X25080/160/320/642/128

SPI Serial E^{2}PROM With Block Lock™ Protection

FEATURES

- 2.0MHz Clock Rate
- SPI Modes 0 & 3
- 1K/2K/4K/8K/16K X 8 Bits
  - —32 Byte Page Mode
- Low Power CMOS
  - —<1µA Standby Current
  - —<5mA Active Current
- 2.7V To 5.5V Power Supply
- Block Lock Protection
  - —Protect 1/4, 1/2 or all of E^{2}PROM Array
- Built-in Inadvertent Write Protection
  - —Power-Up/Power-Down protection circuitry
  - —Write Enable Latch
  - —Write Protect Pin
- Self-Timed Write Cycle
  - —5mS Write Cycle Time (Typical)
- High Reliability
  - —Endurance: 100,000 cycles
  - —Data Retention: 100 Years
  - —ESD protection: 2000V on all pins
- 8-Pin Mini-DIP Package
- Also available in TSSOP and SOIC

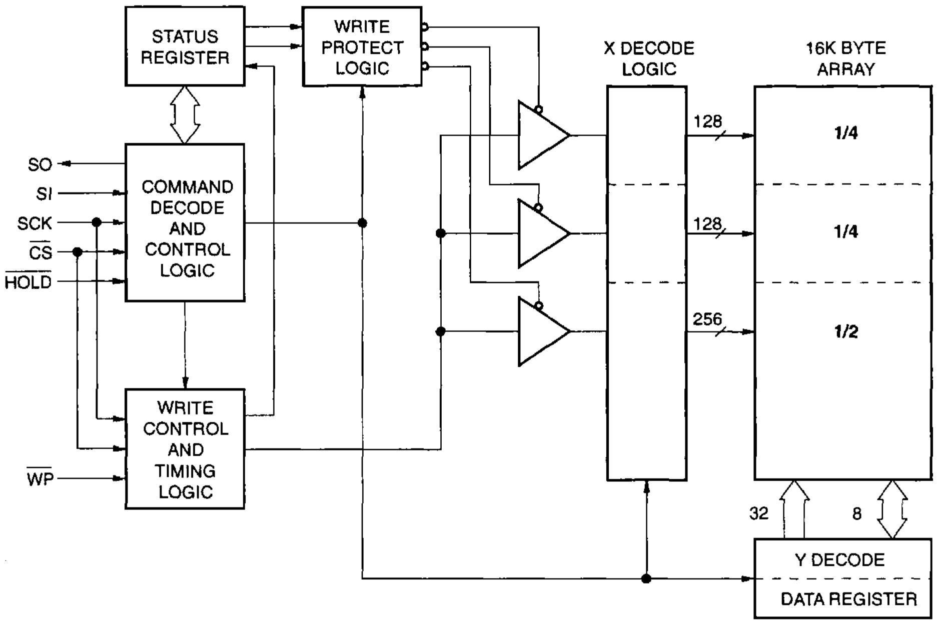
DESCRIPTION

The X25080/160/320/642/128 family are 8/16/32/64/128K CMOS bit serial E^{2}PROMs, internally organized x 8. They feature Serial Peripheral Interface (SPI) and software protocol allowing operation on a simple three wire bus. The bus signals are a clock input (SCK) plus separate data in (SI) and data out (SO) lines. Access to the device is controlled through a chip select ($\overline{CS}$) input, allowing any number of devices to share the same bus.

The X25080/160/320/642/128 also features two additional inputs that provide the end user with added flexibility. By asserting the $\overline{HOLD}$ input, the X25080/160/320/642/128 will ignore transitions on its inputs, thus allowing the host to service higher priority interrupts. The $\overline{WP}$ input can be used as a hardwire input to the X25080/160/320/642/128 disabling all write attempts to the status register; thus providing a mechanism for limiting end user capability of altering 0, 1/4, 1/2 or all of the memory.

The X25080/160/320/642/128 utilizes Xicor's proprietary Direct Write™ cell, providing a minimum endurance of 100,000 cycles and a minimum data retention of 100 years.

FUNCTIONAL DIAGRAM



3091 ILL F01.1

Direct Write™ and Block Lock™ Protection is a trademark of Xicor, Inc.

PIN DESCRIPTIONS

Serial Output (SO)

SO is a push-pull serial data output pin. During a read cycle, data is shifted out on this pin. Data is clocked out by the falling edge of the serial clock.

Serial Input (SI)

SI is the serial data input pin. All opcodes, byte addresses, and data to be written to the memory are input on this pin. Data is latched by the rising edge of the serial clock.

Serial Clock (SCK)

The Serial Clock controls the serial bus timing for data input and output. Opcodes, addresses, or data present on the SI pin are latched on the rising edge of the clock input, while data on the SO pin change after the falling edge of the clock input.

Chip Select ($\overline{\text{CS}}$)

When $\overline{\text{CS}}$ is high, the X25080/160/320/642/128 is deselected and the SO output pin is at high impedance and unless an internal write operation is underway the X25080/160/320/642/128 will be in the standby power mode. $\overline{\text{CS}}$ low enables the X25080/160/320/642/128, placing it in the active power mode. It should be noted that after power-on, a high to low transition on $\overline{\text{CS}}$ is required prior to the start of any operation.

Write Protect ($\overline{\text{WP}}$)

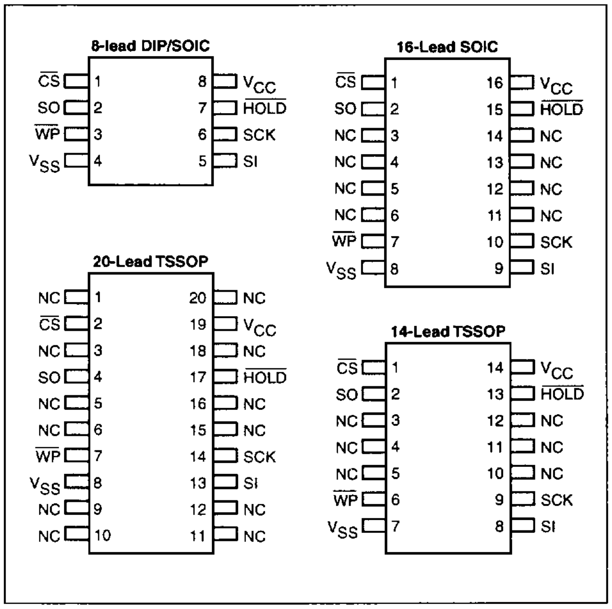
When $\overline{\text{WP}}$ is low and the nonvolatile bit WPEN is "1", nonvolatile writes to the X25080/160/320/642/128 status register are disabled, but the part otherwise functions normally. When $\overline{\text{WP}}$ is held high, all functions, including nonvolatile writes operate normally. $\overline{\text{WP}}$ going low while $\overline{\text{CS}}$ is still low will interrupt a write to the X25080/160/320/642/128 status register. If the internal write cycle has already been initiated, $\overline{\text{WP}}$ going low will have no effect on write.

The $\overline{\text{WP}}$ pin function is blocked when the WPEN bit in the status register is "0". This allows the user to install the X25080/160/320/642/128 in a system with $\overline{\text{WP}}$ pin grounded and still be able to write to the status register. The $\overline{\text{WP}}$ pin functions will be enabled when the WPEN bit is set "0".

Hold ($\overline{\text{HOLD}}$)

$\overline{\text{HOLD}}$ is used in conjunction with the $\overline{\text{CS}}$ pin to select the device. Once the part is selected and a serial sequence is underway, $\overline{\text{HOLD}}$ may be used to pause the serial communication with the controller without resetting the serial sequence. To pause, $\overline{\text{HOLD}}$ must be brought low while SCK is Low. To resume communication, $\overline{\text{HOLD}}$ is brought high, again while SCK is low. If the pause feature is not used, $\overline{\text{HOLD}}$ should be held high at all times.

PIN CONFIGURATION



3091 ILL F02.1

PIN NAMES

SYMBOL	DESCRIPTION
$\overline{\text{CS}}$	Chip Select Input
SO	Serial Output
SI	Serial Input
SCK	Serial Clock Input
$\overline{\text{WP}}$	Write Protect Input
V_{SS}	Ground
V_{CC}	Supply Voltage
$\overline{\text{HOLD}}$	Hold Input
NC	No Connect

3091 PGM T01

PRINCIPLES OF OPERATION

The X25080/160/320/642/128 family are serial E^{2}PROMs designed to interface directly with the synchronous serial peripheral interface (SPI) of many popular microcontroller families.

The X25080/160/320/642/128 family contains an 8-bit instruction register. It is accessed via the SI input, with data being clocked in on the rising SCK. $\overline{\text{CS}}$ must be low and the $\overline{\text{HOLD}}$ and $\overline{\text{WP}}$ inputs must be high during the entire operation. The $\overline{\text{WP}}$ input is "Don't Care" if WPEN is set "0".

Table 1 contains a list of the instructions and their operation codes. All instructions, addresses and data are transferred MSB first.

Data input is sampled on the first rising edge of SCK after $\overline{\text{CS}}$ goes low. SCK is static, allowing the user to stop the clock and then resume operations. If the clock line is shared with other peripheral devices on the SPI bus, the user can assert the $\overline{\text{HOLD}}$ input to place the X25080/160/320/642/128 into a "PAUSE" condition. After releasing $\overline{\text{HOLD}}$, the X25080/160/320/642/128 device will resume operation from the point when $\overline{\text{HOLD}}$ was first asserted.

Write Enable Latch

The X25080/160/320/642/128 device contains a write enable latch. This latch must be SET before a write operation will be completed internally. The WREN instruction will set the latch and the WRDI instruction will reset the latch. This latch is automatically reset upon a power-on condition and after the completion of a byte, page, or status register write cycle.

Status Register

The RDSR instruction provides access to the status register. The status register may be read at any time, even during a write cycle. The status register is formatted as follows:

7	6	5	4	3	2	1	0
WPEN	X	X	X	BP1	BP0	WEL	WIP

3091 PGM T02

WPEN, BP0 and BP1 are set by the WRSR instruction. WEL and WIP are read only and automatically set by other operations.

The **Write-In-Process (WIP)** bit indicates whether the X25080/160/320/642/128 device is busy with a write operation. When set to a "1" a write is in progress, when set to a "0" no write is in progress. During a write all other bits are set to "1".

The **Write Enable Latch (WEL)** bit indicates the status of the write enable latch. When set to a "1" the latch is set, when set to a "0" the latch is reset.

The **Block Protect (BP0 and BP1)** bits are nonvolatile and allows the user to select one of four levels of protection. The X25080/160/320/642/128 device array is divided into four equal segments. One, two, or all four of the segments may be protected. That is, the user may read the segments but will be unable to alter (write) data within the selected segments. The partitioning is controlled as illustrated below.

Status Register Bits		Array Addresses Protected
BP1	BP0	
0	0	None
0	1	upper fourth
1	0	upper half
1	1	All

3091 PGM T03

Write-Protect Enable

The Write-Protect-Enable (WPEN) is available for the X25080/160/320/642/128 device as an enable bit for the $\overline{\text{WP}}$ pin.

Table 1. Instruction Set

Instruction Name	Instruction Format*	Operation
WREN	0000 0110	Set the Write Enable Latch (Enable Write Operations)
WRDI	0000 0100	Reset the Write Enable Latch (Disable Write Operations)
RDSR	0000 0101	Read Status Register
WRSR	0000 0001	Write Status Register
READ	0000 0011	Read Data from Memory Array beginning at selected address
WRITE	0000 0010	Write Data to Memory Array beginning at Selected Address (1 to 32 Bytes)

3091 PGM T04

*Instructions are shown MSB in leftmost position. Instructions are transferred MSB first.

WPEN	$\overline{WP}$	WEL	Protected Blocks	Unprotected Blocks	Status Register
0	X	0	Protected	Protected	Protected
0	X	1	Protected	Writable	Writable
1	Low	0	Protected	Protected	Protected
1	Low	1	Protected	Writable	Protected
X	High	0	Protected	Protected	Protected
X	High	1	Protected	Writable	Writable

3091 PGM T05

The Write Protect ($\overline{WP}$) pin and the nonvolatile Write Protect Enable (WPEN) bit in the Status Register control the programmable hardware write protect feature. Hardware write protection is enabled when $\overline{WP}$ pin is low, and the WPEN bit is "1". Hardware write protection is disabled when either the $\overline{WP}$ pin is high or the WPEN bit is "0". When the chip is hardware write protected, nonvolatile writes are disabled to the Status Register, including the Block Protect bits and the WPEN bit itself, as well as the block-protected sections in the memory array. Only the sections of the memory array that are not block-protected can be written.

Note: Since the WPEN bit is write protected, it cannot be changed back to a "0", as long as the $\overline{WP}$ pin is held low.

Clock and Data Timing

Data input on the SI line is latched on the rising edge of SCK. Data is output on the SO line by the falling edge of SCK.

Read Sequence

When reading from the E^2PROM array, $\overline{CS}$ is first pulled low to select the device. The 8 bit READ instruction is transmitted to the X25080/160/320/642/128 device, followed by the 16 bit. After the read opcode and address are sent, the data stored in the memory at the selected address is shifted out on the SO line. The data stored in memory at the next address can be read sequentially by continuing to provide clock pulses. The address is automatically incremented to the next higher address after each byte of data is shifted out. When the highest address is reached the address counter rolls over to address $0000 allowing the read cycle to be continued indefinitely. The read operation is terminated by taking $\overline{CS}$ high. Refer to the read E^2PROM array operation sequence illustrated in Figure 1.

To read the status register the $\overline{CS}$ line is first pulled low to select the device followed by the 8 bit instruction. After the read status register opcode is sent, the contents of the status register is shifted out on the SO line. The read status register sequence is illustrated in Figure 2.

Write Sequence

Prior to any attempt to write data into the X25080/160/320/642/128 device the write enable latch must first be set by issuing the WREN instruction (See Figure 3). $\overline{CS}$ is first taken low, then the WREN instruction is clocked into the X25080/160/320/642/128 device. After all eight bits of the instruction are transmitted, $\overline{CS}$ must then be taken high. If the user continues the write operation without taking $\overline{CS}$ high after issuing the WREN instruction the write operation will be ignored.

To write data to the E^2PROM memory array, the user issues the WRITE instruction, followed by the address and then the data to be written. This is minimally a thirty-two clock operation. $\overline{CS}$ must go low and remain low for the duration of the operation. The host may continue to write up to 32 bytes of data to the X25080/160/320/642/128 device. The only restriction is the 32 bytes must reside on the same page. If the address counter reaches the end of the page and the clock continues the counter will roll back to the first address of the page and overwrite any data that may have been written.

For the write operation (byte or page write) to be completed, $\overline{CS}$ can only be brought high after bit 0 of data byte N is clocked in. If it is brought high at any other time the write operation will not be completed. Refer to Figures 4 and 5 below for a detailed illustration of the write sequences and time frames in which $\overline{CS}$ going high are valid.

To write to the status register, the WRSR instruction is followed by the data to be written. Data bits 0, 1, 4, 5 and 6 must be "0". This sequence is shown in Figure 6.

While the write is in progress, following a status register or E^2PROM write sequence, the status register may be read to check the WIP bit. During this time the WIP bit will be high.

Hold Operation

The $\overline{HOLD}$ input should be high (at V_{IH}) under normal operation. If a data transfer is to be interrupted $\overline{HOLD}$ can be pulled low to suspend the transfer until it can be resumed. The only restriction is the SCK input must be low when $\overline{HOLD}$ is first pulled low and SCK must also be low when $\overline{HOLD}$ is released.

The $\overline{HOLD}$ input may be tied high either directly to V_{CC} or tied to V_{CC} through a resistor.

Operational Notes

The X25080/160/320/642/128 device powers-on in the following state:

- The device is in the low power standby state.
- A high to low transition on $\overline{CS}$ is required to enter an active state and receive an instruction.
- SO pin is high impedance.
- The write enable latch is reset.

Data Protection

The following circuitry has been included to prevent inadvertent writes:

- The write enable latch is reset upon power-up.
- A write enable instruction must be issued to set the write enable latch.
- $\overline{CS}$ must come high at the proper clock count in order to start a write cycle.

Figure 1. Read E²PROM Array Operation Sequence

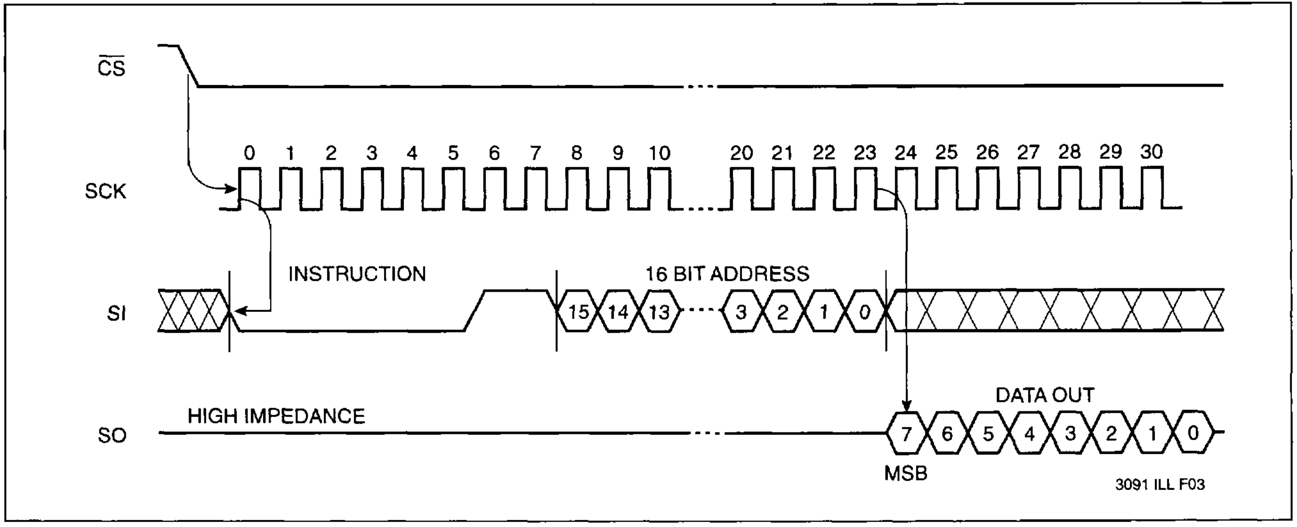


Figure 2. Read Status Register Operation Sequence

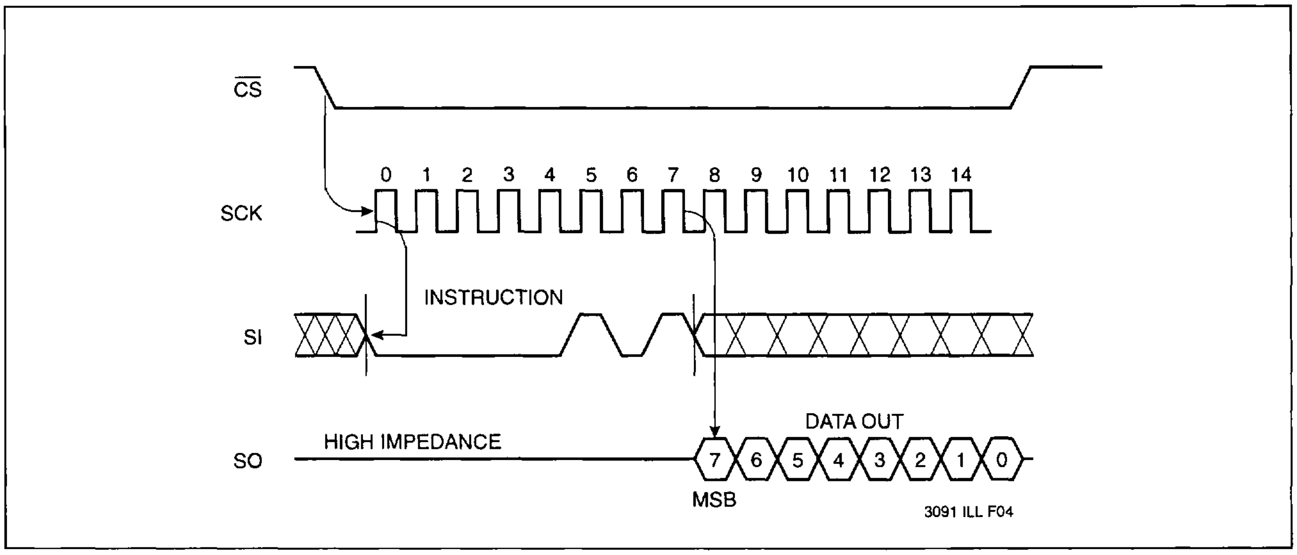


Figure 3. Write Enable Latch Sequence

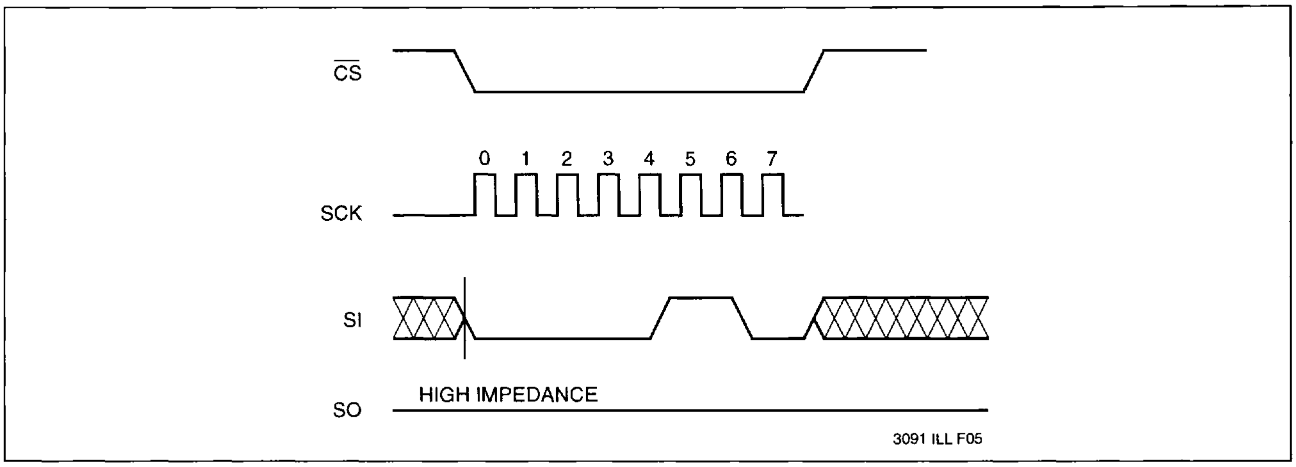


Figure 4. Byte Write Operation Sequence

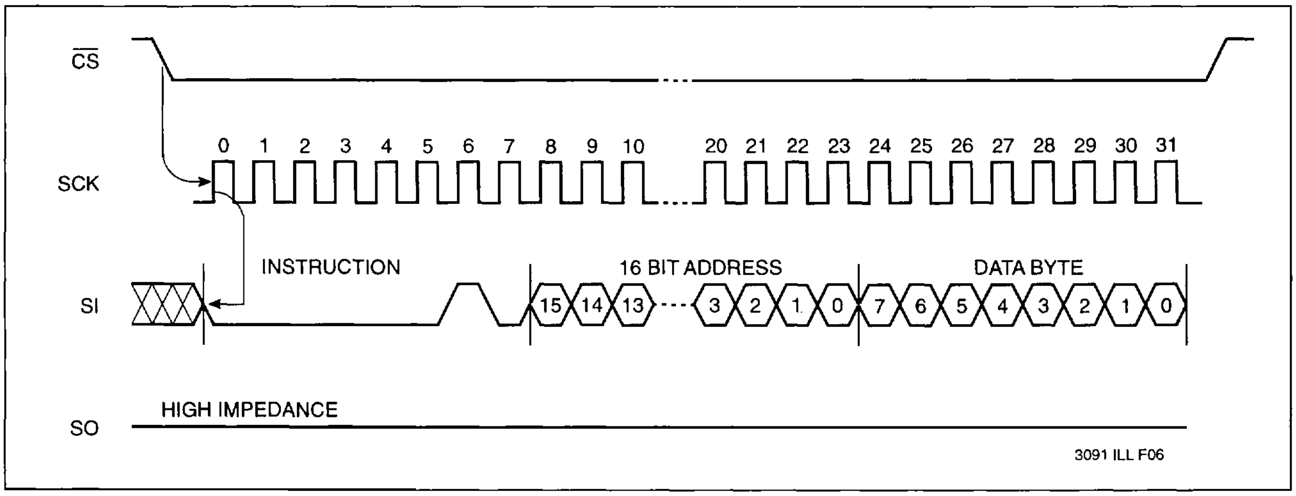


Figure 5. Page Write Operation Sequence

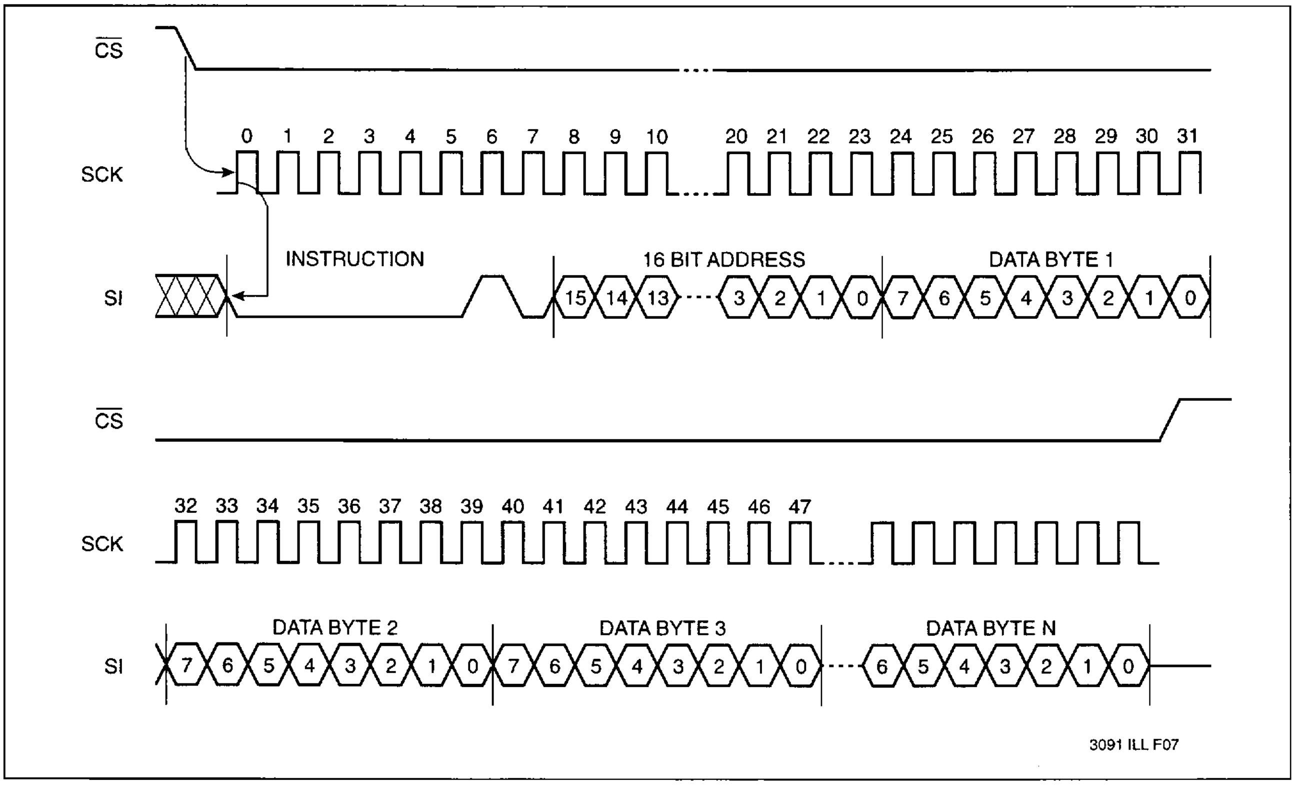
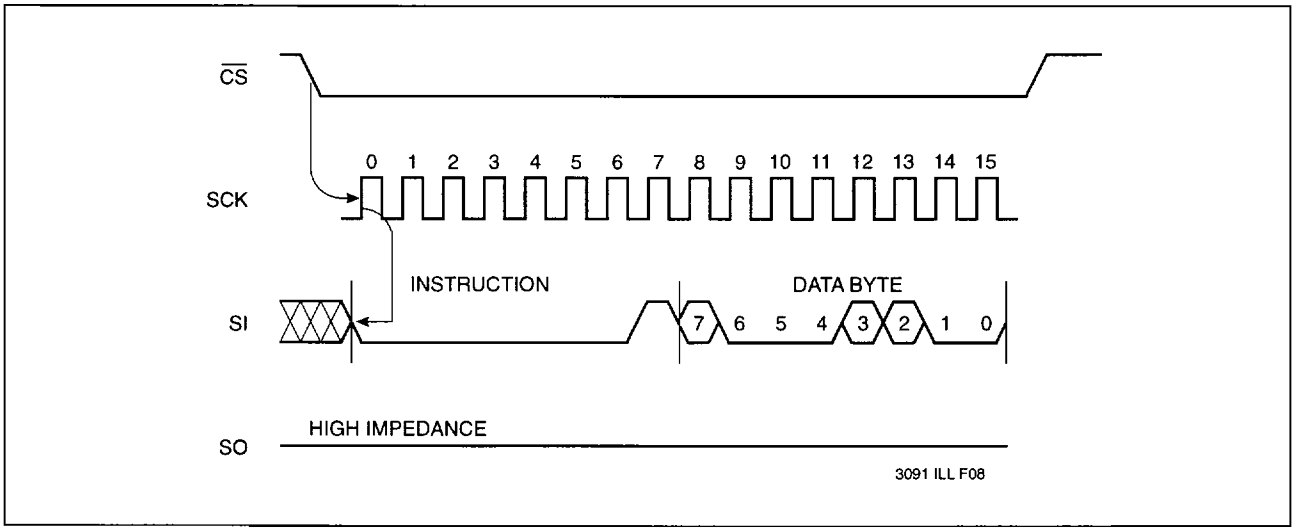


Figure 6. Write Status Register Operation Sequence



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ....–65°C to +135°C
Storage Temperature ....–65°C to +150°C
Voltage on any Pin with Respect to Ground –1.0V to +7V
D.C. Output Current 5mA
Lead Temperature
(Soldering, 10 Seconds) .... 300°C

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and the functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Temp	Min.	Max.
Commercial	0°C	70°C
Industrial	–40°C	+85°C
Military	–55°C	+125°C

3091 PGM T06

Supply Voltage	Limits
X25080/160/320/642/128	5V ± 10%
X25080/160/320/642/128-2.7	2.7V to 5.5V

3091 PGM T07

D.C. OPERATING CHARACTERISTICS (Over the recommended operating conditions unless otherwise specified.)

Symbol	Parameter	Limits		Units	Test Conditions
		Min.	Max.		
I_{CC}	V_{CC} Supply Current (Active)		5	mA	SCK = V_{CC} x 0.1/V_{CC} x 0.9 @ 2MHz, SO = OPEN, $\overline{CS}$ = Gnd
I_{SB}	V_{CC} Supply Current (Standby)		1	µA	$\overline{CS}$ = V_{CC}, V_{IN} = Gnd or V_{CC}
I_{LI}	Input Leakage Current		10	µA	V_{IN} = GND to V_{CC}
I_{LO}	Output Leakage Current		10	µA	V_{OUT} = GND to V_{CC}
V_{IL}(1)	Input Low Voltage	–1.0	V_{CC} x 0.3	V	
V_{IH}(1)	Input High Voltage	V_{CC} x 0.7	V_{CC} + 0.5	V	
V_{OL1}	Output Low Voltage		0.4	V	V_{CC} = 5V, I_{OL} = 3mA
V_{OH1}	Output High Voltage	V_{CC} – 0.8		V	V_{CC} = 5V, I_{OH} = -1.6mA
V_{OL2}	Output Low Voltage		0.4	V	V_{CC} = 3V, I_{OL} = 1.5mA
V_{OH2}	Output High Voltage	V_{CC} – 0.3		V	V_{CC} = 3V, I_{OH} = -0.4mA

3091 PGM T08

POWER-UP TIMING

Symbol	Parameter	Min.	Max.	Units
t_{PUR}(3)	Power-up to Read Operation		1	ms
t_{PUW}(3)	Power-up to Write Operation		5	ms

3091 PGM T09

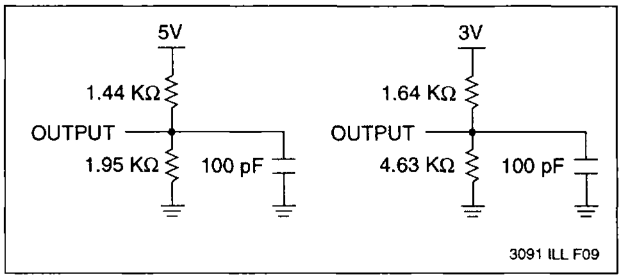
CAPACITANCE T_A = 25°C, f = 1.0MHz, V_{CC} = 5V.

Symbol	Test	Max.	Units	Conditions
C_{OUT}(2)	Output Capacitance (SO)	8	pF	V_{OUT} = 0V
C_{IN}(2)	Input Capacitance (SCK, SI, $\overline{CS}$, $\overline{WP}$, $\overline{HOLD}$)	6	pF	V_{IN} = 0V

3091 PGM T10

Notes: (1) V_{IL} Min. and V_{IH} Max. are for reference only and are not tested.
(2) This parameter is periodically sampled and not 100% tested.
(3) t_{PUR} and t_{PUW} are the delays required from the time V_{CC} is stable until the specified operation can be initiated. These parameters are periodically sampled and not 100% tested.

EQUIVALENT A.C. LOAD CIRCUIT



A.C. TEST CONDITIONS

Input Pulse Levels	V_{CC} x 0.1 to V_{CC} x 0.9
Input Rise and Fall Times	10ns
Input and Output Timing Level	V_{CC} x 0.5

3091 PGM T11

A.C. CHARACTERISTICS (Over recommended operating conditions, unless otherwise specified)

Data Input Timing

Symbol	Parameter	Min.	Max.	Units
f_{SCK}	Clock Frequency	0	2.0	MHz
t_{CYC}	Cycle Time	500		ns
t_{LEAD}	$\overline{CS}$ Lead Time	250		ns
t_{LAG}	$\overline{CS}$ Lag Time	250		ns
t_{WH}	Clock High Time	200		ns
t_{WL}	Clock Low Time	200		ns
t_{SU}	Data Setup Time	50		ns
t_{H}	Data Hold Time	50		ns
t_{RI}(4)	Data In Rise Time		2.0	µs
t_{FI}(4)	Data In Fall Time		2.0	µs
t_{HD}	$\overline{HOLD}$ Setup Time	100		ns
t_{CD}	$\overline{HOLD}$ Hold Time	100		ns
t_{CS}	$\overline{CS}$ Deselect Time	2.0		µs
t_{WC}(5)	Write Cycle Time		10	ms

3091 PGM T12

Data Output Timing

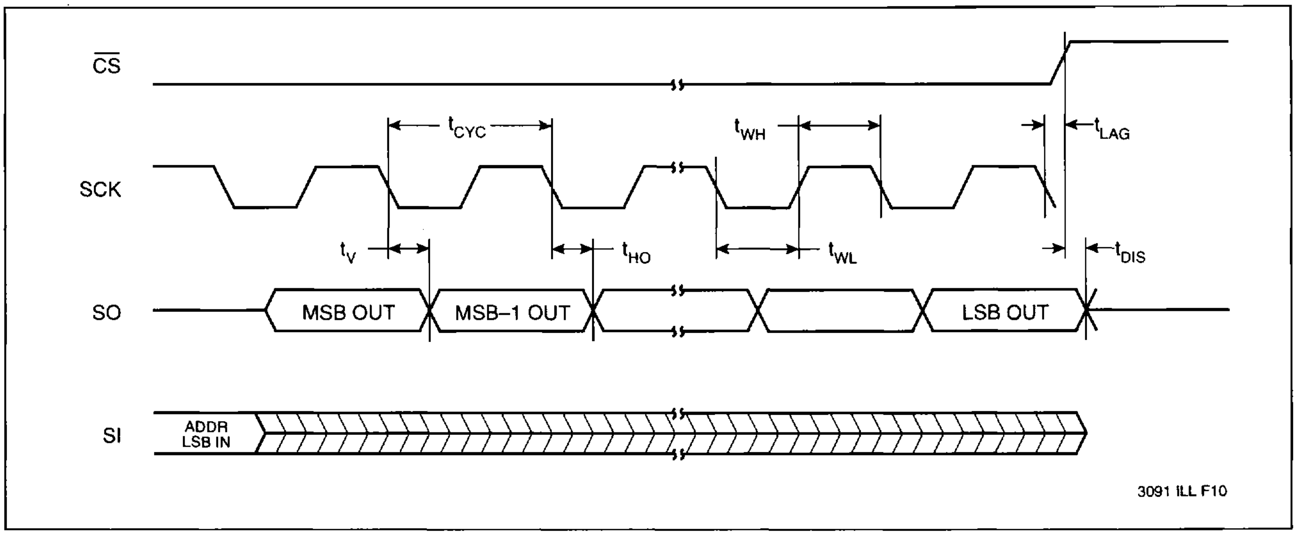
Symbol	Parameter	Min.	Max.	Units
f_{SCK}	Clock Frequency	0	2.0	MHz
t_{DIS}	Output Disable Time		250	ns
t_{V}	Output Valid from clock Low		200	ns
t_{HO}	Output Hold Time	0		ns
t_{RO}(4)	Output Rise Time		100	ns
t_{FO}(4)	Output Fall Time		100	ns
t_{LZ}(4)	$\overline{HOLD}$ High to Output in Low Z	100		ns
t_{HZ}(4)	$\overline{HOLD}$ Low to Output in High Z	100		ns

3091 PGM T13

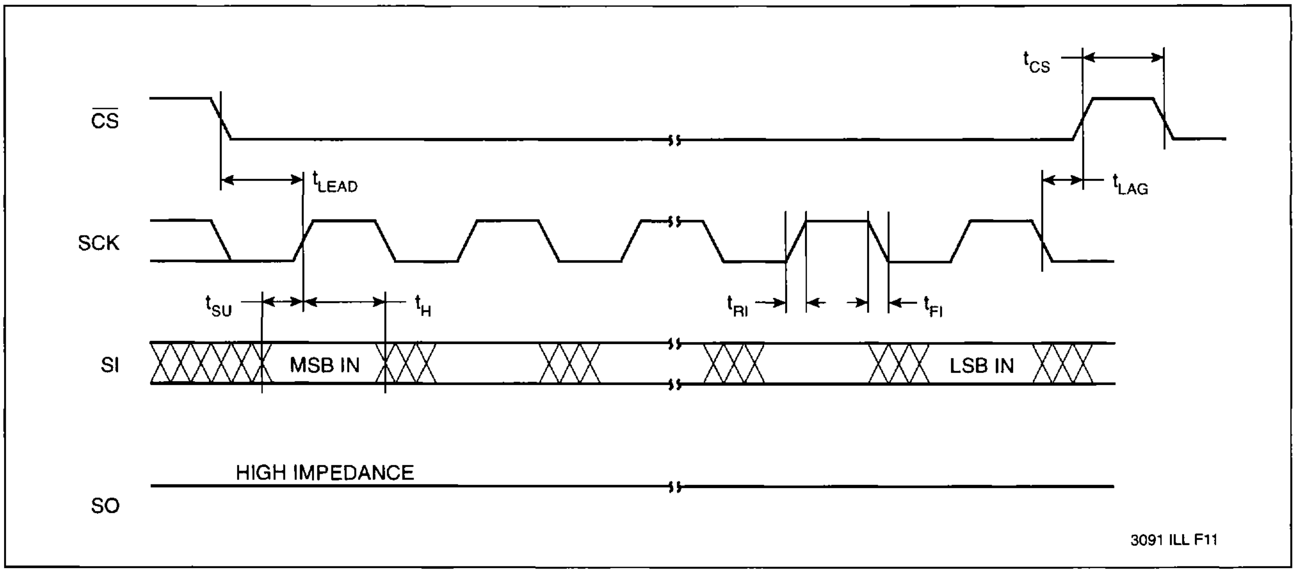
Notes: (4) This parameter is periodically sampled and not 100% tested.

(5) t_{WC} is the time from the rising edge of $\overline{CS}$ after a valid write sequence has been sent to the end of the self-timed internal nonvolatile write cycle.

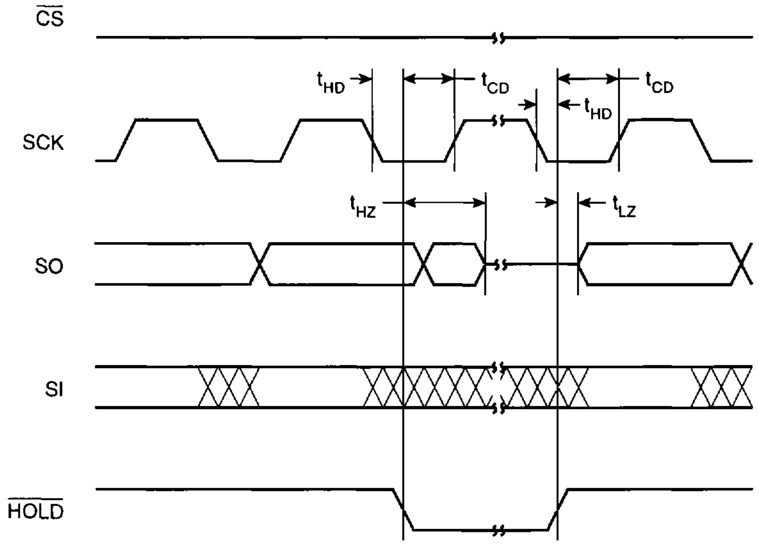
Serial Output Timing



Serial Input Timing



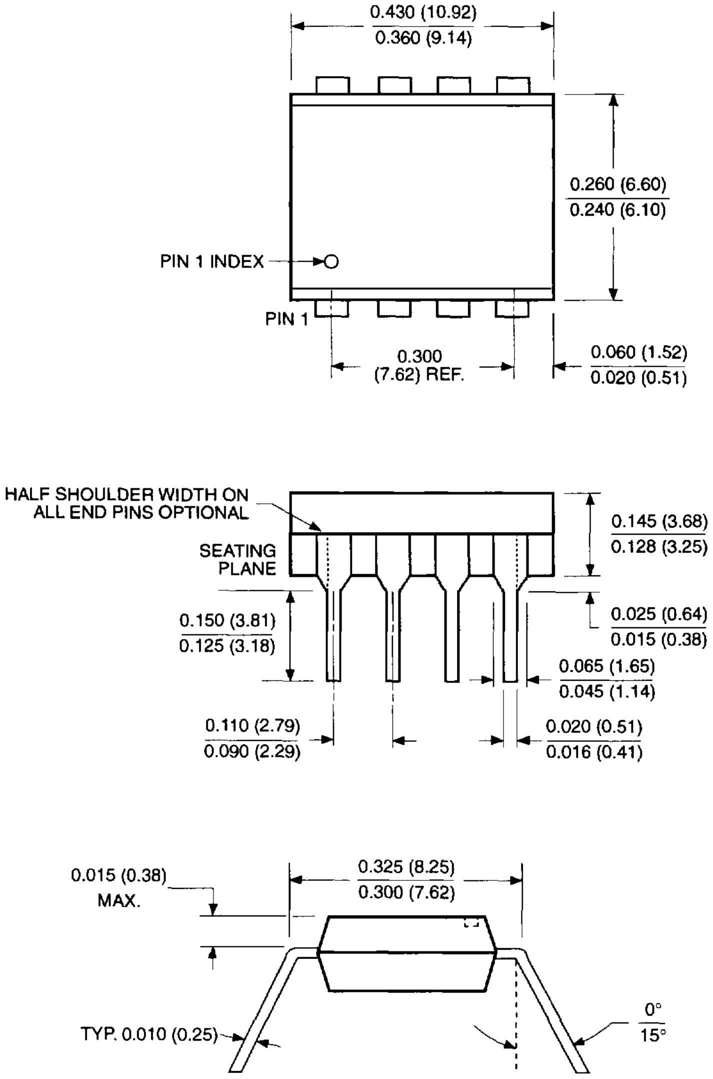
Hold Timing



3091 ILL F12

PACKAGING INFORMATION

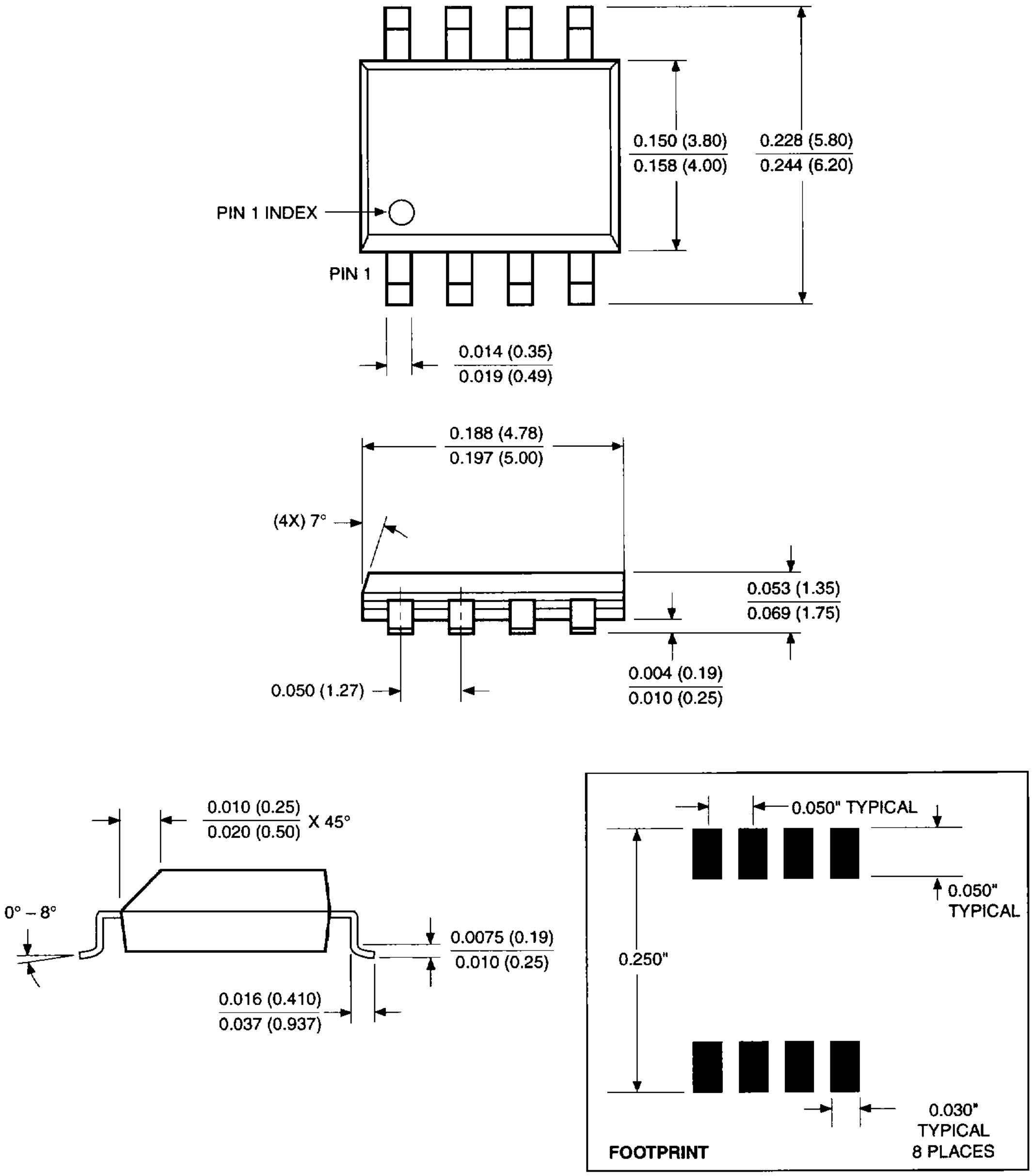
8-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



NOTE:

1. **ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)**
2. **PACKAGE DIMENSIONS EXCLUDE MOLDING FLASH**

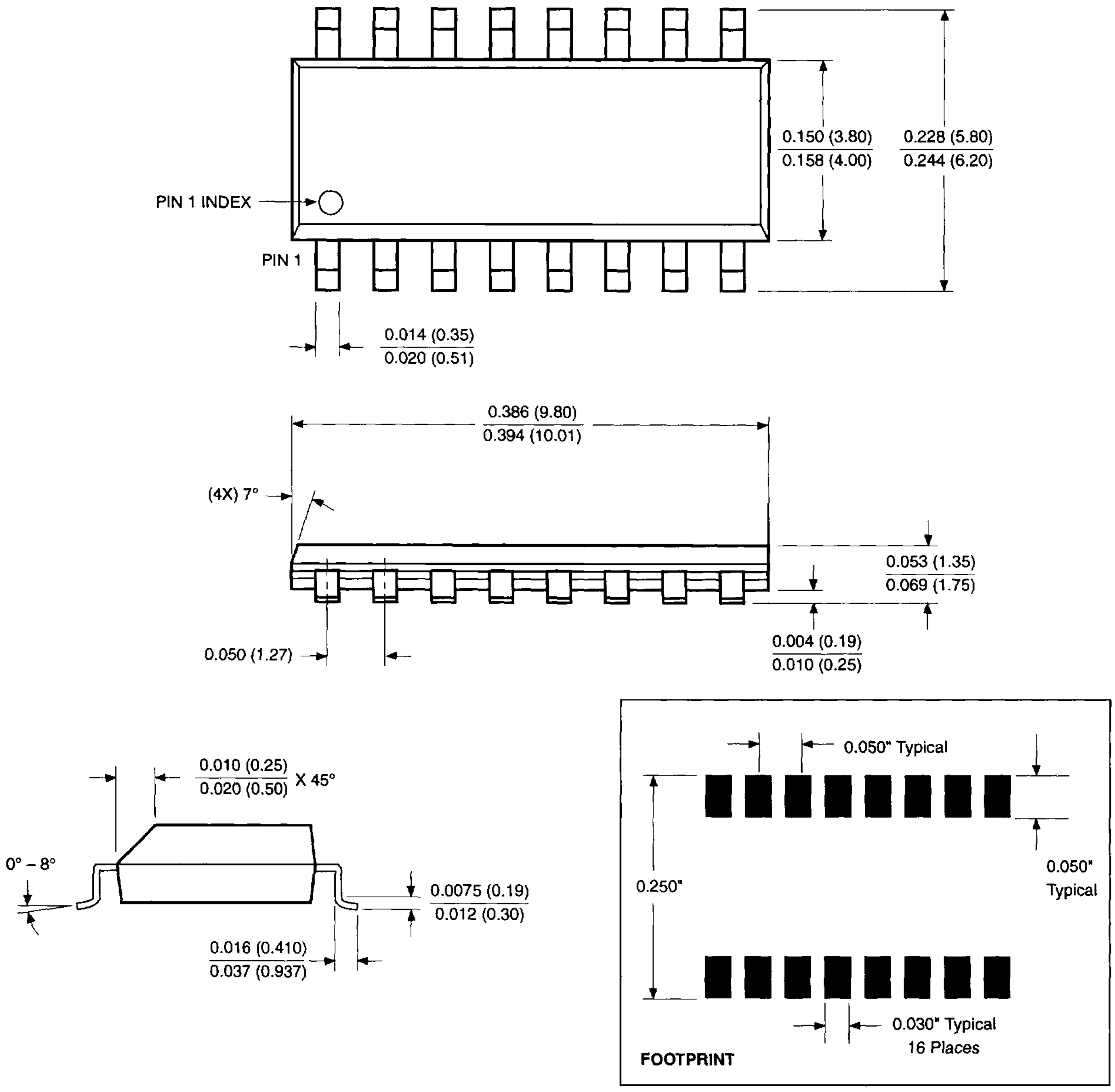
8-LEAD PLASTIC SMALL OUTLINE GULL WING PACKAGE TYPE S



NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESIS IN MILLIMETERS)

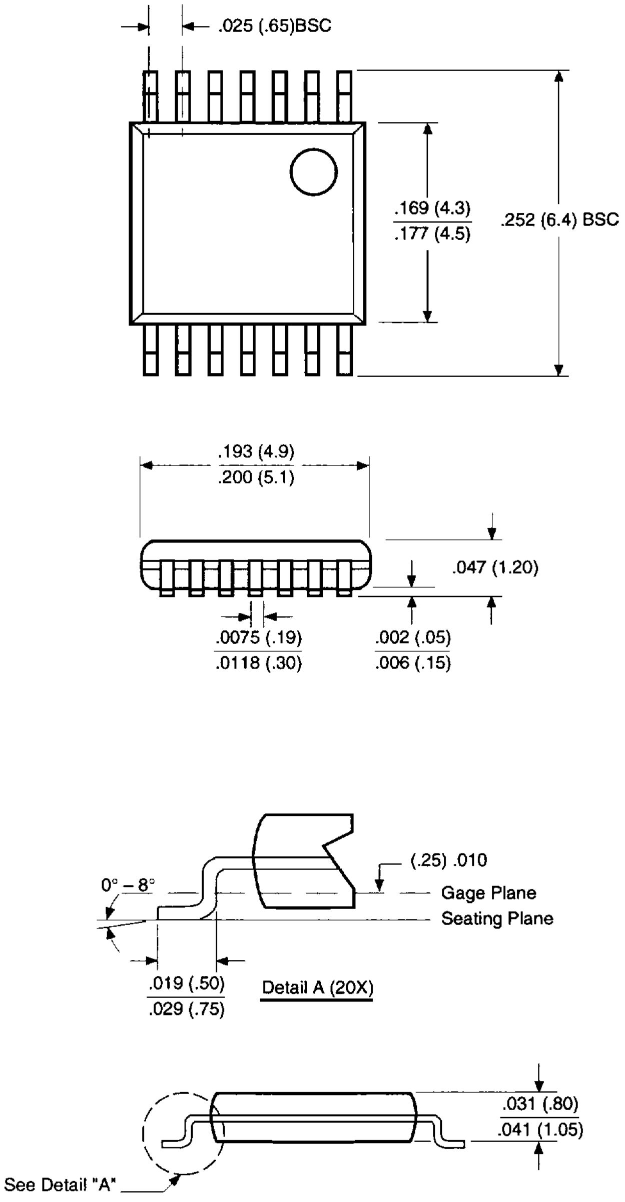
PACKAGING INFORMATION

16-LEAD PLASTIC SMALL OUTLINE GULL WING PACKAGE TYPE S



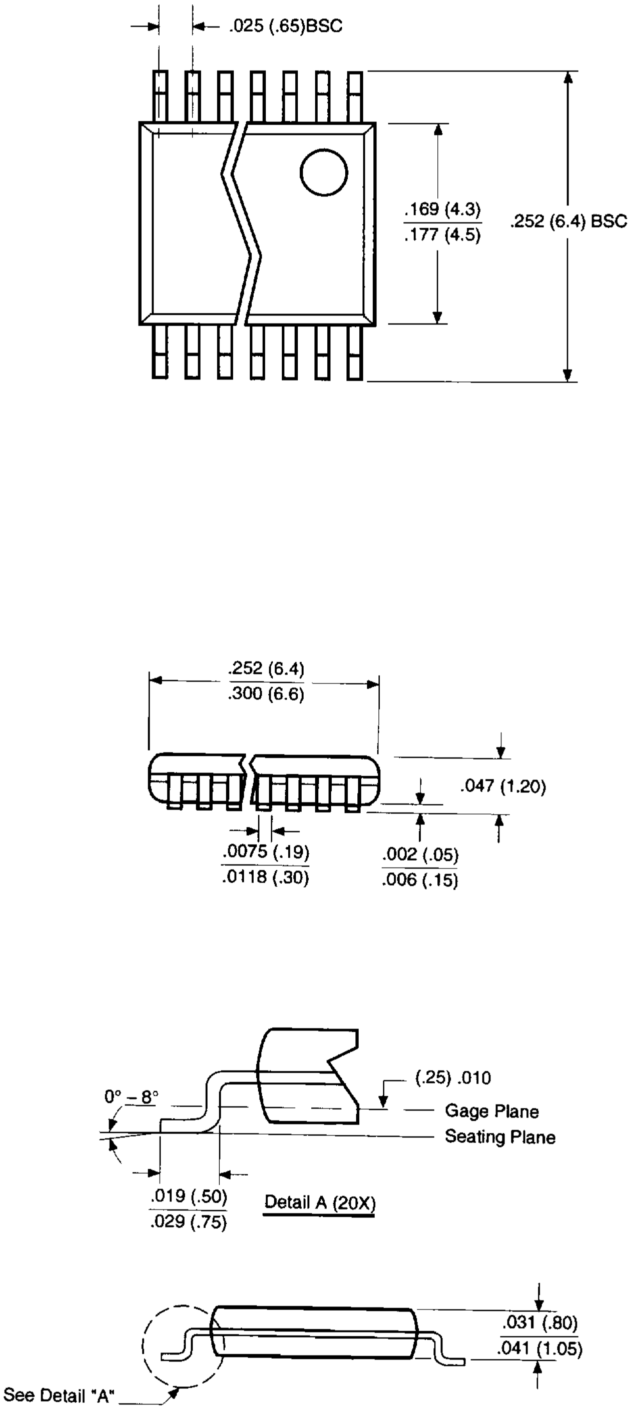
NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)

14-LEAD PLASTIC, TSSOP PACKAGE TYPE V



NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)

20-LEAD PLASTIC, TSSOP PACKAGE TYPE V



NOTE: ALL DIMENSIONS IN INCHES (IN PARENTHESES IN MILLIMETERS)

ORDERING INFORMATION

X25080 P T -V
X25160
X25320
X25642
X26128

Device

V_{CC} Limits
Blank = 5V ± 10%
2.7 = 2.7V to 5.5V

Temperature Range
Blank = Commercial = 0°C to +70°C
I = Industrial = –40°C to +85°C
M = Military = –55°C to +125°C

Package

X25080
X25160
X25320
P = 8 Lead Plastic DIP
S = 8 Lead SOIC
V = 14 Lead TSSOP

X25642
P = 8 Lead SOIC
S8 = 8 Lead SIOC
V = 20 Lead TSSOP

X25128
P = 8 Lead Plastic DIP
S = 16 Lead SOIC

Part Mark Convention

X25xxx — P = 8-Lead Plastic DIP
S = 16-Lead SOIC

X

Blank = 5V ± 10%, 0°C to +70°C
I = 5V ± 10%, –40°C to +85°C
F = 2.7V to 5.5V, 0°C to +70°C
G = 2.7V to 5.5V, –40°C to +85°C

LIMITED WARRANTY

Devices sold by Xicor, Inc. are covered by the warranty and patent indemnification provisions appearing in its Terms of Sale only. Xicor, Inc. makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. Xicor, Inc. makes no warranty of merchantability or fitness for any purpose. Xicor, Inc. reserves the right to discontinue production and change specifications and prices at any time and without notice.

Xicor, Inc. assumes no responsibility for the use of any circuitry other than circuitry embodied in a Xicor, Inc. product. No other circuits, patents, licenses are implied.

U.S. PATENTS

Xicor products are covered by one or more of the following U.S. Patents: 4,263,664; 4,274,012; 4,300,212; 4,314,265; 4,326,134; 4,393,481; 4,404,475; 4,450,402; 4,486,769; 4,488,060; 4,520,461; 4,533,846; 4,599,706; 4,617,652; 4,668,932; 4,752,912; 4,829, 482; 4,874, 967; 4,883, 976. Foreign patents and additional patents pending.

LIFE RELATED POLICY

In situations where semiconductor component failure may endanger life, system designers using this product should design the system with appropriate error detection and correction, redundancy and backup features to prevent such an occurrence.

Xicor's products are not authorized for use in critical components in life support devices or systems.

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Xicor®

Applications Brief

Programmable Hardware Write Protection?
(X25080/160/320/640/642/128)

One of the major considerations when designing with any form of memory is ensuring data is adequately protected. This is particularly true of programmable memory which may be subject to inadvertent write conditions. To guard against this, serial E^{2}PROMs have up to now been protected using either a single hardware write product pin or some form of software block protection. Xicor's high density SPI devices are the first serial E^{2}PROMs that combine the advantages of both these techniques to provide Programmable Hardware Write Protection.

This hybrid form of write protection allow selectable blocks of memory to be permanently protected via a hardware write product pin that is enabled through software. The advantage of this scheme is that data can be downloaded to the device in-system and then be 'secured' through hardware without changing the status of any of the pins. Traditionally, devices using hardware protection have needed to be pre-programmed before being mounted onto a circuit board, thereby adding an additional manufacturing step. This new feature is particularly useful for systems utilizing surface mount devices that are cumbersome to preprogram, and in applications where programming may be required just prior to shipment.

The truth table for setting the various levels of protection available on the high density SPI devices is shown below:

$\overline{\text{WP}}$ Pin	WPEN Bit	BP0 and BP1 Protected Memory Blocks	BP0 and BP1 Unprotected Memory Blocks	WEL	BP0/1 and WPEN Bits	Type of Write Block Protection
L	1	Not Writable	Writable	Settable	Not Writable	Hardware
H	X	Not Writable	Writable	Settable	Writable	Software
X	0	Not Writable	Writable	Settable	Writable	Software

The WPEN (Write Protect Enable) bit, used to enable the hardware write protection, and the BP0 and BP1 (Block Protect) bits, used to define the blocks of the E^{2}PROM array to be protected, are nonvolatile E^2 that are read and written via the status register. The status of the WEL (Write Enable Latch) is determined by reading the status register and must be set before a write can occur. This latch also provides additional protection by being automatically reset on the completion of a write cycle or power down condition.

It should be noted that once hardware write protection has been enabled, it can only be disabled by changing the state of the write protect pin, and not through software.

```
;************************************************************************
;*        Copyright (c) 1995 Xicor, Inc.
;*        AUTHOR: Richard Downing
;************************************************************************
;* The purpose of this code is to provide routines to interface the Xicor X25080/160/320/640/642/128
;* SPI serial EEPROMs with the 8031 microcontroller. The interface uses the 8031's general purpose
;* parallel port 1 and connects P1.0 to the chip select line (/CS), P1.1 to the serial input data line (SI),
;* P1.2 to the serial clock line (SCK) and P1.3 to the serial output data line (SO).
;*
;*  All the SPI serial EEPROM commands are provided. These are :-
;*
;*  1. Set Write Enable Latch
;*  2. Reset Write Enable Latch
;*  3. Write Status Register
;*  4. Read Status Register
;*  5. Single Byte Write
;*  6. Single Byte Read
;*  7. Page Write
;*  8. Sequential Read
;*
;*  The code writes 00H to the Status Register; reads the Status Register; writes 11H to
;*  address 55H in Byte Mode; performs a single Byte Read from address 55H; writes
;*  22H, 33H, 44H to addresses 300H, 301H, 302H in Page Mode; and performs a
;*  Sequential Read from addresses 300H, 301H, 302H.
;************************************************************************
;* CONSTANTS
cs                      bit     P1.0    ; Port 1 bit 0 used for chip select (/CS)
si                      bit     P1.1    ; Port 1 bit 1 used for serial input (CI)
sck                     bit     P1.2    ; Port 1 bit 2 used for serial clock (SCK)
so                      bit     P1.3    ; Port 1 bit 3 used for serial output (SO)
WREN_INST               equ     06H     ; Write enable latch instruction (WREN)
WRDI_INST               equ     04H     ; Write disable latch instruction (WRDI)
WRSR_INST               equ     01H     ; Write status register instruction (WRSR)
RDSR_INST               equ     05H     ; Read status register instruction (RDSR)
WRITE_INST              equ     02H     ; Write memory instruction (WRITE)
READ_INST               equ     03H     ; Read  memory instruction (READ)
BYTE_ADDR               equ     55H     ; Memory address for byte mode operations
BYTE_DATA               equ     11H     ; Data byte for byte write operation
PAGE_ADDR               equ     300H    ; Memory address for page mode operations
PAGE_DATA1              equ     22H     ; 1st data byte for page write operation
PAGE_DATA2              equ     33H     ; 2nd data byte for page write operation
PAGE_DATA3              equ     44H     ; 3rd data byte for page write operation
STATUS_REG              equ     00H     ; Status register
MAX_POLL                equ     99H     ; Maximum number of  polls
INIT_STATE              equ     09H     ; Initialization value for control ports
SLIC                    equ     030H    ; Address location of SLIC
;************************************
;* INTERNAL RAM
STACK_TOP               equ     060H    ; Stack top
;************************************
;* CODE
        ORG     0000H                   ; Reset vectors to this location
        ljmp    main
        ORG     0100H
main:
        mov     SP, #STACK_TOP          ; Initialize stack pointer
        clr     EA                      ; Disable interupts
        mov     P1, #INIT_STATE         ; Initialize control lines (/CS, SO high; SCK, SI low)
        lcall   wren_cmd                ; Set write enable latch
        lcall   wrsr_cmd                ; Write 00H to status register
        lcall   wren_cmd                ; Set write enable latch
        lcall   byte_write              ; Write 11H to address 55H (Byte Write)
        lcall   byte_read               ; Read from address location 55H (Byte Read)
        lcall   wren_cmd                ; Set write enable latch
        lcall   page_write              ; Write 22H/33H/44H to addresses 300/1/2H (Page Write)
        lcall   sequ_read               ; Read from address locations 300/1/2H (Sequential Read)
        jmp     SLIC
;***************************
;* Name:  WREN_CMD
;* Description: Set Write Enable Latch
;* Function: This routine sends the command to enable writes to the EEPROM memory array or status
;* register
;* Calls: outbyt
;* Input: None
;* Outputs: None
;* Register Usage: A
;***************************
wren_cmd:
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #WREN_INST
        lcall   outbyt                  ; Send WREN instruction
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        ret
;***************************
;* Name:  WRDI_CMD
;* Description: Reset Write Enable Latch
;* Function: This routine sends the command to disable writes to the EEPROM memory array or status
;* register
;* Calls: outbyt
;* Input: None
;* Outputs: None
;* Register Usage: A
;***************************
wrdi_cmd:
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #WRDI_INST
        lcall   outbyt                  ; Send WRDI instruction
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        ret
;***************************
;* Name:  WRSR_CMD
;* Description: Write Status Register
;* Function: This routine sends the command to write the BP0, BP1  and WPEN EEPROM bits in the
;* status register
;* Calls: outbyt, wip_poll
;* Input: None
;* Outputs: None
;* Register Usage: A
;***************************
```

```
wrsr_cmd:
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #WRSR_INST
        lcall   outbyt                  ; Send WRSR instruction
        mov     A, #STATUS_REG
        lcall   outbyt                  ; Send status register
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        lcall   wip_poll                ; Poll for completion of write cycle
        ret
;***************************
;* Name:  RDSR_CMD
;* Description: Read Status Register
;* Function: This routine sends the command to read the status register
;* Calls: outbyt, inbyt
;* Input: None
;* Outputs: A = status register
;* Register Usage: A
;***************************
rdsr_cmd:
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #RDSR_INST
        lcall   outbyt                  ; Send RDSR instruction
        lcall   inbyt                   ; Read status register
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        ret
;***************************
;* Name:  BYTE_WRITE
;* Description: Single Byte Write
;* Function: This routine sends the command to write a single byte to the EEPROM memory array
;* Calls: outbyt, wip_poll
;* Input: None
;* Outputs: None
;* Register Usage: A
;***************************
byte_write:
        mov     DPTR, #BYTE_ADDR        ; Set address of byte to be written
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #WRITE_INST
        lcall   outbyt                  ; Send WRITE instruction
        mov     A, DPH
        lcall   outbyt                  ; Send high order address byte
        mov     A, DPL
        lcall   outbyt                  ; Send low order address byte
        mov     A, #BYTE_DATA
        lcall   outbyt                  ; Send data byte
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        lcall   wip_poll                ; Poll for completion of write cycle
        ret
```

```
;****************************
;* Name:  BYTE_READ
;* Description: Single Byte Read
;* Function: This routine sends the command to read a single byte from the EEPROM memory array
;* Calls: outbyt, inbyt
;* Input: None
;* Outputs: A = read byte
;* Register Usage: A
;****************************
byte_read:
         mov     DPTR, #BYTE_ADDR      ; Set address of byte to be read
         clr     sck                   ; Bring SCK low
         clr     cs                    ; Bring /CS low
         mov     A, #READ_INST
         lcall   outbyt                ; Send READ instruction
         mov     A, DPH
         lcall   outbyt                ; Send high order address byte
         mov     A, DPL
         lcall   outbyt                ; Send low order address byte
         lcall   inbyt                 ; Read data byte
         clr     sck                   ; Bring SCK low
         setb    cs                    ; Bing /CS high
         ret
;****************************
;* Name:  PAGE_WRITE
;* Description: Page Write
;* Function: This routine sends the command to write three consecutive bytes to the EEPROM memory
;* array using page mode
;* Calls: outbyt, wip_poll
;* Input: None
;* Outputs: None
;* Register Usage: A
;****************************
page_write:
         mov     DPTR,  #PAGE_ADDR     ; Set address of 1st byte to be written
         clr     sck                   ; Bring SCK low
         clr     cs                    ; Bring /CS low
         mov     A, #WRITE_INST
         lcall   outbyt                ; Send WRITE instruction
         mov     A, DPH
         lcall   outbyt                ; Send high order address byte
         mov     A, DPL
         lcall   outbyt                ; Send low order address byte
         mov     A, #PAGE_DATA1
         lcall   outbyt                ; Send 1st data byte
         mov     A, #PAGE_DATA2
         lcall   outbyt                ; Send 2nd data byte
         mov     A, #PAGE_DATA3
         lcall   outbyt                ; Send 3rd data byte
         clr     sck                   ; Bring SCK low
         setb    cs                    ; Bring /CS high
         lcall   wip_poll              ; Poll for completion of write cycle
         ret
;****************************
;* Name:  SEQU_READ
;* Description: Sequential Read
;* Function: This routine sends the command to read three consecutive bytes from the EEPROM memory
;* array using sequential mode
;* Calls: outbyt, inbyt
;* Input: None
;* Outputs: A = last byte read
;* Register Usage: A
;***************************
sequ_read:
        mov     DPTR, #PAGE_ADDR        ; Set address of 1st byte to be read
        clr     sck                     ; Bring SCK low
        clr     cs                      ; Bring /CS low
        mov     A, #READ_INST
        lcall   outbyt                  ; Send READ instruction
        mov     A, DPH
        lcall   outbyt                  ; Send high order address byte
        mov     A, DPL
        lcall   outbyt                  ; Send low order address byte
        lcall   inbyt                   ; Read 1st data byte
        lcall   inbyt                   ; Read 2nd data byte
        lcall   inbyt                   ; Read 3rd data byte
        clr     sck                     ; Bring SCK low
        setb    cs                      ; Bring /CS high
        ret
;***************************
;* Name:  WIP_POLL
;* Description: Write-In-Progress Polling
;* Function: This routine polls for the completion of a nonvolatile write cycle by examining the WIP bit
;* of the status register
;* Calls: rdsr_cmd
;* Input: None
;* Outputs: None
;* Register Usage: R1, A
;***************************
wip_poll:
        mov     R1, #MAX_POLL           ; Set maximum number of polls
wip_poll1:
        lcall   rdsr_cmd                ; Read status register
        jnb     ACC.0, wip_poll2        ; If  WIP bit '0' write cycle completed
        djnz    R1, wip_poll1           ; If  WIP bit '1' continue polling
wip_poll2:
        ret
;***************************
;* Name:  OUTBYT
;* Description: Sends byte to EEPROM
;* Function: This routine shifts out a byte, starting with the MSB, to the EEPROM
;* Calls: None
;* Input: A = byte to be sent
;* Outputs: None
;* Register Usage: R0, A
;***************************
outbyt:
        mov     R0, #08                 ; Set bit counter to eight
outbyt1:
        clr     sck                     ; Bring SCK low
        rlc     A                       ; Shift byte left through carry
        mov     si, C                   ; Send data bit in carry
        setb    sck                     ; Bring SCK high
        djnz    R0, outbyt1             ; Finish if  last data bit
        setb    si                      ; Place SI in known condition
        ret
;***************************
;* Name:  INBYT
;* Description: Recieves byte from EEPROM
;* Function: This routine recieves a byte, MSB first, from the EEPROM
;* Calls: None
;* Input: None
;* Outputs: A = recieved byte
;* Register Usage: R0, A
;***************************
inbyt:
        mov     R0, #08                 ; Set bit counter to eight
inbyt1:
        setb    sck                     ; Bring SCK high
        clr     sck                     ; Bring SCK low
        mov     C, so                   ; Receive data bit and store in carry
        rlc     A                       ; Shift byte left through carry
        djnz    R0, inbyt1              ; Finish if last data bit
        ret
END
```