

DescriptionThe Field of Invention

- 5 The invention disclosed herein is concerned with radio design structures.
 More particularly, this invention relates to two-way, mobile radio design structures.

Background of the Invention

- 10 The need for this invention arose from problems of terribly inflexibly architected radio structures, from complex and unreliable cabling interconnections between radio peripheral components and from the general lack of having human engineering as a design goal.

- Current mobile radio products are designed around subsystem control options where each option is independent in function from one another, yet highly interdependent upon others for command and control.
 15 Accordingly, due to the high degree of interdependence, a total radio system had to be envisioned prior to designing any individual subsystem. Typically, one or two major system components were responsible for orchestrating all of the command and control overhead for the other subsystems. The radio structure was constrained by the hardware, command and control structure envisioned at the design inception of the radio system. Later radio subsystems were difficult to accommodate because the basic structure had already
 20 been cast. Thus, the radio system, once architected, did not allow for convenient integration of new subsystems, features and enhancements.

- Moreover, current mobile radios require fairly complex intercabling among option subsystems which are, often, exclusively located in the control head package. The radio to control head cabling is also complex, costly and unreliable. Each individual switch, control or indicator was separately hard-wired to the
 25 subsystem or option that it was to control. For example, PCT application no WO 8400652 describes a communication system including a mobile radio unit under dedicated MPU control and having numerous external control lines to, for example, the emergency switch, status switches, call, horn, and light switches.

Others have serialised the heretofore parallel link between the control head and the basic radio.

- Still others have register-modelled the entire mobile radio system with respect to the R.F. link (see US-
 30 A-4481670 or US-A-4590473). However, none have internally, serially-bussed every subsystem interconnecting link and internally, register-modelled every device to form a completely new radio architecture.

Current control head packages are physically large (when one or more subsystem option is configured) and often require redundant circuitry for power supply and general interfacing.

- Human interface is lacking due to the multiplicity of generic, cryptically-identified switches, power-
 35 consuming, cryptically-identified indicators and erroneous or non-existent operator feedback and verification.

Mode select systems where the radio may operate in a variety of communication environments and sophisticated data control systems are seriously limited in functional capabilities because of this current inflexible, confusing and complex approach.

- The structure herein proposed, eliminates many of these problems and provides a method for future
 40 expansion with a STRUCTURED subsystem interface. Moreover, the concepts to be presented apply to dash-mount, trunk-mount and remote-controlled mobile radio systems, alike.

BRIEF SUMMARY OF THE INVENTION

- In accordance with the present invention there is provided a method of communicating information to
 45 control the operation of a radio device, characterized by the steps of:

- (a) generating in any of a plurality of register-modelled processor means cooperatively operation within the radio device, an information packet having at least an operation code and an argument,
- (b) transmitting, serially, said information packet to at least one of the plurality of registered-modelled
 50 processor means over a wire-line serial communication link,
- (c) receiving, serially, said information packet from said wire-line serial communication link,
- (d) performing the operation designated by said operation code and said argument.

- The instant invention solves the problem by internally register-modelling and internally bussing the basic radio structure with addressable, "smart" peripheral radio subsystems to allow for inherent flexibility
 55 and expansion.

This invention represents a significant advance over the prior art and over this technical field by providing a reliable, internally register-modelled, addressable, internally bussed mobile radio architecture with logical, human engineered control and operator validation feedback.

The invention allows for distribution of the command and control processing among various radio subsystems, providing them with a greater degree of flexibility and autonomy.

An advantage of the invention is that it provides a radio system capable of having a multiplicity of various radios, functions, features and enhancements in a single, mobile system without unnecessary redundancy of control heads and functional subsystems.

The information packet may comprise an address, an operation code, optional data, and an error detection device, such as a cyclical redundancy check packet. The operation code is chosen from the group of primitive operation codes reset, read, write, bit set, bit clear, acknowledge, and negative acknowledge.

Thus, there has been provided a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback.

Brief Description of the Drawings

Additional objects, features, and advantages, in accordance with the present invention, will be more clearly understood, by way of unrestricted example, from the following detailed description, taken together with the accompanying drawings, in which:

Figure 1 is a block diagram illustrating a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback in accordance with the preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the general internally register-modelled structure that is representative of each subsystem microcomputer.

FIGURE 3 is the serial link communications protocol according to the present invention.

FIGURE 4 is register-modelled synthesized radio according to the preferred embodiment of the present invention.

FIGURE 5 is an exemplary flow diagram of a device driver routine for the synthesized radio of Figure 4.

FIGURE 6 is another register-modelled synthesized radio according to the preferred embodiment of the present invention.

FIGURE 7 is an exemplary flow diagram of a device driver routine for the synthesized radio of Figure 6.

FIGURE 8 is a register-modelled channelized radio according to the preferred embodiment of the present invention.

FIGURE 9 is an exemplary flow diagram of a device driver routine for the channelized radio of Figure 8.

FIGURE 10 is the serial link polling scheme according to the present invention.

FIGURE 11 is an illustration of a conventional control head that is not necessarily fully configured.

FIGURE 12 is an illustration of a human-engineered, fully configured control head according to the preferred embodiment of the present invention.

The invention will be readily appreciated by reference to the detailed description when considered in conjunction with the accompanying drawings in which like reference numerals designate like parts throughout the Figures.

The Description of the Preferred Embodiment

FIGURE 1 is a block diagram illustrating a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback in accordance with the preferred embodiment of the present invention.

FIGURE 2 is a block diagram illustrating the general internally register-modelled structure that is representative of each subsystem microcomputer. This basic structure provides the radio system capability of having a multiplicity of various radios, functions, features and enhancements in a single, mobile system without unnecessary redundancy of control heads and functional subsystems.

The General Register Model

As illustrated in Figure 2, a general approach to radio system design is offered by viewing a radio peripheral 200 as a digital machine that contains a set of defined, addressable registers 210. The contents of the set of registers 210 can then be used to define the operations that a radio device 220 is to perform and the virtual state of the device 220, peripheral, or subsystem 200 at any point in time.

To manipulate the contents of the registers 210 which control the device 220, a small set of general, primitive instructions has been defined. The most primitive are instructions to read or write (alter) the contents of an addressed register 210. These primitive instructions are transmitted over a bus 230 to cause

data to be written into, read from, modified, or tested with respect to any addressed register 240 in the peripheral 200. The primitive instruction set and transmission format remain the same for all peripherals 200; independent of future functional enhancements that may be added to the device 220, peripheral 200, or the various combinations of different peripherals 200 that might comprise future systems (Figure 1).

- 5 Since the functions off the various peripherals 200 are controlled by the contents of its register set 210 and each register's contents 240 can be addressed by a single instruction; combinations of similar primitive functions (macros) can be invoked to perform new major peripheral 200 or device 220 function in parallel. This feature provides the capability for a peripheral 200 to respond to new "commands" without actually implementing new command operation codes in the peripheral 200. The primitive instruction set can remain
10 stable, promoting upward compatibility of peripherals 200.

A conceptual architecture for integrating current and future system options and command/control subsystems with common linkages into a structured unified two-way mobile radio system is further disclosed and illustrated in Figure 1.

15 GENERAL

What is disclosed in Figure 1 is a multi-processor system utilizing a common serial linkage 230 for parametric data transfer and system control. Each major system option or function is independently performed by "local" processors 110 (i.e., peripherals 200) and interfaced to the system on a serial data
20 link 230.

Several definitional terms are used to describe the four basic elements of the system: the CORE processor 120, 140 & 200, the CONTROL processor 150, LOCAL processors 110 and the linkage BUS 230. The minimum system contains the Core processor 200, the Control processor 150 and the Bus 230.

The CORE processor 200 directly interfaces to the radio 220 (Figure 2) and performs many of the low-
25 level tasks 250 (Figure 2) associated with a particular radio. These tasks may include synthesizer/element control 130, transmit power level control, audio muting, channel scanning logic, receive/transmit and transmit/receive sequence timing 260 (Figure 2), subaudible signalling generation and detection, hardware diagnostics, etc. 250 (Figure 2).

The Core processor 200 also provides an interface 270 (Figure 2) to the serial Bus 230 where each of
30 the relevant tasks are available to the Control processor 150 or other Local processors 110 on the Bus 230.

The Core processor 200 is simply a collection of various "drivers" 210/280 (Figure 2) that intimately define the radio interface. The drivers are controlled or scheduled by the Control processor 150 or Local processors 110. The interface to the drivers consists of a structured map 210 which defines the entry and exit parameters 240 necessary to perform the function of the driver 280.

35 As an example consider the driver which changes the receiver's 200 frequency. One radio 200 contains a frequency synthesizer 130 requiring 24 bits of information, control and timing signals and a refresh protocol. Another radio 120 contains channel elements requiring simple element selection logic.

The driver for the first radio 200 is unique and much more complex than the driver for the other radio 120. However, both drivers contain a "common" standard for interface to the system, i.e., the driver
40 parameters 240 (Figure 2) are standard enough to allow use of that driver 280 (Figure 2) by other processors 140 in the system 100.

The Core processor 200 should be as simple as possible to interface to the radio hardware 220 (Figure 2) without an integrated operating system. The Control processor 150 or Local processors 110 provide the necessary logic to control the system 100.

45 The Control processor 150 (i.e., control head) provides the human interface to the system 100 and is common to all radio products 120, 140 & 200. Its primary function is to control the display 160 and keyboard 170 entry of data and control information. It also acts as a system monitor providing status information to the user.

The Control processor 150 also interfaces to the serial bus 230 providing access of its drivers 282 to
50 other processors 110 in the system 100 in a standard fashion. As a control element, it provides the operating system logic for the Core processors 120, 140 & 200 and Local processors 110.

Any and all unique parameters of the system 100, such as frequency information, Unit ID codes 290 (Figure 2), PL/DPL codes, mode linkages, scan lists, etc., are provided to all system peripherals 200, 110, etc. by the Control processor 150. It provides the "data base" of the system 100 and transfers this
55 information over the bus 230. This does not imply that the non-volatile code information 290 (Figure 2) is, necessarily, resident in the Control processor 150. It does imply that any and all relevant code information may be obtained 290 (Figure 2) or modified by the Control processor 150.

An example is provided by the PL/DPL drivers 280 resident in the Core processor 200. The total set of

possible seed codes for PL and DPL may be hard programmed in the Core processor memory. This data base 280 will be accessible to the Control processor 150 for distribution 230 to other Local processors 110 requiring the codes, i.e., Local processors 110 must go "through" the Control processor 150 to obtain the codes. The Control processor 150 may also contain its own data base 282 of parametric information.

5 The keyboard system 170 of the Control processor 150 will be general enough to accommodate simple mode control systems and sophisticated data entry systems. Various levels of entry modes will be available to multiplex the use of the keyboard 170 and display 160. The simple mode controlled systems will use a one-to-one mapping (i.e., so-called "soft" keys) of the keyboard keys 170 for mode selections.

10 Analog control functions, such as, volume and squelch will be controlled from the keyboard 170 in digital form and then converted to analog form by the Core processor 120, 140 & 200.

The display system 160 will include alphanumeric graphics with user prompt (i.e., so-called menu) logic and error control logic with operator feedback. The display system 160 is controlled by the Control processor 150 and is accessible 230 to other Local processors 110 in the system 100 for status and monitoring purposes as well as keyboard 160 entry feedback.

15 The Local processors 110 provide the options 180 and expansions of the system 100. Major communications functions, such as Digital Voice Storage, Telephone signalling, Multi-frequency and Single-frequency trunking, etc., are all Local processor systems 110.

20 Local processors 110 may be mated to the serial bus 230 internal to 190 or external to 180 the radio package 200. Local processors requiring interface to the audio, detector and modulator circuits 190 are located within the radio package 200. However, all control and data is communicated on the serial bus 230.

All Local processors 110 have a common structured interface allowing their use with all radio products 110, 120, 140, & 200 and option subsystems 180 & 190. This approach eliminates the redesign of these major systems for each particular radio configuration, which is the general approach used today. It makes the system 100 "I/O independent."

25 The serial Bus 230 provides the physical interface of all processors 110, 120, 140, 150, 180, 190 & 200 in the system 100. It consists of a 2-wire link (signal and ground) and may be bussed internal 190 to the radio as well as external 180. When used externally, the cable may be twisted-pair, shielded-audio or fiber-optic. The link may also be remoted via infrared, ultrasonic or RF.

30 The communications protocol, illustrated in Figure 3, is designed for simplex operation of synchronous data packets containing source and destination addressing, control, data and error detection codes.

It is desirable that the Control 150, Core 120, 140 & 200 and Local processors 110 perform all of the required signalling required for the Bus 230 interface to reduce the costs of the system 100.

CORE PROCESSOR

35 Returning to Figure 2, the Core processor 200 provides the interface to the radio hardware system 220 and the serial Bus 230. It does so by implementing various drivers 280 that are generic to that hardware configuration.

40 CONTROL DRIVERS

The drivers 280 make up the essential control elements that are necessary to "drive" the commands generated by the Control processor 150 and Local processors 110. In this regard, the drivers 280 are global and accessible to all other elements on the Bus 230.

45 Each driver 280 contains a Bus 230 interface specification that is unique to the function the driver performs and standard among Core processors 120, 140 & 200. An example will illustrate this approach.

Consider the driver used to select the current receive frequency of the radio. The example uses three methods of frequency selection by three unique radio products and their associated Core processors 120, 140 & 200.

50 Referring to Figure 4, the first radio 200 contains a synthesizer 130 requiring 24 bits of raw data for frequency selection. The data is provided by the Core processor 400 in a parallel fashion by way of a 4-bit bidirectional data bus 410 and a 3-bit address bus 420. The Core processor 400 may read or write the synthesizer data by selecting one of the six addressable registers 210 (Figure 1) contained within the synthesizer system and strobing the appropriate data.

55 After writing new data into the synthesizer 130, the Core processor 400 may read the register contents 210 to verify the write. If the write operation was successful, the command is considered complete. If the write was not successful, an error status is reported 230 to the command originator. This system does not require a "refresh" operation by the Core processor 400.

Returning to Figure 2, the driver 280 to select the current receive frequency of the first radio 200 requires two entry parameters and one exit parameter. The entry parameters 240 are the receive frequency name, (e.g., RF-1, RF-2, RF-5, etc.) and the address of the device originating the command. The exit parameter is an error control code that is returned 230 to the device originating the command. Error-0 designates a successful operation, Error-1 designates a faulty write operation and Error-2 designates an improper entry parameter selection.

Upon entry, the driver 280 takes the frequency name and cross-checks that name with available receive frequencies stored in the frequency codeplug 290. If the name is valid, the driver 280 obtains the 24 bits of raw data from the codeplug 290 and attempts to change the receive frequency. If the name is invalid, the driver 280 signals 230 the originating device with an Error-2 parameter 240.

The driver 280 then goes through the procedure to write the receive frequency data to the synthesizer 130. The written data is then read for verification. If the write was successful (data properly loaded and synthesizer locked), the driver signals the originator with an Error-0 parameter. If the write was faulty, an Error-1 parameter is signalled.

If at any time the synthesizer 130 should go out of lock, the Core processor 400 may signal 230 the last originating device with an Error-1 parameter.

The flow diagram of the receive frequency driver for the first radio 200 is shown in Figure 5.

A second radio 140, illustrated in Figure 6, also contains a synthesizer system 130 for frequency selection. Unlike the first radio 200, the synthesizer 130 of the second radio 140 requires a serial load operation with 20 bits of raw data and does not allow the Core processor 400 to read the synthesizer register contents 210. For reliability purposes, the Core processor 400 must refresh the synthesizer 130 periodically.

The Core processor 400 of the second radio 140 must provide the standard entry and exit interface as outlined for the first radio 200 and provide the proper timing and control required for the synthesizer 130. This may include assembly of the data 610 into the proper serial format, generating the serial clock 620, generating start/stop bits, etc.

The driver 280, to select the current receive frequency of the second radio 140, requires the same entry and exit parameters 210 as the first radio 200 (i.e., frequency name and originating device address for entry and error control code for exit). Thus, the instruction sequence and operation codes to be transmitted in the bus communications protocol are identical for all radios 120, 140 & 200.

More particularly, in the second radio 140, upon entry, the driver 400 cross-checks the frequency name with the frequency codeplug 290 and reports an error if the name is invalid. The driver 400 then obtains the 20 bits of raw data from the codeplug 290 and attempts to change the receive frequency by properly formatting the serial data 610 and generating the proper timing signals 620.

Since the loaded data 610 of the second radio 140 may not be read by the Core processor 400, a refresh operation must be performed periodically and an Error-1 exit parameter will be signalled 230 to the originating device only if synthesizer 130 lock 630 is not achieved. Otherwise, an Error-0 code is signalled 230.

The flow diagram of the receive frequency driver for the second radio is illustrated in Figure 7.

Illustrated in Figure 8 is a third radio 120 that does not contain a synthesizer. Frequency generation is accomplished through channel elements 810. The current receive frequency is selected by simply gating on one-or-N elements 810. Figure 9 illustrates the flow diagram for changing the frequency in the third radio 120.

In comparing the drivers of the first 200, second 140 and third 120 radios, note that the Bus interface specification is identical in all three radios. This approach provides the "I/O independence" of the Core processor allowing common access by the Control processor 150 and Local processors 110 within the system 100.

Thus, only the Core processor 120, 140 & 200 need be designed for each new radio product to mate with existing options 110, 150, 180 & 190.

The total set of singular control drivers will vary from radio to radio and need not be restricted to present control functions. For example, the first radio 200 may contain 5 drivers: Select transmit/receive frequency, Select PL/DPL seed code, Select audio muting mode, Select transmit mode and Select receive mode.

The second radio 140 may contain 10 drivers which include all of the drivers of the first radio 200 plus, for example: Select channel scan mode, Update scan list, Select priority channel for scan, Select default scan transmit frequency and Select scan interrupt logic. The second radio 140 would be a superset of the first radio 200.

The Control 150 and Local 110 processors may be designed to accommodate all Core processor

drivers and have the capability to recognize which Core processors are resident in the system 100. Moreover, it is possible, for backward compatibility, that any new Core Processor designs provide the drivers (or equivalents) of previous designs.

One possible set of singular drivers that may be implemented in a Core processor is as follows:

5

SELECT RECEIVE FREQUENCY

SELECT TRANSMIT FREQUENCY

10

SELECT PL/DPL RECEIVE CODE

SELECT PL/DPL TRANSMIT CODE

SELECT AUDIO MUTE MODE

15

ENABLE/DISABLE PL/DPL TRANSMIT/RECEIVE

SELECT AUDIO VOLUME LEVEL

SELECT CARRIER SQUELCH DETECT LEVEL

SELECT TRANSMIT POWER LEVEL

20

ENABLE/DISABLE TRANSMIT CIRCUITS

ENABLE/DISABLE STANDBY POWER MODE

SELECT TRANSMIT TIME-OUT TIME

25

SELECT MODULATION CHANNEL

ENABLE/DISABLE DIAGNOSTICS MODE

UPDATE CHANNEL SCAN LIST

30

SELECT SCAN PRIORITY SAMPLE FREQUENCIES

ENABLE/DISABLE SCAN MODE

ENABLE/DISABLE SCAN INTERRUPT LOGIC

ENABLE/DISABLE SCAN STATUS REPORTING

35

SELECT SCAN TRANSMIT DEFAULT FREQUENCY

ENABLE/DISABLE DIAGNOSTIC STATUS REPORTING

SELECT SCAN SERVICE LOGIC

40

UPDATE CODEPLUG PROGRAM LIST

PROGRAM CODEPLUG

READ CODEPLUG CONTENTS

INITIALIZE SYSTEM

45

ENABLE/DISABLE RECEIVE/TRANSMIT STATUS REPORTING

ENABLE/DISABLE PUBLIC ADDRESS MODE

MACRO COMMAND PROCESSORS

50

MACRO CONTROL DRIVERS

Many of the drivers presented above require parametric data transfer to and from the originating device. The structured Bus 230 allows for addressing and execution of each individual driver. However, to address a driver, a fair amount of data transfer must take place which takes time.

Certain processes require a series of events to take place (i.e., a series of drivers must be executed to complete the process that the device was instructed to perform). As an example, consider switching the radio from receive mode to transmit mode resulting from the operator pressing the Push-To-Talk switch.

The drivers required are: (1) selecting the transmit frequency, (2) selecting the transmit PL/DPL code, (3) selecting the modulation channel, (4) selecting the transmit time-out time, (5) disabling the receive mode, (6) selecting the power output level, (7) enabling the transmit control circuits and (8) enabling the PL/DPL encoder.

5 The Control processor (or Local processor 110) could perform this series of events by signalling 230 the Core processor for each event. However, this would take too much time and would burden the Control processor. Moreover, the series of events may be different from one Core processor to the next, which blemishes the I/O independence.

To alleviate this problem and speed up the process, Macro Control drivers are used. Macro drivers
10 simply link the execution of individual drivers to form a process. The linkage will occur at the microprocessor 400 machine speed rather than at Bus 230 speed.

Macro drivers are split into two modes: setup and execution. The setup mode passes 230 and retains 210 all of the parameter data required to execute the Macro. The execution mode links the individual drivers 280. Processes requiring fast response times may require Macro setup at system initialization.

15 Individual parameters 210 (such as, transmit frequency, PL/DPL code, etc.) may be updated as necessary internal to the Core processor without status reporting to the Control processor 150. One such process would be the switching from receive mode to transmit mode and the converse. Other Macro control drivers of the Core processor are:

20
CHANNEL SCAN PROCESS
PL/DPL DETECTION
STATUS REPORTING
25 SYSTEM DIAGNOSTICS
DIAGNOSTIC DRIVERS

Each element of the system 100, including the Core processor, contains diagnostic drivers to test the
30 integrity of the system. Diagnostic reporting is maintained through the Control processor 150 of "off-line" via specialized test equipment.

Two types of diagnostic feedback are provided to the operator: fatal and non-fatal. A fatal report will result if the Control Head MPU 155 or Control processor 150 fails to acknowledge command signalling 230 or if one of the major hardware elements fail 220. A non-fatal report will occur if one of the Local processors
35 110 fail interrogation by the Control processor 150 or if a minor element of the radio hardware fails.

CORE PROCESSOR ADDRESSING

Referring to Figure 3, each element in the radio system 100 contains a unique address that is primarily
40 used as enable logic. Four bits of data 310 are used for destination processors selection. This will allow up to 15 major peripherals to be resident on a single Bus 230. The three lowest-order addresses (0H through 2H) are reserved for Core processor 120, 140 & 200 addresses.

Thus, up to three Core processors 120, 140 & 200 may reside on a single Bus 230. This allows for multi-radio installations as illustrated in Figure 1. Only one Control processor 150 is necessary to control the
45 operation of three radios, such as High-Band, UHF-450 and UHF-800, for example.

Local processors 110 may be shared among the three radios 120, 140 & 200.

CORE PROCESSOR STATUS REPORTING DRIVERS

50 Drivers 280 are included in the system's many register-modelled processors 400 to report the status of various control signals, such as channel activity, PL/DPL detect, synthesizer lock, standby mode, transmit power, etc.

These register-modelled processors 400 are normally interrogated by the Control processor 150 (or Local processor 110) or may be combined with a Macro driver for continuous monitoring.

55 CODEPLUG PROGRAMMING

It is desirable that nonvolatile parameters 290 of the register-modelled processor 400 (or the rest of the

system) which are normally stored in codeplug 290 be programmable via the Bus 230. The control element may be the Control processor 150 or external equipment.

This will allow personalization of each system 100 to be dynamically and automatically configured via factory computers. Systems 100 may be easily reconfigured in the field and no alternative codeplug 290
5 stocking is required.

ANALOG CONTROL

Audio volume level and carrier squelch circuits are controlled via analog potentiometers in the control
10 head 150 and are normally under user control. A four bit word is output by the Control Head MPU 155 to provide 16 levels of audio volume and 16 levels of squelch attack. A D/A converter following the Control Head MPU 155 will provide a DC level to the volume and squelch circuits 220. The volume D/A is logarithmic and the squelch is linear.

The drivers 280 for volume and squelch will be accessible to the Control processor 150 and Local
15 processors 110. The user will have the capability to set a precise level via the Control processor 150 keyboard 170 or simply ramp-up or ramp-down to a desired level.

Local processors 110 will be able to set fixed volume and squelch levels for selective calling and transmit inhibit logic, respectively.

20 MODULATION CHANNELS

Many of the Local processor systems 110, like Digital Voice Storage, Trunking, Digital Voice Privacy, Queued Community Repeater, etc., are R.F. signalling systems requiring access to the modulator circuits. This system provides four channels for this function internal to the radio package 220. The channels are
25 controlled through the Control Head MPU 155 and assignments are made by the Control processor 150. Local processors 110 may override these assignments.

Three of the four channels are processed by the radio circuits, providing AGC, preemphasis and splatter filtering. The remaining channel is routed directly to the modulator and is normally used for baseband data systems, such as Trunking and Digital Voice Privacy.

30

TRANSMIT POWER LEVEL

This system 100 provides two bits of transmit power level control allowing up to four discrete levels of transmit power. The Control Head MPU 155 controls the interface and assignments are made from the
35 Control Processor 150 or Local processors 110.

POWER STANDBY MODE

Drivers 280 are provided in the register-modelled processor 400 to place the radio hardware in the
40 standby mode to reduce system power consumption. The driver 280 will retain all current parameters 210 of the system 100 and allow a timed transition from full-power to standby. The register-modelled processor 400 will retain power during standby and Local processors 190, within the radio package, have the option of standby control. The control processor 150 (or Local processor 110) may also control the standby mode.

45 CHANNEL SCAN

Channel scan systems require direct interface to the radio hardware 220. The register-modelled processor 400 will provide the necessary drivers 280 to perform the scan logic.

However, the parameters 210 of the scan are provided by the Control processor 150 or a separate scan
50 Local processor 110. The parameters 210 include: non-priority scan list, priority scan list, transmit default frequency, etc. Scan status will be available to the controlling element for further control, such as PL/DPL scan or signalling controlled scan.

PUBLIC ADDRESS MODE

55

The register-modelled processor 400 will have the capability to patch one of the four modulation channels 250 directly to the audio control circuits bypassing the receiver circuits. This will allow Public Address modes that are necessary for certain systems including Digital Voice Storage.

CONTROL PROCESSOR

The Control processor 150 provides the human interface to the system. It does so by way of a keyboard system 170 and display system 160. It also defines the configuration of the system 100.

6

KEYBOARD SYSTEM

The keyboard system 170 is composed of a matrix keyboard and various drivers 282 that interfaces with the Bus 230. The drivers 282 are locally controlled by the Control processor 150/155 and remotely
10 controlled by Local processors 110.

There are two basic keyboard configurations available to the user and are normally system dependent. The dynamic configuration allows for full data entry and option selection via so-called "soft" key identification. The static mode allows for a one-to-one mapping (i.e., so-called "menu" mapping) of the keys to codeplug defined modes.

15 With the static configuration, the operator need only press one or two keys to define the current operating mode of a radio subsystem. The selected mode will define the transmit and receive frequencies, scan list and logic, transmit time-out time, PL/DPL transmit and receive codes, etc. It will also place any Local processors 110 into their "default" run mode.

The controlling parameters of the static mode configuration, that apply to operation, may be resident in
20 both the register-modelled processor 400 and the Control processor 150 and are initialized at system power-up.

The dynamic mode allows all of the capabilities of the static mode plus extended control and data entry. For example, the operator will be able to enter his own priorities and non-priorities for channel scan, select his Unit and Group IDs with the Select Call Local processor 110(A), select any desired muting modes,
25 select a telephone patch 110(B), etc.

Keyboard entry feedback will be provided with tones and display techniques. Complex entries are handled through logical, human engineered prompting (i.e., menus) on the display with error control logic.

Three drivers are used for multiplexed access of the keyboard and display: single key, multiple key and command parsing. The Control processor 150 uses all three drivers and provides access to Local
30 processors 110. The single key and multiple key drivers are available to Local processors 110 for data entry and may bypass the Control processor 150. Command parsing is performed exclusively by the Control processor 150.

DISPLAY SYSTEM

35

The display system 160 provides two major functions: keyboard 170 entry feedback with prompting and system status reporting. The display 160 is a character dot matrix with graphics capability under direct control of the Control processor 150.

The display drivers 282 are used by the Control processor 150 and Local processors 110 to provide
40 entry feedback and status monitoring of the system.

SYSTEM CONTROL

The Control processor 150 is the primary interface and operating system of the Core processor 120,
45 140 or 200 and defines the minimum system. It also controls the interface of Local processors 110 in the system 100. At initialization time, the Control processor 150 interrogates all of the possible board addresses and configures the system 100 to recognize only relevant addresses.

The control processor 150 provides the Bus 230 synchronization and timing.

50 LOCAL PROCESSORS

Local processors 110 provide the options and expansions of system 100 and are mated to the serial Bus 230 internally 190 and externally 180 to the radio package 200.

Each Local processor board 110 contains a programmable address for each major function. Local
55 processor systems 110 may be combined on a single board with the same address and utilize the same bus interface 230.

Local processors 110 communicate with the Control processor 150 and the Core processors as an integrated system. The Bus interface is structured and standard to all Local processors 110 which allow

their use with all radio products 120, 140 & 200 and option systems 180 & 190.

While the Local processors 110 are not stand-alone systems, a system radio may add or delete Local processors 110 from the system 100 without hardware or software modifications. The Control processor 150 will recognize the change. Thus, the radio system 100 becomes installation and I/O independent, increasing the reliability of the communications system 100.

Local processor default parameters 210 always reside with the Local processor hardware and not with the Control processor 150. However, the Control processor 150 may alter various parameters 210 of the Local processors 110 under control of the Local processor 110. For example, consider the Local processor for Select Call 110(A).

The Select Call system 110(A) requires a codeplug 290 to be resident containing the Unit, Group and Fleet Identification codes, muting mode and system configuration. By way of the Control processor 150, the operator may change the Unit and Group ID codes but will not be able to change the Fleet code or configuration. Changing the Unit code does not replace the hard-programmed Unit code 290 but simply provides a "temporary" 210 Unit code by which the radio may be addressed.

It is up to the individual Local processor 110 to decide on the volatility of its parameters.

Local processors 110 that demand system control, such as Trunking 110(C) and Queued community Repeater, may bypass the control of the Control processor 150 and make it a "slave" to the Local processor 110. In this mode, the Control processor 150 will continue to provide the Bus timing and synchronization and operator interface but does not control the system 100.

The Control processor 150 may have the capability to place Local processors 110 on-line or off-line due to a fault in the system or reconfiguration. For example, the operator may mode select 170 which channels are Trunked or Conventional.

The following represents a list of possible Local processor systems:

25

SELECT CALL

DIGITAL VOICE STORAGE

QUICK CALL II

30

TELEPHONE PATCH

MULTI-FREQUENCY TRUNKING

QUEUED COMMUNITY REPEATER

35

SINGLE TONE ENCODE/DECODE

DIGITAL VOICE PRIVACY

ANALOG VOICE SCRAMBLER

40

CONSOLE SYSTEM

DATA TERMINAL

SCAN

LORAN-C LOCATION SYSTEM

45

ZVEI/CCIR SIGNALLING

SERIAL BUS

50

The serial Bus 230 provides the physical interface of all processors 400 in the system. It consists of a 2-wire link (signal and ground) and may be bussed internal 190 to the radio as well as external 180.

SYNCHRONOUS DATA

55

The system Bus 230 is fully synchronous using baseband data at 2400 bits per second. No clock signal is required since clock recovery is performed by each of the system processors. (Due to the speed requirements of the system and the response times necessary for various commands, a preprocessor may

be required for some of the Local processors 110. The preprocessor is a simple 4-bit machine that performs all of the necessary signalling to interface with the serial bus 230. This includes signalling intensive systems like Select Call 110(A) and Trunking 110(C).)

5 CONTROL PROCESSOR POLLING

Referring to Figure 3, signalling among processors 400 in the system 100 is accomplished through variable sized data packets. The data packets are asserted on the bus asynchronously or by way of a polling format generated by the Control processor 150.

10 Each data packet consists of a 12-bit sync code, 4-bit source address, 4-bit destination address, 4-bit byte count, 8-bit operation code, from 0 to 15 data bytes and an 8-bit Cyclically Redundancy Check (CRC) code.

The sync code and source address are generated by the control processor with the poll format found in Figure 10. The dead period is 4 bits long and is the zone where an asserting processor 400 will place the destination address. Upon recognizing a valid address in the dead zone, the Control processor 150 will
15 relinquish control of the bus 230 and resume control at the end of the data packet CRC code, where the poll will continue.

CONTROL PROCESSOR PRIORITY

20

Since the Control processor 150 controls the poll addressing (or sequencing), it may assert control at any time and establish priority over other processors in the system 100. This feature allows fast response times to operator inputs, such as Push-To-Talk and mode selections, etc.

25 COMMAND ACKNOWLEDGEMENT

All commands issued by the Control processor 150 or Local processors 110 are acknowledged in this system 100. The acknowledge packet must occur within the next poll cycle or retransmission will result. The number of retransmissions will be limited and the operator will be informed of a fault if the destination
30 processor 400 fails to respond.

DYNAMIC ADDRESSING

Each of the Local processors 110 and the Control processor 150 contain a hard-programmed board
35 address that is used in select logic during the poll sequence. At power-up time, the Control processor 150 interrogates all of the possible address values to determine the poll sequence. The Control processor 150 may alter the board address dynamically to conform to a sequential poll. A board is deselected from the system by excluding its address from the poll.

40 HARDWARE

Some consideration should be given to the hardware requirements of the system. Since it is a multi-processor 400 system to be used in a mobile environment, it is recommended that low power microprocessors be used and internally "bus" oriented systems be avoided. This limits the design to single-chip CMOS
45 processors with today's technology.

The Bus interface circuit 270 should be designed for fail-soft and not load the Bus 230 if a local failure occurs. The system may continue to function, for example, if one of the Local processors 110 fail.

The D/A interface used with the volume and squelch (and power level) functions should be standard from radio to radio so that the same data word will cause the same output level.

50 The display 160 and keyboard 170 functions of the Control processor 150 should be stand-alone systems so that the Control processor 150 is not burdened by display refresh or keyboard scanning, etc.

The radio package 120, 140 & 200 should provide power switching capability if the standby function is to be used.

The Bus receiver interface circuit 270 should provide sufficient gain to compensate for voltage offsets
55 but not too much gain to cause noise when no signal is present. This may cause unnecessary falsing in the system.

Any external control elements of the processor systems should be "latched" so that the primary task is monitoring the serial Bus 230 for data. The acknowledge function will allow certain tasks to miss the poll

address. However, the number of retransmissions are limited and should be considered in the design.

The Bus receiver logic of all processors will employ PLL clock recovery techniques to provide mid-bit sampling of the data. This clock is also used for asserting data on the Bus and care should be taken to provide "smooth" bit transitions without "glitches." Also, the Bus data is baseband requiring proper phasing of the receiver.

MECHANICS

The mechanics of the Control processor system 150 (i.e., the control head) will be intimately related to the electronics. The package should be as small as possible with enough room to accommodate the display 160 and keyboard 170. The Bus 230 interconnect scheme should allow for expansion with Local processors 110 that are external 180 to the radio 120, 140 & 200. A significant amount of styling is required.

Thus, there has been provided a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback.

Further, there has been provided a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback that:

- provides an incredibly flexibly architected radio structure;
- distributes the command and control processing among various radio subsystems, providing them with a greater degree of flexibility and autonomy;
- provides simple and reliable cabling interconnections between radio peripheral components;
- employs a "smart" control head that may be multiplexed with each functional subsystem to facilitate operator input, output and feedback required for the command and control of each subsystem;
- provides a radio system capable of having a multiplicity of various radios, functions, features and enhancements in a single, mobile system without unnecessary redundancy of control heads and functional subsystems; and
- provides logical, human engineered system control and operator validation feedback.

It will be appreciated by those skilled in the art that this register-modelled invention may be practiced by making design tradeoffs between the complexity and simplicity of the operation code set and the register modelling. The register set can be virtually eliminated, as such, and simulated in software variables with an extensive and particularized operation code set - much larger and richer than the read, write and acknowledge set of primitives discussed above. However, some of the I/O independence is lost through this compromised, but expedient approach.

For example, rather than simply writing data to be displayed into a register as a data argument to a write op-code, a new op-code, defined as write-to-display could be defined with a data argument that would simulate the same function. However, some of the I/O independence is lost because the new write-to-display op-code is not an op-code that is generic to the entire system, but only has meaning to a display device. Thus, there is a design tradeoff between simplicity and primitiveness of the operations codes or complexity and richness and I/O independence. But, in either case, the objects of the present invention are met.

An intermediate approach uses the primitive set of op-codes, namely: reset, read, write, bit clear, bit set, acknowledge and negative acknowledge, but then adds more, yet somewhat generic op-codes, namely: request display, show display, short register is, long register is, name is, and display is for commonly used display and status functions, and read a short register, read a long register, write to a short register, write to a long register, update a short register, update a long register, not unlike multiple precision numbers, in order to more efficiently handle multibyte strings. In this type of register-modelling, registers are used as virtual representations of a physical device, only at a more fundamental level. The register models a physical component (e.g., the display) rather than the entire peripheral device (e.g., the control head). However, the underlying concept is the same.

An even greater tradeoff is presented when the primitive set of operation codes is splintered into a rich variety of nearly discrete op-codes that implement functions very closely associated with the physical device itself.

Claims

1. A method of communicating information to control the operation of a radio device, characterized by the steps of:
 - (a) generating in any of a plurality of register-modelled processor means (400, 200, 110A-C, 120,



address. However, the number of retransmissions are limited and should be considered in the design.

The Bus receiver logic of all processors will employ PLL clock recovery techniques to provide mid-bit sampling of the data. This clock is also used for asserting data on the Bus and care should be taken to provide "smooth" bit transitions without "glitches." Also, the Bus data is baseband requiring proper phasing of the receiver.

MECHANICS

The mechanics of the Control processor system 150 (i.e., the control head) will be intimately related to the electronics. The package should be as small as possible with enough room to accommodate the display 160 and keyboard 170. The Bus 230 interconnect scheme should allow for expansion with Local processors 110 that are external 180 to the radio 120, 140 & 200. A significant amount of styling is required.

Thus, there has been provided a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback.

Further, there has been provided a reliable, internally, register-modelled, addressable, internally, serially-bussed mobile radio architecture with logical, human engineered control and operator validation feedback that:

- provides an incredibly flexibly architected radio structure;
- distributes the command and control processing among various radio subsystems, providing them with a greater degree of flexibility and autonomy;
- provides simple and reliable cabling interconnections between radio peripheral components;
- employs a "smart" control head that may be multiplexed with each functional subsystem to facilitate operator input, output and feedback required for the command and control of each subsystem;
- provides a radio system capable of having a multiplicity of various radios, functions, features and enhancements in a single, mobile system without unnecessary redundancy of control heads and functional subsystems; and
- provides logical, human engineered system control and operator validation feedback.

It will be appreciated by those skilled in the art that this register-modelled invention may be practiced by making design tradeoffs between the complexity and simplicity of the operation code set and the register modelling. The register set can be virtually eliminated, as such, and simulated in software variables with an extensive and particularized operation code set - much larger and richer than the read, write and acknowledge set of primitives discussed above. However, some of the I/O independence is lost through this compromised, but expedient approach.

For example, rather than simply writing data to be displayed into a register as a data argument to a write op-code, a new op-code, defined as write-to-display could be defined with a data argument that would simulate the same function. However, some of the I/O independence is lost because the new write-to-display op-code is not an op-code that is generic to the entire system, but only has meaning to a display device. Thus, there is a design tradeoff between simplicity and primitiveness of the operations codes or complexity and richness and I/O independence. But, in either case, the objects of the present invention are met.

An intermediate approach uses the primitive set of op-codes, namely: reset, read, write, bit clear, bit set, acknowledge and negative acknowledge, but then adds more, yet somewhat generic op-codes, namely: request display, show display, short register is, long register is, name is, and display is for commonly used display and status functions, and read a short register, read a long register, write to a short register, write to a long register, update a short register, update a long register, not unlike multiple precision numbers, in order to more efficiently handle multibyte strings. In this type of register-modelling, registers are used as virtual representations of a physical device, only at a more fundamental level. The register models a physical component (e.g., the display) rather than the entire peripheral device (e.g., the control head). However, the underlying concept is the same.

An even greater tradeoff is presented when the primitive set of operation codes is splintered into a rich variety of nearly discrete op-codes that implement functions very closely associated with the physical device itself.

Claims

1. A method of communicating information to control the operation of a radio device, characterized by the steps of:
 - (a) generating in any of a plurality of register-modelled processor means (400, 200, 110A-C, 120,



- 140, 180) cooperatively operation within the radio device (100), an information packet (310) having at least an operation code and an argument,
- (b) transmitting, serially, said information packet (310) to at least one of the plurality of registered-modelled processor means over a wire-line serial communication link (230),
- 5 (c) receiving, serially, said information packet from said wire-line serial communication link (230),
- (d) performing the operation designated by said operation code and said argument.
2. A method as claimed in claim 1 wherein said register-modelled processor means have predetermined addresses and wherein said generating and performing steps are further characterized, respectively, by
- 10 the steps of:
- generating an information packet (310) further having an address corresponding to another register-modelled processor means, and
- performing said operation when said address corresponds to a receiving register-modelled processor mean's predetermined address.
- 15 3. A method as claimed in claim 1 wherein said generating and said performing steps are further characterized, respectively, by the steps of:
- generating an information packet (310) further having a register address, and
- performing said operation upon information in an addressed register when said address corresponds to a register defined within a receiving register-modelled processors means register set.
- 20 4. A method as claimed in claim 1 wherein said generating and performing steps are further characterized, respectively, by the steps of:
- 25 generating an information packet (310) further having a register address and having data as an argument to said operation code, and
- performing said operation upon said data in an addressed register when said address corresponds to a register defined within a receiving register-modelled processor mean's register set.
- 30 5. A method as claimed in claim 1 wherein said operation code is a primitive operation code chosen from the group of primitive operation codes: reset, read, write, bit set, bit clear, acknowledge, and negative acknowledge.
- 35 6. A method as claimed in claim 1 wherein said information packet (310) further comprises an error detection device.
7. A method as claimed in claim 6 wherein said error detection device further comprises a cyclical redundancy check (CRC) packet.
- 40 8. A radio device, characterized by:
- a plurality of register-modelled processor means (400, 200, 110A-C, 120, 140, 180) having addressable register means for modelling the virtual state of said register-modelled processor means;
- communication means, comprising a wire-line serial bus (230), interconnecting said register-modelled processor means, for communicating between said addressable register means; and
- 45 a communication protocol for passing information to or from said addressable register means, comprising an information packet,
- whereby the virtual state of said register-modelled processor means may be determined or altered by, respectively, communicating information from or to said register means.
- 50 9. A radio device according to claim 8 wherein said information packet comprises
- an address;
- an operation code, further comprising:
- a primitive operation code chosen from the group of primitives: reset, read, write, bit set, bit clear, acknowledge and negative acknowledge;
- 55 optional data; and
- an error detection device, further comprising:
- a cyclical redundancy check packet.



Revendications

1. Un système de radiocommunication comprenant :
au moins deux parties de système de radiocommunication interdépendantes (110, 150, 200), un
5 bus de données (230) connecté à chacune de ces parties de système de radiocommunication, et un
dispositif de commande central (150) qui est connecté au bus de données,
caractérisé en ce que le système de radiocommunication est modélisé par registres, dans la
mesure où chaque partie de système de radiocommunication comprend un registre dans le but de
10 déterminer ou de modifier l'état virtuel de cette partie du système de radiocommunication conformément à des données de paramètres qui lui sont transmises, et un processeur local (110, 400) pour
commander le fonctionnement de cette partie du système de radiocommunication, de façon pratique-
ment indépendante des autres parties du système de radiocommunication, sous la dépendance de
données qui sont reçues dans le registre de cette partie,
et en ce que le dispositif de commande central (150) transmet des données de paramètres à partir
15 des registres ou vers ceux-ci, par l'intermédiaire du bus de données, pour la commande d'ensemble
des parties de système de radiocommunication.
2. Un système de radiocommunication selon la revendication 1, dans lequel l'un des processeurs locaux
(110) commande directement un premier émetteur-récepteur radiofréquence (220), tandis qu'un autre
20 (110) est un système de transmission de signaux RF qui détermine directement les paramètres de
transmission de signaux RF du premier émetteur-récepteur radiofréquence.
3. Un système de radiocommunication selon la revendication 2, dans lequel l'un des processeurs locaux
commande directement un second émetteur-récepteur radiofréquence (120, 140), et l'autre (110)
25 détermine directement des paramètres de transmission de signaux RF des premier (220) et second
(120, 140) émetteurs-récepteurs radiofréquence.
4. Un système de radiocommunication selon l'une quelconque des revendications précédentes, dans
lequel l'une des parties de système de radiocommunication est une tête de commande (150) ayant une
30 visualisation (160), un clavier (170) et un processeur ayant un registre connecté au bus de données.
5. Un système de radiocommunication selon l'une quelconque des revendications précédentes, dans
lequel l'ensemble de parties interdépendantes comprend de multiples modems fonctionnant en radio-
fréquence.
- 35 6. Un système de radiocommunication selon l'une quelconque des revendications précédentes, dans
lequel l'une des parties de l'ensemble des parties interdépendantes consiste en un dispositif périphé-
rique pour un système de radiocommunication mobile.
- 40 7. Un système de radiocommunication selon l'une quelconque des revendications précédentes, dans
lequel le bus de données est un bus série.
8. Un système de radiocommunication selon l'une quelconque des revendications précédentes, compre-
nant en outre un protocole de communication pour transmettre des données de paramètres aux
45 processeurs locaux et au dispositif de commande central, ou à partir de ces derniers, comprenant un
paquet d'information (310) qui contient :
une adresse,
un code d'opération fondamental qui est choisi dans le groupe suivant : restauration, lecture,
écriture, instauration de bit, restauration de bit, acquittement et acquittement négatif, et
50 des données facultatives
grâce à quoi l'état virtuel d'une partie peut être déterminé ou modifié, respectivement par la
transmission de données de paramètres à partir du registre adressable de cette partie, ou vers ce
registre.
- 55 9. Un système de radiocommunication selon la revendication 8, dans lequel le paquet d'information
contient en outre des données de détection d'erreur qui consistent en un paquet de contrôle par
redondance cyclique.



Patentansprüche

1. Verfahren zum Übermitteln von Information zur Kontrolle des Betriebs eines Funkgeräts, gekennzeichnet durch folgende Schritte:
 - 5 (a) Erzeugen in irgendeinem einer Mehrzahl von registermodellierten Prozessorvorrichtungen (400, 200, 110A-C, 120, 140, 180), die in dem Funkgerät (100) Zusammenarbeiten, eines Informationspakets (310) mit wenigstens einem Operationscode und einem Argument,
 - (b) Serielles Übertragen des Informationspakets (310) zu wenigstens einem von der Mehrzahl der registermodellierten Prozessorvorrichtungen über eine serielle Drahtleitungs-Kommunikationsverbindung (230),
 - 10 (c) Serielles Empfangen dieses Informationspakets von der seriellen Drahtleitungs-Kommunikationsverbindung (230),
 - (d) Durchführen der durch den Operationscode und das Argument bezeichneten Operation.
- 15 2. Verfahren nach Anspruch 1, wobei die registermodellierten Prozessorvorrichtungen vorgegebene Adressen besitzen und wobei die Erzeugungs- und Durchführungsschritte jeweils weiterhin durch folgende Schritte gekennzeichnet sind:

Erzeugen eines Informationspakets (310), das außerdem eine Adresse besitzt, die einer anderen registermodellierten Prozessorvorrichtung entspricht, und

- 20 Durchführen der Operation, wenn die Adresse der vorgegebenen Adresse einer empfangenden, registermodellierten Prozessorvorrichtung entspricht.
3. Verfahren nach Anspruch 1, wobei die Erzeugungs- und Durchführungsschritte jeweils weiterhin durch folgende Schritte gekennzeichnet sind:
 - 25 Erzeugen eines Informationspakets (310), das außerdem eine Registeradresse besitzt, und
- Durchführen der Operation basierend auf der Information in einem adressierten Register, wenn die Adresse einem in einem Registersatz in einer empfangenden, registermodellierten Prozessorvorrichtung bestimmten Register entspricht.
- 30 4. Verfahren nach Anspruch 1, wobei die Erzeugungs- und Durchführungsschritte jeweils weiterhin durch folgende Schritte gekennzeichnet sind:

Erzeugen eines Informationspakets (310), das außerdem eine Registeradresse und Daten als ein Argument für den Operationscode besitzt, und

- 35 Durchführen der Operation auf der Basis der Daten in einem adressierten Register, wenn die Adresse einem in einem Registersatz in einer empfangenden, registermodellierten Prozessorvorrichtung bestimmten Register entspricht.
5. Verfahren nach Anspruch 1, wobei der Operationscode ein primitiver Operationscode ist, der aus der folgenden Gruppe primitiver Operationscodes ausgewählt ist: Zurücksetzen, Lesen, Schreiben, Bitsetzen, Bitlöschen, Bestätigen und negatives Bestätigen.
- 40 6. Verfahren nach Anspruch 1, wobei das Informationspaket (310) außerdem eine Fehlererkennungsvorrichtung umfaßt.
- 45 7. Verfahren nach Anspruch 6, wobei die Fehlererkennungsvorrichtung ein zyklisches Redundanzüberprüfungspaket (CRC) umfaßt.
8. Funksystem gekennzeichnet durch:

eine Mehrzahl von registermodellierten Prozessorvorrichtungen (400, 200, 110A-C, 120, 140, 180)

- 50 mit adressierbaren Registervorrichtungen zum Nachbilden des virtuellen Zustands der registermodellierten Prozessorvorrichtungen;

Kommunikationsvorrichtungen mit einem seriellen Drahtleitungsbus (230), der die registermodellierten Prozessorvorrichtungen verbindet, zum Kommunizieren zwischen den adressierbaren Registervorrichtungen; und

- 55 ein Kommunikationsprotokoll mit einem Informationspaket zum Weiterleiten von Information zu oder von den adressierbaren Registervorrichtungen

wodurch der virtuelle Zustand der registermodellierten Prozessorvorrichtungen durch Übertragen von Information von oder zu den Registervorrichtungen bestimmt oder geändert werden kann.



9. Funksystem nach Anspruch 8, wobei das Informationspaket umfaßt:
einen Adresse;
einen Operationscode, der weiterhin umfaßt:
einen primitiven Operationscode, der aus folgender Gruppe primitiver Operationscodes ausgewählt
5 ist: Zurücksetzen, Lesen, Schreiben, Bitsetzen, Bitlöschen, Bestätigen und negatives Bestätigen.
Operationsdaten; und
eine Fehlerfeststellungsvorrichtung, die weiterhin umfaßt:
ein zyklisches Redundanzprüfpacket.

10

15

20

25

30

35

40

45

50

55