УДК 004.32

REVIEW KEY FEATURES AND ADVANTAGES OF USB

С. В. Здищук
студент-магістр кафедри АУВП,
Луцький національний технічний університет
E-mail: zedazen@gmail.com

Ця оглядова стаття в скороченій формі демонструє нам головні переваги USB порту над іншими системними портами та шинами у якості базової платформи розробки периферійних пристроїв та додатків для роботи з ними. Також вона може слугувати базовим джерелом для непрофесійних розробників, хто хоче покращити свої знання про USB.

Ключові слова: *USB порт, шина USB, системна архітектура USB, ключові особливості USB.*

Эта обзорная статья в краткой форме показывает нам главные преимущества USB порта над другими системными портами и шинами в качестве базовой платформы разработки периферических устройств и приложений для работы с ними. Также она может служить базовым источником для непрофессиональных разработчиков, которые хотят улучшить свои знания о USB.

Ключевые слова: *USB порт, шина USB, системная архитектура USB, ключевые особенности USB.*

This overview article shows us in short form main advantages of the USB over other system ports and buses as the base platform for peripheral development. Also it can be a base source for non professional developers, who want to improve their knowledge about USB.

Keywords: *USB port, USB bus, USB system architecture, USB key features.*

Introduction

Many PCs designed today still implement peripheral devices based on interfaces used in the original IBM PC designs of the early 1980s. These implementations have numerous shortcomings that cause both designers and users considerable frustration. This article discusses the primary design goals of USB 2.0 and reviews the shortcomings of the legacy implementation.

Installation and Configuration of Expansion Cards

When peripherals are purchased, many of them require the installation of expansion cards. This, of course, may involved removing the cover of the PC, setting the switches and jumpers to configure the card, inserting the card, and replacing the cover. The trouble only begins there. Once the system is powered up the software for this device may have to be installed from sources, which can also be a frustrating process for novice and experienced user alike.

No Hot Attachment of Peripherals

When most legacy I/O devices are attached to the system, they will not work without first restarting the system. Restarting the system is required so that the new peripheral can be detected by software. In the process, system resources must be selected and assigned to the new device (e.g., I/O space, IRQ line, and DMA channel) in order for it to work correctly and to ensure that the resource selected is not already being used by another device in the system.

Cost

The cost of implementing systems and peripheral devices based on the original PC design is fairly expensive due to the high cost of the standard peripheral connectors and associated cables. Since most of the standard connectors on the PC are already used by a wide variety of peripheral devices, it may be necessary to build an expansion card to provide a way to attach your peripheral device to the system, making the solution even more costly.

The USB Paradigm

The design goals of a new peripheral standard should overcome the existing shortcomings perceived by manufacturers and users, while providing for further growth, performance, and expansion. The design goals of USB include:

- a single connector type to connect any PC peripheral;
- ability to attach many peripheral devices to the same connector;
- a method of easing the system resource conflicts;
- hot plug support;
- automatic detection and configuration of peripheral devices;
- low-cost solution for both system and peripheral implementations;
- enhanced performance capability;
- support for attaching new peripheral designs;
- support for legacy hardware and software;
- low-power implementation.

Fig. 1 provides a system view of USB implemented in a PCI-based system. In this implementation the USB host controller resides on the PCI bus. The controller acting as a bus master obtains data structures from memory that describe the USB transactions that have been scheduled by system software for delivery over the USB.

USB breaks away from the resource problems associated with legacy PC I/O implementations. The resource constraints related to I/O address space, IRQ lines, and DMA channels no longer exist with the USB implementation. Each device residing on the USB is assigned an address known only to the USB subsystem and does not consume

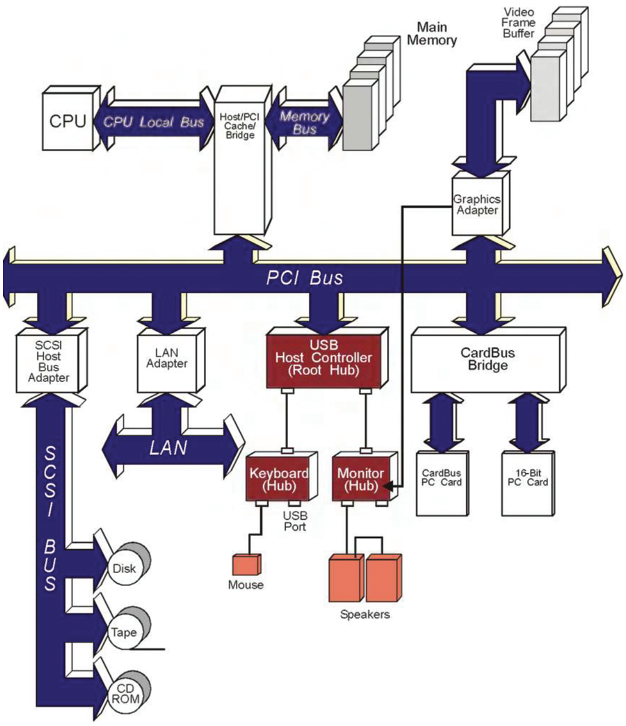

Fig. 1. USB System Implemented in a PCI-Based Platform

any system resources. USB supports up to 127 device addresses that limit the number of USB devices supported in a single USB implementation. USB devices typically contain a number of individual registers or ports that can be indirectly accessed by USB device drivers. These registers are known as USB device endpoints.

When a transaction is sent over the USB, all devices (except low-speed devices) will see the transaction. Each transaction begins with a packet transmission that defines the type of transaction being performed along with the USB device and endpoint addresses. This addressing is managed by USB software. Other non-USB devices and related software within the system are not impacted by these addresses. Every USB device must have an internal default address location (called endpoint zero) that is reserved for configuring the device. Via endpoint zero, USB system software reads standard descriptors from the device. These descriptors provide configuration information necessary for hardware and software initialization. In this manner, system software can detect the device type (or class information) and determine how the device is intended to be accessed.

Enhanced System Performance

The Universal Serial Bus (USB) creates a solution for attaching PC peripherals that balances performance and cost. USB supports three transmission rates:

- 1,5 Mb/s;
- 12 Mb/s;
- 480 Mb/s.

The 1.0 and 1.1 (1.x) versions of USB support only the 1,5 Mb/s and 12 Mb/s speeds. These transmission rates were intended to support low- and medium-speed peripherals, while the 2.0 version of the USB specification defines a 480 Mb/s rate that can support selected high-speed devices, and permits a larger number of low- or full-speed devices to operate on a single bus. Table 1 lists the types of devices that fall into these performance ranges.

Table 1

Applications, Relative Performance Required and Desired Attributes

Performance	Applications	Attributes
Low Speed: Interactive Devices 10—100 Kb/s	Keyboard, Mouse Stylus Game peripherals Virtual Reality peripherals	Lower cost Hot plug Ease of use Multiple peripherals
Medium Speed: Phone, Audio 500—10,000 Kb/s	ISDN PBX POTS Digital audio Scanner Printer	Lower cost Ease of use Guaranteed latency Guaranteed bandwidth Hot plug Multiple devices
High Speed: Video, Disk, LAN 25—500 Mb/s	Mass storage Video conferencing Imaging Broadband	Low cost Hot plug High bandwidth Guaranteed bandwidth Guaranteed latency Multiple devices Ease of use

Hot Plug and Play Support

Hot Plug and Automatic configuration is crucial to satisfying end user requirements. USB can detect the attachment of a new peripheral and automatically install the relevant software needed to access the device. This process also eliminates the need to set switches and jumpers when configuring a peripheral device and eliminates the need to restart the system when the device is attached. In short, the peripheral can simply be attached by the user and be ready for immediate use.

Expandability

Hub devices provide additional ports for attaching other USB devices as illustrated in Figure 1. Hubs can be standalone devices, or can be integrated into other USB peripherals such as printers or keyboards. Physical USB devices that contain hubs and that have one or more internal devices attached to the hub ports are called compound devices.

Legacy Hardware/Software Support

Older operating systems have no knowledge of USB, so the system designer must choose whether to provide USB support. Additionally, traditional system firmware (initialization code, boot code, and BIOS) is based on standard PC legacy hardware and must be adapted to support USB if USB boot support is desired.

Low Cost

USB can reduce the overall cost of peripheral design and system support.

Much of this cost reduction on the peripheral side comes from the ability to connect the peripheral directly to a USB port, thereby eliminating the requirement to design an expansion card to provide a connection for the device. Another source of cost reduction for both the system and peripheral designers is the connectors and cables. The standard USB cables and connectors create a very large market for these items and competition between vendors reduces their cost. The USB serial bus implementation also reduces cost when compared to parallel bus implementations that require a much larger number of pins and traces.

The system cost savings can be realized by eliminating the cost of the wide variety of connections that must be supported for standard peripherals such as the parallel, serial, keyboard, and mouse connectors. In the short term USB has been added while the older connectors remain.

Summary of Key USB Features

Table 2 lists the key features that comprise the USB implementation.

Conclusions

This overview article shows us in short form main advantages of the USB over other system ports and buses as the base platform for development.

Table 2

Key features of the USB

Feature	Description
Low Cost	The USB provides a low-cost solution for attaching peripheral devices to PCs
Hot Pluggable	Device attachment is automatically detected by the USB and software automatically configures the device for immediate use, without user intervention
Single Connector Type	The USB defines a single connector used to attach any USB device. Additional connectors can be added with USB hubs
127 Devices	Supports the attachment of 127 devices per USB
Low Speed, Full Speed, and High Speed Devices	The USB 2.0 supports three device speeds: 1,5 Mb/s, 12M b/s, and 480 Mb/s
Cable Power	Peripherals can be powered directly from the cable. 5.0vdc power is available from the cable. The current available can vary from 100 ma — 500 ma depending on the hub port
System Resource Requirement Eliminated	USB devices, unlike their ISA, EISA, and PCI cousins, require no memory or 1/O address space and need no IRQ lines
Error Detection and Recovery	USB transactions include error detection mechanisms that are used to ensure that data is delivered without error. In the event of errors, transactions can be retried
Power Conservation	USB devices automatically enter a suspend state after 3 ms of no bus activity. During suspend devices can consume no more than 500 ua of current
Support for Four Types of Transfers	The USB defines four different transfer types to support different transfer characteristics required by devices. Transfer types include: bulk, isochronous, interrupt, and control transfers
Ability to Extend Bus	USB hubs can be installed to add additional ports that permit additional devices to be attached

Literature

1. Axelson, Jan. Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems (Complete Guides series) [Текст] / Jan Axelson. — Видавництво «Lakeview Research», 2007.
2. McDowell, Steven. USB Explained [Текст] / Steven McDowell, Martin Seyer. — Видавництво «Prentice Hall Ptr», 2006.
3. Anderson, Don. Universal Serial Bus System Architecture (2-nd Edition) (PC System Architecture Series) [Текст] / Don Anderson. — Видавництво «Addison-Wesley Longman Publishing Co., Inc.», 2007.
4. Axelson, Jan. USB Complete: Everything You Need to Develop Custom USB Peripherals (Complete Guides series) [Текст] / Jan Axelson. — Видавництво «Penram International Publishing Inc.», 2005.