US009535454B2

(12) **United States Patent**
Ahern et al.

(10) **Patent No.: US 9,535,454 B2**
(45) **Date of Patent: Jan. 3, 2017**

(54) **COMPUTING MODULE WITH SERIAL DATA CONNECTIVITY**

(71) Applicant: **INTELLECTUAL VENTURES I LLC**, Wilmington, DE (US)

(72) Inventors: **Frank W. Ahern**, Scottsdale, AZ (US); **Desi Rhoden**, Phoenix, AZ (US); **Jeff Doss**, Phoenix, AZ (US); **Charles Mollo**, Phoenix, AZ (US)

(73) Assignee: **INTELLECTUAL VENTURES I LLC**, Wilmington, DE (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 942 days.

(21) Appl. No.: **13/652,823**

(22) Filed: **Oct. 16, 2012**

(65) **Prior Publication Data**

US 2013/0042046 A1 Feb. 14, 2013

Related U.S. Application Data

(63) Continuation of application No. 13/240,773, filed on Sep. 22, 2011, now Pat. No. 8,291,140, which is a continuation of application No. 12/644,511, filed on Dec. 22, 2009, now Pat. No. 8,060,675, which is a continuation of application No. 11/300,131, filed on Dec. 13, 2005, now Pat. No. 7,657,678, which is a continuation of application No. 09/559,678, filed on Apr. 27, 2000, now Pat. No. 7,734,852, which is a continuation-in-part of application No. 09/130,057, filed on Aug. 6, 1998, now Pat. No. 6,088,752, and a continuation-in-part of application No. 09/130,058, filed on Aug. 6, 1998, now Pat. No. 6,070,214.

(60) Provisional application No. 60/198,317, filed on Apr. 19, 2000.

(51) **Int. Cl.**
G06F 13/00 (2006.01)
G06F 1/16 (2006.01)
G06F 13/40 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 1/1632*** (2013.01); ***G06F 13/4013*** (2013.01); ***G06F 13/4045*** (2013.01)

(58) **Field of Classification Search**
CPC G06F 13/14; G06F 13/16; G06F 13/20; G06F 13/36
USPC 710/305–306, 100; 370/413
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,689,740 A 8/1987 Moelands et al.
4,710,922 A 12/1987 Scott
5,613,141 A * 3/1997 Szatkowski G06F 13/128 710/2
5,832,310 A * 11/1998 Morrissey G06F 13/122 370/392
5,848,046 A 12/1998 Sawada
(Continued)

OTHER PUBLICATIONS

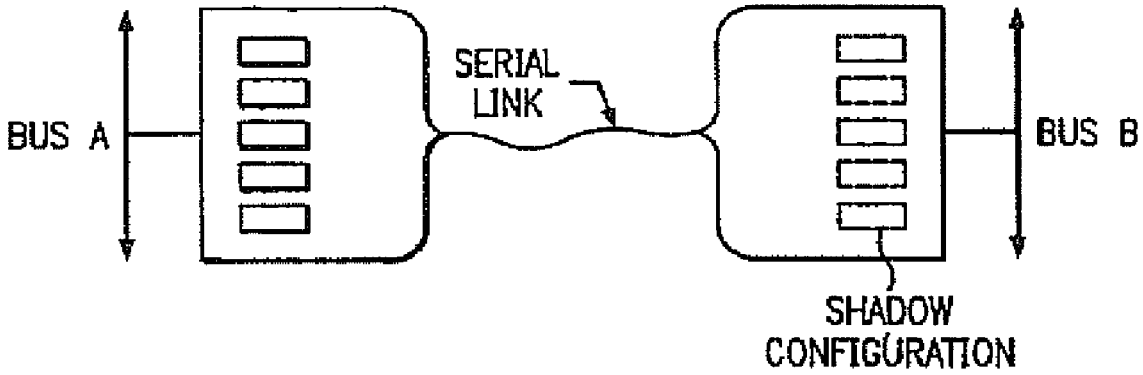
Hewlett Packard; Gigabit Ethernet Transceiver Chip Preliminary Technical Data; May 1997; pp. 711-725; www.datasheetcatalog.com.

Primary Examiner — Tim T Vo
Assistant Examiner — Kim T. Huynh

(57) **ABSTRACT**

A computing module includes an interface to asynchronously, serially exchange parallel system bus data with one or more other modules of a computer system that includes the computing module. The computing module can asynchronously, serially transfer first parallel bus data to another module of the computer system, and can asynchronously, serially receive second parallel bus data from another module of the computer system.

45 Claims, 2 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

5,862,375	A *	1/1999	Gephardt	G06F 12/0684 710/10
5,991,304	A *	11/1999	Abramson	370/413
6,070,214	A *	5/2000	Ahern	G06F 13/4045 710/106
6,192,422	B1 *	2/2001	Daines	H04L 12/46 709/232
6,202,108	B1 *	3/2001	Autechaud et al.	710/61
7,263,564	B2 *	8/2007	Chen	G06F 1/3215 710/109
2005/0188142	A1 *	8/2005	Nakajima	G06F 1/3287 710/303

* cited by examiner

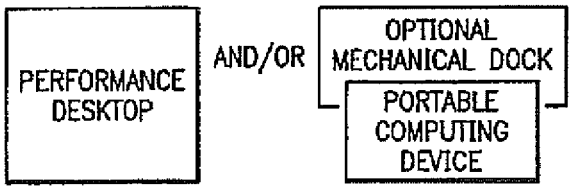


FIG. 1
(PRIOR ART)

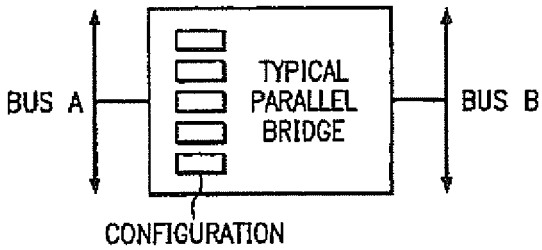


FIG. 2
(PRIOR ART)

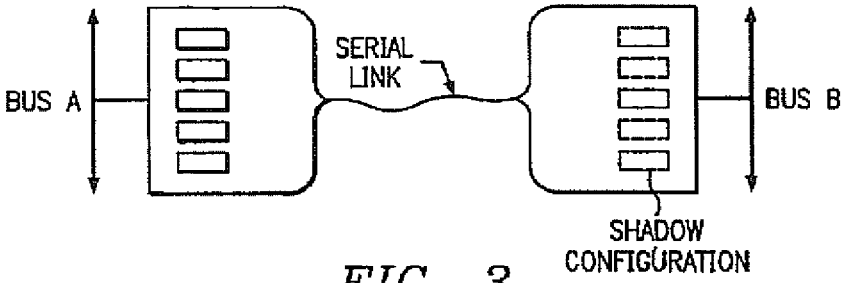


FIG. 3

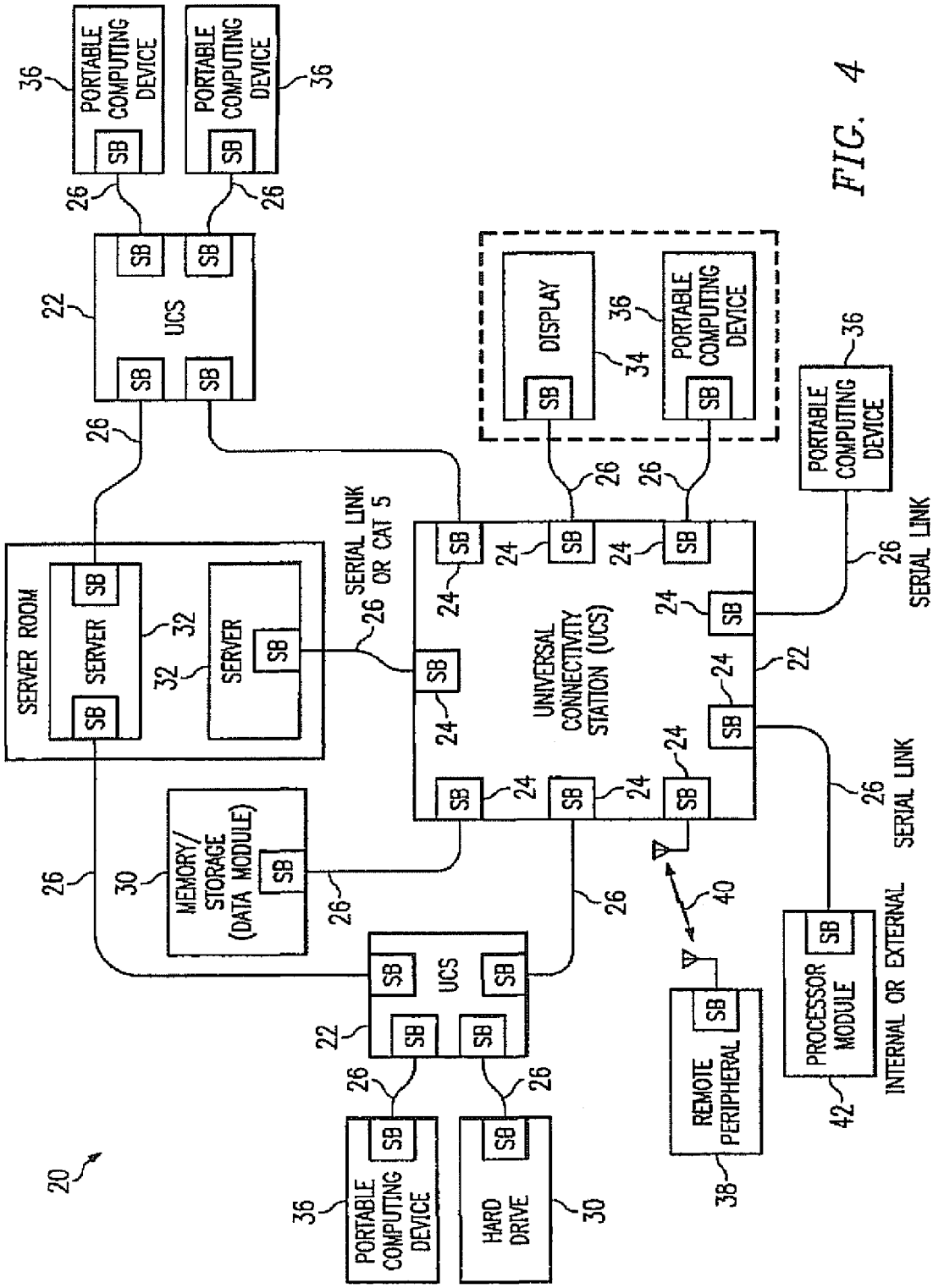

PORTABLE COMPUTING DEVICE
PORTABLE COMPUTING DEVICE
UCS
SERVER ROOM
SERVER
SERVER
SERIAL LINK OR CAT 5
UNIVERSAL CONNECTIVITY STATION (UCS)
DISPLAY
PORTABLE COMPUTING DEVICE
PORTABLE COMPUTING DEVICE
SERIAL LINK
MEMORY/ STORAGE (DATA MODULE)
UCS
PORTABLE COMPUTING DEVICE
HARD DRIVE
REMOTE PERIPHERAL
PROCESSOR MODULE
INTERNAL OR EXTERNAL
SERIAL LINK
SB
20
22
24
26
30
32
34
36
38
40
42

FIG. 4

COMPUTING MODULE WITH SERIAL DATA CONNECTIVITY

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation of and claims priority to and the benefit of patent application Ser. No. 13/240,773, filed on Sep. 22, 2011; which claims priority to and the benefit of patent application Ser. No. 12/644,511, filed on Dec. 22, 2009; which claims priority to and the benefit of patent application Ser. No. 11/300,131, filed on Dec. 13, 2005; which claims priority to and the benefit of the patent application Ser. No. 09/559,678, filed on Apr. 27, 2000; which claims priority to and the benefit of the Provisional Patent Application Ser. No. 60/198,317; which is a Continuation-In-Part of patent application Ser. No. 09/130,057, filed on Aug. 6, 1998, and is a Continuation-In-Part of patent application Ser. No. 09/130,058, filed on Aug. 6, 1998; all of which are incorporated by reference in their entireties.

FIELD OF THE INVENTION

The present invention is generally related to computers and data processing systems, and more particularly to computer systems having at least one host processor and connectable to a plurality of peripherals, expansion devices, and/or other computers, including notebook and other portable and handheld computers, storage devices, displays, USB, IEEE 1394, audio, keyboards, mice and so forth.

BACKGROUND OF THE INVENTION

Computer systems today are powerful, but are rendered limited in their ability to be divided into modular components due to a variety of technical limitations of today's PCI bus technology. And in their ability to adapt to changing computing environments. The PCI bus is pervasive in the industry, but as a parallel data bus is not easily extended over any distance or bridged to other remote PCI based devices due to loading and physical constraints, most notably the inability to extend the PCI bus more than a few inches. Full bridges are known, such as used in traditional laptop computer/docking stations. However, separating the laptop computer from the docking station a significant distance has not been possible. Moreover, the processing power of computer systems has been resident within the traditional computer used by the user because the microprocessor traditionally is directly connected to and resident on the PCI motherboard. Thus, upgrading processing power usually meant significant costs and/or replacing the computer or computer system.

PCI

The PCI bus is primarily a wide multiplexed address and data bus that provides support for everything from a single data word for every address to very long bursts of data words for a single address, with the implication being that burst data is intended for sequential addresses. Clearly the highest performance of the PCI bus comes from the bursts of data, however most PCI devices require reasonable performance for even the smallest single data word operations. Many PCI devices utilize only the single data mode for their transfers. In addition, starting with the implementation of the PCI 2.1 version of the specification, there has been at least pseudo isochronous behavior demanded from the bus placing limits on an individual device's utilization of the bus, thus virtually guaranteeing every device gets a dedicated segment of time on a very regular interval and within a relatively short time

period. The fundamental reason behind such operation of the PCI bus is to enable such things as real time audio and video data streams to be mixed with other operations on the bus without introducing major conflicts or interruption of data output. Imagine spoken words being broken into small unconnected pieces and you get the picture. Prior to PCI 2.1 these artifacts could and did occur because devices could get on the bus and hold it for indefinite periods of time. Before modification of the spec for version 2.1, there really was no way to guarantee performance of devices on the bus, or to guarantee time slot intervals when devices would get on the bus. Purists may argue that PCI is still theoretically not an isochronous bus, but as in most things in PC engineering, it is close enough.

Traditional High Speed Serial

Typical high speed serial bus operation on the other hand allows the possibility of all sizes of data transfers across the bus like PCI, but it certainly favors the very long bursts of data unlike PCI. The typical operation of a serial bus includes an extensive header of information for every data transaction on the bus much like Ethernet, which requires on the order of 68 bytes of header of information for every data transaction regardless of length. In other words, every data transaction on Ethernet would have to include 68 bytes of data along with the header information just to approach 50% utilization of the bus. As it turns out Ethernet also requires some guaranteed dead time between operations to "mostly" prevent collisions from other Ethernet devices on the widely disperse bus, and that dead time further reduces the average performance.

The typical protocol for a serial bus is much the same as Ethernet with often much longer header information. Virtually all existing serial bus protocol implementations are very general and every block of data comes with everything needed to completely identify it. FiberChannel (FC) has such a robust protocol that virtually all other serial protocols can be transmitted across FC completely embedded within the FC protocol, sort of like including the complete family history along with object size, physical location within the room, room measurements, room number, street address, city, zip code, country, planet, galaxy, universe, . . . etc. and of course all the same information about the destination location as well, even if all you want to do is move the object to the other side of the same room. Small transfers across many of these protocols, while possible, are extremely expensive from a bandwidth point of view and impractical in a bus applications where small transfers are common and would be disproportionally burdened with more high overhead than actual data transfer. Of course the possibility of isochronous operation on the more general serial bus is not very reasonable.

Recreating High Speed Serial for PCI

In creating the proprietary Split-Bridge™ technology, Mobility electronics of Phoenix Ariz., the present applicant, actually had to go back to the drawing board and design a far simpler serial protocol to allow a marriage to the PCI bus, because none of the existing implementations could coexist without substantial loss of performance. For a detailed discussion of Applicant's proprietary Split-Bridge™ technology, cross reference is made to Applicant's co-pending commonly assigned patent applications identified as Ser. No. 09/130,057 and 09/130,058 both filed Aug. 6, 1998, the teachings of each incorporated herein by reference. The Split-Bridge™ technology approach is essentially custom fit for PCI and very extensible to all the other peripheral bus protocols under discussion like PCIx, and LDT™ set forth by AMD Corporation. LDT requires a clock link in addition

to its data links, and is intended primarily as a motherboard application, wherein Split-Bridge™ technology is primarily intended to enable remote bus recreation. As the speeds of motherboard buses continue to grow faster, Split-Bridge™ can be readily adapted to support these by increasing the serial bus speed and adding multiple pipes. Split-Bridge™ technology fundamentals are a natural for extending anything that exists within a computer. It basically uses a single-byte of overhead for 32 bits of data and address—actually less when you consider that byte enables, which are not really "overhead", are included as well.

Armed with the far simpler protocol, all of the attributes of the PCI bus are preserved and made transparent across a high speed serial link at much higher effective bandwidth than any existing serial protocol. The net result is the liberation of a widely used general purpose bus, and the new found ability to separate what were previously considered fundamental inseparable parts of a computer into separate locations. When the most technical reviewers grasp the magnitude of the invention, then the wheels start to turn and the discussions that follow open up a new wealth of opportunities. It now becomes reasonable to explore some of the old fundamentals, like peer-to-peer communication between computers that has been part of the basic PCI specification from the beginning, but never really feasible because of the physical limits of the bus prior to Split-Bridge™ technology. The simplified single-byte overhead also enables very efficient high speed communication between two computers and could easily be extended beyond PCI.

The proprietary Split-Bridge™ technology is clearly not "just another high speed link" and distinguishing features that make it different represent novel approaches to solving some long troublesome system architecture issues.

First of all is the splitting of a PCI bridge into two separate and distinct pieces. Conceptually, a PCI bridge was never intended to be resident in two separate modules or chips and no mechanism existed to allow the sharing of setup information across two separate and distinct devices. A PCI bridge requires a number of programmable registers that supply information to both ports of a typical device. For the purpose of the following discussion, the two ports are defined into a north and south segment of the complete bridge.

The north segment is typically the configuration port of choice and the south side merely takes the information from the registers on the north side and operates accordingly. The problem exists when the north and south portions are physically and spatially separated and none of the register information is available to the south side because all the registers are in the north chip. A typical system solution conceived by the applicant prior to the invention of Split-Bridge™ technology would have been to merely create a separate set of registers in the south chip for configuration of that port. However, merely creating a separate set of registers in the south port would still leave the set up of those registers to the initialization code of the operating system and hence would have required a change to the system software.

Split-Bridge™ technology, on the other hand, chose to make the physical splitting of the bridge into two separate and spaced devices "transparent" to the system software (in other words, no knowledge to the system software that two devices were in fact behaving as one bridge chip). In order to make the operations transparent, all accesses to the configuration space were encoded, serialized, and "echoed" across the serial link to a second set of relevant registers in the south side. Such transparent echo between halves of a PCI bridge or any other bus bridge is an innovation that significantly enhances the operation of the technology.

Secondly, the actual protocol in the Split-Bridge™ technology is quite unique and different from the typical state of the art for serial bus operations. Typically transfers are "packetized" into block transfers of variable length. The problem as it relates to PCI is that the complete length of a given transfer must be known before a transfer can start so the proper packet header may be sent.

Earlier attempts to accomplish anything similar to Split-Bridge™ technology failed because the PCI bus does not inherently know from one transaction to the next when, or if, a transfer will end or how long a block or burst of information will take. In essence the protocol for the parallel PCI bus (and all other parallel, and or real time busses for that matter) is incompatible with existing protocols for serial buses.

An innovative solution to the problem was to invent a protocol for the serial bus that more or less mimics the protocol on the PCI. With such an invention it is now possible to substantially improve the performance and real time operation here to for not possible with any existing serial bus protocol.

The 8 bit to 10 bit encoding of the data on the bus is not new, but follows existing published works. However, the direct sending of 32 bits of information along with the 4 bits of control or byte enables, along with an additional 4 bits of extension represents a 40 bit for every 36 bits of existing PCI data, address, and control or a flat 10% overhead regardless of the transfer size or duration, and this approach is new and revolutionary. Extending the 4 bit extension to 12 or more bits and including other functionality such as error correction or retransmit functionality is also within the scope of the Split-Bridge™ technology.

New Applications of the Split-Bridge™ Technology

Basic Split-Bridge™ technology was created for the purpose of allowing a low cost, high speed serial data communications between a parallel system bus and remote devices. By taking advantage of the standard and pervasive nature of the PCI bus in many other applications in computing, dramatic improvements in the price performance for other machines is realized. The present invention comprises a revolutionary application rendered possible due to the attributes of applicant's proprietary Split-Bridge™ technology.

SUMMARY

The present invention achieves technical advantages as a modular computer system having a universal connectivity station adapted to connect and route data via serial data links to a plurality of devices, these serial links and interfaces at each end thereof employing proprietary Split-Bridge™ technology disclosed and claimed in co-pending and commonly assigned patent applications identified as Ser. No. 09/130,057 and 09/130,058, the teachings of which are incorporated herein by reference.

The present invention derives technical advantages as a modular computer system by separating into two or more spatially separate and distinct pieces, a computer core and a universal connectivity station (UCS). The core is the performance module of the modular computer system and may include some or all of the central processing unit (CPU), memory, AGP Graphics, and System Bus Chip adapted to communicably couple these three together or in combination with other items. The UCS communicably couples the processor module via high speed serial links based on the

proprietary Split-Bridge™ technology of Mobility Electronics of Phoenix Ariz., the applicant of the present invention, to other computers or to other individual modules such as storage modules including hard disk drives, a user interface module consisting of a keyboard, mouse, monitor and printer, as well as a LAN Module such as any Internet connection or another UCS, another UCS, audiovisual device, LAN storage just to name a few. In addition, the UCS is adapted to couple via a Split-Bridge™ technology serial link with a portable or handheld computer or device remotely located from the UCS but still functionally coupled to the modular computer system via the UCS. The UCS and associated Split-Bridge™ technology serial links are all transparent to the modules which can have parallel data busses including those based on PCI or Cardbus architectures.

The modular computer system of the present invention including the UCS is a novel approach to computer architecture and upgrade ability. Advantageously, the separate performance module may be selectively upgraded or modified as desired and as technology increases the performance of key components including microprocessor speed, standards, and architectures without necessitating the replacement or modification of the rest of the computer system. The UCS allows the performance module to be upgraded while the rest of the system devices coupled thereto do not need to be modified. Upgrading to single or multiple processors in the performance module or modules is readily possible. Whole organizations can standardize to a single UCS regardless of the type of performance or portability required by the users, thus addressing for the first time the means of systems level support. In security sensitivity environments, it is possible to separate the "stored media" or computer central processor, or any other component of the system and connectivity from the operators, and still maintain the speed element so important in today's businesses.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates prior art computer systems depicted as a traditional performance desk top computer shown at **10**, and a portable computing device **12**, such as a notebook or laptop computer, mechanically coupled to mechanical docking station **14**;

FIG. **2** is a block diagram of a prior art bridge **16** used to couple two system computing buses, such as used between the portable computing device **12** and the mechanical docking station **14** shown in FIG. **1**;

FIG. **3** illustrates the proprietary Split-Bridge™ technology serial communication technology of the applicant enabling high speed serial communications within the modular computer system of the present invention; and

FIG. **4** is a block diagram of the modular computer system of the present invention utilizing a universal connectivity station (UCS) communicably coupled to a plurality of devices via serial links, such as the Split-Bridge™ technology serial links employed using fixed wire, optical, or wireless communication links.

DETAILED DESCRIPTION

Referring to FIG. **3**, there is depicted the proprietary Split-Bridge™ technology serial communications technology of the present applicant, discussed in great detail in commonly assigned U.S. patent application Ser. No. 09/130,057 filed Aug. 6, 1998, and Ser. No. 09/130,058 also filed Aug. 6, 1998 the teachings of which are incorporated herein by reference.

Applicant Split-Bridge™ technology revolutionizes the status quo for computer systems. The Split-Bridge™ technology does not require the need for custom hardware or custom software to achieve full performance serial communication between devices, including devices having parallel data buses including the PCI bus. In fact, for each device in a modular computer system, the Split-Bridge™ technology appears just like a standard PCI bridge, and all software operating systems and device drivers already take such standard devices into consideration. By utilizing standard buses within each device operating within the modular computer system, each device does not require any additional support from the Operating System (OS) software. The modular computing system has simple elegance, allowing the PCI bus which is so pervasive in the computer industry, that possible applications of the initial PCI form of Split-Bridge™ technology are all most limitless.

Originally implemented in PCI, there is nothing fundamental that ties the Split-Bridge™ technology to PCI, and thus, the Split-Bridge™ technology can migrate as bus architectures grow and migrate. The 64 bit PCI is compatible with the Split-Bridge™ technology, as is future PCIx and/or LDT or other bus technologies that are currently under consideration in the industry and which are straight forward transitions of the Split-Bridge™ technology. Implementations with other protocols or other possible and natural evolutions of the Split-Bridge™ technology, including digital video (DV) technology that can be streamed over the high-speed serial link.

Referring to FIG. **4**, there is depicted at **20** a modular computer system according to one illustrative embodiment of the present invention. The modular computer system **20** is based around one or more universal connectivity stations generally shown at **22** each having a plurality of interface ports **24** which are preferably based on the proprietary Split-Bridge™ technology of the present applicant, Mobility Electronics of Phoenix Ariz. Each UCS **22** provides input/output, or I/O, capability of the computer or computer system **20**, as well as modular expansion capability and features. UCS **22** includes all possible variations and combinations of port replication and connectivity, including but not limited to the following ports: P/S2, mouse and keyboard, serial, parallel, audio, USB, IEEE 1394, or firewire, SCSI, and the like. Each UCS **22** also includes the ability to expand the capability or features of the computer system **20** by adding any type of drive bays, including EIDE, USB, and 1394 CD Roms, DVD's, hard drives, tape back up's, ZIP Drives®, Jazz® drives, and the like.

A plurality of interconnecting and interactive devices are communicably coupled to each UCS **22** via respective high speed serial links generally shown at **26** based on the proprietary Split-Bridge™ technology. In the hardwire embodiment, the serial links **26** comprise of a pair of simplex links forms a duplex link interconnecting each end of the Split-Bridge™ technology interfaces as shown. The serial links **26** may also employ optical fiber and optical transceivers if desired. The various modules making up modular computer system **20** may include, and a plurality of, but are not limited to, a memory/storage device **30**, servers **32** having one or multiple processors and possibly serving other UCS's **22**, as shown, and modular computer systems, remote users and so forth, a display **34**, a portable computing device **36**, such as a notebook computer, a laptop computer, a portable digital assistant (FDA), and a remote wireless

peripheral **38** which may interconnected via a wireless link shown at **40** and implementing the proprietary Split-Bridge™ technology. Examples of remote wireless terminals **38** may include 3rd generation (3G) devices now being developed and employed, including wireless personal devices having capabilities for voice, data, video and other forms of information which can be unidirectionally or bidirectionally streamed between the remote peripheral **38** and UCS **22**. An appropriate antenna resides at each of the remote peripheral **38** and UCS **22** which are interconnected to respective transceivers communicably coupled to the respective ends of the Split-Bridge™ technology interfaces.

Moreover, multiple UCS's **22** can be integrated to communicate with each other via serially links **26**, each UCS's **22** locally serving multiple modules. Multiple computers can be connected to a common UCS, or to multiple UCS's. For example, a computer or server room can have racks of computer processors or servers, each separately connected over a system of up to hundreds of feet, to one or many UCS's located throughout an office or other environment. This allows the desktop to have just a terminal or whatever capabilities the IT manager desires, enhancing security and control.

System **20** also provides the ability to simultaneously connect multiple computers **36** and allows full peer-to-peer communications, allowing the processor module (CPU) **42** to communicate with the portable device computer **36** or to the computer room computers **32**, allowing all of these computers to share information and processing capability. This also allows certain of the computers, such as the portable computer **36**, to upgrade its processing capability when it is connected to the UCS **22** with other higher capability computers.

Still referring to FIG. **4**, the modular computer system **20** of the present invention further comprises a processor module **42**, which may be remotely positioned from the UCS **22**, but for purposes of inclusion, could internally reside with the UCS **22**. The processor module **42**, from a performance point of view, is the heart and sole of the modular computer system **20** and can be made up of one or more core parts including: the CPU, memory, APG Graphics, and a system bus interface to connect the other 3 together. The processor module **42** operates in conjunction with memory such as a hard disk drive, which can reside within the processor module **42**, or be remotely located as shown at **30** if desired. The APG Graphics could be located separately within the system and interconnected via a serial link **26**, or even located within UCS **22** if desired.

Advantageously, the processor module **42** which may comprise of a high speed microprocessor or microprocessors, digital signal processors (DSP's), and can be upgraded or interchanged from the systems **20** without effecting the other devices or operation of the system, thereby permitting increased performance at a very low cost. Computers today typically require the replacement or upgrading of other devices when the performance portion of the computer system is replaced. The modular computer system **20** of the present invention revolutionizes the computer architectures available by separating out the processor module **42** from the rest of the computer system **20**. Each of the modules **30**,**32**,**34**,**36**, and **38** all have functional access and use of the processor module **42** via the UCS **22** over the respective serial links **26** and **40**, and from a performance point of view, appear to each of these devices to be hardwired to the processor module **42**. That is, the Split-Bridge™ technology links interconnecting each of the devices via the UCS **22** to the processor module **42** is transparent to each device, thus requiring no change to the OS of each device, the format of data transfer therebetween, or any other changes. This is rendered possible by the revolutionary Split-Bridge™ technology.

Another advantage of computer system **20** is that the data module **30** may be customized, portable, and used only by one user. This allows the user to take the portable module **30** with them from location to location, system **20** to system **20**. The data module **30** can store each user's unique information, and can be accessed and used on any processor module **42** and UCS **22**.

As discussed in considerable detail in the cross-referenced and commonly assigned patent applications, the Split-Bridge™ technology provides that information from the parallel buses of each device be first loaded into first-in first-out (FIFO) registers before being serialized into frames for transmission over the high speed serial link. Received frames are deserialized and loaded into FIFO registers at the other end thereof, such as UCS **22**, for being placed onto the destination bus of the opposing device. Interrupts, error signals and status signals are sent along the serial link. Briefly, the proprietary Split-Bridge™ technology takes address and data from a bus, one transaction at a time, together with 4 bits that act either as control or byte enable signals. Two or more additional bits may be added to tag each transaction as either an addressing cycle, an acknowledging of a non-posted write, a data burst, end of data burst or cycle. If these transactions are posted writes they can be rapidly stored in a FIFO register before being encoded into a number of frames that are sent serially over the link. When pre-fetched reads are allowed, the FIFO register can store pre-fetched data in case the initiator requests it. For single cycle writes or other transactions that must await a response, the bridge can immediately signal the initiator to retry the request, even before the request is passed to the target.

In the preferred embodiment of the modular computer system of the present invention, one or more of the busses in the plurality of devices, as well as in the UCS **22**, employ the PCI or PCMCIA standard, although it is contemplated that other bus standards can be used as well. The preferred Split-Bridges™ technology operates with a plurality of configuration registers that is loaded with information specified under the PCI standard. The Split-Bridges™ technology transfers information between busses depending on whether the pending address falls within a range embraced by the configuration registers. This scheme works with devices on the other side of the Split-Bridge™ technology, which can be given unique base addresses to avoid addressing conflicts.

As disclosed in great detail in the co-pending and cross-referenced commonly assigned patent applications, the Split-Bridges™ technology may be formed as two separate application-specific integrated circuits (ASICs) joined by a duplex link formed as a pair of simplex links. Preferably, these two integrated ASICs have the same structure, but can act in two different modes in response to a control signal applied to one of its pins. Working with hierarchical busses (primary and secondary busses) these integrated circuits are placed in a mode appropriate for its associated bus. The ASIC associated with the secondary bus preferably has an arbitrator that can grant masters control of the secondary bus. The ASIC can also supply a number of ports to support other devices such as a USB and generic configurable I/O ports, as well as parallel and serial ports.

The UCS preferably comprises a PCI bus having a plurality of PC card slots located with the UCS housing. Each PC card slot is provided with a Split-Bridge™ tech-

nology interface, and preferably one of the ASICs assembled with a standardized serial connector comprising at least 4 pins, as depicted in the cross referenced commonly assigned patent applications, the teachings of which are incorporated herein by reference.

The modular computer system **20** of the present invention derives technical advantages in that the UCS station **22** with its associated interface cards and parallel data bus interconnecting each interface card, is truly functionally transparent to each of the interconnected modules including the memory storage device **30**, the server **32**, the display **34**, the portable computing device **36**, the remote wireless peripheral **38**, and the processor module **42**. This integration of devices into a modular computer system has truly enormous potential and uses depending on the desired needs and requirements of one's computing system. However, the physical location and proximity of each of the devices forming the modular computer system are no longer strictly limited due to the high speed serial interconnection links of the proprietary Split-Bridge™ technology. Each of the devices can be remotely located, or located in proximity to one another as desired. For instance, the display **34** and portable computing device **36** may be resident within one's office, with the UCS **22** in another room, and with the memory storage device **30**, server **32**, and performance module **42** remotely located in yet still another room or location. Moreover, a plurality of portable computing devices **36** can all be remotely located from UCS **22**, and from each other, allowing networking to modular system **20** either through wireless serial links as depicted at **26**, or wirelessly as depicted at **40**.

The proprietary Split-Bridge™ technology presently allows for extended communication distances of 5 meters, but through advancement in technology can continue to be extended. For instance, using optical communication links in place of copper wire simplex links, along with suitable optical transceivers, can yield links that are exceptionally long. Using wireless technology, as depicted at **40**, allows a remote peripheral **38** to be located perhaps anywhere in the world, such as by implementing repeaters incorporating the proprietary Split-Bridge™ technology high speed serial communication technology. Additional techniques can be used by slowing the transfer rate, and increasing the number of pipes, to achieve link distances of hundreds of feet, and allowing the use of CAT5 cable.

Though the invention has been described with respect to a specific preferred embodiment, many variations and modifications will become apparent to those skilled in the art upon reading the present application. It is therefore the intention that the appended claims be interpreted as broadly as possible in view of the prior art to include all such variations and modifications.

What is claimed is:

1. A system, comprising:

a connector configured to be connectable to a serial link; and

a module coupled to the connector, wherein the module is operable to receive first serial data transmitted over the serial link through the connector and transmit second serial data to the serial link through the connector, and wherein the module includes:

a receiver operable to receive the first serial data from the serial link through the connector;

a first circuit configured to deserialize the first serial data to generate deserialized data;

a decoder configured to decode the deserialized data;

a first buffer operable to store data that has been deserialized and decoded by the first circuit and the decoder, respectively;

a second buffer operable to receive data, wherein the second buffer is in data communication with a second circuit that is configured to serialize data received via the second buffer to generate the second serial data; and

a transmitter operable to transmit the second serial data to the serial link through the connector;

wherein the module is configured to determine whether an amount of data stored in the first buffer equals or exceeds a fill amount associated with a storage capacity of the first buffer; and

wherein the module is operable to generate a flow control signal in response to the amount of data stored in the first buffer equaling or exceeding the fill amount, and wherein the module is further configured to transmit the flow control signal through the serial link and the connector without passing the flow control signal through the second buffer.

2. The system of claim **1**, wherein:

the system is configured to transmit the first serial data and the second serial data asynchronously over the serial link.

3. The system of claim **1**, wherein:

the data received by the second buffer comprises parallel data.

4. The system of claim **1**, wherein:

the module is further operable to repeatedly transmit the second serial data through the serial link and the connector without receiving acknowledgement during transmission.

5. The system of claim **1**, wherein:

the serial link comprises a first twin axial line and a second twin axial line.

6. The system of claim **1**, wherein:

the module is further operable to transmit the first serial data over the serial link and to transfer the second serial data over the serial link using differential signals.

7. The system of claim **1**, wherein:

the flow control signal is configured to be used to control transmission of the first serial data over the serial link to prevent overflow of the first buffer.

8. The system of claim **1**, wherein:

the module is further operable to generate control bits;

the second serial data comprises one or more serial data frames; and

the module is further operable to transmit the control bits through the serial link between transmissions of the one or more serial data frames, without passing the control bits through the second buffer.

9. The system of claim **8**, wherein:

the control bits correspond to one or more comma markers; and

the control bits are configured to be used in establishing serial communications over the serial link.

10. The system of claim **1**, wherein:

the data received by the second buffer comprises one or more control bits;

the second serial data comprises serialized data frames including serialized control bits; and

the module is further operable to transmit the serialized data frames over the serial link.

11. The system of claim **1**, wherein:

the module is further operable to transmit one or more bits corresponding to one or more serialized interrupts,

error signals, or status signals over the serial link between transmissions of the second serial data, without passing the one or more bits corresponding to the one or more serialized interrupts, the error signals, or the status signals through the second buffer.

12. The system of claim **1**, wherein:

the module is further operable to generate one or more control bits corresponding to one or more interrupts, without the control bits passing through the second buffer;

the second serial data comprises serial data frames; and

the module is further operable to form one or more encoded control bits using the one or more control bits and include the one or more encoded control bits in the serial data frames.

13. The system of claim **1**, wherein:

the module is further operable to initiate transmission of the second serial data over the serial link in response to changes to information stored in the second buffer so that information stored in a third buffer included in another module coupled to the serial link through another connector matches the information stored in the second buffer.

14. The system of claim **13**, wherein:

the other module is included in a computer system; and

the module and the other module are operable to cause the information stored in the third buffer to match the information stored in the second buffer in a manner which is transparent to system software operating on the computer system.

15. The system of claim **1**, wherein:

the system comprises a hard disk drive.

16. The system of claim **1**, further comprising:

an interface included in the module coupled to the first buffer and operable to receive the data stored in the first buffer;

a bus coupled to the module and configured to receive the data from the interface; and

a processor coupled to the bus.

17. The system of claim **1**, wherein:

the module is further configured to generate a clock signal, for use by the module, using the first serial data.

18. A method of changing storage in a second buffer in response to one or more changes in storage in a first buffer, the method comprising:

storing data bits in the first buffer from a first module included in a computing system;

serializing data bits obtained from the first buffer to form serialized data bits using a first serializer included in the first module;

transmitting the serialized data bits over a serial link;

receiving the serialized data bits at a disk drive system;

deserializing the serialized data its using a deserializer included in a second module included in the disk drive system; and

storing data bits obtained from the deserializer in the second buffer included in the second module, wherein changing of the storage in the second buffer is initiated by the first module to result in the second buffer information being equivalent to the first buffer information and is performed in a manner which is transparent to system software operating on the computing system.

19. The method of claim **18**, wherein:

the serialized data bits are included in serialized data frames; and

the data bits are provided to the first module over a parallel bus included in the computing system.

20. The method of claim **19**, further comprising:

embedding one or more control bits with the data bits;

storing the one or more control bits embedded with the data bits in the first buffer;

using the first serializer, serializing the one or more control bits to form serialized one or more control bits included in the serialized data frames;

receiving the serialized one or more control bits at a disk drive system;

deserializing the serialized one or more control bits using the deserializer; and

storing the one or more control bits obtained from the deserializer in the second buffer.

21. The method of claim **20**, wherein:

the one or more control bits correspond to one or more interrupts.

22. The method of claim **21**, wherein:

at least two bits are used to represent the one or more interrupts.

23. The method of claim **18**, further comprising:

generating one or more control bits using the first module, wherein the one or more control bits are not stored in the first buffer; and

using the first module, transmitting the one or more control bits over the serial link between transmissions of the serialized data bits.

24. The method of claim **23**, wherein:

the one or more control bits correspond to one or more interrupts or error signals.

25. The method of claim **24**, wherein:

the one or more control bits are 8-bit to 10-bit encoded.

26. The method of claim **18**, wherein:

the serial link comprises a twin axial line; and

said transmitting the serialized data bits is performed over the twin axial line.

27. The method of claim **26**, wherein:

said transmitting the serialized data bits is performed over the twin axial line using differential signals.

28. The method of claim **18**, further comprising:

using the second module, generating a clock signal, for use by the second module, using the serialized data bits.

29. The method of claim **18**, wherein:

said transmitting the serialized data bits is repeatedly performed by the first module without receiving acknowledgement of said transmitting from the second module over the serial link during transmission.

30. The method of claim **18**, wherein:

the system software of the computing system does not include a driver for control of said changing.

31. A system, comprising:

a connector configured to be connectable to a serial link; and

a module coupled to the connector, wherein the module is operable to receive first serial data transmitted over the serial link through the connector and transmit second serial data to the serial link through the connector, and wherein the module includes:

a receiver operable to receive the first serial data from the serial link through the connector;

a first circuit configured to deserialize the first serial data to generate deserialized data and decode the deserialized data;

a first buffer operable to store data that has been deserialized and decoded by the first circuit;

a second buffer operable to receive data, wherein the second buffer is data communication with a second circuit that is configured to serialize data received via the second buffer to generate second serial data; and

a transmitter operable to transmit the second serial data to the serial link through the connector;

wherein the module is configured to:

determine that an amount of data stored in the first buffer meets a prescribed criterion;

generate a flow control signal in response to the determination; and

cause transmission of the flow control signal via the serial link and the connector without passing the flow control signal through the second buffer.

32. A system, comprising:

a module configured to generate a clock signal, for use by the module, using first serial data, wherein the module includes:

a receiver operable to receive the first serial data from a serial link;

a first circuit configured to deserialize the first serial data to generate deserialized data;

a first buffer coupled to the first circuit and operable to store data that has been deserialized using the first circuit;

a second buffer operable to receive data from a bus, wherein the second buffer is coupled to a second circuit configured to serialize data obtained from the second buffer to generate second serial data; and

a transmitter configured to transmit the second serial data to the serial link;

wherein the module is configured to request retransmission of the first serial data in response to the module detecting a transmission error in the first serial data.

33. The system of claim **32**, wherein:

the module comprises an ASIC.

34. The system of claim **32**, wherein the module is configured to:

determine that an amount of data stored in the first buffer meets a prescribed criterion;

generate a flow control signal in response to the determination; and

cause transmission of the flow control signal via the serial link and the connector without passing the flow control signal through the second buffer.

35. The system of claim **34**, further comprising:

an interface included in the module, coupled to the first buffer, and operable to receive data from the first buffer;

a bus coupled to the module and configured to receive the data from the interface; and

a processor coupled to the bus.

36. The system of claim **35**, wherein:

the serial link comprises a first twin axial line and a second twin axial line.

37. The system of claim **35**, wherein:

the system comprises a hard disk drive.

38. A system, comprising:

a module including:

a receiver operable to receive first serial data from a serial link;

a first circuit configured to deserialize the first serial data to generate deserialized data;

a first buffer in data communication with the first circuit and operable to store data originating from the first circuit;

a second buffer operable to receive data via a bus, wherein the second buffer is in data communication with a second circuit configured to serialize data originating from the second buffer to generate second serial data; and

a transmitter operable to transmit the second serial data via the serial link;

wherein the module is configured to request retransmission of the first serial data in response to the module detecting a transmission error in the first serial data; and

wherein the module is further configured to detect whether an amount of stored data exceeds an acceptable threshold and, if the amount of data exceeds the acceptable threshold, cause the transmitter to send a flow control signal through the serial link without passing the flow control signal through the second buffer.

39. The system of claim **38**, wherein:

the module comprises an ASIC.

40. The system of claim **38**, wherein:

the system comprises a hard disk drive.

41. The system of claim **38**, further comprising:

an interface, included in the module, in data communication with the first buffer and operable to receive the data via the first buffer;

a bus in data communication with the interface and configured to receive the data from the interface; and

a processor coupled to the bus.

42. The system of claim **38**, wherein:

the serial link comprises a first twin axial line and a second twin axial line;

the receiver is operable to receive the first serial data from the first twin axial line as first differential signals; and

the transmitter is operable to transmit the second serial data to the second twin axial line through the connector as second differential signals.

43. The system of claim **38**, wherein:

the flow control signal is configured to be used to control transmission of the first serial data over the serial link to prevent overflow of the first buffer.

44. The system of claim **38**, wherein:

the module is further configured to generate a clock signal, for use by the module, using the first serial data.

45. The system of claim **31**, wherein:

the second buffer comprises a FIFO.

* * * * *