P	C
			x								x

Operands	Clocks	Transfers	Bytes	Coding Example
register, 1	2	-	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

SAHF	SAHF (no operands) Store AH into flags	Flags	O	D	I	T	S	Z	A	P	C
							r	r	r	r	r

Operands	Clocks	Transfers	Bytes	Coding Example
no operand	4	-	1	SAHF

SAL/SHL	SAL/SHL destination, count Shift arithmetic left/Shift logical left	Flags	O	D	I	T	S	Z	A	P	C
			x								x

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
memory, CL	20+EA+4*bit	2	2-4	SAL STORE_COUNT, CL

STC	STC (no operands) Set carry flag	Flags	O	D	I	T	S	Z	A	P	C
											1

Operands	Clocks	Transfers	Bytes	Coding Example
no operand	2	-	1	STC

STI	STI (no operand) Set interrupt enable flag	Flags	O	D	I	T	S	Z	A	P	C
					1						

Operands	Clocks	Transfers	Bytes	Coding Example
no operand	2	-	1	STI

SUB	SUB destination, source Subtraction	Flags	O	D	I	T	S	Z	A	P	C
			x				x	x	x	x	x

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

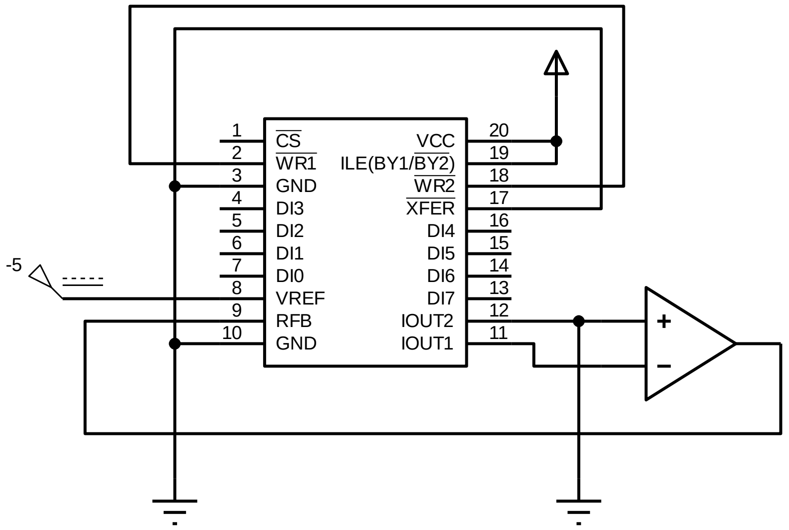
XOR	XOR destination, source Logical exclusive or	Flags	O	D	I	T	S	Z	A	P	C
			0				x	x	u	x	0

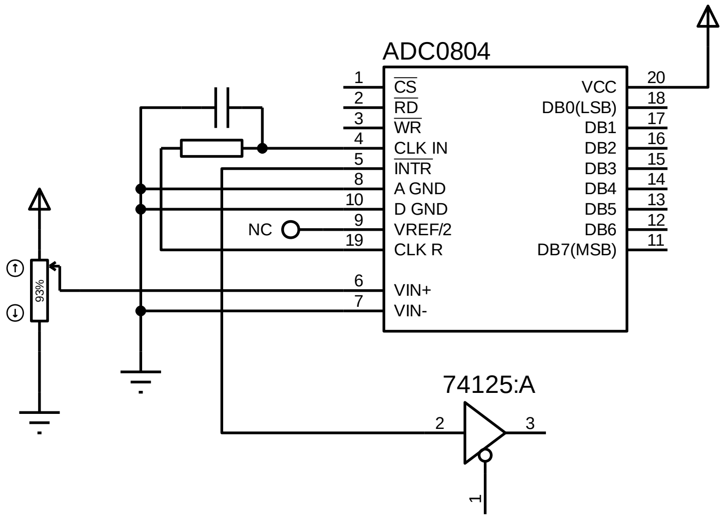
Operands	Clocks	Transfers	Bytes	Coding Example
register, register	3	-	2	XOR CX, BX
register, memory	9+EA	1	2-4	XOR CL, MASK_BYTE
memory, register	16+EA	2	2-4	XOR ALPHA[SI], DX
accumulator, immediate	4	-	2-3	XOR AL, 01000010B
register, immediate	4	-	3-4	XOR SI, 00C2H
memory, immediate	17+EA	2	3-6	XOR RETURN_CODE, 0D2H

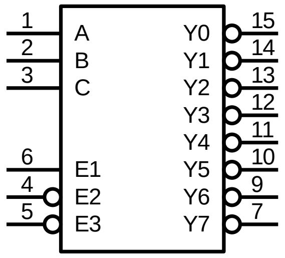
74138 3x8 Dekoder Fonksiyon Tablosu

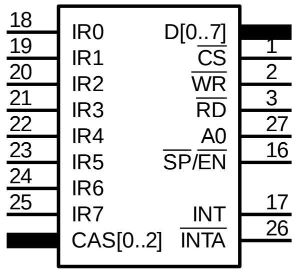
INPUTS						OUTPUTS								SELECTED OUTPUT
ENABLE			SELECT											
E1	$\overline{E2}$	$\overline{E3}$	C	B	A	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$	
L	X	X	X	X	X	H	H	H	H	H	H	H	H	NONE
X	X	H	X	X	X	H	H	H	H	H	H	H	H	NONE
X	H	X	X	X	X	H	H	H	H	H	H	H	H	NONE
H	L	L	L	L	L	L	H	H	H	H	H	H	H	$\overline{Y0}$
H	L	L	L	L	H	H	L	H	H	H	H	H	H	$\overline{Y1}$
H	L	L	L	H	L	H	H	L	H	H	H	H	H	$\overline{Y2}$
H	L	L	L	H	H	H	H	H	L	H	H	H	H	$\overline{Y3}$
H	L	L	H	L	L	H	H	H	H	L	H	H	H	$\overline{Y4}$
H	L	L	H	L	H	H	H	H	H	H	L	H	H	$\overline{Y5}$
H	L	L	H	H	L	H	H	H	H	H	H	L	H	$\overline{Y6}$
H	L	L	H	H	H	H	H	H	H	H	H	H	L	$\overline{Y7}$

X : Don't Care, L : Low, H : High









8259 ICW_1

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	1	LTIM	0	SNGL	IC_4

LTIM	Açıklama
0	Kenar tetikleme
1	Seviye tetikleme

SNGL	Açıklama
0	Kaskat bağlı 8259'lar
1	Tek 8259

IC_4	Açıklama
0	IC_4 kullanılmayacak
1	IC_4 kullanılacak

8259 ICW_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	X	X	X

$(A_7A_6A_5A_4A_3000)_2$ IR0 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	Açıklama
0	IR_i'ye slave bağlı değil
1	IR_i'ye slave bağlı

8259 ICW_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

$(ID_2ID_1ID_0)_2$ 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	SFNM	BUF	M/S	AEOI	μP

BUF	*M/S*	*Buffered – Master/Slave*
0	X	Non-buffered
1	0	Buffered slave
1	1	Buffered master

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_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

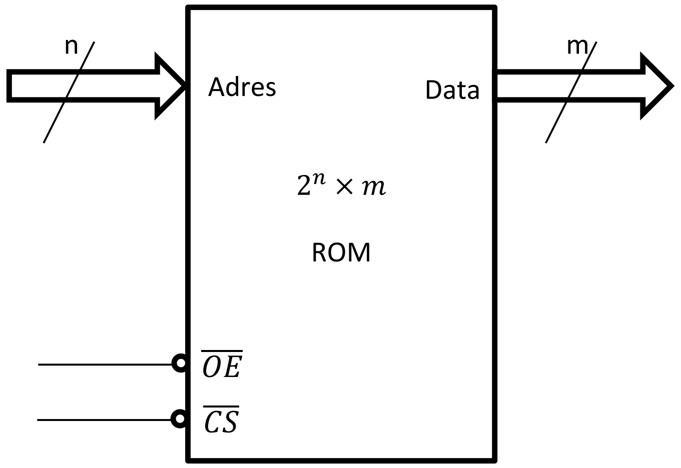
M_i	Açıklama
0	Mask reset
1	Mask set

8259 OCW_2

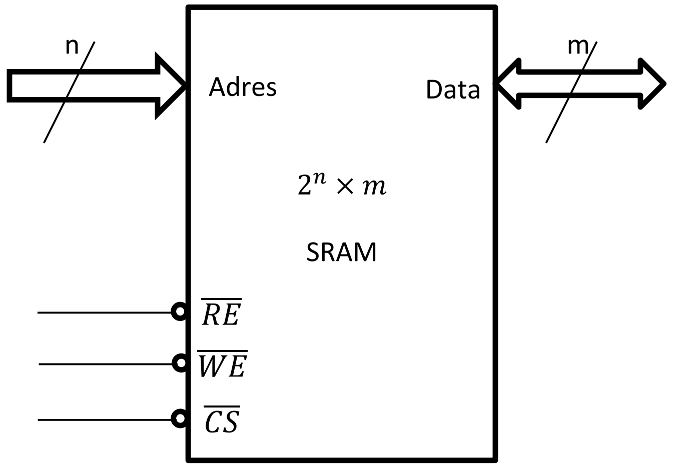
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	R	SL	EOI	0	0	L_2	L_1	L_0

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	P	RR	RIS



n
m
Adres
Data
$2^n \times m$
ROM
$\overline{OE}$
$\overline{CS}$



n
m
Adres
Data
$2^n \times m$
SRAM
$\overline{RE}$
$\overline{WE}$
$\overline{CS}$