



# **87C196JT/87C196JT 20 MHz SPECIFICATION UPDATE**

Release Date: March, 1997

Order Number: 272859-003

The 87C196JT/87C196JT 20 MHz may contain design defects or errors known as errata. Characterized errata that may cause the 87C196JT/87C196JT 20 MHz's behavior to deviate from published specifications are documented in this specification update.

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel retains the right to make changes to specifications and product descriptions at any time, without notice.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

\*Third-party brands and names are the property of their respective owners.

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

Copyright © 1997, INTEL CORPORATION

**REVISION HISTORY .....1**

**PREFACE .....2**

**SUMMARY TABLE OF CHANGES .....4**

**IDENTIFICATION INFORMATION .....6**

**ERRATA .....6**

**SPECIFICATION CHANGES .....6**

**SPECIFICATION CLARIFICATIONS .....6**

**DOCUMENTATION CHANGES .....11**



**REVISION HISTORY**

| <b>Rev. Date</b> | <b>Version</b> | <b>Description</b>   |
|------------------|----------------|--|
| 07/01/96         | 001            | This is the new Specification Update document. It contains all identified errata published prior to this date. |
| 01/08/97         | 002            | Added documentation changes 001–003.   |
| 03/12/97         | 003            | B-step device identification information.  |

## 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 information that was not previously published.

### ***Affected Documents/Related Documents***

| Title  | Order      |
|--|------------|
| <i>87C196KR/KQ 87C196JV/JT 87C196JR/JQ Advanced 16-bit CHMOS Microcontroller datasheet</i> | 270827-006 |
| <i>87C196JT 20 MHz JQ Advanced 16-bit CHMOS Microcontroller datasheet</i>                  | 272529-002 |
| <i>8XC196Kx, 8XC196Jx, 87C196CA Microcontroller Family User's Manual</i>                   | 272258-002 |

### ***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 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 87C196JT/87C196JT 20 MHz 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*

#### ***Stepping***

|                              |   |
|------------------------------|---|
| X:                           | Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping. |
| (No mark)<br>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**

| Number | Steppings |   |   | Page | Status | ERRATA   |
|--------|-----------|---|---|------|--------|--|
|        | #         | # | # |      |        |  |
|        |           |   |   |      |        | None for this revision of this specification update. |

**Specification Changes**

| Number | Steppings |    |   | Page | Status | SPECIFICATION CHANGES                 |
|--------|-----------|----|---|------|--------|---------------------------------------|
|        | A#        | B# | # |      |        |                                       |
| 001    | X         | X  |   | 6    |        | DC Characteristics, Idle Mode Current |

**Specification Clarifications**

| Number | Steppings |    |   | Page | Status | SPECIFICATION CLARIFICATIONS           |
|--------|-----------|----|---|------|--------|--|
|        | A#        | B# | # |      |        |  |
| 001    | X         | X  |   | 6    |        | EPA Timer Reset/Write Conflict         |
| 002    | X         | X  |   | 6    |        | Valid Time Matches                     |
| 003    | X         | X  |   | 7    |        | P6_ PIN.4-.7 Not Updated Immediately   |
| 004    | X         | X  |   | 7    |        | Write Cycle During Reset               |
| 005    | X         | X  |   | 7    |        | Indirect Shift Count Value             |
| 006    | X         | X  |   | 7    |        | P2.7 (CLKOUT)                          |
| 007    | X         | X  |   | 8    |        | EPA Overruns                           |
| 008    | X         | X  |   | 10   |        | Indirect Addressing With AutoIncrement |

**Documentation Changes**

| Number | Document Revision | Page | Status | DOCUMENTATION CHANGES   |
|--------|-------------------|------|--------|---|
| 001    | 272258-002        | 11   | Doc    | High address inputs to the low EPROM should be A14:8 in Figure 15-18. |
| 002    | 272258-002        | 12   | Doc    | Addresses 201DH and 201FH should contain FFH in Table 4-2.            |
| 003    | 272258-002        | 13   | Doc    | In Table 7-2, P2_DIR description change.                              |

## IDENTIFICATION INFORMATION

### ***Markings***

A-step devices are identified by the letter “A” following the eight-digit FPO number on the top of the package.

B-step devices are identified by the letter “B” following the eight-digit FPO number on the top of the package.

## ERRATA

None for this revision of this specification update.

## SPECIFICATION CHANGES

### ***001. DC Characteristics, Idle Mode Current***

**PROBLEM:** The DC Characteristics Idle Mode Current specification has been revised from a maximum of 30 mA to a maximum of 32 mA.

## SPECIFICATION CLARIFICATIONS

### ***001. EPA Timer Reset/Write Conflict***

**PROBLEM:** If software writes to the EPA timer at the same time that an EPA channel resets that timer, it is indeterminate which action will take precedence. Software should not write to a timer that is being reset by EPA signals.

### ***002. Valid Time Matches***

**PROBLEM:** A timer must increment or decrement to the compare value in order for a match to occur. Loading a timer with a value that is equal to an EPA compare value does cause a match. Likewise, with an EPA compare value of 0, a timer reset does not cause a match.

**003. P6\_\_REG.4-.7 Not Updated Immediately**

**PROBLEM:** A value written to any of the upper four bits of P6\_REG is temporarily held in a buffer until the corresponding P6\_MODE bit is cleared, at which time the value is loaded into the P6\_REG bit. A value read from a P6\_REG bit is the value currently in the register, not the value in the buffer. Therefore, any change to a P6\_REG bit can be read only after the corresponding P6\_MODE bit is cleared.

**004. Write Cycle During Reset**

**PROBLEM:** If a reset occurs while the microcontroller is writing to an external memory device, the contents of the external memory device may be corrupted.

**005. Indirect Shift Count Value**

**PROBLEM:** The SHRL and SHLL instructions function correctly with count values 0–31, inclusive. However, a shift count value of XX10000B causes 32 shifts, which results in no shift taking place. With all other count values, the upper 3 bits are masked off and the remaining bits specify the number of shifts. Also, a shift count value of XX1XXXXB causes the overflow flag and the overflow-trap flag to be set.

**IMPACT:** Customers using SHRL and SHLL instructions with a count value greater than 31 will be affected.

**WORKAROUND:** Ensure that the count value never exceeds 31.

**STATUS:** Refer to Summary Table of Changes to determine the affected stepping(s).

**006. P2.7 (CLKOUT)**

**PROBLEM:** P2.7 (CLKOUT) does not operate in open drain mode.

## 007. EPA Overruns

**PROBLEM:** The EPA can lock up if overruns are handled incorrectly. Overruns occur when an EPA input transitions at a rate that cannot be handled by the EPA interrupt service routine. If no overrun handling strategy is in place, and if the following three conditions exist, a situation may occur where both the capture buffer and the EPAX\_TIME register contain data, and no EPA interrupt pending bit is set:

- an input signal with a frequency high enough to cause overruns is present on an enabled EPA pin, and
- the overwrite bit is set (EPAX\_CON.0 = 1; old data is overwritten on overrun), and
- the EPAX\_TIME register is read at the exact instant that the EPA recognizes the captured edge as valid.

The input frequency at which this occurs depends on the length of the interrupt service routine as well as other factors. Unless the interrupt service routine includes a check for overruns, this situation will remain the same until the device is reset or the EPAX\_TIME register is read. The act of reading EPAX\_TIME allows the buffered time value to be moved into EPAX\_TIME. This clears the buffer and allows another event to be captured. Remember that the act of transferring the buffer contents to the EPAX\_TIME register is what actually sets the EPAX interrupt pending bit and generates the interrupt.

**WORKAROUND:** Any one of the following methods can be used to prevent or recover from an EPA overrun situation.

- Clear EPAX\_CON.0

When the overwrite bit (EPAX\_CON.0) is zero and both the EPAX\_TIME register than the buffer are full, the EPA does not consider a captured edge until the EPAX\_TIME register is read and the data in the capture buffer is transferred to EPAX\_TIME. This prevents overruns by ignoring new input capture events when both the capture buffer and EPAX\_TIME contain valid capture times. The OVRx pending bit in EPA\_PEND is set to indicate that an overrun occurred.

- Enable the OVRx interrupt and read the EPAX\_TIME register within the ISR

If an overrun occurs, the overrun (OVRx) interrupt will be generated. The OVRx interrupt will then be acknowledged and its interrupt service routine will read the EPAX\_TIME register. After the CPU reads the EPAX\_TIME register, the buffered data moves from the buffer to the EPAX\_TIME register. This sets the EPA interrupt pending bit.

- Check for pending EPAX interrupts before exiting an EPAX ISR

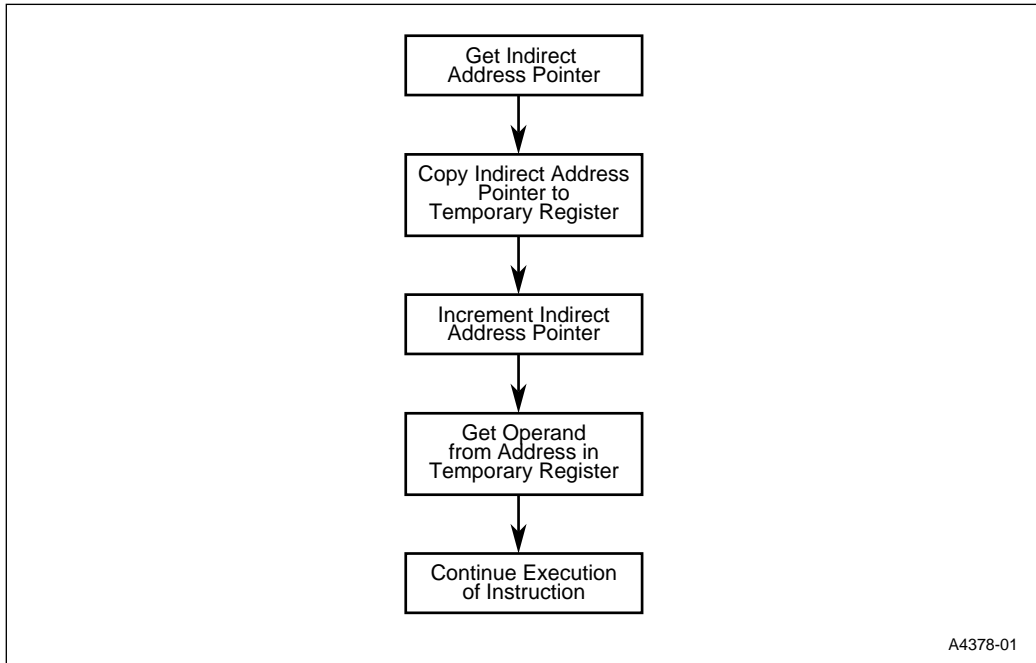
Another method for avoiding this situation is to check for pending EPA interrupts before exiting the EPA interrupt service routine. This is an easy way to detect overruns and additional interrupts. It can also save loop time by eliminating the latency necessary to service the pending interrupt. However, this method cannot be used with the peripheral transaction server (PTS).

**STATUS:** Refer to Summary Table of Changes to determine the affected stepping(s).

**008. Indirect Addressing With AutoIncrement**

**PROBLEM:** For indirect addressing with autoincrement, a pointer that points to itself results in an access to the incremented pointer address rather than the original pointer address.

The CPU stores the pointer's value in a temporary register, increments the pointer, then accesses the operand at the address contained in the temporary register, as shown in this flowchart.



Therefore, if the pointer points to itself, the CPU accesses the operand at the incremented address contained in the pointer.

For example, assume  $ax = 1CH$  and  $bx = 40H$ . The following code causes the CPU to access the operand at the incremented address:

```
ld    ax,#ax
ldb   bx,[ax]+

ld    1CH,#1CH;1CH←1CH;load location 1CH with value 1CH
ldb   40H,[1CH]+;temp←1CH;save 1CH into temp register
      ;1CH←1DH;increment the contents of 1CH
      ;40H←1DH;load the contents of location 1CH
      ;(location 1CH
      ;now contains the value 1DH) into 40H
```

**WORKAROUND:** Avoid using an indirect address pointer that points to itself.

For example, assume  $ax = 1CH$ ,  $bx = 1DH$ , and  $cx = 40H$ . The following code causes the CPU to access the operand at the intended, unincremented address:

```
ld    ax,#bx      ;where bx≠ax
ldb   cx,[ax]+

ld    1CH,#1DH;1CH←1DH;load location 1CH with value 1DH
ldb   40H,[1CH]+;temp←1DH;save 1DH (contents of 1CH) into temp
      ;register
      ;1CH←1EH;increment the contents of 1CH
      ;40H←1DH;load the contents of temp into 40H
```

**STATUS:** Refer to Summary Table of Changes to determine the affected stepping(s).

## DOCUMENTATION CHANGES

- 001.** *High address inputs to the low EPROM should be A14:8 in Figure 15-18.*

**ITEM:** The next revision of the user's manual will include the following corrected graphic.

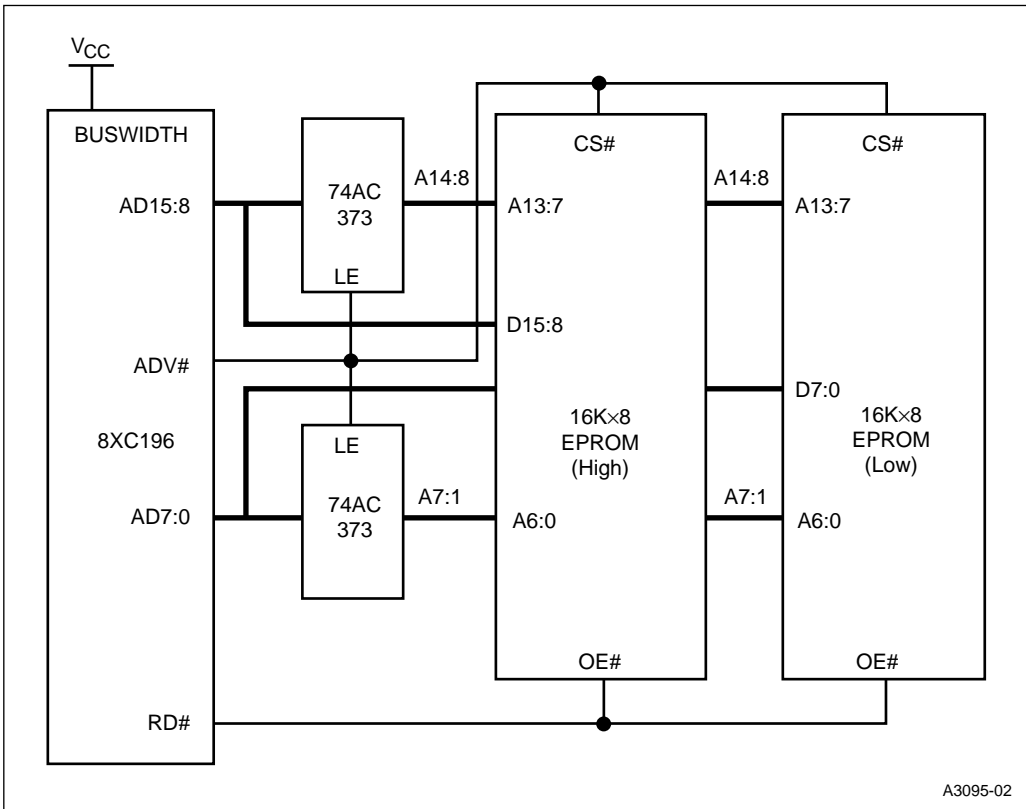


Figure 1. 16-bit System with EPROM

**002.** *Addresses 201DH and 201FH should contain FFH in Table 4-2.*

**ITEM:** The next revision of the user's manual will include the following corrected table.



**Table 0-1. Special-purpose Memory Addresses**

| Hex Address  | Description                             |
|--------------|---|
| 207F<br>205E | Reserved (each byte must contain FFH)   |
| 205D<br>2040 | PTS vectors                             |
| 203F<br>2030 | Upper interrupt vectors                 |
| 202F<br>2020 | Security key                            |
| 201F         | Reserved (must contain FFH)             |
| 201E         | Reserved (must contain FFH)             |
| 201D         | Reserved (must contain FFH)             |
| 201C         | Reserved (must contain FFH)             |
| 201B         | Reserved (must contain 20H)             |
| 201A         | CCB1                                    |
| 2019         | Reserved (must contain 20H)             |
| 2018         | CCB0                                    |
| 2017<br>2016 | OFD flag (see page 13-12 and page 16-8) |
| 2015<br>2014 | Reserved (each byte must contain FFH)   |
| 2013<br>2000 | Lower interrupt vectors                 |

**003. In Table 7-2, P2\_DIR description change.**

**ITEM:** The next revision of the user's manual will include the following corrected register description.

| Mnemonic | Address | Description   |
|----------|---------|---|
| P2_DIR   | 1FCBH   | Port Direction Register<br>Each bit controls the configuration of the corresponding pin. Clearing a bit configures the corresponding pin as a complementary output; setting a bit configures the corresponding pin as an open-drain output or a high-impedance input. |