



# IXP1200 Network Processor

## B0 and C0 to D0 Stepping Change Document

---

*January 2002*



## Revision History

---

Date	Revision	Description
January 2002	001	First release

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 may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The IXP1200 Network Processor may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Copyright © Intel Corporation, 2002

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

# Contents

---

1.0	Introduction.....	5
2.0	New Features.....	5
3.0	Errata Change Summary.....	5
4.0	Package Marking.....	8
5.0	D0 Specification Changes.....	10
5.1	SRAM Bus Signal Timing Parameters.....	10
5.2	FCLK AC Parameter Measurements.....	10

## Figures

1	B0 Stepping Package Marking.....	9
2	C0 Stepping Package Marking.....	9
3	D0 Stepping Package Marking.....	10

## Tables

1	B0 to D0 Errata Fix Summary.....	6
2	C0 to D0 Errata Fix Summary.....	8



## 1.0 Introduction

The purpose of this document is to provide customers of the IXP1200 Network Processor with information about the D0 stepping change. This document describes the new features available with the D0 stepping and details errata that have been fixed for this new release of the device.

## 2.0 New Features

The D0 stepping of the IXP1200 contains these new features when compared to the B0 stepping:

- An increase in Instruction Store from 1K to 2K in each of the Six Microengines. Current software will require some changes to take advantage of this new feature. See the *IXP1200 Programmer's Reference Manual* and the *IXP1200 Hardware Reference Manual* for complete details.
- The IXP1200 is now available in 166 MHz, 200 MHz, and 232 MHz versions. No software changes are required for this enhancement. However, a new PLL\_CFG register setting is required. See the *IXP1200 Programmer's Reference Manual* and the *IXP1200 Hardware Reference Manual* for complete details.
- The marking has changed (see [Section 4.0](#)).
- Use the Intel<sup>®</sup> Exchange Architecture (IXA) Software Development Kit (SDK) 2.01 or later.

The D0 stepping of the IXP1200 contains these new features when compared to the C0 stepping:

- No new features are effected on the D0 stepping.
- The IXP1200 D0 is a direct pin-for-pin replacement for the C0 stepping. The marking has changed (see [Section 4.0](#)).
- Use the Intel<sup>®</sup> Exchange Architecture (IXA) Software Development Kit (SDK) 2.01 or later.

## 3.0 Errata Change Summary

[Table 1](#) lists errata that have been fixed for the D0 stepping when compared to the B0 stepping. Complete details on these errata as well as all errata associated with the IXP1200 Network Processor is available in the *IXP1200 Specification Update* (posted on the Field Division Database link (FDBL)) available from your Intel representative.

[Table 2](#) lists errata that have been fixed for the D0 stepping when compared to the C0 stepping. Complete details on these errata as well as all errata associated with the IXP1200 Network Processor is available in the *IXP1200 Specification Update* (posted on the Field Division Database link (FDBL)) available from your Intel representative.

Table 1. B0 to D0 Errata Fix Summary (Sheet 1 of 2)

Problem	Implication
<p><b>Packet Status Length</b></p> <p>When the IX Bus is configured in 32-bit unidirectional mode with a packet status length of 64 bits, the status may be received in error.</p>	<p>Using 64-bit status in 32-bit unidirectional IX Bus mode can have unpredictable results.</p>
<p><b>PCI Extended Capabilities Support Detection</b></p> <p>Bit 20 of the PCI_CMD_STAT, which is aliased as PCI Status Register bit 4, is improperly hardwired to 0, causing improper detection of IXP1200 PCI Extended Capabilities. This is a PCI 2.2 compatibility issue for customers who may be using the IXP1200 in a PC architecture using PCI BIOS and/or PCI plug-and-play drivers.</p>	<p>PCI Extended Capabilities support in the IXP1200 may not be properly detected by BIOS and/or PCI plug-and-play drivers. The IXP1200 may fail Microsoft's Hardware Compatibility Tests.</p>
<p><b>Flow-Thru SRAM Read-Modify-Write</b></p> <p>When the IXP1200 is configured with Flow-Thru SRAMs, a StrongARM Core byte write operation to SRAM will write incorrect data to the location that the operation was performed on.</p>	<p>Incorrect data will be in the SRAM location that the StrongARM Core operation byte write was performed on.</p>
<p><b>Non-aligned PCI DMA Transfers</b></p> <p>If using non-longword aligned PCI or SDRAM addresses (address bit 1 or 0 =1) during DMA Inbound or DMA Outbound transfers, either DMA channel 1 or DMA channel 2 may experience unpredictable behavior.</p>	<p>Incorrect DMA operation may cause unpredictable results, including data errors.</p>
<p><b>Bit Test &amp; Set and Bit Test &amp; Clear SRAM Operations from the StrongARM Core</b></p> <p>StrongARM Core instructions that use the Bit Test &amp; Set or the Bit Test &amp; Clear SRAM address ranges (0x1980 0000 - 0x19FF FFFF and 0x1900 0000 - 0x197F FFFF, respectively) do not return correct data value in the SRAM_TEST_MOD register.</p>	<p>StrongARM applications that share data structures with the Microengines cannot rely on the SRAM Bit Test &amp; Set and Bit Test &amp; Clear operations to provide atomic access to those data structures. When a StrongARM application writes to an SRAM Bit Test &amp; Set or Bit Test &amp; Clear address, the SRAM controller places the original test data value in the SRAM_TEST_MOD register and modifies the contents of memory by setting or clearing the specified bits. The value in the SRAM_TEST_MOD register is read by the application to determine the pre-modified state of the specified bit(s). The bit setting and clearing works correctly, however, the SRAM controller incorrectly returns invalid data in the SRAM_TEST_MOD register. This will create a state whereby the application cannot determine if any of the specified bits were set or cleared.</p>
<p><b>IXP1200 PCI_INT_LAT Register Bits [11:8]</b></p> <p>Bits [11:8] of the IXP1200 PCI_INT_LAT register are writable from PCI. These bits correspond to PCI Interrupt Configuration Register bits [11:8]. These bits should be read-only from PCI.</p>	<p>The IXP1200 interrupt to PCI (INTA#) could be inadvertently disabled by BIOS or PCI plug-and-play drivers if these bits are written by PCI enumeration code to a value of 0000.</p>
<p><b>PCI CSR (Control and Status Register) Access</b></p> <p>StrongARM Core writes to any of the registers in the PCI Unit can get blocked if the write coincides in time with a PCI Master-to-CSR access. The StrongARM Core does the write operation, but the write data may not be written to the register.</p>	<p>The implications of this problem vary widely because of the number of PCI registers affected (a total of 82 registers). Some examples are:</p> <ol style="list-style-type: none"> <li>1. Incorrect operation of DMA channels</li> <li>2. Incorrect operation of PCI I<sub>2</sub>O operations</li> <li>3. Incorrect operation of PCI interrupts</li> <li>4. Incorrect operation of Timers 1 through 4</li> </ol>

**Table 1. B0 to D0 Errata Fix Summary (Sheet 2 of 2)**

<p><b>Non-aligned PCI DMA Transfers</b> If using non-longword aligned PCI or SDRAM addresses (address bit 1 or 0 =1) during DMA Inbound or DMA Outbound transfers, either DMA channel 1 or DMA channel 2 may experience unpredictable behavior.</p>	<p>Incorrect DMA operation may cause unpredictable results, including data errors.</p>
<p><b>Read-Lock CAM Operations from the StrongARM* Core to SRAM</b> StrongARM* core instructions that use the SRAM CAM address range (0x1200 0000 - 0x127F FFFF) to perform a read-locked access can not rely on the lock attempt succeeding.</p>	<p>StrongARM* applications that share data structures with the Microengines cannot rely on the SRAM CAM to provide atomic access to those data structures. When a StrongARM* application issues a read_lock operation, the operation may be placed in the read_lock fail queue by the SRAM controller. The application determines whether or not the operation was placed in the read_lock fail queue by checking the value of the RLS bit of the SRAM_CSR register.</p> <p>To determine when a failed read_lock request is eventually moved from the read_lock rail queue to the CAM, the application polls the SRAM_CSR register until the RLRS bit is set to 1. The SRAM controller is incorrectly failing to set the RLRS bit when read_lock operation is moved from the read_lock fail queue to the SRAM CAM. Therefore, a StrongARM* application is not notified when a read_lock request is ultimately granted. This will cause locks to be placed in the CAM without application awareness.</p>
<p><b>Inoperative PCI_OUT_INT_MASK Register</b> The IXP1200 Network Processor PCI_OUT_INT_MASK register is not functional. This register is intended to prevent the IXP1200 from asserting <b>pci_irq_1</b>. A write to the PCI_OUT_INT_MASK register does not change the value of the register. A read of the PCI_OUT_INT_MASK register returns the value of the PCI_CAP_PTR register.</p>	<p>The Outbound Post List Interrupt and Doorbell Interrupt cannot be disabled.</p>
<p><b>SDRAM Arbiter</b> Commands are dropped in the SDRAM controller when using <code>sdram[ ], optimize_mem</code>. Chip would “hang” due to a lockout condition in the SDRAM arbiter A sequence of SDRAM commands to different queues would eventually cause the arbiter to NOT grant any command that isn’t intended for the high priority queue.</p>	<p>Using the <code>optimize_mem</code> token on SDRAM references may freeze microengines.</p>
<p><b>SA1200 Software Reset</b> PCI-SA1200 Reset register does not function as specified. Problems have been observed in attempting to reset the IXP1200 via the PCI_RST# input or the IXP1200_RESET register (SW reset).</p>	<p>Hangs the system, typically after running for a period of time.</p>
<p><b>PCI Parity Error Signal</b> Parity Error Signal not asserting. The Parity bit in the PCI interface is not set correctly. The Parity error indication bit in the PCI_STATUS register is correct.</p>	<p>Bad parity.</p>

Table 2. C0 to D0 Errata Fix Summary

Problem	Implication
<p><b>SRAM Arbiter</b></p> <p>Commands are dropped in the SDRAM controller when using <code>sram[ ]</code>, <code>optimize_mem</code>. Chip would “hang” due to a lockout condition in the SDRAM arbiter. A sequence of SDRAM commands to different queues would eventually cause the arbiter to NOT grant any command that is not intended for the high priority queue.</p>	<p>Using the <code>optimize_mem</code> token on SDRAM references may freeze microengines.</p>
<p><b>SA1200 Software Reset</b></p> <p>PCI-SA1200 Reset register does not function as specified</p>	<p>Hangs the system, typically after running for a period of time.</p>
<p><b>Inoperative PCI_OUT_INT_MASK Register</b></p> <p>The IXP1200 Network Processor <code>PCI_OUT_INT_MASK</code> register is not functional. This register is intended to prevent the IXP1200 from asserting <code>pci_irq_1</code>. A write to the <code>PCI_OUT_INT_MASK</code> register does not change the value of the register. A read of the <code>PCI_OUT_INT_MASK</code> register returns the value of the <code>PCI_CAP_PTR</code> register.</p>	<p>The Outbound Post List Interrupt and Doorbell Interrupt cannot be disabled.</p>
<p><b>PCI Parity Error Signal</b></p> <p>Parity Error Signal not asserting.</p> <p>The Parity bit in the PCI interface is not set correctly. The Parity error indication bit in the <code>PCI_STATUS</code> register is correct.</p>	<p>Bad parity.</p>

## 4.0 Package Marking

The mark orientation has changed. [Figure 1](#) shows the B0 mark, [Figure 2](#) shows the C0 mark, and [Figure 3](#) shows the D0 mark.

Figure 1. B0 Stepping Package Marking

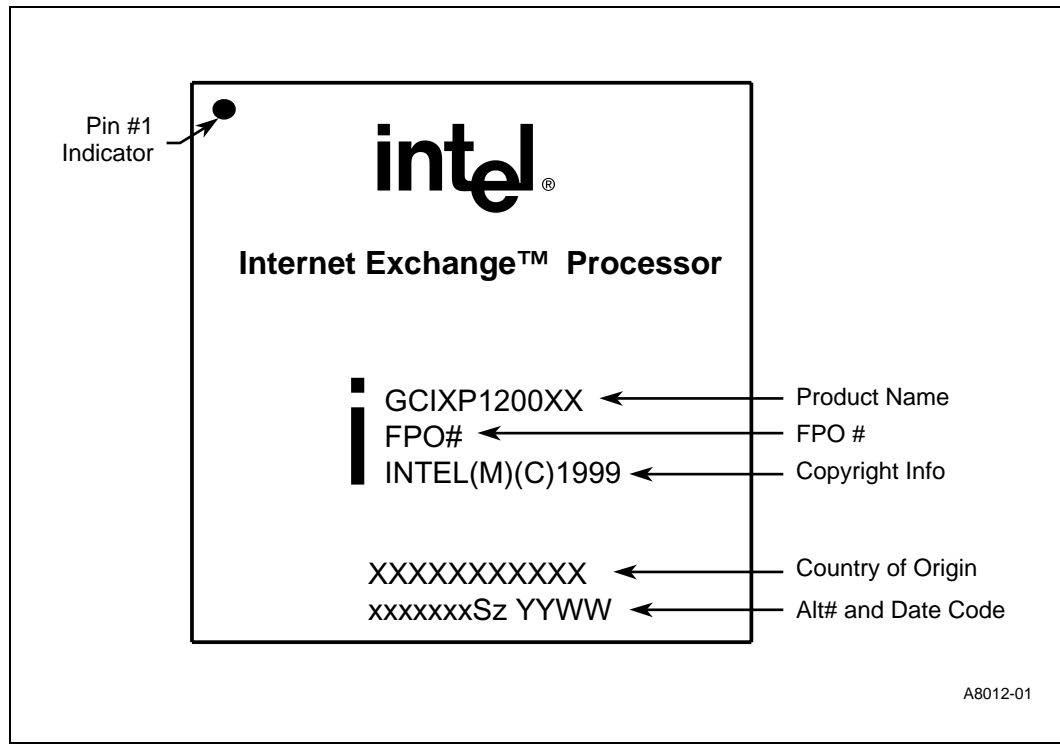
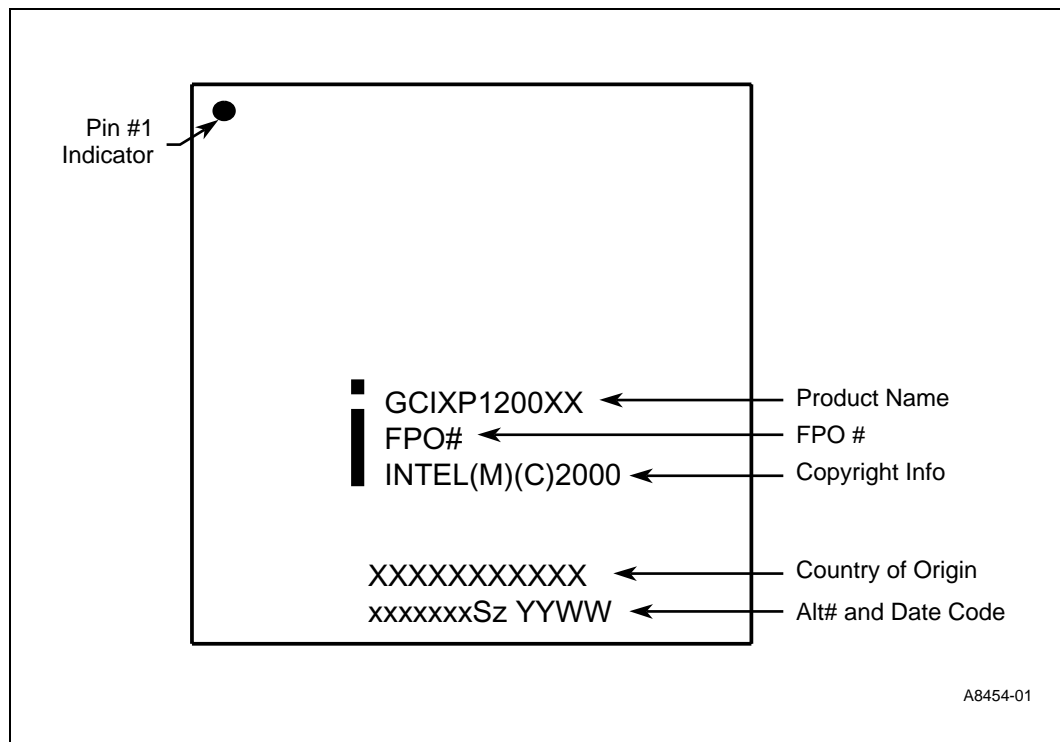
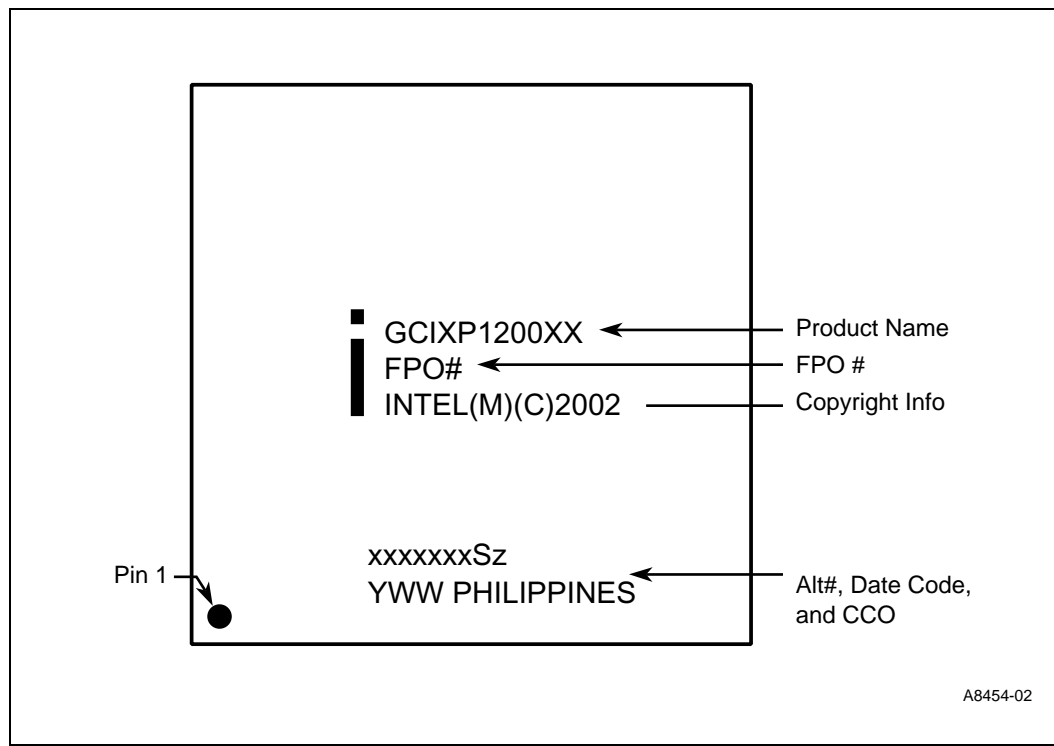


Figure 2. C0 Stepping Package Marking



**Figure 3. D0 Stepping Package Marking**



## 5.0 D0 Specification Changes

### 5.1 SRAM Bus Signal Timing Parameters

The maximum clock to data output valid delay ( $T_{val}$ ) value for 232 MHz operation was originally specified as 3.4 ns. The new  $T_{val}$  value is 3.3 ns.

### 5.2 FCLK AC Parameter Measurements

The parameter values for  $T_{high}$ ,  $T_{low}$ , and the  $T_r$  and  $T_f$  units have changed as follows:

- The minimum Clock high time ( $T_{high}$ ) was originally specified as 4.5 ns. The new  $T_{high}$  value is 3.8 ns.
- The minimum Clock low time ( $T_{low}$ ) was originally specified as 4.5 ns. The new  $T_{high}$  value is 3.8 ns.
- Both  $T_{high}$  and  $T_{low}$  have been further clarified by the statement “ $T_{high}$  and  $T_{low}$  are based on a 50% duty cycle and can vary worst case 45-55%”.
- The units used for Clock rise ( $T_r$ ) and Clock fall ( $T_f$ ) time was originally specified as V/ns. The new unit is ns.