8XC196NT
SPECIFICATION UPDATE

Release Date: October, 1996

Order Number 272865-002

The 8XC196NT may contain design defects or errors known as errata. Characterized errata that may cause the 8XC196NT's behavior to deviate from published specifications are documented in this specification update.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call in North America 1-800-879-4683, Europe 44-0-1793-431-155, France 44-0-1793-421-777,

Germany 44-0-1793-421-333 other Countries 708-296-9333

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
07/01/96	001	This is the new Specification Update document. It contains all identified errata published prior to this date.
09/24/96	002	Added documentation change number 001 (I_{CC}, I_{IDLE} versus Frequency)

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 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 additional information that was not previously published.

Affected Documents/Related Documents

Title	Order
8XC196NT CHMOS Microcontroller datasheet	272267-004
8XC196NT Microcontroller User's Manual	272317-003

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 to be used 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 the 8XC196NT product. 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
	D	#	#			
9600001	X			6	Fix	Illegal Opcode Interrupt Vector in 1-Mbyte Mode
9600002	X			6	Fix	Aborted Interrupt Vectors to Lowest Priority in 1-Mbyte Mode
9600003	X			7	Fix	PTS Request During Interrupt Latency in 1-Mbyte Mode
9600004	X			8	Fix	Modes 1 and 2 Need a Latch on Upper Address Lines
9600005	X			8	Fix	Jump Instruction at 64K Page Boundary
9600006	X			8	Fix	External Memory Accesses After Idle Mode Exit in Bus-timing Modes 1 and 2

Specification Changes

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

Specification Clarifications

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

Documentation Changes

No.	Document Revision	Page	Status	DOCUMENTATION CHANGES
001	**-004**	10	Doc	Added I_{CC}, I_{IDLE} versus Frequency graph; will be added to revision -005 of the datasheet.

IDENTIFICATION INFORMATION

Markings

Bottom mark: N87C196NT (OTPROM) or N80C196NT (no nonvolatile memory)

Stepping indicator: D-step components are identified by a “D” at the end of the topside tracking number.

ERRATA

9600001. Illegal Opcode Interrupt Vector in 1-Mbyte Mode

PROBLEM: The illegal opcode interrupt should be generated when there is an attempt to execute an undefined opcode, and should vector to address FF2012H to handle the interrupt. But in 1-Mbyte mode only, the vector address for the illegal opcode interrupt is not generated correctly and a random vector address is generated.

IMPLICATION: Code that executes an illegal opcode (10h or E5h) will vector to a random address.

WORKAROUND: Use a C-compiler or assembler that will flag the Illegal opcode, or put a reset opcode (FFh) at the end of any data tables or unused memory locations.

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

9600002. Aborted Interrupt Vectors to Lowest Priority in 1-Mbyte Mode

PROBLEM: In 1-Mbyte mode only, an aborted interrupt (either intentional or unintentional) may cause an undesired branch to the lowest-priority interrupt vector (FF2000H) even if the lowest-priority interrupt is masked. This may occur if any bit in the INT_MASK, INT_MASK1, INT_PEND, or INT_PEND1 register is cleared after the corresponding INT_PEND or INT_PEND1 bit is set.

Example: If the EXTINT0 interrupt on the 8XC196NT is enabled by setting INT_MASK.3, and a rising edge on EXTINT0 occurs, INT_PEND.3 is then set. The following instruction may cause the CPU to vector to FF2000H instead of FF2006H.

```
ANDB    INT_MASK,#0F7H          ;masks EXTINT0
                                ;may cause vector to FF2000H
```

IMPLICATION: This may occur if any bit in the INT_MASK, INT_MASK1, INT_PEND, or INT_PEND1 register is cleared after the corresponding INT_PEND or INT_PEND1 bit is set. An undesired branch to the lowest-priority interrupt may occur if an interrupt is aborted, unless the workaround is used.

WORKAROUND: Use a disable interrupt (DI) instruction before clearing a bit in INT_MASK, INT_MASK1, INT_PEND, or INT_PEND1. The following code example demonstrates how to safely disable the EXTINT0 interrupt.

```
DI                              ;disable interrupts
ANDB    INT_MASK, #0F7H         ;mask EXTINT0
EI                              ;enable interrupts
```

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

9600003. PTS Request During Interrupt Latency in 1-Mbyte Mode

PROBLEM: This erratum occurs only in 1-Mbyte mode. If a standard interrupt occurs at approximately the same time (this time is code dependent and therefore cannot be stated as an exact number of state times) as a PTS serviced interrupt, the PTS interrupt may be processed as a standard interrupt. The standard interrupt service routine for a PTS-serviced interrupt (usually referred to as the End-of-PTS) is typically used to modify the PTS control block and re-enable the PTS by setting the corresponding bit in the PTSSEL register. When this anomaly occurs, the End-of-PTS service routine will execute regardless of the value in PTSCOUNT. As a result, an indeterminate number of PTS cycles will fail to occur. This applies to all interrupts serviced by the PTS.

IMPLICATION: Those customers using both the PTS and standard interrupts. If the workaround described below is not used, an indeterminate number of PTS cycles may be missed.

WORKAROUND: In the standard interrupt service routine (End-of-PTS) for each PTS enabled, the first instruction following a PUSHF or PUSHA should determine whether the associated bit in the PTSSEL register is set or cleared. Checking this bit will determine whether the desired number of cycles were completed or a premature End-of-PTS occurred. If the bit is set, the associated pending bit in the INT_PEND or INT_PEND1 should be set and a RET instruction executed. This will cause a PTS cycle to occur. If the associated bit in the PTSSEL register is cleared, the normal End-of-PTS procedure should be executed. The following is an example of an End-of-PTS service routine for the external interrupt 0 (EXTINT0).

```
CSEG AT 0FF2006H
DCW    EXTINT0_END_OF_PTS
CSEG AT 0FF3000H

EXTINT0_END_OF_PTS:

        PUSHA
        JBC       PTSSEL,3,OKAY
        ORB       INT_PEND,#08H
        POPA
        RET
        OKAY:
        ;
        ; reload PTSCOUNT
        ; re-enable PTSSEL
        ;
        POPA
        RET
```

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

9600004. Modes 1 and 2 Need a Latch on Upper Address Lines

PROBLEM: In all bus timing modes, for 16-bit bus-width operation, latch the upper and lower address/data lines. In modes 1 and 2, for 8-bit bus-width operation, also latch the upper and lower address/data lines; the upper address lines are not driven throughout the entire bus cycle. In modes 0 and 3, for 8-bit bus-width operation, latch only the lower address/data lines. In these modes, it is not necessary to latch the upper address lines because these lines are driven throughout the entire bus cycle.

IMPLICATION: If the latch is not used, the 8XC196NT could read from and write to undesired memory locations.

WORKAROUND: Add a latch to the upper address lines.

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

9600005. Jump Instruction at 64K Page Boundary

PROBLEM: Any jump, conditional jump, or call instruction located within six bytes of the top of any 64K page boundary, (e.g., 01FFFAH-01FFFFH) may cause a jump to the wrong page.

IMPLICATION: A jump to the wrong page could cause the device to reset due to an undesired code fetch.

WORKAROUND: To ensure that this problem does not occur, place at least six NOPs at the top of each page.

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

9600006. External Memory Accesses After Idle Mode Exit in Bus-timing Modes 1 and 2

PROBLEM: When operating in bus-timing modes 1 and 2, the 8XC196NT cannot access external memory immediately after exiting idle mode. Accesses to internal memory are unaffected.

IMPLICATION: Applications using external memory with bus-timing mode 1 or 2 and idle mode will be affected. Attempts to access external memory immediately upon exiting idle mode will result in corrupted data. If the access is a code fetch, the results are unpredictable.

WORKAROUND: For 80C196NT microcontrollers, which have no internal nonvolatile memory, do not use idle mode with bus-timing mode 1 or 2.

For 87C196NT microcontrollers, which have internal OTPROM, you can use idle mode with bus-timing mode 1 or 2 if you ensure that all interrupt vectors, interrupt service routines, and the stack are located in internal memory. (Interrupt vectors must reside in OTPROM; interrupt service routines must reside in either OTPROM or code RAM; and the stack must be located in either code RAM or register RAM.) Ensure that software does not access external memory until at least 10ms after the microcontroller exits idle mode.

STATUS: This erratum is intended to be fixed in a future stepping of the component. Refer to Summary Table of Changes to determine the affected stepping(s).

SPECIFICATION CHANGES

None for this revision of this specification update.

SPECIFICATION CLARIFICATIONS

None for this revision of this specification update.

DOCUMENTATION CHANGES

001. I_{CC}, I_{IDLE} versus Frequency

ITEM: The following graphic illustrating I_{CC} and I_{IDLE} versus Frequency will be added to the next revision (-005) of the 8XC196NT CHMOS Microcontroller datasheet.

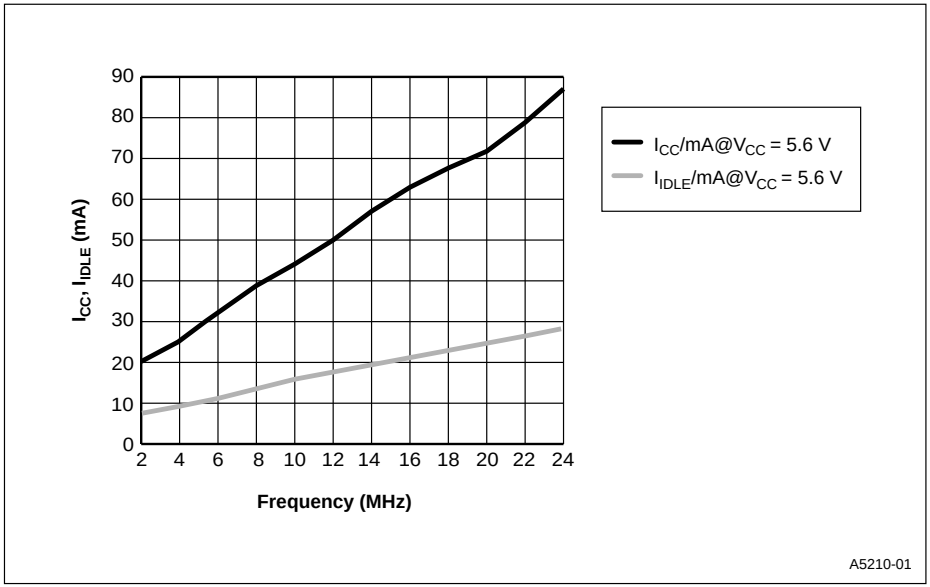

Figure 1. I_{CC}, I_{IDLE} versus Frequency