8XC51FX AND 8XC51

SPECIFICATION UPDATE

Release Date: July, 1996

Order Number: 272871-001

The 8XC51FX and 8XC51 may contain design defects or errors known as errata. Characterized errata that may cause the 8XC51FX and 8XC51's behavior to deviate from published specifications are documented in this specification update.

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
07/01/96	001	This is the new Specification Update document. It contains identified errata published prior to this date.

PREFACE

As of July, 1996, Intel's Semiconductor Products Group has consolidated available historical device and documentation errata into this new document type called the Specification Update. We have endeavored to include all documented errata in the consolidation process, however, we make no representations or warranties concerning the completeness of the Specification Update.

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This is the first release of the 8XC51FX and 8XC51 Specification Update. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Title	Order
Embedded Microcontrollers	270646-007
MCS®-51 Microcontroller Family User's Manual	272383-001

Nomenclature

Errata are design defects or errors. These may cause the published (component, board, system) behavior to deviate from published specifications. Hardware and software designed for use with any component, board, and system must consider all errata documented.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in any new release of the specification.

NOTE:

Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

SUMMARY TABLE OF CHANGES

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to 8XC51FX and 8XC51 products. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Steps

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page):	Page location of item in this document.

Status

Doc:	Document change or update will be implemented.
Fix:	This erratum is intended to be fixed in a future step of the component.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Eval:	Plans to fix this erratum are under evaluation.

Row

|	Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Errata

No.	Steppings			Page	Status	ERRATA
	#	#	#			
						None for this revision of this specification update.

Specification Changes

No.	Steppings			Page	Status	SPECIFICATION CHANGES
	#	#	#			
						None for this revision of this specification update.

Specification Clarifications

No.	Steppings			Page	Status	SPECIFICATION CLARIFICATIONS
	#	#	#			
001				7		FX-Core ROM/EPROM Program Lock Differences
002				8		Programming Differences between FX-Core and Non FX-Core
003				11		MCS 51 FX QFP Pin 39 Change
004				11		Proper 8XC51FX Encryption Array Usage

Documentation Changes

No.	Document Revision	Page	Status	DOCUMENTATION CHANGES
				None for this revision of this specification update.

IDENTIFICATION INFORMATION

Markings

Marked with identifier “A” at the end of FPO# for FX core material

Example: LxxxxxxxA

ERRATA

None for this revision of this specification update.

SPECIFICATION CHANGES

None for this revision of this specification update.

SPECIFICATION CLARIFICATIONS

001. FX-Core ROM/EPROM Program Lock Differences

PROBLEM: There have been a number of questions regarding the differences between ROM and EPROM program lock schemes on FX-core devices.

All ROM/EPROM products have one program lock bit and a 64-byte encryption array. Line two of the table below indicates the degree of code protection available with the bit enabled. The features on lines three and four are NOT available on the ROM/EPROM devices.

To invoke the lock features, the customer must submit the encryption table along with his code, and the lock bit is programmed in the factory. The following file naming convention should be used:

Program Code - filename.HEX Encryption Array - filename.KEY

The encryption array must be 64 bytes. If a customer submits an encryption file, the lock bit will automatically be programmed by the factory. The lock bit feature is not available without an encryption array file.

All EPROM/OTP devices have three program lock bits and a 64-byte encryption array. The table below describes the security features of each bit. Any or all of the features can be enabled by programming the appropriate bit. The customer is responsible for programming the array as well as the lock bits to invoke the protection desired. Erasing the EPROM also erases the encryption array and the program lock bits, returning the part to full functionality.

Program Lock Bits				Protection	Type
	LB1	LB2	LB3		
1	U	U	U	No program lock features enabled. (Code verify will still be encrypted by the encryption array if programmed.)	
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the EPROM is disabled.	
3	P	P	U	Same as 2, also verify is disabled.	
4	P	P	P	Same as 3, also external execution is disabled	

002. Programming Differences between FX-Core and Non FX-Core

PROBLEM: From a programming perspective, FX-core products are incompatible with non FX-core products. There are two reasons for this. First, we wanted to take advantage of the FX-cores advanced 1 micron process technology's ability to program EPROM cells faster. This process allows the floating gate of an EPROM cell to be charged to the same potential in one fifth the time. Our customers can reduce the programming time while maintaining the same quality level. Second, we increased the protection of the EPROM array by doubling the encryption array and adding a third lockbit. Below is a summary of differences between FX-core and non FX-core products.

1. Extended Encryption Array
   The FX-core has a 64-byte encryption array (0-3FH); the non-FX has 32 byte encryption array (0-1FH).
2. Additional Lockbit
   The FX-core has 3 program lockbits; the non-FX has only two. When the third bit is programmed, external execution is disabled.
3. Modified Quick Pulse
   The FX-core can be programmed with only five 100-us pulses; the non-FX needs 25 pulses.
4. Additional Signature Byte
   The FX-core has 3 signature bytes, located at 30H, 31H, and 60H; the non-FX has only 2 signature bytes, located at 30H and 31H.
5. New Mode Pin
   The FX-core has a new control pin (P3.3); this is not required for non FX-core devices.

Quick Pulse vs. Modified Quick Pulse

The Quick Pulse programming algorithm was developed several years ago to permit EPROM programming with a lower VPP voltage (12.75 volts), and a shorter programming pulse. The Quick Pulse algorithm requires each location programmed to have the ALE/PROG# pin pulsed 25 times with each pulse being 100us in duration. This allows an Intel MCS(R) 51 controller with an 8-Kbyte EPROM to be programmed in approximately 26 seconds.

The Modified Quick Pulse algorithm is an improvement, cutting the programming time significantly. Rather than pulsing the ALE/PROG# pin 25 times for each location, it is pulsed only five times. Since the pulse width remains the same, the programming time for an Intel 8-Kbyte EPROM controller can be as low as five seconds.

The following table identifies the correct algorithm to be used for each product. A non FX-core device cannot be reliably programmed using the Modified Quick Pulse algorithm because the number of programming pulses is reduced. In addition, FX-core devices using the Modified Quick Pulse algorithm require an additional control signal to pin P3.3 as mentioned above.

MCS(R) 51 CHMOS Programming Algorithm Table

Product Name	Memory	Prgm Alg --> Quick Pulse		Modified Quick Pulse	
	Core ------->	FX	non-FX	FX	non-FX
87C51	4k		x	x	x
87C54	16k	x			
87C58	32k	x			
87C51FA	8k	x	x		
87C51FB	16k	x	x		
87C51FC	32k	x			
87C51GB	8k	x			

Program Lock Bits / Encryption Array

Program memory can be protected to varying degrees from software piracy by using the program lock bits and encryption array features.

The FX-core contains 3 lock bits rather than two. The new lock bit gives the device an added level of protection by restricting external execution. There are 3 different levels of protection available on FX-core products. When LB1, LB2, and LB3 are programmed, maximum lockbit protection is achieved. It is important to note that the user can program these lockbits.

The encryption array has also been expanded to 64 bytes giving added protection. The array is a 64 byte password to the ROM / EPROM; therefore, the user must know the encryption code in order to correctly read the devices memory contents. For additional information, refer to *Embedded Microcontrollers* (literature order number 270646).

The following table lists the lockbits and encryption array available on various products.

Device	Lockbits	Encrypt Array
83C51FA	None	None
83C51FB	LB1	64 Bytes
83C51FC	LB1	64 Bytes
87C51FA	LB1,2,3	64 Bytes
87C51FB	LB1,2,3	64 Bytes
87C51FC	LB1,2,3	64 Bytes

Third Party programming Support

003. MCS 51 FX QFP Pin 39 Change

PROBLEM: Non-FX devices that currently offer the QFP package show pin 39 as "NC" (no connect). Pin 39 on the FX QFP is internally tied to Vss. If pin 39 of the FX device is connected to anything other than Vss, problems will occur. For this reason we strongly recommend that no connections be made to any pin labeled "NC".

Several 51 products have converted to the FX core. All FX-core devices are available in the "S" (QFP) package. Customers planning to use FX QFP devices in non-FX designs should pay special attention to the pin 39 change.

The following is a list of FX-core products to date:

87C51	80C54	87C51FA
80C32	87C54	83C51FB
80C52	80C58	87C51FB
87C52	87C58	83C51FC
		87C51FC

The products below are available in a QFP package but are non-FX:

80C31BH
80C51BH
80C51FA
83C51FA

004. Proper 8XC51FX Encryption Array Usage

PROBLEM: Customers using the program lock features on FX devices should not use the same value for all bytes in the Encryption Array, or leave large blocks of code memory unprogrammed.

To enable FX ROM (or QROM) lock features, customers must submit a 64-byte encryption file along with their program code. The factory then programs the device encryption array and sets Lock Bit 1 before the device is shipped. Those customers using EPROM devices must program the encryption array themselves.

There is one important factor that needs to be considered before using the encryption array as a means for program protection.

The encryption feature exclusively NORs each code byte with one of the encryption bytes. All unprogrammed encryption bytes have the value OFFH. If the encryption array is left unprogrammed, any code byte XNORed with OFFH leaves the byte unchanged. Similarly, if a code byte has the value OFFH, verification of the byte will produce the encryption byte value. If a large block (>64 bytes) of code is left unprogrammed (an unprogrammed byte has value OFFH), a verification routine will, essentially, display the contents of the encryption array.

For this reason, it is strongly recommended that all unused code bytes be programmed with some value other than OFFH, and not all of them the same value. This practice will ensure the maximum possible program protection for this feature.

DOCUMENTATION CHANGES

None for this revision of this specification update.