United States District Court,
S.D. California.

HEWLETT-PACKARD COMPANY and Hewlett-Packard Development Company, L .P,
Plaintiffs.
v.
GATEWAY, INC,
Defendant.
Gateway, Inc,
Counterclaim-Plaintiff.
v.
Hewlett-Packard Development Company L.P., Hewlett-Packard Company and Compaq Information Technologies Group, L.P,
Counterclaim-Defendants.

Civil No. 04CV0613-B(LSP)

Feb. 13, 2006.

John Allcock, DLA Piper US, San Diego, CA, for Plaintiffs.

Darryl J. Adams, Dean M. Munyon, James D. Smith, Wayne Harding, Dewey Ballantine, W. Bryan Farney, Dechert LLP, Austin, TX, Jonathan D. Baker, Dechert LLP., Mountain View, CA, for Defendant.

CLAIM CONSTRUCTION ORDER FOR UNITED STATES PATENT NUMBER 5,802,318

RUDI M. BREWSTER**, Senior District Judge.**

Pursuant to Markman v. Westview Instruments, Inc., 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996), on November 29-30, 2005, December 1, 2005, and January 10-11, 2006, the Court conducted a Markman hearing in the above-titled patent infringement action regarding construction of the disputed claim terms for U.S. Patent Number 5,802,318 ("the '318 patent"). Plaintiffs Hewlett-Packard Company and Hewlett-Packard Development Company, L.P. (collectively "HP") were represented by the law firm of DLA Piper Rudnick Gray Cary U.S. LLP, and Defendant Gateway, Inc. ("Gateway") was represented by the law firm of Dewey Ballantine LLP.

At the Markman hearing, the Court, with the assistance of the parties, analyzed the claim terms in order to prepare jury instructions interpreting the pertinent claims at issue in the '318 patent. Additionally, the Court prepared a case glossary for terms found in the claims and the specification for the '318 patent considered to be technical in nature which a jury of laypersons might not understand clearly without specific definition.

After careful consideration of the parties' arguments and the applicable statutes and case law, the Court **HEREBY CONSTRUES** the claims in dispute in the '318 patent and **ISSUES** the relevant jury instructions

as written in Exhibit A, attached hereto. Further, the Court **HEREBY DEFINES** all pertinent technical terms as written in Exhibit B, attached hereto.

IT IS SO ORDERED.

EXHIBIT A FN1

UNITED STATES PATENT NUMBER 5.802,318-CLAIM CHART

VERBATIM CLAIM LANGUAGE	***COURT'S CONSTRUCTION***
Claim 1	
1. A serial bus host controller for coupling a serial bus keyboard to a computer system via a standardized serial bus which transfers data in a packetized protocol, the serial bus host controller for sending and receiving serial bus packets, the serial bus host controller comprising:	1. A ***serial bus host controller*** [*circuitry that supports a device's communication over a serial bus by connecting the serial bus with a bus in the computer system*] for coupling a ***serial bus keyboard*** [*a keyboard that communicates over a serial bus*] to a computer system via a standardized ***serial bus*** [*a hardware line used for transferring data in a bit stream among the components of a computer system*] which transfers data in a ***packetized protocol*** [*a set of rules governing data packet transmission*], the ***serial bus host controller*** for sending and receiving ***serial bus packets*** [*bundles of data organized in groups for transmission over the serial bus*], the ***serial bus host controller comprising*** [*including but not limited to*]:
a keyboard controller emulator for generating and receiving data, status and commands pertaining to the serial bus keyboard, said keyboard controller emulator including:	(a) a ***keyboard controller emulator*** [*hardware or hardware and software that imitates a keyboard controller in generating and receiving data, status and commands pertaining to a serial bus keyboard as if the keyboard controller were present*] for generating and receiving data, status and commands pertaining to the ***serial bus keyboard,*** said ***keyboard controller emulator including*** [*at least but not limited to*]:
a serial bus address register for storing the serial bus address of the serial bus keyboard;	(a)(i) a ***serial bus address*** [*an identifier designating a particular device on the serial bus*] ***register*** [*a device, other than main memory, which holds a set of data bits for a particular purpose*] for storing the ***serial bus address*** of the ***serial bus keyboard;***
a data buffer; and	(a)(ii) a ***data*** [*commands and information pertaining to the keyboard*] ***buffer*** [*a* ***register*** *for temporarily storing* ***data***]; and
a status register;	(a)(iii) a ***status*** [*report of* ***data*** *pertaining to the keyboard and/or mouse*] ***register;***
a detector for detecting when said data buffer and said status register are accessed; and	(b) a ***detector*** [*a circuit for sensing*] for detecting when said ***data buffer*** and said ***status register*** are ***accessed*** [*written into or read from*]; and
an interrupt generator for providing a system management interrupt to the computer system when said data buffer and said status register are accessed.	(c) an ***interrupt generator*** [*a circuit device or code that creates a signal requesting attention from the processor*] for providing a ***system management interrupt*** [*a signal provided to the computer system when the* ***data buffer*** and the ***status register*** are ***accessed***] to the computer system when said ***data buffer*** and said ***status register*** are ***accessed.***
Claim 2	
2. The serial bus host controller of claim 1, wherein said detector	2. The ***serial bus host controller*** of claim 1, wherein said ***detector further detects if a packet is received from said serial bus keyboard***

further detects if a packet is received from said serial bus keyboard, and wherein said interrupt generator provides a system management interrupt to the computer system if a packet is received from said serial bus keyboard.	[the ***detector*** also senses if a ***packet*** is from the serial bus keyboard or not], and wherein said interrupt generator provides a system management interrupt to the computer system if a packet is received from said serial bus keyboard.
Claim 3	
3. The serial bus host controller of claim 2, wherein said interrupt causes the computer system to place the serial bus keyboard data into said data buffer and causes said status buffer to be updated if a packet is received from said serial bus keyboard.	3. The ***serial bus host controller*** of claim 2, wherein ***said interrupt*** (*the* ***system management interrupt***) causes the computer system to place the ***serial bus keyboard data*** into said ***data buffer*** and causes said ***status buffer*** (***the status register***) to be updated if a ***packet*** is received from said ***serial bus keyboard.***
Claim 4	
4. The serial bus host controller of claim 1, further including a switch for enabling and disabling said keyboard controller emulator.	4. The ***serial bus host controller*** of claim 1, further including a switch for enabling and disabling said ***keyboard controller emulator.***
Claim 5	
5. The serial bus host controller of claim 1 wherein said serial bus keyboard includes a serial bus mouse.	5. The ***serial bus host controller*** of claim 1 wherein said ***serial bus keyboard*** includes a serial bus mouse [*a computer mouse that communicates over a serial bus*].
Claim 6	
6. A computer system for coupling to a serial bus keyboard via a standardized serial bus which transfers data in a packetized protocol, the computer system comprising:	6. A computer system for coupling to a ***serial bus keyboard*** via a standardized ***serial bus*** which transfers ***data*** in a packetized protocol. the computer system ***comprising:***
a serial bus keyboard; and	(a) a ***serial bus keyboard:*** and
a serial bus host controller coupled to said serial bus keyboard for sending and receiving serial bus packets, comprising:	(b) a ***serial bus host controller*** coupled to said ***serial bus keyboard*** for sending and receiving ***serial bus packets, comprising:***
a keyboard controller emulator for generating and receiving data, status and commands pertaining to the serial bus keyboard, said keyboard controller emulator including:	(c) ***a keyboard controller emulator*** for generating and receiving data, status and commands pertaining to the ***serial bus keyboard,*** said ***keyboard controller emulator including:***
a serial bus address register for	(c)(i) a ***serial bus address register*** for storing the ***serial bus address*** of

storing the serial bus address of the serial bus keyboard;	the ***serial bus keyboard:***
a data buffer; and	(c)(ii) a ***data buffer:*** and
a status register;	(c)(iii) a ***status register;***
a detector for detecting when said data buffer and said status register are accessed; and	(d)a ***detector*** for detecting when said ***data buffer*** and said ***status register*** are ***accessed;*** and
an interrupt generator for providing a system management interrupt to the computer system when said data buffer and said status register are accessed.	(e) an ***interrupt generator*** for providing a ***system management interrupt*** to the computer system when said ***data buffer*** and said ***status register*** are ***accessed.***
Claim 7	
7. The computer system of claim 6, wherein said detector further detects if a packet is received from said serial bus keyboard; and wherein said interrupt generator provides a system management interrupt to the computer system if a packet is received from said serial bus keyboard.	7. The computer system of claim 6, wherein said ***detector*** further detects if a ***packet*** is received from said serial bus keyboard, and wherein said ***interrupt generator*** provides a ***system management interrupt*** to the computer system if a ***packet*** is received from said ***serial bus keyboard.***
Claim 8	
8. The computer system of claim 7, wherein said interrupt causes the computer system to place the serial bus keyboard data into said data buffer and causes said status buffer to be updated if a packet is received from said serial bus keyboard.	8. The computer system of claim 7, wherein ***said interrupt*** (the system management interrupt) causes the computer system to place the ***serial bus keyboard data*** into said ***data buffer*** and causes ***said status buffer*** (*the* ***status register***) to be updated if a ***packet*** is received from said ***serial bus keyboard.***
Claim 9	
9. The computer system of claim 6, further including a switch for enabling and disabling said keyboard controller emulator.	9. The computer system of claim 6, further including a switch for enabling and disabling said ***keyboard controller emulator.***
Claim 10	
10. The computer system of claim 6 wherein said serial bus keyboard includes a serial bus mouse.	10. The computer system of claim 6 wherein said ***serial bus keyboard*** includes a ***serial bus mouse.***
Claim 11	
11. A serial bus host controller for coupling a serial bus keyboard to a computer system via a standardized serial bus which transfers data in a packetized protocol, the serial bus host controller for sending and	11. A ***serial bus host controller*** for coupling a ***serial bus keyboard*** to a computer system via a standardized ***serial bus*** which transfers data in a ***packetized protocol,*** the ***serial bus host controller*** for sending and receiving ***serial bus packets,*** the ***serial bus host controller comprising:***

receiving serial bus packets, the serial bus host controller comprising:	
a keyboard controller emulator for generating and receiving data, status and commands pertaining to the serial bus keyboard, including:	(a) a ***keyboard controller emulator*** for generating and receiving ***data, status*** and commands pertaining to the ***serial bus keyboard, including:***
a serial bus address register for storing the serial bus address of the serial bus keyboard;	(a)(i) a ***serial bus address register*** for storing the ***serial bus address*** of the ***serial bus keyboard;***
a data input buffer;	(a)(ii) a ***data input buffer*** [*a* ***register*** *for temporarily storing* ***data*** *to be sent to the keyboard*];
a data output buffer, said keyboard controller providing an interrupt when said data output buffer is changed; and	(a)(iii) a ***data output buffer*** [*a* ***register*** *for temporarily storing* ***data*** *received from the keyboard*], said ***keyboard controller*** [*indefinite, unclear antecedent basis*] providing an ***interrupt*** [*a signal from a device to a computer's processor requesting attention from the processor*] when said ***data output buffer*** is changed; and
a status register; and	(a)(iv) a ***status register:*** and
a detector for detecting when the data buffers and said status register are accessed;	(b) a ***detector*** for detecting when the ***data buffers*** and said ***status register*** are ***accessed;***
packet parsing logic for determining if a packet is received from the serial bus keyboard;	(c) ***packet parsing logic for determining if a packet is received from the serial bus keyboard;*** [*a circuit for examining* ***packets*** *and sensing whether they are from the* ***serial bus keyboard*** *or not*]
data output buffer writing logic for writing into said data output buffer a data value extracted from a packet received from said serial bus keyboard; and	(d) ***data output buffer writing logic*** [*circuit that writes to the* ***data output buffer***] for writing into said ***data output buffer*** a ***data*** value extracted from a ***packet*** received from said ***serial bus keyboard;*** and
data input buffer reading logic for reading from said data input buffer a data value written therein by the computer system for said serial bus keyboard and developing a packet for transmission to said serial bus keyboard,	(e) ***data input buffer reading logic*** [*circuit that reads the* ***data input buffer***] for reading from said ***data input buffer*** a ***data*** value written therein by the computer system for said ***serial bus keyboard*** and developing a ***packet*** for transmission to said ***serial bus keyboard.***
wherein when said data value is written into said data output buffer a keyboard controller interrupt is generated.	(f) wherein when said ***data*** value is written into said ***data output buffer*** a keyboard controller ***interrupt*** is generated.
Claim 12	
12. The serial bus host controller of claim 11 wherein said serial bus keyboard includes a serial bus mouse.	12. The ***serial bus host controller*** of claim 11 wherein said ***serial bus keyboard*** includes a ***serial bus mouse.***
Claim 13	

13. The serial bus host controller of claim 11 wherein said interrupt is a system management interrupt.	13. The ***serial bus host controller*** of claim 11 wherein said ***interrupt*** is a ***system management interrupt.***
Claim 14	
14. The serial bus host controller of claim 11, further including a switch for enabling and disabling said keyboard controller emulator.	14. The ***serial bus host controller*** of claim 11, further including a switch for enabling and disabling said ***keyboard controller emulator.***
Claim 15	
15. A computer system for coupling to a serial bus keyboard via a standardized serial bus which transfers data in a packetized protocol, the computer system comprising:	15. A computer system for coupling to a ***serial bus keyboard*** via a standardized ***serial bus*** which transfers ***data*** in a ***packetized protocol.*** the computer system ***comprising:***
a serial bus keyboard; and	(a) a ***serial bus keyboard:*** and
a serial bus host controller coupled to said serial bus keyboard for sending and receiving serial bus packets, comprising:	(b) a ***serial bus host controller*** coupled to said ***serial bus keyboard*** for sending and receiving ***serial bus packets, comprising:***
a keyboard controller emulator for generating and receiving data, status and commands pertaining to the serial bus keyboard, including:	(b)(i) a ***keyboard controller emulator*** for generating and receiving ***data, status*** and commands pertaining to the ***serial bus keyboard, including:***
a serial bus address register for storing the serial bus address of the serial bus keyboard;	(b)(i)(A) a ***serial bus address register*** for storing the ***serial bus address*** of the ***serial bus keyboard:***
a data input buffer;	(b)(i)(B) a ***data input buffer:***
a data output buffer, said keyboard controller providing an interrupt when said data output buffer is changed; and	(b)(i)(C) a ***data output buffer,*** said ***keyboard controller*** [*indefinite, no antecedent basis*] providing an ***interrupt*** when said ***data output buffer*** is changed; and
a status register; and	(b)(i)(D) a ***status register;*** and
a detector for detecting when the data buffers and said status register are accessed;	(b)(ii) a ***detector*** for detecting when the ***data buffers*** and said ***status register*** are ***accessed:***
packet parsing logic for determining if a packet is received from the serial bus keyboard;	(b)(iii) ***packet parsing logic for determining if a packet is received from the serial bus keyboard;***
data output buffer writing logic for writing into said data output buffer a data value extracted from a packet received from said serial bus keyboard; and	(c) ***data output buffer writing logic*** for writing into said ***data output buffer*** a ***data*** value extracted from a ***packet*** received from said ***serial bus keyboard;*** and
data input buffer reading logic for reading from said data input buffer	(d) ***data input buffer reading logic*** for reading from said ***data input buffer*** a ***data*** value written therein by the computer system for said

a data value written therein by the computer system for said serial bus keyboard and developing a packet for transmission to said serial bus keyboard,	***serial bus keyboard*** and ***developing*** a ***packet*** for transmission to said ***serial bus keyboard.***
wherein when said data value is written into said data output buffer a keyboard controller interrupt is generated.	(e) wherein when said ***data*** value is written into said ***data output buffer*** a keyboard controller ***interrupt*** is generated.
Claim 16	
16. The computer system of claim 15 wherein said serial bus keyboard includes a serial bus mouse.	16. The computer system of claim 15 wherein said ***serial bus keyboard*** includes a ***serial bus mouse.***
Claim 17	
17. The computer system of claim 15 wherein said interrupt is a system management interrupt.	17. The computer system of claim 15 wherein said ***interrupt*** is a ***system management interrupt.***
Claim 18	
18. The computer system of claim 15, further including a switch for enabling and disabling said keyboard controller emulator.	18. The computer system of claim 15, further including a switch for enabling and disabling said ***keyboard controller emulator.***
Claim 19	
19. A method of communicating with a serial bus keyboard over a standardized serial bus in a computer system having a serial bus host controller, said host controller including a keyboard controller emulator, the keyboard controller emulator including a queue and scheduler for communicating directly with the serial bus, the method comprising the steps of:	19. A method of communicating with a ***serial bus keyboard*** over a standardized ***serial bus*** in a computer system having a ***serial bus host controller,*** said host controller including a ***keyboard controller emulator,*** the ***keyboard controller emulator*** including a queue and scheduler for communicating directly with the ***serial bus,*** the method ***comprising*** the steps of;
(a) powering on the computer system;	(a) powering on the computer system;
(b) enabling a keyboard controller queue and scheduler;	(b) enabling a ***keyboard controller queue and scheduler*** [*a circuit or device which can be hardware or hardware and software, that contains queues and that schedules data for transmission to and from a* ***serial bus keyboard***]:
(c) said keyboard controller queue and scheduler polling the serial bus for a serial bus keyboard;	(c) said ***keyboard controller queue and scheduler polling*** [*interrogating*] ***serial bus*** for a ***serial bus keyboard;***
(d) determining an address of said serial bus keyboard;	(d) determining an ***address*** [*an identifier designating a particular device*] of said ***serial bus keyboard;***

(e) storing said serial bus keyboard address in a register;	(e) storing said ***serial bus keyboard address*** [*the address of a serial bus keyboard*] in a ***register;***
(f) said keyboard controller emulator processing packets, if any, with the serial bus keyboard until the host controller is initialized;	(f) said ***keyboard controller emulator*** processing packets, if any, with the ***serial bus keyboard*** until the host controller is ***initialized*** [*commencing operation*]:
(g) loading a host controller device driver; and	(g) loading a ***host controller device driver*** [*software that enables communication between the processor and a device through a host controller*]; and
(h) disabling said keyboard controller queue and scheduler after said host controller device driver is loaded.	(h) disabling said ***keyboard controller queue and scheduler*** after said ***host controller device driver*** is loaded.
Claim 20	
20. The method of claim 19, wherein said keyboard controller emulator includes a data input buffer, the method further comprising the steps of:	20. The method of claim 19, wherein said ***keyboard controller emulator*** includes a ***data input buffer,*** the method further ***comprising*** the steps of:
(i) said keyboard controller queue and scheduler generating a packet for said serial bus keyboard if data is received into said data input buffer before said device driver is loaded; and	(i) said ***keyboard controller queue and scheduler*** generating a ***packet*** for said ***serial bus keyboard*** if ***data*** is received into said ***data input buffer*** before ***said device driver*** (*the host controller device driver*) is loaded; and
(j) said keyboard controller queue and scheduler transmitting said packet for said serial bus keyboard utilizing the stored address if data is received into said data input buffer before said device driver is loaded.	(j) said ***keyboard controller queue and scheduler*** transmitting said ***packet*** for said ***serial bus keyboard*** utilizing ***the stored address*** (*the serial bus keyboard address stored in the register of claim element 19(e)*) if ***data*** is received into said ***data input buffer*** before said ***device driver*** is loaded.
Claim 21	
21. The method of claim 19, wherein said keyboard controller emulator includes a data output buffer, the method further comprising the steps of:	21. The method of claim 19, wherein said ***keyboard controller emulator*** includes a ***data output buffer,*** the method further ***comprising*** the steps of:
(k) placing keyboard data into said data output buffer if a packet is received from said serial bus keyboard before said device driver is loaded; and	(k) placing keyboard ***data*** into said ***data output buffer*** if a ***packet*** is received from said ***serial bus keyboard*** before ***said device driver*** (*the host controller device driver*) is loaded; and
(l) generating an interrupt to said computer system if data is placed into said data output buffer.	(l) generating an ***interrupt*** to said computer system if ***data*** is placed into said ***data output buffer.***
Claim 22	

22. A method of communicating with a serial bus keyboard over a standardized serial bus in a computer system having a serial bus host controller and a basic input/output system (BIOS) for controlling device initialization, said host controller including a keyboard controller emulator, the method comprising the steps of:	22. A method of communicating with a ***serial bus keyboard*** over a standardized ***serial bus*** in a computer system having a ***serial bus host controller*** and a basic input/output system (BIOS) for controlling device ***initialization*** [*commencement of operation*], said host controller including a ***keyboard controller emulator,*** the method comprising the steps of:
(a) powering on the computer system;	(a) powering on the computer system;
(b) disabling the keyboard controller emulator;	(b) disabling the ***keyboard controller emulator;***
(c) the BIOS polling the serial bus for a serial bus keyboard;	(c) the BIOS ***polling*** the ***serial bus*** for a ***serial bus keyboard;***
(d) the BIOS determining an address of said serial bus keyboard;	(d) the BIOS determining an ***address*** of said ***serial bus keyboard;***
(e) the BIOS storing said serial bus keyboard address in a register;	(e) the BIOS storing said ***serial bus keyboard address*** in a ***register;***
(f) the BIOS enabling the keyboard controller emulator;	(f) the BIOS enabling the ***keyboard controller emulator;***
(g) the BIOS handling any communications between the computer system and the serial bus keyboard until a host controller device driver is loaded;	(g) the BIOS handling any communications between the computer system and the ***serial bus keyboard*** until a ***host controller device driver*** is loaded;
(h) loading a host controller device driver; and	(h) loading a host ***controller device driver;*** and
(i) disabling the keyboard controller emulator.	(i) disabling the ***keyboard controller emulator.***
Claim 23	
23. The method of claim 22, wherein said keyboard controller emulator includes a data input buffer, the method further comprising the step of:	23. The method of claim 22, wherein said ***keyboard controller emulator*** includes a ***data input buffer,*** the method further ***comprising*** the step of:
(j) the BIOS generating a packet for said serial bus keyboard if data is received into said data input buffer before said device driver is loaded.	(j) the BIOS generating a ***packet*** for said ***serial bus keyboard*** if ***data*** is received into said ***data input buffer*** before ***said device driver*** (*the host controller device driver*) is loaded.
Claim 24	
24. The method of claim 22, wherein said keyboard controller emulator includes a data output buffer, the method further	24. The method of claim 22, wherein said ***keyboard controller emulator*** includes a ***data output buffer,*** the method further ***comprising*** the steps of:

comprising the steps of:	
(k) the BIOS placing keyboard data into said data output buffer if a packet is received from said serial bus keyboard before said device driver is loaded; and	(k) the BIOS placing keyboard ***data*** into said ***data output buffer*** if a ***packet*** is received from said ***serial bus keyboard*** before ***said device driver*** (*the host controller device driver*) is loaded; and
(l) generating an interrupt to said computer system if data is placed into said data output buffer.	(l) generating an ***interrupt*** to said computer system if ***data*** is placed into said ***data output buffer.***
Claim 25	
25. A method of communicating with a serial bus keyboard over a standardized serial bus in a computer system having a serial bus host controller, said host controller including a keyboard controller emulator, the method comprising the steps of:	25. A method of communicating with a ***serial bus keyboard*** over a standardized ***serial bus*** in a computer system having a ***serial bus host controller,*** said host controller including a ***keyboard controller emulator,*** the method ***comprising*** the steps of:
(a) powering on the computer system;	(a) powering on the computer system;
(b) polling the serial bus for a serial bus keyboard;	(b) ***polling*** the ***serial bus*** for a ***serial bus keyboard;***
(c) determining an address of said serial bus keyboard;	(c) determining an ***address*** of said ***serial bus keyboard;***
(d) storing said serial bus keyboard address in a register;	(d) storing said ***serial bus keyboard address*** in a ***register;***
(e) loading a host controller keyboard device driver;	(e) loading a ***host controller keyboard device driver*** [*software that enables communication between the processor and a* ***serial bus keyboard*** *through a host controller*];
(f) generating a system management interrupt (SMI) if keyboard data is received by the host controller; and	(f) generating a ***system management interrupt (SMI)*** if keyboard ***data*** is received by the host controller; and
(g) processing the keyboard data.	(g) processing the keyboard ***data.***
Claim 26	
26. The method of claim 25, wherein step (f) further comprises the steps of:	26, The method of claim 25, wherein step (f) further ***comprises*** the steps of:
(h) receiving a serial bus packet; and	(h) receiving a ***serial bus packet;*** and
(i) determining if said serial bus packet contains keyboard data.	(i) determining if said ***serial bus packet*** contains keyboard ***data.***
Claim 27	
27. The method of claim 25, wherein step (g) is performed in system management mode.	27. The method of claim 25, wherein step (g) is performed in system management mode.
Claim 28	

28. The method of claim 25, wherein the keyboard emulator includes a data output buffer and a status register, and wherein step (g) further comprises the steps of:	28. The method of claim 25, wherein ***the keyboard emulator*** (*the keyboard controller emulator*) includes a ***data output buffer*** and a ***status register,*** and wherein step (g) further ***comprises*** the steps of:
(j) writing the keyboard data into the keyboard output buffer if keyboard data is received by the host controller;	(j) writing the keyboard ***data*** into ***the keyboard output buffer*** (*the* ***data output buffer***) if keyboard ***data*** is received by the host controller;
(k) setting the status register to reflect the data written to the keyboard output buffer; and	(k) setting the ***status register*** to reflect the data written to ***the keyboard output buffer*** (*the* ***data output buffer***); and
(l) providing a keyboard controller interrupt.	(l) providing a keyboard controller ***interrupt.***
Claim 29	
29. The method of claim 28, wherein the computer system further includes an SMI status register, the method further comprising the steps of:	29. The method of claim 28, wherein the computer system further includes an ***SMI status register,*** the method further ***comprising*** the steps of:
(m) reading the SMI status register before step (j); and	(m) reading the ***SMI status register*** before step (j); and
(n) determining a source of the system management interrupt from said SMI status register.	(n) determining a source of the ***system management interrupt*** from said ***SMI status register.***
Claim 30	
30. The method of claim 28, wherein said keyboard controller interrupt is simulated with an INT instruction.	30. The method of claim 28, wherein said keyboard controller ***interrupt*** is simulated with an ***INT instruction*** [*an instruction that causes the processor to behave as if the processor had received an* ***interrupt***].
Claim 31	
31. The method of claim 25, wherein the computer system further includes an SMI status register and wherein the keyboard emulator includes a data input buffer, the method further comprising the steps of.	31. The method of claim 25, wherein the computer system further includes an ***SMI status register*** and wherein the ***keyboard emulator*** (*the keyboard controller emulator*) includes a ***data input buffer,*** the method further ***comprising*** the steps of:
(o) generating a system management interrupt (SMI) if data is written to the keyboard input buffer;	(o) generating a ***system management interrupt (SMI)*** if ***data*** is written to ***the keyboard input buffer*** (*the data input buffer*);
(p) reading the SMI status register;	(p) reading the ***SMI status register;***
(q) determining a source of a system management interrupt from said SMI status register;	(q) determining a source of a ***system management interrupt*** from said ***SMI status register;***

(r) reading the keyboard input buffer if the source is the keyboard emulator; and	(r) reading ***the keyboard input buffer*** (*the data input buffer*) if the source is the ***keyboard emulator*** (*the keyboard controller emulator*); and
(s) forwarding the keyboard input buffer data to the host controller for transmission to the serial bus keyboard if the source is the keyboard emulator.	(s) forwarding ***the keyboard input buffer*** (*the data input buffer*) ***data*** to the host controller for transmission to the ***serial bus keyboard*** if the source is ***the keyboard emulator*** (*the keyboard controller emulator*).
Claim 32	
32. A method of communicating with a serial bus keyboard over a standardized serial bus in a computer system having a serial bus host controller, the host controller including a keyboard controller emulator, the keyboard controller emulator having a data output buffer and a status register, the method comprising the steps of:	32. A method of communicating with a ***serial bus keyboard*** over a standardized ***serial bus*** in a computer system having a ***serial bus host controller,*** the host controller including a ***keyboard controller emulator,*** the keyboard controller emulator having a ***data output buffer*** and a ***status register,*** the method comprising the steps of:
(a) initializing the host controller;	(a) ***initializing*** [*commencing operation*] the host controller;
(b) generating a host controller interrupt if a packet is received by the host controller;	(b) generating a ***host controller interrupt*** [*a signal from a host controller to a computer's processor, such as an* ***SMI*** *or a non-SMI* ***interrupt,*** *requesting attention from the processor*] if a ***packet*** is received by the host controller;
(c) determining a source of the packet;	(c) determining a source of the ***packet;***
(d) writing packet data into the data output buffer if the packet source is the serial bus keyboard;	(d) writing ***packet data*** into the ***data output buffer*** if the ***packet*** source is the ***serial bus keyboard;***
(e) updating the status register if data is written into the data output buffer; and	(e) updating the ***status register*** if ***data*** is written into the ***data output buffer;*** and
(f) generating a keyboard controller interrupt.	(f) generating a keyboard controller ***interrupt.***
Claim 33	
33. The method of claim 32 wherein said keyboard controller interrupt is simulated with an INT instruction.	33. The method of claim 32 wherein said keyboard controller ***interrupt*** is simulated with an ***INT instruction.***

EXHIBIT B

UNITED STATES PATENT NUMBER 5,802,318-GLOSSARY OF TERMS

CLAIM TERM	**DEFINITION**
Accessed	written into or read from
Address	an identifier designating a particular device

Buffer	a register for temporarily storing data
Comprising	including but not limited to
Data	commands and information pertaining to the keyboard
Data input buffer reading logic	circuit that reads the data input buffer
Data output buffer writing logic	circuit that writes to the data output buffer
Detector	a circuit for sensing
Detector further detects if a packet is received from said serial bus keyboard	the detector also senses if a packet is from the serial bus keyboard or not
Host controller device driver	software that enables communication between the processor and a device through a host controller
Host controller interrupt	a signal from a host controller to a computer's processor, such as an SMI or a non-SMI interrupt, requesting attention from the processor
Host controller keyboard device driver	software that enables communication between the processor and a serial bus keyboard through a host controller
Including	at least but not limited to
Initialization	commencement of operation
Initialized	commencing operation
Initializing	commencing operation
Input buffer	a register for temporarily storing data to be sent to the keyboard
INT instruction	an instruction that causes the processor to behave as if the processor had received an interrupt
Interrupt	a signal from a device to a computer's processor requesting attention from the processor
Interrupt generator	a circuit device or code that creates a signal requesting attention from the processor
Keyboard controller emulator	hardware or hardware and software that imitates a keyboard controller in generating and receiving data, status and commands pertaining to a serial bus keyboard as if the keyboard controller were present
Keyboard controller queue and scheduler	a circuit or device which can be hardware or hardware and software, that contains queues and that schedules data for transmission to and from a serial bus keyboard
Output buffer	a register for temporarily storing data received from the keyboard
Packet parsing logic for determining if a packet is received from the serial bus keyboard	a circuit for examining packets and sensing whether they are from the serial bus keyboard or not
Packetized protocol	a set of rules governing data packet transmission
Polling	interrogating
Register	a device, other than main memory, which holds a set of data bits for a particular purpose
Serial bus	a hardware line used for transferring data in a bit stream among the components of a computer system

Serial bus address	an identifier designating a particular device on the serial bus
Serial bus host controller	circuitry that supports a device's communication over a serial bus by connecting the serial bus with a bus in the computer system
Serial bus keyboard	a keyboard that communicates over a serial bus
Serial bus keyboard address	the address of a serial bus keyboard
Serial bus mouse	a computer mouse that communicates over a serial bus
Serial bus packets	bundles of data organized in groups for transmission over the serial bus
SMI	System Management Interrupt
Status	report of data pertaining to the keyboard and/or mouse
System management interrupt	a signal provided to the computer system when the data buffer and the status register are accessed

FN1. All terms appearing in bold face type and underlined have been construed by the court and appear with their definitions in the glossary in Exhibit B. The definition for each construed term appears in italics after its first use in the patent.

S.D.Cal.,2006.
Hewlett-Packard Co. v. Gateway, Inc.