

Description

The μ PD80C42 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 μ PD80C42 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 μ PD82C43, which interfaces directly to the μ PD80C42. 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 STOP pin, the μ PD80C42 is pin-for-pin compatible with the μ PD8041A and the μ PD8741A.

Features

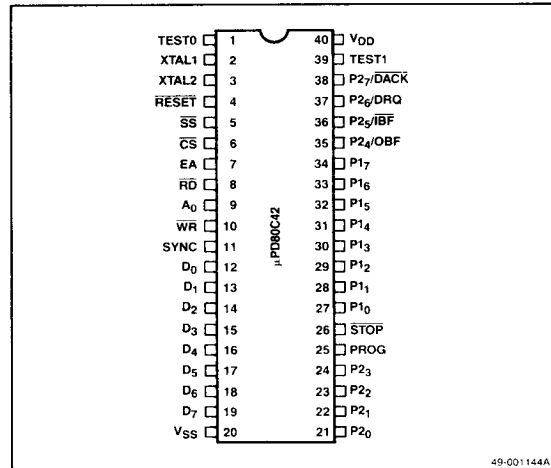
- CMOS technology
- Low power consumption
- 8048-, 8085A-, and 8086-bus compatible
- 8-bit CPU with 2K \times 8 ROM and 128 \times 8 RAM
- 8-bit timer/counter
- 18 I/O lines
- 8-bit status register
- Two data registers for asynchronous slave-to-master interface
- Interrupt, DMA, or polled operation
-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

Ordering Information

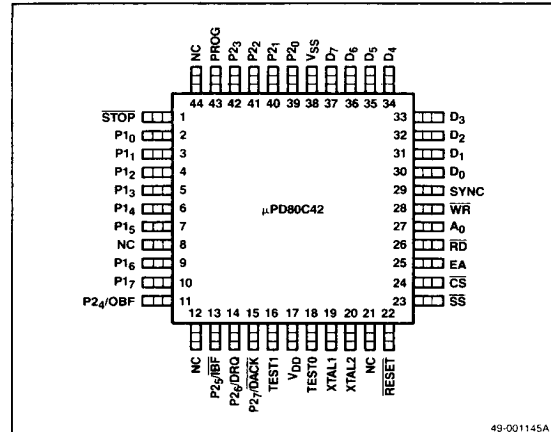
| Part Number | Package Type | Max Frequency of Operation |
|-------------------|-------------------------|----------------------------|
| μ PD80C42C | 40-pin plastic DIP | 8 MHz |
| μ PD80C42G-22 | 44-pin plastic miniflat | 8 MHz |

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 ₀ | Address input |
| 10 | WR | Write input |
| 11 | SYNC | Synchronize output |
| 12–19 | D ₀ –D ₇ | Bidirectional port |
| 20 | V _{SS} | Ground |
| 21–24 | P ₂₀ –P ₂₃ | Quasi-bidirectional port 2 |
| 35–38 | P ₂₄ / OBF, P ₂₅ / IBF, P ₂₆ / DRQ, P ₂₇ / DACK | Output buffer full, input buffer full, DMA request, DMA acknowledge |
| 25 | PROG | PROG output strobe |
| 26 | STOP | STOP input |
| 27–34 | P ₁₀ –P ₁₇ | Quasi-bidirectional port 1 |
| 39 | TEST1 | Test 1 input |
| 40 | V _{DD} | 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 ₀ | Address input |
| 28 | WR | Write input |
| 29 | SYNC | Synchronize output |
| 30–37 | D ₀ –D ₇ | Bidirectional port |
| 38 | V _{SS} | Ground |
| 39–42 | P ₂₀ –P ₂₃ | Quasi-bidirectional port 2 |
| 11, 13–15 | P ₂₄ / OBF, P ₂₅ / IBF, P ₂₆ / DRQ, P ₂₇ / 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 ₁₀ –P ₁₇ | Quasi-bidirectional port 1 |
| 16 | TEST1 | Test 1 input |
| 17 | V _{DD} | 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.

A₀ (Address 0)

A₀ is an address input that the master CPU uses to determine the bus operation as follows:

| Cycle | A ₀ | 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 P₂₀-P₂₃.

STOP (Stop)

The STOP input controls the hardware STOP mode.

D₀-D₇ (Port)

D₀-D₇ is a bidirectional port that transfers data between the data bus buffer (DBBOUT, DBBIN) registers and the 8-bit master CPU data bus.

P₁₀-P₁₇ (Port 1)

P₁₀-P₁₇ is a quasi-bidirectional, 8-bit port.

P₂₀-P₂₇ (Port 2)

P₂₀-P₂₇ is a quasi-bidirectional, programmable 8-bit port. P₂₄-P₂₇ (high-order bits) are alternative pins for the following interrupt request and DMA handshaking functions:

P₂₄ = OBF (Output buffer full)

P₂₅ = IBF (Input buffer full)

P₂₆ = DRQ (DMA request)

P₂₇ = DACK (DMA acknowledge)

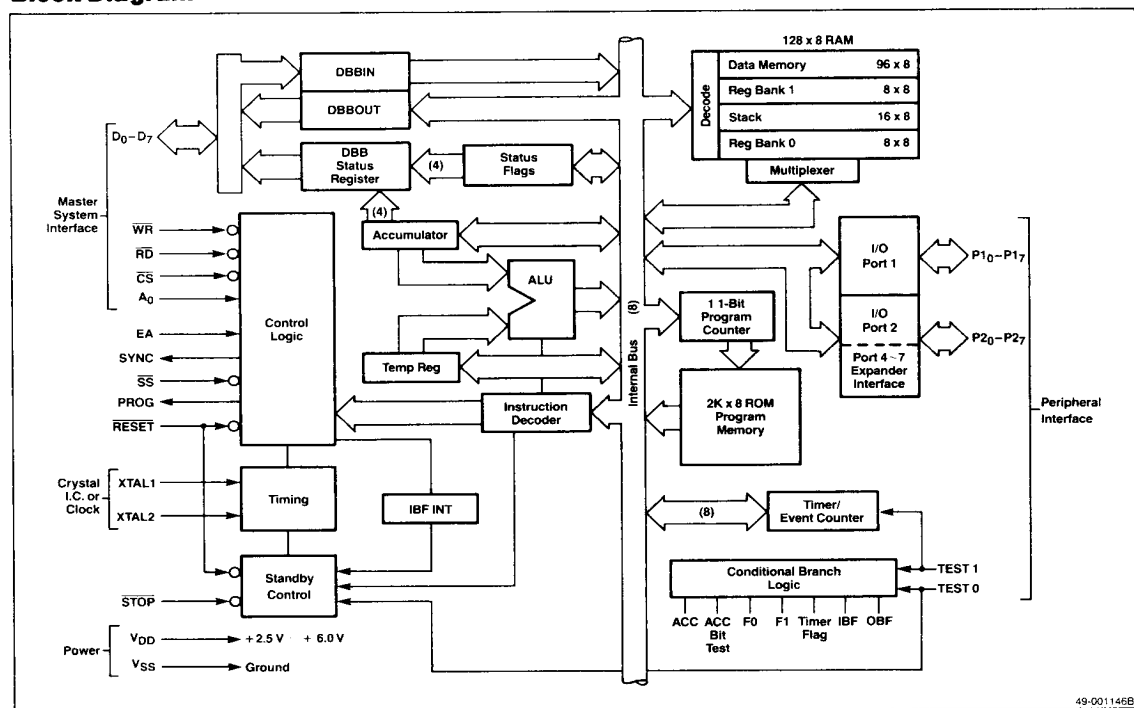
V_{DD} (Power Supply)

V_{DD} is the positive power supply (+2.5V to +6.0V)

V_{SS} (Ground)

V_{SS} is the ground potential.

Block Diagram



49-001146B

Absolute Maximum Ratings

T_A = 25°C

| | |
|--|-----------------------------------|
| Power supply voltage, V _{DD} | -0.3 V to +7 V |
| Input voltage, V _I | -0.3 V to V _{DD} + 0.3 V |
| Output voltage, V _O | -0.3 V to V _{DD} + 0.3 V |
| Operating temperature, T _{OP} | -40°C to +85°C |
| Storage temperature, T _{STG} | -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_A = -40°C to +85°C, V_{DD} = +5 V ± 10%, V_{SS} = 0 V

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

Note: (1) The input voltage pin is V_I ≤ V_{IL} or V_I ≥ V_{IH}.

Extended Voltage Range

T_A = -40°C to +85°C, V_{DD} = +2.5 V to +6.0 V, V_{SS} = 0 V

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

Note:

(1) The input voltage pin is V_I ≤ V_{IL} or V_I ≥ V_{IH}.

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 | | |
| CS, A ₀ setup to RD low | t _{AR} | 0 | | | ns | |
| CS, A ₀ hold from RD high | t _{RA} | 0 | | | ns | |
| RD pulse width | t _{RR} | 200 | | | ns | |
| CS, A ₀ to data output delay | t _{AD} | | | 150 | ns | C _L = 100 pF |
| RD low to data output delay | t _{RD} | | | 140 | ns | C _L = 100 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 | | |
| CS, A ₀ setup to WR low | t _{AW} | 0 | | | ns | |
| CS, A ₀ hold from WR high | t _{WA} | 0 | | | ns | |
| WR pulse width data setup to WR high | t _{WW} t _{DW} | 200 | | | 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 | | |
| CS, A ₀ setup to RD low | t _{AR} | 300 | | | ns | |
| 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 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 6.0V , $V_{SS} = 0\text{V}$

| Parameter | Symbol | Limits | | | Unit | Test Conditions |
|--------------------------------------|------------------------------------|--------|-----|-----|------|-----------------|
| | | Min | Typ | Max | | |
| CS, A ₀ setup to WR low | t _{AW} | 300 | | | ns | |
| CS, A ₀ hold from WR high | t _{WA} | 200 | | | ns | |
| WR pulse width data setup to WR high | t _{WW} t _{DW} | 2000 | | | 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 pF |
| Input port control hold from PROG low | t _{PC1} | 0 | | 80 | ns | C _L = 20 pF |
| Output port control hold from PROG low | t _{PC2} | 135 | | | ns | C _L = 20 pF |
| Input data setup to PROG low | t _{PR} | | | 650 | ns | C _L = 80 pF |
| Input data hold from PROG high | t _{PF} | 0 | | 150 | ns | C _L = 20 pF |
| Output data setup to PROG high | t _{DP} | 200 | | | ns | C _L = 80 pF |
| Output data hold from PROG high | t _{PD} | 60 | | | ns | C _L = 20 pF |
| PROG pulse width | t _{PP} | 700 | | | ns | |

AC Characteristics (cont)

Extended Voltage Range — Port 2

V_{DD} = +2.5V to +6.0V

| Parameter | Symbol | Limits | | | Unit | Test Conditions |
|--|------------------|--------|-----|------|------|------------------------|
| | | Min | Typ | Max | | |
| Port control setup to PROG low | t _{CP} | 460 | | | ns | C _L = 80 pF |
| Input port control hold from PROG low | t _{PC1} | 0 | | 200 | ns | C _L = 20 pF |
| Output port control hold from PROG low | t _{PC2} | 1135 | | | ns | C _L = 20 pF |
| Input data setup to PROG low | t _{PR} | | | 2715 | ns | C _L = 80 pF |
| Input data hold from PROG high | t _{PF} | 0 | | 500 | ns | C _L = 20 pF |
| Output data setup to PROG high | t _{DP} | 1850 | | | ns | C _L = 80 pF |
| Output data hold from PROG high | t _{PD} | 450 | | | ns | C _L = 20 pF |
| PROG pulse width | t _{PP} | 3250 | | | ns | |

Standard Voltage Range — DMA

V_{DD} = +5V ±10%

| Parameter | Symbol | Limits | | | Unit | Test Conditions |
|---------------------------|------------------|--------|-----|-----|------|-------------------------|
| | | Min | Typ | Max | | |
| DACK setup to RD, WR | t _{ACC} | 0 | | | ns | |
| DACK hold from RD, WR | t _{CAC} | 0 | | | ns | |
| DACK to data output delay | t _{ACD} | | | 140 | ns | |
| RD, WR to DRQ clear delay | t _{CRQ} | | | 130 | ns | C _L = 150 pF |

Extended Voltage Range — DMA

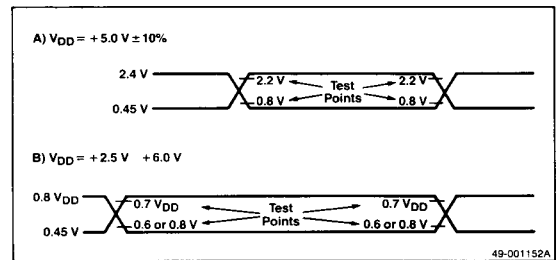
V_{DD} = +2.5V to +6.0V

| Parameter | Symbol | Limits | | | Unit | Test Conditions |
|---------------------------|------------------|--------|-----|------|------|-------------------------|
| | | Min | Typ | Max | | |
| DACK setup to RD, WR | t _{ACC} | 200 | | | ns | |
| DACK hold from RD, WR | t _{CAC} | 200 | | | ns | |
| DACK to data output delay | t _{ACD} | | | 1500 | ns | |
| RD, WR to DRQ clear delay | t _{CRQ} | | | 700 | ns | C _L = 150 pF |

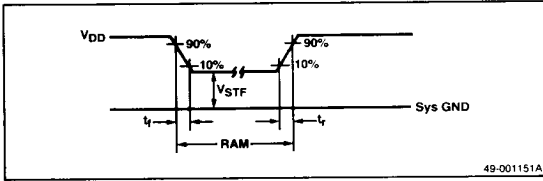
Standby Flag Retention Conditions

| Parameter | Symbol | Limits | | | Unit | Test Conditions |
|--|------------------|--------|-----|-----|------|-----------------|
| | | Min | Typ | Max | | |
| Preservation of standby flag voltage fall time | t _f | 100 | | | μs | |
| Preservation of standby flag voltage rise time | t _r | 100 | | | μs | |
| Standby flag retention voltage | V _{STF} | 2.0 | | | V | |

Input Waveforms for AC Test



Standby Flag Retention Timing

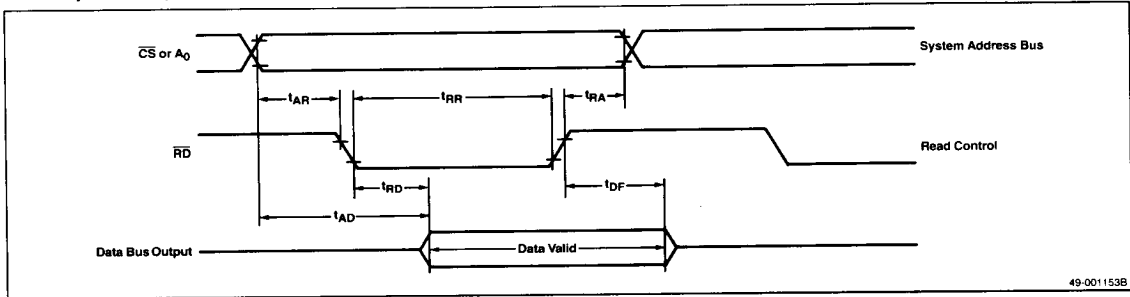


Bus Timing Requirements

| 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 |

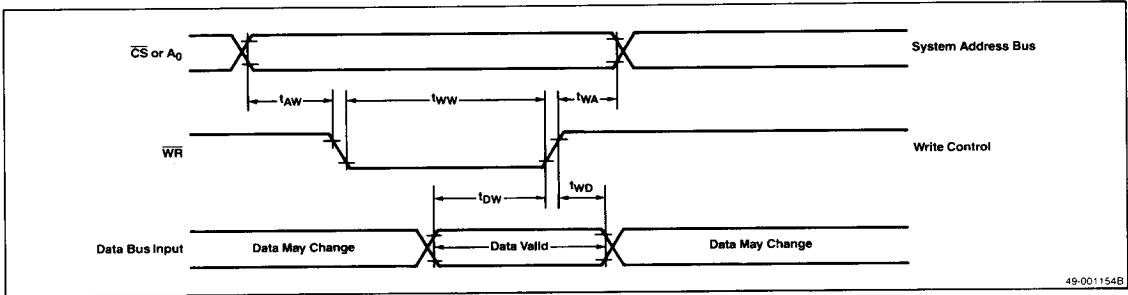
Timing Waveforms

Read Operation (DBBOUT Register)



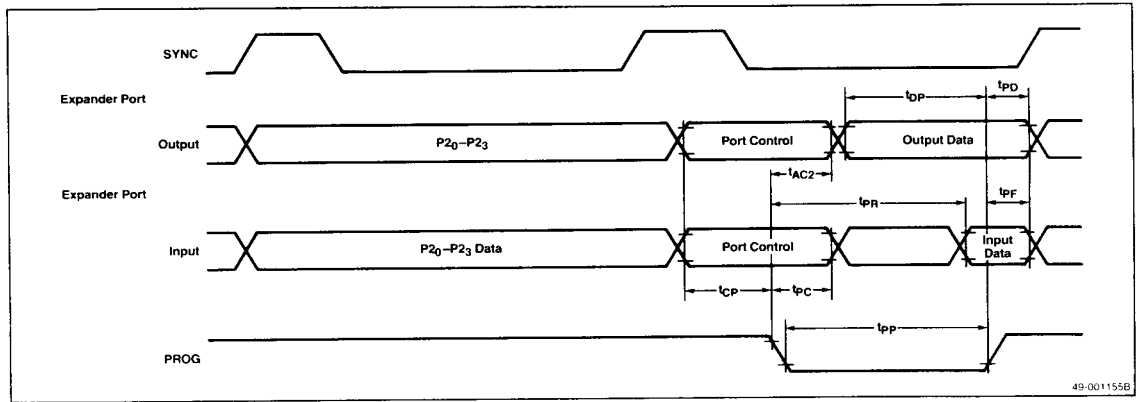
4

Write Operation (DBBIN Register)

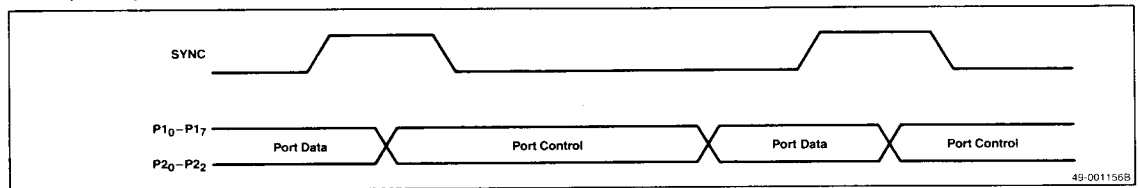


Timing Waveforms (cont)

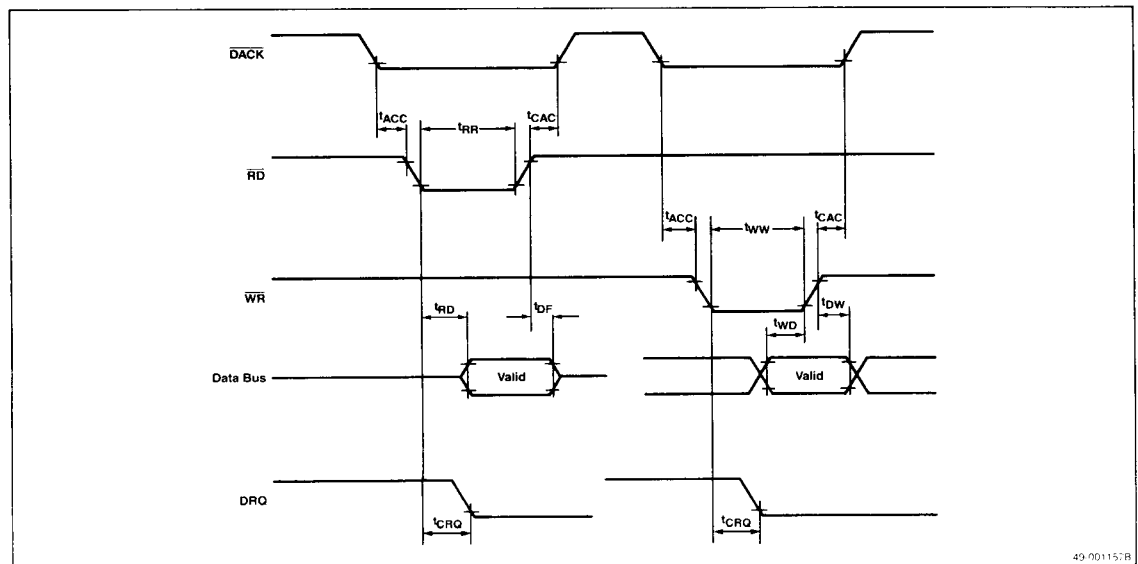
PORT2



PORT (EA = 1)



DMA

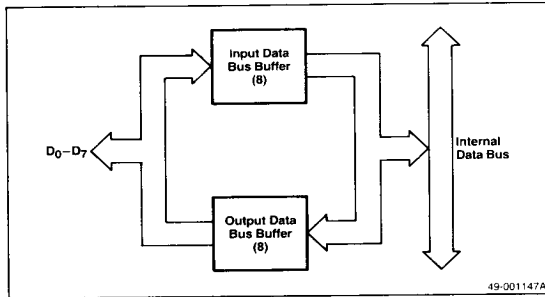


Functional Description

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 (D₀-D₇) 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 (ST₀-ST₇) that contains information about the current status of the master or slave processor. The MOV STS, A instruction makes status bits ST₄-ST₇ user-definable by moving accumulator bits 4-1 to bits ST₄-ST₇ of the status register (ST₀-ST₃ are not affected). Bits ST₀-ST₃ give the status of the Output Buffer Full (OBF) and Input Buffer Full (IBF) bits, and flag bits (F₀, F₁). Figure 2 shows the status register format.

Figure 2. Status Register Format

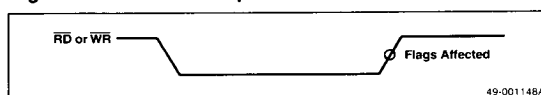
| ST ₇ | ST ₆ | ST ₅ | ST ₄ | ST ₃ | ST ₂ | ST ₁ | ST ₀ |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| UD | UD | UD | UD | F ₁ | F ₀ | IBF | OBF |

The MOV STS, A instruction is coded as follows:

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

Figure 3 shows how ST₀-ST₃ 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 ST₀ (OBF) and ST₁ (IBF) externally available in order to interrupt the master processor by executing the EN FLAGS instruction. When the EN FLAGS instruction is executed, P₂₄ becomes the OBF pin. A 1 written to P₂₄ enables OBF and outputs its status. A 0 written to P₂₄ 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 P₂₅ as the \overline{IBF} pin. A 1 written to P₂₅ enables \overline{IBF} to output the inverse of the IBF status bit. A 0 written to P₂₅ 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
```

P₂₆ and P₂₇ are port pins or DMA handshake pins that allow a DMA interface. Use the EN DMA instruction to enable P₂₆ and P₂₇ as DRQ (DMA Request) and DACK (DMA Acknowledge), respectively. A 1 written to P₂₆ 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, P₂₇ (\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.

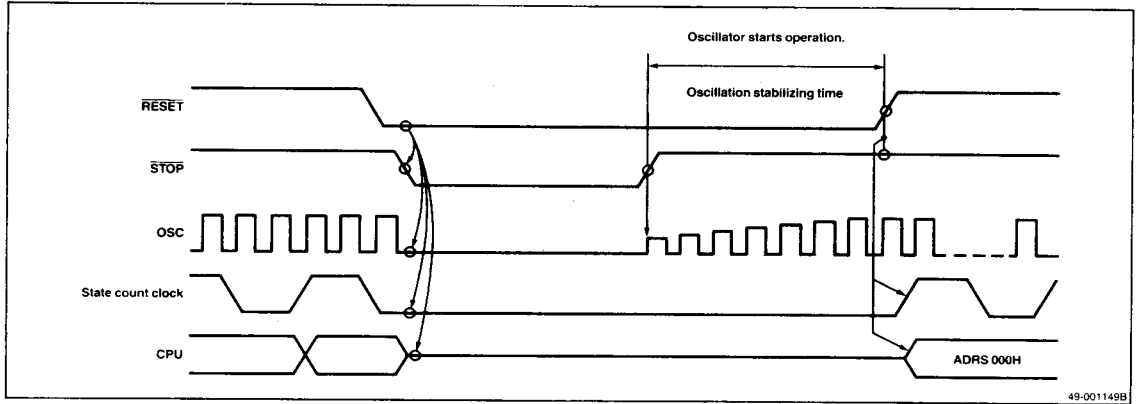
STOP Mode

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



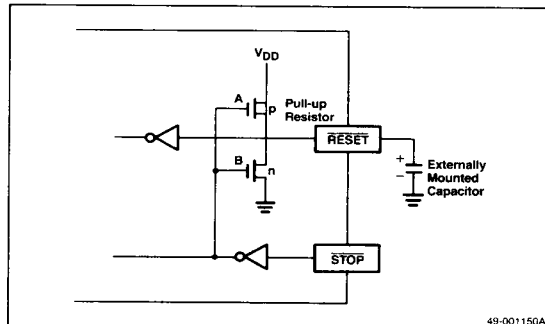
- Release hardware STOP mode by returning V_{CC} to $+5V \pm 10\%$. After STOP goes high, hold RESET low long enough to allow the oscillator to stabilize. Figure 5 shows how to control oscillator settling time with the STOP pin by adding an external capacitor to the RESET line.
- Release the software 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 ₀ -P1 ₇ , P2 ₀ -P2 ₇ | High-Z | High level | High level |
| D ₀ -D ₇ | 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 | |
|--------------------|---|---|----------------|----|----|----|----|----|----|----|--------|-------|---|
| | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | |
| Accumulator | | | | | | | | | | | | | |
| 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 |
| | | | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | | | |
| ADD A, Rr | (A), (C) ← (A) + (Rr) r = 0-7 | Add contents of designated register to the accumulator.(2) | 0 | 1 | 1 | 0 | 1 | r2 | r1 | r0 | | 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 | r0 | | 1 | 1 |
| ADD C 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 | 1 | 2 | 2 |
| | | | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | | | |
| ADD C 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 | r2 | r1 | r0 | | 1 | 1 |
| ADD C 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 | r0 | | 1 | 1 |
| ANL A, # data | (A) ← (A) AND data | Logical AND specified immediate data with accumulator. | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 2 | 2 |
| | | | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | | | |
| ANL A, Rr | (A) ← (A) AND (Rr) r = 0-7 | Logical AND contents of designated register with accumulator. | 0 | 1 | 0 | 1 | 1 | r2 | r1 | r0 | | 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 | r0 | | 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 | 0 | 1 | 1 | 1 | 2 | 2 |
| | | | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | | | |
| ORL A, Rr | (A) ← (A) OR (Rr); r = 0-7 | Logical OR contents of designated register with accumulator. | 0 | 1 | 0 | 0 | 1 | r2 | r1 | r0 | | 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 | r0 | | 1 | 1 |
| RL A | (An + 1) ← (An), (A0) ← (A7); n = 0-6 | Rotate accumulator left by 1 bit without carry. | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| RLC A | (An + 1) ← (An), (A0) ← (C) (C) ← (A7); n = 0-6 | Rotate accumulator left by 1 bit through carry. | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| RR A | (An) ← (An + 1), (A7) ← (A0); 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 | | |
| Accumulator (cont) | | | | | | | | | | | | |
| 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, # ddata | $(A) \leftarrow (A) \text{ XOR data}$ | 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 |
| Branch | | | | | | | | | | | | |
| DJNZ Rr, addr | $(Rr) \leftarrow (Rr) - 1$; if Rr ≠ 0: $(PC_7-PC_0) \leftarrow a7-a0$ 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 | | |
| JFI 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 | $(PC_{10}-PC_0) \leftarrow (a7-a_0)$ | 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 | | |
| JNTI 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 ≠ 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 | | |
| Branch (cont) | | | | | | | | | | | | |
| JOBP addr | (PC ₇ -PC ₀) ← a ₇ -a ₀ if OBF = 1 (PC) ← (PC) + 2 if OBF = 0 | Jump to specified address if output is low. | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 2 |
| JTF addr | (PC ₇ -PC ₀) ← a ₇ -a ₀ if TF = 1 then reset TF (PC) ← (PC) + 2 if TF = 0 | Jump to specified address if timer flag is set to 1. | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 2 |
| JTO addr | (PC ₇ -PC ₀) ← a ₇ -a ₀ if TO = 1 (PC) ← (PC) + 2 if TO = 0 | Jump to specified address if test 0 is a 1. | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 2 | 2 |
| JTI addr | (PC ₇ -PC ₀) ← a ₇ -a ₀ if T1 = 1 (PC) ← (PC) + 2 if T1 = 0 | Jump to specified address if test 1 is a 1. | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 2 |
| JZ addr | (PC ₇ -PC ₀) ← a ₇ -a ₀ if A = 0 (PC) ← (PC) + 2 if A = 1 | Jump to specified address if accumulator is 0. | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 2 |
| Control | | | | | | | | | | | | |
| EN I | | Enable the interrupt. | 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 |
| EN DMA | | Enables DMA handshake lines. | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| EN Flags | | Enables master interrupts. | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| SEL RB0 | (BS) ← 0 | Select bank 0 (locations 0-7) of data memory. | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| SEL RB1 | (BS) ← 1 | Select bank 1 (locations 24-31) of data memory. | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| HALT | | Initiates halt mode. | 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 | 1 | 0 | 1 | 1 |
| STOP H | | Sets CPU to software stop mode. (Port output high level) | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Data Moves | | | | | | | | | | | | |
| MOV A, # data | (A) ← data | Move immediate the specified data into the accumulator. | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 2 |
| MOV A, Rr | (A) ← (Rr); r = 0-7 | Move the contents of the designated registers into the accumulator. | 1 | 1 | 1 | 1 | 1 | r ₂ | r ₁ | r ₀ | 1 | 1 |
| MOV A, @Rr | (A) ← ((Rr)); r = 0-1 | Move indirect the contents of data memory location into the accumulator. | 1 | 1 | 1 | 1 | 0 | 0 | 0 | r ₀ | 1 | 1 |
| MOV A, PSW | (A) ← (PSW) | Move contents of the program status word into the accumulator. | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| MOV Rr, # data | (Rr) ← data; r = 0-7 | Move immediate the specified data into the designated register. | 1 | 0 | 1 | 1 | 1 | r ₂ | r ₁ | r ₀ | 2 | 2 |
| MOV Rr, A | (Rr) ← (A); r = 0-7 | Move accumulator contents into the designated register. | 1 | 0 | 1 | 0 | 1 | r ₂ | r ₁ | r ₀ | 1 | 1 |
| MOV @Rr, A | ((Rr)) ← (A); r = 0-1 | Move indirect accumulator contents into data memory location. | 1 | 0 | 1 | 0 | 0 | 0 | 0 | r ₀ | 1 | 1 |
| MOV @Rr, # data | ((Rr)) ← data; r = 0-1 | Move immediate the specified data into data memory. | 1 | 0 | 1 | 1 | 0 | 0 | 0 | r ₀ | 2 | 2 |
| MOV PSW, A | (PSW) ← (A) | Move contents of accumulator into the program status word. | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

Instruction Set (cont)

| Mnemonic | Operation | Description | Operation Code | | | | | | | | Cycles | Bytes |
|--------------------------|--|--|----------------|----|----|----|----|----------------|----------------|----------------|--------|-------|
| | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | |
| Data Moves (cont) | | | | | | | | | | | | |
| MOVPA, @A | A ← ((PC ₁₀ -PC ₈), (A)) | Move data in the current page into the accumulator. | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| MOVPA, @A | (A) ← ((011), (A)) | Move program data in page 3 into the accumulator. | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| XCH A, Rr | (A) ↔ (Rr); r = 0-7 | Exchange the accumulator and designated register's contents. | 0 | 0 | 1 | 0 | 1 | r ₂ | r ₁ | r ₀ | 1 | 1 |
| XCH A, @Rr | (A) ↔ (Rr); r = 0-1 | Exchange indirect contents of accumulator and location in data memory. | 0 | 0 | 1 | 0 | 0 | 0 | 0 | r ₀ | 1 | 1 |
| XCHD A, @Rr | (A ₃ -A ₀) ↔ (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 ₀ | 1 | 1 |
| Flags | | | | | | | | | | | | |
| CPL C | (C) ← NOT (C) | Complement contents of carry bit. | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| CPL FO | (FO) ← NOT (FO) | Complement contents of flag FO. | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| CPL F1 | (F1) ← NOT (F1) | Complement contents of flag F1. | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| CLRC | (C) ← 0 | Clear contents of carry bit to 0. | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| CLR FO | (FO) ← 0 | Clear contents of flag 0 to 0. | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| CLR F1 | (F1) ← 0 | Clear contents of flag 1 to 0. | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| Input / Output | | | | | | | | | | | | |
| ANL Pp, # data | (Pp) ← (Pp) AND data p = 1-2 | Logical AND immediate specified data with designated port (1 or 2). | 1 | 0 | 0 | 1 | 1 | 0 | p ₁ | p ₀ | 2 | 2 |
| ANLD Pp, A | (Pp) ← (Pp) 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 ₁ | p ₀ | 2 | 1 |
| IN A, DBB | (A) ← (DBBIN); IBF ← 0 | | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| IN A, Pp | (A) ← (Pp); p = 1-2 | Input data from designated port (1-2) into accumulator. | 0 | 0 | 0 | 0 | 1 | 0 | p ₁ | p ₂ | 2 | 1 |
| MOVD A, Pp | (A ₃ -A ₀) ← (Pp); (A ₇ -A ₄) ← 0 p = 4-7 | Move contents of designated port (4-7) into accumulator. | 0 | 0 | 0 | 0 | 1 | 1 | p ₁ | p ₀ | 2 | 1 |
| MOVD Pp, A | (Pp) ← (A ₃ -A ₀); p = 4-7 | Move contents of accumulator to designated port (4-7). | 0 | 0 | 1 | 1 | 1 | 1 | p ₁ | p ₀ | 2 | 1 |
| MOV STS, A | (ST ₇ -ST ₄) ← (A ₇ -A ₄) | Move contents of accumulator to designated port (4-7). | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

Instruction Set (cont)

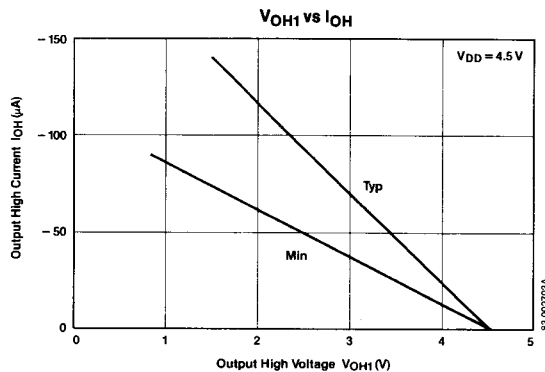
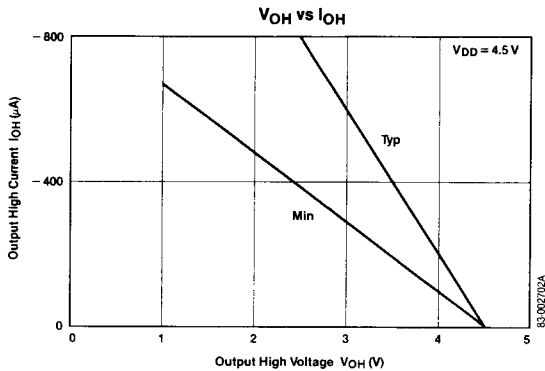
| Mnemonic | Operation | Description | Operation Code | | | | | | | | Cycles | Bytes |
|------------------------------|---|---|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|-------|
| | | | D ₇ | D ₆ | D ₅ | D ₄ | D ₃ | D ₂ | D ₁ | D ₀ | | |
| Input / Output (cont) | | | | | | | | | | | | |
| ORLD Pp, A | (Pp) ← (Pp) OR (A ₃ -A ₀); p = 4-7 | Logical OR contents of accumulator with designated port (4-7). | 1 | 0 | 0 | 0 | 1 | 1 | P ₁ | P ₀ | 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 | P ₁ | P ₀ | 2 | 2 |
| OUT DBB, A | (DBBOUT) ← (A), OBF ← 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| OUTL Pp, A | (Pp) ← (A); p = 1-2 | Output contents of accumulator to designated port (1-2). | 0 | 0 | 1 | 1 | 1 | 0 | P ₁ | P ₀ | 2 | 1 |
| Registers | | | | | | | | | | | | |
| DEC Rr | (Rr) ← (Rr) - 1; r = 0-7 | Decrement by 1 contents of designated register. | 1 | 1 | 0 | 0 | 1 | r ₂ | r ₁ | r ₀ | 1 | 1 |
| INC Rr | (Rr) ← (Rr) + 1; r = 0-7 | Increment by 1 contents of designated register. | 0 | 0 | 0 | 1 | 1 | r ₂ | r ₁ | r ₀ | 1 | 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 | r ₀ | 1 | 1 |
| Subroutine | | | | | | | | | | | | |
| CALL addr | ((SP)) ← (PC), (PSW ₇ -PSW ₄) (SP) ← (SP) + 1 (PC ₁₀ -PC ₀) ← a ₁₀ -a ₀ | Call designated subroutine. | a ₁₀ | a ₉ | a ₈ | 1 | 0 | 1 | 0 | 0 | 2 | 2 |
| RET | (SP) ← (SP) - 1 (PC) ← ((SP)) | Return from subroutine without restoring program status word. | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| RETR | (SP) ← (SP) - 1 (PC) ← ((SP)) (PSW ₇ -PSW ₄) ← ((SP)) | Return from subroutine restoring program status word. | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 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) | Move contents of timer / counter into accumulator. | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| MOV T, A | (T) ← (A) | Move contents of accumulator into timer / counter. | 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 |

Symbol Definitions

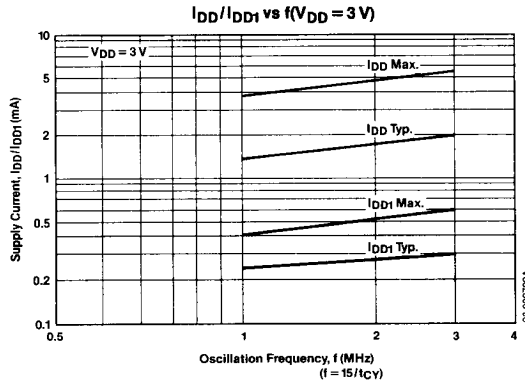
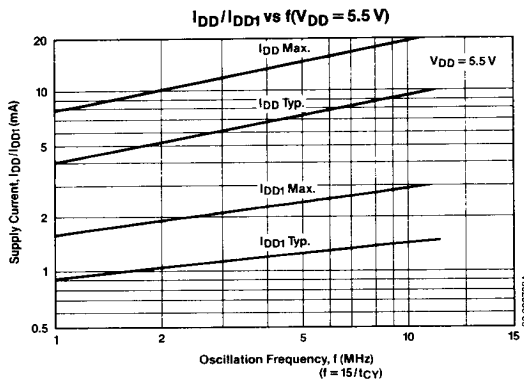
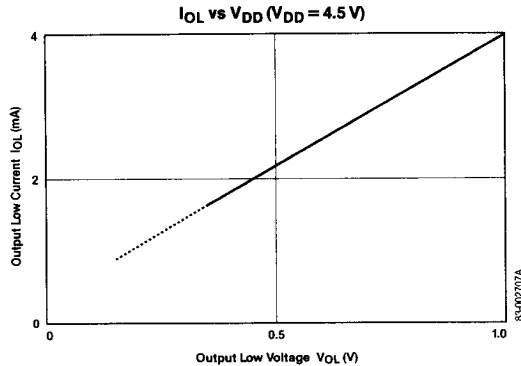
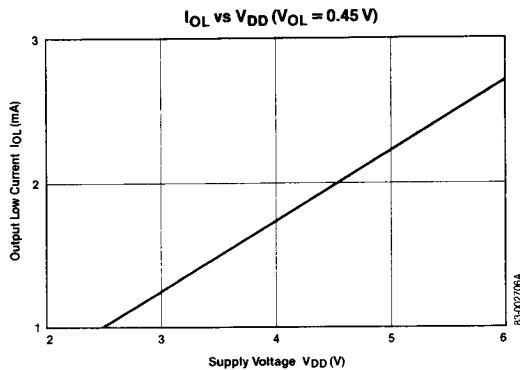
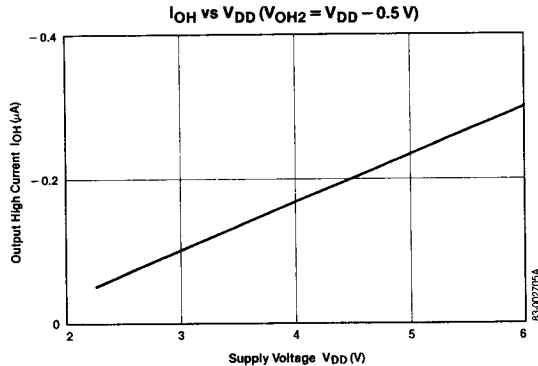
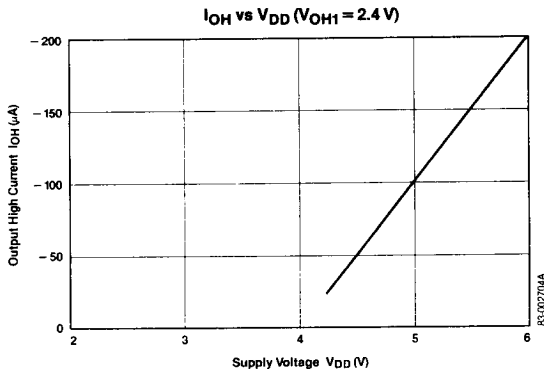
| Symbol | Description |
|--------|-------------------------------|
| A | Accumulator |
| AC | Auxiliary carry flag |
| addr | Program memory address |
| b | Accumulator bit (b = 0-7) |
| C | Carry flag |
| CNT | Counter |
| data | 8-bit data |
| DBB | Data bus buffer |
| F0, F1 | Flags 0, 1 (C / D flag) |
| 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)



4

Operating Characteristics (cont)

