SINGLE-CHIP 8-BIT CMOS MICROCONTROLLER

DESCRIPTION

The PC80CXX family of single-chip 8-bit CMOS microcontrollers consists of:

- The PCB80C49 with resident mask programmed 2 K x 8 ROM, 128 x 8 RAM.
- The PCB80C39 without resident program memory for use with external EPROM/ROM, 128 x 8 RAM.

All versions are pin and function compatible to their NMOS counter parts but with additional features and high performance.

The PC80CXX family are designed to be efficient control processors as well as arithmetic processors. Their instruction set allows the user to directly set and reset individual I/O, and to test individual individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.
An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories can be expanded using standard devices. Input/output capabilities can be expanded using standard devices.
The family has low power consumption and in addition a power down mode is provided.
For further detailed information see the 8048 family specification.

Features

- 8-bit CPU, ROM, RAM, I/O in a single 40-pin package
- PCB80C49: 2K x 8 ROM, 128 x 8 RAM
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions: 1 or 2 cycles
- Easily expandable memory and I/O
- TTL compatible inputs and outputs
- Single 5 V supply
- Wide frequency operating range
- Low current consumption
- Available with extended temperature ranges: (PCB version) 0 to + 70 °C
  (PCF version) −40 to + 85 °C
  (PCA version) −40 to + 110 °C
- Frequency range: 1 to 15 MHz for all temperature ranges

APPLICATIONS

- Peripheral interfaces and controllers
- Test and measurement instruments
- Sequencers
- Audio/video systems
- Environmental control systems
- Modems and data enciphering

PACKAGE OUTLINES

PCB/F/A80C39/C49P: 40-lead DIL; plastic (SOT129).
PCB/F/A80C39/C49WP: 44-lead PLCC; plastic leaded chip carrier (SOT187AA).

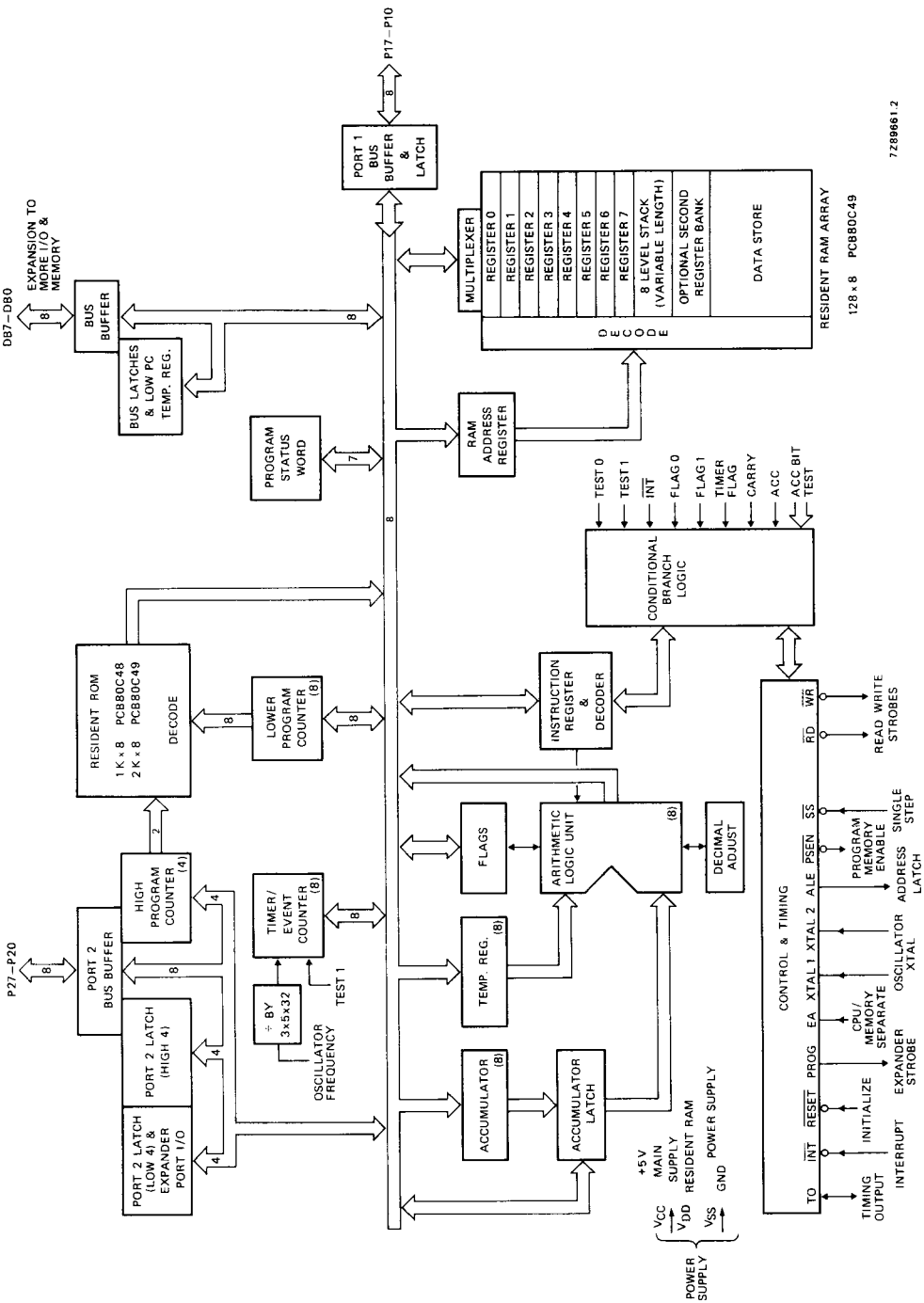


Fig. 1 Block diagram.

DEVELOPMENT DATA

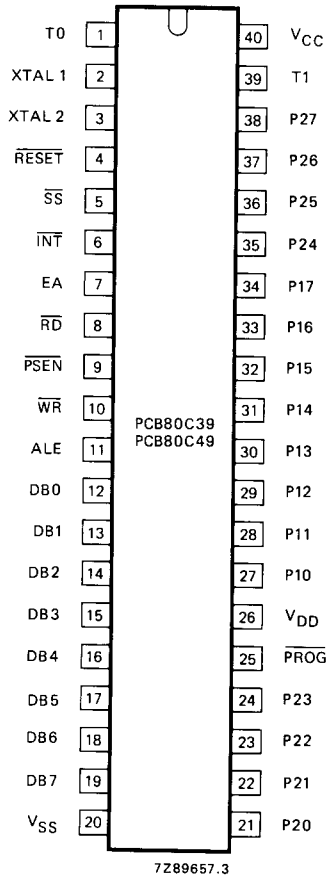


Fig. 2(a) Pinning diagram, DIL; for pin designation see next page.

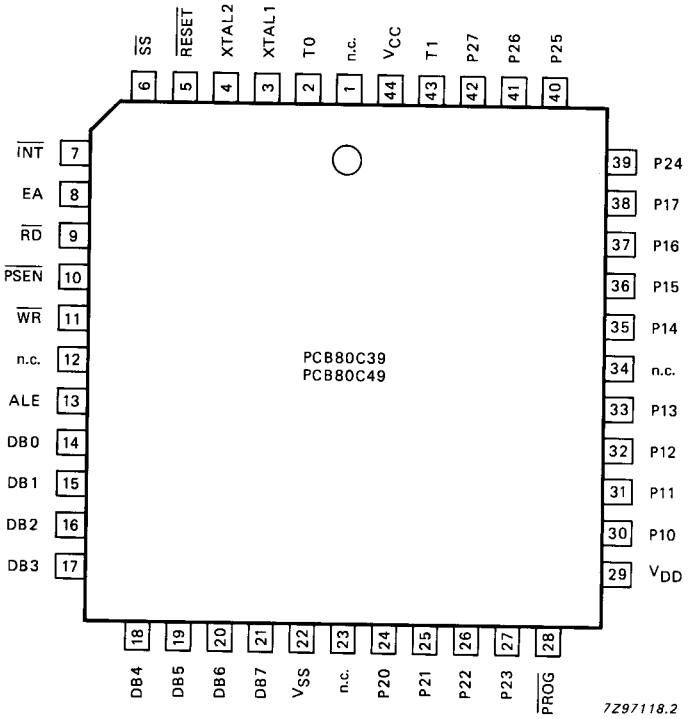


Fig. 2(b) Pinning diagram, PLCC.

Where: n.c. = not connected.

PIN DESIGNATION

designation	pin no.	function
DB0–DB7	12–19	**BUS.** Bidirectional I/O port that can be read or written using the $\overline{RD}$ and $\overline{WR}$ strobes. This port can also be statically latched. Contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$ and $\overline{WR}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
P10–P17	27–34	**Port 1.** 8-bit quasi-bidirectional I/O port (note 1).
P20–P27	21–24, 35–38	**Port 2.** 8-bit quasi-bidirectional I/O port (note 1). P20–P23 contain the 4 higher order address bits during an access of external program memory.
$\overline{PROG}$	25	**Output strobe** (active LOW) for I/O expander.
T0	1	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{INT}$	6	**Interrupt input pin.** When LOW causes an interrupt in the current program if external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after a RESET.
$\overline{RESET}$	4	**Reset input pin** used to initialize the microcontroller. Active LOW. During program verification the address is latched by a '0' to '1' transition on $\overline{RESET}$ and the data at the addressed location is output on BUS (note 2).
ALE	11	**Address latch enable.** Occurs once each machine cycle and is useful for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
$\overline{RD}$	8	**Read BUS.** Active LOW strobe used to gate data onto BUS lines when reading from an external source.
$\overline{WR}$	10	**Write BUS.** Active LOW strobe used to write data from BUS lines to an external designation.
EA	7	**External access input.** When HIGH, forces instruction fetch from external memory.
$\overline{PSEN}$	9	**Program store enable.** Active LOW strobe that occurs only during a fetch from external program memory.
$\overline{SS}$	5	**Single step input.** Active LOW which is used with ALE to cause the microcontroller to execute a single instruction.
V_{DD}	26	**RAM power supply,** + 5 V during normal operation and power-down mode.

designation	pin no.	function
XTAL1	2	**One side of crystal** (or inductor) input for internal oscillator. Can also be used as an input for an external timing source (note 2).
XTAL2	3	**Other side of crystal.** XTAL2 must be driven with the inverted signal of XTAL1 when an external timing source of $11 < f_{osc} \leqslant 15$ MHz is used.
V_{SS}	20	**Ground.**
V_{CC}	40	**Mains power supply,** +5 V during normal operation.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic 1 to the line. $\overline{\text{RESET}}$ sets all lines to logic 1.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

Program memory (see Fig. 3)

The resident program memory is:
2048 byte ROM
The PCB80C39 has no resident program memory.
The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-bytes banks MB0 and MB1.
The program memory is also divided into pages of 256 bytes for conditional branches.

There are three locations in program memory of special importance. These locations contain the first instruction to be executed after one of three events.
The three locations and their contents are:
location 0 — activation, then deactivation of the $\overline{\text{RESET}}$ line,
location 3 — activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled,
location 7 — an overflow of the timer/counter if the T/C interrupt is enabled.

Data memory (see Fig. 4)

The resident data memory is: 128 byte RAM.

All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1.
The first eight locations of RAM (0 to 7) are designated as working registers and are directly addressable by several instructions. By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0 (0 to 7).
RAM locations 8 to 23 are designated as the stack. Two locations (bytes) are used per CALL, allowing up to eight levels of subroutine nesting.
If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM is required, an I/O port can be used to select one (256 byte) bank of external memory at a time.

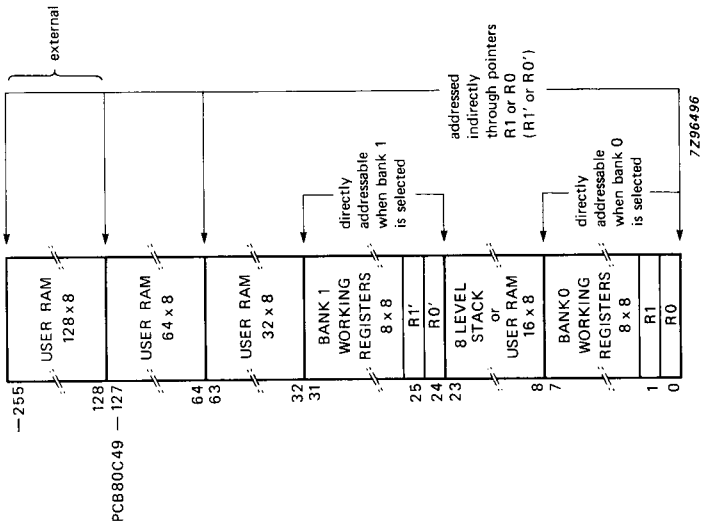


Fig. 4 Data memory map. In addition R0 or R1 (R0' or R1') may be used to address 256 words of an external RAM.

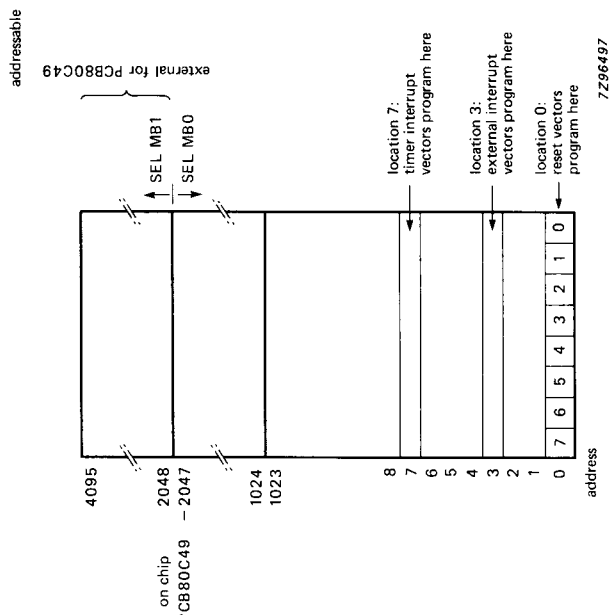


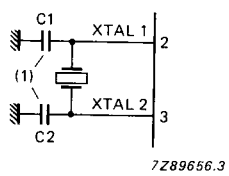
Fig. 3 Program memory map.

Program counter and stack

The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is '0', the PC addresses an internal program memory. At the boundary of the internal program memory an automatic switch over to external memory is made. When EA is '1', all the program is fetched from external ROM/EPROM. The total address space is 4K bytes. An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.
The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Oscillator and clock

The PCB80C49 contains its own internal oscillator and clock driver.
A crystal, inductor or external pulse generator determines the oscillator frequency (see Figs 5, 6 and 7). The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This CLK signal is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).



For quartz crystal
1 to 15 MHz: C1 = C2 = 15 to 25 pF
For ceramic resonators
1 to 15 MHz: C1 = C2 = 30^{+5}_{-10} pF

(1) Including crystal-socket stray capacitance.

Fig. 5 Crystal oscillator mode. Typical values are given. Crystal serial impedance should be $< 75\ \Omega$ at 6 MHz and $< 180\ \Omega$ at 3,6 MHz.

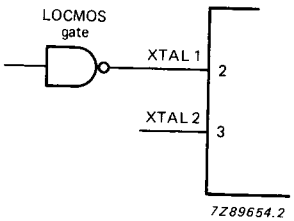
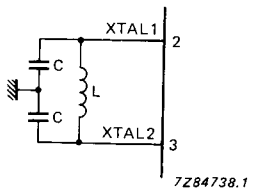


Fig. 6 Driving from an external source. Test conditions at XTAL1; Minimum HIGH (> 0,7 of V_{CC}) and LOW (< 0,13 of V_{CC}) times, should be at least 45% of a clock period. XTAL2 must be driven with the inverted signal of XTAL1 when an external timing source of $11 < f_{osc} \leqslant 15$ MHz is used.



L (μH)	C (pF)	f_{nom} (MHz)
45	20	5,2
120	20	3,2

Fig. 7 LC oscillator mode.

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by either an external input ($\overline{INT}$, pin 6) or the overflow of the internal timer/event counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine, After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The PCB80CXX family has 27 I/O lines. These lines are arranged as three 8-line ports, which serve individually as either inputs, outputs or together as bidirectional ports, plus 3 'test' inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, e.g., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The circuit configuration is shown in Fig. 8. Each line has a unique high-impedance pull-up transistor TR3, this is turned on when the line is pulled above 2 V by an external source or by writing a logic 1 to the port. This pull-up is sufficient to provide the source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the same pin to used for both input and output. When a logic 1 is written to a line, a second high impedance transistor TR2, pulls the line up to 5 V. To provide fast switching during a '0' to '1' transition, a relatively low-impedance transistor TR1 (approx. 750 Ω) is switched on for 1/5 of a machine cycle whenever a '1' is written to the line. Whenever a '0' is written to the line, a low-impedance (approx. 250 Ω) transistor TR4, overcomes the weak light pull-up and provides TTL current sinking capability.

Since the pull-down transistor TR4 is a low-impedance device, a '1' must first be written to any line which is to be used as an input. RESET initializes all lines to the high impedance '1' state. This structure allows input and output on the same pin and also allows a mixture of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is a true bidirectional 8-bit port with associated input and output strobes. The BUS port can operate in three different modes: as a latched I/O port, as a bidirectional bus port, or in an expanded system as a program memory address output port. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed. The BUS port lines are either active HIGH, active LOW, or high impedance (floating).

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$.

The latched mode (INS, OUTL) is intended for use in the single-chip configuration, where BUS is not being used as an expanded port. OUTL and MOVX instructions can be mixed if required. However, when using a MOVX instruction a previously latched output will be lost and BUS will be left in the high impedance state. INS does not put the BUS in a high impedance state. Therefore, in order to read an external byte (and not the previously latched value) using the INS instruction, it is necessary to follow an OUTL with a MOVX instruction to place BUS in a high impedance state.

OUTL should never be used in a system with external program memory, since latching BUS may cause the next instruction to be incorrectly fetched.

Test (T0, T1) and $\overline{INT}$ inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These pins are T0, T1, and $\overline{INT}$ and they allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and $\overline{INT}$ pins have other possible functions as well.

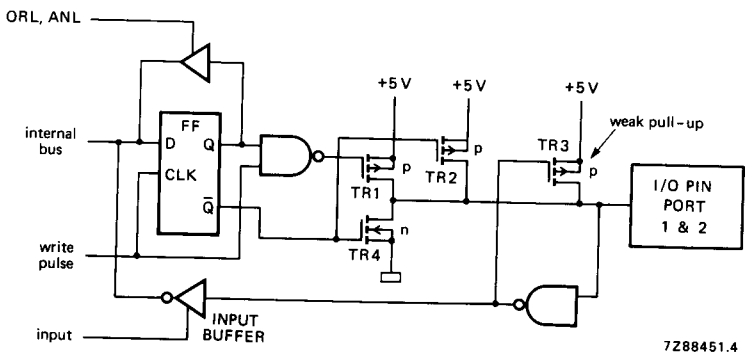


Fig. 8 Quasi-bidirectional port structure.

FUNCTIONAL DESCRIPTION (continued)

$\overline{\text{RESET}}$ input (see Fig. 10)

The $\overline{\text{RESET}}$ input provides a means to initialize the processor. This Schmitt-trigger input has an internal pull-up resistor. The combination of an external 47 kΩ resistor and a 1 μF capacitor provides a reset pulse of sufficient duration to guarantee that all circuitry is reset. If the reset pulse is generated otherwise, the $\overline{\text{RESET}}$ pin must be held at ground for at least 10 ms after the (0,13 V_{CC}) power supply is within tolerance. Only five machine cycles (2,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized.

Single step input ($\overline{\text{SS}}$)

Under control of the $\overline{\text{SS}}$ line, the processor can be forced to execute one instruction and then to wait until the single step switch is activated again.

IDLE mode

The PCB80CXX family is provided with a IDLE mode in which the internal oscillator, the internal timer and the external interrupt and counter are still functioning, while the status of following parts is maintained: RAM and register/Port 1 and 2/Bus. The IDLE mode is entered after execution of the IDLE instruction (opcode 1H). The IDLE mode is terminated by one of the two possible interrupts, if enabled, or a RESET signal. If an external interrupt terminates the IDLE mode, the next instruction that is executed is at location 3 of the program store. If a timer/counter interrupt terminates the IDLE mode, the next instruction is at location 7. The reset signal will terminate the IDLE mode, and also initialize the processor.

Power-down mode

In the PCB80CXX family, power can be removed from all but the 64 x 8 bit and 128 x 8 bit data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM are maintained. V_{CC} serves as the + 5 V supply pin for most of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In standby, V_{CC} is at ground and V_{DD} is maintained at + 5 V.

Applying $\overline{\text{RESET}}$ to the processor through the $\overline{\text{RESET}}$ pin inhibits any access to the RAM by the processor and guarantees that the RAM cannot be inadvertently altered when power is removed from V_{CC}. Fig. 9 shows a typical power-down sequence.

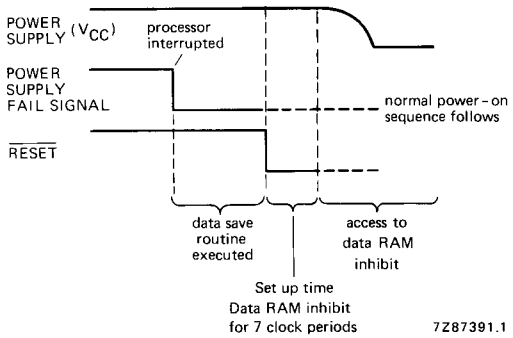


Fig. 9 Power-down sequence.

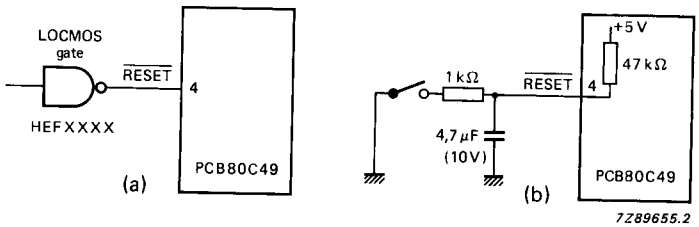


Fig. 10(a) External reset circuit; (b) power-on reset.

DEVELOPMENT DATA

Instruction set

The PCB80CXX instruction set consists of over 90 one and two-byte instructions (see Table 2). Program code efficiency is high because:

- working registers and program variables are stored in the RAM, which require only a single byte to address,
- program memory is divided into pages of 256 bytes, which means that branch destination addresses require one byte.

In addition to performing logical and arithmetic operations, the instruction set manipulates and tests both bits and bytes. A set of MOVE instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) upon the content of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' DJNZ instruction saves a byte every time it is used as opposed to using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The PCB80CXX can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are required for real-time applications. Instruction timing is shown in Table 3.

Note: The OUTL, ANL and ORL instructions relating to BUS, are used with internal program memory only.

Table 1

Symbol definitions used in Table 2.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
$\overline{\text{I}}$	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Notes to Table 2.

1. Instruction 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 content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

Table 2 Instruction set

	mnemonic	function	description	instruction code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	cycles	bytes	flags C	AC	F0	F1	BS
ACCUMULATOR	ADD A,Rr	(A)←(A) + (Rr) for r = 0–7	Add contents of designated register to A	0	1	1	0	1	r	r	r	1	1	•	•			
	ADD A,@Rr	(A)←(A) + ((Rr)) for r = 0–1	Add indirect the contents of the data memory location to A	0	1	1	0	0	0	0	r	1	1	•	•			
	ADD A,#data	(A)←(A) + data	Add immediate data to A	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•	•			
	ADDC A,Rr	(A)←(A) + (C) + (Rr) for r = 0–7	Add with carry the contents of designated register to A	0	1	1	1	1	r	r	r	1	1	•	•			
	ADDC A,@Rr	(A)←(A) + (C) + ((Rr)) for r = 0–1	Add indirect with carry the contents of the data memory location to A	0	1	1	1	0	0	0	r	1	1	•	•			
	ADDC A,#data	(A)←(A) + (C) + data	Add immediate data with carry to A	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•	•			
	ANL A,Rr	(A)←(A) AND (Rr) for r = 0–7	Logical AND contents of designated register with A	0	1	0	1	1	r	r	r	1	1					
	ANL A,@Rr	(A)←(A) AND ((Rr)) for r = 0–1	Logical AND indirect the contents of data memory with A	0	1	0	1	0	0	0	r	1	1					
	ANL A,#data	(A)←(A) AND data	Logical AND immediate data with A	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
	CLR A	(A)←0	Clear the contents of A	0	0	1	0	0	1	1	1	1	1					
	CPL A	(A)←NOT(A)	Complement the contents of A	0	0	1	1	0	1	1	1	1	1					
	DA A		Decimal adjust the contents of A	0	1	0	1	0	1	1	1	1	1	•	•			
	DEC A	(A)←(A) −1	Decrement the contents of A by 1	0	0	0	0	0	1	1	1	1	1					
	INC A	(A)←(A) + 1	Increment the contents of A by 1	0	0	0	1	0	1	1	1	1	1					
	ORL A,Rr	(A)←(A) OR (Rr) for r = 0–7	Logical OR contents of designated register with A	0	1	0	0	1	r	r	r	1	1					
	ORL A,@Rr	(A)←(A) OR ((Rr)) for r = 0–1	Logical OR indirect the contents of data memory location with A	0	1	0	0	0	0	0	r	1	1					

ACCUMULATOR (continued)	ORL A,#data	(A)←(A) OR data	Logical OR immediate data with A	0	1	0	0	0	0	1	1	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							
	RL A	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$ for n = 0–6	Rotate A left by 1-bit without carry	1	1	1	0	0	1	1	1	1	1					
	RLC A	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(C)$ $(C)\leftarrow(A_7)$ for n = 0–6	Rotate A left by 1-bit through carry	1	1	1	1	0	1	1	1	1	1	•				
	RR A	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$ for n = 0–6	Rotate A right by 1-bit without carry	0	1	1	1	0	1	1	1	1	1					
	RRC A	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$ $(C)\leftarrow(A_0)$ n = 0–6	Rotate A right by 1-bit through carry	0	1	1	0	0	1	1	1	1	1	•				
	SWAP A	$(A_{4-7})\leftrightarrow(A_{0-3})$	Swap the two 4-bit nibbles in A	0	1	0	0	0	1	1	1	1	1					
	XRL A,Rr	(A)←(A) XOR (Rr)	Logical XOR contents of designated register with A	1	1	0	1	1	r	r	r	1	1					
	XRL A,@Rr	(A)←(A) XOR ((Rr))	Logical XOR indirect the contents of data memory location with A	1	1	0	1	0	0	0	r	1	1					
	XRL A,#data	(A)←(A) XOR data	Logical XOR immediate data with A	1	1	0	1	0	0	1	1	2	2					
				d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0							

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
BRANCH	DJNZ Rr, addr	(Rr)←(Rr) −1 if (Rr) not zero: (PC_{0-7})←addr	Decrement the specified register and test contents	1 1 1 0 1 r r r a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JBb addr	(PC_{0-7})←addr if Bb = 1; (PC)←(PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set	b_2 b_1 b_0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JC addr	(PC_{0-7})←addr if C = 1; (PC)←(PC) + 2 if C = 0	Jump to specified address if carry flag is set	1 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JF0 addr	(PC_{0-7})←addr if (F0 = 1; (PC)←(PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set	1 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JF1 addr	(PC_{0-7})←addr if F1 = 1; (PC)←(PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set	0 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JMP addr	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11})←(DBF)	Direct jump to specified address within the 2K address block	a_{10} a_9 a_8 0 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JMPP @A	(PC_{0-7})←((A))	Jump indirect to specified address within address page	1 0 1 1 0 0 1 1	2	1					
	JNC addr	(PC_{0-7})←addr if C = 0; (PC)←(PC) + 2 if C = 1	Jump to specified address if carry flag is LOW	1 1 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNI	(PC_{0-7})←addr if $\overline{INT}$ = 0; (PC)←(PC) + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is LOW	1 0 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNT0 addr	(PC_{0-7})←addr if T0 = 0; (PC)←(PC) + 2 if T0 = 1	Jump to specified address if T0 is LOW	0 0 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNT1 addr	(PC_{0-7})←addr if T1 = 0; (PC)←(PC) + 2 if T1 = 1	Jump to specified address if T1 is LOW	0 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					

BRANCH (continued)	JNZ addr	$(PC_{0-7}) \leftarrow$addr if A $\neq$ 0; (PC)$\leftarrow$(PC) + 2 if A = 0	Jump to specified address if A is non-zero	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JTF addr	$(PC_{0-7}) \leftarrow$addr if TF = 1; (PC)$\leftarrow$(PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT0 addr	$(PC_{0-7}) \leftarrow$addr if T0 = 1; (PC)$\leftarrow$(PC) + 2 if T0 = 0	Jump to specified address if T0 = 1	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT1 addr	$(PC_{0-7}) \leftarrow$addr if T1 = 1; (PC)$\leftarrow$(PC) + 2 if T1 = 0	Jump to specified address if T1 = 1	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JZ addr	$(PC_{0-7}) \leftarrow$addr if A = 0; (PC)$\leftarrow$(PC) + 2 if A $\neq$ 0	Jump to specified address if A is zero	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
CONTROL	EN I		Enable external ($\overline{INT}$) interrupt	0	0	0	0	0	1	0	1	1	1					
	DIS I		Disable external ($\overline{INT}$) interrupt	0	0	0	1	0	1	0	1	1	1					
	SEL RB0	(BS)$\leftarrow$0	Select bank 0 (locations 0–7) of data memory	1	1	0	0	0	1	0	1	1	1					•
	SEL RB1	(BS)$\leftarrow$1	Select bank 1 (locations 24–31) of data memory	1	1	0	1	0	1	0	1	1	1					•
	SEL MB0	(DBF)$\leftarrow$0	Select program memory bank 0; addresses 0–2047	1	1	1	0	0	1	0	1	1	1					
	SEL MB1	(DBF)$\leftarrow$1	Select program memory bank 1; addresses 2048–4095	1	1	1	1	0	1	0	1	1	1					
	ENT0 CLK		Enable clock output onto T0	0	1	1	1	0	1	0	1	1	1					

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
DATA MOVES	MOV A,#data	(A)←data	Move immediate data into A	0 0 1 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV A,Rr	(A)←(Rr) for r = 0–7	Move the contents of the designated register into A	1 1 1 1 1 r r r	1	1					
	MOV A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of data memory into A	1 1 1 1 0 0 0 r	1	1					
	MOV A,PSW	(A)←(PSW)	Move contents of the program status word into A	1 1 0 0 0 1 1 1	1	1					
	MOV Rr,#data	(Rr)←data for r = 0–7	Move immediate data into the designated register	1 0 1 1 1 r r r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV Rr,A	(Rr)←(A) for r = 0–7	Move A contents into the designated register	1 0 1 0 1 r r r	1	1					
	MOV @Rr,A	((Rr))←(A) for r = 0–1	Move indirect A contents into data memory location	1 0 1 0 0 0 0 r	1	1					
	MOV @Rr,#data	((Rr))←data for r = 0–1	Move indirect the specified data into data memory	1 0 1 1 0 0 0 r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV PSW,A	(PSW)←A	Move contents of A into the program status word	1 1 0 1 0 1 1 1	1	1	•	•	•		•
	MOVP A,@A	(A)←((A))	Move data in the current page into A	1 0 1 0 0 0 1 1	2	1					
	MOVP3 A,@A	(A)←((A)) in page 3	Move data in page 3 of memory bank Ø into A	1 1 1 0 0 0 1 1	2	1					
	MOVX A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of external memory location into A	1 0 0 0 0 0 0 r	2	1					
	MOVX @Rr,A	((Rr))←(A) for r = 0–1	Move indirect the contents of A into external memory	1 0 0 1 0 0 0 r	2	1					
	XCH A,Rr	(A)↔(Rr) for r = 0–7	Exchange A with designated register contents	0 0 1 0 1 r r r	1	1					
	XCH A,@Rr	(A)↔((R)) for r = 0–1	Exchange indirect A contents with location in data memory	0 0 1 0 0 0 0 r	1	1					

D.M.	XCHD A,@Rr	$(A_{0-3}) \leftrightarrow (Rr_{0-3})$ for r = 0–1	Exchange indirect 4-bit contents of A with data memory	0	0	1	1	0	0	0	r	1	1					
FLAGS	CPL C	(C)←NOT (C)	Complement content of carry bit	1	0	1	0	0	1	1	1	1	1	•				
	CPL F0	(F0)←NOT (F0)	Complement content of flag F0	1	0	0	1	0	1	0	1	1	1			•		
	CPL F1	(F1)←NOT (F1)	Complement content of flag F1	1	0	1	1	0	1	0	1	1	1				•	
	CLR C	(C)←0	Clear content of carry bit to 0	1	0	0	1	0	1	1	1	1	1	•				
	CLR F0	(F0)←0	Clear content of flag F0 to 0	1	0	0	0	0	1	0	1	1	1			•		
	CLR F1	(F1)←0	Clear content of flag F1 to 0	1	0	1	0	0	1	0	1	1	1				•	
INPUT/OUTPUT	ANL BUS,#data	(BUS)←(BUS) AND data	Logical AND immediate data with BUS	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ANL Pp,#data	(Pp)←(Pp) AND data; p = 1–2	Logical AND immediate data with designated port (1 or 2)	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	ANLD Pp,A	(Pp)←(Pp) AND (A_{0-3}); p = 4–7	Logical AND contents of A with designated port (4–7)	1	0	0	1	1	1	p	p	2	1					
	IN A,Pp	(A)←(Pp) p = 1–2	Input data from designated port (1–2) into A	0	0	0	0	1	0	p	p	2	1					
	INS A,BUS	(A)←(BUS)	Input strobed BUS data into A	0	0	0	0	1	0	0	0	2	1					
	MOVD A,Pp	(A_{0-3})←(Pp); p = 4–7 (A_{4-7})←0	Move contents of designated port (4–7) into A	0	0	0	0	1	1	p	p	2	1					
	MOVD Pp,A	(Pp)←(A_{0-3}) p = 4–7	Move contents of A to designated port (4–7)	0	0	1	1	1	1	p	p	1	1					
	ORLD Pp,A	(Pp)←(Pp) OR (A_{0-3}); p = 4–7	Logical OR contents of A with designated port (4–7)	1	0	0	0	1	1	p	p	1	1					
	ORL BUS,#data	(BUS)←(BUS) OR data	Logical OR immediate data with BUS	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ORL Pp,#data	(Pp)←(Pp) OR data p = 1–2	Logical OR immediate data with designated port (1–2)	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	OUTL BUS,A	(BUS)←(A)	Output contents A onto BUS	0	0	0	0	0	0	1	0	2	1					
	OUTL Pp,A	(Pp)←(A) p = 1–2	Output contents A to designated port (1–2)	0	0	1	1	1	0	p	p	2	1					

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
REGISTER	DEC Rr	(Rr)←(Rr) −1 for r = 0–7	Decrement contents of designated register by 1	1 1 0 0 1 r r r	1	1					
	INC Rr	(Rr)←(Rr) + 1 for r = 0–7	Increment contents of designated register by 1	0 0 0 1 1 r r r	1	1					
	INC @Rr	((R))←((Rr)) + 1 for r = 0–1	Increment indirect the contents of data memory location by 1	0 0 0 1 0 0 0 r	1	1					
SUBROUTINE	CALL addr	((SP))←(PC), (PSW_{4-7}) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$	Call designated subroutine	a_{10} a_9 a_8 1 0 1 0 0 / a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_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_{4-7})←((SP))	Return from subroutine restoring program status word	1 0 0 1 0 0 1 1	2	1					
TIMER/COUNTER	EN TCNTI		Enable timer/counter interrupt	0 0 1 0 0 1 0 1	1	1					
	DIS TCNTI		Disable timer/counter interrupt	0 0 1 1 0 1 0 1	1	1					
	MOV A,T	(A)←(T)	Move contents of timer/counter into A	0 1 0 0 0 0 1 0	1	1					
	MOV T,A	(T)←(A)	Move contents of A into timer/counter	0 1 1 0 0 0 1 0	1	1					
	STOP TCNT		Stop count for event counter or timer	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					
	IDLE		Entering IDLE mode	0 0 0 0 0 0 0 1	1	1					
	NOP		No operation	0 0 0 0 0 0 0 0	1	1					

Table 3 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	* –	–	–
OUTL P,A			–		output to port	–	–	* –	–	–
ANL P,#data		*	–		read port	fetch immediate data	–	*increment program counter	output to port	–
ORL P,#data		*	–		read port	fetch immediate data	–	*increment program counter	output to port	–
INS A,BUS			–		–	–	read port	* –	–	–
OUTL BUS,A			–		output to port	–	–	* –	–	–
ANL BUS,#data		*	–		read port	fetch immediate data	–	*increment program counter	output to port	–
ORL BUS,#data		*	–		read port	fetch immediate data	–	*increment program counter	output to port	–
MOVX @R,A			output RAM address		output data to RAM	–	–	* –	–	–
MOVX A,@R			output RAM address		–	–	read data	* –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	* –	–	–

Table 3 Instruction timing (continued)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	* –	–	–
ANLD P,A	| | |		output opcode/ address	| | |	output data	–	–	* –	–	–
ORLD P,A	| | |		output opcode/ address	| | |	output data	–	–	* –	–	–
J (conditional)	| | |	*	sample condition	| increment timer	–	fetch immediate data	–	*update program counter	–	–
STRT CNT STRT T	| | |	*	–	–	start counter					
STOP TCNT	| | |	*	–	–	stop counter					
EN I	| | |	*	–	enable interrupt	–					
DIS I	| | |	*	–	disable interrupt	–					
ENT0 CLK	fetch instruction	*increment program counter	–	enable clock	–					

* Valid instruction addresses are output at this time if external program memory is being accessed.

S5	S1	S2	S3	S4	S5	S1
	INPUT INSTR.	DECODE	EXECUTION			INPUT
OUTPUT ADDRESS		INC. PC	OUTPUT ADDRESS			

7Z84731

Fig. 11 Instruction cycle.

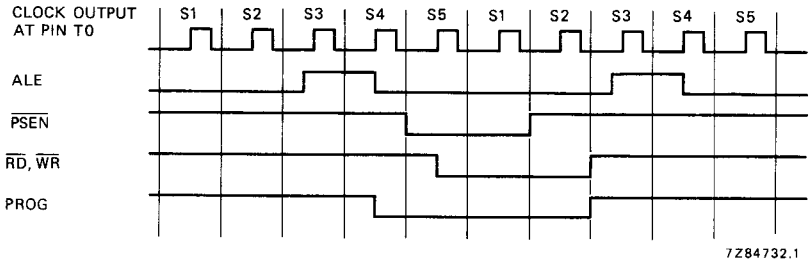


Fig. 12 Instruction cycle timing.

Table 4 Instruction map.

first hex ↓ / second hex →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE mode	OUTL BUS,A	ADD A, #data	JMP page 0	EN I		DEC A	INS A,BUS	IN A, Pp			MOVD A,Pp			
1	INC @Rr		JB0 addr	ADDC A, #data	CALL page 0	DIS I	JTF addr	INC A	INC Rr							
2	XCH A, @Rr			MOV A, #data	JMP page 1	EN TCNTI	JNT0 addr	CLR A	XCH A,Rr							
3	XCHD A, @Rr		JB1 addr		CALL page 1	DIS TCNTI	JT0 addr	CPL A		OUTL Pp,A			MOVD Pp,A			
4	ORL A, @Rr		MOV A, T	ORL A, #data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr							
5	ANL A, @Rr		JB2 addr	ANL A, #data	CALL page 2	STRT T	JT1 addr	DA, A	ANL A,Rr							
6	ADD A, @Rr		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr							
7	ADDC A, @Rr		JB3 addr		CALL page 3	ENT0 CLK	JF1 addr	RR A	ADDC A,Rr							
8	MOVX A, @Rr			RET	JMP page 4	CLR F0	JN1 addr		ORL BUS, #data	ORL Pp, #data			ORLD Pp,A			
9	MOVX @Rr,A		JB4 addr	RETR	CALL page 4	CPL F0	JNZ addr	CLR C	ANL BUS, #data	ANP Pp, #data			ANLD Pp,A			
A	MOV @Rr, A			MOVP A, @A	JMP page 5	CLR F1		CPL C	MOV Rr,A							
B	MOV @Rr, #data		JB5 addr	JMPP @A	CALL page 5	CPL F1	JF0 addr		MOV R, #data							
C					JMP page 6	SEL RB0	JZ addr	MOV A, PSW	DEC Rr							
D	XRL A, @Rr		JB6 addr	XRL A, #data	CALL page 6	SEL RB1		MOV PSW, A	XRL A,Rr							
E				MOVP3 A @A	JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr							
F	MOV A, @Rr		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr							

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

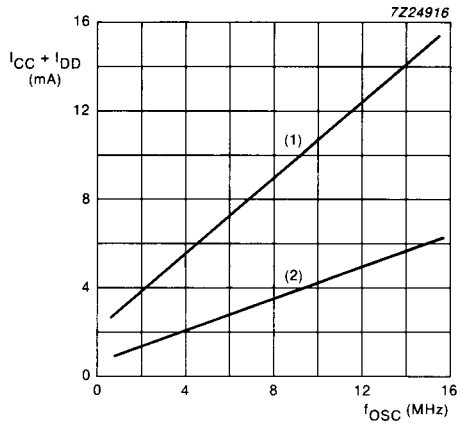
parameter	symbol	min.	max.	unit
Input, output current on any single pin	I_I, I_O	–	± 10	mA
Total power dissipation	P_{tot}	–	0.5	W
Storage temperature range	T_{stg}	–65	+ 150	°C
Operating ambient temperature range				
PCB80C39/49 version	T_{amb}	0	+ 70	°C
PCF80C39/49 version	T_{amb}	–40	+ 85	°C
PCA80C39/49 version	T_{amb}	–40	+ 110	°C

DC CHARACTERISTICS

$V_{CC} = V_{DD} = 5$ V (± 10%); $V_{SS} = 0$ V; $T_{amb} = 0$ to + 70 °C (PCB80C39/49); –40 to + 85 °C (PCF80C39/49); –40 to + 110 °C (PCA80C39/49). All voltages with respect to V_{SS} unless otherwise specified.

DEVELOPMENT DATA

parameter	conditions	symbol	min.	max.	unit
Supply voltage	$V_{CC} = V_{DD}$ (note 2)	V_{CC}	4.5	5.5	V
Supply current					
operating	f_{CLK} = 15 MHz	$I_{CC} + I_{DD}$	–	15	mA
IDLE mode	f_{CLK} = 15 MHz	I_{IDLE}	–	6	mA
power down mode	V_{DD} = 2 V; $\overline{RESET}$ = LOW	I_{PD}	–	2	µA
Inputs					
Input voltage LOW all inputs except $\overline{RESET}$; XTAL1; XTAL2		V_{IL}	–0.5	0.18 V_{CC}	V



(1) Operational mode.
(2) Idle mode.

Fig.13 Typical values of maximum supply current ($I_{CC} + I_{DD}$) as a function of the oscillator frequency (f_{OSC}) at $V_{CC(max.)}$ = 5.5 V; T_{amb} = 0 to + 70 °C.

DC CHARACTERISTICS (continued)

parameter	conditions	symbol	min.	max.	unit
Input voltage LOW $\overline{RESET}$; XTAL1; XTAL2		V_{IL1}	−0.5	0.13 V_{CC}	V
Input voltage HIGH all inputs except $\overline{RESET}$; XTAL1; XTAL2	(PCB80C39/49 version)	V_{IH}	0.4 V_{DD}	V_{CC}	V
	(PCF/PCA80C39/49 version note 1)	V_{IH}	0.43 V_{DD}	V_{CC}	V
Input voltage HIGH $\overline{RESET}$; XTAL1; XTAL2		V_{IH1}	0.7 V_{DD}	V_{CC}	V
Outputs					
Output voltage LOW BUS	I_{OL} = 2 mA	V_{OL}	–	0.45	V
Output voltage LOW $\overline{RD}$; $\overline{WR}$; $\overline{PSEN}$; ALE	I_{OL} = 1.8 mA	V_{OL1}	–	0.45	V
Output voltage LOW $\overline{PROG}$	I_{OL} = 1 mA	V_{OL2}	–	0.45	V
Output voltage LOW all other outputs	I_{OL} = 1.6 mA	V_{OL3}	–	0.45	V
Output voltage HIGH BUS	$-I_{OH}$ = 400 μA	V_{OH}	0.75 V_{CC}	–	V
Output voltage HIGH $\overline{RD}$; $\overline{WR}$; $\overline{PSEN}$; ALE	$-I_{OH}$ = 100 μA	V_{OH1}	0.75 V_{CC}	–	V
Output voltage HIGH all other outputs	$-I_{OH}$ = 40 μA	V_{OH2}	0.75 V_{CC}	–	V
Input leakage current $\overline{INT}$; T1; EA	without internal pull-up $V_{SS} < V_I < V_{CC}$	$\pm I_{IL}$	–	10	μA
Input leakage current P10-P17; P20-P27; $\overline{SS}$	with internal pull-up (PCB80C39/49 version) $V_{SS} + 0.45 < V_I < V_{CC}$	$-I_{IL1}$	–	500	μA
	PCF/PCA80C39/49 versions $V_{SS} + 0.45 < V_I < V_{CC}$	$-I_{IL1}$	–	600	μA

parameter	conditions	symbol	min.	max.	unit
Input leakage current $\overline{RESET}$	(PCB80C39/49 version) $V_{SS} < V_I < V_{IH1}$	$-I_{ILR}$	20	300	μA
	(PCF/PCA80C39/49 version) $V_{SS} < V_I < V_{IH1}$	$-I_{ILR}$	20	350	μA
Output leakage current BUS; TO at high impedance state	$V_{SS}+0.45 < V_I < V_{CC}$	$\pm I_{OL}$	–	10	μA

Notes to the DC characteristics

1. Levels are not fully compatible according to the TTL specification.
2. Note here that V_{DD} and V_{CC} refer to pins 26 and 40. These pins are the RAM and processor power supplies respectively.

DEVELOPMENT DATA

AC CHARACTERISTICS

$V_{CC} = V_{DD} = 5\ V \pm 10\%$; $V_{SS} = 0\ V$; T_{amb} (PCB version) = 0 to + 70 °C; note 1;
T_{amb} (PCF version) = −40 to + 85 °C; note 1; T_{amb} (PCA version) = −40 to + 110 °C; note 1).

See waveforms Figs 14, 15, 16, 17 and 18.

parameter	f (t_{CL}) (note 2)	symbol	11 MHz		unit
			min.	max.	
ALE pulse width	$7/30t_{CL}-170$	t_{LL}	150	–	ns
Address set-up time to ALE	$2/15t_{CL}-110$	t_{AL}	70	–	ns
Address hold time from ALE	$1/15t_{CL}-40$	t_{LA}	50	–	ns
Control pulse width $\overline{RD}$, $\overline{WR}$	$1/2t_{CL}-200$	t_{CC1}	480	–	ns
Control pulse width $\overline{PSEN}$	$2/5t_{CL}-200$	t_{CC2}	350	–	ns
Data set-up time before $\overline{WR}$	$13/30t_{CL}-200$	t_{DW}	390	–	ns
Data hold time after $\overline{WR}$ (note 3)	$1/15t_{CL}-50$	t_{WD}	40	–	ns
Data hold time $\overline{RD}$, $\overline{PSEN}$	$1/10t_{CL}-30$	t_{DR}	0	110	ns
$\overline{RD}$ to data input	$2/5t_{CL}-170$	t_{RD1}	–	375	ns
$\overline{PSEN}$ to data input	$3/10t_{CL}-170$	t_{RD2}	–	240	ns
Address set-up time to $\overline{WR}$	$1/3t_{CL}-150$	t_{AW}	300	–	ns
Address set-up time to data input ($\overline{RD}$)	$7/10t_{CL}-250$	t_{AD1}	–	730	ns
Address set-up time to data input ($\overline{PSEN}$)	$1/2t_{CL}-220$	t_{AD2}	–	460	ns
Address floating to $\overline{RD}$, $\overline{WR}$	$2/15t_{CL}-40$	t_{AFC1}	140	–	ns
Address floating to $\overline{PSEN}$	$1/30t_{CL}-40$	t_{AFC2}	10	–	ns
ALE to control pulse $\overline{RD}$, $\overline{WR}$	$1/5t_{CL}-75$	t_{LAFC1}	200	–	ns
ALE to control pulse $\overline{PSEN}$	$1/10t_{CL}-75$	t_{LAFC2}	60	–	ns
Control pulse to ALE $\overline{RD}$, $\overline{WR}$, $\overline{PROG}$	$1/15t_{CL}-40$	t_{CA1}	50	–	ns
Control pulse to ALE $\overline{PSEN}$	$4/15t_{CL}-40$	t_{CA2}	320	–	ns
Port control set-up to $\overline{PROG}$	$1/10t_{CL}-80$	t_{CP}	50	–	ns

AC CHARACTERISTICS (continued)

parameter	f (t_{CL})	symbol	11 MHz		unit
			min.	max.	
Port control hold to $\overline{PROG}$	$4/15t_{CL}-260$	t_{PC}	100	–	ns
$\overline{PROG}$ to time port 2 input must be valid	$17/30t_{CL}-120$	t_{PR}	–	650	ns
Input data hold time from $\overline{PROG}$	$1/10t_{CL}$	t_{PF}	0	140	ns
Output data set-up time	$2/5t_{CL}-290$	t_{DP}	250	–	ns
Output data hold time	$1/10t_{CL}-90$	t_{PD}	40	–	ns
$\overline{PROG}$ pulse width	$7/10t_{CL}-250$	t_{PP}	700	–	ns
Port 2 I/O data set-up time to ALE	$4/15t_{CL}-200$	t_{PL}	160	–	ns
Port 2 I/O data hold time to ALE	$1/10t_{CL}-120$	t_{LP}	15	–	ns
Port output from ALE	$3/10t_{CL}+100$	t_{PV}	–	510	ns
Cycle time	$(1/f_{XTAL}) \times 15$	t_{CL}	1,36	15	μs
T0 repetition rate	$3/15t_{CL}$	t_{0PRR}	270	–	ns
Clock period (note 2)	$1/(f_{XTAL})$	t_{CY}	90,9	1000	ns

Notes to AC characteristics

1. Control outputs: C_L = 80 pF
   Bus outputs: C_L = 150 pF.
2. f(t_{CL}) assumes 50% duty cycle on XTAL1 and XTAL2; minimum frequency = 1 MHz; maximum frequency = 15 MHz for all versions.
3. Bus high-impedance load: 20 pF.

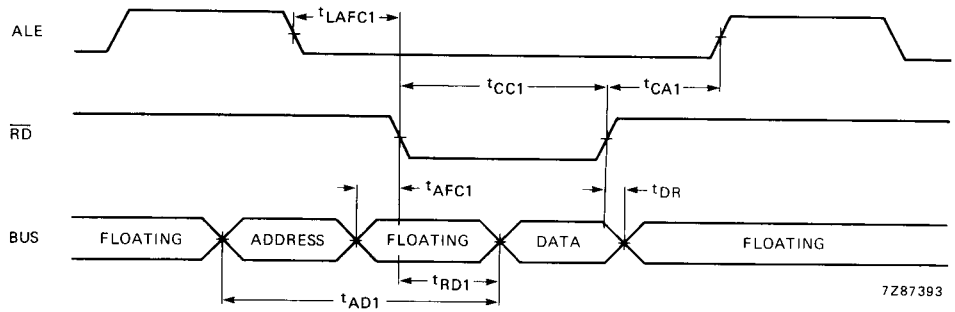


Fig.14 Read from external data memory.

DEVELOPMENT DATA

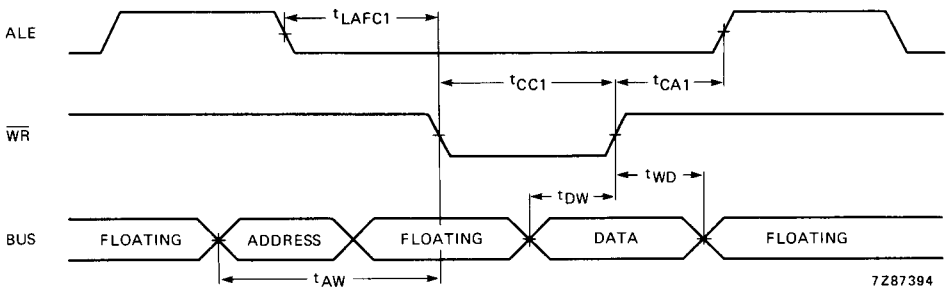


Fig.15 Write to external data memory.

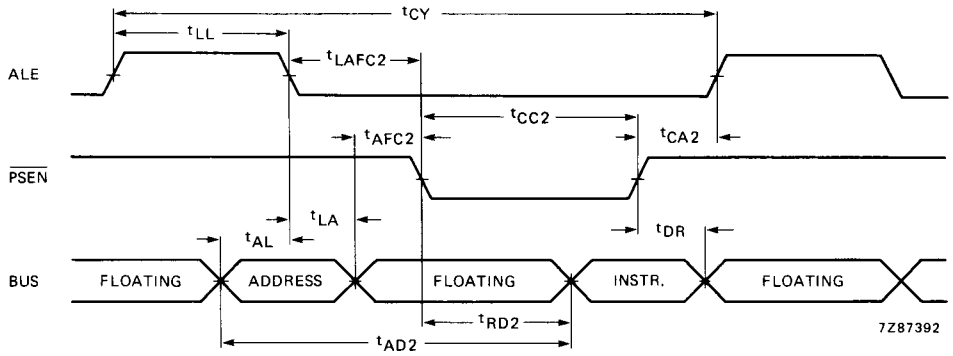


Fig.16 Instruction fetch from external program memory.

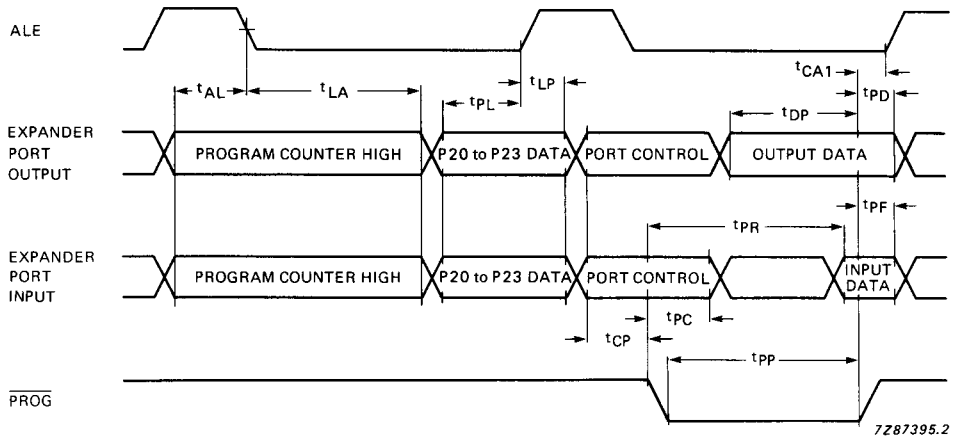


Fig.17 Port 2 timing.

DEVELOPMENT DATA

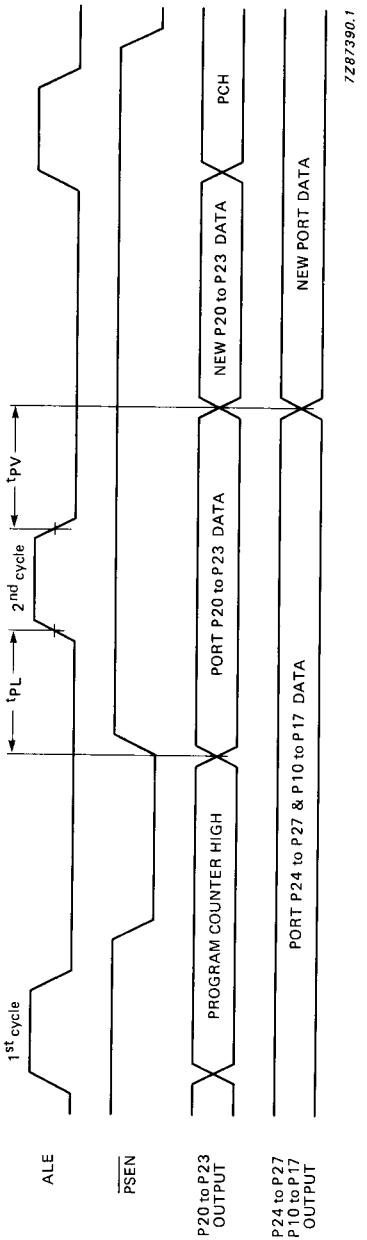


Fig.18 I/O port timing.