Trials@uspto.gov
571-272-7822

Paper No. 47
Date Entered: October 27, 2016

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMERICAN MEGATRENDS, INC.,
MICRO-STAR INTERNATIONAL CO., LTD,
MSI COMPUTER CORP.,
GIGA-BYTE TECHNOLOGY CO., LTD., and
G.B.T., INC.,
Petitioners,

v.

KINGLITE HOLDINGS, LLC,
Patent Owner.

Case IPR2015-01081
Patent 5,987,604

Before GLENN J. PERRY, TREVOR M. JEFFERSON, and BRIAN J. McNAMARA, Administrative Patent Judges.

McNAMARA, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and
37C.F.R. § 42.73

BACKGROUND

On October, 29, 2015, we instituted an *inter partes* review of claims 1–30 of U. S. Patent No. 5,987,604 ("the '604 Patent"). Paper 18 ("Dec. to Inst."). Patent Owner filed a Patent Owner Response (Paper 30, "PO Resp."), Petitioner filed a Petitioner Reply (Paper 39, "Pet. Reply"). Patent Owner filed a Motion to Exclude (Paper 23, "Mot. To Exclude"), Petitioner filed an Opposition (Paper 43, "Opp. To Mot. To Exclude"), and Patent Owner filed a Reply (Paper 44 "Reply to Opp. to Mot. to Exclude"). A transcript of an oral hearing held on June 29, 2016 (Paper 46, "Hr'g Tr.") has been entered into the record.

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is issued pursuant to 35 U.S.C. §318(a). We base our decision on the preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d).

Having reviewed the arguments of the parties and the supporting evidence, we conclude that Petitioner has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable.

THE '604 PATENT

The '604 Patent discloses an x86 based system in which a system management interrupt (SMI) invokes an operating mode known as system management mode (SMM). Ex. 1001, col. 3, ll. 10–20. In SMM, system firmware is used to perform power management or control other system functions, e.g., controlling hardware specific features, in a manner that is transparent to the operating system and applications software. *Id.*

The '604 Patent discloses creating a virtual monitor that runs under SMM with memory paging to execute SMI code in virtual mode, thereby allowing SMM code written to run below the 1 MB boundary to execute

above the 1 MB boundary by mapping the entire SMI handler above the 1 MB boundary and executing the code as a page enabled, protected mode, virtual task with SMM. *Id.* at col. 2, ll. 27–35. When an SMI is generated, the central processing unit (CPU) asserts an SMI activate (SMIACT) control signal to access system management random access memory (SMRAM) 20, which is a dedicated memory space not accessible to the operating system or application software. *Id*. at col. 3, ll. 21–27, col. 6, ll. 6–10. After storing the current operating location for return purposes in SMRAM 20, CPU 14 jumps to a location in SMRAM 20 to execute the SMI handler code that performs the system management activity. *Id.* at col. 3, ll. 27–31. Upon completion of the system management activity, control is returned to the previously interrupted operating system. *Id*. at col 3, ll. 31–37.

The ’604 Patent notes that conventional SMIs execute slowly because they operate below the 1 MB boundary in an uncached memory area, in order to avoid cache conflict with overlapping memory. *Id.* at col. 1, ll. 26–31. Such conventional SMMs operate in real mode with 4GB memory segments and 1 MB addressable program memory and are relatively slow. According to the ’604 Patent, a conventional alternative approach transfers SMI code and data from a region above the 1 MB boundary to a cacheable region below the 1 MB boundary, allowing standard execution of SMI code in real mode, but requiring transfer of the SMI code back to a cacheable area above the 1 MB area after completing the SMI. *Id*. at col. 1, ll. 37–45.

In accordance with the invention in the ’604 Patent, the data and code in SMRAM 20 located in the physical memory is mapped to location 22, which is above the 1 MB region in system space 50. *Id*. at col. 6, ll. 10–14, col. 7, ll. 27–30. CPU 14 then jumps to a location in SMRAM 20 that is

above the 1 MB region in the system address space 50 to execute the SMI handler code. *Id.* at col. 6, l. 15–19, col. 7, ll. 32–37. The SMI handler then jumps to location 24, which must be within 64 K of location 22, where the CPU mode change takes place, configuring CPU 14 to operate in protected mode. *Id*. at col 6, ll. 20–28.

The SMI handler then creates page tables and is configured to create a task state segment (TSS) and input/output (I/O) bit map, unless they were created during power on self-test (POST). *Id*. at col. 6, ll. 31– 39. The SMI handler then invokes the paging feature of CPU 14 and switches to virtual mode. *Id.* at col. 6, ll. 53–54.

In virtual mode, CPU 14 determines logical operations the same way as in real mode and the physical address is determined using the page tables. *Id.* at col. 6, l. 66–col. 7, l. 1. The SMI handler begins the task required to perform system management activities. *Id.* at col. 7, ll. 1–3. During execution of system management activities, the software occasionally configures the processor to operate in protected mode to facilitate execution of specific tasks, such as cache flushes and accessing the floating point unit of the CPU. *Id.* at col. 7, ll. 3–6. Upon completion of the special task, the SMI handler configures the CPU to operate in virtual mode to resume system management activity. *Id.* at col. 7, ll. 12–14.

Upon completion of system management activity, the SMI handler configures the CPU to exit the virtual mode, causing the CPU to operate in the protected mode. *Id.* at col. 7, ll. 14–19. Paging is then disabled and the SMI handler executes a resume instruction to return control to the previously interrupted operating system or application execution. *Id*. at col 7, ll. 18–22.

ILLUSTRATIVE CLAIM

Claims 1, 11 and 21 are independent. Claim 1 is drawn to an apparatus, claim 11 is drawn to a method, and claim 21 is drawn to a computer-executable process. Claim 1 is representative:

> 1. An apparatus for executing instructions in a system management mode in a processor-based system, comprising:
> a memory for storing instruction sequences by which the processor-based system is processed;
> a processor having a system address space, the processor for executing the stored instruction sequences; and
> wherein the stored instruction sequences cause the processor to: (a) configure the processor to operate in a protected mode while in system management mode, the processor operating at an address greater than one megabyte; (b) invoke a paging feature of the processor; (c) configure the processor to operate in a virtual mode; and (d) process the instruction sequences stored, wherein the process steps occur upon the receipt of an instruction to process a system management request.

Independent claims 11 and 21 recite as limitations parts (a)–(d) of the wherein clause of claim 1.

ART CITED IN PETITIONER'S CHALLENGES

Petitioner cited the following references in its challenges to patentability:

Reference	Designation	Exhibit No.
U.S. Patent No. 5,644,755, iss. July 1, 1997.	Wooten	Ex. 1002
U.S. Patent No. 6,093,213, iss. July 25, 2000.	Favor	Ex. 1003

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.