US 20090100209A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2009/0100209 A1

(43) Pub. Date: Apr. 16, 2009

Szabelski

(54) **UNIVERSAL SERIAL BUS HUB WITH SHARED HIGH SPEED HANDLER**

(76) Inventor: **Piotr Szabelski**, Santa Clara, CA (US)

Correspondence Address:
Jeffrey C. Hood
Meyertons Hood Kivlin Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398 (US)

(21) Appl. No.: **12/340,916**

(22) Filed: **Dec. 22, 2008**

Related U.S. Application Data

(60) Continuation of application No. 11/223,570, filed on Sep. 9, 2005, now Pat. No. 7,484,018, which is a division of application No. 10/374,852, filed on Feb. 24, 2003, now Pat. No. 6,959,355.

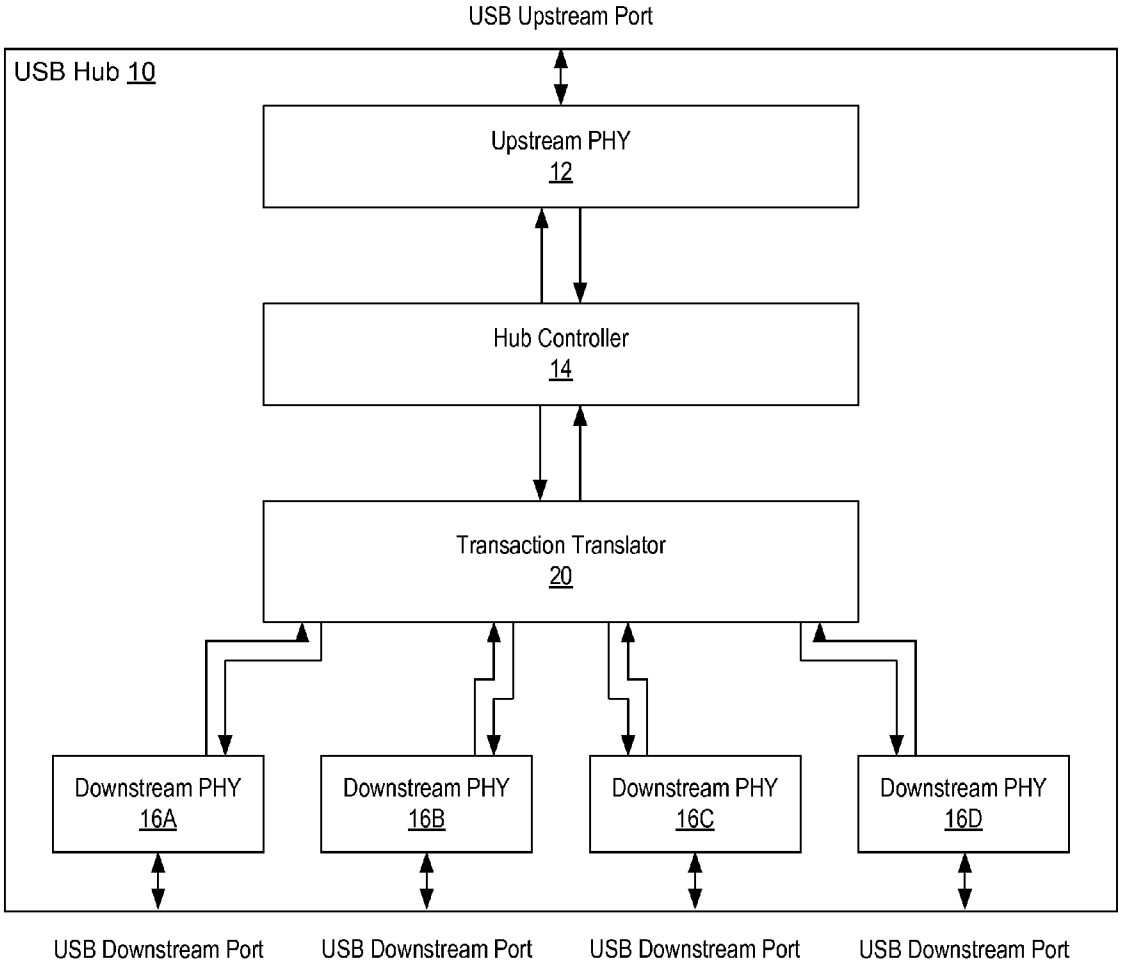
Publication Classification

(51) **Int. Cl.**
G06F 13/00 (2006.01)
G06F 13/20 (2006.01)
G06F 12/00 (2006.01)

(52) **U.S. Cl.** ... **710/313**; 710/305; 711/147; 711/E12.001

(57) **ABSTRACT**

A device may include an upstream port and several downstream ports configured to transfer data at a different data transfer rate than the upstream port. The device may also include several downstream data handlers, each coupled to a respective one of the downstream ports, and an upstream data handler coupled to the upstream port. The data handlers are configured to implement a USB protocol. The upstream data handler is configured to store specific transactions (comprising data) received through the upstream port. Each respective downstream data handler is configured to access respective transactions of the stored specific transactions intended for the downstream port associated with the respective downstream data handler, and transmit to its associated respective downstream port the data comprised in its respective transactions. Accordingly, the upstream data handler is shared between the various downstream data handlers.



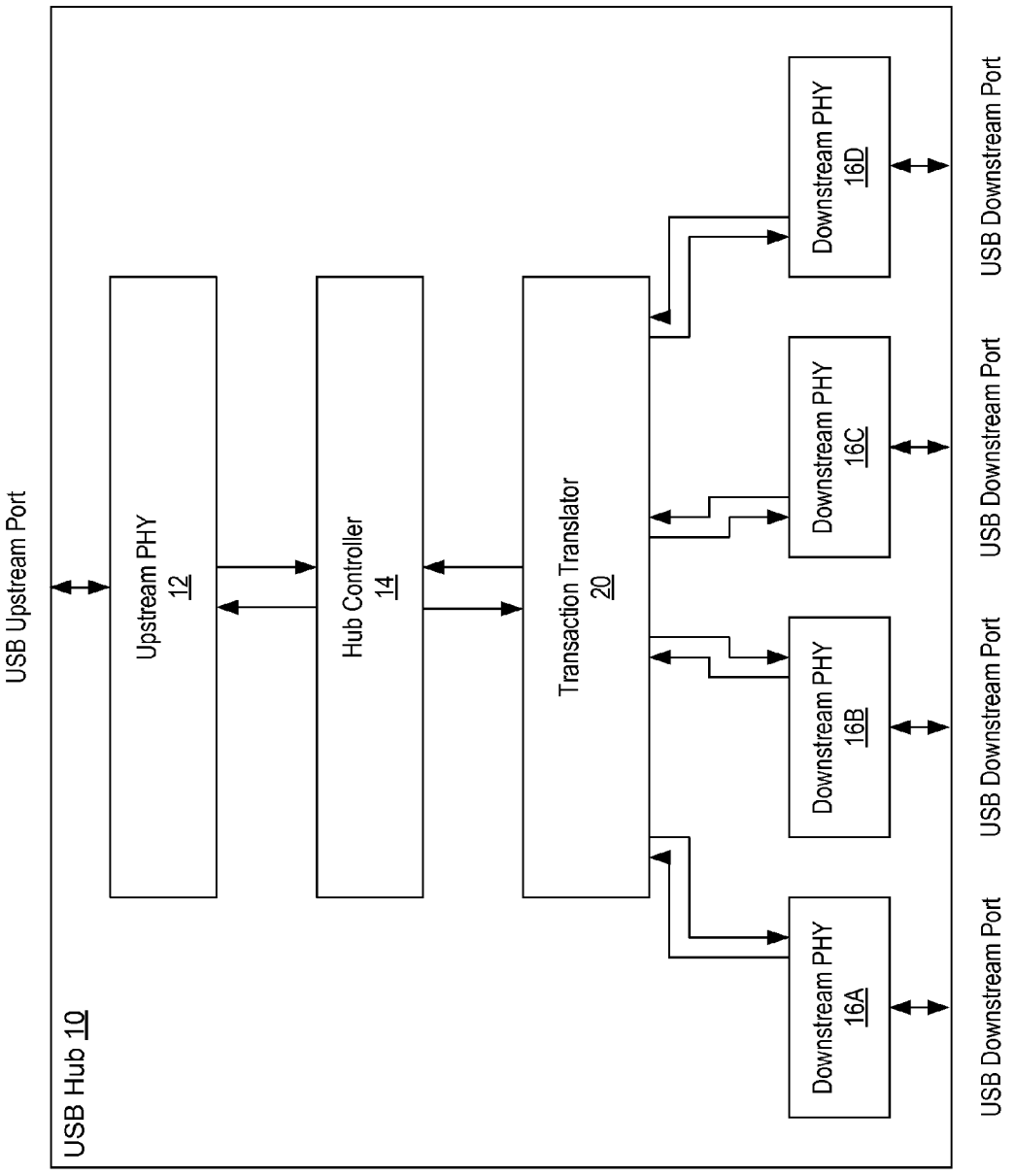

USB Upstream Port
USB Hub 10
Upstream PHY 12
Hub Controller 14
Transaction Translator 20
Downstream PHY 16A
Downstream PHY 16B
Downstream PHY 16C
Downstream PHY 16D
USB Downstream Port
USB Downstream Port
USB Downstream Port
USB Downstream Port

FIG. 1

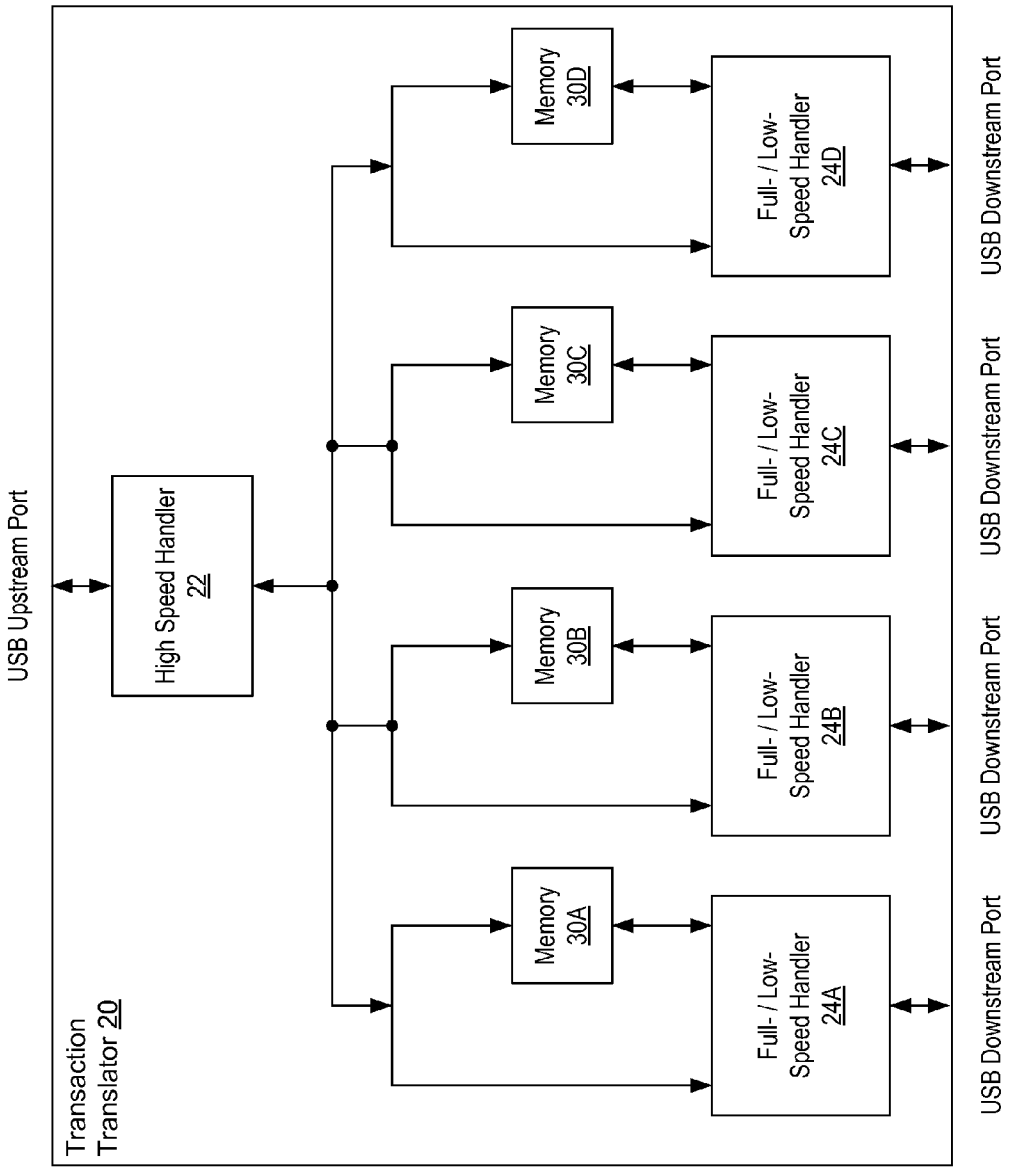

USB Upstream Port
Transaction Translator 20
High Speed Handler 22
Memory 30A
Memory 30B
Memory 30C
Memory 30D
Full- / Low-Speed Handler 24A
Full- / Low-Speed Handler 24B
Full- / Low-Speed Handler 24C
Full- / Low-Speed Handler 24D
USB Downstream Port
USB Downstream Port
USB Downstream Port
USB Downstream Port

FIG. 2A

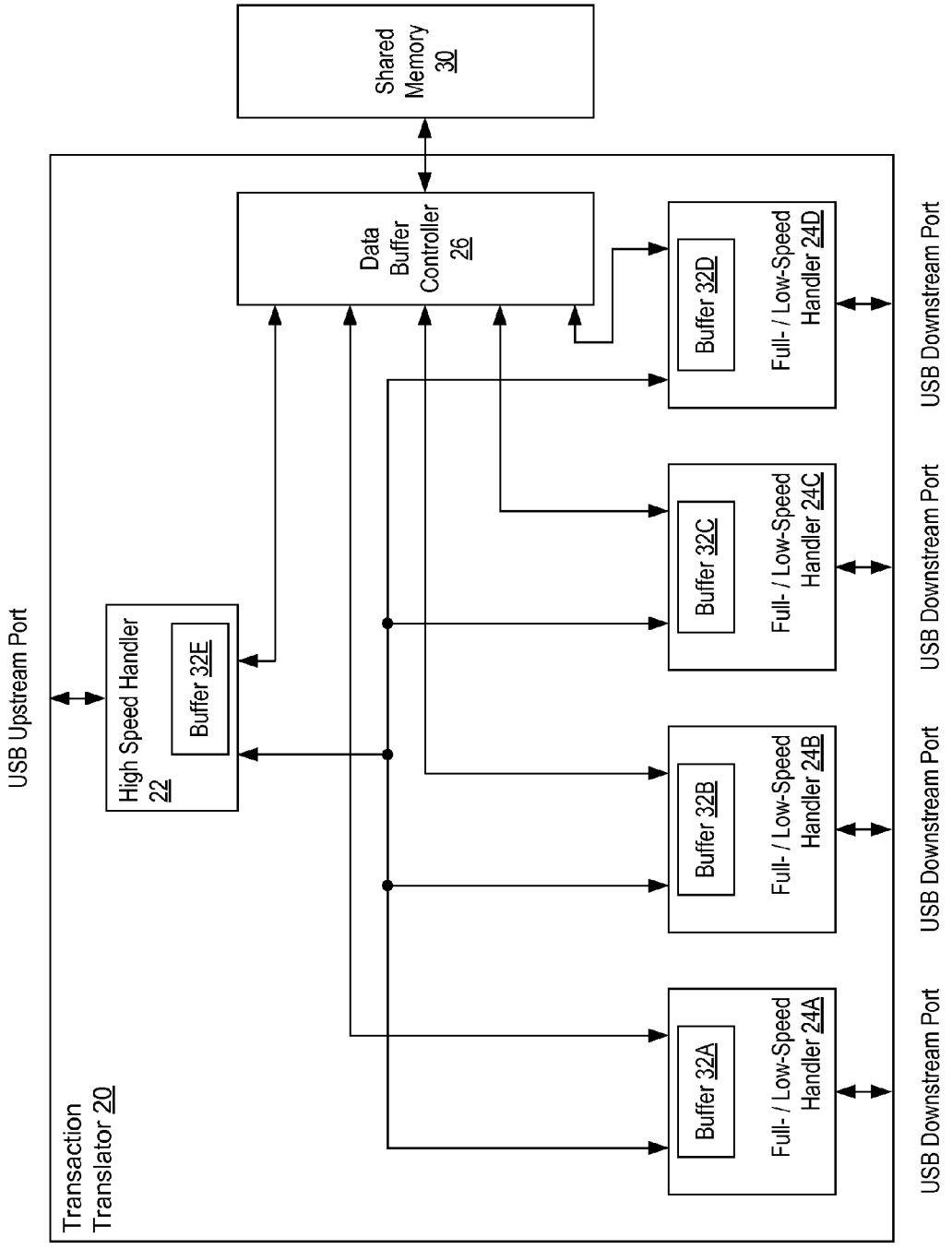

Shared Memory 30
Data Buffer Controller 26
Transaction Translator 20
USB Upstream Port
High Speed Handler 22
Buffer 32E
Full- / Low-Speed Handler 24A
Buffer 32A
Full- / Low-Speed Handler 24B
Buffer 32B
Full- / Low-Speed Handler 24C
Buffer 32C
Full- / Low-Speed Handler 24D
Buffer 32D
USB Downstream Port
USB Downstream Port
USB Downstream Port
USB Downstream Port

FIG. 2B

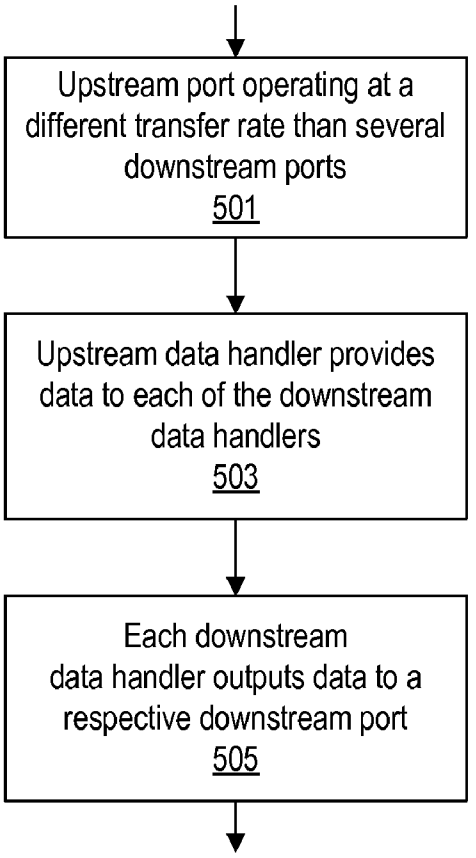

Upstream port operating at a different transfer rate than several downstream ports
501
Upstream data handler provides data to each of the downstream data handlers
503
Each downstream data handler outputs data to a respective downstream port
505

FIG. 3

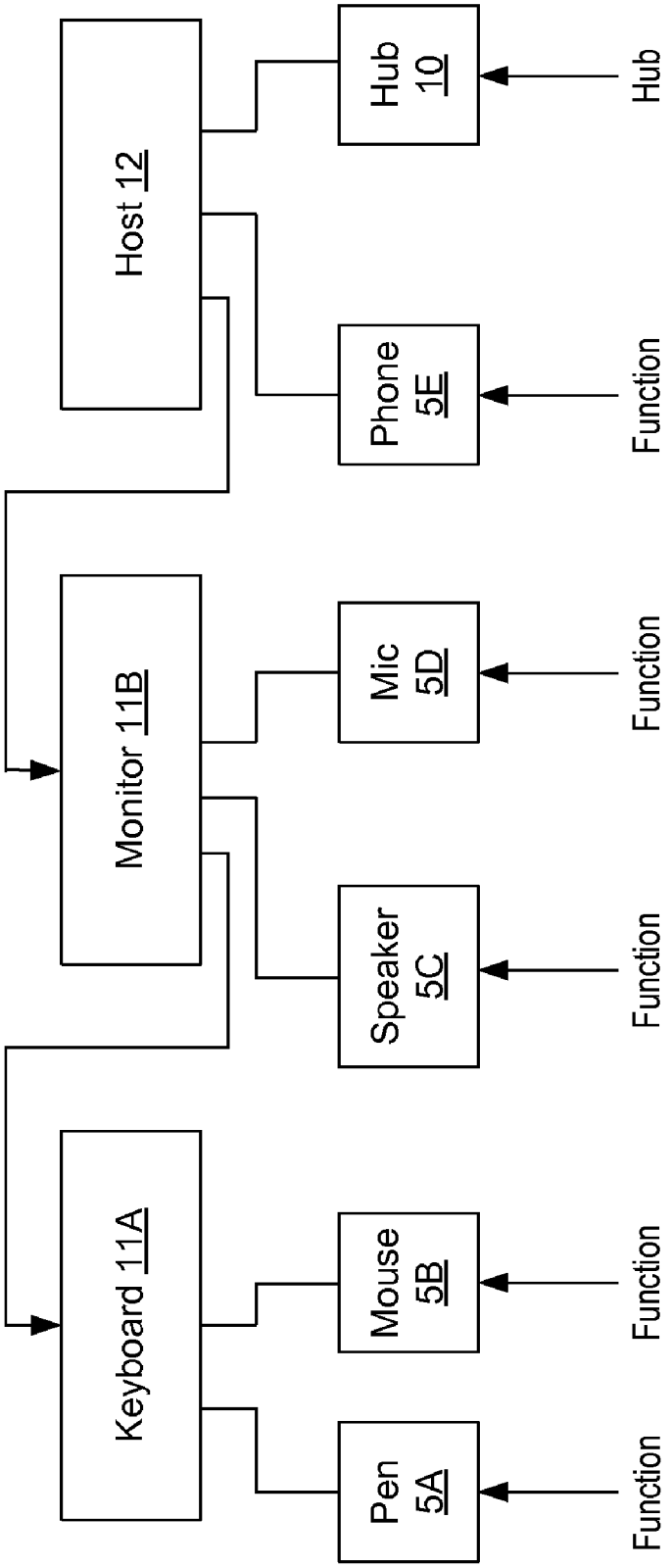

Host 12
Hub 10
Hub
Phone 5E
Function
Monitor 11B
Mic 5D
Function
Speaker 5C
Function
Keyboard 11A
Mouse 5B
Function
Pen 5A
Function

FIG. 4

UNIVERSAL SERIAL BUS HUB WITH SHARED HIGH SPEED HANDLER

CONTINUATION DATA

[0001] This application is a continuation of application Ser. No. 11/223,570 titled "Universal Serial Bus Hub With Shared High Speed Handler Implementing Respective Downstream Transfer Rates" and filed on Sep. 9, 2005, which is a divisional of application Ser. No. 10/374,852 titled "Universal Serial Bus Hub With Shared High Speed Handler" and filed on Feb. 24, 2003, whose inventor is Piotr Szabelski, both of which are hereby incorporated by reference in their entirety as though fully and completely set forth herein.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] This invention relates to computer systems, and more particularly, to universal serial bus hubs used in computer systems.

[0004] 2. Description of the Related Art

[0005] Components in computer systems communicate over various buses. One popular type of bus is the Universal Serial Bus (USB). The USB is a cable bus that allows a host computer to exchange data with a range of peripheral devices. USB peripherals share USB bandwidth through a host-scheduled, token-based protocol. A USB allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

[0006] USB hubs allow multiple peripherals to be attached at a single host attachment point. Thus, a hub converts a single host attachment point into multiple peripheral attachment points. Each attachment point is referred to as a port. A hub typically includes an upstream port, which couples the hub to the host, and several downstream ports, which each couple the hub to another hub or peripheral. Each downstream port may be individually enabled and attached to a high-, full-, or low-speed device.

[0007] A USB hub typically includes a hub controller, a hub repeater, and a transaction translator. The hub repeater provides a USB protocol-controlled switch between the upstream port and downstream ports as well as support for reset and suspend/resume signaling. The host controller facilitates communication to and from the host. The transaction translator allows full- and/or low-speed downstream devices to communicate with a high-speed host. Typically, the number of transaction translators included in a USB hub limits the number of simultaneous transfers that can take place to full- and/or low-speed downstream devices used in a system with a high-speed host.

SUMMARY

[0008] Various embodiments of a method and apparatus for sharing a single upstream data handler between multiple downstream data handlers in a transaction translator for use in a USB hub are disclosed. In one embodiment, a device may include an upstream port and several downstream ports configured to transfer data at a different data transfer rate than the upstream port. The device may also include several downstream data handlers, each coupled to a respective one of the downstream ports, and an upstream data handler coupled to the upstream port. The data handlers are configured to implement a USB protocol. The upstream data handler is configured to provide data received via the upstream port to each of the downstream data handlers. Accordingly, the upstream data handler is shared between the various downstream data handlers.

[0009] In some embodiments, a method may involve: an upstream port of a USB (Universal Serial Bus) hub operating at a different transfer rate than each of several downstream ports of the USB hub; an upstream handler associated with the upstream port providing data received via the upstream port to each of several downstream handlers, where each of the downstream handlers is associated with a respective one of the downstream ports; and each of the downstream handlers providing data to a respective one of the downstream ports for output.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Other aspects of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

[0011] FIG. **1** shows a block diagram of a USB hub, according to one embodiment.

[0012] FIG. **2**A is a block diagram of a transaction translator including multiple downstream data handlers that share a single upstream data handler, according to one embodiment.

[0013] FIG. **2**B is a block diagram of a transaction translator including multiple downstream data handlers that share a single upstream data handler and a single memory device, according to one embodiment.

[0014] FIG. **3** is a flowchart of one embodiment of a method of operating a USB hub that includes multiple transaction translators that share memory.

[0015] FIG. **4** is a block diagram of a system that includes one or more USB hubs.

[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 herein be described in detail. It should be understood, however, that the drawings and description thereto are not intended to limit the invention to the particular form disclosed, but, on the contrary, the invention is to cover all modifications, equivalents, and alternatives falling with the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF EMBODIMENTS

[0017] A USB (Universal Serial Bus) hub may include transaction translator functionality to translate data streams for transfer between ports operating at different rates. When data is being transferred between ports operating at the same rate, the data handling devices may be inactive. The transaction translator may include an independent data handler for each downstream port. A single data handler at each upstream port may transfer data to and from each of the independent downstream data handlers.

[0018] A USB hub having a transaction translator that includes a single high speed handler and multiple full- and/or low-speed handlers may be used to couple various devices within a computer system. For example, a hub may couple a host to one or more devices such as: human interface devices such as mice, keyboards, tablets, digital pens, and game controllers; imaging devices such as printers, scanners, and cameras; mass storage devices such as CD-ROM drives, floppy disk drives, and DVD drives; and other hubs. An exemplary USB hub that implements a USB protocol is described with respect to FIGS. **1-4** herein.

[0019] FIG. **1** shows a block diagram of a USB hub **10**, according to one embodiment. As shown, the USB hub **10** includes an upstream (e.g., facing toward a host) port and four downstream (e.g., facing away from a host) ports. Note that the number of ports may vary among embodiments. Each port is coupled to a physical layer device (PHY). Upstream PHY **12** couples the upstream port to the hub controller **14**. Downstream PHYs **16**A-**16**D (collectively, PHYs **16**) couple a respective downstream port to transaction translator **20**. When the upstream port is operating at high speed and a destination downstream port is operating at full or low speed, hub controller **14** may receive a high-speed data stream from upstream PHY **12** and provide the data to transaction translator **20**. USB hub **10** may also handle transfers from downstream PHYs **16** to upstream PHY **12** by having transaction translator **20** transform a low- or full-speed data stream received via a downstream PHY **16** into a high-speed data stream for transmission via upstream PHY **12**. Each port is an example of a means for receiving a serial data stream.

[0020] Note that the illustrated embodiment shows portions of a hub **10** configuration needed to allow a high-speed upstream device to communicate with one or more full- and/or low-speed downstream devices through the use of transaction translator **20**. Hub **10** may also support communication between high-speed upstream devices and high-speed downstream devices and/or between full- and/or low-speed upstream devices and full- and/or low-speed downstream devices (e.g., via direct connection of the upstream PHY and downstream PHYs). The transaction translator **20** may be inactive if the upstream and downstream devices are operating at the same rate.

[0021] The number of downstream data handlers within transaction translator **20** may determine how many of the downstream PHYs **16** are able to transfer data at substantially the same time. For example, if there are four downstream data handlers within transaction translator **20**, each PHY **16** may be able to transfer data at substantially the same time as the other PHYs **16** are transferring data.

[0022] FIG. **2**A shows a block diagram of a transaction translator **20**, according to one embodiment. The transaction translator includes a data handler **22** or **24** for each port. Instead of including an independent upstream data handler **22** for each downstream data handler **24**, the upstream data handler **22** is shared between the downstream data handlers **24**. By providing an independent data handler **24** for each downstream port, data may be transferred via each downstream port at substantially the same time. Note that in embodiments where multiple upstream ports are implemented, a separate high-speed handler **22** may be implemented for each upstream port. In such embodiments, each high-speed handler **22** may be shared between several downstream handlers **24**.

[0023] When a the upstream port is operating at high speed and the downstream ports are operating at full- and/or low-speed, transaction translator **20** may translate data streams between the different transfer rates. In the illustrated embodiment, if a high-speed data stream is being provided to transaction translator **20** via the upstream port, the high-speed handler may store the data into the memory device **30**A-**30**D coupled to the destination full- and/or low-speed handler **24**. For example, if the transaction translator **20** receives a high-speed data stream to be transferred to a low-speed device via the port coupled to downstream handler **24**B, the upstream data handler **22** may store data received in that data stream in memory device **30**B at a rate substantially similar to the rate at which the data is received. The downstream data handler **24**B at the destination port may then read the data out of memory device **30**B at a rate substantially similar to the rate at which data is transferred from the destination downstream port.

[0024] When a downstream device operating at full- or low-speed is sending data to an upstream device operating at high-speed, the data may be received via one of the downstream ports for transmission via the upstream port. For example, the data may be received via the downstream port coupled to data handler **24**C. Data handler **24**C may store the received data in memory device **30**C. Data handler **22** may then output the data from memory device **30**C at the higher rate via the upstream port. Other downstream data handlers **24** may operate similarly. Due to the inclusion of multiple downstream data handlers, each downstream data handler **24** may be receiving data from a downstream device at substantially the same time as another downstream data handler.

[0025] FIG. **2**B illustrates a block diagram of a transaction translator **20**, according to another embodiment. In FIG. **2**B, a single high-speed handler **22** is shared between several downstream handlers **24**. The high-speed handler **22** is configured to send and receive a high-speed data stream via the upstream port.

[0026] In some embodiments, the transaction translator **20** may include a shared memory device **30** that is shared between the downstream data handlers **24**, as shown in FIG. **2**B. Each handler **22** and **24** is configured to send requests to access shared memory device **30** to data buffer controller **26**. In the illustrated embodiment, shared memory device **30** is a single-ported memory device, and thus the high- and full- and/or low-speed handlers arbitrate for access to the shared memory device. Data buffer controller **26** is configured to arbitrate between the handlers' requests to determine which handler's request to provide to the shared memory device **30**. Data buffer controller **26** may additionally perform address remapping on at least some of the handlers' requests in some embodiments. Note that in other embodiments, the shared memory device **30** may have more than one port, thus allowing more than one data handler to access the shared memory device at substantially the same time.

[0027] Each handler **22** and **24** includes buffers **32** to store data being transferred to or from shared memory device **30** prior to transmitting that data to another handler or subsequent to receiving that data from one of the hub's ports. For example, high-speed handler **22** is configured to receive a high-speed stream of data via the upstream port. Portions of the received data may be temporarily buffered in buffer **32**E while high-speed handler **22** arbitrates for access to shared memory **30**. When access is granted, high-speed handler **22** transfers the buffered data to shared memory **30**. In many embodiments, buffer **32**E may include two independently accessible buffers so that incoming data can be stored in one buffer while data is written to shared memory device **30** from the other buffer area. High-speed handler **22** may also transmit information to the full- and/or low-speed handler **24** that the data stream is being transmitted to indicating the location of the data to be handled by that full- and/or low-speed handler. Alternatively, different portions of the shared memory **30** may be allocated to each full- and/or low-speed handler **24**, allowing the high-speed handler **22** to indicate which handler **24** is the recipient of the data stream by writing the data into the portion of the shared memory **30** allocated to that handler.

[0028] When a data stream received by one of the full- and/or low-speed handlers **24** is being output by high-speed handler **22**, the receiving full- and/or low-speed handler **24** may transmit information to the high-speed handler **22** indicating the location of the data in shared memory device **30**. High-speed handler **22** may then arbitrate for access to shared memory device **30** and store a portion of the data in buffer **32**E for transfer at the high-speed rate to the upstream port. As when high-speed handler **22** is receiving a high-speed data stream, the buffer **32**E may include two independently accessible buffer areas so that data can be transferred to the upstream port from one buffer area while the other buffer area is being loaded with more data from shared memory device **30**. Note that in other embodiments, buffer **32**E may be a dual-ported device so that data can be transferred into and/or out of the buffer for transfers via the upstream port at substantially the same time as data is also being transferred to and/or from shared memory device **30**. As mentioned previously, there may be more than two independently accessible buffers in buffer **32**E. The size of each buffer in buffer **32**E may be the same as (or greater than) the amount of data accessible in shared memory device **30** by a single access request in some embodiments. In some embodiments, the size of the buffers **32**E in the high-speed handler **22** may be larger than the size of buffers **32**A-**32**D in the full- and/or low speed handlers **24**.

[0029] Full- and/or low-speed handlers **24**A-**24**D may each use their respective buffers **32**A-**32**D in much the same way as high speed handler **22** when sending and receiving data via a respective downstream port.

[0030] FIG. **3** is a flowchart of one embodiment of a method of operating a serial bus hub that uses a shared upstream handler to transfer data to several downstream data handlers. Such a hub may be used to transfer data between connections that are operating at different rates. At **501**, an upstream port of a USB (Universal Serial Bus) hub operates at a different transfer rate than each of several downstream ports of the USB hub. The upstream port may receive data from a host to be transferred to destination devices via the downstream ports. For example, an upstream port of the hub may receive data from the host in several different data transfers. Each transfer may involve data to be transferred to a different downstream port.

[0031] At **503**, an upstream data handler associated with the upstream port provides data to each of the downstream handlers. For example, the upstream data handler may provide data received in one transfer from the host to one downstream handler and a data received in another transfer from the host to another downstream handler. The upstream handler may provide the data to the various downstream handlers by storing the data in various memory devices associated with the downstream handlers. Each of the downstream handlers is associated with a respective one of the USB hub's downstream ports. In response to receiving the data from the upstream data handler, each of the downstream handlers provides data to a respective one of the downstream ports, as indicated at **505**. The data is then output at the downstream ports' transfer rate, which differs from the transfer rate of the upstream port.

USB Protocol

[0032] In many embodiments, a serial hub may be configured to implement the USB protocol, which defines a polled bus on which a host may initiate data transfers. Typical USB transactions involve several packets. The host initiates a transaction by sending a packet indicating the type and direction (upstream or downstream) of the transaction being initiated, the address of the target device, and an endpoint. If a downstream transfer is requested, the target device receives data transferred from the host. Similarly, if an upstream transfer is requested, the target device sends data to the host. A handshake packet may then be sent to the host from the target device to indicate whether the transfer was successful. The USB protocol describes the transfer between a source or destination on the host and an endpoint on a device as a pipe. Pipes may be either stream pipes or message pipes. Data transferred via a stream pipe has no USB-defined structure, unlike data transferred via a message pipe. Different pipes may have different bandwidths, speeds, and endpoint characteristics (e.g., sink or source, buffer size, etc.) and be used to transfer packets of different sizes.

[0033] FIG. **4** illustrates an exemplary computer system that may include one or more USB hubs **10** as described above. In the embodiment illustrated in FIG. **4**, a hub included within host **12** couples directly to hub **10**, phone **5**E, and monitor **11**B. Monitor **11**B includes another hub, which couples directly to microphone **5**D, speaker **5**C and keyboard **11**A. Keyboard **11**A includes yet another hub, which couples directly to mouse **5**B and pen **5**A.

[0034] Any and/or all of the hubs shown in FIG. **4** may be implemented similarly to those described above. Typically, some of the hubs will connect functions operating at the same rate while other hubs will connect functions operating at different rates. Whenever a high-speed function communicates with a non-high-speed function via a hub, transaction translators included in the hub may be used to convert data streams between the different rates. Such transaction translators may share a memory device, as described above. Note that non-USB embodiments of a serial hub may be included in similar computer systems.

[0035] As shown in FIG. **4**, several devices in a computer system may be coupled to a host by various USB connections. A device that is configured to transmit and/or receive data and/or control information over a USB connection may be referred to as a function. Functions are typically implemented as separate peripheral devices that connect to a USB connection, which in turn plugs into a port on a hub. In FIG. **4**, exemplary functions include pen **5**A, mouse **5**B, speaker **5**C, microphone **5**D, and phone **5**E. Some devices, referred to as compound devices, may be implemented in a single physical package that includes one or more functions and/or a hub. Exemplary compound devices in FIG. **4** include keyboard **11**A and monitor **11**B. All of these functions are coupled to host **12**, which may also include a hub that allows the various functions to communicate with the host processor. An additional hub **10** may be coupled to the host in order to provide additional connectivity for other devices (e.g., cameras, printers, scanners, etc.).

[0036] 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.

What is claimed is:

1. A device, comprising:

an upstream port configured to operate at a first data transfer rate;

a plurality of downstream ports configured to operate at a different data transfer rate than the first data transfer rate;

an upstream data handler coupled to the upstream port and configured to store specific transactions received through the upstream port, wherein the specific transactions comprise data;

a plurality of downstream data handlers, wherein each respective downstream data handler of the plurality of downstream data handlers is:

associated with a respective downstream port of the plurality of downstream ports;

operable to access respective transactions of the stored specific transactions, wherein the respective transactions are intended for the respective downstream port associated with the respective downstream data handler; and

transmit to its associated respective downstream port, at the different data transfer rate, the data comprised in its respective transactions;

wherein the specific transactions comprise transactions defined by USB (Universal Serial Bus) protocol.

2. The device of claim **1**, wherein the upstream data handler is configured to accept the specific transactions through the upstream port at the first data rate.

3. The device of claim **1**, wherein the upstream data handler is configured to store the data comprised in the specific transactions in a plurality of memory devices; and

wherein each of the plurality of memory devices is associated with a respective one of the plurality of downstream data handlers.

4. The device of claim **1**, wherein the upstream data handler is configured to store the data comprised in the specific transactions in a shared memory device; and

wherein each respective downstream data handler is configured to retrieve the data comprised in its respective transactions from the shared memory device for output via its associated respective downstream port.

5. The device of claim **1**, wherein each respective downstream data handler is configured to provide data to its associated respective downstream port substantially simultaneously with another one of the plurality of downstream data handlers providing data to its associated respective downstream port.

6. The device of claim **1**, wherein the specific transactions comprise USB high-speed split transactions.

7. The device of claim **6**, wherein the high-speed split transactions comprise one or more of:

start-split transactions; or

complete-split transactions.

8. The device of claim **7**, wherein the upstream data handler is configured to accept the start-split transactions and respond to the complete-split transactions.

9. The device of claim **1**, wherein the upstream data handler is a USB high-speed handler, and each respective downstream data handler is a full/low-speed handler.

10. The device of claim **1**, further comprising:

an upstream physical layer device coupled to the upstream port;

a hub controller coupled between the upstream physical layer device and the upstream data handler; and

a plurality of downstream physical layer devices coupled between the plurality of downstream data handlers and the plurality of downstream ports.

11. A method, comprising:

a plurality of downstream ports of a USB (Universal Serial Bus) hub operating at a different transfer rate than an upstream port of the USB hub;

an upstream handler associated with the upstream port accepting through the upstream port, at the first data rate, specific transactions comprising data, and storing the specific transactions;

each respective downstream handler of a plurality of downstream handlers accessing respective transactions of the stored specific transactions, wherein each respective downstream handler is associated with a respective downstream port of the plurality of downstream ports, and wherein the respective transactions are intended for the respective downstream port; and

each respective downstream handler transmitting to its associated respective downstream port, at the different data transfer rate, the data comprised in its respective transactions.

12. The method of claim **11**, wherein said storing the specific transactions comprises storing the specific transactions in a plurality of memory devices; and wherein each of the plurality of memory devices is associated with a respective downstream handler.

13. The method of claim **11**, wherein said storing the specific transactions comprises storing the specific transactions in a shared memory device, the method further comprising:

each respective downstream handler retrieving the data comprised in its respective transactions from the shared memory device.

14. The method of claim **13**, wherein said each respective downstream handler transmitting to its associated respective downstream port includes the plurality of downstream data handlers transmitting to their associated respective ones of the plurality of downstream ports simultaneously.

15. A system, comprising:

a shared memory device;

a faster handler configured to store in the shared memory device specific transactions intended for a plurality of slower ports and received through a faster port;

a plurality of slower handlers each coupled to the shared memory device to access respective ones of the stored specific transactions, and transfer data comprised in the respective ones of the stored specific transactions to a respective one of the plurality of slower ports, wherein the plurality of slower ports have respective lower data transfer rates than the faster port;

wherein the faster handler and each of at least two of the slower handlers are conjunctively operable to use the shared memory device to convert from one respective data transfer rate to another respective data transfer rate; and

wherein the faster handler and the plurality of slower handlers are configured to implement a USB (Universal Serial Bus) protocol.

16. The system of claim **15**, wherein the faster handler includes at least two buffers, wherein the faster handler is configured to transfer data between the faster port and one of the at least two buffers while also transferring data between the shared memory device and a different one of the at least two buffers.

17. The system of claim **16**, wherein a capacity of each of the at least two buffers is equal to an amount of data accessible in the shared memory device in response to a single request initiated by the faster handler.

18. The system of claim **15**, further comprising:

a memory arbiter coupled to the shared memory device, the faster handler, and the plurality of slower handlers and configured to arbitrate between requests to access the shared memory device generated by the faster handler and the plurality of slower handlers.

19. The system of claim **18**, wherein the memory arbiter is configured to allow the faster handler to access the shared memory device more frequently than any of the plurality of slower handlers access the shared memory device.

20. The system of claim **18**, wherein the memory arbiter is configured to allow the faster handler to access the shared memory device at least every other arbitration cycle in the memory arbiter.

21. The system of claim **18**, wherein there are N slower handlers in the plurality of slower handlers, and wherein the memory arbiter is configured to allow one of the plurality of slower handlers to access the shared memory device at least every 2N arbitration cycles in the memory arbiter, wherein N is a positive integer.

22. The system of claim **18**, wherein the memory arbiter is configured to arbitrate between the plurality of slower handlers on a round-robin basis.

23. The system of claim **18**, wherein the memory arbiter is configured to map data written by the faster handler into a region of the shared memory device corresponding to one of the plurality of slower handlers to which the data is being transferred.

24. The system of claim **18**, wherein the memory arbiter is configured to map data written by one of the plurality of slower handlers into a region of the shared memory device corresponding to that one of the slower handlers.

25. The system of claim **15**, wherein the shared memory device is a single-ported memory device.

26. A method, comprising:

a plurality of ports in a USB (Universal Serial Bus) hub each receiving a respective data stream, wherein at least one of the plurality of ports receives its respective data stream at a different rate than other ones of the plurality of ports, wherein a respective one of a plurality of handlers is associated with each of the plurality of ports, wherein the plurality of handlers are configured to implement a USB protocol;

arbitrating between the plurality of handlers for access to a shared memory device;

a first handler, associated with a port of the at least one of the plurality of ports, storing, in response to being selected by said arbitrating, to the shared memory device specific transactions included in the respective data stream received by the port of the at least one of the plurality of ports;

each of at least two handlers associated with respective ports of the other ones of the plurality of ports accessing the specific transactions and transmitting data comprised in the specific transactions to the respective ports of the other ones of the plurality of ports; and

wherein the first handler and each of the at least two handlers associated with respective ports of the other ones of the plurality of ports are conjunctively operable to convert from one respective data transfer rate to another respective data transfer rate.

27. The method of claim **26**, wherein the first handler is a faster handler and the port of the at least one of the plurality of ports is a faster port of the plurality of ports, wherein the faster handler includes at least two buffers, wherein the faster handler is configured to transfer data between the faster port and one of the at least two buffers while also transferring data between the shared memory device and a different one of the at least two buffers.

28. The method of claim **27**, wherein a capacity of each of the at least two buffers is equal to an amount of data accessible in the shared memory device in response to a single request initiated by the faster handler.

29. The method of claim **26**, wherein the first handler is a faster handler associated with a faster port of the plurality of ports, wherein said arbitrating allows the faster handler to access the shared memory device more frequently than any other handlers of the plurality of handlers.

30. The method of claim **26**, wherein the shared memory device is a single-ported memory device.

31. The method of claim **26**, wherein the specific transactions are high-speed split transactions.

* * * * *