

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Nuvoton Technology Corp.

Petitioner

v.

MicroChip Technology, Inc.

Patent Owner

U. S. Patent No. **7,930,576**

Filing Date: Dec. 18, 2007

Issue Date: Apr. 19, 2011

Title: Sharing Non-Sharable Devices Between an Embedded Controller and
a Processor in a Computer System

Inter Partes Review No. IPR2020-00394

Declaration of Michael C. Brogioli, Ph.D

Table of Contents

I.	Introduction.....	1
II.	Qualifications and Background.....	2
III.	Materials Considered.....	7
IV.	Legal Standards.....	8
	A. Claim Construction	9
	B. Obviousness Under U.S.C. § 103	10
V.	THE '576 PATENT.....	12
	A. Filing.....	12
	B. Background.....	12
	C. Challenged Claims	16
	1. Claim 17.....	16
	2. Claim 18.....	17
	3. Claim 20.....	17
	4. Claim 21.....	17
	5. Claim 188.....	17
	6. Claim 189.....	17
	7. Claim 213.....	17
VI.	Claim Construction of Terms for the '576 Patent.....	18
VII.	Level of Ordinary Skill	18
VIII.	Summary of Opinions on Invalidity.....	19
IX.	Ground 1: <i>Le</i> Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103).	20

A.	Claim 17.....	23
1.	<i>Le</i> teaches Element [P]: “a method for sharing a non-volatile memory between a processor and a microcontroller in a system”	23
2.	<i>Le</i> teaches Element [1] “in response to a power on reset, holding a system reset signal in a reset state, thereby prohibiting the processor from accessing the non-volatile memory”	23
3.	<i>Le</i> teaches or suggests Element [2]: “while the system reset signal is held in the reset state, fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller”	24
4.	<i>Le</i> teaches Element [3]:” after the program instructions or data have been loaded, permitting the processor in the system access to the non-volatile memory, and releasing the system reset signal from the reset state”	25
B.	Claim 18.....	26
C.	Claim 20.....	26
D.	Claim 21.....	27
E.	Claim 188.....	28
F.	Claim 189.....	29
G.	Claim 213.....	29
X.	Ground 2: <i>Le</i> in view of <i>Wang</i> Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103)	30
A.	Claim 17.....	31
1.	<i>Le</i> in View of Wang Renders Element [2] obvious.....	31
2.	<i>Le</i> teaches or suggests Element [3].....	33

3. Claims 18, 20, 21, 188, 189, and 213 33

XI. Ground 3: *Le* in View of *Triece* Renders Claims 17, 18, 20, 21, 188, 189, and 213 obvious (35 U.S.C. § 103). 33

A. Claim 17..... 34

1. *Le* in view of *Triece* renders Element [2] obvious 34

2. *Le* teaches or suggests Element [3]..... 35

B. Claim 18..... 35

C. Claims 20, 21, 188, 189, and 213 36

XII. Grounds 4-6: *Le* in View of *Lin*, *Le* in View of *Wang* and *Lin*, and/or *Le* in View of *Triece* and *Lin* Render Claim 213 Obvious (35 U.S.C. § 103). 36

XIII. Conclusion 40

I. Introduction

1. I, Dr. Michael Brogioli, submit this declaration to state my opinions on the matters described below.

2. I have been retained by Petitioner, Nuvoton Technology Corporation (“Nuvoton”), as an independent expert in this proceeding before the United States Patent and Trademark Office. Although I am being compensated at my usual rate of \$600 per hour for the time I spend on this matter, no part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding.

3. I understand that this proceeding involves U.S. Patent No. 7,930,576 (“the ’576 patent”). I have been asked to provide my opinions regarding to the patentability of the claims of the ’576 patent. I understand that the application for the ’576 patent was filed on December 18, 2007, as U.S. Application No. 11/958,601. I also understand that a copy of the ’576 patent is provided as Exhibit 1001 to Nuvoton’s petition for *inter partes* review.

4. I have been asked by Nuvoton to consider whether certain claims of the ’576 patent would have been obvious based on certain prior art references. I have also been asked to consider the state of the art and prior art as of April 10, 2007. Based on the prior art references I discuss below in this declaration, it is my opinion that one of ordinary skill in the art would have found claims 17, 18, 20, 21,

188, 189, and 213 of the '576 patent to have been obvious. My opinions are provided below.

II. Qualifications and Background

5. I am an Adjunct Professor of Electrical and Computer engineering at Rice University in Houston, Texas, and Managing Director of Polymathic Consulting in Austin, Texas. I received my Bachelor of Electrical Engineering from Rensselaer Polytechnic Institute in 1999, my Master of Science in Electrical and Computer Engineering from Rice University in 2003, and my Doctorate of Electrical and Computer Engineering from Rice University in 2007.

6. While at Rice University, I developed various computer architecture designs for embedded systems and microcontroller based SOC architectures and peripherals. For example, from 1999 to 2003, I worked in low-power computing, specifically focusing on dynamic power management of hardware components within various computer architectures. From 2002 to 2004, I developed Spinach, a simulator design toolset for modeling programmable network interface architectures, which models system components common to all programmable computing environments as well as components specific to embedded systems. From 2004 to 2009, I developed Spinach DSP-FPGA, a modular and composable simulator design infrastructure for programmable and reconfigurable embedded SOC architectures specifically targeting mobile, low-power, and embedded and

portable computing devices. From 2005 to 2009, I developed and published a retargetable compiler infrastructure and hardware design exploration toolkit for systems related to embedded computing technologies. Many of these tools have been used at United States universities in electrical and computer engineering research.

7. In the late 1990s, I was a hardware and software developer at Vicarious Visions in New York, developing third-party titles for Nintendo's handheld consoles, in addition to various serial communications-based computer peripherals. Specifically, I designed serial communications based peripherals to connect video game consoles to PC computers, allowing real time data transfer of graphics from a PC-based workstation to the console itself, as well as various hardware prototypes. During my career, I have served as Chief Technology Officer, often in co-founding roles.

8. From June 2006 to August 2007, I worked as the Technical Co-Founder of Method Seven LLC in Boston, MA, working with high-performance software and hardware systems architecture. I am currently a co-founder, co-inventor, and Chief Technology Officer of Network Native, an Internet of Things technology company.

9. I have held the position of Adjunct Professor at Rice University since 2009 and the position of Managing Director at Polymathic Consulting since 2011.

At Rice University, I instruct graduate-level curriculum in the areas of embedded computing, hardware and software systems. I also advise on university research and various design initiatives. At Polymathic Consulting, I work with a range of technologists from early stage start-ups to Fortune 500 companies on similar technologies.

10. From 2008 to 2009, I was Senior Engineer working in high-performance compiler designs and next-generation microprocessors and architectures at Freescale Semiconductor in Austin, TX. From November 2009 to October 2011, I was Chief Architect, Senior Member Technical Staff, at Freescale Semiconductor in Austin, TX (formerly Motorola), responsible for management of technology, engineering roadmaps, design lead on software infrastructure, and next-generation microprocessor architectures for embedded computing. During my tenure at Freescale Semiconductor, I was in charge of system developer tools for processor design, both in the hardware and software spaces. These included tools used for the programming of processors, simulation and design of processors, and related technologies.

11. I have worked for Texas Instruments' Advanced Architecture and Chip Technology division in Houston, Texas, in the areas of high-performance mobile and embedded systems design at the hardware and systems software level, specifically around heterogeneous computing, high-speed bus, and interconnect

technologies. I also have worked at Intel Corporation's Microprocessor Research Labs in the areas of computer architecture and compiler technologies.

12. I am recognized as an expert in computer architecture, computer hardware and computer software systems as they relate to the subject matter at hand. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and have been a Program Committee member and Chair for various tracks of the IEEE and ACM Design Automation Conference from 2011 to the present. I am also on the Technical Steering Committee of Design Automation Conference's Embedded Computing track.

13. Over the past fifteen-plus years, I have authored numerous peer-reviewed publications and engineering books on computer hardware and software design. Many of these incorporate technologies specific to the subject matter at hand, including embedded computing hardware and software systems, and related memory systems and optimization. These publications are disclosed in my curriculum vitae, attached as Exhibit A. My curriculum vitae contains more information on my background and experience, as well as the cases in which I have served as an expert witness the past four years.

14. I am a named inventor on multiple United States Patents, as well as various pending applications.

15. I have served as an engineering consultant and testifying witness on matters related to, and including, microcontrollers and related peripherals and interconnects.

16. During my time in industry and as a consultant, I have worked extensively on, and submitted opinions on, issues relating to the development and deployment of embedded computing and low-power and system management technologies. These include various aspects of embedded devices and related systems, including connectivity, microprocessor-related systems, and system software.

17. I am being compensated for services provided in this matter at my customary rate, plus travel expenses. My compensation is not conditioned on the conclusions I reach based on my analysis or the outcome of this matter. Similarly, my compensation is not dependent upon and in no way affects the substance of my statements in this declaration.

18. I have no financial interest in Petitioner or any of its subsidiaries. I also do not have any financial interest in Microchip Technology Inc. I do not have any financial interest in the '576 patent and have not had any contact with the named inventors of the '576 patent (Ian F. Harris and Drew J. Dutton).

19. Based on my educational and work experience, I believe I am well qualified to serve as a technical expert in this proceeding.

20. My curriculum vitae, which includes a more detailed summary of my background, experience, patents, and publications, is attached as Exhibit 1011 to Nuvoton's petition for *inter partes* review.

III. Materials Considered

21. In forming my opinions, I have reviewed the following documents:

Exhibit 1001: U.S. Patent No. 7,930,576

Exhibit 1003: Prosecution History for U.S. Patent No. 7,930,576

Exhibit 1004: Prosecution History for Reexamination No.
90/011,754

Exhibit 1005: Le, U.S. Patent No. 6,154,838

Exhibit 1006: Wang, U.S. Patent Application Publication No.
2004/0049727

Exhibit 1007: Triage, U.S. Patent Application Publication No.
2003/0056071

Exhibit 1008: Lin, U.S. Patent No. 5,423,019

Exhibit 1009: IEEE 100, The Authoritative Dictionary of IEEE
Standards Terms, seventh addition

Exhibit 1010: INTEL MCS 51[®] Microcontroller Family User's
Manual, Feb. 1994

IV. Legal Standards

22. In forming my opinions and considering the subject matter of the '576 patent and its claims in light of the prior art, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

23. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, after the claims are construed, the content of the prior art is compared to the construed claims.

24. I understand that a claimed invention is only patentable when it is new, useful, and non-obvious in light of the "prior art." That is, the invention, as defined by the claims of the patent, must not be anticipated by or rendered obvious by the prior art.

25. For purposes of this declaration, I have been asked to opine on certain issues regarding the technology at issue, the level of ordinary skill in the art, the scope of the '576 patent claims, and obviousness. I have been informed of the following legal standards, which I have applied in forming my opinions.

A. Claim Construction

26. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of one of ordinary skill in the art at the time of the alleged invention.

27. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by one of ordinary skill in the art in light of the specification and the prosecution history pertaining to the patent. I understand that if a special definition is given to a claim term by the inventor, then that definition governs. I have been informed that express representations made during prosecution have to be accounted for in determining the claim scope. I have been further informed that claim amendments made during prosecution in order to overcome a prior art rejection require a narrow reading of the amended claim language. I understand, however, that claim terms are generally not limited by the embodiments described in the specification.

28. I understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have

been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

B. Obviousness Under U.S.C. § 103

29. I have been advised by counsel that a patent claim may be unpatentable as obvious under 35 U.S.C. § 103 if the differences between the patented subject matter and the prior art are such that the patented subject matter as a whole would have been obvious to a person of ordinary skill in the art at the time the invention was made. I have also been advised that several factual inquiries underlie a determination of obviousness. These inquiries include (1) the scope and content of the prior art; (2) the level of ordinary skill in the field of the invention; (3) the differences between the claimed invention and the prior art; and (4) any objective evidence of non-obviousness.

30. I have also been advised that example reasons to combine or modify prior art references that may support a conclusion of obviousness include:

1. combining prior art elements according to known methods to yield predictable results;
2. simple substitution of one known element for another to obtain predictable results;
3. use of a known technique to improve similar techniques;

4. combining elements in a way that would be “obvious to try,”
particularly when there exists a design need or market pressure to
solve a problem and there are a finite number of identified, predictable
solutions such that a person of ordinary skill in the art would have had
good reason to pursue the known options within his or her technical
grasp;
5. design incentives or market forces that would prompt variations of
known work if those variations were predictable to a person of
ordinary skill in the art;
6. a teaching, suggestion, or motivation in the prior art to combine or
modify prior art references to arrive at the claimed subject matter; and
7. optimization of a recognized result-effective variable by a person of
ordinary skill in the art if that optimization would be routine.

31. I have been informed by counsel that there are also reasons that would prevent a person of ordinary skill in the art from modifying or combining prior art references. Examples of prior art references that a person of ordinary skill in the art would not combine or modify to achieve the claimed invention include: prior art references that teach away from one another; prior art references that teach away from the claimed invention; prior art references whose combination or modification would change the principle of operation of either prior art reference;

and prior art references whose combination or modification would render them inoperable or unsuitable for their intended purpose.

32. I have been informed that in determining whether a person of ordinary skill in the art would combine or modify prior art references, the entire contents of each prior art reference must be considered, including parts of those references that would suggest against the proposed combination or modification.

V. THE '576 PATENT

A. Filing

33. The application for the '576 patent was filed on December 18, 2007. Ex. 1001, cover page. The patent lists U.S. Provisional Application No. 60/910,863, filed on April 10, 2007, as a priority document. *Id.*

B. Background

34. The '576 patent relates to the field of computing devices and, more specifically, to sharing devices such as non-volatile memory. *Id.*, 1:17-20. Computer hardware systems typically included a main processor (CPU) and one or more microcontrollers for performing auxiliary functions like keyboard control. *Id.*, 1:22-25. The processor and microcontrollers would read firmware instructions from a non-volatile memory to boot up, reset, manage power, etc. *Id.*, 1:25-30. In one approach to provide the non-volatile memory, the processor and microcontroller “might arbitrate for access to a shared device that stores firmware

for the multiple processors.” Ex. 1004, p. 99 (Declaration of Applicant’s prior expert in reexamination); Ex. 1001, 1:35-37. According to Applicant’s prior expert, “designing an arbitration circuit adds its own cost and complexities to system design, and depending on the arbitration scheme that is chosen, arbitration may degrade overall system performance, for example by introducing additional delay in resolving access contention by multiple processors.” Ex. 1004, p. 99; Ex. 1001, 2:4-7. The prior declarant also asserts that the ’576 patent “overcomes the need for arbitration, and its associated costs, by providing the microcontroller with a system state ... in which it is assured safe access to the shared non-volatile memory.” Ex. 1004, p. 99-100. In one embodiment, the ’576 patent provides that state by having the microcontroller “hold a system reset signal in a reset state, thereby prohibiting a processor in the system from accessing a non-volatile memory in the system,” in “response to a power on reset” and while the microcontroller accesses the non-volatile memory. Ex. 1001, 4:10-46. Thus, the ’576 patent focuses on how access to a shared memory is controlled rather than how the microcontroller accesses the memory.

35. During original prosecution, there were four independent claims, which issued as claims 1 (a system), 6 (a method), 11 (a system), and 17 (a method). Ex. 1001, 16:2-19, 16:32-46, 16:59-17:8, 17:30-18:8. The claims fall into two groups:

- Group 1 includes claims 1-10. These claims require that (1) “in response to a change in system state to a first state wherein the microcontroller is assured safe access to the non-volatile memory, holding the system in the first state and switching access to the non-volatile memory from the processor to the microcontroller”; and (2) “while the system is held in the first state, fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller.” *Id.*, col. 16.
- Group 2 includes claims 11-23. These claims (1) do not require “switching access to the non-volatile memory from the processor to the microcontroller,” but (2) require that the “fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller” occur while a system reset signal is held in a reset state, and (3) require “prohibiting the processor from accessing the non-volatile memory.” *Id.*, cols. 16-18.

Neither group of claims requires sharing memory without arbitration circuitry.

36. During original prosecution, the examiner rejected both groups of claims as obvious over the same primary reference applied in this Petition, *Le*,

U.S. Patent No. 6,154,838, in view of *Pang*, U.S. Patent No. 6,925,522. Ex. 1003, p. 71. The examiner added *Pang*, believing that “Le fails to teach to load the program instructions/data into a memory of the microcontroller.” *Id.*, p. 72. This belief was erroneous. As detailed below, a person of ordinary skill in the art (POSITA) would have understood the architecture of the 8051 microcontroller *Le* uses to suggest that the claimed loading occurs when the microcontroller operates on instructions or data in an external memory. The examiner did not appear to have the benefit of this knowledge on how microcontrollers work with external memories when making the statement in the office action. Applicant argued that *Pang* fails to cure the deficiency in *Le* resulting from the examiner’s oversight. *Id.*, pp. 106-110.

37. For group 1, Applicant argued that *Le* and *Pang* also do not disclose the claimed method of access control (i.e., the switching recited in group 1). *Id.* But for group 2, Applicant did not dispute that *Le* discloses the claimed method of access control (i.e., “in response to a power on reset, holding a system reset signal in a reset state, thereby prohibiting the processor from accessing the non-volatile memory”). *Id.*

38. In response, the examiner allowed the claims. *Id.*, pp. 115, 118.

39. The claims were challenged in reexamination No. 90/011,754 based on art not in this Petition. Ex. 1004, p. 241. The reexamination confirmed the

claims and added new claims 24-213. Ex. 1004, p. 16. The claims were challenged in reexamination No. 90/011,754 based on different art than is in this petition. Ex. 1004, p. 241. The reexamination confirmed the claims added new claims 24-213. *Id.*, p. 16.

C. Challenged Claims

40. The challenged claims, claims 17, 18, 20, 21, 188, 189, and 213, come from set 2 (i.e., they do not require “switching”) and read as follows:

1. Claim 17

[P] 17. A method for sharing a non-volatile memory between a processor and a microcontroller in a system, comprising:

[1] in response to a power on reset, holding a system reset signal in a reset state, thereby prohibiting the processor from accessing the non-volatile memory;

[2] while the system reset signal is held in the reset state, fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller; and

[3] after the program instructions or data have been loaded, permitting the processor in the system access to the non-volatile memory, and releasing the system reset signal from the reset state.

2. Claim 18

18. The method of claim 17, wherein the program instructions or data comprise firmware code or data for the microcontroller.

3. Claim 20

20. The method of claim 17, wherein the microcontroller is coupled to the processor via a system reset bus, and wherein the microcontroller is operable to assert the system reset signal over the system reset bus.

4. Claim 21

21. The method of claim 17, wherein the microcontroller is coupled to the non-volatile memory via a non-volatile memory bus, and wherein the microcontroller is operable to fetch the program instructions or data from the non-volatile memory via the non-volatile memory bus.

5. Claim 188

188. The method of claim 17, wherein the microcontroller comprises a keyboard controller for managing of controlling communications between a keyboard and the processor.

6. Claim 189

189. The method of claim 188, wherein the non-volatile memory stores keyboard controller (KBC) code.

7. Claim 213

213. The method of claim 17, further comprising:

communicating management signals between a system interface and the microcontroller, wherein the system interface is coupled to the processor and is further coupled to the microcontroller via a system management bus, and wherein the microcontroller comprises a system reset line for communicating the system reset signal or other signal to or from the system interface.

VI. Claim Construction of Terms for the '576 Patent

41. I have been informed that in an *inter partes* review proceeding, claim terms should be construed under the same standard applied in federal district court litigation. I have been asked for purposes of this declaration to assume that the preambles of the independent claims are limiting. I have been asked to assume that the claims terms have their plain and ordinary meaning to a person skilled in the art in light of the specification and the prosecution history.

VII. Level of Ordinary Skill

42. A POSITA for the '576 patent would have had (1) a Bachelor of Science or equivalent degree from an accredited institution in electrical engineering, computer engineering, or a similar field; (2) a working knowledge of circuits; and (3) two years of experience in related microelectronics design and development. Additional education could have substituted for professional experience, and significant work experience could have substituted for formal education.

VIII. Summary of Opinions on Invalidity

43. In my expert report below, I have identified the following combinations of prior art that, in my opinion, render obvious the '576 patent claims:

Ground	Basis
1	<i>Le</i> renders claims 17, 18, 20, 21, 188, 189, and 213 obvious (35 U.S.C. § 103).
2	<i>Le</i> in view of <i>Wang</i> renders claims 17, 18, 20, 21, 188, 189, and 213 obvious (35 U.S.C. § 103).
3	<i>Le</i> in view of <i>Triece</i> renders claims 17, 18, 20, 21, 188, 189, and 213 obvious (35 U.S.C. § 103).
4	<i>Le</i> in view of <i>Lin</i> renders claim 213 obvious (35 U.S.C. § 103).
5	<i>Le</i> in view of <i>Wang</i> and <i>Lin</i> renders claim 213 obvious (35 U.S.C. § 103).

6	<i>Le</i> in view of <i>Triece</i> and <i>Lin</i> renders claim 213 obvious (35 U.S.C. § 103).
---	--

IX. Ground 1: *Le* Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103).

44. *Le* renders claims 17, 18, 20, 21, 188, 189, and 213 obvious because *Le* teaches or at least suggests every element of the claims.

45. I have been informed that *Le* is prior art under at least 35 U.S.C. §§ 102 (a), (b), and (e).

46. During examination, the examiner incorrectly characterized the operation of a microcontroller with regard to the claim language “loading the program instructions or data into the memory of the microcontroller.” A POSITA would have known that this would be common behavior for a microcontroller to perform as part of a checksum process. For example, the Intel MCS-51 Microcontroller documentation details how this exact behavior is performed by the device during general operation. The 8051 microcontroller is the original member of the MCS-51 family, and is the core for all MCS-51 devices. Ex. 1010, p. 1-3.

47. The Intel MCS-51 microcontroller contains an 8-bit CPU for executing instructions as well as 128 bytes of on-chip RAM. Ex. 1010, p. 1-3.

Figure 1 from the below depicts the block level layout of the 8051 core that is part

of the Intel MCS-51 microcontroller. Of note are the 8051 CPU, 128 bytes of RAM, and I/O ports which are coupled via the internal bus. *Id.*

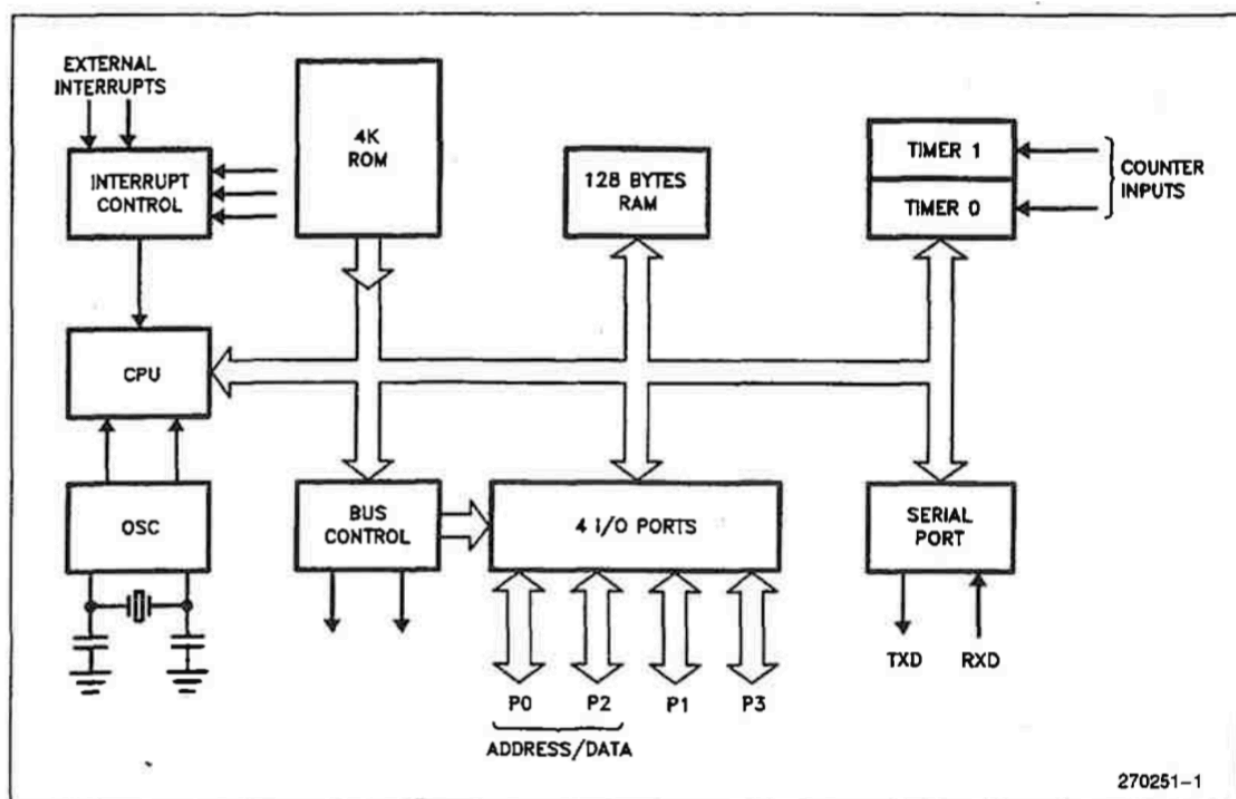


Figure 1, Ex. 1010, p. 1-3

48. The 128 bytes of RAM are comprised of a stack pointer, 4 banks of 8 registers denoted as R0-R7, and a possible bit-addressable space (bit addresses 0-7F) as depicted in Figure 7 below. Ex. 1010, 1-8. Note that all registers in Figure 7 below are part of, and comprised of RAM (memory) storage. *Id.*

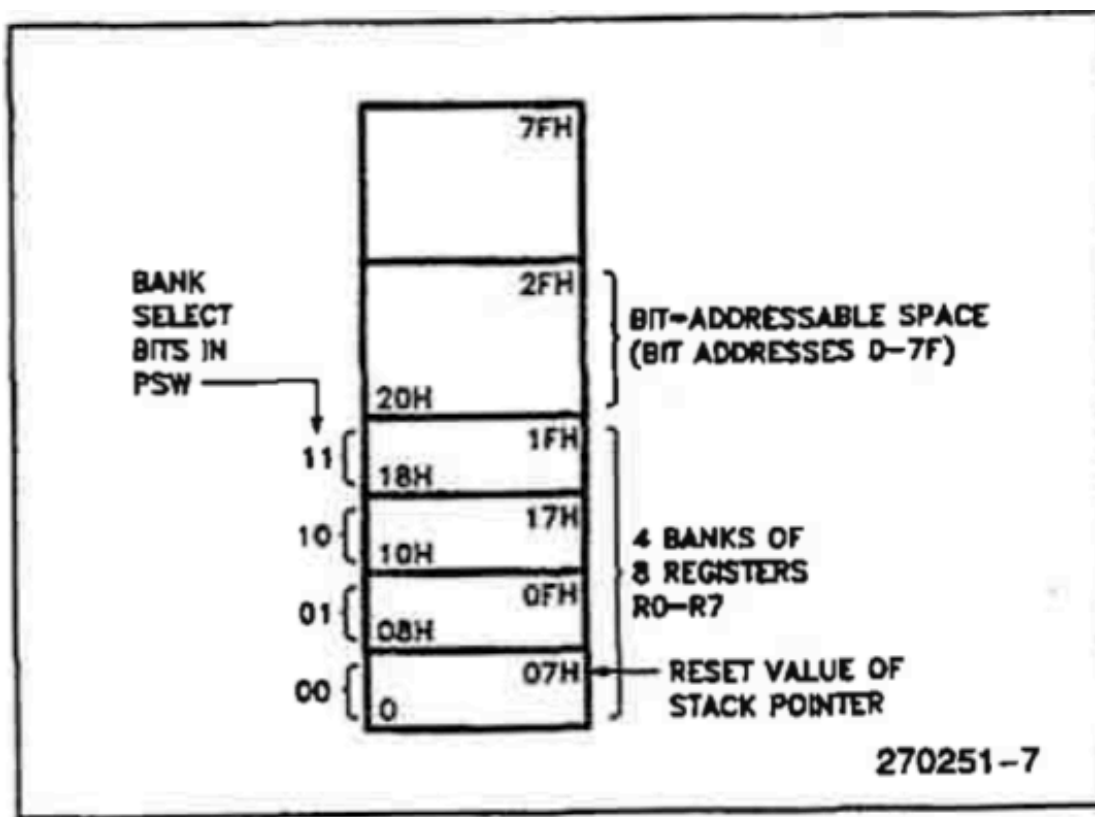


Figure 7, Ex. 1010, p. 1-8.

49. During operation, the MSC-51 would use one of the addressing modes of the 8051 core to perform the step of “fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller.” The fetched instructions would be loaded from external non-volatile memory, and stored within one of the 128 bytes of MSC-51 RAM (memory) as shown in Figure 7 above. *Id.* A checksum would then be performed on each of the fetched program instructions or data via logic operations (e.g., ANL, ORL, XRL) within the MSC-51’s 8051 processor instruction set such as ADD operations. *Id.*, pp. 2-22, 2-61, 2-63, 2-73. The program instructions and

data fetched from the non-volatile memory are ultimately transferred from the I/O ports, into the local 128 bytes of the MSC-8051 RAM (memory), and upon execution of logic operations during checksum, are also placed in various local registers of the 8051 CPU, such as the A (accumulator) register which also comprise RAM (memory).

A. Claim 17

- 1. *Le* teaches Element [P]: “a method for sharing a non-volatile memory between a processor and a microcontroller in a system”**

50. *Le* discloses a method for “sharing the flash ROM 122 [(the claimed “non-volatile memory”)] between a microcontroller and a microprocessor ...” *Le*, 9:29-32; *see also id.*, 1:16-20, 3:50-55, Fig. 3.

- 2. *Le* teaches Element [1] “in response to a power on reset, holding a system reset signal in a reset state, thereby prohibiting the processor from accessing the non-volatile memory”**

51. *Le* discloses that its arbiter assigns ownership of the flash ROM (non-volatile memory) to the microcontroller when power is first applied to the system (in response to a power on reset). *Id.*, 3:55-57. “[T]he microprocessor initially owns the flash ROM during the power-on reset and boot-up.” *Id.*, 2:54-56. In this state, the microcontroller controls the reset input of the CPU (holding a system reset signal in a reset state, prohibiting the processor from accessing the non-volatile memory). *Id.*, 12:8-14, 12:41-45, Fig. 3.

3. ***Le* teaches or suggests Element [2]: “while the system reset signal is held in the reset state, fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a memory of the microcontroller”**

52. *Le* discloses the microcontroller initially controls the reset line of the CPU. *Supra* at ¶ 51; *Le*, 14:9-10 (disclosing that the microcontroller “releases the system reset line in step 224 to allow the CPU to boot.”), Fig. 4. The microcontroller does not release the flash ROM to the microprocessor until “after the microcontroller boots up and checks the integrity of the flash ROM and updates the content of the flash ROM with valid firmware if necessary.” *Id.*, 3:57-62. In performing those tasks, a POSITA would have understood *Le* to teach that its microcontroller is fetching program instructions or data from the non-volatile memory and loading the program instructions or data into a microcontroller memory.

53. While the claim only requires data or program instructions, *Le* discloses both. *Le*, 3:14-17 (“all data stored in the flash ROM are accessible”), 3:17-23 (“flash ROM may store a number of optional data”), 3:58-59 (“updates the content of the flash ROM with valid firmware”), 6:43-456 (“flash ROM 122 contains the BIOS information for the computer system”), 15:18-19 (“[M]icrocontroller 174 executable code is located in the bottom 32K of the flash ROM 122”).

54. *Le* discloses that the microcontroller checks the integrity of the flash ROM just before releasing the CPU reset. *Id.*, 3:57-62. For example, *Le* discloses performing a checksum of flash ROM content. *Id.*, 12:19-25. A POSITA would have understood this to teach or suggest fetching the content of the flash ROM and loading it into the memory of the microcontroller so the microcontroller can perform the calculations necessary to generate the checksum. Even if the microcontroller only used registers to perform the checksum, the registers of an 8051 microcontroller (which is the microcontroller *Le* discloses) are still part of the 8051 memory. *Supra* at ¶¶ 47-49.

4. *Le* teaches Element [3]:” after the program instructions or data have been loaded, permitting the processor in the system access to the non-volatile memory, and releasing the system reset signal from the reset state”

55. *Le* discloses that “the microcontroller 174 releases that reset input of the CPU 100 [(“releasing the system reset signal from the reset state”)] to allow the CPU 100 to boot-up with ownership of the flash ROM 122 [(“permitting the processor in the system access to the non-volatile memory”)].” *Le*, 12:41-45. This release occurs after the microcontroller boots and performs a checksum of the flash ROM, which is after the program instructions or data have been loaded. *Supra* at ¶ 52.

B. Claim 18

56. Claim 18 depends from claim 17 and further requires that “the program instructions or data comprise firmware code or data for the microcontroller.” *Le* discloses that “microcontroller 174 executable code is located in the bottom 32K of the flash ROM 122.” *Id.*, 15:18-19.

C. Claim 20

57. Claim 20 depends from claim 17 and further requires that (1) “the microcontroller is coupled to the processor via a system reset bus,” and (2) “the microcontroller is operable to assert the system reset signal over the system reset bus.”

58. *Le* discloses the microcontroller releasing a “system reset line,” which “controls the reset input of the CPU,” to allow the CPU to boot. *Le*, 9:29-38, 12:41-45, 13:61-14:10, , Fig. 2. A POSITA would consider the “system reset line” to be a system reset bus because a single line that can carry a signal between two devices can be a bus. Ex. 1009, p. 128. A POSITA would have understood that the system reset line to allow for “coupling” of the microcontroller and the processor because it allows signals to pass between the two devices.

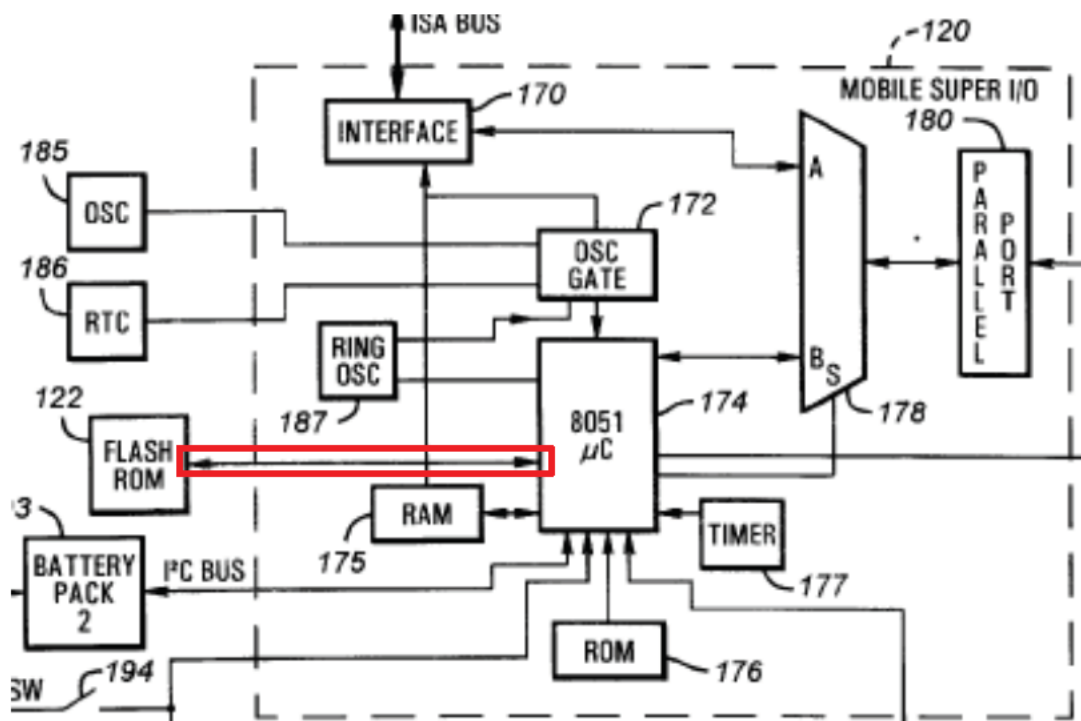
59. *Le* also teaches the releasing and control discussed above are performed by the microcontroller (*Le*, 9:29-38, 12:41-45, 13:61-14:10, Fig. 2), so it

teaches the microcontroller asserts the system reset signal over the system reset bus.

D. Claim 21

60. Claim 21 depends from claim 17 and further requires (1) “the microcontroller is coupled to the non-volatile memory via a non-volatile memory bus;” and (2) “the microcontroller is operable to fetch the program instructions or data from the non-volatile memory via the non-volatile memory bus.”

61. *Le* discloses that “the microcontroller 174 releases its ownership of the flash ROM 122 [(“non-volatile memory”)]... by tristating its signal lines coupled to the flash ROM 122 [(i.e., by tristating the non-volatile memory bus)].” *Le*, 12:37-40. The red box in the excerpt of Figure 2 (annotated below) shows a bus between the flash ROM 122 and the microcontroller 174. *Id.*, 9:29-42. Fig. 2. The microcontroller in *Le* is coupled to the non-volatile memory via a non-volatile memory bus.



62. *Le* also discloses a microcontroller that fetches program instructions or data from the flash ROM. A POSITA would have understood that the bus transfers data between the flash ROM and the microcontroller because it is the only path for transferring data. *Le* teaches that the microcontroller can fetch the program instructions or data from the non-volatile memory via the non-volatile memory bus.

E. Claim 188

63. Claim 188 depends from claim 17 and further requires that “the microcontroller comprises a keyboard controller for managing of controlling communications between a keyboard and the processor.” *Le* teaches that. *Id.*, 2:52-54, 10:18-21 (“The internal events which wake up the microcontroller 174 include

events relating to ... the keyboard”), 14:47-51 (“the keyboard and the mouse need to be disabled as the microcontroller 174 can no longer service the devices”), Fig. 2 (showing the keyboard connected through the microcontroller).

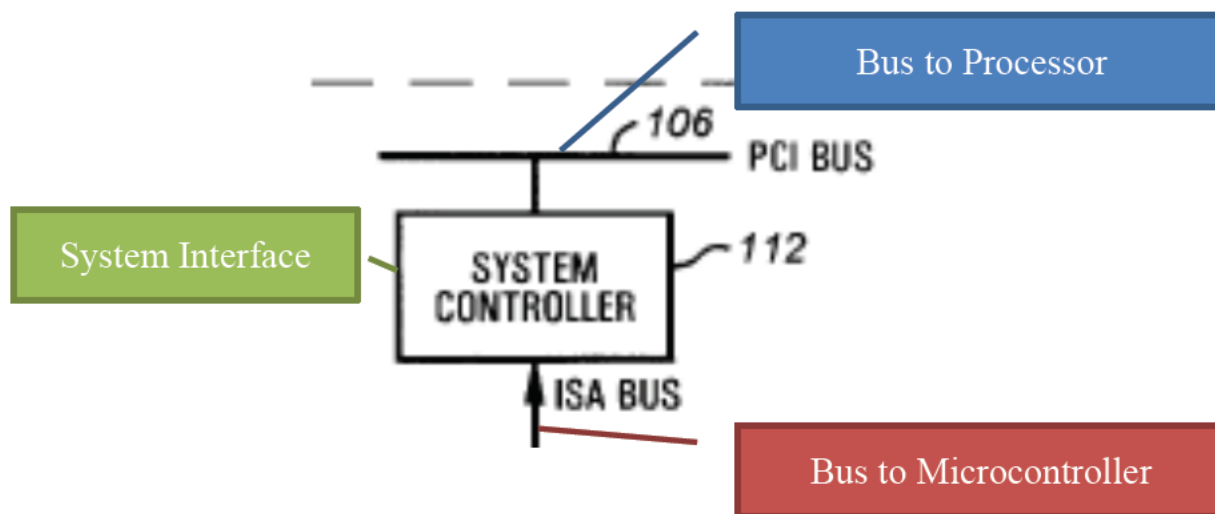
F. Claim 189

64. Claim 189 depends from claim 188 and further requires that “the non-volatile memory stores keyboard controller (KBC) code.” *Le* discloses the microcontroller firmware is stored in its flash ROM (the non-volatile memory) and discloses that the microcontroller may execute that firmware to provide “a keyboard controller.” *Le*, 2:52-54, 10:18-21 (“The internal events which wake up the microcontroller 174 include events relating to ... the keyboard”), 14:47-51 (“the keyboard and the mouse need to be disabled as the microcontroller 174 can no longer service the devices”), Fig. 2 (showing the keyboard connected through the microcontroller). A POSITA would have understood that the firmware in the non-volatile memory would include keyboard controller code.

G. Claim 213

65. Claim 213 depends from claim 17 and further requires “communicating management signals between a system interface and the microcontroller, wherein the system interface is coupled to the processor and is further coupled to the microcontroller via a system management bus, and wherein the microcontroller comprises a system reset line for communicating the system

reset signal or other signal to or from the system interface.” *Le* discloses a system controller 112 (a system interface). *Le*, 6:21-39. The signals between the microcontroller and the system controller are communicated via an ISA bus (a system management bus). *Id.*, 9:29-38. *Le* discloses the microcontroller releasing a “system reset line” to allow the CPU to boot. *Id.*, 9:29-38, 13:61-14:10, Fig. 2. *Le* also discloses the system controller (system interface) coupled to the processor via PCI bus 106. *Id.*, 9:29-38, Fig. 2 (see annotated excerpt below). A POSITA would have understood this disclosure to be the microcontroller with a system reset line for communicating the system reset signal or other signal to or from the system interface.



X. Ground 2: *Le* in view of *Wang* Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103)

66. *Wang*, U.S. Patent Application Publication No. 2004/0049727, was filed on December 31, 2002. I have been advised that *Wang* is prior art.

A. Claim 17

67. As discuss above, claim 17 has four elements. *Le* teaches Elements [P] and [1] in Ground 1 in Sections IX.A.1, *supra*. The combination of *Le* and *Wang* renders the remaining two elements obvious.

1. *Le* in View of Wang Renders Element [2] obvious

68. *Le* discloses performing a checksum of the contents of an external flash ROM while the system reset signal is held in the release state. *Supra* at IX.A; *Le*, 3:57-62, 12:19-25, 14:9-10. *Le*'s flash ROM contains data and program instructions. *Supra* at IX.A; *id.*, 3:14-23, 3:57-62, 6:43-46, 15:18-19. *Le* does not state that a checksum of the data and program instructions in the flash ROM includes fetching the data and program instruction in the flash ROM and loading them into a memory of the microcontroller for performing the checksum.

69. *Wang* teaches performing a checksum of an external flash ROM where the microprocessor fetches the contents of the flash ROM and loads them into the RAM of the processing device performing a checksum computation. *Wang*, ¶¶ 19-20 (“While reading data, the microprocessor 10 reads from the data blocks 41 of the flash ROM 40, and temporarily saves the data in the RAM 30. ... [T]he microprocessor 10 performs a cyclic redundancy check on the data that has been read. All bytes are added up to a sum.”), 22.

70. It would have been obvious to a POSITA to combine these teachings of *Le* and *Wang* because a combination of these known prior-art elements of calculating a checksum would yield a predictable result. A POSITA would have found it obvious to implement the *Wang* procedure in the *Le* microcontroller because in that combination, each element performs the same function as it does separately. A POSITA would have recognized that the results of the combination were predictable because *Le* expressly calls for the calculation of a checksum.

71. This combination is just the application of a known technique to a known method ready for improvement to yield predictable results. *Le* discloses a “base” method that calls out the need to calculate a checksum. *Wang* discloses a known technique for calculating a checksum applicable to the base method.

72. A POSITA would have recognized that applying the known technique would have yielded predictable results and resulted in an improved system for several independent reasons. The *Le* and *Wang* systems have similar hardware. *Le* did not require any specific method for calculating a checksum, and the 8051 microcontroller in *Le* can perform all of the functions in *Wang*’s checksum process. This combination would have involved the application of a known technique for calculating a checksum in a system that discloses that a checksum should be calculated to yield the predictable result of verifying the integrity of the flash ROM using a checksum.

2. *Le* teaches or suggests Element [3]

73. *See* Section IX.A.1.d. The release of the reset input by the microcontroller would occur after the checksum process described in Wang was complete.

3. Claims 18, 20, 21, 188, 189, and 213

74. *Le* teaches the limitations in dependent claims 18, 20, 21, 18, 20, 21, 188, 189, and 213 for the same reasons in Ground 1 above. *See* sections IX.B-G, *supra*.

XI. Ground 3: *Le* in View of *Triece* Renders Claims 17, 18, 20, 21, 188, 189, and 213 obvious (35 U.S.C. § 103).

75. *Triece*, U.S. Patent Application Publication No. 2003/0056071, was filed September 18, 2001. I have been advised that *Triece* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

76. For independent claim 17, *Triece* discloses a prior-art method of executing a program from an external memory by fetching and loading the program from the external memory into an internal memory. *Triece* and *Le* are analogous art because both relate to managing the relationship between a microcontroller and an external memory.

A. Claim 17

77. *Le* teaches Elements [P] and [1] for the reasons in Ground 1 in Sections IX.A.1-2. The combination of *Le* and *Triece* suggests the remaining two elements obvious.

1. *Le* in view of *Triece* renders Element [2] obvious

78. In *Le*, the microcontroller does not release the flash ROM to the microprocessor until “[a]fter the microcontroller boots up and checks the integrity of the flash ROM and updates the content of the flash ROM with valid firmware if necessary.” *Le*, 3:57-62. In this way, *Le* discloses booting the microcontroller while the system reset signal is held in the reset state. It would have been obvious to a POSITA to fetch and load program instructions from an external memory into the memory of the microcontroller as part of the boot-up process. Such importations were commonplace, routine, and conventional in multiple applications of microcontrollers prior to the priority date of the ’576 patent.

79. Booting microcontrollers by loading programs from an external memory (here the flash ROM of *Le*) into the internal RAM of the microcontroller was known. *Triece* discloses “a method for loading a boot memory onboard a microcontroller.” *Triece*, ¶ 11. The method includes “loading a program from an external program memory into the boot memory, configuring the boot memory to be addressed as a program memory, and executing the program in the boot

memory.” *Id.* The method may be implemented by fetching a word of the program from the external memory into a working register and then loading the word in the working register into the boot memory. *Id.*, ¶ 12. This provides the ability to perform in-system program modification (*id.*, ¶ 2) or having a microcontroller that does not require an onboard program ROM (*id.*, ¶¶ 3, 5). A POSITA would have understood that removing the onboard ROM in *Le* would reduce the overall part count and the cost of the system.

2. *Le* teaches or suggests Element [3]

80. *Le* discloses that “the microcontroller 174 releases that reset input of the CPU 100 [(“releasing the system reset signal from the reset state”)] to allow the CPU 100 to boot-up with ownership of the flash ROM 122 [(“permitting the processor in the system access to the non-volatile memory”)].” *Le*, 12:41-45. Second, this release occurs after the microcontroller boots and performs a checksum of the flash ROM after the program instructions or data have been loaded). *Supra* at IX.A. Thus, the release of the reset input by the microcontroller would occur after the booting process describe in *Triece* has been complete.

B. Claim 18

81. Claim 18 depends from claim 17 and further requires that the program instructions or data comprise firmware code or data for the microcontroller. *Le* discloses that “microcontroller 174 executable code is located in the bottom 32K of

the flash ROM 122.” *Le*, 15:18-19. In the combination of *Le* and *Triece* above, the flash ROM would also include the firmware from which the microcontroller boots. *Supra* at IX.B.

C. Claims 20, 21, 188, 189, and 213

82. *Le* in view of *Triece* renders claims 20, 21, 188, 189, and 213 obvious. *Le* discloses the limitations of those claims as set forth in Ground 1 above.

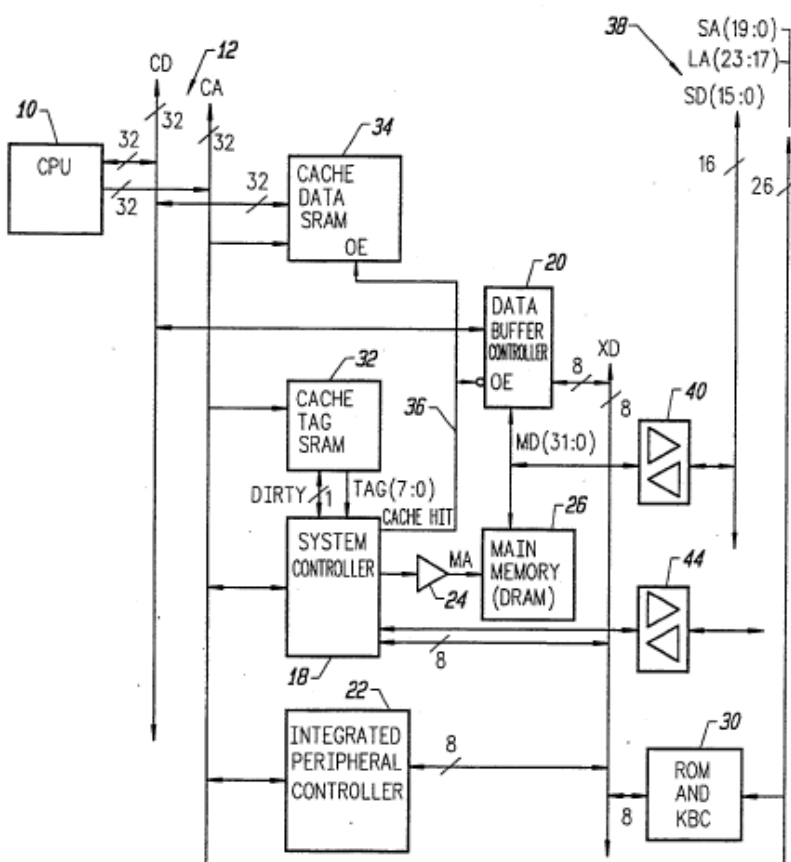
XII. Grounds 4-6: *Le* in View of *Lin*, *Le* in View of *Wang* and *Lin*, and/or *Le* in View of *Triece* and *Lin* Render Claim 213 Obvious (35 U.S.C. § 103).

83. *Le* in view of *Lin*, *Le* in view of *Wang* and *Lin*, and/or *Le* in view of *Triece* and *Lin* renders claim 213 obvious. As discussed above, *Le* (Ground 1), *Le* in view of *Wang* (Ground 2), and *Le* in view of *Triece* (Ground 3) each render claim 17, from which claim 213 depends, obvious. And *Le* also teaches or suggests the elements of claim 213. *Lin* further elaborates on the elements of claim 213 to the extent Patent Owner disputes that the elements of claim 213 are not taught or suggested by *Le*. *Le* and *Lin* are analogous art because they both relate to computer architectures that include a main processor and a microcontroller, such as a keyboard controller.

84. *Lin*, U.S. Patent No. 5,423,019, was filed on October 26, 1993. *Lin*, cover page. I have been advised that *Lin* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

85. *Lin* teaches or suggests “communicating management signals between a system interface and the microcontroller, wherein the system interface is coupled to the processor and is further coupled to the microcontroller via a system management bus, and wherein the microcontroller comprises a system reset line for communicating the system reset signal or other signal to or from the system interface,” as recited in claim 213. *Lin* discloses a keyboard controller (microcontroller), a system controller 18 (system interface), and a CPU 10 (processor). *Lin*, Fig. 2 (excerpted below), 7:27-56. The system controller (system interface) and the keyboard controller (microcontroller) are coupled via the XD bus (a system management bus). *Lin*, Fig. 1, 7:63-68 (“The system controller 18 is also connected bi-directionally to an 8-bit ROM and keyboard data bus XD which is also connected bi-directionally to the ... keyboard controller unit 30.”). The system controller 18 has an input pin to receive a “CPU Reset input from Keyboard Controller” (i.e., “a system reset line for communicating the system reset signal or other signal to or from the system interface and communicating management signals between a system interface and the microcontroller”) and an output pin for the CPU Reset. *Lin*, 10:60-11:8. Similarly, *Le* also discloses the microcontroller releasing a “system reset line” to allow the CPU to boot. *Le*, 13:61-14:10. The system controller 18 and the CPU 10 are connected via CPU bus 12. *Lin*, Fig. 1

(excerpted below), 7:27-56. That bus includes a reset signal line to place the CPU in reset. *Lin*, 9:42-45.



86. It would have been obvious to a POSITA to combine the teachings of *Le* and *Lin*. For example, a POSTISA would have understood that, because Wang disclosed a known architecture for communicatively coupling a keyboard controller to a processor, the combination of these known prior art elements according to the know methods would yield a predictable result. A POSITA could have combined the elements as claimed by known methods (implementing the procedure in *Le* in the architecture of *Lin*), and in that combination, each element

merely performs the same function as it does separately (i.e., *Le*, alone or in combination with *Wang* and *Triece*, still uses its method to share an external flash ROM with the processor and *Wang* still provides an architecture that includes a keyboard controller in communication with a CPU). A POSITA would have recognized that the results of the combination were predictable because *Le* expressly calls for communication of a reset signal between the microcontroller and the CPU.

87. This is nothing more than applying a known technique to a known method ready for improvement to yield predictable results. First, *Le* disclose a “base” method that specifically lays out the need for an improvement—i.e., the need to communicate a reset signal from a keyboard controller to a CPU. Second, *Lin* discloses a known technique for communicating a reset signal from a keyboard controller to a CPU that is applicable to the base method. And a POSITA skill would have recognized that applying the known technique would have yielded predictable results and resulted in an improved system where the keyboard controller and CPU could communicate as *Le* intended. Accordingly, this would be nothing more than applying a known technique for calculating a checksum in a system that discloses that a checksum should be calculated to yield the predicable result of providing a communication path between the microcontroller and the CPU.

XIII. Conclusion

88. In signing this declaration, I understand that the declaration will be filed as evidence in a contested proceeding before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this proceeding and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the allotted time for cross examination.

89. I declare that all statements made herein of my knowledge are true, 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: January 7, 2020

By: Michael C. Brogioli

Dr. Michael C. Brogioli