

Blind prefetching improves PCI Express-to-PCI-bridge performance

STANDARD BRIDGES ALLOW DESIGNERS TO COMBINE THE HIGH-PERFORMANCE PCI EXPRESS INTERCONNECT WITH LEGACY PCI-BUS ARCHITECTURE. ADVANCED BRIDGE FEATURES, SUCH AS BLIND PREFETCHING, BOOST DATA THROUGHPUT.

New high-performance bridging devices, available from a number of vendors, enable designers to migrate legacy PCI-bus designs to the advanced PCIe (Peripheral Component Interconnect Express) serial architecture. These bridges reduce the time it takes for data to pass through the system, thus minimizing clock-hogging latency. However, many of these bridges provide an additional function that further maximizes throughput: blind prefetching.

Blind prefetching allows the bridge to read a predefined amount of data, in sequential addresses, from PCIe memory and to buffer the data in the bridge whenever a device on the PCI side of the bridge reads one or more double words—data types representing 32 bits or 4 bytes—from memory on the PCIe side of the bridge. The amount of data the bridge buffers is typically more than what the PCI device initially requests. When using the blind-prefetch feature, some bridges burst as much as 4 kbytes of data in a single transaction, where-

as conventional PCIe-to-PCI bridges can transfer only one double word at a time during normal operation. Each double-word transfer requires some setup time to process the transaction, thus adding to the total latency through the bridge. The bridge's burst transaction minimizes the setup time and, ultimately, the latency to only one transaction. With blind prefetching, the initial latency penalty occurs only once for every 4 kbytes of transferred data. Therefore, the blind-prefetch capability allows for maximum read performance by minimizing the latency time for devices reading large amounts of sequential data through the bridge.

For PCIe-to-PCI reads, the memory-read request determines the number of bytes to read, and prefetching does not occur. However, for PCI-to-Pcie reads, prefetching occurs in the prefetchable space for all memory-read commands, including memory read, memory-read line, and memory-read multiple, that the PCI bus issues. The prefetchable-memory-base and limit-configuration registers determine whether to forward

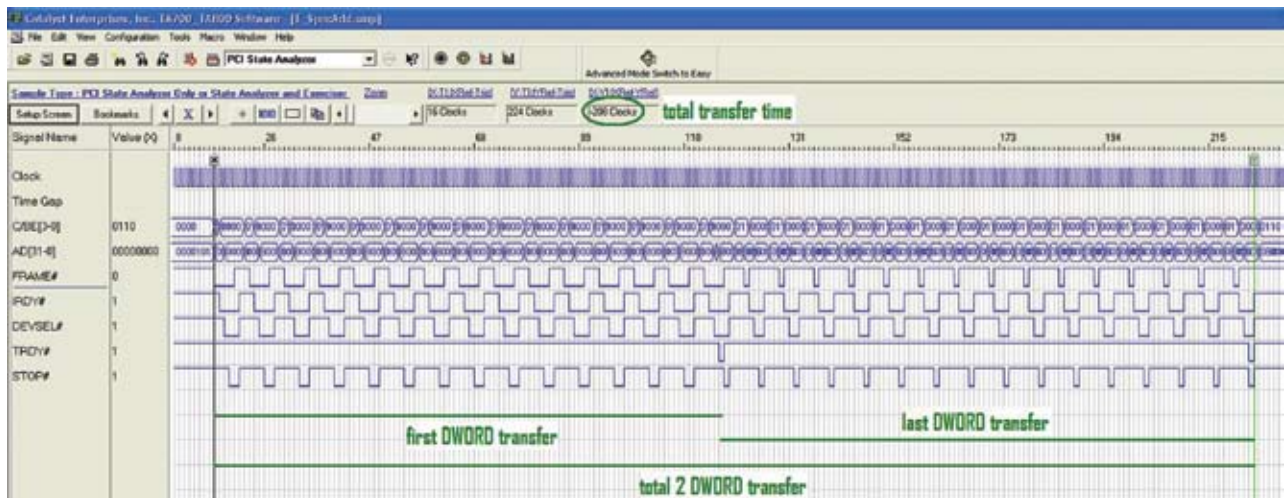


Figure 1 A two-double-word memory-read transaction without blind prefetch requires 208 clock cycles to complete.

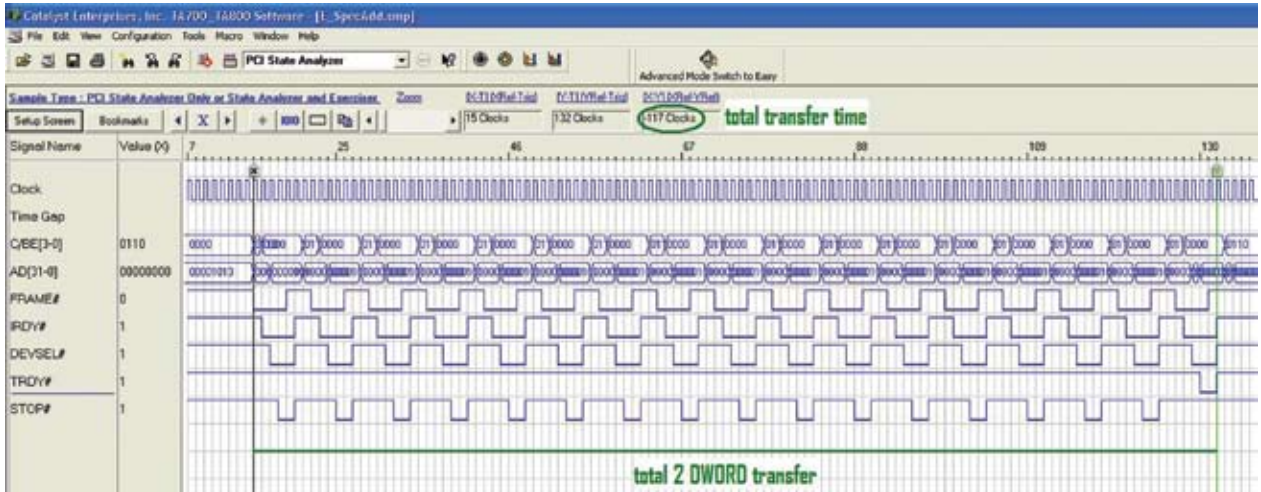


Figure 2 With blind prefetch, a two-double-word memory-read transaction finishes in 117 clock cycles.

prefetchable-memory transactions across the bridge. The primary bus forwards memory transactions that fall within the range that the prefetchable-memory-base and limit-configuration registers define. The secondary bus receives these transactions downstream, and the bridge ignores the memory transactions on the secondary bus that are within the range.

The primary bus ignores memory transactions that do not fall within this range and forwards them upstream from the secondary bus provided they are not in the address range that the set of memory-mapped I/O-address registers defines or that the VGA (video-graphics-array) mechanism forwards downstream. For prefetching to occur, memory-read commands

must support the blind-prefetch feature. This feature greatly improves read performance because the bridge can burst its prefetchable data onto the PCI bus whenever the endpoint requests it.

Figure 1 shows a two-double-word transfer without blind prefetch. It takes 208 clock cycles for the transaction to complete from when you first assert the

frame command. The transfer separates into two one-double-word transfers: The TRDY# (transfer-ready) signal asserts twice, and each assertion is only one clock cycle. Each transfer inherits a setup time, which ultimately adds to the increased

latency through the bridge. **Figure 2** depicts the same two-double-word with blind prefetch. The total transfer time improves to only 117 clock cycles, from 208 cycles. Although the two-double-word transfer still breaks down into two separate transactions, the TRDY# signal asserts once for two clock cycles but

with only one setup time for the whole transaction.

The two-double-word transfer saves 91 clock cycles, delivering a 44% increase when using the blind-prefetch capability of a PCI-to-PCIe bridge. Therefore, the blind-prefetch feature reduces latency by nearly 80% when transferring 4 kbytes of data in PCI-to-PCIe designs and enables a fivefold increase in transfer rate over normal operation.

A number of the PCI-to-PCIe bridges now on the market implement the prefetching algorithm by simply configuring the blind-prefetch-enable bit

in the device-specific-control register. This approach enables a memory-read command on the PCI bus, allowing the PCIe memory space to read at least one cache line from the PCIe interface. The bridge can read additional double words carrying 0 to 4 kbytes of data by setting the PCI-control-regis-

ter programmed-prefetch-size field.

To maximize the bridge's read prefetch size, you must follow the following steps:

1. Set the maximum-read-request-size field in the PCIe device-control register to the maximum of 4096 bytes.
2. Set the programmed-prefetch-size field in the PCI-control register to the maximum of 4096 bytes.
3. Set the PCI-bus-latency-timer register to FFh to ensure that the host does not prematurely release the bus. In forward-mode designs, you may need to increase the PCI-latency timer of the bridge's downstream PCI-bus master, secondary PCI-latency timer, or both to ensure that the device does not prematurely release the bus. The precise values depend on the traffic pattern.

The maximum-read-request size defines the upper limit of the programmed-

MORE AT EDN.COM ▶

+ Go to www.edn.com/ms4270 and click on Feedback Loop to post a comment on this article.

prefetch size, and the programmed-prefetch size is less than or equal to the maximum-read-request size. Program the maximum prefetch size to the largest possible for the transfer without exceeding the transfer size. If the prefetch size is greater than the number of bytes the bridge is reading, the reading and discarding of excess prefetch data will affect performance. If you require a higher speed bus, you can adjust the PCLKO frequency, or you can set the PCLKO clock-frequency field in the device-initialization register to a maximum frequency of 100 MHz.

To maintain optimum performance, the anticipated read-burst size on the PCI bus must closely match one of the programmed-prefetch-size settings of the bridge. If the read-burst size is significantly smaller than the programmed-prefetch size, the bridge must discard the unused data after each read. This approach can negatively impact read performance by generating extra delay. Also, varying the read-burst size can affect the performance for some read transactions. In an ideal situation, the endpoint implements reads as bursts through DMA control, so that the read-burst size is fixed or controllable.

With multiple vendors offering a new generation of PCIe-bridging devices, designers can extend the lives of and add performance to boards and systems based on the conventional PCI bus. These PCI-to-PCIe migrations are introducing design complexities that advanced functions, such as blind prefetching, can alleviate. Designers with an understanding of such functions can minimize development efforts and time to market for the next generation of PCIe-based systems. **EDN**

AUTHOR'S BIOGRAPHY



Eugene Cabanban is a senior product-marketing engineer at PLX Technology (Sunnyvale, CA). He holds a bachelor's degree in computer engineering

from the University of California—Irvine and a master's in business administration from Santa Clara University (Santa Clara, CA). You can reach him at ecabanban@plxtech.com.