Z86C11 CMOS Z8® 4K ROM MCU

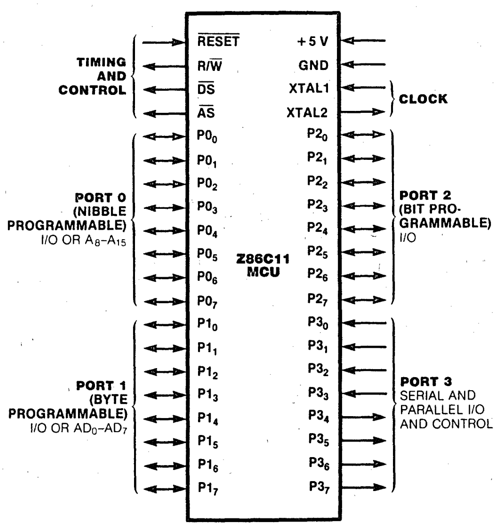
June 1987

FEATURES

- Complete microcomputer, 4K bytes of ROM, **256** bytes of RAM, 32 I/O lines, and up to 60K bytes addressable external space each for program and data memory.
- **256**-byte register file, including **236** general-purpose registers, four I/O port registers, and 16 status and control registers.
- Vectored, priority interrupts for I/O, counter/timers, and UART.
- Full-duplex UART and two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler.
- Register Pointer so that short, fast instructions can access any of **16** working-register groups in 1.5 μs.
- On-chip oscillator which accepts crystal or external clock drive.
- Standby modes—Halt and Stop
- Single +5V power supply—all pins TTL-compatible.
- **12 MHz, 16 MHz**
- CMOS process

GENERAL DESCRIPTION

The Z86C11 microcomputer (Figures 1 and 2) introduces a new level of sophistication to single-chip architecture. Compared to earlier single-chip microcomputers, the Z86C11 offers faster execution; more efficient use of memory; more sophisticated interrupt, input/output and bit-manipulation capabilities; and easier system expansion.



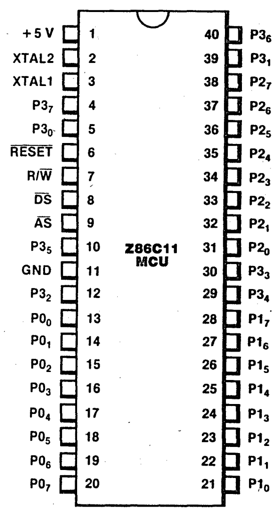


Figure 2. 40-pin Dual-In-Line Package (DIP), Pin Assignments

Under program control, the Z86C11 can be tailored to the needs of its user. It can be configured as a stand-alone microcomputer with 4K bytes of internal ROM, a traditional microprocessor that manages up to 120K bytes of external memory, or a parallel-processing element in a system with other processors and peripheral controllers linked by the Z-BUS® bus. In all configurations, a large number of pins remain available for I/O.

FIELD PROGRAMMABLE VERSION

The Z86E11 is a pin compatible "one time programmable" version of the Z86C11. The Z86C11 contains 4K bytes of EPROM memory in place of the 4K bytes of masked ROM in the Z86C11. The Z86E11 also contains a programmable memory protect feature to provide program security by disabling all external accesses to the internal EPROM array. This is preliminary information, and is subject to change.

ARCHITECTURE

Z86C11 architecture is characterized by a flexible I/O scheme, an efficient register and address space structure and a number of ancillary features that are helpful in many applications.

Microcomputer applications demand powerful I/O capabilities. The Z86C11 fulfills this with 32 pins dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable under software control to provide timing, status signals, serial or parallel I/O with or without handshake, and an address/data bus for interfacing external memory.

Because the multiplexed address/data bus is merged with the I/O-oriented ports, the Z86C11 can assume many different memory and I/O configurations. These configurations range from a self-contained microcomputer to a microprocessor that can address 120K bytes of external memory (Figure 3).

Three basic address spaces are available to support this wide range of configurations: program memory (internal and external), data memory (external) and the register file (internal). The 256-byte random-access register file is composed of 236 general-purpose registers, four I/O port registers, and 16 control and status registers.

To unburden the program from coping with real-time problems such as serial data communication and counting/timing, an asynchronous receiver/transmitter (UART) and two counter/timers with a large number of user-selectable modes are offered on-chip. Hardware support for the UART is minimized because one of the on-chip timers supplies the bit rate.

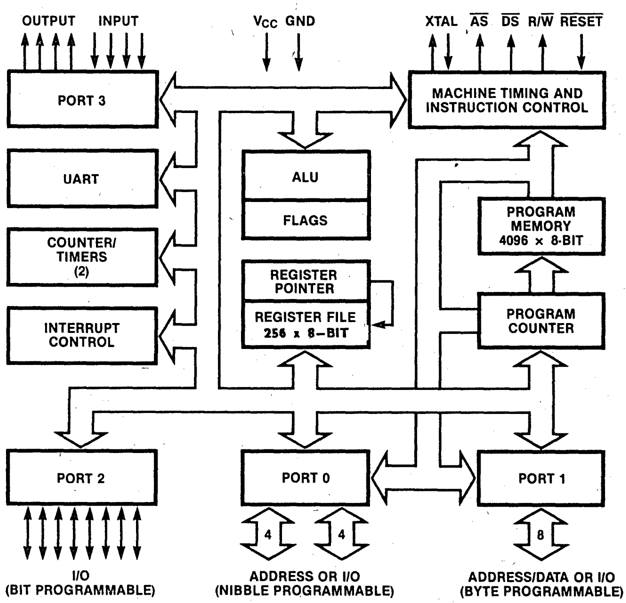


Figure 3. Functional Block Diagram

STANDBY MODE

The Z86C11's standby modes are:

- Stop
- Halt

The Stop instruction stops the internal clock and clock oscillation; the Halt instruction stops the internal clock but not clock oscillation.

A reset input releases the standby mode.

POWER DOWN INSTRUCTIONS

The Z86C91 has two instructions to reduce power consumption during standby operation. HALT turns off the processor and UART while the counter/timers and external interrupts IRQ0, IRQ1, and IRQ2 remain active.

When an interrupt occurs the processor resumes execution after servicing the interrupt. STOP turns off the clock to the entire Z86C91 and reduces the standby current to 10 microamps. The stop mode is terminated by reset, which causes the processor to restart the application program at address 12.

To complete an instruction prior to entering standby mode, use the instructions:

LD TMR, #00
NOP
STOP or HALT

PIN DESCRIPTION

$\overline{AS}$. *Address Strobe* (output, active Low). Address Strobe is pulsed once at the beginning of each machine cycle. Addresses output via Port 1 for all external program or data memory transfers are valid at the trailing edge of $\overline{AS}$. Under program control, $\overline{AS}$ can be placed in the high-impedance state along with Ports 0 and 1, Data Strobe and Read/Write.

$\overline{DS}$. *Data Strobe* (output, active Low). Data Strobe is activated once for each external memory transfer.

$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$. *I/O Port Lines* **(input/outputs, TTL–compatible). These 32 lines are divided into four 8–bit I/O ports that can be configured under program control for I/O or external memory interface (Figure 3).**

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ initializes the Z86C11. When $\overline{RESET}$ is deactivated, program execution begins from internal program location $000C_H$.

R/$\overline{W}$. *Read/Write* (output). R/$\overline{W}$ is Low when the Z86C11 is writing to external program or data memory.

XTAL1, XTAL2. Crystal 1, Crystal 2 (time–base input and output). These pins connect a parallel–resonant crystal (12 MHz maximum) or an external single–phase clock (12 MHz maximum) to the on–chip clock oscillator and buffer.

ADDRESS SPACE

Program Memory. The 16-bit program counter addresses 64K bytes of program memory space. Program memory can be located in two areas: one internal and the other external (Figure 4). The first 4096 bytes consist of on-chip mask-programmed ROM. At addresses 4096 and greater, the Z86C11 executes external program memory fetches.

The first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts.

Data Memory. The Z86C11 can address 60K bytes of external data memory beginning at location 4096 (Figure 5). External data memory may be included with or separated from the external program memory space. $\overline{DM}$, an optional I/O function that can be programmed to appear on pin $P3_4$, is used to distinguish between data and program memory space.

Register File. The **256**-byte register file includes four I/O port registers (R0-R3), **236** general-purpose registers (R4-R **239**) and 16 control and status registers (R240-R255). These registers are assigned the address locations shown in Figure 6.

Z86C11 instructions can access registers directly or indirectly with an 8–bit address field. The Z86C11 also allows short 4–bit register addressing using the Register Pointer (one of the control registers). In the 4–bit mode, the register file is divided into 16 working register groups, each occupying 16 contiguous locations (Figure 6). The Register Pointer addresses the starting location of the active working–register group (Figure 7).

Note: Register Bank E0-EF can only be accessed through working register and indirect addressing modes.

Stacks. Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory between locations 4096 and 65535. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4-R127).

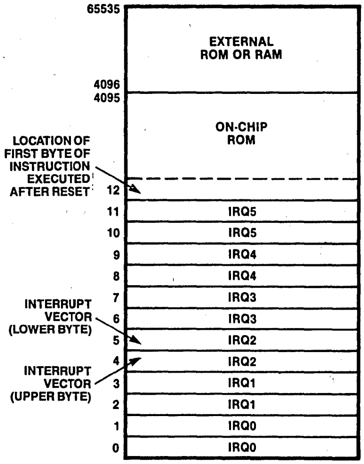


Figure 4. Program Memory Map

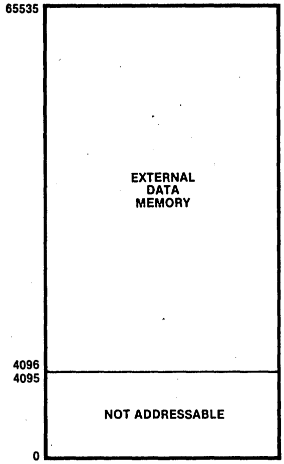


Figure 5. Data Memory Map

LOCATION		IDENTIFIERS
255	STACK POINTER (BITS 7-0)	SPL
254	STACK POINTER (BITS 15-8)	SPH
253	REGISTER POINTER	RP
252	PROGRAM CONTROL FLAGS	FLAGS
251	INTERRUPT MASK REGISTER	IMR
250	INTERRUPT REQUEST REGISTER	IRQ
249	INTERRUPT PRIORITY REGISTER	IPR
248	PORTS 0-1 MODE	P01M
247	PORT 3 MODE	P3M
246	PORT 2 MODE	P2M
245	T0 PRESCALER	PRE0
244	TIMER/COUNTER 0	T0
243	T1 PRESCALER	PRE1
242	TIMER/COUNTER 1	T1
241	TIMER MODE	TMR
240	SERIAL I/O	SIO
239–4	GENERAL-PURPOSE REGISTERS	
3	PORT 3	P3
2	PORT 2	P2
1	PORT 1	P1
0	PORT 0	P0

Figure 6. The Register File

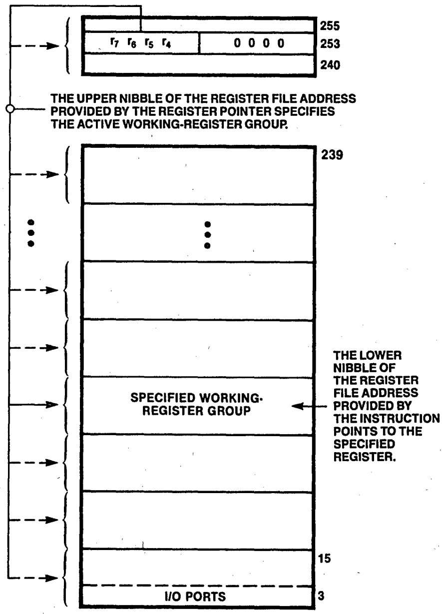


Figure 7. The Register Pointer

SERIAL INPUT/OUTPUT

Port 3 lines $P3_0$ and $P3_7$ can be programmed as serial I/O lines for full-duplex serial asynchronous receiver/transmitter operation. The bit rate is controlled by Counter/Timer 0, with a maximum rate of 62.5K bits/second for 8 MHz.

The Z86C11 automatically adds a start bit and two stop bits to transmitted data (Figure 8). Odd parity is also available as an option. Eight data bits are always transmitted, regardless of parity selection. If parity is enabled, the eighth bit is the odd parity bit. An interrupt request (IRQ_4) is generated on all transmitted characters.

Received data must have a start bit, eight data bits and at least one stop bit. If parity is on, bit 7 of the received data is replaced by a parity error flag. Received characters generate the IRQ_3 interrupt request.

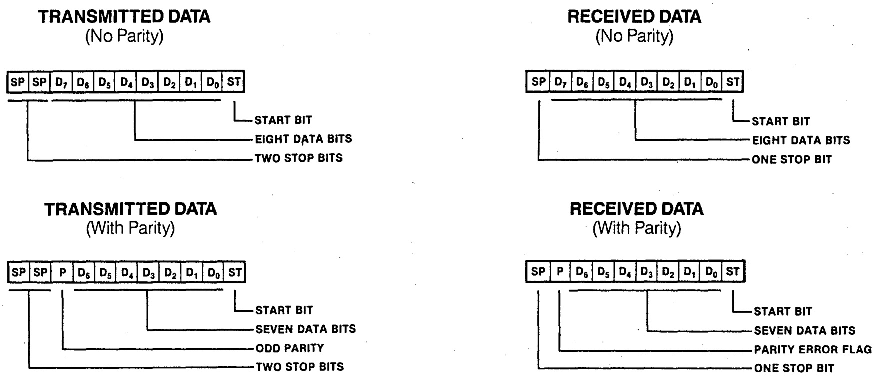


Figure 8. Serial Data Formats

COUNTER/TIMERS

The Z86C11 contains two 8-bit programmable counter/timers (T_0 and T_1), each driven by its own 6-bit programmable prescaler. The T_1 prescaler can be driven by internal or external clock sources; however, the T_0 prescaler is driven by the internal clock only.

The 6-bit prescalers can divide the input frequency of the clock source by any number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request—IRQ_4 (T_0) or IRQ_5 (T_1)—is generated.

The counters can be started, stopped, restarted to continue, or restarted from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode). The counters, but not the prescalers, can be read any time without disturbing their value or count mode.

The clock source for T_1 is user–definable and can be the internal microprocessor clock divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input as an external clock (1 MHz maximum), a trigger input that can be retriggerable or non–retriggerable, or as a gate input for the internal clock. The counter/timers can be programmably cascaded by connecting the T_0 output to the input of T_1. Port 3 line $P3_6$ also serves as a timer output (T_{OUT}) through which T_0, T_1 or the internal clock can be output.

I/O PORTS

The Z86C11 has 32 lines dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable as input, output or address/data. Under software control, the ports can be programmed to provide address outputs, timing, status signals, serial I/O, and parallel I/O with or without handshake. All ports have active pull-ups and pull-downs compatible with TTL loads.

Port 1 can be programmed as a byte I/O port or as an address/data port for interfacing external memory. When used as an I/O port, Port 1 may be placed under handshake control. In this configuration, Port 3 lines $P3_3$ and $P3_4$ are used as the handshake controls RDY_1 and $\overline{DAV}_1$ (Ready and Data Available).

Memory locations greater than 4096 are referenced through Port 1. To interface external memory, Port 1 must be programmed for the multiplexed Address/Data mode. If more than 256 external locations are required, Port 0 must output the additional lines.

Port 1 can be placed in the high-impedance state along with Port 0, $\overline{AS}$, $\overline{DS}$ and R/$\overline{W}$, allowing the Z86C11 to share common resources in multiprocessor and DMA applications. Data transfers can be controlled by assigning $P3_3$ as a Bus Acknowledge input, and $P3_4$ as a Bus Request output.

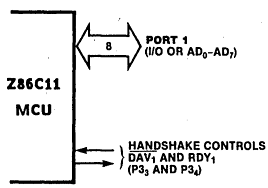


Figure 9a. Port 1

Port 0 can be programmed as a nibble I/O port, or as an address port for interfacing external memory. When used as an I/O port, Port 0 may be placed under handshake control. In this configuration, Port 3 lines $P3_2$ and $P3_5$ are used as the handshake controls $\overline{DAV}_0$ and RDY_0. Handshake signal assignment is dictated by the I/O direction of the upper nibble $P0_4$-$P0_7$.

For external memory references, Port 0 can provide address bits A_8-A_{11} (lower nibble) or A_8-A_{15} (lower and upper nibble) depending on the required address space. If the address range requires 12 bits or less, the upper nibble of Port 0 can be programmed independently as I/O while the lower nibble is used for addressing. When Port 0 nibbles are defined as address bits, they can be set to the high-impedance state along with Port 1 and the control signals $\overline{AS}$, $\overline{DS}$ and R/$\overline{W}$.

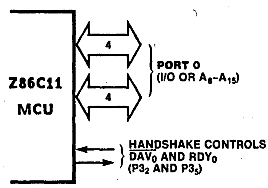


Figure 9b. Port 0

Port 2 bits can be programmed independently as input or output. This port is always available for I/O operations. In addition, Port 2 can be configured to provide open-drain outputs.

Like Ports 0 and 1, Port 2 may also be placed under handshake control. In this configuration, Port 3 lines $P3_1$ and $P3_6$ are used as the handshake controls lines $\overline{DAV}_2$ and RDY_2. The handshake signal assignment for Port 3 lines $P3_1$ and $P3_6$ is dictated by the direction (input or output) assigned to bit 7 of Port 2.

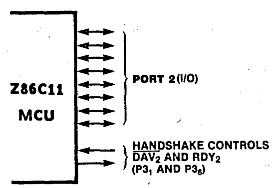


Figure 9c. Port 2

Port 3 lines can be configured as I/O or control lines. In either case, the direction of the eight lines is fixed as four input ($P3_0$-$P3_3$) and four output ($P3_4$-$P3_7$). For serial I/O, lines $P3_0$ and $P3_7$ are programmed as serial in and serial out respectively.

Port 3 can also provide the following control functions: handshake for Ports 0, 1 and 2 ($\overline{DAV}$ and RDY); four external interrupt request signals (IRQ_0-IRQ_3); timer input and output signals (T_{IN} and T_{OUT}) and Data Memory Select ($\overline{DM}$).

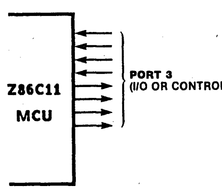


Figure 9d. Port 3

INTERRUPTS

The Z86C11 allows six different interrupts from eight sources: the four Port 3 lines $P3_0$-$P3_3$, Serial In, Serial Out, and the two counter/timers. These interrupts are both maskable and prioritized. The Interrupt Mask register globally or individually enables or disables the six interrupt requests. When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register.

All Z86C11 interrupts are vectored. When an interrupt request is granted, an interrupt machine cycle is entered. This disables all subsequent interrupts, saves the Program Counter and status flags, and branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request.

Polled interrupt systems are also supported. To accommodate a polled structure, any or all of the interrupt inputs can be masked and the Interrupt Request register polled to determine which of the interrupt requests needs service.

CLOCK

The on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal or to any suitable external clock source (XTAL1 = Input, XTAL2 = Output).

The crystal source is connected across XTAL1 and XTAL2, using the recommended capacitors ($C_1 \leqslant 15$ pf) from each pin to ground. The specifications for the crystal are as follows:

- AT cut, parallel resonant
- **Fundamental type, 12 MHz maximum**
- Series resistance, $R_s \leqslant 100\ \Omega$

INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate
R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working register pair address

Symbols. The following symbols are used in describing the instruction set.

dst	Destination location or contents
src	Source location or contents
cc	Condition code (see list)
@	Indirect address prefix
SP	Stack pointer (control registers 254-255)
PC	Program counter
FLAGS	Flag register (control register 252)
RP	Register pointer (control register 253)
IMR	Interrupt mask register (control register 251)

Assignment of a value is indicated by the symbol "←". For example,

$$dst \leftarrow dst + src$$

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr(n)" is used to refer to bit "n" of a given location. For example,

dst (7)

refers to bit 7 of the destination operand.

Flags. Control Register R252 contains the following six flags:

C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag

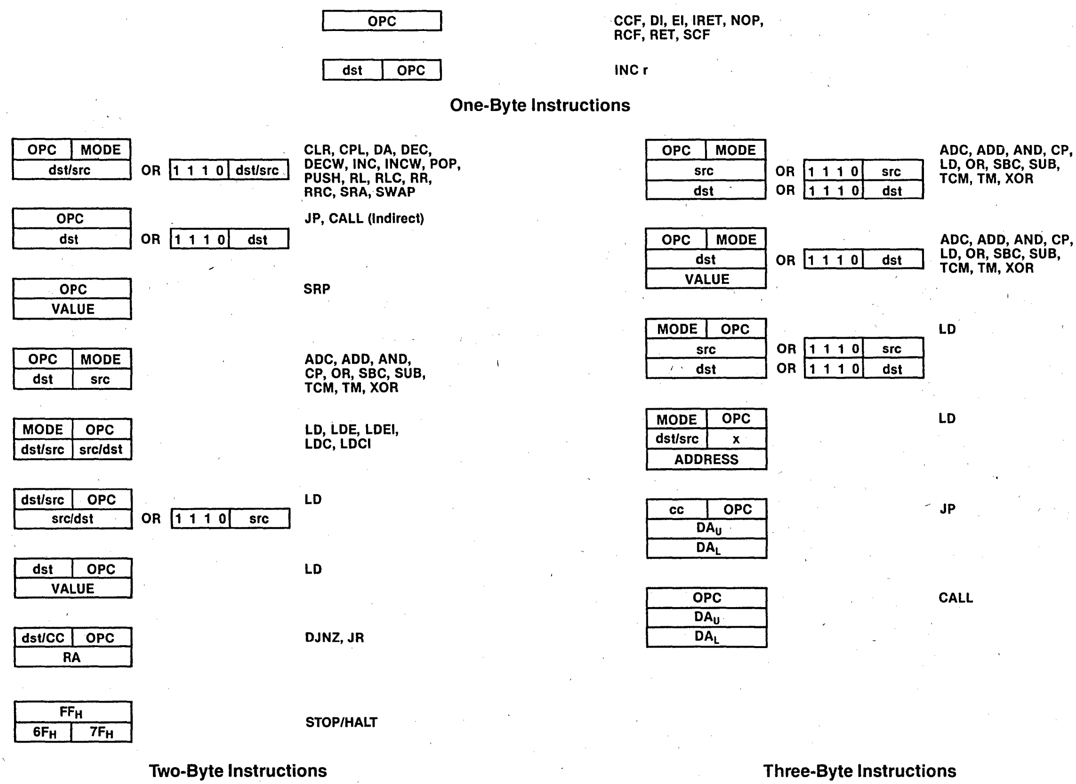
Affected flags are indicated by:

0	Cleared to zero
1	Set to one
*	Set or cleared according to operation
—	Unaffected
X	Undefined

CONDITION CODES

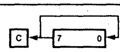
Value	Mnemonic	Meaning	Flags Set
1000		Always true	—
0111	C	Carry	C = 1
1111	NC	No carry	C = 0
0110	Z	Zero	Z = 1
1110	NZ	Not zero	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No overflow	V = 0
0110	EQ	Equal	Z = 1
1110	NE	Not equal	Z = 0
1001	GE	Greater than or equal	(S XOR V) = 0
0001	LT	Less than	(S XOR V) = 1
1010	GT	Greater than	[Z OR (S XOR V)] = 0
0010	LE	Less than or equal	[Z OR (S XOR V)] = 1
1111	UGE	Unsigned greater than or equal	C = 0
0111	ULT	Unsigned less than	C = 1
1011	UGT	Unsigned greater than	(C = 0 AND Z = 0) = 1
0011	ULE	Unsigned less than or equal	(C OR Z) = 1
0000		Never true	—

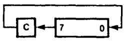
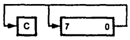
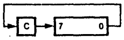
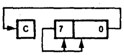
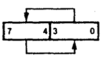
INSTRUCTION FORMATS



INSTRUCTION SUMMARY

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	C	Z	S	V	D	H
ADC dst,src dst ← dst + src + C	(Note 1)		1☐	*	*	*	*	0	*
ADD dst,src dst ← dst + src	(Note 1)		0☐	*	*	*	*	0	*
AND dst,src dst ← dst AND src	(Note 1)		5☐	—	*	*	0	—	—
CALL dst SP ← SP − 2 @SP ← PC; PC ← dst	DA IRR		D6 D4	—	—	—	—	—	—
CCF C ← NOT C			EF	*	—	—	—	—	—
CLR dst dst ← 0	R IR		B0 B1	—	—	—	—	—	—
COM dst dst ← NOT dst	R IR		60 61	—	*	*	0	—	—
CP dst,src dst − src	(Note 1)		A☐	*	*	*	*	—	—
DA dst dst ← DA dst	R IR		40 41	*	*	*	X	—	—
DEC dst dst ← dst − 1	R IR		00 01	—	*	*	*	—	—
DECW dst dst ← dst − 1	RR IR		80 81	—	*	*	*	—	—
DI IMR (7) ← 0			8F	—	—	—	—	—	—
DJNZ r,dst r ← r − 1 if r ≠ 0 PC ← PC + dst Range: +127, −128	RA		rA r = 0 − F	—	—	—	—	—	—
EI IMR (7) ← 1			9F	—	—	—	—	—	—
HALT			7F						
INC dst dst ← dst + 1	r R IR		rE r = 0 − F 20 21	—	*	*	*	—	—
INCW dst dst ← dst + 1	RR IR		A0 A1	—	*	*	*	—	—
IRET FLAGS ← @SP; SP ← SP + 1 PC ← @SP; SP ← SP + 2; IMR (7) ← 1			BF	*	*	*	*	*	*

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	C	Z	S	V	D	H
JP cc,dst if cc is true PC ← dst	DA IRR		cD c = 0 − F 30	—	—	—	—	—	—
JR cc,dst if cc is true, PC ← PC + dst Range: +127, −128	RA		cB c = 0 − F	—	—	—	—	—	—
LD dst,src dst ← src	r r R r X r Ir R R R IR IR	Im R r X r Ir r R IR IM IM R	rC r8 r9 r = 0 − F C7 D7 E3 F3 E4 E5 E6 E7 F5	—	—	—	—	—	—
LDC dst,src dst ← src	r Irr	Irr r	C2 D2	—	—	—	—	—	—
LDCI dst,src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	C3 D3	—	—	—	—	—	—
LDE dst,src dst ← src	r Irr	Irr r	82 92	—	—	—	—	—	—
LDEI dst,src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	83 93	—	—	—	—	—	—
NOP			FF	—	—	—	—	—	—
OR dst,src dst ← dst OR src	(Note 1)		4☐	—	*	*	0	—	—
POP dst dst ← @SP; SP ← SP + 1	R IR		50 51	—	—	—	—	—	—
PUSH src SP ← SP − 1; @SP ← src		R IR	70 71	—	—	—	—	—	—
RCF C ← 0			CF	0	—	—	—	—	—
RET PC ← @SP; SP ← SP + 2			AF	—	—	—	—	—	—
RL dst 	R IR		90 91	*	*	*	*	—	—

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	C	Z	S	V	D	H
RLC dst 	R IR		10 11	*	*	*	*	—	—
RR dst 	R IR		E0 E1	*	*	*	*	—	—
RRC dst 	R IR		C0 C1	*	*	*	*	—	—
SBC dst,src dst ← dst ← src ← C	(Note 1)		3☐	*	*	*	*	1	*
SCF C ← 1			DF	1	—	—	—	—	—
SRA dst 	R IR		D0 D1	*	*	*	0	—	—
SRP src RP ← src		Im	31	—	—	—	—	—	—
STOP			6F						
SUB dst,src dst ← dst ← src	(Note 1)		2☐	*	*	*	*	1	*
SWAP dst 	R IR		F0 F1	X	*	*	X	—	—
TCM dst,src (NOT dst) AND src	(Note 1)		6☐	—	*	*	0	—	—
TM dst,src dst AND src	(Note 1)		7☐	—	*	*	0	—	—
XOR dst,src dst ← dst XOR src	(Note 1)		B☐	—	*	*	0	—	—

NOTE: These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a ☐ in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

Addr Mode dst	Addr Mode src	Lower Opcode Nibble
r	r	2
r	Ir	3
R	R	4
R	IR	5
R	IM	6
IR	IM	7

R240 SIO
SERIAL I/O REGISTER
($F0_H$; Read/Write)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_7–D_0: SERIAL DATA (D_0 = LSB)

R241 TMR
TIMER MODE REGISTER
($F1_H$; Read/Write)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_0: 0 = NO FUNCTION; 1 = LOAD T_0
- D_1: 0 = DISABLE T_0 COUNT; 1 = ENABLE T_0 COUNT
- D_2: 0 = NO FUNCTION; 1 = LOAD T_1
- D_3: 0 = DISABLE T_1 COUNT; 1 = ENABLE T_1 COUNT
- D_5D_4: T_{IN} MODES
  - EXTERNAL CLOCK INPUT = 00
  - GATE INPUT = 01
  - TRIGGER INPUT = 10 (NON-RETRIGGERABLE)
  - TRIGGER INPUT = 11 (RETRIGGERABLE)
- D_7D_6: T_{OUT} MODES
  - NOT USED = 00
  - T_0 OUT = 01
  - T_1 OUT = 10
  - INTERNAL CLOCK OUT = 11

R242 T1
COUNTER TIMER 1 REGISTER
($F2_H$; Read/Write)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_7–D_0: T_1 INITIAL VALUE (WHEN WRITTEN) (RANGE 1-256 DECIMAL 01-00 HEX); T_1 CURRENT VALUE (WHEN READ)

R243 PRE1
PRESCALER 1 REGISTER
($F3_H$; Write Only)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_0: COUNT MODE
  - 0 = T_1 SINGLE-PASS
  - 1 = T_1 MODULO-N
- D_1: CLOCK SOURCE
  - 1 = T_1 INTERNAL
  - 0 = T_1 EXTERNAL TIMING INPUT (T_{IN}) MODE
- D_7–D_2: PRESCALER MODULO (RANGE: 1-64 DECIMAL 01-00 HEX)

R244 T0
COUNTER/TIMER 0 REGISTER
($F4_H$; Read/Write)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_7–D_0: T_0 INITIAL VALUE (WHEN WRITTEN) (RANGE: 1-256 DECIMAL 01-00 HEX); T_0 CURRENT VALUE (WHEN READ)

R245 PRE0
PRESCALER 0 REGISTER
($F5_H$; Write Only)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_0: COUNT MODE
  - 0 = T_0 SINGLE-PASS
  - 1 = T_0 MODULO-N
- D_1: RESERVED
- D_7–D_2: PRESCALER MODULO (RANGE: 1-64 DECIMAL 01-00 HEX)

R246 P2M
PORT 2 MODE REGISTER
($F6_H$; Write Only)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- D_7–D_0: $P2_0$-$P2_7$ I/O DEFINITION
  - 0 DEFINES BIT AS OUTPUT
  - 1 DEFINES BIT AS INPUT

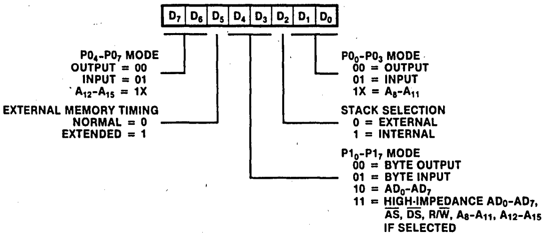
R247 P3M
PORT 3 MODE REGISTER
($F7_H$; Write Only)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

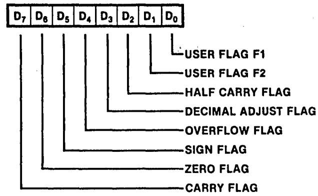
- D_0: 0 PORT 2 PULL-UPS OPEN DRAIN; 1 PORT 2 PULL-UPS ACTIVE
- D_1: RESERVED
- D_2: 0 P32 = INPUT, P35 = OUTPUT; 1 P32 = $\overline{DAV0}$/RDY0, P35 = RDY0/$\overline{DAV0}$
- D_4D_3:
  - 0 0 P33 = INPUT, P34 = OUTPUT
  - 0 1 / 1 0 P33 = INPUT, P34 = $\overline{DM}$
  - 1 1 P33 = $\overline{DAV1}$/RDY1, P34 = RDY1/$\overline{DAV1}$
- D_5: 0 P31 = INPUT (T_{IN}), P36 = OUTPUT (T_{OUT}); 1 P31 = $\overline{DAV2}$/RDY2, P36 = RDY2/$\overline{DAV2}$
- D_6: 0 P30 = INPUT, P37 = OUTPUT; 1 P30 = SERIAL IN, P37 = SERIAL OUT
- D_7: 0 PARITY OFF; 1 PARITY ON

Figure 11. Control Registers

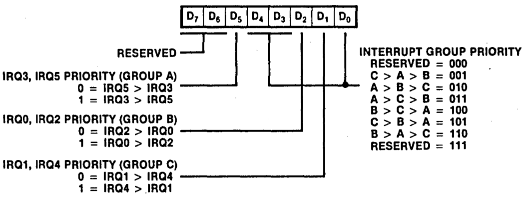
R248 P01M
PORT 0 AND 1 MODE REGISTER
(F8H; Write Only)



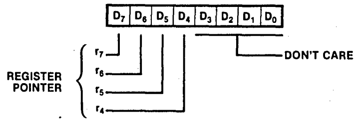
R252 FLAGS
FLAG REGISTER
(FCH; Read/Write)



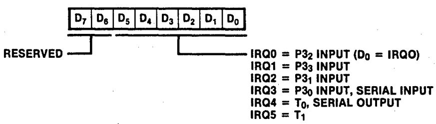
R249 IPR
INTERRUPT PRIORITY REGISTER
(F9H; Write Only)



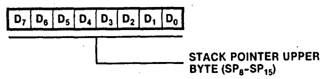
R253 RP
REGISTER POINTER
(FDH; Read/Write)



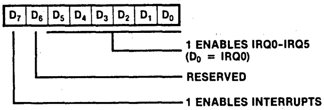
R250 IRQ
INTERRUPT REQUEST REGISTER
(FAH; Read/Write)



R254 SPH
STACK POINTER
(FEH; Read/Write)



R251 IMR
INTERRUPT MASK REGISTER
(FBH; Read/Write)



R255 SPL
STACK POINTER
(FFH; Read/Write)

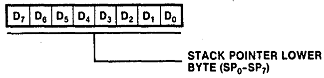
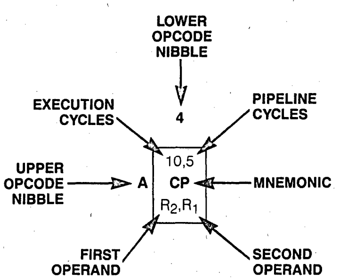


Figure 11. Control Registers (Continued)

OPCODE MAP

Lower Nibble (Hex) across the top, Upper Nibble (Hex) down the side.

Upper \ Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6.5 **DEC** R_1	6.5 **DEC** IR_1	6.5 **ADD** r_1,r_2	6.5 **ADD** r_1,Ir_2	10.5 **ADD** R_2,R_1	10.5 **ADD** IR_2,R_1	10.5 **ADD** R_1,IM	10.5 **ADD** IR_1,IM	6.5 **LD** r_1,R_2	6.5 **LD** r_2,R_1	12/10.5 **DJNZ** r_1,RA	12/10.0 **JR** cc,RA	6.5 **LD** r_1,IM	12/10.0 **JP** cc,DA	6.5 **INC** r1	
1	6,5 **RLC** R_1	6,5 **RLC** IR_1	6,5 **ADC** r_1,r_2	6,5 **ADC** r_1,Ir_2	10,5 **ADC** R_2,R_1	10,5 **ADC** IR_2,R_1	10,5 **ADC** R_1,IM	10,5 **ADC** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	
2	6,5 **INC** R_1	6,5 **INC** IR_1	6,5 **SUB** r_1,r_2	6,5 **SUB** r_1,Ir_2	10,5 **SUB** R_2,R_1	10,5 **SUB** IR_2,R_1	10,5 **SUB** R_1,IM	10,5 **SUB** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	
3	8,0 **JP** IRR_1	6,1 **SRP** IM	6,5 **SBC** r_1,r_2	6,5 **SBC** r_1,Ir_2	10,5 **SBC** R_2,R_1	10,5 **SBC** IR_2,R_1	10,5 **SBC** R_1,IM	10,5 **SBC** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	
4	8,5 **DA** R_1	8,5 **DA** IR_1	6,5 **OR** r_1,r_2	6,5 **OR** r_1,Ir_2	10,5 **OR** R_2,R_1	10,5 **OR** IR_2,R_1	10,5 **OR** R_1,IM	10,5 **OR** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	
5	10,5 **POP** R_1	10,5 **POP** IR_1	6,5 **AND** r_1,r_2	6,5 **AND** r_1,Ir_2	10,5 **AND** R_2,R_1	10,5 **AND** IR_2,R_1	10,5 **AND** R_1,IM	10,5 **AND** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	
6	6,5 **COM** R_1	6,5 **COM** IR_1	6,5 **TCM** r_1,r_2	6,5 **TCM** r_1,Ir_2	10,5 **TCM** R_2,R_1	10,5 **TCM** IR_2,R_1	10,5 **TCM** R_1,IM	10,5 **TCM** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	6,0 **STOP**
7	10/12,1 **PUSH** R_2	12/14,1 **PUSH** IR_2	6,5 **TM** r_1,r_2	6,5 **TM** r_1,Ir_2	10,5 **TM** R_2,R_1	10,5 **TM** IR_2,R_1	10,5 **TM** R_1,IM	10,5 **TM** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	7,0 **HALT**
8	10,5 **DECW** RR_1	10,5 **DECW** IR_1	12,0 **LDE** r_1,Irr_2	18,0 **LDEI** Ir_1,Irr_2					↓	↓	↓	↓	↓	↓	↓	6.1 **DI**
9	6,5 **RL** R_1	6,5 **RL** IR_1	12,0 **LDE** r_2,Irr_1	18,0 **LDEI** Ir_2,Irr_1					↓	↓	↓	↓	↓	↓	↓	6.1 **EI**
A	10,5 **INCW** RR_1	10,5 **INCW** IR_1	6,5 **CP** r_1,r_2	6,5 **CP** r_1,Ir_2	10,5 **CP** R_2,R_1	10,5 **CP** IR_2,R_1	10,5 **CP** R_1,IM	10,5 **CP** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	14.0 **RET**
B	6,5 **CLR** R_1	6,5 **CLR** IR_1	6,5 **XOR** r_1,r_2	6,5 **XOR** r_1,Ir_2	10,5 **XOR** R_2,R_1	10,5 **XOR** IR_2,R_1	10.5 **XOR** R_1,IM	10,5 **XOR** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	16.0 **IRET**
C	6,5 **RRC** R_1	6,5 **RRC** IR_1	12,0 **LDC** r_1,Irr_2	18,0 **LDCI** Ir_1,Irr_2				10,5 **LD** r_1,x,R_2	↓	↓	↓	↓	↓	↓	↓	6.5 **RCF**
D	6,5 **SRA** R_1	6,5 **SRA** IR_1	12,0 **LDC** r_2,Irr_1	18,0 **LDCI** Ir_2,Irr_1	20,0 **CALL*** IRR_1		20,0 **CALL** DA	10,5 **LD** r_2,x,R_1	↓	↓	↓	↓	↓	↓	↓	6.5 **SCF**
E	6,5 **RR** R_1	6,5 **RR** IR_1		6,5 **LD** r_1,IR_2	10,5 **LD** R_2,R_1	10,5 **LD** IR_2,R_1	10,5 **LD** R_1,IM	10,5 **LD** IR_1,IM	↓	↓	↓	↓	↓	↓	↓	6.5 **CCF**
F	8,5 **SWAP** R_1	8,5 **SWAP** IR_1		6,5 **LD** Ir_1,r_2		10,5 **LD** R_2,IR_1			↓	↓	↓	↓	↓	↓	↓	6.0 **NOP**
Bytes per Instruction	2	2	2	2	3	3	3	3	2	2	2	2	2	3	1	1



Legend:
R = 8-bit address
r = 4-bit address
R_1 or r_1 = Dst address
R_2 or r_2 = Src address

Sequence:
Opcode, First Operand, Second Operand

NOTE: The blank areas are not defined.

*2-byte instruction; fetch cycle appears as a 3-byte instruction

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins with respect
to GND . . . . –0.3V to +7.0V
Operating Ambient
Temperature . . . . .See Ordering Information
Storage Temperature . . . . –65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

The DC characteristics listed below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

Standard conditions are as follows:

- **+4.5 ≤ Vcc ≤ + 5.5V**
- GND = 0V
- **0 C ≤ T_A ≤ +70 C for S (Standard temperature)**
- **–40 C ≤ T_A ≤+100 C for E (Extended temperature)**

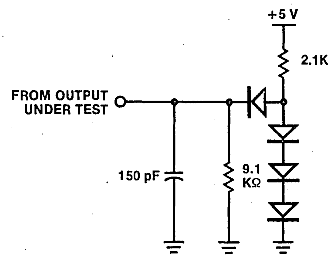


Figure 12. Test Load 1

DC CHARACTERISTICS

Symbol	Parameter	Min	Typ	Max	Unit	Condition
V_{CH}	Clock Input High Voltage	3.8		V_{CC}	V	Driven by External Clock Generator
V_{CL}	Clock Input Low Voltage	–0.3		0.8	V	Driven by External Clock Generator
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{IL}	Input Low Voltage	–0.3		0.8	V	
V_{RH}	Reset Input High Voltage	3.8		V_{CC}	V	
V_{RL}	Reset Input Low Voltage	–0.3		0.8	V	
V_{OH}	Output High Voltage	2.4			V	I_{OH} = –250 μA
V_{OH}	**Output High Voltage**	**V_{CC} -100mV**			**V**	**I_{OH} = -100μA**
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage	–10		10	μA	0V ≤ V_{IN} ≤ + 5.25V
I_{OL}	Output Leakage	–10		10	μA	0V ≤ V_{IN} ≤ + 5.25V
I_{IR}	Reset Input Current			–50	μA	V_{CC} = +5.25V, V_{RL} = 0V
I_{CC}	Supply Current			30	mA	All outputs and I/O pins floating, 12 MHz
I_{CC1}	Standby Current		**5**		mA	Halt Mode
I_{CC2}	Standby Current			10	μA	Stop Mode

$I_{CC}2$ requires loading TMR (%F1) with any value prior to STOP execution.

Use the sequence:

```
LD TMR, #00
NOP
STOP
```

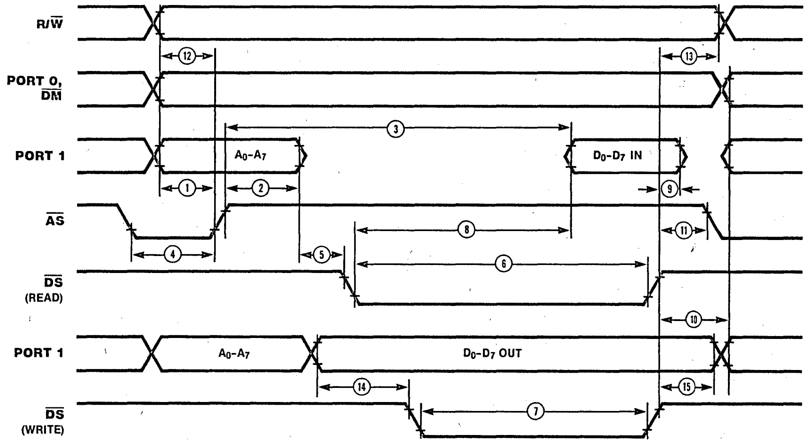


Figure 13. External I/O or Memory Read/Write

AC CHARACTERISTICS

External I/O or Memory Read and Write Timing

Number	Symbol	Parameter	12 MHz Min	12 MHz Max	16 MHz Min	16 MHz Max	Notes*†°
1	TdA(AS)	Address Valid to $\overline{AS}$ ↑ Delay	35		20		2,3
2	TdAS(A)	$\overline{AS}$ ↑ to Address Float Delay	45		30		2,3
3	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Required Valid		220		180	1,2,3
4	TwAS	$\overline{AS}$ Low Width	55		35		2,3
5	TdAz(DS)	Address Float to $\overline{DS}$ ↓	0		0		
6	TwDSR	$\overline{DS}$ (Read) Low Width	185		135		1,2,3
7	TwDSW	$\overline{DS}$ (Write) Low Width	110		80		1,2,3
8	TdDSR(DR)	$\overline{DS}$ ↓ to Read Data Required Valid		130		75	1,2,3
9	ThDR(DS)	Read Data to $\overline{DS}$ ↑ Hold Time	0		0		
10	TdDS(A)	$\overline{DS}$ ↑ to Address Active Delay	45		20		2,3
11	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	55		20		2,3
12	TdR/W(AS)	R/$\overline{W}$ Valid to $\overline{AS}$ ↑ Delay	30		20		2,3
13	TdDS(R/W)	$\overline{DS}$ ↑ to R/W Not Valid	35		20		2,3
14	TdDW(DSW)	Write Data Valid to $\overline{DS}$ (Write) ↓ Delay	35		25		2,3
15	TdDS(DW)	$\overline{DS}$ ↑ to Write Data Not Valid Delay	35		20		2,3
16	TdA(DR)	Address Valid to Read Data Required Valid		255		200	1,2,3
17	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	55		40		2,3

NOTES:
1. When using extended memory timing add 2 TpC.
2. Timing numbers given are for minimum TpC.
3. See clock cycle time dependent characteristics table.

* All units in nanoseconds (ns).
† Test Load 1
° All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".

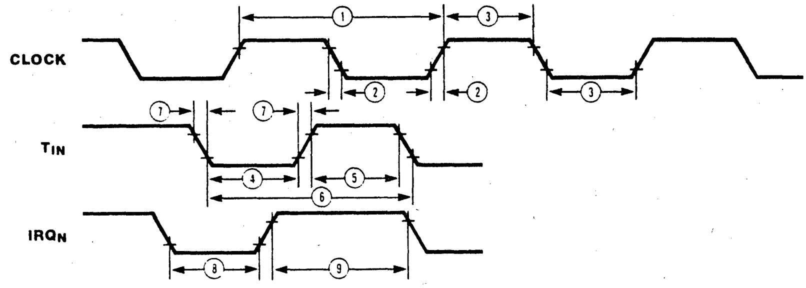


Figure 14. Additional Timing

AC CHARACTERISTICS

Additional Timing Table

Number	Symbol	Parameter	12 MHz Min	12 MHz Max	16 MHz Min	16 MHz Max	Notes*
1	TpC	Input Clock Period	83	1000	62.5	1000	1
2	TrC,TfC	Clock Input Rise and Fall Times		15		10	1
3	TwC	Input Clock Width	70		21		1
4	TwTinL	Timer Input Low Width	70		50		2
5	TwTinH	Timer Input High Width	3TpC		3TpC		2
6	TpTin	Timer Input Period	8TpC		8TpC		2
7	TrTin,TfTin	Timer Input Rise and Fall Times		100		100	2
8A	TwIL	Interrupt Request Input Low Time	70		50		2,4
8B	TwIL	Interrupt Request Input Low Time	3TpC		3TpC		2,5
9	TwIH	Interrupt Request Input High Time	3TpC		3TpC		2,3

NOTES:
1. Clock timing references use 3.8V for a logic "1" and 0.8V for a logic "0".
2. Timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
3. Interrupt request via Port 3.
4. Interrupt request via Port 3 ($P3_1$-$P3_3$).
5. Interrupt request via Port 3 ($P3_0$).
* Units in nanoseconds (ns).

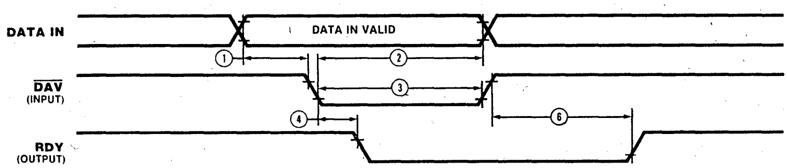


Figure 15a. Input Handshake

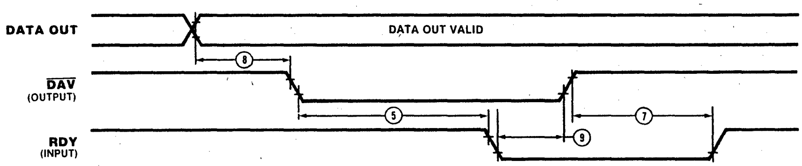


Figure 15b. Output Handshake

AC CHARACTERISTICS
Handshake Timing

Number	Symbol	Parameter	12MHz, 16MHz Min	12MHz, 16MHz Max	Notes†*
1	TsDI(DAV)	Data In Setup Time	0		
2	ThDI(DAV)	Data In Hold Time	145		
3	TwDAV	Data Available Width	110		
4	TdDAVIf(RDY)	$\overline{DAV}$ ↓ Input to RDY ↓ Delay	20	115	1,2
5	TdDAVOf(RDY)	$\overline{DAV}$ ↓ Output to RDY ↓ Delay	0		1,3
6	TdDAVIr(RDY)	$\overline{DAV}$ ↑ Input to RDY ↑ Delay		115	1,2
7	TdDAVOr(RDY)	$\overline{DAV}$ ↑ Output to RDY ↑ Delay	0		1,3
8	TdDO(DAV)	Data Out to $\overline{DAV}$ ↓ Delay	Tpc		1
9	TdRDY(DAV)	RDY ↓ Input to $\overline{DAV}$ ↑ Delay	0	130	1

NOTES:
1. Test load 1
2. Input handshake
3. Output handshake
† All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
* Units in nanoseconds (ns).