Peripheral Component Interconnect (PCI) Bus

Aarti Phadke, Divisha Yadav, Komal Pandey, Preksha Bhandari and Purva Mangle

***Abstract*—This report presents an overview of the Peripheral Component Interconnect (PCI) Bus, a widely used standard for connecting peripheral devices to a computer's central processing unit (CPU). It explains the architecture, working principles, and signaling mechanisms of the PCI bus along with its role in enabling high-speed data transfer between hardware components. The report also covers different PCI versions such as PCI-X and PCI Express (PCIe), their performance improvements, and backward compatibility. Furthermore, it highlights key concepts like bus arbitration, configuration space, and plug-and-play functionality. Finally, the report discusses applications of PCI in modern computing, including graphics cards, network interfaces, and storage controllers.**

***Index Terms*—PCI, PCI Express, bus architecture, plug and play, computer hardware.**

I. Introduction

THE Peripheral Component Interconnect (PCI) Bus is a high-speed communication pathway that connects peripheral devices—such as graphics cards, sound cards, and network adapters—to the CPU and main memory. Introduced by Intel in the early 1990s, PCI replaced older standards like ISA and EISA, offering higher bandwidth and plug-and-play capabilities. It allows multiple devices to share the same data, address, and control lines through a well-organized bus system. The PCI bus became a major step forward in system performance and modularity, making it easier to upgrade and add new components without complex configurations.

II. Features of PCI Bus

TABLE I
Key Features of PCI Bus

Feature	Description
Data Bus Width	32-bit or 64-bit
Clock Speed	33 MHz or 66 MHz
Data Transfer Rate	Up to 533 MB/s (PCI-X), several GB/s (PCIe)
Plug and Play	Automatic device configuration
Bus Mastering	Devices can control data transfer directly
Interrupt Handling	Shared and level-triggered interrupts
Backward Compatibility	Supported by newer PCI standards

III. Architecture and Working

The PCI (Peripheral Component Interconnect) bus architecture provides a high-performance, processor-independent interconnection mechanism between the CPU and peripheral devices such as network cards, sound cards, and storage controllers. It was designed to replace earlier, slower interfaces by providing a standardized platform capable of plug-and-play operation and efficient communication among devices.

The PCI architecture is divided into several logical and physical layers that enable parallel data transfer and shared bus control. A typical PCI system consists of the following major components:

1) CPU and Memory Subsystem: Acts as the master that initiates communication with peripheral devices through the PCI bridge. It sends commands and receives data as part of input/output or memory operations.

2) PCI Host Bridge (Chipset): Serves as an interface between the CPU's local bus and the PCI bus. It manages address translation, buffering, and timing synchronization to ensure smooth communication between the two domains.

3) PCI Bus: The shared parallel communication channel that carries address, data, and control signals. It supports multiple masters and slaves and uses arbitration to control bus access.

4) PCI Devices: Include peripherals such as network adapters, USB controllers, and graphics cards. Each device has a configuration ROM or registers that store device identification, interrupt lines, and address space requirements.

5) Configuration Space: A dedicated 256-byte address space for each PCI device used to store device-specific information and settings, enabling automatic detection and configuration by the system firmware or operating system.

A. Working Principle

The operation of the PCI bus involves initialization, configuration, and data transfer phases. During system startup, the PCI firmware (BIOS or UEFI) performs device enumeration by reading configuration ROMs of all PCI devices. This process identifies connected devices, allocates base addresses, assigns interrupt lines, and records the configuration in the system tables.

Once configuration is complete, data transfer can occur between the CPU and devices or directly between devices in case of bus mastering. The PCI protocol supports three types of transactions:

1) Memory Read/Write: Used to transfer data between system memory and a PCI device. 2) I/O Read/Write: Accesses specific I/O port addresses for device communication. 3) Configuration Read/Write: Used by the CPU to access the configuration space of devices.

The PCI bus employs a shared communication medium using parallel lines. Arbitration ensures that only one device acts as the master at any given time. When a device requests control of the bus, the PCI arbiter grants access based on predefined priority rules. Once granted, the master initiates the transaction, sends address and control signals, and exchanges data with the target device.

During a transaction, handshaking signals like FRAME#, IRDY# (Initiator Ready), and TRDY# (Target Ready) coordinate the timing of data transfers. Each data phase completes when both initiator and target are ready, ensuring synchronous and reliable operation.

PCI also supports burst transfers, allowing multiple data words to be transferred consecutively without repeated address phases. This increases data throughput and reduces latency, making PCI suitable for high-speed peripherals.

B. Interrupt Handling and Plug-and-Play

PCI provides interrupt lines (INTA#, INTB#, INTC#, INTD#) to signal events requiring CPU attention. These interrupts can be shared among multiple devices, and their mapping is managed dynamically by the BIOS or operating system. The plug-and-play capability allows new devices to be added without manual configuration, as the system automatically detects and assigns resources.

IV. Advantages of PCI Architecture

The PCI bus architecture offers high bandwidth, bus mastering support, backward compatibility, and flexible configuration. Its processor-independent design allows it to work across multiple hardware platforms, enabling widespread use in personal computers, servers, and embedded systems.

Overall, the architecture of the PCI bus ensures efficient communication, modularity, and scalability, forming the foundation of many modern interconnection standards such as PCI-X and PCI Express.

V. Bus Arbitration and Configuration

Bus arbitration ensures that only one device uses the PCI bus at a time. The arbiter grants bus access based on priority or round-robin logic. Each PCI device includes configuration registers containing vendor ID, device ID, class codes, and base address registers. During initialization, the BIOS or operating system reads these to allocate resources, achieving automatic configuration (Plug and Play).

VI. Performance Parameters

The PCI bus performance depends on:

- Bandwidth: Based on bus width and clock frequency (e.g., 32-bit × 33 MHz = 133 MB/s).
- Latency: Reduced by burst transfers and pipelined communication.
- Bus Utilization: Efficiency of active data transfer.
- Power Efficiency: PCI Express (serial) provides better power efficiency and throughput.

PCIe replaces the shared bus with dedicated point-to-point serial links, significantly increasing speed and reducing signal interference.

VII. Challenges in Traditional PCI Bus

- Limited bandwidth due to shared bus design.
- Electrical interference at higher clock rates.
- Scalability challenges with more connected devices.
- Higher power consumption compared to PCIe.

These limitations motivated the transition to PCI Express, which offers serial data transmission with separate lanes for each device.

VIII. Recent Advancements

PCI Express (PCIe) architecture enables full-duplex serial communication and scalable lane configurations (x1, x4, x8, x16). Each lane supports data rates up to 32 GT/s in PCIe 5.0. Advancements include:

- Hot-plugging support.
- Advanced Error Reporting (AER).
- Virtual Channels and QoS.
- Low-Power States for energy efficiency.

PCIe Gen 6 and Gen 7 employ PAM4 signaling and Compute Express Link (CXL) integration for higher data throughput in AI and cloud computing systems.

IX. Applications

- Graphics Processing Units (GPUs)
- Network Interface Cards (NICs)
- NVMe Solid-State Drives (SSDs)
- Sound and Video Cards
- Embedded and Industrial Control Systems

X. Conclusion

The PCI Bus has been a cornerstone of computer hardware communication, providing standardized and reliable connectivity. While traditional PCI faced challenges in scalability and bandwidth, PCI Express has overcome these limitations with serial links, higher speeds, and energy efficiency. Today, PCIe remains critical in advanced computing, AI accelerators, and embedded platforms due to its versatility, scalability, and backward compatibility.

Acknowledgment

The authors sincerely express their gratitude to Prof. Aarti Phadke of K. J. Somaiya College of Engineering for her invaluable guidance and support throughout this work.

References

[1] T. Hamacher, Z. Vranesic, and S. Zaky, *Computer Organization*, 5th ed., McGraw-Hill Education, 2012.

[2] HowStuffWorks, "How PCI Express Works," Available: https://computer.howstuffworks.com/pci-express.htm. [Accessed: Oct. 16, 2025].

[3] Wikipedia, "Peripheral Component Interconnect," Available: https://en.wikipedia.org/wiki/Peripheral$_{C}omponent_{I}nterconnect.[Accessed: Oct.16, 2025].FPGA4Fun, "PCIBusTutorial," Available: https://www.fpga4fun.com/PCI1.html.[Accessed: Oct.16, 2025].

[4] L. Jin, "The design of PCI bus interface," *ResearchGate*, 2002. Available: https://www.researchgate.net/publication/4012315$_{T}he_{d}esign_{o}f_{P}CI_{b}us_{i}nterface.[Oct.16, 2025].Meinberg, "PCIBusOverview," Available: https://www.meinbergglobal.com/english/info/pci.htm.[Accessed: Oct.16, 2025].

[5] ScienceDirect, "Peripheral Component Interconnect (PCI) Bus," Available: https://www.sciencedirect.com/topics/engineering/peripheral-component-interconnect-bus. [Accessed: Oct. 16, 2025].