

Intel[®] 80303 and 80302 I/O Processors

Specification Update

March 2007

Notice: The Intel[®] 80303 and 80302 I/O Processors 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.

Order Number: 273355-011US



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, life sustaining, critical control or safety systems, or in nuclear facility 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 information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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 order 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 Web Site](#).

[When the doc contains software source code, include a copy of the software license or a hyperlink to its permanent location.]

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

Code Names are only for use by Intel to identify products, platforms, programs, services, etc. ("products") in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as "commercial" names for products. Also, they are not intended to function as trademarks.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel Leap ahead, Intel Leap ahead logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skood, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2007, Intel Corporation. All rights reserved.



Contents

Revision History	4
Introduction	5
Summary Tables of Current Product Issue Activity	7
General Product Information	10
Errata	12
Specification Changes	14
Specification Clarifications	15
Document-Only Changes	17



Revision History

Date	Revision	Description
March 2007	011	Added Document Change 34.
05/01/03	010	Added Errata 2. Revised Specification Clarifications 4, 7 and 8.
08/27/02	009	Reworded Specification Clarification 4. Added Specification Clarifications 7 and 8.
11/15/01	008	Added Specification Clarifications 5 and 6. Added Document Changes 32 and 33.
08/22/01	007	Added Specification Clarification 4. Added Document Changes 30 and 31.
04/24/01	006	Added Document Changes 25 through 29. Revised Device ID Registers "A-2" Revision ID Registers data. Added Note to Device ID Registers.
04/02/01	005	Added Specification Clarification 3.
03/22/01	004	Added Errata 1. Added Specification Change 1. Added Specification Clarifications 1 and 2. Added Document Changes 13 through 24. Updated Die Details Table and Device ID Registers for A-2 step.
02/23/01	003	Die Details and Device ID Registers tables, corrected stepping A-0 to A-1.
08/2000	002	Updated Die Details Table. Revised Device ID Registers Table. Added Document Changes 1 through 12.
06/2000	001	This is the new Specification Update document. It contains all identified errata published prior to this date.



Introduction

Purpose/Scope/Audience

This document is an update to the specifications listed in the [Affected Documents/Related Documents](#) table that follows. This document is a compilation of [Errata](#), [Specification Changes](#), [Specification Clarifications](#), and [Document-Only Changes](#). It is intended for hardware and software system designers and manufacturers as well as developers of applications, operating systems, or tools.

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

Table 1. Affected Documents/Related Documents

Title	Number
Intel® 80303 I/O Processor Developer's Manual	273353
Intel® 80303 I/O Processor Data Sheet	273358
Intel® 80303 I/O Processor Design Guide	273308



Conventions and Terminology

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 **Document-Only Changes** are removed from the Specification Update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

Table 2. Conventions and Terminology

Term	Definition
Document-Only Changes	Document-Only Changes are changes to an Intel Parent Specification that result in changes only to an Intel customer document but no changes to a specification or to a parameter for an Intel product. An example of a document-only change is the correction of a typographical error.
Errata (plural) Erratum (singular)	Errata are design defects or errors. These may cause the Intel® 80303 and 80302'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.
Parent Specification	A parent specification is a top-level specification from which other documents can be derived, depending on the product or platform. Typically, a parent specification includes a product's pinout, architectural overview, device operation, hardware interface, or electrical specifications. Examples of parent specifications include the following: Datasheet, Developer's Manual, Technical Product Specification (also known as "TPS"). The derived documents may be used for purposes other than that for which the parent specification is used.
Specification Changes	Specification Changes are the result of adding, removing, or changing a feature, after which an Intel product subsequently operates differently than specified in an Intel Parent Specification , but typically the customer does not have to do anything to achieve proper device functionality as a result of Intel adding, removing, or changing a feature.
Specification Clarifications	Specification Clarifications are changes to a document that arise when an Intel Parent Specification must be reworded so that the specification is either more clear or not in conflict with another specification.



Summary Tables of Current Product Issue Activity

Table 4 through Table 7 indicate the [Errata](#), [Specification Changes](#), [Specification Clarifications](#), or [Document-Only Changes](#) that apply to the Intel® 80303 and 80302 product. Intel may fix some of the [Errata](#) in a future stepping of the component as noted in [Table 3](#) or account for the other outstanding issues through [Specification Changes](#), [Specification Clarifications](#), or [Document-Only Changes](#). Table 4 through Table 7 use the codes listed in [Table 3](#).

Table 3. Codes Used in Summary Tables

Code	Column	Definition
X	Stepping	Indicates either that, for the stepping/revision listed, - an erratum eXists and is not yet fixed - a specification change or specification clarification applies
No mark or blank	Stepping	Indicates either that, for the stepping/revision listed, - an erratum is fixed - a specification change or specification clarification does not apply
Plan Fix	Status	This erratum may be fixed in a future stepping/revision.
Fixed	Status	This erratum has been previously fixed.
No Fix	Status	There are no plans to fix this erratum.
A change bar to the left of a table row indicates an item that is either new or modified from the previous version of the Specification Update document.		



Table 4. Errata

No.	Stepping/ Revision			Status	Errata
	A-0	A-1	A-2		
1	X	X	X	NoFix	Single-bit and Multi-bit Error Reporting Cannot Be Individually Enabled by ECC Control Register
2	X	X	X	NoFix	Instruction Sequence Can Scoreboard a Register Indefinitely

Table 5. Specification Changes

No.	Stepping/ Revision			Specification Changes
	A-2	#	#	
1	X			Summary of the Intel® 80302 I/O Processor

Table 6. Specification Clarifications

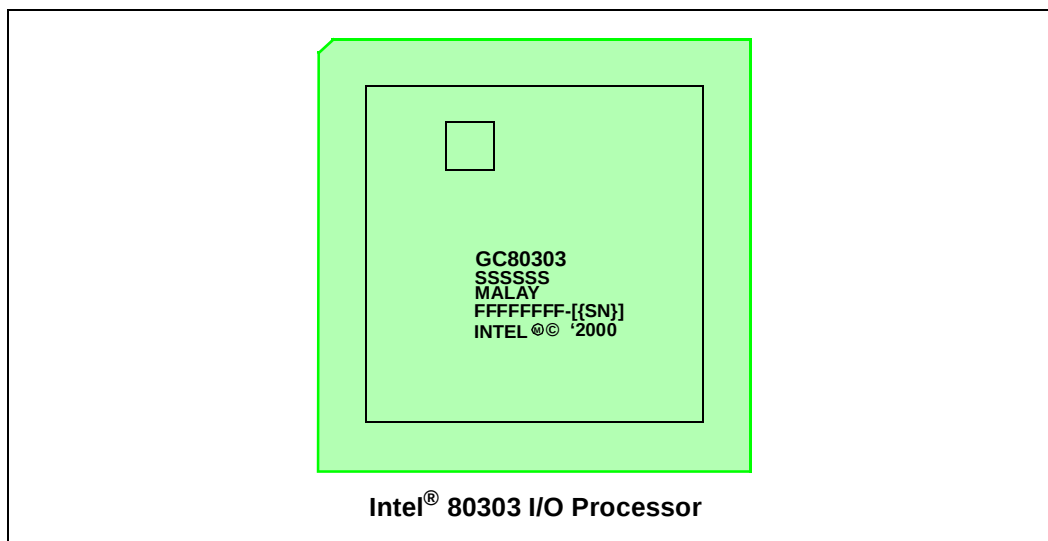
No.	Stepping/Revision			Specification Clarifications
	A-0	A-1	A-2	
1	X	X	X	ECC is Always Enabled
2	X	X	X	32-bit SDRAM is Not Supported
3	X	X	X	Non-Battery Backup Systems
4	X	X	X	POCCDR and SOCCDR Functionality
5	X	X	X	'Bus Hold' Devices on the RAD Bus
6	X	X	X	SREQ64# Functionality
7	X	X	X	PCI Local Bus Specification, Revision 2.3 Compliance
8	X	X	X	DMA and AAU End of Chain Functionality

**Table 7. Document-Only Changes**

No.	Document Title	Rev.	Document-Only Changes
1	Developer's Manual	001	Title Page revision number
2	Developer's Manual	001	Figure 9-3 on page 9-9 did not print correctly
3	Developer's Manual	001	Figure 13-22 on page 13-40 did not print correctly
4	Developer's Manual	001	Figure 13-18, page 13-35
5	Developer's Manual	001	Figure 15-2 on page 15-3 did not print correctly
6	Developer's Manual	001	Incorrect Vendor ID in ATU register
7	Developer's Manual	001	Section 23.2 on page 23-2 has incorrect text
8	Developer's Manual	001	Table 24-4 on page 24-8 is incorrect
9	Developer's Manual	001	Figure 25-1 on pg 25-1 has incorrect data
10	Developer's Manual	001	Section 25.1.3 on page 25-2
11	Developer's Manual	001	Figure 25-2 on pg 25-2 did not print correctly
12	Developer's Manual	001	Table 25-2 on page 25-3 did not print completely
13	Developer's Manual	001	Section 1.2.2 on page 1-2 has incorrect data
14	Developer's Manual	001	Figure 12-2 on page 12-10 has incorrect data
15	Developer's Manual	001	Section 19.1 on page 19-1 has incorrect data
16	Developer's Manual	001	Table 14-46 on page 14-109 has missing data
17	Developer's Manual	001	Section 13.2.4.3 on page 13-30 has incorrect data
18	Developer's Manual	001	Figure 15-3 on page 15-7 has missing text
19	Developer's Manual	001	Section 15.7.39 on page 15-100 has incorrect data
20	Developer's Manual	001	Table 8-17 on page 8-38 has incorrect data
21	Developer's Manual	001	Section 11.2.8 on page 11-5 has incorrect data
22	Developer's Manual	001	Section 13.2.3.1 on page 13-13 has incorrect data
23	Developer's Manual	001	Table 13-4 on page 13-9 has incorrect data
24	Developer's Manual	001	Table 8-15 on page 8-36 needs clarification
25	Developer's Manual	001	Table 13-13 on page 13-30 has incorrect data
26	Developer's Manual	001	Section 13.2.4.3, First Paragraph after Table 13-13 has Incorrect Data
27	Developer's Manual	001	Section 13.2.4.3, First Paragraph after "Current" Figure 13-16. H-Matrix has Incorrect Data
28	Developer's Manual	001	Section 11.3.1.5 FAIL# Code
29	Developer's Manual	001	Section 13.5 Reset Conditions has Incorrect Data
30	Developer's Manual	001	Section 13.2.4.2, First Sentence has Incorrect Data
31	Developer's Manual	001	Section 13.6.2, Second Sentence has Incorrect Data
32	Datasheet	007	Section 4.5.2 on page 50 is only correct for A-0 and A-1 steppings
33	Developer's Manual	001	Section 17.5.1 on page 17-12 is only correct for A-0 and A-1 steppings
34	Datasheet	011	Package change for the Intel® 80303/80302/80312 I/O processors products and lists the new order codes

General Product Information

Figure 1. Product Markings



Die Details

Part Number	Stepping	QDF/ Spec Number	Voltage (V)	Intel® i960® Core Processor Speed (MHz)	Notes
GC80303	A-0	Q176	3.3	100	Samples - limited testing
GC80303	A-0	Q196	3.3	100	Samples - limited testing
GC80303	A-1	Q189	3.3	100	Samples - limited testing
GC80303	A-1	SL4Q4	3.3	100	Production
GC80303	A-2	SL57T	3.3	100	Production - Yield improvement only, no functionality changes.
GC80302	A-2	Q229	3.3	100	Samples - limited testing, 66 MHz internal bus and SDRAM memory interface.
GC80302	A-2	SL5HS	3.3	100	Production - 66 MHz internal bus and SDRAM memory interface.



Device ID Registers

Device and Stepping	Processor Device ID Register (PDIDR - 0x1710)	PCI-to-PCI Bridge Unit Revision ID (RIDR - 0x1008)	Address Translation Unit Revision ID Register (ATURID - 0x1208)	Intel® i960® Core Processor Device ID (DEVICEID - 0xFF00 8710)
80303 A-0	08879013	0x00	0x00	00823013
80303 A-1	18879013	0x01	0x01	00823013
80303 A-2	18879013	0x01	0x01	00823013
80302 A-2	18878013	0x01	0x01	00823013

Note: There are no functionality differences between the A-1 and A-2 steppings of the 80303. Therefore, the Device IDs are the same.

Errata

1. Single-bit and Multi-bit Error Reporting Cannot Be Individually Enabled by ECC Control Register

Problem: The ECC Control Register ECCR is described as having the ability to select multi-bit error and/or single-bit error reporting (see Table 13-24 on page 13-31 of the *Intel® 80303 I/O Processor Developer's Manual*). However, the algorithm does not allow individual enabling; that is, the reporting is either on or off for both multi-bit and single bit error reporting.

Implication: The error reporting selection (enabled or disabled) will apply to both multi-bit and single-bit errors.

Workaround: There is no current workaround. If either the ECCR.0 bit or the ECCR.1 bit is selected for reporting, then both multi-bit and single-bit error reporting are enabled. If neither bit is selected for reporting, then both multi-bit and single-bit error reporting are disabled.

Status: NoFix. See the [Table , "Summary Tables of Current Product Issue Activity" on page 7](#).

2. Instruction Sequence Can Scoreboard a Register Indefinitely

Problem: Register scoreboarding maintains register coherency by preventing parallel execution units from accessing registers for which there is an outstanding operation (see section 3.2.3 in the *Intel® 80303 I/O Processor Developer's Manual*).

An instruction sequence that coincides with some specific instruction cache conditions can scoreboard a local or global register indefinitely. When this happens, processing can stall at the next access to that register, awaiting a scoreboard release that does not come. In that case, external bus accesses cease.

A hardware reset is the only way to release the scoreboard.

The following three conditions are required to scoreboard a register:

1. Execution of the following three-instruction sequence:

- a. `emul`
- b. `ld, ldos, ldis, ldob, or ldib`
- c. `mulo` or `muli`

Only two-word, MEMB format *load* instructions that execute in two clock cycles cause the failure. [Table 8](#) lists all the versions of these instructions that can produce this failure. Any version can be used for each instruction and still produce the failure as long as the sequential order is maintained.

2. The *emul* must appear at address 0xxxxxxx8.
3. Instruction caching must be enabled. The *emul* instruction must be fetched from external memory along with the first word of the *load* instruction. Also, the second word of the *load* and the *multiply* instruction must already reside in cache. To accomplish this, the code must have run once in order to load the instructions into cache followed by code which causes the invalidation of the cache line containing the *emul* instruction. At this point, re-execution of the code sets up the failure condition.



Once the failure condition occurs, the processor will continue code execution until an instruction using the scoreboarded register is encountered, then indefinite processor stall will occur.

Nominally, the *emul* multiplies two 32-bit operands to produce a long ordinal (64-bit) result stored in two adjacent registers. When the errata occurs, the low-order register receives the correct value, but the high-order register becomes scoreboarded indefinitely. The scoreboarded register is always odd-numbered (i.e., g1, g3, g5, ..., r7, r9, r11, ...) since the *emul* instruction always directs the high-order result to the odd-numbered register of the destination pair.

In some cases, the result of the *mulo* or *muli* instruction is corrupted, too, but such has never been observed apart from the scoreboard failure. Once the scoreboard failure has occurred, subsequent *mulo* or *muli* instructions that are separated by *load* instructions can also produce faulty results in some cases. The details of this secondary behavior has not been studied as thoroughly as the primary scoreboarding issue.

When the scoreboard stalls the processor, higher-level processes, such as higher-priority interrupts and faults, can run as normal unless they access the scoreboarded register and also stall.

Not all 80303 processors have been observed to exhibit this errata.

Table 8. Instruction Versions that Can Produce a Scoreboard Failure in this Sequence

Instruction	Data Type	Addressing Mode	Format
1. Extended Multiply	All	n/a	<i>emul</i> reg/lit, reg/lit, reg
2. Load	- Word - <i>ld</i> - Ordinal short - <i>ldos</i> - Integer short - <i>ldis</i> - Ordinal byte - <i>ldob</i> - Integer byte - <i>ldib</i>	- Absolute Displacement - Register Indirect with Displacement - Index with Displacement	<i>ld</i> exp, reg <i>ld</i> exp(reg), reg <i>ld</i> exp[reg*scale], reg <i>ldos</i> exp, reg <i>ldos</i> exp(reg), reg <i>ldos</i> exp[reg*scale], reg <i>ldis</i> exp, reg <i>ldis</i> exp(reg), reg <i>ldis</i> exp[reg*scale], reg <i>ldob</i> exp, reg <i>ldob</i> exp(reg), reg <i>ldob</i> exp[reg*scale], reg <i>ldib</i> exp, reg <i>ldib</i> exp(reg), reg <i>ldib</i> exp[reg*scale], reg
3. Multiply	- Ordinal - Integer	n/a	<i>mulo</i> reg/lit, reg/lit, reg <i>muli</i> reg/lit, reg/lit, reg

Implication: Systems containing this instruction sequence may exhibit sporadic and unrepeatable stall failures depending on where these instructions appear in the executable memory image and the runtime dynamics as they affect the Icache.

Workaround: Avoid this sequence of instructions in systems that employ the instruction cache.

Status: **NoFix.** Refer to Table , “Summary Tables of Current Product Issue Activity” on page 7 to determine the affected stepping(s).



Specification Changes

1. Summary of the Intel® 80302 I/O Processor

Problem: The Intel® 80302 I/O processor is based on the A-2 stepping of the Intel® 80303 I/O processor. The 80302 I/O processor is identical to the 80303 I/O processor, except the SDRAM and internal bus run at 66 MHz. For applications that use the I²C unit, the I²C clock is generated from the internal bus clock, so the ICCR (I²C Clock Count Register) needs to be properly adjusted.

The Device ID Register (DIDR; 1002H), in the Bridge configuration header, is 0308H for the 80302 I/O processor.

The ATU Device ID Register (ATUDID; 1202H), in the ATU configuration header, is 5308H for the 80302 I/O processor.

The 80303 I/O processor manual, datasheet and design guide should be used when designing with the 80302 I/O processor.

Status: The 80302 I/O processor will be introduced with the A-2 stepping.



Specification Clarifications

1. ECC is Always Enabled

Problem: ECC is always enabled, therefore do not design an Intel® 80303 I/O processor based product without ECC implemented, this causes severe system errors. On the Intel® 80960RM/RN I/O processors, ECCR.3 can be cleared to disable ECC, but with the 80303 I/O processor, ECCR.3 is reserved.

2. 32-bit SDRAM is Not Supported

Problem: The memory controller on the 80303 I/O processor supports between 32 and 512 Mbytes of 64-bit SDRAM, but 32-bit SDRAM is not supported. On the 80960RM/RN I/O processors, 32-bit memory was selected by the 32BITMEM_EN# pin (multiplexed on RAD[2]), and by reading a '0' from SDCR.2, this would indicate a 32-bit data bus width. But, for the 80303 I/O processor the 32BITMEM_EN# pin does not exist and SDCR.2 is reserved.

3. Non-Battery Backup Systems

Problem: Applications that do not support battery back-up should follow these recommendations:

1. Pull the PWRDELAY pin low through a 1.5K pull-down. Pulling it low has the effect of keeping the power fail state machine in reset, therefore not allowing the power fail sequence to ever occur.
2. Pull the CKE pins high on the SDRAMs, and leave the SCKE signals on the 80303 as 'no connects'. This keeps the SDRAM from entering a pseudo, self-refresh mode which can cause a lock-up condition on the SDRAM device.

4. POCCDR and SOCCDR Functionality

Problem: The Primary Outbound Configuration Cycle Data Register (POCCDR) and Secondary Outbound Configuration Cycle Data Register (SOCCDR) are used to initiate configuration cycles to PCI target devices. On page 15-57, Table 15-26 in the *Intel® 80303 I/O Processor Developer's Manual*, these registers are stated as "Not Available in PCI Configuration Space".

To clarify, when these registers are either read or written via PCI during a scan of configuration space, an unwanted configuration cycle is initiated by the 80303 to the address held in the Primary Outbound Configuration Cycle Address Register (POCCAR) or Secondary Outbound Configuration Cycle Address Register (SOCCAR) based on a read or write to POCCDR or SOCCDR respectively. An invalid address causes the 80303 to signal a master abort. Only the first 64 bytes in the ATU Configuration Header is read during configuration. Anything above 64 bytes up to 256 bytes is defined as device-specific and not accessed by a master. This does not have to rule out access by any master, only a master which does not have knowledge of the device-specific registers.

5. 'Bus Hold' Devices on the RAD Bus

Problem: There are six user mode configuration pins (RST_MODE#, ONCE#, STEST, RETRY, SPMEM# and 32BITPCI_EN#) and three test mode configuration pins (on RAD8, 7 and 0) that are multiplexed on the RAD[8:0] signals. All these signals have internal pull-ups, so there is no need for external pull-ups. But, if the application requires an

active low signal, then an external pull-down needs to be added. The configuration signals are latched on the rising edge of P_RST#. Devices with a 'bus hold' feature (i.e., CPLD) connected to the RAD bus may pull the RAD[8:0] signals low at the rising edge of P_RST#, causing the 80303 to enter an undesired mode. 80303 designs that use 'bus hold' devices should either turn off the 'bus hold' feature or verify that proper signal levels are being maintained at the rising edge of P_RST#.

6. SREQ64# Functionality

Problem: There is an SREQ64# functionality difference between the A-1 and A-2 steppings of the 80303 I/O processors. (This functionality is also on the 80302 since it is based on the A-2 stepping.) During the power up sequence, the S_REQ64# signal is sampled by PCI devices on the secondary PCI bus to determine 64-bit or 32-bit PCI operation. On the A-1 stepping, S_REQ64# is deasserted one P_CLK after the deassertion of S_RST# (as stated in the Developer's Manual and Datasheet). On the A-2 stepping, SREQ64# is deasserted approximately 600ps after the deassertion of S_RST#.

The *PCI Local Bus Specification*, Revision 2.2 has a setup and hold specification for REQ64# with respect to RST#. Even though the Intel Datasheets and Developer's Manuals state that, "S_REQ64# is deasserted one P_CLK after the deassertion of S_RST#", the *PCI Local Bus Specification*, Revision 2.2 states that the RST# to REQ64# hold time is 0-50ns. Since the RST# to REQ64# hold time can be zero, compliant devices should be sampling REQ64# during the REQ64# to RST# setup time which is a minimum of 10 clock cycles. (see pages 128 and 135, table 4-6 and figure 4-11 of the *PCI Local Bus Specification*, Revision 2.2)

The implication of this change is that some 64-bit PCI devices on the secondary PCI bus only works in 32-bit PCI mode. This could be due to using a non-PCI compliant device or because of trace delays between the S_RST# and S_REQ64# signals. Verify proper functionality on 80303 A-2 designs. The processor stepping identification is listed page 10. Also see Documentation Changes #32 and 33 for corrections to the datasheet and manual.

7. PCI Local Bus Specification, Revision 2.3 Compliancy

Problem: The Intel® 80303 I/O processor (80303) was designed to be compliant with the *PCI Local Bus Specification*, Revision 2.2. (This functionality is also on the 80302 since it is based on the A-2 stepping.). Since the release of the 80303, the PCI Special Interest Group has released a new specification revision, *PCI Local Bus Specification*, Revision 2.3. There are no plans to step the 80303 to make it compliant with the *PCI Local Bus Specification*, Revision 2.3.

8. DMA and AAU End of Chain Functionality

Problem: There is a case where a race condition occurs between the End of Chain (EOC), Channel Active (CA) and resume bit, which causes a bogus EOC. The Intel® 80303 I/O processor (80303) (this functionality is also on the 80302 since it is based on the A-2 stepping.) asserts the EOC bit when the NDAR is zero, even when the chain resume bit is set. When the resume bit is set, the CA bit is cleared for one cycle and then set again, modifying the CA and EOC at the same time.

Consider the case when a chain has been added to the list after the last descriptor is read by the DMA. In this case, the resume bit gets set by software. The EOC occurs because the NDAR was zero when read and the CA bit is momentarily cleared. The DMA processes the resume and sets the CA bit again. It remains active until it again reaches an NDAR of 0.

One way to handle this condition, is for the software to track the last descriptor believed to be in memory. To compare the NDAR and DAR in the DMA descriptor MMR space, to see when they are 0, and are the last expected DAR. In this situation, the DMA is already idle and the CA bit is clear. When not, ignore the EOC interrupt. A bogus EOC is detected when NDAR is not 0 and resume is set.



Document-Only Changes

1. Title Page revision number

Problem: Manual indicates Revision 0.5.

Workaround: This type of revision numbering is not used with published documents. Refer to the Document Number 272353-001. The extension -001 is the correct revision number for this document.

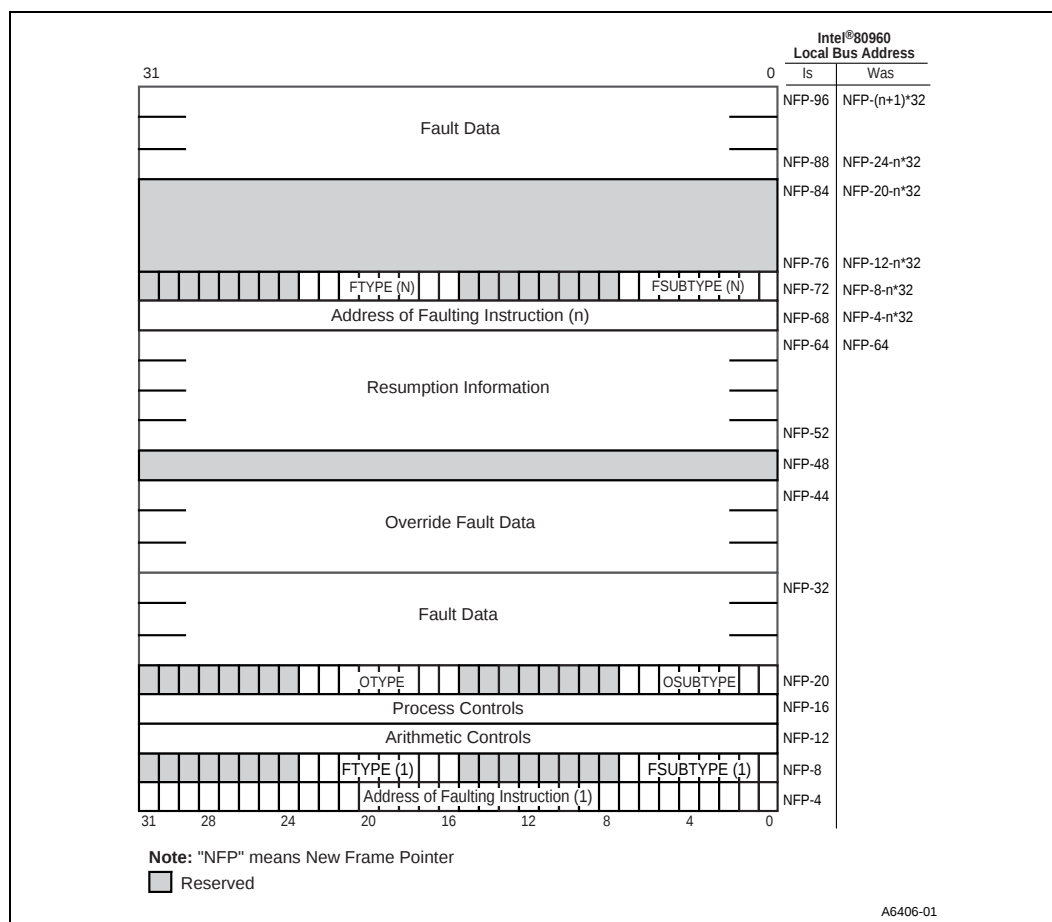
Ignore revision number 0.5.

Affected Docs: Intel® 80303 I/O Processor Developer's Manual.

2. Figure 9-3 on page 9-9 did not print correctly

Problem: Figure 9-3 on page 9-9 did not print correctly.

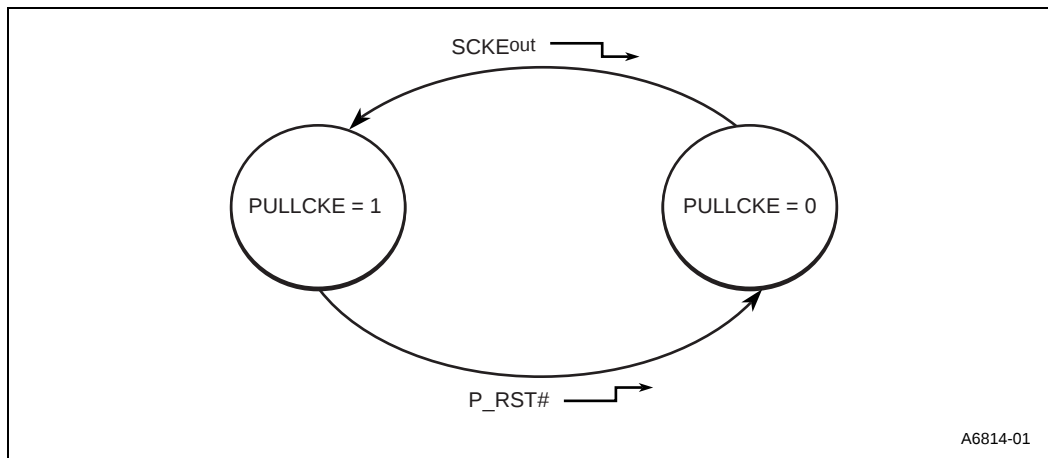
Workaround: Replace Figure 9-3 with the following:



3. Figure 13-22 on page 13-40 did not print correctly

Problem: Figure 13-22 on page 13-40 did not print correctly.

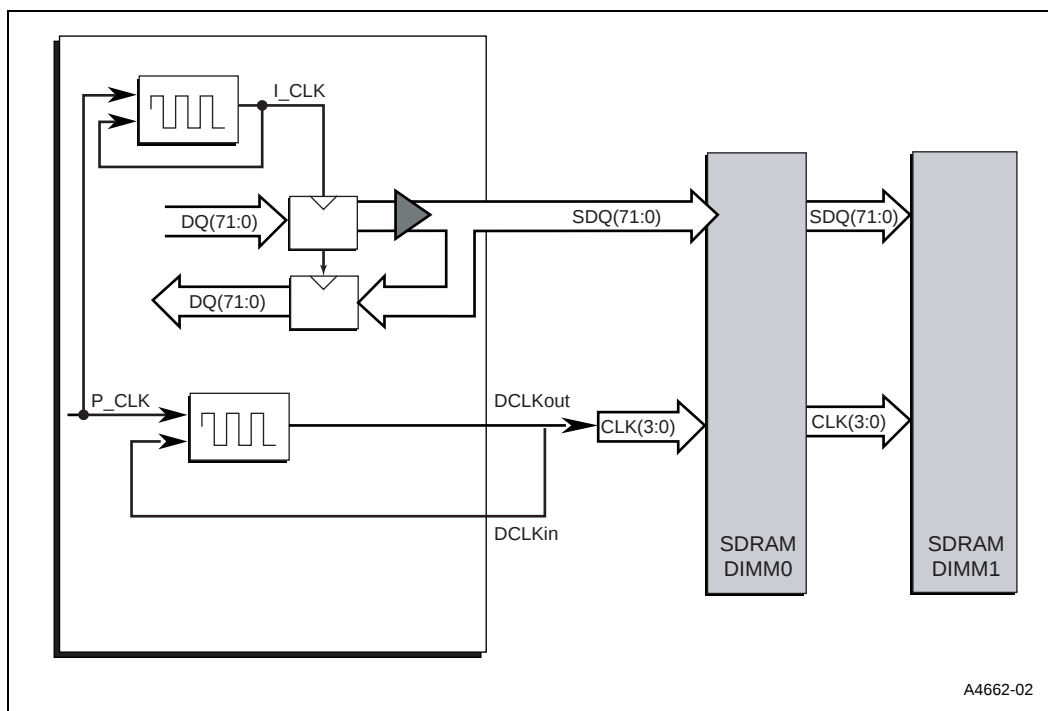
Workaround: Replace Figure 13-22 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual.

4. Figure 13-18, page 13-35

Problem: Replace Figure 13-18 with the following:



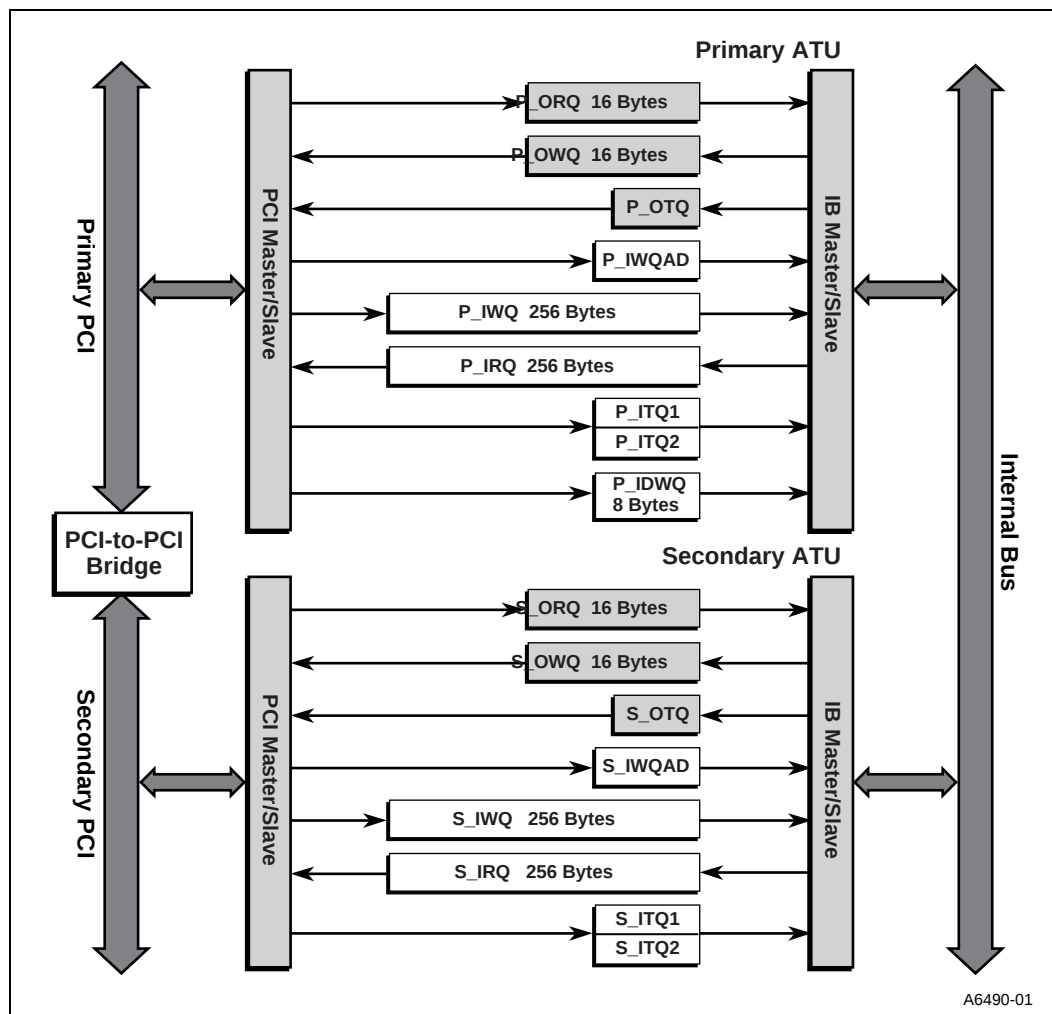
Affected Docs: Intel® 80303 I/O Processor Developer's Manual.



5. Figure 15-2 on page 15-3 did not print correctly

Problem: Figure 15-2 on page 15-3 did not print correctly.

Workaround: Replace Figure 15-2 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual.



6. Incorrect Vendor ID in ATU register

Problem: The value for the Vendor ID register (ATUVID) is incorrect.

Workaround: Replace Table 15-28 on page 15-60 with the following table:

		Intel® i960® Core Local Bus Address 1200H	PCI Configuration Address Offset 00H - 01H	Attribute Legend: RV = Reserved PR = Preserved RS = Read/Set RW = Read/Write RC = Read Clear RO = Read Only NA = Not Accessible
Bit	Default	Description		
15:00	8086H	ATU Vendor ID - This is a 16-bit value assigned to Intel. This register, combined with the DID, uniquely identify the PCI device. Access type is Read/Write to allow the 80303 I/O processor to configure the register as a different vendor ID to simulate the interface of a standard mechanism currently used by existing application software.		

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual.*

7. Section 23.2 on page 23-2 has incorrect text

Problem: The text and register descriptions in section 23.2 are incorrect.

Workaround: Replace Section 23.2 Register Definitions with the following:

All three GPIO registers are visible as 80303 I/O processor memory mapped registers and can be accessed through the internal memory bus. Each is a 8-bit register and is memory-mapped in the 80303 processor memory space. The memory-mapped addresses of the GPIO control registers are found in [Appendix C, "Peripheral Memory-Mapped Registers."](#)

There are four control and status registers for the PCI And Peripheral Interrupt Controller:

- GPIO Output Enable Register
- GPIO Input Data Register
- GPIO Output Data Register

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual.*



8. Table 24-4 on page 24-8 is incorrect

Problem: Table 24-4 on page 24-8 is incorrect.

Workaround: Replace Table 24-4 with the following table:

#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"0	(CBSC_1,	scl,	bidir,	X,	1,	1,	Z)," &
"1	(BC_1,	*,	control,	1)," &			
"2	(CBSC_1,	sda,	bidir,	X,	3,	1,	Z)," &
"3	(BC_1,	*,	control,	1)," &			
"4	(BC_1,	rale,	output3,	X,	111,	1,	Z)," &
"5	(BC_1,	roez,	output3,	X,	111,	1,	Z)," &
"6	(CBSC_1,	rad(15),	bidir,	X,	7,	0,	Z)," &
"7	(BC_1,	*,	control,	0)," &			
"8	(BC_1,	rcez(1),	output3,	X,	9,	0,	Z)," &
"9	(BC_1,	*,	control,	0)," &			
"10	(BC_1,	rcez(0),	output3,	X,	11,	0,	Z)," &
"11	(BC_1,	*,	control,	0)," &			
"12	(CBSC_1,	rad(6),	bidir,	X,	28,	1,	Z)," &
"13	(BC_1,	rwes,	output3,	X,	111,	1,	Z)," &
"14	(CBSC_1,	rad(9),	bidir,	X,	18,	0,	Z)," &
"15	(CBSC_1,	rad(10),	bidir,	X,	16,	0,	Z)," &
"16	(BC_1,	*,	control,	0)," &			
"17	(CBSC_1,	rad(11),	bidir,	X,	18,	0,	Z)," &
"18	(BC_1,	*,	control,	0)," &			
"19	(CBSC_1,	rad(13),	bidir,	X,	20,	0,	Z)," &
"20	(BC_1,	*,	control,	0)," &			
"21	(CBSC_1,	rad(12),	bidir,	X,	22,	0,	Z)," &
"22	(BC_1,	*,	control,	0)," &			
"23	(CBSC_1,	rad(8),	bidir,	X,	28,	1,	Z)," &
"24	(CBSC_1,	rad(14),	bidir,	X,	25,	0,	Z)," &
"25	(BC_1,	*,	control,	0)," &			
"26	(CBSC_1,	rad(2),	bidir,	X,	28,	1,	Z)," &
"27	(CBSC_1,	rad(1),	bidir,	X,	28,	1,	Z)," &
"28	(BC_1,	*,	control,	1)," &			
"29	(CBSC_1,	rad(16),	bidir,	X,	30,	0,	Z)," &
"30	(BC_1,	*,	control,	0)," &			
"31	(CBSC_1,	rad(0),	bidir,	X,	28,	1,	Z)," &
"32	(CBSC_1,	rad(5),	bidir,	X,	28,	1,	Z)," &
"33	(CBSC_1,	rad(4),	bidir,	X,	28,	1,	Z)," &
"34	(CBSC_1,	rad(3),	bidir,	X,	28,	1,	Z)," &
"35	(BC_4,	nc1,	input,	X)," &			
"36	(CBSC_1,	rad(7),	bidir,	X,	28,	1,	Z)," &
"37	(CBSC_1,	GPIO(0),	bidir,	X,	38,	1,	Z)," &
"38	(BC_1,	*,	control,	1)," &			
"39	(CBSC_1,	GPIO(1),	bidir,	X,	40,	1,	Z)," &
"40	(BC_1,	*,	control,	1)," &			
"41	(CBSC_1,	GPIO(2),	bidir,	X,	42,	1,	Z)," &
"42	(BC_1,	*,	control,	1)," &			
"43	(BC_1,	i_rstz,	output3,	X,	111,	1,	Z)," &
"44	(CBSC_1,	GPIO(3),	bidir,	X,	45,	1,	Z)," &
"45	(BC_1,	*,	control,	1)," &			



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"46	(CBSC_1,	GPIO(4),	bidir,	X,	47,	1,	Z)," &
"47	(BC_1,	*,	control,	1)," &			
"48	(CBSC_1,	GPIO(5),	bidir,	X,	49,	1,	Z)," &
"49	(BC_1,	*,	control,	1)," &			
"50	(CBSC_1,	GPIO(6),	bidir,	X,	51,	1,	Z)," &
"51	(BC_1,	*,	control,	1)," &			
"52	(CBSC_1,	GPIO(7),	bidir,	X,	53,	1,	Z)," &
"53	(BC_1,	*,	control,	1)," &			
"54	(BC_4,	irqz(0),	input,	X)," &			
"55	(BC_4,	irqz(1),	input,	X)," &			
"56	(BC_4,	irqz(2),	input,	X)," &			
"57	(BC_4,	irqz(3),	input,	X)," &			
"58	(BC_4,	irqz(4),	input,	X)," &			
"59	(BC_4,	irqz(5),	input,	X)," &			
"60	(BC_1,	nc2,	output3,	X,	65,	1,	Z)," &
"61	(BC_4,	lcdinitz,	input,	X)," &			
"62	(BC_4,	logic1,	input,	X)," &			
"63	(BC_1,	failz,	output3,	X,	110,	1,	Z)," &
"64	(BC_4,	nmiz,	input,	X)," &			
"65	(BC_1,	*,	control,	1)," &			
"66	(BC_4,	nc3,	input,	X)," &			
"67	(BC_4,	nc4,	input,	X)," &			
"68	(BC_4,	nc5,	input,	X)," &			
"69	(BC_4,	nc6,	input,	X)," &			
"70	(BC_4,	nc7,	input,	X)," &			
"71	(BC_4,	nc8,	input,	X)," &			
"72	(BC_4,	nc9,	input,	X)," &			
"73	(BC_4,	nc10,	input,	X)," &			
"74	(BC_4,	nc11,	input,	X)," &			
"75	(BC_4,	nc12,	input,	X)," &			
"76	(BC_4,	nc20,	input,	X)," &			
"77	(BC_4,	nc13,	input,	X)," &			
"78	(BC_4,	nc15,	input,	X)," &			
"79	(BC_4,	nc14,	input,	X)," &			
"80	(BC_4,	nc16,	input,	X)," &			
"81	(BC_4,	pwrdelay,	input,	X)," &			
"82	(BC_1,	nc17,	output3,	X,	65,	1,	Z)," &
"83	(BC_1,	nc19,	output3,	X,	65,	1,	Z)," &
"84	(BC_1,	nc18,	output3,	X,	85,	1,	Z)," &
"85	(BC_1,	*,	control,	1)," &			
"86	(BC_1,	clk(3),	output3,	X,	111,	1,	Z)," &
"87	(BC_1,	clk(2),	output3,	X,	111,	1,	Z)," &
"88	(BC_1,	clk(1),	output3,	X,	111,	1,	Z)," &
"89	(BC_1,	clk(0),	output3,	X,	111,	1,	Z)," &
"90	(BC_1,	clkout,	output3,	X,	111,	1,	Z)," &
"91	(CBSC_1,	dq(35),	bidir,	X,	95,	0,	Z)," &
"92	(CBSC_1,	dq(0),	bidir,	X,	93,	0,	Z)," &
"93	(BC_1,	*,	control,	0)," &			
"94	(CBSC_1,	dq(32),	bidir,	X,	95,	0,	Z)," &
"95	(BC_1,	*,	control,	0)," &			
"96	(CBSC_1,	dq(2),	bidir,	X,	93,	0,	Z)," &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"97	(CBSC_1,	dq(36),	bidir,	X,	95,	0,	Z), " &
"98	(CBSC_1,	dq(33),	bidir,	X,	95,	0,	Z), " &
"99	(CBSC_1,	dq(1),	bidir,	X,	93,	0,	Z), " &
"100	(CBSC_1,	dq(34),	bidir,	X,	95,	0,	Z), " &
"101	(CBSC_1,	dq(37),	bidir,	X,	95,	0,	Z), " &
"102	(CBSC_1,	dq(4),	bidir,	X,	93,	0,	Z), " &
"103	(CBSC_1,	dq(39),	bidir,	X,	95,	0,	Z), " &
"104	(CBSC_1,	dq(41),	bidir,	X,	95,	0,	Z), " &
"105	(CBSC_1,	dq(38),	bidir,	X,	95,	0,	Z), " &
"106	(CBSC_1,	dq(3),	bidir,	X,	93,	0,	Z), " &
"107	(CBSC_1,	dq(43),	bidir,	X,	108,	0,	Z), " &
"108	(BC_1,	*,	control,	0), " &			
"109	(CBSC_1,	dq(40),	bidir,	X,	95,	0,	Z), " &
"110	(BC_1,	*,	control,	1), " &			
"111	(BC_1,	*,	control,	1), " &			
"112	(CBSC_1,	dq(5),	bidir,	X,	202,	0,	Z), " &
"113	(CBSC_1,	dq(6),	bidir,	X,	202,	0,	Z), " &
"114	(CBSC_1,	dq(8),	bidir,	X,	202,	0,	Z), " &
"115	(CBSC_1,	dq(42),	bidir,	X,	204,	0,	Z), " &
"116	(CBSC_1,	dq(7),	bidir,	X,	202,	0,	Z), " &
"117	(CBSC_1,	dq(45),	bidir,	X,	204,	0,	Z), " &
"118	(CBSC_1,	dq(44),	bidir,	X,	204,	0,	Z), " &
"119	(CBSC_1,	dq(9),	bidir,	X,	202,	0,	Z), " &
"120	(CBSC_1,	dq(47),	bidir,	X,	204,	0,	Z), " &
"121	(CBSC_1,	dq(12),	bidir,	X,	202,	0,	Z), " &
"122	(CBSC_1,	dq(10),	bidir,	X,	202,	0,	Z), " &
"123	(CBSC_1,	dq(11),	bidir,	X,	202,	0,	Z), " &
"124	(CBSC_1,	dq(46),	bidir,	X,	204,	0,	Z), " &
"125	(CBSC_1,	dq(14),	bidir,	X,	202,	0,	Z), " &
"126	(CBSC_1,	dq(13),	bidir,	X,	202,	0,	Z), " &
"127	(CBSC_1,	scb(5),	bidir,	X,	200,	0,	Z), " &
"128	(CBSC_1,	dq(15),	bidir,	X,	202,	0,	Z), " &
"129	(CBSC_1,	scb(4),	bidir,	X,	200,	0,	Z), " &
"130	(BC_1,	sdqm(4),	output3,	X,	205,	1,	Z), " &
"131	(BC_1,	scasz,	output3,	X,	205,	1,	Z), " &
"132	(CBSC_1,	scb(0),	bidir,	X,	200,	0,	Z), " &
"133	(BC_1,	swez,	output3,	X,	205,	1,	Z), " &
"134	(CBSC_1,	scb(1),	bidir,	X,	200,	0,	Z), " &
"135	(BC_1,	sdqm(0),	output3,	X,	205,	1,	Z), " &
"136	(BC_1,	scez(1),	output3,	X,	205,	1,	Z), " &
"137	(BC_1,	sdqm(5),	output3,	X,	205,	1,	Z), " &
"138	(BC_1,	sdqm(1),	output3,	X,	205,	1,	Z), " &
"139	(BC_1,	scez(0),	output3,	X,	205,	1,	Z), " &
"140	(BC_1,	sa(0),	output3,	X,	205,	1,	Z), " &
"141	(BC_1,	srasz,	output3,	X,	205,	1,	Z), " &
"142	(BC_1,	sa(1),	output3,	X,	205,	1,	Z), " &
"143	(BC_1,	sa(2),	output3,	X,	205,	1,	Z), " &
"144	(BC_1,	sa(4),	output3,	X,	205,	1,	Z), " &
"145	(BC_1,	sa(3),	output3,	X,	205,	1,	Z), " &
"146	(BC_1,	sa(5),	output3,	X,	205,	1,	Z), " &
"147	(BC_1,	sa(6),	output3,	X,	205,	1,	Z), " &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"148	(BC_1,	sa(8),	output3,	X,	20 5,	1,	Z)," &
"149	(BC_1,	sa(7),	output3,	X,	20 5,	1,	Z)," &
"150	(BC_1,	sa(9),	output3,	X,	20 5,	1,	Z)," &
"151	(BC_1,	sba(0),	output3,	X,	20 5,	1,	Z)," &
"152	(BC_1,	sa(11),	output3,	X,	20 5,	1,	Z)," &
"153	(BC_1,	sa(10),	output3,	X,	20 5,	1,	Z)," &
"154	(BC_1,	sba(1),	output3,	X,	20 5,	1,	Z)," &
"155	(BC_4,	dclk _{in} ,	input,	X)," &			
"156	(BC_1,	sa(12),	output3,	X,	20 5,	1,	Z)," &
"157	(BC_1,	sdqm(2),	output3,	X,	20 5,	1,	Z)," &
"158	(BC_1,	sdqm(3),	output3,	X,	20 5,	1,	Z)," &
"159	(BC_1,	scke(0),	output3,	X,	20 5,	1,	Z)," &
"160	(BC_1,	sdqm(6),	output3,	X,	20 5,	1,	Z)," &
"161	(CBSC_1,	scb(2),	bidir,	X,	20 0,	0,	Z)," &
"162	(CBSC_1,	scb(3),	bidir,	X,	20 0,	0,	Z)," &
"163	(BC_1,	sdqm(7),	output3,	X,	20 5,	1,	Z)," &
"164	(BC_1,	sa(13),	output3,	X,	20 5,	1,	Z)," &
"165	(CBSC_1,	scb(6),	bidir,	X,	20 0,	0,	Z)," &
"166	(CBSC_1,	dq(16),	bidir,	X,	20 1,	0,	Z)," &
"167	(CBSC_1,	dq(17),	bidir,	X,	20 1,	0,	Z)," &
"168	(CBSC_1,	scb(7),	bidir,	X,	20 0,	0,	Z)," &
"169	(CBSC_1,	dq(48),	bidir,	X,	20 4,	0,	Z)," &
"170	(CBSC_1,	dq(18),	bidir,	X,	20 1,	0,	Z)," &
"171	(CBSC_1,	dq(49),	bidir,	X,	20 4,	0,	Z)," &
"172	(CBSC_1,	dq(19),	bidir,	X,	20 1,	0,	Z)," &
"173	(CBSC_1,	dq(50),	bidir,	X,	20 4,	0,	Z)," &
"174	(CBSC_1,	dq(20),	bidir,	X,	20 1,	0,	Z)," &
"175	(CBSC_1,	dq(51),	bidir,	X,	20 4,	0,	Z)," &
"176	(CBSC_1,	dq(21),	bidir,	X,	20 1,	0,	Z)," &
"177	(BC_1,	scke(1),	output3,	X,	20 5,	1,	Z)," &
"178	(CBSC_1,	dq(22),	bidir,	X,	20 1,	0,	Z)," &
"179	(CBSC_1,	dq(52),	bidir,	X,	20 4,	0,	Z)," &
"180	(CBSC_1,	dq(23),	bidir,	X,	20 1,	0,	Z)," &
"181	(CBSC_1,	dq(53),	bidir,	X,	20 4,	0,	Z)," &
"182	(CBSC_1,	dq(24),	bidir,	X,	20 1,	0,	Z)," &
"183	(CBSC_1,	dq(54),	bidir,	X,	20 4,	0,	Z)," &
"184	(CBSC_1,	dq(55),	bidir,	X,	20 4,	0,	Z)," &
"185	(CBSC_1,	dq(25),	bidir,	X,	20 1,	0,	Z)," &
"186	(CBSC_1,	dq(26),	bidir,	X,	20 1,	0,	Z)," &
"187	(CBSC_1,	dq(56),	bidir,	X,	20 4,	0,	Z)," &
"188	(CBSC_1,	dq(27),	bidir,	X,	20 1,	0,	Z)," &
"189	(CBSC_1,	dq(57),	bidir,	X,	20 4,	0,	Z)," &
"190	(CBSC_1,	dq(28),	bidir,	X,	20 1,	0,	Z)," &
"191	(CBSC_1,	dq(59),	bidir,	X,	20 4,	0,	Z)," &
"192	(CBSC_1,	dq(58),	bidir,	X,	20 4,	0,	Z)," &
"193	(CBSC_1,	dq(30),	bidir,	X,	20 1,	0,	Z)," &
"194	(CBSC_1,	dq(29),	bidir,	X,	20 1,	0,	Z)," &
"195	(CBSC_1,	dq(60),	bidir,	X,	20 4,	0,	Z)," &
"196	(CBSC_1,	dq(61),	bidir,	X,	20 4,	0,	Z)," &
"197	(CBSC_1,	dq(31),	bidir,	X,	20 1,	0,	Z)," &
"198	(CBSC_1,	dq(62),	bidir,	X,	20 4,	0,	Z)," &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"199	(CBSC_1,	dq(63),	bidir,	X,	204,	0,	Z), " &
"200	(BC_1,	*,	control,	0), " &			
"201	(BC_1,	*,	control,	0), " &			
"202	(BC_1,	*,	control,	0), " &			
"203	(BC_1,	*,	control,	0), " &			
"204	(BC_1,	*,	control,	0), " &			
"205	(BC_1,	*,	control,	1), " &			
"206	(CBSC_1,	s_ad(38),	bidir,	X,	325,	0,	Z), " &
"207	(CBSC_1,	s_ad(34),	bidir,	X,	325,	0,	Z), " &
"208	(CBSC_1,	s_ad(42),	bidir,	X,	325,	0,	Z), " &
"209	(CBSC_1,	s_ad(36),	bidir,	X,	325,	0,	Z), " &
"210	(CBSC_1,	s_ad(33),	bidir,	X,	325,	0,	Z), " &
"211	(CBSC_1,	s_ad(32),	bidir,	X,	325,	0,	Z), " &
"212	(CBSC_1,	s_ad(40),	bidir,	X,	325,	0,	Z), " &
"213	(CBSC_1,	s_ad(37),	bidir,	X,	325,	0,	Z), " &
"214	(CBSC_1,	s_ad(46),	bidir,	X,	324,	0,	Z), " &
"215	(CBSC_1,	s_ad(35),	bidir,	X,	324,	0,	Z), " &
"216	(CBSC_1,	s_ad(44),	bidir,	X,	324,	0,	Z), " &
"217	(CBSC_1,	s_ad(39),	bidir,	X,	324,	0,	Z), " &
"218	(CBSC_1,	s_ad(41),	bidir,	X,	324,	0,	Z), " &
"219	(CBSC_1,	s_ad(50),	bidir,	X,	324,	0,	Z), " &
"220	(CBSC_1,	s_ad(48),	bidir,	X,	324,	0,	Z), " &
"221	(CBSC_1,	s_ad(45),	bidir,	X,	324,	0,	Z), " &
"222	(CBSC_1,	s_ad(47),	bidir,	X,	323,	0,	Z), " &
"223	(CBSC_1,	s_ad(54),	bidir,	X,	323,	0,	Z), " &
"224	(CBSC_1,	s_ad(43),	bidir,	X,	323,	0,	Z), " &
"225	(CBSC_1,	s_ad(49),	bidir,	X,	323,	0,	Z), " &
"226	(CBSC_1,	s_ad(52),	bidir,	X,	323,	0,	Z), " &
"227	(CBSC_1,	s_ad(53),	bidir,	X,	323,	0,	Z), " &
"228	(CBSC_1,	s_ad(51),	bidir,	X,	323,	0,	Z), " &
"229	(CBSC_1,	s_ad(58),	bidir,	X,	323,	0,	Z), " &
"230	(CBSC_1,	s_ad(56),	bidir,	X,	322,	0,	Z), " &
"231	(CBSC_1,	s_ad(57),	bidir,	X,	322,	0,	Z), " &
"232	(CBSC_1,	s_ad(55),	bidir,	X,	322,	0,	Z), " &
"233	(CBSC_1,	s_ad(60),	bidir,	X,	322,	0,	Z), " &
"234	(CBSC_1,	s_ad(59),	bidir,	X,	322,	0,	Z), " &
"235	(CBSC_1,	s_ad(62),	bidir,	X,	322,	0,	Z), " &
"236	(CBSC_1,	s_ad(61),	bidir,	X,	322,	0,	Z), " &
"237	(CBSC_1,	s_cbez(5),	bidir,	X,	331,	0,	Z), " &
"238	(CBSC_1,	s_par64,	bidir,	X,	239,	0,	Z), " &
"239	(BC_1,	*,	control,	0), " &			
"240	(CBSC_1,	s_cbez(4),	bidir,	X,	331,	0,	Z), " &
"241	(CBSC_1,	s_req64z,	bidir,	X,	242,	0,	Z), " &
"242	(BC_1,	*,	control,	0), " &			
"243	(CBSC_1,	s_ad(63),	bidir,	X,	322,	0,	Z), " &
"244	(CBSC_1,	s_cbez(7),	bidir,	X,	331,	0,	Z), " &
"245	(CBSC_1,	s_ack64z,	bidir,	X,	246,	0,	Z), " &
"246	(BC_1,	*,	control,	0), " &			
"247	(CBSC_1,	s_ad(0),	bidir,	X,	329,	0,	Z), " &
"248	(CBSC_1,	s_ad(2),	bidir,	X,	329,	0,	Z), " &
"249	(CBSC_1,	s_cbez(6),	bidir,	X,	331,	0,	Z), " &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"250	(CBSC_1,	s_ad(8),	bidir,	X,	328,	0,	Z)," &
"251	(CBSC_1,	s_ad(3),	bidir,	X,	329,	0,	Z)," &
"252	(CBSC_1,	s_ad(4),	bidir,	X,	329,	0,	Z)," &
"253	(CBSC_1,	s_ad(7),	bidir,	X,	329,	0,	Z)," &
"254	(CBSC_1,	s_ad(1),	bidir,	X,	329,	0,	Z)," &
"255	(CBSC_1,	s_cbez(0),	bidir,	X,	330,	0,	Z)," &
"256	(CBSC_1,	s_ad(5),	bidir,	X,	329,	0,	Z)," &
"257	(CBSC_1,	s_ad(6),	bidir,	X,	328,	0,	Z)," &
"258	(CBSC_1,	s_ad(10),	bidir,	X,	328,	0,	Z)," &
"259	(BC_4,	s_m66en,	input,	X)," &			
"260	(CBSC_1,	s_ad(12),	bidir,	X,	328,	0,	Z)," &
"261	(CBSC_1,	s_ad(11),	bidir,	X,	328,	0,	Z)," &
"262	(CBSC_1,	s_ad(9),	bidir,	X,	329,	0,	Z)," &
"263	(CBSC_1,	s_ad(14),	bidir,	X,	328,	0,	Z)," &
"264	(CBSC_1,	s_serrz,	bidir,	X,	265,	0,	Z)," &
"265	(BC_1,	*,	control,	0)," &			
"266	(CBSC_1,	s_ad(15),	bidir,	X,	328,	0,	Z)," &
"267	(CBSC_1,	s_cbez(1),	bidir,	X,	330,	0,	Z)," &
"268	(CBSC_1,	s_ad(13),	bidir,	X,	328,	0,	Z)," &
"269	(CBSC_1,	s_pa r,	bidir,	X,	270,	0,	Z)," &
"270	(BC_1,	*,	control,	0)," &			
"271	(CBSC_1,	s_stopz,	bidir,	X,	280,	0,	Z)," &
"272	(CBSC_1,	s_perrz,	bidir,	X,	273,	0,	Z)," &
"273	(BC_1,	*,	control,	0)," &			
"274	(CBSC_1,	s_lockz,	bidir,	X,	275,	0,	Z)," &
"275	(BC_1,	*,	control,	0)," &			
"276	(CBSC_1,	s_trdyz,	bidir,	X,	280,	0,	Z)," &
"277	(CBSC_1,	s_framez,	bidir,	X,	278,	0,	Z)," &
"278	(BC_1,	*,	control,	0)," &			
"279	(CBSC_1,	s_devselz,	bidir,	X,	280,	0,	Z)," &
"280	(BC_1,	*,	control,	0)," &			
"281	(CBSC_1,	s_irdyz,	bidir,	X,	282,	0,	Z)," &
"282	(BC_1,	*,	control,	0)," &			
"283	(CBSC_1,	s_ad(16),	bidir,	X,	327,	0,	Z)," &
"284	(CBSC_1,	s_ad(18),	bidir,	X,	327,	0,	Z)," &
"285	(CBSC_1,	s_cbez(2),	bidir,	X,	330,	0,	Z)," &
"286	(CBSC_1,	s_ad(17),	bidir,	X,	327,	0,	Z)," &
"287	(CBSC_1,	s_ad(20),	bidir,	X,	327,	0,	Z)," &
"288	(CBSC_1,	s_ad(21),	bidir,	X,	327,	0,	Z)," &
"289	(CBSC_1,	s_ad(19),	bidir,	X,	327,	0,	Z)," &
"290	(CBSC_1,	s_ad(24),	bidir,	X,	327,	0,	Z)," &
"291	(CBSC_1,	s_ad(22),	bidir,	X,	327,	0,	Z)," &
"292	(CBSC_1,	s_ad(26),	bidir,	X,	326,	0,	Z)," &
"293	(CBSC_1,	s_ad(23),	bidir,	X,	326,	0,	Z)," &
"294	(CBSC_1,	s_cbez(3),	bidir,	X,	330,	0,	Z)," &
"295	(CBSC_1,	s_ad(28),	bidir,	X,	326,	0,	Z)," &
"296	(CBSC_1,	s_ad(25),	bidir,	X,	326,	0,	Z)," &
"297	(CBSC_1,	s_ad(30),	bidir,	X,	326,	0,	Z)," &
"298	(CBSC_1,	s_ad(27),	bidir,	X,	326,	0,	Z)," &
"299	(CBSC_1,	s_ad(29),	bidir,	X,	326,	0,	Z)," &
"300	(CBSC_1,	s_ad(31),	bidir,	X,	326,	0,	Z)," &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"301	(BC_1,	s_rstz,	output3,	X,	333,	1,	Z), " &
"302	(BC_1,	s_holdaz,	output3,	X,	321,	0,	Z), " &
"303	(BC_4,	s_holdz,	input,	X), " &			
"304	(BC_1,	s_gntz(0),	output3,	X,	321,	0,	Z), " &
"305	(BC_1,	s_clk(0),	output3,	X,	333,	1,	Z), " &
"306	(BC_1,	s_clk(1),	output3,	X,	333,	1,	Z), " &
"307	(BC_1,	s_clk(2),	output3,	X,	333,	1,	Z), " &
"308	(BC_1,	s_clk(3),	output3,	X,	333,	1,	Z), " &
"309	(BC_1,	s_clk(4),	output3,	X,	333,	1,	Z), " &
"310	(BC_1,	s_clk(5),	output3,	X,	333,	1,	Z), " &
"311	(BC_1,	r_clkout,	output3,	X,	332,	1,	Z), " &
"312	(BC_4,	r_clkin,	input,	X), " &			
"313	(BC_4,	s_reqz(0),	input,	X), " &			
"314	(BC_4,	s_reqz(1),	input,	X), " &			
"315	(BC_4,	s_reqz(2),	input,	X), " &			
"316	(BC_1,	s_gntz(1),	output3,	X,	321,	0,	Z), " &
"317	(BC_1,	s_gntz(2),	output3,	X,	321,	0,	Z), " &
"318	(BC_4,	s_reqz(3),	input,	X), " &			
"319	(BC_1,	s_gntz(3),	output3,	X,	321,	0,	Z), " &
"320	(BC_4,	s_reqz(4),	input,	X), " &			
"321	(BC_1,	*,	control,	0), " &			
"322	(BC_1,	*,	control,	0), " &			
"323	(BC_1,	*,	control,	0), " &			
"324	(BC_1,	*,	control,	0), " &			
"325	(BC_1,	*,	control,	0), " &			
"326	(BC_1,	*,	control,	0), " &			
"327	(BC_1,	*,	control,	0), " &			
"328	(BC_1,	*,	control,	0), " &			
"329	(BC_1,	*,	control,	0), " &			
"330	(BC_1,	*,	control,	0), " &			
"331	(BC_1,	*,	control,	0), " &			
"332	(BC_1,	*,	control,	1), " &			
"333	(BC_1,	*,	control,	1), " &			
"334	(BC_1,	s_gntz(4),	output3,	X,	337,	0,	Z), " &
"335	(BC_4,	s_reqz(5),	input,	X), " &			
"336	(BC_1,	s_gntz(5),	output3,	X,	337,	0,	Z), " &
"337	(BC_1,	*,	control,	0), " &			
"338	(BC_1,	p_intz(1),	output3,	X,	339,	0,	Z), " &
"339	(BC_1,	*,	control,	0), " &			
"340	(BC_4,	p_gntz,	input,	X), " &			
"341	(CBSC_1,	p_ad(28),	bidir,	X,	455,	0,	Z), " &
"342	(BC_4,	p_rstz,	input,	X), " &			
"343	(BC_1,	p_intz(3),	output3,	X,	344,	0,	Z), " &
"344	(BC_1,	*,	control,	0), " &			
"345	(CBSC_1,	p_ad(30),	bidir,	X,	455,	0,	Z), " &
"346	(BC_1,	p_intz(2),	output3,	X,	347,	0,	Z), " &
"347	(BC_1,	*,	control,	0), " &			
"348	(BC_1,	p_intz(0),	output3,	X,	349,	0,	Z), " &
"349	(BC_1,	*,	control,	0), " &			
"350	(CBSC_1,	p_ad(31),	bidir,	X,	455,	0,	Z), " &
"351	(BC_1,	p_reqz,	output3,	X,	352,	0,	Z), " &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"352	(BC_1,	*,	control,	0)," &			
"353	(CBSC_1,	p_ad(26),	bidir,	X,	455,	0,	Z)," &
"354	(CBSC_1,	p_ad(29),	bidir,	X,	455,	0,	Z)," &
"355	(CBSC_1,	p_ad(22),	bidir,	X,	454,	0,	Z)," &
"356	(BC_4,	p_idsel,	input,	X)," &			
"357	(CBSC_1,	p_ad(25),	bidir,	X,	455,	0,	Z)," &
"358	(CBSC_1,	p_ad(24),	bidir,	X,	455,	0,	Z)," &
"359	(CBSC_1,	p_ad(27),	bidir,	X,	455,	0,	Z)," &
"360	(CBSC_1,	p_ad(20),	bidir,	X,	454,	0,	Z)," &
"361	(CBSC_1,	p_ad(23),	bidir,	X,	454,	0,	Z)," &
"362	(CBSC_1,	p_cbez(3),	bidir,	X,	453,	0,	Z)," &
"363	(CBSC_1,	p_ad(21),	bidir,	X,	454,	0,	Z)," &
"364	(CBSC_1,	p_ad(18),	bidir,	X,	454,	0,	Z)," &
"365	(CBSC_1,	p_ad(19),	bidir,	X,	454,	0,	Z)," &
"366	(CBSC_1,	p_ad(16),	bidir,	X,	454,	0,	Z)," &
"367	(CBSC_1,	p_ad(17),	bidir,	X,	454,	0,	Z)," &
"368	(CBSC_1,	p_framez,	bidir,	X,	369,	0,	Z)," &
"369	(BC_1,	*,	control,	0)," &			
"370	(CBSC_1,	p_cbez(2),	bidir,	X,	453,	0,	Z)," &
"371	(CBSC_1,	p_stopz,	bidir,	X,	373,	0,	Z)," &
"372	(CBSC_1,	p_trdyz,	bidir,	X,	373,	0,	Z)," &
"373	(BC_1,	*,	control,	0)," &			
"374	(CBSC_1,	p_devselz,	bidir,	X,	373,	0,	Z)," &
"375	(CBSC_1,	p_jrdyz,	bidir,	X,	376,	0,	Z)," &
"376	(BC_1,	*,	control,	0)," &			
"377	(CBSC_1,	p_ad(15),	bidir,	X,	452,	0,	Z)," &
"378	(CBSC_1,	p_par,	bidir,	X,	379,	0,	Z)," &
"379	(BC_1,	*,	control,	0)," &			
"380	(CBSC_1,	p_perrz,	bidir,	X,	381,	0,	Z)," &
"381	(BC_1,	*,	control,	0)," &			
"382	(BC_4,	p_lockz,	input,	X)," &			
"383	(CBSC_1,	p_ad(11),	bidir,	X,	452,	0,	Z)," &
"384	(CBSC_1,	p_ad(13),	bidir,	X,	452,	0,	Z)," &
"385	(CBSC_1,	p_cbez(1),	bidir,	X,	453,	0,	Z)," &
"386	(CBSC_1,	p_serrz,	bidir,	X,	387,	0,	Z)," &
"387	(BC_1,	*,	control,	0)," &			
"388	(CBSC_1,	p_cbez(0),	bidir,	X,	453,	0,	Z)," &
"389	(CBSC_1,	p_ad(9),	bidir,	X,	452,	0,	Z)," &
"390	(CBSC_1,	p_ad(12),	bidir,	X,	452,	0,	Z)," &
"391	(CBSC_1,	p_ad(14),	bidir,	X,	452,	0,	Z)," &
"392	(CBSC_1,	p_ad(10),	bidir,	X,	452,	0,	Z)," &
"393	(BC_4,	p_m66en,	input,	X)," &			
"394	(CBSC_1,	p_ad(6),	bidir,	X,	452,	0,	Z)," &
"395	(CBSC_1,	p_ad(4),	bidir,	X,	451,	0,	Z)," &
"396	(CBSC_1,	p_ad(8),	bidir,	X,	451,	0,	Z)," &
"397	(CBSC_1,	p_ad(7),	bidir,	X,	451,	0,	Z)," &
"398	(BC_4,	p_clk,	input,	X)," &			
"399	(CBSC_1,	p_ad(2),	bidir,	X,	451,	0,	Z)," &
"400	(CBSC_1,	p_ad(5),	bidir,	X,	451,	0,	Z)," &
"401	(CBSC_1,	p_ad(3),	bidir,	X,	451,	0,	Z)," &
"402	(CBSC_1,	p_ad(0),	bidir,	X,	451,	0,	Z)," &



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"403	(CBSC_1,	p_req64z,	bidir,	X,	404,	0,	Z)," &
"404	(BC_1,	*,	control,	0)," &			
"405	(CBSC_1,	p_ad(1),	bidir,	X,	451,	0,	Z)," &
"406	(CBSC_1,	p_ack64z,	bidir,	X,	407,	0,	Z)," &
"407	(BC_1,	*,	control,	0)," &			
"408	(CBSC_1,	p_cbez(7),	bidir,	X,	450,	0,	Z)," &
"409	(CBSC_1,	p_cbez(5),	bidir,	X,	450,	0,	Z)," &
"410	(CBSC_1,	p_cbez(6),	bidir,	X,	450,	0,	Z)," &
"411	(CBSC_1,	p_par64,	bidir,	X,	412,	0,	Z)," &
"412	(BC_1,	*,	control,	0)," &			
"413	(CBSC_1,	p_cbez(4),	bidir,	X,	450,	0,	Z)," &
"414	(CBSC_1,	p_ad(62),	bidir,	X,	449,	0,	Z)," &
"415	(CBSC_1,	p_ad(63),	bidir,	X,	449,	0,	Z)," &
"416	(CBSC_1,	p_ad(60),	bidir,	X,	449,	0,	Z)," &
"417	(CBSC_1,	p_ad(59),	bidir,	X,	449,	0,	Z)," &
"418	(CBSC_1,	p_ad(61),	bidir,	X,	449,	0,	Z)," &
"419	(CBSC_1,	p_ad(57),	bidir,	X,	449,	0,	Z)," &
"420	(CBSC_1,	p_ad(58),	bidir,	X,	449,	0,	Z)," &
"421	(CBSC_1,	p_ad(56),	bidir,	X,	449,	0,	Z)," &
"422	(CBSC_1,	p_ad(55),	bidir,	X,	448,	0,	Z)," &
"423	(CBSC_1,	p_ad(53),	bidir,	X,	448,	0,	Z)," &
"424	(CBSC_1,	p_ad(52),	bidir,	X,	448,	0,	Z)," &
"425	(CBSC_1,	p_ad(54),	bidir,	X,	448,	0,	Z)," &
"426	(CBSC_1,	p_ad(51),	bidir,	X,	448,	0,	Z)," &
"427	(CBSC_1,	p_ad(49),	bidir,	X,	448,	0,	Z)," &
"428	(CBSC_1,	p_ad(50),	bidir,	X,	448,	0,	Z)," &
"429	(CBSC_1,	p_ad(48),	bidir,	X,	448,	0,	Z)," &
"430	(CBSC_1,	p_ad(46),	bidir,	X,	447,	0,	Z)," &
"431	(CBSC_1,	p_ad(47),	bidir,	X,	447,	0,	Z)," &
"432	(CBSC_1,	p_ad(44),	bidir,	X,	447,	0,	Z)," &
"433	(CBSC_1,	p_ad(45),	bidir,	X,	447,	0,	Z)," &
"434	(CBSC_1,	p_ad(43),	bidir,	X,	447,	0,	Z)," &
"435	(CBSC_1,	p_ad(41),	bidir,	X,	447,	0,	Z)," &
"436	(CBSC_1,	p_ad(42),	bidir,	X,	447,	0,	Z)," &
"437	(CBSC_1,	p_ad(38),	bidir,	X,	447,	0,	Z)," &
"438	(CBSC_1,	p_ad(37),	bidir,	X,	446,	0,	Z)," &
"439	(CBSC_1,	p_ad(40),	bidir,	X,	446,	0,	Z)," &
"440	(CBSC_1,	p_ad(34),	bidir,	X,	446,	0,	Z)," &
"441	(CBSC_1,	p_ad(39),	bidir,	X,	446,	0,	Z)," &
"442	(CBSC_1,	p_ad(35),	bidir,	X,	446,	0,	Z)," &
"443	(CBSC_1,	p_ad(33),	bidir,	X,	446,	0,	Z)," &
"444	(CBSC_1,	p_ad(32),	bidir,	X,	446,	0,	Z)," &
"445	(CBSC_1,	p_ad(36),	bidir,	X,	446,	0,	Z)," &
"446	(BC_1,	*,	control,	0)," &			
"447	(BC_1,	*,	control,	0)," &			
"448	(BC_1,	*,	control,	0)," &			
"449	(BC_1,	*,	control,	0)," &			
"450	(BC_1,	*,	control,	0)," &			
"451	(BC_1,	*,	control,	0)," &			
"452	(BC_1,	*,	control,	0)," &			
"453	(BC_1,	*,	control,	0)," &			



#	Cell Type	Name	Function	Safe bit	Control Signal	Disable Value	Disable Result
"454	(BC_1,	*,	control,	0)," &			
"455	(BC_1,	*,	control,	0)";			

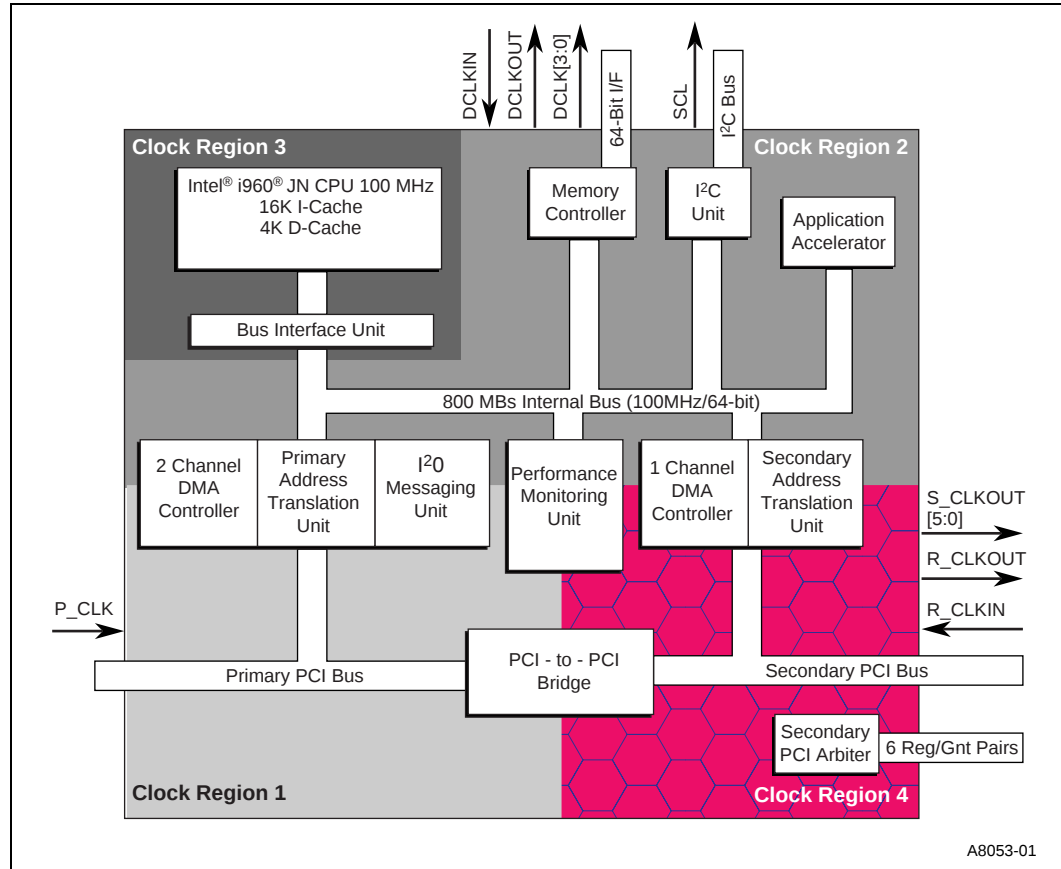
Affected Docs: *Intel® 80303 I/O Processor Developer's Manual.*



9. Figure 25-1 on page 25-1 has incorrect data

Problem: The Internal bus in diagram shows 66 MHz bus speed. The actual bus speed is 100 MHz

Workaround: Replace Figure 25-1 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual

10. Section 25.1.3 on page 25-2

Problem: The third sentence of the first paragraph is incorrect. It states the maximum bus speed of the region is 66 MHz. It is actually 100 MHz.

Workaround: Change the third sentence to the following:

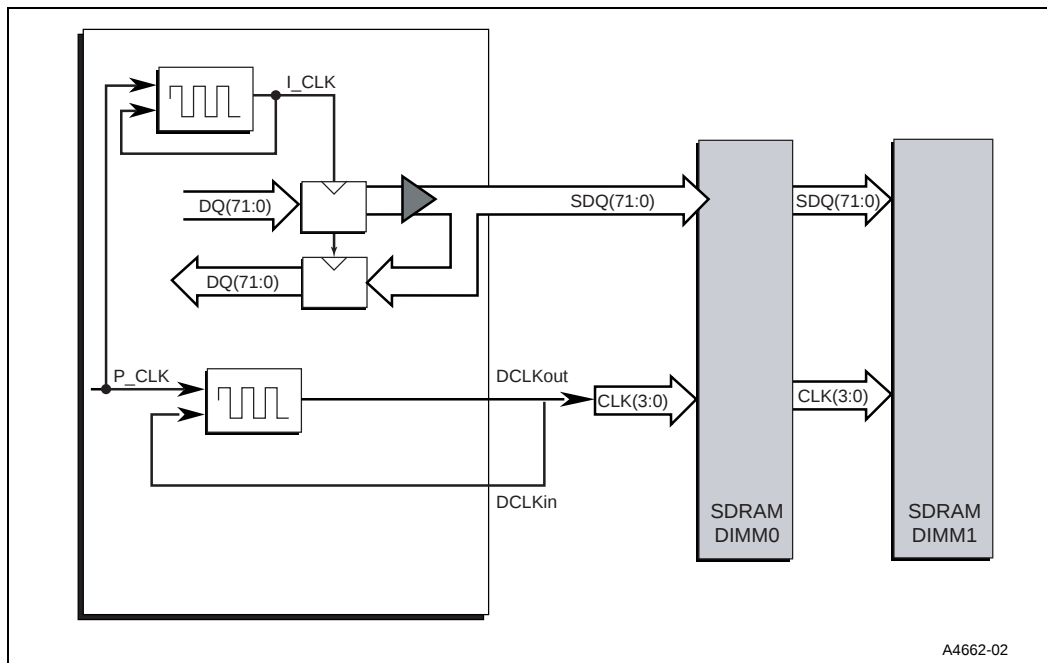
"It supports clock frequencies up to a maximum of 100 MHz."

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

11. Figure 25-2 on page 25-2 did not print correctly

Problem: Figure 25-2 on page 25-2 did not print correctly.

Workaround: Replace Figure 25-2 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual.

12. Table 25-2 on page 25-3 did not print completely

Problem: Table 25-2 on page 25-3 did not print completely

Workaround: Replace Table 25-2 with the following:

Input Clock	Region/Clock	Buffered/PLL	P_M66EN	S_M66EN
P_CLK = 33 MHz	Region 1: 1x P_CLK	Buffered	0	0
	Region 2: 3x P_CLK	PLL		
	Region 3: 3x P_CLK	PLL		
	Region 4: 1x P_CLK	Buffered		
P_CLK = 66 MHz	Region 1: 1x P_CLK	Buffered	1	1
	Region 2: 3/2x P_CLK	PLL		
	Region 3: 3/2x P_CLK	PLL		
	Region 4: 1x P_CLK	Buffered		
P_CLK = 66 MHz	Region 1: 1x P_CLK	Buffered	1	0
	Region 2: 3/2x P_CLK	PLL		
	Region 3: 3/2x P_CLK	PLL		
	Region 4: 1/2x P_CLK	Buffered		

Note: Combination of P_M66EN=0 and S_M66EN=1 is not supported by the Intel® 80303 I/O processor. When P_M66EN=0, the 80303 I/O processor forces S_M66EN=0 ensuring the unsupported condition never occurs.

Affected Docs: Intel® 80303 I/O Processor Developer's Manual



13. Section 1.2.2 on page 1-2 has incorrect data

Problem: The second sentence of the first paragraph is incorrect. It states the Internal Bus operates at 66 MHz. It is actually 100 MHz.

Workaround: Change the second sentence to the following:

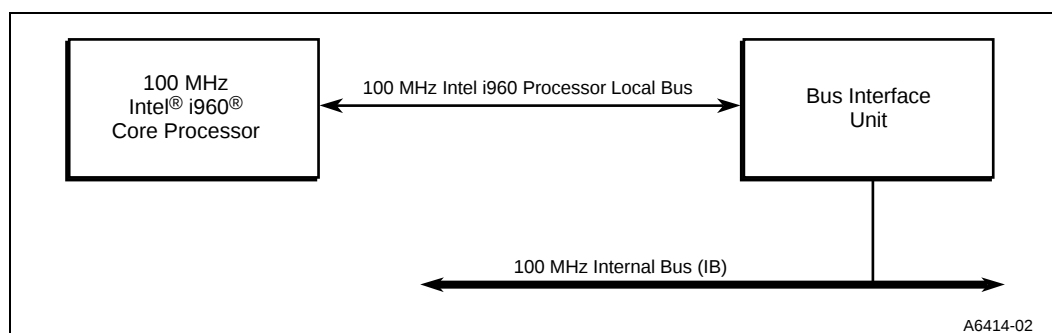
"The Internal Bus operates at 100 MHz and is 64 bits wide."

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

14. Figure 12-2 on page 12-10 has incorrect data

Problem: The Internal bus in diagram shows 66 MHz bus speed. The actual bus speed is 100 MHz

Workaround: Replace Figure 12-2 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual

15. Section 19.1 on page 19-1 has incorrect data

Problem: The last bullet incorrectly states, '64-bit/66MHz PCI and 80303 I/O processor internal bus interface.' The internal bus on the 80303 I/O processor is 100 MHz.

Workaround: Change the last bullet to the following: '64-bit/66MHz PCI and 64-bit/100 MHz internal bus interface.'

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

16. Table 14-46 on page 14-109 has missing data

Problem: Table 14-46 is missing the bit description for bit 12. Add the following:

Bit	Default	Description
12	Varies with inverse of the external state of RAD[2]/SPME M# at Primary PCI bus reset	Special Downstream Window Enable - When set, a special downstream memory window which includes the addresses FEC0_0000h through FECF_FFFFh is opened. This window provides support for an alternate address mechanism to a Hot-Plug Controller.

Workaround: When clear, the Special Downstream Memory window is closed.

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

17. Section 13.2.4.3 on page 13-30 has incorrect data

Problem: The first sentence incorrectly states, 'If enabled'. ECC is always enabled on the 80303 I/O processor, it is not optional.

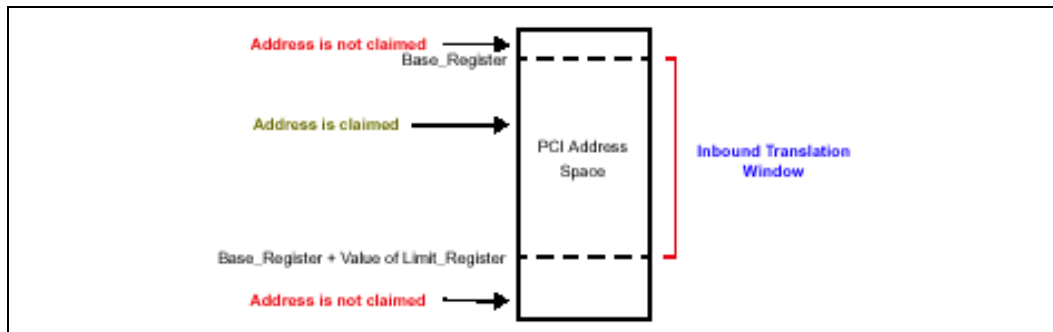
Workaround: Remove 'If enabled'.

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

18. Figure 15-3 on page 15-7 has missing text

Problem: The figure shows 'se_Register + Value of Limit_Register'. It should be 'Base_Register + Value of Limit_Register'.

Workaround: Replace Figure 15-3 with the following:



Affected Docs: Intel® 80303 I/O Processor Developer's Manual

19. Section 15.7.39 on page 15-100 has incorrect data

Problem: The last paragraph is incorrect. It states, 'Note that bits 4:0, bits 12:11, bit 9 and bit 7 can result in an NMI# interrupt driven to the i960 core processor.' Bit 12 is a reserved bit, so it should be removed from this sentence.

Workaround: Change the last paragraph to the following: 'Note that bits 4:0, 11, 9 and 7 can result in an NMI# interrupt driven to the i960 core processor.'

Affected Docs: Intel® 80303 I/O Processor Developer's Manual



20. Table 8-17 on page 8-38 has incorrect data

Problem: The bit locations for External Interrupt 5 are incorrectly shown as bits '9:4'. It should be '7:4'.

Workaround: Replace Table 8-17 with the following:

		31	28	24	20	16	12	8	4	0	
IOP Attributes	[	rv	rv	rv	rv	rv	rv	rv	rv	rv	
		rv	rv	rv	rv	rv	rv	rv	rv	rv	
PCI Attributes	[	na	na	na	na	na	na	na	na	na	
		na	na	na	na	na	na	na	na	na	
Intel® i960® Core internal bus address		FF00 8524H									
IMAP1											
		Attribute Legend:									
		RV = Reserved									
		PR = Preserved									
		RS = Read/Set									
		RW = Read/Write									
		RC = Read Clear									
		RO = Read Only									
		NA= Not Accessible									
Bit	Default	Description									
31:16	Default Value loaded from image in Control Table	Reserved (initialize to 0)									
15:12		External Interrupt 7 Field - IMAP1.x7									
11:8		External Interrupt 6 Field - IMAP1.x6									
7:4		External Interrupt 5 Field - IMAP1.x5									
3:0		External Interrupt 4 Field - IMAP1.x4									

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

21. Section 11.2.8 on page 11-5 has incorrect data

Problem: The last sentence in the third paragraph states, 'Specifications for a cold and warm reset can be found in the 80960RM I/O Processor Data Sheet and the 80960RN I/O Processor Data Sheet.' This sentence should be removed, it does not pertain to the 80303 I/O processor.

Workaround: Change text to the following: 'The 80303 I/O processor complies with the *PCI Local Bus Specification*, Revision 2.2. Reset parameters are defined in this specification.'

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

22. Section 13.2.3.1 on page 13-13 has incorrect data

Problem: The first sentence states, 'The MCU supports an ECC only memory subsystem ranging from 32 to 528 Mbytes.' It should be 512 Mbytes, not 528 Mbytes.

Workaround: Change this sentence to the following: 'The MCU supports an ECC only memory subsystem ranging from 32 to 512 Mbytes.'

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

23. Table 13-4 on page 13-9 has incorrect data

Problem: Table 13-4 lists incorrect wait states for the flash bus.

Workaround: Replace Table 13-4 with the following:

Flash Speed	Address-to-Data Wait States	Recovery Wait States
<= 55 ns	8	4
<= 115 ns	12	4
<= 175 ns	20	4

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

24. Table 8-15 on page 8-36 needs clarification

Problem: ICON.10, Global Interrupt Enable bit, does not state what bit value enables interrupts.

Workaround: Add this sentence to the bit description, 'A '0' will globally enable interrupts, and a '1' globally disables interrupts.'

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

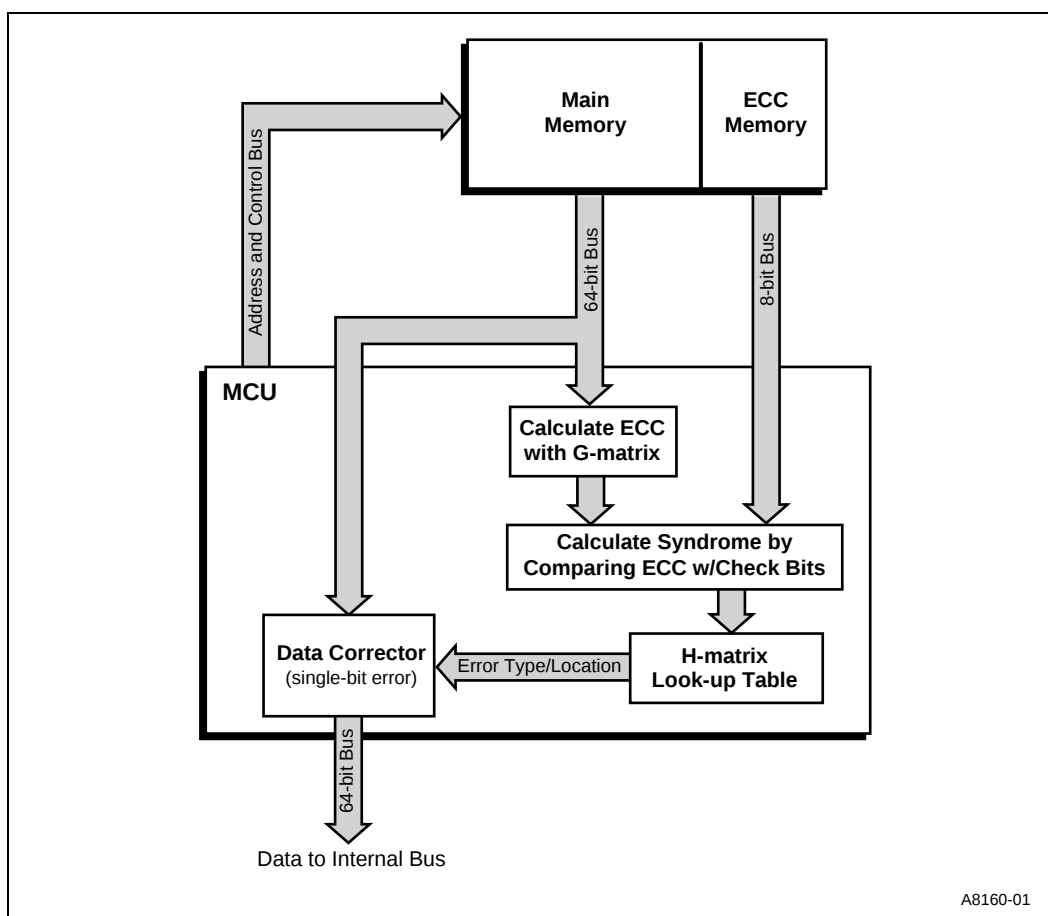
25. Table 13-13 on page 13-30 has incorrect data

Problem: Syndrome Decoding Error Types and Symptoms are incorrectly stated.

Workaround: Replace Table 13-13 with the following and, add the adjacent paragraph with "new" Figure 13-16:

Error Type	Symptom
None	The syndrome is 0000 0000.
Single-Bit	Use the H-Matrix in Figure 13-17 to determine which bit the MCU will invert to fix the error.
Multi-Bit	If the Syndrome does not match an 8-bit value in the H-matrix, the error is uncorrectable

Figure 13-16 shows how the data flows through the ECC hardware for a read transaction.



Affected Docs: Intel® 80303 I/O Processor Developer's Manual



26. Section 13.2.4.3, First Paragraph after Table 13-13 has Incorrect Data

Problem: First sentence incorrectly states error types for corrected Table 13-13:
...If decoding the syndrome indicates a double-bit or nibble error...

Should read as follows:
..."When" decoding the syndrome indicates a "multi"-bit error...

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*

27. Section 13.2.4.3, First Paragraph after "Current" Figure 13-16. H-Matrix has Incorrect Data

Problem: First sentence incorrectly states error types for corrected Table 13-13:
...If error reporting is enabled in the ECCR and the MCU detects a nibble, single-bit, or double-bit error...

Should read as follows:
..."When" error reporting is enabled in the ECCR and the MCU detects a single-bit or "multi"-bit error...

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*

28. Section 11.3.1.5 FAIL# Code

Problem: The verbiage in this section is residual from the *Intel® i960® I/O Processor Developer's Manual*, where the internal bus was accessible from the outside. The internal bus is not accessible from the outside for i960 RM/RN I/O processor. Since the customer cannot "see" the internal bus, whatever is on it is not useful and is only confusing. Therefore, this section has been removed.

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*

29. Section 13.5 Reset Conditions has Incorrect Data

Problem: The last sentence in the first paragraph incorrectly states:
Reads issued prior to a write to the same address results in an ECC error (**if enabled**) and **is** not recommended.

This should state:
Reads issued prior to a write to the same address results in an ECC error and **are** not recommended.

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*

30. Section 13.2.4.2, First Sentence has Incorrect Data

Problem: On page 13-28, the first sentence reads: "If the internal bus master writes less than the data bus width programmed in the SDCR, then the MCU translates the write transaction into a read-modify-write transaction." Please remove "programmed in the SDCR" from the sentence.

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*

31. Section 13.6.2, Second Sentence has Incorrect Data

Problem: On page 13-46, the second sentence reads: "The SDCR specifies the drive strength for the MCU pins, the bus width, and power failure handling." Please "remove, the bus width, and power failure handling" from the sentence. It should read "The SDCR specifies the drive strength for the MCU pins."

Affected Docs: *Intel® 80303 I/O Processor Developer's Manual*



32. Section 4.5.2 on page 50 is only correct for A-0 and A-1 steppings

Problem: The second sentence in Note 7 states, 'S_REQ64# is deasserted one P_CLK after the deassertion of S_RST#'. This statement is not correct for the A-2 stepping of the 80303 and 80302 I/O processors.

Workaround: This statement is only correct for the A-0 and A-1 steppings of the 80303. See Specification Clarification #6 for A-2 stepping functionality.

Affected Docs: Intel® 80303 I/O Processor Datasheet

33. Section 17.5.1 on page 17-12 is only correct for A-0 and A-1 steppings

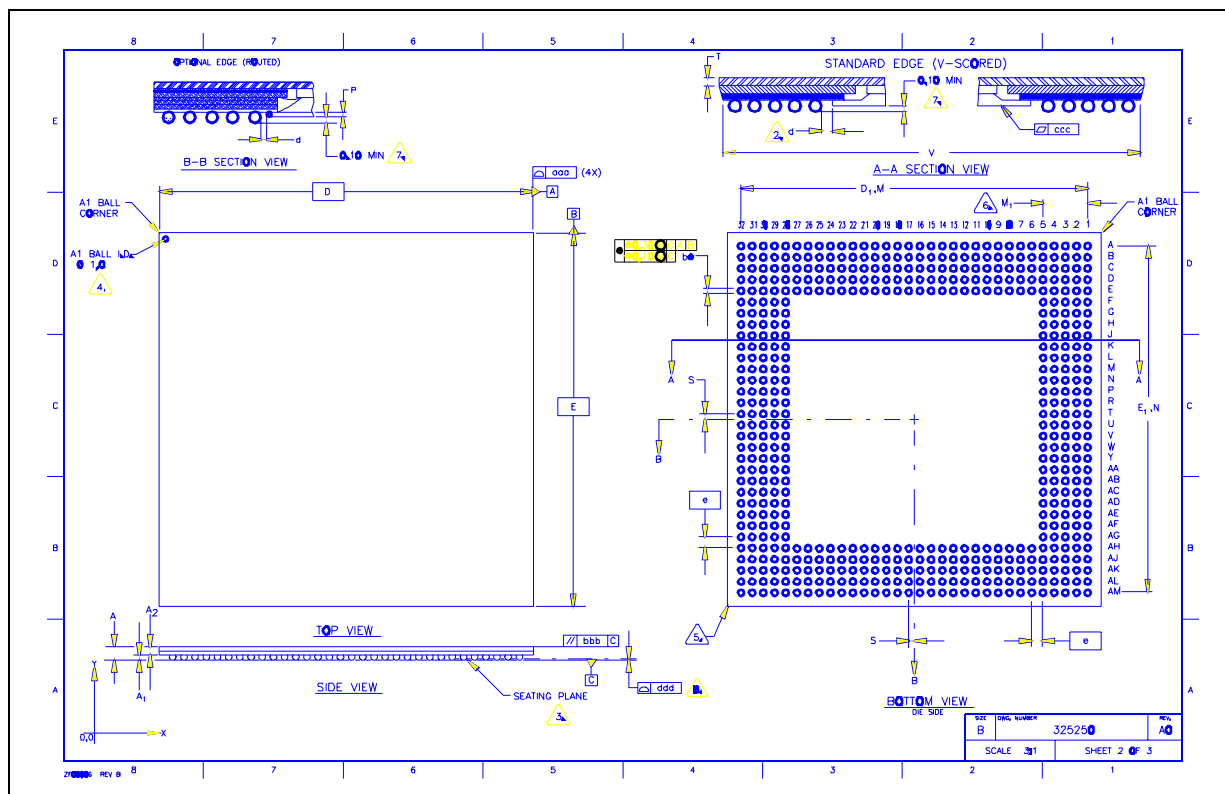
Problem: The last sentence states, 'S_REQ64# remains valid for one clock (P_CLK) after S_RST# deasserts'. This statement is not correct for the A-2 stepping of the 80303 and 80302 I/O processors.

Workaround: This statement is only correct for the A-0 and A-1 steppings of the 80303. See Specification Clarification #6 for A-2 stepping functionality.

Affected Docs: Intel® 80303 I/O Processor Developer's Manual

34. Package change for the Intel® 80303/80302/80312 I/O processors products and lists the new order codes

Problem: Product Change Notification 107318-02 describes a package change for the Intel® 80303/80302/80312 I/O processor products and lists the new order codes. The new SBGA (Super Ball Grid Array) package has the same ballout, functionality, and is still 42.5mm x 42.5mm, but the height and look of the package are different. The SBGA is a thinner package and has a full body heat spreader.





Standard Body Size Dimension Table

Body Size	42.5 x 42.5mm Package			Body Size
Symbol	Minimum	Nominal	Maximum	Notes
A	— — —	— — —	1.70	Overall Thickness
A1	0.50	0.60	0.70	Ball Height
A2	0.80	0.91	1.00	Body Thickness
D	— — —	42.50	— — —	Body Size
D1	39.27	39.37	39.47	Ball Footprint
E	— — —	42.50	— — —	Body Size
E1	39.27	39.37	39.47	Ball Footprint
M , N	32 x 32			Ball Matrix
M1 	5			Number Of Rows Deep
b	0.60	0.75	0.90	Ball Diameter
d	0.25			Minimum Distance Encap To Balls
e	1.27			Ball Pitch
aaa			0.20	Package Body Profile
bbb			0.25	Parallel
ccc			0.10	Encap Flatness Over Die
ddd 			0.20	Coplanarity
S	— — —	0.635	— — —	Solder Ball Placement
T	0.050	0.125	0.175	V-Score Web Thickness
V	41.7	— — —	42.4	V-Score Bottom Size

Affected Docs: Intel® 80303 I/O Processor Datasheet

§ §

