T-49-19-07

June 1987

Z8601/Z8603
Z8611/Z8613 Z8®

Z8601 Single-Chip MCU with 2K ROM
Z8603 Prototyping Device with 2K EPROM Interface
Z8611 Single-Chip MCU with 4K ROM
Z8613 Prototyping Device with 4K EPROM Interface

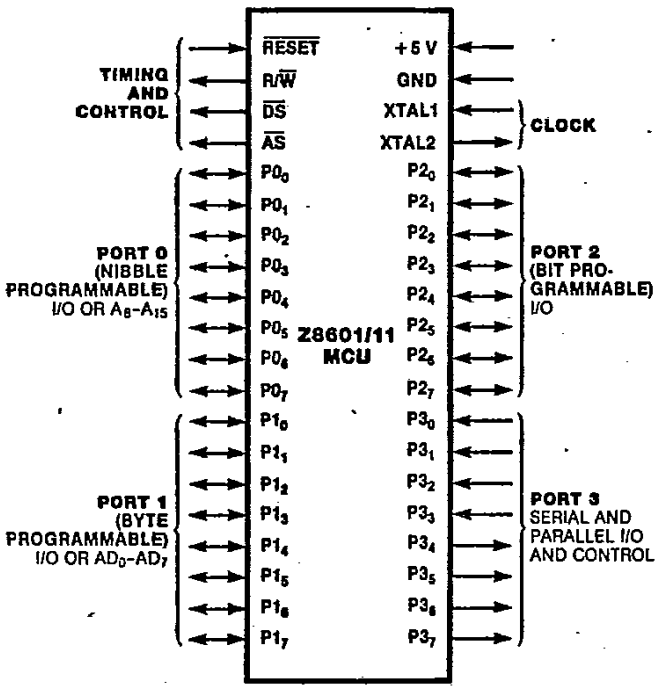
Features

- Complete microcomputer, 2K (8601) or 4K (8611) bytes of ROM, 128 bytes of RAM, 32 I/O lines, and up to 62K (8601) or 60K (8611) bytes addressable external space each for program and data memory.
- 144-byte register file, including 124 general-purpose registers, four I/O port registers, and 16 status and control registers.
- Average instruction execution time of 1.5 μs, maximum of 1 μs.
- 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 nine working register groups in 1 μs.
- On-chip oscillator which accepts crystal or external clock drive.
- Single +5 V power supply—all pins TTL compatible.
- 12.5 MHz.

General Description

The Z8 microcomputer introduces a new level of sophistication to single-chip architecture. Compared to earlier single-chip microcomputers, the Z8 offers faster execution; more efficient use of memory; more sophisticated interrupt, input/output and bit-manipulation capabilities; and easier system expansion.

Under program control, the Z8 can be tailored to the needs of its user. It can be configured as a stand-alone microcomputer with 2K or 4K bytes of internal ROM, a traditional microprocessor that manages up to 124K 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.



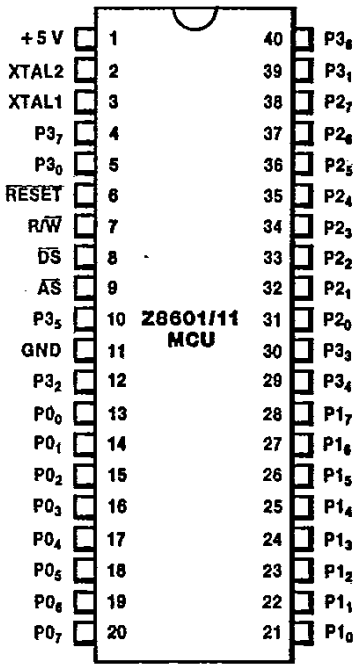


Figure 2a. 40-pin Dual-In-Line Package (DIP), Pin Assignments

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.

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ initializes the Z8. When $\overline{RESET}$ is deactivated, program execution begins from internal program location $000C_H$.

$\overline{ROMless}$. (input, active LOW). This pin is only available on the 44 pin version of the Z8611. When connected to GND disables the internal ROM and forces the part to function as a Z8681 ROMless Z8. When left unconnected or pulled high to V_{cc} the part will function normally as a Z8611.

R/$\overline{W}$. *Read/Write* (output). R/$\overline{W}$ is Low when the Z8 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 12.5 MHz crystal or an external single-phase 12.5 MHz clock to the on-chip clock oscillator and buffer.

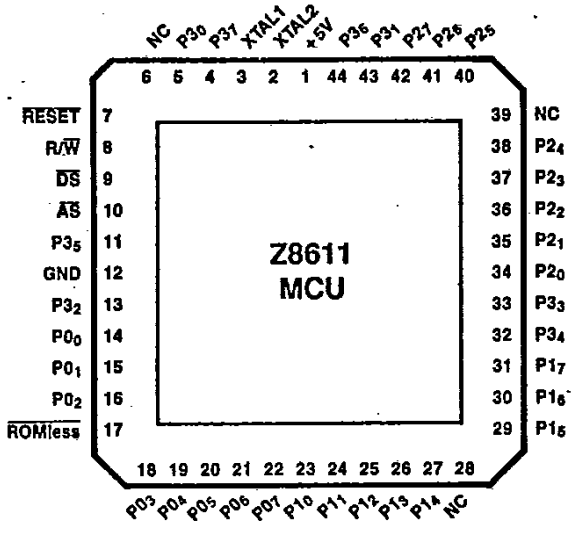


Figure 2b. 44-pin Chip Carrier, Pin Assignments

Architecture

Z8 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 Z8 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 Z8 can assume many different memory and I/O configurations. These configurations range from a self-contained microcomputer to a microprocessor that can address 124K (Z8601) or 120K (Z8611) bytes of external memory.

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 144-byte random-access register file is composed of 124 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.

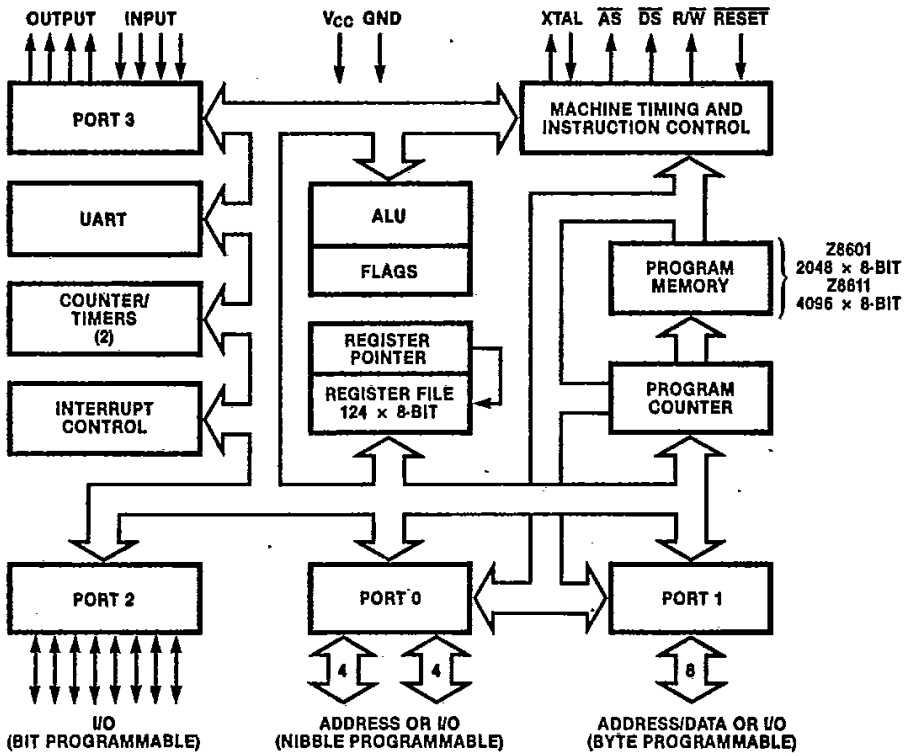


Figure 3. Functional Block Diagram

Address Spaces

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 2048 (Z8601) or 4096 (Z8611) bytes consist of on-chip mask-programmed ROM. At addresses 2048 (Z8601) or 4096 (Z8611) and greater, the Z8 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 Z8 can address 62K (Z8601) or 60K (Z8611) bytes of external data memory beginning at location 2048 (Z8601) or 4096 (Z8611) (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 144-byte register file includes four I/O port registers (R0–R3), 124 general-purpose registers (R4–R127) and 16 control and status registers (R240–R255). These registers are assigned the address locations shown in Figure 6.

Z8 instructions can access registers directly or indirectly with an 8-bit address field. The Z8 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

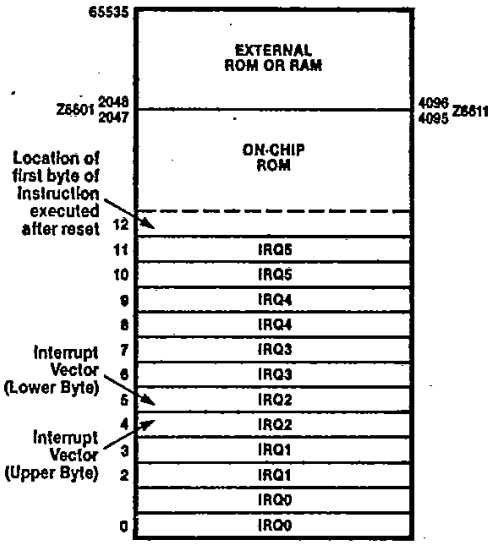


Figure 4. Program Memory Map

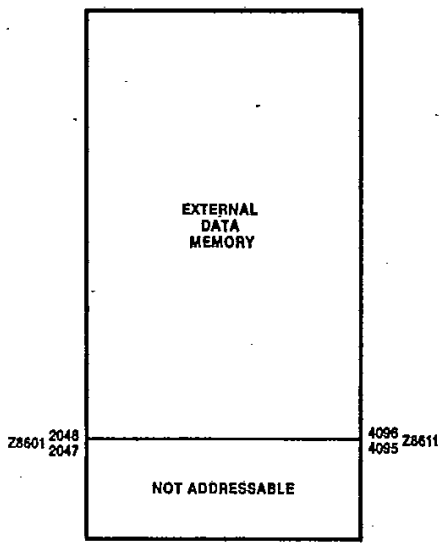


Figure 5. Data Memory Map

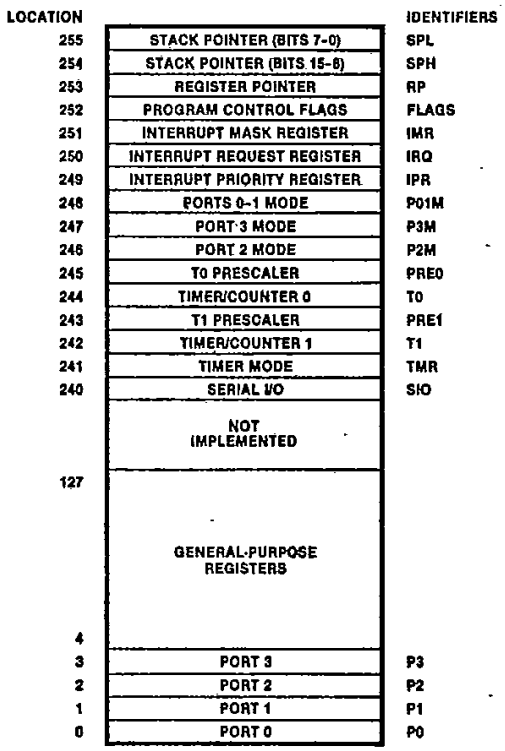


Figure 6. The Register File

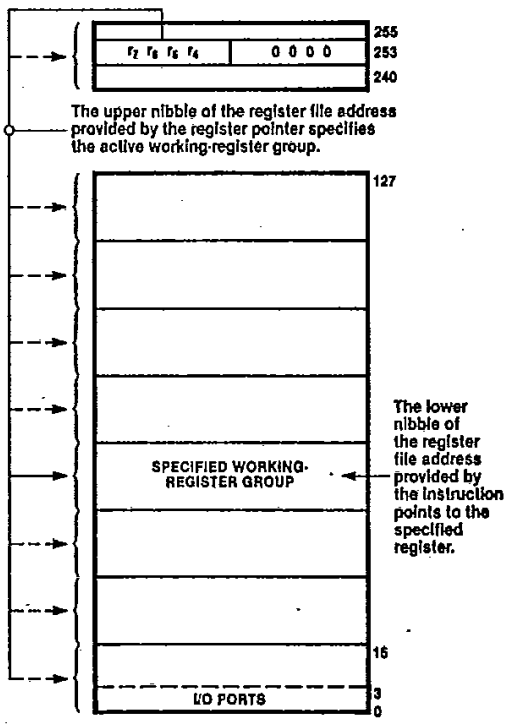


Figure 7. The Register Pointer

divided into nine working-register groups, each occupying 16 continguous locations (Figure 6). The Register Pointer addresses the starting location of the active working-register group.

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 2048 (8601) or 4096 (8611) and 65535. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4–R127).

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, at 12 MHz.

The Z8 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.

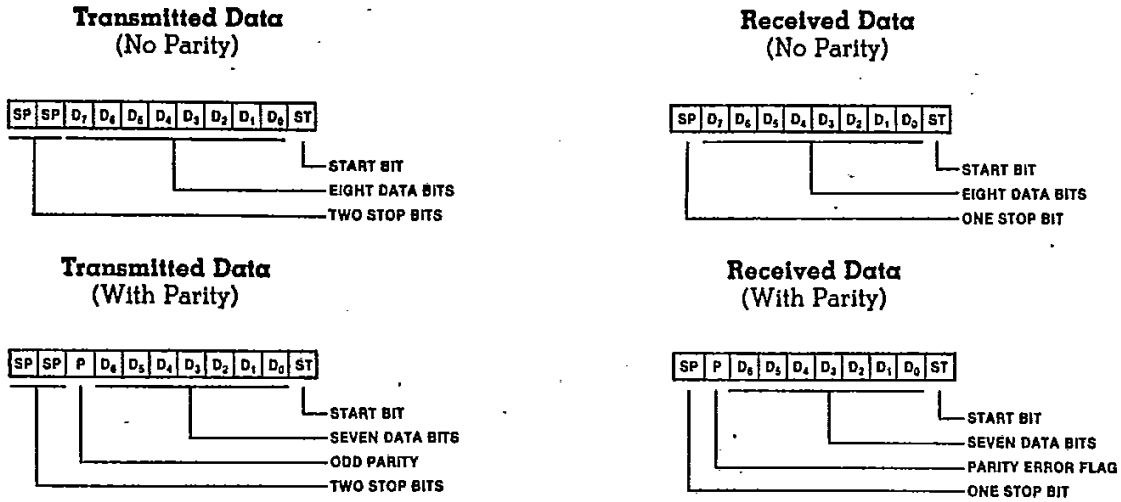


Figure 8. Serial Data Formats

Counter/ Timers

The Z8 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, a trigger input that can be retriggerable or nonretriggerable, 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 Z8 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 2048 (Z8601) or 4096 (Z8611) 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 Z8 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.

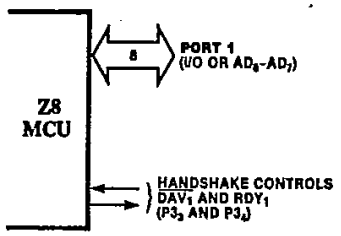


Figure 9a. Port 1

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 highimpedance state along with Port 1 and the control signals $\overline{AS}$, $\overline{DS}$ and $R/\overline{W}$.

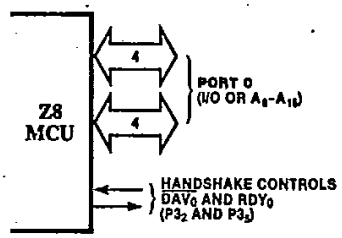


Figure 9b. Port 0

Port 2 bits can be programmed independently as input or output. The 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.

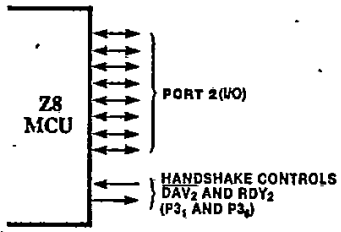


Figure 9c. Port 2

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 ($\overline{DM}$).

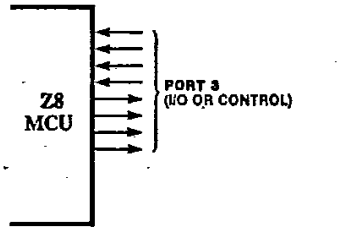


Figure 9d. Port 3

Interrupts

The Z8 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 Z8 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 ($C_1 \leq 15$ pF) from each pin to ground. The specifications for the crystal are as follows:

- AT cut, parallel resonant
- Fundamental type, 12.5 MHz maximum
- Series resistance, $R_s \leq 100\ \Omega$

Z8603/13 Protopack Emulator

The Z8 Protopack is used for prototype development and preproduction of mask-programmed applications. The Protopack is a ROMless version of the standard Z8601 or Z8611 housed in a pin-compatible 40-pin package (Figure 11).

To provide pin compatibility and interchangeability with the standard maskprogrammed device, the Protopack carries piggy-back a 24-pin socket for a direct interface to program memory (Figure 1). The Z8603 24-pin socket is equipped with 11 ROM address lines, 8 ROM data lines and necessary control lines for interface to 2716 EPROM for the first 2K bytes of program memory. The Z8613 24-pin socket is equipped with 12 ROM address lines, 8 ROM data lines and necessary control lines for interface to 2732 EPROM for the first 4K bytes of program memory.

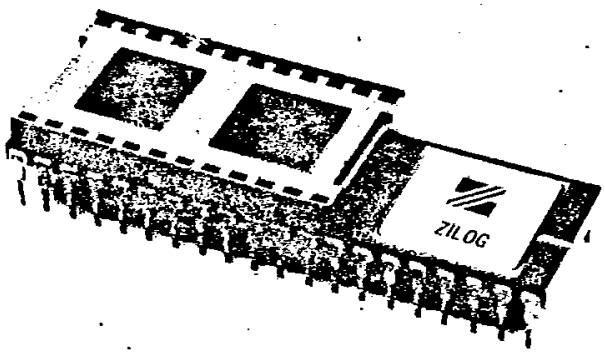


Figure 11. The Z8 Microcomputer Protopack Emulator

Pin compatibility allows the user to design the pc board for a final 40-pin maskprogrammed Z8, and, at the same time, allows the use of the Protopack to build the prototype and pilot production units. When the final program is established, the user can then switch over to the 40-pin mask-programmed Z8 for large volume production. The Protopack is also useful in small volume applica tions where masked ROM setup time, mask charges, etc., are prohibitive and program flexibility is desired.

Compared to the conventional EPROM versions of the single-chip microcomputers, the Protopack approach offers two main advantages:

- Ease of developing various programs during the prototyping stage. For instance, in applications where the same hardware configuration is used with more than one program, the Protopack allows economical program storage in separate EPROMs (or PROMs), whereas the use of separate EPROM-based single-chip microcomputers is more costly.
- Elimination of long lead time in procuring EPROM-based microcomputers.

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 ← 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

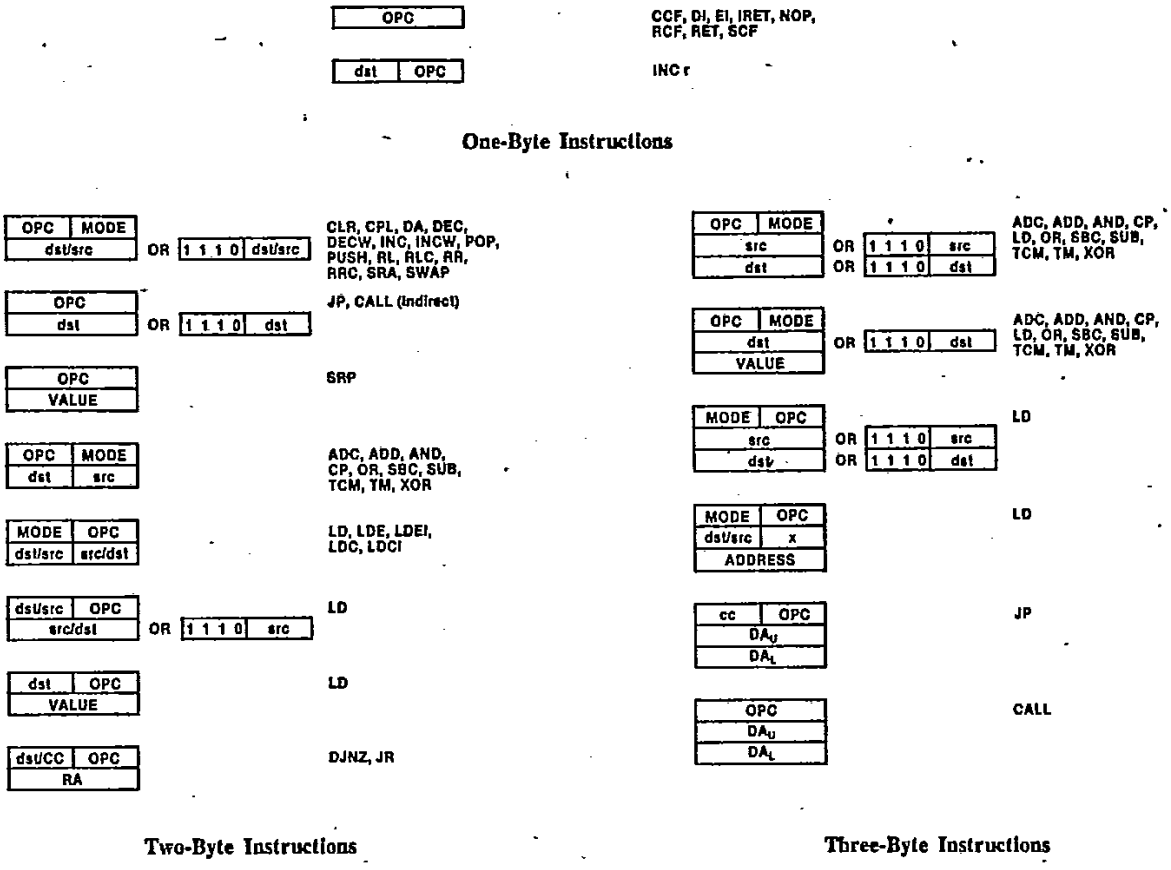


Figure 12. Instruction Formats

Instruction Summary

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected 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	- - - - - -
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	* * * * * *
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	- - - - - -

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C Z S V D H
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	* * * * - -
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	- - - - - -
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 - -

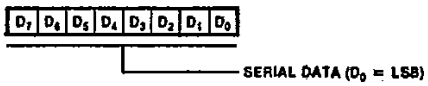
Note 1

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 right of the applicable addressing mode pair.

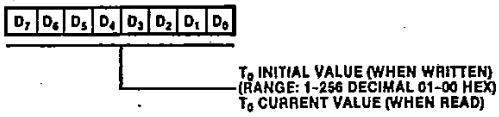
For example, to determine the opcode of a ADC instruction use the addressing modes r (destination) and Ir (source). The result is 13.

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

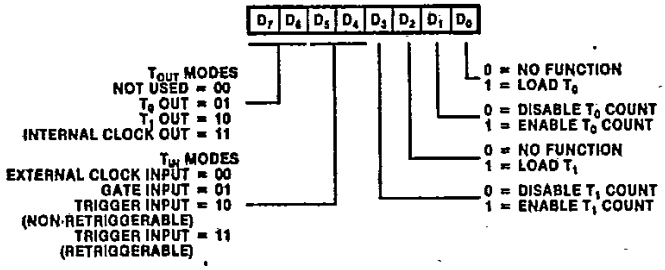
R240 SIO
Serial I/O Register
($F0_H$; Read/Write)



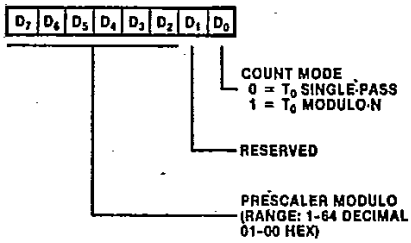
R244 T0
Counter/Timer 0 Register
($F4_H$; Read/Write)



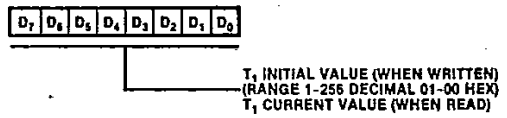
R241 TMR
Timer Mode Register
($F1_H$; Read/Write)



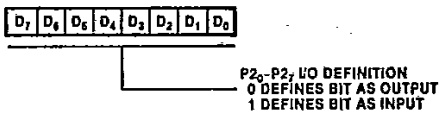
R245 PRE0
Prescaler 0 Register
($F5_H$; Write Only)



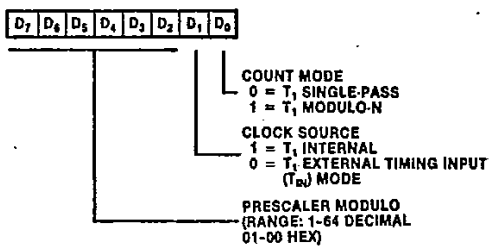
R242 T1
Counter Timer 1 Register
($F2_H$; Read/Write)



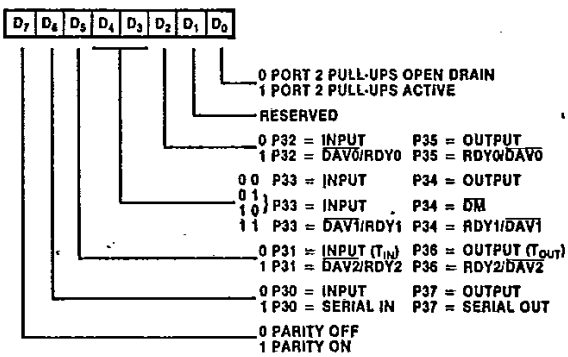
R246 P2M
Port 2 Mode Register
($F6_H$; Write Only)



R243 PRE1
Prescaler 1 Register
($F3_H$; Write Only)

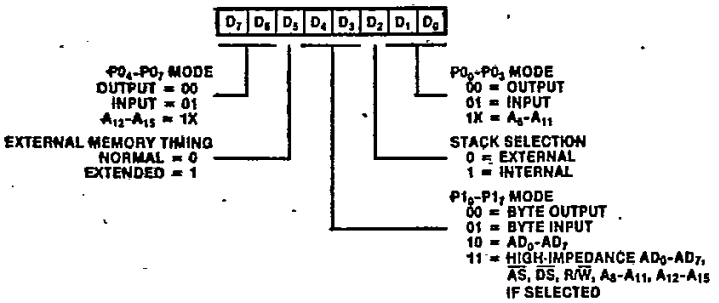


R247 P3M
Port 3 Mode Register
($F7_H$; Write Only)

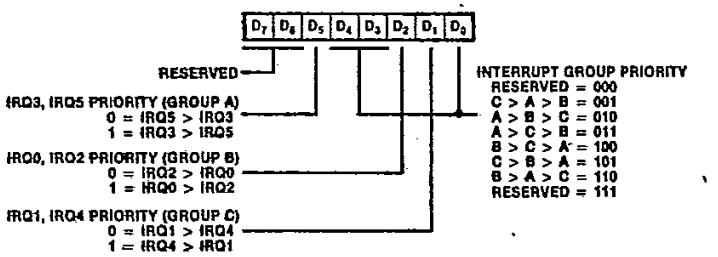


Registers
(Continued)

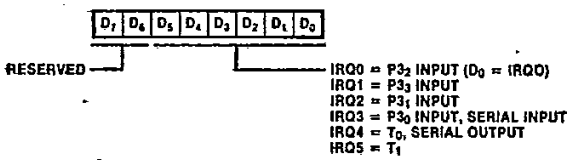
R248 P01M
Port 0 and 1 Mode Register
($F8_H$; Write Only)



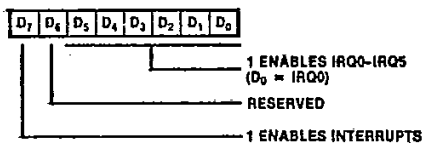
R249 IPR
Interrupt Priority Register
($F9_H$; Write Only)



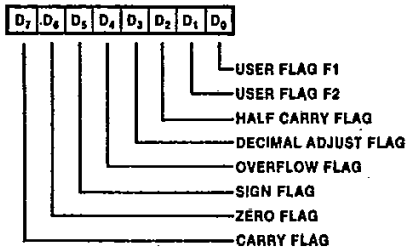
R250 IRQ
Interrupt Request Register
(FA_H; Read/Write)



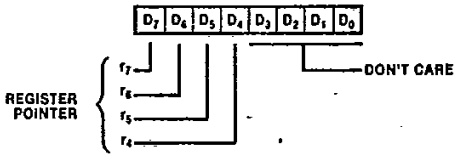
R251 IMR
Interrupt Mask Register
(FB_H; Read/Write)



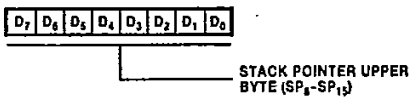
R252 FLAGS
Flag Register
(FC_H; Read/Write)



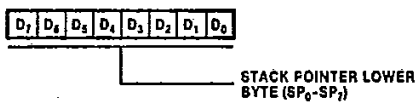
R253 RP
Register Pointer
(FD_H; Read/Write)



R254 SPH
Stack Pointer
(FE_H; Read/Write)

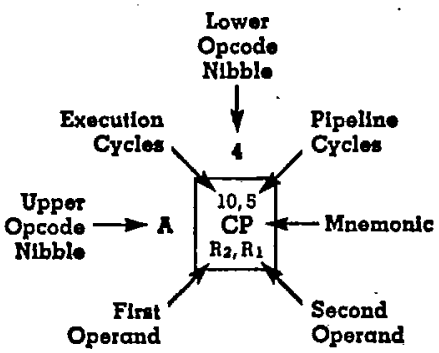


R255 SPL
Stack Pointer
(FF_H; Read/Write)



Opcode Map

Upper Nibble (Hex)	Lower Nibble (Hex) 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6,5 DEC R1	6,5 DEC IR1	6,5 ADD r1,r2	6,5 ADD r1,Ir2	10,5 ADD R2,R1	10,5 ADD IR2,R1	10,5 ADD R1,IM	10,5 ADD IR1,IM	6,5 LD r1,R2	6,5 LD r2,R1	12/10,5 DJNZ r1,RA	12/10,0 JR cc,RA	6,5 LD r1,IM	12/10,0 JP cc,DA	6,5 INC r1	
1	6,5 RLC R1	6,5 RLC IR1	6,5 ADC r1,r2	6,5 ADC r1,Ir2	10,5 ADC R2,R1	10,5 ADC IR2,R1	10,5 ADC R1,IM	10,5 ADC IR1,IM								
2	6,5 INC R1	6,5 INC IR1	6,5 SUB r1,r2	6,5 SUB r1,Ir2	10,5 SUB R2,R1	10,5 SUB IR2,R1	10,5 SUB R1,IM	10,5 SUB IR1,IM								
3	8,0 JP IRR1	6,1 SRP IM	6,5 SBC r1,r2	6,5 SBC r1,Ir2	10,5 SBC R2,R1	10,5 SBC IR2,R1	10,5 SBC R1,IM	10,5 SBC IR1,IM								
4	8,5 DA R1	6,5 DA IR1	6,5 OR r1,r2	6,5 OR r1,Ir2	10,5 OR R2,R1	10,5 OR IR2,R1	10,5 OR R1,IM	10,5 OR IR1,IM								
5	10,5 POP R1	10,5 POP IR1	6,5 AND r1,r2	6,5 AND r1,Ir2	10,5 AND R2,R1	10,5 AND IR2,R1	10,5 AND R1,IM	10,5 AND IR1,IM								
6	6,5 COM R1	6,5 COM IR1	6,5 TCM r1,r2	6,5 TCM r1,Ir2	10,5 TCM R2,R1	10,5 TCM IR2,R1	10,5 TCM R1,IM	10,5 TCM IR1,IM								
7	10/12,1 PUSH R2	12/14,1 PUSH IR2	6,5 TM r1,r2	6,5 TM r1,Ir2	10,5 TM R2,R1	10,5 TM IR2,R1	10,5 TM R1,IM	10,5 TM IR1,IM								
8	10,5 DECW RR1	10,5 DECW IR1	12,0 LDE r1,Irr2	18,0 LDEI Ir1,Irr2												6,1 DI
9	6,5 RL R1	6,5 RL IR1	12,0 LDE Irr1	18,0 LDEI Ir2,Irr1												6,1 EI
A	10,5 INCW RR1	10,5 INCW IR1	6,5 CP r1,r2	6,5 CP r1,Ir2	10,5 CP R2,R1	10,5 CP IR2,R1	10,5 CP R1,IM	10,5 CP IR1,IM								14,0 RET
B	6,5 CLR R1	6,5 CLR IR1	6,5 XOR r1,r2	6,5 XOR r1,Ir2	10,5 XOR R2,R1	10,5 XOR IR2,R1	10,5 XOR R1,IM	10,5 XOR IR1,IM								16,0 IRET
C	6,5 RRC R1	6,5 RRC IR1	12,0 LDC r1,Irr2	18,0 LDCI Ir1,Irr2				10,5 LD r1,x,R2								6,5 RCF
D	6,5 SRA R1	6,5 SRA IR1	12,0 LDC r2,Irr1	18,0 LDCI Ir2,Irr1	20,0 CALL* IRR1		20,0 CALL DA	10,5 LD r2,x,R1								6,5 SCF
E	6,5 RR R1	6,5 RR IR1		6,5 LD r1,Ir2	10,5 LD R2,R1	10,5 LD IR2,R1	10,5 LD R1,IM	10,5 LD IR1,IM								6,5 CCF
F	8,5 SWAP R1	8,5 SWAP IR1		6,5 LD Ir1,r2		10,5 LD R2,IR1										6,0 NOP
Bytes per Instruction	2				3				2					3	1	



Legend:
R = 8-Bit Address
r = 4-Bit Address
R1 or r1 = Dst Address
R2 or r2 = 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 . . . . . -0.3 V to +7.0 V

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.

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 reference pin.

Standard conditions are:

- □ $+4.75\ V \leq V_{CC} \leq +5.25\ V$
- □ GND = 0 V
- □ $0°C \leq T_A \leq +70°C$

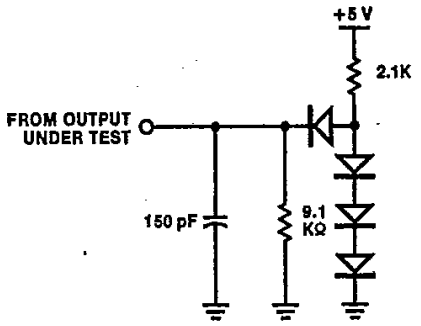


Figure 14. Test Load 1

DC Characteristics

Symbol	Parameter	Min	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	$I_{OH} = -250\ \mu A$
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = +2.0$ mA
I_{IL}	Input Leakage	-10	10	μA	$0\ V \leq V_{IN} \leq +5.25\ V$
I_{OL}	Output Leakage	-10	10	μA	$0\ V \leq V_{IN} \leq +5.25\ V$
I_{IR}	Reset Input Current		-50	μA	$V_{CC} = +5.25\ V$, $V_{RL} = 0\ V$
I_{CC}	V_{CC} Supply Current		150	mA	

External I/O or Memory Read and Write Timing

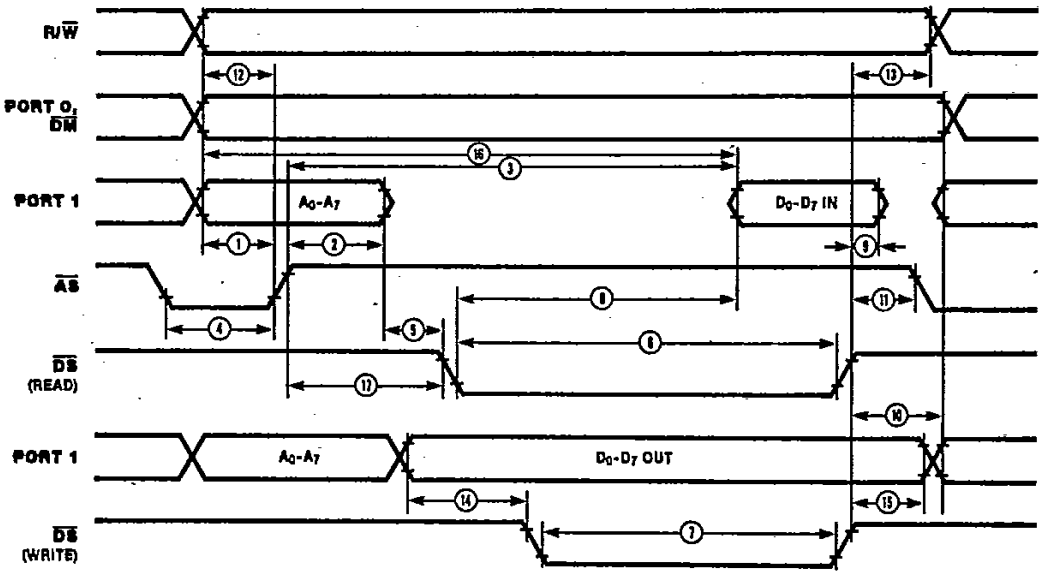


Figure 15. External I/O or Memory Read/Write

No.	Symbol	Parameter	8 MHz Min	8 MHz Max	12.5 MHz Min	12.5 MHz Max	Notes*†°
1	TdA(AS)	Address Valid to $\overline{AS}$ ↑ Delay	50		35		2,3
2	TdAS(A)	$\overline{AS}$ ↑ to Address Float Delay	60		45		2,3
3	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Required Valid		320		220	1,2,3
4	TwAS	$\overline{AS}$ Low Width	80		55		1,2,3
5	TdAz(DS)	Address Float to $\overline{DS}$ ↓	0		0		
6	TwDSR	$\overline{DS}$ (Read) Low Width	250		185		1,2,3
7	TwDSW	$\overline{DS}$ (Write) Low Width	160		110		1,2,3
8	TdDSR(DR)	$\overline{DS}$ ↓ to Read Data Required Valid		200		130	1,2,3
9	ThDR(DS)	Read Data to $\overline{DS}$ ↑ Hold Time	0		0		
10	TdDS(A)	$\overline{DS}$ ↑ to Address Active Delay	80		45		2,3
11	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		55		2.3
12	TdR/W(AS)	R/$\overline{W}$ Valid to $\overline{AS}$ ↑ Delay	50		30		2,3
13	TdDS(R/W)	$\overline{DS}$ ↑ to R/$\overline{W}$ Not Valid	60		35		2,3
14	TdDW(DSW)	Write Data Valid to $\overline{DS}$ (Write) ↓ Delay	50		35		2,3
15	TdDS(DW)	$\overline{DS}$ ↑ to Write Data Not Valid Delay	80		45		2,3
16	TdA(DR)	Address Valid to Read Data Required Valid		410		255	1,2,3
17	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	80		55		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.

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

Additional Timing Table

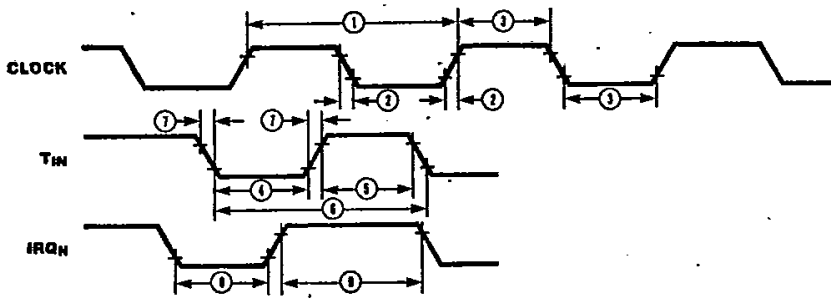


Figure 16. Additional Timing

No.	Symbol	Parameter	8 MHz Min	8 MHz Max	12.5 MHz Min	12.5 MHz Max	Notes*
1	TpC	Input Clock Period	125	1000	80	1000	1
2	TrC,TfC	Clock Input Rise And Fall Times		25		15	1
3	TwC	Input Clock Width	37		26		1
4	TwTinL	Time Input Low Width	100		70		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	100		70		2,3
8b	TwIL	Interrupt Request Input Low Time		3TpC	3TpC		2,4
9	TwIH	Interrupt Request Input High Time		3TpC	3TpC		2,3

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

Memory Port Timing

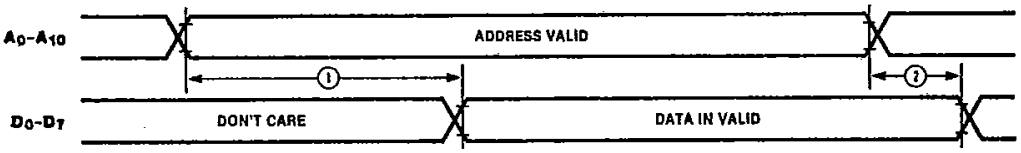


Figure 17. Memory Port Timing

No.	Symbol	Parameter	Min	Max	Notes*
1	TdA(DI)	Address Valid to Data Input Delay		320	1,2
2	ThDI(A)	Data In Hold time	0		1

NOTES:
1. Test Load 2.
2. This is a Clock-Cycle-Dependent parameter. For clock frequencies other than the maximum, use the following formula: 5 TpC – 95

*Units are nanoseconds unless otherwise specified.

Handshake Timing

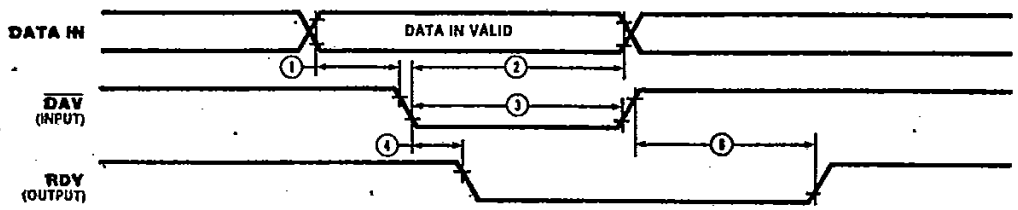


Figure 18a. Input Handshake

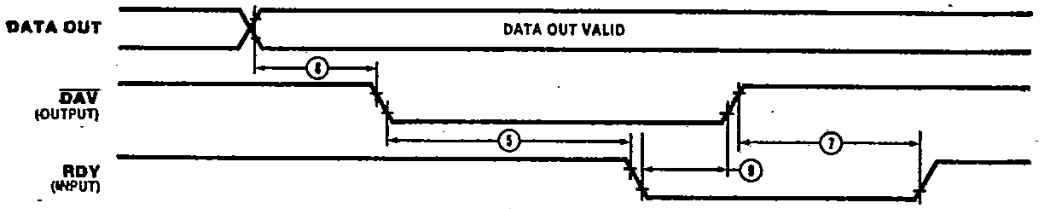


Figure 18b. Output Handshake

No.	Symbol	Parameter	Min	Max	Notes*
1	TsDI(DAV)	Data In Setup Time	0		
2	ThDI(DAV)	Data In Hold time	160		
3	TwDAV	Data Available Width	120		
4	TdDAVIf(RDY)	$\overline{DAV}$ ↓ Input to RDY ↓ Delay		120	1,2
5	TdDAVOf(RDY)	$\overline{DAV}$ ↓ Output to RDY ↓ Delay	0		1,3
6	TdDAVIr(RDY)	$\overline{DAV}$ ↑ Input to RDY ↑ Delay		120	1,2
7	TdDAV0r(RDY)	$\overline{DAV}$ ↑ Output to RDY ↑ Delay	0		1,3
8	TdDO(DAV)	Data Out to $\overline{DAV}$ ↓ Delay	30		1
9	TdRDY(DAV)	Rdy ↓ Input to $\overline{DAV}$ ↑ Delay	0	140	1

NOTES:
1. Test load 1
2. Input handshake
3. Output handshake

† All timing references use 2.0 V for a logic "1" and 0.8 V for a logic "0".

* Units in nanoseconds (ns).

Clock-Cycle-Time-Dependent Characteristics

Number	Symbol	Equation
1	TdA(AS)	TpC-50
2	TdAS(A)	TpC-40
3	TdAS(DR)	4TpC-110*
4	TwAS	TpC-30
5	TwDSR	3TpC-65*
7	TwDSW	2TpC-55*
8	TdDSR(DR)	3TpC-120*
10	Td(DS)A	TpC-40
11	TdDS(AS)	TpC-30
12	TdR/W(AS)	TpC-55
13	TdDS(R/W)	TpC-50
14	TdDW(DSW)	TpC-50
15	TdDS(DW)	TpC-40
16	TdA(DR)	5TpC-160*
17	TdAS(DS)	TpC-30

*Add 2TpC when using extended memory timing.