(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 June 2017 (29.06.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/112093 A1

(51) International Patent Classification:
G06F 13/38 (2006.01)

(21) International Application Number:
PCT/US2016/059932

(22) International Filing Date:
1 November 2016 (01.11.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/757,939 23 December 2015 (23.12.2015) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **MCGOWAN, Steven B.**; 2565 SW 112th Place, Portland, Oregon 97225 (US). **CHEN, Huimin**; 8412 NW Hawkins Boulevard, Portland, Oregon 97229 (US).

(74) Agents: **MALLIE, Michael J.** et al.; Blakely Sokoloff Taylor & Zafman Llp, 1279 Oakmead Parkway, Sunnyvale, California 94085 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: PTM FOR USB RETIMERS

(57) Abstract: A system and method of conducting precision time management in a universal serial bus system with a retimer. The method includes initiating, from the retimer, a link delay management request on an upstream-facing port of the retimer. The method further includes receiving, at a downstream-facing port of the retimer, a link delay management request and responding to the request received on the downstream-facing port.

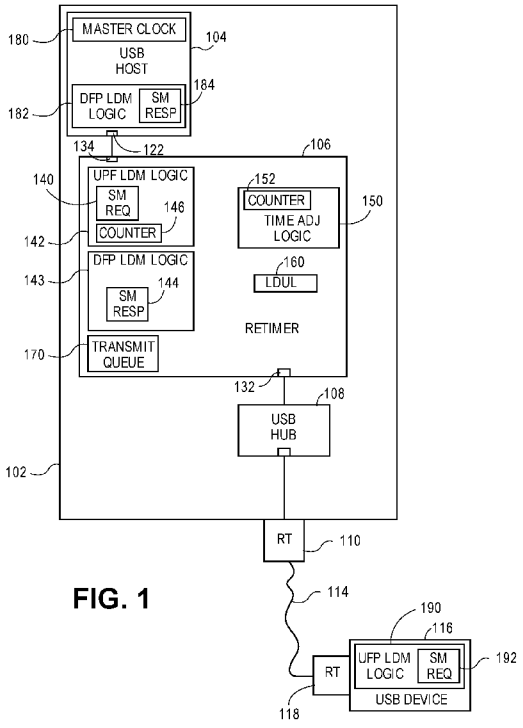

FIG. 1

WO 2017/112093 A1

PTM FOR USB RETIMERS

BACKGROUND

FIELD

Embodiments of the invention relate to Universal Serial Bus systems. More specifically, embodiments of the invention relate to precision time management in Universal Serial Systems with retimers.

BACKGROUND

Systems using Universal Serial Bus (USB) are ubiquitous in the modern computing environment. The most recent iteration of the USB follows The Universal Serial Bus 3.1 specification, published July 26, 2013 (USB Spec 3.1) provides for precision time management (PTM), which enables USB devices to maintain a very accurate local clock that is synchronized to a platform master clock. To extend the range of USB, retimers have been defined that re-synchronize and re-drive packets on a link to enable longer trace lines on PCBs and longer cables. Unfortunately, existing retimers introduce highly variable delays in the packets passing through the retimer, rendering PTM unusable for devices in the path of a retimer.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that different references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

Figure 1 is a block diagram of a system according to one embodiment of the invention.

Figures 2A and 2B are a flowchart of operation in a retimer according to one embodiment of the invention.

Figure 3 is a block diagram of an exemplary computer system according to one embodiment of the invention.

DETAILED DESCRIPTION

Figure 1 is a block diagram of a system according to one embodiment of the invention. A platform 102 has a Universal Serial Bus (USB) host 104 residing thereon. Host 104 maintains a master clock 180 and includes downstream facing port (DFP) link delay management (LDM) logic 182. Included in the DFP LDM logic 182 is a response state machine 184 that is responsible for responding to LDM requests received on the DFP 122 of the host 104.

The host 104 is connected through a retimer 106 to a hub 108 all residing on platform 102. Hub 108 is coupled through a retimer 110, which is instantiated in a cable 114 to USB device 114. For purposes of this application, all terms are used consistently with their meanings as set

forth in USB Spec 3.1. USB device 114 may be a speaker, a microphone, a camera or any other USB device.

In embodiments of the invention, to permit, for example, USB device 116 to participate in precision time management (PTM), retimer 106 participates in link delay management (LDM) exchanges. An LDM exchange has three components: the upstream link delay, the downstream link delay, and the delay through the host's LDM request state machine 184. The delay in state machine 184 is reported to the retimer 106 in a received LDM response packet (received on upstream facing port (UFP)134).

Thus, retimer 106 sends link delay management requests on its UFP 134 (to be received by downstream-facing port 122 of the USB host 104). Retimer 106 must also respond to link delay management requests received on its downstream-facing port (DFP) 132 from USB hub 108.

To facilitate the LDM exchanges, retimer 106 includes UPF LDM logic 142, which includes a link delay requester state machine 140, and a link delay responder state machine 144. UPF LDM logic 142 also includes a counter 146 to determine if a timeout condition has occurred without receiving a response to the request. Generally, the retimer sends an LDM request packet on UFP port 134. Retimer 106 retains a copy of the transmit timestamp that can then be compared to the timing information received in response from host 104 to determine the link delay between the host and retimer 106 (the upstream link delay).

When the host 104 receives the LDM request, its DFP LDM logic 184 starts a delay timer and queues an LDM response packet to be transmitted on the DFP 122. When the LDM response packet is transmitted, the host 104 records the value of the delay timer in the LDM response packet to record the delay introduced between receiving the LDM request packet and transmitting the LDM response packet.

The retimer 106 LDM request state machine 140 records the time that it receives the LDM response packet in the receive timestamp. Then the retimer subtracts the transmit timestamp from the receive timestamp to determine how long the LDM exchange took. It then subtracts the delay through the host's LDM request state machine 184 to determine the round trip link delay. Finally the retimer 106 divides the round trip link delay by two to determine an accurate indication of the delay across the upstream link. The upstream link delay is then retained for later use. According to one embodiment, link delay 160 in the upstream link may be retained in a non-volatile storage unit.

To handle LDM requests received on its DFP 132, retimer 106 includes DFP LDM logic 143 with a LDM response state machine 144. When retimer 106 receives a LDM request on its downstream-facing port 132, the retimer 106 uses responder state machine 144 to respond to the request, providing timing information indicating the time between receipt of the request and

transmission of the response out the downstream-facing port 132. This is analogous to the operation of the host response described above. The retimer's support for LDM exchanges on its upstream and downstream ports 134, 132, allow the retimer 106 to accurately determine the delay on the link with the host 104, and the hub 108 to accurately determine the delay on link between the hub 108 and retimer 106. Because the retimer 106 participates independently in the UFP and DFP LDM exchanges, the asymmetries and resulting link delay measurement errors that it would introduce between its upstream and downstream links absent its participation, are eliminated.

During normal operation, the retimer uses time adjustment logic 150 in conjunction with the previously-determined upstream link delay 160 to accurately adjust the timestamp fields of isochronous timestamp packets (ITPs) so that downstream USB entities can accurately discern the boundaries of microframes, thus permitting devices connected through the retimer(s) to use PTM consistent with USB Spec 3.1.

When an ITP is received on upstream-facing port 134, if the delayed bit is not set in the ITP, time adjustment logic 150 initiates the process of correcting the timestamp field in the ITP. Particularly, it resets an ITP delay counter 152 to zero. The ITP delay counter 152 is incremented by the PTM clock. The ITP is queued for transmission on downstream-facing port 132 on transmit queue 170. The queued packet timestamp information is modified in two respects. First, the value of the bus interval $\text{boundary}_{(\text{RXITP})}$ is copied to the bus interval $\text{boundary}_{(\text{TXITP})}$ subfield in the queued packet. Secondly, the time adjustment logic 252 calculates a delta timing value for the packet to be transmitted. In some embodiments, this calculation is given by Delta_{TX} = LDM Link Delay + $\text{Delta}_{(\text{RXITP})}$ + ITP Delay Counter – Correction $_{(\text{RXITP})}$. In other embodiments, the delta value is calculated without evaluating the correction value within the retimer, but instead merely copying the received correction value present in the ITP received on the UFP. Notably, the Delta value for the time at which the packet will be transmitted, not the time at which the packet is initially queued. The Delta value is inserted into the $\text{Delta}_{(\text{TXITP})}$ subfield if the ITP prior to transmission. The retimer 106 then calculates the cyclic redundancy check (CRC) values for the modified ITP.

Retimer 110 includes the same elements as 106 and performs the same link delay management exchanges and timing adjustments as described with references to retimer 106 above. Retimer 110 is instantiated in a USB cable connector for cable 114. For most cables of significant length, a second retimer 118 may be instantiated in a connector at the opposite end of the cable 114. Retimer 118 also includes the same element and performs the same link delay management exchanges and timing adjustments as described with references to retimer 106 above. That is; retimer 110 send LDM requests to hub 108 and responds to requests from

retimer 118. Retimer 118 send requests to retimer 110 and responds to requests from USB device 116. USB device 116 includes UFP LDM logic 190 including a LDM requestor state machine to initiate the requests to retimer 118. While the depicted embodiment shows three retimers in the path between host 104 and device 116, more or fewer retimers may be use. Additional retimers can be introduced to improve signal quality as the signal quality drops due to path length or other factors. Each retimer in the path desirably performs the LDM exchanges and time adjustment as describe above.

Figures 2A and 2B are a flowchart of operation in a retimer in one embodiment of the invention. In the retimers engaged in link delay management exchanges 200, the retimer transmits an LDM request on its upstream-facing port (UFP) at block 202Concurrently with transmission, the retimer starts a timeout counter. At block 204, the retimer stores the timestamp indicative of the time at which transmission on the upstream port of the link delay management request began. This stored time may be referred to as t_1. At decision block 206, a determination is made if a response has been received on the upstream-facing port. If no response has been received, a determination is made whether at timeout has occurred at decision block 208. If a timeout has occurred, the system may reattempt a link delay management request at block 202. Otherwise, the system continues to wait for a response.

If a response is received, the time the response is received is noted at block 210. The time of receipt of a response may be referred to as t_4. At block 212, the timing information in the response is retrieved. Generally speaking, the timestamp field in the response will provide an indication of the difference between the time at which the request was received at the downstream-facing port (t_2) of the receiver and the time at which that receiver transmitted the response (t_3). Thus, four times are available: the time the original request was sent, the two times defining receipt of request and response (this may be given as either the time of receipt and the time of response, or merely a difference value, e.g. $t_3 - t_2$), and the time at which the response was received by the retimer. By subtracting the time in the responder from the aggregate time from sending and receiving the response, and dividing by 2, a value for the link delay and the upstream link is obtained at block 214. That is, $LD = ((t_4-t_1)-(t_3-t_2))/2$. The value of the link delay in the upstream link is retained in the retimer for later use at block 216.

Concurrently with link delay management requests on the upstream facing port, the retimer may be responding to a link delay management request on its downstream-facing port (DFP). That is, at decision block 218, a determination is made whether a link delay management request has been received on the downstream-facing port. If no request has been received, no action is taken until a request is received. If a request is received, at block 220 the retimer transmits a response on its downstream facing port. Generally, the response will take the form of an LDM

response packet, including, a response delay field to provide an indication of the time between when the request was received at the downstream-facing port and when the response was actually transmitted on the downstream-facing port.. This allows downstream bus entities to correctly calculate delay in the link between the retimer and the requester.

Once link delay management exchanges have completed, in normal operation, at block 222, a determination is made if an ITP has been received. If an ITP has been received, a determination is made at block 224 whether the "delayed" bit in the ITP is set. If the delayed bit is set, at block 222, the ITP is forwarded on the downstream-facing port without modification at block 226. If the delayed bit is not set, the ITP delay counter in the retimer is set to zero at block 228. At block 230, the ITP is queued for retransmission on the downward-facing port. At block 232, the queued ITP is modified by copying a received interval boundary into the transmit interval boundary subfield of the queued ITP. That is, bus interval $\text{boundary}_{(\text{RXITP})}$ is copied to bus interval $\text{boundary}_{(\text{TXITP})}$ subfield of the isochronous timestamp field in the queued ITP.

At block 234, a Delta value is calculated for the queued ITP. The Delta value is equal to the LDM link delay (for the upstream port) + the Delta present in the received ITP field $\text{Delta}_{(\text{RXITP})}$ + the value of the ITP delay counter in the retimer – the correction value from the received packet; that is, $\text{correction}_{(\text{RXTIP})}$.

The value of the ITP used is the value when the packet will be transmitted, not the value when the packet is queued. If the calculated Delta is greater than or equal to 0, or less than 7500, the retimer inserts the Delta value in the Delta subfield $\text{Delta}_{(\text{TXITP})}$ and sets the correction value in the packet to be transmitted to be 0. That is, $\text{correction}_{(\text{TXITP})}$ is set to zero. If the Delta value calculated is greater than or equal to 7500, the retimer sets the Delta value equal to 7500 in the packet to be transmitted. If the Delta value calculated at the retimer sets the $\text{Delta}_{(\text{TXITP})}$ to 0, it calculates the correction value of the negative of the Delta value and inserts that into the ITP packet. Thereafter, at block 238, the cyclic redundancy check values for the ITP are recalculated for the modified ITP. After modification, the packet is transmitted on the downstream-facing port 240, and the process continues. In some cases, an ITP may be received before the LDM exchange has occurred. In such cases, the retime will simply set the "delayed" bit in the ITP and forward it downstream.

Turning to **Figure 3**, a block diagram of an exemplary computer system formed with a processor that includes execution units to execute an instruction, where one or more of the interconnects implement one or more features in accordance with one embodiment of the present invention is illustrated. System 300 includes a component, such as a processor 302, to employ execution units including logic to perform algorithms for process data, in accordance with the present invention, such as in the embodiment described herein. System 300 is representative of

processing systems based on the PENTIUM III™, PENTIUM 4™, Xeon™, Itanium, XScale™ and/or StrongARM™ microprocessors available from Intel Corporation of Santa Clara, California, although other systems (including PCs having other microprocessors, engineering workstations, set-top boxes and the like) may also be used. In one embodiment, sample system 300 executes a version of the WINDOWS™ operating system available from Microsoft Corporation of Redmond, Washington, although other operating systems (UNIX and Linux for example), embedded software, and/or graphical user interfaces, may also be used. Thus, embodiments of the present invention are not limited to any specific combination of hardware circuitry and software.

Embodiments are not limited to computer systems. Alternative embodiments of the present invention can be used in other devices such as handheld devices and embedded applications. Some examples of handheld devices include cellular phones, Internet Protocol devices, digital cameras, personal digital assistants (PDAs), and handheld PCs. Embedded applications can include a micro controller, a digital signal processor (DSP), system on a chip, network computers (NetPC), set-top boxes, network hubs, wide area network (WAN) switches, or any other system that can perform one or more instructions in accordance with at least one embodiment.

In this illustrated embodiment, processor 302 includes one or more execution units 308 to implement an algorithm that is to perform at least one instruction. One embodiment may be described in the context of a single processor desktop or server system, but alternative embodiments may be included in a multiprocessor system. System 300 is an example of a 'hub' system architecture. The computer system 300 includes a processor 302 to process data signals. The processor 302, as one illustrative example, includes a complex instruction set computer (CISC) microprocessor, a reduced instruction set computing (RISC) microprocessor, a very long instruction word (VLIW) microprocessor, a processor implementing a combination of instruction sets, or any other processor device, such as a digital signal processor, for example. The processor 302 is coupled to a processor bus 310 that transmits data signals between the processor 302 and other components in the system 300. The elements of system 300 (e.g., graphics accelerator 312, memory controller hub 316, memory 320, I/O controller hub 344, wireless transceiver 326, Flash BIOS 328, Network controller 334, Audio controller 336, Serial expansion port 338, I/O controller 340, etc.) perform their conventional functions that are well known to those familiar with the art.

In one embodiment, the processor 302 includes a Level 1 (L1) internal cache memory 304. Depending on the architecture, the processor 302 may have a single internal cache or multiple levels of internal caches. Other embodiments include a combination of both internal and

external caches depending on the particular implementation and needs. Register file 306 is to store different types of data in various registers including integer registers, floating point registers, vector registers, banked registers, shadow registers, checkpoint registers, status registers, and instruction pointer register.

Execution unit 308, including logic to perform integer and floating point operations, also resides in the processor 302. The processor 302, in one embodiment, includes a microcode (ucode) ROM to store microcode, which, when executed, is to perform algorithms for certain macroinstructions or handle complex scenarios. Here, microcode is potentially updateable to handle logic bugs/fixes for processor 302. For one embodiment, execution unit 308 includes logic to handle a packed instruction set 309. By including the packed instruction set 309 in the instruction set of a general-purpose processor 302, along with associated circuitry to execute the instructions, the operations used by many multimedia applications may be performed using packed data in a general-purpose processor 302. Thus, many multimedia applications are accelerated and executed more efficiently by using the full width of a processor's data bus for performing operations on packed data. This potentially eliminates the need to transfer smaller units of data across the processor's data bus to perform one or more operations, one data element at a time.

Alternate embodiments of an execution unit 308 may also be used in micro controllers, embedded processors, graphics devices, DSPs, and other types of logic circuits. System 300 includes a memory 320. Memory 320 includes a dynamic random access memory (DRAM) device, a static random access memory (SRAM) device, flash memory device, or other memory device. Memory 320 stores instructions and/or data represented by data signals that are to be executed by the processor 302.

Note that any of the aforementioned features or aspects of the invention may be utilized on one or more interconnects illustrated in FIG. 3. For example, an on-die interconnect (ODI), which is not shown, for coupling internal units of processor 302 implements one or more aspects of the invention described above. Or the invention is associated with a processor bus 310 (e.g., Intel Quick Path Interconnect (QPI) or other known high performance computing interconnect), a high bandwidth memory path 318 to memory 320, a point-to-point link to graphics accelerator 312 (e.g., a Peripheral Component Interconnect express (PCIe) compliant fabric), a controller hub interconnect 322, an I/O or other interconnect (e.g., USB, PCI, PCIe) for coupling the other illustrated components. Some examples of such components include the audio controller 336, firmware hub (flash BIOS) 328, wireless transceiver 326, data storage 324, legacy I/O controller 310 containing user input and keyboard interfaces 342, a serial expansion port 338 such as Universal Serial Bus (USB), and a network controller 334. The data storage device 324 can

comprise a hard disk drive, a floppy disk drive, a CD-ROM device, a flash memory device, or other mass storage device.

The following examples pertain to further embodiments. The various features of the different embodiments may be variously combined with some features included and others excluded to suit a variety of different applications. Some embodiments pertain to a retimer for a universal serial bus with an upstream-facing port (UFP) and a downstream-facing port (DFP). The retimer includes logic to determine a link delay on an upstream link.

In further embodiments, the logic initiates a link delay management exchange on the upstream port.

In further embodiments, the logic sends a link management packet out the upstream-facing port.

In further embodiments, the retimer has logic to respond to a link delay management exchange received on the downstream-facing port.

In further embodiments, the logic to determine the link day includes a link delay management requestor state machine and a timeout counter.

In further embodiments, the retimer has timing adjustment logic to aggregate upstream link delay, transmission delay in the retimer, and retimer delay, and adjust a master clock timing information of an isochronous timestamp packet by the aggregate.

In further embodiments, the time adjustment logic includes an isochronous timestamp packet delay counter.

Some embodiments pertain to a universal serial bus (USB) system with a USB host, a USB device, and a retimer having an upstream-facing port directed towards the host and a downstream facing port directed toward the device, the retimer including logic to initiate a link delay management exchange on the upstream-facing port.

Further embodiments have a USB hub between the host and the retimer.

In further embodiments, the retimer has logic to respond directly to a link delay management request received on the downstream-facing port.

In further embodiments, the retimer is instantiated in a USB cable.

In further embodiments, the logic to initiate the link delay management exchange includes a state machine and a counter.

Some embodiments pertain to method of conducting precision time management in a universal serial bus system with a retimer. The method includes initiating, from the retimer, a link delay management request on an upstream-facing port of the retimer. The method further includes receiving, at a downstream-facing port of the retimer, a link delay management request and responding to the request received on the downstream-facing port.

In further embodiments, initiating includes transmitting a link delay management packet on the upstream-facing port and maintaining a record of a time the transmitting began.

In further embodiments, the method includes receiving a response link delay management packet on the upstream-facing port in response to the transmitting, calculating the link delay from the data in the response link delay management packet and maintaining a record of the upstream link delay in the retimer.

In further embodiments, responding includes transmitting a response link delay management packet on a downstream-facing port including timing information indicating a time interval between receipt of the request and the transmitting of the response.

In further embodiments, after the link delay management exchange completes, the method includes receiving an isochronous timestamp packet on the upstream-facing port of the retime and modifying, in the retimer, the isochronous timestamp packet based, at least in part, on an upstream link delay determined from the link delay management request. The modified packet is then transmitted on the downstream-facing port of the retimer.

In further embodiments, the method includes maintaining an isochronous timestamp packet delay counter and incrementing the counter responsive to a precision time management clock.

In further embodiments, the method includes verifying that a delayed bit is not set in the received isochronous timestamp packet and setting the counter to zero responsive to the verifying. The method also includes queuing the received ITP packet for transmission on the downstream-facing port of the retimer and using values at transmission time for the modifying.

In further embodiments, modifying includes copying a received bus interval boundary to a transmit bus interval boundary sub-field of an isochronous timestamp field of the isochronous timestamp packet. Modifying also includes calculating a delta value for a time the isochronous timestamp packet will be transmitted and inserting the delta value into a delta sub-field of the isochronous timestamp field.

In further embodiments, the delta value is given by the formula:

$$\Delta_{tx} = \text{link delay} + \Delta_{rx} + \text{ITP delay counter} - \text{correction}_{rx}.$$

In further embodiments, the method includes recalculating a cyclical redundancy check for the isochronous timestamp packet as modified.

In further embodiments, the method includes identifying that the delayed bit is set in the received isochronous timestamp packet and transmitting the received packet on the downstream port without modification.

While embodiments of the invention are discussed above in the context of flow diagrams reflecting a particular linear order, this is for convenience only. In some cases, various operations may be performed in a different order than shown or various operations may occur in

parallel. It should also be recognized that some operations described with respect to one embodiment may be advantageously incorporated into another embodiment. Such incorporation is expressly contemplated.

In the foregoing specification, the invention has been described with reference to the specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A retimer for a universal serial bus comprising:
an upstream-facing port (UFP);
a downstream-facing port (DFP); and
logic to determine a link delay on an upstream link.

2. The retimer of claim 1, wherein the logic initiates a link delay management exchange on the upstream port.

3. The retimer of claim 1, wherein the logic sends a link management packet out the upstream-facing port.

4. The retimer of claim 1, further comprising:
logic to respond to a link delay management exchange received on the downstream-facing port.

5. The retimer of claims 1 or 4, wherein the logic to determine the link day comprises:
a link delay management requestor state machine; and
a timeout counter.

6. The retimer of claims 1 or 4, further comprising:
timing adjustment logic to aggregate upstream link delay, transmission delay in the retimer, and retimer delay, and adjust a master clock timing information of an isochronous timestamp packet by the aggregate.

7. The retimer of claim 6, wherein the time adjustment logic comprises:
an isochronous timestamp packet delay counter.

8. A universal serial bus (USB) system comprising:
a USB host;
a USB device;
a retimer having an upstream-facing port directed towards the host and a downstream facing port directed toward the device, the retimer including logic to initiate a link delay management exchange on the upstream-facing port.

9. The system of claim 8, further comprising:
a USB hub between the host and the retimer.

10. The system of claim 8, wherein the retimer comprises:
logic to respond directly to a link delay management request received on the downstream-facing port.

11. The system of claims 8 or 10, wherein the retimer is instantiated in a USB cable.

12. The system of claims 8 or 10, wherein the logic to initiate the link delay management exchange comprises:

a state machine; and

a counter.

13. A method of conducting precision time management in a universal serial bus system with a retimer, the method comprising:

initiating, from the retimer, a link delay management request on an upstream-facing port of the retimer;

receiving, at a downstream-facing port of the retimer, a link delay management request; and

responding to the request received on the downstream-facing port.

14. The method of claim 13, wherein initiating comprises:

transmitting a link delay management packet on the upstream-facing port; and

maintaining a record of a time the transmitting began.

15. The method of claim 14, further comprising:

receiving response link delay management packet on the upstream-facing port in response to the transmitting;

calculating the link delay from the data in the response link delay management packet; and

maintaining a record of the upstream link delay in the retimer.

16. The method of claim 13, wherein responding comprises:

transmitting an response link delay management on a downstream-facing port including timing information indicating a time interval between receipt of the request and the transmitting of the response.

17. The method of claim 13, wherein after the link delay management exchange completes, the method further comprises:

receiving an isochronous timestamp packet on the upstream-facing port of the retimer;

modifying, in the retimer, the isochronous timestamp packet based, at least in part, on an upstream link delay determined from the link delay management request; and

transmitting the packet as modified on the downstream-facing port of the retimer.

18. The method of claim 17, further comprising:

maintaining an isochronous timestamp packet delay counter; and

incrementing the counter responsive to a precision time management clock.

19. The method of claim 18, further comprising:

verifying that a delayed bit is not set in the received isochronous timestamp packet;

setting the counter to zero responsive to the verifying;

queuing the receipt packet for transmission on the downstream-facing port of the retimer; and

using values at transmission time for the modifying.

20. The method of claim 17, wherein modifying comprises:

copying a received bus interval boundary to a transmit bus interval boundary sub-field of an isochronous timestamp field of the isochronous timestamp packet;

calculating a delta value for a time the isochronous timestamp packet will be transmitted; and

inserting the delta value into a delta sub-field of the isochronous timestamp field.

21. The method of claim 20, wherein the delta value is given by the formula:

Δ_{tx} = link delay + Δ_{rx} + ITP delay counter – $\text{correction}_{rx.}$

22. The method of claim 13, further comprising:

recalculating a cyclical redundancy check for the isochronous timestamp packet as modified.

23. The method of claim 13, further comprising:

identifying the delayed bit is set in the received isochronous timestamp packet; and

transmitting the received packet on the downstream facing port without modification.

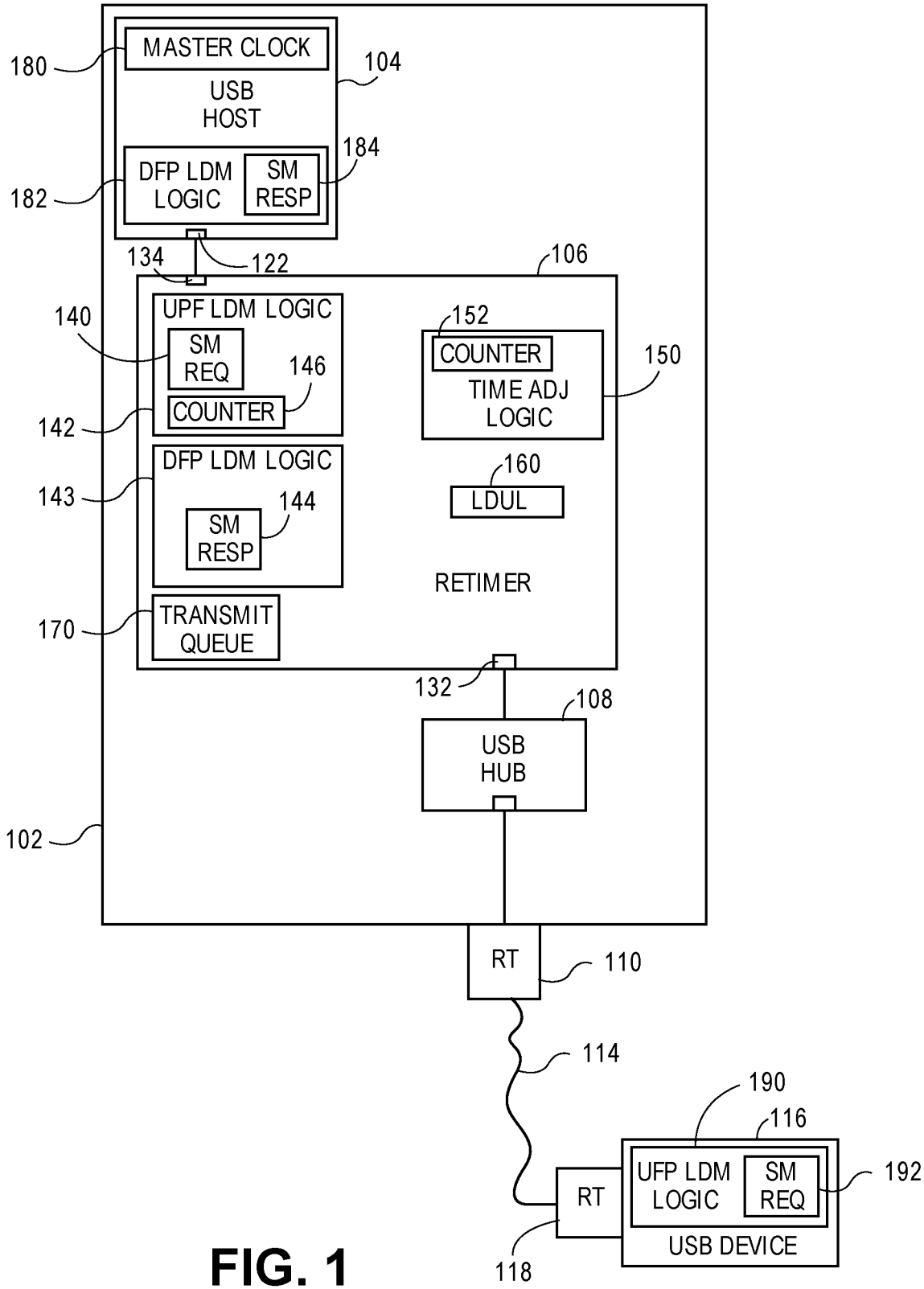
MASTER CLOCK
180
104
USB
HOST
184
DFP LDM
LOGIC
SM
RESP
182
134
122
106
UPF LDM LOGIC
140
SM
REQ
146
COUNTER
142
152
COUNTER
150
TIME ADJ
LOGIC
DFP LDM LOGIC
143
160
LDUL
144
SM
RESP
RETIMER
TRANSMIT
QUEUE
170
132
108
USB
HUB
102
RT
110
114
190
116
UFP LDM
LOGIC
SM
REQ
192
RT
USB DEVICE
118

FIG. 1

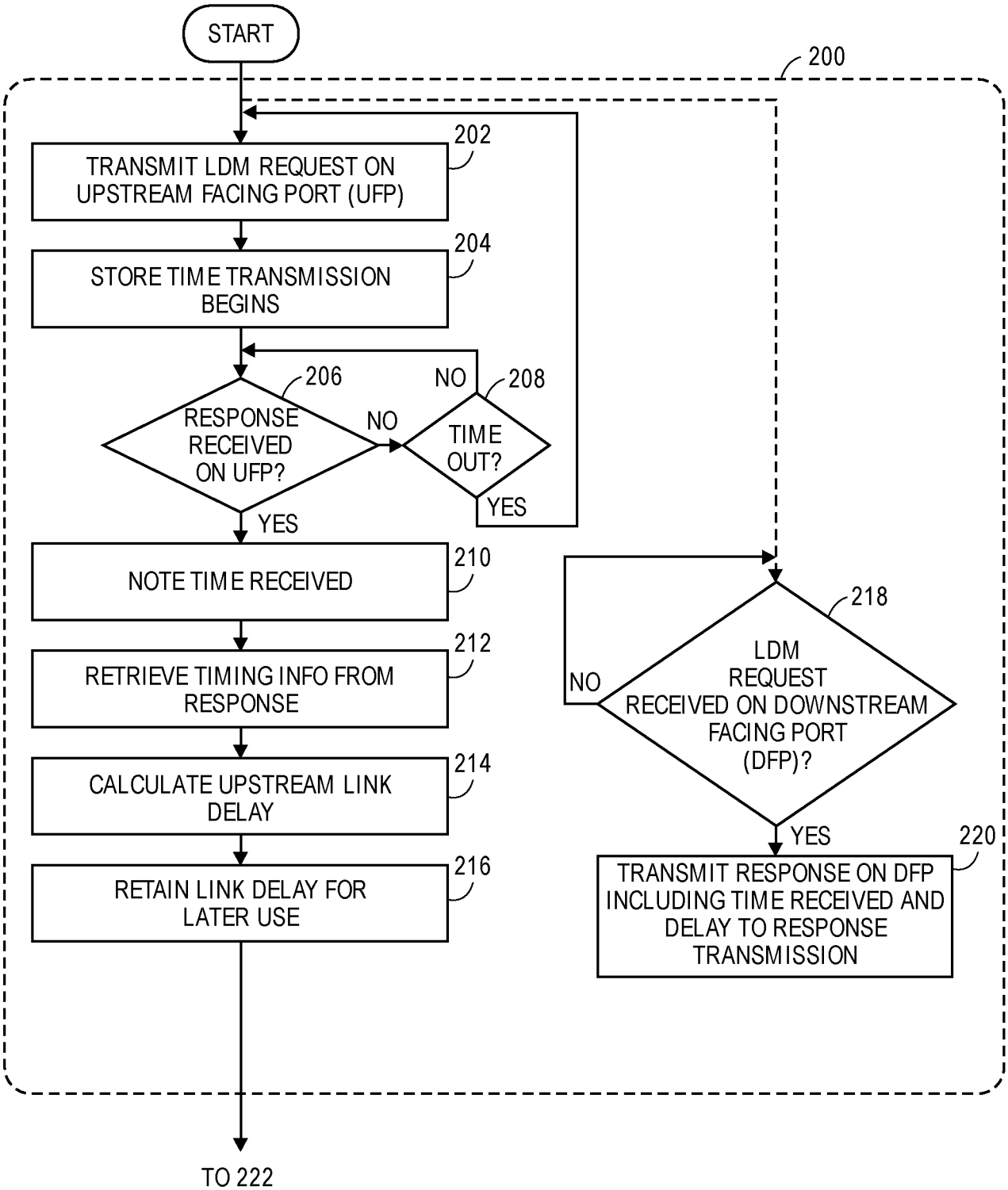
START
200
202
TRANSMIT LDM REQUEST ON UPSTREAM FACING PORT (UFP)
204
STORE TIME TRANSMISSION BEGINS
206
RESPONSE RECEIVED ON UFP?
NO
208
TIME OUT?
NO
YES
YES
210
NOTE TIME RECEIVED
212
RETRIEVE TIMING INFO FROM RESPONSE
214
CALCULATE UPSTREAM LINK DELAY
216
RETAIN LINK DELAY FOR LATER USE
TO 222
218
LDM REQUEST RECEIVED ON DOWNSTREAM FACING PORT (DFP)?
NO
YES
220
TRANSMIT RESPONSE ON DFP INCLUDING TIME RECEIVED AND DELAY TO RESPONSE TRANSMISSION

FIG. 2A

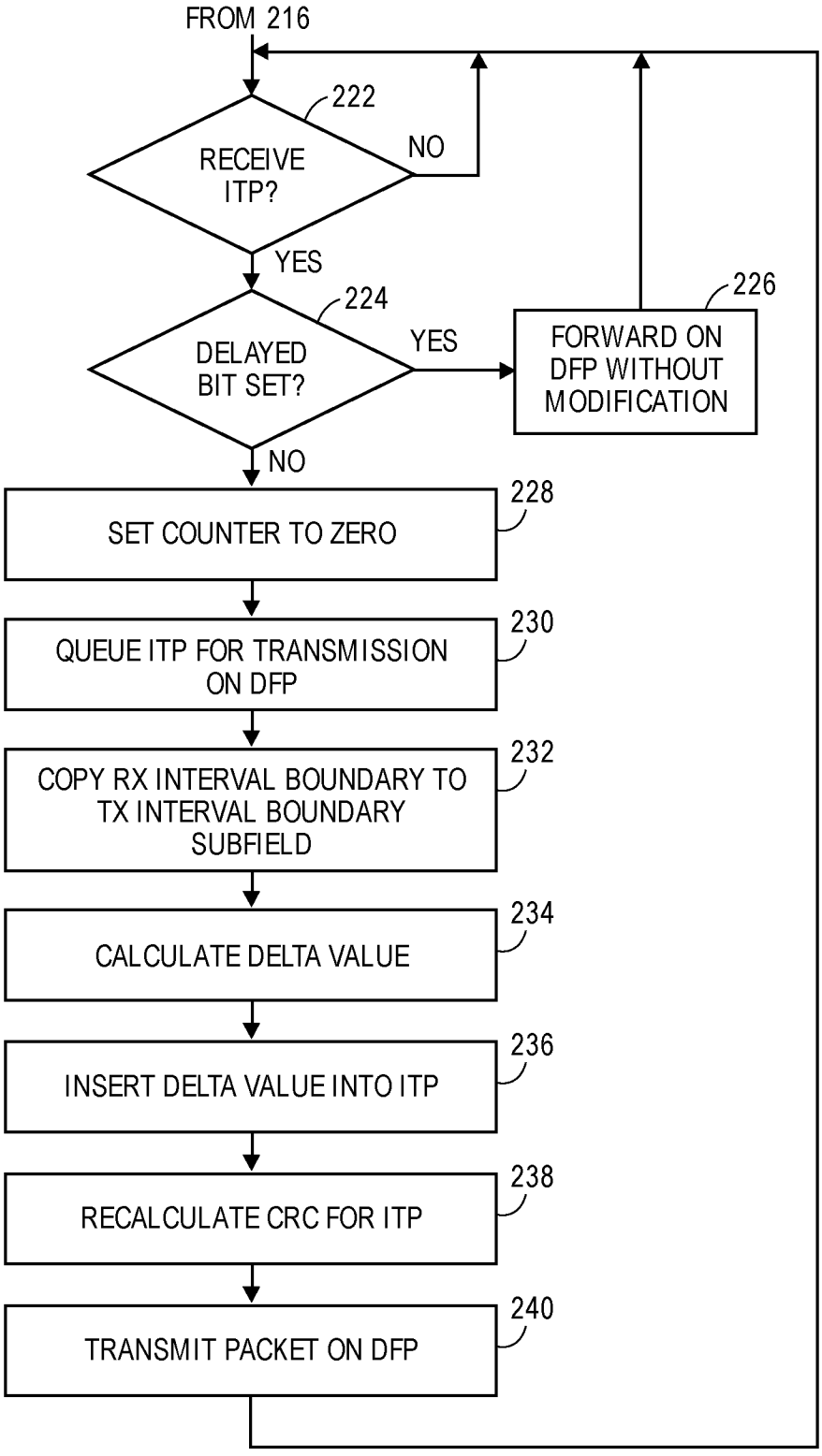
FROM 216
222
RECEIVE ITP?
NO
YES
224
DELAYED BIT SET?
YES
226
FORWARD ON DFP WITHOUT MODIFICATION
NO
228
SET COUNTER TO ZERO
230
QUEUE ITP FOR TRANSMISSION ON DFP
232
COPY RX INTERVAL BOUNDARY TO TX INTERVAL BOUNDARY SUBFIELD
234
CALCULATE DELTA VALUE
236
INSERT DELTA VALUE INTO ITP
238
RECALCULATE CRC FOR ITP
240
TRANSMIT PACKET ON DFP

FIG. 2B

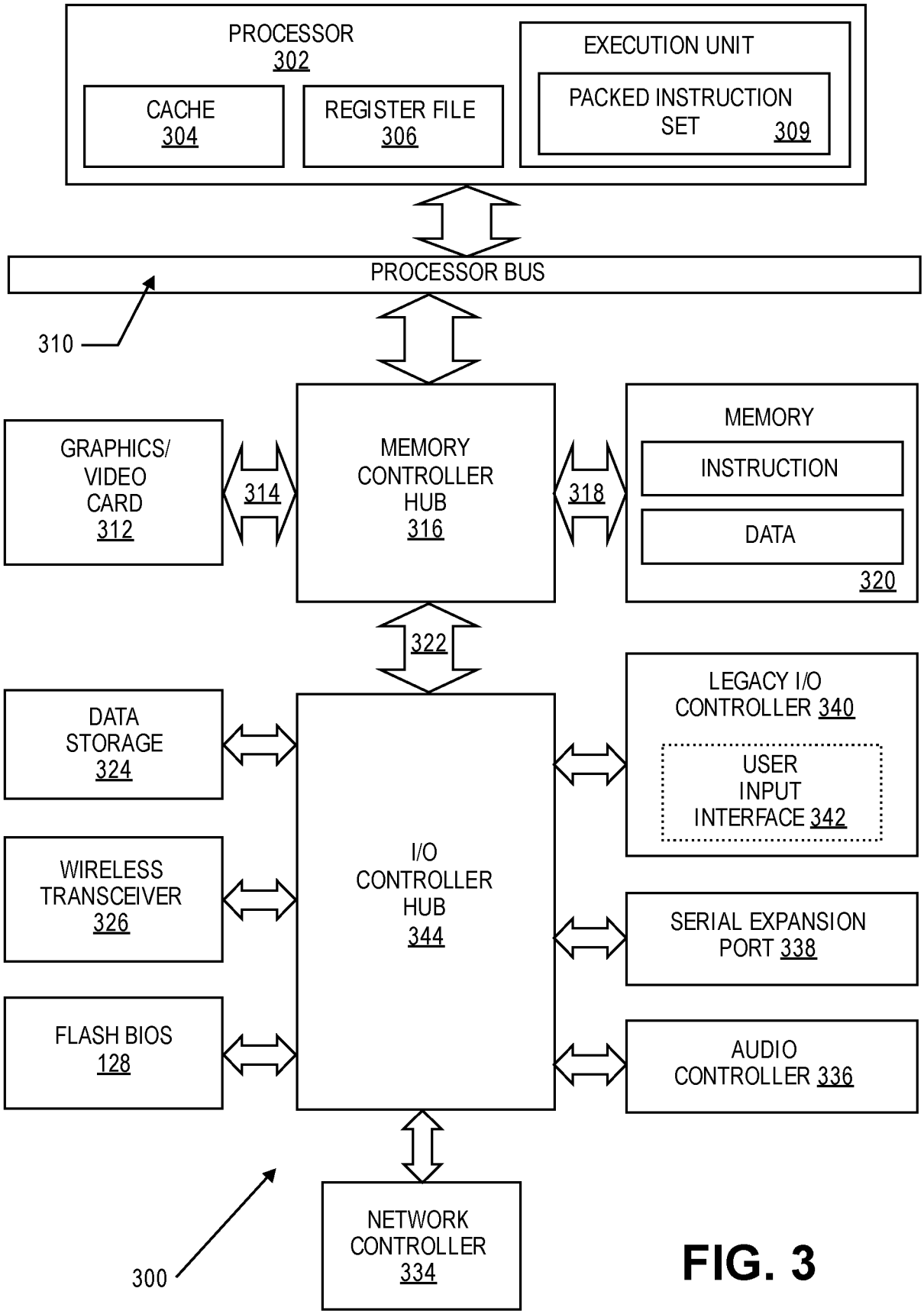
PROCESSOR
302
CACHE
304
REGISTER FILE
306
EXECUTION UNIT
PACKED INSTRUCTION SET
309
PROCESSOR BUS
310
GRAPHICS/ VIDEO CARD
312
314
MEMORY CONTROLLER HUB
316
318
MEMORY
INSTRUCTION
DATA
320
322
DATA STORAGE
324
WIRELESS TRANSCEIVER
326
FLASH BIOS
128
I/O CONTROLLER HUB
344
LEGACY I/O CONTROLLER 340
USER INPUT INTERFACE 342
SERIAL EXPANSION PORT 338
AUDIO CONTROLLER 336
NETWORK CONTROLLER
334
300

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/059932

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/38(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 13/38; H03L 7/00; G06F 5/00; H03D 3/24; G11C 7/22

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: universal serial bus (USB) 3.1, delay link, determining, retimer, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	HEWLETT-PACKARD COMPANY et al., `Universal Serial Bus 3.1 Appendix E,` revision 1.0, 11 March 2015 Retrived from <https://doc.xdevs.com/doc/Standards/USB 3.1/usb_31_030215> See sections E.1.1-E.2.1 and E.2.3; and figures E-1, E-2, and E-3.	1-23
Y	HEWLETT-PACKARD COMPANY et al., `Universal Serial Bus 3.1 Specification,` revision 1.0, 26 July 2013 Retrived from <http://www.usb.org/developers/docs></Retrived> See sections 6.4.3.2., 8.4.6, and 8.4.8; and figure 8-11.	1-23
Y	US 2011-0058433 A1 (HAE-RANG CHOI et al.) 10 March 2011 See paragraphs [0027]-[0028]; claim 1; and figure 2.	6-7
A	US 2006-0064522 A1 (ADAM MARK WEIGOLD et al.) 23 March 2006 See paragraphs [0140]-[0149] and figures 3-4.	1-23
A	US 2008-0056426 A1 (XIAOMIN SI et al.) 06 March 2008 See paragraphs [0036]-[0041] and figure 2.	1-23

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 February 2017 (16.02.2017)	**16 February 2017 (16.02.2017)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2016/059932

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0058433 A1	10/03/2011	JP 2011-060409 A	24/03/2011
		KR 10-1045087 B1	29/06/2011
		KR 10-2011-0025393 A	10/03/2011
		US 8144531 B2	27/03/2012
US 2006-0064522 A1	23/03/2006	AT 410736 T	15/10/2008
		AT 459053 T	15/03/2010
		AT 487183 T	15/11/2010
		AT 492850 T	15/01/2011
		AU 2003-243837 A1	02/02/2004
		AU 2003-243837 B2	12/08/2010
		AU 2009-201930 A1	04/06/2009
		AU 2009-201930 B2	14/04/2011
		AU 2010-202685 A1	15/07/2010
		AU 2010-202685 B2	07/03/2013
		CA 2492811 A1	22/01/2004
		CA 2492811 C	08/05/2012
		CA 2749078 A1	22/01/2004
		CN 101281512 A	08/10/2008
		CN 101794269 A	04/08/2010
		CN 101794269 B	22/05/2013
		CN 1679013 A	05/10/2005
		CN 1679013 B	28/04/2010
		EP 1535170 A1	01/06/2005
		EP 1535170 A4	06/09/2006
		EP 1535170 B1	08/10/2008
		EP 1901178 A1	19/03/2008
		EP 1901178 B1	24/02/2010
		EP 1959349 A2	20/08/2008
		EP 1959349 A3	17/12/2008
		EP 1959349 B1	22/12/2010
		EP 2040175 A1	25/03/2009
		EP 2040175 B1	03/11/2010
		EP 2161666 A2	10/03/2010
		EP 2161666 A3	26/05/2010
		EP 2270676 A1	05/01/2011
		HK 1079302 A1	03/09/2010
		HK 1147133 A1	07/02/2014
		JP 2005-533305 A	04/11/2005
		JP 2009-289292 A	10/12/2009
		JP 2010-182337 A	19/08/2010
		JP 2013-254510 A	19/12/2013
		JP 5232110 B2	10/07/2013
		JP 5378297 B2	25/12/2013
		JP 5642852 B2	17/12/2014
		US 2004-0088445 A1	06/05/2004
		US 2013-191562 A1	25/07/2013
		US 7539793 B2	26/05/2009
		US 8285897 B2	09/10/2012

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/059932

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		WO 2004-008330 A1	22/01/2004
US 2008-0056426 A1	06/03/2008	CN 101136739 A	05/03/2008
		CN 101136739 B	11/08/2010
		TW 200812243 A	01/03/2008
		US 7672417 B2	02/03/2010
		WO 2008-027066 A2	06/03/2008
		WO 2008-027066 A3	24/06/2010