NEC ELECTRONICS (EUROPE) GMBH

USER'S MANUAL

SE-78C06

μCOM-87 SERIES

11/82 V1.0

CONTENTS

CHAPTER 1 GENERAL

The SE-78C06 is incorporated into the evaluation system during development of the 1-chip microcomputer μPD78C06G program. The program developed by EVAKIT-87LC is written into PROM (μPD2732D) and this PROM is mounted in SE-78C06; thus, the practical operation of the program can be checked.

Features

- A 4K-byte user memory can be mounted.
  The maximum size of the external memory to be accessed is 60K bytes.
- A port E simulation circuit is mounted to provide the same functions as those of μPD78C06G (mask ROM).
- Build-in oscillation circuit
  The standby function can be evaluated.
- The same connector as that of EVAKIT-87LC is used to facilitate connection to the user system.

CHAPTER 2 SYSTEM CONFIGURATION

2.1 System Configuration

Figure 1.1 shows the SE-78C06 configuration.

(1) Evaluation chip

The μPD78C05G is mounted as an evaluation chip for μPD78C06G.

(2) Power on reset circuit

The SE-78C06 can be reset by inputting a reset signal from the reset input terminal at power on operation.

(3) Oscillator

A 4 MHz crystal oscillator is mounted. It can be replaced with another crystal oscillator having another frequency.

(4) PROM for user ROM

The μPD78C06G has a 4K-byte mask ROM. The SE-78C06 can mount a memory equivalent to this 4K-byte mask ROM as a PROM (μPD2732D).

The PROM is allocated to addresses 0 to 4095 of μPD78C05G.

(5) Port E operation instruction decode circuit

Port E of μPD78C06G can be set for two modes: address bus and latch modes.

There are PEX and PER instruction as port E operation instructions and their operations are analyzed by decode

circuit and a decoded signal is output to the port E emulation circuit.

(6) Port E emulation circuit

Port E of μPD78C05 (evaluation chip), has only an address bus mode, that is, an external latch is provided to emulate the latch mode of port E.
Switching between the address bus and latch modes is made according to the decode signal from the port E operation instruction decode circuit.

(7) 2 WAIT insertion circuit

2 WAIT of μPD78C06G is automatically inserted to access the internal ROM.
For SE-78C06, 2 WAIT is inserted from the external circuit when an address (0 - 4095) corresponding to the internal ROM address is accessed; thus, realizing a timing equivalent to μPD78C06G.

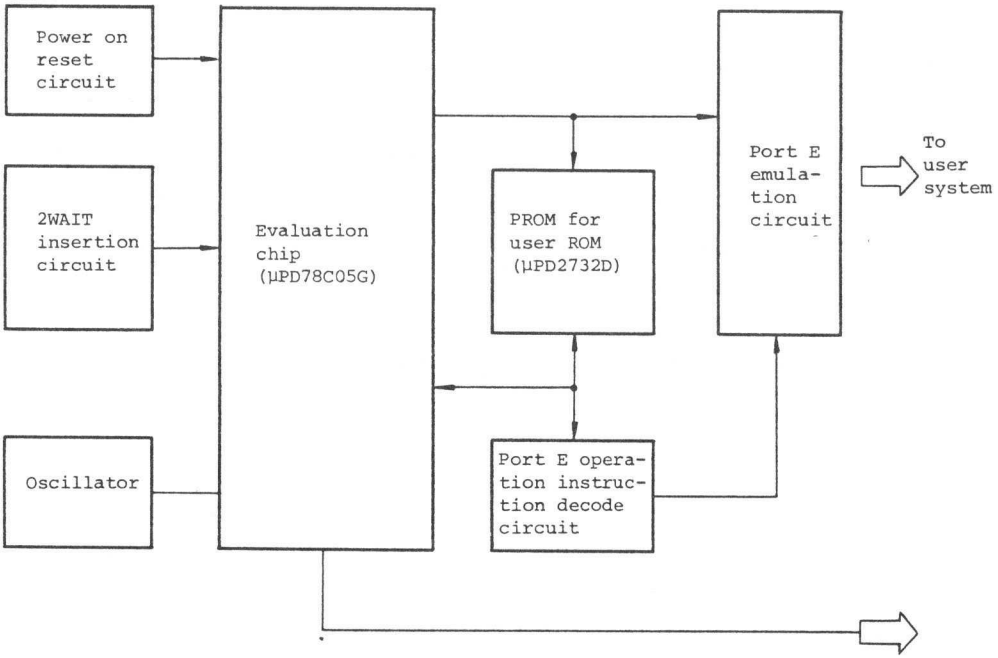


Fig. 1.1 SE-78C06 configuration

2.2 SE-78C06 Hardware Specifications

Item	Description	
Major ICs	Evaluation chip	μPD78C05G x 1 (Socket mounting)
	ROM equivalent to internal ROM	μPD2732D x 1 (Socket mounting)
	IC for instruction decoder	μPB426D x 1 (Mounted in Socket)
	Port E emulation	μPB8282C x 2
	Data bus	μPB8286C x 1
Input/output	• Header type flat cable connector (34-pin) x 2 • Connection cable x 2	
Environmental conditions	Operating conditions: 0 - 40°C, 10 - 90% RH Storage conditions : -10 to +50°C, 10 - 90% RH	
Power requirement	+5 VDC ±5%, 0.5 A max.	
Outside dimensions	135 x 180 mm	

CHAPTER 3 USAGE

3.1 User System Connection

The SE-78C06 is connected to the user system with two flat cable connectors and connection cables.

Figure 3.1 shows the connection outline. Table 3.1 lists the connector pins. Figure 3.2 shows the connector pin arrangement.

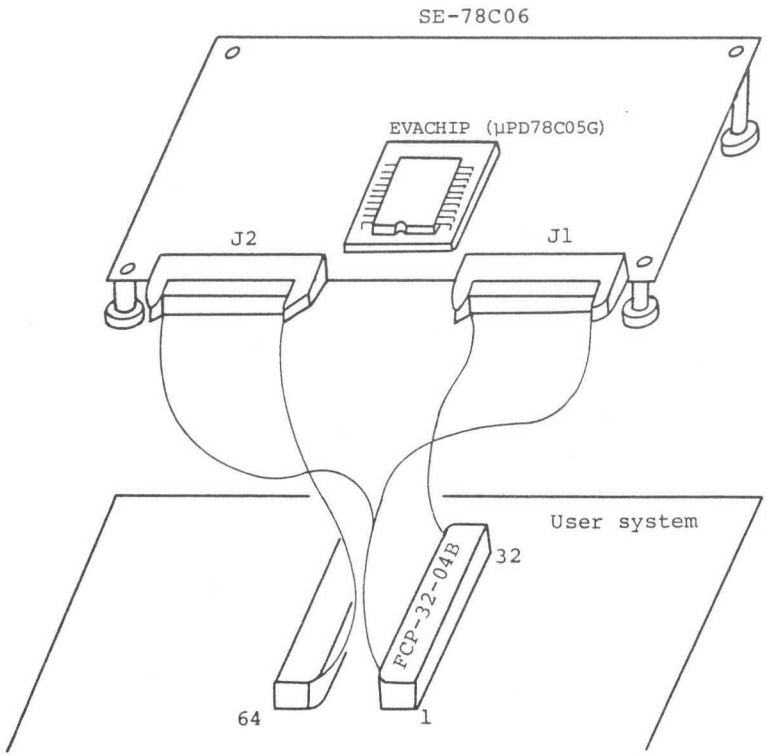


Fig. 3.1 Outline of connection to user system

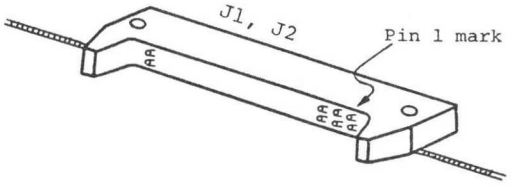
Table 3.1 Connector pins

Signal name	No.		Signal name
GND	1	18	$\overline{WR}$
-	2	19	$\overline{RD}$
PE15	3	20	PC5
ØOUT	4	21	PC4
DB7	5	22	PC3
DB6	6	23	PC2
DB5	7	24	PC1
DB4	8	25	PC0
DB3	9	27	REL
DB2	10	26	T0
DB1	11	28	SCK
DB0	12	29	SI
-	13	30	SO
ITN1	14	31	$\overline{RESET}$
INT0	15	32	-
$\overline{WAIT}$	16	33	-
(M_1)	17	34	GND

J1

Signal name	No.		Signal name
GND	1	18	PB7
Vcc	2	19	PB6
PE14	3	20	PB5
PE13	4	21	PB4
PE12	5	22	PB3
PE11	6	23	PB2
PE10	7	24	PB1
PE9	8	25	PB0
PE8	9	26	PA7
PE7	10	27	PA6
PE6	11	28	PA5
PE5	12	29	PA4
PE4	13	30	PA3
PE3	14	31	PA2
PE2	15	32	PA1
PE1	16	33	PA0
PE0	17	34	-

J2



Pin 1 is just below ▽ mark. Pin 2 is under pin 1.
Pin 3 is on the left on pin 1. Pin 4 is under pin 3.
The subsequent pins are arranged in this order.

Fig. 3.2 Connector pin arrangement

J1, J2: 34-pin flat cable connectors (Yamaichi FAP-34-03)

P1, P2: 32-pin connectors (Yamaichi FCP-32-04A/B)

Pin connectors at the end of the attached flat cable are listed:

FCP-32-04B for J1

FCP-32-04A for J2

If this pin connector is connected to the user system as shown in Figure 3.3, the pin arrangement is the same as that of the μPD78C05G.

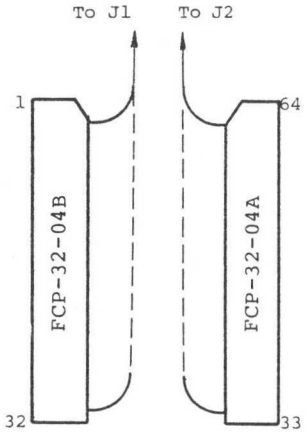


Fig. 3.3 Pin connector connection

3.2 User Program Execution

Since the SE-7806 has an internal power on reset circuit, the user program can be executed from address 0 by supplying power to the power terminal.

The user program can be executed from address 0 by input of a reset signal from the external, because connector J1 has a reset input pin.

If an address area other than the PROM (μPD2732D) of the SE-78C06 is accessed, the μPB8286C for the data bus becomes active, and an instruction, or data is fetched from the prototype system.

When the internal RAM address area (addresses of 65408 to 65535) is accessed, the data bus pin is set to Hi-Z (high impedance) and $\overline{RD}$ and $\overline{WR}$ pins are set to 1 (high).

APPENDIX A μCOM-87LC INSTRUCTION SET

Operand representation format/description method

Representation format	Description method
r	A, B, C, D, E, H, L
r1	B, C, D, E, H, L
r2	A, B, C
sr	PA PB MK MB TM S TMM SM SC
sr1	PA PB PC MK S TMM SC
sr2	PA PB PC MK
rp	SP, B, D, H
rp1	V, B, D, H
rpa	B, D, H, D+, H+, D-, H-
wa	8 bit immediate data
word	16 bit immediate data
byte	8 bit "
bit	3 bit "
f	F0, F1, FT, FS,

Remarks: 1. In sr to sr2, PA = PORTA, PB = PORTB, PC = PORTC, MK = MASK.reg, MB = MODE.B, TM = TIMER.REG, S = SERIAL I/O, TMM = TIMER MODE REG, SM = SERIAL MODE REG, SC = STANDBY CONTROL REG

2. In rp to rp1, SP = STACK POINTER, B = BC, D = DE, H = HL, V = FFH.A°

3. In rpa, B = (BC), D = (DE, H = (HL), D+ = $(DE)^+$ H+ = $(HL)^+$, D- = $(DE)^-$, H- = $(HL)^-$

4. In f, F0 = INTF0, F1 = INTF1, FT = INTFT, FS - INTFS

Instruction group	Mnemonic	Operand	Byte	Cycle clock	Operation	Skip condition	Flag	
							CY	Z
8-bit data transfer instruction	MOV	r1,A	1	4/6	r1←A			
	MOV	A,r1	1	4/6	A←r1			
	MOV	sr,A	2	10/14	sr←A			
	MOV	A,sr1	2	10/14	A←sr1			
	MOV	r,word	4	17/25	r←(word)			
	MOV	word,r	4	17/25	(word)←r			
	MVI	r,byte	2	7/11	r←byte			
	STAW	wa	2	10/14	(FFH.wa)←A			
	LDAW	wa	2	10/14	A←(FFH.wa)			
	STAX	rpa	1	7/9	(rpa)←A			
	LDAX	rpa	1	7/9	A←(rpa)			
16-bit data transfer instruction	SBCD	word	4	20/28	(word)←C,(word+1)←B			
	SDED	word	4	20/28	(word)←E,(word+1)←D			
	SHLD	word	4	20/28	(word)←L,(word+1)←H			
	SSPD	word	4	20/28	(word)←SPL,(word+1)←SPH			
	LBCD	word	4	20/28	C←(word),B←(word+1)			
	LDED	word	4	20/28	E←(word),D←(word+1)			
	LHLD	word	4	20/28	L←(word),H←(word+1)			
	LSPD	word	4	20/28	SPL←(word),SPH←(word+1)			
	PUSH	rpl	2	17/21	(SP-1)←rplH (SP-2)←rplL			
	POP	rpl	2	14/18	rplL←(SP) rplH←(SP+1),SP←SP+2			
	LXI	rp,word	3	10/16	rp←word			
Arithmetic operation instruction	ADD	A,r	2	8/12	A←A+r		↕	↕
	ADDX	rpa	2	11/15	A←A+(rpa)		↕	↕
	ADC	A,r	2	8/12	A←A+r+CY		↕	↕
	ADCX	rpa	2	11/15	A←A+(rpa)+CY		↕	↕
	SUB	A,r	2	8/12	A←A-r		↕	↕
	SUBX	rpa	2	11/15	A←A-(rpa)		↕	↕

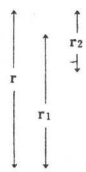
Instruction group	Mnemonic	Operand	Byte	Cycle clock	Operation	Skip condition	Flag	
							CY	Z
Arithmetic operation instruction	SBB	A,r	2	8/12	A←A-r-CY		↕	↕
	SBBX	rpa	2	11/15	A←A-(rpa)-CY		↕	↕
	ADDNC	A,r	2	8/12	A←A+r	No Carry	↕	↕
	ADDNCX	rpa	2	11/15	A←A+(rpa)	No Carry	↕	↕
	SUBNB	A,r	2	8/12	A←A-r	No Borrow	↕	↕
	SUBNBX	rpa	2	11/15	A←A-(rpa)	No Borrow	↕	↕
Logical operation instruction	ANA	A,r	2	8/12	A←A∧r			↕
	ANAX	rpa	2	11/15	A←A∧(rpa)			↕
	ORA	A,r	2	8/12	A←A∨r			↕
	ORAX	rpa	2	11/15	A←A∨(rpa)			↕
	XRA	A,r	2	8/12	A←A∨r			↕
	XRAX	rpa	2	12/15	A←A-(rpa)			↕
	GTA	A,r	2	8/12	A-r-1	No Borrow	↕	↕
	GTAX	rpa	2	11/15	A-(rpa)-1	No Borrow	↕	↕
	LTA	A,r	2	8/12	A-r	Borrow	↕	↕
	LTAX	rpa	2	11/15	A-(rpa)	Borrow		↕
	ONAX	rpa	2	8/12	A∧(rpa)	No Zero		↕
	OFFAX	rpa	2	11/15	A∧(rpa)	Zero	↕	↕
	NEA	A,r	2	8/12	A-r	No Zero	↕	↕
	NEAX	rpa	2	11/15	A-(rpa)	No Zero	↕	↕
	EQA	A,r	2	8/12	A-r	Zero	↕	↕
	EQAX	rpa	2	11/15	A-(rpa)	Zero	↕	↕
Immediate data operation instruction (accumulator)	XRI	A,byte	2	7/11	A←A∨byte			↕
	ADINC	A,byte	2	7/11	A←A+byte	No Carry	↕	↕
	SUINB	A,byte	2	7/11	A←A-byte	No Borrow	↕	↕
	ADI	A,byte	2	7/11	A←A+byte		↕	↕
	ACI	A,byte	2	7/11	A←A+byte+CY		↕	↕
	SUI	A,byte	2	7/11	A←A-byte		↕	↕

Instruction group	Mnemonic	Operand	Byte	Cycle clock	Operation	Skip condition	Flag	
							CY	Z
Immediate data operation instruction (accumulator)	SBI	A,byte	2	7/11	A←A-byte-CY			↕
	ANI	A,byte	2	7/11	A←A∧byte			↕
	ORI	A,byte	2	7/11	A←A∨byte			↕
	GTI	A,byte	2	7/11	A-byte-1	No Borrow	↕	↕
	LTI	A,byte	2	7/11	A-byte	Borrow	↕	↕
	ONI	A,byte	2	7/11	A∧byte	No Zero		↕
	OFFI	A,byte	2	7/11	A∧byte	Zero		↕
	NEI	A,byte	2	7/11	A-byte	No Zero	↕	↕
	EQI	A,byte	2	7/11	A-byte	Zero	↕	↕
Immediate data operation instruction(privileged register)	ANI	sr2,byte	3	17/23	sr2←sr2∧byte sr2:PA,PB,MK			↕
	ORI	sr2,byte	3	17/23	sr2←sr2∨byte sr2:PA,PB,MK			↕
	ONI	sr2,byte	3	14/20	sr2∧byte	No Zero		↕
	OFFI	sr2,byte	3	14/20	sr2∧byte	Zero		↕
Working register operation instruction	ANIW	wa,byte	3	16/22	(FFH.wa)←(FFH.wa)∧byte			↕
	ORIW	wa,byte	3	16/22	(FFH.wa)←(FFH.wa)∨byte			↕
	GTIW	wa,byte	3	13/19	(FFH.wa)-byte-1	No Borrow	↕	↕
	LTIW	wa,byte	3	13/19	(FFH.wa)-byte	Borrow	↕	↕
	ONIW	wa,byte	3	13/19	(FFH.wa)∧byte	No Zero		↕
	OFFIW	wa,byte	3	13/19	(FFH.wa)∧byte	Zero		↕
	NEIW	wa,bute	3	13/19	(FFH.wa)-byte	No Zero	↕	↕
	EQIW	wa,byte	3	13/19	(FFH.wa)-byte	Zero	↕	↕
Increment/decrement instruction	INR	r2	1	4/6	r2←r2+1	Carry		↕
	INRW	wa	2	13/17	(FFH.wa)←(FFH.wa)+1	Carry		↕
	DCR	r2	1	4/6	r2←r2-1	Borrow		↕
	DCRW	wa	2	13/17	(FFH.wa)←(FFH.wa)-1	Borrow		↕
	INX	rp	1	7/9	rp←rp+1			
	DCX	rp	1	7/9	rp←rp-1			

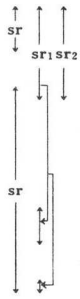
Instruction group	Mnemonic	Operand	Byte	Cycle clock	Operation	Skip condition	Flag CY	Flag Z
Other operational instructions	DAA		1	4/6	Decimal Adjust Accumulator			
	STC		2	8/12	CY←1		1	
	CLC		2	8/12	CY←0		0	
Rotation instruction	RLD		2	17/21	Rotate Left Digit			
	RRD		2	17/21	Rotate Right Digit			
	RAL		2	8/12	Am_{+1}←Am, A_0←CY, CY←A_7			
	RAR		2	8/12	Am_{-1}←Am, A_7←CY, CY←A_0			
Jump instruction	JMP	word	3	10/16	PC←word			
	JB		1	4/6	PCH←B, PCL←C			
	JR	word	1	10/12	PC←PC+1+jdjspl			
	JRE	word	2	13/17	PC←PC+2+jdisp			
Call instruction	CALL	word	3	16/22	(SP-1)←(PC+3)H, (SP-2)←(PC+3)L, PC←word			
	CALF	word	2	13/17	(SP-1)←(PC+2)H, (SP-2)←(PC+2)L PC15∿11←00001, PC10∿0←fa			
	CALT	word	1	19/21	(SP-1)←(PC+1)H, (SP-2)←(PC+1)L PCL←(128+2ta), PCH←(129+2ta)			
Return instruction	RET		1	10/12	PCL←(SP), PCH←(SP+1) SP←SP+2			
	RETS		1	10+n 12+n	PCL←(SP), PCH←(SP+1) SP←SP+2, PC←PC+n	Unconditional skip		
	RETI		1	13/15	PCL←(SP), PCH←(SP+1) PSW←(SP+2), SP←SP+3			
Skip instruction	SKNC		2	8/12	Skip if No Carry	CY=0		
	SKNZ		2	8/12	Ship if No Zero	Z=0		
	SKNIT	f	2	8/12	Skip if No INT× other wise reset INT×	f=0		
CPU control instruction	NOP		1	4/6	No Operation			
	EI		2	8/12	Enable Interrupt			
	DI		2	8/12	Disable Interrupt			
Register control instruction	SIO		1	4/6	Start(Trigger) Serial I/O			
	STM		1	4/6	Start Timer			
Output instruction	PEX		2	11/15	$PE_{15\sim8}$←B, $PE_{7\sim0}$←C			
	PER		2	8/12	PortE AB Mode			

Remarks: The clock cycle on the left of the slash is that of the external memory or internal RAM and the clock cycle on the right of the slash is thst of the internal ROM.

R_2 R_1 R_0	
0 0 0	—
0 0 1	A
0 1 0	B
0 1 1	C
1 0 0	D
1 0 1	E
1 1 0	H
1 1 1	L



S_3 S_2 S_1 S_0	
0 0 0 0	PA
0 0 0 1	PB
0 0 1 0	PC
0 0 1 1	MK
0 1 0 0	MB
0 1 0 1	—
0 1 1 0	TM
0 1 1 1	—
1 0 0 0	S
1 0 0 1	TMM
1 0 1 0	SM
1 0 1 1	SC

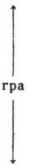


P_1 P_0	rp
0 0	SP
0 1	BC
1 0	DE
1 1	HL

Q_1 Q_0	rp_1
0 0	*
0 1	BC
1 0	DE
1 1	HL

* : FFH - A

A_2 A_1 A_0	
0 0 0	—
0 0 1	(BC)
0 1 0	(DE)
0 1 1	(HL)
1 0 0	(DE) +
1 0 1	(HL) +
1 1 0	(DE) −
1 1 1	(HL) −



I_2 I_1 I_0	INTF
0 0 0	INTF0
0 0 1	INTFT
0 1 0	INTF1
0 1 1	—
1 0 0	INTFS

Flag operation

Operation					D6	D5	D4	D3	D2	D0
reg, memory		immediate		skip	Z	SK	HC	L1	L0	CY
ADD ADC SUB SBB	ADDX ADCX SUBX SBBX	ADI ACI SUI SBI			↕	0	↕	0	0	↕
ANA ORA XRA	ANAX ORAX XRAX	ANI ORI XRI	ANIW ORIW		↕	0	●	0	0	●
ADDNC SUBNB GTA LTA	ADDNCX SUBNBX GTAX LTAX	ADINC SUINB GTI LTI	 GTIW LTIW		↕	↕	↕	0	0	↕
	ONAX OFFAX	ONI OFFI	ONIW OFFIW		↕	↕	●	0	0	●
NEA EQA	NEAX EQAX	NEI EQI	NEIW EQIW		↕	↕	↕	0	0	↕
INR DCR	INRW DCRW				↕	↕	↕	0	0	●
DAA					↕	0	↕	0	0	↕
RLL, RLR					●	0	●	0	0	↕
RLD, RRD					●	0	●	0	0	●
STC					●	0	●	0	0	1
CLC					●	0	●	0	0	0
		MVI A, byte			●	0	●	1	0	●
		MVI L, byte LXI H, word			●	0	●	0	1	●
				SKN SKNIT	●	↕	●	0	0	●
				RETS	●	1	●	0	0	●
Other instructions					●	0	●	0	0	●

↕ : Affected (set or reset)
1: Set
0: Reset
: Not affected

Operational instruction operand selection table

	A	B	C	D	E	H	L	(BC)	(DE)	(HL)	(DE)+	(HL)+	(DE)−	(HL)−	byte
V	①														③
A	①	①	①	①	①	①	①	②	②	②	②	②	②	②	③
B	①														③
C	①														③
D	①														③
E	①														③
H	①														③
L	①														③
PA															③
PB															③
PC															③
MK															③
(FFH+wa)															④

①	ADD, ADC, SUB, SBB, ADDNC, SUBNB, ANA, ORA, XRA, GTA, LTA, NEA, EQA	A, r
②	ADDX, ADCX, SUBX, SBBX, ADDNCX, SUBNBX, ANAX, ORAX, XRAX, GTAX, LTAX, ONAX, OFFAX, NEAX, EQAX	rpa
③	XRI, ADINC, SUINB, ADI, ACI, SUI, SBI, ANI, ORI, GTI, LTI, ONI, OFFI, NEI, EQI	A, byte sr2, byte
④	ANIW, ORIW, GTIW, LTIW, ONIW, OFFIW, NEIW, EQIW	wa, byte

μCOM-87LC mnemonic-machine word correspondence table (alphabetical order)

ACI	A, byte	56××
ADC	A, r	60D1～60D7
ADCX	rpa	70D1～70D7
ADD	A, r	60C1～60C7
ADDNC	A, r	60A1～60A7
ADDNCX	rpa	70A1～70A7
ADDX	rpa	70C1～70C7
ADI	A, byte	46××
ADINC	A, byte	26××
ANA	A, r	6089～608F
ANAX	rpa	7089～708F
ANI	A, byte	07××
ANI	sr_2, byte	6488××～648B××
ANIW	wa, byte	05××××
CALF	word	7800～7FFF
CALL	word	44××××
CALT	word	80～BF
CLC		482A
DAA		61
DCR	r_2	51～53
DCRW	wa	30××
DCX	rp	03, 13, 23, 33
DI		4824
EI		4820
EQA	A, r	60F9～60FF
EQAX	rpa	70F9～70FF
EQI	A, byte	77××
EQIW	wa, byte	75××××
GTA	A, r	60A9～60AF
GTAX	rpa	70A9～70AF
GTI	A, byte	27××
GTIW	wa, byte	25××××
INR	r_2	41～43
INRW	wa	20××
INX	rp	02, 12, 22, 32
JB		73
JMP	word	54××××
JR	word	C0～FF
JRE	word	4E00～4FFF
LBCD	word	701F×××
LDAW	wa	28××
LDAX	rpa	29～2F
LDED	word	702F××××
LHLD	word	703F××××
LSPD	word	700F××××
LTA	A, r	60B9～60BF
LTAX	rpa	70B9～70BF
LTI	A, byte	37××
LTIW	wa, byte	35××××
LXI	rp, word	04, 14, 24, 34××××
MOV	r_1, A	1A～1F
MOV	A, r_1	0A～0F
MOV	sr, A	4DC0, 1, 3, 4, 6, 8～B
MOV	A, sr_1	4CC0～3, 8, 9, B
MOV	r, word	7069××××～706F××××
MOV	word, r	7079××××～707F××××
MVI	r, byte	69××～6F××
NEA	A, r	60E9～60EF
NEAX	rpa	70E9～70EF
NEI	A, byte	67××
NEIW	wa, byte	65××××
NOP		00
OFFAX	rpa	70D9～70DF
OFFI	A, byte	57××
OFFI	sr_2, byte	64D8××～64DB××
OFFIW	wa, byte	55××××
ONAX	rpa	70C9～70CF
ONI	A, byte	47××
ONI	sr_2, byte	64C8××～64CB××
ONIW	wa, byte	45××××
ORA	A, r	6099～609F
ORAX	rpa	7099～709F
ORI	A, byte	17××
ORI	sr_2, byte	6498××～649B××
ORIW	wa, byte	15××××
PER		483C
PEX		482D
POP	rp_1	480F, 1F, 2F, 3F
PUSH	rp_1	480E, 1E, 2E, 3E
RET		08
RETI		62
RETS		18
RLD		4838
RLL	A	4830
RLR	A	4831
RRD		4839
SBB	A, r	60F1～60F7
SBBX	rpa	70F1～70F7
SBCD	word	701E××××
SBI	A, byte	76××
SDED	word	702E××××
SHLD	word	703E××××
SIO		09
SKN	CY	481A
SKN	Z	481C
SKNIT	irf	4810～2, 4
SSPD	word	700E××××
STAW	wa	38××
STAX	rpa	39～3F
STC		482B
STM		19
SUB	A, r	60E1～60E7
SUBNB	A, r	60B1～60B7
SUBNBX	rpa	70B1～70B7
SUBX	rpa	70E1～70E7
SUI	A, byte	66××
SUINB	A, byte	36××
XRA	A, r	6091～6097
XRAX	rpa	7091～7097
XRI	A, byte	16××

R_2	R_1	R_0	
0	0	0	—
0	0	1	A
0	1	0	B
0	1	1	C
1	0	0	D
1	0	1	E
1	1	0	H
1	1	1	L

r: r_1, r_2

S_3	S_2	S_1	S_0	
0	0	0	0	PA
0	0	0	1	PB
0	0	1	0	PC
0	0	1	1	MK
0	1	0	0	MB
0	1	0	1	—
0	1	1	0	TM
0	1	1	1	—
1	0	0	0	S
1	0	0	1	TMM
1	0	1	0	SM
1	0	1	1	SC

sr: sr, sr_1, sr_2

P_1	P_0	rp
0	0	SP
0	1	BC
1	0	DE
1	1	HL

Q_1	Q_0	rp_1
0	0	*
0	1	BC
1	0	DE
1	1	HL

*: FFH · A

A_2	A_1	A_0	
0	0	0	—
0	0	1	(BC)
0	1	0	(DE)
0	1	1	(HL)
1	0	0	(DE)+
1	0	1	(HL)+
1	1	0	(DE)−
1	1	1	(HL)−

rpa: (BC), (DE), (HL), (DE)+, (HL)+, (DE)−, (HL)−

I_2	I_1	I_0	INTF
0	0	0	INTF0
0	0	1	INTFT
0	1	0	INTF1
0	1	1	—
1	0	0	INTFS

μCOM-87 LC Correspondence between machine language and mnemonic (3/1)

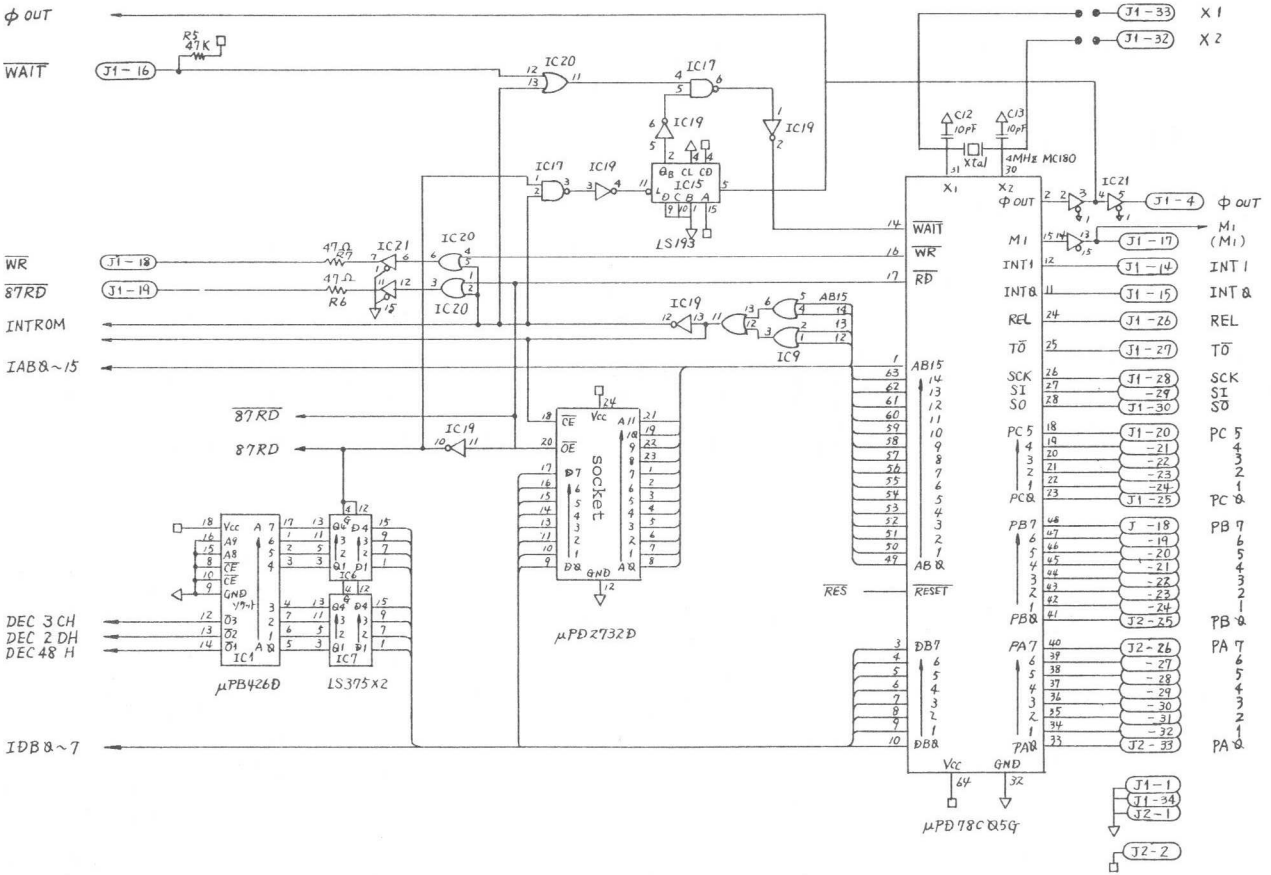
00	NOP		40			4DC3	MOV	MK, A	6034	···					
01	···		41	INR	A	4DC4	MOV	MB, A	6035	···					
02	INX	SP	42	INR	B	4DC5	···		6036	···					
03	DCX	SP	43	INR	C	4DC6	MOV	TM, A	6037	···					
04	LXI	SP, word	44	CALL	word	4DC7	···		6038	···					
05	ANIW	wa, byte	45	ONIW	wa, byte	4DC8	MOV	S, A	6039	···					
06			46	ADI	A, byte	4DC9	MOV	TMM, A	603A	···					
07	ANI	A, byte	47	ONI	A, byte	4DCA	MOV	SM, A	603B	···					
08	RET		4800	···		4DCB	MOV	SC, A	603C	···					
09	SIO		4801	···		4E00	JRE	word	603D	···					
0A	MOV	A, B	4802	···					603E	···					
0B	MOV	A, C	4803	···		4FFF			603F	···					
0C	MOV	A, D	4804	···		50			6040	···					
0D	MOV	A, E	480A	···		51	DCR	A	6041	···					
0E	MOV	A, H	480C	···		52	DCR	B	6042	···					
0F	MOV	A, L	480E	PUSH	V	53	DCR	C	6043	···					
10	···		480F	POP	V	54	JMP	word	6044	···					
11	···		4810	SKNIT	F0	55	OFFIW	wa, byte	6045	···					
12	INX	B	4811	SKNIT	FT	56	ACI	A, byte	6046	···					
13	DCX	B	4812	SKNIT	F1	57	OFFI	A, byte	6047	···					
14	LXI	B, word	4813	···		58	···		6050	···					
15	ORIW	wa, byte	4814	SKNIT	FS	59	···		6051	···					
16	XRI	A, byte	481A	SKN	CY	5A	···		6052	···					
17	ORI	A, byte	481C	SKN	Z	5B	···		6053	···					
18	RETS		481E	PUSH	B	5C	···		6054	···					
19	STM		481F	POP	B	5D	···		6055	···					
1A	MOV	B, A	4820	EI		5E	···		6056	···					
1B	MOV	C, A	4824	DI		5F	···		6057	···					
1C	MOV	D, A	482A	CLC		6008	···		6060	···					
1D	MOV	E, A	482B	STC		6009	···		6061	···					
1E	MOV	H, A	482C	···		600A	···		6062	···					
1F	MOV	L, A	482D	PEX		600B	···		6063	···					
20	INRW	wa	482E	PUSH	D	600C	···		6064	···					
21	···		482F	POP	D	600D	···		6065	··					
22	INX	D	4830	RLL	A	600E	···		6066	···					
23	DCX	D	4831	RLR	A	600F	···		6067	···					
24	LXI	D, word	4832	···		6010	···		6068	···					
25	GTIW	wa, byte	4833	···		6011	···		6069	···					
26	ADINC	A, byte	4834	···		6012	···		606A	···					
27	GTI	A, byte	4835	···		6013	···		606B	···					
28	LDAW	wa	4836	···		6014	···		606C	···					
29	LDAX	B	4837	···		6015	···		606D	···					
2A	LDAX	D	4838	RLD		6016	···		606E	···					
2B	LDAX	H	4839	RRD		6017	···		606F	···					
2C	LDAX	D+	483C	PER		6018	···		6070	···					
2D	LDAX	H+	483E	PUSH	H	6019	···		6071	···					
2E	LDAX	D−	483F	POP	H	601A	···		6072	···					
2F	LDAX	H−	49	···		601B	···		6073	···					
30	DCRW	wa	4A	···		601C	···		6074	···					
31	···		4B	···		601D	···		6075	···					
32	INX	H	4C00	···		601E	···		6076	···					
33	DCX	H	⋮			601F	···		6077	···					
34	LXI	H, word	4CBF			6020	···		6078	···					
35	LTIW	wa, byte	4CC0	MOV	A, PA	6021	···		6079	···					
36	SUINB	A, byte	4CC1	MOV	A, PB	6022	···		607A	···					
37	LTI	A, byte	4CC2	MOV	A, PC	6023	···		607B	···					
38	STAW	wa	4CC3	MOV	A, MK	6024	···		607C	···					
39	STAX	B	4CC8	MOV	A, S	6025	···		607D	···					
3A	STAX	D	4CC9	MOV	A, TMM	6026	···		607E	···					
3B	STAX	H	4CCB	MOV	A, SC	6027	···		607F	···					
3C	STAX	D+	4D00	···		6030	···		6088	···					
3D	STAX	H+	4DC0	MOV	PA, A	6031	···		6089	ANA	A, A				
3E	STAX	D−	4DC1	MOV	PB, A	6032	···		608A	ANA	A, B				
3F	STAX	H−	4DC2	···		6033	···		608B	ANA	A, C				

µCOM-87 LC Correspondence between machine language and mnemonoc (3/2)

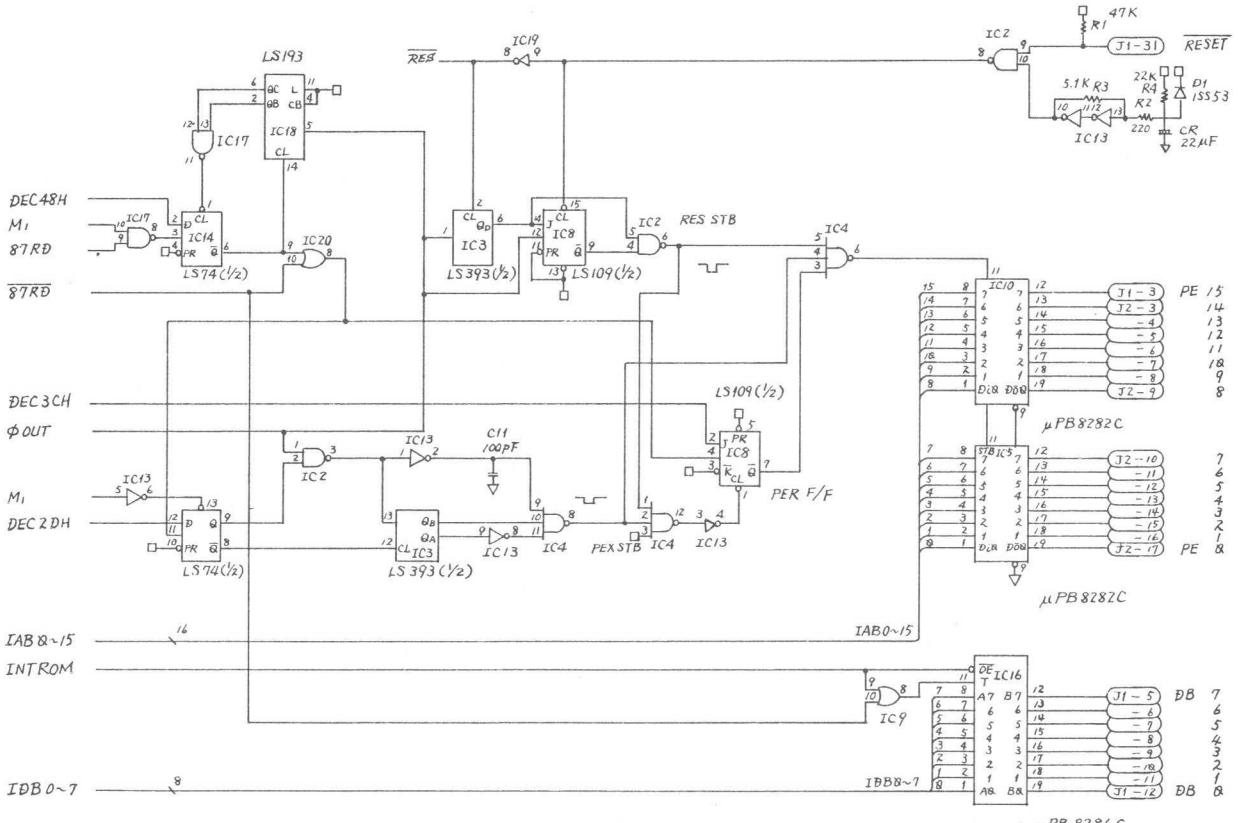
608C	ANA	A, D	60CC	···		6411	···	6451	···		
608D	ANA	A, E	60CD	···		6412	···	6452	···		
608E	ANA	A, H	60CE	···		6413	···	6453	···		
608F	ANA	A, L	60CF	···		6414	···	6454	···		
6090	···		60D0	···		6415	···	6455	···		
6091	XRA	A, A	60D1	ADC	A, A	6416	···	6456	···		
6092	XRA	A, B	60D2	ADC	A, B	6417	···	6457	···		
6093	XRA	A, C	60D3	ADC	A, C	6418	···	6458	···		
6094	XRA	A, D	60D4	ADC	A, D	6419	···	6459	···		
6095	XRA	A, E	60D5	ADC	A, E	641A	···	645A	···		
6096	XRA	A, H	60D6	ADC	A, H	641B	···	645B	···		
6097	XRA	A, L	60D7	ADC	A, L	641C	···	645C	···		
6098	···		60D8	···		641D	···	645D	···		
6099	ORA	A, A	60D9	···		641E	···	645E	···		
609A	ORA	A, B	60DA	···		641F	···	645F	···		
609B	ORA	A, C	60DB	···		6420	···	6460	···		
609C	ORA	A, D	60DC	···		6421	···	6461	···		
609D	ORA	A, E	60DD	···		6422	···	6462	···		
609E	ORA	A, H	60DE	···		6423	···	6463	···		
609F	ORA	A, L	60DF	···		6424	···	6464	···		
60A0	···		60E0	···		6425	···	6465	···		
60A1	ADDNC	A, A	60E1	SUB	A, A	6426	···	6466	···		
60A2	ADDNC	A, B	60E2	SUB	A, B	6427	···	6467	···		
60A3	ADDNC	A, C	60E3	SUB	A, C	6428	···	6468	···		
60A4	ADDNC	A, D	60E4	SUB	A, D	6429	···	6469	···		
60A5	ADDNC	A, E	60E5	SUB	A, E	642A	···	646A	···		
60A6	ADDNC	A, H	60E6	SUB	A, H	642B	···	646B	···		
60A7	ADDNC	A, L	60E7	SUB	A, L	642C	···	646C	···		
60A8	···		60E8	···		642D	···	646D	···		
60A9	GTA	A, A	60E9	NEA	A, A	642E	···	646E	···		
60AA	GTA	A, B	60EA	NEA	A, B	642F	···	646F	···		
60AB	GTA	A, C	60EB	NEA	A, C	6430	···	6470	···		
60AC	GTA	A, D	60EC	NEA	A, D	6431	···	6471	···		
60AD	GTA	A, E	60ED	NEA	A, E	6432	···	6472	···		
60AE	GTA	A, H	60EE	NEA	A, H	6433	···	6473	···		
60AF	GTA	A, L	60EF	NEA	A, L	6434	···	6474	···		
60B0	···		60F0	···		6435	···	6475	···		
60B1	SUBNB	A, A	60F1	SBB	A, A	6436	···	6476	···		
60B2	SUBNB	A, B	60F2	SBB	A, B	6437	···	6477	···		
60B3	SUBNB	A, C	60F3	SBB	A, C	6438	···	6478	···		
60B4	SUBNB	A, D	60F4	SBB	A, D	6439	···	6479	···		
60B5	SUBNB	A, E	60F5	SBB	A, E	643A	···	647A	···		
60B6	SUBNB	A, H	60F6	SBB	A, H	643B	···	647B	···		
60B7	SUBNB	A, L	60F7	SBB	A, L	643C	···	647C	···		
60B8	···		60F8	···		643D	···	647D	···		
60B9	LTA	A, A	60F9	EQA	A, A	643E	···	647E	···		
60BA	LTA	A, B	60FA	EQA	A, B	643F	···	647F	···		
60BB	LTA	A, C	60FB	EQA	A, C	6440	···	6488	ANI	PA, byte	
60BC	LTA	A, D	60FC	EQA	A, D	6441	···	6489	ANI	PB, byte	
60BD	LTA	A, E	60FD	EQA	A, E	6442	···	648A	···		
60BE	LTA	A, H	60FE	EQA	A, H	6443	···	648B	ANI	MK, byte	
60BF	LTA	A, L	60FF	EQA	A, L	6444	···	6490	···		
60C0	···		61	DAA		6445	···	6491	···		
60C1	ADD	A, A	62	RETI		6446	···	6492	···		
60C2	ADD	A, B	63	···		6447	···	6493	···		
60C3	ADD	A, C	6408	···		6448	···	6498	ORI	PA, byte	
60C4	ADD	A, D	6409	···		6449	···	6499	ORI	PB, byte	
60C5	ADD	A, E	640A	···		644A	···	649A	···		
60C6	ADD	A, H	640B	···		644B	···	649B	ORI	MK, byte	
60C7	ADD	A, L	640C	···		644C	···	64A0	···		
60C8	···		640D	···		644D	···	64A1	···		
60C9	···		640E	···		644E	···	64A2	···		
60CA	···		640F	···		644F	···	64A3	···		
60CB	···		6410	···		6450	···	64A8	···		

μCOM-87 LC Correspondence between machine language and mnemonic (3/3)

64A9	···		706A	MOV	B, word	70C2	ADDX	D	74B8	···	
64AA	···		706B	MOV	C, word	70C3	ADDX	H	74C0	···	
64AB	···		706C	MOV	D, word	70C4	ADDX	D+	74C8	···	
64B0	···		706D	MOV	E, word	70C5	ADDX	H+	74D0	···	
64B1	···		706E	MOV	H, word	70C6	ADDX	D−	74D8	···	
64B2	···		706F	MOV	L, word	70C7	ADDX	H−	74E0	···	
64B3	···		7078	···		70C9	ONAX	B	74E8	···	
64B8	···		7079	MOV	word, A	70CA	ONAX	D	74F0	···	
64B9	···		707A	MOV	word, B	70CB	ONAX	H	74F8	···	
64BA	···		707B	MOV	word, C	70CC	ONAX	D+	75	EQIW	wa, byte
64BB	···		707C	MOV	word, D	70CD	ONAX	H+	76	SBI	A, byte
64C0	···		707D	MOV	word, E	70CE	ONAX	D−	77	EQI	A, byte
64C1	···		707E	MOV	word, H	70CF	ONAX	H−	78	CALF	word
64C2	···		707F	MOV	word, L	70D1	ADCX	B	⌇		
64C3	···		7089	ANAX	B	70D2	ADCX	D	7F		
64C8	ONI	PA, byte	708A	ANAX	D	70D3	ADCX	H	80	CALT	word
									⌇		
64C9	ONI	PB, byte	708B	ANAX	H	70D4	ADCX	D+	BF		
64CA	ONI	PC, byte	708C	ANAX	D+	70D5	ADCX	H+	C0	JR	word
64CB	ONI	MK, byte	708D	ANAX	H+	70D6	ADCX	D−	⌇		
64D0	···		708E	ANAX	D−	70D7	ADCX	H−	FF		
64D1	···		708F	ANAX	H−	70D9	OFFAX	B			
64D2	···		7091	XRAX	B	70DA	OFFAX	D			
64D3	···		7092	XRAX	D	70DB	OFFAX	H			
64D8	OFFI	PA, byte	7093	XRAX	H	70DC	OFFAX	D+			
64D9	OFFI	PB, byte	7094	XRAX	D+	70DD	OFFAX	H+			
64DA	OFFI	PC, byte	7095	XRAX	H+	70DE	OFFAX	D−			
64DB	OFFI	MK, byte	7096	XRAX	D−	70DF	OFFAX	H−			
64E0	···		7097	XRAX	H−	70E1	SUBX	B			
64E1	···		7099	ORAX	B	70E2	SUBX	D			
64E2	···		709A	ORAX	D	70E3	SUBX	H			
64E3	···		709B	ORAX	H	70E4	SUBX	D+			
64E8	···		709C	ORAX	D+	70E5	SUBX	H+			
64E9	···		709D	ORAX	H+	70E6	SUBX	D−			
64EA	···		709E	ORAX	D−	70E7	SUBX	H−			
64EB	···		709F	ORAX	H−	70E9	NEAX	B			
64F0	···		70A1	ADDNCX	B	70EA	NEAX	D			
64F1	···		70A2	ADDNCX	D	70EB	NEAX	H			
64F2	···		70A3	ADDNCX	H	70EC	NEAX	D+			
64F3	···		70A4	ADDNCX	D+	70ED	NEAX	H+			
64F8	···		70A5	ADDNCX	H+	70EE	NEAX	D−			
64F9	···		70A6	ADDNCX	D−	70EF	NEAX	H−			
64FA	···		70A7	ADDNCX	H−	70F1	SBBX	B			
64FB	···		70A9	GTAX	B	70F2	SBBX	D			
65	NEIW	wa, byte	70AA	GTAX	D	70F3	SBBX	H			
66	SUI	A, byte	70AB	GTAX	H	70F4	SBBX	D+			
67	NEI	A, byte	70AC	GTAX	D+	70F5	SBBX	H+			
68	···		70AD	GTAX	H+	70F6	SBBX	D−			
69	MVI	A, byte	70AE	GTAX	D−	70F7	SBBX	H−			
6A	MVI	B, byte	70AF	GTAX	H−	70F9	EQAX	B			
6B	MVI	C, byte	70B1	SUBNBX	B	70FA	EQAX	D			
6C	MVI	D, byte	70B2	SUBNBX	D	70FB	EQAX	H			
6D	MVI	E, byte	70B3	SUBNBX	H	70FC	EQAX	D+			
6E	MVI	H, byte	70B4	SUBNBX	D+	70FD	EQAX	H+			
6F	MVI	L, byte	70B5	SUBNBX	H+	70FE	EQAX	D−			
700E	SSPD	word	70B6	SUBNBX	D−	70FF	EQAX	H−			
700F	LSPD	word	70B7	SUBNBX	H−	71	···				
701E	SBCD	word	70B9	LTAX	B	72	···				
701F	LBCD	word	70BA	LTAX	D	73	JB				
702E	SDED	word	70BB	LTAX	H	7488	···				
702F	LDED	word	70BC	LTAX	D+	7490	···				
703E	SHLD	word	70BD	LTAX	H+	7498	···				
703F	LHLD	word	70BE	LTAX	D−	74A0	···				
7068	···		70BF	LTAX	H−	74A8	···				
7069	MOV	A, word	70C1	ADDX	B	74B0	···				



APPENDIX C SE-78C06 CIRCUIT DIAGRAM (1/2)



APPENDIX C SE-78C06 CIRCUIT DIAGRAM (2/2)

NEC OFFICES

NEC Electronics (Europe) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30, W.-Germany, Tel. (0211) 650301, Telex 8587419

NEC Electronics (Germany) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30, Tel. (0211) 650302, Telex 8587419
- Hindenburgstr. 28/29, 3000 Hannover 1, Tel. (0511) 881013–16. Telex 9230109
- Bayerstr. 21, 8000 München 2, Tel. (089) 591364–68, Telex 522971
- Solitudestr. 218, 7000 Stuttgart 31, Tel. (0711) 881119, Telex 7252220
- NEC Electronics (BNL) – 33 Alard du Hamelstraat, 5622 CC Eindhoven, Tel. (040) 445855, Telex 51923
- NEC Electronics (Scandinavia) – Box 4039, S-18304 Täby, Tel. (08) 756 7245, Telex 13839

NEC Electronics France, Tour Chenonceaux, 204, Rond Point du Pont de Sèvres, F-92516 Boulogne Billancourt, Tel. (01) 620-6400, Telex 203544

NEC Electronics Italiana S.R.L., Via Cardano 3, I-20124 Milano, Tel. (02) 67 09 108, Telex 315355

NEC Electronics UK, Limited, 116, Stevenston Street, New Stevenston, Motherwell ML 14LT, GB, Tel. (0698) 732221, Telex 777565