# LOW POWER AND LOW PIN COUNT BI-DIRECTIONAL DUAL DATA RATE DEVICE INTERCONNECT INTERFACE

## BACKGROUND

Field of the Invention

**[0002]** The present invention relates generally to digital device interfaces, and more specifically to low pin count interfaces to interconnect USB devices on a circuit board.

Description of the Related Art

**[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, 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 (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 (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]** Embedded and portable products are often implemented with a USB interface because of their popularity, driver support, interoperability and relative low cost of USB devices. However, standard USB devices include an analog physical layer (PHY) component, along with pull-up and pull-down resistors that constantly consume power (even when in suspend or standby).

These aspects of USB make it less attractive to power conscious embedded devices, particularly those operating from a battery. It has therefore become desirable to provide USB connectivity, without the added power consumption of analog PHYs and pull-up and pull-down resistors.

**[0006]** The USB-IF (USB Implementers Forum) created and released an Interchip 1.0 specification addressing some of the analog PHY issues of USB, and is therefore more attractive for portable devices, but the interface is not capable of HS (high speed) USB transfer speeds, and retains the differential data (D+/D-) aspect of USB, which typically requires eye diagrams, clock recovery and synchronization. This increases the complexity of the implementation over a straightforward digital clock and data interface, and limits the transmission length to 10 cm or less. It also makes it very difficult to carry this scheme to 480Mb/s performance.

**[0007]** Various interface solutions have been implemented and proposed in the past. For example, US Patent 4,785,396 defines use of Manchester encoding over differential data lines. This requires clock recovery and is very similar to Interchip USB in terms of its limitations. It is also constructed to connect multiple devices together, instead of the point-to-point structure inherent to USB. It uses pull-ups to indicate idle state, and uses other "static" states to indicate collision detection and valid data. It doesn't have the ability to communicate USB Reset, or USB Resume without an intervening protocol, which would unnecessarily burden the implementation. Also, the defined arbitration mechanism can be deemed unnecessary since it is already defined in USB upper level protocol.

**[0008]** US Patent 5,7903,993 defines a separate interrupt line to allow "slave" devices to operate on a bus with the single master. It also defines a command sequence, which is a packet protocol that would essentially act as a header to all USB traffic, or would need to be used to transmit control information such as a USB Reset. This would not work well in a USB environment without substantial modification to the USB Device itself.

**[0009]** US Patent 4,847,867 defines use of a bidirectional data line, but the master sources the clock at all times. The protocol uses the rising edge of the clock to send and the falling edge to receive data.

**[0010]** US Patent 6,073,186 defines use of a bidirectional data line, but the master sources the clock at all times. This would not provide a robust solution with a dual data rate scheme over longer circuit trace lengths. It would therefore not be viable for certain applications. Also, no provision is made for static signaling such as USB Reset and Resume.

**[0011]** US Patent 7,003,585 B2 defines use of a transceiver scheme for clock extraction, and defines a transport protocol for the interface that is redundant in a USB environment. This is not

a digital replacement serial bus for USB, and would therefore not meet the requirements for a portable device. It is designed mainly for long cable runs (an example of 75' is given in the background section).

**[0012]** US Patent 5,970,096 describes a bus structure between one master and one or many slave devices. It describes an idle condition using the static condition of the interface, but no provision is made for USB Reset or Resume through static states on the bus.

**[0013]** US Patent Application US 2002/0095541 A1 describes an analog serial data transfer scheme and doesn't apply to digital serial transfers.

**[0014]** US Patent Application US 2004/0263321 A1 describes a serial data transmission system that can also supply power from a transmission source to a receiver. It focuses on power delivery and doesn't provide static states for USB Reset or USB Resume. It also wouldn't map successfully to a simple two-wire scheme to replace standard USB.

**[0015]** 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

**[0016]** Various embodiments of the present invention provide for USB connectivity without the added power consumption of analog PHYs and pull-up and pull-down resistors. Some embodiments comprise a simple low-pin count interface to interconnect USB devices on a circuit board. In one set of embodiments, a 2-wire digital bus may be configured to interconnect USB devices and/or controllers without using a USB PHY or USB cabling. The interconnected devices and/or controllers may typically be non-removable, but may be – in certain embodiments – removable modules such as smart cards or expansion "daughter" boards. In certain embodiments, the simple low pin count interface is not used as a plug-and-play replacement for standard USB, or IEEE1394, where peripherals are attached and/or detached routinely.

**[0017]** One embodiment comprises a 2-wire bidirectional bimodal bus that allows simultaneous host and device connectivity over the same interface, allowing a USB device, e.g. a cell phone, to appear as a USB device to an external PC, while also appearing as a USB host to a connected peripheral, such as a card reader. In one set of embodiments, the 2-wire bidirectional bimodal bus may be a 2-wire serial 240Mhz DDR interconnect with one clock line, and one data line. The clock and data may only transition during an active transfer. The data may be clocked on rising and falling edges of the clock, thus a 480Mbit USB High-Speed data rate can be maintained. The clock may be source synchronous with the data. This may simplify the design

of both the IC's and printed circuit boards comprised in systems configured with the 2-wire bidirectional bimodal bus.

**[0018]** Static states, or no active data transmission states may be configured to allow for the following USB functionality:

**[0019]** USB Idle (CLOCK line high, DATA line low) for 1 or more clock periods;

**[0020]** USB Attach or Resume (CLOCK line low, DATA line high), for 1ms, to match current USB specification requirements;

**[0021]** USB Suspend, identical to idle state, but the time period may be 3ms to match current USB specification requirements; and

**[0022]** USB Reset (CLOCK line low, DATA line low) for 2.5μs or 10ms to match current USB specification requirements.

**[0023]** In general, various embodiments of the invention may combine a simple clock source synchronous DDR data transfer mechanism with static bus state signaling to replace a complex bus like USB with an easy to implement digital serial interconnect bus. This may eliminate the various pull-up and pull-down resistors required in USB, and enable the interconnect bus to operate with little or no leakage current when the bus is in an idle state, or data transmission state. All of the required functionality may be implemented using only two signal pins. The interconnect bus may also enable silicon solutions for HS USB that do not require a PLL, since the clock may be provided by the transmission source and may thus not need to be recovered from the serial data stream. Various embodiments of the invention may provide an easy reuse mechanism for USB silicon by enabling a designer to remove the analog PHY and replace it with a serial digital I/O transfer mechanism, while retaining the IP's USB timers, and other protocol specific features.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0024]** 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:

**[0025]** Figure 1 is a block diagram of one embodiment in which a 2-wire bidirectional bimodal bus is used to couple a system on a chip (SOC) and a card reader (CR) to a USB hub;

**[0026]** Figure 2 shows a sample timing diagram illustrating the data transfer operation of the 2-wire bidirectional bimodal bus; and

**[0027]** Figure 3 shows a sample timing diagram illustrating the data transfer, USB suspend, and USB resume operations of the 2-wire bidirectional bimodal bus.

**[0028]** 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 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 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

**[0029]** In one set of embodiments, a 2-wire (or at least 2-wire) digital bus may be configured to interconnect USB devices and controllers without using a USB physical layer (PHY) or USB cable. One wire may be a bidirectional clock (CLK) line/signal and the other wire may be a bidirectional data line/signal. In certain embodiments, the clock signal may only be active during a data transfer, and data may be clocked on rising and falling edges of the clock, thereby implementing a double data rate (DDR) transfer protocol. The transmitting device may source the CLK signal simultaneously and synchronously with the data, while the receiving device may clock the data during transitions of the clock line. In one embodiment, the bus may comprise one or more double data rate (DDR) data lines, which may all be source synchronous with the CLK line. Voltage levels may be process/device specific. Figure 1 illustrates a block diagram of an exemplary system in which a 2-wire bidirectional bimodal bus 110 is used to couple a system on a chip (SOC) 102 and a card reader (CR) 108 to a USB hub 104. A personal computing (PC) system 106 may also be coupled to USB hub 104 through a standard USB connection 112.

**[0030]** In one set of embodiments, SOC 102 (which may be comprised in a cell phone) may be docked to PC 106 through HUB 104 as a peripheral USB device, while also operating as a USB host to another connected peripheral USB device, in this case CR 108. By coupling CR 108 to USB hub 104 via 2-wire bidirectional bimodal bus 110, and also coupling SOC 102 to USB hub 104 via 2-wire bidirectional bimodal bus 110, CR 108 may be recognized as a peripheral USB device by both PC 106 and SOC 102, while SOC 102 may be recognized as a peripheral USB device by PC 106, and as a USB host by CR 108.

**[0031]** As stated above, 2-wire bidirectional bimodal bus (or interface) 110 may comprise a CLK line and a data line, where a combination of states of the CLK line and the data line may be configured to implement select USB functionality to allow USB connectivity via the 2-wire bus. For example, the USB Idle state may be implemented by holding the CLK line in a high state while holding the data line in a low state. Similarly, the USB Reset state may be implemented by holding both the CLK line and the data line in a low state, and the USB Resume/Connect state may be implemented by holding the CLK line in a low state while holding data line in a high state. USB Data Transmit/Receive operations may be achieved by using the CLK signal to clock data transmitted over the data line.

**[0032]** In order to insure compliance with USB 2.0 requirements, the required effective data transfer rate of 480Mb/s may be achieved by implementing 2-wire bus 110 as a DDR bus with the CLK signal driven at 240MHz. No bit stuffing may be required, and by maintaining USB Reset/Resume timing, existing internal timers may be re-used on a USB function that normally connects to a standard PHY. The CLK signal and data signal may be generated as source synchronous signals, making any data rate possible as required. For example, in one set of embodiments the CLK signal may be driven at 533MHz, which, in case of a DDR transfer configuration, may achieve an effective data rate of 1066Mb/s. Interfaces for 2-wire bus 110 may be implemented with standard DDR capable input/output (I/O) pads, using spread spectrum clock/data lines to reduce spikes due to electromagnetic interference (EMI). Source synchronous data and clock signals would both spread synchronously.

**[0033]** In some embodiments, a wide frequency tolerance may allow for implementing the CLK signal using simpler and/or easier mechanisms than a phase-locked loop (PLL), using, for example, a ring oscillator instead. For example, for USB 2.0, a 400-480 Mb/s rate may be achieved with a 200MHz – 240MHz clock tolerance for a DDR transfer mechanism. The specified protocol for 2-wire bus 110 may especially be applicable to the upper USB protocol, where error correction code (ECC), data corruption and data arbitration are handled (among others), since only four basic states may be defined, namely USB Idle, USB Reset, USB Resume/Connect and USB Data Transfer. In preferred embodiments, 2-wire bus 110 may generally be used for point-to-point interconnect, e.g. host to device, and may remain idle when no data is transmitted.

**[0034]** Fig. 2 shows a sample timing diagram illustrating the data transfer operation of the 2-wire bus, with CLK line 202 and Data line 204 transitioning from an Idle state, to a data transfer state, and back to an idle state at the end of the transfer. As shown in the diagram, the data may

be clocked on both the rising and falling edges of the clock, implementing a DDR transfer protocol. When the data transmission is complete, the bus may return to an idle state. In typical USB system topology, a host coupling to a device generally operates as a master device and the device operates as a slave device, but both the host and the device are capable of driving signal states or data onto the bus. The Host device may place weak keepers (a weak pull-up on the clock line, and a weak pull-down on the data line) on the bus when the bus is in the idle state, and may remove the keepers when a data transfer occurs. When data is driven onto the bus, the transmitting device may also drive the clock signal synchronously with the data.

**[0035]** As previously mentioned, in addition to Idle and Data transmission states, 2-wire bus 110 may be configured to also support USB Reset, USB Resume, and USB Suspend. These states may be implemented as driven states, and may overdrive the weak keepers, permitting a USB device to send a Wake-up signal to a suspended USB host, thereby precipitating a wake event. It should be noted that a USB host may similarly send out a Resume signal to a suspended USB device, waking up the bus in a similar manner. Thus, a Resume signal and a Wake-up signal may be considered to be functionally identical, with the former indicating a host driven event and the latter indicating a device driven event. Figure 3 shows a sample timing diagram illustrating the USB Reset, USB Suspend, and USB Resume operations of the 2-wire bus in addition to the Idle and Data transmission states. Similar to Fig. 1, 2-wire bus 110 may be in the Idle state while CLK line 202 is held high and Data line 204 is held low for one or more clock periods. Once the end of the Idle state is reached, data may be sampled at both the rising edges and falling edges of CLK signal 202. Following the last sampled data, the Idle state may again be entered by holding CLK line 202 high and Data line 204 low. A USB Suspend mode may be entered by extending to 3ms the time period during which CLK line 202 is held high and Data line 204 is held low, to match current USB specification requirements. USB Attach or Resume mode may be entered by transitioning CLK line 202 from high to low, and transitioning Data line 204 from low to high, maintaining the new states for at least 1ms to match current USB specification requirements. As also illustrated in the timing diagram shown in Fig. 3, USB Reset may be actuated by holding both CLK line 202 and Data line 204 low for 2.5μs or 10ms, to match current USB specification requirements. It should be noted that the respective timing periods during which the state of CLK line 202 and the state of Data line 204 are held constant for respectively conveying USB Attach/Resume state, USB Suspend state and/or USB Reset state, may be shortened for performance enhancement.

**[0036]** In general, a 2-wire (minimum configuration) interconnect bus may be configured to have command/control state information transmitted over the same line(s) over which data is transmitted, without driving a signal (i.e. using static states.) The interconnect bus may be used for protocols other than USB, as long as the protocol is designed to prevent two sources from writing to the bus at the same time or otherwise prevent bus contention through an inherent/built-in handshake mechanism, if the interconnect bus does not have a request/grant structure. Status information may be conveyed on the interconnect bus by line states being held for certain specified time periods. In other words, static states held for certain time periods may convey information in the absence of a clock and/or other edge information.

**[0037]** In some embodiments, the interconnect bus interface may be implemented using standard I/O drivers, which may be tri-stated during operation when not being driven. The clock line and the data line may both be bidirectional. Data transfer may be configured according to a DDR scheme (to increase efficiency at lower clock speeds), and the clock frequency may be set to any value required to maintain a desired effective data rate for specific transfer requirements. (i.e. 6MHz for FS USB, 533 MHz to match DDR memory speeds common on today's PC motherboards, etc...). The interconnect bus specifications may thus be extended to USB 3.0 and beyond, by simply increasing the clock frequency. For higher data rates with the same clock source frequency, the number of data lines may also be increased, and any or all of the data lines may be configured as bidirectional data lines. In one set of embodiments, the signaling voltage level used may implementation dependent, and may be set to any level as long as the transmitting device and the receiving device operate at compatible voltage levels. For example, 1.8V signaling may be implemented, but alternative voltage classes, such as 1.2V, may be specified as well.

**[0038]** A device operating as a host when coupled to the interconnect bus may implement weak bus keepers (i.e. apply weak pull-up/pull-down) to maintain a bus idle state when no active transfers are in progress. The host device may temporarily remove the weak keepers while transmitting and/or receiving data. In one set of embodiments this may be accomplished by the device (i.e. the device that is not operating as a host) establishing an Idle state for a specified number (e.g. 2) of clock cycles after final data transmission by driving the clock line high and the data line low – as shown for example in Figs. 2 and 3 – during which time period the device operating as a host may apply the Idle state weak keepers, effectively pulling up the clock line and pulling down the data line. In one set of embodiments, a simple spread spectrum clicking scheme may be used to reduce effects of EMI since the data and clock edges are always

synchronous to each other, making it possible to vary them in the frequency domain without difficulty. Some embodiments may be implemented using slew rate controlled I/O ports to further reduce and/or eliminate effects of EMI.

**[0039]** Further modifications and alternative embodiments of various aspects of the invention may be apparent to those skilled in the art in view of this description. Accordingly, this description is to be construed as illustrative only and is for the purpose of teaching those skilled in the art the general manner of carrying out the invention. It is to be understood that the forms of the invention shown and described herein are to be taken as embodiments. Elements and materials may be substituted for those illustrated and described herein, parts and processes may be reversed, and certain features of the invention may be utilized independently, all as would be apparent to one skilled in the art after having the benefit of this description of the invention. Changes may be made in the elements described herein without departing from the spirit and scope of the invention as described in the following claims.

**WHAT IS CLAIMED IS**:

1. An interconnect bus comprising:

a bidirectional clock line; and

a bidirectional data line operable to carry USB data;

wherein a clock signal transmitted over the clock line is source synchronous with data transmitted over the data line;

wherein the data is sampled at least at a rising edge of the clock signal; and

wherein the interconnect bus is operable to convey USB command/control state information over the clock line and the data line, wherein respective combinations of static states of the clock line and the data line correspond to respective USB command/control state information.

2. The interconnect bus of claim 1, wherein USB command/control state information comprises:

USB Idle state;

USB Attach/Resume state;

USB Suspend state; and

and USB Reset state.

3. The interconnect bus of claim 2;

wherein the USB Idle state is conveyed by holding the clock line high and the data line low for a specified number of clock periods; and/or

wherein the USB Attach/Resume state is conveyed by holding the clock line low and the data line high for a specified first time period; and/or

wherein the USB Suspend state is conveyed by holding the clock line high and the data line low for a specified second time period; and/or

wherein the USB Reset state is conveyed by holding the clock line low and the data line low for a specified third time period.

4. The interconnect bus of claim 3, wherein the specified number of clock periods, the specified first time period, the specified second time period, and the specified third time period satisfy USB specification requirements.

5. The interconnect bus of claim 1, wherein the interface is operable to couple USB devices without using a physical layer (PHY) and/or USB cables.

6. The interconnect bus of claim 1, wherein the data is sampled at the rising edge and/or at a falling edge of the clock signal.

7. The interconnect bus of claim 1, wherein the clock signal is active only during data transfer on the data line.

8. A method comprising:

transmitting USB data between a first USB device and a second USB device over an interconnect bus, wherein the interconnect bus comprises two or more signal lines coupling together the first USB device and the second USB device without using a physical layer (PHY) and/or USB cables; and

conveying USB command/control state information recognized by the first USB device and the second USB device over the interconnect bus without any of the two or more signal lines being driven by clocked signals.

9. The method of claim 8, wherein one of the two or more signal lines is a bidirectional clock line and at least one of remaining ones of the two or more signal lines is a bidirectional data line.

10. The method of claim 9, wherein the first USB device is a USB host device, the method further comprising the first USB device applying weak pull-up/pull-down to maintain a USB Idle state on the interconnect bus when no active data transfer takes place on the data line.

11. The method of claim 10, further comprising the USB host device operable temporarily removing the weak pull-up/pull-down when it is receiving and/or transmitting data on the data line.

12. The method of claim 9, further comprising:

transmitting a clock signal over the clock line; and

sampling data transmitted over the data line at a rising edge and at a falling edge of the clock signal.

13. The method of claim 9, further comprising:

the first USB device and the second USB device each sending and/or receiving data over the data line; and

the first USB device and the second USB device each sending a clock signal over the clock line when performing said sending the data, in order to produce a clock signal that is source synchronous with the data.

14. The method of claim 9, wherein remaining ones of the two or more signal lines comprise a plurality of bidirectional data lines.

15. The method of claim 8, further comprising:

coupling a USB hub between the first USB device and the second USB device via the interconnect bus; and

coupling a USB host device to the USB hub, wherein the first USB device appears as a device to the USB host device and appears as a USB host to the second USB device.

16. The method of claim 15, wherein the second USB device appears as a device to the fist USB device and to the USB host device.

17. The method of claim 15, wherein said coupling the USB host device to the USB hub comprises coupling the USB host device to the USB hub via a standard USB connection.

18. A system comprising:

an interconnect bus comprising:

a bidirectional clock line; and

a bidirectional data line;

wherein a clock signal transmitted over the clock line is source synchronous with data transmitted over the data line;

wherein the data is sampled at least at a rising edge of the clock signal; and

wherein the interconnect bus is operable to transmit data over the data line according to a bus protocol configured with a handshake mechanism, to intrinsically prevent bus contention;

wherein the interconnect bus is operable to convey command/control state information of the bus protocol over the clock line and the data line, wherein respective combinations of static states of the clock line and the data line held for specified time periods correspond to respective command/control state information.

19. The system of claim 18, wherein the interconnect bus further comprises one or more additional data lines.

20. The system of claim 18, wherein the data is sampled at the rising edge and/or at a falling edge of the clock signal.