PCI 9656

Design Notes Rev. 1.2
January 2004

Design Notes Documentation

A. Affected Silicon Revision

This document details Design Notes for the following silicon:

Product	Part Number	Description	Status
PCI 9656BA	PCI9656-BA66BI	64-bit, 66MHz PCI, 32-bit, 66MHz Local, PBGA Package	Production Released

B. Silicon Documentation

The following documentation is the baseline functional description of the silicon.

Document	Revision	Description	Publication Date
PCI 9656BA Data Book	1.1	Silicon Data Book	October 2003

C. Design Note Documentation Revision History

Revision	Description
1.0	Baseline
1.1	Added Design Notes #3 and 4.
1.2	Added Design Note #5.

D. Design Notes Summary

#	Description
1	DMA with Local Constant Address must be Lword aligned
2	VPD Implementation
3	DMA Demand Mode Documentation (replaces Data Book v1.1 Sections 3.4.4.11 and 5.4.4.10)
4	Local-to-PCI Demand Mode DMA DREQ# negation coincident with a Direct Slave Preemption or DREQ# negation on second data transfer leaves one Lword of data in the FIFO (C/J mode)
5	VIO Power Sequencing

E. Design Notes Details

1. DMA with Local Constant Address must be Lword aligned

Design Issue: If the DMA Channel x Local Addressing Mode register bit is set (DMAMODEx[11] = 1, DMA with Constant Address is enabled), the DMA channel x Local Address must be Lword aligned (DMALADRx[1:0] = 00b). If a non-Lword aligned address is used (DMALADRx[1:0] ≠ 00b), the PCI 9656 initially provides a byte or word aligned Constant Address, and valid TSIZ[1:0] value (M-mode only), on the LA[1:0] for C/J-mode or LA[31:30] and TSIZ[1:0] for M-mode pins for the DMA transfer, but upon a disconnect then subsequent reconnection the silicon will drive the two lower address and TSIZ pins to 00b. Additionally, for J-Mode the LAD[1:0] pins will provide a byte or word aligned address only with the first ADS# assertion.

Recommendation: For Constant Address DMA do not use the lowest 2 address bits of the Local Address LA/LAD[1:0] (C/J-mode) or LA[30:31] (M-mode) to decode non-Lword aligned byte or word addresses. Instead always set the DMA Channel x Local Address to be Lword aligned (DMALADRx[1:0] = 00b), and use upper address lines when designing 8- or 16-bit device select decoding logic.

2. VPD Implementation

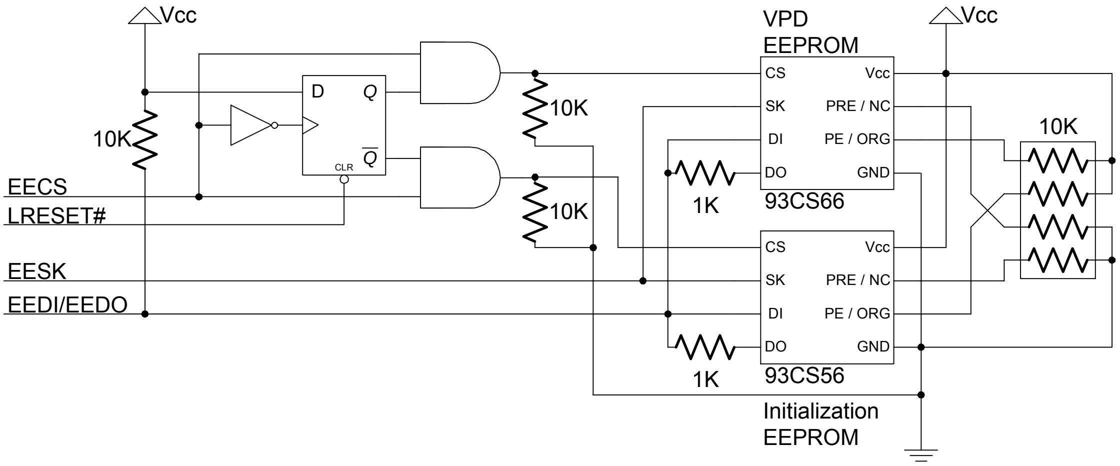
Documented Behavior: The PCI 9656 supports the Vital Product Data (VPD) optional feature detailed in PCI Specification r2.2. The PCI 9656 VPD feature supports storage of a VPD data structure within a serial EEPROM (2k- or 4k-bit, 3-wire interface). The PCI Specification defines the first element within this data structure as the Identifier String at VPD address 0h. Additionally, the PCI Specification requires that the VPD registers (in PCI Configuration Space) be used to read/write VPD data only.

At power-up reset, the PCI 9656 reads 100 bytes of configuration data from a serial EEPROM, beginning with the PCI Device ID value stored at address 0h. Although this address seemingly conflicts with VPD requirements, PLX choose to store configuration data beginning at address 0, in order to maintain compatibility with legacy devices (such as PCI 9054). Additionally, the PCI 9656 provides CNTRL[31,27:24] register bits to allow programming of configuration data within the serial EEPROM, without using the VPD registers.

Solution/Workaround: VPD data storage can be implemented with a serial EEPROM used exclusively for VPD (and not containing PCI 9656 configuration data). Because the PCI 9656 will attempt to read configuration data through the serial EEPROM interface at power-up reset, the VPD EEPROM should not be enabled until PCI 9656 initialization completes (when EECS de-asserts).

The following sample circuit includes two serial EEPROMs, one for initialization and the other for VPD, with only one EEPROM enabled at any time. The circuit initially enables the Initialization EEPROM during power-up reset. After the first EECS de-assertion, the circuit enables access to the VPD EEPROM and disables access to the Initialization EEPROM. The D flip-flop and inverter can both be replaced by a negative edge-triggered J-K flip-flop such as the 74HCT107, with its J input tied high and its K input tied low.

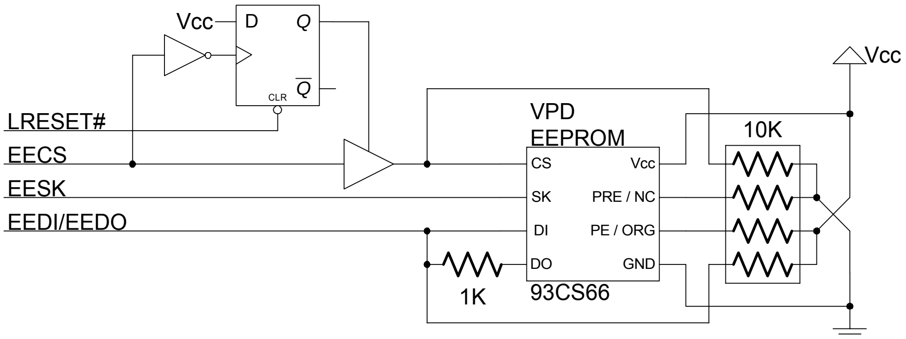
Sample VPD implementation circuit



To allow programming of the Initialization EEPROM, this circuit can be modified such that the flip-flop CLR input is driven by the output of an AND gate instead of from LRESET# directly, with LRESET# connected to one of the AND gate inputs. If any AND gate input is low, the gate output is low, which will enable the Initialization EEPROM and disable the VPD EEPROM.

The following sample circuit includes one serial EEPROM for VPD-only access, and applies to designs having a Local Bus processor that will initialize the PCI 9656, including setting of the Init Done bit (LMISC1[2] = 1).

Sample VPD implementation circuit without initialization EEPROM, and Local CPU sets PCI 9656 Init Done bit



If neither a serial initialization EEPROM nor Local CPU is present to set the Init Done bit, the EEDI/EEDO line must be pulled low with a 1K resistor (which will cause the PCI 9656 itself to set the Init Done bit). To implement VPD in such case, the EEDI/EEDO line in the above circuit would have to be switched to connect only to a pull-down (removing connection to the EEPROM) during the time that the *Q* output of the D flip-flop is low.

The PCI 9656 will reload its configuration registers from serial EEPROM if the Reload Configuration Registers register bit (CNTRL[29]) is transitioned from 0 to 1, or when its Power State (PMCSR[1:0]) is transitioned from D3hot to D0 state. Transitioning the Power State from D3hot to D0 state causes LRESET# assertion, and therefore the circuit will correctly access the Initialization EEPROM in such case. Prior to any setting of the Reload Configuration Registers bit (CNTRL[29]), the designer should ensure that Initialization EEPROM access is enabled, by first performing a Software Reset (CNTRL[30]) to cause LRESET# assertion, since PCI 9656 registers must be reloaded with configuration data and not VPD data.

3. DMA Demand Mode Documentation

Note: This documentation replaces PCI 9656BA Data Book v1.1 Section 3.4.4.11.

3.4.4.11 DMA Demand Mode, Channel *x*

DMA Demand mode allows the transfer of DMA data to be controlled by the DREQ*x*# input pin(s). When the DREQ*x*# signal/pin is asserted, the PCI 9656 starts a DMA transfer based on the values programmed into the DMA registers or by using internally stored values when resuming the transfer after it was paused in response to a DREQ# de-assertion. The PCI 9656 asserts DACK*x*# to indicate that the DMA transfer is in progress on the Local Bus. DACK# asserts with TS# and remains asserted until one clock before the PCI 9656 de-asserts BB# to release the Local Bus. Data is only written to, or read from the Local Bus while DACK# is asserted. DACK*x*# de-assertion indicates that the DMA transfer has completed or is being paused in response to a DREQ# de-assertion or because the PCI 9656 needs to release the Local Bus as described in section 3.4.4.2.

While processing a Demand Mode DMA the amount of data transferred during a Local Bus access depends upon the length and timing of the DREQ*x*# assertion. If DREQ# is not asserted long enough to transfer all the data to complete the DMA transfer the PCI 9656 will transfer data in Lword sized chunks until it releases the Local Bus. For an 8-bit Local Bus device, the PCI 9656 releases the Local Bus after transferring the last byte of an Lword or for a 16-bit Local Bus device the PCI 9656 releases the bus after transferring the last word of an Lword unless it is the last byte or word of the transfer.

Each DMA channel is capable of two types of transfers, PCI-to-Local or Local-to-PCI as selected by the DMADPR*x*[3] register bit. In either case the PCI 9656 becomes a Master on both the PCI and Local Buses to execute the DMA transfer. If the DMA Local-to-PCI FIFO becomes full, DACK# is de-asserted and the DREQ*x*# assertion is ignored. Subsequent transfers do not occur until DACK*x*# is re-asserted indicating additional room is now available in the FIFO.

The DMA Channel *x* Continuous Burst Enable DMAMODE*x*[8], the Fast/Slow Terminate register setting DMAMODE*x*[15], the PCI starting or resuming address, the Local starting or resuming address, and Local Bus widths modify the functionality of the DMA Demand Mode transfer. The functional changes that these items cause are explained in detail in the subsequent sections.

DREQ# may be asserted from one clock to more clocks than is necessary to transfer all the DMA data. This combined with the other factors that modify the PCI 9656 functionality create a wide variety of responses to the assertion of DREQ#. The simplest way to use Demand Mode DMA is to use Slow Terminate Mode (DMAMODE*x*[15] =0) assert the DREQ# pin, wait for DACK# to assert, allow some data to transfer before negating DREQ#, continue to accept or provide data until the Local bus is released by the PCI 9656. If the DREQ*x*# assertion for a DMA transfer will never resume for ongoing transfers, a DMA Abort procedure or the EOT# input signal/pin assertion may be applied to terminate/complete the DMA transfer and/or flush the DMA FIFO.

Note: With DMA Constant Address mode enabled (DMAMODE*x*[11]=1), if burst is enabled (DMAMODE*x*[8]=1), the PCI 9656 enables Continuous Burst mode regardless of the Continuous Burst bit setting (DMADMODE*x*[7]=x).

3.4.4.11.1 Fast/Slow Terminate Mode

In Slow Terminate mode (DMAMODE*x*[15]=0) the PCI 9656 adheres to the M mode protocol and de-asserts the BDIP# output to indicate the current data transfer is the last Lword to be transferred of a burst before releasing the Local Bus while processing the Demand Mode DMA.

In Fast Terminate Mode (DMAMODE*x*[15]=1) the PCI 9656 is **not** required to adhere to the M mode protocol regarding the de-assertion of the BDIP# output to indicate the current data transfer is the last Lword to be transferred of a burst before releasing the Local Bus. Additionally, it forces the PCI 9656 into Continuous Burst mode regardless of the DMAMODE*x*[7] register bit setting.

Slow Terminate mode is the default setting. Running in Slow Terminate Mode is recommended unless the user needs the PCI 9656 to get off the bus immediately and is willing to account for a disconnection without BDIP# assertion on the last Data Transfer. Running in Fast Terminate mode is usually combined with EOT# assertion to terminate/complete the DMA transfer and/or flush the DMA FIFO.

3.4.4.11.2 PCI-to-Local DMA Demand Mode

For a PCI-to-Local Bus DMA Demand Mode transfer, independent of the DREQ*x*# assertion the PCI 9656 will start reading data from the PCI Bus into the DMA FIFO upon the setting of the DMA Channel *x* Start bit (DMACSR*x*[1] =1). The PCI 9656 will continue to read data from the PCI Bus and attempt to keep the FIFO full until the entire DMA transfer is completed. The PCI 9656 waits for at least 2 Lwords of data to be read into the DMA FIFO and DREQ*x*# to be asserted before attempting to arbitrate for and write data to the Local Bus. The PCI 9656 behavior in response to DREQ# assertion depends on when and how long DREQ# is asserted, where it is de-asserted relative to DACK#, data availability, PCI and Local Bus starting or resumption addresses, the Fast/Slow Terminate mode setting, the DMA Channel *x* Continuous Burst Enable DMAMODE*x*[8] setting, and transfer/protocol dependencies.

DREQ*x*# assertion for one or more Local clock periods before at least 2 Lwords of data is available in the FIFO will not be responded to by the PCI 9656. All of the detailed response descriptions below assume 2 Lwords have already been read into the FIFO or that DREQ*x*# will be asserted long enough to read 2 Lwords into the FIFO.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer and flush the FIFO. But before the PCI 9656 releases the Local bus it may or may not want to write additional data. Details of the terminating Local bus behavior when EOT# is asserted are also provided below.

Slow Terminate Mode setting DMAMODE*x*[15]=0 in Burst-4 mode DMAMODE*x*[8,7]=10b

- If the Local Bus starting or resuming address is double Q-word aligned (LA[28:31]=0h), and DREQ*x*# is asserted for one Local Bus clock cycle, or is asserted then de-asserted before the fourth data transfer of a 4 Lword burst the PCI 9656 will execute/complete a 4 Lword burst write and release the Local Bus. If DREQ*x*# remains asserted the PCI 9656 will complete the current burst of 4 Lwords and will continue to execute and complete additional burst writes of 4 Lwords. If DREQ*x*# is de-asserted on the fourth data transfer of any 4 Lword burst the PCI 9656 will complete the current burst and will want to complete one more burst of 4 Lwords.

- If the Local Bus starting or resuming address is not double Q-word aligned (LA[28:31]≠0h) the PCI 9656 will single cycle up to next double Q-word aligned boundary and then start bursting 4 Lwords at a time. If DREQ*x*# is de-asserted during a burst the PCI 9656 will complete the current burst and will want to complete an additional burst of 4 Lwords if DREQ*x*# is de-asserted on the fourth data transfer. If DREQ*x*# is de-asserted during or before the start of a single cycle write the PCI 9656 will complete that write and release the local bus. If DREQ*x*# is de-asserted during the TS# assertion the PCI 9656 will want to complete the current single cycle write and another single cycle write or a burst write of 4.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is writing the fourth data of the 4 Lword burst the PCI 9656 will want to complete the current burst of 4 Lwords and one more burst of 4 Lwords before releasing the Local Bus. Otherwise it will complete the current burst of 4 Lwords and then release the Local bus. If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is executing a single cycle write the PCI 9656 will complete that write and will want to complete another single cycle write or a burst write of 4 before terminating the ongoing DMA Data transfer, flushing the DMA FIFO, and releasing the Local Bus.

Slow Terminate Mode setting DMAMODE*x*[15]=0 in Continuous Burst mode DMAMODE*x*[8,7]=11b

- Regardless of the Local Bus starting or resuming address if DREQ*x*# is asserted for one Local Bus clock cycle, or is asserted then de-asserted before or with TS# assertion the PCI 9656 will execute a 1 Lword single cycle write before it releases the Local Bus.
- If the Local Bus starting or resuming address is double Q-word aligned (LA[28:31]=0h), and DREQ*x*# remains asserted past the assertion of TS# the PCI 9656 will start a burst write. It will continue bursting until DREQ*x*# is de-asserted. Upon DREQ*x*# de-assertion the PCI 9656 will complete the current data cycle and one more data cycle with BDIP# de-asserted before releasing the Local Bus.
- If the Local Bus starting or resuming address is not double Q-word aligned (LA[28:31]≠0h), and DREQ*x*# remains asserted past the assertion of TS# the PCI 9656 will single cycle up to the next double Q-word aligned boundary and then start a burst. Once the 9656 starts a burst it will continue to burst write data until DREQ*x*# is de-asserted. Upon DREQ*x*# de-assertion the PCI 9656 will complete the current data cycle and one more data cycle with BDIP# de-asserted before releasing the Local Bus.
- If DREQ*x*# is de-asserted while the 9656 is executing a single cycle transfer the PCI 9656 will want to complete the current single cycle and may or may not want to complete one more single cycle or a burst of 1.
- If EOT# is asserted along with DREQ*x*# de-assertion while the PCI 9656 is bursting (writing) data the PCI 9656 will want to complete the current data cycle and one more data cycle with BDIP# de-asserted before releasing the Local Bus. If the 9656 is executing a single cycle transfer when EOT# is asserted along with DREQ*x*# de-assertion it will want to complete the current single cycle and may or may not want to complete one more single cycle or a burst of 1. If the 9656 is asserting TS# for a burst transfer when EOT# is asserted along with DREQ*x*# de-assertion it will want to complete a burst of 2 Lwords before terminating the ongoing DMA Data transfer, flushing the DMA FIFO, and releasing the Local Bus.

Fast Terminate Mode setting DMAMODE*x*[15]=1, Continuous Burst mode DMAMODE*x*[8,7]=1xb

In Fast Terminate Mode DMAMODE*x*[15]=1 the PCI 9656 defaults into Continuous Burst mode regardless of the setting of the DMAMODE*x*[7] bit. With these settings the PCI 9656 operates the same as the Slow Terminate/Continuous burst description above with the following differences.

- BDIP# is never asserted.
- If DREQ*x*# is de-asserted during a single cycle transfer the PCI 9656 will release the Local Bus after completion of the current write transfer. For a burst the 9656 will write one more Lword with DREQ*x*# de-asserted and TA# asserted (or the internal Wait State Counter(s) has decremented to 0) or wait for TA# assert to complete the write cycle before releasing the Local bus.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is starting a transfer to write data or is waiting (TA# de-asserted) to write data during a burst the PCI 9656 will want to write one more Lword of data before it releases the Local Bus. The PCI 9656 will complete the final data transfer if TA# is asserted with EOT# or when TA# becomes asserted after EOT# assertion. If EOT# is asserted while waiting to complete a single cycle transfer the PCI 9656 will complete that transfer before releasing the Local Bus.

3.4.4.11.3 Local-to-PCI DMA Demand Mode

For a Local-to-PCI Bus DMA Demand Mode transfer, the PCI 9656 does not arbitrate on the Local Bus to read data into the DMA FIFO until the DMA Channel *x* Start bit (DMACSR*x*[1] =1) and the DREQ*x*# signal/pin is asserted. As long as DREQ*x*# is asserted the PCI 9656 will continue to read data from the Local Bus and attempt to keep the FIFO full until the entire DMA transfer is completed. The PCI 9656 waits for at least 1 Lword of data to be read into the DMA FIFO before attempting to arbitrate for and write data to the PCI Bus. The PCI 9656 behavior in response to DREQ# assertion depends on when and how long DREQ# is asserted, the PCI and Local Bus starting or resumption addresses, where it is de-asserted relative to DACK#, the Fast/Slow Terminate mode setting, the DMA Channel *x* Continuous Burst Enable DMAMODE*x*[8] setting, data availability, the Fast/Slow Terminate mode setting, and transfer/protocol dependencies.

For transfers that are reading data from a 32-bit Local Bus device the PCI 9656 will require 1 or 2 Lwords to be read (see below) and put into the DMA FIFO before it arbitrates for and writes data to the PCI bus. As a result, if 2 Lwords must be read, one Lword of data will remain in the DMA FIFO until the second Lword is read. Upon reading the required amount of data the PCI 9656 will then arbitrate for and write the data to the PCI bus.

For Non-Lword aligned transfers that read data from 8- or 16-bit Local Bus devices the PCI 9656 requires a minimum amount of bytes or words to be read from the Local Bus device before it arbitrates for and writes data to the PCI bus. The number of bytes or words that it needs to read before actually transferring the data to the PCI bus is determined by the PCI starting address. As a result, there could be seven or fewer bytes remaining in the DMA FIFO at any given time.

Therefore, once a DMA Demand Mode transfer is started (DREQ*x*# asserted) it may be necessary for the PCI 9656 to read more bytes, words or Lwords from the Local Bus than are required to complete the DMA transfer. If bytes, words, or an Lword remain in the DMA FIFO due to a DREQ*x*# de-assertion, when DREQ*x*# is re-asserted (and additional data is read) that data is transferred to the PCI Bus. If the DREQ*x*# assertion never resumes for ongoing transfers, a DMA Abort procedure or EOT# assertion can be applied to terminate/complete the ongoing DMA transfer.

Because there are 2 Lwords per DMA FIFO location, upon a DREQ*x*# de-assertion the PCI 9656 may or may not want to continue to read or write more data. The assertion of DREQ*x*# until DACK*x*# is asserted and at least 4 Lwords of data are read into the DMA FIFO is recommended.

PCI 9656 response descriptions are provided below to allow smaller transfers inside an ongoing DMA transfer.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer. But before the PCI 9656 releases the Local bus it may or may not want to read additional data. Details of the terminating Local bus behavior when EOT# is asserted are provided below. All data that is read into the DMA FIFO will be written to the PCI Bus.

Slow Terminate Mode (DMAMODE*x*[15]=0) in Burst-4 mode (DMAMODE*x*[8,7]=10b)

- If the Local Bus starting or resuming address is double Q-word aligned (LA[28:31]=0h), and DREQ*x*# is asserted for one Local Bus clock cycle, or is asserted then de-asserted before the fourth data transfer of a 4 Lword burst the PCI 9656 will execute/complete a 4 Lword burst read and then release the Local Bus. If DREQ*x*# remains asserted the PCI 9656 will complete the current burst of 4 Lwords and will continue to execute and complete additional burst reads of 4 Lwords. If DREQ*x*# is de-asserted on or after the third data transfer of any 4 Lword burst the PCI 9656 will complete the current burst and will want to complete one more burst of 4 Lwords.
- If the Local Bus starting address is not double Q-word aligned (LA[28:31]≠0h) the PCI 9656 will single cycle up to next double Q-word aligned boundary and then start bursting 4 Lwords at a time. If DREQ*x*# is de-asserted during a burst the PCI 9656 will complete the current burst and may want to complete an additional burst of 4 Lwords. If DREQ*x*# is de-asserted during or before the start of a single cycle read the PCI 9656 will complete that read and any additional single cycle reads, then a burst of 4.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is reading the fourth data of the 4 Lword burst the PCI 9656 will want to complete the current burst of 4 Lwords and one more burst of 4 Lwords before releasing the Local Bus. Otherwise it will complete the current burst of 4 Lwords and then release the Local bus.

Slow Terminate Mode (DMAMODE*x*[15]=0) in Continuous Burst mode (DMAMODE*x*[8,7]=11b)

- Depending upon the PCI and Local bus starting or resuming addresses, where and how long DREQ*x*# is asserted and where it is de-asserted the PCI 9656 will want to execute zero to three single cycle transfers followed by bursts of any size. The size of the burst is determined by the length of time DREQ*x*# remains asserted. Any assertion of DREQ*x*# will cause the PCI 9656 to read at least one Lword of data. The only time the PCI 9656 will start or resume the transfer with a Burst is if the address is double Q-word aligned (LA[28:31]=0h).
- Once DREQ*x*# is de-asserted the DMA controller(s) will release the Local Bus after completion of one to two additional Lword data transfers. The transfer will be paused on an Lword boundary (LA[30:31]=00b). For a burst the PCI 9656 will read one more Lword with BDIP# asserted and release the local bus after another Lword is read with BDIP# de-asserted and TA# asserted (or the internal Wait State Counter(s) has decremented to 0). If DREQ*x*# is de-asserted while the PCI 9656 is single cycling the PCI 9656 will need to complete the current single cycle read and one more Lword read before releasing the bus.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is reading data or is waiting (TA# de-asserted) to read data during a burst the PCI 9656 will want to read two more Lwords of data before it releases the Local Bus. One during the cycle that EOT# is asserted or when TA# becomes asserted and one more additional Lword with BDIP# de-asserted. If EOT# is asserted while waiting to complete a single cycle transfer the PCI 9656 will complete that transfer and one additional Lword data transfer before releasing the Local Bus.

Fast Terminate Mode (DMAMODE*x*[15]=1) DMAMODE*x*[8,7]=1xb

In Fast Terminate Mode DMAMODE*x*[15]=1 the PCI 9656 defaults into Continuous Burst mode regardless of the setting of the DMAMODE*x*[7] bit. With these settings the PCI 9656 operates the same as the Slow Terminate/Continuous burst description above with the following differences.

- BDIP# is never asserted.
- If DREQ*x*# is de-asserted during a single cycle the PCI 9656 will release the Local Bus after completion of the current read and may or may not want to execute one more single cycle read. For a burst the PCI 9656 will read one more Lword with DREQ*x*# de-asserted and TA# asserted (or the internal Wait State Counter(s) has decremented to 0) or wait for TA# assertion and read the last data before releasing the Local bus.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is starting a transfer to read data or is waiting (TA# de-asserted) to read data during a burst the PCI 9656 will want to read one more Lword of data before it releases the Local Bus. The PCI 9656 will complete the final data transfer if TA# is asserted with EOT# or when TA# becomes asserted after EOT# assertion. If EOT# is asserted while waiting to complete a single cycle transfer the PCI 9656 will complete that transfer before releasing the Local Bus.

Note: This documentation replaces PCI 9656BA Data Book v1.1 Section 5.4.4.10.

5.4.4.10 DMA Demand Mode, Channel *x*

DMA Demand mode allows the transfer of DMA data to be controlled by the DREQ*x*# input pin(s). When the DREQ*x*# signal/pin is asserted, the PCI 9656 starts a DMA transfer based on the values programmed into the DMA registers or by using internally stored values when resuming the transfer after it was paused in response to a DREQ# de-assertion. The PCI 9656 asserts DACK*x*# to indicate that the DMA transfer is in progress on the Local Bus. DACK# asserts with ADS# and remains asserted until one clock before the PCI 9656 de-asserts LHOLD to release the Local Bus. Data is only written to, or read from the Local Bus while DACK# is asserted. DACK*x*# de-assertion indicates that the DMA transfer has completed or is being paused in response to a DREQ# de-assertion or because the PCI 9656 needs to release the Local Bus as described in section 5.4.4.2.

While processing a Demand Mode DMA the amount of data transferred during a Local Bus access depends upon the length and timing of the DREQ*x*# assertion. If DREQ# is not asserted long enough to transfer all the data to complete the DMA transfer the PCI 9656 will transfer data in Lword sized chunks until it releases the Local Bus. For an 8-bit Local Bus device, the PCI 9656 releases the Local Bus after transferring the last byte of an Lword or for a 16-bit Local Bus device the PCI 9656 releases the bus after transferring the last word of an Lword unless it is the last byte or word of the transfer.

Each DMA channel is capable of two types of transfers, PCI-to-Local or Local-to-PCI as selected by the DMADPR*x*[3] register bit. In either case the PCI 9656 becomes a Master on both the PCI and Local Buses to execute the DMA transfer. If the DMA Local-to-PCI FIFO becomes full, DACK# is de-asserted and the DREQ*x*# assertion is ignored. Subsequent transfers do not occur until DACK*x*# is re-asserted indicating additional room is now available in the FIFO.

The Fast/Slow Terminate register setting DMAMODE*x*[15], the PCI starting or resuming address, the Local starting or resuming address, and Local Bus widths modify the functionality of the DMA Demand Mode transfer. The functional changes that these items cause are explained in detail in the subsequent sections.

DREQ# may be asserted from one clock to more clocks than is necessary to transfer all the DMA data. This combined with the other factors that modify the PCI 9656 functionality create a wide variety of responses to the assertion of DREQ#. The simplest way to use Demand Mode DMA is to use Slow Terminate Mode (DMAMODE*x*[15] =0) assert the DREQ# pin, wait for DACK# to assert, allow some data to transfer before negating DREQ#, continue to accept or provide data until last Lword is transferred with BLAST# asserted. If the DREQ*x*# assertion for a DMA transfer will never resume for ongoing transfers, a DMA Abort procedure or the EOT# input signal/pin assertion may be applied to terminate the DMA transfer and/or flush the DMA FIFO.

Note: With DMA Constant Address mode enabled (DMAMODE*x*[11]=1), if burst is enabled (DMAMODE*x*[8]=1), the PCI 9656 enables Continuous Burst mode regardless of the Continuous Burst bit setting (DMADMODE*x*[7]=x).

5.4.4.10.1 Fast/Slow Terminate Mode

In Slow Terminate mode (DMAMODE*x*[15]=0) the PCI 9656 adheres to the C/J mode protocol and asserts the BLAST# output to indicate the current data transfer is the last Lword to be transferred before releasing the Local Bus while processing the Demand Mode DMA.

In Fast Terminate Mode (DMAMODE*x*[15]=1) the PCI 9656 is **not** required to adhere to the C/J mode protocol regarding the assertion of the BLAST# output to indicate the current data transfer is the last Lword to be transferred before releasing the Local Bus. However the PCI 9656 will still assert BLAST# in Fast Terminate Mode if the DREQ*x*# input is de-asserted and the external READY#=0 (asserted), on the transfer before the Local Address is to be LA[2:0]=000b (C mode) or LAD[2:0]=000b (J mode), or if the internal Wait State Counter(s) decrements to 0 for the current Lword.

Slow Terminate mode is the default setting. Running in Slow Terminate Mode is recommended unless the user needs the PCI 9656 to get off the bus immediately and is willing to account for a disconnection with or without BLAST# assertion on the last Data Transfer. Running in Fast Terminate mode is usually combined with EOT# assertion to terminate/complete the DMA transfer and/or flush the DMA FIFO.

5.4.4.10.2 PCI-to-Local DMA Demand Mode

For a PCI-to-Local Bus DMA Demand Mode transfer, independent of the DREQ*x*# assertion the PCI 9656 will start reading data from the PCI Bus into the DMA FIFO upon the setting of the DMA Channel *x* Start bit (DMACSR*x*[1] =1). The PCI 9656 will continue to read data from the PCI Bus and attempt to keep the FIFO full until the entire DMA transfer is completed. The PCI 9656 waits for at least 2 Lwords of data to be read into the DMA FIFO and DREQ*x*# to be asserted before attempting to arbitrate for and write data to the Local Bus. The PCI 9656 behavior in response to DREQ# assertion depends on when and how long DREQ# is asserted, where it is de-asserted relative to DACK#, data availability, the PCI and Local Bus starting or resumption addresses, the Fast/Slow Terminate mode setting, and transfer/protocol dependencies.

DREQ*x*# assertion for one or more Local clock periods before at least 2 Lwords of data is available in the FIFO will not be responded to by the PCI 9656. All of the detailed response descriptions below assume 2 Lwords have already been read into the FIFO or that DREQ# will be asserted long enough to read 2 Lwords into the FIFO.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer and flush the FIFO. But before the PCI 9656 releases the Local bus it may or may not want to write additional data. Details of the terminating Local bus behavior when EOT# is asserted are also detailed below.

Slow Terminate Mode setting DMAMODE*x*[15]=0

- DREQ*x*# assertion for one Local clock period when at least 2 Lwords of data are available in the FIFO, or assertion up to but not including the first clock cycle with DACK# asserted results in a one Lword transfer on the Local Bus with a BLAST# assertion.

- DREQ*x*# de-assertion on the clock period of DACK# assertion or any clock period after DACK*x*# assertion results in one to two more Lword transfers on the Local Bus. BLAST# is asserted with the last Lword.
- DREQ*x*# assertion (when 2 or more Lwords are available in the FIFO), then de-assertion followed by a re-assertion up to and including the first cycle of DACK# assertion will cause the PCI 9656 to keep LHOLD asserted and continue to transfer additional Lwords as long as DREQ# stays asserted and data is available. Otherwise the PCI 9656 will assert BLAST#, transfer one Lword release and re-arbitrate for the Local Bus when it detects DREQ# is asserted again. For each transfer BLAST# is asserted with the last Lword.
- If EOT# is asserted and DREQ*x*# is de-asserted on the same clock the PCI 9656 will transfer one additional Lword with BLAST# asserted before terminating the ongoing DMA Data transfer and flushing the DMA FIFO.

Fast Terminate Mode setting DMAMODE*x*[15]=1

- DREQ*x*# assertion for one Local clock period when at least 2 Lwords of data are available in the FIFO, or assertion up to but not including the first clock cycle with DACK# asserted results in a one Lword transfer on the Local Bus with a BLAST# assertion.
- DREQ*x*# de-assertion on the same clock as DACK*x*# assertion or any other clock cycle thereafter in which DACK# is asserted results in one additional Lword transferred to the Local Bus without BLAST# assertion.
- DREQ*x*# assertion (when 2 or more Lwords are available in the FIFO), then de-assertion followed by a re-assertion up to and including the first cycle of DACK# assertion will cause the PCI 9656 to keep LHOLD asserted and continue to transfer additional Lwords as long as DREQ# stays asserted and data is available. Otherwise the PCI 9656 will assert BLAST#, transfer one Lword release and re-arbitrate for the Local Bus when it detects DREQ# is asserted again. Upon the following DREQ# de-assertion one additional Lword is transferred on the Local Bus without a BLAST# assertion.
- If DREQ*x*# is de-asserted during the transfer the PCI 9656 may assert BLAST# if any of the following conditions are met; the data is the last Lword for the entire DMA transfer, the Local Bus Latency Timer expires.
- If EOT# is asserted and DREQ*x*# is de-asserted on the same clock that the PCI 9656 transfers valid data, the PCI 9656 will not assert BLAST# or provide valid data on subsequent clock cycles. The PCI 9656 releases the local bus two clocks after EOT# is asserted, terminates the ongoing DMA Data transfer and flushes the DMA FIFO.

5.4.4.10.3 Local-to-PCI DMA Demand Mode

For a Local-to-PCI Bus DMA Demand Mode transfer, the PCI 9656 does not arbitrate on the Local Bus to read data into the DMA FIFO until the DMA Channel *x* Start bit (DMACSR*x*[1] =1) and the DREQ*x*# signal/pin is asserted. As long as DREQ*x*# is asserted the PCI 9656 will continue to read data from the Local Bus and attempt to keep the FIFO full until the entire DMA transfer is completed. The PCI 9656 waits for at least 1 Lword of data to be read into the DMA FIFO before attempting to arbitrate for and write data to the PCI Bus. The PCI 9656 behavior in response to DREQ# assertion depends on when and how long DREQ# is asserted, where it is de-asserted relative to DACK#, data availability, the Fast/Slow Terminate mode setting, the PCI and Local Bus starting or resumption addresses, and transfer/protocol dependencies.

For transfers that are reading data from a 32-bit Local Bus device the PCI 9656 will require 1 or 2 Lwords to be read (see below) and put into the DMA FIFO before it arbitrates for and writes data to the PCI bus. As a result, if 2 Lwords must be read, one Lword of data will remain in the DMA FIFO until the second Lword is read. Upon reading the required amount of data the PCI 9656 will then arbitrate for and write the data to the PCI bus.

For Non-Lword aligned transfers that read data from 8- or 16-bit Local Bus devices the PCI 9656 requires a minimum amount of bytes or words to be read from the Local Bus device before it arbitrates for and writes data to the PCI bus. The number of bytes or words that it needs to read before actually transferring the data to the PCI bus is determined by the PCI starting address. As a result, there could be seven or fewer bytes remaining in the DMA FIFO at any given time.

Therefore, once a DMA Demand Mode transfer is started (DREQ*x*# asserted) it may be necessary for the PCI 9656 to read more bytes, words or Lwords from the Local Bus than are required to complete the DMA transfer. If bytes, words, or an Lword remain in the DMA FIFO due to a DREQ*x*# de-assertion, when DREQ*x*# is re-asserted (and additional data is read) that data is transferred to the PCI Bus. If the DREQ*x*# assertion never resumes for ongoing transfers, a DMA Abort procedure or EOT# assertion can be applied to terminate/complete the ongoing DMA transfer.

The assertion of DREQ*x*# until DACK*x*# is asserted and at least 4 Lwords of data are read into the DMA FIFO is recommended. However the following information is provided to allow smaller transfers inside an ongoing DMA transfer.

Because there are 2 Lwords per DMA FIFO location, the PCI 9656 puts data into the DMA FIFO in 2-Lword quantities. Therefore, when DREQ*x*# is de-asserted the PCI 9656 will continue to read data up to the next Quad word aligned boundary (X0h or X8h address boundary (Local Address LA[2:0]=000b (C mode) or LAD[2:0]=000b (J mode)).

Detailed descriptions of the PCI 9656's response to DREQ*x*# assertion when the DMA transfer first starts or resumes reading data from Local Bus are provided below.

EOT# assertion causes the PCI 9656 to terminate the ongoing DMA transfer. But before the PCI 9656 releases the Local bus it may or may not want to read additional data. Details of the terminating Local bus behavior when EOT# is asserted are detailed below. All data that is read into the DMA FIFO will be written to the PCI Bus.

Slow Terminate Mode setting DMAMODE*x*[15]=0

- If the PCI destination address is X0h/X8h, and DREQx# is asserted for one Local Bus clock cycle, or is asserted then de-asserted on any cycle before READY# is asserted the PCI 9656 will require 2 Lwords to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and remains asserted until it is de-asserted on the first or second cycle that data is transferred (READY# asserted) the PCI 9656 will require 1 additional Lword to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 3, 5, 7… (Odd) Lword with READY# asserted the PCI 9656 will require 1 additional Lword to be read into the DMA FIFO.
- If the PCI destination address is X0h/X8h, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 4, 6, 8… (Even) Lword with READY# asserted the PCI 9656 will require 2 additional Lwords to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted for one Local Bus clock cycle, or is asserted then de-asserted before or on the same cycle that the PCI 9656 asserts ADS# the PCI 9656 will require 1 Lword to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and remains asserted on the cycle that the PCI 9656 asserts ADS# and then de-asserts before any data is transferred (READY# not asserted) the PCI 9656 will require 2 Lwords to be read into the DMA FIFO.
- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 1st or an even 2, 4, 6, 8… Lword with READY# asserted the PCI 9656 will require 1 additional Lword to be read into the DMA FIFO.

- If the PCI destination address is X4h/XCh, and DREQ*x*# is asserted and held asserted until it is de-asserted on a cycle that is transferring the 3, 5, 7… (Odd) Lword with READY# asserted the PCI 9656 will require 2 additional Lwords to be read into the DMA FIFO.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is reading data (READY# asserted) the PCI 9656 will assert BLAST# and will want to read one more Lword of data before releasing the Local Bus. Otherwise upon the re-assertion of READY# the PCI 9656 will want to read one Lword before asserting BLAST# and reading a second Lword.

Fast Terminate Mode setting DMAMODE*x*[15]=1

- If DREQ*x*# is asserted for one Local clock period or is asserted and remains asserted until it is de-asserted before or during the actual transfer of data the PCI 9656 will not assert BLAST# but will require 1 to 2 additional Lword(s) to be read into the DMA FIFO.
- If DREQ*x*# is de-asserted during the transfer the PCI 9656 may assert BLAST# if any of the following conditions are met; the starting PCI address is X4h/XCh, the data is the last data for the entire DMA transfer, or the Local Bus Latency Timer expires.
- If EOT# is asserted along with DREQ*x*# de-assertion as the PCI 9656 is reading data (READY# asserted) the PCI 9656 will read one more Lword of data (during the cycle that EOT# is asserted) and then immediately release the Local Bus. Otherwise the PCI 9656 will wait for the re-assertion of READY# to read one Lword of data before releasing the Local Bus.

4. Local-to-PCI Demand Mode DMA DREQ# negation coincident with a Direct Slave Preemption or DREQ# negation on second data transfer leaves one Lword of data in the FIFO (C/J mode)

Design Issue #1: Depending upon the PCI destination address, when DREQ# is asserted (to start or resume a Demand Mode DMA Local-to-PCI transfer), and where DREQ# is negated (to pause the transfer) the silicon will predictably require 1 to 2 additional Lwords to be read into the DMA FIFO. Additionally, once the data is read into the FIFO the silicon will write all of the data to the PCI bus. However, if a Direct Slave transfer preempts the DMA coincident with the de-assertion of DREQ# the silicon may not read the additional Lword it needs to assemble and write a 64-bit quantity to the PCI bus. In this case, one Lword of data will remain in the DMA FIFO until the transfer is resumed with a re-assertion of DREQ#.

Recommendation: To ensure the silicon will not leave any data in the DMA FIFO when the transfer is paused detect this condition by checking the expected Lword count (even or odd depending upon the PCI starting or resuming address) when BLAST# is asserted by the silicon in response to a DREQ# negation. If the condition is detected and one additional Lword is needed, assert DREQ# for one clock cycle on the second clock cycle after the BLAST# assertion. Once the Direct Slave transaction(s) are completed the silicon will read the additional Lword and transfer a 64-bit quantity of data to the PCI bus.

Design Issue #2: If DREQ# is asserted (to start or resume a Demand Mode DMA Local-to-PCI transfer) and then DREQ# is subsequently negated (to pause the transfer) on the second data transfer one additional Lword (third data transfer) will be read into the DMA FIFO and will remain their until the transfer is resumed with a re-assertion of DREQ#.

Recommendation: Avoid this issue by not negating DREQ# as described above or use the same solution as described in recommendation for Design Issue #1.

5. VIO Power Sequencing

Design Issue: PCI 9656 Data Book Section 13.1 states, with respect to power sequencing requirements, the following: there are five different power sources-Vring, Card_Vaux, Vcore, 2.5Vaux, and VIO. To properly sequence power to these five sources, the only requirement is that Vcore, 2.5Vaux, and VIO must receive power no later than 10 ms after Vring and Card_Vaux receive power... **Caution:** Violating the above power sequencing requirement **will** damage the PCI 9656 device.

Each PCI pin/pad contains two clamping diodes, one to VIO and the other to ground. If the VIO voltage source is not powered and it presents a low impedance path to ground, the PCI 9656's VIO pins can source high current, which could damage the part immediately or cause undue long-term electrical stress to the part. The amount of current each PCI pin/pad will source is dependent upon the device that is driving the signal/pad, or upon the value of the pull-up resistor when the signal is not driven.

Engineering Change: For designs and add-in cards that have an independent voltage source for VIO for which proper power sequencing cannot be guaranteed, a resistor **MUST** be used between the VIO voltage source and PCI 9656 VIO pins to limit the current and protect the device from damage or long term undue stress. Use the following guidelines to determine the value of this required resistance.

A 40-200 ohm resistance between the VIO voltage source and PCI 9656 VIO pins is recommended if VIO will be a maximum of 3.6 Volts (3.3 Volt signaling environments only). For designs that can operate in either 3.3 or 5.0 Volt signaling environments, 40-70 ohm resistance is recommended. A single resistor can be used if the VIO pins are bused, or multiple parallel resistors can be used between the VIO voltage source and the VIO pins. The power dissipation rating of the resistor(s) depends upon the size of the resistance and the signaling environment. For example, if a single 50 ohm resistor is used in a 5V signaling environment, the worst case power dissipation would be 480mW calculated as (V*V)/R (5.5V (maximum signal amplitude plus 10%) - 0.6V (1 diode drop)) squared divided by 50 ohms = 480mW. If four 200 ohm resistors are used in parallel, each would need to be able to dissipate 120mW.

Any resistance value within the recommended ranges will prevent the part from being damaged while providing enough clamping action to keep the Input Voltage (V_{IN}) below its maximum rating. A resistance value at the lower end of the range is recommended to provide better clamping action and therefore provide more Input Voltage (V_{IN}) margin.