

### Description

The μPD80C42 is a CMOS programmable peripheral interface controller which contains its own 8-bit micro-computer. It is well suited for use in master/slave configurations or as an intelligent peripheral device in applications requiring very low power consumption. The μPD80C42 has a CPU, 2K bytes of RAM, and 8-bit timer/counter, and I/O ports. I/O capability can be expanded by adding a μPD82C43, which interfaces directly to the μPD80C42. The external bus structure and associated control signals allow easy interfacing to 8048, 8085, and other microprocessor systems. The two standby modes allow even further reduction of power consumption in energy conscious systems.

With the exception of the  $\overline{STOP}$  pin, the μPD80C42 is pin-for-pin compatible with the μPD8041A and the μPD8741A.

### Features

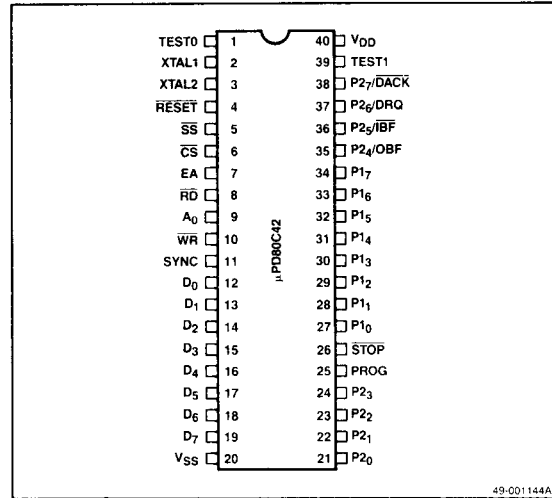
- CMOS technology
- Low power consumption
- 8048-, 8085A-, and 8086-bus compatible
- 8-bit CPU with  $2K \times 8$  ROM and  $128 \times 8$  RAM
- 8-bit timer/counter
- 18 I/O lines
- 8-bit status register
- Two data registers for asynchronous slave-to-master interface
- Interrupt, DMA, or polled operation
- Expandable I/O
- Two power down modes
- 8041A-, 8741A-pin compatible
- On-chip clock generator
- Single +5V power supply

### Ordering Information

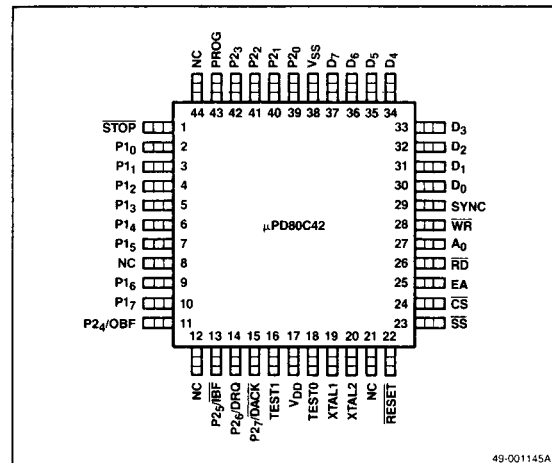
Part Number	Package Type	Max Frequency of Operation
μPD80C42C	40-pin plastic DIP	8 MHz
μPD80C42G-22	44-pin plastic miniflat	8 MHz

### Pin Configurations

#### 40-Pin Plastic DIP



#### 44-Pin Plastic Miniflat



## Pin Identification

### Plastic DIP

No.	Symbol	Function
1	TEST0	Test 0 input
2, 3	XTAL1, XTAL2	Crystal input
4	RESET	Reset input
5	SS	Single-step input
6	CS	Chip select input
7	EA	External access input
8	RD	Read input
9	A <sub>0</sub>	Address input
10	WR	Write input
11	SYNC	Synchronize output
12–19	D <sub>0</sub> –D <sub>7</sub>	Bidirectional port
20	V <sub>SS</sub>	Ground
21–24	P <sub>20</sub> –P <sub>23</sub>	Quasi-bidirectional port 2
35–38	P <sub>24</sub> / OBF, P <sub>25</sub> / IBF, P <sub>26</sub> / DRQ, P <sub>27</sub> / DACK	Output buffer full, input buffer full, DMA request, DMA acknowledge
25	PROG	PROG output strobe
26	STOP	STOP input
27–34	P <sub>10</sub> –P <sub>17</sub>	Quasi-bidirectional port 1
39	TEST1	Test 1 input
40	V <sub>DD</sub>	Positive power supply
—	NC	No connection

### Pin Functions

#### XTAL1, XTAL2 (Crystal)

XTAL1 and XTAL2 are the inputs for the crystal oscillator for the LC circuit generating internal clock signals. Use XTAL1 as the external clock input.

#### TEST0 (Test 0)

TEST0 is a testable input using conditional jump instructions JT0 and JNT0. TEST0 also resets the HALT mode.

#### TEST1 (Test 1)

TEST1 is a testable input using conditional jump instructions JT0 and JNT0. TEST1 is also an input to the event counter.

#### RESET (Reset)

RESET inputs a system reset, resets the HALT mode, and controls the STOP mode.

### Plastic Miniflat

No.	Symbol	Function
18	TEST0	Test 0 input
19, 20	XTAL1, XTAL2	Crystal input
22	RESET	Reset input
23	SS	Single-step input
24	CS	Chip select input
25	EA	External access input
26	RD	Read input
27	A <sub>0</sub>	Address input
28	WR	Write input
29	SYNC	Synchronize output
30–37	D <sub>0</sub> –D <sub>7</sub>	Bidirectional port
38	V <sub>SS</sub>	Ground
39–42	P <sub>20</sub> –P <sub>23</sub>	Quasi-bidirectional port 2
11, 13–15	P <sub>24</sub> / OBF, P <sub>25</sub> / IBF, P <sub>26</sub> / DRQ, P <sub>27</sub> / DACK	Output buffer full, input buffer full, DMA request, DMA acknowledge
43	PROG	PROG output strobe
1	STOP	STOP input
2–7, 9–10	P <sub>10</sub> –P <sub>17</sub>	Quasi-bidirectional port 1
16	TEST1	Test 1 input
17	V <sub>DD</sub>	Positive power supply
8, 12, 21, 44	NC	No connection

#### SS (Single-Step)

SS is an input used with SYNC to step the program through each instruction.

#### CS (Chip Select)

CS inputs the chip select signal. An active low enables the data bus.

#### EA (External Access)

EA is an input that inhibits internal program memory fetches. Use EA to check the ROM contents when debugging programs.

#### WR (Write)

WR is an input used by the master CPU to write data and commands into the data bus buffer in (DBBIN) register.

#### RD (Read)

RD is the input used by the master CPU to read data or

status words from the data bus buffer out (DBBOUT) or status registers.

### A<sub>0</sub> (Address 0)

A<sub>0</sub> is an address input that the master CPU uses to determine the bus operation as follows:

Cycle	A <sub>0</sub>	Operation
Read	0	Data
	1	Status
Write	0	Data
	1	Command

### SYNC (Synchronization)

SYNC is an output that occurs once per instruction cycle. SYNC is used as a strobe for external circuitry or to synchronize the single-step operation.

### PROG (PROG output)

When using the I/O expansion port (μPD82C43), PROG outputs a strobe that outputs data/addresses P<sub>20</sub>-P<sub>23</sub>.

### STOP (Stop)

The STOP input controls the hardware STOP mode.

### D<sub>0</sub>-D<sub>7</sub> (Port)

D<sub>0</sub>-D<sub>7</sub> is a bidirectional port that transfers data between the data bus buffer (DBBOUT, DBBIN) registers and the 8-bit master CPU data bus.

### P<sub>10</sub>-P<sub>17</sub> (Port 1)

P<sub>10</sub>-P<sub>17</sub> is a quasi-bidirectional, 8-bit port.

### P<sub>20</sub>-P<sub>27</sub> (Port 2)

P<sub>20</sub>-P<sub>27</sub> is a quasi-bidirectional, programmable 8-bit port. P<sub>24</sub>-P<sub>27</sub> (high-order bits) are alternative pins for the following interrupt request and DMA handshaking functions:

P<sub>24</sub> = OBF (Output buffer full)

P<sub>25</sub> = IBF (Input buffer full)

P<sub>26</sub> = DRQ (DMA request)

P<sub>27</sub> = DACK (DMA acknowledge)

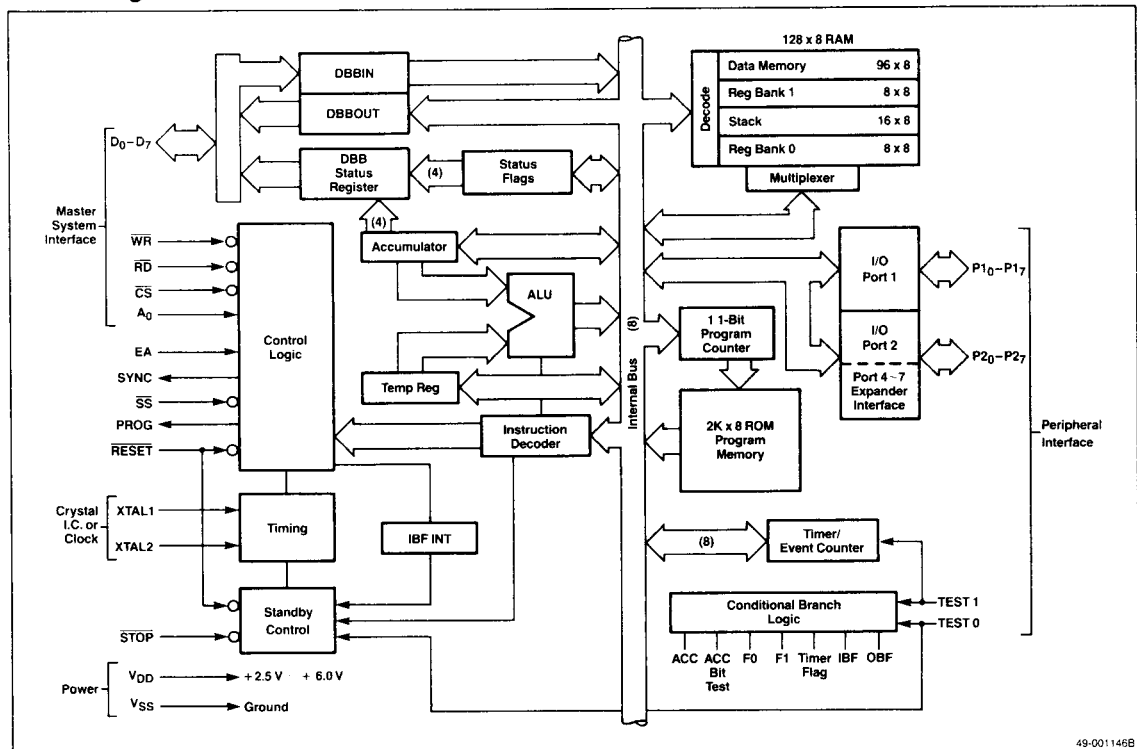
### V<sub>DD</sub> (Power Supply)

V<sub>DD</sub> is the positive power supply (+2.5 V to +6.0 V)

### V<sub>SS</sub> (Ground)

V<sub>SS</sub> is the ground potential.

## Block Diagram



49-001146B

### Absolute Maximum Ratings

T<sub>A</sub> = 25°C

Power supply voltage, V <sub>DD</sub>	-0.3 V to +7 V
Input voltage, V <sub>I</sub>	-0.3 V to V <sub>DD</sub> +0.3 V
Output voltage, V <sub>O</sub>	-0.3 V to V <sub>DD</sub> +0.3 V
Operating temperature, T <sub>OP</sub> T	-40°C to +85°C
Storage temperature, T <sub>STG</sub>	-65°C to +150°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.

### DC Characteristics

#### Standard Voltage Range

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +5 V ± 10%, V<sub>SS</sub> = 0 V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Input voltage low	V <sub>IL</sub>	-0.3		+0.8	V	
Input voltage high	V <sub>IH</sub>	2.2		V <sub>DD</sub>	V	Except RESET, XTAL1, XTAL2
	V <sub>IH1</sub>	V <sub>DD</sub> - 1		V <sub>DD</sub>	V	RESET, XTAL1, XTAL2
Output voltage low	V <sub>OL</sub>			+0.45	V	I <sub>OL</sub> = 2.0 mA
Output voltage high	V <sub>OH</sub>	2.4			V	D <sub>0</sub> -D <sub>7</sub> , SYNC, PROG; I <sub>OH</sub> = -400 μA
	V <sub>OH1</sub>	2.4			V	Port 1, port 2; I <sub>OH</sub> = -50 μA
	V <sub>OH2</sub>	V <sub>DD</sub> - 0.5			V	All outputs; I <sub>OH</sub> = -0.2 μA
Input current	I <sub>ILP</sub>			-500	μA	Port 1, port 2; V <sub>I</sub> ≤ V <sub>IL</sub>
	I <sub>ILC</sub>			-40	μA	SS, RESET; V <sub>I</sub> ≤ V <sub>IL</sub>
Input leakage current	I <sub>LI1</sub>			±1	μA	T <sub>0</sub> , T <sub>1</sub> , STOP, CS, A <sub>0</sub> , RD, WR; V <sub>SS</sub> ≤ V <sub>I</sub> ≤ V <sub>DD</sub>
	I <sub>LI2</sub>			±3	μA	EA; V <sub>SS</sub> ≤ V <sub>I</sub> ≤ V <sub>DD</sub>
Output leakage current	I <sub>LO</sub>			±1	μA	V <sub>SS</sub> ≤ V <sub>O</sub> ≤ V <sub>DD</sub> High impedance, D <sub>0</sub> -D <sub>7</sub> , port
Standby current	I <sub>DD1</sub>	1.5	3.0		mA	HALT mode; t <sub>CY</sub> = 1.25 μs
	I <sub>DD2</sub>	2	20		μA	STOP mode (1)
Supply current	I <sub>DD</sub>		10	20	mA	t <sub>CY</sub> = 1.25 μs
Data retention voltage	V <sub>DDDR</sub>	2.0			V	STOP mode (STOP, RESET ≤ 0.4 V) or RESET (RESET ≤ 0.4 V)

**Note:** (1) The input voltage pin is V<sub>I</sub> ≤ V<sub>IL</sub> or V<sub>I</sub> ≥ V<sub>IH</sub>.

#### Extended Voltage Range

T<sub>A</sub> = -40°C to +85°C, V<sub>DD</sub> = +2.5 V to +6.0 V, V<sub>SS</sub> = 0 V

Parameter	Symbol	Limits			Unit	Test Conditions	
		Min	Typ	Max			
Input voltage low	V <sub>IL</sub>	-0.3		+0.6	V	2.5 V ≤ V <sub>DD</sub> ≤ 4.5 V	
		-0.3		+0.8	V	4.5 V ≤ V <sub>DD</sub> ≤ 6.0 V	
Input voltage high	V <sub>IH</sub>	0.7 V <sub>DD</sub>		V <sub>DD</sub>	V	Except RESET, XTAL1, XTAL2	
	V <sub>IH1</sub>	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	RESET, XTAL1, XTAL2	
Output voltage low	V <sub>OL</sub>			+0.45	V	I <sub>OL</sub> = 1.0 mA	
Output voltage high	V <sub>OH</sub>	0.75 V <sub>DD</sub>			V	D <sub>0</sub> -D <sub>7</sub> , SYNC, PROG; I <sub>OH</sub> = -100 μA	
	V <sub>OH1</sub>	0.7 V <sub>DD</sub>			V	Port 1, port 2; I <sub>OH</sub> = -10 μA	
Input current	I <sub>ILP</sub>			-500	μA	Port 1, port 2; V <sub>I</sub> ≤ V <sub>IL</sub>	
				-40	μA	SS, RESET; V <sub>I</sub> ≤ V <sub>IL</sub>	
Input leakage current	I <sub>LI1</sub>			±1	μA	T <sub>0</sub> , T <sub>1</sub> , STOP, CS, A <sub>0</sub> , RD, WR; V <sub>SS</sub> ≤ V <sub>I</sub> ≤ V <sub>DD</sub>	
				±5	μA	EA; V <sub>SS</sub> ≤ V <sub>I</sub> ≤ V <sub>DD</sub>	
Output leakage current	I <sub>LO</sub>			±1	μA	V <sub>SS</sub> ≤ V <sub>O</sub> ≤ V <sub>DD</sub> High impedance, D <sub>0</sub> -D <sub>7</sub> , port	
				300	600	μA	HALT mode; V <sub>DD</sub> = 3 V; t <sub>CY</sub> = 5 μs
Standby current	I <sub>DD1</sub>			2.0	4.0	mA	V <sub>DD</sub> = 6 V t <sub>CY</sub> = 1.25 μs
				1	20	μA	STOP mode (1); V <sub>DD</sub> = 3 V
				2	50	mA	V <sub>DD</sub> = 6 V
Supply current	I <sub>DD</sub>			2.0	5.5	mA	V <sub>DD</sub> = 3 V; t <sub>CY</sub> = 5 μs
				16	30	mA	V <sub>DD</sub> = 6 V; t <sub>CY</sub> = 1.25 μs
Data retention voltage	V <sub>DDDR</sub>	2.0			V	STOP mode (STOP, RESET ≤ 0.4 V) or RESET (RESET ≤ 0.4 V)	

**Note:**

(1) The input voltage pin is V<sub>I</sub> ≤ V<sub>IL</sub> or V<sub>I</sub> ≥ V<sub>IH</sub>.

## AC Characteristics

### Standard Voltage Range — DBB Read

$T_A = -0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{DD} = +5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
$\overline{\text{CS}}$ , $A_0$ setup to RD low	$t_{AR}$	0			ns	
$\overline{\text{CS}}$ , $A_0$ hold from RD high	$t_{RA}$	0			ns	
RD pulse width	$t_{RR}$	200			ns	
$\overline{\text{CS}}$ , $A_0$ to data output delay	$t_{AD}$			150	ns	$C_L = 100\text{ pF}$
RD low to data output delay	$t_{RD}$			140	ns	$C_L = 100\text{ pF}$
RD high to data float delay	$t_{DF}$	0		85	ns	
Cycle time	$t_{CY}$	1.25		15	μs	

### Standard Voltage Range — DBB Write

$T_A = -0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{DD} = +5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
$\overline{\text{CS}}$ , $A_0$ setup to WR low	$t_{AW}$	0			ns	
$\overline{\text{CS}}$ , $A_0$ hold from WR high	$t_{WA}$	0			ns	
WR pulse width	$t_{WW}$	200			ns	
data setup to WR high	$t_{DW}$	130			ns	
Data hold from WR high	$t_{WD}$	0			ns	

### Extended Voltage Range — DBB Read

$T_A = -0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{DD} = +2.5\text{V}$  to  $+6.0\text{V}$ ,  $V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
$\overline{\text{CS}}$ , $A_0$ setup to RD low	$t_{AR}$	300			ns	
$\overline{\text{CS}}$ , $A_0$ hold from RD high	$t_{RA}$	200			ns	
RD pulse width	$t_{RR}$	2000			ns	
RD low to data output delay	$t_{RD}$			1500	ns	$C_L = 100\text{ pF}$
RD high to data float delay	$t_{DF}$	0		400	ns	
Cycle time	$t_{CY}$	5		15	μs	

### Extended Voltage Range — DBB Write

$T_A = -0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{DD} = +2.5\text{V}$  to  $6.0\text{V}$ ,  $V_{SS} = 0\text{V}$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
$\overline{\text{CS}}$ , $A_0$ setup to WR low	$t_{AW}$	300			ns	
$\overline{\text{CS}}$ , $A_0$ hold from WR high	$t_{WA}$	200			ns	
WR pulse width	$t_{WW}$	2000			ns	
data setup to WR high	$t_{DW}$	1500			ns	
Data hold from WR high	$t_{WD}$	200			ns	

### Standard Voltage Range — Port 2

$V_{DD} = +5\text{V} \pm 10\%$

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Port control setup to PROG low	$t_{CP}$	100			ns	$C_L = 80\text{ pF}$
Input port control hold from PROG low	$t_{PC1}$	0		80	ns	$C_L = 20\text{ pF}$
Output port control hold from PROG low	$t_{PC2}$	135			ns	$C_L = 20\text{ pF}$
Input data setup to PROG low	$t_{PR}$			650	ns	$C_L = 80\text{ pF}$
Input data hold from PROG high	$t_{PF}$	0		150	ns	$C_L = 20\text{ pF}$
Output data setup to PROG high	$t_{DP}$	200			ns	$C_L = 80\text{ pF}$
Output data hold from PROG high	$t_{PD}$	60			ns	$C_L = 20\text{ pF}$
PROG pulse width	$t_{PP}$	700			ns	

4

### AC Characteristics (cont)

#### Extended Voltage Range — Port 2

V<sub>DD</sub> = +2.5 V to +6.0 V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Port control setup to PROG low	t <sub>CP</sub>	460			ns	C <sub>L</sub> = 80 pF
Input port control hold from PROG low	t <sub>PC1</sub>	0		200	ns	C <sub>L</sub> = 20 pF
Output port control hold from PROG low	t <sub>PC2</sub>	1135			ns	C <sub>L</sub> = 20 pF
Input data setup to PROG low	t <sub>PR</sub>			2715	ns	C <sub>L</sub> = 80 pF
Input data hold from PROG high	t <sub>PF</sub>	0		500	ns	C <sub>L</sub> = 20 pF
Output data setup to PROG high	t <sub>DP</sub>	1850			ns	C <sub>L</sub> = 80 pF
Output data hold from PROG high	t <sub>PD</sub>	450			ns	C <sub>L</sub> = 20 pF
PROG pulse width	t <sub>pp</sub>	3250			ns	

#### Standard Voltage Range — DMA

V<sub>DD</sub> = +5 V ± 10%

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
DACK setup to RD, WR	t <sub>ACC</sub>	0			ns	
DACK hold from RD, WR	t <sub>CAC</sub>	0			ns	
DACK to data output delay	t <sub>ACD</sub>			140	ns	
RD, WR to DRQ clear delay	t <sub>CRQ</sub>			130	ns	C <sub>L</sub> = 150 pF

#### Extended Voltage Range — DMA

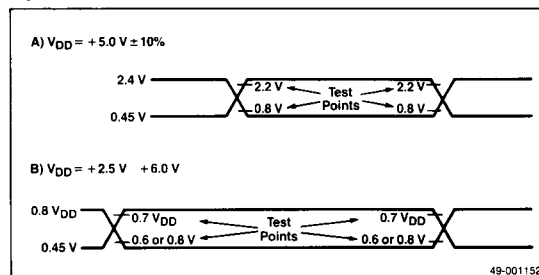
V<sub>DD</sub> = +2.5 V to +6.0 V

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
DACK setup to RD, WR	t <sub>ACC</sub>	200			ns	
DACK hold from RD, WR	t <sub>CAC</sub>	200			ns	
DACK to data output delay	t <sub>ACD</sub>			1500	ns	
RD, WR to DRQ clear delay	t <sub>CRQ</sub>			700	ns	C <sub>L</sub> = 150 pF

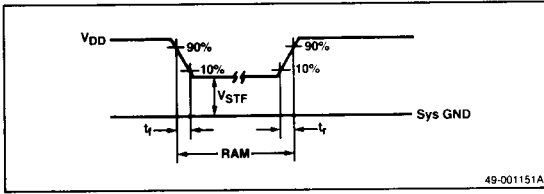
#### Standby Flag Retention Conditions

Parameter	Symbol	Limits			Unit	Test Conditions
		Min	Typ	Max		
Preservation of standby flag voltage fall time	t <sub>f</sub>	100			μs	
Preservation of standby flag voltage rise time	t <sub>r</sub>	100			μs	
Standby flag retention voltage	V <sub>STF</sub>	2.0			V	

#### Input Waveforms for AC Test



## Standby Flag Retention Timing

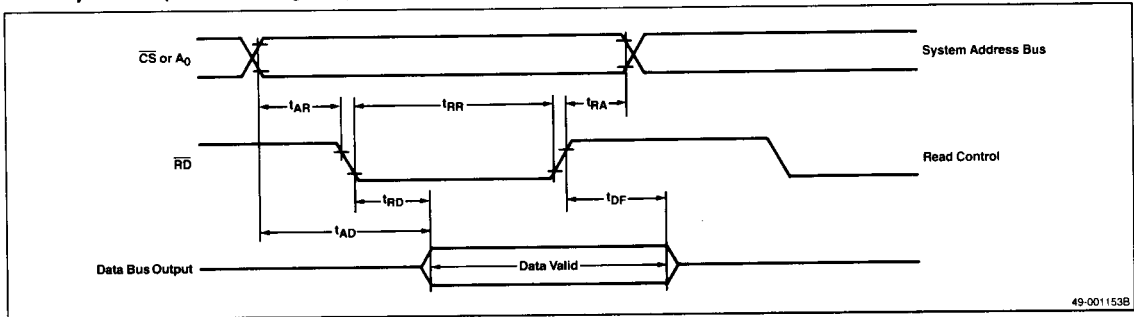


## Bus Timing Requirements

Symbol	Timing Formula	Min/Max	Unit
$t_{CP}$	$(1/10) t_{CY} - 40$	Min	ns
$t_{PC2}$	$(4/15) t_{CY} - 200$	Min	ns
$t_{PR}$	$(17/30) t_{CY} - 120$	Max	ns
$t_{PF}$	$(1/10) t_{CY}$	Max	ns
$t_{DP}$	$(2/5) t_{CY} - 150$	Min	ns
$t_{PD}$	$(1/10) t_{CY} - 50$	Min	ns
$t_{PP}$	$(7/10) t_{CY} - 250$	Min	ns
$t_{CY}$	$(1/f_{XTAL}) \times 15$		μs

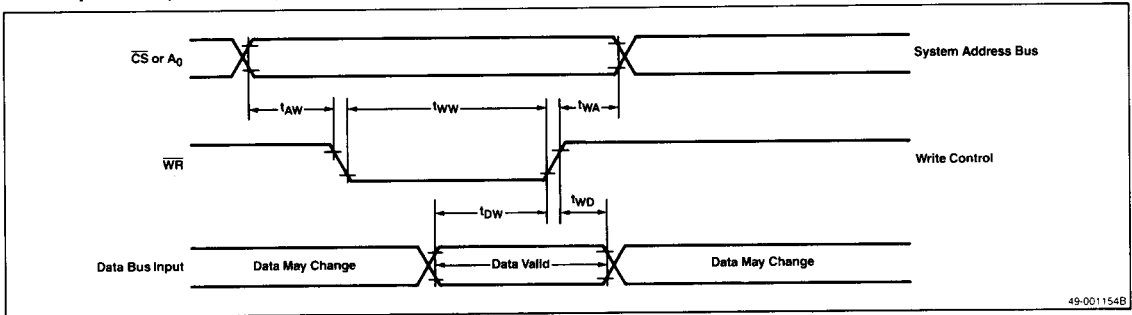
## Timing Waveforms

### Read Operation (DBBOUT Register)



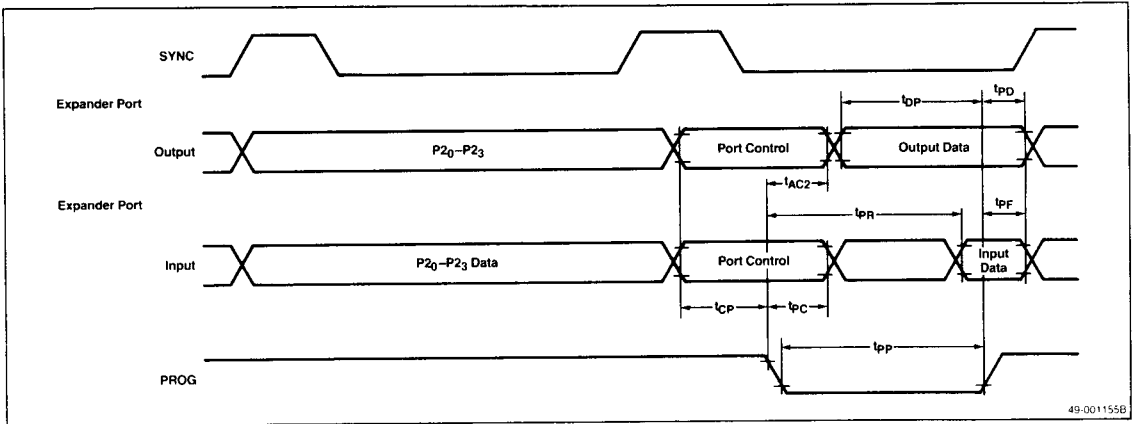
4

### Write Operation (DBBIN Register)

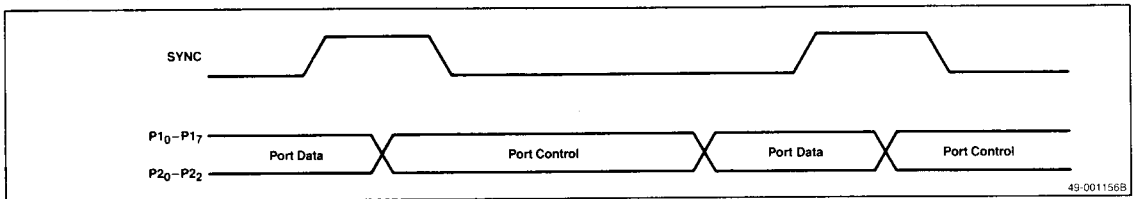


**Timing Waveforms (cont)**

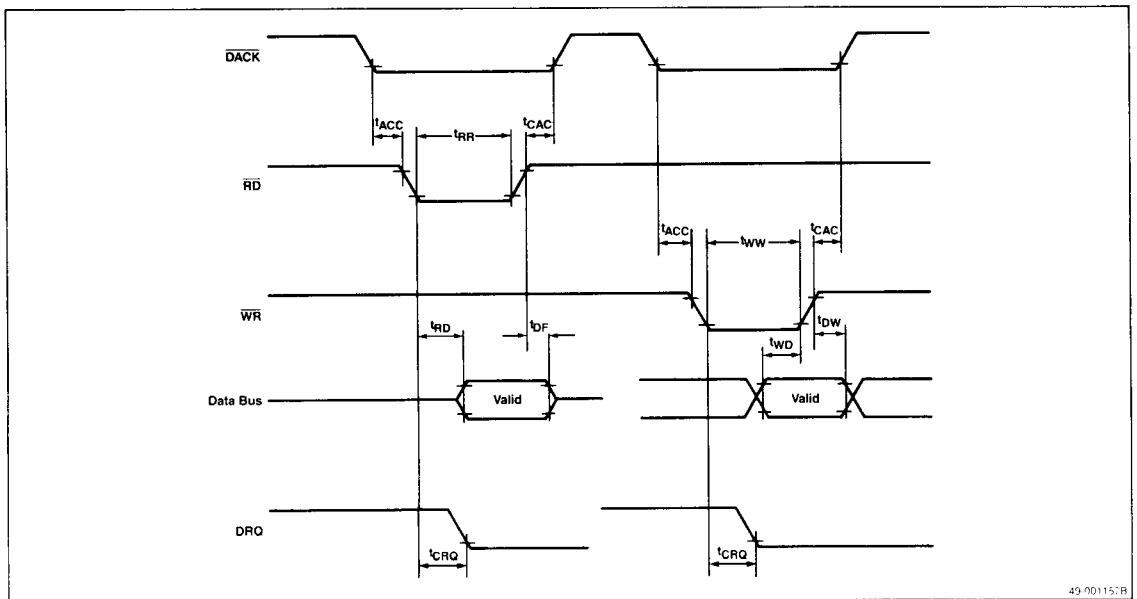
**PORT2**



**PORT (EA = 1)**



**DMA**



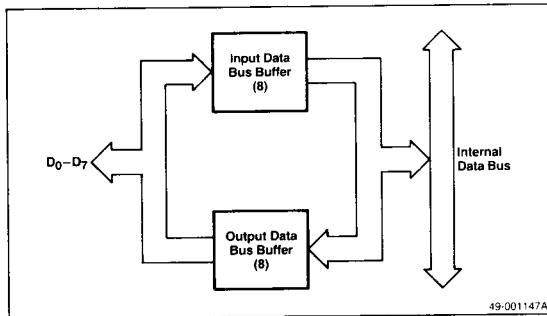


## Functional Description

### Data Bus Buffer In (DBBIN) and Data Bus Buffer Out (DBBOUT) Registers

As figure 1 shows, the DBBIN and DBBOUT registers transfer data to and from the master processors by way of the 8-bit external data bus (D<sub>0</sub>-D<sub>7</sub>) and the 8-bit internal data bus.

Figure 1. μPD80C42 Data Flow



### Data Bus Buffer (DBB) Status Register

The μPD80C42 has an 8-bit status register (ST<sub>0</sub>-ST<sub>7</sub>) that contains information about the current status of the master or slave processor. The MOV STS, A instruction makes status bits ST<sub>4</sub>-ST<sub>7</sub> user-definable by moving accumulator bits 4-1 to bits ST<sub>4</sub>-ST<sub>7</sub> of the status register (ST<sub>0</sub>-ST<sub>3</sub> are not affected). Bits ST<sub>0</sub>-ST<sub>3</sub> give the status of the Output Buffer Full (OBF) and Input Buffer Full (IBF) bits, and flag bits (F<sub>0</sub>, F<sub>1</sub>). Figure 2 shows the status register format.

Figure 2. Status Register Format

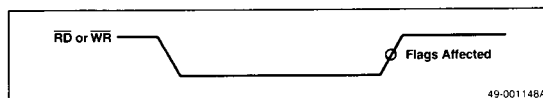
ST <sub>7</sub>	ST <sub>6</sub>	ST <sub>5</sub>	ST <sub>4</sub>	ST <sub>3</sub>	ST <sub>2</sub>	ST <sub>1</sub>	ST <sub>0</sub>
UD	UD	UD	UD	F <sub>1</sub>	F <sub>0</sub>	IBF	OBF

The MOV STS, A instruction is coded as follows:

```
1 0 0 1 0 0 0 0 90H
```

Figure 3 shows how ST<sub>0</sub>-ST<sub>3</sub> change internally on the trailing-edge of  $\overline{RD}$  or  $\overline{WR}$  ( $\overline{RD}$  and  $\overline{WR}$  are edge-sensitive).

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



You can make ST<sub>0</sub> (OBF) and ST<sub>1</sub> (IBF) externally available in order to interrupt the master processor by executing the EN FLAGS instruction. When the EN FLAGS instruction is executed, P<sub>24</sub> becomes the OBF pin. A 1 written to P<sub>24</sub> enables OBF and outputs its status. A 0 written to P<sub>24</sub> disables OBF by holding it low. Use OBF to indicate that valid data is available from the output data bus buffer register.

You can also use the EN FLAGS instruction to use P<sub>25</sub> as the  $\overline{IBF}$  pin. A 1 written to P<sub>25</sub> enables  $\overline{IBF}$  to output the inverse of the IBF status bit. A 0 written to P<sub>25</sub> disables  $\overline{IBF}$  by holding it low, making data at the data bus invalid.

The EN FLAGS instruction is coded as follows:

```
1 1 1 1 0 1 0 1 F5H
```

P<sub>26</sub> and P<sub>27</sub> are port pins or DMA handshake pins that allow a DMA interface. Use the EN DMA instruction to enable P<sub>26</sub> and P<sub>27</sub> as DRQ (DMA Request) and  $\overline{DACK}$  (DMA Acknowledge), respectively. A 1 written to P<sub>26</sub> activates DRQ, thus issuing a DMA request. Deactivate DRQ with the EN DMA instruction,  $\overline{DACK}$  ANDed with  $\overline{RD}$ , or  $\overline{DACK}$  ANDed with  $\overline{WR}$ . When EN DMA is executed, P<sub>27</sub> ( $\overline{DACK}$ ) functions as a chip select input for the data bus buffer registers during DMA transfers.

The EN DMA instruction is coded as follows:

```
1 1 1 0 0 1 0 1 E5H
```

### HALT Mode

The HALT mode allows the μPD80C42 to conserve power during periods of inactivity. In the HALT mode, the oscillator remains active but the internal system clock stops. The HALT instruction allows the processor to enter the HALT mode.

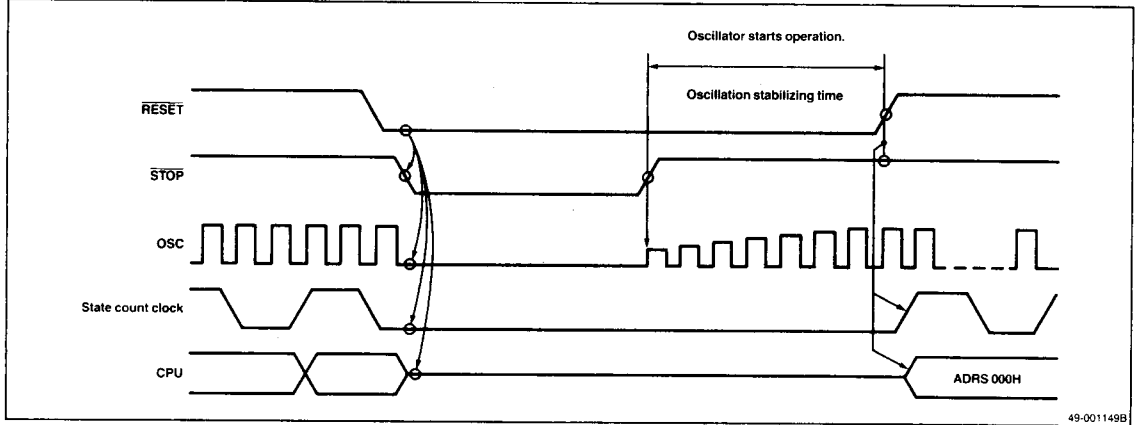
### STOP Mode

The STOP mode disables the oscillator but maintains the contents of RAM. STOP mode conserves even more power than HALT mode. Enter STOP mode through software with the STOP instruction or through hardware with the  $\overline{STOP}$  pin. In hardware STOP mode, the power supply voltage can drop as low as 2.0V. In software STOP mode, it can drop as low as 2.5V while still maintaining the RAM contents.

Control the STOP mode with hardware, with the  $\overline{RESET}$  and  $\overline{STOP}$  pins, as follows:

- Bring  $\overline{RESET}$  low for at least six machine cycles, then bring  $\overline{STOP}$  low. This assures proper termination of CPU operations. Figure 4 shows the timing for controlling STOP mode with hardware.

Figure 4. STOP Mode Control Timing



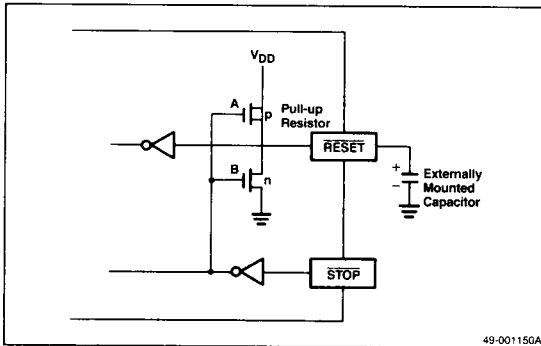
- Release hardware STOP mode by returning  $V_{CC}$  to  $+5V \pm 10\%$ . After STOP goes high, hold RESET low long enough to allow the oscillator to stabilize. Figure 5 shows how to control oscillator settling time with the STOP pin by adding an external capacitor to the RESET line.
- Release the software STOP modes by applying a low level to the RESET pin to initiate oscillator operation. After sufficient oscillator stabilization time has passed, return RESET to a high level. Program execution will then begin at address 0.

The following table shows the states of the output pins during both hardware and software STOP mode.

Table 1. Output Pins During STOP Mode

Output Pin	State		
	STOPZ Instruction	STOPH Instruction	Hardware STOP
P1 <sub>0</sub> -P1 <sub>7</sub> , P2 <sub>0</sub> -P2 <sub>7</sub>	High-Z	High level	High level
D <sub>0</sub> -D <sub>7</sub>	High-Z	High-Z	High-Z
PROG	High level	High level	High level
SYNC	Low level	Low level	Low level

Figure 5. STOP Mode Control Circuit



## Instruction Set

Mnemonic	Operation	Description	Operation Code								Cycles	Bytes	
			D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>			
<b>Accumulator</b>													
ADD A, # data	(A), (C) ← (A) + data	Add immediate the specified data to the accumulator.(2)	0	0	0	0	0	0	1	1	1	2	2
	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>					
ADD A, Rr	(A), (C) ← (A) + (Rr) r = 0-7	Add contents of designated register to the accumulator.(2)	0	1	1	0	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>	1	1	1
ADD A, @ Rr	(A), (C) ← (A) + ((Rr)) r = 0-1	Add indirect the contents the data memory location to the accumulator.(2)	0	1	1	0	0	0	0	r <sub>0</sub>	1	1	1
ADDC A, # data	(A), (C) ← (A) + (C) + data	Add immediate with carry the specified data to the accumulator.(2)	0	0	0	1	0	0	1	1	2	2	2
	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>					
ADDC A, Rr	(A), (C) ← (A) + (C) + (Rr) r = 0-7	Add with carry the contents of the designated register to the accumulator.(2)	0	1	1	1	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>	1	1	1
ADDC A, @ Rr	(A), (C) ← (A) + (C) + ((Rr)) r = 0-1	Add indirect with carry the contents of data memory location to the accumulator.(2)	0	1	1	1	0	0	0	r <sub>0</sub>	1	1	1
ANL A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	0	1	0	1	0	0	1	1	2	2	2
	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>					
ANL A, Rr	(A) ← (A) AND (Rr) r = 0-7	Logical AND contents of designated register with accumulator.	0	1	0	1	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>	1	1	1
ANL A, @ Rr	(A) ← (A) AND ((Rr)) r = 0-1	Logical AND indirect the contents of data memory with accumulator.	0	1	0	1	0	0	0	r <sub>0</sub>	1	1	1
CPL A	(A) ← NOT (A)	Complement the contents of the accumulator.	0	0	1	1	0	1	1	1	1	1	1
CLR A	(A) ← 0	Clear the contents of the accumulator.	0	0	1	0	0	1	1	1	1	1	1
DA A		Decimal adjust the contents of the accumulator.(2)	0	1	0	1	0	1	1	1	1	1	1
DEC A	(A) ← (A) - 1	Decrement by 1 the accumulator's contents.	0	0	0	0	0	1	1	1	1	1	1
INC A	(A) ← (A) + 1	Increment by 1 the accumulator's contents.	0	0	0	1	0	1	1	1	1	1	1
ORL A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	0	1	0	0	0	1	1	1	2	2	2
	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>					
ORL A, Rr	(A) ← (A) OR (Rr); r = 0-7	Logical OR contents of designated register with accumulator.	0	1	0	0	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>	1	1	1
ORL A, @ Rr	(A) ← (A) OR ((Rr)) r = 0-1	Logical OR indirect the contents of data memory location with accumulator.	0	1	0	0	0	0	0	r <sub>0</sub>	1	1	1
RL A	(A <sub>n</sub> +1) ← (A <sub>n</sub> ), (A <sub>0</sub> ) ← (A <sub>7</sub> ) n = 0-6	Rotate accumulator left by 1 bit without carry.	1	1	1	0	0	1	1	1	1	1	1
RLC A	(A <sub>n</sub> +1) ← (A <sub>n</sub> ), (A <sub>0</sub> ) ← (C) (C) ← (A <sub>7</sub> ) n = 0-6	Rotate accumulator left by 1 bit through carry.	1	1	1	1	0	1	1	1	1	1	1
RR A	(A <sub>n</sub> ) ← (A <sub>n</sub> +1), (A <sub>7</sub> ) ← (A <sub>0</sub> ) n = 0-6	Rotate accumulator right by 1 bit without carry.	0	1	1	1	0	1	1	1	1	1	1

**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code								Cycles	Bytes
			D7	D6	D5	D4	D3	D2	D1	D0		
<b>Accumulator (cont)</b>												
RRC A	$(A_n) \leftarrow (A_{n+1})$ , $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$ $n = 0-6$	Rotate accumulator right by 1 bit through carry.	0	1	1	0	0	1	1	1	1	1
SWAP A	$(A_7-A_4) \leftrightarrow (A_3-A_0)$	Swap the 2 4-bit nibbles in the accumulator.	0	1	0	0	0	1	1	1	1	1
XRL A, # data	$(A) \leftarrow (A) \text{ XOR data}$	Logical XOR specified immediate data with accumulator.	1	1	0	1	0	0	1	1	1	2
			d7	d6	d5	d4	d3	d2	d1	d0		
XRL A, Rr	$(A) \leftarrow (A) \text{ XOR (Rr)}$ $r = 0-7$	Logical XOR contents of designated register with accumulator.	1	1	0	1	1	r2	r1	r0	1	1
XRL A, @ Rr	$(A) \leftarrow (A) \text{ XOR ((Rr))}$ $r = 0-1$	Logical XOR indirect the contents of data memory location with accumulator.	1	1	0	1	0	0	0	r0	1	1
<b>Branch</b>												
DJNZ Rr, addr	$(Rr) \leftarrow (Rr) - 1$ ; if $Rr \neq 0$ ; $(PC_7-PC_0) \leftarrow a7-a_0$ $r = 0-7$	Decrement the specified register and test contents.	1	1	1	0	1	r2	r1	r0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JBB addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $Bb = 1$ $(PC) \leftarrow (PC) + 2$ if $Bb = 0$	Jump to specified address if accumulator bit is set.	b2	b1	b0	1	0	0	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JC addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $C = 1$ $(PC) \leftarrow (PC) + 2$ if $C = 0$	Jump to specified address if carry flag is set.	1	1	1	1	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JFO addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $F0 = 1$ $(PC) \leftarrow (PC) + 2$ if $F0 = 0$	Jump to specified address if flag F0 is set.	1	0	1	1	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JF1 addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $F1 = 1$ $(PC) \leftarrow (PC) + 2$ if $F1 = 0$	Jump to specified address if flag F1 is set.	0	1	1	1	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JMP addr	$(PC_{10}-PC_8) \leftarrow (a7-a_0)$	Direct jump to specified address within the 2K address block.	a10	a9	a8	0	0	1	0	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JMPP @ A	$(PC_7-PC_0) \leftarrow ((A))$	Jump indirect to specified address with address page.	1	0	1	1	0	0	1	1	2	1
JNC addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $C = 0$ $(PC) \leftarrow (PC) + 2$ if $C = 1$	Jump to specified address if carry flag is low.	1	1	1	0	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JNIBF addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $IBF = 0$ $(PC) \leftarrow (PC) + 2$ if $IBF = 1$	Jump to specified address if interrupt is low.	1	1	0	1	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JNTO addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $T0 = 0$ $(PC) \leftarrow (PC) + 2$ if $T0 = 1$	Jump to specified address if test 0 is low.	0	0	1	0	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JNT1 addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $T1 = 0$ $(PC) \leftarrow (PC) + 2$ if $T1 = 1$	Jump to specified address if test 1 is low.	0	1	0	0	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		
JNZ addr	$(PC_7-PC_0) \leftarrow a7-a_0$ if $A \neq 0$ $(PC) \leftarrow (PC) + 2$ if $A = 0$	Jump to specified address if accumulator is non-zero.	1	0	0	1	0	1	1	0	2	2
			a7	a6	a5	a4	a3	a2	a1	a0		

## Instruction Set (cont)

Mnemonic	Operation	Description	Operation Code										Cycles	Bytes
			D7	D6	D5	D4	D3	D2	D1	D0				
<b>Branch (cont)</b>														
JOBF addr	$(PC_7-PC_0) \leftarrow a_7-a_0$ if OBF = 1 $(PC) \leftarrow (PC) + 2$ if OBF = 0	Jump to specified address if output is low.	1	0	0	0	0	0	1	1	0	0	2	2
JTF addr	$(PC_7-PC_0) \leftarrow a_7-a_0$ if TF = 1 then reset TF $(PC) \leftarrow (PC) + 2$ if TF = 0	Jump to specified address if timer flag is set to 1.	0	0	0	1	0	1	0	1	0	0	2	2
JTO addr	$(PC_7-PC_0) \leftarrow a_7-a_0$ if TO = 1 $(PC) \leftarrow (PC) + 2$ if TO = 0	Jump to specified address if test 0 is a 1.	0	0	1	1	0	1	1	0	0	0	2	2
JT1 addr	$(PC_7-PC_0) \leftarrow a_7-a_0$ if T1 = 1 $(PC) \leftarrow (PC) + 2$ if T1 = 0	Jump to specified address if test 1 is a 1.	0	1	0	1	0	1	0	1	1	0	2	2
JZ addr	$(PC_7-PC_0) \leftarrow a_7-a_0$ if A = 0 $(PC) \leftarrow (PC) + 2$ if A = 1	Jump to specified address if accumulator is 0.	1	1	0	0	0	1	1	0	0	0	2	2
<b>Control</b>														
EN I		Enable the interrupt.	0	0	0	0	0	0	1	0	1	0	1	1
DIS I		Disable the external interrupt input.	0	0	0	1	0	1	0	1	0	1	1	1
EN DMA		Enables DMA handshake lines.	1	1	1	0	0	1	0	1	0	1	1	1
EN Flags		Enables master interrupts.	1	1	1	1	0	1	0	1	0	1	1	1
SEL RBO	$(BS) \leftarrow 0$	Select bank 0 (locations 0-7) of data memory.	1	1	0	0	0	1	0	1	0	1	1	1
SEL RB1	$(BS) \leftarrow 1$	Select bank 1 (locations 24-31) of data memory.	1	1	0	1	0	1	0	1	0	1	1	1
HALT		Initiates halt mode.	0	0	0	0	0	0	0	0	0	1	1	1
STOP Z		Sets CPU to software stop mode. (Port output high impedance)	1	0	0	0	0	0	0	1	0	1	1	1
STOP H		Sets CPU to software stop mode. (Port output high level)	1	1	1	0	0	0	1	0	1	0	1	1
<b>Data Moves</b>														
MOV A, # data	$(A) \leftarrow \text{data}$	Move immediate the specified data into the accumulator.	0	0	1	0	0	0	1	1	1	1	2	2
MOV A, Rr	$(A) \leftarrow (Rr); r = 0-7$	Move the contents of the designated registers into the accumulator.	d7	d6	d5	d4	d3	d2	d1	d0				
MOV A, @ Rr	$(A) \leftarrow ((Rr)); r = 0-1$	Move indirect the contents of data memory location into the accumulator.	1	1	1	1	1	1	r2	r1	r0		1	1
MOV A, PSW	$(A) \leftarrow (PSW)$	Move contents of the program status word into the accumulator.	1	1	0	0	0	1	1	1	1		1	1
MOV Rr, # data	$(Rr) \leftarrow \text{data}; r = 0-7$	Move immediate the specified data into the designated register.	1	0	1	1	1	1	r2	r1	r0		2	2
MOV Rr, A	$(Rr) \leftarrow (A); r = 0-7$	Move accumulator contents into the designated register.	1	0	1	0	1	1	r2	r1	r0		1	1
MOV @ Rr, A	$((Rr)) \leftarrow (A); r = 0-1$	Move indirect accumulator contents into data memory location.	1	0	1	0	0	0	0	0	r0		1	1
MOV @ Rr, # data	$((Rr)) \leftarrow \text{data}; r = 0-1$	Move immediate the specified data into data memory.	1	0	1	1	1	1	0	0	0	r0		2
MOV PSW, A	$(PSW) \leftarrow (A)$	Move contents of accumulator into the program status word.	1	1	0	1	0	1	1	1	1		1	1

**Instruction Set (cont)**

Mnemonic	Operation	Description	Operation Code								Cycles	Bytes									
			D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>											
<b>Data Moves (cont)</b>																					
MOVPA, @A	$A \leftarrow ((PC_{10}-PC_8), (A))$	Move data in the current page into the accumulator.	1	0	1	0	0	0	1	1	1	2	1								
MOVPA, @A	$(A) \leftarrow ((011, A))$	Move program data in page 3 into the accumulator.	1	1	1	0	0	0	1	1	1	2	1								
XCH A, Rr	$(A) \leftrightarrow (Rr); r = 0-7$	Exchange the accumulator and designated register's contents.	0	0	1	0	1	r <sub>2</sub>	r <sub>1</sub>	r <sub>0</sub>		1	1								
XCH A, @Rr	$(A) \leftrightarrow (Rr); r = 0-1$	Exchange indirect contents of accumulator and location in data memory.	0	0	1	0	0	0	0	r <sub>0</sub>		1	1								
XCHD A, @Rr	$(A_3-A_0) \leftrightarrow ((Rr)_3-(Rr)_0); r = 0-1$	Exchange indirect 4-bit contents of accumulator and data memory.	0	0	1	1	0	0	0	r <sub>0</sub>		1	1								
<b>Flags</b>																					
CPL C	$(C) \leftarrow \text{NOT}(C)$	Complement contents of carry bit.	1	0	1	0	0	1	1	1	1	1	1								
CPL F0	$(F0) \leftarrow \text{NOT}(F0)$	Complement contents of flag F0.	1	0	0	1	0	1	0	1	1	1	1								
CPL F1	$(F1) \leftarrow \text{NOT}(F1)$	Complement contents of flag F1.	1	0	1	1	0	1	0	1	1	1	1								
CLRC	$(C) \leftarrow 0$	Clear contents of carry bit to 0.	1	0	0	1	0	1	1	1	1	1	1								
CLRF0	$(F0) \leftarrow 0$	Clear contents of flag 0 to 0.	1	0	0	0	0	1	0	1	1	1	1								
CLRF1	$(F1) \leftarrow 0$	Clear contents of flag 1 to 0.	1	0	1	0	0	1	0	1	1	1	1								
<b>Input / Output</b>																					
ANL Pp, # data	$(Pp) \leftarrow (Pp) \text{ AND data}$ $p = 1-2$	Logical AND immediate specified data with designated port (1 or 2).	1	0	0	1	1	0	p <sub>1</sub>	p <sub>0</sub>	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	2	2	
ANLD Pp, A	$(Pp) \leftarrow (Pp) \text{ AND } (A_3-A_0); p = 4-7$	Logical AND contents of accumulator with designated port (4-7).	1	0	0	1	1	1	p <sub>1</sub>	p <sub>0</sub>									2	1	
IN A, DBB	$(A) \leftarrow (\text{DBBIN}); \text{IBF} \leftarrow 0$		0	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1
IN A, Pp	$(A) \leftarrow (Pp); p = 1-2$	Input data from designated port (1-2) into accumulator.	0	0	0	0	1	0	p <sub>1</sub>	p <sub>2</sub>									2	1	1
MOVD A, Pp	$(A_3-A_0) \leftarrow (Pp); (A_7-A_4) \leftarrow 0; p = 4-7$	Move contents of designated port (4-7) into accumulator.	0	0	0	0	1	1	p <sub>1</sub>	p <sub>0</sub>									2	1	1
MOVED Pp, A	$(Pp) \leftarrow (A_3-A_0); p = 4-7$	Move contents of accumulator to designated port (4-7).	0	0	1	1	1	1	p <sub>1</sub>	p <sub>0</sub>									2	1	1
MOV STS, A	$(ST_7-ST_4) \leftarrow (A_7-A_4)$	Move contents of accumulator to designated port (4-7).	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

## Instruction Set (cont)

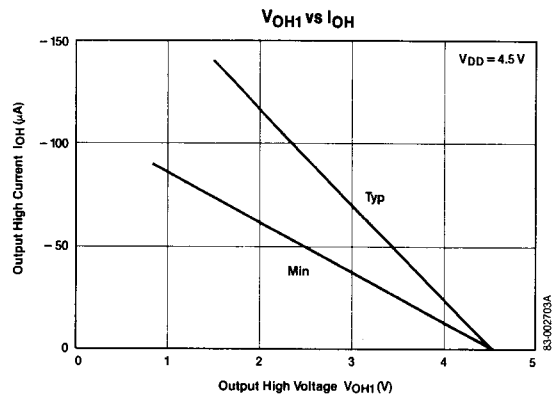
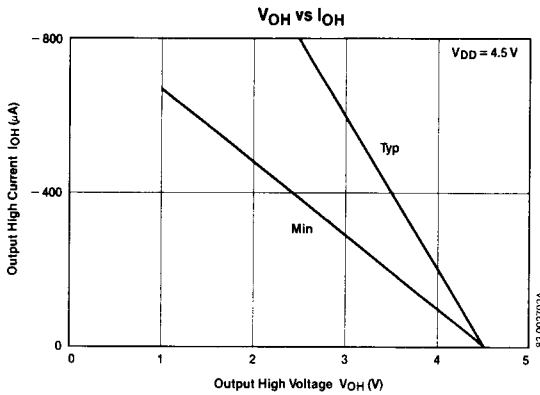
Mnemonic	Operation	Description	Operation Code																Bytes	
			D7	D6	D5	D4	D3	D2	D1	D0										
<b>Input / Output (cont)</b>																				
ORLD Pp, A	(Pp) ← ((Pp) OR (A3-A0)); p = 4-7	Logical OR contents of accumulator with designated port (4-7).	1	0	0	0	1	1	p1	p0									2	1
ORL Pp, # data	(Pp) ← ((Pp) OR data) p = 1-2	Logical OR immediate specified data with designated port (1-2).	1	0	0	0	1	0	p1	p0	d7	d6	d5	d4	d3	d2	d1	d0	2	
OUT DBB, A	(DBBOUT) ← (A), OBF ← 1		0	0	0	0	0	0	1	0									1	
OUTL Pp, A	(Pp) ← (A); p = 1-2	Output contents of accumulator to designated port (1-2).	0	0	1	1	1	0	p1	p0									2	1
<b>Registers</b>																				
DEC Rr	(Rr) ← (Rr) - 1; r = 0-7	Decrement by 1 contents of designated register.	1	1	0	0	1	r2	r1	r0									1	
INC Rr	(Rr) ← (Rr) + 1; r = 0-7	Increment by 1 contents of designated register.	0	0	0	1	1	r2	r1	r0									1	
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0-1	Increment indirect by 1 the contents of data memory location.	0	0	0	1	0	0	0	r0									1	
<b>Subroutine</b>																				
CALL addr	((SP)) ← (PC), (PSW7-PSW4) (SP) ← (SP) + 1 (PC10-PC0) ← a10-a0	Call designated subroutine.	a10	a9	a8	1	0	1	0	0	a7	a6	a5	a4	a3	a2	a1	a0	2	
RET	(SP) ← (SP) - 1 (PC) ← ((SP))	Return from subroutine without restoring program status word.	1	0	0	0	0	0	1	1									2	
RETR	(SP) ← (SP) - 1 (PC) ← ((SP)) (PSW7-PSW4) ← ((SP))	Return from subroutine restoring program status word.	1	0	0	1	0	0	1	1									2	
<b>Timer / Counter</b>																				
EN TCNTI		Enable internal interrupt flag for timer / counter output.	0	0	1	0	0	1	0	1									1	
DIS TCNTI		Disable internal interrupt flag for timer / counter output.	0	0	1	1	0	1	0	1									1	
MOV A, T	(A) ← (T)	Move contents of timer / counter into accumulator.	0	1	0	0	0	0	1	0									1	
MOV T, A	(T) ← (A)	Move contents of accumulator into timer / counter.	0	1	1	0	0	0	1	0									1	
STOP TCNT		Stop count for event counter.	0	1	1	0	0	1	0	1									1	
START CNT		Start count for event counter.	0	1	0	0	0	1	0	1									1	
START T		Start count for timer.	0	1	0	1	0	1	0	1									1	
<b>Miscellaneous</b>																				
NOP		No operation performed.	0	0	0	0	0	0	0	0									1	

**Symbol Definitions**

Symbol	Description
A	Accumulator
AC	Auxiliary carry flag
addr	Program memory address
b	Accumulator bit (b = 0-7)
C	Carry flag
CNT	Counter
data	8-bit data
DBB	Data bus buffer
F0, F1	Flags 0, 1 (C / D flag)
I	Interrupt
IBF	Input buffer full flag
OBF	Output buffer full flag
PC	Program counter
Pp	Port (p = 1-2 or 4-7)
PSW	Program status word
Rr	Register (r = 0-1 or r = 0-7)

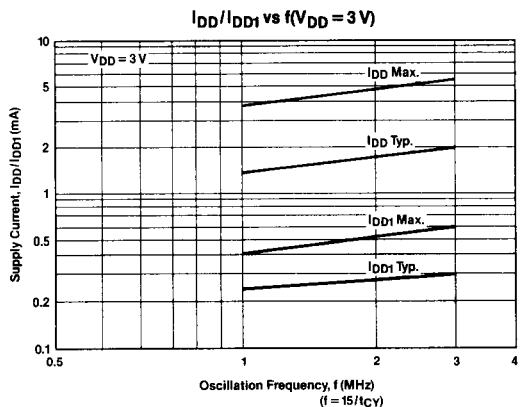
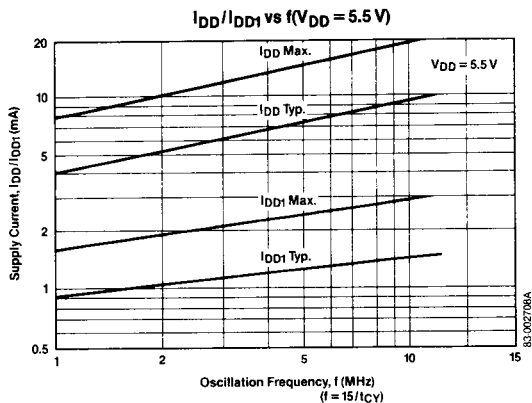
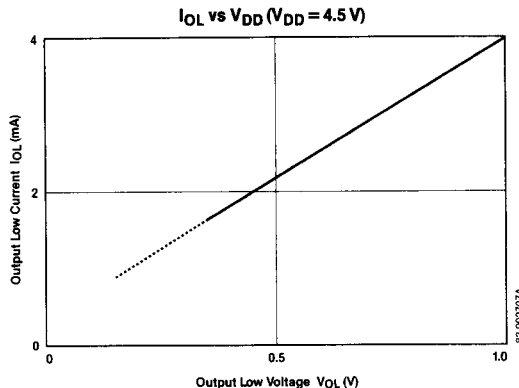
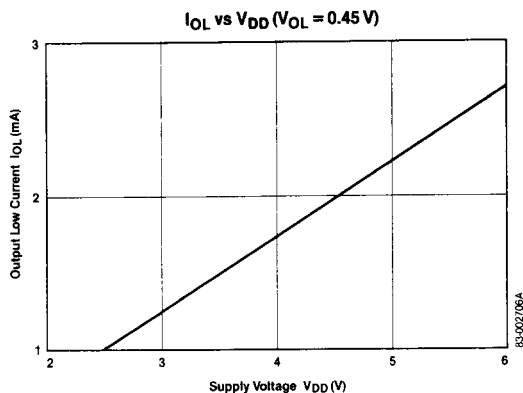
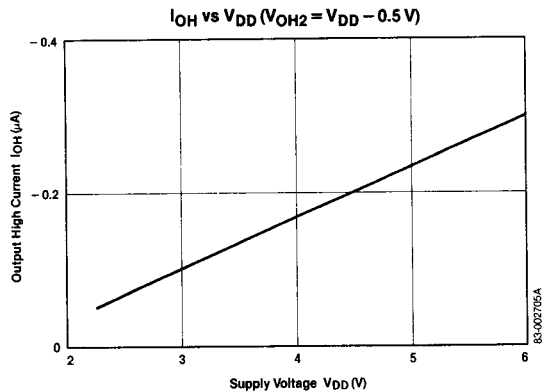
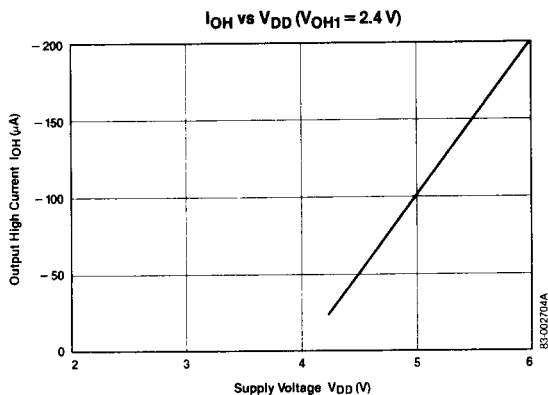
Symbol	Description
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	TEST0, TEST1 pin
#	Immediate data
@	Indirect address
(x)	Contents of register X
((x))	Contents of memory addressed by X
←	Transfer direction, result
AND	Logical product (logical AND)
OR	Logical sum (logical OR)
XOR	Exclusive OR
—	Complement

**Operating Characteristics**





## Operating Characteristics (cont)



## Operating Characteristics (cont)

