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. Intel Corp, Intervenor. Civil No. 04CV0613-B(LSP)

Dec. 12, 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, 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 5,596,759

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 1-3, 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 5,596,759 ("the '759 patent"). Plaintiff Hewlett-Packard Development Company, L.P. ("HP") was represented by the law firm of DLA Piper Rudnick Gray Cary U.S. LLP, Defendant Gateway, Inc. ("Gateway") was represented by the law firm of Dewey Ballantine LLP, and Intervenor Intel Corporation ("Intel") was represented by the law firm of Weil, Gotshal & Manges 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 '759 patent. Additionally, the Court prepared a case glossary for terms found in the claims and the specification for the '759 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 '759 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 5,596,759-CLAIM CHART

VERBATIM CLAIM	COURTS CONSTRUCTION
LANGUAGE	
Claim 1	
1. A method of initializing a	1. A method of initializing a multi-processor computer system, the multi-
multi-processor computer	processor computer system including at least two processors, [circuitry that
system, the multi-processor	has the ability to fetch, decode, and execute instructions and to transfer
computer system including at	information to and from other resources over the computer's main data
least two processors, one of	transfer path, the bus] one of which is considered a primary processor [the
which is considered a	processor that performs a complete <i>initialization</i> , among other things] during
primary processor during	<i>initialization</i> [the process by which the computer system is made ready for
initialization, said processors	use], said processors being powered up together [power is supplied to both
being powered up together;	processors at the same time]; common peripherals [hardware devices, other
common peripherals; and a	than the computer (processor, memory, and data paths), such as a hard disk
common storage element	drive or printer, that can be accessed by both processors]; and a common
which stores initialization	storage element [a device that stores data and that can be accessed by both
code used during start up of	processors] which stores initialization code [computer code used to start up
the computer system, the	the computer system] used during start up of the computer system, the
common storage element	common storage element being common to each processor, the initialization
being common to each	code having processor and common peripheral portions [the initialization
processor, the initialization	code has a section that is capable of initializing processors and a section that
code having processor and	is capable of initializing common peripherals], the method comprising the
common peripheral portions,	steps of:
the method comprising the	
steps of:	
(a) each processor executing	(a) each processor executing initialization code stored in the common storage
initialization code stored in	element [each processor executes initialization code stored in the designated
the common storage element:	device that stores data and that can be accessed by both processors];
(b) each processor	(b) each processor <i>performing</i> [<i>executing</i>] processor <i>initialization coae</i> to
initialization and to	
initialization code to	
(a) as a processor	(a) and any and the maining if it is the main and any forming
determining if it is the	(c) each processor determining in it is the <i>primary processor</i> and <i>performing</i>
nrimary processor and	common peripheral <i>initialization code</i> only if it is said <i>primary processor</i> .
primary processor and performing common	
perioding collinoli	
peripheral initialization code	

only if it is said primary				
Claim 6				
6 The method of alaim 1 further 6 The	mathed of alaim 1 further comprising the stops of			
o. The method of claim 1, further o. The comprising the steps of:	method of claim 1, further comprising the steps of.			
(h) applying and releasing a system.	lying and releasing a system reset [a signal that causes the			
(ii) applying and releasing a system (ii) appreset to the multiprocessor computer system	to return to the state it would be in if the power were turned off			
system before steps (a) (c) the and then on again 1 to the multiprocessor computer system before				
system before steps (a)-(c), the and then on again 1 to the multiprocessor computer system before system reset causing all but one of steps (a) (c) the system reset causing all but one of said processors to				
said processors to be restrained; and be restrained [held in an inactive state]; and				
(i) said primary processor causing (i) said primary processor causing each other processor to be				
each other processor to be released. release	d.			
Claim 7				
7 The method of claim 6 wherein the multi-	7 The method of claim 6 wherein the multi processor			
processor computer system further includes	computer system further includes an <i>active processor</i>			
an active processor identifying value, the	identifying value [the value that indicates which one of the			
method further including the steps of:	processors. if any, is currently active], the method further			
	including the steps of:			
(j) each processor acquiring the active	(j) each processor acquiring the <i>active processor identifying</i>			
processor identifying value; and	value; and			
(k) each processor selecting a portion of said	(k) each processor selecting a portion of said <i>initialization</i>			
initialization code to execute based on the	code to execute based on the active processor identifying			
active processor identifying value,	value,			
wherein each processor is identified as a	wherein each processor is identified as a primary or secondary			
primary or secondary processor, and	processor, and			
wherein each processor identified as a	wherein each processor identified as a <i>primary processor</i>			
primary processor initializes the common	initializes the <i>common peripherals</i> and each processor			
peripherals and each processor identified as	a identified as a secondary processor does not initialize the			
secondary processor does not initialize the	common peripherals.			
common peripherals.				
Claim 8				
8. The method of claim 6, wherein the	8. The method of claim 6, wherein the multiprocessor computer			
multi-processor computer system further	system further includes a main memory for storing <i>redirection</i>			
includes a main memory for storing	vectors [an address obtained from a memory location that			
redirection vectors and wherein step (1)	directs the processor to a new memory location containing			
further comprises the steps of:	<i>code to be executed</i>] and wherein step (1) further comprises the			
(1)	steps of:			
(1) said primary processor writing an	(1) said <i>primary processor</i> writing an address into a <i>reairection</i>			
address into a redirection vector location of	vector location of main memory [a location in main memory where a radiraction water is stored] and address pointing to a			
starting address of a portion of said	starting address of a portion of said <i>initialization code</i> not			
initialization code not causing the common	causing the <i>common peripherals</i> to be initialized: and			
peripherals to be initialized and	causing the common peripheruis to be initialized, and			
(m) said primary processor causing each	(m) said <i>primary processor</i> causing each other processor to be			
other processor to be released after setting	released after <i>setting the redirection vector</i> [storing the			
the redirection vector.	address of code to be executed in the redirection vector			

location].				
Claim 11				
11. A multiprocessor computer system,	11. A multi	processor computer system, comprising:		
comprising:				
at least two processors, one of which is	at least two	processors, one of which is considered a primary		
considered a primary processor during	processor d	uring initialization, said processors being powered up		
initialization, said processors being	together;			
powered up together;	-			
a common storage element containing	a <i>common</i> s	storage element containing processor executable		
processor executable initialization code	initialization code used during start-up of the computer system,			
used during start-up of the computer	the common storage element being common to each processor,			
system, the common storage element	said <i>initiali</i>	zation code having processor and common peripheral		
being common to each processor, said	initializatio	n code [the initialization code has a section that is		
initialization code having processor and	capable of i	initializing processors and a section that is capable of		
common peripheral initialization code;	initializing	common peripherals];		
a common peripheral including a hard	a common j	peripheral including a hard disk;		
disk;				
wherein said initialization code when	wherein sai	d <i>initialization code</i> when executed by said <i>processors</i>		
executed by said processors causes said	causes said	causes said <i>processors</i> to perform the steps of:		
processors to perform the steps of:				
(a) each processor executing said	(a) <i>each pro</i>	pcessor executing said initialization code stored in the		
initialization code stored in the common	common ste	orage element [each processor executes initialization		
storage element;	code stored	in the designated device that stores data and that can		
	be accessea	by both processors];		
(b) each processor performing said	(b) each pro	bcessor <i>performing</i> said processor <i>initialization code</i> to		
processor initialization code to initialize	initialize its	elf; and		
itself; and				
(c) each processor determining if it is the	(c) each pro	cessor determining if it is the <i>primary processor</i> and		
primary processor and performing said	performing	said common peripheral <i>initialization code</i> only if it is		
common peripheral initialization code	said <i>primar</i>	y processor.		
only if it is said primary processor.				
Claim 16				
16. The multiprocessor computer system	of claim 11	16. The multiprocessor computer system of claim 11,		
wherein said initialization code when exe	ecuted by	wherein said <i>initialization code</i> when executed by said		
said processors causes said processors to further		<i>processors</i> causes said <i>processors</i> to further perform		
perform the steps of:		the steps of:		
(h) applying and releasing a system reset to the		(h) applying and releasing a system reset to the		
multiprocessor computer system before steps (a)-(c),		multiprocessor computer system before steps (a)-(c),		
the system reset causing all but one of said		the system reset causing all but one of said processors		
processors to be restrained; and		to be <i>restrainea;</i> and		
(1) said primary processor causing each other		(1) said <i>primary processor</i> causing each other		
processor to be released.		<i>processor</i> to be released.		
<i>Claim 17</i>		17 The multi-measurement of the 14		
17. The multiprocessor computer system of claim		17. The multiprocessor computer system of claim 16,		
10, wherein the multi-processor computer system		wherein the multi-processor computer system further		
further includes an active processor identifying		includes an <i>active processor identifying value</i> and		

value and wherein said initialization code when	wherein said <i>initialization code</i> when executed by said	
executed by said processors causes said processors to	processors causes said processors to further perform	
further perform the steps of:	the steps of:	
(j) each processor acquiring the active processor	(j) each processor acquiring the <i>active processor</i>	
identifying value; and	identifying value; and	
(k) each processor selecting a portion of said	(k) each processor selecting a portion of said	
initialization code to execute based on the active	initialization code to execute based on the active	
processor identifying value,	processor identifying value,	
wherein each processor is identified as a primary or	wherein each processor is identified as a primary or	
secondary processor, and	secondary processor, and	
wherein each processor identified as a said primary	wherein each processor identified as a said <i>primary</i>	
processor initializes the common peripherals and	processor initializes the common peripherals and each	
each processor identified as a secondary processor	processor identified as a secondary processor does not	
does not initialize the common peripherals.	initialize the <i>common peripherals</i> .	
Claim 18		
18. The multiprocessor computer system of claim 16	, 18. The multiprocessor computer system of claim 16,	
wherein the multi-processor computer system further	wherein the multi-processor computer system further	
includes a main memory for storing redirection includes a main memory for storing <i>redirection</i>		
vectors and wherein step (i) of said initialization code vectors and wherein step (i) of said initialization code		
when executed by said processors causes said	when executed by said <i>processors</i> causes said	
processors to further perform the steps of:	<i>processors</i> to further perform the steps of:	
(1) said primary processor writing an address into a	(1) said <i>primary processor writing an address</i> into a	
redirection vector location of main memory, said	redirection vector location of main memory, said	
address pointing to a starting address of a portion of	address pointing to a starting address of a portion of	
said initialization code not causing the common	said <i>initialization code</i> not causing the <i>common</i>	
peripherals to be initialized; and	<i>peripherals</i> to be initialized; and	
(m) said primary processor causing each other	(m) said <i>primary processor</i> causing each other	
processor to be released after setting the	processor to be released after setting the redirection	
redirection vector.	vector.	

EXHIBIT B

UNITED STATES PATENT NUMBER 5,596,759-GLOSSARY OF TERMS

TERM	DEFINITION
Active processor identifying value	the value that indicates which one of the processors, if any, is currently active
Common peripherals	hardware devices, other than the computer (processor, memory, and data paths), such as a hard disk drive or printer, that can be accessed by both processors
Common storage element	a device that stores data and that can be accessed by both processors
Each processor executing initialization code stored in the common storage element	each processor executes initialization code stored in the designated device that stores data and that can be accessed by both processors
Each processor executing said initialization code stored in the	each processor executes initialization code stored in the designated device that stores data and that can be accessed by both processors

common storage element	
Initialization	the process by which the computer system is made ready for use
Initialization code	computer code used to start up the computer system
Initialization code having	the initialization code has a section that is capable of initializing
processor and common peripheral	processors and a section that is capable of initializing common
initialization code	peripherals
Performing	executing
Primary processor	the processor that performs a complete initialization, among other things
Processors	circuitry that has the ability to fetch, decode, and execute instructions and to transfer information to and from other resources over the computer's main data-transfer path, the bus
Redirection vectors	an address obtained from a memory location that directs the processor to a new memory location containing code to be executed
Redirection vector location of	a location in main memory where a redirection vector is stored
main memory	
Restrained	held in an inactive state
Said processors being powered up together	power is supplied to both processors at the same time
Setting the redirection vector	storing the address of code to be executed in the redirection vector location
System reset	a signal that causes the system to return to the state it would be in if the power were turned off and then on again
The initialization code having	the initialization code has a section that is capable of initializing
processor and common peripheral portions	processors and a section that is capable of initializing common peripherals

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.,2005. Hewlett-Packard Development Company, L.P. v. Gateway, Inc.

Produced by Sans Paper, LLC.