describe the tilt of the device to the user via audio. Furthermore, in another aspect, no indication of angular displacement or tilt region is displayed or otherwise generated.

[0497] Returning to FIG. 30, a selection of the first control is received (step ES307). In one aspect, the control is a keypad button, and selection occurs when the user depresses the button, thereby enabling a signal to be generated and transmitted to the processor indicating that a selection of the keypad button has occurred. In another aspect, the control is not a physical control, but rather an icon on a touch-sensitive screen. In this aspect, selection occurs when the user touches an area of the touch-sensitive screen associated with the icon, where a touch-sensitive screen application reads the coordinates of the touch, correlates the coordinates with the location of the icon, and transmits a signal indicating that the control has been selected. Other types of control selections are also contemplated.

[0498] According to the FIG. 32 implementation, device E500 includes a keypad, or grouping of controls, which enables the user to enter text in order to interact with the GUI presented on display E505. Each control corresponds to multiple output signals, each output signal associated with a characters. In one aspect, the keypad includes eight controls, labeled "2" to "9", that each correspond to multiple letters and a number. For example, the control labeled "2" corresponds to the letters "A," "B," and "C," and the number "2." In addition, other controls included in the keypad perform other text entry functions. For example, the control labeled "*" is used to change the case of a next character that is output. The control labeled "0" is used to advance to a subsequent character after a current character has been specified, and the control labeled "#" is used to insert a "space" character.

[0499] One of the first plurality of output signals is output based at least upon the selection and the angular displacement (step ES309), or at least upon the selection, the angular displacement, and the plurality of tilt regions. Since the first control is associated with a first plurality of output signals, the angular displacement, or the angular displacement and the plurality of tilt regions are used to determine which one of the first plurality of output signals are output. In one aspect, the neutral position of the device is determined in relation to one axis, where three tilt regions are defined around that one axis, and where the first control is associated with three tilt regions. In this case, if the angular displacement is in the first tilt region, the first output signal is output, if the angular displacement is in the second tilt region, the second output signal is output, and if the angular displacement is in the third tilt region, the third output signal is output. hi an alternative aspect, the output signal is output based upon the angular displacement and the number of output signals associated with the first control, based upon a formula or an algorithm.

[0500] Various figures depict front and side views of the FIG. 32 device in different states of manipulation. In particular, FIGS. 34A and 34B illustrate front and side views, respectively, of device E500 in the neutral position. FIG. 35A illustrates a front view of the device manipulated in a negative roll about the X-axis and FIG. 35B illustrates a front view of the device manipulated in a positive roll about the X-axis. Similarly, FIG. 36A illustrates a side view of the device manipulated in a positive pitch about the Y-axis and FIG. 36B illustrates a side view of the device manipulated in a negative pitch about the Y-axis. In FIGS. 35 and 36, the device has been tilted approximately .±.30.degree. about the respective axes from the neutral position, shown in FIG. 34.

[0501] The orientation of the device, as indicated by the angular displacement measured by the tilt sensor, when a control of the keypad is selected affects the output signal output by the device, affecting, for example, the character generated by the control selection. Each of the multiple characters or output signals represented by a single control of a keypad correspond to a different orientation of the device. When one of the controls of the keypad is selected, the device identifies the plurality of characters that correspond to the selected control and the orientation of the device indicated by the tilt sensor. One of the multiple characters and a case for the character are identified based on the identified orientation, and the identified character is output.

[0502] The degree to which the device has been rolled to the left or right when a control is selected affects which one of the multiple characters represented by the control is output. In one implementation, the controls that represent multiple characters represent three letters, and the letters represented by the control are listed from left to right on the control. The device is configured to indicate that the device is rolled left, rolled right, or not rolled left or right. In one such implementation, rolling the device to the left when the control is selected indicates that the leftmost listed character should be output. Similarly, rolling the device to the right when the control is selected indicates that the rightmost listed character should be output. Finally, keeping the device oriented in the neutral position when the control is selected indicates that the center character should be output.

[0503] In another implementation, rolling the device to the left when the control is selected indicates that the rightmost listed character should be output, rolling the device to the right when the control is selected indicates that the leftmost listed character should be output, and keeping the device oriented in the neutral position when the control is selected indicates that the center character should be output. Such an implementation may be used, for example, because rolling the device to the left causes the rightmost listed character to appear above and more prominently than the other listed characters, and rolling the device to the right causes the leftmost listed character to appear above and more prominently than the other listed characters.

[0504] In other implementations, the controls of the keypad represent more than three characters, such as three letters and a number, or four letters and a number. For example, the control on a conventional telephone labeled "7" corresponds to the letters "P,""Q,""R," and "S," and the number "7." In such a case, the tilt sensor is configured to identify more than three discrete left-to-right roll positions such that one of the more than three characters represented by a selected control may be identified based only on the roll orientation of the device. Each of the discrete roll positions correspond to one of the characters represented by the selected control . For example, if the selected control is the key labeled "7", the device being rolled as illustrated in region E404 of FIG. 31B would indicate that the letter "P" should be output, the device being rolled as illustrated in region E405 of FIG. 31B would indicate that the letter "Q" should be output, the device being rolled as illustrated in region E406 of FIG. 31B would indicate that the letter "R" should be output, the device being rolled as illustrated in region E407 of FIG. 31B would indicate that the letter "S" should be output, and the device being oriented in the neutral position, as illustrated in FIG. 28, would indicate that the number "7" should be output.

[0505] While the roll orientation of the device is used to identify a character to be output, the pitch orientation of the device is used to identify a case for the character. In one implementation, the device being pitched (or tilted) forward when a control is selected causes a character that is identified by the roll (left-to-right tilt) orientation of the device to be output in upper case. Similarly, the device not being pitched forward or backward (in a neutral pitch position) when a control is selected causes a character that is identified by the roll (left-to-right tilt) orientation of the device to be output in lower case.

[0506] In some implementations, the device being pitched (or tilted) backward may cause a symbol to be output. The symbol may be a symbol corresponding to the number represented by the selected control on a conventional computer keyboard. For example, if the control that represents the number "1" is selected while the device is pitched backward, the symbol "!" may be output, because the symbol "!" corresponds to the number "1" on a conventional computer keyboard (e.g., pressing "Shift" and "1" on a computer keyboard outputs the character "!").

[0507] The tilt sensor is capable of detect more tilt positions in the pitch direction than is necessary to indicate the case of the character to be output. As such, the pitch positions that are not used to indicate the case of the character may be used to select the character. For example, a control may represent three letters and a number, and three roll positions may be used to select among the three letters. Two pitch positions may select the case for letters, and a third pitch tilt position may select the number represented by the key.

[0508] Furthermore, the tilt sensor independently indicates whether the device has been rolled left, neutral, or right or whether the device has pitched forward, neutral, or backwards, thereby allowing the tilt sensor to indicate whether the device is in one of nine orientations. Each of the nine orientations may correspond to a character and a case for the character.

[0509] FIG. 37 is a table showing one possible mapping of device orientations to output signals corresponding to characters and cases that may be output when the control labeled "2" on the keypad is selected. In the illustrated mapping, the device being rolled left and pitched forward causes the capital letter "A" to be output, the device not being rolled or pitched in either direction case the lower case letter "b" to be output, and the device being pitched backwards causes the number "2" to be output. In other implementations in which the tilt sensor may identify more than three roll positions or more than three pitch positions, more orientations that may be mapped to characters and cases are available.

[0510] Output signals corresponding to characters are described as being selected based on a first axis angular displacement or tilt position of the device, and output signals corresponding to upper or lower cases for the characters are described throughout as being selected based on a second axis angular displacement or position of the device. In other implementations, the angular displacement in different axes may effectuate the output of signals corresponding to characters or upper and lower cases of characters. In general, any orientation of the device may be mapped to any character and case for the character, regardless of which of the axes was used to select the character or the case.

[0511] In addition to outputting a signal corresponding to a character that is output in response to selection of a control, the orientation of the device may be used to indicate a menu option that is to be selected. For example, selection of a control that does not correspond to any characters, such as the "1" key on a telephone, causes a menu to be presented on the display of the telephone, where each option of the menu correspond to a different orientation of the telephone. The orientation of the device when a control indicating that a selection from the menu should be made (e.g., an "OK" key, an "Enter" key, or the "1" key) is selected may indicate which of the menu options is selected. In one aspect, a menu of symbols similar to what is illustrated in FIGS. 38A and 38B is displayed when the "1" key is selected. Tilting the device and selecting the "1" key again may cause a corresponding symbol to be output. After a symbol has been output, letters and numbers may be output, as described above, until the "1" key is selected again to display the symbol menu. Fully inverting the device, shaking the device, or otherwise moving the device in a manner that is not interpreted as a tilt of the device generates another menu.

[0512] A first output signal is output if the angular displacement is within the first tilt region when the selection is received, where a second output signal is output if the angular displacement is within the second tilt region when the selection is received. Furthermore, a third or fourth output signal is output if the angular displacement is within the third or fourth tilt region, respectively, when the selection is received.

[0513] If a plurality of first-axis tilt regions are defined about the first axis and a plurality of second-axis tilt regions are defined about the second axis, the one of the first plurality of output signals may be also output based upon the plurality of first-axis tilt regions and/or the plurality of second-axis tilt regions. When the selection is received, a first output signal may be output if the first-axis component is within a first first-axis tilt region and if the second-axis component is within a first second-axis tilt region, a second output signal may be output if the first-axis component is within a second first-axis tilt region and if the second-axis component is within the first second-axis tilt region, a third output signal may be output if the first-axis component is within the second first-axis tilt region and if the second-axis component is within a second second-axis tilt region, and/or a fourth output signal may be output if the first-axis component is within the second first-axis tilt region and if the second-axis component is within the second second-axis tilt region.

[0514] Alternatively, in another aspect, when the selection is received, a first output signal may be output if the first component is within a first first-axis tilt region and if the second-axis component is within a first second-axis tilt region, a second output signal may be output if the first component is within the first first-axis tilt region and if the second-axis component is within a second second-axis tilt region, a third output signal may be output if the first component is within the first first-axis tilt region and if the second-axis component is within a third second-axis tilt region, a fourth output signal may be output if the first component is within a second first-axis tilt region and if the second-axis component is within the first second-axis tilt region, a fifth output signal may be output if the first component is within the second first-axis tilt region and if the second-axis component is within the second second-axis tilt region, a sixth output signal may be output if the first component is within the second first-axis tilt region and if the second-axis component is within the third second-axis tilt region, a seventh output signal may be output if the first component is within a third first-axis tilt region and if the second-axis component is within the first second-axis tilt region, an eighth output signal may be output if the first component is within the third first-axis tilt region and if the second-axis component is within the second second-axis tilt region, and/or a ninth output signal may be output if the first component is within the third first-axis tilt region and if the second-axis component is within the third second-axis tilt region.

[0515] The output signal is displayed (step ES310), and method E300 ends (step ES311). The output signal is displayed on a display, such as display E105. In an alternate aspect, the output signal is not displayed.

[0516] In the FIG. 32 implementation, device E500 also includes display E505, which is used to present a graphical user interface ("GUI") to a user of device E500. The GUI enables a user of device E500 to perform functions that require the user to enter text into device E500. For example, the user may identify an entry for a person within a phonebook stored on device E500 by entering a name of the person. As another example, the user may add an entry for a person to the phonebook by entering information describing the person, such as the person's name and one or more phone numbers used by the person. Furthermore, the GUI enables the user to specify a text message that is to be sent from device E500 or to specify another textual note that is to be stored on device E500. Device E500 also displays a GUI that enables a user to specify a text message.

[0517] Interpreting control selections based on device orientations when the control selections are made increases the number of operations that may be performed with a single control selection. For example, each control selection may be interpreted in a number of manners that is equal to the number of distinct orientations of the device that may be detected. Furthermore, the orientation of the device may indicate how selection of control that do not correspond to any characters may be interpreted. Therefore, a user may be enabled to quickly perform relatively complex operations simply by tilting the device and selecting controls. For example, selecting the "*" key while the device is rolled to the left may cause a particular mode of text entry (e.g., numbers only, all capital letters) to be used for text entry until the next time the "*" key is selected when the device is rolled to the left. In another aspect, the tilt sensor effectuates tilt scrolling, such that, upon receipt of the selection of a control, a user interface is scrolled corresponding to the direction of the tilt. A forward pitch occurring at the time of control selection, for example, would result in the user interface, or a menu item on the user interface, scrolling upward.

[0518] According to another general aspect, a computer program product, tangibly stored on a computer-readable medium, is recited. The computer program product is operable to cause a computer to perform operations including determining a neutral position of a device in relation to at least a first axis, the device including at least a first control associated with a first plurality of output signals, and measuring an angular displacement of the device about at least the first axis. The computer program product is also operable to cause a computer to perform operations including receiving a selection of the first control, and outputting one of the first plurality of output signals based at least upon the selection and the angular displacement.

[0519] Finally, although a number of implementations have been described or exemplified as a telephone device, it is contemplated that the concepts related herein are by no means limited to telephony, and are in fact applicable to a broad variety of devices, including any device in which the number of controls is minimized due to device design and layout restrictions. Sample devices include computer keyboards, remote controls, watches, joysticks or game controllers, or other computer input or consumer electronic devices.

[0520] Accordingly, a number of implementations have been described. Nevertheless, it will be understood that various modifications may be made. For example, elements of different implementations may be combined, supplemented, or removed to produce other implementations. Further, various technologies may be used, combined, and modified to produce an implemention, such technologies including, for example, a variety of digital electronic circuitry, hardware, software, firmware, integrated components, discrete components, processing devices, memory storage devices, communication devices, lenses, filters, display devices, and projection devices.

Game System

[0521] With reference to FIG. 39, a game system 39 according to some embodiments will be described. FIG. 39 is an external view illustrating the game system 39. In the following description, the game system 39 according to some embodiments includes a stationary game apparatus.

[0522] As shown in FIG. 39, the game system F1 includes a stationary game apparatus (hereinafter, referred to simply as a "game apparatus") F3, which is connected to a display (hereinafter, referred to as a "monitor") F2 of a home-use television receiver or the like having a speaker F2a via a connection cord, and a controller F7 for giving operation information to the game apparatus F3. The game apparatus F3 is connected to a receiving unit F6 via a connection terminal. The receiving unit F6 receives transmission data which is wirelessly transmitted from the controller F7. The controller F7 and the game apparatus F3 are connected to each other by wireless communication. On the game apparatus F3, an optical disc F4 as an example of an exchangeable information storage medium is detachably mounted. The game apparatus F3 includes a power ON/OFF switch, a game process reset switch, and an OPEN switch for opening a top lid of the game apparatus F3 on a top main surface of the game apparatus F3. When a player presses the OPEN switch, the lid is opened, so that the optical disc F4 can be mounted or dismounted.

[0523] Further, on the game apparatus F3, an external memory card F5 is detachably mounted when necessary. The external memory card F5 has a backup memory or the like mounted thereon for fixedly storing saved data or the like. The game apparatus F3 executes a game program or the like stored on the optical disc F4 and displays the result on the monitor F2 as a game image. The game apparatus F3 can also reproduce a state of a game played in the past using saved data stored in the external memory card F5 and display the game image on the monitor F2. A player playing with the game apparatus F3 can enjoy the game by operating the controller F7 while watching the game image displayed on the monitor F2.

[0524] The controller F7 wirelessly transmits the transmission data from a communication section F75 included therein (described later) to the game apparatus F3 connected to the receiving unit F6, using the technology of, for example, Bluetooth (registered trademark). The controller F7 has two control units, a core unit F70 and a subunit F76, connected to each other by a flexible connecting cable F79. The controller F7 is an operation means for mainly operating a player object appearing in a game space displayed on the monitor F2. The core unit F70 and the subunit F76 each includes an operation section such as a plurality of operation buttons, a key, a stick and the like. As described later in detail, the core unit F70 includes an imaging information calculation section F74 for taking an image viewed from the core unit F70. As an example of an imaging target of the imaging information calculation section F74, two LED modules F8L and F8R are provided in the vicinity of a display screen of the monitor F2. The LED modules F8L and F8R each outputs infrared light forward from the monitor F2. Although in the present embodiment the core unit F70 and the subunit F76 are connected to each other by the flexible cable, the subunit F76 may have a wireless unit, thereby eliminating the connecting cable F79. For example, the subunit F76 has a Bluetooth (registered trademark) unit as the wireless unit, whereby the subunit F76 can transmit operation data to the core unit F70.

[0525] Next, with reference to FIG. 40, a structure of the game apparatus F3 will be described. FIG. 40 is a functional block diagram of the game apparatus F3.

[0526] As shown in FIG. 40, the game apparatus F3 includes, for example, a RISC CPU (central processing unit) F30 for executing various types of programs. The CPU F30 executes a boot program stored in a boot ROM (not shown) to, for example, initialize memories including a main memory F33, and then executes a game program stored on the optical disc F4 to perform game process or the like in accordance with the game program. The CPU F30 is connected to a GPU (Graphics Processing Unit) F32, the main memory F33, a DSP (Digital Signal Processor) F34, and an ARAM (audio RAM) F35 via a memory controller F31. The memory controller F31 is connected to a controller I/F (interface) F36, a video I/F F37, an external memory I/F F38, an audio I/F F39, and a disc I/F F41 via a predetermined bus. The controller I/F F36, the video I/F F37, the external memory I/F F38, the audio I/F F39 and the disc I/F F41 are respectively connected to the receiving unit F6, the monitor F2, the external memory card F5, the speaker F2a, and a disc drive F40.

[0527] The GPU F32 performs image processing based on an instruction from the CPU F30. The GPU F32 includes, for example, a semiconductor chip for performing calculation process necessary for displaying 3D graphics. The GPU F32 performs the image process using a memory dedicated for image process (not shown) and a part of the storage area of the main memory F33. The GPU F32 generates game image data and a movie to be displayed on the monitor F2 using such memories, and outputs the generated data or movie to the monitor F2 via the memory controller F31 and the video I/F F37 as necessary.

[0528] The main memory F33 is a storage area used by the CPU F30, and stores a game program or the like necessary for processing performed by the CPU F30 as necessary. For example, the main memory F33 stores a game program read from the optical disc F4 by the CPU F30, various types of data or the like. The game program, the various types of data or the like stored in the main memory F33 are executed by the CPU F30.

[0529] The DSP F34 processes sound data or the like generated by the CPU F30 during the execution of the game program. The DSP F34 is connected to the ARAM F35 for storing the sound data or the like. The ARAM F35 is used when the DSP F34 performs a predetermined process (for example, storage of the game program or sound data already read). The DSP F34 reads the sound data stored in the ARAM F35, and outputs the sound data to the speaker F2a included in the monitor F2 via the memory controller F31 and the audio I/F F39.

[0530] The memory controller F31 comprehensively controls data transmission, and is connected to the various I/Fs described above. The controller I/F F36 includes, for example, four controller I/Fs F36a, F36b, F36c and F36d, and communicably connects the game apparatus F3 to an external device which is engageable via connectors of the controller I/Fs F36a, F36b, F36c and F36d. For example, the receiving unit F6 is engaged with such a connector and is connected to the game apparatus F3 via the controller I/F F36. As described above, the receiving unit F6 receives the transmission data from the controller F7 and outputs the transmission data to the CPU F30 via the controller I/F F36. The video I/F F37 is connected to the monitor F2. The external memory I/F F38 is connected to the external memory card F5 and is accessible to a backup memory or the like provided in the external memory card F5. The audio I/F F39 is connected to the speaker F2a built in the monitor F2 such that the sound data read by the DSP F34 from the ARAM F35 or sound data directly outputted from the disc drive F40 can be outputted from the speaker F2a. The disc I/F F41 is connected to the disc drive F40. The disc drive F40 reads data stored at a predetermined reading position of the optical disc F4 and outputs the data to a bus of the game apparatus F3 or the audio I/F F39.

[0531] Next, with reference to FIGS. 41 and 42, the controller F7 will be described. FIG. 41 is a perspective view illustrating an outer appearance of the controller F7. FIG. 42 is a perspective view illustrating a state of the connecting cable F79 of the controller F7 shown in FIG. 41 being connected to or disconnected from the core unit F70.

[0532] As shown in FIG. 41, the controller F7 includes the core unit F70 and the subunit F76 connected to each other by the connecting cable F79. The core unit F70 has a housing F71 including a plurality of operation sections F72. The subunit F76 has a housing F77 including a plurality of operation sections F78. The core unit F70 and the subunit F76 are connected to each other by the connecting cable F79.

[0533] As shown in FIG. 42, the connecting cable F79 has a connector F791 detachably connected to the connector F73 of the core unit F70 at one end thereof, and the connecting cable F79 is fixedly connected to the subunit F76 at the other end thereof. The connector F791 of the connecting cable F79 is engaged with the connector F73 provided at the rear surface of the core unit F70 so as to connect the core unit F70 and the subunit F76 to each other by the connecting cable F79.

[0534] With reference to FIGS. 43 and 44, the core unit F70 will be described. FIG. 43 is a perspective view of the core unit F70 as seen from the top rear side thereof. FIG. 44 is a perspective view of the core unit F70 as seen from the bottom front side thereof.

[0535] As shown in FIGS. 43 and 44, the core unit F70 includes the housing F71 formed by plastic molding or the like. The housing F71 has a generally parallelepiped shape extending in a longitudinal direction from front to rear. The overall size of the housing F71 is small enough to be held by one hand of an adult or even a child.

[0536] At the center of a front part of a top surface of the housing F71, a cross key F72a is provided. The cross key F72a is a cross-shaped four-direction push switch. The cross key F72a includes operation portions corresponding to the four directions (front, rear, right and left) represented by arrows, which are respectively located on cross-shaped projecting portions arranged at intervals of 90 degrees. The player selects one of the front, rear, right and left directions by pressing one of the operation portions of the cross key F72a. Through an operation on the cross key F72a, the player can, for example, instruct a direction in which a player character or the like appearing in a virtual game world is to move or a direction in which the cursor is to move.

[0537] Although the cross key F72a is an operation section for outputting an operation signal in accordance with the aforementioned direction input operation performed by the player, such an operation section may be provided in another form. For example, the cross key F72a may be replaced with a composite switch including a push switch including a ring-shaped four-direction operation section and a center switch provided at the center thereof. Alternatively, the cross key F72a may be replaced with an operation section which includes an inclinable stick projecting from the top surface of the housing F71 and outputs an operation signal in accordance with the inclining direction of the stick. Still alternatively, the cross key F72amay be replaced with an operation section which includes a disc-shaped member horizontally slidable and outputs an operation signal in accordance with the sliding direction of the disc-shaped member. Still alternatively, the cross key F72a may be replaced with a touch pad. Still alternatively, the cross key F72a may be replaced with an operation section which includes switches representing at least four directions (front, rear, right and left) and outputs an operation signal in accordance with the switch pressed by the player.

[0538] Behind the cross key F72a on the top surface of the housing F71, a plurality of operation buttons F72b, F72c, F72d, F72e, F72f and F72g are provided. The operation buttons F72b, F72c, F72d, F72e, F72f and F72g are each an operation section for outputting a respective operation signal assigned to the operation buttons F72b, F72c, F72d, F72e, F72f or F72g when the player presses a head thereof. For example, the operation buttons F72b, F72c, and F72d are assigned with functions of a first button, a second button, and an A button. Further, the operation buttons F72e, F72f and F72g are assigned with functions of a minus button, a home button and a plus button, for example. The operation buttons F72b, F72c, F72d, F72e, F72f and F72g are assigned with respective functions in accordance with the game program executed by the game apparatus F3. In an exemplary arrangement shown in FIG. 43, the operation buttons F72b, F72c and F72d are arranged in a line at the center in the front-rear direction on the top surface of the housing F71. The operation buttons F72e, F72f and F72g are arranged in a line in the left-right direction between the operation buttons F72b and F72d on the top surface of the housing F71. The operation button F72f has a top surface thereof buried in the top surface of the housing F71, so as not to be inadvertently pressed by the player.

[0539] In front of the cross key F72a on the top surface of the housing F71, an operation button F72h is provided. The operation button F72h is a power switch for remote-controlling the power of the game apparatus 3 to be on or off. The operation button F72h also has a top surface thereof buried in the top surface of the housing F71, so as not to be inadvertently pressed by the player.

[0540] Behind the operation button F72c on the top surface of the housing F71, a plurality of LEDs F702 are provided. The controller F7 is assigned a controller type (number) so as to be distinguishable from the other controllers F7. For example, the LEDs F702 are used for informing the player of the controller type which is currently set to controller F7 that he or she is using. Specifically, when the core unit F70 transmits the transmission data to the receiving unit F6, one of the plurality of LEDs F702 corresponding to the controller type is lit up.

[0541] On the top surface of the housing F71, a sound hole for externally outputting a sound from a speaker F706 shown in FIG. 45, which will be described below, is provided between the operation buttons F72e, F72f, and F72g and the operation button F72b.

[0542] On a bottom surface of the housing F71, a recessed portion is formed. As described later in detail, the recessed portion is formed at a position at which an index finger or middle finger of the player is located when the player holds the core unit F70. On a rear slope surface of the recessed portion, an operation button F72i is provided. The operation button F72i is an operation section acting as, for example, a B button. The operation button F72i is used, for example, as a trigger switch in a shooting game, or for attracting attention of a player object to a predetermined object.

[0543] On a front surface of the housing F71, an image pickup element F743 included in the imaging information calculation section F74 is provided. The imaging information calculation section F74 is a system for analyzing image data taken by the core unit F70 and detecting for the centroid, the size and the like of an area having a high brightness in the image data. The imaging information calculation section F74 has, for example, a maximum sampling period of about 200 frames/sec., and therefore can trace and analyze even a relatively fast motion of the core unit F70. The imaging information calculation section F74 will be described later in detail. On a rear surface of the housing F71, the connector F73 is provided. The connector F73 is, for example, a 32-pin edge connector, and is used for engaging and connecting the core unit F70 with the connector F791 of the connecting cable F79.

[0544] With reference to FIGS. 45 and 46, an internal structure of the core unit F70 will be described. FIG. 45 is a perspective view illustrating, as seen from the rear side of the core unit F70, a state where an upper casing (a part of the housing F71) of the core unit F70 is removed. FIG. 46 is a perspective view illustrating, as seen from the front side of the core unit F70, a state where a lower casing (a part of the housing F71) of the core unit F70 is removed. FIG. 46 is a perspective view illustrating a reverse side of a substrate F700 shown in FIG. 45.

[0545] As shown in FIG. 45, the substrate F700 is fixed inside the housing F71. On a top main surface of the substrate F700, the operation buttons F72a, F72b, F72c, F72d, F72e, F72f, F72g and F72h, an acceleration sensor F701, the LEDs F702, an antenna F754 and the like are provided. These elements are connected to a micro computer F751 (see FIGS. 46 and 55) and the like via lines (not shown) formed on the substrate F700 and the like. The wireless module F753 not shown (see FIG. 55) and the antenna F754 allow the core unit F70 to act as a wireless controller. The quartz oscillator F703 not shown, which is provided in the housing F71, generates a reference clock of the micro computer F751 described later. On the top main surface of the substrate F700, the speaker F706 and an amplifier F708 are provided. The acceleration sensor F701 is provided near the edge of the substrate F700 offset from the center thereof. Therefore, a change of a direction of the gravitational acceleration and an acceleration containing a centrifugal force component can be detected based on a rotation of the core unit F70 about the longitudinal direction thereof, so that a predetermined calculation is used to determine the rotation of the core unit F70 with favorable accuracy based on the acceleration data having been detected.

[0546] As shown in FIG. 46, at a front edge of a bottom main surface of the substrate F700, the imaging information calculation section F74 is provided. The imaging information calculation section F74 includes an infrared filter F741, a lens F742, the image pickup element F743 and an image processing circuit F744 located in this order from the front surface of the core unit F70 on the bottom main surface of the substrate F700. At a rear edge of the bottom main surface of the substrate F700, the connector F73 is attached. Further, a sound IC F707 and the micro computer F751 are provided on the bottom main surface of the substrate F700. The sound IC F707, which is connected to the micro computer F751 and the amplifier F708 via lines formed on the substrate F700 and the like, outputs a sound signal to the speaker F706 via the amplifier F708 based on the sound data transmitted from the game apparatus F3. On the bottom main surface of the substrate F700, a vibrator F704 is provided. The vibrator F704 is, for example, a vibration motor or a solenoid. The core unit F70 is vibrated by an actuation of the vibrator F704, and the vibration is conveyed to the player's hand holding the core unit F70. Thus, a so-called vibration-feedback game is realized. The vibrator F704 is disposed slightly toward the front of the housing F71, thereby allowing the housing F71 held by the player to strongly vibrate, that is, allowing the player to easily feel the vibration.

[0547] With reference to FIGS. 47 to 50, the subunit F76 will be described. FIG. 47 is a perspective view illustrating a first example of the subunit F76. FIG. 48 is a perspective view illustrating a state where an upper casing (a part of the housing F77) of the subunit F76 shown in FIG. 47 is removed. FIG. 49A is a top view illustrating a second example of the subunit F76. FIG. 49B is a bottom view illustrating the second example of the subunit F76. FIG. 49C is a left side view illustrating the second example of the subunit F76. FIG. 50 is a perspective view illustrating the second example of the subunit F76 as seen from the top front side thereof.

[0548] As shown in FIG. 47, the subunit F76 includes the housing F77 formed by, for example, plastic molding. The housing F77 extends in a longitudinal direction from front to rear, and has a streamline solid shape including a head which is a widest portion in the subunit F76. The overall size of the subunit F76 is small enough to be held by one hand of an adult or even a child.

[0549] In the vicinity of the widest portion on the top surface of the housing F77, a stick F78a is provided. The stick F78a is an operation section which includes an inclinable stick projecting from the top surface of the housing F77 and outputs an operation signal in accordance with the inclining direction of the stick. For example, a player can arbitrarily designate a direction and a position by inclining a tip of the stick in any direction of 360 degrees, whereby the player can instruct a direction in which a player character or the like appearing in a virtual game world is to move, or can instruct a direction in which a cursor is to move.

[0550] In front of the housing F77 of the subunit F76, a plurality of operation buttons F78d and F78e are provided. The operation buttons F78d and F78e are each an operation section for outputting a respective operation signal assigned to the operation buttons F78d and F78e when the player presses a head thereof. For example, the operation buttons F78d and F78e are assigned with functions of an X button and a Y button, for example. The operation buttons F78d and F78e are assigned with respective functions in accordance with the game program executed by the game apparatus F3. In an exemplary arrangement shown in FIG. 47, the operation buttons F78d and F78e are aligned from the top to bottom on the front surface of the housing F77.

[0551] In FIG. 48, a substrate is fixed in the housing F77. The stick F78a, an acceleration sensor F761 and the like are provided on the top main surface of the substrate. The stick F78a, the acceleration sensor F761 and the like are connected to the connecting cable F79 via lines (not shown) formed on the substrate and the like.

[0552] As shown in FIGS. 49A, 49B, 49C and 50, the subunit F76 of the second example includes the housing F77, the stick F78a, the operation buttons F78d and F78e as in the case of the subunit F76 of the first example, and the subunit F76 of the second example has the operation buttons F78b and F78c on the top surface of the housing F77.

[0553] Behind the stick F78a on the top surface of the housing F77, the subunit F76 of the second example has a plurality of operation buttons F78b and F78c. The operation buttons F78b and F78c are each an operation section for outputting a respective operation signal assigned to the operation buttons F78b and F78c when the player presses a head thereof. The operation buttons F78b and F78c are assigned with respective functions in accordance with the game program executed by the game apparatus F3. In an exemplary arrangement shown in FIGS. 49A, 49B, and 49C and 50, the operation buttons F78b and F78c are arranged in a line at the center of the top surface of the housing F77 in the left-right direction.

[0554] Although the stick F78a is an operation section for outputting an operation signal in accordance with a direction input operation performed by the player as described above, such an operation section may be provided in another form. Hereinafter, with reference to FIGS. 51 to 54, a first through a fifth exemplary modifications, each of which includes the subunit F76 of the second example having an operation section for outputting an operation signal in accordance with the direction input operation, will be described.

[0555] As the first exemplary modification, as shown in FIG. 51, the subunit F76 may include a cross key F78f similar to the cross key F72a of the core unit F70 instead of the stick F78a. As the second exemplary modification, as shown in FIG. 52, the subunit F76 may include a slide pad F78g which includes a disc-shaped member horizontally slidable and outputs an operation signal in accordance with the sliding direction of the disc-shaped member, instead of the stick F78a. As the third exemplary modification, as shown in FIG. 53, the subunit F76 may include a touch pad F78h instead of the stick F78a. As the fourth exemplary modification, as shown in FIG. 54, the subunit F76 may include an operation section which has buttons F78i, F78j, F78k, and F78l representing at least four directions (front, rear, right and left), respectively, and outputs an operation signal in accordance with the button (F78i, F78j, F78k, or F78l) pressed by a player, instead of the stick F78a. As the fifth exemplary modification, the subunit F76 may include a composite switch including a push switch having a ring-shaped four-direction operation section and a center switch provided at the center thereof, instead of the stick F78a.

[0556] Next, with reference to FIG. 55, an internal structure of the controller F7 will be described. FIG. 55 is a block diagram illustrating the structure of the controller F7.

[0557] As shown in FIG. 55, the core unit F70 includes the communication section F75 in addition to the operation section F72, the imaging information calculation section F74, the acceleration sensor F701, the speaker F706, the sound IC F707, and the amplifier F708 as described above. Further, the subunit F76, which has the operation section F78 and the acceleration sensor F761 as described above, is connected to the micro computer F751 via the connecting cable F79 and the connectors F791 and F73.

[0558] The imaging information calculation section F74 includes the infrared filter F741, the lens F742, the image pickup element F743 and the image processing circuit F744. The infrared filter F741 allows only infrared light to pass therethrough, among light incident on the front surface of the core unit F70. The lens F742 collects the infrared light which has passed through the infrared filter F741 and outputs the infrared light to the image pickup element F743. The image pickup element F743 is a solid-state imaging device such as, for example, a CMOS sensor or a CCD. The image pickup element F743 takes an image of the infrared light collected by the lens F742. Accordingly, the image pickup element F743 takes an image of only the infrared light which has passed through the infrared filter F741 and generates image data. The image data generated by the image pickup element F743 is processed by the image processing circuit F744. Specifically, the image processing circuit F744 processes the image data obtained from the image pickup element F743, identifies a spot thereof having a high brightness, and outputs process result data representing the identified position coordinates and size of the area to the communication section F75. The imaging information calculation section F74 is fixed to the housing F71 of the core unit F70. The imaging direction of the imaging information calculation section F74 can be changed by changing the direction of the housing F71. The housing F71 is connected to the subunit F76 by the flexible connecting cable F79, and therefore the imaging direction of the imaging information calculation section F74 is not changed by changing the direction and position of the subunit F76. As described later in detail, a signal can be obtained in accordance with the position and the motion of the core unit F70 based on the process result data outputted by the imaging information calculation section F74.

[0559] The core unit F70 preferably includes a three-axis acceleration sensor F701. Further, the subunit F76 preferably includes a three-axis acceleration sensor F761. The three axis acceleration sensors F701 and F761 each detects for a linear acceleration in three directions, i.e., the up/down direction, the left/right direction, and the forward/backward direction. Alternatively, a two axis acceleration detection means which detects for only a linear acceleration along each of the up/down and left/right directions (or other pair of directions) may be used in another embodiment depending on the type of control signals used in the game process. For example, the three axis acceleration sensors F701 and F761 or the two axis acceleration sensors F701 and F761 may be of the type available from Analog Devices, Inc. or STMicroelectronics N.V. Preferably, each of the acceleration sensors F701 and F761 is of an electrostatic capacitance (capacitance-coupling) type that is based on silicon micro-machined MEMS (Micro Electro Mechanical Systems) technology. However, any other suitable acceleration detection technology (e.g., piezoelectric type or piezoresistance type) now existing or later developed may be used to provide the three axis acceleration sensors F701 and F761 or two axis acceleration sensors F701 and F761.

[0560] As one skilled in the art understands, the acceleration detection means, as used in the acceleration sensors F701 and F761, are capable of detecting for only acceleration (linear acceleration) along a straight line corresponding to each axis of the acceleration sensor. In other words, each of the direct outputs of the acceleration sensors F701 and F761 is limited to signals indicative of linear acceleration (static or dynamic) along each of the two or three axes thereof. As a result, the acceleration sensors F701 and F761 cannot directly detect movement along a non-linear (e.g. arcuate) path, rotation, rotational movement, angular displacement, tilt, position, attitude or any other physical characteristic.

[0561] However, through additional processing of the acceleration signals output from each of the acceleration sensors F701 and F761, additional information relating to the core unit F70 and the subunit F76 can be inferred or calculated, as one skilled in the art will readily understand from the description herein. For example, by detecting static acceleration (i.e., gravity), the outputs of the acceleration sensors F701 and F761 can be used to infer tilt of the object (core unit F70 or subunit F76) relative to the gravity vector by correlating tilt angles with detected acceleration. In this way, the acceleration sensors F701 and F761 can be used in combination with the micro computer F751 (or another processor) to determine tilts, attitudes or positions of the core unit F70 and the subunit F76. Similarly, various movements and/or positions of the core unit F70 and the subunit F76 can be calculated or inferred through processing of the acceleration signals generated by the acceleration sensors F701 and F761 when the core unit F70 containing the acceleration sensor F701 or the subunit F76 containing the acceleration sensor F761 is subjected to dynamic accelerations by, for example, the hand of a user, as described herein. In another embodiment, each of the acceleration sensors F701 and F761 may include an embedded signal processor or other type of dedicated processor for performing any desired processing of the acceleration signals outputted from the acceleration detection means prior to outputting signals to micro computer F751. For example, the embedded or dedicated processor could convert the detected acceleration signal to a corresponding tilt angle when the acceleration sensor is intended to detect static acceleration (i.e., gravity). Data representing the acceleration detected by each of the acceleration sensors F701 and F761 is outputted to the communication section F75.

[0562] In another exemplary embodiment, at least one of the acceleration sensors F701 and F761 may be replaced with a gyro-sensor of any suitable technology incorporating, for example, a rotating or vibrating element. Exemplary MEMS gyro-sensors that may be used in this embodiment are available from Analog Devices, Inc. Unlike the acceleration sensors F701 and F761, a gyro-sensor is capable of directly detecting rotation (or angular rate) around at least one axis defined by the gyroscopic element therein. Thus, due to the fundamental differences between a gyro-sensor and an acceleration sensor, corresponding changes need to be made to the processing operations that are performed on the output signals from these devices depending on which device is selected for a particular application.

[0563] More specifically, when the tilt or attitude is calculated using a gyro-sensor instead of the acceleration sensor, significant changes are necessary. Specifically, when using a gyro-sensor, the value of the tilt is initialized at the start of the detection. Then, data on the angular rate which is output from the gyro-sensor is integrated. Next, a change amount in tilt from the value of the tilt initialized is calculated. In this case, the calculated tilt corresponds to an angle. In contrast, when the acceleration sensor calculates the tilt, the tilt is calculated by comparing the value of the gravitational acceleration of each axial component with a predetermined reference. Therefore, the calculated tilt can be represented as a vector. Thus, without initialization, an absolute direction can be determined with an acceleration detection means. The type of the value calculated as the tilt is also very different between a gyro sensor and an acceleration sensor; i.e., the value is an angle when a gyro sensor is used and is a vector when an acceleration sensor is used. Therefore, when a gyro sensor is used instead of an acceleration sensor or vice versa, data on tilt also needs to be processed through a predetermined conversion taking into account the fundamental differences between these two devices. Due to the fact that the nature of gyroscopes is known to one skilled in the art, as well as the fundamental differences between the acceleration detection means and the gyroscope, further details are not provided herein. While a gyro-sensor is advantageous in that a rotation can be directly detected, an acceleration sensor is generally more cost effective when used in connection with the controller described herein.

[0564] The communication section F75 includes the micro computer F751, a memory F752, the wireless module F753 and the antenna F754. The micro computer F751 controls the wireless module F753 for wirelessly transmitting the transmission data while using the memory F752 as a storage area during the process. Further, the micro computer F751 controls the sound IC F707 and the vibrator F704 based on data from the game apparatus F3 having been received by the wireless module F753 via the antenna F754. The sound IC F707 processes sound data transmitted from the game apparatus F3 via the communication section F75, and the like.

[0565] Data from the core unit F70 including an operation signal (core key data) from the operation section F72, acceleration signals (core acceleration data) from the acceleration sensor F701, and the process result data from the imaging information calculation section F74 are outputted to the micro computer F751. An operation signal (sub key data) from the operation section F78 of the subunit F76 and acceleration signals (sub acceleration data) from the acceleration sensor F761 are outputted to the micro computer F751 via the connecting cable F79. The micro computer F751 temporarily stores the input data (core key data, sub key data, core acceleration data, sub acceleration data, and process result data) in the memory F752 as the transmission data which is to be transmitted to the receiving unit F6. The wireless transmission from the communication section F75 to the receiving unit F6 is performed periodically at a predetermined time interval. Since game process is generally performed at a cycle of 1/60 sec., data needs to be collected and transmitted at a cycle of a shorter time period. Specifically, the game process unit is 16.7 ms (1/60 sec.), and the transmission interval of the communication section F75 structured using the Bluetooth(R) technology is 5 ms. At the transmission timing to the receiving unit F6, the micro computer F751 outputs the transmission data stored in the memory F752 as a series of operation information to the wireless module F753. The wireless module F753 uses, for example, the Bluetooth(R) technology to modulate the operation information onto a carrier wave of a predetermined frequency, and radiates the low power radio wave signal from the antenna F754. Thus, the core key data from the operation section F72 included in the core unit F70, the sub key data from the operation section F78 included in the subunit F76, the core acceleration data from the acceleration sensor F701 included in the core unit F70, the subacceleration data from the acceleration sensor F761 included in the subunit F76, and the process result data from the imaging information calculation section F74 are modulated onto the low power radio wave signal by the wireless module F753 and radiated from the core unit F70. The receiving unit F6 of the game apparatus F3 receives the low power radio wave signal, and the game apparatus F3 demodulates or decodes the low power radio wave signal to obtain the series of operation information (the core key data, the sub key data, the core acceleration data, the sub acceleration data and the process result data). Based on the obtained operation information and the game program, the CPU F30 of the game apparatus F3 performs the game process. In the case where the communication section F75 is structured using the Bluetooth(R) technology, the communication section F75 can have a function of receiving transmission data which is wirelessly transmitted from other devices.

[0566] As shown in FIG. 56, in order to play a game using the controller F7 with the game system F1, a player holds the core unit F70 with one hand (for example, a right hand) (see FIGS. 57 and 58), and holds the subunit F76 with the other hand (for example, a left hand) (see FIG. 60). The player holds the core unit F70 so as to point the front surface of the core unit F70 (that is, a side having an entrance through which light is incident on the imaging information calculation section F74 taking an image of the light) to the monitor F2. On the other hand, two LED modules F8L and F8R are provided in the vicinity of the display screen of the monitor F2. The LED modules F8L and F8R each outputs infrared light forward from the monitor F2.

[0567] When a player holds the core unit F70 so as to point the front surface thereof to the monitor F2, infrared lights outputted by the two LED modules F8L and F8R are incident on the imaging information calculation section F74. The image pickup element F743 takes images of the infrared lights incident through the infrared filter F741 and the lens F742, and the image processing circuit F744 processes the taken images. The imaging information calculation section F74 detects infrared components outputted by the LED modules F8L and F8R so as to obtain positions and area information of the LED modules F8L and F8R. Specifically, the imaging information calculation section F74 analyzes image data taken by the image pickup element F743, eliminates images which do not represent the infrared lights outputted by the LED modules F8L and F8R from the area information, and identifies points each having a high brightness as positions of the LED modules F8L and F8R. The imaging information calculation section F74 obtains position coordinates, coordinates of the centroid, and the like of each of the identified points having the high brightness and outputs the same as the process result data. When such process result data is transmitted to the game apparatus F3, the game apparatus F3 can obtain, based on the position coordinates and the coordinates of the centroid, operation signals relating to the motion, attitude, position and the like of the imaging information calculation section F74, that is, the core unit F70, with respect to the LED modules F8L and F8R. Specifically, the position having a high brightness in the image obtained through the communication section F75 is changed in accordance with the motion of the core unit F70, and therefore a direction input or coordinate input is performed in accordance with the position having the high brightness being changed, thereby enabling a direction input or a coordinate input to be performed along the moving direction of the core unit F70.

[0568] Thus, the imaging information calculation section F74 of the core unit F70 takes images of stationary markers (infrared lights from the two LED modules F8L and F8R in the present embodiment), and therefore the game apparatus F3 can use the process result data relating to the motion, attitude, position and the like of the core unit F70 in the game process, whereby an operation input, which is different from an input made by pressing an operation button or using an operation key, is further intuitively performed. As described above, since the markers are provided in the vicinity of the display screen of the monitor F2, the motion, attitude, position and the like of the core unit F70 with respect to the display screen of the monitor F2can be easily calculated based on positions from the markers. That is, the process result data used for obtaining the motion, attitude, position and the like of the core unit F70 can be used as operation input immediately applied to the display screen of the monitor F2.

[0569] With reference to FIGS. 57 and 58, a state of a player holding the core unit F70 with one hand will be described. FIG. 57 shows an exemplary state of a player holding the core unit F70 with a right hand as seen from the front surface side of the core unit F70. FIG. 58 shows an exemplary state of a player holding the core unit F70 with a right hand as seen from the left side of the core unit F70.

[0570] As shown in FIGS. 57 and 58, the overall size of the core unit F70 is small enough to be held by one hand of an adult or even a child. When the player puts a thumb on the top surface of the core unit F70 (for example, near the cross key F72a), and puts an index finger in the recessed portion on the bottom surface of the core unit F70 (for example, near the operation button F72i), the light entrance of the imaging information calculation section F74 on the front surface of the core unit F70 is exposed forward to the player. It should be understood that also when the player holds the core unit F70 with a left hand, the holding state is the same as that described for the right hand.

[0571] Thus, the core unit F70 allows a player to easily operate the operation section F72 such as the cross key F72a or the operation button F72i while holding the core unit F70 with one hand. Further, when the player holds the core unit F70 with one hand, the light entrance of the imaging information calculation section F74 on the front surface of the core unit F70 is exposed, whereby the light entrance can easily receive infrared lights from the aforementioned two LED modules F8L and F8R. That is, the player can hold the core unit F70 with one hand without preventing the imaging information calculation section F74 from functioning. That is, when the player moves his or her hand holding the core unit F70 with respect to the display screen, the core unit F7o can further perform an operation input enabling a motion of the player's hand to directly act on the display screen.

[0572] As shown in FIG. 59, the LED modules F8L and F8R each has a viewing angle .theta.1. The image pickup element F743 has a viewing angle .theta.2. For example, the viewing angle .theta.1 of the LED modules F8L and F8R is 34 degrees (half-value angle), and the viewing angle .theta.2 of the image pickup element F743 is 41 degrees. When both the LED modules F8L and F8R are in the viewing angle .theta.2 of the image pickup element F743, and the image pickup element F743 is in the viewing angle .theta.1 of the LED module F8L and the viewing angle .theta.1 of the LED module F8R, the game apparatus F3 determines a position of the core unit F70 using positional information relating to the point having high brightness of the two LED modules F8L and F8R.

[0573] When either the LED module F8L or LED module F8R is in the viewing angle .theta.2 of the image pickup element F743, or when the image pickup element F743 is in either the viewing angle .theta.1 of the LED module F8L or the viewing angle .theta.1

of the LED module F8R, the game apparatus F3 determines a position of the core unit F70 using the positional information relating to the point having high brightness of the LED module F8L or the LED module F8R.

[0574] As described above, the tilt, attitude or position of the core unit F70 can be determined based on the output (core acceleration data) from the acceleration sensor F701 of the core unit F70. That is, the core unit F70 functions as an operation input means for performing an operation in accordance with a player moving a hand holding the core unit F70, for example, upward, downward, leftward, or rightward.

[0575] Next, with reference to FIG. 60, a state of a player holding the subunit F76 with one hand will be described. FIG. 60 shows an exemplary state of a player holding the subunit F76 with a left hand as seen from the right side of the subunit F76.

[0576] As shown in FIG. 60, the overall size of the subunit F76 is small enough to be held by one hand of an adult or even a child. For example, a player can put a thumb on the top surface of the subunit F76 (for example, near the stick F78a), put an index finger on the front surface of the subunit F76 (for example, near the operation buttons F78d and F78e), and put a middle finger, a ring finger and a little finger on the bottom surface of the subunit F76 so as to hold the subunit F76. It should be understood that also when the player holds the subunit F76 with a right hand, the holding state is similar to that described for the left hand. Thus, the subunit F76 allows the player to easily operate the operation section F78 such as the stick F78a and the operation buttons F78d and F78e while holding the subunit F76 with one hand.

[0577] As described above, the tilt, attitude or position of the subunit F76 can be determined based on the output (sub acceleration data) from the acceleration sensor F761 of the subunit F76. That is, the subunit F76 functions as an operation input means for performing an operation in accordance with the player moving a hand holding the subunit F76, for example, upward, downward, leftward, and rightward.

[0578] Here, an exemplary game played using the aforementioned controller F7 will be described. As a first example, a shooting game played using the controller F7 will be described. FIG. 61 is a diagram illustrating an exemplary game image displayed on the monitor F2 when the game apparatus F3 executes the shooting game.

[0579] As shown in FIG. 61, a portion of a three-dimensional virtual game space S is displayed on the display screen of the monitor F2. As a game object acting in accordance with an operation of the controller F7, a portion of the player character P and a portion of a gun G held by the player character P are displayed on the display screen. Moreover, the virtual game space S displayed on the display screen represents a field of front vision of the player character P, and for example an opponent character E is displayed as a shooting target in FIG. 61. A target indicating a position at which the player character P shoots the gun G is displayed on the display screen as the target cursor T.

[0580] In the shooting game having such a game image displayed on the monitor F2, a player operates the core unit F70 with one hand and operates the subunit F76 with the other hand as shown in FIG. 18 so as to play the game. For example, when the player inclines the stick F78a (see FIGS. 49A, 49B, 49C and 50) on the subunit F76, the player character P is moved in the virtual game space S in accordance with the inclining direction. Further, when the player moves his or her hand holding the core unit F70 with respect to the display screen, the target cursor T is moved in accordance with the motion, attitude, position and the like of the core unit F70 with respect to the monitor F2 (LED modules F8L and F8R). When the player presses the operation button F72i (shown in FIG. 44) on the core unit F70, the player character P shoots the gun G at the target cursor T.

[0581] That is, while the player uses the stick F78a on the subunit F76 so as to instruct the player character P to move, the player can operate the core unit F70 as if the core unit F70 is a gun for the shooting game, thereby enhancing enjoyment in playing a shooting game. The player can perform an operation of moving the player character P and an operation of moving the target cursor T by using respective units held by different hands, whereby the player can perform the respective operations as independent ones. For example, since the virtual game space S displayed on the display screen is changed in accordance with the movement of the player character P, it is sometimes difficult to keep the target positioned near a position observed by the player in the virtual game space S because, for example, the player may be paying attention to the opponent character E suddenly jumping into the virtual game space S. However, while the player is moving the player character P with one hand (for example, a thumb of a left hand), the player can control a motion of the arm (for example, a right arm) which is not used for moving the player character P such that the core unit F70 has its front surface pointed to the observed position, thereby substantially enhancing flexibility for operating the controller F7 and increasing the reality of the shooting game. Further, in order to move the target cursor T, the player moves the controller. However, the operation of moving the controller does not hinder the player from performing a direction instruction operation for moving the player character P, thereby enabling the player to stably perform the two direction instruction operations. That is, by using the controller F7, the player can freely use his or her left and right hands and can perform a new operation having increased flexibility, which cannot be achieved using a physically single controller.

[0582] In a second example, a player inclines the stick F78a on the subunit F76 so as to move the player character Pin the virtual game space S in accordance with the inclining direction as in the first example. The player moves a hand holding the core unit F70 with respect to the display screen so as to move a sight point of a virtual camera in accordance with a position of the core unit F70 with respect to the monitor F2 (LED modules F8L and F8R). These operations allow the player to observe a position to which the core unit F70 is pointed in the virtual game space S while operating the stick F78a on the subunit F76 so as to instruct the player character P to move.

[0583] In the above description, the controller F7 and the game apparatus F3 are connected to each other by wireless communication. However, the controller F7 and the game apparatus F3 may be electrically connected to each other by a cable. In this case, the cable connected to the core unit F70 is connected to a connection terminal of the game apparatus F3.

[0584] Moreover, in the present embodiment, only the core unit F70 among the core unit F70 and the subunit F76 of the controller F7 has the communication section F75. However, the subunit F76 may have the communication section for wirelessly transmitting the transmission data to the receiving unit F6. Further, both the core unit F70 and the subunit F76 may have the respective communication sections. For example, the respective communication sections included in the core unit F70 and the subunit F76 may wirelessly transmit the transmission data to the receiving unit F6, or the communication section of the subunit F76 may wirelessly transmit the transmission data to the communication section F75 of the core unit F70, and the communication section F75 of the core unit F70 may wirelessly transmit, to the receiving unit F6, the received transmission data from the subunit F76 and the transmission data of the core unit F70. In these cases, the connecting cable F79 for electrically connecting between the core unit F70 and the subunit F76 can be eliminated.

[0585] In the above description, the receiving unit F6 connected to the connection terminal of the game apparatus F3 is used as a receiving means for receiving transmission data which is wirelessly transmitted from the controller F7. Alternatively, the receiving means may be a receiving module built in the game apparatus F3. In this case, the transmission data received by the receiving module is outputted to the CPU F30 via a predetermined bus.

[0586] Although in the present embodiment the imaging information calculation section F74 included in the core unit F70 is described as an example of a determining section for outputting a signal (process result data) in accordance with a motion of the core unit F70 body, the imaging information calculation section F74 may be provided in another form. For example, the core unit F70 may include the acceleration sensor F701 as described above, or may include a gyro sensor. The acceleration sensor or the gyro sensor can be used to determine a motion or attitude of the core unit F70, and, therefore, can be used as a determining section for outputting a signal in accordance with the motion of the core unit F70 body using the detection signal for the motion or attitude. In this case, the imaging information calculation section F74 may be eliminated from the core unit F70, or sensor and the imaging information calculation section can be used in combination.

[0587] Further, although in the present embodiment only the core unit F70 includes the imaging information calculation section F74, the subunit F76 may also include a similar imaging information calculation section.

[0588] Further, when the controller F7 includes a plurality of units, each of which may have a plurality of operation means such as the imaging information calculation section, the acceleration sensor, the gyro sensor, the stick, the cross key, and the operation button, various combination of the operation means can realize various controllers. Here, the operation means included in the core unit F70 and the subunit F76 are classified into an operation means A and an operation means B. The operation means A, such as the imaging information calculation section F74, the acceleration sensors F701 and F761, and the gyro sensor, outputs a signal in accordance with the movement of the unit body. The operation means B, such as the stick, the cross key, the operation button, the touch pad, outputs a signal in accordance with the player pressing a button, tilting a component or touching the same.

[0589] When the core unit F70 includes the operation means A and the subunit F76 includes the operation means B, the player can move one hand holding the core unit F70 while the player makes an input with a finger of the other hand holding the subunit F76 as in the case of a conventional controller. That is, the player can perform different operations with a right and a left hands, respectively, thereby realizing a new operation which cannot be performed by a conventional controller. In this case, according to various embodiments, operation data outputted by the operation means A corresponds to first operation data, and operation data outputted by the operation means B corresponds to second operation data. Further, the controller may be constructed such that the subunit F76 may include the operation means A, the core unit F70 may include the operation means A, and the subunit F76 may include the operation means A and the operation means B. In this manner, the player can move both hands individually, thereby realizing an increasingly improved operation. In this case, according to various embodiments, operation data outputted by the operation means A of the subunit F76 corresponds to third operation data.

[0590] Further, when the core unit F70 and the subunit F76 each includes the operation means A, the player can move one handholding the core unit F70 while the player can move the other hand holding the subunit F76 so as to make an input. That is, the player can move a right and a left hands individually, thereby realizing a new operation which cannot be performed by a conventional controller. In this case, according to various embodiments, operation data outputted by the respective operation means A of the core unit F70 and the subunit F76 correspond to first operation data and second operation data. Further, each of the core unit F70 and the subunit F76 may include both the operation means A and the operation means B. In this manner, the player can perform operations by moving both hands and using fingers of both hands, thereby realizing a new operation. In this case, according to various embodiments, operation data outputted by the operation means B of the core unit F70 corresponds to first key operation data, and operation data outputted by the operation means B of the subunit F76 corresponds to second key operation data.

[0591] Furthermore, when each of the core unit F70 and the subunit F76 includes the operation means A, one of the core unit F70 or the subunit F76 may include various types of operation means A. As described above, when the operation means A includes the imaging information calculation section, a direction, a position and the like of the unit with respect to the imaging target (marker) can be calculated, thereby enabling an operation based on the direction and the position of the unit with respect to the monitor F2. On the other hand, when the operation means A includes the acceleration sensor or the gyro sensor, a tilt, an attitude, a position and the like of the unit itself can be calculated, thereby enabling an operation based on the attitude and the position of the unit. Accordingly, when the core unit F70 includes the imaging information calculation section and one of the acceleration sensor or the gyro sensor, and the subunit F76 includes the acceleration sensor or the gyro sensor, the core unit F70 can perform the aforementioned two operations. In this case, according to various embodiments, operation data outputted by the imaging information calculation section of the core unit F70 corresponds to first operation data, operation data outputted by the acceleration sensor or the gyro sensor of the subunit F76 corresponds to second operation data, and operation data outputted by the acceleration sensor or the gyro sensor of the core unit F70 corresponds to third operation data.

[0592] In the present embodiment, image data taken by the image pickup element F743 is analyzed so as to obtain position coordinates and the like of an image of infrared lights from the LED modules F8L and F8R, and the core unit F70 generates process result data from the obtained coordinates and the like and transmits the process result data to the game apparatus 3. However, the core unit F70 may transmit data obtained in another process step to the game apparatus F3. For example, the core unit F70 transmits to the game apparatus F3 image data taken by the image pickup element F743, and the CPU F30 may perform the aforementioned analysis so as to obtain process result data. In this case, the image processing circuit F744 can be eliminated from the core unit F70. Alternatively, the core unit F70 may transmit, to the game apparatus F3, the image data having been analyzed halfway. For example, the core unit F70 transmits to the game apparatus F3 data indicating a brightness, a position, an area size and the like obtained from the image data, and the CPU F30 may perform the remaining analysis so as to obtain process result data.

[0593] Although in the present embodiment infrared lights from the two LED modules F8L and F8R are used as imaging targets of the imaging information calculation section F74 in the core unit F70, the imaging target is not restricted thereto. For example, infrared light from one LED module or infrared lights from at least three LED modules provided in the vicinity of the monitor F2 may be used as the imaging target of the imaging information calculation section F74. Alternatively, the display screen of the monitor F2 or another emitter (room light or the like) can be used as the imaging target of the imaging information calculation section F74. When the position of the core unit F70 with respect to the display screen is calculated based on the positional relationship between the imaging target and the display screen of the monitor F2, various emitters can be used as the imaging target of the imaging information calculation section F74.

[0594] The aforementioned shapes of the core unit F70 and the subunit F76 are merely examples. Further, the shape, the number, setting position and the like of each of the operation section F72 of the core unit F70 and the operation section F78 of the subunit F76 are merely examples. In various embodiments, the shape, the number, the setting position and the like of each of the core unit F70, the subunit F76, the operation section F72, and the operation section F78 may vary and still fall within the scope of various embodiments. Further, the imaging information calculation section F74 (light entrance of the imaging information calculation section F74) of the core unit F70 may not be positioned on the front surface of the housing F71. The imaging information calculation section F74 may be provided on another surface at which light can be received from the exterior of the housing F71.

[0595] Further, although the speaker F706, the sound IC F707, and the amplifier F708 as described above are included in the core unit F70, any devices at hand capable of outputting a sound may be included in either the subunit F76 or the core unit F70.

[0596] Thus, the controller according to various embodiments allows a player to operate both the core unit F70 and the subunit F76 included therein so as to enjoy a game. For example, the core unit F70 has a function of outputting a signal in accordance with a motion of the unit body including the imaging information calculation section F74 and the accelerator sensor F701, and the subunit F76 has a function of outputting a signal in accordance with a direction input operation performed by the player. For example, when used is a controller into which the core unit F70 and the subunit F76 are integrated, the whole controller has to be moved so as to output a signal in accordance with the motion of the unit body, thereby exerting some influence on the direction input operation. Further, the integration of the core unit F70 and the subunit F76 causes the opposite influence, that is, flexibility, which is realized by separation between the core unit F70 and the subunit F76, is substantially reduced. As another example, the core unit F70 may have a function of outputting a signal in accordance with a motion of the unit body including the imaging information calculation section F74 and the acceleration sensor F701, and the subunit F76 may have a function of outputting a signal in accordance with the motion of the unit body including the acceleration sensor F761. Therefore, the player can move both hands holding the different units individually so as to make an input. Accordingly, the core unit F70 and the subunit F76 can be separated into a right unit and a left unit as in the case of a conventional controller for the game apparatus, and simultaneously the core unit F70 and the subunit F76 allow the player to freely use his or her right and left hands, thereby providing the player with a new operation, which cannot be performed by the integrated controller. Further, the controller can be operated with substantially enhanced flexibility, thereby providing a player with a game operation having increased reality.

[0597] The game controller and the game system according to various embodiments can realize an operation having increased flexibility, and are useful as a game controller which includes two independent units and is operated by a player holding the two independent units, a game system including the game controller, and the like.

Motion Control for Gaming Devices

[0598] In some embodiments, a gaming device, such as a mobile gaming device, receives inputs in the form of motion. For example, a human holding a mobile gaming device may make commands or provide instructions by tilting the device, moving the device in some direction, rotating the device, shaking the device, hitting the device against something, tossing the device, or providing any other motion-based inputs to the device. The motions may translate to one or more commands or instructions used in a game. The motions may also translate to commands or instructions or requests used for other purposes, e.g., beyond the play of a game. Commands, instructions, requests, and specifications may include: (a) an instruction to place a bet; (b) a specification of the size of a bet; (c) an instruction to begin a game; (d) an instruction to pursue a particular strategy in a game; (e) an instruction to hold a particular card in a game of video poker; (f) an instruction to hit in a game of blackjack; (g) an instruction to cash out; (h) an instruction to switch games; (i) a specification of a particular type of game to play; (j) an instruction to make a particular selection in a bonus round; (k) a request to order a drink; (l) a request to order food; (m) an instruction to summon a casino representative; (n) a request to redeem comp points; (o) a request to receive a comp benefit; (p) an instruction to open up a line of communication with another person (e.g., with a friend who is also in a casino); (q) an instruction to make a withdrawal from an account (e.g., from a bank account); (r) an instruction to fund an account (e.g., to fund an account a player has with a casino with gaming credits); (s) a request to make a purchase; (t) a request to purchase show tickets; (u) an instruction to make a reservation at a restaurant; (v) a request for information; (w) a request for information about a pay table (e.g., about the payouts on a pay table); (x) a request for a location of a particular room; (y) a request to check into a hotel room; (z) a request to reserve a hotel room; (aa) a request to check on show times; (ab) a request to claim a jackpot; (ac) a request to make a phone call; (ad) a specification of a phone number; (ad) a request to access a network; (ae) a request to access the Internet; (af) a specification of a Web or URL address; (ag) a request to receive information about another player; (ah) a request to see information about a game outcome of another player; (ai) a request to see information about the gaming history of another player; (aj) a request to receive information about one or more players, dealers, gaming devices, or game tables (e.g., a request to see the most recent outcomes for any of the aforementioned); and any other request, instruction, command, or specification. The mobile gaming device may include hardware and/or software for detecting motions. The mobile gaming device may work in conjunction with external hardware or software for detection motions. The mobile gaming device or another device may include software for translating motions detected into instructions that can be used in conducting a game or in any other fashion.

[0599] Herein, "motion control" may include using motion as an input to a game, using motion as a command, and/or using motion as instructions. Motion control may include using the motion of a mobile gaming device to provide inputs to the games played on the mobile gaming device, to select games to play, to indicate a player's desire to cash out, or to provide various other instructions or indications.
1. TECHNOLOGIES. Various technologies may be used to enable motion control. Such technologies may include technologies for sensing motion, including such information as acceleration, velocity, angular motion, displacement, position, angular displacement, angular velocity, angular acceleration, impact shocks, and any other information that may be associated with motion. Technologies may include sensors, including hardware sensors. Technologies may also include software for translating information received

from sensors into information about the position, trajectory, or other spatial information about a mobile gaming device. For example, software may be used to translate acceleration information into position information, e.g., through double integration. Various technologies may or may not be described in the following references, each of which is hereby incorporated by reference herein: (1) U.S. Patent Application 20040046736 "Novel man machine interfaces and applications"; (2) U.S. Patent Application 20030100372 "Modular entertainment and gaming systems"; (3) U.S. Pat. No. 7,058,204 "Multiple camera control system"; (4) U.S. Pat. No. 5,534,917 "Video image based control system"; (5) U.S. Patent Application 20060281453 "ORIENTATION-SENSITIVE SIGNAL OUTPUT"; (6) U.S. Patent Application 20060098873 "Multiple camera control system"; (7) U.S. Pat. No. 6,850,221 "Trigger operated electronic device"; (8) U.S. Patent Application 20070072680 "Game controller and game system"; (9) U.S. Patent Application 20070066394 "VIDEO GAME SYSTEM WITH WIRELESS MODULAR HANDHELD CONTROLLER"; (10) U.S. Patent Application 20070050597 "Game controller and game system"; (11) U.S. Patent Application 20070049374 "Game system and storage medium having game program stored thereon"; (12) U.S. Patent Application 20060139322 "Man-machine interface using a deformable device"; (13) U.S. Pat. No. 6,676,522 United States Patent "Gaming system including portable game devices"; (14) U.S. Pat. No. 6,846,238 "Wireless game player"; (15) U.S. Pat. No. 6,702,672 "Wireless interactive gaming system"; (16) U.S. Pat. No. 7,148,789 "Handheld device having multiple localized force feedback"; (17) U.S. Pat. No. 7,209,118 "Increasing force transmissibility for tactile feedback interface devices"; (18) U.S. Pat. No. 6,965,868, "System and method for promoting commerce, including sales agent assisted commerce, in a networked economy"; and (19) U.S. Pat. No. 7,058,204, "Multiple camera control system".

1.1. CAMERA IN THE DEVICE. A camera on the mobile gaming device may capture images. As the mobile gaming device moves, different images will likely be captured by the camera. Stationary objects may appear to move between the images captured in successive frames. From the apparent motion of the stationary objects, the motion of the mobile gaming device may be inferred.

1.2. EXTERNAL CAMERAS. External cameras, such as stationary wall-mounted cameras, may film the mobile gaming device and/or the player holding the mobile gaming device. From footage of the mobile gaming device, algorithms may infer the motion of the mobile gaming device.

1.3. EXTERNAL READERS (E.G., RANGE FINDERS). External sensors or readers may detect the motion of the mobile gaming device. For example, ultrasound waves or lasers may be bounced off the mobile gaming device. From changes in the reflected sound or light, the motion of the mobile gaming device may be inferred.

1.4. ACCELEROMETERS. A mobile gaming device may include built-in accelerometers. These may detect changes in velocity, which may be used to infer other aspects of motion, such as change in position or velocity.

1.5. GYROSCOPE SENSORS. A mobile gaming device may contain internal gyroscopes. These may detect an orientation of the mobile gaming device. Information from a gyroscope may be used to infer other information, such as angular displacement.

1.6. POSITION DETECTORS INTERNAL (GPS). A mobile gaming device may include position detectors, such as sensors for a global positioning system or for a local positioning system. Position, when measured over time, may be used to infer other aspects of motion, such as velocity or acceleration.

1.7. POSITION DETECTORS EXTERNAL. External detectors may measure the position of a mobile gaming device. For example, the mobile gaming device may emit a signal in all directions. Based on the time it takes the signal to reach various fixed detectors, the position of the mobile gaming device may be inferred.

1.8. RFID. DETECT AS THE SIGNAL STRENGTH OF AN RFID GETS STRONGER

OR WEAKER. A mobile gaming device may contain a radio frequency identification (RFID) tag or other radio frequency emitting device. Based on the reception of the signal from the RFID tag, information about the position of the mobile gaming device may be inferred. For example, if the signal received is weak, it may be inferred that the mobile gaming device is far from a fixed receiver. If the received signal is strong, it may be inferred that the mobile gaming device is near to the fixed receiver.
2. SWITCH FOR MOTION COMMANDS. ENABLE SWITCH FOR THE MOTION COMMAND. PRESS MOTION BUTTON, AND WHILE PRESSED, MOTION WORKS. COULD BE A CONSTANT COMMAND OR TOGGLE ON OR OFF TO MAKE THE COMMAND BE IN FORCE. TO ENABLE THAT YOU'RE IN MOTION CONTROL MODE, YOU COULD GO THROUGH ALL THESE MOTIONS. In various embodiments, motion control may alternately enabled or disabled. At some times motion control may be in use, while at other times motion control may not be in use. For example, at a first time a the motion of a mobile gaming device may cause decisions to be made in a game, while at a second time the motion of a mobile gaming device may not have any effect on a game. When motion control is enabled, a player may be able to conveniently engage in game play. When motion control is off, a player may move a mobile gaming device inadvertently without worry that such movement will affect a game. Thus, there may be reasons at various times to have motion control enabled, and reasons at various times to have motion control disabled.
2.1. TOGGLE ON AND OFF. In various embodiments, a player must provide continuous, substantially continuous, or persistent input in order to maintain the enablement of motion control. Continuous input may include continuous pressure, such as the continuous pressing and holding of a button. Continuous input may include continuous squeezing of buttons or of the device (e.g., the mobile gaming device) itself. In some embodiments, continuous input may include repeated pressing of a button such that, for example, each button press occurs within a predetermined time interval of the previous button press. In various embodiments, continuous input may include continuous contact. For example, to maintain the enablement of motion control a player must maintain a finger or other appendage in constant contact with a touch sensitive device (e.g., on the mobile gaming device).

[0611] In various embodiments, a continuous input may require the player to continuously supply heat, such as body heat through contact. In various embodiments, continuous input may require the player to continuously supply a finger print, e.g., through keeping a finger in continuous contact with fingerprint reader. In various embodiments, continuous input may include continuous noise or sound generation, e.g., continuous humming by the player.

[0612] So long as a player provides a continuous input, the player may be able to move the mobile gaming device or some other device in order to control action in a game or to otherwise provide commands, instructions or other inputs. For example, to provide an input using motion, a player may press a button on a mobile gaming device and, while the button is pressed, move the mobile gaming device around. Should the player let go of the button, the motion of the mobile gaming device would cease to be used as an input. Should the player then resume pressing the button, the player may once again use the motion of the mobile gaming device as an input.

[0613] In various embodiments, a continuous input may be provided to the mobile gaming device, e.g., when the player holds a button on the mobile gaming device. In various embodiments, a player may provide continuous input to another device. For example, the player may hold down a foot pedal. The foot pedal may be in communication with the mobile gaming device, either directly or indirectly, or the foot pedal may be in communication with another device which would be controlled by the motion of the mobile gaming device. Thus, based on whether the foot pedal is pressed, a determination may be made as to whether the motion of the mobile gaming device will be used to control a game or to provide other input.

[0614] In some embodiments, a continuous input from the player is necessary to disable motion control. In the absence of the continuous input (e.g., if a button is not pressed), the motion of the mobile gaming device will be used to control a game or to provide other direction.

2.2. CONSTANT COMMAND. In various embodiments, a single input, set or inputs, or an otherwise limited set of inputs may switch motion control on or off. For example, a player may press a button to switch motion control on. The player may press the same button again to switch motion control off. As another example, a player may flip a switch one way to switch motion control on, and may flip the switch the other way to switch motion control off. As another example, a player may select from a menu an option to enable motion control. The player may later select another option from the menu to disable motion control.

[0616] Once motion control has been enabled (e.g., with a single press of a button), the motion of the mobile gaming device may be used to control a game or to provide other directions. No further input to enable motion control may be required beyond the initial flipping of a switch or pressing of a button, for example.

2.2.1. MOTION CONTROL GOES OFF WHEN: In some embodiments, motion control may be automatically disabled under certain circumstances. For example, when the player has selected from a menu an option to enable motion control, motion control may remain enabled until some triggering condition occurs which will cause motion control to be automatically disabled.
2.2.1.1. NO MOTION FOR A WHILE. If, for some period of time, there has been no motion, no significant motion, no detectable motion, and/or no motion that is translatable into a coherent instruction, then motion control may be automatically switched off. Motion control may be automatically switched off after 30 seconds, for example.
2.2.1.2. DEVICE LOWERED OR PUT IN POCKET. If a mobile gaming device has been lowered then motion control may be disabled. For example, it may be presumed that a player has put down a mobile gaming device and is no longer playing the mobile gaming device, therefore motion control may be disabled automatically. If a mobile gaming device has been placed in a player's pocket, motion control may be disabled automatically. If, for example, the sensors in the mobile gaming device no longer detect light, and/or detect proximate body heat, motion control may be disabled.
2.2.2. KEYBOARD LOCKING TO AVOID SWITCHING ON MOTION CONTROL ACCIDENTALLY. In various embodiments, a key, switch, or other input device may be manipulated (e.g., pressed) in order to enable motion control. It is possible, in some embodiments, that a player would inadvertently press a button or otherwise manipulate an input device so as to enable motion control. In various embodiments, a key pad of a mobile gaming device may be locked. For example, the player may press a key or sequence of keys that lock the keypad so that that the same input devices which would enable motion control are temporarily non-functional. In various embodiments, only the input devices that could be used to enable motion control are disabled.
2.3. In various embodiments, an alert is provided when motion control is enabled. For example, a mobile gaming device may beep, buzz, or emit a tone when motion control is enabled. A text message may be displayed, lights may flash, or other visual alerts may be output when motion control is enabled. In various embodiments, a voice output may be used to alert a player that motion control is enabled.

[0622] In various embodiments, an alert may indicate that motion control has been disabled. The alert may take the form of text, flashing lights, audio, voice, buzzing, vibrating, or any other form.

3. USE OF VERY PRECISE OR DEFINITIVE MOTION FOR IMPORTANT THINGS (WHERE MONEY IS ON THE LINE), AND LESS PRECISE MOTION FOR LESS IMPORTANT THINGS. THIS WAY, ACCIDENTS LIKE "BET MAX" ARE AVOIDED. ALSO, CERTAIN BETS, LIKE "BET MAX" ARE NOT ALLOWED WHEN MOTION IS ON. In various embodiments, the nature or degree of motion required to provide an instruction may depend on the nature of the instruction itself. Some instructions may require a motion encompassing relatively small displacements, small accelerations, small changes in angle, and/or other small changes. Other instructions may require a motion encompassing relatively large displacements, relatively large accelerations, relatively large changes in angle, or relatively large amounts of other changes. What constitutes a large displacement, acceleration, change in angle, or other change may be defined in various ways, e.g., by some threshold. Thus, for example, a displacement of more than six inches may be considered large or at least may be considered sufficient for one type of instruction. Some instructions may require motions with a large number of repetition or a long sequence of motion (e.g., the device is moved up then down, then side to side, then up again). Some instructions may require motions with little repetition or with a small sequence of motion (e.g., up then down).

3.1. SIZE OF BET. The nature of motion required may depend on the size of a bet placed. For a player to place a large bet (e.g., a bet greater than a certain threshold amount), the player may be required to use motions encompassing large displacements, accelerations, changes in angle, and or other large changes. For a smaller bet, the player may use motions encompassing smaller changes. In various embodiments, the degree of motion may not itself specify the size of a bet. For example, making a motion encompassing a large displacement may not in and of itself specify that a bet should be $25. The specification of a bet may still require a precise sequence of motions, such as one motion for each digit describing the bet, or such as one motion for each credit bet. However, a large bet may require that each of the motions used be more expansive or more emphasized than what would be required with a smaller bet. What constitutes a large bet may vary, and may include any bet that is greater than some threshold, such as $10. Further, there may be multiple thresholds of bets, with each threshold requiring a more emphatic or expansive set of motions.

3.2. SIZE OF POTENTIAL PAYOUT. The nature of motion required may depend on the size of a potential payout. For example, a player may be engaged in a game of video poker and may receive an intermediate outcome comprising five cards. If the intermediate outcome includes four cards to a royal flush, the player may have a large potential payout should he complete the royal flush. Accordingly, when the player selects cards to keep and/or cards to discard, expansive or emphatic motion may be required. If the intermediate outcome does not make a large payout likely, then less expansive or emphatic motions may be required for the player to choose discards. In various embodiments, a mobile gaming device, casino server, or other device may determine whether or not a large payout is possible and/or the probability of a large payout. Based on the size of the payout, the probability of the payout, and/or the possibility of the payout, the nature of the motion required to make a decision in a game may be varied.

3.3. MAKING A SUBOPTIMAL DECISION. In various embodiments, the motion required to make an optimal decision may be less than that required to make a suboptimal decision. For example, to make a blackjack decision that maximizes a player's expected winnings may require a relatively small displacement, while to make another decision may require a large displacement. In various embodiments, a mobile gaming device, casino server, or other device may determine a strategy that maximizes

a player's expected winnings, that maximizes a potential payout for the player, or that maximizes some other criteria for the player. The mobile gaming device may accept relatively less expansive motions which provide instruction to follow the best strategy, while the mobile gaming device may require relatively more expansive motions if such motions correspond to an instruction to follow a strategy other than the best strategy.
4. CALIBRATION SEQUENCE, TUTORIAL. MAY BE REQUIRED SO YOU CAN'T LATER CLAIM THAT YOU DIDN'T MEAN TO MAKE A BET. In various embodiments, a player may go through an exercise to calibrate a mobile gaming device to his way of providing motion. Each player may be unique. For example, each player may have arms of a different length, hands of a different size, different body mechanics, different muscle strengths, and other differences which may effect the way a player moves a mobile gaming device. Thus, a player may go through a process of training the mobile gaming device to recognize the individual player's motions. In various embodiments, the mobile gaming device may guide the player through a sequence of steps in order to calibrate the mobile gaming device. The mobile gaming device may provide the player with instructions, e.g., using the screen display of the mobile gaming device or using voice prompts.
4.1. MAKE A MOTION X TIMES. OK, THIS WILL BE HOW YOU BET. In various embodiments, the mobile gaming device may instruct the player to make a particular motion. Exemplary instructions may include: "move the mobile gaming device up"; "move the mobile gaming device up 6 inches"; "move the mobile gaming device down"; "move the mobile gaming device left"; "move the mobile gaming device right"; "tilt the mobile gaming device left"; "tilt the mobile gaming device right"; "rotate the screen of the mobile gaming device towards you"; "shake the mobile gaming device"; "tap the mobile gaming device against something". The mobile gaming device may instruct the player to make a sequence of motions. Exemplary instructions may include: "move the mobile gaming device up and then to the right"; "move the mobile gaming device up, then down, then up again"; "tilt the mobile gaming device left and move it left". The mobile gaming device may instruct the player to perform a given motion one or more times. For example, the mobile gaming device may instruct the player to perform a given motion five times. When the player performs a motion multiple times, the mobile gaming device may have more data with which to establish an "average" motion or an expected range of motion which will be used to correspond to a given instruction. In various embodiments, a player may be asked to repeat the same motion several times in succession. In various embodiments, a player may be asked to perform a number of different motions such that certain motions are repeated, but not necessarily right after one another. Throughout the process of a player making motions (e.g., while holding the mobile gaming device), the mobile gaming device or another device may record data about the motions. For example, the mobile gaming device may record the amount of displacement, the amount of acceleration, the speed, the time taken to complete a motion, the amount of angular rotation, and/or any other aspects of the motion. In the future, the mobile gaming device or other device may associate similar data with the same motion. For example, if a player was asked to move a mobile gaming device in a particular way and if data was recorded about the way in which the player actually did move the mobile gaming device, then it may be assumed that if similar data is received in the future then the player has again tried to move the mobile gaming device in the same particular way. In various embodiments, certain motions from the player may not be accepted. For example, the mobile gaming device may have software with inbuilt expectations about what an "up" motion should be. If the mobile gaming device has asked the player to move the mobile gaming device "up" and the mobile gaming device detects what it interprets as a downward motion, then the mobile gaming device may take various actions. The mobile gaming device may ask the player to please try again. The mobile gaming device may tell the player that he has not followed instructions and that he should have moved the mobile gaming device up.

4.2. TEST In various embodiments, a player may be asked to perform a motion of his choice. The mobile gaming device may then try to identify the motion. The mobile gaming device may indicate, for example, whether the motion was up, down, to the left, etc. The mobile gaming device may indicate the instruction that the motion was interpreted as. For example, the mobile gaming device may indicate that the motion was a "discard first card" instruction or that the motion was a "spin reels" motion. After a mobile gaming device indicates its interpretation of a motion, the player may confirm whether or not the mobile gaming device was correct. For example, the player may press a "correct" or "incorrect" button on the mobile gaming device. If the mobile gaming device has incorrectly identified one or more player motions, then the player may be asked to go through a process of training, e.g., an additional process of training. In various embodiments, training may continue until the mobile gaming device can successfully identify all player motions and/or all player instructions (e.g., until the mobile gaming device is correct on 50 straight trials).
4.3. TUTORIAL.

In various embodiments, a training session or tutorial may be geared towards a player. The mobile gaming device, another device, or a human (e.g., a casino representative) may show the player which motions to use for various instructions. For example, the mobile gaming device may tell the player to tilt the mobile gaming device twice to the left to discard the first card in a game of video poker. The player may then be asked to try the motion one or more times. At some point, a player may be tested as to his understanding of which motions perform which commands. The player may be asked to do various things, such as to initiate a game, such as to make a "double down" decision in blackjack, such as to cash out, or such as any other thing. In various embodiments, the player may be required to repeat the tutorial and/or may be prevented from gaming using motion control until he passes a test of his knowledge of which motions perform which instructions. Passing may include, for example, providing accurate motions for all 10 things one is asked to do. In some embodiments, a player may be required to take a game-specific tutorial and/or to pass a game-specific test prior to playing a particular game. The game may require specialized motions and it may therefore be prudent for the player to take a tutorial on such motions. Absent taking a game-specific test or tutorial, a player may still be allowed to play other games.
4.4. SIGN OR OTHERWISE VERIFY YOU WENT THROUGH THE TUTORIAL. In various embodiments, a player may be asked to confirm or verify that he completed a tutorial, such as a tutorial which instructs the player on what motions to use for particular instructions. The player may confirm by providing a biometric reading (e.g., by touching his thumb to a touchpad), by signing something (e.g., by signing the screen of his mobile gaming device with a stylus), by recording a voiced statement to the effect that he has completed the tutorial, or by providing any other confirmation.
4.5. MOTION AIDS, CAN BE TURNED ON OR OFF. FOR EXAMPLE, LITTLE ARROWS ON THE SCREEN EXPLAIN HOW TO MOVE THE DEVICE TO MAKE VARIOUS BETS. BUT AS YOU GET USED TO THESE, YOU CAN TURN THE ARROWS OFF. In various embodiments, a player may be provided with various aids or hints during a game, the aids telling the player how to provide certain instructions. For example, text displayed on the screen of a mobile gaming device may tell the player what motion to make to "hit", what motion to make to "stand", and so on. In a game of video poker, a voice may be emitted from the mobile gaming device telling the player how to discard the first card, how to discard the second card, and so on. For example, the voice may say, "tilt the device forward to discard the third card". In another example, arrows may appear showing the player how to move the device to provide a particular instruction. For example, an arrow pointing left that is superimposed on a card may tell the player to tilt the device left in order to discard the card. In various embodiments, the aids or hints may be turned on or off by the player. An inexperienced

player may wish to have the aids on. However, eventually the player may become so familiar with the motion control that the player may wish to turn off the aids. The mobile gaming device may then no longer provide hints or aids as to what motions to make in order to provide a particular instruction. In some embodiments, hints or aids may appear automatically or by default, such as when a player first begins playing a new type of game (e.g., such as when the player first starts playing video poker). In some embodiments, the default setting is not to have aids.
4.6. CUSTOMIZE MOTIONS. I WANT X TO MEAN Y. THESE CAN BE COMPLICATED SETS OF INSTRUCTIONS. In various embodiments, a player may customize the motions that will correspond to various instructions. The mobile gaming device may take the player through a calibration sequence where the mobile gaming device asks the player what motion he would like to make to correspond to a given instruction. The player may be asked to make the motion some number of times, such as a fixed number of times or such as a number of times needed in order for the player to establish a consistency of motion or for the mobile gaming device to extract the essential parameters of the motion. The calibration sequence may proceed through one or more instructions until the player has created a motion corresponding to each. In various embodiments, each instruction may correspond to a default motion. The player may have the opportunity to change the default motion to another motion that better suits his preferences. In various embodiments, a player may wish for a motion to correspond to a sequence of instructions, e.g., a long or complicated sequence of instructions. For example, the player may wish for a single motion to correspond to the following sequence: (1) bet $5; (2) initiate a game of video poker; and (3) automatically choose discards in accordance with optimal strategy. The motion may be a motion where the player shakes the mobile gaming device twice, for example. Thus, in various embodiments, a simple motion may be used to execute a lengthy or complicated set of instructions. This may allow a player to conveniently perform desired sequences of actions.
5. THERE MAY BE CONFIRMATION. A DISPLAY MAY SAY, "YOU HAVE MOTIONED TO BET 10." In various embodiments, following a motion made by a player (e.g., following the player moving the mobile gaming device), a confirmation or an interpretation of the player's motion may be output. The mobile gaming device or another device may make such a confirmation. The mobile gaming device may display a message on its display screen indicating how the player's motion has been interpreted. For example, the mobile gaming device may display a message indicating that the player has instructed that a bet of 10 be placed on a game. The mobile gaming device may also output a message in the form of audio (e.g., using synthetic voice) or in any other format. The player may have the opportunity to view the message and to take action if he believes his motions have been misinterpreted as the wrong instructions. For example, the mobile gaming device may output an audio message using synthetic voice. The audio message may say, "You have chosen to stand. Shake the mobile gaming device if this is not your intention." The player may have some limited period of time in which to take an action to prevent the mobile gaming device from carrying out the misconstrued instruction. If the player takes no action, the instruction that has been construed by the mobile gaming device may be carried out. The player may also have the opportunity to confirm an interpretation of his motion and, for example, to thereby cause his instructions to be executed more quickly. For example, the player may shake a mobile gaming device once to confirm an interpretation of the player's prior motion by the mobile gaming device, and to thereby allow the mobile gaming device to execute the player's instruction.
5.1. THERE MAYBE VERIFICATION. A PERSON MUST MOTION AGAIN TO COMPLETE A BET. In some embodiments, a player must confirm an interpretation of a motion before his instruction will be executed. In some embodiments, a person must repeat a motion one or more times (e.g., the player must provide the same instruction

two or more times) before the instruction will be carried out. In some embodiments, higher levels of verification may be required for instructions with greater consequence, such as instructions to bet large amounts or such as instructions provided when a player has the potential to win a large payout. For example, a player may have 3 seconds to stop a mobile gaming device from executing its interpretation of an instruction to bet $50, but only 1 second to stop a mobile gaming device from executing its interpretation of an instruction to bet $5.

6. MOTION TO VERIFY PLAYER IDENTITY. FOR EXAMPLE, EACH PLAYER MAY MOVE A DEVICE IN A UNIQUE WAY. In various embodiments, the motion of a mobile gaming device or other device may be used as a biometric or as a way to presumably uniquely identify a person. It may be presumed, for example, that each person has a different way in which they would move a mobile gaming device. Software within a mobile gaming device or within another device may capture motion data (e.g., using accelerometers, gyroscopes, cameras, etc.). The software may then determine salient features or statistics about the motion. For example, the software may determine a degree of curvature or loopiness to the motion, a maximum acceleration, a maximum speed, a total displacement, a presence of vibrations, and/or any other characteristics of the motion. When a player attempts to verify his identity by supplying a motion sample (e.g., by moving a mobile gaming device), software may compare his newly supplied motion to a motion previously supplied by the purported player. If the motions match (e.g., if the values of salient features of the motion are the same within some confidence interval), then the player may be presumed to be who he says he is. Having confirmed his identity, a player may be granted certain privileges, such as the right to engage in gaming activities using the mobile gaming device.

6.1. ENTER A PASSWORD WITH MOTION. A PASS-SEQUENCE OF MOTIONS. In various embodiments, a player may enter a password using a set of motions. A password may comprise, for example, a sequence of directional motions, such as "up", "down", "left", and "right". A password may consist of 7 such motions, for example. A player may use such a password to verify his identity, for example. Having provided a correct password, a player may be granted certain privileges, such as the right to engage in gaming activities using the mobile gaming device.

7. STANDARD MOTION USED ACROSS MULTIPLE GAMES. In various embodiments, two or more games may receive similar instructions. For example, two or more games may each receive similar instructions as to how much a player wishes to bet. In various embodiments, a given motion may have the same interpretation (e.g., may convey the same instruction or set of instructions) across multiple games. A player may thereby need to learn to use certain motions only once, yet be able to play many games using those motions.

7.1. In various embodiments, a set of standards may be developed, where such standards indicate what motions are to correspond to what instructions. Games that conform to such standards may be labeled as such. For example, a game that accepts a certain set of motions for standard instructions in the game may be associated with a claim that says, "Conforms to Motion 5.0 Standards" or some similar claim. In various embodiments, there may be multiple different standards. A given game may be capable of accepting motions according to multiple different standards. In various embodiments, a player may choose which standard he wishes for a game to employ. For example, a player may have learned to use motions based on a first standard and so may indicate that a game should use the first standard in interpreting his motions as opposed to using a second standard.

7.2. CASHOUT. An instruction which may be common to two or more games is an instruction to cash out. Such an instruction may correspond to a standard motion, such as shaking the mobile gaming device up and down twice.

7.3. QUIT A GAME. An instruction which may be common to two or more games is an instruction to quit the game. Such an instruction may correspond to a standard motion.

7.4. INITIATE A GAME. An instruction which may be common to two or more games is an instruction to initiate or start play of the game. Following such an instruction in a slot machine game, for example, the reels (or simulated reels) may begin to spin. Following such an instruction in a game of video poker, for example, an initial set of five cards may be dealt. Such an instruction may correspond to a standard motion, such as tapping the mobile gaming device against something.
7.5. MAKING A BET. Instructions which may be common to two or more games may include instructions to specify a bet size. One common instruction may be an instruction to increment a bet by one unit or one credit. Such an instruction would, for example, increase a bet from $3 to $4, or a bet from $0.75 to $1.00. One common instruction may be an instruction to increment a bet by a fixed monetary value, such as by 1 quarter or by one dollar. With instructions available to increment bets, a player may specify a bet size by repeatedly incrementing a bet until it reaches the desired size. In various embodiments, instructions to decrement a bet may also be available and may also be standardized. An exemplary such instruction may include an instruction to reduce a bet size by one credit.
7.5.1. NUMERALS. In various embodiments, a bet size may be specified with numerals. Standard instructions based on motions may be available for specifying numerals. For example, a first motion may correspond to the number "1", a second motion may correspond to the number "2", and so on.
7.6. REPEAT LAST ACTION. An instruction may include an instruction to repeat a prior action, such as the last action performed. For example, if the player has just used a first motion to instruct a mobile gaming device to discard the first card in a hand of video poker, the player may use a second motion to instruct the mobile gaming device to repeat the last instruction (i.e., the instruction indicated by the first motion), and to apply the last instruction to the second card in the hand of video poker. In various embodiments, an instruction may include an instruction to repeat a prior game. The instruction may indicate that the amount bet and the number of pay lines played from the prior game should be repeated with the current game. Instructions to repeat a prior action, to repeat a most recent action, or to repeat a game may be common to one or more games, and thus may have standard motion associated with them.
7.7. REPEAT LAST ACTION FROM THIS SITUATION. An instruction may include an instruction to repeat an action from a similar situation in the past. For example, if a player is playing a game of blackjack, the player may provide an instruction to make the same decision that he had made in a previous game in which he had the same point total and in which the dealer had the same card showing. Such an instruction may be associated with a motion. Such a motion may be standardized across two or more games.
7.8. MOTION GENERATES RANDOM NUMBERS. In some embodiments, motion is used to generate one or more random numbers used in a game. For example, readings from various sensors on the mobile gaming device may be captured when the mobile gaming device is moving. Such readings may be converted into numbers (e.g., using some algorithm). The numbers may, in turn, be used in an algorithm for generating a game outcome. In some embodiments, numbers generated by motion are used as the only input to an algorithm for generating an outcome. In some embodiments, numbers generated from the motion of a mobile gaming device may be paired with other numbers (e.g., with random numbers generated by a separate internal algorithm of the mobile gaming device; e.g., with a number representing a time) in order to generate the outcome of a game.
7.8.1. THE IMAGE CAPTURED IS CONVERTED INTO A RANDOM NUMBER. In some embodiments, an image captured from a camera of a mobile gaming device may be converted into a number. In some embodiments, a sequence of images captured during the motion of a mobile gaming device may be used in combination to generate a random number. For example, numbers representing pixel values may be combined

using some function to arrive at a number, e.g., a random number.
7.8.2. THE POSITIONS ARE USED AS A RANDOM NUMBER. In some embodiments, the various positions (e.g., coordinates in two-dimensional or three-dimensional space) to which a mobile gaming device is moved are used to generate numbers, such as random numbers. In some embodiments, accelerations, velocities, durations of motions, paths taken, angular changes, angular accelerations, and any other aspect of motion may be used to generate numbers.
7.9. MOVE THE MOBILE GAMING DEVICE TO KEEP REELS SPINNING. WHEN YOU STOP MOVING, THE REELS STOP. In some embodiments, a player may move a mobile gaming device to draw out the duration of a game. For example, the reels in slot machine game may continue to spin as the player continues to move the mobile gaming device. The reels may stop spinning once the player has stopped moving the mobile gaming device.
8. NEW ARRANGEMENT OF GAME SYMBOLS TO MAKE MOTION CONTROL EASIER. In various embodiments, game indicia, game controls, or other visuals used in a game may be arranged on a display screen of a mobile gaming device in a way that makes it intuitive for a player to interact with such visuals. For example, a player may have available to him four possible motions: (1) tilt the mobile gaming device forward, or away from himself; (2) tilt the device left; (3) tilt the device right; and (4) tilt the device backwards, or towards himself. To make such motions intuitive to use, visuals in a game may be clearly situated in one of four areas of a display screen, namely the top, bottom, left side, and right side. A player may thus readily interact with a visual on top of the screen using a forward tilting motion, with a visual on the left side of the screen using a left tilting motion, with a visual on the right side of the screen using a right tilting motion, and with a visual at the bottom of the screen using a backwards tilting motion. In various embodiments, indicia or other visuals are displayed in an area of a display screen such that the direction of such area from the center of the display screen corresponds to a direction of motion that a player must use in order to interact with such indicia.
8.1. IN VIDEO POKER, CARDS ARE ARRANGED AROUND THE PERIPHERY OF THE SCREEN. THIS WAY, YOU CAN TILT FORWARD, RIGHT, BACK, LEFT, AND MORE CLEARLY INDICATE WHICH CARD TO HOLD. In some embodiments, the cards dealt in a game of video poker may be displayed respectively at the four corners of a display screen on the mobile gaming device, with a fifth card displayed perhaps in the center of the screen. A player may indicate that he wishes to discard a particular card by tilting the mobile gaming device towards the corner of the display screen in which the particular card is displayed. To discard the card in the center, for example, the player may move the mobile gaming device up and down. Thus, by displaying cards in an arrangement other than a linear arrangement, intuitive motion control is facilitated.
8.1.1. PENTAGONAL DISPLAY. In various embodiments, a display may have the shape of a pentagon. A pentagonal display may allow for each corner of the display to be occupied by a different card in a game of video poker, for example. A player may then be able to tilt or otherwise move the mobile gaming device in the direction of one of the corners in order to hold or discard the card which is shown in that corner. In various embodiments, displays of other shapes may be used. The shape of a display may be chosen which most conveniently or intuitively corresponds to a game. In some embodiments, the hardware used for a display may itself maintain a standard form, such as a rectangular form. However, the display may simulate another display which is of a different shape. For example, a rectangular display may simulate a pentagonal display by only illuminating a pentagonal portion of the display screen.
8.2. BETTING BUTTONS MAY ALSO BE ALLOCATED AROUND THE PERIPHERY OF THE SCREEN. In various embodiments, control buttons or control-related visuals may be situated in areas of a display screen that make interaction with such buttons

using motion intuitive. Control visuals may correspond to instructions that may be used in a game. Control visuals may include rectangular areas of a display screen labeled "spin", "bet max", "bet 1" "cash out". Control visuals may correspond to any other instructions. Control buttons may be clearly located, for example, near the top, bottom, left, or right side of a display screen. The player may then tilt the mobile gaming device in the same direction as is represented by the location of a control visual relative to the center of the display screen in order to convey the instruction indicated by the control visual. For example, if a control visual labeled "spin" is located on the right side of a display screen, the player may tilt the mobile gaming device to the right in order to spin the reels of a slot machine game (e.g., in order to start a new game).
8.3. BINARY SEARCH SETUP FOR PLAYING WITH MOTION. FOR EXAMPLE, THIS ALLOWS YOU TO MAKE FINE-GRAINED DECISIONS WITH LIMITED INPUTS (E.G., WITH ONLY RIGHT, LEFT, FORWARD, BACK). In various embodiments, a player may specify an instruction from a range or continuum of possible instructions using a limited set of possible motions (e.g., using only two motions, such as a motion to the left and a motion to the right). To begin with, any instruction may be possible. With each motion a player makes, the player may eliminate a portion of the instructions from consideration. For example, with each motion, the player may eliminate approximately half the remaining possible instructions from consideration. Eventually, after a sequence of motions, only a single instruction will remain. This instruction may then be executed by the mobile gaming device. In some embodiments, a set of possible instructions may be visually indicated with a list on a display screen. The player may tilt the mobile gaming device forward to select the top half of the remaining instructions on the list, and may tilt the mobile gaming device backwards to select the bottom half of the remaining instructions on the list. The remaining instructions may be highlighted, or the instructions which have been eliminated from consideration may disappear. After a sequence of motions from the player, only a single instruction may remain, and may be executed by the mobile gaming device.
9. IT IS POSSIBLE TO HAVE A DEVICE THAT A PERSON DOESN'T NEED TO LOOK AT. MOTION INPUTS CAN OBVIATE THE NEED TO PRESS BUTTONS. THE DEVICE CAN BUZZ TO TELL YOU A GAME IS OVER, AND PERHAPS HOW MUCH YOU'VE WON. In various embodiments, a mobile gaming device may include a device with no display screen. The device may include speakers or other audio output devices. In various embodiments, a mobile gaming device may include a display device, but the display device may not be in use. In various embodiments, a person may play a game using motion control. The person may be informed of a game outcome via voice output from the mobile gaming device. For example, the mobile gaming device may broadcast a synthetic voice telling the player that "the player has lost" or that "the player has won $100". A player may also be informed of an outcome with other audio effects. For example, the sound of chimes may represent a win while a buzzer may represent a loss. The player may then play another game. In this way, the player may proceed to play game after game, without ever looking at the device. A player may thus play, for example, in a dark room. A player may also play while driving or while his vision is otherwise occupied.
10. YOU CAN PRACTICE THIS DEVICE WHEN YOU ARE IN THE CASINO OR EVEN AT HOME. In various embodiments, a player may use motion control on a mobile gaming device in a practice mode, a learning mode, a free play mode, or in some other mode where the player has no money at risk or where the player has a reduced amount of money at risk (e.g., as compared to normal play). The use of motion control in a practice mode may allow the player to learn how to use motion control or may alleviate reservations the player might have with regard to motion control. In various embodiments, a switch, button, or other means of selection may allow a player to switch from practice mode to real mode and/or the other way around. In some embodiments, a mobile gaming device may automatically go into practice mode when it

is outside of designated or legal gaming areas, such as when it is off the casino floor. A mobile gaming device may detect its own location using positioning technologies, such as GPS, for example.
10.1. USE A VIDEO GAME CONTROLLER LIKE THE WII TO PRACTICE. In various embodiments, a device other than a mobile gaming device may be used in order to simulate the use of a mobile gaming device. For example, a device used in a computer game console may be used to simulate the use of a mobile gaming device. An exemplary such device is a controller for Nintendo's(R) Wii(TM) system which takes as inputs the motions a player makes with the controller. In various embodiments, for example, a Wii console or some other computer console may display graphics representative of casino game graphics or otherwise representative of graphics that might appear on the display screen of a mobile gaming device. The player may move the controller in the same way that he would move the actual mobile gaming device. The graphics displayed may then change as they would on an actual mobile gaming device. Thus, a player may simulate the experience of using a mobile gaming device with a controller for a computer game console. When the player later uses a real mobile gaming device in a casino, for example, the player may benefit from having practiced before.
11. CUSTOMIZE TO YOUR GESTURES. TRAIN THE DEVICE AS TO HOW EXTREME YOU WANT YOUR GESTURES. SOME PEOPLE WANT MODERATE GESTURES. OTHERS WANT TO MAKE EMPHATIC GESTURES. In various embodiments, a person may calibrate the mobile gaming device to recognize or to respond to various degrees or types of gestures. Some people may naturally make large or sweeping motions, while other people may prefer more subdued motions. A person may be asked, e.g., through prompts displayed on a mobile gaming device, to make one or more motions while holding the mobile gaming device. The mobile gaming device may note various characteristics of the motion based on sensor readings (e.g., based on readings from accelerometers stored in the mobile gaming device). For example, the mobile gaming device may note whether the motions made by the person have large or small displacements, rapid or gradual accelerations, long or short durations, and/or whether the motions made by a person have any of two alternate characteristics or have any of three or more alternate characteristics. The mobile gaming device, a casino server, or another device may then store information about the nature of a person's motions. When, in the future, the person provides motions as a means for conveying instructions, the motions may be registered or followed only if such motions matched those provided during the calibration phase. For example, if a person used large and expansive motions during calibration, the person may not be able to provide instruction using small subdued motions.
12. EXAMPLES OF MOTIONS. Following are some exemplary instructions that may be provided in a game and/or that may be provided to a mobile gaming device. Associated with the exemplary instructions are exemplary motions of a device, such as of a mobile gaming device, that may be used by a player to indicate a desire that the instructions be carried out.
12.1. HOW TO BET. To provide an instruction to bet one credit, a player may shake the mobile gaming device once. To add an extra credit, the player may shake the mobile gaming device again. To add another extra credit, the player may shake the mobile gaming device again, and so on.
12.2. HOW TO STAND. To provide an instruction to stand in a game of blackjack, a player may tilt a mobile gaming device to the left. To provide an instruction to hit, the player may tilt the gaming device to the right. To provide an instruction to split, the player may move the gaming device down then up.
12.3. HOW TO SELECT A GAME. To select a game, a player may tilt the mobile gaming device to the right. Each time the player tilts the mobile gaming device to the right, a different game from a list of games may be highlighted. When the player's

desired game is highlighted, the player may tap the mobile gaming device against something.

12.4. HOW TO START A GAME. To start a game, a player may move the mobile gaming device in a clockwise circular motion in a plane parallel to the ground.

12.5. HOW TO MAKE A SELECTION IN A BONUS ROUND. To make a selection in a bonus round, a player may continue tilting the mobile gaming device to the right, with each tilt highlighting a different selection (e.g., a different door with a hidden present behind it). When the player's desired selection is highlighted, the player may tap the mobile gaming device against something to make the selection.

12.6. HOW TO CASH OUT. To cash out, a player may move the mobile gaming device up and down three times. Cashing out may include transferring a balance of credit stored locally on a mobile gaming device to a balance stored centrally, such as with a casino server. Cashing out may include causing a mobile gaming device or a nearby device (e.g., a device with which the mobile gaming device is in communication) to print out a ticket which is redeemable for cash.

13. USE MOTION OF THE MOBILE DEVICE TO CONTROL A STATIONARY GAMING DEVICE OR OTHER DEVICE. In various embodiments, the motion of a mobile gaming device may be used to control action at a stationary gaming device or at any other device. In various embodiments, the motion of a mobile gaming device may be used to provide instructions to a stationary gaming device or to any other device. The mobile gaming device may be in communication with the stationary gaming device, either directly (e.g., through direct wireless contact), or indirectly (e.g., with signals relayed through one or more intermediary devices, such as the casino server). In various embodiments, motions of the mobile gaming device may provide instructions to a stationary gaming device, where such instructions may include instructions to bet, to initiate a game, to cash out, to choose a particular choice from among several choices in a bonus round, to bet a particular amount, to discard a particular card, to make a particular decision in blackjack, to claim a jackpot, to call over a casino representative, or to take any other action. In various embodiments, the motions of a mobile gaming device may be translated in a direct or linear fashion to the motions of a cursor or pointer on the screen of a stationary gaming device. For example, when the mobile gaming device is moved to the right, the cursor may move to the right of the screen, and when the mobile gaming device is moved to the left, the cursor may move to the left of the screen. A player may activate or manipulate a control on the stationary gaming device by moving the mobile gaming device in such a way as to position the cursor on the stationary gaming device over the desired control. The player may then provide a final motion, such as shaking the mobile gaming device, to cause the control to be activated. Thus, for example, a player may move a mobile gaming device to the right in order to move a cursor on the screen of a stationary gaming device to the right to be positioned over a "bet" button (e.g., a rendition of a "bet" button). The player may then shake the mobile gaming device to actually place a bet of 1 credit. A player may use the mobile gaming device to control other devices as well, such as ATM machines or vending machines. For example, a player may use the motion of a mobile gaming device to select a product in a vending machine and to then purchase the product. For example, the products in a vending machine may have associated indicator lights. When the player moves the mobile gaming device the indicator light associated with one product may go off and the indicator light associated with another product may go on. The second product may lie in a direction from the first product which is the same direction that was indicated by the motion of the mobile gaming device. In some embodiments, a person may use the motions of a mobile device, such as a mobile gaming device, to control a point of sale terminal.

14. USE OF MOTION AND OTHER TYPES OF INPUT. In various embodiments, a player need not exclusively use motion control to play a game or to perform other actions with a mobile gaming device. For example, a player may specify a bet size by

pressing a key pad, but may actually start a game using a motion, such as shaking the mobile gaming device. In some embodiments, a player may have a choice of ways in which to convey a given instruction. The same instruction may be conveyed through motion or through other means, such as through button presses. Thus, according to a player's fancy, the player may choose one way or the other for providing the same instruction.

The Following are Embodiments, not Claims:
A. A method comprising:

[0670] detecting a first signal from a motion sensor, in which the first signal endures throughout a first period of time;

[0671] determining whether a second signal has endured throughout the first period of time;

[0672] determining, if the second signal has endured throughout the first period of time, an instruction based on the first signal; and

[0673] executing the instruction in a gambling game if the second signal has endured throughout the first period of time.
B. The method of embodiment A in which detecting a first signal includes detecting a first signal from a motion sensor contained within a mobile gaming device, in which the first signal endures throughout a first period of time.
C. The method of embodiment B in which the motion sensor comprises an accelerometer.
D. The method of embodiment B in which the motion sensor comprises a camera.
E. The method of embodiment B further including detecting a second signal from a button on the mobile gaming device, in which the second signal is generated through the application of pressure to the button.
F. The method of embodiment E in which determining whether a second signal has endured throughout the first period of time includes determining whether continuous pressure has been applied to the button throughout the first period of time.
G. The method of embodiment E in which the instruction is one of: (a) an instruction to place a bet; (b) an instruction to place a bet of a certain amount; (c) an instruction to begin the gambling game; (d) an instruction to discard a card; (e) an instruction to receive another card; (f) an instruction to receive no further cards; (g) an instruction to select an option in a bonus round; (h) an instruction to cash out; (i) an instruction to select a pay line; and (j) an instruction to begin a bonus round.
H. The method of embodiment E in which the first signal is generated through motion of the mobile gaming device.
I. A method comprising:

[0682] detecting a first signal from a motion sensor of a mobile gaming device;

[0683] interpreting the first signal as a specification of a first bet in a first game to be played at the mobile gaming device, the first bet denominated in valueless currency;

[0684] detecting a second signal from the motion sensor;

[0685] interpreting the second signal as a specification of a second bet in a second game to be played at the mobile gaming device, the second bet denominated in valuable currency; and

[0686] determining an outcome of the second game only if the first game has been completed.
J. The method of embodiment I in which the valueless currency is not exchangeable for United States dollars, and in which the valuable currency is exchangeable for United States dollars.
K. The method of embodiment I in which the second signal has similar characteristics to the first signal.
L. The method of embodiment I further including displaying, prior to detecting the first signal, a message on a display screen of the mobile gaming device, the message providing instructions to move the mobile gaming device in a particular way in order specify the first bet.
M. The method of embodiment I further including:

[0691] asking the player to provide a first proof of his identity following the completion of the first game;

[0692] asking the player to provide a second proof of his identity prior to determining the outcome; and

[0693] verifying that the second proof matches the first proof.
N. The method of embodiment M in which the first proof is a first fingerprint supplied to the mobile gaming device, and in which the second proof is a second fingerprint supplied to the mobile gaming device.
O. A method comprising:

[0696] receiving a signal indicative of a bet at a mobile gaming device with a rectangular display screen;

[0697] determining five cards;

[0698] displaying a first of the five cards in a first corner of the display screen;

[0699] displaying a second of the five cards in a second corner of the display screen;

[0700] displaying a third of the five cards in a third corner of the display screen;

[0701] displaying a fourth of the five cards in a fourth corner of the display screen;

[0702] determining a particular card of the five cards to be discarded;

[0703] determining a sixth card;

[0704] replacing the particular card with the sixth card;

[0705] determining a payout based on the sixth card and based on cards of the five cards that were not discarded; and

[0706] adjusting a credit balance based on the payout.
P. The method of embodiment O further including displaying a fifth of the five cards in the center of the display screen.
Q. The method of embodiment O in which determining a particular card of the five cards to be discarded includes:

[0709] detecting a motion of the mobile gaming device;

[0710] determining that the first of the five cards is to be discarded if the motion is a tilting of the mobile gaming device towards the first corner of the display screen;

[0711] determining that the second of the five cards is to be discarded if the motion is a tilting of the mobile gaming device towards the second corner of the display screen;

[0712] determining that the third of the five cards is to be discarded if the motion is a tilting of the mobile gaming device towards the third corner of the display screen; and

[0713] determining that the fourth of the five cards is to be discarded if the motion is a tilting of the mobile gaming device towards the fourth corner of the display screen.
R. The method of embodiment O in which determining a payout includes determining a payout based on the sixth card, based on cards of the five cards that were not discarded, and based on the rules of video poker.

[0715] The following sections I-X provide a guide to interpreting the present application.

I. Determining

[0716] The term "determining" and grammatical variants thereof (e.g., to determine a price, determining a value, determine an object which meets a certain criterion) is used in an extremely broad sense. The term "determining" encompasses a wide variety of actions and therefore "determining" can include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, "determining" can include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, "determining" can include resolving, selecting, choosing, establishing, and the like.

[0717] The term "determining" does not imply certainty or absolute precision, and therefore "determining" can include estimating, extrapolating, predicting, guessing and the like.

[0718] The term "determining" does not imply that mathematical processing must be performed, and does not imply that numerical methods must be used, and does not imply that an algorithm or process is used.

[0719] The term "determining" does not imply that any particular device must be used. For example, a computer need not necessarily perform the determining.

II. Forms of Sentences

[0720] Where a limitation of a first claim would cover one of a feature as well as more than one of a feature (e.g., a limitation such as "at least one widget" covers one widget as well as more than one widget), and where in a second claim that depends on the first claim, the second claim uses a definite article "the" to refer to the limitation (e.g., "the widget"), this does not imply that the first claim covers only one of the feature, and this does not imply that the second claim covers only one of the feature (e.g., "the widget" can cover both one widget and more than one widget).

[0721] When an ordinal number (such as "first", "second", "third" and so on) is used as an adjective before a term, that ordinal number is used (unless expressly specified otherwise) merely to indicate a particular feature, such as to distinguish that particular feature from another feature that is described by the same term or by a similar term. For example, a "first widget" may be so named merely to distinguish it from, e.g., a "second widget". Thus, the mere usage of the ordinal numbers "first" and "second" before the term "widget" does not indicate any other relationship between the two widgets, and likewise does not indicate any other characteristics of either or both widgets. For example, the mere usage of the ordinal numbers "first" and "second" before the term "widget" (1) does not indicate that either widget comes before or after any other in order or location; (2) does not indicate that either widget occurs or acts before or after any other in time; and (3) does not indicate that either widget ranks above or below any other, as in importance or quality. In addition, the mere usage of ordinal numbers does not define a numerical limit to the features identified with the ordinal numbers. For example, the mere usage of the ordinal numbers "first" and "second" before the term "widget" does not indicate that there must be no more than two widgets.

[0722] When a single device, article or other product is described herein, more than one device/article (whether or not they cooperate) may alternatively be used in place of the single device/article that is described. Accordingly, the functionality that is described as being possessed by a device may alternatively be possessed by more than one device/article (whether or not they cooperate).

[0723] Similarly, where more than one device, article or other product is described herein (whether or not they cooperate), a single device/article may alternatively be used in place of the more than one device or article that is described. For example, a plurality of computer-based devices may be substituted with a single computer-based device. Accordingly, the various functionality that is described as being possessed by more than one device or article may alternatively be possessed by a single device/article.

[0724] The functionality and/or the features of a single device that is described may be alternatively embodied by one or more other devices which are described but are not explicitly described as having such functionality/features. Thus, other embodiments need not include the described device itself, but rather can include the one or more other devices which would, in those other embodiments, have such functionality/features.

III. Terms

[0725] The term "product" means any machine, manufacture and/or composition of matter, unless expressly specified otherwise.

[0726] The term "process" means any process, algorithm, method or the like, unless expressly specified otherwise.

[0727] Each process (whether called a method, algorithm or otherwise) inherently includes one or more steps, and therefore all references to a "step" or "steps" of a process have an inherent antecedent basis in the mere recitation of the term 'process' or a like term. Accordingly, any reference in a claim to a 'step' or 'steps' of a process has sufficient antecedent basis.

[0728] The term "invention" and the like mean "the one or more inventions disclosed in this application", unless expressly specified otherwise.

[0729] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", "certain embodiments", "one embodiment", "another embodiment" and the like mean "one or more (but not all) embodiments of the disclosed invention(s)", unless expressly specified otherwise.

[0730] The term "variation" of an invention means an embodiment of the invention, unless expressly specified otherwise.

[0731] A reference to "another embodiment" in describing an embodiment does not imply that the referenced embodiment is mutually exclusive with another embodiment (e.g., an embodiment described before the referenced embodiment), unless expressly specified otherwise.

[0732] The terms "including", "comprising" and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0733] The terms "a", "an" and "the" mean "one or more", unless expressly specified otherwise.

[0734] The term "plurality" means "two or more", unless expressly specified otherwise.

[0735] The term "herein" means "in the present application, including anything which may be incorporated by reference", unless expressly specified otherwise.

[0736] The phrase "at least one of", when such phrase modifies a plurality of things (such as an enumerated list of things) means any combination of one or more of those things, unless expressly specified otherwise. For example, the phrase "at least one of a widget, a car and a wheel" means either (i) a widget, (ii) a car, (iii) a wheel, (iv) a widget and a car, (v) a widget and a wheel, (vi) a car and a wheel, or (vii) a widget, a car and a wheel. The phrase "at least one of", when such phrase modifies a plurality of things does not mean "one of each of" the plurality of things.

[0737] Numerical terms such as "one", "two", etc. when used as cardinal numbers to indicate quantity of something (e.g., one widget, two widgets), mean the quantity indicated by that numerical term, but do not mean at least the quantity indicated by that numerical term. For example, the phrase "one widget" does not mean "at least one widget", and therefore the phrase "one widget" does not cover, e.g., two widgets.

[0738] The phrase "based on" does not mean "based only on", unless expressly specified otherwise. In other words, the phrase "based on" describes both "based only on" and "based at least on". The phrase "based at least on" is equivalent to the phrase "based at least in part on".

[0739] The term "represent" and like terms are not exclusive, unless expressly specified otherwise. For example, the term "represents" does not mean "represents only", unless expressly specified otherwise. In other words, the phrase "the data represents a credit card number" describes both "the data represents only a credit card number" and "the data represents a credit card number and the data also represents something else".

[0740] The term "whereby" is used herein only to precede a clause or other set of words that express only the intended result, objective or consequence of something that is previously and explicitly recited. Thus, when the term "whereby" is used in a claim, the clause or other words that the term "whereby" modifies do not establish specific further limitations of the claim or otherwise restricts the meaning or scope of the claim.

[0741] The term "e.g." and like terms mean "for example", and thus does not limit the term or phrase it explains. For example, in the sentence "the computer sends data (e.g., instructions, a data structure) over the Internet", the term "e.g." explains that "instructions" are an example of "data" that the computer may send over the Internet, and also explains that "a data structure" is an example of "data" that the computer may send over the Internet. However, both "instructions" and "a data structure" are merely examples of "data", and other things besides "instructions" and "a data structure" can be "data".

[0742] The term "respective" and like terms mean "taken individually". Thus if two or more things have "respective" characteristics, then each such thing has its own characteristic, and these characteristics can be different from each other but need not be. For example, the phrase "each of two machines has a respective function" means that the first such machine has a function and the second such machine has a function as well. The function of the first machine may or may not be the same as the function of the second machine.

[0743] The term "i.e." and like terms mean "that is", and thus limits the term or phrase it explains. For example, in the sentence "the computer sends data (i.e., instructions) over the Internet", the term "i.e." explains that "instructions" are the "data" that the computer sends over the Internet.

[0744] Any given numerical range shall include whole and fractions of numbers within the range. For example, the range "1 to 10" shall be interpreted to specifically include whole numbers between 1 and 10 (e.g., 1, 2, 3, 4, . . . 9) and non-whole numbers (e.g., 1.1, 1.2, . . . 1.9).

[0745] Where two or more terms or phrases are synonymous (e.g., because of an explicit statement that the terms or phrases are synonymous), instances of one such term/phrase does not mean instances of another such term/phrase must have a different meaning. For example, where a statement renders the meaning of "including" to be synonymous with "including but not limited to", the mere usage of the phrase "including but not limited to" does not mean that the term "including" means something other than "including but not limited to".

IV. Disclosed Examples and Terminology are not Limiting

[0746] Neither the Title (set forth at the beginning of the first page of the present application) nor the Abstract (set forth at the end of the present application) is to be taken as limiting in any way as the scope of the disclosed invention(s). An Abstract has been included in this application merely because an Abstract of not more than 150 words is required under 37 C.F.R. $1.72(b).

[0747] The title of the present application and headings of sections provided in the present application are for convenience only, and are not to be taken as limiting the disclosure in any way.

[0748] Numerous embodiments are described in the present application, and are presented for illustrative purposes only. The described embodiments are not, and are not intended to be, limiting in any sense. The presently disclosed invention(s) are widely applicable to numerous embodiments, as is readily apparent from the disclosure. One of ordinary skill in the art will recognize that the disclosed invention(s) may be practiced with various modifications and alterations, such as structural, logical, software, and electrical modifications. Although particular features of the disclosed invention(s) may be described with reference to one or more particular embodiments and/or drawings, it should be understood that such features are not limited to usage in the one or more particular embodiments or drawings with reference to which they are described, unless expressly specified otherwise.

[0749] No embodiment of method steps or product elements described in the present application constitutes the invention claimed herein, or is essential to the invention claimed herein, or is coextensive with the invention claimed herein, except where it is either expressly stated to be so in this specification or expressly recited in a claim.

[0750] The preambles of the claims that follow recite purposes, benefits and possible uses of the claimed invention only and do not limit the claimed invention.

[0751] The present disclosure is not a literal description of all embodiments of the invention(s). Also, the present disclosure is not a listing of features of the invention(s) which must be present in all embodiments.

[0752] Devices that are described as in communication with each other need not be in continuous communication with each other, unless expressly specified otherwise. On the contrary, such devices need only transmit to each other as necessary or desirable, and may actually refrain from exchanging data most of the time. For example, a machine in communication with another machine via the Internet may not transmit data to the other machine for long period of time (e.g. weeks at a time). In addition, devices that are in communication with each other may communicate directly or indirectly through one or more intermediaries.

[0753] A description of an embodiment with several components or features does not imply that all or even any of such components/features are required. On the contrary, a variety of optional components are described to illustrate the wide variety of possible embodiments of the present invention(s). Unless otherwise specified explicitly, no component/feature is essential or required.

[0754] Although process steps, algorithms or the like may be described or claimed in a particular sequential order, such processes may be configured to work in different orders. In other words, any sequence or order of steps that may be explicitly described or claimed does not necessarily indicate a requirement that the steps be performed in that order. The steps of processes described herein may be performed in any order possible. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step). Moreover, the illustration of a process by its depiction in a drawing does not imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to the invention(s), and does not imply that the illustrated process is preferred.

[0755] Although a process may be described as including a plurality of steps, that does not imply that all or any of the steps are preferred, essential or required. Various other embodiments within the scope of the described invention(s) include other processes that omit some or all of the described steps. Unless otherwise specified explicitly, no step is essential or required.

[0756] Although a process may be described singly or without reference to other products or methods, in an embodiment the process may interact with other products or methods. For example, such interaction may include linking one business model to another business model. Such interaction may be provided to enhance the flexibility or desirability of the process.

[0757] Although a product may be described as including a plurality of components, aspects, qualities, characteristics and/or features, that does not indicate that any or all of the plurality are preferred, essential or required. Various other embodiments within the scope of the described invention(s) include other products that omit some or all of the described plurality.

[0758] An enumerated list of items (which may or may not be numbered) does not imply that any or all of the items are mutually exclusive, unless expressly specified otherwise. Likewise, an enumerated list of items (which may or may not be numbered) does not imply that any or all of the items are comprehensive of any category, unless expressly specified otherwise. For example, the enumerated list "a computer, a laptop, a PDA" does not imply that any or all of the three items of that list are mutually exclusive and does not imply that any or all of the three items of that list are comprehensive of any category.

[0759] An enumerated list of items (which may or may not be numbered) does not imply that any or all of the items are equivalent to each other or readily substituted for each other.

[0760] All embodiments are illustrative, and do not imply that the invention or any embodiments were made or performed, as the case may be.

V. Computing

[0761] It will be readily apparent to one of ordinary skill in the art that the various processes described herein may be implemented by, e.g., appropriately programmed general purpose computers, special purpose computers and computing devices. Typically a processor (e.g., one or more microprocessors, one or more microcontrollers, one or more digital signal processors) will receive instructions (e.g., from a memory or like device), and execute those instructions, thereby performing one or more processes defined by those instructions. Instructions may be embodied in, e.g., one or more computer programs, one or more scripts.

[0762] A "processor" means one or more microprocessors, central processing units (CPUs), computing devices, microcontrollers, digital signal processors, or like devices or any combination thereof, regardless of the architecture (e.g., chip-level multiprocessing/multi-core, RISC, CISC, Microprocessor without Interlocked Pipeline Stages, pipelining configuration, simultaneous multithreading).

[0763] Thus a description of a process is likewise a description of an apparatus for performing the process. The apparatus that performs the process can include, e.g., a processor and those input devices and output devices that are appropriate to perform the process.

[0764] Further, programs that implement such methods (as well as other types of data) may be stored and transmitted using a variety of media (e.g., computer readable media) in a number of manners. In some embodiments, hard-wired circuitry or custom hardware may be used in place of, or in combination with, some or all of the software instructions that can implement the processes of various embodiments. Thus, various combinations of hardware and software may be used instead of software only.

[0765] The term "computer-readable medium" refers to any medium, a plurality of the same, or a combination of different media, that participate in providing data (e.g., instructions, data structures) which may be read by a computer, a processor or a like device. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media include, for example, optical or magnetic disks and other persistent memory. Volatile media include dynamic random access memory (DRAM), which typically constitutes the main memory. Transmission media include coaxial cables, copper wire and fiber optics, including the wires that comprise a system bus coupled to the processor. Transmission media may include or convey acoustic waves, light waves and electromagnetic emissions, such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, an EPROM, a FLASH-EEPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0766] Various forms of computer readable media may be involved in carrying data (e.g. sequences of instructions) to a processor. For example, data may be (i) delivered from RAM to a processor; (ii) carried over a wireless transmission medium; (iii) formatted and/or transmitted according to numerous formats, standards or protocols, such as Ethernet (or IEEE 802.3), SAP, ATP, Bluetooth- and TCP/IP, TDMA, CDMA, and 3G; and/or (iv) encrypted to ensure privacy or prevent fraud in any of a variety of ways well known in the art.

[0767] Thus a description of a process is likewise a description of a computer-readable medium storing a program for performing the process. The computer-readable medium can store (in any appropriate format) those program elements which are appropriate to perform the method.

[0768] Just as the description of various steps in a process does not indicate that all the described steps are required, embodiments of an apparatus include a computer/computing device operable to perform some (but not necessarily all) of the described process.

[0769] Likewise, just as the description of various steps in a process does not indicate that all the described steps are required, embodiments of a computer-readable medium storing a program or data structure include a computer-readable medium storing a program that, when executed, can cause a processor to perform some (but not necessarily all) of the described process.

[0770] Where databases are described, it will be understood by one of ordinary skill in the art that (i) alternative database structures to those described may be readily employed, and (ii) other memory structures besides databases may be readily employed. Any illustrations or descriptions of any sample databases presented herein are illustrative arrangements for stored representations of information. Any number of other arrangements may be employed besides those suggested by, e.g., tables illustrated in drawings or elsewhere. Similarly, any illustrated entries of the databases represent exemplary information only; one of ordinary skill in the art will understand that the number and content of the entries can be different from those described herein.

Further, despite any depiction of the databases as tables, other formats (including relational databases, object-based models and/or distributed databases) could be used to store and manipulate the data types described herein. Likewise, object methods or behaviors of a database can be used to implement various processes, such as the described herein. In addition, the databases may, in a known manner, be stored locally or remotely from a device which accesses data in such a database.

[0771] Various embodiments can be configured to work in a network environment including a computer that is in communication (e.g., via a communications network) with one or more devices. The computer may communicate with the devices directly or indirectly, via any wired or wireless medium (e.g. the Internet, LAN, WAN or Ethernet, Token Ring, a telephone line, a cable line, a radio channel, an optical communications line, commercial on-line service providers, bulletin board systems, a satellite communications link, a combination of any of the above). Each of the devices may themselves comprise computers or other computing devices, such as those based on the Intel(R) Pentium(R) or Centrino(TM) processor, that are adapted to communicate with the computer. Any number and type of devices may be in communication with the computer.

[0772] In an embodiment, a server computer or centralized authority may not be necessary or desirable. For example, the present invention may, in an embodiment, be practiced on one or more devices without a central authority. In such an embodiment, any functions described herein as performed by the server computer or data described as stored on the server computer may instead be performed by or stored on one or more such devices.

[0773] Where a process is described, in an embodiment the process may operate without any user intervention. In another embodiment, the process includes some human intervention (e.g., a step is performed by or with the assistance of a human).

VI. Continuing Applications

[0774] The present disclosure provides, to one of ordinary skill in the art, an enabling description of several embodiments and/or inventions. Some of these embodiments and/or inventions may not be claimed in the present application, but may nevertheless be claimed in one or more continuing applications that claim the benefit of priority of the present application.

[0775] Applicants intend to file additional applications to pursue patents for subject matter that has been disclosed and enabled but not claimed in the present application.

VII. 35 U.S.C. $112, paragraph 6

[0776] In a claim, a limitation of the claim which includes the phrase "means for" or the phrase "step for" means that 35 U.S.C. $112, paragraph 6, applies to that limitation.

[0777] In a claim, a limitation of the claim which does not include the phrase "means for" or the phrase "step for" means that 35 U.S.C. $112, paragraph 6 does not apply to that limitation, regardless of whether that limitation recites a function without recitation of structure, material or acts for performing that function. For example, in a claim, the mere use of the phrase "step of" or the phrase "steps of" in referring to one or more steps of the claim or of another claim does not mean that 35 U.S.C. $112, paragraph 6, applies to that step(s).

[0778] With respect to a means or a step for performing a specified function in accordance with 35 U.S.C. $112, paragraph 6, the corresponding structure, material or acts described in the specification, and equivalents thereof, may perform additional functions as well as the specified function.

[0779] Computers, processors, computing devices and like products are structures that can perform a wide variety of functions. Such products can be operable to perform a specified function by executing one or more programs, such as a program stored in a memory device of that product or in a memory device which that product accesses. Unless expressly specified otherwise, such a program need not be based on any particular algorithm, such as any particular algorithm that might be disclosed in the present application. It is well known to one of ordinary skill in the art that a specified function may be implemented via different algorithms, and any of a number of different algorithms would be a mere design choice for carrying out the specified function.

[0780] Therefore, with respect to a means or a step for performing a specified function in accordance with 35 U.S.C. $112, paragraph 6, structure corresponding to a specified function includes any product programmed to perform the specified function. Such structure includes programmed products which perform the function, regardless of whether such product is programmed with (i) a disclosed algorithm for performing the function, (ii) an algorithm that is similar to a disclosed algorithm, or (iii) a different algorithm for performing the function.

[0781] Where there is recited a means for performing a function that is a method, one structure for performing this method includes a computing device (e.g., a general purpose computer) that is programmed and/or configured with appropriate hardware to perform that function. Also included is a computing device (e.g., a general purpose computer) that is programmed and/or configured with appropriate hardware to perform that function via other algorithms as would be understood by one of ordinary skill in the art.

VIII. Disclaimer

[0782] Numerous references to a particular embodiment do not indicate a disclaimer or disavowal of additional, different embodiments, and similarly references to the description of embodiments which all include a particular feature do not indicate a disclaimer or disavowal of embodiments which do not include that particular feature. A clear disclaimer or disavowal in the present application shall be prefaced by the phrase "does not include" or by the phrase "cannot perform".

IX. Incorporation By Reference

[0783] Any patent, patent application or other document referred to herein is incorporated by reference into this patent application as part of the present disclosure, but only for purposes of written description and enablement in accordance with 35 U.S.C. $112, paragraph 1, and should in no way be used to limit, define, or otherwise construe any term of the present application, unless without such incorporation by reference, no ordinary meaning would have been ascertainable by a person of ordinary skill in the art. Such person of ordinary skill in the art need not have been, in any way limited, by any embodiments provided in the reference.

[0784] Any incorporation by reference does not, in and of itself, imply any endorsement of, ratification of or acquiescence in any statements, opinions, arguments or characterizations contained in any incorporated patent, patent application or other document, unless explicitly specified otherwise in this patent application.

X. Prosecution History

[0785] In interpreting the present application (which includes the claims), one of ordinary skill in the art shall refer to the prosecution history of the present application, but not to the prosecution history of any other patent or patent application, regardless of whether there are other patent applications that are considered related to the present application, and regardless of whether there are other patent applications that share a claim of priority with the present application.

XI. Some Embodiments

[0786] In various embodiments, a distributed gaming system enables participants to engage in gaming activities from remote and/or mobile locations. The possible gaming activities include gambling, such as that provided by casinos. Gambling activities may include any casino-type gambling activities including, but not limited to, slot machines, video poker, table games (e.g., craps, roulette, blackjack, pai gow poker, Caribbean stud poker, baccarat, etc.), the wheel of fortune game, keno, sports betting, horse racing, dog racing, jai alai, and other gambling activities. The gaming activities can also include wagering on any type of event. Events can include, for example, sporting events, such as horse or auto racing, and athletic competitions such as football, basketball, baseball, golf, etc. Events can also include such things that do not normally involve wagering. Such events may include, without limitation, political elections, entertainment industry awards, and box office performance of movies. Gaming can also include non-wagering games and events. Gaming can also include lotteries or lottery-type activities such as state and interstate lotteries. These can include all forms of number-selection lotteries, "scratch-off" lotteries, and other lottery contests. The gaming system may be implemented over a communications network such as a cellular network or a private wireless and/or wireline network. Examples of the latter include WiFi and WiMax networks. In some embodiments, the gaming system communications network is entirely independent of the Internet. In some embodiments, the gaming system operation makes minimal use of the Internet, such that only information for which there are no security issues is transmitted via the Internet and/or such that information may be encrypted. In various embodiments, the communications network enables players to participate in gaming from remote locations (e.g., outside of the gaming area of a casino). Also, the system may enable players to be mobile during participation in the gaming activities. In various embodiments, the system has a location verification or determination feature, which is operable to permit or disallow gaming from the remote location depending upon whether or not the location meets one or more criteria. The criterion may be, for example, whether the location is within a pre-defined area in which gaming is permitted by law.

[0787] As shown in FIG. 1, for example, gaming system 10 may include at least one user 12. The system may include additional users such that there is at least a first user 12 and a second user 14. Multiple users may access a first gaming system 10, while other multiple users access a second gaming system (not shown) in communication with first gaming system 10. Users 12 and 14 may access system 10 by way of a gaming communication device 13. Gaming communication device 13 may comprise any suitable device for transmitting and receiving electronic communications. Examples of such devices include, without limitation, mobile phones, personal data assistants (PDAs), computers, mini-computers, etc. Gaming communication devices 13 transmit and receive gaming information to and from communications network 16. Gaming information is also transmitted between network 16 and a computer 18, such as a server, which may reside within the domain of a gaming service provider 20. The location of computer 18 may be flexible, however, and computer 18 may reside adjacent to or remote from the domain of gaming service provider 20. Various embodiments may not include a gaming service provider. The computer 18 and/or gaming service provider 20 may reside within, adjacent to, or remote from a gaming provider (not shown in FIG. 1). The gaming service provider may be an actual controller of games, such as a casino. As an example, a gaming service provider may be located on the grounds of a casino and the computer 18 may be physically within the geographic boundaries of the gaming service provider. As discussed, however, other possibilities exist for remote location of the computer 18 and the gaming service provider 20. Computer 18 may function as a gaming server. Additional computers (not expressly shown) may function as database management computers and redundant servers, for example.

[0788] In various embodiments, software resides on both the gaming communication device 13 and the computer 18. Software resident on gaming communication device 13 may be operable to present information corresponding to gaming activities (including gambling and non-gambling activities discussed herein) to the user. The information may include, without limitation, graphical representations of objects associated with the activities, and presentation of options related to the activities and selectable by the user. The gaming communication device software may also be operable to receive data from the computer and data input by the user. Software resident on the computer may be able to exchange data with the gaming communication device, access additional computers and data storage devices, and perform all of the functions described herein as well as functions common to known electronic gaming systems.

[0789] Gaming information transmitted across network 16 may include any information, in any format, which is necessary or desirable in the operation of the gaming experience in which the user participates. The information may be transmitted in whole, or in combination, in any format including digital or analog, text or voice, and according to any known or future transport technologies, which may include, for example, wireline or wireless technologies. Wireless technologies may include, for example, licensed or license-exempt technologies. Some specific technologies which may be used include, without limitation, Code Division Multiple Access (CDMA), Global System for Mobile Communication (GSM), General Packet Radio Service (GPRS), WiFi (802.11x), WiMax (802.16x), Public Switched Telephone Network (PSTN), Digital Subscriber Line (DSL), Integrated Services Digital Network (ISDN), or cable modem technologies. These are examples only and one of ordinary skill will understand that other types of communication techniques are also contemplated. Further, it will be understood that additional components may be used in the communication of information between the users and the gaming server. Such additional components may include, without limitation, lines, trunks, antennas, switches, cables, transmitters, receivers, computers, routers, servers, fiber optical transmission equipment, repeaters, amplifiers, etc.

[0790] In some embodiments, the communication of gaming information takes place without involvement of the Internet. However, in some embodiments, a portion of the gaming information may be transmitted over the Internet. Also, some or all of the gaming information may be transmitted partially over an Internet communications path. In some embodiments, some information is transmitted entirely or partially over the Internet, but the information is either not gaming information or is gaming information that does not need to be maintained secretly. For instance, data that causes a graphical representation of a table game on the user's gaming communication device might be transmitted at least partially over the Internet, while wagering information transmitted by the user might be transmitted entirely over a non-Internet communications network.

[0791] According to some embodiments, as shown in FIG. 2 for example, the communications network comprises a cellular network 22. Cellular network 22 comprises a plurality of base stations 23, each of which has a corresponding coverage area 25. Base station technology is generally known and the base stations may be of any type found in a typical cellular network. The base stations may have coverage areas that overlap. Further, the coverage areas may be sectorized or non-sectorized. The network also includes mobile stations 24, which function as the gaming communication devices used by users to access the gaming system and participate in the activities available on the gaming system. Users are connected to the network of base stations via transmission and reception of radio signals. The communications network also includes at least one voice/data switch, which may be connected to the wireless portion of the network via a dedicated, secure landline. The communications network may also include a gaming service provider, which is likewise connected to the voice/data switch via a dedicated, secure landline. The voice/data switch may be connected to the wireless network of base stations via a mobile switching center (MSC), for example and the landline may be provided between the voice/data switch and the MSC.

[0792] Users access the gaming system by way of mobile stations which are in communication with, and thus part of, the communications network. The mobile station may be any electronic communication device that is operable in connection with the network as described. For example, in this particular embodiment, the mobile station may comprise a cellular telephone.

[0793] In various embodiments, in the case of a cellular network for example, the gaming system is enabled through the use of a private label carrier network. Each base station is programmed by the cellular carrier to send and receive private secure voice and/or data transmissions to and from mobile station handsets. The handsets may be pre-programmed with both gaming software and the carrier's authentication software. The base stations communicate via private T1 lines to a switch. A gaming service provider leases a private T1 or T3 line, which routes the calls back to gaming servers controlled by the gaming service provider. Encryption can be installed on the telephones if required by a gaming regulation authority, such as a gaming commission.

[0794] The cellular network may be a private, closed system. Mobile stations communicate with base stations and base stations are connected to a centralized switch located within a gaming jurisdiction. At the switch, voice calls are transported either locally or via long distance. Specific service provider gaming traffic is transported from the central switch to a gaming server at a host location, which can be a casino or other location.

[0795] As subscribers launch their specific gaming application, the handset will only talk to certain base stations with cells or sectors that have been engineered to be wholly within the gaming jurisdiction. For example, if a base station is close enough to pick up or send a signal across state lines, it will not be able to communicate with the device. When a customer uses the device for gaming, the system may prohibit, if desired, the making or receiving voice calls. Moreover, voice can be eliminated entirely if required. Further, the devices may not be allowed to "connect" to the Internet. This ensures a high level of certainty that bets/wagers originate and terminate within the boundaries of the gaming jurisdiction and the "private" wireless system cannot be circumvented or bypassed. Although in some embodiments some data and/or voice traffic may be communicated at least partially over the Internet, the communication path may not include the Internet in other embodiments. Alternatively, in some embodiments, certain non-gaming information may be transported over a path which includes the Internet, while other information relating to the gaming activities of the system is transported on a path that does not include the Internet.

[0796] As shown in FIG. 3, a gaming communication device 32 is in communication with a gaming service provider over a network 34. The gaming service provider preferably has one or more servers, on which are resident various gaming and other applications. As shown in FIG. 3, some example gaming applications include horse racing and other sports, financial exchange, casino and/or virtual casino, entertainment and other events exchange, and news and real time entertainment. Each of these applications may be embodied in one or more software modules. The applications may be combined in any possible combination. Additionally, it should be understood that these applications are not exhaustive and that other applications may exist to provide an environment to the user that is associated with any of the described or potential activities.

[0797] In another embodiment, as shown in FIG. 4, for example, the communications network comprises a private wireless network. The private wireless network may include, for example, an 802.11x (WiFi) network technology to cover "Game Spots" or "Entertainment Spots." In FIG. 4, various WiFi networks are indicated as networks 41. Networks 41 may use other communications protocols to provide a private wireless network including, but not limited to, 802.16x (WiMax) technology. Further, networks 41 may be interconnected. Also, a gaming system may comprise a combination of networks as depicted in FIG. 4. For example, there is shown a combination of private wireless networks 16, a cellular network comprising a multi-channel access unit or sectorized base station 42, and a satellite network comprising one or more satellites 46.

[0798] With respect to the private wireless network, because the technology may cover small areas and provide very high-speed throughput, the private wireless network is particularly well-suited for gaming commission needs of location and identity verification for the gaming service provider products. The gaming spots enabled by networks 41 may include a current casino area 48, new areas such as swimming pools, lakes or other recreational areas 49, guest rooms and restaurants such as might be found in casino 48 or hotels 45 and 47, residential areas 40, and other remote gaming areas 43. The configuration of the overall gaming system depicted in FIG. 4 is intended only as an example and may be modified to suit various embodiments.

[0799] In some embodiments, the system architecture for the gaming system includes:
(1) a wireless LAN (Local Access Network) component, which consists of mostly 802.11x (WiFi) and/or 802.16x WiMax technologies; robust security and authentication software; gaming software; mobile carrier approved handsets with Windows(R) or Symbian(R) operating systems integrated within; and
(a) CDMA-technology that is secure for over-the-air data protection;
(b) at least two layers of user authentication, (that provided by the mobile carrier and that provided by the gaming service provider);
(c) compulsory tunneling (static routing) to gaming servers;
(d) end-to-end encryption at the application layer; and
(e) state-of-the-art firewall and DMZ technologies;
(2) an MWAN (Metropolitan Wireless Access Network), which consists of licensed and license-exempt, point-to-point links, as well as licensed and license-exempt, point-to-multi-point technologies;
(3) private MAN (Metropolitan Access Network) T1 and T3 lines to provide connectivity

where wireless services cannot reach; and
(4) redundant private-line communications from the mobile switch back to the gaming server.

[0809] Each of the "Game Spots" or "Entertainment Spots" is preferably connected via the MWAN/MAN back to central and redundant game servers. For accessing the private wireless networks 41, the gaming communication devices may be WiFi- or WiMax-enabled PDAs or mini-laptops, and do not have to be managed by a third-party partner.

[0810] In various embodiments, the gaming system includes a location verification feature, which is operable to permit or disable gaming from a remote location depending upon whether or not the location meets one or more criteria. A criterion may be, for example, whether the location is within a pre-defined area in which gaming is permitted by law. As another example, a criterion may be whether the location is in a no-gaming zone, such as a school. The location verification technology used in the system may include, without limitation, "network-based" and/or "satellite-based" technology. Network-based technology may include such technologies as multilateration, triangulation and geo-fencing, for example. Satellite-based technologies may include global positioning satellite (GPS) technology, for example.

[0811] As previously discussed, the cellular approach preferably includes the use of at least one cellular, mobile, voice and data network. For gaming in certain jurisdictions, such as Nevada for example, the technology may involve triangulation, global positioning satellite (GPS) technology, and/or geo-fencing to avoid the potential for bets or wagers to be made outside Nevada state lines. In some embodiments, the network would not cover all of a particular jurisdiction, such as Nevada. For instance, the network would not cover areas in which cellular coverage for a particular base station straddled the state line or other boundary of the jurisdiction. This is done in order to permit the use of location verification to insure against the chance of bets originating or terminating outside of the state. Triangulation may be used as a method for preventing gaming from unapproved locations. Triangulation may be accomplished, for example, by comparing the signal strength from a single mobile station received at multiple base stations, each having GPS coordinates. This technology may be used to pinpoint the location of a mobile station. The location can then be compared to a map or other resource to determine whether the user of the mobile station is in an unapproved area, such as a school. Alternatively, GPS technology may be used for these purposes.

[0812] As shown in FIG. 5, the gaming system includes a plurality of gaming communication devices 54, 55, and 56. Device 54 is located outside the gaming jurisdiction 58. Devices 55 and 56 are both located inside gaming jurisdiction 58. However only device 56 is located within geo-fence 57, which is established by the coverage areas of a plurality of base station 53. Thus, geo-fencing may be used to enable gaming via device 56 but disable gaming via devices 54 and 55. Even though some gaming communication devices that are within the gaming jurisdiction 58, such as device 55, are not permitted access to the gaming system, the geo-fence 57 ensures that no gaming communication devices outside jurisdiction 58, such as device 54, are permitted access.

[0813] Geo-fencing may not specify location. Rather, it may ensure that a mobile station is within certain boundaries. For instance, geo-fencing may be used to ensure that a mobile station beyond state lines does not access the gaming system. Triangulation on the other hand may specify a pinpoint, or near-pinpoint, location. For example, as shown in FIG. 5, device 56 is triangulated between three of the base stations 53 to determine the location of device 56. Triangulation may be used to identify whether a device, such as a mobile station, is located in a specific spot where gambling is unauthorized (such as, for example, a school). Preferably, the location determination technology utilized in conjunction with the present invention meets the Federal Communication Commission's (FCC's) Phase 2 E911 requirements. Geological Institute Survey (GIS) mapping may also be utilized to compare identified coordinates of a gaming communication device with GIS map features or elements to determine whether a device is in an area not authorized for gaming. It should be noted that any type of location verification may be used such as triangulation, geo-fencing, global positioning satellite (GPS) technology, or any other type of location determining technology, which can be used to ensure, or provide an acceptable level of confidence, that the user is within an approved gaming area.

[0814] In various embodiments, location verification is accomplished using channel address checking or location verification using some other identifying number or piece of information indicative of which network or portion of a network is being accessed by the gaming communication device. Assuming the usage of an identifying number for this purpose, then according to one method of location checking, as an example, a participant accesses the gaming system via a mobile telephone. The identifying number of the mobile telephone, or of the network component being accessed by the mobile telephone, identifies the caller's connection to the mobile network. The number is indicative of the fact that the caller is in a defined area and is on a certain mobile network. A server application may be resident on the mobile telephone to communicate this information via the network to the gaming service provider. In a some embodiments, the identifying number or information is passed from a first network provider to a second network provider. For example, a caller's home network may be that provided by the second provider, but the caller is roaming on a network (and in a jurisdiction) provided by the first provider. The first provider passes the identifying information through to the second provider to enable the second provider to determine whether the caller is in a defined area that does or does not allow the relevant gaming activity. In various embodiments, the gaming service provider either maintains, or has access to, a database that maps the various possible worldwide mobile network identifying numbers to geographic areas. Various embodiments contemplate using any number or proxy that indicates a network, portion of a network, or network component, which is being connected with a mobile telephone. The identifying number may indicate one or more of a base station or group of base stations, a line, a channel, a trunk, a switch, a router, a repeater, etc.

[0815] In various embodiments, when the user connects his mobile telephone to the gaming server, the gaming server draws the network identifying information and communicates that information to the gaming service provider. The software resident on the gaming communication device may incorporate functionality that will, upon login or access by the user, determine the user's location (based at least in part on the identifying information) and send a message to the gaming service provider. The identifying number or information used to determine location may be country-specific, state-specific, town-specific, or specific to some other definable boundaries.

[0816] In connection with any of the location determination methods, the gaming system may periodically update the location determination information. This may be done, for example, during a gaming session, at pre-defined time intervals to ensure that movement of the gaming communication device to an unauthorized area is detected during play, and not just upon login or initial access.

[0817] Thus, depending on the location determination technology being used, the decision whether to permit or prohibit a gaming activity may be made at the gaming communication device, at the gaming server, or at any of the components of the telecommunication network being used to transmit information between the gaming communication device and the gaming server (such as at a base station, for example).

[0818] An aspect of the private wireless network related to preventing gaming in unauthorized areas is the placement of sensors, such as Radio Frequency Identification (RFID) sensors on the gaming communication devices. The sensors trigger alarms if users take the devices outside the approved gaming areas. Further, the devices may be "tethered" to immovable objects. Users might simply log in to such devices using their ID and password.

[0819] In various embodiments, a gaming system may include the ability to determine the location of the gaming communication device within a larger property, such as a casino complex. This may allow certain functionalities of the device to be enabled or disabled based upon the location of the device within the property. For example, government regulations may prohibit using the device to gamble from the guest rooms of a casino complex. Therefore, particular embodiments may include the ability to determine the location of the device within the property and then disable the gambling functionality of the device from a guest room, or other area where gambling is prohibited. FIG. 6 illustrates an example of a wireless gaming system in which the location of a gaming communication device 604 may be determined in accordance various embodiments.

[0820] As shown in FIG. 6, a wireless gaming system comprises a wireless network that at least partially covers casino complex 600 in which one or more gaming communication devices 604 may be used to participate in a variety of gaming activities. The wireless network may comprise at least three signal detection devices 602, although various embodiments may include fewer or greater than three signal detection. As shown in FIG. 6, the wireless network comprises four signal detection devices 602, each located at one corner of casino complex 600. In various embodiments, these signal detection devices may comprise wireless access points, wireless routers, wireless base stations, satellites, or any other suitable signal detection device. Furthermore, although signal detection devices 602 are illustrated as being located on the boundaries of casino complex 600, signal detection devices may be located anywhere inside or outside of casino complex 600, provided the signal detection devices are operable to receive signals originating from a gaming communication device 604 inside casino complex 600. In various embodiments, signal detection devices 602 may also be used to transmit, as well as receive, signals to gaming communication device 604.

[0821] In various embodiments, casino complex 600 may be divided into one or more zones 608, which represent different areas of the casino complex, such as the lobby, guest rooms, restaurants, shops, entertainment venues, and pool areas. For example, as shown in FIG. 6, zone 608a may correspond to the casino lobby, zone 608b may correspond to guest rooms, zone 608c may correspond to restaurants, and zone 608d may correspond to the gaming floor of the casino. Each zone 608 may be further divided into one or more sub-zones 606, each specifying a particular location within zone 608. Sub-zones 606 may be arranged in a grid formation, each sub-zone 606 having a uniform size. In some embodiments, each sub-zone may comprise 9 square feet (i.e., 3 feet by 3 feet). In some embodiments, each sub-zone may comprise 100 square feet (i.e., 10 feet by 10 feet). The choice of the size of an area covered by a sub-zone may depend on administrator preferences, technical limitations of the wireless network, and governmental regulations, as well as other considerations.

[0822] Particular embodiments may use this mapping of casino complex 600 into a plurality of zones 608 and sub-zones 606 to determine the location of gaming communication device 604 within the complex. These embodiments may utilize the signal received by signal detection devices 602 from gaming communication device 604 to determine the location of the device.

[0823] In various embodiments, the location of gaming communication device 604 may be determined based upon the strength of the signal received by each signal detection device 602 from device 604. In various embodiments, this may be accomplished using a Received Signal Strength Indication (RSSI) value or any other suitable indication of signal strength. Generally, the closer a sub-zone is to a signal detection device, the stronger the signal the signal detection device will receive from a gaming communication device located in that sub-zone. Therefore, given a plurality of signal strength readings taken from different points in the casino complex (i.e., signal detection devices 602), these different signal strength readings may be used to determine the location of the device.

[0824] With this in mind, each sub-zone 606 of casino complex 600 may be associated with a reference set of signal strengths received by the signal detection devices from a device located in that particular sub-zone. Typically, these values are generated, and periodically recalibrated, by taking a reference reading from a gaming communication device located that sub-zone. After each sub-zone is associated with a reference set of signal strengths, these reference signal strengths may be compared with the signal strengths received by the signal detection devices from a gaming communication device. Since each sub-zone has a unique set of signal strengths, this comparison may be used to identify the particular zone in which the gaming communication device is located.

[0825] In various embodiments, the location of gaming communication device 604 may be determined based upon an elapsed time between the transmission of the signal from device 604 and the receipt of the signal by each signal detection device 602. In various embodiments, this elapsed time may be determined based on a Time Difference of Arrival (TDOA), or any other suitable technology. As before in the case of signal strengths, each sub-zone 606may be associated with a predetermined, or reference, set of elapsed times from transmission to receipt of a signal from a gaming communication device. This set of elapsed times will be different for each sub-zone of the casino complex, as the time it takes a signal to reach each signal detection device will depend on the proximity of the sub-zone to each base station. By comparing the time from transmission to receipt of a signal received by the signal detection devices from a gaming communication device, the sub-zone in which the device is located may be determined.

[0826] Once the location of the gaming communication device has been determined, particular embodiments may then enable and/or disable particular functions of the device based on this determination. For example, as mentioned previously, particular embodiments may disable the gaming communication device's gambling functionality from a user's guest room, while still allowing the user to use other device functions, such as purchasing merchandise or services, or buying tickets to an entertainment event. Once the user leaves his or her guest room, the gambling functionality of the gaming communication device may be enabled. Similarly, particular embodiments may prevent the gaming communication device from being used to make financial transactions from the casino floor. Once the user leaves the casino floor, such functionality may be enabled. Similarly, other functionalities of the gaming communication device may be enabled or disabled based upon the location of the device within the property in accordance with various embodiments.

[0827] In various embdoiments, the various functionalities of the gaming communication device may be enabled or disabled based upon the zone 608 in which the device is located. In such embodiments, each zone 608 of the casino complex may be associated with a set of allowed activities. For example, the "lobby" zone 608a of the casino complex may have all activities allowed, while the "guest room" zone 608b of the property may have all activities allowed except gambling. Based upon the gaming communication device's location, the functionality of the gaming communication device may be limited to the set of allowed activities for the zone in which the device is located. As the gaming communication device travels from zone to zone, the location of the device may be re-determined, and the functionality of the device may be updated to reflect the set of allowed activities for the zone in which the device is now located.

[0828] Various embodiments may also use the location determination to send location-specific information to the gaming communication device. For example, a reminder that an entertainment event to which the user has tickets is about to begin may be sent to the user's device if the device (and therefore the user) is located in a different part of the casino complex. In another embodiment, a user may be alerted that the user's favorite dealer is on the casino floor if the user is located in his or her guest room.

[0829] In various embodiments, the location of the gaming communication device may be used to deliver goods and services purchased or ordered by the user of the device. For example, in various embodiments, the user may purchase food and beverages using the device. The location of the device may then be used to deliver the food and beverages to the user, even if the user relocates to another sub-zone after placing his or her order.

[0830] The determination of the gaming communication device's location may also be used to provide the user with directions to another part of the casino complex. For example, a user that is located on the casino floor that wishes to go to a specific restaurant within the complex may be given direction based upon his or her location. These directions may then be updated as the user progresses towards his or her desired location. In the event the user gets off-course, the location determination, which may be updated during the user's travel, may be used to alert the user that he/she has gotten off-course and then plot a new course to the desired destination.

[0831] It should be understood that the foregoing descriptions encompass but some of the implementation technologies that may be used, according to various embodiments. Other technologies may be used and are contemplated, according to various embodiments. Various embodiments may be performed using any suitable technology, either a technology currently existing or a technology which has yet to be developed.

User Profiles

[0832] According to various embodiments, the wireless gaming system can incorporate a user profile element. One or more user profiles may be created, maintained, and modified, for example, on one or more of the servers of the gaming system. Generally, the user profiles include information relating to respective users. The information may be maintained in one or more databases. The information may be accessible to the gaming server and/or to one or more mobile devices. The devices which may access the information may, according to certain embodiments, include gaming devices or gaming management devices. Gaming management devices may include wireless devices used by casino staff to provide gaming services or gaming management services.

[0833] Various embodiments include software and/or hardware to enable the provision, modification, and maintenance of one or more user profiles. Thus, one or more user profiles may each comprise a set of data maintained in a data storage device. The data set(s) for each respective user profile may reflect any of a number of parameters or pieces of information, which relate to the particular user(s) corresponding to the profile(s). Although not intended to be exhaustive, such information may include, for example, gaming activity preferences, such as preferred game and/or game configuration, preferred screen configuration, betting preferences, gaming location preferences, dining and other service preferences, and so forth. The information may also include user identity information, such as name, home address, hotel name and room number, telephone numbers, social security numbers, user codes, and electronic files of fingerprint, voice, photograph, retina scan, or other biometric information. User profile information may also include information relating to the user, but not determined by the user or the user's activities. Such information may include any information associated with, or made part of, a profile. For example, an entity such as a casino, may include as part of a profile certain rules governing the distribution of promotions or offers to the user. User profile information can include any codes, account numbers, credit information, approvals, interfaces, applications, or any other information which may be associated with a user. Thus, user profile information may include any information that is particular to a given user. For example, profile information may include the location(s) at which a particular user has played, skill levels, success levels, types of games played, and betting styles, and trends of information relating to the user's activities.

[0834] In various embodiments, user profile information may include concierge or other service information that is associated with a user. Concierge services may include restaurant services, entertainment services, hotel services, money management services, or other appropriate services that may be offered to the user of a gaming device. For example, restaurant services may include, without limitation, services that allow the user to order drinks, order food, make reservations, or perform other restaurant related activities. As another example, entertainment services may include, without limitation, services that allow the user to purchase show tickets, arrange appointments or services, virtually shop, arrange transportation, or perform other entertainment related activities. Hotel services may include, for example, services that allow the user to check in, check out, make spa appointments, check messages, leave messages, review a hotel bill, or perform other guest-related activities. Money management services may include, for example, services that allow the user to transfer funds, pay bills, or perform other money management activities.

[0835] The gaming system may be configured to establish a new profile for any user who is using a gaming device for the first time. Alternatively, a new profile may be established for a prior user who has not played for a predetermined time period. The gaming system may set up the profile, monitor user activities, adjust the profile, and adjust information (such as graphics) displayed to the user. The gaming system may be configured to use the profile information to alter the presentation of gaming information to the user. For example, if a prior user has returned to the gaming system, the system may consult the profile for the user and determine that in the prior session of gaming the user lost money on craps but won money on blackjack. Based on this information, the system may adjust the default gaming screen and present a blackjack table for the user. As a further example, the profile information may indicate that the majority of the user's prior blackjack time was spent on $25 minimum tables. The system may, accordingly, make a further adjustment to the gaming environment and make the blackjack table being presented a $25 table. In this sense, the gaming system enables personalized wireless gaming based on one or more criteria maintained in a user profile.

[0836] The user profiles may be established, maintained, and periodically updated as necessary to enable a gaming provider to provide an enhanced, current, and/or customized gaming experience. Updates may be undertaken based on any suitable trigger, such as the occurrence of an event, the occurrence of a user activity, or the passage of a certain predetermined time period. Any or all of the profile information may be updated.

Alerts

[0837] In some embodiments, the gaming system may be configured to initiate one or more alerts to one or more users based on any number of criteria. For instance, an alert may be based on the location of a user. The system may also be configured to keep track of other non-location dependent parameters. The initiation of an alert may depend on a time parameter. Gaming alerts can also be based on this and/or other information maintained in a user profile. Alerts can be prioritized for presentation and the content and display of the alerts may be customized by the user or another entity. As a related concept, the system may be configured to provide directions and/or maps. Another related concept involves enabling a user to view a certain activity or area remotely. The alert may be generated in response to the existence of data within a user profile. Additionally, the content and presentation of the alert may be determined based on information in the user profile. Thus, when the alerts occur and what the alerts indicate may be customized or tailored according to user preferences (or any other information maintained about the user (e.g., in a user profile).

[0838] In some embodiments, an alert may be presented or displayed to the user in a format determined, at least in part, by any of the parameters described or contemplated herein. For example, if the user is located outdoors, the display may be automatically brightened in order to allow the user to more easily view the alert. The alert may be presented in any one or a combination of textual, visual, oral, or other information exchange formats. Alerts presented to users on the screen of a gaming communication device, for example, may be configured in any desirable manner. Preferably, the information is displayed in a way as to most effectively utilize the screen real estate to convey the alert message. Thus, different alerts of differing types, or having differing priorities, can be displayed differently on the gaming device. For example, a more important alert can be displayed as a popup while secondary alerts scroll at the bottom of the screen. The player can register for alerts and determine his own particular alert configuration preferences.

[0839] According to some embodiments, directional information may be provided to one or more users. The directional information may be associated with an alert. The directional information may be based on any of the parameters described herein (e.g., profiles, alerts, locations, changes in play or other activities, etc). Directions may be given to activities, locations, seats, tables, recreational spots, restaurants, change cages, information booths, casinos, hotels, sports venues, theaters, etc. For example directions may be given to a particular table or gaming area, a casino other than the one where the user is presently located or where another user is located, a restaurant that is specified in a user profile, a sports book area of a casino, a hotel room, etc.

[0840] The directions can be presented orally, textually, and/or graphically (e.g., as map with zoom capabilities). An example of how directions would be provided involves a user profile indicating that the user likes to play high-limit blackjack on Saturday nights, but that the user does not have a particular casino preference. If the user enters any casino for which the system is operable, the system provides the user with an alert inviting the player to the high-limit blackjack tables and directional information in the form of a visual route. Another example involves a user leaving a sports book in a casino and the user has indicated that he wants to play craps. The device gives walking directions to the craps tables. Another example involves a user that has a preferred list of dinner restaurants. At a predetermined time (e.g., 8:00 pm), the system presents the user with the list, lets the user make a selection and a reservation. The system then provides the user with verbal directions from the user's current location to the selected restaurant. The system may also be configured to provide ancillary information based, at least in part, on the alert, the profile, or the directional information being provided. For example, the system may notify a user that the user will need a cab, or will need to take the tram, or will need a jacket and tie, or will need an umbrella, etc. depending on where the user is going and the route he is taking.

[0841] According to various embodiments, the system enables a user to view a certain activity or area remotely. For example, cameras (or other viewing devices) may be disposed throughout a casino property (or other relevant area). At kiosks, or on the wireless gaming devices, users can "peek" into one or more selected areas to see the activity in the selected area(s). For example, from the pool, a user can tell if the craps tables have changed limits or are filling up with people. From the craps table, a user can see if the restaurant or bar is becoming crowded.

[0842] According to various embodiments, the operation of the alerts module and the alerts methods are integrated with various techniques for managing user profile information. An example of this aspect is that the system may be configured to recognize that a user has certain preferred dealers or stickmen when playing certain casino games. When those dealers or stickmen are on duty, and if the user is located in a certain area, or within a certain distance, an alert may be sent inviting the user to participate in the gaming activity at the particular table where the dealer or stickman is on duty.

[0843] Thus, when user profile information indicates that a one or more predetermined criteria are met, the system may send an alert to the corresponding user or to another user. For example, the system may "learn" that a player is a fan of certain sports teams. The system monitors information about upcoming events that involve those teams and, at a predetermined time, checks to see if the user has placed a bet on the event(s). If not, the system invites the user to visit a sports book to make a bet. As another example, the system knows a user prefers $10 minimum tables and alerts the user to the opening of a seat at such a table. As another example, the alerts can be triggered by information which is not directly related to or associated with the particular user (e.g., non-user specific information). For instance an alert might be triggered by a certain time or the occurrence of a certain event (e.g., the odds given on a certain sports event changing by a certain predetermined amount).

Service Applications

[0844] According to various embodiments, gaming services may be provided as an application add-on to a pre-existing communication or data service. Thus, gaming service applications may be made available to customers of a pre-existing communication or data service. For example, customers of a particular wireless telephone or data service may be offered any one or combination of the various gaming service applications discussed herein as an additional feature that is bundled with the telephone or data service. Although this document may refer to the communication service bundled with offered gaming service applications as including pre-existing communication services, it is recognized that the gaming services applications may be offered and accepted as part of a package with newly-activated communications service plan. In still other embodiments, the gaming service may be established first and the communication service may be added later.

[0845] The gaming service applications bundled with, or otherwise offered in conjunction with communication services, may be customized to meet the needs of the customers, service providers, or both. For example, a service provider may elect to make certain gaming service applications available to only a subset of the service providers' customers. Accordingly, not all customers associated with a service provider may be offered gaming services. As an another example of customized gaming service applications, a communication service may offer customers a number of gaming service plans which may provide different levels of service. For example, certain services such as advertisement services and/or promotional services may be free to customers of the communications service. Such levels of service may be customer-selected, service provider-selected, or both.

[0846] Customers may be billed separately for add-on gaming services, or in conjunction with the invoice the customer already receives for the pre-existing communications service. For instance, in certain embodiments, gaming services may be billed as an add-on in the same way that Caller ID services, call waiting services, and call messaging services result in fees that are in addition to the basic fees associated with communication services.

Peer-To-Peer Wireless Gaming

[0847] According to various embodiments, gaming services enable peer-to-peer wireless gaming. Specifically, the system may enable multiple players to participate in the same gaming activity at the same time from dispersed locations. This may be particularly desirable in the case of certain games such as, but without limitation, horse racing, poker, and blackjack. The system may also enable a single player to participate in multiple positions with respect to a particular game. For example, a user may be permitted to play multiple hands of blackjack. Particular aspects include such features as providing assistance to a user in finding a particular activity. For example, a first player may want to play poker at a six-person table. The gaming system may be used to identify such a poker table that has a position available for the first user's participation. Additionally or alternatively, a first player might want to play poker at the same table as a second player, and the system may be configured to assist the first player in finding a game in which the second player is already participating.

[0848] Location determination techniques may be incorporated to enable peer-to-peer gaming or related services. For example, a "buddy network" may be established to track members of a selected group. For example, a group of friends might all be in a gambling jurisdiction but be located at various dispersed places within that jurisdiction. The gaming system allows the establishment of a private buddy network of peers for this group of friends. The system enables one or more members of the group to track one or more other members of the group. In various embodiments, the system may also allow messages from and to one or more group members. For example, the system also allows members to invite other members to participate in certain wireless gaming activities. Additionally or alternatively, the system may allow members of the group to bet on the performance of another member of the group who is participating in a virtual or actual game.

[0849] Location determination techniques may also be incorporate to establish an "alert system." The alert system may be used to invite certain types of players to participate in a gaming activity. Criteria may then be used to identify users of gaming devices that meet the criteria. For example, a gaming participant may wish to initiate a gaming activity with other users of gaming devices that qualify as "high rollers" or "high stakes gamers." As other examples, a celebrity user may wish to initiate a gaming activity with other celebrities, or a senior citizen may wish to initiate a gaming activity with other senior citizens. In each instance, the user may identify criteria that may then be used to identify other gaming participants that meet these criteria for the initiation of a peer-to-peer gaming event.

[0850] It should be understood that the foregoing descriptions encompass but some of the implementation technologies that may be used, according to various embodiments. Other technologies may be used and are contemplated, according to various embodiments. Various embodiments may be performed using any suitable technology, either a technology currently existing or a technology which has yet to be developed.

Gaming and Wireless System

[0851] Various embodiments include a gaming system including hand-held personal gaming devices. The gaming system is adapted to present one or more games to a user of one of the hand-held gaming devices.

[0852] In various embodiments, the gaming system includes a portable gaming device or interface. The portable gaming device has a display for displaying game information to a player, at least one input device for receiving input from the player and is capable of receiving and sending information to a remote device/location. The gaming system also includes a game server for generating game data, transmitting game data to the portable gaming device and receiving information, such as player input, from the portable gaming device. The gaming system further includes a payment transaction server for validating payment and establishing entitlement of a player to play a game via the portable gaming device as provided by the game server.

[0853] In various embodiments, the gaming system includes one or more stationary gaming machines or other devices capable of printing tickets having a value associated therewith. The portable gaming device includes a ticket reader for reading ticket information for use by the payment transaction server in verifying the associated value for permitting the player to play the game.

[0854] In one or more embodiments, the portable gaming devices communicate with other devices (such as the game server) via a wireless communication channel. Appropriate relays and transceivers are provided for permitting the wireless communication.

[0855] In one or more embodiments, the portable gaming device includes a plurality of interfaces for changing the configuration of the gaming device or interacting with one or more transaction servers. In some embodiments, a login interface is provided for receiving login information regarding a user of the device. In various embodiments, the number of interfaces or other functions or features displayed or permitted to be accessed are configured depending upon the user of the device. In the event a gaming representative identifies himself, interfaces permitting access to a variety of control functions may be provided. In the event a player identifies themselves, such control functions may not be accessible, but instead only consumer-related functions may be accessible such as game play.

[0856] In one or more embodiments the gaming system includes one or more transaction servers, such as a food transaction server. Using an interface of the portable gaming device a player or other user may request services from the food transaction server. For example, a player may request food, drink, a restaurant reservation or other service.

[0857] One or more embodiments comprise a method of playing a game via a portable gaming device associated with a gaming network. In some embodiments, a player obtains a portable gaming device, such as by checking out the device from the hostess station of a restaurant or the front desk of a hotel/casino. The player provides value to the gaming operator, such as a credit card or cash deposit. This value is associated with the server and matched with a ticket number, player tracking number or other identifier.

[0858] The game device is configured for player play using the login interface. The act of logging in may be performed by the player or the gaming operator. The player next establishes entitlement to obtain services, such as the playing of a game, by showing the existence of value. In some embodiments, the player scans his ticket using the ticket reader of the device. The scanned information is transmitted to the payment transaction server for verifying entitlement of the player to play a game or obtain other services. In the event the entitlement is verified, then the player is permitted to engage in the play of a game or request service.

[0859] In the event a player wishes to play a game, the player indicates such by selecting a particular game using a game play interface. Upon receipt of such an instruction, the game server generates game data and transmits it to the personal gaming device. The transmitted data may comprise sound and video data for use by the personal gaming device in presenting the game. The player is allowed to participate in the game by providing input to the game server through the personal gaming device. The game server determines if the outcome of the game is a winning or losing outcome. If the outcome is a winning outcome, an award may be given. This award may be cash value which is associated with the player's account at the payment transaction server. If the outcome is a losing outcome, then a bet or wager placed by the player may be lost, and that amount deducted from the player's account at the transaction server.

[0860] FIG. 8 is a block diagram of a gaming system in accordance with various embodiments.

[0861] As illustrated, the gaming system B20 includes a plurality of gaming machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j. In some embodiments, these gaming machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j are of the stationary type. In general, the gaming machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j are arranged to present one or more games to a player. In various embodiments, the games are of the type requiring the placement of a wager or bet and are of the type by which a player receiving a winning outcome is provided an award, such as a monetary award. These devices may comprise for example, video poker and slot machines. In addition, the gaming system B20 includes one or more hand-held, portable gaming devices (PGDs) B24. The PGD B24 is also arranged to present one or more games to a player, and as described below, may be used as an access point for a variety of other services. The device referred to herein as a "personal gaming device" may be referred to by other terminology, such as a portable gaming interface, personal game unit or the like, but regardless of the name of the device, such may have one or more of the characteristics herein.

[0862] In addition, in various embodiments, the PGD B24 is in communication with at least one gaming server B28. As described below, in various embodiments, the one or more games which are presented via the PGD B24 to the player are provided by the gaming server B28.

[0863] The gaming machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j and each PGD B24 is in communication with a payment system referred to herein as the "EZ-Pay" system. This system includes a server B26 for receiving and transmitting information. In general, the EZ Pay system is utilized to accept payment from a player for the playing of games and obtaining of other goods and services, and for paying a player winnings or awards.

[0864] In the embodiments illustrated, the gaming system B20 includes other servers B30, B32 for transmitting and/or receiving other information. In some embodiments, one server B30 comprises a prize transaction server. Another server B32 comprises a food transaction server. In a some embodiments, information may be transmitted between the PGD B24 and these servers B30, B32.

[0865] The EZ Pay system, according to various embodiments, will now be described in more detail with reference to FIG. 9. The EZ Pay system may constitute an award ticket system which allows award ticket vouchers to be dispensed in lieu of the traditional coin awards or reimbursements when a player wins a game or wishes to cash out. These tickets may also be used by gaming machines and other devices for providing value, such as for payment of goods or services including as a bet or ante for playing a game.

[0866] FIG. 9 illustrates some embodiments of such a system in block diagram form. As illustrated, a first group of gaming machines B22a, B22b, B22c, B22d, and B22e is shown connected to a first clerk validation terminal (CVT) B34 and a second group of gaming machines B22f, B22g, B22h, B22i, and B22j is shown connected to a second CVT B36. All of the gaming machines print ticket vouchers which may be exchanged for cash or accepted as credit or indicia in other gaming machines. When the CVTs B34,B36 are not connected to one another, a ticket voucher printed from one gaming machine may only be used as indicia of credit in another gaming machine which is in a group of gaming machines connected to the same CVT. For example an award ticket printed from gaming machine B22a might be used as credit of indicia in gaming machines B22b, B22c, B22d, and B22e, which are connected to the common CVT B34, but may not be used in gaming machines B22f, B22g, B22h, B22i, and B22j since they are each connected to the CVT B36.

[0867] The CVTs B34,B36 store ticket voucher information corresponding to the outstanding ticket vouchers that are waiting for redemption. This information is used when the tickets are validated and cashed out. The CVTs B34,B36 store the information for the ticket vouchers printed by the gaming machines connected to the CVT. For example, CVT B34 stores ticket voucher information for ticket vouchers printed by gaming machines B22a, B22b, B22c, B22d, and B22e. When a player wishes to cash out a ticket voucher and the CVTs B34,B36 are not connected to one another, the player may redeem a voucher printed from a particular gaming machine at the CVT associated with the gaming machine. To cash out the ticket voucher, the ticket voucher is validated by comparing information obtained from the ticket with information stored with the CVT. After a ticket voucher has been cashed out, the CVT marks the ticket as paid in a database to prevent a ticket voucher with similar information from being cashed multiple times.

[0868] Multiple groups of gaming machines connected to the CVTs B34,B36 may be connected together in a cross validation network B38. The cross validation network typically comprises one or more concentrators B40 which accept input from two or more CVTs and enables communications to and from the two or more CVTs using one communication line. The concentrator B40 is connected to a front end controller B42 which may poll the CVTs B34,B36 for ticket voucher information. The front end controller B42 is connected to an EZ pay server B26 which may provide a variety of information services for the award ticket system including accounting B44 and administration B46.

[0869] The cross validation network allows ticket vouchers generated by any gaming machine connected to the cross validation network to be accepted by other gaming machines in the cross validation network B38. Additional, the cross validation network allows a cashier at a cashier station B48, B50, B52 to validate any ticket voucher generated from a gaming machine within the cross validation network B38. To cash out a ticket voucher, a player may present a ticket voucher at one of the cashier stations B48, B50, B52. Information obtained from the ticket voucher is used to validate the ticket by comparing information on the ticket with information stored on one of the CVTs B34,B36 connected to the cross validation network B38. As tickets are validated, this information may be sent to another computer B54 providing audit services.

[0870] As described above, the gaming system B20 may also include one or more hand-held PGDs B24. In various embodiments, the PGD B24 is a portable device capable of transmitting and receiving information via a wireless communication link/network.

[0871] Referring again to FIG. 8, the gaming system B20 may include a printer B56, wireless communication relays B58 and B60, and wireless transceivers B62, B64, B66 and B68 connected to the remote transaction servers B26, B28, B30 and B32. In various embodiments, a player may obtain the PGD B24, and after being provided with the appropriate authority, may play one or more games and/or obtain other services including food services or accommodation services.

[0872] FIG. 10 illustrates the PGD B24 and a block diagram of a game and service system which may be implemented by the gaming system B20 illustrated in FIG. 8. In various embodiments, the game and service system B100 is comprised of at least one PGD B24 and a number of input and output devices. The PGD B24 is generally comprised of a display screen B102 which may display a number of game service interfaces B106. These game service interfaces B106 are generated on the display screen B102 by a microprocessor of some type (not shown) within the PGD B24. Examples of a hand-held PGD B24 which may accommodate the game service interfaces B106 shown in FIG. 10 are manufactured by Symbol Technologies, Incorporated of Holtsville, N.Y. The interface or menu data may be stored in a local memory, or the data may be transmitted to the PGD B24 from a remote location (such as a data server). This reduces the memory requirement of the device.

[0873] The game service interfaces B106 may be used to provide a variety of game service transactions and gaming operations services, including the presentation for play by a user of one or more games. The game service interfaces B106, including a login interface B105, an input/output interface B108, a transaction reconciliation interface B110, a ticket validation interface B115, a prize services interface B120, a food services interface B125, an accommodation services interface B130, a gaming operations interface B135, and a game play interface B137 may be accessed via a main menu with a number of sub-menus that allow a game service representative or player to access the different display screens relating to the particular interface.

[0874] In one or more embodiments, some or all of the interfaces may be available to a user of the PGD B24. For example, in one or more embodiments, the PGD B24 may have a dual purpose of both being usable by a player to play games and engage in other activities, and also be used by gaming operations personnel for use in providing services to players and performing administrative functions. In various embodiments, certain PGDs B24 may be specially configured for use only by players, and other PGDs B24 may be specially configured for use only by gaming or other personnel. In such event, the interfaces B106 may be custom programmed.

[0875] In one or more embodiments, only certain interfaces B106 may be displayed, depending on the status of the user of the PGD B24. In some embodiments, the particular interfaces B106 which are displayed and thus accessible for use are determined by the status of the user as indicated through a login function. In various embodiment, when the PGD B24 is operable (such as when a power button is activated) the default status for the PGD B24 is the display of the login interface B105. Once a user of the PGD B24 has logged in, then the status of the PGD display is changed.

[0876] In one or more embodiments, the login interface B105 may allow a game service representative to enter a user identification of some type and verify the user identification with a password. When the display screen B102 is a touch screen, the user may enter the user/operator identification information on a display screen comprising the login interface B105 using an input stylus B103 and/or using one or more input buttons B104. Using a menu on the display screen of the login interface, the user may select other display screens relating to the login and registration process. For example, another display screen obtained via a menu on a display screen in the login interface may allow the PGD B24 to scan a finger print of the game service representative for identification purposes or scan the finger print of a game player.

[0877] In the event a user identifies themselves as a gaming operator or representative, then the PGD B24 may be arranged to display one or more other interfaces such as those listed above and described in detail below. In one or more embodiments, the default status or login may be a "player" mode login.

[0878] In various embodiments, the login interface B105 may allow a player to identify themselves to configure the PGD B24 to permit the player to access a plurality of player services, such as playing games and the like. In various embodiments, the login interface B105 includes a request that the user identify themselves as a "player" or "authorized personnel." In the event "authorized personnel" is selected, then the above-referenced user identification (including password) may be requested. If "player" is selected, then in various embodiments the player is requested to provide an EZ pay ticket. As described in more detail below, in various embodiments, a player who wishes to play one or more games or obtain other goods or services uses an EZ pay ticket to provide the credit or payment therefor. The ticket may be obtained from a cashier or by play of another gaming device (such as devices B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j in FIG. 8). The ticket may be verified through the EZ pay system described above.

[0879] In various embodiments, the PGD B24 includes a ticket reader B145 and a card reader B140. In some embodiments, the ticket reader B145 may be of a variety of types. In some embodiments, the reader comprises a bar-code reading optical scanner. In this arrangement, a user of the PGD B24 may simply pass the bar-coded ticket in front of the bar-code reader. In some embodiments, the card reader B140 comprises a magnetic-stripe card type reader for reading information associated with a magnetic stripe of a card, such as a player tracking card.

[0880] After having provided the appropriate authorization, access may be provided to the user of the PGD B24 of one or more of the following interfaces B106.

[0881] In one or more embodiments, an authorized user may be provided with access to the input/output interface B108. In a various embodiments, such access is only provided to a game service operator and not a player. In one or more embodiments, the input/output interface B108 permits a user to select, from a list of devices stored in memory on the PGD B24, a device from which the PGD may input game service transaction information or output game service transaction information. For example, the PGD B24 may communicate with the ticket reader B145. As another example, the PGD B24 may input information from the card reader B140. Such input may be useful, for example, if a game service operator wishes to verify the authenticity of a player tracking card or the like.

[0882] The PGD B24 may output game and service transaction information to a number of devices. For example, to print a receipt, the PGD B24 may output information to a printer B150. In this game service transaction, the PGD B24 may send a print request to the printer B150 and receive a print reply from the printer B150. The printer B150 may be a large device at some fixed location or a portable device carried by the game service representative. As another example, the output device may be the card reader B140 that is able to store information on a magnetic card or smart card. Other devices which may accept input or output from the PGD B24 are personal digital assistants, microphones, keyboard, storage devices, gaming machines and remote transaction servers.

[0883] The PGD B24 may communicate with the various input mechanisms and output mechanisms using both wire and wire-less communication interfaces. For example, the PGD B24 may be connected to the printer B150 by a wire connection of some type. However, the PGD B24 may communicate with a remote transaction server B160 via a wire-less communication interface including a spread spectrum cellular network communication interface. An example of a spread spectrum cellular network communication interface is Spectrum 24 offered by Symbol Technologies of Holtsville, N.Y., which operates between about 2.4 and 2.5 Gigahertz. The information communicated using the wire-less communication interfaces may be encrypted to provide security for certain game service transactions such as validating a ticket for a cash pay out. Some devices may accommodate multiple communication interfaces. Such a spread spectrum network is but one possible communication scheme.

[0884] Another type of interface that may be stored on the PGD B24 is the award ticket validation interface B115. In some embodiments, this interface is only available to an authorized game service representative, and not a player. Some embodiments of the award ticket interface B115 may accommodate the EZ pay ticket voucher system and validate EZ pay tickets as previously described. However, when other ticket voucher systems are utilized, the award ticket validation interface B115 may be designed to interface with the other ticket voucher systems. Using the award ticket validation interface B115, a game service representative may read information from a ticket presented to the game service representative by a game player using the ticket reader and then validate and pay out an award indicated on the ticket.

[0885] In various embodiments, the award ticket contains game service transaction information which may be verified against information stored on a remote transaction server B160. To validate the ticket may require a number of game service transactions. For example, after obtaining game service transaction information from the award ticket, the PGD B24 may send a ticket validation request to the remote transaction server B160 using the spread spectrum communication interface and receive a ticket validation reply from the remote server B160. In particular, the validation reply and the validation request may be for an EZ pay ticket. After the award ticket has been validated, the PGD B24 may send a confirmation of the transaction to the remote server B160. Details of the game service transaction information validation process are described with the reference to FIG. 12. In various embodiments, the award ticket interface may be configured to validate award information from a smart card or some other portable information device or validate award information directly from a gaming machine.

[0886] As game and service transactions are completed, game and service transaction information may be stored on a storage device B155. The storage device B155 may be a remote storage device or a portable storage device. The storage device B155 may be used as a back-up for auditing purposes when the memory on the PGD B24 fails and may be removable from the PGD B24.

[0887] A type of game service interface that may be stored on the PGD B24 is the prize service interface B120. As an award on a gaming machine (i.e., machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j in FIG. 8) or while playing a game via the PGD B24, a game player may receive a ticket (such as issued by other machine) that is redeemable for merchandise including a bicycle, a computer or luggage or receive such an award directly (such as while playing the PGD B24 itself). Using the prize service interface B120, a game service representative or player may validate the prize service ticket and then check on the availability of certain prizes. For example, when the prize service ticket indicates the game player has won a bicycle, the game service representative may check whether the prize is available in a nearby prize distribution center. Alternatively, a player may be permitted to do the same thing. In some embodiments, a player may be awarded a prize of a particular level, there being one or more particular prizes on that level. In such events, the player may use the interface B120 to determine what prizes are currently available in the prize level just awarded. The PGD B24 may validate a prize ticket and check on the availability of certain prizes by communicating with a remote prize server. Further, the game service representative may have the prize shipped to a game player's home or send a request to have the prize sent to a prize distribution location. The game service transactions needed to validate the prize ticket including a prize validation request and a prize validation reply, to check on the availability of prizes and to order or ship a prize may be implemented using various display screens located within the prize interface. The different prize screens in the prize service interface B120 may be accessed using a menu located on each screen of the prize service interface. In some embodiments, the prize service interface B120 may be configured to validate prize information from a smart card or some other portable information device or validate award information directly from a gaming machine.

[0888] A type of game service interface that may be stored on the PGD B24 is the food service interface B125. As an award on a gaming machine or as compensation for a particular amount of game play, a game player may receive a free food or drink. Using the food service interface B125, the player may redeem the food or drink award, or a game service representative may validate such an award (for example, the award may be provided to a player of a gaming device B22a in the form of a ticket) and check on the availability of the award. For example, when the game player has received an award ticket valid for a free meal, the food service interface may be used to check on the availability of a dinner reservation and make a dinner reservation. As another example, the PGD B24 may be used to take a drink or food order by the player thereof. Such an order may be processed via the remote food server B32 (see also FIG. 8). The transactions needed to validate a food ticket or award, to check on the availability of food services, request a food service and receive a reply to the food service request may be implemented using various display screens located within the food service interface B125. These display screens may be accessed using a menu located on each screen of the food service interface. In some embodiments, the food service interface may be configured to validate food service information from a smart card or some other portable information device.

[0889] Another type of game service interface that may be stored on the PGD B24 is an accommodation service interface B130. As an award for game play or as compensation for a particular amount of game play, a game player may receive an award in the form of an accommodation service such as a room upgrade, a free night's stay or other accommodation prize. using the accommodation service interface B130, the player may check on the availability of certain accommodation prizes. For example, when the game player has received an award for a room upgrade, the accommodation service interface may be used to check on the availability of a room and to make a room reservation. Regardless of whether the player has won an accommodation award, the player may utilize the accommodation service interface B130 to reserve a room (such as an additional night's stay) or an upgrade to a room. In some embodiments, a player of a game may be issued a ticket (such as from a free-standing game device B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j in FIG. 8), and a gaming representative may use the accommodation service interface B130 in order to validate the player's award ticket and check on the availability of the award and institute the award. As another example, the PGD B24 may be used to order a taxi or some other form of transportation for a player at a gaming machine preparing to leave the game playing area. The game playing area may be a casino, a hotel, a restaurant, a bar or a store.

[0890] The PGD B24 may validate the accommodation service award and check on the availability of certain accommodation awards by communicating with a remote accommodation server. The transactions needed to validate the accommodation ticket, check on the availability of accommodation services, request an accommodation service and receive a reply to the accommodation service request may be implemented using various display screens located within the accommodation service interface. These display screens may be accessed using a menu located on each screen of the accommodation service interface. In some embodiments, the accommodation service interface may be configured to validate accommodation service information from a smart card or some other portable information device.

[0891] A type of game service interface that may be stored on the PGD B24 is a gaming operations service interface B135. Using the gaming service interface B135 on the PGD B24, a game service representative may perform a number of game service transactions relating to gaming operations. For example, when a game player has spilled a drink in the game playing area, a game service representative may send a request to maintenance to have someone clean up the accident and receive a reply from maintenance regarding their request. The maintenance request and maintenance reply may be sent and received via display screens selected via a menu on the screens of the gaming operations service interface. As another example, when a game service representative observes a damaged gaming machine such as a broken light, the game service representative may send a maintenance request for the gaming machine using the PGD B24. In one or more embodiments, a player may be permitted various options through the gaming service interface B135. For example, a player may be permitted to request a gaming service representative or attendant using the interface B135.

[0892] A type of game service interface that may be stored on the PGD B24 is a transaction reconciliation interface B110. In various embodiments, the PGD B24 contains a memory storing game service transaction information. The memory may record the type and time when a particular game service transaction is performed. At certain times, the records of the game service transactions stored within the PGD B24 may be compared with records stored at an alternate location. For example, for an award ticket validation, each time an award ticket is validated and paid out, a confirmation is sent to a remote server B160. Thus, information regarding the award tickets, which were validated and paid out using the PGD B24, should agree with the information regarding transactions by the PGD stored in the remote server B160. The transaction reconciliation process involves using the transaction reconciliation interface B110 to compare this information. In various embodiments, only a gaming service representative (and not a player) is permitted access to the transaction reconciliation interface B110.

[0893] A type of game service interface that may be stored on the PGD B24 is a voice interface B138. Using the spread spectrum cellular or other communication network incorporated into the PGD, a player and/or game service representative may use the PGD B24 as a voice communication device. This voice interface B138 may be used to supplement some of the interfaces previously described. For example, when a game player spills a drink the game service representative may send maintenance request and receive a maintenance reply using the voice interface B138 on the PGD B24. As another example, when a game player requests to validate a food service such as free meal, such a request may be made by the player or a game service representative at a restaurant or other location using the voice interface B138 on the PGD B24. In some embodiments, a player may be permitted to contact a player of another PGD B24, such as by inputting a code number assigned to the PGD B24 through which communication is desired. Such would permit, for example, a husband and wife using two different PGDs B24 to communicate with one another. The voice interface B138 may also permit a player to contact the front desk of a hotel/casino, an operator of a switchboard at the gaming location or the like.

[0894] A type of game service interface that may be stored on the PGD B24 is a game play interface B137. In various embodiments, a player is permitted to access the game play interface B137 in order to select from one or more games for play. The game play interface B137 may include a menu listing one or more games which the player may play via the PGD B24. In various embodiments, game play is facilitated with the game server B28 (see FIG. 8).

[0895] In one or more embodiments, the gaming control code is not resident at the PGD B24, but instead at a secure, remote server. Referring to FIG. 8, game play data is transmitted from the game server B28 to the PGD B24, and from the PGD B24 to the game server B28. Preferably, the PGD B24 is adapted to receive and process data, such as by receiving video data and processing the data to present the information on the display B102. Likewise, the PGD B24 is arranged to accept input and transmit that input or instruction to the game server B28. This arrangement has the benefit that nearly all aspects of the play of a game can be monitored, as it requires the game play data to pass to or from a remote location. This avoids, for example, storage of the gaming software at the PGD B24 where it might be tampered with, copied or the like.

[0896] In one or more embodiments, each PGD B24 has a unique identifier which is utilized to identify which PGD B24 data is transmitted from and to which data is to be transmitted to. In some embodiments, the game server B28 may thus be used to present the same or different games to a plurality of players using different PGDs B24, with the game data regarding a particular game being played at a particular PGD B24 being directed to that PGD B24 using its particular identifier.

[0897] As will be appreciated by those of skill in the art, the PGD B24 may have a variety of configurations. As stated above, the PGD B24 may be used in the gaming system B20 in which gaming code is not stored directly at the PGD. In such an embodiment, the PGD B24 may have a much more limited amount of data memory. In some embodiments, the PGD B24 includes a processor for executing control code, such as that necessary to operate the display B102, accept input from the stylus B103 or input buttons B104 or the like. In addition, the PGD B24 preferably includes a buffer memory for accepting data transmitted from the game server B28. This data may comprise data for displaying game information, such as video and sound content.

[0898] Various aspects of the use of the PGD B24 described above will now be described. In one or more embodiments, the PGD B24 may be used directly by a player. In various embodiments, a player may use the PGD B24 to play one or more games, and obtain products and services, such as food.

[0899] A method of use of the PGD B24, according to some embodiments, is illustrated in FIGS. 11(a) and 11(b). In general, a player must first obtain a PGD B24. For example, a player may check out a PGD B24 from a gaming operator. The player then establishes entitlement to use the PGD B24. In some embodiments, the player must indicate player status at the login interface, and obtain a valid ticket in order to activate the PGD B24. Once activated, the player is permitted to engage in a variety of transactions using the interfaces B106, such as playing a game, redeeming prizes and awards, placing food and drink orders, placing reservations, seeking gaming operator support and seeking a variety of other goods and services as described in more detail below.

[0900] One example of a method of use of the PGD B24 by a player will be described with reference to FIG. 11(a). In a first step B400, the player first obtains the PGD B24. In some embodiments, a gaming operator may have certain locations at which a player may obtain the PGD B24, such as the front desk of a hotel/casino, the hostess stand at a restaurant, from a gaming attendant or other location as desired. In some embodiments, a gaming operator may actually permit a player to retain the PGD B24, such as by renting, selling or giving the PGD B24 away to a player.

[0901] In a step B402, the PGD B24 is activated. In some embodiments, this step includes turning on the PGD B24 (such as with a power switch) and logging in. In some embodiments, when the PGD B24 is turned on, the login interface B105 is automatically displayed. The login interface B105 may include "player" and "authorized personnel" buttons which may be activated using the stylus B103. The player may indicate "player" status by selecting the player button with the stylus B103.

[0902] In some embodiments, the gaming operator may log the player in. For example, when a player obtains the PGD B24 from a hostess at a restaurant, the hostess may log in the player in player mode. In some embodiments, the gaming operator may have certain PGDs B24 which are for use by players and certain others which are for use by gaming personnel. In such event, the PGDs B24 which are configured for player status may automatically be configured for player mode after being turned on.

[0903] In a step B404, a player establishes entitlement to use the PGD B24. In some embodiments, this step comprises the player providing a valid ticket which is verifiable using the EZ pay portion of the gaming system B20. In some embodiments, a player may have obtained a ticket through play of a gaming machine, such as gaming machines B22a, B22b, B22c, B22d, B22e, B22f, B22g, B22h, B22i, B22j of the gaming system B20. In some embodiments, a player may be issued a ticket by a game service representative. For example, a player may provide credit at a cashier cage (such as with a credit card or cash) and be issued a ticket. A player may also pay cash or the like to a restaurant hostess and be issued a ticket.

[0904] Once the player has a ticket, the ticket may be scanned using the ticket reader B145 of the PGD B24. For example, the player may pass the ticket in front of the ticket reader B145. Once the information is read by the PGD B24, the data may be transmitted to the EZ pay server B26 for validation. Preferably, this validation confirms that the particular ticket is authorized, including the fact that it is outstanding and has value associated therewith.

[0905] In one or more embodiments, entitlement may be established in other manners. For example, in some embodiments, entitlement may be established with a player tracking or identification card which may be read using the card reader B140 of the PGD B24.

[0906] Establishing entitlement to use the PGD B24 may ensure that the player has funds for paying to obtain services and products available by use of the PGD B24. In one or more embodiments, however, this step may be eliminated. For example, in some embodiments, a player may be permitted to use the PGD B24 and then pay for goods or services in other manners. In some embodiments, a player may, for example, order food and then pay the server for the food using a room charge or cash at the time the food is delivered. In some embodiments, a player may use a credit card to pay to play games or to pay for food or the like. In such event, a credit card may be read by the card reader B140 at the time the services or products are to be provided or are ordered by the player.

[0907] In a step B406, the player is then permitted to select one or more selections from the interfaces B106. As stated above, a player may not be permitted access to all of the interfaces B106. In any event, a player may select, such as with the stylus B103, a service from the group of interfaces B106. An example of the engagement of a particular activity using the PGD B24 will be described below with reference to FIG. 11(b).

[0908] Once a player no longer desires to engage in any more activities using the PGD B24, the use session of the PGD B24 is ended in a step B408, and in one or more embodiments, the PGD B24 is returned to the gaming operator. In various embodiments, once a player no longer wishes to use the PGD B24, the player returns the PGD B24 to the gaming operator. At that time, the gaming operator may confirm that all transactions using the PGD B24 are closed or complete, and pay the player any winnings. In some embodiments, a player B24 is issued a new ticket representing the player's credit (including any payments made in order to first use the PGD B24, plus any winnings, less any expenditures).

[0909] An example of a method of using the PGD B24 wherein the player has selected the option of game play using the game play interface B137 will be described in detail with reference to FIG. 11(b). In a step B410 (which step comprises a particular embodiment of step B406 of FIG. 11(a)), a player has selected the event or service of "game play" using the game play interface B137.

[0910] In some embodiments, when a player has selected the game play interface B137, a menu may be displayed to the player of the one or more games which the player may be permitted to play. In some embodiments, when the player selects the game play interface B137, a signal is transmitted from the PGD B24 to the remote game server B28 instructing the game server B28 that the player wishes to play a game. In response, the game server B28 may send the latest game menu to the PGD B24 for display. In this arrangement, the menu of games which is available may be continuously updated at one or more central locations (such as the server B28) instead of at each PGD B24.

[0911] If the system B20 permits the player to select a game from a menu of games, then the method includes the step of the player selecting a particular game to be played. Once a game is selected, or if only a single game option is provided, then game play begins. In some embodiments, the game server B28 transmits data to the PGD B24 for use by the PGD B24 in presenting the game, such as video and audio content.

[0912] In some embodiments, in a step B412 a player is required to place a bet or ante to participate in a game. In some embodiments, the player may place the bet or ante using the EZ pay system. As stated above, the player preferably establishes entitlement to use the PGD B24 with an EZ pay ticket or other entitlement, which ticket demonstrates that the player has monies or credits on account which may be used to pay for goods and services. These services include game play services.

[0913] In some embodiments, when the player establishes entitlement to use the PGD B24, the value of the player's credits or monies are displayed to the player so that the player is visually reminded of these amounts. When a player begins play of a game, the player may input a bet and ante which is no more than the value of the credits or monies which the player has on account. Once a player has placed a bet or ante, that information is transmitted to the EZ pay server B26 and is deducted from the player's account. A new credit value is then displayed at the PGD B24 to the player.

[0914] In various embodiments, a player may provide credit for a bet or ante in other manners. For example, a player may swipe a credit card through the card reader B140 in order to provide the necessary credit for the bet or ante.

[0915] In a step B414, the player is then permitted to engage in the game. In some embodiments, game play comprises the game server B28 executing game code and transmitting information to the PGD B24 for presenting certain aspects of the game to the player. When necessary, the player is permitted to provide input, and the input data is transmitted from the PGD B24 to the game server B28.

[0916] As one example of a game, the game may comprise video poker. In this embodiment, the game server B28 executes code for randomly generating or selecting five cards. Data representing video images of the cards is transmitted to the PGD B24, where the images of the five dealt cards are displayed on the display screen B102.

[0917] The instruction "draw" or "stay" may be displayed to the player. At that time, the player may select one or more of the cards to hold or replace. In the event the player elects to replace any card, that instruction is transmitted to the game server B28 which then randomly generates or selects replacement cards. The replacement card data is transmitted to the PGD B24 and images of the replacement cards are displayed.

[0918] In the event the hand of five cards (including any replacement cards) is determined by the game server B28 to comprise a predetermined winning hand, then the player may be paid a winning amount. If not, then the player loses his bet or ante. This step comprises step B416 of the method, that of determining the outcome of the game.

[0919] If the outcome is a winning outcome, then the player may be paid a winning by crediting the player's account through the EZ pay server B26. In that event, the player's credits value as displayed is updated to reflect the player's winnings.

[0920] A player may then elect to play the game again, play a different game, or select one or more other services offered. In some embodiments, a "return to main menu" button or the like may be displayed to the player at all times, permitting the player to return to a display including the various interfaces B106.

[0921] In some embodiments, when the player has completed use of the PGD B24, the player returns the PGD B24 to the gaming operator. For example, the player may return the PGD B24 to a cashier cage or a game service operator. In various embodiments, the game service operator or other party then issues the player a ticket for any credit or value which remains in the player's account. The PGD B24 may then be deactivated so that it readied for use by another player. In some embodiments, the PGD B24 may be deactivated by turning its power off. In some embodiments, a "logout" interface or option may be provided which causes the PGD B24 to return to a default state seeking the login of a player or user.

[0922] The PGD B24 may be used by a game service operator. Several examples of a method of such use are detailed below in conjunction with FIGS. 8 and 9.

[0923] When a game service representative contacts a game player seeking a game service in the game playing area B70 (see FIG. 8), the game service representative uses an appropriate game service interface on the display screen of the PGD B24, as described with reference to FIG. 10, to provide the game service requested by the game player. For example, when a game player requests an EZ pay ticket validation, the game service representative brings the EZ pay ticket validation interface onto the display screen of the PGD B24 using menus available on the display screen B102. Then, the game service representative scans the EZ pay ticket using a ticket reader connected to the PGD B24 to obtain unique ticket information. Next, the PGD B24 sends an EZ pay ticket validation request using the wire-less communication interface to the EZ pay server B26.

[0924] In various embodiments, the ticket validation request is composed of one or more information packets compatible with the wire-less communication standard being employed. Using a wireless link B72, the one or more information packets containing the ticket validation request are sent to the transceiver B62 connected to the EZ pay server. The transceiver B62 is designed to receive and send messages from the one or more PGDs B24 in the game playing area B70 in a communication format used by the PGDs. Depending on the location of the PGD B24 in the game playing area B70, the communication path for the information packets to and from the PGD B24 may be through one or more wire-less communication relays including B58 and B60. For example, when the PGD B24 is located near gaming machine B22a, the communication path for a message from the PGD B24 to the EZ pay server B26 may be from the PGD B24 to the relay B60, from the relay B60 to the relay B58, from the relay B58 to the transceiver B62 and from the transceiver B62 to the EZ pay server B26. As the location of the PGD B24 changes in the game playing area B70, the communication path between the PGD B24 and the EZ pay server B26 may change.

[0925] After receiving an EZ pay ticket validation reply from the EZ pay server B26, the EZ pay ticket may be validated using an appropriate display screen on the PGD B24. After cashing out the ticket, the game service representative may send a confirmation of the transaction to the EZ pay server B26 using the PGD B24. The transaction history for the PGD B24 may be stored on the PGD B24 as well as the EZ pay server B26. Next, a receipt for the transaction may be printed out. The receipt may be generated from a portable printer carried by the game server representative ad connected to the PGD B24 in some manner or the receipt may be generated from a printer B56 at a fixed location.

[0926] After providing a number of game services comprising a number of game service transactions to different game players in the game playing area B70 using the PGD B24, a game service representative may log-off of the PGD B24 and return it to location for secure storage. For example, at the end of a shift, the game service representative may check the PGD B24 at some of the locations, the device is unassigned to the particular game service representative and then may be assigned to another game service representative. However, before the PGD B24 is assigned to another game service representative, the transaction history stored on the PGD B24 may be reconciled with a separate transaction history stored on a transaction server such as the EZ pay server B26.

[0927] The assigning and unassigning of the PGD B24 to a game service representative and the transaction reconciliation are performed for security and auditing purposes. Another security measure which may be used on the PGD B24 is a fixed connection time between the PGD B24 and a transaction server. For example, after the PGD B24 has been assigned to a game service representative and the game service representative has logged on the PGD B24, the PGD B24 may establish a connection with one or more transaction servers including the EZ pay server B26, a server B28, a server B30, or a server B32. The connection between a transaction server and the PGD B24 allows the PGD B24 to send information to the transaction server and receive information from the transaction server. The length of this connection may be fixed such that after a certain amount of time the connection between the PGD B24 and the transaction server is automatically terminated. To reconnect to the transaction server, the login and registration process must be repeated on the PGD B24.

[0928] A transaction server may provide one or more game service transactions. However, the PGD B24 may connect with multiple transaction servers to obtain different game service transactions. For example, server B30 may be a prize transaction server allowing prize service transactions and server B415 may be a food transaction server allowing food service transactions. When a game service representative receives a prize service request from a game player, the PGD B24 may be used to contact the prize transaction server B30 using a wire-less communication link between the PGD B24 and a transceiver B64 connected to the prize transaction server B30. Similarly, when a game service representative receives a food service request from a game player, the PGD B24 may be used to contact the food transaction server B32 using a wire-less communication link between the PGD B24 and a transceiver B66 connected to the food transaction server B32.

[0929] The different transaction servers including the servers B26, B28, B30, B32 may be on separate networks or linked in some manner. For example, server B32 is connected to network B74, server B26 is connected to network B38, server B30 is connected to network B76, and server B28 is connected to network B78. In this embodiment, a network link B80 exists between network B76 and network B38. Thus, server B26 may communicate with server B30 via the network link B80. A communication link between different servers may allow the servers to share game service transaction information and allow different communication paths between the PGDs and the transaction servers. Likewise, a network link B82 exists between network B78 and network B38, permitting the game server to communicate with the EZ pay server B26.

[0930] FIG. 12 is a flow chart depicting a method for providing a game service using a hand-held device. In step B500, a game service representative receives the PGD B24 and logs in to the device to assign the device. The check out process and assign process are for security and auditing purposes. In a step B505, the game service representative contacts a game player in the game playing area requesting a game service of some type. In a step B510, the game service representative selects an appropriate interface on the PGD B24 using menus on the display screen B102 of the PGD that allow the game service representative to provide a requested game service. In a step B515, the game service representative inputs game service transaction information required to perform a game service transaction. For example, to validate an award ticket, the game service representative may read information from the ticket using a ticket reader. As another example, to provide a food service including dinner reservation, the game service representative may enter a game player's name to make the reservation.

[0931] In a step B520, the transaction information obtained in step B515 is validated as required. For example, when a player attempts to cash out an award ticket, the information from the award is validated to ensure the ticket is both genuine (e.g. the ticket may be counterfeit) and has not already been validated. The validation process requires a number of transfers of information packets between the PGD B24 and the transaction server. The details of the validation process for an award ticket validation are described with reference to FIG. 13. When the transaction information is valid, in a step B522, a game service transaction is provided. For example, a room reservation may be made for a player requesting an accommodation service. A confirmation of the game service transaction may be sent to the transaction server for transaction reconciliation in a step B545. In one or more embodiments, the method may include the step of generating a receipt regarding the game service transaction.

[0932] In a step B535, after providing the service, a game player may request another game service. When a game player requests an additional game service, the game service representative returns to step B510 and selects an appropriate interface for the game service. When a game player does not request an additional service and it is not the end of a shift, in a step B530, the game service representative returns to step B505 and contacts a new game player. In a step B540, when a shift has ended, the game service representative logs out of the PGD B24 and checks the device at a secure location so that the PGD may be assigned to a different game service representative. In step B545, before the PGD B24 is assigned to a different game service representative, a transaction history reconciliation is performed to ensure that the transaction history stored on the PGD is consistent with the transactions previously confirmed with a transaction server during the game service representative's shift. The transaction history on the PGD B24 may be stored on a removable memory storage device on the PGD. Thus, the memory may be removed from the device for transaction reconciliation and replaced with a new memory. Thus, the device with the new memory may be assigned to a new game service representative while the transaction history from the previous game service representative assigned to the device is reconciled.

[0933] FIG. 13 is a flow chart depicting a method for validating information for providing a personal game service. In the embodiment shown in the figure, a ticket is validated in a manner consistent with an EZ pay ticket system. The EZ pay ticket is usually used for award tickets. However, the system may be adapted to provide tickets for other services include food services, prize services or accommodation services. In a step B600, a request for game service transaction information read from a ticket is sent via a wire-less communication interface on the PGD B24 to the appropriate transaction server as described with reference to FIG. 8. In a step B605, the server identifies which clerk validation ticket (CVT) B34,B36 owns the ticket. When a CVT owns a ticket, the CVT has stored information regarding the status of a particular ticket issued from a gaming machine connected to the CVT B34,B36. In a step B610, the server sends a request to pay the ticket to the CVT identified as the owner of the ticket. Typically, the pay request indicated a service on the ticket has been requested. For a cash ticket, a pay request means a request to cash out the ticket has been made. For a free meal, a pay request means a request to obtain the meal has been made. In a step B615, the CVT receives the pay request for the ticket and marks the ticket pending. While the ticket is pending, any attempts to validate a ticket with similar information is blocked by the CVT.

[0934] In a step B620, the CVT B34,B36 sends back a reply with context information to the server. As an example, the context information may be the time and place when the ticket was issued. The information from the CVT to the server may be sent as one or more data packets according to a communication standard shared by the CVT and server. In a step B625, after receiving the validation reply from the CVT, the server marks the pay request pending and sends a pay order to the PGD B24. While the pay request is pending, the server will not allow another ticket with the same information as the ticket with the pay request pending to be validated.

[0935] In a step B630, the game service representative may choose to accept or reject the pay order form the server. When the game service representative accepts the pay order from the server, in a step B640, the PGD B24 sends a reply to the transaction server confirming that the transaction has been performed. The transaction server marks the request paid which prevents another ticket with identical information from being validated. In a step B645, the server sends a confirmation to the CVT which allows the CVT to mark the request from pending to paid. When the game service representative rejects the pay order from the server, in a step B650, the PGD B24 sends a reply to the server to mark the pay request from pending to unpaid. When the ticket is marked unpaid, it may be validated by another PGD B24 or other validation device. In a step B655, the server sends the reply to the CVT to mark the pay request from pending to unpaid which allows the ticket to be validated.

[0936] In one or more embodiments of the invention, a ticket may be used to provide credit/value for establishing entitlement to a service or a good, such as the right to play a game or obtain food. The PGD B24 may include a card reader B140. In such an arrangement, a user of the PGD B24 may use a credit card or other magnetic stripe type card for providing creditvalue. In various embodiments, the PGD B24 may include one or more other types of devices for obtaining/receiving information, such as a smart card reader. In such arrangements, the PGD B24 device may read information from the credit card, smart card or other device. These cards may comprise the well known credit or debit cards. This information may be used to provide the credit/value. In the example of a credit card, the user's account information may be read from the card and transmitted from the PGD B24 to the controller B42. Credit card/credit validation information may be associated with a credit card server (not shown). This credit card server may be associated with a bank or other entity remote from the casino or place of use of the PGD B24 and the controller B42. A communication link may be provided between the controller B42 and remote server for sending credit card information there over.

[0937] In some embodiments, when a player utilizes a smart card or credit card the amount of associated credit or value may be transmitted to the EZ Pay server B26, and then the credited amount may be treated in exactly the same manner as if the credit/value had been provided by a ticket. When a player wishes to cash out, the EZ Pay server B26 has a record of the original amount credited and the amounts of any awards, losses or payments, and may then issue the player a ticket representing the user's total credit.

[0938] In accordance with the invention, a gaming system is provided which includes one or more portable gaming devices. The portable gaming devices permit a player to play one or more games at a variety of locations, such as a hotel room, restaurant or other location. These locations may be remote from traditional gaming areas where free-standing, generally stationary gaming machines are located.

[0939] In one or more embodiments, a player may use the portable gaming device to not only play games, but obtain other products and services. In addition, in one or more embodiments, the portable gaming device may be used by game service representatives to perform a variety of functions and provide a variety of services to a player.

[0940] It should be understood that the foregoing descriptions encompass but some of the implementation technologies that may be used, according to various embodiments. Other technologies may be used and are contemplated, according to various embodiments. Various embodiments may be performed using any suitable technology, either a technology currently existing or a technology which has yet to be developed.

Wireless Interactive System

[0941] According to various embodiments, a wireless interactive gaming system includes one or more wireless gaming devices, a receiver, and a central processor. The wireless interactive gaming system may also include a terminal which is in communication with the central processor.

[0942] In a gaming environment that employs a wireless interactive gaming system, a player receives a wireless gaming device from a game official who represents a gaming establishment or the "house". The wireless gaming device is capable of receiving wager information as commands entered by the player and transmitting the received wager information along with identification information to the receiver by wireless transmission.

[0943] The wireless interactive gaming system may support a number of wireless gaming devices within one gaming establishment. The range for the wireless transmission from a wireless gaming device may be up to 100 feet.

[0944] According to various embodiments, a player inputs information into a wireless gaming device, e.g., by pressing push buttons or keys on the device. The wireless gaming device may include any number, e.g. from 5 to 20, of buttons in a keypad-type arrangement. Buttons may be marked with the digits 0 through 9 and may also include a "$" (dollar sign) key and an "enter" key, so that the player may easily input wager information. In various embodiments, the wireless gaming device includes at least eight player selection buttons (e.g., digits) and at least five special function buttons, (e.g., to request the player's balance). In various embodiments, the player can input some or all of the wager information into the wireless gaming device by swiping a smart card, which contains a microprocessor chip or a magnetic stripe with encoded information, through a smart card reader on the wireless gaming device.

[0945] In various embodiments, the wireless gaming device may include an identifier. The identifier may be, e.g., a series of alphanumeric characters, a bar code, or a magnetic stripe affixed to the device. In various embodiments, the identifier may be a digital code stored in a secure memory, e.g., an electronically erasable programmable read only memory (EEPROM). The identifier may thus be readable directly by the game official if it is a series of alphanumeric characters, or it may be read automatically by a bar code reader or a magnetic stripe reader. In various embodiments, the identifier may be programmed in EEPROM or read from EEPROM through an RS-232 port, which may be directly connected to encoder and decoder circuitry in a terminal.

[0946] A wireless gaming device may store an encryption key. The encryption key may be used to encrypt information that is transmitted to the receiver from the device. Encryption of the information transmitted to the receiver may limits tampering with the wireless gaming device and may prevent unauthorized or counterfeit devices from being used with the system.

[0947] In various embodiments, the encryption key may be stored in the EEPROM. The EEPROM may have the advantage of being a memory device which is difficult to access if the appropriate encoding circuitry is not available. Thus, it is contemplated that the encoding circuitry that downloads the encryption key into the device may be securely held by the game official.

[0948] Alternately, the encryption key stored in the EEPROM may be updated and changed for each player who receives a wireless gaming device by directly connecting the device to encoding and decoding circuitry in the terminal through a port at the time the wireless gaming device is delivered to the player. Moreover, other digital information related to the game being played may be downloaded from the terminal to the EEPROM through a direct connection with the wireless gaming device.

[0949] In various embodiments, a microprocessor controls the operation of a wireless gaming device. The microprocessor receives digital wager information entered by the player using buttons or keys of the wireless gaming device. The microprocessor stores an identification code associated with the wireless gaming device that is a digital equivalent of the identifier of the wireless gaming device. The microprocessor also executes software applications for encrypting the identification code and the player's wager information for transmission to the receiver. The software contains an algorithm that encrypts a data packet including the identification code and wager information using the encryption key.

[0950] In various embodiments, a wireless gaming device has a unique address, i.e. identification code, for communications with the receiver and stores a player identification that is programmed into the device by the central processor. The wireless gaming device may include a wager amount register, which is maintained and updated using the keys on the device. The value stored in the wager amount register may be included in transmissions from the device to the central processor. The value of the wager amount register may default to a predetermined value, e.g. $1, when the device is initialized, and can be further adjusted by the player. The wireless gaming device may also include an account balance register, which is maintained in the device and is updated by the central processor periodically. The value of the account balance register should default to $0 when the device is initialized.

[0951] The wireless gaming device may include player function keys. The player function keys may be used to accomplish the following functions:
1. Transmit a message to the receiver;
2. Request account balance information;
3. Adjust the state of the device;
4. Affect the data to be sent in the next transmitted message;
5. Increment the wager amount register by a predetermined amount, e.g., $10, $5 or $1;
6. Reset the wager amount register to the default value, e.g., $1.

[0958] The firmware of the wireless gaming device may only allow for one press of buttons or keys every 100 ms. In various embodiments, key presses are not queued; thus, when a key press message is queued to be sent, no other player input is accepted until the queued message has been sent.

[0959] The wireless gaming device may include a transmitter. The transmitter may receive encrypted digital information from the microprocessor and convert it to a signal for wireless transmission to the receiver. The transmitter transmits signals wirelessly, e.g., using radio frequency signals or infrared signals. Communications between the receiver and the wireless gaming device may be asynchronous at 2400 bits per second.

[0960] The wireless gaming device may include an identifying circuit that drives the transmitter to periodically send an identification signal to the receiver. The use of the identifying circuit permits the receiver and the central processor to be assured that the wireless gaming device is still active, functioning and present in the gaming establishment. Thus, if the wireless gaming device were removed from the gaming establishment, the receiver and central processor would no longer receive and detect the periodic identification signal sent by the identifying circuit and the transmitter, and the game official may be alerted that the wireless gaming device has been removed from the gaming establishment.

[0961] The wireless gaming device may contain a real-time clock that permits the microprocessor to monitor the current time and date. The clock may consist of a timing circuit. The microprocessor can use the time and date information obtained from clock to perform calculations and other functions based on the current time and date.

[0962] The wireless gaming device may also include a tag, such as an electronic or magnetic component, which activates an alarm when passed through a sensing apparatus located at the entrance and/or exit of the gaming establishment. Activation of the alarm by passing the wireless gaming device with the tag through the sensing apparatus notifies the game official of an attempted removal of the wireless gaming device from the gaming establishment.

[0963] The wireless gaming device may be powered by a battery source contained within the device. A portable power source such as battery source permits extended cordless operation of the wireless gaming device throughout a gaming environment. The battery source may be part of a removable, rechargeable battery pack that allows the device to be recharged when it is not in use.

[0964] In some embodiments, the wireless gaming device displays information such as game information on a device display, such as a liquid crystal display (LCD) with a back-light. The LCD can be used to display the values stored in the wager amount register and in the account balance register. The wireless gaming device may include a display receiver which receives digital information transmitted from the receiver or from the central processor.

[0965] The device may also include a bicolor light emitting diode (LED). The bicolor LED is capable of displaying at least two colors, e.g., red and green.

The green light may flash each time the wireless gaming device sends a transmission to the receiver, for a period of time to ensure that it is visible to the player. The red light may illuminate when a key is pressed on the wireless gaming device, and remain lit until the transmission is received by the receiver; no additional key entry will be enabled when the red light is lit. The wireless gaming device may also include additional light emitting diodes, for example to indicate when the account balance register is being updated and the balance information is being displayed on the LCD.

[0966] The receiver is capable of receiving signals transmitted from the transmitter in the wireless gaming device. The receiver contains a decoder, which converts the received signals, e.g., into digital information. This digital information contains at least the identification code of the wireless gaming device and the player's wager information. The receiver sends the digital information obtained by the decoder to the central processor. Communications between the central processor and the receiver may be by an RS-232 electrical interface data serial communications link, with communications being asynchronous at either 9600 or 19,200 bytes per second, in various embodiments.

[0967] The receiver may receive signals from many wireless devices either simultaneously or in rapid succession, e.g., using multiplexing techniques, so that many players can place wagers using their wireless gaming devices during a short time interval. The receiver differentiates signals received from the various devices by the identification codes which are present in the signals received by the receiver.

[0968] The central processor receives the identification code of a wireless gaming device and the player's wager information from the receiver. The central processor also decrypts this information using the encryption key. The central processor is capable of receiving data from multiple wireless gaming devices in an apparently simultaneous manner.

[0969] In various embodiments, an account for the player is stored in a database of the central processor. The database stores the monetary value of the balance of the account associated with the identifier of the wireless gaming device.

[0970] The central processor manages the player's account in the database based on signals received from the player's wireless gaming device as the player places wagers and when prizes are awarded during play of the game. The central processor subtracts money from the player's account balance when the player places a wager. The player's account balance may be automatically increased by the central processor when the player wins a game on which he has placed a wager.

[0971] The central processor also stores and is capable of executing software applications containing algorithms to calculate players' account balances, wagers, and winnings. The central processor should be able to execute all of the algorithms which define the actions performed on the players' accounts during the progress of the game, as wagers are entered, as winnings paid out, and when funds are added to the players' accounts.

[0972] Algorithms in the software in the central processor may also calculate odds and payouts for certain games, such as lottery-type games, during play of the game. The odds and payouts at a particular point in time may depend on the characteristics of the game being conducted by the central processor, and may change as the game progresses. These algorithms may be executed by the central processor to provide exact calculations of the odds of specific game events occurring and the associated prizes for a player's correctly predicting the occurrence of one of those events. The algorithms may be executed continuously, so that real-time odds and payout can be calculated as the game progresses.

[0973] The central processor may perform various actions on players' accounts, resulting in various impacts on the accounts. For example, if the player wins a game, his account is credited for the payout based on his wager. If the player places a wager using the wireless gaming device, his account is debited by the amount of the wager. If the game official receives additional funds from the player, the balance of the player's account is credited by the amount of the funds. If the game official closes the player's account and disburses funds to him, the balance of the player's account is debited by the amount disbursed.

[0974] The central processor may be located in the gaming establishment that houses the receiver. In various embodiments, the central processor may be located remotely from the receiver, communicating with the receiver via electronic digital telephone communication or wireless transmission, such as a serial communication link. Additionally, the central processor may perform a multitude of functions for various receivers in a variety of gaming environments.

[0975] In some embodiments, communication among the central processor, the receiver, and the wireless gaming device involves a polling scheme. Polling enables many wireless gaming devices to communicate with a receiver without interference between them. Such a polling scheme may include the transmission of digital signals in the form of strings of hexadecimal characters. Preferably, all communications between the central processor, the receiver and the wireless gaming device are encrypted.

[0976] In such a polling scheme, hexadecimal characters may be reserved for specific control protocols. For example, an attention character is a header character used to begin all transmissions from the central processor to the receiver, and serves to delineate messages and synchronize the receipt of messages in the receiver. The same function is implied when the attention character follows in response to a message transmission. An acknowledgement character is another header character which provides acknowledgement to the transmitting device that the previous message's data has been received and verified. The acknowledgement character can also function as an attention character to begin a subsequent message. An end of message character is used to indicate the end of a transmission. Also, a complement next byte character allows for use of reserved protocol characters within a normal transmission message by avoiding a false control signal when a message data byte matches one of the control characters. When a message byte that needs to be sent matches one of the protocol control characters, the complement next byte character is sent, followed by the one's complement of the matching message byte.

[0977] Verification of received data may be accomplished using a single byte checksum of the message information. This checksum may be the one's complement of the sum of the original message data, not including the header character. If the checksum results in a value equal to one of the protocol control characters, it will be treated in accordance with the function of the complement next byte character.

[0978] In the polling scheme described above, there are three different modes of communication over the link between the central processor and the receiver. First, the central processor may send messages intended for the receiver. Second, the central processor may send messages intended for the wireless gaming device. Third, the wireless gaming device may send messages intended for the central processor. In various embodiments, messages sent by the central processor may be in the form of a character string formatted with a header character, followed by the identification code of the intended device, the command or message, an end of message character, and a checksum character. Messages received by the receiver or the wireless gaming device may be acknowledged by transmission of an acknowledgement character, but the central processor need not acknowledge messages sent from the wireless gaming devices. Messages sent by the central processor to be received by the wireless gaming device may be broadcast to all of the wireless gaming devices. A device address may be reserved as a broadcast address for all of the wireless gaming devices, and all devices will receive messages sent to this address; in this case, no acknowledgement need be returned from any of the wireless gaming devices.

[0979] Each command or message may begin with a command code to signal how the information contained in the message is to be used. Command codes for messages sent by the central processor to the receiver and the wireless gaming device include the following:
1. Send a device address list to the receiver;
2. Send account balance information to the addressed device;
3. Send command to disable the addressed device;
4. Send command to enable the addressed device.

[0984] In various embodiments, messages sent between the receiver and the wireless gaming device may be in the form of a character string formatted with a header character, followed by the identification code of the intended device, the current wager amount, the request, command or data, an end of message character, and a checksum character. Command codes for requests, commands and data sent between the receiver and the wireless gaming device include the following:
1. Read user identification;
2. Read device address;
3. Read balance register;
4. Read wager amount register;
5. Provide device status;
6. Write user identification;
7. Write device address;
8. Write balance register;
9. Write wager amount;
10. Perform self test.

[0995] These command codes may be used to program the device addresses and user identification information into the wireless gaming devices, as well as to initialize the device to the default state, i.e., the player's account balance of $0. The account balance register and the user identification may each comprise two characters, the least significant byte and the most significant byte, allowing for the use a greater range of numbers for these values.

[0996] Various embodiments include methods by which the central processor communicates with a wireless gaming device. The central processor transmits a string of hexadecimal characters, including, e.g., a header character, followed by the device's identification code, followed by a request, command or data, followed by an end of message character, followed by a checksum character. After the central processor transmits the character string, the wireless gaming device receives the string, recognizes its identification code, and executes any instructions in the string. When the central processor sends an instruction to all wireless gaming devices simultaneously, all currently active devices receive and execute the instruction. The wireless gaming device does not send an acknowledgement message to the central processor, although the receiver may receive a transmission from the wireless gaming device that the instruction was received properly. The central processor also communicates with the receiver in a similar manner, except that the receiver may send an acknowledgement message to the central processor which includes the acknowledgement control protocol character.

[0997] Similarly, the wireless gaming device communicates with the receiver and the central processor using, e.g., hexadecimal character strings. The receiver regularly and periodically polls the active wireless gaming device for information requests or wagering requests. If the player has entered a request into the wireless gaming device since the last time the wireless gaming device was polled, then the player's request will be transmitted to the receiver.

[0998] Various embodiments include methods by which the wireless gaming device receives and relays player requests to the central processor. First, the player enters a request into the wireless gaming device using buttons or keys. The player then presses a button labeled, e.g., "enter" or "send," instructing the wireless gaming device to send the request the next time the receiver polls the wireless gaming device. When this button has been pressed, the red light of the bicolor LED is illuminated, thereby informing the player that the request is waiting to be sent. The request is converted into a hexadecimal character string, including, e.g., a header character, an identification code (or, alternatively, a separate identification string reserved for a specific player), the current wager amount, the player's request (e.g., to change the wager amount or to send a balance update), an end of message character, and a checksum character. The next time the receiver polls the device, the transmitter of the device transmits the character string to the receiver. When the wireless gaming device is polled by the receiver, the green light of bicolor LED flashes, informing the player that the request has been transmitted. The receiver receives the request string, and transmits the string to the central processor. The central processor then acts on the player's request.

[0999] Using the terminal, the game official may process wagering transactions and distribute wireless gaming devices. In various embodiments, the terminal may include a bar code reader and/or a magnetic stripe reader for rapid entry of the identifier of a wireless gaming device prior to delivering the wireless gaming device to the player. Reading devices provide information in the form of digital data to the terminal. The terminal includes a keyboard by which the game official can manually enter data to be sent to the central processor. Using either reading device, the keyboard, or a combination of these, the game official communicates with the central processor to establish a player's account, increase the balance of the account when the player tenders funds to the game official, and decrease the balance of the account when the player seeks to collect the cash value of his account balance.

[1000] The player establishes a balance of the account associated with his wireless gaming device, identified by an identifier, when he receives the wireless gaming device from the game official. The player may increase the monetary value of the balance of the account by paying additional funds, in the form of cash or credit, to the game official, who accesses the account stored in the central processor through the terminal to increase the balance of the account.

[1001] The wireless gaming device is returned to the game official after the player has played one or more games. The readers may be used to read the identifier for closing out the player's account stored in the database of the central processor. The terminal includes a terminal display which notifies the game official of the balance of the player's account, so that the player may be paid the cash value of the remaining balance of his account.

[1002] In some embodiments, an account status display device is located in the gaming establishment to display players' account information. In various embodiments, the display device may be, e.g., a liquid crystal display or a cathode ray tube display. The display device is controlled by the central processor, which sends information to the display device for display to the players.

[1003] A player may look at the display device to confirm that wagers transmitted from the wireless gaming device were received by the receiver and sent to the central processor, to determine the monetary balance of the player's account, and to verify that the player's winnings have been credited to his account. The display device displays key information necessary for a player to participate in a game. The information displayed for each player may include the account number, the player's account balance, the player's last wager, and the player's last prize award or win.

[1004] The display device is divided into specific areas, e.g., a display area, each area showing the account information for one player. The size of the display area may be determined by the size of the display device and the number of players who possess wireless display devices. It is contemplated that only active accounts will be displayed on the display device. If additional display devices are required to display the information concerning a large number of accounts, the central processor may be configured to drive multiple similar display devices.

[1005] The display device may also be used to display the odds and payouts for game wagers. Alternately, a separate display device driven by the central processor may be used to display the odds and payout information. Further, the odds and payouts may be displayed on the device display 21.

[1006] Procedures for using the wireless interactive gaming system, according to some embodiments, are now described. In some embodiments, a player tenders money in the form of cash or credit, e.g., $100, to a game official in the gaming establishment to establish an account. The game official chooses a wireless gaming device and uses, e.g., the bar code reader on the terminal to enter the identifier of the wireless gaming device into the terminal. The game official also inputs the amount of money tendered, i.e. $100, into the terminal via keyboard. The game official hands the wireless gaming device to the player and tells the player that his account is, e.g., Account No. 12.

Alternately, the player may identify his account number directly from the identifier on the wireless gaming device. The information entered by the game official into the terminal is sent to the central processor, which establishes an account record for the player in the database.

[1007] For this example, the central processor may be conducting a racing game in which players choose a winning racing element on which to place a wager for the next racing game to be displayed in the gaming establishment. To place a wager, the player presses buttons on the wireless gaming device.

[1008] In some embodiments, the player first presses the button that corresponds to the number assigned to the racing element that he chooses, e.g., "3", and then the wager amount, e.g., "$" and "5", for a $5 wager. The player then presses the "enter" key to transmit his wager to the central processor.

[1009] In an alternate embodiment, the game may be simplified so that all wagers are placed for a fixed amount, e.g., $1, by pressing a single button on the wireless gaming device. By pressing the button that corresponds to the number assigned to the chosen racing element, e.g., "3", the player places a $1 bet on racing element number 3. The player can then place a larger wager on racing element number 3, by pressing the "3" button the number of times corresponding to the number of $1 bets he desires to make, e.g., by pressing "3" five times to wager $5 on racing element number 3.

[1010] Each time the player enters a wager, the wireless gaming device forms a data packet containing the player's wager information and the identification code of the wireless gaming device. The data packet is encrypted and transmitted by the transmitter via wireless communication.

[1011] The decoder in the receiver receives the encrypted data packet transmitted by the transmitter. The encrypted data packet is sent to the central processor, where it is decrypted. The central processor uses the information it has obtained to update the player's account in the database by subtracting the wagered amount from the player's account balance and registers the player's wager on the game.

[1012] After the game has been played, the central processor awards prizes to winning players based on the wagers they have made and the odds associated with the winning outcome of the game. If the player in possession of the wireless gaming device is a winner, the central computer updates the player's account in the database by adding the monetary amount of the prize to the player's account balance. Otherwise, the player's account remains unchanged.

[1013] When the player has finished playing games in the gaming establishment, he returns the wireless gaming device to the game official. The game official again inputs the identifier of the wireless gaming device into the terminal, e.g., by using the bar code reader of the terminal. The terminal accesses the player's account information stored in the database of the central processor to obtain the player's remaining account balance. The terminal display displays the player's remaining account balance to the game official, who then tenders the monetary value of that amount to the player. The account is closed, and the transaction is recorded in the central processor.

[1014] It should be understood that the foregoing descriptions encompass but some of the implementation technologies that may be used, according to various embodiments. Other technologies may be used and are contemplated, according to various embodiments. Various embodiments may be performed using any suitable technology, either a technology currently existing or a technology which has yet to be developed.

Hand-Held Wireless Game Player

[1015] Various embodiments include a hand-held wireless game player for playing a game of chance. The hand-held wireless game player may be generally characterized as including: 1) a wire-less communication interface; 2) a display screen; 3) one or more input mechanisms; and 4) a microprocessor configured i) to present the game of chance on the display screen using operating instructions received via the wireless communication interface from a master gaming controller located on a gaming machine and ii) to send information from input signals generated from the one or more input mechanisms to the master gaming controller via the wire-less communication interface. The wireless game player may be played in a plurality of venue locations physically separate from the location of the gaming machine where the plurality of venue locations are selected from the group consisting of a keno parlor, a bingo parlor, a restaurant, a sports book, a bar, a hotel, a pool area and a casino floor area. The game of chance played on the wireless game player may be selected from the group consisting of slot games, poker, pachinko, multiple hand poker games, pai-gow poker, black jack, keno, bingo, roulette, craps and a card game. Other games are also contemplated, in various embodiments.

[1016] In various embodiments, the wireless communication interface may use a wireless communication protocol selected from the group consisting of IEEE 802.11a, IEEE 802.11b, IEEE 802.11x, hyperlan/2, Bluetooth, and HomeRF. The wireless game player may also comprise a wire network interface for connecting the wireless game player to a wire network access point. In addition, the wireless game player may also comprise a peripheral interface for connecting to a peripheral gaming device where the peripheral interface is a serial interface, a parallel interface, a USB interface, a FireWire interface, an IEEE 1394 interface. The peripheral gaming device may be a printer, a card reader, a hard drive and a CD-DVD drive.

[1017] In various embodiments, the one or more inputs mechanisms on the wireless game player may be selected from the group consisting of a touch screen, an input switch, an input button and biometric input device where the biometric input device may be a finger print reader. The wireless game player may also include a detachable memory interface designed to receive a detachable memory where the detachable memory unit stores graphical programs for one or more games of chance played on the wireless game player. The wireless game player may also comprise one or more of the following: 1) an audio output interface for receiving a head phone jack, 2) an antenna, 3) a sound projection device, 4) a battery, 5) a power interface for supplying power to the wireless game player from an external power source and for charging the battery from the external power source, 6) a memory unit where the memory unit may store graphical programs for one or more games of chance played on the wireless game player, 7) an electronic key interface designed to receive an electronic key, and 8) a video graphics card for rendering images on the display screen where the video graphics card may be used to render 2-D graphics and 3-D graphics.

[1018] It should be understood that the foregoing descriptions encompass but some of the implementation technologies that may be used, according to various embodiments. Other technologies may be used and are contemplated, according to various embodiments. Various embodiments may be performed using any suitable technology, either a technology currently existing or a technology which has yet to be developed.

Espacenet

Claims: JP2010528716 (A) — 2010-08-26

GAME WITH HAND MOTION CONTROL

Claims not available for JP2010528716 (A)
Claims of corresponding document: US2008300055 (A1)

A high quality text as facsimile in your desired language may be available amongst the following family members:

US2008300055 (A1) WO2008150809 (A1) US2018018014 (A1) US2020033938 (A1) US2022107682 (A1)

- Original claims
- Claims tree

What is claimed is:

1. A method comprising:
receiving a first wireless signal from a first device;
receiving a second wireless signal from a second device;
determining from the first wireless signal a first player identifier;
determining from the second wireless signal a second player identifier;
displaying a message that asks a player to identify himself;
receiving via tactile input an indication of a third player identifier;
determining that the third player identifier matches the first player identifier;
receiving a third wireless signal from the first device;
interpreting the third wireless signal as a command in a gambling game; and
carrying out the command in the gambling game.

2. The method of 1 in which the first device is one of: (a) a wristband; (b) a watch; (c) a bracelet; (d) an armband; and (e) a mobile gaming device.

3. The method of 1 in which determining from the first wireless signal a first player identifier includes determining from the first wireless signal a name of a first player.

4. The method of 1 in which the first player identifier and the second player identifier correspond to different players.

5. The method of 1 in which receiving via tactile input an indication of a third player identifier includes receiving an indication of a third player identifier, in which the third

Petitioner Exhibit 1002-0676

player identifier has been inputted using buttons.

6. The method of 1 in which receiving via tactile input an indication of a third player identifier includes receiving an indication of a third player identifier, in which the third player identifier has been inputted using a joystick.

7. The method of 1 in which receiving via tactile input an indication of a third player identifier includes receiving an indication of a third player identifier, in which the third player identifier has been inputted using a touch screen.

8. The method of 1 in which receiving via tactile input an indication of a third player identifier includes receiving an indication of a third player identifier, in which the third player identifier has been inputted using a track ball.

9. The method of 1 in which the third wireless signal encodes a set of motions made by the first device.

10. The method of 1 in which interpreting the third wireless signal includes interpreting the third wireless signal as a command to discard a card in a game of video poker.

11. The method of 1 in which interpreting the third wireless signal includes interpreting the third wireless signal as a command to initiate a slot machine game.

12. An apparatus comprising:
a band formed into a loop;
a power source attached to the band;
a motion sensor attached to the band;
an electromagnetic transmitter attached to the band;
an audio speaker attached to the band;
a haptics transducer attached to the band;
a processor attached to the band; and
an electromagnetic receiver attached to the band.

13. The apparatus of 12 in which the haptics transducer is operable to generate vibrations in response to an electric signal from the processor.

14. The apparatus of 12 in which the motion sensor is an accelerometer.

15. The apparatus of 12 in which the processor is operable to:
receive a first electronic signal from the motion sensor;
determine a first command for a first gambling game based on the first electronic signal;
transmit the first command to the electromagnetic transmitter; and
direct the electromagnetic transmitter to transmit the first command to a first gaming device.

16. The apparatus of 12 in which the processor is operable to:
receive from the electromagnetic receiver instructions that have been received wirelessly by the electromagnetic receiver;
receive a second electronic signal from the motion sensor;
follow the instructions in order to determine a second command for a second gambling game based on the second electronic signal;
transmit the second command to the electromagnetic transmitter; and
direct the electromagnetic transmitter to transmit the second command to the gaming

device.

17. The apparatus of 12 further including a switch attached to the band, in which the switch has two stable positions, and in which the processor is operable to detect the position of the switch and to direct the electromagnetic transmitter to transmit signals only if the switch is in a first of the two stable positions.

18. The apparatus of 12 further including a piezoelectric sensor attached to the band.

19. An apparatus comprising:
a housing, the housing including a top surface that is parallel to the ground;
a coin hopper disposed within the housing;
a bill validator attached to the housing;
a display screen attached to the housing;
a processor disposed within the housing;
a wireless receiver attached to the housing;
a wireless transmitter attached to the housing;
a first light source attached to the top surface of the housing, in which the first light source is operable to emit light of a first frequency; and
a second light source attached to the top surface of the housing at least one foot from the first light source, in which the second light source is configured to emit light of a second frequency which is different from the first frequency.

20. The apparatus of 19 in which the processor is operable to:
conduct gambling games; and
alter the course of a gambling game based on wireless signals received at the wireless receiver.

(19) 日本国特許庁 (JP)　　(12) 公 表 特 許 公 報 (A)　　(11) 特許出願公表番号

特表2010-528716
(P2010-528716A)

(43) 公表日　平成22年8月26日 (2010.8.26)

(51) Int.Cl.			F I			テーマコード（参考）
A63F	**13/00**	**(2006.01)**	Ａ６３Ｆ	13/00	F	２Ｃ００１
A63F	**13/06**	**(2006.01)**	Ａ６３Ｆ	13/06		

審査請求 未請求　予備審査請求 未請求　（全 195 頁）

(21) 出願番号	特願2010-510470 (P2010-510470)
(86) (22) 出願日	平成20年5月28日 (2008.5.28)
(85) 翻訳文提出日	平成22年1月27日 (2010.1.27)
(86) 国際出願番号	PCT/US2008/065000
(87) 国際公開番号	WO2008/150809
(87) 国際公開日	平成20年12月11日 (2008.12.11)
(31) 優先権主張番号	11/754,944
(32) 優先日	平成19年5月29日 (2007.5.29)
(33) 優先権主張国	米国 (US)

(71) 出願人　506361856
シーエフピーエイチ，　エル．エル．シー．
アメリカ合衆国　ニューヨーク州　１００２２　ニューヨーク　イースト・５９ス・ストリート　１１０

(74) 代理人　100070150
弁理士　伊東　忠彦

(74) 代理人　100091214
弁理士　大貫　進介

(74) 代理人　100107766
弁理士　伊東　忠重

最終頁に続く

(54)【発明の名称】手のモーションコントロールを有するゲーム

(57)【要約】
様々な実施形態において、リストバンドのモーションがゲームをコントロールするために用いられる。第１の装置から第１のワイヤレス信号を受信する工程と、第２の装置から第２のワイヤレス信号を受信する工程と、前記第１のワイヤレス信号から、第１のプレーヤ識別子を決定する工程と、前記第２のワイヤレス信号から、第２のプレーヤ識別子を決定する工程と、プレーヤに自身を同定させるように要求するメッセージを表示する工程と、触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程と、前記第３のプレーヤの識別子が前記第１のプレーヤの識別子と一致することを決定する工程と、前記第１の装置から第３のワイヤレス信号を受信する工程と、前記第３のワイヤレス信号を、ギャンブルゲームにおけるコマンドとして解釈する工程と、前記ギャンブルゲームにおける前記コマンドを実行する工程とを含む方法を提供する。
【選択図】なし

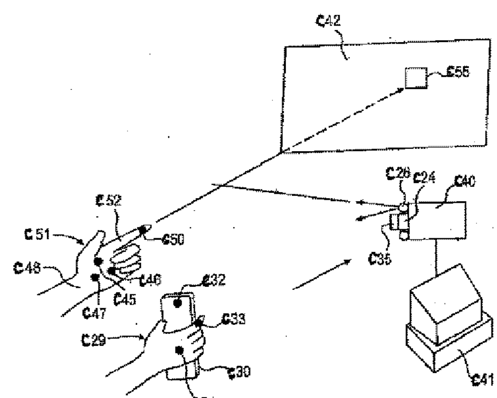


FIG. 14B

【特許請求の範囲】
【請求項1】
第１の装置から第１のワイヤレス信号を受信する工程と、
第２の装置から第２のワイヤレス信号を受信する工程と、
前記第１のワイヤレス信号から、第１のプレーヤ識別子を決定する工程と、
前記第２のワイヤレス信号から、第２のプレーヤ識別子を決定する工程と、
プレーヤに自身を同定させるように要求するメッセージを表示する工程と、
触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程と、
前記第３のプレーヤの識別子が前記第１のプレーヤの識別子と一致することを決定する工程と、
前記第１の装置から第３のワイヤレス信号を受信する工程と、
前記第３のワイヤレス信号を、ギャンブルゲームにおけるコマンドとして解釈する工程と、
前記ギャンブルゲームにおける前記コマンドを実行する工程と
を含む、方法。
【請求項2】
前記第１の装置は、（ａ）リストバンド、（ｂ）時計、（ｃ）ブレスレット、（ｄ）アームバンド、および（ｅ）携帯ゲーム装置のうちの１つである、請求項１に記載の方法。
【請求項3】
前記第１のワイヤレス信号から、第１のプレーヤの識別子を決定する工程が、前記第１のワイヤレス信号から、第１のプレーヤの名前を決定する工程を含む、請求項１に記載の方法。
【請求項4】
前記第１のプレーヤの識別子および前記第２のプレーヤの識別子が異なるプレーヤに対応する、請求項１に記載の方法。
【請求項5】
前記触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程が、第３のプレーヤの識別子の指示を受信する工程を含み、前記第３のプレーヤの識別子はボタンを用いて入力される、請求項１に記載の方法。
【請求項6】
前記触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程が、第３のプレーヤの識別子の指示を受信する工程を含み、前記第３のプレーヤの識別子はジョイスティックを用いて入力される、請求項１に記載の方法。
【請求項7】
前記触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程が、第３のプレーヤの識別子の指示を受信する工程を含み、前記第３のプレーヤの識別子はタッチスクリーンを用いて入力される、請求項１に記載の方法。
【請求項8】
前記触知性の入力部を介して、第３のプレーヤの識別子の指示を受信する工程が、第３のプレーヤの識別子の指示を受信する工程を含み、前記第３のプレーヤの識別子はトラックボールを用いて入力される、請求項１に記載の方法。
【請求項9】
前記第３のワイヤレス信号は、前記第１の装置によってなされる一連のモーションをエンコードする、請求項１に記載の方法。
【請求項10】
前記第３のワイヤレス信号を解釈する工程が、前記第３のワイヤレス信号を、ビデオポーカーのゲームにおいて、カードを捨てるコマンドとして解釈する工程を含む、請求項１に記載の方法。
【請求項11】

前記第３のワイヤレス信号を解釈する工程が、前記第３のワイヤレス信号を、スロットマシンゲームを開始するコマンドとして解釈する工程を含む、請求項１に記載の方法。

【請求項12】

ループに形成されるバンドと、
前記バンドに取り付けられる電源と、
前記バンドに取り付けられるモーションセンサと、
前記バンドに取り付けられる電磁送信器と、
前記バンドに取り付けられるオーディオスピーカと、
前記バンドに取り付けられる触覚変換器と、
前記バンドに取り付けられるプロセッサと、
前記バンドに取り付けられる電磁受信器と
を備える、装置。

【請求項13】

前記触覚変換器は、前記プロセッサからの電気信号に応答して振動を生成するように動作可能である、請求項１２に記載の装置。

【請求項14】

前記モーションセンサが加速度計である、請求項１２に記載の装置。

【請求項15】

前記プロセッサは、
前記モーションセンサから、第１の電気信号を受信し、
前記第１の電気信号に基づいて、第１のギャンブルゲームのための第１のコマンドを決定し、
前記第１のコマンドを前記電磁送信器に送信し、
前記電磁送信器に、前記第１のコマンドを第１のゲーム装置に送信するようにさせるように動作可能である、請求項１２に記載の装置。

【請求項16】

前記プロセッサは、
前記電磁受信器によってワイヤレスで受信された命令を、前記電磁受信器から受信し、
前記モーションセンサから、第２の電気信号を受信し、
前記第２の電気信号に基づいて、第２のギャンブルゲームのための第２のコマンドを決定するために、前記命令に従い、
前記第２のコマンドを前記電磁送信器に送信し、
前記電磁送信器に、前記第２のコマンドをゲーム装置に送信するようにさせるように動作可能である、請求項１２に記載の装置。

【請求項17】

前記バンドに取り付けられるスイッチをさらに含む、請求項１２に記載の装置であって、
前記スイッチは２つの安定した位置を有し、
前記プロセッサは、前記スイッチの位置を検出し、かつ、前記スイッチが前記２つの安定した位置のうちの第１の位置にある場合にのみ、前記電磁送信器に信号を送信するようにさせるように動作可能である、装置。

【請求項18】

前記バンドに取り付けられる圧電センサをさらに備える、請求項１２に記載の装置。

【請求項19】

地面と平行する上表面を含むハウジングと、
前記ハウジング内に配置されたコインホッパーと、
前記ハウジングに取り付けられた紙幣鑑別機と、
前記ハウジングに取り付けられたディスプレイスクリーンと、
前記ハウジング内に配置されたプロセッサと、
前記ハウジングに取り付けられたワイヤレス受信器と、

前記ハウジングに取り付けられたワイヤレス送信器と、
前記ハウジングの前記上表面に取り付けられ、第１の周波数の光を発するように動作可能である、第１の光源と、
前記ハウジングの前記上表面に取り付けられ、前記第１の光源から少なくとも１フィート離れ、前記第１の周波数とは異なる第２の周波数の光を発するように構成される、第２の光源と
を備える、装置。
【請求項20】
前記プロセッサは、
ギャンブルゲームを実行し、
前記ワイヤレス受信器において受信されたワイヤレス信号に基づいて、ギャンブルゲームの進行を変更するように動作可能である、請求項１９に記載の装置。
【発明の詳細な説明】
【技術分野】
【0001】
本出願は、「Ｇａｍｅ　Ｗｉｔｈ　Ｈａｎｄ　Ｍｏｔｉｏｎ　Ｃｏｎｔｒｏｌ」と題された、２００７年５月２９日に出願の米国特許出願第１１／７５４，９４４号の優先権の利益を主張する。上述の出願の全体は本明細書において参照することにより援用される。本出願は手のモーションコントロールを有するゲームに関する。
【背景技術】
【0002】
本出願は手のモーションコントロールを有するゲームに関する。
【図面の簡単な説明】
【0004】
【図1】図１は、一部の実施形態に係るゲームシステムを示す。
【図2】図２は、一部の実施形態に係る通信ネットワークを示す。
【図3】図３は、一部の実施形態に係るゲーム通信装置と通信するゲームサービス・プロバイダを示す。
【図4】図４は、一部の実施形態に係るゲームネットワークを示す。
【図5】図５は、一部の実施形態に係るゲームシステムを示す。
【図6】図６は、一部の実施形態に係るワイヤレスゲームシステムを示す。
【図7】図７は、一部の実施形態に係る宣伝コンテンツを有する携帯ゲーム装置を示す。
【図8】図８は、一部の実施形態に係るゲームシステムのブロック図である。
【図9】図９は、一部の実施形態に係る、図８に示されるゲームシステムの一部を形成する支払いシステムのブロック図である。
【図10】図１０は、一部の実施形態に係る、図８に示されるゲームシステムの携帯ゲーム装置の概略図である。
【図11a】図１１（ａ）は、一部の実施形態に係る、プレーヤによる携帯ゲーム装置の使用方法のフロー図である。
【図11b】図１１（ｂ）は、一部の実施形態に係る、プレーヤによる携帯ゲーム装置の特定の使用方法のフロー図である。
【図12】図１２は、一部の実施形態に係る、ゲームサービス・オペレータによる、携帯ゲーム装置の使用方法のフロー図である。
【図13】図１３は、一部の実施形態に係る、携帯ゲーム装置の使用方法のフロー図である。
【図14a】いくつかの一カメラベースの実施形態を示す。
【図14b】いくつかの３－Ｄ（３次元の）検出の実施形態を示す。
【図14c】２つのカメラ「双眼鏡」ステレオカメラを備えるいくつかの実施形態を示す。
【図14d】いくつかの実施形態に従ういくつかのステップを示す。
【図14e】いくつかの実施形態に従うカラーマッピングのためのプロセスを示す。
【図15】いくつかの実施形態に従うマルチカメラコントロールシステムの実装のハードウ

ェア構成要素、およびその物理的なレイアウトを示す。
【図16A】いくつかの実施形態に従う、カメラと図１５の様々な画像領域との間の幾何学的関係を示す。
【図16B】いくつかの実施形態に従う、図１５カメラのうちの１つに取り込まれた画像を示す。
【図17】いくつかの実施形態に従う、マルチカメラコントロールシステムと関連したマイクロコンピュータプログラム内で実行されるプロセスを示すフロー図である。
【図18】いくつかの実施形態に従う、図１７に示されたプロセスの一部分、特に、カメラによって取り込まれた画像信号から物体を検出してその位置を抽出することに関係するプロセスをより詳細に示すフロー図である。
【図19A】いくつかの実施形態に従う、カメラによって取得され、図１８に示されるプロセスの一部によって生成された、グレースケールビットマップ画像として提示されたサンプル画像データを示す。
【図19B】いくつかの実施形態に従う、図１８に示されるプロセスの一部によって生成されたグレースケールビットマップ画像として提示されたサンプル画像データを示す。
【図19C】いくつかの実施形態に従う、図１８に示されるプロセスの一部によって生成されたグレースケールビットマップ画像として提示されたサンプル画像データを示す。
【図19D】いくつかの実施形態に従う、図１８に示されるプロセスの一部によって生成されたグレースケールビットマップ画像として提示されたサンプル画像データを示す。
【図19E】いくつかの実施形態に従う、図１８に示されるプロセスの一部によって生成された、サンプル中の追跡されている物体に属する可能性が高い画素を識別するバイナリのビットマップ画像として提示されたサンプルデータを示す。
【図20】いくつかの実施形態に従う、図１８に記載されるプロセスの一部、特に追跡されている物体に属する可能性が高いとして識別された画素のマップが与えられたとき、例えば図１９Ｅに示されるデータが与えられたとき、その物体を分類して識別することに関係するプロセスをより詳細に示すフロー図である。
【図21A】いくつかの実施形態に従う、図２０に示されるプロセスがこのサンプル中で物体に属するとして選択したデータサンプルの識別とともに、図１９Ｅに提示される、バイナリのビットマップ画像として提示されたサンプルデータを示す。
【図21B】いくつかの実施形態に従う、図２０に概略が示されたプロセスが、物体に属するとして選択したデータサンプルの識別とともに、棒グラフとして提示された図１９Ｅに提示されるサンプルデータを示す。グラフの中の特定の点が識別されている。
【図21C】いくつかの実施形態に従う、物体およびこのサンプルにおけるその物体の鍵となる部分に属するとして図２０に示されるプロセスが選択したデータサンプルの識別とともに、バイナリのビットマップ画像として提示された異なるセットのサンプルデータを示す。
【図22】いくつかの実施形態に従う、図１８に示されるプロセスの一部、特に、物体が塞いでいるバックグラウンド領域の記載を発生および維持することに関係するプロセスをより詳細に示すフロー図である。
【図23A】いくつかの実施形態に従う、式３が基礎を置く配置、すなわち物体が検知された画像面上の位置が与えられたときに、カメラの視野内でのその物体の位置を画定する角度を示す。
【図23B】いくつかの実施形態に従う、式４、５および６が基礎を置く配置、すなわちカメラの位置と追跡されている物体との間の関係を示す。
【図24】いくつかの実施形態に従う、式８、すなわち位置を精緻化するために、物体の位置の変化が与えられたときに座標に加えられてもよい減衰量を示すグラフである。
【図25A】いくつかの実施形態に従う、注目する物体が画面ポインタを二次元でコントロールするシステムによってコントロールされるアプリケーションプログラムの一例である。
【図25B】いくつかの実施形態に従う、現実の座標と図２５Ａのアプリケーションプログ

ラムによって使用される画面座標との間のマッピングを示す。
【図26A】いくつかの実施形態に従う、注目する物体が三次元のバーチャル・リアリティ環境内で画面ポインタをコントロールする、マルチカメラコントロールシステムによってコントロールされるアプリケーションプログラムの一例である。
【図26B】いくつかの実施形態に従う、注目する物体が三次元のバーチャル・リアリティ環境内で画面ポインタをコントロールする、マルチカメラコントロールシステムによってコントロールされるアプリケーションプログラムの一例である。
【図27A】いくつかの実施形態に従う、起動しようとする意図と関連していてもよいジェスチャを識別するためのジェスチャ検出方法によって使用される検出面への、領域の分割を示す。
【図27B】カーソル方向を選択することに関連していてもよいジェスチャを識別するためのジェスチャ検出方法によって使用される検出ボックスへの、いくつかの実施形態に従う注目する領域の分割を示す。
【図27C】いくつかの実施形態に従う、カーソル方向を選択することと関連していてもよいジェスチャを識別するためのジェスチャ検出によって使用される方向検出ボックスへの、領域の代替的な分割を示す。
【図27D】いくつかの実施形態に従う、図２７Ｃの隣あう区画の関係をより詳細に示す。
【図28】装置がニュートラル位置にある状態の、いくつかの実施形態に従う装置の外観を示す。
【図29】いくつかの実施形態に従う、図２８の実装の内部構造の一例を示す。
【図30】いくつかの実施形態に従う、別の例示的な実装に係る方法を示す流れ図である。
【図31】図３１Ａ～図３１Ｄは、いくつかの実施形態に従う、ニュートラル軸の周りに画定される傾斜領域の例を示す。
【図32】いくつかの実施形態に従う、別の例示的な実装に係る例示的な装置の外側上面図を示す。
【図33】図３３Ａ～３３Ｅは、いくつかの実施形態に従う、例示的なインジケータを示す。
【図34】図３４Ａおよび３４Ｂは、いくつかの実施形態に従う、それぞれ、ニュートラル位置で示された図３２の装置の正面図および側面図を示す。
【図35】図３５Ａおよび３５Ｂは、いくつかの実施形態に従う、図３２の装置がそれぞれネガティブロール配向およびポジティブロール配向で操作される状態の図３２の装置の正面図を示す。
【図36】図３６Ａおよび３６Ｂは、いくつかの実施形態に従う、図３２の装置がそれぞれポジティブピッチ配向およびネガティブピッチ配向で操作される状態の図３２の装置の側面図を示す。
【図37】いくつかの実施形態に従う、コントロールが選択されたときの出力である文字およびケースに対応する信号を出力するために使用される装置配向の１つの可能なマッピングを示す表である。
【図38】図３８Ａおよび３８Ｂは、いくつかの実施形態に従う、別の例示的な実装に従って表示されるシンボルのメニューを示す。
【図39】いくつかの実施形態に従う、ゲームシステムＦ１を示す外観図である。
【図40】図３９に示されるゲーム装置Ｆ３の機能ブロック図である。
【図41】図３９に示されるコントローラＦ７の外観を示す斜視図である。
【図42】コアユニットＦ７０に接続しようとされている、またはコアユニットＦ７０から切断されようとしている図４１に示されるコントローラＦ７の接続ケーブルＦ７９の状態を示す斜視図である。
【図43】上方後ろ側から見た図４１に示されるコアユニットＦ７０の斜視図である。
【図44】底面の正面側から見た図４１に示されるコアユニットＦ７０の斜視図である。
【図45】図４１に示されるコアユニットＦ７０の上側のケーシングが取り外されている状態を示す斜視図である。

【図46】図４１に示されるコアユニットＦ７０の下側のケーシングが取り外されている状態を示す斜視図である。
【図47】図４１に示されるサブユニットＦ７６第１の例を示す斜視図である。
【図48】図４７に示されるサブユニットＦ７６の上側のケーシングが取り外されている状態の斜視図である。
【図49】図４９Ａ、４９Ｂ、および４９Ｃは、図４１に示されるサブユニットＦ７６の第２の例の、それぞれ上面図、底面図および左側面図である。
【図50】上方前側から見た図４１に示されるサブユニットＦ７６の斜視図である。
【図51】図４１に示されるサブユニットＦ７６の第１の変形の一例を示す上面図である。
【図52】図４１に示されるサブユニットＦ７６の第２の変形の一例を示す上面図である。
【図53】図４１に示されるサブユニットＦ７６の第３の変形の一例を示す上面図である。
【図54】図４１に示されるサブユニットＦ７６の第４の変形の一例を示す上面図である。
【図55】図４１に示されるコントローラＦ７の構造を示すブロック図である。
【図56】図４１に示されるコントローラＦ７を用いてコントロールされているゲームの状態を示す略図である。
【図57】コアユニットＦ７０の前面側から見た場合の、右手でコアユニットＦ７０を保持しているプレーヤの例示的な状態を示す。
【図58】コアユニットＦ７０の左側から見た場合の、右手でコアユニットＦ７０を保持しているプレーヤの例示的な状態を示す。
【図59】ＬＥＤモジュールＦ８Ｌの視角、ＬＥＤモジュールＦ８Ｒの視角、および画像ピックアップエレメントＦ７４３の視角を示す略図である。
【図60】サブユニットＦ７６の右側から見た場合の、左手でサブユニットＦ７６を保持しているプレーヤの例示的な状態を示す。
【図61】ゲーム装置Ｆ３がシューティングゲームを実行しているときにモニタＦ２上に表示される例示的なゲーム画像を示す。

【図1】

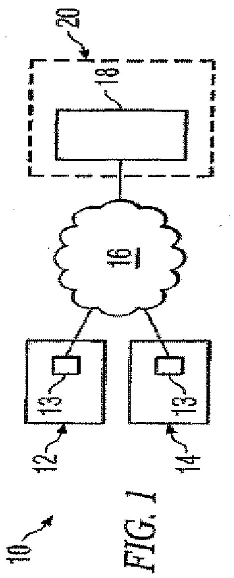


FIG. 1

【図2】

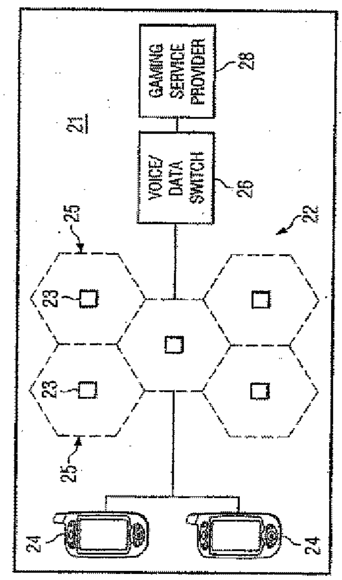

GAMING SERVICE PROVIDER
28
VOICE/ DATA SWITCH
26
21
22
23
23
24
24
25
25

FIG. 2

【図3】

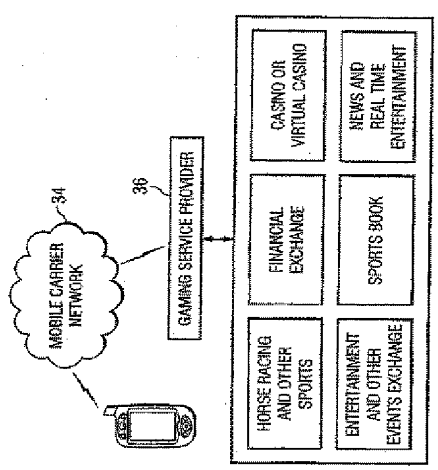

MOBILE CARRIER NETWORK
34
36
GAMING SERVICE PROVIDER
HORSE RACING AND OTHER SPORTS
FINANCIAL EXCHANGE
CASINO OR VIRTUAL CASINO
ENTERTAINMENT AND OTHER EVENTS EXCHANGE
SPORTS BOOK
NEWS AND REAL TIME ENTERTAINMENT

FIG. 3

【図4】

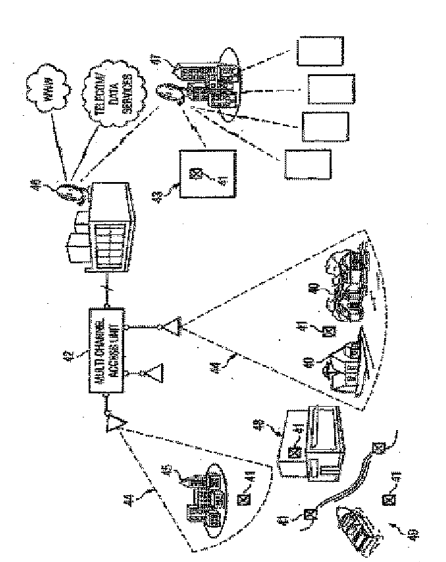


FIG. 4

【図5】

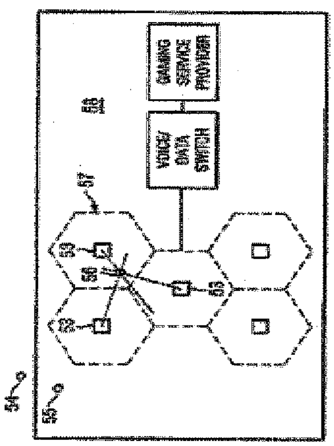


FIG. 5

【図6】

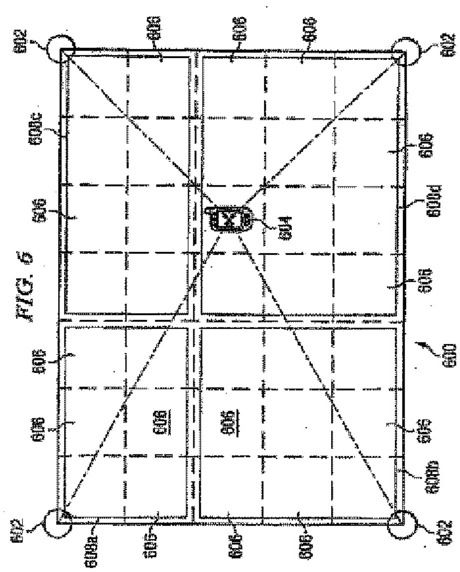


FIG. 6

【図7】

この図は公序良俗違反のため不掲載とする

【図8】

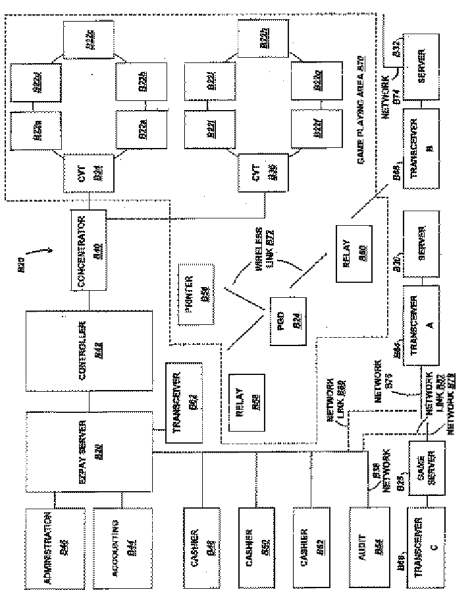


FIG. 8

【図9】

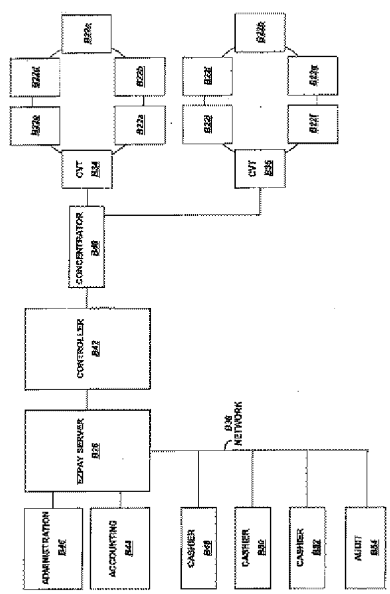

CVT
CVT
CONCENTRATOR
CONTROLLER
EZPAY SERVER
NETWORK
ADMINISTRATION
ACCOUNTING
CASHIER
CASHIER
CASHIER
AUDIT

FIG. 9

【図10】

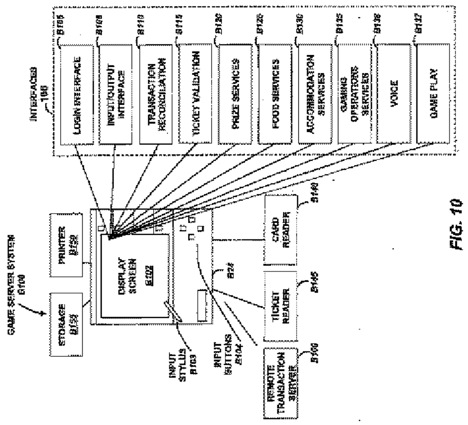

INTERFACES
LOGIN INTERFACE
INPUT/OUTPUT INTERFACE
TRANSACTION RECONCILIATION
TICKET VALIDATION
PRIZE SERVICES
FOOD SERVICES
ACCOMMODATION SERVICES
GAMING OPERATIONS SERVICES
VOICE
GAME PLAY
GAME SERVER SYSTEM
STORAGE
PRINTER
DISPLAY SCREEN
INPUT STYLUS
INPUT BUTTONS
CARD READER
TICKET READER
REMOTE TRANSACTION SERVER

FIG. 10

【図11a】

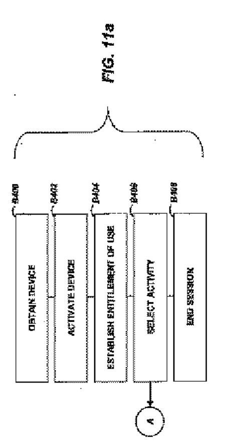

8400
OBTAIN DEVICE
8402
ACTIVATE DEVICE
8404
ESTABLISH ENTITLEMENT OF USE
8406
SELECT ACTIVITY
8408
END SESSION
A

FIG. 11a

【図11b】

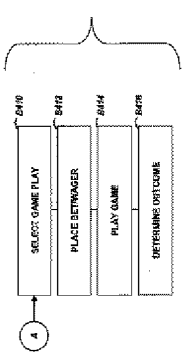

A
SELECT GAME PLAY
PLACE BET/WAGER
PLAY GAME
DETERMINE OUTCOME

FIG. 11b

【図12】

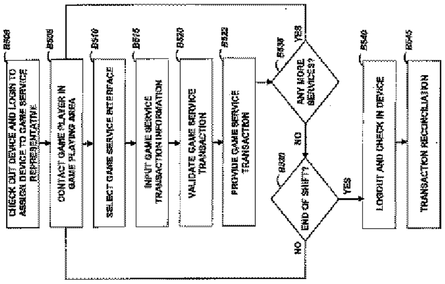

CHECK OUT DEVICE AND LOGIN TO ASSIGN DEVICE TO GAME SERVICE REPRESENTATIVE
CONTACT GAME PLAYER IN GAME PLAYING AREA
SELECT GAME SERVICE INTERFACE
INPUT GAME SERVICE TRANSACTION INFORMATION
VALIDATE GAME SERVICE TRANSACTION
PROVIDE GAME SERVICE TRANSACTION
ANY MORE SERVICES?
YES
NO
END OF SHIFT?
YES
NO
LOGOUT AND CHECK IN DEVICE
TRANSACTION RECONCILIATION

FIG. 12

【図13】

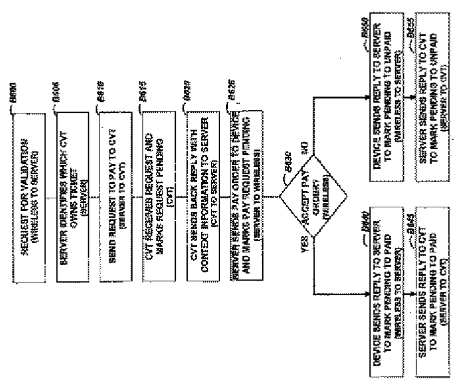


FIG. 13

【図14A】

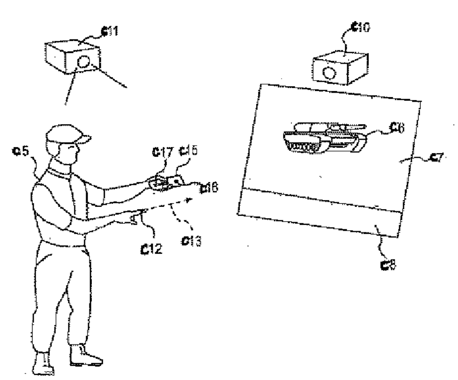


FIG. 14A

【図14B】

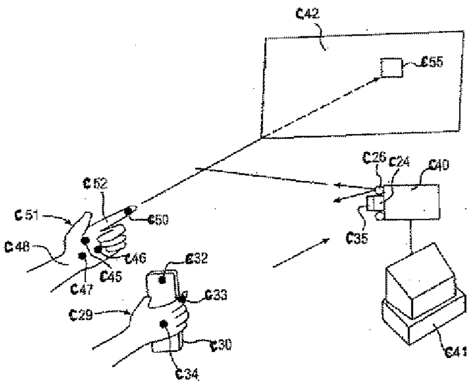

C42
C55
C26
C24
C40
C52
C51
C50
C48
C46
C35
C32
C45
C47
C33
C29
C30
C34
C41

FIG. 14B

【図14C】

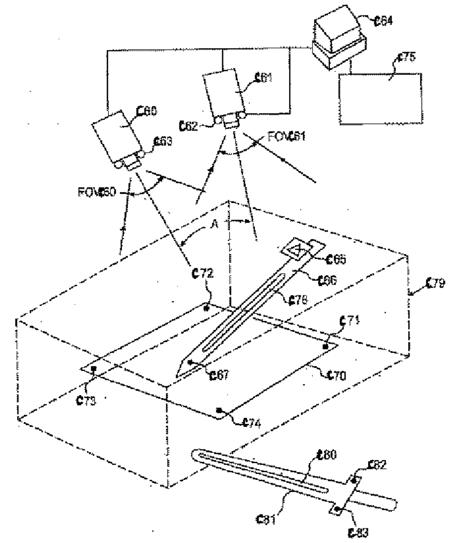


FIG. 14C

【図14D】

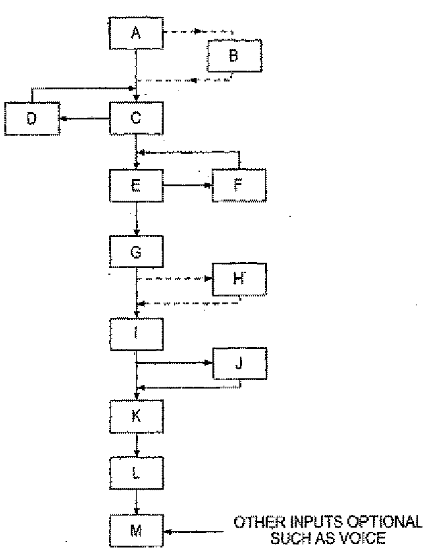


FIG. 14D

【図14E】

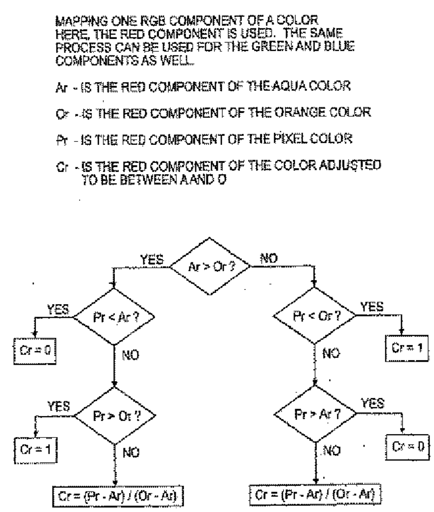

MAPPING ONE RGB COMPONENT OF A COLOR
HERE, THE RED COMPONENT IS USED. THE SAME
PROCESS CAN BE USED FOR THE GREEN AND BLUE
COMPONENTS AS WELL.
Ar - IS THE RED COMPONENT OF THE AQUA COLOR
Or - IS THE RED COMPONENT OF THE ORANGE COLOR
Pr - IS THE RED COMPONENT OF THE PIXEL COLOR
Cr - IS THE RED COMPONENT OF THE COLOR ADJUSTED
TO BE BETWEEN A AND O
Ar > Or ?
YES
NO
Pr < Ar ?
YES
Cr = 0
NO
Pr > Or ?
YES
Cr = 1
NO
Cr = (Pr - Ar) / (Or - Ar)
Pr < Or ?
YES
Cr = 1
NO
Pr > Ar ?
YES
Cr = 0
NO
Cr = (Pr - Ar) / (Or - Ar)

FIG. 14E

【図15】

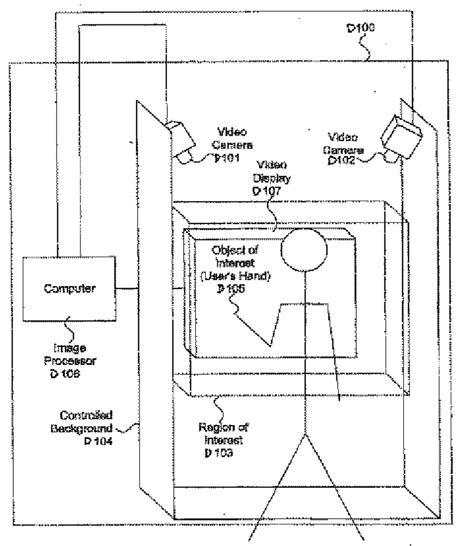

D100
Video Camera D101
Video Camera D102
Video Display D107
Object of Interest (User's Hand) D105
Computer
Image Processor D106
Controlled Background D104
Region of Interest D103

FIG. 15

【図16A】

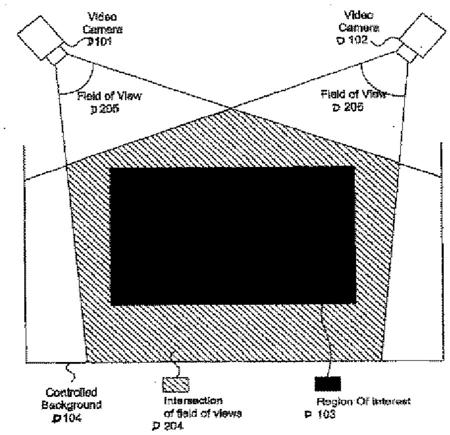

Video Camera 101
Video Camera 102
Field of View 205
Field of View 206
Controlled Background 104
Intersection of field of views 204
Region Of Interest 103

FIG. 16A

【図16B】

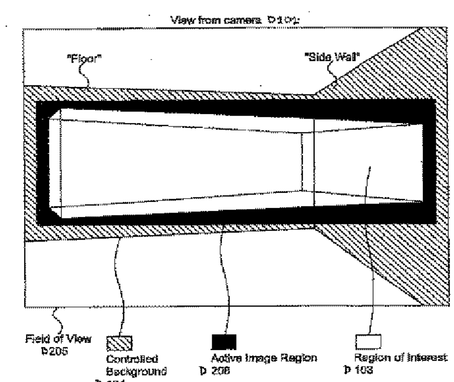

View from camera D101:
"Floor"
"Side Wall"
Field of View
D205
Controlled Background
D 104
Active Image Region
D 208
Region of Interest
D 103

FIG. 16B

【図17】

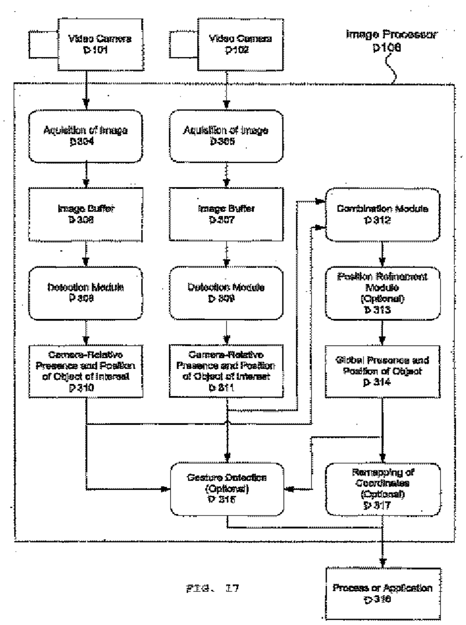

Video Camera
D101
Video Camera
D102
Image Processor
D106
Aquisition of Image
D304
Aquisition of Image
D305
Image Buffer
D306
Image Buffer
D307
Combination Module
D312
Detection Module
D308
Detection Module
D309
Position Refinement
Module
(Optional)
D313
Camera-Relative
Presence and Position
of Object of Interest
D310
Camera-Relative
Presence and Position
of Object of Interest
D311
Global Presence and
Position of Object
D314
Gesture Detection
(Optional)
D315
Remapping of
Coordinates
(Optional)
D317
Process or Application
D316

FIG. 17

【図18】

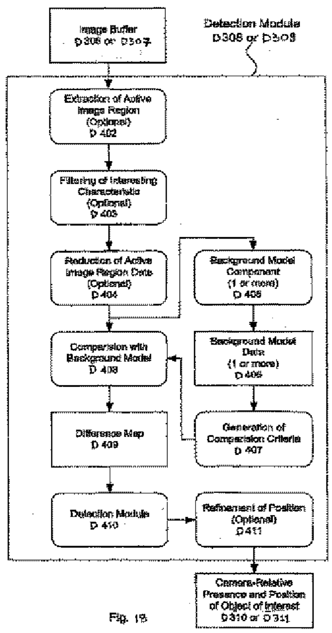


Image Buffer
D306 or D307
Detection Module
D308 or D309
Extraction of Active Image Region (Optional) D 402
Filtering of Interesting Characteristic (Optional) D 403
Reduction of Active Image Region Data (Optional) D 404
Background Model Component (1 or more) D 405
Comparision with Background Model D 408
Background Model Data (1 or more) D 406
Difference Map D 409
Generation of Comparision Criteria D 407
Detection Module D 410
Refinement of Position (Optional) D 411
Camera-Relative Presence and Position of Object of Interest D 310 or D 311

Fig. 18

【図19A】


Raw image
Fig. 19A

【図19B】

Filtered Image
Fig. 19 B

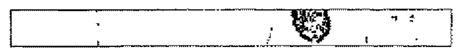
【図19C】

Low-Value Threshold
Fig. 19c

【図19D】

High-Value Threshold
Fig. 19D

【図19E】



Binary Difference
Fig. 19E

【図20】

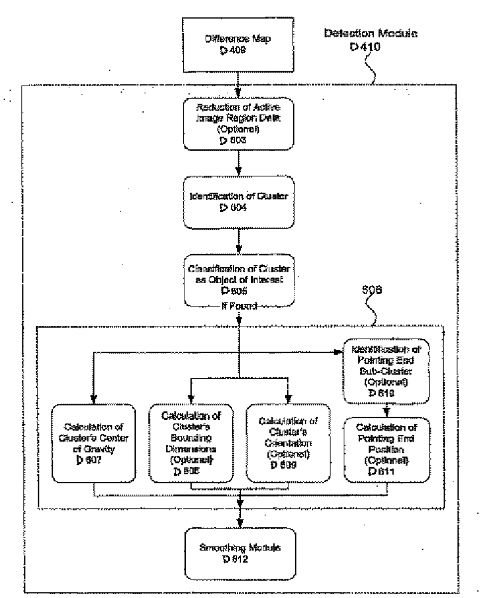
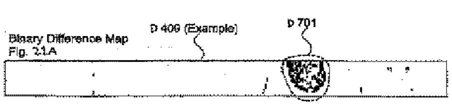
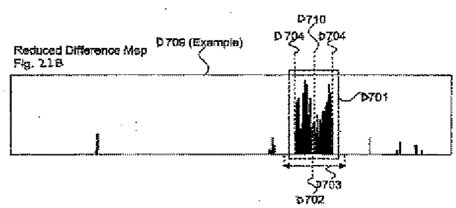

Difference Map
409
Detection Module
410
Reduction of Active
Image Region Data
(Optional)
603
Identification of Cluster
604
Classification of Cluster
as Object of Interest
605
If Found
606
Identification of
Pointing End
Sub-Cluster
(Optional)
610
Calculation of
Cluster's Center
of Gravity
607
Calculation of
Cluster's
Bounding
Dimensions
(Optional)
608
Calculation of
Cluster's
Orientation
(Optional)
609
Calculation of
Pointing End
Position
(Optional)
611
Smoothing Module
612

Fig. 2-0

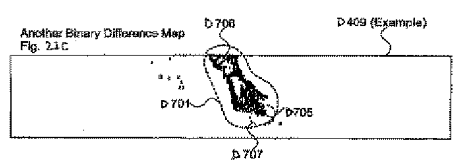
【図21A】



【図21B】


Reduced Difference Map
Fig. 21B
D709 (Example)
D710
D704
D704
D701
D703
D702

【図21C】


Another Binary Difference Map
Fig. 21C
D 708
D 409 (Example)
D 701
D705
D 707

【図22】

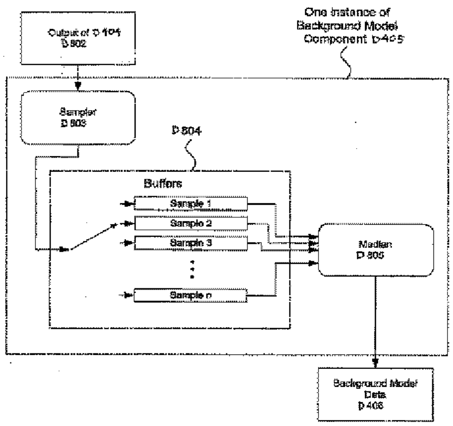

One instance of
Background Model
Component
Sampler
D 804
Buffers
Sample 1
Sample 2
Sample 3
Sample n
Median
Background Model
Data

Fig. 22.

【図A】

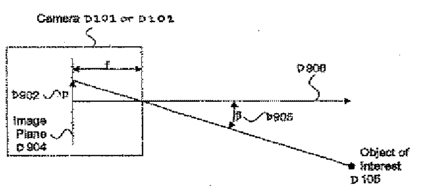

Camera D101 or D102
f
D906
D902
P
β
D905
Image
Plane
D 904
Object of
Interest
D 105

Fig. 23B

【図B】

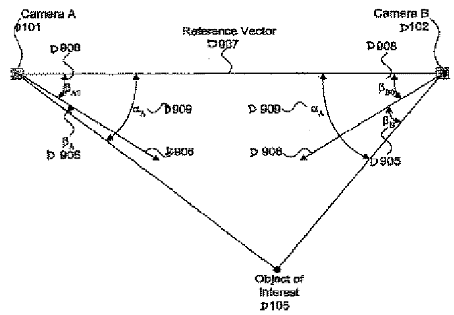

Camera A
101
Reference Vector
907
Camera B
102
908
908
909
909
906
906
905
905
Object of
interest
105

Fig. 13 B

【図24】

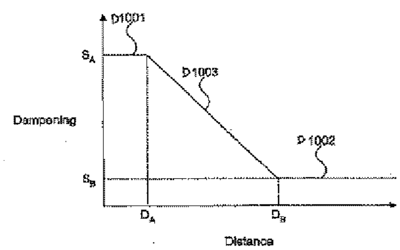


Fig. 24

【図25A】

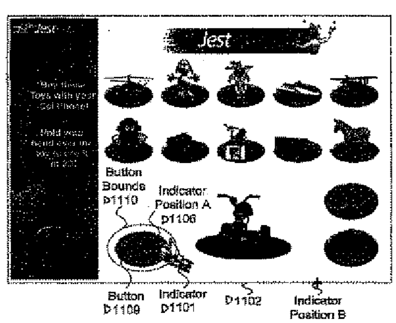

Jest
Button
Bounds
D1110
Indicator
Position A
D1106
Button
D1109
Indicator
D1101
D1102
Indicator
Position B
D1108

Fig. 25A

【図25B】

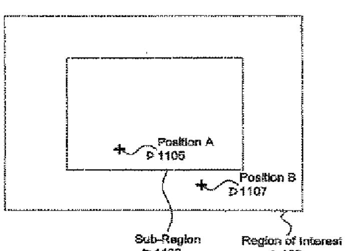

Position A
1105
Position B
1107
Sub-Region
1103
Region of Interest
103

Fig. 25 B

【図26A】

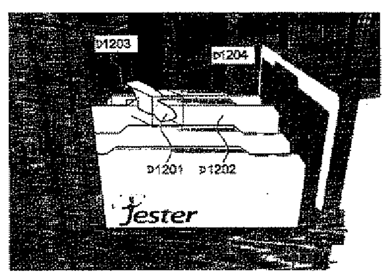


Fig. 26A

【図26B】

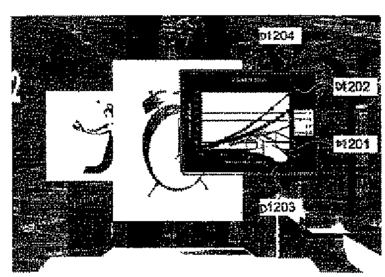


Fig. 26B

【図27A】

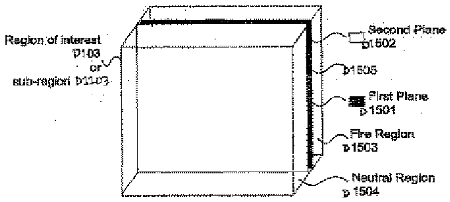

Region of interest
103
or
sub-region 1103
Second Plane
1502
1505
First Plane
1501
Fire Region
1503
Neutral Region
1504

Fig. 27 A

【図27B】

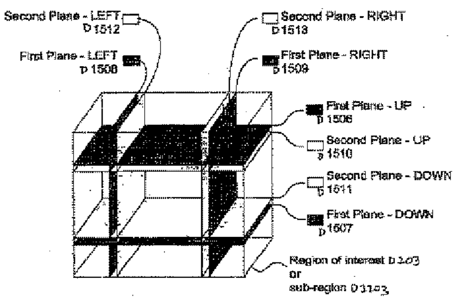

Second Plane - LEFT
D 1512
First Plane - LEFT
D 1508
Second Plane - RIGHT
D 1513
First Plane - RIGHT
D 1509
First Plane - UP
D 1506
Second Plane - UP
D 1510
Second Plane - DOWN
D 1511
First Plane - DOWN
D 1507
Region of interest D 203
or
sub-region D 1103

Fig. 27 B

【図27C】

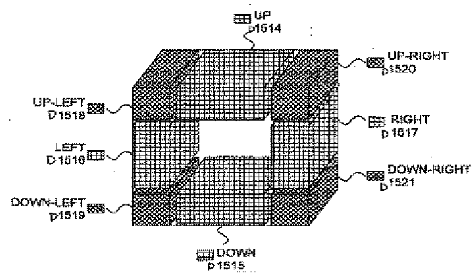

UP
p1514
UP-RIGHT
p1520
UP-LEFT
p1518
RIGHT
p1517
LEFT
p1516
DOWN-RIGHT
p1521
DOWN-LEFT
p1519
DOWN
p1515

Fig. 27C

【図27D】

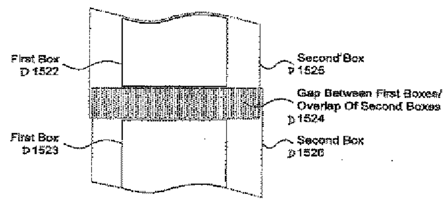

First Box
1522
Second Box
1525
Gap Between First Boxes/
Overlap Of Second Boxes
1524
First Box
1523
Second Box
1526

Fig. 27D

【図28】

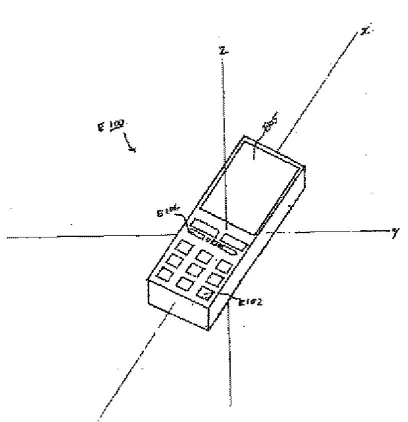


FIG. 28

【図29】

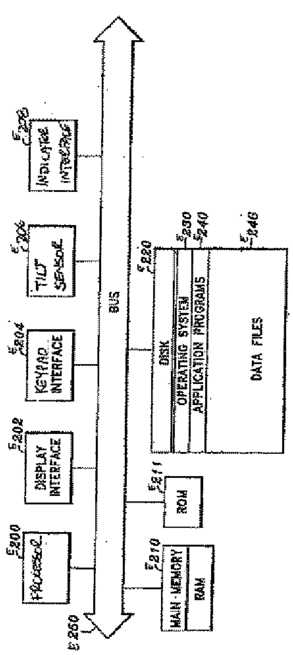
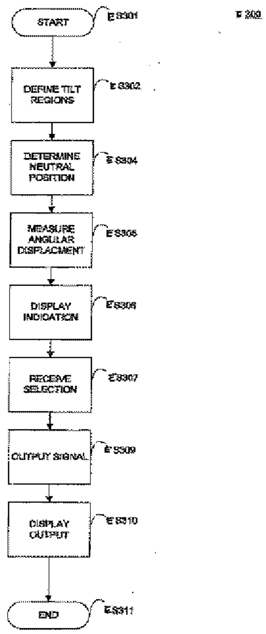

DISPLAY INTERFACE
KEYPAD INTERFACE
TILT SENSOR
BUS
MAIN MEMORY
RAM
ROM
DISK
OPERATING SYSTEM
APPLICATION PROGRAMS
DATA FILES

FIG. 29

【図30】

FIGURE 30



【図31A】

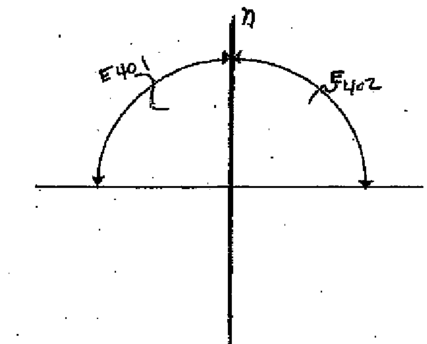


Figure 31A

【図31B】

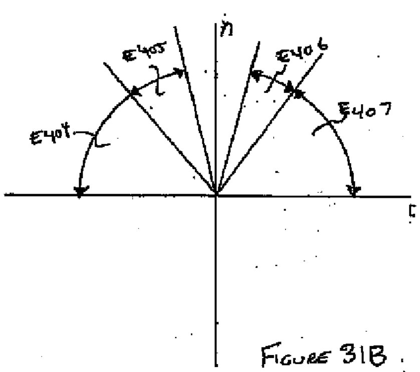
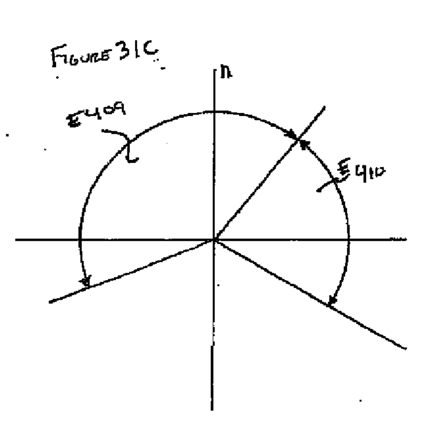
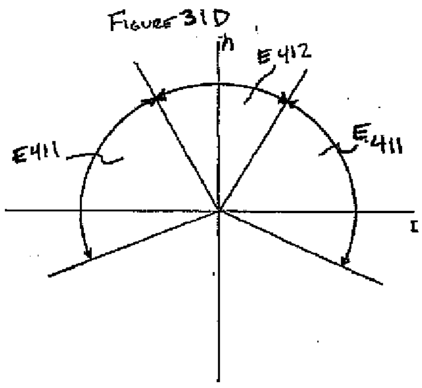


FIGURE 31B

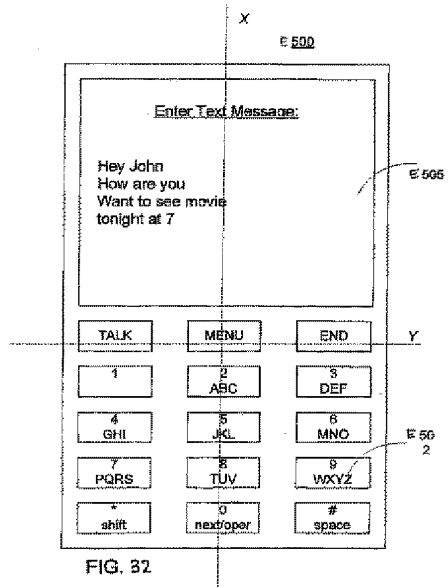
【図31C】



【図31D】



【図32】


X
500
Enter Text Message:
Hey John
How are you
Want to see movie
tonight at 7
505
TALK
MENU
END
Y
1
2
ABC
3
DEF
4
GHI
5
JKL
6
MNO
502
7
PQRS
8
TUV
9
WXYZ
*
shift
0
next/oper
#
space
FIG. 32

【図33A】

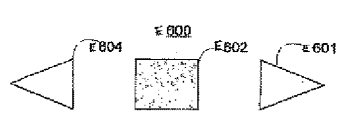


Figure 33A

【図33B】

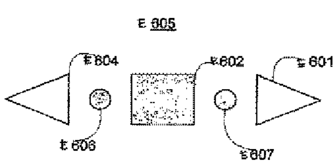
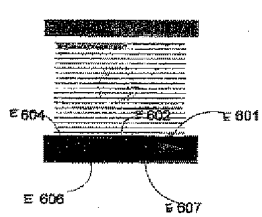


Figure 33B

【図33C】

Figure 33C



【図33D】

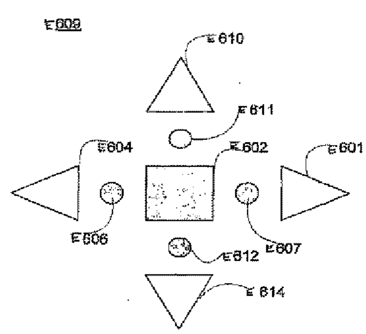


Figure 33D

【図33E】

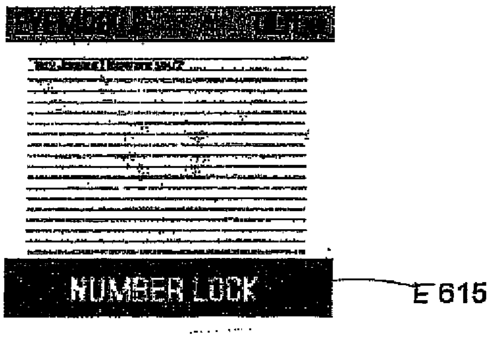


Figure 33E

【図34A】

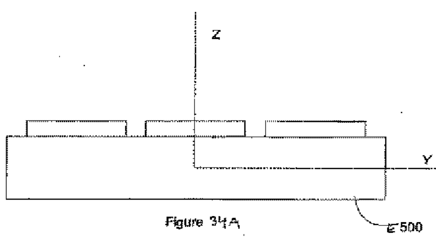


Figure 34A

【図34B】

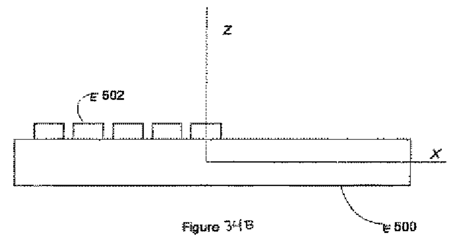


Figure 34B

【図35A】

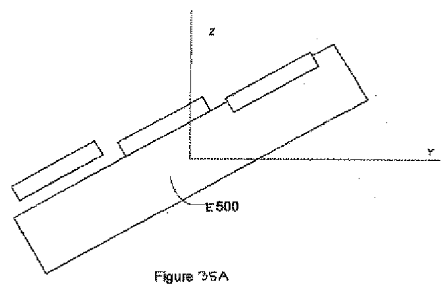


Figure 35A

【図35B】

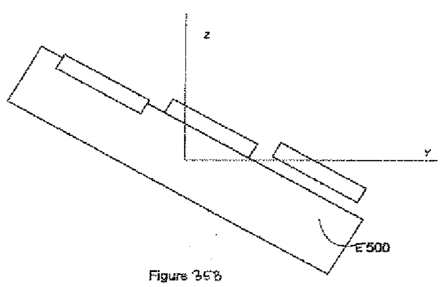


Figure 35B

【図36A】

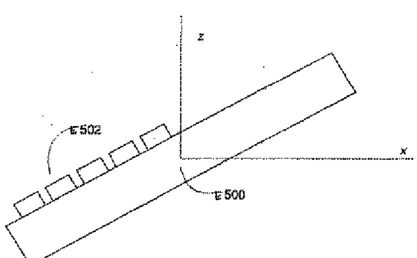


Figure 36A

【図36B】

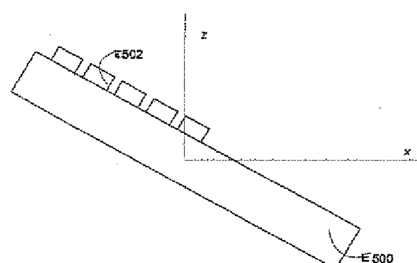


Figure 36B

【図37】

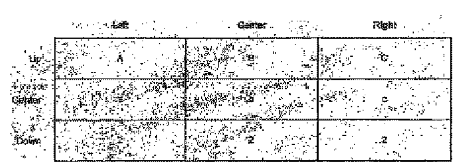


Figure 37

【図38A】

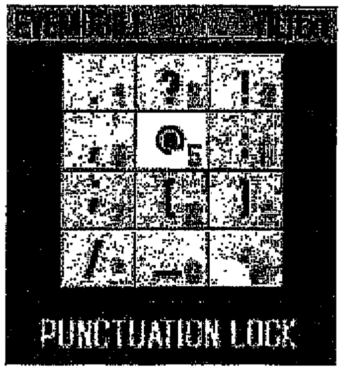


Figure. 38A

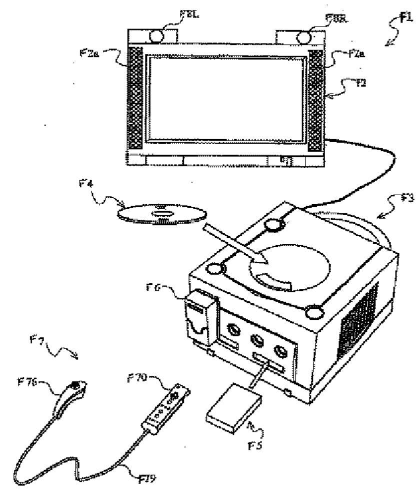
【図38B】

	Left	Center	Right
Up	!	@	#
Center	$	%	^
Down	&	*	(

Figure 38 B

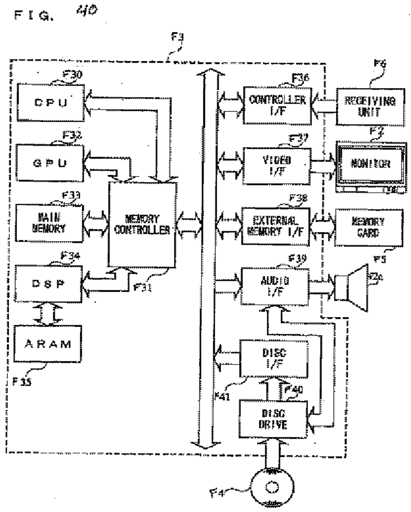
【図39】

F I G. 39



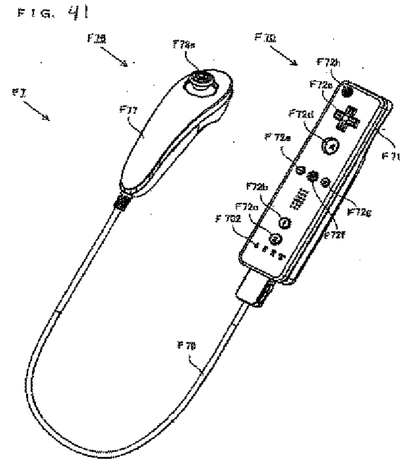
【図40】

FIG. 40


F3
F30
CPU
F32
GPU
F33
MAIN MEMORY
MEMORY CONTROLLER
F34
DSP
F31
ARAM
F35
F36
CONTROLLER I/F
F6
RECEIVING UNIT
F37
VIDEO I/F
MONITOR
F38
EXTERNAL MEMORY I/F
MEMORY CARD
F5
F39
AUDIO I/F
DISC I/F
F41
F40
DISC DRIVE
F4

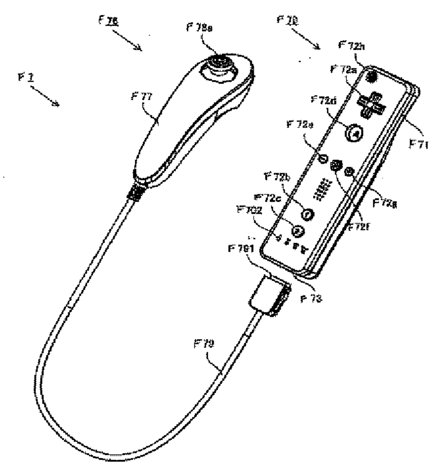
【図41】

FIG. 41



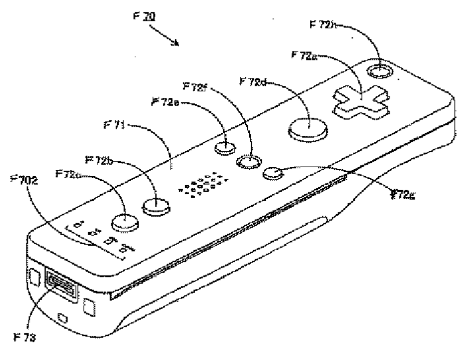
【図42】

FIG. 42


F1
F76
F78a
F77
F70
F72h
F72a
F72d
F72e
F71
F72b
F72c
F702
F72g
F72f
F701
F73
F79

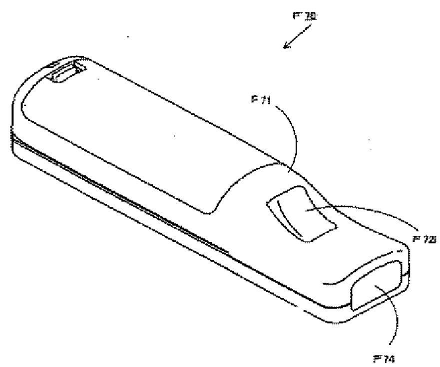
【図43】

F I G. 43



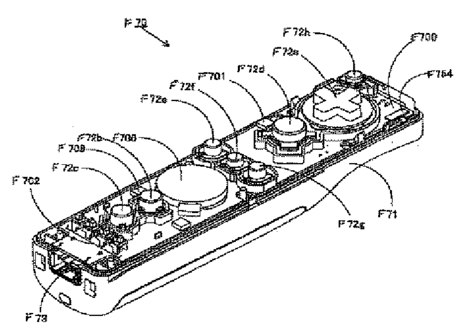
【図44】

FIG. 44



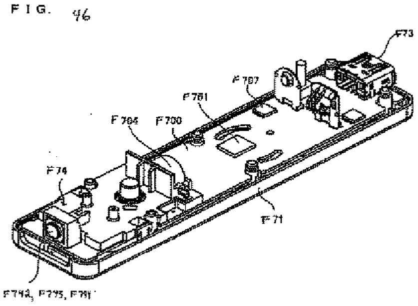
【図45】

F I G. 45


F 70
F 72h
F700
F 72a
F764
F 72d
F701
F 72f
F 72e
F72b
F706
F 708
F 72c
F 702
F 72g
F71
F 73

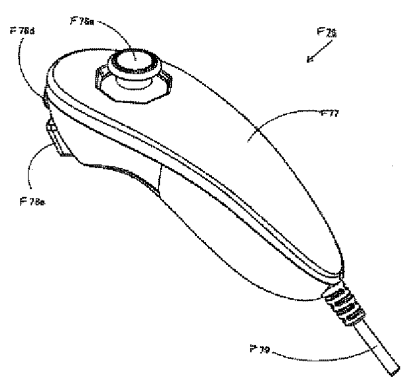
【図46】

FIG. 46



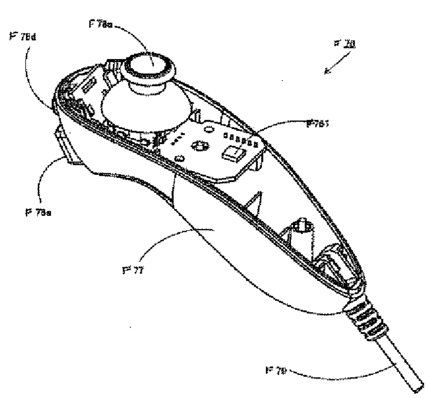
【図47】

FIG. 47



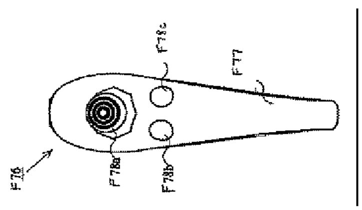
【図48】

FIG. 48

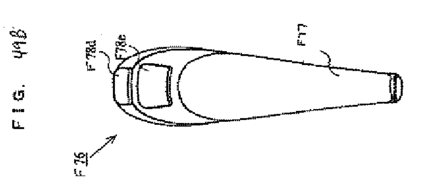


【図49A】

FIG. 49A

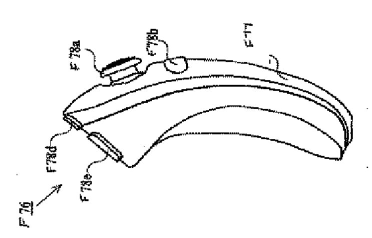


【図49B】



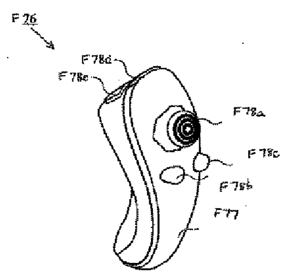
【図49C】

FIG. 49C



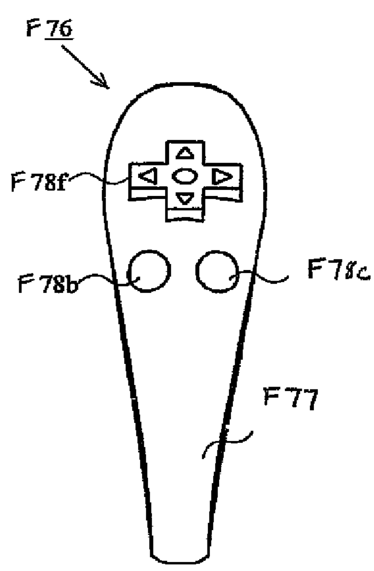
【図50】

FIG. 50



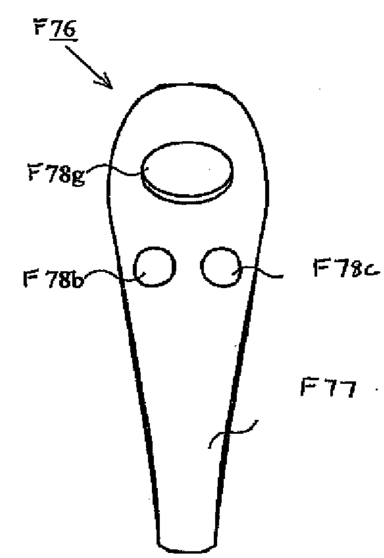
【図51】

FIG. 51



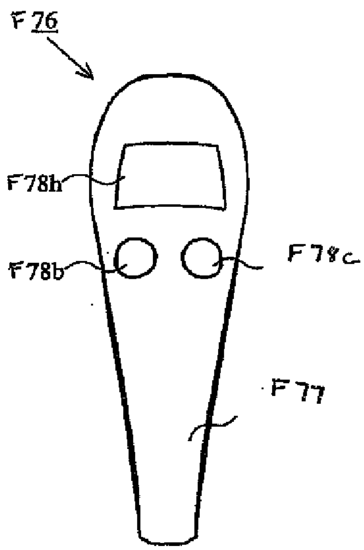
【図52】

FIG. 52

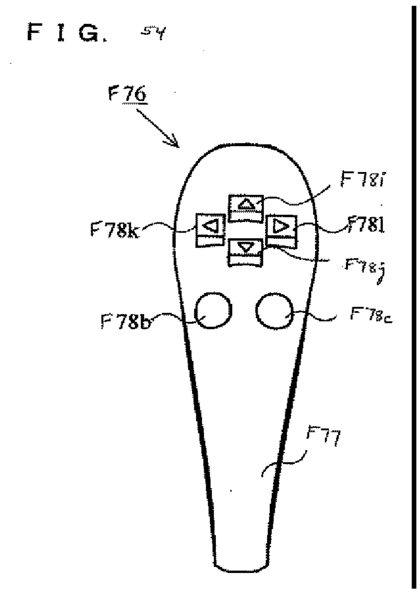


【図53】

FIG. 53

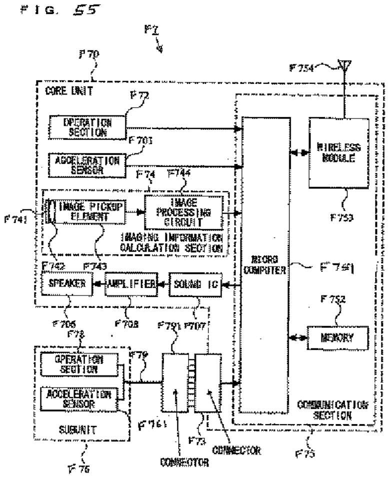


【図54】



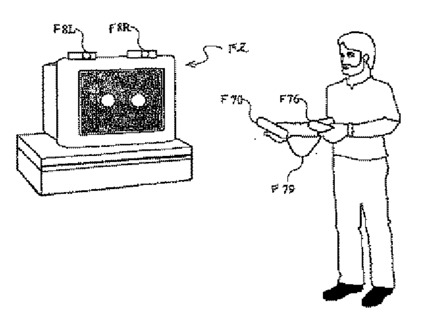
【図55】

FIG. 55


F7
F70
CORE UNIT
F72
OPERATION SECTION
F701
ACCELERATION SENSOR
F74
F744
IMAGE PICKUP ELEMENT
IMAGE PROCESSING CIRCUIT
IMAGING INFORMATION CALCULATION SECTION
F741
F742
F743
SPEAKER
AMPLIFIER
SOUND IC
F706
F708
F791
F707
F754
WIRELESS MODULE
F753
MICRO COMPUTER
F752
MEMORY
COMMUNICATION SECTION
F75
F78
OPERATION SECTION
ACCELERATION SENSOR
SUBUNIT
F79
F761
F73
CONNECTOR
CONNECTOR
F76

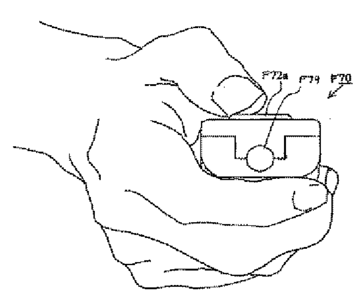
【図56】

FIG. 56

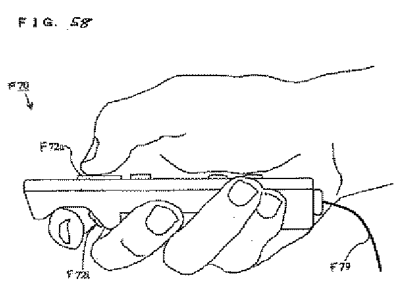


【図57】

FIG. 57

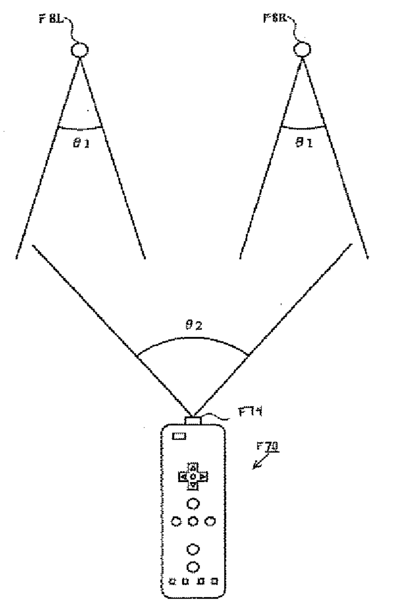


【図58】



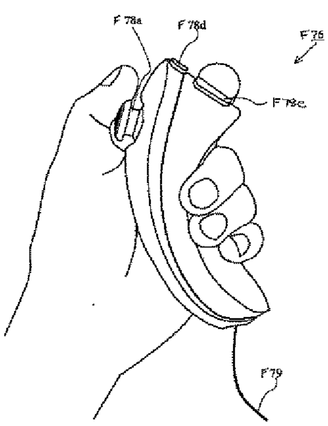
【図59】

F I G. 59



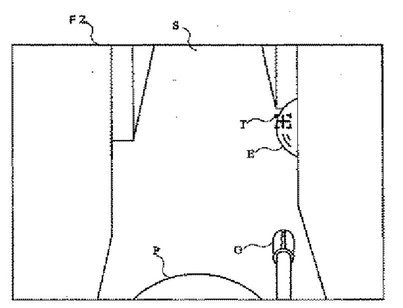
【図60】

F I G. 60



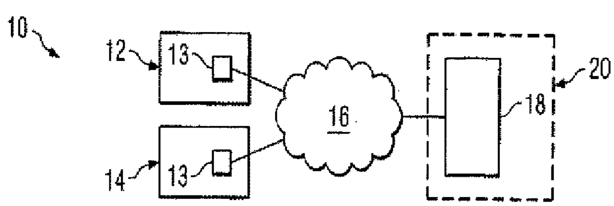
【図61】

F I G. 61

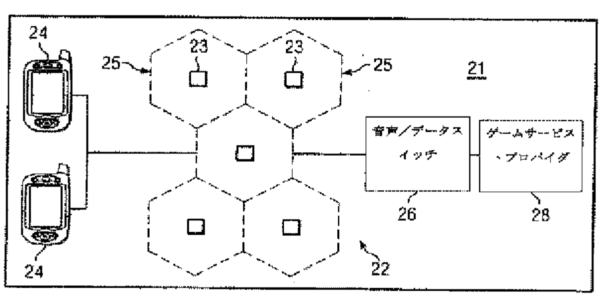


【誤訳訂正書】
【提出日】平成22年3月23日(2010.3.23)
【誤訳訂正1】
【訂正対象書類名】図面
【訂正対象項目名】全図
【訂正方法】変更
【訂正の内容】

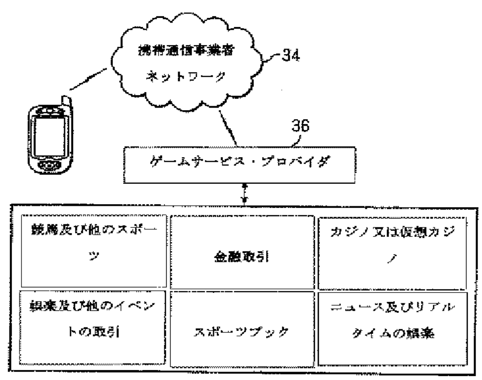
【図1】



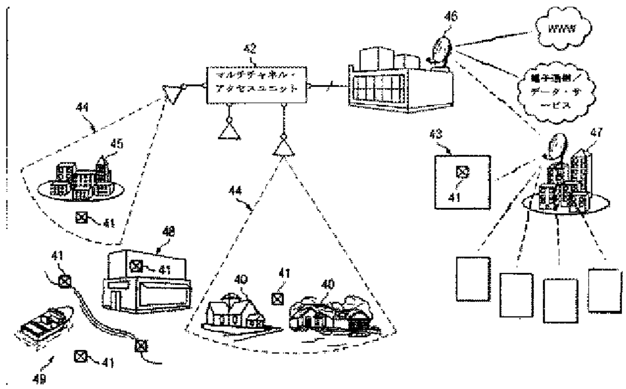
【図2】



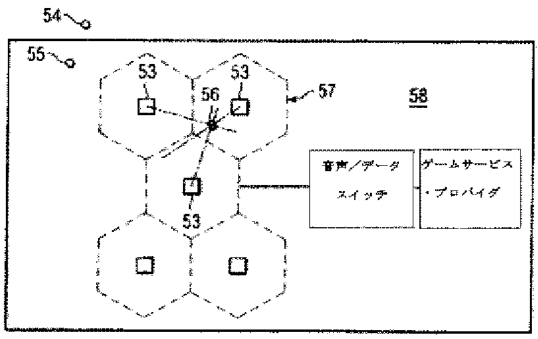
【図3】


携帯通信事業者
ネットワーク
34
36
ゲームサービス・プロバイダ
競馬及び他のスポーツ
金融取引
カジノ又は仮想カジノ
娯楽及び他のイベントの取引
スポーツブック
ニュース及びリアルタイムの娯楽

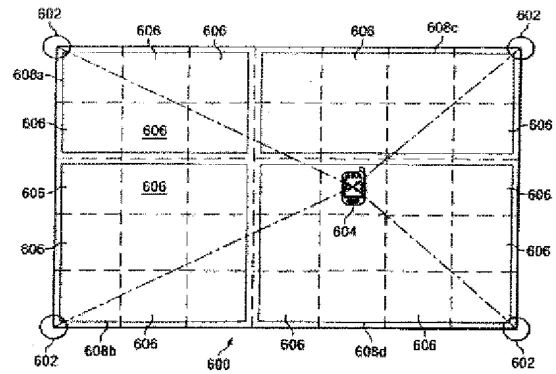
【図4】



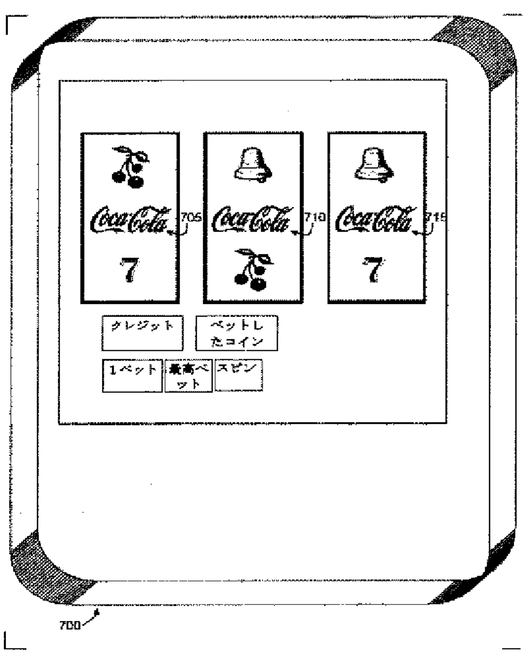
【図5】



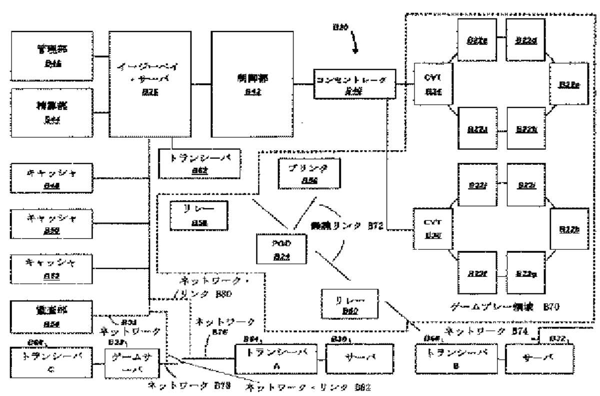
【図6】


602
606
606
606
608c
602
608a
606
606
606
606
606
606
606
606
604
606
606
602
608b
606
600
606
608d
606
602

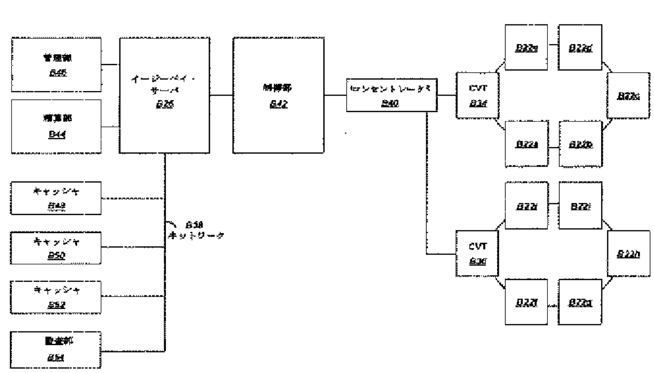
【図7】



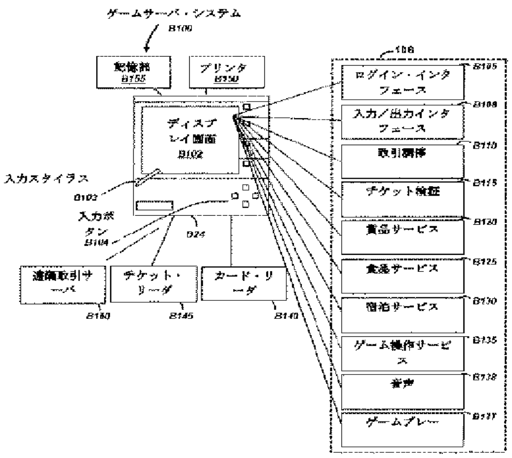
【図8】


管理部
制御部
トランシーバ
プリンタ
キャッシャ
キャッシャ
キャッシャ
リレー
POD
リレー
CVT
CVT
ネットワーク・リンク B80
ネットワーク
ネットワーク
ゲームプレー領域 B70
ネットワーク B74
トランシーバ C
ゲームサーバ
トランシーバ A
サーバ
トランシーバ B
サーバ
ネットワーク B78
ネットワーク・リンク B82

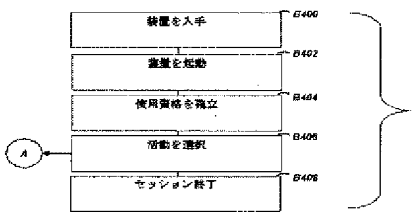
【図9】



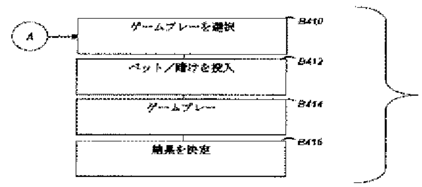
【図10】


ゲームサーバ・システム
B100
記憶部
B155
プリンタ
B150
ディスプレイ画面
B102
入力スタイラス
B103
入力ボタン
B104
B24
遠隔取引サーバ
B160
チケット・リーダ
B145
カード・リーダ
B140
106
ログイン・インタフェース
B105
入力／出力インタフェース
B108
取引調停
B110
チケット検証
B115
賞品サービス
B120
食品サービス
B125
宿泊サービス
B130
ゲーム操作サービス
B135
音声
B138
ゲームプレー
B137

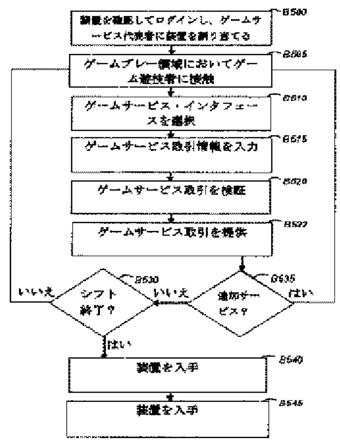
【図11a】


装置を入手
装置を起動
使用資格を確立
活動を選択
セッション終了
A
B400
B402
B404
B406
B408

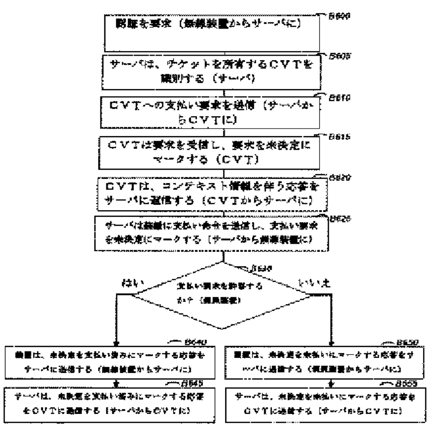
【図11b】



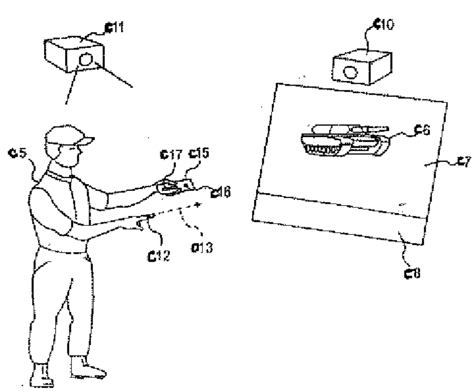
【図12】


装置を確認してログインし、ゲームサービス代表者に装置を割り当てる
B500
ゲームプレー領域においてゲーム遊技者に接触
B505
ゲームサービス・インタフェースを選択
B510
ゲームサービス取引情報を入力
B515
ゲームサービス取引を検証
B520
ゲームサービス取引を提供
B522
シフト終了？
B530
追加サービス？
B535
いいえ
いいえ
はい
はい
装置を入手
B540
装置を入手
B545

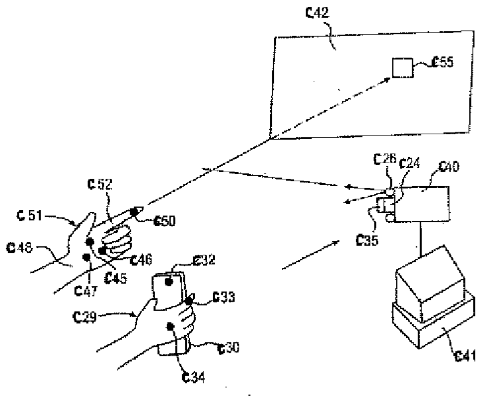
【図13】


サーバは、チケットを所有するＣＶＴを識別する（サーバ）
ＣＶＴへの支払い要求を送信（サーバからＣＶＴに）
ＣＶＴは要求を受信し、要求を未決定にマークする（ＣＶＴ）
はい
いいえ

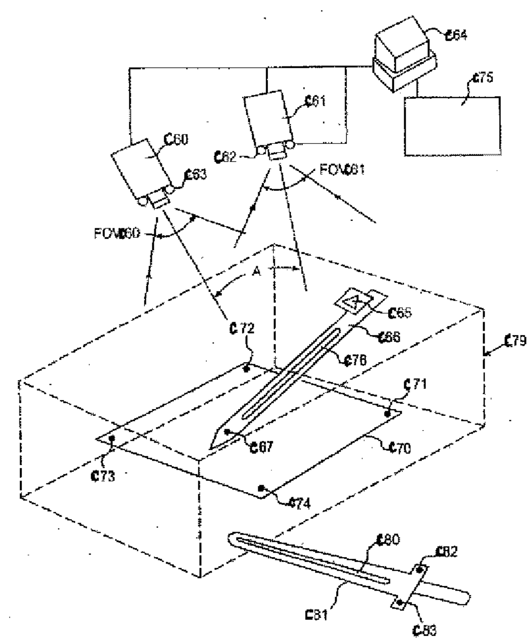
【図14A】



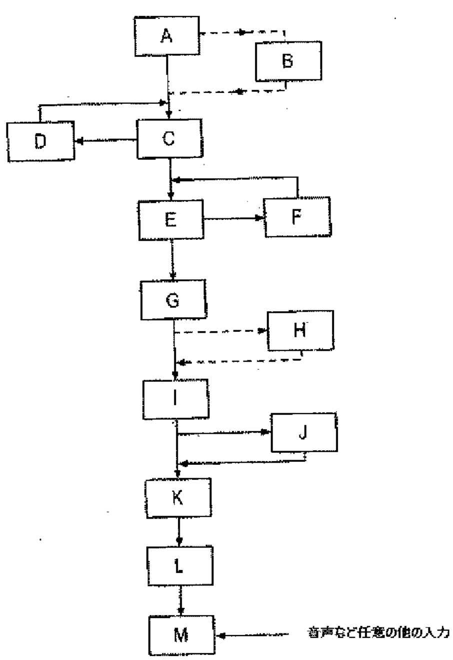
【図14B】


C42
C55
C26
C24
C40
C52
C51
C60
C48
C46
C45
C47
C35
C32
C33
C29
C30
C34
C41

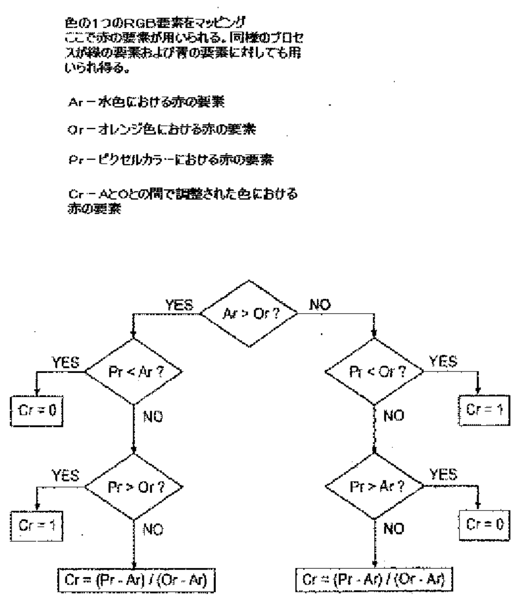
【図14C】


C64
C75
C61
C60
C62
C63
FOV61
FOV60
A
C65
C72
C66
C79
C76
C71
C67
C70
C73
C74
C80
C82
C81
C83

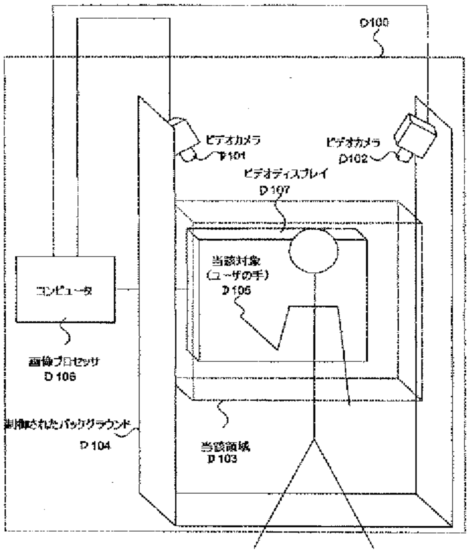
【図14D】



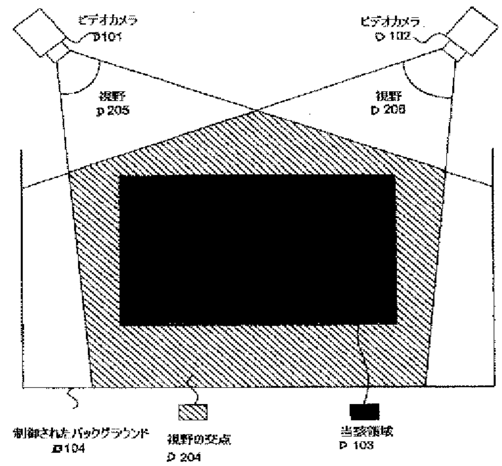
【図14E】


色の1つのRGB要素をマッピング
ここで赤の要素が用いられる。同様のプロセスが緑の要素および青の要素に対しても用いられ得る。
Ar－水色における赤の要素
Or－オレンジ色における赤の要素
Pr－ピクセルカラーにおける赤の要素
Cr－AとOとの間で調整された色における赤の要素
Ar > Or ?
YES
NO
Pr < Ar ?
Pr < Or ?
Cr = 0
Cr = 1
Pr > Or ?
Pr > Ar ?
Cr = 1
Cr = 0
Cr = (Pr - Ar) / (Or - Ar)
Cr = (Pr - Ar) / (Or - Ar)

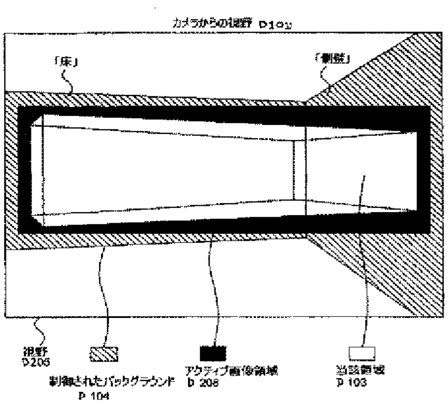
【図15】


D100
ビデオカメラ
D101
ビデオカメラ
D102
ビデオディスプレイ
D107
コンピュータ
当該対象
(ユーザの手)
D105
画像プロセッサ
D106
制御されたバックグラウンド
D104
当該領域
D103

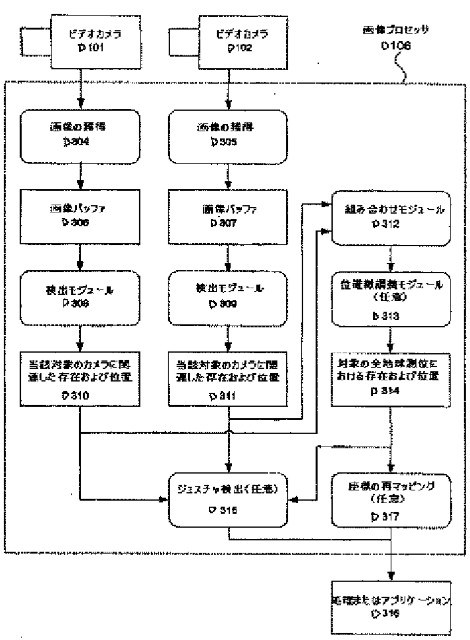
【図16A】


ビデオカメラ
101
ビデオカメラ
102
視野
205
視野
206
制御されたバックグラウンド
104
視野の交点
204
当該領域
103

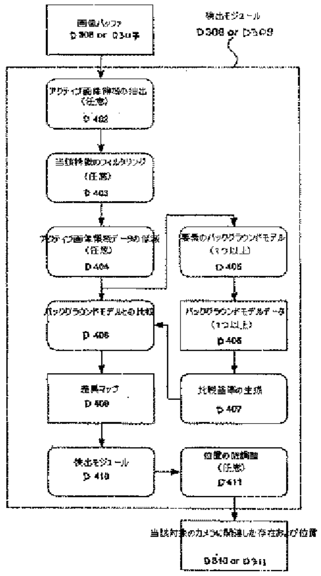
【図16B】


カメラからの視野
「床」
「側壁」
視野
制御されたバックグラウンド
P 104
アクティブ画像領域
D 208
当該領域
P 103

【図17】


ビデオカメラ
101
ビデオカメラ
102
画像プロセッサ
106
画像の獲得
304
画像の獲得
305
画像バッファ
306
画像バッファ
307
組み合わせモジュール
312
検出モジュール
検出モジュール
309
当該対象のカメラに関連した存在および位置
310
当該対象のカメラに関連した存在および位置
311
対象の全地球測位における存在および位置
314
ジェスチャ検出（任意）
315
座標の再マッピング
（任意）
317
処理またはアプリケーション
316

【図18】


画像バッファ
検出モジュール
(任意)
D 402
D 403
D 404
D 405
(1つ以上)
バックグラウンドモデルとの比較
D 406
バックグラウンドモデルデータ
比較基準の生成
D 407
D 409
検出モジュール
D 410
位置の微調整
D 411

【図19A】

生のデータ

【図19B】

フィルタリングされた画像

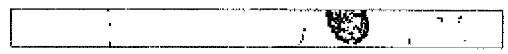
【図19C】

低い値の閾値

【図19D】

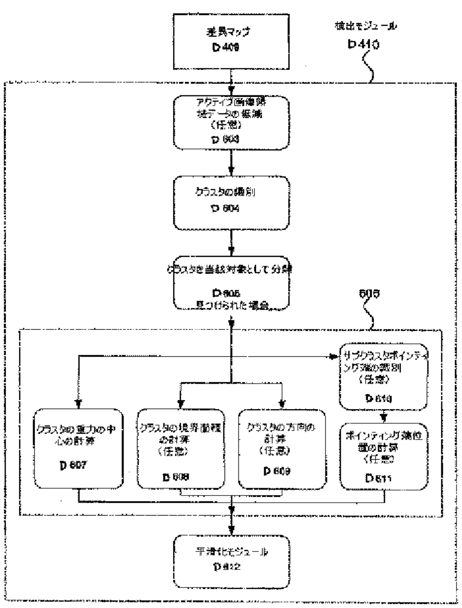
高い値の閾値

【図19E】

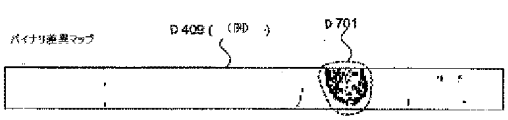


バイナリの差異

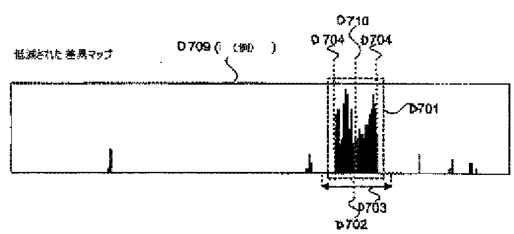
【図20】


差異マップ
D 409
検出モジュール
D 410
アクティブ画像領域データの低減
（任意）
D 603
クラスタの識別
D 604
クラスタを当該対象として分類
D 605
見つけられた場合
606
クラスタの重力の中心の計算
D 607
クラスタの境界面積の計算
（任意）
D 608
クラスタの方向の計算
（任意）
D 609
サブクラスタポインティング端の識別
（任意）
D 610
ポインティング端位置の計算
（任意）
D 611
平滑化モジュール
D 612

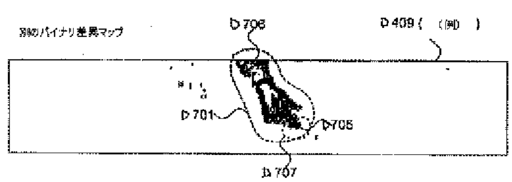
【図21A】



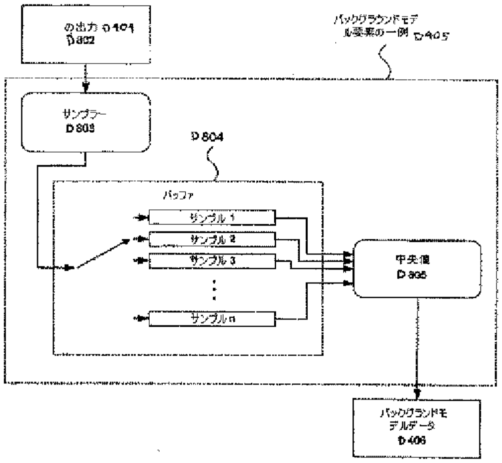
【図21B】



【図21C】

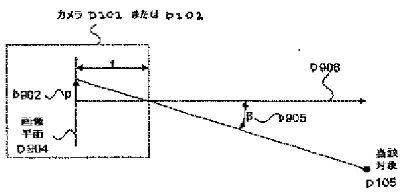


【図22】

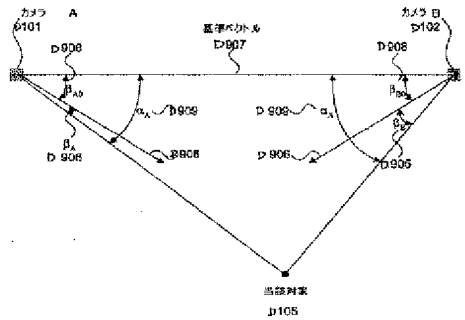


の出力 D404
D802
バックグラウンドモデル要素の一例 D405
サンプラー
D803
D804
バッファ
サンプル 1
サンプル 2
サンプル 3
サンプル n
中央値
D805
バックグラウンドモデルデータ
D406

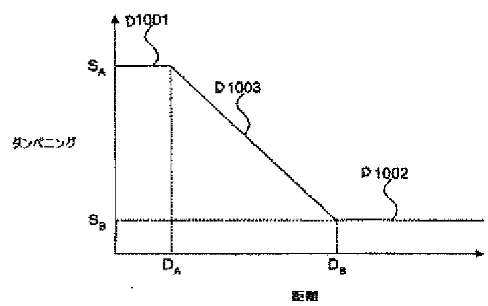
【図23A】



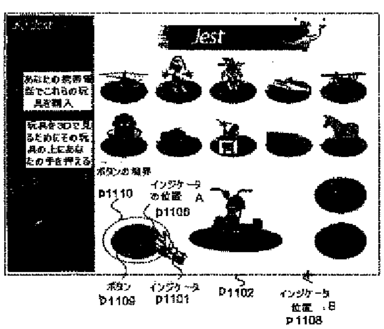
【図23B】



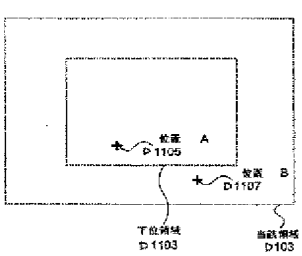
【図24】



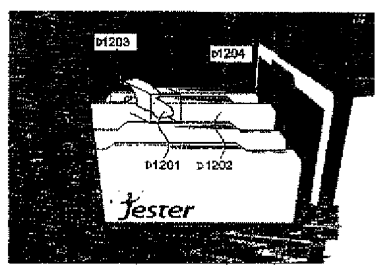
【図25A】


Jest
あなたの携帯電話でこれらの玩具を購入
玩具を3Dで見るためにその玩具の上にあなたの手を押える
ボタンの境界
P1110
インジケータの位置　A
P1106
ボタン
P1109
インジケータ
P1101
P1102
インジケータ位置　B
P1108

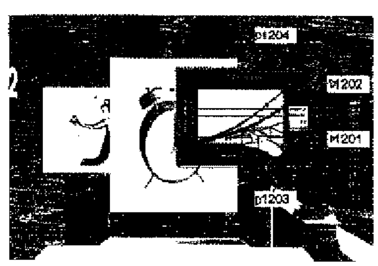
【図25B】



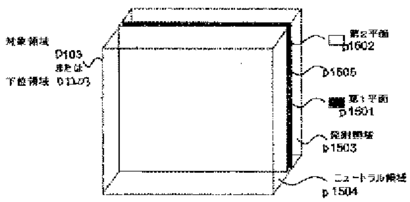
【図26A】



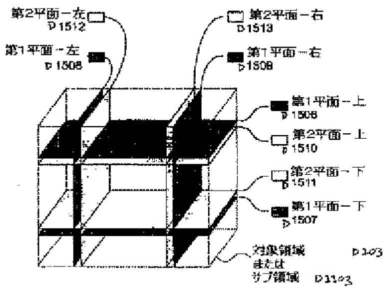
【図26B】



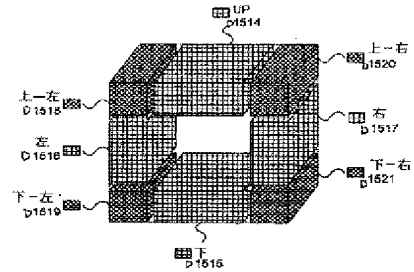
【図27A】


対象領域
D103
または
下位領域
第2平面
p1602
p1605
第1平面
p1501
p1503
ニュートラル領域
p1504

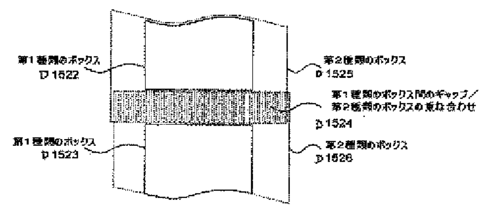
【図27B】


第2平面－左
1512
第1平面－左
第2平面－右
1513
第1平面－右
第1平面－上
第2平面－上
1510
第2平面－下
1511
第1平面－下
1507
対象領域
または
サブ領域

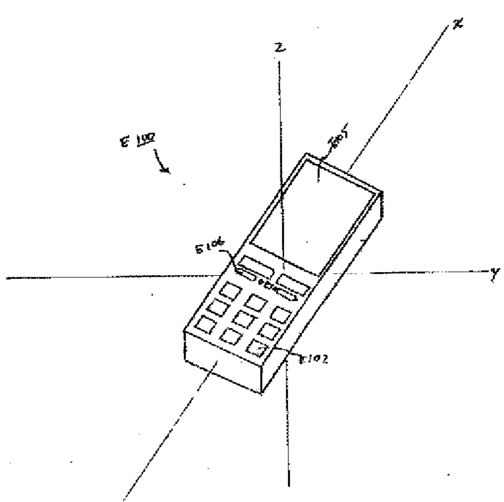
【図27C】



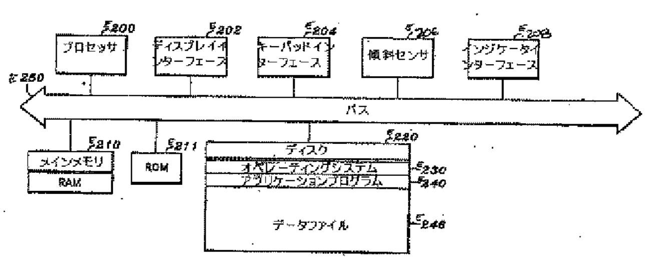
【図27D】


第１種類のボックス
1522
第２種類のボックス
1525
第１種類のボックス間のギャップ／
第２種類のボックスの重ね合わせ
1524
第１種類のボックス
1523
第２種類のボックス
1526

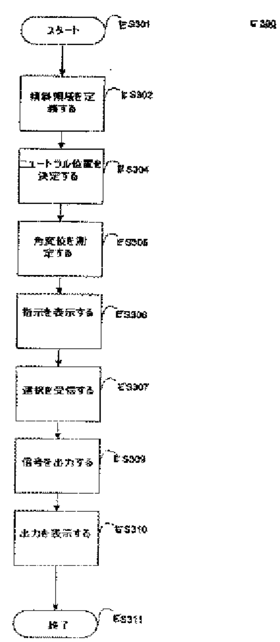
【図28】



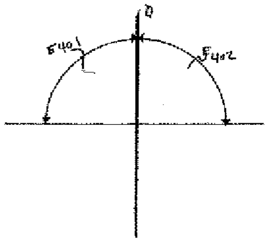
【図29】


200
プロセッサ
202
ディスプレイインターフェース
204
キーパッドインターフェース
206
傾斜センサ
208
インジケータインターフェース
250
バス
210
メインメモリ
RAM
211
ROM
220
ディスク
オペレーティングシステム
230
アプリケーションプログラム
240
データファイル
246

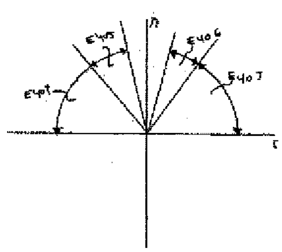
【図30】


300
スタート
S301
傾斜領域を定義する
S302
ニュートラル位置を決定する
S304
角変位を測定する
S305
指示を表示する
S306
選択を受信する
S307
信号を出力する
S309
出力を表示する
S310
終了
S311

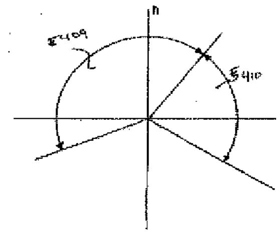
【図31A】



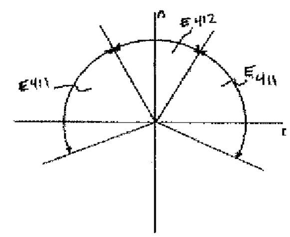
【図31B】



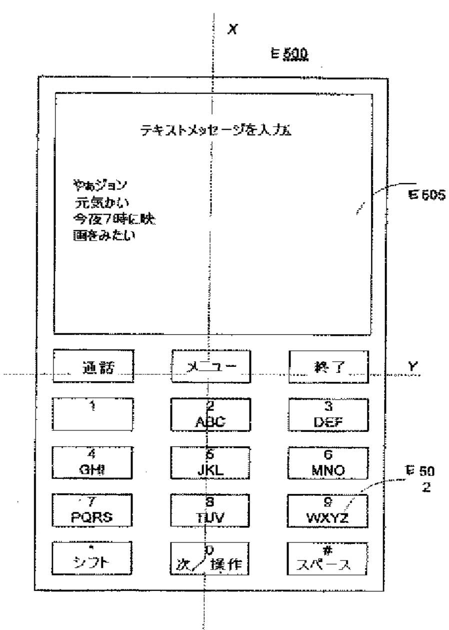
【図31C】



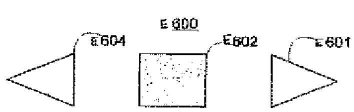
【図31D】



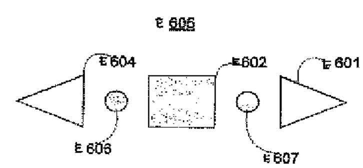
【図32】


X
E 500
テキストメッセージを入力
やあジョン
元気かい
今夜7時に映
画をみたい
E 605
通話
メニュー
終了
Y
1
2
ABC
3
DEF
4
GHI
5
JKL
6
MNO
E 502
7
PQRS
8
TUV
9
WXYZ
*
シフト
0
次／操作
#
スペース

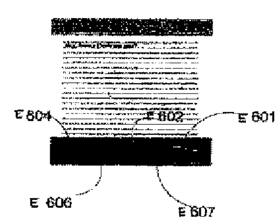
【図33A】



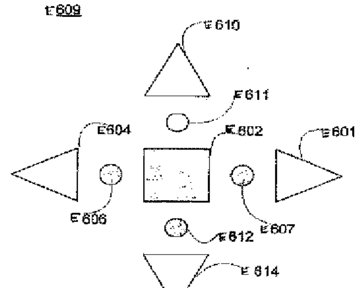
【図33B】



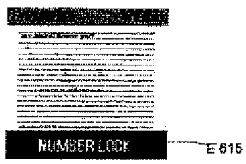
【図33C】



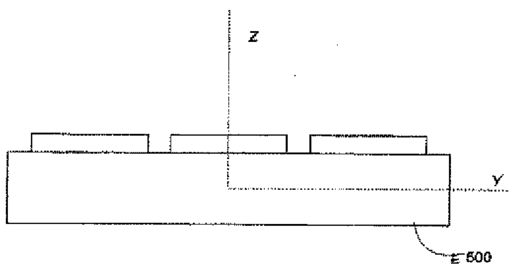
【図33D】



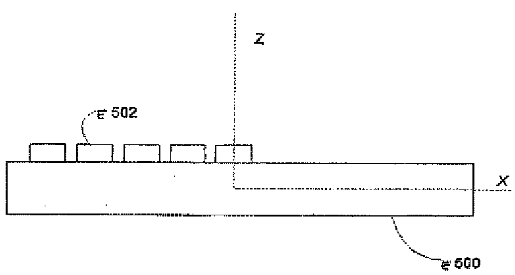
【図33E】



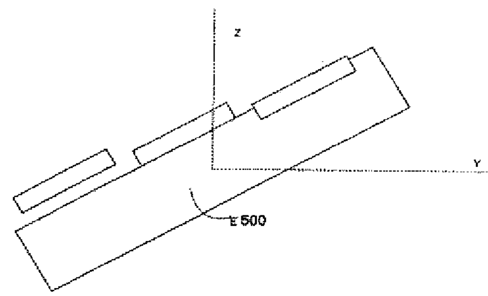
【図34A】



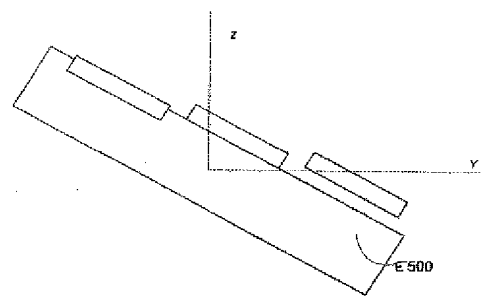
【図34B】



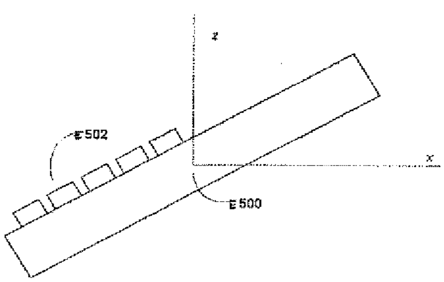
【図35A】



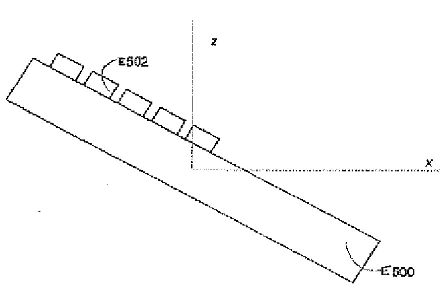
【図35B】



【図36A】



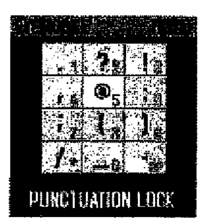
【図36B】



【図37】

	左	中央	右
上	A	B	C
中央	a	b	c
下	2	2	2

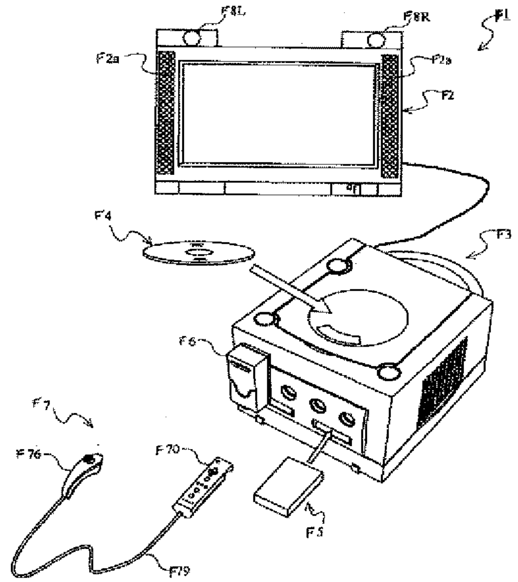
【図38A】



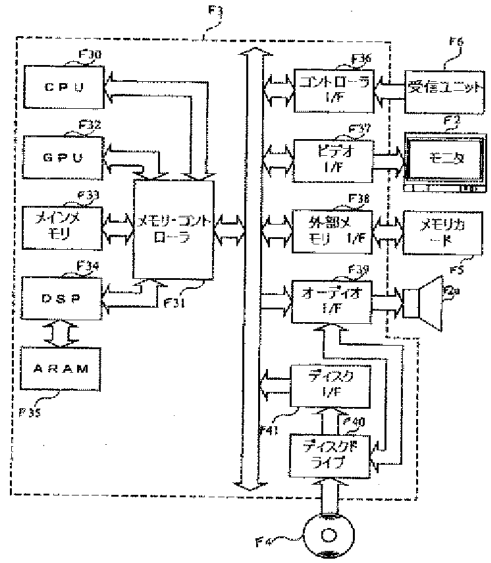
【図38B】

	左	中央	右
上	!	@	#
中央	$	%	^
下	&	*	(

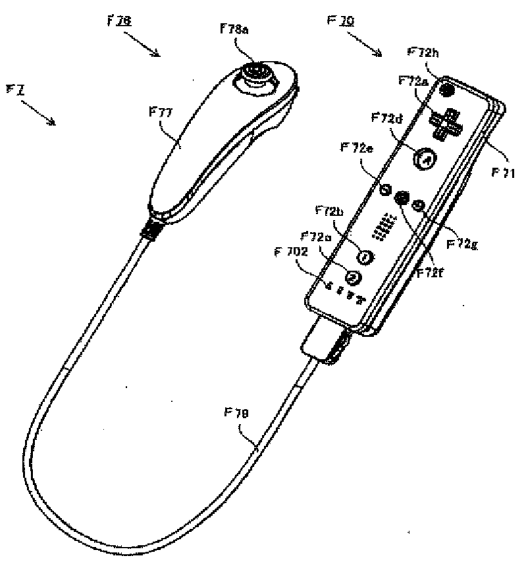
【図39】



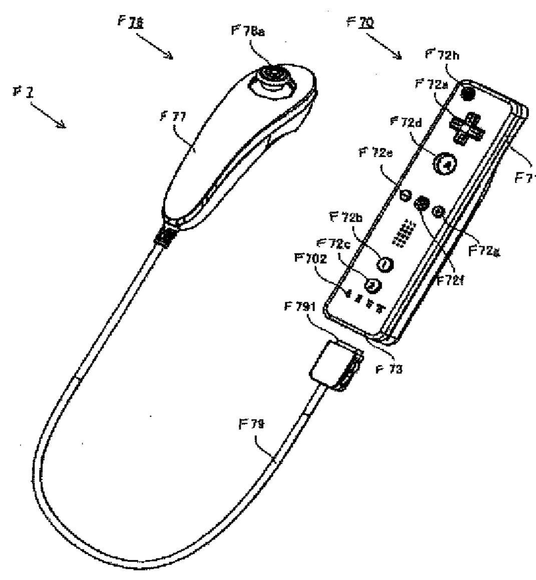
【図40】


F3
F30
CPU
F32
GPU
F33
メインメモリ
メモリ・コントローラ
F34
DSP
F31
ARAM
F35
F36
コントローラ I/F
F6
受信ユニット
F37
ビデオ I/F
F2
モニタ
F38
外部メモリ I/F
メモリカード
F5
F39
オーディオ I/F
F2a
ディスク I/F
F41
F40
ディスクドライブ
F4

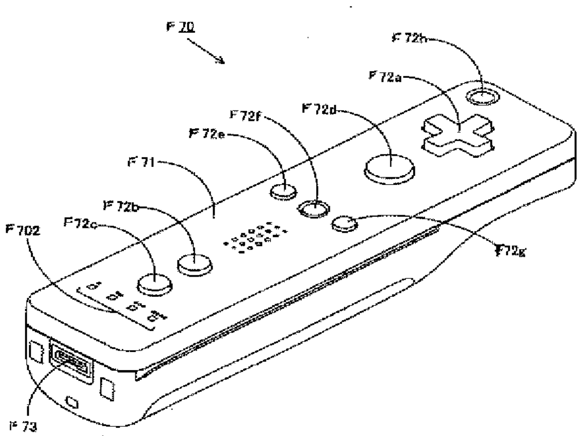
【図41】


F7
F78
F78a
F70
F72h
F72a
F77
F72d
F72e
F71
F72b
F72g
F72c
F702
F72f
F79

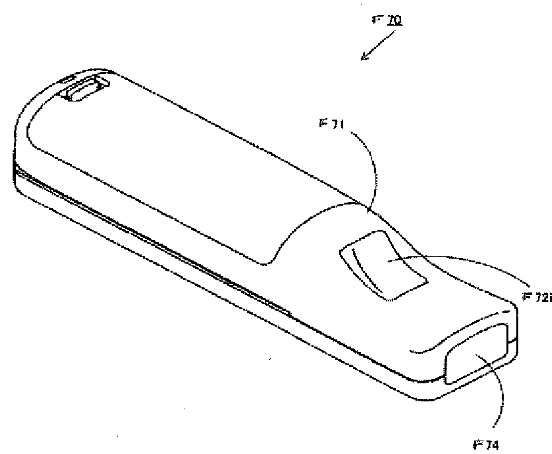
【図42】


F 7
F 76
F 78a
F 77
F 70
F 72h
F 72a
F 72d
F 72e
F 71
F 72b
F 72c
F 702
F 72g
F 72f
F 791
F 73
F 79

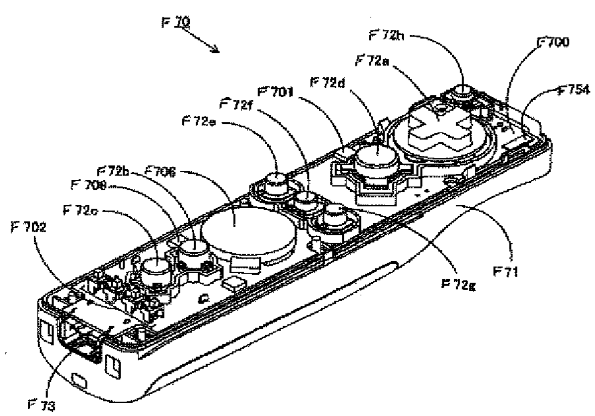
【図43】



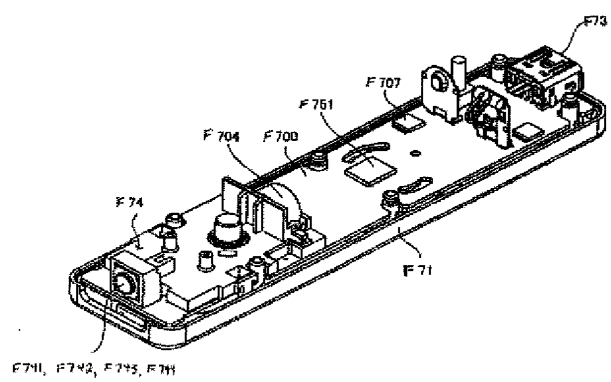
【図44】



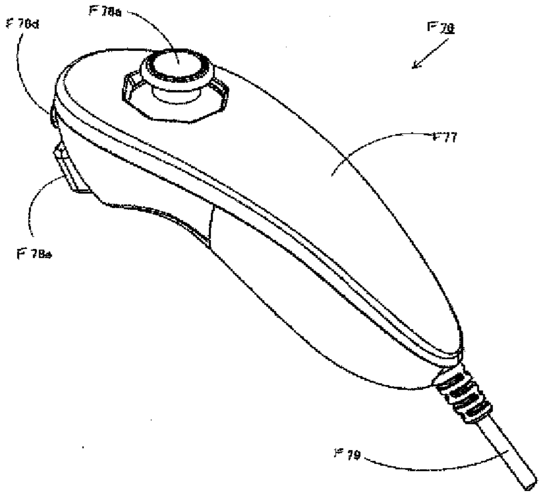
【図45】


F 70
F 72h
F 700
F 72a
F 754
F 701
F 72d
F 72f
F 72e
F 72b
F 706
F 708
F 72c
F 702
F 71
F 72g
F 73

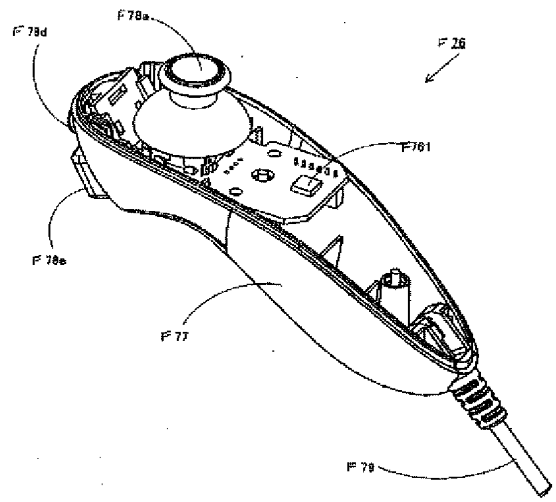
【図46】



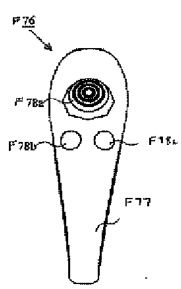
【図47】



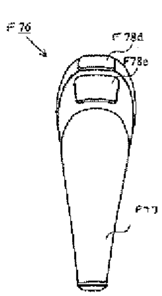
【図48】



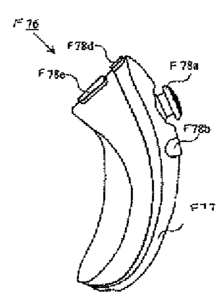
【図49A】



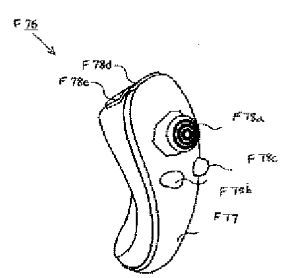
【図49B】



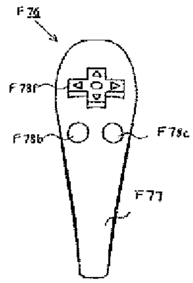
【図49C】



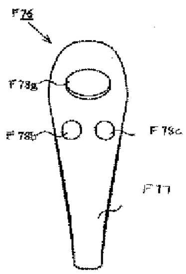
【図50】



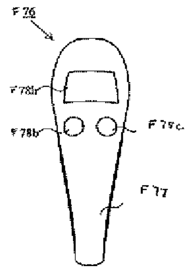
【図51】



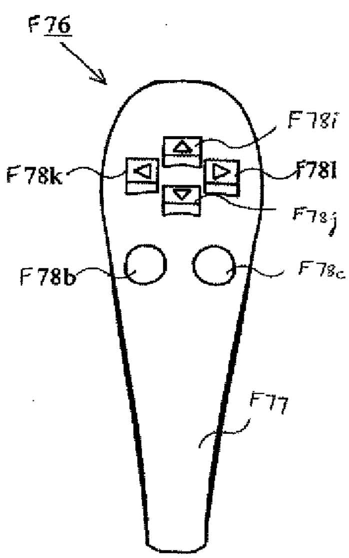
【図52】



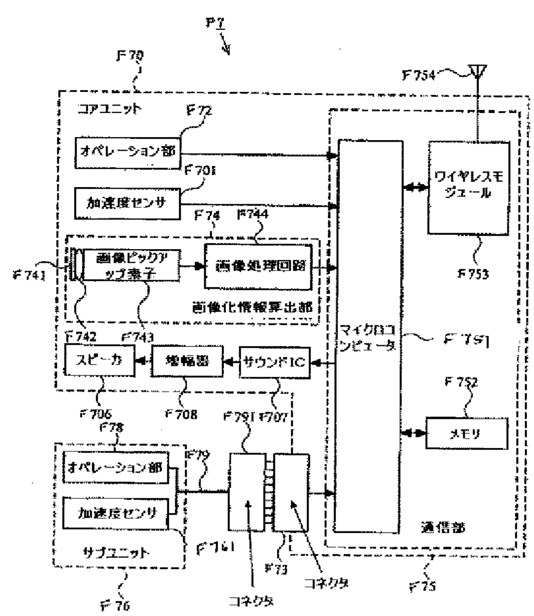
【図53】



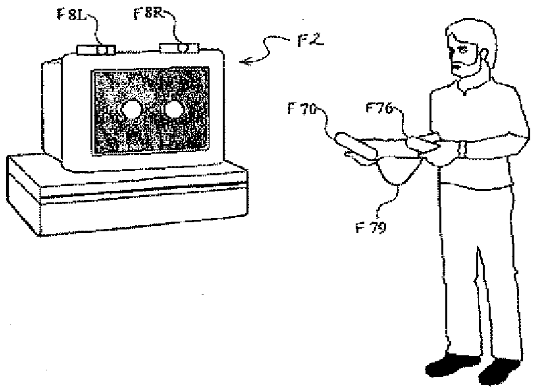
【図54】



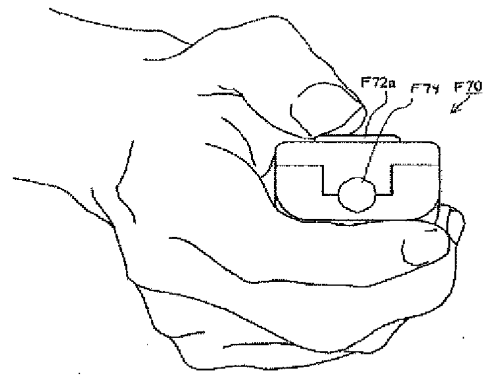
【図55】


P7
F70
コアユニット
F72
オペレーション部
F701
加速度センサ
F74
F744
F741
画像ピックアップ素子
画像処理回路
画像化情報算出部
F742
F743
スピーカ
増幅器
サウンドIC
F706
F708
F707
F754
ワイヤレスモジュール
F753
マイクロコンピュータ
F752
メモリ
通信部
F75
F78
オペレーション部
加速度センサ
サブユニット
F76
F79
F73
コネクタ
コネクタ

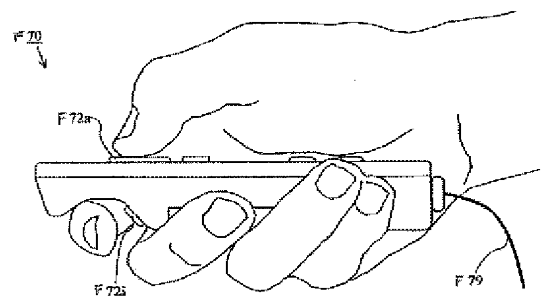
【図56】



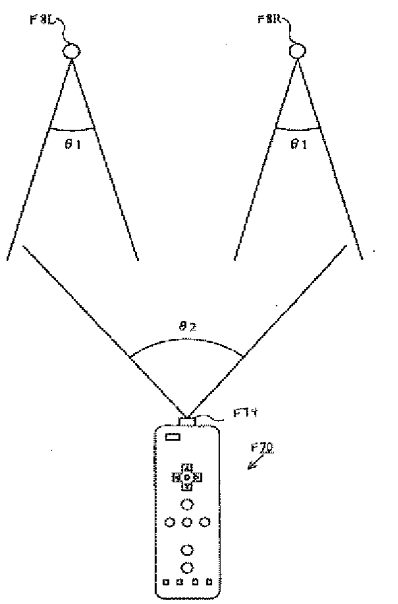
【図57】



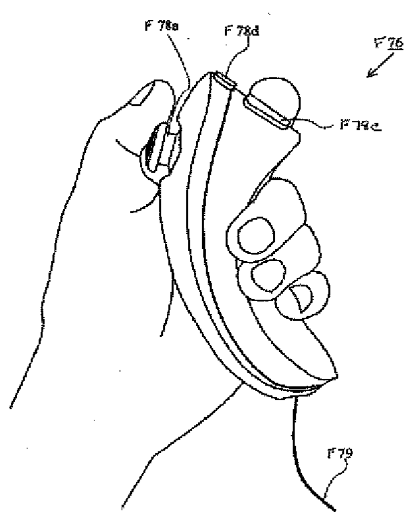
【図58】



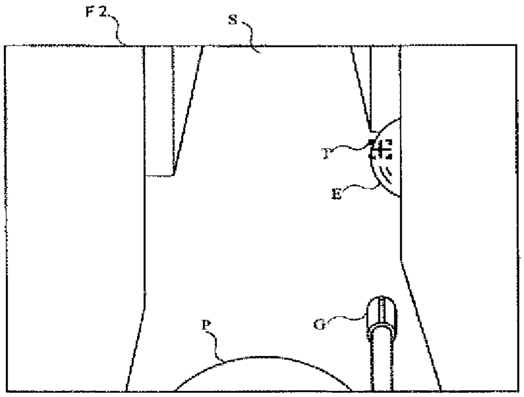
【図59】



【図60】



【図61】



【国際調査報告】

【図61】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/065000

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - A63F 9/24 (2006.04)
USPC - 463/36
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) - A63F 9/24 (2006.04)
USPC - 463/36

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2006/0019745 A1 (BENBRAHIM) 26 January 2006 (26.01.2006) entire document	12 --- 1-11, 13-20
Y	US 5,796,354 A (CARTABIANO et al) 18 August 1998 (18.08.1998) entire document	1-11, 13-20
Y	US 2003/0054868 A1 (PAULSEN et al) 20 March 2003 (20.03.2003) entire document	6-8
Y	US 2006/0287086 A1 (ZALEWSKI et al) 21 December 2006 (21.12.2006) entire document	14, 16

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

* 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 another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 August 2008	03 SEP 2008
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 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2005)

(81)指定国　AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MT,NL,NO,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,SV,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

特許法第６４条第２項第４号の規定により図面の一部または全部を不掲載とする。
（特許庁注：以下のものは登録商標）
１．ＰＥＮＴＩＵＭ
２．イーサネット
３．ＧＳＭ

(72)発明者　ラトニック，ハワード，ダブリュ
アメリカ合衆国　ニューヨーク州　１００２１，ニューヨーク，イースト・７１スト・ストリート　１１
(72)発明者　オルデルッキ，ディーン，ピー
アメリカ合衆国　コネティカット州　０６８８０，ウエストポート，マリオン・ロード　８
(72)発明者　ジェルマン，ジョフリー，エム
アメリカ合衆国　ニューヨーク州　１１２１７，ブルックリン，バークレイ・プレイス　１４，アパートメント　３
Ｆターム(参考) 2C001 AA13　CA05　CA09

STATE OF THE ART

In Arrival of 2 iPhones, 3 Lessons

By David Pogue

Sept. 17, 2013

We can draw three lessons from the arrival of Apple's two new iPhone models, the 5C and 5S.

LESSON 1 Apple may have set its own bar for innovation too high.

Year after year, Steve Jobs used to blow our minds with products we didn't know we wanted. Now, two years after his death, we still expect every new iPhone to clean our gutters, cook our popcorn and levitate. So when the hardware revisions are minor each year, we're disappointed.

And sure enough, after Apple showed off its two new iPhone models last week, its stock dropped. Analysts shrugged that they contain nothing "transformative." The blogger-haters had a field day.

The budget model, the new iPhone 5C, comes in five colors ($100 for the 16-gigabyte model with a two-year contract, $550 without). It's essentially identical to last year's iPhone 5, except that its back and sides are a single piece of plastic instead of metal and glass.

Actually, "plastic" isn't quite fair. The 5C's case is polycarbonate, lacquered like a glossy piano. Better yet, its back edges are curved for the first time since the iPhones of 2008. You can tell by touch which way it's facing in your pocket.

It's a terrific phone. The price is right. It will sell like hot cakes; the new iPhones go on sale Friday. But just sheathing last year's phone in shiny plastic isn't a stunning advance.

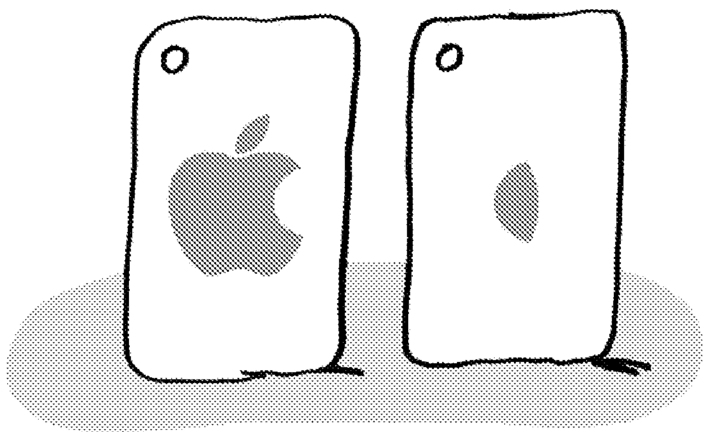
LESSON 2 The smartphone is mature.

The App Store filled a huge hole. Siri voice command answered a desperate need. And high-resolution Retina displays helped compensate for the tiny screen.

But today, every phone has that stuff; the big holes have been plugged. Maybe the age of annual mega-leaps is over.

The new 5S ($200 with contract, $650 without) looks exactly like last year's thin and gorgeous iPhone 5. You can now get it with its brushed aluminum body in dark gray (with black glass accents), silver (white accents) or a surprisingly classy-looking gold (white accents).

Apple says the 5S's chip is twice as fast as before. Nobody was exactly complaining about the iPhone's speed before, but, sure, it's plenty quick. Since it's a 64-bit chip, Apple says the graphics in 3-D video games look especially smooth and detailed.



Stuart Goldenberg

There's also a second chip devoted to tracking motion data from the phone's compass, gyroscope and tilt sensor. Apple says this coprocessor should save battery life when you use fitness tracking apps, because it can monitor your data all day long; the main chip, which requires six times as much power, can remain asleep.

Those are both fairly invisible changes, though.

The new camera will mean more to you. Its sensor is 15 percent bigger, and the individual light-detecting pixels are bigger. Take photos side-by-side with the iPhone 5S's predecessor, and the difference is immediately obvious; lowlight pictures are far better on the new phone. Clearer, brighter, better color.

The 5S also has two LED flashes — one pure white, one amber — that fire simultaneously. When mixed in the right balance, their light can match the color tone of your subject (moonlight, streetlights, fluorescents, whatever). Apple says this idea is a first in both phones and cameras.

It really works. Flash photos look much, much better. No longer will your loved ones' skin look either nuclear white or "Avatar" blue.

The 5S's camera also offers a burst mode (10 frames a second), 3X zooming during video capture, Instagram-style photo filters and truly wowing slow-motion video. (Weirdly, filtered photos and slo-mo videos don't survive the transfer to your computer, although you can send them by e-mail or text message.) Sample photos and videos accompany this column online.

The most heavily promoted feature is the 5S's fingerprint sensor, which, ingeniously, is built into the Home button. You push the Home button to wake the phone, leave your finger there another half second, and boom: you've unlocked a phone that nobody else can unlock, without the hassle of inputting the password. (And yes, a password is a hassle; half of smartphone users never bother setting one up.)

The best part is that it actually works — every single time, in my tests. It's nothing like the balky, infuriating fingerprint-reader efforts of earlier cellphones. It's genuinely awesome; the haters can go jump off a pier.

The 5S can also scan your fingerprint when you're buying books, music, apps and videos from Apple, saving you the password entry (although this, too, is buggy; Apple says a fix is due on Friday).

You can teach your iPhone 5S to recognize up to five fingerprints — all yours, yours and your spouse's, or whatever.

Apple says your fingerprint is stored only on your phone, encrypted and never transmitted or stored online. And using the fingerprint reader is optional; you can always use a regular password instead.

The sound quality of both new iPhones is excellent, whether up to your ear or filling your office with music. Apple says battery life is about 25 percent better than before; I've been getting nearly two days of moderate use on a charge.

So yes, Lesson 2 is that the speed of innovation seems to be slowing down, but don't let that depress you. Focus instead on the silver lining: you can keep your current phone longer without feeling obsolete quite so soon.

(Speaking of obsolescence: If you've held out upgrading since the iPhone 4 or 4S, remember that the new phones use Apple's new charging connector. It doesn't fit any existing charging cables, speaker docks or alarm clocks without a $35 adapter. Grrr.)

LESSON 3 If we're reaching a point of diminishing returns in hardware breakthroughs, the software breakthroughs are only just getting under way.

The new iPhones come with iOS 7, a redesigned operating system. You can also install it on recent iPhone, iPad and iPod Touch models.

This software looks *nothing* like the old iOS. It's all white and clean, almost barren. Its Home screen and dialogue boxes use thin fonts and a color palette of bright, light hues.

Above all, it completely abandons Apple's formerly favorite design principle, skeuomorphism, in which on-screen things depict real-world materials (lined yellow paper for Notes, leather binding for Calendar, wooden shelves for iBooks).

You might love this design, and you might loathe it. You also might get used to it. But in any case, iOS 7 is more efficient to navigate, because nothing on the screen is eye candy; everything is a button, so you spend less time hunting for things.

Furthermore, Apple did an insane amount of work on features. Some are big-ticket items like Siri, which responds faster, has a more realistic voice and understands new kinds of commands (including "Make the screen brighter" and "Turn on Bluetooth").

A supremely useful Control Center offers one-touch buttons to change important settings (thanks for the idea, Android!). AirDrop shoots pictures, maps, Web sites and other items to nearby iOS 7 gadgets, quickly and wirelessly.

You can read a full review of iOS 7 on my blog at nytimes.com/pogue. (A note: I have written a how-to manual to the iPhone and to iOS 7 for an independent publisher; it was neither commissioned by nor written in cooperation with Apple.)

Now, Apple's competition in the Android world is fierce and gaining; the competitors include phones that are equally beautiful (from HTC), phones that take spoken commands without your having to press a button (from Motorola) and phones in every conceivable screen size (Samsung).

But that doesn't mean that the iPhones have been *overtaken*. The iPhone's ecosystem is a deal-sweetening perk — the best apps; the best-stocked online stores for music and movies; smooth synchronizing of your calendars, addresses and even photos among Apple phones, tablets and Macs; and enough cases and accessories to reach from the landfill to the moon.

If you wanted to summarize all three of this week's lessons into a single final thesis, here it is: Apple still believes in superb design and tremendous polish. The iPhone is no longer the only smartphone that will keep you delighted for the length of your two-year contract — but it's still among the few that will.

Specification for RFID Air Interface

EPC™ Radio-Frequency Identity Protocols
Class-1 Generation-2 UHF RFID
Protocol for Communications at 860 MHz – 960 MHz
Version 1.2.0

Contents

Index of Figures

Index of Tables

Foreword

This Class-1 specification defines the base (or foundation) of four EPCglobal radio-frequency identification (RFID) protocols. The feature set for all four protocols is located at www.epcglobalinc.org/standards. The feature set for the Class-1 protocol is normative to this document.

The feature set for higher-class protocols is informative to this document.

Regardless of the class definitions, higher-class Tags shall not conflict with the operation of, nor degrade the performance of, Class-1 Tags located in the same RF environment.

Introduction

This specification defines the physical and logical requirements for a passive-backscatter, Interrogator-talks-first (ITF), radio-frequency identification (RFID) system operating in the 860 MHz – 960 MHz frequency range. The system comprises Interrogators, also known as Readers, and Tags, also known as Labels.

An Interrogator transmits information to a Tag by modulating an RF signal in the 860 MHz – 960 MHz frequency range. The Tag receives both information and operating energy from this RF signal. Tags are passive, meaning that they receive all of their operating energy from the Interrogator's RF waveform.

An Interrogator receives information from a Tag by transmitting a continuous-wave (CW) RF signal to the Tag; the Tag responds by modulating the reflection coefficient of its antenna, thereby backscattering an information signal to the Interrogator. The system is ITF, meaning that a Tag modulates its antenna reflection coefficient with an information signal only after being directed to do so by an Interrogator.

Interrogators and Tags are not required to talk simultaneously; rather, communications are half-duplex, meaning that Interrogators talk and Tags listen, or vice versa.

1. Scope

This document specifies:

- Physical interactions (the signaling layer of the communication link) between Interrogators and Tags,
- Interrogator and Tag operating procedures and commands, and
- The collision arbitration scheme used to identify a specific Tag in a multiple-Tag environment.

2. Conformance

2.1 Claiming conformance

A device shall not claim conformance with this specification unless the device complies with

a) all clauses in this specification (except those marked as optional), and

b) the conformance document associated with this specification, and,

c) all local radio regulations.

Conformance may also require a license from the owner of any intellectual property utilized by said device.

2.2 General conformance requirements

2.2.1 Interrogators

To conform to this specification, an Interrogator shall:

- Meet the requirements of this specification,
- Implement the mandatory commands defined in this specification,
- Modulate/transmit and receive/demodulate a sufficient set of the electrical signals defined in the signaling layer of this specification to communicate with conformant Tags, and
- Conform to all local radio regulations.

To conform to this specification, an Interrogator may:

- Implement any subset of the optional commands defined in this specification, and
- Implement any proprietary and/or custom commands in conformance with this specification.

To conform to this specification, an Interrogator shall not:

- Implement any command that conflicts with this specification, or
- Require using an optional, proprietary, or custom command to meet the requirements of this specification.

2.2.2 Tags

To conform to this specification, a Tag shall:

- Meet the requirements of this specification,
- Operate over the frequency range from 860 – 960 MHz, inclusive,
- Implement the mandatory commands defined in this specification,
- Modulate a backscatter signal only after receiving the requisite command from an Interrogator, and
- Conform to all local radio regulations.

To conform to this specification, a Tag may:

- Implement any subset of the optional commands defined in this specification, and
- Implement any proprietary and/or custom commands as defined in Proprietary commands and 2.3.4, respectively.

To conform to this specification, a Tag shall not:

- Implement any command that conflicts with this specification,

- Require using an optional, proprietary, or custom command to meet the requirements of this specification, or
- Modulate a backscatter signal unless commanded to do so by an Interrogator using the signaling layer defined in this specification.

2.3 Command structure and extensibility

This specification allows four command types: (1) mandatory, (2) optional, (3) proprietary, and (4) custom. Subclause 6.3.2.11 and Table 6.18 define the structure of the command codes used by Interrogators and Tags for each of the four types, as well as the availability of future extensions.

All commands defined by this specification are either mandatory or optional.

Proprietary or custom commands are vendor-defined.

2.3.1 Mandatory commands

Conforming Tags shall support all mandatory commands. Conforming Interrogators shall support all mandatory commands.

2.3.2 Optional commands

Conforming Tags may or may not support optional commands. Conforming Interrogators may or may not support optional commands. If a Tag or an Interrogator implements an optional command, it shall implement it in the manner specified in this specification.

2.3.3 Proprietary commands

Proprietary commands may be enabled in conformance with this specification, but are not specified herein. All proprietary commands shall be capable of being permanently disabled. Proprietary commands are intended for manufacturing purposes and shall not be used in field-deployed RFID systems.

2.3.4 Custom commands

Custom commands may be enabled in conformance with this specification, but are not specified herein. An Interrogator shall issue a custom command only after (1) singulating a Tag, and (2) reading (or having prior knowledge of) the Tag manufacturer's identification in the Tag's TID memory. An Interrogator shall use a custom command only in accordance with the specifications of the Tag manufacturer identified in the Tag ID. A custom command shall not solely duplicate the functionality of any mandatory or optional command defined in this specification by a different method.

2.4 Reserved for Future Use (RFU)

This specification denotes some Tag memory addresses, Interrogator command codes, and bit fields within some Interrogator commands as being RFU. In general, EPCglobal is reserving these RFU values for use in higher-class specifications. Under some circumstances EPCglobal may permit another standards body or related organization to use one or more of these RFU values for standardization purposes. In these circumstances the permitted body shall keep EPCglobal appraised, in a timely manner, of its use or potential use of these RFU values. Third parties, including but not limited to solution providers and end users, shall not use these RFU values for proprietary purposes.

Bit E_h of XPC_W1 (EPC memory location 211_h – see 6.3.2.1.2.5) shall be reserved for use as a protocol functionality indicator. If another standards-setting body authorizes any physical or logical Tag operation that is incompatible with this EPCglobal specification then that standards-setting body shall employ bit E_h as follows:

- If bit E_h of XPC_W1 contains a logical 0 then the application is referred to as an EPCglobal™ Application and a Tag shall follow the physical and logical requirements specified by this EPC™ Class-1 Generation-2 UHF RFID Protocol. If bit E_h contains a logical 1 then the application is referred to as a non-EPCglobal™ Application and a Tag may follow the physical and logical requirements specified by a non-EPC™ Protocol. The default value for bit E_h shall be 0.

3. Normative references

The following referenced documents are indispensable to the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition (including any amendments) applies.

EPCglobal™: *EPC™ Tag Data Standards*

EPCglobal™ (2004): *FMCG RFID Physical Requirements Document*

EPCglobal™ (2007): *ILT JRG Protocol Requirements Document V1.2.3*

European Telecommunications Standards Institute (ETSI), EN 300 220 (all parts): *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW*

European Telecommunications Standards Institute (ETSI), EN 302 208: *Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 1 – Technical characteristics and test methods*

European Telecommunications Standards Institute (ETSI), EN 302 208: *Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 2 – Harmonized EN under article 3.2 of the R&TTE directive*

ISO/IEC Directives, Part 2: *Rules for the structure and drafting of International Standards*

ISO/IEC 3309: *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure*

ISO/IEC 15961: *Information technology, Automatic identification and data capture – Radio frequency identification (RFID) for item management – Data protocol: application interface*

ISO/IEC 15962: *Information technology, Automatic identification and data capture techniques – Radio frequency identification (RFID) for item management – Data protocol: data encoding rules and logical memory functions*

ISO/IEC 15963: *Information technology — Radiofrequency identification for item management — Unique identification for RF tags*

ISO/IEC 18000-1: *Information technology — Radio frequency identification for item management — Part 1: Reference architecture and definition of parameters to be standardized*

ISO/IEC 18000-6: *Information technology automatic identification and data capture techniques — Radio frequency identification for item management air interface — Part 6: Parameters for air interface communications at 860–960 MHz*

ISO/IEC 19762: *Information technology AIDC techniques – Harmonized vocabulary – Part 3: radio-frequency identification (RFID)*

U.S. Code of Federal Regulations (CFR), Title 47, Chapter I, Part 15: *Radio-frequency devices, U.S. Federal Communications Commission*

4. Terms and definitions

The principal terms and definitions used in this specification are described in ISO/IEC 19762.

4.1 Additional terms and definitions

Terms and definitions specific to this document that supersede any normative references are as follows:

- **Air interface**

  The complete communication link between an Interrogator and a Tag including the physical layer, collision arbitration algorithm, command and response structure, and data-coding methodology.

- **Command set**

  The set of commands used to explore and modify a Tag population.

- **Continuous wave**

  Typically a sinusoid at a given frequency, but more generally any Interrogator waveform suitable for powering a passive Tag without amplitude and/or phase modulation of sufficient magnitude to be interpreted by a Tag as transmitted data.

- **Cover-coding**

  A method by which an Interrogator obscures information that it is transmitting to a Tag. To cover-code data or a password, an Interrogator first requests a random number from the Tag. The Interrogator then performs a bit-wise EXOR of the data or password with this random number, and transmits the cover-coded (also called ciphertext) string to the Tag. The Tag uncovers the data or password by performing a bit-wise EXOR of the received cover-coded string with the original random number.

- **Dense-Interrogator environment**

  An operating environment (defined below) within which most or all of the available channels are occupied by active Interrogators (for example, 25 active Interrogators operating in 25 available channels).

- **Dense-Interrogator mode**

  A set of Interrogator-to-Tag and Tag-to-Interrogator signaling parameters used in dense-Interrogator environments.

- **Extended temperature range**

  –40 °C to +65 °C (see nominal temperature range).

- **EPCglobal™ Application**

  An application whose usage denotes an acceptance of EPCglobal™ standards and policies (see non-EPCglobal™ Application).

- **Full-duplex communications**

  A communications channel that carries data in both directions at once. See also half-duplex communications.

- **Half-duplex communications**

  A communications channel that carries data in one direction at a time rather than in both directions at once. See also full-duplex communications.

- **Inventoried flag**

  A flag that indicates whether a Tag may respond to an Interrogator. Tags maintain a separate **inventoried** flag for each of four sessions; each flag has symmetric *A* and *B* values. Within any given session, Interrogators typically inventory Tags from *A* to *B* followed by a re-inventory of Tags from *B* back to *A* (or vice versa).

- **Inventory round**

  The period initiated by a *Query* command and terminated by either a subsequent *Query* command (which also starts a new inventory round) or a *Select* command.

- **Multiple-Interrogator environment**

  An operating environment (defined below) within which a modest number of the available channels are occupied by active Interrogators (for example, 5 active Interrogators operating in 25 available channels).

- **Nominal temperature range**

  –25 °C to +40 °C (see extended temperature range).

- **Non-EPCglobal™ Application**

  An application whose usage does not denote an acceptance of EPCglobal™ standards and policies (see EPCglobal™ Application).

- **Operating environment**

  A region within which an Interrogator's RF transmissions are attenuated by less than 90dB. In free space, the operating environment is a sphere whose radius is approximately 1000m, with the Interrogator located at the center. In a building or other enclosure, the size and shape of the operating environment depends on factors such as the material properties and shape of the building, and may be less than 1000m in certain directions and greater than 1000m in other directions.

- **Operating procedure**

  Collectively, the set of functions and commands used by an Interrogator to identify and modify Tags. (Also known as the *Tag-identification layer*.)

- **PacketCRC**

  A 16-bit cyclic-redundancy check (CRC) code that a Tag may dynamically calculate over its PC, optional XPC, and EPC and backscatter during inventory (see StoredCRC).

- **PacketPC**

  Protocol-control information that a Tag with nonzero-valued XI dynamically calculates and backscatters during inventory (see StoredPC).

- **Passive Tag (or passive Label)**

  A Tag (or Label) whose transceiver is powered by the RF field.

- **Permalock or Permalocked**

  A memory location whose lock status is unchangeable (i.e. the memory location is permanently locked or permanently unlocked) except by recommissioning the Tag is said to be permalocked.

- **Persistent memory or persistent flag**

  A memory or flag value whose state is maintained during a brief loss of Tag power.

- **Physical layer**

  The data coding and modulation waveforms used in Interrogator-to-Tag and Tag-to-Interrogator signaling.

- **Protocol**

  Collectively, a physical layer and a Tag-identification layer specification.

- ***Q***

  A parameter that an Interrogator uses to regulate the probability of Tag response. An Interrogator commands Tags in an inventory round to load a *Q*-bit random (or pseudo-random) number into their slot counter; the Interrogator may also command Tags to decrement their slot counter. Tags reply when the value in their slot counter (i.e. their slot – see below) is zero. *Q* is an integer in the range (0,15); the corresponding Tag-response probabilities range from $2^0 = 1$ to $2^{-15} = 0.000031$.

- **Random-slotted collision arbitration**

  A collision-arbitration algorithm where Tags load a random (or pseudo-random) number into a slot counter, decrement this slot counter based on Interrogator commands, and reply to the Interrogator when their slot counter reaches zero.

- **Recommissioning**

  A significant altering of a Tag's functionality and/or memory contents, as commanded by an Interrogator, typically in response to a change in the Tag's usage model or purpose.

- **Session**

  An inventory process comprising an Interrogator and an associated Tag population. An Interrogator chooses one of four sessions and inventories Tags within that session. The Interrogator and associated Tag population operate in one and only one session for the duration of an inventory round (defined above). For each session, Tags maintain a corresponding **inventoried** flag. Sessions allow Tags to keep track of their inventoried status separately for each of four possible time-interleaved inventory processes, using an independent **inventoried** flag for each process.

- **Single-Interrogator environment**
  An operating environment (defined above) within which there is a single active Interrogator at any given time.
- **Singulation**
  Identifying an individual Tag in a multiple-Tag environment.
- **Slot**
  Slot corresponds to the point in an inventory round at which a Tag may respond. Slot is the value output by a Tag's slot counter; Tags reply when their slot (i.e. the value in their slot counter) is zero. See also *Q* (above).
- **StoredCRC**
  A 16-bit cyclic-redundancy check (CRC) code that a Tag calculates over its StoredPC and EPC and stores in EPC memory at power-up, and may backscatter during inventory (see PacketCRC).
- **StoredPC**
  Protocol-control information stored in EPC memory that a Tag with zero-valued XI backscatters during inventory (see PacketPC).
- **Tag-identification layer**
  Collectively, the set of functions and commands used by an Interrogator to identify and modify Tags (also known as the *operating procedure*).
- **Tari**
  Reference time interval for a data-0 in Interrogator-to-Tag signaling. The mnemonic "Tari" derives from the ISO/IEC 18000-6 (part A) specification, in which Tari is an abbreviation for Type A Reference Interval.

5. Symbols, abbreviated terms, and notation

The principal symbols and abbreviated terms used in this specification are detailed in ISO/IEC 19762, *Information technology AIDC techniques – vocabulary*. Symbols, abbreviated terms, and notation specific to this document are as follows:

5.1 Symbols

BLF	Backscatter-link frequency (BLF = $1/T_{pri}$ = DR/TRcal)
DR	Divide ratio
FT	Frequency tolerance
M	Number of subcarrier cycles per symbol
M_h	RF signal envelope ripple (overshoot)
M_{hh}	FHSS signal envelope ripple (overshoot)
M_l	RF signal envelope ripple (undershoot)
M_{hl}	FHSS signal envelope ripple (undershoot)
M_s	RF signal level when OFF
M_{hs}	FHSS signal level during a hop
Q	Slot-count parameter
R=>T	Interrogator-to-Tag
RTcal	Interrogator-to-Tag calibration symbol
T_1	Time from Interrogator transmission to Tag response
T_2	Time from Tag response to Interrogator transmission
T_3	Time an Interrogator waits, after T_1, before it issues another command
T_4	Minimum time between Interrogator commands

T_f or $T_{f,10\text{-}90\%}$	RF signal envelope fall time
T_{hf}	FHSS signal envelope fall time
T_{hr}	FHSS signal envelope rise time
T_{hs}	Time for an FHSS signal to settle to within a specified percentage of its final value
T_{pri}	Backscatter-link pulse-repetition interval (T_{pri} = 1/BLF = TRcal/DR)
T_r or $T_{r,10\text{-}90\%}$	RF signal envelope rise time
T_s	Time for an RF signal to settle to within a specified percentage of its final value
T=>R	Tag-to-Interrogator
TRcal	Tag-to-Interrogator calibration symbol
x_{fp}	floating-point value
$xxxx_2$	binary notation
$xxxx_h$	hexadecimal notation

5.2 Abbreviated terms

AFI	Application family identifier
AM	Amplitude modulation
ASK	Amplitude shift keying
CEPT	Conference of European Posts and Telecommunications
Ciphertext	Information that is cover-coded
CRC	Cyclic redundancy check
CW	Continuous wave
dBch	Decibels referenced to the integrated power in the reference channel
DSB	Double sideband
DSB-ASK	Double-sideband amplitude-shift keying
EPC	Electronic product code
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FDM	Frequency-Division Multiplexing
FHSS	Frequency-hopping spread spectrum
Handle	16-bit Tag-authentication number
NSI	Numbering system identifier
PC	Protocol control
PIE	Pulse-interval encoding
Pivot	Decision threshold differentiating an R=>T data-0 symbol from a data-1 symbol
Plaintext	Information that is not cover-coded
ppm	Parts-per-million
PSK	Phase shift keying or phase shift keyed
PR-ASK	Phase-reversal amplitude shift keying
RF	Radio frequency
RFID	Radio-frequency identification
RFU	Reserved for future use
RN16	16-bit random or pseudo-random number
RNG	Random or pseudo-random number generator
ITF	Interrogator talks first (reader talks first)
SSB	Single sideband
SSB-ASK	Single-sideband amplitude-shift keying
TDM	Time-division multiplexing or time-division multiplexed (as appropriate)
Tag ID	Tag-identification or Tag identifier, depending on context
Word	16 bits
UMI	User-memory indicator
XI	XPC_W1 indicator

XPC	Extended protocol control
XPC_W1	XPC word 1
XPC_W2	XPC word 2
XEB	XPC extension bit

5.3 Notation

This specification uses the following notational conventions:

- States and flags are denoted in bold. Example: **ready**.
- Commands are denoted in italics. Variables are also denoted in italics. Where there might be confusion between commands and variables, this specification will make an explicit statement. Example: *Query*.
- Command parameters are underlined. Example: <u>Pointer</u>.
- For logical negation, labels are preceded by '~'. Example: If **flag** is true, then **~flag** is false.
- The symbol, R=>T, refers to commands or signaling from an Interrogator to a Tag (Reader-to-Tag).
- The symbol, T=>R, refers to commands or signaling from a Tag to an Interrogator (Tag-to-Reader).

6. Protocol requirements

6.1 Protocol overview

6.1.1 Physical layer

An Interrogator sends information to one or more Tags by modulating an RF carrier using double-sideband amplitude shift keying (DSB-ASK), single-sideband amplitude shift keying (SSB-ASK), or phase-reversal amplitude shift keying (PR-ASK) using a pulse-interval encoding (PIE) format. Tags receive their operating energy from this same modulated RF carrier.

An Interrogator receives information from a Tag by transmitting an unmodulated RF carrier and listening for a backscattered reply. Tags communicate information by backscatter modulating the amplitude and/or phase of the RF carrier. The encoding format, selected in response to Interrogator commands, is either FM0 or Miller-modulated subcarrier. The communications link between Interrogators and Tags is half-duplex, meaning that Tags shall not be required to demodulate Interrogator commands while backscattering. A Tag shall not respond to a mandatory or optional command using full-duplex communications.

6.1.2 Tag-identification layer

An Interrogator manages Tag populations using three basic operations:

a) **Select.** The operation of choosing a Tag population for inventory and access. A *Select* command may be applied successively to select a particular Tag population based on user-specified criteria. This operation is analogous to selecting records from a database.

b) **Inventory.** The operation of identifying Tags. An Interrogator begins an inventory round by transmitting a *Query* command in one of four sessions. One or more Tags may reply. The Interrogator detects a single Tag reply and requests the PC/XPC word(s), EPC, and CRC from the Tag. Inventory comprises multiple commands. An inventory round operates in one and only one session at a time.

c) **Access.** The operation of communicating with (reading from and/or writing to) a Tag. An individual Tag must be uniquely identified prior to access. Access comprises multiple commands, some of which employ one-time-pad based cover-coding of the R=>T link.

6.2 Protocol parameters

6.2.1 Signaling – Physical and media access control (MAC) parameters

Table 6.1 and Table 6.2 provide an overview of parameters for R=>T and T=>R communications according to this specification; for detailed requirements refer to the referenced Subclause. For those parameters that do not apply to or are not used in this specification, the notation “N/A” shall indicate that the parameter is “Not Applicable”.

Table 6.1 – Interrogator-to-Tag (R=>T) communications

Ref.	Parameter Name	Description	Subclause
Int:1	Operating Frequency Range	860 – 960 MHz, as required by local regulations	6.3.1.1
Int:1a	Default Operating Frequency	Determined by local radio regulations and by the radio-frequency environment at the time of the communication	6.3.1.1
Int:1b	Operating Channels (spread-spectrum systems)	In accordance with local regulations; if the channelization is unregulated, then as specified	6.3.1.2.10, Annex G
Int:1c	Operating Frequency Accuracy	As specified	6.3.1.2.1
Int:1d	Frequency Hop Rate (frequency-hopping [FHSS] systems)	In accordance with local regulations	6.3.1.2.9
Int:1e	Frequency Hop Sequence (frequency-hopping [FHSS] systems)	In accordance with local regulations	6.3.1.2.9
Int:2	Occupied Channel Bandwidth	In accordance with local regulations	N/A
Int:2a	Minimum Receiver Bandwidth	In accordance with local regulations	N/A
Int:3	Interrogator Transmit Maximum EIRP	In accordance with local regulations	N/A
Int:4	Interrogator Transmit Spurious Emissions	As specified; tighter emission limits may be imposed by local regulations	6.3.1.2.11
Int:4a	Interrogator Transmit Spurious Emissions, In-Band (spread-spectrum systems)	As specified; tighter emission limits may be imposed by local regulations	6.3.1.2.11
Int:4b	Interrogator Transmit Spurious Emissions, Out-of-Band	As specified; tighter emission limits may be imposed by local regulations	6.3.1.2.11
Int:5	Interrogator Transmitter Spectrum Mask	As specified; tighter emission limits may be imposed by local regulations	Figure 6.6, Figure 6.7
Int:6	Timing	As specified	6.3.1.6, Figure 6.16, Table 6.13
Int:6a	Transmit-to-Receive Turn-Around Time	MAX(RTcal,10T_{pri}) nominal	6.3.1.6, Figure 6.16, Table 6.13
Int:6b	Receive-to-Transmit Turn-Around Time	3T_{pri} minimum; 20T_{pri} maximum when Tag is in **reply** & **acknowledged** states; no limit otherwise	6.3.1.6, Figure 6.16, Table 6.13
Int:6c	Dwell Time or Interrogator Transmit Power-On Ramp	1500 µs, maximum settling time	6.3.1.2.6, Figure 6.3, Table 6.6
Int:6d	Decay Time or Interrogator Transmit Power-Down Ramp	500 µs, maximum	6.3.1.2.7 Figure 6.3, Table 6.7
Int:7	Modulation	DSB-ASK, SSB-ASK, or PR-ASK	6.3.1.2.2
Int:7a	Spreading Sequence (direct-sequence [DSSS] systems)	N/A	N/A
Int:7b	Chip Rate (spread-spectrum systems)	N/A	N/A
Int:7c	Chip Rate Accuracy (spread-spectrum systems)	N/A	N/A
Int:7d	Modulation Depth	90% nominal	6.3.1.2.5, Figure 6.2, Table 6.5
Int:7e	Duty Cycle	48% – 82.3% (time the waveform is high)	6.3.1.2.3, Figure 6.1, Table 6.5
Int:7f	FM Deviation	N/A	N/A
Int:8	Data Coding	PIE	6.3.1.2.3, Figure 6.1
Int:9	Bit Rate	26.7 kbps to 128 kbps (assuming equiprobable data)	6.3.1.2.4
Int:9a	Bit Rate Accuracy	+/– 1%, minimum	6.3.1.2.4
Int:10	Interrogator Transmit Modulation Accuracy	As specified	6.3.1.2.4

Ref.	Parameter Name	Description	Subclause
Int:11	Preamble	Required	6.3.1.2.8
Int:11a	Preamble Length	As specified	6.3.1.2.8
Int:11b	Preamble Waveform(s)	As specified	Figure 6.4
Int:11c	Bit Sync Sequence	None	N/A
Int:11d	Frame Sync Sequence	Required	6.3.1.2.8, Figure 6.4
Int:12	Scrambling (spread-spectrum systems)	N/A	N/A
Int:13	Bit Transmission Order	MSB is transmitted first	6.3.1.4
Int:14	Wake-up Process	As specified	6.3.1.3.4
Int:15	Polarization	Not specified	N/A

Table 6.2 – Tag-to-Interrogator (T=>R) communications

Ref.	Parameter Name	Description	Subclause
Tag:1	Operating Frequency Range	860 – 960 MHz, inclusive	6.3.1.1
Tag:1a	Default Operating Frequency	Tags respond to Interrogator signals that satisfy Int:1a	6.3.1.1
Tag:1b	Operating Channels (spread-spectrum systems)	Tags respond to Interrogator signals that satisfy Int:1b	6.3.1.2.10
Tag:1c	Operating Frequency Accuracy	As specified	6.3.1.3.3 Table 6.9
Tag:1d	Frequency Hop Rate (frequency-hopping [FHSS] systems)	Tags respond to Interrogator signals that satisfy Int:1d	6.3.1.2.9
Tag:1e	Frequency Hop Sequence (frequency-hopping [FHSS] systems)	Tags respond to Interrogator signals that satisfy Int:1e	6.3.1.2.9
Tag:2	Occupied Channel Bandwidth	In accordance with local regulations	N/A
Tag:3	Transmit Maximum EIRP	In accordance with local regulations	N/A
Tag:4	Transmit Spurious Emissions	In accordance with local regulations	N/A
Tag:4a	Transmit Spurious Emissions, In-Band (spread-spectrum systems)	In accordance with local regulations	N/A
Tag:4b	Transmit Spurious Emissions, Out-of-Band	In accordance with local regulations	N/A
Tag:5	Transmit Spectrum Mask	In accordance with local regulations	N/A
Tag:6a	Transmit-to-Receive Turn-Around Time	$3T_{pri}$ minimum, $32T_{pri}$ maximum in **reply** & **acknowledged** states; no limit otherwise	6.3.1.6, Figure 6.16, Table 6.13
Tag:6b	Receive-to-Transmit Turn-Around Time	MAX(RTcal,$10T_{pri}$) nominal	6.3.1.6, Figure 6.16, Table 6.13
Tag:6c	Dwell Time or Transmit Power-On Ramp	Receive commands 1.5 ms after power-up	6.3.1.3.4
Tag:6d	Decay Time or Transmit Power-Down Ramp	N/A	N/A

Ref.	Parameter Name	Description	Subclause
Tag:7	Modulation	ASK and/or PSK modulation (selected by Tag)	6.3.1.3.1
Tag:7a	Spreading Sequence (direct sequence [DSSS] systems)	N/A	N/A
Tag:7b	Chip Rate (spread-spectrum systems)	N/A	N/A
Tag:7c	Chip Rate Accuracy (spread-spectrum systems)	N/A	N/A
Tag:7d	On-Off Ratio	Not specified	N/A
Tag:7e	Subcarrier Frequency	40 kHz to 640 kHz	6.3.1.3.3 Table 6.9
Tag:7f	Subcarrier Frequency Accuracy	As specified	Table 6.9
Tag:7g	Subcarrier Modulation	Miller, at the data rate	6.3.1.3.2.3, Figure 6.13
Tag:7h	Duty Cycle	FM0: 50%, nominal Subcarrier: 50%, nominal	6.3.1.3.2.1, 6.3.1.3.2.3
Tag:7I	FM Deviation	N/A	N/A
Tag:8	Data Coding	Baseband FM0 or Miller-modulated subcarrier (selected by the Interrogator)	6.3.1.3.2
Tag:9	Bit Rate	FM0: 40 kbps to 640 kbps Subcarrier modulated: 5 kbps to 320 kbps	6.3.1.3.3, Table 6.9, Table 6.10
Tag:9a	Bit Rate Accuracy	Same as Subcarrier Frequency Accuracy; see Tag:7f	6.3.1.3.3, Table 6.9, Table 6.10
Tag:10	Tag Transmit Modulation Accuracy (frequency-hopping [FHSS] systems	N/A	N/A
Tag:11	Preamble	Required	6.3.1.3.2.2, 6.3.1.3.2.4
Tag:11a	Preamble Length	As specified	Figure 6.11, Figure 6.15
Tag:11b	Preamble Waveform	As specified	Figure 6.11, Figure 6.15
Tag:11c	Bit-Sync Sequence	None	N/A
Tag:11d	Frame-Sync Sequence	None	N/A
Tag:12	Scrambling (spread-spectrum systems)	N/A	N/A
Tag:13	Bit Transmission Order	MSB is transmitted first	6.3.1.4
Tag:14	Reserved	Deliberately left blank	N/A
Tag:15	Polarization	Tag dependent; not specified by this document	N/A
Tag:16	Minimum Tag Receiver Bandwidth	Tag dependent; not specified by this document.	N/A

6.2.2 Logical – Operating procedure parameters

Table 6.3 and Table 6.4 identify and describe parameters used by an Interrogator during the selection, inventory, and access of Tags according to this specification. For those parameters that do not apply to or are not used in this specification, the notation "N/A" shall indicate that the parameter is "Not Applicable".

Table 6.3 – Tag inventory and access parameters

Ref.	Parameter Name	Description	Subclause
P:1	Who Talks First	Interrogator	6.3
P:2	Tag Addressing Capability	As specified	6.3.2.1
P:3	Tag EPC	Contained in Tag memory	6.3.2.1
P:3a	EPC Length	As specified	6.3.2.1.2.2
P:3b	EPC Format	NSI < 100_h: As specified in EPCglobal™ Tag Data Standards, NSI ≥ 100_h: As specified in ISO/IEC 15961	6.3.2.1.2.2 6.3.2.1.2.3 6.3.2.1.2.4
P:4	Read size	Multiples of 16 bits	6.3.2.11.3.2 Table 6.33
P:5	Write Size	Multiples of 16 bits	6.3.2.11.3.3, Table 6.35, 6.3.2.11.3.7, Table 6.46
P:6	Read Transaction Time	Varied with R=>T and T=>R link rate and number of bits being read	6.3.2.11.3.2
P:7	Write Transaction Time	20 ms (maximum) after end of *Write* command	6.3.2.11.3.3, 6.3.2.11.3.7, Figure 6.22
P:8	Error Detection	Interrogator-to-Tag: *Select* command: 16-bit CRC *Query* command: 5-bit CRC Other Inventory commands: Command length Access commands: 16-bit CRC Tag-to-Interrogator: PC/XPC, EPC: 16-bit CRC RN16: None or 16-bit CRC (varies by command) handle: 16-bit CRC All other: 16-bit CRC	6.3.2.11 and its subsections
P:9	Error Correction	None	N/A
P:10	Memory Size	Tag dependent, extensible (size is neither limited nor specified by this document)	N/A
P:11	Command Structure and Extensibility	As specified	Table 6.18

Table 6.4 – Collision management parameters

Ref.	Parameter Name	Description	Subclause
A:1	Type (Probabilistic or Deterministic)	Probabilistic	6.3.2.6
A:2	Linearity	Linear up to 2^{15} Tags in the Interrogator's RF field; above that number, NlogN for Tags with unique EPCs	6.3.2.8
A:3	Tag Inventory Capacity	Unlimited for Tags with unique EPCs	6.3.2.8

6.3 Description of operating procedure

The operating procedure defines the physical and logical requirements for an Interrogator-talks-first (ITF), random-slotted collision arbitration, RFID system operating in the 860 MHz – 960 MHz frequency range.

6.3.1 Signaling

The signaling interface between an Interrogator and a Tag may be viewed as the physical layer in a layered network communication system. The signaling interface defines frequencies, modulation, data coding, RF envelope, data rates, and other parameters required for RF communications.

6.3.1.1 Operational frequencies

Tags shall receive power from and communicate with Interrogators within the frequency range from 860 MHz to 960 MHz, inclusive. An Interrogator's choice of operational frequency will be determined by local radio regulations and by the local radio-frequency environment. Interrogators certified for operation in dense-Interrogator environments shall support, but are not required to always use, the optional dense-Interrogator mode described in Annex G.

6.3.1.2 Interrogator-to-Tag (R=>T) communications

An Interrogator communicates with one or more Tags by modulating an RF carrier using DSB-ASK, SSB-ASK, or PR-ASK with PIE encoding. Interrogators shall use a fixed modulation format and data rate for the duration of an inventory round, where "inventory round" is defined in 4.1. The Interrogator sets the data rate by means of the preamble that initiates the round.

The high values in Figure 6.1, Figure 6.2, Figure 6.3, Figure 6.4, and Figure 6.5 correspond to emitted CW (i.e. an Interrogator delivering power to the Tag or Tags) whereas the low values correspond to attenuated CW.

6.3.1.2.1 Interrogator frequency accuracy

Interrogators certified for operation in single- or multiple-Interrogator environments shall have a frequency accuracy that meets local regulations.

Interrogators certified for operation in dense-Interrogator environments shall have a frequency accuracy of +/– 10 ppm over the nominal temperature range (–25°C to +40°C) and +/– 20 ppm over the extended temperature range (–40°C to +65°C). Interrogators rated by the manufacturer to have a temperature range wider than nominal but different from extended shall have a frequency accuracy of +/– 10 ppm over the nominal temperature range and +/– 20 ppm to the extent of their rated range. If local regulations specify tighter frequency accuracy then the Interrogator shall meet the local regulations.

6.3.1.2.2 Modulation

Interrogators shall communicate using DSB-ASK, SSB-ASK, or PR-ASK modulation, detailed in Annex H. Tags shall demodulate all three modulation types.

6.3.1.2.3 Data encoding

The R=>T link shall use PIE, shown in Figure 6.1. Tari is the reference time interval for Interrogator-to-Tag signaling, and is the duration of a data-0. High values represent transmitted CW; low values represent attenuated CW.

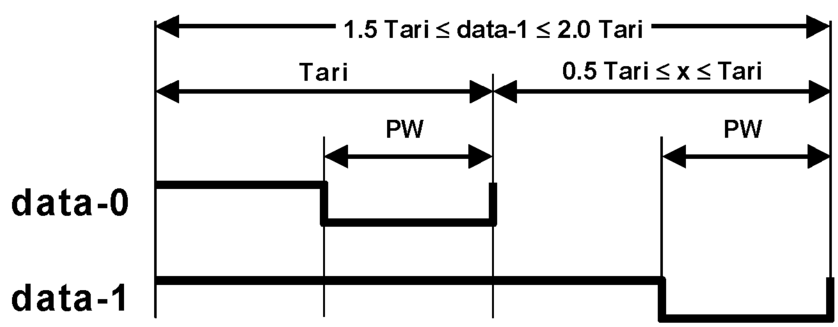


Figure 6.1 – PIE symbols

Pulse modulation depth, rise time, fall time, and PW shall be as specified in Table 6.5, and shall be the same for a data-0 and a data-1. Interrogators shall use a fixed modulation depth, rise time, fall time, PW, Tari, data-0 length, and data-1 length for the duration of an inventory round. The RF envelope shall be as specified in Figure 6.2.

6.3.1.2.4 Tari values

Interrogators shall communicate using Tari values in the range of 6.25µs to 25µs. Interrogator compliance shall be evaluated using at least one Tari value between 6.25µs and 25µs with at least one value of the parameter *x*. The tolerance on all parameters specified in units of Tari shall be +/–1%. The choice of Tari value and *x* shall be in accordance with local radio regulations.

6.3.1.2.5 R=>T RF envelope

The R=>T RF envelope shall comply with Figure 6.2 and Table 6.5. The electric or magnetic field strength A (as appropriate) is the maximum amplitude of the RF envelope, measured in units of V/m or A/m, respectively. Tari is defined in Figure 6.1. The pulsewidth is measured at the 50% point on the pulse. An Interrogator shall not change the R=>T modulation type (i.e. shall not switch between DSB-ASK, SSB-ASK, or PR-ASK) without first powering down its RF waveform (see 6.3.1.2.7).

6.3.1.2.6 Interrogator power-up waveform

The Interrogator power-up RF envelope shall comply with Figure 6.3 and Table 6.6. Once the carrier level has risen above the 10% level, the power-up envelope shall rise monotonically until at least the ripple limit M_l. The RF envelope shall not fall below the 90% point in Figure 6.3 during interval T_s. Interrogators shall not issue commands before the end of the maximum settling-time interval in Table 6.6 (i.e. before the end of T_s). Interrogators shall meet the frequency-accuracy requirement specified in 6.3.1.2.1 by the end of interval T_s in Figure 6.3.

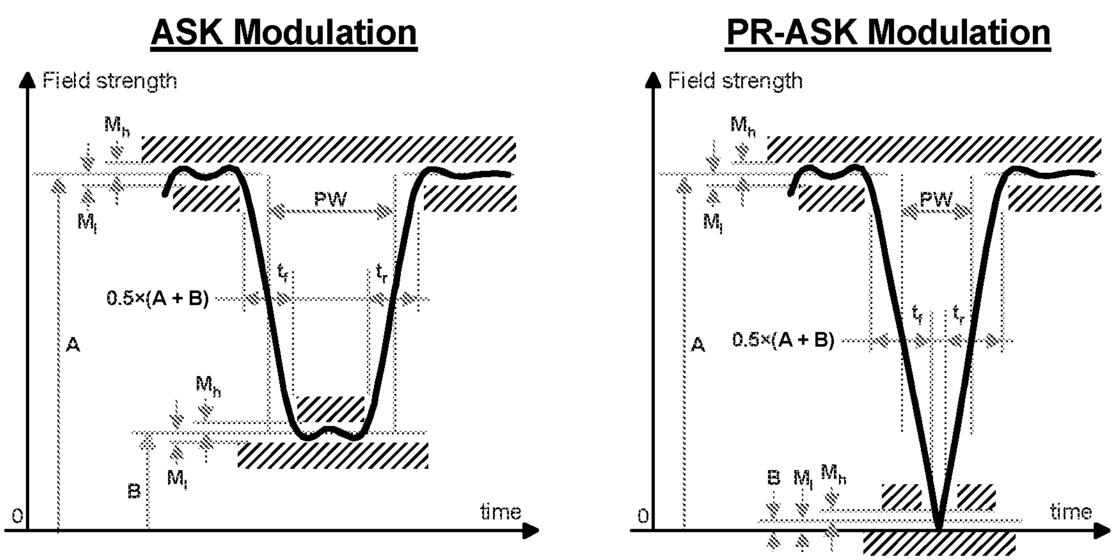


Figure 6.2 – Interrogator-to-Tag RF envelope

Table 6.5 – RF envelope parameters

Tari	Parameter	Symbol	Minimum	Nominal	Maximum	Units
6.25 µs to 25 µs	Modulation Depth	(A–B)/A	80	90	100	%
	RF Envelope Ripple	$M_h = M_l$	0		0.05(A–B)	V/m or A/m
	RF Envelope Rise Time	$t_{r,10-90\%}$	0		0.33Tari	µs
	RF Envelope Fall Time	$t_{f,10-90\%}$	0		0.33Tari	µs
	RF Pulsewidth	PW	MAX(0.265Tari, 2)		0.525Tari	µs

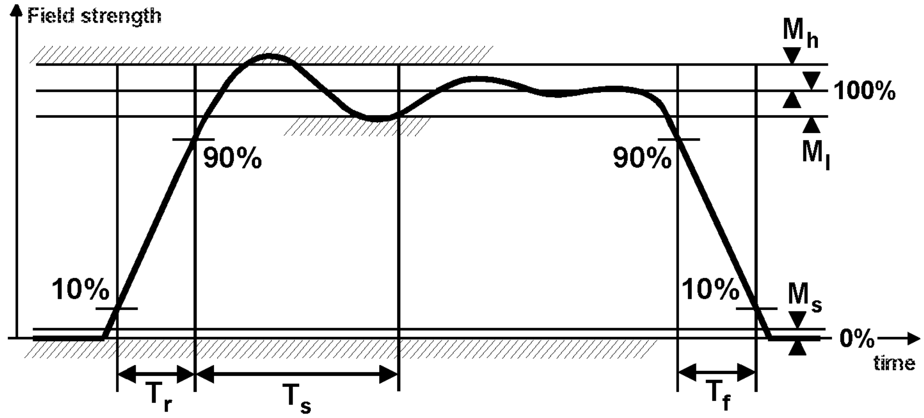


Figure 6.3 – Interrogator power-up and power-down RF envelope

Table 6.6 – Interrogator power-up waveform parameters

Parameter	Definition	Minimum	Nominal	Maximum	Units
T_r	Rise time	1		500	µs
T_s	Settling time			1500	µs
M_s	Signal level when OFF			1	% full scale
M_l	Undershoot			5	% full scale
M_h	Overshoot			5	% full scale

Table 6.7 – Interrogator power-down waveform parameters

Parameter	Definition	Minimum	Nominal	Maximum	Units
T_f	Fall time	1		500	µs
M_s	Signal level when OFF			1	% full scale
M_l	Undershoot			5	% full scale
M_h	Overshoot			5	% full scale

6.3.1.2.7 Interrogator power-down waveform

The Interrogator power-down RF envelope shall comply with Figure 6.3 and Table 6.7. Once the carrier level has fallen below the 90% level, the power-down envelope shall fall monotonically until the power-off limit M_s. Once powered off, an Interrogator shall remain powered off for a least 1ms before powering up again.

6.3.1.2.8 R=>T preamble and frame-sync

An Interrogator shall begin all R=>T signaling with either a preamble or a frame-sync, both of which are shown in Figure 6.4. A preamble shall precede a *Query* command (see 6.3.2.11.2.1) and denotes the start of an inventory round. All other signaling shall begin with a frame-sync. The tolerance on all parameters specified in units of Tari shall be +/–1%. PW shall be as specified in Table 6.5. The RF envelope shall be as specified in Figure 6.2.

A preamble shall comprise a fixed-length start delimiter, a data-0 symbol, an R=>T calibration (RTcal) symbol, and a T=>R calibration (TRcal) symbol.

- **RTcal:** An Interrogator shall set RTcal equal to the length of a data-0 symbol plus the length of a data-1 symbol (RTcal = 0_{length} + 1_{length}). A Tag shall measure the length of RTcal and compute *pivot* = RTcal / 2. A Tag shall interpret subsequent Interrogator symbols shorter than *pivot* to be data-0s, and subsequent Interrogator symbols longer than *pivot* to be data-1s. A Tag shall interpret symbols longer than 4 RTcal to be invalid. Prior to changing RTcal, an Interrogator shall transmit CW for a minimum of 8 RTcal.

- **TRcal:** An Interrogator shall specify a Tag's backscatter link frequency (its FM0 datarate or the frequency of its Miller subcarrier) using the TRcal and divide ratio (DR) in the preamble and payload, respectively, of a *Query* command that initiates an inventory round. Equation (1) specifies the relationship between the backscatter link frequency (BLF), TRcal, and DR. A Tag shall measure the length of TRcal, compute BLF, and adjust its T=>R link rate to be equal to BLF (Table 6.9 shows BLF values and tolerances). The TRcal and RTcal that an Interrogator uses in any inventory round shall meet the constraints in Equation (2):

$$\mathrm{BLF} = \frac{\mathrm{DR}}{\mathrm{TRcal}} \quad (1)$$

$$1.1 \times \mathrm{RTcal} \le \mathrm{TRcal} \le 3 \times \mathrm{RTcal} \quad (2)$$

A frame-sync is identical to a preamble, minus the TRcal symbol. An Interrogator, for the duration of an inventory round, shall use the same length RTcal in a frame-sync as it used in the preamble that initiated the round.

6.3.1.2.9 Frequency-hopping spread-spectrum waveform

When an Interrogator uses frequency-hopping spread spectrum (FHSS) signaling, the Interrogator's RF envelope shall comply with Figure 6.5 and Table 6.8. The RF envelope shall not fall below the 90% point in Figure 6.5 during interval T_{hs}. Interrogators shall not issue commands before the end of the maximum settling-time interval in Table 6.8 (i.e. before the end of T_{hs}). The maximum time between frequency hops and the minimum RF-off time during a hop shall meet local regulatory requirements. Interrogators shall meet the frequency-accuracy requirement specified in 6.3.1.2.1 by the end of interval T_{hs} in Figure 6.5.

6.3.1.2.10 Frequency-hopping spread-spectrum channelization

Interrogators certified for operation in single-Interrogator environments shall meet local regulations for spread-spectrum channelization. Interrogators certified for operation in multiple- or dense-Interrogator environments shall meet local regulations for spread-spectrum channelization, unless the channelization is unregulated, in which case Interrogators shall adopt the channel plan at www.epcglobalinc.org/regulatorychannelplans for the chosen regulatory region (see also Annex G, which describes multiple- and dense-Interrogator channelized signaling).

6.3.1.2.11 Transmit mask

Interrogators certified for operation according to this protocol shall meet local regulations for out-of-channel and out-of-band spurious radio-frequency emissions.

Interrogators certified for operation in multiple-Interrogator environments, in addition to meeting local regulations, shall also meet the Multiple-Interrogator Transmit Mask defined in this specification:

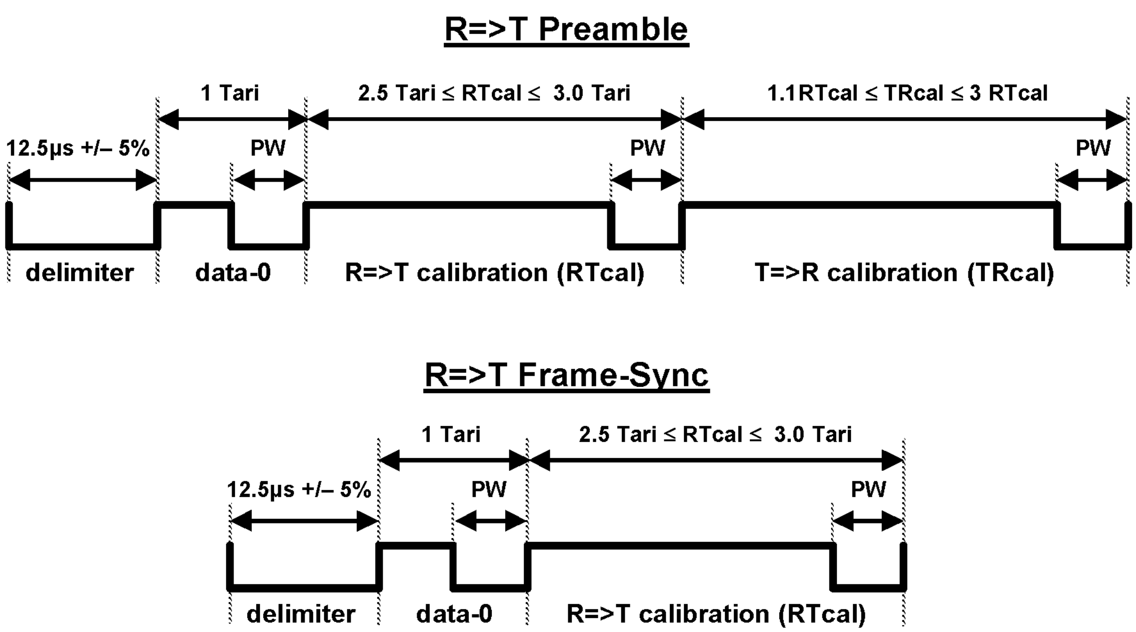


Figure 6.4 – R=>T preamble and frame-sync

Multiple-Interrogator Transmit Mask: For an Interrogator transmitting random data in channel *R*, and any other channel *S*≠*R*, the ratio of the integrated power *P*() in channel *S* to that in channel *R* shall not exceed the specified values:

- $|R - S| = 1$: $10\log_{10}(P(S) / P(R)) < -20$ dB
- $|R - S| = 2$: $10\log_{10}(P(S) / P(R)) < -50$ dB
- $|R - S| = 3$: $10\log_{10}(P(S) / P(R)) < -60$ dB
- $|R - S| > 3$: $10\log_{10}(P(S) / P(R)) < -65$ dB

Where *P*() denotes the total integrated power in the specified channel. This mask is shown graphically in Figure 6.6, with dBch defined as dB referenced to the integrated power in the reference channel. The channel width shall be as specified by local regulations, unless the width is unregulated, in which case Interrogators shall adopt the width shown at www.epcglobalinc.org/regulatorychannelplans for the chosen regulatory region. The channel spacing shall be set equal to the channel width (measured channel center to channel center). For any transmit channel *R*, two exceptions to the mask are permitted, provided that

- neither exception exceeds –50 dBch, and
- neither exception exceeds local regulatory requirements.

An exception occurs when the integrated power in a channel *S* exceeds the mask. Each channel that exceeds the mask shall be counted as an exception.

Interrogators certified for operation in dense-Interrogator environments shall meet both local regulations and the Transmit Mask shown in Figure 6.6 of this specification, except when operating in the optional dense-Interrogator mode described in Annex G, in which case they shall instead meet the Dense-Interrogator Transmit Mask described below and shown in Figure 6.7. Interrogators may meet the Dense-Interrogator Transmit Mask during non-dense-Interrogator operation. Regardless of the mask used, Interrogators certified for operation in dense-Interrogator environments shall not be permitted the two exceptions to the transmit mask that are allowed for Interrogators certified for operation in multiple-Interrogator environments.

Dense-Interrogator Transmit Mask: For Interrogator transmissions centered at a frequency f_c, a 2.5/Tari bandwidth R_{BW} also centered at f_c, an offset frequency f_o = 2.5/Tari, and a 2.5/Tari bandwidth S_{BW} centered at

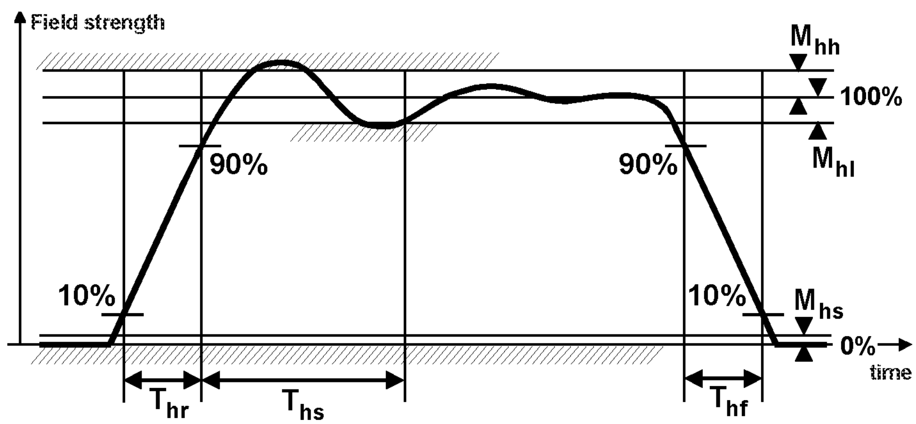


Figure 6.5 – FHSS Interrogator RF envelope

Table 6.8 – FHSS waveform parameters

Parameter	Definition	Minimum	Nominal	Maximum	Units
T_{hr}	Rise time			500	µs
T_{hs}	Settling time			1500	µs
T_{hf}	Fall time			500	µs
M_{hs}	Signal level during hop			1	% full scale
M_{hl}	Undershoot			5	% full scale
M_{hh}	Overshoot			5	% full scale

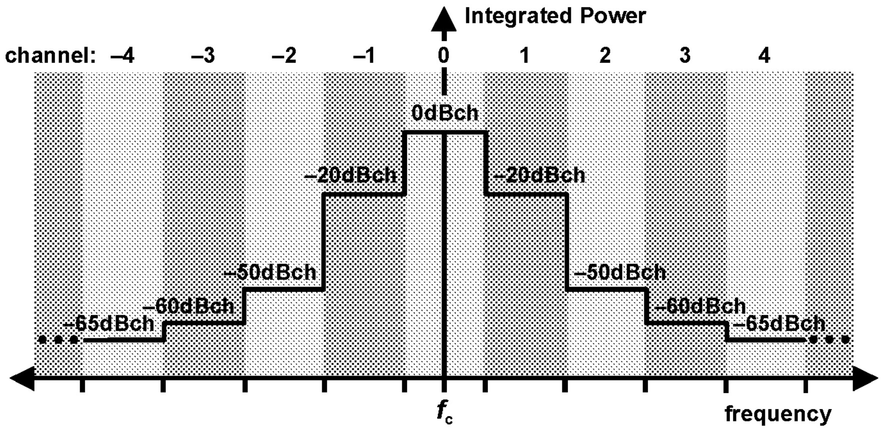


Figure 6.6 – Transmit mask for multiple-Interrogator environments

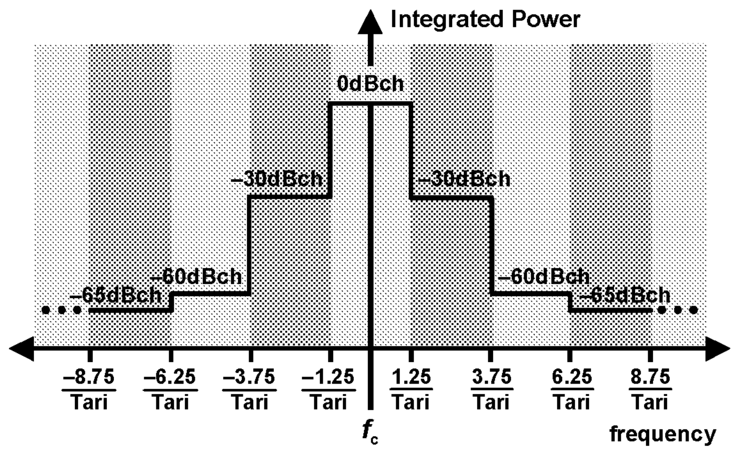


Figure 6.7 – Transmit mask for dense-Interrogator environments

$(n \times f_o) + f_c$ (integer n), the ratio of the integrated power $P()$ in S_{BW} to that in R_{BW} with the Interrogator transmitting random data shall not exceed the specified values:

- $|n| = 1$: $10\log_{10}(P(S_{BW}) / P(R_{BW})) < -30$ dB
- $|n| = 2$: $10\log_{10}(P(S_{BW}) / P(R_{BW})) < -60$ dB
- $|n| > 2$: $10\log_{10}(P(S_{BW}) / P(R_{BW})) < -65$ dB

Where $P()$ denotes the total integrated power in the 2.5/Tari reference bandwidth. This mask is shown graphically in Figure 6.7, with dBch defined as dB referenced to the integrated power in the reference channel.

6.3.1.3 Tag-to-Interrogator (T=>R) communications

A Tag communicates with an Interrogator using backscatter modulation, in which the Tag switches the reflection coefficient of its antenna between two states in accordance with the data being sent.

A Tag shall backscatter using a fixed modulation format, data encoding, and data rate for the duration of an inventory round, where "inventory round" is defined in 4.1. The Tag selects the modulation format; the Interrogator selects the encoding and data rate by means of the *Query* command that initiates the round. The low values in Figure 6.9, Figure 6.10, Figure 6.11, Figure 6.13, Figure 6.14, and Figure 6.15 correspond to the antenna-reflectivity state the Tag exhibits during the CW period prior to a T=>R preamble (e.g. ASK Tag absorbing power), whereas the high values correspond to the antenna-reflectivity state the Tag exhibits during the first high pulse of a T=>R preamble (e.g. ASK Tag reflecting power).

6.3.1.3.1 Modulation

Tag backscatter shall use ASK and/or PSK modulation. The Tag vendor selects the modulation format. Interrogators shall demodulate both modulation types.

6.3.1.3.2 Data encoding

Tags shall encode the backscattered data as either FM0 baseband or Miller modulation of a subcarrier at the data rate. The Interrogator commands the encoding choice.

6.3.1.3.2.1 FM0 baseband

Figure 6.8 shows basis functions and a state diagram for generating FM0 (bi-phase space) encoding. FM0 inverts the baseband phase at every symbol boundary; a data-0 has an additional mid-symbol phase inversion. The state diagram in Figure 6.8 maps a logical data sequence to the FM0 basis functions that are transmitted. The state labels, S_1–S_4, indicate four possible FM0-encoded symbols, represented by the two phases of each of the FM0 basis functions. The state labels also represent the FM0 waveform that is transmitted upon entering the state. The labels on the state transitions indicate the logical values of the data sequence to be encoded. For example, a transition from state S_2 to S_3 is disallowed because the resulting transmission would not have a phase inversion on a symbol boundary.

Figure 6.9 shows generated baseband FM0 symbols and sequences. The duty cycle of a 00 or 11 sequence, measured at the modulator output, shall be a minimum of 45% and a maximum of 55%, with a nominal value of 50%. FM0 encoding has memory; consequently, the choice of FM0 sequences in Figure 6.9 depends on prior transmissions. FM0 signaling shall always end with a “dummy” data-1 bit at the end of a transmission, as shown in Figure 6.10.

6.3.1.3.2.2 FM0 preamble

T=>R FM0 signaling shall begin with one of the two preambles shown in Figure 6.11. The choice depends on the value of the TRext bit specified in the *Query* command that initiated the inventory round, unless a Tag is replying to a command that writes to memory, in which case a Tag shall use the extended preamble regardless of TRext (i.e. the Tag replies as if TRext=1 regardless of the TRext value specified in the *Query*—see 6.3.2.11.3). The “v” shown in Figure 6.11 indicates an FM0 violation (i.e. a phase inversion should have occurred but did not).

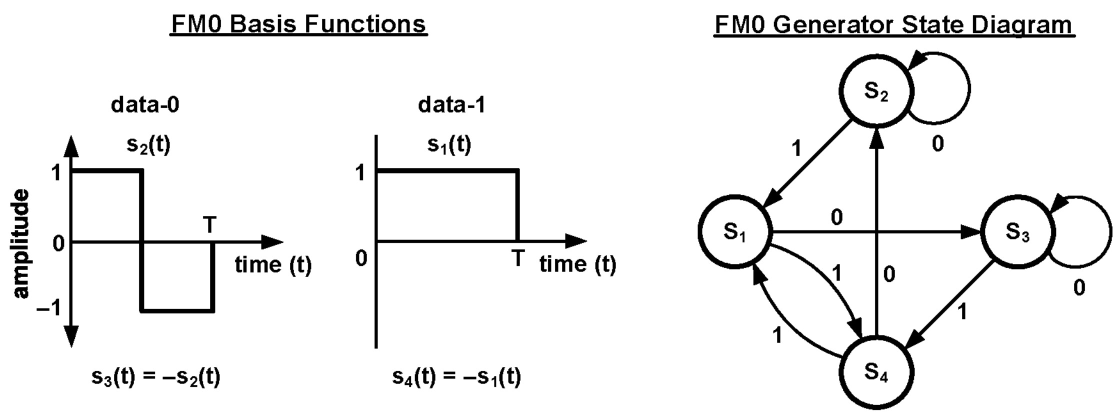


Figure 6.8 – FM0 basis functions and generator state diagram

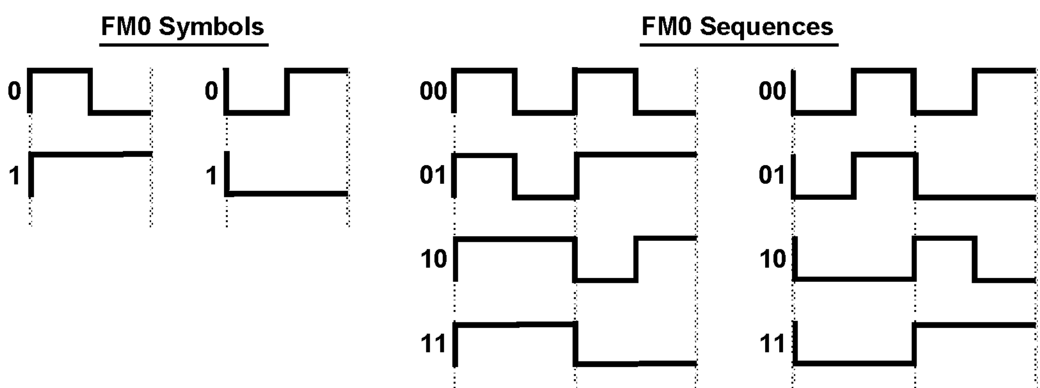


Figure 6.9 – FM0 symbols and sequences

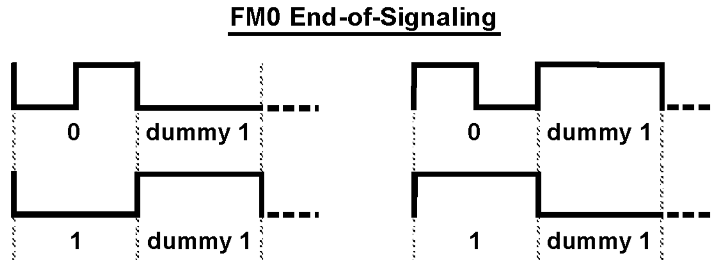


Figure 6.10 – Terminating FM0 transmissions

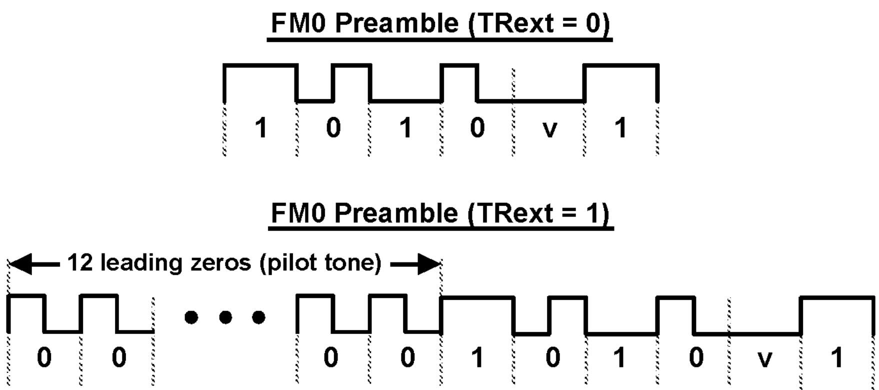


Figure 6.11 – FM0 T=>R preamble

6.3.1.3.2.3 Miller-modulated subcarrier

Figure 6.12 shows basis functions and a state diagram for generating Miller encoding. Baseband Miller inverts its phase between two data-0s in sequence. Baseband Miller also places a phase inversion in the middle of a data-1 symbol. The state diagram in Figure 6.12 maps a logical data sequence to baseband Miller basis functions. The state labels, S_1–S_4, indicate four possible Miller-encoded symbols, represented by the two phases of each of the Miller basis functions. The state labels also represent the baseband Miller waveform that is generated upon entering the state. The transmitted waveform is the baseband waveform multiplied by a square-wave at M times the symbol rate. The labels on the state transitions indicate the logical values of the data sequence to be encoded. For example, a transition from state S_1 to S_3 is disallowed because the resulting transmission would have a phase inversion on a symbol boundary between a data-0 and a data-1.

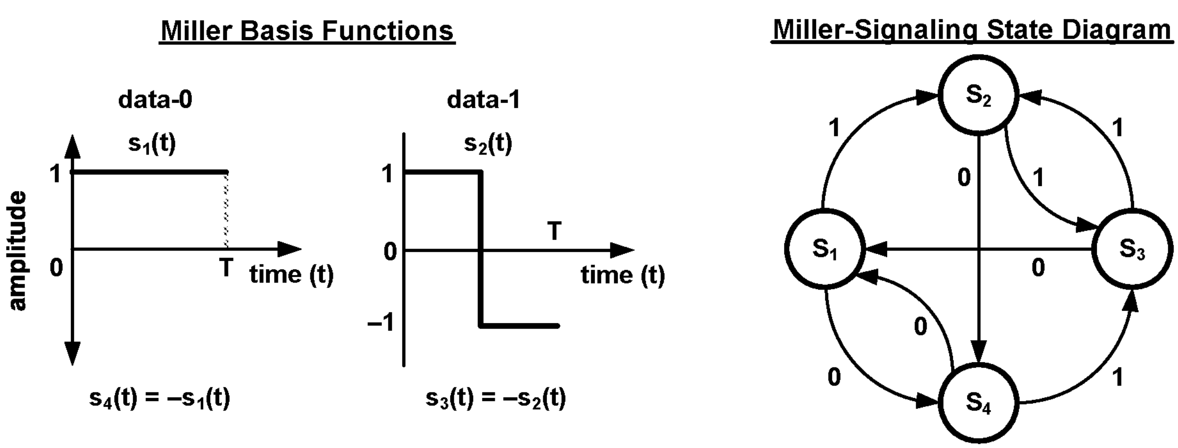


Figure 6.12 – Miller basis functions and generator state diagram

Figure 6.13 shows Miller-modulated subcarrier sequences; the Miller sequence shall contain exactly two, four, or eight subcarrier cycles per bit, depending on the M value specified in the *Query* command that initiated the inventory round (see Table 6.10). The duty cycle of a 0 or 1 symbol, measured at the modulator output, shall be a minimum of 45% and a maximum of 55%, with a nominal value of 50%. Miller encoding has memory; consequently, the choice of Miller sequences in Figure 6.13 depends on prior transmissions. Miller signaling shall always end with a "dummy" data-1 bit at the end of a transmission, as shown in Figure 6.14.

6.3.1.3.2.4 Miller subcarrier preamble

T=>R subcarrier signaling shall begin with one of the two preambles shown in Figure 6.15. The choice depends on the value of the TRext bit specified in the *Query* command that initiated the inventory round, unless a Tag is replying to a command that writes to memory, in which case a Tag shall use the extended preamble regardless of TRext (i.e. the Tag replies as if TRext=1 regardless of the TRext value specified in the *Query*—see 6.3.2.11.3).

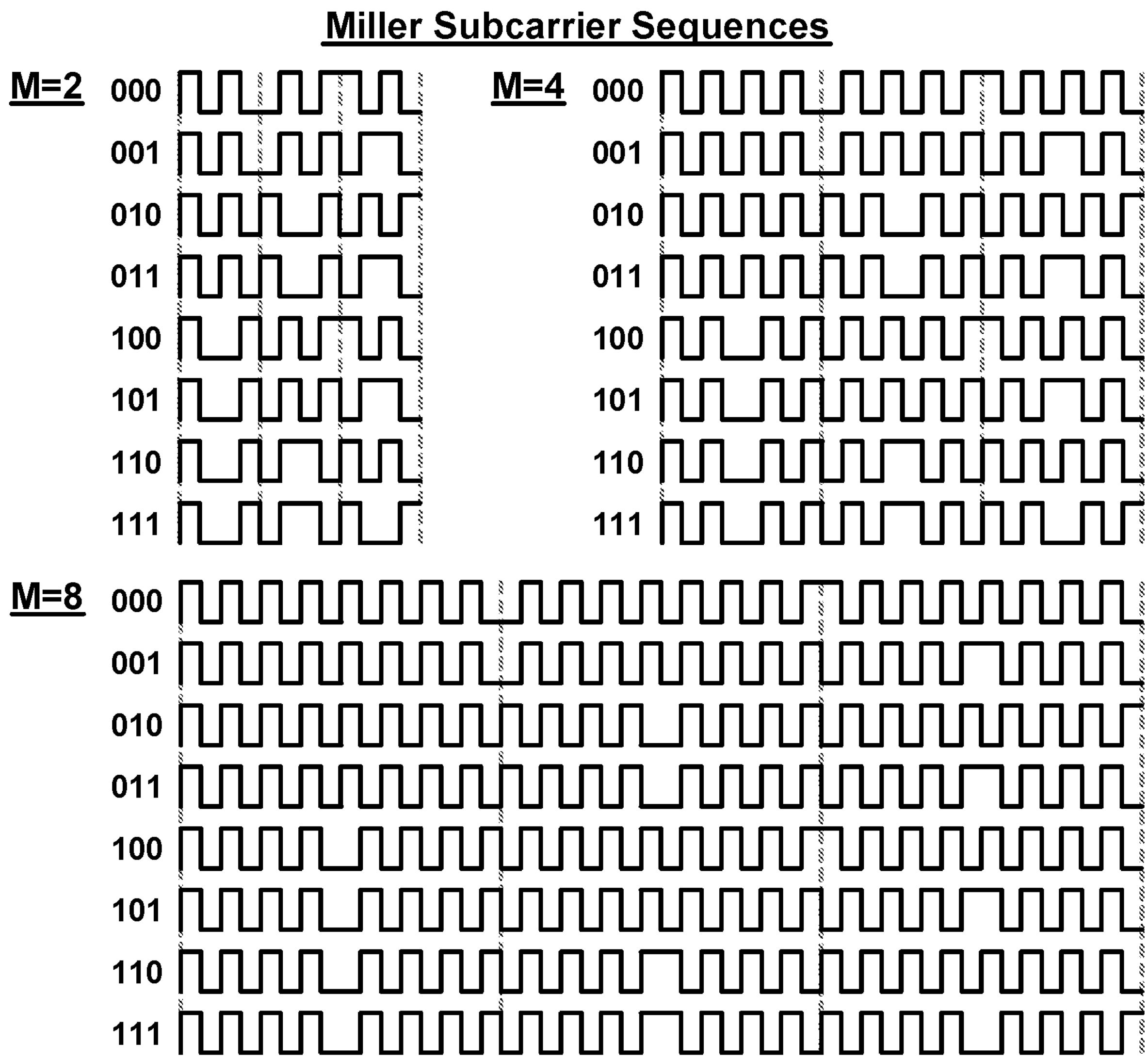


Figure 6.13 – Subcarrier sequences

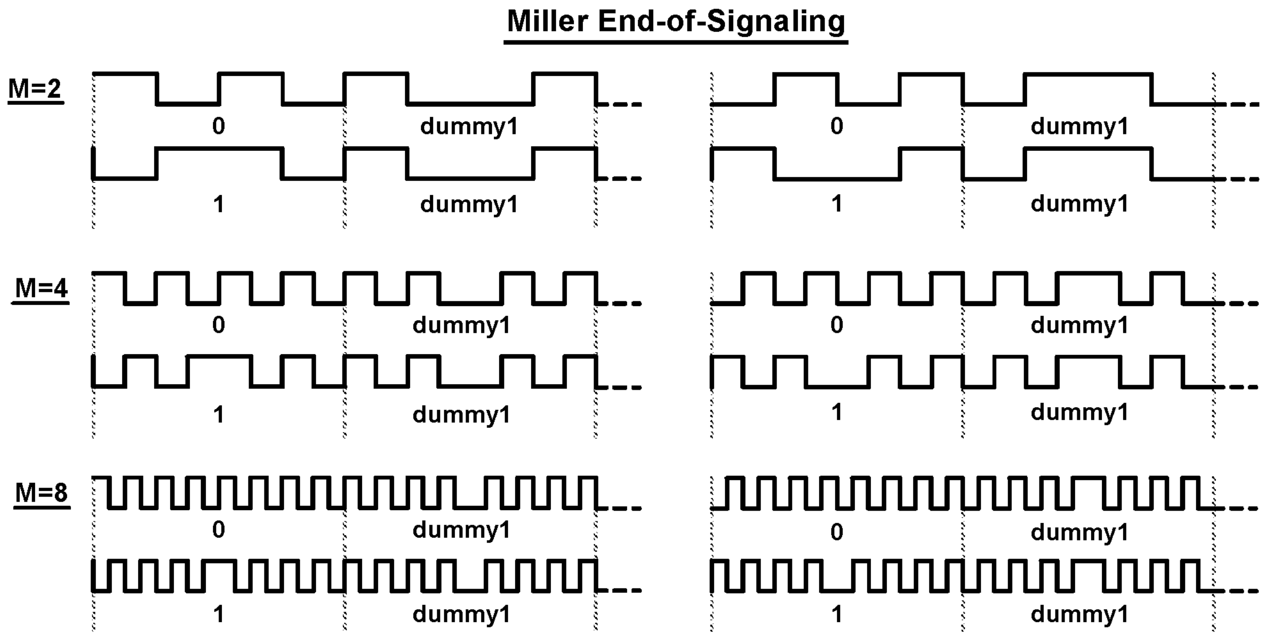


Figure 6.14 – Terminating subcarrier transmissions

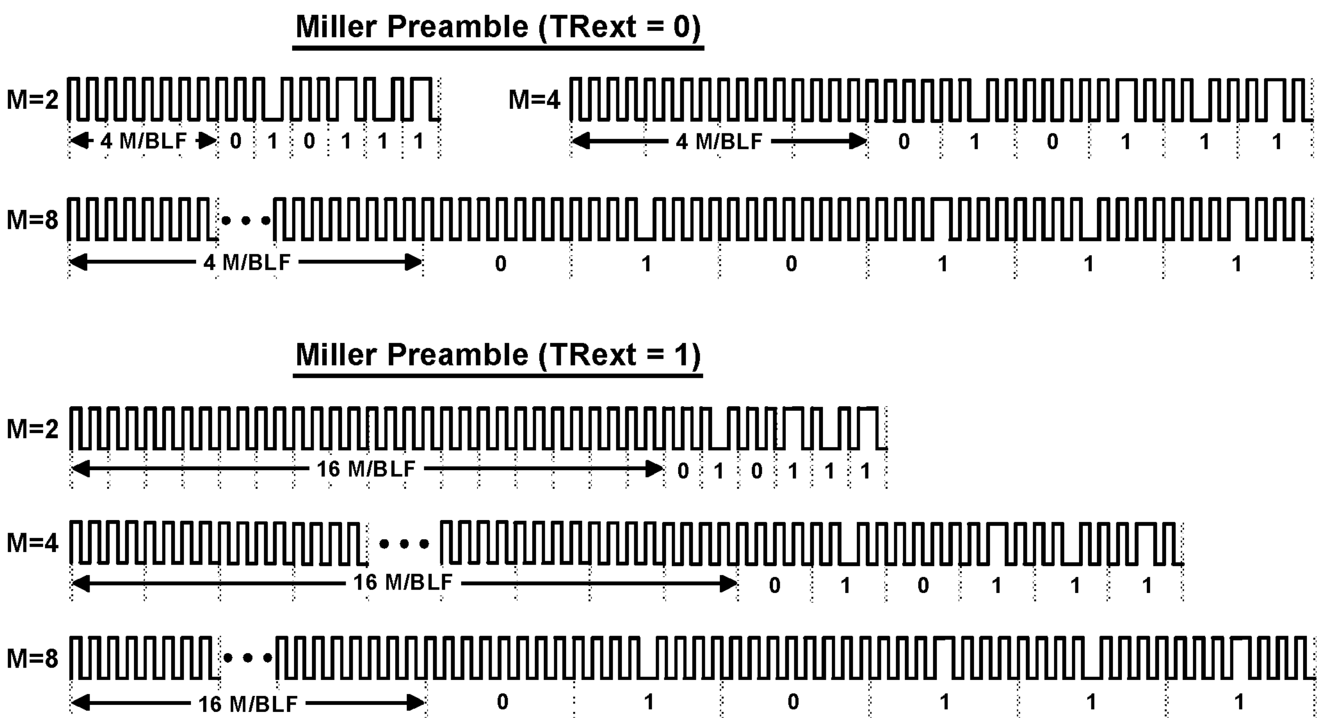


Figure 6.15 – Subcarrier T=>R preamble

6.3.1.3.3 Tag supported Tari values and backscatter link rates

Tags shall support all R=>T Tari values in the range of 6.25µs to 25µs, over all parameters allowed by 6.3.1.2.3.

Tags shall support the T=>R link frequencies and tolerances specified in Table 6.9 and the T=>R data rates specified in Table 6.10. The frequency-variation requirement in Table 6.9 includes both frequency drift and short-term frequency variation during Tag response to an Interrogator command. The *Query* command that initiates an inventory round specifies DR in Table 6.9 and M in Table 6.10; the preamble that precedes the *Query* specifies TRcal. BLF is computed using Eq. (1). These four parameters together define the backscatter frequency, modulation type (FM0 or Miller), and T=>R data rate for the round (see also 6.3.1.2.8).

6.3.1.3.4 Tag power-up timing

Tags energized by an Interrogator shall be capable of receiving and acting on Interrogator commands within a period not exceeding the maximum settling-time interval specified in Table 6.6 or Table 6.8, as appropriate (i.e. by the end of T_s or T_{hs}, respectively).

6.3.1.3.5 Minimum operating field strength and backscatter strength

For a Tag certified to this protocol, the Tag manufacturer shall specify:

1. free-space sensitivity,

Table 6.9 – Tag-to-Interrogator link frequencies

DR: Divide Ratio	TRcal[1] (µs +/– 1%)	BLF: Link Frequency (kHz)	Frequency Tolerance FT (nominal temp)	Frequency Tolerance FT (extended temp)	Frequency variation during backscatter
64/3	33.3	640	+ / – 15%	+ / – 15%	+ / – 2.5%
	33.3 < TRcal < 66.7	320 < BLF < 640	+ / – 22%	+ / – 22%	+ / – 2.5%
	66.7	320	+ / – 10%	+ / – 15%	+ / – 2.5%
	66.7 < TRcal < 83.3	256 < BLF < 320	+ / – 12%	+ / – 15%	+ / – 2.5%
	83.3	256	+ / – 10%	+ / – 10%	+ / – 2.5%
	83.3 < TRcal ≤ 133.3	160 ≤ BLF < 256	+ / – 10%	+ / – 12%	+ / – 2.5%
	133.3 < TRcal ≤ 200	107 ≤ BLF < 160	+ / – 7%	+ / – 7%	+ / – 2.5%
	200 < TRcal ≤ 225	95 ≤ BLF < 107	+ / – 5%	+ / – 5%	+ / – 2.5%
8	17.2 ≤ TRcal < 25	320 < BLF ≤ 465	+ / – 19%	+ / – 19%	+ / – 2.5%
	25	320	+ / – 10%	+ / – 15%	+ / – 2.5%
	25 < TRcal < 31.25	256 < BLF < 320	+ / – 12%	+ / – 15%	+ / – 2.5%
	31.25	256	+ / – 10%	+ / – 10%	+ / – 2.5%
	31.25 < TRcal < 50	160 < BLF < 256	+ / – 10%	+ / – 10%	+ / – 2.5%
	50	160	+ / – 7%	+ / – 7%	+ / – 2.5%
	50 < TRcal ≤ 75	107 ≤ BLF < 160	+ / – 7%	+ / – 7%	+ / – 2.5%
	75 < TRcal ≤ 200	40 ≤ BLF < 107	+ / – 4%	+ / – 4%	+ / – 2.5%

Note 1: Allowing two different TRcal values (with two different DR values) to specify the same BLF offers flexibility in specifying Tari and RTcal.

Table 6.10 – Tag-to-Interrogator data rates

M: Number of subcarrier cycles per symbol	Modulation type	Data rate (kbps)
1	FM0 baseband	BLF
2	Miller subcarrier	BLF/2
4	Miller subcarrier	BLF/4
8	Miller subcarrier	BLF/8

2. minimum backscattered modulated power (ASK modulation) or change in radar cross-section or equivalent (phase modulation), and
3. the manufacturer's normal operating conditions,

for the Tag mounted on one or more manufacturer-selected materials.

6.3.1.4 Transmission order

The transmission order for all R=>T and T=>R communications shall be most-significant bit (MSB) first.

Within each message, the most-significant word shall be transmitted first.

Within each word, the MSB shall be transmitted first.

6.3.1.5 Cyclic-redundancy check (CRC)

A CRC is a cyclic-redundancy check that a Tag uses to ensure the validity of certain R=>T commands, and an Interrogator uses to ensure the validity of certain backscattered T=>R replies. This protocol uses two CRC types: (i) a CRC-16, and (ii) a CRC-5. Annex F describes both CRC types.

To generate a CRC-16 a Tag or Interrogator shall first generate the CRC-16 precursor shown in Table 6.11, and then take the ones-complement of the generated precursor to form the CRC-16.

A Tag or Interrogator shall verify the integrity of a received message that uses a CRC-16. The Tag or Interrogator may use one of the methods described in Annex F to verify the CRC-16.

At power-up, a Tag calculates and saves into memory a 16-bit StoredCRC. During inventory, a Tag may backscatter either this StoredCRC, or a 16-bit PacketCRC that the Tag calculates dynamically. See 6.3.2.1.2.1.

Tags shall append a CRC-16 to those replies that use a CRC-16 — see 6.3.2.11 for command-specific replies.

To generate a CRC-5 an Interrogator shall use the definition in Table 6.12.

A Tag shall verify the integrity of a received message that uses a CRC-5. The Tag may use the method described in Annex F to verify a CRC-5.

Interrogators shall append the appropriate CRC to R=>T transmissions as specified in Table 6.18.

6.3.1.6 Link timing

Figure 6.16 illustrates R=>T and T=>R link timing. The figure (not drawn to scale) defines Interrogator interactions with a Tag population. Table 6.13 shows the timing requirements for Figure 6.16, while 6.3.2.11 describes the commands. Tags and Interrogators shall meet all timing requirements shown in Table 6.13. RTcal is defined in 6.3.1.2.8; T_{pri} is the T=>R link period (T_{pri} = 1 / BLF). As described in 6.3.1.2.8, an Interrogator shall use a fixed R=>T link rate for the duration of an inventory round; prior to changing the R=>T link rate, an Interrogator shall transmit CW for a minimum of 8 RTcal.

Table 6.11 – CRC-16 precursor

CRC-16 precursor				
CRC Type	Length	Polynomial	Preset	Residue
ISO/IEC 13239	16 bits	$x^{16} + x^{12} + x^5 + 1$	$FFFF_h$	$1D0F_h$

Table 6.12 – CRC-5 definition. See also Annex F

CRC-5 Definition				
CRC Type	Length	Polynomial	Preset	Residue
—	5 bits	$x^5 + x^3 + 1$	01001_2	00000_2

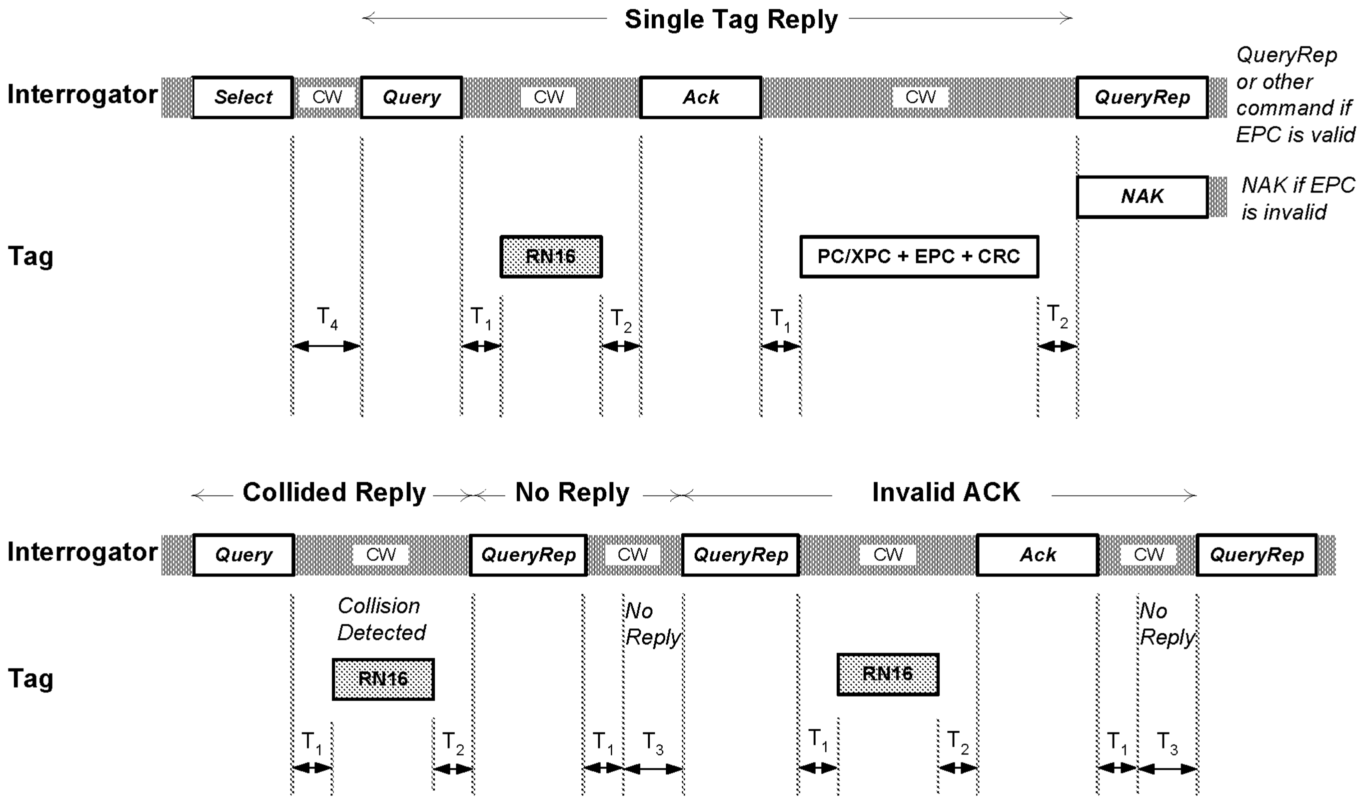


Figure 6.16 – Link timing

Table 6.13 – Link timing parameters

Parameter	Minimum	Nominal	Maximum	Description
T_1	$MAX(RTcal, 10T_{pri}) \times (1 - \lvert FT \rvert) - 2\mu s$	$MAX(RTcal, 10T_{pri})$	$MAX(RTcal, 10T_{pri}) \times (1 + \lvert FT \rvert) + 2\mu s$	Time from Interrogator transmission to Tag response (specifically, the time from the last rising edge of the last bit of the Interrogator transmission to the first rising edge of the Tag response), measured at the Tag's antenna terminals
T_2	$3.0T_{pri}$		$20.0T_{pri}$	Interrogator response time required if a Tag is to demodulate the Interrogator signal, measured from the end of the last (dummy) bit of the Tag response to the first falling edge of the Interrogator transmission
T_3	$0.0T_{pri}$			Time an Interrogator waits, after T_1, before it issues another command
T_4	2.0 RTcal			Minimum time between Interrogator commands

The following items apply to the requirements specified in Table 6.13:

1. T_{pri} denotes either the commanded period of an FM0 symbol or the commanded period of a single subcarrier cycle, as appropriate.
2. A Tag may exceed the maximum value for T_1 when responding to commands that write to memory — see, for example, 6.3.2.11.3.3.
3. The maximum value for T_2 shall apply only to Tags in the **reply** or **acknowledged** states (see 6.3.2.4.3 and 6.3.2.4.4). For a Tag in the **reply** or **acknowledged** states, if T_2 expires (i.e. reaches its maximum value):
   - Without the Tag receiving a valid command, the Tag shall transition to the **arbitrate** state (see 6.3.2.4.2),
   - During the reception of a valid command, the Tag shall execute the command,
   - During the reception of an invalid command, the Tag shall transition to **arbitrate** upon determining that the command is invalid.

   In all other states the maximum value for T_2 shall be unrestricted. A Tag shall be allowed a tolerance of $20.0T_{pri} \leq T_{2(max)} \leq 32T_{pri}$ in determining whether T_2 has expired. "Invalid command" is defined in 6.3.2.11.
4. An Interrogator may transmit a new command prior to interval T_2 (i.e. during a Tag response). In this case the responding Tag is not required to demodulate or otherwise act on the new command, and may undergo a power-on reset.
5. FT is the frequency tolerance specified in Table 6.9
6. T_1+T_3 shall not be less than T_4.

6.3.2 Tag selection, inventory, and access

Tag selection, inventory, and access may be viewed as the lowest level in the data link layer of a layered network communication system.

6.3.2.1 Tag memory

Tag memory shall be logically separated into four distinct banks, each of which may comprise zero or more memory words. A logical memory map is shown in Figure 6.17. The memory banks are:

Reserved memory shall contain the kill and and/or access passwords, if passwords are implemented on the Tag. The kill password shall be stored at memory addresses 00_h to $1F_h$; the access password shall be stored at memory addresses 20_h to $3F_h$. See 6.3.2.1.1.

EPC memory shall contain a StoredCRC at memory addresses 00_h to $0F_h$, a StoredPC at addresses 10_h to $1F_h$, a code (such as an EPC, and hereafter referred to as an EPC) that identifies the object to which the Tag is or will be attached beginning at address 20_h, and if the Tag implements Extended Protocol Control (XPC) then either one or two XPC word(s) beginning at address 210_h. See 6.3.2.1.2.

TID memory shall contain an 8-bit ISO/IEC 15963 allocation class identifier at memory locations 00_h to 07_h. TID memory shall contain sufficient identifying information above 07_h for an Interrogator to uniquely identify the custom commands and/or optional features that a Tag supports. See 6.3.2.1.3.

User memory is optional. See 6.3.2.1.4.

The logical addressing of all memory banks shall begin at zero (00_h). The physical memory map is vendor-specific. Commands that access memory have a MemBank parameter that selects the bank, and an address parameter, specified using the EBV format described in Annex A, to select a particular memory location within that bank. When Tags backscatter memory contents, this backscatter shall fall on word boundaries (except in the case of a truncated reply – see 6.3.2.11.1.1).

MemBank is defined as follows:

00_2 Reserved
01_2 EPC
10_2 TID
11_2 User

Operations in one logical memory bank shall not access memory locations in another bank.

Memory writes, detailed in 6.3.2.9, involve the transfer of 16-bit words from Interrogator to Tag. A *Write* command writes 16 bits (i.e. one word) at a time, using link cover-coding to obscure the data during R=>T transmission. The optional *BlockWrite* command writes one or more 16-bit words at a time, without link cover-coding. The optional *BlockErase* command erases one or more 16-bit words at a time. A *Write*, *BlockWrite*, or *BlockErase* shall not alter a Tag's killed status regardless of the memory address (whether valid or invalid) specified in the command.

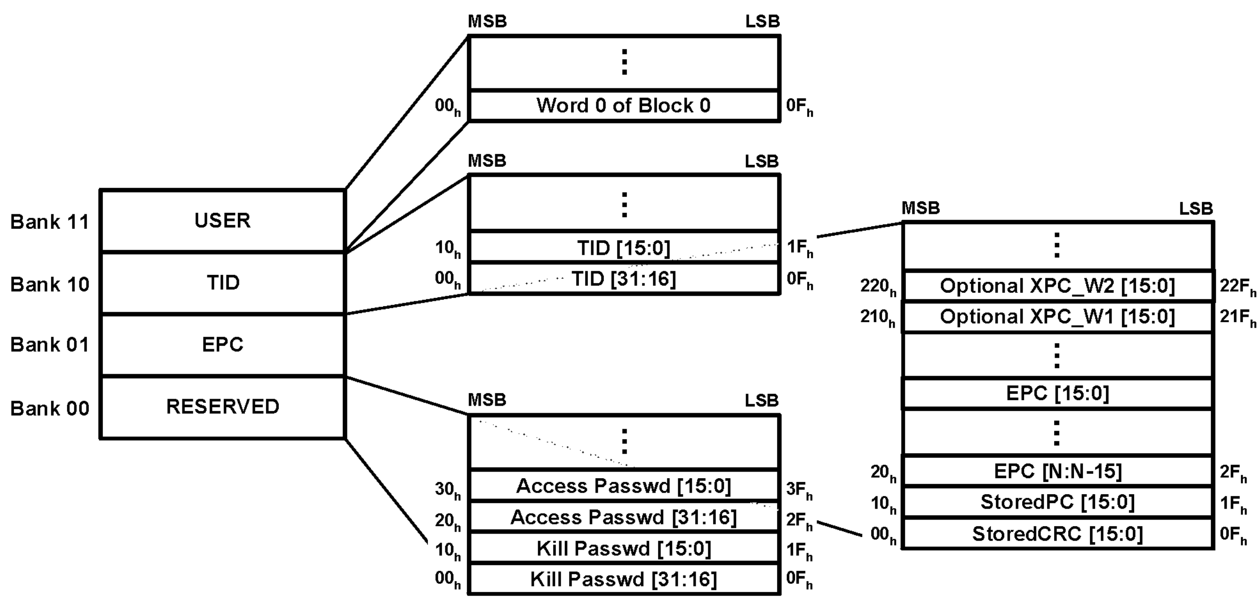


Figure 6.17 – Logical memory map

Interrogators may lock, permanently lock, unlock, or permanently unlock the kill password, access password, EPC memory, TID memory, and User memory, thereby preventing or allowing subsequent changes (as appropriate). A Tag may optionally have its User memory partitioned into blocks; if it does, then an Interrogator may permanently lock these individual blocks. Recommissioning may alter the memory locking and/or permalocking. See 6.3.2.9 for a description of memory locking and unlocking, and 6.3.2.10 for a description of Tag recommissioning. If the kill and/or access passwords are locked they are usable by only the *Kill* and *Access* commands, respectively, and are rendered both unwriteable and unreadable by any other command. Locking or permanently locking the EPC, TID, or User memory banks, or permanently locking blocks within the User memory bank, renders the locked memory location unwriteable but leaves it readable.

6.3.2.1.1 Reserved Memory

Reserved memory contains the kill (see 6.3.2.1.1.1) and/or access (see 6.3.2.1.1.2) passwords, if passwords are implemented on the Tag. If a Tag does not implement the kill and/or access password(s), the Tag shall logically operate as though it has zero-valued password(s) that are permanently read/write locked (see 6.3.2.11.3.5), and the corresponding physical memory locations in Reserved memory need not exist.

6.3.2.1.1.1 Kill password

The kill password is a 32-bit value stored in Reserved memory 00_h to $1F_h$, MSB first. The default (unprogrammed) value shall be zero. An Interrogator may use the kill password to (1) recommission a Tag, and/or (2) kill a Tag and render it nonresponsive thereafter. A Tag shall not execute a recommissioning or kill operation if its kill password is zero. A Tag that does not implement a kill password operates as if it has a zero-valued kill password that is permanently read/write locked.

6.3.2.1.1.2 Access password

The access password is a 32-bit value stored in Reserved memory 20_h to $3F_h$, MSB first. The default (unprogrammed) value shall be zero. A Tag with a nonzero access password shall require an Interrogator to issue this password before transitioning to the **secured** state. A Tag that does not implement an access password operates as if it has a zero-valued access password that is permanently read/write locked

6.3.2.1.2 EPC Memory

EPC memory contains a StoredCRC at memory addresses 00_h to $0F_h$, a StoredPC at 10_h to $1F_h$, an EPC beginning at 20_h, and if a Tag implements recommissioning then a first XPC word (XPC_W1) at 210_h to $21F_h$ and an optional second XPC word (XPC_W2) at 220_h to $22F_h$. Higher-class Tags may implement an XPC_W1 and an XPC_W2 without implementing recommissioning. The StoredCRC, StoredPC, EPC, and XPC word or words shall be stored MSB first (i.e. the EPC's MSB is stored in location 20_h).

The StoredCRC is described in 6.3.2.1.2.1.

The StoredPC, as described in 6.3.2.1.2.2, is subdivided into an EPC length field in memory locations 10_h to 14_h, a User-memory indicator (UMI) in location 15_h, an XPC_W1 indicator (XI) in location 16_h, and a Numbering System Identifier (NSI) in locations 17_h to $1F_h$.

The EPC is a code that identifies the object to which a Tag is affixed. The EPC for EPCglobal™ Applications is described in 6.3.2.1.2.3; the EPC for non-EPCglobal™ Applications is described in 6.3.2.1.2.4. Interrogators may issue a *Select* command that includes all or part of the EPC in the mask. Interrogators may issue an *ACK* command to cause a Tag to backscatter its EPC. Under certain circumstances the Tag may truncate its backscattered EPC (see 6.3.2.11.1.1). An Interrogator may issue a *Read* command to read all or part of the EPC.

6.3.2.1.2.1 CRC-16 (StoredCRC and PacketCRC)

All Tags shall implement a StoredCRC. Tags that support XPC functionality shall also implement a PacketCRC.

At power-up a Tag shall calculate a CRC-16 over (a) the StoredPC and (b) the EPC specified by the EPC length field in the StoredPC (see 6.3.2.1.2.2) and shall map the calculated CRC-16 into EPC memory 00_h to $0F_h$, MSB first. This CRC is denoted the StoredCRC. Because the StoredPC and EPC comprise an integer number of EPC-memory words, a Tag calculates this StoredCRC on word boundaries. Although Tags include the XI value in the StoredPC in their StoredCRC calculation, regardless of XI value a Tag shall omit XPC_W1 and XPC_W2 from the calculation. A Tag shall finish its StoredCRC calculation and memory mapping by the end of interval T_s or T_{hs} (as appropriate) in Figure 6.3 or Figure 6.5, respectively. Interrogators may issue a *Select* command that includes all or part of the StoredCRC in Mask. Interrogators may issue a *Read* command to instruct a Tag to backscatter its StoredCRC. A Tag shall not recalculate this StoredCRC for a truncated reply (see 6.3.2.11.1.1).

Table 6.14 – Tag data and CRC-16 backscattered in response to an *ACK* command

XI	XEB	Truncation	Tag Backscatter			
			PC	XPC	EPC	CRC-16
0	0	Deasserted	StoredPC	None	Full	StoredCRC or PacketCRC
0	0	Asserted	00000_2	None	Truncated	StoredCRC
0	1	Deasserted	Invalid[1]			
0	1	Asserted	Invalid[1]			
1	0	Deasserted	PacketPC	XPC_W1	Full	PacketCRC
1	0	Asserted	00000_2	None	Truncated	StoredCRC
1	1	Deasserted	PacketPC	Both XPC_W1 and XPC_W2	Full	PacketCRC
1	1	Asserted	00000_2	None	Truncated	StoredCRC

Note 1: XI is the bitwise logical OR of the 16 bits of XPC_W1, and XEB is the MSB (bit F_h) of XPC_W1, so if XEB=1 then XI=1.

In response to an *ACK* command a Tag backscatters a protocol-control (PC) word (either StoredPC or PacketPC – see 6.3.2.1.2.2), optional XPC word or words (see 6.3.2.1.2.5), EPC (see 6.3.2.1.2.3 and 6.3.2.1.2.4), and a CRC-16 that protects the backscattered data. The CRC-16 shall be either (a) the StoredCRC or, alternatively, (b) a PacketCRC that the Tag shall calculate dynamically over the backscattered PC word, optional XPC word or words, and EPC. Whether a Tag backscatters its StoredCRC or a PacketCRC shall be as defined in Table 6.14, depending on the Tag's XI value and whether truncation (see 6.3.2.11.1.1) is asserted or deasserted.

Tags that do not support XPC functionality may, but are not required to, implement a PacketCRC.

A Tag shall backscatter its CRC MSB first, regardless of the CRC type.

If XI is asserted then a Tag's PacketCRC is different from its StoredCRC.

As required by 6.3.1.5 an Interrogator shall verify, using a Tag's backscattered CRC-16, the integrity of a received PC word, optional XPC word or words, and EPC.

6.3.2.1.2.2 Protocol-control (PC) word (StoredPC and PacketPC)

All Tags shall implement a StoredPC whose fields, comprising EPC length, a UMI, an XI, and an NSI, shall be as defined below. Tags that support XPC functionality shall also implement a PacketPC that differs from the StoredPC in its EPC length field. The type of PC (StoredPC or PacketPC) that a Tag backscatters in response to an *ACK* shall be as defined in Table 6.14.

The StoredPC shall be located in EPC memory at addresses 10_h to $1F_h$, with bit values defined as follows:

- Bits 10_h – 14_h: EPC length field. The length of the EPC, in words:
  - 00000_2: Zero words.
  - 00001_2: One word (addresses 20_h to $2F_h$ in EPC memory).
  - 00010_2: Two words (addresses 20_h to $3F_h$ in EPC memory).

    ●
    ●
    ●
  - 11111_2: 31 words (addresses 20_h to $20F_h$ in EPC memory).

  If a Tag does not support XPC functionality then the maximum value of the EPC length field in the StoredPC shall be 11111_2 (allows a 496-bit EPC), as shown above. If a Tag supports XPC functionality then the maximum value of the EPC length field in the StoredPC shall be 11101_2 (allows a 464-bit EPC). A Tag that supports XPC functionality shall ignore a *Write* or *BlockWrite* command to the StoredPC if the EPC length field exceeds 11101_2, and shall instead backscatter an error code (see Annex I).

- Bit 15_h: A User-memory indicator (UMI). If bit 15_h is deasserted then the Tag either does not implement User memory or User memory contains no information. If bit 15_h is asserted then User memory contains information. A Tag may implement the UMI using Method 1 or Method 2 described below, unless the Tag implements block permalocking and/or recommissioning, in which case the Tag shall use Method 1.

- Method 1: The Tag computes the UMI. At power-up, and prior to calculating its StoredCRC (see 6.3.2.1.2.1), a Tag shall compute the logical OR of bits 03_h to 07_h of User memory and map the computed value into bit 15_h. A Tag shall use this computed UMI value in its StoredCRC calculation. If an Interrogator modifies any of bits 03_h to 07_h of User memory then the Tag shall recompute and remap its UMI into bit 15_h. If recommissioning renders User memory inaccessible (see 6.3.2.10) then the Tag shall deassert and remap its UMI into bit 15_h. After remapping the UMI the StoredCRC may be incorrect until the Interrogator power cycles the Tag. The UMI shall not be directly writeable by an Interrogator — when an Interrogator writes the StoredPC the Tag shall ignore the data value that the Interrogator provides for bit 15_h.
- Method 2: An Interrogator writes the UMI. If an Interrogator writes a zero value into bits 03_h to 07_h of User memory then the Interrogator shall deassert bit 15_h. If an Interrogator writes a nonzero value into 03_h to 07_h of User memory then the Interrogator shall assert bit 15_h. If an Interrogator locks or permalocks EPC memory then the Interrogator shall also lock or permalock, respectively, the word located at address 00_h of User memory, and vice versa. This latter requirement ensures that a condition in which User memory previously contained data but was subsequently erased does not cause a Tag to wrongly indicate the presence of User memory, and vice versa.

- Bit 16_h: An XPC_W1 indicator (XI). If bit 16_h is deasserted then the Tag either does not implement an XPC_W1 or the XPC_W1 is zero-valued, in which case the Tag shall backscatter its StoredPC or PacketPC, but not an XPC_W1, in response to an *ACK* (see Table 6.14). If bit 16_h is asserted then the Tag implements an XPC_W1 and one or more bits of XPC_W1 have nonzero values. In this latter case the Tag shall backscatter its XPC_W1 immediately after the StoredPC or PacketPC during inventory.

  If a Tag implements an XPC_W1 then at power-up, and prior to calculating its StoredCRC (see 6.3.2.1.2.1), the Tag shall compute the bitwise logical OR of its XPC_W1 and map the computed value into bit 16_h (i.e. into the XI). A Tag shall use this computed XI value in its StoredCRC calculation. If an Interrogator recommissions the Tag (see 6.3.2.10) then the Tag shall recompute and remap its XI into bit 16_h after recommissioning. After recomputing the XI the StoredCRC may be incorrect until the Interrogator power cycles the Tag. The XI bit shall not be directly writeable by an Interrogator — when an Interrogator writes the StoredPC the Tag shall ignore the data value that the Interrogator provides for bit 16_h.

- Bits 17_h – $1F_h$: A numbering system identifier (NSI). The MSB of the NSI is stored in memory location 17_h. If bit 17_h contains a logical 0, then the application is referred to as an EPCglobal™ Application and bits 18_h – $1F_h$ shall be as defined in the EPCglobal™ Tag Data Standards. If bit 17_h contains a logical 1, then the application is referred to as a non-EPCglobal™ Application and bits 18_h – $1F_h$ shall contain the entire AFI defined in ISO/IEC 15961. The default value for bits 18_h – $1F_h$ is 00000000_2.

The default (unprogrammed) StoredPC value shall be 0000_h.

If an Interrogator changes the EPC length (via a memory write operation), and if it wishes the Tag to subsequently backscatter the new EPC length, then it must write a new EPC length field into the Tag's StoredPC. Note: After changing the EPC length field an Interrogator should power-cycle the Tag to ensure a correct StoredCRC.

A Tag shall backscatter an error code (see Annex I) if an Interrogator attempts to write an EPC length field that the Tag does not support.

The PacketPC differs from the StoredPC in its EPC length field, which a Tag shall adjust to match the length of the backscattered data that follow the PC word. Specifically, if XI is asserted but XEB is not asserted then the Tag backscatters an XPC_W1 before the EPC, so the Tag shall add one to (i.e. increment) its EPC length field. If both XI and XEB are asserted then the Tag backscatters both an XPC_W1 and an XPC_W2 before the EPC, so the Tag shall add two to (i.e. double increment) its EPC length field. Because Tags that support XPC functionality have a maximum EPC length field of 11101_2, double incrementing will increase the value to 11111_2. A Tag shall not, under any circumstances, allow its EPC length field to roll over to 00000_2. Note that incrementing or double incrementing the EPC length field does not alter the values stored in bits 10_h – 14_h of EPC memory; rather, a Tag increments the EPC length field in the backscattered PacketPC but leaves the memory contents unaltered.

Tags that do not support XPC functionality need not implement a PacketPC.

If an Interrogator that does not support an XPC_W1 receives a Tag reply with XI asserted then the Interrogator shall treat the Tag's reply as though its CRC-16 integrity check had failed.

If an Interrogator that does not support an XPC_W2 receives a Tag reply with XEB asserted then the Interrogator shall treat the Tag's reply as though its CRC-16 integrity check had failed.

During truncated replies a Tag substitutes 00000_2 for the PC word — see Table 6.14 and 6.3.2.11.1.1.

6.3.2.1.2.3 EPC for an EPCglobal™ Application

The EPC structure for an EPCglobal™ Application shall be as defined in the EPCglobal™ Tag Data Standards.

6.3.2.1.2.4 EPC for a non-EPCglobal™ Application

The EPC structure for a non-EPCglobal™ Application shall be as defined in ISO/IEC 15961.

6.3.2.1.2.5 Extended Protocol Control (XPC) word or words (optional)

A Tag may implement an XPC_W1 logically located at addresses 210_h to $21F_h$ of EPC memory. If a Tag implements an XPC_W1 then it may additionally implement an XPC_W2 logically located at address 220_h to $22F_h$ of EPC memory. A Tag shall not implement an XPC_W2 without also implementing an XPC_W1. If implemented, these XPC words shall be exactly 16 bits in length and are stored MSB first. If a Tag does not support one or both of these optional XPC words then the specified memory locations need not exist.

A Tag shall not implement any non-XPC memory element at EPC memory locations 210_h to $22F_h$, inclusive. This requirement shall apply both to Tags that support an XPC word or words and to those that do not.

If a Tag supports recommissioning (see 6.3.2.10) then it shall implement an XPC_W1.

If a Tag implements an XPC_W2 then, at power-up, the Tag shall compute the bitwise logical OR of the XPC_W2 and map the computed value into bit 210_h of EPC memory (i.e. into the most significant bit of XPC_W1). Bit 210_h is denoted the XPC Extension Bit (XEB). If a Tag does not implement an XPC_W2 then the XEB shall be zero.

The remainder of this section 6.3.2.1.2.5 assumes that a Tag implements an XPC_W1 and an XPC_W2.

When this document refers to the 3 least-significant-bits (LSBs) of XPC_W1 it specifically means locations $21D_h$, $21E_h$, and $21F_h$ of EPC memory. The 3 LSBs of XPC_W1 indicate whether and how a Tag was recommissioned.

For virgin Class-1 Tags the 3 LSBs of XPC_W1 shall be zero-valued. A Tag writes a nonzero value to one or more of these 3 LSBs during Tag recommissioning (see 6.3.2.10). For Class-1 Tags all the other bits in XPC_W1, namely EPC memory locations 210_h to $21C_h$, inclusive, as well as all the bits in XPC_W2, shall be RFU and zero-valued. Tag vendors and end users shall not use these RFU bits for proprietary purposes.

The 3 LSBs of XPC_W1 are not writeable using a *Write* or *BlockWrite*, nor erasable using a *BlockErase*. They can only be asserted by a *Kill* command, meaning that the Tag asserts these bits upon receiving a valid *Kill* command sequence with asserted recommissioning bits (see 6.3.2.11.3.4). A Tag shall not modify the 3 LSBs of XPC_W1 except during Tag recommissioning.

If a Class-1 Tag receives a *Write*, *BlockWrite*, or *BlockErase* command that attempts to write to XPC_W1 or to XPC_W2 it responds with an error code (see Annex I).

An Interrogator may issue a *Select* command (see 6.3.2.11.1) with a Mask that covers all or part of XPC_W1 and/or XPC_W2. For example, Mask may have the value 000_2 for the 3 LSBs of XPC_W1, in which case recommissioned Tags will be non-matching, as will Tags that do not implement recommissioning; only recommissionable Tags that have not yet been recommissioned will be matching.

An Interrogator may read a Tag's XPC_W1 and XPC_W2 using a *Read* command (see 6.3.2.11.3.2).

Bit E_h of XPC_W1 (EPC memory location 211_h) is reserved for use as a protocol functionality indicator. See 2.4.

The following encoding provides a general mapping between the 3 XPC_W1 LSBs and a Tag's recommissioned status. Table 6.15 provides a detailed mapping between these 3 LSBs and a Tag's recommissioned status:

- **An asserted LSB** (EPC memory location $21F_h$) indicates that block permalocking has been disabled, and any blocks of User memory that were previously block permalocked are no longer block permalocked. An asserted LSB also indicates that the *BlockPermalock* command has been disabled. If a Tag did not implement block permalocking prior to recommissioning then block permalocking shall remain disabled.
- **An asserted 2SB** (EPC memory location $21E_h$) indicates that User memory has been rendered inaccessible. The 2SB has precedence over the LSB — if both are asserted then User memory is inaccessible.
- **An asserted 3SB** (EPC memory location $21D_h$) indicates that the Tag has unlocked its EPC, TID, and User memory banks. The Tag has also write-unlocked its kill and access passwords, and rendered the kill and access passwords permanently unreadable regardless of the values of the Tag's lock bits (see 6.3.2.11.3.5). If an Interrogator subsequently attempts to read the Tag's kill or access passwords the Tag backscatters an error code (see Annex I). Note that portions or banks of Tag memory, if factory set and locked, may not be unlockable regardless of recommissioning. Note also that an Interrogator may subsequently re-lock any memory banks that have been unlocked by recommissioning.

Table 6.15 – XPC_W1 LSBs and a Tag's recommissioned status

LSBs of XPC_W1	Tag Status	Notes
000	1. The Tag has not been recommissioned	1. A Class-1 Tag's XI bit is deasserted
001	1. Block permalocking has been disabled, and any blocks of User memory that were previously block permalocked are no longer block permalocked 2. The *BlockPermalock* command has been disabled	1. The lock bits are the sole determinant of the User memory bank's lock status (see 6.3.2.11.3.5)
010	1. The User memory bank has been rendered inaccessible 2. A Tag whose XPC_W1 LSBs are 010 acts identically to one whose XPC_W1 LSBs are 011	1. The Tag deasserts its UMI bit 2. User memory is inaccessible, so block permalocking and the *BlockPermalock* command are disabled
011	1. The User memory bank has been rendered inaccessible 2. A Tag whose XPC_W1 LSBs are 011 acts identically to one whose XPC_W1 LSBs are 010	1. The Tag deasserts its UMI bit 2. User memory is inaccessible, so block permalocking and the *BlockPermalock* command are disabled
100	1. The EPC, TID, and User memory banks have been un-locked 2. The kill and access passwords have been write-unlocked 3. The kill and access passwords have been rendered permanently unreadable, regardless of the status of the Tag's lock bits (see 6.3.2.11.3.5)	1. If the Tag supports block permalocking then the *BlockPermalock* command remains enabled 2. Any blocks of User memory that were previously block permalocked remain block permalocked, and vice versa 3. Portions or banks of Tag memory, if factory set and locked, may not be unlockable regardless of recom-missioning 4. If an Interrogator attempts to read the Tag's kill or access passwords the Tag responds by backscat-tering an error code (see Annex I) 5. The kill and/or access passwords, as well as one or more memory blocks and/or banks, may be changed and/or relocked after recommissioning
101	1. Block permalocking has been disabled, and any blocks of User memory that were previously block permalocked are no longer block permalocked 2. The *BlockPermalock* command has been disabled 3. The EPC, TID, and User memory banks have been un-locked 4. The kill and access passwords have been write-unlocked 5. The kill and access passwords have been rendered permanently unreadable, regardless of the status of the Tag's lock bits (see 6.3.2.11.3.5)	1. The lock bits are the sole determinant of the User memory bank's lock status (see 6.3.2.11.3.5) 2. Portions or banks of Tag memory, if factory set and locked, may not be unlockable regardless of recom-missioning 3. If an Interrogator attempts to read the Tag's kill or access passwords the Tag responds by backscat-tering an error code (see Annex I) 4. The kill and/or access passwords, as well as one or more memory blocks and/or banks, may be changed and/or relocked after recommissioning
110	1. The EPC and TID memory banks have been unlocked 2. The kill and access passwords have been write-unlocked 3. The kill and access passwords have been rendered permanently unreadable, regardless of the status of the Tag's lock bits (see 6.3.2.11.3.5) 4. The User memory bank has been rendered inaccessible 5. A Tag whose XPC_W1 LSBs are 110 acts identically to one whose XPC_W1 LSBs are 111	1. Portions or banks of Tag memory, if factory set and locked, may not be unlockable regardless of recom-missioning 2. If an Interrogator attempts to read the Tag's kill or access passwords the Tag responds by backscat-tering an error code (see Annex I) 3. The kill and/or access passwords, as well as one or more memory blocks and/or banks, may be changed and/or relocked after recommissioning 4. The Tag deasserts its UMI bit 5. User memory is inaccessible, so block permalocking and the *BlockPermalock* command are disabled
111	1. The EPC and TID memory banks have been unlocked 2. The kill and access passwords have been write-unlocked 3. The kill and access passwords have been rendered permanently unreadable, regardless of the status of the Tag's lock bits (see 6.3.2.11.3.5) 4. The User memory bank has been rendered inaccessible 5. A Tag whose XPC_W1 LSBs are 111 acts identically to one whose XPC_W1 LSBs are 110	1. Portions or banks of Tag memory, if factory set and locked, may not be unlockable regardless of recom-missioning 2. If an Interrogator attempts to read the Tag's kill or access passwords the Tag responds by backscat-tering an error code (see Annex I) 3. The kill and/or access passwords, as well as one or more memory blocks and/or banks, may be changed and/or relocked after recommissioning 4. The Tag deasserts its UMI bit 5. User memory is inaccessible, so block permalocking and the *BlockPermalock* command are disabled

6.3.2.1.3 TID Memory

TID memory locations 00_h to 07_h shall contain one of two ISO/IEC 15963 class-identifier values — either $E0_h$ or $E2_h$. TID memory locations above 07_h shall be defined according to the registration authority defined by this class-identifier value and shall contain, at a minimum, sufficient identifying information for an Interrogator to uniquely identify the custom commands and/or optional features that a Tag supports. TID memory may also contain Tag- and vendor-specific data (for example, a Tag serial number).

Note: The Tag manufacturer assigns the class-identifier value (i.e. $E0_h$ or $E2_h$), for which ISO/IEC 15963 defines the registration authorities. The class-identifier does not specify the Application. If the class identifier is $E0_h$, TID memory locations 08_h to $0F_h$ contain an 8-bit manufacturer identifier, TID memory locations 10_h to $3F_h$ contain a 48-bit Tag serial number (assigned by the Tag manufacturer), the composite 64-bit Tag ID (i.e. TID memory 00_h to $3F_h$) is unique among all classes of Tags defined in ISO/IEC 15693, and TID memory is permalocked at the time of manufacture. If the class identifier is $E2_h$, TID memory location 08_h to 13_h contain a 12-bit Tag mask-designer identifier (obtainable from the registration authority), TID memory locations 14_h to $1F_h$ contain a vendor-defined 12-bit Tag model number, and the usage of TID memory locations above $1F_h$ is defined in the EPCglobal™ Tag Data Standards.

6.3.2.1.4 User Memory

A Tag may contain User memory. User memory allows user-specific data storage.

If a Tag's User memory has not yet been programmed then the 5 LSBs of the first byte of User memory (i.e. memory addresses 03_h to 07_h) shall have the default value 00000_2.

During recommissioning an Interrogator may instruct a Tag to render its User memory inaccessible, causing the entire memory bank to become unreadable, unwriteable, and unselectable. A Tag with inaccessible User memory shall function as though its User memory bank no longer exists.

6.3.2.1.4.1 User memory for an EPCglobal™ Application

If User memory is included on a Tag then its encoding shall be as defined in the EPCglobal™ Tag Data Standards.

6.3.2.1.4.2 User memory for a non-EPCglobal™ Application

If User memory is included on a Tag then its encoding shall be as defined in ISO/IEC 15961 and ISO/IEC 15962.

6.3.2.2 Sessions and inventoried flags

Interrogators shall support and Tags shall provide 4 sessions (denoted S0, S1, S2, and S3). Tags shall participate in one and only one session during an inventory round. Two or more Interrogators can use sessions to independently inventory a common Tag population. The sessions concept is illustrated in Figure 6.18.

Tags shall maintain an independent **inventoried** flag for each session. Each of the four **inventoried** flags has two values, denoted *A* and *B*. At the beginning of each and every inventory round an Interrogator chooses to inventory either *A* or *B* Tags in one of the four sessions. Tags participating in an inventory round in one session shall neither use nor modify the **inventoried** flag for a different session. The **inventoried** flags are the only resource a Tag provides separately and independently to a given session; all other Tag resources are shared among sessions.

After singulating a Tag an Interrogator may issue a command that causes the Tag to invert its **inventoried** flag for that session (i.e. $A \to B$ or $B \to A$).

The following example illustrates how two Interrogators can use sessions and **inventoried** flags to independently and completely inventory a common Tag population, on a time-interleaved basis:

- Interrogator #1 powers-on, then
  - It initiates an inventory round during which it singulates *A* Tags in session S2 to *B*,
  - It powers off.
- Interrogator #2 powers-on, then
  - It initiates an inventory round during which it singulates *B* Tags in session S3 to *A*,
  - It powers off.

This process repeats until Interrogator #1 has placed all Tags in session S2 into *B*, after which it inventories the Tags in session S2 from *B* back to *A*. Similarly, Interrogator #2 places all Tags in session S3 into *A*, after which it inventories the Tags in session S3 from *A* back to *B*. By this multi-step procedure each Interrogator can independently inventory all Tags in its field, regardless of the initial state of their **inventoried** flags.

A Tag's **inventoried** flags shall have the persistence times shown in Table 6.16. A Tag shall power-up with its **inventoried** flags set as follows:

- The S0 **inventoried** flag shall be set to *A*.
- The S1 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the flag was set longer in the past than its persistence time, in which case the Tag shall power-up with its S1 **inventoried** flag set to *A*. Because the S1 **inventoried** flag is not automatically refreshed, it may revert from *B* to *A* even when the Tag is powered.
- The S2 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the Tag has lost power for a time greater than its persistence time, in which case the Tag shall power-up with the S2 **inventoried** flag set to *A*.
- The S3 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the Tag has lost power for a time greater than its persistence time, in which case the Tag shall power-up with its S3 **inventoried** flag set to *A*.

A Tag shall set any of its inventoried flags to either *A* or *B* in 2 ms or less, regardless of the initial flag value. A Tag shall refresh its S2 and S3 flags while powered, meaning that every time a Tag loses power its S2 and S3 **inventoried** flags shall have the persistence times shown in Table 6.16.

A Tag shall not change the value of its S1 **inventoried** flag from *B* to *A*, as the result of a persistence timeout, while the Tag is participating in an inventory round, is in the midst of being inventoried, or is in the midst of being accessed. If a Tag's S1 flag persistence time expires during an inventory round then the Tag shall change the flag to *A* only (i) as instructed by an Interrogator (e.g. by a *QueryAdjust* or *QueryRep* with matching session at the end of an inventory or access operation), or (ii) at the end of the round (e.g. upon receiving a *Select* or *Query*). In case (i), if the Tag's S1 flag persistence time expires while the Tag is in the midst of being inventoried or accessed then the Tag shall change the flag to *A* at the end of the inventory or access operation. In case (ii), the Tag shall invert its S1 flag prior to evaluating the *Select* or *Query*.

6.3.2.3 Selected flag

Tags shall implement a selected flag, **SL**, which an Interrogator may assert or deassert using a *Select* command. The Sel parameter in the *Query* command allows an Interrogator to inventory Tags that have **SL** either asserted or deasserted (i.e. **SL** or ~**SL**), or to ignore the flag and inventory Tags regardless of their **SL** value. **SL** is not associated with any particular session; **SL** may be used in any session, and is common to all sessions.

A Tag's **SL** flag shall have the persistence times shown in Table 6.16. A Tag shall power-up with its **SL** flag either asserted or deasserted, depending on the stored value, unless the Tag has lost power for a time greater than the **SL** persistence time, in which case the Tag shall power-up with its **SL** flag deasserted (set to ~**SL**). A Tag shall be capable of asserting or deasserting its **SL** flag in 2 ms or less, regardless of the initial flag value. A Tag shall refresh its **SL** flag when powered, meaning that every time a Tag loses power its **SL** flag shall have the persistence times shown in Table 6.16.

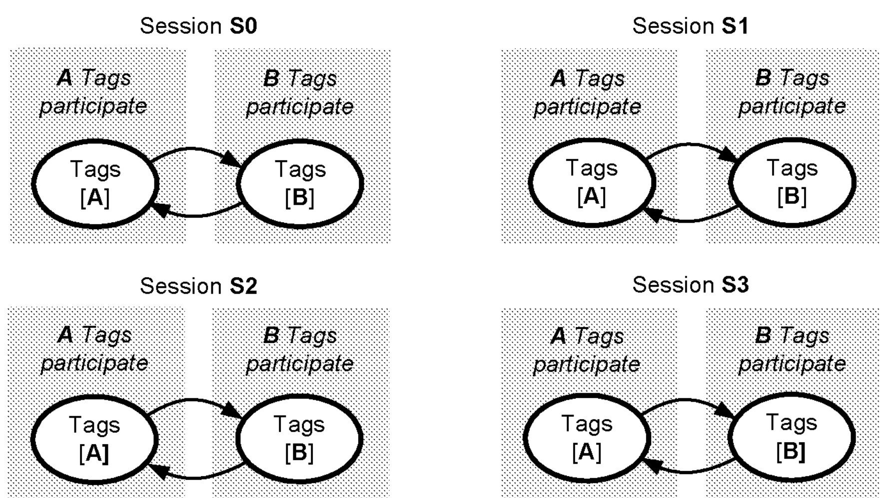


Figure 6.18 – Session diagram

Table 6.16 – Tag flags and persistence values

Flag	Required persistence
S0 **inventoried** flag	Tag energized: Indefinite Tag not energized: None
S1 **inventoried** flag[1]	Tag energized: Nominal temperature range: 500ms < persistence < 5s Extended temperature range: Not specified Tag not energized: Nominal temperature range: 500ms < persistence < 5s Extended temperature range: Not specified
S2 **inventoried** flag[1]	Tag energized: Indefinite Tag not energized: Nominal temperature range: 2s < persistence Extended temperature range: Not specified
S3 **inventoried** flag[1]	Tag energized: Indefinite Tag not energized: Nominal temperature range: 2s < persistence Extended temperature range: Not specified
Selected (**SL**) flag[1]	Tag energized: Indefinite Tag not energized: Nominal temperature range: 2s < persistence Extended temperature range: Not specified

Note 1: For a randomly chosen and sufficiently large Tag population, 95% of the Tag persistence times shall meet the persistence requirement, with a 90% confidence interval.

6.3.2.4 Tag states and slot counter

Tags shall implement the states and the slot counter shown in Figure 6.19. Annex B shows the associated state-transition tables; Annex C shows the associated command-response tables.

6.3.2.4.1 Ready state

Tags shall implement a **ready** state. **Ready** can be viewed as a "holding state" for energized Tags that are neither killed nor currently participating in an inventory round. Upon entering an energizing RF field a Tag that is not killed shall enter **ready**. The Tag shall remain in **ready** until it receives a *Query* command (see 6.3.2.11.2.1) whose inventoried parameter (for the session specified in the *Query*) and sel parameter match its current flag values. Matching Tags shall draw a *Q*-bit number from their RNG (see 6.3.2.5), load this number into their slot counter, and transition to the **arbitrate** state if the number is nonzero, or to the **reply** state if the number is zero. If a Tag in any state except **killed** loses power it shall return to **ready** upon regaining power.

6.3.2.4.2 Arbitrate state

Tags shall implement an **arbitrate** state. **Arbitrate** can be viewed as a "holding state" for Tags that are participating in the current inventory round but whose slot counters (see 6.3.2.4.8) hold nonzero values. A Tag in **arbitrate** shall decrement its slot counter every time it receives a *QueryRep* command (see 6.3.2.11.2.3) whose session parameter matches the session for the inventory round currently in progress, and it shall transition to the **reply** state and backscatter an RN16 when its slot counter reaches 0000_h. Tags that return to **arbitrate** (for example, from the **reply** state) with a slot value of 0000_h shall decrement their slot counter from 0000_h to $7FFF_h$ at the next *QueryRep* (with matching session) and, because their slot value is now nonzero, shall remain in **arbitrate**.

6.3.2.4.3 Reply state

Tags shall implement a **reply** state. Upon entering **reply** a Tag shall backscatter an RN16. If the Tag receives a valid acknowledgement (*ACK*) it shall transition to the **acknowledged** state, backscattering the reply shown in Table 6.14. If the Tag fails to receive an *ACK* within time $T_{2(max)}$, or receives an invalid *ACK* or an *ACK* with an erroneous RN16, it shall return to **arbitrate**. Tag and Interrogator shall meet all timing requirements specified in Table 6.13.

6.3.2.4.4 Acknowledged state

Tags shall implement an **acknowledged** state. A Tag in **acknowledged** may transition to any state except **killed**, depending on the received command (see Figure 6.19). If a Tag in the **acknowledged** state receives a valid *ACK* containing the correct RN16 it shall re-backscatter the reply shown in Table 6.14. If a Tag in the **acknowledged** state fails to receive a valid command within time $T_{2(max)}$ it shall return to **arbitrate**. Tag and Interrogator shall meet all timing requirements specified in Table 6.13.

6.3.2.4.5 Open state

Tags shall implement an **open** state. A Tag in the **acknowledged** state whose access password is nonzero shall transition to **open** upon receiving a *Req_RN* command, backscattering a new RN16 (denoted handle) that the Interrogator shall use in subsequent commands and the Tag shall use in subsequent replies. Tags in the **open** state can execute all access commands except *Lock* and *BlockPermalock*. A Tag in **open** may transition to any state except **acknowledged**, depending on the received command (see Figure 6.19). If a Tag in the **open** state receives a valid *ACK* containing the correct handle it shall re-backscatter the reply shown in Table 6.14. Tag and Interrogator shall meet all timing requirements specified in Table 6.13 except $T_{2(max)}$; in the **open** state the maximum delay between Tag response and Interrogator transmission is unrestricted.

6.3.2.4.6 Secured state

Tags shall implement a **secured** state. A Tag in the **acknowledged** state whose access password is zero shall transition to **secured** upon receiving a *Req_RN* command, backscattering a new RN16 (denoted handle) that the Interrogator shall use in subsequent commands and the Tag shall use in subsequent replies. A Tag in the **open** state whose access password is nonzero shall transition to **secured** upon receiving a valid *Access* command sequence, maintaining the same handle that it previously backscattered when it transitioned from the **acknowledged** state to the **open** state. Tags in the **secured** state can execute all access commands. A Tag in **secured** may transition to any state except **open** or **acknowledged**, depending on the received command (see Figure 6.19). If a Tag in the **secured** state receives a valid *ACK* containing the correct handle it shall re-backscatter the reply shown in Table 6.14. Tag and Interrogator shall meet all timing requirements specified in Table 6.13 except $T_{2(max)}$; in the **secured** state the maximum delay between Tag response and Interrogator transmission is unrestricted.

6.3.2.4.7 Killed state

Tags shall implement a **killed** state. A Tag in either the **open** or **secured** states shall enter the **killed** state upon receiving a valid *Kill* command sequence (see 6.3.2.11.3.4) with a valid nonzero kill password, zero-valued Recom bits (see 6.3.2.10), and valid handle. If a Tag does not implement recommissioning then it treats nonzero Recom bits as though Recom = 0. *Kill* permanently disables a Tag. Upon entering the **killed** state a Tag shall notify the Interrogator that the kill operation was successful, and shall not respond to an Interrogator thereafter. Killed Tags shall remain in the **killed** state under all circumstances, and shall immediately enter killed upon subsequent power-ups. Killing a Tag is irreversible.

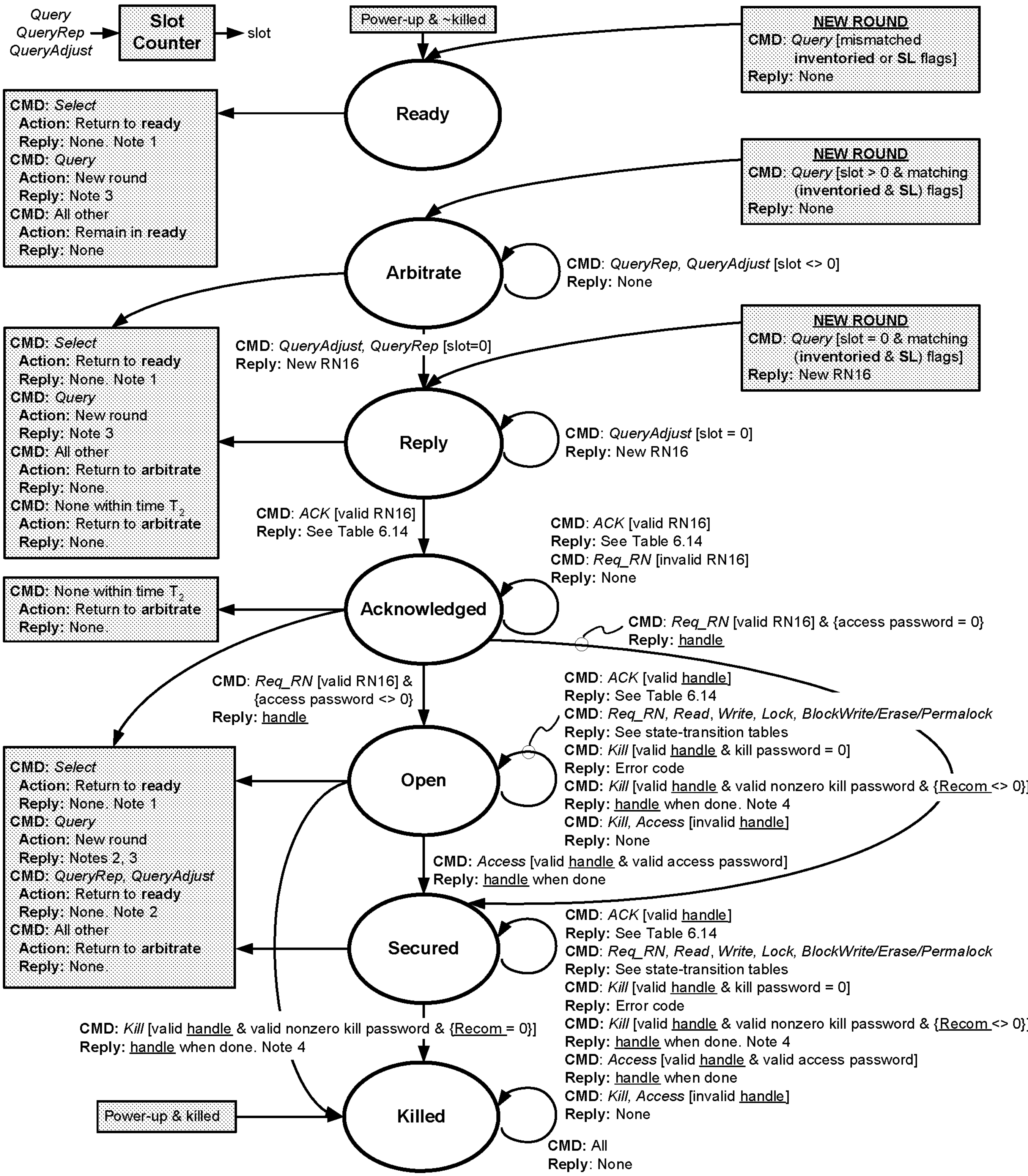
6.3.2.4.8 Slot counter

Tags shall implement a 15-bit slot counter. Upon receiving a *Query* or *QueryAdjust* command a Tag shall preload into its slot counter a value between 0 and 2^Q–1, drawn from the Tag's RNG (see 6.3.2.5). *Q* is an integer in the range (0, 15). A *Query* specifies *Q*; a *QueryAdjust* may modify *Q* from the prior *Query*.

Tags in the **arbitrate** state decrement their slot counter every time they receive a *QueryRep* with matching session, transitioning to the **reply** state and backscattering an RN16 when their slot counter reaches 0000_h. Tags whose slot counter reached 0000_h, who replied, and who were not acknowledged (including Tags that responded to an original *Query* and were not acknowledged) shall return to **arbitrate** with a slot value of 0000_h and shall decrement this slot value from 0000_h to $7FFF_h$ at the next *QueryRep*. The slot counter shall be capable of continuous counting, meaning that, after the slot counter rolls over to $7FFF_h$ it begins counting down again, thereby effectively preventing subsequent replies until the Tag loads a new random value into its slot counter. See also Annex J.

6.3.2.5 Tag random or pseudo-random number generator

Tags shall implement a random or pseudo-random number generator (RNG). The RNG shall meet the following randomness criteria independent of the strength of the energizing field, the R=>T link rate, and the data stored in the Tag (including but not limited to the StoredPC, XPC word or words, EPC, and StoredCRC). Tags shall generate 16-bit random or pseudo-random numbers (RN16) using the RNG, and shall have the ability to extract *Q*-bit subsets from an RN16 to preload the Tag's slot counter (see 6.3.2.4.8). Tags shall have the ability to temporarily store at least two RN16s while powered, to use, for example, as a handle and a 16-bit cover-code during password transactions (see Figure 6.23 or Figure 6.25).



NOTES

1. *Select*: Assert/deassert **SL** or set **inventoried** to *A* or *B*.
2. *Query*: *A*→ *B* or *B*→ *A* if the new session matches the prior session; otherwise no change to the **inventoried** flag.
   QueryRep/QueryAdjust: *A*→ *B* or *B*→ *A* if the session matches the prior *Query*; otherwise, the command is invalid and ignored by the Tag.
3. *Query* starts a new round and may change the session. Tags may go to **ready**, **arbitrate**, or **reply**.
4. If a Tag does not implement recommissioning then the Tag treats nonzero Recom bits as though Recom = 0.

Figure 6.19 – Tag state diagram

Probability of a single RN16: The probability that any RN16 drawn from the RNG has value RN16 = *j*, for any *j*, shall be bounded by $0.8/2^{16} < P(RN16 = j) < 1.25/2^{16}$.

Probability of simultaneously identical sequences: For a Tag population of up to 10,000 Tags, the probability that any two or more Tags simultaneously generate the same sequence of RN16s shall be less than 0.1%, regardless of when the Tags are energized.

Probability of predicting an RN16: An RN16 drawn from a Tag's RNG 10ms after the end of T_r in Figure 6.3 shall not be predictable with a probability greater than 0.025% if the outcomes of prior draws from the RNG, performed under identical conditions, are known.

This protocol recommends that Interrogators wait 10ms after T_r in Figure 6.3 or T_{hr} in Figure 6.5 before issuing passwords to Tags.

6.3.2.6 Managing Tag populations

Interrogators manage Tag populations using the three basic operations shown in Figure 6.20. Each of these operations comprises one or more commands. The operations are defined as follows:

a) **Select:** The process by which an Interrogator selects a Tag population for inventory and access. Interrogators may use one or more *Select* commands to select a particular Tag population prior to inventory.

b) **Inventory:** The process by which an Interrogator identifies Tags. An Interrogator begins an inventory round by transmitting a *Query* command in one of four sessions. One or more Tags may reply. The Interrogator detects a single Tag reply and requests the PC word, optional XPC word or words, EPC, and CRC-16 from the Tag. An inventory round operates in one and only one session at a time. Annex E shows an example of an Interrogator inventorying and accessing a single Tag.

c) **Access:** The process by which an Interrogator transacts with (reads from or writes to) individual Tags. An individual Tag must be uniquely identified prior to access. Access comprises multiple commands, some of which employ one-time-pad based cover-coding of the R=>T link.

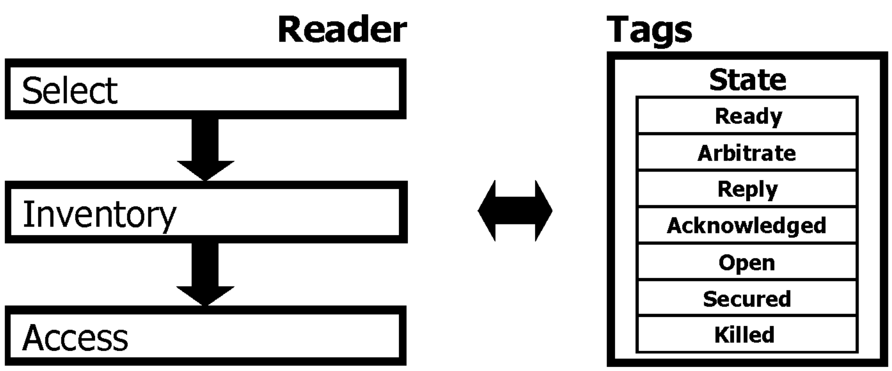


Figure 6.20 – Interrogator/Tag operations and Tag state

6.3.2.7 Selecting Tag populations

The selection process employs a single command, *Select*, which an Interrogator may apply successively to select a particular Tag population based on user-defined criteria, enabling union (U), intersection (∩), and negation (~) based Tag partitioning. Interrogators perform U and ∩ operations by issuing successive *Select* commands. *Select* can assert or deassert a Tag's **SL** flag, or it can set a Tag's **inventoried** flag to either *A* or *B* in any one of the four sessions. *Select* contains the parameters Target, Action, MemBank, Pointer, Length, Mask, and Truncate.

- Target and Action indicate whether and how a *Select* modifies a Tag's **SL** or **inventoried** flag, and in the case of the **inventoried** flag, for which session. A *Select* that modifies **SL** shall not modify **inventoried**, and vice versa.
- MemBank specifies if the mask applies to EPC, TID, or User memory. *Select* commands apply to a single memory bank. Successive *Selects* may apply to different memory banks.
- Pointer, Length, and Mask: Pointer and Length describe a memory range. Mask, which must be Length bits long, contains a bit string that a Tag compares against the specified memory range.
- Truncate specifies whether a Tag backscatters its entire EPC, or only that portion of the EPC immediately

following Mask. Truncated EPCs are always followed by the Tag's StoredCRC (see Table 6.14). A Tag does not recalculate its StoredCRC for a truncated reply.

By issuing multiple identical *Select* commands an Interrogator can asymptotically single out all Tags matching the selection criteria even though Tags may undergo short-term RF fades.

A *Query* command uses **inventoried** and **SL** to decide which Tags participate in an inventory. Interrogators may inventory and access **SL** or ~**SL** Tags, or they may choose to ignore the **SL** flag entirely.

6.3.2.8 Inventorying Tag populations

The inventory command set includes *Query*, *QueryAdjust*, *QueryRep*, *ACK*, and *NAK. Query* initiates an inventory round and decides which Tags participate in the round ("inventory round" is defined in 4.1).

Query contains a slot-count parameter *Q*. Upon receiving a *Query* participating Tags pick a random value in the range (0, 2^Q–1), inclusive, and load this value into their slot counter. Tags that pick a zero transition to the **reply** state and reply immediately. Tags that pick a nonzero value transition to the **arbitrate** state and await a *QueryAdjust* or a *QueryRep* command. Assuming that a single Tag replies, the query-response algorithm proceeds as follows:

a) The Tag backscatters an RN16 as it enters **reply**,
b) The Interrogator acknowledges the Tag with an *ACK* containing this same RN16,
c) The acknowledged Tag transitions to the **acknowledged** state, backscattering the reply shown in Table 6.14,
d) The Interrogator issues a *QueryAdjust* or *QueryRep,* causing the identified Tag to invert its **inventoried** flag (i.e. *A*→*B* or *B*→*A*) and transition to **ready**, and potentially causing another Tag to initiate a query-response dialog with the Interrogator, starting in step (a), above.

If the Tag fails to receive the *ACK* in step (b) within time T_2 (see Figure 6.16), or receives the *ACK* with an erroneous RN16, it returns to **arbitrate**.

If multiple Tags reply in step (a) but the Interrogator, by detecting and resolving collisions at the waveform level, can resolve an RN16 from one of the Tags, the Interrogator can *ACK* the resolved Tag. Unresolved Tags receive erroneous RN16s and return to **arbitrate** without backscattering the reply shown in Table 6.14.

If the Interrogator sends a valid *ACK* (i.e. an *ACK* containing the correct RN16) to the Tag in the **acknowledged** state, the Tag re-backscatters the reply shown in Table 6.14.

At any point the Interrogator may issue a *NAK*, in response to which all Tags in the inventory round that receive the *NAK* return to **arbitrate** without changing their **inventoried** flag.

After issuing a *Query* to initiate an inventory round, the Interrogator typically issues one or more *QueryAdjust* or *QueryRep* commands. *QueryAdjust* repeats a previous *Query* and may increment or decrement *Q*, but does not introduce new Tags into the round. *QueryRep* repeats a previous *Query* without changing any parameters and without introducing new Tags into the round. An inventory round can contain multiple *QueryAdjust* or *QueryRep* commands. At some point the Interrogator will issue a new *Query*, thereby starting a new inventory round.

Tags in the **arbitrate** or **reply** states that receive a *QueryAdjust* first adjust *Q* (increment, decrement, or leave unchanged), then pick a random value in the range (0, 2^Q–1), inclusive, and load this value into their slot counter. Tags that pick zero transition to the **reply** state and reply immediately. Tags that pick a nonzero value transition to the **arbitrate** state and await a *QueryAdjust* or a *QueryRep* command.

Tags in the **arbitrate** state decrement their slot counter every time they receive a *QueryRep*, transitioning to the **reply** state and backscattering an RN16 when their slot counter reaches 0000_h. Tags whose slot counter reached 0000_h, who replied, and who were not acknowledged (including Tags that responded to the original *Query* and were not acknowledged) return to **arbitrate** with a slot value of 0000_h and decrement this slot value from 0000_h to $7FFF_h$ at the next *QueryRep*, thereby effectively preventing subsequent replies until the Tag loads a new random value into its slot counter.

Although Tag inventory is based on a random protocol, the *Q*-parameter affords network control by allowing an Interrogator to regulate the probability of Tag responses. *Q* is an integer in the range (0,15); thus, the associated Tag-response probabilities range from $2^0 = 1$ to $2^{-15} = 0.000031$.

Annex D describes an exemplary Interrogator algorithm for choosing *Q*.

The scenario outlined above assumed a single Interrogator operating in a single session. However, as described in 6.3.2.2, an Interrogator can inventory a Tag population in one of four sessions. Furthermore, as described in 6.3.2.11.2, the *Query*, *QueryAdjust*, and *QueryRep* commands each contain a <u>session</u> parameter. How a Tag responds to these commands varies with the command, <u>session</u> parameter, and Tag state, as follows:

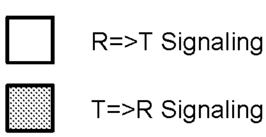
- *Query*: A *Query* command starts an inventory round and chooses the session for the round. Tags in any state except **killed** execute a *Query*, starting a new round in the specified session and transitioning to **ready**, **arbitrate**, or **reply**, as appropriate (see Figure 6.19).
  - If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose <u>session</u> parameter matches the prior session it inverts its **inventoried** flag (i.e. *A*→*B* or *B*→*A*) for the session before it evaluates whether to transition to **ready**, **arbitrate**, or **reply**.
  - If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose <u>session</u> parameter does not match the prior session it leaves its **inventoried** flag for the prior session unchanged as it evaluates whether to transition to **ready**, **arbitrate**, or **reply**.
- *QueryAdjust, QueryRep*: Tags in any state except **ready** or **killed** execute a *QueryAdjust* or *QueryRep* command if, and only if, (i) the <u>session</u> parameter in the command matches the <u>session</u> parameter in the *Query* that started the round, and (ii) the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively). Tags ignore a *QueryAdjust* or *QueryRep* with mismatched session.
  - If a Tag in the **acknowledged**, **open**, or **secured** states receives a *QueryAdjust* or *QueryRep* whose <u>session</u> parameter matches the <u>session</u> parameter in the prior *Query*, and the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively), it inverts its **inventoried** flag (i.e. *A*→*B* or *B*→*A*) for the current session then transitions to **ready**.

To illustrate an inventory operation, consider a specific example: Assume a population of 64 powered Tags in the **ready** state. An Interrogator first issues a *Select* to select a subpopulation of Tags. Assume that 16 Tags match the selection criteria. Further assume that 12 of the 16 selected Tags have their **inventoried** flag set to *A* in session S0. The Interrogator issues a *Query* specifying (**SL**, *Q* = 4, S0**,** *A*). Each of the 12 Tags picks a random number in the range (0,15) and loads the value into its slot counter. Tags that pick a zero respond immediately. The *Query* has 3 possible outcomes:

a) **No Tags reply:** The Interrogator may issue another *Query*, or it may issue a *QueryAdjust* or *QueryRep*.

b) **One Tag replies** (see Figure 6.21): The Tag transitions to the **reply** state and backscatters an RN16. The Interrogator acknowledges the Tag by sending an *ACK*. If the Tag receives the *ACK* with a correct RN16 it backscatters the reply shown in Table 6.14 and transitions to the **acknowledged** state. If the Tag receives the *ACK* with an incorrect RN16 it transitions to **arbitrate**. Assuming a successful *ACK*, the Interrogator may either access the acknowledged Tag or issue a *QueryAdjust* or *QueryRep* with matching <u>session</u> parameter to invert the Tag's **inventoried** flag from *A*→*B* and send the Tag to **ready** (a *Query* with matching prior-round <u>session</u> parameter will also invert the **inventoried** flag from *A*→*B*).

c) **Multiple Tags reply:** The Interrogator observes a backscattered waveform comprising multiple RN16s. It may try to resolve the collision and issue an *ACK*; not resolve the collision and issue a *QueryAdjust, QueryRep*, or *NAK*; or quickly identify the collision and issue a *QueryAdjust* or *QueryRep* before the collided Tags have finished backscattering. In the latter case the collided Tags, not observing a valid reply within T_2 (see Figure 6.16), return to **arbitrate** and await the next *Query* or *QueryAdjust* command.



Symbol	Description
P	Preamble (R=>T or T=>R)
FS	Frame-Sync
RN16	16-bit Random Number
PC	Either StoredPC or PacketPC
XPC	Optional XPC word or words
CRC-16	Either StoredCRC or PacketCRC

Figure 6.21 – One Tag reply

Table 6.17 – Access commands and Tag states in which they are permitted

Command	State			Remark
	Acknowledged	Open	Secured	
Req_RN	permitted	permitted	permitted	–
Read	–	permitted	permitted	–
Write	–	permitted	permitted	requires prior *Req_RN*
Kill	–	permitted	permitted	requires prior *Req_RN*
Lock	–	–	permitted	–
Access	–	permitted	permitted	optional command; requires prior *Req_RN*
BlockWrite	–	permitted	permitted	optional command
BlockErase	–	permitted	permitted	optional command
BlockPermalock	–	–	permitted	optional command

6.3.2.9 Accessing individual Tags

After acknowledging a Tag, an Interrogator may choose to access it. The access command set comprises *Req_RN, Read, Write*, *Kill*, *Lock*, *Access*, *BlockWrite*, *BlockErase*, and *BlockPermalock*. Tags execute access commands in the states shown in Table 6.17.

An Interrogator accesses a Tag in the **acknowledged** state as follows:

Step 1. The Interrogator issues a *Req_RN* to the acknowledged Tag.

Step 2. The Tag generates and stores a new RN16 (denoted <u>handle</u>), backscatters the <u>handle</u>, and transitions to the **open** state if its access password is nonzero, or to the **secured** state if its access password is zero. The Interrogator may now issue further access commands.

All access commands issued to a Tag in the **open** or **secured** states include the Tag's <u>handle</u> as a parameter in the command. When in either of these two states, Tags verify that the <u>handle</u> is correct prior to executing an access command, and ignore access commands with an incorrect <u>handle</u>. The <u>handle</u> value is fixed for the entire duration of a Tag access.

Tags in the **open** state can execute all access commands except *Lock* and *BlockPermalock*. Tags in the **secured** state can execute all access commands. A Tag's response to an access command includes, at a minimum, the Tag's <u>handle</u>; the response may include other information as well (for example, the result of a *Read* operation).

An Interrogator may issue an *ACK* to a Tag in the **open** or **secured** states, causing the Tag to backscatter the reply shown in Table 6.14.

Interrogator and Tag can communicate indefinitely in the **open** or **secured** states. The Interrogator may terminate the communications at any time by issuing a *Query*, *QueryAdjust*, *QueryRep*, or a *NAK*. The Tag's response to a *Query*, *QueryAdjust*, or *QueryRep* is described in 6.3.2.8. A *NAK* causes all Tags in the inventory round to return to **arbitrate** without changing their **inventoried** flag.

The *Write*, *Kill*, and *Access* commands send 16-bit words (either data or half-passwords) from Interrogator to Tag. These commands use one-time-pad based link cover-coding to obscure the word being transmitted, as follows:

Step 1. The Interrogator issues a *Req_RN,* to which the Tag responds by backscattering a new RN16. The Interrogator then generates a 16-bit ciphertext string comprising a bit-wise EXOR of the 16-bit word to be transmitted with this new RN16, both MSB first, and issues the command with this ciphertext string as a parameter.

Step 2. The Tag decrypts the received ciphertext string by performing a bit-wise EXOR of the received 16-bit ciphertext string with the original RN16.

If an Interrogator issues a command containing cover-coded data or a half-password and fails to receive a response from the Tag, the Interrogator may reissue the command unchanged. If the Interrogator issues a command with new data or new half-password, then the Interrogator shall first issue a *Req_RN* to obtain a new RN16 and shall use this new RN16 for the cover-coding.

To reduce security risks, this specification recommends that (1) Tags use unique kill passwords, and (2) memory writes be performed in a secure location.

The *BlockWrite* command (see 6.3.2.11.3.7) communicates multiple 16-bit words from Interrogator to Tag. Unlike *Write*, *BlockWrite* does not use link cover-coding.

A Tag responds to a command that writes to or erases memory (i.e. *Write*, *Kill*, *Lock*, *BlockWrite*, *BlockErase*, and *BlockPermalock* with Read/Lock=1 (see 6.3.2.11.3.9)) by backscattering its handle, indicating that the operation was successful, or by backscattering an error code (see Annex I), indicating that the operation was unsuccessful. The Tag reply uses the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. the Tag replies as if TRext=1 regardless of the TRext value specified in the *Query* command that initiated the inventory round). See 6.3.2.11.3 for detailed descriptions of a Tag's reply to each particular access command.

Issuing an access password to a Tag is a multi-step procedure described in 6.3.2.11.3.6 and outlined in Figure 6.25.

Tag memory may be unlocked or locked. The lock status may be changeable or permalocked (i.e. permanently unlocked or permanently locked). Recommissioning the Tag may change the lock status, even if the memory was previously permalocked. An Interrogator may write to unlocked memory from either the **open** or **secured** states. An Interrogator may write to locked memory that is not permalocked from the **secured** state only. See 6.3.2.10, 6.3.2.11.3.5, 6.3.2.11.3.9, Table 6.43, and Table 6.50 for a detailed description of memory lock, permalock, recommissioning, and the Tag state required to modify memory.

This protocol recommends that Interrogators avoid powering-off while a Tag is in the **reply**, **acknowledged**, **open** or **secured** states. Rather, Interrogators should end their dialog with a Tag before powering off, leaving the Tag in either the **ready** or **arbitrate** state.

6.3.2.10 Killing or recommissioning a Tag

Killing or recommissioning a Tag is a multi-step procedure, described in 6.3.2.11.3.4 and shown in Figure 6.23, in which an Interrogator sends two successive *Kill* commands to a Tag. The first *Kill* command contains the first half of the kill password, and the second *Kill* command contains the second half. Each *Kill* command also contains 3 RFU/Recom bits. In the first *Kill* command these bits are RFU and zero valued; in the second *Kill* command they are called *recommissioning* (or Recom) bits and may be nonzero valued. The procedures for killing or recommissioning a Tag are identical, except that the recommissioning bits in the second *Kill* command are zero when killing a Tag dead and are nonzero when recommissioning it. Regardless of the intended operation, a Tag shall not kill or recommission itself without first receiving the correct kill password by the procedure shown in Figure 6.23.

If a Tag does not implement recommissioning then it shall ignore the recommissioning bits and treat them as though they were zero, meaning that, if the Tag receives a properly formatted *Kill* command sequence with the correct kill password it shall kill itself dead regardless of the values of the recommissioning bits. The remainder of this section 6.3.2.10 assumes that a Tag implements recommissioning.

Upon receiving a properly formatted *Kill* command sequence with the correct kill password and one or more nonzero recommissioning bits, a Tag shall assert those LSBs of its XPC_W1 that are asserted in the recommissioning bits (for example, if a Tag receives 100_2 for the recommissioning bits then it asserts the 3SB of its XPC_W1). The XPC_W1 LSBs shall be one-time-writeable, meaning that they cannot be deasserted after they are asserted. By storing the Tag's recommissioned status in the XPC_W1 a subsequent reader can know how the Tag was recommissioned (see Table 6.15). A Tag shall perform the following operations based on the recommissioning bit values it receives (see also 6.3.2.1.2.5):

- **Asserted LSB:** The Tag shall disable block permalocking and unlock any blocks of User memory that were previously permalocked. The Tag shall disable support for the *BlockPermalock* command. If the Tag did not implement block permalocking prior to recommissioning then block permalocking shall remain disabled. The lock status of User memory shall be determined solely by the lock bits (see 6.3.2.11.3.5).
- **Asserted 2SB:** The Tag shall render its User memory inaccessible, causing the entire memory bank to become unreadable, unwriteable, and unselectable (i.e. the Tag functions as though its User memory bank no longer exists).The 2SB has precedence over the LSB — if both are asserted then User memory is inaccessible.
- **Asserted 3SB:** The Tag shall unlock its EPC, TID, and User memory banks, regardless of whether these banks were locked or permalocked. Portions of User memory that were block permalocked shall remain block permalocked, and vice versa, unless the LSB is also asserted, in which case the Tag shall unlock its permalocked blocks. The Tag shall write-unlock its kill and access passwords, and shall render the kill

and access passwords permanently unreadable regardless of the values of the Tag's lock bits (see 6.3.2.11.3.5). If an Interrogator subsequently attempts to read the Tag's kill or access passwords the Tag shall backscatter an error code (see Annex I). A Tag that receives a subsequent *Lock* command with pwd-read/write=0 shall lock or permalock the indicated password(s) in the writeable state, but the passwords shall still remain unreadable.

A Tag shall not execute any of the above recommissioning operations more than once. As one example, a Tag does not allow any of its memory banks to be unlocked more than once by recommissioning.

A Tag may execute multiple *Kill* command sequences, depending on the nature and ordering of the operations specified in these command sequences. Specifically:

- A Tag that is killed dead shall not allow a subsequent recommissioning.
- A previously recommissioned Tag that receives a properly formatted *Kill* command sequence with the correct kill password and Recom = 000_2 shall be killed dead.
- A Tag that receives a properly formatted *Kill* command sequence with the correct kill password but with redundant recommissioning bits (for example, the recommissioning bits are 100_2 but the Tag's XPC_W1 already contains 100_2) shall not perform the requested recommissioning operation again. Instead, the Tag shall merely verify that its XPC_W1 contains the asserted values and respond affirmatively to the Interrogator. An Interrogator may choose to send a *Kill* sequence with redundant recommissioning bits if, for example, it had sent a prior *Kill* sequence but did not observe an affirmative response from the Tag.
- A Tag that receives a properly formatted *Kill* command sequence with the correct kill password and with newly asserted recommissioning bits shall perform the recommissioning operation indicated by the newly asserted bits, responding affirmatively to the reader when done. A Tag shall compute the logical OR of the LSBs of its current XPC_W1 and the recommissioning bits, and shall store the resulting value into its XPC_W1. For example, if a Tag whose XPC_W1 LSBs are 100_2 receives a *Kill* command sequence whose recommissioning bits are 010_2, the Tag renders its User memory bank inaccessible and stores 110_2 into its XPC_W1 LSBs.

An Interrogator may subsequently re-lock any of the memory banks or passwords that have been unlocked by recommissioning.

Portions or entire banks of Tag memory, if factory locked, may not be unlockable by recommissioning. An Interrogator may determine whether a Tag supports recommissioning, and if so which (if any) memory portions are not recommissionable, by reading the Tag's TID memory prior to recommissioning.

A Tag that does not implement a kill password, or a Tag whose kill password is zero, shall not execute a kill or recommissioning operation. Such a Tag shall respond with an error code (as shown in Figure 6.23) to a *Kill* command sequence regardless of the RFU or recommissioning bit settings.

A Tag shall accept all eight possible combinations of the 3 recommissioning bits, executing those portions that it is capable of executing, ignoring those it cannot, and responding affirmatively to the Interrogator when done. Several examples of operations that a Tag may be incapable or partially capable of executing are:

- A Tag that does not have User memory cannot unlock it
- A Tag that does not implement an Access password cannot unlock it for writing
- A Tag in which a portion of TID memory is factory locked cannot unlock this portion

6.3.2.11 Interrogator commands and Tag replies

Interrogator-to-Tag commands shall have the format shown in Table 6.18.

- *QueryRep* and *ACK* have 2-bit command codes beginning with 0_2.
- *Query*, *QueryAdjust*, and *Select* have 4-bit command codes beginning with 10_2.
- All other base commands have 8-bit command codes beginning with 110_2.
- All extended commands have 16-bit command codes beginning with 1110_2.
- *QueryRep*, *ACK*, *Query*, *QueryAdjust,* and *NAK* have the unique command lengths shown in Table 6.18. No other commands shall have these lengths. If a Tag receives one of these commands with an incorrect length it shall ignore the command.
- *Query, QueryAdjust*, and *QueryRep* contain a session parameter.
- *Query* is protected by a CRC-5, shown in Table 6.12 and detailed in Annex F.

- *Select*, *Req_RN*, *Read*, *Write*, *Kill*, *Lock*, *Access*, *BlockWrite*, *BlockErase*, and *BlockPermalock* are protected by a CRC-16, defined in 6.3.1.5 and detailed in Annex F.
- R=>T commands begin with either a preamble or a frame-sync, as described in 6.3.1.2.8. The command-code lengths specified in Table 6.18 do not include the preamble or frame-sync.
- Tags shall ignore invalid commands. In general, "invalid" means a command that (1) is incorrect given the current Tag state, (2) is unsupported by the Tag, (3) has incorrect parameters, (4) has a CRC error, (5) specifies an incorrect session, or (6) is in any other way not recognized or not executable by the Tag. The actual definition of "invalid" is state-specific and defined, for each Tag state, in Annex B and Annex C.

Table 6.18 – Commands

Command	Code	Length (bits)	Mandatory?	Protection
QueryRep	**0**0	4	Yes	Unique command length
ACK	**0**1	18	Yes	Unique command length
Query	**10**00	22	Yes	Unique command length and a CRC-5
QueryAdjust	**10**01	9	Yes	Unique command length
Select	**10**10	> 44	Yes	CRC-16
Reserved for future use	**10**11	–	–	–
NAK	**11**000000	8	Yes	Unique command length
Req_RN	**11**000001	40	Yes	CRC-16
Read	**11**000010	> 57	Yes	CRC-16
Write	**11**000011	> 58	Yes	CRC-16
Kill	**11**000100	59	Yes	CRC-16
Lock	**11**000101	60	Yes	CRC-16
Access	**11**000110	56	No	CRC-16
BlockWrite	**11**000111	> 57	No	CRC-16
BlockErase	**11**001000	> 57	No	CRC-16
BlockPermalock	**11**001001	> 66	No	CRC-16
Reserved for future use	**11**001010 … **11**011111	–	–	–
Reserved for custom commands	**1110**0000 00000000 … **1110**0000 11111111	–	–	Manufacturer specified
Reserved for proprietary commands	**1110**0001 00000000 … **1110**0001 11111111	–	–	Manufacturer specified
Reserved for future use	**1110**0010 00000000 … **1110**1111 11111111	–	–	–

6.3.2.11.1 Select commands

The Select command set comprises a single command: *Select*.

6.3.2.11.1.1 ***Select*** (mandatory)

Select selects a particular Tag population based on user-defined criteria, enabling union (U), intersection (∩), and negation (~) based Tag partitioning. Interrogators perform U and ∩ operations by issuing successive *Select* commands. *Select* can assert or deassert a Tag's **SL** flag, which applies across all four sessions, or it can set a Tag's **inventoried** flag to either *A* or *B* in any one of the four sessions.

Interrogators and Tags shall implement the *Select* command shown in Table 6.19. Target shall indicate whether the *Select* modifies a Tag's **SL** or **inventoried** flag, and in the case of the **inventoried** flag, for which session. Action shall elicit the Tag response shown in Table 6.20. The criteria for determining whether a Tag is matching or non-matching are specified in the MemBank, Pointer, Length and Mask fields. Truncate indicates whether a Tag's backscattered reply shall be truncated to include only those EPC and StoredCRC bits that follow Mask. *Select* passes the following parameters from Interrogator to Tags:

- Target indicates whether the *Select* command modifies a Tag's **SL** flag or its **inventoried** flag, and in the case of **inventoried** it further specifies one of four sessions. A *Select* command that modifies **SL** does not modify **inventoried**, and vice versa. Class-1 Tags shall ignore *Select* commands whose Target is 101_2, 110_2, or 111_2.
- Action indicates whether matching Tags assert or deassert **SL**, or set their **inventoried** flag to *A* or to *B*. Tags conforming to the contents of the MemBank, Pointer, Length, and Mask fields are considered matching. Tags not conforming to the contents of these fields are considered non-matching.
- MemBank specifies whether Mask applies to EPC, TID, or User memory. *Select* commands shall apply to a single memory bank. Successive *Selects* may apply to different banks. MemBank shall not specify Reserved memory; if a Tag receives a *Select* specifying MemBank = 00_2 it shall ignore the *Select*. MemBank parameter value 00_2 is reserved for future use (RFU).
- Pointer, Length, and Mask: Pointer and Length describe a memory range. Pointer references a memory bit address (Pointer is not restricted to word boundaries) and uses EBV formatting (see Annex A). Length is 8 bits, allowing Masks from 0 to 255 bits in length. Mask, which is Length bits long, contains a bit string that a Tag compares against the memory location that begins at Pointer and ends Length bits later. If Pointer and Length reference a memory location that does not exist on the Tag then the Tag shall consider the *Select* to be non-matching. If Length is zero then all Tags shall be considered matching, unless Pointer references a memory location that does not exist on the Tag or Truncate=1 and Pointer is outside the EPC specified in the StoredPC, in which case the Tag shall consider the *Select* to be non-matching.
- Truncate: If an Interrogator asserts Truncate, and if a subsequent *Query* specifies Sel=10 or Sel=11, then a matching Tag shall truncate its reply to an *ACK* to that portion of the EPC immediately following Mask, followed by the StoredCRC. If an Interrogator asserts Truncate, it shall assert it:
  - in the last *Select* that the Interrogator issues prior to sending a *Query*,
  - only if the *Select* has Target = 100_2, and
  - only if Mask ends in the EPC.

  These constraints *do not* preclude an Interrogator from issuing multiple *Select* commands that target the **SL** and/or **inventoried** flags. They *do* require that an Interrogator that is requesting Tags to truncate their replies assert Truncate in the last *Select*, and that this last *Select* targets the **SL** flag. Tags shall power-up with Truncate deasserted.

  Tags shall decide whether to truncate their backscattered EPC on the basis of the most recently received *Select*. If a Tag receives a *Select* with Truncate=1 but Target<>100_2 the Tag shall ignore the *Select*. If a Tag receives a *Select* in which Truncate=1 but MemBank<>01, the Tag shall consider the *Select* to be invalid. If a Tag receives a *Select* in which Truncate=1, MemBank=01, but Mask ends outside the EPC specified in the StoredPC, the Tag shall consider the *Select* to be non-matching.

  A Tag shall preface a truncated reply with five leading zeros (00000_2) inserted between the preamble and the truncated reply. Specifically, when truncating its replies a Tag backscatters 00000_2, then the portion of its EPC following Mask, and then its StoredCRC. See Table 6.14.

  A Tag does not recalculate its StoredCRC for a truncated reply.

Mask may end at the last bit of the EPC, in which case a truncating Tag shall backscatter 00000_2 followed by its StoredCRC.

Truncated replies never include an XPC_W1 or an XPC_W2, because Mask must end in the EPC.

A recommissioned Tag shall not truncate its replies. A recommissioned Tag that receives a *Select* with Truncate=1 shall evaluate the *Select* normally, but when replying to a subsequent *ACK* it shall backscatter its PacketPC, XPC_W1, optionally its XPC_W2 (if XEB is asserted), an EPC whose length is as specified in the EPC length field in the StoredPC, and its PacketCRC.

Interrogators can use a *Select* command to reset all Tags in a session to **inventoried** state *A*, by issuing a *Select* with Action = 000_2 and a Length value of zero.

Because a Tag stores its StoredPC and StoredCRC in EPC memory, a *Select* command may select on them. Because a Tag computes its PacketPC and PacketCRC dynamically and does not store them in memory, a *Select* command is unable to select on them.

Because a Tag may compute its PC and/or CRC dynamically, its response to a *Select* command whose Pointer, Length, and Mask include the StoredPC or StoredCRC may produce unexpected behavior. Specifically, a Tag's backscattered reply may appear to not match Mask even though the Tag's behavior indicates matching, and vice versa. For example, suppose an Interrogator sends a *Select* to match a 00100_2 EPC length field in the StoredPC. Further assume that a Tag matches, but has an asserted XI. The Tag will dynamically increment its EPC length field to 00101_2 when responding to an *ACK*, and will backscatter this incremented value in the PacketPC. The Tag was matching, but the backscattered EPC length field appears to be non-matching.

Interrogators shall prepend a *Select* with a frame-sync (see 6.3.1.2.8). The CRC-16 that protects a *Select* is calculated over the first command-code bit to the Truncate bit.

Tags shall not reply to a *Select*.

Table 6.19 – *Select* command

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 6.20	00: RFU 01: EPC 10: TID 11: User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	

Table 6.20 – Tag response to Action parameter

Action	Matching	Non-Matching
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*
001	assert **SL** or **inventoried** → *A*	do nothing
010	do nothing	deassert **SL** or **inventoried** → *B*
011	negate **SL** or (*A* → *B*, *B* → *A*)	do nothing
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*
101	deassert **SL** or **inventoried** → *B*	do nothing
110	do nothing	assert **SL** or **inventoried** → *A*
111	do nothing	negate **SL** or (*A* → *B*, *B* → *A*)

6.3.2.11.2 Inventory commands

The inventory command set comprises *Query*, *QueryAdjust, QueryRep, ACK*, and *NAK*.

6.3.2.11.2.1 *Query* (mandatory)

Interrogators and Tags shall implement the *Query* command shown in Table 6.21. *Query* initiates and specifies an inventory round. *Query* includes the following fields:

- DR (TRcal divide ratio) sets the T=>R link frequency as described in 6.3.1.2.8 and Table 6.9.
- M (cycles per symbol) sets the T=>R data rate and modulation format as shown in Table 6.10.
- TRext chooses whether the T=>R preamble is prepended with a pilot tone as described in 6.3.1.3.2.2 and 6.3.1.3.2.4; however, a Tag's reply to access commands that write to memory (i.e. *Write*, *Kill*, *Lock*, *BlockWrite*, *BlockErase*, and *BlockPermalock*) always uses a pilot tone regardless of the value of TRext.
- Sel chooses which Tags respond to the *Query* (see 6.3.2.11.1.1 and 6.3.2.8).
- Session chooses a session for the inventory round (see 6.3.2.8).
- Target selects whether Tags whose **inventoried** flag is *A* or *B* participate in the inventory round. Tags may change their inventoried flag from *A* to *B* (or vice versa) as a result of being singulated.
- Q sets the number of slots in the round (see 6.3.2.8).

Interrogators shall prepend a *Query* with a preamble (see 6.3.1.2.8).

The CRC-5 that protects a *Query* is calculated over the first command-code bit to the last Q bit. If a Tag receives a *Query* with a CRC-5 error it shall ignore the command.

Upon receiving a *Query*, Tags with matching Sel and Target shall pick a random value in the range (0, $2^Q - 1$), inclusive, and shall load this value into their slot counter. If a Tag, in response to the *Query*, loads its slot counter with zero, then its reply to a *Query* shall be as shown in Table 6.22; otherwise the Tag shall remain silent.

A *Query* may initiate an inventory round in a new session, or in the prior session. If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose session parameter matches the prior session it shall invert its **inventoried** flag (i.e. *A*→*B* or *B*→*A*) for the session before it evaluates whether to transition to **ready**, **arbitrate**, or **reply**. If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose session parameter does not match the prior session it shall leave its **inventoried** flag for the prior session unchanged when beginning the new round.

Tags shall support all DR and M values specified in Table 6.9 and Table 6.10, respectively.

Tags in any state other than **killed** shall execute a *Query* command, starting a new round in the specified session and transitioning to **ready**, **arbitrate**, or **reply**, as appropriate (see Figure 6.19). Tags in the **killed** state shall ignore a *Query*.

Table 6.21 – *Query* command

	Command	DR	M	TRext	Sel	Session	Target	Q	CRC-5
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~**SL** 11: **SL**	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	

Table 6.22 – Tag reply to a *Query* command

	Response
# of bits	16
description	RN16

6.3.2.11.2.2 ***QueryAdjust*** (mandatory)

Interrogators and Tags shall implement the *QueryAdjust* command shown in Table 6.23. *QueryAdjust* adjusts *Q* (i.e. the number of slots in an inventory round – see 6.3.2.8) without changing any other round parameters.

QueryAdjust includes the following fields:

- Session corroborates the session number for the inventory round (see 6.3.2.8 and 6.3.2.11.2.1). If a Tag receives a *QueryAdjust* whose session number is different from the session number in the *Query* that initiated the round it shall ignore the command.
- UpDn determines whether and how the Tag adjusts *Q*, as follows:

  110: Increment *Q* (i.e. $Q = Q + 1$).

  000: No change to *Q*.

  011: Decrement *Q* (i.e. $Q = Q - 1$).

  If a Tag receives a *QueryAdjust* with an UpDn value different from those specified above it shall ignore the command. If a Tag whose *Q* value is 15 receives a *QueryAdjust* with UpDn = 110 it shall change UpDn to 000 prior to executing the command; likewise, if a Tag whose *Q* value is 0 receives a *QueryAdjust* with UpDn = 011 it shall change UpDn to 000 prior to executing the command.

Tags shall maintain a running count of the current *Q* value. The initial *Q* value is specified in the *Query* command that started the inventory round; one or more subsequent *QueryAdjust* commands may modify *Q*.

A *QueryAdjust* shall be prepended with a frame-sync (see 6.3.1.2.8).

Upon receiving a *QueryAdjust* Tags first update *Q*, then pick a random value in the range (0, 2^Q-1), inclusive, and load this value into their slot counter. If a Tag, in response to the *QueryAdjust*, loads its slot counter with zero, then its reply to a *QueryAdjust* shall be shown in Table 6.24; otherwise, the Tag shall remain silent. Tags shall respond to a *QueryAdjust* only if they received a prior *Query*.

Tags in any state except **ready** or **killed** shall execute a *QueryAdjust* command if, and only if, (i) the session parameter in the command matches the session parameter in the *Query* that started the round, and (ii) the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively).

Tags in the **acknowledged**, **open**, or **secured** states that receive a *QueryAdjust* whose session parameter matches the session parameter in the prior *Query*, and who are not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively), shall invert their **inventoried** flag (i.e. *A*→*B* or *B*→*A*, as appropriate) for the current session and transition to **ready**.

Table 6.23 – *QueryAdjust* command

	Command	Session	UpDn
# of bits	4	2	3
description	1001	00: S0 01: S1 10: S2 11: S3	110: $Q = Q + 1$ 000: No change to *Q* 011: $Q = Q - 1$

Table 6.24 – Tag reply to a *QueryAdjust* command

	Response
# of bits	16
description	RN16

6.3.2.11.2.3 ***QueryRep*** **(mandatory)**

Interrogators and Tags shall implement the *QueryRep* command shown in Table 6.25. *QueryRep* instructs Tags to decrement their slot counters and, if slot = 0 after decrementing, to backscatter an RN16 to the Interrogator.

QueryRep includes the following field:

- Session corroborates the session number for the inventory round (see 6.3.2.8 and 6.3.2.11.2.1). If a Tag receives a *QueryRep* whose session number is different from the session number in the *Query* that initiated the round it shall ignore the command.

A *QueryRep* shall be prepended with a frame-sync (see 6.3.1.2.8).

If a Tag, in response to the *QueryRep*, decrements its slot counter and the decremented slot value is zero, then its reply to a *QueryRep* shall be as shown in Table 6.26; otherwise the Tag shall remain silent. Tags shall respond to a *QueryRep* only if they received a prior *Query*.

Tags in any state except **ready** or **killed** shall execute a *QueryRep* command if, and only if, (i) the session parameter in the command matches the session parameter in the *Query* that started the round, and (ii) the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively).

Tags in the **acknowledged**, **open**, or **secured** states that receive a *QueryRep* whose session parameter matches the session parameter in the prior *Query*, and who are not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.11.3.4 or 6.3.2.11.3.6, respectively), shall invert their **inventoried** flag (i.e. *A*→*B* or *B*→*A*, as appropriate) for the current session and transition to **ready**.

Table 6.25 – *QueryRep* command

	Command	Session
# of bits	2	2
description	00	00: S0 01: S1 10: S2 11: S3

Table 6.26 – Tag reply to a *QueryRep* command

	Response
# of bits	16
description	RN16

6.3.2.11.2.4 *ACK* (mandatory)

Interrogators and Tags shall implement the *ACK* command shown in Table 6.27. An Interrogator sends an *ACK* to acknowledge a single Tag. *ACK* echoes the Tag's backscattered RN16.

If an Interrogator issues an *ACK* to a Tag in the **reply** or **acknowledged** states, then the echoed RN16 shall be the RN16 that the Tag previously backscattered as it transitioned from the **arbitrate** state to the **reply** state. If an Interrogator issues an *ACK* to a Tag in the **open** or **secured** states, then the echoed RN16 shall be the Tag's handle (see 6.3.2.11.3.1).

An *ACK* shall be prepended with a frame-sync (see 6.3.1.2.8).

The Tag reply to a successful *ACK* shall be as shown in Table 6.28. As described in 6.3.2.11.1.1, the reply may be truncated, in which case the Tag reply is 00000_2 followed by the truncated EPC followed by the StoredCRC. A Tag that receives an *ACK* with an incorrect RN16 or an incorrect handle (as appropriate) shall return to **arbitrate** without responding, unless the Tag is in **ready** or **killed**, in which case it shall ignore the *ACK* and remain in its present state.

If a Tag does not support XPC functionality then the maximum length of its backscattered EPC is 496 bits. If a Tag supports XPC functionality then the maximum length of its backscattered EPC is reduced by two words to accommodate the XPC_W1 and optional XPC_W2, so is 464 bits. See 6.3.2.1.2.2. In either case a Tag's reply to an *ACK* shall not exceed 528 bits in length.

Table 6.27 – *ACK* command

	Command	RN
# of bits	2	16
description	01	Echoed RN16 or handle

Table 6.28 – Tag reply to a successful *ACK* command

	Response
# of bits	21 to 528
description	See Table 6.14

6.3.2.11.2.5 *NAK* (mandatory)

Interrogators and Tags shall implement the *NAK* command shown in Table 6.29. Any Tag that receives a *NAK* shall return to the **arbitrate** state without changing its **inventoried** flag, unless the Tag is in **ready** or **killed**, in which case it shall ignore the *NAK* and remain in its current state.

A *NAK* shall be prepended with a frame-sync (see 6.3.1.2.8).

Tags shall not reply to a *NAK*.

Table 6.29 – *NAK* command

	Command
# of bits	8
description	11000000

6.3.2.11.3 Access commands

The set of access commands comprises *Req_RN, Read, Write*, *Kill*, *Lock*, *Access*, *BlockWrite*, *BlockErase*, and *BlockPermalock*. As described in 6.3.2.9, Tags execute *Req_RN* from the **acknowledged**, **open**, or **secured** states. Tags execute *Read*, *Write*, *BlockWrite*, and *BlockErase* from the **secured** state; if allowed by the lock status of the memory location being accessed, they may also execute these commands from the **open** state. Tags execute *Access* and *Kill* from the **open** or **secured** states. Tags execute *Lock* and *BlockPermalock* only from the **secured** state.

All access commands issued to a Tag in the **open** or **secured** states include the Tag's handle as a parameter in the command. When in either of these two states, Tags shall verify that the handle is correct prior to executing an access command, and shall ignore access commands with an incorrect handle. The handle value is fixed for the entire duration of a Tag access.

A Tag's reply to all access commands that read or write memory (i.e. *Read, Write*, *Kill*, *Lock*, *BlockWrite*, *BlockErase*, and *BlockPermalock*) includes a 1-bit header. Header=0 indicates that the operation was successful and the reply is valid; header=1 indicates that the operation was unsuccessful and the reply is an error code.

A Tag's reply to all access commands that write to memory (i.e. *Write*, *Kill*, *Lock*, *BlockWrite*, *BlockErase*, and *BlockPermalock* with Read/Lock=1 (see 6.3.2.11.3.9) shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. the Tag shall reply as if TRext=1 regardless of the TRext value specified in the *Query* command that initiated the inventory round).

After an Interrogator writes to a Tag's StoredPC or EPC, or changes bits 03_h to 07_h of User memory from zero to a nonzero value (or vice versa), or recommissions the Tag, the StoredCRC may be incorrect until the Interrogator powers-down and then re-powers-up its energizing RF field, because a Tag calculates its StoredCRC at power-up.

If an Interrogator attempts to write to EPC memory locations 00_h to $0F_h$ (i.e. to the StoredCRC) using a *Write*, *BlockWrite*, or *BlockErase* command the Tag shall ignore the command and respond with an error code (see Annex I for error-code definitions and for the reply format).

If an Interrogator attempts to write to EPC memory locations 210_h to $22F_h$ (i.e. to the XPC_W1 or XPC_W2) of a Class-1 Tag using a *Write*, *BlockWrite*, or *BlockErase* command the Tag shall ignore the command and respond with an error code (see Annex I for error-code definitions and for the reply format).

If an Interrogator attempts to write to a memory bank or password that is permalocked unwriteable, or to a memory bank or password that is locked unwriteable and the Tag is not in the **secured** state, or to a memory block that is permalocked, using a *Write*, *BlockWrite*, or *BlockErase* command, or if a portion of the Data in a *Write* command overlaps a permalocked memory block, the Tag shall ignore the command and respond with an error code (see Annex I for error-code definitions and for the reply format).

Req_RN, Read, Write, *Kill*, and *Lock* are required commands; *Access*, *BlockWrite*, *BlockErase*, and *BlockPermalock* are optional. Tags shall ignore optional access commands that they do not support.

See Annex K for an example of a data-flow exchange during which an Interrogator accesses a Tag and reads its kill password.

6.3.2.11.3.1 *Req_RN* (mandatory)

Interrogators and Tags shall implement the *Req_RN* command shown in Table 6.30. *Req_RN* instructs a Tag to backscatter a new RN16. Both the Interrogator's command, and the Tag's response, depend on the Tag's state:

- **Acknowledged** state: When issuing a *Req_RN* command to a Tag in the **acknowledged** state, an Interrogator shall include the Tag's last backscattered RN16 as a parameter in the *Req_RN*. The *Req_RN* command is protected by a CRC-16 calculated over the command bits and the RN16. If the Tag receives the *Req_RN* with a valid CRC-16 and a valid RN16 it shall generate and store a new RN16 (denoted handle), backscatter this handle, and transition to the **open** or **secured** state. The choice of ending state depends on the Tag's access password, as follows:
  - Access password <> 0: Tag transitions to **open** state.
  - Access password = 0: Tag transitions to **secured** state.

  If the Tag receives the *Req_RN* command with a valid CRC-16 but an invalid RN16 it shall ignore the *Req_RN* and remain in the **acknowledged** state.
- **Open** or **secured** states: When issuing a *Req_RN* command to a Tag in the **open** or **secured** states, an Interrogator shall include the Tag's handle as a parameter in the *Req_RN*. The *Req_RN* command is protected by a CRC-16 calculated over the command bits and the handle. If the Tag receives the *Req_RN* with a valid CRC-16 and a valid handle it shall generate and backscatter a new RN16. If the Tag receives the *Req_RN* with a valid CRC-16 but an invalid handle it shall ignore the *Req_RN*. In either case the Tag shall remain in its current state (**open** or **secured**, as appropriate).

If an Interrogator wishes to ensure that only a single Tag is in the **acknowledged** state it may issue a *Req_RN*, causing the Tag or Tags to each backscatter a handle and transition to the **open** or **secured** state (as appropriate). The Interrogator may then issue an *ACK* with handle as a parameter. Tags that receive an *ACK* with an invalid handle shall return to **arbitrate** (Note: If a Tag receives an *ACK* with an invalid handle it returns to **arbitrate**, whereas if it receives an access command with an invalid handle it ignores the command).

The first bit of the backscattered RN16 shall be denoted the MSB; the last bit shall be denoted the LSB.

A *Req_RN* shall be prepended with a frame-sync (see 6.3.1.2.8).

The Tag reply to a *Req_RN* shall be as shown in Table 6.31. The RN16 or handle are protected by a CRC-16.

Table 6.30 – *Req_RN* command

	Command	RN	CRC-16
# of bits	8	16	16
description	11000001	Prior RN16 or handle	

Table 6.31 – Tag reply to a *Req_RN* command

	RN	CRC-16
# of bits	16	16
description	handle or new RN16	

6.3.2.11.3.2 *Read* (mandatory)

Interrogators and Tags shall implement the *Read* command shown in Table 6.33. *Read* allows an Interrogator to read part or all of a Tag's Reserved, EPC, TID, or User memory. *Read* has the following fields:

- <u>MemBank</u> specifies whether the *Read* accesses Reserved, EPC, TID, or User memory. *Read* commands shall apply to a single memory bank. Successive *Reads* may apply to different banks.
- <u>WordPtr</u> specifies the starting word address for the memory read, where words are 16 bits in length. For example, <u>WordPtr</u> = 00_h specifies the first 16-bit memory word, <u>WordPtr</u> = 01_h specifies the second 16-bit memory word, etc. <u>WordPtr</u> uses EBV formatting (see Annex A).
- <u>WordCount</u> specifies the number of 16-bit words to be read. If <u>WordCount</u> = 00_h the Tag shall backscatter the contents of the chosen memory bank starting at <u>WordPtr</u> and ending at the end of the bank, unless <u>MemBank</u> = 01_2, in which case the Tag shall backscatter the memory contents specified in Table 6.32.

Table 6.32 – Tag backscatter when <u>WordCount</u>=00_h and <u>MemBank</u>=01_2

<u>WordPtr</u> Memory Address	Tag Implements XPC_W1?	Tag Implements XPC_W2?	What the Tag Backscatters
Within the StoredCRC, StoredPC, or the EPC specified by bits 10_h–14_h of the StoredPC	Don't care	Don't care	EPC memory starting at <u>WordPtr</u> and ending at the EPC length specified by bits 10_h–14_h of the StoredPC
Within physical EPC memory but above the EPC specified by bits 10_h–14_h of the StoredPC	No	N/A. See note 1	EPC memory starting at <u>WordPtr</u> and ending at the end of physical EPC memory
Within physical EPC memory but above the EPC specified by bits 10_h–14_h of the StoredPC	Yes	No	EPC memory starting at <u>WordPtr</u> and ending at the end of physical EPC memory. Includes the XPC_W1 if <u>WordPtr</u> is less than or equal to 210_h and physical EPC memory extends to or above address 210_h
Within physical EPC memory but above the EPC specified by bits 10_h–14_h of the StoredPC	Yes	Yes	EPC memory starting at <u>WordPtr</u> and ending at the end of physical EPC memory. Includes the XPC_W1 and XPC_W2 if <u>WordPtr</u> is less than or equal to 210_h and physical EPC memory extends to or above address 210_h. Includes the XPC_W2 if <u>WordPtr</u> is equal to 220_h and physical EPC memory extends to or above 220_h.
210_h. Above physical EPC memory	No	N/A. See note 1	Error code
210_h. Above physical EPC memory	Yes	No	XPC_W1
210_h. Above physical EPC memory	Yes	Yes	XPC_W1 and XPC_W2
220_h. Above physical EPC memory	No	N/A. See note 1	Error code
220_h. Above physical EPC memory	Yes	No	Error code
220_h. Above physical EPC memory	Yes	Yes	XPC_W2
Not 210_h or 220_h. Above physical EPC memory.	Don't care	Don't care	Error code

Note 1: If a Tag does not implement an XPC_W1 then it does not implement an XPC_W2. See 6.3.2.1.2.5.

The *Read* command also includes the Tag's <u>handle</u> and a CRC-16. The CRC-16 is calculated over the first command-code bit to the last <u>handle</u> bit.

If a Tag receives a *Read* with a valid CRC-16 but an invalid <u>handle</u> it shall ignore the *Read* and remain in its current state (**open** or **secured**, as appropriate).

A *Read* shall be prepended with a frame-sync (see 6.3.1.2.8).

If all of the memory words specified in a *Read* exist and none are read-locked, the Tag reply to the *Read* shall be as shown in Table 6.34. The Tag responds by backscattering a header (a 0-bit), the requested memory words, and its <u>handle</u>. The reply includes a CRC-16 calculated over the 0-bit, memory words, and <u>handle</u>.

If a one or more of the memory words specified in the *Read* command either do not exist or are read-locked, the Tag shall backscatter an error code, within time T_1 in Table 6.13, rather than the reply shown in Table 6.34 (see Annex I for error-code definitions and for the reply format).

Table 6.33 – *Read* command

	Command	MemBank	WordPtr	WordCount	RN	CRC-16
# of bits	8	2	EBV	8	16	16
description	11000010	00: Reserved 01: EPC 10: TID 11: User	Starting address pointer	Number of words to read	handle	

Table 6.34 – Tag reply to a successful *Read* command

	Header	Memory Words	RN	CRC-16
# of bits	1	Variable	16	16
description	0	Data	handle	

6.3.2.11.3.3 ***Write*** (mandatory)

Interrogators and Tags shall implement the *Write* command shown in Table 6.35. *Write* allows an Interrogator to write a word in a Tag's Reserved, EPC, TID, or User memory. *Write* has the following fields:

- MemBank specifies whether the *Write* occurs in Reserved, EPC, TID, or User memory.
- WordPtr specifies the word address for the memory write, where words are 16 bits in length. For example, WordPtr = 00_h specifies the first 16-bit memory word, WordPtr = 01_h specifies the second 16-bit memory word, etc. WordPtr uses EBV formatting (see Annex A).
- Data contains a 16-bit word to be written. Before each and every *Write* the Interrogator shall first issue a *Req_RN* command; the Tag responds by backscattering a new RN16. The Interrogator shall cover-code the data by EXORing it with this new RN16 prior to transmission.

The *Write* command also includes the Tag's handle and a CRC-16. The CRC-16 is calculated over the first command-code bit to the last handle bit.

If a Tag in the **open** or **secured** states receives a *Write* with a valid CRC-16 but an invalid handle, or it receives a *Write* before which the immediately preceding command was not a *Req_RN*, it shall ignore the *Write* and remain in its current state.

A *Write* shall be prepended with a frame-sync (see 6.3.1.2.8).

After issuing a *Write* an Interrogator shall transmit CW for the lesser of T_{REPLY} or 20ms, where T_{REPLY} is the time between the Interrogator's *Write* command and the Tag's backscattered reply. An Interrogator may observe several possible outcomes from a *Write*, depending on the success or failure of the Tag's memory-write operation:

- **The *Write* succeeds:** After completing the *Write* a Tag shall backscatter the reply shown in Table 6.36 and Figure 6.22 comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle. If the Interrogator observes this reply within 20 ms then the *Write* completed successfully.
- **The Tag encounters an error:** The Tag shall backscatter an error code during the CW period rather than the reply shown in Table 6.36 (see Annex I for error-code definitions and for the reply format).
- **The *Write* does not succeed:** If the Interrogator does not observe a reply within 20ms then the *Write* did not complete successfully. The Interrogator may issue a *Req_RN* command (containing the Tag's handle) to verify that the Tag is still in the Interrogator's field, and may reissue the *Write* command.

Upon receiving a valid *Write* command a Tag shall write the commanded Data into memory. The Tag's reply to a *Write* shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. a Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the round).

Table 6.35 – *Write* command

	Command	MemBank	WordPtr	Data	RN	CRC-16
# of bits	8	2	EBV	16	16	16
description	11000011	00: Reserved 01: EPC 10: TID 11: User	Address pointer	RN16 ⊗ word to be written	handle	

Table 6.36 – Tag reply to a successful *Write* command

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

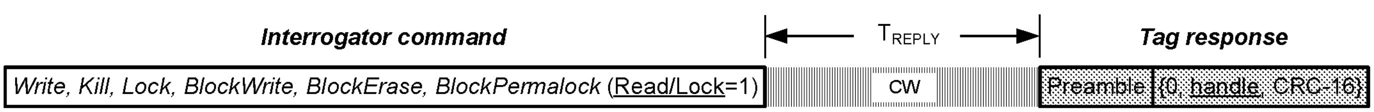


Figure 6.22 – Successful *Write* sequence

6.3.2.11.3.4 *Kill* (mandatory)

Interrogators and Tags shall implement the *Kill* command shown in Table 6.37 and Table 6.38. *Kill* allows an Interrogator to permanently disable a Tag. *Kill* also allows an Interrogator to recommission recommissionable Tags.

To kill or recommission a Tag, an Interrogator shall follow the multi-step procedure outlined in Figure 6.23. Briefly, the Interrogator issues two *Kill* commands, the first containing the 16 MSBs of the Tag's kill password EXORed with an RN16, and the second containing the 16 LSBs of the Tag's kill password EXORed with a different RN16. Each EXOR operation shall be performed MSB first (i.e. the MSB of each half-password shall be EXORed with the MSB of its respective RN16). Just prior to issuing each *Kill* command the Interrogator first issues a *Req_RN* to obtain a new RN16.

Tags shall incorporate the necessary logic to successively accept two 16-bit subportions of a 32-bit kill password. Interrogators shall not intersperse commands other than *Req_RN* between the two successive *Kill* commands. If a Tag, after receiving a first *Kill*, receives any valid command other than *Req_RN* before the second *Kill* it shall return to **arbitrate**, unless the intervening command is a *Query*, in which case the Tag shall execute the *Query* (inverting its **inventoried** flag if the session parameter in the *Query* matches the prior session).

Kill contains 3 RFU/Recom bits. In the first *Kill* command these bits are RFU. Interrogators shall set them to 000_2 when communicating with Class-1 Tags, and Class-1 Tags shall ignore them. Higher-class Tags may use these bits to expand the functionality of the *Kill* command. As described in 6.3.2.10, in the second *Kill* command these bits are called *recommissioning* (or Recom) bits and may be nonzero. The procedures for killing or recommissioning a Tag are identical, except that the recommissioning bits in the second *Kill* command are zero when killing a Tag and are nonzero when recommissioning it. Regardless of the intended operation, a Tag does not kill or recommission itself without first receiving the correct kill password by the procedure shown in Figure 6.23.

If a Tag does not implement recommissioning then it ignores the recommissioning bits and treats them as though they were zero, meaning that, if the Tag receives a properly formatted *Kill* command sequence with the correct kill password it kills itself dead regardless of the values of the recommissioning bits.

Tag's whose kill password is zero do not execute a kill or a recommissioning operation; if such a Tag receives a *Kill* command it ignores the command and backscatters an error code (see Figure 6.23).

The Tag reply to the first *Kill* command shall be as shown in Table 6.39. The reply shall use the TRext value specified in the *Query* command that initiated the round.

After issuing the second *Kill* command an Interrogator shall transmit CW for the lesser of T_{REPLY} or 20ms, where T_{REPLY} is the time between the Interrogator's second *Kill* command and the Tag's backscattered reply. An Interrogator may observe several possible outcomes from a *Kill* command sequence, depending on the success or failure of the Tag's kill or recommissioning operation:

- **The *Kill* or recommissioning succeeds:** After completing the operation the Tag shall backscatter the reply shown in Table 6.40 and Figure 6.22 comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle. If the Interrogator observes this reply within 20 ms then the operation completed successfully. If the Tag is killed dead then immediately after this reply the Tag shall render itself silent and shall not respond to an Interrogator thereafter.
- **The Tag encounters an error:** The Tag shall backscatter an error code during the CW period rather than the reply shown in Table 6.40 (see Annex I for error-code definitions and for the reply format).
- **The *Kill* or recommissioning does not succeed:** If the Interrogator does not observe a reply within 20ms then the operation did not complete successfully. The Interrogator may issue a *Req_RN* (containing the Tag's handle) to verify that the Tag is still in the Interrogator's field, and may reinitiate the multi-step kill procedure outlined in Figure 6.23.

A *Kill* shall be prepended with a frame-sync (see 6.3.1.2.8).

Upon receiving a valid *Kill* command sequence a Tag shall render itself killed or recommissioned, as appropriate. The Tag's reply to the second *Kill* command shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. a Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the round).

Table 6.37 – First *Kill* command

	Command	Password	RFU/Recom	RN	CRC-16
# of bits	8	16	3	16	16
description	11000100	(½ kill password) ⊗ RN16	000_2	handle	

Table 6.38 – Second *Kill* command

	Command	Password	RFU/Recom	RN	CRC-16
# of bits	8	16	3	16	16
description	11000100	(½ kill password) ⊗ RN16	Recommissioning bits	handle	

Table 6.39 – Tag reply to the first *Kill* command

	RN	CRC-16
# of bits	16	16
description	handle	

Table 6.40 – Tag reply to a successful *Kill* procedure

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

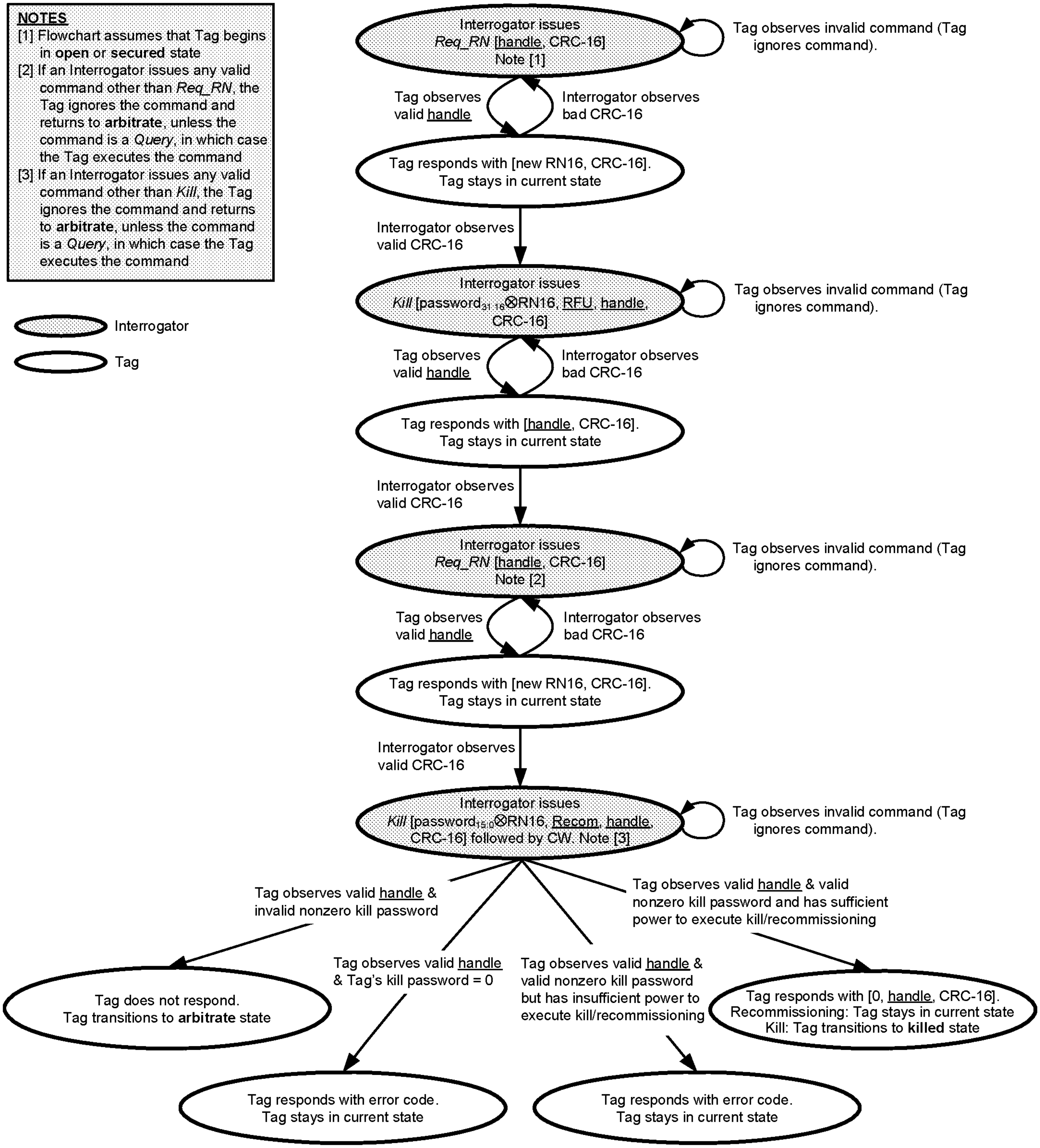


Figure 6.23 – *Kill* procedure

6.3.2.11.3.5 *Lock* (mandatory)

Interrogators and Tags shall implement the *Lock* command shown in Table 6.41 and Figure 6.24. Only Tags in the **secured** state shall execute a *Lock* command. *Lock* allows an Interrogator to:

- Lock individual passwords, thereby preventing or allowing subsequent reads and writes of that password,
- Lock individual memory banks, thereby preventing or allowing subsequent writes to that bank, and
- Permalock (make permanently unchangeable) the lock status for a password or memory bank.

Lock contains a 20-bit payload defined as follows:

- The first 10 payload bits are Mask bits. A Tag shall interpret these bit values as follows:
  - Mask = 0: Ignore the associated Action field and retain the current lock setting.
  - Mask = 1: Implement the associated Action field and overwrite the current lock setting.
- The last 10 payload bits are Action bits. A Tag shall interpret these bit values as follows:
  - Action = 0: Deassert lock for the associated memory location.
  - Action = 1: Assert lock or permalock for the associated memory location.

The functionality of the various Action fields is described in Table 6.43.

The payload of a *Lock* command shall always be 20 bits in length.

If an Interrogator issues a *Lock* command whose Mask and Action fields attempt to change the lock status of a nonexistent memory bank or nonexistent password, the Tag shall ignore the entire *Lock* command and instead backscatter an error code (see Annex I).

The *Lock* command differs from the optional *BlockPermalock* command in that *Lock* reversibly or permanently locks a password or an entire EPC, TID, or User memory bank in a writeable or unwriteable state, whereas *BlockPermalock* permanently locks blocks of User memory in an unwriteable state. Table 6.50 specifies how a Tag shall react to a *Lock* command that follows a prior *BlockPermalock* command, or vice versa.

Permalock bits, once asserted, cannot be deasserted except by recommissioning the Tag (see 6.3.2.10). If a Tag receives a *Lock* whose payload attempts to deassert a previously asserted permalock bit, the Tag shall ignore the *Lock* and backscatter an error code (see Annex I). If a Tag receives a *Lock* whose payload attempts to reassert a previously asserted permalock bit, the Tag shall ignore this particular Action field and implement the remainder of the *Lock* payload, unless the Tag has been recommissioned and the corresponding memory location is no longer permalocked, in which case the Tag shall reassert the permalock bit.

A Tag's lock bits cannot be read directly; they can be inferred by attempting to perform other memory operations.

All Tags shall implement memory locking, and all Tags shall implement the *Lock* command. However, Tags need not support all the Action fields shown in Figure 6.24, depending on whether the password location or memory bank associated with an Action field exists and is lockable and/or unlockable. Specifically, if a Tag receives a *Lock* it cannot execute because one or more of the passwords or memory banks do not exist, or one or more of the Action fields attempt to change a permalocked value, or one or more of the passwords or memory banks are either not lockable or not unlockable, the Tag shall ignore the entire *Lock* and instead backscatter an error code (see Annex I). The only exception to this general rule relates to Tags whose only lock functionality is to permanently lock **all** memory (i.e. all memory banks and all passwords) at once; these Tags shall execute a *Lock* whose payload is $FFFFF_h$, and shall backscatter an error code for any payload other than $FFFFF_h$.

A *Lock* shall be prepended with a frame-sync (see 6.3.1.2.8).

After issuing a *Lock* an Interrogator shall transmit CW for the lesser of T_{REPLY} or 20ms, where T_{REPLY} is the time between the Interrogator's *Lock* command and the Tag's backscattered reply. An Interrogator may observe several possible outcomes from a *Lock*, depending on the success or failure of the Tag's memory-write operation:

- **The *Lock* succeeds:** After completing the *Lock* the Tag shall backscatter the reply shown in Table 6.42 and Figure 6.22 comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle. If the Interrogator observes this reply within 20 ms then the *Lock* completed successfully.
- **The Tag encounters an error:** The Tag shall backscatter an error code during the CW period rather than the reply shown in Table 6.42 (see Annex I for error-code definitions and for the reply format).
- **The *Lock* does not succeed:** If the Interrogator does not observe a reply within 20ms then the *Lock* did not complete successfully. The Interrogator may issue a *Req_RN* command (containing the Tag's handle) to verify that the Tag is still in the Interrogator's field, and may reissue the *Lock*.

Upon receiving a valid *Lock* command a Tag shall perform the commanded lock operation. The Tag's reply to a *Lock* shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. a Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the round).

Table 6.41 – *Lock* command

	Command	Payload	RN	CRC-16
# of bits	8	20	16	16
description	11000101	Mask and Action Fields	handle	

Table 6.42 – Tag reply to a *Lock* command

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

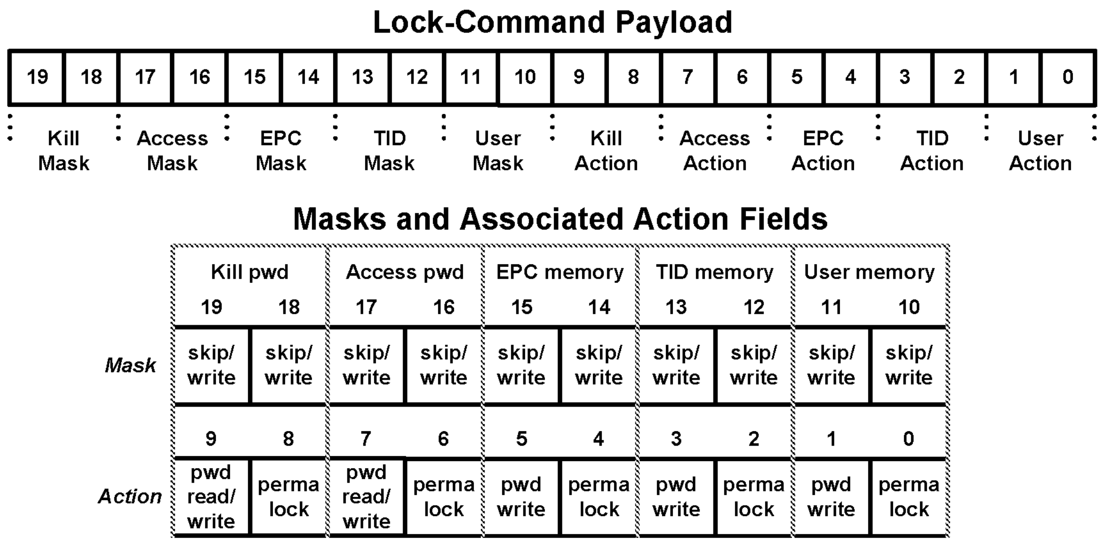


Figure 6.24 – *Lock* payload and usage

Table 6.43 – *Lock* Action-field functionality

pwd-write	permalock	Description
0	0	Associated memory bank is writeable from either the **open** or **secured** states.
0	1	Associated memory bank is permanently writeable from either the **open** or **secured** states and may never be locked.
1	0	Associated memory bank is writeable from the **secured** state but not from the **open** state.
1	1	Associated memory bank is not writeable from any state.
pwd-read/write	**permalock**	**Description**
0	0	Associated password location is readable and writeable from either the **open** or **secured** states.
0	1	Associated password location is permanently readable and writeable from either the **open** or **secured** states and may never be locked.
1	0	Associated password location is readable and writeable from the **secured** state but not from the **open** state.
1	1	Associated password location is not readable or writeable from any state.

6.3.2.11.3.6 *Access* (optional)

Interrogators and Tags may implement an *Access* command; if they do, the command shall be as shown in Table 6.44. *Access* causes a Tag with a nonzero-valued access password to transition from the **open** to the **secured** state (a Tag with a zero-valued access password is never in the **open** state — see Figure 6.19) or, if the Tag is already in the **secured** state, to remain in **secured**.

To access a Tag, an Interrogator shall follow the multi-step procedure outlined in Figure 6.25. Briefly, an Interrogator issues two *Access* commands, the first containing the 16 MSBs of the Tag's access password EXORed with an RN16, and the second containing the 16 LSBs of the Tag's access password EXORed with a different RN16. Each EXOR operation shall be performed MSB first (i.e. the MSB of each half-password shall be EXORed with the MSB of its respective RN16). Just prior to issuing each *Access* command the Interrogator first issues a *Req_RN* to obtain a new RN16.

Tags shall incorporate the necessary logic to successively accept two 16-bit subportions of a 32-bit access password. Interrogators shall not intersperse commands other than *Req_RN* between the two successive *Access* commands. If a Tag, after receiving a first *Access*, receives any valid command other than *Req_RN* before the second *Access* it shall return to **arbitrate**, unless the intervening command is a *Query*, in which case the Tag shall execute the *Query* (inverting its **inventoried** flag if the session parameter in the *Query* matches the prior session).

An *Access* shall be prepended with a frame-sync (see 6.3.1.2.8).

The Tag reply to an *Access* command shall be as shown in Table 6.45. If the *Access* is the first in the sequence, then the Tag backscatters its handle to acknowledge that it received the command. If the *Access* is the second in the sequence and the entire received 32-bit access password is correct, then the Tag backscatters its handle to acknowledge that it has executed the command successfully and has transitioned to the **secured** state; otherwise the Tag does not reply and returns to **arbitrate**. The reply includes a CRC-16 calculated over the handle.

Table 6.44 – *Access* command

	Command	Password	RN	CRC-16
# of bits	8	16	16	16
description	11000110	(½ access password) ⊗ RN16	handle	

Table 6.45 – Tag reply to an *Access* command

	RN	CRC-16
# of bits	16	16
description	handle	

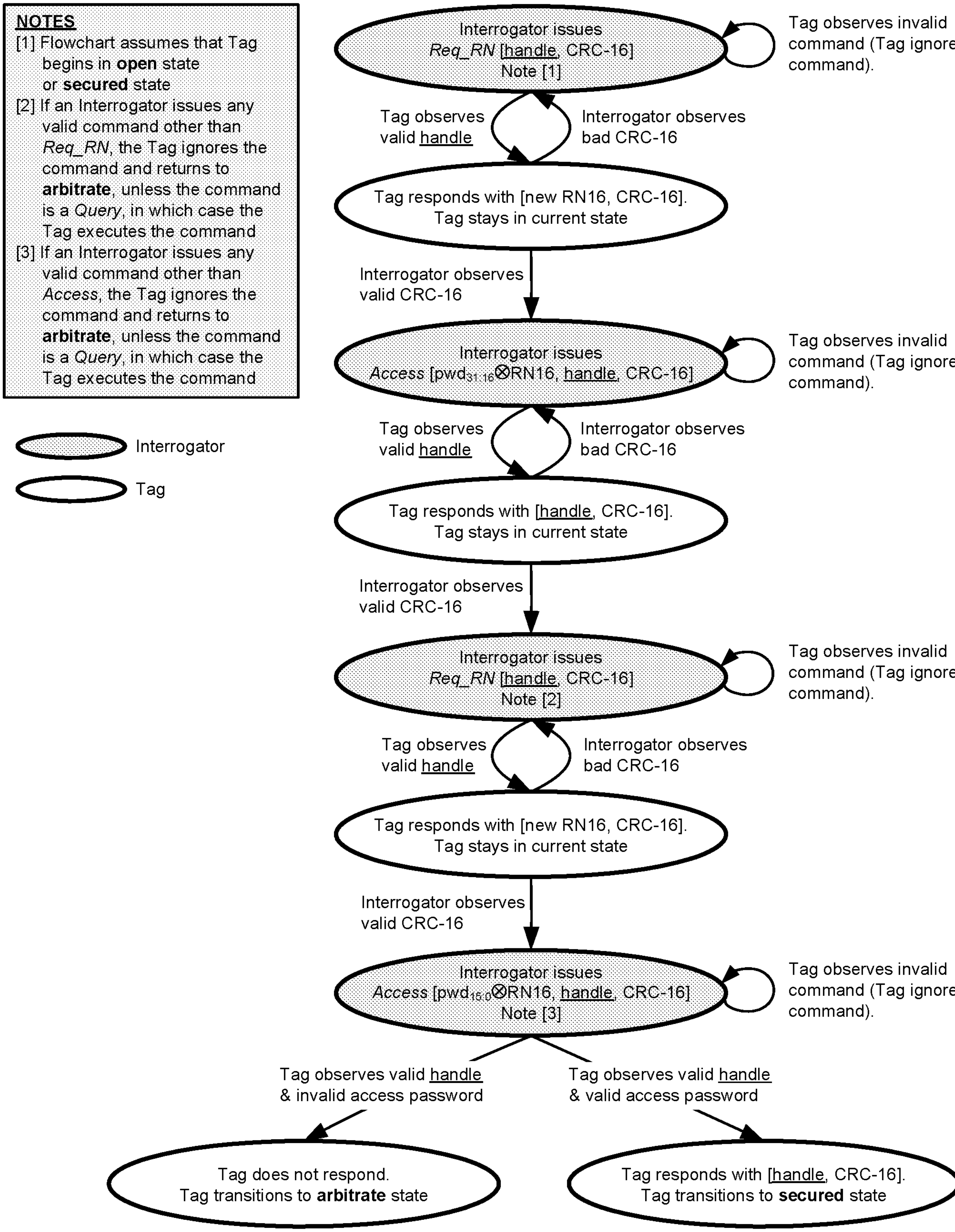


Figure 6.25 – *Access* procedure

6.3.2.11.3.7 *BlockWrite* (optional)

Interrogators and Tags may implement a *BlockWrite* command; if they do, they shall implement it as shown in Table 6.46. *BlockWrite* allows an Interrogator to write multiple words in a Tag's Reserved, EPC, TID, or User memory using a single command. *BlockWrite* has the following fields:

- MemBank specifies whether the *BlockWrite* occurs in Reserved, EPC, TID, or User memory. *BlockWrite* commands shall apply to a single memory bank. Successive *BlockWrites* may apply to different banks.
- WordPtr specifies the starting word address for the memory write, where words are 16 bits in length. For example, WordPtr = 00_h specifies the first 16-bit memory word, WordPtr = 01_h specifies the second 16-bit memory word, etc. WordPtr uses EBV formatting (see Annex A).
- WordCount specifies the number of 16-bit words to be written. If WordCount = 00_h the Tag shall ignore the *BlockWrite*. If WordCount = 01_h the Tag shall write a single data word.
- Data contains the 16-bit words to be written, and shall be $16 \times$WordCount bits in length. Unlike a *Write*, the data in a *BlockWrite* are not cover-coded, and an Interrogator need not issue a *Req_RN* before issuing a *BlockWrite*.

The *BlockWrite* command also includes the Tag's handle and a CRC-16. The CRC-16 is calculated over the first command-code bit to the last handle bit.

If a Tag receives a *BlockWrite* with a valid CRC-16 but an invalid handle it shall ignore the *BlockWrite* and remain in its current state (**open** or **secured**, as appropriate).

A *BlockWrite* shall be prepended with a frame-sync (see 6.3.1.2.8).

After issuing a *BlockWrite* an Interrogator shall transmit CW for the lesser of T_{REPLY} or 20ms, where T_{REPLY} is the time between the Interrogator's *BlockWrite* command and the Tag's backscattered reply. An Interrogator may observe several possible outcomes from a *BlockWrite*, depending on the success or failure of the Tag's memory-write operation:

- **The *BlockWrite* succeeds:** After completing the *BlockWrite* a Tag shall backscatter the reply shown in Table 6.47 and Figure 6.22 comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle. If the Interrogator observes this reply within 20 ms then the *BlockWrite* completed successfully.
- **The Tag encounters an error:** The Tag shall backscatter an error code during the CW period rather than the reply shown in Table 6.47 (see Annex I for error-code definitions and for the reply format).
- **The *BlockWrite* does not succeed:** If the Interrogator does not observe a reply within 20ms then the *BlockWrite* did not complete successfully. The Interrogator may issue a *Req_RN* command (containing the Tag's handle) to verify that the Tag is still in the Interrogator's field, and may reissue the *BlockWrite*.

Upon receiving a valid *BlockWrite* command a Tag shall write the commanded Data into memory. The Tag's reply to a *BlockWrite* shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. a Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the round).

Table 6.46 – *BlockWrite* command

	Command	MemBank	WordPtr	WordCount	Data	RN	CRC-16
# of bits	8	2	EBV	8	Variable	16	16
description	11000111	00: Reserved 01: EPC 10: TID 11: User	Starting address pointer	Number of words to write	Data to be written	handle	

Table 6.47 – Tag reply to a successful *BlockWrite* command

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

6.3.2.11.3.8 ***BlockErase*** (optional)

Interrogators and Tags may implement a *BlockErase* command; if they do, they shall implement it as shown in Table 6.48. *BlockErase* allows an Interrogator to erase multiple words in a Tag's Reserved, EPC, TID, or User memory using a single command. *BlockErase* has the following fields:

- MemBank specifies whether the *BlockErase* occurs in Reserved, EPC, TID, or User memory. *BlockErase* commands shall apply to a single memory bank. Successive *BlockErases* may apply to different banks.
- WordPtr specifies the starting word address for the memory erase, where words are 16 bits in length. For example, WordPtr = 00_h specifies the first 16-bit memory word, WordPtr = 01_h specifies the second 16-bit memory word, etc. WordPtr uses EBV formatting (see Annex A).
- WordCount specifies the number of 16-bit words to be erased. If WordCount = 00_h the Tag shall ignore the *BlockErase*. If WordCount = 01_h the Tag shall erase a single data word.

The *BlockErase* command also includes the Tag's handle and a CRC-16. The CRC-16 is calculated over the first command-code bit to the last handle bit.

If a Tag receives a *BlockErase* with a valid CRC-16 but an invalid handle it shall ignore the *BlockErase* and remain in its current state (**open** or **secured**, as appropriate).

A *BlockErase* shall be prepended with a frame-sync (see 6.3.1.2.8).

After issuing a *BlockErase* an Interrogator shall transmit CW for the lesser of T_{REPLY} or 20ms, where T_{REPLY} is the time between the Interrogator's *BlockErase* command and the Tag's backscattered reply. An Interrogator may observe several possible outcomes from a *BlockErase*, depending on the success or failure of the Tag's memory-erase operation:

- **The *BlockErase* succeeds:** After completing the *BlockErase* a Tag shall backscatter the reply shown in Table 6.49 and Figure 6.22 comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle. If the Interrogator observes this reply within 20 ms then the *BlockErase* completed successfully.
- **The Tag encounters an error:** The Tag shall backscatter an error code during the CW period rather than the reply shown in Table 6.49 (see Annex I for error-code definitions and for the reply format).
- **The *BlockErase* does not succeed:** If the Interrogator does not observe a reply within 20ms then the *BlockErase* did not complete successfully. The Interrogator may issue a *Req_RN* command (containing the Tag's handle) to verify that the Tag is still in the Interrogator's field, and may reissue the *BlockErase*.

Upon receiving a valid *BlockErase* command a Tag shall erase the commanded memory words. The Tag's reply to a *BlockErase* shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. a Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the round).

Table 6.48 – *BlockErase* command

	Command	MemBank	WordPtr	WordCount	RN	CRC-16
# of bits	8	2	EBV	8	16	16
description	11001000	00: Reserved 01: EPC 10: TID 11: User	Starting address pointer	Number of words to erase	handle	

Table 6.49 – Tag reply to a successful *BlockErase* command

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

6.3.2.11.3.9 *BlockPermalock* (optional)

Interrogators and Tags may implement a *BlockPermalock* command; if they do, they shall implement it as shown in Table 6.51. *BlockPermalock* allows an Interrogator to:

- Permalock one or more blocks (individual sub-portions) in a Tag's User memory, or
- Read the permalock status of the memory blocks in a Tag's User memory.

A single *BlockPermalock* command can permalock between 0 and 4080 blocks of User memory. The block size is vendor-defined. The memory blocks need not be contiguous.

Only Tags in the **secured** state shall execute a *BlockPermalock* command.

The *BlockPermalock* command differs from the *Lock* command in that *BlockPermalock* permanently locks blocks of User memory in an unwriteable state, whereas *Lock* reversibly or permanently locks a password or an entire memory bank in a writeable or unwriteable state. Table 6.50 specifies how a Tag shall react to a *BlockPermalock* command (with Read/Lock = 1) that follows a prior *Lock* command, or vice versa.

Table 6.50 – Precedence for *Lock* and *BlockPermalock* commands

<table>
<tr><th colspan="3">First Command</th><th colspan="3">Second Command</th><th>Tag Action and Response to 2nd Command</th></tr>
<tr><td rowspan="5">Lock</td><td>pwd-write</td><td>permalock</td><td rowspan="5" colspan="3">BlockPermalock
(Read/Lock = 1)</td><td></td></tr>
<tr><td>0</td><td>0</td><td>Permalock the blocks indicated by Mask; respond as described in this section 6.3.2.11.3.9</td></tr>
<tr><td>0</td><td>1</td><td>Reject the BlockPermalock; respond with an error code</td></tr>
<tr><td>1</td><td>0</td><td>Permalock the blocks indicated by Mask; respond as described in this section 6.3.2.11.3.9</td></tr>
<tr><td>1</td><td>1</td><td>Permalock the blocks indicated by Mask; respond as described in this section 6.3.2.11.3.9</td></tr>
<tr><td rowspan="5" colspan="3">BlockPermalock
(Read/Lock = 1)</td><td rowspan="5">Lock</td><td>pwd-write</td><td>permalock</td><td></td></tr>
<tr><td>0</td><td>0</td><td>Implement the Lock, but do not un-permalock any blocks that were previously permalocked; respond as described in 6.3.2.11.3.5</td></tr>
<tr><td>0</td><td>1</td><td>Reject the Lock; respond with an error code</td></tr>
<tr><td>1</td><td>0</td><td>Implement the Lock, but do not un-permalock any blocks that were previously permalocked; respond as described in 6.3.2.11.3.5</td></tr>
<tr><td>1</td><td>1</td><td>Implement the Lock; respond as described in 6.3.2.11.3.5</td></tr>
</table>

The *BlockPermalock* command has the following fields:

- MemBank specifies whether the *BlockPermalock* applies to EPC, TID, or User memory. *BlockPermalock* commands shall apply to a single memory bank. Successive *BlockPermalocks* may apply to different memory banks. Class-1 Tags shall only execute a *BlockPermalock* command if MemBank = 11 (User memory); if a Class-1 Tag receives a *BlockPermalock* with MemBank<>11 it shall ignore the command and instead backscatter an error code (see Annex I), remaining in the **secured** state. Higher-class Tags may use the other MemBank values to expand the functionality of the *BlockPermalock* command.
- Read/Lock specifies whether a Tag backscatters the permalock status of, or permalocks, one or more blocks within the memory bank specified by MemBank. A Tag shall interpret the Read/Lock bit as follows:
  - Read/Lock = 0: A Tag shall backscatter the permalock status of blocks in the specified memory bank, starting from the memory block located at BlockPtr and ending at the memory block located at BlockPtr+(16×BlockRange)–1. A Tag shall backscatter a "0" if the memory block corresponding to that bit is not permalocked and a "1" if the block is permalocked. An Interrogator omits Mask from the *BlockPermalock* command when Read/Lock = 0.
  - Read/Lock = 1: A Tag shall permalock those blocks in the specified memory bank that are specified by Mask, starting at BlockPtr and ending at BlockPtr+(16×BlockRange)–1.