United States District Court,
S.D. California.

HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P,
Plaintiff.
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)

Jan. 26, 2005.

John Allcock, DLA Piper US, San Diego, CA, for Plaintiff/Counterclaim-Defendants.

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

CLAIM CONSTRUCTION ORDER FOR UNITED STATES PATENT NUMBER 5,353,415

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

Pursuant to Markman v. Westview Instruments, Inc., 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996), on January 11-13, 2005, 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 5,353,415 ("the '415 patent"). Plaintiff Hewlett-Packard Development Company, L.P. ("HP") was 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 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 '415 patent. Additionally, the Court prepared a case glossary for terms found in the claims and the specification for the '415 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 '415 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

UNITED STATES PATENT NUMBER 5.353.415-CLAIM CHART

VERBATIM CLAIM LANGUAGE	COURT'S CLAIM CONSTRUCTION
Claim 1	
A computer system which performs concurrent bus cycle operations, comprising:	A computer system which performs *concurrent bus cycle operations* [*bus transactions occurring or pending on two or more buses at least partially at the same time*], comprising:
a host bus;	a *host bus* [*processor/memory bus, to which one or more CPU boards are coupled*];
an expansion bus coupled to said host bus;	an *expansion bus* [*bus that enables the expansion of a computer by providing an interface to devices*] coupled to said host bus;
expansion bus memory coupled to said expansion bus, said expansion bus memory storing data;	*expansion bus memory* [*a memory device situated on the expansion bus where information can be stored and retrieved*] coupled to said expansion bus, said expansion bus memory storing data;
a processor;	a *processor* [*the control unit of a computer*];
a cache system coupled to said processor and to said host bus, said cache system including a cache controller and cache memory, wherein said cache controller snoops said host bus when said cache controller does not control said host bus and generates write-back cycles when a snoop hit occurs to a dirty line in said cache memory, wherein said cache controller generates expansion bus read cycles when said processor requests data residing in said expansion bus memory that does not reside in said cache memory;	a *cache system coupled to said processor and to said host bus* [*a cache memory subsystem coupled to the processor and the host bus*], said cache system including a *cache controller* [*a device that, among other things, manages operation of the cache memory*] and *cache memory* [*a small amount of very fast, zero wait state memory that is used to stored frequently used code and data*], wherein said cache controller *snoops* [*monitors a bus for data requests and data writes directed to other devices on that bus*] said host bus when said cache controller does not control said host bus and generates *write-back cycles* [*bus transactions to provide updated data from the cache memory to the corresponding location in main memory and the requesting device*] when a snoop hit occurs to a *dirty line* [*data in a cache memory that has been modified but not yet updated in the corresponding main memory location(s)*] in said cache memory wherein said cache controller generates expansion bus read cycles when said processor requests data residing in said expansion bus memory that does not reside in said cache memory;
cycle generation means coupled to said cache system and said host bus for generating an expansion bus read cycle after said processor requests data residing in said expansion bus memory that does not reside in said	cycle generation means coupled to said cache system and said host bus for generating an expansion bus read cycle after said processor requests data residing in said expansion bus memory that does not reside in said cache memory;

cache memory;	**Means-plus-function claim:** The function of this limitation is: *generating an expansion bus read cycle after said processor requests data residing in said expansion bus memory that does not reside in said cache memory*.
	The structure disclosed to perform this function is: *EISA Bus Controller (EBC 40)*.
a bus master coupled to said host bus which generates cycles onto said host bus which must be snooped by said cache controller; and	a *bus master* [*a device capable of controlling the bus*] coupled to said host bus which generates cycles onto said host bus which must be snooped by said cache controller; and
an expansion bus controller coupled between said host bus and said expansion bus which latches the address of said expansion bus read cycle and obtains said requested data from said expansion bus memory, wherein said expansion bus controller obtains said requested data concurrently with said cache controller snooping said bus master host bus cycle.	an *expansion bus controller* [*a device that interfaces the expansion bus and the host bus*] coupled between said host bus and said expansion bus which *latches* [*stores*] the address of said expansion bus read cycle and obtains said requested data from said expansion bus memory, wherein said expansion bus controller obtains said requested data concurrently with said cache controller snooping said bus master host bus cycle.
Claim 2	
The computer system of claim 1, wherein when said bus master host bus cycle causes a snoop hit to a dirty line in said cache memory, said cache controller performs a write-back cycle on said host bus concurrently with said expansion bus controller obtaining said requested data.	The computer system of claim 1, wherein when said bus master host bus cycle causes a snoop hit to a dirty line in said cache memory, said cache controller performs a write-back cycle on said host bus *concurrently* [*at least partially at the same time*] with said expansion bus controller obtaining said requested data.
Claim 4	
A computer system which performs concurrent bus cycle operations, comprising:	A computer system which performs concurrent bus cycle operations, comprising:
a host bus;	a host bus;
a processor coupled to said host bus;	a processor coupled to said host bus;
a plurality of processor ports storing data;	a plurality of processor ports storing data;
a local I/O bus coupled between said processor and said processor ports;	a *local I/O bus* [*an input/output bus. separate from the host bus, that transfers data between the processor and processor ports without using the host bus*] coupled between said processor and said processor ports;
a cache system coupled between said local I/O bus and said processor and further coupled between said processor and said host bus, said cache system including a cache controller and cache	a cache system coupled between said local I/O bus and said processor and further coupled between said processor and said host bus, said cache system including a cache controller and cache memory, wherein said cache controller snoops said host bus when said cache controller does not control said host bus and generates

memory, wherein said cache controller snoops said host bus when said cache controller does not control said host bus and generates write-back cycles when a snoop hit occurs to a dirty line in said cache memory;	write-back cycles when a snoop hit occurs to a dirty line in said cache memory;
cycle generation means coupled to said cache system and said local I/O bus for generating a local I/O cycle after said processor writes or reads data to or from one of said processor ports that does not reside in said cache memory; and	cycle generation means coupled to said cache system and said local I/O bus for generating a local I/O cycle after said processor writes or reads data to or from one of said processor ports that does not reside in said cache memory; and **Means-plus-function claim:** The function of this limitation is: *generating a local I/O cycle after said processor writes or reads data to or from one of said processor ports that does not reside in said cache memory*.
	The structure disclosed to perform this function is: *local I/O control logic 140 found in Fig. 3, plus Fig. 4A as it relates to states LIO0, LI02-7, and transitions to and from those states, and col. 10, line 48-col. 14, line 20, as it relates to states LIO0, LI02-7,* and transitions to and from those states.
a bus master coupled to said host bus which generates cycles onto said host bus which must be snooped by said cache controller; and	a bus master coupled to said host bus which generates cycles onto said host bus which must be snooped by said cache controller; and
wherein said cache controller snoops said bus master host bus cycle approximately concurrently with said cycle generating means generating said local I/O cycle.	wherein said cache controller snoops said bus master host bus cycle approximately concurrently with said cycle generating means generating said local I/O cycle.

Claim 5

The computer system of claim 4, wherein said processor generates a processor cycle to said cache system concurrently with said cache controller snooping said bus master host bus cycle and said cycle generating means generating said local I/O cycle.	
The computer system of claim 4, wherein said processor generates a *processor cycle to said cache system* [*a bus transaction between the processor and the cache system*] concurrently with said cache controller snooping said bus master host bus cycle and said cycle	

generating means generating said local I/O cycle.	
Claim 6	
The computer system of claim 4, wherein said local I/O cycle is a postable write cycle writing data to one of said processor ports, the computer system further comprising:	The computer system of claim 4, wherein said local I/O cycle is a *postable write cycle* [*a write transaction to a destination device that passes through an intermediary device, where the intermediary device stores the address and data and returns a ready signal, with the expectation that the intermediary device will initiate a transaction to complete the write operation to the destination device*] writing data to one of said processor ports, the computer system further comprising:
a data buffer coupled between said cache memory and said processor ports which receives said write data from said cache memory; and	a data buffer coupled between said cache memory and said processor ports which receives said write data from said cache memory; and
wherein said cache controller snoops said bus master host bus cycle concurrently with said cache memory providing said write data to said data buffer.	wherein said cache controller snoops said bus master host bus cycle concurrently with said cache memory providing said write data to said data buffer.
Claim 7	
A method for performing concurrent operations in a computer system comprising a host bus; a processor coupled to the host bus; a bus master coupled to the host bus; a cache system coupled between the host bus and the processor, the cache system including a cache controller that snoops host bus cycles when said cache controller does not control said host bus and performs write back cycles when a snoop hit occurs to a dirty line in said cache memory an expansion bus coupled to the host bus; an expansion bus controller coupled between said host bus and said expansion bus; and cache interface logic which generates cache controller cycles into said host bus, the method comprising:	A method for performing concurrent operations in a computer system comprising a host bus; a processor coupled to the host bus; a bus master coupled to the host bus; a cache system coupled between the host bus and the processor, the cache system including a cache controller that snoops host bus cycles when said cache controller does not control said host bus and performs write-back cycles when a snoop hit occurs to a dirty line in said cache memory an expansion bus coupled to the host bus; an expansion bus controller coupled between said host bus and said expansion bus; and *cache interface logic* [*a logic circuit that provides an interface between the cache controller and the host bus, and initiates host bus transactions for the cache controller*] which generates cache controller cycles into said host bus, the method comprising;
a) the cache controller generating an expansion bus read cycle including an address of requested data;	a) the cache controller generating an expansion bus read cycle including an address of requested data;
b) the cache interface logic latching said address;	b) the cache interface logic latching said address;
c) the cache interface logic presenting said address onto the host bus after step	c) the cache interface logic *presenting said address onto the host bus* [*placing the address of the data to be read from the expansion*

b);	*bus memory onto the address line(s) of the host bus*] after step b);
d) the expansion bus controller latching said expansion bus read cycle address after step c);	d) the expansion bus controller latching said expansion bus read cycle address after step c);
e) the bus master performing a host bus cycle that is required to be snooped by the cache controller cacne controller	e) the bus master performing a host bus cycle that is required to be snooped by the cache controller
f) the expansion bus controller performing expansion bus cycles to obtain the requested data after step d); and	f) the expansion bus controller performing expansion bus cycles to obtain the requested data after step d); and
g) the cache controller snooping said bus master host bus cycle concurrently with step f).	g) the cache controller snooping said bus master host bus cycle concurrently with step f).
Claim 8	
The method of claim 7, further comprising:	The method of claim 7, further comprising:
h) the cache controller completing snooping said bus master host bus cycle after step g);	h) the cache controller completing snooping said bus master host bus cycle after step g);
I) the expansion bus controller presenting said requested data onto the host bus; and	I) the expansion bus controller presenting said requested data onto the host bus; and
j) the cache controller obtaining said requested data from the host bus after step I).	j) the cache controller obtaining said requested data from the host bus after step I).

EXHIBIT B

GLOSSARY OF TERMS

TERM	**DEFINITION**
bus	set of hardware lines (wires) used for data transfer among the components of a computer system
bus master	a device capable of controlling the bus
cache controller	a device that, among other things, manages operation of the cache memory
cache interface logic	a logic circuit that provides an interface between the cache controller and the host bus, and initiates host bus transactions for the cache controller
cache memory	a small amount of very fast, zero wait state memory that is used to stored frequently used code and data
cache system coupled to said processor and to said host bus	a small amount of very fast, zero wait state memory that is used to stored frequently used code and data
concurrent bus cycle operations	bus transactions occurring or pending on two or more buses at least partially at the same time

concurrently	at least partially at the same time
dirty line	data in a cache memory that has been modified but not yet updated in the corresponding main memory location(s)
expansion bus	bus that enables the expansion of a computer by providing an interface to devices
expansion bus controller	a device that interfaces the expansion bus and the host bus
expansion bus memory	a memory device situated on the expansion bus where information can be stored and retrieved
host bus	processor/memory bus, to which one or more CPU boards are coupled
latches	stores
local I/O bus	an input/output bus, separate from the host bus, that transfers data between the processor and processor ports without using the host bus
postable write cycle	a write transaction to a destination device that passes through an intermediary device, where the intermediary device stores the address and data and returns a ready signal, with the expectation that the intermediary device will initiate a transaction to complete the write operation to the destination device
presenting said address onto the host bus	placing the address of the data to be read from the expansion bus memory onto the address line(s) of the host bus
processor	the control unit of a computer
processor cycle to said cache system	bus transaction between the processor and the cache system
snoops	monitors a bus for data requests and data writes directed to other devices on that bus
write-back cycles	bus transactions to provide updated data from the cache memory to the corresponding location in main memory and the requesting device

S.D.Cal.,2005.
Hewlett-Packard Development Co., L.P. v. Gateway, Inc.

Produced by Sans Paper, LLC.