



(51) International Patent Classification:

G06F 13/38 (2006.01) H04N 21/6334 (2011.01)
G06F 12/02 (2006.01) H04N 21/835 (2011.01)
G06F 13/12 (2006.01)

(21) International Application Number:

PCT/IB2014/002048

(22) International Filing Date:

7 October 2014 (07.10.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: ELLIPTIC TECHNOLOGIES INC.
[CA/CA]; 540-400 March Rd., Kanata, ON K2K 3H4
(CA).

(72) Inventors: ELIAS, Andrew, Alexander; 50 Hansen Avenue, Ottawa, Ontario K2K 2C8 (CA). THIBAUT, Jean-Pierre; 199 Harcourt Avenue, Ottawa, Ontario K2B 5C2 (CA). BOWLER, Nick; 17 Carmichael Court, Ottawa, Ontario K2K 1K1 (CA). LOUGHEED, Steven; 200 Parkin Circle, Ottawa, Ontario K1T 4G4 (CA). LEWIS, Michael, James; 25 Erskine Place, Ottawa, Ontario K2H 8R8 (CA).

(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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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).

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

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

(54) Title: SIDE CHANNEL COMMUNICATION HARDWARE DRIVER

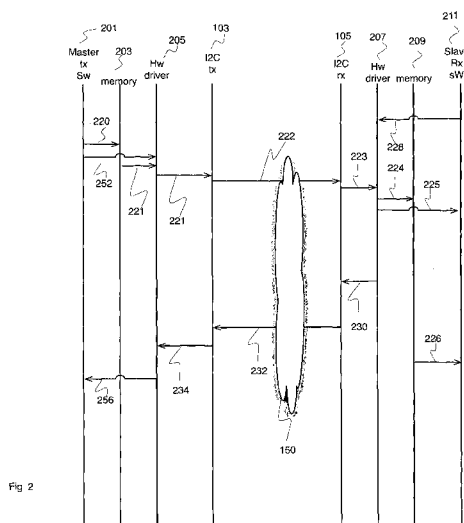


Fig 2

(57) Abstract: A system and method for communicating data between a first software and a second software located on first and second devices, respectively, has a hardware driver and memory associated with each device. Each communication of data from the first software to the second software allocates memory to manage data to be communicated from the first software to the second software, provides memory allocation information to the hardware driver associated with the first software, and transmits the data from the first hardware driver to the second hardware driver for delivery to the second software via the memory associated with the second software.



SIDE CHANNEL COMMUNICATION HARDWARE DRIVER**FIELD OF THE INVENTION**

[0001] The present disclosure relates to encryption and content protection, specifically to the control channel used for authentication.

SUMMARY

[0002] In accordance with one embodiment, a system and method are provided for communicating data between a first software and a second software located on first and second devices, respectively. Each of the devices has a hardware driver and memory associated therewith. Each communication of data from the first software to the second software allocates memory to manage data to be communicated from the first software to the second software, provides memory allocation information to the hardware driver associated with the first software, and transmits the data from the first hardware driver to the second hardware driver for delivery to the second software via the memory associated with the second software. Each communication of data from the second software to the first software allocates memory to manage data to be communicated with the first software, provides memory allocation information to the first hardware driver, and transmits the data from the second hardware driver to the first hardware driver for delivery to the first software via the first memory.

[0003] In one implementation, the first software is a master software and the second software is a slave software. The hardware drivers preferably handle the time-critical components of the communication of data between the first and second software, to reduce the number of interrupts to the first and second software. The hardware driver that receives communicated data stores that data in the associated memory, and notifies the associated software that the data is available.

[0004] The foregoing and additional aspects and embodiments of the present disclosure will be apparent to those of ordinary skill in the art in view of the detailed description of various embodiments and/or aspects, which is made with reference to the drawings, a brief description of which is provided next.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 illustrates a prior art message exchange for authentication over I2C.

[0006] FIG. 2 illustrates an example of the use of a hardware driver for the sideband communication channel when a master writes to a slave.

[0007] FIG. 3 illustrates an example of the use of a hardware driver for the sideband communication channel when a master reads from a slave.

[0008] While the present disclosure is susceptible to various modifications and alternative forms, specific embodiments or implementations have been shown by way of example in the drawings and will be described in detail herein. It should be understood, however, that the disclosure is not intended to be limited to the particular forms disclosed. Rather, the disclosure is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of an invention as defined by the appended claims.

DETAILED DESCRIPTION

[0009] The HDCP content protection standard uses a side-band communication channel to perform authentication between an HDCP transmitter and an HDCP receiver. There are time limits for completing the authentication process. Certain steps within the authentication algorithm are also subject to time limits. While HDCP version 1.x was amenable to hardware (HW) implementation, the HDCP version 2.x is more complex and lends itself best to software (SW) implementation. However, the overhead of SW responding to interrupts (to service transfers) can jeopardize completion of the authentication within the allotted time(s).

[0010] Although the embodiment is described herein using HDCP as an example, it applies to any configurations where software performs time-sensitive data exchanges over a physical hardware channel, while the frequency of software interrupts need to be limited.

[0011] In the case of HDCP over HDMI, the side-band channel is Inter-integrated Circuit (I2C). When an I2C slave device cannot immediately respond to a transfer request, it is allowed to delay its response through the use of a mechanism called clock stretching. In practice, some I2C devices do not support clock stretching and are not able to inter-operate with compliant devices, therefore the avoidance of clock stretching is desirable, which requires the I2C slave to be able to respond to transfer requests with minimal delay.

[0012] Referring to FIG. 1, the master transmit SW 101 starts an authentication process 110 via a I2C transmit side-channel 103, the start notification 111 is sent via a network 150 and is received in the I2C receive side 105 of the slave which generates an interrupt 112 for the slave receive SW 107. The receive SW transmits the required data 113 via the I2C side channel 105 which transmits 114 over the network 150. On the master side,

the reception of the data 114 generates an interrupt 115 to the master SW 101. Several exchanges of data occur during the authentication process each involving interrupts to the master or slave SW, along with unpredictable response time.

[0013] In the case of HDCP over DisplayPort (DP), side-channel transfers are conducted over the auxiliary channel (AUX) and are limited to bursts of 16 bytes (or less). Larger quantities of data must be broken into multiple bursts, which results in a large number of SW interrupts.

[0014] There is also a need to abstract the side-channel interface from the SW to allow generic SW communication independent of the side-channel used (e.g. I2C or AUX).

[0015] Existing solutions to these problems include implementing the HDCP protocol entirely in HW, which is very costly. Another option is to tune the software to respond to the side-channel interrupts faster (e.g. by increasing the priority of these interrupts). Another solution is to simply tolerate the authentication failures resulting from the delays until the authentication eventually succeeds. However, this last solution results in poor user experience and interoperability problems.

[0016] In one embodiment, the side-channel communication can be off-loaded from the SW and implemented in a HW driver. The SW overhead is reduced and the response time is faster and more predictable therefore the protocol timing requirements can be met. This embodiment also abstracts the communication channel from SW, allowing a common SW to be used for different channels. The SW focuses on high-level messaging and protocol compliance while the HW is used to meet low level timing constraints. The manner in which the offloading is achieved depends on whether it resides in a “master” (transmitting) device or in a “slave” (receiving) device. The partitioning of the HW and SW components allows the SW to be agnostic to the communication channel. The HW driver handles the time-critical components of the communication, minimizing the number of interrupts to the SW. The partitioning allows for HW to handle the communications without waiting for the SW response.

[0017] FIG. 2 provides an example of a master writing to a slave. On the transmit (master) side 201, the SW places data 220 in memory 203. The SW gives the pointer and data size 252 to the HW driver 205. The HW driver 205 reads the data from memory 221 and makes it available to the I2C tx 103 for transmission 222 over the network 150 to the I2C RX 105, which in turns makes the data 223 available to the HW driver slave 207. ON the receive side, independent from the master’s writing process, the slave SW 211 allocates memory space and provides the pointers to the memory space 228 to the HW driver. When data 223

is received by the HW driver 207, the HW driver 207 stores it into memory 209, and notifies the slave SW 225 that the data is available. At an appropriate time, the slave SW reads the data 226 from the memory 209. The timing of the slave SW reading from memory does not gate the following acknowledgement process, therefore does not add any delay in the writing process. The HW driver 207 acknowledges 230 the receipt of the data via the I2C RX 105, the acknowledgement is transmitted to the I2C TX 103 on the master side, which makes it available to the master HW driver 205. The acknowledgement is passed on to the master TX SW for processing.

[0018] FIG. 3 depicts an example where the master reads information from the slave. The slave SW 211 allocates a region of memory 320 and provides the pointer to the memory allocation 322 the slave HW driver 207. When the master SW 201 needs data from the slave SW 211, it sends a request 310 to the master HW driver 205, which in turn, sends the request 312 to the I2C TX 103. The request 314 is transmitted, over the network 150, to the I2C RX 105 of the slave device. The I2C RX 105, makes the request 316 available to the slave HW driver 207 which in turn reads the requested data 317 from the memory 209. The slave SW 211 may optionally be notified that the read has occurred 318. The HW driver 207 makes the requested data 317 available to the slave I2C 105 for transmission 326 to the master I2C 103 via the network 150. The master I2C 103 provides the data 328 to the HW driver 205, which in turn writes it into memory 203. The master SW is notified 332 that the requested data has been put in memory. The master SW 201 reads the data 332 at an appropriate time, without impacting other data transfers between the master and the slave.

[0019] Multiple memory regions may be allocated, each intended for different variables that are used in the higher-level protocol (e.g. HDCP authentication). Once this is done, the HW driver slave 207 responds to transfer requests via the I2C TX side channel 105 in a timely fashion without SW involvement.

[0020] The size of the SW transfer to memory need not be limited to what the underlying communication channel supports. The HW driver segments the transfer as per the channel requirements (e.g. in the case of AUX, which has max 16-byte bursts, the SW issue a single command to transfer 32 bytes into memory and notifies the HW driver. The HW driver does, in this case, two 16-byte bursts, and notify the SW only when both are completed. Channel-specific exceptions, such as, but not limited to, error conditions on the channel can be handled by the HW driver (by re-attempting the transfer until it completes successfully), without any SW involvement.

[0021] Each request (on the slave or master side) has an identifier (e.g. an address) that the HW driver can use to match to the memory region that should be used to write or read the data in the case of a write or read transfer, respectively. If there is no identifier match, the HW driver can wait for the SW to update the allocation(s) until something results in a match or, depending on the channel, it can decline the transfer. As in the transmit case, the SW is notified when data is fully transferred, possibly over many physical transfers (e.g. in the case of AUX).

[0022] If multiple memory allocations are required concurrently, the SW may optionally store a map of these allocations in memory and point HW to the location of this map. Instead of providing the HW driver with a single pointer, SW may provide multiple pointers, each intended for a different variable. The SW creates a lookup table to maintain the memory pointer for a variable and a lookup key that the HW uses to search the table. The use of a lookup table allows the SW to manage the allocations dynamically.

[0023] The term “variable” means any values (including constant values) transferred by a master to a slave (or vice versa) over a side channel.

[0024] The HW driver may collect statistics on successful and unsuccessful transfers. The SW may consult these statistics to decide whether to abort an outstanding command, or perhaps to diagnose interoperability problems.

[0025] Different communication channels may have different particularities that the HW driver can isolate from the SW, for example with respect to failures or maximum transmission size. In I2C (for HDMI), the master may lose arbitration of the physical channel to a different master, or the slave may answer a write with a negative acknowledge (NACK). In both cases, the HW driver can re-attempt the transfer until successful, without involving the SW. Similarly in AUX, the slave can respond negatively to the transfer request (with DEFER or NACK responses). Again, the HW driver can re-attempt until it's successful. The HW driver can implement any channel particularities while SW is agnostic to these and shielded from any additional interrupts required to manage them.

[0026] Optionally, if some variable is relatively small in size and/or expected to be read relatively frequently (e.g. a status variable), it may be kept into a HW register directly accessible by the HW driver without the need for SW to allocate memory and handle pointers. For example, the RxStatus in HDCP for HDMI. Instead of using system memory, the variable can be stored in HW. The HW driver provides dedicated storage space for this variable and provide SW with a mechanism to update the variable's value. The SW is no longer interrupted every time the variable is read. Optionally, the HW driver can inform the

SW about the frequency at which these HW registers are accessed and the SW can access them regularly.

[0027] Although the algorithms described above including those with reference to the foregoing flow charts have been described separately, it should be understood that any two or more of the algorithms disclosed herein can be combined in any combination. Any of the methods, algorithms, implementations, or procedures described herein can include machine-readable instructions for execution by: (a) a processor, (b) a controller, and/or (c) any other suitable processing device. Any algorithm, software, or method disclosed herein can be embodied in software stored on a non-transitory tangible medium such as, for example, a flash memory, a CD-ROM, a floppy disk, a hard drive, a digital versatile disk (DVD), or other memory devices, but persons of ordinary skill in the art will readily appreciate that the entire algorithm and/or parts thereof could alternatively be executed by a device other than a controller and/or embodied in firmware or dedicated hardware in a well known manner (e.g., it may be implemented by an application specific integrated circuit (ASIC), a programmable logic device (PLD), a field programmable logic device (FPLD), discrete logic, etc.). Also, some or all of the machine-readable instructions represented in any flowchart depicted herein can be implemented manually as opposed to automatically by a controller, processor, or similar computing device or machine. Further, although specific algorithms are described with reference to flowcharts depicted herein, persons of ordinary skill in the art will readily appreciate that many other methods of implementing the example machine readable instructions may alternatively be used. For example, the order of execution of the blocks may be changed, and/or some of the blocks described may be changed, eliminated, or combined.

[0028] It should be noted that the algorithms illustrated and discussed herein as having various modules which perform particular functions and interact with one another. It should be understood that these modules are merely segregated based on their function for the sake of description and represent computer hardware and/or executable software code which is stored on a computer-readable medium for execution on appropriate computing hardware. The various functions of the different modules and units can be combined or segregated as hardware and/or software stored on a non-transitory computer-readable medium as above as modules in any manner, and can be used separately or in combination.

[0029] While particular implementations and applications of the present disclosure have been illustrated and described, it is to be understood that the present disclosure is not limited to the precise construction and compositions disclosed herein and that various

modifications, changes, and variations can be apparent from the foregoing descriptions without departing from the spirit and scope of an invention as defined in the appended claims.

WHAT IS CLAIMED IS:

1. A communication system for communicating data between a first software and a second software located on different devices, said system comprising:

first and second hardware drivers associated with said first and second software, respectively,

a first memory coupled with said first software and with said first hardware driver; and

a second memory coupled with said second software and with said second hardware driver;

wherein said first software allocates memory to manage data to be communicated with said second software and provides memory allocation information to said first hardware driver, and said first hardware driver transmits said data to said second hardware driver for delivery to said second software via said second memory.

2. The system of claim 1 wherein said second software allocates memory to manage data to be communicated with said first software and provides memory allocation information to said second hardware driver, and said second hardware driver transmits said data to said first hardware driver for delivery to said first software via said first memory.

3. The system of claim 1 wherein said first software is a master software and said second software is a slave software.

4. The system of claim 1 in which said hardware drivers handle the time-critical components of the communication of data between said first and second software, to reduce the number of interrupts to said first and second software.

5. The system of claim 1 in which the hardware driver that receives communicated data stores said data in the associated memory, and notifies the associated software that said data is available.

6. A method of communicating data between a first software and a second software located on first and second devices, respectively, each of said devices having a hardware driver and memory associated therewith, said method comprising:

allocating memory to manage data to be communicated from said first software to said second software,

providing memory allocation information to said hardware driver associated with said first software,

transmitting said data from said first hardware driver to said second hardware driver for delivery to said second software via said memory associated with said second software.

7. The method of claim 6 which includes allocating memory to manage data to be communicated with said first software, providing memory allocation information to said first hardware driver, and transmitting said data from said second hardware driver to said first hardware driver for delivery to said first software via said first memory.

8. The method of claim 6 wherein said first software is a master software and said second software is a slave software.

9. The method of claim 6 in which time-critical components of the communication of data between said first and second software is handled by said hardware devices, to reduce the number of interrupts to said first and second software.

10. The method of claim 6 in which the hardware driver that receives communicated data stores said data in the associated memory, and notifies the associated software that said data is available.

1/3

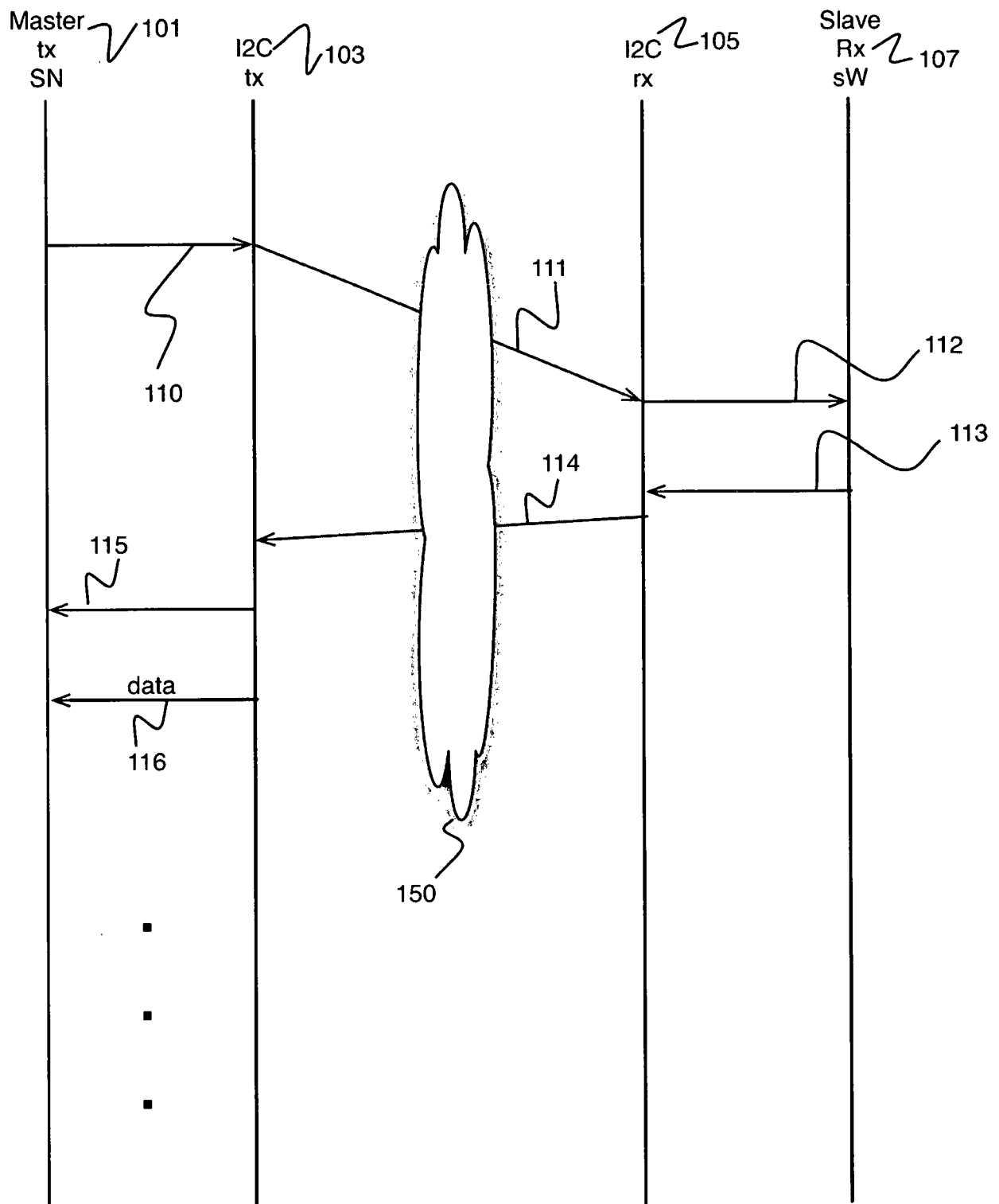


Fig 1
Prior art

2/3

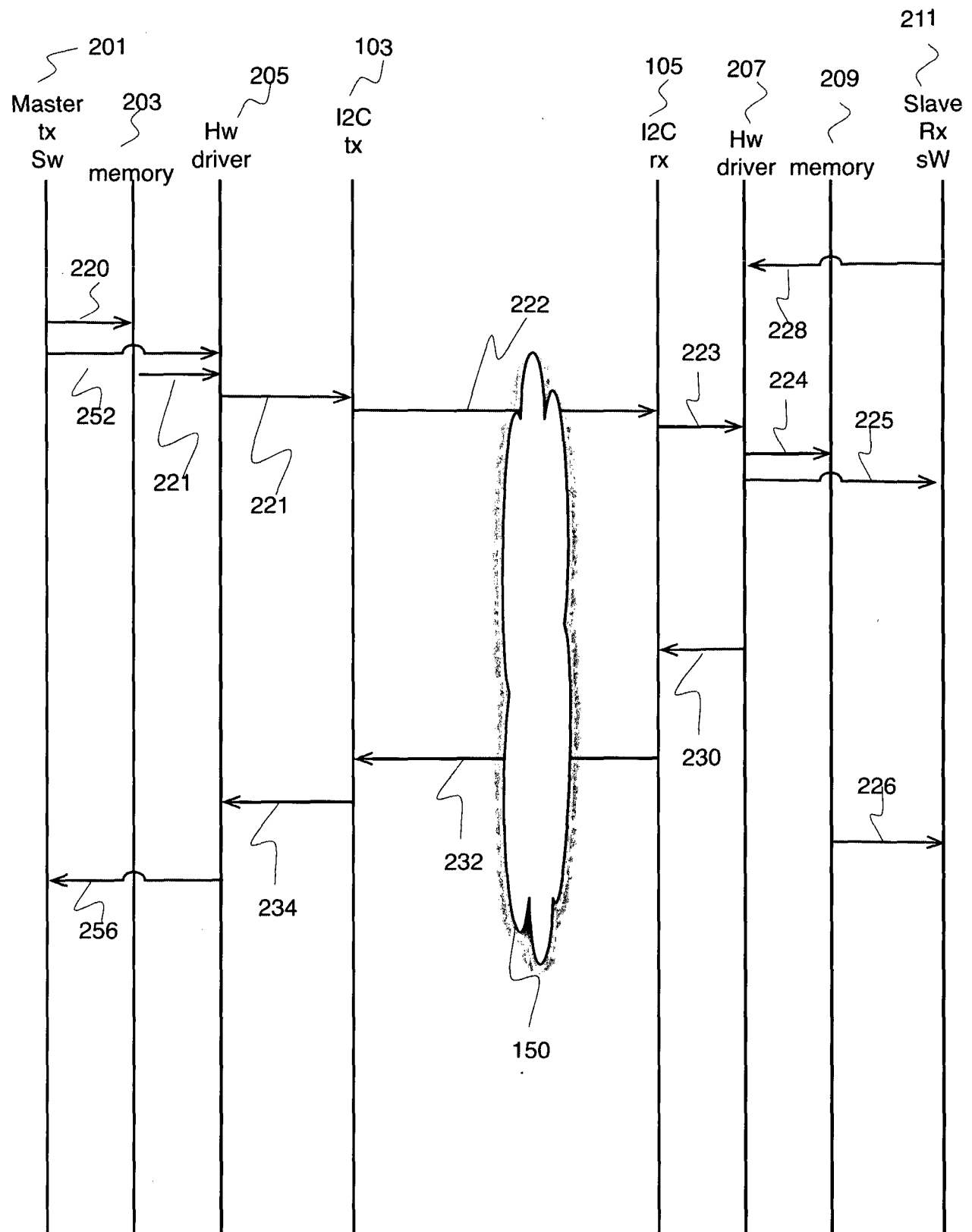


Fig 2

3/3

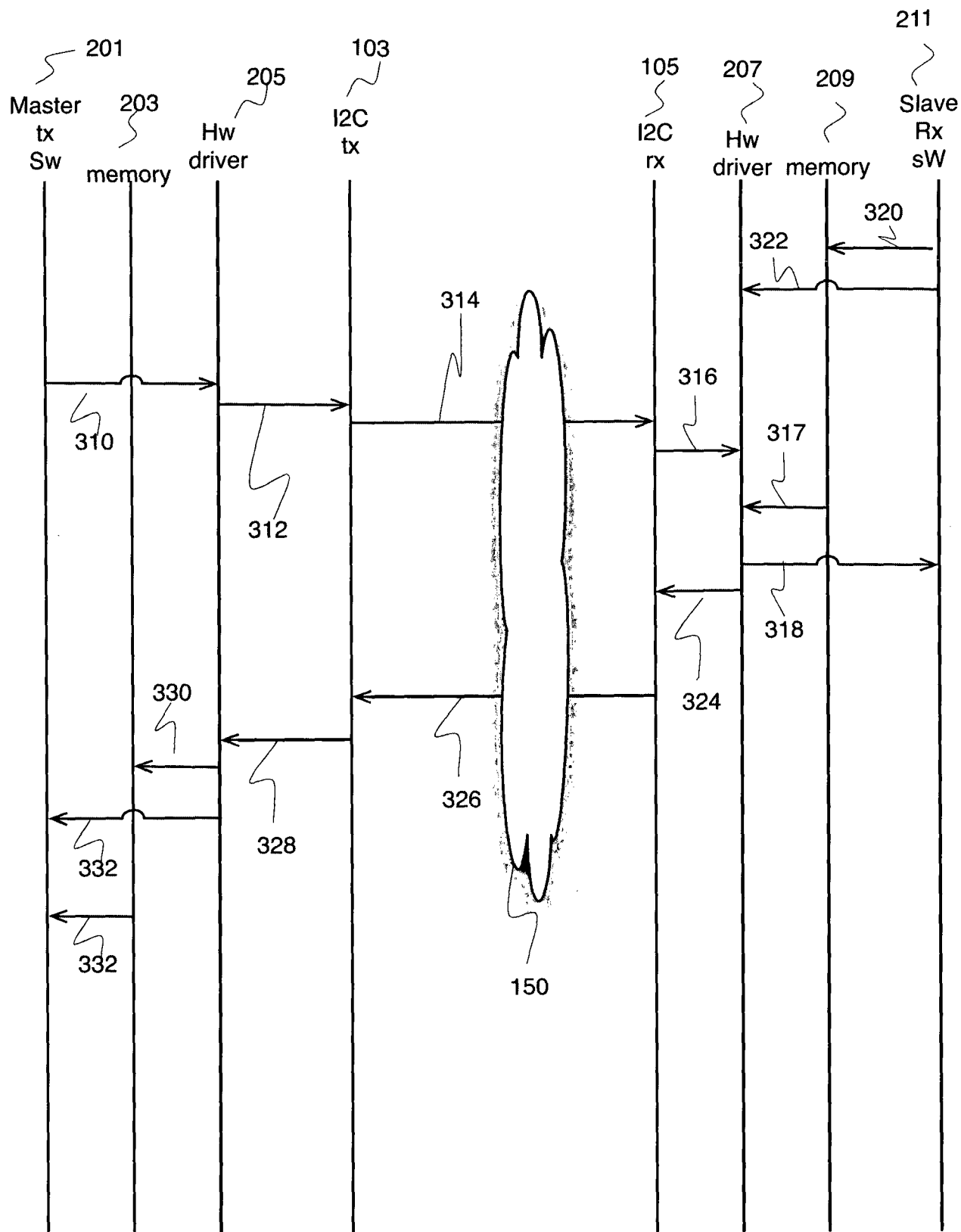


Fig 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2014/002048

A. CLASSIFICATION OF SUBJECT MATTER

IPC: G06F 13/38 (2006.01), G06F 12/02 (2006.01), G06F 13/12 (2006.01), H04N 21/6334 (2011.01), H04N 21/835 (2011.01)

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)

IPC: G06F 13/38 (2006.01), G06F 12/02 (2006.01), G06F 13/12 (2006.01), H04N 21/63 (2011.01), H04N 21/83 (2011.01)

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

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

Database: TotalPatent

Keywords: device; manager; host; virtual; table; communicat!; hdcp; master; slave; driver; memory

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US8732381 B2 (Myrah et al.), 20 May 2014 (20-05-2014)	1-3, 5-8, 10
Y	* Abstract, Fig. 1, claim 1, column 2 line 45-column 7 line 50	4, 9
Y	US6173343 B1 (Delorme), 09 January 2001 (09-01-2001) * column 4 line 53-column 5 line 10	4, 9
A	US7904878 B2 (Kolathur et al.), 08 March 2011 (08-03-2011) * Abstract; column 2 line 53-column 4 line 62	1, 6
A	US8848910 B2 (Mulcahy), 30 September 2014 (30-09-2014) * Abstract; Figure; column 1 line 59-column 2 line 60	1, 6

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search
17 February 2015 (17-02-2015)Date of mailing of the international search report
16 March 2015 (16-03-2015)Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer

Leslie Yeow (819) 934-0345

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/IB2014/002048

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US8732381 B2	20 May 2014 (20-05-2014)	US2013117767A1	09 May 2013 (09-05-2013)
US6173343 B1	09 January 2001 (09-01-2001)	US6173343B1 EP0903670A1	09 January 2001 (09-01-2001) 24 March 1999 (24-03-1999)
US7904878 B2	08 March 2011 (08-03-2011)	US2008155572A1	26 June 2008 (26-06-2008)
US8848910 B2	30 September 2014 (30-09-2014)	US2010260336A1	14 October 2010 (14-10-2010)