

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NXP USA, INC.,

Petitioner,

v.

MEDIATEK INC.,

Patent Owner.

Case IPR2022-00759

Patent 8,464,037 B2

DECLARATION OF DR. VIJAY K. MADISETTI

TABLE OF CONTENTS

I.	INTRODUCTION AND SCOPE OF WORK	1
II.	PROFESSIONAL BACKGROUND.....	2
III.	COMPENSATION	6
IV.	LEGAL CONSIDERATIONS	7
V.	TASK SUMMARY AND BASIS FOR OPINIONS	10
VI.	FIELD OF TECHNOLOGY	11
	A. Secure Hardware	12
	B. Secure Boot	13
VII.	LEVEL OF ORDINARY SKILL IN THE ART AT THE TIME OF THE INVENTION.....	13
VIII.	OVERVIEW OF THE '037 PATENT	15
	A. Overview	15
	B. Challenged Claims	23
	C. Prosecution History	26
IX.	COMPARISON OF THE PRIOR ART TO THE CHALLENGED CLAIMS	27
	A. Smith.....	28
	1. Smith versus Claims 19-21, 1-3 and 5-6.....	28
	a) Limitations 19[pre] and 19[a]: “A central processing unit (CPU), comprising: a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory; and”	28
	(1) EDRAM 111 of chip 105 is a volatile random access memory	30
	(2) ROM 113 of chip 105 is a non-volatile memory.....	30

(3)	A POSA would have considered it logical and straightforward for chip 105 to have included a CPU core.....	31
(4)	A POSA would have considered it logical and straightforward for chip 105 to have included a bus system.....	33
(5)	“Chip” 105’s components would have been on a substrate	35
b)	Limitation 19[b]: “pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key.”	35
(1)	Certain codes 115	35
(2)	UID 119	37
c)	Claim 20: “The central processing unit of claim 19, further comprising a locking mechanism configured to enable a portion of said non-volatile memory containing said symmetric encryption key upon one of a reset event and a power-on event only.”	43
d)	Limitation 21[pre]: “A computer system, comprising:”	46
e)	Limitation 21[a]: “a central processing unit comprising:”	47

f)	Limitations 21[b][i]-21[b][ii]: “a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory, and pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key; and”	47
g)	Limitation 21[c]: “a non-volatile boot memory configured to hold at least said boot routine and a second boot routine.”	47
h)	Limitation 1[pre]: “A method for booting a computer system, the method comprising:”	48
i)	Limitation 1[a]: “upon one of a power-on and a reset event, executing instructions stored in an internal non-volatile memory area of a central processing unit,”	48
j)	Limitation 1[b]: “said internal non-volatile memory area containing an encryption key;”	52
k)	Limitation 1[c]: “determining a current signature of at least a signed portion of boot data,”	52
l)	Limitation 1[d]: “said signed portion of boot data containing a stored signature of said portion of boot data,”	54
m)	Limitation 1[e]: “said stored signature created by applying said encryption key; and”	54
n)	Limitation 1[f]: “executing boot instructions contained in said signed portion of boot data, when said current signature and said stored signature are identical.”	55

o)	Claim 2: “The method of claim 1, wherein executing said instructions stored in said internal non-volatile memory area comprises initializing an internal random access memory of said central processing unit and loading said at least a portion of boot data from a non-volatile memory into said initialized random access memory.”.....	56
p)	Claim 3: “The method of claim 2, wherein executing boot instructions of said signed portion of boot data comprises initializing a system memory of said computer system and loading said boot data from said non-volatile memory into said system memory and verifying an integrity of said boot data.”.....	56
q)	Claim 5: “The method of claim 1, further comprising locking said encryption key for external access prior to executing said boot instructions.”.....	61
r)	Claim 6: “The method of claim 5, further comprising determining whether an upgrade version of said boot data is available.”.....	62
B.	Smith and Bolotin.....	65
1.	Smith and Bolotin versus Claim 20	65
a)	Overview of Bolotin	66
b)	Motivation to modify Smith’s chip 105 chip to include circuitry like that provided by Bolotin’s micro-controller 212 and locking function controller 215.....	68
c)	Smith’s chip 105, modified in view of Bolotin, meets claim 20	69
C.	Smith and Little Versus Claims 19 and 21, and Smith, Bolotin, and Little Versus Claim 20.....	70
D.	Smith and Challenger Versus Claims 5 and 6.....	71
1.	Overview of Challenger.....	72
2.	A POSA would have been motivated to modify Smith in view of Challenger.....	73

3.	Modification of Smith in view of Challenger.....	74
4.	Claim 5: “The method of claim 1, further comprising locking said encryption key for external access prior to executing said boot instructions.”	76
5.	Claim 6: “The method of claim 5, further comprising determining whether an upgrade version of said boot data is available.”	76

LIST OF DOCUMENTS CONSIDERED

Exhibit	Shorthand	Description
1001	'037 Patent	U.S. Patent No. 8,464,037 B2
1003	Challener	U.S. Patent App. Pub. No. US 2003/0041254 A1
1004	Smith	U.S. Patent App. Pub. No. US 2008/0165952 A1
1005	Stevens	U.S. Patent No. 6,633,976 B1
1006	Bolotin	WIPO Int'l Pub. No. WO 2007/014074 A1
1007	Petković	Petković and Jonker (Eds.), <i>Security, Privacy, and Trust in Modern Data Management</i> , Springer (2007)
1010	Amazon	November 1, 2007 archived copy of https://www.amazon.com/Security-Privacy-Management-Data-Centric-Applications/dp/3540698604 from https://web.archive.org/web/20071101011130/https://www.amazon.com/Security-Privacy-Management-Data-Centric-Applications/dp/3540698604
1011	Archive	Declaration of Nathaniel E Frank-White of the Internet Archive
1012	Little	U.S. Patent App. Pub. No. US 2006/0112266 A1
1013	Chen	U.S. Patent No. 7,546,447 B2
1014	Bhatt	U.S. Patent App. Pub. No. US 2006/0107321 A1
1015	Kumar	U.S. Patent App. Pub. No. US 2008/0159541 A1
1016	'037 PH	Prosecution history of U.S. Patent No. 8,464,037 B2
1017	Microsoft	Excerpts of <i>Microsoft Computer Dictionary</i> , Fifth Edition (2002)
1018	Dyer	U.S. Patent App. Pub. No. US 2009/0236691 A1
1019	Fadell	U.S. Patent App. Pub. No. US 2004/0224638 A1

I. INTRODUCTION AND SCOPE OF WORK

1. My name is Vijay K. Madiseti. I am over the age of twenty-one (21) years, of sound mind, capable of making the statements set forth in this declaration, and competent to testify about the matters set forth below. All the facts and statements contained in this declaration are within my personal knowledge and they are, in all things, true and correct.

2. I have been retained by NXP USA, Inc. (“NXP”) to provide my opinions and views on the materials I have reviewed related to U.S. Patent No. 8,464,037 B2 (the “’037 Patent”), and the scientific and technical knowledge regarding that subject matter. I have been asked to consider what a person of ordinary skill in the art at the time of the alleged invention of the ’037 Patent (which I often refer to as a “POSA”) would have understood about the issues I address below.

3. My opinions, set forth below, are guided by my appreciation of how a POSA would have understood the state of the relevant art, the teachings of the references I discuss below, and the claims of the ’037 Patent at the time of the alleged invention (which I may also refer to as the “time of the invention” or the “alleged invention date” or just the “invention date”), which I have been asked by counsel for NXP to assume is April 30, 2008.

II. PROFESSIONAL BACKGROUND

4. My *curriculum vitae* is attached as Attachment A.

5. I have over thirty years of experience as an electrical and computer engineer in industry, education, and consulting.

6. I am a Professor in the Colleges of Computing and Engineering at Georgia Tech and now serve as a Full Professor in the School of Cybersecurity and Privacy. I have worked extensively in the field of information systems (including information security), digital communications and have studied telecommunications and systems engineering since 1981. I also have over 20 years of industry experience in computer engineering, secure information systems, distributed computer systems, networking, software engineering, signal processing, and telecommunications, including wireless communications and signal processing. Throughout this time, I have designed, implemented, and tested various products in the fields of electronics, computer engineering, and communications.

7. In 1984, I received the Bachelor of Technology in Electronics and Electrical Communications Engineering from the Indian Institute of Technology (IIT), and in 1989, a Ph.D. degree in Electrical Engineering and Computer Sciences (EECS) from University of California at Berkeley in 1989. That year, I also received the Demetri Angelakos Outstanding Graduate Student Award from the University of California, Berkeley, and the IEEE/ACM Ira M. Kay Memorial Paper Prize.

8. In 1989, I also joined the faculty of Georgia Tech. I began working at Georgia Tech as an assistant professor, became an associate professor in 1995, and have held my current position since 1998. As a member of the faculty at Georgia Tech, I have been active in, among other technologies, cloud computing, distributed computing, image and video processing, computer engineering, information security, embedded systems, chip design, software systems, wireless networks and cellular communications.

9. I have been involved in research and technology in the area of computing and digital signal processing since the late 1980s, and I am the Editor-in-Chief the IEEE Press/CRC Press's 3-volume Digital Signal Processing Handbook (Editions 1 & 2) (1998, 2010).

10. Over the past three decades, I studied, used, and designed distributed systems, including one of the first published works in distributed secure content delivery networks, that included processing of multimedia signals over the internet, called BEEHIVE, in addition to image and video processing and wireless networking circuits for several applications, including digital and video cameras, mobile phones and networking products for leading commercial firms.

11. Between 1994-1998 as part of a research initiative in collaboration with the US Army Research Laboratory and Georgia Tech, I and my students developed one of the first published implementations of a distributed signal processing

environment, where secure sensor data (from US Army) was accessed over the network and processed and rendered/displayed at distributed and protected server locations (Georgia Tech, Rice University, and Spelman College) utilizing Java-based object broker technologies. This secure environment, BEEHIVE, implemented within Java, utilized resource objects (including servers), data sources (such as cameras and sensors), sink sources (such as display), and service objects (representing Java applications), broker objects (e.g., schedulers), and user COE interfaces (e.g., GUIs) that download applications that run locally on secure data obtained from networked sources consistent with information policies and user privileges.

12. I also have significant experience in designing and implementing electronic equipment using various source code languages, including C, assembly code, VHDL, and Verilog. In 2000, I published a book entitled “VHDL: Electronics Systems Design Methodologies.” In 1997, I was awarded the VHDL International Best PhD Dissertation Advisor for my contributions in the area of rapid prototyping.

13. Since 1995, I have authored, co-authored, or edited several books in the areas of communications, signal processing, chip design, and software engineering, including VLSI Digital Signal Processors (1995), Quick-Turnaround ASIC Design in VHDL (1996), The Digital Signal Processing Handbook (1997 & 2010), Cloud

Computing: A Hands-On Approach (2013), Internet of Things: A Hands-On Approach (2014), Big Data Science & Analytics (2016).

14. I have authored over 100 articles, reports, and other publications pertaining to electrical engineering, and in the areas of computer engineering, communications signal processing, and communications. All of my publications, including the ones identified above, are set forth in my attached CV.

15. I have implemented several electronic devices, including audio and video codecs, echo cancellers, equalizers, and multimedia audio/video compression applications and modules for a leading commercial mobile phone manufacturer. I have also designed tools for porting mobile applications from one platform to another for a leading mobile phone manufacturer. I have also developed hardware and software designs for a leading set-top box manufacturer. I am familiar with the design, test and implementation of embedded systems, such as image and video processing systems, security systems, barrier control systems, and associated software and hardware architectures.

16. I have been elected a Fellow of the Institute of Electrical and Electronics Engineers (“IEEE”) in recognition of my contributions to embedded computing systems. The IEEE is a worldwide professional body consisting of more than 300,000 electrical and electronic engineers. Fellow is the highest grade of

membership of the IEEE, with only one-tenth of one percent of the IEEE membership being elected to the Fellow grade each year.

17. In 2006, I was awarded the Frederick Emmons Terman Medal from the American Society of Engineering Education (“ASEE”) and HP Corporation for my contribution to electrical engineering while under the age of 45.

18. In addition to research and commercialization projects in secure computing, in virtualization and cloud computing and advanced cloud applications and data analytics, I have also taught courses and written books in these areas. These books are widely used by dozens of universities all over the world.

19. I also have nearly thirty issued or pending applications for US Patents in the past twenty years in areas that include content management and digital rights management, encryption, and crypto-currency applications.

20. Having the above knowledge and experience, I am well qualified to offer the opinions I express in this declaration.

III. COMPENSATION

21. My work on this matter is being billed at \$650/hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services provided for in this proceeding. My compensation is not dependent on my testimony or the outcome of this or any other proceeding.

IV. LEGAL CONSIDERATIONS

22. I am not a lawyer and I offer no legal opinions. For the purposes of this declaration, I have been informed by counsel for NXP about certain aspects of the law that are relevant to my analysis, as summarized below.

23. My opinions address what would have been logical to, and within the skill level of, a POSA at the time of the alleged invention, given the state of the relevant art, the knowledge and skill that a POSA would have, the teachings of the references discussed below, and how a POSA would have understood those teachings. My opinions also address whether a POSA would have had a reasonable expectation of the modified systems discussed below successfully functioning in their modified forms as discussed below.

24. I understand that a POSA is a hypothetical person, not a genius. He generally appreciates the conventional wisdom. I understand that a POSA is also a person of ordinary creativity, not an automaton (see Section VII below, for qualifications of a POSA). I also understand that in determining who would qualify as a POSA, it is appropriate to consider criteria such as: (a) the type of problems encountered in the art; (b) prior art solutions to those problems; (c) the rapidity with which innovations are made; (d) the sophistication of the technology; and (e) the education level of active workers in the field at the time of the invention.

25. I understand that a given reference would have been relevant for a POSA to consider if it came from the same field of endeavor or was reasonably related to a problem faced by the '037 Patent inventors, and that this relevance determination underlies a principle known as “analogous art,” which should be construed broadly.

26. I have not been asked to take a position on whether a given claim would have been legally obvious to a POSA at the time of the alleged invention, but I have been told that some or all of my opinions are being used to support the argument that claims of the '037 Patent are obvious.

27. Nevertheless, I understand that a claimed invention is obvious and, therefore, not patentable if the subject matter claimed would have been obvious to a POSA at the time that the invention was made. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a modification or combination was within the skill of a POSA. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in which each element performs the same function it was known to perform, and the arrangement yields expected results,

(2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was “obvious to try” because of a design need or market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSA to modify or combine prior art references to arrive at the claimed invention.

28. I understand that a U.S. Patent document is presumed to have sufficient description to include sufficient detail for a person of ordinary skill in the art to make and use the subject matter that document describes.

29. I understand that a claim term is given the meaning that the term would have to a POSA at the time of the invention, which generally is the ordinary and customary meaning of the term. I further understand that the ordinary and customary meaning of a term may be evidenced by a variety of sources, including the words of the claims themselves, the specification, and the prosecution history, which collectively are referred to as intrinsic evidence. I also understand that prior art references cited in the patent’s prosecution history are considered intrinsic evidence. I further understand that extrinsic evidence, i.e., evidence outside the patent and its prosecution history, such as dictionaries and technical articles and other evidence

concerning relevant scientific principles, the meaning of technical terms, and the state of the art, may inform the context in which a POSA would have understood the claims of a patent. I understand this ordinary and customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term, or a claim term does not convey any particular structure to a POSA (e.g., “means for”).

V. TASK SUMMARY AND BASIS FOR OPINIONS

30. I have been asked to review the '037 Patent. I have been asked to provide my opinions from the perspective of a POSA, having knowledge of the relevant art, as of the Earliest Claimed Priority Date (“ECPD”) of the '037 Patent, i.e., April 30, 2008, and the opinions stated in this declaration are from that perspective.

31. My opinions are based on my education, training, and experience as well as items that I reviewed to prepare my opinions, including the entirety of the '037 Patent, at least the portions cited below of its prosecution history, and the entirety of the publications listed in the table at the beginning of this declaration. All of my opinions are from the perspective of a POSA, based on their general knowledge, which is reflected in the references I cite other than Smith, Bolotin, Challener, and Little, as of no later than April 30, 2008.

VI. FIELD OF TECHNOLOGY

32. The following is a description of the field of the technology of the '037 Patent from the perspective of a POSA as of the ECPD. The '037 Patent describes mechanisms for establishing a secure computer system, i.e., a computer system of enhanced integrity and security that behaves in a way that it is expected to because it is less subject to tampering, and more specifically, mechanisms for securely booting a computer system based on the use of an encryption key and pre-boot instructions both of which are stored in the computer system's secure read-only memory ("ROM"). '037 Patent, Abstract, 1:8-12.

33. Below, I use Petković in discussing aspects of the field of technology because those portions of Petković I cite reflect the general knowledge of a POSA at the ECPD. I purchased the copy I provided to counsel recently from the publisher, Springer. The following confirms to me that Petković was publicly-available prior to the TOI: (1) Petković's 2007 copyright date, which appears on the fourth, unnumbered page as "© Springer-Verlag Berlin Heidelberg 2007"; and (2) Petković's availability for purchase through Amazon by November 1, 2007 (*see* Amazon at 1-2; Archive at ¶¶ 1-7 (¶¶ 5-6: explaining URL and attachment). I reviewed Archive (a declaration from an Internet Archive employee), which I was informed pertains to a different proceeding but is being filed in this proceeding, to understand how to read the date of an archive.org URL. I understand based on the

Internet Archive declaration that the date corresponding to the URL of Amazon (<https://web.archive.org/web/20071101011130/https://www.amazon.com/Security-Privacy-Management-Data-Centric-Applications/dp/3540698604>) is November 1, 2007. I loaded this URL into a browser and confirm that Amazon is a true and correct copy of what appears at <https://web.archive.org/web/20071101011130/https://www.amazon.com/Security-Privacy-Management-Data-Centric-Applications/dp/3540698604>.

A. Secure Hardware

34. Security of a computer system is typically established in the hardware of the computer system itself. Petković, p. 120.¹ Some portion of the hardware is immutable enabling security or “trust” in the whole platform. *Id.* That trust can be leveraged either by delegating all critical functions to the trusted hardware, or by using the trusted hardware to extend a trusted platform in software. *Id.*, pp. 121-122. For the latter case, a secure non-volatile memory is used to ensure the security of the computer system’s boot process, where a non-modifiable part of the BIOS (basic input/output system)² is stored and executed before any other code. Petković, pp. 121-122. The secure memory may also be used to maintain keys or to securely

¹ All Petković cites are to Petković’s native page numbers.

² ’037 Patent, 2:38-43.

store the signatures the code components are compared against, allowing a secure update of system components. *Id.*, pp. 122, 124.

B. Secure Boot

35. Ensuring that the software or firmware code executed on a computer system is indeed the code that is expected to be executed is essential to providing a secure a computer platform. Petković, p. 123. In a system where trust must be extended outside the security hardware onto a general use platform, then boot protection, i.e., preventing unverified code configurations to be executed, is needed. *Id.*, p. 123. To achieve this objective, the system needs a trusted non-modifiable basic boot block, i.e., the first code that is executed when the platform is powered on, which is usually part of the BIOS ROM. *Id.* The BIOS verifies the next piece of code and then hands over control to that code, and thereby extends the chain of trust. *Id.* In such an implementation, every component called in the boot sequence can verify a hash or a digital signature of its successor, and abort if an error is detected. *Id.*, p. 124.

VII. LEVEL OF ORDINARY SKILL IN THE ART AT THE TIME OF THE INVENTION

36. Considering the factors discussed in paragraphs 23-24, it is my opinion that, as of the ECPD, a POSA would have had at least a bachelor's degree in electrical engineering, computer engineering, computer science, or a related degree,

and at least 2-3 years of experience in processor architecture, computer security, or a related field.

37. In making such determination, I took into account the typical coverage of an introductory course on computer systems prior to the time of the invention, which was often taught as a senior undergraduate course in electrical engineering, computer engineering, or computer science, as well as the coverage of a first-year graduate course on computer and information security and encryption prior to the time of the invention. Specifically, such an introductory course on computer security would have covered, at a minimum, paradigms for establishing a core root of trust in information systems and for extending a chain of trust to other or new system elements, as well as techniques for authenticating, encrypting, and decrypting data. A person with a BS degree in EE/CE/CS who did not otherwise learn about them during their undergraduate studies would have learned about them through the work experience described above.

38. I am familiar with the knowledge and capabilities that a POSA of computer security would have possessed at the ECPD. My experience in relation to both academia and industry up to and including the ECPD allowed me to become personally familiar with the knowledge and capabilities of a POSA in the area of computer system security. By the ECPD, my expertise met or exceeded that of a

POSA. Thus, I am capable of rendering an informed opinion on the level of ordinary skill in the art as of April 30, 2008.

VIII. OVERVIEW OF THE '037 PATENT

A. Overview

39. The '037 Patent is entitled “Computer System Comprising a Secure Boot Mechanism on the Basis of Symmetric Key Encryption.” The '037 Patent's Background section includes a Description of the Related Art that highlights those increasing capabilities and proliferation of computer systems and networks has manifested “the possibility of distributing malicious software applications,” which may be used to tamper with a computer platform without the platform user's knowledge for purposes of sniffing or spoofing or in order to start a denial of service attack. '037 Patent at 1:26-46. In order to minimize these issues and enhance computer platform integrity, the Background highlights that symmetric or asymmetric encryption/decryption techniques are available. *Id.* at 1:46-51. The Background explains that “in asymmetric encryption techniques, a pair of a private key and a public key may be used for encrypting data by means of one of the keys and deciphering the data by using the other key.” *Id.* at 1:51-55. In symmetric encryption, a unique key is used for both the encryption and decryption of data, whereas access to the unique key may have to be restricted to authorized parties. *Id.* at 1:55-58. The Background then notes that while such techniques may enhance the

integrity of data communication and data storage, “the actual configuration of a computer platform” may be susceptible to attack, “as many applications intended for increasing computer security may rely on a trusted platform configuration.” *Id.* at 1:58-64.

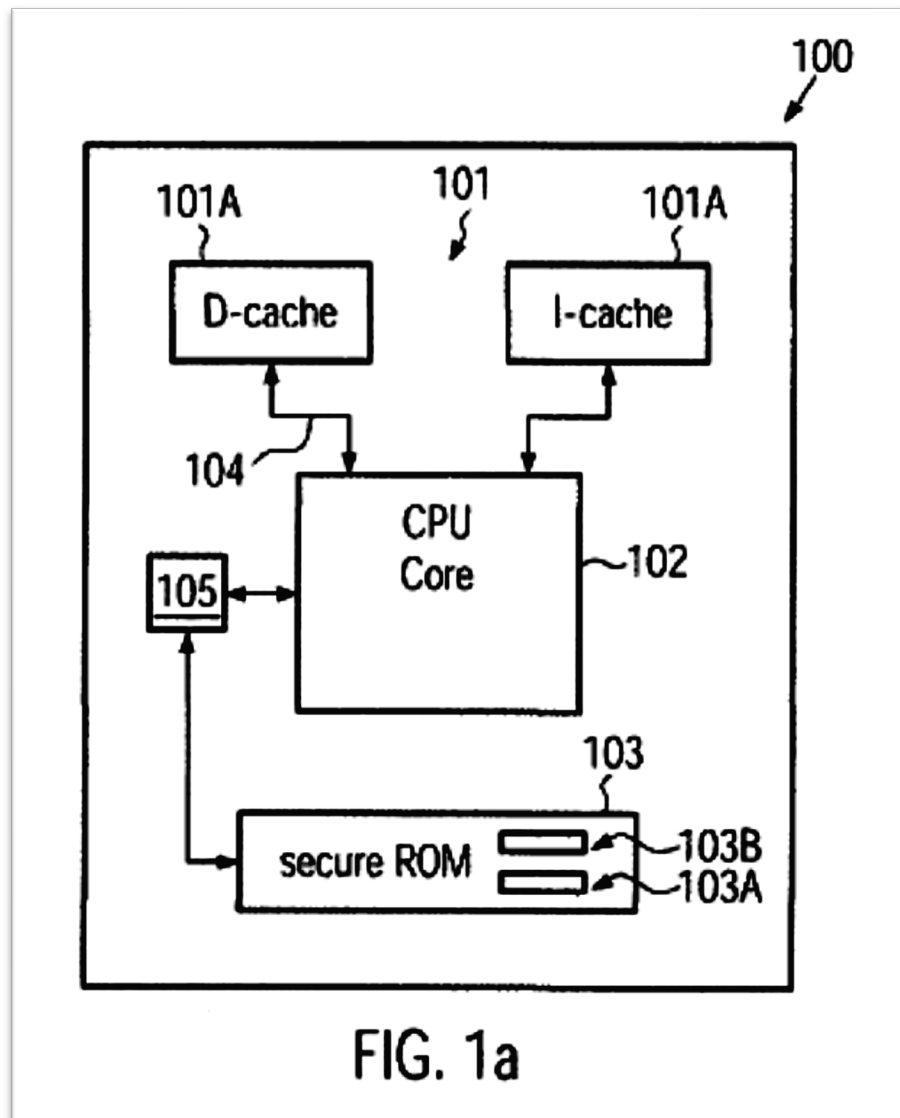
40. The Background then describes a trusted computer platform as “a computer system in which the hardware configuration as well as the software applications may be assumed to be in a dedicated configuration.” *Id.* at 1:65-2:1. The Background then explains that a secure compute platform may only be established via “a ‘chain’ of trusted platform levels, wherein each subsequent level may be initialized by the preceding level only if the integrity of the subsequent level has been verified.” *Id.* at 2:1-5. The root of the chain of trust “may have to be provided with a high degree of tamper resistance so as to also accomplish a high confidence with respect to integrity of the system.” *Id.* at 2:5-11.

41. The Background then argues that the chain of trust approach should be used for the initialization and booting or “boot strapping” process of a computer platform because “it may not be sufficient to assume that the various system activities prior to loading the operating system may be considered to be secure so as to rely on the integrity of the operating system and the user application, since the operating system may be invoked by a ‘mistrusted’ process.” *Id.* at 2:12-33.

42. The Background then describes a “typical” sequence for initializing a computer system “after applying power or after a reset event causes a ‘power-on self-test’ of the CPU. *Id.* at 2:34-38. First, the processor starts executing instructions at a dedicated entry address indicated by the processor’s reset vector. *Id.* The reset vector “may typically be an address of a software routine, frequently referred to as BIOS (basic input/output system), which may be stored in a dedicated external non-volatile memory.” *Id.* at 2:38-43. Then, under BIOS control, the remaining self-test may be performed and the hardware configuration of the platform may be determined. *Id.* at 2:43-46. Thereafter, “a search for a bootable device may be performed ... from which a primary boot block may be loaded into the system memory.” *Id.* at 2:46-52. Control is then passed to the primary boot block, which may load an operating system into the system memory. *Id.*

43. The Background then states that providing a reliable core root of trust for a boot process is difficult “in view of platform modifications, such as replacement of the non-volatile memory chip including the BIOS software,” and suggests that undesirable tradeoffs between security and flexibility in updating BIOS software are frequently required. *Id.* at 2:53-64. The ’037 Patent, the Background states, is directed to reducing the effects of such issues. *Id.* at 2:65-67.

44. In the Detailed Description, the '037 Patent introduces FIG. 1a (shown below) illustrating a CPU 100 for implementing a computer platform “with a high degree of tamper resistance.” *Id.* at 7:7-10.



45. According to the '037 Patent, the CPU 100 may comprise a CPU core 102, which maybe functionally connected to a random-access memory (RAM) 101. *Id.* at 7:10-17. The RAM 101 may comprise a data cache 101A and an instruction

cache 101B. *Id.* at 7:17-20. The '037 Patent notes that “any appropriate memory technology may be used as long as direct control of the memory 101 via the CPU core 102 may be accomplished without the possibility of external access during the initialization of the CPU.” *Id.* at 7:20-27.

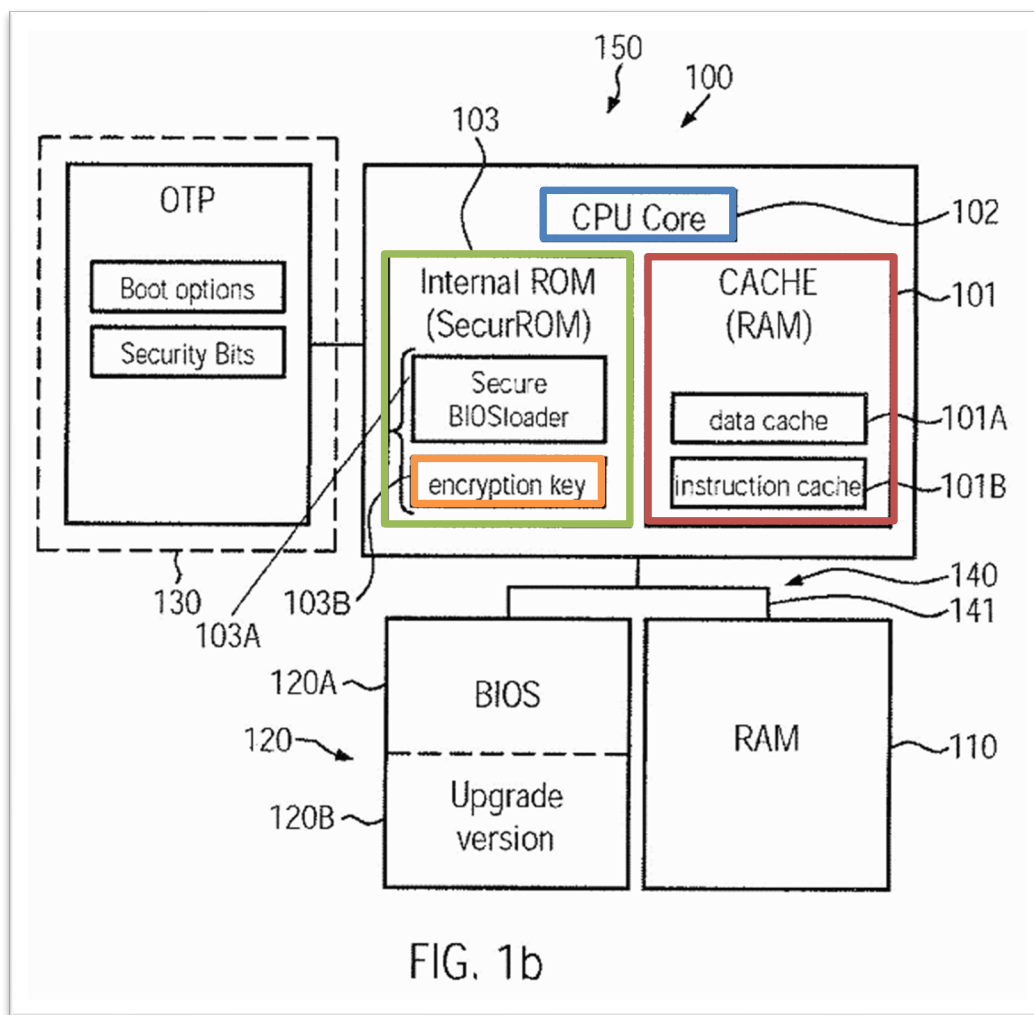
46. The CPU may further comprise a non-volatile memory 103, “which may not enable external access for modifying the contents of the memory 103,” establishing the memory 103 as a secure memory area, and which may not be overwritten with new data once programmed. *Id.* at 7:27-41. Securely stored in the memory 103 are a set of data values 103A in the form of instructions executable by the CPU core and data values, such as instruction operands. *Id.* at 7:35-41. The memory 103 may also contain a symmetric encryption key 103B, which may be used for signing a portion of boot information. *Id.* at 7:41-45.

47. The CPU 100 of FIG. 1a also includes a lock mechanism 105, which may be configured to enable CPU access to the memory or at least the encryption key 103B “upon power-on or a reset event only,” while disabling the memory 103, by instructing the lock mechanism 105 during the boot process after verification of at least a portion of the boot information. *Id.* at 7:53-66. The '037 Patent states lock mechanism 105 “may comprise circuitry for shutting down supply energy for the memory 103 and also for the mechanism 105 itself, thereby locking the mechanism 105 and also the memory 103 non-responsive to any accesses performed after a

request for ‘locking’ the memory 103 or at least a portion thereof including the symmetric encryption key 103B.” *Id.* at 7:60-66.

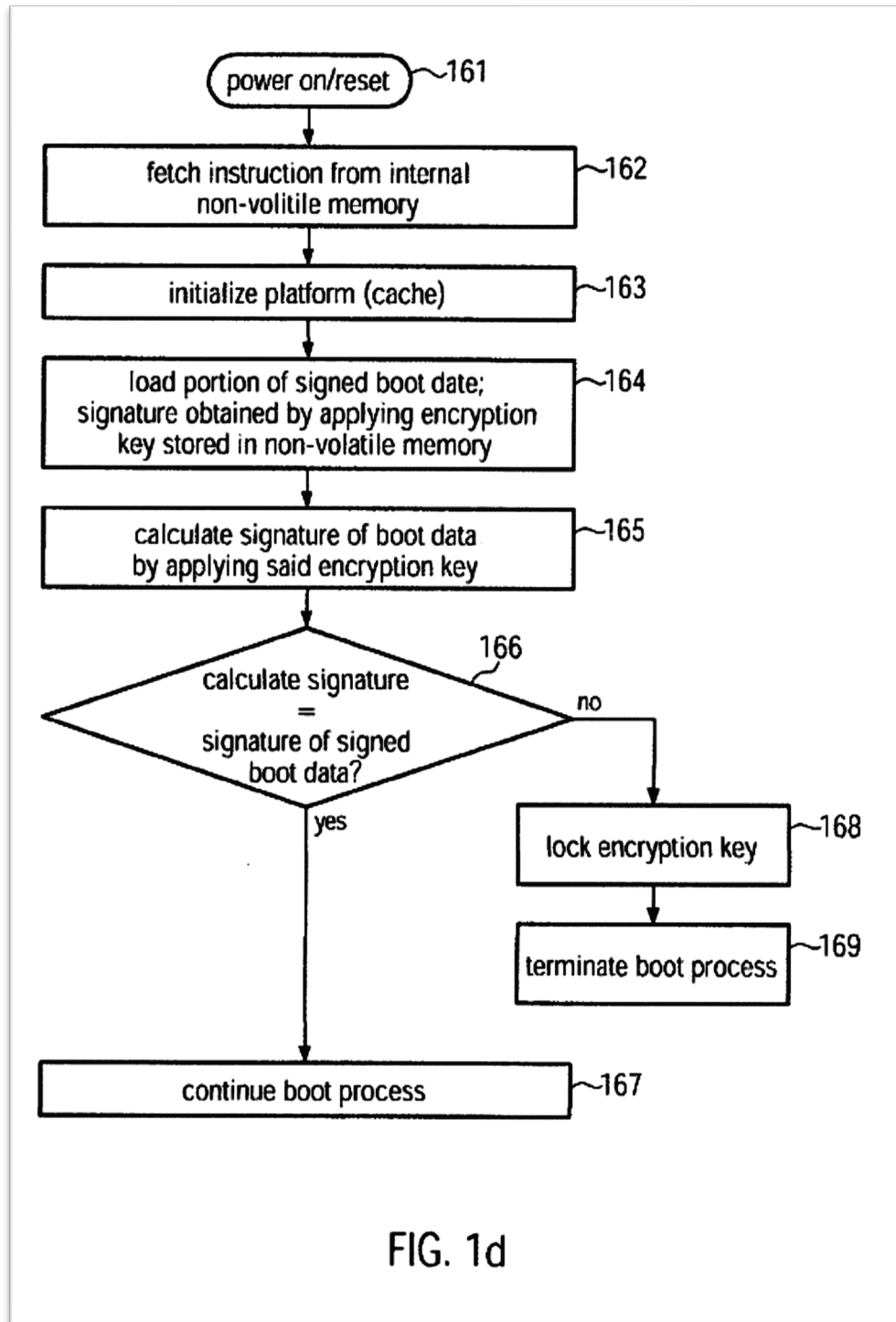
48. The CPU’s internal components are interconnected by a bus system 104, such that upon power-on or reset, a jump to a specified address of memory 103 occurs, ensuring that execution of instructions starts from the memory 103. *Id.* at 8:9-17. The ’037 Patent then describes that the CPU components may be formed as a single integrated circuit on a common substrate using well-established manufacturing techniques. *Id.* at 8:18-40.

49. FIG. 1b (annotated below) illustrates a computer system 150 using the CPU 100 described in FIG. 1a and providing the secure computer system of the invention. *Id.* at 8:41-62. The computer system 150’s CPU 100 contains a CPU core 102 (outlined in blue), a secure (lockable) ROM 103 (outlined in green) and RAM in the form of a cache 101 (outlined in red). *Id.* at 8:41-49.



50. The ROM 103 contains a data set 103A in the form of a secure BIOS loader containing pre-boot instructions and data values, and a symmetric encryption key 103B (outlined in **orange**). *Id.* at 7:35-41. The symmetric encryption key is used to sign and verify a portion of boot information before it is used by the CPU. *Id.* at 7:41-45. Computer system 150 also includes a system memory 110, such as a RAM, and a non-volatile memory 120, such as a flash memory. *Id.* at 8:66-9:9.

51. FIG. 1d (shown below) illustrates a process for verifying a portion of a boot routine based on verification of the boot routine's signature. *Id.* at 10:38-41.



52. The '037 Patent describes a step 161 of a power-on or reset to initiate secure boot processing, including a self-test by the CPU core. *Id.* at 11:26-31. In step 162, the first instruction is retrieved from the internal non-volatile memory 103, such that the CPU 100 may execute based on the data set 103A. *Id.* at 11:34-40. In step 163, the internal RAM 101 may be initialized based on instructions contained in the data set 103A. *Id.* at 11:40-42. In step 164, a portion of the boot information, e.g., portion 121A including the signature 121S, is loaded into the internal RAM 101. *Id.* at 11:42-46. In step 165, a signature of the boot data is generated by, for example, calculating a respective hash value and applying the encryption key 103B. *Id.* at 11:48-52. Then in step 166, the currently calculated signature is compared to the signature 121S. *Id.* at 11:53-57. If the values match, the process advances to step 167 to continue the boot process. *Id.* If the values do not match, then in step 168, the encryption key may be locked via the lock mechanism 105 and in step 169, the boot process may be terminated. *Id.* at 11:57-64.

B. Challenged Claims

53. I understand that Claims 1-3, 5-6, and 19-21 are challenged claims. Claims 1, 19 and 21 are independent. Claims 2-3 and 5-6 depend from independent Claim 1. Claim 20 depends from Claim 1. Below is a listing of the claims that I use when referring to certain limitations:

Claim	Limitation No.	Limitation
1	1[pre]	A method for booting a computer system, the method comprising:
	1[a]	upon one of a power-on and a reset event, executing instructions stored in an internal non-volatile memory area of a central processing unit,
	1[b]	said internal non-volatile memory area containing an encryption key;
	1[c]	determining a current signature of at least a signed portion of boot data,
	1[d]	said signed portion of boot data containing a stored signature of said portion of boot data,
	1[e]	said stored signature created by applying said encryption key; and
	1[f]	executing boot instructions contained in said signed portion of boot data, when said current signature and said stored signature are identical.
2	2[pre]	The method of claim 1, wherein executing said instructions stored in said internal non-volatile memory area comprises
	2[a]	initializing an internal random access memory of said central processing unit and loading said at least a portion of boot data from a non-volatile memory into said initialized random access memory.
3	3[pre]	The method of claim 2, wherein executing boot instructions of said signed portion of boot data comprises
	3[a]	initializing a system memory of said computer system and loading said boot data from said non-volatile memory into said system memory and verifying an integrity of said boot data.
5	5	The method of claim 1, further comprising locking said encryption key for external access prior to executing said boot instructions.
6	6	The method of claim 5, further comprising determining whether an upgrade version of said boot data is available.
19	19[pre]	A central processing unit (CPU), comprising:

Claim	Limitation No.	Limitation
	19[a]	a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory; and
	19[b]	pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key.
20	20	The central processing unit of claim 19, further comprising a locking mechanism configured to enable a portion of said non-volatile memory containing said symmetric encryption key upon one of a reset event and a power-on event only.
21	21[pre]	A computer system, comprising:
	21[a]	a central processing unit comprising:
	21[b][i]	a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory, and
	21[b][ii]	pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key; and
	21[c]	a non-volatile boot memory configured to hold at least said boot routine and a second boot routine.

C. Prosecution History

54. The examiner first rejected claims 1-4, 19 and 21 as anticipated by Little. '037 PH at 138-142. Little discloses a method of booting a computer system using a public key stored in internal ROM. Little at Abstract, ¶¶ 30-35. The examiner also rejected claims 5 and 20 as obvious over Little in view of Chen. '037 PH at 142-143. Chen teaches a similar booting method in which the encryption key is locked for external access before executing boot instructions. *Id.*

55. The Applicants responded by arguing that Little is silent with respect to the stored signature being created by applying the encryption key, and that Little only describes using public keys for authenticating a digital signature and does not describe using encryption keys for creating a digital signature. '037 PH at 172-173.

56. The examiner then rejected claims 1-4, 19 and 21 as anticipated by Bhatt. *Id.* at 182-184. Bhatt discloses a method of booting a computer system using a verification code stored in boot storage to generate a signature used to compare to a signature value stored in ID storage. *Id.* at 182-183. Boot code is executed if the computed signature and the signature value match. *Id.* at 182-183.

57. The Applicants again responded without amendments and argued that while Bhatt discloses that a signature is computed using a hashing or similar algorithm using a secret key, Bhatt does not disclose that the key is stored in internal

memory, and that it is that secret key that corresponds to the encryption key of the claims. *Id.* at 205-206.

58. In the next office action, the examiner rejected claims 1-5 and 19-21 as being anticipated by Bhatt in view of Kumar. *Id.* at 304-306. Kumar discloses a method involving secure booting, including use of an internal non-volatile memory area containing an encryption key. *Id.* at 305.

59. Once again, the Applicants responded without amendment and argued that while Kumar describes storing a cryptographic key called a PUK in a non-volatile form, Kumar teaches that the key is used to encrypt and protect another key, and therefore the PUK is not an encryption key used to determine a current signature of a portion of boot data. *Id.* at 326-327. Applicants also argued that Bhatt teaches away from the claims and from storing previously calculated signatures with portions of boot code in external memory. *Id.* at 328. The examiner then allowed the claims without further remarks. *Id.* at 333-337.

IX. COMPARISON OF THE PRIOR ART TO THE CHALLENGED CLAIMS

60. I understand that NXP is asserting that Claims 1-3, 5-6, and 19-21 are invalid as obvious over the following prior art and combinations of prior art:

Ground	Claims	Prior Art
1	1-3, 5-6, 19-21	Smith
2	20	Smith, Bolotin
3	19-21	Smith, Little
4	20	Smith, Bolotin, Little
5	5-6	Smith, Challenger

A. Smith

61. Smith discloses a method and apparatus for secure booting of a device using codes stored in a secure ROM of the device and code images that are verified before they are executed. Smith at Title, Abstract. Smith’s FIGs. 1 and 2 depict the hardware configuration of the system (100) used for the secure booting. Smith at ¶¶ 12, 13, 30, 33, 38, FIGs. 1, 2.

1. Smith versus Claims 19-21, 1-3 and 5-6

- a) **Limitations 19[pre] and 19[a]: “A central processing unit (CPU), comprising: a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory; and”**

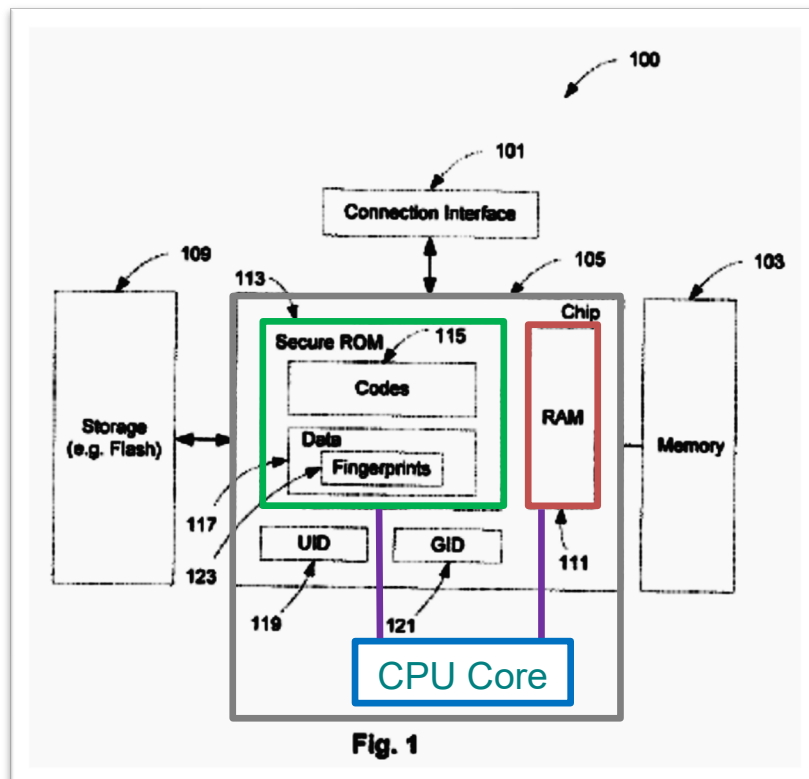
62. As explained below, Smith’s FIG. 1 chip 105 includes “a volatile random access memory” (**red**) and “a non-volatile memory” (**green**). And a POSA would have considered it logical and straightforward for that chip to further include:

- (1) a CPU core (**blue**) (discussed below in Section IX.A.1.a)(3)) so that it could perform Smith’s secure-booting processes as Smith describes and

as the chip's EDRAM memory suggests, thereby rendering the chip "a central processing unit (CPU);" and

- (2) a bus system (purple) (discussed below in Section IX.A.1.a)(4)) to allow the necessary communication between the CPU core and the memories.

Finally, Smith's "chip" 105 includes (or a POSA would have found it logical and straightforward for it to have included) a single substrate (grey) on which the above-referenced components were formed (discussed below in Section IX.A.1.a)(5)). These structures are shown on an annotated version of Smith's FIG. 1, reproduced below.



(1) EDRAM 111 of chip 105 is a volatile random access memory

63. Smith discloses that chip 105 includes a “RAM (Random Access Memory) component 111” and provides “EDRAM (Embedded Dynamic Random Access Memory)” as a specific example. Smith at ¶ 30, FIG. 1. As a POSA would have known, being a type of dynamic RAM, such EDRAM “requires power continually in order to refresh or maintain the data in the memory” (*id.* at ¶ 64) and is therefore volatile. Microsoft³ at 558 (defining “**volatile memory**” as “[m]emory, such as RAM, that loses its data when the power is shut off”).

(2) ROM 113 of chip 105 is a non-volatile memory

64. Smith’s chip 105 further includes “a ROM 113” (Smith at ¶ 30), which is a “Read Only Memory” (*id.* at ¶ 28). As a POSA would have understood, ROM is a non-volatile memory. Microsoft at 367 (defining “**nonvolatile memory**” as “[a] storage system that does not lose data when power is removed from it” and is “[i]ntended to refer to ... ROM”), 333 (“memory storage can be ... *nonvolatile* and hold data permanently, such as ROM” (emphasis added)).

³ I use Microsoft for certain dictionary definitions that reflect what a POSA would have understood about the meaning of certain claim terms. I obtained the copy I used, which I understand is being filed, online.

(3) A POSA would have considered it logical and straightforward for chip 105 to have included a CPU core

65. Smith's secure-booting processes involve the execution of codes, specifically, "executing codes embedded inside a device to verify a code image loaded in a memory of the device" before the code image is executed. Smith at Abstract; *see also*, e.g., ¶¶ 7-9, FIG. 3 ("[e]xecut[ing] instructions in a ROM chip" to "[v]erify [a] code image," and, if so, "[e]xecut[ing] the code image to perform booting operations"), FIG. 4 and ¶ 42 (executing a "Secure Hash Algorithm" and an "Advanced Encryption Standard[] algorithm" as part of verifying a code image), FIG. 6 ("[r]eceiving an instruction ... to execute the code image"), FIG. 8 (same as well as "executing the trusted kernelcache" and "[r]eceiving and executing command calls"), FIG. 9 ("[e]xecuting the verified code image"), FIG. 10 (similar). Executing such codes, as a POSA would have understood, was performed using a CPU core. Smith explains as much, disclosing that "a microcontroller or other form of *processing core*"—i.e., a CPU core—is what "*executes program code instructions.*" *Id.* at ¶ 71 (emphasis added).

66. A POSA would have considered it logical and straightforward to have included a CPU core like Smith describes in Smith's chip 105. First, Smith explains that his secure-booting processes are performable with his FIG. 1 system 100. *See* Smith at ¶ 30 ("FIG. 1 ... illustrat[es] ... system components *for secure booting*"

(emphasis added)). With respect to Smith's FIGs. 3 and 4 processes, Smith makes this explicit: those processes "may be performed by system 100 of FIG. 1." *Id.* at ¶ 38 (FIG. 3); *see also id.* at ¶ 42 (FIG. 4). With the understanding that these secure-booting processes were performed using a CPU core (as discussed above), Smith's explanation that system 100 is capable of performing them would have both suggested to a POSA the inclusion of a CPU core in that system and motivated them to accomplish such inclusion to enable its Smith-described capability. In the single-chip version of Smith's system 100, this logical-and-straightforward-to-include CPU core would have been on Smith's chip 105. Smith at ¶ 30 ("System 100 may reside in *one* or more chips inside a device." (emphasis added)).

67. Second, as a POSA would have understood, Smith's on-chip EDRAM 111 (*see* Section IX.A.1.a)(1)) is "a capacitor-based dynamic random access memory usually *integrated on the same die or in the same package as the main ASIC or processor.*" Dyer at ¶ 28 (emphasis added). Such presence of EDRAM 111 on chip 105 would have therefore at least suggested to a POSA the further presence of a CPU core on the chip with which the EDRAM was integrated, particularly in view of their understanding that the secure-booting processes performable with Smith's FIG. 1 system 100 including the chip involved the use of a CPU core (as discussed above).

(4) A POSA would have considered it logical and straightforward for chip 105 to have included a bus system

68. A POSA would have understood that the CPU core of Smith's chip 105 (*see* Section IX.A.1.a)(3)) needed to communicate with the volatile memory (*see* Section IX.A.1.a)(1)) and non-volatile memory (*see* Section IX.A.1.a)(2)) of the chip in order to perform Smith's secure-boot processes. Smith's FIG. 3 secure-boot process, for example, is explicitly performable by Smith's FIG. 1 system 100 and includes "[e]xecut[ing] instructions in a ROM chip to locate a code image for booting a device," "[l]oad[ing] the code image into a memory," and "[e]xecut[ing] the code image" if it is verified. Smith at FIG. 3; *see also id.* at ¶¶ 38-40. Those code-image-locating instructions "may be read from a code section of the ROM chip as in codes 115 of FIG. 1" (*id.* at ¶ 38)—i.e., may reside in Smith's ROM 113 (*id.* at ¶ 30, FIG. 1). And the code image may be loaded into Smith's RAM 111 for post-verification execution. *Id.* at ¶¶ 38, 40 ("the [executed] code image may be LLB, an iBoot or a Kernelcache as shown in 225, 227 and 223), FIG. 2 (showing LLB and iBoot in Smith's RAM 111), ¶ 31 ("codes 115 may cause loading a code image into a device memory such as ... RAM 111").

69. A POSA also knew how to achieve such communication between the CPU core, RAM 111, and ROM 113 of Smith's chip 105: via a bus. Microsoft at 77-78 (defining "**bus**" as "[a] set of hardware lines (conductors) used for data

transfer among the components of a computer system”). As Smith explains with respect to his FIG. 12 data processing system 1200 that “may be implemented as part of system [100] as shown in FIG. 1” and includes, akin to Smith’s chip 105, a “memory 1201 for storing ... programs for execution by the processing system” and “a processing system 1211, which may be one or more microprocessors” (Smith at ¶ 66), “[i]t will be appreciated that one or more buses ... may be used to interconnect the various components as is well known in the art” (*id.* at ¶ 68). *See also id.* at ¶¶ 64, 65, 72, FIG. 11 (describing and depicting buses as “well-known” structures for interconnecting components).

70. To a POSA, it would have been logical and straightforward to have included a bus like Smith describes on Smith’s chip 105 because doing so would have provided for the above-described communication between the CPU core, RAM 111, and ROM 113 and thereby enabled chip 105 to perform the above-described secure-boot processes. Specifically, and as reflected in **purple** in the annotated version of Smith’s FIG. 1, reproduced above, that bus would have included a portion connecting the CPU core to ROM 113 so that the CPU core could read and execute the code-image-locating instructions contained therein, as well as a portion connecting the CPU core to RAM 111 so that the CPU core could read and execute the code image loaded therein. Such a bus satisfies the claimed “bus system.”

(5) “Chip” 105’s components would have been on a substrate

71. As a POSA would have understood, the volatile and non-volatile memories, CPU core, and bus system of Smith’s “chip” 105 would have been “fabricated on a single chip of ... semiconductor material.” Microsoft, 98, 277 (a “**chip**” is an “integrated circuit,” which is “[a] device consisting of a number of connected circuit elements ... fabricated on a single chip of ... semiconductor material”). Such chip of semiconductor material corresponds to the claimed substrate.

b) Limitation 19[b]: “pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key.”

72. Smith discloses this limitation.

(1) Certain codes 115

73. Several of Smith’s “codes 115” qualify as the claimed “pre-boot information.”

74. For example, Smith describes “codes 115” that “may cause hardware initialization for the device to support a connection or communication interface such as USB (Universal Serial Bus) ... [and] include instructions to change the clock rate of the device” (hereinafter, “First Codes 115”). Smith at ¶ 30. First Codes 115 qualify as the claimed “pre-boot information” because, as a POSA would have

understood based on Smith, their execution: (1) would have caused one or more activities (the referenced hardware initialization and the referenced clock rate change) that preceded the code image execution that Smith describes as part of process 300 for secure booting (*see id.* at ¶¶ 38-41) and that Smith describes and illustrates as being for the purpose of performing “booting operations” (*see id.* at ¶ 40, FIG. 3 (block 311: “Execute the code image to perform booting operations”), and/or (2) would not have been changed or interrupted, which the ’037 Patent indicates qualifies them as “pre-boot” information (*see* ’037 Patent at 3:21-25).

75. Smith also describes “codes 115” as including instructions that, when executed, locate a code image stored in a storage component of the device in which chip 105 was positioned (hereinafter, “Second Codes 115”). Smith at ¶ 38. Second Codes 115 also qualify as the claimed “pre-boot information” because, as a POSA would have understood based on Smith, their execution: (1) would have caused the referenced code image location (*see* Smith FIG. 3 at blocks 301 and 319) before booting operations started (*see* Smith FIG. 3 at block 311 and Smith ¶ 40 (describing “booting operations” starting with the execution of the code image)), and/or (2) would not have been changed or interrupted, which the ’037 Patent indicates qualifies them as “pre-boot” information (*see* ’037 Patent at 3:21-25).

76. Second, Smith discloses that ROM 113 (i.e., the claimed “non-volatile memory,” as explained in Section IX.A.1.a)(2)) “stores the codes 115.” Smith at

¶¶ 30, 38 (explaining the process therein described (process 300) is performed with system 100 and that the instructions of the Second Codes 115 are “in a ROM chip,” which a POSA would have understood to refer to ROM 113 in which codes 115 are depicted as stored in FIG. 1). Accordingly, First Codes 115 and Second Codes 115 qualify as the claimed “pre-boot information stored in non-volatile memory.”

77. Next, as a POSA would have understood, the hardware initialization and/or the clock rate change caused by First Codes 115 would necessarily have been caused by instructions corresponding to those codes being executed by the CPU core of chip 105. *See* Smith at ¶ 30 (“Codes 115 may include *instructions* to change the clock rate of the device.” (emphasis added)). And, as a POSA would have understood, the code image location caused by Second Codes 115 also would have necessarily been caused by instructions corresponding to those codes being executed by the CPU core of chip 105. *See id.* at ¶ 38 (“the processing logic of process 300 may locate a code image by executing *instructions* in a ROM chip at block 301 ... read from a code section of the ROM chip as in codes 115 of FIG. 1” (emphasis added)). Therefore, both First Codes 115 and Second Codes 115 include “instructions executable by [the] CPU core,” as claimed.

(2) UID 119

78. Smith also discloses “pre-boot information including ... a symmetric encryption key.” Smith discloses that UID 119 for a device that system 100 is

securely booting is a symmetric encryption key for that device. *See* Smith at FIG. 2, ¶¶ 28, 32 (UID 119 is unique for the device).

79. First, UID 119 qualifies as “pre-boot information” because, as a POSA would have understood based on Smith, its use: (1) would have been considered by a POSA as “pre-boot” (and, thus, UID 119 as “pre-boot information”) because such use preceded the code image execution (block 311) of process 300 that Smith describes in FIG. 3 as performing “booting operations” (*see* Smith at ¶ 40, FIG. 3 (block 311 text), and/or (2) would not have been changed or interrupted, which the ’037 Patent indicates qualifies it as “pre-boot” information (*see* ’037 Patent at 3:21-25).

80. Second, UID 119 is a symmetric encryption key. Smith discloses that the UID may be used to create a signature value over a code image “through well-known hashing and encryption algorithms.” Smith at ¶ 39. The same UID is used to generate a current signature—described as deriving another signature—for the code image using the same well-known hashing and encryption algorithms. *Id.* at ¶ 39 (describing block 305), FIG. 3 (depicting block 305). Elsewhere, Smith describes that current-signature generation with reference to process 400 shown in FIG. 4 as “perform[ing] an encrypting operation at block 403 to generate a signature 405 based on a key string 413, UID 401 and seed string 407 associated with a device.” *Id.* at ¶ 42, FIG. 4. A POSA would have understood UID 119 corresponds

to UID 401 because Smith states, “UID 401 may be stored within the device as UID 119 shown in FIG. 1.” *Id.* at ¶ 42, FIG. 4. Smith also states the encrypting operation can be “based on AES (Advanced Encryption Standard) algorithm.” *Id.* at ¶ 42, FIG. 4. Because AES is an algorithm for symmetric key encryption, a POSA would have understood the UID to be a symmetric encryption key. Madisetti, ¶ 107.⁴ Indeed, Smith confirms, “the verification of the code image is based on a key stored within the secure storage area. *The key may be unique to each device.*” Smith at ¶ 7 (emphasis added); *see also id.* at ¶ 8 (“Upon being successfully verified using a key stored within the secure ROM and *the key uniquely identifying the device ...*.” (emphasis added)), claim 5 (“wherein the key [stored within the ROM] is configured to uniquely identify the device”), claim 13 (same), claim 19 (describing a computer implemented method involving “a key stored within the secure ROM [that] uniquely identif[ies] the portable device”), claim 22 (describing machine readable media that, when executed by a machine, perform a method that involves “a key stored within the secure ROM [that] uniquely identif[ies] the portable device”), claims 25 and 28

⁴ Hereinafter, such current-signature generation described in this paragraph 111 with reference to Smith paragraphs 39 and 42 and FIGs. 3-4 shall be referred to as Smith’s “Current Signature Generation.”

(similar to claims 19 and 22, respectively, except referencing a “unique identifier (ID)” rather than “key”).

81. As confirmation, the '037 Patent itself describes that the symmetric encryption key may be in the form of an identifier that is unique to a computer system. The '037 Patent states, “[i]n symmetric key encryption techniques, a unique key is used for both the encryption and decryption of data, wherein access to the unique key may have to restricted to authorized parties.” '037 Patent at 1:55-58. The '037 Patent also states that the unique symmetric encryption key may be unique to each computer system, like the UID in Smith, so as to not undermine the security of a computer system when another computer system’s symmetric key has been compromised: “the desired degree of security may be determined by restricting the usage of the unique encryption key to a specified type of central processing unit so that manipulations or loss of secrecy for one unique encryption key may not affect other CPUs that have been fabricated on the basis of a different unique encryption key.” '037 Patent at 6:61-66. Going further, the '037 Patent explains that the symmetric encryption key may be determined and—like Smith’s UID (Smith at ¶ 28)—incorporated into the CPU during manufacturing: “the degree of security may be determined on the basis of the granularity of the symmetric encryption key, which may be incorporated into the non-volatile memory during fabrication of the CPU device.” '037 Patent at 14:26-28.

82. Accordingly, Smith discloses pre-boot information including a symmetric encryption key.

83. Third, Smith discloses that UID 119 is stored in secure ROM 113. As Smith states:

[T]he security of booting a device may be performed by the code and data stored inside a secure storage area such as a ROM (Read Only Memory), also referred to as a secure ROM, integrated together within the device. The content of a secure ROM may be stored during a manufacturing stage of the device. The secure ROM may be associated with a UID (Unique Identifier) of the device.

Smith at ¶ 28. Here, Smith describes a single secure ROM that contains the code and data for performing the secure booting processes of the device. *Id.* This passage then references the “content” of the secure ROM, and then specifically identifies the UID as an instance of that content in the secure ROM. *Id.* In system 100 depicted in FIGs. 1 and 2—the only system Smith discloses for performing Smith’s secure booting processes—secure ROM 113 is the only secure ROM that is depicted. *Id.* at FIGs. 1, 2. Further, in the context of the process described in FIG. 6, which Smith explains is performable with system 100 from FIG. 1, Smith recites regarding the UID’s secure storage location: “the processing logic of process 600 may derive a hash signature from the code image based on a UID stored in a secure ROM of the device, such as UID 119 as shown in FIG. 1.” *Id.* at ¶ 47. Similarly, Smith recites,

“[t]he device may perform a chain of certifications to verify the received code image is trusted based on a UID and a fingerprint stored inside a secure ROM of the device such as UID 119 and fingerprints 123 of FIG. 1.” *Id.* at ¶ 47. Smith paragraphs 49, 51, 53, and 61 contain additional, similar disclosures. In the “Summary of the Description” section, Smith states the “unique identifier (ID) uniquely identifying the device”—which a POSA would understand to be the UID Smith references elsewhere—can be stored in the secure ROM.” *Id.* at ¶ 9. Smith’s claims 5, 19, 22, 25, and 28 provide similar support, as explained above in paragraph 80. In view of these Smith disclosures, and because the only secure ROM depicted in FIGs. 1 and 2 or otherwise referred to in Smith is secure ROM 113, a POSA would have understood that the secure ROM storing UID 119 is secure ROM 113.

84. Smith also discloses that UID 119 is “for verifying at least a portion of a boot routine signed by using said symmetric encryption key.” First, Smith expressly discloses a process that “may verify whether the loaded code image could be trusted based on a UID associated with the device such as UID 119 of FIG. 1.” Smith at ¶ 39, FIGs. 1 and 3. Smith also discloses that the loaded code is a boot routine, namely the low level boot (“LLB”) code image 229 depicted in FIG. 2. *Id.* at FIGs. 2 and 3, ¶ 40. LLB 229 is a boot routine that is executed after it is loaded into RAM 111 as LLB 225. *Id.* at FIGs. 2 and 3, ¶¶ 40, 33-34. Thus, Smith discloses using UID 119 to verify (at least a portion of) a boot routine.

85. Second, Smith discloses the boot routine to be verified is signed using UID 119. Specifically, Smith discloses that the loaded code image being verified contains a header value that “may include a signature value signed over the code image according to the UID of the device through well-known hashing and encryption algorithms.” Smith at ¶ 39. To the extent the Patent Owner (“PO”) argues more than the foregoing is required, Smith’s Current Signature Generation, which applies to boot routine LLB 225 (*id.* at ¶ 40, FIG. 2), also meets this aspect of Limitation 19[b].

86. Smith therefore discloses a symmetric encryption key in the form of UID 119 for verifying at least a portion of a boot routine signed by using the symmetric encryption key.

c) Claim 20: “The central processing unit of claim 19, further comprising a locking mechanism configured to enable a portion of said non-volatile memory containing said symmetric encryption key upon one of a reset event and a power-on event only.”

87. Smith discloses processing logic in the form of a trusted code image that disables access to or locks UID 119 (i.e., the claimed symmetric encryption key) in secure ROM 113. Smith at ¶ 61. Specifically, Smith explains that “the processing logic of process 1000[, performable by system 100 for executing unverified code image,] may disable accessing to a UID of a secure ROM in a device such as UID 119 in FIG. 1. In one embodiment, a trusted code image may be configured to turn

off accessing to the UID when executed.” *Id.* at ¶ 61, FIGs. 1, 2, 10. Smith also recites that the “trusted code image may be a boot image verified by codes inside a secure ROM ... such as codes 115 in FIG. 1.” *Id.* at ¶ 61. Smith also discloses a hardware switch configured to disable access to UID 119 of secure ROM 113. *Id.* at ¶ 61. Accordingly, Smith discloses that, in a process performed by system 100 of FIGs. 1 and 2, UID 119 in secure ROM 113 may be locked from access.

88. Smith also discloses to a POSA that ROM 113 is accessible or enabled before UID 119 is used following a reset or reboot of the system. For example, Smith discusses accessing the UID when codes 115 in ROM 113 are verifying a first code image, like LLB 225, using the UID. Smith at ¶ 32. As a POSA would have understood, it would not have been possible to access UID 119 in ROM 113 if at least the portion of ROM 113 where UID 119 was stored had not earlier been enabled.

89. To a POSA, these Smith disclosures describe a lock mechanism for disabling or enabling access to at least the portion of ROM 113 storing UID 119.

90. Smith also discloses enabling the above-described portion of ROM 113 upon one of a reset event and power-on event only. According to Smith, the unlocked (or enabled) state of ROM 113—when codes 115 are executed to access UID 119 and commence a secure boot process—corresponds to a reset event: “the processing logic of process 900 *may reset the device to reboot* from the operating

system stored inside the device.” Smith at ¶ 60 (emphasis added). Because a reset initiates a reboot of the system, and a reboot of the system is disclosed to enable access to the UID, Smith discloses to a POSA enabling the above-described portion of ROM 113 upon a reset event.

91. A POSA also would have understood Smith to disclose enabling the above-described portion of ROM 113 upon a power-on event. More specifically, a POSA would have appreciated that a boot process for a computer, including at least Smith’s secure boot process 300, is associated with a power-up event in the same way that a reboot is associated with a reset event. Furthermore, Smith does not disclose any access to UID 119 upon any event other than in the context of a system boot or reboot, which would necessarily follow a power-on event or reset event. A POSA would have also understood that to implement the functionality described in Smith for locking and unlocking access to the above-described portion of ROM 113 necessarily would have required something (such as circuitry and the CPU core executing instructions) to do so.

92. Accordingly, Smith discloses “a locking mechanism configured to enable a portion of said non-volatile memory containing said symmetric encryption key upon one of a reset event or power-on event only.”

d) Limitation 21[pre]: “A computer system, comprising:”

93. To the extent limiting, Smith discloses claim 21’s preamble. Smith’s “invention relates to booting a computing device securely.” Smith at ¶ 1. The one example system that Smith provides for carrying out such secure booting is system 100, depicted in FIGs. 1 and 2. *See, e.g., id.* at FIGs. 1, 2, ¶¶ 30-32, 33 (describing FIG. 2’s system 100 as “executing secure booting”), 38 (describing FIG. 3 as illustrating a process for performing securing booting using system 100), 42 (describing FIG. 4 as illustrating a process performable by system 100 to generate a signature from a code image), 45 (describing FIG. 6 as illustrating a process to securely recover an operating environment from a host to a device performable by system 100). Smith explains that system 100 resides in the device that is the subject of the secure booting. *Id.* at ¶¶ 30-32. Smith then explains that some versions of the invention, which necessarily would have been understood to include system 100, could be part of a media player (like an Apple iPod®) that a POSA would have understood qualified as a “computer system” because it had the basic components of a computer system: a storage device, a display device, an input device, and a processor. *Id.* at ¶¶ 69-71; *see also* Fadell (identified in Smith paragraph 70 as an example media player, and incorporated by reference in Smith) at ¶¶ 44-47 (describing the media player’s components a POSA would have understood reflected same qualified as a “computer system”); *see also* Smith at ¶¶ 66-68 and FIG. 12

(describing and depicting data processing system 1200 that “may be implemented as part of system [100] as shown in FIG. 1” and includes the basis components of a computer system).

e) Limitation 21[a]: “a central processing unit comprising:”

94. Smith’s computer system (identified in Sec. IX.A.1.d)) includes chip 105, which is a central processing unit that satisfies Limitation 21[a] for the same reasons Limitation 19[pre] is satisfied. *See* Sec. IX.A.1.a).

f) Limitations 21[b][i]-21[b][ii]: “a substrate having formed thereon circuit elements defining a CPU core, a volatile random access memory, a non-volatile memory and a bus system for connecting said CPU core, said volatile random access memory and said non-volatile memory, and pre-boot information stored in said non-volatile memory, said pre-boot information including instructions executable by said CPU core and a symmetric encryption key for verifying at least a portion of a boot routine signed by using said symmetric encryption key; and”

95. The central processing unit that satisfies Limitation 21[a] meets Limitations 21[b][i] and 21[b][ii] for the same reasons Limitations 19[a] and 19[b], respectively, are satisfied. *See* Secs. IX.A.1.a) and IX.A.1.b).

g) Limitation 21[c]: “a non-volatile boot memory configured to hold at least said boot routine and a second boot routine.”

96. Smith discloses this limitation. Smith’s computer system (identified in Sec. IX.A.1.d)) includes storage component 109 that takes the form of flash memory

(Smith at ¶ 31, FIGs. 1 and 2), which a POSA would have known was non-volatile memory. '037 Patent at 7:27-32 (indicating “flash memory” is a form of “non-volatile memory”). Smith’s flash memory form of storage component 109 is configured to hold the LLB boot routine, which as stored in storage component 109 is referred to as LLB 229, and a second boot routine in the form of iBoot 231. Both LLB 225 and iBoot 231 are boot routines because each is code that is executed as part of a boot process. Smith at FIGs. 2, 3, ¶¶ 34, 40. Thus, Smith’s computer system includes the claimed “non-volatile boot memory.”

h) Limitation 1[pre]: “A method for booting a computer system, the method comprising:”

97. To the extent limiting, Smith discloses claim 1’s preamble. Smith discloses a method of booting a computer system, such as an Apple iPod®. *See* Smith at Title, FIG. 3, ¶ 38; Sec. IX.A.1.d).

i) Limitation 1[a]: “upon one of a power-on and a reset event, executing instructions stored in an internal non-volatile memory area of a central processing unit,”

98. First, ROM 113 meets the claimed “internal non-volatile memory area of a central processing unit” for the same reasons set forth in Sections IX.A.1.a) and IX.A.1.a)(2) that ROM 113 qualifies as the “non-volatile memory” of claim 19. To the extent PO argues otherwise, ROM 113 is an “internal” area of the CPU (chip 105) at least because ROM 113 is on the same substrate as the CPU’s CPU core. Sec. IX.A.1.a)(5).

99. Second, a POSA would have found it logical and straightforward based on Smith to (as discussed below) make any modification to Smith necessary to enable the execution of instructions stored in ROM 113 upon one of a power-on and a reset event, as recited in claim 1, which execution initializes RAM 111 (“an internal random access memory” as set forth in claim 2) and loads the LLB boot image (“a signed portion of boot data” as set forth in claim 1 and referred to in claim 2 as “said at least a portion of boot data”) from storage component 109 (“a non-volatile memory” as set forth in claim 2) into RAM 111. The execution of the above-described instructions is addressed first (and referred to as the “Mapped Limitation 1[a] Instructions Execution”), followed by why such execution occurs upon one of a power-on and a reset event.

100. Smith discloses that executing instructions in codes 115, stored in ROM 113, causes the loading of code image LLB from storage component 109 into RAM 111. Smith at ¶¶ 30 (codes 115 are stored in ROM 113), 31 (codes 115 cause loading a code image into RAM 111), 32 (system 100, as reflected in FIG. 2, loads the LLB boot image (as LLB 229) from storage component 109 into RAM 111 (as LLB 225)), 38 (describing, as part of process 300 performed by system 100, executing instructions in codes 115 of “a code section of the ROM chip” (which a POSA would have understood as located in ROM 113) to locate a code image “stored in a memory component” (which a POSA would have understood as memory component 109)

and load the located code image “into a memory” (which a POSA would have understood as RAM 111)).

101. As a POSA would have understood, it would not have been possible to load the LLB code image into RAM 111 without first initializing RAM 111. *See* Microsoft, 273 (defining the verb “**initialize**” as “1. To prepare a storage medium, such as a disk or a tape, for use.”). Furthermore, a POSA would have understood codes 115 to accomplish such initialization at least because they are the first codes (and, thus, the first instructions) that are executed in Smith. *See* Smith at ¶ 38. At a minimum, a POSA would have found it logical and straightforward to make any modification to codes 115 (or any other aspect of Smith’s system 100) required so that instructions of codes 115 to (when executed) initialized RAM 111 before loading the LLB code image into RAM 111 because: (1) as explained above, RAM 111 had to be initialized before such loading, otherwise such loading could not have taken place, and (2) it would have been logical to put such instructions in codes 115 stored in ROM 113 because an on-chip ROM (like ROM 113) was where such instructions were typically stored (*see* Stevens at 2:30-61 (minimal amount of BIOS initialization code operative to initialize “the system memory” is stored in a first portion of “the critical nonvolatile storage device,” which can be ROM)).

102. Smith discloses that the Mapped Limitation 1[a] Instructions Execution occurs upon a power-on event because a POSA would have understood the Mapped

Limitation 1[a] Instructions Execution to necessarily occur after the computer system were turned on in order for the computer system to be usable for normal operation.

103. As for the Mapped Limitation 1[a] Instructions Execution occurring upon a reset event, Smith discloses resetting the device (mapped to the “computer system” of the preamble) containing Smith’s system (100) following: (1) securely restoring software components from a host to the device (Smith at ¶ 58 (describing block 821 in FIG. 8: following receipt of a command to “restart the device” “the processing logic of process 800 may reset the device[, which then] reboot[s] from the operating system stored in the mass storage of the device”), FIG. 8 (showing block 821)) and (2) securely updating an application from a host to the device (*id.* at ¶¶ 20, 60 (describing block 911 in FIG. 9: “the processing logic of process 900 may reset the device to reboot from the operating system stored inside the device”), FIG. 9 (showing block 911)). Madisetti, ¶ 121. A POSA would have understood that rebooting the device in each of these two circumstances would have involved either a warm reboot—which did not involve turning off power and turning power back—or a cold reboot—which did involve turning off power to the device and then turning on the power to the device. Microsoft, 442 (defining “**reboot**” as “[t]o restart a computer by reloading the operation system. *See also* boot², cold boot, warm boot.”), 107 (defining “**cold boot**” as “[a] startup process that begins with turning

on the computer's power.”), 561 (defining “**warm boot**” as “[t]he restarting of a running computer without first turning off the power.”).

104. Thus, Smith teaches rebooting (as just discussed) and only two reboot options existed (warm and cold), both of which a POSA would have been aware of and both of which would have produced predictable results. As a result, a POSA would have found it logical and straightforward to try a cold reboot of Smith's device in carrying out Smith's disclosed (but type-unspecified) rebooting. And because a cold reboot would have involved turning the power to the computer system off and then on, the Mapped Limitation 1[a] Instructions Execution would have necessarily occurred thereafter, just as it would have upon the power-on event discussed earlier.

j) Limitation 1[b]: “said internal non-volatile memory area containing an encryption key;”

105. Smith discloses this limitation because ROM 113 contains UID 119 for the same reasons UID 119 is stored in ROM 113 set forth in Section IX.A.1.b)(2), and UID 119 is an encryption key for the same reasons UID 119 is a symmetric encryption key in Section IX.A.1.b)(2).

k) Limitation 1[c]: “determining a current signature of at least a signed portion of boot data,”

106. Smith discloses this limitation. The LLB code image, which is loaded into RAM 111 as LLB 225 (Smith at ¶ 33, FIG. 2), is at least a signed portion of boot data for which a current signature will be determined. A POSA would have

understood the LLB code image as “boot” data because LLB stands for “Low Level Boot” (*id.* at ¶ 33), the LLB code image is described elsewhere as a “boot image” (*see id.* at ¶ 56 (describing receiving boot images in the FIG. 8 restore process), FIG. 8), and its execution involves “booting operations” (*id.* at ¶ 40, FIG. 3 (block 311)). Furthermore, due to the “at least” claim language, the LLB code image also qualifies as the claimed “boot data” because the LLB code image is at least a portion of itself. Should PO disagree, the LLB code image is also at least a portion of itself and any one or more of the other code images stored in storage component 109, which a POSA would have understood are also “boot” data for at least the latter two of the three reasons provided above for the LLB code image: the iBoot code image (231) and the Kernelcache code image (233). *Id.* at FIG. 2, ¶¶ 33-35. The LLB code image is signed because it includes a header value that includes “a signature value signed over the code image according to [UID 119] through well-known hashing and encryption algorithms.” *Id.* at ¶¶ 39 (explaining such signature value of a code image), 40 (reflecting that the LLB code image is a code image to which paragraph 39’s discussion applies).

107. Next, Smith discloses “determining a current signature” of the loaded LLB code image. In particular, Smith recites, “the processing logic of process 300 derives another signature value from the [LLB] code image according to [UID 119] through the same well-known hashing and encryption algorithms.” *Id.* at ¶ 39; *see*

also id. at ¶ 42 (describing how UID 119 is used to determine a current signature of the loaded LLB code image), FIG. 4. Smith’s deriving of another signature value from the LLB code image constitutes determining a current signature of the LLB code image loaded into RAM 111.

l) Limitation 1[d]: “said signed portion of boot data containing a stored signature of said portion of boot data,”

108. Smith discloses this limitation. The signature value signed over the LLB code image discussed above in Section IX.A.1.j), which Smith also refers to as “the extracted signature value” (Smith at ¶ 39), qualifies as the claimed “stored signature” of the LLB code image.

m) Limitation 1[e]: “said stored signature created by applying said encryption key; and”

109. Smith discloses this limitation. As explained above in Section IX.A.1.j), the stored signature is created by applying the encryption key (i.e., UID 119). That is, Smith recites that the LLB code image’s “signature value [is] signed over the code image according to [UID 119] through well-known hashing and encryption algorithms.” Smith at ¶¶ 39 (explaining such signature value of a code image), 40 (reflecting that the LLB code image is a code image to which paragraph 39’s discussion applies).

- n) **Limitation 1[f]: “executing boot instructions contained in said signed portion of boot data, when said current signature and said stored signature are identical.”**

110. Smith discloses this limitation. The loaded LLB code image is executed if, and thus when, the determined current signature and the stored signature match. Smith discloses comparing the determined current signature (“the derived signature value”) to the stored signature value (“the extracted signature value”) “to verify whether the code image is trusted.” Smith at ¶ 39. Smith then explains that verification is successful “if the derived signature value and the extracted signature match with each other[, i.e., are identical].” *Id.* Finally, Smith explains that the LLB code image is executed if the verification is successful, which a POSA would understand as synonymous with when the verification is successful. *Id.* at ¶ 40, FIG. 3 (showing blocks 307 and 311). A POSA would have understood such execution as the execution of boot instructions contained within the LLB code image.

111. Thus, Smith discloses “executing boot instructions contained in said signed portion of boot data, when said current signature and said stored signature are identical,” as claimed.

- o) Claim 2: “The method of claim 1, wherein executing said instructions stored in said internal non-volatile memory area comprises initializing an internal random access memory of said central processing unit and loading said at least a portion of boot data from a non-volatile memory into said initialized random access memory.”**

112. Claim 2 is met for the reasons set forth in Section IX.A.1.i) that the Mapped Limitation 1[a] Instructions Execution initializes RAM 111 and loads the LLB code image from storage component 109 into initialized RAM 111.

- p) Claim 3: “The method of claim 2, wherein executing boot instructions of said signed portion of boot data comprises initializing a system memory of said computer system and loading said boot data from said non-volatile memory into said system memory and verifying an integrity of said boot data.”**

113. For this claim, which depends from claim 2, I map the recited signed portion of boot data to the iBoot code image, rather than the LLB code image. My analysis of Limitation 1[a] in Section IX.A.1.i) applies with equal force to loading the iBoot code image into RAM 111 for at least the following reasons.

114. Smith’s FIG. 3 and the associated description thereof concern a process for secure booting. *See* Smith at ¶¶ 38-41, FIG. 3. The process includes executing instructions (such as codes 115) stored in secure ROM 113 to locate a code image within the device (i.e., the claimed “computer system,” as explained in Section IX.A.1.h)) that is being securely booted. Smith at ¶ 38, FIG. 3. The code image in question can be stored (and therefore located) in storage component 109. *Id.* at ¶ 38

(explaining the code image can be a “storage component” of the device, such as “a flash memory or a mass storage device,” which a POSA would have understood to refer to storage component 109, which “may be a flash memory[] ... or other mass storage” device (*id.* at ¶ 31)); *id.* at ¶ 38 (explaining “process 300 may be performed by system 100 of FIG. 1”), FIG. 1. Once found, the code image is then loaded “into a memory” (*id.* at ¶ 38), which a POSA would have understood to be either RAM 111 or memory component 103—the only memories into which Smith discloses loading a code (from storage component 109) using system 100 (*id.* at ¶ 31). After the loaded code image is verified based on UID 119 (*id.* at ¶ 39)—which, as applied to the iBoot code image, satisfies Limitations 1[c]-1[e] for the same reasons discussed above in Sections IX.A.1.k)-IX.A.1.m) with respect to the LLB code image—the loaded code image is executed (Smith at ¶ 40). Smith discloses iBoot 227 as one such code image (*id.* at ¶ 40), which Smith shows loaded into RAM 111 in FIG. 2 (*id.* at FIG. 2).

115. Correspondingly, claim 2—just as it was met for the reasons set forth in Section IX.A.1.i) that the Mapped Limitation 1[a] Instructions Execution initializes RAM 111 and loads the LLB code image from storage component 109 into initialized RAM 111 based on my claim 1 mapping of the LLB code image to the signed portion of boot data—would also have been met for those same reasons applied to the iBoot code image, as explained in paragraph 114.

116. Next, executing boot instructions contained in the iBoot code image (Smith at ¶ 40) includes the “initializing” and “loading” of Limitation 3[a] for at least the following reasons. Smith describes the execution of the loaded and verified code image as the performance of booting operations that include “device hardware initialization.” *Id.* at ¶ 40. With the iBoot code image loaded into RAM 111 (as explained above), memory component 103 would have been device hardware that need not have been initialized at that point, and Smith elsewhere describes iBoot as a code image that, when executed, locates and loads “into a memory component of the device” the Kernalcache code image (*id.* at ¶ 52⁵), which is “boot” data (as explained above in paragraph 106) and thus satisfies “said boot data.” *See* ’037 Patent at 6:34-44 (describing the copying of the “remaining boot information”), 9:9-17 (“In one illustrative embodiment, the non-volatile memory 120 may contain the boot information divided into two portions, wherein a first portion may comprise data and instructions for initializing other system components, such as the system memory 110 for accommodating *the second portion of boot information* and

⁵ While this paragraph concerns a recovery process, Smith is addressing the same iBoot code image Smith addresses elsewhere, so a POSA would have understood the iBoot code image to, when executed as described in Smith paragraph 41, perform the same operations described in Smith paragraph 52.

executing *the second portion* once the first portion has been verified on the basis of the protected environment provided by the CPU 100.” (emphasis added)), 10:14-31 (“Thus, if the portion 121A may be provided, as shown in FIG. 1 c, this portion may serve as an initial BIOS loader (IBL) for *a second portion 121B*, wherein the first portion 121A may also contain a hash value obtained over the portion 121B. ... The portion 121A may, therefore, be contained in instructions for performing necessary platform initialization actions, such as initialization of a memory controller 141 and the like, followed by *the shadowing of the portion 121B* from the memory 120, i.e., from one of the portions 120A, 120B, into the initialized system memory 110.”), FIG. 1c. As a result, a POSA would have interpreted the referenced “device hardware initialization” to include the initialization of memory component 103. At a minimum, it would have been logical and straightforward to a POSA for the iBoot code image to have been modified to include boot instructions that caused such locating and loading of the Kernelcache code image because (1) a code image could not have been stored in memory component 103 until memory component 103 had been initialized, (2) Smith does not otherwise describe the initialization of memory component 103, and (3) Smith discloses executing the iBoot code image from RAM 111 as a predicate to locating the Kernelcache code image and loading same into memory component 103.

117. Finally, although Smith paragraph 41 does not state that the execution of the loaded and verified iBoot code image verifies the Kernelcache code image, Smith does disclose that such verification—which a POSA would have understood as “verifying an integrity of [the Kernelcache code image],” as claimed—takes place, both (1) as part of the booting process depicted in FIG. 3 in which, after execution of the iBoot code image, control passes from block 317 to block 319, for finding the Kernelcache code image, then to block 303, for loading the Kernelcache code image into memory, and then to block 305, for verifying that Kernelcache code image (Smith, FIG. 3, ¶¶ 40-41), and (2) during a recovery process in which Smith describes verifying the located and loaded-into-memory Kernelcache code image to be a trusted code image “according to a process such as described at block 305 of FIG. 3” (*id.*, ¶ 52). A POSA would have found it logical and straightforward for the execution of the iBoot code image to have performed such verification, considering that: (1) the iBoot code image would have been trusted at that point, and (2) Smith discloses the execution of one code image can be used to verify a second code image (*see* Smith at claims 1-3 (describing the execution of a first verified code image verifies a second code image); *see also id.* at ¶ 54 (describing the execution of the Kernelcache code image locate, load, and verify an operating system component during a recovery process)). And a POSA would have considered a modification to Smith’s iBoot code image that caused the iBoot code image to, when executed,

perform such verifying of the Kernelcache code image to be the application of a Smith-disclosed verification approach (i.e., one verified code image being executed to verify a second code image, as described in Smith claims 1-3) to achieve a Smith-disclosed verification of the Kernelcache code image, which is one of only three disclosed code images (LLB, iBoot, and Kernelcache), while yielding nothing unpredicted.

q) Claim 5: “The method of claim 1, further comprising locking said encryption key for external access prior to executing said boot instructions.”

118. Claim 5 recites that the method of claim 1 further comprises “locking said encryption key for external access prior to executing said boot instructions.” As explained in Section IX.A.1.i), it would have been logical and straightforward to a POSA to try a cold reboot of the computer system, which would have cycled the power thereto off and on. That turning off of the power to the computer system would have locked UID 119 for external access prior to executing the boot instructions (I addressed such executing in Section IX.A.1.n)), and been followed by the performance of claim 1’s limitations, as described above in Sections IX.A.1.i)-IX.A.1.n).

r) Claim 6: “The method of claim 5, further comprising determining whether an upgrade version of said boot data is available.”

119. Smith discloses “determining whether an upgrade version of said boot data is available” because Smith discloses determining whether an updated version of the LLB code image (which qualifies as the claimed “boot data” as explained in Section IX.A.1.j)) is available. Smith discloses updating the software and the firmware of the computer system (*see* Smith at ¶ 6 (explaining the need for security and the flexibility to update the device’s software and firmware); Sec. IX.A.1.g) (explaining why the claimed method concerns a computer system with reference to Section IX.A.1.d)). Smith also discloses a particular mode for accomplishing such updates: DFU mode, which stands for “device firmware upgrade” mode. Smith at ¶ 38; *see also id.* at ¶ 43 (describing some purposes of, and some conditions for entering, DFU mode). A POSA would have understood Smith’s reference to “firmware” in “D[evice] F[irmware] U[pgrade]” mode to encompass boot data, such as the LLB code image, because Smith associates DFU mode with the recovery of Smith’s code images when they cannot be verified. Smith at ¶¶ 48-54, FIG. 7.

120. In particular, Smith discloses the computer system—after not successfully locating and verifying the LLB code image—being coupled to a host computer, publishing a status on the computer system’s recovery state, receiving from the host computer an LLB code image, and loading and verifying that the

received LLB code image is trusted, after which that received LLB code image is executed. *Id.* at ¶¶ 48-50, FIG. 7. A POSA would have understood the LLB code image received from the host computer to be an upgrade relative to at least an existing LLB code image that could not be verified; a POSA would have understood such an unverified image to possess a problem of some kind (such as damage or corruption) that a POSA would not have understood the new image to possess.

121. Furthermore, a POSA would have understood that Smith discloses determining whether an upgrade version of the LLB code image is available in each of the following ways, any one of which is adequate. First, the entry of the computer system into DFU mode (shown in FIG. 7 as “DFU 713”) establishes such a determination because the computer system would not have entered that upgrade mode if no upgrade were available. Smith at ¶ 48, FIG. 7. Second, the individual who “coupled” the computer system to the host computer for receiving the upgraded LLB code image made the determination that such upgraded code image was available; otherwise, they would not have coupled the computer system to the host computer. *Id.* at ¶ 49. Third, the computer system’s publication of its status to the host computer and subsequent receipt of the host-computer-transmitted LLB code image (as described in Smith ¶ 49) involved determining such LLB code image was available.

122. To the extent PO disagrees that Smith's paragraphs 48-50 and FIG. 7 disclose the claimed determining, such determining is also disclosed in Smith paragraphs 59-60 and FIGs. 5 and 9. There, Smith discloses establishing a network connection between the computer system and a host computer (such as is shown in FIG. 5), publishing by the computer system to the host computer a status indicating the computer system is an "update mode," and receiving by the computer system from the host computer a code image that includes a software package related to an updated version of an application of the computer system. Smith at ¶¶ 59-60, FIGs. 9 and 5. A POSA would have understood the updated LLB code image received from the host computer to be an upgrade relative to the non-updated LLB code image being replaced. And because Smith explains that process 900 described in paragraphs 59-60 is performable with system 100 (Smith at ¶ 59), and the LLB code image is one of only three code images specifically identified by Smith for system 100 (*see id.*, *e.g.*, ¶ 40), a POSA would have understood the code image referenced in Smith paragraph 60 to apply to the LLB code image.

123. Furthermore, a POSA would have understood that Smith's paragraphs 59-60 and FIGs. 5 and 9 disclose determining whether an upgrade version of the LLB code image is available in each of the following ways, any one of which is adequate. First, the entry of the computer system into "update mode" (*see* block 903 in FIG. 9) establishes such a determination because the computer system would not

have entered that update mode if no update were available. Smith at ¶ 59, FIG. 9. Second, the individual who enabled the establishment of “a network connection” between the computer system and the host computer for receiving the updated LLB code image made the determination that such updated code image was available; otherwise, they would not have enabled that connection between the computer system and the host computer. *Id.* at ¶ 59, FIGs. 5, 9. Third, the computer system’s publication of its status to the host computer and subsequent receipt of the host-computer-transmitted LLB code image (as described in Smith ¶¶ 59-60 and shown in FIG. 9) involved determining such LLB code image was available.

B. Smith and Bolotin

1. Smith and Bolotin versus Claim 20

124. I have been informed by counsel that the phrase “a locking mechanism configured to enable a portion of said non-volatile memory containing said symmetric encryption key upon one of a reset event and power-on event only” in claim 20 may be interpreted as a “means-plus-function” term. If so, I understand that the claim language will be interpreted to cover the structures(s) (and their equivalents) disclosed in the ’037 Patent for performing the function recited in the claim, which I understand are referred to as “corresponding structure.”

125. To a POSA, the specification identifies only lock mechanism 105 in this regard, which is illustrated schematically on CPU 100 in FIG. 1a. ’037 Patent

at 7:53-8:8, FIG. 1a. The '037 Patent explains that the lock mechanism 105 may comprise “circuitry for shutting down supply energy for the memory [secure ROM] 103.” *Id.* at 7:60-66. And to “enabl[e]” the memory, a POSA would have understood mechanism 105 to be able to provide power to the memory, at least in view of the '037 Patent’s description that the lack of power to the memory renders the memory non-responsive. *Id.* at 7:60-66. Therefore, a POSA would have understood the corresponding structure to be circuitry for enabling or disabling power to the secure ROM containing the symmetric encryption key.

a) Overview of Bolotin

126. Bolotin, which is entitled “Memory Lock System,” is analogous art to the invention of claim 20 because, as the balance of this Bolotin overview and the subsequent discussion of the motivation-to-combine reflects, Bolotin concerns at least one problem faced by the '037 Patent’s inventors: controlling access to memory containing sensitive information. *See* Bolotin at Title; '037 Patent at 3:11-25 (discussing controlling access to pre-boot information), 3:56-61 (discussing storing same in non-volatile internal memory). In particular, Bolotin discloses an embodiment of a self-contained and independent memory lock system that “includes a controller, such as a micro-controller 212, connected by a communication channel 214 to a memory 216, such as a flash memory, micro disk, or other file storage

system.” *Id.* at 5,⁶ 2, FIG. 2. A POSA would have understood such memory to be non-volatile. *See* ’037 Patent at 7:27-32 (indicating “flash memory” is a form of “non-volatile memory”). The microcontroller is also connected to a power connector 204. Bolotin at 5, FIG. 2.

127. “The micro-controller 212 can have part of its operations dedicated to a data transfer function between the connector 204 and the memory 216 and another part to the locking function.” *Id.* at 5, FIG. 2. “[P]ower for the memory lock system 200 can be drawn from the electronic connector 204 from the electronic system to which it is connected.” *Id.* at 5, FIG. 2; Madisetti, ¶ ___. Bolotin explains that:

Where the power for the memory lock system 200 is drawn from the electronic connector 204, the micro-controller 212 can use the locking function controller 215 to shut down the power to or reset the state of the data transfer function controller 213 to prevent or lock data transfer in the micro-controller 212. The micro-controller 212 can also shut down power to or otherwise disable the memory 216 to prevent or lock data transfer in the memory.

Bolotin, 5, FIG. 2.

128. Bolotin also explains that “compare circuitry or firmware” of memory lock system 200, which a POSA would have understood to correspond to the schematically-illustrated micro-controller 212 and locking function controller 215 of FIG. 2, is what renders the system unlocked and operable. *Id.* at 5, 8-9. A POSA

⁶ Bolotin citations are to native page numbers.

would have therefore understood from this that Bolotin discloses circuitry for locking and enabling power to reach a memory containing sensitive information.

b) Motivation to modify Smith's chip 105 chip to include circuitry like that provided by Bolotin's micro-controller 212 and locking function controller 215

129. A POSA would have been motivated to modify Smith's chip 105 with circuitry like that provided by Bolotin's micro-controller 212 and locking function controller 215 to provide a self-contained means in Smith's computer system for enabling or preventing external access to sensitive information stored in ROM 113 (such as UID 119). As a POSA would have known, the proliferation of computing devices resulted in an acute need to improve the security of computer systems, including hand-held computing devices. *See* Bolotin at 1 ("Security is a critical issue with almost all aspects of computer use and mobile electronic device use."), 2-3 (security for the data on portable storage devices and electronic products using them "is greater than ever").

130. Bolotin teaches circuitry (*id.* at 5, 8-9, FIG. 2) that can shut down or enable power to a memory in order to lock or enable data transfer with the memory. As a POSA would have appreciated from Bolotin's disclosure, controlling power to a memory through circuitry was a straightforward and effective way to predictably and securely control access to the memory. *Id.* at 5, 8-9, FIG. 2. Bolotin also teaches that its memory lock systems (which includes system 200) can be integrated

internally with computer systems like “iPODs.” *Id.* at 8. Bolotin thus teaches the desirability of using circuitry to allow or shut off power to memory as an effective means of allowing or preventing access to the memory in devices like the computer system disclosed in Smith. *Id.* at 5, 8-9, FIG. 2; Sec. IX.A.1.d) (explaining why Smith discloses secure booting of a computer system like an Apple iPod®).

c) Smith’s chip 105, modified in view of Bolotin, meets claim 20

131. Motivated as explained above, a POSA would have modified Smith’s chip 105 in view of Bolotin to provide circuitry like that provided by Bolotin’s micro-controller 212 and locking function controller 215 for achieving Smith’s already-disclosed functions of enabling or disabling access to at least the portion of secure ROM 113 that includes UID 119 (*see* Sec. IX.A.1.c)). As modified, Smith’s chip 105 would have included circuitry for enabling/disabling access to ROM 113 (and, thus, UID 119) under the conditions discussed in Section IX.A.1.c). Because Smith’s ROM 113 is on chip 105 along with chip 105’s CPU core (*see* Section IX.A.1.a) and its subsections), it would have been straightforward for a POSA to have modified chip 105 to predictably implement such circuitry in view of the above-discussed disclosures of Bolotin. Such circuitry meets the corresponding-structure circuitry under a means-plus-function construction of the “locking mechanism” element (*see* paragraphs 124-125).

C. Smith and Little Versus Claims 19 and 21, and Smith, Bolotin, and Little Versus Claim 20

132. As explained in IX.A.1.a) (and, in particular, Section IX.A.1.a)(3)), Smith's system includes a CPU core that implements his secure-boot processes, and a POSA would have found it logical and straightforward for Smith's chip 105 to include it. To the extent that PO contends otherwise, Little discloses an "on-chip ROM [that] is physically integrated with [a] processor" (Little at Abstract; *see also id.* at ¶¶ 7, 30, 35, FIG. 2), and a POSA would have found it logical and straightforward to configure Smith's chip such that Smith's CPU core was accordingly integrated with Smith's on-chip ROM 113.

133. Little's ROM 19 "contains the authentication code, including one or more public keys, required for authenticating the program software residing in external memory 116 in known fashion." Little at ¶ 31, FIG. 2. Smith's ROM 113 also stores "a fingerprint 123 based on a signature from a trusted entity such as a public key certificate" that can be used to verify whether an operating system component stored in storage component 109 can be trusted and therefore executed. Smith at ¶¶ 32, 35, FIG. 2. And like Little's authentication code in Little's ROM 19, Smith teaches that the storage of fingerprint 123 in Smith's ROM 113 is secure. *See* Little at Abstract (touting that "the code stored in ROM can neither be tampered with nor replaced, and can therefore be assumed at all times to be authentic"), ¶¶ 5, 7, FIG. 2; Smith at ¶¶ 29 (the "secure ROM" stores the fingerprint), 28 (describing the

secure ROM and that its content may be stored during the manufacturing stage), FIGs. 1 and 2 (depicting “Secure ROM” 113 containing “Fingerprints” 123).

134. To provide such security, Little explains that “[t]he on-chip ROM is physically integrated with the processor itself, so the code stored in ROM can neither be tampered with or replaced.” *See* Little at Abstract; *see also id.* at ¶¶ 7, 34. Particularly in view of their aligned goals, a POSA would have found it logical and straightforward to do the same in Smith—i.e., incorporating Smith’s “Secure ROM” (i.e., ROM 113) into Smith’s CPU core—to achieve the same benefit.

135. Thus, should PO disagree with the analysis in Section IX.A.1.a)(3) that a POSA would have found it logical and straightforward to modify Smith’s chip 105 to include the CPU core, it remains my opinion that Smith meets claims 19 and 21 as set forth above in Section IX.A.1, and that Smith in view of Bolotin meets claim 20 as set forth above in Section IX.B.1, except for the modification to Smith’s chip 105 in view of Little (as just discussed), which impacts my Section IX.A.1.a) (concerning Limitations 19[pre]-19[a]) and my Sections IX.A.1.e)-IX.A.1.f) (concerning Limitations 21[a]-21[b][i]).

D. Smith and Challenger Versus Claims 5 and 6

136. Should PO disagree with the analysis above in Section IX.A.1.q), a POSA would nonetheless have found that Smith modified in view of Challenger meets claim 5.

1. Overview of Challenger

137. Challenger is analogous art to the '037 Patent at least because both concern the secure storage of sensitive information in computer systems. *See* '037 Patent at 3:11-19; Challenger at ¶ 1. In addition, and further reflecting that Challenger is analogous art to the '037 Patent, both concern ensuring a symmetric encryption key is securely stored in a computer system. *See* '037 Patent at Abstract; Challenger at Abstract.

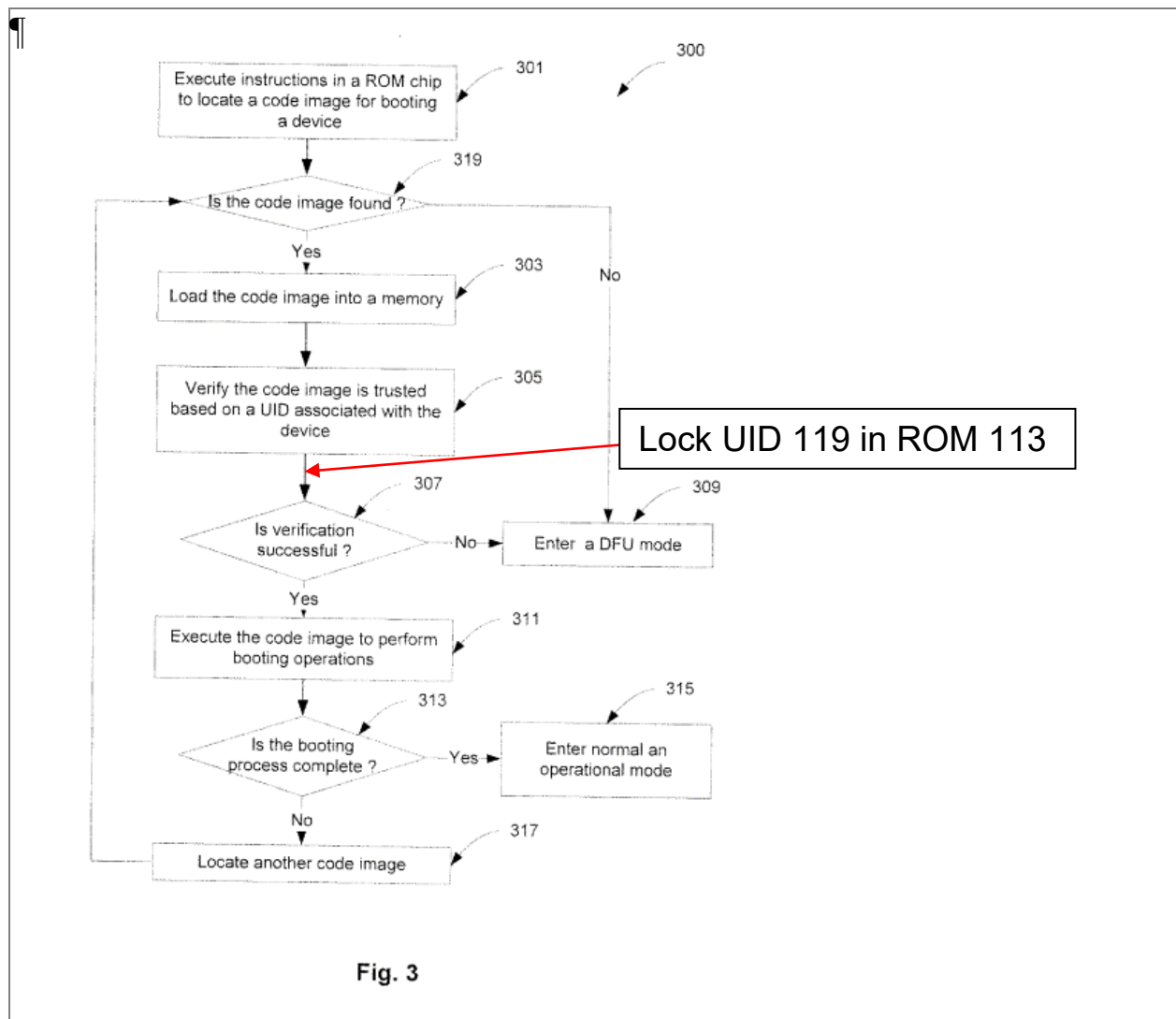
138. Challenger discloses making available (i.e., unlocking), at power-on, system configuration and access data—that, during POST (Power-On-Self-Test), is “used to verify the identity [of the] user and to determine the appropriate access rights and Boot devices”—in a non-volatile memory. Challenger at ¶ 2. Challenger also discloses locking the system configuration and access data from external access after such use of same but before executing boot instructions. *Id.* Specifically, Challenger recites: “Just before boot, POST ... locks the system configuration and access data in a [non-volatile random access memory] NVRAM device.” Challenger at ¶¶ 2, 18. As a POSA would have appreciated, the aspect or aspects of the system configuration and access data Challenger locks just before boot were not needed during boot and experienced improved security as a result of the locking. *Id.* at ¶ 2.

2. A POSA would have been motivated to modify Smith in view of Challenger

139. A POSA would have been motivated to modify Smith's system 100 (including the relevant processing logic of process 300 performable by system 100) in view of Challenger to lock external access to UID 119 (in ROM 113) after UID 119 is used to verify the LLB code image is trusted but before executing the LLB code image (which Smith characterizes as the performance of booting operations (Smith at FIG. 3 (block 311), ¶ 40), similar to Challenger's disclosure of using system configuration and access data to verify a user's identity and determine appropriate access rights and then locking such data prior to booting (Challenger at ¶ 2). Furthermore, a POSA also would have been motivated to modify the processing logic of process 300 to unlock UID 119 just prior to the time it is next needed, which would have been to verify the next code image (as reflected in the process flow of Smith's FIG. 3) to enable the use of UID 119 as Smith describes. *See* Smith at ¶ 41, FIG. 3 (showing that, when the booting process is not complete, the process flow includes the verification of another code image using UID 119). As a POSA would have appreciated, in both Challenger and Smith, the locked data is not (in the case of Challenger), or would not be (in the case of Smith, modified as discussed), needed during the locked time period, and external access to the locked data might have otherwise been possible.

3. Modification of Smith in view of Challenger

140. Motivated as discussed above, a POSA would have found it logical and straightforward to modify Smith's system 100 (including the relevant processing logic of process 300 performable by system 100) to lock UID 119 (in ROM 113) to external access just after UID 119 was used to verify the LLB code image (and, thus, just before the verified LLB code image was executed) and then unlock UID 119 just prior to its use to verify the next code image. A POSA had the knowledge and skill to accomplish this by modifying the processing logic of process 300. *See* Smith at ¶¶ 38-41 (describing how process 300 is performed by system 100 using process logic), FIG. 3. Specifically, a POSA would have modified such processing logic to lock UID 119 to external access after block 305 (where it is used to verify the LLB code image) and prior to block 311 (where the LLB code image is executed to perform booting operations), and then unlock UID 119—as necessary—prior to verifying the next code image (located and found at blocks 317 and 319, respectively) at the next performance of block 305 following a determination that the booting process is not complete (at block 313). This locking modification is shown in the annotated version of Smith FIG. 3 below (the unlocking modification has not been shown):



141. Smith confirms the feasibility of these modifications. Specifically, Smith discloses that, in another process involving the execution of an unverified code image, processing logic can be used to “disable access[] to a UID of a secure ROM in a device such as UID 119 [stored in ROM 113] in FIG. 1.” Smith at ¶ 61. Smith discloses additional techniques for disabling access to UID 119 depending on computer system diagnostic or testing requirements (configuring a trusted code image to, when executed, “turn off access[] to the UID” and using a hardware switch

with settings “that turn off access[] to the UID”) that: further reinforce that a POSA had the skill to perform the proposed modification, and reflect that controlling external access to UID 119 under certain circumstances was warranted. *Id.*

4. Claim 5: “The method of claim 1, further comprising locking said encryption key for external access prior to executing said boot instructions.”

142. Smith, modified as discussed above in Section IX.D.3 in view of Challenger, includes locking UID 119 (in ROM 113) for external access prior to the execution of the LLB code image. Such locking is depicted in the annotated version of Smith FIG. 3 depicted above, which occurs after UID 119 is used as described in Section IX.A.1.j) to determine a current signature of the loaded LLB code image (corresponding to block 305 depicted in FIG. 3 and described in Smith paragraph 39) and before execution of the loaded LLB code image (corresponding to block 311 depicted in FIG. 3 and described in Smith paragraph 40). Smith at FIG. 3, ¶¶ 39-40.

5. Claim 6: “The method of claim 5, further comprising determining whether an upgrade version of said boot data is available.”

143. Smith discloses the “determining” of claim 6, which depends from claim 5, for the same reasons given above in Section IX.A.1.r).

144. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: 4/20/22



Vijay K. Madisetti

ATTACHMENT A

Dr. Vijay K. Madiseti

Fellow, IEEE

vkm@madiseti.com

Cell: 770-527-0177

Address:

**56 Creekside Park Drive
Johns Creek, GA 30022**

Employment:

- 1989-present: Full Professor of Cybersecurity & Privacy, and Electrical & Computer Engineering (**Georgia Tech, Atlanta, GA 30332**).

Areas of Technical Interest – Enterprise Software, Cloud Computing Embedded Systems, Wireless & Mobile Communications, Computer Engineering, Circuit Design (Analog/Digital), Biomedical Instrumentation Hardware/Software, Software Engineering, Digital Signal Processing, Sensors, Wireline & Wireless Computer Networks, Software Systems, Control Systems, Chip Design, Cloud Computing.

Education: **PhD** (EECS, University of California at Berkeley, 1989), **B.Tech** (Hons) in Electronics and Electrical Communications Engineering (Indian Institute of Technology, Kharagpur, 1984).

Startup Companies:

Director, **VP Technologies, Inc.** (1995-): A startup commercialized through Georgia Tech's Advanced Technology Development Corporation (ATDC) focusing on digital software and hardware design services for military market. <http://www.vptinc.net>

Director, **Soft Networks, LLC** (2001-2007): A startup commercialized through Georgia Tech support focusing on software development tools and compilers for Cellular/WiFi/VOIP/telecommunication products. <http://www.soft-networks.com>

Director, **Elastic Video Inc.** (2007- 2009): A startup commercialized through Georgia Tech's VentureLab (<http://venturelab.gatech.edu>) development image and video processing software for wireless & IP networking.

Litigation Experience (2015-2021) With Testimony

(Note: There may be multiple cases between the parties, e.g., District Court v. ITC, US versus Foreign Cases)

Chrimar v. Adtran, Alcatel, et al.

Technology: Power over Ethernet (2015-2017)

Submitted reports & Deposition & Trial

Chamberlain v. Ryobi/TTI

Case No: 1:16-cv-06097 (ND Illinois)

Expert for Ryobi

Technology: Wireless/IoT/Barrier Movement (2016 – 2019)

Submitted reports & deposition & trial testimony

IOEngine v. IMC/Imation

Case No: cv-14-1572-GMS (US Delaware)

Expert for IMC/Imation

Technology: Networked Storage Device (2016-2017)

Submitted reports & deposition & trial testimony

Huawei v. Samsung

Case No: 3:16-cv-2787-WHO (ND Cal)

Expert for Samsung

Technology: 4G/LTE Random Access Protocols (2016-2019)

Submitted reports & deposition

Hitachi Maxell v. ZTE/Huawei

Case No: 5:16-cv-00178-RWS (ED Texas)

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted reports & deposition (2017 – 2018)

Hitachi Maxell v. Apple

Case No: 5:19-cv-00036-RWS

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted Expert Reports (2020-)

Qualcomm v. Apple

Case No: 17-ccv-0108-GPC-MDD (SD Cal) Also, Related ITC/FTC Matters

Expert for Qualcomm

Technology: 4G/Wireless Communications/Smartphones (2017-2019)

Submitted Reports and Deposition

Qualcomm v. Apple

Case No: 3:17-cv-01375-DMS-MDD (SD Cal)

Expert for Qualcomm

Technology: 4G/Wireless Communications/Smartphones (2017-2019)

Submitted Reports and Deposition

Optis v. Huawei

Case No: 2:17-cv-123 (E.D. Texas)

Expert for Optis Wireless

Technology: 4G/Video (2017-2019)

Submitted reports & deposition

Beckman Coulter v. Sysmex

Case No: 1:17-cv-24049-DPG (ND Illinois)

Expert for Sysmex

Technology: Medical Instrumentation Automation (2017-present)

Testifying Expert

TQ Delta v. Xyxel/Adtran

Expert for TQ Delta

Technology: DSL Technologies (2018-present)

Testifying Expert

3GL/KPN v. LG, Blackberry, HTC

Expert for 3GL/KPN

Technology: 4G/LTE Protocols (2018-present)

Reports and Deposition / Testifying Expert

Rovi v. Comcast

ITC No. 337-TA-1103 (ITC)

Expert for Rovi

Technology: Digital Video & interactive GUI (Jan 2018-May 2018)

Reports and Deposition

Cirba/Densify v. VMWare

Case No: 1:19-cv-00742-LPS

Expert for Cirba/Densify

Technology: Virtualization (2019-present)

Reports Deposition/PI Hearing / Testifying Expert

Power Integrations v. On Semiconductor

Case No: 17-cv-03189-BLF

Expert for On Semiconductor

Technology: Power Electronics (2018-2019)

Reports & Deposition/Testifying Expert.

Pathway v. Lumens, IPEVO, Aver JDA

Case No: ITC 337-TA-1045

Expert for Lumens JDA

Technology: Document Cameras (2016-2019)

Reports & Deposition (for IPR/ITC, no trial)

Wilan v. Apple

Case No: 3:14-cv-2235

Expert for WiLan

Technology: Voice over LTE/LTE Protocols (2017-2020)

Reports and Testified through Deposition/Trials

St. Lawrence Comm. v. Amazon

Case No. 2:19-CV-00027-JRG

Expert for Amazon

Technology: Multimedia Codecs

Submitted Expert Reports/Deposition (2020-2020)

PanOptis v. Apple

Case No. 2:19-CV-00066-JRG

Expert for PanOptis

Technology: LTE

Submitted Expert Reports/Deposition/Trial (2019-2020)

GAS v. Sprint Communications

Case No. 2:20-cv-00007-RWS

Expert for General Access (GAS)

Technology: LTE

Submitted Expert Reports/Deposition (9/21-)

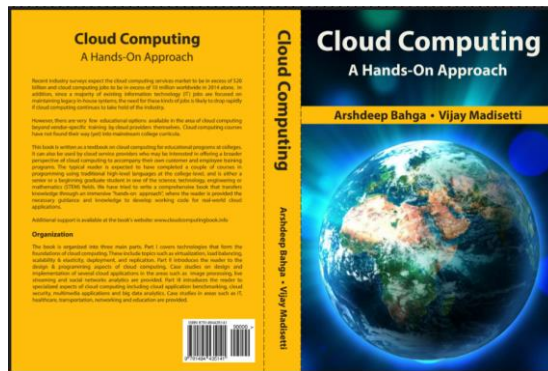
Additional matters include declarations supporting IPRs at the PTAB for Google (US Patent 8,601,154), On Semiconductor (US Patent 6,212,079), Ubisoft (US Patent 5,490,216), Lumens JDA (2017-2019), Broadcom (WiFi), Sony (US Patent 6,101,534), Kia, (US Patent 5,530,431), Qualcomm, Ericsson, Daimler Benz, VW, Estech, Amazon, Ring, Digital Ally, On Semiconductor, United Patents, Lenovo, BMC Software, BMW, Daimler and Dolby.

Earned Degrees

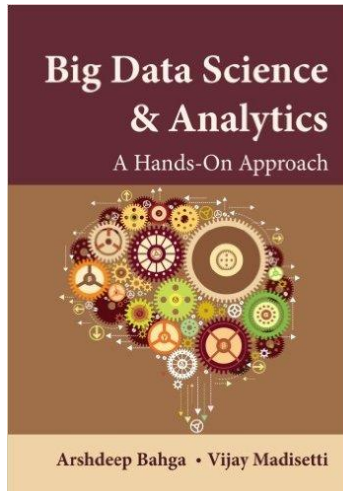
- 1. B. Tech (Hons), Electronics & Electrical Comm. Engineering**
Indian Institute of Technology (IIT), Kharagpur, India
1984.
- 2. Ph.D., Electrical Engineering & Computer Sciences (EECS)**
University of California (UC), Berkeley. CA
1989.

Books

- 1. VLSI Digital Signal Processors**
Madiseti, V.K.;
Boston: MA, IEEE Press: Butterworth Heinemann, 1995, 525 pp.
- 2. Quick-Turnaround ASIC Design in VHDL**
Romdhane, M., Madiseti, V.K., Hines, J.
Boston: MA, Kluwer Academic Press, 1996, 190 pp.
- 3. The Digital Signal Processing Handbook (First Edition)**
Madiseti, V. K., Williams, D. (Editors)
CRC Press, Boca Raton, Fla, 1998, 2500 pp.
- 4. VHDL: Electronics Systems Design Methodologies.**
Madiseti, V. K. (Editor)
Boston: MA, IEEE Standards Press, 2000, ISBN 0-7381-1878-8.
- 5. Platform-Centric Approach to System-on-Chip (SoC) Design.**
Madiseti, V. K., Arpnikanondt, A.
Springer, Boston: MA, Springer, 2004, 280 pp.
- 6. The Digital Signal Processing Handbook – Second Edition.**
Madiseti, V. K. (2009)
CRC Press, Boca Raton, Fla.
- 7. Cloud Computing: A Hands-On Approach**
A Bahga, V. Madiseti (2013)
Amazon CreateSpace Publishing, 2013, 454 pp.
- 8. Internet of Things: A Hands-On Approach**
A Bahga, V. Madiseti (2014)
Amazon CreateSpace Publishing, 2014, 450 pp.



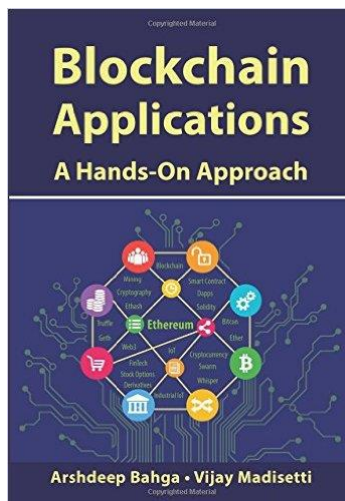
- 9. Big Data Science & Analytics: A Hands-On Approach**
A Bahga, V. Madisetti (2016)
 Amazon CreateSpace Publishing, 2016, 542 pp.



10. Blockchain Applications: A Hands-On Approach

A Bahga, V. Madiseti (2017)

Amazon CreateSpace Publishing, 2017, 380 pp.



Edited Books & Collection of Papers

1. Advances in Parallel & Distributed Simulation

Madiseti, V.K.; Nicol, D., Fujimoto, R. (Editors)

San Diego, CA: SCS Press, 1991, 200 pp.

2. Modeling, Analysis, Simulation of Computer & Telecommunications Systems

Madiseti, V., Gelenbe, E., Walrand, J. W. (Editors)

Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. Modeling & Simulations on Microcomputers

Madiseti, V.K. (Editor)

San Diego, CA: SCS Press, 138 pp. 1990.

Editorship of Journals & Transactions

1. IEEE Design & Test of Computers

Special Issue: Reengineering Digital Systems

April – June 1999 (Vol 16, No 2)

Madiseti, V.K (Editor)

Los Alamitos: CA, IEEE Computer Society Press, 1999.

2. IEEE Design & Test of Computers

Special Issue: Rapid Prototyping of Digital Systems

Fall 1996 (Vol 13, No 3)

Madiseti, V., Richards, M. (Editors)

Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. IEEE Transactions on Circuits & Systems II

Associate Editor: 1993-1995.

4. International Journal in Computer Simulation

Associate Editor: 1990-1993

5. International Journal in VLSI Signal Processing

Editorial Board: 1995 - Present

Recent Issued US Patents

- [1] [10,503,927](#) [Method and system for securing cloud storage and databases from insider threats and optimizing performance](#), Issued Dec 10, 2019
- [2]. [10,460,283](#) [Smart contract optimization for multiparty service or product ordering system](#), Issued Oct 29, 2019
- [3]. [10,459,946](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued Oct 29, 2019

- [4]. [10,402,589](#) [Method and system for securing cloud storage and databases from insider threats and optimizing performance](#), Issued Sep 3, 2019
- [5]. [10,394,845](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), Issued Aug 27, 2019
- [6]. [10,289,631](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), Issued May 24, 2019
- [7]. [10,255,342](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued April 9, 2019
- [8]. [10,243,743](#) [Tokens or crypto currency using smart contracts and blockchains](#), Issued March 26, 2019
- [9]. [10,204,339](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), Issued February 12, 2019
- [10]. [10,204,148](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued Feb 12, 2019
- [11]. [10,121,143](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), Issued Nov 6, 2018
- [12]. [10,102,526](#) [Method and system for blockchain-based combined identity, ownership, integrity and custody management](#), October 16, 2018
- [13]. [10,102,265](#) [Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing](#), October 16, 2018
- [14]. [9,935,772](#) [Methods and systems for operating secure digital management aware applications](#), April 3, 2018
- [15]. [9,769,213](#) [Method and system for secure digital object management](#), Issued Sep 19, 2017

Refereed Journal Publications

1. Trends in the Electronic Control of Mine Hoists

*Madiseti, V. and Ramlu, M.,
IEEE Transactions on Industry Applications, Vol IA-22, No. 6,
November/December 1986. Pages 1105-1112*

2. Multilevel range/NEXT performance in digital subscriber loops

Brand, G.; Madiseti, V.; Messerschmitt, D.G.;
Communications, Speech and Vision, IEE Proceedings I [see also IEE

Proceedings-Communications] ,Volume: 136 , Issue: 2 , April 1989
Pages:169 – 174

3. Seismic migration algorithms on parallel computers

Madiseti, V.K.; Messerschmitt, D.G.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 39 , Issue: 7 , July 1991
Pages:1642 – 1654

4. Asynchronous algorithms for the parallel simulation of event-driven dynamical systems

Madiseti, V.K.; Walrand, J.C.; Messerschmitt, D.G.:
ACM Transactions on Modeling and Computer Simulation, v 1, n 3, July 1991,
Pages: 244-74

5. Synchronization mechanisms for distributed event-driven computation

Madiseti, V.K.; Hardaker, D.:
ACM Transactions on Modeling and Computer Simulation, v 2, n 1, Jan. 1992,
Pages: 12-51

6. Efficient VLSI Architectures for the Arithmetic Fourier Transform (AFT)

Kelley, B.T.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 1 , January
1993
Pages:365-378

7. The fast discrete Radon transform. I. Theory

Kelley, B.T.; Madiseti, V.K.;
Image Processing, IEEE Transactions on ,Volume: 2 , Issue: 3 , July 1993
Pages:382 – 400

8. The Georgia Tech digital signal multiprocessor

Barnwell, T.P., III; Madiseti, V.K.; McGrath, S.J.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 7 , July 1993
Pages:2471 – 2487

9. The MIMDIX Environment for Parallel Simulation

Madiseti, V.K.; Hardaker, D.; Fujimoto, R.M.:
Journal of Parallel and Distributed Computing, v18, no. 4, August 1993,
Pages: 473-83.

- 10. LMSGEN: a prototyping environment for programmable adaptive digital filters in VLSI**
Romdhane, M.S.B.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:33 – 42
- 11. Fixed-point co-design in DSP**
Egolf, T.W.; Famorzadeh, S.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:113 - 126
- 12. A fast spotlight-mode synthetic aperture radar imaging system**
Madiseti, V.K.;
Communications, IEEE Transactions on ,Volume: 42 , Issue: 234 , February-April 1994
Pages:873 – 876
- 13. Rapid prototyping on the Georgia Tech digital signal multiprocessor**
Curtis, B.A.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 3 , March 1994
Pages:649 – 662
- 14. Low-power signaling in asymmetric noisy channels via spectral shaping**
Sipitca, M.; Madiseti, V.K.;
Signal Processing Letters, IEEE, Volume: 1 , Issue: 8 , Aug 1994
Pages:117 – 118
- 15. A quantitative methodology for rapid prototyping and high-level synthesis of signal processing algorithms**
Madiseti, V.K.; Curtis, B.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 11 , Nov. 1994
Pages:3188 – 3208
- 16. Computer Simulation of Application-Specific Signal Processing Systems**
Casinovi, G.; Madiseti, V.K.;
International Journal in Computer Simulation, Vol. 4, No. 4, Nov 1994.
- 17. System partitioning of MCMs for low power**
Khan, S.A.; Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 12 , Issue: 1 , Spring 1995
Pages:41 – 52

18. Error correcting run-length limited codes for magnetic recording

Jaejin Lee; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 6 , Nov. 1995

Pages:3084 – 3086

19. Virtual prototyping of embedded microcontroller-based DSP systems

Madiseti, V.K.; Egolf, T.W.;

Micro, IEEE ,Volume: 15 , Issue: 5 , Oct. 1995

Pages:9 – 21

20. Constrained multitrack RLL codes for the storage channel

Lee, J.; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 3 , May 1995

Pages:2355 – 2364

21. Rapid digital system prototyping: current practice, future challenges

Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:12 – 22

22. Conceptual prototyping of scalable embedded DSP systems

Dung, L.-R.; Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:54 – 65

23. Advances in rapid prototyping of digital systems

Madiseti, V.K.; Richards, M.A.;

Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996

Pages:9

24. Combined modulation and error correction codes for storage channels

Jaejin Lee; Madiseti, V.K.;

Magnetics, IEEE Transactions on ,Volume: 32 , Issue: 2 , March 1996

Pages:509 – 514

25. Model-based architectural design and verification of scalable embedded DSP systems-a RASSP approach

Dung, L.-R.; Madiseti, V.K.; Hines, J.W.;

Chapter in VLSI Signal Processing, IX, 1996.

Pages:147 – 156

26. Low-power digital filter implementations using ternary coefficients

Hezar, R.; Madiseti, V.K.;

Chapter in VLSI Signal Processing, IX, 1996.,

Pages:179 – 188

27. All-digital oversampled front-end sensors

Romdhane, M.S.B.; Madiseti, V.K.;

Signal Processing Letters, IEEE, Volume: 3 , Issue: 2 , Feb. 1996

Pages:38 – 39

28. Modeling COTS components in VHDL

Calhoun, S., Reese, R; Egolf, T., Madiseti, V.K.;

Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996

Pages: 24 – 31

29. VHDL-Based Rapid Systems Prototyping

Egolf, T.; Madiseti, V.K.;

Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996

Pages: 40-52

30. Interface design for core-based systems

Madiseti, V.K.; Lan Shen;

Design & Test of Computers, IEEE ,Volume: 14 , Issue: 4 , Oct.-Dec. 1997

Pages:42 - 51

31. Incorporating cost modeling in embedded-system design

Debardelaben, J.A.; Madiseti, V.K.; Gadiant, A.J.;

Design & Test of Computers, IEEE ,Volume: 14 , Issue: 3 , July-Sept. 1997

Pages:24 – 35

32. On homomorphic deconvolution of bandpass signals

Marenco, A.L.; Madiseti, V.K.;

Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and

Signal Processing, IEEE Transactions on] ,Volume: 45 , Issue: 10 , Oct. 1997

Pages:2499 – 2514

**33. A case study in the development of multi-media educational material:
the VHDL interactive tutorial**

Gadiant, A.J.; Stinson, J.A., Jr.; Taylor, T.C.; Aylor, J.H.; Klenke, R.H.;

Salinas, M.H.; Madiseti, V.K.; Egolf, T.; Famorzadeh, S.; Karns, L.N.; Carter,

H.W.;

Education, IEEE Transactions on ,Volume: 40 , Issue: 4 , Nov. 1997

Pages:17 pp.

34. Adaptive mobility management in wireless networks

Jeongwook Kim; Madiseti, V.K.;

Electronics Letters ,Volume: 34 , Issue: 15 , 23 July 1998

Pages:1453 – 1455

35. Efficient implementation of two-band PR-QMF filterbanks

Hezar, R.; Madiseti, V.K.;

Signal Processing Letters, IEEE ,Volume: 5 , Issue: 4 , April 1998

Pages:92 – 94

36. On fast algorithms for computing the inverse modified discrete cosine transform

Yun-Hui Fan; Madiseti, V.K.; Mersereau, R.M.;

Signal Processing Letters, IEEE ,Volume: 6 , Issue: 3 , March 1999

Pages:61 – 64

37. System on chip or system on package?

Tummala, R.R.; Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:48 – 56

38. Reengineering legacy embedded systems

Madiseti, V.K.; Jung, Y.-K.; Khan, M.H.; Kim, J.; Finnessy, T.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:38 – 47

39. Reengineering digital systems

Madiseti, V.K.;

Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999

Pages:15 – 16

40. Parameter optimization of robust low-bit-rate video coders

Sangyoun Lee; Madiseti, V.K.;

Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 9

Issue: 6 , Sept. 1999

Pages:849 – 855

41. Closed-form for infinite sum in bandlimited CDMA

Jatunov, L.A.; Madiseti, V.K.;

Communications Letters, IEEE ,Volume: 8 , Issue: 3 , March 2004

Pages:138 – 140

42. A new protocol to enhance path reliability and load balancing in mobile ad hoc networks

Argyriou, A.; Madiseti, V.K.;

Journal of Ad Hoc Networks, Elsevier Press, 2004

- 43. Closed-form analysis of CDMA systems using Nyquist pulse**
Jatunov, L.A.; Madiseti, V. K.;
Communications Letters, IEEE (Under Revision), 2005.
- 44. Systematic Design of End-to-End Wireless Mobility Management Protocols,**
Argyriou, A.; Madiseti, V. K.;
ACM/Springer Wireless Networks (WINET), Accepted 2005.
- 45. A Novel End-to-End Approach for Video Streaming Over the Internet,**
Argyriou, A.; Madiseti, V. K.;
Kluwer Telecommunications Systems, Vol. 28, No. 2, Pages 133-150, Jan 2005. *Special Issue on Multimedia Streaming.*
- 46. An Analytical Framework of RD Optimized Video Streaming with TCP,**
Argyriou, A.; Madiseti, V. K.;
IEEE Transactions on Multimedia, Submitted for review in March 2005.
- 47. Modeling the Effect of Handoffs on Transport Protocol Performance,**
Argyriou, A.; Madiseti, V. K.;
IEEE Transactions on Mobile Computing, Submitted for review in March 2005
- 48. Throughput Models for Transport Protocols with CBR and VBR Traffic Workloads",**
Argyriou, A.; Madiseti, V. K.;
ACM Transactions on Multimedia Computing, Communications & Applications, Submitted for review in April 2005.
- 49. "Electronic System, Platform & Package Codesign",**
Madiseti, V. K.
IEEE Design & Test of Computers, Volume 23, Issue 3, June 2006. pages 220-233.
- 50. "The Design of an End-to-End Handoff Management Protocol",**
A. Argyriou, Madiseti, V. K.
Wireless Networks, Springer, May 2006.
- 51. "A Soft-Handoff Transport Protocol for Media Flows in Heterogeneous Mobile Networks ",**
A. Argyriou, Madiseti, V. K.
Computer Networks, Vol 50, Issue 11, Pages 1860-1871, August 2006.

- 52. "Computationally Efficient SNR Estimation for Bandlimited WCDMA Systems"**
L. Jatunov, Madiseti, V. K.
IEEE Transactions on Wireless Communications, Volume 5, Issue 13,
December 2006, Pages 3480-3491.
- 53. "Space-Time Codes for Wireless & Mobile Applications",**
M. Sinnokrot, Madiseti, V.K.
DSP Handbook, Second Edition, 2009 (to be published)
- 54.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems",** *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013
- 55.A. Bahga, V. Madiseti, "Cloud-Based Information Integration & Informatics Framework for Healthcare Applications",** *IEEE Computer*, February 2015.
- 56.A. Bahga, V. Madiseti, "A Cloud-based Approach for Interoperable EHRs",** *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 57.A. Bahga, V. Madiseti, "Cloud-Based Information Technology Framework for Data Driven Intelligent Transportation Systems",** *Journal of Transportation Technologies*, vol.3 no.2, April 2013
- 58.A. Bahga, V. Madiseti, "Performance Evaluation Approach for Multi-tier Cloud Applications",** *Journal of Software Engineering and Applications*, vol. 6, no. 2, pp. 74-83, Mar 2013.
- 59.Yusuf, A., V. Madiseti, "Configuration for Predicting Travel Time Using Wavelet Packets and Support Vector Regression",** *Journal of Transportation Technologies*, vol 3, no. 3, June 2013.
- 60.A. Bahga, V. Madiseti, "Analyzing Massive Machine Maintenance Data in a Computing Cloud",** *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1831 - 1843, 2012.
- 61.N. Radia. Y. Zhang, M. Tatimapula, V. Madiseti, "Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions,"** *Proceedings of the IEEE*, Vol 100, Issue 4, pp. 841-854, 2012.
- 62.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems",** *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013

- 63.**A. Bahga, V. Madiseti, "**A Cloud-based Approach for Interoperable EHRs**", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 64.**A. Bahga, V. Madiseti, "**Cloud-Based Information Integration & Informatics Framework for Healthcare Applications**", *IEEE Computer*, February 2015.

Peer Reviewed Conference Publications

- 1. Dynamically-reduced complexity implementation of echo cancelers**
Madiseti, V.; Messerschmitt, D.; Nordstrom, N.;
Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. ,Volume: 11 , Apr 1986
- 2. Seismic migration algorithms using the FFT approach on the NCUBE multiprocessor**
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988
International Conference on , 11-14 April 1988
- 3. Seismic migration algorithms on multiprocessors**
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988
International Conference on , 11-14 April 1988
Pages:2124 - 2127 vol.4
- 4. WOLF: A rollback algorithm for optimistic distributed simulation systems**
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Conference Proceedings, 1988 Winter , December 12-14, 1988
Pages:296 – 305
- 5. Efficient distributed simulation**
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Symposium, 1989. The 22nd Annual , March 28-31, 1989
Pages:5 - 6
- 6. High speed migration of multidimensional seismic data**
Kelley, B.; Madiseti, V.;

Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991
International Conference on , 14-17 April 1991
Pages:1117 - 1120 vol.2

7. Performance of a fast analog VLSI implementation of the DFT

Buchanan, B.; Madiseti, V.; Brooke, M.;
Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on
, 9-12 Aug. 1992
Pages:1353 - 1356 vol.2

8. Task scheduling in the Georgia Tech digital signal multiprocessor

Curtis, B.A.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on ,Volume: 5 , 23-26 March 1992
Pages:589 - 592 vol.5

9. The fast discrete Radon transform

Kelley, B.T.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on ,Volume: 3 , 23-26 March 1992
Pages:409 - 412 vol.3

10.Yield-based system partitioning strategies for MCM and ASEM design

Khan, S.; Madiseti, V.;
Multi-Chip Module Conference, 1994. MCMC-94, Proceedings., 1994 IEEE,15-
17 March 1994
Pages:144 - 149

11. Multitrack RLL codes for the storage channel with immunity to intertrack interference

Lee, J.; Madiseti, V.K.;
Global Telecommunications Conference, 1994. GLOBECOM '94.
'Communications: The Global Bridge', IEEE,Volume: 3 , 28 Nov.-2 Dec. 1994
Pages:1477 - 1481 vol.3

12. A parallel mapping of backpropagation algorithm for mesh signal processor

Khan, S.A.; Madiseti, V.K.;
Neural Networks for Signal Processing [1995] V. Proceedings of the 1995
IEEE Workshop , 31 Aug.-2 Sept. 1995
Pages:561 - 570

13. Virtual prototyping of embedded DSP systems

Madiseti, V.K.; Egolf, T.; Famorzadeh, S.; Dung, L.-R.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995

International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2711 - 2714 vol.4

14. Assessing and improving current practice in the design of application-specific signal processors

Shaw, G.A.; Anderson, J.C.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2707 - 2710 vol.4

15. Introduction to ARPA's RASSP initiative and education/facilitation program

Corley, J.H.; Madiseti, V.K.; Richards, M.A.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2695 - 2698 vol.4

16. DSP design education at Georgia Tech

Madiseti, V.K.; McClellan, J.H.; Barnwell, T.P., III;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 5 , 9-12 May 1995
Pages:2869 - 2872 vol.5

17. Rapid prototyping of DSP systems via system interface module generation

Famorzadeh, S.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1256 - 1259 vol. 2

18. Rapid prototyping of DSP chip-sets via functional reuse

Romdhane, M.S.B.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1236 - 1239 vol. 2

19. A constructive deconvolution procedure of bandpass signals by homomorphic analysis

Marenco, A.L.; Madiseti, V.K.;
Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International ,Volume: 3 , 27-31 May 1996
Pages:1592 - 1596 vol.3

20. BEEHIVE: an adaptive, distributed, embedded signal processing environment

Famorzadeh, S.; Madiseti, V.; Egolf, T.; Nguyen, T.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 1 , 21-24 April 1997
Pages:663 - 666 vol.1

21. Target detection from coregistered visual-thermal-range images

Perez-Jacome, J.E.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 4 , 21-24 April 1997
Pages:2741 - 2744 vol.4

22. Variable block size adaptive lapped transform-based image coding

Klausutis, T.J.; Madiseti, V.K.;
Image Processing, 1997. Proceedings., International Conference on ,Volume: 3 , 26-29 Oct. 1997
Pages:686 - 689 vol.3

23. A Rate 8/10 (0, 6) MTR Code And Its Encoder/decoder

Jaejin Lee; Madiseti, V.K.;
Magnetics Conference, 1997. Digests of INTERMAG '97., 1997 IEEE International , 1-4 April 1997
Pages:BS-15 - BS-15

24. VHDL models supporting a system-level design process: a RASSP approach

DeBardelaben, J.A.; Madiseti, V.K.; Gadiant, A.J.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:183 - 188

25. A performance modeling framework applied to real time infrared search and track processing

Pauer, E.K.; Pettigrew, M.N.; Myers, C.S.; Madiseti, V.K.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:33 - 42

26. System design and re-engineering through virtual prototyping: a temporal model-based approach

Khan, M.H.; Madiseti, V.K.;
Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on ,Volume: 2 , 1-4 Nov. 1998
Pages:1720 - 1724 vol.2

27. A debugger RTOS for embedded systems

Akgul, T.; Kuacharoen, P.; Mooney, V.J.; Madiseti, V.K.;
Euromicro Conference, 2001. Proceedings. 27th , 4-6 Sept. 2001
Pages:264 - 269

28. Adaptability, extensibility and flexibility in real-time operating systems

Kuacharoen, P.; Akgul, T.; Mooney, V.J.; Madiseti, V.K.;
Digital Systems, Design, 2001. Proceedings. Euromicro Symposium on , 4-6 Sept. 2001
Pages:400 - 405

29. Effect of handoff delay on the system performance of TDMA cellular systems

Turkboylari, M.; Madiseti, V.K.;
Mobile and Wireless Communications Network, 2002. 4th International Workshop on , 9-11 Sept. 2002
Pages:411 - 415

30. Enforcing interdependencies and executing transactions atomically over autonomous mobile data stores using SyD link technology

Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on , 19-22 May 2003
Pages:803 - 809

31. Performance evaluation and optimization of SCTP in wireless ad-hoc networks

Argyriou, A.; Madiseti, V.;
Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , 20-24 Oct. 2003
Pages:317 - 318

32. Implementation of a calendar application based on SyD coordination links

Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Parallel and Distributed Processing Symposium, 2003. Proceedings. International , 22-26 April 2003
Pages:8 pp.

33. Bandwidth aggregation with SCTP

Argyriou, A.; Madiseti, V.;
Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE Volume: 7 , 1-5 Dec. 2003
Pages:3716 - 3721 vol.7

34. Software streaming via block streaming

Kuacharoen, P.; Mooney, V.J.; Madisetti, V.K.;
Design, Automation and Test in Europe Conference and Exhibition, 2003
, 2003
Pages:912 – 917

35. Frequency-dependent space-interleaving for MIMO OFDM systems

Mohajerani, P.; Madisetti, V.K.;
Radio and Wireless Conference, 2003. RAWCON '03. Proceedings , Aug. 10-13, 2003
Pages:79 - 82

36. A media streaming protocol for heterogeneous wireless networks

Argyriou, A.; Madisetti, V.;
Computer Communications, 2003. CCW 2003. Proceedings. 2003 IEEE 18th Annual Workshop on , 20-21 Oct. 2003
Pages:30 – 33

37. Realizing load-balancing in ad-hoc networks with a transport layer protocol

Argyriou, A.; Madisetti, V.;
Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE ,Volume: 3 , 21-25 March 2004
Pages:1897 - 1902 Vol.3

38.Streaming H.264/AVC video over the Internet

Argyriou, A.; Madisetti, V.;
Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE , 5-8 Jan. 2004
Pages:169 – 174

Other Publications

1. **A Transport Layer Technology for Improving the QoS of Networked Multimedia Applications <draft-madisetti-arguriou-qos-sctp-00.txt>.**
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Jul 25, 2002.
2. **Voice & Video over Mobile IP Networks <draft-madisetti-argyriou-voice-video-mip-00.txt>**
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Nov 20, 2002.
3. **Enhancements to ECRTTP with Applications to Robust Header Compression for Wireless Applications. <draft-madisetti-rao-suresh-rohc-00.txt>**
Madisetti, V.; Rao, S., Suresh, N.
IETF Internet-Draft, June 30, 2003.

Ph.D. Students Graduated

1. **Brian T. Kelley, 1992**
VLSI Computing Architectures for High Speed Signal Processing
Member of Technical Staff, Motorola.

Winner of Dr. Thurgood Marshall Dissertation Fellowship Award
2. **Bryce A. Curtis, 1992**
Special Instruction Set Multiple Chip Computer for DSP
Member of Technical Staff, IBM
3. **Jaejin Lee, 1994**
Robust Multitrack Codes for the Magnetic Channel
Professor, Yonsei University, Korea
4. **Mohamed S. Ben Romdhane, 1995**
Design Synthesis of Application-Specific IC for DSP
Director of IP, Rockwell.
5. **Shoab A. Khan, 1995**
Logic and Algorithm Partitioning on MCMs
Professor, National University of Science & Technology, Pakistan
6. **Lan-Rong Dung, 1997**
VHDL-based Conceptual Prototyping of Embedded DSP Architectures
Professor, National Chaio Tung University, Taiwan.

Winner of VHDL International Best PhD Thesis Award, 1997

7. Thomas W. Egolf, 1997

Virtual Prototyping of Embedded DSP Systems

Distinguished Member of Technical Staff, Agere

8. Alvaro Marenco, 1997

On Homomorphic Deconvolution of Bandpass Signals

Professor, Texas A&M University.

Winner of GIT ECE Outstanding Teaching Assistant Award

9. Shahram Famorzadeh, 1997

BEEHIVE: A Distributed Environment for Adaptive Signal Processing

Member of Technical Staff, Rockwell.

10. Timothy J. Klausutis, 1997

Adaptive Lapped Transforms with Applications to Image Coding.

US Air Force/Univ. of Florida.

11. Lan Shen, 1998

Temporal Design of Core-Based Systems

Member of Technical Staff, IBM

12. James DeBardelaben, 1998

Optimization Based Approach to Cost Effective DSP Design

Research Scientist, Johns Hopkins University

Georgia Tech ECE Faculty Award

13. Sangyoun Lee, 1999

Design of Robust Video Signal Processors

Professor, Yonsei University

US Army Sensors Lab Research Excellence Award, 1999

14. Rahmi Hezar, 2000

Oversampled Digital Filters

Member of Technical Staff, Texas Instruments

15. Yong-kyu Jung, 2001

Model-Based Processor Synthesis

Professor, Texas A&M University

16. Mustafa Turkboylari, 2002

Handoff Algorithms for Wireless Applications

Member of Technical Staff, Texas Instruments

17. Yun-Hui Fan, 2002

A Stereo Audio Coder with Nearly Constant Signal to Noise Ratio

Post-Doctoral Research Associate, Northeastern University

18. Subrato K. De, 2002

Design of a Retargetable Compiler for DSP
Member of Technical Staff, Qualcomm

US Army Sensors Lab Research Excellence Award, 1999

19. Chonlameth Aripnikanondt, 2004

System-on-Chip Design with UML
Professor, King Mongkut's University, Thailand.

US Army Sensors Lab Research Excellence Award, 1999

20. Loran Jatunov, 2004

Performance Analysis of 3G CDMA Systems
Senior Research Scientist, Soft Networks, LLC.

21. Antonios Argyriou, 2005, Serving in Hellenic Army.

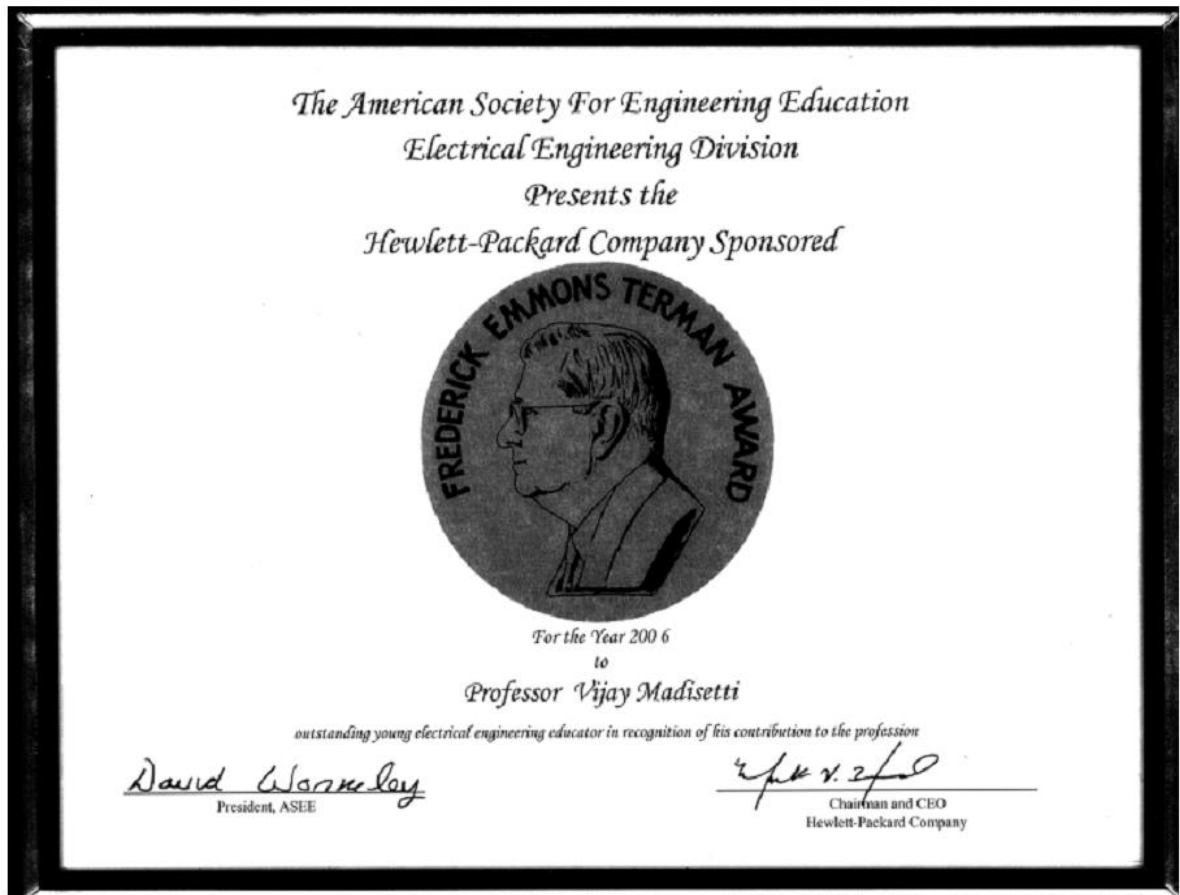
22. Pilho Kim, 2009, Scientist, VP Technologies, Inc.

23. M. Sinnokrot, 2009, Staff Engineer, Qualcomm.

Awards & Honors

1. **Jagadis Bose National Science Talent Fellowship**, Indian Institute of Technology, Kharagpur, 1980-1984.
2. **General Proficiency Prize**, Indian Institute of Technology, Kharagpur, 1984.
3. **Demetri Angelakos Outstanding Graduate Student Award**, Univ. of California, Berkeley, 1989
4. **Ira Kay IEEE/ACM Best Paper Award** for Best Paper presented at IEEE Annual Simulation Symposium, 1989
5. **IBM Faculty Development Award** 1990
6. **Technical Program Chair**, IEEE Workshop on Parallel and Distributed Simulation. 1990.
7. **Technical Program Chair**, IEEE MASCOTS'94
8. **NSF RI Award**, 1990
9. **VHDL International Best PhD Dissertation Advisor Award**, 1997

10. Georgia Tech Outstanding Doctoral Dissertation Advisor Award, 2001.
11. ASEE 2006 Frederick Emmons Terman Medal, 2006.
12. Fellow of IEEE



Intellectual Property Disclosures (Georgia Tech)

<u>Patent</u>	<u>Date</u>	<u>Description</u>
2843	2004	Method and Apparatus for Improving the Performance of Wireless LANs
2825	2003	Method and Apparatus for Optimal Partitioning and Ordering of Antennas for Layered Space-Time Block Codes in MIMO Communications Systems
2815	2003	How to Rapidly Develop a SyD Application
GSU-023	2003	Rapid Development of SyD Applications

2810	2003	System on Mobile Devices Middleware Design
2718	2003	A Transport Layer Algorithm for Improved Anycast Communication
2717	2003	A Novel Transport Layer Load-Balancing Algorithm
2716	2003	A Transport Layer QoS Algorithm
2715	2003	A Novel Transport Layer Algorithm for MPLS Performance
2659	2002	A New Algorithm and Technology for Implementing Mobile IP with Applications to Voice and Video over Mobile IP
2656	2002	Debugging with Instruction-Level Reverse Execution
2655	2002	Embedded Software Streaming
2539		System of Databases: An Enabling Technology for Programming
2517	2002	A Dynamic Instantiated Real-Time Operating System Debugger
2516	2002	A Dynamic Real-Time Operating System
GSU-009	2001	System of Databases: Architecture,, Global Queries, Triggers and Constraints
2480	2001	Mobile Fleet Application based on SyD Technology
2479	2001	System of Databases: A model with coordination links and a calendar application
1893	1999	Beehive
1726	1995	Very High Scale Integrated Circuit Hardware Description Language Models (VHDL Models)
1401	1995	Self-Compensation Receiver (SCR)

Issued Patents: 9,935,772, 9,769,213