87C196KT Microcontroller

Specification Update

November, 1998

Notice: The 87C196KT microcontroller may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this specification update.

Order Number: 272862-004
## Contents

Revision History ........................................................................................................ 5
Preface ..................................................................................................................... 6
Summary Table of Changes .................................................................................... 8
Identification Information ..................................................................................... 10
Errata ..................................................................................................................... 11
Specification Changes ......................................................................................... 12
Specification Clarifications ................................................................................... 13
Documentation Changes ....................................................................................... 17
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/01/96</td>
<td>001</td>
<td>This is the new Specification Update document. It contains all identified errata published prior to this date.</td>
</tr>
<tr>
<td>01/08/96</td>
<td>002</td>
<td>Added documentation change #1–3.</td>
</tr>
<tr>
<td>04/07/97</td>
<td>003</td>
<td>Added errata #1.</td>
</tr>
<tr>
<td>11/11/98</td>
<td>004</td>
<td>Added errata #2.</td>
</tr>
</tbody>
</table>
As of July, 1996, Intel's Computing Enhancement 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**

<table>
<thead>
<tr>
<th>Title</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>87C196KT Advanced 16-Bit CHMOS Microcontroller Automotive datasheet</td>
<td>273178</td>
</tr>
<tr>
<td>8XC196Kx, 8XC196Jx, 87C196CA Microcontroller Family User’s Manual</td>
<td>272258</td>
</tr>
</tbody>
</table>
Preface

Nomenclature

Errata are design defects or errors. These may cause the Product Name’s behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

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.).
The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the Product Name 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) 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

<table>
<thead>
<tr>
<th>No.</th>
<th>Steppings</th>
<th>Page</th>
<th>Status</th>
<th>ERRATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X X X X</td>
<td>11</td>
<td>NoFix</td>
<td>Executing Routines in the User's ROM While the Device is Operating in Serial Programming Mode</td>
</tr>
<tr>
<td>2</td>
<td>X X X X</td>
<td>11</td>
<td>NoFix</td>
<td>A/D Conversion Error on First Conversion</td>
</tr>
</tbody>
</table>

### Specification Changes

None for this revision of the specification update.

### Specification Clarifications

<table>
<thead>
<tr>
<th>No.</th>
<th>Steppings</th>
<th>Page</th>
<th>Status</th>
<th>SPECIFICATION CLARIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>13</td>
<td></td>
<td>EPA Timer Reset/Write Conflict</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>13</td>
<td></td>
<td>Valid Time Matches</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>13</td>
<td></td>
<td>P6_REG.4-.7 Not Updated Immediately</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>13</td>
<td></td>
<td>Write Cycle During Reset</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>13</td>
<td></td>
<td>Indirect Shift Count Value</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>13</td>
<td></td>
<td>Port 4 Address Behavior</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>13</td>
<td></td>
<td>EPA Overruns</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>15</td>
<td></td>
<td>Indirect Addressing With AutoIncrement</td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>16</td>
<td></td>
<td>CLKOUT During RESET</td>
</tr>
</tbody>
</table>

### Documentation Changes

<table>
<thead>
<tr>
<th>No.</th>
<th>Document Revision</th>
<th>Page</th>
<th>Status</th>
<th>DOCUMENTATION CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>272258-002</td>
<td>17</td>
<td>Doc</td>
<td>High address inputs to the low EPROM should be A14:8</td>
</tr>
<tr>
<td>2</td>
<td>272258-002</td>
<td>18</td>
<td>Doc</td>
<td>Addresses 201DH and 201FH should contain FFH</td>
</tr>
<tr>
<td>3</td>
<td>272258-002</td>
<td>18</td>
<td>Doc</td>
<td>P2_DIR description change</td>
</tr>
</tbody>
</table>
Identification Information

Markings

C-step devices are identified by the letter “C” following the eight-digit FPO number.
**Errata**

1. **Executing Routines in the User’s ROM While the Device is Operating in Serial Programming Mode**
   
   **Problem:** All code fetches above the first 8K bytes of user ROM while the device is operating in serial port programming mode will be directed to external memory. Therefore, if the user wants to call any routines in the user ROM, the entire routine must be within the first 8K bytes of memory (0A000 – 0BFFFFH in serial port programming mode). For example, if the RISM “GO” command is used with a target address of 0C000H, the device will attempt to fetch code from external memory rather than the on-board ROM.

   **Implication:** This errata only affects code fetches from the user ROM. Data fetches to the entire ROM work correctly. It is not possible to execute code from above the first 8K byte of user ROM while the device is operating in Serial Port Programming mode.

   **Workaround:** None.

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

2. **A/D Conversion Error on First Conversion**
   
   **Problem:** The first A/D conversion performed after a reset of the device may produce a result that is less accurate than the accuracy specification in the datasheet. The amount of error is dependent on several environmental conditions including process variation, light exposure, temperature, and $V_{CC}/V_{REF}$ differential.

   **Implication:** If the application is sensitive to A/D accuracy, an error in the application may occur as a result of the first conversion error.

   **Workaround:** Do not compare for the A/D result on the first conversion. All subsequent conversions should produce accurate results.

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

None for this revision of this specification update.
## Specification Clarifications

1. **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.

2. **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.

3. **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.

4. **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.

5. **Indirect Shift Count Value**  
**Problem:** The SHRL and SHLL instructions function correctly with count values 0–31, inclusive. However, a shift count value of XX100000B 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 XX111111B causes the overflow flag and the overflow-trap flag to be set.  
**Implication:** 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).

6. **Port 4 Address Behavior**  
**Problem:** For bus timing modes 1 and 2, specified only on the 87C196KS C-step, port 4 does not retain the address during the data portion of the bus cycle. Designs using an 8-bit external memory system in bus mode 1 or mode 2 require an external latch on port 4 to retain the address during the data portion of the bus cycle. Designs using an 8-bit external memory system in mode 0 or mode 3 do not require an external latch. Designs using 16-bit external memory systems require external latches on both port 3 and port 4 in all bus timing modes.

7. **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 EP\textsubscript{Ax} 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 (EP\textsubscript{Ax} CON.0 = 1; old data is overwritten on overrun), and
- the EP\textsubscript{Ax} 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 EP\textsubscript{Ax} TIME register is read. The act of reading EP\textsubscript{Ax} TIME allows the buffered time value to be moved into EP\textsubscript{Ax} TIME. This clears the buffer and allows another event to be captured. Remember that the act of transferring the buffer contents to the EP\textsubscript{Ax} TIME register is what actually sets the EP\textsubscript{Ax} 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 EP\textsubscript{Ax} CON.0**
  
  When the overwrite bit (EP\textsubscript{Ax} CON.0) is zero and both the EP\textsubscript{Ax} TIME register than the buffer are full, the EPA does not consider a captured edge until the EP\textsubscript{Ax} TIME register is read and the data in the capture buffer is transferred to EP\textsubscript{Ax} TIME. This prevents overruns by ignoring new input capture events when both the capture buffer and EP\textsubscript{Ax} TIME contain valid capture times. The OVR\textsubscript{x} pending bit in EPA\textsubscript{PEND} is set to indicate that an overrun occurred.

- **Enable the OVR\textsubscript{x} interrupt and read the EP\textsubscript{Ax} TIME register within the ISR**
  
  If an overrun occurs, the overrun (OVR\textsubscript{x}) interrupt will be generated. The OVR\textsubscript{x} interrupt will then be acknowledged and its interrupt service routine will read the EP\textsubscript{Ax} TIME register. After the CPU reads the EP\textsubscript{Ax} TIME register, the buffered data moves from the buffer to the EP\textsubscript{Ax} TIME register. This sets the EPA interrupt pending bit.

- **Check for pending EP\textsubscript{Ax} interrupts before exiting an EP\textsubscript{Ax} 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).
8. 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 the following 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]+  ;1CH ( 1CH;load location 1CH with value 1CH
ld     1CH,#1CH  ;1CH ( 1CH;load location 1CH with value 1CH
1db    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
```
**Specification Clarifications**

**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).

**9. CLKOUT During RESET**

**Problem:** For all steppings of the 87C196KT/KS, the CLKOUT function during reset (P2.7) differs from the 87C196KR C-step. During reset on the 87C196KT/KS, CLKOUT does not toggle; it remains in the high state. During reset on the 87C196KR C-step, CLKOUT continues to toggle.
1. **High address inputs to the low EPROM should be A14:8**

**Issue:** In the 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User's Manual, the block diagram for a 16-bit system with EPROM incorrectly listed the high address inputs to the low EPROM as A15:8. The correct inputs are A14:8, as reflected in the following figure:

**Figure 1. 16-bit System with EPROM**

**Affected Docs:** 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User's Manual, order number 273178.
2. Addresses 201DH and 201FH should contain FFH

Issue: In the 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User’s Manual, Table 4-2, addresses 201DH and 201FH should contain FFH, as shown in the following table:

Table 1. Special-purpose Memory Addresses

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

Affected Docs: 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User’s Manual, order number 273178.

3. P2_DIR description change

Issue: In the 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User’s Manual, the P2_DIR register is incorrectly described. The correct description is shown in the following table:

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2_DIR</td>
<td>1FCBH</td>
<td>Port Direction Register 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.</td>
</tr>
</tbody>
</table>

Affected Docs: 87C196Kx, 8XC196Jx, 87C196CA Microcontroller Family User’s Manual, order number 273178.