



(51) International Patent Classification:

G06F 1/00 (2006.01) G06F 13/38 (2006.01)
G06F 1/04 (2006.01)

(21) International Application Number:

PCT/US201 1/067612

(22) International Filing Date:

28 December 2011 (28.12.2011)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant (for all designated States except US): **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, MS: RNB-4-150, Santa Clara, CA 95052 (US).

(72) Inventors; and

(75) Inventors/ Applicants (for US only): **RIFANI, Michael, C** [US/US]; 16730 S.W. Whitetail Ln., Beaverton, OR 97007 (US). **KYKER, Alan, B** [US/US]; 810 Suffolk PL, Winters, CA 95694 (US). **GEIST, Alan, S** [US/US]; 14275 NW Old Germantown Rd, Portland, OR 97231 (US). **LEE, David, M** [US/US]; 2203 NW Crestview Way, Portland, OR 97229 (US).

(74) Agent: **VINCENT, Lester J.**; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, CA 94085-4040 (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, 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, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

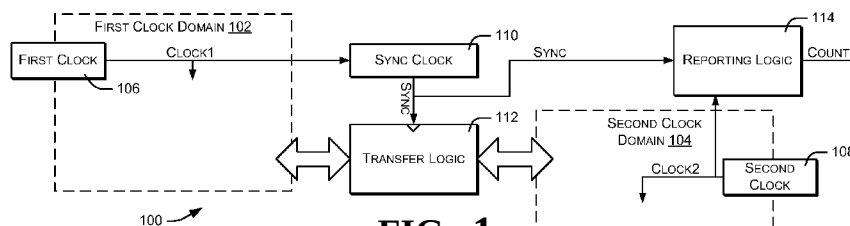
Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

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

(54) Title: DATA TRANSFER BETWEEN ASYNCHRONOUS CLOCK DOMAINS

**FIG. 1**

(57) Abstract: Some implementations disclosed herein provide techniques and arrangements for transferring data between asynchronous clock domains. A synchronization signal may be generated by a first of the clock domains, and data may be transferred between the domains in response to the synchronization signal. Clock cycles of the second of the clock domains may be monitored in comparison to the synchronization signal to report the number of second clock domain cycles occurring per occurrence of the synchronization signal. This information may be recorded by testing and validation equipment to facilitate error analyses.



DATA TRANSFER BETWEEN ASYNCHRONOUS CLOCK DOMAINS

TECHNICAL FIELD

[0001] Some embodiments of the invention relate generally to asynchronous clock domains in systems or devices such as processors. More particularly, some embodiments of the invention relate to system or device configurations that facilitate testing of systems having asynchronous clock domains.

BACKGROUND ART

[0002] Integrated circuits traditionally use synchronous protocols for data transfer. Existing testing and validation technologies rely heavily on cycle-by-cycle, deterministic, synchronous models.

[0003] In a massively parallel architecture or a platform-level design, the number and diversity of interacting clock domains increases. Synchronizing all of the clock domains can be prohibitive because of engineering costs, power consumption, and project-level risks. Accordingly, such architectures and designs increasingly utilize multiple asynchronous clock domains. However, it is difficult to validate or emulate a fully asynchronous architecture with industry-standard validation hardware and software.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The detailed description is set forth with reference to the accompanying drawing figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items or features.

[0005] Fig. 1 is a block diagram illustrating an example of transferring data between asynchronous clock domains according to some implementations.

[0006] Fig. 2 is a timing diagram illustrating timing relationships in the system illustrated by Fig. 1.

[0007] Figs. 3-5 are block diagrams illustrating further examples of transferring data between asynchronous clock domains according to some implementations.

[0008] Fig. 6 is a timing diagram illustrating timing relationships between the SYNC and SYNCK signals of Fig. 5.

[0009] Fig. 7 is a flow diagram illustrating an example process of transferring data between asynchronous clock domains according to some implementations.

[0010] Fig. 8 is a block diagram of an illustrative architecture of a system in which the described techniques may be implemented.

DETAILED DESCRIPTION

[0011] Large-scale integrated circuits and other systems, including processors and other devices, increasingly use multiple asynchronous clock

domains. Asynchronous clock domains often communicate with each other through an intermediate buffer that can be accessed from the different clock domains.

[0012] During development, devices and logic can be monitored to confirm that they operate as intended and produce expected results. However, available testing and diagnostic equipment is typically not designed for operation with asynchronous clock domains. Rather, such testing and diagnostic equipment typically relies on deterministic relationships between components and subsystems. Determinism in this context refers to the certainty of clock cycle on which a processor event occurs. It ensures repeatability of the event, which is fundamental to most high-volume manufacturing experiments involving that event. The non-deterministic nature of asynchronous clock domains presents a challenge, particularly when a developer wants to record and subsequently reproduce conditions and events leading to a detected fault or error.

[0013] The examples described below provide a way for test equipment to record and reproduce the runtime interactions between asynchronous clock domains. In certain embodiments, a first clock domain and a second clock domain transfer data in response to a synchronization signal that is generated synchronously with the clock of the first clock domain. During each period of the synchronization signal, referred to herein as a synchronization cycle, the number of clock cycles in the second clock domain may vary. However, the implementations described below generate a reporting output to indicate the correspondence between synchronization cycles and cycles of the second clock

domain. The reporting output may be recorded by testing equipment, and allows the testing equipment to determine, in response to actual operating conditions, the number of clock cycles that occur in both the first and second clock domains during each synchronization period. This allows subsequent emulation of device operation, and recreation of recorded events.

[0014] Fig. 1 illustrates a device or system 100 having a first clock domain 102 and a second clock domain 104. As an example, the system 100 may be a processor or similar type of device.

[0015] The first clock domain 102 has or is associated with a first clock 106 that generates a first clock signal CLOCK1. The second clock domain 104 has or is associated with a second clock 108 that generates a second clock signal CLOCK2. The first and second clock signals CLOCK1 and CLOCK2 are asynchronous, and therefore may not have deterministic frequency or phase relationships.

[0016] The first and second clock domains 102 and 104 may each have various components and/or elements, not shown, that operate in response to the respective clock signals CLOCK1 and CLOCK2.

[0017] The system 100 may have a synchronization clock 110 that is responsive to the first clock signal CLOCK1 to generate a synchronization signal or pulse SYNC. In some embodiments, the synchronization clock 110 may be implemented as a frequency divider, so that the synchronization pulse SYNC occurs at a lower frequency than that of the first clock signal CLOCK1 while having a fixed phase relationship with the first clock signal CLOCK1. For

example, the synchronization clock 110 may produce a single synchronization pulse corresponding to every N cycles of the first clock signal **CLOCK1**, where N is an integer. The periods defined by the synchronization signal will be referred to herein as synchronization periods or cycles.

[0018] The system 100 may also have data transfer logic 112 to transfer data between the first and second clock domains 102 and 104. The data transfer logic 112 may comprise one or more intermediate registers or buffers, and may be responsive to the synchronization pulse **SYNC** to initiate and/or perform data transfers or exchanges. During an exchange cycle, data may be transferred from the first clock domain 102 to the second clock domain 104, and/or from the second clock domain 104 to the first clock domain 102. A more specific implementation of the transfer logic 112 will be described below, in conjunction with the description of Fig. 4.

[0019] The system 100 may include reporting logic 114 that produces a clock count signal **COUNT** corresponding to each synchronization cycle. The clock count signal **COUNT** may be generated in response to the synchronization pulse **SYNC** and the second clock signal **CLOCK2**. The clock count signal **COUNT** may indicate, for every **SYNC** pulse or corresponding synchronization cycle, the number of occurring cycles of the second clock signal **CLOCK2**. Note that because of the asynchronous relationship between the first and second clock domains, the number of **CLOCK2** cycles occurring between **SYNC** pulses may be indeterminate, and may vary over time. The clock count signal

COUNT indicates the number of actually occurring CLOCK2 cycles corresponding to individual synchronization periods.

[0020] The clock count signal COUNT may be produced and output in synchronization with the first clock signal CLOCK1, the second clock signal CLOCK2, and/or the synchronization pulse SYNC.

[0021] The clock count signal COUNT provides a mechanism for correlating and reporting the runtime operations of the first clock domain 102 and the second clock domain 104, which can be recorded and used by testing and validation equipment to reproduce conditions and sequences that precede error conditions. Because the SYNC pulse is generated from the first clock signal CLOCK1, the number of CLOCK1 cycles per SYNC pulse is known. Because the second clock domain 104 is asynchronous with the first clock domain 102, the number of CLOCK1 cycles per SYNC pulse is potentially variable. However, the reporting logic 114 monitors actual observed performance, and provides the clock count signal COUNT as an indication of how many CLOCK2 cycles actually occur for each SYNC pulse. Thus, the two clock domains can be allowed to run asynchronously, while their operations can be monitored, recorded, and correlated to each other by validation and testing equipment: for every SYNC pulse, it is possible to determine and record the number of corresponding CLOCK1 and CLOCK2 cycles that actually occurred.

[0022] Fig. 2 illustrates signal relationships in the system 100. Note that in this illustration, signals are considered to become active on their rising edges.

Fig. 2 shows relationships between the CLOCK1, SYNC, CLOCK2, and COUNT signals of Fig. 1.

[0023] The CLOCK1 signal may be a cyclical, repetitive, and/or periodic signal such as the square wave illustrated. The synchronization pulse SYNC may similarly be a cyclical, repetitive, or periodic signal, such as a repetitively or periodically occurring pulse. As described above, the synchronization pulse SYNC may be synchronized with the CLOCK1 signal, and may be at a lower frequency than that of the CLOCK1 signal. In the illustrated example, the synchronization pulse SYNC is repeated once for every two cycles of the CLOCK1 signal. More generally, the synchronization pulse SYNC may occur once for every N cycles of the CLOCK1 signal, where N is an integer. Each synchronization pulse SYNC corresponds to a respective synchronization cycle.

[0024] The CLOCK2 signal may be a cyclical, repetitive, and/or periodic signal such as the square wave illustrated. The CLOCK2 signal may have a different frequency than that of the CLOCK1 signal, and may have an indeterminate or variable phase relationship with both the CLOCK1 signal and the SYNC signal.

[0025] In this example, the frequency of the CLOCK2 signal is such that it occurs either once or twice for each repetition, period, or cycle of the SYNC pulse. At the rising edge of each SYNC pulse, the COUNT signal is updated to indicate the actual number of CLOCK2 cycles (indicated by x's in Fig. 2) that occurred during the previous SYNC period, where synchronization periods are defined by the rising edges of the SYNC signal (indicated by dashed vertical

lines in Fig. 2). A low value of the **COUNT** signal in this example corresponds to one cycle of the **CLOCK2** signal, and a high value of the **COUNT** signal corresponds to two cycles of the **CLOCK2** signal. The **COUNT** signal may of course be used to indicate different **CLOCK2** counts, in situations where the nominal frequencies of the **CLOCK1** and **CLOCK2** signals are different than shown. In addition, the **COUNT** signal may in some situations comprise a multi-bit signal or value.

[0026] Fig. 3 illustrates an example device or system 300 in which an intermediate storage element or buffer 302 is used for asynchronously transferring data between a first clock domain 304 and an asynchronous second clock domain 306. The example device or system 300 uses the techniques described above to transfer buffer pointers between the first and second clock domains 304 and 306.

[0027] The intermediate storage buffer 302 may comprise an elastic buffer such as a first-in-first-out (FIFO) buffer. In the given example, the first clock domain 304 writes data to the FIFO buffer 302 in synchronization with a first clock signal **CLOCK1**. The second clock domain 306 reads data from the FIFO buffer in synchronization with a second clock signal **CLOCK2**. The first and second clock signals **CLOCK1** and **CLOCK2** may be asynchronous.

[0028] In order to coordinate writing and reading between the first and second clock domains 304 and 306, write and read pointers are maintained within the first and second clock domains 304 and 306. More specifically, the first clock domain 304 maintains a write pointer 308, indicating the address of

the next position of the FIFO buffer 302 to be written. After the first clock domain 304 writes to this position of the FIFO buffer 302, the write pointer 308 is incremented.

[0029] The second clock domain 306 maintains a read pointer 310, indicating the address of the next position of the FIFO buffer 302 to be read. After the second clock domain 306 reads from this position of the FIFO buffer 302, the read pointer 310 is incremented.

[0030] The first clock domain 304 may also have a shadow or duplicate read pointer 312, which is updated from time to time to reflect the value of the read pointer 310 of the second clock domain 306. To prevent overwriting data that has not yet been read by the second clock domain 306, the first clock domain 304 does not perform writes to locations beyond the address indicated by the shadow read pointer 312.

[0031] Similarly, the second clock domain 306 may have a shadow or duplicate write pointer 314, which is updated from time to time to reflect the value of the write pointer 308 of the first clock domain 304. To avoid reading invalid data, the second clock domain 306 does not perform reads from locations beyond the address indicated by the shadow write pointer 314.

[0032] The system 100 may have a synchronization clock 316, which is configured similarly to the synchronization clock 110 of Fig. 1 to generate a synchronization signal or pulse **SYNC** based on the first clock signal **CLOCKI**. In this embodiment, the **SYNC** pulse is used to update the shadow read pointer 312 and the shadow write pointer 314. More specifically, the **SYNC** pulse is

used to clock or latch data from the read pointer 310 into the shadow read pointer 312, and from the write pointer 308 into the shadow write pointer 314.

[0033] The shadow read pointer 312 may comprise a latch or register that receives the current value of the read pointer 310 from the second clock domain 306. This value is captured by the latch or register 312 upon or in response to receiving the SYNC pulse. Similarly, the shadow write pointer 314 may comprise a latch or register configured to receive the current value of the write pointer 308 from the first clock domain 304. This value is captured by the latch or register 314 upon or in response to receiving the SYNC pulse.

[0034] Similar to the embodiment of Fig. 1, the system 300 may have reporting logic 318 to report the number of CLOCK2 cycles that occur during every repetition or period of the SYNC signal. The reporting logic 318 may generate a clock count signal COUNT to indicate correlation between clock cycles of the first clock domain 304 and clock cycles of the second clock domain 306. As described above, the COUNT signal can be used by testing and validation equipment to record and later emulate conditions and sequences that precede error conditions.

[0035] The shadow read pointer 312 and the shadow write pointer may be clocked directly by the SYNC pulse, or may be clocked in synchronization with the respective clock domains in response to respective SYNC pulses. For example, the read pointer 312 may be clocked by the CLOCK1 signal in response to each SYNC pulse. Similarly, shadow the write pointer 314 may be clocked by the CLOCK2 signal in response to each SYNC pulse.

[0036] Fig. 4 illustrates an example implementation of a system 400 that demonstrates the concepts described above. The system 400 has a first clock domain 402 and a second clock domain 404. The first clock domain 402 operates in response to a first clock signal CLOCK1. The second clock domain 404 operates in response to an asynchronous second clock signal CLOCK2.

[0037] The first and second clock domains 402 and 404 transfer data using an intermediate elastic buffer or FIFO (not shown), which may be similar to the FIFO buffer 302 of Fig. 3. The buffer is addressed by pointers as described with reference to Fig. 3. Specifically, the first clock domain 402 maintains a write pointer 406 and a shadow read pointer 408. The second clock domain 404 maintains a read pointer 410 and a shadow write pointer 412. The shadow write pointer 412 is updated periodically to reflect the value of the write pointer 406. The shadow read pointer 408 is updated periodically to reflect the value of the read pointer 410.

[0038] In this example, a synchronization clock is implemented as a divide-by- N frequency divider 414 within the first clock domain 402. The frequency divider 414 generates SYNC signal as a function of the first clock signal CLOCK1. The SYNC signal may comprise a periodic or repetitive pulse that occurs at a lower frequency than the CLOCK1 signal, while also being synchronous with the CLOCK1 signal.

[0039] The SYNC signal is communicated or propagated from the first clock domain 402, to the second clock domain 404, and then back to the first clock domain 402. More specifically, SYNC signal is transmitted from the first clock

domain 402 to the second clock domain 404 through a first metastable-hardened flip-flop 416. The flip-flop 416 produces a **WRITE SYNC** signal, which may be a delayed version of the **SYNC** signal. The **WRITE SYNC** signal is transmitted from the second clock domain 404 back to the first clock domain 402 through a second metastable-hardened flip-flop 418. The flip-flop 418 produces a **READ SYNC** signal, which may be a delayed version of the **WRITE SYNC** signal.

[0040] In this embodiment, **D Q** latches are used to transfer pointers between the first and second clock domains 402 and 404, in response to the various versions or instances of the synchronization signal, which include the **SYNC** signal, the **WRITE SYNC** signal, and the **READ SYNC** signal.

[0041] A first latch or transfer register 420, within the first clock domain 402, receives at its input the value of the write pointer 406 of the first clock domain 402. A synchronization cycle is initiated by the **SYNC** signal, which latches the write pointer value into the latch 420 so that it can be received and read by the second clock domain 404.

[0042] A second latch or transfer register 422, within the second clock domain 404, receives the write pointer value from the first latch 420, and is responsive to the **WRITE SYNC** signal to latch this value.

[0043] A third latch or transfer register 424, within the second clock domain 404, receives at its input the value of the read pointer 410 of the second clock domain 404. This value is latched into the third latch 424 in response to the

WRITE SYNC signal, so that the value can be received and read by the first clock domain 402.

[0044] A fourth latch or transfer register 426, within the first clock domain 402, receives the read pointer value from the third latch 424, and is responsive to the **READ SYNC** signal to latch the read pointer value from the third latch 424.

[0045] These events can be summarized as the following sequence of actions, which together may be referred to as a synchronization cycle in this embodiment:

- **SYNC** initiates the synchronization cycle and latches the write pointer into the output latch 420;
- **WRITE SYNC** latches the write pointer from the output latch 420 into the input latch 422;
- **WRITE SYNC** also latches the read pointer 410 into the output latch 424; and
- **READ SYNC** latches the read pointer from the output latch 424 into the input latch 426.

[0046] The second clock domain 404 may have reporting logic 428, similar to the reporting logic described above. The reporting logic 428 is responsive to the **WRITE SYNC** signal and to the second clock **CLOCK2**, and produces a **COUNT** output indicating, for each cycle or pulse of **WRITE SYNC**, the number of corresponding **CLOCK2** cycles that occurred during the previous **WRITE SYNC** cycle or synchronization cycle.

[0047] The value of N can be chosen based on the ratio of the first clock signal CLOCK1 and the second clock signal CLOCK2, in a manner that minimizes the potential for data overwrites and/or data starvation. If the synchronization pace defined by the SYNC signal is too fast, the first clock domain may initiate a synchronization cycle before the second clock domain has had a chance to process a previous synchronization cycle. If the synchronization pace defined by the SYNC signal is too slow, the second clock domain may at times be starved for data, even though there is unread data in the FIFO buffer.

[0048] The elements within the dashed box 430 may be considered or referred to as write pointer transfer logic. The elements within the dashed box 432 may be considered or referred to as read pointer transfer logic. The write and read pointer logic 430 and 432 represent an example implementation of the transfer logic 112 shown in Fig. 1.

[0049] Fig. 5 illustrates an example embodiment 500, which is a variation of the embodiment of Fig. 4. In Fig. 5, the write and read pointer logic 430 and 432 are each replicated K times to produce a multi-register write buffer 502 and a multi-register read buffer 504. Each of these buffers may be configured to operate in FIFO fashion, allowing the first and second clock domains 402 and 404 to process synchronization cycles at different rates. For example, this may allow the first clock domain 402 to initiate synchronization cycles at a rate that is faster than the rate at which the second clock domain 404 is able to process the synchronization cycles.

[0050] The embodiment of Fig. 5 includes a K-Phase clock generator 506 that is responsive to the single-bit SYNC signal to create a multi-bit SYNCK signal. The SYNCK signal comprises K signal bits, which are used to clock respective instances of write pointer logic 430 and read pointer logic 432.

[0051] Fig. 6 illustrates the relationship of the individual bits of the SYNCK signal to the single-bit SYNC signal. This example assumes $K=4$. The SYNCK signal comprises K individual bits or signals, referred to as SYNC1, SYNC2, SYNC3, and SYNC4. Each of the individual SYNCK signals is generated by dividing the SYNC signal by K . In addition, the individual SYNCK signals are staggered from each other, so that a single SYNCK signal is generated for each occurrence of the SYNC signal.

[0052] Referring again to Fig. 5, each of the write transfer logic instances 430 may correspond to and be responsive to a different one of the individual SYNCK signal bits. Similarly, each of the read transfer logic instances 432 may be responsive to a different one of the individual SYNCK signal bits. This arrangement effectively implements FIFO logic for the transferred pointers, allowing them to be written and read at different paces by the first and second clock domains 402 and 404.

[0053] Fig. 7 illustrates an example of a method 700 for transferring data between asynchronous clock domains. An action 702 comprises clocking a first clock domain with a first cyclical clock signal. An action 704 comprises clocking a second clock domain with a second cyclical clock signal. As

discussed above, the first and second cyclical clock signals may be asynchronous.

[0054] An action 706 comprises generating a synchronization signal. The synchronization signal may be synchronous with the first clock signal, and may be generated by dividing the frequency of the first clock signal by an integer N . Thus, the first clock signal may have a frequency that is a fixed integer multiple of the synchronization signal.

[0055] An action 708 may comprise propagating the synchronization signal from the first clock domain to the second clock domain, and then from the second clock domain back to the first clock domain.

[0056] An action 710 may comprise aligning the pointers between the first and second clock domains in response to the synchronization signal. A first buffer pointer may be transferred from the first clock domain to the second clock domain in response to propagating the synchronization signal from the first clock domain to the second clock domain. A second buffer pointer may be transferred from the second clock domain to the first clock domain in response to propagating the synchronization signal from the second clock domain back to the first clock domain.

[0057] An action 712 may comprise indicating and/or reporting the correspondence between the synchronization signal and cycles of the second clock signal. For example, the action 710 may comprise indicating, for each occurrence of the synchronization signal, the number of corresponding cycles of the second clock signal.

[0058] FIG. 8 is a block diagram of an illustrative architecture of a system 800 in which the techniques described above may be implemented. The system 800 may include one or more processors 802-1, ..., 802-N (where N is a positive integer ≥ 1), each of which may include one or more processor cores 804-1, ..., 804-M (where M is a positive integer ≥ 1). In some implementations the processor(s) 802 may be a single core processor, while in other implementations, the processor(s) 802 may have a large number of processor cores, each of which may include some or all of the components illustrated in FIG. 8.

[0059] The processor(s) 802 and processor core(s) 804 can be operated, via an integrated memory controller (IMC) 810 in connection with a local interconnect 816, to read and write to a memory 812. The processor(s) 802 and processor core(s) 804 can also execute computer-readable instructions stored in the memory 812 or other computer-readable media. The memory 812 may include volatile and nonvolatile memory and/or removable and non-removable media implemented in any type of technology for storage of information, such as computer-readable instructions, data structures, program modules or other data. Such memory may include, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology. In the case in which there are multiple processor cores 804, in some implementations, the multiple processor cores 804 may share a shared cache 814, which may be accessible via the local interconnect 816.

[0060] Storage 818 may be provided for storing data, code, programs, logs, and the like. The storage 818 may include solid state storage, magnetic disk storage, RAID storage systems, storage arrays, network attached storage, storage area networks, cloud storage, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, or any other medium which can be used to store desired information and which can be accessed by a computing device. Depending on the configuration of the system 800, the memory 812 and/or the storage 818 may be a type of computer readable storage media and may be a non-transitory media.

[0061] In various embodiments, the local interconnect 816 may also communicate with a graphical controller or graphics processing unit 820 to provide graphics processing. Additionally, in some embodiments, the local interconnect 816 may communicate with a system agent 822. The system agent 822 may be in communication with a hub 824, which connects a display engine 826, a PCIe 828, and a DMI 830.

[0062] The memory 812 may store functional components that are executable by the processor(s) 802. In some implementations, these functional components comprise instructions or programs 832 that are executable by the processor(s) 802. The example functional components illustrated in FIG. 8 further include an operating system (OS) 834 to manage operation of the system 800.

[0063] The system 800 may include one or more communication devices 836 that may include one or more interfaces and hardware components for

enabling communication with various other devices over a communication link, such as one or more networks 838. For example, communication devices 836 may facilitate communication through one or more of the Internet, cable networks, cellular networks, wireless networks (e.g., Wi-Fi, cellular) and wired networks. Components used for communication can depend at least in part upon the type of network and/or environment selected. Protocols and components for communicating via such networks are well known and will not be discussed herein in detail.

[0064] The system 800 may further be equipped with various input/output (I/O) devices 840. Such I/O devices 840 may include a display, various user interface controls (e.g., buttons, joystick, keyboard, touch screen, etc.), audio speakers, connection ports and so forth. An interconnect 824, which may include a system bus, point-to-point interfaces, a chipset, or other suitable connections and components, may be provided to enable communication between the processors 802, the memory 812, the storage 818, the communication devices 836, and the I/O devices 840.

[0065] Although the subject matter has been described in language specific to structural features and/or methodological acts, the subject matter defined in the appended claims is not limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims. This disclosure is intended to cover any and all adaptations or variations of the disclosed implementations, and the following claims should not be construed to be limited to the specific

implementations disclosed in the specification. Instead, the scope of this document is to be determined entirely by the following claims, along with the full range of equivalents to which such claims are entitled.

CLAIMS

What is claimed is:

1. A processor, comprising:
first and second clock domains to be clocked by first and second asynchronous clock signals, respectively;
first logic to transfer data between the first and second clock domains in response to a synchronization signal; and
second logic to indicate cycles of the second clock signal that correspond to the synchronization signal.
2. The processor of claim 1, wherein the synchronization signal comprises a single bit.
3. The processor of claim 1, wherein the synchronization signal comprises a repeating pulse.
4. The processor of claim 1, wherein:
the synchronization signal comprises multiple bits; and
the first logic comprises transfer registers corresponding to the multiple bits of the synchronization signal.
5. The processor of claim 1, further comprising an elastic buffer to be accessed asynchronously by the first and second clock domains, wherein the

data exchanged between the first and second clock domains comprises one or more pointers to the elastic buffer.

6. The processor of claim 1, further comprising an elastic buffer to be accessed asynchronously by the first and second clock domains, wherein the data exchanged between the first and second clock domains comprises read and write pointers associated with the elastic buffer.

7. The processor of claim 1, wherein the synchronization signal is to be propagated from the first clock domain to the second clock domain, and then from the second clock domain to the first clock domain.

8. The processor of claim 1, wherein the synchronization signal is to be propagated from the first clock domain to the second clock domain to transfer a buffer write pointer from the first clock domain to the second clock domain, and then from the second clock domain to the first clock domain to transfer a buffer read pointer from the second clock domain to the first clock domain.

9. The processor of claim 1, wherein the synchronization signal is synchronous with the first clock signal.

10. The processor of claim 1, further comprising a frequency divider to generate the synchronization signal in response to the first clock signal.

11. The processor of claim 1, wherein each cycle of the synchronization signal corresponds to a fixed multiple of first clock signal cycles.

12. A system, comprising:

a buffer to transfer data between first and second clock domains, wherein the first and second clock domains are clocked by first and second cyclical clock signals, respectively, and wherein the buffer is indexed by one or more buffer pointers;

a synchronization clock to generate a synchronization signal based at least in part on the first clock signal;

synchronization logic to transfer the one or more buffer pointers between the first and second timing domains in response to the synchronization signal; and

reporting logic to indicate correlation of the second clock signal to the synchronization signal.

13. The system of claim 12, wherein the synchronization signal comprises a single bit.

14. The system of claim 12, wherein the synchronization logic comprises multiple transfer registers.

15. The system of claim 12, wherein the synchronization signal comprises pulses, and wherein the reporting logic is to indicate how many cycles of the second clock signal correspond to each of the pulses.

16. The system of claim 12, wherein the buffer comprises a first-in-first-out buffer, and the buffer pointers comprise read and write pointers.

17. The system of claim 12, wherein the synchronization clock comprises a frequency divider to generate the synchronization signal in response to at least the first clock signal.

18. The system of claim 12, wherein the synchronization signal is to be propagated from the first clock domain to the second clock domain, and then from the second clock domain to the first clock domain.

19. The system of claim 12, wherein the synchronization signal is to be propagated from the first clock domain to the second clock domain to transfer a first buffer pointer from the first clock domain to the second clock domain, and then from the second clock domain to the first clock domain to transfer a second buffer pointer from the second clock domain to the first clock domain.

20. The system of claim 12, wherein the synchronization signal is to be synchronous with the first clock signal.

21. The system of claim 12, wherein each cycle of the synchronization signal is to correspond to a fixed multiple of first clock signal cycles.

22. A method, comprising:
clocking a first clock domain with a first cyclical clock signal;
clocking a second clock domain with a second cyclical clock signal,
wherein the first and second cyclical clock signals are asynchronous;
generating a synchronization signal;
transferring data between the first and second clock domains in response to the synchronization signal; and
reporting correspondence between the synchronization signal and cycles of the second clock signal.

23. The method of claim 22, further comprising generating the synchronization signal based at least in part on the first clock signal.

24. The method of claim 22, further comprising generating the synchronization signal in synchronization with the first clock signal.

25. The method of claim 22, further comprising dividing the first clock signal to produce the synchronization signal.

26. The method of claim 22, wherein the synchronization signal comprises cycles that are synchronous with the first clock signal, and the reporting comprises indicating how many cycles of the second clock signal occur during each cycle of the synchronization signal.

27. The method of claim 22, further comprising:

propagating the synchronization signal from the first clock domain to the second clock domain and back to the first clock domain;

transferring a first buffer pointer from the first clock domain to the second clock domain in response to propagating the synchronization signal from the first clock domain to the second clock domain; and

transferring a second buffer pointer from the second clock domain to the first clock domain in response to propagating the synchronization signal back to the first clock domain.

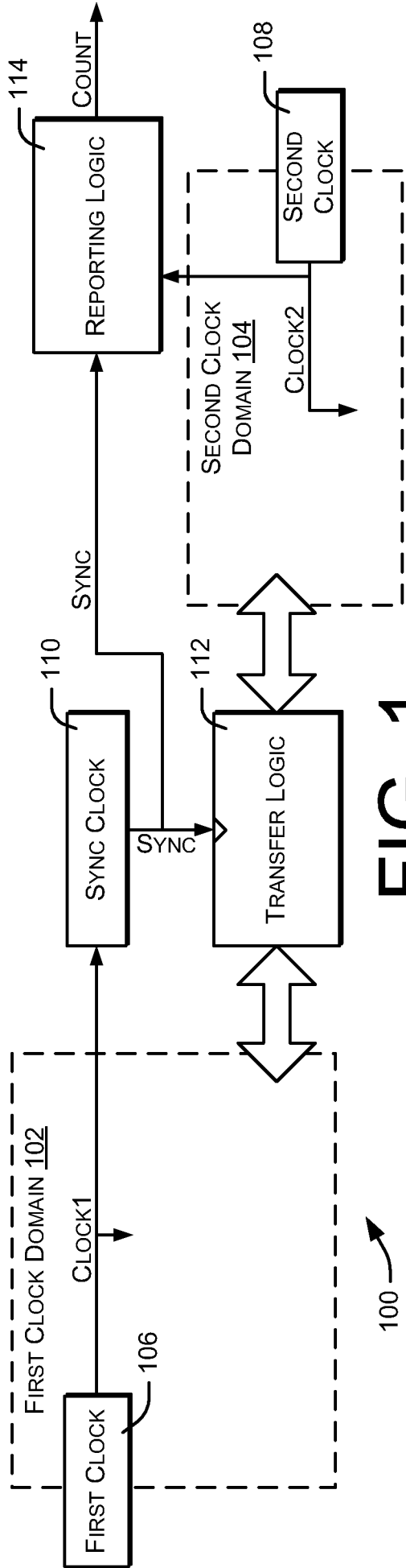


FIG. 1

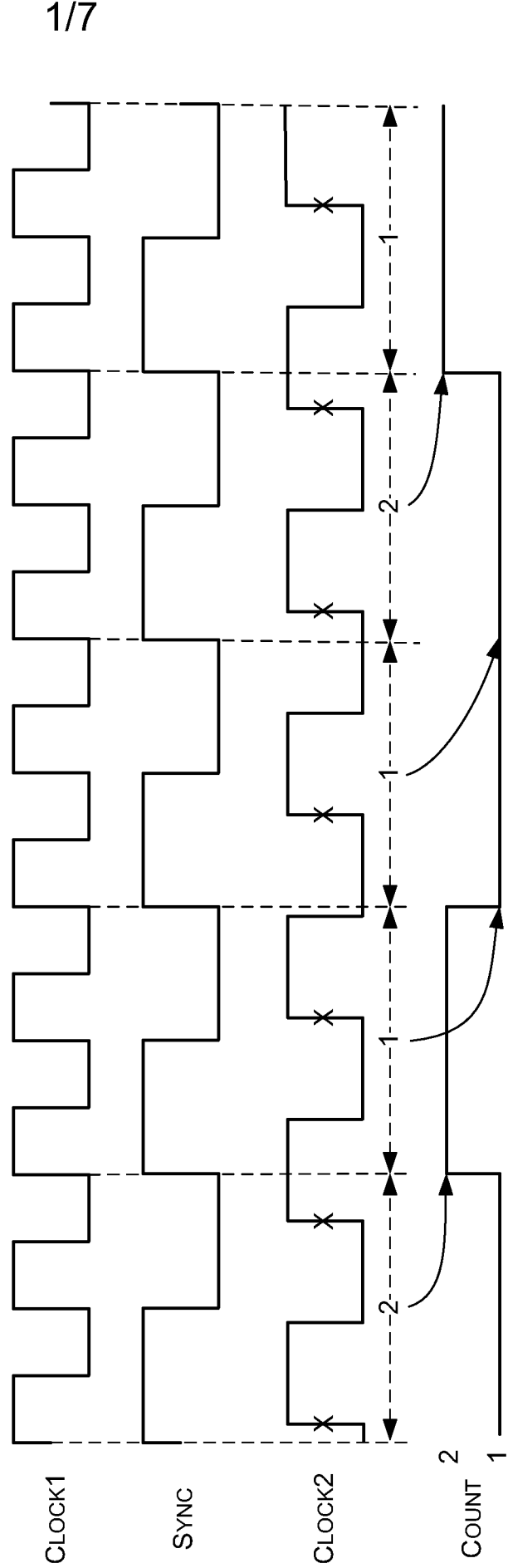


FIG. 2

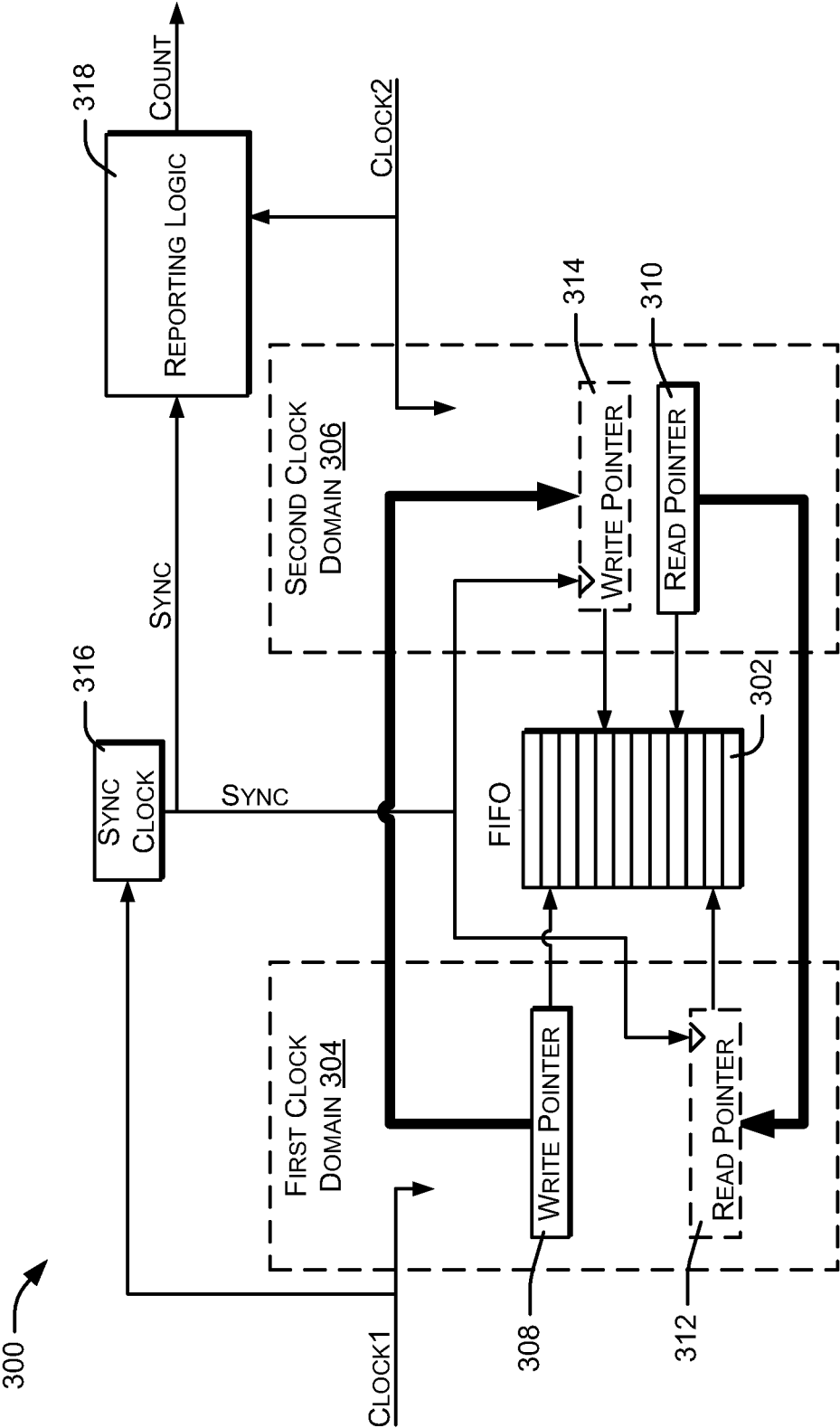


FIG. 3

3/7

400

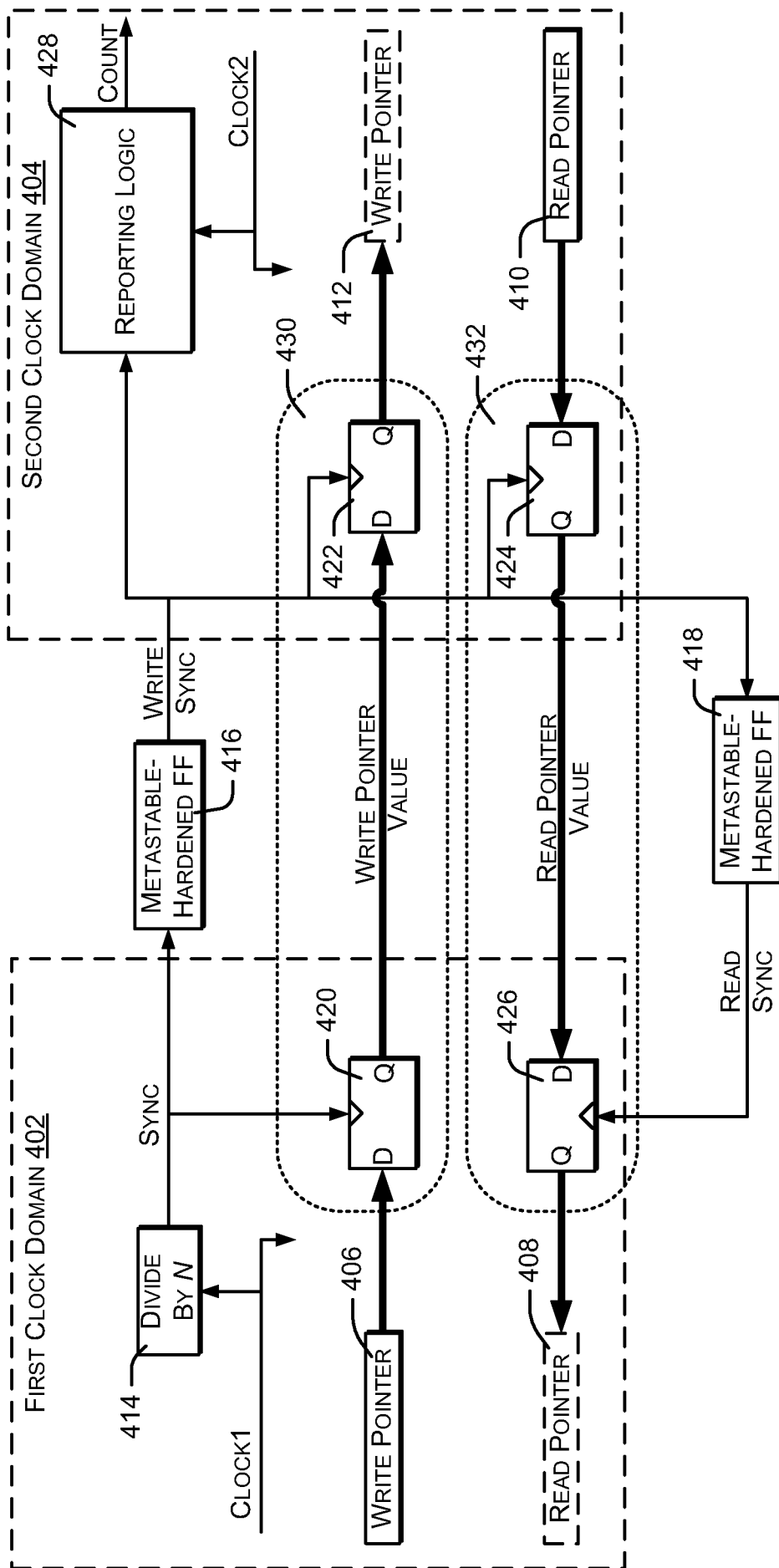


FIG. 4

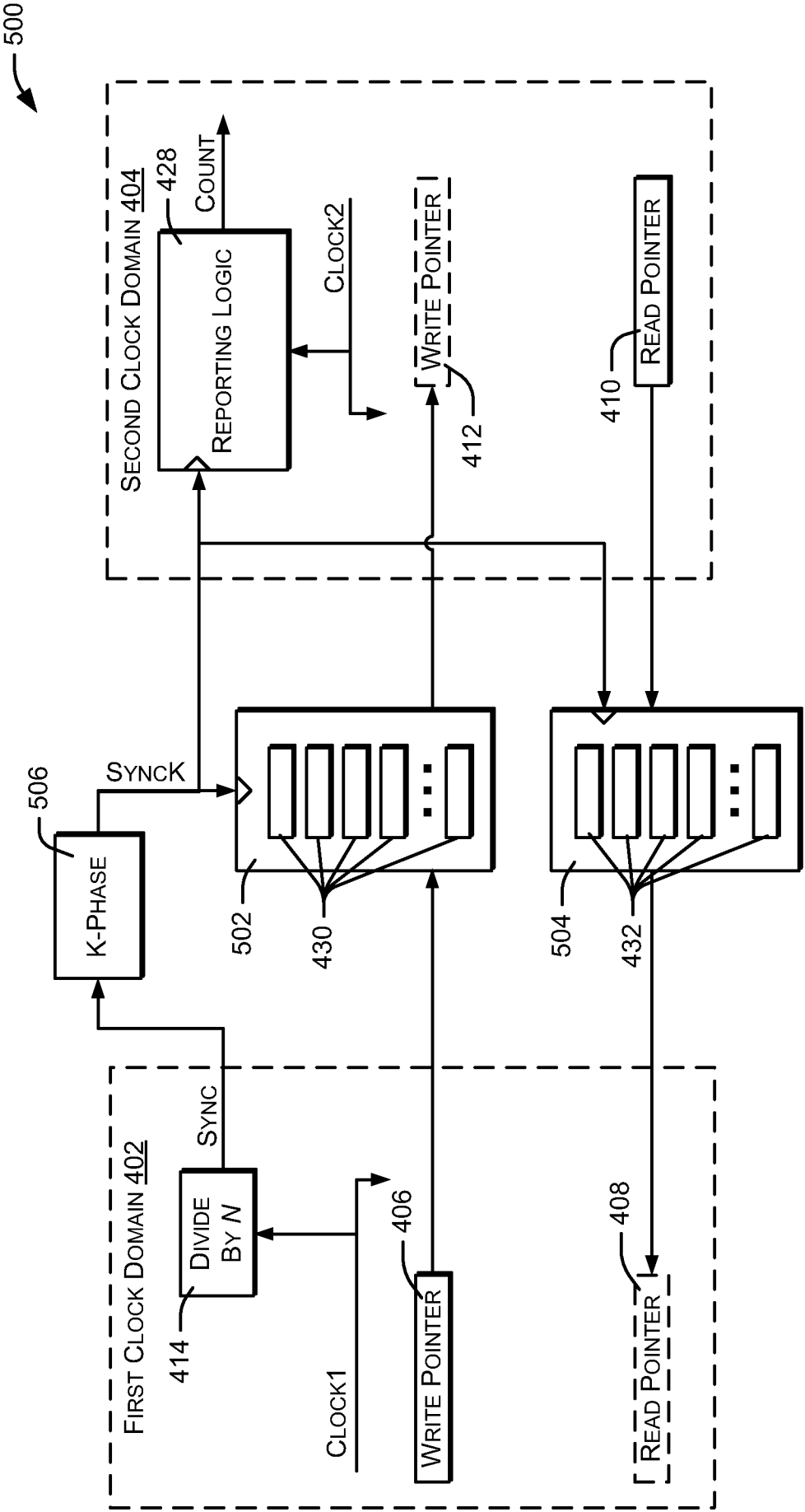


FIG. 5

5/7

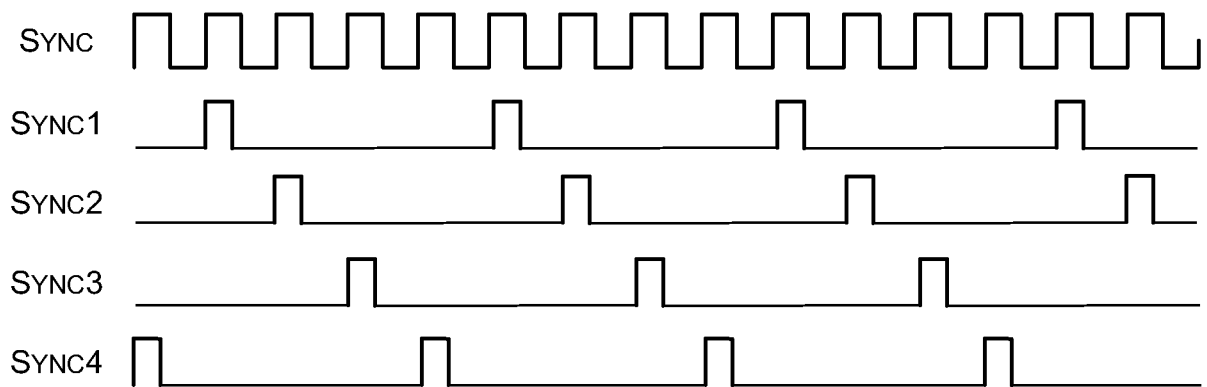


FIG. 6

6/7

700 →

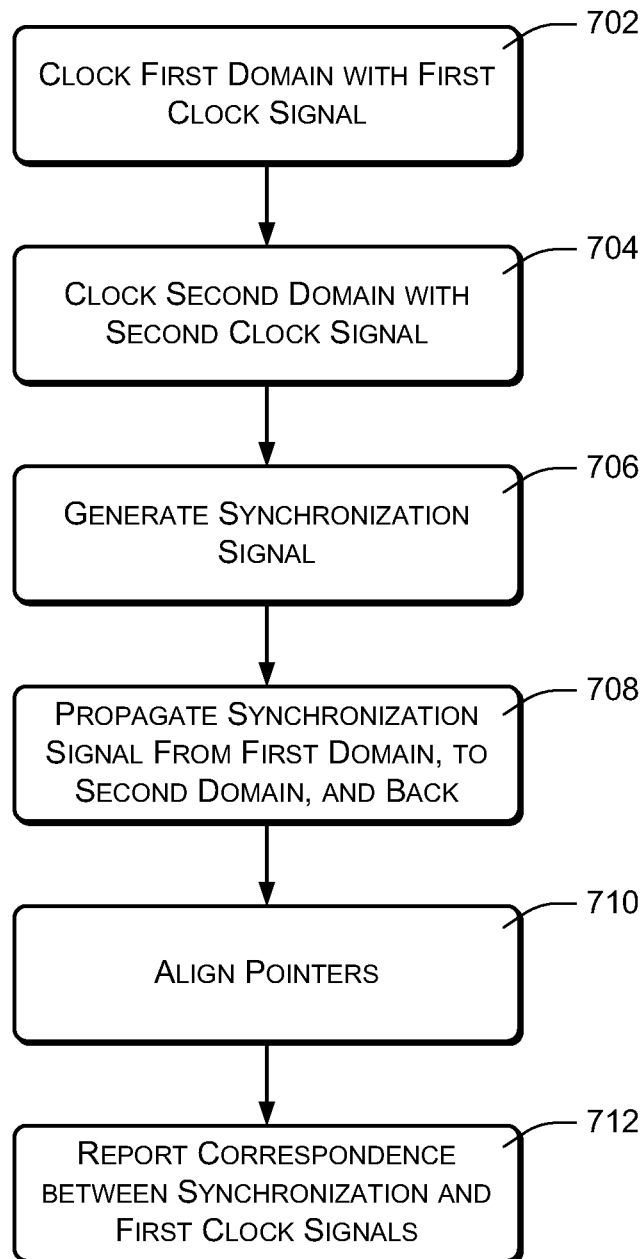


FIG. 7

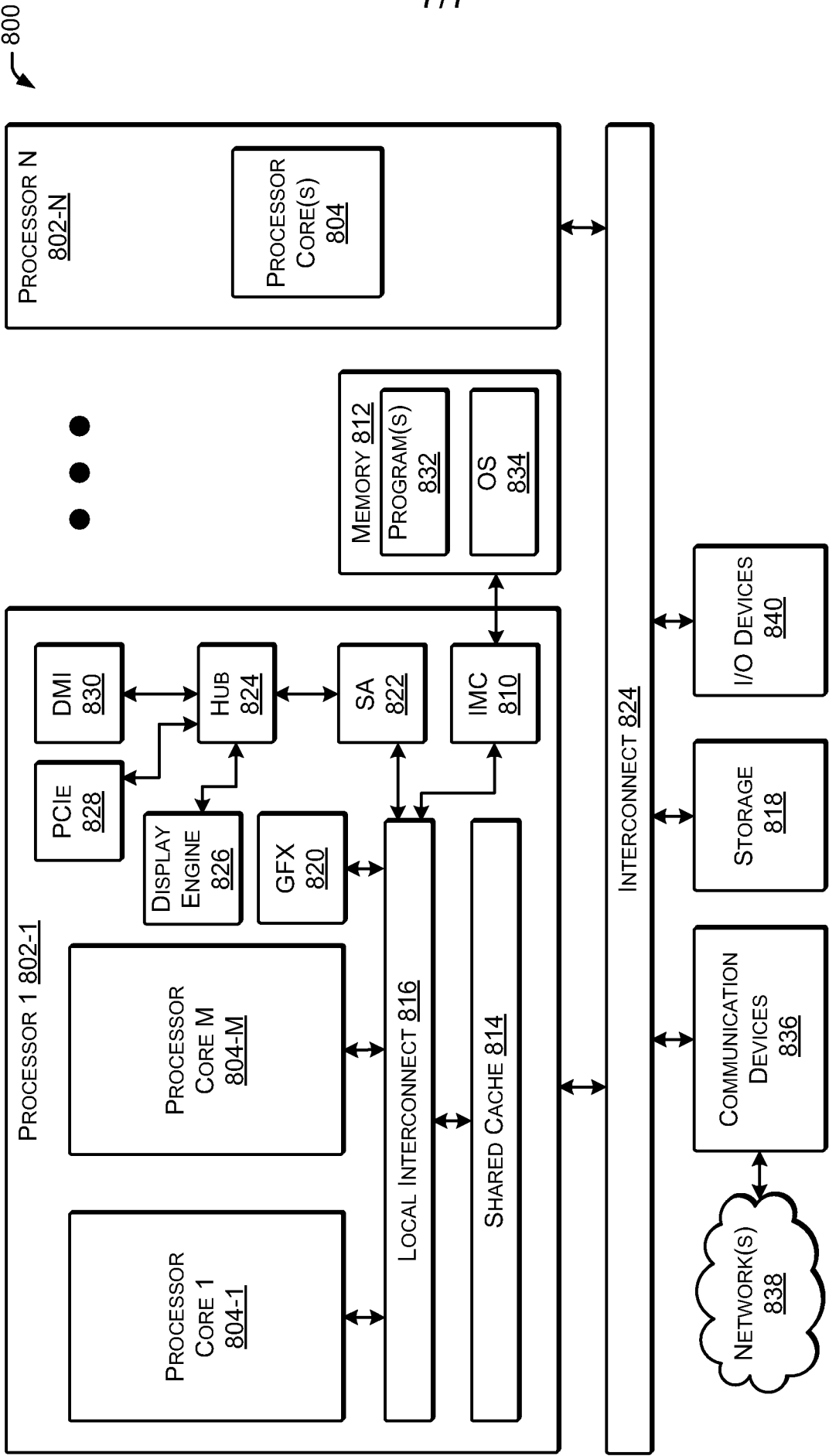


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/067612**A. CLASSIFICATION OF SUBJECT MATTER****G06F 1/00(2006.01)i, G06F 1/04(2006.01)1, G06F 13/38(2006.01)1**

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 1/00; H04L 7/00; G06F 13/40; G01L 15/00; G 11C 8/18

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: "CLOCK", "SYNCHRONIZATION" , "DATA" , "TRANSFER" , "ASYNCHRONOUS"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2003-0123588 A1 (RUPAL PARIKH) 03 July 2003 See abstract, paragraph [0014] - [0048] and figures 1-3	1
Y	US 2003-0105607 A1 (MICHAEL F. JONES et al.) 05 June 2003 See abstract, paragraph [0041] - [0153], claims 15-17 and figure 1	1
A	EP 0772133 A2 (MOTOROLA, INC.) 07 May 1997 See abstract, column 3 line 7 - column 10 line 29 and figure 1	1-27
A	US 2011-0116337 A1 (HAY TIMOTHY NICHOLAS et al.) 19 May 2011 See abstract, paragraph [0053] - [0061] and figure 1	1-27

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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

27 SEPTEMBER 2012 (27.09.2012)

Date of mailing of the international search report

27 SEPTEMBER 2012 (27.09.2012)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan
City, 302-70 1, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

YUN, BYEONG SOO

Telephone No. 82-42-481-8530



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US201 1/067612

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003-0 123588 A1	03.07.2003	US 7 16 1999 B2	09.01.2007
US 2003-0 105607 A1	05.06.2003	us 2005-0102592 A1	12.05.2005
		us 2005-0149800 A1	07.07.2005
		us 7035755 B2	25.04.2006
		us 7043390 B2	09.05.2006
		us 7370255 B2	06.05.2008
		Wo 03-0 16931 A1	27.02.2003
EP 0772 133 A2	07.05.1997	EP 0772 133 A3	22.10.1997
		JP 03-6 16468 B2	02.02.2005
		JP 09- 167134 A	24.06.1997
		us 0578 1765A A	14.07.1998
US 2011-0116337 A1	19.05.2011	None	