

PCI 9060 Frequently Asked Questions

The PCI 9060 FAQ details the most frequently asked questions about this PCI product. The information is divided into five categories: Direct Slave Access, Direct Master Access, DMA, Configuration Cycles, and Miscellaneous to reflect the major functionality of the device. The information in the FAQ will be continuously updated.

Direct Slave Access

1. Can the PCI 9060 perform long burst?

The PCI 9060 can be programmed to perform single cycle, burst 4 (burst 4 Lwords at a time), and burst forever. When the burst mode is disabled in the Local Bus Region Descriptor Register, the PCI 9060 performs single cycle. When burst mode is enabled and BTERM# input is disabled, it bursts up to 4 Lwords, i.e., bursting can start on any boundary and continue up to an address boundary. After the data at the boundary has been transferred, the PCI 9060 generates a new address cycle (ADS#). When burst mode and BTERM# input are enabled, the PCI 9060 performs long bursts. The PCI 9060 will generate a new address cycle once BTERM# is asserted.

2. How can the wait state be generated on the local side of the PCI 9060?

There are two ways to generate wait states. One way is to program the internal wait states in the "Local Bus Region Descriptor" register [LOC 98h] bit [5:2]. The other way is to enable the READYi# mode (again in the "Local Bus Region Descriptor" register and assert READYi# when the local side is ready to receive or provide data.

3. My Local Bus is exclusively 16 bit wide. The PCI bus is 32 bits wide. What do I need to do to communicate between the two buses?

16-bit devices on the local bus is no problem. The PCI 9060 can set Space 0 or the Expansion ROM to 16 or even 8 bits wide. It will simply break up the 32-bit accesses from the PCI side into multiple 16-bit transactions. A 16-bit processor will be using the PCI 9060's SX mode for multiplexed 16-bit buses. External FIFO on the 16-bit bus present a slight problem as the 9060 always attempts to build 32-bit Lwords on direct slave reads. However, when prefetch is disabled and the PCI 9060 prefetch is programmed for single reads, it will perform only a single 16-bit read regardless of the C/BE's on the PCI side.

4. I have multiple devices on my Local Bus but only one Direct Slave address space. How do I accomplish the Address mapping?

A single address space can encompass several devices simply by

using address decoding to generate different chip selects on the local bus. If needed, the Expansion ROM space and Space 1 (SD only) can also be used. Changing bus sizing on the fly will be required if different bus sizes are used by different devices in the same address space.

5. When does the local bus latency counter start counting?

The local bus latency counter starts after PCI 9060 first asserts ADS# on the local bus

6. During the Direct Slave Accesses, do I need to read the addresses on every access or data phase?

No. It is not necessary to read the addresses on every access or data phase. The integrated address counter in the PCI 9060 increments the address automatically. The PCI 9060 latches data on every rising edge of the local clock (LCLK) and holds it until READYi# is received. However, an external address counter is needed in case of reading/writing to the local SRAM (Memory).

Direct Master Access

1. Is it possible to perform Direct Master and Direct Slave transfers simultaneously?

The PCI 9060 can handle Direct Master, Direct Slave, and DMA cycles all at the same time. Once the originator initiates the transfer, the PCI 9060 will handle all the arbitration. However, a simultaneous Direct Slave and Direct Master cycle can result in a deadlock. In this case, the PCI 9060 will request that one of the originators back off the bus.

DMA

1. The PCI 9060 has been occupying the Local Bus for an extended period of time performing a DMA transfer, but the Local Processor needs the bus. What can be done to kick the PCI 9060 off the Local Bus? Will I have to wait for the DMA to finish before Direct Slave Reads can be performed?

This is exactly the reasoning behind the BREQi pin. This pin, when asserted, will force the PCI 9060 off the local bus within 2 clocks. If a Direct Master or Direct Slave transaction is required at that time, the PCI 9060 will prioritize that in front of the ongoing DMA.

2. I am doing a Direct Slave access to the DMA registers from the PCI side, but my Local Bus hangs. What are some possible causes?

To program the PCI 9060's DMA registers from the PCI side, either Space 0 or the Expansion ROM space must be used to program the

configuration registers from the local side. In essence, the PCI 9060 is both master and slave of the local bus. Some designers feel that since the 9060 is writing to itself, they shouldn't have to generate a chip select to the PCI 9060. However, this is not the case. $S0$ must be asserted on the same cycle that $ADS\#$ is asserted on. This may mean a requirement for a fast address decoder if you are running a 40MHz local bus. Also, the address lines on a DMA writeback mimic the data on the cycle after $ADS\#$ is asserted. This may confuse other logic on the local bus and consequently hang the local bus. However, the PCI 9060SD and PCI 9080 offers direct access to the internal DMA registers from the PCI side.

3. I would like to use the PCI 9060's DMA controller, but I don't know in advance how many bytes I'll be transferring. What are some possible workarounds?

The PCI 9060SD features a EOT (end of transfer) pin. For the other devices which do not support EOT, use the Demand Mode request instead. Once the demand mode request is deasserted, the on-going DMA will cease. At this point, the DMA transfer must be aborted by writing to an internal register.

4. How can I generate a PCI interrupt when a DMA operation is done?

The PCI 9060 asserts $INTA\#$ (PCI interrupt) when it samples $LINTi\#$ (local interrupt input) at the rising edge of the local clock. When "DMA Mode Register" bit [10] is set to a one, the PCI 9060 asserts $LINTo\#$ (local interrupt output) once a DMA operation is finished. Therefore, $LINTo\#$ can be tied to $LINTi\#$ to generate PCI interrupt when a DMA operation is done.

5. Can I generate an interrupt upon completion of each descriptor block transfer in chain DMA?

Yes. Set DMA Descriptor Pointer Register bit [2] to a one

6. How does the PCI 9060 handle priority between Channel 0 and Channel 1?

The PCI 9060 can be programmed to support fixed or rotating priority. In the fixed priority, if DMA channel 0 is programmed to have a higher priority than channel 1 then DMA channel 1 can not start until channel 0 is finished. On the other hand, DMA channel 0 and channel 1 will rotate in rotating priority.

7. Does PCI 9060's DMA controller support unaligned transfers?

Yes. For unaligned local-to-PCI transfer, the PCI 9060 reads a partial Lword from the local bus. It then continues to read Lwords

from the local bus. The Lwords are assembled, aligned to the PCI bus address and loaded into the FIFO. For PCI-to-local transfers, Lwords are read from the PCI bus and loaded into the FIFO. On the local side, the Lwords are assembled from the FIFO, aligned to the local bus address and written to the local bus.

Configuration Cycles

1. **I am trying to configure another PCI device with the PCI 9060, but my system hangs when I try this. What could be causing the system to hang during the configuration accesses?**

Most likely you've hit on an IDSEL line with no device attached. Therefore, no DEVSEL is returned to the 9060. This is defined as a PCI Master Abort. To recover from a Master Abort, a flag must be cleared from the 9060's internal registers. Otherwise, no further Direct Master transactions can take place. The same applies to Target Abort situations.

2. **Is the EEPROM programmable in the circuit?**

Yes. The EEPROM is programmable in the circuit through the EEPROM Control register (PCI 6Ch, LOC ECh). Bit 24 needs to be toggled to generate an EEPROM clock for every bit to be read/written to the EEPROM. A pre-programmed EEPROM (non-blank) is required for the system to boot properly. However in the PCI9080, A blank EEPROM is supported. When a value of "FFFFFFFFh" is detected in the first word location, the PCI9080 loads its default values. The EEPROM can then be programmed by writing to the EEPROM Control register.

3. **The PCI 9060 does not seem to read the new EEPROM I've written, what can cause that?**

There are two possibilities: 1. PCI 9060 supports "93CS46" EEPROM only. It does not support "93C46" EEPROM. 2. PE signal (pin #6 of the EEPROM) must be pulled-up through a 4.7K ohm resistor and PRE signal (pin #7) must be pulled-down through a 100 ohm resistor, respectively.

4. **Do I program locations 00h and 01h to 055h and 0Aah, respectively, when I want to use a new EEPROM containing configuration information only?**

No. When using a new EEPROM containing configuration information only, locations 00h and 01h do not need to be programmed to 055h and 0AAH, respectively. That is only necessary when invoking the Expansion ROM.

5. **How can I find the memory location of the add-in after the**

system boots up?

PCI configuration register 18h contains the base address for Local Address Space 0 and Local configuration register 04h contains the local remap address. Therefore a configuration read access to PCI register 18h and Memory or I/O access to Local Configuration register 04h will indicate the memory location of the add-in card.

6. How long does it take to initialize the PCI 9060?

If an EEPROM is connected to the PCI 9060 then it takes anywhere from 172 (1 MHz EEPROM) clocks to 556 clocks to initialize the PCI 9060. If the EEPROM is not installed then the default value will be used. In this case it takes 12 (1 MHz EPROM) clocks. The PCI 9060 detects the EEPROM present by reading a "0" at the start bit 0. If the EEPROM is detected, the information in the EEPROM will be loaded. If it is not detected, the PCI 9060's default value will be loaded. The PCI 9060 supports "Short Load" by pulling down on SHORT# pin and "Long Load" by pulling SHORT# pin high. Five 32-bit words are stored sequentially in the EEPROM in "Short EEPROM Load" and seventeen 32-bit words are stored in "Long EEPROM Load". The EEPROM clock can be derived from the PCI clock or by using a 1MHz external oscillator.

Miscellaneous

1. Does PLX provide Verilog models?

Yes. PLX provides models for all of our 9060 family of product through a third party vendor. This model can be acquired from Simutech, Inc.

*19545 NW vonNeumann Dr. Suite 135
Beaverton, Oregon 97006
<http://www.simutech.com>*

2. Does the Interrupt Control/Status register get reset when issuing a "software reset"?

No. The "software reset" only resets the local configuration registers and not the PCI configuration registers or Shared Runtime registers.

3. Is it possible to write to the Expansion ROM Space? What if my BIOS boots an enabled Expansion ROM as code?

The Expansion ROM space is simply a memory window for Direct Slave transactions. There is no restriction on reads or writes as long as the PCI 9060 is configured to point to a writable device. Changing the access code will insure that the Expansion ROM contents are not run during boot up as code by BIOS. Another solution is to disable the Expansion ROM space at startup and later to enable it. But by then, BIOS may have already allocated all available PCI spaces. A workaround is to map Space 0 bigger than

necessary during boot up, then afterwards, reducing it's size and enabling the expansion ROM space at that time. The Expansion ROM space can then be mapped to the area that Space 0 just vacated.

4. Is the PCI 9060 PCI 2.1 compliant?

The PCI 9060 is PCI 2.0 compliant and 2.1 compatible. The 9060 can be used to design a 2.1 compliant product if the local side of the design meets the minimum requirements for direct slave reads. (return data within 16 clocks) The PCI 9060ES, 9060SD, 9050, and 9080 implement the Delayed Read function of the 2.1 spec. and guarantees data is returned in 16 clocks regardless of the local bus design. The 9060ES, 9060SD, 9050 and 9080 are PCI 2.1 compliant.

5. How can I acquire the source code for the software files listed on the PLX web site?

Contact your local PLX representative or send the request to apps@plxtech.com.

6. How much logic do I need to make my add-on state machine?

PCI 9060 provides a glueless interface to the i960. It also supports other processors: Motorola 68040, PowerPC 403, and 603 are just some, with some glue logic. Please refer to the PCI Bus Interface and Clock Distribution Application Notes for more detailed information on Schematic Orcad files, PAL equations, etc., for interface between PCI 9060 and other microprocessors.

7. How much power does the PCI 9060 use and what is its thermal characteristics?

The maximum power supply current (I_{cc}) for the PCI 9060 is 130mA. This power consumption is measured at V_{dd} equals 5.25V and PCI/Local clock equal 33Mhz.

8. Are any pull up or pull down resistors required for the PCI 9060?

For Add-in cards, PCI bus interface signals do not require any pull up or pull down since they are already handled by central resources on the motherboard or the backplane. For embedded systems, however, pull-ups are needed for PCI bus interface signals. All Local bus interface input signals contain internal pull-up through a 10K ohm resistor, and therefore a pull-down resistor is needed for any active high signal. External pull-up/pull-down resistors are required for Local input/output signals. For good design practice all input and input/output signals should be pull-up/pull-down to an in-active state when they are

not driven by anybody.

9. Is it possible to detect which revision level the PCI 9060 is?

Yes. The revision ID is located in the PCI Revision Register (Offset 08h). This register is readable from both PCI and Local side.

10. How do I initiate a FIFO transfer?

The FIFO transfer is initiated automatically. Data transfers from PCI-to-Local or Local-to-PCI during Direct Slave Accesses, Direct Master Accesses and DMA operation must go through the internal FIFO. The PCI 9060 manages all aspects of the FIFO on transfers initiated as a read or write.

11. How is the FIFO advance determined?

PCI 9060 contains integrated address counter. Local address is incremented automatically on the rising edge of the local clock unless wait state is added.

12. How do I perform Endian conversion?

PCI 9060 does not support Big/Little Endian conversion. However, all other PLX PCI bridge devices do. Then Endian conversion can be performed through software (programming local configuration register) or controlled by hardware (BIGEND# input pin).

13. How can I generate an interrupt to the PCI bus from the add on side?

There are two ways to generate an interrupt to the PCI bus from the add on side: 1. Write to the Local-to-PCI doorbell register and set a 1 to bit [8] and bit [9] of the Interrupt Control/Status register (LOC E8h). 2. Assert LINTi# to the PCI 9060 and set a 1 to bit [8] and bit [11] of the Interrupt Control/Status register (LOC E8h).

[Return to Top](#)
[Return to 9060 FAQ](#)