(51) International Patent Classification:
G06F 9/46 (2006.01)

(71) Applicant: HUAWEI TECHNOLOGIES CO., LTD. [CN/CN]; Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong 518129 (CN).

(72) Inventors: ZHANG, Qifan; 389 Emile-Pomiville Ave., Lachine, Quebec H8R 0A3 (CA). GE, Yiqun; 944 Fletcher Circle, Ottawa, Ontario K2T 0B7 (CA). SHI, Wuxian; 944 Fletcher Circle, Ottawa, Ontario K2T 0B7 (CA). HUANG, Tao; 296 Goldridge Drive, Ottawa, Ontario K2T 1L1 (CA). TONG, Wen; 12 Whistone Drive, Ottawa, Ontario K2C 4A7 (CA).

(61) Priority Data:
61/874,794 6 September 2013 (06.09.2013) US
61/874,810 6 September 2013 (06.09.2013) US
61/874,856 6 September 2013 (06.09.2013) US
61/874,914 6 September 2013 (06.09.2013) US
61/874,880 6 September 2013 (06.09.2013) US
61/874,889 6 September 2013 (06.09.2013) US
61/874,866 6 September 2013 (06.09.2013) US

(43) International Application Date
12 March 2015 (12.03.2015)

(30) Priority Data:
61/874,794 6 September 2013 (06.09.2013) US
61/874,810 6 September 2013 (06.09.2013) US
61/874,856 6 September 2013 (06.09.2013) US
61/874,914 6 September 2013 (06.09.2013) US
61/874,880 6 September 2013 (06.09.2013) US
61/874,889 6 September 2013 (06.09.2013) US
61/874,866 6 September 2013 (06.09.2013) US


(41) Filing Date:
8 September 2014 (08.09.2014)

(42) Filing Language:
English

(21) International Application Number:
PCT/US20 14/0546 18

(22) International Filing Date:
8 September 2014 (08.09.2014)

(25) Filing Language:
English

(26) Publication Language:
English

(74) Agent: MCCUTCHEON, Robert D.; MUNCK WILSON MANDALA, LLP, 600 Banner Place Tower, 12770 Coit Road, Dallas, Texas 75251 (US).

(54) Title: METHOD AND APPARATUS FOR ASYNCHRONOUS PROCESSOR WITH A TOKEN RING BASED PARALLEL PROCESSOR SCHEDULER

(57) Abstract: A method of operating a clock-less asynchronous processing system comprising a plurality of successive asynchronous processing components. The method comprises providing a first token signal path in the plurality of processing components to allow propagation of a token through the processing components. Possession of the token by one of the processing components enables the processing component to conduct a transaction with a resource component that is shared among the processing components. The method comprises propagating the token from one processing component to another processing component along the token signal path.

Published:
— with international search report (Art. 21(3))
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))
METHOD AND APPARATUS FOR ASYNCHRONOUS PROCESSOR WITH A TOKEN RING BASED PARALLEL PROCESSOR SCHEDULER

TECHNICAL FIELD

[0001] The present disclosure relates generally to asynchronous processors, and more particularly to an asynchronous processor with a token ring based parallel processor scheduler.

BACKGROUND

[0002] High performance synchronous digital processing systems utilize pipelining to increase parallel performance and throughput. In synchronous systems, pipelining results in many partitioned or subdivided smaller blocks or stages and a system clock is applied to registers between the blocks/stages. The system clock initiates movement of the processing and data from one stage to the next, and the processing in each stage must be completed during one fixed clock cycle. When certain stages take less time than a clock cycle to complete processing, the next processing stages must wait – increasing processing delays (which are additive).

[0003] In contrast, asynchronous systems (i.e., clockless) do not utilize a system clock and each processing stage is intended, in general terms, to begin its processing upon completion of processing in the prior stage. Several benefits or features are present with asynchronous processing systems. Each processing stage can have a different processing delay, the input data can be processed upon arrival, and consume power only on demand.

[0004] FIGURE 1 illustrates a prior art Sutherland asynchronous micro-pipeline architecture 100. The Sutherland asynchronous micro-pipeline architecture is one form of asynchronous micro-pipeline architecture that uses a handshaking protocol built by Muller-C elements to control the micro-pipeline building blocks. The architecture 100 includes a plurality of
computing logic 102 linked in sequence via flip-flops or latches 104 (e.g., registers). Control signals are passed between the computing blocks via Muller C-elements 106 and delayed via delay logic 108. Further information describing this architecture 100 is published by Ivan Sutherland in Communications of the ACM Volume 32 Issue 6, June 1989 pages 720-738, ACM New York, NY, USA, which is incorporated herein by reference.

[0005] Now turning to FIGURE 2, there is illustrated a typical section or processing stage of a synchronous system 200. The system 200 includes flip-flops or registers 202, 204 for clocking an output signal (data) 206 from a logic block 210. On the right side of FIGURE 2 there is shown an illustration of the concept of meta-stability. Set-up times and hold times must be considered to avoid meta-stability. In other words, the data must be valid and held during the set-up time and the hold time, otherwise a set-up violation 212 or a hold violation 214 may occur. If either of these violations occurs, the synchronous system may malfunction. The concept of meta-stability also applies to asynchronous systems. Therefore, it is important to design asynchronous systems to avoid meta-stability. In addition, like synchronous systems, asynchronous systems also need to address various potential data/instruction hazards, and should include a bypassing mechanism and pipeline interlock mechanism to detect and resolve hazards.

[0006] Accordingly, there are needed asynchronous processing systems, asynchronous processors, and methods of asynchronous processing that are stable and detect and resolve potential hazards.
SUMMARY

[0007] According to one embodiment, there is provided a method of operating a clock-less asynchronous processing system comprising a plurality of successive asynchronous processing components. The method comprises providing a first token signal path in the plurality of processing components to allow propagation of a token through the processing components. Possession of the token by one of the processing components enables the processing component to conduct a transaction with a resource component that is shared among the processing components. The method comprises propagating the token from one processing component to another processing component along the token signal path.

[0008] In another embodiment, there is provided a clock-less asynchronous processing system. The processing system comprises a plurality of successive asynchronous processing components, each processing component comprising token processing logic configured to receive, hold and pass a token from a given processing component to another processing component. The token processing logic comprises a token signal path in the plurality of processing components to allow propagation of the token through the processing components. Possession of the token by one of the processing components enables the processing component to conduct a transaction with a resource component that is shared among the processing components.
For a more complete understanding of the present disclosure, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, wherein like numbers designate like objects, and in which:

FIGURE 1 illustrates a prior art asynchronous micro-pipeline architecture.

FIGURE 2 is a block diagram illustrating the concept of meta-stability in a synchronous system;

FIGURE 3A illustrates an asynchronous processing system in accordance with disclosed embodiments of the present disclosure;

FIGURE 4 illustrates an example of a token ring architecture in accordance with disclosed embodiments of the present disclosure;

FIGURE 5 illustrates an example of an asynchronous processor architecture in accordance with disclosed embodiments of the present disclosure;

FIGURE 6 illustrates token based pipelining with gating within an ALU in accordance with disclosed embodiments of the present disclosure;

FIGURE 7 illustrates token based pipelining for an inter-ALU token passing system in accordance with disclosed embodiments of the present disclosure;

FIGURE 8 illustrates a block diagram of an exemplary token ring based array architecture in accordance with disclosed embodiments of the present disclosure;

FIGURE 9 illustrates an exemplary embodiment of a token ring based parallel processor asynchronous scheduler in accordance with disclosed embodiments of the present disclosure;

FIGURE 10 illustrates a more detailed view of the token ring based parallel processor asynchronous scheduler of FIGURE 9.
in accordance with disclosed embodiments of the present disclosure;

[0020] FIGURE 11 illustrates an example communication system in which the asynchronous processor and processing system may be utilized; and

[0021] FIGURES 12A and 12B illustrate example devices in which the asynchronous processor and processing system may be utilized.
DETAILED DESCRIPTION

[0022] Asynchronous technology seeks to eliminate the need of synchronous technology for a global clock-tree which not only consumes an important portion of the chip power and die area, but also reduces the speed of the faster parts of the circuit to match the slower parts (i.e., the final clock-tree rate derives from the slowest part of a circuit). To remove the clock-tree (or minimize the clock-tree), asynchronous technology requires special logic to realize a handshaking protocol between two consecutive clock-less processing circuits. Once a clock-less processing circuit finishes its operation and enters into a stable state, a signal (e.g., a "Request" signal) is triggered and issued to its ensuing circuit. If the ensuing circuit is ready to receive the data, the ensuing circuit sends a signal (e.g., an "ACK" signal) to the preceding circuit. Although the processing latencies of the two circuits are different and varying with time, the handshaking protocol ensures the correctness of a circuit or a cascade of circuits.

[0023] Hennessy and Patterson coined the term "hazard" for situations in which instructions in a pipeline would produce wrong answers. A structural hazard occurs when two instructions might attempt to use the same resources at the same time. A data hazard occurs when an instruction, scheduled blindly, would attempt to use data before the data is available in the register file.

[0024] With reference to FIGURE 3A, there is shown a block diagram of an asynchronous processing system 300 in accordance with the present disclosure. The system 300 includes an asynchronous scalar processor 310, an asynchronous vector processor 330, a cache controller 320 and L1/L2 cache memory 340. As will be appreciated, the term "asynchronous processor" may refer to the processor 310, the processor 330, or the processors 310, 330 in combination. Though only one processor 310, 330 is
shown, the processing system 300 may include more than one of these processors. In addition, it will be understood that each processor may include therein multiple CPUs, control units, execution units and/or ALUs, etc. For example, the asynchronous scalar processor 310 may include multiple execution units with each execution unit having a desired number of pipeline stages. In one example, the processor 310 may include sixteen execution units with each execution unit having five stages. Similarly, the asynchronous vector processor 330 may include multiple execution units with each execution unit having a desired number of pipeline stages.

The L1/L2 cache memory 340 may be subdivided into L1 and L2 cache, and may also be subdivided into instruction cache and data cache. Likewise, the cache controller 320 may be functionally subdivided.

Aspects of the present disclosure provide architectures and techniques for a clock-less asynchronous processor architecture that utilizes a token ring based parallel processor scheduler. A token system is a two-dimensional system. Within a functional unit, tokens gate each other to form a closed loop. Across functional units, a token signal is delayed "deliberately" to avoid a structural hazard. A token-based asynchronous processor uses a token system to "emulate" a pipeline to yield instruction-level parallelism (ILP) to preserve the program order, and avoid the data/structural/control hazards.

FIGURE 4 illustrates an example of a token ring architecture 600 as an alternative to the architecture above in FIGURE 1. The components of this architecture are supported by standard function libraries for chip implementation. For example, the token ring architecture 600 comprises a token processing logic unit 610. The token processing logic 610 comprises token-sense-latch-logic 612 and a variable delay chain 614. In some embodiments, the token processing logic unit 610
may also comprise pulse/active generation logic 616. The token processing logic unit 610 may include any suitable circuitry for detecting reception of a token. The token processing logic unit 610 is configured to propagate the token from one processing component to other processing components along a token signal path.

[0028] As described above with respect to FIGURE 1, the Sutherland asynchronous micro pipeline architecture requires the handshaking protocol, which is realized by the non-standard Muller-C elements. In order to avoid using Muller-C elements (as in Figure 1), a series of token processing logic units are used to control the processing of different computing logic (not shown), such as processing units on a chip (e.g., ALUs) or other functional calculation units, or the access of the computing logic to system resources, such as registers or memory. To cover the long latency of some computing logic, the token processing logic unit 610 is replicated to several copies and arranged in a series of token processing logic units as shown at 620. Each token processing logic unit 610 in the series 620 controls the passing of one or more token signals 630 (associated with one or more resources). A token signal 630 passing through the token processing logic units in series 620 forms a token ring 640. The token ring 640 regulates the access of the computing logic (not shown) to the system resource (e.g., memory, register) associated with that token signal. The token processing logic 610 accepts, holds, and passes the token signal 630 between each other in a sequential manner. When the token signal 630 is held by the token processing logic 610, the computing logic associated with that token processing logic is granted the exclusive access to the resource corresponding to that token signal, until the token signal is passed to a next token processing logic in the ring. Holding and passing the token signal concludes the computing logic's access or use of the corresponding resource, and is
referred to herein as consuming the token. Once the token is consumed, it is released by the given token processing logic unit to a subsequent token processing logic unit in the ring.

[0029] FIGURE 5 illustrates an asynchronous processor architecture 3101. The architecture includes a plurality of self-timed (asynchronous) arithmetic and logic units (ALUs) 3122 coupled in parallel in a token ring architecture as described above with respect to FIGURE 4. Each ALU 3122 may correspond to the token processing logic unit 610 of FIGURE 4. The asynchronous processor architecture 3101 also includes a feedback engine 3120 for properly distributing incoming instructions between the ALUs 3122, an instruction/timing history table 3115 accessible by the feedback engine 3120 for determining the distribution of instructions, a register (memory) 3102 accessible by the ALUs 3122, and a crossbar 3124 for exchanging needed information between the ALUs 3122. The history table 3115 is used for indicating timing and dependency information between multiple input instructions to the processor system. Instructions from the instruction cache/memory are received by the feedback engine 3120 which detects or calculates the data dependencies and determines the timing for instructions using the history table 3115. The feedback engine 3120 pre-decodes each instruction to decide how many input operands this instruction requires. The feedback engine 3120 then looks up the history table 3115 to find whether this piece of data is on the crossbar 3124 or on the register file 3102. If the data is found on the crossbar 3124, the feedback engine 3120 calculates which ALU produces the data. This information is tagged to the instruction dispatched to the ALUs 3122. The feedback engine 3120 also updates the history table 3115 accordingly. A more detailed explanation of the asynchronous architecture 3101 is provided in co-pending application entitled "Method and Apparatus for Asynchronous Processor Pipeline and Bypass Passing", attorney
FIGURE 6 illustrates token based pipelining with gating within an ALU, also referred to herein as token based pipelining for an intra-ALU token gating system 2800. The intra-ALU token gating system 2800 comprises a plurality of tokens including a launch token 2802 associated with a start and decode instruction, a register access token 2804 associated with reading values from a register file, a jump token 2806 associated with a program counter jump, a memory access token 2808 associated with accessing a memory, an instruction pre-fetch token 2810 associated with fetching the next instruction, an other resources token 2812 associated with use of other resources, and a commit token 2814 associated with register and memory commit.

Designated tokens are used to gate other designated tokens in a given order of the pipeline. This means that when a designated token passes through an ALU, a second designated token is then allowed to be processed and passed by the same ALU in the token ring architecture. In other words, releasing one token by the ALU becomes a condition to consume (process) another token in that ALU in that given order.

A particular example of a token-gating relationship is illustrated in FIGURE 6. It will be appreciated by one skilled in the art that other token-gating relationships may be used. In the illustrated example, the launch token (L) 2802 gates the register access token (R) 2804, which in turn gates the jump token (PC token) 2806. The jump token 2806 gates the memory access token (M) 2808, the instruction pre-fetch token (F) 2810, and possibly other resource tokens 2812 that may be used. This means that tokens M 2808, F 2810, and other resource tokens 2812 can only be consumed by the ALU after passing the jump token 2806. These tokens gate the commit token (W) 2814 to register or memory. The commit token 2814 is also referred to herein as a
token for writing the instruction. The commit token 2814 in turn gates the launch token 2802. The gating signal from the gating token (a token in the pipeline) is used as input into a consumption condition logic of the gated token (the token in the next order of the pipeline). For example, the launch token (L) 2802 generates an active signal to the register access or read token (R) 2804, when the launch token (L) 2802 is released to the next ALU. This guarantees that any ALU would not read the register file until an instruction is actually started by the launch token 2802.

[0033] FIGURE 7 illustrates token based pipelining for an inter-ALU token passing system 2900. The inter-ALU token passing system 2900 comprises a first ALU 2902 and a second ALU 2904. A consumed token signal triggers a pulse to a common resource. For example, the register read token 2804 in the first CPU 2902 triggers a pulse to the register file (not shown). The token signal is delayed before it is released to the next ALU (e.g., the second ALU 2904) for a period of time such that there is no structural hazard on this common resource (e.g., the register file) between the first ALU 2902 and the second ALU 2904. The tokens not only preserve multiple ALUs from launching and committing (or writing) instructions in the program counter (PC) order, but also avoid structural hazard among the multiple ALUs.

[0034] FIGURE 8 illustrates a block diagram of an exemplary token ring based array architecture 2700. As illustrated, the token ring based array architecture 2700 comprises a plurality of processing units 2702, a token signal path or ring 2704 comprising a plurality of tokens, a multiplexor 2706, and a plurality of external resources 2708 shared between the processing units 2702. In the illustrated example, the processing units 2702 are identical in design and function to one another. In a non-limiting example, the processing units 2702
implement arithmetic and logic units (ALUs). The ALUs 2702 may be asynchronous units.

The token ring 2704 allows propagation of a token through the ALUs 2702. Token processing logic is provided (not shown) for propagating the token from one ALU to other ALUs amongst the ALUs 2702 along the token ring 2704. The token processing logic is configured to propagate the token between the ALUs 2708 at a propagation rate that is related to a transaction rate of the shared external resource 2708. For example, the rate at which the ALU completes a transaction may vary depending on the specific transaction requested.

Each token in the token ring 2704 is a signal indicator for the availability of one or more of the external resources 2708. The token is such that only one ALU amongst the ALUs 2702 can possess it at any given time. In a specific example of implementation, possession of the token by a given ALU enables the given ALU to conduct a transaction with the shared external resource 2708. Conversely, lack of possession of the token by the given ALU prevents the given ALU from conducting a transaction with the shared external resource 2708. In this manner, the token allows preventing more than one ALU from conducting a transaction with the external resource 2708 at a given time. After a given ALU conducts a transaction with the shared external resource 2708, or if the given ALU does not wish to conduct a transaction with the shared external resource 2708, the ALU releases or "passes" the token to the next ALU. Serialized in this way, multiple ALUs can share a common external resource. As illustrated, multiple tokens may be required to control access to the shared external resources 2708 via an N-bit selection control signal 2712 and the multiplexor 2706.

FIGURE 9 illustrates an exemplary embodiment of a token ring based parallel processor asynchronous scheduler 3000. As illustrated, a multiple token ring 3010 similar to the token ring
2704 described above with respect to FIGURE 8 is used to control access of different external common resources 2708 between a first ALU (e.g., ALU 0) 2902, a second ALU (e.g., ALU 1) 2904, etc. In addition, token dependency and gating similar to the intra-token gating system 2800 described above with respect to FIGURE 28 is used to form a pipeline with different stages within a given ALU. By using a multiple token ring to control access of different external common resources and token dependency and gating to form a pipeline with different stages, multiple asynchronous ALUs can be combined as a parallel processor 3030. As a result, natural pipeline stages may be formed unlike a synchronous processor that has fixed period pipeline stages.

[0038] FIGURE 10 illustrates a more detailed view of the token ring based parallel processor asynchronous scheduler of FIGURE 9, where token ring signal paths of the tokens are illustrated by "dashed" lines, and where token dependence signal paths of the tokens are illustrated by "solid" lines. For example, inter-ALU token passing as described above with respect to FIGURE 7 is illustrated by the launch token 2802 being passed from ALU 0 2902 to ALU 1 2904 via token ring signal path 3104. In addition, intra-ALU token passing as described above with respect to FIGURE 6 is illustrated by launch token dependency signal 3106 from the launch token 2802 gating the register access token 2804. The other tokens (e.g., the register access token (R) 2804, the jump token (PC token) 2806, etc.) may be similarly passed between the ALUs and within the ALUs, respectively.

[0039] FIGURE 11 illustrates an example communication system 1400 that may be used for implementing the devices and methods disclosed herein. In general, the system 1400 enables multiple wireless users to transmit and receive data and other content. The system 1400 may implement one or more channel access methods, such as code division multiple access (CDMA), time division multiple access (TDMA), frequency division multiple access
FDMA), orthogonal FDMA (OFDMA), or single-carrier FDMA (SC-FDMA).

[0040] In this example, the communication system 1400 includes user equipment (UE) 1410a-1410c, radio access networks (RANs) 1420a-1420b, a core network 1430, a public switched telephone network (PSTN) 1440, the Internet 1450, and other networks 1460. While certain numbers of these components or elements are shown in FIGURE 14, any number of these components or elements may be included in the system 1400.

[0041] The UEs 1410a-1410c are configured to operate and/or communicate in the system 1400. For example, the UEs 1410a-1410c are configured to transmit and/or receive wireless signals or wired signals. Each UE 1410a-1410c represents any suitable end user device and may include such devices (or may be referred to) as a user equipment/device (UE), wireless transmit/receive unit (WTRU), mobile station, fixed or mobile subscriber unit, pager, cellular telephone, personal digital assistant (PDA), smartphone, laptop, computer, touchpad, wireless sensor, or consumer electronics device.

[0042] The RANs 1420a-1420b here include base stations 1470a-1470b, respectively. Each base station 1470a-1470b is configured to wirelessly interface with one or more of the UEs 1410a-1410c to enable access to the core network 1430, the PSTN 1440, the Internet 1450, and/or the other networks 1460. For example, the base stations 1470a-1470b may include (or be) one or more of several well-known devices, such as a base transceiver station (BTS), a Node-B (NodeB), an evolved NodeB (eNodeB), a Home NodeB, a Home eNodeB, a site controller, an access point (AP), or a wireless router, or a server, router, switch, or other processing entity with a wired or wireless network.

[0043] In the embodiment shown in FIGURE 11, the base station 1470a forms part of the RAN 1420a, which may include other base stations, elements, and/or devices. Also, the base station 1470b
forms part of the RAN 1420b, which may include other base stations, elements, and/or devices. Each base station 1470a-1470b operates to transmit and/or receive wireless signals within a particular geographic region or area, sometimes referred to as a "cell." In some embodiments, multiple-input multiple-output (MIMO) technology may be employed having multiple transceivers for each cell.

[0044] The base stations 1470a-1470b communicate with one or more of the UEs 1410a-1410c over one or more air interfaces 1490 using wireless communication links. The air interfaces 1490 may utilize any suitable radio access technology.

[0045] It is contemplated that the system 1400 may use multiple channel access functionality, including such schemes as described above. In particular embodiments, the base stations and UEs implement LTE, LTE-A, and/or LTE-B. Of course, other multiple access schemes and wireless protocols may be utilized.

[0046] The RANs 1420a-1420b are in communication with the core network 1430 to provide the UEs 1410a-1410c with voice, data, application, Voice over Internet Protocol (VoIP), or other services. Understandably, the RANs 1420a-1420b and/or the core network 1430 may be in direct or indirect communication with one or more other RANs (not shown). The core network 1430 may also serve as a gateway access for other networks (such as PSTN 1440, Internet 1450, and other networks 1460). In addition, some or all of the UEs 1410a-1410c may include functionality for communicating with different wireless networks over different wireless links using different wireless technologies and/or protocols.

[0047] Although FIGURE 11 illustrates one example of a communication system, various changes may be made to FIGURE 11. For example, the communication system 1400 could include any number of UEs, base stations, networks, or other components in
any suitable configuration, and can further include the EPC illustrated in any of the figures herein.

[0048] FIGURES 12A and 12B illustrate example devices that may implement the methods and teachings according to this disclosure. In particular, FIGURE 12A illustrates an example UE 1410, and FIGURE 12B illustrates an example base station 1470. These components could be used in the system 140A or in any other suitable system.

[0049] As shown in FIGURE 12A, the UE 1410 includes at least one processing unit 1500. The processing unit 1500 implements various processing operations of the UE 1410. For example, the processing unit 1500 could perform signal coding, data processing, power control, input/output processing, or any other functionality enabling the UE 1410 to operate in the system 1400. The processing unit 1500 also supports the methods and teachings described in more detail above. Each processing unit 1500 includes any suitable processing or computing device configured to perform one or more operations. Each processing unit 1500 could, for example, include a microprocessor, microcontroller, digital signal processor, field programmable gate array, or application specific integrated circuit. The processing unit 1500 may be an asynchronous processor as described herein.

[0050] The UE 1410 also includes at least one transceiver 1502. The transceiver 1502 is configured to modulate data or other content for transmission by at least one antenna 1504. The transceiver 1502 is also configured to demodulate data or other content received by the at least one antenna 1504. Each transceiver 1502 includes any suitable structure for generating signals for wireless transmission and/or processing signals received wirelessly. Each antenna 1504 includes any suitable structure for transmitting and/or receiving wireless signals. One or multiple transceivers 1502 could be used in the UE 1410, and one or multiple antennas 1504 could be used in the UE 1410.
Although shown as a single functional unit, a transceiver 1502 could also be implemented using at least one transmitter and at least one separate receiver.

[0051] The UE 1410 further includes one or more input/output devices 1506. The input/output devices 1506 facilitate interaction with a user. Each input/output device 1506 includes any suitable structure for providing information to or receiving information from a user, such as a speaker, microphone, keypad, keyboard, display, or touch screen.

[0052] In addition, the UE 1410 includes at least one memory 1508. The memory 1508 stores instructions and data used, generated, or collected by the UE 1410. For example, the memory 1508 could store software or firmware instructions executed by the processing unit(s) 1500 and data used to reduce or eliminate interference in incoming signals. Each memory 1508 includes any suitable volatile and/or non-volatile storage and retrieval device(s). Any suitable type of memory may be used, such as random access memory (RAM), read only memory (ROM), hard disk, optical disc, subscriber identity module (SIM) card, memory stick, secure digital (SD) memory card, and the like.

[0053] As shown in FIGURE 12B, the base station 1470 includes at least one processing unit 1500, at least one transmitter 1552, at least one receiver 1554, one or more antennas 1556, one or more network interfaces 1560, and at least one memory 1558. The processing unit 1500 implements various processing operations of the base station 1470, such as signal coding, data processing, power control, input/output processing, or any other functionality. The processing unit 1500 can also support the methods and teachings described in more detail above. Each processing unit 1500 includes any suitable processing or computing device configured to perform one or more operations. Each processing unit 1500 could, for example, include a microprocessor, microcontroller, digital signal processor, field
programmable gate array, or application specific integrated circuit. The processing unit 1500 may be an asynchronous processor as described herein.

[0054] Each transmitter 1552 includes any suitable structure for generating signals for wireless transmission to one or more UEs or other devices. Each receiver 1554 includes any suitable structure for processing signals received wirelessly from one or more UEs or other devices. Although shown as separate components, at least one transmitter 1552 and at least one receiver 1554 could be combined into a transceiver. Each antenna 1556 includes any suitable structure for transmitting and/or receiving wireless signals. While a common antenna 1556 is shown here as being coupled to both the transmitter 1552 and the receiver 1554, one or more antennas 1556 could be coupled to the transmitter (s) 1552, and one or more separate antennas 1556 could be coupled to the receiver (s) 1554. Each memory 1558 includes any suitable volatile and/or non-volatile storage and retrieval device (s).

[0055] Additional details regarding UEs 1410 and base stations 1470 are known to those of skill in the art. As such, these details are omitted here for clarity.

[0056] In some embodiments, some or all of the functions or processes of the one or more of the devices are implemented or supported by a computer program that is formed from computer readable program code and that is embodied in a computer readable medium. The phrase "computer readable program code" includes any type of computer code, including source code, object code, and executable code. The phrase "computer readable medium" includes any type of medium capable of being accessed by a computer, such as read only memory (ROM), random access memory (RAM), a hard disk drive, a compact disc (CD), a digital video disc (DVD), or any other type of memory.
It may be advantageous to set forth definitions of certain words and phrases used throughout this patent document. The terms "include" and "comprise," as well as derivatives thereof, mean inclusion without limitation. The term "or" is inclusive, meaning and/or. The phrases "associated with" and "associated therewith," as well as derivatives thereof, mean to include, be included within, interconnect with, contain, be contained within, connect to or with, couple to or with, be communicable with, cooperate with, interleave, juxtapose, be proximate to, be bound to or with, have, have a property of, or the like.

While this disclosure has described certain embodiments and generally associated methods, alterations and permutations of these embodiments and methods will be apparent to those skilled in the art. Accordingly, the above description of example embodiments does not define or constrain this disclosure. Other changes, substitutions, and alterations are also possible without departing from the spirit and scope of this disclosure, as defined by the following claims.
WHAT IS CLAIMED IS:

1. A method of operating a clock-less asynchronous processing system comprising a plurality of successive asynchronous processing components, the method comprising:

   providing a first token signal path in the plurality of processing components to allow propagation of a token through the processing components, wherein possession of the token by one of the processing components enables the processing component to conduct a transaction with a resource component that is shared among the processing components; and

   propagating the token from one processing component to another processing component along the first token signal path.

2. The method in accordance with Claim 1, wherein propagating the token is performed at a propagation rate that is related to a latency associated with the processing component.

3. The method in accordance with Claim 2, wherein the latency is variable and is based on an operation to be conducted by the processing component.

4. The method in accordance with Claim 1, wherein propagating the token is performed at a propagation rate that is related to a transaction rate associated with the shared resource component.

5. The method as defined in Claim 4, wherein the transaction rate is variable and is based on the transaction to be conducted with the shared resource component.

6. The method in accordance with Claim 1, wherein lack of possession of the token by the processing component prevents the
processing component from conducting a transaction with the shared resource component.

7. The method in accordance with Claim 1, further comprising:
in response to determining that the processing component desires no transaction with the shared resource component, releasing the token so that the token is propagated along the token signal path to another processing component.

8. The method in accordance with Claim 1, further comprising:
providing a second token signal path in the plurality of processing components separate and distinct from the first token signal path to allow propagation of a second token through the processing components, wherein the first token signal path and the second token signal path form a multi-token ring.

9. The method in accordance with Claim 8, further comprising:
providing an intra-processing component gating system, wherein a first designated token of a plurality of tokens is used to gate other designated tokens in a given order.

10. The method in accordance with Claim 9, wherein releasing the designated token by the processing component becomes a condition to consume another token in the processing component in the given order.
11. The method in accordance with Claim 8, further comprising:
providing an inter-processing component passing system, wherein the first token is delayed from passing from a first processing component to a second processing component to avoid a structural hazard.

12. The method in accordance with Claim 11, further comprising:
providing an intra-processing component gating system, wherein a first designated token of a plurality of tokens is used to gate other designated tokens in a given order;
wherein the inter-processing component passing system and the intra-processing component gating system form a pipeline with different stages.

13. A clock-less asynchronous processing system comprising:
a plurality of successive asynchronous processing components, each processing component comprising token processing logic configured to receive, hold and pass a token from a given processing component to another processing component;
wherein the token processing logic comprises a token signal path in the plurality of processing components to allow propagation of the token through the processing components, wherein possession of the token by one of the processing components enables the processing component to conduct a transaction with a resource component that is shared among the processing components.

14. The processing system in accordance with Claim 13, wherein the token processing logic is configured to propagate the token at a propagation rate that is related to a latency associated with the processing component.
15. The processing system in accordance with Claim 14, wherein the latency is variable and is based on an operation to be conducted by the processing component.

16. The processing system in accordance with Claim 13, wherein lack of possession of the token by the processing component prevents the processing component from conducting a transaction with the shared resource component.

17. The processing system in accordance with Claim 13, wherein the token processing circuitry further comprises intra-processing component gating circuitry, wherein a first designated token of a plurality of tokens is used to gate other designated tokens in a given order.

18. The processing system in accordance with Claim 17, wherein releasing the first designated token by the processing component becomes a condition to consume another token in the processing component in the given order.

19. The processing system in accordance with Claim 13, wherein the token processing circuitry further comprises inter-processing component passing circuitry, wherein the token is delayed from passing from a first processing component to a second processing component to avoid a structural hazard.

20. The processing system in accordance with Claim 19, wherein the token processing circuitry further comprises intra-processing component gating circuitry, wherein a first designated token of a plurality of tokens is used to gate other designated tokens in a given order;
wherein the inter-processing component passing circuitry and the intra-processing component gating circuitry form a pipeline with different stages.
FIG. 1
(PRIOR ART)

FIG. 3A
FIG. 12A

FIG. 12B
**INTERNATIONAL SEARCH REPORT**

**International application No.**

**PCT/US 14/54618**

---

**A. CLASSIFICATION OF SUBJECT MATTER**

- **IPC(8):** G06F 9/46 (2014.01)
- **CPC:** G06F 9/4881

According to International Patent Classification (IPC) or to both national classification and IPC.

---

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

- **IPC(8):** G06F 9/46 (2014.01)
- **CPC:** G06F 9/4881

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

USPC: 718/100-102 (keyword limited - see terms below)

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

- PatBase; GOOGLE: GoogleScholar; GooglePatents

Search Terms: processing, signal, path, rate, speed, latency, delay, propagation, transmit, passing, token, hazard, fault, error, share, resource, asynchronous

---

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 2006/0075210 A1 (Manohar et al.) 06 April 2006 (06.04.2006), entire document, especially; abstract, para. [0013], [0020], [0031], [0039]</td>
<td>1 - 20</td>
</tr>
<tr>
<td>Y</td>
<td>US 2004/0215772 A1 (Dinker et al.) 28 October 2004 (28.10.2004), entire document, especially; abstract, para. [0009], [0010], [0012], [0013], [0034], [0159]</td>
<td>1 - 20</td>
</tr>
<tr>
<td>Y</td>
<td>US 2010/0278190 A1 (Yip et al.) 04 November 2010 (04.11.2010), entire document, especially; abstract, para. [0033], [0036]</td>
<td>2 - 5, 14, 15</td>
</tr>
</tbody>
</table>

* Further documents are listed in the continuation of Box C.

---

**Date of the actual completion of the international search**

18 November 2014 (18.11.2014)

**Date of mailing of the international search report**

30 DEC 2014

---

**Name and mailing address of the ISA/US**

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents

P.O. Box 1450, Alexandria, Virginia 22313-1450

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774