US 20120066417A1

(19) **United States**

(12) **Patent Application Publication**

Foster

(10) **Pub. No.: US 2012/0066417 A1**

(43) **Pub. Date: Mar. 15, 2012**

(54) **SYNCHRONISATION AND TRIGGER DISTRIBUTION ACROSS INSTRUMENTATION NETWORKS**

(75) Inventor: **Peter Graham Foster**, Parkside (AU)

(73) Assignee: **CHRONOLOGIC PTY. LTD.**, Adelaide, SA (AU)

(21) Appl. No.: **13/320,388**

(22) PCT Filed: **May 20, 2010**

(86) PCT No.: **PCT/AU2010/000608**

§ 371 (c)(1), (2), (4) Date: **Nov. 14, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/179,904, filed on May 20, 2009.

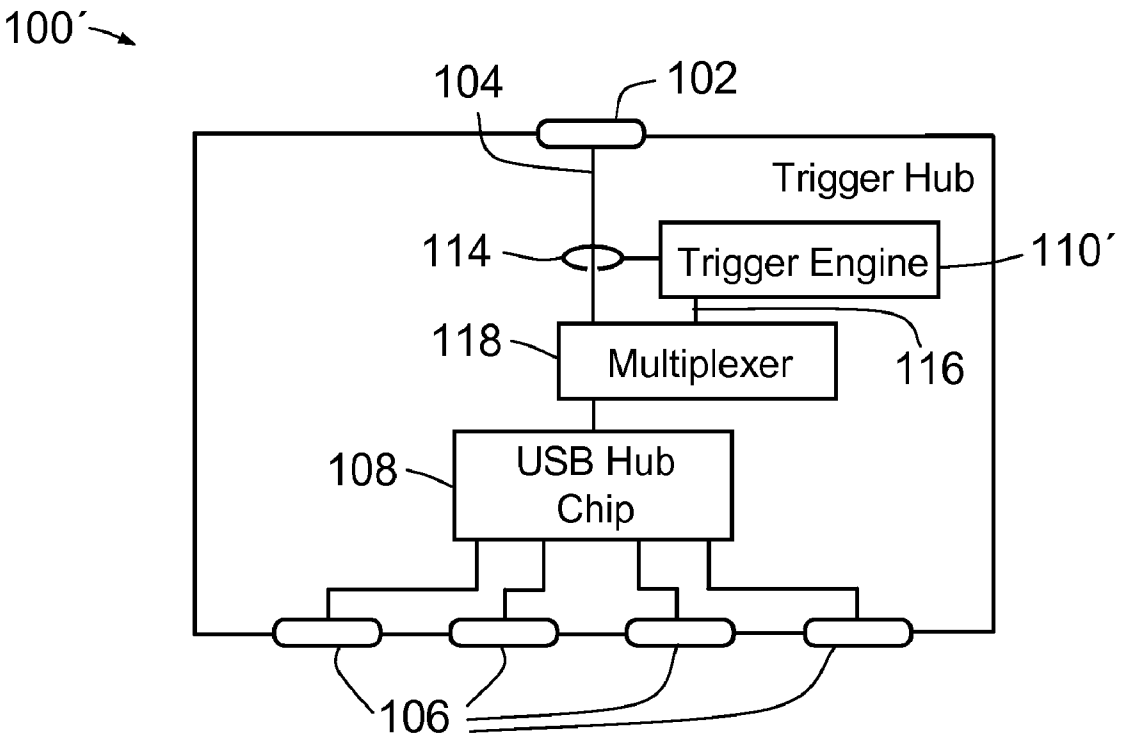
Publication Classification

(51) **Int. Cl.**
G06F 3/00 (2006.01)

(52) **U.S. Cl.** **710/60**

(57) **ABSTRACT**

A system for synchronising the operation of a measurement instrument having a microcontroller, a local oscillator and function circuitry to an external timebase is provided. The system includes a USB Host Controller; an interrupt generator adapted to respond to ITPs by generating respective interrupts and passing the interrupts to the microcontroller; and a timer for measuring an interval between receptions of the ITPs in a time domain of the local oscillator.



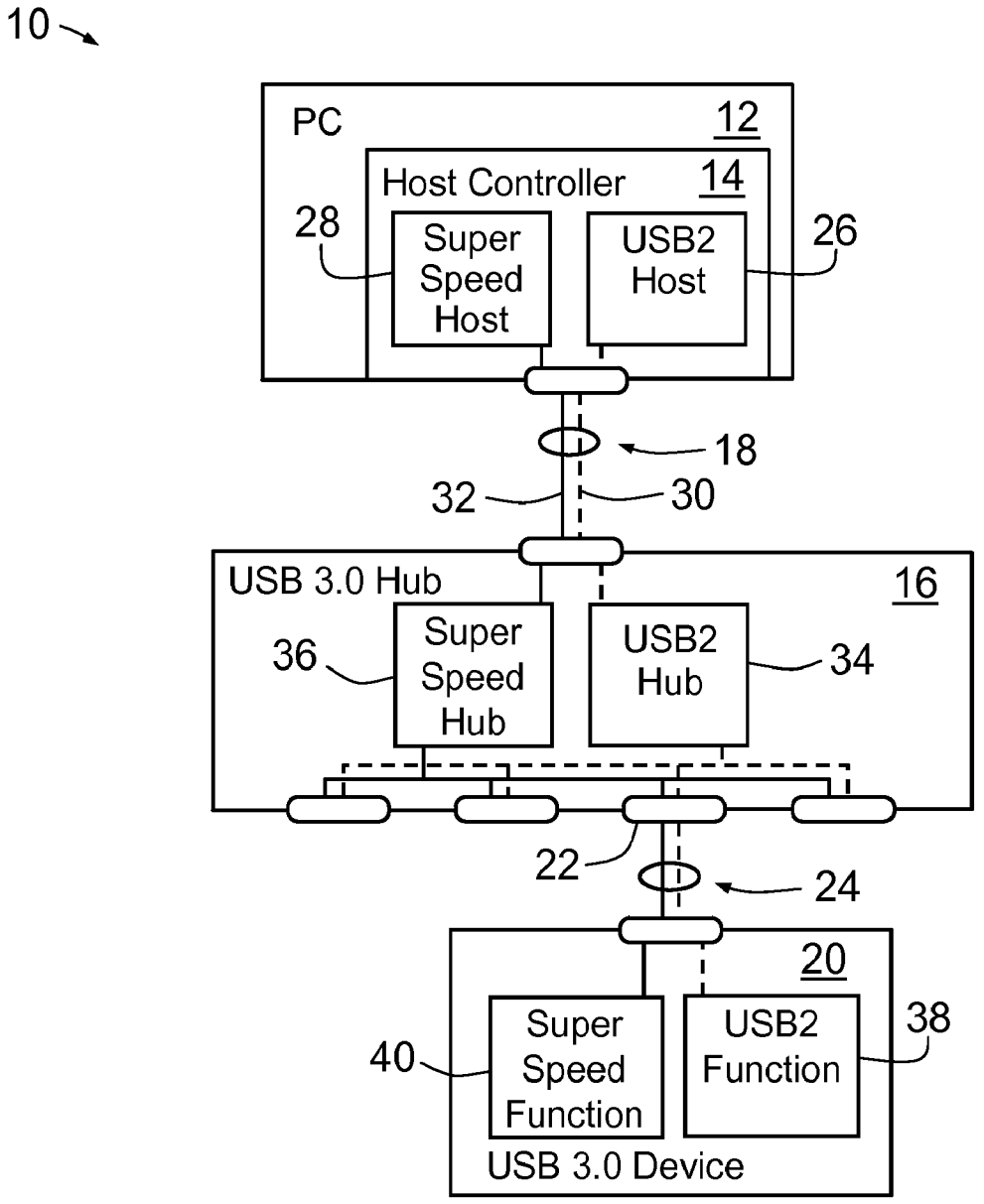


Figure 1
(*background art*)

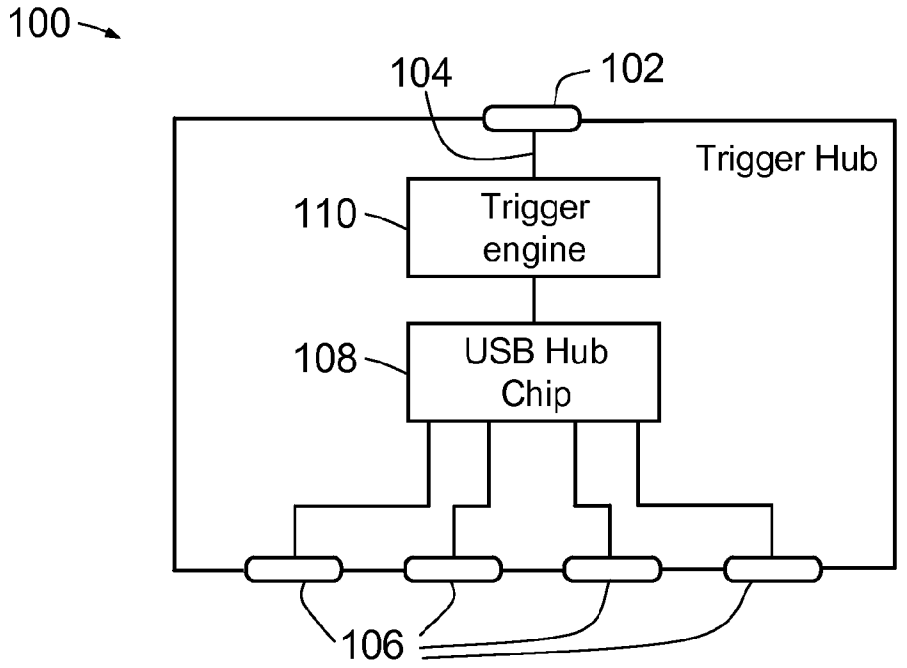


Figure 2A

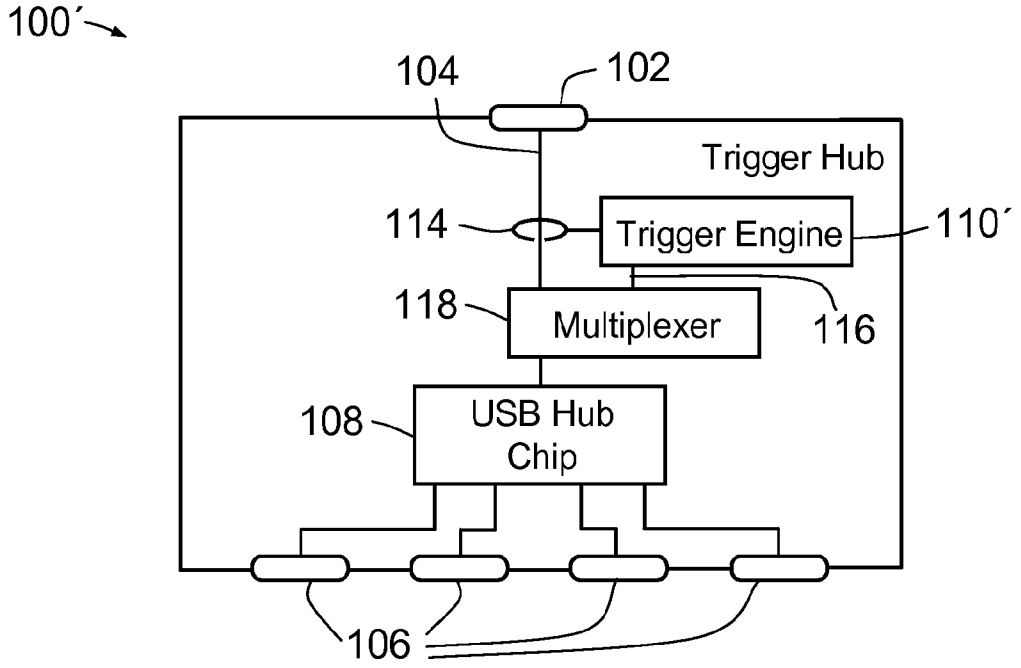


Figure 2B

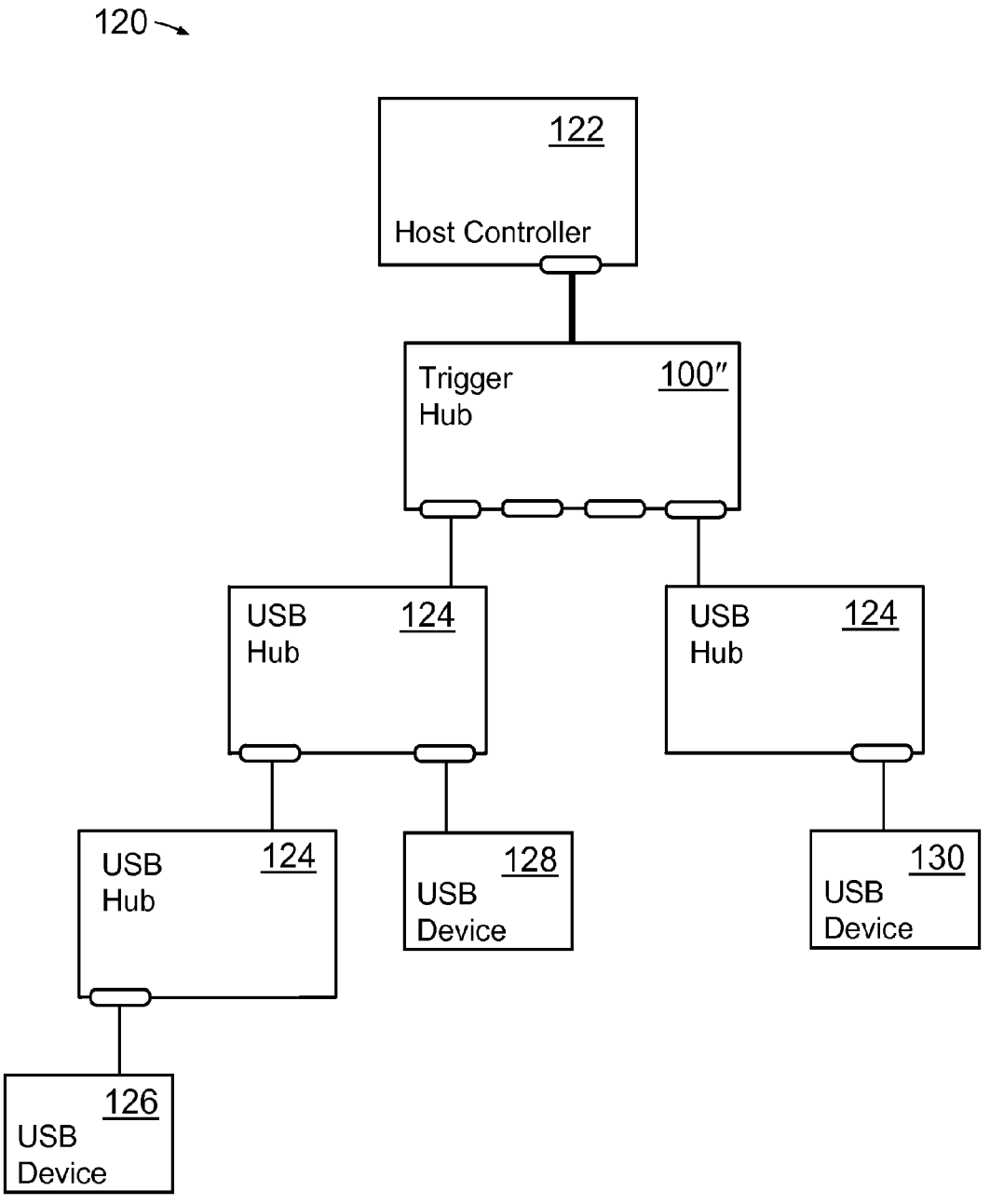

120
122
Host Controller
Trigger Hub
100″
USB Hub
124
USB Hub
124
USB Hub
124
128
USB Device
130
USB Device
126
USB Device

Figure 3

SYNCHRONISATION AND TRIGGER DISTRIBUTION ACROSS INSTRUMENTATION NETWORKS

RELATED APPLICATION

[0001] This application is based on and claims the benefit of the filing date of U.S. application No. 61/179,904 filed 20 May 2009, the content of which as filed is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

[0002] The present invention relates to a method and apparatus for providing a synchronization and timing system, with connectivity based on revision three of the Universal Serial Bus (USB) architecture (or USB 3.0), of particular but by no means exclusive use in providing clocks, data acquisition and automation and control of test and measurement equipment, instrumentation interfaces and process control equipment, synchronized to an essentially arbitrary degree in either a local environment or in a distributed scheme.

BACKGROUND OF THE INVENTION

[0003] The USB specification up to and including revision 2.0 was intended to facilitate the interoperation of devices from different vendors in an open architecture. USB 2.0 data is encoded using differential signalling (viz. in which two wires transfer the information) in the form of the difference between the signal levels of those two wires. The USB 2.0 specification is intended as an enhancement to the PC architecture, spanning portable, desktop and home environments.

[0004] However, USB was user focussed so the USB 2.0 specification lacked a mechanism for synchronising devices to any great precision. Several proposals attempted to address this and other deficiencies. For example, U.S. Pat. No. 6,343,364 (Leydier et al.) discloses an example of frequency locking to USB traffic, which is directed toward a smart card reader. This document teaches a local, free-running clock that is compared to USB SYNC and packet ID streams; its period is updated to match this frequency, resulting in a local clock with a nominal frequency of 1.5 MHz. This provides a degree of synchronization sufficient to read smart card information into a host PC but, as this approach is directed to a smart card reader, inter-device synchronization is not addressed.

[0005] WO 2007/092997 (Foster et al.) discloses a synchronized USB device that allows the generation of accurate clock frequencies on board the USB device regardless of the accuracy of the clock in the Host PC. The USB SOF packet is decoded by the USB device, and treated as a clock carrier signal instead of acting as a clock reference.

[0006] The carrier signal, once decoded from the USB traffic, is combined with a scaling factor to generate synchronization information and hence to synthesize a local clock signal with precise control of the clock frequency. In this way, the frequency of the local clock signal can be more accurate than the somewhat ambiguous frequency of the carrier signal.

[0007] This arrangement is said to be able to produce a local clock signal to arbitrarily high frequencies, such as a clock frequency of tens of megahertz, and thus to ensure that the local clock of each device connected to a given USB is synchronized in frequency. U.S. application Ser. No. 10/620,769 also teaches a method and apparatus to further synchronize multiple local clocks in phase by measurement of signal propagation time from the host to each device and provision of clock phase compensation on each of the USB devices.

[0008] U.S. patent application Ser. No. 12/279,328 (Foster et. al.) teaches synchronisation of the local clocks of a plurality of USB devices to a timebase received from another interface. In one embodiment, a USB device contains a local clock that is synchronised to an externally provided time signature across Ethernet using the IEEE-1588 protocol. In yet another embodiment the USB device's clock is synchronised to a timebase derived from a Global Positioning System (GPS) synchronised clock.

[0009] All of the above systems work within the bounds of conventional USB 2.0 and as such are limited in several areas. USB 2.0 is limited in range by the device response timeout. This is the window of time that the USB Host Controller allocates for receipt of a signal from a given USB device in response to a request from the USB Host Controller. The physical reach of USB 2.0 is therefore approximately 25 m.

[0010] The USB 3.0 specification was released in November 2008 and is also focussed on consumer applications. The USB 3.0 specification makes significant changes to the architecture of USB. In particular, the background art synchronisation schemes discussed above will not work with the new 5 Gb/s protocol (termed 'SuperSpeed USB') because it does away with the broadcast mechanism for SOF packets.

[0011] USB 3.0 defines two parallel and independent USB busses on the same connection cable. Firstly, the USB 2.0 bus remains unchanged (for backward compatibility) and offers Low Speed (1.5 Mb/s), Full Speed (12 Mb/s) and High Speed (480 Mb/s) protocols. The second bus—for 5 Gb/s traffic—provides the SuperSpeed USB. These busses operate independently, except that operation of the busses to a given USB device is mutually exclusive. That is, if a SuperSpeed connection is possible, then the USB 2.0 bus in disconnected to that device.

[0012] The dual-bus architecture of USB 3.0 is depicted schematically at **10** in FIG. **1**. Personal Computer **12**, containing USB Host Controller **14**, is connected to USB 3.0 Hub **16** by first USB 3.0-compliant cable **18**; USB 3.0 device **20** is connected to a downstream port **22** of USB 3.0 Hub **16** by second USB 3.0-compliant cable **24**.

[0013] USB Host Controller **14** contains both a USB 2.0 Host **26** and a SuperSpeed Host **28**. These two hosts **26**, **28** are independent of one another, and each host **26**, **28** is capable of connecting up to 127 devices (including hubs). USB 3.0-compliant cables are compound cables, containing a USB 2.0-compliant cable and a series of shielded conductors capable of transmitting SuperSpeed signals. Hence, USB 3.0-compliant cable **18** comprises USB 2.0-compliant cable **30** and shielded conductors **32**.

[0014] USB 3.0 Hub **16** contains both a USB 2.0 Hub function **34** and a SuperSpeed Hub function **36**, each connected directly to its respective Host **26**, **28** by compound cable **18**. USB 3.0 device **20** contains both a USB 2.0 device function **38** and a SuperSpeed device function **40**, each connected back to its respective hub function **34**, **36** of USB 3.0 Hub **16** by compound cable **24**.

[0015] At enumeration of USB 3.0 device **20**, SuperSpeed Host **28** checks for the presence of a SuperSpeed device function (**40**). If a SuperSpeed device is found, then a connection is established. If a SuperSpeed device is not found (as in the case where only a USB 2.0 device is connected to port **22**), then the USB 2.0 Host **26** checks for the presence of a USB 2.0 device function (**38**) at device **20**. Once the Host

Controller **14** determines which device function is connected, it tells the USB 3.0 Hub **16** to only enable communication for downstream port **22** corresponding to whether the USB 2.0 device function **38** or SuperSpeed device function **40** is attached. This means that only one of the two parallel busses is in operation at any one time to an end device such as USB 3.0 device **20**.

[0016] Furthermore, SuperSpeed USB has a different architecture from that of the USB 2.0 bus. Very high speed communication systems consume large amounts of power owing to high bit rates. A design requirement of SuperSpeed USB was lower power consumption, to extend the battery life of user devices. This has resulted in a change from the previous broadcast design of the USB 2.0: SuperSpeed is not a broadcast bus, but rather directs communication packets to a specific node in the system and shuts down communication on idle links.

[0017] This significantly affects any extension of the synchronisation schemes of, for example, U.S. patent application Ser. No. 12/279,328, whose method and apparatus for synchronising devices is based on a broadcast clock carrier signal that is delivered to each device on the bus, which is unsuitable in SuperSpeed USB.

[0018] A SuperSpeed Hub function acts as a device to the host (or upstream port) and as a host to the device (or downstream port). This means that the SuperSpeed Hub function acts to buffer and schedule transactions on its downstream ports rather than merely acting as a repeater. Similarly, the SuperSpeed Hub function does so with scheduling transmissions on the upstream port. A heavily burdened Hub function can therefore add significant non-deterministic delays in packet transmission through the system. This also precludes the use of USB 2.0 synchronisation schemes such as that of U.S. patent application Ser. No. 12/279,328 from operating on SuperSpeed USB.

[0019] The crude Isochronous synchronisation of USB 2.0 has been significantly improved in the USB 3.0 specification. Opening an Isochronous communication pipe between a Host Controller and a USB device guarantees a fixed bandwidth allocation in each Service Interval for the communication pipe. The Isochronous Protocol of USB 3.0 contains a so-called Isochronous Timestamp Packet (ITP), which is sent at somewhat regular intervals to each Isochronous Endpoint and which contains a timestamp of the beginning of ITP transmission by the USB Host Physical Layer (Phy) in the time domain of the Host Controller. The Isochronous Timestamp Packet is accurate to about 25 ns. SuperSpeed USB shuts down idle links to conserve power, but links must be active in order to receive an Isochronous Timestamp Packet. The Host Controller must therefore guarantee that all links to a device are in full active mode (termed power state U**0**) before transmission of the Isochronous Timestamp Packet.

[0020] Unfortunately the Isochronous Timestamp packet can be delayed in propagation down the USB network. USB 3.0 also does not provide a way of determining the propagation time of packets in SuperSpeed USB and hence no way of accurately knowing the phase relationship between time domains on different USB devices. Phase differences of several hundred nanoseconds are expected to be a best case scenario with SuperSpeed USB making it impractical for instrumentation or other precision timing requirements.

[0021] U.S. Pat. No. 5,566,180 (Eidson et al.) discloses a method of synchronising clocks in which a series of devices on a communication network transmit their local time to each other and network propagation time is determined by the ensemble of messages. Further disclosures by Eidson (U.S. Pat. Nos. 6,278,710, 6,665,316, 6,741,952 and 7,251,199) extend this concept but merely work toward a synchronisation scheme in which a constant stream of synchronising messages are transferred between each of the nodes of a distributed instrument network via Ethernet. This continual messaging consumes bandwidth and limits the accuracy of the possible synchronisation to several hundred nano-seconds in a point-to-point arrangement and substantially lower accuracy (typically micro-seconds) in a conventional switched subnet.

[0022] It should be understood that the terms 'clock signals' and 'synchronisation' in this disclosure are used to refer to clock signals, trigger signals, delay compensation information and propagation time measurement messages. It should also be understood that a 'notion of time' in this disclosure is used to denote an epoch or 'real time' and can also be used to refer to the combination of a clock signal and an associated epoch.

SUMMARY OF THE INVENTION

[0023] It is a general object of the present invention to enable precision synchronisation of a plurality USB devices, up to a predefined maximum, according to the USB3 Specification.

[0024] According to a first broad aspect, the present invention provides a system for synchronising the operation of a measurement instrument having a microcontroller, a local oscillator and function circuitry to an external timebase, the system comprising:

[0025] a USB Host Controller to which the local oscillator is attachable;

[0026] an interrupt generator adapted to respond to receipt of one or more Isochronous Timestamp Packets by generating respective interrupts and passing the interrupts to the microcontroller; and

[0027] a timer or counter adapted to measure an interval between receptions of the Isochronous Timestamp Packets in a time domain of the local oscillator (from which, if desired, information can be derived about a frequency of the local oscillator with respect to the timestamped Isochronous Timestamp Packets);

[0028] wherein the system is adapted to control the instrument to open one or more Isochronous communication pipes or endpoints to the USB Host Controller, to ensure that the instrument is in link state U**0** in preparation for receiving a plurality of Isochronous Timestamp Packets (ITPs) and to control the USB Host Controller to send a plurality of the Isochronous Timestamp Packets to the Isochronous communication pipes or endpoints, the microcontroller is configured to respond to each of the plurality of interrupts (such as with an interrupt service routine provided therein) by generating an output signal adapted to be used as a synchronization reference signal, and the instrument is configured to create a mapping between respective local times of a plurality of operations or events of the function circuitry of the instrument and a time domain of the USB Host Controller.

[0029] In one embodiment, the system comprises the instrument. In a particular embodiment, the instrument is a SuperSpeed USB device.

[0030] Thus, the operation of, for example, a SuperSpeed USB device can be synchronised, as each of the operations (or events) can be mapped back to the time domain of the USB Host Controller.

[0031] The microcontroller may be further configured to read respective timestamps from each of a selected pair of the Isochronous Timestamp Packets, and calculate a period between the selected pair of the Isochronous Timestamp Packets from their respective timestamps (and, further, may optionally form a comparison between the interval between selected successive Isochronous Timestamp Packets as determined by the timer or counter and the calculated period.

[0032] Thus, knowledge of the time domain of the USB Host Controller may be improved

[0033] In one embodiment, the system is further adapted system to omit from the comparison any of the Isochronous Timestamp Packets that contains a Packet_Delayed flag.

[0034] In one embodiment, the microcontroller is further configured to make a plurality of such readings, and to generate a time series of determinations of the local clock period and timestamps.

[0035] The microcontroller or USB Host Controller may be adapted to statistically analyze the time series and to improve synchronisation accuracy therefrom.

[0036] Also according to this aspect there is provided a method of synchronising the operation of a measurement instrument

to an external timebase, the method comprising:

the instrument, having a microcontroller and a local oscillator attached to a USB Host Controller, opening one or more Isochronous communication pipes or endpoints to the USB Host Controller;

ensuring that the instrument is in link state U**0** in preparation for receiving a plurality of Isochronous Timestamp Packets (ITP);

the USB Host Controller sending a plurality of Isochronous Timestamp Packets to the Isochronous communication pipes or endpoints;

[0037] the instrument receiving the plurality of Isochronous Timestamp

[0038] Packets and in response generating a respective plurality of interrupts and passing the interrupts to the microcontroller;

[0039] measuring a time interval between receptions of the Isochronous Timestamp Packets in a time domain of the local oscillator (and optionally deriving information about the frequency of the local oscillator with respect to the timestamped Isochronous Timestamp Packet therefrom); and

[0040] the instrument creating a mapping between respective local times of a plurality of operations or events of function circuitry of the instrument and a time domain of the USB Host Controller.

[0041] In one embodiment, the instrument is a SuperSpeed USB device.

[0042] The method may comprise the microcontroller responding to each of the interrupts (such as with an interrupt service routine provided therein) by generating an output signal adapted to be used as a synchronization reference signal for a plurality or all of the Isochronous Timestamp Packets, and deriving information about the frequency of the local oscillator with respect to the timestamped Isochronous Timestamp Packet therefrom with a first counter/timer measuring the period of interval between reception of successive the output signals.

[0043] In one embodiment, the method includes mapping each of the operations or events back to the time domain of the USB Host Controller.

[0044] In one embodiment, the instrument includes instrumentation and measurement or data acquisition circuitry.

[0045] In one embodiment, the method further comprises:

[0046] reading respective timestamps from each of the plurality of Isochronous Timestamp Packets; and

[0047] calculating a period between a selected successive pair of the Isochronous Timestamp Packets from their respective timestamps.

[0048] The method may include forming a comparison between the interval between the selected successive Isochronous Timestamp Packets as determined by the timer or counter and the calculated period.

[0049] The method may include omitting any of the Isochronous Timestamp Packets that contain a Packet_Delayed flag from the comparison.

[0050] The method may comprise making a plurality of the readings and generating a time series of determinations of the local clock period and timestamps.

[0051] The method may further comprise statistically analyzing the time series and improving synchronisation accuracy therefrom.

[0052] In a particular embodiment, the interrupts result in interrupt service routines generated in response to the detection of an Isochronous Timestamp Packet by the SuperSpeed USB device.

[0053] In a particular embodiment, the interrupts are hardware interrupts generated in response to the detection of an Isochronous Timestamp Packet by the SuperSpeed USB device.

[0054] In one embodiment, the method further comprises:

[0055] measuring signal propagation time from a selected point in a USB network containing the SuperSpeed USB device to the SuperSpeed USB device; and

[0056] adjusting the phase of the local oscillator;

[0057] whereby the local oscillator is synchronised to a desired phase.

[0058] In an embodiment, the timer/counter uses only those of the Isochronous Timestamp Packets whose transmission was not delayed in a USB network. Furthermore statistical means may be used to improve the accuracy of measurements and improve synchronisation. This may be by averaging the measurements of the interval between reception of successive the output signals, or by performing a statistical analysis on the set of such measurements of the time intervals.

[0059] In a particular embodiment, the method includes measuring the interval between receptions of the synchronisation reference signals with a counter/timer function clocked from the local oscillator.

[0060] In an embodiment, the time interval used to measure the frequency of the local oscillator is an interval between receptions of successive Isochronous Timestamp Packets. The accuracy of measurement of the local oscillator frequency may be increased by measurement over multiple successive intervals and using statistical analysis.

[0061] In one embodiment, the microcontroller has timer/counter functionality. It will be readily understood by those skilled in the art that such counter/timer functionality can be deployed in logic devices external to the microcontroller, for

example Field Programmable Gate Arrays (FPGA) or Complex Programmable Logic Devices (CPLD) or a dedicated counter/timer circuit.

[0062] In one embodiment, the microcontroller has interrupt service routine capability whereby the interrupt service routines can be called or triggered in response to detection of the Isochronous Timestamp Packets.

[0063] The interrupts may be software interrupts, but in one embodiment, the interrupts are hardware interrupts wherein there is minimal latency in generating the required output from an Interrupt Service Routine (ISR).

[0064] In one embodiment, the local oscillator is a free-running local oscillator, but it will be understood by those skilled in the art that the local oscillator may assume alternative forms and comprise, for example, a Voltage Controlled Crystal Oscillator (VCXO), a Temperature Compensated Crystal Oscillator (TCXO), an Oven Controlled Crystal Oscillators (OCXO) or a multi-tap clock.

[0065] It will also be understood by those skilled in the art that the counter/timer circuitry may not be clocked directly from the local oscillator but instead via a clock source divided or multiplied in frequency from the local oscillator.

[0066] The method of this aspect may be used to synchronise a plurality of USB devices attached to a common USB host controller, whereby the operations or functions of the USB devices are synchronised to an arbitrarily precise degree. Furthermore the method may include determining the propagation time of the timestamped Isochronous Timestamp Packet from the USB Host Controller to the USB device according to any of such method of the present invention described herein, depending on the particular synchronisation channel employed in the respective embodiment.

[0067] It will be understood by those skilled in the art that measuring the period between receptions of successive carrier signals with respect to the free-running local oscillator is equivalent to knowing the time of the plurality of receptions of Isochronous Timestamp Packets in the time domain of the free-running oscillator. It will also be understood that such a relative notion of time can be referenced to an absolute notion of time.

[0068] In one embodiment a SuperSpeed synchronisation channel is provided, adapted to determine the respective period of time between reception of successive Isochronous Timestamp Packets. However, this channel may be replaced with a non-SuperSpeed synchronisation channel. The non-SuperSpeed synchronisation channel may be adapted to any one of the methods of synchronisation as disclosed in this invention. The non-SuperSpeed synchronisation channel may include a periodic timing signal adapted for use as the reference timing signal instead of the Isochronous Timestamp Packet in the SuperSpeed synchronisation channel.

[0069] Furthermore, it is often desirable to know the absolute phase relationship between the plurality of timebases, but one device may be close to the Host Controller and another may be a long distance away. Since each of a plurality of instruments (such as SuperSpeed USB devices) synchronised according to this method would be synchronised to the reception of the plurality of Isochronous Timestamp Packets, absolute phase information is not automatically available.

[0070] Therefore if the instrument (such as a SuperSpeed USB device) is one of a plurality of like devices, the method may also include an apparatus for determining the relative propagation time of the multicast Isochronous Timestamp Packets from the USB Host Controller to each of the devices, such as by any of the methods for determining propagation time of the present invention described herein. Furthermore each of the devices may contain a mechanism adapted to adjust the phase of their respective local notions of time, resulting in a synchronised USB with known absolute phase relationships between devices.

[0071] It will be understood by those skilled in the art that such a measurement synchronisation method as described for a USB instrument is also applicable to any measurement instrument that receives a plurality of timestamped signals with which to determine the rate of its local clock and hence the timebase of the local measurement system, including to permit synchronisation of a plurality of disparate networks (for example but not limited to PXI, PXI-express, IEEE-1588 ethernet, PCI, VXI and USB).

[0072] According to this aspect, there is also provided an apparatus for synchronising the operation of a measurement instrument having a local oscillator and function circuitry to an external timebase, the apparatus comprising:

[0073] circuitry for decoding a plurality of time-stamped packets from the communication bus;

[0074] circuitry for measuring a time interval between receptions of the time-stamped packets in a time domain of the local oscillator; and

[0075] a computing mechanism configured to create a mapping between respective local times of a plurality of operations or events of the function circuitry and a time domain of a communication bus to which the instrument is attached.

[0076] The apparatus may comprise the instrument.

[0077] In one embodiment, the measurement instrument is a SuperSpeed USB device and the communication bus is a SuperSpeed USB.

[0078] In one embodiment, the instrument is an Ethernet-based device.

[0079] In an embodiment, the local oscillator is a free-running oscillator.

[0080] In a particular embodiment, the local oscillator is a voltage controller oscillator (VCXO), a temperature compensated crystal oscillator (TCXO), an oven controlled crystal oscillator (OCXO) or some other clock circuitry.

[0081] In a second broad aspect, the invention provides a method of synchronising the operation of a plurality of instruments each having a microcontroller and a local oscillator attached to a common USB Host Controller, the method comprising:

[0082] for each respective instrument:

[0083] (a) the instrument opening one or more Isochronous communication pipes or endpoints to the USB Host Controller;

[0084] (b) ensuring that the instrument is in link state U**0** in preparation for receiving a plurality of Isochronous Timestamp Packets (ITP);

[0085] (c) the USB Host Controller sending a plurality of Isochronous Timestamp Packets to the Isochronous communication pipes or endpoints;

[0086] (d) the instrument receiving the plurality of Isochronous Timestamp Packets and in response generating a respective plurality of interrupts and passing the interrupts to the microcontroller;

[0087] (e) a timer measuring a time interval between receptions of the Isochronous Timestamp Packets in a time domain of the local oscillator; and

[0088] (f) the instrument creating a mapping between respective local times of a plurality of operations or events of the function circuitry of the instrument and a time domain of the USB Host Controller.

[0089] The instruments may be SuperSpeed USB devices.

[0090] In a third broad aspect, the invention provides a system for reducing triggering latency and acquiring data across a plurality of different instrumentation interfaces, comprising:

[0091] a USB host controller; and

[0092] a plurality of USB devices in data communication with the USB host controller, each of the USB devices having a synchronous clock, a synchronized real time clock register and a memory;

[0093] wherein the USB devices are controllable to synchronously commence acquiring data, to store to their respective memory the data once acquired and to store to their respective memory time stamp information indicative of the time of acquisition of at least some of the acquired data; a first of the USB devices is configured to respond to a data collection command to collect data by sending to the USB host controller a first message that includes data indicative of a time of receipt of the data collection command, the USB host controller is configured to respond to the first message by sending the other USB devices a second message including the data indicative of the time of receipt by the first USB device of the data collection command, and the other USB devices are configured to respond to the second message by reading their respective memories and sending acquired data stored therein to the USB host controller commencing from a location in each respective memory corresponding to the time of receipt or a next available location.

[0094] In a fourth broad aspect, the invention provides a USB system, comprising:

[0095] a plurality of USB networks, each comprising a USB host controller and a plurality of synchronised USB devices, the plurality of USB networks being synchronised such that the plurality of USB devices across the plurality of USB networks are mutually synchronised;

[0096] wherein the USB devices are configured (i) to execute a plurality of instructions upon receipt of an external trigger signal, the instructions comprising that the respective USB device make a measurement of one or more parameters of the respective USB device, (ii) to time stamp the respective measurement, and (iii) record the measurement in a respective local memory; and

[0097] wherein a first of the USB devices is configured to notify its respective USB host controller upon receipt of the external trigger signal, the respective USB host controller is configured to respond to being notified of the receipt by notifying each of the other USB devices connected to the respective USB host controller and the one or more other of the USB host controllers of the receipt of the external trigger signal and its associated timestamp by the first USB device, the other of the USB host controllers are configured to notify their respective USB devices of the occurrence of the external trigger signal and its associated timestamp, the plurality of USB devices on the other of the USB networks are configured to transmit a content of their respective local memories to their respective USB host controller in response to receiving the notification of the receipt, and the plurality of USB devices on the other of the USB networks are configured to execute their respective plurality of instructions in response to receiving notification of the receipt.

[0098] In one embodiment, the USB devices are configured to continue executing their prescribed instructions while transmitting the contents of their respective local memories to their respective Host Controllers, to continue to buffer data to their respective local memories and to begin steaming data directly to their respective Host Controllers once the contents of their respective memories have been completely transferred to their respective Host Controllers.

[0099] In a fifth broad aspect, the invention provides a trigger engine or device, comprising:

[0100] a local clock synchronisable to the respective local clocks of a plurality of USB devices attached to a common USB host controller;

[0101] wherein the trigger device is configured to time stamp an external trigger signal upon receipt thereof, and to notify the USB host controller of the receipt and of the associated time stamp.

[0102] In one embodiment, the trigger device is configured to execute a set of instructions and buffer time stamped results thereof in a local memory, and to transmit the buffered and time stamped results to the USB host controller after receipt of the external trigger signal.

[0103] The device may be adapted to continue executing the set of instructions while transferring the results from the memory to the USB host controller, including buffering data to the memory, and to begin transferring the data to the USB host controller after the results have been transferred to from the memory to the USB host controller.

[0104] According to this aspect, the invention also provides a USB device, comprising:

[0105] a local clock; and

[0106] a local memory;

[0107] wherein the local clock is synchronisable to respective local clocks of one or more other USB devices, and the USB device is configured to receive a set of instructions from a USB host controller, to initiate the instructions and buffer results in the local memory, and to respond to receiving notification of the time that an external trigger signal was received by a trigger device by transmitting the buffered results to the USB host controller.

[0108] In one embodiment, the USB device is adapted to continue executing the instructions while transmitting the results to the USB host controller, and to buffer data to the memory and to initiate transferring the data to the USB host controller after the results have been transferred to from the memory to the USB host controller.

[0109] In a sixth broad aspect, the invention provides a method of reducing the triggering latency of a plurality of devices or instruments attached to a plurality of instrumentation networks, the networks being in data communication with each other and connected by one or more synchronisation channels, the method comprising:

[0110] synchronising the instrumentation networks via the synchronisation channels;

[0111] synchronising the devices or instruments to their respective instrumentation networks;

[0112] configuring the devices or instruments so as to respond to a trigger notification from one or more of the

other trigger devices or instruments by executing a plurality of instructions, the instructions being adapted to instruct the devices or instruments to one or more measurements of one or more parameters of the respective devices or instruments;

[0113] arming the devices or instruments such that the devices or instruments commence performing their prescribed functions synchronously, receive and store to their respective memory data acquired as a result of performing the prescribed functions and store to their respective memory time stamp information indicative of the time of acquisition of the acquired data;

[0114] one of the trigger devices or instruments responding to receipt of a trigger event by transmitting or initiating transmission of a trigger notification to the other devices or instruments across the plurality of different instrumentation networks;

[0115] the devices or instruments on the instrumentation networks transmitting the content of their respective local memories to their respective network communication controllers in response to receiving the trigger notification; and

[0116] the devices or instruments executing their respective plurality of instructions in response to receiving the trigger notification.

[0117] This aspect also provides an apparatus for transferring trigger signals from a first USB device to a plurality of second USB Devices attached to a USB network with minimal latency, the apparatus comprising:

[0118] USB hub circuitry adapted to communicate upstream toward a Host Controller and to communicate downstream toward a plurality of USB devices; and

[0119] trigger engine circuitry adapted to observe communication traffic upstream of the USB hub circuitry and to transmit communication packets downstream to an upstream port of the USB hub circuitry;

[0120] wherein the trigger engine is configured to decode upstream directed communication from a USB network connected downstream of the apparatus, to search for specific trigger request signals transmitted upstream from the USB network, and to transmit trigger command packets downstream toward the USB hub circuitry upon reception of the trigger request signals.

[0121] In one embodiment, the apparatus is configured so that communication from upstream of the apparatus to the USB Hub circuitry passes through the trigger engine.

[0122] The apparatus may further comprise multiplexer circuitry adapted to selectively route either an upstream signal path or output of the trigger engine to the upstream port of the USB hub circuitry.

[0123] According to this aspect there is still further provided an apparatus for transferring trigger signals from a first USB device to a plurality of second USB Devices attached to a USB network with minimal latency, the apparatus comprising:

[0124] circuitry adapted for communicating upstream toward a USB Host Controller and communicating downstream toward a plurality of USB devices;

[0125] trigger engine circuitry adapted to observe all communication traffic through the apparatus; and

[0126] circuitry adapted to transmit communication packets downstream;

[0127] wherein the apparatus is adapted to decode upstream directed communication from the plurality of USB devices, to search for specific trigger request signals transmitted upstream from the USB network, and to transmit trigger command packets downstream toward the plurality of USB devices upon reception of the trigger request signals.

[0128] In one embodiment, the apparatus further comprises multiplexer circuitry adapted to selectively disable an upstream signal connection.

[0129] According to this aspect there is still further provided a method of transferring trigger signals from a first USB device to one or more second USB devices attached to a USB network with minimal latency, the method comprising:

[0130] arming the first USB device to receive a Trigger Condition;

[0131] the first USB device detecting the Trigger Condition;

[0132] the first USB Device responding by transmitting a Trigger Request message upstream to a USB Host Controller of the USB network;

[0133] detecting passage of the Trigger Request with a Trigger Engine located in the USB network; and

[0134] the Trigger Engine responding to detection of the Trigger Request by transmitting a Trigger Command message downstream to the second USB devices.

[0135] In one embodiment, the method further comprises:

[0136] opening a communication pipe between a USB Host Controller and the plurality of USB devices; and

[0137] inserting the Trigger Command into a data packet passing downstream in the communication pipe.

[0138] In one embodiment, the method further comprises:

[0139] selectively disconnecting the plurality of USB devices upstream of the Trigger Engine; and

[0140] transmitting the Trigger Command downstream of the Trigger Engine.

[0141] The Trigger Condition may be reception of a signal from an external source.

[0142] The USB devices may be non-SuperSpeed Devices. The Trigger Engine may broadcast the Trigger Command to all attached downstream USB devices.

[0143] The USB devices may be SuperSpeed USB Devices. The Trigger Engine may selectively transmit the Trigger Command to the second USB devices, each of the second USB devices being addressed using a USB Routing String.

[0144] In a certain embodiment, the method includes detecting passage of the Trigger Request passing through the Trigger Engine.

[0145] According to this aspect there is also provided a method of triggering a plurality of first USB devices to perform their respective predefined operations with a trigger signal applied at least one second USB device with minimal latency, the first and second USB devices attached to a common USB network, the method comprising:

[0146] arming the second USB device to receive a Trigger Condition;

[0147] the second USB Device detecting the Trigger Condition;

[0148] the second USB Device responding by transmitting a Trigger Request message upstream to a USB Host Controller of the USB network;

[0149] detecting passage of the Trigger Request with a Trigger Engine; and

[0150] the Trigger Engine responding to detecting the passage of the Trigger Request by transmitting a Trigger Command message downstream to the plurality of first USB devices.

[0151] Thus, the plurality of second USB devices execute their respective predefined operations with minimal trigger propagation latency.

[0152] In one embodiment, the method further comprises:

[0153] opening a communication pipe between a USB Host Controller and the plurality of USB devices; and

[0154] inserting the Trigger Command into a data packet passing downstream in the communication pipe.

[0155] In an embodiment, the method further comprises:

[0156] selectively disconnecting the plurality of USB devices upstream of the Trigger Engine; and

[0157] transmitting the Trigger Command downstream of the Trigger Engine.

[0158] The Trigger Condition may be reception of a signal from an external source.

[0159] The USB devices may be non-SuperSpeed Devices. The Trigger Engine may broadcast the Trigger Command to all attached downstream USB devices.

[0160] The USB devices may be SuperSpeed USB Devices. TheTrigger Engine may selectively transmit the Trigger Command to the second USB devices, the second USB devices being addressed using a USB Routing String.

[0161] In one embodiment, the method includes detecting passage of the Trigger Request passing through the Trigger Engine.

[0162] In a seventh broad aspect, the invention provides a method of triggering a plurality of USB devices to perform their respective predefined operations upon reception of a plurality of trigger signals, the USB devices attached to a common USB network, comprising:

[0163] arming the USB devices to receive respective trigger conditions;

[0164] configuring a trigger engine attached to the USB network with a specific combinatorial trigger state, the combinatorial trigger state being a combination of trigger states of the USB devices,

[0165] the USB devices detecting the respective trigger conditions;

[0166] the USB devices transmitting respective trigger request messages upstream to a USB Host Controller of the USB network, the trigger request messages being indicative of respective trigger states of the USB devices;

[0167] detecting passage of the trigger requests with the trigger engine;

[0168] the trigger engine maintaining a record of a trigger status of each of the USB devices;

[0169] the trigger engine comparing the record of trigger status with the combinatorial trigger state;

[0170] the trigger engine transmitting a trigger command message downstream to the USB devices upon the trigger status matching the combinatorial trigger state.

[0171] Thus, the plurality of USB devices execute their respective predefined operations with minimal trigger propagation latency.

[0172] In one embodiment, the combinatorial trigger state includes time dependence, such that temporal relationships between transitions of the respective plurality of trigger states is important.

[0173] The method may further comprise:

[0174] opening a communication pipe between a USB Host Controller and the plurality of USB devices; and

[0175] inserting the trigger command into a data packet passing downstream in the communication pipe.

[0176] In one embodiment, the method further comprises:

[0177] selectively disconnecting the USB devices upstream of the trigger engine; and

[0178] transmitting the trigger command downstream of the trigger engine.

[0179] The Trigger Condition may be reception of a signal from an external source.

[0180] The USB devices are non-SuperSpeed Devices. The trigger engine may broadcasts the trigger command to the USB devices.

[0181] The USB devices may be SuperSpeed USB Devices. The trigger engine may selectively transmit the trigger command to the USB devices, each of the USB devices being addressed using a USB Routing String.

[0182] In one embodiment, the method includes detecting passage of the trigger request passing through the trigger engine.

[0183] According to this aspect, there is also provided an apparatus for triggering a plurality of USB devices attached to a common USB network to perform respective predefined operations upon reception of respective trigger signals, the apparatus comprising:

[0184] circuitry adapted for communicating upstream toward a USB Host Controller and communicating downstream toward the USB devices;

[0185] trigger engine circuitry adapted to observe communication traffic through the apparatus;

[0186] circuitry adapted to compare a plurality of trigger request signals with a predefined Combinatorial Trigger State; and

[0187] circuitry adapted to transmit communication packets downstream;

[0188] wherein the apparatus is adapted to decode upstream directed communication from the USB devices, to search for specific trigger request signals transmitted upstream from the USB network, and to transmit trigger command packets downstream toward the USB devices upon reception of the plurality of trigger request signals in correct combinatorial relationship.

[0189] The combinatorial trigger state may includes time dependence.

[0190] The apparatus may be further adapted to:

[0191] open a communication pipe between a USB Host Controller and the USB devices; and

[0192] insert the trigger command into a data packet passing downstream in the communication pipe.

[0193] The apparatus may be further adapted to:

[0194] selective disconnecting the USB devices upstream of a trigger device; and

[0195] transmit the trigger command downstream of the trigger engine.

[0196] The Trigger Condition may be reception of a signal from an external source.

[0197] The USB devices are non-SuperSpeed Devices. The trigger command may be broadcast by a trigger engine to the USB devices.

[0198] The USB devices may be SuperSpeed USB Devices. The trigger command may be selectively transmit-

ted by a trigger engine to the USB devices, each of the USB devices being addressed using a USB Routing String.

[0199] The apparatus may be configured to detect passage of the trigger request passing through the trigger engine.

[0200] It should be noted that all the various features of each of the above aspects of the invention can be combined as suitable and desired.

[0201] Furthermore, it should be noted that the invention also provides apparatuses and systems arranged to perform each of the methods of the invention described above.

[0202] In addition, apparatuses according to the invention can be embodied in various ways. For example, such devices could be constructed in the form of multiple components on a printed circuit or printed wiring board, on a ceramic substrate or at the semiconductor level, that is, as a single silicon (or other semiconductor material) chip.

BRIEF DESCRIPTION OF THE DRAWINGS

[0203] In order that the present invention may be more clearly ascertained, embodiments will now be described, by way of example, with reference to the accompanying drawing, in which:

[0204] FIG. **1** is a schematic diagram of the dual-bus architecture of USB3 according to the background art;

[0205] FIG. **2**A is a schematic representation of an apparatus for reducing the triggering latency of USB devices according to an embodiment of the present invention;

[0206] FIG. **2**B is a schematic representation of an apparatus for reducing the triggering latency of USB devices according to another embodiment of the present invention; and

[0207] FIG. **3** is a schematic representation of a USB network according to another embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0208] FIG. **2**A is a schematic representation of an apparatus in the form of a Trigger Hub **100** according to an embodiment of the present invention, for reducing the triggering latency of USB devices. Trigger Hub **100** has an upstream port **102**, a plurality of downstream ports **106**, a USB Hub chip **108** and a Trigger Controller or Engine **110** (connected to upstream port **102** by upstream line **104**).

[0209] Trigger Engine **110** is thus located between upstream port **102** and USB Hub chip **108** such that all upstream communication passes through it. In this way Trigger Engine **110** is able to observe all communication traffic.

[0210] However, Trigger Engine **110** need not be located 'inline' between upstream port **102** and USB Hub chip **108**: according to a variant of the embodiment of FIG. **2**A, Trigger Engine **110** may be adapted to observe traffic without being inline. FIG. **2**B is a schematic representation of a Trigger Hub **100'** according to this variant in which, as compared with Trigger Hub **100** of FIG. **2**A, like reference numerals have been used to identify like features. Trigger Hub **100'** thus includes an upstream port **102**, a plurality of downstream ports **106** and a USB Hub chip **108**, situated as in Trigger Hub **100** of FIG. **2**A. Trigger Engine **110'** of Trigger Hub **100'**, however, is arranged merely to observe traffic at monitoring point **114** on upstream line **104**, and has a separate downstream connection **116** to USB Hub chip **108**. Trigger Hub **100'** optionally includes a multiplexer **118** to selectively connect either upstream line **104** or Trigger Engine **110'** to USB Hub chip **108**.

[0211] FIG. **3** is a schematic representation of a USB network **120** according to another embodiment of the present invention. USB network **120**, though in some respects a typical and conventional USB network, includes a Trigger Hub **100"** (comprising either Trigger Hub **100** of FIG. **2**A or Trigger Hub **100'** of FIG. **2**B) to reduce the latency in triggering signals passing from one USB device to another. USB network **120** also contains a USB Host Controller **122**, a plurality of USB Hubs **124** and a plurality of USB Devices **126**, **128**, **130**.

[0212] In one scenario illustrating the operation of USB network **120** (and hence of Trigger Hub **100** or Trigger Hub **100'**), USB device **126** wishes to send a low latency trigger to USB devices **128** and **130** (so is henceforth referred to as the 'trigger device' **126'**). When arming USB devices **128** and **130** to receiver a trigger signal, Host Controller **122** opens a communication pipe downstream to each of devices **128** and **130**. This will be referred to as the 'Trigger Pipe' and may be an isochronous pipe in which a fixed bandwidth is made available during each USB frame or it may be another form of data pipe.

[0213] Trigger device **126** is armed and waiting to receive a trigger (typically from an external device). Once the trigger has been received, trigger device responds by sending a message (or trigger request) is transmitted upstream, passing through Trigger Hub **100"** on its way to Host Controller **122**. Trigger Engine **110** decodes the upstream message and observes that trigger device **126** has issued a trigger request.

[0214] Trigger Engine **110** then broadcasts a message, in the form of a Trigger Command, on all downstream ports **106** of Trigger Hub **100"**. In the case of non-SuperSpeed USB, the message passes down the entire network and is received by all attached devices. In the case of SuperSpeed USB, Trigger Engine **110** would need to transmit each message individually to a specific network address, defined by the USB Routing String.

[0215] Each armed USB device **128**, **130** receives the downstream trigger command message and executes its own predefined commands. This approach obviates the need for triggers to pass through the operating system, the slowest part of the chain.

[0216] Trigger Hub **100"** does not have the right to instigate packet transfers according to the USB Specification' only Host Controller **122** has that right. This is why the Trigger Pipe was opened by Host Controller **122** at the beginning of the trigger arming process. In the case of an Isochronous pipe, a small bandwidth channel is opened between Trigger Host **100"** and each attached USB device **126**, **128**, **130**. In the case of non-SuperSpeed devices, only one pipe is required since all packets are broadcast downstream. Once Trigger Hub **100"** receives the trigger command from trigger device **126**, Trigger Hub **100"** waits for the next scheduled downstream transmission time in the Trigger Pipe and inserts a trigger command into the downstream packet. This Trigger Command is then received by USB devices **128** and **130**, which respond by executing their commands.

[0217] In another embodiment, Trigger Engine **110** injects phantom packets downstream. Trigger Engine **110** either waits until it comes across a gap in the downstream packet that it can use to inject the phantom packet, or momentarily disconnects reception of signals from upstream port **102**, allowing Trigger Engine **110** to transmit the downstream Trigger Command.

[0218] Furthermore, these techniques are applicable when combinatorial trigger logic must be applied to a plurality of Trigger Requests. If one considers the case where both USB Devices **126** and **128** are designated Trigger Devices, USB Device **130** may be triggered to perform a function when both USB device **126** and USB device **128** meet certain parameters.

[0219] According to this embodiment, therefore, Trigger Engine **110** contains combinatorial logic. Trigger Engine **110** is programmed by Host Controller **122** with the required combination of triggers. Once Trigger Hub **100"** receives valid trigger requests from both USB devices **126** and **128** it issues a Trigger Command to USB Device **130**.

[0220] Furthermore, this combinatorial logic may include time windowing or other advanced multi-device triggering constraints: essentially any level of complexity of combinatorial logic may be accommodated according to this embodiment. Once all combinations of trigger requests have been received and are valid, a trigger command may be issued by Trigger Engine **110**.

[0221] Thus, these embodiments address the limitation imposed on the speed at which triggers can be transferred from one USB device to the next arising from the Host-centric nature of conventional USB architecture, whereby all communication is initiated by a Host with no direct inter-device communication (a limitation exacerbated by the fact that communication must pass through the operating system, which can have the effect of delaying trigger propagation by milliseconds, or even seconds in cases with a heavily loaded processor).

[0222] Modifications within the scope of the invention may be readily effected by those skilled in the art. It is to be understood, therefore, that this invention is not limited to the particular embodiments described by way of example hereinabove and that combinations of the various embodiments described herein are readily apparent to those skilled in the art.

[0223] In the preceding description of the invention and in the claims that follow, except where the context requires otherwise owing to express language or necessary implication, the expression "Host Controller" embraces all forms of USB Host Controller, including standard USB Host controllers, USB-on-the-go Host Controllers and wireless USB Host Controllers.

[0224] In the preceding description of the invention and in the claims that follow, except where the context requires otherwise owing to express language or necessary implication, the word "comprise" or variations such as "comprises" or "comprising" is used in an inclusive sense, that is, to specify the presence of the stated features but not to preclude the presence or addition of further features in various embodiments of the invention.

[0225] Further, any reference herein to background art is not intended to imply that such background art forms or formed a part of the common general knowledge in any country.

1-76. (canceled)

77. An apparatus for transferring trigger signals from a first USB device to a plurality of second USB Devices attached to a USB network with minimal latency, the apparatus comprising:

circuitry adapted for communicating upstream toward a USB Host Controller and communicating downstream toward a plurality of USB devices; and

trigger engine circuitry adapted to observe communication traffic through said apparatus;

wherein said apparatus is adapted to decode upstream directed communication from said plurality of USB devices, to search for specific trigger request signals transmitted upstream from said USB network, and to transmit trigger command packets downstream toward said plurality of USB devices upon reception of said trigger request signals.

78. An apparatus as claimed in claim **77**, wherein said trigger engine circuitry is located inline upstream of said circuitry, and is configured to transmit said communication traffic upstream of said circuitry.

79. An apparatus as claimed in claim **77**, further comprising multiplexer circuitry adapted to selectively disable an upstream signal connection.

80. An apparatus as claimed in claim **77**, wherein

the circuitry comprises USB hub circuitry; and

the trigger engine circuitry is adapted to observe communication traffic upstream of said USB hub circuitry and to transmit communication packets downstream to an upstream port of said USB hub circuitry;

wherein said trigger engine is configured to decode upstream directed communication from a USB network connected downstream of said apparatus, to search for specific trigger request signals transmitted upstream from said USB network, and to transmit trigger command packets downstream toward said USB hub circuitry upon reception of said trigger request signals.

81. An apparatus as claimed in claim **80**, configured so that communication from upstream of said apparatus to said USB Hub circuitry passes through said trigger engine.

82. An apparatus as claimed in claim **81**, further comprising multiplexer circuitry adapted to selectively route either an upstream signal path or output of said trigger engine to said upstream port of said USB hub circuitry.

83. An apparatus as claimed in claim **77**, further comprising:

circuitry adapted to compare a plurality of trigger request signals with a predefined Combinatorial Trigger State;

wherein said apparatus is adapted to decode upstream directed communication from said USB devices, to search for specific trigger request signals transmitted upstream from said USB devices, and to transmit trigger command packets downstream toward said USB devices upon reception of said plurality of trigger request signals in correct combinatorial relationship.

84. An apparatus as claimed in claim **83**, wherein said combinatorial trigger state includes time dependence.

85. An apparatus as claimed in claim **83**, further adapted to:

open a communication pipe between a USB Host Controller and said USB devices; and

insert said trigger command into a data packet passing downstream in said communication pipe.

86. An apparatus as claimed in claim **83**, further adapted to:

selectively disconnect said USB devices upstream of a trigger device; and

transmit said trigger command downstream of said trigger engine.

87. An apparatus as claimed in claim **80**, wherein said USB devices include non-SuperSpeed Devices and said trigger command is broadcast by the trigger engine to said non-SuperSpeed USB devices.

88. An apparatus as claimed in claim **80**, wherein said USB devices include SuperSpeed USB Devices and said trigger command is selectively transmitted by a trigger engine to said SuperSpeed USB devices, each of said SuperSpeed USB devices being addressed using a USB Routing String.

89. A method of transferring trigger signals from a first USB device to one or more second USB devices attached to a USB network with minimal latency, the method comprising:

arming said first USB device to receive a Trigger Condition;

said first USB device detecting said Trigger Condition;

said first USB Device responding by transmitting a Trigger Request message upstream to a USB Host Controller of said USB network;

detecting passage of said Trigger Request with a Trigger Engine located in said USB network; and

said Trigger Engine responding to detection of said Trigger Request by transmitting a Trigger Command message downstream to said second USB devices.

90. A method as claimed in claim **89**, further comprising:

opening a communication pipe between a USB Host Controller and said plurality of USB devices; and

inserting said Trigger Command into a data packet passing downstream in said communication pipe.

91. A method as claimed in claim **89**, further comprising:

selectively disconnecting said plurality of USB devices upstream of said Trigger Engine; and

transmitting said Trigger Command downstream of said Trigger Engine.

92. A method as claimed in claim **89**, wherein said second USB devices include non-SuperSpeed Devices and said Trigger Command message is broadcast by the Trigger Engine to said non-SuperSpeed Devices.

93. A method as claimed in claim **89**, wherein said second USB devices include SuperSpeed USB Devices, wherein said Trigger Engine selectively transmits said Trigger Command to said SuperSpeed USB devices, each of said SuperSpeed USB devices being addressed using a USB Routing String.

94. A method as claimed in claim **89**, further comprising:

arming a plurality of said USB devices to receive respective trigger conditions;

configuring the trigger engine with a specific combinatorial trigger state, said combinatorial trigger state being a combination of trigger states of said USB devices,

said USB devices detecting said respective trigger conditions;

said USB devices transmitting respective trigger request messages upstream to the USB Host Controller of said USE network, said trigger request messages being indicative of respective trigger states of said USB devices;

detecting passage of said trigger requests with said trigger engine;

said trigger engine maintaining a record of a trigger status of each of said USB devices;

said trigger engine comparing said record of trigger status with said combinatorial trigger state;

said trigger engine transmitting a trigger command message downstream to said USB devices upon said trigger status matching said combinatorial trigger state.

95. A method as claimed in claim **94**, wherein said combinatorial trigger state includes time dependence, such that temporal relationships between transitions of said respective plurality of trigger states is important.

* * * * *