

**PCI-X, A NEW BUS PROTOCOL, AIMS TO PROVIDE HIGH-SPEED I/O FOR SERVERS AND WORKSTATIONS. LEARN THE FUNDAMENTALS OF PCI-X AND HOW IT DIFFERS FROM PCI.**

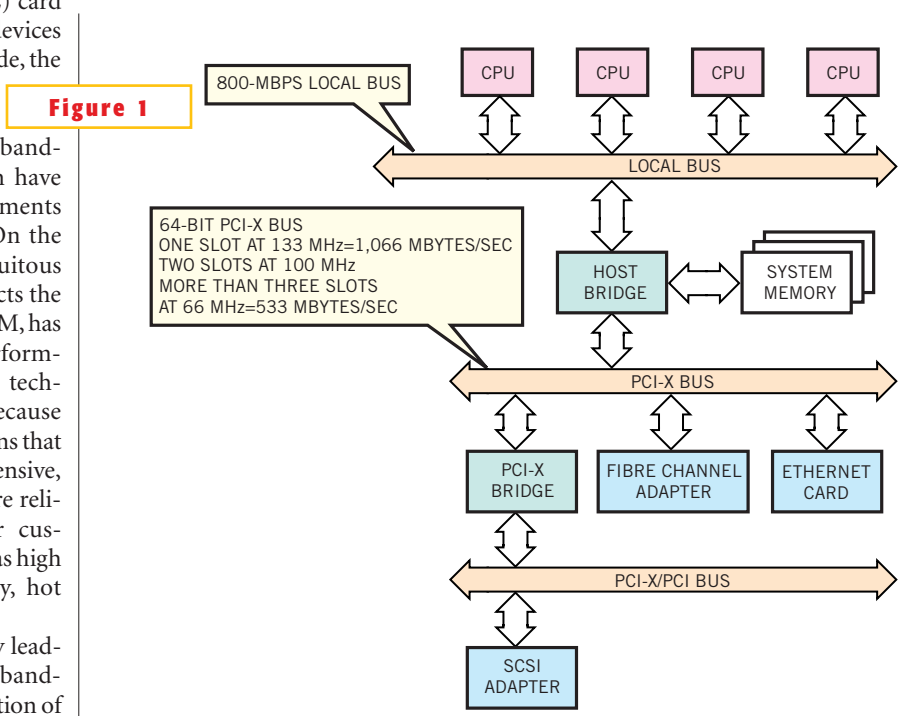
# PCI-X boosts bus bandwidth to 1 Gbps

**H**IGH-PERFORMANCE SERVERS that exchange massive amounts of data with storage devices and networks-interface cards (NICs) drive the Internet/e-commerce world. Because the server world is performance-driven, processor and system vendors find that system throughput is vital for differentiating products. A typical server comprises a single processor or dual processors with a peripheral-component-interconnect (PCI) bus and a NIC for network communication and an Ultrawide SCSI card or Fibre Channel (FC) card for connection to storage devices (Figure 1). In the past decade, the processor speed, Ethernet speed, asynchronous-transfer-mode (ATM) bandwidth, and FC bandwidth have undergone rapid advancements with increases in speed. On the other hand, PCI, the ubiquitous I/O bus, which interconnects the CPU to Ethernet, FC, or ATM, has not evolved to meet the performance requirements of the technologies that rely on it. Because many of the new applications that run on servers are I/O-intensive, they need a faster and more reliable I/O channel. Server customers want features such as high bandwidth, error recovery, hot swapping, and reliability.

Server-business industry leaders have recognized the bandwidth and scalability limitation of PCI and have proposed several new methods for solving the PCI's I/O problems. The two cur-

rent efforts center on PCI-X (an extension of the PCI standard) and Infiniband, a switch-based I/O interconnect. PCI-X is backward-compatible with PCI, so it preserves investments in PCI software and hardware. The Infiniband, a new bus architecture, requires new drivers and OS support.

The joint efforts of Compaq, IBM, and HP (www.compaq.com, www.ibm.com, www.hp.com), major players in the server market, resulted in PCI-X. This evolutionary standard increases the clock fre-



**Figure 1**

**A typical PCI-X system includes one or two processors, a PCI bus, and a network-interface card.**

quency of the current PCI standard and maximizes system throughput.

**WHY IS PCI-X IMPORTANT?**

PCI, the basis for PCI-X, is a popular bus standard that dominates most PC-, workstation-, and server-based systems. Recently, it has extended its reach to communications devices, such as routers; NICs; and embedded systems via compact PCI. The peak bandwidth of PCI is 512 Mbytes/sec, and, in the past, most applications accepted this bandwidth. These days, a server may have an FC port, an Ethernet port, and a SCSI card. With FC running at 1 Gbps; Ethernet, at 100 Mbps; and Ultrawide SCSI, PCI is easily saturated. Designers drafted PCI-X to support a peak bandwidth of 1 Gbps and to maintain backward compatibility with PCI.

Designers created PCI-X to solve the ever-increasing need for greater I/O bandwidth. PCI-X provides some relief for the congestion in server I/O (Table 1). PCI-X provides this capability by increasing the bus frequency of PCI from 66 to 133 MHz and by increasing the efficiency of bus traffic. The enhanced bus-usage features of PCI-X should increase system bandwidth, even though it runs at the same bus speed as conventional PCI. Storage and networking devices, such as Ethernet, Ultrawide SCSI, and FC will benefit from increased bandwidth.

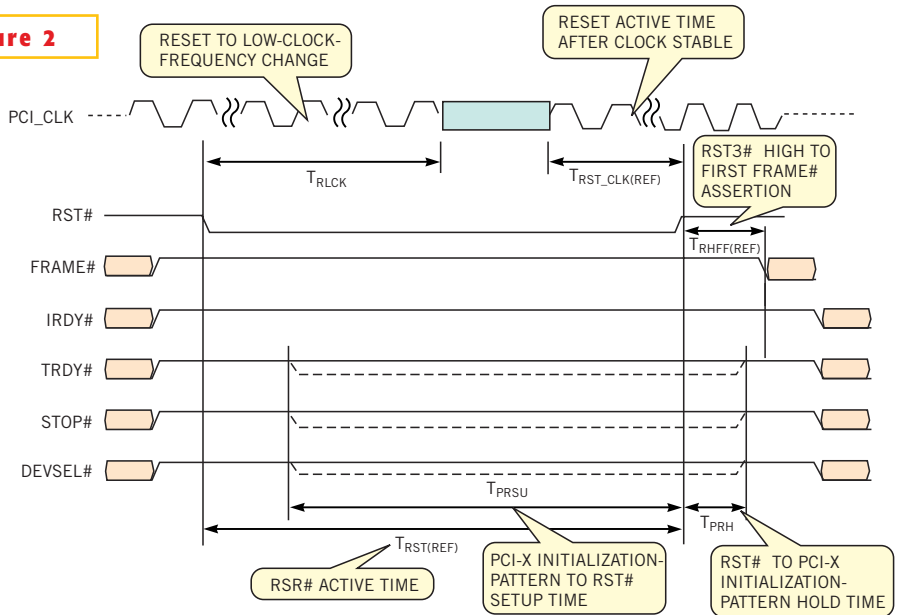
The number of slots available at 66 MHz systems also limits PCI. Due to stringent electrical requirements, PCI systems at 66 MHz can support only one or two slots. PCI-X now solves this limitation because it can support four or more slots at 66 MHz.

**WHAT IS PCI-X?**

Designers developed PCI-X as an extension of the PCI-bus architecture. It extensively employs PCI by using the same signal names, naming conventions, and several other features. PCI-X surpasses PCI by increasing the operating frequency and the maximum peak bandwidth. PCI-X also includes several necessary protocol changes because PCI-X is a register-to-register protocol.

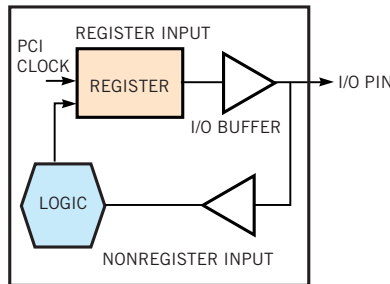
PCI-X touts several new features resulting in an increase in speed and no waiting. It also adds an attribute phase, split transactions, and a new initialization sequence.

**Figure 2**



After determining the system composition, the source bridge drives an initialization pattern, and RST# is asserted.

**Figure 3**



The I/O structure of a conventional PCI device latches the output signal but not the input.

A PCI-X device can run at 33 to 133 MHz. In a PCI-X system, the system runs at the speed of the slowest device. Due to backward compatibility requirements, all PCI-X devices must be able to run at lower speeds to work in older systems or PCI-X systems with a slower device or PCI device. At 133 MHz, the PCI-X bus can accommodate only one PCI-X device and, at 100 MHz, can accommodate two PCI-X devices. At 66 MHz, it can accommodate four or more devices. Note that designers introduced the 100-MHz version to give designers the opportunity to trade off between number of slots and bus speed/bandwidth. You can use a PCI-X bridge at higher speed to increase

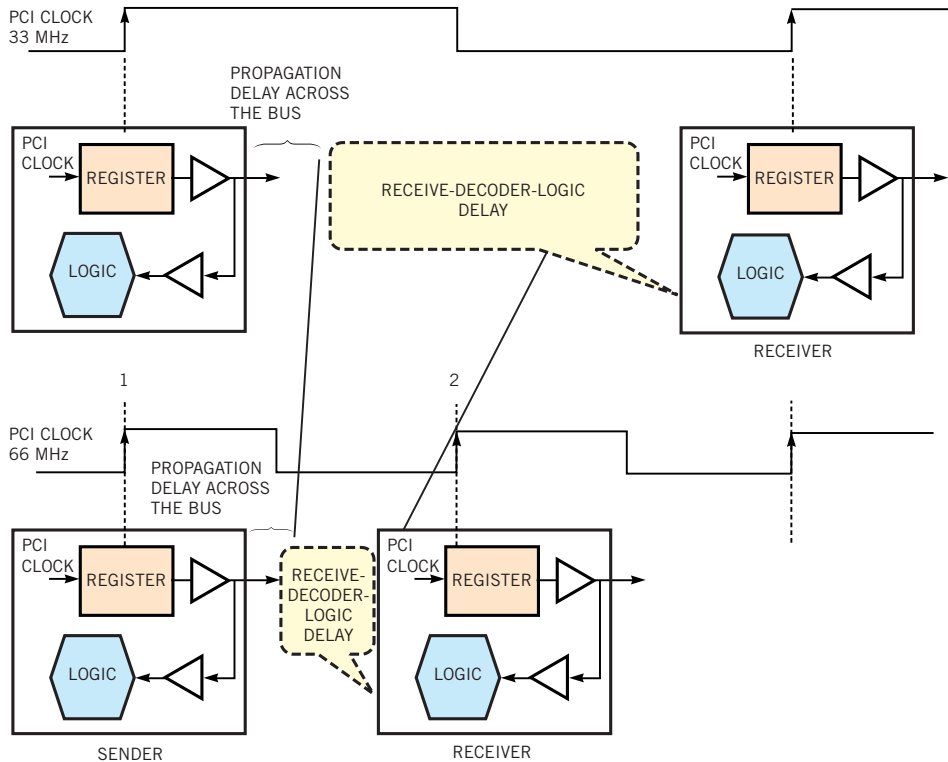
the number of available slots in the system. PCI-X also supports 32- or 64-bit-wide data buses widths.

PCI-X increases bus usage by adding features such as wait-state and disconnect rules. PCI-X no longer allows targets and initiators to insert wait states during data transfers and allows targets to insert wait states only at the beginning of data transfers. The target and master, once they commit to a transfer, cannot insert any wait states during the transfer. Fewer wait states allow for increased bus usage at the expense of having more buffers in the design. PCI-X also optimizes the

bus bandwidth by allowing for larger burst length and restricting the areas at which the initiator or target can stop a data transfer. A new concept, the allowable disconnect boundary (ADB), serves as the basis for this feature. Devices can stop transfers only at an ADB, which is 128 bytes long.

Designers also added an attribute phase to every transaction. The attribute immediately follows the address phase. The attribute phase helps optimize system performance by including such properties as the byte count, bus number, and device number as part of the transaction. It also helps the bridge optimize performance during data transfers. The

**Figure 4**



At 66 MHz, the setup, skew, and settling time leaves only 9 nsec for the PCI 2.2 device to process the data.

attribute phase also assists in interfacing the PCI-X bus to packet-based systems in which the length of a transaction is communicated at the beginning of the transaction. This feature should make it easier to make the transition between PCI-X systems and serial buses.

Perhaps the most significant new feature of PCI-X is the addition of split transactions. Split transactions primarily work to replace delayed transactions in PCI. Delayed transactions in the PCI architecture refer to a transaction (read) by a PCI device in which the target device terminates the transfer with a retry and fetches the data from memory. The master later retries the transaction, and the target completes it if it has the data ready. A PCI-X split transaction completes in two or more transactions. It replaces delayed transactions in PCI. It also solves the major drawback of PCI: delayed transactions, in which a PCI device terminates a read (or write). These transactions force the initiator to repeatedly retry the transaction until the target has the data, significantly reducing system bandwidth. A PCI-X device uses split transactions to solve this problem; however, this approach increases the com-

plexity of PCI-X devices. Only PCI-X bridges and initiators must support split transactions. Host bridges should eventually also support this feature to increase system throughput.

You can configure a PCI-X device or add-in-card to function in PCI-X mode or in conventional PCI mode. At startup, the system goes through an initialization sequence to determine the components that detect and initialize the devices that make up the system. You can meet this initialization requirement either by using add-in-cards that support PCI-X or by using the PCI-X initialization pattern. The output of the PCIXCAP and M66EN pins indicates whether the board supports PCI-X and at which frequency. Table 2 shows how an add-in-card indicates its capability with the combination of the M66EN and PCIXCAP pins.

You can program a PCI-X device to support conventional PCI or PCI-X based on the initialization patterns. After determining the system composition, the source bridge drives an initialization pattern, and RST# is asserted (Figure 2). All devices attached to the bus sample this initialization pattern and set up an op-

erating speed and a mode (PCI or PCI-X). When the bus is idle (FRAME# and IRDY# de-asserted) and the device asserts one or more of DEVSEL#, TRDY#, and STOP#, the device enters PCI-X mode; otherwise, it enters conventional PCI mode.

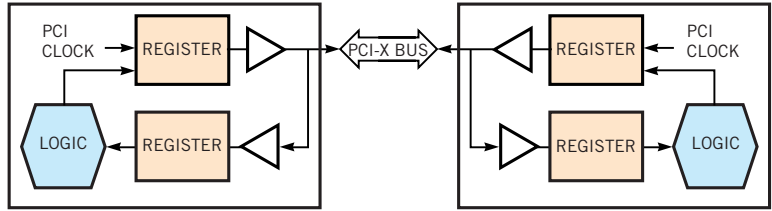
In a conventional PCI2.2-device I/O-pin setup, the output pin of the device is latched and sampled at every rising edge of the PCI clock (Figure 3). On the other hand, the input has no storage register, and this nonlatched input becomes a bottleneck when the bus frequency exceeds 66 MHz.

Figure 4 shows the increasing frequency effect on design timing. One device, the sender, drives the signal, and the receiver receives the signal. At 33 MHz, the clock period is 30 nsec, meaning that the receiver has 30 nsec to latch a signal and respond to that signal while meeting the setup timing requirement of 7 nsec. After subtracting the bus (propagation) delay of 2 nsec and clock skew of 1 nsec, the receiver logic has 20 nsec to meet the setup time. At 33 MHz, it is easy for the receiver to meet the setup time requirement. Unfortunately, this situation is more difficult for PCI 2.2 (66 MHz). At

66 MHz (clock period of 15 nsec) the receiver has 15 nsec to latch a signal and respond to that signal while meeting setup timing requirement of 3 nsec. Subtracting 2 nsec for flight time and 1 nsec for clock skew leaves 9 nsec for the device's logic to process the signal. This timing requirement is difficult to meet, especially for the address decoder.

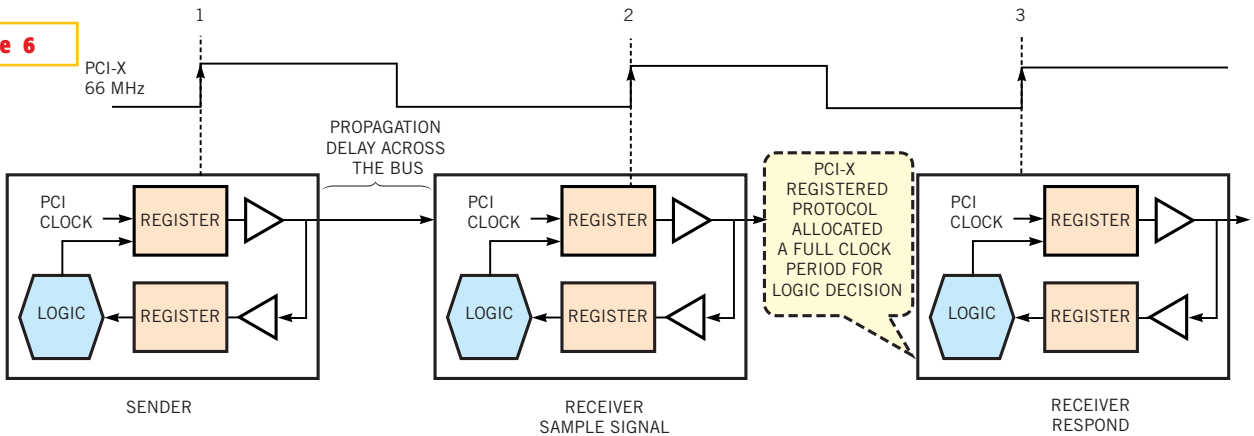
Register-to-register design serves as a basis for PCI-X architecture. PCI-X in-

**Figure 5**



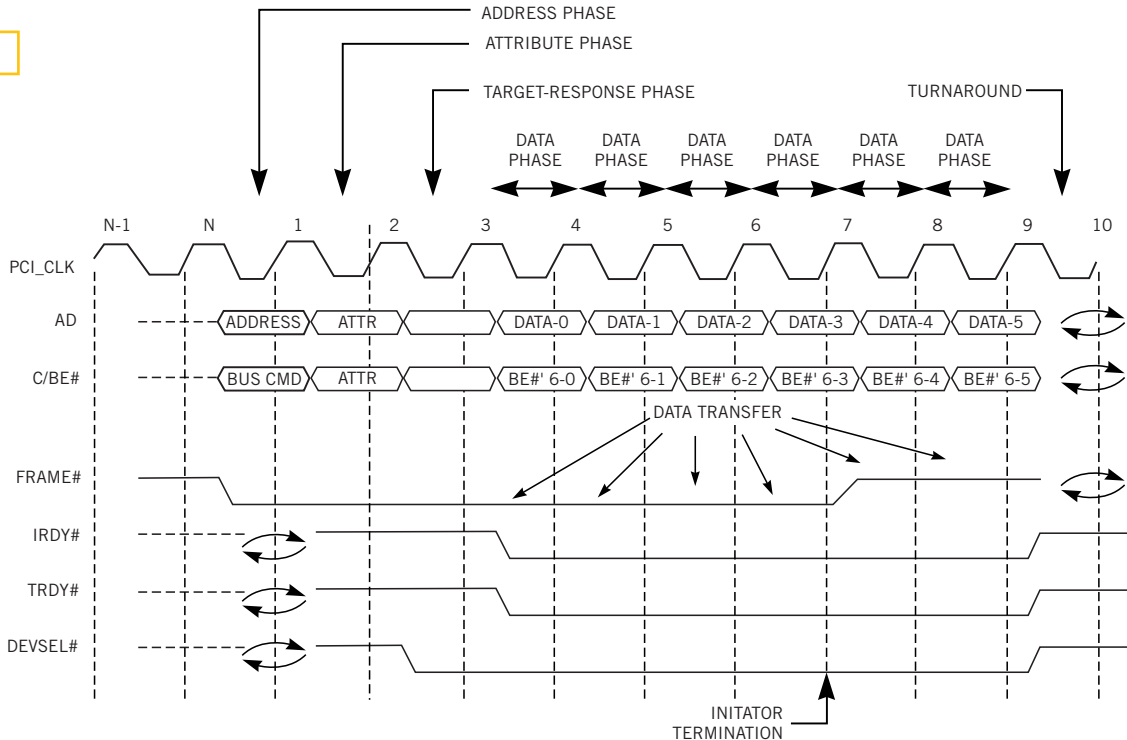
**A PCI-X device latches both the input and the output data.**

**Figure 6**



**With storage registers on all I/O signals, PCI-X allows a full clock cycle for a device to respond after latching a signal.**

**Figure 7**



**The PCI-X write cycle includes an attribute phase after the address phase.**

puts and outputs are latched (Figure 5). By including a register on the inputs, you have a full clock cycle for a device to respond after latching a signal (Figure 6).

**BASIC PCI-X PROTOCOL**

Compare a typical PCI protocol with PCI-X (figures 7 and 8). As in PCI, the master (initiator) first requests the bus. After receiving GNT#, the master asserts FRAME# to start the bus cycle. At the same time, the master drives the address onto the AD bus. The cycle in which the address is driven is the address phase. Immediately following the address phase, the attribute phase occurs, during which the master drives the attribute on the AD bus. After the attribute phase is the target-response phase in which the target responds to the cycle with DEVSEL#. The target may take as many as four clocks in asserting DEVSEL# depending on the speed of the target. The target indicates that it is ready to transfer data by asserting TRDY#. A data phase occurs in any clock in which IRDY# and TRDY# are asserted. During the initiator termination, the initiator indicates that it is ready to end the transfer by deasserting FRAME# two clocks before deasserting IRDY#. In the next phase, turnaround, both the initiator and the target turn around their signals. PCI essentially has a subset of the

**TABLE 1—BANDWIDTH OF BUS IN PCI AND PCI-X MODES**

Specification	Bus width (bits)	Bus frequency (MHz)	Maximum peak bandwidth (Mbps)
PCI 2.2	32	33	133
PCI 2.2	64	33	266
PCI 2.2	64	66	533
PCI-X 1.0	64	133	1066

**TABLE 2—M66EN AND PCIXCAP ENCODING**

M66EN		Conventional-device frequency capability (MHz)	PCI-X device frequency capability (MHz)PCIXCAP
Grounded	Grounded	33	Not capable
Not connected	Grounded	66	Not capable
Ground	Pulldown	33	PCI-X 66
Not connected	Pulldown	66	PCI-X 66
Ground	Not connected	33	PCI-X 133
Not connected	Not connected	66	PCI-X 133

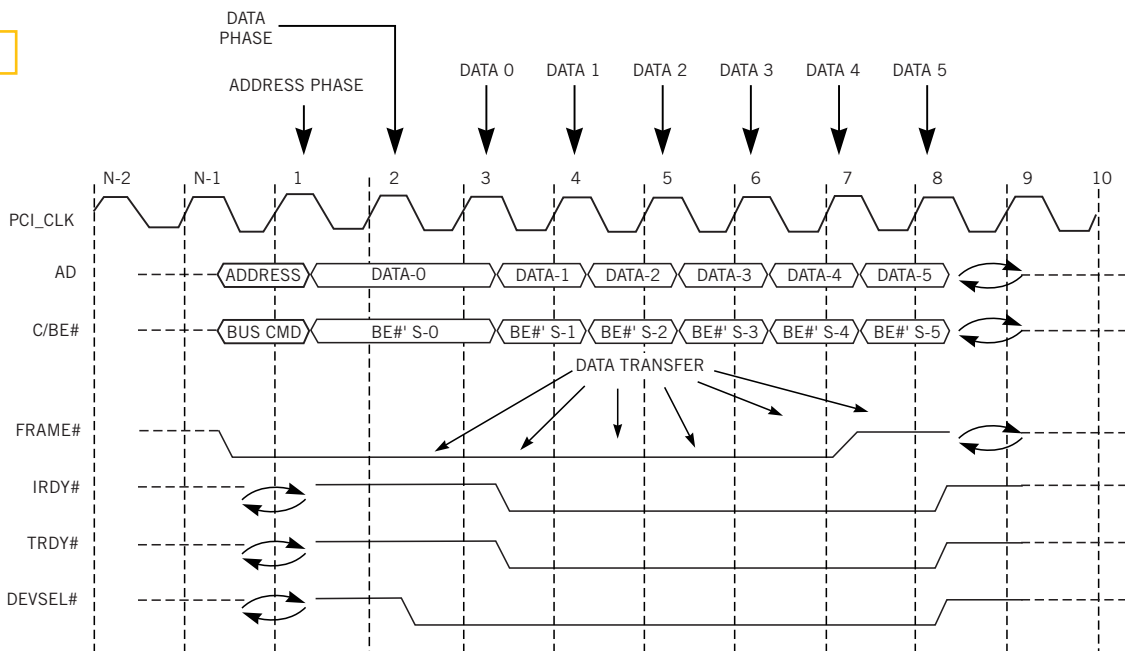
**TABLE 3—PCI-X INITIALIZATION PATTERN**

DEVSEL#	STOP#	TRDY#	Mode	Minimum clock frequency (MHz)	Maximum clock frequency (MHz)
1	1	1	Conventional 33	0	33
			Conventional 66	33	66
1	1	0	PCI-X	50	66
1	0	1	PCI-X	66	100
1	0	0	PCI-X	100	133

phases of the PCI-X. PCI-X adds one extra clock due to the attribute phase. Also, assuming that both devices respond with

the fastest possible DEVSEL#, PCI can save one more clock and complete the same transfer in eight clocks.

**Figure 8**



When you install them in a legacy system, PCI-X devices must be compatible with standard PCI-bus activity.

With all the new enhancements to PCI to support PCI-X, the specification's provisions allow for an easy migration from PCI to PCI-X.

PCI-X systems support slot interoperability. To achieve easy migration of existing cards to PCI-X system, PCI-X systems need to support existing PCI cards. If you plug a PCI 2.2 card into a PCI-X system, the PCI-X system must slow down to the speed of the PCI card at 33 MHz. Conversely, PCI systems must accept PCI-X cards; however, this combination also requires that the system run at the lowest speed (33 MHz).

Designers created PCI-X systems to support existing software (OS and device drivers) without any modifications. This function serves to preserve investment in software. Note that, to take advantage of new features, such as hot-plug or performance-tuning registers, you have to change the device drivers and OS.

The PCI-X specification recommends an easy migration path from PCI commands to PCI-X commands and vice versa. For example, you can replace the

memory-write and invalidate command with a memory-write block to perform the same functions.

PCI-X uses the same pins and pin functions as PCI. For example, FRAME# indicates the start of a PCI and PCI-X cycle. Although the timing and protocol has changed, the signals still perform the same general functions.

### WHAT'S NEXT FOR PCI-X?

Clearly, PCI-X initially targeted the server market. Several vendors now develop chips that work with PCI-X. With clear advantages over PCI, such as performance, backward compatibility, and an easy migration path from PCI, companies designing for the server market prefer PCI-X. □

### REFERENCES

1. Compaq PCI-X enablement program [www.compaq.com/products/server/technology/pci-x-enablement.html](http://www.compaq.com/products/server/technology/pci-x-enablement.html).
2. Compaq Computer Corp, PCI-X: An Evolution of the PCI Bus [www.compaq.com/support/techpubs/white](http://www.compaq.com/support/techpubs/white)

[papers/tc990903tb.html](http://papers/tc990903tb.html).

3. PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0.

---

### AUTHORS' BIOGRAPHIES

*Laverty Nwaekwe is a senior R&D engineer at Synopsys Inc (Beaverton, OR), where he has worked for six years. In his current position, he develops bus-interface models and test suites. He has an MS in computer engineering from the University of Southern California (Los Angeles) and a BSEE from the University of Benin (Nigeria). His spare-time interests include skiing and soccer.*

*Syed Chowdhury is a technical marketing manager at Synopsys, where he has worked for more than two years. In his current position, he provides new-product direction for bus-functional and processor-software models. He has an MSEE and a BSEE from Portland State University (Portland, OR) and is pursuing a degree in engineering management, also from Portland State. His spare-time interests include running, playing tennis, and reading biographies.*