Introducing...

QuickLogic’s
The Basics of PCI

QuickPCI - The PCI Solution for System Needs
Typical System Implementation

- CPU
- Cache Memory Subsystem
- DRAM Subsystem
- PCI Bridge/Memory Controller
- PCI Add-in Slots
- ISA Bus Bridge
- ISA Add-in Slots
- ISA Bus
- Host Bus
- PCI Local Bus
Desktop Platforms

- Processor
- Bridge/Memory Controller
- Cache
- DRAM
- Graphics
- IDE
- Exp Bus Xface
- ISA/EISA - MicroChannel®
- ISA/EISA - MicroChannel®
- LAN, SCSI, Video, etc
- Fax Modem Add-in
- PCI Local Bus

PCI

3
Server Platforms

- Processor/Cache
- Processor/Cache
- Memory Controller
- Host Bridge
- System Bus
- PCI Local Bus
- Expansion Bus Bridge
- Expansion Bus Bridge
- Bridge I/O
- 32/64-Bit PCI Local Bus
- SCSI
- SCSI
- LAN
- LAN
- PCI to PCI Bridge
- PCI Local Bus
- PCI
- Conn 1
- Conn 4
Mobile Platforms
PCI Auto-Configuration

PCI Add-in Card + PCI Based Mother Board → Auto Configuration System
PCI Overview
PCI Features

- 32-Bit or 64-Bit address and data
- 66 or 33 down to 0 MHz synchronous operation
- Single or multiple bus masters
- Reflected bus signaling
- Stepped signaling
- Bus parity error reporting
- 5 or 3.3 volt operation
- Cache support
- JTAG testing
PCI Bus

- Bus Signals
- Bus Commands
- Bus Transactions
- Arbitration
**PCI Bus Signals**

**Required Pins**
- Address & Data
  - AD[31::00]
  - C/BE#[3::0]
  - PAR
  - FRAME#
  - TRDY#
  - IRDY#
  - STOP#
  - DEVSEL#
  - IDSEL
- Interface Control
  - REQ#
  - GNT#
  - PERR#
  - SERR#
  - CLK
  - RST
  - Present 1-2

**Optional Pins**
- Address & Data
  - AD[63::32]
  - C/BE#[7::4]
- Interface Control
  - PAR64
  - REQ64#
  - ACK64#
  - M66EN
  - LOCK#
  - Interrupt A-D#
  - Clkrun#
- Cache Support
  - SBO#
  - SDONE
  - TDI
  - TDO
  - TCK
  - TMS
  - TRST#

**PCI Pin List**

**PCI**
PCI Bus Control Signals

- **System**
  - CLK
  - RST#
- **Address Data**
  - AD[31:00]
  - C/BE[3:0]#
  - PAR
PCI Bus Control Signals

- **Interface Control**
  - FRAME# - LOCK#
  - IRDY# - IDSEL#
  - TRDY# - DEVSEL#
  - STOP#

- **Arbitration**
  - REQ#
  - GNT#
PCI bus access

- PCI is a Multimaster Bus
- All transactions *initiated* by a master
- All transactions to/from a *target*
PCI Bus Control Signals

- **FRAME#**
  - driven by master to indicate transfer start and end

- **IRDY#**
  - driven by master to indicate it is ready to transfer data

- **TRDY#**
  - driven by master to indicate it is ready to transfer data
Bus transaction start

![Bus transaction diagram]
## PCI Command Definition

<table>
<thead>
<tr>
<th>C/BE [3::0]#</th>
<th>Command Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Interrupt acknowledge</td>
</tr>
<tr>
<td>0001</td>
<td>Special Cycle</td>
</tr>
<tr>
<td>0010</td>
<td>I/O Read</td>
</tr>
<tr>
<td>0011</td>
<td>I/O Write</td>
</tr>
<tr>
<td>0110</td>
<td>Memory Read</td>
</tr>
<tr>
<td>0111</td>
<td>Memory Write</td>
</tr>
<tr>
<td>1010</td>
<td>Configuration Read</td>
</tr>
<tr>
<td>1011</td>
<td>Configuration Write</td>
</tr>
<tr>
<td>1100</td>
<td>Memory Read Multiple</td>
</tr>
<tr>
<td>1101</td>
<td>Dual Address Cycle</td>
</tr>
<tr>
<td>1110</td>
<td>Memory Read Line</td>
</tr>
<tr>
<td>1111</td>
<td>Memory Write and Invalidate</td>
</tr>
</tbody>
</table>
PCI Commands

The PCI bus commands, encoded on the C/BE[3::0]# lines, are summarized below:

– Interrupt Acknowledge (0000)
  • Performs a read implicitly addressed to the system interrupt controller
  • Address bits are logical don’t cares during the address phase
  • Byte enables indicate the size of the vector to be returned

– Special Cycle (0001)
  • Provides a simple message broadcast mechanism on the PCI bus
  • Logical equivalent of a wire to all agents
  • Alternative to physical signals during sideband communications
PCI Commands (Cont.)

- I/O Read (0010)
  - Reads data from an agent mapped in I/O address space
  - Byte enables indicate the size of the transfer and must be consistent with the byte address
- I/O Write (0011)
  - Writes data to an agent mapped in I/O address space
  - Byte enables indicate the size of the transfer and must be consistent with the byte address
- Reserved (0100, 0101, 1000, 1001)
  - Reserved for future use
- Memory Read (0110)
  - Reads data from an agent mapped in memory address space
  - Target can do anticipatory read if no side effects can be guaranteed (Prefetchable Memory)
  - Target must ensure coherency of data in temporary buffers after this transaction
PCI Commands (Cont.)

- **Memory Write (0111)**
  - Writes data to an agent mapped in memory address space
  - Target assumes responsibility for coherency of data
- **Configuration Read (1010)**
  - Reads the configuration space of each agent
- **Configuration Write (1011)**
  - Transfers data to the configuration space of each agent
- **Memory Read Multiple (1100)**
  - Fetches a full cache line and starts fetching the next
  - Differs from Memory Read by the intent to fetch more than one cache line before disconnecting
**PCI Commands (Cont.)**

- **Dual Address Cycle (1101)**
  - Generates two 32-bit address cycles to produce one 64-bit address
- **Memory Read Line (1110)**
  - Reads up to the cacheline boundary
  - Differs from Memory Read by the intent to complete a full cacheline read
- **Memory Write and Invalidate (1111)**
  - Transfers a complete cacheline to memory, then invalidates the line in cache (Writeback & HITM#)
  - Differs from Memory Write by its guarantee of a minimum transfer of one complete cacheline
Memory Access

- **Reads**
  - Memory Read
    - burst of less than a cache line
  - Memory Read Line
    - burst of cache line
  - Memory Read Multiple

- **Writes**
  - Memory Write
  - Memory Write & Invalidate
Bus Read
PCI Bus Write
Premature Termination

- Master Abort
  - No DEVSEL from any target
- Disconnect
  - Target can’t continue burst, stop after current transfer
- Retry
  - Target can’t complete current cycle, retry later
- Abort
  - Target has fatal error, don’t retry
Master Abort

- CLK
- FRAME#
- AD
- C/BE#
- IRDY#
- TRDY#
- DEVSEL #

Categories:
- Fast
- Medium
- Slow
- Bridge

PCI
Disconnect

- **CLK**
- **FRAME#**
- **AD**
- **IRDY#**
- **TRDY#**
- **STOP#**
- **DEVSEL #**

**Data**

**Data transfer**

**wait**

**Data transfer**

**TRDY - True, Stop - True, Devsel - True**
Retry

CLK

FRAME#

IRDY#

TRDY#

STOP#

DEVSEL#

TRDY - False, Stop - True, Devsel - True

PCI
Target Abort

TRDY - False, Stop - True, Devsel - False

PCI
PCI Arbitration
PCI Bus Arbitration
Arbitration

- Arbitration is access based
  - Master must arbitrate for each bus access
- Central arbitration scheme
  - Each master has a unique request and grant signal
- Arbitration is hidden
  - Occurs during previous bus cycle
Arbitration Scheme

- An arbiter may implement any scheme which is fair
  - Fairness means that, in general, the arbiter must advance to a new agent when the current master releases the bus
  - Each potential master must be granted access to the bus independently of other requests
Bus Parking

- Parking permits the arbiter to select an agent, by asserting its GNT#, when no other agent is using or requesting the bus.
- The arbiter determines how this selection is made:
  - Fixed, Last Used, …, or None.
Delayed Transactions
Delayed Transactions

- Used by two types of devices:
  - I/O Controllers
    - Single delayed transaction in general
  - Bridges
    - Particularly PCI-to-PCI bridges
    - Multiple delayed transactions in general
Delayed Transactions

- Master issues read
- Slave latches date
- Slave terminates with retry
- Slave transfers data from memory to buffer
- Master issues repeat read
- Slave terminates with TRDY
Latency
Latency

- Arbitration latency
  - Bus acquisition latency
- Master data latency
- Target latency
  - Initial
  - Subsequent
Bus Access Latency

- Number of clocks from the assertion of REQ# to the completion of the first data transaction
  - Sum of:
    - Arbitration Latency
    - Bus Acquisition Latency
    - Master Latency
    - Target Latency
Latency

Master asserts REQ#

Master receives GNT#

Arbitration Latency

Acquisition Latency

Target asserts TRDY#

Target Latency

Master asserts FRAME#
Latency Timer

- Latency Timer (LT) controls trade-off between high throughput and low latency
- Latency Timer counter sets minimum number of clocks before the master must surrender GNT#
Exclusive Access

Lock
Exclusive Access

- Lock is an exclusive access mechanism which also permits non-exclusive accesses to proceed.
- This is referred to as Resource Lock. A hardware lock can be held across several accesses without interfering with other, non-exclusive, real-time transfers.
- This mechanism locks only the target of the original locked access. Other transfers, to agents other than the locked device, may share the bus.
- Exclusive access is provided through the use of the LOCK# signal.
PCI Hardware
PCI Card Connectors

- **5V 32-Bit Connector**
  - Rear
  - Front
  - Keyway
  - 33 MHz

- **5V 64-Bit Connector**
  - Rear
  - Front
  - 33/66 MHz

- **3.3V 32-Bit Connector**
  - Rear
  - Front

- **3.3V 64-Bit Connector**
  - Rear
  - Front
5 V To 3.3 V Migration Path

- **5 Volt Card**
- **Dual-Voltage Card**
- **3 Volt Card**

- **5 Volt System**
  - Key away from backpanel

- **3 Volt System**
  - Key near backpanel
PCI Component Pin Out
Trace Length

Less than 1.5" Trace Length
PCI Component Bus Loading

- Clock Driver
- PCI Device
- PCI Device

LOW

HIGH

V_{test} 5v = 1.5 v
V_{test} 3v = 0.4 Vcc

V_{test} 1v = 1.5 v

V_{test} 3v = 0.4 Vcc

Driving PCI

Driving 50pF Test Load

V_{ih}

V_{il}

T_{prop}

T_{prop}
PCI Load Definition

- Tprop = 10ns @33MHz, 5ns @ 66MHz
- <10 PF per PCI line
- <20 NH per PCI line
- 60 - 100 Ohms Z
- 150 to 190 ps/inch signal velocity (50%C ± 5%)
- Shared signals are limited to 1.5 inches
- Unshared signals are limited to 2 inches
- Clock must be 2.5 ± .1 inch, only one load
- Max Clock skew 2ns @ 33MHz, 1ns @ 66 MHz
- No maximum number of loads is given in the specification. Pullup resistor calculations assume 10 to be typical, and 16 to be worst case loading
PCI Load Restrictions

It is specifically a violation of the PCI specification to:

- Attach an expansion ROM directly, or via bus transceivers, on any PCI pins
  - ROM code may not execute in-place
- Attach two or more PCI devices on an expansion board except behind a PCI component
- Attach any logic, other than a single PCI device, which snoops PCI pins
- Use PCI component sets which place more than one load on any PCI pin,
  - Separate address and data path components
- Use a PCI component which has more than 10pF per pin
- Attach any pull-up resistors, or other discrete devices, to PCI signals except behind PCI component
66MHz Operation

- PCI operates at 66 MHz only in the 3.3V environment
- Agents indicate 66MHz capability both electrically and programmatically
  - M66EN pin
  - 66M bit in status register
- Any 33MHz board on the bus grounds the M66EN pin and disables 66 MHz operation
  - Pin B49 grounded indicates 33MHz
PCI BIOS

32-Bit Operating Systems
PCI BIOS

- Only BIOS is permitted to access PCI:
  - Configuration Registers
  - Interrupt Routing Logic
  - Special Cycle generation logic
PCI BIOS Functions

- Determine configuration mechanism
- Detect all PCI buses in the system
- Search for PCI devices
- Read/Write configuration registers
- Assign Interrupts to PCI devices
- Determine Special Cycle abilities
- Permit Special Cycle generation
System Initialization

- Configuration allows software (BIOS) to initialize the system
- Each device has configuration registers
- At power up software scans bus(es)
- Software analyses system requirements
- Configuration registers are set to configure individual devices
Configuration Types

- Specific bus commands
  - configuration read (C/BE# = 1010)
  - configuration write (C/BE# = 1011)
- Type 0
  - local PCI bus
    • IDSEL line indicates device
    • address field indicates register
- Type 1
  - remote PCI bus (through bridge)
    • address field indicates bus, device and register
## Configuration Space Header

<table>
<thead>
<tr>
<th>Device ID</th>
<th>Vendor ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Command</td>
</tr>
<tr>
<td>Class Code</td>
<td>Rev</td>
</tr>
<tr>
<td>BIST</td>
<td>Header Type</td>
</tr>
<tr>
<td>Base Address 0</td>
<td></td>
</tr>
<tr>
<td>Base Address 1</td>
<td></td>
</tr>
<tr>
<td>Base Address 2</td>
<td></td>
</tr>
<tr>
<td>Base Address 3</td>
<td></td>
</tr>
<tr>
<td>Base Address 4</td>
<td></td>
</tr>
<tr>
<td>Base Address 5</td>
<td></td>
</tr>
<tr>
<td>Cardbus CIS Pointer</td>
<td></td>
</tr>
<tr>
<td>Subsystem ID</td>
<td>Subsystem Vendor ID</td>
</tr>
<tr>
<td>Expansion ROM Base Address</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Max_Lat</td>
<td>Min_Gnt</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Command Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>I/O Space Enable</td>
</tr>
<tr>
<td>1</td>
<td>Memory Space Enable</td>
</tr>
<tr>
<td>2</td>
<td>Bus Master Enable</td>
</tr>
<tr>
<td>3</td>
<td>Special Cycle Enable</td>
</tr>
<tr>
<td>4</td>
<td>Memory Write and Invalidate Enable</td>
</tr>
<tr>
<td>5</td>
<td>VGA Palette Snoop Enable</td>
</tr>
<tr>
<td>6</td>
<td>Parity Error Enable</td>
</tr>
<tr>
<td>7</td>
<td>Wait Cycle Control</td>
</tr>
<tr>
<td>8</td>
<td>SERR# Enable</td>
</tr>
<tr>
<td>9</td>
<td>Fast Back to Back Enable</td>
</tr>
<tr>
<td>10:15</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
IDSEL Line Addressing

- Addressing the Configuration Space of a device is done using the IDSEL signal as a chip select, and setting AD[1::0] to 00 to indicate a Type 0 configuration transaction.
- The method used to generate IDSEL is system specific, however, if no other mapping is required the following may be used:
- The IDSEL of Device 0 is connected to AD[16], Device 1 is connected to AD[17], etc. through Device 16 connected to AD[31].
Interrupt Line

- 8-Bit register which contains interrupt line routing
- Value in this register is the interrupt number (IRQ) to which the device’s interrupt pin is connected
- If, during initialization, the device requests an IRQ, one is assigned.
  - The IRQ number is placed in this register by configuration software at initialization time
Interrupt Pin

- The interrupt pin register indicates which of the four interrupts, INTA#-INTD#, the device or function uses.
- A value of 1 corresponds to INTA#, 2 to INTB# etc.
- Devices which do not use interrupts place 0 in this register.
- This register is read only.
- Each function within a device may use one interrupt.
  - A single function device must use INTA#.
  - If a multifunction device requires two interrupts they must be INTA# and INTB# etc.
- Interrupts may be shared (chained).
Interrupt Routing

- Slot 1: A, B, C, D
- Slot 2: A, B, C, D
- Slot 3: A, B, C, D

Routing:
- Route 0: A to IRQ5
- Route 1: B to IRQ9
- Route 2: C to IRQ10
- Route 3: D to IRQ11, IRQ12

PCI 66
MIN_GNT and MAX_LAT

- Minimum Grant and Maximum Latency
- MIN_GNT defines the burst-period length required by the device
- MAX_LAT specifies how often the device must gain access to the PCI bus
  - Maximum time between accesses
- Both registers are read only and have a granularity of 250 nanoseconds
- Values of 0 in these registers indicates no major latency requirements for the device
Documentation

- PCI Local bus Specification Rev 2.1
  - PCI Special Interest Group
  - http://www.pcisig.com/
- PCI System Architecture, 3rd Edition
  - Mindshare
  - ISBN# 0-201-40993-3
- PCI Hardware & Software Architecture and Design
  - Edward Solari & George Willse