PCI Tutorial

© 2000 Xilinx, Inc.
All Rights Reserved

Agenda

PCI Fundamentals

- PCI Local Bus Architecture
- PCI Signals
- Basic Bus Operations
- PCI Addressing and Bus Commands
- PCI Configuration
- Electrical and Timing Specifications
- 64-bit Extension
- 66-MHz Overview
- PCI Variations

Xilinx PCI Solution

- The PCI Challenge
- Xilinx PCI with Design Examples
- Xilinx PCI Design Flow Overview
- Available Resources

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Fundamentals and Concepts

Xilinx PCI
The CORE of a GREAT IDEA

PCI Local Bus Architecture

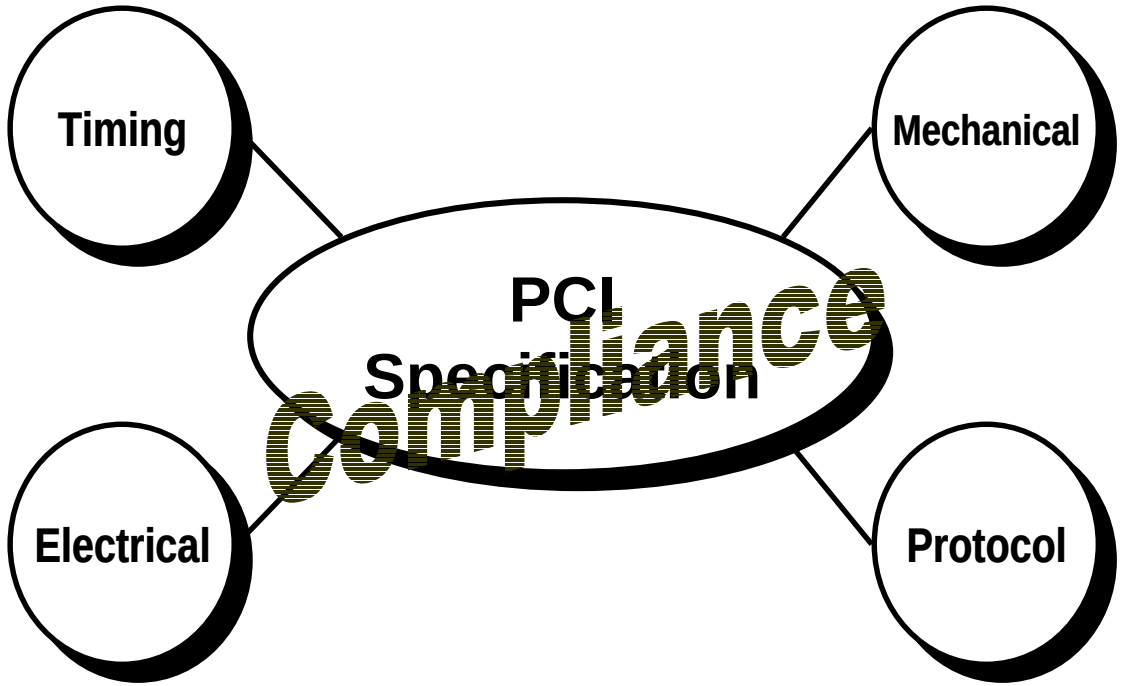
Xilinx PCI

XILINX®

© 2000 Xilinx, Inc.
All Rights Reserved

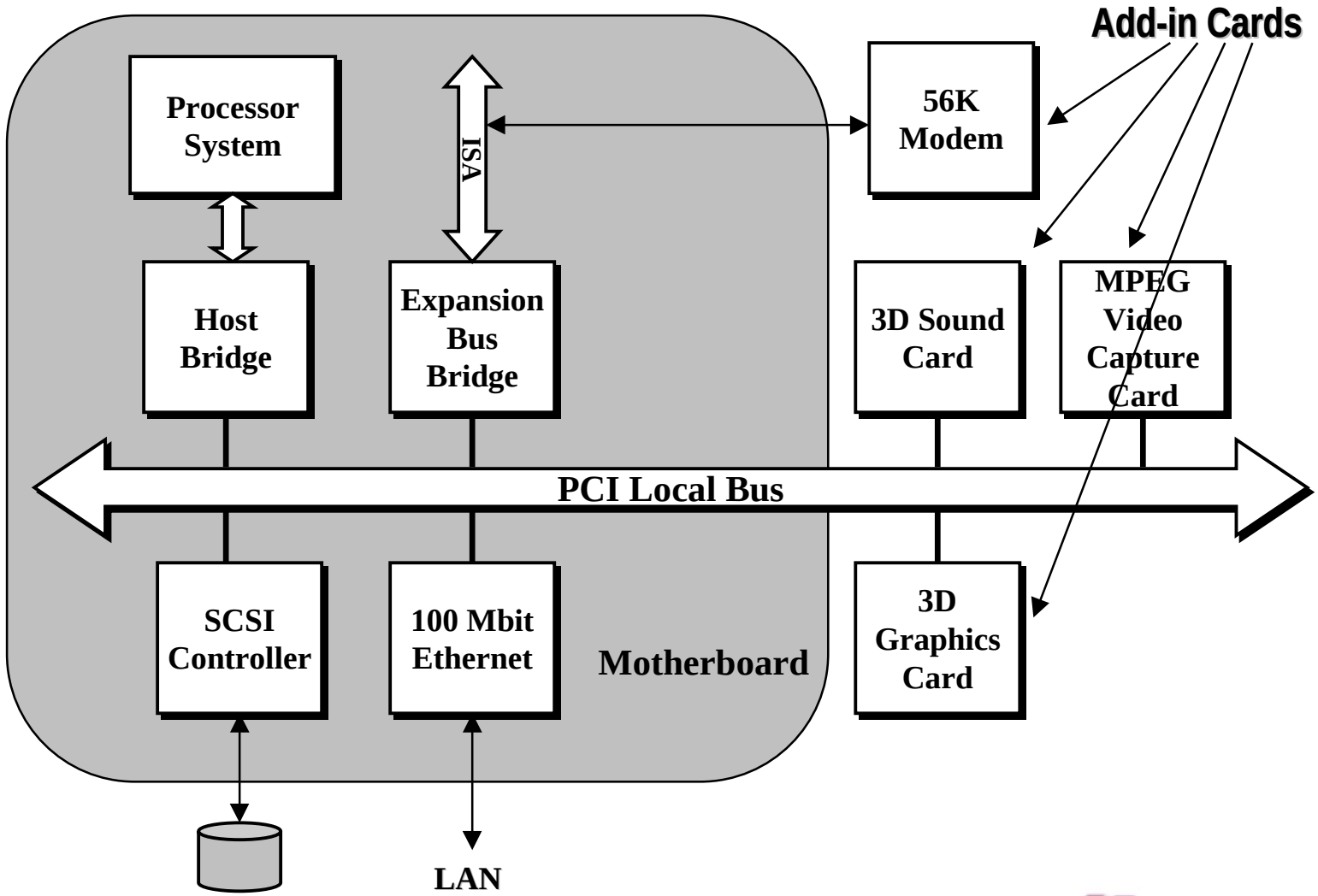
Overview of the PCI Specification

- ***The PCI Local Bus Specification*** **covers many different requirements for PCI compliance**



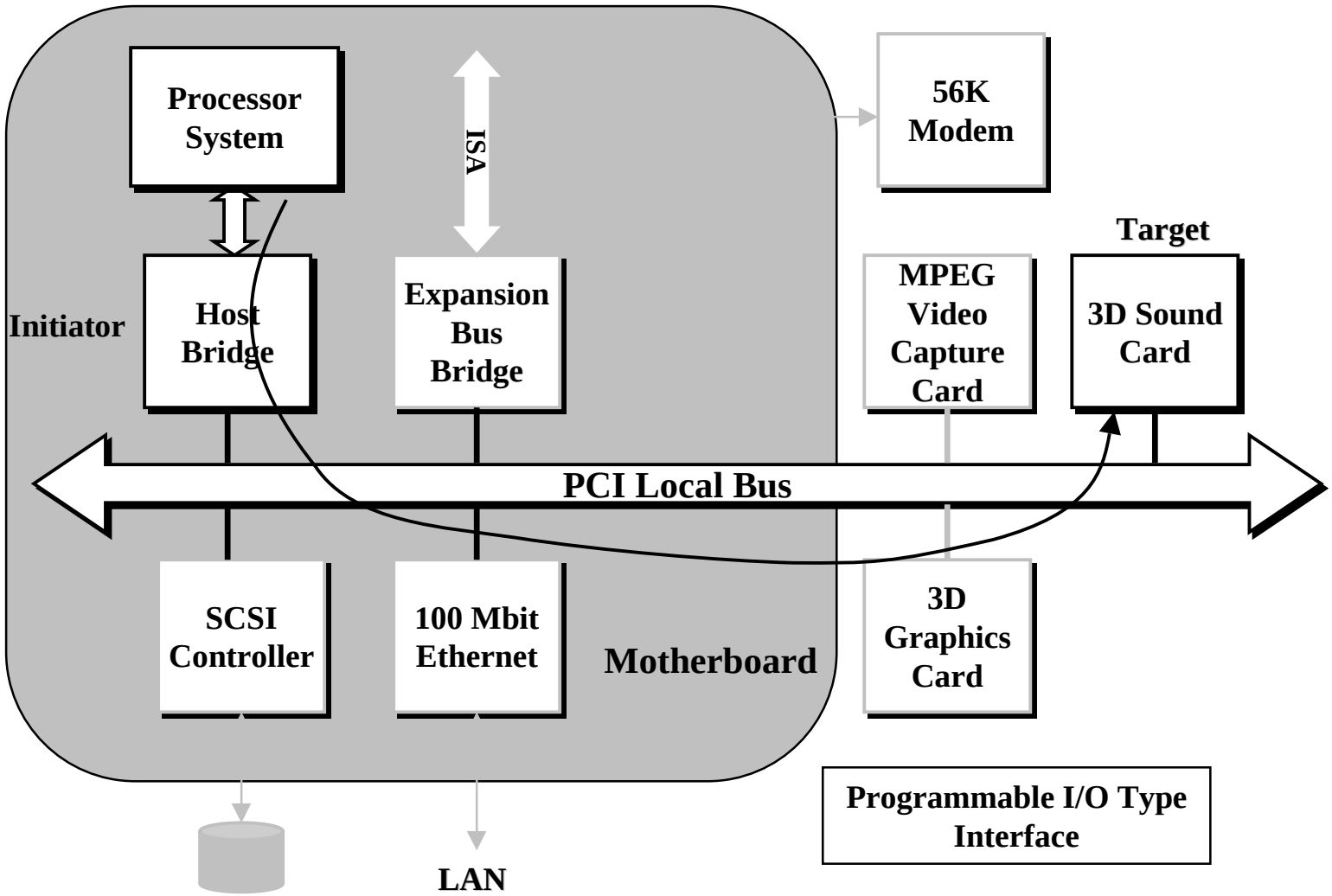
© 2000 Xilinx, Inc.
All Rights Reserved

Basic Bus Architecture



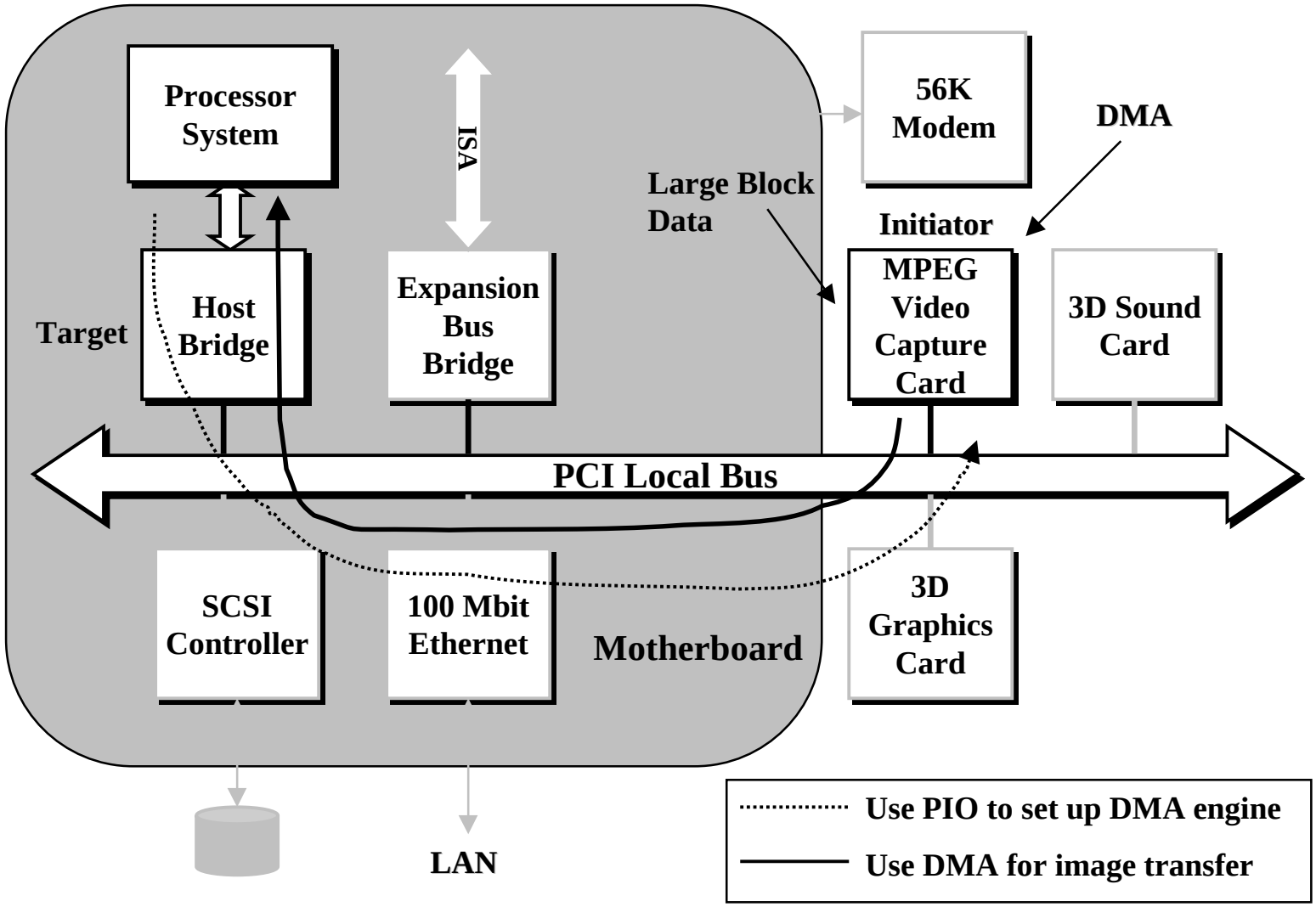
© 2000 Xilinx, Inc.
All Rights Reserved

Sample Transaction – PIO



© 2000 Xilinx, Inc.
All Rights Reserved

Sample Transaction – DMA



© 2000 Xilinx, Inc.
All Rights Reserved

Key Terms

- **Initiator**
  - *Or Master*
  - *Owns the bus and initiates the data transfer*
  - *Every Initiator must also be a Target*
- **Target**
  - *Or Slave*
  - *Target of the data transfer (read or write)*
- **Agent**
  - *Any initiator/target or target on the PCI bus*

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Signals

Xilinx PCI

© 2000 Xilinx, Inc.
All Rights Reserved

Clock and Reset

- **CLK**
  - *PCI input clock*
  - *All signals sampled on rising edge*
  - *33MHz is really 33.33333MHz (30ns clk. period)*
  - *The clock is allowed to vary from 0 to 33 MHz*
    - The frequency can change “on the fly”
    - Because of this, no PLLs are allowed
- **RST#**
  - *Asynchronous reset*
  - *PCI device must tri-state all I/Os during reset*

© 2000 Xilinx, Inc.
All Rights Reserved

Transaction Control
Initiator Signals

- **FRAME# – I/O**
  - *Signals the start and end of a transaction*
- **IRDY# – I/O**
  - *"I-Ready"*
  - *Assertion by initiator indicates that it is ready to send or receive data*

© 2000 Xilinx, Inc.
All Rights Reserved

Transaction Control
Target Signals

- **DEVSEL# – I/O**
  - *Device select*
  - *Part of PCI's distributed address decoding*
    - Each target is responsible for decoding the address associated with each transaction
    - When a target recognizes its address, it asserts DEVSEL# to claim the corresponding transaction

© 2000 Xilinx, Inc.
All Rights Reserved

Transaction Control
Target Signals

- **TRDY# – I/O**
  - *"T-Ready"*
  - *When the target asserts this signal, it tells the initiator that it is ready to send or receive data*
- **STOP# – I/O**
  - *Used by target to indicate that it needs to terminate the transaction*

© 2000 Xilinx, Inc.
All Rights Reserved

Transaction Control
Configuration Signals

- **Uses the same signals as the target, plus . . .**
- **IDSEL – I**
  - *"ID-Sel"*
  - *Individual device select for configuration – one unique IDSEL line per agent*
  - *Solves the "chicken-and-egg" problem*
    - Allows the system host to configure agents before these agents know the PCI addresses to which they must respond

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Address and Data Signals

- **AD[31:0] – I/O**
  - *32-bit address/data bus*
  - *PCI is little endian (lowest numeric index is LSB)*
- **C/BE#[3:0] – I/O**
  - *4-bit command/byte enable bus*
  - *Defines the PCI command during address phase*
  - *Indicates byte enable during data phases*
    - Each bit corresponds to a “byte-lane” in AD[31:0] – for example, C/BE#[0] is the byte enable for AD[7:0]

© 2000 Xilinx, Inc.
All Rights Reserved

Address and Data Signals

- **PAR – I/O**
  - *Parity bit*
  - *Used to verify correct transmittal of address/data and command/byte-enable*
  - *The XOR of AD[31:0], C/BE#[3:0], and PAR should return zero (even parity)*
    - In other words, the number of 1's across these 37 signals should be even

© 2000 Xilinx, Inc.
All Rights Reserved

Arbitration Signals

- **For initiators only!**
- **REQ# – O**
  - *Asserted by initiator to* **request** *bus ownership*
  - *Point-to-point connection to arbiter – each initiator has its own REQ# line*
- **GNT# – I**
  - *Asserted by system arbiter to* **grant** *bus ownership to the initiator*
  - *Point-to-point connection from arbiter – each initiator has its own GNT# line*

© 2000 Xilinx, Inc.
All Rights Reserved

Error Signals

- **PERR# – I/O**
  - *Indicates that a data* **parity error** *has occurred*
  - *An agent that can report parity errors can have its PERR# turned off during PCI configuration*
- **SERR# – I/O**
  - *Indicates a serious* **system error** *has occurred*
    - Example: Address parity error
  - *May invoke NMI (non-maskable interrupt, i.e., a restart) in some systems*

© 2000 Xilinx, Inc.
All Rights Reserved

Basic Bus Operations

Xilinx PCI

© 2000 Xilinx, Inc.
All Rights Reserved

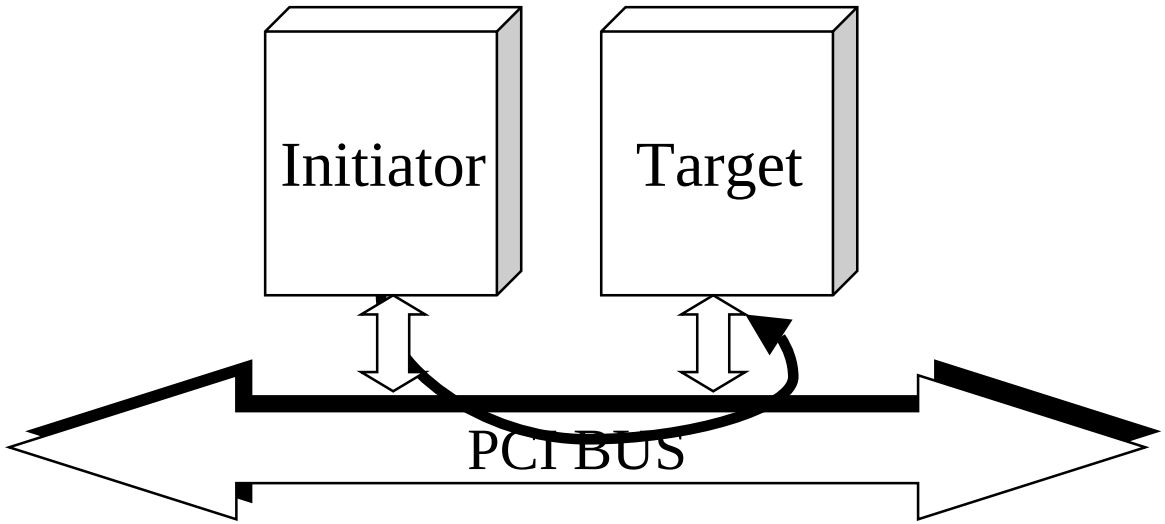
Terms

- **Doubleword**
  - *32 bits, most often known as a "DWORD"*
- **Quadword**
  - *64 bits, sometimes known as a "QWORD"*
- **Burst transaction**
  - *Any transaction consisting of more than one data phase*
- **Idle state (no bus activity)**
  - *Indicated by FRAME# and IRDY# deasserted*

© 2000 Xilinx, Inc.
All Rights Reserved

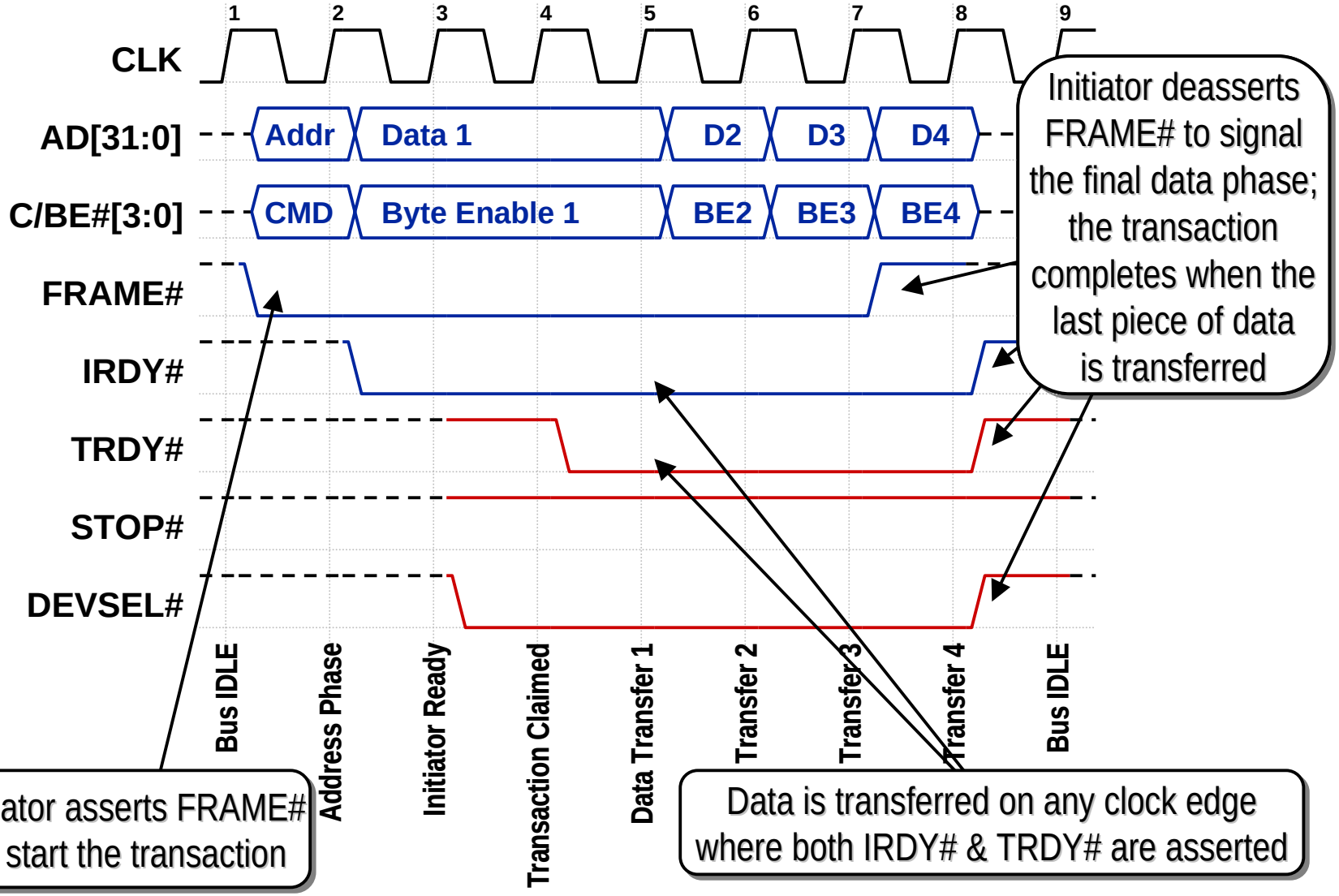
Example #1 – Basic Write

- A four-DWORD burst from an initiator to a target



© 2000 Xilinx, Inc.
All Rights Reserved

Basic Write Transaction



© 2000 Xilinx, Inc.
All Rights Reserved

Write Example – Things to Note

- **The initiator has a phase profile of 3-1-1-1**
  - *First data can be transferred in three clock cycles (idle + address + data = "3")*
  - *The 2nd, 3rd, and last data are transferred one cycle each ("1-1-1")*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Write Example – Things to Note

- **The target profile is 5-1-1-1**
  - *Medium decode – DEVSEL# asserted on 2nd clock after FRAME#*
  - *One clock period of latency (or wait state) in the beginning of the transfer*
    - DEVSEL# asserted on clock 3, but TRDY# not asserted until clock 4
  - *Ideal target write is 3-1-1-1*
- **Total of 4 data phases, but required 8 clocks**
  - *Only 50% efficiency*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

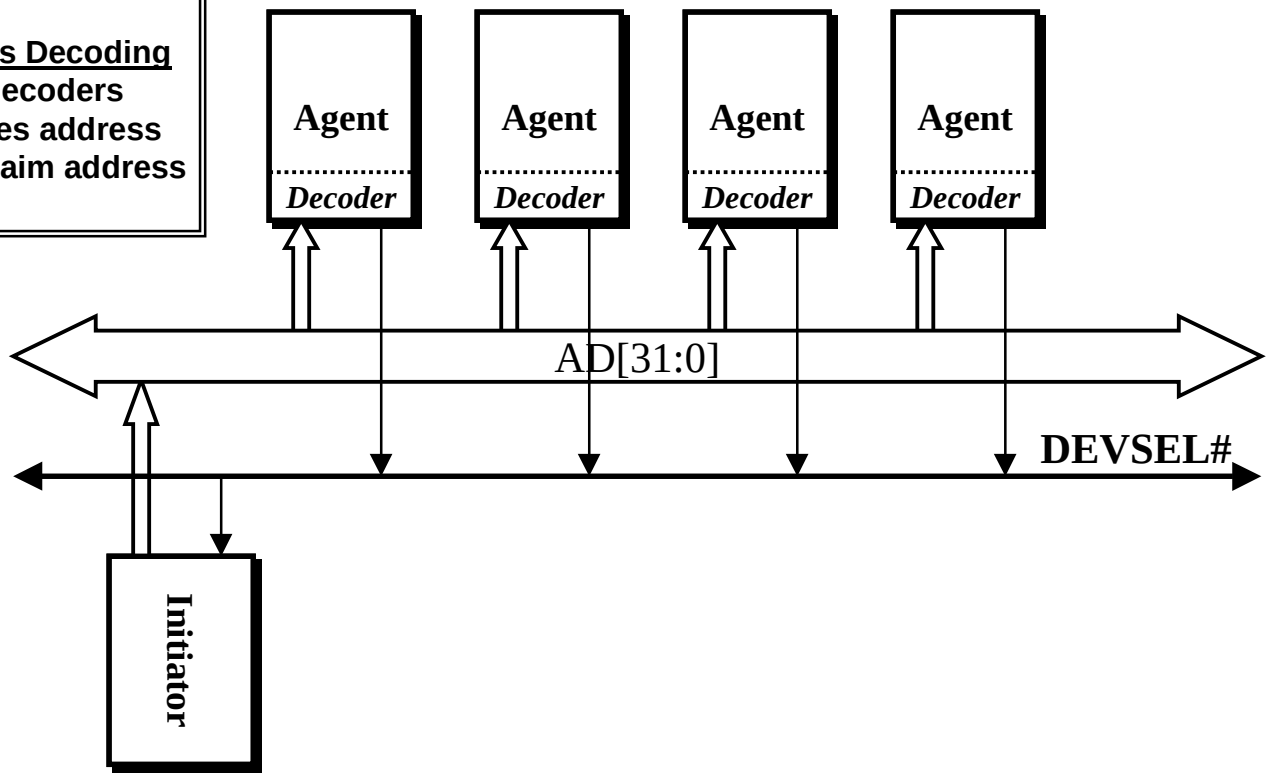
Target Address Decoding

- **PCI uses distributed address decoding**
  - *A transaction begins over the PCI bus*
  - *Each potential target on the bus decodes the transaction's PCI address to determine whether it belongs to that target's assigned address space*
    - One target may be assigned a larger address space than another, and would thus respond to more addresses
  - *The target that owns the PCI address then claims the transaction by asserting DEVSEL#*

© 2000 Xilinx, Inc.
All Rights Reserved

Distributed Address Decoding

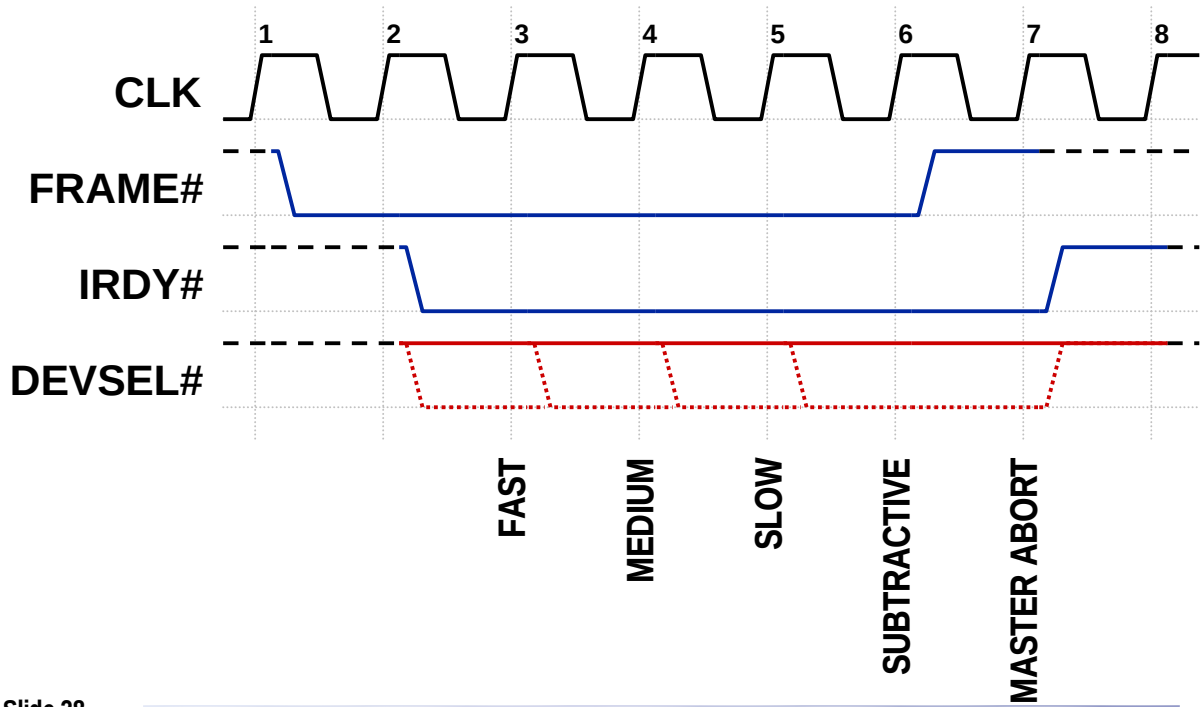
Distributed Address Decoding
Programmable decoders
Each agent decodes address
DEVSEL# used to claim address



© 2000 Xilinx, Inc.
All Rights Reserved

Target Decode

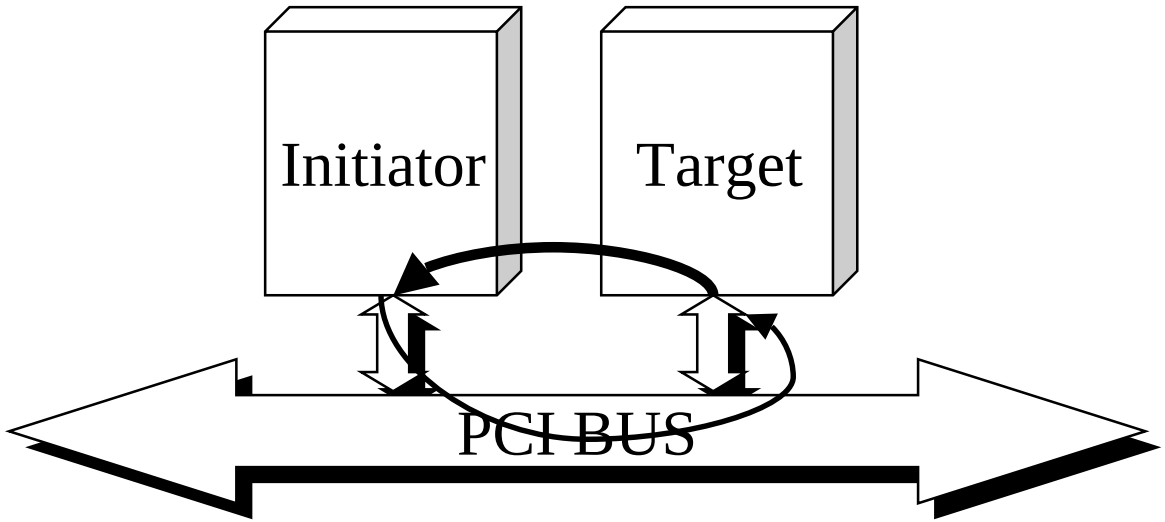
- Address decoders come in different speeds
- If a transaction goes unclaimed (nobody asserts DEVSEL#), “Master Abort” occurs



© 2000 Xilinx, Inc.
All Rights Reserved

Example #2 – Target Read

- A four-DWORD burst read from a target by an initiator



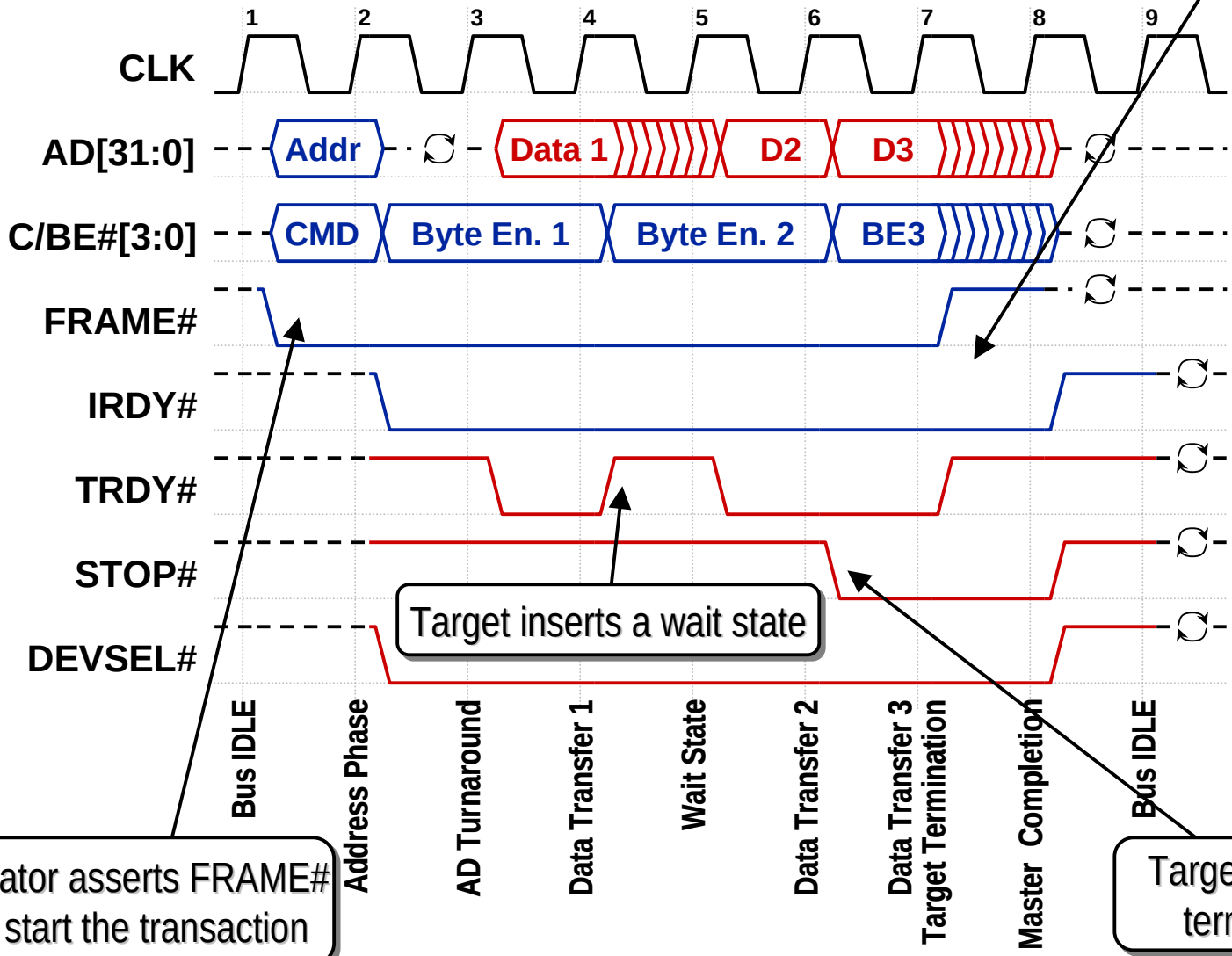
© 2000 Xilinx, Inc.
All Rights Reserved

More Terms

- **Turnaround cycle**
  - *"Dead" bus cycle to prevent bus contention*
- **Wait state**
  - *A bus cycle where it is possible to transfer data, but no data transfer occurs*
  - *Target deasserts TRDY# to signal it is not ready*
  - *Initiator deasserts IRDY# to signal it is not ready*
- **Target termination**
  - *Target asserts STOP# to indicate that it needs to terminate the current transaction*

© 2000 Xilinx, Inc.
All Rights Reserved

Target Read Transaction



© 2000 Xilinx, Inc.
All Rights Reserved

Target Read – Things to Note

- **Wait states may be inserted dynamically by the initiator or target by deasserting IRDY# or TRDY#**
- **Either agent may signal the end of a transaction**
  - *The target signals termination by asserting STOP#*
  - *The initiator signals completion by deasserting FRAME#*

© 2000 Xilinx, Inc.
All Rights Reserved

Zero and One Wait State

- **These terms are used in popular PCI parlance* to describe how an agent signals its *x*RDY# signal during each data phase**

  **Although the PCI spec uses the term "wait state," it does not use terms such as "zero-wait-state agent" and "one-wait-state device."*

- **A one-wait-state agent inserts a wait state at the beginning of each data phase**
  - *This is done if an agent – built in older, slower silicon – needs to pipeline critical paths internally*
  - *Reduces bandwidth by 50%*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Zero and One Wait State

- **The need to insert a wait state is typically an issue only when the agent is sourcing data (initiator write or target read)**
  - *This is because such an agent would have to sample its counterpart's xRDY# signal to see if that agent accepted data, then fan out to 36 or more clock enables (for AD[31:0] and possibly C/BE#[3:0]) to drive the next piece of data onto the PCI bus . . . all within 11 ns!*
    - And even that 11 ns would be eaten up by a chip's internal clock-distribution delay

© 2000 Xilinx, Inc.
All Rights Reserved

Types of Target Termination

- **Target Retry**
  "I'm not ready, try again later."
- **Target Disconnect with Data**
  "I couldn't eat another bite . . . OK, just one more."
- **Target Disconnect Without Data**
  "I couldn't eat another bite . . . and I'm not kidding!"
- **Target Abort**
  "Major snafu alert!"

© 2000 Xilinx, Inc.
All Rights Reserved

Xilinx PCI
The CORE of a GREAT IDEA

PCI Addressing and Bus Commands

Xilinx PCI

XILINX®

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Address Space

- **A PCI target can implement up to three different types of address spaces**
  - *Configuration space*
    - Stores basic information about the device
    - Allows the central resource or O/S to program a device with operational settings
  - *I/O space*
    - Used mainly with PC peripherals and not much else
  - *Memory space*
    - Used for just about everything else

© 2000 Xilinx, Inc.
All Rights Reserved

Types of PCI Address Space

- **Configuration space**
  - *Contains basic device information, e.g., vendor or class of device*
  - *Also permits Plug-N-Play*
    - Base address registers allow an agent to be mapped dynamically into memory or I/O space
    - A programmable interrupt-line setting allows a software driver to program a PC card with an IRQ upon power-up (without jumpers!)

© 2000 Xilinx, Inc.
All Rights Reserved

Types of PCI Address Space

- **Configuration space (cont'd)**
  - *Contains 256 bytes*
    - The first 64 bytes (00h – 3Fh) make up the standard configuration header, predefined by the PCI spec
    - The remaining 192 bytes (40h – FFh) represent user-definable configuration space
      - *This region may store, for example, information specific to a PC card for use by its accompanying software driver*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Types of PCI Address Space

- **I/O space**
  - *This space is where basic PC peripherals (keyboard, serial port, etc.) are mapped*
  - *The PCI spec allows an agent to request 4 bytes to 2GB of I/O space*
    - For *x*86 systems, the maximum is 256 bytes because of legacy ISA issues

© 2000 Xilinx, Inc.
All Rights Reserved

Types of PCI Address Space

- **Memory space**
  - *This space is used by most everything else – it's the general-purpose address space*
    - The PCI spec recommends that a device use memory space, even if it is a peripheral
  - *An agent can request between 16 bytes and 2GB of memory space*
    - The PCI spec recommends that an agent use at least 4kB of memory space, to reduce the width of the agent's address decoder

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Commands

- **PCI allows the use of up to 16 different 4-bit commands**
  - *Configuration commands*
  - *Memory commands*
  - *I/O commands*
  - *Special-purpose commands*
- **A command is presented on the C/BE# bus by the initiator during an address phase (a transaction's first assertion of FRAME#)**

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

PCI Commands

	C/BE#	Command
	0000	Interrupt Acknowledge
	0001	Special Cycle
	0010	I/O Read
	0011	I/O Write
	0100	Reserved
	0101	Reserved
	0110	Memory Read
	0111	Memory Write
	1000	Reserved
	1001	Reserved
With IDSEL	1010	Configuration Read
With IDSEL	1011	Configuration Write
	1100	Memory Read Multiple
	1101	Dual Address Cycle
	1110	Memory Read Line
	1111	Memory Write and Invalidate

Memory

I/O

Configuration

Special-Purpose

Reserved

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

PCI Configuration

© 2000 Xilinx, Inc.
All Rights Reserved

The Plug-and-Play Concept

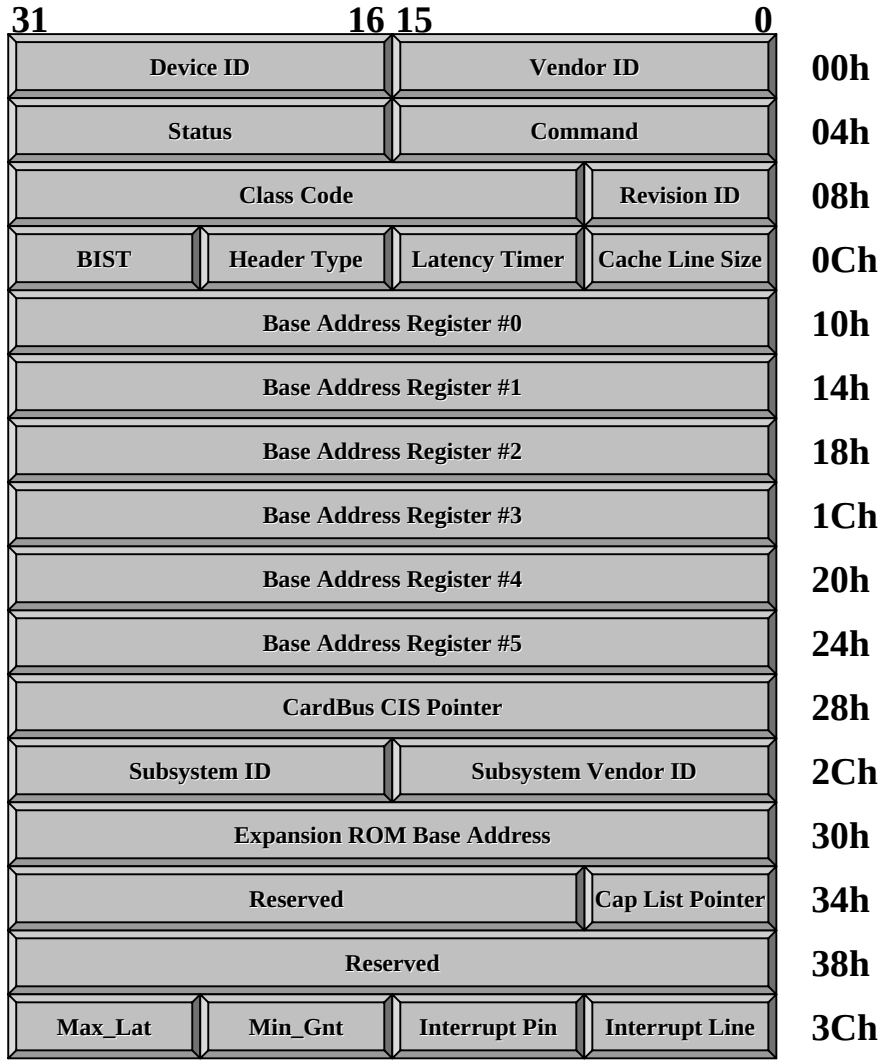
- **Plug-and-Play (PNP)**
  - *Allows add-in cards to be plugged into any slot without changing jumpers or switches*
    - Address mapping, IRQs, COM ports, etc., are assigned dynamically at system start-up
  - *For PNP to work, add-in cards must contain basic information for the BIOS and/or O/S, e.g.:*
    - Type of card and device
    - Memory-space requirements
    - Interrupt requirements

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

The Plug-and-Play Concept

- **To make PNP possible in PCI, each PCI device maintains a 256-byte configuration space**
  - *The first 64 bytes (shown here) are predefined in the PCI spec and contain standard information*
  - *The upper 192 bytes may be used to store device-specific information*



© 2000 Xilinx, Inc.
All Rights Reserved

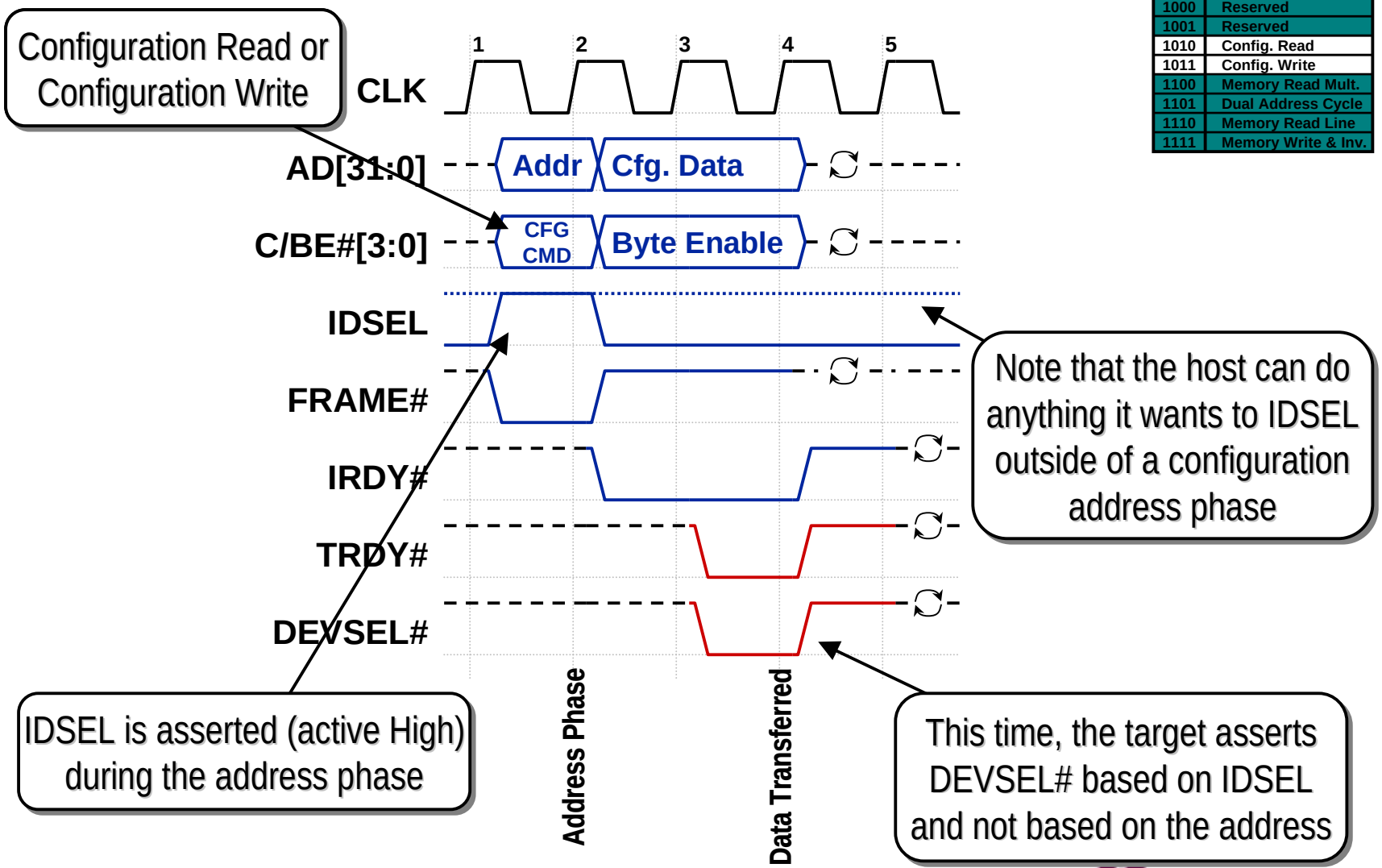
Configuration Transactions

- **Are generated by a host or PCI-to-PCI bridge**
- **Use a set of IDSEL signals as chip selects**
  - *Dedicated address decoding*
  - *Each agent is given a unique IDSEL signal*
- **Are typically single data phase**
  - *Bursting is allowed, but is very rarely used*
- **Two types (specified via AD[1:0] in addr. phase)**
  - *Type 0: Configures agents on same bus segment*
  - *Type 1: Configures across PCI-to-PCI bridges*

© 2000 Xilinx, Inc.
All Rights Reserved

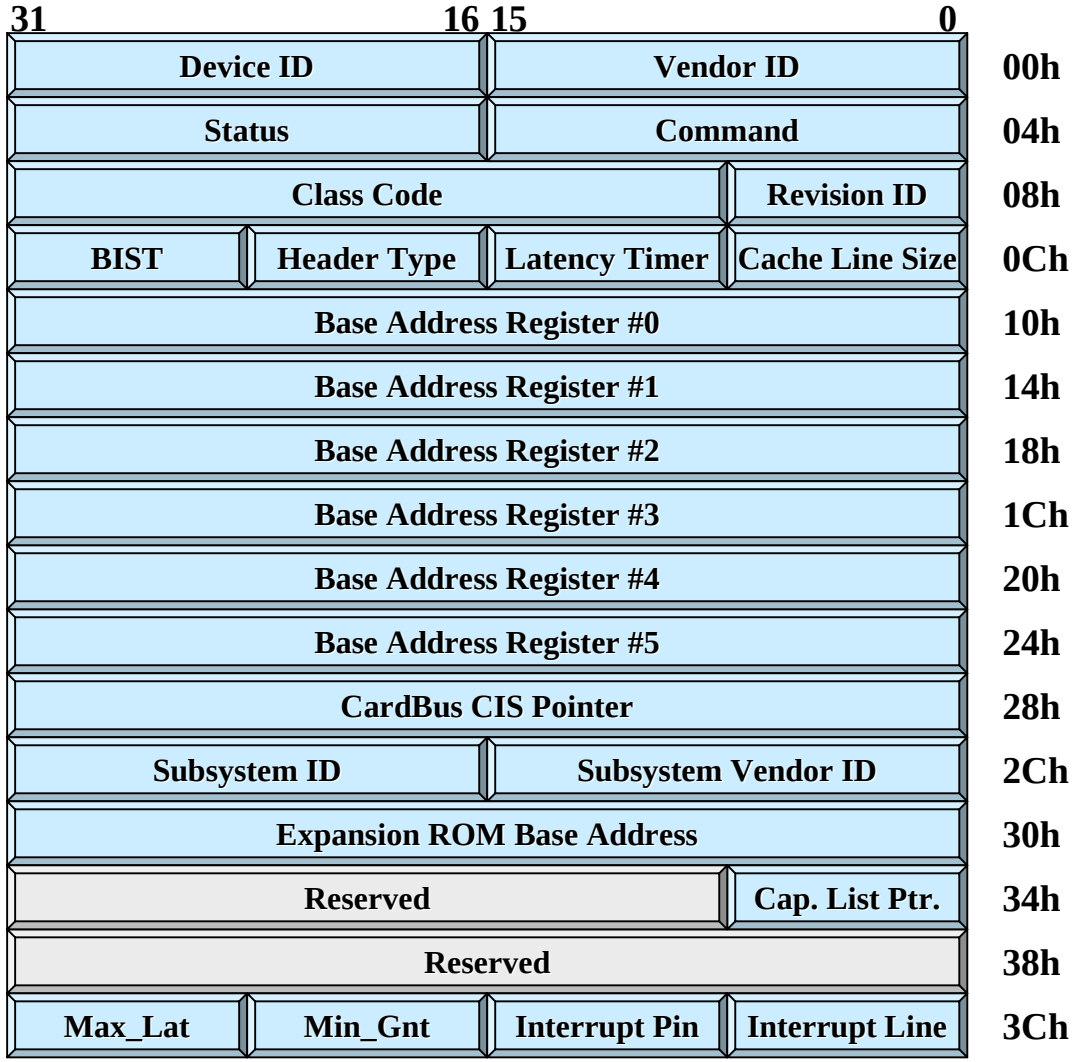
Configuration Example

C/BE#	Command
0000	Interrupt Ack.
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read
0111	Memory Write
1000	Reserved
1001	Reserved
1010	Config. Read
1011	Config. Write
1100	Memory Read Mult.
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write & Inv.



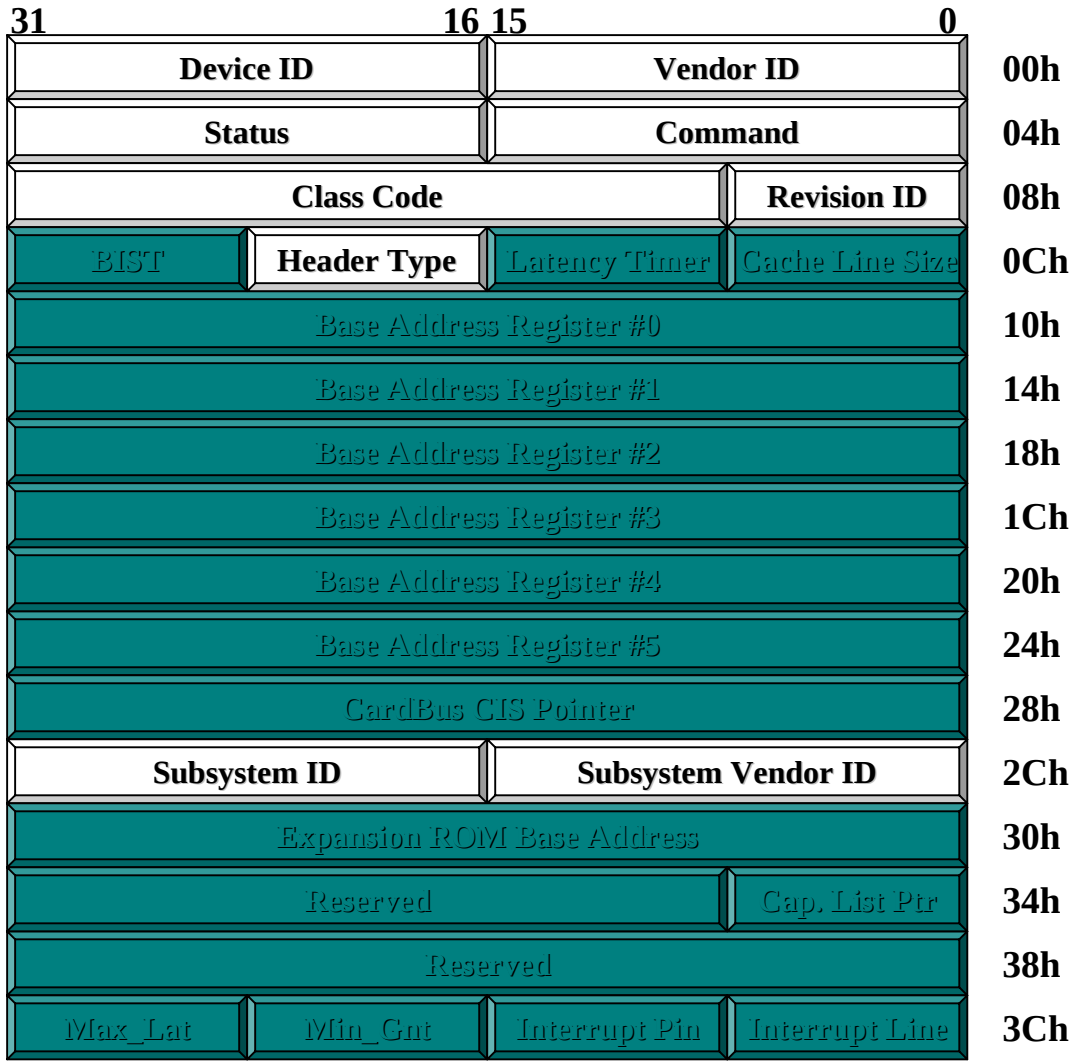
© 2000 Xilinx, Inc.
All Rights Reserved

Standard PCI Configuration Header



© 2000 Xilinx, Inc.
All Rights Reserved

Required by PCI Spec 2.2



© 2000 Xilinx, Inc.
All Rights Reserved

Xilinx PCI
The CORE of a GREAT IDEA

Electrical & Timing Specifications

Xilinx PCI

XILINX®

© 2000 Xilinx, Inc.
All Rights Reserved

Signaling Environments

- **The PCI spec describes two different electrical environments**
  - *5V signaling*
  - *3.3V signaling*
- **Technically, these names have *nothing* to do with the actual supply voltage**
  - *Rather, they are tied to logic-level thresholds*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

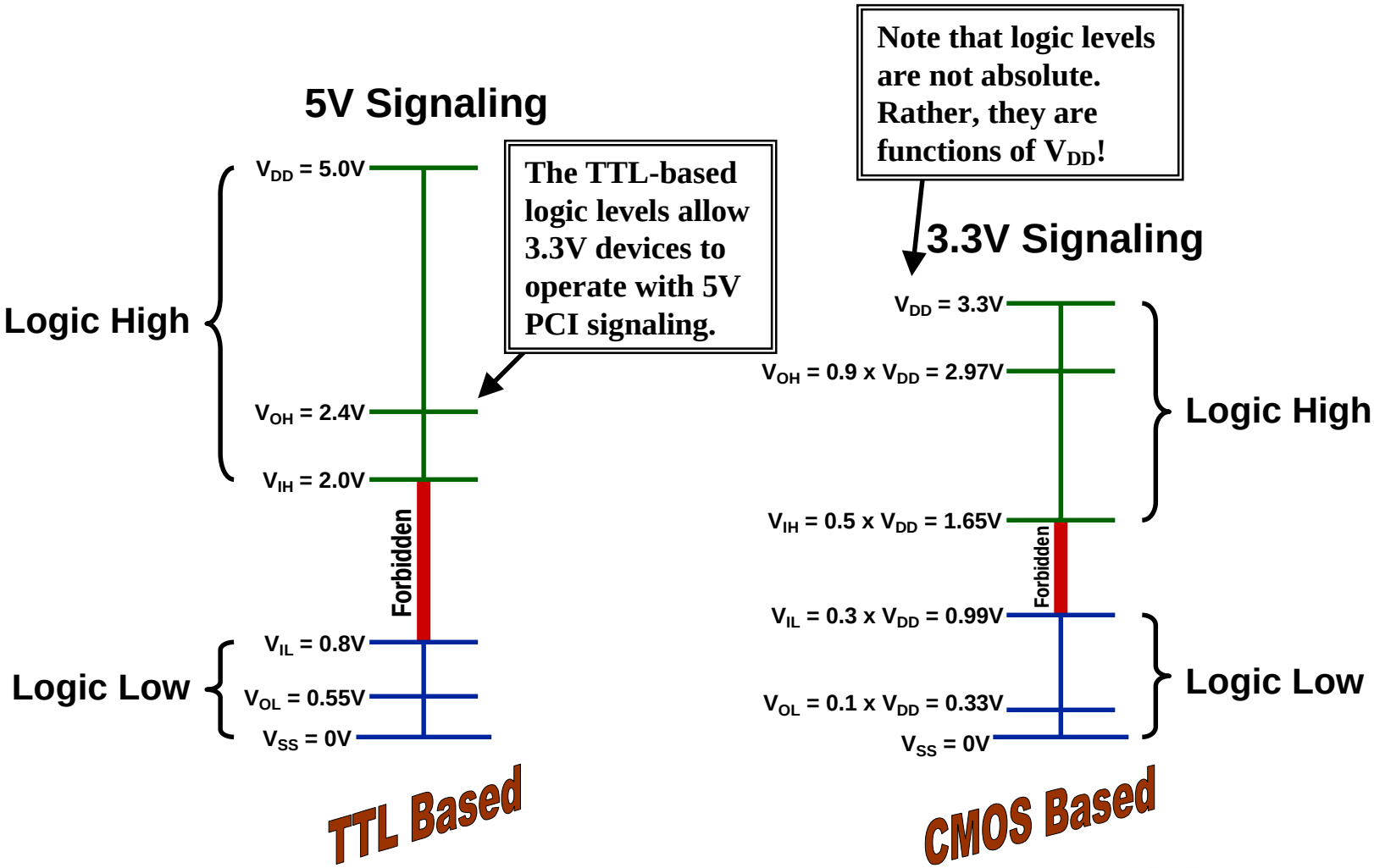
Signaling Environments

- **5V signaling is the most common**
- **66MHz PCI buses can only use 3.3V signaling**
  - *Note that 33MHz PCI can still use either*
- **Some plug-in cards can support both signaling environments – these are known as “universal” cards**

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

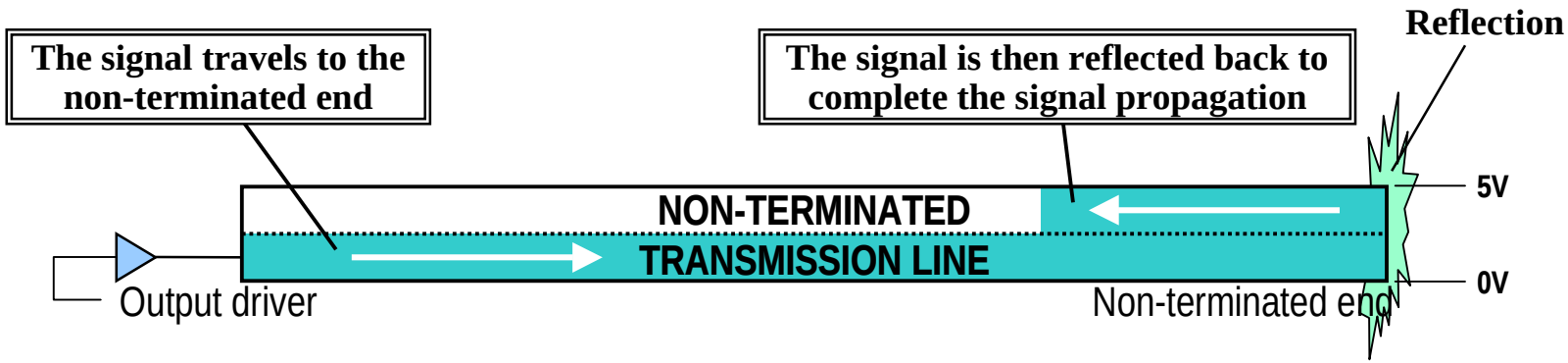
Signaling Environments



© 2000 Xilinx, Inc.
All Rights Reserved

Reflective-Wave Switching

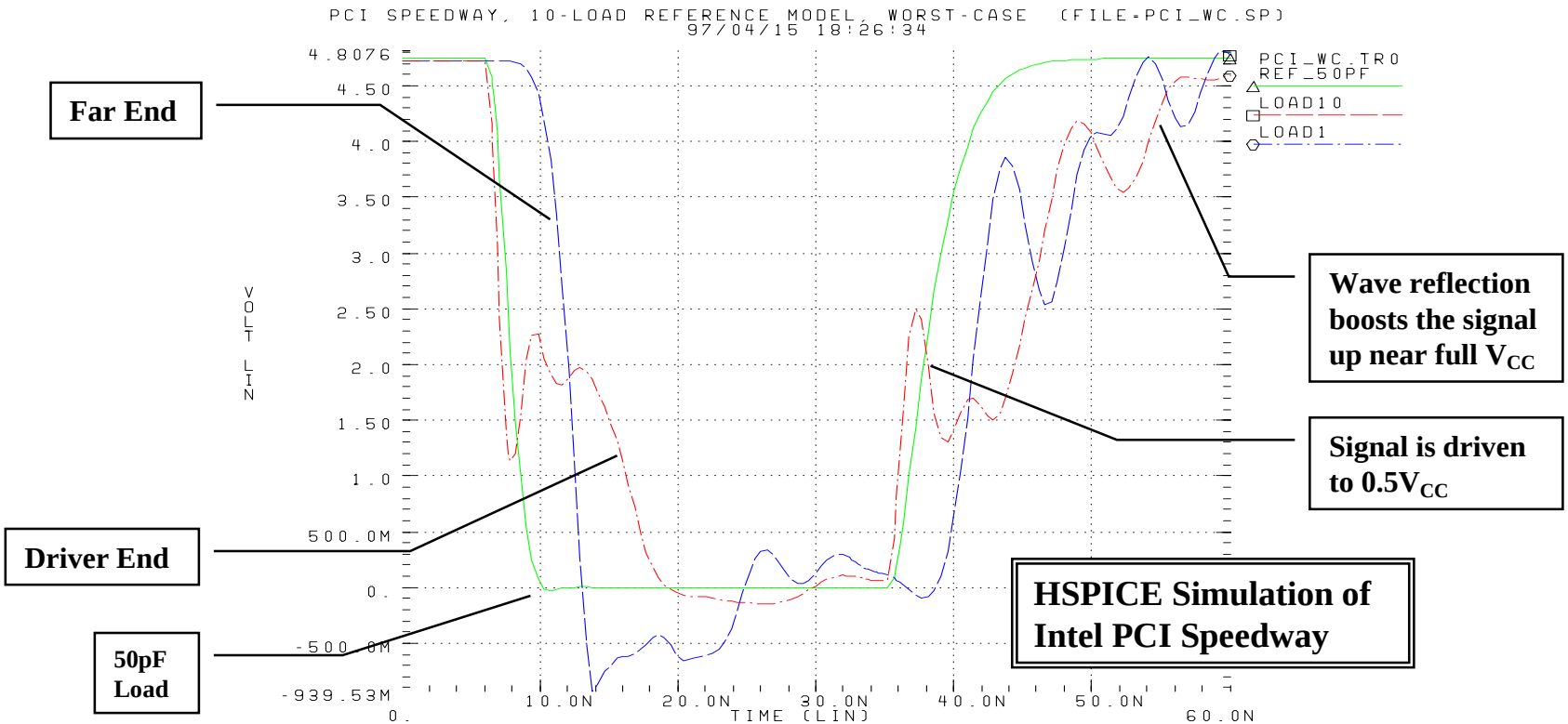
- **PCI uses “reflective waves”**
  - *Each wire on the PCI bus is a non-terminated transmission line, which causes signals to reflect over the length of the trace*
  - *Valid voltage levels are obtained after one reflection; this reduces the cost of PCI by not requiring high-powered output drivers*



© 2000 Xilinx, Inc.
All Rights Reserved

Reflective-Wave Switching

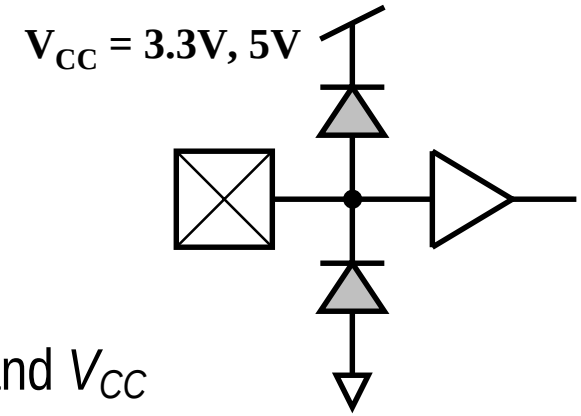
- This display shows how a reflection looks



© 2000 Xilinx, Inc.
All Rights Reserved

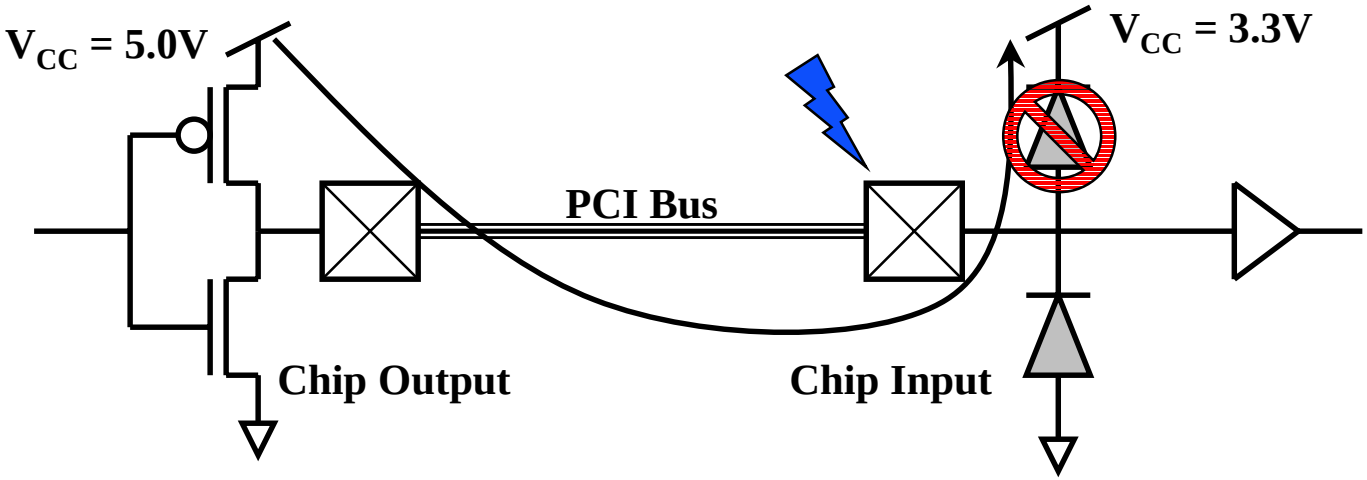
PCI – AC Specifications
Input Requirements – Clamp Diodes

- **Clamp diodes protect inputs from momentary short-circuit current caused by tri-stating delays**
- **5V signaling environment**
  - *Inputs must be clamped to ground*
  - *Upper clamp to 5V rail is optional*
- **3.3V signaling environment**
  - *Inputs must be clamped to both ground* and V_{CC}
  - *For universal cards (or any other 3.3V device that interfaces to 5V PCI), inputs must be clamped to the 5V supply, not the 3.3V supply*



© 2000 Xilinx, Inc.
All Rights Reserved

Bad Diode Clamping
Why Must Universal Cards Clamp to 5V?

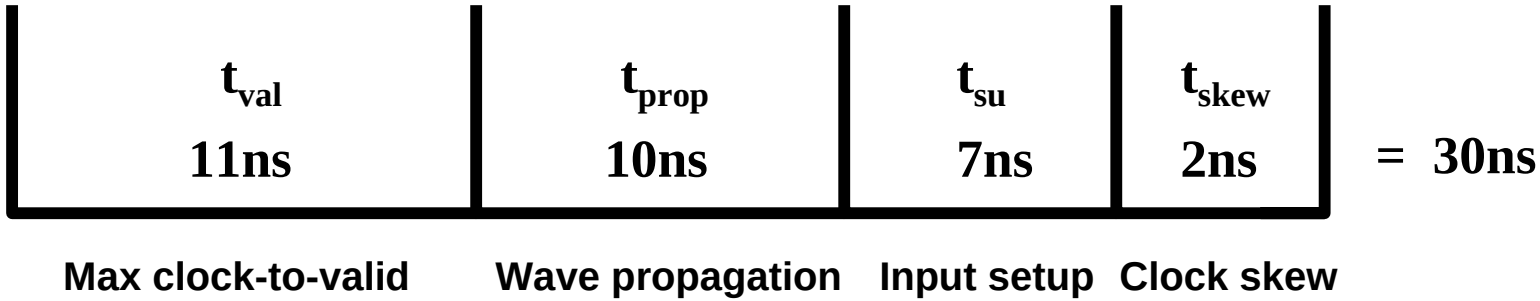


- A universal card clamped to 3.3V, that is plugged into a 5V signaling bus, will create a short-circuit connection between the motherboard's 5V and 3.3V supplies!

© 2000 Xilinx, Inc.
All Rights Reserved

33MHz PCI Timing Specification

30ns Bus Cycle Time



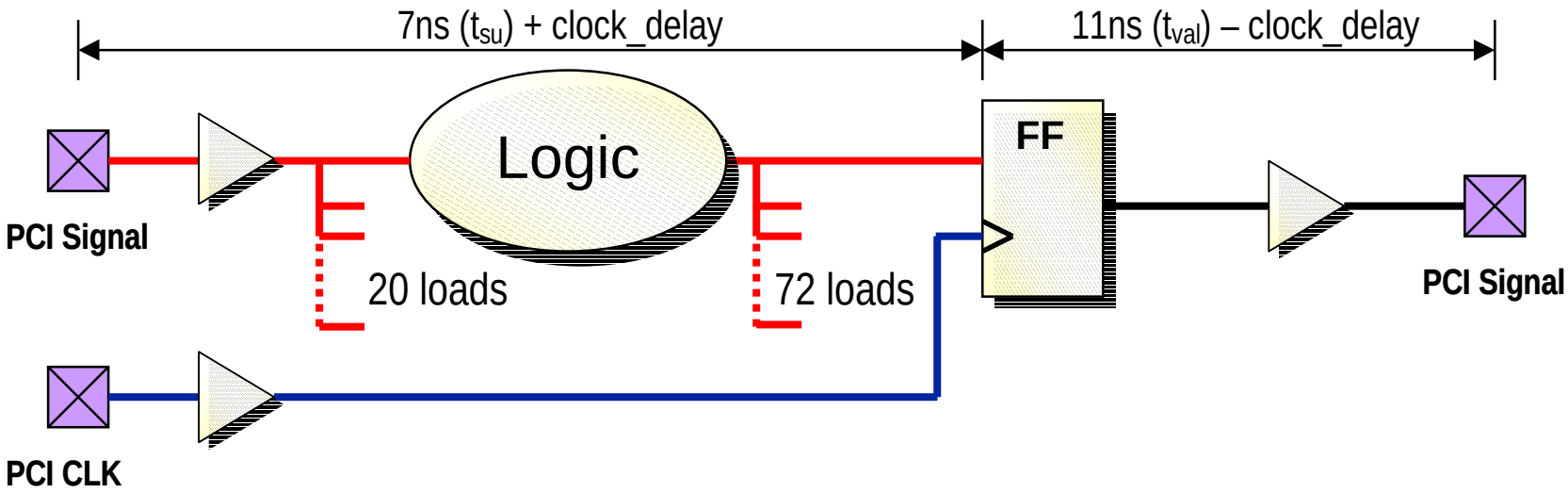
Other Requirements:

Hold time : 0ns
Min clock-to-out : 2ns
Output off time : 28ns

All timing parameters are measured at the package pin

© 2000 Xilinx, Inc.
All Rights Reserved

Why Is PCI Timing So Tough?



- **PCI handshaking performed every clock cycle**
  - *These signals* can't be pipelined
- **7ns setup + clock_delay → 100+ MHz! [33 MHz PCI]**
- **3ns setup + clock_delay → 220+ MHz! [66 MHz PCI]**

© 2000 Xilinx, Inc.
All Rights Reserved

Add-In Card Design

- **Trace length**
  - *All 32-bit PCI signals must be no more than 1.5″*
  - *All 64-bit ext. signals must be no more than 2.0″*
- **Clock trace must be exactly 2.5″ (± 0.1″)**
  - *Routed to only one load*
  - *Needed for clock-skew control*
- **PCI device requirements**
  - *One pin per signal!*
  - *Max input capacitance is 10pF (unless the device is on the motherboard, where 16pF is OK)*

© 2000 Xilinx, Inc.
All Rights Reserved

System Issues – Bus Loading

- **No PCI spec requirement as to the loading on the bus; however:**
  - *A driver must meet the 10ns propagation spec*
- **The rule of thumb is 10 loads max for 33MHz**
  - *Motherboard devices count as one load*
  - *Each add-in card slot counts as two load*
  - *Since most PC motherboards must have ≥2 PCI devices, they usually have no more than 4 slots*
- **More slots are available using PCI-to-PCI bridges or peer-to-peer PCI systems**

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

64-Bit and 66MHz PCI

Xilinx PCI

© 2000 Xilinx, Inc.
All Rights Reserved

The 64-Bit PCI Extension

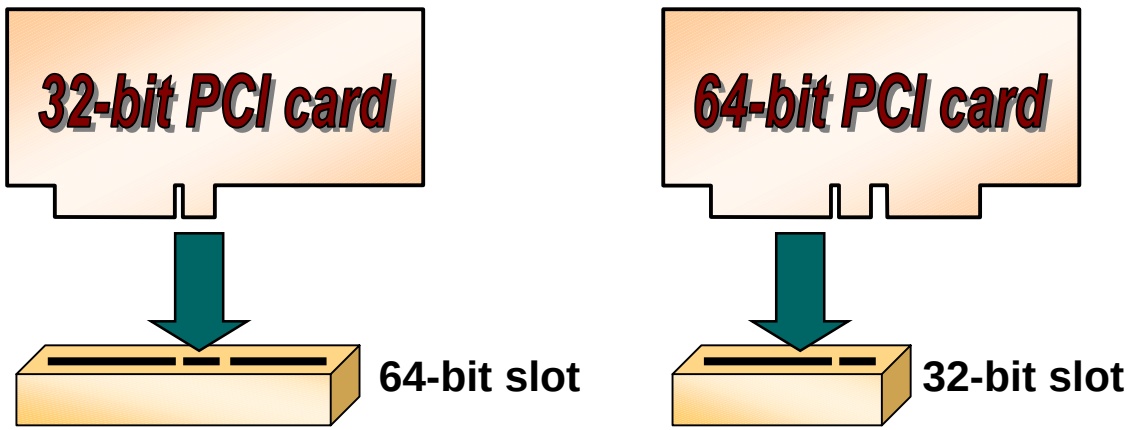
- Doubles the available PCI bandwidth (keeping the clock frequency at 33MHz) to 264 MB/sec
- 64-bit PCI can use both 5V and 3.3V signaling
- Only used for memory transactions

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

The 64-Bit PCI Extension

- **Mixing and matching is allowed**
  - *A 32-bit card can be plugged into a 64-bit slot*
  - *A 64-bit card can be plugged into a 32-bit slot*
  - *Use of the 32-bit vs. 64-bit datapath is negotiated between the initiator and target at the start of each transaction*



© 2000 Xilinx, Inc.
All Rights Reserved

Additional 64-Bit Pins

- **AD[63:32]**
- **C/BE#[7:4]**
  - *Used only for byte enables (not for PCI commands) except in the special case of Dual Address Cycle, discussed later*
- **PAR64**
  - *The XOR of AD[63:32], C/BE#[7:4], and PAR64 must equal zero (even parity)*

© 2000 Xilinx, Inc.
All Rights Reserved

Additional 64-Bit Pins

- **REQ64#**
  - *Mirrors FRAME# – indicates that the initiator requests a 64-bit transaction*
- **ACK64#**
  - *Mirrors DEVSEL# – indicates that the target is able to fulfill the transaction request using the 64-bit datapath*
  - *By not asserting ACK64#, the target tells the initiator that it is a 32-bit agent*

© 2000 Xilinx, Inc.
All Rights Reserved

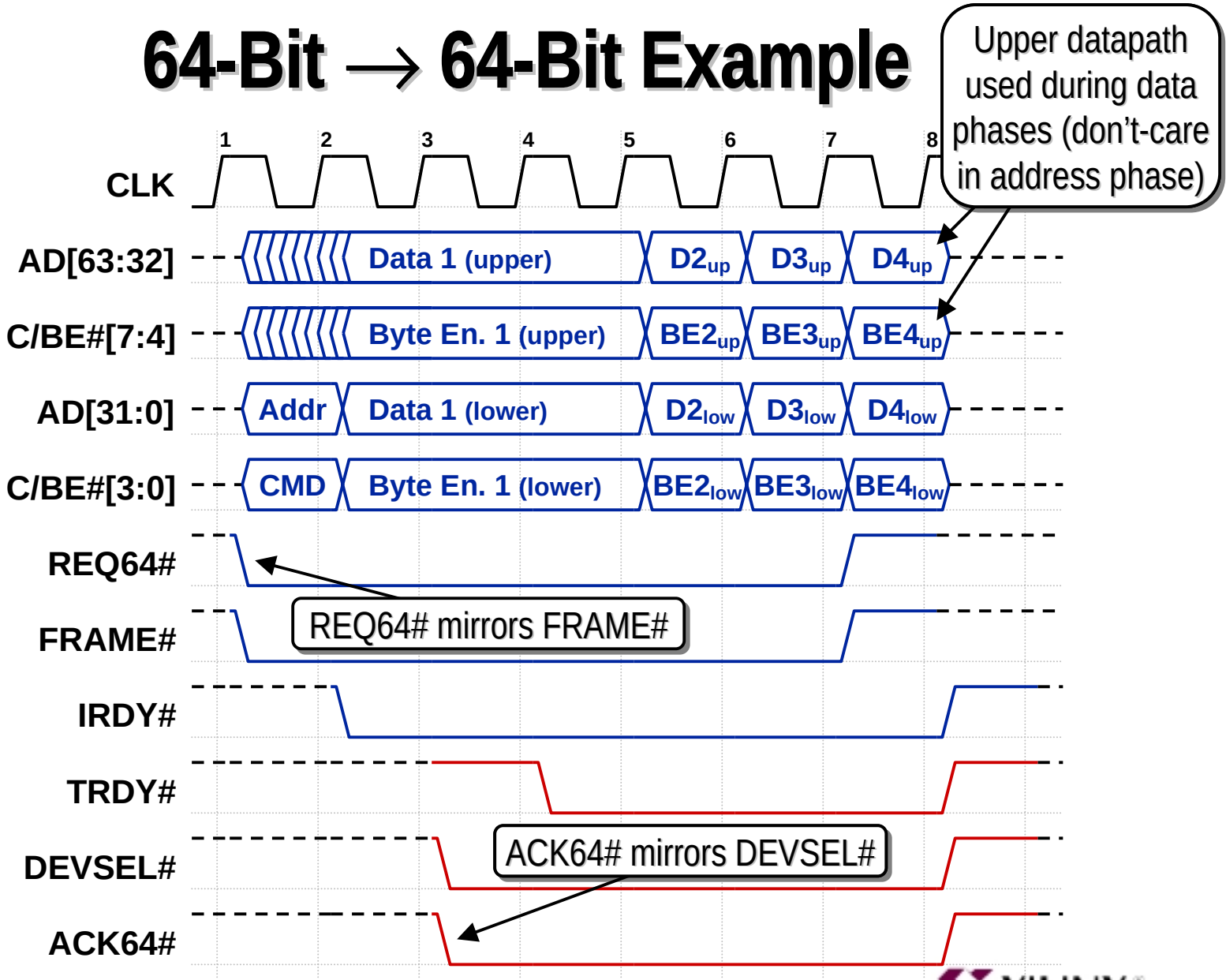
64-Bit Initiator to 64-Bit Target

- **Works the same as a standard 32-bit transaction, with the following additions:**
  - *The initiator asserts REQ64# to mirror FRAME#*
  - *The target, in response, asserts ACK64# to mirror DEVSEL#*
- **Data is transferred on AD[31:0] *and* AD[63:32]**
  - *C/BE#[7:4] and PAR64 are also used*
- **The starting address must be QUADWORD aligned (i.e., divisible by 8: AD[2] = 0)**

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

64-Bit → 64-Bit Example



© 2000 Xilinx, Inc.
All Rights Reserved

66MHz PCI Overview

- Pushes PCI bandwidth as high as 528MB/sec
- Most often used with the 64-bit extension, although it is legal to have 32-bit 66MHz PCI
- The signaling protocol is the same as with 33MHz PCI

© 2000 Xilinx, Inc.
All Rights Reserved

66MHz PCI Overview

- 66MHz PCI can only use 3.3V signaling
- A device that can operate at 66MHz has the 66MHz Capable bit set in the Status Register
- The loading allowance is cut in half (5 loads), so only one or two add-in slots are possible
- For open systems, 66MHz add-in cards must also work on a 33MHz PCI bus

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Variations

Xilinx PCI

© 2000 Xilinx, Inc.
All Rights Reserved

Overview of PCI Variations

- **As a well-defined standard, PCI and its various flavors have been widely adopted by many industries that require high-bandwidth data systems**
  - *Industrial computing*
  - *Datacom and telecom*
  - *Portable systems*
  - *Desktop systems*
- **Up-and-coming variations on PCI look to push its maximum bandwidth even higher**

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

PCI Variations
Same Protocol (Different Form Factor)

- **PMC**
  - *PCI in a mezzanine form factor*
- **CompactPCI**
  - *PCI in a Eurocard (VME-style) form factor*
  - *Used as a passive backplane*
  - *Used in physically rugged environments such as industrial and telecom systems*
- **Mini PCI**
  - *New proposed PCI standard for portable systems*

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Variations
Modified Protocol

- **CardBus**
  - *PCI in a PCMCIA form factor (portable systems)*
  - *Point-to-point, only slightly different protocol from standard PCI*
- **AGP (Advanced Graphics Port)**
  - *PCI-like point-to-point protocol*
  - *Primarily used for PC graphics cards*
- **PCI-X**
  - *New proposal to push bandwidth over 1GB/sec*
  - *Backward compatible with standard PCI*

© 2000 Xilinx, Inc.
All Rights Reserved

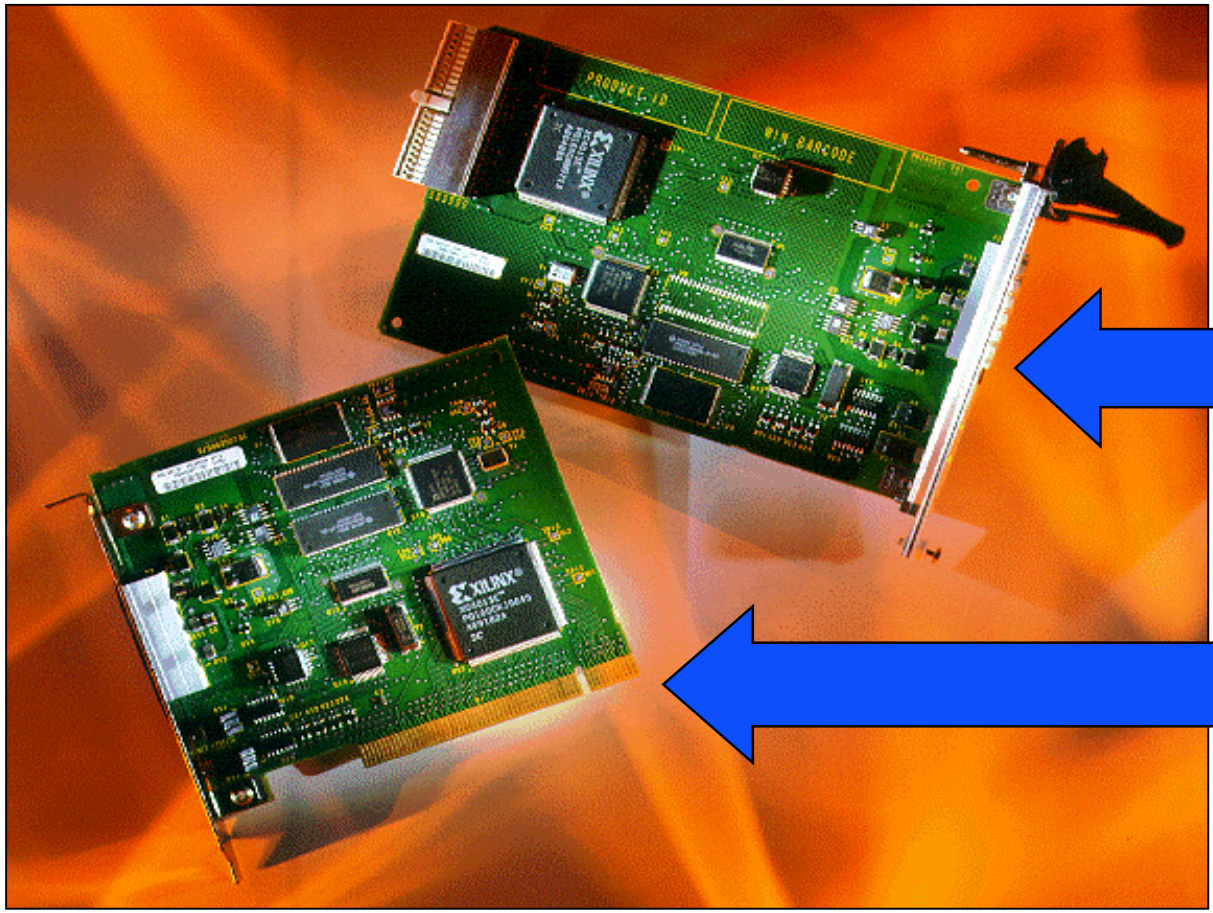
XILINX®

CompactPCI

- Used in Telecom and Industrial Applications that Require a Rugged Form Factor (Eurocard)
- IC Devices Maintain Specs from Standard PCI
- CompactPCI Imposes Additional Requirements for Add-in Cards as well as for the Host System
- Supports up to Eight Plug-in Slots
- Spec Maintained by the PCI Industrial Computer Manufacturers Group (PICMG), of which Xilinx is a Member

© 2000 Xilinx, Inc.
All Rights Reserved

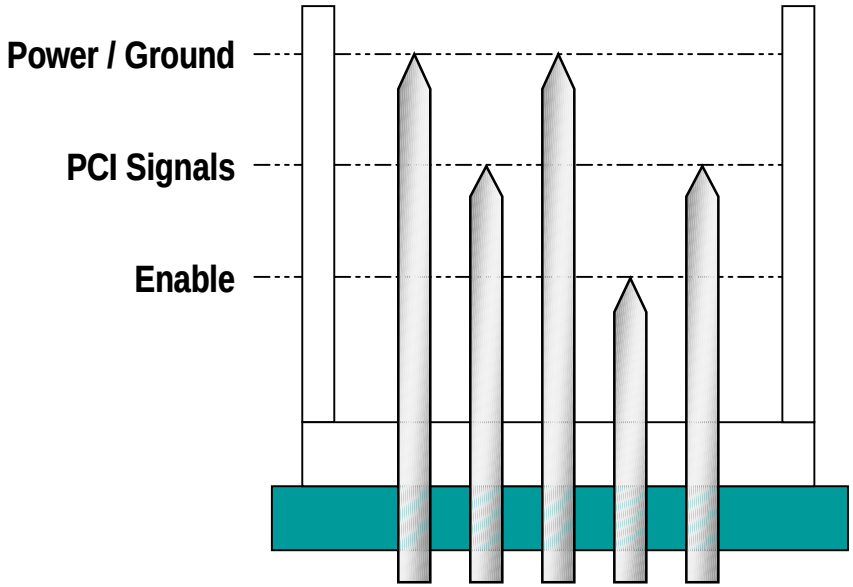
PCI / CompactPCI Comparison



© 2000 Xilinx, Inc.
All Rights Reserved

Hot-Swap CompactPCI

- **Live Insertion and Removal of Cards**
  - *High-availability systems (e.g., telecom)*
- **Electrical Issues**
  - *Signal lines need precharge to ~1V*
  - *Leakage current must be less than 10µA*
  - *Limited early power (2A maximum)*



Staged pins on backplane

- **Adds Hot-Swap Register to configuration space**
- **Not to be confused with Hot-Plug**

© 2000 Xilinx, Inc.
All Rights Reserved

Hot-Swap CompactPCI
Levels of Compliance

- **"Capable" – The Minimum Level of Compliance**
  - *Must be at least PCI v2.1 compliant*
  - *Must be able to tolerate precharge voltage (~1V)*
  - *Must be able to tolerate asynchronous RST#*
- **"Friendly" – The Intermediate Level of Compliance**
  - *Must at least be Hot-Swap "Capable"*
  - *Must also support the Capabilities List Pointer (PCI v2.2)*
- **"Ready" – The Highest Level of Compliance**
  - *Must at least be Hot-Swap "Friendly"*
  - *Must also support internal Bias Voltage and Early power*

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

The Xilinx PCI Solution

The PCI Challenge

1. Must Guarantee Timing

- *Setup, Max, and Min*

2. Must Guarantee the Behavior

- *Now and in the future*

3. Must Meet Performance Requirements

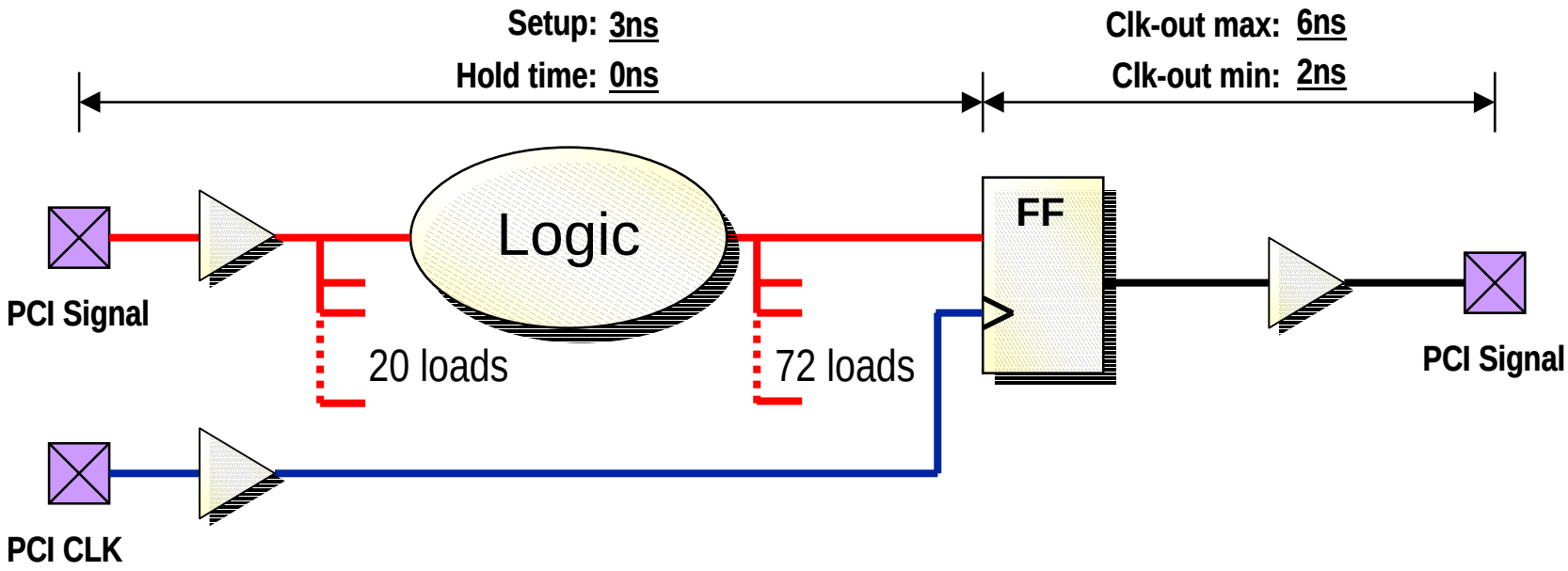
- *Sustained throughput at 66MHz*

4. Must Meet Cost Target

- *Single-chip solution*

© 2000 Xilinx, Inc.
All Rights Reserved

The PCI Challenge
The Critical Path



- The critical path is equivalent to **220+ MHz!**

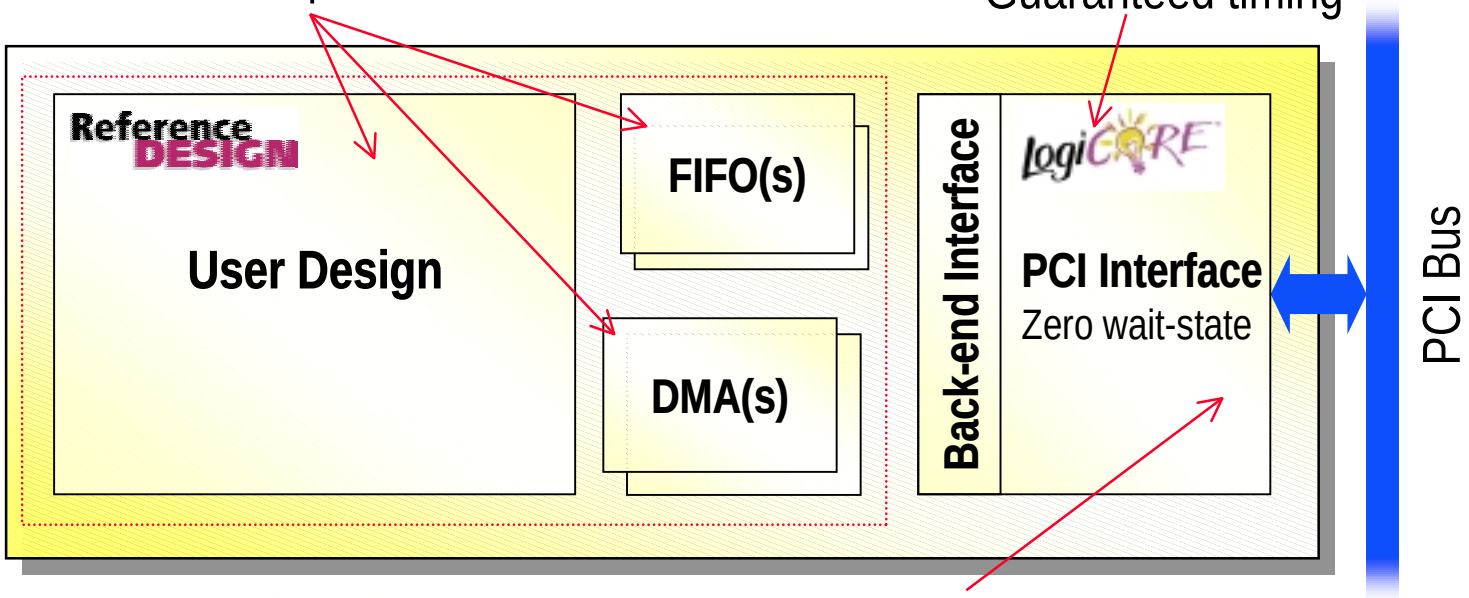
© 2000 Xilinx, Inc.
All Rights Reserved

Xilinx PCI Solution

The Real-PCI™

Real Flexibility
- Uses standard FPGAs
- Back-end de-coupled from core

Real Compliance
- PCI Initiator and Target
- Guaranteed timing



SPARTAN-III FPGA

Real Performance
- Zero wait-state at 0-33MHz
- Full 32 or 64-bit data path

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Why The Real-PCI From Xilinx?

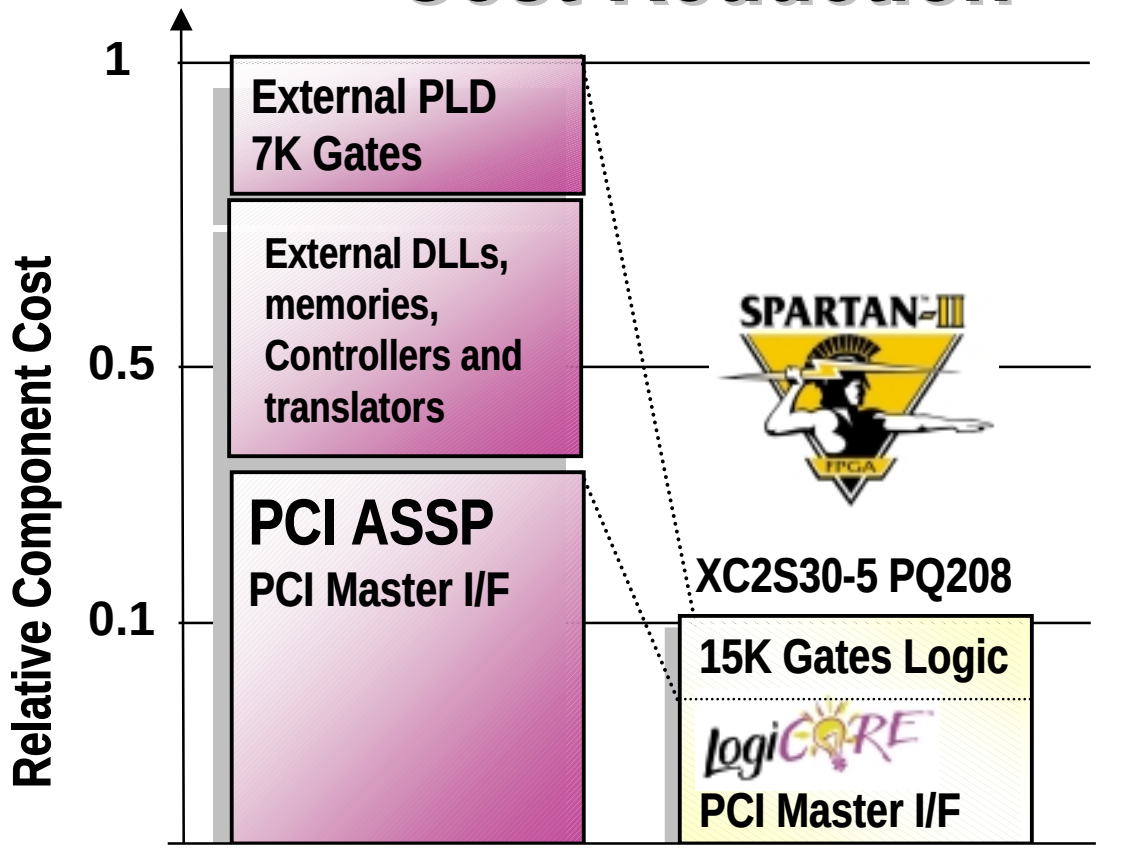
"The Xilinx PCI Core has the most flexibility and complete feature set of any vendor offering a programmable PCI core that we extensively evaluated. We were most impressed with the silicon features, as well as a smooth migration path across various cores and device densities"

"If our system performance requirements change, we have every confidence that we can import a new core to the existing design with minimal engineering effort".

Joel Van Doren, Senior design engineer
Xerox Engineering Systems

© 2000 Xilinx, Inc.
All Rights Reserved

Cost-Reduction

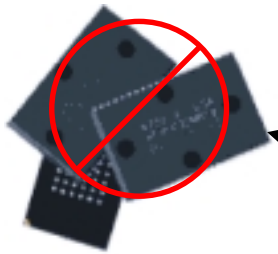


The Real-PCI™ Solution <$6

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

ASSP Replacement & Integration



System & Memory controllers, DLLs, Level translators ($20)

Standard Chip PCI Master I/F ($15)

External PLD
PCI Master I/F ($5)

The Real-PCI™

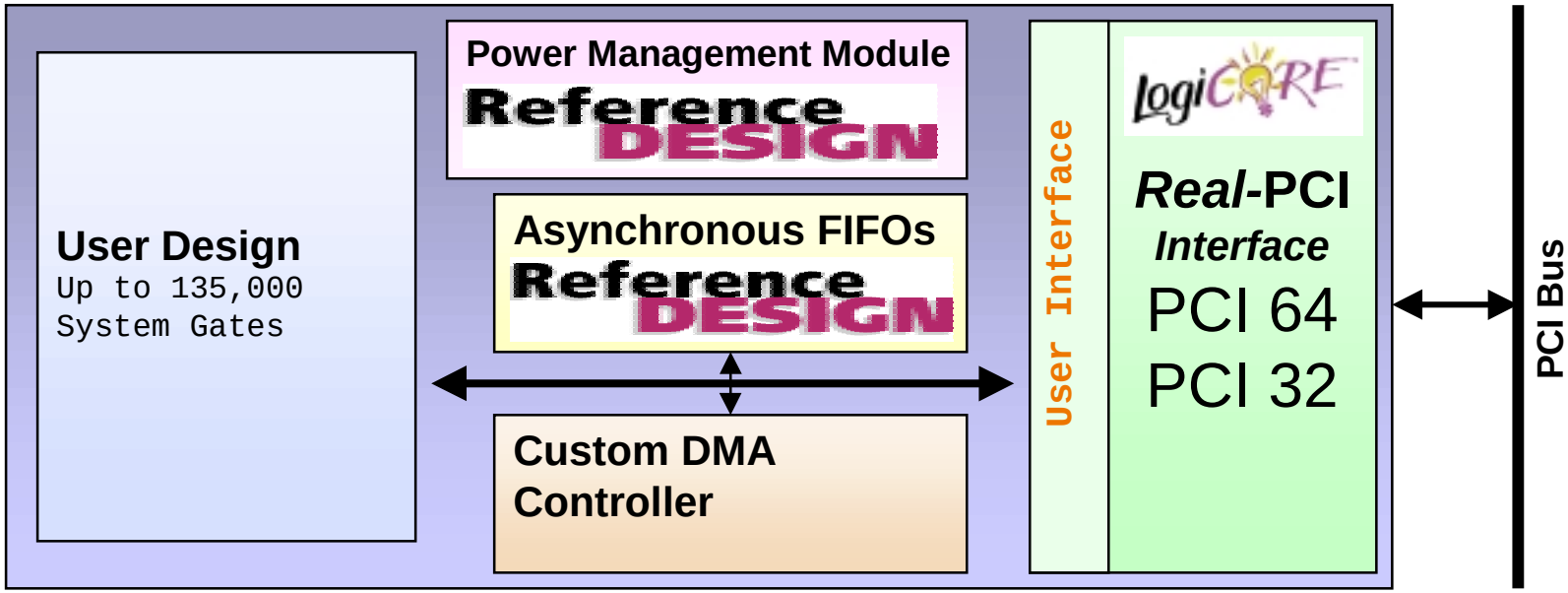
*Supported Devices
XC2S30 XC2S50
XC2S100 XC2S150

Memory ($9)

© 2000 Xilinx, Inc.
All Rights Reserved

Supporting Reference Designs

- Asynchronous FIFOs & DMA Controller
- Power Management Module



© 2000 Xilinx, Inc.
All Rights Reserved

Extra Support from PCI Experts

- **Provide Worldwide Access to Certified PCI Experts that allow:**
  - *Support for targeting additional devices*
  - *Complete turnkey integration*

COMIT™ SYSTEMS

Memec Design Services

MULTI VIDEO DESIGN

NALLATECH LIMITED

© 2000 Xilinx, Inc.
All Rights Reserved

Ordering Information

	PCI64 Design Kit DO-DI-PCI64-DK	**PCI64** DO-DI-PCI64	**PCI32 Design Kit** DO-DI-PCI32-DK	**PCI32 Spartan** DO-DI-PCI32-S
Design File Access				
64-bit 66 MHz	Virtex-E Virtex	Virtex-E Virtex	-	-
64-bit 33 MHz	Virtex-E Virtex Spartan-II	Virtex-E Virtex Spartan-II	-	-
32-bit 33 MHz	Virtex-E Virtex Spartan-II Spartan-XL Spartan 4000XLA	Virtex-E Virtex Spartan-II Spartan-XL Spartan 4000XLA	Virtex-E Virtex Spartan-II Spartan-XL Spartan 4000XLA	Spartan-II Spartan-XL Spartan
Development Tools				
Prototyping board	Nallatech BallyLINX	-	VCC HotPCI	-
SW Driver Development Tools	NuMega SoftICE Driver Suite	-	NuMega SoftICE Driver Suite	-
Miscellaneous				
Reference Designs	PCI Bridge Designs Power Management	PCI Bridge Designs Power Management	PCI Bridge Designs Power Management	PCI Bridge Designs Power Management
Design Examples	Reference Drivers Board Gerber files)	-	Reference Drivers	-
PCI System Architecture Text Book	PCI System Arch Text Book Design Guide	Online	PCI System Arch Text Book Design Guide	Online

© 2000 Xilinx, Inc.
All Rights Reserved

The Xilinx PCI Design Flow

- The PCI Lounge
- Core Configuration and Download
- Simulation and Synthesis
- Implementation – Xilinx Alliance/Foundation

© 2000 Xilinx, Inc.
All Rights Reserved

The PCI Lounge

- **A Special Area for Xilinx PCI Customers on WebLINX**
  - *PCI core configuration and download*
  - *Latest design data – no CD required!*
- **To Gain Access, Customers Submit an Online Registration form**
  - *The customer selects a unique username and password*
  - *The customer also gives his or her product serial number as proof of purchase*

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Lounge Registration

- **Register your LogiCORE product at:**

  http://www.xilinx.com/products/logicore/lounge/lounge.htm

- **During the registration process, you will be asked for:**
  - *Your LogiCORE product's serial number*
  - *Xilinx User ID and password (the same as your SmartAgent ID)*
    - You can create your own User ID during the registration process, if necessary
  - *The type of LogiCORE PCI Interface you purchased*

© 2000 Xilinx, Inc.
All Rights Reserved

PCI Lounge Registration

- In less than 2 business days, you will be sent a confirmation that you have access

© 2000 Xilinx, Inc.
All Rights Reserved

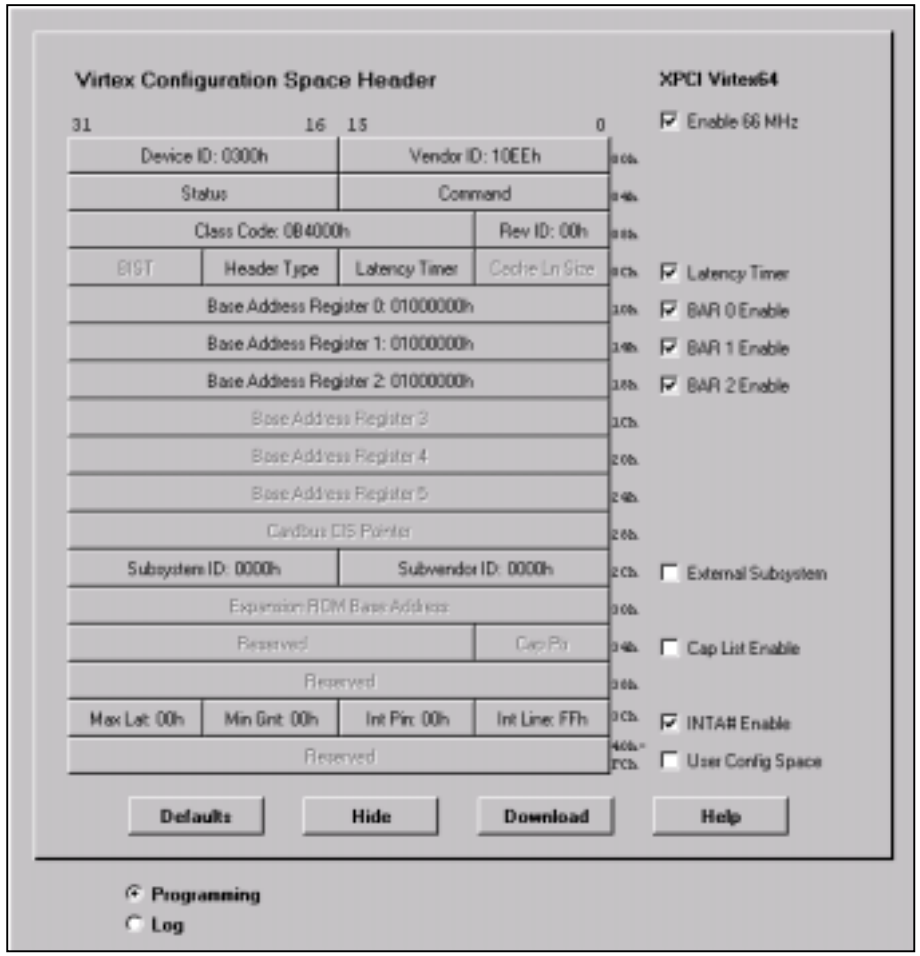
Downloading from the Internet

- **Once you are registered for the PCI Lounge, you can download your LogiCORE Interface in *two easy steps*:**
  1. *Customize your LogiCORE PCI Interface on WebLINX*
  2. *Download your tailor-made core*

© 2000 Xilinx, Inc.
All Rights Reserved

Downloading from the Internet

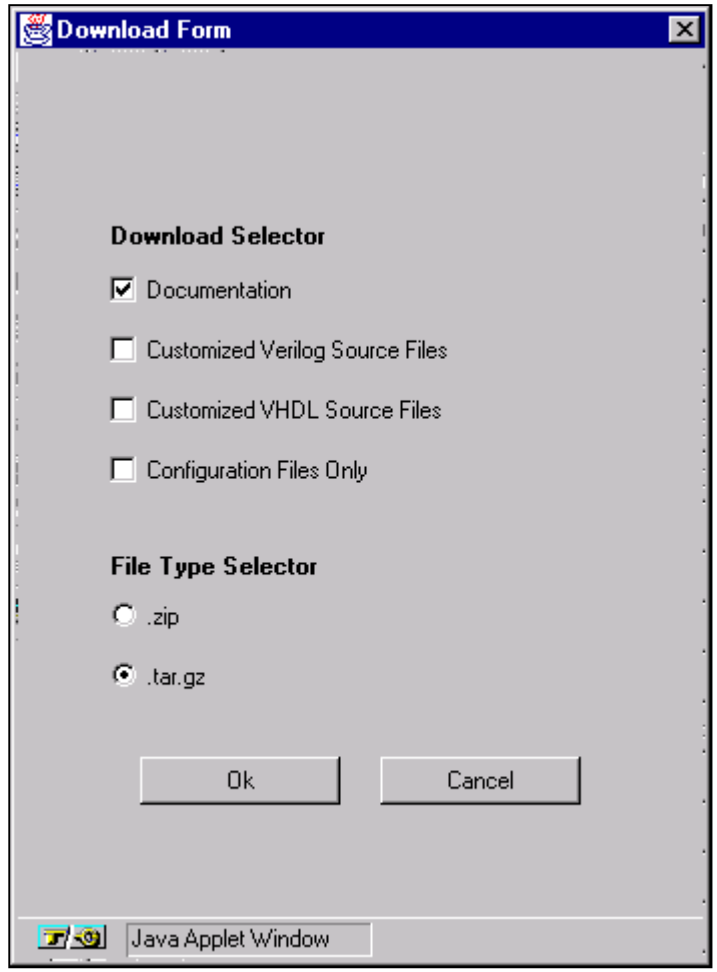
- **A Java-based GUI allows you to customize configuration options easily**
  - *See Chapter 3 of the* LogiCORE PCI User's Guide *for more detailed information*



© 2000 Xilinx, Inc.
All Rights Reserved

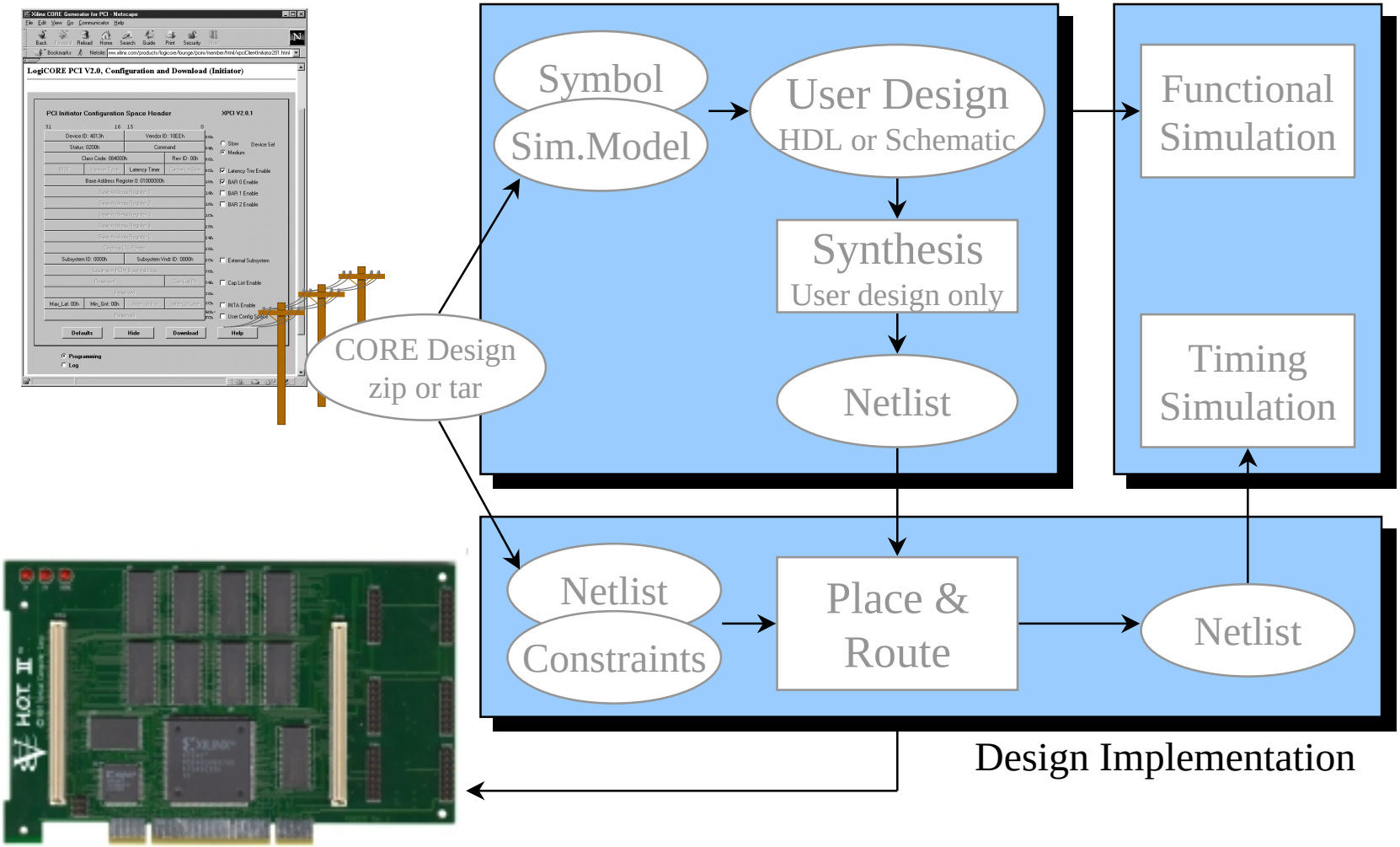
Downloading from the Internet

- **When it comes time to download, you can tailor the downloaded fileset for your design environment, based on:**
  - *Device and package*
  - *Design-entry tool*
  - *Archiving method*
    - .zip for PC
    - .tar.gz for Unix



© 2000 Xilinx, Inc.
All Rights Reserved

Xilinx PCI Design Flow Overview



© 2000 Xilinx, Inc.
All Rights Reserved

Simulation and Synthesis

- Functional and Timing simulation with -

- Synthesis with -

© 2000 Xilinx, Inc.
All Rights Reserved

XILINX®

Xilinx Implementation

- **Supported by Standard Xilinx Alliance and Foundation Series, V2.1i software**

© 2000 Xilinx, Inc.
All Rights Reserved

Available PCI Resources

- **LogiCORE PCI Lounges on WebLINX**
  - *www.xilinx.com/pci*
- **Xilinx Hotline**
  - *Expert PCI support engineers*
- **WWW based solutions and PCI Expert Journal**
  - *support.xilinx.com*
- **PCI-SIG discussion group**
  - *Mailing list subscription information on www.pcisig.com/forum.html*

© 2000 Xilinx, Inc.
All Rights Reserved