PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :		(11) International Publication Number:	WO 00/18052
H04J 4/00	A1	(43) International Publication Date:	30 March 2000 (30.03.00)

(21) International Application Number: PCT/US99/20587

(22) International Filing Date: 8 September 1999 (08.09.99)

(30) Priority Data:
09/159,734 24 September 1998 (24.09.98) US

(71) Applicant: OMNIPOINT CORPORATION [US/US]; 3 Bethesda Metro Center, Suite 400, Bethesda, MD 20814 (US).

(72) Inventor: WADE, Roy; 3885 Highview Drive, Colorado Springs, CO 80908 (US).

(74) Agent: LYON & LYON LLP; Hemminger, Steven, D., Suite 4700, 633 West Fifth Street, Los Angeles, CA 90071-2066 (US).

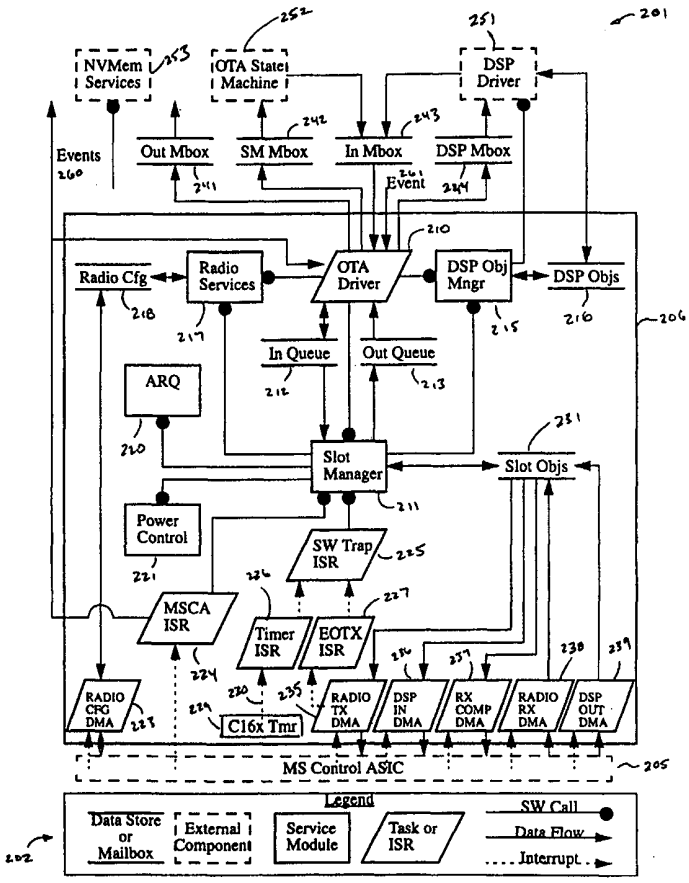
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MANAGEMENT OF TIME SLOT OPERATIONS FOR WIRELESS COMMUNICATION

(57) Abstract

A user station includes a control system for managing time slot operations for a system using aspects of TDMA/TDD and FDMA/FDD, to allow the user station to communicate periodically with a base station. The control system comprises an over-the-air driver for interfacing with upper layer protocol functions, and a slot manager for interfacing with the hardware to handle slot-related functions. The slot manager maintains a list or queue of slot objects, each defining a group of one or more time slots forming a logical communication channel. The slot manager configures the hardware to process each time slot at the appropriate time within a time frame. Different slot objects may be created for voice, signalling or other data traffic, and each slot object may cause the user station to communicate with the base station in one or more time slots of the time frame. The over-the-air driver instructs the slot manager to add, modify or delete slot objects based on requests from upper layer protocol functions. The control system may include service routines for performing data transfers in response to interrupts generated by a hardware controller (such as a control ASIC). The user station hardware may include a set of direct access data channels controlled through the use of interrupts to a peripheral event controller (PEC). The user station control system may also comprise components for radio reconfiguration, ARQ processing and power control processing.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DESCRIPTION

Management Of Time Slot Operations For Wireless Communication

Background Of The Invention

1. Field Of The Invention

The field of the present invention relates to methods and apparatus for managing and controlling time slot operations within a system for multiple access communication.

2. Background

A variety of techniques are known for allowing multiple users to communicate with one or more fixed stations (i.e., base stations) by making use of shared communication resources. Examples of multiple access communication systems include various types of cellular telephone networks and local wireless communication systems, such as wireless private branch exchange (PBX) networks. In such multiple access communication systems, transmissions from different sources may be distinguished in a variety of manners, such as on the basis of different frequencies, time slots, and/or codes, for example.

As referred to herein, a communication system in which transmissions are distinguished according to the transmission frequency may be referred to as a frequency division multiple access (FDMA) communication system. A communication system in which a forward link transmission over one frequency is paired with a reverse link transmission over a different frequency may be referred to as a frequency division duplex (FDD) communication system.

A communication system in which transmissions are distinguished according to the relative timing of the transmission (i.e., by use of time slots) may be referred to as a time division multiple access (TDMA) communication system. A communication system in which a forward link transmission during one time slot (or time segment) is paired with a reverse link transmission occurring during a different time slot (or time segment) may be referred to as a time division duplex (TDD) communication system.

A communication system in which transmissions are distinguished according to which code is used to encode the transmission may be referred to as a code division multiple access (CDMA) communication system. In a CDMA communication system, the data to be transmitted is generally encoded in some fashion, in a manner which causes the signal to be "spread" over a broader frequency range and also typically causes the signal

power to decrease as the frequency bandwidth is spread. At the receiver, the signal is decoded, which causes it to be "despread" and allows the original data to be recovered. Distinct codes can be used to distinguish transmissions, thereby allowing multiple simultaneous communication, albeit over a broader frequency band and generally at a lower power level than "narrowband" FDMA or TDMA systems. Different users may thereby transmit simultaneously over the same frequency without necessarily interfering with one another.

Various "hybrid" communication systems incorporating aspects of more than one multiple access communication technique have been developed or proposed. For example, a GSM system may be viewed as a "hybrid" communication system utilizing aspects of both FDD and TDMA.

In communication systems employing aspects of TDMA or TDD, including "hybrid" type systems, management of slot-related operations and forward and reverse links by a user station can be challenging, due in part to timing and synchronization constraints. For example, the user station may need to precisely control its reception frame window and its transmission frame window in relation to the time frame established by the base station. The user station may also need to process received data (for, e.g., decoding or correcting/processing) while at the same time, or immediately thereafter, having to transmit data in return to the base station.

In addition, the bursty nature of packetized TDMA or TDD communications may lead to particular difficulties at the user station in handling data. For example, bearer or voice data may need to be broken into a number of data packets to be transmitted over several or possibly many time frames, depending on the amount of data to be transmitted, and similarly may need to be reconstructed from data packets received from the base station over several or possibly many time frames. The data may also need to be processed by encoding or decoding algorithms (e.g., vocoding) after being received or before being transmitted, as well as processed for errors, therefore adding to the complexity of control operations at the user station.

In addition to various timing and environmental constraints, a user station may need to have provision for handling different message types, such as signalling (i.e., control) traffic messages as well as normal bearer traffic messages. For example, the user station may need the ability to exchange signalling traffic messages in order to establish communication or handoff to another base station. The user station hardware interface may need to respond rapidly and efficiently to instructions reveled through signalling messages from the base station Also, it may be desirable to support advanced protocol features, such as super-rate or sub-rate communication, at the user station.

In communication systems employing aspects of FDMA or FDD (in addition to TDMA or TDD), further difficulties may be encountered. For example, the radio at the user station may need to be reconfigured periodically to switch between receive and transmit frequencies. The radio reconfiguration may need to occur simultaneous with processing of data just received or about to be transmitted. The user station may also need to control the radio as to allow monitoring of neighboring base stations and handover when needed.

It would therefore be advantageous to provide apparatus and methods for managing time slot operations within a communication system having aspects of TDMA or TDD. It would further be advantageous to provide apparatus and methods which can handle multiple simultaneous or near simultaneous tasks necessary to manage and communicate in time slots within a multiple access communication system. It would further be advantageous to provide a user station capable of processing data received from a base station while also being capable of preparing to transmit information to the base station, and doing so efficiently with a minimal chance of losing or missing data. It would further be advantageous to provide a user station having the ability to dynamically reconfigure its radio to transmit and/or receive on different frequencies (and/or using different codes) between time slots, so as to perform FDMA or FDD (and/or CDMA) communication. It would, in addition, be advantageous to provide such a user station having advanced protocol capabilities, and having, for example, the ability to transmit and/or receive at super-rates or sub-rates. It would also be advantageous to provide a user station control system capable of managing time slots and peripheral activities associated with communication in a TDMA or TDD environment, in addition to dynamic radio re-configuration to support different communication environments and protocols (e.g., FDD or FDMA) and to support handover between base stations.

Summary Of The Invention

The invention provides in one aspect methods and apparatus for managing time slot operations in a wireless communication system, and especially suited for use in a communication system employing aspects of FDMA or FDD as well as TDMA or TDD.

In one embodiment, a user station control system comprises an event-driven architecture utilizing interrupts to communicate with and/or to initiate action by a microcontroller. The microcontroller responds to such interrupts by transferring data to or from various hardware components, such as transmit and receive frame memory buffers, a digital signal processor (DSP) and vocoder, and a radio channel. In one embodiment, the microcontroller responds to a variety of interrupts, each of which requests the transfer of a

discrete amount (e.g., a single byte) of data. Dedicated counters keep track of the total number of bytes to transfer, and interrupt routines activate the microcontroller to enact each data transfer, or to inform the microcontroller that a data transfer has occurred, until the appropriate counter reaches a predefined count value or until otherwise disabled.

In another embodiment, a user station control system comprises a variety of software components for managing and controlling time slot operations. The user station control system may comprise a slot manager for setting up time slots, and for maintaining active time slots in a list or queue format. The user station control system may further comprise an over-the-air (OTA) driver for interfacing with various hardware components (e.g., a DSP/vocoder) and also for instructing the slot manager when to add, update or delete active time slots from the time slot list or queue. The slot manager and OTA driver may communicate via slot message queues. The user station control system also preferably comprises a variety of direct memory access data transfer tasks, which allow access to hardware registers and transference of data between various hardware sources and destinations. In addition, the user station control system may comprise components for radio reconfiguration, ARQ processing and power control processing.

In another embodiment, a microcontroller is programmed in accordance with algorithms for controlling and managing time slot operations in a user station. A control ASIC interfaces with the microcontroller, and provides an interface to various hardware components including, for example, a program memory, random-access memory, DSP/vocoder, linearizer and radio. The microcontroller may comprise a peripheral event controller which responds to interrupts received from the control ASIC. The microcontroller may instruct the control ASIC as to a predefined amount of data (e.g., number of bytes) to be transferred between hardware components, and the control ASIC may then, for each byte to be transferred, interrupt the peripheral event controller of the microcontroller to initiate a direct transfer of the data.

In another embodiment, a user station control system is provided to manage time slot operations, wherein a user station is capable of communicating with more than one base station at the same time. The user station control system preferably maintains, in the form of a slot object, a discrete data profile for each time slot or related group of time slots, and dynamically re-configures the user station radio (including its operative frequency, if applicable) to transmit or receive according to the data profile for the current time slot. When processing for the current time slot is finished, the control and/or configuration parameters for the next time slot are loaded. Each time slot preferably includes within its data profile an indicator (i.e., pointer) identifying the next active time slot. A slot object may be added for each active communication channel (whether

signalling, voice or other data), and may be deleted when the communication channel is closed down. In one aspect, the user station control system provides a capability for the user station to handoff from one base station to another, or monitor nearby base stations, by providing the ability for the user station to communicate contemporaneously (in different time slots of the same time frame) with different base stations.

In another embodiment, a user station control system for managing time slot operations within a TDMA or TDD communication system (and/or FDD or FDMA communication system) provides GSM functionality by interfacing with a subscriber identity module (SIM) card located within the user station. The user station control system may comprise a control ASIC which provides an interface between a microcontroller and the SIM card.

Further embodiments, modifications, variations and enhancements of the invention are also disclosed herein.

Brief Description Of The Drawings

Fig. 1 is a diagram of a cellular system.

Fig. 2 is an abstract diagram of components of a preferred software system and their associated interfaces, in accordance with various aspects of the present invention.

Fig. 3 is a block diagram of a user station or handset in which the invention may be employed.

Fig. 4 is a diagram of a frame structure for FDD communication between a base station and multiple user stations.

Figs. 5 and 6 are diagrams of alternative frame structures for FDD communication between a base station and multiple user stations.

Fig. 7 is a diagram of a process flow for controlling a user station.

Fig. 8 is a data path diagram illustrating the flow of signalling traffic in accordance with a preferred embodiment of a user station.

Fig. 9 is a timing diagram illustrating timing and sequencing for signalling traffic.

Fig. 10 is a data path diagram illustrating the flow of voice or bearer traffic in accordance with a preferred embodiment of a user station.

Fig. 11 is a timing diagram illustrating timing and sequencing for voice or bearer traffic.

Fig. 12 is a timing diagram illustrating slot object access control.

Fig. 13 is a timing diagram depicting a slot interrupt sequence.

Fig. 14 is a table diagram for automatic repeat request (ARQ) states.

Fig. 15 is a top-level system diagram of a preferred control section for a user station in accordance with certain aspects of the invention as described herein.

Fig. 16 is a diagram of a preferred system memory map.

Fig. 17 is a diagram of a memory map for volatile memory space.

Fig. 18 is a diagram of a transmit/receive frame memory structure.

Fig. 19 is an example illustrating a relationship between slot objects and time slots.

Detailed Description Of Preferred Embodiments

Figure 1 is a diagram of a cellular communication system 101 having base stations and user stations. In Fig. 1, a communication system 101 for communication among a plurality of user stations 102 includes a plurality of cells 103, each with a base station 104, typically located at or near the center of the cell 103. Each station (both the base stations 104 and the user stations 102) may generally comprise a receiver and a transmitter. The user stations 102 and base stations 104 preferably communicate using frequency division duplex (FDD) techniques as further described herein, in which base stations 104 communicate over one frequency band and user stations 102 communicate over another frequency band. Communication is also preferably conducted such that different user stations 102 transmit at different times (i.e., during different time slots), as further described herein.

As further shown in Fig. 1, the communication system 101 may also comprise a base station controller 105 which connects to the base stations 104 in a particular geographic region. The base station controller 105 aggregates inputs from multiple base stations 104 and relays information from the base stations 104 to a mobile switching center (MSC) (not shown) and ultimately to a public switched telephone network (PSTN, or "network") (not shown). The base station controller 105 also relays information from the network to the individual base stations 104. The base station controller 105 may, if necessary, perform conversion of signalling messages relating to, e.g., mobility management and call control, to make the signalling messages compatible with the communication protocol used by the base stations 104.

In a preferred embodiment, a wireless communication system includes a frame structure according to which transmissions to and from user stations are distinguished based on time (i.e., TDMA or TDD), and according to which base station transmissions are transmitted on a different frequency band than user station transmissions (i.e., FDMA or FDD). An exemplary frame structure which may be utilized in such a wireless communication system is shown in Fig. 4, and described below. Examples of alternative frame structures are shown in Figs. 5 and 6, and are described hereinafter. However, other

frame structures besides those shown in Figs. 4, 5 and 6 can be used in conjunction with the inventive features described herein.

In the frame structure 401 shown in Fig. 4, a repeating time frame 402 comprises a plurality of time slots 403, each of the time slots 403 having a first time segment 417 and a second time segment 418. The time slots 403 can be defined on multiple frequency bands 411, 412, in order to effectuate FDD or FDMA communication. Accordingly, the time slots 403 can be broken down into base transmit time slots 405 with respect to a base transmission frequency band 411, and user transmit time slots 406 with respect to a user transmission frequency band 412. Each of the base transmit time slots 405 is preferably paired with a corresponding one of the user transmit time slots 406, with such a pair defining a duplex channel for communication (up to N total duplex channels). The N duplex channels may be assigned, on demand, to user stations 102.

In a preferred embodiment, the number of time slots 403 in a time frame 402 is sixteen, and the number of base transmit time slots 405 and user transmit time slots 406 is likewise sixteen. Also in a preferred embodiment, the pairing between base transmit time slots 405 and user transmit time slots 406 is made such that each user transmit time slot 406 is paired with a base transmit time slot 405 that occurs eight time slots later (i.e., one-half of time frame 402). The user stations 102 preferably transmit in the first time segment 417 of the user transmit time slots 406, while the base station 104 preferably transmits in the second time segment 418 of the base transmit time slots 405. In other words, during the first time segment 417 of a user time slot 406, a user station 102 to which the duplex channel is assigned transmits a user station message to the base station 104, and during the second time segment 418 of the corresponding base transmit time slot 405, the base station 104 transmits a base station message to the user station 102 to which the duplex channel is assigned.

The frame structure 401 is designed in one regard to allow relatively convenient adaptation of TDD equipment to perform FDD functionality. Consequently, the base station messages and user station messages are separated in time (i.e., user stations 102 transmit during the first time segments 417 of time slots 403 while the base station 104 transmits in the second time segments 418 of time slots 403) even though time separation would not be necessary to avoid interference because the base station messages and user station messages are also separated by frequency. The frame structure 401 is advantageous when converting a TDD base station using a single frequency band for both base and user transmissions to a base station employing FDD, because the adapted base station can maintain the same relative timing between user transmit time segments and base transmit time segments in each time slot while having only to switch frequencies

between the first time segment 417 and the second time segment 418 in each time slot 403. Consequently, for the frame structure 401 of Fig. 4, the base station 104 need not have the capability of transmitting and receiving simultaneously over two different frequency bands, and does not need, for example, a diplexer.

While preferred embodiments of the invention are described below in relation to the frame structure 401 shown in Fig. 4, alternative frame structures are introduced here which could also be used in conjunction with the inventive features and processes disclosed herein. In particular, alternative frame structures are illustrated in Figs. 5 and 6. For the frame structure 501 shown in Fig. 5, a repeating time frame 502 comprises a plurality of time slots 503, which may be broken down, similar to Fig. 4, as base transmit time slots 505 with respect to a base transmission frequency band 511 and user transmit time slots 506 with respect to a user transmission frequency band 512. Each of the base transmit time slots 505 is preferably paired with a corresponding one of the user transmit time slots 506, with such a pair defining a duplex channel for communication (up to N total duplex channels). However, unlike the frame structure of Fig. 4, the time slots 503 are not divided into a first time segment and second time segment. Rather, during each user transmit time slot 506, the user station 102 to which the duplex channel is assigned transmits a user station message to the base station 104 over the user transmission frequency band 512, and during each base transmit time slot 505, the base station 104 transmits a base transmit message to the user station 102 to which the duplex channel is assigned over the base transmission frequency band 511. The base station 104 thereby carries out multiple access communication with a plurality of user stations 102. Generally, the base station 104 communicates with up to N user stations 102 in the N time slots 503 of time frame 502, unless more than one of the time slots (user time slots 506 or base time slots 505, or both) are aggregated to a single user station 102, or else more than one user station 102 is assigned to share a single time slot 503 by communicating during alternate time frames 502. In one aspect, the frame structure 501 is advantageous in that it can support twice as many user stations 102 as the frame structure 401, all other things being equal, but requires that the base station 104 be able to transmit and receive simultaneously over two different frequency bands 511, 512, unlike the frame structure 401 of Fig. 4.

The alternative frame structure 602 shown in Fig. 6 is, like that of Fig. 4, one in which the base station 104 need not be able to transmit and receive simultaneously over two different frequency bands. For the frame structure 601 shown in Fig. 6, a repeating time frame 602 comprises a plurality of base transmit time slots 605 and a plurality of user transmit time slots 606. Each of the base transmit time slots 605 is preferably paired with a corresponding one of the user transmit time slots 606, with such a pair defining a duplex

channel for communication (up to N total duplex channels). During the first half 602a of the time frame 602, the base station 104 transmits over a base transmission frequency band 611 in each of the base transmit time slot 605 in succession. With respect to the user transmission frequency band 612, the first half 602a of the time frame 602 is "dark" or unused. During the second half 602b of the time frame 602, the user stations 102 transmit in succession over the user transmission frequency band 612. With regard to the base transmission frequency band 611, the second half 602b of the time frame 602 is "dark" or unused. Preferably, the base transmit time slots 605 are paired with user transmit time slots 606 in the same relative sequential order, so that the first base transmit time slot 605 is paired with the first user transmit time slot 606, the second base transmit time slot 605 is paired with the second user transmit time slot 606, and so on. In such a preferred embodiment, each base transmit time slot 605 is paired with a user transmit time slot 606 separated from it by one-half of the time frame 602.

In the embodiments disclosed herein, communication may be established between a base station 104 and a user station 102 by any suitable means, including the methods and techniques described in, for example, U.S. Patent Application Ser. No. 08/284,053 filed August 30, 1994, assigned to the assignee of the present invention, and hereby incorporated by reference as if set forth fully herein. According to one technique described more fully in Ser. No. 08/284,053, a base station 104 transmits a "general poll" message in each time slot which is available for communication. A user station 102 desiring to establish communication with the base station 104 responds to one of the general poll messages by transmitting a general response message to the base station 104, preferably in the same time slot (or the user transmit time slot corresponding to the base transmit time slot) in which the general poll message was transmitted. As part of the general response message, the user station 102 transmits a user station identifier. Upon receipt of the general response message, the base station 104 transmits a specific poll message to the user station 102, preferably in the same time slot as the general poll message was transmitted. Upon receipt of the specific poll message, the user station 102 may transmit a specific response message, or else may commence communication in the appropriate time slot.

In one embodiment, a user station 102 may handoff communication to a base station 104 by first establishing communication with a new base station 104, requesting a circuit switch to the new base station 104, and then relinquishing the original link. The user station 102 may monitor the quality of signals received from other local base stations 104 and, when it appears that a handoff is desirable, establish a communication link with a new base station 104 by responding to a general poll message in an available time slot at

the new base station 104. Once the new communication link is established, the user station 104 instructs the old base station 104 to switch over the circuits for the call, and then terminates the communication link with the old base station 104. Until the old link is relinquished, the user station 102 maintains two separate links in two separate time slots, and potentially on different frequencies and/or with different code sets.

Certain features of the invention are conveniently described with reference to a preferred user station 301 (e.g., a handset) as shown in Fig. 3. As shown in Fig. 3, the user station 301 includes a radio transceiver 305 (comprising a transmitter 315 and a receiver 316), an antenna 306 connected to the radio transceiver 305, and an over-the-air controller 310 also connected to the radio transceiver 305. The over-the-air controller 310 is connected to a memory buffer 311. The over-the-air controller 310 manages the retrieval of information from the memory buffer 311 by the radio transceiver 305 for transmission to the base station 104 with which the user station 301 communicates, and the storage of information into the memory buffer 311 by the radio transceiver 305 when such information is received from the base station 104.

The memory buffer 311 is connected to an analog-to-digital (A/D) converter 331 and a digital-to-analog (D/A) converter 332. Both the A/D converter 331 and the D/A converter 332 are connected to a vocoder 335, which is connected to a speaker/microphone 336. Information received from the base station 104 and stored in the memory buffer 311 is converted from a digital format to an analog format by D/A converter 332. The vocoder 335 then processes the analog formatted information and sends a signal to speaker/microphone 336 to generate audible voice or other sounds to the user/listener. The speaker/microphone 336 also picks up audible voice or other sounds from the user, and relays an analog data signal to the vocoder 335. The vocoder 335 processes the analog data signal, and sends the processed analog data signal to A/D converter 331 for conversion to a digital format. The digital formatted data is then stored in the memory buffer 311, and made available for transmission to the base station 104 in the appropriate time slot.

Figure 3 also shows further details of the over-the-air controller 310. As shown therein, the over-the-air controller 310 comprises a clock 320 connected to a time frame counter 321 and a time slot counter 322. The time frame counter 321 and time slot counter 322 are connected to control logic 323, which uses outputs from the time frame counter 321 and time slot counter 322 to format messages for over-the-air communication. Under control of the over-the-air controller 310, the radio transceiver 305 stores and removes information from the memory buffer 311. The radio transceiver 305 further comprises a transmit/receive (T/R) switch 317 to allow selection between a transmission

mode and a reception mode. The control logic 323 of the over-the-air controller 310 controls the T/R switch 317, and thereby selects between the transmission mode and reception mode based, for example, upon the current portion of a time frame (such as time frame 401 described elsewhere herein). When in a transmission mode, the over-the-air controller 310 selects the user transmit frequency (e.g., user transmission frequency band 412), and when in a reception mode the over-the-air controller 310 selects the base transmit frequency (e.g., base transmission frequency band 411).

In a preferred embodiment, the control logic 323 comprises a microcontroller or microprocessor 327 (such as a Siemens C163/C165 processor) and a control ASIC 328 (referred to herein occasionally as the "MSCA"). The MSCA 328 may provide an interface between the microprocessor 327 and many of the other hardware components in the user station 301, including the radio transceiver 305 and the speaker/microphone 336, as well as a coder/decoder (CODEC) and digital signal processing circuitry. The MSCA 328 may comprise a variety of general and special purpose registers in order to facilitate performance of its various functions. Examples of such registers may include the following:

- Unencoded Receive Data Register
- Encoded Transmit Data Register
- Slot Map Pointer Register
- Test Register
- General Status Register
- Interrupt Control Register
- Compare Input Register
- Compare Count Register
- DSP Data/Control Bytes Register
- DSP Source Register
- Receive Source Register
- Transmit Destination Register
- Sleep Control Register
- Receive Status Register
- RSSI Byte I & 2 Register
- RC Count Register
- Protocol Control Register
- Radio Configuration/Slot Map Data Register
- Radio Configuration Code Register
- General Purpose Register (write)

- General Status Register

In a preferred embodiment, the microcontroller comprises a peripheral event controller (PEC), which receives occasional interrupts from the MSCA 328 for data transfer or similar tasks. A radio transceiver control process preferably provides coordinated updates of all registers of the MSCA 328 and signals that are involved in over-the-air or slot-dependent operations.

Further details of what may generally be termed a "control section" 1501 of a user station or handset in accordance with a preferred embodiment are shown in Fig. 15. As shown in the embodiment illustrated in Fig. 15, the control section 1501 comprises an over-the-air line card assembly (OTA LCA) 1502 comprising controller logic 1503 and radio interface logic 1504. The controller logic 1503 may comprise a control ASIC 1591 (similar in general function to control ASIC 328 of Fig. 3). The OTA LCA 1502 is connected to a microcontroller 1505 (e.g., a Siemens SABC-165 microcontroller). In a preferred embodiment, the microcontroller 1505 comprises, among other things, a microprocessor and one or more serial ports (at least one of which is a high speed synchronous serial port). The OTA LCA 1502 and microcontroller 1505 are connected by a set of control lines 1561, including an interrupt link 1542, a control link 1543, a clock signal line 1544, and a set of chip select (CS) lines 1545. In one aspect, the OTA LCA 1502 provides a control interface between the radio transceiver 305 (see Fig. 3) and the microcontroller 1505. The microcontroller 1505 can move data to and from the OTA LCA 1502, read the status of the OCA LCA 1502, and/or issue commands to the OTA LCA 1502 instructing it to perform certain predefined actions.

In addition to the control lines 1561, a system bus 1530 connects the OTA LCA 1502 to the microcontroller 1505. The system bus is also connected to a static RAM (SRAM) 1524 and a flash memory 1525. The SRAM 1524 may, for example, be 128k in size (8-bit words), and the flash memory 1525 may, for example, be 256k in size (16-bit words).

As further shown in Fig. 15, the controller logic 1503 of the OTA LCA 1502 is connected to a keypad 1523, a battery monitor 1522, a digital signal processor/vocoder (DSP/vocoder) 1510 and a linearizer ASIC 1513. Both the controller logic 1503 and the radio interface logic 1504 are connected to a radio interface ASIC 1514, and thereby to the radio transceiver (such as radio transceiver 305 shown in Fig. 3). The controller logic 1503 of the OTA LCA 1502 is also preferably connected via a SIM link 1533 to a SIM port 1520, which is coupled to a subscriber identity module (SIM) card (not shown). In a preferred embodiment, the SIM card is utilized to perform a number of GSM-related and user environment functions. Examples of such include provision of system security and

storage of subscriber-specific information, including such things as subscriber authentication information and subscriber-specific data. The SIM port 1520 is also connected over a status signal line 1534 to the microcontroller 1505.

The microcontroller 1505 is additionally connected to a liquid crystal display (LCD) controller 1512, which acts as the interface between the microcontroller 1505 and a visual display (not shown). The microcontroller 1505 is further connected to a data services port 1521, which can be used for downloading software, or testing and debugging. The data services port 1521 may comprise an asynchronous/synchronous serial interface port operable in two different modes, and which may be alternately connected to an RS232 port for code download, or directly to the asynchronous USART port of the microcontroller 1505.

Figure 16 depicts a system memory map associated with the control section 1501 shown in Fig. 15. As illustrated in Fig. 16, the system memory 1601 is divided into an imaged memory portion 1602 and a non-imaged memory portion 1603. The entirety of the imaged memory portion 1602 may be contained within the same segment of the system memory 1601 (e.g., segment 0). In the embodiment shown in Fig. 16, the imaged memory portion 1602 comprises one or more flash memory sections 1611, 1612 and 1613 (associated with the flash memory 1525 shown in Fig. 15), an SRAM section 1614 (associated with SRAM 1524 shown in Fig. 15), an internal RAM section 1627 (associated with a random-access memory or cache internal to the microcontroller 1505 shown in Fig. 15), and a peripheral/LCA register section 1616 (associated with the OTA LCA 1502 shown in Fig. 15 and various peripheral devices), as well as various unused sections 1615, if any. The "boot up" code for the microcontroller 1505 may be located in the first (i.e., lowest) flash memory section 1611. In addition, memory needed in certain direct memory access (DMA) operations is preferably imaged into segment 0 according to the above structure. The imaged memory portion 1603 in the embodiment shown in Fig. 16 comprises a flash memory section 1618, an SRAM section 1619, a prototype SRAM section 1620 and a prototype flash memory section 1621. The prototype SRAM section 1620 and prototype flash memory section 1621 are intended to assist in code development, and may require the use of additional chip select signal lines.

Figure 17 is a diagram further detailing the SRAM memory map 1701. As shown in Fig. 17, the SRAM is divided into a plurality of general software regions 1702, 1709 and 1710, a base/frequency map 1703, and a plurality of various buffers including a command buffer 1704, SIM buffer 1705, receive buffer 1706, transmit buffer 1707, and linearizer buffer 1708. The transmit buffer 1707 and receive buffer 1706 are generally utilized by the radio transceiver 305 (see Fig. 3), the linearizer buffer 1708 is generally

utilized by the linearizer ASIC 1513, and the SIM buffer is generally utilized by the SIM card (connected via SIM port 1520).

Data may be stored in the SRAM transmit buffer 1707 or receive buffer 1706 according to the format depicted in Fig. 18. According to that data format, a data message storage block 1801 comprises a header field 1808, a D-channel field 1807, a bearer data field 1806, and a vocoder control field 1805, and may also comprise a reserved portion 1804. In a preferred embodiment, the header field 1808 comprises 17 bits (for a message to be transmitted) or 23 bits (for a received message) of information, the D-channel field 1807 comprises one byte of information, the bearer data field 1806 comprises 24 bytes of information, and the vocoder control field 1805 comprises ten bytes of information.

In one aspect, the control section 1501 provides an interrupt-driven architecture for controlling communication transactions. Events which occur external to the microcontroller 1505 are known as peripheral events, and may cause interrupts. These interrupts are typically generated by the OTA LCA 1502 and coupled to the microcontroller 1505 across the interrupt link 1542. The microcontroller 1505 preferably includes internal circuitry, specifically a peripheral event controller (PEC) for handling such interrupts. The interrupts caused by peripheral events may cause the peripheral event controller to initiate data moves. Specifically, the peripheral event controller may respond to an interrupt by performing a DMA-type byte transfer over the system bus 1530. Different applications may require more than one byte of data. In one embodiment, a separate interrupt is used for each byte of data to be transferred (e.g., 20 bytes of data requires 20 individual interrupt requests). The number of data transfers needed for a particular application (and hence the number of interrupts) is preferably known in advance by the software and is based on the type of interrupt. After the required number of bytes have been moved, the microcontroller 1505 vectors to the appropriate interrupt service routine.

In addition to PEC interrupts, standard interrupts may also be generated from other hardware or software sources. The priority level of PEC interrupts and standard interrupts may be set by the software. The OTA LCA 1502 may include an interrupt state machine for handling event signals and generating interrupts as necessary. If the microcontroller 1505 is asleep or in a power down mode when an interrupt occurs, the interrupt state machine may instruct the clock circuit (for a sleep or power down mode) and/or reset circuit (for a power down mode) to revive the microcontroller 1505. Once the interrupt state machine sees a response from the reset and/or clock circuit indicating that the microcontroller 1505 is fully operational, the interrupt state machine applies the interrupt to the microcontroller 1505. At such a time, the interrupt state machine will present all the

interrupts that have been thus far requested. For example, if an initial PEC interrupt request originates from a SIM receive process while the microcontroller 1505 was asleep or in power down, and during the time that the microcontroller 1505 is being revived another PEC interrupt request is received from a different source, both interrupt requests are presented to the microcontroller 1505 at the same time. As each interrupt request is serviced, the request is cleared. Only when all interrupt requests presented to the microcontroller 1505 have been serviced is the state machine able to process a new request.

In one embodiment, the types of peripheral event interrupts are given by Table 1-1 below.

Table 1-1

INTERRUPT	SOURCE	DESTINATION
1	SRAM Transmit Buffer	Transmitter
2	Receiver	SRAM Receive Buffer
3	DSP Encoded Data	SRAM Transmit Buffer
4	SRAM SIM Buffer / SIM Receiver	SIM Transmitter
5	Linearizer SRAM Buffer Linearizer Data/Control	Linearizer Data/Control Linearizer SRAM Buffer
6	SRAM Command Buffer	Command Compare
7	SRAM Receive Buffer	Receive DSP Data/Control
8	FLASH	Radio Config. / Slot Map Data
9	OTA LCA Error	General Status

The peripheral event interrupts may be asserted by application of a high pulse to an assigned input pin of the microcontroller 1505 (e.g., pins EX0IN - EX7IN and pin T2IN of the Siemens SABC-165 microcontroller).

The functions of the peripheral event interrupts may be explained in more detail as follows. When the OTA LCA 1502 is in need of data to transmit, it asserts a type 1 interrupt (according to Table 1-1), and the microcontroller 1505 responds with an operation including the steps of reading data from the SRAM transmit buffer 1707 (see Fig. 17) and writing the data to the OTA LCA transmit register (0E010H). The data is

thereafter transferred to the radio transceiver 305 (see Fig. 3) at the appropriate time for transmission.

For the reception of data, the OTA LCA 1502 asserts a type 2 interrupt (according to Table 1-1) to request that data in the OTA LCA receive register (0E011H) be placed into the receive SRAM receive buffer 1706 (see Fig. 17). The microcontroller 1505 responds with an operation including the steps reading the data from the OTA LCA receive register and writing the data to the SRAM receive buffer 1706.

A type 3 interrupt and type 7 interrupt (according to Table 1-1) are used in conjunction with various actions performed by or in relation to the DSP/vocoder 1510. The DSP/vocoder 1510, as described below, encodes data to be transmitted and decodes received data. For the transmit process, encoded data is transferred from the DSP/vocoder 1510 to the OTA LCA 1502 at certain time intervals. When a type 3 interrupt is asserted, the microcontroller 1510 transfers the data from the OTA LCA encoded data register (0E012H) to the SRAM transmit buffer 1707 for transmission at a later time. A type 7 interrupt is used in conjunction with the receipt of data and the transfer of that data to the DSP/vocoder 1510. The DSP/vocoder 1510 expects control information to follow the transmission of bearer information. When data is received from the radio transceiver 305 and stored in the SRAM receive buffer 1706 (see Fig. 17), the OTA LCA 1502 asserts a type 7 interrupt. In response, the microcontroller 1505 writes a byte of DSP/vocoder data or control information from the SRAM receive buffer 1706 to the DSP data/control bytes register (0E013H). This process may be repeated as necessary until all the pertinent data has been transferred to the DSP/vocoder 1510.

A type 4 interrupt is used in conjunction with activities performed by or in relation to the SIM card connected to the SIM port 1520. To transfer data to the SIM card, data is moved from the SIM buffer 1705 in the SRAM 1701 to the SIM transmit register (0E006H) within the OTA LCA 1502 in response to assertion of a type 4 interrupt. As data is needed, the OTA LCA 1502 asserts the type 4 interrupt to ensure a continuous flow of data to the SIM card. Once the SIM card receives all of its command and data information, it responds with output data for further processing by the microcontroller 1505. The SIM output data is moved from the SIM transmit register (0E006H) to the SIM buffer 1705 also through a type 4 interrupt. The microcontroller 1505 preferably takes account of the current state of the SIM card to determine the source and destination of data, as the type 4 interrupt can have more than one effect depending upon the state of the SIM card.

A type 5 interrupt may be used to send commands to the linearizer ASIC 1513. The microcontroller 1505 sends a command to the linearizer ASIC 1513 by writing a

command word to the linearizer command/data location (address 0E019H) in the peripheral address map. When the write is complete, the linearizer state machine will begin to transfer data to the linearizer ASIC 1513. In addition, the linearizer state machine will make supplementary requests for data as each byte is sent out. The microcontroller 1505 may keep track of the number of bytes sent to the linearizer ASIC 1513.

A type 6 interrupt is utilized during the course of receiving data, to pre-validate and/or pre-process the received data prior to the completion of an entire frame of data. To initiate this process, the microcontroller 1505 constructs a set of command templates and masks, placed in the SRAM command buffer 1704, for the information it expects to see with each reception. A compare state machine within the OTA LCA 1502 asserts the type 6 interrupt twice for each comparison operation, a first time to retrieve a byte of the command template from the SRAM command buffer 1704 and a second time to retrieve a corresponding mask byte from the SRAM command buffer 1704. With each incoming command byte from the receive frame, the compare state machine retrieves a command template byte and mask byte (i.e., a command/mask pair) from the SRAM command buffer 1704, and attempts to validate the incoming command. Further details of this process are described later herein.

A type 8 interrupt allows the microcontroller 1505 to change operative radio frequencies. The radio configuration data (e.g., frequency and code set, if applicable) is stored in the flash memory 1525, and is transferred to the RIF ASIC 1514 by transfer of a special command code (referred to as RFLOAD) written by the microcontroller 1505 to the radio configuration/slot map data register (0E002H) of the OTA LCA 1502. The RFLOAD command code governs what type of radio configuration data is to be transferred to the RIF ASIC 1514. When the OTA LCA 1502 receives the RFLOAD command, it asserts a type 8 interrupt to initiate transfer of the radio configuration data to the RIF ASIC 1514.

A type 9 interrupt is used by the OTA LCA 1502 to inform the microcontroller 1505 of an error of the end of the operation. The microcontroller 1505 responds to a type 9 interrupt by reading the general status register to determine the source of the error, if any.

In operation, the control section 1501 assists in the transmission and reception of data by a user station 102 or 301 within a multiple access communication system. The control section 1501 preferably operates according to a frame structure such as disclosed in Figs. 4, 5 or 6 herein, but may also be useful with a wide variety of alternative frame structures. Information is passed to and from the radio transceiver 305 (see Fig. 3) across the radio link 1590 shown in Fig. 15. The information is processed, as hereafter described,

by the RIF ASIC 1514 and OTA LCA 1502, under control of the microcontroller 1505, and transferred to and from the DSP/vocoder 1510 and the CODEC 1511. The DSP/vocoder 1510 may incorporate vocoder functionality, and may provide voice data to and remove voice data from the CODEC 1511 according to an applicable sub-rate speech coder algorithm. Voice data may be formatted as 64K μ-law pulse-coded modulated (PCM) data.

The CODEC 1511 provides an interface between the digital circuitry of the control section 1501 and the analog circuitry of the end application (e.g., speaker/microphone 336, as shown in Fig. 3). The CODEC 1511 may operate at any suitable frequency, such as 2.048 MHz, and may be synchronized to the master radio clock at an 8 KHz rate. A frame sync clock may also be provided to the CODEC 1511. The frame sync clock may be any suitable frequency, such as 8 KHz, and may also be synchronized to the master radio clock. A presently preferred CODEC 1511 is commercially available from Texas Instruments, under the product name TLV320AC36IPT. A presently preferred DSP/vocoder 1510 is also commercially available from Texas Instruments, under the product name TMS320LC53SPZ57.

The compressed data created by the DSP/vocoder 1510 is stored, via the OTA LCA 1502, in the SRAM transmit buffer 1707 prior to transmission to the base station 104. At a time prior to the start of the transmit frame, the OTA LCA 1502 begins requesting data from the DSP/vocoder 1510, which responds with a serial stream of data over the serial link 1540. The DSP/vocoder 1510 may provide a single byte per request from the OTA LCA 1502, and when the data is received by the OTA LCA 1502, a PEC interrupt (i.e., a type 3 interrupt as defined herein) is asserted by the OTA LCA 1502. In response, the microcontroller 1505 writes the data to the SRAM transmit buffer 1707. In a preferred embodiment, an entire frame of data, including bearer (e.g., voice) data and status bytes, will be moved from the DSP/vocoder 1510 into the SRAM transmit buffer 1707 before the transmit frame starts.

After the transmit frame starts, the OTA LCA 1502 asserts a type 1 interrupt (as defined later herein) to request a byte of encoded data from the SRAM transmit buffer 1707. In response, the microcontroller 1505 transfers the encoded (voice) data from the SRAM transmit buffer 1707 to the OTA LCA 1502. The OTA LCA 1502 continues to request encoded data in the same manner until the entire transmit frame is sent.

Received data in compressed form is sent, via the OTA LCA 1502, to the DSP/vocoder 1510 for decompression. The received data is first stored in the SRAM receive buffer 1706, then transferred out of the receive buffer 1706 by the OTA LCA 1502. Prior to receiving the first byte of compressed bearer (e.g., voice) data, the OTA

LCA 1502 writes the sync bytes to the DSP/vocoder 1510. The sync bytes may be stored within the OTA LCA 1502 and transferred to the DSP/vocoder 1510 without intervention by the microcontroller 1505. As data is received over the radio link 1590 from the radio transceiver 305 (see Fig. 3), it is stored temporarily in the OTA LCA 1502. For each byte of data received a PEC interrupt (i.e., type 2 interrupt as defined herein) is asserted by the OTA LCA 1502. In response, the microcontroller 1505 transfers the byte of data in the SRAM receive buffer 1706. After each byte of bearer (e.g., voice) data is received and stored in the SRAM receive buffer 1706, it is serialized by the OTA LCA 1502 and sent to the DSP/vocoder 1510. After receipt of the bearer data is complete, the OTA LCA 1502 generates an interrupt to the microcontroller 1505 requesting the receipt of control bytes for the DSP/vocoder 1510. In response to the interrupt, the microcontroller 1505 transfers the control data to the OTA LCA 1502, which transfers the control data serially to the DSP/vocoder 1510.

During the course of receiving data, it may be desirable to begin validation and pre-processing before receipt of an entire frame of data. To initiate this process, the microcontroller 1505 constructs a set of command templates and masks for those commands that it expects to see with each receive frame. The set of command templates and masks are temporarily stored in the SRAM command buffer 1704. If the user station 301 is monitoring more than one time slot, then a separate area may be set up within the SRAM command buffer 1704 for each time slot that the user station 301 is monitoring.

A single command template and mask together comprise a command/mask pair. The command template of the command/mask pair represents a command (or data) that the microcontroller 1505 expects to see in the next receive frame. The command template is compared against one or more command bytes after they are received as part of the receive frame. The number of command bytes to compare is controlled by a compare count command communicated to the OTA LCA 1502 from the microcontroller 1505. The mask of the command/mask pair provides the ability to ignore certain selected bits of the command template in the comparison operation. For example, in one embodiment, any mask bit of the mask that contains a logical "0" causes the corresponding bit of the command template to be excluded from the compare operation and not affect the outcome.

Once the microcontroller 1505 has built up the set of command/mask pairs in the SRAM command buffer 1604, the microcontroller 1505 then writes a compare count value to the Compare Count register within the OTA LCA 1502. The compare count value indicates how many control bytes are to be compared, including header bytes. The compare count value consequently will ordinarily be equal to at least the number of header bytes (e.g., three bytes, in a preferred embodiment). A compare state machine within the

OTA LCA 1502 reads the compare count value and loads it into a decrementing register in preparation for the series of command compares. The compare state machine then asserts an interrupt (i.e., a type 6 interrupt as defined herein) twice for each comparison operation, once to retrieve a command template byte from the SRAM command buffer 1704 and a second time to retrieve a corresponding mask byte from the SRAM command buffer 1704. With each incoming command byte from the receive frame, the compare state machine uses two interrupts to retrieve a command template byte and mask byte (i.e., a command/mask pair) from the SRAM command buffer 1704.

With each comparison between the unmasked portion of the command template byte and the incoming command byte, the decrementing register (i.e., the Compare Counter) in the OTA LCA 1502 is decremented. Once the receive process has started, each command byte will be compared until the count in the Compare Counter reaches zero, or else a bad comparison is made (i.e., the unmasked portion of the command template byte and the incoming command byte do not match), the compare process terminates; however, the Compare Counter continues to decrement with each received byte. When the Compare Counter reaches zero, the interrupt control register is tested to determine if an interrupt should be issued; if so, a general interrupt may be issued and the compare complete bit will be set. Interrupts may be set to issue at the end of a receive frame and/or at the end of a compare operation.

Once a compare sequence progresses beyond the first byte (i.e., the Compare Counter is decremented at least once), the OTA LCA 1502 will proceed with transferring data to the DSP/vocoder 1510 and SRAM 1524. During this process, RSSI information is preferably generated regardless of when an interrupt is issued and regardless of the outcome of the compare sequence, so as to allow the user station 301 to search for better time slots or a better base station 104 for communication based on signal strength measurements. If a bad comparison with the compare/mask pair occurs after the first byte but before the Compare Counter reaches zero, the PEC interrupts used for storing data in the SRAM command buffer 1704 are discontinued to save power. When the Compare Counter reaches zero, the entire receive frame is stored in the SRAM receive buffer 1706.

If the receive frame from the base station 104 comprises a general poll message or specific poll message (both of which may be used in establishing communication between a base station 104 and a user station, as described previously herein), then the above process may be altered somewhat. In a preferred embodiment, a message from the base station 104 comprises a first header byte comprising a field indicating the packet type (e.g., general poll message, specific poll message, or other type of message). If it is determined that the packet type indicates receipt of a general poll message or a specific

poll message, the compare state machine performs the compare/mask operation using the current command template to see whether a general poll message or specific poll message is a command type that is expected. If so, then the comparison process continues as described above. However, if the receive frame comprises a general poll message or specific poll message but the command template does not indicate that either such message is expected, then the compare state machine will not decrement the Compare Counter, will not store the header byte in the SRAM receive buffer 1706, and will not indicate an error. Instead, the OTA LCA 1502 ignores the remainder of the incoming message (although it may continue to generate RSSI information). At the beginning of the next receive frame, the OTA LCA 1502 makes a similar test for a general poll message or specific poll message, and responds in a similar way, and repeats this process until a general poll message or specific poll message is received or else a time-out occurs (which causes the microcontroller 1505 to reset the compare operation). The result of this special case is that, when the user station 301 is looking for a general poll message or specific poll message, the microcontroller 1505 is only interrupted/awakened on a valid general poll message or specific poll message, which results in a power savings.

The OTA LCA 1502 preferably supports a SIM interface that is compliant with the ISA/IEC 7816-3 interface standard. The SIM interface may be managed by the microcontroller 1505 through various interface registers in the OTA LCA 1502. The microcontroller 1505 is responsible for managing the setup of the SIM port 1520 over the SIM interface. The SIM port 1520 may, if desired, include the capability of inverting or not inverting data, under instruction of the microcontroller 1505. The SIM port 1520 may also include the capability of rearranging data prior to transmission and after reception to conform to a most-significant-byte (MSB) first or least-significant-byte (LSB) first format, so that fewer operations for formatting data are required by the microcontroller 1505.

To transmit data to the SIM card, the microcontroller 1505 writes the data to be transmitted in the SRAM SIM buffer 1705, prior to enabling the SIM interface. Once the data is in the SRAM SIM buffer 1705, the microcontroller 1505 writes the number of bytes of data to a SIM transmit counter, thereby enabling the SIM interface. The OTA LCA 1502 asserts a PEC interrupt (i.e., a type 4 interrupt as defined herein) to obtain the first byte of data from the SRAM SIM buffer 1705, and does so for each byte of data to be transmitted thereafter. With each byte of data retrieved, the SIM transmit counter is decremented until it reaches zero, at which point the transmit process is complete.

In case of a parity error, the SIM interface preferably re-transmits the data autonomously, without the need for intervention by the microcontroller 1505. The SIM interface does not make another transmit request (via a PEC interrupt) until the previous

byte has been transmitted successfully, or a reset from the microcontroller 1505 occurs. If an attempted transmission fails for a predefined number of times (e.g., 3 times), then the General Status register is set to indicate a SIM transmit error and the microcontroller 1505 may issue a reset to the SIM control circuitry via the General Purpose register.

The SIM port 1520 is placed in a receive mode (to receive data from the SIM card) upon power-up, or after a successful transmission sequence to the SIM card has been completed, or when an external SIM reset is performed. After each byte of data is received at the SIM port 1520 from the SIM card, the OTA LCA 1502 asserts an interrupt (i.e., a type 4 interrupt as defined herein) to move the data out of the OTA LCA 1502 and into the SRAM SIM buffer 1705. The microcontroller 1505 preferably keeps track of how many bytes have been transferred (as the hardware does not know how many bytes are to be received), and when the desired number of bytes have been received it resets the SIM state machine. During the receive operation, the microcontroller 1505 may perform a watchdog function to ensure that the SIM card is not disconnected or does not fail during the process.

In addition to providing an interface between the microcontroller 1505 and the SIM port 1520, the OTA LCA 1502 also provides an interface between the microcontroller 1505 and the linearizer ASIC 1513. The linearizer ASIC 1513 is connected to the OTA LCA 1502 over a bi-directional, serial interface 1541 which is used for transporting both commands and data. The OTA LCA 1502 comprises a linearizer interface which may be instructed by the microcontroller 1505 to serialize and transmit data to the linearizer ASIC 1513, or to receive data from the linearizer ASIC 1513. The linearizer interface may also be placed in an autonomous special function mode whereby it periodically loads a temperature compensation instruction in a register accessible to the linearizer ASIC 1513 without requiring intervention by the microcontroller 1505.

The user station 301 may include provisions for power reduction by selectively deactivating circuitry at times when the full processing potential of the device is not needed. When the microcontroller 1505 determines that a power reduction mode is appropriate, it instructs the OTA LCA 1502 to enter the power reduction mode by writing a command to a Sleep Control register. If the microcontroller 1505 is to shut itself completely off, it first stores information in the Interrupt Control register (0E016H) to indicate which events will cause interrupts and return the microcontroller 1505 to full operation. If the microcontroller 1505 desires to save some power but will not be inactive for a long period of time, it may choose to place itself in standby mode with the microcontroller 1505 reduced to its minimum frequency. To enter the standby mode, the microcontroller 1505 writes an appropriate command to the Sleep Control register of the

OTA LCA 1502. The OTA LCA 1502 responds by reducing the frequency of the master clock to the minimum allowable frequency (e.g., 1 MHz). The microcontroller 1505 restores itself to full speed prior to performing an operation by requesting an interrupt which is handled by the OTA LCA 1502. The microcontroller 1505 may also selectively place any or all of the DSP/vocoder 1510, CODEC 1511, digital radio ASIC, and linearizer ASIC 1513 into a power reduction mode, and to selectively restore any device from a power reduction mode to full operation, by issuing commands to the OTA LCA 1502.

Features of the software-based control system for user station 301 will now be described in more detail, with particular reference to Fig. 2 as well as occasionally with reference to features shown in the user station diagram of Fig. 3 and the control section diagram of Fig. 15. While reference is made to the preferred user station configurations illustrated in Figs. 3 and 15, it is to be understood that the inventive concepts and features relating to Fig. 2 are independent of those particular embodiments and will find applicability in systems having a variety of other configurations.

In Fig. 2, a software control system 201 (sometimes referred to herein as the radio-transceiver component of the software, or RTRX) generally comprises the components shown in solid lines, while components to which the control system 201 interfaces are generally shown in dotted lines. In addition to the components shown in Fig. 2, the software control system 201 also comprises a real-time operating system (RTOS) not shown in Fig. 2, which provides a software platform that facilitates interaction or operation of the components shown in Fig. 2. A number of the components of Fig. 2 are described individually below, followed by a description of a preferred interaction and operation of the various components to manage communications of a user station 301.

Two principal components of the software control system 201 include the over-the-air (OTA) driver 210 and the slot manager 211. The OTA driver 210 preferably controls transmit and receive operations for the user station 301, while the slot manager 211 preferably manages slot object-related operations. The OTA driver 210 and slot manager 211 communicate using message queues 212, 213, as further described herein.

The software control system 201 also comprises a variety of interrupt service routines (ISRs), including an MSCA ISR 224, software trap ISR 225, a timer ISR 226, and an end-of-transmit ISR 227, to handle various conditions and events requiring service. In addition, a peripheral event controller (PEC) ISR (not shown) is included in the software control system 201, to handle interrupts generated in conjunction with PEC related direct memory access (DMA) transfers.

As further shown in Fig. 2, the software control system 201 comprises a radio services module 217, which can be accessed by both the OTA driver 210 and the slot manager 211 and provides a common interface to a digital ASIC (hereinafter digital radio ASIC, or DRA) in the radio transceiver 305. The software control system 201 further comprises a digital signal processor (DSP) object manager 215 which, like the radio services module 217, can be accessed by both the OTA driver 210 and the slot manager 211. The general purpose of the DSP object manager 215 is to define and coordinate updates to the DSP/vocoder objects 216. Other parts of the software control system 201 include an automatic repeat request (ARQ) module 220 and a power control module 221, both of which are accessible to the slot manager 211. The ARQ module 220 is involved in processing header information for automatic repeat requests, and the power control module 221 processes power control commands received from the base station 104.

The software control system 201 communicates with other system components through a variety of mailboxes 241, 242, 243 and 244, each preferably having specific assigned functions, as further detailed herein. The OTA driver 210 may, for example, communicate with the OTA state machine 252 through a Slot Message (SM) mailbox 242 (designated SM Mbox in Fig. 2), or with a Digital Signal Processor (DSP) driver 251 through a DSP mailbox 244 (designated DSP Mbox in Fig. 2). The slot manager 211 of the software control system 201 can access various memory locations (including special-purpose registers) through a variety of direct memory access (DMA) interface components 235, 236, 237, 238 and 239, as further detailed herein.

The slot manager 211 is, in one aspect, the heart of the software control system 201. The slot manager 211 may define slot objects 231 and perform slot object related operations. Such operations may include the ability to create, delete, update and query a slot object 231. In addition, the slot manager 211 may carry out all slot operations which must be executed in response to hardware slot synchronous events (e.g., interrupts). The slot manager 211 may utilize its inherent knowledge of interrupts to manage the operations of slot objects 231, to synchronize DSP control and status updates, and to synchronize radio configuration updates.

The slot manager 211 also functions as a service module to the OTA driver 210 and the slot-related interrupt service routines (ISRs). The slot manager 211 may be invoked by the OTA driver 210 to initialize, create, update, query or delete a slot object 231. If a request cannot be serviced due to potential interrupt conflicts, as determined by the slot manager 211, then the OTA driver 210 may queue the request in the slot message input queue 212 for subsequent interrupt-synchronous processing. As part of its interrupt-related processing, the slot manager 211 may process messages that are queued in the slot

message input queue 212, and then queue corresponding responses in the slot message output queue 213. Slot event messages may also be queued in the slot message output queue 213. Slot event messages serve as "unsolicited" notification to the OTA driver 210 that a slot has complete a set of transmit and/or receive transactions. In one aspect, the slot event messages provide a snapshot of the most recent transaction status, including, for example, transmit status, RSSI values, and receive data.

During its various processes, the slot manager 211 may use service functions to set up hardware, configure ISRs, set up PEC DMA paths, and to update slot objects 231. The slot manager 211 may, for example, invoke the DSP object manager 215 to create and initialize DSP objects 216, to commit DSP control updates, or to update DSP status. The slot manager 211 may invoke the radio services module 217 to commit configuration updates, synchronize the radio transceiver to time slots, switch between receive and transmit frequencies, read RSSI values, or set the transmit power level. The slot manager 211 may invoke the ARQ module 220 to control error detection and recovery processing on OTA packets. In addition, the slot manager 211 may invoke the power control module 221 to control the transmit power level of the radio transceiver 305 (see Fig. 3). The slot manager 211 also may use power-on/self-test functions to set up the various PEC DMA paths.

The MSCA ISR 224 handles the general status interrupts that are from time to time generated by the MSCA 1591. Non-slot related interrupts are sensed, then signalled externally to the OTA driver 210 through the external event interface 261. Slot-related interrupts (e.g., interrupts occurring when the radio transceiver 305 receives data on an active slot) are handled by invoking the associated function(s) of the slot manager 211.

The software trap ISR 225 handles software interrupts generated by the software control system 201. The software trap ISR 225 invokes the appropriate function(s) of the slot manager 211 based upon the source of the interrupt. The software trap ISR 225 may be invoked, for example, upon an end-of-transmit interrupt or a one-shot timer interrupt.

The end-of-transmit (EOTX) ISR 227 handles an interrupt generated upon completion of the transmit portion of a slot. The EOTX ISR 227 sets a flag and generates a software trap (thus invoking the software trap ISR 225) to perform end-of-transmit processing. The peripheral event controller of the OTA LCA 1502 invokes the EOTX ISR 227 upon completion of a DMA transfer of transmit data from the SRAM 1524 to the radio transceiver 305.

The timer ISR 226 handles an interrupt generated upon expiration of a local one-shot timer 229. To perform timer expiration processing, the timer ISR 226 may, for example, update a semaphore or set a flag.

The peripheral event controller (PEC) ISRs (not shown in Fig. 2) handle interrupts generated upon completion of PEC-controlled DMA transfers. The PEC ISRs are triggered upon completion of DMA data transfers requested by the OTA LCA 1502 through the associated PEC channel.

A variety of DMA data transfer tasks 223, 235, 236, 237, 238 and 239 are provided for moving data. A radio transceiver DMA data transfer task 235 moves transmit data from SRAM 1524 to the radio transceiver 305. A DSP input DMA data transfer task 236 moves DSP/vocoder control data from the SRAM 1524 to the DSP/vocoder 1510. A radio receive DMA data transfer task 238 moves receive data from the radio transceiver 305 to the SRAM 1524. A DSP output DMA data transfer task 239 moves DSP/vocoder status and transmit data from the DSP/vocoder 1510 to the SRAM 1524. A receiver compare DMA data transfer task 237 moves receive compare data from SRAM 1524 to the OTA LCA 1502. A radio configuration DMA data transfer task 223 moves radio configuration data between the SRAM 1524 and the radio transceiver 305. Each of the above DMA data transfer tasks 223, 235, 236, 237, 238 and 239 interface in some manner with the control ASIC or MSCA 205 (e.g., control ASIC 1591 in Fig. 15, or control ASIC 328 in Fig. 3).

The radio services module 217 provides a common interface to the digital radio ASIC (DRA) in the radio transceiver 305. The radio services module 217 provides operations to configure receive and transmit frequencies, codes, and preambles, to control transmit power, to synchronize the radio to time slots, to read RSSI values, and to configure various thresholds. The radio services module 217 utilizes registers of the OTA LCA 1502 and DMA I/O operations triggered by the control ASIC 205 for communication with the digital radio ASIC (DRA) in the radio transceiver 305.

The DSP object manager 215 defines and coordinates updates to DSP/vocoder objects 216. The DSP object manager 215 interacts with the DSP/vocoder 1510 at an "object" level, thus treating a DSP/vocoder object 216 as a block of data with an associated identifier, while the external DSP driver 251 actually updates the DSP/vocoder bit-level controls and interprets the DSP/vocoder bit-level statuses. The DSP object manager 215 invokes the DSP driver 251 to create, initialize or destroy DSP/vocoder objects 216. Additionally, the DSP object manager 215 accepts updates to the control and status blocks of DSP/vocoder objects 216.

The slot manager 211, as previously described, accesses an ARQ service module 220 and power control service module 221 as needed. The ARQ service module 220, if included, provides over-the-air error detection, error recovery, error notification and message sequencing based upon the particular ARQ mechanism implemented in over-the-air packets. The power control service module 221, if included, controls the transmit

power level of the radio transceiver 305 of the user station 301 based upon the power control mechanism implemented in the over-the-air packets.

Operation of one embodiment in accordance with the software control system 201 shown in Fig. 2 may be explained with reference to the process flow diagram of Fig. 7. As illustrated in Fig. 7, a power-on/reset step 705 initiates a control process 701, and typically involves, for example, a user (i.e., a human being) pushing a power-on button on a handset. Once power has been applied to the circuitry, the control process 701 moves to a self-test step 706, wherein a power-on self test is commenced. During the self-test step 706, a suite of processor and peripheral confidence tests are executed to verify the integrity of the electrical components of the system. If there are no errors as a result of the power-on self-test, control is passed to the real-time operating system (RTOS), and the control process 701 continues with an initialization step 707.

The initialization step 707 comprises a series of sub-steps 721 through 729, as illustrated in Fig. 7. In the first sub-step 721 of the initialization step 707, the OTA driver 210 is started, thereby ensuring the availability of RTRX services to other software components. After starting execution, in a next sub-step 722 the OTA driver 210 performs initialization of the software control system 201. As part of this initialization task, the OTA driver 210 associates an event with the RTRX input mailbox 243, which later causes the operating system to automatically signal when one or more messages reside in the RTRX input mailbox 243. In a next sub-step 723 of the initialization step 707, the peripheral event controller (PEC) ISRs are installed. In the following two sub-steps 774 and 725, respectively, the MSCA ISR 224 and the timer ISR 226 are installed. The slot message input queue 212 and slot message output queue 213 are initialized in a following sub-step 726. In a next sub-step 727, the radio configuration is initialized, followed by sub-step 728 wherein the transceiver-related hardware (i.e., radio transceiver 305 of Fig. 3) is configured to its idle state, and, finally, followed by sub-step 729 wherein the DSP objects 216 are initialized.

After the initialization step 707, the OTA driver 210 blocks further execution, and the control process 701 enters an idle mode (illustrated as a loop between steps 710 and 711) wherein the software control system 201 awaits either a message to be sent to its RTRX input mailbox 243, or a slot event to be signalled by the MSCA ISR 224. The software control system 201 can either execute a software loop where it continuously polls for the above two conditions, or preferably remains completely idle by halting execution and awaiting an interrupt to re-activate execution. The "loop" depicted by steps 710 and 711 in Fig. 7 is intended to encompass at least both those possible embodiments. Accordingly, as shown in Fig. 7, the control process 701 in step 710 senses the occurrence

of a slot event by receiving a signal from the MSCA ISR 244, resulting in a message relating to the slot event to be placed in the slot message output queue 213. Upon sensing a slot event, the OTA driver 210 handles messages queued in the slot message output queue 213, as indicated by step 712, and processes these messages to completion. Typically the slot event messages will lead to the sending of a message to the slot message (SM) mailbox 242 associated with the OTA state machine 252. Additionally, the OTA driver 210 processes messages queued in the slot message input queue 212, as indicated in step 713 of the control process 701, until either no messages remain or a message is required to be re-queued. Once processing in step 713 is complete, the OTA driver 210 again blocks execution and returns to the idle mode.

If no slot event is sensed in step 710, then the control process 701 moves to step 711, wherein the control process 701 senses whether a message has been received in the RTRX input mailbox 243. If so, the OTA driver 210 then attempts to process the message to completion and, if appropriate, sends a corresponding response. However, operations involving data elements shared by the OTA driver 210 and the software modules invoked by the various ISRs 224, 225, 226 and 227 may require coordination to ensure exclusive access. This shared element access is arbitrated through simple, non-RTOS semaphores. Requests which the OTA driver 210 cannot completely process are preferably queued (i.e., placed in the slot message input queue 212) in first-in first-out (FIFO) order.

Upon receiving a message in its slot message input queue 212, the OTA driver 210 invokes the appropriate service module to perform all operations associated with the request which do not require exclusive access. The OTA driver 210 may, for example, invoke the radio services module 217 to initialize the radio configuration and to register updates to the radio configuration. The OTA driver 210 invokes the DSP object manager module 215 to register updates to DSP controls and to request DSP status. The OTA driver 210 invokes the slot manager 211 to initialize, create, update, query, or delete slot objects 231, as further described below. After invoking the service modules which do not require exclusive access, the OTA driver 210 then attempts to obtain the associated semaphore for those service modules or other software components requiring exclusive access. If the OTA driver 210 is granted exclusive access, it finishes servicing the request, using the appropriate service module, and sends a corresponding response message to the return mailbox specified by the request. If the OTA driver 210 is unable to gain exclusive access when needed, the request is placed in the slot message input queue 212 for subsequent interrupt-synchronous processing. While waiting for such to occur, the OTA driver 210 again blocks execution, and returns to the idle mode, waiting for another slot event of input message to the RTRX input mailbox 243.

A slot object 231, as used herein, preferably comprises a data structure corresponding to a logical communication channel. The slot manager 211 is primarily responsible for organizing and maintaining slot objects 231 during communication between the user station 301 and a base station 104. A slot object 231 preferably comprises, among other things, command data and/or configuration information, including, for example, transmit power level information, ARQ information, state machine information, and data pointers (which identify the memory locations of data messages). A slot object 231 also preferably comprises a list of one or more time slot tags corresponding to the time slots which form the logical communication channel. Generally, a slot object 231 will comprise only a single time slot (and thus a single time slot tag); however, it may comprise a "super-slot" (i.e., multiple time slots) or a "sub-slot" (i.e., a time slot utilized once every N time frames). Each time slot tag preferably includes a next slot pointer indicating the next time slot, from among any of the active time slots within the slot objects 231, that is either a transmit or receive time slot for the user station 301. In addition, the slot object 231 preferably comprises information indicating whether its associated time slots are transmit or receive time slots, and whether they are to transmit signalling, voice or other data messages.

To maintain an ordinary duplex communication channel, two slot objects 231 are set up. A first slot object 231 is set up with a single transmit time slot (and hence a single time slot tag), and a second slot object 231 is set up with a single receive time slot (and hence a single time slot tag). The time slot tag of the first slot object 231 indicates in which slot position the user station 301 will transmit, and the first slot object 231 comprises information, as described above, utilized in the transmission process, including a pointer to the message to be transmitted. The time slot tag of the second slot object 231 indicates in which slot position the user station 301 will receive, and the second slot object 231 comprises information, as described above, utilized in the receive process, including a pointer to the appropriate receive message buffer. The time slot tag of the first slot object 231 comprises a next slot pointer which will (in the absence of any other slot objects 231) identify the receive time slot of the second slot object 231 as the next active time slot, while, similarly, the time slot tag of the second slot object 231 comprises a next slot pointer which will identify the transmit time slot of the first slot object 231 as the next active time slot.

If the user station 301 is to transmit signalling information (possibly in addition to voice or data information being exchanged over the ordinarily duplex communication channel), then a slot object 231 may be added to define a logical signalling channel. The new slot object 231 will comprise one or more time slot tags indicating, among other

things, which time slots are to be used to transmit, or receive, signalling information. When the new slot object 231 is added, any other time slot tags within the existing slot objects 231 are modified, as appropriate, to update their next time slot pointers so as to maintain proper sequencing.

After being created, a slot object 231 exists until deleted by the slot manager 211. The user station 301 will communicate each time frame in each time slot defined by the time slot tags of the existing slot objects 231. A slot object 231 may be created simply to communicate in one time slot of one time frame, and then get deleted. This short lifespan may occur when the user station 301 transmits a brief signalling message, for example. For a lengthy call, a slot object 231 may last indefinitely. A slot object 231 may be modified dynamically by the slot manager 211. A slot object 231 may thereby not only define multiple time slots within a time frame, but these time slots can be changed from time frame to time frame, thus allowing, for example, the exercise of fast control traffic operations such as described in, e.g., U.S. Patent Application Ser. No. 09/122,565 filed August 24, 1998, assigned to the assignee of the present invention, and hereby incorporated by reference as if set forth fully herein.

Slot objects 231 are generally created, modified or deleted at the request of upper layer protocol requests. These requests are received and queued by the OTA driver 210 for processing by the slot manager 211. The slot manager 211 ensures that slot objects 231 are only created, modified or deleted at times which do not interfere with crucial or ongoing hardware operations. The slot manager 211 also ensures that the slot object tags of the existing slot objects 231 are modified, if necessary, to reflect new time slots added from addition of a new slot object 231 or old time slots removed by deletion of an existing slot object 231. In one aspect, the slot manager 211 provides isolation between the OTA driver 210 and other components of the software control system 201, and the system hardware and/or physical layer of the user station 301.

In a preferred embodiment, time slots defined by the existing slot objects 231 are processed sequentially according to their relative position within the time frame. At the end of processing a given time slot, the control and/or configuration parameters for the next time slot (as indicated by the next slot pointer of the current time slot tag) are loaded into the hardware by the slot manager 211. The loading of the new control and/or configuration parameters may, for example, cause the radio transceiver 305 to be switched from a receive mode to a transmit mode, or vice versa, and may also cause the radio transceiver 305 to load a new spread spectrum code set, if desired. Because the loading of the control and/or configuration parameters may take more than a nominal amount of time, in a preferred embodiment at least one time slot separates each active time slot for the user

station 301, unless adjacent active time slots employ the same control and configuration parameters such that the loading of new parameters is unnecessary.

An example of slot objects 231 showing their relationship to the time slots of a time frame appears in Fig. 19. In the example shown there, a collection of three slot objects 1902, 1903 and 1904 has been created by the slot manager 211. Each slot object 1902, 1903 and 1904 comprises a slot profile which includes command data and/or configuration information, such as the slot type (transmit or receive), message type (voice, signalling, data, etc.), and other information (such as transmit power level, ARQ information, etc.) as described above. Each slot object 1902, 1903 and 1904 has at least one time slot tag identifying an active time slot for the user station 301. The first slot object 1902 has one time slot tag 1911 corresponding to transmit time slot #1 of a time frame 1920, and indicating (via a next slot pointer) that the next active time slot for the user station 301 is time slot #5. The second slot object 1903 has one time slot tag 1912 corresponding to receive time slot #9, and indicating (via a next slot pointer) that the next active time slot for the user station 301 is time slot #12. According to their respective slot profiles, the first and second slot objects 1902, 1903 are set up for voice traffic. In one aspect, the first and second slot objects 1902, 1903 may collectively comprise a duplex voice channel, according to the frame structures of Figs. 4, 5 or 6, for example. The third slot object 1904 has three time slot tags 1913, 1914 and 1915. Its first time slot tag 1913 corresponds, as shown, to receive time slot #5; the second time slot tag 1914 corresponds to receive time slot #12; and the third time slot tag 1915 corresponds to receive time slot #14. In one aspect, the three time slot tags 1913, 1914 and 1915 of the third slot object collectively define a receive communication channel for signalling data for the user station 301. If desired, a fourth slot object could be added to the collection of slot objects to define a transmit communication channel for signalling data.

Further details of the preferred operation, timing and sequencing of the various components of the software control system 201 will now be described.

In one aspect, the OTA driver 210 assists in control of the operation of the software control system 201. The OTA driver 210 performs initialization of the software control system 201, monitors the RTRX input mailbox 243, routes requests to service modules, monitors the slot message input queue 212 and slot message output queue 213, processes slot events, and responds to the RTRX output mailbox 241.

When a slot event requires processing, the OTA driver 210 preferably processes the slot event to completion before the next slot event is signalled. The slot manager 211 may indirectly control this sequencing by ensuring that at least one inactive slot (e.g., 1.25 milliseconds, in one embodiment) occurs between active slots.

Table 2-1 below lists some of the inputs to the OTA driver 210 and their sources, as well as comments regarding each.

Table 2-1

Item	Source	Function or Comment
Mailbox Event	RTOS	Signalled when one or more messages reside in the RTRX Input Mailbox.
External Message	Input Mailbox	See description herein regarding external message processing
Slot Event	MSCA ISR	Signalled upon ISR slot processing completion.
Slot Message Input Queue Message	Self	Queued External Message
Slot Message Output Queue Message	Slot Manager	Queued slot-related message.
Radio Request Response	Radio Services	Initialization and update status.
DSP Request response	DSP Object Manager	Update status and query information.
Slot Request Response	Slot Manager	Initialize, create, update, query, and delete status.

In a preferred embodiment, the OTA driver 210 responds to the following RTRX input mailbox messages: RTRX Configure Request, RTRX Reset Request, Link Request, Radio Configure Request, and DSP Update Request. A brief description of how, in one embodiment, the OTA driver 210 responds to each of these RTRX input mailbox messages is provided below.

When the OTA driver 210 receives an RTRX Configure Request message in the RTRX input mailbox 261, it performs the following processing steps:

1. Updates the RTRX configuration parameters as specified by the RTRX Configure Request message.

2. Sends an RTRX Configure Response message to the return mailbox specified in the RTRX Configure Request message.

The RTRX Configure Request message should be followed by an RTRX Reset Request to commit the new configuration.

When the OTA driver 210 receives an RTRX Reset Request message in the RTRX input mailbox 261, it performs the following processing steps:

1. Shuts down the software control system 201.
2. Re-initializes the software control system 201 according to the initialization steps as previously described herein.
3. Sends an RTRX Reset Response message to the return mailbox specified in the RTRX Reset Request message.

When the OTA driver 210 receives a Link Request message in the RTRX input mailbox 261, it performs the following processing steps:

1. Processes each Link Request, as follows:
   a. Link Create Request - invokes the slot manager 211 to create the slot as specified.
   b. Link Delete Request - invokes the slot manager 211 to delete the slot.
   c. Link Query Request - invokes the slot manager 211 to obtain a snapshot of the slot's latest configuration and status.
   d. Link Update Request - invokes the slot manager 211 to reconfigure the slot as specified by the request.
2. If the slot manager 211 cannot completely service request (because of potential interrupt access conflicts), the OTA driver 210 queues the request in the slot message input queue 212.
3. If the request is successfully serviced, the OTA driver 210 sends the associated Link Response to the return mailbox specified in the Link Request message.

When the OTA driver 210 receives a Radio Configure Request message in the RTRX input mailbox 261, it performs the following processing steps:

1. Invokes the radio services module 217 to update the radio configuration as specified by the Radio Configure Request message. The new configuration is thereafter committed at a synchronous time as determined by the slot manager 211.
2. Sends a Radio Configure Response message to the return mailbox specified in the Radio Configure Request message.

When the OTA driver 210 receives a DSP Update Request message in the RTRX input mailbox 261, it performs the following processing steps:

1. Invokes the DSP object manager 215 to update the DSP object 216 as specified by the DSP Update Request message.

2. Sends a DSP Register Update Confirmation message to the mailbox associated with the DSP object 216.

In addition to RTRX input mailbox messages, the OTA driver 210 also processes messages received in the slot message input queue 212. For example, when the OTA driver 210 receives a Slot Request message in the slot message input queue 212, it performs the following processing steps:

1. Processes each Slot Request message, as follows:

a. Slot Create Request - invokes the slot manager 211 to create the slot as specified by the Slot Create Request.

b. Slot Delete Request - invokes the slot manager 211 to delete the slot.

c. Slot Query Request - invokes the slot manager to obtain a snapshot of the slot's latest configuration and status.

d. Slot Update Request - invokes the slot manager to reconfigure the slot as specified.

2. If the slot manager 211 cannot completely service request, then the OTA driver 210 may re-queue the request in the slot message input queue 212.

3. If the request is successfully serviced, the OTA driver 210 sends the associated Link Response message to the return mailbox specified in the Slot Request message.

In addition, the OTA driver 210 processes messages queued in the slot message output queue 213. For example, when the OTA driver 210 receives a Slot Response message in the slot message output queue 213, it performs the following processing steps:

1. Processes each Slot Response message, as follows:

a. Slot Create Response - sends a Link Create Response message to the return mailbox specified in the request.

b. Slot Delete Response - invokes the slot manager 211 to perform final delete operations on the slot, then sends a Link Delete Response message to the return mailbox specified in the request.

c. Slot Query Response - invokes the slot manager 211 to obtain a snapshot of the slot's latest configuration and status, then sends a Link Query Response message to the return mailbox specified in the request.

d. Slot Update Response - sends a Link Update Response message to the return mailbox specified in the request.

When the OTA driver 210 receives a Slot Receive Complete message in the slot message output queue 213, it performs the following processing steps, depending on the conditions described below:

1. If the slot manager 211 indicates that a Link Event Indication message should be issued, the OTA driver 210 then:

   a. Invokes the slot manager 211 to obtain a snapshot of the slot's latest configuration and status.

   b. Sends a Link Event Indication message to the mailbox associated with the slot object 231.

2. If the DSP object manager 215 indicates that the status of the associated DSP object 216 was updated, the OTA driver 210 then:

   a. Sends a DSP Register Update Indication message to the mailbox associated with the DSP object 216.

Upon detecting a slot event, the OTA driver 210 processes all messages queued in the slot message output queue 213 to completion, as specified in the process steps above, and then processes all messages queued in the slot message input queue 212 to completion, as specified in the process steps above, or until forced to re-queue a message due to an access conflict or for some other reason.

If an error is detected during operation, the OTA driver 210 sends an Error Indication message to the RTRX output mailbox 241. Examples of error conditions include an unrecognized external message received from the RTRX input mailbox 243, or an unrecognized message received from the slot message input queue 212 or slot message output queue 213.

Table 2-2 below lists some of the outputs from the OTA driver 210 and their destinations, as well as comments regarding some of the OTA driver outputs.

Table 2-2

Item	Destination	Comment
External Message	Output Mailbox	
External Message	State Machine Mailbox	
External Message	DSP Mailbox	
Slot Message Input Queue Message	Self or Slot Manager	Queued External Message
Radio Requests	Radio Services	Initialize and update.
DSP Requests	DSP Object Manager	Update and query.
Slot Requests	Slot Manager	Initialize, create, update, query and delete.

The slot manager 211, as previously described, preferably defines all slot objects 231 and performs all operations related to the slot objects 231. For example, the slot manager 211 may create, delete, update or query slot objects 231. In addition, the slot manager 211 handles those slot operations which are executed in response to hardware slot-synchronous events (e.g., interrupts), and can use its inherent knowledge of interrupts to manage the operations on slot objects 231 and perform other functions.

Table 3-1 below lists major inputs to the slot manager 211 and their sources, along with comments.

Table 3-1

Item	Source	Comment
Slot Requests	Any	Initialize, create, update, query, delete, and interrupt processing requests.
Slot Message Input Queue Message	OTA Driver	Queued slot request
Radio Request Response	Radio Services	Request status, RSSI.
DSP Request Response	DSP Object Manager	Initialization status, DSP control.
ARQ Request Response	ARQ	ARQ status and message sequence number.
Power Control Request Response	Power Control	Power Level recommendation
PECCx Event Counters	PEC ISRs	End-of-PEC event counters.

Table 3-2 below illustrates data elements used by the slot manager 211 in a preferred embodiment, along with a brief description of each.

Table 3-2

Element	Comment
Slot List	Static list of Slot Structures.
Current Slot	Slot List reference to the current active slot.
Slot Message Input Queue	Queue slot-related requests.
Slot Message Output Queue	Queue slot--related responses and events.

Table 3-3 below describes a structure which is used in one embodiment comprising the slot manager 211 containing the configuration and status of a slot.

Table 3-3

Element	Comment
Slot Identifier	Unique slot identifier.
Slot Type	Current slot type.
Slot State	Current slot state.
Route Configuration	Addressing information for Link Event Indication messages for this slot.
Receive Configuration	Current receive configuration.
Receive Configuration Update	New receive configuration.
Receive Status	Current receive status.
Transmit Configuration	Current transmit configuration
Transmit Configuration Update	New transmit configuration.
Transmit Status	Current transmit status.
Radio Configuration Identifier	Current radio configuration reference.
Radio Status	current radio status.
DSP Identifier	Current DSP Object reference.
FDD Direction	Current RX/TX direction.
ARQ State	Current ARQ status.
TX Power State	Current TX Power level state.
PECCx Event Counter	Most recent PECC's Event Counter values
Next Slot Relative Count	Relative number of slots to the next slot.
Next Slot Reference	Pointer to the Slot Structure of the next slot.

In one aspect, the slot manager 211 comprises a service module for the OTA driver 210 and slot-related ISRs that can be called upon demand as needed. However, the slot manager 211 may in some cases be subject to specific timing requirements imposed, for example, by the interrupt processing functions (e.g., the ISRs 224, 225, etc.). Timing and sequencing of various operations may, in particular, require consideration in relation to signalling traffic processing and voice traffic processing. Such timing and sequencing considerations may be explained with reference to Figs. 8, 9, 10 and 11, of which Figs. 8 and 9 relate to signalling traffic processing, while Figs. 10 and 11 relate to voice traffic processing.

Figure 8 depicts a signalling traffic path in accordance with one embodiment disclosed herein, while Fig. 9 illustrates data paths and sequencing for an active signalling

traffic slot. Although Fig. 8 depicts a microphone 827, speaker 828, CODEC 826 and DSP/vocoder 825, in a preferred embodiment these components are not utilized as part of the signalling traffic path. Rather, an SRAM transmit buffer 810, SRAM receive buffer 860, control ASIC 805, and digital radio ASIC (DRA) 820 shown in Fig. 8 are components utilized in the signalling traffic path. The SRAM transmit buffer 810 comprises storage area for a user station header 813, D channel information 812, bearer data 811, and DSP/vocoder status information 814. The SRAM receive buffer 860 comprises storage area for a base station header 863, D channel information 862, bearer data 861, and DSP/vocoder control information 864.

For a transmit sequence for signalling traffic from the user station 301, the following steps occur, as illustrated in the timing/sequence relationship shown in Fig. 9:

1. During a user transmit portion 902 of a time slot (for example, time slot 1 shown in Fig. 4 with respect to user station M9), the user station header 813, D channel information 812, and bearer data 811 are transferred from the SRAM transmit buffer 810 to the DRA 820 via a peripheral event controller (PEC) channel associated with the radio transmit DMA data transfer task 235, using a peripheral event controller counter to control the number of bytes transmitted.

2. Upon the above information being transmitted, and at or before the end of the user transmit portion 902, an end-of-transfer interrupt associated with the radio transmit DMA data transfer task 235 is generated.

3. The radio transmit DMA ISR invokes the end-of-transmit (EOTX) ISR 227, which issues a software trap and thereby invokes the software trap ISR 225.

4. The software trap ISR 225 invokes the slot manager 211 for "transmit frame complete" processing.

5. Radio receive frequency data is transferred from the SRAM 821 to the DRA 820 via a PEC channel associated with the radio configuration DMA data transfer task 223.

6. When the radio configuration data has been transferred, an end-of-transfer interrupt occurs associated with the radio configuration DMA data transfer task 223.

The radio transceiver 305 then re-adjusts its frequency synthesizer to a reception frequency.

For a receive sequence whereby signalling traffic is sent to the user station 301, the following steps occur, also as illustrated in the timing/sequence relationship shown in Fig. 9:

1. During a base transmit portion 952 of a time slot (for example, time slot 9 shown in Fig. 4 with respect to the base station), the base station header 863, D channel

information 862, and bearer data 861 are transferred from the DRA 820 to an SRAM receive buffer 860 via PEC channel associated with the radio receive DMA data transfer task 238, using a peripheral event controller counter to control the number of bytes transferred.

2. Compare data for the receive frame (i.e., the results of the command/mask comparison operations) is transferred from the SRAM command buffer 1704 to the control ASIC 805 via a PEC channel associated with the RX compare DMA data transfer task 237.

3. Upon the above information being transmitted, and at or before the end of the base transmit portion 952, an end-of-transfer interrupt associated with the RX compare DMA data transfer task 237 is generated.

4. The control ASIC 805 triggers a "receive compare complete" interrupt.

5. The MSCA ISR 224 invokes the slot manager 211 to perform "receive compare complete" processing.

6. An end-of-transfer interrupt associated with the radio receive DMA data transfer task 238 is generated.

7. The control ASIC 805 triggers a "receive frame complete" interrupt.

8. The MSCA ISR 224 invokes the slot manager 211 to perform "receive frame complete" processing.

9. Radio transmit frequency data is transferred from the SRAM 821 to the DRA 820 via a PEC channel associated with the radio configuration DMA data transfer task 223.

10. When the radio configuration data has been transferred, an end-of-transfer interrupt occurs associated with the radio configuration DMA data transfer task 223.

Figure 10 depicts a voice traffic path in accordance with one embodiment disclosed herein, while Fig. 11 illustrates data paths and sequencing for an active voice traffic slot. Figure 10 depicts a microphone 1027 and speaker 1028 connected to a CODEC 1026, which is connected to a DSP/vocoder 1025. As further illustrated in Fig. 10, the DSP/vocoder 1025 is connected to a control ASIC 1005, which is connected to an SRAM 1021 (comprising an SRAM transmit buffer 1010 and an SRAM receive buffer 1060) and digital radio ASIC (DRA) 1020. Similar to Fig. 8, the SRAM transmit buffer 1010 comprises storage area for a user station header 1013, D channel information 1012, bearer data 1011, and DSP/vocoder status information 1014. The SRAM receive buffer 1060 comprises storage area for a base station header 1063, D channel information 1062, bearer data 1061, and DSP/vocoder control information 1064.

For a transmit sequence for voice traffic from the user station 301, the following steps occur, as illustrated in the timing/sequence relationship shown in Fig. 11:

1. During a user transmit portion 1102 of a time slot (for example, time slot 1 shown in Fig. 4 with respect to user station M9), the user station header 1013, D channel information 1012, and bearer data 1011 are transferred from the SRAM transmit buffer 1010 to the DRA 1020 via a PEC channel associated with the radio transmit DMA data transfer task 235, using a peripheral event controller counter to control the number of bytes transmitted.

2. Upon the above information being transmitted, and at or before the end of the user transmit portion 1102, an end-of-transfer interrupt associated with the radio transmit DMA data transfer task 235 is generated.

3. The radio transmit DMA ISR invokes the end-of-transmit (EOTX) ISR 227, which issues a software trap and thereby invokes the software trap ISR 225.

4. The software trap ISR 225 invokes the slot manager 211 for "transmit frame complete" processing.

5. Radio receive frequency data is transferred from the SRAM 1021 to the DRA 1020 via a PEC channel associated with the radio configuration DMA data transfer task 223.

6. When the radio configuration data has been transferred, an end-of-transfer interrupt occurs associated with the radio configuration DMA data transfer task 223.

The above steps for transmitting voice information are essentially the same steps previously described as carried out for transmitting signalling information from the user station 301.

For a receive sequence whereby voice traffic is sent to the user station 301, the following steps occur, also as illustrated in the timing/sequence relationship shown in Fig. 11:

1. Prior to the base transmit portion 1152 of a time slot (for example, time slot 9 shown in Fig. 4 with respect to the base station), the DSP/vocoder status information 1014 and bearer data 1011 are transferred from the DSP/vocoder 1025 to the SRAM transmit buffer 1010 via a PEC channel associated with the DSP output DMA data transfer task 239.

2. Next, DSP control data 1064 is transferred from the SRAM receive buffer 1060 to the DSP/vocoder 1025 via a PEC channel associated with the DSP input DMA data transfer task 236.

3. Upon the above information being transmitted, and at or before the start of the base transmit portion 1152, an end-of-transfer interrupt associated with the DSP input DMA data transfer task 236 is generated.

4. During the base transmit portion 1152, the base station header 1063, D channel information 1062, and bearer data 1061 are transferred from the DRA 1020 to an SRAM receive buffer 1060 via a PEC channel associated with the radio receive DMA data transfer task 238, using a peripheral event controller counter to control the number of bytes transferred.

5. Compare data for the receive frame (i.e., the results of the command/mask comparison operations) is transferred from the SRAM command buffer 1704 to the control ASIC 1005 via a PEC channel associated with the RX compare DMA data transfer task 237.

6. Upon the above information being transmitted, and at or before the end of the base transmit portion 1152, an end-of-transfer interrupt associated with the RX compare DMA data transfer task 237 is generated.

7. The control ASIC 1005 triggers a "receive compare complete" interrupt.

8. The MSCA ISR 224 invokes the slot manager 211 to perform "receive compare complete" processing.

9. An end-of-transfer interrupt associated with the DSP output DMA data transfer task 239 is generated.

10. An end-of-transfer interrupt associated with the radio receive DMA data transfer task 238 is generated.

11. The control ASIC 1005 triggers a "receive frame complete" interrupt.

12. The MSCA ISR 224 invokes the slot manager 211 to perform "receive frame complete" processing.

13. Radio transmit frequency data is transferred from the SRAM 1021 to the DRA 1020 via a PEC channel associated with the radio configuration DMA data transfer task 223.

14. When the radio configuration data has been transferred, an end-of-transfer interrupt occurs associated with the radio configuration DMA data transfer task 223.

Steps 4 through 14 above for receiving voice information are essentially the same steps as carried out for receiving signalling information from the user station 301.

In addition to assisting with the transmission and reception of signalling information and voice/data information, the slot manager 211 also provides access control to slot objects 231. More specifically, in a preferred embodiment, the slot manager 211 implements a semaphore that controls exclusive access to the slot objects 231. In addition

to providing exclusive access between the OTA driver 210 and ISR processing, the semaphore ensures that the configuration of a slot object 231 will not change after it has been committed (i.e., after the hardware has been configured). The semaphore is implemented through a timer 229 having a suitable resolution, such as, e.g., one microsecond resolution. The timer 229 is preferably implemented as a one-shot timer.

In operation, the semaphore is cleared by the slot manager 211 at slot-synchronous times (i.e., at transitions between time slots 1202) to allow access to slot objects 231 for a time period defined by the value loaded in timer 229, and the semaphore is set upon timer expiration to disallow access to slot objects 231. This operation is illustrated in Fig. 12, which shows a time frame 1201 divided into a series of time slots 1202 (for simplicity, Fig. 12 does not show time slots on different base and user frequency bands). As shown in Fig. 12, the semaphore starts out as set during a time period 1220. The slot manager 211 clears the semaphore at a point in time 1205, and the timer 229 (after being loaded with an appropriate count value reflecting the desired access time) is started. Access to the slot object 231 is then available for a time period 1221 indicated in Fig. 12, until the timer 229 times out at a later point in time 1206. The semaphore then remains set for a time period 1222, during which access to the slot object 231 is not allowed. At a next point in time 1210, the slot manager 211 clears the semaphore, thereby allowing access to the slot object 231. Access is allowed during the time period 1223, until a point in time 1211 when the timer 229 times out. The semaphore then returns to its set state. Preferably, as shown in Fig. 12, each setting or clearing of the semaphore is carried out on a transition between time slots 1202 (i.e., slot-synchronously).

As previously indicated, the slot manager 211 preferably functions as a service module to the OTA driver 210 and can be invoked to initialize, create, update, query, or delete a slot object. The slot manager 211 processes individually each slot request received from the OTA driver 210, and queues an associated response (e.g., success, fail, or incomplete) in the slot output message queue 213. Further details about these operations, as illustrated by a preferred embodiment, are explained below.

With regard to initialization of slot objects 231, the slot manager 211 performs several steps. First, the slot manager 211 installs the timer ISR 226. Next, the slot manager initializes the slot objects 231, the slot message input queue 212, the slot message output queue 213, and the DSP objects 216. Finally, the peripheral event controller (PEC) interrupts relating to slot object DMA functions are enabled by the slot manager 211.

To create a slot object 231, the slot manager 211 first attempts to allocate a slot object 231. If the allocation attempt is unsuccessful, a failure status indicator is returned; otherwise the slot object is initialized and configured as specified by the request. Then, if

access to slot objects 231 is not allowed (by reason of the semaphore), an incomplete status indicator is returned. Otherwise, the slot object 231 is inserted into a slot list. If there are no other active slots, then the radio services module 217 is invoked to commit any outstanding updates and to complete radio initialization. The slot manager 211 then performs "receive setup" processing, as described below. Finally, slot-related interrupts generated by or through the control ASIC 205 are enabled. Thereafter, the MSCA ISR 224 will be triggered by a general status interrupt upon the first successful receive. The slot manager 211 then returns a success status indicator.

To delete a slot object 231, the slot manager 211 first determines whether access to the slot objects 231 is allowed (as dictated by the semaphore). If access is not allowed, an incomplete status indicator is returned. Otherwise, the slot object 231 is removed from the slot list. If there are no other active slots, then slot-related interrupts generated by or through the control ASIC 205 are disabled, and the radio services module 217 is invoked to reset the radio transceiver 305 to its idle state. The receive state machine in the control ASIC 205 is reset. The slot object 231 which has been removed from the slot list is then de-allocated, and a success status indicator is returned.

To update a slot object 231, the slot manager 211 first saves the specified update configuration. If access to slot objects 231 is not allowed, an incomplete status indicator is returned. Otherwise, the configuration updates are "committed" by switching the active configuration for the slot object 231 to the saved update configuration. If the slot object 231 being updated is currently active, then the slot manager 211 performs "receive setup" or "transmit setup" processing as appropriate, and as described in more detail below. Finally, a success status indicator is returned.

To query a slot object 231, the slot manager 211 first determines whether access to the slot objects 231 is allowed (as dictated by the semaphore). If access is not allowed, an incomplete status indicator is returned. Otherwise, the slot manager 211 accesses the slot object's current configuration and current status. The slot manager 211 then returns a success status indicator.

In addition to manipulating slot objects 231, the slot manager 211 is also involved in interrupt processing. For example, the slot manager 211 performs processing when a receive compare complete interrupt, a receive frame complete interrupt, or a transmit frame complete interrupt occurs.

When a receive compare complete interrupt occurs, the slot manager 211 stops the one-shot timer 229. It then checks the Head Check Field (HCF) Status from a Receive Status Register of the control ASIC 205. If the HCF status is "good", then the slot manager extracts the OTA header fields, including Packet Type, Channel Utilization, Next

Slot Pointer, ARQ, and Power. On the other hand, if the HCF status is "bad", the OTA header fields are set to default values, as follows: Packet Type is set to unknown; Channel Utilization is set to unknown; Next Slot Pointer is set to indicate the "same slot next frame"; ARQ is set to indicate no change; and Power is set to indicate no change. The slot manager 211 then updates the "receive status" for the current slot, including HCF status, Packet Type, Channel Utilization, and RX Compare Status (which is extracted from the General Status Register of the control ASIC 205). Finally, the slot manager 211 loads the Slot Map Pointer Register of the control ASIC 205 to select the next time slot in which to receive.

When a transmit frame complete interrupt occurs, the slot manager 211 sets up to receive or transmit, as appropriate, for the next slot object 231 as further detailed herein.

When a receive frame complete interrupt occurs, the slot manager 211 first examines the Packet Frame Check Word (FCW) Status from the Receive Status Register of the control ASIC 205. If the FCW Status is "good", the slot manager 211 then determines whether an attempt is being made to synchronize to a time slot and, if so, whether synchronization has been achieved. If it has, the slot manager 211 clears radio synchronization control. The slot manager 211 then checks the receive Compare Status and, if "good", resets the slot object compare control (i.e., the command/mask templating operation) to compare only with the OTA Header field, with a "don't care" Packet type. The slot manager 211 then invokes the ARQ service module 220 to perform ARQ processing, and sets the receive status of the message. The slot manager 211 then determines the transmit status for the message. If the ARQ passed, the message transmit status is "good", and the slot manager 211 switches to a default transmit message. If, however, ARQ failed, and if a predefined number of retries have been exhausted, the message transmit status is changed to "bad". The slot manager 211 then switches to a default transmit message. If ARQ failed but the predefined number of retries has not yet been exhausted, the message transmit status is changed to (or kept at) "unknown". The current transmit message is kept for a subsequent re-transmission.

The slot manager 211 continues interrupt processing for the receive compare complete interrupt, by invoking the power control module 221 to determine the transmit power level for the next transmission on the current time slot. The slot manager 211 then reads the RSSI values from the RSSI registers of the control ASIC 205. It allocates a new receive buffer in the SRAM 1524 for the next receive frame on the same time slot. Next, the slot manager 211 processes messages in the slot input message queue 212, as further detailed herein. If the slot is in a voice traffic mode, the slot manager 211 copies the DSP/vocoder status data to the associated DSP object 216. As the final step performed in

response to a receive frame complete interrupt, the slot manager 211 sets up to receive or transmit, as appropriate, for the next slot object 231 as further detailed herein.

For receive setup processing, the slot manager 211 first determines whether there are any pending radio configuration updates and, if so, invokes the radio services module 217 to commit the updates. The slot manager 211 also invokes the radio services module 217 to configure the radio receive frequency. Next, the slot manager 211 enables the radio receive DMA path, and disables the radio transmit functionality. The slot manager 211 then sets radio synchronization control as follows. If the user station 301 is in the process of acquiring a time slot (i.e., synchronizing), the slot manager 211 sets radio synchronization control to synchronize upon the first "good" (i.e., error-free) qualified receive slot. If the user station 301 is in the process of synchronizing with the digital signal processor (DSP) (e.g., DSP/vocoder 1510 shown in Fig. 15), the slot manager 211 sets radio synchronization control to generate a DSP synchronization pulse for the current time slot, and then enables the DSP (e.g., DSP/vocoder 1510) through a General Purpose Register in the control ASIC 205 or 1591, for example. If neither of these synchronization activities are in process, the slot manager 211 clears radio synchronization control.

As the next step in receive setup processing, the slot manager 211 sets up the receive data paths for the time slot. In this regard, the slot manager 211 sets up and enables a peripheral event controller (PEC) channel associated with the receive compare DMA data transfer task 237, thereby allowing direct memory access (DMA) operations for transfer of compare data from memory (e.g., from SRAM command buffer 1704 shown in Fig. 17) to a Command Compare Register in the control ASIC 205. The slot manager 211 sets up and enables a PEC channel associated with the radio receive DMA data transfer task 238, thereby enabling DMA operations for the transfer of data from a Receive Source Register in the control ASIC 205 to receive data memory (e.g., SRAM receive buffer 1706). Next, if the time slot is to be used for voice traffic, the slot manager 211 copies DSP/vocoder control data from the associated DSP object 216 to DSP control memory (e.g., vocoder control field 1805 shown in Fig. 18). The slot manager 211 then sets up and enables a PEC channel associated with the DSP input DMA data transfer task 236, thereby enabling DMA operations for the transfer of data from memory (e.g., SRAM receive buffer 1706) to a DSP Data/Control Register in the control ASIC 205.

As the next step in receive setup processing, the slot manager 211 sets up a pre-transmit data path for the time slot. In this regard, the slot manager 211 sets up and enables the PEC channel associated with the DSP output DMA data transfer task 239, thereby enabling DMA operations for transferring data from a DSP Source Register in the control ASIC 205 to memory (e.g., SRAM transmit buffer 1707). The slot manager 211

sets a Data Notice Select in the control ASIC 205 based upon the traffic mode (i.e., signalling or voice) of the time slot. Next, the slot manager 211 starts the one-shot timer 229. If the user station 301 is attempting to acquire synchronization for this time slot, it is possible that an interrupt may not occur; therefore, as a fail-safe mechanism, the timer ISR 226 is configured to process queued messages upon expiration of the timer 229. On the other hand, if the user station 302 is already synchronized to the current time slot, then interrupts should occur; in such a case, the timer ISR 226 is configured by the slot manager 211 to set the slot object access semaphore upon expiration, thereby blocking access to slob objects 231 when the timer 226 times out. Finally, the slot manager 211 loads the Compare Count register in the control ASIC 205 to set up and initiate the command/mask templating operation upon reception of the receive frame.

For transmit setup processing, the slot manager 211 first determines whether there are any pending radio configuration updates and, if so, invokes the radio services module 217 to commit the updates. The slot manager 211 invokes the radio services module 217 to configure the radio transmit frequency. Next, the slot manager 211 disables the radio receive DMA path, enables the radio transmit functionality, and clears radio synchronization control. The slot manager 211 then sets up the OTA header (e.g., user station header 813 in Fig. 8, or user station header 1013 in Fig. 10, or user station header 1808 in Fig. 18, depending on whether bearer traffic message(s) or signalling traffic message(s) will be sent). For example, the slot manager 211 may set up the Packet Type field as specified in the transmit configuration data, or the ARQ field as determined by the ARQ service module 220. Next, the slot manager 211 sets up the transmit data path for the time slot. The slot manager 211 sets up and enables the PEC channel associated with the radio transmit DMA data transfer task 235, thereby enabling DMA operations to transfer data from transmit data memory (e.g., SRAM transmit buffer 1707) to a Transmit Destination Register in the control ASIC 205. The slot manager 211 starts the one-shot timer 229 and configures the timer ISR 226 to set the slot object access semaphore upon expiration, thereby blocking access to slot objects 231 when the one-shot timer 229 times out. Finally, the slot manager 211 invoke the radio services module 217 to load the desired transmit power level.

Upon expiration of the one-shot timer 229, the slot manager 211 performs certain processing operations. The slot manager 211 processes messages queued in the slot message input queue 212, and then configures the timer ISR 226 to process queued messages upon time-out by the one-shot timer 229.

Table 3-4 below summarizes the major outputs of the slot manager 211 and their destinations, along with comments.

Table 3-4

Item	Destination	Comment
Slot Request Response	Any	Initialize, create, update, query, delete, and interrupt processing request status.
Slot Message Output Queue Message	OTA Driver	Queued slot message.
Radio Request	Radio services.	Update and RSSI request.
DSP Request	DSP Object Manager	Initialize and update request.
ARQ Request	ARQ	ARQ status and message sequence number request.
Power Control Request	Power Control	Power control request.

Further details will now be provided concerning the operation of various other system components of the software control system 201 shown in Fig. 2, starting with the MSCA ISR 224. The MSCA ISR 224 handles the general status interrupts that are generated by the control ASIC 205. Non-slot related interrupts are sensed, then signalled externally through an external event interface. Non-slot related interrupts may include, for example, SIM TX error (i.e., SIM transmit port error), Mode Change (i.e., a port connector change has occurred or is sensed), and a Keypad Event (i.e., one or more keypad keys was pressed and released). Slot-related interrupts are handled by invoking the associated function of the slot manager 211. Slot-related interrupts may include, for example, RX Compare Complete (i.e., the receive command/mask templating operation has been completed for a slot) and RX Frame Complete (i.e., the receive portion of a slot has been completed). Slot interrupts are generated when the user station 301 receives data on an active time slot. The MSCA ISR 224 is preferably capable of drawing upon ISR services of the real-time operating system, if needed for its functionality.

When invoked, the MSCA ISR 224 reads the MSCA General Status Register (GSR) to determine the cause of the interrupt. The MSCA ISR 224 then proceeds to service the interrupt, based upon its source. For example, for an RX Compare Complete interrupt, the MSCA ISR 224 may invoke the slot manager 211 to perform RX compare complete processing. For an RX Frame Complete interrupt, the MSCA ISR 224 may

invoke the slot manager 211 to perform RX frame complete processing, and then signal a slot event.

The MSCA ISR 224 executes upon each General Status interrupt received from the control ASIC 205. The control ASIC 205 is preferably given the highest possible interrupt priority that is lower than the priority of the RTOS system timer interrupt. However, the highest priorities are preferably reserved for the peripheral event controller (PEC) interrupts; therefore, it is possible for the MSCA ISR 224 to be interrupted by a PEC interrupt.

With regard to timing and sequencing of interrupts, non-slot related interrupts handled by the MSCA ISR 224 are asynchronous and aperiodic. On the other hand, slot-related interrupts are synchronous with the time slots, as may be explained by way of example with reference to Fig. 13. There, a time slot 1302 is divided into a user station transmit frame 1303 and a base station transmit frame 1304. The user station transmit frame 1303 and base station transmit frame 1305 are separated by a guard time and turn-around gap (collectively denoted by gap 1304 in Fig. 13). The user station transmit frame 1303 commences after a variable radio gap 1310 occurring at the starting point 1321 of the time slot 1302. A second turn-around gap 1311 occurs between the end of the base station transmit frame 1305 and the ending point 1322 of the time slot 1302 (i.e., the start of the next time slot 1352).

In the example shown in Fig. 13, during a first time period 1361 an RX Compare Complete interrupt occurs, as described elsewhere herein. The length of the first time period 1361 reflects processing of the header in the base station message and therefore depends in part upon the size of the header in the base station message. The RX Compare Complete interrupt during the period 1361 invokes the MSCA ISR 224. During a second time period 1362, interrupt processing responsive to the RX Compare Complete interrupt occurs, and is preferably completed before the end of the base station transmit frame 1305. After the end of the base station transmit frame 1305, preferably a fixed time period thereafter, an RX Frame Complete interrupt occurs at point 1363 in Fig. 13. Processing in response to the RX Frame Complete interrupt occurs over time period 1364, and is preferably complete before the next time slot 1352. However, if the next time slot 1352 is not in use, processing may continue up until the end of the next time slot 1352.

Table 4-1 below summarizes the major outputs of the MSCA 224 and their destinations, along with comments.

Table 4-1

Item	Destination	Comment
Slot Requests	Slot Manager	Interrupt processing request.
Slot Event	OTA Driver	Signalled upon ISR slot processing completion
Keypad Event	External	Signalled upon key press or release.
Mode Change Event	External	Signalled upon port connector change.
SIM TX Error Event	External	Signalled upon SIM port transmit error.

The software trap ISR 225 handles software interrupts (traps) generated by the software control system 201. The software trap ISR 225 invokes the appropriate functions of the slot manager 211 functions based upon the source of the interrupt. The software trap ISR 225 is preferably a "cooperative" ISR in the sense that it is free to use real-time operating system services for interrupt handling. One purpose of the software trap ISR 225 is to promote non-cooperative ISRs to cooperative ISRs.

A software trap may be triggered by any of a variety of events, including TX frame complete or one shot timer expiration. It may be invoked, for example, by the end-of-transmit (EOTX) ISR 227 or the timer ISR 226.

The end-of-transmit (EOTX) ISR 227 handles the interrupt generated upon the completion of the user station transmit frame within a time slot. The EOTX ISR 227 may be invoked by a radio transmit DMA peripheral event controller ISR. The EOTX ISR 227 sets a flag and generates a software trap to perform end-of-transmit processing. The radio transmit DMA PEC ISR invokes the EOTX ISR 227 upon completion of DMA transfer of transmit data from memory to the radio. The EOTX ISR 227 may, in some embodiments, be non-cooperative ISR and therefore unable to use real-time operating system (RTOS) services. However, the EOTX ISR 227 may invoke the software trap ISR 225 to allow access to such services.

The timer ISR 226 handles the interrupt generated upon expiration of the one-shot timer 229. The timer ISR 226 may update a semaphore or set a flag and then generates a software trap to transfer control over to the software trap ISR 225 to perform timer expiration processing. The timer ISR 226 may, in some embodiments, be a non-cooperative ISR and therefore unable to use real-time operating system services without the assistance of the software trap ISR 225.

The peripheral event controller (PEC) ISRs handle interrupts generated upon completion of PEC-controlled DMA transfers. The PEC ISRs are triggered upon completion of DMA I/O requested by the control ASIC 205 through the associated PEC channel. Typically, upon being invoked, a PEC ISR updates a counter or otherwise calls a function, as illustrated by examples explained earlier herein. Table 5-1 below summarizes the major inputs to the PEC ISRs and their sources, along with comments.

Table 5-1

Item	Source	Comment
EX01N Interrupt	MSCA	Signalled upon completion of transfer from memory to the Transmit Destination Register.
EX11N Interrupt	MSCA	Signalled upon completion of transfer from the Receive Source Register to memory.
EX21N Interrupt	MSCA	Signalled upon completion of transfer from the DSP Source Register to memory.
EX41N Interrupt	MSCA	When in data traffic mode: Signalled upon completion of transfer from memory to the Unencoded Transmit Data Register.
EX51N Interrupt	MSCA	Signalled upon completion of transfer from memory to the Compare Input Register.
EX61N Interrupt	MSCA	Signalled upon completion of transfer from memory to the DSP Data/Control Byte Register.
EX71N Interrupt	MSCA	When configuring the radio: Signalled upon completion of transfer from memory to the Radio Configuration/Slot Map Data Register. When in data traffic mode: Signalled upon completion of transfer from the Unencoded Receive Data Register to memory.

In one embodiment, the PEC ISRs are "non-cooperative" in the sense of not interfacing with the real-time operating system. The system processor (e.g., micro-

controller 1505 shown in Fig. 15) preferably reserves the highest priorities and group numbers for the PEC interrupts. Simultaneous requests for PEC channels are prioritized according to the PEC channel number, where channel 0 has the lowest priority and channel 7 has the highest priority. Table 5-2 below shows PEC interrupt assignments in one embodiment.

Table 5-2

PEC Channel (PECCx)	Interrupt
PECC7 - Radio Configuration DMA	EX71N
PECC6 - RX Compare DMA	EX51N
PECC5 - DSP Output DMA	EX21N
PECC4 - Radio RX DMA	EX11N
PECC3 - DSP Input DMA	EX61N
PECC2 - Radio TX DMA	EX01N
PECC1 - Not presently used	EX41N

At the end of a data transfer for which a PEC channel has been set up, the associated PEC ISR is invoked to performs end-of-PEC transfer processing. If the interrupt action specifies an event counter to update, the PEC ISR increments the event counter. If the interrupt action specifies a function to call, the PEC ISR calls the specified function. In a presently preferred embodiment, only the PEC ISR associated with the radio transmit DMA transfer task 235 specifies a function to call.

The PEC event counters may be monitored by external (i.e., non-PEC ISR) software processing components. These external software processing components may clear (i.e., zero) a given PEC event counter, or else let it increment until it rolls over.

Table 5-3 below summarizes the major outputs of the PEC ISRs and their destinations, along with comments.

Table 5-3

Item	Destination	Comment
PECCx Event Counters (1 through 7)	MSCA ISR	End-of-PEC interrupt counters.
Interrupt	End-of-TX ISR	TX Frame Complete notification

In one embodiment, the radio services module 217 provides a common interface to the digital radio ASIC (DRA). The radio services module 217 preferably provides functions to configure receive and transmit frequencies, codes, and preambles, to control

transmit power, to synchronize the radio to time slots, to read RSSI values, and to configure various thresholds. To communicate with the DRA, the radio services module 217 may utilize registers in the control ASIC 205, and DMA I/O operations triggered by the control ASIC 205 through, for example, the radio configuration DMA data transfer task 223.

Table 6-1 below summarizes the major data elements utilized by the radio services module 217, along with comments.

Table 6-1

Element	Comment
Radio Configuration	Current radio configuration.
Radio Configuration Update	New radio configuration.
Synthesizer 1 Frequency Table	Hardware specific values loaded to generate receive and transmit frequencies.
Synthesizer 2 Frequency Table	Hardware specific values loaded to generate specific frequencies.
Receive Code Table	CDMA receive codes.
Receive Preamble Table	Receive preambles.
Transmit Code Table	CDMA transmit codes.
Transmit Preamble Table	Transmit preambles.

The radio services module 217 preferably provides the following operations for configuring or otherwise manipulating radio functionality:

1. Initialize (resets and initializes the radio transceiver 305 to an idle state)
2. Reset (resets the radio to its idle state)
3. Configuration Update Control
    a. Register Update (saves a new configuration set for future loading)
    b. Commit Update (switches to and loads a new configuration set)
4. Receive Enable Control (enables or disables the radio-to-memory data path)
5. Transmit Enable Control (enables or disables transmit for the current receive slot)
6. Synchronization Control
    a. Acquire Slot (clears synchronization and searches for the next good receive slot by, e.g., finding the next receive preamble).

b. Synchronize Slot (locks onto the next good receive slot)

c. Synchronize DSP (issues DSP synchronization pulse on a slot)

7. Timing Chain Control (selects radio timing chain)

8. Slot Selection Control (selects the next receive slot, relative to the current receive slot; referred to as the "next slot pointer")

9. Transmit Power Control

a. Control Power Level (increases or decreases power level control)

b. Write Power Level (loads power level control)

10. Load CDMA Configuration (loads the DRA with the currently selected configuration set parameters, including receive frequency, receive code, receive preamble, transmit frequency, transmit code, and transmit preamble)

11. Load Static Configuration (loads the DRA with "static" parameters, including detect/track threshold, ratchet notch threshold, and synchronization offset)

12. Read RSSI (reads RSSI average and difference values associated with the current receive slot)

The DSP object manager 215 defines and coordinates updates to the DSP objects 216. The DSP object manager 215 preferably treats the DSP/vocoder at an "object" level. In other words, a DSP object 216 comprises, in one aspect, a block of data with an associated identifier, while the "external" DSP driver 251 actually updates the bit-level controls and interprets the bit-level status flags of the DSP/vocoder. The DSP object manager 215 invokes the DSP driver 251 to create, initialize, and destroy DSP objects 216, and accepts updates to the control and status blocks of DSP objects 216.

The DSP object manager 215 is executed upon demand and preferably has no knowledge of timing requirements. The DSP objects 216 which are managed serve as a central synchronization point for the DSP driver 251 (which updates DSP/vocoder control information asynchronously upon demand) and the slot manager 211 (which updates DSP/vocoder status information after the receive frame of an active bearer traffic slot).

The ARQ service module 220 of the software control system 201 provides error detection, error recovery, error notification, and message sequencing based upon the ARQ mechanism implemented in the bearer packets. The ARQ service module 220 is invoked after the receive frame of a time slot but before the transmit frame of the associated time slot, so that a request to re-send the last received packet can be sent to the base station before the next packet is sent. The ARQ service module 220 may functionally separated into two state machines, one for receive ARQ and one for transmit ARQ. Figure 14 is a diagram of a state table 1401 illustrating the functional behavior of the ARQ service module 220. In Fig. 14, "Msg CRC Status" represents the error status as derived from a

cyclic redundancy check (CRC); "RX Msg#" indicates whether or not the base station message's "TX Msg#" matched the expected value (i.e., whether the received message both is and was expected to be either the "same" message or "next" message); "RX ACK" indicates whether an acknowledge (ACK) or non-acknowledge ("NAK") message was received from the base station 104; "Special Msg?" indicates whether the message is of a type which will not cause the RX Msg# to be promoted; "TX ACK" indicates whether an ACK or NAK message will be returned to the base station 104; "TX Msg#" indicates whether the user station 301 will transmit the same data packet as previously sent or a new data packet; and "Next Expected RX Msg#" indicates whether the next received frame from the base station 104 should constitute the same data packet as previously received or a new data packet.

In a preferred embodiment, the receive ARQ state machine in the ARQ service module 220 is generally responsible for filtering duplicate messages from the base station 104. The base station 104 may, for example, send duplicate messages when it has not received an ACK from the user station 301 to its previously transmitted message. Duplicate messages are filtered by the receive ARQ state machine by examining the "RX Msg#" bit of the message (i.e., data packet) received from the base station 104. If the "RX Msg#" bit matches the "Expected Msg#", then the message is considered valid and is processed; otherwise, the message is considered a duplicate message and is discarded.

In the case of certain special messages (e.g., CT-HLD (control traffic hold), and CT-SPO (control traffic specific poll) during lost link recovery) received from the base station 104, the "RX Msg#" is not promoted at the user station 301. The user station 301 processes these special messages even though the "RX Msg#" may not match the "Expected Msg#." In addition, the user station 301 will accept any value for the "RX Msg#" when initially being paged. After initially responding to the general poll control traffic message (CT-GPO) from the base station 104, the user station 301 will begin keeping track of the "RX Msg#" from the base station 104 on an ongoing basis.

The receive ARQ state machine of the ARQ service module 220 is also responsible for promoting the user station's "TX Msg#", based on the ACK/NAK value of the ARQ field in the data packet received from the base station 104. When the base station 104 acknowledges receipt of a user station message by setting the "ACK" bit in the ARQ field of a message to the user station 301, the user station 301 promotes the "TX Msg#" (if it has new data to transmit) and then transmit the new data and promoted "TX Msg#" in the next message to the base station 104. If the user station has received an "ACK" in response to its last transmitted message but has no new data to send, the user station 301 will transmit a control traffic hold message (CT-HLD) without promoting its "TX Msg#".

The transmit ARQ state machine of the ARQ service module 220 is responsible for informing the base station 104 whether or not a particular message was received successfully at the user station 301, by sending an "ACK" or "NAK as appropriate in the next message transmitted by the user station 301. The transmit ARQ state machine is also responsible for indicating to the higher layers (in ISO terminology) that a message transmitted by the user station 301 was or was not successfully received at the base station 104, as indicated by the ACK/NAK value in the ARQ field of a received message.

When receiving a message from the base station 104, the user station 301 may perform a cyclic redundancy check (CRC) to determine whether the message was received with errors. The CRC status of the message received at the user station 301 generally impacts both the transmit ARQ state machine and the receive ARQ state machine of the ARQ service module 220, as indicated, for example, by the state table shown in Fig. 14.

Messages received at the user station 301 have both "header CRC" and "frame CRC." If the header CRC is good, then, regardless of the frame CRC, the header is deemed to contain valid information that may be processed by the user station 301. For example, if the user station 301 receives a message with good header CRC but with a bad frame CRC, and with the ACK bit set, the user station 301 may subsequently transmit a new message because it knows that its last transmitted message has been successfully received at the base station 104 (by virtue of the ACK bit being set). The D and B channels of the received message, however, are ignored because the bad frame CRC indicating that the packet contains errors. Therefore, the user station 301 will transmit a "NAK" with regard to the received message, and will discard the received message.

The power control service module 221 of the software control system 201 controls the transmit power level of the radio transceiver 305 of the user station 301 based upon a power control mechanism implemented in the header of the data packets transmitted between the user station 301 and the base station 104. The power control service module 221 is preferably invoked after a receive frame of a time slot but before the next transmit frame of the associated time slot, so as to allow for closed loop power control of the user station 301 by the base station 104. In a presently preferred embodiment, the user station 301 adjusts its transmit power in response to a power control field (i.e., a set of bits) contained in the header of base station messages, as per the Table 7-1 below.

Table 7-1

Previous Power Bit	Current Power Bit	TX Power Level Recommendation
0	0	Decrease TX power
1	1	Increase TX power
0	1	Maintain current TX power
1	0	Maintain current TX power
Missing	0	Maintain current TX power
Missing	1	Increase TX power
0	Missing	Maintain current TX power
1	Missing	Increase TX power
Missing	Missing	Increase TX power

Preferably, there are at least two consecutive packets received before any action in adjusting the power level is taken by the user station 301, and the user station's power level is not adjusted upon reception of the initial packet. In one aspect, the user station's transmit power level is "soft-limited" in that, when at the highest or lowest transmit power level, an increase or decrease request, respectively, will have no affect upon the transmit power level.

Further details will now be provided about certain hardware aspects of the control section 1501 depicted in Fig. 15. In a preferred embodiment, the control section 1501, and specifically the control ASIC 1502 thereof, comprises registers having functions and access modes as given by the information appearing in Table 8-1 below.

Table 8-1

ADDRESS	REGISTER NAME	REGISTER CONTENTS	ACCESS MODE
0E01AH	SLOT MAP POINTER	SLOT MAP POINTER	WRITE
0E019H	LINEARIZER 1 & 2	LINEARIZER CONTROL & DATA	READ/WRITE
0E018H	TEST MODE	MS LOOP BACK	WRITE
0E017H	GENERAL STATUS	COMPARE COMPLETE, COMPARE ERROR, RX FRAME COMPLETE, RX ERROR, SIM TX ERROR, MODE CHANGE, KEY PRESS/RELEASE	READ
0E016H	INTERRUPT CONTROL	INTERRUPT @ END COMPARE, INTERRUPT @ END RX, INTERRUPT @ RC ROLLOVER	WRITE
0E015H	COMPARE INPUT	COMMAND COMPARE DATA, MASK	WRITE
0E014H	COMPARE COUNT	NUMBER OF COMMAND COMPARES	WRITE
0E013H	DSP DATA/CONTROL BYTES	DSP DATA/CONTROL BYTES	WRITE
0E012H	DSP SOURCE	DSP ENCODED DATA FOR TRANSMIT	READ
0E011H	RECEIVER SOURCE	RECEIVED DATA FROM OTA	READ
0E010H	TRANSMIT DESTINATION	TRANSMITTER AS DESTINATION	WRITE
0E00FH	SLEEP CONTROL	μC POWER DOWN, μC STANDBY, DSP SLEEP,	WRITE

ADDRESS	REGISTER NAME	REGISTER CONTENTS	ACCESS MODE
		CODEC SLEEP, RADIO ASIC SLEEP, LINEARIZER ASIC SLEEP, μC ALIVE	
0E00EH	RESET	RADIO ASIC, DSP, RC COUNTER, OTA LCA	WRITE/READ
0E00DH	RECEIVE STATUS	FRAME ERROR, HEADER ERROR, FREQUENCY ERROR, TRANSMITTER FAIL	READ
0E00CH	RSSI BYTE 1 & 2	RSSI DATA FIRST AND SECOND BYTE	READ
0E00BH	SYNC BYTE 1 & 2	DSP SYNC BYTE 1 & 2	WRITE
0E00AH	KEY	INTERNAL ENCRYPTION KEY	READ
0E009H	UNUSED		
0E008H	SIM TRANSMIT COUNT	SIM TRANSMIT BYTE COUNT	WRITE
0E007H	SIM CONTROL	SIM INVERT, PARITY MODE E/O, STOP CLK HIGH/LOW, TX/RX MSB FIRST, ENABLE CLOCK, ENABLE VCC, RESET LEVEL	WRITE
0E006H	SIM TRANSMIT	SIM TRANSMIT DATA/COMMAND	WRITE
0E006H	SIM RECEIVE	SIM RECEIVE DATA/COMMAND	READ
0E005H	LINEARIZER CONTROL	BUSY, AUTONOMOUS MODE ENABLE, RECEIVE MODE	READ/WRITE
0E004H	UNUSED		
0E003H	KEYRESP	KEYPAD PRESS ID (ROW 8-1)	READ
0E003H	KEYPOLL	KEYPAD COMMAND	WRITE

ADDRESS	REGISTER NAME	REGISTER CONTENTS	ACCESS MODE
		(COLUMN 4-1), ENABLE	
0E002H	RADIO CONFIGURATION/ SLOT MAP DATA	DESTINATION FOR RADIO CONFIG / SLOT MAP DATA	WRITE
0E001H	RADIO CONFIGURATION CODE	RADIO CONFIG CODE SETTING	WRITE
0E000H	GENERAL PURPOSE REGISTER	RESET SIM, RESET RX, TX ENABLE, RINGER ENABLE, DSP ENABLE	WRITE
0E000H	GENERAL PURPOSE REGISTER	HANDS FREE, CONFIG PINS	READ

In a preferred embodiment, the Slot Map Pointer Register (0E01AH) provides a target address for sending the slot map pointer data to the radio interface circuitry. When the microcontroller 1505 wants to indicate the next time slot for transmission by the radio, the microcontroller 1505 writes a new value to the Slot Map Pointer Register. The data is then read by the radio interface circuitry (including, e.g., the RIF ASIC 1514) to determine the next time slot to operate in. The Slot Map Pointer Register definition is as follows:

ADDRESS: 0E01AH

MSB LSB

7	6	5	4	3	2	1	0
N/A	N/A	MAP5	MAP4	MAP3	MAP2	MAP1	MAP0

Bits 5-0 MAPx - Slot pointer map value.

'Value'- Slot pointer map value. This will be a hex value indicating the number of slots to skip.

'00H' - Inactive (Initial condition)

Bits 7-6 Unused

The Linearizer Register (0E019H) acts as the control port to and from the linearizer ASIC 1513. When the microcontroller 1505 desires to send control commands or data to the linearizer ASIC 1513, it writes a byte to the Linearizer Register. In response, the linearizer interface circuitry in the OTA LCA 1502 transmits the command/data to the linearizer ASIC 1513. If the command issued is a read command, information stored in the Linearizer Register is read, but only bits 5-0 are considered valid. The register definition is as follows:

ADDRESS: 0E019H

MSB LSB

7	6	5	4	3	2	1	0
CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0

Bits 7-0 CMDx - Command or data to/from the linearizer ASIC

'Value'- Command/data value.

'00H' - Inactive (Initial condition)

The Test Register (0E018H) defines various test modes that the OTA LCA may be placed into by the controller. The register definition is as follows:

ADDRESS: 0E018H

MSB LSB

7	6	5	4	3	2	1	0
MS LOOP-BACK	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Bits 6-0 Unused

Bit 7 MS LOOPBACK - This bit activates or deactivates the MS loop back operation.

'1' - Handset is placed in loop back

'0' - No loop back (Initial condition).

The General Status Register (0E017H) contains bit specific status information relating to certain events that may cause the microcontroller 1505 to be interrupted. When the microcontroller 1505 receives a general status interrupt, it responds by reading the General Status Register to determine the reason for the interrupt. After the microcontroller 1505 reads the General Status Register, the OTA LCA 1502 clears the status bit(s) to ensure that the data present in the register always indicates the true reason for an interrupt. Initial condition is set to '0' for all bits. The register definition is as follows:

ADDRESS: 0E017H

MSB LSB

7	6	5	4	3	2	1	0
KEY PRS-RLS	MODE CHG	SIM TX ERROR	N/A	RX ERROR	RXCMPLT	COMPARE ERROR	CMPCMPLT

Bit 0 CMPCMPLT - Compare complete - Indicates compare sequence has completed (compare counter = 0).

'1' - Compare complete

'0' - Inactive

Bit 1 COMPARE ERROR - Indicates that the compare sequence did not match on all bytes that were compared.

'1' - Compare Error

'0' - Inactive

Bit 2 RXCMPLT - Receive complete - Indicates that the reception of data in the receive portion of a slot has completed.

'1' - Receive complete

'0' - Inactive

Bit 3 RX ERROR - Receive Error - Indicates that data received from radio was in error due to CRC over header or CRC over packet failure.

'1' - Receive error

'0' - Inactive

Bit 4 Unused

Bit 5 SIM TX ERROR - SIM port failed to get a successful transmission after 3 attempts.

'1' - SIM transmit error

'0' - Inactive

Bit 6 MODE CHG - A change of accessory attached to test port connector has been detected.

'1' - Change detected

'0' - Inactive

Bit 7 KEY PRS-RLS - Keypad pressed or released- When an interrupt occurs and this bit is set, it indicates that the keypad has been pressed. When the controller is interrupted a second time with this bit active, it indicates that the keypad has been released.

'1' - Keypad pressed or released

'0' - Inactive

The Interrupt Control Register (0E016H) defines when and if the OTA LCA 1502 will generate interrupts to the microcontroller 1505 upon specified conditions. Initial conditions are set to '0' for all bits. The register definition is as follows:

ADDRESS: 0E016H

MSB LSB

7	6	5	4	3	2	1	0
N/A	N/A	N/A	N/A	N/A	INTRRCRL	INTRXFRM	INTCMPC

Bit 0 INTCMPC - Interrupt on compare complete - Indicates to the OTA LCA that a general status interrupt should be issued at end of compare sequence (error or no error).

'1' - Interrupt on compare sequence complete

'0' - No interrupt on compare sequence complete

Bit 1 INTRXFRM - Interrupt on RX frame complete - Indicates to the OTA LCA that a general status interrupt should be issued at end of receive frame (good or bad receive).

'1' - Interrupt on end of receive frame

'0' - No interrupt on end of receive frame

Bit 2 INTRRCRL - Interrupt on RC circuit roll over - Indicates to the μC that the RC long timer counter had rolled over. This may be used for a long duration counter and allow the controller to be awakened periodically over long durations.

'1' - Interrupt at end of time

'0' - No interrupt at end of time

Bits 7-3 Unused

The Command Compare Input Register (0E015H) causes a command template and mask pair to perform a compare operation. This register provides the target address for writing the command template and mask to the compare circuitry. The microcontroller 1505, in response to a PEC interrupt, will write the data byte first, then on the next PEC interrupt will write the mask for the data to this address. The compare circuitry of the OTA LCA 1502 stores the data appropriately. The register definition for the Command Compare Input Register is as follows:

ADDRESS: 0E015H

MSB LSB

7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - The data or mask value for a single compare operation.

'Value'- Data/Mask value. Note that a '0' will cause the corresponding data bit to be masked.

'00H' - Inactive (Initial condition)

The Compare Count Register (0E014H) defines a number of receive bytes to compare during a command template/mask compare sequence. The microcontroller 1505 writes to the Compare Count Register to control the number of receive bytes to compare (not the number of bytes to use out of compare buffer), and thereby activates a compare sequence for the next receive frame. The register definition is as follows:

ADDRESS: 0E014H

MSB LSB

7	6	5	4	3	2	1	0
N/A	N/A	N/A	CNT4	CNT3	CNT2	CNT1	CNT0

Bits 4-0 CNTx - Compare count value - Each count corresponds to a command template and mask from the compare buffer in the SRAM

'Value'- Count value (May range from, e.g., 03 to 1C hex)

'00H' - Inactive (Initial condition)

Bits 7-5 Unused

The DSP Data/Control Bytes Register (0E013H) is used in conjunction with operations relating to the DSP/vocoder 1510. When a frame sync is received by the OTA LCA 1502, encoded data is sent to the DSP/vocoder 1510 for decoding. The data generally consists of bearer data and DSP control information. To perform this transfer of data, the OTA LCA 1502 utilizes PEC requests to retrieve the stored information, one byte at a time. In response, the microcontroller 1505 writes the data and control bytes to the DSP Data/Control Bytes Register for transmission to the DSP/vocoder 1510. The register definition is as follows:

ADDRESS: 0E013H

MSB LSB

7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - Control information for DSP/vocoder.

'Value'- Control or data value.

'00H' - Inactive (Initial condition)

The DSP Source Register (0E012H) provides the mechanism for moving DSP encoded data and status bytes into the SRAM transmit buffer 1707 (see Fig. 17). After the DSP/vocoder 1510 has encoded data for transmission, the encoded data is sent to the OTA LCA 1502, and PEC requests are then made to have the encoded data placed in the SRAM transmit buffer 1707. When the microcontroller 1505 places information in the DSP Source Register, the OTA LCA 1502 thereafter places the data on the system bus 1530 for writing into the SRAM 1524. The register definition is as follows:

ADDRESS: 0E012H

MSB							LSB
7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - DSP encoded data

'Value'- DSP encoded data value.

'00H' - Inactive (Initial condition)

The Receive Source Register (0E011H) provides the address and data for read operations associated with data received over the link with the base station 104. As data comes in from the radio transceiver 305, the OTA LCA 1502 makes PEC requests to have data stored to the SRAM receive buffer 1706. With each read operation in which the microcontroller 1505 accesses the Receive Source Register, the OTA LCA 1502 responds with the newly received data to be placed in the SRAM receive buffer 1706. The register definition is as follows:

ADDRESS: 0E011H

MSB							LSB
7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx -Receive OTA data.

'Value'- Data value. This will be header information, D-channel or B-channel data.

'00H' - Inactive (Initial condition)

The Transmit Destination Register (0E010H) provides a target address for PECs associated with transmission of data over the communication link with the base station 104. When the OTA LCA 1502 requests data for transmission to the base station 104, the microcontroller 1505 writes the data to the Transmission Data Register in response to the PEC request. The register definition is as follows:

ADDRESS: 0E010H

MSB							LSB
7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - Data to be transmitted.

'Value'- Data value. This will be header, D-channel or B-channel data.

'00H' - Inactive (Initial condition)

The Sleep Control Register (0E00FH) defines the sleep control settings for the OTA LCA 1502 to the microcontroller 1505 and other hardware or peripherals. When the OTA LCA 1502 takes the microcontroller 1505 out of the standby or power down mode, the OTA LCA 1502 clears (i.e., resets) the two controller power saving mode bits of the Sleep Control Register. If both modes are active, the mode with the greatest power savings will prevail. The microcontroller 1505 is in charge of turning off the CODEC 1511, DSP/vocoder 1510, linearizer ASIC 1513 and the digital radio ASIC (DRA) as needed. The register definition is as follows:

ADDRESS: 0E00FH

MSB LSB

7	6	5	4	3	2	1	0
N/A	ALIVE	LINEARIZER ASIC SLP	RADIO ASIC SLP	CODEC SLEEP	DSP SLEEP	μC STNDBY	μC PWRDWN

80C165 Controller Specific:

Bit 0 μC PWRDWN - Indicates that the controller is going into power down mode and that the OTA LCA 1502 should reduce the controller clock to 0 Hz (set by microcontroller, cleared by OTA LCA).

'1' - Power down mode

'0' - Normal Operation (Initial condition)

Bit 1 μC STNDBY - Indicates that the microcontroller is going into its standby mode and that the OTA LCA 1502 should reduce the clock to the minimum frequency (set by controller, cleared by OTA LCA).

'1' - Standby mode

'0' - Normal operation (Initial condition)

Peripheral Power Saving registers:

Bit 2 DSP SLEEP - Instructs the OTA LCA 1502 to place the DSP in sleep mode (cleared and set by controller).

'1' - DSP sleep mode.

'0' - Normal Operation (Initial condition)

Bit 3 CODEC SLEEP - Instructs the OTA LCA 1502 to place the CODEC in sleep mode (cleared and set by controller).

'1' - Codec sleep mode.

'0' - Normal Operation (Initial condition)

Bit 4 RADIO ASIC SLP - Instructs the OTA LCA 1502 to place the digital radio ASIC (DRA)in sleep mode (cleared and set by controller).

'1' - Digital radio ASIC sleep mode

'0' - Normal Operation (Initial condition)

Bit 5 LINEARIZER ASIC SLP - Instructs the OTA LCA 1502 to place the linearizer ASIC 1513 in sleep mode (cleared and set by controller).

'1' - Linearizer ASIC sleep mode

'0' - Normal Operation (Initial condition)

Bit 6 ALIVE - Indicates that the controller has completed initialization. The bit acts as a flag indicator and is not latched. This bit will be written by the controller whenever it gets a reset operation.

'1' - μC has completed initialization (OTA LCA will reset)

'0' - NOP (Initial condition)

Bit 7 Unused

The Reset Register (0E00EH) is a read and write register which provides the microcontroller 1505 with reset control for several circuit blocks and external devices. Some resets are provided in other modules where it seemed reasonable. The OTA LCA 1502 provides a time duration of the reset operation, since this function is implemented for power up as well. The register definition is as follows:

ADDRESS: 0E00EH

MSB LSB

7	6	5	4	3	2	1	0
N/A	RIFASIC -RST	OTARS T	RCCNT -RST	DSP- RST	N/A	N/A	N/A

Bits 2-0 Unused

Bit 3 DSPRST - DSP reset - Instructs the OTA LCA 1502 to reset the DSP/vocoder

'1' - Reset (Initial condition - cleared by OTA LCA)

'0' - No reset

Bit 4 RCCNTRST - Instructs the OTA LCA 1502 to reset RC counter

'1' - Reset (Initial condition - cleared by OTA LCA)

'0' - No reset

Bit 5 OTARST - OTA reset - Resets the OTA LCA 1502

'1' - Reset (Initial condition - cleared by μC)

'0' - No reset

Bit 6 RIFASICRST - RIF ASIC reset - Instructs the OTA LCA 1502 to reset the digital radio ASIC (DRA)

'1' - Reset (Initial condition - cleared by OTA LCA)

'0' - No reset

Bit 7 Unused

The Receive Status Register (0E00DH) indicates the type of error condition when data is received from the radio transceiver 305 in error. The microcontroller 1505 may be informed of an error through a general status interrupt and, in response, may choose to read the Receive Status Register to determine the nature of the error. The register definition is as follows:

ADDRESS: 0E00DH

MSB							LSB
7	6	5	4	3	2	1	0
N/A	N/A	N/A	N/A	TRNSERR	FREQERR	HDRERR	FRMERR

Bit 0 FRMERR - Frame error - this will occur when the CRC over the entire receive frame did not validate.

'1' - Error

'0' - No error (Initial condition)

Bit 1 HDRERR - Header error - this will occur when the CRC over the received header did not validate.

'1' - Error

'0' - No error (Initial condition)

Bit 2 FREQERR - Frequency synthesizer lock error.

'1' - Error

'0' - No error (Initial condition)

Bit 3 TRNSERR - Indicates that the transmitter has failed.

'1' - Error

'0' - No error (Initial condition)

Bits 7-4 Unused

The RSSI Byte 1 & 2 Register (0E00CH) stores RSSI information. The RSSI bytes are preferably stored for each receive frame along with the header type. This allows the microcontroller 1505 to determine the signal strength. To retrieve the stored RSSI information, the microcontroller performs two reads from the RSSI Byte 1 & 2 Register. The first read obtains the first received RSSI byte (average), and the second read obtains the second received byte (difference). The RSSI Byte 1 & 2 Register contains the value of the last RSSI byte received, except upon power up. The hardware within the OTA LCA 1502 performs multiplexing of the internal RSSI register to the RSSI Byte 1 & 2 Register in the appropriate manner. To ensure that the RSSI read function in the OTA LCA 1502 hardware does not get out of sequence, both bytes should be consecutively read. The register definition is as follows:

ADDRESS: 0E00CH

MSB							LSB
7	6	5	4	3	2	1	0
RSSI7	RSSI6	RSSI5	RSSI4	RSSI3	RSSI2	RSSI1	RSSI0

Bits 7-0 RSSIx - RSSI byte.

'Value'- RSSI value. Hex value with maximum value of 3F.

'00H' - Inactive (Initial condition)

The SYNC Byte 1 & 2 Register (0E00BH) provides access to synchronization information used in conjunction with the DSP/vocoder 1510. To initiate DSP/vocoder operation, the OTA LCA 1502 writes two synchronization bytes to the DSP/vocoder 1510. These two bytes are "static" and, as they may present a problem to PEC flow, are stored within the OTA LCA 1502. To store the two sync bytes to the OTA LCA 1502, the microcontroller 1505 writes the sync bytes in two successive writes to the SYNC Byte 1 & 2 Register. The first write transfers the first sync byte to be sent to the DSP/vocoder 1510, while the second write transfers the second sync byte to be sent to the DSP/vocoder 1510. The hardware within the OTA LCA 1502 performs appropriate multiplexing to the two SYNC registers internal to the OTA LCA 1502. The register definition is as follows:

ADDRESS: 0E00BH

MSB							LSB
7	6	5	4	3	2	1	0
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0

Bits 7-0 SYNCx - May be synchronization byte 1 or 2 depending on which write.

'Value'- Sync byte.

'00H' - Inactive (Initial condition)

The SIM Transmit Count Register (0E008H) controls interactions with the SIM port 1520. The microcontroller 1505 writes to the SIM Transmit Count Register to define how many bytes will be transmitted to the SIM card through the SIM port 1520. After each successful transmission, the value stored in the SIM Transmit Count Register is decremented, until it eventually reaches zero. When that occurs, the SIM port 1520 changes to a receive mode. The register definition is as follows:

ADDRESS: 0E008H

MSB LSB

7	6	5	4	3	2	1	0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

Bits 7-0 CNTx - The SIM transmit byte count

'Value'- Byte count value.

'00H' - No transmit operation - inactive (Initial condition)

The SIM Control Register (0E007H) provides a means of controlling the SIM port 1520 according to the ISO/IEC 7816-3 specification. The microcontroller 1505 provides the enable settings and the times at which the settings are to be made. When the microcontroller 1505 initiates a reset operation, some of the information that is returned by the SIM card provides values for the control settings of the SIM port 1520. These setting are written to the SIM Control Register to control communication over the SIM port 1520. The register definition is as follows:

ADDRESS: 0E007H

MSB LSB

7	6	5	4	3	2	1	0
RESET LEVEL	ENABL EVCC	ENABLE CLOCK	MSBFRST	N/A	CLKSHL	PE/O	SIMINV

Bit 0 SIMINV - SIM invert data - This bit indicates to the SIM that the data should be inverted prior to transmission. In addition, it instructs the SIM that the data presented to a PEC will be inverted for the read sequence. The OTA LCA SIM port will perform the inversion of the data

'1' - Invert data

'0' - No inversion of data (Initial condition)

Bit 1 PE/O - Parity even or odd - Indicates to the SIM port that the parity generated and checked by the SIM transmitter should be odd or even.

'1' - Even parity

'0' - Odd parity - (Initial condition)

Bit 2 CLKSHL - Clock stop high or clock stop low - When the clock is to be terminated, this setting defines the value that it should be terminated to.

'1' - Clock terminate high

'0' - Clock terminate low (Initial condition)

Bit 3 Unused

Bit 4 MSBFRST - Most significant bit first during transmit and receive.

'1' - LSB first

'0' - MSB first (Initial condition)

Bit 5 ENABLE CLOCK - Enable the clock signal to the SIM card.

'1' - Enable CLOCK - provide CLOCK.

'0' - No CLOCK provided to SIM card (Initial condition) - clock output is '0'

Bit 6 ENABLE VCC - Enable power to the VCC pins of the SIM card.

'1' - Enable VCC - provide power to VCC

'0' - No power to the VCC pin of SIM card (Initial condition) - VCC level is '0'

Bit 7 RESET LEVEL - Apply the reset level to the SIM interface reset line.

'1' - High value applied to Reset signal

'0' - Low value applied to Reset signal (Initial condition)

The SIM Transmit Register (0E006H) provides a destination for data writes by the microcontroller 1505 in response to a PEC request for SIM data. When the SIM port 1520

is transmitting data to the SIM card, the OTA LCA 1502 makes a request of the microcontroller 1502, via the peripheral event controller, to retrieve more data to transmit to the SIM card. The PEC requests will continue until the SIM Transmit Count Register reaches zero, or a reset occurs. The register definition is as follows:

ADDRESS: 0E006H

MSB							LSB
7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - SIM port data to be transmitted to SIM card.

'Value'- Data value. Refer to ISO/IEC 7813-3 for details on the data values.

'00H' - Inactive (Initial condition)

The SIM Receive Register (0E006H) provides the source address for data reads by the microcontroller 1505 in response to a PEC request. Data/control information is read in through the SIM port 1520, then placed in the SRAM 1524 through a PEC request in conjunction with the data move by the microcontroller 1505. The register definition is as follows:

ADDRESS: 0E006H

MSB							LSB
7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Bits 7-0 DATAx - The data value to be read by controller.

'Value'- Data value. Refer to ISO/IEC 7813-3 for details on the data values.

'00H' - Inactive (Initial condition)

The Linearizer Control Register (0E005H) provides a control mechanism for the Linearizer ASIC 1513. The microcontroller 1505 writes to the Linearizer Control Register to set the desired control settings. The microcontroller 1505 may also read from this register to determine if the Linearizer ASIC 1513 is busy, or to determine the status of the control settings. The register definition is as follows:

ADDRESS: 0E005H

MSB							LSB
7	6	5	4	3	2	1	0
AUTOMODE	RECMODE	BUSY	N/A	N/A	N/A	N/A	N/A

Bits 0-4 Unused

Bit 5 BUSY - Indicates the status of the Linearizer. This bit is read-only.

'1' - Indicates the Linearizer interface is busy transmitting to the Linearizer ASIC 1513

'0' - Not busy (Initial condition)

Bit 6 RECMODE - This sets the Linearizer into receive mode. This bit is read and write.

'1' - Receive mode enabled.

'0' - Receive mode inactive (Initial condition)

Bit 7 AUTOMODE - This sets the Linearizer into autonomous mode. In this mode the Linearizer interface will issue update instructions without controller intervention. This bit is read and write.

'1' - Autonomous mode enabled

'0' - Autonomous mode inactive (Initial condition)

The Keyresp Register (0E003H) is a read only register which indicates the row of the key that is currently pressed by the user. The register definition is as follows:

ADDRESS: 0E003H

MSB LSB

7	6	5	4	3	2	1	0
ROW8	ROW7	ROW6	ROW5	ROW4	ROW3	ROW2	ROW1

Bits 7-0 ROWx - Indicates if a key in that row is depressed.

'1' - No key pressed (Initial condition)

'0' - Key in that row is pressed.

The Keypoll Register (0E003H) is a write-only register which allows the microcontroller 1505 to write values to the keypad matrix columns. The microcontroller 1505 is thereby able to detect the key that is currently pressed by the user. For key press identification, the microcontroller 1505 applies a '0' to one column at a time. The microcontroller 1505 reads the values of the rows in the key response register. The columns must be enabled for the values to be written to the keypad. The register definition is as follows:

ADDRESS: 0E003H

MSB LSB

7	6	5	4	3	2	1	0
COL4	COL3	COL2	COL1	COL_EN	N/A	N/A	N/A

Bits 2-0 Unused

Bit 3 COL_EN - Column Enable

'1' - use value of COLx to keypad

'0' - disable register values to keypad (Initial condition)

Bits 7-4 COLx - writing a 0 to any bit location will activate the corresponding column in the keypad matrix.

'1' - Inactive - no sample (Initial condition)

'0' - Sample column.

The Radio Configuration/Slot Map Data Register (0E002H) provides a destination register for radio configuration data that is transferred from flash memory 1525 through PEC operations. The register definition is as follows:

ADDRESS: 0E002H

MSB LSB

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bits 7-0 D7-D0 - Radio configuration or slot map data byte.

'Value'- Data value

'00H' - Inactive (Initial condition)

The Radio Configuration Code Register (0E001H) contains instructions to the radio interface line card assembly (RIF LCA) to reconfigure the radio transceiver 305. As the data is retrieved by the RIF LCA, the OTA LCA 1502 issues PEC requests to transfer further data to the RIF LCA. The register definition is as follows:

ADDRESS: 0E001H

MSB LSB

7	6	5	4	3	2	1	0
N/A	N/A	N/A	RFD4	RFD3	RFD2	RFD1	RFD0

Bits 4-0 RFD - Radio Data Code

'value' - Code setting for RF data.

'00000' - Normal operation (Initial condition)

Bits 7-5 Unused

The General Purpose Register (0E000H) is a write-only register which provides a variety of control functions to several areas of the OTA LCA 1502. The register definition is as follows:

ADDRESS: 0E000H

MSB LSB

7	6	5	4	3	2	1	0
DSP ENABLE	RINGER CTL1	RINGER CTL0	TX ENABLE	RSTRX	RSTSIM	N/A	N/A

Bits 1-0 Presently unused

Bit 2 RSTSIM - Reset SIM port - This signal provides the means to reset the SIM state machine. Generally a reset is attempted when the predefined number of bytes has been received, or a SIM error has occurred.

'1' - Reset SIM state machine

'0' - No reset (Initial condition)

Bit 3 RSTRX - Reset OTA LCA receive state machine. This bit will be cleared by the OTA LCA 1502 when the receive state machine has reset.

'1' - Reset

'0' - No reset (Initial condition)

Bit 4 TX ENABLE - Enable transmitter to transmit. This bit will be cleared by the OTA LCA 1502 once the transmit sequence has begun.

'1' - Enabled transmit

'0' - Disable (Initial condition)

Bits 6-5 RINGER CTLx- Turn on ringer to generate a tone. The ringer is disabled by software.

'11' - Tone 3 (medium) enable (if available)

'10' - Tone 2 (soft) enable

'01' - Tone 1 (loud) enable

'00' - Ringer disabled (Initial condition)

Bit 7 DSP ENABLE - Turn DSP function on.

'1' - DSP enabled

'0' - DSP disabled (Initial condition)

The General Purpose Register (0E000H) is a read-only register, which is read by the microcontroller 1505 at any time to determine the current status and/or mode of

operation. Generally the General Purpose Register will be read after the OTA LCA 1502 interrupts the microcontroller 1505 and indicates a change of mode. The register definition is as follows:

ADDRESS: 0E000H

MSB LSB

7	6	5	4	3	2	1	0
N/A	N/A	N/A	N/A	N/A	CONFIG2	CONFIG1	HANDS FREE

Bit 0 HANDSFREE

'1' - Handsfree not active

'0' - Handsfree kit is attached

Bits 2-1 CONFIGx

'Value'- Value indicates mode

Bits 7-3 Unused

While certain principles of operation have been described herein with reference to software embodiments and others with reference to hardware embodiments, those skilled in the art will appreciate that various features and functions may be implemented as either software or hardware, or a combination thereof, depending upon a variety of tradeoffs and other factors such as speed, size, cost, simplicity and efficiency. It will be understood that such implementation decisions and details are well within the purview of those skilled in the art and do not depart from the inventive features described herein.

The principles of the present invention are applicable to both mobile and fixed systems, and the embodiments disclosed herein may be deployed in a mobile communication environment or a fixed wireless local-loop system. The invention may also operate in conjunction or accordance with or in addition to features and techniques described in copending U.S. patent application serial number 09/159,714 and/or 09/160,010, each of which is incorporated by reference as if set forth fully herein.

In a preferred embodiment, the base station 104 and user stations 102 (or 301) communicate using spread spectrum communication. Each of the embodiments previously described can be configured to operate using spread spectrum communication. Suitable spread spectrum transmission and reception techniques are described, for example, in U.S. Patent Nos. 5,016,255, 5,022047 or 5,659,574, each of which is assigned to the assignee of the present invention, and each of which is hereby incorporated as if fully set forth herein. Different cells 103 (see Fig. 1) may be assigned different spread

spectrum codes (or different sets of spread spectrum codes, from which individual codes may be temporarily assigned to individual user stations 102), thereby obtaining benefits of CDMA techniques. In addition to using CDMA to distinguish transmissions in different cells 103, different frequencies may also be assigned to different cells 103, in the same or a different repeating pattern than CDMA codes are assigned.

While preferred embodiments of the invention have been described herein, many variations are possible which remain within the concept and scope of the invention. Such variations would become clear to one of ordinary skill in the art after inspection of the specification and the drawings. The invention therefore is not to be restricted except within the spirit and scope of any appended claims.

Claims

1. In a station capable of communicating according to time slots of a time frame, said station comprising a radio for transmitting and receiving signals, a control system comprising:

a memory;

a plurality of slot objects stored in said memory, each of said slot objects comprising a list of one or more time slots defining a communication channel; and

a slot manager, said slot manager allowing creation or deletion of said slot objects.

2. The control system of claim 1, further comprising a driver for interfacing to upper layer protocol components.

3. The control system of claim 2, wherein said slot manager and said driver communicate via input and output message queues.

4. The control system of claim 2, wherein said driver queues message to said slot manager to add, modify or delete slot objects in response to requests from upper layer protocol components.

5. The control system of claim 1, wherein each of the one or more time slots listed in a slot object comprises a next slot pointer indicating a subsequent time slot for communication.

6. The control system of claim 1, wherein each of said slot objects comprises radio configuration information.

7. The control system of claim 1, wherein the communication channel defined by the one or more time slots of a slot object can be either a normal traffic channel or a signalling channel.

8. The control system of claim 1, further comprising a plurality of interrupt service routines for transferring data between source data storage locations and destination data storage locations in response to interrupts.

9. The control system of claim 8, wherein each interrupt routine transfers a single byte of information between a source data storage location and a destination data storage location in response to a corresponding one of said interrupts.

10. The control system of claim 1, wherein said slot objects are processed sequentially in an order determined by the relative time slot positions of the time slots of said slot objects.

11. The control system of claim 10, wherein a slot object is processed by (a) configuring the radio in accordance with radio configuration information stored in said slot object, and (b) transmitting or receiving information in at least one time slot listed in said slot object.

12. The control system of claim 11, wherein a slot object is further processed by identifying a subsequent time slot for processing, and repeating steps (a) and (b) for the subsequent time slot.

13. An apparatus, comprising:

a radio transceiver;

a microprocessor;

a peripheral event controller connected to said microprocessor, said peripheral event controller receiving a plurality of interrupt signals;

a plurality of source data storage locations;

a plurality of destination data storage locations; and

an interface connected to said radio transceiver and to said peripheral event controller, whereby data from said source data storage locations is transferred by said microcontroller to said destination data storage locations in response to interrupts generated from said interface.

14. The apparatus of claim 13, wherein one byte of data is transferred by said microcontroller in response to each interrupt generated from said interface.

15. The apparatus of claim 13, further comprising a memory for storing programming instructions for said microprocessor, said programming instructions comprising a plurality of interrupt service routines.

16. The apparatus of claim 15, wherein each of said interrupt service routines corresponds to an interrupt to said peripheral event controller.

17. The apparatus of claim 16, wherein each of said interrupt service routines causes said microcontroller to transfer data between a source data location register and a destination data location register in response to its corresponding interrupt.

18. The apparatus of claim 13, further comprising a memory for storing a plurality of slot objects, each of said slot objects comprising a list of one or more time slots defining a communication channel.

19. The apparatus of claim 18, wherein said slot objects are created, modified or deleted by said microcontroller according to a set of programming instructions.

20. In a station capable of communicating according to time slots of a time frame, a method comprising the steps of:

generating one or more slot objects, each slot object comprising a list of one or more time slots collectively defining a communication channel;

processing said one or more slot objects in a sequence determined by the order of time slots listed in the slot objects; and

configuring a radio transceiver at the station to transmit or receive in the time slots listed in the slot objects.

21. The method of claim 20, wherein each of said one or more slot objects comprises a slot profile.

22. The method of claim 21, wherein the slot profile identifies whether the one or more time slots of the slot object are transmit time slots or receive time slots.

23. The method of claim 22, wherein the slot profile comprises radio configuration information.

24. The method of claim 21, wherein the slot profile comprises a message type identifier indicating the type of messages to be transmitted or received over the communication channel defined by the one or more time slots of the slot object.

25. The method of claim 24, wherein said message type identifier indicates whether the type of messages to be transmitted or received are voice messages or signalling messages.

26. The method of claim 20, wherein each of said one or more slot objects comprises a plurality of time slot tags, each time slot tag identifying a time slot for communication.

27. The method of claim 26, wherein each time slot tag further comprises a next slot pointer identifying a subsequent time slot for communication, said subsequent time slot being listed in the slot objects.

28. The method of claim 20, wherein said step of configuring a radio transceiver at the station to transmit or receive in the time slots listed in the slot objects comprises the steps of

configuring the radio transceiver to transmit or receive in communication with a first target station in the time slots listed in a first slot object, and

configuring the radio transceiver to transmit or receive in communication with a second target station in the time slots listed in a second slot object.

29. In a station capable of communicating according to time slots of a time frame, a method for controlling communication, comprising the steps of:

setting up a plurality of direct access memory channels between source data storage locations and destination data storage locations, said direct access memory channels managed by a microcontroller;

generating interrupts periodically to the microcontroller; and

in response to said interrupts, transferring data across said direct access memory channels between said source data storage locations and said destination data storage locations.

30. The method of claim 29, wherein the step of transferring data across said direct access memory channels between said source data storage locations and said destination data storage locations comprises the step of, for each interrupt, transferring a single byte of data between a source data storage location and a destination data storage location.

31. In a station capable of communicating according to time slots of a time frame, said station comprising a radio transceiver, a method of communication comprising the steps of:

generating a first slot object, said first slot object identifying one or more time slots defining a first communication channel with a first target station;

generating a second slot object, said second slot object identifying one or more time slots defining a second communication channel with a second target station;

setting up the radio transceiver to transmit or receive over said first communication channel with the first target station in the one or more time slots identified in said first slot object; and

setting up the radio transceiver to transmit or receive over said second communication channel with the second target station in the one or more time slots identified in said second slot object.

32. The method of claim 31, wherein said first slot object and said second slot object each comprise radio configuration information.

33. The method of claim 31, wherein said first slot object and said second slot object each identify a classification of messages to be transmitted or received over said first communication channel and said second communication channel, respectively.

34. The method of claim 33, wherein said classification of messages to be transmitted or received includes a voice message type classification and a signalling message type classification.

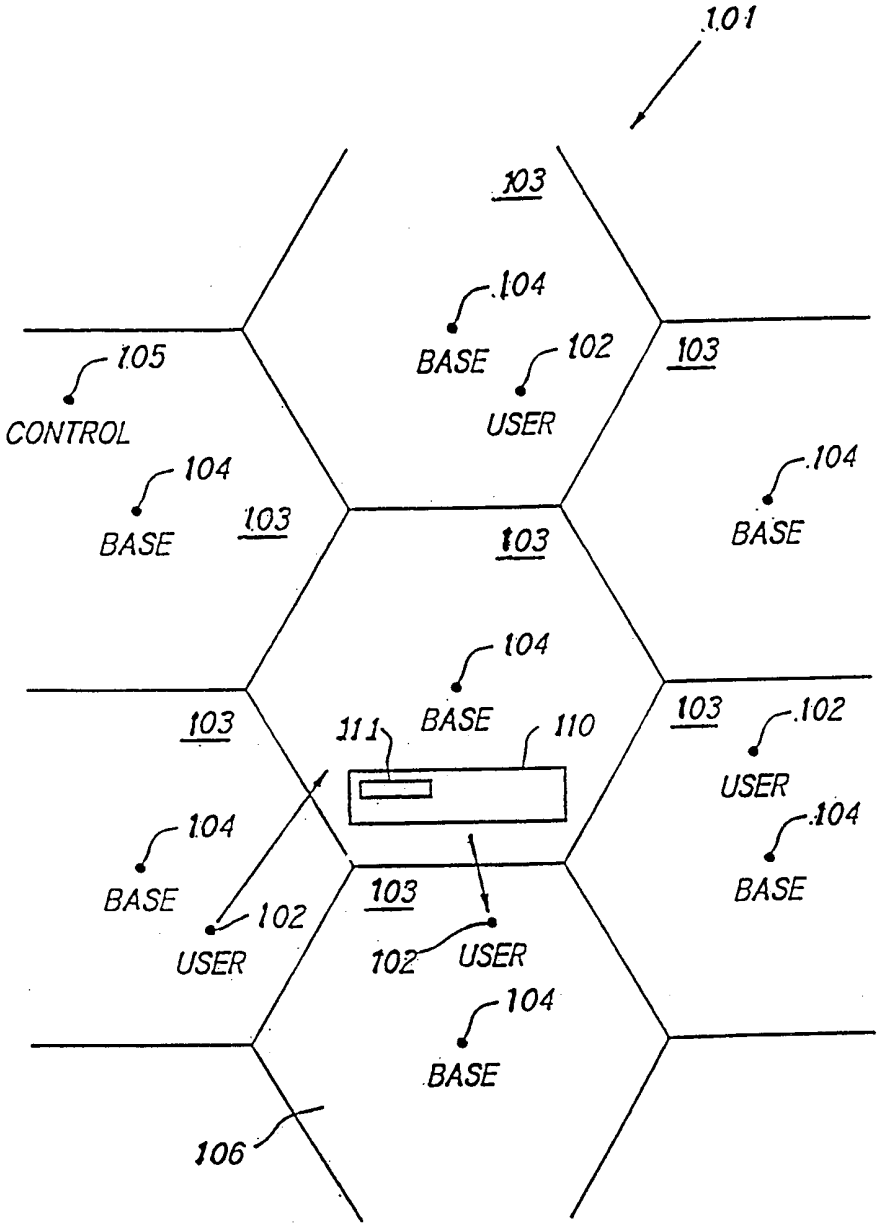

101
103
104
BASE
102
USER
105
CONTROL
104
BASE
103
103
104
BASE
103
104
BASE
111
110
103
102
USER
104
BASE
104
BASE
103
102
USER
102
USER
104
BASE
106

FIG 1

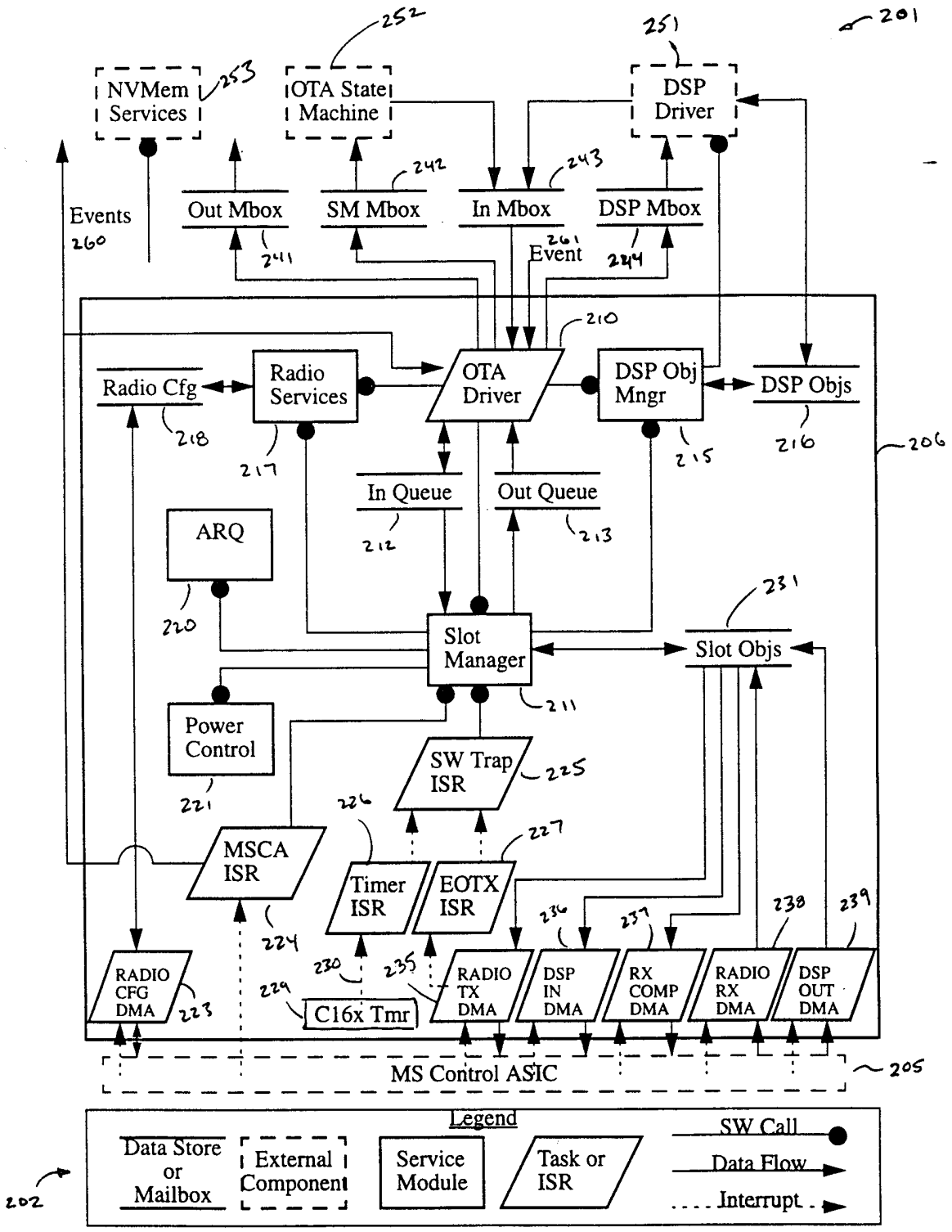

NVMem Services
253
OTA State Machine
252
DSP Driver
251
201
Events
260
Out Mbox
241
SM Mbox
242
In Mbox
243
DSP Mbox
244
Event
261
210
Radio Cfg
218
Radio Services
217
OTA Driver
DSP Obj Mngr
215
DSP Objs
216
206
In Queue
212
Out Queue
213
ARQ
220
231
Slot Manager
Slot Objs
211
Power Control
221
SW Trap ISR
225
226
227
MSCA ISR
224
Timer ISR
EOTX ISR
236
237
238
239
RADIO CFG DMA
223
230
235
229
C16x Tmr
RADIO TX DMA
DSP IN DMA
RX COMP DMA
RADIO RX DMA
DSP OUT DMA
MS Control ASIC
205
Legend
Data Store or Mailbox
External Component
Service Module
Task or ISR
SW Call
Data Flow
Interrupt
202

FIG. 2

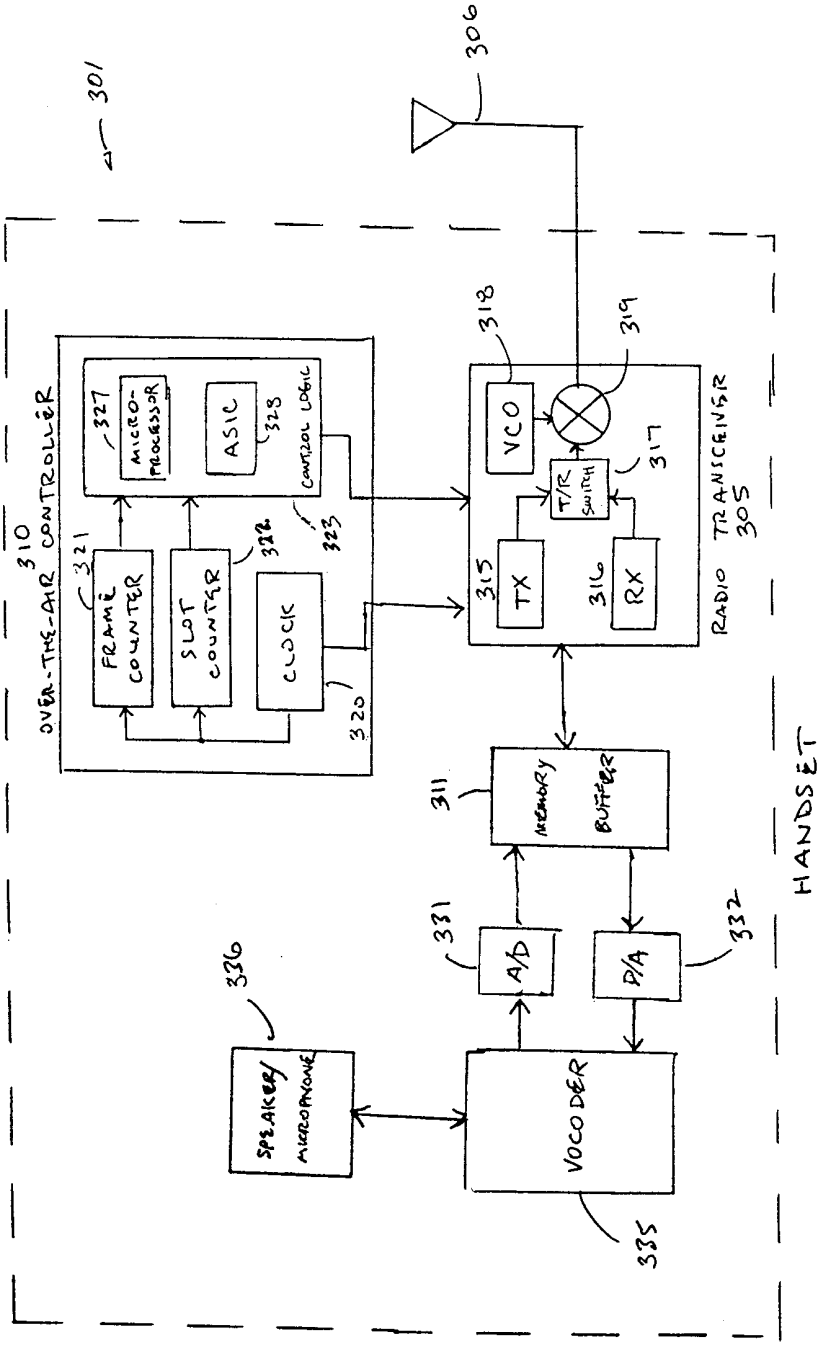

301
306
310
OVER-THE-AIR CONTROLLER
327
MICRO-PROCESSOR
ASIC
328
CONTROL LOGIC
323
321
FRAME COUNTER
SLOT COUNTER
322
CLOCK
320
318
VCO
319
T/R SWITCH
317
315
TX
316
RX
RADIO TRANSCEIVER 305
311
MEMORY BUFFER
331
A/D
D/A
332
336
SPEAKER/ MICROPHONE
VOCODER
335
HANDSET

FIG. 3

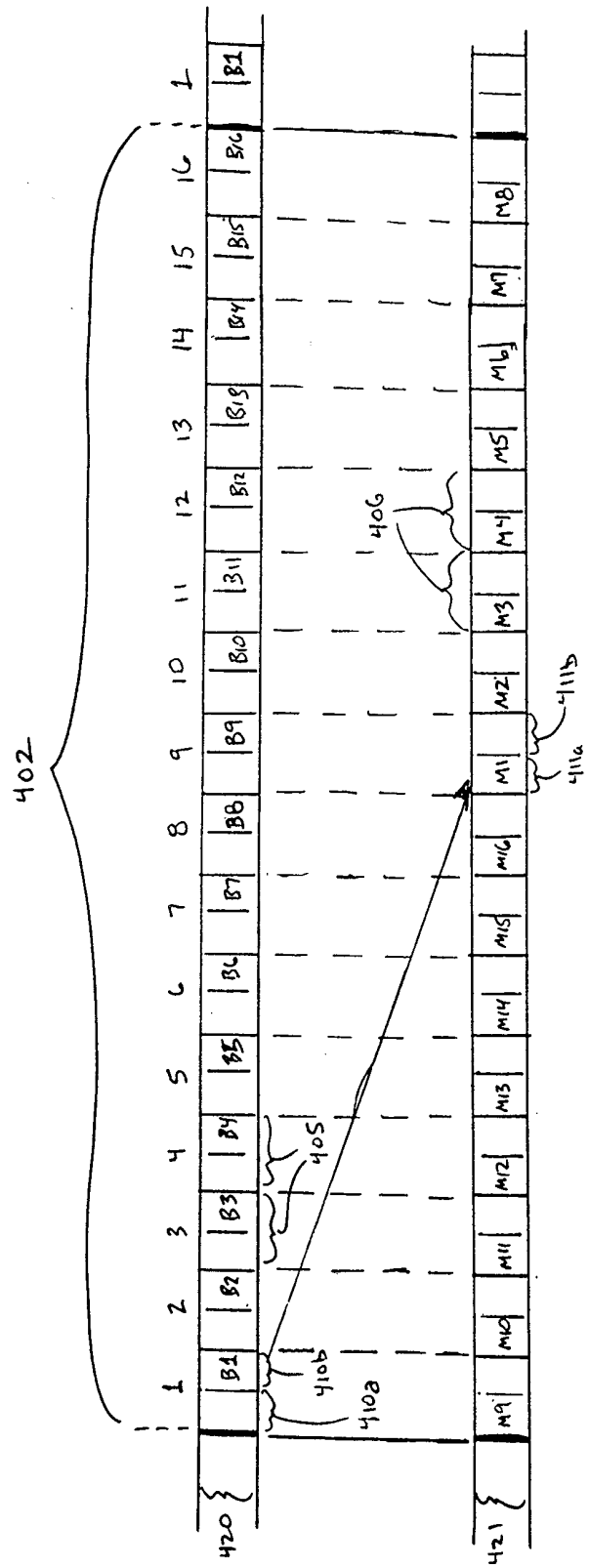

402
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
M1
M2
M3
M4
M5
M6
M7
M8
M9
M10
M11
M12
M13
M14
M15
M16
410a
410b
405
406
411a
411b
420
421
401

FIG. 4

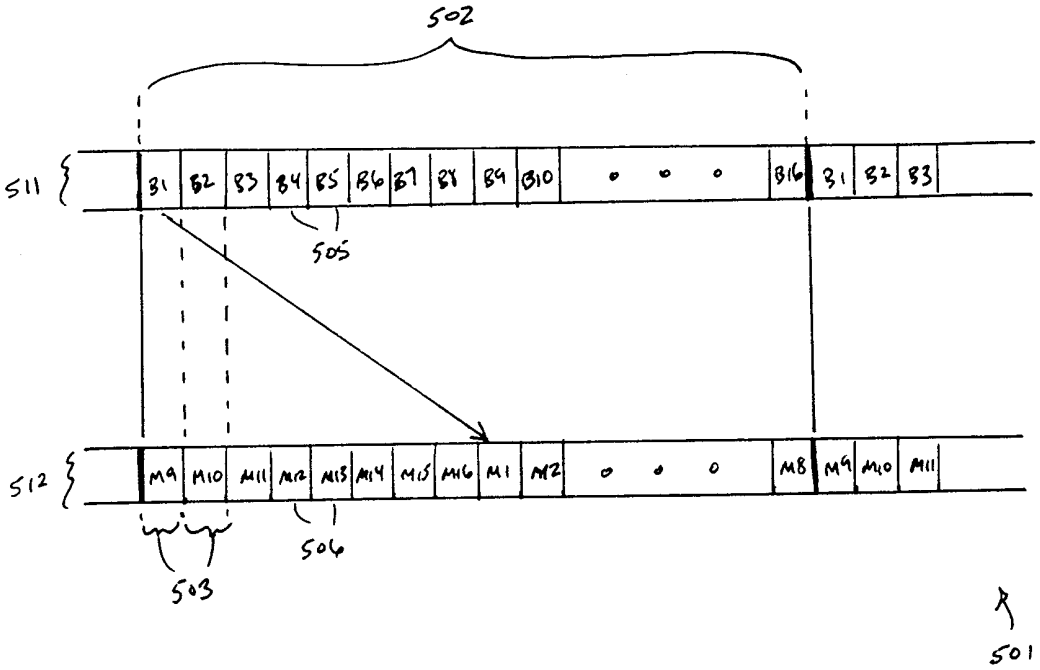

502
511
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B16 B1 B2 B3
505
512
M9 M10 M11 M12 M13 M14 M15 M16 M1 M2 M8 M9 M10 M11
503
506
501

FIG. 5

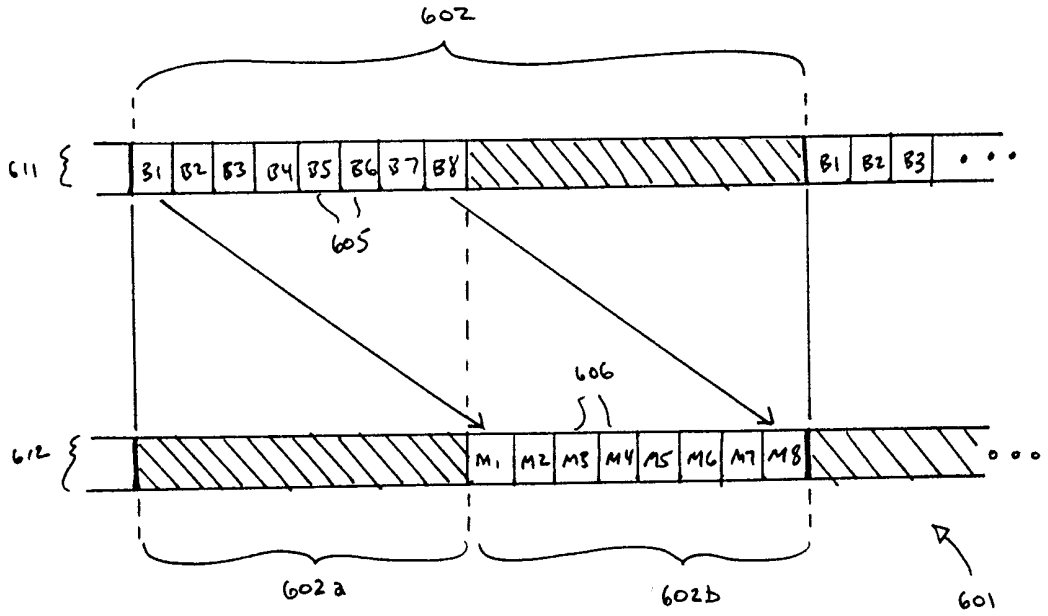

602
611
B1 B2 B3 B4 B5 B6 B7 B8 B1 B2 B3
605
606
612
M1 M2 M3 M4 M5 M6 M7 M8
602a
602b
601

FIG. 6

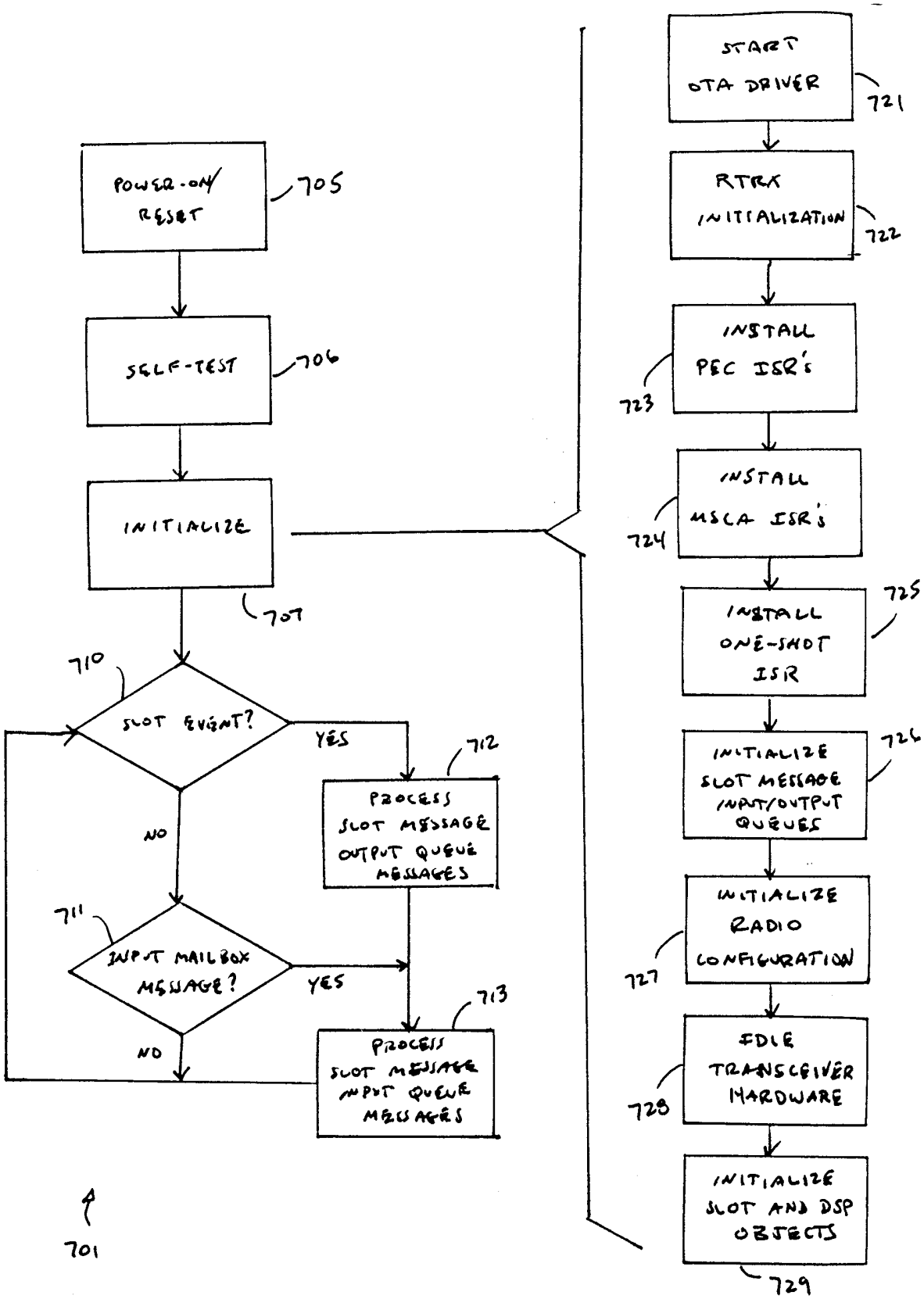

POWER-ON RESET
705
SELF-TEST
706
INITIALIZE
707
710
SLOT EVENT?
YES
712
PROCESS SLOT MESSAGE OUTPUT QUEUE MESSAGES
NO
711
INPUT MAILBOX MESSAGE?
YES
713
PROCESS SLOT MESSAGE INPUT QUEUE MESSAGES
NO
701
START OTA DRIVER
721
RTRX INITIALIZATION
722
INSTALL PEC ISR's
723
INSTALL MSCA ISR's
724
INSTALL ONE-SHOT ISR
725
INITIALIZE SLOT MESSAGE INPUT/OUTPUT QUEUES
726
INITIALIZE RADIO CONFIGURATION
727
IDLE TRANSCEIVER HARDWARE
728
INITIALIZE SLOT AND DSP OBJECTS
729

FIG. 7

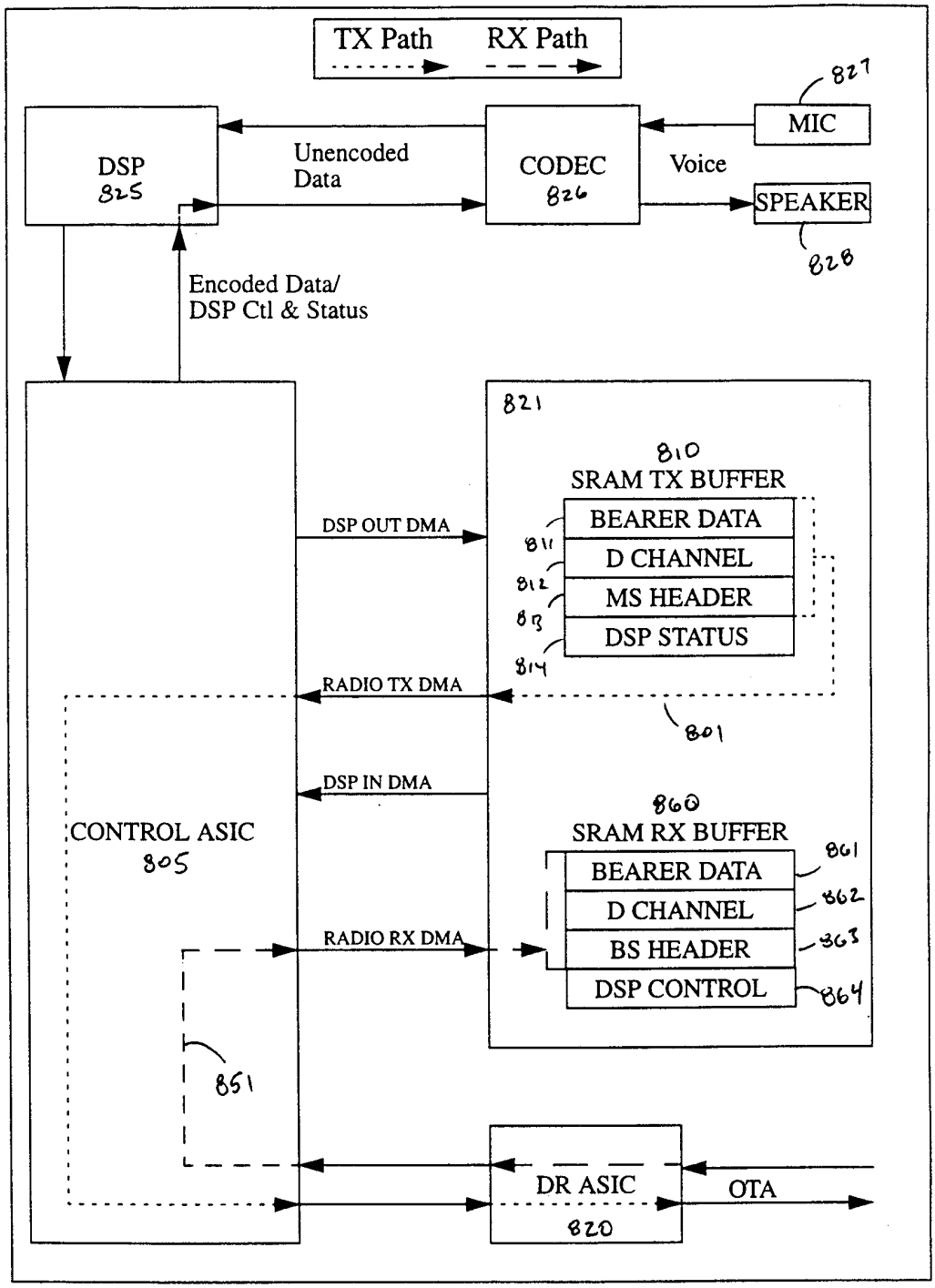

TX Path
RX Path
827
MIC
DSP
825
Unencoded
Data
CODEC
826
Voice
SPEAKER
828
Encoded Data/
DSP Ctl & Status
821
810
SRAM TX BUFFER
DSP OUT DMA
BEARER DATA
D CHANNEL
MS HEADER
DSP STATUS
811
812
813
814
RADIO TX DMA
801
DSP IN DMA
860
SRAM RX BUFFER
CONTROL ASIC
805
BEARER DATA
D CHANNEL
BS HEADER
DSP CONTROL
861
862
863
864
RADIO RX DMA
851
DR ASIC
820
OTA

FIG. 8

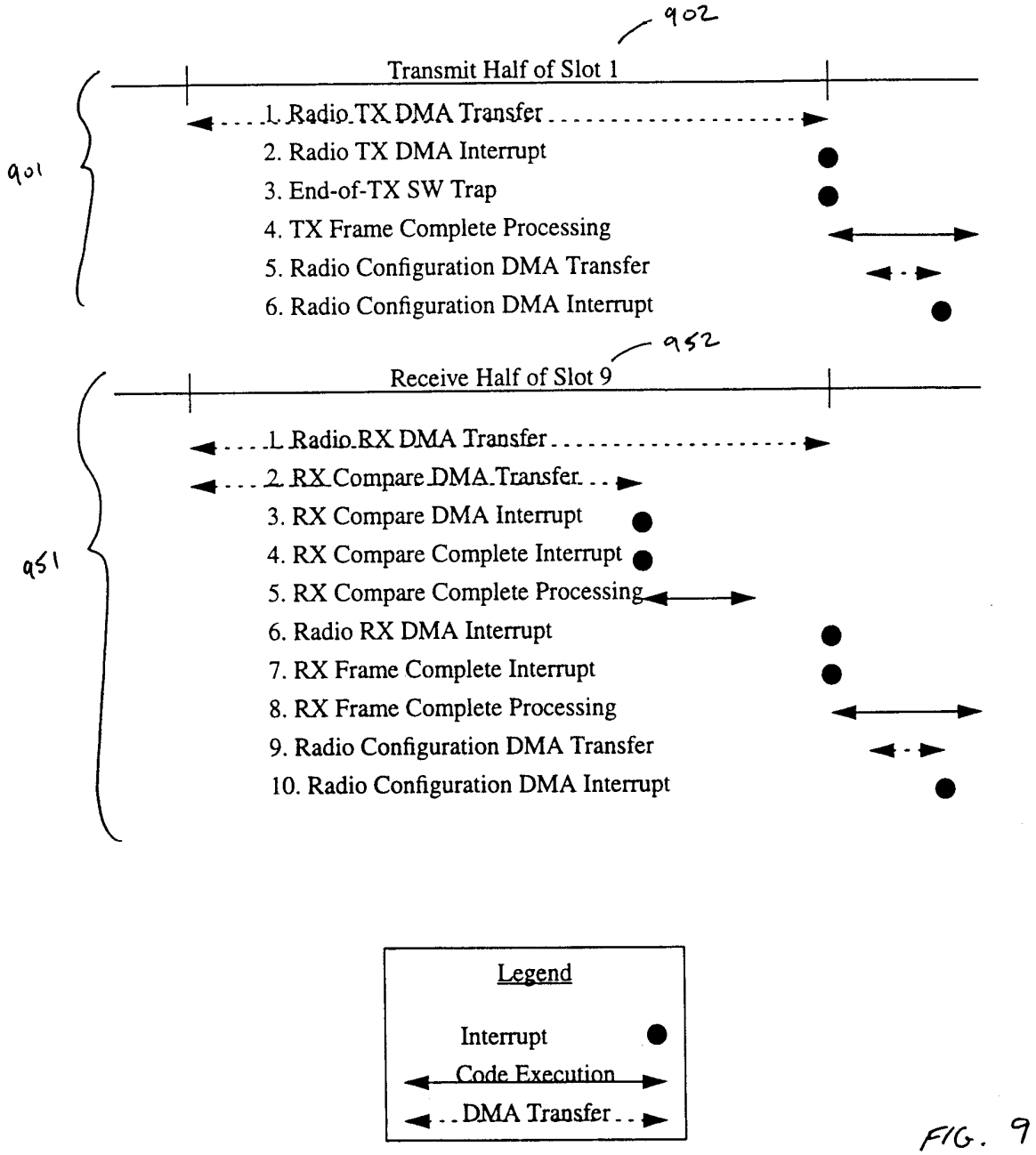

902
Transmit Half of Slot 1
1. Radio TX DMA Transfer
2. Radio TX DMA Interrupt
3. End-of-TX SW Trap
4. TX Frame Complete Processing
5. Radio Configuration DMA Transfer
6. Radio Configuration DMA Interrupt
901
952
Receive Half of Slot 9
1. Radio RX DMA Transfer
2. RX Compare DMA Transfer
3. RX Compare DMA Interrupt
4. RX Compare Complete Interrupt
5. RX Compare Complete Processing
6. Radio RX DMA Interrupt
7. RX Frame Complete Interrupt
8. RX Frame Complete Processing
9. Radio Configuration DMA Transfer
10. Radio Configuration DMA Interrupt
951
Legend
Interrupt
Code Execution
DMA Transfer

FIG. 9

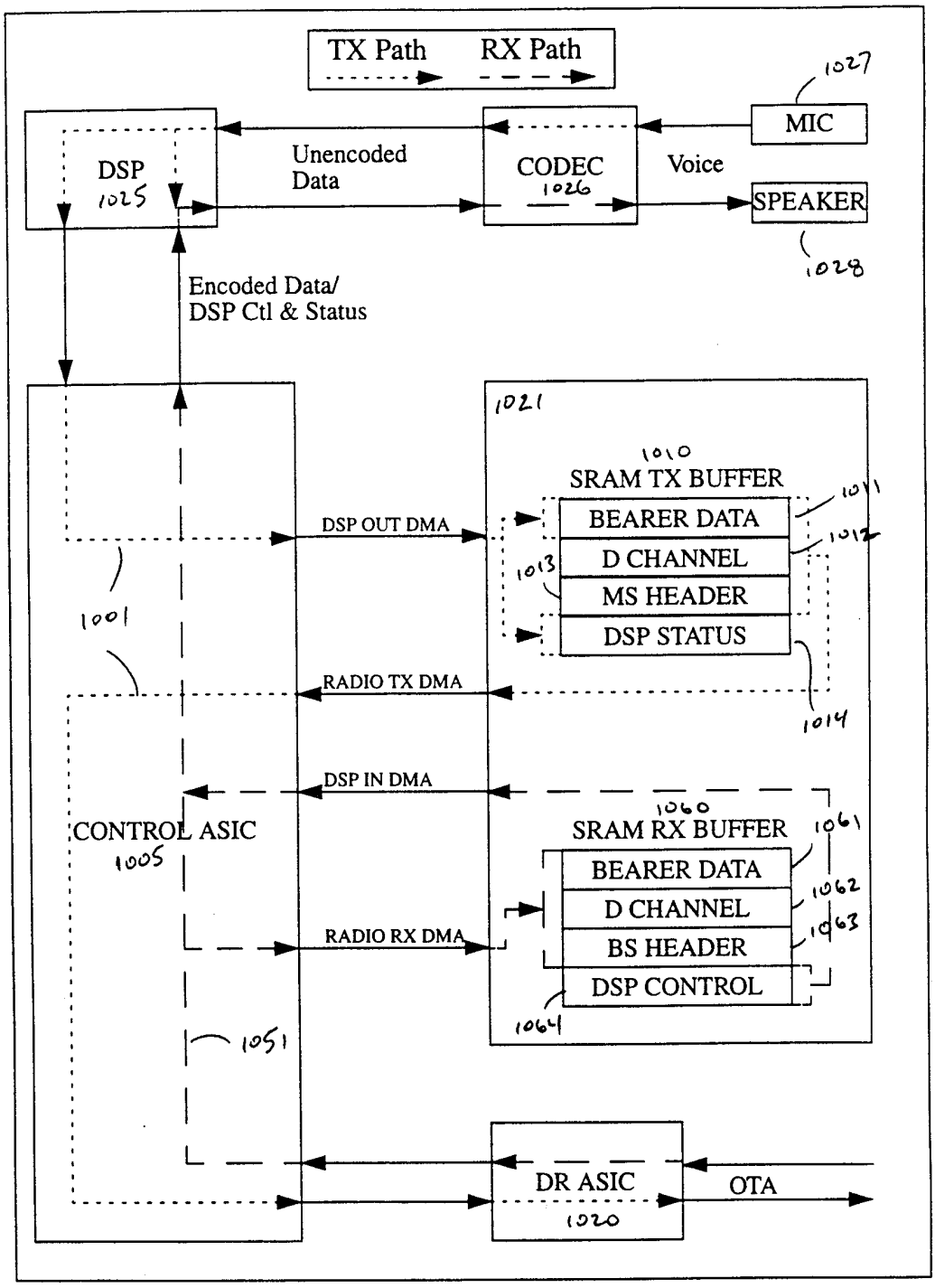

TX Path
RX Path
1027
MIC
DSP
1025
Unencoded
Data
CODEC
1026
Voice
SPEAKER
1028
Encoded Data/
DSP Ctl & Status
1021
1010
SRAM TX BUFFER
1011
BEARER DATA
1012
D CHANNEL
1013
MS HEADER
DSP STATUS
DSP OUT DMA
1001
RADIO TX DMA
1014
DSP IN DMA
1060
SRAM RX BUFFER
1061
BEARER DATA
1062
D CHANNEL
1063
BS HEADER
DSP CONTROL
1064
CONTROL ASIC
1005
RADIO RX DMA
1051
DR ASIC
1020
OTA

FIG. 10

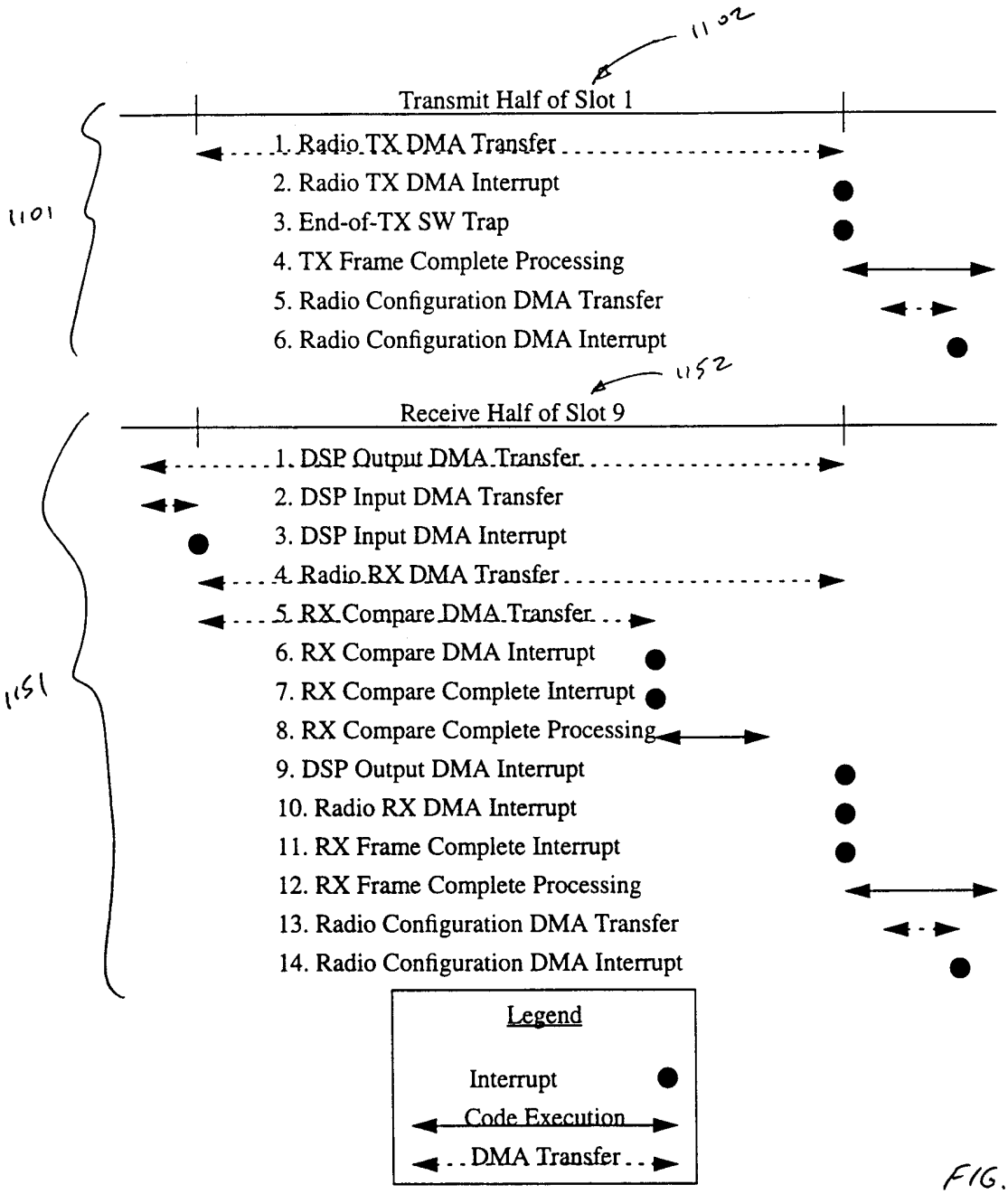

1102
Transmit Half of Slot 1
1. Radio TX DMA Transfer
2. Radio TX DMA Interrupt
3. End-of-TX SW Trap
4. TX Frame Complete Processing
5. Radio Configuration DMA Transfer
6. Radio Configuration DMA Interrupt
1101
1152
Receive Half of Slot 9
1. DSP Output DMA Transfer
2. DSP Input DMA Transfer
3. DSP Input DMA Interrupt
4. Radio RX DMA Transfer
5. RX Compare DMA Transfer
6. RX Compare DMA Interrupt
7. RX Compare Complete Interrupt
8. RX Compare Complete Processing
9. DSP Output DMA Interrupt
10. Radio RX DMA Interrupt
11. RX Frame Complete Interrupt
12. RX Frame Complete Processing
13. Radio Configuration DMA Transfer
14. Radio Configuration DMA Interrupt
1151
Legend
Interrupt
Code Execution
DMA Transfer

FIG. 11

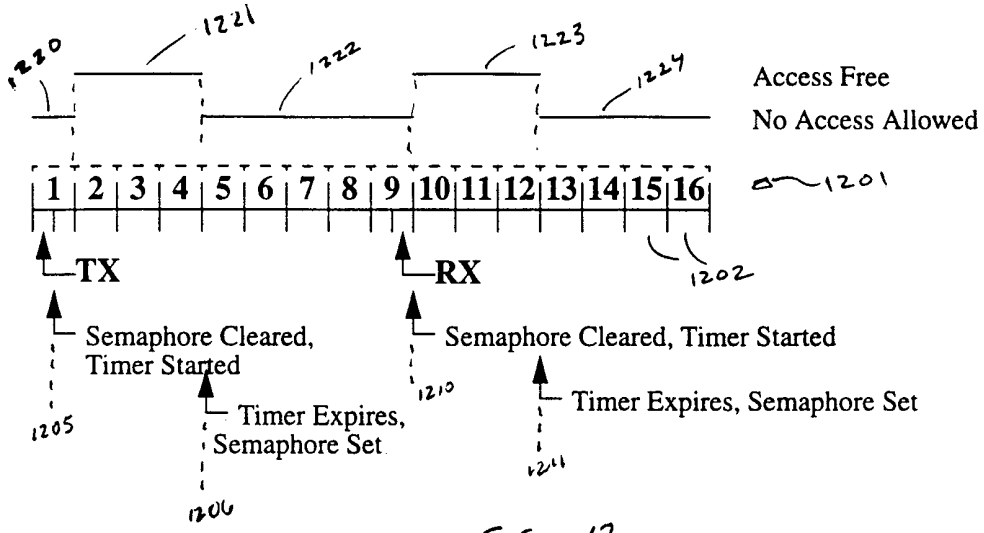

1220
1221
1222
1223
1224
Access Free
No Access Allowed
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1201
TX
RX
1202
Semaphore Cleared, Timer Started
1205
Timer Expires, Semaphore Set
1206
Semaphore Cleared, Timer Started
1210
Timer Expires, Semaphore Set
1211

FIG. 12

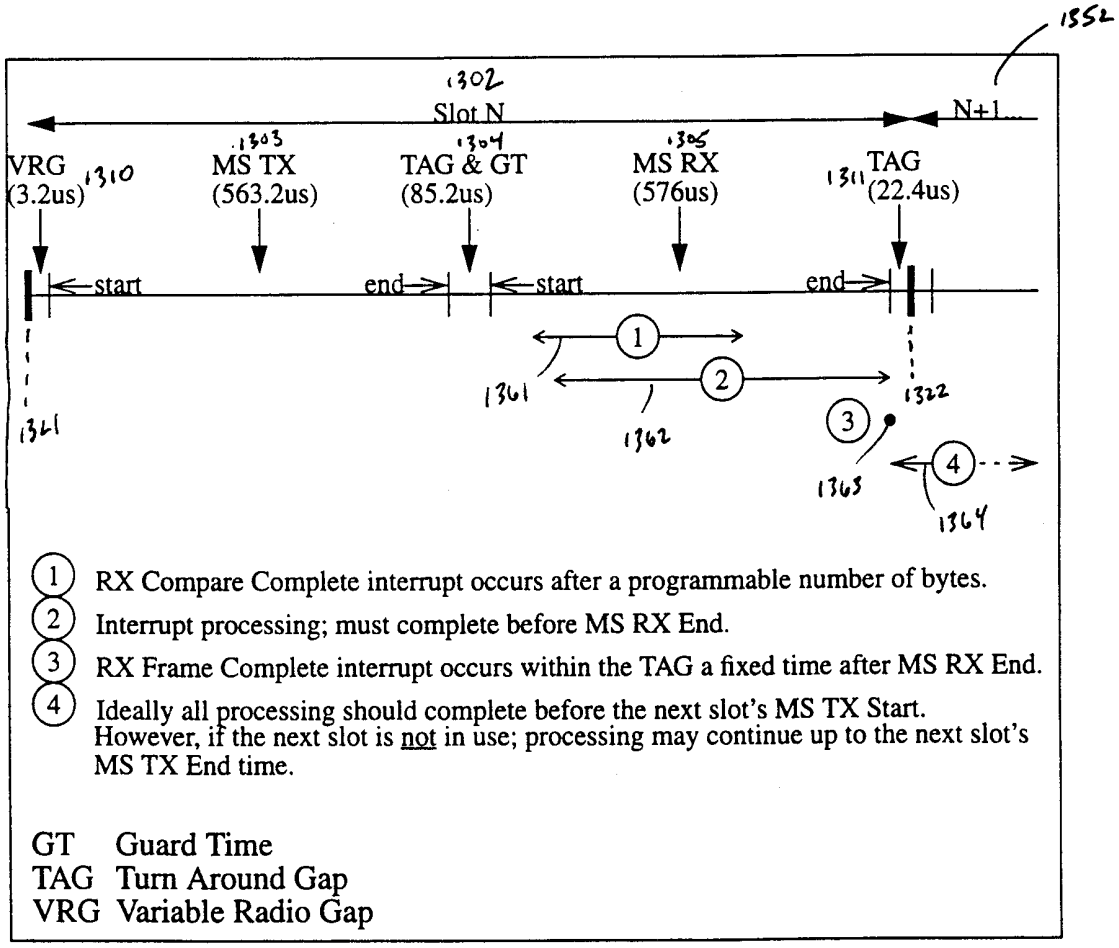

1552
1302
Slot N
N+1
VRG (3.2us)
1310
1303
MS TX (563.2us)
1304
TAG & GT (85.2us)
1305
MS RX (576us)
1311
TAG (22.4us)
start
end
start
end
1321
1361
1362
1322
1363
1364
1 RX Compare Complete interrupt occurs after a programmable number of bytes.
2 Interrupt processing; must complete before MS RX End.
3 RX Frame Complete interrupt occurs within the TAG a fixed time after MS RX End.
4 Ideally all processing should complete before the next slot's MS TX Start. However, if the next slot is not in use; processing may continue up to the next slot's MS TX End time.
GT Guard Time
TAG Turn Around Gap
VRG Variable Radio Gap

FIG. 13

Message CRC Status	RX Msg#	RX ACK	Special Msg	TX ACK	TX Msg#	Next Expected RX Msg#
Bad head & bad frame CRC, or no Msg RX	-	-	Cannot determine.	NAK	Same	Same
Good head & bad frame CRC	Expected	ACK	"	NAK	Next	Same
"	"	NAK	"	NAK	Same	Same
"	Unex-pected	Either	"	NAK	Same	Same
Good frame CRC	Expected	ACK	No	ACK	Next	Next
"	"	NAK	No	ACK	Same	Next
"	"	-	Yes (n/a - will not occur)	-	-	-
"	Unex-pected	Either	N	ACK	Same	Same
"	"	ACK	Y	ACK	Next	Same
"	"	NAK	Y	ACK	Same	Same

a. The TX Msg# is only promoted if 1) an ACK to the current TX Msg# has been received and 2) if the MS has new data to send.

FIG. 14

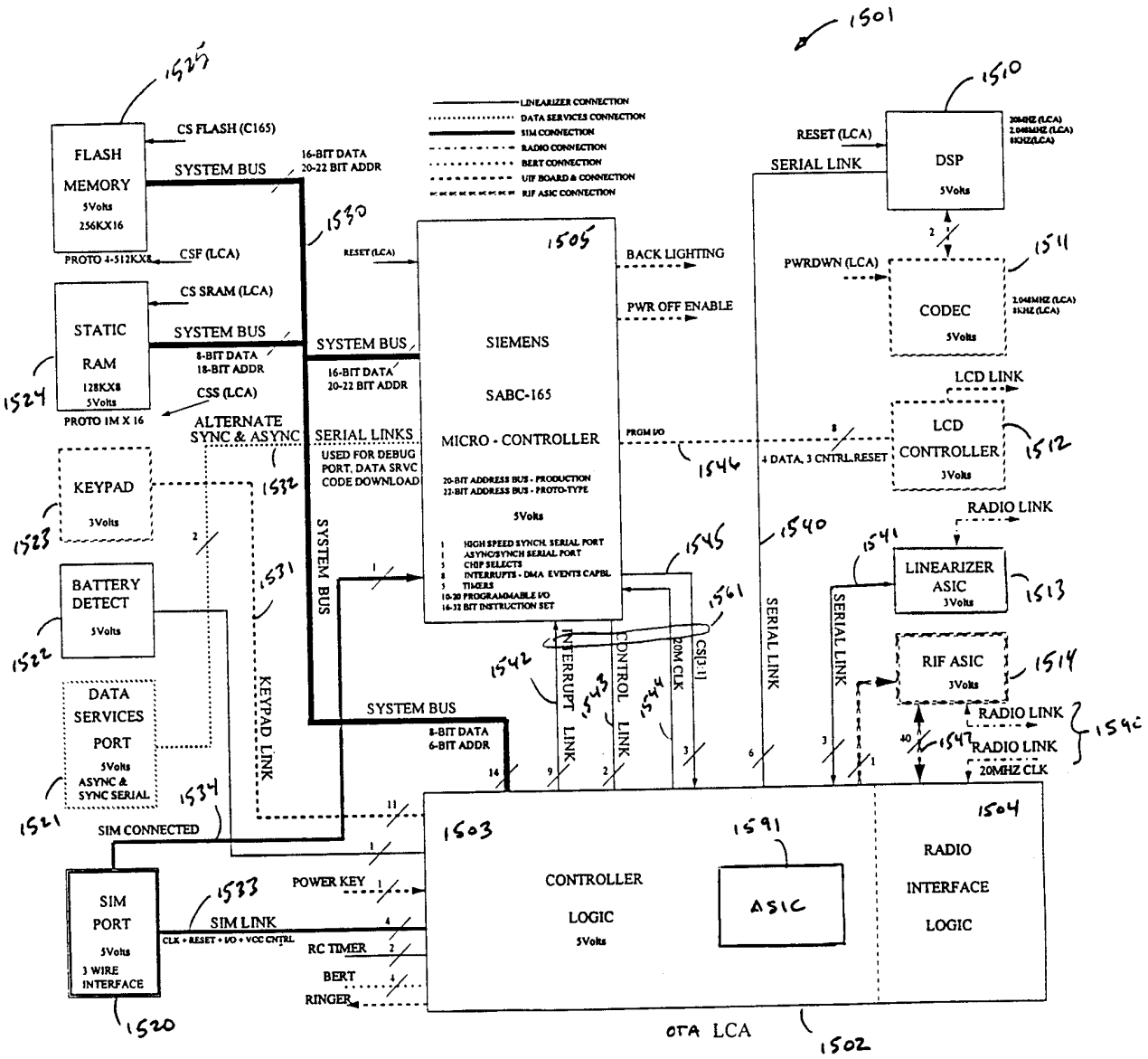

1501
1525
FLASH MEMORY 5Volts 256KX16
CS FLASH (C165)
SYSTEM BUS
16-BIT DATA 20-22 BIT ADDR
PROTO 4-512KX8
CSF (LCA)
1530
STATIC RAM 128KX8 5Volts
CS SRAM (LCA)
SYSTEM BUS
8-BIT DATA 18-BIT ADDR
CSS (LCA)
1524
PROTO 1M X 16
ALTERNATE SYNC & ASYNC
SERIAL LINKS
USED FOR DEBUG PORT, DATA SRVC CODE DOWNLOAD
1532
KEYPAD 3Volts
1528
BATTERY DETECT 5Volts
1522
1531
DATA SERVICES PORT 5Volts ASYNC & SYNC SERIAL
1521
KEYPAD LINK
SYSTEM BUS
1534
SIM CONNECTED
SIM PORT 5Volts 3 WIRE INTERFACE
1520
1533
SIM LINK
CLK + RESET + I/O + VCC CNTRL
POWER KEY
RC TIMER
BERT
RINGER
LINEARIZER CONNECTION
DATA SERVICES CONNECTION
SIM CONNECTION
RADIO CONNECTION
BERT CONNECTION
UIF BOARD & CONNECTION
RIF ASIC CONNECTION
1505
SIEMENS SABC-165 MICRO - CONTROLLER
20-BIT ADDRESS BUS - PRODUCTION
22-BIT ADDRESS BUS - PROTO-TYPE
5Volts
1 HIGH SPEED SYNCH. SERIAL PORT
1 ASYNC/SYNCH SERIAL PORT
5 CHIP SELECTS
8 INTERRUPTS - DMA EVENTS CAPBL
5 TIMERS
10-20 PROGRAMMABLE I/O
16-32 BIT INSTRUCTION SET
RESET (LCA)
BACK LIGHTING
PWR OFF ENABLE
PRGM I/O
1546
1545
1561
INTERRUPT LINK
CONTROL LINK
20M CLK
CS[3:1]
1542
1543
1544
1503
CONTROLLER LOGIC 5Volts
1591
ASIC
1504
RADIO INTERFACE LOGIC
OTA LCA
1502
1510
DSP 5Volts
RESET (LCA)
SERIAL LINK
PWRDWN (LCA)
1511
CODEC 5Volts
LCD LINK
LCD CONTROLLER 3Volts
1512
4 DATA, 3 CNTRL, RESET
RADIO LINK
1540
1541
LINEARIZER ASIC 3Volts
1513
SERIAL LINK
RIF ASIC 3Volts
1514
RADIO LINK
RADIO LINK
20MHZ CLK
1547
1540

FIG. 15

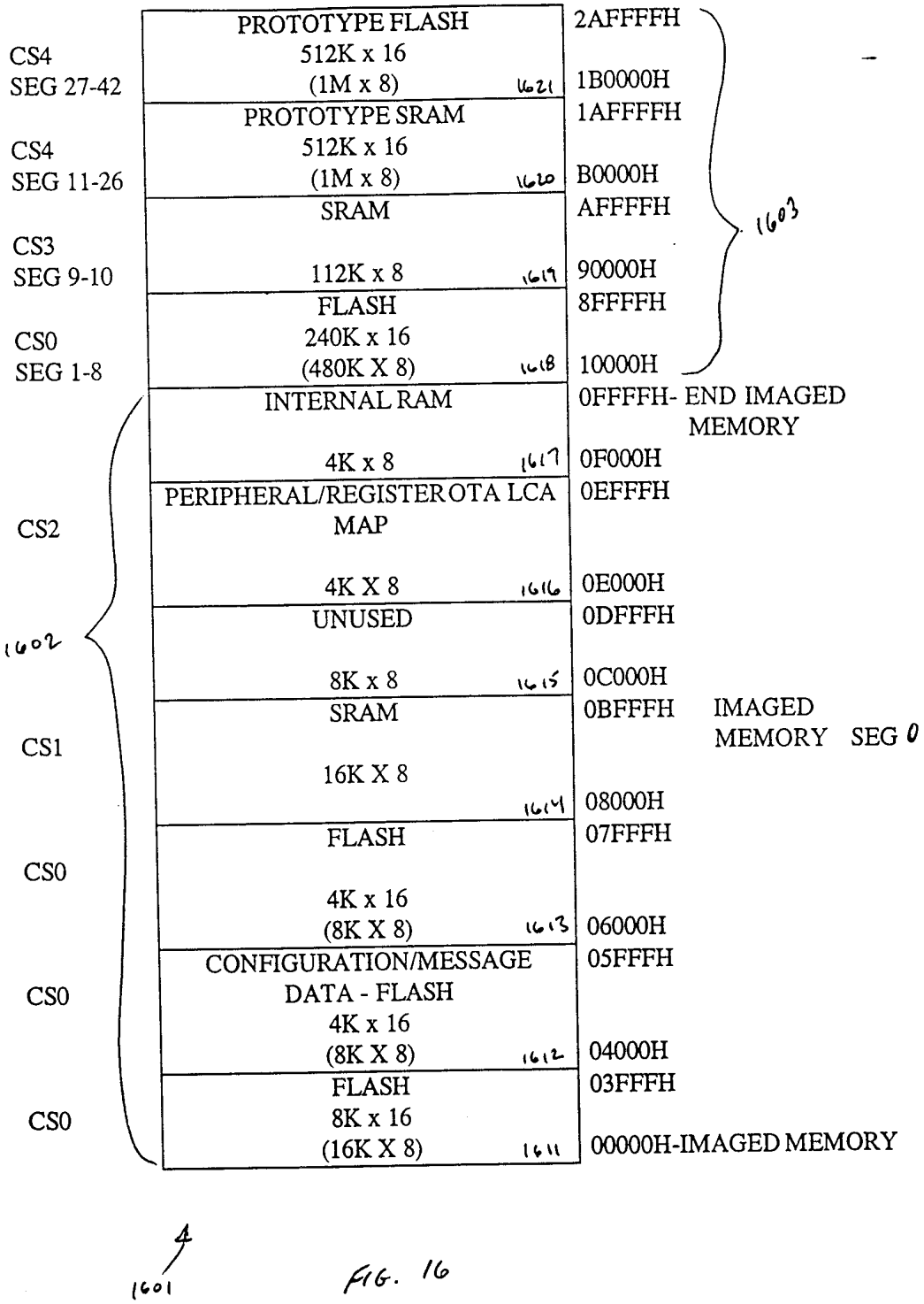

PROTOTYPE FLASH
512K x 16
(1M x 8)
1621
2AFFFFH
1B0000H
CS4
SEG 27-42
PROTOTYPE SRAM
512K x 16
(1M x 8)
1620
1AFFFFH
B0000H
CS4
SEG 11-26
SRAM
112K x 8
1619
AFFFFH
90000H
CS3
SEG 9-10
FLASH
240K x 16
(480K X 8)
1618
8FFFFH
10000H
CS0
SEG 1-8
1603
INTERNAL RAM
4K x 8
1617
0FFFFH- END IMAGED MEMORY
0F000H
PERIPHERAL/REGISTER OTA LCA MAP
4K X 8
1616
0EFFFH
0E000H
CS2
UNUSED
8K x 8
1615
0DFFFH
0C000H
1602
SRAM
16K X 8
1614
0BFFFH IMAGED MEMORY SEG 0
08000H
CS1
FLASH
4K x 16
(8K X 8)
1613
07FFFH
06000H
CS0
CONFIGURATION/MESSAGE DATA - FLASH
4K x 16
(8K X 8)
1612
05FFFH
04000H
CS0
FLASH
8K x 16
(16K X 8)
1611
03FFFH
00000H-IMAGED MEMORY
CS0
1601

FIG. 16

SEGMENT MAPPING	USE	SEGMENT 9 ADDRESS	SEGMENT 0 ADDRESS
1710	GENERAL SOFTWARE	AFFFFH	
SEG 9-10	80K x 8	9C000H	
1709	GENERAL SOFTWARE	9BFFFH	0BFFFH
SEG 0, 9	13K x 8	98C00H	08C00H
	LINEARIZER BUFFER	98BFFH	08BFFH
1708			
SEG 0	256 x 8	98B00H	08B00H
	TRANSMIT BUFFER	98AFFH	08AFFH
1707			
SEG 0	1K x 8	98700H	08700H
	RECEIVE BUFFER	986FFH	086FFH
1706			
SEG 0	1K x 8	98300H	08300H
	SIM BUFFER	982FFH	082FFH
1705			
SEG 0	256 x 8	98200H	08200H
	COMMAND BUFFER	981FFH	081FFH
1704			
SEG 0	128 x 8	98180H	08180H
	BASE/ FREQUENCY MAP	9817FH	0817FH
1703			
SEG 0	384 x 8	98000H	08000H
	GENERAL SOFTWARE	97FFFH	
1702			
SEG 9	32K x 8	90000H	

1701

FIG. 17

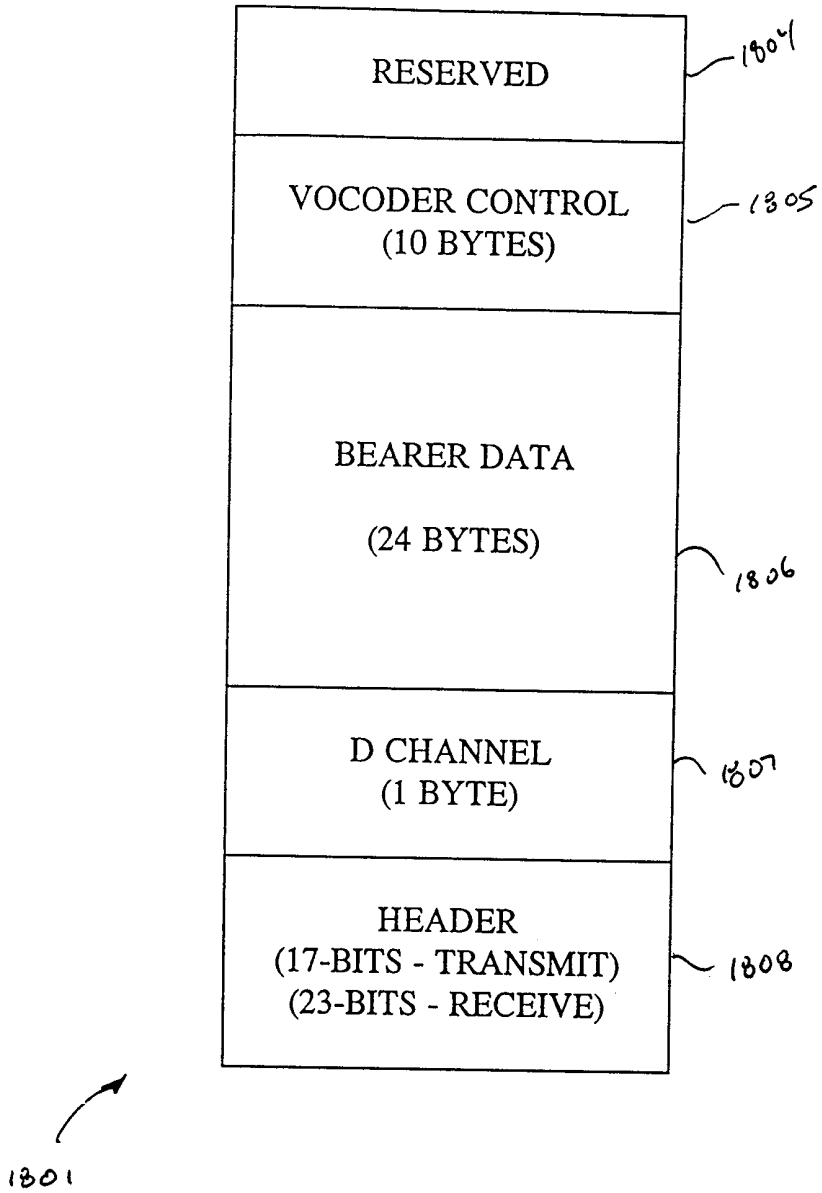

RESERVED
1804
VOCODER CONTROL
(10 BYTES)
1805
BEARER DATA
(24 BYTES)
1806
D CHANNEL
(1 BYTE)
1807
HEADER
(17-BITS - TRANSMIT)
(23-BITS - RECEIVE)
1808
1801

FIG. 18

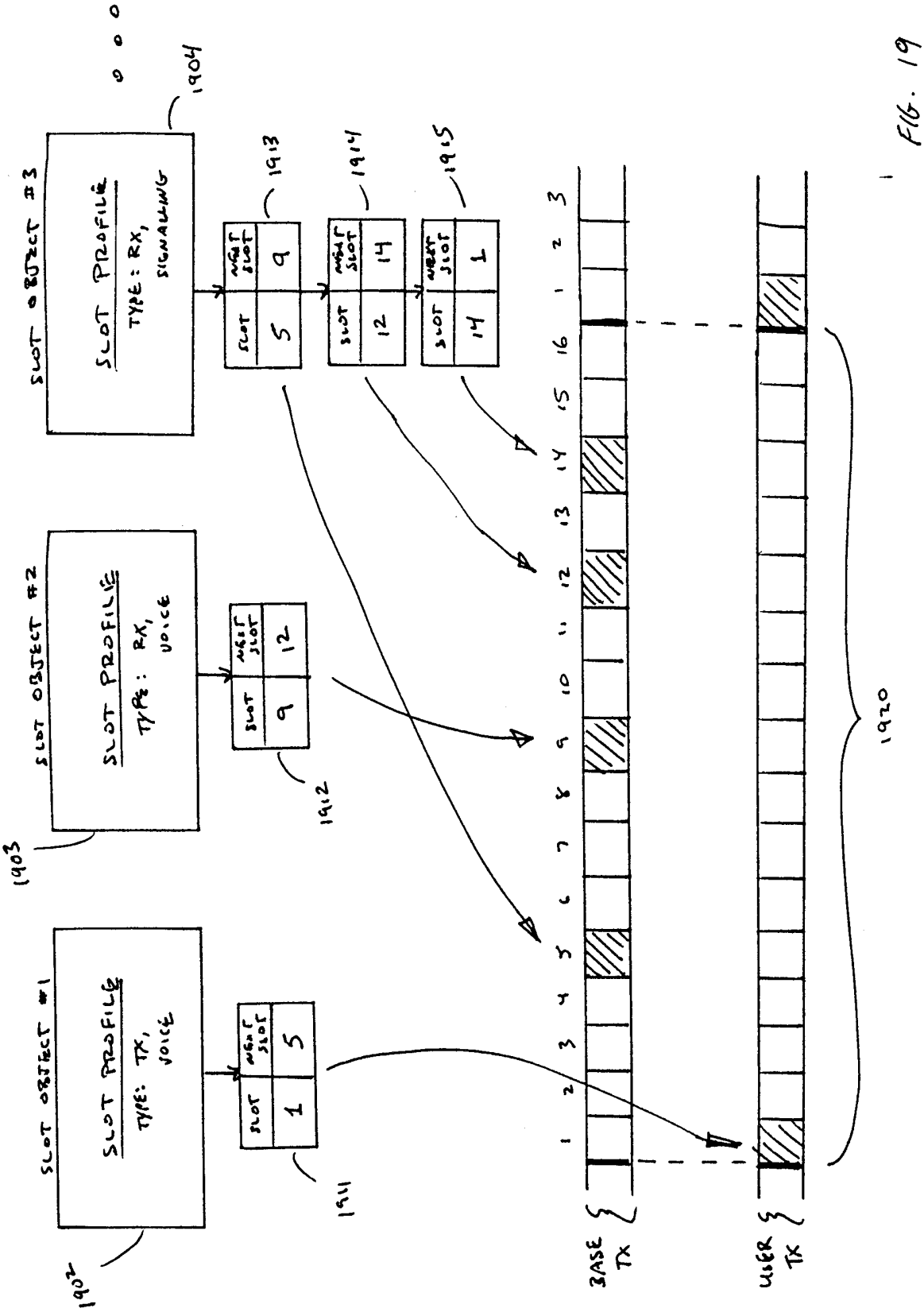

SLOT OBJECT #3
SLOT PROFILE
TYPE: RX, SIGNALING
1904
SLOT OBJECT #2
SLOT PROFILE
TYPE: RX, VOICE
1903
SLOT OBJECT #1
SLOT PROFILE
TYPE: TX, VOICE
1902
SLOT
NEXT SLOT
1911
1912
1913
1914
1915
BASE TX
USER TX
1920

FIG. 19

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/20587

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04J 4/00
US CL :370/330, 335, 337, 436
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)

U.S. : 370/330, 335, 337, 436, 442, 336, 441

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)

WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4,803,681 A (TAKAHASHI) 07 February 1989, (entire document)	1-34
A	US 5,090,013 A (FADEM) 18 February 1992, (entire document)	1-34
A	US 5,805,581 A (UCHIDA et al.) 08 September 1998, (entire document)	1-34

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

*	Special categories of cited documents:	"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
"A"	document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier document published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"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		

Date of the actual completion of the international search 03 JANUARY 2000	Date of mailing of the international search report 08 FEB 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MITCHELL SLAVITT Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet)(July 1992) ★