Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2934353 A1 2015/06/25

(21) 2 934 353

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) **Date de dépôt PCT/PCT Filing Date:** 2014/12/19
(87) **Date publication PCT/PCT Publication Date:** 2015/06/25
(85) **Entrée phase nationale/National Entry:** 2016/06/16
(86) **N° demande PCT/PCT Application No.:** US 2014/071726
(87) **N° publication PCT/PCT Publication No.:** 2015/095827
(30) **Priorité/Priority:** 2013/12/20 (US14/136,721)

(51) **Cl.Int./Int.Cl.** ***H04N 21/6587*** (2011.01), ***H04N 5/76*** (2006.01), ***H04N 5/781*** (2006.01)

(71) **Demandeur/Applicant:**
OPENTV, INC., US

(72) **Inventeurs/Inventors:**
WILLARD, PIERRE, US;
GOGOI, AMARENDRA N., US;
KASI, CHANDRIKA, US

(74) **Agent:** RIDOUT & MAYBEE LLP

(54) **Titre : ENREGISTREMENT DE PLUSIEURS VUES**
(54) **Title: MULTIPLE VIEWS RECORDING**

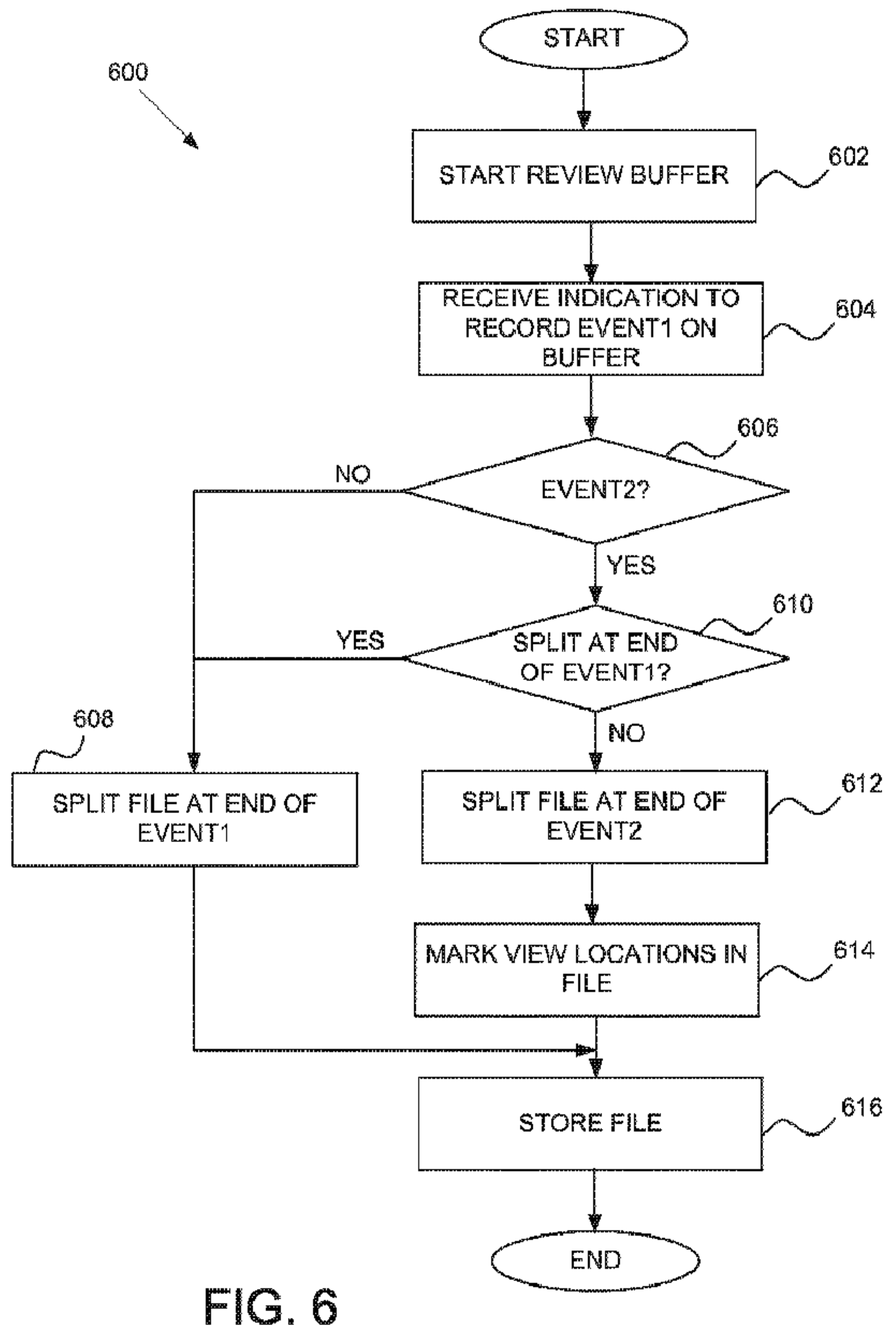


FIG. 6

(57) **Abrégé/Abstract:**
A system and method for providing multiple views recording is provided. In example embodiments, an indication to record serial events including a first event and a second event on a same channel is received. A recording stream is started at a beginning of the

CA 2934353 A1 2015/06/25

(21) **2 934 353**

(13) **A1**

(57) **Abrégé(suite)/Abstract(continued):**
first event and ends at an end of the second event. View markings are inserted in the recording stream that differentiate a first view corresponding to the first event from a second view corresponding to the second event. Thus, the recording stream creates a single file that contains multiple views. The single file is stored for later playback.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 June 2015 (25.06.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/095827 A3

(51) **International Patent Classification**:
H04N 21/6587 (2011.01) *H04N 5/781* (2006.01)
H04N 5/76 (2006.01)

(21) **International Application Number**:
PCT/US2014/071726

(22) **International Filing Date**:
19 December 2014 (19.12.2014)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
14/136,721 20 December 2013 (20.12.2013) US

(63) **Related by continuation (CON) or continuation-in-part (CIP) to earlier application**:
US 14/136,721 (CON)
Filed on 20 December 2013 (20.12.2013)

(71) **Applicant**: **OPENTV, INC.** [US/US]; 275 Sacramento Street, San Francisco, California 94111 (US).

(72) **Inventors**: **WILLARD, Pierre**; 275 Sacramento Street, San Francisco, California 94111 (US). **GOGOI, Amarendra N.**; 275 Sacramento Street, San Francisco, California 94111 (US). **KASI, Chandrika**; 275 Sacramento Street, San Francisco, California 94111 (US).

(74) **Agents**: **SCHEER, Bradley W.** et al.; Schwegman, Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (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, 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,

[Continued on next page]

(54) **Title**: MULTIPLE VIEWS RECORDING

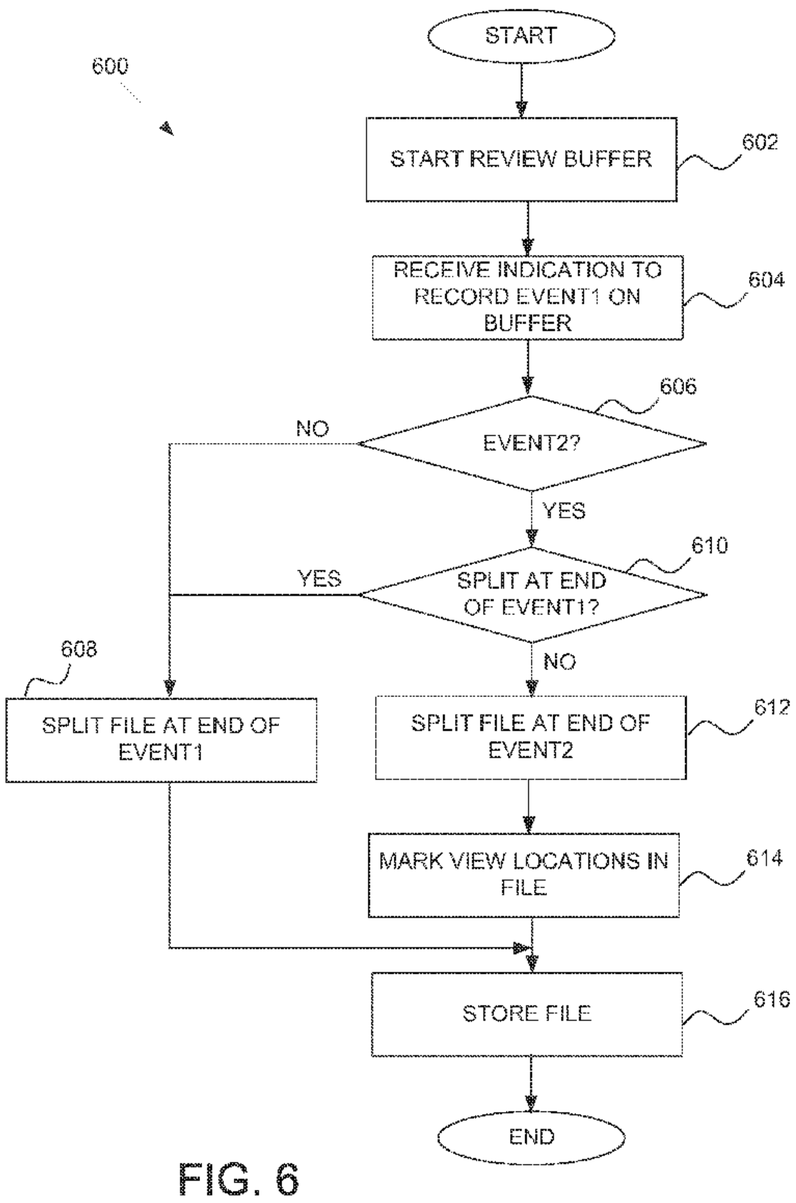


FIG. 6

(57) **Abstract**: A system and method for providing multiple views recording is provided. In example embodiments, an indication to record serial events including a first event and a second event on a same channel is received. A recording stream is started at a beginning of the first event and ends at an end of the second event. View markings are inserted in the recording stream that differentiate a first view corresponding to the first event from a second view corresponding to the second event. Thus, the recording stream creates a single file that contains multiple views. The single file is stored for later playback.

WO 2015/095827 A3

WO 2015/095827 A3

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(88) Date of publication of the international search report: 5 November 2015

Published:

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

MULTIPLE VIEWS RECORDING

RELATED APPLICATION

[0000] This application claims the priority benefit of U.S. Patent Application No. 14/136,721, filed December 20, 2013, which is incorporated herein by reference in its entirety.

FIELD

[0001] The present disclosure relates generally to providing video content recording, and in a specific example embodiment, to providing multiple views recording.

BACKGROUND

[0002] Conventionally, when a user desires to record two events (e.g., television shows) back-to-back, two different files are created – one for the first event and one for the second event. Because it is desirable to add padding to a beginning and end of a scheduled event, the two different files contain overlapping portions. Additionally, two tuners may be needed in order to record two back-to-back events with appropriate padding since the padding at the end of the first event overlaps with the padding at the beginning of the second event.

BRIEF DESCRIPTION OF DRAWINGS

[0003] Various ones of the appended drawings merely illustrate example embodiments of the present invention and cannot be considered as limiting its scope.

[0004] FIGS. 1A-C are diagrams illustrating a comparison of conventional recording versus multiple views recording.

[0005] FIGS. 2A-C are diagrams illustrating an example generation of a file from a review buffer.

[0006] FIGS. 3A-C are diagrams illustrating another example generation of a file from a review buffer.

[0007] FIG. 4 is a block diagram illustrating an example embodiment of a set top box for providing multiple views recording.

[0008] FIG. 5 is a flow diagram of an example method for generating a multiple views recording.

[0009] FIG. 6 is a flow diagram of an example method for generating a recording from a review buffer.

[0010] FIG. 7 is a flow diagram of an example method for presenting a view from a stored file that contains multiple views.

[0011] FIG. 8 is a simplified block diagram of a machine in an example form of a computing system within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein may be executed.

DETAILED DESCRIPTION

[0012] The description that follows includes systems, methods, techniques, instruction sequences, and computing machine program products that embody illustrative embodiments of the present invention. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide an understanding of various embodiments of the inventive subject matter. It will be evident, however, to those skilled in the art that embodiments of the inventive subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures, and techniques have not been shown in detail.

[0013] Example embodiments described herein provide systems and methods for providing multiple views recording. In example embodiments, an indication to record serial events including a first event and a second event on a same channel is received. A recording stream is started at a beginning of the first event and ends at an end of the second event. In example embodiments, the recording stream may be made up of several elementary streams. For example the elementary streams may include a video elementary stream, one or more audio elementary stream (e.g., one audio elementary stream per language), and other types of elementary streams. View markings are inserted in the recording stream that differentiate a first view corresponding to the first event from a second view corresponding to the second event. The view markings may include padding added to the first and second events. Thus, the recording stream creates a single file that contains multiple views. The single file is stored for later playback.

[0014] In a further embodiment, an indication to record one or more events contained within a review buffer is received. A point to split a file containing at least one event of the one or more events from the review buffer is determined. The point may correspond to an end (with padding) of one of the one or more events. The file that contains the at least one

event from the review buffer is split based on the determination. The file is stored into a file storage for later playback.

[0015] As such, some example embodiments are directed to back-to-back or serial recording of two events (e.g., two television shows or other programming content) on a same channel by a set-top box (STB). Typically when recording an event, padding is desired in case the event starts earlier or ends later than scheduled (e.g., in an electronic program guide; "EPG"). For example, the recording of the event may begin a predetermined pad time (e.g., a front pad time of 1-5 minutes) before the scheduled event and/or end a second predetermined pad time (e.g., a rear pad time of 1-5 minutes) after the scheduled event ends.

[0016] FIGS. 1A-C are diagrams illustrating a comparison of conventional recording versus an example multiple views recording embodiment. With padding, recording streams of the two serial events will overlap. That is padding at the end of a recording stream for the first event will overlap padding at a beginning of a recording stream of the second event. This may cause a conflict in the system. Accordingly, overlapping padding in the two recording streams may not be recorded if only a single tuner is available. FIG. 1A shows a conventional single tuner recording process. In the example of FIG. 1A, a single tuner (tuner1) may add padding to the beginning of a recording stream of event1, but refrain from adding padding to the end of the recording stream of event1 since tuner1 must, instead start recording event2. However, tuner1 may add padding to the end of the recording stream of event2. The result of the recording process in FIG. 1A is two files with incomplete padding (e.g., no padding at the end of the first recording stream or beginning of the second recording stream) that are stored separately.

[0017] However, if two tuners are available for recording, then, as shown in FIG. 1B, the first turner (tuner1) may record event1 with sufficient padding at the beginning and end of the recording stream of event1. Similarly, a second tuner (tuner2) may record event2 with sufficient padding at the beginning and end of the recording stream of event2. Alternatively, recording of event1 and event2 by a single tuner (tuner1) may be possible in limited cases when shared between the two events. In these cases, the overlapping padding may be duplicated (e.g., copied) by tuner1 in the two recording streams. However, using a single tuner1 to generate two different recording streams may not be supported by hardware in the STB. Additionally, the hardware may have limitations (e.g., max N streams). Furthermore, the result of the recording process of FIG. 1B is still two files with a

duplicated portion in both (e.g., the overlapping padding). Therefore, even though the recording streams are from the same channel, in order to record both streams with sufficient padding, more resources are needed. Such resources may include tuners, demultiplexers, storage space, and CPU processing power (e.g., to copy portions of the recording stream).

[0018] As such, example embodiments provide multiple views recording as shown in the example of FIG. 1C. A same tuner (tuner1) may continuously record a single stream that includes multiple events. The recording of an event (or multiple events) inside a stream that starts and ends at a specific time with appropriate padding may be referred to as a "view." As such, a resulting file of the recording stream may comprise one or more views (e.g., multiple views) created using less resources. The resulting file may contain view markings that indicate the beginning and end of each event along with the appropriate padding (e.g., mark the views). During playback of an event, the STB will know where the event starts and ends in the single file (including padding) based on the view markings. It is noted that while FIG. 1C illustrates two overlapping views in the recording stream, any number of overlapping views may be present in the recording stream (e.g., views for event1, event 2, event 3, and so on).

[0019] FIGS. 2A-C are diagrams illustrating an example generation of a file from a review buffer. The review buffer is a temporary recording of what the user is currently watching that provides views of past events for a predetermined amount of time (e.g., 2 hours). Typically, the review buffer records, in the background, for the predetermined amount of time (e.g., 2 hours) events on a same channel that the user is watching. Using the review buffer, the user can pause, rewind, and re-view portions of a current or past event. For example, the user can rewind "live" television for two hours (or the length of time permitted by the review buffer). Referring to FIG. 2A, a review buffer 202 is shown. The review buffer 202 provides views of event1, event2, and event3. A front end 204 of the review buffer 202 may record what the user is currently, in real-time, viewing. Assuming that the review buffer 202 is 2 hours long, event1 and event 2 may be half hour events while event3 and event4 may be full hour events.

[0020] FIG. 2B illustrates the review buffer 202 approximately a half hour later than that shown in FIG. 2A. A recording stream of the review buffer 202 has dropped the recording of event1 and, instead, is recording a portion of event4. Anytime during a period of the review buffer (e.g., within a 2 hour window of the review buffer 202), a user may indicate a desire to record, for later playback, an event in a portion of the review buffer 202. In the

example shown in FIG. 2B, the user has indicated a desire to record event2 (as indicated by shading). Even though event2 occurred roughly 1.5 hours in the past, because a recording or view of event2 still exists in the review buffer 202, the STB can create a file that includes the view of event2 from the review buffer 202.

[0021] Referring now to FIG. 2C, the review buffer 202 has advanced in time to record portions of event3, event4, and event5. Since the review buffer 202 has advanced past an end of event2, the recording of event2 may be split or detached (shown by split lines 206) from the review buffer 202 (with sufficient padding). The recording of event2 may then be stored as a file for later viewing. It is noted that the file containing event2 is not a copy of the review buffer, but instead, is just a saved portion of the review buffer 202. Because copying does not need to be performed, resources may be saved (e.g., CPU processing power).

[0022] FIGS. 3A-C are diagrams illustrating another example generation of a recording from a review buffer 302. While the example shown in FIGS. 2 recorded a single event for later playback from the review buffer 202, the example of FIGS. 3 is directed to recording multiple events from the review buffer 302. For example, FIG. 3A illustrates that a user has indicated a desire to record event1 and event3 for later playback.

[0023] In one embodiment, the STB may split the file for each desired event once the review buffer 302 has advanced past the end of the event (with sufficient padding). As shown in FIG. 3B, a first file may be created from the review buffer 302 at the end of event1 and a second file may be created from the review buffer 302 at the end of event3. The two files are distinguished by the split lines 304. It is noted that the files may contain the appropriate padding at the beginning and end of the event.

[0024] In an alternative embodiment, the STB may split the file for the two desired events at the end of the second desired event as shown in FIG. 3C. As such, a single file as indicated by the split lines 304 may be created from the review buffer 302 that includes a view for event1 and a view for event3. The view for event1 and event3 may be differentiated in the file by view markings 306 to indicate an end or begin (with appropriate padding) for each event. In a further embodiment, a view of event2 may be removed from the file to condense the file and reduce storage requirements.

[0025] It is noted that for the examples discussed in FIGS. 2 and 3, the view markings 306 may be inserted into the review buffer 202 or 302 while the review buffer 202 or 302 still retains the events being recorded for later playback. These view markings 306 may be

used by the STB to split or detach the recording to create the file(s). Because copying does not need to be performed, resources may be saved (e.g., CPU processing power).

[0026] In example embodiments, an event may be defined by the EPG by a start time and end time on a specific channel. As such, the user may indicate a triplet of information (e.g., start time, end time, and channel information) in order to record the event. For example, the user may indicate to record channel X from 9pm to 11pm (plus possible padding). The event may then comprise programming content from 9pm to 11pm on channel X. The use may then indicate to record channel X from 8pm to 10pm. In this case, the event may comprise programming content from 8pm to 10pm on channel X. In this case, the views within the recording stream (of the events) overlap for 1 hour of programming content (plus padding). Therefore, overlaps in views may include more than just overlapping padding (e.g., at least overlapping padding, but may include overlaps in part of the event programming content).

[0027] FIG. 4 is a block diagram illustrating an example embodiment of a set top box (STB) 400 for providing multiple views recording. The STB 400 generates files with multiple views or generates a file from a review buffer (e.g., separating at least one view from the views of the review buffer) with reduction in use of resources. Accordingly, the STB 400 may comprise a recording module 402, at least one tuner 404, a view module 406, and file storage 408 all communicatively coupled together. It is noted that other components not necessary to describe example embodiments may also be included in the STB 400 (e.g., decoders, demultiplexers, processors, modulators).

[0028] The recording module 402 manages the recording of streams by each tuner 404 including review buffers. In example embodiments, the recording module 402 may receive an indication of one or more events that a user wants to record. The recording module 402 may instruct the tuner 404 to record the one or more events using a single recording stream that results in a single file. If multiple views (e.g., recordings of multiple events) are within the single file, the recording module 402 may insert view markings at a beginning and end for each desired event that may include appropriate padding. The generated file is then stored to the file storage 408 for later playback.

[0029] Additionally, the recording module 402 may split or detach a file containing a desired event from a review buffer. By splitting the file from the review buffer, instead of copying a portion of the review buffer, the recording module 402 reduces use of resources (e.g., CPU processing power required to copy and store large bites of data). The generated

file is stored to the file storage 408 for later playback. It is noted that while example embodiments describe the tuner 404 as recording the events, in alternative embodiments, a separate component of the STB 400 may perform the recording operation.

[0030] The view module 406 manages the playback of views from stored files. Accordingly, the view module 406 may receive a request from a user to playback a particular event. The view module 406 accesses the file storage 408 to obtain the corresponding file that contains the event. Assuming the file contains more than one event, and as such contains multiple views, the view module 406 determines locations of the view markings that correspond to the requested event. The view module 406 then causes presentation of the view (and only of that view), as indicated by the view markings, for the event. The view may include padding at a beginning and/or end of the event.

[0031] It is noted that while the file storage 408 is shown to be a part of the STB 400, in alternative embodiments, the file storage 408 may be located elsewhere. For example, the file storage 408 may be located in a "cloud" or be at a service provider location.

[0032] Referring now to FIG. 5, a flow diagram of an example method 500 for generating a multiple views recording is shown. The operations of the method 500 may be performed by components of the STB 400. In operation 502, an indication to record serial events is received. In example embodiments, the recording module 402 may receive a request from a user to record at least two events that are back-to-back on a same channel.

[0033] In operation 504, the STB 400 starts to record a first event (event1). The recording may be performed by the tuner 404. The corresponding recording stream may include a front padding added to the beginning of event1. The tuner 404 may continue the recording stream through the end of event1 and beginning of a second event (event2). At operation 506, recording may stop at an end of event2. In example embodiments, padding may be added to the end of event2 in the recording stream.

[0034] In operation 508, views corresponding to event1 and event2 may be marked with view markings to indicate where a view of event 1 starts and ends and where a view of event2 starts and end in a file created from the recording stream. The view markings may be inserted by, for example, the recording module 402 into the recording stream that constitutes the file. It is noted that while the method 500 indicates that the view markings are inserted after the recording is completed, example embodiments may insert the view markings on-the-fly during the recording of the events. The file with appropriate view markings is stored to the file storage 408 for later playback in operation 510.

[0035] FIG. 6 is a flow diagram of an example method 600 for generating a recording from a review buffer. The review buffer is a temporary recording of what the user is watching that provides views of past events for a predetermined amount of time (e.g., 2 hours). In operation 602, a review buffer starts to record. Typically, the review buffer starts to record as soon as a user starts to view an event and continues to record (typically on a same channel) until the user stops viewing. A length of the review buffer may correspond to a predetermined length of time (e.g., 2 hours).

[0036] In operation 604, an indication to record a first event (event1) is received. Accordingly, the recording module 402 may receive the indication or request to record event1. As long as event1 is within a window of the review buffer (e.g., within the 2 hour length of time of the review buffer), the recording module 402 may trigger the recording of event1.

[0037] A determination may be made at operation 606 as to whether a second event (event2) is also indicated to be recorded. If no event2 is indicated to be recorded, then at operation 608, a file that contains a view of event1 may be split off or detached from the review buffer. In example embodiments, the file is split off, with appropriate padding, at a point after the review buffer advances beyond an end of event1.

[0038] If, however, event2 is also requested to be recorded in operation 606, then a further determination may be made at operation 610 as to whether to split a file at an end of event1 or event2. For example, if there is a huge gap in time between event1 and event2 (e.g., 1.5 hours of another event in between), the recording module 402 may determine that it would be more efficient to split a first file after the end of event1 and create a second file (by splitting a second file off of the review buffer) that only contains event2. Accordingly, if at operation 610, it is determined to split the file at the end of event1, then the method 500 proceeds to operation 608 where the file is split from the review buffer at a point after the end of event1 with appropriate padding.

[0039] However, if the determination at operation 610 is to not split the file at the end of event1, then at operation 612, the file may be split from the review buffer at a point after an end of event2. Views that correspond to event1 and event2 (with appropriate padding if required) may be marked in the file in operation 614 by inserting view markings. It is noted that, in various embodiments, the view markings may be inserted (e.g., by the recording module 402) at any time before or after the splitting of the file from the review buffer. The file is stored to the file storage 408 in operation 616.

[0040] FIG. 7 is a flow diagram of an example method 700 for presenting a view from a stored file that contains multiple views. The operations of the method 700 may be performed by the view module 406. In operation 702, an indication to view a recorded event is received. For example, the view module 406 may receive a request to view the recorded event. In operation 704, a file that corresponds to the recorded file is accessed from the file storage 408.

[0041] In operation 706, view markings of a view that corresponds to the recorded event are determined. Because the file contains more than one recorded event, the file is marked with viewing markings that indicate where each view in the file begins and ends (sometimes with appropriate padding). The view module 406 determines or identifies the view using these view markings. In operation 708, the view is presented to the user by the view module 406.

[0042] Example embodiments allow multiple events on a same channel to be recorded in a single file using a single tuner. As such, there is no need for hardware to shared one tuner input with several output files. Additionally, any number of views may be created and stored within a single stream or file; there is no hardware limitation. By recording within a single file or splitting a file from a recording buffer, there is reduced stress on CPU and hardware since there is no need to record/duplicate overlapping segments (e.g., padding) on the same channel nor is there a need to copy large amounts of data to create a new file. The elimination of overlapping recordings also reduces storage space requirements. Accordingly, one or more of the methodologies discussed herein may provide the technical effect of reducing computing resources used by one or more devices within the system. Examples of such computing resources include, without limitation, processor cycles, memory usage, storage space, and power consumption.

[0043] FIG. 8 is a block diagram illustrating components of a machine 800, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. 8 shows a diagrammatic representation of the machine 800 in the example form of a computer system and within which instructions 824 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 800 to perform any one or more of the methodologies discussed herein may be executed. In alternative embodiments, the machine 800 operates as a standalone device or may be connected (e.g., networked) to other machines. In a networked

deployment, the machine 800 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 800 may be a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), a cellular telephone, a smartphone, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions 824, sequentially or otherwise, that specify actions to be taken by that machine. Further, while only a single machine is illustrated, the term "machine" shall also be taken to include a collection of machines that individually or jointly execute the instructions 824 to perform any one or more of the methodologies discussed herein.

[0044] The machine 800 includes a processor 802 (e.g., a central processing unit (CPU), a graphics processing unit (GPU), a digital signal processor (DSP), an application specific integrated circuit (ASIC), a radio-frequency integrated circuit (RFIC), or any suitable combination thereof), a main memory 804, and a static memory 806, which are configured to communicate with each other via a bus 808. The machine 800 may further include a graphics display 810 (e.g., a plasma display panel (PDP), a light emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)). The machine 800 may also include an alpha-numeric input device 812 (e.g., a keyboard), a cursor control device 814 (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, remote control device, or other pointing instrument), a storage unit 816, a signal generation device 818 (e.g., a speaker), and a network interface device 820.

[0045] The storage unit 816 includes a machine-readable medium 822 on which is stored the instructions 824 embodying any one or more of the methodologies or functions described herein. The instructions 824 may also reside, completely or at least partially, within the main memory 804, within the processor 802 (e.g., within the processor's cache memory), or both, during execution thereof by the machine 800. Accordingly, the main memory 804 and the processor 802 may be considered as machine-readable media. The instructions 824 may be transmitted or received over a network 826 via the network interface device 820.

[0046] As used herein, the term "memory" refers to a machine-readable medium able to store data temporarily or permanently and may be taken to include, but not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, and cache memory. While the machine-readable medium 822 is shown in an example

embodiment to be a single medium, the term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store instructions. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions for execution by a machine (e.g., machine 800), such that the instructions (e.g., instructions 824), when executed by one or more processors of the machine (e.g., processor 802), cause the machine to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable medium" shall accordingly be taken to include, but not be limited to, one or more data repositories in the form of a solid-state memory, an optical medium, a magnetic medium, or any suitable combination thereof.

[0047] Furthermore, the tangible machine-readable medium is non-transitory in that it does not embody a propagating signal. However, labeling the tangible machine-readable medium as "non-transitory" should not be construed to mean that the medium is incapable of movement – the medium should be considered as being transportable from one physical location to another. Additionally, since the machine-readable medium is tangible, the medium may be considered to be a machine-readable device.

[0048] The instructions 824 may further be transmitted or received over a communications network 826 using a transmission medium via the network interface device 820 and utilizing any one of a number of well-known transfer protocols (e.g., HTTP). Examples of communication networks include a local area network (LAN), a wide area network (WAN), the Internet, mobile telephone networks, POTS networks, and wireless data networks (e.g., WiFi and WiMAX networks). The term "transmission medium" shall be taken to include any intangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine 800, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software.

[0049] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the

operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[0050] Certain embodiments are described herein as including logic or a number of components, modules, or mechanisms. Modules may constitute either software modules (e.g., code embodied on a machine-readable medium or in a transmission signal) or hardware modules. A "hardware module" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware modules of a computer system (e.g., a processor or a group of processors) may be configured by software (e.g., an application or application portion) as a hardware module that operates to perform certain operations as described herein.

[0051] In some embodiments, a hardware module may be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware module may include dedicated circuitry or logic that is permanently configured to perform certain operations. For example, a hardware module may be a special-purpose processor, such as a field programmable gate array (FPGA) or an ASIC. A hardware module may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware module may include software encompassed within a general-purpose processor or other programmable processor. It will be appreciated that the decision to implement a hardware module mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software) may be driven by cost and time considerations.

[0052] Accordingly, the phrase "hardware module" should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. As used herein, "hardware-implemented module" refers to a hardware module. Considering embodiments in which hardware modules are temporarily configured (e.g., programmed), each of the hardware modules need

not be configured or instantiated at any one instance in time. For example, where a hardware module comprises a general-purpose processor configured by software to become a special-purpose processor, the general-purpose processor may be configured as respectively different special-purpose processors (e.g., comprising different hardware modules) at different times. Software may accordingly configure a processor, for example, to constitute a particular hardware module at one instance of time and to constitute a different hardware module at a different instance of time.

[0053] Hardware modules may provide information to, and receive information from, other hardware modules. Accordingly, the described hardware modules may be regarded as being communicatively coupled. Where multiple hardware modules exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware modules. In embodiments in which multiple hardware modules are configured or instantiated at different times, communications between such hardware modules may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware modules have access. For example, one hardware module may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware module may then, at a later time, access the memory device to retrieve and process the stored output. Hardware modules may also initiate communications with input or output devices, and may operate on a resource (e.g., a collection of information).

[0054] The various operations of example methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented modules that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented module" refers to a hardware module implemented using one or more processors.

[0055] Similarly, the methods described herein may be at least partially processor-implemented, a processor being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented modules. Moreover, the one or more processors may also operate to support performance of the relevant operations in a "cloud computing" environment or as a

"software as a service" (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an application program interface (API)).

[0056] The performance of certain of the operations may be distributed among the one or more processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the one or more processors or processor-implemented modules may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the one or more processors or processor-implemented modules may be distributed across a number of geographic locations.

[0057] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of embodiments of the present invention. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single invention or inventive concept if more than one is, in fact, disclosed.

[0058] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[0059] As used herein, the term "or" may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present invention. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined

structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present invention as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method comprising:

receiving an indication to record serial events including a first event and a second event on a same channel;

starting a recording stream at a beginning of the first event;

inserting, using a hardware processor, view markings in the recording stream that differentiate a first view corresponding to the first event from a second view corresponding to the second event;

stopping the recording stream at an end of the second event, the recording stream creating a single file that contains multiple views; and

storing the single file for later playback.

2. The method of claim 1, wherein inserting the view markings comprises:

adding padding to a beginning of the first event and padding to an end of the first event in the recording stream; and

adding padding to a beginning of the second event and padding to an end of the second event in the recording stream.

3. The method of claim 2, wherein at least the padding added to the end of the first event of the first view and the padding added to the beginning of the second event of the second view overlap.

4. The method of claim 1, further comprising:

receiving a request to view the first event;

accessing the single file that contains the first event;

determining the view markings in the recording stream that differentiate the first view corresponding to the first event; and

causing presentation of the first view to the user.

5. The method of claim 1, wherein the inserting of the view markings occurs on-the-fly during a recording of the recording stream.

6. A method comprising:

receiving an indication to record one or more events contained within a review buffer;

determining a point to split a file containing at least one event of the one or more events from the review buffer;

splitting, using a hardware processor, the file that contains the at least one event from the review buffer based on the determining; and

storing the file that contains the at least one event for later playback into a file storage.

7. The method of claim 6, further comprising:

inserting view markings at a beginning and end of the one or more events in the review buffer; and

using at least one of the view markings in the determining of the point to split the file from the review buffer.

8. The method of claim 7, wherein the view markings include padding to a beginning and end of each of the one or more events in the review buffer.

9. The method of claim 6, further comprising inserting view markings into the file to differentiate each of the at least one event in the file.

10. The method of claim 9, wherein the view markings include padding to a beginning and end of each of the at least one event in the file.

11. A system comprising:

a hardware processor;

a recording module to receive an indication to record serial events including a first event and a second event on a same channel; and

a tuner to start a recording stream at a beginning of the first event, and stop the recording stream at an end of the second event;

the recording module further to insert, using the hardware processor, view markings in the recording stream that differentiate a first view corresponding to the first event from a

second view corresponding to the second event, the recording stream creating a single file that contains multiple views, and to store the single file for later playback.

12. The system of claim 11, wherein the recording module inserts the view markings by:

adding padding to a beginning of the first event and padding to an end of the first event in the recording stream; and

adding padding to a beginning of the second event and padding to an end of the second event in the recording stream.

13. The system of claim 12, wherein at least the padding added to the end of the first event of the first view and the padding added to the beginning of the second event of the second view overlap.

14. The system of claim 11, further comprising a view module to:

receive a request to view the first event;

access the single file that contains the first event;

determine the view markings in the recording stream that differentiate the first view corresponding to the first event; and

cause presentation of the first view to the user.

15. The system of claim 11, wherein the recording module is to insert the view markings on-the-fly during a recording of the recording stream.

16. A non-transitory machine-readable storage medium in communication with at least one processor, the non-transitory machine-readable storage medium storing instructions which, when executed by the at least one processor of a machine, cause the machine to perform operations comprising:

receiving an indication to record one or more events contained within a review buffer;

determining a point to split a file containing at least one event of the one or more events from the review buffer;

splitting the file that contains the at least one event from the review buffer based on the determining; and

storing the file that contains the at least one event for later playback into a file storage.

17. The non-transitory machine-readable storage medium of claim 16, wherein the operations further comprise:

inserting view markings at a beginning and end of the one or more events in the review buffer; and

using at least one of the view markings in the determining of the point to split the file from the review buffer.

18. The non-transitory machine-readable storage medium of claim 17, wherein the view markings include padding to a beginning and end of each of the one or more events in the review buffer.

19. The non-transitory machine-readable storage medium of claim 16, wherein the operations further comprise inserting view markings into the file to differentiate each of the at least one event in the file.

20. The non-transitory machine-readable storage medium of claim 19, wherein the view markings include padding to a beginning and end of each of the at least one event in the file.

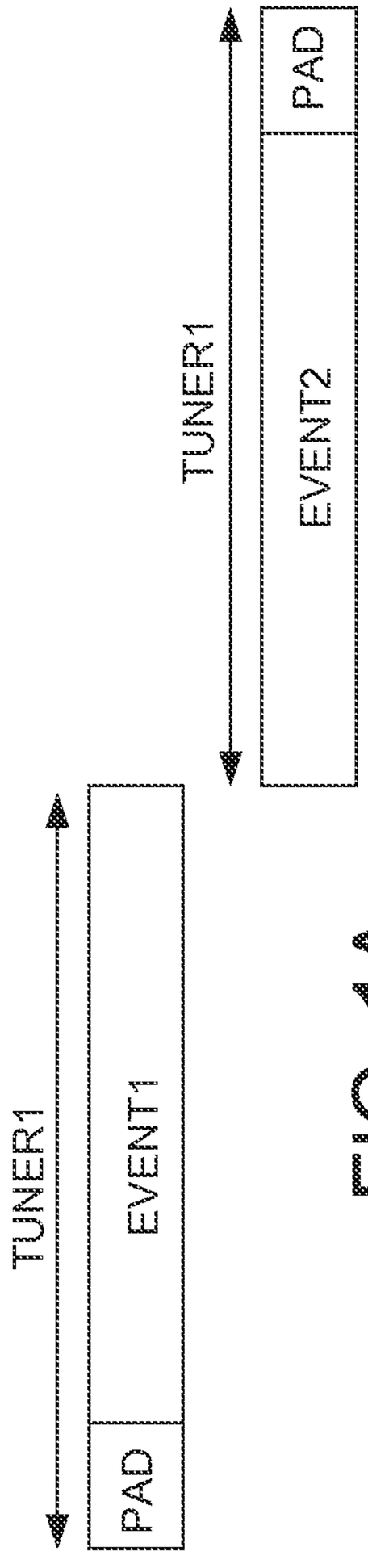


FIG. 1A

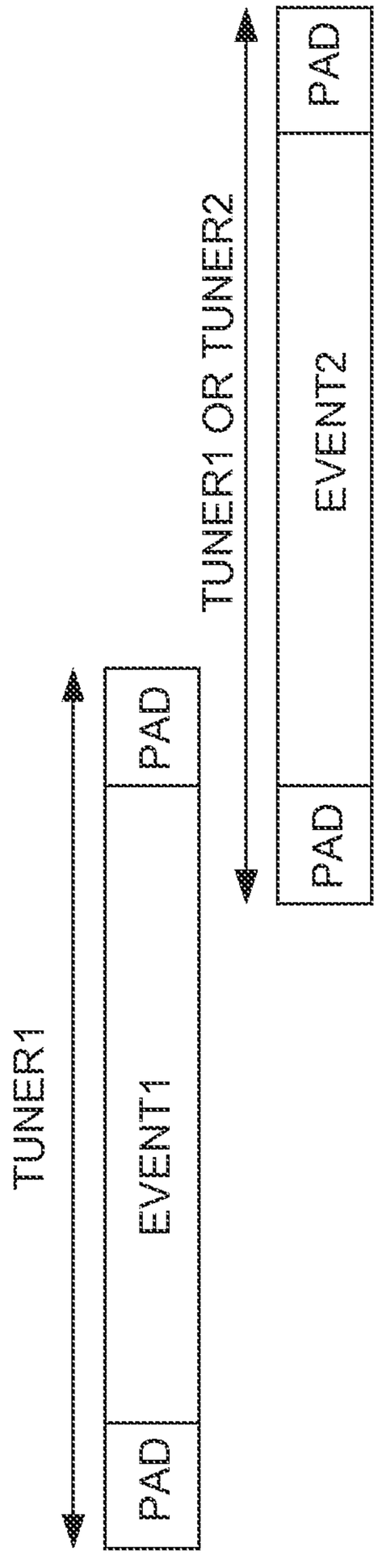


FIG. 1B

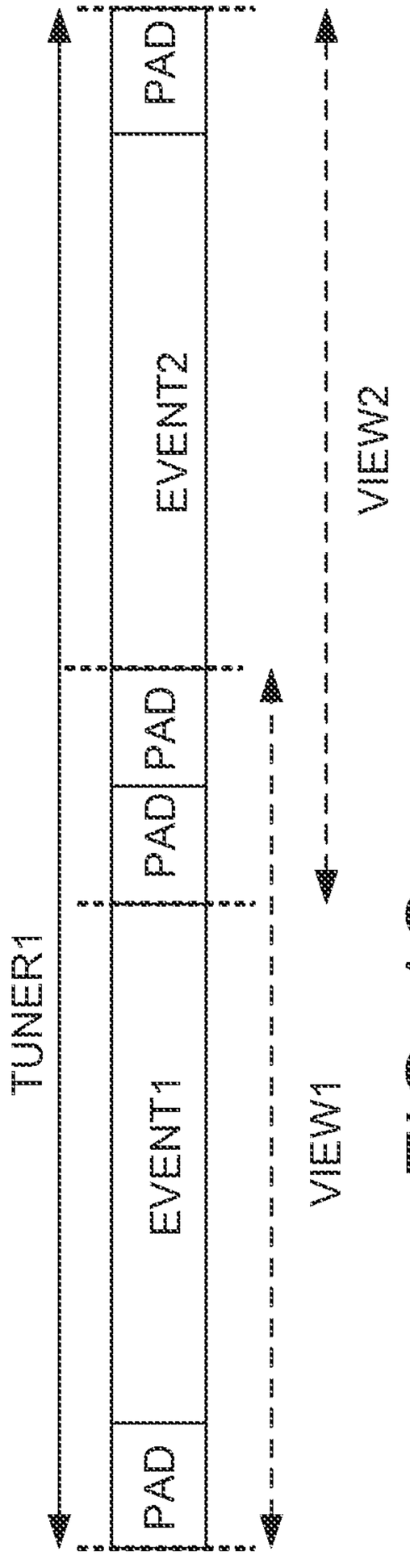


FIG. 1C

CONVENTIONAL SINGLE TUNER RECORDING

CONVENTIONAL 1 OR 2 TUNER RECORDING

MULTIPLE VIEW RECORDING

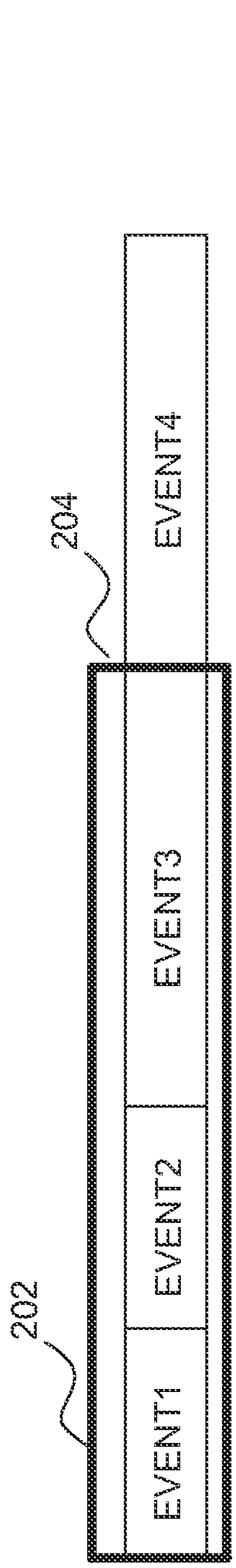


FIG. 2A

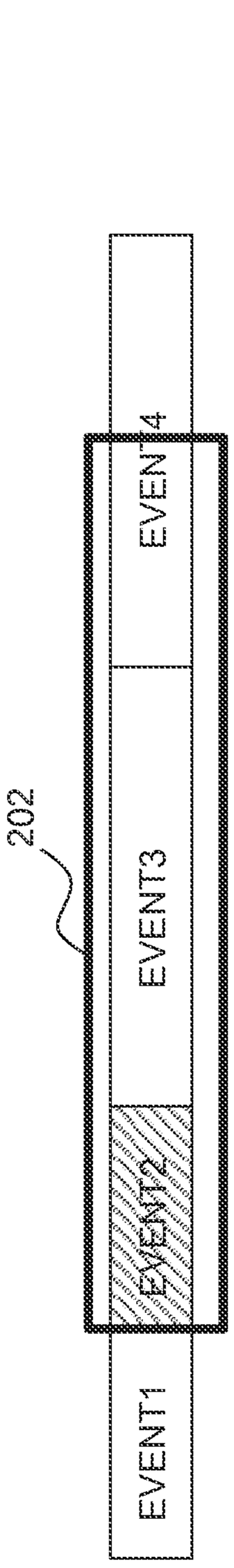


FIG. 2B

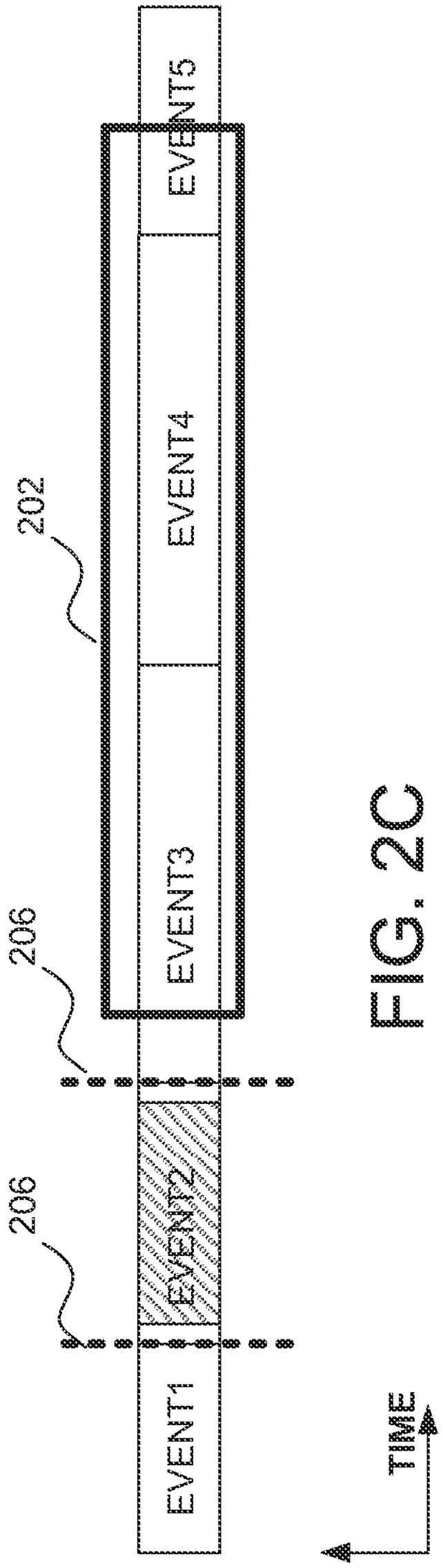


FIG. 2C

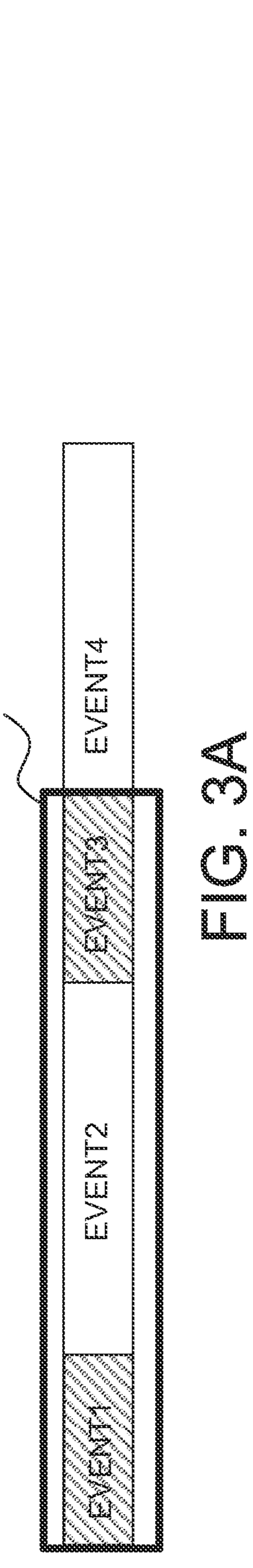


FIG. 3A

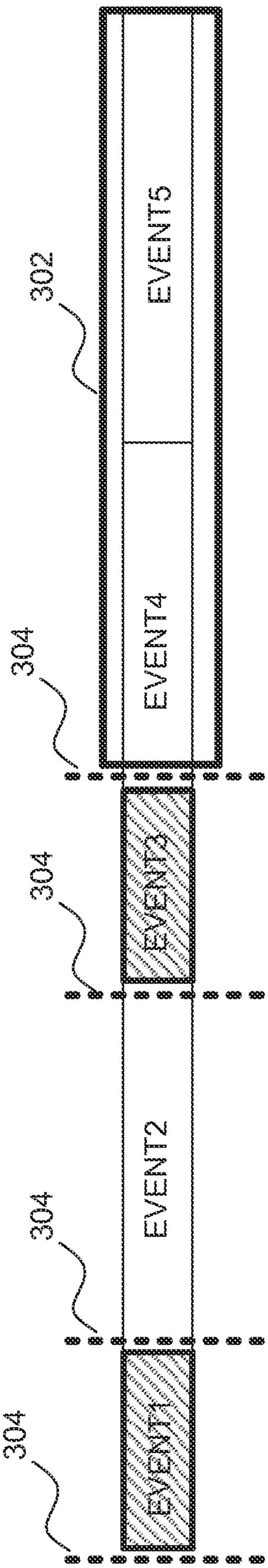


FIG. 3B

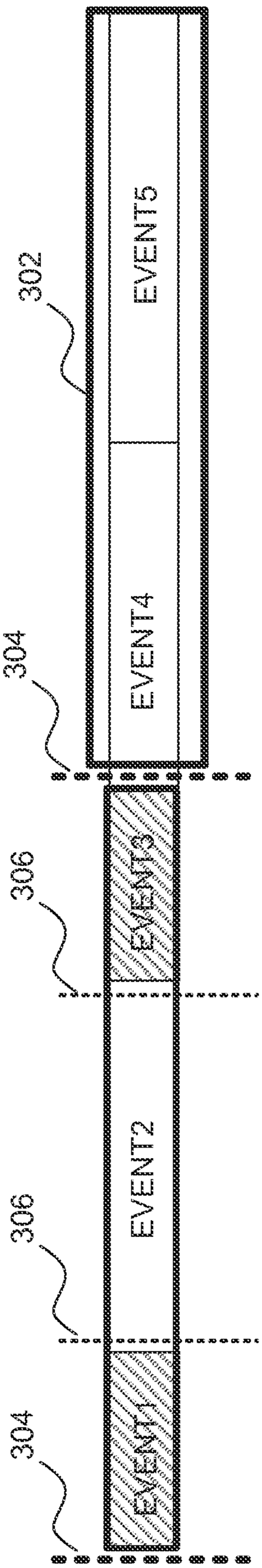


FIG. 3C

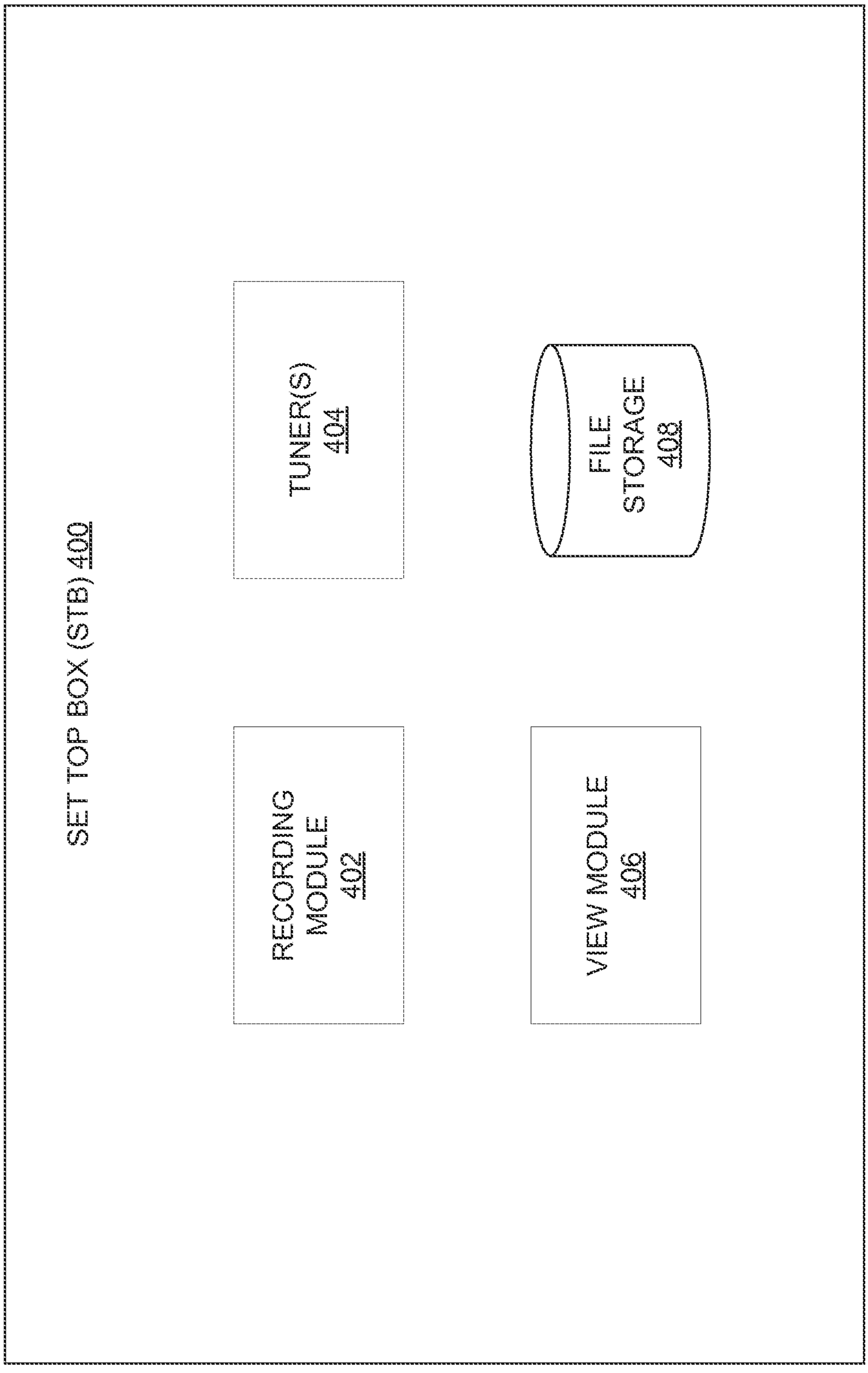

SET TOP BOX (STB) 400
RECORDING MODULE 402
TUNER(S) 404
VIEW MODULE 406
FILE STORAGE 408

FIG. 4

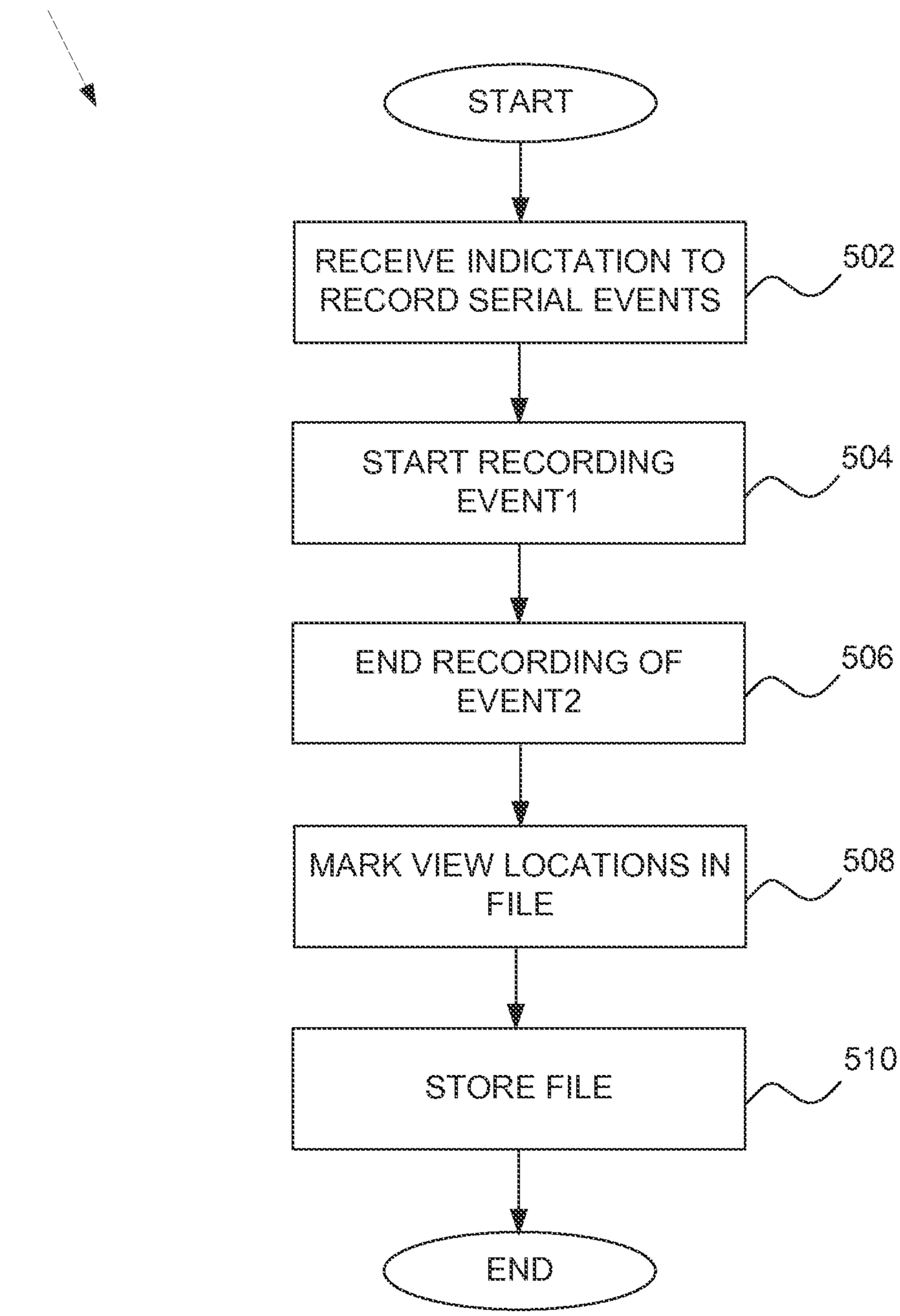

500
START
RECEIVE INDICTATION TO RECORD SERIAL EVENTS
502
START RECORDING EVENT1
504
END RECORDING OF EVENT2
506
MARK VIEW LOCATIONS IN FILE
508
STORE FILE
510
END

FIG. 5

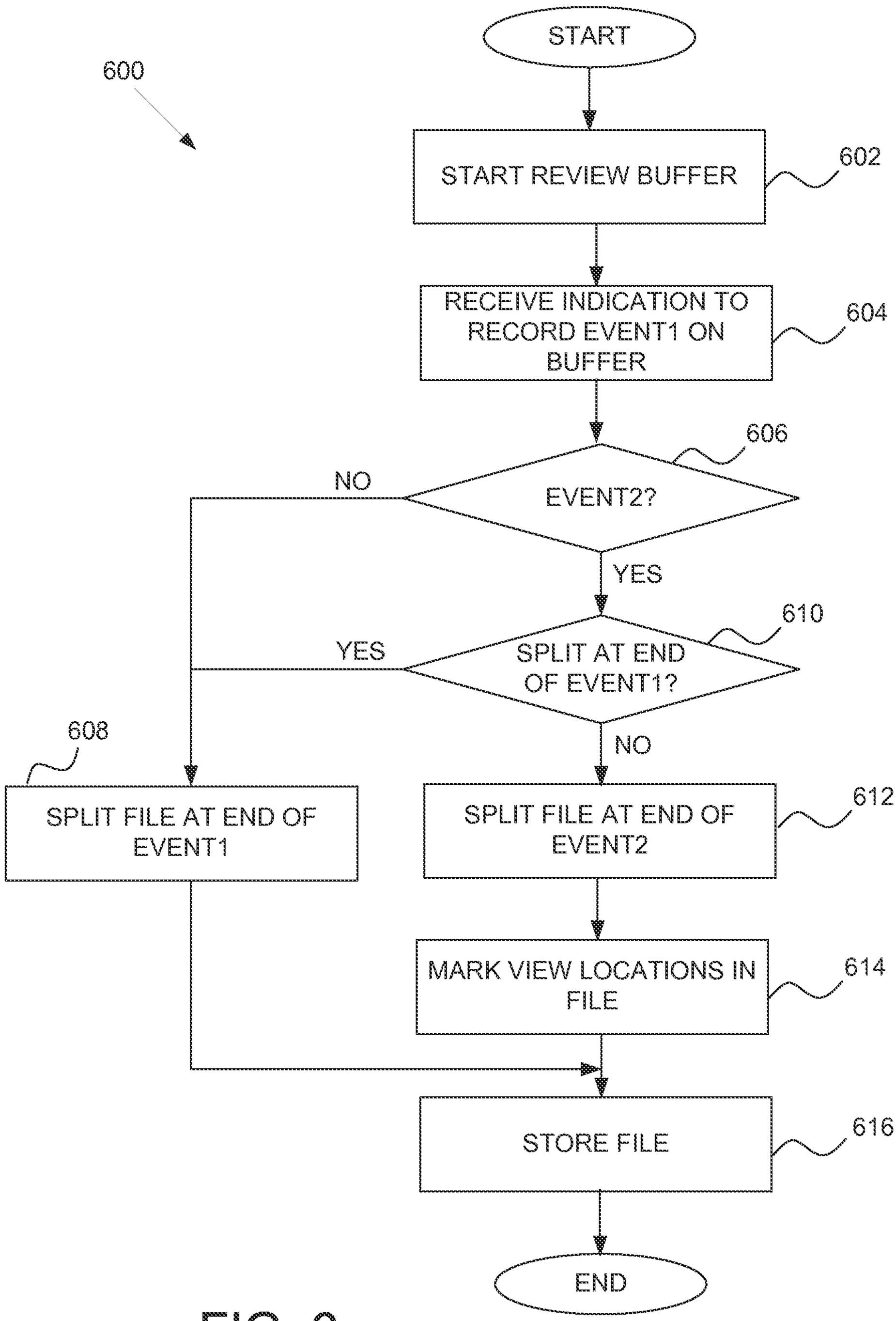

600
START
START REVIEW BUFFER
602
RECEIVE INDICATION TO RECORD EVENT1 ON BUFFER
604
606
EVENT2?
NO
YES
610
SPLIT AT END OF EVENT1?
YES
NO
608
SPLIT FILE AT END OF EVENT1
SPLIT FILE AT END OF EVENT2
612
MARK VIEW LOCATIONS IN FILE
614
STORE FILE
616
END

FIG. 6

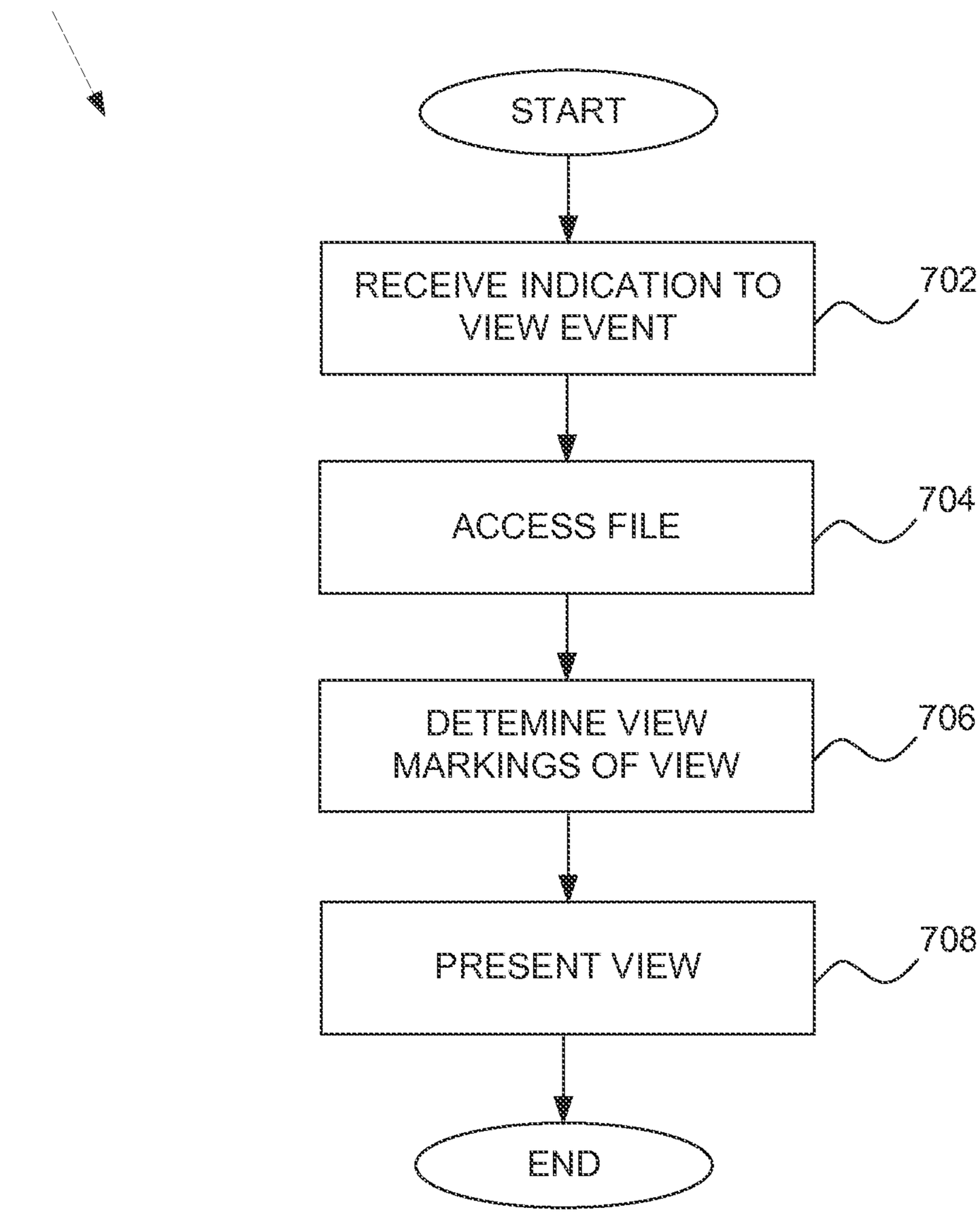

700
START
RECEIVE INDICATION TO VIEW EVENT
702
ACCESS FILE
704
DETEMINE VIEW MARKINGS OF VIEW
706
PRESENT VIEW
708
END

FIG. 7

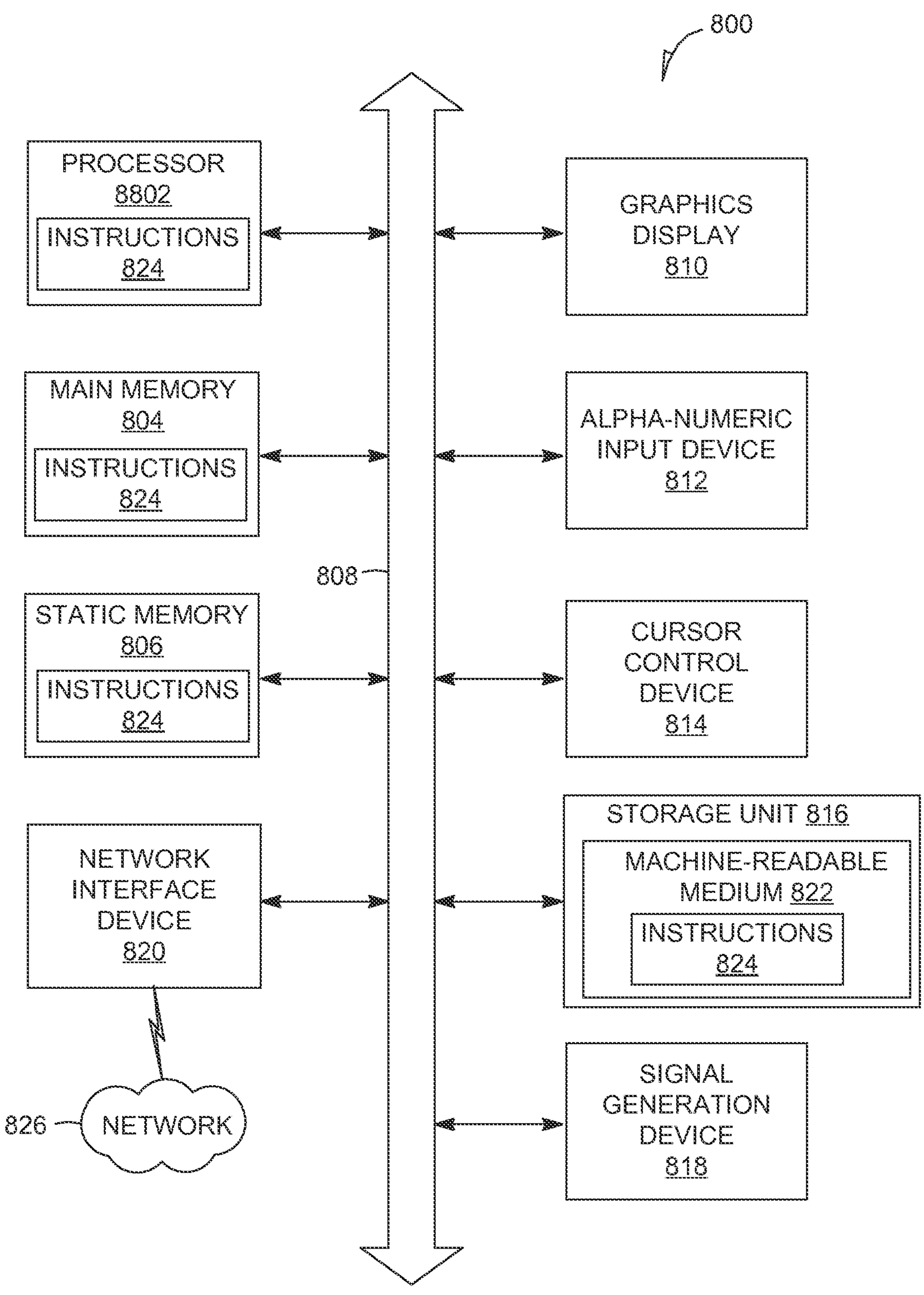

800
PROCESSOR
8802
INSTRUCTIONS
824
MAIN MEMORY
804
INSTRUCTIONS
824
808
STATIC MEMORY
806
INSTRUCTIONS
824
NETWORK
INTERFACE
DEVICE
820
826
NETWORK
GRAPHICS
DISPLAY
810
ALPHA-NUMERIC
INPUT DEVICE
812
CURSOR
CONTROL
DEVICE
814
STORAGE UNIT 816
MACHINE-READABLE
MEDIUM 822
INSTRUCTIONS
824
SIGNAL
GENERATION
DEVICE
818

FIG. 8

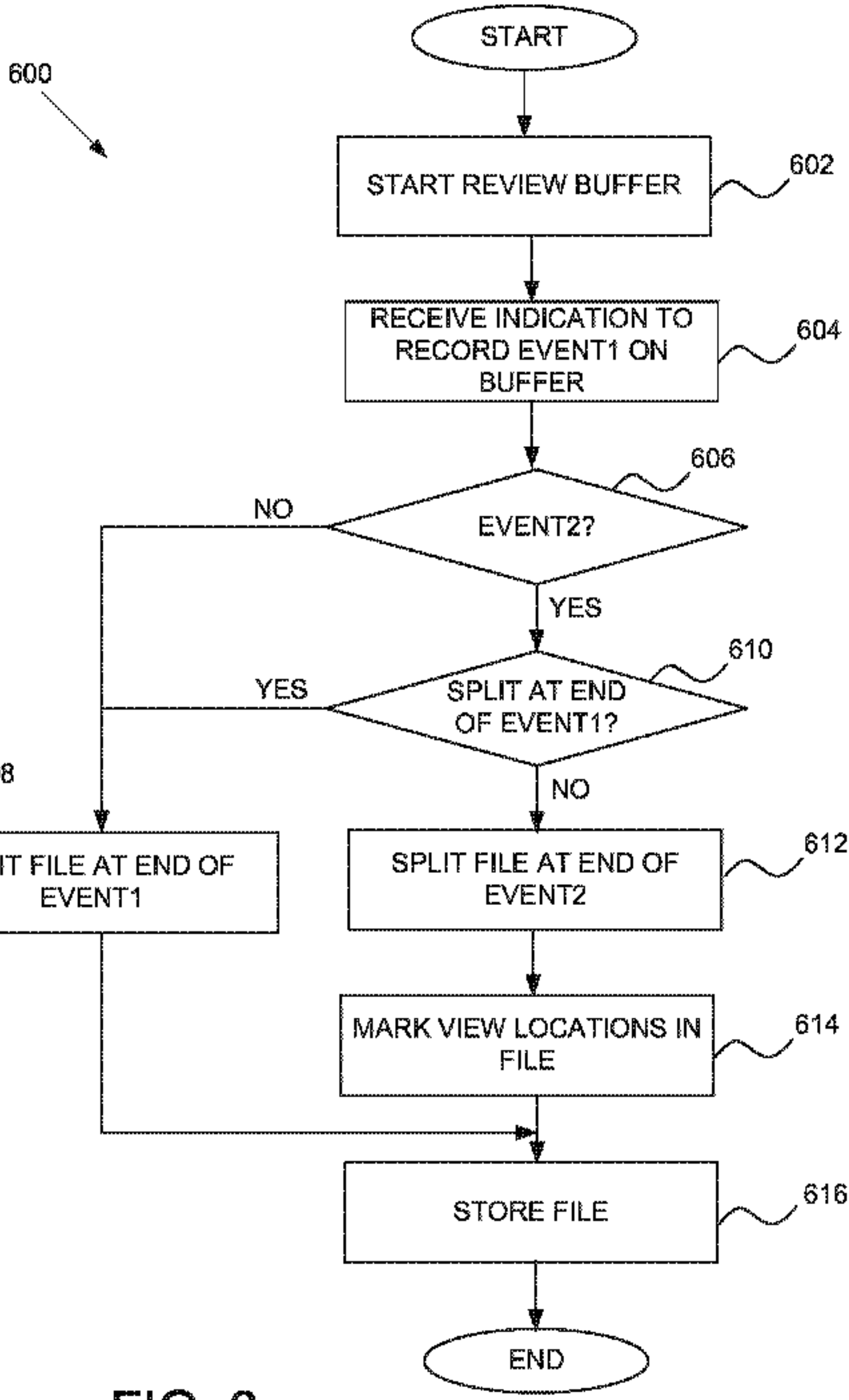

600
START
START REVIEW BUFFER
602
RECEIVE INDICATION TO RECORD EVENT1 ON BUFFER
604
606
NO
EVENT2?
YES
610
YES
SPLIT AT END OF EVENT1?
NO
608
SPLIT FILE AT END OF EVENT1
SPLIT FILE AT END OF EVENT2
612
MARK VIEW LOCATIONS IN FILE
614
STORE FILE
616
END

FIG. 6