

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 B2

*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

---

**PETITION FOR *INTER PARTES* REVIEW UNDER**

**35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100 *ET SEQ.***

## **TABLE OF CONTENTS**

<b>I.</b>	<b>PRELIMINARY STATEMENT .....</b>	<b>1</b>
<b>II.</b>	<b>THE '576 PATENT .....</b>	<b>1</b>
A.	Filing.....	1
B.	Background .....	1
C.	Challenged Claims .....	5
1.	Claim 17.....	5
2.	Claim 18.....	5
3.	Claim 20.....	6
4.	Claim 21 .....	6
5.	Claim 188.....	6
6.	Claim 189.....	6
7.	Claim 213.....	6
<b>III.</b>	<b>STATEMENT OF PRECISE RELIEF REQUESTED .....</b>	<b>7</b>
A.	Claims for Which Review Is Requested .....	7
B.	Prior Art Applied.....	7
C.	Statutory Grounds of Challenge.....	8
D.	Level of Ordinary Skill .....	9
E.	Claim Construction.....	9
<b>IV.</b>	<b>CLAIMS 17, 18, 20, 21, 188, 189, AND 213 ARE UNPATENTABLE .....</b>	<b>10</b>
A.	Ground 1: <i>Le</i> Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103).....	10
1.	Claim 17.....	13

a.	<i>Le</i> teaches Element [P]: “A method for sharing a non-volatile memory between a processor and a microcontroller in a system” .....	13
b.	<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” .....	13
c.	<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” .....	14
d.	<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” .....	15
2.	Claim 18 .....	16
3.	Claim 20 .....	16
4.	Claim 21 .....	17
5.	Claim 188 .....	18
6.	Claim 189 .....	19
7.	Claim 213 .....	19
B.	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) .....	21
1.	Claim 17 .....	21
a.	<i>Le</i> in view of <i>Wang</i> renders Element [2] obvious .....	21
b.	<i>Le</i> teaches or suggests Element [3] .....	23

2.	Claims 18, 20, 21, 188, 189, and 213 .....	23
C.	Ground 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).....	23
1.	Claim 17.....	24
a.	<i>Le</i> in view of <i>Triece</i> suggests Element [2] .....	24
a.	<i>Le</i> teaches or suggests Element [3] .....	25
2.	Claim 18.....	26
3.	Claims 20, 21, 188, 189, and 213 .....	26
D.	Grounds 4-6: <i>Le</i> in View of <i>Lin</i> , <i>Le</i> in View of <i>Wang</i> and <i>Lin</i> , and/or <i>Le</i> in View of <i>Triece</i> and <i>Lin</i> Render Claim 213 Obvious (35 U.S.C. § 103).....	26
<b>V.</b>	<b>MANDATORY NOTICES UNDER 37 C.F.R. § 42.8.....</b>	<b>30</b>
A.	Real Parties-In-Interest.....	30
B.	Related Matters.....	31
C.	Lead and Back-Up Counsel.....	31
D.	Service Information.....	31
<b>VI.</b>	<b>CERTIFICATION UNDER 37 C.F.R. § 42.24(D) .....</b>	<b>32</b>
<b>VII.</b>	<b>PAYMENT OF FEES .....</b>	<b>32</b>
<b>VIII.</b>	<b>TIME FOR FILING PETITION .....</b>	<b>32</b>
<b>IX.</b>	<b>GROUND FOR STANDING.....</b>	<b>32</b>
<b>X.</b>	<b>CONCLUSION .....</b>	<b>32</b>

**LIST OF EXHIBITS**

- Petition Exhibit 1001: U.S. Patent No. 7,930,576
- Petition Exhibit 1002: Declaration of Michael C. Brogioli, Ph.D.
- Petition Exhibit 1003: Prosecution History for U.S. Patent No. 7,930,576
- Petition Exhibit 1004: Prosecution History for Reexamination No. 90/011,754
- Petition Exhibit 1005: *Le*, U.S. Patent No. 6,154,838
- Petition Exhibit 1006: *Wang*, U.S. Patent Application Publication No. 2004/0049727
- Petition Exhibit 1007: *Triece*, U.S. Patent Application Publication No. 2003/0056071
- Petition Exhibit 1008: *Lin*, U.S. Patent No. 5,423,019
- Petition Exhibit 1009: IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms* (7th ed. 2000)
- Petition Exhibit 1010: INTEL MCS<sup>®</sup> 51 Microcontroller Family User's Manual, Feb. 1994
- Petition Exhibit 1011: Curriculum Vitae of Michael C. Brogioli, Ph.D.

## **I. PRELIMINARY STATEMENT**

This Petition, supported by the Declaration of Michael C. Brogioli, Ph.D. (Ex. 1002), explains why claims 17, 18, 20, 21, 188, 189, and 213 of U.S. Patent No. 7,930,576 (“the ’576 patent,” Ex. 1001) are unpatentable based on an oversight by the examiner and on new art that the U.S. Patent and Trademark Office (“PTO”) did not consider during prosecution. Nuvoton Technology Corporation (“Nuvoton” or “Petitioner”) asks the Board to grant this Petition and cancel the challenged claims.

## **II. THE ’576 PATENT**

### **A. Filing**

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

### **B. Background**

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 costs 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, pp. 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.

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.



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.

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.*

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

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. *Id.*, p. 16.

### **C. Challenged Claims**

The challenged claims, claims 17, 18, 20, 21, 188, 189, and 213, come from group 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.

### **III. STATEMENT OF PRECISE RELIEF REQUESTED**

#### **A. Claims for Which Review Is Requested**

Petitioner requests review of claims 17, 18, 20, 21, 188, 189, and 213 under 35 U.S.C. § 311 and their cancellation as unpatentable.

#### **B. Prior Art Applied**

This Petition applies four references: (1) *Le*, U.S. Patent No. 6,154,838 (Ex. 1005), the primary reference in all the asserted grounds, which teaches all or nearly all the elements of the challenged claims; (2) *Wang*, U.S. Patent Application Publication No. 2004/0049727 (Ex. 1006), which teaches how the contents of an external memory are fetched and loaded to perform a checksum; (3) *Triece*, U.S. Patent Application Publication No. 2003/0056071 (Ex. 1007), which teaches how to fetch and load instructions from an external memory when booting a microcontroller; and (4) *Lin*, U.S. Patent No. 5,423,019 (Ex. 1008), which clarifies certain features of the architecture between a keyboard controller and a central processing unit (CPU). Only *Le* was considered by the PTO. Both *Wang* and *Triece* independently teach the claim features that Applicant argued to distinguish the challenged claims from the art cited during original prosecution.

**C. Statutory Grounds of Challenge**

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).

**D. Level of Ordinary Skill**

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. Ex. 1002, ¶ 40.

**E. Claim Construction**

Claim construction in *inter partes* review is governed by the same standard as a district court. 37 C.F.R. § 42.100(b). Terms receive “their ordinary and accustomed meaning as understood by [a POSITA].” *Dow Chem. Co. v. Sumitomo Chem. Co.*, 257 F.3d 1364, 1372 (Fed. Cir. 2001); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (en banc). “The construction that stays true to the claim language and most naturally aligns with the patent’s description of the invention will be, in the end, the correct construction.” *Phillips*, 415 F.3d at 1316 (citation omitted).

Petitioner believes the Board should interpret all terms according to their ordinary and customary meaning.

**IV. CLAIMS 17, 18, 20, 21, 188, 189, AND 213 ARE UNPATENTABLE****A. Ground 1: *Le* Renders Claims 17, 18, 20, 21, 188, 189, and 213 Obvious (35 U.S.C. § 103)**

*Le*, U.S. Patent No. 6,154,838, renders claims 17, 18, 20, 21, 188, 189, and 213 obvious because *Le* teaches or at least suggests every element of the claims.

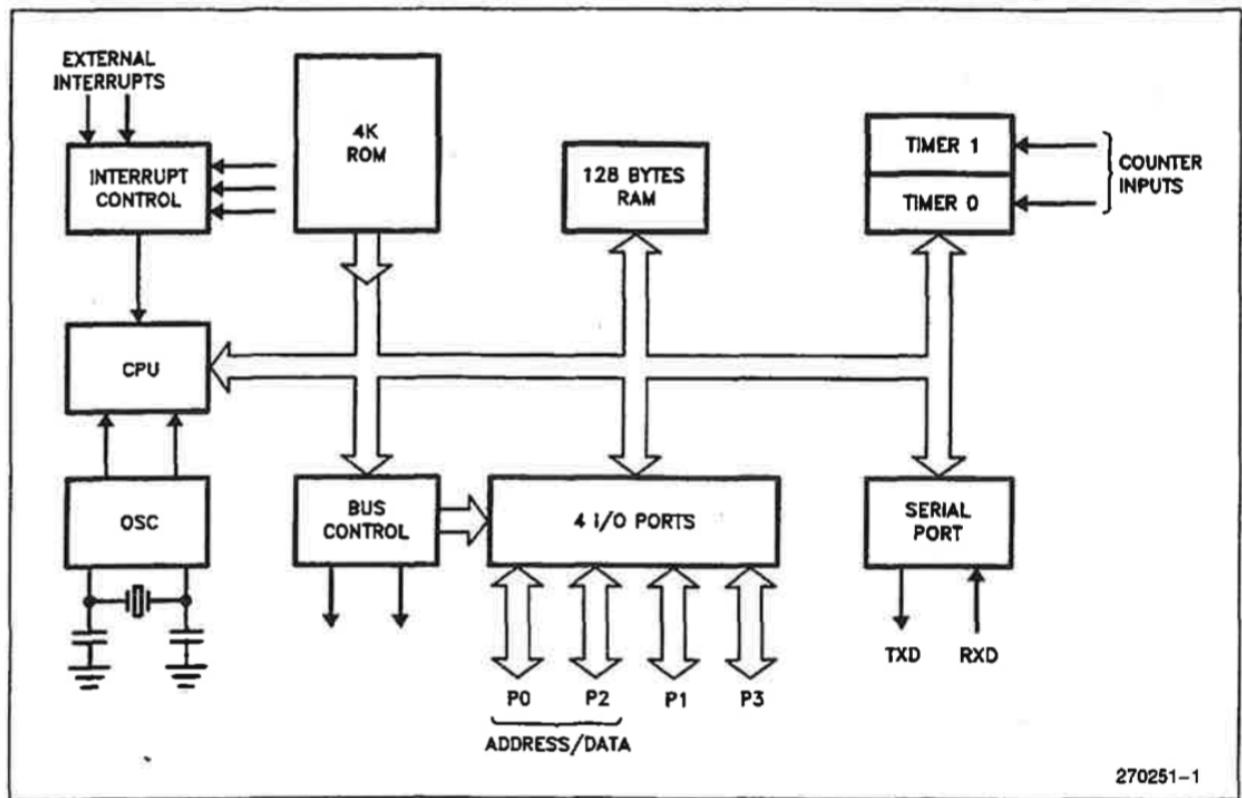
*Le* was filed on October 5, 1998 (eight years before the earliest claimed priority date of the '576 patent) and issued on November 28, 2000 (six years before the earliest claimed priority date of the '576 patent). *Le*, cover page. *Le* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

While the examiner applied *Le* during original prosecution, the examiner's allowance overlooked what a POSITA would have understood *Le* to disclose or suggest. The office action stated that "Le fails to teach to load the program instructions/data into a memory of the microcontroller." Ex. 1003, p. 72. But a POSITA would have understood that *Le* suggests this feature. Ex. 1002, ¶¶ 46-49. This is because *Le* discloses using an 8051 microcontroller. *Le*, 9:53-63; Ex. 1002, ¶¶ 46-49. As Dr. Brogioli explains, a POSITA would have understood that an 8051 microcontroller loads program instructions and data fetched from external memory into its memory when it operates on the data or executes the program instructions. Ex. 1002, ¶¶ 46-49. Dr. Brogioli supports his conclusion with the architecture of the 8051 microcontroller—where the registers storing instructions and data fetched

from external memory are part of the internal memory of the microcontroller. *Id.*

(citing Ex. 1010).

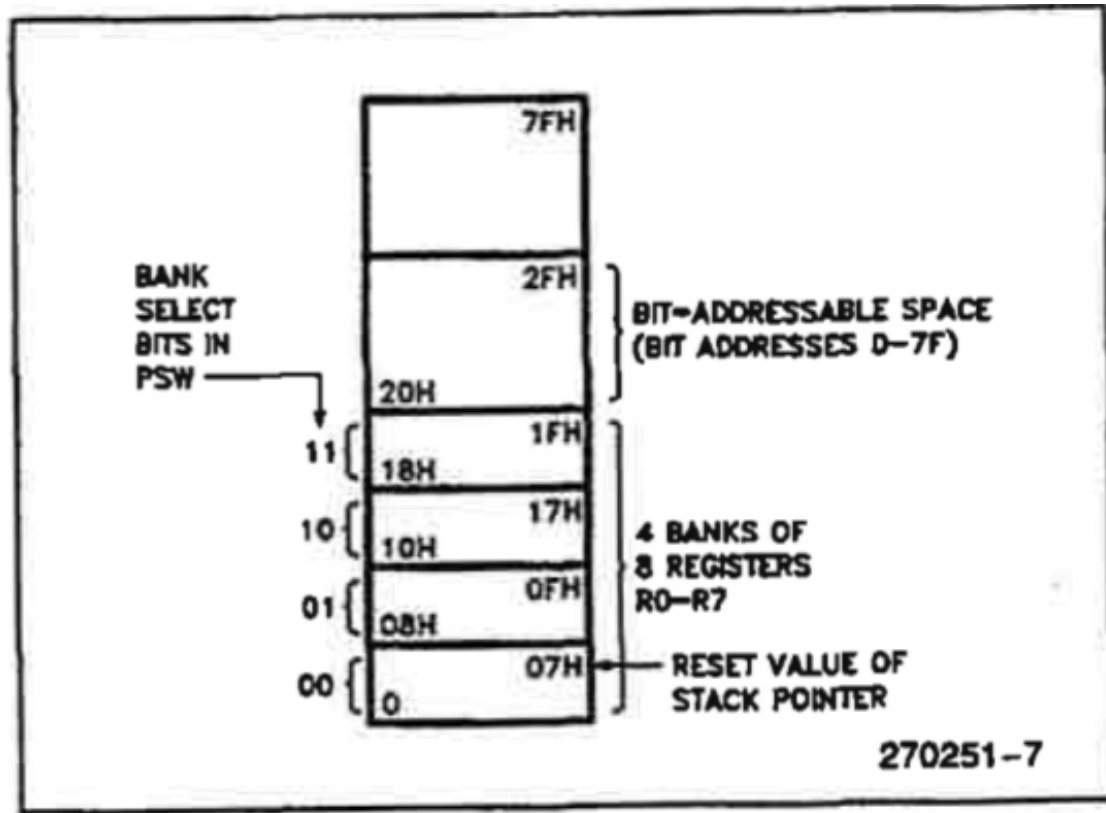
Dr. Brogioli explains that Intel MCS-51 microcontrollers contain an 8-bit CPU for executing instructions and 128 bytes of on-chip RAM. *Id.*, ¶ 47 (citing Ex. 1010, p. 1-3). He also provides the figure below with the block level layout of the 8051 microcontroller core. *Id.* (citing Ex. 1010, p. 1-3 (Fig. 1)). The figure shows the 8051 CPU, 128 bytes of RAM, and I/O ports coupled via the internal bus. *Id.*



The 128 bytes of RAM comprise 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 the



figure below. *Id.*, ¶ 48 (citing Ex. 1010, p. 1-8 (Fig. 7)). All registers are part of RAM (memory) storage. *Id.*



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.” *Id.*, ¶ 49. 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 the figure 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.* (citing Ex. 1010, 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 that also comprises RAM (memory). *Id.*

**1. Claim 17**

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

*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, Figs. 2-3; Ex. 1002, ¶ 50.

- b. *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”**

*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). *Le*, 3:55-57; Ex. 1002, ¶ 51. “[T]he microprocessor initially owns the flash ROM during the power-on reset and boot-up.” *Le*, 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; Ex. 1002, ¶ 49.

- c. ***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”**

*Le* discloses that the microcontroller initially controls the reset line of the CPU. *See supra* Section IV.A.1.b; *Le*, 14:9-10 (disclosing that the microcontroller “releases the system reset line in step 224 to allow the CPU to boot”), Fig. 4; Ex. 1002, ¶ 52. 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 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. Ex. 1002, ¶ 52.

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”), 3:57-62 (“updates the content of the flash ROM with valid firmware”), 6:43-46 (“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.”); Ex. 1002, ¶ 53.

*Le* discloses that the microcontroller checks the integrity of the flash ROM just before releasing the CPU reset. *Le*, 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. Ex. 1002, ¶ 54. 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.

- d. ***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”**

*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; Ex. 1002, ¶ 55. 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. *See supra* Section IV.A.1.b-c.

## 2. Claim 18

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.

## 3. Claim 20

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.”

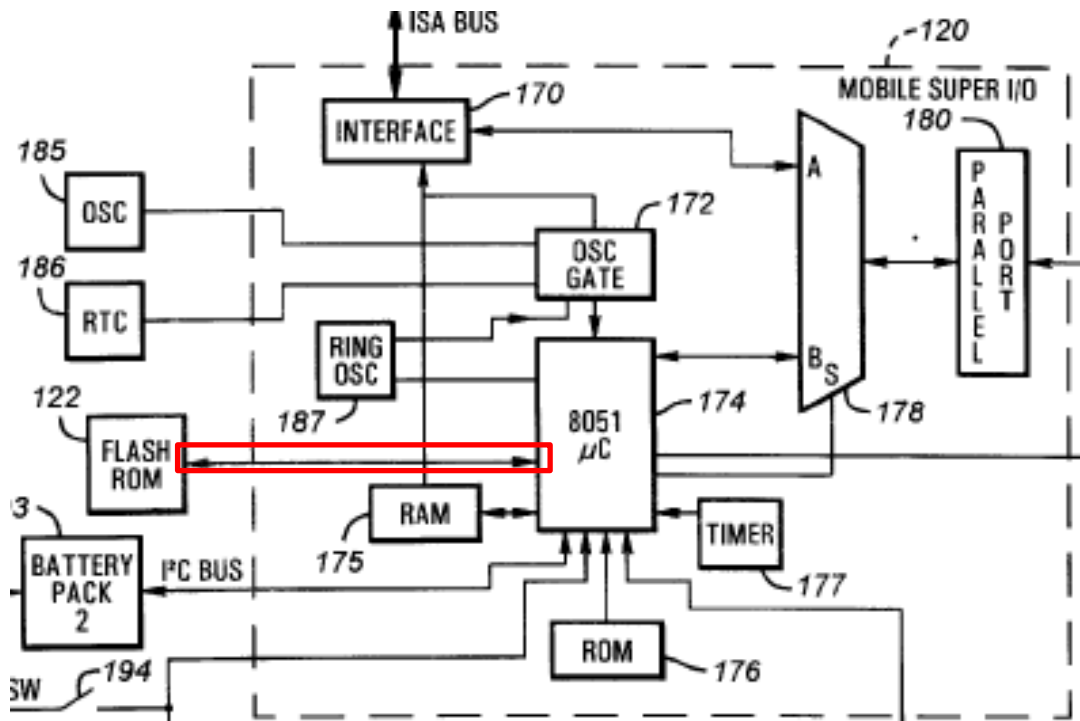
*Le* discloses the microcontroller releasing a “system reset line,” which “controls the reset input of the CPU,” to allow the CPU to boot. Ex. 1002, ¶ 58; *Le*, 9:29-38, 12:41-45, 13:61-14:10, Fig. 2. A POSITA would consider the “system reset line” to be a system reset bus because a single line that can carry a signal between two devices can be a bus. Ex. 1002, ¶ 58 (citing Ex. 1009, p. 128). A POSITA would have understood that the system reset line allows for coupling of the microcontroller and the processor because it allows signals to pass between the two devices. *Id.*

*Le* also teaches that 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 that the microcontroller asserts the system reset signal over the system reset bus. Ex. 1002, ¶ 59.

#### 4. Claim 21

Claim 21 depends from claim 17 and further requires that (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.”

*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. Ex. 1002, ¶ 60.



*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. *Id.*, ¶ 62. *Le* teaches that the microcontroller can fetch the program instructions or data from the non-volatile memory via the non-volatile memory bus.

## 5. Claim 188

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 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); Ex. 1002, ¶ 63.

#### **6. Claim 189**

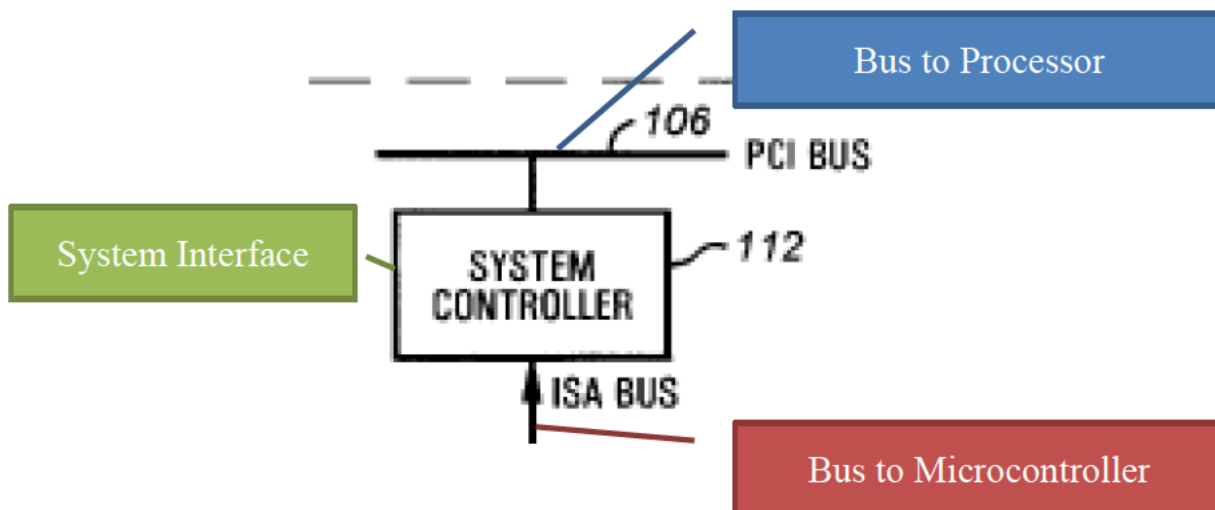
Claim 189 depends from claim 188 and further requires that “the non-volatile memory stores keyboard controller (KBC) code.” *Le* discloses that the microcontroller firmware is stored in its flash ROM (the non-volatile memory) and discloses that the microcontroller may execute that firmware to provide “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. Ex. 1002, ¶ 64.

#### **7. Claim 213**

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. Ex. 1002, ¶ 63.



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

*Wang*, U.S. Patent Application Publication No. 2004/0049727, was filed on December 31, 2002 (four years before the earliest claimed priority date of the '576 patent) and published on March 11, 2004 (three years before the earliest claimed priority date of the '576 patent). *Wang*, cover page. *Wang* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

**1. Claim 17**

As discuss above, claim 17 has four elements. *Le* teaches Elements [P] and [1] in Ground 1 in Section IV.A.1.a-b, *supra*. The combination of *Le* and *Wang* renders the remaining two elements obvious. Ex. 1002, ¶ 67.

**a. *Le* in view of *Wang* renders Element [2] obvious**

*Le* discloses performing a checksum of the contents of an external flash ROM while the system reset signal is held in the release state. *See supra* Section IV.A.1.c; *Le*, 3:57-62, 12:19-25, 14:9-10; Ex. 1002, ¶ 68. *Le*'s flash ROM contains data and program instructions. *See supra* Section IV.A.1.c; *Le*, 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.

*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; Ex. 1002, ¶ 69.

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. Ex. 1002, ¶ 70. A POSITA would have found it obvious to have implemented the *Wang* procedure in the *Le* microcontroller because in that combination, each element performs the same function as it does separately. *Id.* A POSITA would have recognized that the results of the combination were predictable because *Le* expressly calls for the calculation of a checksum. *Id.*

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

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. Ex. 1002, ¶ 72. 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 the functions in *Wang*'s checksum process. *Id.* 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. *Id.*

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

*See supra* Section IV.A.1.d. The release of the reset input by the microcontroller would occur after the checksum process described in *Wang* was complete. Ex. 1002, ¶ 73.

**2. Claims 18, 20, 21, 188, 189, and 213**

*Le* teaches the limitations in dependent claims 18, 20, 21, 188, 189, and 213 for the same reasons in Ground 1 above. *See supra* Section IV.A.2-7.

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

*Triece*, U.S. Patent Application Publication No. 2003/0056071, was filed September 18, 2001 (five years before the earliest claimed priority date of the '576 patent) and published on March 20, 2003 (four years before the earliest claimed

priority date of the '576 patent). *Triece*, cover page. *Triece* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

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.

### **1. Claim 17**

*Le* teaches Elements [P] and [1] for the same reasons set forth in Ground 1 in Section IV.A.1.a-b. The combination of *Le* and *Triece* suggests the remaining elements. Ex. 1002, ¶ 77.

#### **a. *Le* in view of *Triece* suggests Element [2]**

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. Ex. 1002, ¶ 78. 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. *Id.*

Such importations were conventional in multiple applications of microcontrollers prior to the priority date of the '576 patent. *Id.*

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). Ex. 1002, ¶ 79. A POSITA would have understood that removing the onboard ROM in *Le* would reduce the overall part count and the cost of the system. *Id.*

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

*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. *See supra* Section IV.A.1.d. Thus, the release of the reset input by the microcontroller would occur after the booting process in *Triece* has been completed. Ex. 1002, ¶ 80.

## 2. Claim 18

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. Ex. 1002, ¶ 81.

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

*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. Ex. 1002, ¶ 82.

### **D. 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)**

*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. *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.

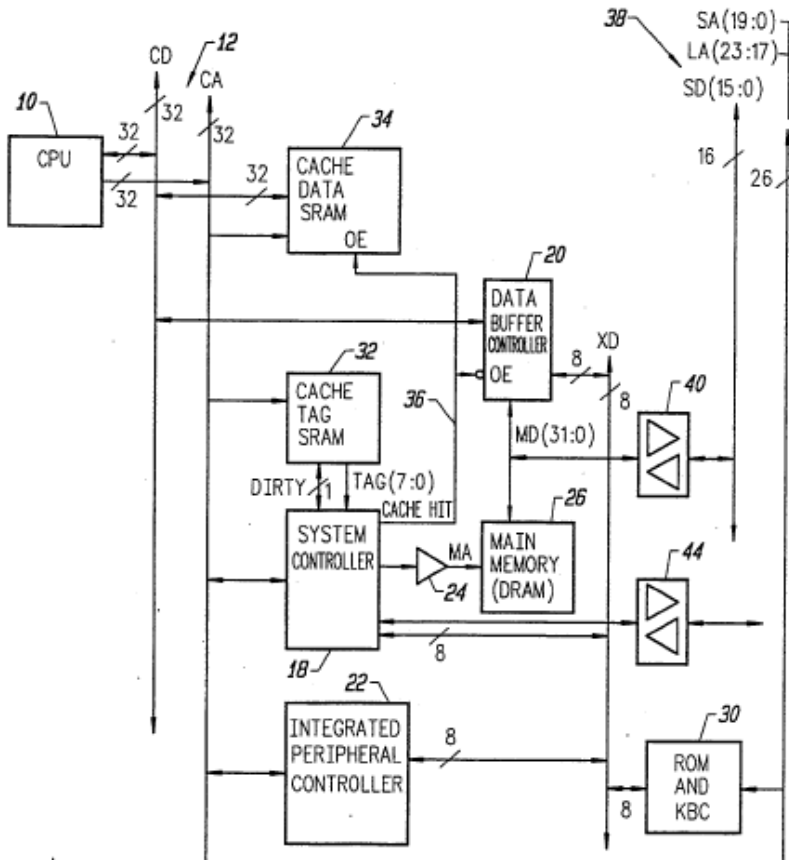
*Lin* further elaborates on the elements of claim 213 to the extent that Patent Owner disputes that the elements of claim 213 are not taught or suggested by *Le*. *Le* and *Lin* are analogous art because they both relate to computer architectures that include a main processor and a microcontroller, such as a keyboard controller.

*Lin*, U.S. Patent No. 5,423,019, was filed on October 26, 1993 (thirteen years before the earliest claimed priority date of the '576 patent) and issued on June 6, 1995 (eleven years before the earliest claimed priority date of the '576 patent). *Lin*, cover page. *Lin* is prior art under at least 35 U.S.C. § 102(a), (b), and (e).

*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. Ex. 1002, ¶ 85. *Lin* discloses a keyboard controller (i.e., microcontroller), a system controller 18 (i.e., system interface), and a CPU 10 (i.e., processor). *Lin*, Fig. 1 (excerpted below), 7:27-56. The system controller (i.e., system interface) and the keyboard controller (microcontroller) are coupled via the XD bus (a system management bus). *Id.*, 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. *Id.*, 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. *Id.*, 9:42-45.



It would have been obvious to a POSITA to combine the teachings of *Le* and *Lin*. Ex. 1002, ¶ 86. For example, a POSITA 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 known methods would yield a predictable result. *Id.* A POSITA would 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). *Id.* 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. *Id.*

This is applying a known technique to a known method ready for improvement to yield predictable results. *Id.*, ¶ 87. First, *Le* discloses 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. *Id.* 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. *Id.* And a POSITA 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. *Id.* 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 predictable result of providing a communication path between the microcontroller and the CPU. *Id.*

## **V. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(b)**

### **A. Real Parties-In-Interest**

The real parties-in-interest are Nuvoton Technology Corporation and Nuvoton Technology Corporation America.

**B. Related Matters**

On October 10, 2018, Patent Owner filed Case No. 1:18-cv-01560-MN (D. Del.), accusing Petitioner of infringing the '576 patent, but did not serve the complaint until January 7, 2019. On March 29, 2019, the case was transferred to the Northern District of California and assigned Case No. 3:19-cv-01690-SI (N.D. Cal.). Petitioner is aware of no other matters involving the '576 patent.

**C. Lead and Back-Up Counsel**

<b>Lead Counsel</b>	<b>Back-up Counsel</b>
E. Robert Yoches (Reg. No. 30,120) Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Telephone: (202) 408-4039 Email: bob.yoches@finnegan.com	Cory Bell (Reg. No. 75,096) Finnegan, Henderson, Farabow, Garrett & Dunner, LLP Two Seaport Lane Boston, MA 02210-2001 Telephone: (617) 646-1641 Email: cory.bell@finnegan.com  J. Preston Long (Reg. No. 65,125) Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Telephone: (202) 408-4347 Email: jp.long@finnegan.com

**D. Service Information**

Petitioner consents to service by e-mail at these addresses:

bob.yoches@finnegan.com, cory.bell@finnegan.com, jp.long@finnegan.com, and  
Nuvoton-PTAB@finnegan.com.

**VI. CERTIFICATION UNDER 37 C.F.R. § 42.24(D)**

As calculated by the “Word Count” feature of Microsoft Word 2016, this Petition contains 6,097 words, excluding: Table of Contents, Table of Authorities, List of Exhibits, Mandatory Notices Under 37 C.F.R. § 42.8, Certification Under § 42.24(d), and Certificate of Service.

**VII. PAYMENT OF FEES**

The Office may charge any additional fees to Deposit Account No. 06-0916.

**VIII. TIME FOR FILING PETITION**

This Petition is filed more than nine months after the ’576 patent was granted and not more than one year after Petitioner was served with a complaint alleging infringement of that patent. *See* 37 C.F.R. § 42.102(a)(1); 37 C.F.R. § 42.101(b).

**IX. GROUNDS FOR STANDING**

Under 37 C.F.R. § 42.104(a), Nuvoton certifies that the ’576 patent is available for *inter partes* review and Nuvoton is not barred or estopped from requesting *inter partes* review based on the grounds identified in this Petition.

**X. CONCLUSION**

For the reasons given in this Petition, claims 17, 18, 20, 21, 188, 189, and 213 are unpatentable. Petitioner asks that the Board institute trial on such claims and cancel them.

Respectfully submitted,

Dated: January 7, 2020

By: /E. Robert Yoches /

E. Robert Yoches, Reg. No. 30,120

## **CERTIFICATE OF SERVICE**

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, the foregoing **Petition for *Inter Partes* Review of 7,930,576, the associated Power of Attorney, and the associated Nuvoton Exhibits 1001-1011** were served on January 7, 2020, by Priority Mail Express at the following address of record for the '576 Patent:

### **Address of Record**

Attn: Mark Brightwell  
Kowert, Hood, Munyon, Rankin & Goetzel, P.C.  
P.O. Box 398  
Austin, Texas 78767

The undersigned further certifies that the foregoing Petition for *Inter Partes* Review, the associated Power of Attorney, and Nuvoton Exhibits 1001 through 1011 were served on January 7, 2020, by FedEx at the following addresses known to Petitioner as likely to effect service for the '576 Patent:

### **Addresses Likely to Effect Service**

Bruce W. Slayden III  
Slayden Grubert Beard PLLC  
401 Congress Avenue, Suite 1650  
Austin, Texas 78701

Travis Jensen  
Orrick, Herrington & Sutcliffe LLP  
1000 Marsh Road  
Menlo Park, California 94025

Dated: January 7, 2020

By: /Lauren K. Young/

Lauren K. Young

Litigation Legal Assistant

FINNEGAN, HENDERSON, FARABOW,

GARRETT & DUNNER, L.L.P.