

Description

The μ PD8041AH and μ PD8741A are programmable peripheral interface controllers intended for use in master/slave configurations with 8048, 8080A, 8085A, 8086, and other 8- and 16-bit microprocessors. The μ PD8041AH/8741A functions as a totally self-sufficient controller with its own program and data memory to effectively unburden the master CPU from I/O handling and peripheral control functions.

The bus structure and data and status registers of the μ PD8041AH/8741A allow easy interface to the master processor bus. This enables the processor to perform control tasks which offload main system processing and more efficiently distribute processing functions.

The μ PD8041AH/8741A contains an 8-bit CPU, $1K \times 8$ program memory, 64×8 data memory, 18 I/O lines, a counter/timer, and a clock generator. The program memory for the μ PD8041AH is factory mask-programmed, while program memory for the μ PD8741A is UV EPROM for more flexibility.

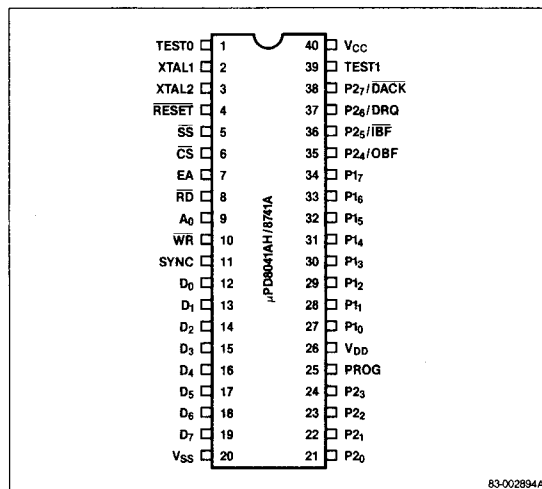
Features

- ☐ Complete single chip microcomputer
 - 8-bit CPU
 - $1K \times 8$ ROM
 - 64×8 RAM
 - 8-bit timer/counter
 - 18 I/O lines
- ☐ 8048-, 8080A-, 8085A-, 8086-compatible bus structure
- ☐ Asynchronous slave-to-master interface
 - 8-bit status register
 - Two data registers
- ☐ Interrupt, DMA, or polled operation
- ☐ Expandable I/O
- ☐ Single +5 V power supply

Ordering Information

Part Number	Package Type	Max Frequency of Operation
μ PD8041AHC	40-pin plastic DIP	11 MHz
μ PD8741AD	40-pin cerdip with quartz window	6 MHz

Pin Configuration



Pin Identification

No.	Symbol	Function
1	T0	Testable input 0
2	XTAL1	Crystal input 1
3	XTAL2	Crystal input 2
4	RESET	Reset input
5	SS	Single step input
6	CS	Chip select input
7	EA	External access input
8	RD	Read strobe input
9	A ₀	Address input 0
10	WR	Write strobe output
11	SYNC	SYNC output
12-19	D ₀ -D ₇	Bidirectional data bus
20	V _{SS}	Ground potential
21-24, 35-38	P ₂₀ -P ₂₇	Quasi-bidirectional Port 2
25	PROG	Program pulse output
26	V _{DD}	Programming supply voltage
27-34	P ₁₀ -P ₁₇	Quasi-bidirectional Port 1
39	T1	Testable input 1
40	V _{CC}	Primary power supply

Pin Functions**XTAL1 (Crystal 1)**

XTAL1 is one side of the crystal or external oscillator or external frequency source.

XTAL2 (Crystal 2)

XTAL2 is the other side of the crystal or frequency source.

T0 (Test 0)

T0 is the testable input using conditional transfer functions JT0, and JNT0. T0 can also be used during programming as a testable flag.

T1 (Test 1)

T1 is the testable input using conditional transfer functions JT1 and JNT1. T1 can be made the counter/timer input using the STRT CNT instruction.

 $\overline{\text{RESET}}$ (Reset)

An active low on $\overline{\text{RESET}}$ initializes the processor. $\overline{\text{RESET}}$ is also used for PROM programming, verification, and power-down.

 $\overline{\text{SS}}$ (Single Step)

An active low on $\overline{\text{SS}}$, together with the SYNC output, allows the processor to single step through each instruction in program memory.

EA (External Access)

An active high on EA disables internal program memory and fetches and accesses external program memory.

 $\overline{\text{RD}}$ (Read)

$\overline{\text{RD}}$ will pulse low when the processor reads data and status words from the data bus buffer or status register.

 $\overline{\text{WR}}$ (Write)

$\overline{\text{WR}}$ will pulse low when the processor writes data or status words to the data bus buffer or status register.

D0–D7 (Data Bus)

D0–D7 is a three-state, bidirectional data bus. D0–D7 interfaces the μPD8041AH/8741A to the 8-bit master system's data bus.

P10–P17 (Port 1)

P10–P17 is an 8-bit quasi-bidirectional port.

P20–P27 (Port 2)

P20–P27 is an 8-bit quasi-bidirectional port. P20–P23 output the high-order four bits of the address during an external program memory fetch. P20–P23 also function as a 4-bit I/O bus for the μPD82C43 I/O port expander. P24–P27 can be used as port lines or interrupt requests ($\overline{\text{IBF}}$ and $\overline{\text{OBF}}$) and DMA handshake signals (DRQ and DACK).

PROG (Program Pulse)

PROG is used in programming the μPD8041AH/8741A. PROG is also used as an output pulse during a fetch when interfacing with the μPD82C43 I/O port expander.

VCC (Primary Power Supply)

VCC is the primary power supply. VCC must be +5 V during programming and operation of the μPD8041AH.

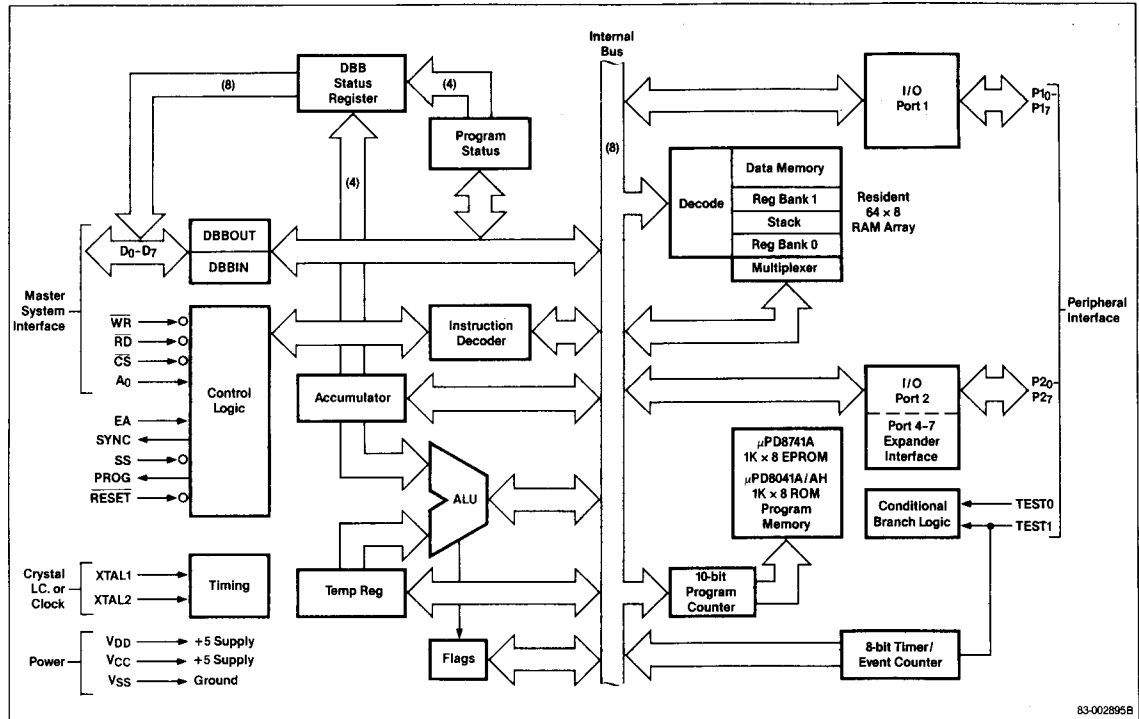
VDD (Programming Supply Voltage)

VDD is the programming supply voltage for programming the μPD8741AH. It is +5 V for normal operation of the μPD8041AH/8741A. VDD is also the low power standby input for the ROM version.

VSS (Ground)

VSS is ground potential.

Block Diagram



Absolute Maximum Ratings

$T_A = 25^\circ\text{C}$

Power supply voltage, V_{CC}	-0.5 V to +7.0 V
Power supply voltage, V_{DD}	-0.5 V to +7.0 V
Input voltage, V_{IN}	-0.5 V to +7.0 V
Output voltage, V_O	-0.5 V to +7.0 V
Operating temperature, T_{OPT}	0°C to $+70^\circ\text{C}$
Storage temperature, T_{STG}	-65°C to $+150^\circ\text{C}$

Comment: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of the specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Capacitance

$T_A = 25^\circ\text{C}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input capacitance	C_I			10	pF	
Output capacitance	C_{IO}			20	pF	

DC Characteristics

T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5 V ± 10%; μPD8041AH: V_{DD} = +5 V ± 5%; μPD8741A: V_{SS} = 0 V

Limits							
Parameter	Symbol	μPD8741A		μPD8041AH		Unit	Test Conditions
		Min	Max	Min	Max		
Input voltage low	V _{IL}	-0.5	0.8	-0.5	0.8	V	All except X1, X2, and RESET
	V _{IL1}	-0.5	0.6	-0.5	0.6	V	X1, X2, RESET
Input voltage high	V _{IH}	2.0	V _{CC}	2.0	V _{CC}	V	Except X1, X2, and RESET
	V _{IH1}	3.8	V _{CC}	3.8	V _{CC}	V	X1, X2, RESET
Output voltage low	V _{OL}		0.45		0.45	V	D ₀ -D ₇ , SYNC, I _{OL} = 2.0 mA
	V _{OL1}		0.45		0.45	V	Except PROG, I _{OL} = 1.0 mA
	V _{OL2}		0.45		0.45	V	PROG, I _{OL} = 1.0 mA
Output voltage high	V _{OH}	2.4		2.4		V	D ₀ -D ₇ , I _{OH} = -400 μA
	V _{OH1}	2.4		2.4		V	All other outputs: I _{OH} = -50 μA
Input current low	I _{LI}		0.5		0.5	mA	P1 ₀ -P1 ₇ , P2 ₀ -P2 ₇ : V _{IL} = 0.8 V
	I _{LI1}		0.2		0.2	mA	SS, RESET; V _{IL} = 0.8 V
Input leakage current	I _{IL}		±10		±10	μA	T ₀ , T ₁ , RD, WR, CS, EA, A ₀ , V _{SS} ≤ V _{IN} ≤ V _{CC}
Output leakage current	I _{OL}		±10		±10	μA	D ₀ -D ₇ , High Z state, V _{SS} + 0.45 V ≤ V _{IN} ≤ V _{CC}
Supply current (total)	I _{DD}		15		15	mA	V _{DD}
	I _{DD} + I _{CC}		135		125	mA	

AC Characteristics

T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5 V ± 10% V_{SS} = 0 V

DBB Read

		Limits					
		μPD8741A		μPD8041AH			
Parameter	Symbol	Min	Max	Min	Max	Unit	Test Conditions
CS, A ₀ setup to RD ↓	t _{AR}	300		0		ns	
CS, A ₀ hold after RD ↑	t _{RA}	30		0		ns	
RD pulse width	t _{RR}	300		160		ns	
CS, A ₀ , to data out delay	t _{AD}		370		130	ns	μPD8041A / 8741A: C _L = 150 pF μPD8041AH: C _L = 100 pF
RD ↓ to data out delay	t _{RD}		200		130	ns	μPD8041A / 8741A: C _L = 150 pF μPD8041AH: C _L = 100 pF
RD ↑ to data float delay	t _{DF}		140		85		
Cycle time	t _{CY}	2.5	15	1.36	15	ns	

AC Characteristics (cont)

T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5 V ± 10% V_{SS} = 0 V

DBB Write

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8741A		μPD8041AH			
		Min	Max	Min	Max		
CS, A ₀ setup to WR ↓	t _{AW}	0		0		ns	
CS, A ₀ hold after WR ↑	t _{WA}	0		0		ns	
WR pulse width	t _{WW}	250		160		ns	μPD8041A / 8741A: t _{CV} = 2.5 μs
Data setup to WR ↑	t _{DW}	150		130		ns	
Data hold after WR ↑	t _{WD}	0		0		ns	

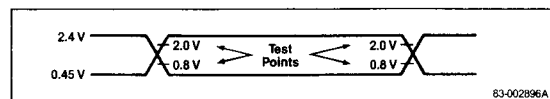
Port 2

Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8741A		μPD8041AH			
		Min	Max	Min	Max		
Port control setup to PROG ↓	t _{CP}	110		100		ns	μPD8041AH: C _L = 80 pF
Port control hold after PROG ↓	t _{PC}	100		60		ns	μPD8041AH: C _L = 20 pF
Input data setup to PROG ↓	t _{PR}		810		650	ns	μPD8041AH: C _L = 80 pF
Input data hold time	t _{PF}	0	150	0	150	ns	μPD8041AH: C _L = 20 pF
Output data setup time	t _{DP}	250		200		ns	μPD8041AH: C _L = 80 pF
Output data hold time	t _{PD}	65		65		ns	μPD8041AH: C _L = 20 pF
PROG pulse width	t _{pp}	1200		700		ns	

DMA

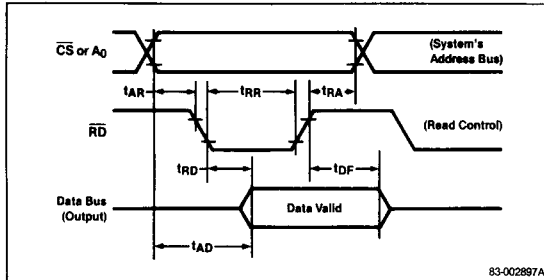
Parameter	Symbol	Limits				Unit	Test Conditions
		μPD8741A		μPD8041AH			
		Min	Max	Min	Max		
DACK setup time to RD, WR	t _{ACC}	0		0		ns	
DACK hold time after RD, WR	t _{CAC}	0		0		ns	
Data output delay after DACK	t _{ACD}		225		130	ns	μPD8041A / 8741A: C _L = 150 pF
DRQ clear delay time after RD, WR	t _{CRQ}		200		130	ns	μPD8041AH: C _L = 100 pF

AC Timing Test Points

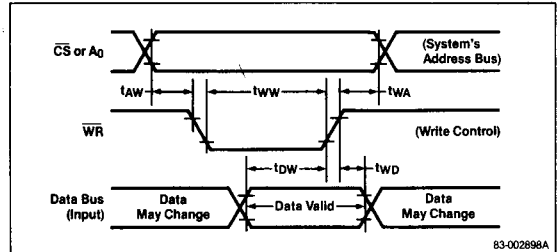


Timing Waveforms

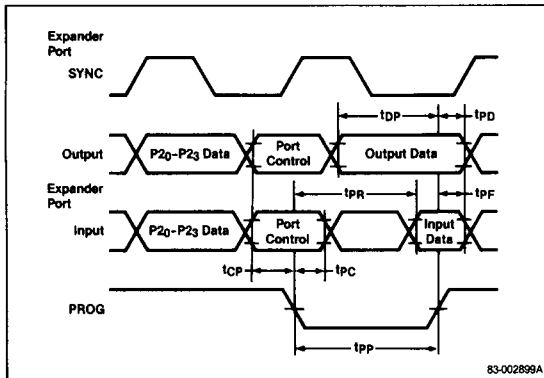
Read Operation (DBBOUT Register)



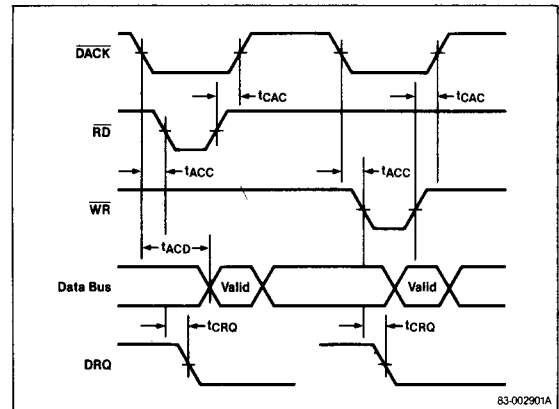
Write Operation (DBBIN Register)



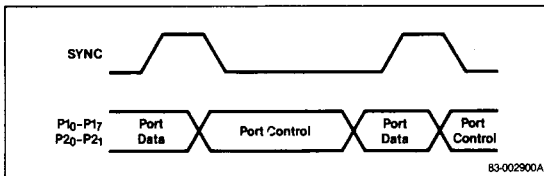
PORT 2



DMA



PORT (EA = 1)



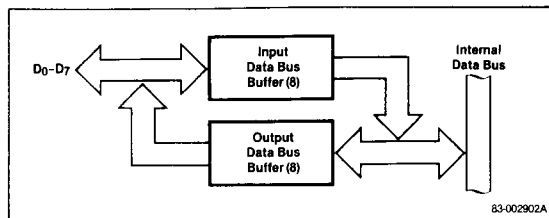
Functional Description

Two data bus buffers, an 8-bit status register, the $\overline{RD}$ and $\overline{WR}$ inputs, and expandable I/O lines enhance the μPD8041AH/8741A. These features enable easier master/slave interface and increased functionality.

Data Bus Buffers

Figure 1 shows how the input and output data bus buffers enable a smooth data flow to and from the master processors.

Figure 1. Data Bus Buffers



Status Register

The 8-bit status register includes four user-definable bits, ST₄-ST₇. Use the MOV STS, A instruction (90H) to define bits ST₄-ST₇ by moving accumulator bits 4-7 to bits 4-7 of the status register. Bits ST₀-ST₃ are not affected.

Figure 2 shows the format of the status register.

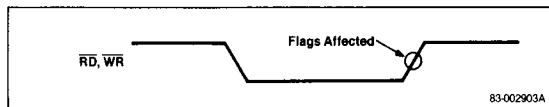
Figure 2. Status Register Format

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ST ₇	ST ₆	ST ₅	ST ₄	F1	F0	IBF	OBF

$\overline{RD}$ and $\overline{WR}$

The $\overline{RD}$ and $\overline{WR}$ inputs are edge-sensitive. Figure 3 shows that status bits $\overline{IBF}$, OBF, F1, and F0 are affected on the trailing edge at $\overline{RD}$ or $\overline{WR}$.

Figure 3. $\overline{RD}$ and $\overline{WR}$ Inputs



Port 24-Port 27

P2₄ and P2₅ can be used as either port lines or buffer status flag lines. This allows you to make OBF and $\overline{IBF}$ status available externally to interrupt the master processor. Upon execution of the EN FLAGS instruction (F5H), P2₄ becomes the OBF pin. When a 1 is written to P2₄, the OBF pin is enabled and the status of OBF is output. A₀ to P2₄ disables the OBF pin AND the pin remains low. This pin indicates valid data is available from the μPD8041AH/8741A.

An EN FLAGS instruction execution also enables P2₅ to indicate that the μPD8041AH/8741A is ready to accept data. A₁ written to P2₅ enables the $\overline{IBF}$ pin and the status of $\overline{IBF}$ is available on P2₅. A₀ written to P2₅ disables the $\overline{IBF}$ pin. If OBF is not true, the data at the data bus is invalid.

P2₆ and P2₇ can be used as either port lines or DMA handshake lines to allow DMA interface. The EN DMA instruction (E5H) enables P2₆ and P2₇ to be used as DRQ (DMA request) and $\overline{DACK}$ (DMA acknowledge), respectively.

When a 1 is written to P2₆, DRQ is activated and a DMA request is issued. The EN DMA instruction deactivates DRQ. You can also deactivate DRQ by adding $\overline{DACK}$ with $\overline{RD}$ or $\overline{WR}$. Execution of the EN DMA instruction enables P2₇ ($\overline{DACK}$) to function as a chip select input for the data bus buffer registers during DMA transfers.

Instruction Set

Instruction Set																			
Mnemonic	Operand	Operation	Operation Code										Flags						
			D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Cycles	Bytes	C	AC	F ₀	F ₁	IBF	OBF	ST ₄ -ST ₇
Accumulator																			
ADD	A, # data	(A) ← (A) + data d ₇	0	0	0	0	0	0	1	1	2	2	•						
ADD	A, Rr	(A) ← (A) + (Rr) r = 0-7	0	1	1	0	1	r	r	r	1	1	•						
ADD	A, @ Rr	(A) ← (A) + ((Rr)) r = 0-1	0	1	1	0	0	0	0	0	1	1	•						
ADDC	A, # data	(A) ← (A) + (C) + data d ₇	0	0	0	1	0	0	1	1	2	2	•						
ADDC	A, Rr	(A) ← (A) + (C) + (Rr) r = 0-7	0	1	1	1	1	1	r	r	1	1	•						
ADDC	A, @ Rr	(A) ← (A) + (C) + ((Rr)) r = 0-1	0	1	1	1	0	0	0	0	1	1	•						
ANL	A, # data	(A) ← (A) AND data d ₇	0	1	0	1	0	0	1	1	2	2							
ANL	A, Rr	(A) ← (A) AND (Rr) r = 0-7	0	1	0	1	1	r	r	r	1	1							
ANL	A, @ Rr	(A) ← (A) AND ((Rr)) r = 0-1	0	1	0	1	0	0	0	0	1	1							
CPL	A	(A) ← NOT (A)	0	0	1	1	0	1	1	1	1	1							
CLR	A	(A) ← 0	0	0	1	0	0	1	1	1	1	1							
DA	A		0	1	0	1	0	1	1	1	1	1	•						
DEC	A	(A) ← (A) - 1	0	0	0	0	0	1	1	1	1	1							
INC	A	(A) ← (A) + 1	0	0	0	1	0	1	1	1	1	1							
ORL	A, # data	(A) ← (A) OR data d ₇	0	1	0	0	0	d ₄	d ₃	d ₂	d ₁	d ₀	2						
ORL	A, Rr	(A) ← (A) OR (Rr) r = 0-7	0	1	0	0	1	r	r	r	1	1							
ORL	A, @ Rr	(A) ← (A) OR ((Rr)) r = 0-1	0	1	0	0	0	0	0	0	1	1							
RL	A	(AN + 1) ← (AN); N = 0-6 (A ₀) ← (A ₇)	1	1	1	0	0	1	1	1	1	1							

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										Flags							
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1	IBF	OBF	ST4	ST7
Accumulator (cont)																				
RLC	A	(AN + 1) ← (AN); N = 0-6 (A0) ← (C) (C) ← (A7)	1	1	1	1	0	1	1	1	1	1	1	1	1	•				
RR	A	(AN) ← (AN + 1); N = 0-6 (A7) ← (A0)	0	1	1	1	0	1	1	1	1	1	1	1	1					
RRC	A	(AN) ← (AN + 1); N = 0-6 (A7) ← (C) (C) ← (A0)	0	1	1	1	0	0	1	1	1	1	1	1	1	•				
SWAP	A	(A4-A7) ↔ (A0-A3)	0	1	0	0	0	1	1	1	1	1	1	1	1					
XRL	A, # data	(A) ← (A) XOR data	1	1	0	1	0	0	1	1	1	2	2							
		d7	d6	d5	d4	d3	d2	d1	d0											
XRL	A, Rr	(A) ← (A) XOR (Rr) r = 0-7	1	1	0	1	1	r	r	r	r	1	1	1	1					
XRL	A, @ Rr	(A) ← (A) XOR ((Rr)) r = 0-1	1	1	0	1	0	0	0	0	r	1	1	1	1					

Instruction Set (cont)

Mnemonic		Operand	Operation	Operation Code										Flags									
				D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1	IBF	OBF	ST ₄	ST ₇		
Branch																							
DJNZ		Ri, addr	(Ri) ← (Ri) - 1; r = 0-7 If (Ri) ≠ 0; (PC ₀ -PC ₇) ← addr	1 a7	1 a6	1 a5	0 a4	1 a3	1 a2	1 a1	1 a0	2	2										
JB _b		addr	(PC ₀ -PC ₇) ← addr if B _b = 1 (PC) ← (PC) + 2 if B _b = 0	b2 a7	b1 a6	b0 a5	1 a4	0 a3	0 a2	1 a1	0 a0	2	2										
JC		addr	(PC ₀ -PC ₇) ← addr if C = 1 (PC) ← (PC) + 2 if C = 0	1 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JF0		addr	(PC ₀ -PC ₇) ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	1 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JF1		addr	(PC ₀ -PC ₇) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	0 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JMP		addr	(PC ₈ -PC ₁₀) ← (addr;g-addr ₁₀) (PC ₀ -PC ₇) ← (addr ₀ -addr ₇) (PC ₁₁) ← DBF	a10 a7	a9 a6	a8 a5	0 a4	0 a3	1 a2	0 a1	0 a0	2	2										
JMPP		@ A	(PC ₀ -PC ₇) ← ((A))	1	0	1	1	0	0	1	1	2	1										
JNC		addr	(PC ₀ -PC ₇) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	1 a7	1 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JNIBF		addr	(PC ₀ -PC ₇) ← addr if IBF = 0 (PC) ← (PC) + 2 if IBF = 1	1 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JOBF			(PC ₀ -PC ₇) ← addr if OBF = 1 (PC) ← (PC) + 2 if OBF = 0	1 a7	0 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JNT0		addr	(PC ₀ -PC ₇) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	0 a7	0 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JNT1		addr	(PC ₀ -PC ₇) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	0 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JNZ		addr	(PC ₀ -PC ₇) ← addr if A = 0 (PC) ← (PC) + 2 if A = 1	1 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JTF		addr	(PC ₀ -PC ₇) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	0 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JT0		addr	(PC ₀ -PC ₇) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	0 a7	1 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JT1		addr	(PC ₀ -PC ₇) ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	0 a7	1 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										
JZ		addr	(PC ₀ -PC ₇) ← addr if A = 0 (PC) ← (PC) + 2 if A = 1	1 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	2	2										

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										Flags							
			D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Cycles	Bytes	C	AC	F ₀	F ₁	IBF	OBF	ST ₄ -ST ₇	
Control																				
EN I	Enable the external interrupt input		0	0	0	0	0	1	0	1	1	1								
DIS I	Disable the external interrupt input		0	0	0	1	0	1	0	1	1	1								
SEL RB0	(BS) ← 0		1	1	0	0	0	1	0	1	1	1								
SEL RB	(BS) ← 1		1	1	0	1	0	1	0	1	1	1								
EN DMA	Enable DMA handshake		1	1	1	1	0	1	0	1	1	1								
EN FLAGS	Enable interrupt to master device		1	1	1	0	0	1	0	1	1	1								
Data Moves																				
MOV	A, # data	(A) ← data	0	0	1	0	0	0	1	1	2	2								
MOV	A, Rr	(A) ← (Rr); r = 0-7	1	1	1	1	1	r	r	r	1	1								
MOV	A, @ Rr	(A) ← ((Rr)); r = 0-1	1	1	1	1	0	0	0	r	1	1								
MOV	A, PSW	(A) ← (PSW)	1	1	0	0	0	1	1	1	1	1								
MOV	Rr, # data	(Rr) ← data; r = 0-7	1	0	1	1	1	r	r	r	2	2								
MOV	Rr, A	(Rr) ← (A); r = 0-7	1	0	1	0	1	r	r	r	1	1								
MOV	@ Rr, A	((Rr)) ← (A); r = 0-1	1	0	1	0	0	0	0	r	1	1								
MOV	@ Rr, # data	((Rr)) ← data; r = 0-1	1	0	1	1	0	0	0	r	2	2								
MOV	PSW, A	(PSW) ← (A)	1	1	0	1	0	1	1	1	1	1								
MOVVP	A, @ A	(PC ₀ -PC ₇) ← (A) (A) ← ((PC))	1	0	1	0	0	0	1	1	2	1								
MOVVP3	A, @ A	(PC ₀ -PC ₇) ← (A) (PC ₈ -PC ₁₀) ← 011 (A) ← ((PC))	1	1	1	0	0	0	1	1	2	1								
XCH	A, Rr	(A) ↔ (Rr); r = 0-7	0	0	1	0	1	r	r	r	1	1								
XCH	A, @ Rr	(A) ↔ ((Rr)); r = 0-1	0	0	1	0	0	0	0	r	1	1								
XCHD	A, @ Rr	(A ₀ -A ₃) ↔ ((Rr)) ₀ -((Rr)) ₃ ; r = 0-1	0	0	1	1	0	0	0	r	1	1								

Instruction Set (cont)

Mnemonic	Operand	Operation	Operation Code										Flags							
			D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Cycles	Bytes	C	AC	F ₀	F ₁	IBF	OBF	ST ₄ -ST ₇	
Flags																				
CPL C		(C) ← NOT (C)	1	0	1	0	0	1	1	1	1	1	1	1	•					
CPL F ₀		(F ₀) ← NOT (F ₀)	1	0	0	1	0	1	0	1	1	1	1	1	•					
CPL F ₁		(F ₁) ← NOT (F ₁)	1	0	1	1	0	1	0	1	1	1	1	1	•					
CLR C		(C) ← 0	1	0	0	1	0	1	1	1	1	1	1	1	•					
CLR F ₀		(F ₀) ← 0	1	0	0	0	0	1	0	1	1	1	1	1	•					
CLR F ₁		(F ₁) ← 0	1	0	1	0	0	1	0	1	1	1	1	1	•					
MOV ST ₅ , A		ST ₄ -ST ₇ ← A ₄ -A ₇	1	0	0	1	0	0	0	0	1	1	1	1						
Input / Output																				
ANL	Pp, # data	(Pp) ← (Pp) AND data p = 1-2	1	0	0	1	1	0	p	p	2	2	2							
		d ₇	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀										
ANLD	Pp, A	(Pp) ← (Pp) AND (A ₀ -A ₃); p = 4-7	1	0	0	1	1	1	p	p	2	1	1							
IN	A, Pp	(A) ← (Pp); p = 1-2	0	0	0	0	1	0	p	p	2	1	1							
IN	A, DBB	(A) ← (DBB)	0	0	1	0	0	0	1	0	1	1	1						•	
MOVD	A, Pp	(A ₀ -A ₃) ← (Pp); p = 4-7 (A ₄ -A ₇) ← 0	0	0	0	0	1	1	p	p	2	1	1							
MOVD	Pp, A	(Pp) ← (A ₀ -A ₃); p = 4-7	0	0	1	1	1	1	p	p	1	1	1							
ORLD	Pp, A	(Pp) ← (Pp) OR (A ₀ -A ₃); p = 4-7	1	0	0	0	1	1	p	p	1	1	1							
ORL	Pp, # data	(Pp) ← (Pp) OR data p = 1-2	1	0	0	0	1	0	p	p	2	2	2							
		d ₇	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀										
OUT	DBB, A	(DBB) ← (A)	0	0	0	0	0	0	1	0	1	1	1							
OUTL	Pp, A	(Pp) ← (A); p = 1-2	0	0	1	1	1	0	p	p	1	1	1							
Registers																				
DEC	Rr (Rr)	(Rr) ← (Rr) - 1; r = 0-7	1	1	0	0	1	r	r	r	1	1	1							
INC	Rr	(Rr) ← (Rr) + 1; r = 0-7	0	0	0	1	1	r	r	r	1	1	1							
INC	@ Rr	((Rr)) ← ((Rr)) + 1; r = 0-1	0	0	0	1	0	0	0	0	r	1	1							

Instruction Set (cont)

Mnemonic Subroutine	Operand	Operation	Operation Code										Flags						
			D7	D6	D5	D4	D3	D2	D1	D0	Cycles	Bytes	C	AC	F0	F1	IBF	OSF	ST ₄ -ST ₇
CALL	addr	((SP)) ← (PC), (PSW ₄ -PSW ₇), (SP) ← (SP) + 1 (PC ₈ -PC ₁₀) ← (addr ₈ -addr ₁₀) (PC ₀ -PC ₇) ← (addr ₀ -addr ₇) (PC ₁₁) ← DBF	a ₁₀ a ₇	a ₉ a ₆	a ₅ a ₄	a ₃ a ₂	a ₁ a ₀	0	0	0	2	2							
RET		(SP) ← (SP) = 1 (PC) ← ((SP))	1	0	0	0	0	1	1	2	1								
RETR		(SP) ← (SP) = 1 (PC) ← ((SP)) (PSW ₄ -PSW ₇) ← ((SP))	1	0	0	1	0	0	1	2	1								
Timer / Counter																			
EN TCNTI	Enable internal interrupt flag for timer / counter output.		0	0	1	0	0	1	0	1	1	1							
DIS TCNTI	Disable internal interrupt flag for timer / counter output.		0	0	1	1	0	1	0	1	1	1							
MOV A, T	(A) ← (T)		0	1	0	0	0	0	1	0	1	1							
MOV T, A	(T) ← (A)		0	1	1	0	0	0	1	0	1	1							
STOP TCNT	Stop count for event counter.		0	1	1	0	0	1	0	1	1	1							
STRT CNT	Start count for event counter.		0	1	0	0	0	1	0	1	1	1							
STRT T	Start count for timer.		0	1	0	1	0	1	0	1	1	1							
Miscellaneous																			
NOP	No operation performed.		0	0	0	0	0	0	0	0	1	1							

Note:

- (1) Operation code designations r and p form the binary representation of the registers and ports involved.
- (2) The dot under the appropriate flag bit indicates that its contents is subject to change by the instruction it appears in.
- (3) References to the address and data are specified in bytes 2 and/or 1 of the instruction.
- (4) Numerical subscripts appearing in the operation column reference the specific bits affected.

Instruction Set (cont)**Symbol Definitions**

Symbol	Description
A	Accumulator
AC	Auxiliary carry flag
addr	Program memory address (12 bits)
B _b	Bit designator (b = 0–7)
BS	Bank switch
BUS	Bus port
C	Carry flag
CLK	Clock signal
CNT	Event counter
D	Nibble designator (4 bits)
data	Number of expression (8 bits)
DBF	Memory bank flip-flop
F0, F1	Flags 0, 1
I	Interrupt
P	In-page operation designator
IBF	Input buffer full flag
Pp	Port designator (p = 1, 2 or 4–7)
PSW	Program status word

Symbol	Description
Rr	Register designator (r = 0, 1 or 0–7)
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	Testable inputs 0, 1
X	External RAM
#	Prefix for immediate data
@	Prefix for indirect address
\$	Current value of program counter
(x)	Contents of external RAM location
((x))	Contents of memory location addressed by the contents of external RAM location
←	Replaced by
OBF	Output buffer full flag
DBB	Data bus buffer
AND	Logical product (logical AND)
OR	Logical sum (logical OR)
XOR	Exclusive-OR