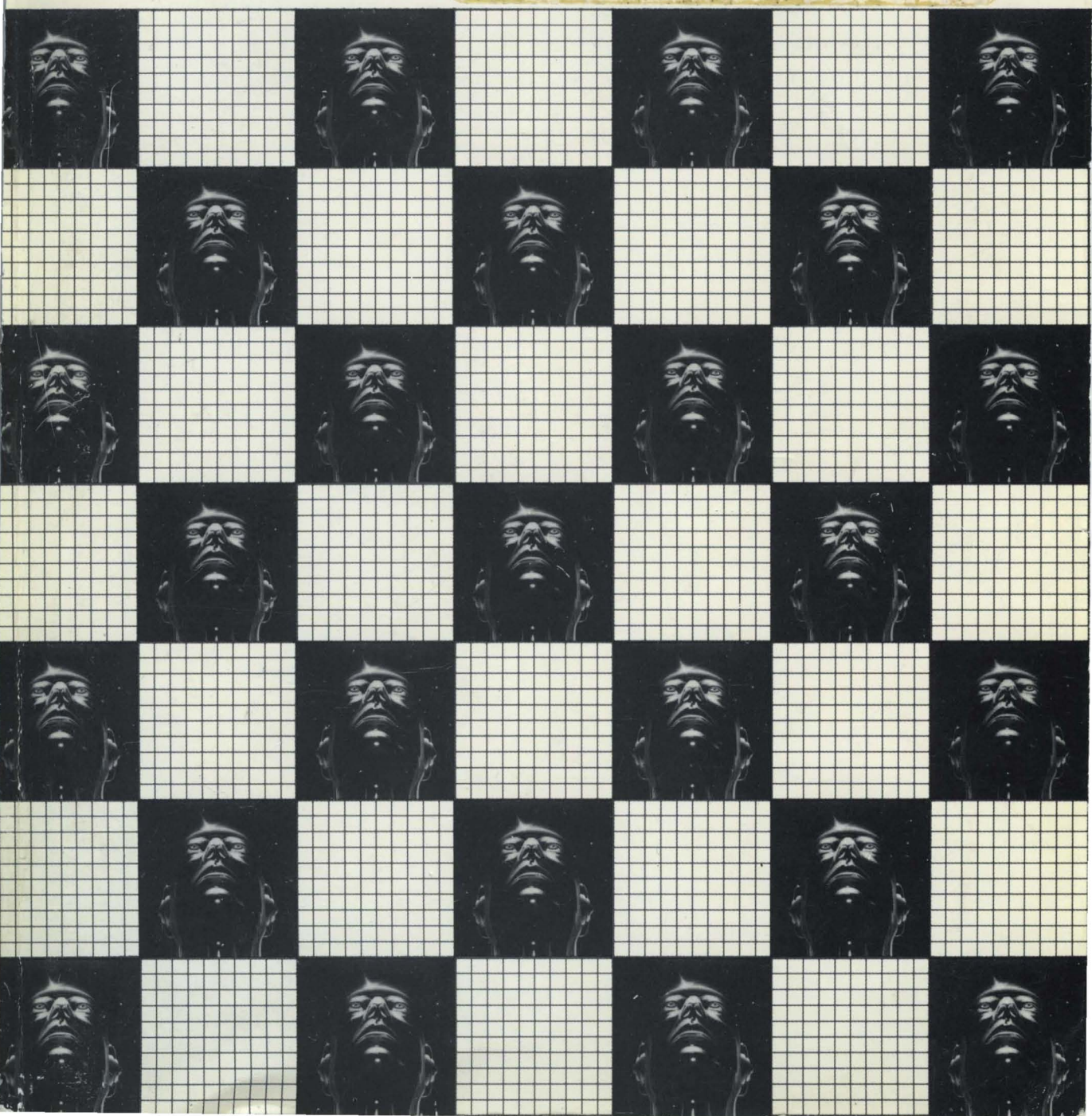
NEC

NEC Electronics U.S.A. Inc.

Microcomputer Division

1982 CATALOG

Western Microtechnology
10040 Bubb Road
Cupertino, CA 95014
Phone (408) 725-1660
TWX 910-338-0013



μPD7500 SERIES CMOS 4-BIT SINGLE CHIP MICROCOMPUTER FAMILY

Description

The μPD7500 Series CMOS 4-Bit Single Chip Microcomputer Family is a broad product line of 10 individual devices designed to fulfill a wide variety of applications. The advanced 4th generation architecture includes all of the functional blocks necessary for a single chip controller, including an ALU, Accumulator, Program Memory (ROM), Data Memory (RAM), four General Purpose Registers, Stack Pointer, Program Status Word (PSW), 8-Bit Timer/Event Counter, Interrupt Controller, Display Controller/Driver, and 8-Bit Serial Interface. The instruction set maximizes the efficient utilization of fixed Program Memory space, and includes a variety of addressing, Table-Look-up, Logical, Single Bit Manipulation, vectored jump, and Condition Skip Instructions.

The μPD7500 Series includes three different devices, the μPD7501, μPD7502, and μPD7503, capable of directly driving Liquid Crystal Displays with up to 12 7-segment digits. The μPD7508A can directly drive up to 35V Vacuum Fluorescent Displays with up to 8 7-segment digits, and the μPD7519 can directly drive up to 35V Vacuum Fluorescent Displays with up to 16 7-segment digits.

All 10 devices are manufactured with a Silicon gate CMOS process, consuming only 900μA max at 5V, and only 400μA max at 3V. The HALT and STOP power-down instructions can significantly reduce power consumption even further.

The flexibility and the wide variety of μPD7500 Series devices available make the μPD7500 series ideally suited for a wide range of battery-powered, solar-powered, and portable products, such as telecommunication devices, hand-held instruments and meters, automotive products, industrial controls, energy management systems, medical instruments, portable terminals, portable measuring devices, appliances, and consumer products.

Features

- ☐ Advanced 4th Generation Architecture
- ☐ Choice of 8-Bit Program Memory (ROM) size:
  - 1K, 2K, 4K internal, or 8K external bytes
- ☐ Choice of 4-Bit Data Memory (RAM) size:
  - 64, 96, 128, 208, 224, or 256 internal nibbles
- ☐ RAM Stack
- ☐ Four General Purpose Registers: D, E, H, and L
  - Can address Data Memory and I/O ports
  - Can be stored to or retrieved from Stack
- ☐ Powerful Instruction Set
  - From 58 to 110 instructions, including:
    - Direct/indirect addressing
    - Table Look-up
    - RAM Stack Push/Pop
    - Single byte subroutine calls
    - RAM and I/O port single bit manipulation
    - Accumulator and I/O port Logical operations
    - 10 μs Instruction Cycle Time, typically
- ☐ Extensive General Purpose I/O Capability
  - One 4-Bit Input Port
  - Two 4-Bit latched tri-state Output Ports
  - Five 4-Bit input/latched tri-state Output Ports
  - Easily expandable with μPD82C43 CMOS I/O Expander
  - 8-Bit Parallel I/O capability
- ☐ Hardware Logic Blocks — Reduce Software Requirements
  - Operation completely transparent to instruction execution
  - 8-Bit Timer/Event Counter
    - Binary-up counter generates INT_T at coincidence
    - Accurate Crystal Clock or External Event operation possible
  - Vectored, Prioritized Interrupt Controller
    - Three external interrupts (INT_0, INT_1, INT_2)
    - Two internal interrupts (INT_T, INT_S)
  - Display Controller/Driver
    - Complete Direct Drive and Control of Multiplexed LCD or Vacuum Fluorescent Display
    - Display Data automatically multiplexed from RAM to dedicated segment/backplane/digit driver lines
  - 8-Bit Serial Interface
    - 3-line I/O configuration generates INT_S upon transmission of eighth bit
    - Ideal for distributed intelligence systems or communication with peripheral devices
  - Complete operation possible in HALT and STOP power-down modes
- ☐ Built-in System Clock Generator
- ☐ Built-in Schmidt-Trigger RESET Circuitry
- ☐ Single Power Supply, Variable from 2.7V to 5.5V
- ☐ Low Power Consumption Silicon Gate CMOS Technology
  - 900 μA max at 5V, 400 μA max at 3V
  - HALT, STOP Power-down instructions reduce power consumption to 20 μA max at 5V, 10μA at 3V (Stop mode)
- ☐ Extended −40°C to +85°C Temperature Range Available
- ☐ Choice of 28-pin or 40-pin dual-in-line packages, or 52-pin or 64-pin flat plastic packages

Features	7500	7501	7502	7503	7506	7507	7507S	7508	7508A	7519
Internal ROM (8-bit words)		1K	2K	4K	1K	2K	2K	4K	4K	4K
Expandable to	8K									
RAM	256×4	96×4	128×4	224×4	64×4	128×4	128×4	224×4	208×4	256×4
I/O Lines	32	24	23	23	22	32	20	32	32	28
8-Bit Timer/Event Counter	•	•	•	•	•	•	•	•	•	•
8-Bit Serial Interface	•	•	•	•		•	•	•	•	•
Registers Outside RAM	4×4	2×4	4×4	4×4	2×4	4×4	4×4	4×4	4×4	4×4
Instructions	110	63	92	92	58	92	91	92	92	92
Min Cycle Time (μs)	6.67	6.67	6.67	6.67	6.67	6.67	6.67	6.67	6.67	6.67
Interrupts	5	4	4	4	2	4	4	4	4	4
Stack Levels	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM
Display Controller/ Driver		LCD	LCD	LCD					VFD drive only	VFD
Analog I/O										14-bit D/A
Current Consumption (max)										
Normal Operation	← 900 μA at 5V ± 10%; 400μA at 3V ± 10% →									
Stop Mode	← 20 μA at 5V ± 10%; 10μA at 3V ± 10% →									
Operating Temperature Range	−10°C to +70°C				−40°C to +85°C					
Packages										
28-pin DIP					•			•		
40-pin DIP						•		•	•	
52-pin Flat					•	•		•		
64-pin Flat		•	•	•						
64-pin QUIL	•									•

Instruction Set

The μPD7500 Series Instruction Set consists of 110 powerful instructions designed to take full advantage of the advanced μPD7500 architecture in your application. It is divided into two subsets, according to the complexity of the device.

Instruction Set "A" is available for the higher-performance μPD7500 Series devices having either a 2K × 8-bit or a 4K × 8-bit Program Memory. It can be used with the μPD7500, μPD7502, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, and μPD7519 products.

Instruction Set "B" is available for the lower-cost μPD7500 Series devices having a 1K × 8-bit Program Memory. Its instructions are a compatible subset of Instruction Set "A," and can be used with the μPD7500, μPD7501, and μPD7506 products.

Instruction Set Symbol Definitions

The following abbreviations are used in the description of the μPD7500 Series Instruction sets:

Symbol	Explanation and Use
A	Accumulator
A_n	Bit "n" of Accumulator
addr	Address
bit	Operand specifying one bit of a nibble
B_n	Bit "n" of two-bit operand B_1 B_0 Bit Specified 0 0 Bit 0 (LSB) 0 1 Bit 1 1 0 Bit 2 1 1 Bit 3 (MSB)
Bank	Bank Flag of PSW (μPD7500 only)
borrow	Resulting value is less than 0H
C	Carry Flag
data	Immediate data operand
D	D Register
D_n	Bit "n" of immediate data operand
DE	DE Register Pair
DL	DL Register Pair
E	E Register
H	H Register
HL	HL Register Pair
IER	Interrupt Enable Register IER bit: 0 1 2 3 Interrupt: INT_T $INT_{0/S}$ INT_1 INT_2
IME	Interrupt Master Enable F/F
INT_n	Interrupt "n"
IRF_n	Interrupt Request Flag "n"
L	L Register
overflow	Resulting value is greater than FH
P( )	Parallel Input/Output Port addressed by the value within the parentheses
PC	Program Counter
PC_n	Bit "n" of Program Counter
PSW	Program Status Word PSW bit: 0 1 2 3 Flag: Carry Bank SK_0 SK_1
rp	Register Pair, specified by the 3-bit immediate data operand D_{2-0}, as follows: D_2 D_1 D_0 rp Additional Action 0 0 0 DL none (instruction set "A" only) 0 0 1 DE none (instruction set "A" only) 1 0 0 HL– decrement L; skip if L = FH 1 0 1 HL+ increment L; skip if L = 0H 1 1 0 HL none
S	Skip Cycles: 0 when skip condition does not occur 1 when skip condition does occur
SIO	Serial I/O Shift Register
SIOCR	Serial I/O Count Register
SP	Stack Pointer
String	String Effect: In a string of similar instructions, only the first encountered is executed; the remainder of the instructions in the string are executed as NOP instructions
taddr	Operand specifying ROM Table Data
T_n	Bit "n" of ROM Table Data
TCR	Timer Counter Register
TMR	Timer Modulo Register
( )	The contents of the RAM location addressed by the value within the parentheses
[ ]	The contents of the ROM location addressed by the value within the brackets
←	Load, Store, or Transfer right operand into left operand
←→	Exchange the left and right operands
NOT	Logical NOT (One's complement)
AND	LOGICAL AND
OR	LOGICAL OR
XOR	LOGICAL Exclusive OR
▒▒▒	Instruction pertains to μPD7500 only

6

Instruction Set "A"

For the μPD7500, μPD7502, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, and μPD7519 devices only

Mnemonic	Function	Description	Instruction Code								Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
Load														
LADR addr	$A \leftarrow (D_{7\text{-}0})$	Load Accumulator from directly addressed RAM	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	0 D_1	0 D_0	38 00-FF	2	2	
LAI data	$A \leftarrow D_{3\text{-}0}$	Load Accumulator with immediate data	0	0	0	1	D_3	D_2	D_1	D_0	10-1F	1	1	String
LAM rp	A←(rp) rp = DL, DE, HL−, HL+, HL If rp = HL−, skip if borrow If rp = HL+, skip if overflow	Load Accumulator from Memory, possible skip	0	1	0	D_2	0	0	D_1	D_0	40, 41 50-52	1	1+S	See explanation of "rp" in symbol definitions
LAMT (μPD7500, μPD7502 only)	ROM addr = $PC_{10\text{-}6}$, 0, C, $A_{3\text{-}0}$ A←[ROM addr]7-4 (HL)←[ROM addr]3-0	Load Accumulator and Memory from Table	0	1	0	1	1	1	1	0	5E	1	2	
LAMTL (μPD7500, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, μPD7519, only)	ROM addr = $PC_{11\text{-}8}$, $A_{3\text{-}0}$, $(HL)_{3\text{-}0}$ A←[ROM addr]7-4 (HL)←[ROM addr]3-0	Load Accumulator and Memory from Table Long	0 0	0 0	1 1	1 1	1 0	1 1	1 0	1 0	3F 34	2	2	
LDEI data	D←$D_{7\text{-}4}$ E←$D_{3\text{-}0}$	Load DE register pair with immediate data	0 D_7	1 D_6	0 D_5	0 D_4	1 D_3	1 D_2	1 D_1	1 D_0	4F 00-FF	2	2	
LDI data	D←$D_{3\text{-}0}$	Load D register with immediate data	0 0	0 0	1 1	1 0	1 D_3	1 D_2	1 D_1	0 D_0	3E 20-2F	2	2	
LEI data	E←$D_{3\text{-}0}$	Load E register with immediate data	0 0	0 0	1 0	1 0	1 D_3	1 D_2	1 D_1	0 D_0	3E 00-0F	2	2	
LHI data	H←$D_{3\text{-}0}$	Load H register with immediate data	0 0	0 0	1 1	1 1	1 D_3	1 D_2	1 D_1	0 D_0	3E 30-3F	2	2	
LHLI data	H←$D_{7\text{-}4}$ L←$D_{3\text{-}0}$	Load HL register pair with immediate data	0 D_7	1 D_6	0 D_5	0 D_4	1 D_3	1 D_2	1 D_1	0 D_0	4E 00-FF	2	2	String
LHLT taddr	ROM addr = 0C0H + $D_{3\text{-}0}$ H←[ROM addr]7-4 L←[ROM addr]3-0	Load HL register pair from ROM Table	1	1	0	0	D_3	D_2	D_1	D_0	C0-CF	1	2	String
LLI data	L←$D_{3\text{-}0}$	Load L register with immediate data	0 0	0 0	1 0	1 1	1 D_3	1 D_2	1 D_1	0 D_0	3E 10-1F	2	2	
Store														
ST	(HL)←A	Store A to Memory	0	1	0	1	0	1	1	1	57	1	1	
Transfer														
TAD	D←A	Transfer A to D	0 1	0 0	1 1	1 0	1 1	1 0	1 1	0 0	3E AA	2	2	
TAE	E←A	Transfer A to E	0 1	0 0	1 0	1 0	1 1	1 0	1 1	0 0	3E 8A	2	2	
TAH	H←A	Transfer A to H	0 1	0 0	1 1	1 1	1 1	1 0	1 1	0 0	3E BA	2	2	
TAL	L←A	Transfer A to L	0 1	0 0	1 0	1 1	1 1	1 0	1 1	0 0	3E 9A	2	2	
TDA	A←D	Transfer D to A	0 1	0 0	1 1	1 0	1 1	1 0	1 1	0 1	3E AB	2	2	
TEA	A←E	Transfer E to A	0 1	0 0	1 0	1 0	1 0	1 1	1 0	0 1	3E 8B	2	2	
THA	A←H	Transfer H to A	0 1	0 0	1 1	1 1	1 1	1 0	1 1	0 1	3E BB	2	2	
TLA	A←L	Transfer L to A	0 1	0 0	1 0	1 1	1 1	1 0	1 1	0 1	3E 9B	2	2	
Exchange														
XAD	A↔D	Exchange A with D	0	1	0	0	1	0	1	0	4A	1	1	
XADR addr	A↔($D_{7\text{-}0}$)	Exchange A with directly addressed RAM	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	0 D_1	1 D_0	39 00-FF	2	2	
XAE	A↔E	Exchange A with E	0	1	0	0	1	0	1	1	4B	1	1	
XAH	A↔H	Exchange A with H	0	1	1	1	1	0	1	0	7A	1	1	
XAL	A↔L	Exchange A with L	0	1	1	1	1	0	1	1	7B	1	1	
XAM rp	A↔(rp) rp = DL, DE, HL−, HL+, HL If rp = HL−, skip if borrow If rp = HL+, skip if overflow	Exchange A with Memory, Possible Skip	0	1	0	D_2	0	1	D_1	D_0	44, 45 54-56	1	1+S	See explanation or "rp" in symbol definitions
XHDR addr	H↔($D_{7\text{-}0}$)	Exchange H with directly addressed RAM	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	1 D_1	0 D_0	3A 00-FF	2	2	
XLDR addr	L↔($D_{7\text{-}0}$)	Exchange L with directly addressed RAM	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	1 D_1	1 D_0	3B 00-FF	2	2	

Instruction Set "A" (Cont.)

For the μPD7500, μPD7502, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, and μPD7519 devices only

Mnemonic	Function	Description	Instruction Code									Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
Arithmetic														
ACSC	A, C←A+(HL)+C skip if carry	Add with carry; skip if carry	0	1	1	1	1	1	0	0	7C	1	1+S	Carry = 1
ADSC	A←A + D	Add D to A, skip if overflow	0 1	0 0	1 1	1 0	1 1	1 0	1 0	0 1	3E A9	2	2+S	Overflow
AESC	A←A + E	Add E to A, skip if overflow	0 1	0 0	1 0	1 0	1 1	1 0	1 0	0 1	3E 89	2	2+S	Overflow
AHSC	A←A + H	Add H to A, skip if overflow	0 1	0 0	1 1	1 1	1 1	1 0	1 0	0 1	3E B9	2	2+S	Overflow
AISC data	A←A + D_{3-0} skip if overflow	Add immediate skip if overflow	0	0	0	0	D_3	D_2	D_1	D_0	00-0F	1	1+S	Overflow
ALSC	A←A + L	Add H to A, skip if overflow	0 1	0 0	1 0	1 1	1 1	1 0	1 0	0 1	3E 99	2	2+S	Overflow
ASC	A←A + (HL) skip if overflow	Add memory; skip if overflow	0	1	1	1	1	1	0	1	7C	1	1+S	Carry = 1
SDSB	A←A − D skip if borrow	Subtract D from A, skip if borrow	0 1	0 0	1 1	1 0	1 1	1 0	1 0	0 0	3E A8	2	2+S	Borrow
SESB	A←A − E skip if borrow	Subtract E from A, skip if borrow	0 1	0 0	1 0	1 0	1 1	1 0	1 0	0 0	3E 88	2	2+S	Borrow
SHSB	A←A − H	Subtract H from A, skip if borrow	0 1	0 0	1 1	1 1	1 1	1 0	1 0	0 0	3E B8	2	2+S	Borrow
SLSB	A←A − L	Subtract L from A, skip if borrow	0 1	0 0	1 0	1 1	1 1	1 0	1 0	1 0	3E 98	2	2+S	Borrow
Logical														
ANL	A←A AND (HL)	AND Accumulator and Memory	0 1	0 0	1 1	1 1	1 0	1 0	1 1	1 0	3F B2	2	2	
EXL	A←A XOR (HL)	Exclusive-Or Accumulator and Memory	0	1	1	1	1	1	1	0	7E	1	1	
ORL	A←A OR (HL)	OR Accumulator and Memory	0 1	0 0	1 1	1 1	1 0	1 1	1 1	1 0	3F B6	2	2	
Accumulator														
CMA	A←NOT A	Complement Accumulator	0	1	1	1	1	1	1	1	7F	1	1	
RAL	C←A_3 A_3←A_2 A_2←A_1 A_1←A_0 A_0←C (old)	Rotate Accumulator left through Carry	0 1	0 0	1 1	1 1	1 0	1 1	1 1	1 1	3F B7	2	2	
RAR	C←A_0 A_0←A_1 A_1←A_2 A_2←A_3 A_3←C (old)	Rotate Accumulator right through Carry	0 1	0 0	1 1	1 1	1 0	1 0	1 1	1 1	3F B3	2	2	
Program Status Word														
RC	C←0	Reset Carry	0	1	1	1	1	0	0	0	78	1	1	
SC	C←1	Set Carry	0	1	1	1	1	0	0	1	79	1	1	
Increment and Decrement														
DDE	DE←DE − 1	Decrement DE	0 1	0 0	1 0	1 0	1 1	1 1	1 0	0 0	3E 8C	2	2	
DDRS addr	(D_{7-0})←(D_{7-0}) − 1 skip if (D_{7-0}) = FH	Decrement directly addressed RAM; skip if borrow	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	1 D_2	0 D_1	0 D_0	3C 00-FF	2	2+S	(D_{7-0}) = FH
DES	E←E − 1 skip if E = FH	Decrement E; skip if borrow	0	1	0	0	1	0	0	0	48	1	1+S	E = FH
DHL	HL←HL − 1	Decrement HL	0 1	0 0	1 0	1 1	1 1	1 0	1 0	0 0	3E 9C	2	2	
DLS	L←L − 1 skip if L = FH	Decrement L; skip if borrow	0	1	0	1	1	0	0	0	58	1	1+S	L = FH
IDE	DE←DE + 1	Increment DE	0 1	0 0	1 0	1 0	1 1	1 1	1 0	0 1	3E 8D	2	2	
IDRS addr	(D_{7-0})←(D_{7-0}) − 1 skip if (D_{7-0}) = 0H	Increment directly addressed; skip if overflow	0 D_7	0 D_6	1 D_5	1 D_4	1 D_3	1 D_2	0 D_1	1 D_0	3D 00-FF	2	2+S	(D_{7-0}) = 0H
IES	E←E + 1 skip if E = 0H	Increment E; skip if overflow	0	1	0	0	1	0	0	1	49	1	1+S	E = 0H
IHL	HL←HL + 1	Increment HL	0 1	0 0	1 0	1 1	1 1	1 0	1 0	0 1	3E 9D	2	2	
ILS	L←L + 1 skip if L = 0H	Increment L; skip if overflow	0	1	0	1	1	0	0	1	59	1	1+S	L = 0H 1
Bit Manipulation														
RMB bit	(HL)bit←0 bit = B_{1-0} (0-3)	Reset Memory bit	0	1	1	0	1	0	B_1	B_0	68-6B	1	1	
SMB bit	(HL)bit←1 bit = B_{1-0} (0-3)	Set Memory bit	0	1	1	0	1	1	B_1	B_0	6C-6F	1	1	

6

Instruction Set "A" (Cont.)

For the μPD7500, μPD7502, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, and μPD7519 devices only

Mnemonic	Function	Description	Instruction Code									Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
		Branch												
CALL addr	$(SP-1)\leftarrow PC_{7-4}$ $(SP-2)\leftarrow PC_{3-0}$ $(SP-3)\leftarrow PSW$ $(SP-4)\leftarrow PC_{11-8}$ $SP\leftarrow SP-4$ $BANK\leftarrow 0$ $PC_{11}\leftarrow 0$ $PC_{10-0}\leftarrow D_{10-0}$	Call subroutine	0 D_7	0 D_6	1 D_5	1 D_4	0 D_3	D_{10} D_2	D_9 D_1	D_8 D_0	30-37 00-FF	2	2	
CALT addr	$(SP-1)\leftarrow PC_{7-4}$ $(SP-2)\leftarrow PC_{3-0}$ $(SP-3)\leftarrow PSW$ $(SP-4)\leftarrow PC_{11-8}$ ROM addr = 0C0H + D_{5-0} $BANK\leftarrow 0$ $PC_{11-10}\leftarrow 00$ $PC_{9-7}\leftarrow$ [ROM addr]$_{7-5}$ $PC_{6-5}\leftarrow 00$ $PC_{4-0}\leftarrow$ [ROM addr]$_{4-0}$	Call subroutine through ROM Table (single byte)	1	1	D_5	D_4	D_3	D_2	D_1	D_0	D0-FF	1	2	
JAM data	$PC_{11-8}\leftarrow D_{3-0}$ $PC_{7-4}\leftarrow A$ $PC_{3-0}\leftarrow$ (HL)	Vectored Jump on Accumulator and Memory	0 0	0 0	1 0	1 1	1 D_3	1 D_2	1 D_1	1 D_0	3F 10-1F	2	2	
JCP addr	$PC_{5-0}\leftarrow D_{5-0}$	Jump within current page	1	0	D_5	D_4	D_3	D_2	D_1	D_0	80-BF	1	1	
JMP addr	$PC_{11-0}\leftarrow D_{11-0}$	Jump to specified address	0 D_7	0 D_6	1 D_5	0 D_4	D_{11} D_3	D_{10} D_2	D_9 D_1	D_8 D_0	20-2F 00-FF	2	2	
JUMPL addr	$BANK\leftarrow D_{12}$ $PC_{11-0}\leftarrow D_{11-0}$	Jump Long to specified address	0 0 D_7	0 0 D_6	1 D_{12} D_5	1 0 D_4	1 D_{11} D_3	1 D_{10} D_2	1 D_9 D_1	1 D_8 D_0	3F 00-0F 20-2F 00-FF	3	3	
RT	$PC_{11-8}\leftarrow$ (SP) $BANK\leftarrow (SP+1)$ $PC_{3-0}\leftarrow (SP+2)$ $PC_{7-4}\leftarrow (SP+3)$ $SP\leftarrow SP+4$	Return from Subroutine	0	1	0	1	0	0	1	1	53	1	1	
RTPSW	$PC_{11-8}\leftarrow$ (SP) $PSW\leftarrow (SP+1)$ $PC_{3-0}\leftarrow (SP+2)$ $PC_{7-4}\leftarrow (SP+3)$ $SP\leftarrow SP+4$	Return from Subroutine and restore PSW	0	1	0	0	0	0	1	1	43	1	2	
RTS	$PC_{11-8}\leftarrow$ (SP) $BANK\leftarrow (SP+1)$ $PC_{3-0}\leftarrow (SP+2)$ $PC_{7-4}\leftarrow (SP+3)$ $SP\leftarrow (SP+4)$ Skip unconditionally	Return from Subroutine; then skip next instruction	0	1	0	1	1	0	1	1	5B	1	1+S	Unconditional
		Stack												
POPDE	$E\leftarrow$ (SP) $D\leftarrow (SP+1)$ $SP\leftarrow SP+2$	Pop DE register pair off Stack	0 1	0 0	1 0	1 0	1 1	1 1	1 1	0 1	3E 8F	2	2	
POPHL	$L\leftarrow$ (SP) $H\leftarrow (SP+1)$ $SP\leftarrow SP+2$	Pop HL register pair off Stack	0 1	0 0	1 0	1 1	1 1	1 1	1 1	0 1	3E 9F	2	2	
PSHDE	$(SP-1)\leftarrow D$ $(SP-2)\leftarrow E$ $SP\leftarrow SP-2$	Push DE register pair on Stack	0 1	0 0	1 0	1 0	1 1	1 1	1 1	0 0	3E 8E	2	2	
PSHHL	$(SP-1)\leftarrow H$ $(SP-2)\leftarrow L$ $SP\leftarrow SP-2$	Push HL register pair on Stack	0 1	0 0	1 0	1 1	1 1	1 1	1 1	0 0	3E 9E	2	2	
TAMSP	$SP_{7-4}\leftarrow A$ $SP_{3-1}\leftarrow$ (HL)$_{3-1}$ $SP_0\leftarrow 0$	Transfer Accumulator and Memory to Stack Pointer	0 0	0 0	1 1	1 1	1 0	1 0	1 0	1 1	3F 31	2	2	
TSPAM	$A\leftarrow SP_{7-4}$ (HL)$_{3-1}\leftarrow SP_{3-1}$ (HL)$_0\leftarrow 0$	Transfer Stack Pointer to Accumulator and Memory	0 0	0 0	1 1	1 1	1 0	1 1	1 0	1 1	3F 35	2	2	
		Conditional Skip												
SKABT bit	Skip if A_{bit} = 1 bit = B_{1-0}(0-3)	Skip if Accumulator bit true	0	1	1	1	0	1	B_1	B_0	74-77	1	1+S	A_{bit} = 1
SKAEI data	Skip if A = D_{3-0}	Skip if Accumulator equals immediate data	0 0	0 1	1 1	1 0	1 D_3	1 D_2	1 D_1	1 D_0	3F 60-6F	2	2+S	A = D_{3-0}
SKAEM	Skip if A = (HL)	Skip if Accumulator equals Memory	0	1	0	1	1	1	1	1	5F	1	1+S	A = (HL)
SKC	Skip if C = 1	Skip if Carry	0	1	0	1	1	0	1	0	5A	1	1+S	C = 1
SKDEI data	Skip if D = D_{3-0}	Skip if D equals immediate data	0 0	0 1	1 1	1 0	1 D_3	1 D_2	1 D_1	0 D_0	3E 60-6F	2	2+S	D = D_{3-0}
SKEEI data	Skip if E = D_{3-0}	Skip if E equals immediate data	0 0	0 1	1 0	1 0	1 D_3	1 D_2	1 D_1	0 D_0	3E 40-4F	2	2+S	E = D_{3-0}
SKHEI data	Skip if H = D_{3-0}	Skip if H equals immediate data	0 0	0 1	1 1	1 1	1 D_3	1 D_2	1 D_1	0 D_0	3E 70-7F	2	2+S	H = D_{3-0}

Instruction Set "A" (Cont.)

For the μPD7500, μPD7502, μPD7503, μPD7507, μPD7507S, μPD7508, μPD7508A, and μPD7519 devices only

Mnemonic	Function	Description	Instruction Code									Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
Conditional Skip (Cont.)														
SKLEI data	Skip if L = D_{3-0}	Skip if L equals immediate data	0 0	0 1	1 0	1 1	1 D_3	1 D_2	1 D_1	0 D_0	3E 50-5F	2	2+S	L = D_{3-0}
SKMBF bit	Skip if $(HL)_{bit}$ = 0 bit = B_{1-0}(0-3)	Skip if Memory bit false	0	1	1	0	0	0	B_1	B_0	60-63	1	1+S	$(HL)_{bit}$ = 0
SKMBT bit	Skip if $(HL)_{bit}$ = 1 bit = B_{1-0}(0-3)	Skip if Memory bit true	0	1	1	0	0	1	B_1	B_0	64-67	1	1+S	$(HL)_{bit}$ = 1
SKMEI	Skip if (HL) = D_{3-0}	Skip if Memory equals immediate data	0 0	0 1	1 1	1 1	1 D_3	1 D_2	1 D_1	1 D_0	3F 70-7F	2	2+S	(HL) = D_{3-0}
Timer/Event Counter														
TAMMOD	TMR_{7-4}←A TMR_{3-0}←(HL)	Transfer Accumulator and Memory to Timer Modulo Register	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 1	3F 3F	2	2	
TCNTAM	A←TCR_{7-4} (HL)←TCR_{3-0}	Transfer Timer Count Register to Accumulator and Memory	0 0	0 0	1 1	1 1	1 1	1 0	1 1	1 1	3F 3B	2	2	
TIMER	TCR_{7-0}←0 IRF_T←0	Start Timer	0 0	0 0	1 1	1 1	1 0	1 0	1 1	1 0	3F 32	2	2	
Interrupt Control														
DI data	IME F/F←0 if data = 0 IER_{3-0}←IER_{3-0} AND NOT D_{3-0} if data < > 0	Disable Interrupt, Interrupt Master Enable F/F or specified	0 1	0 0	1 0	1 0	1 D_3	1 D_2	1 D_1	1 D_0	3F 80-8F	2	2	
EI data	IME F/F←1 if data = 0 IER_{3-0}←IER_{3-0} OR D_{3-0} if data < > 0	Enable Interrupt, Interrupt Master Enable F/F or specified	0 1	0 0	1 0	1 1	1 D_3	1 D_2	1 D_1	1 D_0	3F 90-8F	2	2	
SKI data	Skip if IRF_n AND D_{3-0} < > 0 IRF_n←IRF_n AND NOT D_{3-0}	Skip if Interrupt Request Flag is true	0 0	0 1	1 0	1 0	1 D_3	1 D_2	1 D_1	1 D_0	3F 40-4F	2	2+S	IRF_n = 1
Serial Interface														
SIO	SIOCR←0 $IRF_{0/S}$←0	Start Serial I/O Operation	0 0	0 0	1 1	1 1	1 0	1 0	1 1	1 1	3F 33	2	2	
TAMSIO	SIO_{7-4}←A SIO_{3-0}←(HL)	Transfer Accumulator and Memory to SI Shift Register	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 0	3F 3E	2	2	
TSIOAM	A←SIO_{7-4} (HL)←SIO_{3-0}	Transfer SI Shift Register to Accumulator and Memory	0 0	0 0	1 1	1 1	1 1	1 0	1 1	1 0	3F 3A	2	2	
Parallel I/O														
ANP data	P(P_{3-0})←P(P_{3-0}) AND D_{3-0}	AND output port latch with immediate data	0 D_3	1 D_2	0 D_1	0 D_0	1 P_3	1 P_2	0 P_1	0 P_0	4C 00-FF	2	2	
IP port	A←P(P_{3-0})	Input from port, immediate address	0 1	0 1	1 0	1 0	1 P_3	1 P_2	1 P_1	1 P_0	3F C0-CF	2	2	
IP1 (except μPD7507S)	A←P(1)	Input from Port 1	0	1	1	1	0	0	0	1	71	1	1	
IP54	A←P(5) (HL)←P(4)	Input Byte from Ports 5 and 4	0 0	0 0	1 1	1 1	1 1	1 0	1 0	1 0	3F 38	2	2	
IPL	A←P(L)	Input from Port specified by L	0	1	1	1	0	0	0	0	70	1	1	
OP port	P(P_{3-0})←A	Output to port, immediate address	0 1	0 1	1 1	1 0	1 P_3	1 P_2	1 P_1	1 P_0	3F E0-EF	2	2	
OP3	P(3)←A	Output to Port 3	0	1	1	1	0	0	1	1	73	1	1	
OP54	P(5)←A P(4)←(HL)	Output Byte to Ports 5 and 4	0 0	0 0	1 1	1 1	1 1	1 1	1 0	1 0	3F 3C	2	2	
OPL	P(L)←A	Output to port specified by L	0	1	1	1	0	0	1	0	72	1	1	
ORP data	P(P_{3-0})←P(P_{3-0}) OR D_{3-0}	OR output port latch with immediate data	0 D_3	1 D_2	0 D_1	0 D_0	1 P_3	1 P_2	0 P_1	1 P_0	4D 00-FF	2	2	
RPBit	μPD42C43 I/O Expander Port (L-1) Bit (L_{1-0})←0	Reset Port Bit specified by L	0	1	0	1	1	1	0	0	5C	1	1	
SPBit	μPD42C43 I/O Expander Port (L-2) bit (L_{1-0})←1	Set Port Bit specified by L	0	1	0	1	1	1	0	1	5D	1	1	
CPU Control														
HALT		Enter HALT Mode	0 0	0 0	1 1	1 1	1 0	1 1	1 1	1 0	3F 36	2	2	
NOP		No operation	0	0	0	0	0	0	0	0	00	1	1	
STOP		Enter STOP Mode	0 0	0 0	1 1	1 1	1 0	1 1	1 1	1 1	3F 37	2	2	

Instruction Set "B"

For the μPD7500, μPD7501, and μPD7506 devices only

Mnemonic	Function	Description	Instruction Code								Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
		Load												
LADR addr	$A \leftarrow (D_{6\text{-}0})$	Load Accumulator from directly addressed RAM	0 0	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	0 D_1	0 D_0	38 00-5F	2	2	
LAI data	$A \leftarrow D_{3\text{-}0}$	Load Accumulator with immediate data	0	0	0	1	D_3	D_2	D_1	D_0	10-1F	1	1	String
LAM rp	$A \leftarrow (rp)_i$ rp = HL−, HL+, HL if rp = HL−, skip if borrow if rp = HL+, skip if overflow	Load Accumulator from Memory, possible skip	0	1	0	1	0	0	D_1	D_0	50-52	1	1+S	See explanation of "rp" in symbol definitions
LAMT	ROM addr = $PC_{10\text{-}6}$, 0, C, $A_{3\text{-}0}$ $A \leftarrow$ [ROM addr]$_{7\text{-}4}$ $(HL) \leftarrow$ [ROM addr]$_{3\text{-}0}$	Load Accumulator and Memory from Table	0	1	0	1	1	1	1	0	5E	1	2	
[illegible]	[illegible]	[illegible]	0	0	1	1	1	1	1	1	3F	2	2	
			0	0	1	1	0	1	0	0	34			
LHI data	$H_3 \leftarrow 0$ $H_{2\text{-}0} \leftarrow D_{2\text{-}0}$	Load H register with immediate data	0	0	1	0	1	D_2	D_1	D_0	28-2F	1	1	
LHLI data	$H_{3\text{-}1} \leftarrow 0$ $H_0 \leftarrow D_4$ $L \leftarrow D_{3\text{-}0}$	Load HL register pair with immediate data	1	1	0	D_4	D_3	D_2	D_1	D_0	C0-DF	1	1	String
		Store												
ST	$(HL) \leftarrow A$	Store A to Memory	0	1	0	1	0	1	1	1	57	1	1	
STII data	$(HL) \leftarrow D_{3\text{-}0}$ $L \leftarrow L + 1$	Store immediate data and increment L	0	1	0	0	D_3	D_2	D_1	D_0	40-4F	1	1	
		Exchange												
XADR addr	$A \leftrightarrow (D_{6\text{-}0})$	Exchange A with directly addressed RAM	0 0	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	0 D_1	1 D_0	39 00-5F	2	2	
XAH	$A \leftrightarrow H$	Exchange A with H	0	1	1	1	1	0	1	0	7A	1	1	
XAL	$A \leftrightarrow L$	Exchange A with L	0	1	1	1	1	0	1	1	7B	1	1	
XAM rp	$A \leftrightarrow (rp)$ rp = HL−, HL+, HL if rp = HL−, skip if borrow if rp = HL+, skip if overflow	Exchange A with Memory, Possible Skip	0	1	0	1	0	1	D_1	D_0	54-56	1	1+S	See explanation or "rp" in symbol definitions
XHDR addr	$H \leftrightarrow (D_{6\text{-}0})$	Exchange H with directly addressed RAM	0 0	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	1 D_1	0 D_0	3A 00-5F	2	2	
XLDR addr	$L \leftrightarrow (D_{6\text{-}0})$	Exchange L with directly addressed RAM	0 0	0 D_6	1 D_5	1 D_4	1 D_3	0 D_2	1 D_1	1 D_0	3B 00-5F	2	2	
		Arithmetic												
ACSC	$A, C \leftarrow A \pm (HL) + C$ skip if carry	Add with carry; skip if carry	0	1	1	1	1	1	0	0	7C	1	1+S	Carry = 1
AISC data	$A \leftarrow A + D_{3\text{-}0}$ skip if overflow	Add immediate; skip if overflow	0	0	0	0	D_3	D_2	D_1	D_0	00-0F	1	1+S	Overflow
ASC	$A \leftarrow A + (HL)$ skip if overflow	Add memory; skip if overflow	0	1	1	1	1	1	0	1	7C	1	1+S	Carry = 1
		Logical												
ANL	A ← A AND (HL)	AND Accumulator and Memory	0 1	0 0	1 1	1 1	1 0	1 0	1 1	1 0	3F B2	2	2	
EXL	A ← A XOR (HL)	Exclusive-Or Accumulator and Memory	0	1	1	1	1	1	1	0	7E	1	1	
ORL	A ← A OR (HL)	OR Accumulator and Memory	0 1	0 0	1 1	1 1	1 0	1 1	1 1	1 0	3F B6	2	2	
		Accumulator												
CMA	A ← NOT A	Complement Accumulator	0	1	1	1	1	1	1	1	7F	1	1	
[illegible]	[illegible]	[illegible]	0	0	1	1	1	1	1	1	3F	2	2	
			1	0	1	1	0	1	1	1	B7			
RAR	$C \leftarrow A_0$ $A_0 \leftarrow A_1$ $A_1 \leftarrow A_2$ $A_2 \leftarrow A_3$ $A_3 \leftarrow C$ (old)	Rotate Accumulator right through Carry	0 1	0 0	1 1	1 1	1 0	1 0	1 1	1 1	3F B3	2	2	
		Program Status Word												
RC	C ← 0	Reset Carry	0	1	1	1	1	0	0	0	78	1	1	
SC	C ← 1	Set Carry	0	1	1	1	1	0	0	1	79	1	1	

Instruction Set "B" (Cont.)

For the μPD7500, μPD7501, and μPD7506 devices only

Mnemonic	Function	Description	Instruction Code									Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
Increment and Decrement														
DDRS addr	$(D_{6\text{-}0}) \leftarrow (D_{6\text{-}0}) - 1$ skip if $(D_{6\text{-}0})$ = FH	Decrement directly addressed RAM; skip if borrow	0 0	0 D_6	1 D_5	1 D_4	1 D_3	1 D_2	0 D_1	0 D_0	3C 00-5F	2	2+S	$(D_{6\text{-}0})$ = FH
DLS	L←L − 1 skip if L = FH	Decrement L; skip if borrow	0	1	0	1	1	0	0	0	58	1	1+S	L = FH
IDRS addr	$(D_{6\text{-}0}) \leftarrow (D_{6\text{-}0}) + 1$ skip if $(D_{6\text{-}0})$ = 0H	Increment directly addressed; skip if overflow	0 0	0 D_6	1 D_5	1 D_4	1 D_3	1 D_2	0 D_1	1 D_0	3D 00-5F	2	2+S	$(D_{6\text{-}0})$ = 0H
ILS	L←L + 1 skip if L = 0H	Increment L; skip if overflow	0	1	0	1	1	0	0	1	59	1	1+S	L = 0H
Bit Manipulation														
RMB bit	$(HL)_{bit} \leftarrow 0$ bit = $B_{1\text{-}0}$ (0-3)	Reset Memory bit	0	1	1	0	1	0	B_1	B_0	68-6B	1	1	
SMB bit	$(HL)_{bit} \leftarrow 1$ bit = $B_{1\text{-}0}$ (0-3)	Set Memory bit	0	1	1	0	1	1	B_1	B_0	6C-6F	1	1	
Branch														
CALL addr	$(SP-1) \leftarrow PC_{7\text{-}4}$ $(SP-2) \leftarrow PC_{3\text{-}0}$ $(SP-3) \leftarrow PSW$ $(SP-4) \leftarrow PC_{10\text{-}8}$ SP←SP − 4 BANK←0 $PC_{10\text{-}0} \leftarrow D_{10\text{-}0}$	Call subroutine	0 D_7	0 D_6	1 D_5	1 D_4	0 D_3	D_{10} D_2	D_9 D_1	D_8 D_0	30-37 00-FF	2	2	
CAL addr	$(SP-1) \leftarrow PC_{7\text{-}4}$ $(SP-2) \leftarrow PC_{3\text{-}0}$ $(SP-3) \leftarrow PSW$ $(SP-4) \leftarrow PC_{10\text{-}8}$ BANK←0 $PC_{10\text{-}0} \leftarrow 001D_4D_3000D_2D_1D_0$	Call short to CAL address subroutine	1	1	1	D_4	D_3	D_2	D_1	D_0	E0-FF	1	2	
JAM data	$PC_{10\text{-}8} \leftarrow D_{2\text{-}0}$ $PC_{7\text{-}4} \leftarrow A$ $PC_{3\text{-}0} \leftarrow (HL)$	Vectored Jump on Accumulator and Memory	0 0	0 0	1 0	1 1	1 0	1 D_2	1 D_1	1 D_0	3F 10-17	2	2	
JCP addr	$PC_{5\text{-}0} \leftarrow D_{5\text{-}0}$	Jump within current page	1	0	D_5	D_4	D_3	D_2	D_1	D_0	80-BF	1	1	
JMP addr	$PC_{10\text{-}0} \leftarrow D_{10\text{-}0}$	Jump to specified address	0 D_7	0 D_6	1 D_5	0 D_4	0 D_3	D_{10} D_2	D_9 D_1	D_8 D_0	20-27 00-FF	2	2	
[illegible] addr	BANK←D_{12}	Jump Long to	0	0	1	1	1	1	1	1	3F	3	3	
	$PC_{11\text{-}0} \leftarrow D_{11\text{-}0}$	specified address	0	0	D_{10}	0	D_{11}	D_{10}	D_9	D_8	00-0F [illegible]			
			[illegible]	[illegible]	[illegible]	[illegible]	D_3	D_2	D_1	D_0	00-FF			
RT	$PC_{10\text{-}8} \leftarrow (SP)$ BANK←(SP + 1) $PC_{3\text{-}0} \leftarrow (SP + 2)$ $PC_{7\text{-}4} \leftarrow (SP + 3)$ SP←SP + 4	Return from Subroutine	0	1	0	1	0	0	1	1	53	1	1	
RTS	$PC_{10\text{-}8} \leftarrow (SP)$ BANK←(SP + 1) $PC_{3\text{-}0} \leftarrow (SP + 2)$ $PC_{7\text{-}4} \leftarrow (SP + 3)$ SP←SP + 4 Skip unconditionally	Return from Subroutine; then skip next instruction	0	1	0	1	1	0	1	1	5B	1	1+S	Unconditional
Stack														
TAMSP	$SP_{7\text{-}4} \leftarrow A$ $SP_{3\text{-}1} \leftarrow (HL)_{3\text{-}1}$ $SP_0 \leftarrow 0$	Transfer Accumulator and Memory to Stack Pointer	0 0	0 0	1 1	1 1	1 0	1 0	1 0	1 1	3F 31	2	2	
TASP	$A \leftarrow SP_{7\text{-}4}$	Transfer Stack	0	0	1	1	1	1	1	1	3F	2	2	
	$(HL)_{3\text{-}1} \leftarrow SP_{3\text{-}1}$	Pointer to	0	0	1	1	0	1	0	1	35			
	$(HL)_0 \leftarrow 0$	Accumulator and Memory												
Conditional Skip														
SKABT bit	Skip if A_{bit} = 1 bit = $B_{1\text{-}0}$(0-3)	Skip if Accumulator bit true	0	1	1	1	0	1	B_1	B_0	74-77	1	1+S	A_{bit} = 1
SKAEI data	Skip if A = $D_{3\text{-}0}$	Skip if Accumulator equals Immediate data	0 0	0 1	1 1	1 0	1 D_3	1 D_2	1 D_1	1 D_0	3F 60-6F	2	2+S	A = $D_{3\text{-}0}$
SKAEM	Skip if A = (HL)	Skip if Accumulator equals Memory	0	1	0	1	1	1	1	1	5F	1	1+S	A = (HL)
SKC	Skip if C = 1	Skip if Carry	0	1	0	1	1	0	1	0	5A	1	1+S	C = 1
SKLEI data	Skip if L = $D_{3\text{-}0}$	Skip if L equals immediate data	0 0	0 1	1 0	1 1	1 D_3	1 D_2	1 D_1	0 D_0	3E 50-5F	2	2+S	L = $D_{3\text{-}0}$
SKMBF bit	Skip if $(HL)_{bit}$ = 0 bit = $B_{1\text{-}0}$(0-3)	Skip if Memory bit false	0	1	1	0	0	0	B_1	B_0	60-63	1	1+S	$(HL)_{bit}$ = 0
SKMBT bit	Skip if $(HL)_{bit}$ = 1 bit = $B_{1\text{-}0}$(0-3)	Skip if Memory bit true	0	1	1	0	0	1	B_1	B_0	64-67	1	1+S	$(HL)_{bit}$ = 1
SKNE	Skip if (HL) = $D_{3\text{-}0}$	Skip if Memory equals Immediate data	0 0	0 1	1 1	1 1	1 D_3	1 D_2	1 D_1	1 D_0	3F 70-7F	2	2+S	(HL) = $D_{3\text{-}0}$

Instruction Set "B" (Cont.)

For the μPD7500, μPD7501, and μPD7506 devices only

Mnemonic	Function	Description	Instruction Code									Bytes	Cycles	Skip Condition
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX			
Timer/Event Counter														
TAMMOD	$TMR_{7-4} \leftarrow A$	Transfer	0	0	1	1	1	1	1	1	3F	2	2	
	$TMR_{3-0} \leftarrow (HL)$	Accumulator and	0	0	1	1	1	1	1	1	3F			
		Memory to Timer												
		Modulo Register												
TCNTAM	$A \leftarrow TCR_{7-4}$	Transfer Timer	0	0	1	1	1	1	1	1	3F	2	2	
(except μPD7506)	$(HL) \leftarrow TCR_{3-0}$	Count Register to	0	0	1	1	1	0	1	1	3B			
		Accumulator and												
		Memory												
TIMER	$TCR_{7-0} \leftarrow 0$	Clear Timer	0	0	1	1	1	1	1	1	3F	2	2	
	$IRF_T \leftarrow 0$	Counter Register	0	0	1	1	0	0	1	0	32			
Interrupts														
SKI data	Skip if IRF_n AND $D_{3-0} <> 0$	Skip if Interrupt	0	0	1	1	1	1	1	1	3F	2	2+S	$IRF_n = 1$
	$IRF_n \leftarrow IRF_n$ AND NOT D_{3-0}	Request Flag is true	0	1	0	0	0	D_2	D_1	D_0	40-47			
Serial Interface														
SIO	$SIOCR \leftarrow 0$	Start Serial I/O	0	0	1	1	1	1	1	1	3F	2	2	
(except μPD7506)	$IRF_0/S \leftarrow 0$	Operation	0	0	1	1	0	0	1	1	33			
TAMSIO	$SIO_{7-4} \leftarrow A$	Transfer Accumu-	0	0	1	1	1	1	1	1	3F	2	2	
(except μPD7506)	$SIO_{3-0} \leftarrow (HL)$	lator and Memory	0	0	1	1	1	1	1	0	3E			
		to SIO Shift Register												
TSIOAM	$A \leftarrow SIO_{7-4}$	Transfer SIO Shift	0	0	1	1	1	1	1	1	3F	2	2	
(except μPD7506)	$HL \leftarrow SIO_{3-0}$	Register to	0	0	1	1	1	0	1	0	3A			
		Accumulator and												
		Memory												
Parallel I/O														
IP port	$A \leftarrow P(P_{3-0})$	Input from port,	0	0	1	1	1	1	1	1	3F	2	2	
		immediate address	1	1	0	0	P_3	P_2	P_1	P_0	C0-CF			
IP1	$A \leftarrow (1)$	Input from Port 1	0	1	1	1	0	0	0	1	71	1	1	
IP54	$A \leftarrow P(5)$	Input Byte from	0	0	1	1	1	1	1	1	3F	2	2	
	$(HL) \leftarrow P(4)$	Ports 5 and 4	0	0	1	1	1	0	0	0	38			
IPL	$A \leftarrow P(L)$	Input from Port	0	1	1	1	0	0	0	0	70	1	1	
		specified by L												
OP port	$P(P_{3-0}) \leftarrow A$	Output to port,	0	0	1	1	1	1	1	1	3F	2	2	
		immediate address	1	1	1	0	P_3	P_2	P_1	P_0	E0-EF			
OP3	$P(3) \leftarrow A$	Output to Port 3	0	1	1	1	0	0	1	1	73	1	1	
(except μPD7506)														
OP54	$P(5) \leftarrow A$	Output Byte to	0	0	1	1	1	1	1	1	3F	2	2	
	$P(4) \leftarrow (HL)$	Ports 5 and 4	0	0	1	1	1	1	0	0	3C			
OPL	$P(L) \leftarrow A$	Output to port	0	1	1	1	0	0	1	0	72	1	1	
		specified by L												
RPBL	μPD82C43 I/O Expander Port	Reset Port Bit	0	1	0	1	1	1	0	0	5C	1	1	
	(L_{3-2}) bit $(L_{1-0}) \leftarrow 0$	Specified by L												
SPBL	μPD82C43 I/O Expander Port	Set Port Bit	0	1	0	1	1	1	0	1	5D	1	1	
	(L_{3-2}) bit $(L_{1-0}) \leftarrow 1$	Specified by L												
CPU Control														
HALT		Enter HALT Mode	0	0	1	1	1	1	1	1	3F	2	2	
			0	0	1	1	0	1	1	0	36			
NOP		No operation	0	0	0	0	0	0	0	0	00	1	1	
STOP		Enter STOP Mode	0	0	1	1	1	1	1	1	3F	2	2	
			0	0	1	1	0	1	1	1	37			

Development Tools

For software development, editing, debugging, and assembly into object code, the NDS Development System, designed and manufactured by NEC Electronics U.S.A., Inc., is available. Additionally, for systems supporting either the ISIS-II (® Intel Corp.), CP/M (® Digital Research Corp.) or FDOS-II (® Motorola, Inc.,) operating systems, or Fortran IV ANSI 1966 V3.9, the ASM75 Cross-Assembler is available.

Once software development is complete, the code can be completely evaluated and debugged with hardware by the Evakit-7500 Evaluation Board. Available options include the Evakit-7500-LCD LCD driver board (for the μPD7501, μPD7502, and μPD7503), Evakit-7500-VFD Vacuum Fluorescent Display driver board (for the μPD7508A and μPD7519), and the Evakit-7500-RTT Real Time Tracer. The SE-7502 System Emulation Board will emulate complete functionality of the μPD7501, μPD7502, or μPD7503 for demonstrating your final system design. The SE-7508 System Emulation Board will emulate complete functionality of the μPD7506, μPD7507, μPD7507S, μPD7508, or μPD7508A for demonstrating your final system design. All of these boards take advantage of the capabilities of the μPD7500 Rom-less evaluation chip to perform their tasks.

Complete operation details on any μPD7500 Series CMOS 4-Bit Microcomputer can be found in the μPD7500 Series CMOS 4-Bit Microcomputer Technical Manual.