PCI 9080-3

Errata Rev. 1.4
March 2002

Errata Documentation

A. Affected Silicon Revision

This document details errata in the following silicon:

Product	Revision	Samples	Production
PCI 9080	Version 03	October 1997	January 1998

B. Documentation Revision

The following documentation is the baseline functional description of the silicon. Errata are defined as behaviors in the affected silicon that do not match behaviors detailed in this documentation.

Document	Revision	Description	Date
Data Book	Version 1.06	PCI 9080 Data Book	January 2000
PCI 9080 Design Notes Documentation	See www.plxtech.com for latest revision	Design Notes Documentation	

C. Errata Summary for the PCI 9080 is as follows:

#	Description
1	EOTx# Pin Usage
2	LLOCKo# (Local Lock output) assertions with Direct Slave Reads
3	Demand Mode DMA in Stop Transfer Mode
4	Unaligned DMA In Big Endian Mode
5	Direct Slave Non-PCI 2.1 and Read Ahead modes, Write Flush mode (revised from PCI 9080 Errata revision 1.3)
6	Direct Master Burst Read past a 64K boundary
7	Simultaneous PCI DMA Descriptor read with local access to internal registers
8	Aborting chained DMA with descriptors on both PCI and local buses
9	Delayed read during 32K PCI clocks timeout
10	LOCK# negation during an idle phase (retracted)
11	DMA Channel 0 Almost Full Threshold value of zero with Local-to-PCI transfers
12	Direct Slave Read Retry
13	Direct Slave Prefetch Burst Read over the Space Boundary
14	PCI Target Abort during DMA Transfer

Errata Summary for the PCI 9080 (continued):

#	Description
15	Local Bus input pin minimum Hold times and output pin Maximum Output Delay specifications (text revised from Errata rev. 1.3, values are unchanged)
16	Messaging Unit data corruption if Queue Prefetch (Inbound Free List FIFO Prefetch and/or Outbound Post List FIFO Prefetch) is enabled
17	Simultaneous Write to Queue Register in Messaging Unit
18	Demand Mode DMA resumes without DREQ# input assertion if the previous DREQ# negation occurs after PCI 9080 asserts BLAST# in response to a BREQ (Bus Request) input assertion
19	Demand Mode DMA Channel 0/1 rotation does not occur if Channel 0 owns the Bus when BREQ (Bus Request) input is used to suspend the DMA transfer
20	Internal pull-up/pull-down resistors
21	DMA Burst-4 Mode with Constant Address
22	J-mode DMA with Constant Address

D. Errata:

1. EOTx# Pin Usage (use any)

Erratum Issue: The use of EOTx# to terminate ongoing DMA transfers may result in corrupted data being transferred, premature indication of FIFO empty via the "DMA Done Bit", or may cause the PCI 9080 to perform an invalid read cycle in place of the Write Back cycle. This is true for aligned and unaligned transfers, and for Local-to-PCI and PCI-to-Local transfers.

Solution/Workaround:

1. Use the DMA engines in demand mode. Use the DREQ# pin to pause ongoing DMA transactions. DMA Abort can then be used to terminate the pending transfers once the DMA channel has paused.
2. Write to DMA control registers to pause ongoing DMA transfers. Use DMA Abort to terminate the DMA transfer.
3. Use of DMA when transferring to and from 32-bit buses must follow these conditions:
   a. The starting addresses on both the PCI and local sides must be 32-bit aligned.
   b. The transfer count must also be in 32-bit multiples.
   c. The transfers must not be in Memory Write and Invalidate or in Clear Count modes (DMAMODEx[16,13] = 00b).
4. Synchronize the PCI clock with the Local clock by using the inverted version of the BCLKo signal to drive LCLK. The inverter should have a propagation delay of less than 5ns.

2. LLOCKo# (Local Lock output) assertions with Direct Slave Reads

Erratum Issue: After the first Direct Slave Read transaction, subsequent Direct Slave Reads will cause the LLOCKo# signal to assert with the LHOLD signal. The PCI 9080 will act as if the two lines are shorted together. This has the consequence of locking the local processor every time.

Solutions/Workarounds:

1. Execute a Direct Slave Write cycle before any Direct Slave Reads. A Direct Slave Write cycle sets the internal state machine such that any number of subsequent Direct Slave Reads will not activate the LLOCKo# bug.
2. Discontinue use of the LLOCKo# pin.

Note: under PCI r2.2, the use of LOCK# has been restricted such that only a bridge is allowed to use lock to prevent a deadlock and for compatibility issues.

3. Demand Mode DMA in Stop Transfer Mode

Erratum Issue: This update only applies to Demand Mode DMA transfers when the Stop Transfer bit (DMAMODEx[15]) is active. If DREQ# is negated before all data transfers are complete (pausing the DMA), subsequent assertions of DREQ# will fail to transfer the last long word of the intended transfers. Furthermore, if chaining DMA is active, the PCI 9080 will fail to read the descriptor of the following link in the chain.

Solution/Workaround:

1. Always specify one extra long word to be transferred when attempting to pause Demand mode transfers using DREQ#.
2. Do not negate DREQ# until all intended transfers are complete.
3. Pause the DMA transfer by writing to the channel Enable bit rather than by using Demand mode.
4. Pause the DMA without the Stop Transfer bit enabled (DMAMODEx[15] =0).

4. Unaligned DMA In Big Endian Mode

Erratum Issue: Unaligned DMA cycles in Big Endian Mode may result in corrupted data being sent across the device.

Solution/Workaround:

(Convention: Local device refers to device that uses Big Endian byte ordering.)

1. To interface the PCI 9080 to a Big Endian device on the local bus, disable the Big Endian feature in the PCI 9080 and connect the data bits [0:31] of the local device directly to the data bit [0:31] of the Local bus and PCI 9080. Therefore, long word transfers are compatible between the two domains. If the local device writes a byte to address 0, the data appears on bits [24:31] and LBE3# to the PCI 9080 is asserted. This corresponds to byte address 3 on the PCI bus. If byte or word transfers to PCI 9080 are performed, then the local device software must map the address so that the data appears on the correct byte lane. Also, if byte or word data structures are shared by the local device and other PCI Little Endian devices, then address mapping in the software must be included.
2. Swap the byte lanes when connecting the local device to the PCI 9080 Local bus. In this case byte addressing is compatible, but word and Lword addressing is not.

5. Direct Slave Non-PCI 2.1 and Read Ahead modes, Write Flush mode (revised from Errata revision 1.3)

Erratum Issue: If the PCI 9080 Local/DMA arbitration register is set for non-PCI 2.1 mode (bit 24 = 0), Read Ahead mode enabled (bit 28 = 1), and Write Flushes Read mode disabled (bit 26 = 0), a Direct Slave write will cause the Local bus to stop reading data. The Local bus never resumes reading data, causing all read attempts to be retried on the PCI bus.

Solution/Workaround: (use all)

1. Set the PCI9080 to operate in PCI 2.1 mode (MARBR[24] = 1).
2. Disable Read Ahead mode (MARBR[28] = 0).
3. Enable PCI Read with Write Flush mode (MARBR[26] = 1).

6. Direct Master Burst Read past a 64K boundary

Erratum Issue: This erratum prevents the PCI 9080 from prefetching read data past a 64K boundary (low 16 bits of the PCI address). This implies that a Direct Master read must not request burst data past this boundary. If it does, the PCI 9080 will hang the Local bus at the boundary, refusing to gather any further data.

Solution/Workaround:

This 64K boundary is not a hard wall. It can be crossed in 2 separate cycles. It only cannot be crossed in a single burst cycle. Any of the following solutions may be implemented:

1. Break all Direct Master burst reads at 64K boundaries. Separating bursts across the boundary into 2 separate bursts will work just fine.
2. Break all Direct Master burst reads at 4, 16, or 32K boundaries. While this places even more restrictions on the requesting device, it may be easier to implement than the 64K boundary. (The PCI 9080 provides an option bit that prevents prefetching past 4K boundaries).
3. The 64K boundary can be crossed if the local master only reads in single cycles.

7. Simultaneous PCI DMA Descriptor read with local access to internal registers

Erratum Issue: If the PCI 9080 encounters a PCI target retry when attempting to read a DMA descriptor from the PCI bus, the PCI 9080 will retry the read after a few clocks. A local read of the PCI 9080's internal configuration registers coincident with this retried descriptor read will result in the local initiator receiving data from the DMA descriptor register for which the descriptor is being read. A local write to configuration registers synchronous to this descriptor read will receive a READYo# from the PCI 9080 even though the write never gets through. (The write data never actually goes anywhere.)

Solution/Workaround 1:

1. On local reads, if the value returned is the same as the DMA configuration register, reread the internal register.
2. Read or write to the internal registers twice and compare the values.

8. Aborting chained DMA with descriptors on both PCI and local buses

Erratum Issue: The PCI 9080 has 2 independent DMA controllers that support chaining DMA with their descriptors located on either the PCI or local buses. This makes it possible for one DMA channel to have descriptors on the PCI bus while the other channel has descriptors on the local bus. If a DMA channel is aborted while doing Local-to-PCI transfers and the Local bus is running faster than the PCI bus, a subsequent DMA Local-to-PCI transfer will result in incorrect data being transferred across the PCI 9080 (when both channels are active with descriptors on both sides of the bus).

Solution/Workaround: Follow the DMA abort with a DMA PCI-to-Local transfer (on the aborted channel) of zero bytes. This will reset the internal state machines for subsequent local to PCI transfers.

9. Delayed read during 32K PCI clocks timeout

Erratum Issue: As per PCI Specification revisions 2.1 and 2.2, when a master performs a delayed read, it must complete that delayed read. When a delayed read is not completed within 32K PCI clocks, the target with the pending delayed read should discard it (protect against violations or PCI 2.0 devices). The PCI 9080 discard timer is enabled when MARBR[24] is set to 1 (see PCI 9080 Design Notes #5).

There is a 1-clock window (which is the last clock before the 32K timeout) in which the PCI 9080 is discarding the delayed read. During this 1-clock window, new PCI cycles are not monitored. If a new PCI cycle is begun during that 1 clock window, the PCI 9080 ignores the PCI cycle. Therefore, a Master Abort will occur on the PCI bus.

Solution/Workaround: This is an extremely unlikely occurrence. If your system experiences this issue, its software should monitor for this and recover from it.

10. LOCK# negation during an idle phase

This erratum was previously published. However, it has since been determined to not be an issue in the PCI 9080-3, and therefore this erratum is retracted.

11. DMA Channel 0 Almost Full Threshold value of zero with Local-to-PCI transfers

Erratum Issue: During a DMA channel 0 Local-to-PCI transfer, DMA Channel 0 uses the Almost Full Threshold register bits (DMATHR[11:8]) to determine when the PCI 9080 should request the PCI bus to send out the data in its FIFO. If the threshold is set to 0 and if only 1 word (or less) of data is in the FIFO, it will be left there and not be sent out to the PCI bus. This can only happen when Demand mode is used to transfer only 1 word to the PCI side at a time. If this 1 word is the last word of a DMA transfer, the thresholds will be ignored and the data will still go through. This problem does not apply to DMA Channel 1.

Solutions/Workarounds: (use any)

1. Do not use DMA channel 0 Demand mode to transfer a single word at a time. Use standard block mode or chaining DMA with a byte count of four to effect the transfers instead.
2. Set the Almost Full Threshold to a non-zero value.

12. Direct Slave Read Retry

Erratum Issue: Direct Slave Writes are posted into the FIFO followed by a single PCI 2.1 deferred Direct Slave Read and more Direct Slave Writes. Any posted Direct Slave Writes after a pending Direct Slave Read will cause the read data never to be prefetched into the read FIFO. This condition will cause any subsequent Direct Slave Reads to be retried indefinitely, therefore hanging the bus. All subsequent Direct Slave Writes are unaffected and execute as expected. The PCI r2.1 Features Enable bit is enabled and the PCI Read with Write Flush Mode bit is disabled (MARBR[26, 24] = 01).

Solutions/Workarounds: (use any)

1. Enable the Retry Writes when reads are pending bit (MARBR[25] = 1).

   Note: MARBR[24] must be set to 1 to enable either/both MARBR[26, 25] options (refer to PCI 9080 Design Notes #5).

2. Do not post writes after a pending read when PCI 9080 does not own the Local Bus.
3. Disable the PCI r2.1 Features Enable bit (MARBR[24] = 0). In such case also disable Read Ahead mode (refer to Errata #5).

13. Direct Slave Prefetch Burst Read over the Space Boundary

Erratum Issue: If a Direct Slave prefetch Read transaction is performed over the defined Space boundary, the Local bus may hang by posting a wrong address during the prefetch read cycle. This is due to the fact that the PCI 9080 performs an automatic increment of the address during prefetch Read.

Solution/Workaround:

1. If crossing a Space boundary is required during a Direct Slave Read, the Space prefetch counter should be set to 4 Lwords and the starting address should be quad-word aligned.
2. Disable Bterm mode for the address space in LBRDx[7].

Both solutions will guarantee a new address cycle and proper address value(s) to appear on the Local bus when the defined Space boundary is crossed.

14. PCI Target Abort during DMA Transfer

This issue was previously published as an erratum. However, it has since been determined to be compliant with the PCI Specification, and therefore this erratum is retracted. The text for this issue has been re-published as PCI 9080 Design Notes #10.

15. Local Bus input pin minimum Hold times and output pin Maximum Output Delay specifications (text revised from Errata revision 1.3, values are unchanged)

Erratum Issue: The PCI 9080 Local Bus input pins are synchronous inputs for which input Setup and Hold times with respect to the rising edge of the local clock must be met. PCI 9080 Local Bus output pin signal timing is also synchronous with respect to the rising edge of the local clock, and output signals are guaranteed to be valid no later than the output delay time specified for each pin.

If input Setup/Hold times are not met, proper PCI 9080 operation cannot be assured. For example, the PCI 9080 might start a Local Bus cycle without asserting ADS# if the LHOLDA input hold timing requirement is not met.

Similarly, if a Local Bus device samples PCI 9080 output pins after the rising local clock edge but before the Output Delay time for those pins has passed, the external device might not see the correct logic levels for those PCI 9080 output and consequently the design might not perform as intended.

PCI 9080 Local Bus input pin minimum Setup and Hold times, and output pin Maximum Output Delay times, are listed below. Some of the input pin Hold time values differ from those listed in Table 6-5 of the PCI 9080 Data Book revision 1.06 (and prior revisions). Maximum Output Delay times for both ADS# and the Address Bus outputs, listed below, differ from those listed in Table 6-6 of the PCI 9080 Data Book revision 1.06 (and prior revisions).

Note that while PCI 9080 Local Bus input Hold times are increased beyond the times listed in the PCI 9080 Data Book, the effect upon a design may be inconsequential particularly if external device signaling is designed to switch on the falling edge of the Local Bus clock.

Solutions/Workarounds:

Below is the AC Electrical Characteristics for Local Inputs, with the corrected times. Timings that differ from the PCI 9080 Data Book version 1.06, Table 6-5 are printed in boldface.

Table 6-5. AC Electrical Characteristics (Local Inputs) Estimated over Operating Range

Signals (Synchronous Inputs) C_L = 50pF, V_{DD} = 5.0V ± 5%	Tsetup(ns)	Old Thold (ns)	**New Thold (ns)**
ADS#	6	1	**2.01**
BIGEND#	4	0	0
BLAST#	6	0	0
BREQi	7	0	0
BTERM#	7	1	**1.5**
DP[3:0]	4	0	0
DREQ[1:0]#	3	1	**1.5**
EOT0#	7	1	**1.5**
EOT1#	1	1	**1.5**
LA[31:0]	5	0	0
LAD[31:0]	5	0	**1.69**
LBE[3:0]#	7	0	**1.25**
LD[31:0]	5	0	0
LHOLDA	7	2	**2.5**
LINTi	7	0	0
LLOCK	4	0	0
LW/R#	9	0	0
READYi#	8	1	**1.5**
S[2:0]	1	2	**2.5**
USERi	4	0	0
WAITi#	13	0	0

Below is the AC Electrical Characteristics for Local Outputs, with the corrected times. Timings that differ from the PCI 9080 Data Book version 1.06, Table 6-6 are printed in boldface.

Table 6-6. AC Electrical Characteristics (Local Outputs) Estimated over Operating Range

Signals (Synchronous Outputs) V_{DD} **= 5.0V ± 5%**	Old Output Tvalid (max)	**New Output Tvalid (max)**
ADS#	14.5	**15.0**
BLAST#	16	16
BREQo	13	13
BTERMo#	15	15
DACK[1:0]#	14	14
DEN#	13	13
DMPAF#	17	17
DP[3:0]	20	20
DT/R#	14	14
LA[31:2] (C Mode)	15.8	**16.3**
LABS[3:1]	12	12
LAD[31:0] (Address, J Mode)	12.1	**12.6**
LAD[31:0] (Data, J Mode)	15	15
LD[31:0] (32- and 16-Bit, C Mode)	15	15
LD[31:0] (8-bit C Mode)	20	20
LBE[3:0]#	16	16
LDSHOLD	12	12
LHOLD	13	13
LINTo#	13	13
LLOCKo#	12	12
LSERR#	12	12
LW/R#	14	14
PCHK#	12	12
READYo#	14	14
USERo	11	11
WAITo#	18	18

16. Messaging Unit data corruption if Queue Prefetch (Inbound Free List FIFO Prefetch and/or Outbound Post List FIFO Prefetch) is enabled

Erratum Issue: When the Messaging Unit is enabled (QSR[0] = 1), the Inbound Free List FIFO holds the Message Frame Addresses (MFA) of available Message Frames (available to an external PCI agent) in shared Local memory. The Outbound Post List FIFO holds the MFA of all currently posted messages (destined to an external PCI agent) that are in shared Local memory.

To reduce read latency, queue prefetching can be enabled in QSR[2] and QSR[3]. However, if queue prefetching is enabled, the Messaging Unit data can return incorrect data due to internal updating of the pointers.

Solution/Workaround: Disable queue prefetching for the Messaging Unit by clearing the QSR[2] and QSR[3] register bits (default value).

17. Simultaneous Write to Queue Register in Messaging Unit

Erratum Issue: The PCI 9080 updates one of four queue pointers automatically each time there is a Read or Write to the messaging unit Inbound (Port 40h) or Outbound (Port 44h) ports. The four registers are the inbound free tail pointer (IFTPR), the inbound post head pointer (IPHPR), the outbound free head pointer (OFHPR), and outbound post tail pointer (OPTPR). If a local master initiates a write to the PCI 9054 messaging unit queue registers simultaneous to a PCI access to the Inbound or Outbound ports, the PCI 9080 will fail to automatically increment the appropriate pointers to reflect this. This can result in overwriting a previous message or retrieving a previously read message from the queue. These pointers cannot be 'corrected' due to the fact that the PCI 9080 cannot determine if an increment failure has occurred. The failure only occurs when the PCI 9080 returns READY# to the local master simultaneous with the PCI 9080 returning TRDY# to the PCI master on an Inbound or Outbound port access.

This erratum does not affect the I_2O protocol. It only affects custom messaging unit implementations.

Solutions/Workarounds: (use any)

1. Disable the PCI r2.1 Features Enable bit by clearing MARBR[24]. With this bit clear, the PCI Target Retry Delay Clocks register bits (LBRD0[31:28]) should be set to a value of 3h or greater.

2. **For Multiple Initiator Implementations:**

   a. Implement a semaphore in an on-chip mailbox or any shared memory region, so that the local master can access the messaging unit queue registers only when the PCI master is not accessing them.

   b. Update the interfering queue registers only from the PCI side, via messages passed through the PCI 9080 internal mailbox registers.

   For Single Initiator Implementations:
   Implement a single request/reply message protocol for custom message passing. This is recommended for applications where there is a single host and the host waits for a reply before initiating a new request.

18. Demand Mode DMA resumes without DREQ# input asserted if the previous DREQ# negation occurs after PCI 9080 asserts BLAST# in response to a BREQ (Bus Request) input assertion

Erratum Issue: If the DREQx# input pin is negated after a BLAST# output assertion caused by a BREQ input assertion while executing a Demand Mode DMA transfer, the PCI 9080 will incorrectly arbitrate for the Local bus and attempt to continue the transfer by asserting ADS# and DACK# without the DREQ# pin being asserted.

Solutions/Workarounds: (use either)

1. Ensure that the DREQx# pin is negated for a minimum of three Local Clocks after the assertion of BLAST#. BLAST# output is asserted upon the second rising edge of the Local Clock after BREQ input is asserted.

2. Suspend Demand Mode transfers by controlling the DREQ0# and DREQ1# inputs, rather than use the BREQ input for such functionality. (Refer to PCI 9080 Errata #3).

19. Demand Mode DMA Channel 0/1 rotation does not occur if Channel 0 owns the Bus when BREQ (Bus Request) input is used to suspend the DMA transfer

Erratum Issue: When DMA Channel 0/1 Priority is configured as Rotational (MARBR[20:19] = 00), if BREQ input is used to suspend a Demand Mode Channel 0 transfer and both DREQ[1:0]# inputs are asserted after the BLAST# output assertion that was caused by the BREQ input assertion, the PCI 9080 will resume Channel 0's DMA transfer instead of rotating to Channel 1.

Solution/Workaround: Use the DREQx# input pins to control which DMA channel is allowed to execute, and which channel will be suspended during Demand Mode transfers. To do this only assert one DREQx# pin at a time.

20. Internal pull-up/pull down resistors

Erratum Issue: PCI 9080 Data Book revisions 1.06 and earlier, in the Pin Summary section (5.1), incorrectly state that Local Bus internal pull-up resistors are 2K ohms and that Local Bus internal pull-down resistors are 100K ohms. All internal pull-up/pull-down resistors are valued at 50K ohms nominal.

Solution/Workaround: Internal pull-up/pull-down resistors are not strong enough to guarantee proper operation of the PCI 9080. Local bus input and I/O signals that are used and that have internal resistors should be connected to external pull-up or pull-down resistors to hold the signal in the inactive state. The EEDO pin requires an external pull-up resistor for reliable EEPROM access.

21. DMA Burst-4 Mode with Constant Address

Erratum Issue: When the PCI 9080 performs a DMA cycle in Burst-4 mode (Burst enabled and BTERM# disabled, DMAMODEx[8:7] = 10b) with a Constant Address (DMAMODEx[11] = 1), if the Local starting address is xCh, the PCI 9080 performs single cycles (ADS# & BLAST# are asserted for each data) rather than a burst. However if the starting address is x0h (or x4h, x8h) the PCI 9080 performs a continuous burst cycle, even though BTERM# is disabled.

Solution/Workaround:

If Constant Address DMA is required use either continuous burst mode (both Burst and Bterm mode bits enabled, DMAMODEx[8:7] = 11b) or non-burst (DMAMODEx[8] = 0) modes, rather than Burst-4 mode.

22. J-mode DMA with Constant Address

Erratum Issue: This erratum applies to PCI 9080 J-mode and not to C-mode.

For DMA the Local Addressing Mode register bit can be set to enable Constant Address on the multiplexed LAD[28:0] address buses (DMAMODEx[11] = 1), or this bit can be cleared (default) to cause the local address to increment during DMA. In J-mode, during the address phase, LAD[1:0]# are valid address bits and these bits toggle to match the LBE[1:0]# state, regardless of whether Constant Address is enabled. Thus with an 8- or 16-bit Local bus the LAD[1:0] address is not held constant.

Solution/Workaround:

When using DMA Constant Address mode, use 32-bit bus width rather than 8- or 16-bit, to keep the LAD[1:0] address bits constant during J-mode DMA transfers.