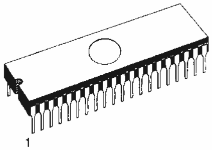
SGS-THOMSON MICROELECTRONICS

Z86E11

4K EPROM, Z8 FAMILY MICROCOMPUTER

- COMPLETE MICROCOMPUTER
- 4K BYTES OF EPROM
- 128 BYTES OF RAM
- 32 I/O LINES
- UP TO 60K BYTES ADDRESSABLE EXTERNAL SPACE EACH FOR PROGRAM AND DATA MEMORY
- FULLY COMPATIBLE WITH STANDARD ROM VERSION
- 144-BYTE REGISTER FILE, INCLUDING 124 GENERAL PURPOSE REGISTERS, 4 I/O PORT REGISTERS, AND 16 STATUS AND CONTROL REGISTERS
- MINIMUM INSTRUCTION EXECUTION TIME 1µs, AT 12MHz
- 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 1.5µs (8MHz)
- ON-CHIP OSCILLATOR WHICH ACCEPTS CRYSTAL OR EXTERNAL CLOCK DRIVE
- SINGLE + 5V POWER SUPPLY ALL PINS TTL COMPATIBLE
- THREE EPROM PROGRAMMING MODES:
- EPROM-LIKE, USING A STANDARD EPROM PROGRAMMER
- SELF PROGRAMMING DURING NORMAL PROGRAM EXECUTION
- AN OFF-CHIP EPROM PROVIDES A PROGRAM/VERIFY FACILITY TO ALLOW A SIMPLE AND TIME-EFFICIENT AUTO LOADING OPERATION
- INTEGRATED PROGRAMMABLE PROTECTIONS AVOID EPROM CONTENT READOUT
- AVAILABLE IN 8MHz AND 12MHz VERSIONS



FDIP40W

(Ordering Information at the end of the datasheet)

Figure 1 : Logic Functions

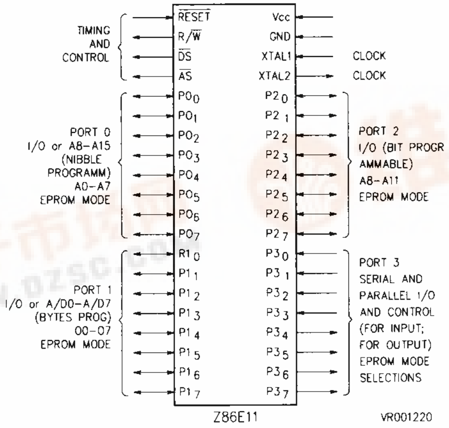
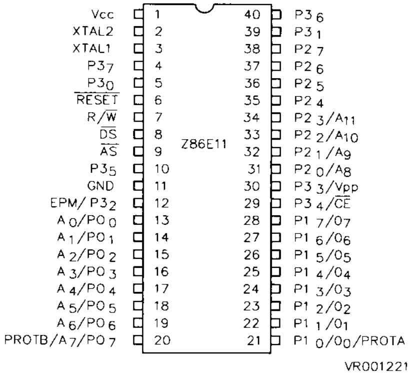

TIMING AND CONTROL
RESET
R/W
DS
AS
Vcc
GND
XTAL1
XTAL2
CLOCK
CLOCK
PORT 0 I/O or A8-A15 (NIBBLE PROGRAMM) A0-A7 EPROM MODE
P0₀ P0₁ P0₂ P0₃ P0₄ P0₅ P0₆ P0₇
P2₀ P2₁ P2₂ P2₃ P2₄ P2₅ P2₆ P2₇
PORT 2 I/O (BIT PROGRAMMABLE) A8-A11 EPROM MODE
PORT 1 I/O or A/D0-A/D7 (BYTES PROG) D0-D7 EPROM MODE
P1₀ P1₁ P1₂ P1₃ P1₄ P1₅ P1₆ P1₇
P3₀ P3₁ P3₂ P3₃ P3₄ P3₅ P3₆ P3₇
PORT 3 SERIAL AND PARALLEL I/O AND CONTROL (FOR INPUT; FOR OUTPUT) EPROM MODE SELECTIONS
Z86E11
VR001220

Figure 2 : 40 Pins DIL Configuration.



GENERAL DESCRIPTION

The Z86E11 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 4K bytes of internal EPROM, 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. In all configurations, a large number of pins remain available for I/O.

The 4Kx8 on-board EPROM can be programmed in three modes, Eprom-like, Self programming and Autoloading. In Eprom-like, the programming procedure is similar to that for a M2732 (using an interface board), with the only exception being for the programming voltage which must be 12.5V related to the SGS-THOMSON NMOS-E3 used technology. Self-programming permits byte-programming during normal microcomputer program execution.

An important facility is the programmable readout protections which allows the user to inhibit external access to proprietary program code by programming 2 non-volatile transistors. These locks can be reset only by erasing the entire EPROM array.
For its characteristics, the Z86E11 can be considered as a low cost development tool for the Z8 microcomputer family.

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/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, 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 120K bytes of external memory (figure 4).

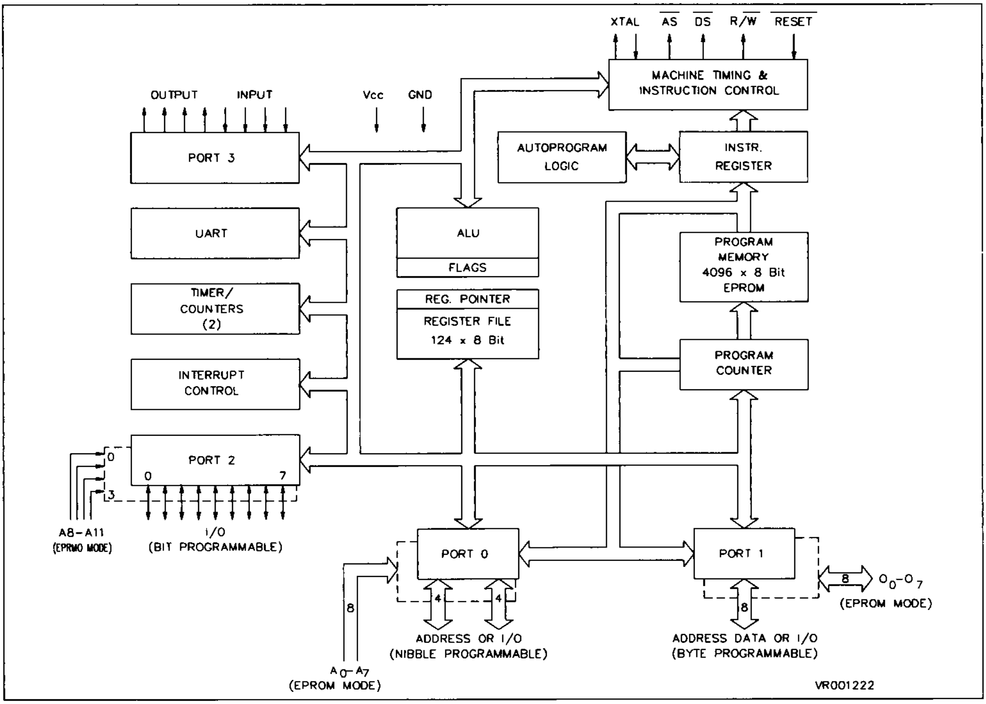
Three basic address spaces are available to support this wide range of configurations : program memory (internal and external), data memory (external) and the register file (internal). The 144-byte random-access register file is composed of 124 general-purpose registers, four I/O port registers, and 16 control and status registers.

To unburden the program from copying 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.

ARCHITECTURE (Continued)

Figure 3 : Block Diagram



PIN DESCRIPTIONS

P0$_0$-P0$_7$. *I/O Port Lines* (input/outputs, TTL compatible). 8 lines nibble-programmable that can be configured under program control for I/O or external memory interface.

P1$_0$-P1$_7$. *I/O Port Lines* (input/outputs, TTL compatible). 8 lines byte programmable that can be configured under program control for I/O or multiplexed address (A$_0$-A$_7$) and data (D$_0$-D$_7$) lines used to interface with program/data memory.

P2$_0$-P2$_7$. *I/O Port Lines* (input/outputs, TTL compatible). 8 lines bit programmable. In addition they can be configured to provide open-drain output.

P3$_0$-P3$_7$. *I/O Port Lines* (TTL compatible) 4 lines input (P3$_0$-P3$_3$), 4 lines output P3$_4$-P3$_7$). They can also be configured as control lines.

$\overline{AS}$. *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.

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ initializes the Z86xx. When $\overline{RESET}$ is deactivated, program execution begins from internal program location 000CH.

R/$\overline{W}$. *Read/Write* (output). R/$\overline{W}$ is Low when the Z86xx 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 (8 or 12MHz maximum) or an external single-phase clock (8 or 12MHz maximum) to the on-chip clock oscillator and buffer.

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 4096 bytes consist of on-chip EPROM. At addresses 4096 and greater, the Z86E11 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 Z86E11 can address 60K bytes of external data memory beginning at locations 4096 (figure 5). External data memory may be included with or separated from the external program memory space.

$\overline{\text{DM}}$, 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.

Z86E11 instructions can access registers directly or indirectly with an 8-bit address field. The Z86E11 also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In this case, the register file is divided into nine working-register groups, each occupying 16 contiguous locations (figure 7). 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 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.

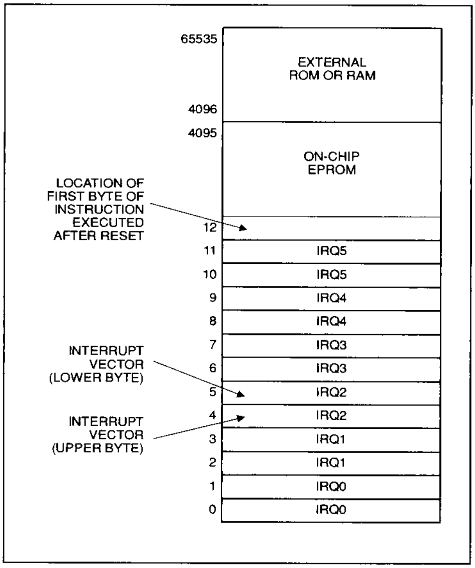
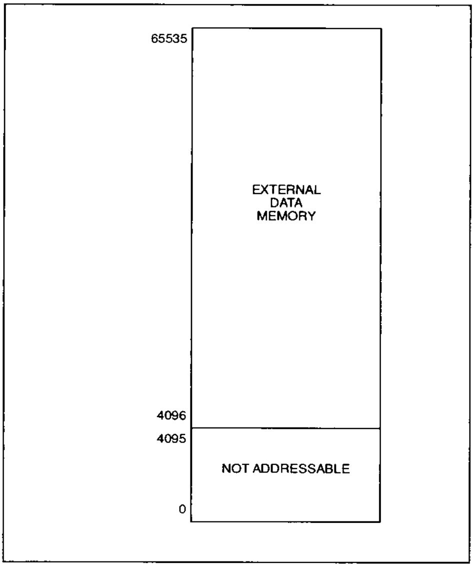

65535
EXTERNAL ROM OR RAM
4096
4095
ON-CHIP EPROM
LOCATION OF FIRST BYTE OF INSTRUCTION EXECUTED AFTER RESET
12
11
IRQ5
10
IRQ5
9
IRQ4
8
IRQ4
7
IRQ3
6
IRQ3
5
IRQ2
4
IRQ2
3
IRQ1
2
IRQ1
1
IRQ0
0
IRQ0
INTERRUPT VECTOR (LOWER BYTE)
INTERRUPT VECTOR (UPPER BYTE)

Figure 5 : Data Memory Map.

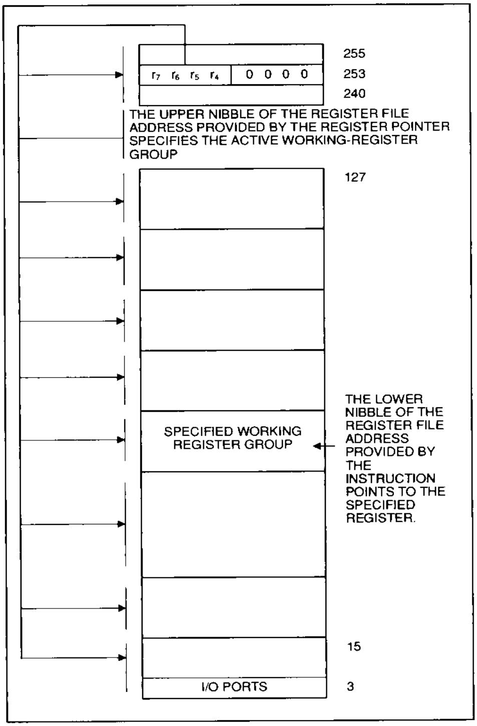


ADDRESS SPACES (Continued)

Figure 6 : Register File.

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	PORT 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
	NOT IMPLEMENTED	
127	GENERAL-PURPOSE REGISTERS	
4		
3	PORT 3	P3
2	PORT 2	P2
1		
0	PORT 0	P0

Figure 7 : Register Pointer.


255
r7 r6 r5 r4 0 0 0 0
253
240
THE UPPER NIBBLE OF THE REGISTER FILE ADDRESS PROVIDED BY THE REGISTER POINTER SPECIFIES THE ACTIVE WORKING-REGISTER GROUP
127
SPECIFIED WORKING REGISTER GROUP
THE LOWER NIBBLE OF THE REGISTER FILE ADDRESS PROVIDED BY THE INSTRUCTION POINTS TO THE SPECIFIED REGISTER.
15
I/O PORTS
3

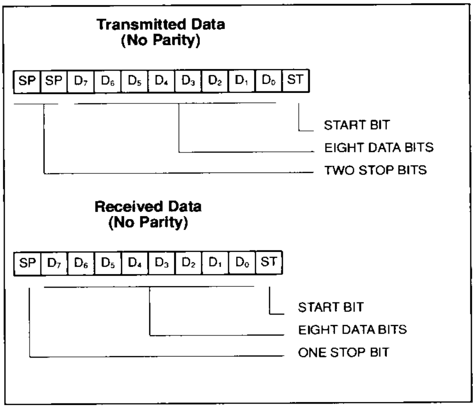
SERIAL INPUT/OUTPUT

Port 3 lines P30 and P37 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 the 8MHz version.

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 (IRQ4) is generated on all transmitted characters.

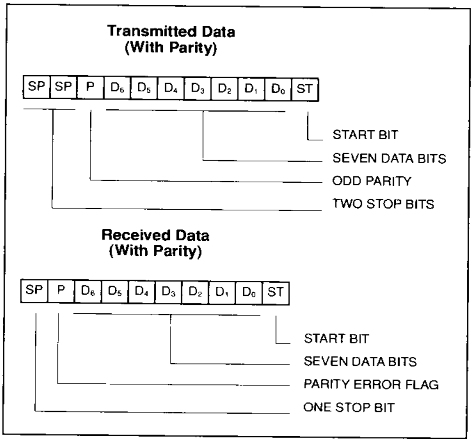
Received data must have a start bit, eight data bits and at least one stop bit. If parity is on, bit 7 of the received data is replaced by a parity error flag. Received characters generate the IRQ3 interrupt request.

Figure 8 : Serial Data Formats.


Transmitted Data (No Parity)
SP SP D7 D6 D5 D4 D3 D2 D1 D0 ST
START BIT
EIGHT DATA BITS
TWO STOP BITS
Received Data (No Parity)
SP D7 D6 D5 D4 D3 D2 D1 D0 ST
START BIT
EIGHT DATA BITS
ONE STOP BIT

SERIAL INPUT/OUTPUT (Continued)

Figure 9 : Serial Data Formats (Continued)



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 prescaler 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, 4MHz maximum for the 8MHz device and 6MHz maximum for the 12MHz device, divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input as an external clock (1.5MHz 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

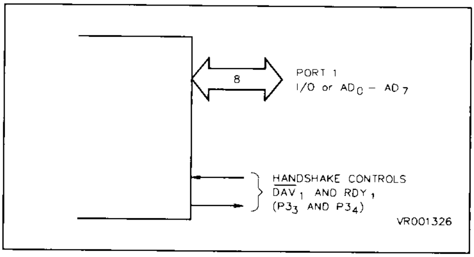
The Z8 has 32 lines dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable as input, output or address/data. Under software control, the ports can be programmed to provide address outputs, timing, status signals, serial I/O, and parallel I/O with 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 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 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 10 : Port 1 Configuration.

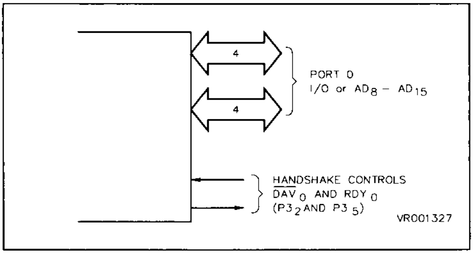


Port 0 can be programmed as a nibble I/O port, or as an address port for interfacing external memory. When used as an I/O port, Port 0 may be placed under handshake control. In this configuration, Port 3 lines $P3_2$ and $P3_5$ are used as the handshake controls DAV_0 and RDY_0. Handshake signal assignment is dictated by the I/O direction of the upper nibble $P0_4$-$P0_7$.

I/O PORTS (Continued)

For external memory references, Port 0 can provide address bits A_8-A_{11} (lower nibble) or A_{12}-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/$\overline{W}$.

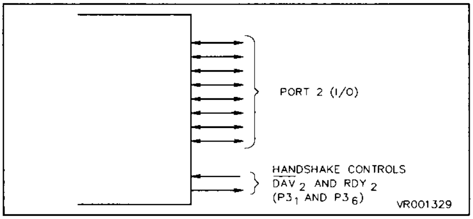
Figure 11 : Port O Configuration.



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 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.

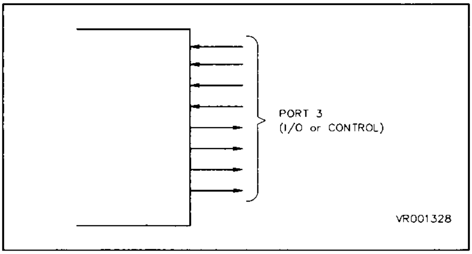
Figure 12 : Port 2 Configuration.



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 13 : Port 3 Configuration.



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

- AT cut, series resonant
- Fundamental types, 8MHz and 12MHz
- Series resistance, $R_S \leq 100\Omega$.

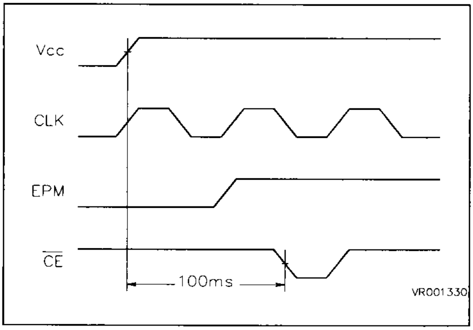
EPROM MODE

Eprom-Like Programming. In this mode, the microcomputer memory is programmed, using a standard EPROM programmer, with the same procedure as for our M2732 (32K bits EPROM). This made possible by the following Z86E11 configuration, where $P1_0$-$P1_7$ are used as 8-bit I/O data (O_0-O_7), $P0_0$-$P0_7$ and $P2_0$-$P2_3$ are used as 12-bit Addresses (A_0-A_{11}); the microcomputer must be in Reset state, forcing the related pin to GND, and the Clock must be active for the complete operation.

Three other pins are available for that purpose: the EPM pin on port $P3_2$, which allows the microcomputer to recognize the Eprom-Like condition when a high voltage (≥ 7V) is applied; the V_{PP} pin on port $P3_3$, which is used to furnish programming voltage fixed at 12.5V ± 300mV; and the $\overline{CE}$ pin on port $P3_4$, which is used to perform program enable/verify.

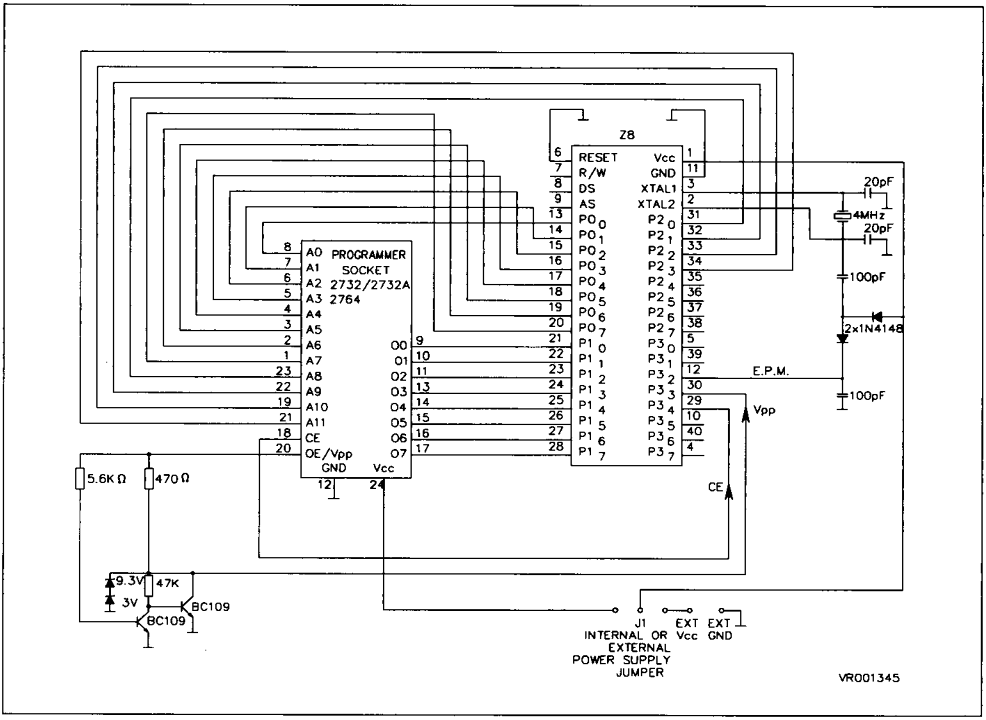
For a correct microcomputer set-up, the V_{CC} must be applied at least 100ms before the programming procedure starts, as shown in Figure 14.

Figure 14 : Set-up Waveforms



A simple interface board, described in Figure 15, allows programming to be carried out through use of a standard EPROM-programmer.

Figure 15 : EPROM Programmer Interface Board



EPROM MODE (Continued)

This board can be used with a general purpose EPROM programmer without doing particular controls (power consumption, etc.) before starting the programming procedure.

This means that the board cannot be used with DATA I/O programmers. In fact it implements the above-mentioned controls, giving an error message and stopping the programming procedure when, for example, the power consumption of the Z86E11 is tested. This occurs because the MCU's power consumption is greater then the consumption of an M2732, due to the on-chip oscillator running on the microcomputer.

To solve this problem, an agreement with DATA I/O has been made with the purpose of designing a special cartridge for the DATA I/O 29B model, which use on-board the UNIPAK 2B firmware.

This cartridge, which is able to program the Z86E11 plus its relevant protection, is directly sold by DATA I/O with the order code 351B-Z8.

Self-Programming. This mode permits programming one byte of the on-chip EPROM during normal microcomputer program execution. The instruction to be used is the Load Constant LDC @RR0,R2 (operating code D2H).

This instruction allows the standard Z8 to load the content of the working register R2 into an external RAM memory allocated in the program memory space, (the address of the memory location is pointed by the pair of working registers R0 and R1). The Z86E11 uses this instruction also to program the 4K bytes on-chip EPROM.

Addressing one of the on-chip EPROM bytes, using this instruction, the programming operation takes place when an high voltage level on the VPP pin (12.5V ± 300mV) is applied.

In this case, both the address and the data memory are stored internally for the necessary programming time, where the time is defined by the execution of 1024 NOP operations (1 NOP operation = 12 external clock pulses). The programming time is contained between 1ms (12MHz external clock) and 12ms (1MHz external clock).

As just mentioned, during this time, the CPU is automatically internally forced to execute NOP instructions (operating code FFH), while a RET instruction (operating code AFH) is automatically executed at the end of programming.

For a correct program restart is necessary to save the address of the Load Constant (LDC) next instruction in the Stack. This can be done by loading into the Stack the return address calling a Subroutine as below, where, to permit a correct return to the main program, it is necessary to disable the interrupt before LDC execution.

```
          "
          , ,
          DI
          CALL WRITE
          EI
          "
          "
    WRITE LDC @RR0,R2
          RET (automatically executed)
          "
          , ,
```

Autoloading/Verify Facility. The most flexible way for on-chip EPROM programming is, as we know, the use of a standard EPROM programmer, selecting the Eprom-like facility, and using an appropriate interface board (Figure 15).

If, however, the planned operation is only a particular memory loading into the on-chip EPROM, it is possible to perform this operation in a much simpler way, using a board which allows the Z86E11 to read and load the renamed particular memory, using the autoloading procedure. Figure 16 shows the autoloading program flow-chart.

The software required for this operation is stored in an external Test-Memory, present in the auto-loading/verify board (IC4, Figure 17) and is listed in Table 1.

When the microcomputer is forced in Test mode by applying a high voltage level (≥ 7V) on the $\overline{\text{RESET}}$ pin, ports P0 and P1 are configured as address/data to access the external memory, only if either the values 00 or 01 are forced on port P2.

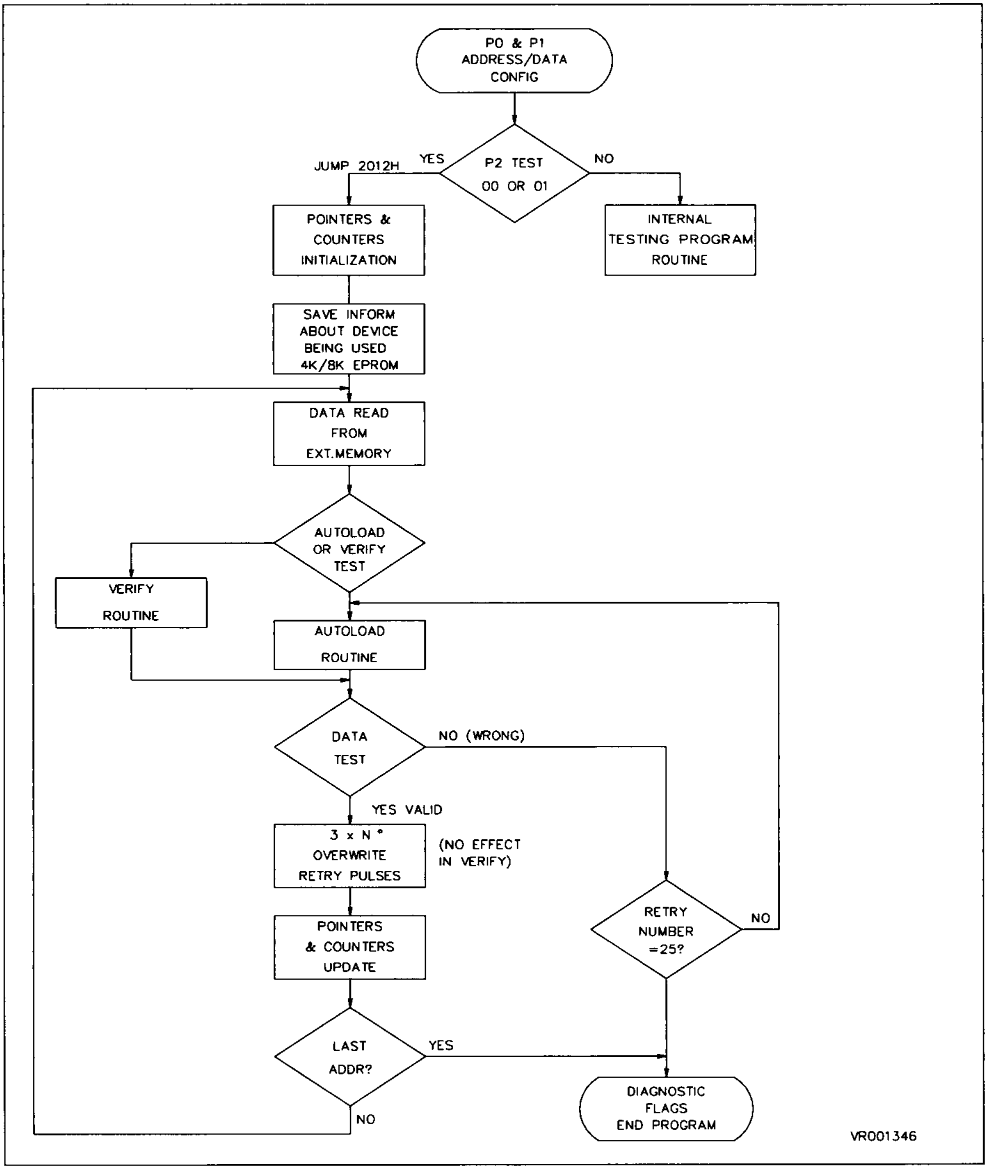
At this point, a test on port P2 is executed to decide if the on-chip EPROM Autolading or verify facility has to be executed.

Consequently, registers required for the operation are initialized, data to be compared or stored is read, and the routine chosen is executed.

The autoloading routine is an intelligent program which executes a number of overwriting cycles equal to three times the number of programmming cycles required to perform a correct byte programming (up to a maximum of 25). Thus, the on-chip 4K EPROM programming time is optimized and equal to 25 seconds with an 8MHz external clock.

EPROM MODE (Continued)

Figure 16 : Autoloading Flow Chart



EPROM MODE (Continued)

Table 1. Test Memory Exadecimal Code

Addresses	Value															
0000 to 000F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0010 to 001F	FF	FF	FC	0E	E7	EF	00	FA	FB	4C	C0	8C	20	98	E2	46
0020 to 002F	E9	40	E6	FF	3C	D6	20	A0	8B	26	FF	FF	FF	FF	FF	FF
0030 to 003F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0040 to 004F	AF	D2	A6	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	98	E2
0050 to 005F	CC	19	C9	FF	C2	A4	B0	EF	D4	E8	06	EF	03	C2	B6	A2
0060 to 006F	AB	6B	06	CA	F3	3C	20	8B	0C	D4	E8	FA	FC	A0	E6	A0
0070 to 007F	E4	8B	2E	3C	80	8B	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF
0080 to 008F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0090 to 009F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00A0 to 00AF	AF	A6	3A	20	6B	06	69	FC	CB	A6	8B	C7	69	FC	DB	A0
00B0 to 00BF	8B	C1	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00C0 to 00CF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00D0 to 00DF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00E0 to 00EF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00F0 to 00FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Note: All the other locations of the device were the test-memory is mapped (e.g. 2732 EPROM), between 100H and FFFH, are blank and contain the exadecimal value FF.

The verify routine is simply a byte-byte comparison between the external memory and the on-chip EPROM.

A possible failure, whether in Autoloading or Verify, produces a high logical level forced on $P3_5$. Similarly, when the operation is finished, the positive conclusion is underlined, bringing $P3_7$ high.

An autoloading/Verify board diagram is shown in Figure 17, where the V_{PP} line control is necessary to not allow high voltage into the device when it has not yet been supplied. Through this board is also possible to activate memory readout protection facility as explained below.

Memory Readout Protection. The protections, once activated, block reading memory content. Such reading can be carried out in two ways:

1. Entering Eprom-Like Mode, using the Verify facility.
2. Entering Test Mode you can execute an external memory program which allows the on-chip EPROM reading through LDC instruction execution.

Programming the first protection bit blocks reading in Eprom-Like Mode (PROTA on port $P1_0$). Another protection bit (PROTB on port $P0_7$) can be activated when the Z86E11 is in external memory configuration.

This protection prevents software manipulation of the external memory by someone who decides to read the on-chip EPROM content to have a complete understanding of the user application board.

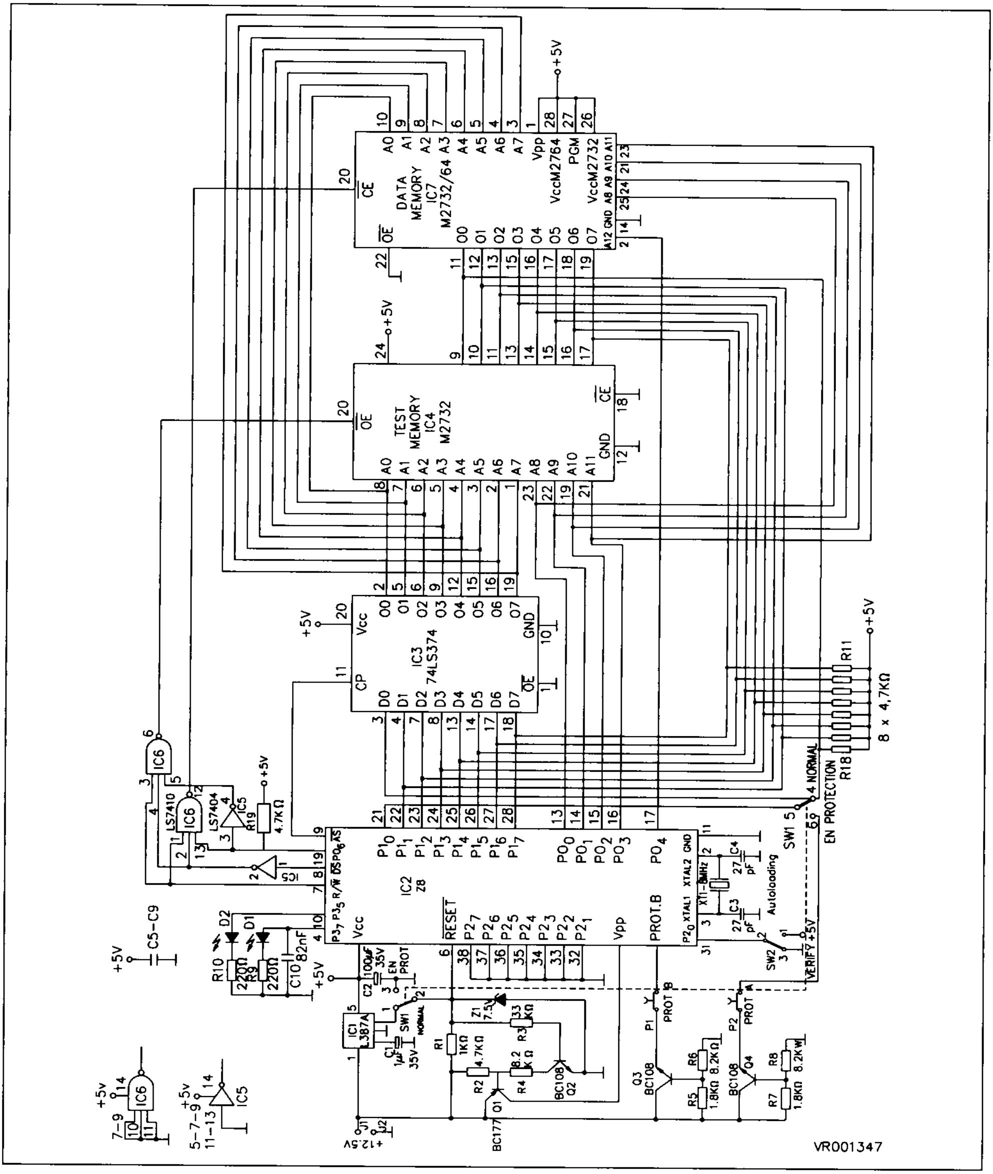
When the Z86E11 works in the external memory facility, the ports P0 and P1 are configurated as an address/data bus, so that the external memory instructions can be executed during normal microcomputer operation. These instructions can be also an appropriate routine capable of pulling out all the on-chip memory content, using the LDC instruction.

When the protection PROTB is activated, any reading attempt of the internal memory content, using the LDC instruction, is becomes useless because the data out will be always "FFH".

Consequently this protection activation inhibits the LDC instruction execution from external to internal memory.

EPROM MODE (Continued)

Figure 17 : Autoloading/Verify and Readout Protection Activation Board



EPROM MODE (Continued)

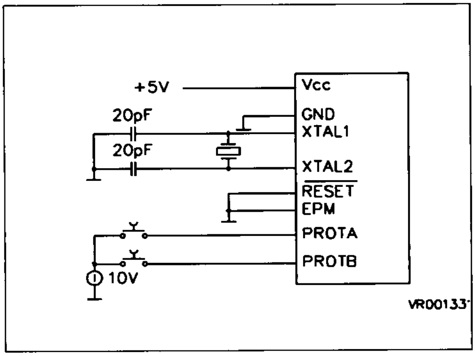
The protections are activated by programming 2 non-volatile transistors simply forcing 10V for a time more than 100ms on the desired pin, on condition that the microcomputer is in Reset state, the Clock signal is present and the EPM pin is not set.

If a complete protection is desired, both protections must be programmed.

A simple diagram for readout protection activation is shown in Figure 18.

To disable protections, a device exposure to short-wave ultraviolet light is required. In this way, also the on-chip EPROM content is erased. The EPROM array needs to be completely erased before its reprogramming and protection enabling.

Figure 18 : Readout Protection Activation Diagram



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 adress only
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	**F1**	user flag
H	Half-carry flag	**F2**	user flag

b7							b0
C	Z	S	V	D	H	F2	F1

Affected flags are indicated by :

O	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 19 : One-Byte Instruction Format.

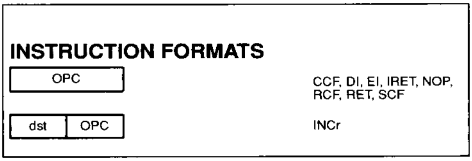


Figure 20 : Two-bytes instruction Format.

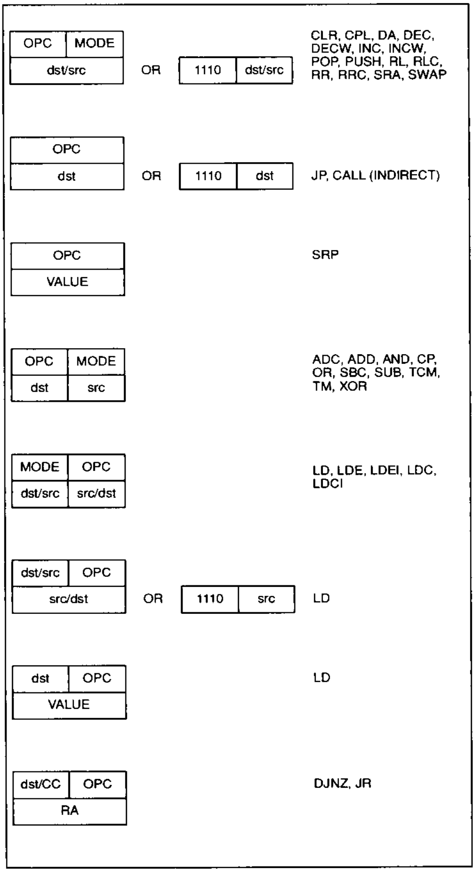
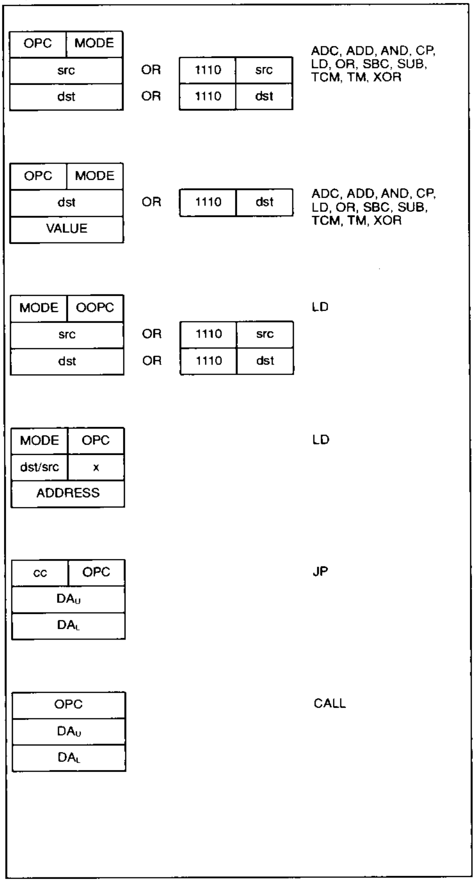
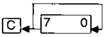
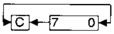
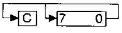
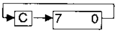


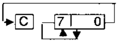
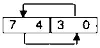
Figure 21 : Three-Bytes Instruction Format.



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

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

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 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.

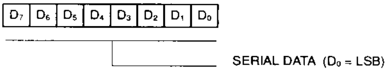
Instruction and Operation	Addr Mode		Opcode Byte (Hex)	Flags Affected					
	dst	src		C	Z	S	V	D	H
SCF C ← 1			DF	1	-	-	-	-	-
SRA dst 	R IR		D0 D1	*	*	*	0	-	-
SRP src RP ← src		Im	31	-	-	-	-	-	-
SUB dst, src dst ← dst - src	(Note 1)		2	*	*	*	*	1	*
SWAP dst 	R IR		F0 F1	X	*	*	X	-	-
TCM dst, src (NOT dst) 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	-	-

Addr Mode		Lower Opcode Nibble
dst	src	
r	r	$\boxed{2}$
r	Ir	$\boxed{3}$
R	R	$\boxed{4}$
R	IR	$\boxed{5}$
R	IM	$\boxed{6}$

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 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.

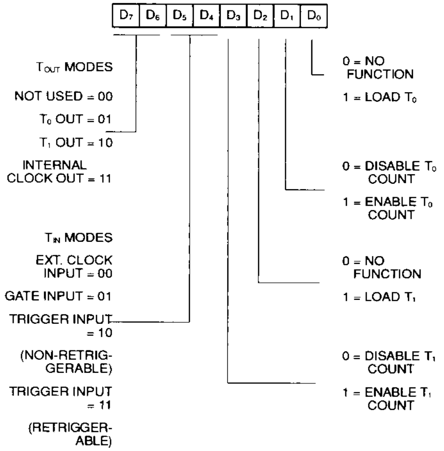
CONTROL REGISTERS

R240 SIO
Serial I/O Register
(F0H ; Read/Write)



SERIAL DATA (D_0 = LSB)

R241 TMR
Timer Mode Register
(F1H ; Read/Write)



T_{OUT} MODES
NOT USED = 00
T_0 OUT = 01
T_1 OUT = 10
INTERNAL CLOCK OUT = 11

T_{IN} MODES
EXT. CLOCK INPUT = 00
GATE INPUT = 01
TRIGGER INPUT = 10
(NON-RETRIGGERABLE)
TRIGGER INPUT = 11
(RETRIGGERABLE)

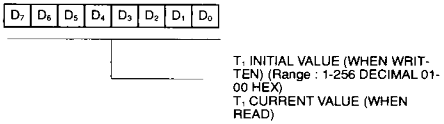
0 = NO FUNCTION
1 = LOAD T_0

0 = DISABLE T_0 COUNT
1 = ENABLE T_0 COUNT

0 = NO FUNCTION
1 = LOAD T_1

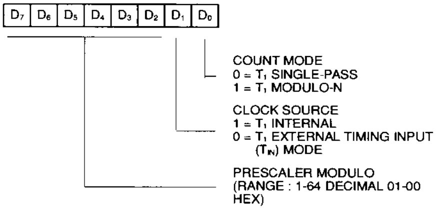
0 = DISABLE T_1 COUNT
1 = ENABLE T_1 COUNT

R242 T1
Counter Timer 1 Register
(F2H ; Read/Write)



T_1 INITIAL VALUE (WHEN WRITTEN) (Range : 1-256 DECIMAL 01-00 HEX)
T_1 CURRENT VALUE (WHEN READ)

R243 PRE1
Prescaler 1 Register
(F3H ; Write Only)

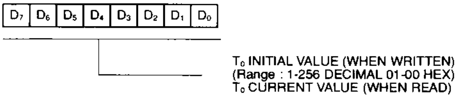


COUNT MODE
0 = T_1 SINGLE-PASS
1 = T_1 MODULO-N

CLOCK SOURCE
1 = T_1 INTERNAL
0 = T_1 EXTERNAL TIMING INPUT (T_{IN}) MODE

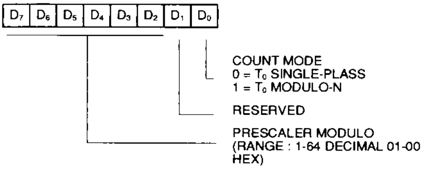
PRESCALER MODULO
(RANGE : 1-64 DECIMAL 01-00 HEX)

R244 T0
Counter/Timer 0 Register
(F4H ; Read/Write)



T_0 INITIAL VALUE (WHEN WRITTEN)
(Range : 1-256 DECIMAL 01-00 HEX)
T_0 CURRENT VALUE (WHEN READ)

R245 PRE0
Prescaler 0 Register
(F5H ; Write Only)

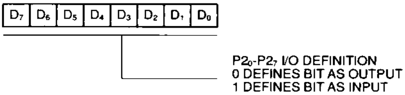


COUNT MODE
0 = T_0 SINGLE-PLASS
1 = T_0 MODULO-N

RESERVED

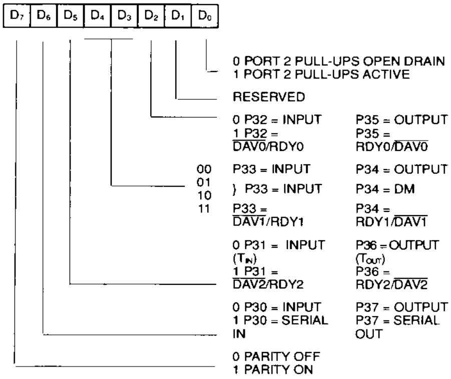
PRESCALER MODULO
(RANGE : 1-64 DECIMAL 01-00 HEX)

R246 P2M
Port 2 Mode Register
(F6H ; Write Only)



$P2_0$-$P2_7$ I/O DEFINITION
0 DEFINES BIT AS OUTPUT
1 DEFINES BIT AS INPUT

R247 P3M
Port 3 Mode Register
(F7H ; Write Only)



0 PORT 2 PULL-UPS OPEN DRAIN
1 PORT 2 PULL-UPS ACTIVE

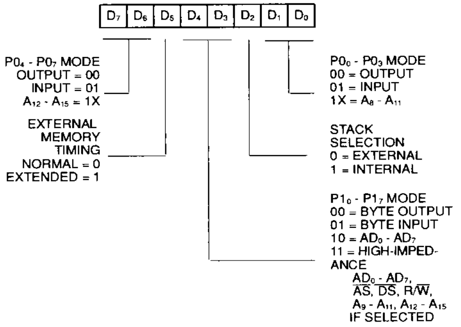
RESERVED

	0 P32 = INPUT	P35 = OUTPUT
	1 P32 = $\overline{DAV0}$/RDY0	P35 = RDY0/$\overline{DAV0}$
00	P33 = INPUT	P34 = OUTPUT
01 10	} P33 = INPUT	P34 = DM
11	P33 = $\overline{DAV1}$/RDY1	P34 = RDY1/$\overline{DAV1}$
	0 P31 = INPUT (T_{IN})	P36 = OUTPUT (T_{OUT})
	1 P31 = $\overline{DAV2}$/RDY2	P36 = RDY2/$\overline{DAV2}$
	0 P30 = INPUT	P37 = OUTPUT
	1 P30 = SERIAL IN	P37 = SERIAL OUT

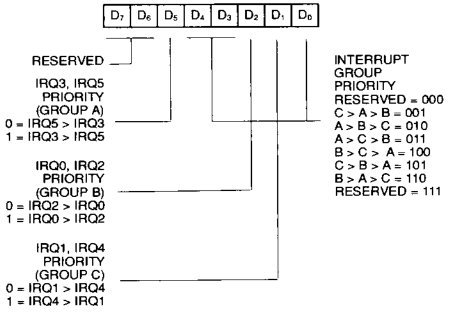
0 PARITY OFF
1 PARITY ON

CONTROL REGISTERS (Continued)

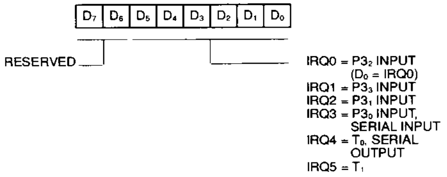
R248 P01M
Port 0 and 1 Mode Register
($F8_H$; Write Only)



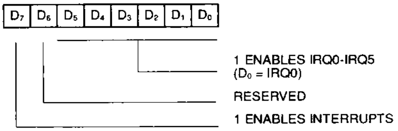
R249 IPR
Interrupt Priority Register
($F9_H$; Write Only)



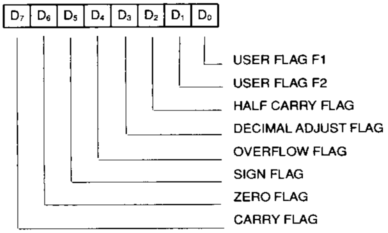
R250 IRQ
Interrupt Request Register
(FA_H ; Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0
RESERVED
IRQ0 = $P3_2$ INPUT (D_0 = IRQ0)
IRQ1 = $P3_3$ INPUT
IRQ2 = $P3_1$ INPUT
IRQ3 = $P3_0$ INPUT, SERIAL INPUT
IRQ4 = T_0, SERIAL OUTPUT
IRQ5 = T_1

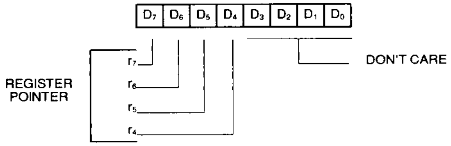
R251 IMR
Interrupt Mask Register
(FB_D ; Read/Write)



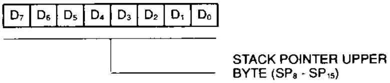
R252 FLAGS
Flag Register
(FC_H ; Read/Write)



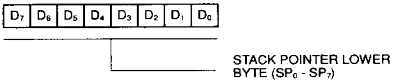
R253 RP
Register Pointer
(FD_H ; Read/Write)



R254 SPH
Stack Pointer
(FE_H ; Read/Write)



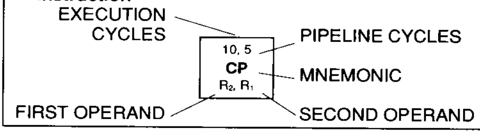
R255 SPL
Stack Pointer
(FF_H ; Read/Write)



OPCODE MAP

Lower Nibble (Hex)

Upper Nibble (Hex)	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** r_1	
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	↓	↓	↓	↓	↓	↓	↓	
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	↓	↓	↓	↓	↓	↓	↓	
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	6, 7 **SWAP** R_1	6, 7 **SWAP** IR_1		6, 5 **LD** Ir_1, r_2		10, 5 **LD** R_2, IR_1			↓	↓	↓	↓	↓	↓	↓	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.

Note : * 2-byte instruction fetch cycle appears as a 3-byte instruction.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Value	unit
	Voltage on any pin relative to ground	-0.3 to 7.0	V
T_A	Operating Ambient Temperature	0 to 70	°C
T_{STG}	Storage Temperature	-65 to 150	°C

Note : Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions 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.

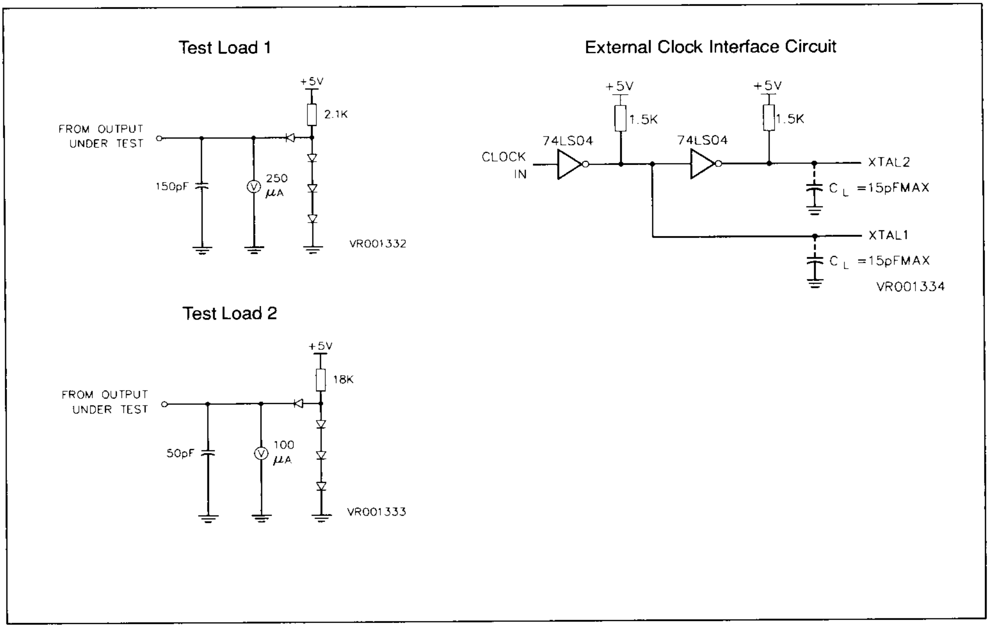
TEST CONDITIONS

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

Standard test conditions are as follows :

- $+4.5 \le V_{CC} \le +5.5V$
- GND = 0V
- $0°C \le T_A \le +70°C$

Figure 22 : Test Circuits



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}	nput 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}= -250µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0mA
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}	V_{CC} Supply Current		180	mA	
I_{PP}	V_{PP} Supply Current		30	mA	$\overline{CE}$ = V_{IL}
V_{PP}	EPROM Programming Voltage	12.2	12.8	V	

Notes:

1. V_{CC} must be applied simultaneously with or before V_{PP} and removed simultaneously with or after V_{PP}. The EPROM must not be inserted into or removed from a board with V_{PP} activated or damage may occur to the device.
2. The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 13V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding its maximum specification.

EXTERNAL I/O OR MEMORY, READ WRITE AND CLOCK CYCLE TIMING

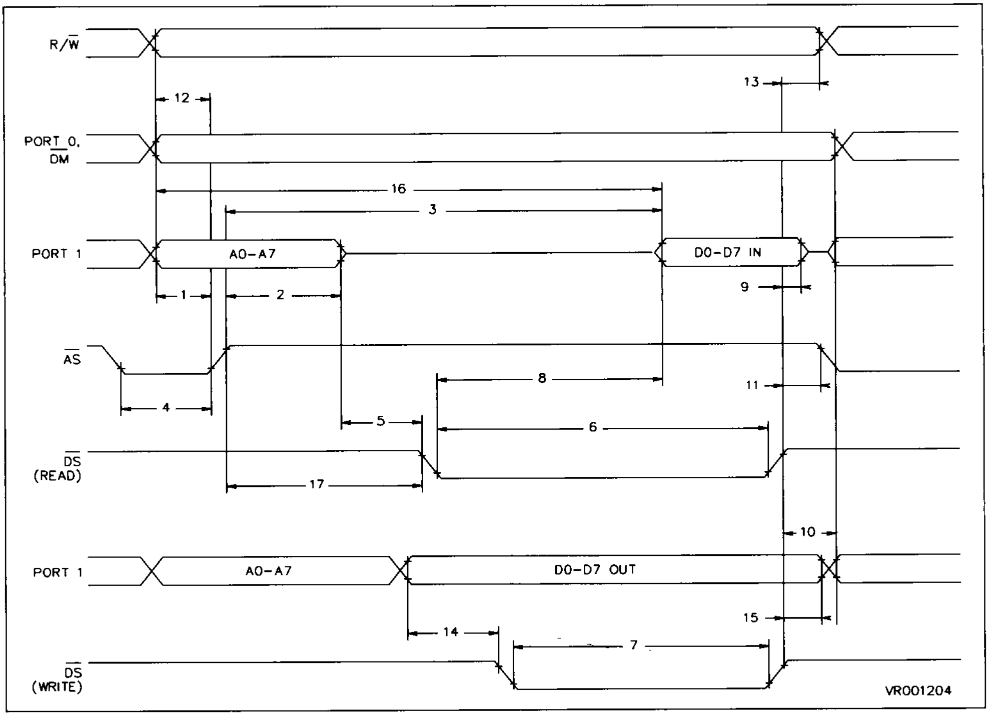
N°	Symbol	Parameter	Z86E11/8MHz			Z86E11/12MHz			Notes
			Min.	Max.	Equation	Min.	Max.	Equation	
1	$T_{DA(AS)}$	Address Valid to $\overline{AS}$ ↑ Delay	50		T_{PC} - 75	35		T_{PC} - 50	1, 2, 3
2	$T_{DAS(A)}$	$\overline{AS}$ ↑ to Address Float Delay	70		T_{PC} - 55	45		T_{PC} - 40	1, 2, 3
3	$T_{DAS(DR)}$	$\overline{AS}$ ↑ to Read Data Required Valid		360	4T_{PC} - 140		220	4T_{PC} - 110	1, 2, 3, 4
4	T_{WAS}	$\overline{AS}$ Low Width	80		T_{PC} - 45	55		T_{PC} - 30	1, 2, 3
5	$T_{DAZ(DS)}$	Address Float to $\overline{DS}$ ↓	0		3T_{PC} - 125	0		3T_{PC} - 65	1
6	T_{WDSR}	$\overline{DS}$ (Read) Low Width	250		2T_{PC} - 90	185		2T_{PC} - 55	1, 2, 3, 4
7	T_{WDSW}	$\overline{DS}$ (Write) Low Width	160		3T_{PC} - 175	110		3T_{PC} - 120	1, 2, 3, 4
8	$T_{DDSR(DR)}$	$\overline{DS}$ ↓ to Read Data Required Valid		200	T_{PC} - 55		130	T_{PC} - 40	1, 2, 3, 4
9	$T_{HDR(DS)}$	Read Data to $\overline{DS}$ ↓ Hold Time	0			0			1
10	$T_{DDS(A)}$	$\overline{DS}$ ↑ to Address Active Delay	70		T_{PC} - 55	45		T_{PC} - 30	1, 2, 3
11	$T_{DDS(AS)}$	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		T_{PC} - 75	55		T_{PC} - 55	1, 2, 3

EXTERNAL I/O OR MEMORY, READ WRITE AND CLOCK CYCLE TIMING (Continued)

N°	Symbol	Parameter	Z86E11/8MHz			Z86E11/12MHz			Notes
			Min.	Max.	Equation	Min.	Max.	Equation	
12	$T_{DR/W(AS)}$	R/$\overline{W}$ Valid to $\overline{AS}$ ↑ Delay	50		$T_{PC} - 65$	30		$T_{PC} - 50$	1, 2, 3
13	$T_{DDS(R/W)}$	$\overline{DS}$ ↑ to R/$\overline{W}$ Not Valid	60		$T_{PC} - 75$	35		$T_{PC} - 50$	1, 2, 3
14	$T_{DDW(DSW)}$	Write Data Valid to $\overline{DS}$ (Write) ↓ Delay	50		$T_{PC} - 55$	35		$T_{PC} - 40$	1, 2, 3
15	$T_{DDS(DW)}$	$\overline{DS}$ ↑ to Write Data Not Valid Delay	70		$5T_{PC} - 215$	45		$5T_{PC} - 160$	1, 2, 3
16	$T_{DA(DR)}$	Address Valid to Read Data Required Valid		410	$T_{PC} - 45$		255	$T_{PC} - 30$	1, 2, 3, 4
17	$T_{DAS(DS)}$	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	80			55			1, 2, 3

Notes : All values in ns.
1. Test Load 1.
2. Timing numbers given are for minimum T_{PC}.
3. Also see clock cycle time dependent characteristics table.
4. When using extended memory timing add 2 T_{PC}.

Figure 23 : External I/O or Memory Read/Write Timing.



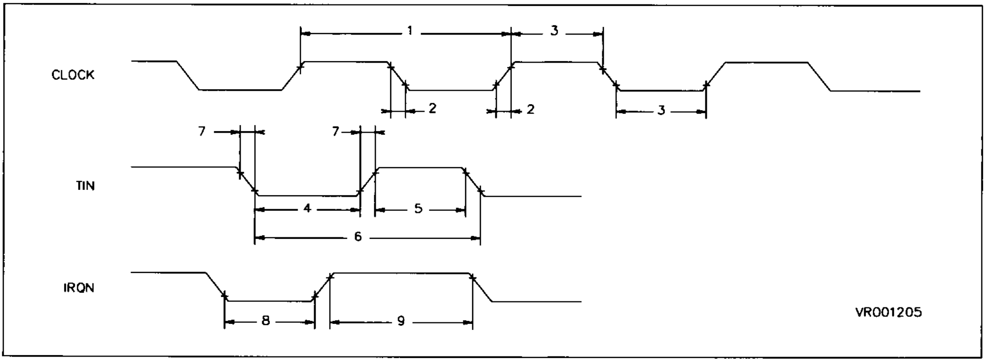
ADDITIONAL TIMING TABLE

N°	Symbol	Parameter	Z86E11/8MHz		Z86E11/12MHz		Unit	Notes
			Min.	Max.	Min.	Max.		
1	T_{PC}	Input Clock Period	125	1000	83	1000	ns	1
2	T_{RC}, T_{FC}	Clock Input Rise And Fall Times		25		15	ns	1
3	T_{WC}	Input Clock Width	37		26		ns	1
4	T_{WTINL}	Timer Input Low Width	100		70		ns	2
5	T_{WTINH}	Timer Input High Width	$3T_{PC}$		$3T_{PC}$		ns	2
6	T_{PTIN}	Timer Input Period	$8T_{PC}$		$8T_{PC}$		ns	2
7	T_{RTIN}, T_{FTIN}	Timer Input Rise And Fall Times		100		100	ns	2
8a	T_{WIL}	Interrupt Request Input Low Time	100		70		ns	2, 3
8b	T_{WIL}	Interrupt Request Input Low Time	$3T_{PC}$		$3T_{PC}$		ns	2, 4
9	T_{WIH}	Interrupt Request Input High Time	$3T_{PC}$		$3T_{PC}$		ns	2, 3

Notes :

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

Figure 24 : Additional Timing.



HANDSHAKE TIMING

N°	Symbol	Parameter	Z86E11/8MHz		Z86E11/12MHz		Unit	Notes
			Min.	Max.	Min.	Max.		
1	$T_{SDI(DAV)}$	Data In Setup Time	0		0		ns	
2	$T_{HDI(DAV)}$	Data In Hold Time	230		160		ns	
3	T_{WDAV}	Data Available Width	175		120		ns	
4	$T_{DDAVIF(RDY)}$	$\overline{DAV}$ ↓ Input to RDY ↓ Delay		175		120	ns	1, 2
5	$T_{DDAVOF(RDY)}$	$\overline{DAV}$ ↓ Output to RDY ↓ Delay	0		0		ns	1, 3
6	$T_{DDAVIR(RDY)}$	$\overline{DAV}$ ↑ Input to RDY ↑ Delay		175		120	ns	1, 2
7	$T_{DDAVOR(RDY)}$	$\overline{DAV}$ ↑ Output to RDY ↑ Delay	0		0		ns	1, 3
8	$T_{DDO(DAV)}$	Data Out to $\overline{DAV}$ ↓ Delay	50		30		ns	1
9	$T_{DRDY(DAV)}$	Rdy ↓ Input to $\overline{DAV}$ ↑ Delay	0	200	0	140	ns	1

Notes :

1. Test Load 1.
2. Input handshake.
3. Output handshake..
4. All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".

Figure 25 : Input Handshake Timing.

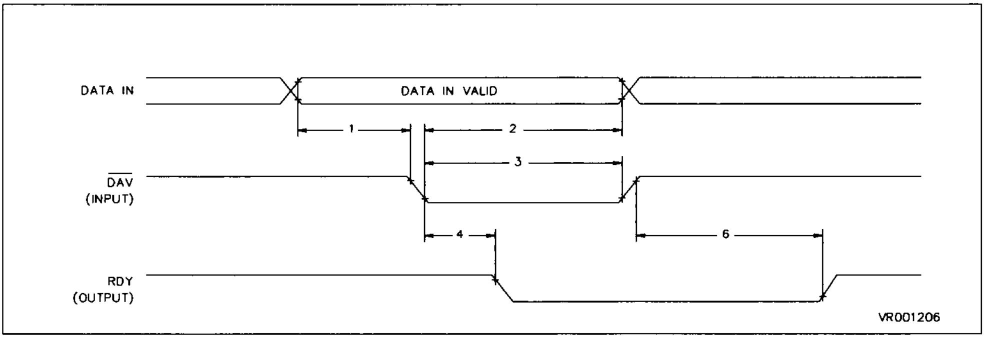


Figure 26 : Output Handshake Timing.

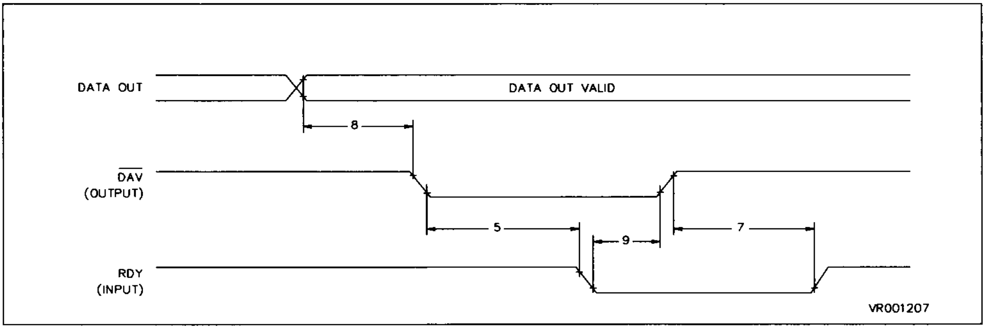
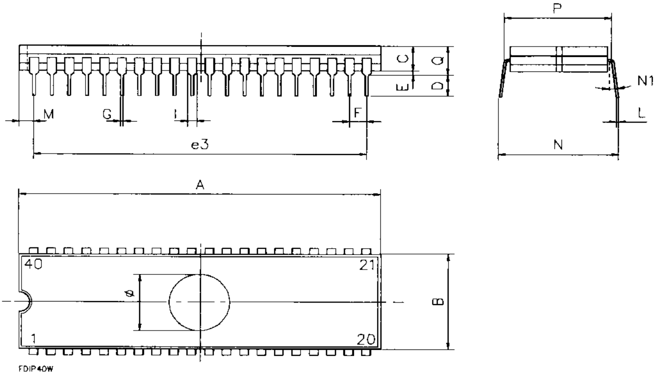


Figure 27 : 40-Lead Ceramic Windows Dual In Line Package (B)



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			53.40			2.102
B	14.50		14.90	0.571		0.587
C	3.90		5.08	0.154		0.200
D	3.40			0.134		
E	0.50		1.78	0.020		0.070
e 3		48.26			1.900	
F	2.29		2.79	0.090		0.110
G	0.40		0.55	0.016		0.022
I	1.27		1.52	0.050		0.060
L	0.22		0.31	0.009		0.012
M	1.52		2.49	0.060		0.098
N						
N1	0°		10°	0°		10°
P	15.40		15.80	0.606		0.622
Q			5.71			0.225
Ø	9.45		9.85	0.372		0.388

ORDERING INFORMATION

Type	Description	Frequency	Range	Package
Z86E11F1	4K EPROM	8MHz	0 to + 70°C	FDIP40 (Ceramic Glass Lens)
Z86E11AF1	4K EPROM	12MHz	0 to + 70°C	FDIP40 (Ceramic Glass Lens)

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsability for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all informations previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or system without express written approval of SGS-THOMSON Microelectronics.

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands
Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom