

MULTI-HOST USB DEVICE CONTROLLER

Priority Claim

[0001] This application claims benefit of priority of provisional application Serial No. 60/792,178 titled "Multi-Host USB Device Controller", filed on April 14, 2006, whose inventors are Mark. R Bohm and Atish Ghosh, and which is hereby incorporated by
5 reference as though fully and completely set forth herein.

BACKGROUND OF THE INVENTION

10 **Field of the Invention**

[0002] The present invention relates generally to computer hardware and, more specifically, to Universal Serial Bus (USB) controllers.

Description of the Related Art

15

[0003] The Universal Serial Bus (USB) allows coupling of peripheral devices to a computer system. USB is a serial cable bus for data exchange between a host computer and a wide range of simultaneously accessible devices. The bus allows peripherals to be attached, configured, used, and detached while the host is in operation. For example,
20 USB printers, scanners, digital cameras, storage devices, card readers, etc. may communicate with a host computer system over USB. USB based systems may require that a USB host controller be present in the host system, and that the operating system (OS) of the host system support USB and USB Mass Storage Class Devices.

[0004] USB devices may communicate over the USB bus at low-speed (LS), full-speed
25 (FS), or high-speed (HS). A connection between the USB device and the host may be established via digital interconnect such as Interchip USB, ULPI, UTMI, etc., or via a four wire interface that includes a power line, a ground line, and a pair of data lines D+ and D-. When a USB device connects to the host, the USB device may first pull a D+

line high – or the D- line if the device is a low speed device – using a pull up resistor on the D+ line. The host may respond by resetting the USB device. If the USB device is a high-speed USB device, the USB device may “chirp” by driving the D- line high during the reset. The host may respond to the “chirp” by alternately driving the D+ and D- lines high. The USB device may then electronically remove the pull up resistor and continue communicating at high speed. When disconnecting, full-speed devices may remove the pull up resistor from the D+ line (i.e., “tri-state” the line), while high-speed USB devices may tri-state both the D+ and D- lines.

[0005] A USB hub may be coupled to a USB host controller to allow multiple USB devices to be coupled to the host system through the USB host controller. In addition, other USB hubs may be coupled to the USB hub to provide additional USB device connections to the USB host controller. In general, the USB specification is structured so that every device is configured and accessed by a single host controller. Consumers typically desire maximum flexibility, and may want to have a simple means by which to cheaply share devices. There are several switching devices that currently allow a device to be switched between multiple USB Host controllers, but the device can generally be configured and accessed by only a single host at any given time. There also exist stand-alone USB switches that provide the capability of switching a device between upstream USB Host Controllers. These solutions, however, fail to permit simultaneous access to the USB device that is downstream of the hub or switch. The USB device is typically accessed by one single host at a time, and when access to the USB device is switched, the device must be re-configured, thereby losing internal state information.

[0006] Other corresponding issues related to the prior art will become apparent to one skilled in the art after comparing such prior art with the present invention as described herein.

SUMMARY OF THE INVENTION

[0007] In one set of embodiments, a single USB device may be shared across multiple USB hosts without needing to be re-configured or re-enumerated each and every time the upstream hosts alternate accessing the USB device. Since each host has simultaneously enumerated the device, there may be no need to detach and reconfigure the device on the fly. Multiple USB hosts may simultaneously share a single device/function, for example a Gigabit Ethernet controller. While in most present day implementations each host that is configured to have Ethernet access is implemented with its own Ethernet controller and Ethernet switch, in various embodiments of the present invention the Ethernet switch may be replaced by a less expensive and more compact multi-host USB controller, allowing each host to access the USB device directly. In another set of embodiments, storage media devices may be configured with a multi-host USB controller to provide a USB based Network Attached Storage (NAS) device that can handle storage requests from multiple USB hosts.

[0008] In various embodiments, by using a multi-host capable device controller, a shared USB device may be simultaneously configured and accessed by two or more USB hosts. The multi-host capable device may include separate buffers for each host, and may be configured with the capability to respond to USB requests from more than one host. The device may maintain a dedicated address, configuration and response information for each host. Each host may therefore establish a dedicated USB connection with the sharing device.

[0009] A USB device may be divided into three segments or blocks. The first block may comprise a USB interface that includes the physical (PHY) or digital link, USB Link layer (SIE), and other circuitry necessary to send and/or receive data over the USB. The second block may comprise an Endpoint Buffer Block, which may include the endpoint buffers that are used by the first and third blocks to buffer data and control reads and writes to/from the USB – transferred through the first block – and/or the Peripheral

Function – transferred through the third block. The third block may comprise the “Peripheral Function” itself, which may include the circuitry necessary for the specific USB device function, for example an Ethernet Controller, printer, Video Camera, etc.

5 [0010] In one set of embodiments, the first block may be replicated for each upstream host port, and some, or all, of the second block may be replicated for each upstream host port as well. In each case, the extent to which blocks and/or portions of the blocks are replicated may vary based on USB device type. The third block may correspond to the USB device that will be shared by multiple USB hosts, and may therefore not need to be duplicated. A fourth block may be added – typically between the first and second blocks,
10 or as part of either the second or third blocks – configured as an arbitration block. The internal arbitration mechanism may enable each host to access the shared Peripheral Function by either interleaving host accesses, or by using a common request/grant structure, which may hold-off one host while another host completes a data transfer to/from the shared device. The selection of the specific mechanism used may be
15 determined according to the specific USB device type that is being shared.

[0011] In some embodiments, the bandwidth from the shared peripheral function to each host may be reduced in order to allow each host equal access. In other embodiments, the bandwidth may not be reduced because the bandwidth of the Peripheral Function may exceed the bandwidth of the host. Other aspects of the present invention will become
20 apparent with reference to the drawings and detailed description of the drawings that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

5 [0012] A better understanding of the present invention may be obtained when the following detailed description is considered in conjunction with the following drawings, in which:

[0013] FIG. 1 shows a system diagram of a multi-host capable USB device coupled to multiple hosts according to one embodiment;

10 [0014] FIG. 2 shows multi-host capable devices coupling to multiple hosts according to one embodiment; and

[0015] FIG. 3 shows a logic diagram of a USB multi-host device according to one embodiment.

15

[0016] While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will
20 herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present invention as defined by the appended claims. Note, the headings are for organizational purposes only and are not
25 meant to be used to limit or interpret the description or claims. Furthermore, note that the word "may" is used throughout this application in a permissive sense (e.g., having the potential to or being able to in some embodiments), not a mandatory sense (i.e., must).

The term "include", and derivations thereof, mean "including, but not limited to". The term "coupled" means "directly or indirectly connected".

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] In one set of embodiments, a multi-host USB device may provide maximum flexibility and a simple means by which to cheaply share devices with multiple hosts, by providing a separate configuration and access interface for each upstream host. FIG. 1 illustrates a block diagram of a multi-host device 106 configured with a USB multi-host device controller 108, with multi-host device 106 coupled to first host 102 and second host 104, which may both establish control with multi-host device 106. As shown in FIG. 2, by way of examples, multi-host device 106 – configured with USB multi-host device controller 108 – may be a personal digital assistant (PDA) 130, a keyboard 126, and/or a printer 120 shared by personal computer (PC) 122 and PC 123. Many other devices may be similarly configured as multi-host devices that include a multi-host device controller 108, and the number and type of such devices is not limited to those show in Fig. 2.

[0018] In one embodiment of multi-host (or USB sharing) device 106, shown in FIG. 3, an upstream port or PHY is configured for each host to be connected. In this case upstream port 302 is configured to interface/couple to the first host (for example first host 102 shown in FIG. 1), and upstream port 304 is configured to interface/couple to the second host (for example second host 104 shown in FIG. 1). Multi-host device 106 may be addressed separately by each host, and may respond to each host within USB specified limits. Multi-host device controller 108 may internally determine which host request to fully service immediately, and may either send not-ready packets in a USB specific manner to the other host, or may interleave the host requests. Peripheral Device/Function 312 – which may comprise the main consumer component, such as a Ethernet Controller, Mass-Storage drive, etc. – may not be aware of the multi-host capability of the USB component, and may be a standard off-the-shelf item.

[0019] Multi-host device 106 may also be configured with Endpoint and status buffers 306 and 308, coupling USB multi-host device controller 108 to PHY 302 and PHY 304, respectively. Endpoint buffers 306 and 308 may be used by upstream ports 302 and 304,

and USB multi-host device controller 108 to buffer data and control reads and writes to/from each respective host corresponding to PHY 302 and PHY 304, and/or peripheral device/function 312 coupled to USB multi-host device controller 108.

5 [0020] In one set of embodiments, USB multi-host device controller 108 may be configured with an internal arbitration mechanism that may permit each host -- first host 102 and second host 104, for example -- to access shared peripheral function 312 by either interleaving host accesses, or by using a common request/grant structure that may hold-off one host while another host completes a data transfer to/from shared device/function 312. The selection of the specific mechanism used may be configured according to the
10 specific USB device type that is being shared. In one set of embodiments, the bandwidth from shared peripheral device/function 312 to each host may be reduced in order to allow each host equal access. In other embodiments, the bandwidth may not be reduced if the bandwidth of the peripheral function exceeds the bandwidth of the host.

[0021] It should be noted that while Fig. 3 shows 2 upstream ports coupling to two hosts,
15 alternate embodiments may be configured with more than two upstream ports, (and correspondingly with possibly more than two endpoint and status buffers), and while those embodiments are not shown, they are possible and are contemplated. For example, a multi-host device (e.g. a keyboard) may be configured with a multi-host device controller to couple to three or four hosts, and so forth.

20 [0022] Although the embodiments above have been described in considerable detail, other versions are possible. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications. Note the section headings used herein are for organizational purposes only and are not
25 meant to limit the description provided herein or the claims attached hereto.

We claim:

1. A USB multi-host device comprising:
first and second upstream ports configured to couple to corresponding first and
5 second hosts;
a USB device/function block; and
a multi-host device controller coupling the USB device/function block to the first
and second upstream ports, wherein the multi-host device controller is operable to
establish a dedicated USB connection between the USB device/function block and the
10 first and second upstream ports, wherein the USB device/function is not reconfigured
each time the first and second hosts alternate accessing the USB device/function block.
2. The USB multi-host device of claim 1, further comprising a first endpoint
buffer coupled between the first upstream port and the multi-host device controller, and a
15 second endpoint buffer coupled between the second upstream port and the multi-host
device controller.
3. A USB multi-host device comprising:
a USB device/function block; and
20 a multi-host device controller coupling the USB device/function block to a first
host and a second host, wherein the multi-host device controller is operable to establish a
dedicated USB connection between the first host and the USB device/function block and
between the second host and the USB device/function block, wherein the USB
device/function is not reconfigured each time the first host and the second host alternate
25 accessing the USB device/function block.

4. The USB multi-host device of claim 3, wherein the USB device/function is not re-enumerated each time the first host and the second host alternate accessing the USB device/function block.

5 5. The USB multi-host device of claim 3, further comprising a first upstream port coupled between the first host and the multi-host device controller, and a second upstream port coupled between the second host and the multi-host device controller.

10 6. The USB multi-host device of claim 3, further comprising a first endpoint buffer coupled between first upstream port and the multi-host device controller, and a second endpoint buffer coupled between the second upstream port and the multi-host device controller.

15 7. A device comprising:
a USB device/function block; and
a multi-host device controller configured to couple the USB device/function block to a plurality of hosts, wherein the multi-host device controller is operable to establish a respective dedicated USB connection between the USB device/function block and each of the plurality of hosts, wherein the USB device/function is not reconfigured each time the
20 plurality of hosts alternate accessing the USB device/function block.

8. The device of claim 7, wherein the multi-host device controller is operable to receive respective host requests from the plurality of hosts, wherein the multi-host device controller is operable to internally determine which of the respective host requests
25 to service immediately.

9. The device of claim 8, wherein the multi-host device controller is operable to interleave the respective host requests.

10. The device of claim 8, wherein the multi-host device controller is operable to send not-ready packets in a USB specific manner to hosts whose request was not immediately serviced.

5

11. The device of claim 7, wherein the multi-host device controller comprises an internal arbitration mechanism configured to permit each of the plurality of hosts to access the USB device/function block by interleaving host accesses and/or by using a common request/grant structure;

10 wherein the common request/grant structure comprises one of the plurality of hosts being granted access to the USB device/function block while remaining ones of the plurality of hosts are held off until the one of the plurality of hosts is done accessing the USB device/function block.

15 12. The device of claim 11, wherein the arbitration mechanism is configured according to a specific USB device type comprised in the USB device/function block.

13. The device of claim 7, wherein a bandwidth from the USB device/function block to each respective one of the plurality of hosts is reduced to allow each respective
20 one of the plurality of hosts equal access to the USB device/function block.

14. The device of claim 13, wherein the bandwidth is not reduced if it exceeds a bandwidth of the respective one of the plurality of hosts.

25 15. The device of claim 7, further comprising a respective upstream port coupled between the multi-host device controller and each of the plurality of hosts.

16. The device of claim 15, further comprising a respective buffer coupled between each respective upstream port and the multi-host device controller.

5 17. The device of claim 7, wherein the multi-host device controller is configured to maintain a dedicated address, configuration, and response information for each of the plurality of hosts.

18. A method for sharing a device between multiple hosts, the method comprising:
10 establishing a respective dedicated USB connection between a shared USB device/function and each of a plurality of hosts;
receiving alternating respective access requests to the shared USB device/function from two or more of the plurality of hosts; and
processing the alternating respective access requests to access the shared USB
15 device/function;
wherein the USB device/function is not reconfigured each time the USB device/function is accessed.

19. The method of claim 18, wherein said processing comprises determining
20 which of the alternating respective access requests to service immediately, and servicing that respective access request.

20. The method of claim 19, wherein said processing comprises holding off
access to the shared USB device/function by those alternating respective access requests
25 that are not immediately serviced, until the shared USB device/function is no longer accessed by the serviced respective access request.

21. The method of claim 18, wherein said processing comprises interleaving accesses requested by the alternating respective access requests to the shared USB device/function.

5 22. The method of claim 18, further comprising maintaining a dedicated address, configuration, and response information for each of the plurality of hosts.