

WO 2014/004186 A1



Published:

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

ASSIGNING ADDRESSES TO DEVICES ON AN INTERCONNECT

Background

[0001] To enable devices coupled to an interconnect such as a universal serial bus (USB) interconnect to communicate, an enumeration process first occurs to identify one or more devices connected to the interconnect. This process is typically implemented by a host controller connected to the bus. Current enumeration methods have complexity in that they often require *a priori* knowledge by the host controller of the connected devices, have predetermined requirements such as geographical addressing, or require additional communications such as separate chip select signals. As such, current enumeration methods may not be suitable in all situations. Also another issue with current methods is that in some cases the deficiency is in added cost due to more circuits, pins, complexity and so forth, to perform enumeration methods.

Brief Description of the Drawings

[0002] FIG. 1 is a block diagram of a pair of devices coupled via an interconnect in accordance with an embodiment of the present invention.

[0003] FIG. 2 is a block diagram of a packet in accordance with an embodiment of the present invention.

[0004] FIG. 3 is a block diagram of a connection of devices to a host controller in accordance with an embodiment of the present invention.

[0005] FIG. 4 is a flow diagram of a method for performing interconnect operations in accordance with an embodiment of the present invention.

[0006] FIG. 5 is a flow diagram of a method for performing ID assignment in accordance with an embodiment of the present invention, from the view of a host controller.

[0007] FIG. 6 is a flow diagram of a method for performing a device ID assignment from a view of a device not having a pre-determined device ID in accordance with one embodiment of the present invention.

[0008] FIG. 7 is a flow diagram of a method for performing a device ID assignment from a view of a device having a pre-determined device ID in accordance with one embodiment of the present invention.

[0009] FIG. 8 is a block diagram of a device configured to perform address allocation in accordance with one embodiment of the present invention.

[0010] FIG. 9 is a block diagram of an example system with which embodiments can be used.

Detailed Description

[0011] Assigning unique addresses to devices connected to a multi-drop bus may be a prerequisite for establishing useful communications among the devices. Consider the case of a host controller (HC) acting as a master coupled to multiple devices, where each device is to be assigned a unique address. Embodiments may provide for address assignment of devices coupled to an interconnect, without geographical addressing or other types of hard coded addresses, although such techniques are possible. In some embodiments, a mixed environment can be used along with either or both of those address techniques.

[0012] In one embodiment, devices coupled to a HC, under command from the HC, can generate addresses based on a random number generated by each device as a means of disambiguation. The random number can be set to be sufficiently large such that the probability of two devices generating the same number is very low. For example in some embodiments this random number may be on the order of between approximately 12 and 32 bits, assuming a system in which less than approximately 16 devices can couple to the same interconnect.

[0013] In an embodiment, the interconnect described is a Low Power Serial (LPS) interconnect, which is a serial multi-drop bus that can be used to primarily couple devices within a system to a main processor such as an SoC, also referred to as an in-box IO situation. This interconnect may provide support for existing software models such as USB. Each device coupled to this interconnect can be configured with a unique address, referred to as a device identifier (ID), or simply an ID. It is permitted that these IDs are statically assigned, and although an assignment

technique described herein applies to dynamic assignment, even devices with statically assigned IDs may be configured to comprehend the protocol.

[0014] Each device coupled to the interconnect may have a storage such as an ID register to store a number at least as large as the maximum number of devices on the interconnect. For purposes of discussion assume that the maximum number of devices is 16 or less, allowing a 4-bit device ID. In addition, each device can be configured to generate a number of a predetermined length that is essentially random, or at least uncorrelated with respect to corresponding random numbers generated by other devices.

[0015] Referring now to FIG. 1, shown is a block diagram of a pair of devices coupled via an interconnect in accordance with an embodiment of the present invention. As shown in FIG. 1, system 10 includes a first device 20 coupled to a second device 30 via an interconnect 25. In an embodiment, first device 20 may correspond to a main processor of a system such as a system-on-a-chip (SoC), while second device 30 may be a peripheral device such as one of multiple sensors or other low data rate devices.

[0016] In the embodiment of FIG. 1, interconnect 25 may be a low power serial (LPS) interconnect having a protocol that will be described further herein. In various embodiments, this interconnect may be of a relatively low overhead and reduced complexity protocol to enable operation at low power, e.g., at a level much less than other serial interconnects such as a universal serial bus (USB) interconnect. Embodiments may be appropriate for general-purpose low or mid-performance communications, such as providing of sensor data, configuration information, or so forth at data rates of less than approximately 1 gigabits per second (Gbps) for point-to-point or appropriately terminated connections, and in the range of between approximately 10-100 Mbps for unterminated connections. In other embodiments, this type of interconnect may be used as a sideband mechanism to provide sideband management of a separate link.

[0017] As further seen in FIG. 1, each device may include a physical unit (PHY) to provide for encoding and handling of transmission/reception via interconnect 25.

As seen, first device 20 may include a PHY 22 and second device 30 may include a PHY 32.

[0018] To provide for communication at low power, embodiments may provide a wire protocol of relatively reduced complexity. This protocol may have various elements including: providing of bus idles between packets, even if the same master communicates multiple packets (thus simplifying receiver complexity); a preamble for purposes of bit locking; a postamble at a transmission end to provide for returning to an idle state of the link to thus enable the receiver to explicitly recognize the end of a transmission without higher-level mechanisms; and providing a predetermined encoding for data communication such as an encoding according to an 8 bit/9 bit (8b/9b) encoding to enable ease of identification of various information of a packet.

[0019] In an embodiment this 8b/9b encoding may be as follows: for each data byte of 8 bits, an additional start bit may be provided. This start bit may be of alternating value (e.g., zero then one). Note that in some embodiments this alternating pattern is not followed for a last byte of a data portion of the packet to mark the last byte before a postamble portion of the packet.

[0020] Referring now to FIG. 2, shown is a block diagram of a packet 100 in accordance with an embodiment of the present invention. As seen in FIG. 2, packet 100 details a protocol of a packet format for communication on a LPS interconnect. In general, packet 100 may be of a relatively limited length as a message portion of the packet can be of relatively small length, e.g., in some embodiments between approximately 1 to 64 bytes.

[0021] As seen, packet 100 can begin with a start bit 105₀, which is one of multiple start bits 105₀ – 105_z, each of which can be interspersed throughout the packet to thus accommodate the 8b/9b encoding of information in the packet. After this start bit, a preamble portion 110 may be provided. This preamble portion may be of a predetermined value, such as the series of alternating logic one and logic zero bits shown in FIG. 2 to thus enable a receiver to identify a beginning of the packet. This is particularly so, since using an open drain-type connection and a pull-

up resistor coupled to the interconnect, potentially long periods of a logic one value may be present on the interconnect prior to transmission of a given packet.

[0022] As further seen in FIG. 2, after preamble portion 110, a second start bit 105_1 may be present. As seen, this start bit is of an alternating value to first start bit 105_0 . Thereafter, a content portion 120 may be provided that can be formed of a length that may be predetermined (or can be determined based on the communication itself). In various embodiments, this content portion 120 may include a first initiating 8-bit block (to thus form a 9 bit block with start bit 105_1) and one or more (and potentially zero) succeeding 9 bit blocks. In an embodiment, each of these blocks of 9 bits can be referred to as a symbol.

[0023] After the content portion 120, another start bit 105_n may be provided. Note that it is assumed that content portion 120 includes, in FIG. 2, an odd number of 9-bit symbols as this succeeding start bit 105_n is of zero value.

[0024] Thereafter, an error detection portion 130 may be provided within the packet. This error detection portion can be an inverted or one's complement version of content portion 120, and as will be described further below, can be used by the receiver for purposes of detecting an error in communication. Finally, a terminal start bit 105_z is provided and thereafter a postamble portion 140 occurs, which again may be of a predetermined value, such as a repeating 4 bit pattern of logic ones and zeros as seen in FIG. 2.

[0025] Although not shown for ease of illustration, understand that after communication of this packet, the interconnect may return to a logic high level until a next packet is communicated, given the open drain configuration and the pull-up resistor present and coupled to the interconnect. Also note that while shown with this particular implementation of a packet for purposes of illustration, understand that other packet types are contemplated for other embodiments. For example, different values of the preamble and postamble portions may be present. Furthermore, different alternating or other patterns for the start bits may be provided. Still further, different arrangements of the content and error detection portions may also be present in some embodiments. Note further that in some implementations, some

type of scrambling of the data information may be provided for purposes of electromagnetic interference (EMI) reduction. Other possibilities exist such as providing for a defined address portion and so forth.

[0026] Note also that in various embodiments, content portion 120 may include, at a beginning portion, a unique identifier of the sender of the packet, which can be implemented via a unique address that can be statically or dynamically determined. By way of this unique identifier provided within the content portion (and thus within each packet communicated) it can be guaranteed that colliding packets will be detected, even in the case where both components start transmitting at the same time.

[0027] In some embodiments, arbitration between components coupled via such an interconnect may be performed using collision detection in which each component can attempt to transmit a packet and monitor for collisions on the interconnect. Accordingly, the PHY present in each component coupled to such an interconnect may provide for collision detection, e.g., via an open drain connection.

[0028] In some embodiments a protocol for this interconnect may provide certain parameters such as minimum idle/turnaround time between transmissions. For example in an embodiment, for an acknowledgment-type response a turnaround time of one symbol time (which as discussed above may correspond to a 9-bit portion) may occur. And for a return to an idle state, a minimum of 1 symbol time may occur.

[0029] In some embodiments, retry mechanisms may also be provided. For example, in an implementation providing for such a retry mechanism, if there is no acknowledgment received when expected, the transmitter may perform a retry transmission after a predetermined number of symbol times if it is the host or upstream device, or after a second, e.g., longer predetermined number of symbol times if it is the target or downstream device. In one such embodiment, the host-based retry mechanism may occur after between approximately 11 and 17 symbols without receipt of an expected acknowledgement, and the receiver-based retry mechanism may occur after between approximately 31 and 43 symbols without

receipt of an expected acknowledgement. In other embodiments, such retry mechanisms may occur after another interval as dictated by an upper level protocol. In general, a retry timer may be set to be greater than X, where X is the expected time within an acknowledgment is to be received. Note also that the values may depend on a largest packet size. Note also that the different and non-overlapping windows in different directions may ensure that repeated collisions are avoided.

[0030] Referring now to FIG. 3, shown is a block diagram of a connection of devices to a host controller in accordance with an embodiment of the present invention. As shown in FIG. 3, system 200 may be part of a computing system, such as a given personal computer (PC)-based system, server system or so forth. In some embodiments, a system may be a portable device such as a laptop or notebook computer, Ultrabook™, smartphone, tablet computer or so forth. As seen, system 200 includes a host controller 210 that acts as a host for one or more LPS interconnects in accordance with an embodiment of the present invention. As shown in FIG. 3, host controller includes three LPS ports 212₀-212₂. Although shown with this limited number of ports for purposes of illustration, understand the scope of the present invention is not limited in this regard. Each port may provide for communication via a link in a bidirectional manner. More specifically, the communication can be of data and clock information, e.g., bidirectional differential data and a differential clock signal.

[0031] As seen, first port 212₀ couples to a hub device 220. In various embodiments, device 220 may be a hub device to enable conversion and interfacing between LPS host controller 210 and devices according to another interconnect protocol such as USB devices (e.g., USB2 devices). As such, hub 220 may provide, in addition to conversion of data information between the different protocols, voltage conversion, e.g., from signals having a 1.0 volt system operation via the LPS links to 3.3 volt signals for communication via USB2. As seen, hub 220 can include a plurality of ports (e.g., ports 222₀-222_n) that can be provided as ports to external devices or internal devices.

[0032] Also seen in FIG. 3, second port 212₁ of host controller 210 may provide for communication with a plurality of LPS devices 230₀-230_n. These devices 230

may be of relatively low speed operation (e.g., less than approximately 100 Mbps) via a multi-drop bus having differential data and clock links. Furthermore, to provide for communications at higher speeds with higher speed devices, a separate port 212₂ may communicate with higher speed LPS devices 235₀-235_n (in that these devices may operate at speeds greater than approximately 100 Mbps).

[0033] Note that host controller 210 may be enabled to provide for support of bridging of LPS and legacy devices for compatibility (such as USB2 devices). In an embodiment, host controller 210 may reuse a USB2 protocol stack and retain a USB programming model, while operating at a lower power level, e.g., due to voltage rails operating at a lower voltage (e.g., one volt). In this way, substantially lower active and idle power can be realized, along with a reduced pin count, as the LPS interconnect may act as a multi-drop bus.

[0034] Referring now to FIG. 4, shown is a flow diagram of a method for performing interconnect operations in accordance with an embodiment of the present invention. As shown in FIG. 4, method 200 may begin with an initialization process including blocks 205 and 210. During such operations, the interconnect may operate in a single-ended open drain signaling mode to ensure that collisions are detected. And, during the enumeration phase all devices operate at a data rate sufficiently slow so that open drain signaling will work (e.g., less than approximately 1 megabit per second (Mbit/s)) and may guarantee that data collisions will be detected. Such initialization operations can be performed after a reset, unless state information has been specifically saved. That is, these initialization operations can be performed when one or more devices coupled to the interconnect do not have a non-volatile storage or do not save their state information prior to entry into a low power or powered off state. In addition, such initialization operations may be performed at runtime as needed, e.g., to support power gated devices.

[0035] Specifically, at block 205 an enumeration phase is performed in which unique device identifiers (device IDs) can be assigned to each of the devices coupled to the interconnect. Note that the device ID assignment may be independent of protocol-configuration operations. That is, one or more devices coupled to the interconnect can have a device ID modified without disrupting other

configured devices. Details of an address assignment process will be discussed further below.

[0036] Next at block 210, device discovery and configuration can be performed for devices that were not previously configured. Such discovery and configuration processes may include determination of device type, manufacturer, model, resource requirements, association to specific driver software, and assignment of resources such as addresses, interrupt vectors, and so forth. After such initialization operations, control passes to block 215 where normal operations may commence. Note that for active transmissions, non-open drain signaling may occur with a switch in idle mode to allow an interrupt mechanism. Thus the bus is parked to idle in some state where typically just one device is allowed to drive the bus. However, any device can request service via this circuit mechanism to enable any device to indicate a request for service, which would typically be followed by some type of query or polling mechanism to determine the source of the request for service.

[0037] In some embodiments there may be certain fundamental assumptions, specifically: devices without pre-established unique IDs can implement a random number generator; bus collisions corrupt data in a consistent detectable way, but are non-damaging; and the ID space is small (e.g., 4 bits).

[0038] In an embodiment, an address allocation protocol may include the following commands:

[0039] ID_Poll command in which addressed device(s) respond with the device's stored random number (note that the device ID is implied by the response), followed by that number's complement;

[0040] Change_Random in which all devices without locked addresses (see below, ID_Lock) regenerate their random numbers;

[0041] ID_Lock command in which an addressed device ignores subsequent Change_Random commands until a reset occurs; and

[0042] ID_Complete in which devices are notified that the address assignment process has completed.

[0043] In general, enumeration begins with the HC issuing an ID_POLL command to which each device decodes and responds. The ID_POLL packet contains a packet identifier and an ID. Devices respond in one of three possible ways: 1) if the ID in the packet matches none of the ID's in the devices nothing is returned; 2) if exactly one device matches it returns its random number followed by the ones complement thereof, such that HC will determine that the random number and its complement are consistent (e.g., such that a sum or an exclusive OR of the values equals a predetermined number); and 3) if two or more devices match an ID of an ID_POLL command, each will return their respective random numbers and their complements, with the result that, since the two random numbers differ, there will be a data corruption. By having the device return its random number and complement, it is possible for the HC to determine that one and exactly one device is responding when a consistent data portion and error detection portion occur. Note that the device's random number can be configured to map to its ID. In an embodiment, a device may implement a predetermined random number if (and only if) it can be known *a priori* that this number will not conflict with any other device and/or that only some part of the number may actually be random. With a sufficiently large random number the probability of detecting a corruption between the random number and its complement (due to data collisions) approaches unity.

[0044] This polling process may occur as the HC walks through its entire ID field, determining which devices have unique ID's and which ones share an ID. To all those devices that have a unique ID, the HC issues an ID_LOCK command, which contains the ID value to be locked. This command locks the ID for the corresponding device (and also its random number generator) so that subsequent re-randomization commands have no effect. All devices having an ID that has not been locked may be configured to invalidate the address value in the command, to prevent such devices from mapping to an ID that has already been selected.

[0045] Next, the HC issues a CHANGE_RANDOM command to which only those devices whose ID is not locked respond. This command may cause the affected devices to regenerate their random numbers which in turn cause generation of new ID mappings.

[0046] The host controller may repeat the above operations until each device has a unique ID. Once the ID assignment phase is complete the random number generator circuits may be disabled to save power.

[0047] Referring now to FIG. 5, shown is a flow diagram of a method for performing ID assignment in accordance with an embodiment of the present invention, from the view of a host controller.

[0048] As seen, method 230 may begin responsive to a reset. Here, a scoreboard (which may be in the host controller or may be implemented through a higher level mechanism (e.g., in a system software)) may be cleared of all allocated IDs. Note that this scoreboard may be a non-volatile or other storage to store an indication of the devices coupled to the interconnect and their corresponding device IDs.

[0049] Still referring to FIG. 5, next the host controller may issue a first command, namely a Change_Random command (block 240). This command causes devices having unallocated IDs to regenerate a new random number and device ID.

[0050] Control next passes to block 245 where an unallocated address (A) can be selected and the host controller can issue a second command, namely an ID_Poll (A) command, to thus request a reply from any devices having this address A. Next control passes to diamond 250 where it can be determined whether any valid replies are received from a device within a predetermined time. This determination can be based on whether a correct reply is received within a predetermined interval, which in an embodiment can be on the order between approximately 1 and 20 symbols. If not, control passes back to block 240 for issuance of another Change Random command.

[0051] If a valid reply is received, control passes to block 260 where the address may thus be allocated and marked as allocated, and a third command, namely an ID_Lock (A) command, can be sent to thus cause the correctly responding device to lock the address to the device. Note that marking of the allocated address may be in the scoreboard. Control next passes to diamond 265 where it can be determined whether all addresses have been checked. If not, control passes back to block 245

above. Otherwise, control passes to block 270 where a fourth command, namely an ID_Complete command, may be issued to thus indicate to the devices that the address allocation process is completed.

[0052] Referring now to FIG. 6, shown is a flow diagram of a method for performing a device ID assignment from a view of a device not having a pre-determined device ID. As seen, method 300 may begin at reset or receipt of a Change_Random command during normal operation (block 305). Control next passes to block 310 where a scoreboard can be cleared of all allocated IDs. Note that this scoreboard may be included in or associated with the device. Control next passes to block 315, where a random number (R) and a random identifier (namely an identifier that acts as a device ID) can be selected from non-allocated device IDs. In this initial flow, there are no allocated IDs to the knowledge at the device, as the scoreboard has just been cleared.

[0053] Next control passes to block 320 where the device waits on a command. If the received command is a Change_Random command, as determined at diamond 325, control passes back to block 315, above. Otherwise, control passes to diamond 330 to determine whether it is an ID_Poll command. If so, control passes to diamond 350 to determine whether the address associated with this ID_Poll command matches the device ID (namely the randomly generated ID). If not, control passes to block 320 above. If the addresses match, control passes to block 355 where the device may issue a response back a host controller. As discussed above, this response may indicate, in a data portion of the reply the new random number (R), and the corresponding complement of this information may be included in the error detection portion of the reply. Thus control passes again to the above block 320.

[0054] If the received message is instead an ID_Lock message, as determined at diamond 340, control passes to diamond 360 to determine whether the address associated with the ID_Lock request matches the random ID. If so, control passes to block 365 where the device may lock its ID to this random ID value and then the device may wait for the address allocation process to complete via an ID_Complete command, which when received at diamond 370 causes the method to conclude.

Note that in an embodiment the ID_Lock command may cause the device to store its random ID and random number, both in an address register of the device as well as in corresponding entry for the device in the scoreboard. Note that if an error occurs, control passes to block 345 where the method may be completed. Although shown at this high level in the embodiment of FIG. 6, understand the scope of the present invention is not limited in this regard.

[0055] Thus embodiments do not require any *a priori* knowledge by the HC, geographical addressing, or the need for separate chip select signals. Furthermore, embodiments can be insensitive to the number of devices being enumerated and can perform discovery either at power-up or when an additional device is powered-on, thus supporting plug and play without selectively enabling and disabling ports. In a multi-crop configuration, the interconnect may appear like a USB interconnect to higher protocol levels, in part due to the address allocation described herein.

[0056] Embodiments thus can be used in many different environments. Referring now to FIG. 7, shown is a flow diagram of a method for performing a device ID assignment from a view of a device having a predetermined device ID. As seen, method 400 may begin at reset or receipt of a Change_Random command during normal operation (block 405). Control next passes to block 410 where the device waits on a command. If the received command is a Change_Random command, as determined at diamond 425, control passes back to wait at block 410. Otherwise, control passes to diamond 430 to determine whether it is an ID_Poll command. If so, control passes to diamond 450 to determine whether the address associated with this ID_Poll command matches the device ID (namely the pre-allocated device ID). If not, control returns to the wait state at block 410. If the addresses match, control passes to block 455 where the device may issue a response back a host controller. In this situation, the response may indicate, in a data portion of the reply the same response format for a device with an unallocated address. Control then passes again to the wait state at block 310.

[0057] If the received message is instead an ID_Lock message, as determined at diamond 440, control passes to diamond 460 to determine whether the address associated with the ID_Lock request matches the pre-allocated ID. If so, control

passes to block 465 where the device may lock its ID to this random ID value and then the device may wait for the address allocation process to complete via an ID_Complete command, which when received at diamond 470 causes the method to conclude. Note that if an error occurs, control passes to block 445 where the method may be completed. Although shown at this high level in the embodiment of FIG. 7, understand the scope of the present invention is not limited in this regard.

[0058] Referring now to FIG. 8, shown is a block diagram of a device configured to perform address allocation as described herein. As seen, device 500 may be an integrated circuit (IC) that includes various logic. In different implementations, this device may be an in-box device such as a peripheral device coupled to a SoC, or may be an external device coupled to a system via an external port.

[0059] In the embodiment shown, device 500 may include a plurality of cores 510_a-510_n, which can be heterogeneous or homogeneous cores. Of course while multiple cores are shown, understand that in other implementations a single core may be present. And still further it is possible that a device may include no cores and instead have one or more other logic blocks. In addition, an IP logic 520 may be present and can be used to perform specialized operations, e.g., one or more functions that can be offloaded to it by one or more of the cores.

[0060] For purposes of performing address allocation operations, an ID logic 535 may be present and can be coupled to an LPS protocol stack 530, which may generally include the same layered architecture as the host controller of FIG. 3. Note that while this implementation is with specific reference to a device coupled to an LPS interconnect 570, understand that embodiments apply equally to other serial interconnect-based schemes.

[0061] ID logic 535 may be coupled to an ID register 550 which may store a device ID that in turn can be generated by ID logic 535 responsive to commands received from a host controller and using a mapping to a random number, which can be generated by a random number generator 540. In some implementations, note that a dedicated random number generator may not be present, and this functionality can be implemented, e.g., within one or more of the cores, IP logic, or ID logic.

[0062] As further seen, device 500 may further include a memory 560 such as a cache memory or so forth. This memory may include a scoreboard 565, which can include a plurality of entries that each stores a valid device identifier, each stored responsive to a command from a host controller to lock a given device ID. In this way, random number generator 540 can generate random numbers that do not conflict with already allocated device IDs that are present in scoreboard 565. Although shown at this high level in the embodiment of FIG. 8, understand the scope of the present invention is not limited in this regard.

[0063] Embodiments thus can be used in many different environments. Referring now to FIG. 9, shown is a block diagram of an example system 600 with which embodiments can be used. As seen, system 600 may be a smartphone or other wireless communicator. As shown in the block diagram of FIG. 9, system 600 may include a baseband processor 610 which may be a multicore processor that can handle both baseband processing tasks as well as application processing. Thus baseband processor 610 can perform various signal processing with regard to communications, as well as perform computing operations for the device. In turn, baseband processor 610 can couple to a user interface/display 620 which can be realized, in some embodiments by a touch screen display. In addition, baseband processor 610 may couple to a memory system including, in the embodiment of FIG. 9 a non-volatile memory, namely a flash memory 630 and a system memory, namely a dynamic random access memory (DRAM) 635. As further seen, baseband processor 610 can further couple to a capture device 640 such as an image capture device that can record video and/or still images. Baseband processor 610 can further couple to one or more sensor devices 645, e.g., via an LPS interconnect as described herein.

[0064] To enable communications to be transmitted and received, various circuitry may be coupled between baseband processor 610 and an antenna 680. Specifically, a radio frequency (RF) transceiver 670 and a wireless local area network (WLAN) transceiver 675 may be present. In general, RF transceiver 670 may be used to receive and transmit wireless data and calls according to a given wireless communication protocol such as 3G or 4G wireless communication protocol

such as in accordance with a code division multiple access (CDMA), global system for mobile communication (GSM), long term evolution (LTE) or other protocol. Other wireless communications such as receipt or transmission of radio signals, e.g., AM/FM, or global positioning satellite (GPS) signals may also be provided. In addition, via WLAN transceiver 675, local wireless signals, such as according to a Bluetooth™ standard or an IEEE 802.11 standard such as IEEE 802.11a/b/g/n can also be realized. Note that the link between baseband processor 610 and one or both of transceivers 670 and 675 may be via a low power converged interconnect that combines and maps functionality of a PCIe™ interconnect and a low power interconnect such as a MIPI interconnect. Although shown at this high level in the embodiment of FIG. 9, understand the scope of the present invention is not limited in this regard.

[0065] Embodiments may be used in many different types of systems. For example, in one embodiment a communication device can be arranged to perform the various methods and techniques described herein. Of course, the scope of the present invention is not limited to a communication device, and instead other embodiments can be directed to other types of apparatus for processing instructions, or one or more machine readable media including instructions that in response to being executed on a computing device, cause the device to carry out one or more of the methods and techniques described herein.

[0066] Embodiments may be implemented in code and may be stored on a non-transitory storage medium having stored thereon instructions which can be used to program a system to perform the instructions. The storage medium may include, but is not limited to, any type of disk including floppy disks, optical disks, solid state drives (SSDs), compact disk read-only memories (CD-ROMs), compact disk rewritables (CD-RWs), and magneto-optical disks, semiconductor devices such as read-only memories (ROMs), random access memories (RAMs) such as dynamic random access memories (DRAMs), static random access memories (SRAMs), erasable programmable read-only memories (EPROMs), flash memories, electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, or any other type of media suitable for storing electronic instructions.

[0067] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is:

- 1 1. An apparatus comprising:
2 intellectual property (IP) logic means for performing operations responsive to
3 instructions;
4 random number generator means for generating a random number
5 responsive to a first command from a host controller coupled to the apparatus via an
6 interconnect; and
7 identification logic means for generating a device identifier for the apparatus, wherein
8 the apparatus is to provide a reply to the host controller including the random
9 number responsive to an identification request from the host controller corresponding
10 to the device identifier.
- 1 2. The apparatus of claim 1, wherein the apparatus is to store the device
2 identifier in an identification storage of the apparatus responsive to an identification
3 lock request from the host controller corresponding to the device identifier.
- 1 3. The apparatus of claim 2, wherein the apparatus is to further store the device
2 identifier in a scoreboard responsive to the identification lock request.
- 1 4. The apparatus of claim 3, wherein the apparatus is to power down the random
2 number generator responsive to the identification lock request from the host
3 controller corresponding to the device identifier.
- 1 5. The apparatus of claim 2, wherein the apparatus is to regenerate the random
2 number responsive to receipt of a change request from the host controller.
- 1 6. The apparatus of claim 5, wherein the apparatus is to regenerate the random
2 number to obtain a device identifier not present in a scoreboard of the apparatus.
- 1 7. The apparatus of claim 6, when the apparatus is to not regenerate the random
2 number if the change request is received after the identification lock request from the
3 host controller corresponding to the device identifier.

- 1 8 The apparatus of any one of claims claim 1-7, wherein the apparatus is
2 coupled to the interconnect via open drain signaling in an enumeration phase.
- 1 9. The apparatus of claim 8, wherein the apparatus is coupled to the
2 interconnect via non-open drain signaling in an operational phase.
- 1 10. The apparatus of any one of claims claim 1-7, wherein the apparatus
2 comprises a semiconductor device provided by a vendor without a pre-established
3 device identifier.
- 1 11. A method comprising:
2 issuing, from a host controller, a first command to a plurality of devices coupled to a
3 multi-drop interconnect to request a reply from at least one of the plurality of devices
4 having a device identifier of a first value;
5 determining if a valid reply is received from one of the plurality of devices and if so,
6 storing the device identifier of the first value in a storage of the host controller; and
7 otherwise issuing a second command to cause at least some of the plurality of
8 devices to regenerate the device identifier having a different value.
- 1 12. The method of claim 11, further comprising if the valid reply is received from
2 one of the plurality of devices, issuing a third command to cause the device identifier
3 of the first value to be locked by the one of the plurality of devices.
- 1 13. The method of claim 12, further comprising iterating issuing the first
2 command, determining if the valid reply is received, and issuing the second
3 command until a device identifier is allocated to each of the plurality of devices.
- 1 14. The method of claim 13, further comprising thereafter issuing an identification
2 completion command to indicate that the device identifier assignment is completed.

1 15. The method of claim 11, wherein at least one of the plurality of devices that is
2 unlocked is to generate a new device identifier responsive to the second command.

1 16. The method of claim 12, further comprising issuing the second command
2 responsive to receipt of an invalid reply from one or more of the plurality of devices.

1 17. The method of claim 16, wherein the invalid reply is due to at least two of the
2 plurality of devices communicating a reply having a common device identifier and a
3 different random number.

1 18. The method of claim 16, further comprising determining that an invalid reply is
2 received responsive to a mismatch between a message portion of the invalid reply
3 and an error detection portion of the invalid reply.

1 19. A system comprising:
2 a host controller coupled to an interconnect;
3 a first device coupled to the interconnect; and
4 a second device coupled to the interconnect, wherein the second device is to receive
5 a poll command from the host controller and to return a device identifier if an
6 identifier of the poll command corresponds to the device identifier, store the device
7 identifier in an identification storage if a lock command is received from the host
8 controller corresponding to the device identifier, and regenerate a random number if
9 a change command is received from the host controller before the lock command.

1 20. The system of claim 19, wherein the second device is to regenerate a new
2 device identifier based on the regenerated random number.

1 21. The system of claim 20, wherein the second device is to regenerate the new
2 device identifier with a value not present in a scoreboard.

- 1 22. The system of claim 19, wherein the second device is to power down a
2 random number generator of the second device responsive to receipt of the lock
3 command from the host controller corresponding to the device identifier.
- 1 23. A communication device arranged to perform the method of any one of claims
2 11 to 18.
- 1 24. At least one machine readable medium comprising a plurality of instructions
2 that in response to being executed on a computing device, cause the computing
3 device to carry out a method according to any one of claims 11 to 18.
- 1 25. An apparatus for processing instructions, configured to perform the method of
2 any one of claims 11 to 18.
- 1 26. An apparatus comprising means for performing the method of any one of
2 claims 11 to 18.

10

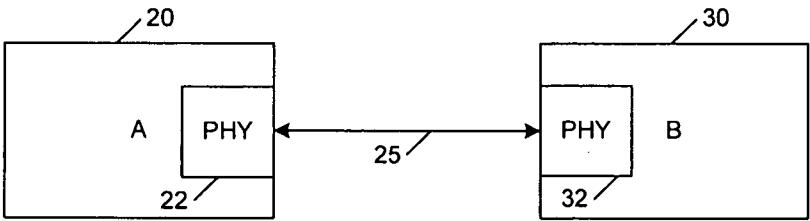


FIG. 1

100

<u>105₀</u>	<u>110</u>	<u>105₁</u>	<u>120</u>	<u>105_n</u>	<u>130</u>	<u>105_z</u>	<u>140</u>
	Preamble		<Content Portion>		<Error Detection Portion>		Postamble
0	1010 1010	1	<Content>	0	<Content Inverted>	0	1100 1100

FIG. 2

2/8

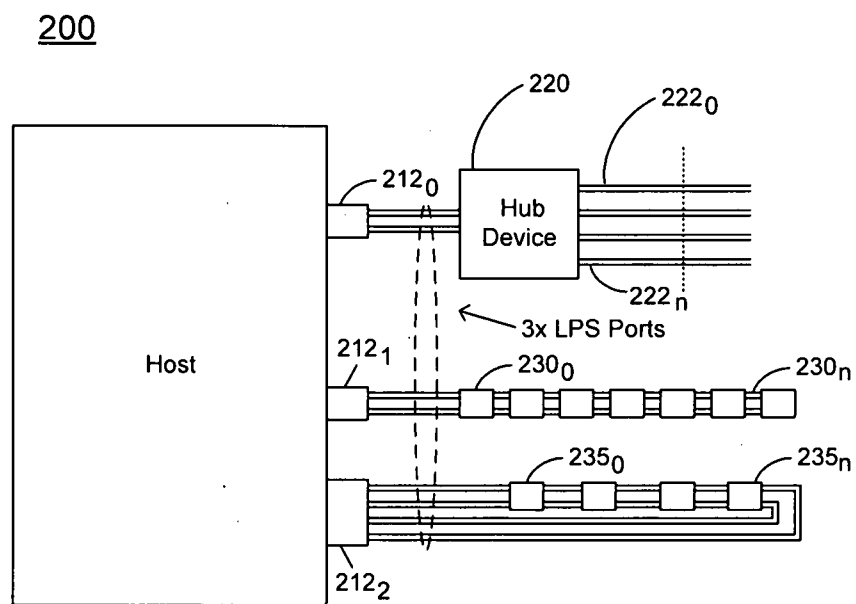


FIG. 3

3/8

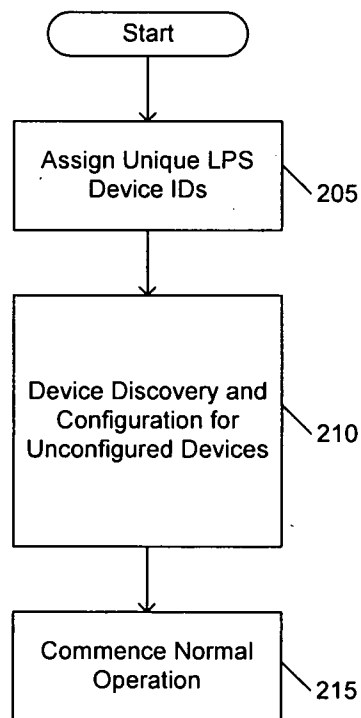
200

FIG. 4

4/8

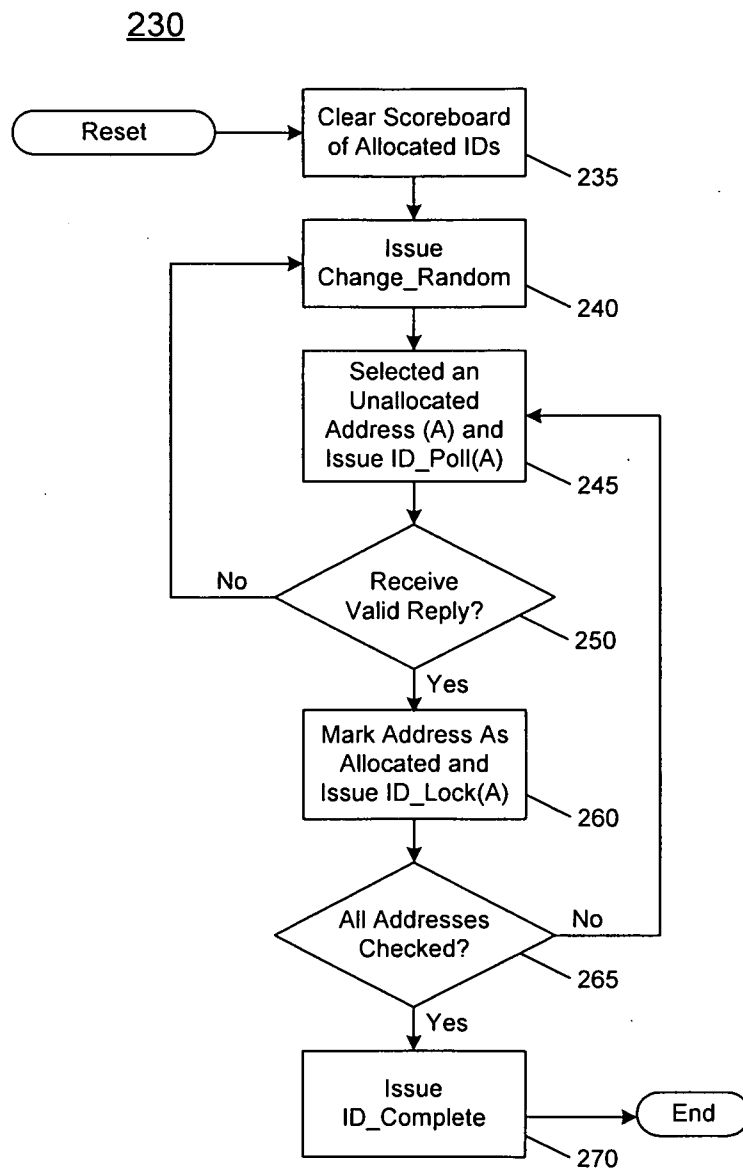


FIG. 5

5/8

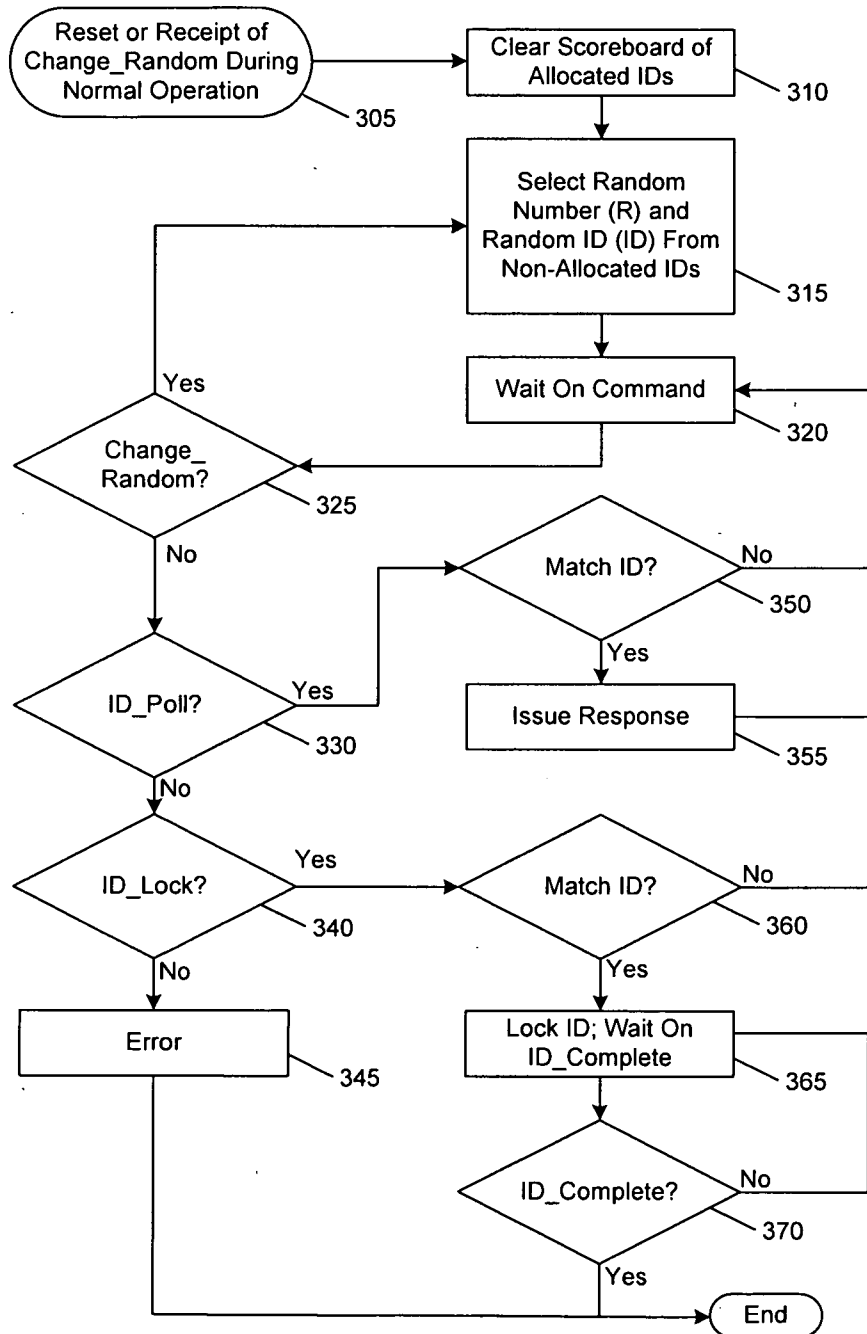
300

FIG. 6

6/8

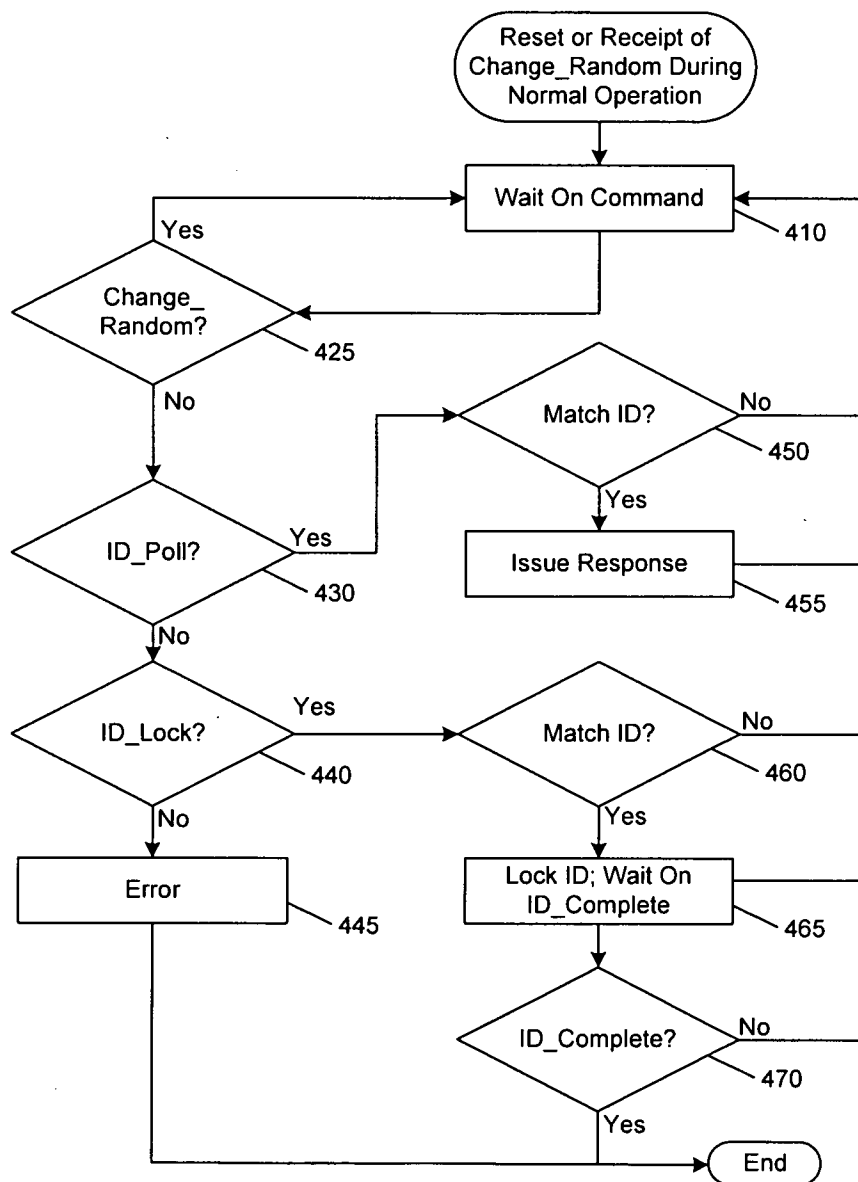
400

FIG. 7

7/8

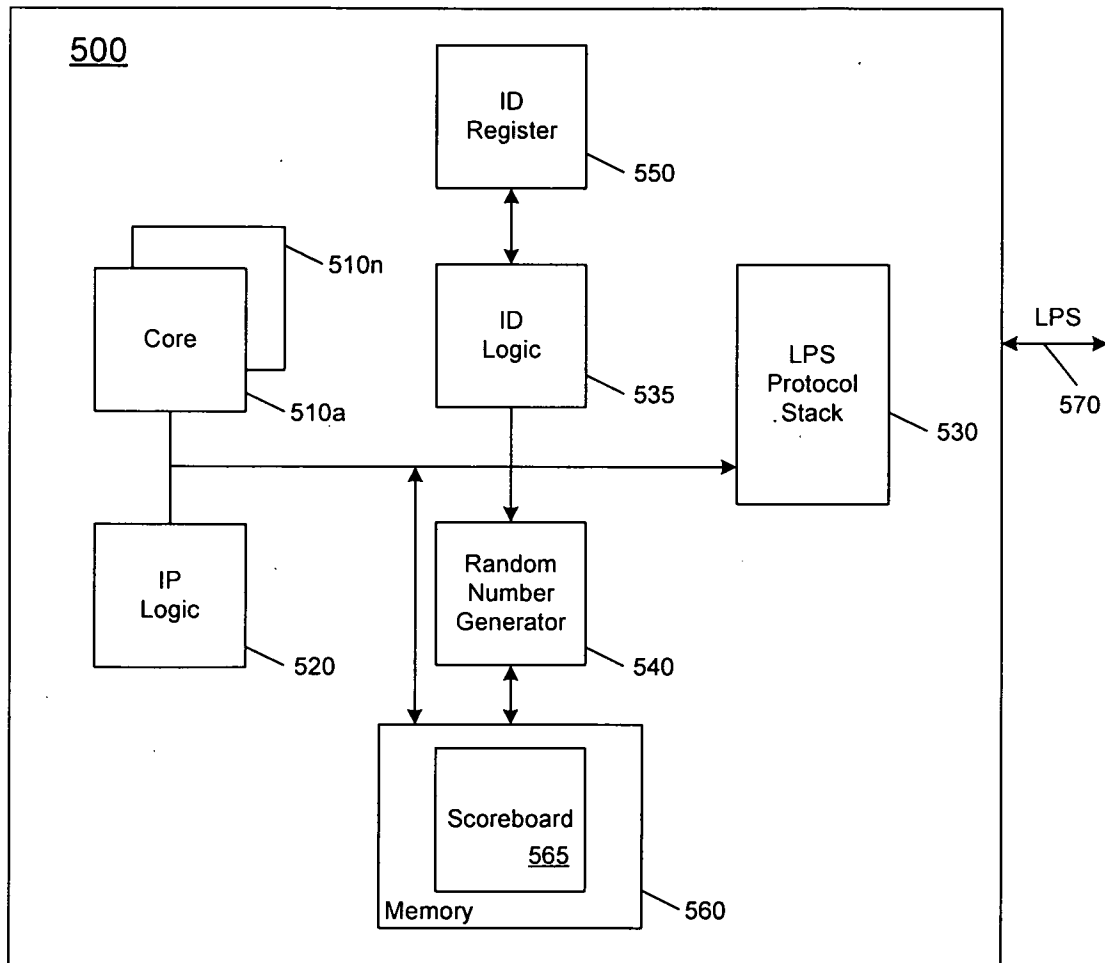


FIG. 8

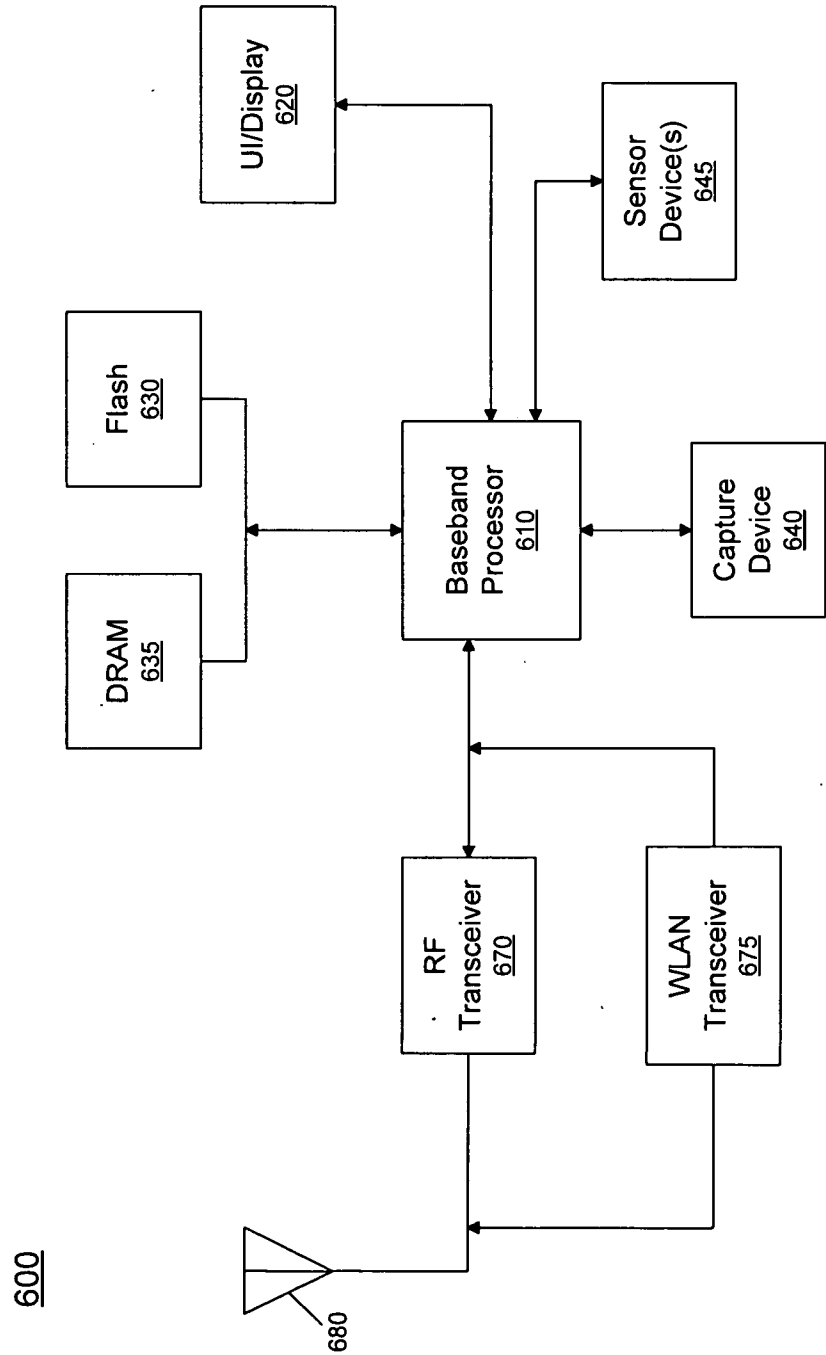


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/046452**A. CLASSIFICATION OF SUBJECT MATTER****G06F 13/14(2006.01)i**

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 13/14; G05B 19/02; H04J 3/16; H04L 12/413; G06F 12/14; H04L 9/00; G06F 21/00

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: identification, device, property, intellectual, random, number, generate, reply, identifier, host, interconnect, and simialr terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7936706 B2 (CLIFTON W. WOOD, JR.) 03 May 2011 See column 4, lines 39-44; column 5, lines 1-14, 17-21; column 6, lines 13-15 ; column 12, lines 53-55; and claims 9, 21, 36.	1-2, 10
A		3-9, 11-26
A	US 2009-0129595 A1 (JOUNI STAM) 21 May 2009 See paragraph [0019]; and figure 2.	1-26
A	US 2010-0007513 A1 (JOHN K. HORKY et al.) 14 January 2010 See paragraphs [0043]-[0047], [0052]-[0056]; and figures 3, 6-7.	1-26
A	US 2012-0102334 A1 (DANIEL O'LOUGHLIN et al.) 26 April 2012 See paragraphs [0130]-[0156]; and figures 7A-7C.	1-26
A	US 2009-0282256 A1 (MILAN RAKIC et al.) 12 November 2009 See paragraphs [0071]-[0072], [0082]-[0086]; and figures 7, 9-10.	1-26



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

16 October 2013 (16.10.2013)

Date of mailing of the international search report

16 October 2013 (16.10.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City,
302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KANG Sung Chul

Telephone No. +82-42-481-8405



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/046452

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7936706 B2	03/05/2011	AT 394849 T	15/05/2008
		AU 1999-32859 A1	06/09/1999
		AU 3285999 A	06/09/1999
		DE 69938655 D1	19/06/2008
		EP 1072128 A1	31/01/2001
		EP 1072128 B1	07/05/2008
		ES 2306519 T3	01/11/2008
		US 2001-0014090 A1	16/08/2001
		US 2005-0207364 A1	22/09/2005
		US 2006-0056325 A1	16/03/2006
		US 2006-0209781 A1	21/09/2006
		US 2008-0007412 A1	10/01/2008
		US 2008-0042806 A1	21/02/2008
		US 2009-0322491 A1	31/12/2009
		US 2012-0265584 A1	18/10/2012
		US 2012-0271683 A1	25/10/2012
		US 2013-0073348 A1	21/03/2013
		US 6275476 B1	14/08/2001
		US 7315522 B2	01/01/2008
		US 7639638 B2	29/12/2009
		US 7672260 B2	02/03/2010
		US 7760677 B2	20/07/2010
		US 8040829 B2	18/10/2011
		US 8488581 B2	16/07/2013
		WO 99-43127 A1	26/08/1999
US 2009-0129595 A1	21/05/2009	CN 101223531 A0	16/07/2008
		EP 1902402 A1	26/03/2008
		FI 119530 B1	15/12/2008
		FI 20050622 A0	10/06/2005
		FI 20050622 D0	10/06/2005
		JP 2008-542940 A	27/11/2008
		WO 2006-131594 A1	14/12/2006
US 2010-0007513 A1	14/01/2010	US 2012-206244 A1	16/08/2012
		US 8269612 B2	18/09/2012
US 2012-0102334 A1	26/04/2012	CA 2743958 A1	27/05/2010
		CN 102648471 A	22/08/2012
		EP 2350910 A1	03/08/2011
		JP 2012-510189 A	26/04/2012
		SG 171730 A1	28/07/2011
		WO 2010-057312 A1	27/05/2010
US 2009-0282256 A1	12/11/2009	CN 102017567 A	13/04/2011
		EP 2274892 A1	19/01/2011
		WO 2009-138825 A1	19/11/2009