

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
26 March 2009 (26.03.2009)

PCT

(10) International Publication Number
WO 2009/038902 A1

(51) International Patent Classification:
G06F 3/14 (2006.01)

(74) Agent: **YAO, Shun**; Park, Vaughan & Fleming LLP, 2820
Fifth Street, Davis, California 95618-7759 (US).

(21) International Application Number:
PCT/US2008/072911

(22) International Filing Date: 12 August 2008 (12.08.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/858,358 20 September 2007 (20.09.2007) US

(71) Applicant (for all designated States except US): **APPLE INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **HOWARD, Brian D.** [US/US]; 59 Linaria Way, Portola Valley, CA 94028 (US). **BAKER, Paul A.** [US/US]; 847 Starlite Lane, Los Altos, California 94024 (US). **CULBERT, Michael F.** [US/US]; 18500 Hillview Drive, Monte Sereno, CA 95030 (US). **CONROY, David G.** [CA/US]; 1071 San Carlos Avenue, El Granada, CA 94018 (US). **ATHAS, William C.** [US/US]; 1047 Bird Avenue, San Jose, CA 95125 (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, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

(54) Title: SWITCHING BETWEEN GRAPHICS SOURCES TO FACILITATE POWER MANAGEMENT AND/OR SECURITY

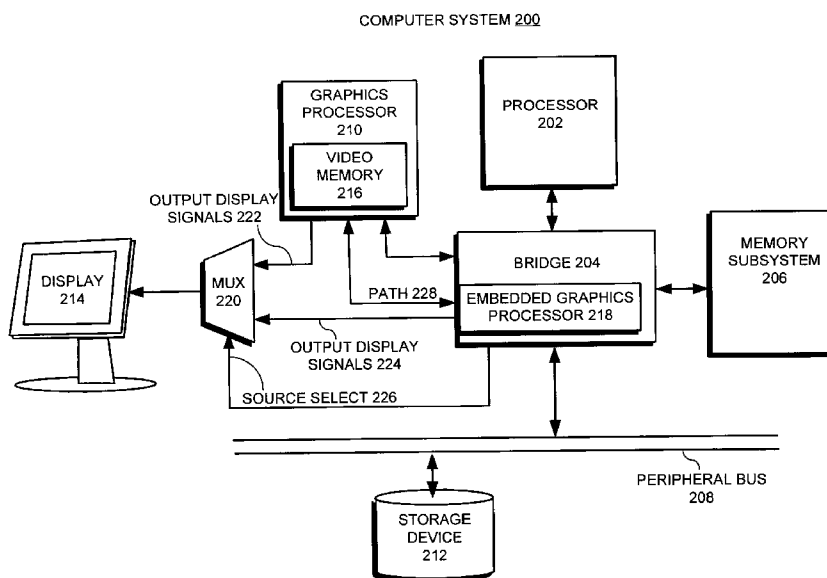


FIG. 2

(57) Abstract: One embodiment of the present invention provides a system that switches between frame buffers which are used to refresh a display. During operation, the system refreshes the display from a first frame buffer which is located in a first memory. Upon receiving a request to switch frame buffers for the display, the system reconfigures data transfers to the display so that the display is refreshed from a second frame buffer which is located in a second memory.

WO 2009/038902 A1

SWITCHING BETWEEN GRAPHICS SOURCES TO FACILITATE POWER MANAGEMENT AND/OR SECURITY

Inventors: Brian D. Howard, Paul Baker,
Michael F. Culbert, David G. Conroy and William C. Athas

BACKGROUND

Field of the Invention

[0001] The present invention relates to techniques for switching between graphics sources in computer systems. More specifically, the present invention relates to a method and an apparatus for reducing power and/or improving security by switching between graphics sources in a computer system.

Related Art

[0002] Rapid advances in computing technology have made it possible to perform trillions of computational operations each second on data sets that are sometimes as large as a trillion bytes. These advances can be largely attributed to the exponential increase in the size and complexity of integrated circuits. Unfortunately, the increase in size and complexity of integrated circuits has been accompanied by a similar increase in power consumption.

[0003] In a parallel development, the rapid proliferation of broadband wireless networks has given rise to a dramatic increase in the number of portable computer systems. Unfortunately, portable computer systems usually have stringent power constraints due to the limited battery power that is available to them. These developments have created a significant need for power-saving techniques.

[0004] Advances in 3D graphics technology have led most of the modern computer systems to use dedicated graphics processors (sometimes referred to as graphics processing units (GPUs)) to drive graphics display devices. Unfortunately, today's GPUs consume a large amount of power, which severely shortens the battery life of portable computer systems, and also causes heat dissipation problems.

[0005] While a graphics display is operating, there are often times when very little graphics processing is needed, for example when the user is reading a document on the display. Unfortunately, existing graphics processors cannot easily switch to a low-power mode to conserve power during these “low activity” periods.

5 [0006] One technique for saving power during such “low activity” periods is to switch the display from a high-power graphics source (e.g., a high-performance GPU) to a low-power graphics source (e.g., a low-performance GPU). Ideally, this switching operation should be invisible to the user, so that the system can seamlessly switch back and forth between the different graphics sources as the graphics processing demands change, or as the
10 system’s need to limit power consumption changes.

[0007] One existing technique provides a mechanical switch which allows a user to switch between a lower-performance graphics source and a higher-performance graphics source. However, this brute-force technique requires the user to fully re-initialize the computer system each time the user switches from one graphics source to another. Requiring
15 a user to re-initialize the computer system to switch from one graphics source to another is simply not acceptable in many situations. An initialization process is one of the most disruptive operations that can be performed on the computer. Typically the user has to save all his or her work before re-initializing the computer, which can take a considerable amount of time. Furthermore, the user must first decide whether their graphics processing
20 requirements will be high or low in the near future, and must then wait for the system to re-initialize, and then be willing to wait for another re-initialization if requirements change.

[0008] Another problem is that some graphics processors render images into a frame buffer located in insecure main memory. This can cause issues with Digital Rights Management (DRM) standards which require that such graphical images be stored securely.

25 [0009] Hence, what is needed is a method and an apparatus that facilitates rapid and/or seamless switching between different graphics sources to reduce power and/or provide security.

SUMMARY

30 [0010] One embodiment of the present invention provides a system that switches between frame buffers which are used to refresh a display. During operation, the system refreshes the display from a first frame buffer which is located in a first memory. Upon

receiving a request to switch frame buffers for the display, the system reconfigures data transfers to the display so that the display is refreshed from a second frame buffer which is located in a second memory.

[0011] In some embodiments, the first memory is a main memory, which is accessible
5 by numerous applications and is hence insecure, and the second memory is a secure frame buffer which is located outside of main memory.

[0012] In some embodiments, switching the display additionally involves transferring data which is used to refresh the display so that the data completely bypasses the insecure main memory. In this variation, the system encrypts the data while the data is stored in the
10 second frame buffer and while the data is in transit to and from the second frame buffer.

[0013] In some embodiments, prior to receiving the request to switch frame buffers, the system: determines a security requirement for data associated with the display; and generates the request to switch frame buffers based on the determined security requirement.

[0014] In some embodiments, prior to receiving the request to switch frame buffers,
15 the system: monitors a level of graphics-processing load for the display; and generates the request to switch based on the level of graphics-processing load.

[0015] In some embodiments, the system: measures a temperature in a computer system which contains the display; and generates the request to switch based on the measured temperature.

[0016] In some embodiments, switching the display so that the display is refreshed from the second frame buffer additionally involves switching a graphics processing unit (GPU) which performs rendering operations for the display. In some embodiments, the GPU is switched between a low-power GPU, which renders to the first frame buffer, and a high-power GPU which renders to the second frame buffer.

[0017] In some embodiments, prior to switching from the low-power GPU to the
25 high-power GPU, the system substantially synchronizes the low-power GPU's output display signals and the high-power GPU's output display signals, thereby facilitating a seamless transition which does not disrupt graphical output on the display.

[0018] In some embodiments, substantially synchronizing the output display signals
30 involves using one or more phase-locked loops (PLL).

[0019] In some embodiments, the switching takes place during a vertical blanking interval associated with a vertical blanking signal for the display.

[0020] Another embodiment of the present invention provides a computer system that switches between a first graphics processor and a second graphics processor to drive a first display and/or a second display. This computer system includes: a processor; a memory; a first graphics processor; a second graphics processor; a first display, and a second display.

5 The computer system also includes a first switch, which selectively couples either the first graphics processor or the second graphics processor to the first display. It also includes a second switch, which selectively couples either the first graphics processor or the second graphics processor to the second display.

[0021] In some embodiments, the first display is an internal display, which is
10 integrated into the computer system, and the second display is an external display, which is coupled to the computer system.

[0022] In some embodiments, the first switch and the second switch are configured either to couple the first graphics processor to both the first display and the second display, or to couple the second graphics processor to both the first display and the second display.

15 [0023] In some embodiments, the first graphics processor is a high-power graphics processing unit (GPU) and the second graphics processor is a low-power GPU.

[0024] In some embodiments, the system includes a synchronization mechanism, which is configured to substantially synchronize the first graphics processor's output display signals and the second graphics processor's output display signals, thereby facilitating a
20 seamless switching process which does not disrupt graphical output.

[0025] In some embodiments, the synchronization mechanism is configured to use one or more phase-locked loops (PLL) to substantially synchronize the output display signals.

[0026] In some embodiments, the first switch and the second switch can include:
25 multiplexers; or wired-OR logic.

BRIEF DESCRIPTION OF THE FIGURES

30 [0027] FIG. 1 illustrates a computer system in accordance with an embodiment of the present invention.

[0028] FIG. 2 illustrates a computer system which can switch between different graphics sources to drive the same display in accordance with an embodiment of the present invention.

5 [0029] FIG. 3 presents a flow chart illustrating the process of switching from a first graphics source to a second graphics source to drive a display in accordance with an embodiment of the present invention.

[0030] FIG. 4 presents a flow chart illustrating the process of switching from the first graphics source to the second graphics source without synchronizing the output display signals in accordance with an embodiment of the present invention.

10 [0031] FIG. 5A illustrates a single vertical blanking interval (VBI) and a corresponding vertical synchronization (V-sync) pulse generated by a graphics source in accordance with an embodiment of the present invention.

[0032] FIG. 5B illustrates two overlapping VBIs generated by two graphics sources in accordance with an embodiment of the present invention.

15 [0033] FIG. 6A presents a schematic of a technique for synchronizing timing signals between two graphics sources in accordance with an embodiment of the present invention.

[0034] FIG. 6B presents a schematic of another technique for synchronizing timing signals between two graphics sources in accordance with an embodiment of the present invention.

20 [0035] FIG. 7 illustrates a computer system comprising two graphics sources in accordance with an embodiment of the present invention.

[0036] FIG. 8 presents a flow chart illustrating the process of switching from the first graphics source to the second graphics source in accordance with an embodiment of the present invention.

25 [0037] FIG. 9 presents a flow chart illustrating the process of switching from the second graphics source to the first graphics source in accordance with an embodiment of the present invention.

[0038] FIG. 10 illustrates a computer system which can switch between different graphics sources to drive an internal display and an external display in accordance with an embodiment of the present invention.

30

DETAILED DESCRIPTION

[0039] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily
5 apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not limited to the embodiments shown, but is to be accorded the widest scope consistent with the claims.

[0040] The data structures and code described in this detailed description are
10 typically stored on a computer-readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, volatile memory, non-volatile memory, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs), DVDs (digital versatile discs or digital video discs), or other media capable of storing computer-readable media now known
15 or later developed.

Computer System

[0041] FIG. 1 illustrates a computer system 100 in accordance with an embodiment of the present invention. As illustrated in FIG. 1, computer system 100 includes processor 102,
20 which is coupled to a memory subsystem 106, peripheral bus 108, and to a graphics processor 110 through bridge 104. Bridge 104 can include any type of core logic unit, bridge chip, or chipsets that are commonly used to couple together components within computing system 100. In one embodiment of the present invention, bridge 104 is a north bridge chip. Processor 102 can include any type of processor, including, but not limited to, a
25 microprocessor, a digital signal processor, a device controller, or a computational engine within an appliance.

[0042] It should be recognized that one or more components of the computer system 100 may be located remotely and accessed via a network.

[0043] Processor 102 communicates with memory subsystem 106 through bridge 104.
30 Memory subsystem 106 can include a number of components, including one or more memory chips which can be accessed by processor 102 at high speed.

[0044] Processor 102 also communicates with storage device 112 through bridge 104 and peripheral bus 108. Storage device 112 can include any type of non-volatile storage

device that can be coupled to a computer system. This includes, but is not limited to, magnetic, optical, and magneto-optical storage devices, as well as storage devices based on flash memory and/or battery-backed up memory.

[0045] Processor 102 additionally communicates with graphics processor 110 through bridge 104. Graphics processor 110 is a specialized graphics-rendering device that provides a signal source to display 114 and drives display 114. Display 114 can include any type of display device that can present information in a visual format (including images and text) to a user. This includes, but is not limited to, cathode ray tube (CRT) displays, light-emitting diode (LED) displays, liquid-crystal displays (LCD), organic LED (OLED) displays, surface-conduction electron-emitter displays (SED), or electronic paper.

[0046] Graphics processor 110 performs both 2D and 3D graphics-rendering operations, such as lighting, shading and transforming, with high performance. To achieve the high performance, graphics processor 110 may utilize dedicated video memory 116 to store frame buffers, textures, vertex arrays, and/or display lists.

[0047] Bridge 104 also includes an embedded graphics processor 118. Embedded graphics processor 118 is typically built for modest performance graphics processing purposes, and hence consumes much less power than graphics processor 110. Note that in FIG. 1, embedded graphics processor 118 is not directly coupled to and does not drive display 114.

[0048] Note that although the present invention is described in the context of computer system 100 illustrated in FIG. 1, the present invention can generally operate on any type of computing device that supports more than one graphics processor. Hence, the present invention is not limited to the computer system 100 illustrated in FIG. 1.

Selectively Switching Between Graphics Sources

[0049] FIG. 2 illustrates a computer system 200 which can switch between different graphics sources to drive the same display in accordance with an embodiment of the present invention. Note that in FIG. 2, the two graphics sources (graphics processor 210 and embedded graphics processor 218) can each independently drive display 214. However, the graphics source that actively drives display 214 at a given time is determined by selecting device 220 which can select between the two graphics sources. Specifically, computer

system 200 can use selecting device 220 to select a graphics source based on its current operation conditions.

[0050] More specifically, output display signals 222 from graphics processor 210, and output display signals 224 from embedded graphics processor 218 are both coupled to inputs of a two-to-one multiplexer (MUX) 220. The output of MUX 220 is controlled by source select 226, which determines which one of the two graphics sources should drive display 214. In this embodiment, source select 226 is the output of bridge chip 204, which comprises specific logic for generating source select 226. Note that source select 226 can also be produced by a logic block other than bridge 204.

[0051] The output display signals from the selected graphics source are then coupled to the inputs of display 214 to actively drive it. Although the selecting device is shown as a multiplexer, it can also include any other type of selecting device, such as a simple wired-OR logic.

[0052] In one embodiment of the present invention, graphics processor 210 and embedded graphics processor 218 can cooperate through a path 228, so that they can synchronize their output display signals. Because the output display signals can include both timing signals and data signals, synchronizing the output display signals can involve synchronizing both the respective timing signals and the respective data signals. Note that path 228 can be realized using hardware and/or software to facilitate synchronizing the two graphics sources.

[0053] In one embodiment of the present invention, graphics processor 210 is a high-performance graphics processor unit (GPU) which consumes a large amount of power, whereas embedded graphics processor 218 is a lower-performance GPU which consumes a smaller amount of power. In this embodiment, when the graphics processing load is light, the system switches the graphics source from graphics processor 210 to embedded graphics processor 218 to drive display 214, and subsequently powers down graphics processor 210 entirely, thereby saving power. On the other hand, when the graphics processing load becomes heavy again, the system switches the graphics source from embedded graphics processor 218 back to graphics processor 210.

[0054] Note that although we have described switching between graphics processors in the context of a standalone graphics processor and an integrated graphics processor illustrated in FIG. 2, the present invention can generally work for a computer system

comprising two or more graphics processors, wherein each of the graphics processors can independently drive the display when properly configured. Moreover, these multiple graphics processors can have different operating characteristics, including different power consumption levels. Furthermore, each of the multiple graphics processors can be either a standalone graphics processor or an integrated graphics processor within a chip. Hence, the present invention is not limited to the computer system 200 illustrated in FIG. 2.

[0055] Note that the above-described technique for switching between different graphics sources does not require shutting down the computer system or re-initializing the computer system. As a result, the switching process can take substantially less time than it would have if a re-initialization had been required. Consequently, the present invention allows rapid and frequent switching between the graphics processors.

[0056] FIG. 3 presents a flowchart illustrating the process of switching from a first graphics source to a second graphics source to drive a display in accordance with an embodiment of the present invention.

[0057] During operation, the system first receives a request to switch the signal source for the display from a first graphics processor which is actively driving the display to a second graphics processor which is in a non-active state (step 302).

[0058] The switching request can be generated by a user who is aware of levels of graphics processing load. Alternatively, the switching request can be generated internally by the system.

[0059] In one embodiment of the present invention, system software continuously monitors the level of graphics processing load. More specifically, the system can determine the level of graphics processing load based on a condition in a graphics command queue associated with the graphics processor. For example, if the command queue is mostly empty, the system asserts a low graphics processing load. On the other hand, if the command queue is mostly full, the system asserts a high graphics processing load.

[0060] Next, based on the level of the graphics processing load, the system software selects one of the two graphics processors, and subsequently generates the request to switch if the non-active graphics processor is selected.

[0061] For example, if the first graphics processor is a high-performance GPU that consumes high power, when the system software detects a considerable decrease in the level of graphics processing load, the system software can issue a request to switch to a second

graphics processor which has lower performance, but which also consumes much less power.

On the other hand, if the first graphics processor is a lower-performance and low-power GPU, the system can issue a request to switch to a high-performance and high-power GPU if the system software detects a considerable increase in the level of graphics processing load.

5 **[0062]** Note that using system software to monitor the graphics processing load and to automatically issue the switching request is significantly faster and possibly more energy efficient than a human-initiated request. Furthermore, using system software can free the user from the monitoring job.

10 **[0063]** Next, in response to the switching request, the system configures the second graphics processor in preparation for driving the display (step 304). In one embodiment of the present invention, configuring the second graphics processor can involve one or more of the following steps: (1) powering up the processor if it is currently powered down; (2) initializing the graphics processor; and (3) generating output signals in preparation for powering up the display.

15 **[0064]** The system then switches the signal source which drives the display from the first graphics processor to the second graphics processor, which causes the second graphics processor to drive the display (step 306). In one embodiment of the present invention, the switching involves using a selecting device such as MUX 220 in FIG. 2, which decouples the first graphics processor from, and couples the second graphics processor to, the display.

20 During the switching operation, different timing controls can be used which will be described in more detail below. In general, obtaining a smoother switching transition requires more precise timing control and, hence, typically requires a more complex switch-controlling mechanism.

25 **[0065]** Once the second graphics processor takes over from the first graphics processor, the system may power down the first graphics processor to conserve power. Note that the above-described switching process does not require re-initializing the whole system to take effect.

30 **[0066]** Note that although we have described switching based on graphics processing load, the switch request can also be generated based on power conditions (e.g., whether the system is running on a battery or an external power source, or whether the battery is low), based on a need to reduce system heat dissipation, based on user preference, or based on any

feature or capability that is different between the two graphics processors.

Timing during Switching

[0067] Switching between different graphics processors to drive the same display device requires a certain level of cooperation between the graphics processors to ensure a substantially seamless transition. We discuss different timing techniques during a switching below by distinguishing them based on whether synchronization is involved in the output display signals.

Switching without Synchronization

[0068] FIG. 4 presents a flowchart illustrating the process of switching from the first graphics source to the second graphics source without synchronizing the output display signals in accordance with an embodiment of the present invention.

[0069] During operation, the first graphics processor fades out the display (step 402). Note that this can be done in a number of ways, including, but not limited to, displaying black or other colors on the screen, turning off the backlight, or powering down the entire display.

[0070] Next, the system switches the signal source that drives the display from the first graphics processor to the second graphics processor, which has been configured to drive the display (step 404). More specifically, the switching involves decoupling the first graphics processor's output signals from the input of the display and coupling the second graphics processor's output signals to the input of the display.

[0071] Upon completing the switching, the second graphics processor then initializes the display if necessary (step 406). Next, the second graphics processor redraws the display screen and subsequently fades in the display screen (step 408).

[0072] In this embodiment, the two graphics sources are not required to synchronize with each other. Consequently, the second signal source does not need to be configured to redraw the display before the switch takes place. Furthermore, the first signal source can be turned off (e.g., through a fade-out operation) prior to performing the switch.

[0073] Note that switching without synchronization is simple but can cause the user to notice the switch. However, if the switching can be completed within a fraction of a second, the user may not even notice the switch. Alternatively, if the switching is done more

slowly, the visual disruption can be reduced by using an appropriate visual effect, such as a fade-out/fade-in effect when the display resolution is changed. Generally, any undesirable visual effects of switching the display from one set of display signals to a different, unsynchronized set of display signals can be hidden by fading out the display during the transition.

Switching with Synchronization

[0074] Synchronizing the output signals prior to switching facilitates a smoother, less noticeable, or even seamless switching process which does not disrupt graphical output on the display. However, the synchronization requires the second graphics source to start generating output signals in preparation for driving the display prior to the switching, so that the output display signals from both graphics sources can be synchronized.

[0075] In one embodiment of the present invention, synchronizing the output signals from the two graphics sources can be achieved by matching up timing information embedded in the output signals. Such timing information can include, but is not limited to, horizontal synchronization (H-sync) pulses, vertical synchronization (V-sync) pulses, horizontal blanking signals, and vertical blanking signals. In particular, V-sync pulses control image refresh on the display by indicating when to start scanning a new frame of data. Typically, V-sync pulses occur within a short time interval between two consecutive image frames, referred to as a vertical blanking interval (VBI), during which the display on the screen is held in a constant state for various housekeeping purposes. FIG. 5A illustrates a single VBI 502 and a corresponding V-sync pulse 504 produced by a graphics source in accordance with an embodiment of the present invention. Note that the V-sync pulse 504 falls within VBI 502.

[0076] In this embodiment, the computer system keeps track of when V-sync pulses occur in the first graphics source, and adjusts the timing sequence of the second graphics source until its V-sync pulses are aligned with those of the first graphics source. In one embodiment, aligning the V-sync pulses from the two graphics sources involves using either software or hardware to cause the timing sequence of the second graphics source to coincide with the timing sequence of the first graphics source. During this alignment period, the first graphics source continues to drive the display. When the V-sync pulses are sufficiently aligned between the two sources, switching can then be performed during a next VBI.

[0077] FIG. 5B illustrates two overlapping VBIs – VBI 506 and VBI 508 generated by two graphics sources in accordance with an embodiment of the present invention. Note that the switching occurs within overlapping period 510 of the two VBIs. Also note that the switching process may appear invisible to a user if it can be completed within overlapping period 510. Furthermore, the substantial synchronization between the two graphics sources facilitates the second graphics source to start driving the display immediately so that it appears to the user as if the display did not change.

[0078] However, it is possible for the switching process to take longer than a single VBI to complete, or to take up a few frame times to resolve. In this case, the system can hide the switching effect by blanking or fading out the screen completely.

[0079] In another embodiment of the present invention, instead of causing the second graphics source to align with the first, the system can allow the V-sync signals of the second graphics source to drift against those of the first graphics source. Such a drift in the timing signals can occur as a result of one or more timing differences. For example, the drift can be caused by a slight difference in the clock frequencies of two graphics processors. Alternatively, the drift can be caused by programming the two graphics processors to operate at slightly different display frame rates.

[0080] In this embodiment of synchronization, the system can monitor the two V-sync signals from the two sources and detect when they overlap with each other, wherein the monitoring can be performed by either software or hardware. When this occurs, the system can switch from one graphics source to the other before the two signals drift away from each other.

Switching with Hardware-Based Synchronization

[0081] In one embodiment of the present invention, one of the graphics sources can be synchronized to the other graphics source using additional hardware, so that the display output timing of the two graphics sources can be aligned precisely. A switch can then be made during a next VBI so that the switch is undetectable by the user. In this embodiment, a smoother switch is made possible by incorporating the additional hardware to adjust the phase and frequency of the second graphics source's display timing generator to align the display output timing to that of the first graphics source.

[0082] FIG. 6A presents a schematic of a technique for synchronizing timing signals between two graphics sources in accordance with an embodiment of the present invention. As illustrated in FIG. 6A, the two graphics sources *A* and *B* comprise timing generator 602 and timing generator 604, respectively. Timing generator 602 produces V-sync pulses in output V-SYNC 606 and vertical blanking intervals in output VBI 608 for graphics source *A*, while timing generator 604 produces V-sync pulses in output V-SYNC 610 and vertical blanking intervals in output VBI 612 for graphics source *B*.

[0083] Graphics sources *A* and *B* also use phase-locked loop (PLL) 614, and PLL 616 to provide frequency references for timing generators 602 and 604, respectively. More specifically, PLL 614 and PLL 616 receive reference frequency inputs f_{REF}^A 618 and f_{REF}^B 620 from the left, and generate reference frequency outputs f_{OUT}^A 622 and f_{OUT}^B 624 as inputs to timing generators 602 and 604. A detailed explanation of the functions of a PLL and associated components can be found in a number of references that describe PLLs (see Floyd M. Gardner, "Charge-Pump Phase-Lock Loops," IEEE Transactions on Communications, Vol. 28, No. 11, November 1980).

[0084] For frequency synthesizing purposes, PLL 614 comprises a divider M_A 626 and a divider N_A 628. Similarly, PLL 616 comprises a divider M_B 630 and a divider N_B 632. The output of PLL 614 and PLL 616, when phase locked, produce output frequency $f_{\text{OUT}}^A = f_{\text{REF}}^A \times (M_A / N_A)$, and $f_{\text{OUT}}^B = f_{\text{REF}}^B \times (M_B / N_B)$, respectively.

[0085] In one embodiment of the present invention, frequency scalar values M_A , M_B , N_A , N_B are programmable and are stored in programmable registers. Specifically, scalars M_A , M_B , N_A , N_B are coupled to and are programmable through a controller 634, which can be implemented either in software or in hardware as microcontroller or a finite state machine. Controller 634 receives a request to switch input – REQSW 636, and additionally receives clock signals V-SYNC_A 606 and VBI_A 608 from graphics source *A*, and V-SYNC_B 610 and VBI_B 612 from graphics source *B*. Controller 634 then measures the phase difference between either the V-sync signals or the VBI signals of the two graphics sources. Using the measured phase difference as a feedback signal, controller 634 can then adjust the phase of V-sync and VBI from one graphics source relative to the other graphics source by synchronously changing the *M* and *N* values in the associated PLL.

[0086] Using the feedback loop, controller 634 continues measuring and adjusting the phase difference. When controller 634 determines that the phase difference is within a

predetermined bound, it then generates a switch enable – OK2SWITCH 638. In one embodiment of the present invention, OK2SWITCH 638 is coupled to source select 226 in FIG. 2, which enables MUX 220 to flip the source.

[0087] Note that the above description allows clocks in both the active graphics source and the non-active graphics source to be changed. In particular, if the PLL scalar values being changed are associated with the source actively driving the display, it may be desirable to adjust the associated frequency slowly and smoothly. Also note that we may not need to obtain a perfect clock alignment to allow a switch. In one embodiment, controller 634 can be configured to align VBIs to obtain just enough overlap so that the switching operation does not cause visible artifacts. When the controller detects there is sufficient overlap, it asserts OK2SWITCH signal to complete the synchronization.

[0088] FIG. 6B presents a schematic of another technique for synchronizing timing signals between two graphics sources in accordance with an embodiment of the present invention.

[0089] In this embodiment, a single PLL 640 is used to synchronize timing signals between the graphics sources *A* and *B*. Note that there's no direct control of the PLL by a controller as in FIG. 6A. Instead, PLL 640 forms a closed loop with one of the timing generators.

[0090] As illustrated in FIG. 6B, timing generators 602 and 604 receive reference frequency inputs f_{REF_A} 642 and f_{REF_B} 644, respectively. The four outputs from timing generators 602 and 604: V-SYNC_A 606, VBI_A 608, V-SYNC_B 610, and VBI_B 612 are coupled to a four-to-two multiplexer MUX 646, which can select either V-SYNC_A 606 and V-SYNC_B 610, or VBI_A 608 and VBI_B 612 to its outputs. The outputs of MUX 646 are then coupled to the inputs of the phase detector of PLL 640. Note that either the V-sync signals or the VBI signals can be used for alignment in this embodiment.

[0091] Next, the VCO output from PLL 640 is coupled to and serves as the input reference frequency for one of the timing generators, and thereby completes the closed-loop with that timing generator. More specifically, the output from PLL 640 is first coupled to the inputs of two multiplexers MUX 648 and MUX 650, which also receive external clock signals EXTCLK_A 652 and EXTCLK_B 654 as inputs, respectively. The outputs of MUX 648 and MUX 650 are controlled by controller 656, which selects either the external clock source or the PLL output as the reference frequency input for a respective timing

generator. Note that controller 656 receives an input from the phase detector of PLL 640 and detects if PLL 640 has locked based on the input.

[0092] During operation, assume that graphics source *A* is actively driving the display. Meanwhile, the VCO output of PLL 640 is selected as the reference frequency $f_{\text{REF_B}}$ 644 for timing generator 604 of the graphics source *B*. Hence, PLL 640 and timing generator 604 form a closed loop, which facilitates the selected timing signals (either V-sync or VBI) from the two timing generators to sync up. When controller 656 detects that PLL 640 has become phase-locked, it then switches the graphics source that drives the display from graphics source *A* to graphics source *B* during the next blanking interval. More specifically, 10 in the following blanking interval, controller 656 switches the $f_{\text{REF_B}}$ input from PLL 640 to the external clock source EXTCLK_B 654. After the switching, PLL 640 can then be used for locking graphics source *A* to graphics source *B*, which is now actively driving the display.

Choosing Graphics Processors without Switching

15 [0093] In one embodiment of the present invention, instead of switching between two graphics processors to drive the same display device, the lower-performance, lower-power graphics processor always drives the display. In this embodiment, when additional graphics performance is required, the higher-performance processor takes over the graphics processing load, rendering its display image into the same frame buffer used by the lower-performance 20 processor. When the system is operating in this manner, the lower-performance processor acts purely as a display output device, i.e., transferring image data from the frame buffer to the display, while the higher-performance device performs all the graphics processing. When less performance is required, the lower-performance device again takes over the graphics processing tasks, and the higher-performance device can be powered down accordingly.

25

Computer System

[0094] FIG. 7 illustrates a computer system 700 which includes two graphics processors in accordance with an embodiment of the present invention. More specifically, computer system 700 includes a processor 702, which is coupled to a bridge chip 704. 30 Bridge chip 704 is itself coupled to main memory 706, display 714 and peripheral bus 708. Peripheral bus 708 can be used to access storage device 710.

[0095] Note that lower-performance, low-power graphics processor 712 is directly coupled to display 714 and always drives it. On the other hand, high-performance, high-power graphics processor 716 is coupled to graphics processor 712, and is typically powered down when it is not in use.

5 [0096] In one embodiment of the present invention, instead of rendering graphics into its own frame buffer, graphics processor 716 renders images directly into frame buffer 707 for graphics processor 712, wherein frame buffer 707 is located in main memory 706. In this embodiment, graphics processor 712 is responsible for displaying the graphics on display 714 by continuously refreshing display 714. Note that because in this embodiment the display is
10 always driven by the same graphics processor and is refreshing from the same frame buffer, no switching hardware is required and there is no hardware switching transition to hide from the user.

 [0097] In an alternative embodiment, when additional graphics processing power or additional security is needed, the system powers up graphics processor 716 to provide the
15 additional graphics-rendering capacity. Graphics processor 716 renders images to its local frame buffer 722 that resides in a special-purpose graphics memory 720, which is coupled to (or integrated into) graphics processor 716. Note that special-purpose graphics memory 720 is more secure than main memory 706 because main memory 706 is typically shared among many different processes and applications. In this embodiment, graphics processor 712 must
20 change the frame buffer (from which it is refreshing display 714) from its own frame buffer 707 to the frame buffer 722 of graphics processor 716. Because display 714 is always driven by the same graphics processor, no switching hardware is required. However, graphics processor 712 must be programmed to make the switch between frame buffers at the correct time (during the vertical blanking interval, for instance) to avoid a transition that is visible to
25 the user.

Switching Smoothly Between Graphics Sources

 [0098] FIG. 8 presents a flow chart illustrating the process of switching from the first graphics source to the second graphics source in accordance with an embodiment of the
30 present invention. At the start of this process, the lower-power internal graphics processor 712 (also referred to as a “GPU”) is rendering into frame buffer 707 in main memory 706, and display 714 is being refreshed from frame buffer 707 (step 802).

[0099] Next, the system determines whether more performance and/or more security is required (step 804). If not, the system returns to step 802.

[00100] Otherwise, if the system determines that more performance and/or more security is required, the system powers up external high-power graphics processor 716 (step 806). Then, external high-power graphics processor 716 renders an identical image into frame buffer 722 in graphics memory 720 (step 808).

[00101] Next, the system waits for a vertical blanking interval (VBI) (step 810). During this vertical blanking interval, internal low-power graphics processor 712 switches its refresh pointer from frame buffer 707 in main memory 706 to frame buffer 722 in graphics memory 720 (step 812). Next, internal low-power graphics processor 712 powers down except for its refresh circuits (step 814). At this point, the switching process is complete.

[00102] The switching can also take place in the other direction. More specifically, FIG. 9 presents a flow chart illustrating the process of switching from the second graphics source to the first graphics source in accordance with an embodiment of the present invention. At the start of this process, high-power internal graphics processor 716 is rendering into frame buffer 722 in graphics memory 720, and display 714 is being refreshed from frame buffer 722 (step 902).

[00103] Next, the system determines whether less performance and/or less security is required (step 904). If not, the system returns to step 902.

[00104] Otherwise, if the system determines that less performance and/or less security is required, the system powers up internal low-power graphics processor 712 (step 906). Then, low-power graphics processor 712 renders an identical image into frame buffer 707 in main memory 706 (step 908). Next, the system waits for a vertical blanking interval (step 910). During this vertical blanking interval, the internal low-power graphics processor 712 switches its refresh pointer from frame buffer 722 in graphics memory 720 to frame buffer 707 in main memory 706 (step 912). Next, the high-power graphics processor 716 powers down (step 914). At this point, the switching process is complete.

[00105] Note that it is also possible (and probably much easier) for software to simply fade out the display, then make the frame buffer switch and then fade the display back in.

This is simpler because software does not have to make two graphics sources draw the same data into two frame buffers at the same time during the transition. Instead, the system can shut down one source, then start up the other source, and then switch frame buffers.

Another Embodiment

[00106] FIG. 10 illustrates a computer system 1000 which can switch between different graphics sources to drive both an internal display and an external display in accordance with an embodiment of the present invention. Note that in FIG. 10, two graphics processors (graphics processor 1010 and embedded graphics processor 1018) can each independently drive internal display 1014 and external display 1015. The graphics source which actively drives display 1014 is determined by multiplexer 1020, and the graphics source which drives display 1015 is determined by multiplexer 1021. Multiplexers 1020 and 1021 can select between the graphics processor 1010 and embedded graphics processor 1018.

[00107] Note that output display signals 1022 from graphics processor 1010, and output display signals 1024 from embedded graphics processor 1018 are coupled to inputs of a multiplexer (MUX) 1020. Similarly, output display signals 1023 from graphics processor 1010, and output display signals 1025 from embedded graphics processor 1018 are coupled to inputs of MUX 1021. Note that each graphics processor has separate output display signals for driving the two displays. (This is because the two displays can use different display signaling protocols, and are in general different in pixel resolution, color depth, color balance, etc.).

[00108] The output of MUX 1020 is controlled by source select 1026, which determines which one of the two graphics sources should drive internal display 1014. Similarly, the output of MUX 1021 is controlled by source select 1027, which determines which one of the two graphics sources will drive external display 1015. In this embodiment, source selects 1026 and 1027 are outputs of bridge chip 1004, which contains circuitry for generating source selects 1026 and 1027. Note that source selects 1026 and 1027 can also be produced by a logic block which is located outside of bridge 1004.

[00109] The output display signals from the selected graphics source are coupled to the inputs of display 1014 and display 1015. Although the selecting device is shown as a multiplexer, it can also include any other type of selecting device, such as simple wired-OR logic.

[00110] In one embodiment of the present invention, graphics processor 1010 is a high-performance graphics processor unit (GPU), which consumes a large amount of power, whereas embedded graphics processor 1018 is a lower-performance GPU, which consumes a

smaller amount of power. In this embodiment, when the graphics processing load is light, the system switches the graphics source from graphics processor 1010 to embedded graphics processor 1018 to drive displays 1014 and 1015, and subsequently powers down graphics processor 1010 entirely, thereby saving power. On the other hand, when the graphics
5 processing load becomes heavy again, the system switches graphics source from embedded graphics processor 1018 back to graphics processor 1010.

[00111] The foregoing descriptions of embodiments of the present invention have been presented only for purposes of illustration and description. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many
10 modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What Is Claimed Is:

1. A method for switching between frame buffers which are used to refresh a display, comprising:

5 refreshing the display from a first frame buffer which is located in a first memory; receiving a request to switch frame buffers for the display; and in response to the request, reconfiguring data transfers to the display so that the display is refreshed from a second frame buffer which is located in a second memory.

10 2. The method of claim 1, wherein the first memory is a main memory, which is accessible by numerous applications and is hence insecure; and wherein the second memory is a secure frame buffer which is located outside of main memory.

15 3. The method of claim 2, wherein switching the display so that the display is refreshed from the second frame buffer additionally involves:

transferring data which is used to refresh the display so that the data completely bypasses the insecure main memory; and

20 encrypting the data while the data is stored in the second frame buffer and while the data is in transit to and from the second frame buffer.

4. The method of claim 1, wherein prior to receiving the request to switch frame buffers, the method further comprises:

25 determining a security requirement for data associated with the display; and generating the request to switch frame buffers based on the determined security requirement.

30 5. The method of claim 1, wherein prior to receiving the request to switch frame buffers, the method further comprises:

monitoring a level of graphics-processing load for the display; and

generating the request to switch based on the level of graphics-processing load.

6. The method of claim 1, wherein prior to receiving the request to switch frame buffers, the method further comprises:

5 measuring a temperature in a computer system which contains the display; and
generating the request to switch based on the measured temperature.

7. The method of claim 1,

wherein switching the display so that the display is refreshed from the second frame
10 buffer additionally involves switching a graphics processing unit (GPU) which performs
rendering operations for the display; and

wherein the GPU is switched between a low-power GPU which renders to the first
frame buffer and a high-power GPU which renders to the second frame buffer.

15 8. The method of claim 7, wherein prior to switching from the low-power GPU
to the high-power GPU, the method further comprises substantially synchronizing the low-
power GPU's output display signals and the high-power GPU's output display signals,
thereby facilitating a seamless transition which does not disrupt graphical output on the
display.

20

9. The method of claim 8, wherein substantially synchronizing the output display
signals involves using one or more phase-locked loops (PLL).

10. The method of claim 1, wherein the switching takes place during a blanking
25 interval associated with a blanking signal for the display.

11. An apparatus that selectively switches between frame buffers which are used
to refresh a display, comprising:

a first frame buffer located in a first memory;
30 a second frame buffer located in a second memory;

23

one or more refresh circuits configured to selectively refresh the display from either the first frame buffer or the second frame buffer; and

a switching mechanism configured to switch the refreshing of the display between the first frame buffer and the second frame buffer upon receiving a request to switch.

5

12. The apparatus of claim 11,
wherein the first memory is a main memory, which is accessible by numerous applications and is hence insecure; and

wherein the second memory is a secure frame buffer which is located outside of main
10 memory.

13. The apparatus of claim 12,
wherein the switching circuit is configured to channel data which is used to refresh the display so that the data completely bypasses the insecure main memory; and

15 wherein the apparatus additionally comprises encryption circuitry which is configured to encrypt the data while the data is stored in the second frame buffer and while the data is in transit to and from the second frame buffer.

14. The apparatus of claim 11, wherein the switching mechanism is configured to:
20 determine a security requirement for data associated with the display; and
generate the request to switch frame buffers based on the determined security requirement.

15. The apparatus of claim 11, wherein the switching mechanism is configured to:
25 monitor a level of graphics-processing load for the display; and
generate the request to switch based on the level of graphics-processing load.

16. The apparatus of claim 11, wherein the switching mechanism is configured to:
measure a temperature in a computer system which contains the display; and
30 generate the request to switch based on the measured temperature.

24

17. The apparatus of claim 11,
wherein while switching between the first frame buffer and the second frame buffer,
the apparatus is configured to switch a graphics processing unit (GPU) which performs
rendering operations for the display; and

5 wherein the GPU is switched from a low-power GPU which renders to the first frame
buffer to a high-power GPU which renders to the second frame buffer.

18. The apparatus of claim 17, wherein prior to switching from the low-power
GPU to the high-power GPU, the switching mechanism is configured to substantially
10 synchronize the low-power GPU's output display signals and the high-power GPU's output
display signals, thereby facilitating a seamless transition which does not disrupt graphical
output.

19. The apparatus of claim 18, wherein substantially synchronizing the output
15 display signals involves using one or more phase-locked loops (PLL).

20. The apparatus of claim 11, wherein the switching takes place during a
blanking interval associated with a blanking signal for the display.

21. A computer system that selectively switches between frame buffers which are
used to refresh a display, comprising:

a processor;

a main memory coupled to the processor;

a first frame buffer located in the main memory;

25 a second frame buffer located in a second memory;

one or more refresh circuits configured to selectively refresh the display from either
the first frame buffer or the second frame buffer; and

a switching mechanism configured to switch the refreshing of the display between the
first frame buffer and the second frame buffer upon receiving a request to switch.

30 22. A computer system that switches between a first graphics processor and a

second graphics processor to drive a first display and/or a second display, comprising:
a processor;
a memory;
the first graphics processor;
5 the second graphics processor;
the first display;
the second display;
a first switch which selectively couples either the first graphics processor or the
second graphics processor to the first display; and
10 a second switch which selectively couples either the first graphics processor or the
second graphics processor to the second display.

23. The computer system of claim 22, further comprising a synchronization
mechanism configured to substantially synchronize the first graphics processor's output
15 display signals and the second graphics processor's output display signals, thereby
facilitating a seamless switching process which does not disrupt graphical output.

24. The computer system of claim 22,
wherein the first graphics processor is a high-power graphics processing unit (GPU);
20 and
wherein the second graphics processor is a low-power GPU.

25. The computer system of claim 22,
wherein the first display is an internal display integrated into the computer system;
25 and
wherein the second display is an external display which is coupled to the computer
system.

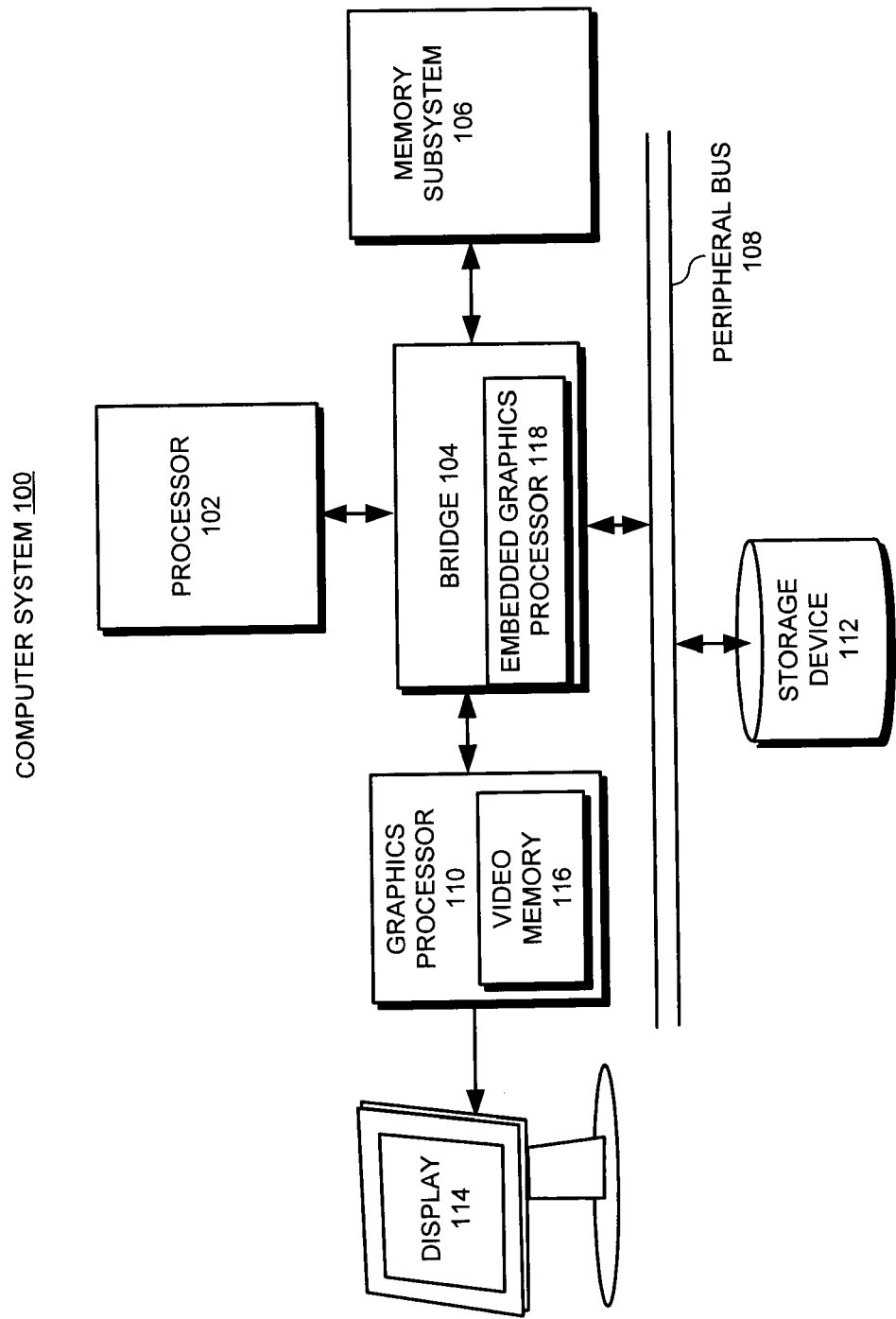
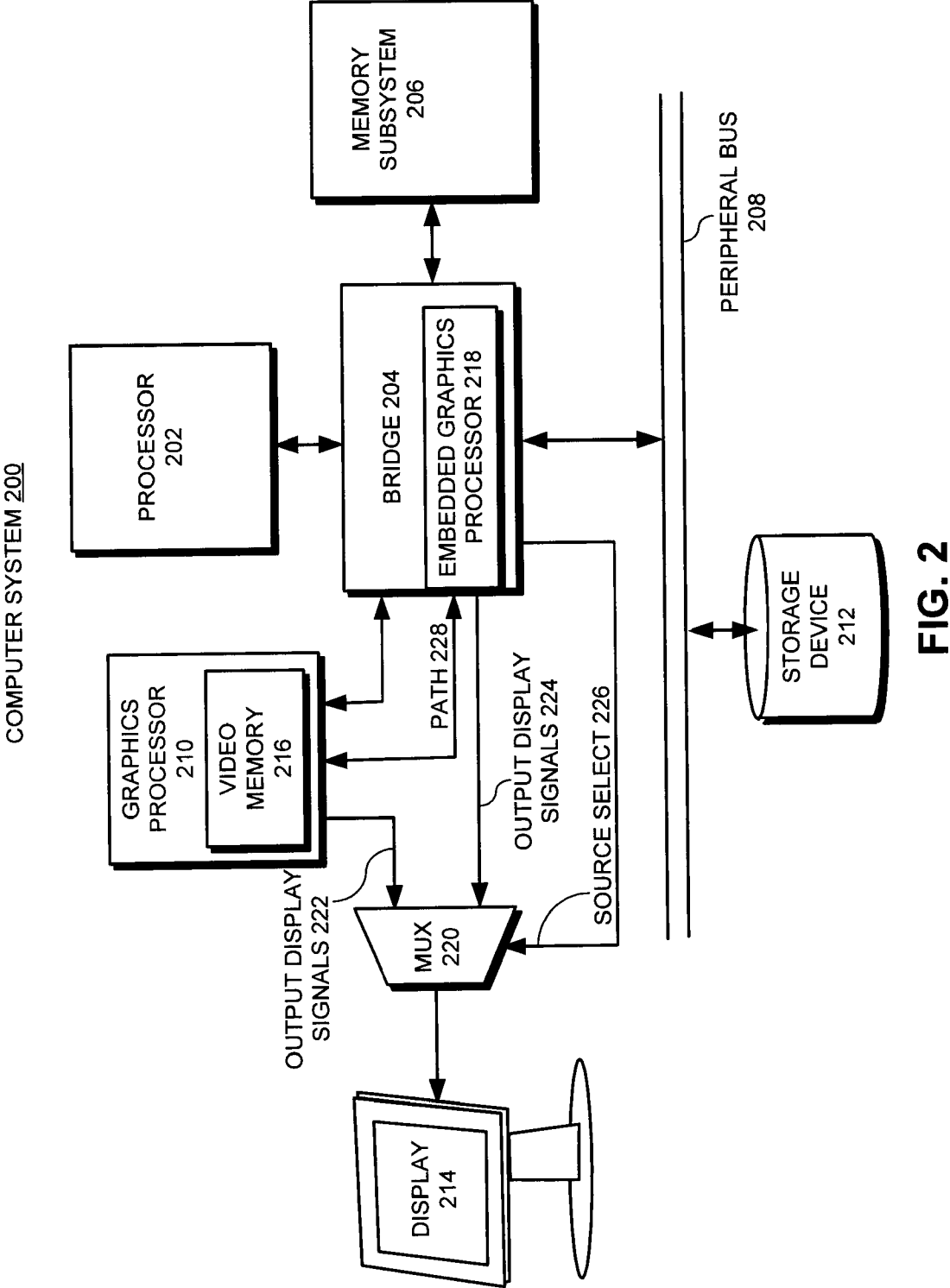
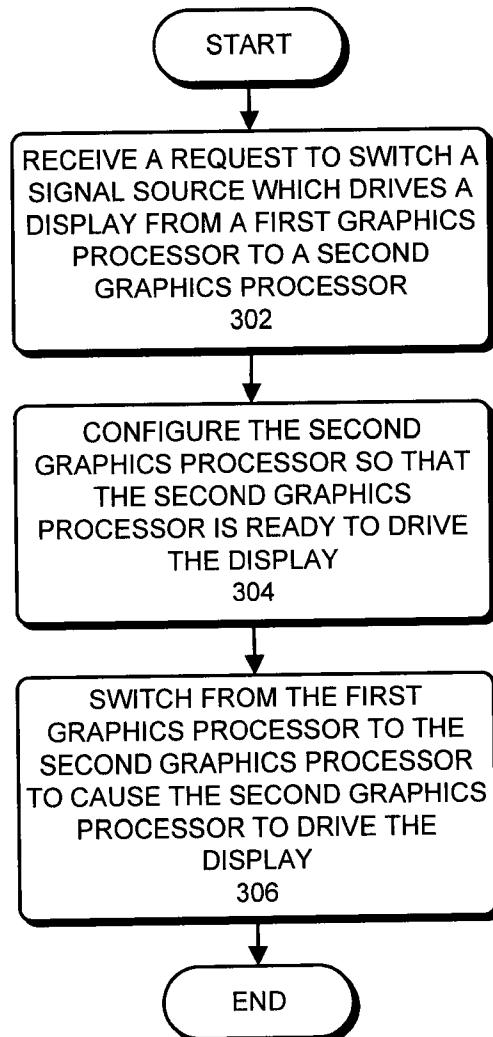
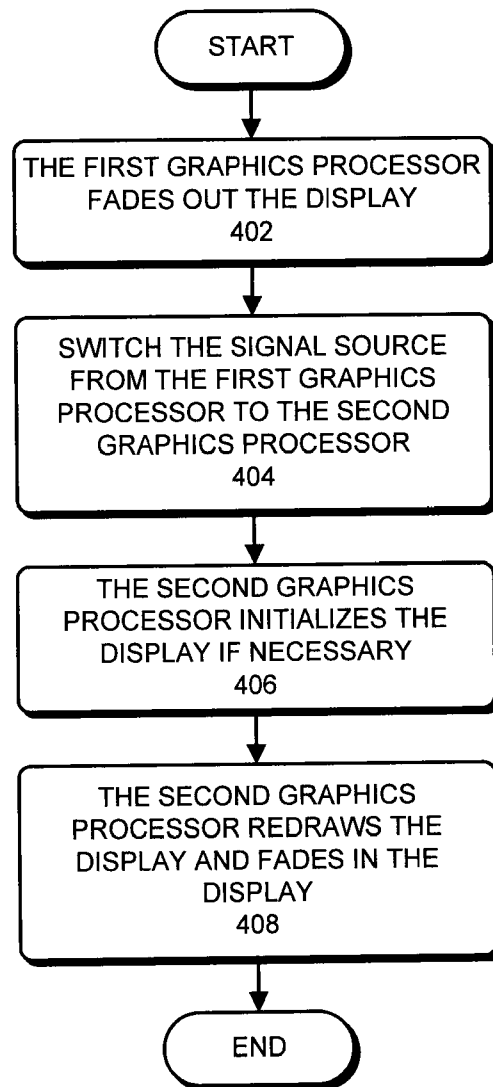
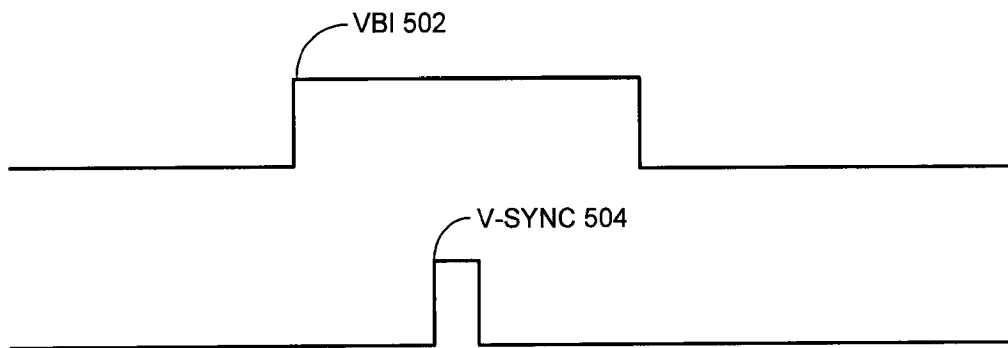
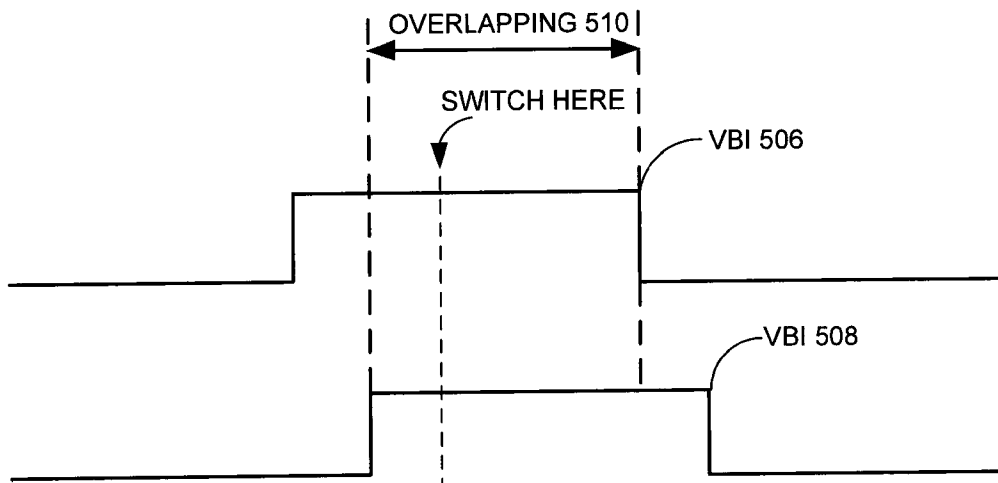


FIG. 1
(PRIOR ART)



**FIG. 3**

**FIG. 4**

**FIG. 5A****FIG. 5B**

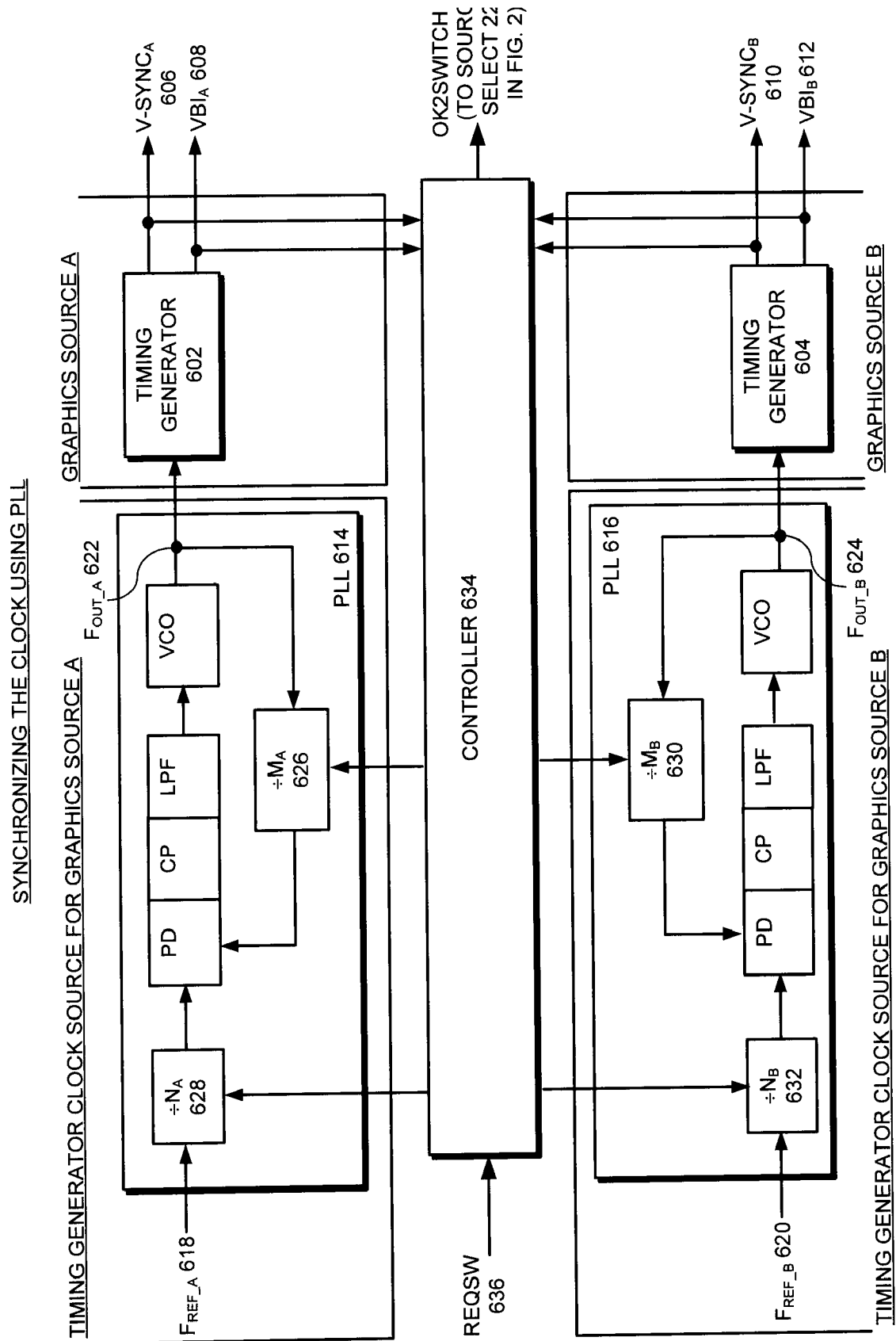


FIG. 6A

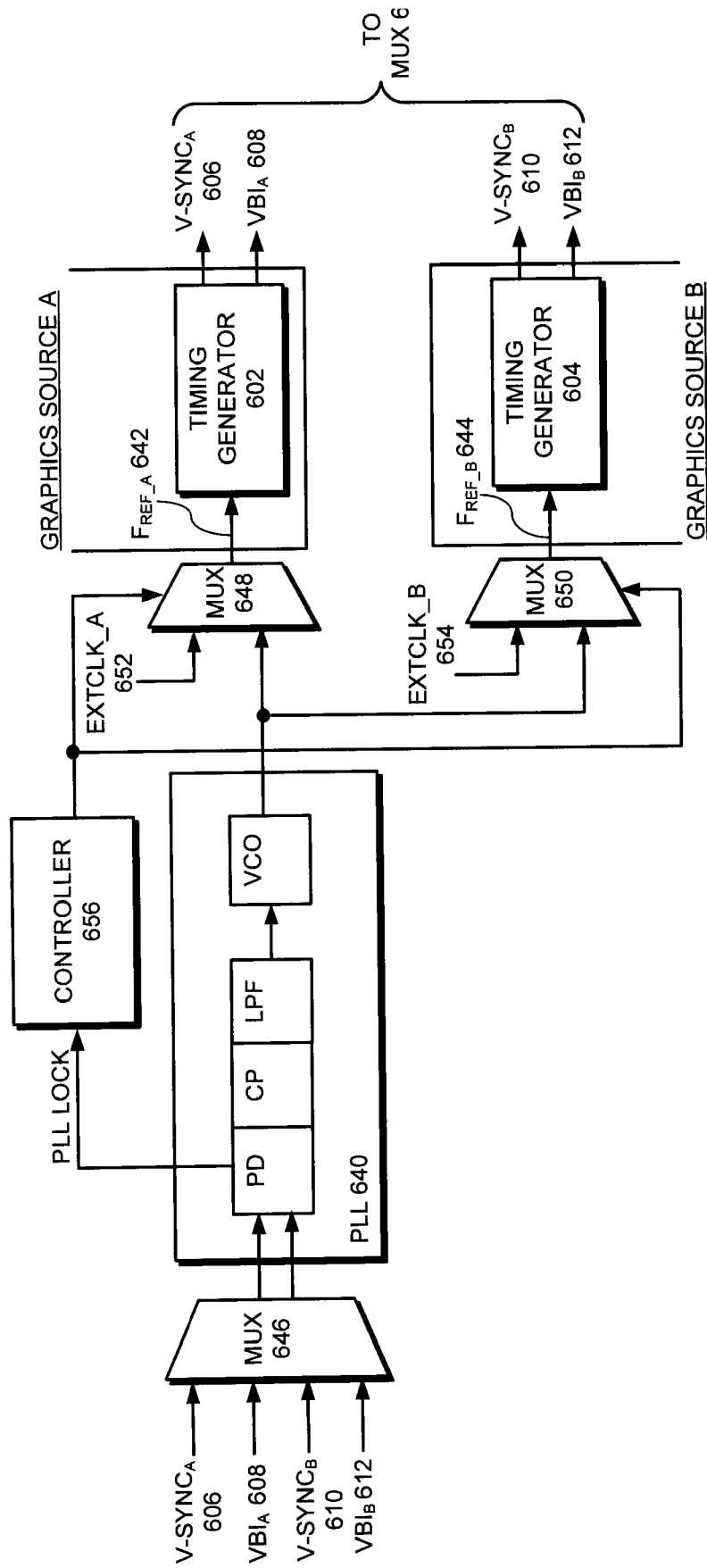


FIG. 6B

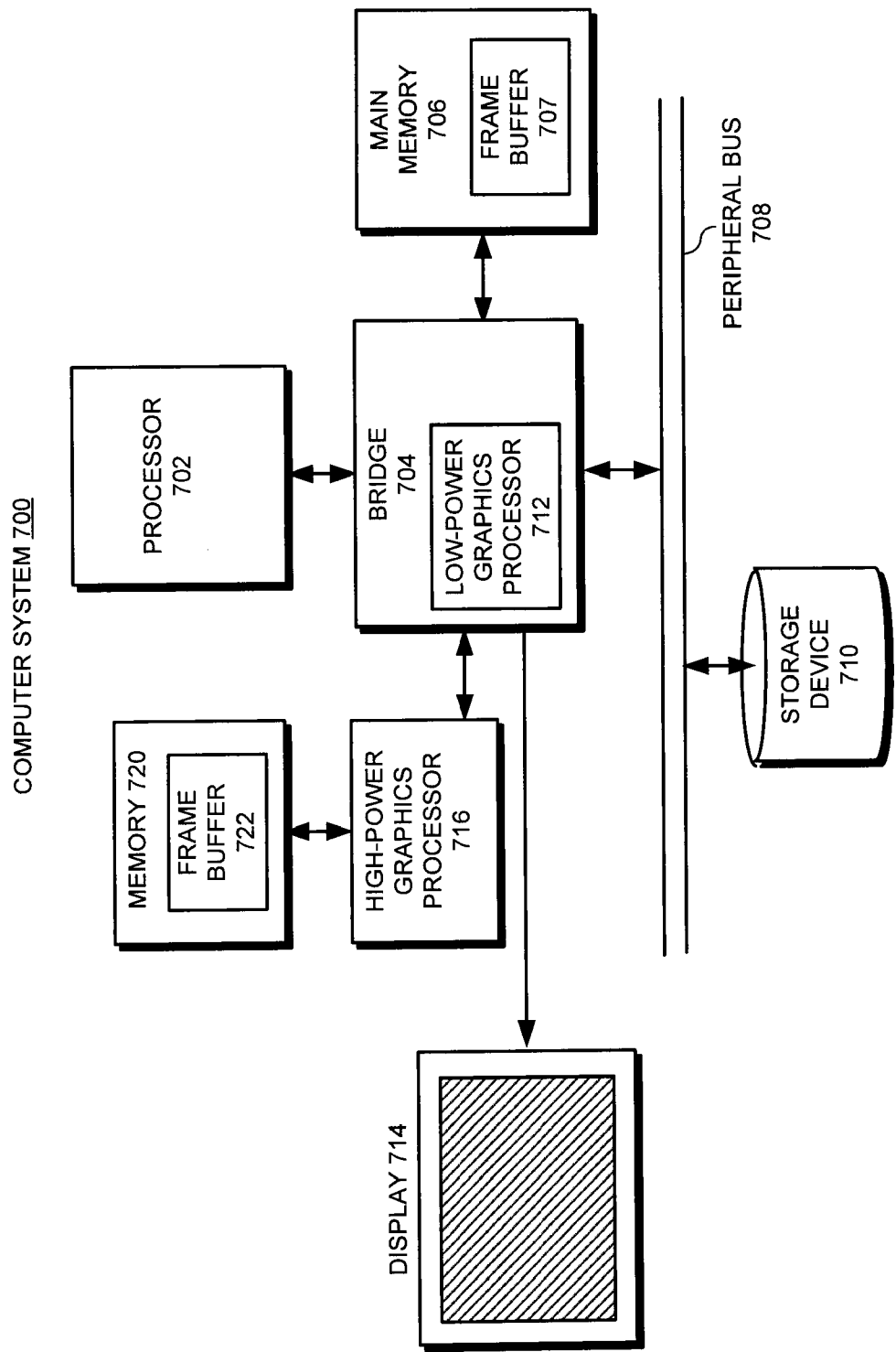


FIG. 7

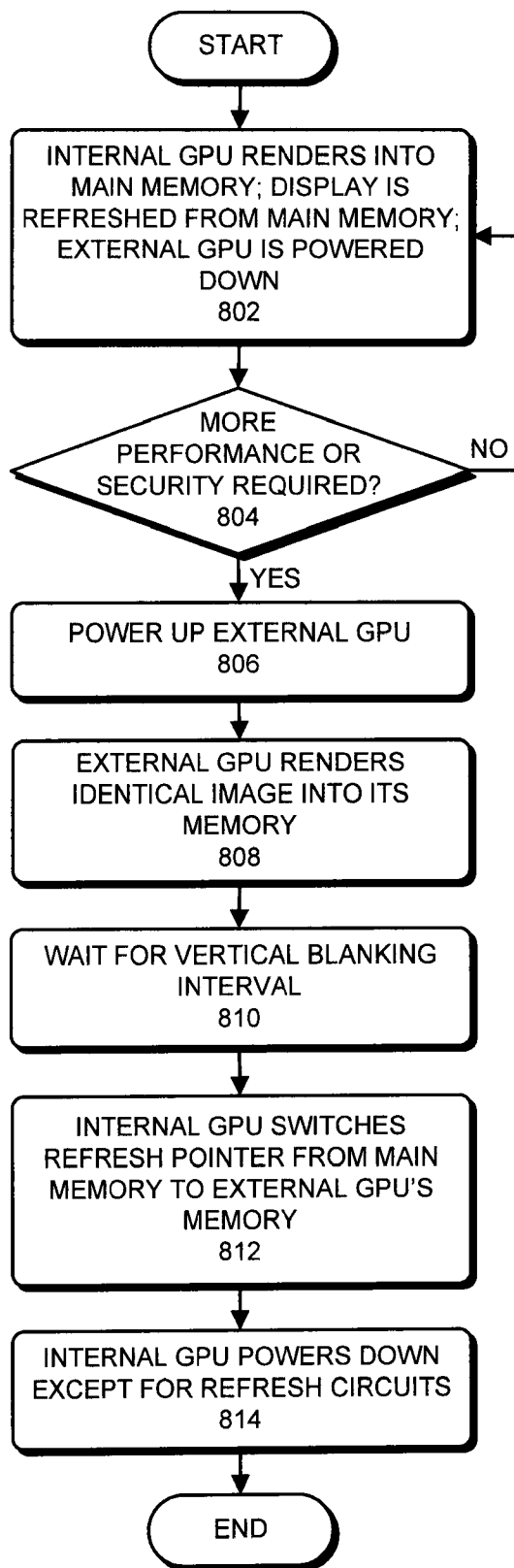


FIG. 8

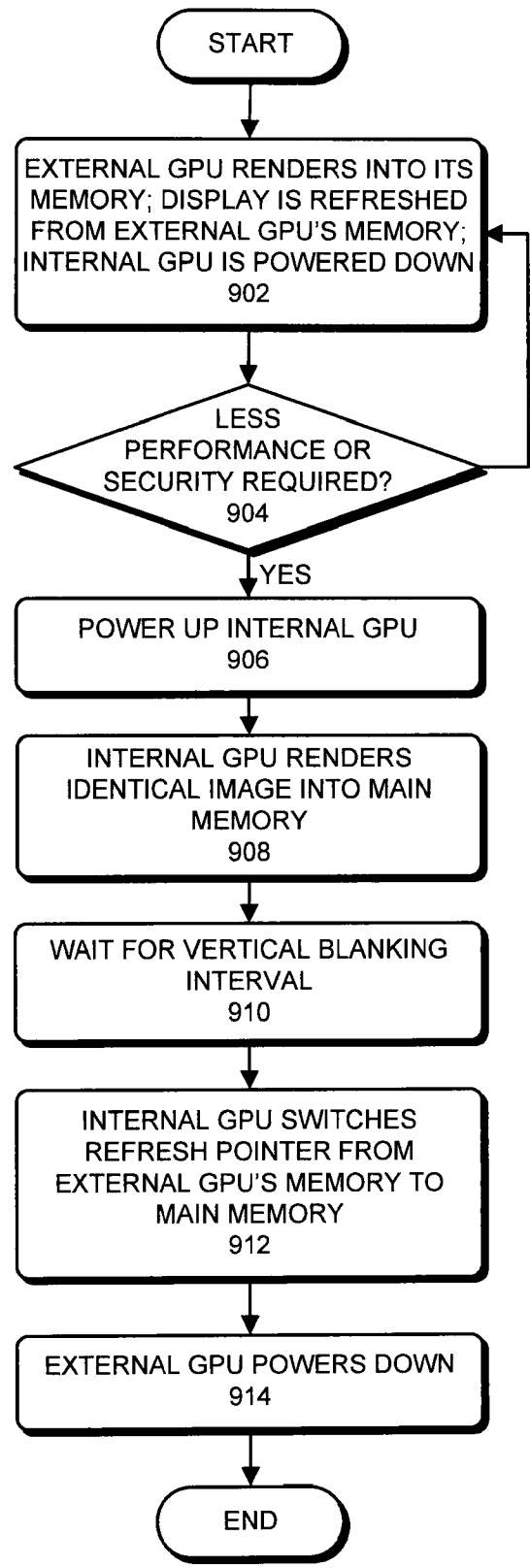
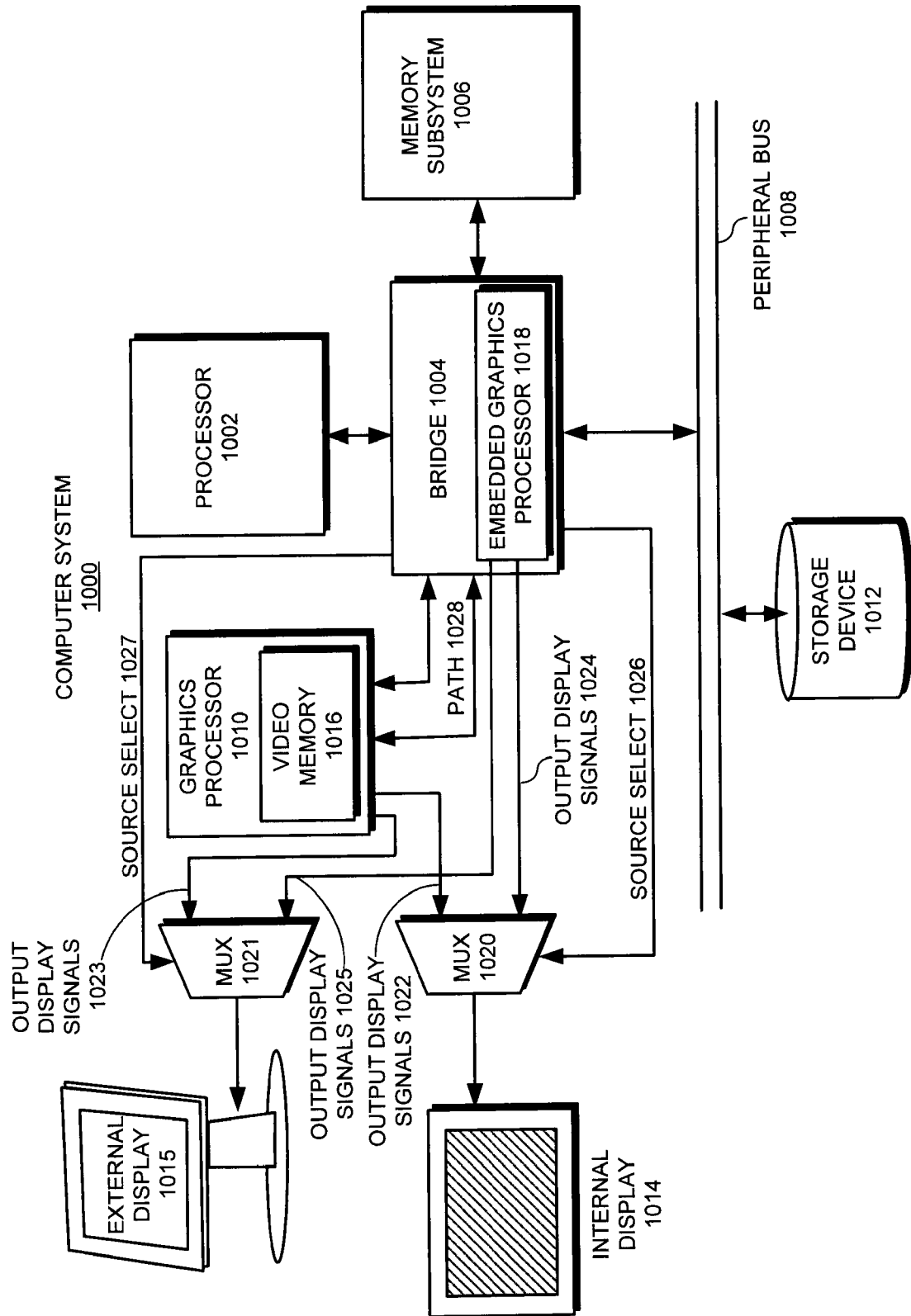


FIG. 9

**FIG. 10**

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2008/072911

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F3/14

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 practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/244131 A1 (UEHARA KEIICHI [JP]) 3 November 2005 (2005-11-03) figures 2,5	1-25
X	US 6 760 031 B1 (LANGENDORF BRIAN K [US] ET AL) 6 July 2004 (2004-07-06) column 3, line 52 - line 67; figures 2,3	1-25
P, X	WO 2007/140404 A (ATI TECHNOLOGIES ULC [CA]; MARINKOVIC SASA [CA]; SALTCHEV ROUMEN [CA];) 6 December 2007 (2007-12-06) figures 4,5	1-25
	----- -/--	

☒ 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 document 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

29 October 2008

Date of mailing of the international search report

05/11/2008

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

Le Chapelain, B

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2008/072911

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DAVID ALLEN: "Electronic Flight Bag (EFB)" 10 October 2006 (2006-10-10), BOEING AEROMAGAZINE, XP002496643 Retrieved from the Internet: URL: http://web.archive.org/web/20061010202846/http://www.boeing.com/commercial/aeromagazine/aero_23/EFB_story.html > [retrieved on 2008-09-22] page 6 - page 7; figure 8 -----	2,3
A	US 2006/007203 A1 (CHEN YU [TW] ET AL) 12 January 2006 (2006-01-12) -----	
A	US 2005/012749 A1 (GONZALEZ NELSON [US] ET AL) 20 January 2005 (2005-01-20) -----	
A	US 2005/225547 A1 (CHOI YOUNG-HUN [KR]) 13 October 2005 (2005-10-13) -----	3

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2008/072911

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 2005244131	A1	03-11-2005	JP 2005316176 A	10-11-2005
US 6760031	B1	06-07-2004	US 2004196289 A1	07-10-2004
WO 2007140404	A	06-12-2007	US 2007283175 A1	06-12-2007
US 2006007203	A1	12-01-2006	DE 202004018289 U1	27-01-2005
			TW 261751 Y	11-04-2005
US 2005012749	A1	20-01-2005	EP 1649357 A1	26-04-2006
			US 2006290700 A1	28-12-2006
			WO 2005010744 A1	03-02-2005
US 2005225547	A1	13-10-2005	CN 1938678 A	28-03-2007
			EP 1735688 A1	27-12-2006
			WO 2005098589 A1	20-10-2005
			KR 20050099305 A	13-10-2005