

4.1.3 Definition of the EE1002 and EE1002A Serial Presence Detect (SPD) EEPROMS

1. SCOPE

This standard defines the specifications of interface parameters, signaling protocols, and features for Serial Presence Detect (SPD) EEPROMs as used for memory module applications.

The purpose is to provide a standard for the EE1002 and EE1002A SPD logic devices for uniformity, multiplicity of sources, elimination of confusion, ease of device specification, and ease of use.

NOTE: The designation EE1002 refers to the part of the part designation of a series of commercial logic parts common in the industry. This number is normally preceded by a series of manufacturer specific characters to make up a complete part designation.

2. DEVICE STANDARD

2.1 Description

The EE1002 is a 256-byte EEPROM device is designed for a range of supply voltages ranging from 1.8 V to 3.3 V nominal. The EE1002A is a similar device with a more restricted supply voltage of 3.3 V nominal only but with higher peak performance characteristics. Both devices are intended to interface to I²C buses which have multiple devices on a shared bus, and must be uniquely addressed on this bus.

Features summary:

- Permanent and Reversible Software Data Protection for Lower 128 Bytes
- Two Wire I²C Serial Interface
- Up to 100 kHz Transfer Rate below 2.2 V
- Up to 400 kHz Transfer Rate at or above 2.2 V
- EE1002: 1.7 to 3.6 V Single Supply Voltage
- EE1002A: 3.0 to 3.6 V Single Supply Voltage
- Byte and Page Write (up to 16 Bytes)
- Random and Sequential Read modes
- Self-Timed Write Cycle
- Schmitt trigger on bus inputs
- Noise filter on bus inputs
- Automatic Address Incrementing
- Packages: PSON-8, TSSOP-8, MLP-8, SOIC-8, MSOP-8

The EE1002 is a 2 Kbit serial EEPROM memory able to lock permanently the data in its first half (from location 0x00 to 0x7F). This facility has been designed specifically for use in DRAM DIMMs (dual interline memory modules) with Serial Presence Detect. All the information concerning the DRAM module configuration (such as its access speed, its size, its organization) can be kept write protected in the first half of the memory.

The first half of the memory area can be write-protected using two different software write protection mechanisms. By sending the device a specific sequence, the first 128 bytes of the memory become write protected: permanently or resetable. In addition, the device allows the entire memory area to be write protected, using the WC# input (for example by tying this input to V_{DDSPD}).

These I²C-compatible electrically erasable programmable memory (EEPROM) devices are organized as 256x8 bits.

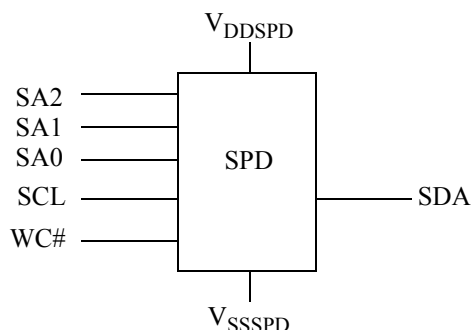


Figure 1 — Device Diagram

2.2 Device Interface

I²C uses a two wire serial interface, comprising a bi-directional data line and a clock line. The device carries a built-in 4-bit Device Type Identifier code (1010) in accordance with the I²C bus definition to access the memory area and a second Device Type Identifier Code (0110) to define the protection. These codes are used together with the voltage level applied on the three chip enable inputs (SA2, SA1, SA0).

The device behaves as a slave device in the I²C protocol, with all memory operations synchronized by the serial clock. Read and Write operations are initiated by a START condition, generated by the bus master. The START condition is followed by a Device Select Code and R/W# bit (as described in Table 2 on page 6), terminated by an acknowledge bit.

When writing data to the memory, the memory inserts an acknowledge bit during the 9th bit time, following the bus master's 8-bit transmission. When data is read by the bus master, the bus master acknowledges the receipt of the data byte in the same way. Data transfers are terminated by a STOP condition after an Ack for WRITE, and after a NoAck for READ.

EE1002 family devices shall not initiate clock stretching, which is an optional I²C bus feature.

2.3 Device Pin Definition

Table 1 — Pin Description for EE1002 and EE1002A

Pin #	Pin Name	Definition
1	SA0	Chip enable
2	SA1	Chip enable
3	SA2	Chip enable
4	V _{SSSPD}	Ground
5	SDA	Serial data
6	SCL	Serial clock
7	WC#	Write control
8	V _{DDSPD}	Supply voltage

2.4 Pin Functional Description

Serial Clock (SCL)

This input signal is used to strobe all data in and out of the device. In applications where this signal is used by slave devices to synchronize the bus to a slower clock, the bus master must have an open drain output, and a pull-up resistor can be connected from Serial Clock (SCL) to V_{DDSPD}. (Figure 2 on page 4 indicates how the value of the pull-up resistor can be calculated). In most applications, though, this method of synchronization is not employed, and so the pull-up resistor is not necessary, provided that the bus master has a push-pull (rather than open drain) output.

Serial Data (SDA)

This bi-directional signal is used to transfer data in or out of the device. It is an open drain output that may be wire-ORed with other open drain or open collector signals on the bus. A pull up resistor must be connected from Serial Data (SDA) to V_{DDSPD}. (Figure 2 on page 4 indicates how the value of the pull-up resistor can be calculated).

Chip Enable (SA0, SA1, SA2)

These input signals are used to set the value that is to be looked for on the three least significant bits (b3, b2, b1) of the 7-bit Device Select Code. In the end application, SA0, SA1 and SA2 must be directly (not through a pull-up or pull-down resistor) connected to V_{DDSPD} or V_{SSSPD} to establish the Device Select Code. When these inputs are not connected, an internal pull-down circuitry makes (SA0, SA1, SA2) = (0, 0, 0).

The SA0 input is used to detect the V_{HV} voltage, when decoding an SWP or CWP instruction. Refer to Table 2 on page 6 for decode details.

Write Control (WC#)

This input signal is provided for protecting the contents of the whole memory from inadvertent write operations. Write Control (WC#) is used to enable (when driven Low) or disable (when driven High) write instructions to the entire memory area or to the Protection Register.

When Write Control (WC#) is tied Low or left unconnected, the write protection of the first half of the memory is determined by the status of the Protection Register.

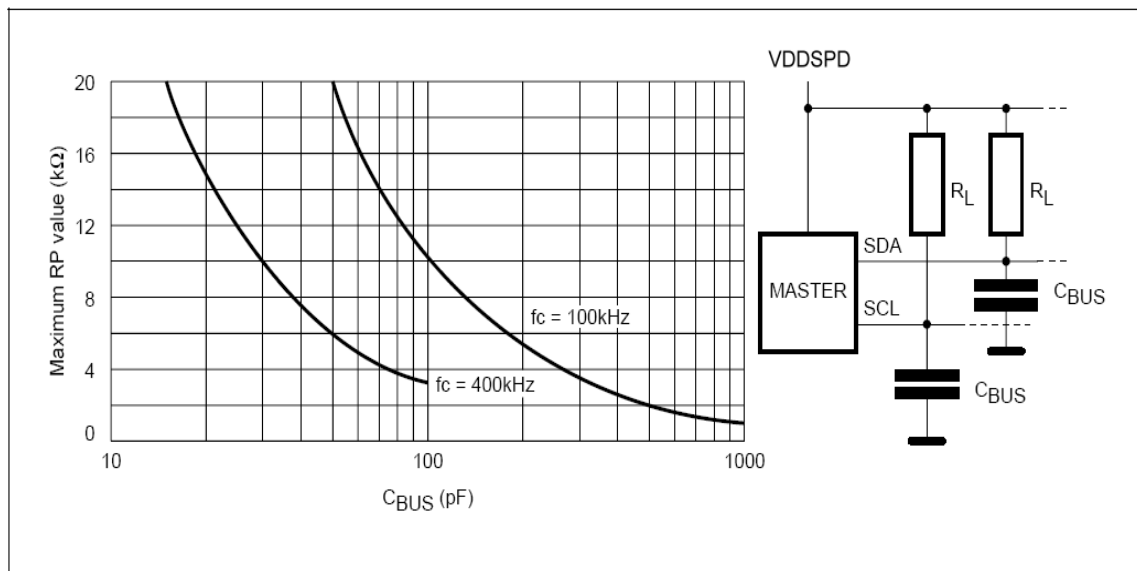


Figure 2 — Maximum R_L Value Versus Bus Capacitance (C_{BUS}) for an I^2C Bus

2.5 Device Reset and Initialization

In order to prevent inadvertent Write operations during Power-up, a Power On Reset (POR) circuit is included.

At Power-up (phase during which V_{DDSPD} is lower than $V_{DDSPDmin}$ but increases continuously), the device will not respond to any instruction until V_{DDSPD} has reached the Power On Reset threshold voltage (this threshold is lower than the minimum V_{DDSPD} operating voltage defined in the DC AND AC PARAMETERS tables). Once V_{DDSPD} has passed the POR threshold, the device is reset.

Prior to selecting the memory and issuing instructions, a valid and stable V_{DDSPD} voltage must be applied. This voltage must remain stable and valid until the end of the transmission of the instruction and, for a Write instruction, until the completion of the internal write cycle (t_W).

At Power-down (phase during which V_{DDSPD} decreases continuously), as soon as V_{DDSPD} drops from the normal operating voltage below the Power On Reset threshold voltage, the device stops responding to any instruction sent to it.

2.6 Device Operation

The device supports the I^2C protocol. This is summarized in Table 3 on page 5. Any device that sends data on to the bus is defined to be a transmitter, and any device that reads the data to be a receiver. The device that controls the data transfer is known as the bus master, and the other as the slave device. A data transfer can only be initiated by the bus master, which will also provide the serial clock for synchronization. The memory device is always a slave in all communication.

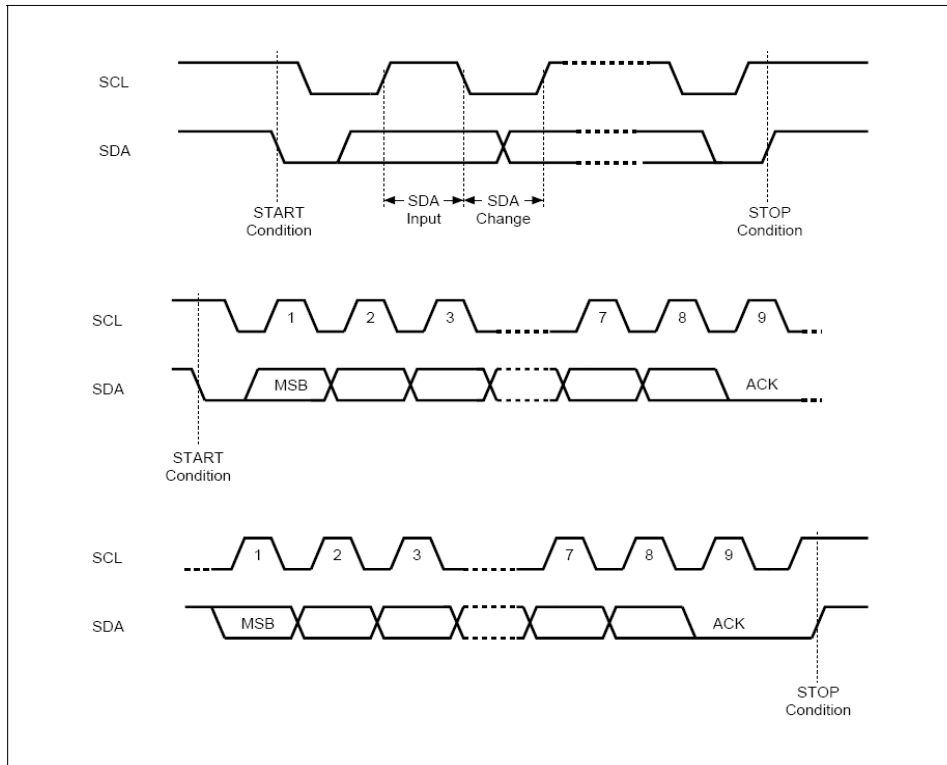


Figure 3 — I²C Bus Protocol

Start Condition

Start is identified by a falling edge of Serial Data (SDA) while Serial Clock (SCL) is stable in the High state. A Start condition must precede any data transfer command. The device continuously monitors (except during a Write cycle) Serial Data (SDA) and Serial Clock (SCL) for a Start condition, and will not respond unless one is given.

Stop Condition

Stop is identified by a rising edge of Serial Data (SDA) while Serial Clock (SCL) is stable and driven High. A Stop condition terminates communication between the device and the bus master. A Read command that is followed by NoAck can be followed by a Stop condition to force the device into the Standby mode. A Stop condition at the end of a Write command triggers the internal EEPROM Write cycle.

Acknowledge Bit (ACK)

The acknowledge bit is used to indicate a successful byte transfer. The bus transmitter, whether it be bus master or slave device, releases Serial Data (SDA) after sending eight bits of data. During the 9th clock pulse period, the receiver pulls Serial Data (SDA) Low to acknowledge the receipt of the eight data bits.

Data Input

During data input, the device samples Serial Data (SDA) on the rising edge of Serial Clock (SCL). For correct device operation, Serial Data (SDA) must be stable during the rising edge of Serial Clock (SCL), and the Serial Data (SDA) signal must change only when Serial Clock (SCL) is driven Low.

Memory Addressing

To start communication between the bus master and the slave device, the bus master must initiate a Start condition. Following this, the bus master sends the Device Select Code, shown in Table 2 on page 6 (on Serial Data (SDA), most significant bit first).

Table 2 — Device Select Code

Memory Area Select Code (two arrays) ²	Chip Enable Signals			Device Type Identifier				Chip Enable Bits			R/W#
				b7 ¹	b6	b5	b4	b3	b2	b1	b0
	SA2	SA1	SA0	1	0	1	0	SA2	SA1	SA0	R/W#
Set Write Protection (SWP)	V _{SSSPD}	V _{SSSPD}	V _{HV}	0	1	1	0	0	0	1	0
Clear Write Protection (CWP)	V _{SSSPD}	V _{DDSPD}	V _{HV}					0	1	1	0
Permanently Set Write Protec- tion (PSWP) ²	SA2	SA1	SA0					SA2	SA1	SA0	0
Read SWP	V _{SSSPD}	V _{SSSPD}	V _{HV}					0	0	1	1
Read PSWP ²	SA2	SA1	SA0					SA2	SA1	SA0	1
Note 1: The most significant bit, b7, is sent first. Note 2: SA0, SA1, and SA2 are compared against the respective external pins on the EE1002.											

The Device Select Code consists of a 4-bit Device Type Identifier, and a 3-bit Chip Enable "Address" (SA2, SA1, SA0). To address the memory array, the 4-bit Device Type Identifier is 1010b; to access the write-protection settings, it is 0110b.

Up to eight memory devices can be connected on a single I²C bus. Each one is given a unique 3-bit code on the Chip Enable (SA0, SA1, SA2) inputs. When the Device Select Code is received, the device only responds if the Chip Enable Address is the same as the value on the Chip Enable (SA0, SA1, SA2) inputs.

The 8th bit is the Read/Write bit (R/W#). This bit is set to 1 for Read and 0 for Write operations.

If a match occurs on the Device Select code, the corresponding device gives an acknowledgment on Serial Data (SDA) during the 9th bit time. If the device does not match the Device Select code, it deselects itself from the bus, and goes into Standby mode.

Table 3 — Operating Modes

Mode	R/W# Bit	WC#	Bytes	Initial Sequence
Current Address Read	1	X	1	START, Device Select, R/W# = 1
Random Address Read	0	X	1	START, Device Select, R/W# = 0, Address
	1	X		reSTART, Device Select, R/W# = 1
Sequential Read	1	X	≥ 1	Similar to Current or Random Address Read
Byte Write	0	V _{IL}	1	START, Device Select, R/W# = 0
Page Write	0	V _{IL}	≤ 16	START, Device Select, R/W# = 0
Note 1: X = V _{IL} or V _{IH} .				

2.7 Software Write Protect

The EE1002 has a hardware write-protection feature, using the Write Control (WC#) signal. This signal can be driven High or Low, and must be held constant for the whole instruction sequence. When Write Control (WC#) is held High, the whole memory array (addresses 0x00 to 0xFF) is write protected. When Write Control (WC#) is held Low, the write protection of the memory array is dependent on whether software write-protection has been set.

Software write-protection allows the bottom half of the memory area (addresses 0x00 to 0x7F) to be write protected irrespective of subsequent states of the Write Control (WC#) signal.

Software write-protection is handled by three instructions:

- SWP: Set Write Protection
- CWP: Clear Write Protection
- PSWP: Permanently Set Write Protection

The level of write-protection (set or cleared) that has been defined using these instructions, remains defined even after a power cycle.

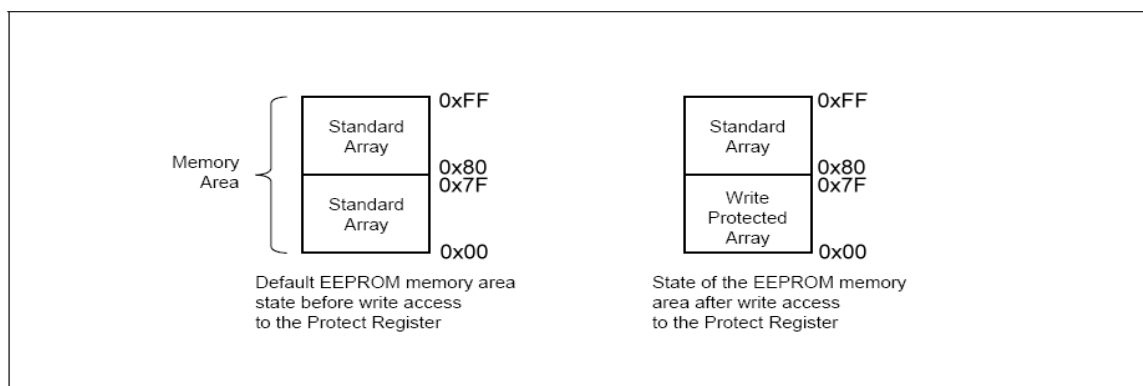


Figure 4 — Result of Setting the Write Protection

SWP and CWP

If the software write-protection has been set with the SWP instruction, it can be cleared again with a CWP instruction.

The two instructions (SWP and CWP) have the same format as a Byte Write instruction, but with a different Device Type Identifier (as shown in Table 2 on page 6). Like the Byte Write instruction, it is followed by an address byte and a data byte, but in this case the contents are all "Don't Care" (Figure 5 on page 8). Another difference is that the voltage, V_{HV} , must be applied on the SA0 pin, and specific logical levels must be applied on the other two (SA1 and SA2, as shown in Table 2 on page 6).

PSWP

If the software write-protection has been set with the PSWP instruction, the first 128 bytes of the memory are permanently write-protected. This write-protection cannot be cleared by any instruction, or by power-cycling the device, and regardless the state of Write Control (WC#). Also, once the PSWP instruction has been

successfully executed, the EE1002 no longer acknowledges any instruction (with a Device Type Identifier of 0110) to access the write-protection settings.

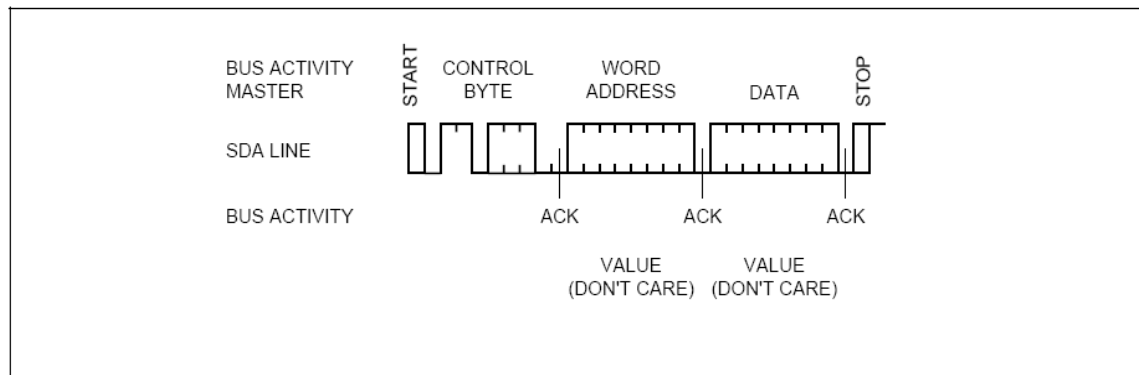


Figure 5 — Setting the Write Protection

Reading Write Protection Status

The status of software write protection can be determined using these instructions:

- Read SWP: Read Write Protection Status
- Read PSWP: Read Permanently Set Write Protection Status

Read SWP

The controller issues a Read SWP command. If Software Write Protection has not been set, the device replies to the data byte with an Ack. If Software Write Protection has been set, the device replies to the data byte with a NoAck.

Read PSWP

The controller issues a Read PSWP command. If Permanent Software Write Protection has not been set, the device replies to the data byte with an Ack. If Permanent Software Write Protection has been set, the device replies to the data byte with a NoAck.

2.8 Write Operations

Following a Start condition the bus master sends a Device Select Code with the R/W# bit reset to 0. The device acknowledges this, as shown in Table 6 on page 9, and waits for an address byte. The device responds to the address byte with an acknowledge bit, and then waits for the data byte.

When the bus master generates a Stop condition immediately after the Ack bit (in the "10th bit" time slot), either at the end of a Byte Write or a Page Write, the internal memory Write cycle is triggered. A Stop condition at any other time slot does not trigger the internal Write cycle.

During the internal Write cycle, Serial Data (SDA) and Serial Clock (SCL) are ignored, and the device does not respond to any requests.