SCENIX

SX Device
In-System Programming
Specifications

Revision History

REVISION	RELEASE DATE	SUMMARY OF CHANGES
2.0	July 15, 1999	First release as a separate stand alone document.
2.1	April 6, 2000	Include information about SX52 with no ES marking (SX 52 rev. 2.X)

Scenix Semiconductor, Inc., 1330 Charleston Road, Mountain View, CA 94043
Telephone: +1 650 210 1532, Web site: hhtp://www.scenix.com

1.0 Introduction

The SX device has a program memory consisting of 2,048 (SX28) or 4,096 (SX52) words of 12 bits per word, plus some additional 12-bit words that specify the device configuration. This memory is a non-volatile, electrically erasable (EEPROM) flash memory, rated for 10,000 rewrite cycles.

Before you can use the SX device, you must write the application code into the program memory. You do this by placing the device into a programming mode and following the protocol for accessing the program memory. You can write to the program memory only in the programming mode, not when the device is executing the application software.

1.1 Erasure and Reprogramming

When you erase the program memory, you automatically erase the entire memory, including the FUSE word, and FUSEX word registers. An erased memory has all bits set to 1. When you program the device, you clear some of the bits to 0. If you want to reprogram a memory location and clear some more bits to 0, you can "overwrite" the memory location without erasing. However, if you want to program a bit to 1 that has already been cleared to 0, the only way to do so is to erase and reprogram the whole EEPROM memory. Parts shipped to the customer are not erased. So erase the part before programming. **Specification for the SX18/20/28AC Erase Time is 500 ms (VCC = 5V, Room temp).**

1.2 In-System and Parallel Programming Modes

There are two basic device programming modes, called the "In-System Programming" (ISP) mode and the "parallel" mode. The In-System Programming mode uses just two device pins, OSC1, and OSC2, and writes the data to the device serially, one bit at a time. This mode lets you program devices that are already installed in the target system. The parallel mode uses a larger set of pins and writes data 12 bits at a time, in parallel. This mode is a little faster but can only be used to program free-standing SX parts, because it uses device I/O pins to program the part.

2.0 In-System Programming (ISP) Mode

The In-System Programming (ISP) mode lets you program or re-program an SX device that has been installed and soldered into the target system. Using the ISP mode has many advantages over traditional programming methods, in all stages of the product life: development, manufacturing, and customer service.

In the product development cycle, a separate "emulation" type device is not required. The controller device used for development is the same as the one used for final production, including the package type and pinout. The SX device can be soldered into the target system, and then programmed and reprogrammed any number of times, without removing and reinstalling it. No special socket or support circuitry is required, so the system can be debugged accurately, even in timing-sensitive and noise-sensitive applications.

For manufacturing, circuit boards can be pre-built with the controller installed and soldered on the board, even before the software has been finalized, to meet short time-to-market requirements.

Additional information such as vendor numbers and serial numbers can be programmed into the device just prior to shipment. There is no risk of stocking out-dated, pre-programmed units because the software can be corrected or updated at any time.

Even after the product is received by the customer, it can be quickly and easily revised or patched by field service personnel. Customers can even reprogram their products themselves if they have the necessary programming equipment. This equipment is relatively inexpensive and easy to use.

2.1 In-System Programming Implementation

The Scenix ISP method is a proprietary system that uses just two device pins for programming I/O: the clock input pin (OSC1) and the clock output pin (OSC2). The VDD, GND, and $\overline{\text{MCLR}}$ pins also need to be connected properly. This system eliminates the need for dedicated programming pins, thus reducing the total device pin count. There is no need for a JTAG tester, an expensive device required by some other programming systems.

OSC1 is used to supply the higher voltage necessary for programming the flash memory , while OSC2 is used to issue commands, to write data to the EEPROM, and to read data back from the EEPROM. The external programming device writes a data stream to OSC2 to specify the ISP programming operations and to supply the data written into the program memory. When the specified operation is a request to read a program memory location, the SX returns the results as a data stream on the same pin, OSC2.

The OSC1 and OSC2 pins are usually connected to passive components such as resistors, capacitors, and crystals. In typical systems, these components do not interfere with the programming signals and are not harmed by the higher voltages used for programming. In these cases, they can be left connected to the SX device during programming. It is usually not necessary to install additional hardware to isolate the ISP circuit from the rest of the system.

There are three stages to the ISP protocol:

- Entering the ISP Mode
- Programming in ISP Mode
- Exiting the ISP Mode

2.2 Entering the ISP Mode

For normal operation of the SX device, the OSC2 pin is either left unconnected, connected to passive components, or used as a clock output pin, depending on the chosen clock configuration. To put the device into the ISP mode, you pull the OSC2 pin low for at least nine consecutive clock cycles on the OSC1 pin (or nine internal clock cycles in the internal clocking mode). This action is a signal to the SX to go into the programming mode. If the clock option is not known, pull OSC2 low for 0.31 ms. This time is calculated based on the minimum clock speed (32KHz).

The exact procedure for entering the ISP mode depends on whether you are using external or internal components for normal clocking of the device. If you are using an external crystal or resonator

(including the XT, LP, or HS mode), an external RC oscillator, or an external clock signal for normal device operation, then you need to use the control signals and timing shown in Figure 2.1 to enter the programming mode. If you are using the internal RC oscillator for normal device operation, then you need to use the control signals and timing shown in Figure 2.2 to enter the programming mode.

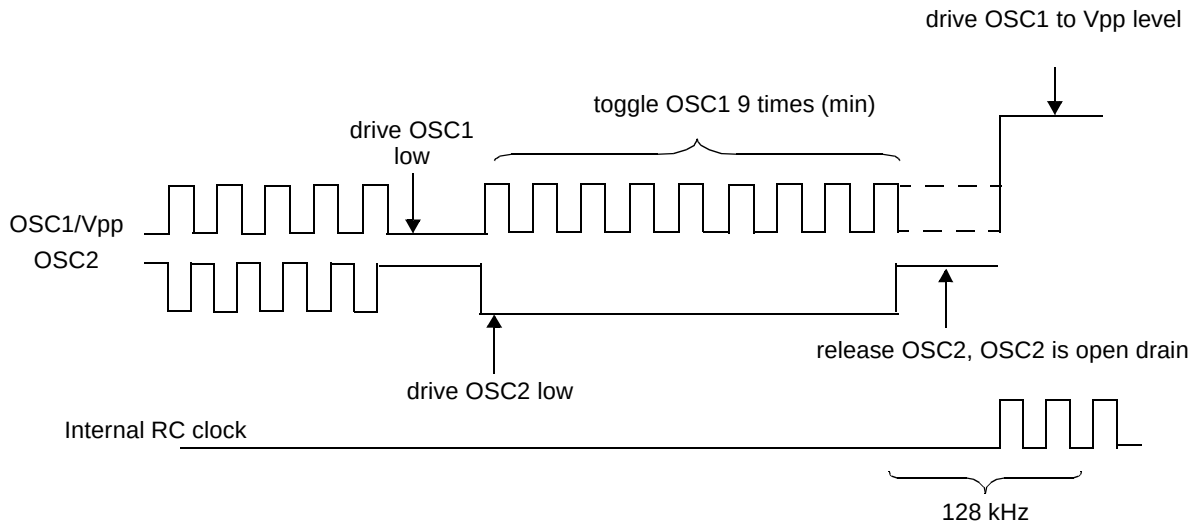

Figure 2.1 ISP Mode Entry with External Clocking

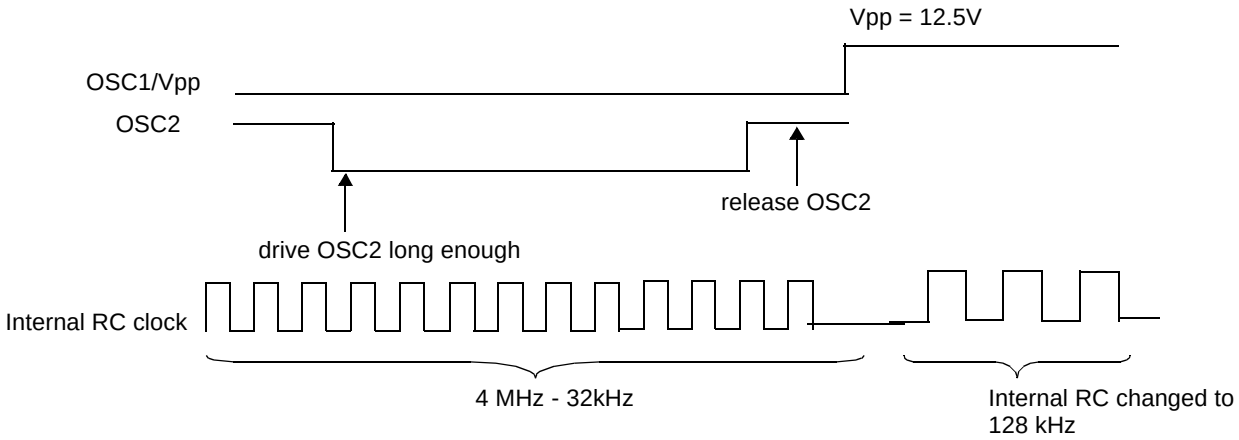

Figure 2.2 ISP Mode Entry with the Internal RC Oscillator

2.2.1 External Clocking

When the device is clocked by external components or an external clock signal, the programmer unit should use the following procedure to place the SX device in the ISP programming mode:

1. Drive the OSC1 pin low to stop the clock.
2. Drive the OSC2 pin low and toggle the OSC1 pin at least nine times. This is the signal to enter the ISP mode. OSC2 pin will be configured as open drain.
3. Release the OSC2 pin. OSC2 pin will be pull high by SX.
4. Apply the VPP programming voltage to the OSC1 pin. The SX internal RC oscillator starts operating at 128 kHz. This clock drives the SX device during ISP mode programming.

2.2.2 Internal RC Oscillator

When the device is clocked by the internal RC oscillator, the programmer unit should use the following procedure to place the SX device in the ISP programming mode:

1. Drive the OSC2 pin low for at least nine internal clock cycles. The internal clock frequency can be any one of eight values ranging from 32 kHz to 4 MHz, depending on the divide-by rate programmed into the FUSE word.
2. Release the OSC2 pin.
3. Apply the VPP programming voltage to the OSC1 pin. The SX internal RC oscillator starts operating at 128 kHz. This clock drives the SX device during ISP mode programming.

2.3 Programming in ISP Mode

Upon entry into the ISP mode, the SX device could be in the middle of executing a program, possibly with some I/O ports configured as outputs and driving other devices in the system. The first action of the ISP logic is to reset the SX device. This puts the device into a known logic state and configures the I/O ports to operate as inputs, thus preventing possible damage to other components in the system.

After the device is reset, the ISP logic executes the ISP protocol. This is a “self-aligned” serial communication protocol that uses the OSC2 pin for both synchronization and for serial I/O. No separate clock pin is needed in this protocol. The OSC2 pin is implemented with an open drain and an internal pullup, allowing it to operate as an input or output.

2.3.1 Frames, Cycles, and Internal Clocks

Communication is carried out in packets called “frames.” Each frame consists of 17 cycles, and each cycle consists of four internal clocks. The period of the internal clock is 7.81 microseconds (frequency is 128 KHz), so each cycle is 31.3 microseconds and each frame is 531 microseconds.

Figure 2.3 shows the timing of an ISP frame. The frame consists of 17 cycles. The first cycle is the “sync” cycle, used to synchronize the programmer unit to the ISP frame. This is followed by four “command” cycles, designated C3 through C0. The programmer unit drives the OSC2 pin during these

cycles to specify a programming operation such as “erase,” “read,” or “write.” The command cycles are followed by 12 “data” cycles, designated D11 through D0. During these cycles, the programmer unit drives the OSC2 pin for a “write” operation, or the SX device drives the pin for a “read” operation.

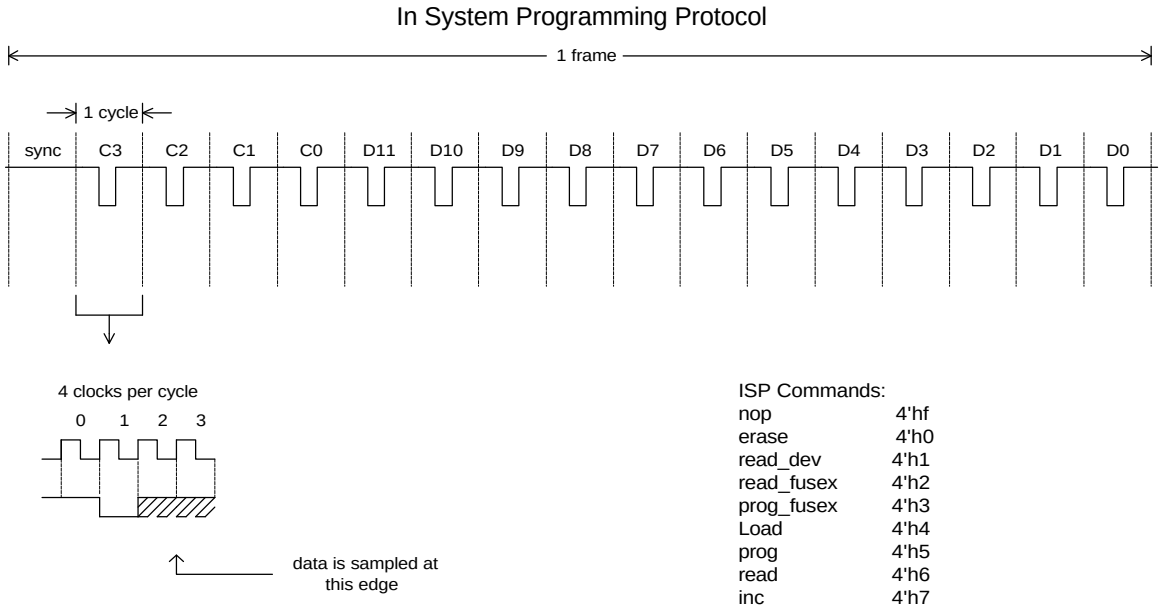

Figure 2.3 ISP Frame

Each of the 17 cycles consists of four internal clock periods.

In the first clock period, nothing drives the OSC2 pin, so the pin is pulled high by an internal pullup resistor.

In the second clock period, the SX device drives the OSC2 pin low. This is the synchronization pulse. The external programming unit uses the leading edge of this pulse to synchronize itself to the SX device. The pulse is omitted in the sync cycle (the first of 17 cycles in a frame) so that the programming unit can determine where the frame starts.

In the third and fourth clock periods, the programmer unit writes a data bit to the SX device or reads a data bit from the SX device, depending on the cycle and the type of command issued. The data bit is placed on the OSC2 pin during these two clock periods, either by the programmer unit or by the SX device, and then sampled by on the rising edge of the fourth clock period.

In the four command cycles (C3-C0), the programmer writes a four-bit command to the ISP logic, which tells the ISP logic what to do during the data cycles. In the 12 data cycles (D11-D0), for a “write” operation, the programmer writes the 12 bits that are to be written to a memory location. For a “read” operation, the programmer reads 12 bits supplied by the SX device from a memory location.

2.4 Internal Hardware

Figure 2.4 is a simplified block diagram of the chip-internal ISP hardware.

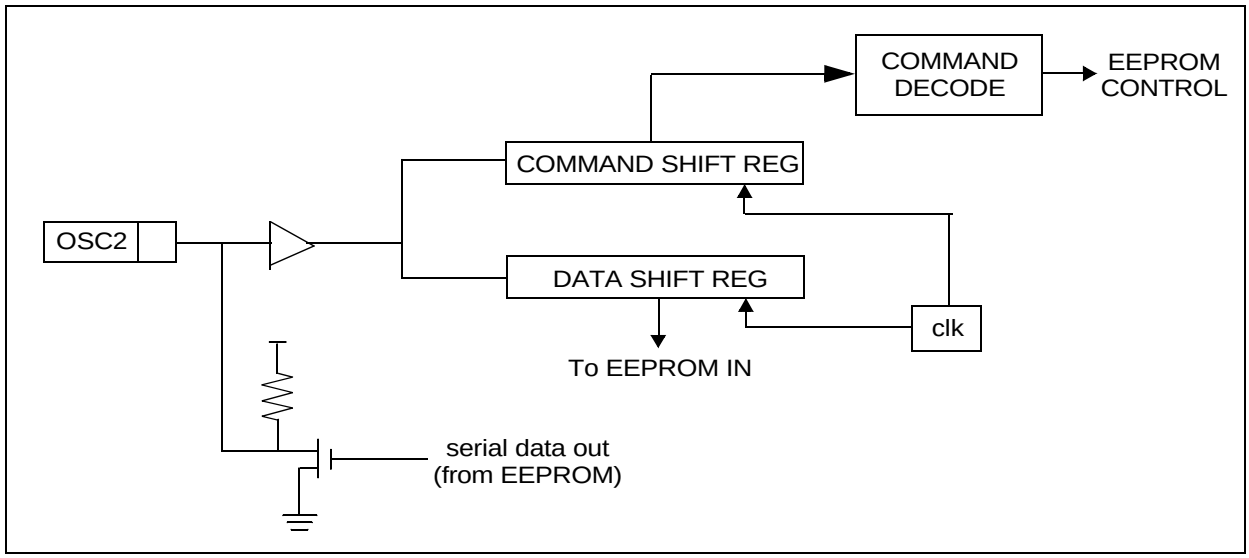

Figure 2.4 ISP Circuit Block Diagram

Serial data written to the OSC2 pin is shifted into the command shift register or data shift register, depending on whether command bits or data bits are being processed within a frame. Command bits are decoded and used to control the flash EEPROM block, while data bits are written to the flash EEPROM.

When the command is to read data from the program memory, the data bits are read from the EEPROM block and shifted out on the OSC2 pin during the data cycles. An open-drain transistor and a pullup resistor pull the OSC2 pin low or high for each bit. This same transistor is used to pull the OSC2 pin low during the second clock within each cycle (except in the sync cycle).

2.5 ISP Commands

The programmer unit writes a 4-bit command during the four command cycles at the beginning of each frame, just after the sync cycle. This 4-bit command tells the ISP logic what to do during the remaining 12 cycles of the frame. Table 2.1 lists and describes the programming commands. Codes not listed in the table are reserved for future expansion.

Table 2.1 ISP Commands

Name	Code	Description
Erase	0000	Erase all EEPROM locations.
Read DEVICE Word	0001	Read DEVICE word (memory size configuration).
Read FUSEX Word	0010	Read FUSEX word (configuration options)
Program FUSEX Word	0011	Program FUSEX word (configuration options)
Load Data	0100	Load data word to be programmed into memory.
Program Data	0101	Program previously loaded data word into memory.
Read Data	0110	Read data word from memory.
Increment Address	0111	Increment the program memory pointer by one.
NOP	1111	No operation.

The commands that erase or program the EEPROM registers must be repeated consecutively for a certain frames in order to work reliably. To determine the minimum required number of repetitions of a command, look in the Electrical Characterization section of the device data sheet and find the minimum time requirement for the operation. Divide this value by the frame period, 0.53 milliseconds, and round up to the nearest whole number.

For example, if you find that the minimum time requirement for an “Erase” operation is 100 ms, divide 100 by 0.53 and round up, and the result is 189. This means that you must repeat the “Erase” command for at least 189 commands in order to complete the “Erase” operation reliably. “NOPs” can be used in-between these 189 erase commands.

No repetition is necessary to read a register or to increment the memory address pointer. You can complete one of these operations in just a single frame.

2.5.1 NOP Command

The NOP (no-operation) command causes the ISP logic to do nothing and wait for the next command. The NOP command has a code of 1111 binary. Whenever the programmer unit is not driving to the OSC2 pin, the internal pullup resistor pulls the pin high, which produces 1111 as the command string and invokes the NOP command by default.

This is an important feature because the programmer unit needs some time to synchronize itself to the pulses generated by the ISP logic, and cannot begin driving the OSC2 until synchronization is achieved. In the meantime, the NOP command is executed by default, causing the ISP logic to wait for the first active command.

2.6 Reading the DEVICE Word

The DEVICE word is a hard-wired, read-only register containing device information such as the number of register banks and the size of the program memory and SX version number. To read the DEVICE register, the programmer unit issues the “Read DEVICE Word” command and reads the 12 bits of data ($FCE) in the data cycle portion of the frame.

2.7 Reading and Programming the FUSEX Word

The FUSEX word is a read/write register that controls device options such as carry flag operation and the brown-out reset function. Certain bits in this register are factory-set to certain values that must not be changed. Therefore, the programmer must always read these bits before erasure and reprogram them to the same values after erasure.

To read the FUSEX register, the programmer unit issues the “Read FUSEX Word” command and reads the 12 bits of data in the data cycle portion of the frame.

To program the FUSEX register, the programmer unit issues the “Load Data” command and writes the 12 bits of data in the data cycle portion of the frame. Then it issues the “Program FUSEX Word” command. This command must be repeated consecutively for a certain number of frames in order to program the register reliably, as explained earlier. **Specification for the SX18/20/28AC FUSEX programming time is 50 ms (VCC = 5V, Room temp).**

After program the FUSEX word, READ explicitly the FUSEX word to update the new value.

FUSEX word will not be updated with the new value until this READ is done.

2.8 Erasing the Memory

The “Erase” command erases all of the EEPROM memory, including the FUSE word and FUSEX word registers. The command must be repeated consecutively for a certain number of frames in order to complete the operation reliably, as described earlier.

The programmer unit should always read the FUSE and FUSEX word before erasure and restore the factory-set bits of that register after erasure.

2.9 Reading the Memory

To read the EEPROM program memory, you use two commands: “Read Data” to read the current memory location and “Increment Address” to change an internal memory address pointer from one location to the next.

Upon entry into the ISP mode, the ISP logic is set to access the FFFh or 1FFFh, which is the address of the FUSE word. The FUSE word controls many of the device configuration options such as the clocking, stack size, and Watchdog options. To read this initial memory location, the programmer unit issues the “Read Data” command and reads the 12 bits of data in the data cycle portion of the frame.

To read the word at the next address, the programmer unit issues the “Increment Address” command. This increments an internal pointer to the program memory, allowing access to address 000h. It does not matter what the programmer unit does during the 12 data cycles of the “Increment Address” frame. Following this frame, the programmer issues another “Read Data” command and reads the 12 bits of data in the data cycle portion of the frame.

This sequence is repeated to read consecutive memory locations. The first memory location is FFFh (2K device) or 1FFFh (4K device) (the FUSE word register), followed by 000h, 001h, 002h, and so on up to the top memory address, 7FFh or FFFh. The programmer can skip over any number of memory locations by repeating the “Increment Address” command consecutively, without using the “Read Data” command. The “Increment Address” command must be used 2,048 or 4,096 times to traverse the whole program memory and 4096 (2K device) or 8192 (4K device) times for wrap around.

2.10 Programming the Memory

To program the EEPROM program memory, you use three commands: “Load Data” to load the a word to be written to a memory location, “Program Data” to write the word into memory, and “Increment Address” to change the memory address pointer from one location to the next.

Upon entry into the ISP mode, the ISP logic is initially set to access the FUSE word. To program this memory location, the programmer unit issues the “Load Data” command and writes the 12 bits of data in the data cycle portion of the frame. Then it issues a “Program Data” command to write the loaded word. It does not matter what the programmer unit does during the 12 data cycles of the “Program Data” frame. This command must be repeated a certain number of times in order to program the register reliably, as explained earlier. **Specification for the SX28 EEPROM program memory is 20ms (VCC = 5V, Room temp).**

After program the FUSE word, READ explicitly the FUSE word to update the new value.

FUSE word will not be updated with the new value until this READ is done.

To program the word at the next address, the programmer unit issues the “Increment Address” command, which increments the internal pointer to access the words at address 000h. Following the “Increment Address” frame, the programmer issues another “Load Data” command and writes the 12 bits of data in the data cycle portion of the frame. Then it issues the “Program Data” command consecutively for a certain number of frames.

This sequence is repeated to program consecutive memory locations. The first memory location is FFFh (2K device) or 1FFFh (4K device) (the FUSE word register), followed by 000h, 001h, 002h, and

so on up to the top memory address, 7FFh or FFFh. The programmer can skip over any number of memory locations by repeating the “Increment Address” command consecutively.

2.11 Trimming the Internal RC Clock

Bits 11, 9 and 8 of FUSEX word are used to calibrate the internal clock. After erase, these bits become 1 and the internal RC clock is trimmed to the slowest speed. It is always advisable to change those trimming bits by one count so that the internal clock doesn’t change speed drastically, which may cause SX out of synchronization with the programmer. The internal clock will be running at 128KHz during ISP, if the internal clock is trimmed at 4MHz.

2.12 Programming the Customer ID

There are 16 12-bit words location available for programming the customer ID. These ID words are located after the program memory space. If the size of the program memory is 2K, then the last address of the program memory is $7FF and the addresses of the customer ID words are from $800 to $80F. If only the first page (512 words) is used for the program memory, then the last address of the program memory is $1FF and the addresses of the ID words are from $200 to $20F, and so on. These words outside of the program memory are only accessible during the programming mode.

2.13 Programming the PINS Bit

Bit 10 in the FUSEX word is used to identify the number of pins in the package. For the 28 pin device, this bit should be programmed to 1. For the 18/20 pins package, this bit should be programmed to 0. The chip will not be programmed, if this bit is not programmed correctly first.

2.14 Exiting the ISP Mode

Exiting from the ISP mode must be done according to the following protocol to prevent possible damage to system components:

1. The programmer drops the voltage on the OSC1 pin from the programming voltage to logic zero. This is a signal to exit from the ISP mode.
2. On the next rising clock edge after the sync cycle, the SX device exits from the ISP mode and generates an internal reset signal that resets the device. The programmer must observe the same protocol until the end of this step.
3. The programmer releases the OSC1 pin, allowing the SX device to begin normal operation.

Table 2.2 Differences of the Programming Spec. between the SX parts (Preliminary)

Parameter	SX28 (New Rev., Date Code Axyywwxx)	SX28 (Old Rev., Date Code yywwxx)	SX 52 (With ES Marking, Date Code Ax99wwxx)	SX 52 (New Rev., Date Code Ax00wwxx)
Program Time	Program Memory = 20 ms; FUSEX word = 50 ms	Program Memory = 100 ms; FUSEX word = 250 ms	Program Memory = 100 ms; FUSEX word = 250 ms	Program Memory = 20 ms; FUSEX word = 50 ms
Erase Time	500 ms	500 ms	500 ms	500 ms
VDD	5V (10% tolerance)	5V (10% tolerance)	5V (10% tolerance)	5V (10% tolerance)
VPP	12.5 (5% tolerance)	12.5 (3% tolerance)	12.5 (3% tolerance)	12.5 (5% tolerance)
Max Current at VPP pin	10 ma	15 ma	15 ma	15 ma
Temp	Room	Room	Room	Room
Device Word Reads	$FCE	$FDE (Rev. 4.1); $FEE (Rev. 2.5)	$001	$002
FUSE & FUSEX Words	After PROGRAM those words, READ explicitly those words to update the new value	New value gets updated after PROGRAM those words	New value gets updated after PROGRAM those words	After PROGRAM those words, READ explicitly those words to update the new value
$\overline{\text{MCLR}}$	Doesn't require the 100 ohm resitor between the $\overline{\text{MCLR}}$ pin and the VPP	100 ohm resistor between the $\overline{\text{MCLR}}$ pin and the VPP required	Doesn't require the100 ohm resistor between the $\overline{\text{MCLR}}$ pin and the VPP	Doesn't require the100 ohm resistor between the $\overline{\text{MCLR}}$ pin and the VPP

CAUTION:

1. Voltage spikes above 13.5V on Vpp can cause permanent damage to the SX.
2. Voltage spikes above 7.5V on Vdd can cause permanent damage to the SX.
3. Not meeting the minimum Program and Erase time and minimum Vdd, Vpp, can cause run mode failures, especially at low voltage, high frequency, high temp.
4. To program the low voltage part (Vdd = 3V) in system, please contact Scenix for more information about programming and erase time. We are currently doing the low voltage programming characterization. Remember, some of the in-system programming tools don't run at 3V. In that case, you may need to provide separate power supply to the tool.
5. Parts shipped out of factory are not erased.

Lit#: SXL-MS01-01