

- 1 -

COMPUTER SOFTWARE CONTROL
AND COMMUNICATION SYSTEM AND METHOD

FIELD OF THE INVENTION

5 The present invention relates generally to a computer software control and communication system for computer systems communicating via a communications bus.

BACKGROUND OF THE INVENTION

10 Communication buses allow for the transfer of data among various processing and data storage mediums. Typical communication buses, such as for example, the Versa Module Eurocard 64 (VME64) bus, allow a plurality of hardware circuitry cards to be attached to a computer controllable backplane in order to allow communication to flow among the various hardware circuitry cards and the computer. These circuitry
15 cards are conventional circuit boards containing electronic means for executing predetermined functions or a predetermined software program, such as for example, setting clock frequency, data transfer, data storage and executing test functions.

 Communication buses are frequently employed by embedded systems. An
20 embedded system can take on many forms, but is generally known by those of skill in the art to include a system in which a computer is built into the system, yet the user does not view the system as including the use of a computer. This can occur, for example, in a television set having one or more computer processors, yet the person using the television set does not realize the set includes such computers. An
25 embedded system also includes a special purpose computer and control circuitry built into a device which is programmed for a particular purpose. Such systems include what are known as "embedded processors" or "embedded microcontrollers." Some examples of embedded systems include flight management systems, weapons systems, aircraft control systems, and avionics systems for the aerospace industry;
30 manufacturing automation systems, manufacturing robots, chemical processing plants and refineries for the process control industry; car management systems for the

automotive industry; and switches, routers and PBX systems for the communications industry.

When new or revised circuit cards are added to an existing system having a communications bus, often times it is difficult to communicate with the card. Sometimes
5 this communication difficulty is due to misdirected software handshakes, faulty data transfers and in some cases, may be due to mismatched technology. In these cases, the software in the embedded system and perhaps the software in other cards must be reprogrammed to communicate with the new or revised circuit card. This process is time
10 intensive, increases the complexity of the system, is costly and usually results in expending further development resources. Due to this deficiency in the prior art, a need exists for a common computer software control and communication system which allows a primary software means to communicate with new or revised circuit cards as needed. Such a system, as disclosed by the present invention, would reduce the design time and reduce the resources typically used when new or revised circuit cards are introduced into an
15 existing system.

The closest prior art references to the present invention are U.S. Patent No. 5,764,505 and U.S. Patent No. 5,995,513.

U.S. Patent No. 5,764,505 to Mixon et al., is a laser control system with multiple CPU's having shared memory on a common bus (STD) to provide high performance,
20 relatively low cost, laser control. This patent discloses an electrical connection to a standard bus independent of software structure and software control. This patent does not disclose the use of any software communication structures.

BRIEF SUMMARY OF THE INVENTION

25 The following summary of the invention is provided to facilitate an understanding of some of the innovative features unique to the present invention, and is not intended to be a full description. A full appreciation of the various aspects of the invention can be gained by taking the entire specification, claims, drawings, and abstract as a whole.

The present invention relates to a computer system and method for facilitating bi-directional communication between a primary software means on a computer and at least one secondary software means, loaded on a computer circuit card, via a communications
5 bus. Both the primary software means and the secondary software means both have predefined register storage space for storing data, and in particular, for storing data relating to system and fault status, among other things.

The present invention provides a simple, yet flexible system and method for
10 communication between a primary processing means and secondary processing means which transfer data via a communications bus. The present invention finds

applications in systems having hardware boxes (or "cages") that support dynamic reconfiguration of functionality every time the system is rebooted. In this fashion, upon each reboot of the system, the software means for directing can detect and communicate with each card in the system and adjust the functionality of each card accordingly if
5 necessary to thereby provide a synergistic communications system.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying figures, in which like reference numerals refer to identical or functionally-similar elements throughout the separate views and which are incorporated in
10 and form part of the specification, further illustrate the present invention and, together with the detailed description of the invention, serve to explain the principles of the present invention.

FIG. 1 is a graphical representation of the present invention;
15

FIG. 2 is a graphical representation of the header register of the present invention;

FIG. 3 is a graphical representation of the communications header register of the present invention; and
20

FIG. 4 is a graphical representation of the run-time operation of the present invention.

- 4 -

DETAILED DESCRIPTION OF THE INVENTION

As seen in FIG. 1, the present invention 10 is a computer system and method for allowing communication to take place between primary (or director) software and at least one secondary (or non-director) software program through a communications data bus by a series of unique header registers stored in memory. The primary program is responsible for the control of the overall system including, without limitation, startup, modes of operation, and system fault status reporting and logging. The secondary software program refers to all other modules in the system that are capable of being controlled by the primary. Each such module includes a header for system level control and status reporting.

In particular, the present invention 10 includes at least one first means for processing data 11 in communication with (e.g., electrically, optically, frequency transmissions, etc.) a computer 9 and having a first means for storing data 17 thereon, at least a secondary means for processing data 31_n (where n equals the number of means for processing) in communication with a computer circuit card 41_n (where n equals the number of cards) having a secondary means for storing data 39_n (where n equals the number of secondary means for storing) and in communication with the computer 9 by an arbitrary asynchronous or synchronous communications bus 51, software means for directing 13 stored on and executed upon the first means for processing data 11 and software means for communicating data 33 stored on and executed upon the secondary means for processing data 31_n .

As seen in FIG. 2, the software means for directing 13 further includes a header register 12 which is stored on the first means for processing 11 and is optionally at least a thirty two bit by four word data register, and is preferably a thirty two bit by seven word data register. In some instances, the header register is a required component where control of the non-director modules by the primary program (director) is desired as will become apparent to those skilled in the art. The first four words of the data register of header register 12 are used to store and transmit data relating to the status information of the first means for processing data 11 (including, for example,

- 5 -

identification data, software revision data, physical or logical slot data, operational data, functional interface data, etc.) and for storage and transmission of data relating to any faults (or, errors) detected by the first means for processing 11. The common header interface can include a header and address pointers for a command linked list interface, status linked list interface, and a functional interface.

In an embodiment of the present invention, the common header is used by the primary software program to provide top level module mode control and to read fault status information from the secondary software program modules. The common header resides, for example, at the VME base address for all secondary software program modules. The common header structures are shown in FIGS. 2 and 3.

If additional functionality is required, the header register 12 requires additional memory storage beyond four thirty two bit words, and preferably includes a thirty two bit by seven word data register. Thus, in the additional three thirty two bit word registers, at least one primary software means for linking 19 is stored. Primary software means for linking 19 is a dynamic variable pointer means (or, linked list) which points to any record, data structure or memory register where data can be stored, whether on the first means for processing 11 or any other processing means (such as, for example, any of the secondary means for processing data 31_n) in communication with the communications bus 51. In a preferred embodiment of the present invention 10, the linked list 19 will store information relating to extended functionality status, extended functionality information and command or actions requests, but can store other pointer information as well. Means for linking 19 is not required to contain a pointer to another register, but rather, can contain a null value. Upon system start up (or, boot up), the means for directing 13 will initialize the header register 12 and maintain knowledge of its contents. Thus, as those of skill in the art will realize, the use of software pointers allows the software to represent complex programming structures while permitting flexibility in software design, and can, for example, utilize memory efficiently (e.g., as desired, memory storage can be created for the pointer as the software means for directing is executed and can be used to store complex software instructions).

- 6 -

Further, those of skill in the art will now come to realize that the software means for directing is a dynamic software means which can be stored and executed upon any of the processing means connected to the communications bus 51. This is especially useful should the first means for processing data 11 fail or experience undesirable errors (such as, for example, if the first means for processing data 11 is damaged). In this case, the software means for directing 13 can then dynamically transfer itself to any other processing means (such as, for example, any of the secondary means for processing data 11) on communications bus 51 and continue to communicate with all software means for communicating data 33. Thus, the term "first" in the first means for processing data 11 is not a term of limitation to only one processing medium.

As seen in FIG. 3, the software means for communicating data 33 further includes a communications header register 35 which is stored on the secondary means for processing 31_n and is optionally at least a secondary thirty two bit by four word data register, and is preferably a secondary thirty two bit by seven word data register although most registers can be adapted for use with the present invention as will be apparent to those skilled in the art. In some instances, the communications header register 35 is a required component where control of the modules by the primary program is desired as will become apparent to those skilled in the art (i.e., to be controlled by the primary software program). The software means for directing 13 communicates primarily with the communications header register 35 via bus 51. In this fashion, the electronics on each circuit card 41_n can communicate with the software means for directing 13 and debugged if necessary without the software means for directing 13 having to understand the functionality of each card 41_n. As such, as long as each circuit card 41_n contains a secondary means for processing data 31_n having a secondary means for storing data 39_n (in which communications header register 35 is stored) and can communicate with communications bus 51, software means for directing 13 can always obtain any data relating to each card's functionality, even if each card 41_n is functionally different than any other card 41_n. The first four words of the secondary data register of communications header register 12 are used to store and transmit data relating to the status information of its corresponding secondary means for processing 31_n and for storage and transmission of data relating to any faults

- 7 -

(or, errors) detected by the corresponding secondary means for processing 31_n. If additional functionality is required, the communications header register 35 requires additional memory storage beyond four words, and preferably is a secondary thirty two bit by seven word data register. In the additional three word registers, a secondary software means for linking 37 is stored. Each secondary software means for linking 37 is a dynamic variable pointer means which points to any record, data structure or memory register where data can be stored, whether on the first means for processing 11 or any other processing means (such as, for example, any of the secondary means for processing data 31_n) in communication with the communications bus 51. In a preferred embodiment of the present invention 10, the linked list 37 stores information relating to extended functionality status, extended functionality information and command or actions requests, but can store other pointer information as well. Means for linking 37 is not required to contain a pointer to another register, but rather, can contain a null value. If the means for directing 13 detects a null pointer, the means for directing 13 will recognize that the corresponding circuit card 41_n does not contain extended functional capabilities. If a data pointer is detected by the means for directing 13, the means for directing 13 will appropriately follow the location or address in memory of the storage register pointed to. Thus, as will become apparent to those of skill in the art, the use of software pointers (or, linked lists) permits flexibility in software design, and can, for example, utilize memory efficiently (e.g., memory storage can be created for the pointer as the software means for directing is executed).

By way of example and not by limitation, the following types of equipment can be used in the present invention. The first means for processing data 11 (and the secondary means for processing data 31) could be any microprocessor or microcontroller (commercial or proprietary), such as Intel microprocessors (Pentium, Pentium II, Pentium III, Xeon, Merced, etc.), Motorola PowerPC microprocessors (603, 604, G3, G4, etc.), MIPS microprocessors, Sun SPARC microprocessor, IBM RS/6000 processors, etc. First means for storing data 17 (and the secondary means for storing data 39) includes volatile RAM such as EDO, SRAM; non-volatile ram (NVRAM); programmable flash memory; or even disk-based memory. Computer 9 can be a conventional computer or even a circuit card with computing functions or any other

- 8 -

component having computing capabilities. Generally, a computer is defined by its components e.g., the processing means and data storage means. Software means for directing 13 and software means for communicating data 33 can be software programs that implements the primary and secondary software programs (director and non-
5 director) functionality for the common header architecture as described herein or can be implemented in hardware or a combination of both. Similarly, header register 12 and communications header register 35 can be software programs or can be implemented in hardware or a combination of both. Software means for linking 19 and secondary software means for linking 37 is a well-known standard that is typically a
10 software data structure referred to as a linked list but any other compilation of information can work as well depending on the application.

Communication bus 51 can be any conventional interconnection data bus. More particularly, any bus that is compatible with a random accessible memory scheme can
15 be used. Any communications medium whether it is a physically connected medium or not can be used as long as the cards can communicate with each other via, for example, random accessible memory addresses. Communications bus 51 can include any hardware interconnection bus that allows communication of data between circuit
20 cards via addressable memory techniques, such as, for example, a Versa Eurocard Module (VME64), CompactPCI and PCI communications buses. The scope of the present invention is for use on a communications bus, which is not necessarily hard-wired among electrical circuitry, but is also intended to cover those communications
buses which communicate data by any means (such as transmission by frequency means).

25

The functionality of each circuit card 41_n is application dependent and can vary depending on the overall system requirements. Without limitation, some examples of circuit card functions can include graphic processing cards, mathematical processing
cards, data storage cards, input/output cards and general purpose processing cards.
30 Typically, each circuit card is housed within a box 43 (also conventionally referred to as "card cages" or simply "cages"), each box 43 connectable to the communications bus 51. Thus, some circuit cards 41_n can be removable depending upon the application.

- 9 -

Because of the nature of communications buses, each card 41_n is able to communicate (and thus, transfer data) with any other card 41_n .

Circuit card 41_n , at a basic level, can include processor(s), memory, and a bus
5 on which they communicate. Generally, there are many components on the cards such as one or two other busses connecting different memories, or input/output devices, and hardware to control power distribution at different voltages and currents. Special purpose chips that perform various functions ranging from providing timers to special
10 mathematical/logical functions to connecting two different busses. Many of the foregoing are commercially available circuit cards and the implementation of such will be apparent to those skilled in the art.

FIG. 4 illustrates the run-time sequence for the present invention to facilitate communications between a first processor 11 in communication with a computer and at
15 least a secondary processor 31_n in communication with a circuit card 41_n having predefined electronic circuitry means and in electrical communication with the first processor 11 by a data communications bus 51. First, a software means for communicating data 33 and a predefined software routine must be loaded on each secondary processor 31_n , the means for communicating data including a
20 communications header register 35. Then, software means for directing 13 is loaded on and executed in the first processor 11, the means for directing 13 including a director header data register 12. Then, by communication between the means for directing 13 and the means for communicating data 33, the electronic circuitry means on each circuit card 41_n is initialized. During this initialization, each circuit card is
25 queried by the means for directing 13 to determine whether the card is completely initialized and to determine whether each card 41_n encountered any errors.

Each means for communicating data 33 maintains a ready state to receive further communication from the means for directing 13. As necessary, the means for
30 directing 13 is thereafter allowed to transfer data with each means for communicating data 33 by polling each communications header register 35. In this regard, the means for directing 13 is allowed to command each circuit card 41_n to commence execution of

any predefined software routine. The means for directing 13 is allowed to electrically test and control the electronic circuitry means on one or more predefined circuit cards 41_n. If needed, the means for directing 13, at a predetermined time, can prepare and control each circuit card 41_n to power down.

5

In one embodiment of the present invention, each card or module performs a minimal set of on-board built-in-tests at start-up to verify execution capabilities. Inter-module tests are then performed to test interfaces between modules including the VME back plane. The resulting fault status is reported in the common header which the primary software program queries. The primary software program is in control of synchronizing all modules during the various modes of the computer system operation. The primary software program is also responsible for maintaining a system status table indicating the health of the particular computer and fault logging. The secondary software program provides top level module control. The secondary software program contains several major synchronization points where it waits for commands from the primary software program. The secondary software programs return state status to the primary software program for control and synchronization.

10

15

CLAIMS

I claim:

1. A computer software control and communication system for a plurality of
5 modules (41) in a computer (10), the plurality of modules (41) connected to a bus (51)
for electrical configuration of the plurality of modules (41), the computer software control
and communication system characterized in that the system comprises:
a common software interface (13, 33) for directing and controlling system
function or state control of the plurality of modules (41) by a director module (11).
10
2. The system of Claim 1 characterized in that the computer software
interface (13, 33) further comprises a director header register (12) stored on the director
module (11) for storage of at least status information and fault information and further
comprises a communications header register (35) stored on each of the modules (41) for
15 storage of at least a secondary status information and at least a secondary fault
information.
3. The system of Claim 1 characterized in that the software interface for
directing and controlling system function or state control (13,33) comprises module level
20 functional modes, module status, and module states (37).
4. The system of Claim 1 characterized in that the communications bus is
selected from the group consisting of a Versa Module Eurocard (VME) communications
bus, a Compact PCI communications bus and a PCI communications bus.

5. A method of controlling and communicating with a plurality of modules (41) in a computer system (10), the plurality of modules (41) connected to a bus (51) for electrical configuration of the plurality of modules (41), the method comprising the step of:

5 directing and controlling system function or state control of the plurality of modules (41) by a director module (11) with a common interface software program (31).

6. The method of Claim 5 wherein the step of directing and controlling
10 system function or state control comprises performing system functions or system transitions in a standardized and synchronized manner.

7. The method of Claim 5 wherein the system function or state control
15 comprise test capabilities, and/or downloading program memory, and/or module power down, and/or normal execution.

8. The method of Claim 5 wherein the step of directing and controlling
each module (41) further includes the step of querying each module (41) to
determine whether the module (41) is completely initialized and to determine
20 whether the module encountered any errors.

9. The method of Claim 5 wherein the communications bus (51) is
selected from the group consisting of a Versa Module Eurocard (VME)
communications bus, a Compact PCI communications bus and a PCI
25 communications bus.