



(51) International Patent Classification:

G06F 9/54 (2006.01) G06F 3/041 (2006.01)
G06F 13/42 (2006.01) G09G 5/00 (2006.01)

(21) International Application Number:

PCT/US2018/030507

(22) International Filing Date:

01 May 2018 (01.05.2018)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/502,329 05 May 2017 (05.05.2017) US

(71) Applicant: MICROCHIP TECHNOLOGY INCORPORATED [US/US]; 2355 West Chandler Blvd., Chandler, AZ 85224 (US).

(72) Inventors: LUND, Morten, Werner; Nedre Flatasveg 3B 03-01, 7099 Flatasen (NO). CLARK, Lloyd; Biskop Skaars Gate 2, 7051 Trondheim (NO). REITAN, Odd, Magne; Kattensvegen 17, 7080 Heimdal (NO).

(74) Agent: BACA, Andrew, J. et al.; Traskbritt P.C., P.O. Box 2550, Salt Lake City, UT 84110-2550 (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, JO, JP, KE, KG, KH, 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).

(54) Title: DEVICES AND METHODS FOR TRANSMISSION OF EVENTS WITH A UNIFORM LATENCY ON SERIAL COMMUNICATION LINKS

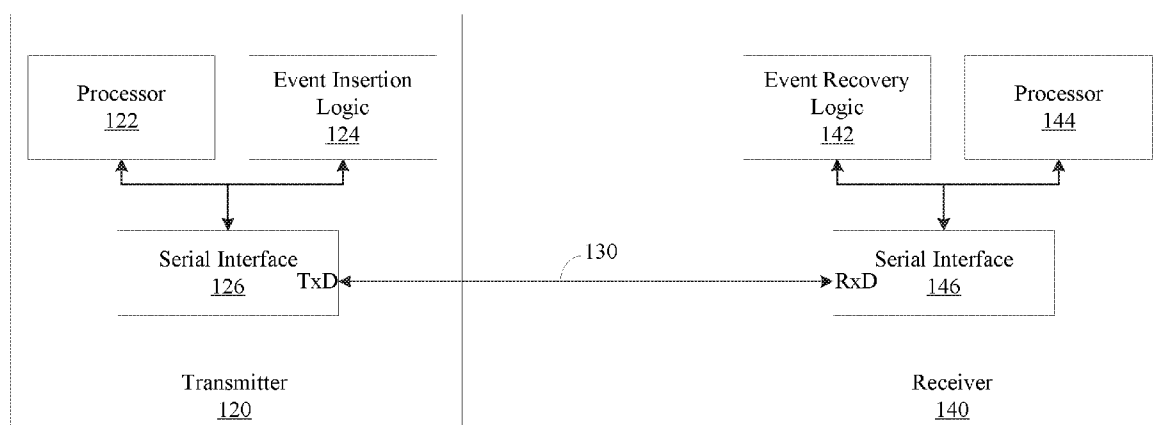


FIG. 1A

(57) Abstract: The present disclosure relates generally to serial communication links and, more specifically, to events communicated on serial communication links and the timing of those events, for example, to achieve uniform delay among multiple event transmissions.

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

DEVICES AND METHODS FOR TRANSMISSION OF EVENTS WITH A UNIFORM LATENCY ON SERIAL COMMUNICATION LINKS

5 CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit under 35 U.S.C. §119(e) of U.S. Provisional Patent Application Serial No. 62/502,329, filed May 5, 2017, the disclosure of which is hereby incorporated herein in its entirety by this reference.

10 TECHNICAL FIELD

Embodiments of the present disclosure relate generally to serial communication links and, more specifically, to events communicated on serial communication links and the timing of those events.

15 BACKGROUND

In many embedded control systems, and other computing systems, movement of data between peripheral devices and a host, or between peripheral devices, may be a significant amount of data traffic on the various buses that may exist in such systems. Moreover, some of this data traffic may include information related to events that occur
20 and timing of these events.

In conventional inter-chip communication, one approach is to communicate such event information on dedicated lines signaling the events to manage the timing of the event communication. However, there is typically an extra cost for additional lines. The cost of adding lines might be high and even be prohibitive due to layout constraints. Another
25 approach is to send the event information as soon as possible as the next communication packet on a serial communication link. However, this approach may lose important event details, for example, timing details about when an actual event occurred. Other deficiencies and limitations in these and other approaches may exist.

There is a need for communication of events and event timing details on serial
30 communication links to indicate relative timing of events between a master and one or more slaves.

DISCLOSURE

Some embodiments of the present disclosure relate, generally, to a serial communication link transmitter. The serial communication link transmitter may include delay circuitry and transmission circuitry. The delay circuitry may be configured to
5 determine an event delay between a predefined bit position of an ongoing frame being transmitted and an event. The transmission circuitry may be configured to send an event frame after the ongoing frame, wherein the transmission circuitry is configured to include in the event frame, delay bits corresponding to the event delay and event identifier bits indicative of the event to which the event frame corresponds.

10 Some embodiments of the present disclosure relate, generally, to a method of transmitting events over a serial communication link. The method may include determining an event delay between a predefined bit position of an ongoing frame being transmitted and an event; encoding an event frame corresponding to the event, wherein the event frame includes: delay bits corresponding to the event delay; and event identifier bits
15 that indicate the event frame corresponds to the event, and transmitting the event frame after the ongoing frame.

Some embodiments of the present disclosure relate, generally, to a serial communication link receiver. The serial communication link receiver may include a receive circuitry and a delay circuitry. The receive circuitry may be configured to receive
20 an event frame. In one embodiment, the event frame includes delay bits indicative of a delay between an event that occurred at a transmitter relative to a predefined bit position in a previous frame received from the transmitter. The delay circuitry may be configured to decode the delay bits; wait a number of clock cycles corresponding to the delay bits; and assert a receiver-side event after waiting the number of clock cycles.

25 Some embodiments of the present disclosure relate, generally, to a method of receiving events over a serial communication link. The method may include receiving an event frame; decoding delay bits from the event frame, the delay bits indicative of a delay between an event that occurred at a transmitter relative to a predefined bit position in a previous frame; waiting a number of clock cycles corresponding to the delay bits; and
30 asserting a receiver-side event after the waiting.

Some embodiments of the present disclosure relate, generally, to a serial communication link. The serial communication link may include a transmitter and a receiver configured to define a uniform latency between a transmitter-side event and a

receiver-side event. The transmitter may include delay circuitry, control circuitry, and transmission circuitry. The delay circuitry may be configured to receive the transmitter-side event and delay the transmitter-side event by a delay time corresponding to a frame time. The control circuitry may be configured to prevent additional frames from being
5 started while the transmitter-side event is being delayed. The transmission circuitry may be configured to send an event frame corresponding to the transmitter-side event after the delay time, wherein the transmission circuitry is configured to include in the event frame, event identifier bits indicating the frame being transmitted is an event frame. The receiver may include receiver circuitry configured to decode the frame being transmitted as the
10 event frame; and assert the receiver-side event responsive to the decoding.

BRIEF DESCRIPTION OF THE DRAWINGS

Advantages of the embodiments of the disclosure will be apparent to those of ordinary skill in the art from the following detailed description and the accompanying
15 drawings:

FIG. 1A shows a block diagram of a transmitter and a receiver with a serial communication link, according to an embodiment of the disclosure.

FIGS. 1B-1E show flowcharts showing processes for transmission of events over a serial communication link, according to embodiments of the disclosure.

20 FIG. 2A shows a detailed timing diagram illustrating transmission of certain events over a serial communication link with a delay count included in the event transmission, according to embodiments of the disclosure.

FIGS. 2B, 2C, and 2D show marked sections of FIG. 2A in an expanded view.

25 FIG. 3A shows a detailed timing diagram illustrating transmission of certain events over a serial communication link with a predetermined delay for the event transmission, according to embodiments of the disclosure.

FIGS. 3B and 3C show marked sections of FIG. 3A in an expanded view.

30 FIG. 3D shows a detailed timing diagram illustrating transmission of certain events over a serial communication link with a predetermined delay for the event transmission, according to embodiments of the disclosure.

FIGS. 3E and 3F show marked sections of FIG. 3D in an expanded view.

FIGS. 4A-4C show frame level timing diagrams illustrating different priority event timings and some errors that may occur over a serial communication link, according to embodiments of the disclosure.

FIG. 5 shows a logic diagram illustrating priority logic as an example for prioritizing events on a serial communication link, according to embodiments of the disclosure.

FIGS. 6A-6C show frame level timing diagrams illustrating event timings for prioritized events over a serial communication link, according to embodiments of the disclosure.

FIG. 7 shows a flowchart of a process for prioritizing events according to an embodiment of the disclosure.

FIG. 8 shows a block diagram of a touch panel system including a system controller, a touch controller, and a display panel with serial communication links according to an embodiment of the disclosure.

MODE(S) FOR CARRYING OUT THE INVENTION

In the following detailed description, reference is made to the accompanying drawings, which form a part hereof, and in which are shown, by way of illustration, specific example embodiments in which the present disclosure may be practiced. These embodiments are described in sufficient detail to enable a person of ordinary skill in the art to practice the present disclosure. However, other embodiments may be utilized, and structural, material, and process changes may be made without departing from the scope of the disclosure. The illustrations presented herein are not meant to be actual views of any particular method, system, device, or structure, but are merely idealized representations that are employed to describe the embodiments of the present disclosure. The drawings presented herein are not necessarily drawn to scale. Similar structures or components in the various drawings may retain the same or similar numbering for the convenience of the reader; however, the similarity in numbering does not mean that the structures or components are necessarily identical in size, composition, configuration, or any other property.

It will be readily understood that the components of the embodiments as generally described herein and illustrated in the drawings may be arranged and designed in a wide variety of different configurations. Thus, the following description of various

embodiments is not intended to limit the scope of the present disclosure, but is merely representative of various embodiments. While the various aspects of the embodiments may be presented in drawings, the drawings are not necessarily drawn to scale unless specifically indicated.

5 Furthermore, specific implementations shown and described are only examples and should not be construed as the only way to implement the present disclosure unless specified otherwise herein. Elements, circuits, and functions may be shown in block diagram form in order not to obscure the present disclosure in unnecessary detail. Conversely, specific implementations shown and described are exemplary only and should
10 not be construed as the only way to implement the present disclosure unless specified otherwise herein. Additionally, block definitions and partitioning of logic between various blocks is exemplary of a specific implementation. It will be readily apparent to one of ordinary skill in the art that the present disclosure may be practiced by numerous other partitioning solutions. For the most part, details concerning timing considerations and the
15 like have been omitted where such details are not necessary to obtain a complete understanding of the present disclosure and are within the abilities of persons of ordinary skill in the relevant art.

Those of ordinary skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For
20 example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout this description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof. Some drawings may illustrate signals as a single signal for clarity of presentation and description. It will be understood by a person of ordinary skill in the art
25 that the signal may represent a bus of signals, wherein the bus may have a variety of bit widths and the present disclosure may be implemented on any number of data signals including a single data signal.

The various illustrative logical blocks, modules, and circuits described in connection with the embodiments disclosed herein may be implemented or performed with
30 a general-purpose processor, a special-purpose processor, a Digital Signal Processor (DSP), an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions

described herein. A general-purpose processor (may also be referred to herein as a host processor or simply a host) may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, such as a
5 combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. A general-purpose computer including a processor is considered a special-purpose computer while the general-purpose computer is configured to execute computing instructions (e.g., software code) related to embodiments of the present disclosure.

10 Also, it is noted that the embodiments may be described in terms of a process that is depicted as a flowchart, a flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe operational acts as a sequential process, many of these acts may be performed in another sequence, in parallel, or substantially concurrently. In addition, the order of the acts may be re-arranged. A process may correspond to a method, a thread,
15 a function, a procedure, a subroutine, a subprogram, etc. Furthermore, the methods disclosed herein may be implemented in hardware, software, or both. If implemented in software, the functions may be stored or transmitted as one or more instructions or code on computer-readable media. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a
20 computer program from one place to another.

It should be understood that any reference to an element herein using a designation such as “first,” “second,” and so forth does not limit the quantity or order of those elements, unless such limitation is explicitly stated. Rather, these designations may be used herein as a convenient method of distinguishing between two or more elements or
25 instances of an element. Thus, a reference to first and second elements does not mean that only two elements may be employed there or that the first element must precede the second element in some manner. In addition, unless stated otherwise, a set of elements may comprise one or more elements.

In an effort to make details in figures clearer, certain marked sections of some
30 figures may be shown in an expanded view in another figures. In some cases section markings may obscure parts of a figure, but will be clear in the expanded view. Everything shown in an expanded view should be considered part of the corresponding figure, even some details that might be obscured in the corresponding figure by the section markings.

Further, any discussion of a figure in this disclosure also applies to its expanded views, if any.

As used herein, the term “substantially” in reference to a given parameter, property, or condition means and includes to a degree that one of ordinary skill in the art would understand that the given parameter, property, or condition is met with a small degree of variance, such as, for example, within acceptable manufacturing tolerances. By way of example, depending on the particular parameter, property, or condition that is substantially met, the parameter, property, or condition may be at least 90% met, at least 95% met, or even at least 99% met.

As used herein, “serial communication link” means a communication link that transmits information as a serial group of bits. The protocol of the link includes a group of bits as an information payload, which may be of various sizes and may include other bits such as, for example, start bits, stop bits, parity bits, and address bits. The physical layer of the link may be a wired bus, such as, for example, RS-232, I²C, and SMBus. The physical layer of the link also may be wireless signals such as, for example, Infrared Data Association (IrDA) signals.

As used herein, the term “frame” defines a group of predetermined number of bits transferred on a serial communication link. As one example, in serial communication links such as a Universal Asynchronous Receiver/Transmitter (UART), a Universal Synchronous Receiver/Transmitter (USRT), or a Universal Synchronous/Asynchronous Receiver/Transmitter (USART), a frame may be defined as 10 bits to include a start bit, an 8-bit data payload, a parity bit, and a stop bit. The frame for one of these serial communication protocols may also be different lengths, such as, for example only, 8 bits to include a start bit, a 7-bit data payload, and a stop bit. As another example, an I²C serial communication protocol (or other protocols with multiple slave devices) may include longer frame sizes to allow inclusion of a slave address as well as a data payload.

Reference throughout this specification to “one embodiment,” “an embodiment,” or similar language means that a particular feature, structure, or characteristic described in connection with the indicated embodiment is included in at least one embodiment of the present disclosure. Thus, the phrases “in one embodiment,” “in an embodiment,” and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.

Some embodiments described herein relate to techniques for providing a uniform latency between an occurrence of an event at a bus master and its reception at a slave. In some embodiments, the event is communicated on a serial bus and the indicator of the event arrives at a slave coupled to the serial bus. In some embodiments, the uniform
5 latency may be a fixed delay already known between the master and the slave. In other embodiments, the uniform latency may be communicated between the master and slave with timing information included. Still other embodiments described herein provide prioritization of multiple events that may occur during any given frame.

In dedicated serial communication systems, there is sometimes a need for
10 transmitting “side information” of certain occurrences (*e.g.*, events) between regular data communication packages (the transmission of the side information is referred to, herein, as “event transmission”). The event transmission should not destroy the main data communication packages, but the event transmissions should still identify uniquely the time of the event. As an example, the communication link may be based on a UART, or its
15 synchronous version USRT, and the communication may be a U(S)ART frame.

Further, if a system supports multiple such events, then the system, according to one embodiment of the disclosure, will prioritize if two or more events occur too frequently (*e.g.*, close in time) to be transmitted in individual frames and still provide the correct timing information. The present disclosure describes systems, devices and methods to
20 prioritize these events in a way that seeks to ensure the highest priority event is transmitted even if a lower priority event comes first, but too close in time to be able to complete the transmission of the low priority event before one has to start the transmission of the high priority event.

Even though the main purpose of a communication link may be to transfer a certain
25 type of data, the transmitter may need to inform the receiver of certain events taking place at the transmitter. A non-limiting system example is a microcontroller (MCU) controlling multiple complex display drivers on a display, such as, for example, a Liquid Crystal Display, (LCD), an Organic Light-Emitting Diode (OLED) display, etc. The display drivers might have complex circuitry for capacitive touch measurement requiring
30 configuration and control by the MCU. Timing information (*e.g.*, events) like horizontal synchronization (HSYNC) and vertical synchronization (VSYNC) might be necessary to time (*e.g.*, synchronize) touch operations to the update rate of the display, for example, to compensate for the noise introduced by the display drivers.

Although embodiments of the disclosure may refer to “events,” e.g., “event frame,” “event insertion logic,” “event recovery logic,” the term “event” is not limited to an event driven system and is intended to encompass side information, generally, including side information about the regular data that is transferred from a transmitter to a receiver.

5 FIG. 1A is a block diagram of a transmitter 120 and a receiver 140 with a serial communication link 130 according to an embodiment of the present disclosure. In one embodiment, the transmitter 120 and receiver 140 may be a master and a slave configured for synchronous communication over, for example, a serial peripheral interface. The transmitter 120 may include a processor 122, event insertion logic 124, and serial
10 interface 126. The processor 122 may be configured to send regular data to the receiver 140 over the communication link 130. The event insertion logic 124 may be configured to provide event information to the receiver 140 using the communication link 130. The event information may be related to events that are created at the transmitter 120 or, in another embodiment, event information provided to the
15 transmitter 120 about events external to the transmitter 120. By way of non-limiting example, event information may include timing information, event type information, status information, etc. In various embodiments, the event insertion logic 124 may be configured to insert the event into a serial communication stream encoded at the serial interface 126 and transmitted on the communication link 130. The serial interface 126 and serial
20 interface 146 may be configured to translate data into frames for transfer over the communication link 130, as well as recover data from transmitted frames. Some routine elements related to synchronous communication are not shown to simplify FIG. 1A, such as the clock (Ck) line.

On the receiver 140 side, the receiver 140 may include event recovery logic 142, a
25 processor 144, and a serial interface 146. The event recovery logic 142 may be configured to recover event information according to the various embodiments described in this disclosure.

While the embodiments described with reference to FIG. 1 relate to synchronous communication, one of ordinary skill in the art would understand that the principles are
30 applicable to asynchronous communication.

A general description of processes for transmission of event frames follows, with reference to FIGS. 1B to 1E.

FIG. 1B shows a flowchart of an event transmission with uniform delay based on a transferred delay value according to an embodiment of the disclosure. In operation 150, an event is received. The event may be related to an event generated at a transmitter or be received by the transmitter for an external event. In operation 152, an event delay is
5 determined, the event delay being defined as the time between a predefined bit position of a present frame being transmitted when the event occurs and the event. In operation 154, the event frame corresponding to the event is generated and the event delay is inserted into an event delay field of the event frame. By way of non-limiting example, the delay value may be a clock count or a value from which a time or clock count is recoverable. In one
10 embodiment, the delay value may indicate where in the present frame (*i.e.*, the ongoing frame) the event occurred compared to a predefined point in the present frame (*e.g.*, the start of the present frame, the end of the frame, etc.). If there was no ongoing transmission or the event frame was otherwise not delayed then the delay value may be indicative of no delay or “0.” In operation 156, the event frame having the delay value is sent over a serial
15 communication link. In one embodiment, if there was an ongoing frame then the event frame may be sent back-to-back with the ongoing frame.

FIG. 1C shows a flowchart of an event transmission with uniform delay based on transferred delay value according to an embodiment of the disclosure. In operation 160, an event frame including an event delay is received over a serial communication link. In
20 operation 162, the event frame is decoded to recover the event delay and an event indicator. In operation 164, a number of clock cycles corresponding to the event delay are waited. In operation 166, a receiver-side event is asserted responsive to the event indicator after waiting the number of clock cycles.

FIG. 1D shows a flowchart of an event transmission with uniform delay according
25 to an embodiment of the disclosure. In operation 170, an event is received. The event may be received related to an event generated at a transmitter or be received by the transmitter for external event. In operation 172, the event is delayed according to a uniform delay. In one embodiment, the uniform delay may be the length of a frame. In operation 174, an event frame is generated that corresponds to the event. In one embodiment, the event
30 frame may include an event indicator indicative of which event of a set of events the event frame corresponds to, and an event frame indicator indicating that that the event frame is, in fact, an event frame. In operation 176, the delayed event frame is transmitted on a serial communication link.

FIG. 1E shows a flowchart of an event transmission with uniform delay according to an embodiment of the disclosure. In operation 180, an event frame is received over a serial communication link. In operation 182, the event frame is decoded to recover an event indicator. In operation 184, a receiver-side event is immediately asserted responsive to the recovered event indicator.

One of ordinary skill in the art would understand that an event frame may comprise one or more fields. For example, a frame may have fields that comprise one or more bits, the one or more bits configurable to be indicative of the various information described in connections with the various embodiments of the present disclosure. One of ordinary skill in the art will recognize many permutations for the fields and the bits that comprise the fields.

FIG. 2A is a detailed timing diagram illustrating transmission of certain events over a serial communication link with a delay count included in the event transmission (e.g., a transferred count value), according to an embodiment of the disclosure. For the discussion of FIG. 2A, FIGS. 2B – 2D show marked sections of FIG. 2A in an expanded view in an effort to make the details of FIG. 2A easier to view. In this embodiment, when an event occurs, the event insertion logic 124 generates an event frame and the event frame is sent back-to-back (*i.e.*, immediately after) with a present frame for which transmission is ongoing. The event frame contains a delay field that indicates where (or when) in the present frame (*i.e.*, the ongoing frame) the event occurred compared to a predefined point in the present frame (*e.g.*, the start of the present frame). Thus, if there is no ongoing frame transmission, the event frame is sent immediately with the delay value 0. The event frame has the highest priority, so it will be sent before other pending data frames.

FIG. 2A shows transmission of events with uniform delay based on transferred counter value for three different locations of the event relative to the present frame being transmitted. FIG. 2A shows as an example of a Universal Synchronous Receiver and Transmitter (USRT) where data is generated on the positive edge of the clock and the frame on the TxD line consists of a stop bit, 9 data bits, no stop bits, and the signal line being high represents an IDLE state. The clock and TxD signals are shown as the top two waveforms.

FIG. 2A shows three event transmission examples - signal group 220, signal group 240 and signal group 260 - where events occur at delay time 9, 4, and 0 relative to the present frame shown on the TxD signal. The reference for time (DLY=0) shown as signal

group 260 corresponds to the clock cycle prior to the start bit. By way of non-limiting example, the event may be a physical input pin or a software-generated event. In the case of a physical pin, the physical pins may be configured to generate an event on a rising edge, a falling edge, or a toggling signal value.

5 In an embodiment where the system supports multiple events, event insertion logic may be configured to encode an event number in the event frame (shown, for example, as EV0 and EV1 in FIG. 2A) together with a delay value (shown, for example, as DL0, DL1, DL2 and DL3 in FIG. 2A) in the event frame. In various embodiments, event numbers may be associated (at the transmitter and/or receiver side) with event sources,
10 event sub-modules, types of events, predefined information associated with the foregoing, and more. With multiple events, several events may occur within (e.g., during) the same present frame. Depending on the application, this may be solved by event insertion logic 124 configured to: (1) in one embodiment, prioritizing one event and discarding the other event(s), (2) in another embodiment, prioritizing one event and sending the remaining
15 event frame(s) back-to-back but with an ERROR bit (not shown in FIG. 2A) to indicate incorrect timing, or (3) in yet another embodiment, sending as two events, but reserving one of the delay values for error signature.

 On the receiver side, the event recovery logic of the receiver may be configured to decode an event frame to find the event delay value (e.g., in clocks defined by rising and
20 falling edges of Ck). The receiver then counts a number of clocks based on the event delay value from a predefined point in the received event frame and asserts the correct event line and the end of the delay. In the example shown in FIG. 2A, the event recovery logic counts from the last bit of the event frame. As shown, the events from the transmitter side are then recovered on the receiver side with a fixed latency of 21 clocks. In various
25 embodiments, the fixed latency may be implemented with registers and the size of the fixed latency may depend, at least in part, on the number of registers in the data path, from where the counters start counting on the transmitter and receiver side, etc.

 For signal group 220, a delay 222 between the start of present frame 212 and the occurrence of event 224 at transmitter (EV_IN) is 9 clocks. The event 224 gets transmitted
30 out as an event frame 214 when the present frame 212 completes. At the end of the event frame 214, the receiver begins counting the number of clocks encoded in the event frame 214 as DL0-DL3 (9 clocks in this example) to create a delay 232. In one embodiment, the receiver may also use the event numbers EVO and EV1 to determine the

source of the event 224 for this event frame 214. When the count terminates, the receiver asserts a receive side event 234 (EV_OUT), which is a uniform latency of 21 clocks relative to when the event 224 originally occurred at the transmitter.

For signal group 240, a delay 242 between the start of the present frame 212 and the occurrence of event 244 at the transmitter (EV_IN) is 4 clocks. The event 244 gets transmitted out as an event frame 214 when the present frame 212 completes. At the end of the event frame 214, the receiver begins counting the number of clocks encoded in the event frame 214 on DL0-DL3 (4 in this case) to create a delay 252. The receiver may also use the event numbers EVO and EV1 to determine the source of the event 244 for the event frame 214. When the count terminates, the receiver asserts a receive side event 254 at the receiver (EV_OUT), which is a uniform latency of 21 clocks relative to when the event 244 originally occurred at the transmitter.

For signal group 260, a delay 262 between the start of the present frame 212 and the occurrence of event 264 at the transmitter (EV_IN) is 0 clocks. The event 264 gets transmitted out as an event frame 214 when the present frame 212 completes. At the end of the event frame 214, the receiver begins counting the number of clocks encoded in the event frame 214 on DL0-DL3 (0 in this case) to create a delay 272. The receiver may also use event numbers EVO and EV1 to determine the source of the event 264 for this event frame 214. When the count terminates, the receiver asserts a receive side event 274 at the receiver (EV_OUT), which is a uniform latency of 21 clocks relative to when the event 264 originally occurred at the transmitter.

One of ordinary skill in the art would understand that the delay bits and event number bits may be positioned differently relative to each other than described with reference to FIG. 2A. Moreover, other embodiments may use a different number of bits or different encodings for defining the event delay than described with reference to FIG. 2A. Also, other embodiments may use a different number of bits (including none) or different encodings for defining the source of the event than described with reference to FIG. 2A. As shown with bits in the event frame after the EV bits, the rest of the event frame (which may be positioned at various locations within the frame) includes a set of unique data bits that identify this frame as an event frame.

Thus, while FIG. 2A shows a specific serial communication link protocol according to an embodiment of the disclosure, other embodiments may include other protocols

including various data sizes and various control bits, and a packet may include multiple physical frames, not only a single frame as shown in FIG. 2A.

FIG. 3A shows a detailed timing diagram illustrating transmission of certain events over a serial communication link with a predetermined delay for the event frame transmission. In this embodiment, when an event occurs, the event may be stored at the transmitter for a time corresponding to the length of a frame/packet. For the discussion of FIG. 3A, FIGS. 3B and 3C show marked sections of FIG. 3A in an expanded view in an effort to make the details of FIG. 3A easier to view. As non-limiting examples, the event storage may be accomplished by putting the event into a shift register of this size, or by storing the event in a register bit while a counter counts down to zero. When the potentially ongoing frame is transmitted, the event frame has the highest priority such that no new frame is started until the delay times out and the event frame is generated. This delay ensures a uniform latency from the event occurrence at the transmitter until the event frame is received at the receiver.

FIG. 3A shows, as an example, a Universal Synchronous Receiver and Transmitter (USRT) where data is generated on the positive edge of the clock and the frame on the TxD line includes a stop bit, 9 data bits, and no stop bits, and the signal line being high represents an IDLE state.

FIG. 3A shows examples where events occur at time 0 (signal group 320) and at time 4 (signal group 360). The reference for time (DLY=0) corresponds to the clock cycle prior to the start bit. By way of non-limiting example, an event may be a physical input pin or a software-generated event. In the case of a physical input pin, the physical pins may be configured to generate the event on a rising edge, a falling edge, or a toggling signal value.

FIG. 3A also shows that for a synchronous communication protocol, the transmitter and receiver may operate at a different (higher) frequency than the communication link, and an event may need to be synchronized to the communication module. In an asynchronous communication protocol, a delay may be fixed from an event until an event frame is generated at a resolution of a system clock.

As with the embodiment discussed with reference to FIG. 2A, if a system using an embodiment of event transmission shown in FIGS. 3A (or 3D) supports multiple events, an event number may be encoded in an event frame. With multiple events, the multiple events may occur within the same present frame. Depending on the application this may be solved by insertion logic configured to: (1) in one embodiment, prioritize one event frame

and discard the other event frames, or (2) in another embodiment, prioritize one event frame and send the remaining event frames back-to-back but with an ERROR bit (shown as 'ERR' in FIG. 3A) to indicate that the latency may not be uniform.

On the receiver side, the receiver asserts its event output immediately (or after a
5 fixed delay) when an event frame is received.

In the example in FIG. 3A, as shown, an event from the transmitter side is then regenerated on the receiver side with a uniform latency of 23 clocks relative to when the event actually occurred. The size of the uniform latency depends on number of registers in the data path, from where the counters start counting on the transmitter and receiver side,
10 etc.

For signal group 320, the delay is shown as 0 clocks. Event 322 is delayed for an event delay frame 324 (e.g., 10 cycles) and is then transmitted out as an event frame 326 from the transmitter. The delay in the transmitter ensures that any ongoing frame being transmitted when the event 322 occurs is completed before (or at the same time as) the end
15 of the delay. At the end of the event frame 326, the receiver asserts a receive side event 328 (EV_OUT), which is a uniform latency of 23 clocks relative to when the event 322 originally occurred at the transmitter.

FIG. 3D shows another detailed timing diagram illustrating transmission of certain events over a serial communication link with a predetermined delay for the event frame
20 transmission. For the discussion of FIG. 3D, FIGS. 3E and 3F show marked sections of FIG. 3D in an expanded view in an effort to make the details of FIG. 3D easier to view. For signal group 360, the delay is shown as 4 clocks. Event 362 is delayed for an event delay frame 364 (e.g., 10 cycles) and is then transmitted out as an event frame 366 from the transmitter. The delay in the transmitter ensures that any ongoing frame being transmitted
25 when the event 362 occurs is completed before (or at the same time as) the end of the delay. Note that in this example, an idle period 368 occurs on the serial communication link between the frame being transmitted when the event 362 occurred and the event frame 366. At the end of the event frame 366, the receiver asserts a receive side event 368 (EV_OUT), which is a uniform latency of 23 clocks relative to when the event 362
30 originally occurred at the transmitter.

As discussed above with reference to FIG. 2A, the event frame may include bits, shown in event frame 366 as EV0-EV2, to indicate the source of the event 362 and a set of unique data bits that identify this as an event frame.

While FIGS. 3A and 3D illustrate a specific serial communication link protocol, other embodiments may include other protocols including various data sizes, numbers and types of control bits, different encoding schemes, and the packet may include multiple physical frames, not only a single frame as shown in FIGS. 3A and 3D.

5 In its various embodiments, the present disclosure enables communication of timing for events on serial communication links with no need for additional lines in addition to those required by the communication system. The event is perceived by a slave as a uniform latency from the occurrence at the transmitter side regardless of where it happens in the communication package.

10 A description of prioritization of event transmission follows with reference to FIGS. 4A-4C, FIG. 5 and FIGS. 6A-6C, according to embodiments of the disclosure. In one embodiment, prioritization logic may be part of event insertion logic, such as event insertion logic 124 (FIG. 1A).

FIGS. 4A-4C show frame level timing diagrams illustrating different priority event
15 timings and some errors that may occur over a serial communication link – errors addressed by embodiments of the present disclosure. The solid line arrows represent higher priority events and the dashed line arrows represent lower priority events. Similarly, the boxes with solid lines illustrate the communication frames containing information about the higher priority event and the boxes with dashed lines illustrate the
20 communication frames containing information about the lower priority event.

 In FIG. 4A, a lower priority event 410 occurs first. However, since communication frame 414 has a certain duration, sending the lower priority event 410 immediately when it occurs results in loss of a higher priority event 412 (at least it cannot have correct timing) even though it has higher priority, because the higher priority event 412 has to wait until a
25 lower priority event frame 416 completes. Thus, although the lower priority event frame 416 may be sent with correct timing, a higher priority event frame 418 is sent with an error indication to indicate that there may be an inconsistency between the latency of when the higher priority event 412 occurred and when the higher priority event frame 418 is received.

30 In FIGS. 4B and 4C, the higher priority event and the lower priority event occur very close in time such that jitter in sampling of the event may produce a random order for when the event frames are transmitted. In FIG. 4B, a higher priority event 422 won so a higher priority event frame 426 is sent out first and at a proper time. Thus, a lower priority

event 420 waits for the next frame and is sent with an error indication to indicate that there may be an inconsistency between the latency of when the lower priority event 420 occurred and when a lower priority event frame 428 is received. In FIG. 4C, a lower priority event 430 won so a lower priority event frame 436 is sent out first and at a proper time.

- 5 Thus, a higher priority event 432 waits for the next frame and is sent with an error indication to indicate that there may be an inconsistency between the latency of when the higher priority event 432 occurred and when a higher priority event frame 438 is received.

One method to correct for these inconsistencies in prioritizing events is to use transmit hardware (which may include software implementations) that creates a uniform
10 latency by inserting a delay from the event equal to the frame length before sending the event frame, such as, for example, by using the embodiment illustrated in FIG. 3. The delay time may be used to force a correct priority.

In FIG. 3 the events are all delayed for one frame. When the delay duration times out, the transmitter checks if there are pending events in the pipeline of higher priority. If
15 there are, the transmitter prioritizes the higher priority events by not starting any new frames until the higher priority event is ready to be transmitted, sends the high priority event, and then sends any events with lower priority afterward, and in the proper priority order – with an error flag set.

FIG. 5 shows a logic diagram illustrating priority logic 510 as an example for
20 prioritizing events on a serial communication link according to embodiments of the disclosure. FIG. 5 illustrates one possible implementation of this behavior; many other circuits and logical implementation may be used to accomplish the prioritization. The delay of events is illustrated in FIG. 5 as a delay circuit 520, here, shift registers. Event submodules (EV0, EV1,... EVn) are delineated in FIG. 5 by horizontal dashed lines. As
25 stated above, counters may also be used for a delay. In this example, EV0 has the highest priority, while increasing event numbers have decreasing priority. The output from a NOR-gate for each event submodule indicates that there is no pending event at that event submodule. A vertical AND-line 512 qualifies each event by indicating at each submodule that there are no pending events with higher priority. If an event has reached the last stage
30 of delay circuit 520, then the qualified event output is asserted if there are no pending events with higher priority. If there are pending events with higher priority, the delay circuit 520 for that event submodule is stopped until all events with higher priority have been transmitted. Thus, regardless of the arrival time of events EV0 through EVn, within a

given data frame (or sequence of event frames) the highest priority event will be sent first followed by any pending lower priority events in the proper priority order.

FIGS. 6A-6C are frame level timing diagrams illustrating event timings for prioritized events over a serial communication link. The solid lines represent higher priority events and the dashed lines represent lower priority events. Similarly, the boxes with solid lines illustrate the communication frames containing information about the higher priority event and the boxes with dashed lines illustrate the communication frames containing information about the lower priority event.

The timing in FIGS. 6A-6C show results that would be achieved for a higher priority event relative to a lower priority event using the uniform delay prioritization discussed with reference to FIG. 5. In FIGS. 6A-6C EXT indicates when the event actual occurs as input to the priority logic and INT indicates when the event would be available for transmission (i.e., at the end of the shift register creating a uniform one frame delay).

FIG. 6A illustrates a situation where a higher priority event 602 (solid lines) occurs after a lower priority event 606 (dashed lines) – but too close in time to be sent in different frames. A delayed higher priority event 604 (INT) wins in the priority logic and will be sent out as a higher priority event frame 610 when the uniform delay is over and will thus be received with a uniform latency relative to when the higher priority event 602 occurred at the transmitter. The lower priority event 606 lost in the priority logic and will thus be sent as a lower priority event frame 612 after the higher priority event frame 610 with an error indication to indicate that there may be an inconsistency between the latency of when the lower priority event 606 occurred and when the lower priority event frame 612 is received.

FIG. 6B illustrates a situation where the two events (higher priority event 622 and lower priority event 626) happen simultaneously or almost simultaneously. Timing jitter in the sampling process determines which event is first registered. However, when the first occurring event has been delayed for one frame, the priority logic checks for pending higher priority events. This way, the higher priority event 622 always wins.

As shown in FIG. 6B, Option A indicates that at the time of sampling the events, the higher priority event 622 was sampled first and a higher priority event frame 630 for the higher priority event 622 was sent out after the uniform delay and will thus be received with a uniform latency relative to when the higher priority event 622 occurred at the transmitter. The lower priority event 626 lost in the priority logic and will thus be sent as a

lower priority event frame 632 after the higher priority event frame 630 with an error indication to indicate that there may be an inconsistency between the latency of when the lower priority event 626 occurred and when the lower priority event frame 632 is received.

Also as shown in FIG. 6B, Option B indicates that at the time of sampling the events, the lower priority event 626 was sampled first. However, because the higher priority event 622 wins in the priority logic, a higher priority event frame 634 for the higher priority event 622 was sent out after the uniform delay and will thus be received with a uniform latency relative to when the higher priority event 622 occurred at the transmitter. The lower priority event lost 626 in the priority logic and will thus be sent as a lower priority event frame 636 after the higher priority event frame 634 with an error indication to indicate that there may be an inconsistency between the latency of when the lower priority event 626 occurred and when the lower priority event frame 636 is received.

FIG. 6C illustrates yet another issue. If a higher priority event 642 occurs one frame later relative to a lower priority event 646, we may have a situation where jitter in the sampling time will determine transmission order. In Option A, the lower priority event 646 is sampled first and its event frame 650 may be transmitted without latency error and a higher priority event frame 652 for the higher priority event 642 is transmitted next without latency error. In Option B, the higher priority event 642 is sampled first and its event frame 654 may be transmitted without latency error. However, a lower priority event frame 656 will be sent after the higher priority event frame 654 with an error indication to indicate that there may be an inconsistency between the latency of when the lower priority event 646 occurred and when the lower priority event frame 656 is received. In either option, the higher priority event frame 652 or 654 is transmitted without a latency error.

In many cases the two (or more) events happen randomly and this behavior is acceptable; the higher priority event frame is always transmitted at the right time. However, in some systems where a fixed relation exists between the two events it may be unacceptable that the lower priority event frame switches between being before or after the higher priority event frame. For such cases, an option may be included to turn off the prioritization that looks for other events in the pipe and just prioritize among those that are ready to be transmitted.

FIG. 7 shows a flowchart of a process for prioritizing events according to an embodiment of the disclosure. In operation 700, two or more events are delayed for a delay time. In one embodiment, the delay time may correspond to a frame time. In

operation 702, additional frames are prevented from being started on a serial communication link while any of the two or more events is being delayed. In operation 704, a first event frame is transmitted that corresponds to one of the two or more events. In one embodiment, the first event has the highest priority after its corresponding delay time. In operation 706, event occurrences that have a priority lower than the first event frame are held until after its corresponding delay time and no higher events are pending. In operation 708, the event frame(s) that correspond(s) to a highest priority event is repeatedly transmitted until event frames have been transmitted for all of the two or more events.

10 FIG. 8 is a block diagram of a touch panel system including a system controller, a touch controller, and a display panel with serial communication links according to an embodiment of the present disclosure.

 In this system, a serial bus is used to distribute Vertical Synchronization (VS) and Horizontal Synchronization (HS) event information from the display controller 816 to all of the touch acquisition sub-systems via a single control line 817, which is also used for data/control transfers. As an example, the control line 817 might be the Master Tx/D of a USART channel, which is used to send configuration data to the source Driver ICs 834, and get Analog-to-Digital Control (ADC) samples representing touch data in return on the Rx/D line. The system Printed Circuit Board (PCB) 810 may be, for instance, a mobile phone, tablet, or any other system with a display, which supports touch sensing. As an example, the system PCB 810 may be connected to the TFT LCD panel 830 using a flexible printed circuit board 826, and the source driver ICs 834 may be mounted on the glass using silver epoxy. For some touch solutions, touch acquisition front-end 838 may be split and implemented on the display source driver ICs 834. The measurements may then be transferred back to the touch controller 818 where the Central Processing Unit (CPU) 812 (and possible Digital Signal Processing (DSP) unit) performs a post-processing operation to filter noise and determine, for example, whether someone touches the screen with one or more fingers, or if some other touch event occurred.

 The display controller 816, display source driver 836, and gate driver circuitry 832 in this embodiment may be totally unaware of the touch system. The display controller 816 controls the screen updating via control line 817. However, for the touch system it may be important to accurately synchronize its acquisition 845 to the display update 817 to avoid the noise from the source driver 834 ICs and gate driver 832 ICs. The touch controller 818

IC receives the HS/VS signals (i.e., events) from the display controller 816, and the event insertion logic 820 prioritizes these events. In one embodiment, the event insertion logic 820 may implement embodiments of a delay circuit and priority logic, such as a delay circuit 520 and priority logic 510 (FIG. 5). In one embodiment, the event insertion

5 logic 820 may implement one or more of the processes for uniform delay described with references to FIGS. 1A-3. Embodiments of touch controller 818 IC may then translate the events into “frames” or “packages” before inserting these frames into a serial stream. In various embodiments, serializer 824 (Tx) will send event frames 821 before data frames 823 (i.e., data frames have lowest priority). The touch acquisition front-end 838 in

10 the source driver ICs 834 will de-serialize (deserializer 840) the serial stream and recover/decode the HS/VS events (event recovery 842) before passing them to the timing and control acquisition 844 stage.

It should be noted that FIG. 8 is discussed as one example of system in accordance with embodiments of the present disclosure. One of ordinary skill in the art will appreciate

15 that there are many other systems where there is a need to transmit timing details or other event details as additional “side information” relative to the regular data transmitted on serial communication links, and such systems may use embodiments of the present disclosure.

As a non-limiting example of event prioritization with touch displays, VSYNC is

20 sent for each new image update, while there are several HSYNCs between each VSYNC, representing new lines within the same image. The prioritizing logic ensures that even if VSYNC and HSYNC appear simultaneously (which is the case in some systems), VSYNC should be assigned the higher priority and will win. However, if an HSYNC appears one USART frame before the VSYNC, jitter in the sampling time will determine whether both

25 are transmitted without errors (if the lower priority HSYNC is detected first) or the higher-priority VSYNC is transferred first, with an error identification on the later occurring event. In either option A or option B, the higher priority VSYNC is always transmitted at the right time, but it might be confusing if an HSYNC belonging to the previous image comes after the VSYNC (and hence a new image) – even if it has an error identification.

30 The timing between HSYNC and VSYNC is application specific – in one particular display it will always behave the same way, and the same behavior should always be expected in the system. Hence, there may be applications where the prioritizing (at least for these two

events) should be turned off – in the sense that the lower-priority event (HSYNC) is discarded if it arrives while the higher-priority VSYNC is being transmitted.

Many of the functional units described in this specification may be described as modules, threads, or other segregations of programming code, in order to more particularly emphasize their implementation independence. Modules may be at least partially implemented in hardware, in one form or another. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable state machines, programmable logic devices, or the like.

Modules may also be implemented using software, stored on a physical storage device (e.g., a computer-readable storage medium), in memory, or a combination thereof for execution by various types of processors.

An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as a thread, object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several storage or memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network. Where a module or portions of a module are implemented in software, the software portions are stored on one or more physical devices, which are referred to herein as computer-readable media.

In some embodiments, the software portions are stored in a non-transitory state such that the software portions, or representations thereof, persist in the same physical location for a period of time. Additionally, in some embodiments, the software portions

are stored on one or more non-transitory storage devices, which include hardware elements capable of storing non-transitory states and/or signals representative of the software portions, even though other portions of the non-transitory storage devices may be capable of altering and/or transmitting the signals. One example of a non-transitory storage device includes a read-only memory (ROM), which may store signals and/or states representative of the software portions for a period of time. However, the ability to store the signals and/or states is not diminished by further functionality of transmitting signals that are the same as or representative of the stored signals and/or states. For example, a processor may access the ROM to obtain signals that are representative of the stored signals and/or states in order to execute the corresponding software instructions.

While the present disclosure has been described herein with respect to certain illustrated embodiments, those of ordinary skill in the art will recognize and appreciate that the present invention is not so limited. Rather, many additions, deletions, and modifications to the illustrated and described embodiments may be made without departing from the scope of the invention as hereinafter claimed along with their legal equivalents. In addition, features from one embodiment may be combined with features of another embodiment while still being encompassed within the scope of the invention as contemplated by the inventors.

Additional non-limiting embodiments of the disclosure include:

Embodiment 1: A touch panel system, comprising: a display system; a touch acquisition front-end; and a touch controller operatively coupled to the touch acquisition front-end by a serial communication link, wherein the touch controller includes a communication interface comprising: delay circuitry configured to determine an event delay between a predefined bit position of an ongoing frame being transmitted and an event; and transmission circuitry configured to send an event frame after the ongoing frame, wherein the transmission circuitry is configured to include in the event frame, delay bits indicative of the event delay and event identifier bits indicative of the event the event frame corresponds to.

Embodiment 2: The touch panel system according to Embodiment 1, wherein the delay circuitry comprises a delay counter configured to determine the event delay by counting a number of clocks from the event to the predefined bit position of the ongoing frame.

Embodiment 3: The touch panel system according to Embodiments 1 and 2, wherein the communication interface further comprises two or more event sub-modules, wherein a first of the two or more event sub-modules corresponds to a first event type and a second of two or more event sub-modules corresponds to a second event type, priority logic configured to associate a different priority with each of the first event type and the second event type.

Embodiment 4: The touch panel system according to Embodiments 1 through 3, wherein the communication interface further comprises priority logic, and wherein responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type the priority logic is configured to: transmit one of the first event and the second event with a higher priority; and discard one of the first event and the second event with a lower priority, wherein the higher priority corresponds to a vertical synchronization event and the lower priority corresponds to a horizontal synchronization event.

Embodiment 5: The touch panel system according to Embodiments 1 through 4, wherein the communication interface further comprises priority logic, and wherein responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type the priority logic is configured to: transmit a higher priority event frame corresponding to one of the first event and the second event with a higher priority; hold one of the first event and the second event with a lower priority while the higher priority event frame is transmitted; and transmit a lower priority event frame corresponding to one of the first event and the second event with the lower priority after the higher priority event frame is transmitted, the lower priority event frame including an error bit, wherein the higher priority corresponds to a vertical synchronization event and the lower priority corresponds to a horizontal synchronization event.

Embodiment 6: The touch panel system according to any one of Embodiments 1 through 5, wherein the communication interface is configured for communication according to a protocol selected from the group consisting of Universal Asynchronous Receiver/Transmitter, Universal Synchronous Receiver/Transmitter, and Universal Synchronous/Asynchronous Receiver/Transmitter.

Embodiment 7: The touch panel system according to any one of Embodiments 1 through 6, wherein the communication interface is configured to discard an event of the two or more events.

Embodiment 8: The touch panel system according to any one of Embodiments 1 through 7, wherein the communication interface is configured to include event indicator bits in the event frame, wherein the event indicator bits are indicative of which event of a set of events the event frame corresponds to.

5 Embodiment 9: A touch panel system, comprising: a display system; a touch controller; and a touch acquisition front-end operatively coupled to the touch acquisition front-end by a serial communication link, wherein the touch acquisition front-end includes a communication interface comprising: receive circuitry configured for receiving an event frame, wherein the event frame includes delay bits indicative of a delay between an event
10 that occurred at a transmitter relative to a predefined bit position in a previous frame received from the transmitter; and delay circuitry configured to: decode the delay bits; count a number of clock cycles corresponding to the delay bits; and assert a receiver-side event after the counting.

Embodiment 10: The touch panel system according to Embodiment 9, wherein the
15 delay circuitry is configured to begin counting the number of clock cycles from a predefined point in the event frame.

Embodiment 11: The touch panel system according to any one of Embodiments 9 or 10, wherein the predefined point in the event frame is the last bit of the event frame.

Embodiment 12: The touch panel system according to any one of Embodiments 9
20 through 11, wherein the receiver-side event corresponds to the event that occurred at the transmitter.

Embodiment 13: The touch panel system according to any one of Embodiments 9 through 12, wherein the delay bits correspond to no delay.

Embodiment 14: The touch panel system according to any one of Embodiments 9
25 through 13, wherein the delay circuitry is configured to decode event indicator bits and assert the receiver-side event responsive to the event indicator bits, wherein the event indicator bits are indicative of which event of a set of events the event frame corresponds to.

Embodiment 15: The touch panel system according to any one of Embodiments 9
30 through 14, wherein the delay circuitry is configured to decode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

Embodiment 16: The touch panel system according to any one of Embodiments 9 through 15, wherein the touch acquisition front-end includes a noise cancellation circuit configured to filter noise responsive to event frames received over the serial communication link.

- 5 Embodiment 17: The touch panel system according to any one of Embodiments 9 through 16, wherein the noise cancellation circuit is configured to filter noise responsive to, at least in part, timing information related to events corresponding to received event frames and types of events corresponding to the received event frames.

- Embodiment 18: The touch panel system according to any one of embodiments 9
10 to 17, wherein the types of events comprise initiation of display update signals at the display system.

 Embodiment 19: The touch panel system according to any one of embodiments 9 to 18, wherein the display update signals comprise horizontal synchronization and vertical synchronization signals.

- 15 Embodiment 20: A serial communication link, comprising: a transmitter and a receiver configured to define a uniform latency between an event occurrence at the transmitter and an event signal being asserted at the receiver; the transmitter comprising: delay circuitry configured for determining an event delay between a predefined bit position of an ongoing frame being transmitted and the event occurrence; and transmission circuitry
20 configured to send an event frame after the ongoing frame, wherein the transmission circuitry is configured to include in the event frame, delay bits corresponding to the event delay and event identifier bits indicative of the frame being transmitted is an event frame; and the receiver comprising: receive circuitry configured to decode the frame being transmitted as the event frame; and delay circuitry configured to: decode the delay bits;
25 count a number of clock cycles corresponding to the delay bits; and assert the event signal after the counting.

- Embodiment 21: An event frame for serial communication, comprising: a number of event fields, each field of the number of event fields having one or more configurable bits, the number of event fields comprising: an event identifier field configurable to
30 indicate an event of a set of transmitter-side events; and a delay field configurable to indicate an event delay between a previous frame and the event, and a start bit and an end bit.

Embodiment 22: The event frame for serial communication according to Embodiment 21, wherein the number of event fields comprises an error field configurable to indicate incorrect timing associated with the event frame.

Embodiment 23: The event frame for serial communication according to any one
5 of Embodiments 21 or 22, wherein the delay field is configurable to indicate the event delay as a delay time between a predefined bit position of the previous frame and an occurrence of the event.

Embodiment 24: The event frame for serial communication according to any one
10 of Embodiments 21 through 23, wherein the predefined bit position of the previous frame is the start of the previous frame.

Embodiment 25: A touch panel system, comprising: a display system; a touch acquisition front-end; and a touch controller operatively coupled to the touch acquisition front-end by a serial communication link, wherein the serial communication link comprises a transmitter and a receiver configured to define a uniform latency between a transmitter-side event and a receiver-side event, the transmitter comprising: delay circuitry configured to receive the transmitter-side event and delay the transmitter-side event by a delay time corresponding to a frame time; and control circuitry configured to prevent additional frames from being started while the transmitter-side event is being delayed; and transmission circuitry configured to send an event frame corresponding to the transmitter-side event after the delay time, wherein the transmission circuitry is configured to include
15 in the event frame, event identifier bits indicating the frame being transmitted is an event frame, and the receiver comprising: receive circuitry configured to: decode the frame being transmitted as the event frame; and assert the receiver-side event responsive to the decoding.

Embodiment 26: The touch panel system according to Embodiment 25, wherein
25 the delay circuitry is configured to decode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

Embodiment 27: The touch panel system according to any one of Embodiments 25
30 or 26, wherein the touch acquisition front-end includes a noise cancellation circuit configured to filter noise responsive to event frames received over the communication link.

Embodiment 28: The touch panel system according to Embodiment 27, wherein the noise cancellation circuit is configured to filter noise responsive to, at least in part, timing information related to events corresponding to received event frames and types of events corresponding to the received event frames.

5 Embodiment 29: The touch panel system according to any one of Embodiments 25 through 28, wherein the types of events comprise initiation of display update signals at the display system.

 Embodiment 30: The touch panel system according to Embodiment 29, wherein the display update signals comprise horizontal synchronization and vertical
10 synchronization signals.

 Embodiment 31: A serial communication link transmitter, comprising: delay circuitry configured to determine an event delay between a predefined bit position of an ongoing frame being transmitted and an event; and transmission circuitry configured to send an event frame after the ongoing frame, wherein the transmission circuitry is
15 configured to include in the event frame, delay bits corresponding to the event delay and event identifier bits indicative of the event to which the event frame corresponds.

 Embodiment 32: The serial communication link transmitter according to Embodiment 31, wherein the delay circuitry comprises a delay counter configured to determine the event delay by counting a number of clocks from the event to the predefined
20 bit position of the ongoing frame.

 Embodiment 33: The serial communication link transmitter according to any one of Embodiments 31 or 32, further comprising: two or more event sub-modules, wherein a first of the two or more event sub-modules corresponds to a first event type and a second of the two or more event sub-modules corresponds to a second event type, and priority logic
25 configured to associate a different priority with each of the first event type and the second event type.

 Embodiment 34: The serial communication link transmitter according to any one of Embodiments 31 through 33, further comprising priority logic, and wherein responsive to an occurrence of a first event corresponding to a first event type and a second event
30 corresponding to a second event type, the priority logic is configured to: transmit one of the first event and the second event with a higher priority; and discard one of the first event and the second event with a lower priority.

Embodiment 35: The serial communication link transmitter according to any one of Embodiments 31 through 34, further comprising priority logic, and wherein responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type the priority logic is configured to: transmit a higher
5 priority event frame corresponding to one of the first event and the second event with a higher priority; hold one of the first event and the second event with a lower priority while the higher priority event frame is transmitted; and transmit a lower priority event frame corresponding to one of the first event and the second event with the lower priority after the higher priority event frame is transmitted, the lower priority event frame including an error
10 bit.

Embodiment 36: The serial communication link transmitter according to any one of Embodiments 31 through 35, wherein the transmission circuitry is configured to send the event frame back-to-back with the ongoing frame.

Embodiment 37: The serial communication link transmitter according to any one of
15 Embodiments 31 through 36, wherein the transmission circuitry is configured to encode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

Embodiment 38: The serial communication link transmitter according to any one of
20 Embodiments 31 through 37, wherein the transmission circuitry is further configured to include event identifier bits in the event frame, wherein the event indicator bits indicate which event of a set of events the event frame corresponds to.

Embodiment 39: A method of transmitting events over a serial communication link, comprising: determining an event delay between a predefined bit position of an
25 ongoing frame being transmitted and an event; encoding an event frame corresponding to the event, wherein the event frame includes: delay bits corresponding to the event delay; and event identifier bits that indicate the event frame corresponds to the event, and transmitting the event frame after the ongoing frame.

Embodiment 40: The method according to Embodiment 39, further comprising
30 sending the event frame back-to-back with the ongoing frame.

Embodiment 41: The method according to any one of Embodiments 39 or 40, further comprising determining the event delay by counting a number of clocks from the event to the predefined bit position of the ongoing frame.

Embodiment 42: The method according to any one of Embodiments 39 through 41, further comprising: responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type: transmitting one of the first event and the second event with a higher priority; and discarding of the first event and
5 the second event with a lower priority.

Embodiment 43: The method according to any one of Embodiments 39 through 42, further comprising: responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type: transmitting a higher priority event frame corresponding to one of the first event and the second event with a
10 higher priority; holding one of the first event and the second event with a lower priority while the higher priority event frame is transmitted; and transmitting a lower priority event frame corresponding to one of the first event and the second event with the lower priority after the higher priority event frame is transmitted, the lower priority event frame including an error bit.

Embodiment 44: A serial communication link receiver, comprising: receive circuitry configured to receive an event frame, wherein the event frame includes delay bits indicative of a delay between an event that occurred at a transmitter relative to a predefined bit position in a previous frame received from the transmitter; and delay circuitry configured to: decode the delay bits; wait a number of clock cycles corresponding to the
20 delay bits; and assert a receiver-side event after waiting the number of clock cycles.

Embodiment 45: The serial communication link receiver according to embodiment 44, wherein the delay circuitry is configured to wait by counting the number of clock cycles starting at a predefined point in the event frame.

Embodiment 46: The serial communication link receiver according to any one of
25 Embodiments 44 or 45, wherein the predefined point in the event frame is the last bit of the event frame.

Embodiment 47: The serial communication link receiver according to any one of Embodiments 44 through 46, wherein the receiver-side event corresponds to the event that occurred at the transmitter.

Embodiment 48: The serial communication link receiver according to any one of
30 Embodiments 44 through 47, wherein the delay bits correspond to no delay.

Embodiment 49: The serial communication link receiver according to any one of Embodiments 44 through 48, wherein the delay circuitry is configured to decode event

indicator bits and assert the receiver-side event responsive to the event indicator bits, wherein the event indicator bits indicate which event of a set of events the event frame corresponds to.

Embodiment 50: The serial communication link receiver according to any one of
5 Embodiments 44 through 49, wherein the delay circuitry is configured to decode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

Embodiment 51: A method of receiving events over a serial communications link,
10 comprising: receiving an event frame; decoding delay bits from the event frame, the delay bits indicative of a delay between an event that occurred at a transmitter relative to a predefined bit position in a previous frame; waiting a number of clock cycles corresponding to the delay bits; and asserting a receiver-side event after the waiting.

Embodiment 52: The method according to Embodiment 51, wherein waiting the
15 number of clock cycles corresponding to the delay bits comprises counting the number of clock cycles from a predefined point in the event frame.

Embodiment 53: The method according to any one of Embodiments 51 or 52, wherein the predefined point in the event frame is the last bit in the event frame.

Embodiment 54: The method according to any one of Embodiments 51 through 53,
20 further comprising: decoding event indicator bits from the event frame, wherein the event indicator bits indicate which event of a set of events the event frame corresponds to; and wherein asserting the receiver-side event is responsive to the event indicator bits.

Embodiment 55: The method according to any one of Embodiments 51 through 54, wherein the event frame is received back-to-back with the previous frame.

25 Embodiment 56: The method according to any one of Embodiments 51 through 55, wherein the delay bits are indicative of no delay.

Embodiment 57: A serial communication link, comprising: a transmitter and a receiver configured to define a uniform latency between a transmitter-side event and a receiver-side event; the transmitter comprising: delay circuitry configured to receive the
30 transmitter-side event and delay the transmitter-side event by a delay time corresponding to a frame time; and control circuitry configured to prevent additional frames from being started while the transmitter-side event is being delayed; and transmission circuitry configured to send an event frame corresponding to the transmitter-side event after the

delay time, wherein the transmission circuitry is configured to include in the event frame, event identifier bits indicating the frame being transmitted is an event frame; and the receiver comprising: receive circuitry configured to: decode the frame being transmitted as the event frame; and assert the receiver-side event responsive to the decoding.

- 5 Embodiment 58: The serial communication link according to Embodiment 57, wherein the delay circuitry comprises a shift register of a size corresponding to a number of bits corresponding to the frame time.

- Embodiment 59: The serial communication link according to any one of Embodiments 57 or 58, wherein the delay circuitry comprises a counter and a register, the
10 counter configured to count a number corresponding to a number of bits corresponding to the frame time.

- Embodiment 60: The serial communication link according to any one of Embodiments 57 through 59, wherein the transmission circuitry is configured to encode frames based on a protocol selected from a group consisting of a Universal Asynchronous
15 Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

- Embodiment 61: The serial communication link according to any one of Embodiments 57 through 60, wherein the receive circuitry is configured to decode frames based on a protocol selected from a group consisting of a Universal Asynchronous
20 Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

CLAIMS

What is claimed is:

- 5 1. A serial communication link transmitter, comprising:
delay circuitry configured to determine an event delay between a predefined bit position of
an ongoing frame being transmitted and an event; and
transmission circuitry configured to send an event frame after the ongoing frame, wherein
the transmission circuitry is configured to include in the event frame, delay bits
10 corresponding to the event delay and event identifier bits indicative of the event to
which the event frame corresponds.
2. The serial communication link transmitter of claim 1, wherein the delay
circuitry comprises a delay counter configured to determine the event delay by counting a
15 number of clocks from the event to the predefined bit position of the ongoing frame.
3. The serial communication link transmitter of claim 1, further comprising:
two or more event sub-modules, wherein a first of the two or more event sub-modules
corresponds to a first event type and a second of the two or more event sub-modules
20 corresponds to a second event type, and
priority logic configured to associate a different priority with each of the first event type
and the second event type.
4. The serial communication link transmitter of claim 1, further comprising
25 priority logic, and wherein responsive to an occurrence of a first event corresponding to a
first event type and a second event corresponding to a second event type the priority logic
is configured to:
transmit one of the first event and the second event with a higher priority; and
discard one of the first event and the second event with a lower priority.

5. The serial communication link transmitter of claim 1, further comprising priority logic, and wherein responsive to an occurrence of a first event corresponding to a first event type and a second event corresponding to a second event type the priority logic is configured to:

- 5 transmit a higher priority event frame corresponding to one of the first event and the second event with a higher priority;
hold one of the first event and the second event with a lower priority while the higher priority event frame is transmitted; and
transmit a lower priority event frame corresponding to one of the first event and the second
10 event with the lower priority after the higher priority event frame is transmitted, the lower priority event frame including an error bit.

6. The serial communication link transmitter of claim 1, wherein the transmission circuitry is configured to send the event frame back-to-back with the ongoing
15 frame.

7. The serial communication link transmitter of claim 1, wherein the transmission circuitry is configured to encode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal
20 Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

8. The serial communication link transmitter of claim 1, wherein the transmission circuitry is further configured to include event identifier bits in the event
25 frame, wherein the event indicator bits indicate which event of a set of events the event frame corresponds to.

9. A method of transmitting events over a serial communication link,
comprising:
determining an event delay between a predefined bit position of an ongoing frame being
transmitted and an event;
- 5 encoding an event frame corresponding to the event, wherein the event frame includes:
delay bits corresponding to the event delay; and
event identifier bits that indicate the event frame corresponds to the event, and
transmitting the event frame after the ongoing frame.
- 10 10. The method of claim 9, further comprising sending the event frame back-to-
back with the ongoing frame.
11. The method of claim 9, further comprising determining the event delay by
counting a number of clocks from the event to the predefined bit position of the ongoing
15 frame.
12. The method of claim 9, further comprising:
responsive to an occurrence of a first event corresponding to a first event type and a second
event corresponding to a second event type:
20 transmitting one of the first event and the second event with a higher priority; and
discarding of the first event and the second event with a lower priority.
13. The method of claim 9, further comprising:
responsive to an occurrence of a first event corresponding to a first event type and a second
25 event corresponding to a second event type:
transmitting a higher priority event frame corresponding to one of the first event
and the second event with a higher priority;
holding one of the first event and the second event with a lower priority while the
higher priority event frame is transmitted; and
30 transmitting a lower priority event frame corresponding to one of the first event and
the second event with the lower priority after the higher priority event frame
is transmitted, the lower priority event frame including an error bit.

14. A serial communication link receiver, comprising:
receive circuitry configured to receive an event frame, wherein the event frame includes
delay bits indicative of a delay between an event that occurred at a transmitter
relative to a predefined bit position in a previous frame received from the
transmitter; and

5 delay circuitry configured to:
decode the delay bits;
wait a number of clock cycles corresponding to the delay bits; and
assert a receiver-side event after waiting the number of clock cycles.

10

15. The serial communication link receiver of claim 14, wherein the delay
circuitry is configured to wait by counting the number of clock cycles starting at a
predefined point in the event frame.

15 16. The serial communication link receiver of claim 15, wherein the predefined
point in the event frame is the last bit of the event frame.

17. The serial communication link receiver of claim 14, wherein the receiver-
side event corresponds to the event that occurred at the transmitter.

20

18. The serial communication link receiver of claim 14, wherein the delay bits
correspond to no delay.

19. The serial communication link receiver of claim 14, wherein the delay
25 circuitry is configured to decode event indicator bits and assert the receiver-side event
responsive to the event indicator bits, wherein the event indicator bits indicate which event
of a set of events the event frame corresponds to.

20. The serial communication link receiver of claim 14, wherein the delay
30 circuitry is configured to decode frames based on a protocol selected from a group
consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous
Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

21. A method of receiving events over a serial communications link,
comprising:
receiving an event frame;
decoding delay bits from the event frame, the delay bits indicative of a delay between an
5 event that occurred at a transmitter relative to a predefined bit position in a previous
frame;
waiting a number of clock cycles corresponding to the delay bits; and
asserting a receiver-side event after the waiting.

10 22. The method of claim 21, wherein waiting the number of clock cycles
corresponding to the delay bits comprises counting the number of clock cycles from a
predefined point in the event frame.

23. The method of claim 22, wherein the predefined point in the event frame is
15 a last bit in the event frame.

24. The method of claim 21, further comprising:
decoding event indicator bits from the event frame, wherein the event indicator bits
indicate which event of a set of events the event frame corresponds to; and
20 wherein asserting the receiver-side event is responsive to the event indicator bits.

25. The method of claim 21, wherein the event frame is received back-to-back
with the previous frame.

25 26. The method of claim 21, wherein the delay bits are indicative of no delay.

27. A serial communication link transmitter, comprising:
a transmitter and a receiver configured to define a uniform latency between a transmitter-side event and a receiver-side event;

the transmitter comprising:

5 delay circuitry configured to receive the transmitter-side event and delay the transmitter-side event by a delay time corresponding to a frame time; and control circuitry configured to prevent additional frames from being started while the transmitter-side event is being delayed; and

10 transmission circuitry configured to send an event frame corresponding to the transmitter-side event after the delay time, wherein the transmission circuitry is configured to include, in the event frame, event identifier bits indicative that the frame being transmitted is an event frame; and

the receiver comprising:

receive circuitry configured to:

15 decode the frame being transmitted as the event frame; and assert the receiver-side event responsive to the decoding.

28. The serial communication link of claim 27, wherein the delay circuitry comprises a shift register of a size corresponding to a number of bits corresponding to the
20 frame time.

29. The serial communication link of claim 27, wherein the delay circuitry comprises a counter and a register, the counter configured to count a number corresponding to a number of bits corresponding to the frame time.

25

30. The serial communication link of claim 27, wherein the transmission circuitry is configured to encode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

30

- 39 -

31. The serial communication link of claim 27, wherein the receive circuitry is configured to decode frames based on a protocol selected from a group consisting of a Universal Asynchronous Receiver/Transmitter, a Universal Synchronous Receiver/Transmitter, or a Universal Synchronous/Asynchronous Receiver/Transmitter.

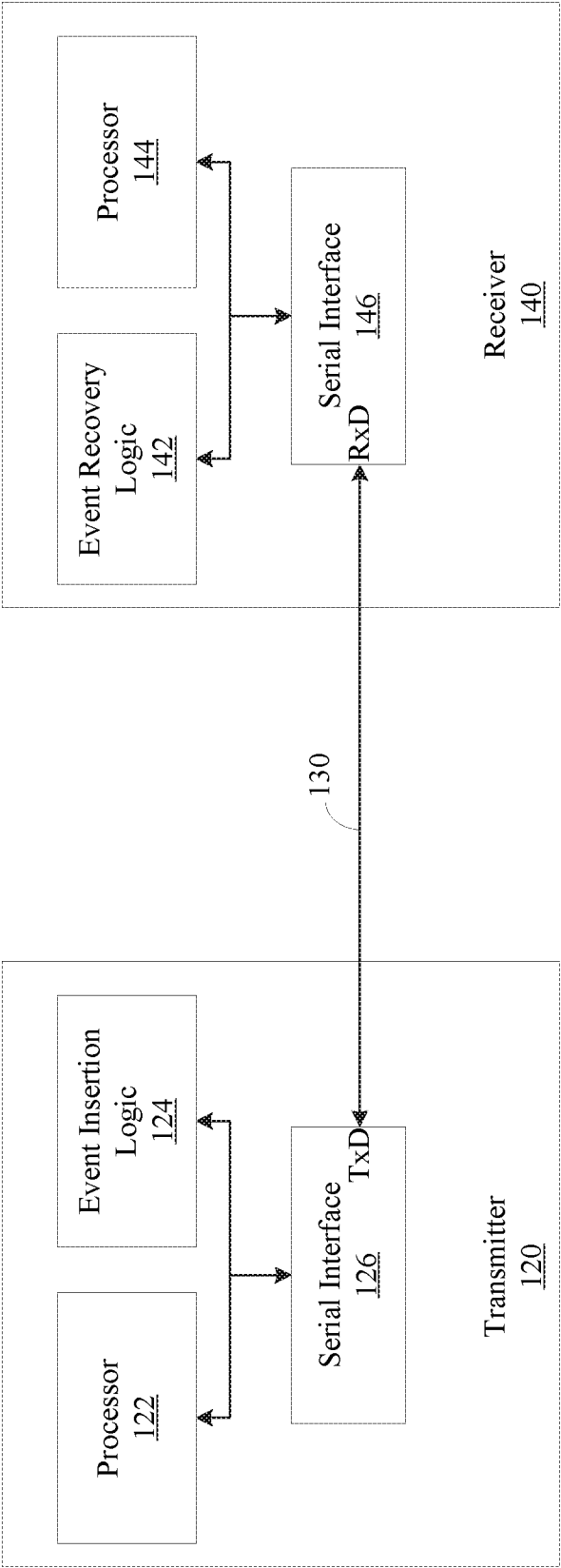
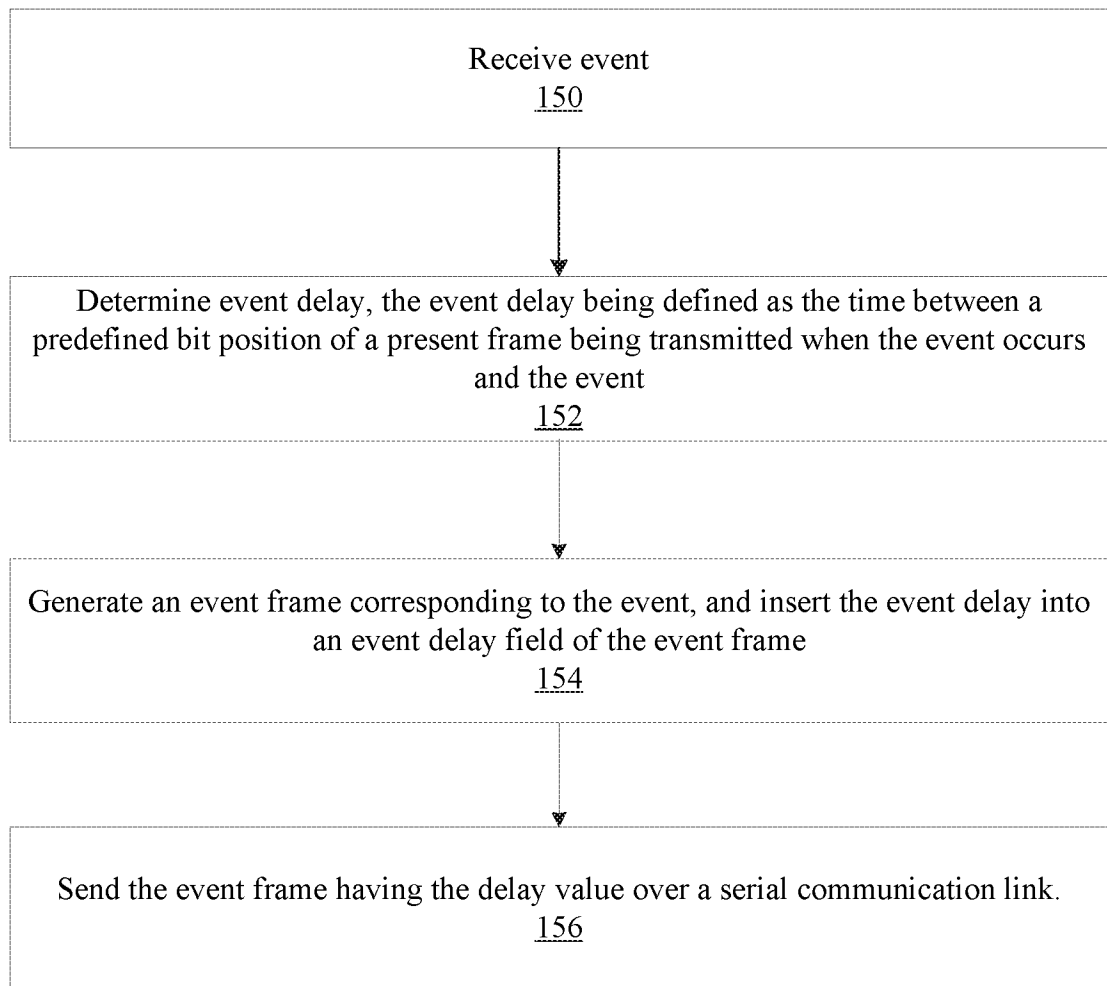
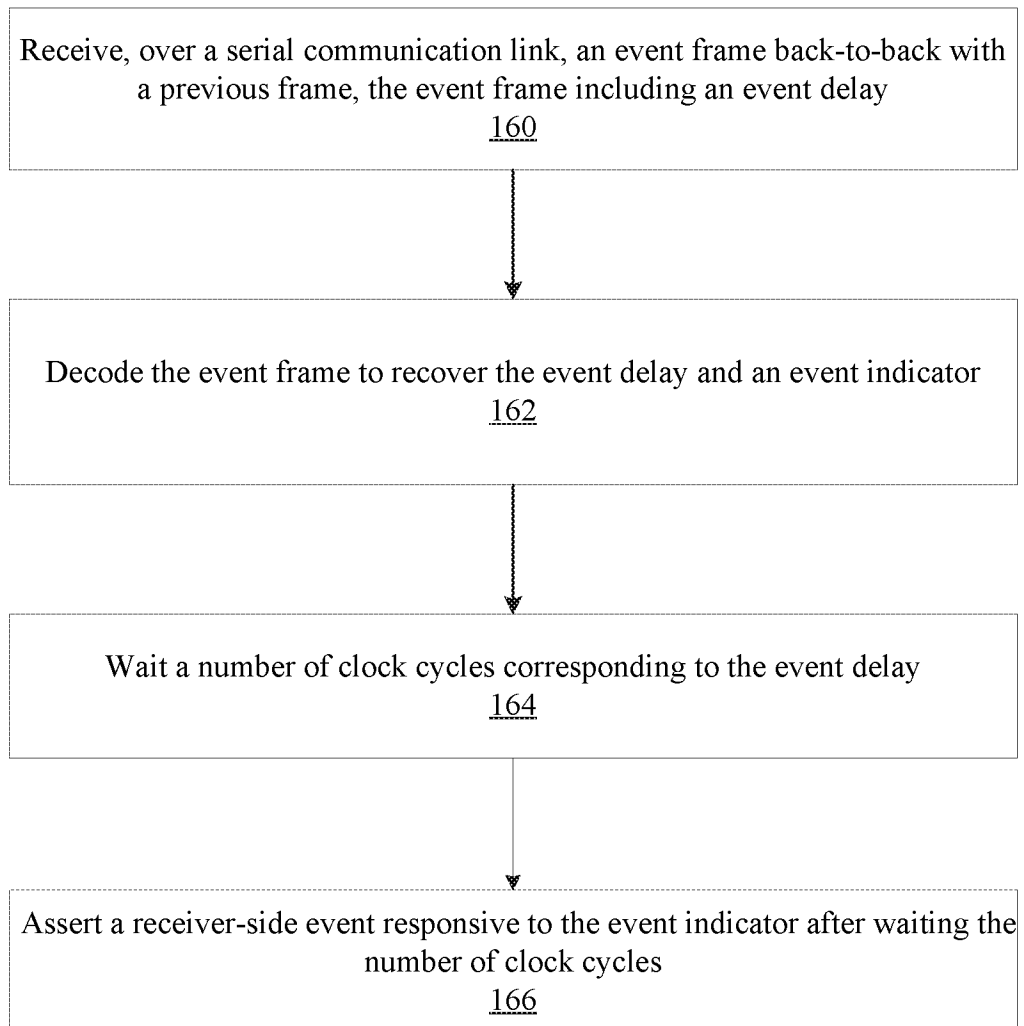
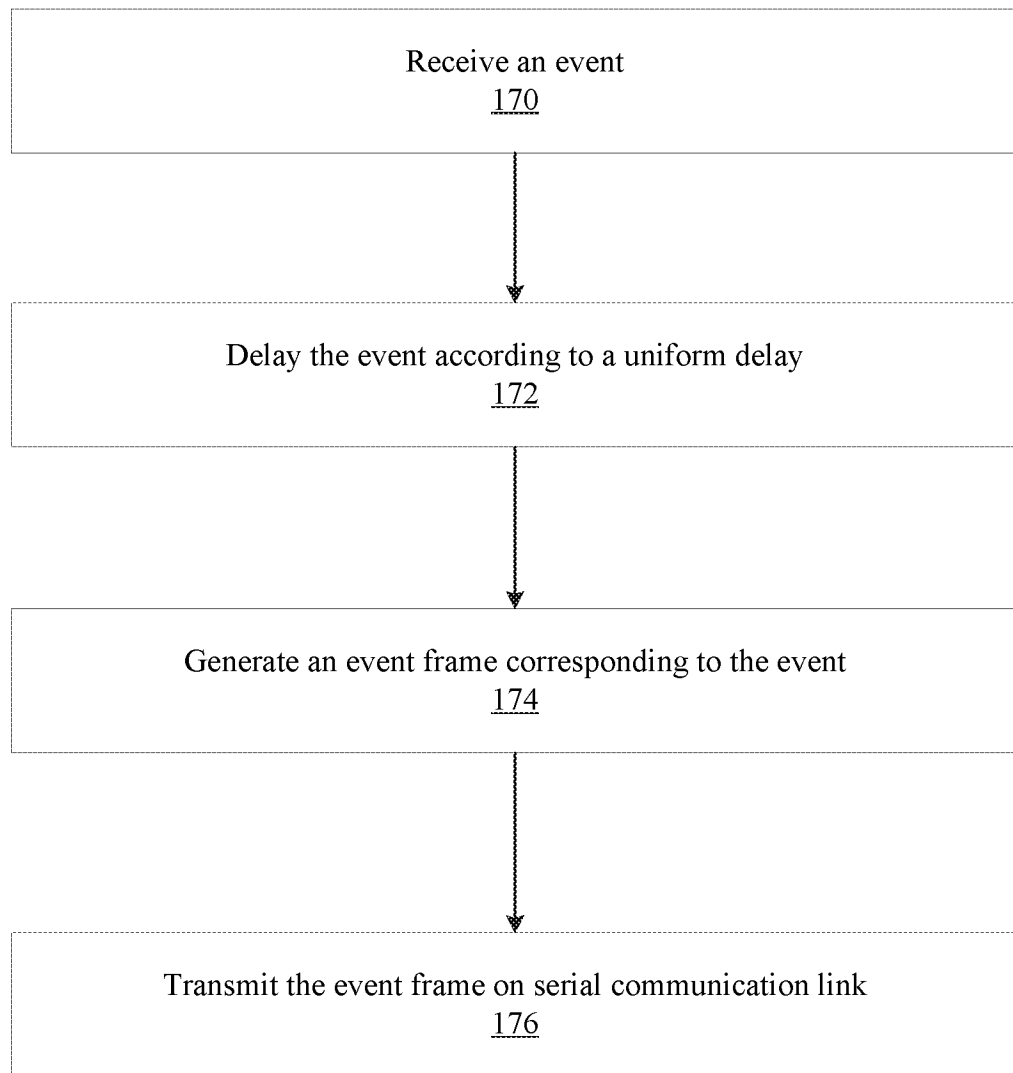
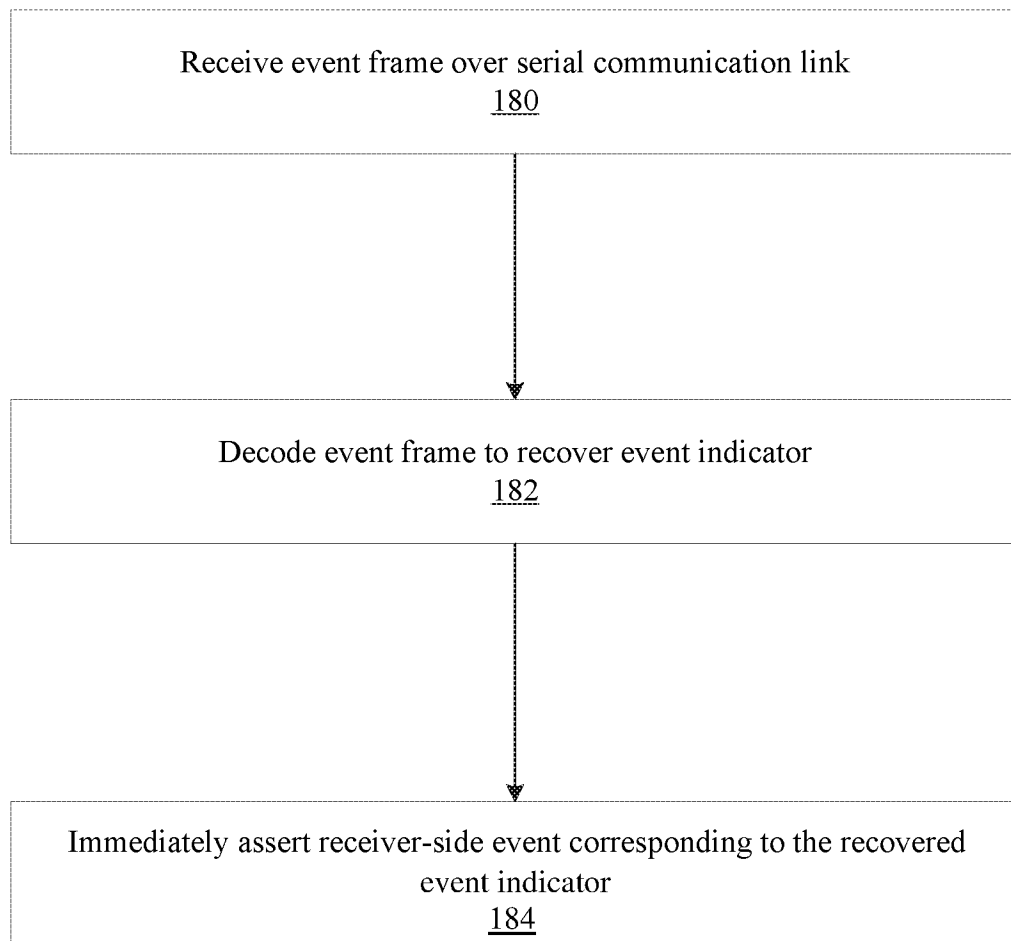


FIG. 1A

**FIG. 1B**

**FIG. 1C**

**FIG. 1D**

**FIG. 1E**

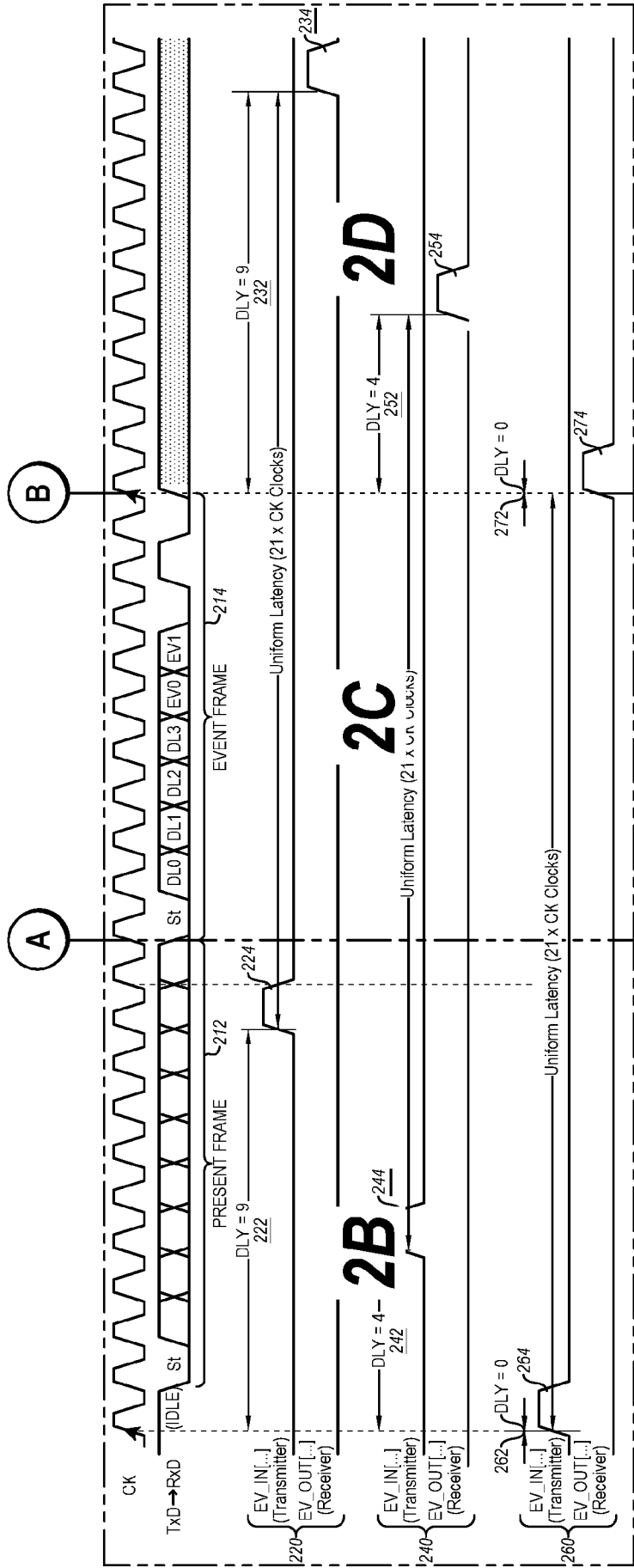


FIG. 2A

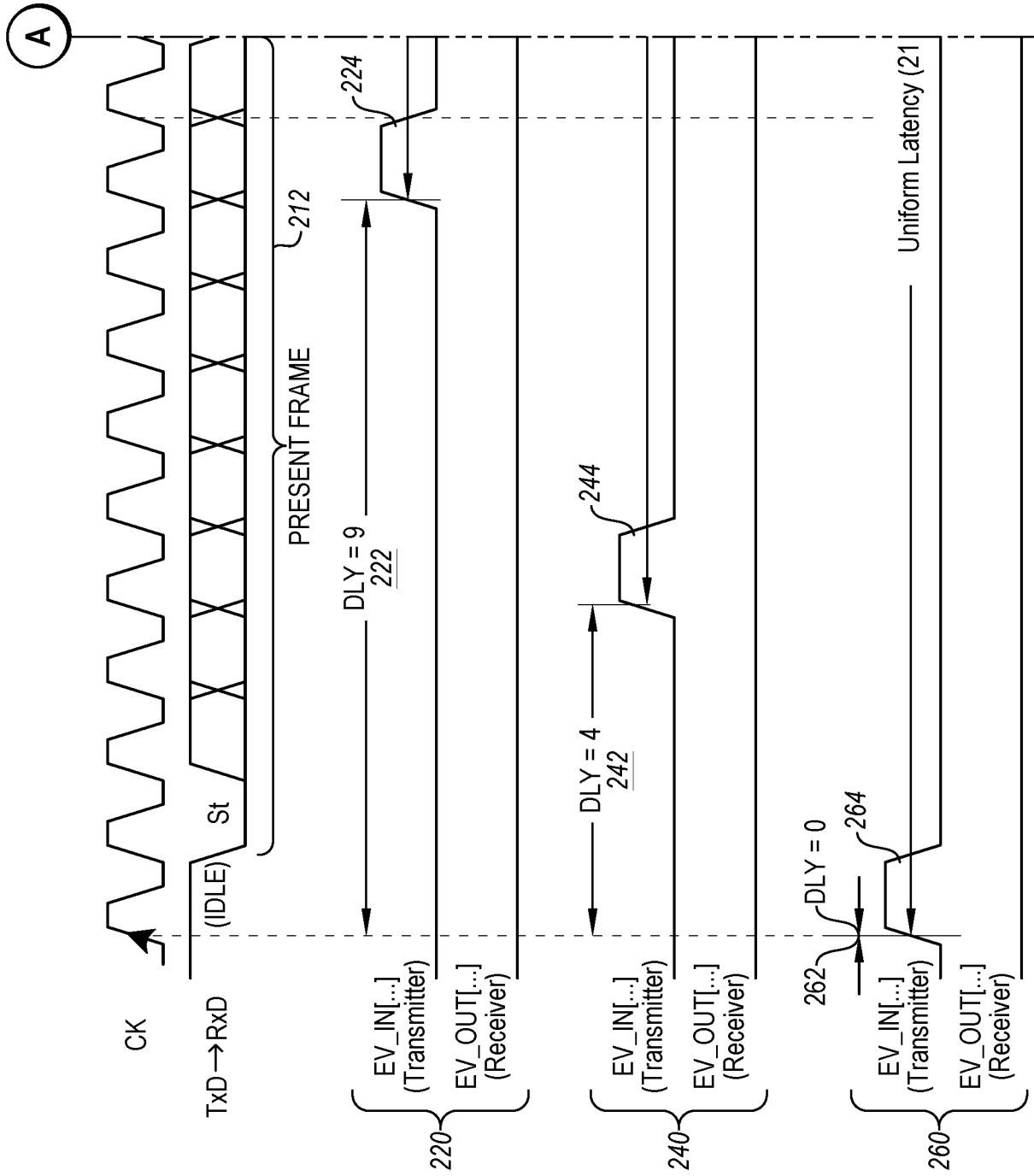


FIG. 2B

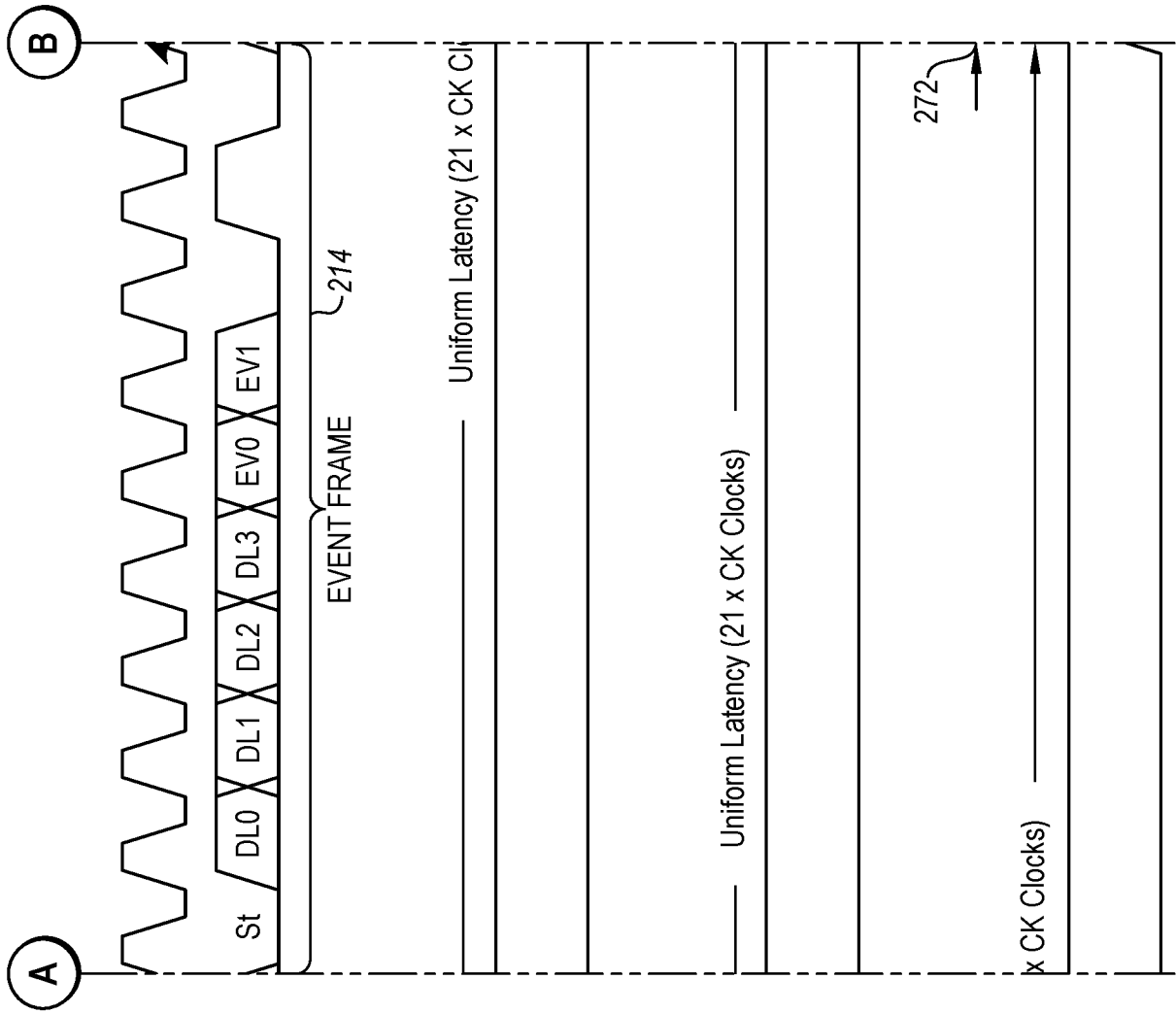


FIG. 2C

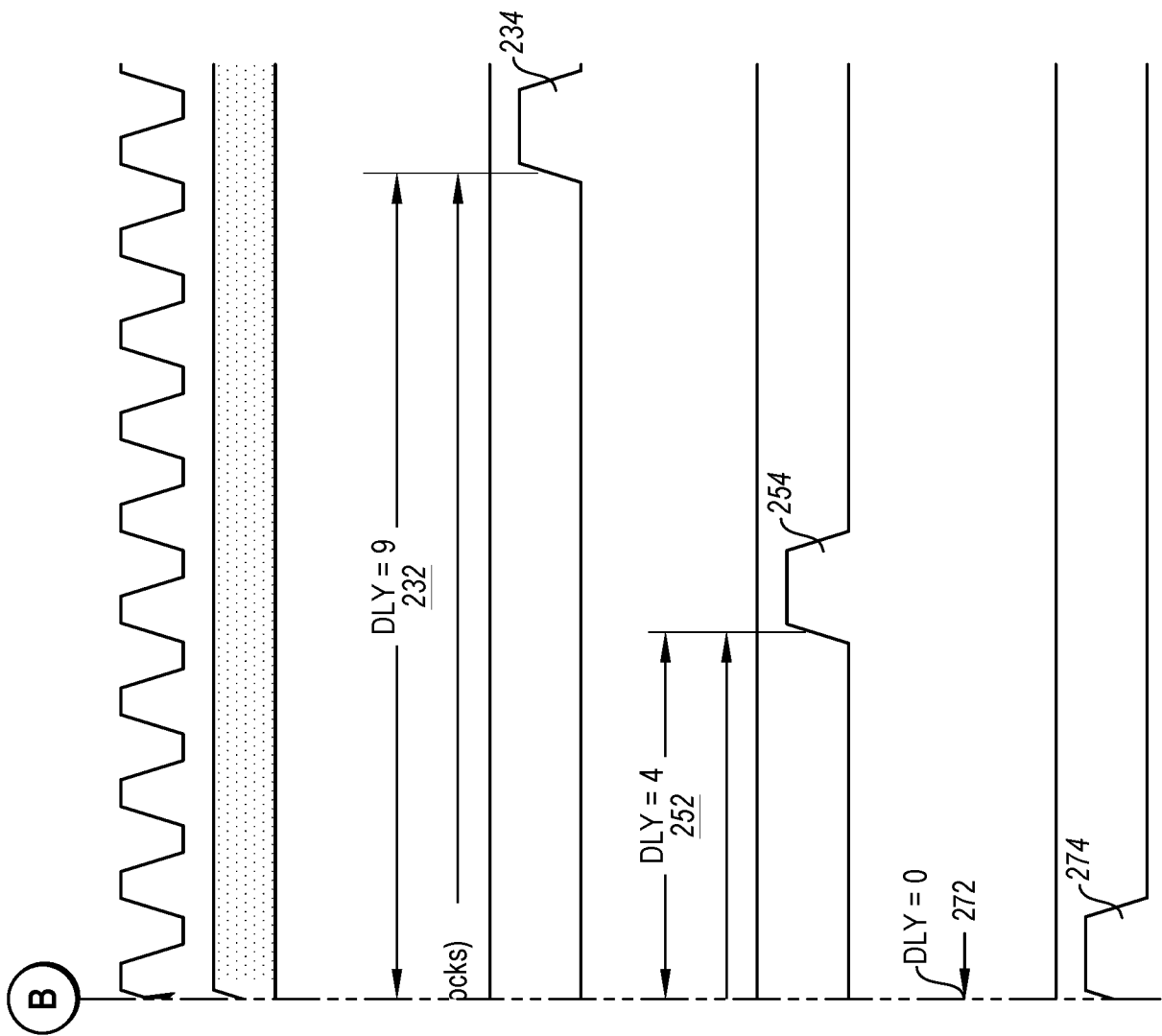


FIG. 2D

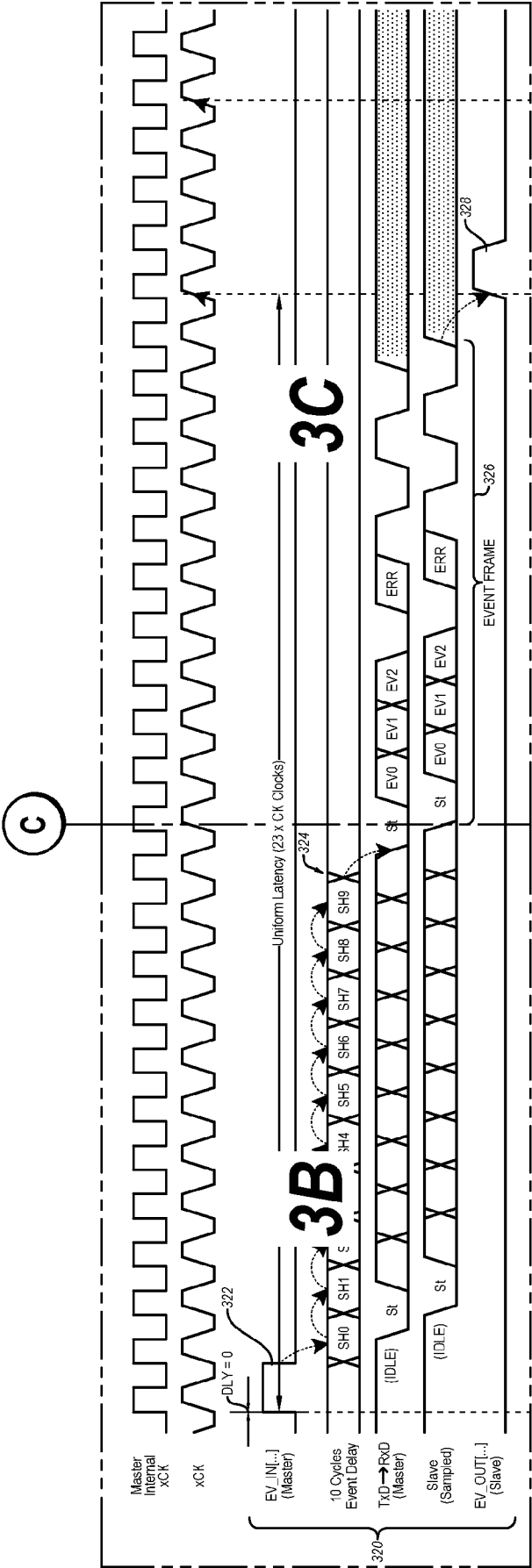


FIG. 3A

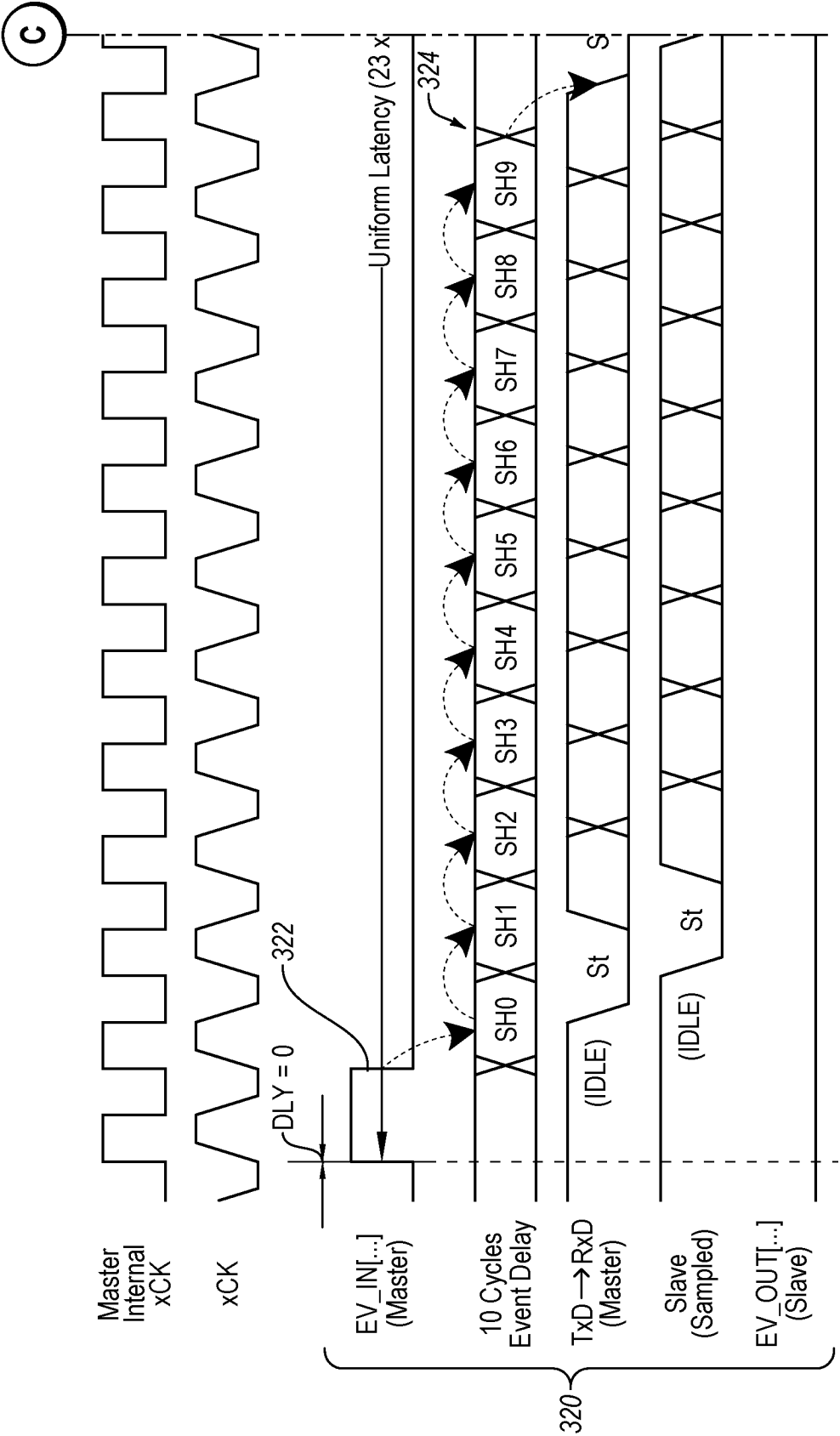


FIG. 3B

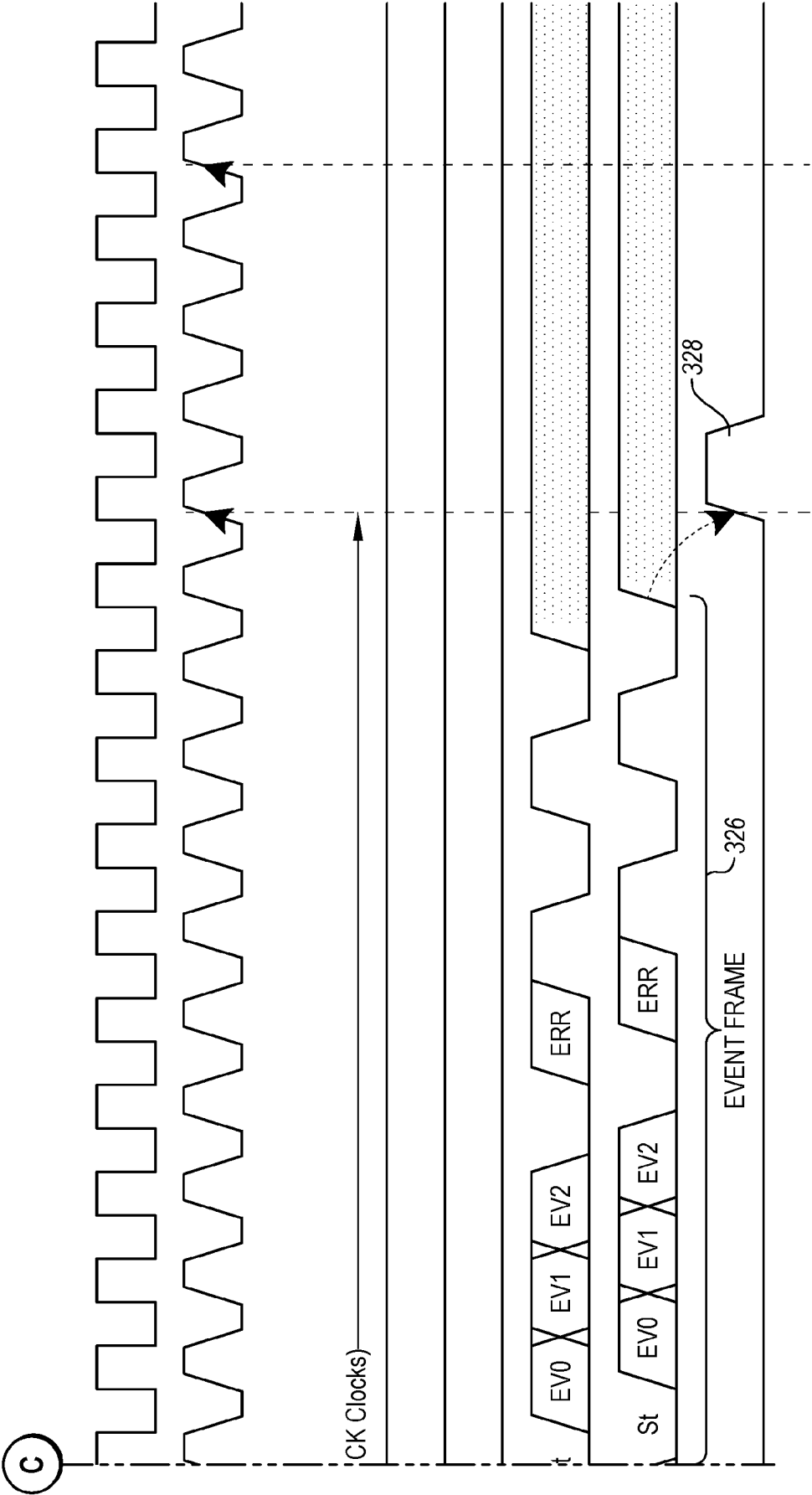


FIG. 3C

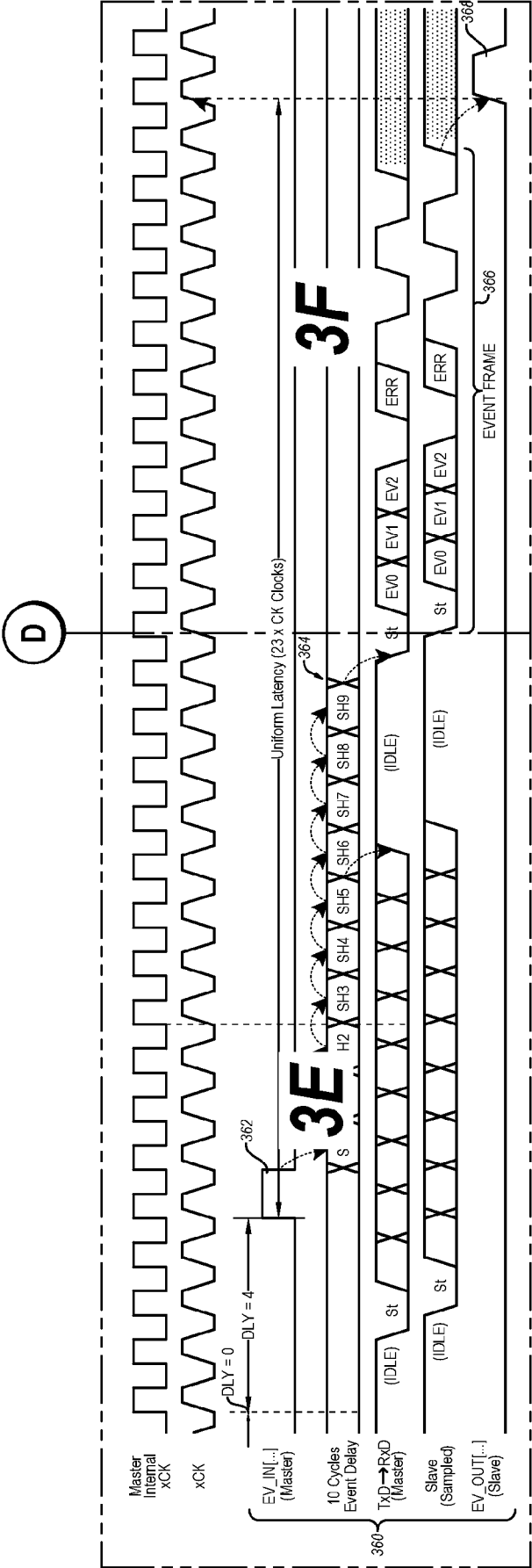


FIG. 3D

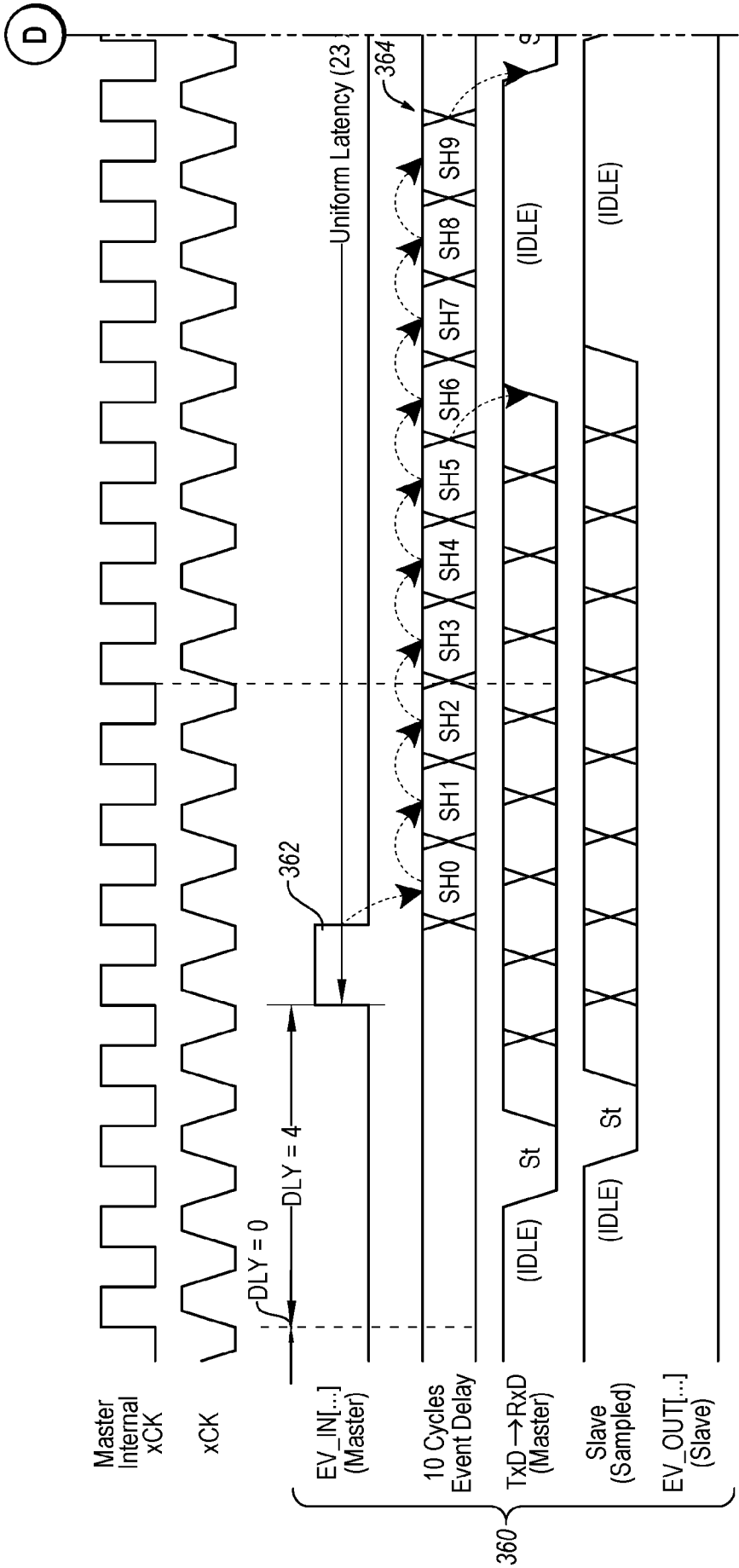


FIG. 3E

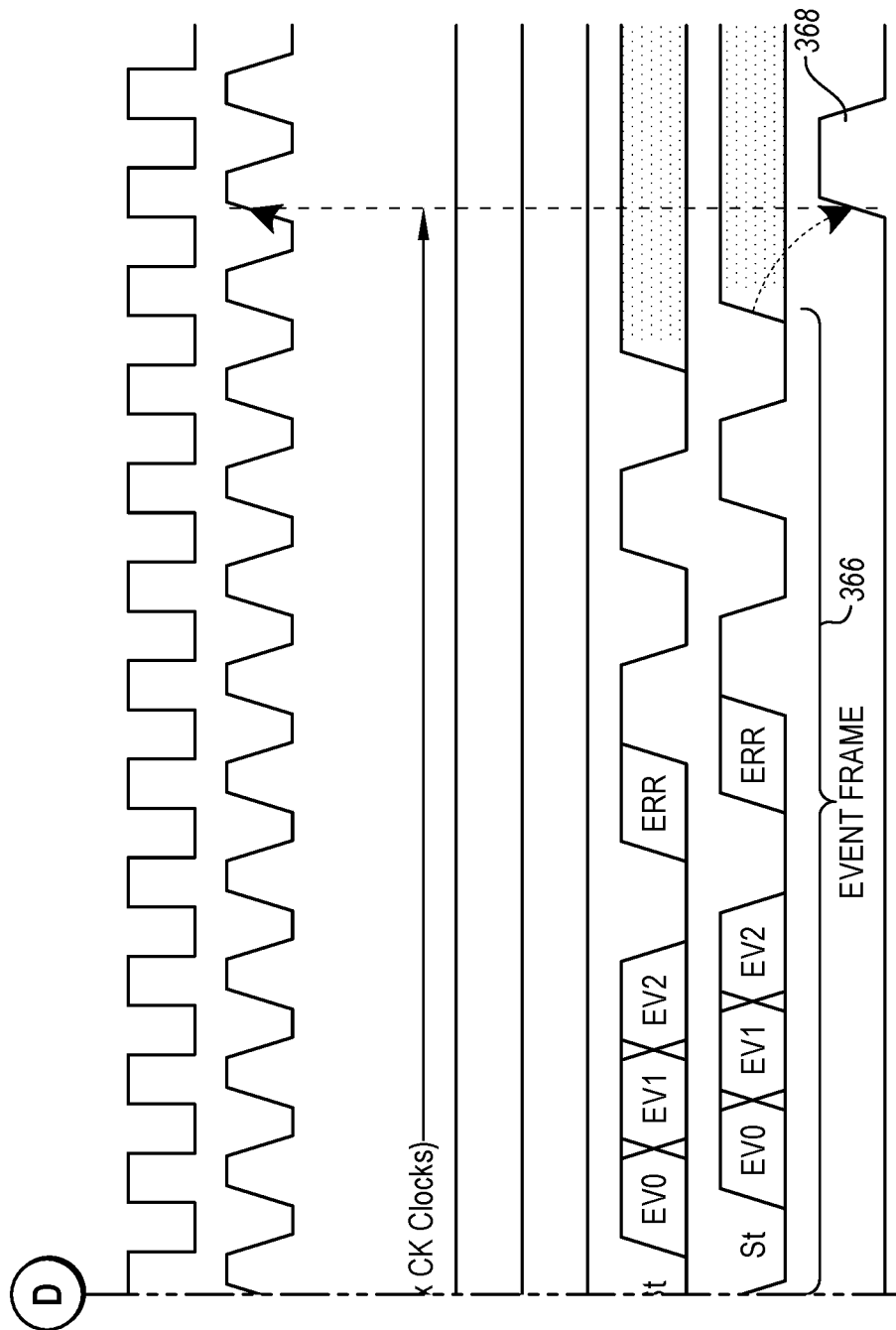
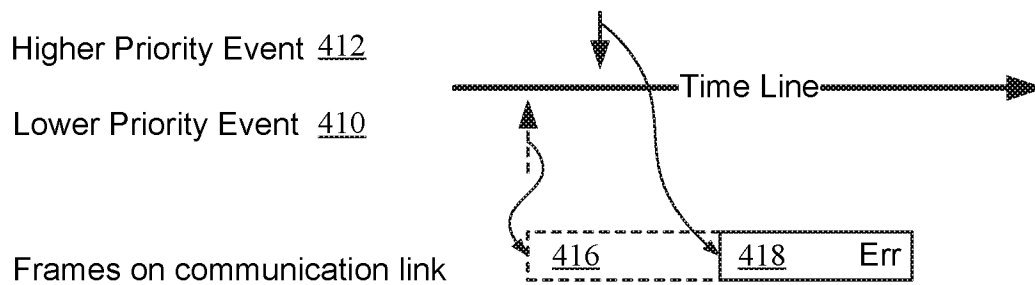
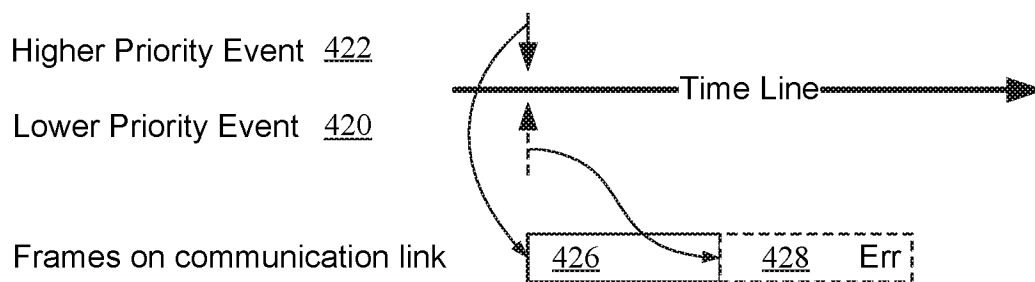
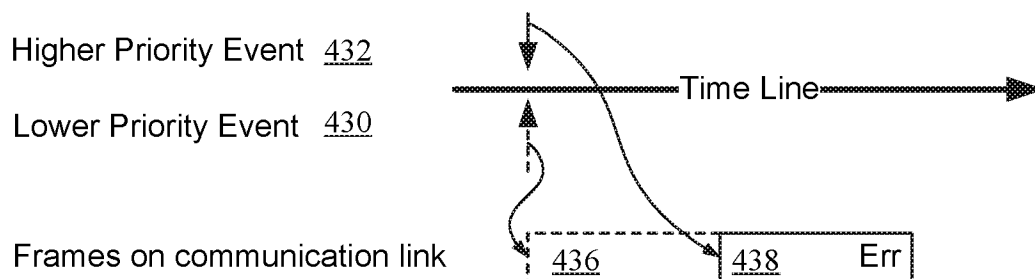


FIG. 3F

**FIG. 4A****FIG. 4B****FIG. 4C**

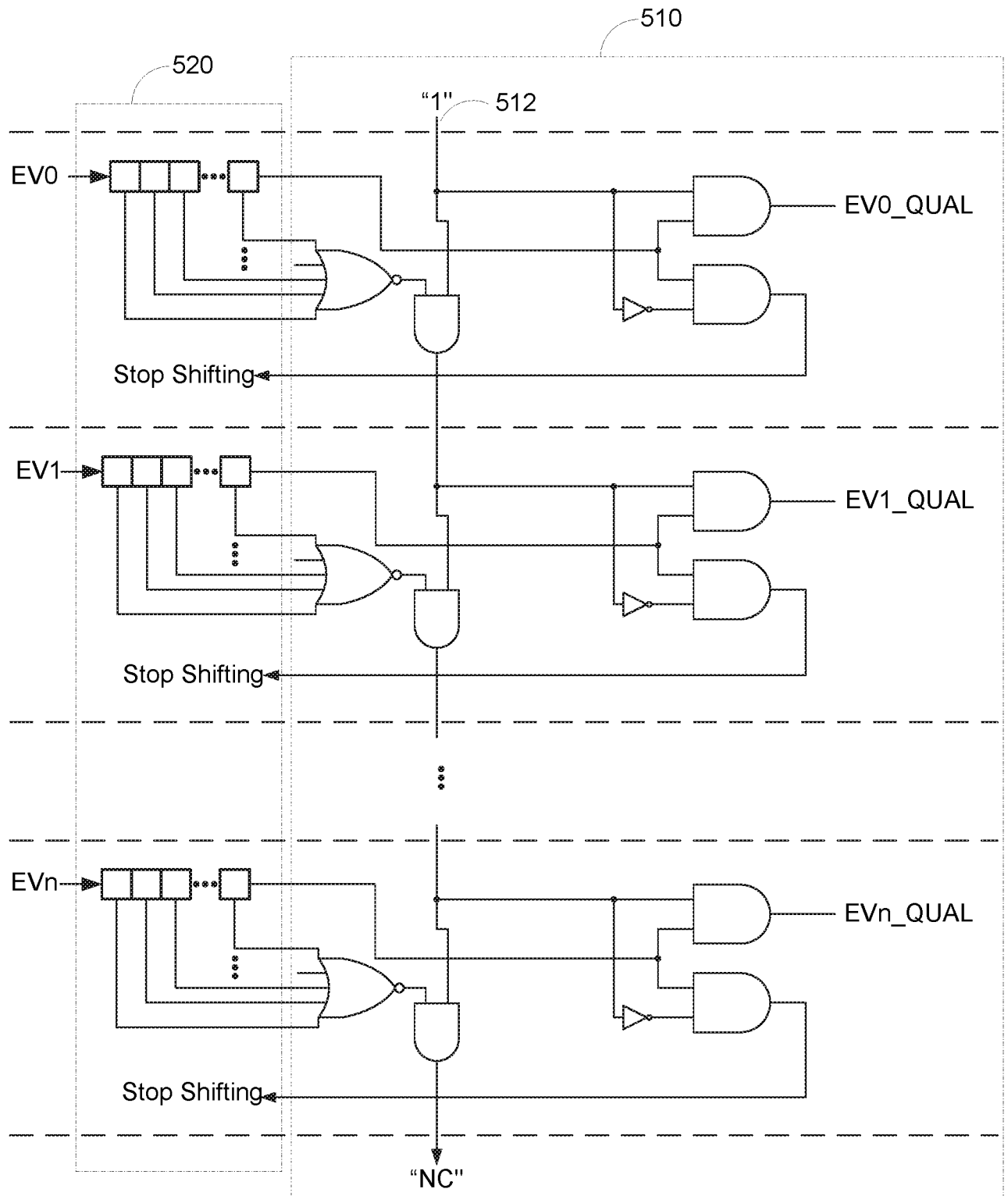


FIG. 5

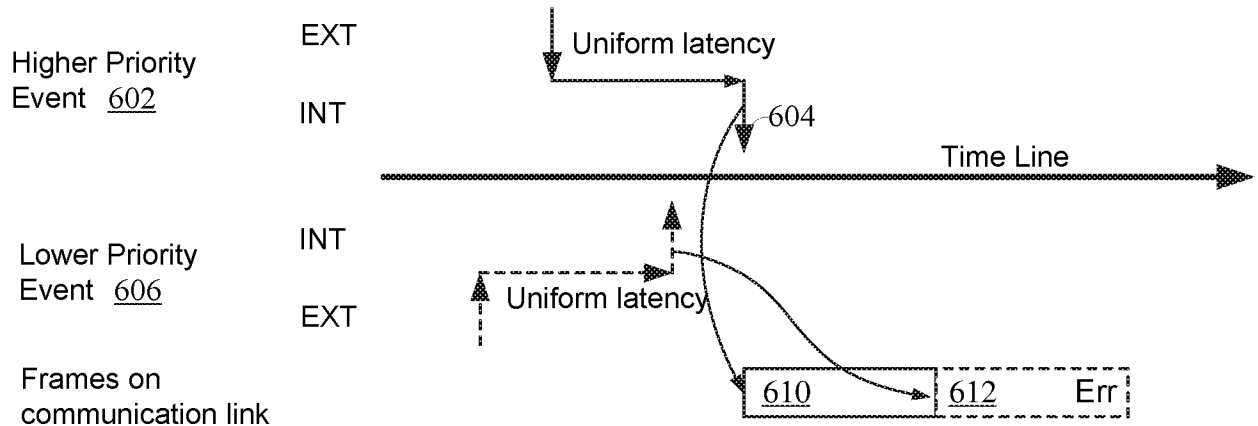


FIG. 6A

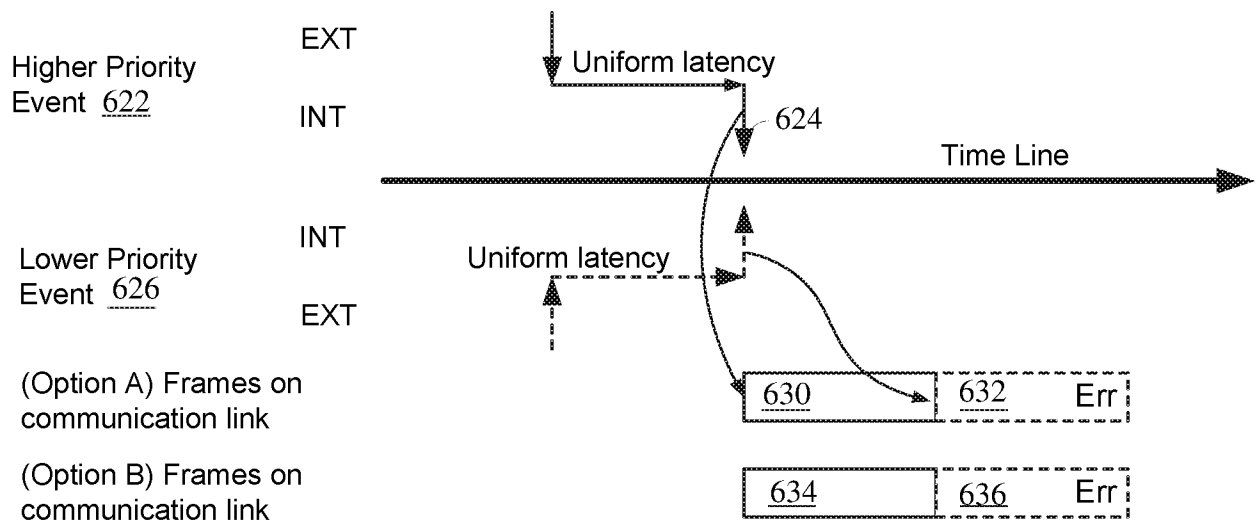


FIG. 6B

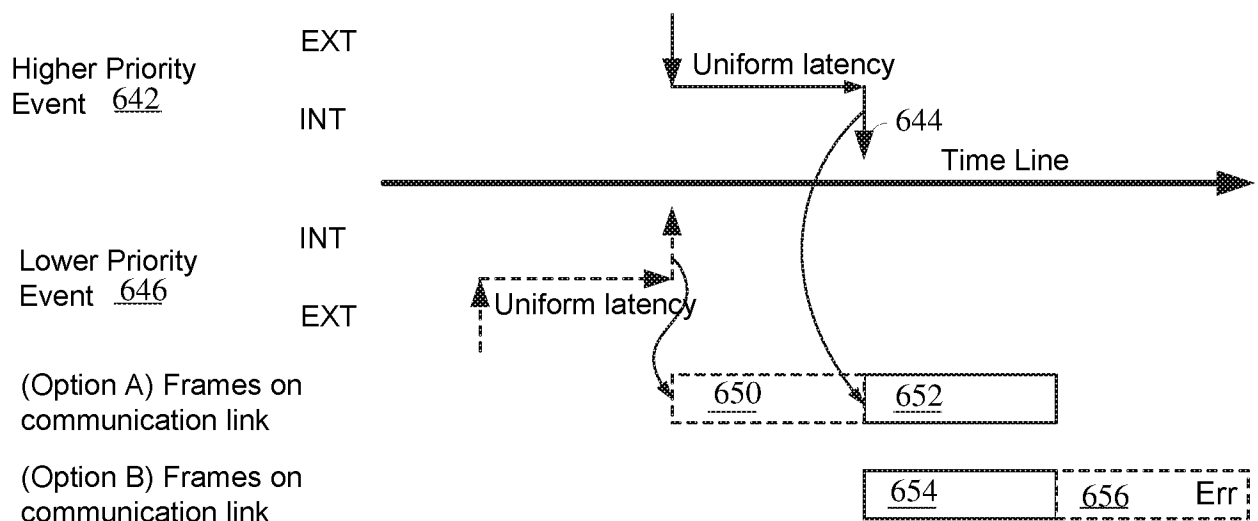
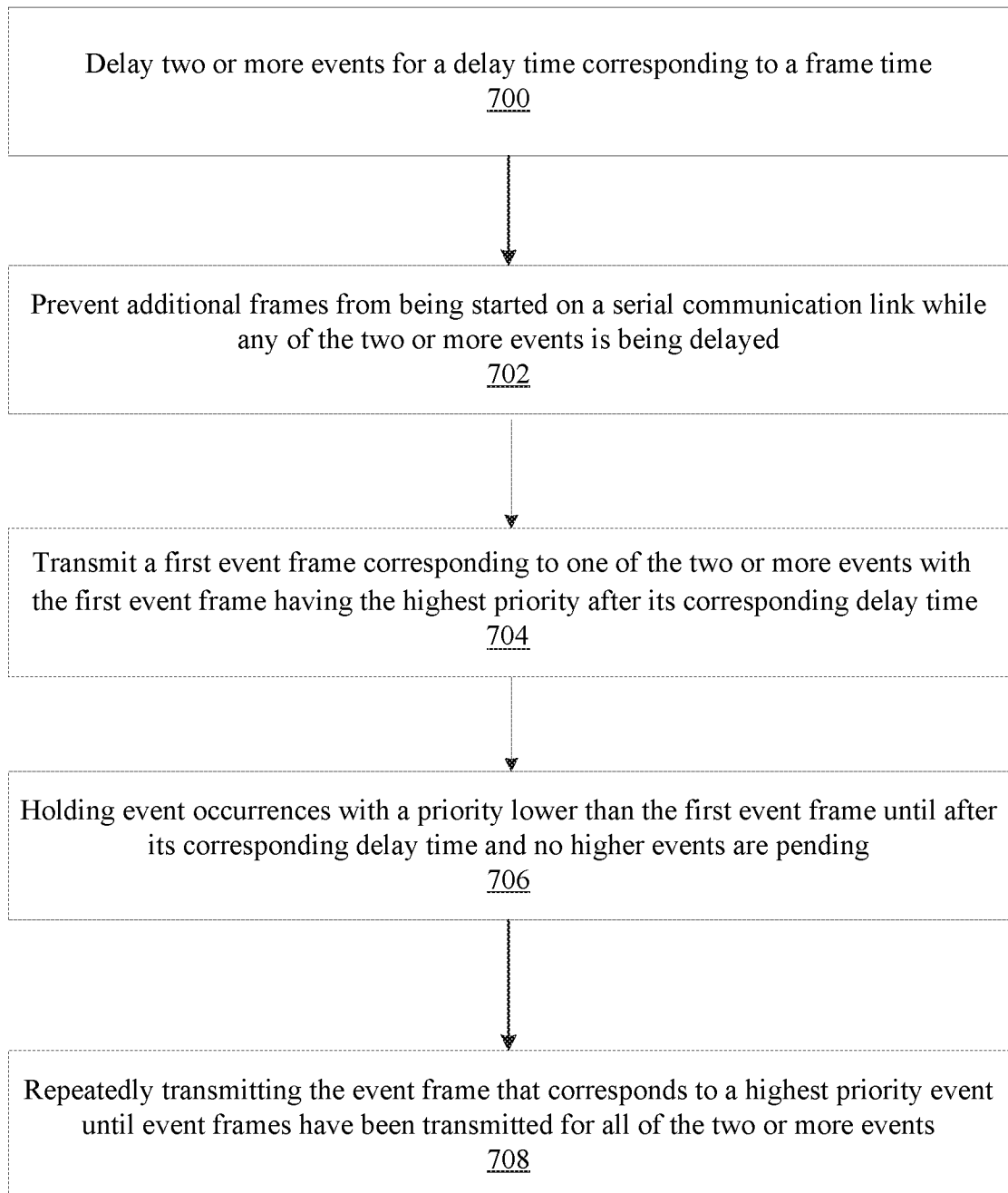


FIG. 6C

**FIG. 7**

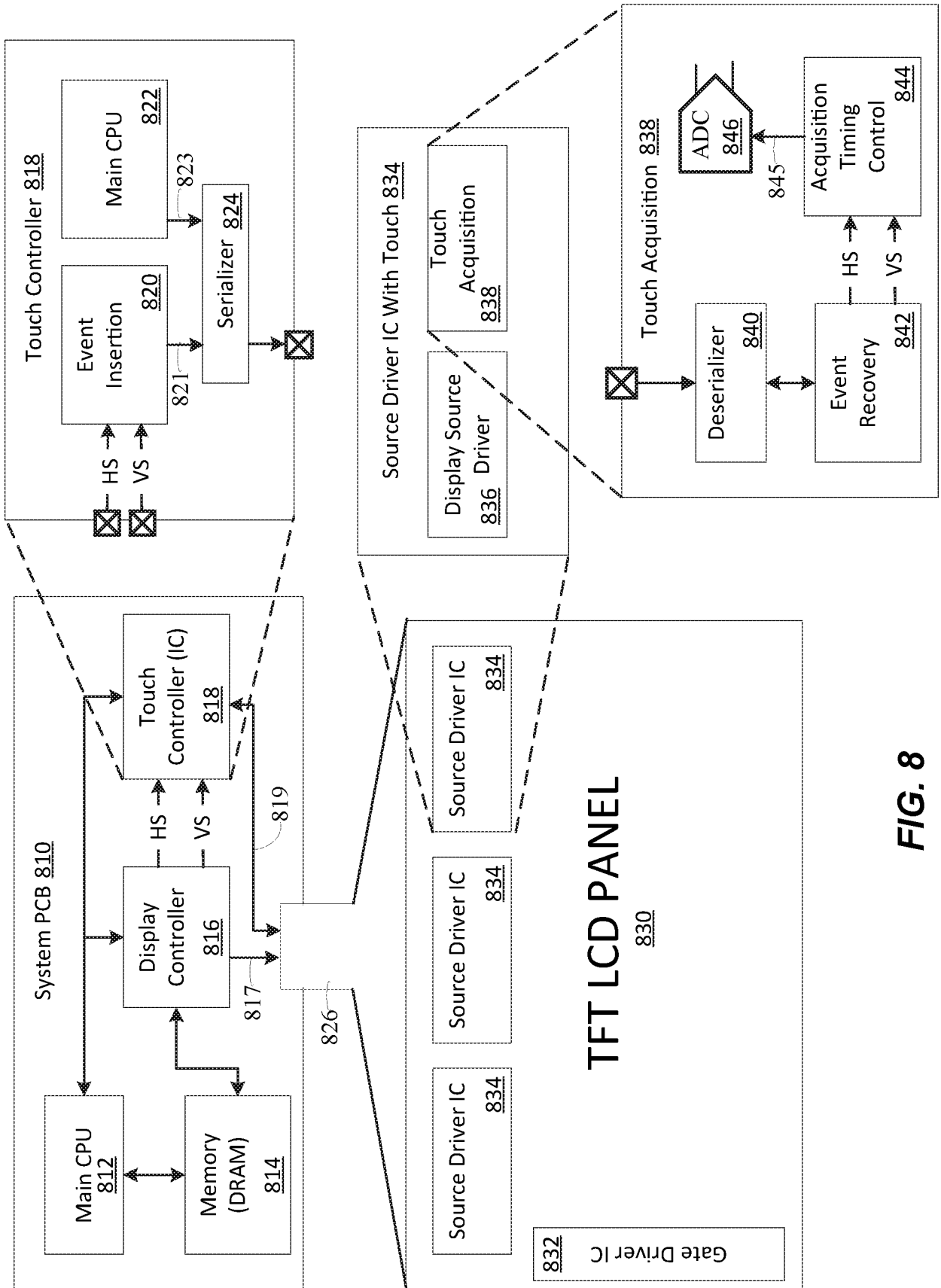


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/030507

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F9/54 G06F13/42 G06F3/041 G09G5/00
ADD.

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 G09G

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EP0-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/147684 A1 (SENGOKU SHOICHIRO [US]) 26 May 2016 (2016-05-26) abstract paragraph [0079] - paragraph [0094] paragraph [0150] - paragraph [0159] paragraph [0163] - paragraph [0170] figures 8, 9, 41, 43 -----	1-26

☐

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

2 August 2018

Date of mailing of the international search report

01/10/2018

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Papadakis, Georgios

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2018/030507

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of Item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-26

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International Application No. PCT/ US2018/ 030507

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-26

Timestamping and asserting a variably delayed event
communicated on a serial communication link

2. claims: 27-31

Providing fixed latency communication of events in a serial
link

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2018/030507

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016147684 A1	26-05-2016	CN 107003958 A	01-08-2017
		EP 3224730 A1	04-10-2017
		US 2016147684 A1	26-05-2016
		WO 2016086031 A1	02-06-2016
