

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

QUALCOMM INCORPORATED

Petitioner,

v.

COLLABO INNOVATIONS, INC.,

Patent Owner.

PTAB Case No. IPR2025-01015

U.S. Patent No. 7,930,575

Title: MICROCONTROLLER FOR CONTROLLING
POWER SHUTDOWN PROCESS

Issue Date: April 19, 2011

**DECLARATION OF DR. MURALI ANNAVARAM IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW UNDER 35 U.S.C. § 312**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	PROFESSIONAL BACKGROUD	1
	A. Overview.....	2
	B. Specific Experience with Microprocessor Design.....	7
III.	COMPENSATION	8
IV.	LEGAL CONSIDERATIONS	8
	A. Obviousness.....	8
	B. Claim Construction.....	11
V.	OVERVIEW OF TASK & BASIS FOR OPINIONS.....	12
VI.	PERSON OF ORDINARY SKILL IN THE ART.....	14
VII.	POWER MANAGEMENT TECHNOLOGY	15
	A. Techniques to Address Power Management Concerns were Known in the Art at the Time of the Invention of the '575 Patent.....	16
	B. Design Principles that a POSITA Would Be Aware of at the Time of the Invention of the '575 Patent	17
	1. Architectural State or Working Context of a Processor.....	17
	2. Clocks Are Indispensable in Digital Designs.....	22
	3. Voltage Scaling and Resumption of Clocks.....	24
	4. Use of Microprograms (or Microcode) Was Known in the Art .	27
	5. Benefits of higher integration on a semiconductor chip	30
	6. Benefits of integrating voltage regulator on SoC.....	33

7.	Benefits of multiple power supplies when multiple voltage domains are present	34
8.	Communications to and between processes/circuit block	35
VIII.	OVERVIEW OF THE '575 PATENT	37
A.	Background.....	37
B.	Prosecution History of the '575 Patent	41
C.	Litigation History of '575 Patent.....	41
D.	IPR Petitions Filed by AMD	42
1.	IPR2019-01523 (Jahagirdar IPR)	42
2.	IPR2019-01524 (Chung IPR)	42
IX.	CLAIM CONSTRUCTION.....	42
A.	Microcontroller.....	43
B.	Microprogram.....	43
C.	Power shutdown microprogram	43
D.	Power supply restoration microprogram.....	44
X.	OVERVIEW OF PRIOR ART	44
A.	U.S. 7,293,183 (Lee)	44
1.	Structure of the System-on-Chip (SOC)	46
2.	Steps Taken to Enter the Power-off Standby Mode.....	47
3.	Steps Taken to Exit the Power-off Standby Mode.....	50
B.	U.S. 7,219,248 (Shikata)	54
1.	Structure of the Semiconductor Integrated Circuit	55
2.	Steps Taken to Transition to the Standby State.....	57

3.	Steps Taken to Transition From the Standby State.....	62
4.	Timing Chart for Entering and Exiting the Standby State	65
C.	U.S. 7,664,970 (Jahagirdar).....	67
D.	U.S. 5,530,879 (Crump)	69
E.	U.S. 5,502,689 (Peterson)	71
F.	U.S. 7,369,815 (Kang).....	73
G.	U.S. 7,167,991 (Higashida).....	76
H.	U.S. 5,390,350 (Chung).....	78
XI.	SUMMARY OF OPINIONS.....	80
XII.	GROUND FOR CHALLENGING THE '575 PATENT	80
A.	Ground I: Lee and Jahagirdar render obvious Claims 1, 3, 5, 8, 11, 12, 15, and 16	80
1.	Motivation to Combine Lee and Jahagirdar.....	81
2.	Combination of Lee and Jahagirdar would be a routine exercise for a POSITA with a reasonable expectation of success	85
3.	Claim 1	87
4.	Claim 3	102
5.	Claim 5	104
6.	Claim 8	104
7.	Claim 11	110
8.	Claim 12	112
9.	Claim 15	117
10.	Claim 16	117

B.	Ground II: Lee, Jahagirdar and Peterson render obvious Claim 2	119
1.	Claim 2	119
2.	Motivation to combine Lee and Jahagirdar with Peterson.....	120
C.	Ground III: Lee, Jahagirdar and Kang render obvious Claim 3	122
1.	Claim 3	123
2.	Motivation to combine Lee and Jahagirdar with Kang.....	124
D.	Ground IV: Lee, Jahagirdar and Higashida render obvious Claim 11.....	126
1.	Claim 11	126
E.	Ground V: Shikata and Crump render obvious claims 1, 5, 11, 12, and 15.....	130
1.	Motivation to Combine Shikata and Crump	130
2.	Combination would be routine for a POSITA with a reasonable expectation of success.....	132
3.	Claim 1	134
4.	Claim 5	148
5.	Claim 11	148
6.	Claim 12.....	150
7.	Claim 15.....	150
F.	Ground VI: Shikata, Crump and Peterson render obvious Claim 2	151
1.	Claim 2	151
2.	Motivation to combine Shikata and Crump with Peterson	152
G.	Ground VII: Shikata, Crump and Kang render obvious Claim 3	153
1.	Claim 3	153

2.	Motivation to combine Shikata and Crump with Kang	154
H.	Ground VIII: Shikata, Crump and Chung render obvious Claim 16.....	155
1.	Claim 16.....	155
2.	Motivation to combine Shikata and Crump with Chung	157
I.	Alleged Secondary Considerations	159
XIII.	CONCLUSION	159

I. INTRODUCTION

1. My name is Dr. Murali Annavaram, Ph.D. I have been retained by Covington & Burling LLP on behalf of Qualcomm Incorporated (“Petitioner”) as an independent expert consultant in this *inter partes* review (“IPR”) proceeding before the United States Patent and Trademark Office (“USPTO”) regarding U.S. Patent No. 7,930,575 (“the ’575 Patent”) assigned to Collabo Innovations, Inc. (“Patent Owner” or “Collabo”). I have been asked to provide my technical opinions and testimony about Claims 1, 2, 3, 5, 8, 11, 12, 15 and 16 of the ’575 Patent. I refer to those claims as the “Challenged Claims.” The full text of the Challenged Claims appears in **Appendix A** to my Declaration.

II. PROFESSIONAL BACKGROUND

2. All of my opinions stated in this Declaration are based on my own personal knowledge, professional judgment and review of various documents. In forming my opinions, I have relied on my knowledge and experience in designing, developing, researching, and teaching the technology referenced in this Declaration.

3. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current *curriculum vitae*, which details my education and professional and academic experience, is being submitted as **Ex. 1004**. The following provides a brief overview

of some of my experience that is relevant to the matters set forth in this Declaration.

A. Overview

4. I am an expert in the field of computer architecture. In particular, my research focuses on efficient power management of chip multiprocessors (CMPs) and graphics processing units (GPUs), power efficient heterogeneous CMP designs, system-level power and performance tradeoffs, and reliability of computing platforms, superconducting CPU designs and efficient machine learning systems.

5. I am currently the Lloyd Hunt Chair professor in the Ming-Hsieh Department of Electrical and Computer Engineering, with a joint appointment in the Thomas Lord Computer Science department at the University of Southern California. I held the Dean's Chair Professorship from 2018-2021. I held the position of Robert G. and Mary G. Lane Early Career Chair at USC from 2012 to 2017. I started as a faculty member at USC in 2007.

6. I am founding and inaugural director of the USC-Meta center for Research and Education in AI and Learning at USC from 2022. I am the architecture thrust leader for the NSF's Expeditions in Computing center at USC focused on energy efficiency CPUs using superconducting technologies.

7. I co-authored Parallel Computer Organization and Design, a widely used textbook for graduate-level computer architecture courses. I teach several

undergraduate and graduate computer architecture courses at USC, including topics such as GPUs, out of order processor design, power and energy efficient design of computer hardware, mobile and edge computing systems, and memory system design.

8. I received a Bachelor of Technology degree in Computer Science from the National Institute of Technology in Warangal, India, in 1993 and a Master of Science degree in Computer Science and Engineering from Colorado State University in 1996. I received my Ph.D. degree in Computer Science and Engineering from The University of Michigan in 2001. I worked as a Senior Research Scientist at Intel Corporation from 2001 to 2007, and as a Visiting Faculty Researcher at Nokia Research Center in 2007 before joining the faculty of the University of Southern California in August 2007. I was a visiting faculty scientist at Meta during 2020 and 2021. From 2022 to 2023, I held the Rukmini Gopalakrishnagar visiting chair professorship at the Indian Institute of Sciences, Bengaluru, India.

9. For my contributions to the computer architecture community, I have been elected a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) and a distinguished member of the Association of Computing Machinery (ACM).

10. My research on server energy efficiency was selected for the IEEE Micro TopPicks award for the most influential computer architecture paper of the year in 2013.

11. I have received numerous other awards, including, for example, the IBM Faculty Partnership award in 2008, the Best Paper Award at the 2009 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS) and the National Science Foundation CAREER award in 2010, Best Paper Award at the Federated Machine Learning workshop, and Best paper and Best Student paper finalist nominations at the Supercomputing 2019 conference.

12. I have had dozens of papers accepted on the topic of efficient computer architecture design at major conferences in the field of computer architecture, including the International Symposium on Computer Architecture (ISCA), the High Performance Computer Architecture (HPCA), and the Symposium on Microarchitecture (MICRO).

13. I have given keynote presentations at the Frontiers in Technology distinguished speaker series at the University of California, Merced, 2021 Annual KIXX forum distinguished speaker series at the King Fahd University, Swedish Multicore Symposium in 2018, and at the IEEE International On-Line Test Symposium in 2014.

14. For my research and publications in this field, I was inducted into the ACM Special Interest Group on Microarchitecture Hall of Fame in December 2015. I am only one of 47 researchers over the past 48 years to receive this honor. I was also

inducted into the IEEE Computer Society Technical Committee on Computer Architecture's High Performance Computer Architecture Hall of Fame. I am one of only 42 researchers over the past 22 years to receive this honor. Finally, I was inducted into the ACM International Symposium on Computer Architecture (ISCA) Hall of Fame in June 2017. I was one of only 81 researchers to have received this honor over the past 44 years at the time of award selection.

15. I served as the General Co-Chair for ISCA 2018 and the Technical Program Chair for HPCA 2021, International Conference on Supercomputing 2024, Architecture Track Program Chair for HiPC 2014. I have also served as a program committee member for several international conferences and symposiums, including ISCA, MICRO, HPCA, SIGMETRICS, DSN, HiPC, and ASPLOS.

16. I served as an associate editor for the technical journal of the Association for Computer Machinery Transactions on Design Automation of Electronic Systems (ACM TODAES), and I served as a journal reviewer for the ACM Transactions on Computer Systems (ACM TOCS), ACM Transactions on Embedded Computing Systems (ACM TECS), ACM Computing Surveys, ACM Transactions on Architecture and Code Optimization (ACM TACO), and IEEE Top Picks in Microarchitecture 2010.

17. I am an inventor or co-inventor of seven issued patents relating to power

and reliability of computing systems.

18. I have graduated 16 PhD students at USC, and 13 of them were solely advised by me. Of these, 6 are now in academia (both in the USA and abroad) as professors working on various computer architecture related problems.

19. My research group has published about 30 papers on the topic of CPUs and GPUs with power optimized designs at top-tier computer architecture conferences in the past 15 years, where the typical acceptance rates are less than 20%. In particular, my group published 3 papers at the flagship computer architecture conference ISCA during 2016, which was a record for the highest number of published papers by a single research group at that time. These papers focused on improving the power and performance tradeoffs for various power efficiency techniques applied to highly parallel and heterogeneous processors such as CMPs and GPUs.

20. My research group has designed and evaluated the energy efficient techniques for heterogeneous CMPs, including building fully functional prototypes that required substantial understanding of the interactions between microarchitecture, circuits, operating systems and compilers. For instance, my group's work on KnightShift energy proportional computing developed an extensive set of power management schemes at chip and system level for heterogeneous computing platforms in active low-power state. In my group we routinely work with various benchmarks,

including SPEC and ParSec, to measure the power and performance impacts of architecture enhancements to chip designs. We also use various profiling tools, such as PIN and Vtune, for measuring various chip and system level activities. We use detailed microarchitecture simulation tools, such as SimpleScalar and GPGPUsim, for power and performance evaluations

B. Specific Experience with Microprocessor Design

21. I worked for 7 years at Intel working on chip multiprocessors and designed a wide range of power efficient techniques, such as EPI Throttling. EPI Throttling later influenced the successful introduction of the TurboBoost technology to reduce the power consumption of multi-core processors when there is variable utilization of the resources.

22. I also designed and evaluated die-stacking microprocessor technology for integrating high performance processors with fast access memory, which is now the highest cited work on the topic of die-stacking computer architecture.

23. I designed application specific optimizations, such as database system oriented optimizations, for efficient microprocessor design.

24. More recently I designed and implemented application-specific checkpointing and restart systems at Meta for enabling large scale machine learning systems to train power efficiently after a failure and restart.

III. COMPENSATION

25. My work on this matter is being billed at my customary rate of \$650 an hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services. I have no pecuniary interest in the outcome of this proceeding. I understand I will be paid regardless of the outcome of any proceeding in which my work is used.

IV. LEGAL CONSIDERATIONS

26. The attorneys for Petitioner have explained to me the legal standards that apply in this case. My understanding of those standards is described below. I am not an attorney, and although I hold several patents, I do not have formal training in the law regarding patents. I have used my understanding of the following legal principles set forth in this section in reaching my opinions.

A. Obviousness

27. I understand that a claimed invention is obvious, and therefore not patentable, if the subject matter claimed would have been obvious to a person of ordinary skill in the art at the time of the invention of the '575 Patent (a "POSITA"), which I have been asked to treat as September 12, 2006.

28. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a

modification or combination was within the skill of a POSITA.

29. I understand that I must consider whether the claimed subject matter as a whole would have been obvious to a person of ordinary skill in the art at the time of the invention in light of the following:

- The scope and content of the prior art;
- Any differences between the prior art and the claim;
- The level of ordinary skill in the pertinent art; and
- Any evidence of secondary indicia of non-obviousness (if available).

30. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I also understand that to establish a finding of obviousness one must show that one of ordinary skill in the art would have had a motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

31. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in which each element performs the same function it was known to perform, and the arrangement yields expected results, (2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was “obvious to try” because of a design need or

market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify or combine prior art references to arrive at the claimed invention.

32. I understand that “combining” prior art references entails considering the combined teachings of those references and determining what those combined teachings would have suggested to a POSITA. I understand that, therefore, it is not necessary that the entire technology of a secondary prior art reference be capable of being physically incorporated into the technology of the prior art reference, but rather that the secondary prior art reference contains teachings that a POSITA would have combined with those of the primary prior art reference.

33. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of the invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged

invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field.

34. I understand that there must be a nexus—that is, a connection between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness. I am not aware of any allegations by the named inventors of the '575 Patent or any assignee of the '575 Patent that any secondary considerations tend to rebut the obviousness of the '575 Patent.

B. Claim Construction

35. I understand that, in the context of patent claims, “comprise” is a non-exclusive transition indicating that while the claimed invention must contain the specified steps or features, it may also include other, unenumerated steps and features. Therefore, I understand that a “comprising” claim may be obvious based on prior art that discloses the claimed features and steps, even if the prior art also discloses other unenumerated features or steps.

36. I understand that a claim term is interpreted according to its ordinary and customary meaning as a POSITA would have understood the term in light of the surrounding claim language, other claims, the specification, and the patent’s prosecution history, which are referred to as intrinsic evidence. I also understand that

prior art references cited in the patent's prosecution history are considered intrinsic evidence. I further understand that evidence outside the patent and its prosecution history (e.g., dictionaries and technical articles), may inform the context in which a POSITA would have understood the claims of a patent. I understand this ordinary and customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term. I understand that this is the same claim construction approach used in district court litigation.

37. I understand that intrinsic evidence includes the patentee's statements in prior IPRs, which is considered part of the patent's prosecution history. I understand that, as intrinsic evidence, those statements affect claim scope. In particular, I understand that when a party clearly and unambiguously argues for a certain claim scope during an IPR to differentiate its invention from prior art, those arguments impact claim scope. I further understand that, when the patentee makes those arguments to distinguish the prior art, and prevails because of them, they carry even greater weight.

V. OVERVIEW OF TASK & BASIS FOR OPINIONS

38. I have been asked to review the '575 Patent. I have been asked to provide opinions related to certain issues from the perspective of a person of ordinary skill, having knowledge of the relevant art, as of September 12, 2006, and—except where

otherwise noted—the opinions stated in this Declaration are from that perspective. The qualifications and abilities of such a person are described in **Section VI** below.

39. In preparing this Declaration, I have considered the '575 patent and its prosecution history as well as the general knowledge of those familiar with the field of technology relating to design and implementation of microprocessors and system-on-chips. Given the obvious advantages of power conserving from voltage and frequency control, I considered techniques for conserving power by controlling supply of voltage and voltage levels to various circuit components, enabling and disabling of clocks and controlling clock speeds for various circuit components, and saving and restoring of architectural contexts of such system using a variety of approaches in order to facilitate rapid recovery from standby states where such context may not be preserved.

40. My opinions are based on my education, training, and experience as well as items that I reviewed to prepare my opinions, including the documents in the **Appendix B** to the Declaration, and any other items that I reference in my below analysis.

41. I understand that other issues may arise that require further explanation, and I will provide that explanation if appropriate. As a result, I respectfully reserve the right to update and supplement this Declaration and the information and opinions provided herein

42. Where my Declaration cites a patent, I have cited to the column(s) and line(s) of the patent. Where my Declaration cites an exhibit that has paragraph numbers, I have cited to the paragraph number. Finally, where there is emphasis in a quotation, I have added that emphasis unless I indicate otherwise.

43. At the request for attorneys for the Petitioner, I have reviewed select portions of the Petition. I identify the reviewed portions below. I concur with the sections of the Petition that I reviewed.

VI. PERSON OF ORDINARY SKILL IN THE ART

44. I understand that several factors are considered in determining the level of ordinary skill in the art, including the educational level of active workers in the field, the types of problems encountered in the art, the nature of prior art solutions to those problems, prior art patents and publications, the activities of others, the sophistication of the technology involved, and the rapidity of innovations in the field.

45. A POSITA would have a B.S. degree in Computer Engineering, Electrical Engineering or an equivalent field along with 3 to 5 years of academic or industry experience in microprocessor architecture, design of computer systems, designs of computer chips, or comparable industry experience. A person with less education but greater pertinent and relevant industry experience would also meet the standard.

46. The background technology described in **Section VII** and the prior art references discussed in **Section XI** and **Section XII** demonstrate the skill level of a POSITA at the time of the invention of the '575 Patent. As these sections demonstrate, at that time, a POSITA would have been aware of the underlying principles of the design and implementation of microprocessors and systems-on-chip, techniques to conserve power consumption by microprocessor by controlling supply of voltage and clocks, various modes of operation of such systems, such as the standby mode where supply of power and clock is disabled, and techniques to save and restore context when transitioning into and out of such standby modes. In addition, a POSITA would be aware of importance of clocks in digital systems, sound design techniques in circuit design when enabling and disabling power supply and clock supply, and tradeoffs of various approaches that can be employed when saving and restoring context.

47. I was at least a person of ordinary skill in the art as of the invention date of the '575 Patent.

VII. POWER MANAGEMENT TECHNOLOGY

48. This section provides an overview of the trends in power management of an overview of the basic principles of circuit design at the time of and before the date of the invention of the '575 Patent.

A. Techniques to Address Power Management Concerns were Known in the Art at the Time of the Invention of the '575 Patent

49. Efficient power management has been a concern and area of study for desired improvement in computer systems and microarchitecture for decades. Power-management concerns are especially prevalent in battery-powered computing devices, such as laptops and mobile telephones. *See* Ex. 1043 (Reneris), 1:31–34. While the simplest and most absolute way to preserve power is to simply shut off a system in its entirety, shutting down a system leads to significant loss of performance perceived by end users.

50. Many less absolute methods of power management have been developed to enable users to obtain a more responsive system behavior. Some well-known methods of more restrictive power management include, rather than shutting off power entirely, changing voltage level, or reducing the operating frequency. *See* Ex. 1044 (Orenstein), 1:32–38; Ex. 1042 (Nicol), 1:7–35. In general, lowering operating voltage of a circuit reduces the power consumption but it also reduces the operating frequency of the circuit. Systems could thus lower their overall power consumption by reducing voltage while still maintaining operations, albeit with worse performance due to lower operating frequency.

51. Other well-known power management techniques seek to balance these

performance tradeoffs. Rather than completely turning a system off, or lowering the overall operating voltage, methods were developed to “achieve energy-efficient computation by selectively turning off (or reducing the performance of) system components when they are idle (or partially unexploited).” Ex. 1032 (Benini), 299. While the discrete operations involved in a given power shutdown procedure may vary, these conservation techniques generally have seen widespread adoption. For example, “a simple policy, ubiquitously used for laptops and palmtops, is the *timeout* policy, which shuts down a component after a fixed inactivity time, under the assumption that it is highly likely that a component remains idle if it has been idle for the timeout time.” Ex. 1032 (Benini), 299.

B. Design Principles that a POSITA Would Be Aware of at the Time of the Invention of the '575 Patent

52. Below is an overview of selected basic principles of circuit design that a POSITA would have been aware of as of the invention date of the '575 Patent.

1. Architectural State or Working Context of a Processor

53. A prime target for targeted power management techniques in a digital system is the CPU, or central processing unit. The CPU is tasked with executing program instructions and accounts for a significant portion of overall power consumption of a computer system. Thus, when the CPU is idle, shutting down power to the CPU can result in significant cost savings. Because a CPU is responsible for

instruction execution and application progress, it warrants special consideration when its power is being removed. The following figure from Jurij, a textbook entitled Processor Architecture, illustrates the general components of a CPU at very high level:

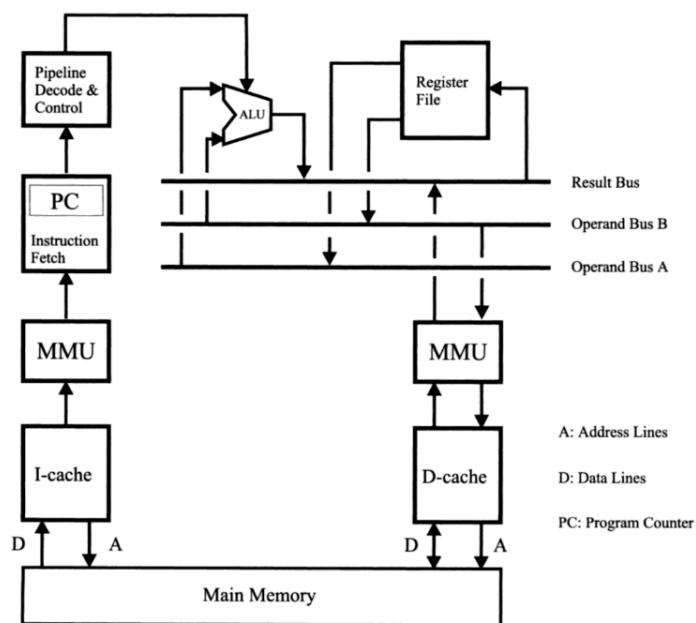


Fig. 1.1. Data path organization of a simple RISC processor

Ex. 1040 (Jurij), Fig. 1.1. The excerpt below from the Flynn Textbook, entitled Computer Architecture, shows a more detailed breakdown of a processor with control and data paths:

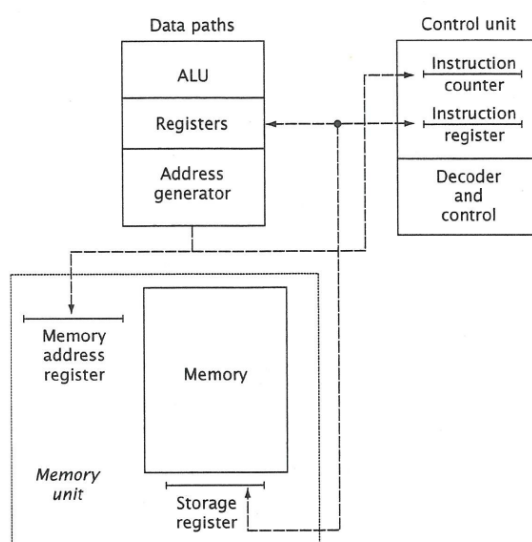


Figure 1.2 Basic units of a processor and (some) paths.

Ex. 1021 (Flynn Textbook), Fig. 1.2. The figure is from Flynn Textbook, a well-known text book entitled Computer Architecture, The register labeled as “Instruction Counter” depicted in the Control Unit is also commonly referred to as a Program Counter.

54. The I-cache and D-cache in Fig. 1 (Ex. 1040 (Jurij)) are cache memories that contain a subset of instructions and data respectively that are fetched from the main memory. The Memory Management Units (MMUs) manages the memory hierarchy As shown in both Jurij and Flynn textbook, ALU (Arithmetic Logic Unit) performs the mathematical and logic operations and stores the results in storage shown as Registers or Register File. During operation, the CPU fetches instructions from cache or main memory (in case of cache miss) and executes those instructions. The

Instruction Fetch Unit fetches the instructions. Fig. 1.1 in Ex. 1040 (Jurij) depicts PC (Program Counter) register inside the Instruction Fetch Unit. Fig. 1.2 in Ex. 1021 (Flynn Textbook) depicts the same register but refers it as an Instruction Counter. PC (Program Counter) denotes the location (i.e., the address in memory) of the instruction being executed by the CPU.

55. Most modern processors do not allow ALU instructions to directly compute on data stored in memory. Instead the data is first loaded into a register and the ALUs operate on register data. Register File in Fig. 1.1 of Ex. 1040 (Jurij) and registers in Fig. 1.2 of Ex. 1021 (Flynn Textbook) hold input data for instructions that are being processed by the processor and will also hold the output values generated from executing an instruction. The CPU may read data stored in registers and perform computations, such as adding two register values, and the result of the computation is also stored in the register. The reason for storing the computation result in a register is that such computation output may be used by other instructions and hence it is more efficient to leave the results in registers when possible, for faster access. Thus, the registers or the Register File contain the current state of an ongoing process being executed by the CPU. The PC (Program Counter) and the contents of the Register File (or Registers), and some additional information such as exception state, are commonly referred to through a variety of monikers, such as “context,” “program context,”

“working context,” and “architectural state.” Registers are generally designed using volatile memory elements with the CPU. Thus, when the power to the CPU is shut down, the context in Registers or Register File is lost. If an application has partially completed executing half of its computations before the CPU is shut down then the partial results for that application that are stored in the register are lost. However, if the context is saved and then subsequently restored when the CPU is powered again, the CPU can resume execution from the same location. Without this context, once power is restored, the CPU will have to boot up and restart, which hurts system performance.

56. Context of a CPU may be saved or restored via one of several mechanisms, such as a hardware mechanism, a microprogram or a program. When one of these mechanism is employed to save context of a CPU, then all the information necessary to resume operation at the same location is saved so that partially completed work is not lost. When the context of the CPU is restored, all Registers and Register File (e.g., PC, PSW etc.) are reloaded with the same values as to enable the CPU to resume execution from the same state it was at the time of shutdown. For example, the restore mechanism may update the Program Counter (PC) to the value it had just prior to being shut down. When the restore process is over, the CPU resumes execution with the instruction indicated by the Program Counter (PC). This step is commonly referred to as *branching to the address indicated by the Program Counter (PC)*. For example,

U.S. Patent No. 7,174,472 (Hill) that is assigned to ARM, a well-known microprocessor vendor, discloses how a Program Counter is employed to resume execution when exiting from the shutdown state. Fig. 5 in Hill is a flow chart of the power up process. Ex. 1028 (Hill), 4:33-34. Hill explains that after the working context has been restored, then “[a]t step 56 the instruction pointed to by the *program counter* [PC] register value PC as the restart point is reloaded into the instruction pipeline 14. At step 58 execution of this restart instruction is commenced.” Ex. 1028 (Hill), 6:53-56.

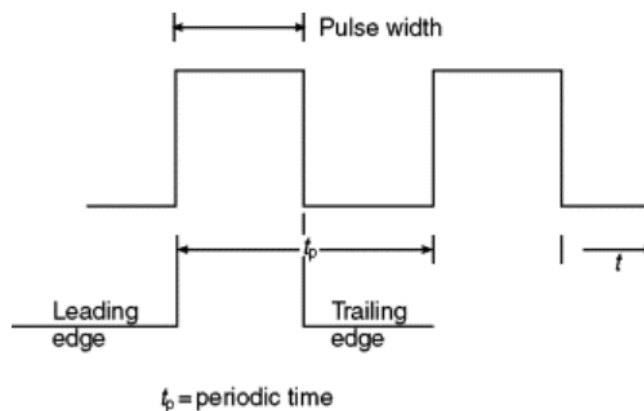
2. Clocks Are Indispensable in Digital Designs

57. Power saving schemes have targeted reducing the power consumption of clock distribution networks. A clock is a basic, necessary building block for computers and electronic circuits, acting as a pulse which dictates the timing and speed of processes within a system. Components within a computer system use this pulse as a synchronization signal, where steps of computation separated by flip-flops are performed on a clock pulse.

58. The clock signal is an essential feature of any digital synchronous system. Contemporary textbooks prior to the purported invention of the '575 Patent demonstrate this simple premise. For example, Holdsworth, a book entitled Digital Logic Design, states:

An essential feature of a synchronous system is that flip-flops which are part of the system should all change at the same instant in time. This is achieved by the use of a synchronising signal which is formally known as the clock. The clock signal is normally periodic, and there must always be a sufficient time period between adjacent clock pulses to ensure that the combinational logic has reached its steady state condition before the next clock pulse in the sequence arrives.

Ex. 1039 (Holdsworth), 163. The periodic nature of the clock signal is generally represented as a square wave, where the period in the cycle in which the clock is “high” is considered the active phase:



Ex. 1039 (Holdsworth), 164.

59. The clock signal is often represented in the art as emanating from a “clock generator.” The clock generator generates the clock which is then propagated by a clock distribution network to various components throughout the system to facilitate their operation. This can be readily seen throughout the literature. *See, e.g.,*

Ex. 1007 (Shikata), Fig. 1 (“clock generating unit”); Ex. 1006 (Jahagirdar), Fig. 2A (“clock generator”); Ex. 1008 (Crump), Fig. 3B (“Clock Synth.”); Ex. 1011 (Kang), Fig. 1 (“Main Oscillator” and “Sleep Oscillator”); Ex. 1010 (Peterson), Fig. 2, 5:29-33 (“clock generator”). To the extent that a reference describing a digital system were to not expressly discuss the operation of the clock signal, a POSITA would readily understand that this is simply because its operation is assumed as a basic building block of circuit design. In view of the above, it is clear that at the time of the invention of the ’575 Patent, a POSITA would be well versed in implementing clock generators, controlling activation and inhibition of clocks and incorporating clocks in a digital design.

3. Voltage Scaling and Resumption of Clocks

60. A POSITA would understand that when the voltage is increased from 0 V to the operating level (e.g., 3.3V or 5V), the voltage does not scale instantaneously. All circuits have associated resistance (R), inductance (L) and capacitance (C) values that limit how rapidly the voltage can rise and fall. Once the voltage reaches the operating level, it oscillates around the level before settling down to a stable and the desired value. A POSITA would know that the time required for the voltage of a circuit to reach the desired value or fall from the desired value to zero V is proportional to $e^{-t/RC}$, where R and C specify the resistance and capacitance

of the circuit and t is the time. Fig. 2 from Hu depicts the key intervals in the power gating cycle:

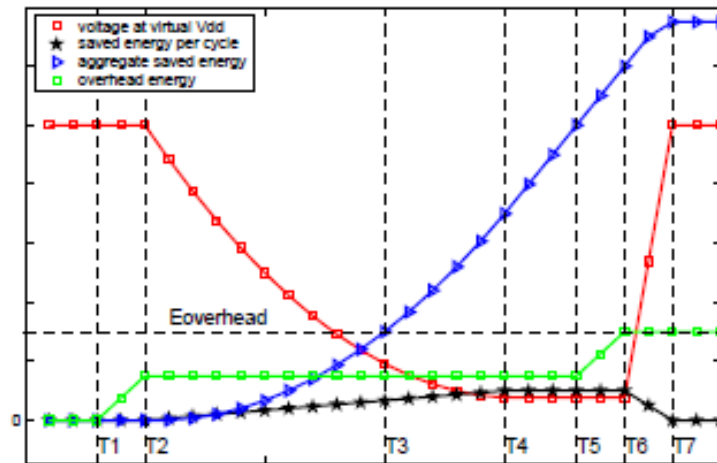


Figure 2: Key intervals in the power gating cycle

Ex. 1029 (Hu), Fig. 2. As shown in Fig. 2, a device takes time interval from T_2 to T_4 for the voltage to ramp down from virtual V_{dd} to zero voltage, and the device takes time interval from T_6 to T_7 to ramp up to V_{dd} from zero voltage. Ex. 1029 (Hu), 33. *See also generally* Ex. 1031 (Clark) (discussing delays in turning on/off voltage lines) and Ex. 1030 (Borkar), 26 (discussing power challenges when scaling technology).

61. The speed and stability of the clock signal depends on receiving stable voltage supply. Thus, a POSITA would recognize that circuits can operate reliably only after certain delay that permits the voltage level to rise and settle down. It has

been a good design practice, and even a requirement, for decades to allow for a delay before resuming the supply of the clock signals or sending other signals (e.g., interrupts) to a circuit block after restoring power to the circuit block. Ex. 1006 (Jahagirdar), 11:18-51. For example, “power-good” is an example of a signal provided by a system’s power supply that indicates the resumed voltage is within the required specification levels. Ex. 1008 (Crump), 16:3-5.

62. Similar consideration must also be given when the power supply to PLL (Phase Lock Loop) circuits and clock generators within a circuit block is restored. Normally, PLL circuits and clock generators receive a master clock signal, which they utilize to create clocks for other components. If power is restored to a PLL circuit that had been shut down, it takes a while for the PLL circuit to latch on to the incoming clock signal and then generate the clock signals for other circuit components. Ex. 1007 (Shikata), Fig. 8. Thus, it is a good design practice to allow sufficient time for the output of the PLL circuits and clock generators to provide a stable and reliable clock supply before propagating the clock signal to other circuits. Ex. 1007 (Shikata), Fig. 8. Digital circuits require a stable clock signal to operate reliably.

63. On the other hand, if the clock signal was inhibited by using an AND gate or a tristate gate, then the resumption of the clock signal does not incur the

abovementioned delay because the PLL circuits and clock generator continued to operate when the clock to the rest of the circuit was inhibited.

4. Use of Microprograms (or Microcode) Was Known in the Art

64. The '575 Patent discusses using “microprograms” to save and restore information from the CPU when entering and exiting the shutdown state. In the industry, microprogram is also referred to as a microcode. The concept of using microprograms or microcode to perform certain tasks (e.g., execute complex instructions) was not novel by the date of the invention of the '575 patent. Microprograms were first employed in the late-1950s in the EDSAC 2 Computer by Maurice Wilkes. Ex. 1023 (Wilkes) (providing an early introduction of microprograms.) Use of microprograms became prevalent in microprocessors developed in the 1960s and later years that employed a wide variety of complex instructions. Ex. 1024 (Wilkes Survey) (providing a survey of growth of interest in microprogramming.); Ex. 1025 (Patterson) (summarizing growth and use of microprogramming by 1983.) Thus, a POSITA would understand that “microprogram” is a specific term of art that refers to a particular type of program that runs at the lowest level of hardware. These microprograms are executed in response to higher-level macro-programs that are derived from human-readable

code. Each microprogram carries out a set of instructions in order to help perform the tasks required for a given macro-instruction or macro-program. As discussed below in **Section IX.B**, Collabo offered the same definition of a microprogram to the PTAB when opposing institution of IPR petitions filed by AMD. *See* Ex. 1022 (Microprogram) (collecting definitions of microprogram from various dictionaries available prior to the invention date of the '575 Patent). Prior art references, such as Jahagirdar and Crump, disclose use of *microprograms* to save and restore context. *See* Ex. 1006 (Jahagirdar), 10:18-38 and Ex. 1008 (Crump), 37:1-13, Fig. 10A. These two references, Jahagirdar and Crump, further evidence the widespread use of microcode in processors available before the invention of the '575 Patent. The assignee of Jahagirdar is Intel, Ex. 1006 (Jahagirdar), Cover [73], and Crump was assigned to IBM (International Business Machines) upon issuance, Ex. 1008 (Crump), Cover [73]. The two companies collectively dominated the microprocessor market in 1990s and early 2000s and underscore the popularity of microprograms.

65. A POSITA would be aware that, at the time of the invention of the '575 Patent, microprograms were one of the several options available to implement the functionality to save and restore working context. Ex. 1025 (Patterson), 50. A POSITA would have known the working context could be saved and restored using either hardware or high-level program, and the POSITA would have been aware of the costs

and benefits of each approach. *See* Ex. 1021 (Flynn Textbook), 5-9 (providing an overview of microprograms and comparing Hardware Decoders to Microprogrammed Decoders). A POSITA would know that the hardware-based approach was the fastest way to save and restore but most inflexible and most expensive to implement. At the other extreme, the program-based approach offered the cheapest and most flexible option but was the slowest among all the available options. Microprograms fall in the middle and offer an approach that saves and restores context quickly yet retains flexibility and is economical. *See generally* Ex. 1025 (Patterson). Patterson is a technical article, entitled Microprogramming, that published in 1992 in Scientific American and notes in its abstract: “In most modern computers the routing of information is controlled at the lowest level by a microprogram: set of stored instructions that functions in place of completely ‘hard-wired’ control system.” Ex. 1025 (Patterson), 50.

66. Thus, a POSITA had various design choices to implement the functionality of saving and restoring context, each with associated benefits and costs, and a POSITA could have chosen any of them based on the needs to the design. In view of the widespread use of microprogram as demonstrated by Jahagirdar (Ex. 1006) assigned to Intel and Crump (Ex. 1008) assigned to IBM, and disclosure in Patterson (Ex. 1025), Flynn Textbook (Ex. 1021), and Wilkes Survey (Ex. 1024), at

the time of the invention of the '575 Patent, a POSITA who wants to strike a balance between speed and flexibility would have chosen to implement the functionality of saving and restoring context using a microprogram (or microcode) over other techniques, such as using hardware or using a program. Given the broad prevalence of microprograms in the computer industry at the time of the invention of '575 Patent and the fact that microprograms have been used since the 1960s, using microprograms to implement functionality that is described in the '575 Patent would have been well within the skill of a POSITA.

5. Benefits of higher integration on a semiconductor chip

67. Technology has consistently trended towards integrating more and more functionality onto a single chip. This is a concept which is generally reflected by the well-known Moore's Law, which states that the number of transistors on a microchip doubles approximately every two years. These increasing count of transistors may be used for a variety of purposes. One common approach is to increase the number of processor cores on a single chip, leading to larger chip multiprocessors (CMP). As each core execute a thread or a process it needs to access more data to process. Thus more core count on a single chip also demands integrating more memory structures on a single chip, such as on-die memory controllers. Otherwise the cores will be ineffective as they wait for long latency memory

operations. In addition, hardware circuitry such as graphics processing unit may also be integrated on chip. When a new circuitry is integrated into a single chip then the communication between the various components within the chip become more efficient as the signals only travel within a chip boundary, which is much more power efficient than having signals cross the chip boundary via external pins. Thus, the increasing levels of system integration have been driven by “the reduction of power, smaller form factor, and lower overall cost” that consolidating additional circuits on the same die provides, as discussed in Saleh, a paper entitled System-on-Chip: Reuse and Integration. Ex. 1033 (Saleh), 2.

68. To summarize, the trend towards higher integration has been driven by the benefits that accrue from faster communication across integrated components. Placing larger number of circuit blocks on the same semiconductor die increases performance. The blocks can communicate with each other quickly if they are on the same die. If the blocks are on separate die, then communication costs go up, thereby impacting overall performance negatively. In addition, higher integration results in lower power consumption. If the circuit blocks are on the same die, they can operate at the same lower voltage levels. However, if the circuits block are on separate dies, then the signal has to leave the integrated circuit package, traverse of a printed circuit board, and then enter another integrated circuit package. This requires the signal

voltage to be ramped up and then down resulting in large power consumption and dissipation.

69. Higher integration also results in lower production costs. If all the components are on the same die, then only one die has to be packaged as opposed to multiple packages each containing one die. Moreover, higher integration does not impact wafer yield negatively as much. Over the course of last several decades, the transistor sizes continue to shrink. Thus, larger number of transistors (and by implication larger number of circuit blocks) can be placed in the die of the same size, allowing for more components to be integrated into a single die.

70. A popular methodology for achieving increased integration which saw widespread adoption starting in the early 2000s was the “system-on-chip” (“SoC”). In an SoC, “predesigned and preverified blocks—often called intellectual property (IP) blocks, IP cores, or virtual components—are obtained from internal sources, or third parties, and combined on a single chip. These reusable IP cores may include embedded processors, memory blocks, interface blocks, analog blocks, and components that handle application specific processing functions.” Ex. 1033 (Saleh), 1 (internal citation omitted). A POSITA would thus understand that it is common practice to include circuit blocks which can be placed on the same chip in order to achieve the cost savings associated with increased integration.

6. Benefits of integrating voltage regulator on SoC

71. Mirroring the trends towards higher integration, it was common by 2006 to integrate power supply circuitry onto a system on a single die, such as through on-chip voltage regulators. For example, Lee discloses that the voltage regulator can be integrated on a system-on-chip along with other components, such as a CPU, RAM, peripheral devices and power controller. Ex. 1005 (Lee), 5:48-51. *See also, e.g.*, Ex. 1034 (Nair), 188 (disclosing implementation details of on-chip voltage regulators for microcontrollers); Ex. 1035 (Carley), 111 (disclosing details of “a **completely** on-chip voltage regulation scheme which minimizes power consumption in digital circuit operating on multiple voltages.”) (emphasis in original). Nair discloses a CMOS-based distributed voltage regulation system for microcontrollers that integrates the voltage regulator on the same die as the microcontroller, and also confirms that the silicon results correlated with the circuit models. Ex. 1034 (Nair), 188, 191. Such solutions would have provided the benefits of higher integration to support smaller dies, better performance, and lower overall power consumption. Ex. 1034 (Nair), 188, 191. By the time of the invention of the '575 Patent, a POSITA would have appreciated the benefits of integrating voltage regulator in a system-on-chip and would have the knowledge to accomplish such an integration because integrating voltage regulators was a routine and well-understood design step by that time frame.

7. Benefits of multiple power supplies when multiple voltage domains are present

72. A digital circuit normally contains a power supply to provide a stable and reliable power supply to various components in the circuit. Traditionally, one power supply was sufficient to use because all of the circuits received the same voltage levels, and all circuits were powered on or off at the same time. Higher integration of circuits on the same semiconductor die has resulted in circuits with different voltage requirements being placed on the same die. For example, at the time of the invention of the '575 Patent, it was common to place a CPU (or a processor) and the RAM (random access memory) on the same die. Ex. 1007 (Shikata), Fig. 1 and Ex. 1005 (Lee), Fig. 1. Power conservation techniques cause the CPU and the RAM to be at various voltage levels depending on the operating mode (e.g., normal, retention level, and standby) of the circuit. It is difficult for a single power supply to provide varying levels of voltage at the same time to different circuits. Thus, it became common to have multiple power supplies on the same semiconductor die where each power supply was dedicated to a particular group of circuits. *See* Ex. 1026 (Igarashi), 36 (disclosing “a low-power design method that utilizes [] multiple supply voltages” and disclosing a “method [that] reduces [] power supply voltage of the non-timing critical parts of circuit, while the power supply

voltage of the timing critical parts of the circuits is kept.”). Likewise, Hill, a reference that discloses saving and restoring context of a processor when entering and exiting standby modes, also relies upon two power supplies to provide power to the processor and to memory cells. Ex. 1028 (Hill), 5:54-56 (noting that the “balloon circuits 30 are supplied with their own storage cell power supply, which is maintained during a power down event.”). Similarly, Shikata also relies upon one power supply to provide power to the CPU and related circuits whereas another power supply to provide power solely to RAMs on the die. Ex. 1007 (Shikata), 6:51-7:49 and Ex. 1028 (Hill), 3:50-56, 5:54-56; *see also* Ex. 1027 (Chang), 436 (noting in 1997 that “recently, the use of multiple supply voltages on the chip is attracting attention” and noting that such a “scheme tends to result in smaller area overhead compared to parallel architectures.”). Thus, the by time of the invention of the ’575 Patent, a POSITA would have appreciated the benefits of having multiple power supplies when a design requires multiple voltage domains and levels, and a POSITA would be able to implement such a scheme because having multiple voltage supplies was routine and well-understood design principle by that time.

8. Communications to and between processes/circuit block

73. Another well-understood building block of digital systems is the means by which various components in that system communicate. In order to operate

effectively, various circuit blocks within a system must be able to operate in parallel and coordinate with each other. The communications between various components of a system serve many functions—“[i]nterprocess communication mechanisms are not merely there to allow the processes in a distributed system to communicate; they are also there to provide a mechanism that shields one process from failures of another.” Ex. 1036 (Mullender), 47. Mullender, a technical article, entitled Interprocess Communication, was published in 1992 and discusses the benefits of interprocess communications: “Interprocess communications mechanisms provide a small and comprehensive set of interaction possibilities between processes which, if used exclusively for the interaction between them, forces clean and simple interface between processes.” Ex. 1036 (Mullender), 47.

74. One essential mechanism used in digital systems to inform a circuit block when an event occurred is the “interrupt,” which has been known in the art for over fifty years. *See* Ex. 1037 (Morgan), 734-735. Morgan is a technical article, entitled An Interrupt Based Organization for Management Information Systems, that was published in 1970 and discussed use of interrupt mechanism both in hardware and software technology. 1037 (Morgan), 734-735. The interrupt signal provides a means by which a first circuit block can inform a second circuit block of an event, pause normal operation of the second block, and request execution of a desired

routine or program by the second circuit block before normal operation of the second circuit block resumes. *See, e.g.*, Ex. 1038 (Fischer-Cripps), 102 (“When an interrupt signal is received, the CPU suspends its activities and runs an interrupt service routine or interrupt handler. After the interrupt service routine has finished executing, normal execution is resumed.”). Fischer-Cripps is a book, entitled Newnes Interfacing Companion, directed towards computers, transducers, instrumentation, and signal processing, and thus, discusses how processes and blocks communicate. Once the interrupt routine is complete, the second circuit block informs the first circuit block which issued the interrupt that the desired operation is complete. The interrupt signal can be a single bit signal conveying minimal information (i.e., a binary value of 0 or a 1) or a multi-bit signal that conveys additional information. By the time of the invention of the ’575 Patent, a POSITA would have appreciated the benefits of inter-block communications, such as through interrupts, in design of systems, and would have been able to implement such schemes because use of inter-block communications was a routine and widely used design principle by that time frame.

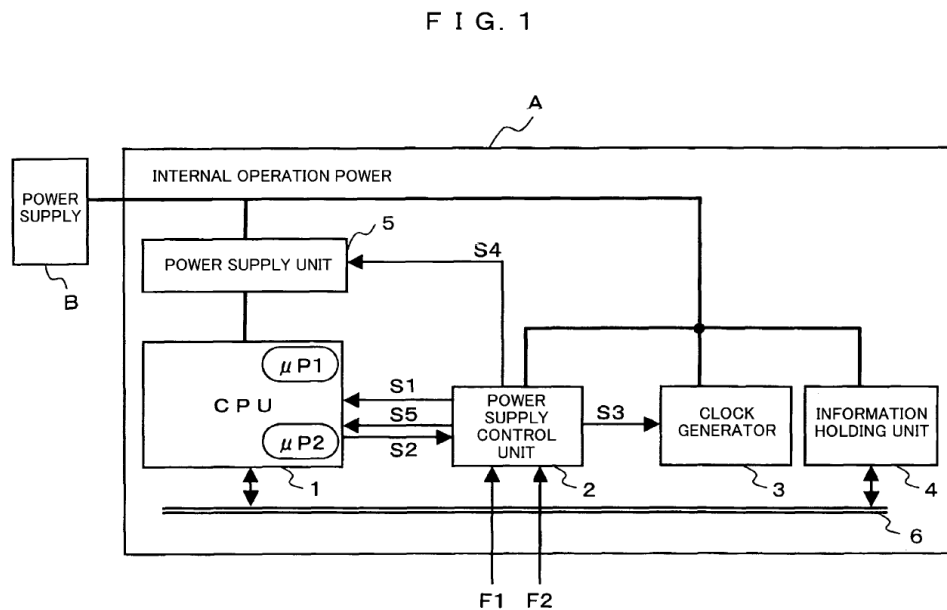
VIII. OVERVIEW OF THE ’575 PATENT

A. Background

75. The ’575 Patent is titled “Microcontroller for Controlling Power

Shutdown Process.” According to the “Summary of the Invention” section of the ’575 Patent, “a main object of the present invention to provide a microcontroller capable of easily and rapidly evacuating and restoring information necessary in proceeding with a program when realizing a standby state in which internal operation voltage supply is shutdown, a state in which an internal operation voltage is lowered, or the like.” Ex. 1001 (’575 Patent), 2:26-31.

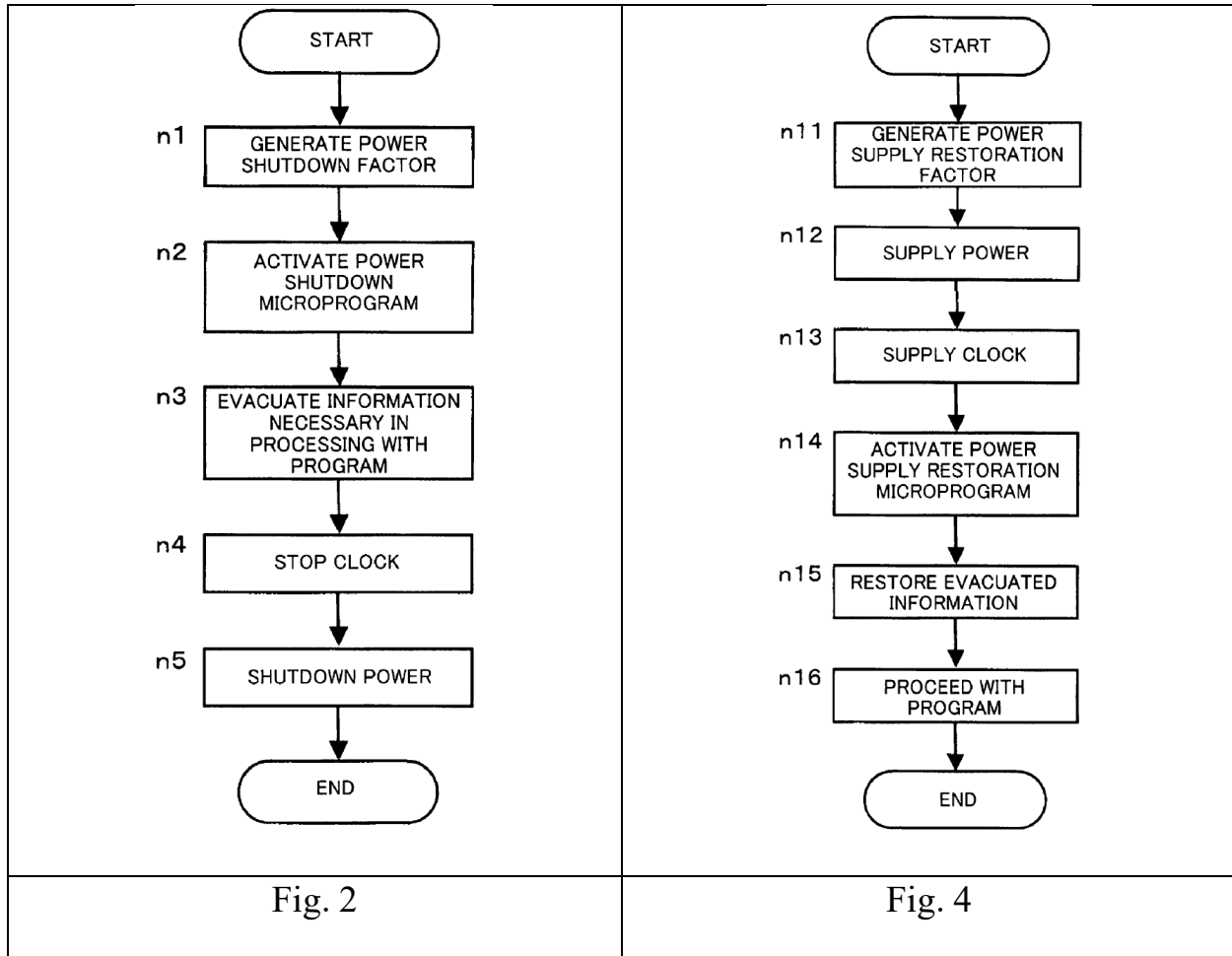
76. Figure 1 of the ’575 Patent is reproduced below:



Ex. 1001 (’575 Patent), Fig. 1. Per the ’575 Patent, Figure 1 is a block diagram showing a configuration of a microcontroller according to a first embodiment of the present invention. Ex. 1001 (’575 Patent), 2:22-24.

77. Figs. 2 and 4 from the ’575 Patent depict flowchart showing the operation

of a power shutdown process and the operation of a power supply restoration process of the microcontroller according to one of the embodiments of the '575 Patent:



Ex. 1001 ('575 Patent), Figs. 2 and 4.

78. With respect to Fig. 2, the patent states that “in step n1, the power shutdown factor F1 is generated, which is then input from the external terminal to the power supply control unit 2, and the power supply control unit 2 outputs the shutdown request signal S1 to the CPU 1.” Ex. 1001 ('575 Patent), 9:39-42. “The CPU 1 upon

receiving the shutdown request signal S1 from the power supply control unit 2, activates the power shutdown microprogram $\mu P1$ in step n2 after terminating the normal command execution.” Ex. 1001 ('575 Patent), 9:43-46. “The power shutdown microprogram $\mu P1$ evacuates the information (PC, PSW, SP, etc.) necessary in proceeding with the program to the information holding unit 4 or the register via the internal bus 6 in step n3, and lastly outputs the evacuation completed signal S2 to the power supply control unit 2.” Ex. 1001 ('575 Patent), 9:50-54. “The power supply control unit 2, upon receiving the evacuation completed signal S2, outputs the clock stop control signal S3 to the clock generator 3, and also outputs the power shutdown control signal S4 to the power supply unit 5 in step n4.” Ex. 1001 ('575 Patent), 9:55-59. “The power supply unit 5, upon receiving the power shutdown control signal S4, shuts down power supply to the CPU 1 in step n5. The power shutdown of the CPU 1 is thereby completed.” Ex. 1001 ('575 Patent), 9:60-63.

79. With respect to Fig. 4, the patent states that “in step n11, the power supply restoration factor F2 is generated, which is then input from the external terminal to the power supply control unit 2, and the power supply control unit 2 outputs the power supply control signal S4 to the power supply unit 5.” Ex. 1001 ('575 Patent), 9:65-10:3. “The power supply unit 5, upon receiving the power supply control signal S4, starts to supply power to the CPU 1 in the next step n12. When the power supply

voltage stabilizes due to functions of a power supply voltage monitoring circuit and stability waiting timer, and an oscillation waveform output by an oscillation circuit stabilizes as a result thereof, the power supply control unit 2, upon detecting such stability, outputs the clock supply control signal S3 to the clock generator 3. Simultaneously, the power supply control unit 2 outputs the restoration request signal S5 to the CPU 1.” Ex. 1001 (’575 Patent), 10:4-14. “In step n13, the clock generator 3 restarts to supply the operation clock. In step n14, the CPU 1, upon receiving the restoration request signal S5, activates the power supply restoration microprogram μ P2. In step n15, the activated power supply restoration microprogram μ P2 restores the information (PC, PSW, SP, etc.) necessary in proceeding with the program that had been evacuated in the information holding unit 4 in time of power shutdown to the CPU 1 via the internal bus 6. Lastly, in step n16, the power supply restoration microprogram μ P2 branches the processes in the CPU 1 to the addresses indicated by the program counter PC. The state thus changes from the shutdown state to the program execution state, and the program proceeds.” Ex. 1001 (’575 Patent), 10:15-27.

B. Prosecution History of the ’575 Patent

80. I have reviewed **Section VI. B** of the Petition.

C. Litigation History of ’575 Patent

81. I have reviewed **Section VI.C** of the Petition.

D. IPR Petitions Filed by AMD

1. IPR2019-01523 (Jahagirdar IPR)

82. I have reviewed **Section VI.D** of the Petition.

2. IPR2019-01524 (Chung IPR)

83. I have reviewed **Section VI.D** of the Petition.

IX. CLAIM CONSTRUCTION

84. For all claim terms, I have applied the ordinary and customary meaning of those terms as understood by one of ordinary skill in the art in light of the specification and prosecution history of the '575 Patent except as noted below.

85. I have reviewed **Section VII** of the Petition.

In forming my opinions disclosed in this Declaration, I have employed the agreed constructions listed below:

Term	Agreed Construction
Whether the preamble of claim 1 (“a microcontroller”) is limiting. (claim 1)	The preamble of claim 1 is limiting.
“power supply device” (claim 1)	“power supply”
“restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown” (claim 1)	“Restores the information necessary to proceed with the program, the information was evacuated from the CPU in the information holding unit in the power shut down process”
“a power shutdown range” (claims 11 and 12)	“circuit for which power supply will be shutdown”

86. I further understand that Collabo and Qualcomm dispute the construction

of other terms and the competing constructions are listed below. I support Qualcomm's proposed constructions and in forming my opinions disclosed in this Declaration, I have employed Qualcomm's proposed constructions, as listed below:

A. Microcontroller

Collabo's Construction	Qualcomm's Construction
Plain and ordinary meaning	"a single integrated chip as opposed to a set of multiple chips (a "chipset")"

87. I have reviewed **Section VII.A** of the Petition.

B. Microprogram

Collabo's Construction	Qualcomm's Construction
Plain and ordinary meaning	"a control program (or a microcode) that runs at the lowest level of the hardware to control internal source of the CPU to execute command, interrupt processing, and the like, and is held entirely within the CPU as opposed to a program that is held in main memory until executed"

88. I have reviewed **Section VII.A** of the Petition.

C. Power shutdown microprogram

Collabo's Construction	Qualcomm's Construction
Plain and ordinary meaning	"a microprogram that evacuates the information necessary in proceeding with the program to the information holding unit, and then outputs the evacuation completed signal to the power supply control unit"

89. I have reviewed **Section VII.A** of the Petition.

D. Power supply restoration microprogram

Collabo's Construction	Qualcomm's Construction
Plain and ordinary meaning	"a microprogram that restores the information necessary in proceeding with the program and evacuated in the information holding unit to the CPU, and then branches the processes to the addresses indicated by the program counter to continue the program execution from the shutdown state"

90. I have reviewed **Section VII.A** of the Petition.

91. Because Collabo is proposing these terms be given their "plain and ordinary meaning," my opinions in this Declaration are equally valid under Collabo's interpretation of the claim terms.

X. OVERVIEW OF PRIOR ART

92. The section provides an overview of the relevant aspects of the prior art cited in this Declaration.

A. U.S. 7,293,183 (Lee)

93. Lee, entitled System for Storing Working Context in a Non-volatile Memory While in a Power-off Suspend Mode and Restoring the Working Context When the Power-off Suspend Mode is Released, was filed on August 26, 2003, and issued as a patent on Nov. 6, 2007. The application filed at the USPTO claims priority to a Korean application filed on August 28, 2002.

94. Lee discloses steps taken by a system-on-chip (SOC) to save working

context when entering a *power-off standby mode* and steps taken by the SOC to restore the working context when exiting that standby mode. The patent mentions that the *power-off standby mode* is a mode employed by some SoCs “during which power supplied to all or several circuits of the SOC, which are not needed to be used for a long period of time, is cut off to remove static leakage current generated in the SOC.” Ex. 1005 (Lee), 1:43-47. The patent mentions that the power-off standby mode minimizes power consumption:

The present invention provides an apparatus and method that can minimize power consumption during a power-off standby mode of a portable apparatus in a system-on-a chip (“SOC”) design.

Ex. 1005 (Lee), 1:59-62.

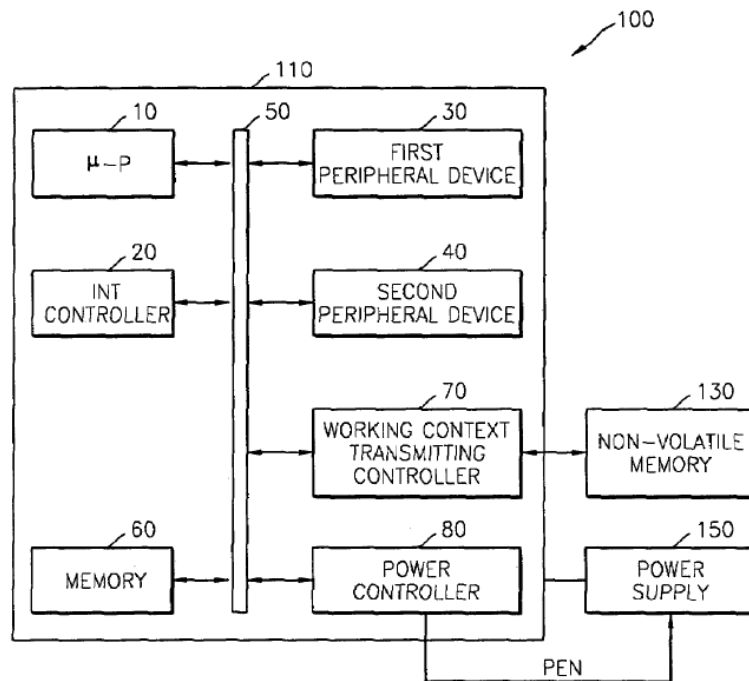
95. Lee recognizes that “when the power supplied to the SOC is cut off, the working context of all or several circuits of the SOC is lost. When power is supplied again to the all or several circuits of the SOC, the all or several circuits of the SOC boot up again. In this case, a long time is required to boot up the circuits, and the all or several circuits of the SOC do not recover to a state just before the power was cut off.” Ex. 1005 (Lee), 1:48-55. Thus, Lee recommends saving the working context when entering the power-off standby mode in order to rapidly recover to the same just prior to entering the standby mode:

The present invention provides an apparatus and method by which working contexts of circuits within an SOC to which the power supply is cut off can be rapidly stored in a low power non-volatile memory to enable rapid recovery to a state just before the power supply was cut off.

Ex. 1005 (Lee), 1:63-67.

1. Structure of the System-on-Chip (SOC)

96. Fig. 1 of Lee is a block diagram of a system including a SoC according to one embodiment of the Lee invention:



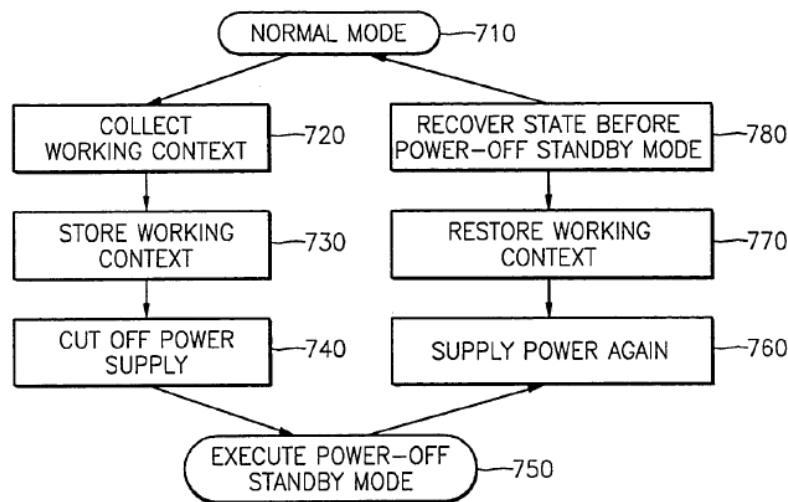
Ex. 1005 (Lee), Fig. 1.

97. Fig. 1 depicts that the system 100 includes a SOC 110 in addition to a

non-volatile memory 130 and a power supply 150 that are coupled with the SOC 110. Ex. 1005 (Lee), 5:35-40. Lee states that “[t]he power supply 150 supplies a predetermined operating power to the SOC 110” and that “[t]he power supply 150 may be outside the SOC 110 or may be implemented as an on-chip regulator inside the SOC 110.” Ex. 1005 (Lee), 5:48-51. The SOC 110 includes a microprocessor 10 that is connected to various components on the chip through an on-chip bus 50. Such components include memory 80, interrupt controller 20, peripheral devices 30 and 40, working context transmitting controller (WCTC) 70 and power controller 80. Ex. 1005 (Lee), 5:51-60.

2. Steps Taken to Enter the Power-off Standby Mode

98. Fig. 7 of Lee is a flow chart illustrating a method of storing and recovering working context according to one embodiment of the Lee invention:



Ex. 1005 (Lee), Fig. 7. The left side of the flow chart describes how SOC 110 transitions from the normal to the power-off standby mode. Fig. 7 depicts that the arrows point from block 780 to 770 and also from block 770 to block 760. My review of the specification indicates that the orientation of the arrows is incorrect and should be opposite to what is depicted in Fig. 7. The arrows should point from block 760 to block 770 and then from block 770 to block 780. Ex. 1005 (Lee), 10:34-57.

99. **Step 710:** In Fig. 7, Step 710 represents that normal operating mode of the SOC. Ex. 1005 (Lee), 10:7-8. Lee discloses various scenarios through which the SOC may transition from Step 710 to Step 720. First, the “power-off standby mode is selected by a predetermined power management program, which may be resident in an operating system (“OS”) of the SOC 110.” Ex. 1005 (Lee), 6:5-9. Second, the SOC 110 transitions to the power-off standby mode to reduce static leakage current of the SOC 110, “[a]fter a predetermined standby time set by a user has passed or a user selects a power-off standby mode.” Ex. 1005 (Lee), 6:10-13.

100. In the second scenario, “the INT controller 20 receives an interrupt signal output from a predetermined hardware module, and transmits the interrupt signal to the microprocessor 10. For example, the INT controller 20 receives an interrupt signal inputted through a first peripheral device 30, such as a keypad of a mobile phone, and

transmits the interrupt signal to the microprocessor 10.” Ex. 1005 (Lee), 6:14-20.

101. **Step 720:** This step is reached when a power-off standby mode is selected and the microprocessor starts to collect the relevant working contexts. Ex. 1005 (Lee), 10:7-11. When the microprocessor receives “the predetermined interrupt signal, . . . [it] transmits all working contexts stored in all registers of the first peripheral device 30, the second peripheral device 40, and the microprocessor 10 to the memory 60 through the on-chip bus 50.” Ex. 1005 (Lee), 6:21-25.

102. **Step 730:** “In step 730, after the collected working context is temporarily stored in the memory 60, the collected working context is transmitted to the non-volatile memory 130 outside the SOC 110.” Ex. 1005 (Lee), 10:12-19. Lee discloses that the working context can be transmitted from memory 60 to non-volatile memory 130 either (i) by the microprocessor 10 or (ii) by the WCTC 70 based on a predetermined storage command output from the microprocessor 10 or one of the other hardware modules 30 or 40 to the WCTC 70. Ex. 1005 (Lee), 6:26-41. In the latter option (i.e., when the WCTC 70 transfer the working context to non-volatile memory 130), the WCTC 70 notifies the power controller 70 or the microprocessor 10 when the entire working context has been copied from memory 60 to non-volatile memory 130. Ex. 1005 (Lee), 6:42-45.

103. **Step 740:** “In step 740, when all the working context is stored in the non-

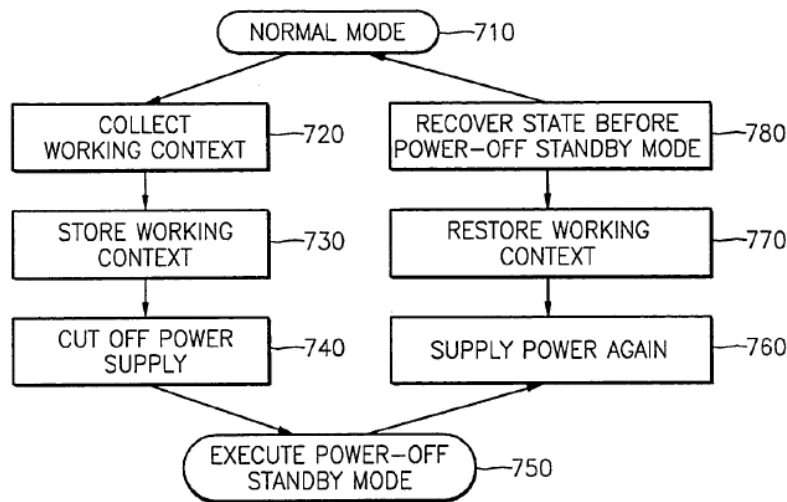
volatile memory 130, the power supply 150 for supplying power to the at least one hardware module 10, 30, or 40 . . . is cut off in response to a power control signal PEN output from the power controller 80.” Ex. 1005 (Lee), 10:20-25. Lee discloses that “[t]he power controller 80 responds to the predetermined command signal . . . to output a power control signal PEN to the power supply 150.” Ex. 1005 (Lee), 6:46-49. The predetermined command signal can be output either from the working context transmitting controller 70 or the microprocessor 10. Ex. 1005 (Lee), 6:46-49. In response to the power control signal PEN, the power supply 150 cuts off power supplied to the at least one operating hardware module 10, 30, or 40 except to the power controller 80 and/or the working context transmitting controller 70. Ex. 1005 (Lee), 6:49-54.

104. **Step 750:** Finally, “the power supplied to the hardware module 10, 30, or 40 is cut off, and in step 750, the SOC 110 executes a power-off standby mode.” Ex. 1005 (Lee), 10:25-27. When the power supply 150 cuts off the power supply, “the SOC 110 enters a power-off standby mode.” Ex. 1005 (Lee), 6:53-54. As mentioned above, even in the power-off standby mode, power is always supplied to the power controller 80, and in some embodiments to the WCTC 70. Ex. 1005 (Lee), 10:29-33.

3. Steps Taken to Exit the Power-off Standby Mode

105. As mentioned above, Fig. 7 is a flow chart illustrating a method of storing

and recovering working context. Ex. 1005 (Lee), 10:4-7. The right side of the flow chart describes how the SOC 110 transitions from the power-off standby mode to the normal mode:



Ex. 1005 (Lee), Fig. 7.

106. **Step 760:** The SOC 110 exits the power-off standby mode “when a user releases a power-off standby mode, e.g., when an interrupt signal is generated from the outside, or when a predetermined standby time set by the user has passed.” Ex. 1005 (Lee), 9:24-28. In that event, the power controller 80 outputs a power control signal PEN to the power supply unit 150 that prompts “the power controller 80 to supply power again to the hardware modules 10, 30 and 40 within the SOC 110, to which power supply is cut off during the power-off standby mode.” Ex. 1005 (Lee), 9:29-41. In context of Fig. 7, Lee discloses that in Step 760, “when the power-off standby mode

is released, the power supply 150 responds to the power control signal PEN output from the power controller 80, such that power is supplied again to the hardware modules 10, 30, and 40.” Ex. 1005 (Lee), 10:33-38.

107. **Step 770:** Lee discloses that “[a] predetermined reset signal generation circuit (not shown [in Fig. 1]) on the SOC 110 outputs a reset release signal for releasing reset of the hardware modules 30, 40, and 70 to the hardware modules 30, 40, and 70 except the microprocessor 10.” Ex. 1005 (Lee), 9:33-37. In response to the reset release signal, the WCTC 70 transmits or restores the working context for hardware modules 10, 30, and 40, that is stored in the non-volatile memory 130, to the memory 60. Ex. 1005 (Lee), 9:38-42. “In this case, the microprocessor 10 still remains in a reset state.” Ex. 1005 (Lee), 9:42-43. When the WCTC 70 has restored all working context to memory 60, the WCTC 70 notifies the reset signal generation circuit (not depicted in Fig. 1), which in turn outputs a reset release signal to the microprocessor 10 for “releasing of reset of the microprocessor 10.” Ex. 1005 (Lee), 9:48-50.

108. Lee teaches those steps in context of Fig. 7: “In step 770, the working context transmitting controller 70 copies or transmits the working context with respect to the hardware modules 10, 30, and 40, which is stored in the non-volatile memory 130, to the memory 60, so that the working context with respect to the hardware modules 10, 30, and 40, which is stored in the non-volatile memory 130, is restored.

In this case, the microprocessor 10 still remains in a reset state.” Ex. 1005 (Lee), 10:39-46.

109. **Step 780:** When the microprocessor 10 is released by a reset release signal, Ex. 1005 (Lee), 9:48-50, it fetches a program code from a predetermined address to determine whether the SOC 110 has recovered from a power-off standby mode. Ex. 1005 (Lee), 9:51-59. If it is determined that the SOC has recovered from the power-off standby more, the microprocessor 10 restores the working context for the hardware modules 30 and 40 from memory 60 and recovers its own context. Ex. 1005 (Lee), 9:60-54.

110. When discussing Fig. 7, Lee states that “[i]n step 780, after the working context with respect to the hardware modules 10, 30, and 40 is restored to the memory 60, the microprocessor 10 recovers a state of the hardware modules 30 and 40 to a state immediately before the power-off standby mode was executed by using the working context with respect to the hardware modules 30 and 40, which is restored to the memory 60. Since the microprocessor 10 has restored using its own working context restored to the memory 60, the SOC 110 performs a predetermined operation in the state immediately before the power-off standby mode was executed.” Ex. 1005 (Lee), 10:47-57. “If it is determined that the SOC [110] has recovered from a power-off standby mode, the microprocessor 10 restores the working context with respect to the

hardware modules 30 and 40 from the memory 60 in response to the reset release signal, and recovers its own working context.” Ex. 1005 (Lee), 9:59-64.

111. **Step 710:** “Accordingly, the microprocessor 10 and the hardware modules 30 and 40 are recovered to a state immediately before the power-off standby mode was executed. Thus, the SOC 110 or 210 or the system 100 or 200 can operate in the state immediately before the power-off standby mode was executed.” Ex. 1005 (Lee), 9:65-10:3.

B. U.S. 7,219,248 (Shikata)

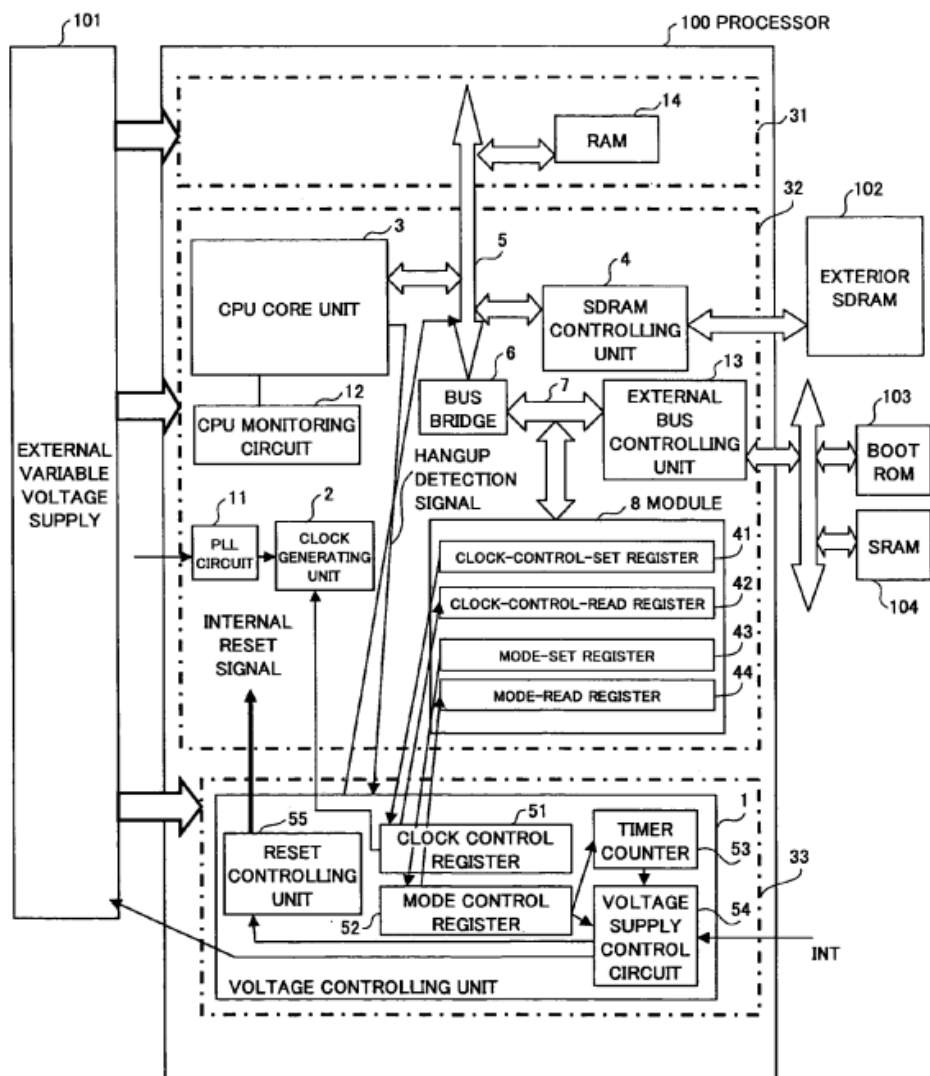
112. Shikata, entitled Semiconductor Integrated Circuit Operable to Control Power Supply Voltage, was filed on Oct. 20, 2004 and issued as a patent on May 15, 2007. The application filed at the USPTO claims priority to an application filed in Japan on June 21, 2004.

113. Shikata discloses a semiconductor integrated circuit that evacuates context of a processor when entering a *standby stage* steps and that restores context of the processor when returning from the standby state. Shikata describes the *standby state* for the processor as the state in which voltage supply to the processor is suspended, i.e., voltage supply is cut-off. In the Field of Invention section, Shikata states that “[t]he present invention generally relates to semiconductor integrated circuits, and particularly relates to a semiconductor integrated circuit which controls a

power supply voltage for the purpose of reducing power consumption.” Ex. 1007 (Shikata), 1:17-20.

1. Structure of the Semiconductor Integrated Circuit

114. Fig. 5 is a block diagram depicting a semiconductor integrated circuit according to the invention disclosed in Shikata:



Ex. 1007 (Shikata), Fig. 5. The semiconductor circuit depicted in Fig. 5 of Shikata is substantially similar to the circuit disclosed in Figs. 1 and 4. Comparing the embodiments shown in Fig. 1 and Fig. 5, Fig. 5 depicts various power-supply blocks whereas Fig. 1 does not. In addition to circuit blocks depicted in Fig. 1, Fig. 5 depicts a reset controlling unit 55, a clock control register 51 and mode control register 52 in the voltage controlling unit 33. Thus, the description in the specification of the circuit blocks shown in Fig. 1 are equally valid for the circuit blocks shown in Fig. 5.

115. Shikata describes the system depicted on Fig. 1 (and, as a result the system shown in Fig. 5) as a “semiconductor integrated circuit (processor) 100.” Ex. 1007 (Shikata), 3:64-67. As depicted in Fig. 5, the key components of processor 100 include CPU core unit 3, RAM 14, PLL circuit 11, clock generating unit 2, and voltage controlling unit 33. *See* Ex. 1007 (Shikata), 3:66-4:13. The processor 100 receives voltage from a variable voltage supply 101. Ex. 1007 (Shikata), 3:66-4:13. In the figure, the power supply to the processor 100 is divided into three blocks: power supply block 31 that contains the RAM 14, power supply block 32 that contains the CPU core unit 3 and associated circuits, and power supply block 33 that contains voltage controlling unit 1 and related circuits.

116. “The operating voltage of the processor 100 is supplied from the external

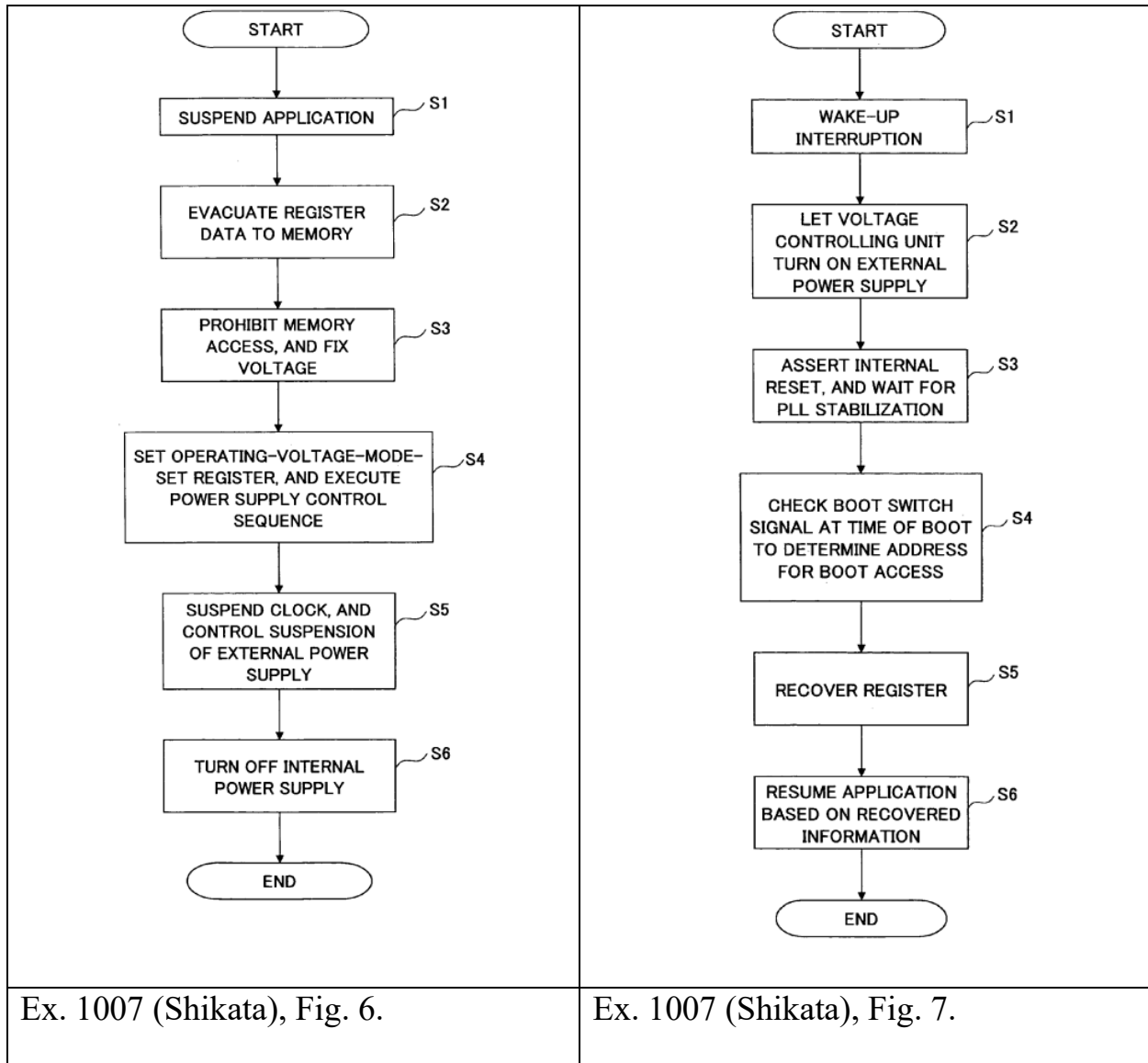
variable voltage supply 101. It is possible to suspend the supply of the operating voltage or lower the operating voltage of the processor 100 by controlling the external variable voltage supply 101 through a control signal output from the voltage controlling unit 1 of the processor 100. The clock generating unit 2 of the processor 100 generates operating clock signals for provision to individual modules including the CPU core unit 3 based on the frequency-multiplied clock signal supplied from the PLL circuit 11.” Ex. 1007 (Shikata), 4:29-39.

117. The voltage controlling unit 1 in Shikata includes a control register 15 that determines how the processor 100 recovers from the standby mode. Ex. 1007 (Shikata), 5:4-20. The control register 15 indicates, among other things, whether the boot code and the register data are stored in the internal RAM 14. Ex. 1007 (Shikata), 5:4-20. In that situation, when the system is returning from the standby state, the voltage controlling unit 1 asserts a boot switch signal to ensure that the system restores register values from the internal RAM 14 and resumes operation instead of booting from scratch, such as the 103 Boot ROM. Ex. 1007 (Shikata), 5:4-20.

2. Steps Taken to Transition to the Standby State

118. Fig. 6 is a flowchart showing a process performed when transitioning to the standby state from the normal operating mode and Fig. 7 is a flowchart showing a process performed when transitioning back from the standby state to the normal

operating mode:



119. **Step S1:** In Step S1, processor 100 starts the process of entering the standby state. Shikata states that “an ongoing application is suspended” in Step S1. Ex. 1007 (Shikata), 8:21.

120. **Step S2:** In Step 2, information necessary to return to the current state

(i.e., the context of the processor 100) is saved in the internal RAM along with a boot program to assist in recovery. Shikata discloses that “register data and the like that need to be saved, together with a boot program for recovery are stored in the internal RAM 14” at Step S2. Ex. 1007 (Shikata), 8:22-24.

121. As shown in Fig. 5, the boot ROM 103 stores a boot program that is used to boot the processor 100 from a complete power-off state. Ex. 1007 (Shikata), 4:42-45. Shikata discloses storing the boot program and the context of the processor 100 in the internal RAM to speed up the recovery from the standby state. Shikata discloses that “[i]n the present invention, the boot program is stored in the internal RAM 14, and register values necessary for the resumption of an operation are saved to the internal RAM 14 when a transition is to be made to shift into a standby state in which power supply is partially suspended for the purpose of reducing power consumption.” Ex. 1007 (Shikata), 4:46-52. Shikata discloses that all the information needed to resume operation upon exit from the standby state is saved.

122. Unlike the boot ROM 13, the internal RAM 14 is on the same integrated circuit as the CPU core unit 3 and other related circuits. As result, transmission of the boot program and context from internal RAM 14 to the CPU core unit 3 is faster than fetching the same information from the boot ROM 13. Shikata echoes the benefits of using the boot program stored in the internal RAM 14:

In the present invention as described above, the boot instructions are fetched directly from the internal RAM 14 connected to the on-chip bus 5, rather than fetched from the boot ROM 103 through a normal access path. This eliminates a need to access the boot ROM 103 that is extremely slow, and significantly reduces the number of processing steps, thereby achieving high-speed register evacuation and recovery in a simple and efficient manner.

Ex. 1007 (Shikata), 5:50-57.

123. **Step S3:** In Step S3, the voltage to the internal RAM 14 is lowered in order to reduce power consumption but still retain data stored in the RAM 14. Shikata discloses that “[a]t step S3, access to the internal RAM 14 to which the register values and the like are saved is prohibited, and the voltage of the internal RAM 14 is changed to and fixed at a lower voltage in order to suppress the leak currents of the internal RAM 14 at the time of standby.” Ex. 1007 (Shikata), 8:24-29.

124. **Step S4:** In Step S4, additional information that will assist in returning from the standby state to the normal operating state is saved. Shikata discloses that “[a]t step S4, information for use at the time of recovery such as information indicative of the need to recover the saved register values is set in the mode-set register or the like, and a predetermined power supply control sequence is activated for the purpose of shifting into a standby state.” Ex. 1007 (Shikata), 8:29-34. For example, “[t]he control register 15 is set to a value indicative of a standby state, a value indicative of a

hangup, a value indicating that the boot code (boot program instructions) and saved register data are stored in the internal RAM 14, etc.” Ex. 1007 (Shikata), 5:7-11.

125. **Step S5:** In Step S5, the clock supply to the circuit blocks in power supply block 32 is suspended and the external variable voltage supply 101 is instructed to suspend voltage to circuits in power supply block 32. Shikata discloses that “[a]t step S5, the clock generating unit 2 suspends clocks according to the instruction from the voltage controlling unit 1, and the voltage controlling unit 1 issues an instruction to the external variable voltage supply 101 so as to suspend the supply of the internal power supply voltage except for the power supply of the voltage controlling unit 1 and the internal RAM 14.” Ex. 1007 (Shikata), 8:34-40. Shikata mentions that the power supply to various power supply blocks is provided independent of each other and, in the standby state, power supply to only the power supply block 32 is suspended:

The power supply blocks 31 through 33 receive respective, separate power supplies from the external variable voltage supply 101, and are configured to control the power supply independently of each other.

In a standby state, power supply is suspended only with respect to the power supply block 32 in which a majority of circuits inclusive of the CPU core unit 3 are situated whereas the power supply block 31 and the power supply block 33 continue to receive the internal voltage.

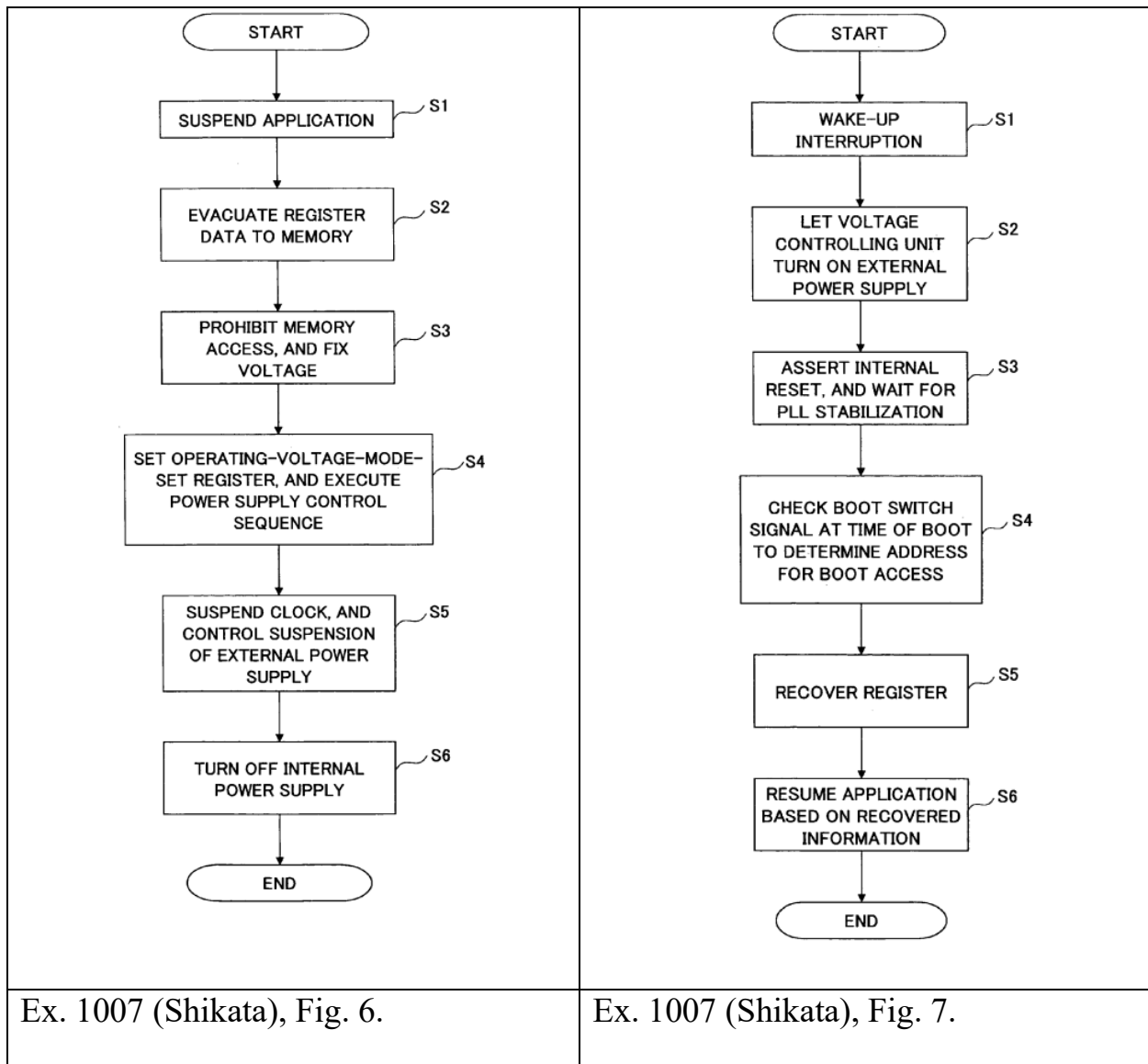
Ex. 1007 (Shikata), 6:48-56.

126. **Step S6:** Finally, in Step S6, the power supply to blocks in power supply

block 32 is suspended. Shikata discloses that “[a]t step S6, the external variable voltage supply 101 suspends voltage supply in response to the instruction.” Ex. 1007 (Shikata), 8:40-42.

3. Steps Taken to Transition From the Standby State

127. Fig. 6 is a flowchart showing a process performed when transitioning to the standby state from the normal operating mode and Fig. 7 is a flowchart showing a process performed when transitioning back from the standby state to the normal operating mode:



128. **Step S1:** In Step S1, the processor 100 starts the process of returning from the standby state to the normal operating state. Shikata discloses that “[a]t step S1 of FIG. 7, the voltage controlling unit 1 detects that an external interruption or a Wake-up interruption such as a timer interruption or the like has occurred.” Ex. 1007 (Shikata), 8:46-48.

129. Fig. 5 of Shikata depicts a timer counter 53 in the voltage controlling unit 1. Ex. 1007 (Shikata), 7:63-65. The timer counter 53 activates the voltage supply control circuit 54 in response to a trigger that is “an interruption signal INT from an exterior” or “a periodic interruption signal supplier from the timer counter 53, thereby returning from the standby state.” Ex. 1007 (Shikata), 7:66-8:2.

130. **Step S2:** At Step S2, the voltage supply to circuits in power supply block 32 is restored. Shikata discloses that “[a]t step S2, the voltage controlling unit 1 provides a control command to the external variable voltage supply 101, and, in response thereto, external variable voltage supply 101 resumes voltage supply to the CPU core unit 3 and the like.” Ex. 1007 (Shikata), 8:48-52.

131. **Step S3:** At Step S3, the processor 100 waits for the clock to stabilize before proceeding. Shikata discloses that “[a]t step S3, an internal reset signal is asserted, and a wait continues until the oscillation of the PLL circuit 11 is stabilized, followed by deactivating the internal reset signal once the oscillation of the PLL circuit 11 is stabilized.” Ex. 1007 (Shikata), 8:52-56.

132. **Step S4:** In Step S4, the processor 100 determines whether the system is recovering from a standby state and if so, it boots from the program stored in the internal RAM. “At step S4, the CPU core unit 3 initiates a bus access for fetching boot instructions. If the boot switch signal is asserted, a boot operation is performed from

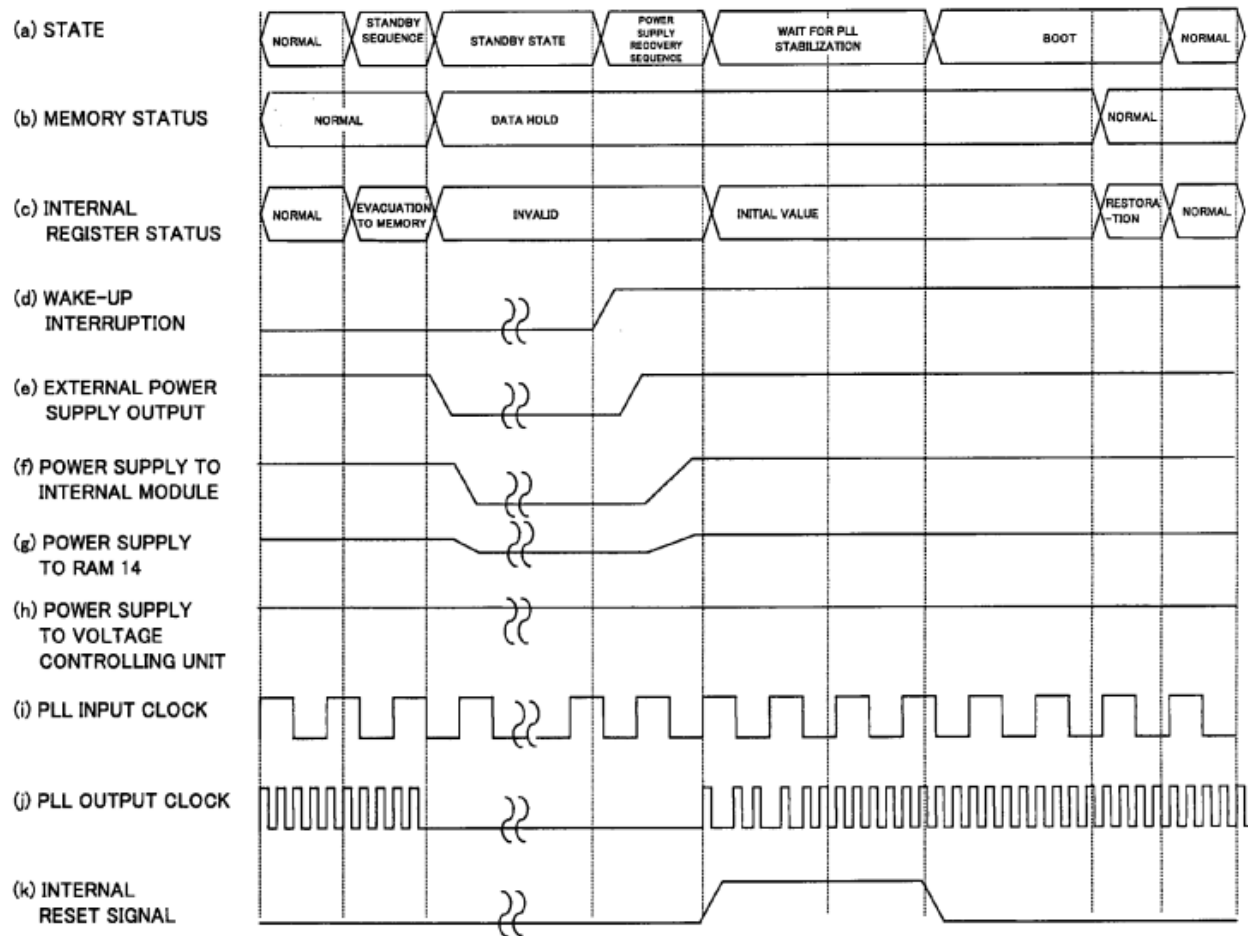
the internal RAM 14 connected to the on-chip bus 5 rather than from the external boot ROM 103.” Ex. 1007 (Shikata), 8:56-60.

133. **Step S5:** In Step S5, the context is restored based on information that was evacuated to the internal RAM 14. Shikata discloses that “[a]t step S5, the boot program stored in one area of the internal RAM 14 recovers the register values and the like saved to another area of the internal RAM 14.” Ex. 1007 (Shikata), 8:60-63.

134. **Step S6:** Finally in Step S6, the processor 100 resumes operation from the Program Counter when it transitioned to the standby state. “At step S6, the execution of the application program is resumed in a continuing fashion based on the recovered register data.” Ex. 1007 (Shikata), 8:63-65.

4. Timing Chart for Entering and Exiting the Standby State

135. Fig. 8 is a timing chart showing a transition to a standby state and a recovery from the standby state:



Ex. 1007 (Shikata), Fig. 8.

136. Shikata provides the following disclosure regarding the sequence shown in Fig. 8:

As shown in FIG. 8, a transition from a normal operation to a standby state is made through an intervening standby sequence (see (a)).

During the standby sequence, internal register values are saved to the memory (see (c)). In the standby state, the power supply output of the

external variable voltage supply **101** is suspended (see (e)), and power supply to internal modules comes to a stop (see (f)).

Even in this case, however, a reduced voltage continues to be supplied to the internal RAM **14** (see (g)), thereby continues the data hold operation (see (b)). Moreover, power supply is maintained with respect to the voltage controlling unit **1** (see (h)). In the standby state, the state of the internal registers becomes unstable (see (c)), and the clock output of the PLL circuit **11** is suspended (see (j)).

A Wake-up interruption then occurs (see (d)), and, in response, each power supply returns to a normal voltage level (see (e), (f), and (g)).

An internal reset signal is then asserted (see (k)) with a wait for PLL stabilization. The oscillation of the PLL circuit **11** is stabilized (see (j)). A boot operation is performed after the stabilization of the PLL (see (a)), and the saved register values are restored (see (c)).

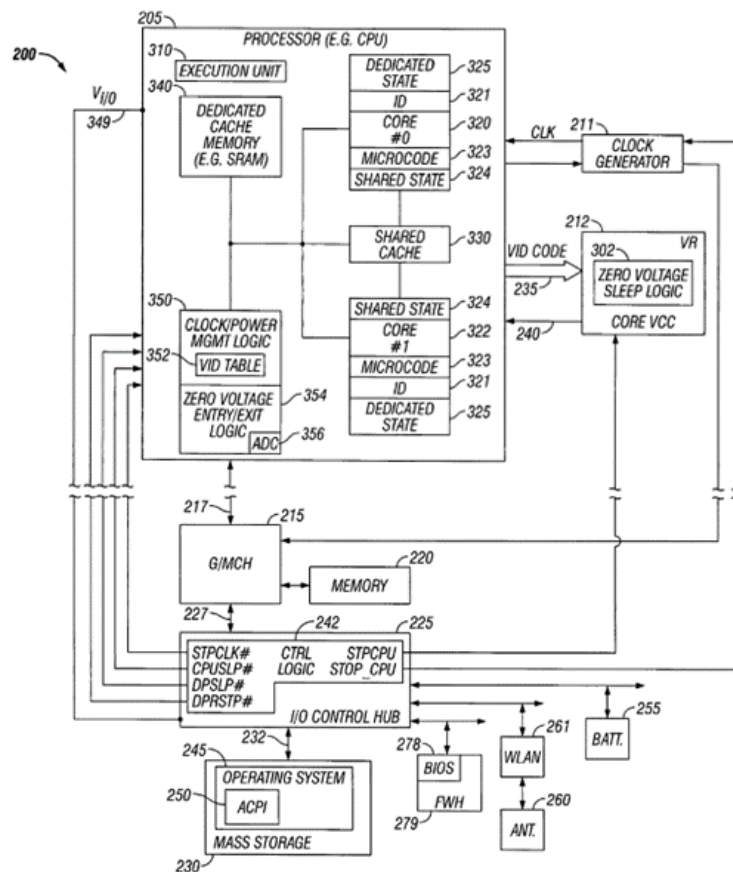
Ex. 1007 (Shikata), 8:67-1:21.

C. U.S. 7,664,970 (Jahagirdar)

137. Jahagirdar, entitled Method and Apparatus for a Zero Voltage Processor State, was filed on Dec. 30, 2005 and was issued as a patent on Feb. 16, 2010. As the title indicates, “[e]mbodiments of the invention relate to a method and apparatus for a zero voltage processor sleep state.” Ex. 1006 (Jahagirdar), Abstract.

138. Figs. 2A and 2B provide a block diagram of the system disclosed by Jahagirdar. Ex. 1006 (Jahagirdar), 2:36-38. As shown in Figs. 2A and 2B, the system

200 comprises, among other things, a processor (e.g., CPU) 205, a voltage regulator 212, an input/output (I/O) control hub 225, clock power management logic 350, zero voltage entry/exit logic 354, memory 220, and a clock generator 211.



Ex. 1006 (Jahagirdar), Figs. 2A-B.

139. When discussing transition into the zero voltage power management state in context of Fig. 1, Jahagirdar discloses that “[d]uring this transition, the critical state of the processor is saved (block 110). The critical state of the processor includes state variables associated with the architectural, micro-architectural, debug state, and/or

similar state variables associated with that processor. The operating voltage of the processor is subsequently reduced to approximately zero such that the processor is in a very deep sleep state that has very low power consumption characteristics (block 115).” Ex. 1006 (Jahagirdar), 3:13-21, Fig. 1.

140. When discussing transition out of the zero voltage power management state in context of Fig. 1, Jahagirdar mentions that “[s]ubsequently, in response to receiving a request to exit the zero voltage power management state, the processor exits the zero voltage power management at a higher reference operating voltage at block 120. The critical state variables associated with the processor are also restored (block 125).” Ex. 1006 (Jahagirdar), 3:24-28, Fig. 1.

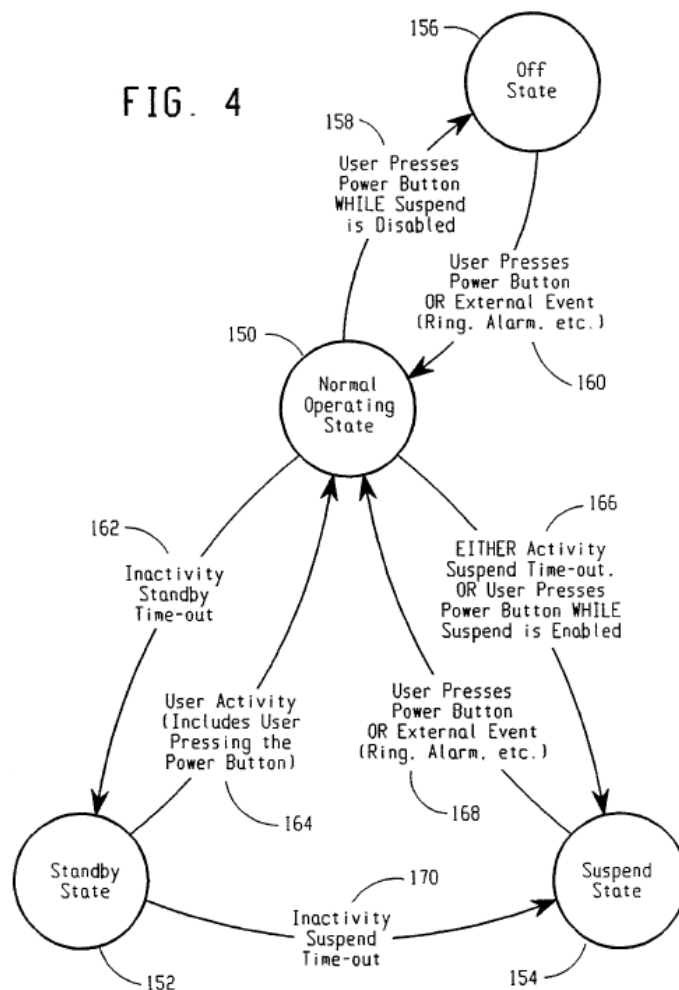
141. In addition, Jahagirdar discloses using microcode (e.g., a microprogram) to transition between various states of the processor 205, including to save and restore the critical state of the processor when transitioning into and out of the zero voltage power management state. Ex. 1006 (Jahagirdar), 10:18-38.

D. U.S. 5,530,879 (Crump)

142. Crump, entitled Integrated Circuit Chip Core Logic System Controller with Power Saving Features for a Microcomputer System, was filed on April 22, 1991 and issued as a patent on Feb. 14, 1995.

143. Crump discloses a computer system with a CPU, a power management

processor, a power supply and other associated circuit. Ex. 1008 (Crump), Abstract. The power supply has several power supply states that are controlled by the power management processor. Ex. 1008 (Crump), Fig. 4 of Crump is a state diagram depicting the four system states: normal, standby, suspend and off. Ex. 1008 (Crump), 5:43-44.



Ex. 1008 (Crump), Fig. 4, 9:54-61.

144. According to Crump, the *normal operating state* 150 is virtually identical

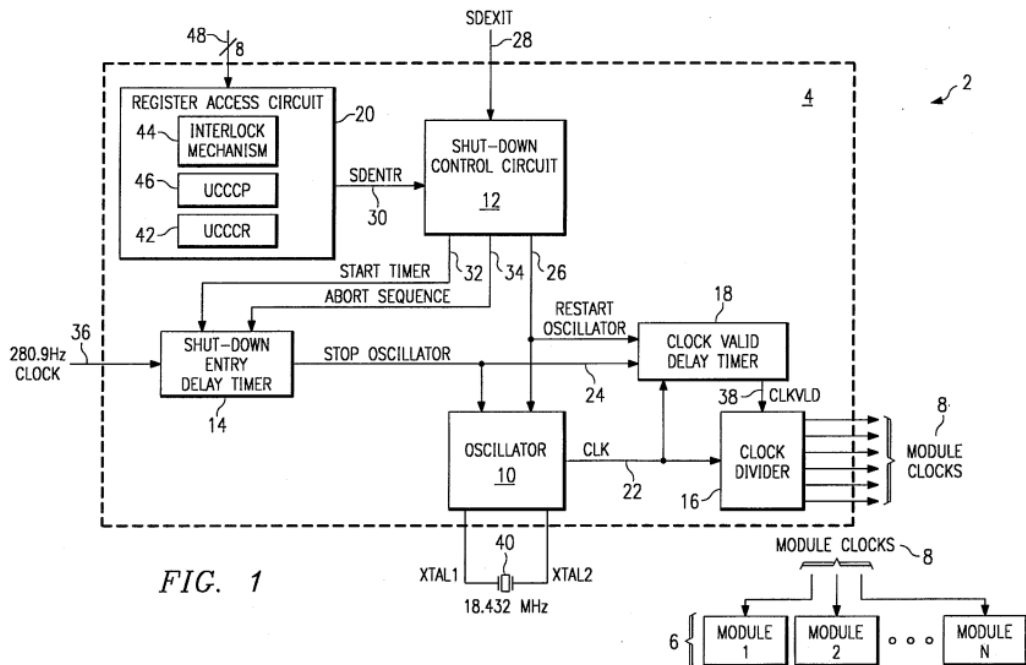
to the operating state of any computer. Ex. 1008 (Crump), 9:62-63. “The second state, the *standby state* 152, uses less electrical power than the normal operating state 150, yet leaves any applications executing as they would otherwise execute.” Ex. 1008 (Crump), 10:21-23. In the standby states 152, the CPU executes a HALT instruction, which significantly reduces power consumption. Ex. 1008 (Crump), 10:58-65. In the third state, the *suspend state* 154, the computer system consumes extremely small amount of electrical power. Ex. 1008 (Crump), 11:7-8. “To enter the suspend state 154, the CPU 40 interrupts any applications and transfers program execution control of the CPU to the power management driver.” Ex. 1008 (Crump), 11:18-20. “The power management driver ascertains the state of the computer system 10 and writes the state of the computer system to the fixed disk storage device 31. The state of the CPU 40 registers, the CPU cache 41, the system RAM 53, the system cache 60, the video controller 56 registers, the video memory 56, and the remaining volatile registers are all written to the fixed disk drive 31. The entire state of the system 10 is saved in such a way that it can be restored without the significant usability penalties.” Ex. 1008 (Crump), 11:21-29.

145. Crump teaches using microcode to save the state of the computer system. Ex. 1008 (Crump), 37:1-13, Fig. 10A.

E. U.S. 5,502,689 (Peterson)

146. Peterson, entitled Clock Generator Capable of Shut-Down Mode and Clock Generation Method, was filed on Feb. 24, 1994 and issued as a patent on March 26, 1996. The application that was filed with the USPTO is a continuation of an application that was filed on July 21, 1992 but was abandoned.

147. Peterson “relates to clock driven electrical systems and, more particularly, to electrical circuits for use in reducing power consumption of clock driven electrical systems.” Ex. 1010 (Peterson), 1:59-62. It introduces “a clock generator capable of being placed into a shut-down mode, during which the clock generator does not generate any clock signals.” Ex. 1010 (Peterson), 3:11-15; *see also id.*, 3:15-19. Peterson describes an interlock mechanism by which its clock generator disables the output clock signal upon receiving a shutdown entry request signal, provided that the signal meets a predetermined protocol requirement. Ex. 1010 (Peterson), 3:40-60; *see also id.*, 7:59-8:16. Fig. 1 of Peterson is a block diagram of the clock generator employing the interlock mechanism:



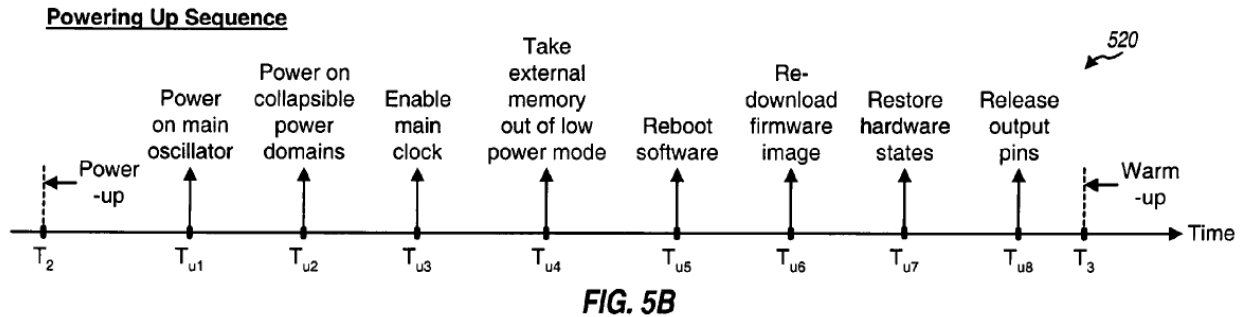
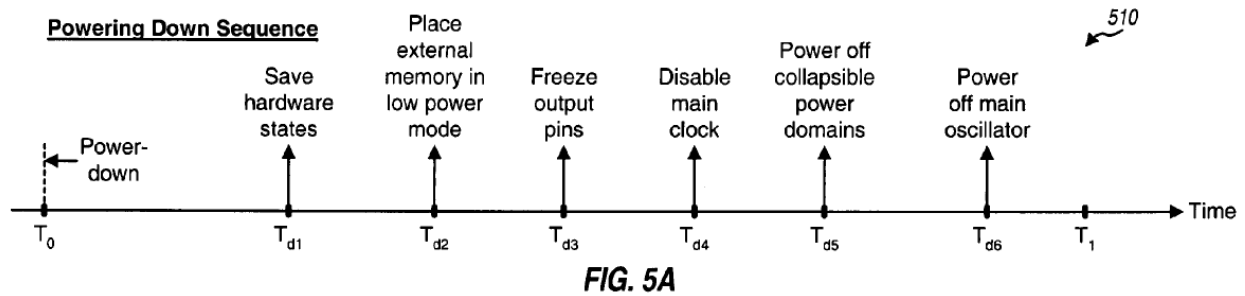
Ex. 1010 (Peterson), Fig. 1. Specifically, entry into shutdown mode “may be initiated by writing the appropriate data . . . to a shut-down control register, such as the UCCR register 42 shown in FIG. 1.” Ex. 1010 (Peterson), 7:59-61. Peterson discloses an “interlock mechanism,” which is an access control mechanism that protects write access to the UCCR register only upon the completion of a particular shut-down register access sequence. Ex. 1010 (Peterson), 7:65-8:17.

F. U.S. 7,369,815 (Kang)

148. Kang, entitled Power Collapse for a Wireless Terminal, was filed on Feb. 24, 2004 and issued as a patent on May 6, 2008. Kang states that the “present invention relates generally to circuits, and more specifically to techniques for conserving power for a wireless terminal.” Ex. 1011 (Kang), 1:12-14.

149. Kang discloses “[t]echniques for performing ‘power collapse’ for a wireless terminal.” Ex. 1011 (Kang), 1:65-66. “Power collapse refers to the powering down of circuit blocks/processing units when not needed to reduce leakage current and conserve power.” Ex. 1011 (Kang), 1:66-2:1. Kang also discloses saving context when powering down the power domains and then restores context when powering up. Ex. 1011 (Kang), 2:21-33 (“A set of tasks is typically performed for powering down the collapsible power domains. For example, the powering down tasks may include saving pertinent hardware registers of the collapsible power domains, . . . powering down a main oscillator and disabling a main clock from the oscillator, and so on. A complementary set of tasks is typically performed for powering up the collapsed power domains. For example, the powering up tasks may include powering up the main oscillator and enabling the main clock, restoring software, firmware, and hardware states, releasing input and output pins, and so on.”).

150. In Figs. 5A and 5B Kang also discloses the sequence of steps that are executed when entering and exiting the power down state:



Ex. 1011 (Kang), Figs. 5A, 5B.

151. As shown above, then executing the power down sequence, the context is saved, the memory is placed in low power mode, the main clock is disabled and then the power to certain blocks removing. Ex. 1011 (Kang), 5:47-60. A reverse sequence is followed when executing the power up sequence. The power is restored, the clocks are enabled, the memory is taken out of the retention mode and then context is restored. Ex. 1011 (Kang), 8:63-9:15.

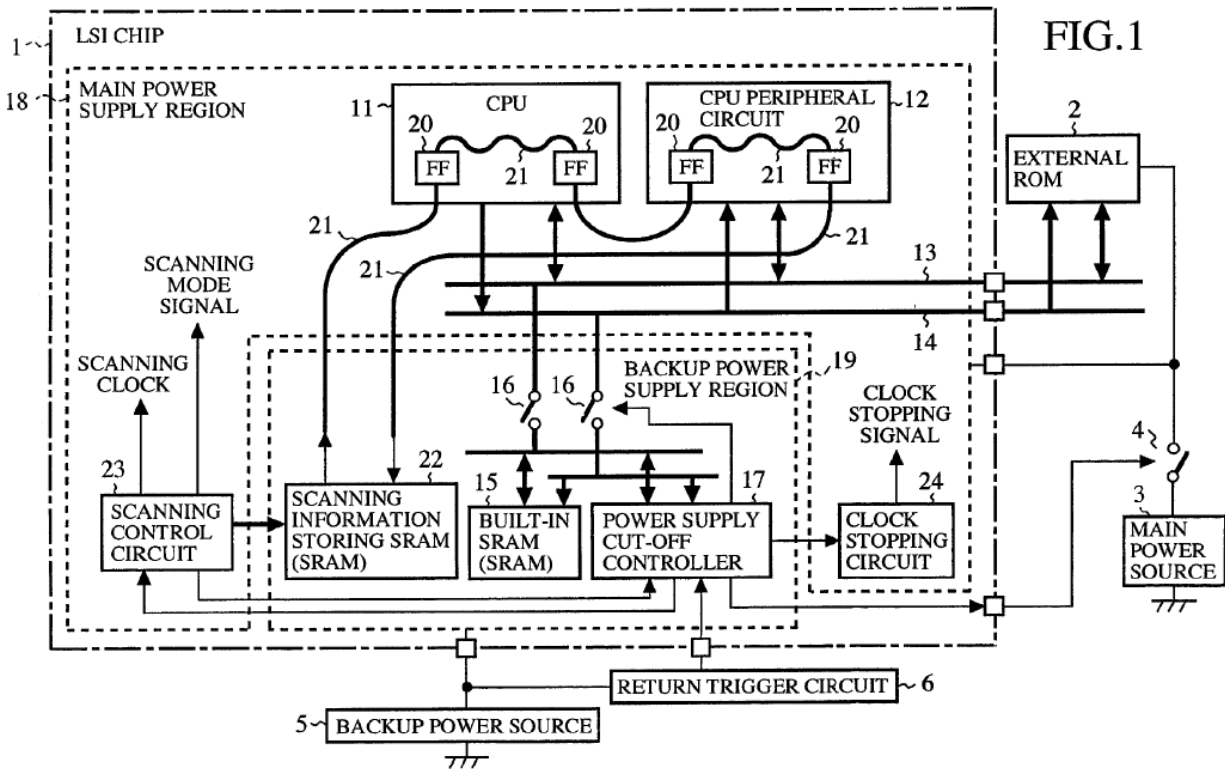
152. Because Kang is directed to wireless applications where terminals are expected to be awake at pre-fixed times, Kang discloses that “[p]ower collapse is typically performed in conjunction with a sleep timeline that indicates when the wireless terminal can go to sleep. The sleep timeline may be different for different

wireless communication systems. A power control unit within the always-on power domain powers down the collapsible power domains after going into sleep and powers up these domains just before waking up from sleep.” Ex. 1011 (Kang), 2:12-18.

G. U.S. 7,167,991 (Higashida)

153. Higashida, entitled Method for Reducing Leakage Current for LSI, was filed on July 10, 2001 and issued as a patent on January 23, 2007. The application that was filed with the USPTO claims priority to application filed in Japan on Dec. 26, 2000. Higashida discloses techniques for reducing power consumption by an LSI chip by placing the chip in an operation standby state. When entering the standby state, Higashida teaches saving the context and restoring it when exiting the standby state. Ex. 1009 (Higashida), Abstract. Higashida was cited by the Examiner during prosecution of the '575 Patent.

154. Fig. 1 of Higashida discloses a block diagram of an LSI chip that embodies the invention of the Higashida. The figure depicts that the LSI Chip 1 comprises, among other things, a CPU 11, a CPU peripheral circuit 12, built in SRAM 15, power supply cut-off controller 17, and a clock stopping circuit 24. Ex. 1009 (Higashida), 5:1-19.



Ex. 1009 (Higashida), Fig. 1.

155. As also depicted in Fig. 1, Higashida divides the LSI chip 1 into two power supply regions, the first being the Main Power Supply Region 18 that receives power from the Main Power Source 3 and the second being the Backup Power Supply Region 29 that receives power from the Backup Power Source 5. Ex. 1009 (Higashida), 5:60-6:30.

156. In context of Fig. 5 that embodies the fifth embodiment, Higashida discloses that in the “first embodiment, power was supplied to the backup power supply region 19 from the backup power source 5 having a voltage equal to that of the

main power source 3. However, power may be supplied from the main power source 3 during the normal operation, and power may be supplied from the backup power source 5 having a voltage lower than that of the main power source 3 in the operation standby state.” Ex. 1009 (Higashida), 12:58-65. Higashida relies upon the backup power source 5 to provide, during the standby mode, a lower voltage to the built-in SRAM than the normal operating voltage. Ex. 1009 (Higashida), 13:10-13 (“a voltage thereof is set lower than that of the main power source 3, enough for holding the content of the built-in SRAM provided in the backup power supply region 19.”)

H. U.S. 5,390,350 (Chung)

157. Chung, entitled Computer System Having Power Management Processor for Switching Power Supply From One State to Another Responsive to a Closure of a Switch, A Detected Ring or an Expiration of a Timer, was filed on Sept. 7, 1994, and issued as a patent on June 25, 1996. Chung discloses a system controller that “manages communication of data between a microprocessor, a memory and input/output controllers in a computer.” Ex. 1012 (Chung), Abstract. Chung also discloses a “power management controller” that operates “upon occurrence of microprocessor inactivity indications to disable power to the microprocessor without disabling power to the input/output controllers.” Ex. 1012 (Chung), Abstract.

158. Chung discloses saving the state of the CPU when entering the “Processor

Power Down (Sleep)” state where the CPU is powered down the CPU, and restoring the state of the CPU when entering the “Processor Power UP (Wake-up)” state. Specifically, Chung discloses that a “monitor program . . . runs in the environment of the operating system to detect when an inactive condition exists in the CPU.” Ex. 1012 (Chung), 31:52-54. If the CPU has been inactive for a predetermined length of time, the CPU executes a power-down sequence where the CPU “executes the register-saving program where the content of all registers and other status indications are saved in a predetermined area of RAM 30 for subsequent retrieval during the resume operation.” Ex. 1012 (Chung), 31:63-66. In addition, during the power-down sequence the system controller “tristates the output driver for the clock at CPUCLK 34 preventing clock pulses to the CPUs.” Ex. 1012 (Chung), 32:13-14.

159. Chung discloses that “[t]he resume or power up is initiated by an activity indication.” Ex. 1012 (Chung), 32:20-21. Chung provides a list of activity indications and states that “[w]hen any of these activity indications are detected, the system controller initiates a power resume operation.” Ex. 1012 (Chung), 32:27-28. During the power resume operation, power supply and clocks to the CPUs are restored. Ex. 1012 (Chung), 32:29-40. Upon exiting the power resume operation, the CPU “can now continue operation from the condition at which it was originally stopped and powered down.” Ex. 1012 (Chung), 35:15-19.

XI. SUMMARY OF OPINIONS

160. I summarize my opinions in the table below:

Grounds	Prior art combinations	Claims of '575 Patent
Ground I	Lee and Jahagirdar	1, 3, 5, 8, 11, 12, 15 and 16
Ground II	Lee, Jahagirdar and Peterson	2
Ground III	Lee, Jahagirdar and Kang	3
Ground IV	Lee, Jahagirdar and Higashida	11
Ground V	Shikata and Crump	1, 5, 11, 12 and 15
Ground VI	Shikata, Crump and Peterson	2
Ground VII	Shikata, Crump and Kang	3
Ground VIII	Shikata, Crump and Chung	16

XII. GROUNDS FOR CHALLENGING THE '575 PATENT

161. My analysis and conclusions in this Section are applicable to the Challenged Claims of the '575 Patent.

A. Ground I: Lee and Jahagirdar render obvious Claims 1, 3, 5, 8, 11, 12, 15, and 16

162. As further described below, Lee in view of Jahagirdar renders obvious Claims 1, 3, 5, 8, 11, 12, 15, and 16 of the '575 Patent.

1. Motivation to Combine Lee and Jahagirdar

163. A POSITA when seeking to implement the SOC 110 disclosed in Lee would realize that Lee assumes that a POSITA would be aware of the basic principles of electrical engineering and computer science, and thus does not disclose routine implementation details. Lee assumes that a POSITA would know how to implement the following features needed to fully implement the SOC: (i) Lee does not disclose a clock generator and supply of clocks to various circuit blocks in the SOC because a POSITA would know how to implement an indispensable component of digital design; and (ii) while Lee discloses saving and restoring working context when the power supply to the CPU is suspended and activated respectively, Lee does not provide the implementation details for such functionality.

164. A POSITA would be motivated to supplement the disclosure of Lee based on the disclosure of a reference, such as Jahagirdar. A POSITA would look to Jahagirdar because Jahagirdar is analogous art and has the same purpose as Lee. As with Lee, Jahagirdar aims to reduce power consumption in a microprocessor based system. Jahagirdar teaches placing the microprocessor in an approximately zero voltage state to conserve power when the microprocessor is idle. In addition, Jahagirdar discloses saving context of the microprocessor when it is placed in the approximately zero voltage state and restoring the context when the power is supplied

again to the microprocessor. Jahagirdar explains that saving and restoring context is more efficient than not saving context as it enables the microprocessor to restart the execution from the location before the shutdown.

165. Based on the disclosure of Jahagirdar and the knowledge of a POSITA, the POSITA would borrow the following well-known techniques from Jahagirdar to augment the invention disclosed in Lee: first, use of a clock generator to provide clocks and to disable and enable clocks when entering and exiting the standby mode, and second, use of a microprogram to save and restore working context when entering and exiting the standby mode.

a) Use of a clock generator

166. A POSITA would incorporate the clock generator 211 depicted in Figs. 2A-B of Jahagirdar in the SOC 110 disclosed in Fig. 1 of Lee, for the reasons described in **Section VII.B.2** (discussing clocks) above. The clock generator 211 would play the same role in Lee as it does in Jahagirdar. In Jahagirdar, the clock generator 211 supplies clock to processor 205, and the supply of clock is inhibited when the processor 205 enters the zero voltage standby state and the clock supply is restored when the processor exits that mode. When clock generator 211 is integrated in Lee, the clock generator would provide clocks to various circuit components in the SOC 110, and the supply of the clock would be deactivated when the SOC enters the power-off shutdown

state and the supply of the clock would be activated when the SOC 110 exits the power-off shutdown state.

167. In Jahagirdar, the activation and deactivation of the supply of clock from clock generator 211 to processor 205 works in tandem with the supply of voltage to processor 205. In other words, Jahagirdar discloses disabling both supply of clock and voltage to processor 211 when the processor is entering the zero voltage state. Likewise, both are restored when the processor 211 exits the zero voltage state. Thus, a POSITA would implement the same scheme in Lee—disable supply of clock from the clock generator to microprocessor 10 and peripheral modules 30 and 40 when entering the power-off shutdown state and enabling the supply of clock when exiting that state. Because the power controller 80 in SOC 110 disclosed in Lee controls the supply of power to various circuit components in the SOC 11, a POSITA would conclude that it would be a good design choice to augment the functionality of the power controller 80 to also control the supply of clock to these circuit components.

168. In addition, a POSITA would integrate the clock generator of Jahagirdar with the SOC 110 (system-on-chip) of Lee instead of implementing the clock generator as a separate integrated chip. A POSITA would be aware of the benefits of an integrated approach, as described in **Section VII.B.5** (discussing benefits of integration) above, and such integration was well within the capability of a POSITA

by the time of the invention of the '575 Patent.

b) Use of a microprogram

169. While Lee discloses that the microprocessor 10 in SOC 110 saves and restores working context when transitioning into and out of the power-off shutdown state, it assumes that it would be within a POSITA's knowledge to implement the save and restore functionality.

170. For the reasons described in **Section VII.B.4** (use of microprograms) above, a POSITA would be motivated to implement the save and restore functionality based on the disclosure of Jahagirdar because Jahagirdar uses microcode (or a microprogram) to save and restore context in its design. Jahagirdar discloses that the microcode in processor 205 controls transitions between various states of the processor and is aware of which registers need to be saved to preserve the architectural state of the processor 205. Because Jahagirdar explicitly states that the microcode saves the architectural state of the processor 205, a POSITA would understand that Jahagirdar employs microcode to restore context too.

171. A POSITA would be aware that various options were available at the time of the invention of the '575 Patent to implement the save and restore working context, and the POSITA would have been aware of the costs and benefits of each option. A POSITA would have known the working context could be saved and restored using

hardware, microprogram, or program. In my opinion, a POSITA would know that the functionality could be implemented using a program or a microcode or a microprogram that runs at the lowest level. **Section VII.B.4** (discussing microprograms) above explains that a POSITA would have appreciated the benefits a microprogram offers over a program or a hardware technique to save and restore context. For the reasons mentioned in that section, a POSITA would have been motivated to follow the approach described in Jahagirdar as opposed to other alternatives.

172. Finally, Lee discloses that the microprocessor 10 notifies the power controller 80 when the working context from the microprocessor 10 and peripheral units 30 and 40 has been saved. A POSITA would understand that it would be a good design choice to incorporate that functionality into the microcode that implements the save and restore. *See Section VII.B.8* above (discussing interprocess communication). In my opinion, a POSITA would appreciate that when the microprogram is finished saving the working context, then it would be efficient for the microprocessor 10 to notify that the evacuation is complete and the system can proceed to the next step.

2. Combination of Lee and Jahagirdar would be a routine exercise for a POSITA with a reasonable expectation of success

173. As discussed in **Sections VII.B.2, VII.B.4, and VII.B.5** above, use of clocks in digital design and use of microprograms (or microcode) was well known and

well understood by the time of the invention of the '575 Patent. As a result, modification of SOC 110 in Lee to incorporate clocks and use of microprograms to save and restore context, as disclosed in Jahagirdar, would be a combination of known elements in prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of SOC 110 disclosed in Lee would be based on applying known techniques (e.g., clocks and microprograms) to a known device (e.g., the SOC 110) to improve the device. As discussed in **Section VII.B.2** (discussing clock) above, use of clocks is indispensable in digital designs, such as the SOC 110 of Lee. In addition, **Section VII.B.4** (discussing microprograms) above discusses that approaches were available to a POSITA to save and restore context but the use of microprograms was the preferred approach. A POSITA would be motivated to use microprograms to implement Lee as opposed to other approaches to save and restore context. Thus, a POSITA would be motivated by the knowledge of the skilled artisan to choose from a finite number of identifiable and predictable solutions. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Jahagirdar to implement and improve the SOC 110 disclosed by Lee. In digital systems like Lee's, the inclusion of a clock generator is critical for proper operation. Furthermore, employing microcode is a routine procedure for implementing saving and restoring functions.

3. Claim 1

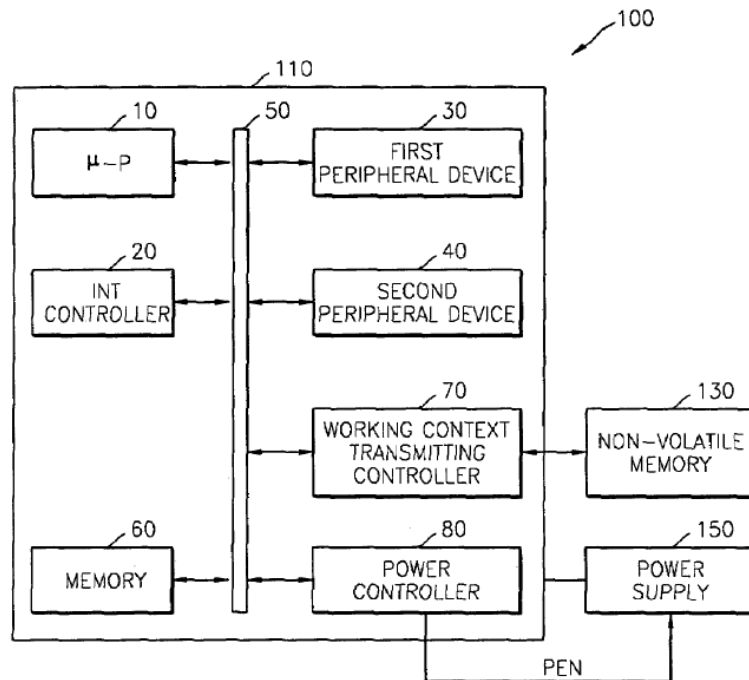
174. Lee in view of Jahagirdar renders Claim 1 obvious.

[1pre] A microcontroller comprising:

175. I have reviewed **Section IX.A.2** of the Petition addressing the preamble of Claim 1. *See also Section IX.A* above regarding claim construction.

a) [1a] a CPU;

176. I have reviewed **Section IX.A.2.b** of the Petition. Fig. 1 of Lee that depicts a system-on-chip (SOC) 110 excerpted below:



Ex. 1005 (Lee), Fig. 1.

177. A POSITA would understand that the microprocessor (μ -P) 10 is the

claimed CPU, and is also referred to as a central processing unit.

- b) **[1b] a power supply unit arranged between the CPU and a power supply device for supplying power to the CPU;**

178. I have reviewed **Section IX.A.2.b** of the Petition. A POSITA would understand that the power supply 150 is connected to the microprocessor 10 within the SOC 110 because power is necessary for its operation. Moreover, a POSITA would understand that the power supply 150 is connected to a power source providing AC or DC power, given that an on-chip regulator within an SOC requires a power source in order to deliver power to the microprocessor. Such a power source corresponds to the claimed power supply device, which the parties agree means “power supply.” It is my opinion that POSITA would understand that the power supply 150 (i.e., the claimed power supply unit) is arranged between the microprocessor 10 (i.e., the claimed CPU) and the power source (i.e., the claimed power supply device). *See* Ex. 1005 (Lee), 5:35-40, 48-50. *See also* **Section IX** above regarding claim construction.

- c) **[1c] a power supply control unit for controlling the CPU and the power supply unit; and**

179. I have reviewed **Section IX.A.2.c** of the Petition.

- d) **[1d] an information holding unit for holding information evacuated from the CPU, the information being necessary in proceeding with a program; and**

180. I have reviewed **Section IX.A.2.d** of the Petition. In my opinion, a

POSITA would understand that the working context of a microprocessor 10 constitutes the information that is necessary in processing with a program that was running on the microprocessor 10 when it received the predetermined interrupt signal.

e) [1e] a clock generator; wherein:

181. I have reviewed **Section IX.A.2.e** of the Petition. For the reasons described above in **Sections XII.A.1 and XII.A.2** and in **Section VII.B.2** (discussing use of clocks) and **VII.B.5** (discussing benefits of integration), it is my opinion that a POSITA would implement the clock generator 211 disclosed in Jahagirdar on the same semiconductor die that contains the SOC 110 of Lee, and not implement the clock generation 211 on a separate chip. As I explained in the Sections cited above, higher integration of circuit components would result in better performance, lower power consumption and lower costs. There would have been a reasonable expectation of success in implementing this change, as it was routine practice to integrate a clock generator within a single-die system. *See Sections VII.B.2* (discussing use of clocks) and **VII.B.5** (discussing benefits of integration) above; *see also* Ex-1051 (Ozawa), Fig. 1 (depicting a CPU and a clock generation circuit CPG on the same chip); Ex-1052 (Totsuka), Fig. 14 (depicting a processor chip 901 containing a clock generating circuit 908).

- f) **[1f] the power supply control unit outputs a shutdown request signal to the CPU in response to an occurrence of a power shutdown factor,**

182. I have reviewed **Section IX.A.2.f** of the Petition.

- g) **[1g] the CPU, upon receiving the shutdown request signal, executes a power shutdown microprogram, evacuates the information necessary in proceeding with the program to the information holding unit, and outputs an evacuation completed signal to the power supply control unit after evacuation is completed,**

183. I have reviewed **Section IX.A.2.g** of the Petition. It is my opinion that a POSITA would understand that transferring the various working context to memory 60 (as disclosed in Lee) would result in transferring the claimed information necessary to proceed with a program when the microprocessor 10 eventually exits the power-off shutdown mode. *See Section VII.B.1* (discussing architectural state of a processor) above. Though Lee mentions that the microprocessor 10 saves its working context and the working context of the peripheral devices 30 and 40 to memory 60 through the on-chip bus 50, it does not provide any specifics regarding the transfer of the various working context to memory 60. As I have discussed in **Section VII.B.4** (discussing microprograms) above, a POSITA at the time of the invention of the '575 Patent would have been aware of several techniques to transfer the abovementioned working context.

184. Jahagirdar discloses using a microprogram (or microcode) to transfer

working context from processor to memory so that the context can be restored later. See **Section X.C** (discussing Jahagirdar) above. Fig. 1 of Jahagirdar “is a flow diagram illustrating a process to transition into and transition out of a zero voltage power management state for a processor.” Ex. 1006 (Jahagirdar), 2:32-35. Fig. 1 shows that when transitioning into the zero voltage state, “the critical state of the processor is saved (block 110). The critical state of the processor includes state variables associated with the architectural, micro-architectural, debug state, and/or similar state variables associated with that processor.” Ex. 1006 (Jahagirdar), 3:14-21. Fig. 4 is a flow diagram illustrating the process to enter the zero voltage processor sleep state. Ex. 1006 (Jahagirdar), 3:43-45. Fig. 4 also illustrates that “each core independently performs a state save when the zero voltage processor sleep state is initiated.” Ex. 1006 (Jahagirdar), 9:64-65.

185. Jahagirdar discloses that *microcode* is employed to transition between various processor states, including the zero voltage processor sleep state. Ex. 1006 (Jahagirdar), 9:40-48. Jahagirdar states that:

It should be noted that the *microcode* 323 of the CPU cores 320 and 322 may generally need to know which control registers need to be saved and restored for the zero voltage processor sleep state. The list of the registers may be a subset of the total registers on the CPU 205. Ex. 1006 (Jahagirdar), 10:18-22.

186. Jahagirdar further discloses that after the microcode operations have been performed, the zero voltage entry/exit logic 354 of the clock/power management logic 350 takes over the data flows. Ex. 1006 (Jahagirdar), 10:32-35.

187. The microcode disclosed in Jahagirdar would meet the claim construction of the term “microprogram” proposed by Qualcomm. *See Section IX.B* above. As Collabo itself has pointed out, a POSITA would understand that microcode and microprogram are synonyms. Ex. 1014 (Jahagirdar POPR), 30 and Ex. 1017 (Chung POPR), 27. The microcode disclosed in Jahagirdar (when implemented in the SOC 110 disclosed in Lee) would reside within the CPU to ensure that the microcode can be accessed and then executed rapidly, as opposed to a program that would reside in memory 60. Fig. 2A depicts that the microcode 323 is stored within the processor (e.g., CPU) 205 disclosed in Jahagirdar. Ex. 1006 (Jahagirdar), Fig. 2A. Like any microcode, the microcode in Jahagirdar runs at the lowest level of the hardware and controls the internal resources of the CPU, in addition to saving and restoring context.

188. Lee discloses that the microprocessor 10 notifies the power controller 80 when the working context has been saved. A POSITA would understand that such a communication between circuit blocks is handled through various mechanism, such as interrupts or any other suitable mechanism. *See Section VII.B.8* above (discussing inter-block communication between circuit blocks). For the reasons I discussed in

Sections XII.A.1 and XII.A.2, and Sections VII.B.4 (discussing microprograms) and **VII.B.8** (discussing inter-block communication between circuit blocks) above, it is my opinion that a POSITA would ensure that when the microprogram has evacuated the working context in SOC 110, the microprogram would cause the microprocessor 10 to send an interrupt to the power controller 80 notifying that the evacuation of the working context has been accomplished. Such a design implementation would be efficient and in accord with sound engineering principles known to a POSITA at the time of the invention of the '575 Patent.

189. Lee discloses that after the working context has been saved in memory 60, either the microprocessor 10 or the WCTC 70 transfers the working context from the memory 60 to non-volatile memory 130. The fact that Lee saves working context in memory 60 and then transmits the context to off-chip non-volatile memory 130 is immaterial to my analysis in the Declaration. Claim 1 of the '575 Patent is a comprising claim. I understand that for infringement purposes, it does not matter whether accused devices perform additional steps or has additional features that are not recited by the claim-at-issue. Likewise for invalidity, it does not matter whether the prior art performs additional steps or has additional features not required by the claim-at-issue.

- h) [1h] the power supply control unit, upon receiving the evacuation completed signal, outputs a power shutdown control signal to the power supply unit,**

190. I have reviewed **Section IX.A.2.h** of the Petition.

- i) **[1i] the power supply unit shuts down power supply to the CPU upon receiving the power shutdown control signal from the power supply control unit,**

191. I have reviewed **Section IX.A.2.i** of the Petition.

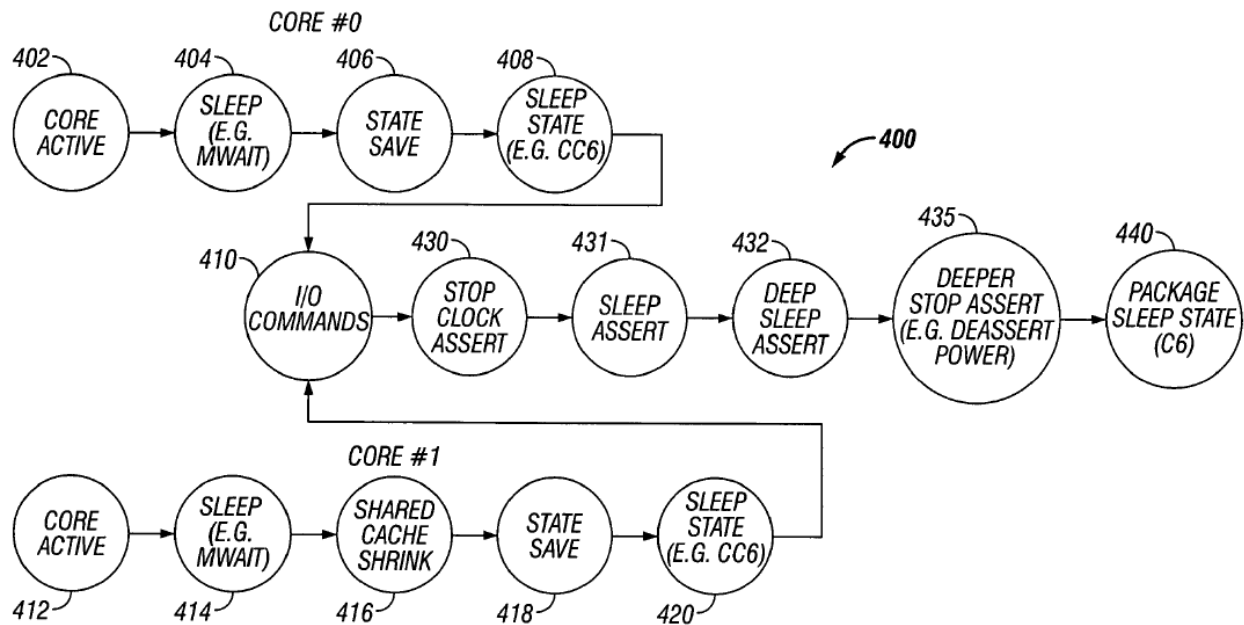
- j) **[1j] the power supply control unit, upon receiving the evacuation completed signal, outputs a clock stop control signal to the clock generator,**

192. I have reviewed **Section IX.A.2.j** of the Petition.

193. Jahagirdar discloses that “a STOP_CPU signal may be asserted by the input/output controller 225 and received by the clock generator 211 to cause the *clock generator to halt the clock signal CLK to the CPU 205.*” Ex. 1006 (Jahagirdar), 7:19-22. Thus, in my opinion, the STOP_CPU signal in Jahagirdar corresponds to the claimed clock stop control signal as it causes the clock generator to stop the clock signal to the CPU. Fig. 4 in Jahagirdar is a flow chart depicting entry into the zero voltage processor sleep state. Ex. 1006 (Jahagirdar), 2:43-45. Jahagirdar discloses that the processor 205 may transition into the Deep Sleep state. Ex. 1006 (Jahagirdar), 7:12-14. “In the Deep Sleep state, in addition to powering down internal processor circuitry, all phase-lock loops (PLLs) in the processor 205 may be disabled.” Ex. 1006 (Jahagirdar), 7:14-18.

194. When discussing Fig. 4, Jahagirdar discloses that “a stop clock signal

may be asserted at circle 430 (e.g., STPCLK#)” followed by other commands. Ex. 1006 (Jahagirdar), 10:48-50. “[T]he CPU 205 responds by shutting off its internal clock distribution and then the PLLs.” Ex. 1006 (Jahagirdar), 10:53-54.



Ex. 1006 (Jahagirdar), Fig. 4.

195. A POSITA would understand that when entering the zero voltage power management state, the PLLs and internal clock distribution of the CPU 205 would be shut off, and then the clock signal CLK from the clock generator 211 to the CPU 205 will be halted.

196. In light of the disclosure of Jahagirdar of how the STOP_CPU signal halts the supply of the clock to the CPU 205, it is my opinion that a POSITA would augment the power controller 80 disclosed in Lee to instruct the clock generator to stop

supplying the clock to the microprocessor 10. Such a signal would correspond to the claimed clock stop control signal. Because the supply to power to the microprocessor 10 is stopped in tandem with disabling supply of clock, POSITA would conclude that it is sound design practice to implement the functionality that disables clock in the power controller 80 instead of another circuit block in the SOC 110. As I discussed in **Section VII.B.2** (discussing clocks), implementing clock generator and integrating the use of clocks in digital designs was routine and well-understood design principle by the time of the invention of the '575 Patent.

- k) **[1k] the power supply control unit outputs a power supply control signal to the power supply unit in response to an occurrence of a power supply restoration factor,**

197. I have reviewed **Section IX.A.2.k** of the Petition.

- l) **[1l] the power supply unit, upon receiving the power supply control signal, starts to supply power to the CPU,**

198. I have reviewed **Section IX.A.2.l** of the Petition.

- m) **[1m] the power supply control unit outputs a restoration request signal to the CPU and outputs a clock supply control signal to the clock generator when a supplied power supply voltage is stabilized,**

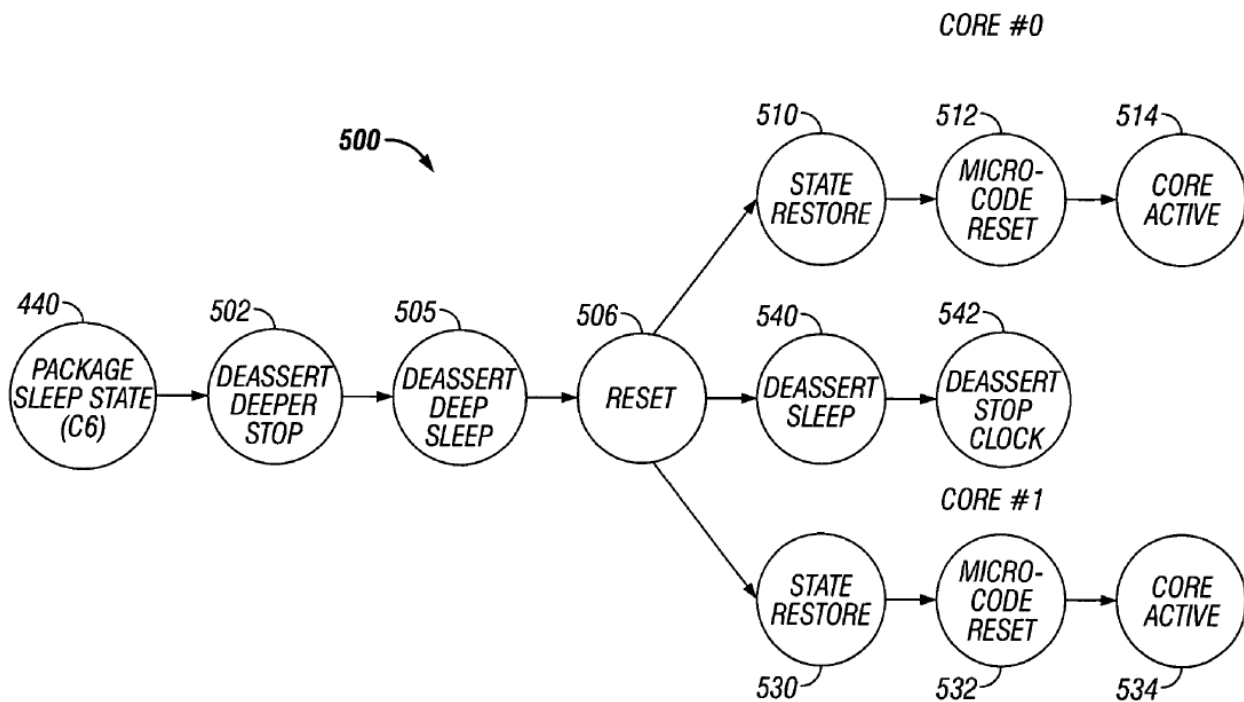
199. I have reviewed **Section IX.A.2.m** of the Petition.

200. Fig. 5 of Jahagirdar is a flow diagram illustrating the process of exiting

the zero voltage processor sleep state. Ex. 1006 (Jahagirdar), 2:46-48. Jahagirdar describes an exit sequence that “may be considered the reverse of what happened during the previously-described entry phase.” Ex. 1006 (Jahagirdar), 11:28-21. Thus, in my opinion Jahagirdar discloses restoring power supply to the CPU 205 and then restoring the supply of the clock to the CPU 205.

201. Mirroring Lee, Jahagirdar first restores the power supply when exiting the zero voltage power management state. As discussed in context of circle 502 of Fig. 5 in Jahagirdar, the control hub de-asserts the deeper stop signal (DPRSTP#), and a VID signal is sent to the voltage regulator 212 to instruct the “voltage regulator 212 to drive the core operational voltage back up to the required VID.” Ex. 1006 (Jahagirdar), 11:37-40.

202. With respect to circle 505, Jahagirdar mentions that “[a]t a predetermined time (e.g. controlled by timers in the ICH 225), *a signal to assert the clocks back on is asserted* and the deep sleep (DPSLP#) signal is de-asserted (circle 505), which initiates the PLLs of the clock/power management logic 350.” Ex. 1006 (Jahagirdar), 11:41-45. A POSITA would understand that after the “*signal to assert the clocks back on*” refers to the STOP_CPU signal that instructs the clock generator 211 to restart the supply of clock signal CLK to the CPU 205 during the exit phase.



Ex. 1006 (Jahagirdar), Fig. 5.

203. Jahagirdar discloses using an analog-to-digital converter (ADC) “as part of the zero voltage entry/exit logic 350 to monitor a voltage supply level and provide an associated digital output.” Ex. 1006 (Jahagirdar), 4:58-61. In my opinion, a POSITA would understand that the use of timer and ADC to monitor voltage supply level ensures that the Jahagirdar’s system waits for the voltage supply levels to stabilize before advancing from circle 502 to circle 505 in Fig. 5. Moreover, a POSITA would expect the clock restoration signal would issue before the CPU restoration signal is issued because a CPU cannot function without a clock. As discussed in **Section X.F** (discussing Kang) above, Kang also discloses the same sequence.

204. It is my opinion that when augmenting the power controller 80 of Lee to manage supply of clock to the microprocessor 110, a POSITA would incorporate the functionality disclosed by Jahagirdar. The POSITA would ensure that the power controller 80 waits for the recently-restored power supply to stabilize before restoring the clock supply to the microprocessor. The reset signal generation circuit in Lee would issue the restoration request signal to the microprocessor 10 after restoring the clock supply. In addition, as I explained in **Section VII.B.3** (discussing voltage and clock scaling) above, POSITA would know that it is good design practice to permit voltage levels to settle when voltage level is raised from 0 to operating level (e.g., 3.3 V or 5 V) to ensure reliable operation of circuits.

n) **[1n] the clock generator, upon receiving the clock supply control signal, restarts to supply an operation clock, and**

205. I have reviewed **Section IX.A.2.n** of the Petition.

206. Fig. 5 in Jahagirdar is a flow diagram illustrating the exit sequence from the zero voltage processor sleep state. Ex. 1006 (Jahagirdar), 5:46-48. Jahagirdar states that “[a]t a predetermined time (e.g. controlled by timers in the ICH 225), a signal to assert the clocks back on is asserted and the deep sleep (DPSLP#) signal is de-asserted (circle 505), which initiates the PLLs of the clock/power management logic 350. After this, CPU 205 initiates an internal RESET (circle 506). After this reset is complete,

CPU 205 has the power and clocks engaged and is ready to restore the critical state variables associated with the CPU 205.” Ex. 1006 (Jahagirdar), 11:41-48.

207. In the context of Fig. 5, Jahagirdar states that “once the stop clock is deasserted and the sleep signal is deasserted, both the CPU 205 and cores 320 and 322 and the chipset are all powered on and are operational, and both cores become active (circle 514 and circle 534).” Ex. 1006 (Jahagirdar), 11:66-12:2. It is my opinion that a POSITA would know that because circuits are “powered on and are operational,” the clock circuits must be turned ON because without such a clock restoration a circuit cannot be operational. *See Section VII.B.2* (discussing clocks) above.

208. It is my opinion that when augmenting the invention disclosed in Lee based on the teachings of Jahagirdar that “a signal to assert the clocks back on is asserted,” Ex. 1006 (Jahagirdar), 11:42, a POSITA would ensure that the clock generator in Lee restarts the clock supply to the microprocessor 10 upon receiving an instruction (i.e., the claimed clock supply control signal) from the power controller 80. *See also Sections VII.B.2* (discussing clocks) and *VII.B.3* (discussing voltage and clock scaling) above.

- o) **[1o] the CPU, upon receiving the restoration request signal, executes a power supply restoration microprogram, restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown,**

and then branches a process to an address indicated by a program counter to continue program execution from a shutdown state.

209. I have reviewed **Section IX.A.2.o** of the Petition.

210. Because Lee discloses that in response to the reset release signal, the microprocessor 10 “recovers its own working context,” Ex. 1005 (Lee), 9:64, a POSITA would understand that as a result of recovering the context, the internal registers of the microprocessor 10, including the Program Counter (PC), that were saved earlier are restored to their prior values. A program counter is a register within the microprocessor that determines which program instruction should be executed next. It is my opinion that a POSITA would understand that the CPU will continue execution from the restored Program Counter, i.e., continue execution of the program from the shutdown state, as required by the claim. A POSITA would further understand that restoring the Program Counter to the saved value in order to continue execution from the Program Counter constitutes “branching the process to a value indicated by a program counter,” as required by the claims. As discussed in **Section VII.B.1** (discussing architectural state of a processor) above, a POSITA would understand that the Program Counter (PC) dictates which instruction a CPU fetches and executes.

211. As discussed in **Section XII.A.3.g** above, Jahagirdar discloses using a microprogram (or microcode) to save and restore context. Jahagirdar discloses that

when exiting from the zero voltage state, “CPU 205 may perform an internal RESET to clear states, and may then restore the state of the CPU 205 from the dedicated sleep state SRAM cache 340, and CPU 205 continues from where it left off in the execution stream.” Ex. 1006 (Jahagirdar), 8:27-31. In context of Fig. 5, Jahagirdar states that “[o]nce the states of the cores 320 and 322 are restored, CPU 205 is ready to continue from where it stopped.” Ex. 1006 (Jahagirdar), 11:58-59.

212. It is my opinion that a POSITA would understand that because Jahagirdar employs microcode to transition between states, including to save context, Jahagirdar teaches employing microcode to restore context when returning from the zero voltage state. For the reasons discussed in **Section VII.B.4** (discussing microprograms) above, a POSITA would implement the functionality of restoring working context from memory 60 to microprocessor 10 or peripheral devices 30 and 40 using a microprogram or a microcode as opposed to other hardware-based or program-based options available to a POSITA at the time of the invention of the '575 Patent.

4. Claim 3

213. Claim 3 provides: “*The microcontroller according to claim 1, wherein the power supply control unit limits a period in which shutdown is possible.*”

214. I have reviewed **Section IX.A.3** of the Petition.

215. In the context of Fig. 4, Jahagirdar discloses that “each core 320 and 322

may execute an MWAIT instruction, and the initializing CPU core goes into a waiting state (e.g. CC6) and waits for the other core to get into the CC6 state as well, before the whole package (e.g. including both cores 320 and 322) may transition into what may be termed the package C6 sleep state.” Ex. 1006 (Jahagirdar), 9:56-61. Thus, in my opinion, Jahagirdar teaches limiting the period in which shutdown is possible.

216. It is my opinion that a POSITA would be motivated to modify Lee to incorporate this capability in the SOC 110. While Jahagirdar discloses the above technique when two processor cores are transitioning to sleep state, a POSITA would appreciate that the SOC 110 in Lee will benefit from this technique. Lee discloses a microprocessor 10 and at least two peripheral modules 30 and 40. *See* Ex. 1005 (Lee), Fig. 1. In my opinion, a POSITA would understand that each of the two peripheral modules may each contain a processor core. Irrespective of that, when the microprocessor 10 and the peripheral modules 30 and 40 are entering the power-off shutdown state, the SOC 110 would benefit if the SOC 110 limits the period in which shutdown is possible in the same manner as disclosed in Jahagirdar. Because the power controller 80 in SOC 110 controls the entry into and out of the power-off shutdown mode, the POSITA would modify the power controller 80 (i.e., claimed power supply control unit) to limit a period in which shutdown is possible, preventing system malfunctions. A POSITA would recognize that the power supplied to the

microprocessor should remain on if a peripheral module is actively engaged and requires input from the microprocessor.

217. It is my opinion that a POSITA would have a reasonable expectation of success when modifying Lee's systems in view of the disclosure in Jahagirdar because such a modification would involve combination of known components in a known manner. At the time of the invention of the '575 Patent, it was routine and well-understood how to design a component like Lee's power controller to discontinue power without causing system malfunctions.

5. Claim 5

218. Claim 5 provides: "*The microcontroller according to claim 1, wherein the power supply restoration factor uses an interrupt system of the microcontroller.*"

219. I have reviewed **Section IX.A.5** of the Petition.

6. Claim 8

220. Claim 8 provides: "*The microcontroller according to claim 1, wherein one of the power shutdown microprogram and the power supply restoration microprogram is selectively executed in an alternative way, according to the generated power shutdown factor or a type of the selected power shutdown factor.*"

221. In my opinion that Claim 8, as drafted, is indefinite at least because neither the claim nor the specification explain how one of the two microprograms are

“selectively executed in an alternative way.” In addition, Claim 8 does not provide an antecedent basis for either the “generated power shutdown factor” or “a type of the selected power shutdown factor.”

222. However, I understand that Collabo asserts that the specification of the '575 Patent clarifies the scope of Claim 8 and that the claim is not indefinite. In particular, Collabo asserts because the number of information necessary in proceeding with the program changes, an efficient application system can be built by executing the microprogram that accounts for the variation in the “number of information.” Ex. 1049 (Collabo Markman), 16-17; *see also* Ex. 1001 ('575 Patent), 12:36-39.

223. It is my opinion that under Collabo's interpretation of the claim, Lee in view of Jahagirdar renders the claim obvious. The combination of Lee and Jahagirdar disclose two embodiments, each of which renders the claim obvious.

a) Lee-based Embodiment

224. As shown in Fig. 1, Lee discloses that the SOC 100 includes microprocessor 10, first peripheral module 30 and a second peripheral module 40, and each of these hardware modules operates in response to a corresponding working context. Ex. 1005 (Lee), Fig. 1, 5:61-67. Lee discloses that the power supply to all or a subset of these hardware modules may be removed when entering the power-off standby mode. For example, when entering the power-off standby mode, Lee describes

the “[s]teps of cutting off power supplied to *all or several circuits or modules* of the SOC 110.” Ex. 1005 (Lee), 6:1-3. Likewise, when describing step 740 of the flow chart depicted in Fig. 7, Lee mentions that after the working context has been saved, “the power supply 150 for supplying power to the at least one hardware module 10, 30, or 40 . . . is cut off in response to a power control signal PEN output from the power controller 80.” Ex. 1005 (Lee), 10:20-25.

225. In my opinion, a POSITA would appreciate the flexibility of the SOC 110 disclosed in Lee to implement a system that removes power to only select hardware modules. For example, a POSITA would implement a system where the power is cut off *only* to the microprocessor 10 or the power is cut off to all the hardware modules 10, 30 and 40. In the former scenario, a POSITA would implement the microprograms to save and restore working context only for the microprocessor 10 and not for the hardware modules 30 and 40.¹ In the latter scenario, the microprograms would save

¹ Even though Lee discloses cut off power to all or several hardware modules, it saves and restores context for all hardware modules. A POSITA would understand that if power is cut off for only a subset of the hardware modules, then it would be efficient to save and restore context for those hardware modules. Such a modification to Lee would be well within the skill of a POSITA in 2006 or earlier.

context and restore contexts for all the hardware modules 10, 30 and 40.

226. In my opinion, a POSITA would recognize the advantages of disabling specific modules based on either the generated power shutdown factor or the particular type of power shutdown factor selected. As explained in **Section XII.A.3.f** above, Lee presents two example power shutdown factors: either a user-defined predetermined time has elapsed, or a user chooses a power-off standby mode. Ex-1005 (Lee), 6:10-20. For instance, when a display timer runs out, a POSITA would appreciate the benefit of deactivating the display driver while keeping the microprocessor active for quick reactivation. On the other hand, if a user selects a complete system power-off standby mode, it would be advantageous for a POSITA to deactivate all modules. There would have been a reasonable expectation of success in implementing a selective execution of micrograms because it was a routine process.

227. Thus, the two micrograms in Lee that correspond to the claimed power shutdown microprogram and the claimed power restoration microprogram would save and restore the amount of information that is dictated by the power shutdown factor. In my opinion, the two micrograms in this embodiment will be “selectively executed in an alternative way,” consistent with Collabo’s interpretation of the claim.

b) Jahagirdar-based Embodiment

228. When discussing the flow chart in Fig. 1, Jahagirdar notes that when transitioning into the zero voltage power management state, the critical state of the processor is saved. Ex. 1006 (Jahagirdar), 3:13-15. Jahagirdar states that “[t]he *critical state* of the processor includes state variables associated with the architectural, micro-architectural, debug state, and/or similar state variables associated with that processor.” Ex. 1006 (Jahagirdar), 3:14-17. A POSITA would understand that Jahagirdar’s system can save the architectural state of the processor and many additionally save the microarchitecture state of the processor. The architectural state, for example, refers to the values in registers, such as, the Program Counter, and Stack Pointer, and the values in the register file in the processor. The architectural state exists only inside the processor and will be lost when the power to the processor is shut off. Thus, when entering the zero voltage power management state, the processor must save the architectural state. A POSITA would further understand that the micro-architectural state may also be saved for efficiency of restoring the program later. The microarchitectural state could include the values in the cache memory. The data in the cache memory is a subset of the data in the main memory. When the power to the processor is cut off, the data in the cache is lost but can be recovered from the main memory.

229. It is my opinion that in view of Jahagirdar’s description of the *critical*

state of the processor, a POSITA would understand that the processor can save all the enumerated information or only the information necessary to resume operation. For example, if the processor saves the micro-architectural state, then it will take longer to enter and exit the zero voltage power management state because the cache data needs to be saved and later restored. However, when the processor resumes operation, all the data is in the cache and the processor encounters fewer cache misses. On the other hand, if the processor does not save the micro-architectural state, the transition into and out of the zero voltage power management state is quicker because less data is saved and restored. However, when the processor resumes operation, it will encounter more cache misses that will cause the cache to fetch the data from the main memory. Saving only select states expedites the transition to the zero voltage state, which is advantageous if slightly more cache misses when waking up is an acceptable system design choice.

230. In my opinion, a POSITA can implement the above options by using signals, such as interrupt signals, to SOC 110 (or a component within the SOC 110) that notifies the SOC 110 what additional information to save beyond the architectural state. Since an interrupt signal is a multi-bit signal, it is obvious to encode the different levels of information saving desired by the system. For example, the system in Jahagirdar already distinguishes the signals that allow the processor to enter various

states (e.g., C1, C2, C3, C4 and C6 states), each of which provides different power savings with different entry/exit latencies. Hence, a signal to the SOC 110 that provides the details of which state to transition to (and as a result, how much information to save when entering the zero voltage power savings state) is obvious. The microprograms resident in the processor will only save and restore the requisite information based on the signal to the SOC 110. The abovementioned interrupt signal corresponds to the claimed power shutdown factor. The power shutdown microprogram would save different amount of information that is dictated by the interrupt signal from the system. The power restoration microprogram would restore the information that was saved when entering the shutdown mode. Thus, the two microprograms in this embodiment will be “selectively executed in an alternative manner,” according to Collabo’s interpretation of the claim. There would have been a reasonable expectation of success in implementing a selective execution of micrograms because it was a routine process.

7. Claim 11

231. Claim 11 provides: “*The microcontroller according to claim 1, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information holding when the power shutdown factor occurs.*”

232. I have reviewed **Section IX.A.6** of the Petition.

233. Jahagirdar discloses that “voltage from the I/O control hub 225 ($V_{I/O}$ 349) may be provided to the processor 205 in order to provide *sufficient power* to the dedicated cache memory 340 such that it can store the critical state variables associated with the processor 205 while the rest of the processor 205 is powered down by the reduction of the operating voltage 240 down to a zero state.” Ex. 1006 (Jahagirdar), 5:43-49. When discussing the Deeper Sleep state, Jahagirdar describes the “Deeper Sleep state in which all processor clocks are stopped and the processor voltage is reduced to a *lower data retention point*.” Ex. 1006 (Jahagirdar), 1:58-60. It is my opinion that a POSITA would understand the application of “sufficient power” to cache memory 340 implies that a lower voltage is applied to the cache memory when the processor 205 is powered down.

234. It is my opinion that a POSITA would be motivated to incorporate the abovementioned functionality in the SOC 110 disclosed in Lee because such a modification would further the goal of Lee to reduce the power consumption by the SOC 110. Based on the disclosure of Jahagirdar, a POSITA would modify the SOC 110 to ensure that when the SOC 110 transitions into the power-off standby mode, then the power supply to the memory 60 is reduced to a lower value that results in conservation of power but does not result in loss of data in memory 60. One benefit of such a scheme is that transfer of working context to off-chip non-volatile memory 130

may not be necessary, thereby resulting in a faster transition to the power-off standby state and a quicker recovery from that state. In this modified SOC 110 of Lee, the power shutdown range constitutes the microprocessor 10 and peripheral circuits 30 and 40 whereas memory 60 is outside the power shutdown range.

235. Moreover, a POSITA would ensure that when incorporating the abovementioned functionality in Jahagirdar into the SOC 110 disclosed in Lee, the POSITA would include a second power supply unit to supply voltage to the memory 60. As I explained in **Section VII.B.7** (discussing benefits of multiple power supplies) above, a POSITA would appreciate the benefits of a second power supply unit when providing one level of voltage to the CPU 10 but providing a different level of voltage to the memory 60 at the same time. There would have been a reasonable expectation of success because reducing voltage in power-saving modes constitutes a routine process. Implementing these measures aligns with Lee's objective of minimizing power consumption while ensuring the rapid saving and restoring of working contexts.

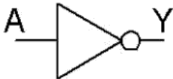
8. Claim 12

236. Claim 12 provides: "*The microcontroller according to claim 1, further comprising a circuit for loading an inverted value to a register in a circuit portion in a power shutdown range.*"

237. I have reviewed **Section IX.A.7** of the Petition.

238. As shown in Fig. 1, the SOC 110 in Lee contains a microprocessor 10. Ex.1005 (Lee), Fig. 1. A POSITA would understand that the assembly language of the microprocessor includes a NOT operation, and the microprocessor 10 would have the circuits to implement that functionality. Such a circuit would load an inverted value to a register. James Lieterman, entitled 32/63-BIT 80x86 Assembly Language Architecture, summarizes assembly language instructions available in various processors sold by Intel and AMD in 2004. Ex. 1050 (Leiterman). The excerpt below from Lieterman depicts a NOT operation. A POSITA would understand that “One’s Complement Negation” refers to inverting a value.

NOT — One’s Complement Negation

	A	Y
	0	1
	1	0

Mnemonic	P	PII	K6	3D!	3Mx+	SSE	SSE2	A64	SSE3	E64T
NOT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Ex. 1050 (Leiterman), 92. The table above lists all the processors that included the above operation:

	Intel
P	Pentium
PII	Pentium II (MMX)
SSE	SSE (Katmai NI)
SSE2	SSE2
SSE3	SSE3 (Prescott NI)
E64T	64-bit Memory

	AMD
K6	K6
3D!	3DNow!
3Mx+	3DNow! and MMX Ext.
A64	AMD64

Ex. 1050 (Leiterman), 8. It is my opinion that the NOT operation is a standard logical and arithmetic operation in computer architecture and included in all microprocessors. That is confirmed by Leiterman that a wide variety of Intel processors and AMD processors include this operation.

239. The Leiterman (excerpted below) depicts the format of the NOT operation and how it is implemented.

not *rmDst(8/16/32/64)* [Un]signed

This instruction logically inverts the bits of the destination. All ones become zeros, and all zeros become ones.

Flags	O.flow	Sign	Zero	Aux	Parity	Carry
	-	-	-	-	-	-

Flags: None are affected by this opcode.

```
mov     eax,00000a5a5h
not     eax
; NOT  00000000000000001010010110100101b  00000a5a5h
;      111111111111110101101001011010b  0ffff5a5ah
```

And guess what you get when you flip it again?

```
; NOT  111111111111110101101001011010b  0ffff5a5ah
;      000000000000000010100101101001b  00000a5a5h
```

The original value! Flip-flop.

Ex. 1050 (Leiterman), 93. The excerpt confirms that NOT operation inverts the bits of the destination register. The instruction is implemented by moving (MOV) a value (e.g., 00000a5a5h in hexadecimal format) into a register EAX. The NOT operation then loads the inverted value in the register EAX.

240. Leiterman also (excerpted below) discloses that EAX is an accumulator, which a POSITA would understand to be a type of register:

Table 3-5: 64-, 32-, and 16-bit general-purpose registers

Register	Extra Functionality
RAX, EAX, AX	The accumulator. If used as a pointer in Protected Mode uses the DS segment register as a default. DS:[EAX]
RBX, EBX, BX	Used as a data pointer using the DS segment register as a default. DS:[EBX]
RCX, ECX, CX	Used as a counter in string (rep) and loop operations. If used as a pointer in Protected Mode uses the DS segment register as a default. DS:[ECX]
RDX, EDX, DX	Input/output port address. If used as a pointer in Protected Mode uses the DS segment register as a default. DS:[EDX]
RSI, ESI, SI	Source index using DS segment register as a default. DS:[ESI]
RDI, EDI, DI	Destination index using ES segment register as a default. ES:[EDI]
RBP, EBP, BP	Pointer to data on the stack (very similar to ESP); uses the SS segment register as a default. SS:[EBP]
RSP, ESP, SP	Stack pointer used the SS segment register as a default. SS:[ESP]
RIP, EIP, IP	Instruction pointer. CS:[EIP]

Ex. 1050 (Leiterman), 44.

241. In **Section VII.B.1** above, I discussed the general architecture of a processor. Fig. 1.1 in that section depicts an ALU (Arithmetic Logic Unit) and Register File. A POSITA will understand that the NOT operation is implemented in the ALU and the Register File depicts registers, such as EAX that store the output of the ALU. As shown in **Section XII.A.3** above, the microprocessor 10 in Lee is in the claimed power shutdown range. In view of my discussion in **Section VII.B.1** (discussing

architectural state of a processor) above and the disclosure of Leiterman, it is my opinion that it would have been well within the skill of a POSITA to implement circuits within a processor that load an inverted value in a register, and such an implementation would have been routine and well-known at the time of the invention of the '575 Patent. In fact, when discussing the flow chart in Fig. 17, the '575 Patent states that the “[i]n step n56, the CPU 1 overwrites the inverted data S13 to pseudo-realize the relevant state to obtain the power shutdown state.” Ex. 1001 ('575 Patent), 15:62-64.

9. Claim 15

242. Claim 15 provides: “*The microcontroller of claim 1, wherein the power supply unit includes a voltage regulator.*”

243. I have reviewed **Section IX.A.8** of the Petition. A POSITA would understand that “on-chip regulator,” which serves as the power supply 150, corresponds to the claimed voltage regulator.

10. Claim 16

244. Claim 16 provides: “*The microcontroller of claim 1, wherein the power supply unit does not shut down the power supply at least to the clock generator upon receiving the power shutdown control signal from the power supply control unit.*”

245. I have reviewed **Section IX.A.9** of the Petition.

246. When the CPU 205 in Jahagirdar is entering the zero voltage power

management state (state C4 or state C6), “a STOP_CPU signal may be asserted by the input/output controller 225 and received by the clock generator 211 to cause the clock generator to *halt the clock signal CLK to the CPU 205.*” Ex. 1006 (Jahagirdar), 7:19-22. It is my opinion that a POSITA would understand that the clock generator 211 will suspend the CLK signal (or not provide the CLK signal) to the CPU 205. Because the clock generator 211 has to “halt the clock signal,” a POSITA would understand that the clock generator 211 must remain powered on in order to accomplish that. A POSITA would know that a clock generator can be simply gated to prevent the propagation of clock without shutting off the power supply to the clock. Indeed, Jahagirdar discloses that the clocks can be halted by the techniques of using an AND gate. *See, e.g.,* Ex. 1006 (Jahagirdar), 6:62-63 (“local clocks may be ***gated.***”), 7:8-9 (“external core clocks may be ***gated.***”)

247. It is my opinion that when modifying Lee to incorporate the clock generator disclosed in Jahagirdar, a POSITA would be motivated to incorporate the functionality of the STOP_CPU signal that instructs the clock generator 211 to halt the clock signal to the microprocessor 10 in the SOC 110. As I discussed in **Section VII.B.3** (discussing voltage and clock scaling) above, a POSITA would have appreciated that halting the clock signal by “gating” the clock while providing power to the clock generator shortens the time needed to exit from the standby state because

the system does not have to wait for the clock generator to be powered up and output a valid clock. It is my opinion that a POSITA, when modifying Lee in view of Jahagirdar as discussed above, would have a reasonable expectation of success because the proposed modification (e.g., gating the clock signal while maintaining power to the clock generator) constituted a routine and well-known design principle at the time of the invention of the '575 Patent.

B. Ground II: Lee, Jahagirdar and Peterson render obvious Claim 2

248. Lee, Jahagirdar and Peterson render obvious Claim 2.

1. Claim 2

249. Claim 2 provides: *“The microcontroller according to claim 1, wherein the clock generator selects one of operation performed after stopping supply of an operation clock and operation stop performed without stopping the supply of the operation clock as operation to be performed upon receiving the clock stop control signal from the power supply control unit.”*

250. I have reviewed **Section IX.B** of the Petition.

251. It is my opinion that Claim 2, as drafted, is indefinite at least because neither the claim nor the specification explain how to perform an “operation stop” as the claim requires “without stopping the supply of the operation clock.” In the district court, Collabo argues that the claim is not indefinite. Ex. 1049 (Collabo Markman),

15. Collabo argues that the specification provides the following explanation that appears to be pertinent to Claim 2:

The clock generator 3 may be configured to select a mode of stopping the supply of the operation clock or a mode of not stopping the supply of the operation clock based on the setting in the register, signal, terminal, or the like as the operation upon receiving the clock stop control signal S3 from the power supply control unit 2. In such a configuration, the operation according to user preference can be set based on mode selection. In the case of the mode of not stopping the supply of the operation clock, restoration is carried out in an arbitrary time by moving the counter of the timer or in any other methods.

Ex. 1001 ('575 Patent), 11:3-14.

252. In light of Collabo's interpretation of the claim, if Claim 2 requires the clock generator to either stop or not stop the clock upon receiving the clock stop control signal from the power supply control unit, then in my opinion, the combination of Lee, Jahagirdar and Peterson renders obvious Claim 2.

2. Motivation to combine Lee and Jahagirdar with Peterson

253. In my opinion, a POSITA would have appreciated the versatility of the clock generator disclosed in Peterson that reduces power consumption in a stand-by mode at the time of the invention of the '575 Patent. But Peterson also discusses the distinction between request to shut down from the actual shutdown so as to prevent an

accidental shutdown of the clock. When implementing Lee's system in view of Jahagirdar, it would have been obvious to incorporate Peterson's teaching of halting the clock signal only after validating that the clock stop control signal adheres to a predetermined protocol requirement.

254. I have discussed in **Sections X.A, X.C, and X.E** above, Lee, Jahagirdar, and Peterson are analogous art, and all pertain to digital systems aimed at reducing power consumption. Ex-1005 (Lee), 6:49-54; Ex-1006 (Jahagirdar), 5:27-42, 10:46-54; Ex-1010 (Peterson), 1:60-63. Each discloses controlling clock generator to achieve this objective. Ex-1005 (Lee), 1:30-34; Ex-1006 (Jahagirdar), 6:46-7:22; Ex-1010 (Peterson), 3:11-15. In addition, a POSITA would appreciate the advantages of a more flexible and versatile clock generator that can be programmed via register as opposed to a clock generator that was not programmable at all. A POSITA would have particularly recognized the advantage of minimizing accidental clock stoppage risk. Ex-1010 (Peterson), 8:13-16 (“[t]his access protection scheme avoids the inadvertent placement of the clock generator into shut-down mode, and thus reduces the risk of accidental clock stoppage.”)

255. In my opinion, a POSITA would have to make only minor modifications to the SOC 110 disclosed in Lee, and such modifications would have been within the skill of a POSITA. As is evident from my discussion in **Sections VII.B.2** (discussing

use of clocks) and **VII.B.3** (discussing voltage and clock scaling), use of clocks in digital design and how to control the clocks was well-known and well understood by the time of the invention of the '575 Patent. Modification of SOC 110 in Lee would require a POSITA to simply add a control generator along with the circuits to confirm that the predetermined protocol has been met, as disclosed in Peterson. Such a modification would be a combination of known elements in prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of SOC 110 disclosed in Lee would be based on applying known techniques (e.g., enabling and disabling clocks following a protocol verification) to a known device (e.g., the SOC 110) to improve the device. Thus, a POSITA would have chosen from a finite number of identifiable and predictable solutions. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Peterson to implement and improve the SOC 110 disclosed by Lee. The method of disabling the clock signal subsequent to verifying the clock stop control signal was a well-established means to circumvent unintended clock stoppage.

C. Ground III: Lee, Jahagirdar and Kang render obvious Claim 3

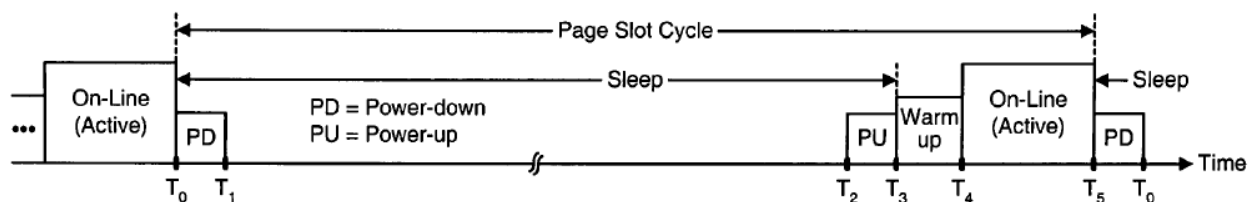
256. Lee, Jahagirdar and Kang render obvious Claim 3.

1. Claim 3

257. Claim 3 provides: “*The microcontroller according to claim 1, wherein the power supply control unit limits a period in which shutdown is possible.*”

258. I have reviewed **Section IX.C** of the Petition.

259. Fig. 4 of Kang depicts a timeline for processing a paging channel for a wireless communication system. Ex. 1011 (Kang), 2:52-53. Fig. 4 of Kang “shows the timeline for processing the paging channel in the 1xEV DV system. In FIG. 4, a new paging slot cycle for terminal 100 starts at time T_0 . Terminal 100 sleeps from time T_0 until its next assigned slot. Terminal 100 wakes up at time T_3 , prior to its next assigned slot, and powers on and warms up the necessary circuitry. Terminal 100 receives and processes the paging channel starting at time T_4 . Terminal 100 finishes processing the paging channel at time T_5 and thereafter goes back to sleep if additional communication is not needed.” Ex. 1011 (Kang), 8:26-35.



Ex. 1011 (Kang), Fig. 4.

260. The terminal wakes up at regular intervals dictated by the paging channel protocol. Fig. 4 “shows an overlay of power collapse over the sleep timeline. After

completion of the on-line processing at time T_0 , terminal 100 performs powering down tasks during the power-down period from time T_0 to time T_1 . Prior to the warm-up time, terminal 100 performs powering up tasks during the power-up period from time T_2 to time T_3 .” Ex. 1011 (Kang), 8:41-46. In addition, Thus, Kang discloses that the terminal 100 should not be powered down during its paging slot but can be powered down between the paging slots.

261. In my opinion a POSITA would be motivated to incorporate the functionality disclosed in Kang in the SOC 110 disclosed in Lee, for the reasons discussed below. In accordance with the system disclosed in Kang, the power controller 80 of Lee would be modified to ensure that a power shutdown signal is not sent to various circuit block at certain time durations, for example, durations when the SOC 110 is expected to be active.

2. Motivation to combine Lee and Jahagirdar with Kang

262. In my opinion, at the time of the invention of the '575 Patent, a POSITA would have appreciated the benefits Kang discloses—ensuring that the microcontroller does not go into the standby mode when the microcontroller is supposed to be active. A POSITA would be motivated to incorporate this functionality in the SOC 110 disclosed in Lee. Such a modification would have improved the performance of the SOC 110 in Lee and made the SOC 110 more suitable for telecommunications

applications where devices are expected to be awake during the assigned slots. The applications of the SOC 110 (that has the same functionality as the claimed microcontroller disclosed in the '575 Patent) in the telecommunications applications. In fact, Collabo has accused that Qualcomm's processors that are employed in the telecommunications area infringe the claims of the '575 Patent. Ex. 1041 (Collabo's Preliminary Infringement Contentions), 2.

263. In my opinion, the proposed modification of SOC 110 in Lee to incorporate functionality disclosed in Kang to limit the period in which shutdown is possible, as disclosed in Lee, would be a combination of known elements in prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of SOC 110 disclosed in Lee would be based on applying known techniques (e.g., when to enter and exit standby modes) to a known device (e.g., the SOC 110) to improve the device. Kang's strategy of ensuring a cellphone remains active during specific time slots was a known method that a POSITA would have found obvious to apply within Lee's systems implemented in view of Jahagirdar. In addition, in my opinion, a POSITA would have appreciated the advantages of ensuring that the device did not enter the standby mode when it was expected to be active to process the incoming signals. Specifically, a POSITA would understand that periodic monitoring of the paging channel is crucial for ensuring a

paging device receives incoming messages. In my opinion, a POSITA would be motivated by the knowledge of the skilled artisan to choose from a finite number of identifiable and predictable solutions to control when the device enters and exits the standby mode. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Kang to implement and improve the power controller 80 of the SOC 110 disclosed by Lee because the modification would involve use of known components in a known and well understood manner. Periodic monitoring of the paging channel was widely acknowledged and widely understood as an effective method for securing message reception on a cellphone.

D. Ground IV: Lee, Jahagirdar and Higashida render obvious Claim 11

1. Claim 11

264. Claim 11 provides: “*The microcontroller according to claim 1, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information holding when the power shutdown factor occurs.*”

265. I have reviewed **Section IX.D** of the Petition.

266. In my opinion, for the reasons discussed below, a POSITA would be motivated to modify Lee to supply a lower voltage to memory 60 when the SOC 110

enters the power-off shutdown mode based on the disclosure of the fifth embodiment of Higashida. The reduced voltage to memory 60 is lower than the voltage provided during normal operation but sufficient to retain data stored in memory 60. The reduced voltage is supplied through a voltage supply unit that is in addition to the power supply 150 depicted in Fig. 1 of Lee.

a) Motivation to combine Lee and Jahagirdar with Higashida

267. In my opinion, a POSITA would be motivated to modify Lee to incorporate the fifth embodiment of Higashida. The POSITA would augment SOC 110 in Lee with a backup power source in addition to the primary power supply device. Mirroring the design in Higashida, the backup power source in Lee would supply power to memory 60 in Lee. When the microprocessor 10 and peripheral units 30 and 40 in Lee transition to the power-off shutdown mode, the power supply to these modules is disabled, but other circuits, such as memory 60, can continue to receive voltage from the backup power source, albeit at a lower level that is sufficient to retain data yet low enough to conserve power. In my opinion, a POSITA would recognize that the design of the SOC 110 of Lee would improve in two ways due to this modification. *First*, the power consumption of the SOC 110 would be lowered, thereby furthering the goals of Lee. Though Lee mentions disabling power supply to the

microprocessor 10 and peripheral units 30 and 40, it is silent on removing power supply to memory 60. If the power supply to memory 60 is not removed in the initial design, then by reducing the power supply to memory 60 as opposed to keeping it at the operational level, the overall power consumption by SOC 110 will be reduced. *Second*, another benefit of such a scheme is that transfer of working context to off-chip non-volatile memory 130 may not be necessary, thereby resulting in a faster transition to the power-off standby state and a quicker recovery from that state. Because the memory 60 continues to receive power, it retains the working context. Thus, a POSITA would appreciate that the design of SOC 110 in Lee can be improved by eliminating the step of moving the working context to the off-chip non-volatile memory 130. In this modified SOC 110 of Lee, the power shutdown range constitutes the microprocessor 10 and peripheral circuits 30 and 40 whereas memory 60 is outside the power shutdown range. In the modified SOC 110, the backup power supply would be the claimed second power supply that supplies lower power to memory 60 when the power shutdown factor occurs.

268. As discussed in **Section VII.B.7** (discussing benefits of multiple power supplies) above, providing different power at different voltage levels to memory and CPU through multiple power sources was well-known and well understood by the time of the invention of the '575 Patent. As a result, modification of SOC 110 in Lee provide

a lower voltage to the memory 60 through a second power supply when the CPU is placed in standby mode, as disclosed in Higashida, would be a combination of known elements in prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of SOC 110 disclosed in Lee would be based on applying known techniques (e.g., providing different voltages to memory and CPU through different power supply units) to a known device (e.g., the SOC 110) to improve the device. As discussed in **Section VII.B.7** (discussing multiple power supplies) above, a POSITA would have appreciated the advantages of using multiple power supplies when CPU and memories receive different voltage levels in order to reduce power consumption as opposed to only one level of power supply to all components in a circuit. A POSITA would specifically appreciate the advantages of preserving information within memory while enhancing power efficiency. Thus, a POSITA would be motivated by the knowledge of the skilled artisan to choose from a finite number of identifiable and predictable solutions. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Higashida to implement and improve the SOC 110 disclosed by Lee, considering that it mirrored routine procedures for voltage reduction within power-saving modes.

E. Ground V: Shikata and Crump render obvious claims 1, 5, 11, 12, and 15

269. My analysis and conclusions in this Section are applicable to claims 1, 5, 11, 12, and 15 of the '575 Patent.

1. Motivation to Combine Shikata and Crump

270. When seeking to implement the processor 100 disclosed in Shikata, a POSITA would recognize that Shikata discloses that an application will be suspended and the processor 100 will be placed in the standby state after the processor has saved its context. However, Shikata lacks explicit disclosure of how to determine whether to suspend an application and place the CPU core unit 3 in processor 100 in the standby state. It also lacks disclosure regarding how to notify CPU core unit 3 to save its context and transition to the standby state. When faced with the above, a POSITA would turn to analogous art, such as Crump. Like the technique disclosed in Shikata, Crump also aims to reduce power consumption in a processor and discloses saving and restoring context when transitioning in and out of a standby state. Crump discloses various scenarios that result in a decision to place the processor in a standby state, and also discloses that a processor can be notified that it must transition to the standby state via an interrupt mechanism.

271. Shikata mentions that context should be saved and later restored using boot ROM. Shikata recommends storing the boot ROM instructions in the on-chip

internal RAM 14 as opposed to the exterior SDRAM 102 because the boot ROM instructions can be fetched faster from on-chip memory as opposed to off-chip memory. Ex. 1007, (Shikata), 2:40-45 (“register values to be readily saved and restored at high speed”). In addition, Shikata contemplates using boot instructions to save and restore context instead of a hardware-based scheme. Ex. 1007 (Shikata), 3:25-30 (“Further, the simple and high-speed evacuation of recovery of register values are achieved . . . without a need to make substantial modification to the logic of the CPU core unit of the processor.”). The purpose in relying upon the on-chip memory is to reduce the time it takes to recover to the normal state when returning from the standby state. A POSITA would appreciate that using a microprogram (or microcode) to save and restore context would further shorten the time needed to restore context and further advance the goals of Shikata.

272. A POSITA would be aware that various options were available at the time of the invention of the '575 Patent to save and restore working context, and the POSITA would have been the aware of the costs and benefits of each option. In my opinion, a POSITA would know that the functionality could be implemented using hardware, a program, or a microcode (or a microprogram) that runs at the lowest level. Crump discloses using microcode for saving and restoring context when the CPUs transition into and out of standby mode. For the reasons described further in **Section**

VII.B.4 (discussing microprograms), a POSITA would have been motivated to follow the approach described in Crump and use microcode to save and restore context.

273. Finally, Shikata discloses the voltage controlling unit eventually suspends the power supply to the power supply block 32 that contains the CPU code unit 3. In my opinion, a POSITA would understand that the CPU core unit 3 notified the voltage controlling unit when the evacuation of the context was finished. When modifying Shikata to incorporate micrograms discloses in Crump, a POSITA would understand that it would be a good design choice to incorporate the step of notification that evacuation is complete in the microprogram. *See Section VII.B.8* above (discussing interprocess communication). In my opinion, a POSITA would appreciate that when the microprogram is finished saving the working context in Shikata, then it would efficient for the CPU core unit 3 to notify that the evacuation is complete and the system can proceed to the next step.

2. Combination would be routine for a POSITA with a reasonable expectation of success

274. As discussed in **Section VII.B.8** (discussing interprocess communication) above, use of interrupts in digital design for communication between circuits and use of microprograms (or microcode) was well-known and well understood by the time of the invention of the '575 Patent. As a result, modification of

processor 100 in Shikata to incorporate interrupts and use microprograms to save and restore context, as disclosed in Crump, would be a combination of known elements in the prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of processor 100 disclosed in Shikata would be based on applying known techniques (e.g., interrupts and microprograms) to a known device (e.g., the processor 100) to improve the device. As discussed in **Section VII.B.8** (discussing interprocess communication) above, use of interrupts is very common in digital designs, such as the processor 100 of Shikata, and use of microprograms is one of the few approaches to saving and restoring context. Shikata discloses using interrupts (SMI) when transitioning from the standby state. A POSITA would readily understand the significance of the SMI signal in instructing the CPU core unit to save register data. In addition, a POSITA would appreciate the advantages of using microprograms to implement Lee as opposed to other approaches to save and restore context. These advantages are especially valuable in light of Shikata's teachings on the necessity to minimize hardware modifications while swiftly saving and restoring register values. Ex. 1007 (Shikata), 2:40-45, 3:25-30. Thus, a POSITA would be motivated by the knowledge of the skilled artisan to choose from a finite number of identifiable and predictable solutions. It would have been obvious to try microcode because it was one of three approaches for managing states, the others being

hardware or program solutions. A POSITA would have assessed the pros and cons of each approach and recognized the obvious choice of microcode for its balanced attributes of flexibility and rapid operation. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Crump to implement and improve the processor 100 disclosed by Shikata. Interrupt signals were a well-established method of communication within digital systems like those described by Shikata. Moreover, using microcode was a conventional technique for implementing save-and-restore functions. The decision among microcode, hardware, or program is driven by requirements for flexibility and latency, and a POSITA would be familiar with the efficiency and adaptability afforded by microcode.

3. Claim 1

275. Shikata in view of Crump renders Claim 1 obvious.

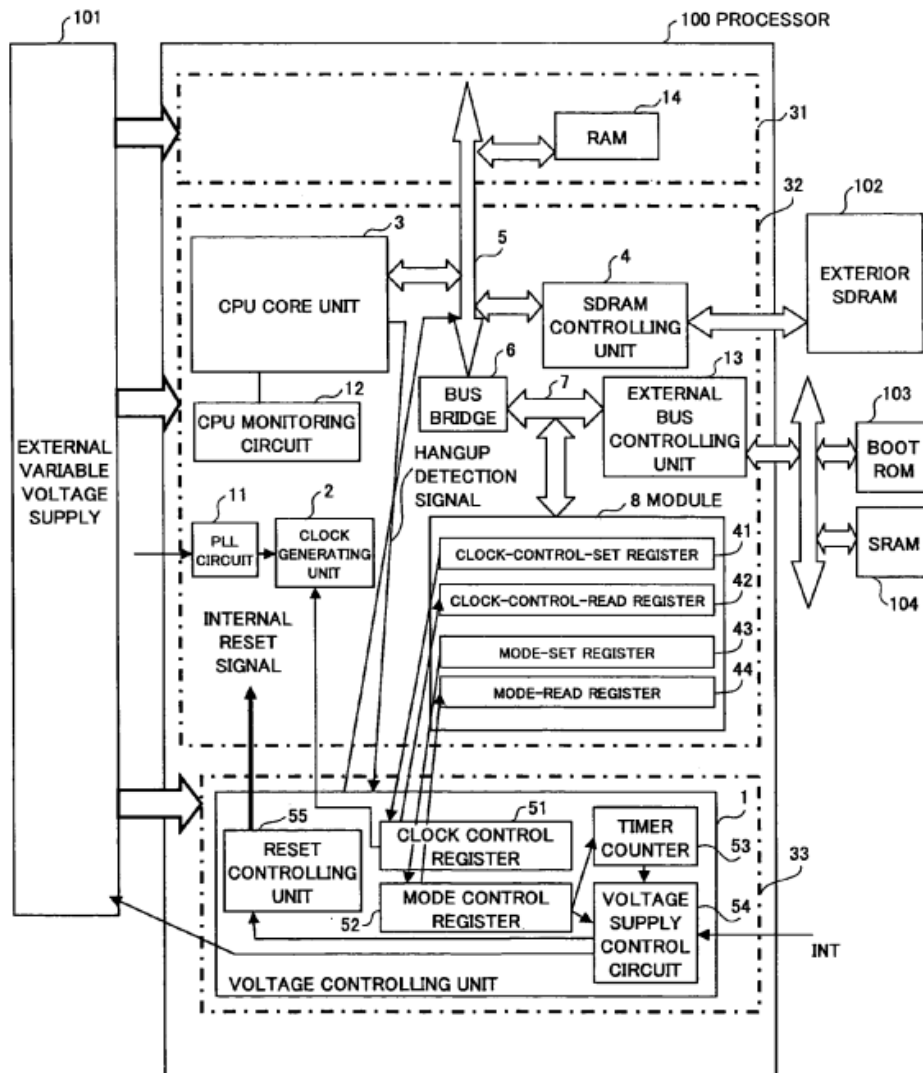
[Ipre] A microcontroller comprising:

276. I have reviewed **Section IX.E.2** of Petition discussing the preamble for Claim 1. *See also* **Section IX.A** above. A POSITA would have known that Shikata's processor is implemented on a single chip.

a) [1a] a CPU;

277. I have reviewed **Section IX.E.2.a** of the Petition. In my opinion, POSITA

would understand that the CPU Core Unit 3 shown in Fig. 5 of Shikata as part of the processor 101 corresponds to the claimed CPU.



Ex. 1007 (Shikata), Fig. 5.

- b) [1b] a power supply unit arranged between the CPU and a power supply device for supplying power to the CPU;

278. I have reviewed Section IX.E.2.b of the Petition.

279. A POSITA would understand that the external variable voltage supply is also connected to a power source that delivers either AC or DC power, as a variable voltage supply needs such a source to power the processor. This power source corresponds to the claimed power supply device, which the parties agree means “power supply.” Thus, a POSITA would understand that the external variable voltage supply is arranged between CPU and a power supply device for supplying power to the CPU.

280. In my opinion, when implementing the system disclosed in Shikata, a POSITA would modify the processor 100 to incorporate the external variable voltage supply 101 on the same semiconductor chip as the other components. As discussed in **Sections VII.B.5** (discussing benefits of integration) and **VII.B.6** (discussing benefits of integrating power supply) above, at the time of the invention of the '575 Patent, a POSITA would have appreciated that by integration of the external variable voltage power supply 101 with the processor 100 would yield a circuit with a higher performance, lower power consumption and lower production costs. Such an integration would have been feasible and within the skill of a POSITA at that time. There would have been a reasonable expectation of success in implementing this change, as it was routine practice to integrate power supply circuitry within a single-die system at the time of the invention of the '575 Patent.

- c) **[1c] a power supply control unit for controlling the CPU and the power supply unit; and**

281. I have reviewed **Section IX.E.2.c** of the Petition.

- d) **[1d] an information holding unit for holding information evacuated from the CPU, the information being necessary in proceeding with a program; and**

282. I have reviewed **Section IX.E.2.d** of the Petition.

- e) **[1e] a clock generator; wherein:**

283. I have reviewed **Section IX.E.2.e** of the Petition.

- f) **[1f] the power supply control unit outputs a shutdown request signal to the CPU in response to an occurrence of a power shutdown factor,**

284. I have reviewed **Section IX.E.2.f** of the Petition.

285. When discussing Step S1 of Fig. 6, Shikata states that “an ongoing application is suspended.” Ex. 1007 (Shikata), 8:21. The processor 100 then enters the standby state. In my opinion, a POSITA would understand that the voltage controlling unit 1 notifies the CPU core unit 3 to execute the standby sequence. In Shikata, the voltage controlling unit includes the reset controlling unit, which sends a signal to power on the CPU core unit when returning from standby. Ex 1007 (Shikata), 8:2-5, Fig. 5. Thus, in my opinion, a POSITA would understand that the voltage controlling unit would initiate the CPU core unit’s shutdown also.

286. Crump discloses controller U2 of power management circuit 106 (i.e., the

claimed “power supply control unit”) that, in response to a press of the power switch 21 (an example of the claimed “occurrence of a power shutdown factor”), outputs an SMI interrupt (i.e., the claimed “shutdown request signal”) to CPU 40 (i.e., the “CPU”). Crump discloses a power management processor that monitors the occurrence of a power shutdown factor:

The power management processor monitors the system power switch to determine when the user has requested the system to be turned on or off. The power management processor also implements a minutes to wake alarm, which can be used to cause the system to transition to the normal operating state from either the off state or the suspend state at a time up to 32 years in the future.

Ex. 1008 (Crump), 4:44-51. Crump further discloses that “[t]he microcontroller U2 can interrupt the CPU 40 via the CPU’s system management interrupt (SMI), if the CPU 40 has an SMI.” Ex. 1008 (Crump), 21:52-53.

287. In my opinion, a POSITA would be motivated to incorporate in Shikata the functionality of monitoring the occurrence of a power shutdown factor disclosed in Crump. Moreover, a POSITA would incorporate in Shikata the functionality of sending a shutdown request signal to the CPU when the power shutdown factor occurs. As I discussed in **Section VII.B.8** above (discussing communication between circuit blocks) above, relying upon communication between circuit blocks leads to more

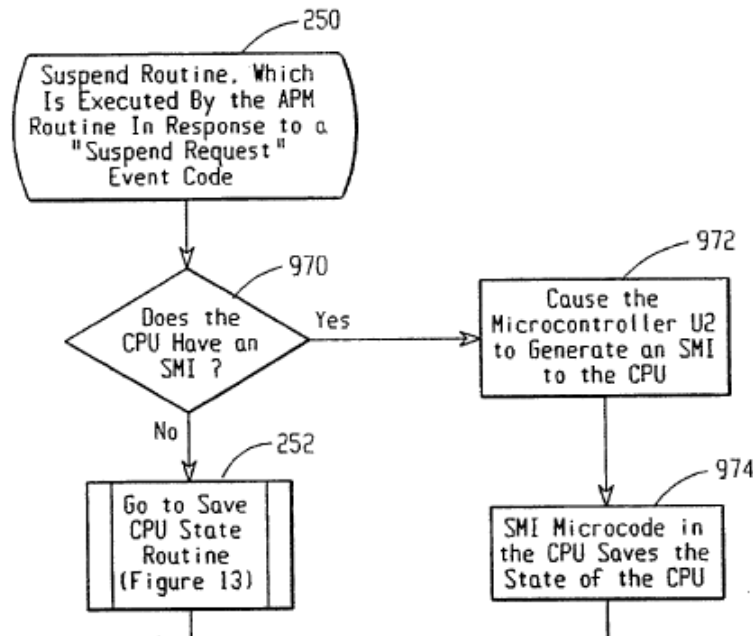
efficient and reliable design.

- g) **[1g] the CPU, upon receiving the shutdown request signal, executes a power shutdown microprogram, evacuates the information necessary in proceeding with the program to the information holding unit, and outputs an evacuation completed signal to the power supply control unit after evacuation is completed,**

288. I have reviewed **Section IX.E.2.g** of the Petition.

289. While Shikata discloses saving register values of the CPU core unit, it does not specify the implementation details. On the other hand, Crump discloses saving register values using microcode. In view of the benefits of employing microcode to save register values as compared to other approaches that I discussed in **Section VII.B.4** (discussing microprograms) above, in my opinion, a POSITA would be motivated to save the register values in the internal RAM by the CPU core unit by executing a microprogram (or microcode).

290. Crump employs a Suspend Routine to save processor context. Figs. 10A through 10F depict flow chart showing the details of the Suspend Routine and the relevant portion is excerpted below:



Ex. 1008 (Crump), Fig. 10A (excerpted). Crump provides the following disclosure regarding the excerpted portion of Fig. 10A

The Suspend Routine is called by the APM [Advanced Processor Management] in response to the Supervisor Routine returning to the APM a "Suspend Request" APM Return Code. . . . First, the flow of the Suspend Routine depends on whether the CPU 40 is an S part having an SMI, at 970. If so, the CPU 40 causes the microcontroller U2 to generate an SMI back to the CPU 40, at 972. ***Responsive to the SMI, microcode in the CPU 40 saves the state of the CPU 40***, as is known to those skilled in the art, to the segment E000H data structure, at 974.

Ex. 1008 (Crump), 37:1-13; *see also id.*, 31:22:4-10, 50:23-31, Fig. 13A.

291. While Shikata discloses that an application will be suspended and the

CPU core unit 3 will be placed in the suspend mode in order to conserve power, it does not explicitly disclose the details of how the decision to enter the shutdown mode is made and how it is conveyed to the processor 100. As disclosed above, Crump discloses using microcode to save context of the processor. In view of the benefits associated with using microcode to save context, a POSITA would use microcode to save context when transitioning to the suspend state and then to notify the voltage controlling unit 1 when the context has been saved. As discussed in **Section VII.B.4** (discussing microprograms) above, at the time of the invention of the '575 Patent, a POSITA would have appreciated the advantages of relying upon a microprogram (or microcode) to save and restore context as opposed to other approaches available at that time. Moreover, using a microprogram (or microcode) to save and restore context as taught in Crump would further the goals of Shikata. When entering the standby state, Shikata discloses storing the boot program in the internal RAM 14 along with the register values (i.e., processor context) to facilitate faster recovery from the standby state. Ex. 1007 (Shikata), 4:45-53. Moving the boot program from external ROOT ROM 103 to internal RAM 14 would facilitate faster recovery, but implementing the save and restore functionality in a microprogram instead of a boot program would be even faster.

292. The microcode disclosed in Crump would meet the claim construction of

the term “microprogram” proposed by Qualcomm. *See Section IX.B* above. As Collabo itself has pointed out, a POSITA would understand that microcode and microprogram are synonyms. Ex. 2017 (Chung POPR), 28. In my opinion, the microcode disclosed in Crump when implemented in the processor 100 disclosed in Shikata would reside within the CPU core unit 3 to ensure that the microcode can be access rapidly, as opposed to a program that would reside in internal RAM 14, BOOT ROM 103 or Exterior SDRAM 102. As depicted in Jahagirdar, the microcode is resides inside the CPU. Ex. 1006, (Jahagirdar), Fig. 2A. Like any microcode, the microcode in Crump runs at the lowest level of the hardware and controls the internal resources of the CPU core unit 3, in addition to saving and restoring context.

293. As discussed below, Shikata discloses that eventually the voltage controlling unit 1 suspends the power supply to the power supply block 32 that contains the CPU core unit 3. In my opinion, based on this disclosure and my discussion in Sections **XII.E.1** and **XII.E.2** above, a POSITA would understand that the CPU core unit 3 would notify the voltage controlling unit 1 when the CPU core unit 3 has saved the register values in the internal RAM 14. Such a notification would correspond to the claimed evacuation completed signal.

- h) [1h] the power supply control unit, upon receiving the evacuation completed signal, outputs a power shutdown control signal to the power supply unit,**

294. I have reviewed **Section IX.E.2.h** of the Petition. Shikata discloses that the voltage controlling unit outputs the control signal or instruction to the external variable voltage supply after the register data are saved. Ex. 1007, 8:21-42. Thus, a POSITA would understand that the voltage controlling unit issues the control signal after the CPU core sets a value in the voltage controlling unit's control register, indicating that the "saved register data are stored in the internal RAM." Ex. 1007, 5:4-11.

- i) **[1i] the power supply unit shuts down power supply to the CPU upon receiving the power shutdown control signal from the power supply control unit,**

295. I have reviewed **Section IX.E.2.i** of the Petition.

- j) **[1j] the power supply control unit, upon receiving the evacuation completed signal, outputs a clock stop control signal to the clock generator,**

296. I have reviewed **Section IX.E.2.j** of the Petition.

297. Shikata discloses this limitation. Shikata discloses that the CPU core unit 3 notifies the voltage controlling unit 1 when the evacuation has been completed. Specifically, in my opinion, a POSITA would understand that the voltage controlling unit issues the control signal after the CPU core sets a value in the voltage controlling unit's control register, indicating that the "saved register data are stored in the internal RAM." Ex. 1007 (Shikata), 5:4-11. Thus, Upon receiving the notification, the voltage

controlling unit 1 (i.e., the power supply control unit) instructs the clock generating unit 2 to suspend the clock.

- k) **[1k] the power supply control unit outputs a power supply control signal to the power supply unit in response to an occurrence of a power supply restoration factor,**

298. I have reviewed **Section IX.E.2.k** of the Petition.

- l) **[1l] the power supply unit, upon receiving the power supply control signal, starts to supply power to the CPU,**

299. I have reviewed **Section IX.E.2.l** of the Petition.

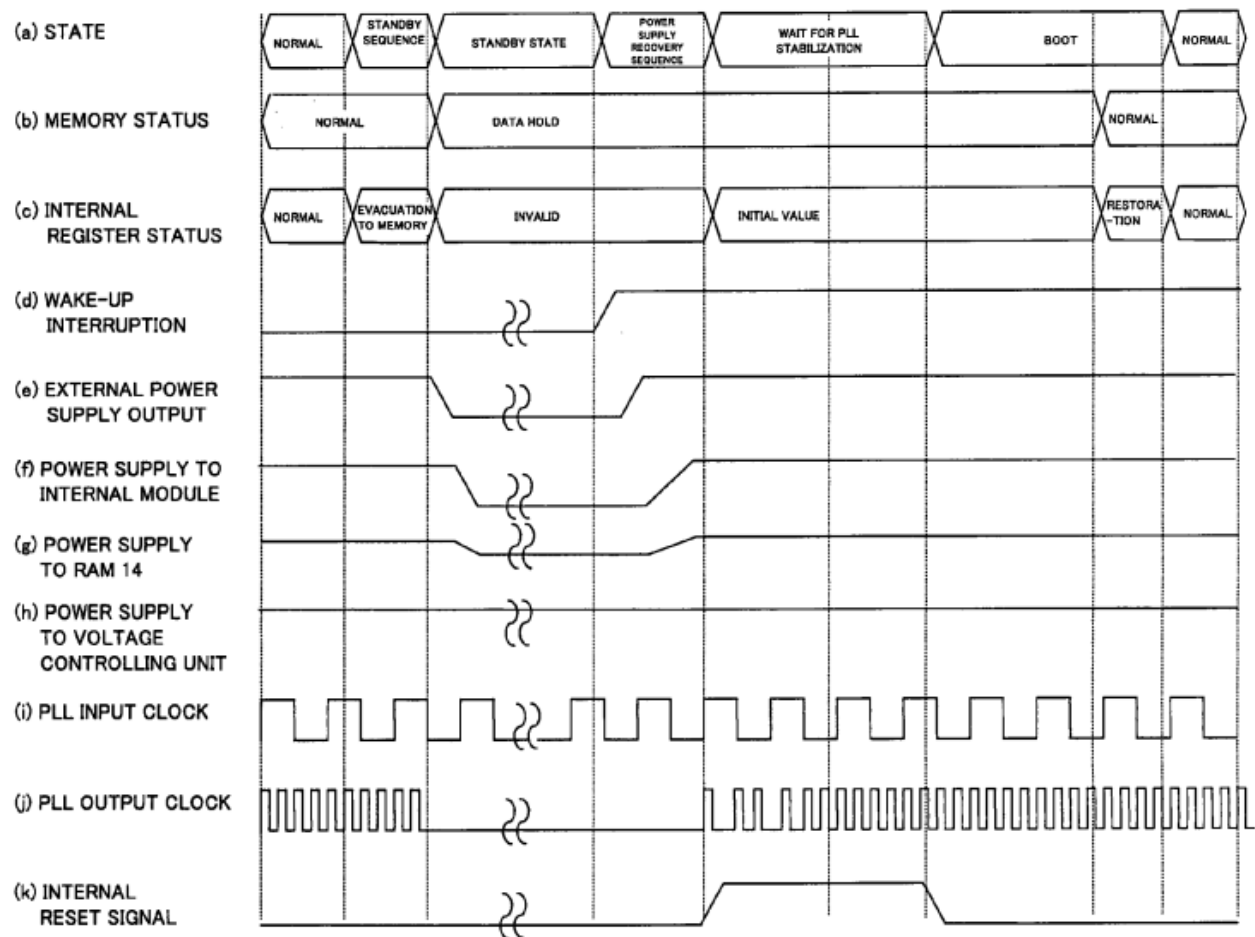
- m) **[1m] the power supply control unit outputs a restoration request signal to the CPU and outputs a clock supply control signal to the clock generator when a supplied power supply voltage is stabilized,**

300. I have reviewed **Section IX.E.2.m** of the Petition.

301. Shikata discloses that in Step S4 of Fig. 7, the CPU core unit 3 initiates fetching of boot instructions. In my opinion, a POSITA would understand that the voltage controlling unit 1 must have instructed the CPU core unit 3 to start that process after the voltage had stabilized. Such an instruction would correspond to the claimed restoration request signal.

302. When reviewing Fig. 8, a POSITA would note that the increase in voltage from zero to the normal voltage level in steps (e), (f) and (g) is indicated by a ramp

because the voltage does not transition from zero to normal instantaneously. Ex. 1007 (Shikata), 9:15-17. Instead, the voltage requires a finite time to ramp up and settle down, as I discussed in **Section VII.B.3** (discussion voltage and clock scaling) above. In my opinion, a POSITA would understand that the slope of the ramp accounts for that delay and the flat portion in steps (e) through (g) indicates that the voltage is now stable at the normal level.



Ex. 1007 (Shikata), Fig. 8.

303. Step (j) of Fig. 8 also depicts that the PLL output clock is output after the voltage levels have stabilized. Ex. 1007 (Shikata), 9:17-19. Thus, the voltage controller unit 1 waits until the voltage levels have stabilized before instructing the PLL circuit 11 and clock generator 2 to start supplying the clock again. In addition, as I explained in **Section VII.B.3** (discussing voltage and clock scaling) above, a POSITA would know that it is good design practice to permit voltage levels to settle when voltage level is raised from 0 to operating level (e.g., 3.3 V or 5 V) to ensure reliable operation of circuits. In addition, Crump discloses using the POWERGOOD signal, a signal that is “well known in the art.” Ex. 1008 (Crump), at 15:63-64. The POWERGOOD signal notifies the system that the power supply has stabilized.

n) **[1n] the clock generator, upon receiving the clock supply control signal, restarts to supply an operation clock, and**

304. I have reviewed **Section IX.E.2.n** of the Petition.

305. In my opinion, a POSITA would understand that the clock generating unit 2 would resume supply of the clock after the output of the PLL circuit 11 has stabilized, as disclosed in the context of Step S3 of Fig. 7. In addition, Shikata mentioned that in step S4 of Fig. 7, the CPU core unit 3 starts to fetch boot instructions. A POSITA would understand that the CPU core unit 3 would not be able to accomplish these steps without the supply of an operational clock.

- o) [1o] the CPU, upon receiving the restoration request signal, executes a power supply restoration microprogram, restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown, and then branches a process to an address indicated by a program counter to continue program execution from a shutdown state.**

306. I have reviewed **Section IX.E.2.o** of the Petition.

307. Shikata discloses that “[a]t step S5 [of Fig. 7], the boot program stored in one area of the internal RAM 14 recovers the register values and the like saved to another area of the internal RAM 14.” Ex. 1007 (Shikata), 8:60-63, 4:67-5:3. Based on this disclosure, in my opinion, a POSITA would understand that the values restored in Shikata in Step S5 of Fig. 7 constitute the claimed information necessary in proceeding with the program.

308. Based on the disclosure of Shikata that “the application program is resumed,” Ex. 1007 (Shikata), 8:63-65, in my opinion, a POSITA would understand that when the register values were being restored, the Program Counter (PC) was restored to the earlier value before the standby sequence was initiated. In my opinion, POSITA would further understand that the Program Counter (PC) provides the location of the instruction that will be executed by a CPU. As discussed in **Section VII.B.1** (discussing architectural context of processor) above, a POSITA would

understand that the Program Counter (PC) dictates which instruction a CPU fetches and executes.

309. As mentioned in **Sections X.D and XII.B.3.g** above, Crump teaches using microcode to save context. Ex. 1008 (Crump), at 37:1-13, Fig. 10A. A POSITA would understand that microcode is employed by Crump to restore context too. As I discussed in **Section VII.B.4** (discussing microprograms) above, at the time of the invention of the '575 Patent, a POSITA would have appreciated the advantages of relying upon a microprogram (or microcode) to save and restore context as opposed to other approaches available at that time, and would rely upon microcode to save and restore context in Shikata instead of the boot ROM code or any other mechanism available to a POSITA in that time frame.

4. Claim 5

310. Claim 5 provides: “*The microcontroller according to claim 1, wherein the power supply restoration factor uses an interrupt system of the microcontroller.*”

311. I have reviewed **Section IX.E.3** of the Petition.

5. Claim 11

312. Claim 11 provides: “*The microcontroller according to claim 1, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information*

holding when the power shutdown factor occurs.”

313. I have reviewed **Section IX.E.4** of the Petition.

314. Regarding supply of power to various power supply blocks 31, 32, and 33, Shikata discloses that “[t]he power supply blocks 31 through 33 receive respective, separate power supplies from the external variable voltage supply 101, and are configured to control the power supply independently of each other.” Ex. 1007 (Shikata), 6:47-51. It is my opinion that, based on the above disclosure, a POSITA would understand that because power supply for each power supply block can be configured independently, each power supply block has a power supply unit dedicated to supplying power to that block. A POSITA would understand that the dedicated power supply unit for the power supply block 31 constitutes the claimed second power supply unit.

315. Moreover, in my opinion, a POSITA would ensure that when implementing Shikata to provide multiple voltage levels, the POSITA would include a second power supply to supply voltage to RAM 14. As I explained in **Section VII.B.7** (discussing multiple power supplies) above, a POSITA would appreciate the benefits of a second power supply when providing one level of voltage to the CPU 10 but providing a different level of voltage to the memory 60 at the same time.

6. Claim 12

316. Claim 12 provides: “*The microcontroller according to claim 1, further comprising a circuit for loading an inverted value to a register in a circuit portion in a power shutdown range.*”

317. I have reviewed **Section IX.E.5** of the Petition.

318. As I discussed in **Section XII.A.8** above, microprocessors include a NOT operation that is a circuit for loading an inverted value. In my opinion, a POSITA would understand that CPU Core Unit 3 in Shikata will include such an operation. As shown in **Section XII.E.3** above, the CPU Core Unit 3 is shut down and is in a power shutdown range.

7. Claim 15

319. Claim 15 provides: “*The microcontroller of claim 1, wherein the power supply unit includes a voltage regulator.*”

320. I have reviewed **Section IX.E.6** of the Petition.

321. As discussed in **Section XII.E.3** above, Shikata discloses an external variable voltage supply 101. Ex. 1007 (Shikata), Fig. 5. In view of the disclosure of Shikata, *id.*, 4:29-30; 8:34-42, 9:46-6 and Fig. 8, in my opinion, a POSITA would understand that the external variable voltage supply 101 includes a voltage regulator so that it can regulate the voltage to the processor.

322. In the event Shikata does not disclose a voltage regulator, in my opinion Crump discloses a voltage regulator. Regarding power supply to the computer system, Crump discloses that “[t]he power supply has several power supply states, which are controlled by the power management processor responsive to the CPU, the switch . . . and the power management processor itself.” Ex. 1008 (Crump), Abstract.

323. It is my opinion that a POSITA would be motivated to include a voltage regulator disclosed in Crump in the external variable voltage supply 101 disclosed in Shikata. When discussing the external variable voltage supply 101, Shikata states that “[i]t is possible to suspend the supply of the operating voltage or lower the operating voltage of the processor 100 by controlling the external variable voltage supply 101 through a control signal output from the voltage controlling unit 1 of the processor 100.” Ex. 1007 (Shikata), 4:31-36. In my opinion, a POSITA would understand that the external variable voltage supply 101 requires a voltage regulator to provide the abovementioned functionality. There would have been a reasonable expectation of success because implementing a voltage regulator was a routine process.

F. Ground VI: Shikata, Crump and Peterson render obvious Claim 2

1. Claim 2

324. Claim 2 provides: “*The microcontroller according to claim 1, wherein the clock generator selects one of operation performed after stopping supply of an*

operation clock and operation stop performed without stopping the supply of the operation clock as operation to be performed upon receiving the clock stop control signal from the power supply control unit.”

325. I have reviewed **Section IX.F** of the Petition.

326. As discussed in **Section XII.B** above, in my opinion Claim 2 is indefinite. I rely upon Collabo’s interpretation understand the scope of the claim. Based on Collabo’s interpretation of the claim, Shikata, Crump and Peterson render obvious Claim 2.

327. For the reasons I discussed in **Section XII.B**, based on the disclosure of Peterson, a POSITA would be motivated to modify the clock generating unit 2 in the same manner as the clock generating unit in Lee. When implementing Shikata’s system in view of Crump, it would have been obvious to incorporate Peterson’s teaching of halting the clock signal only after validating that the clock stop control signal adheres to a predetermined protocol requirement.

2. Motivation to combine Shikata and Crump with Peterson

328. For the reasons discussed in **Section XII.B**, it is my opinion that a POSITA would be motivated to modify the clock generating unit 2 in Shikata based on the disclosure of Peterson. Integrating Peterson’s teachings into Shikata’s systems implemented in view of Crump would have been a combination of prior art elements

according to known methods to yield predictable results. Peterson's approach to disabling the clock signal following protocol verification was an understood technique, one that a POSITA would have found obvious to apply to Shikata's systems implemented in view of Crump. In my opinion, a POSITA would have particularly recognized the advantage of minimizing accidental clock stoppage risk at the time of the invention of the '575 Patent.

329. In my opinion, a POSITA would have had a reasonable expectation of success in incorporating Peterson's teaching. The method of disabling the clock signal subsequent to verifying the clock stop control signal was a well-established means to circumvent unintended clock stoppage.

G. Ground VII: Shikata, Crump and Kang render obvious Claim 3

1. Claim 3

330. Claim 3 provides: "*The microcontroller according to claim 1, wherein the power supply control unit limits a period in which shutdown is possible.*"

331. I have reviewed **Section IX.G** of the Petition.

332. In my opinion, for the reasons discussed below, a POSITA would be motivated to incorporate the functionality disclosed in Kang in the processor 100 disclosed in Shikata. Mirroring the system disclosed in Kang, the voltage controlling unit 1 would be modified to ensure that power shutdown signal is not sent to various

circuit blocks at certain time durations, for example, durations when the processor 100 is expected to be active.

2. Motivation to combine Shikata and Crump with Kang

333. In my opinion, a POSITA would be motivated to incorporate the teaching of Kang to modify the processor 100 disclosed in Shikata for the reasons discussion in **Section XII.C** above. When implementing Shikata's system in view of Crump for a cellphone, it would have been obvious to incorporate Kang's teaching that the system should not be shut down during certain time slots, thereby preventing the cellphone from missing incoming messages. Ex. 1011 (Kang), 1:19-22. In my opinion, incorporating Kang's teaching in Shikata's systems implemented in view of Crump would have been a combination of prior art elements according to known methods to yield predictable results. Kang's strategy of ensuring a cellphone remains active during specific time slots was a known method that a POSITA would have found obvious to apply within Shikata's systems implemented in view of Crump. Specifically, a POSITA would understand that periodic monitoring of the paging channel is crucial for ensuring a cellphone receives incoming messages.

334. In my opinion, at the of the invention of the '575 Patent, a POSITA would have had a reasonable expectation of success in incorporating Kang's teaching because such a modification to Lee would be a combination of known elements in a known and

well-understood manner. Periodic monitoring of the paging channel was widely acknowledged as an effective method for securing message reception on a cellphone.

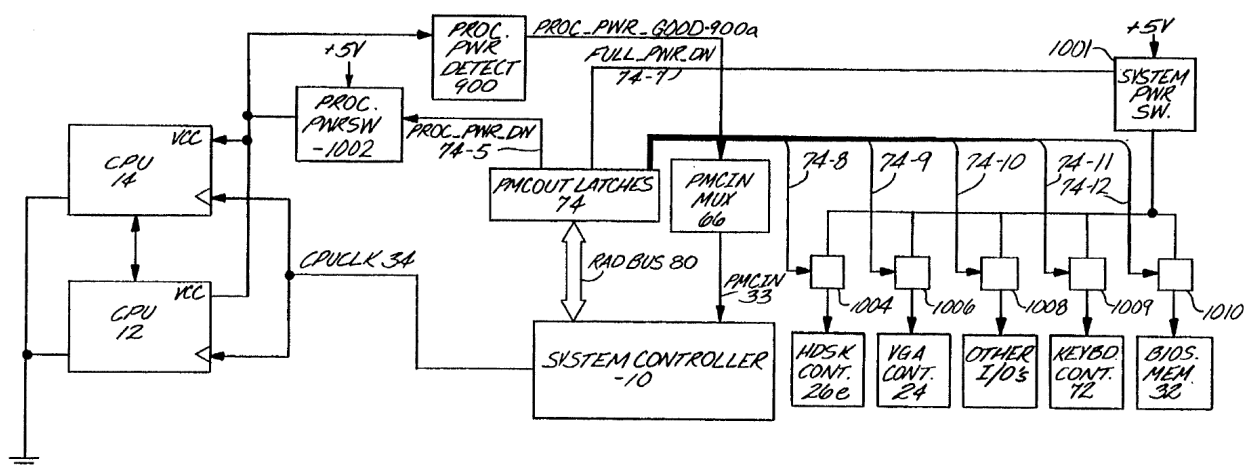
H. Ground VIII: Shikata, Crump and Chung render obvious Claim 16

1. Claim 16

335. Claim 16 provides: “*The microcontroller of claim 1, wherein the power supply unit does not shut down the power supply at least to the clock generator upon receiving the power shutdown control signal from the power supply control unit.*”

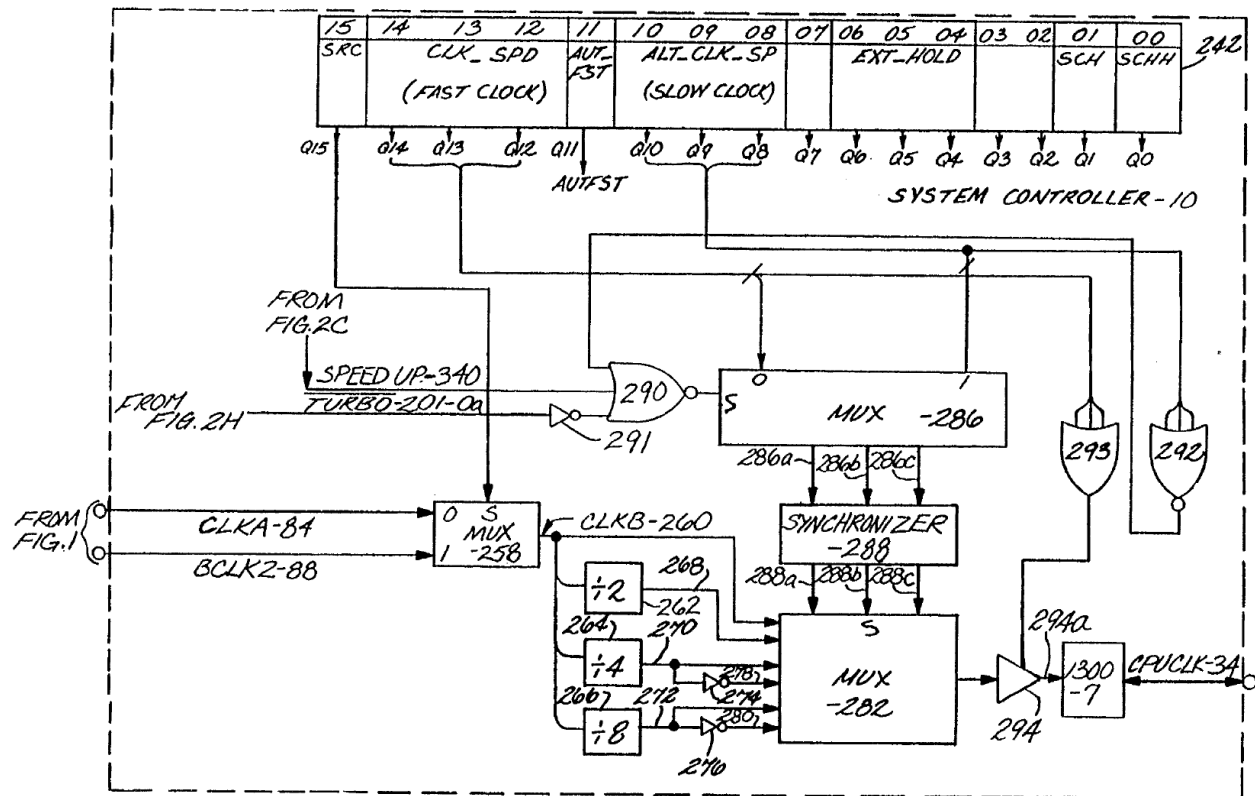
336. I have reviewed **Section IX.H** of the Petition.

337. As depicted in Fig. 8 of Chung, when the CPU 14 or 12 is entering the shutdown mode, it sets signals in the system controller 10 inputs, and the system controller 10 “initiates the power-down sequence.” Ex. 1012 (Chung), 31:52-32:6. Chung discloses that the PMCOOUT latches 74 are updated to command the CPU power supply to be disabled by the power switch 1002. Ex. 1012 (Chung), 32:14-19.



Ex. 1012 (Chung), Fig. 8.

338. In Chung, the power switch 1002 enables and disables the power supply to the CPUs. Power switch 1001 supplies power to system controller 10 and “the circuits connected to the output of CPUCLK control register 242 [depicted in Fig. 2E of Chung] form a clock generator.” Ex. 1012 (Chung), 20:1-3. In Chung, the clock generator continues to receive power but the tristated drivers are used to inhibit the clock from being supplied to the CPU. Ex. 1012 (Chung), 32:7-13. Thus, when the power to the CPUs is disabled, the clock generator continues to receive power though the supply of the clock to the CPUs is inhibited.



Ex. 1012 (Chung), Fig. 2E

339. In my opinion, for the reasons described below, a POSITA would augment the invention disclosed in Shikata to ensure that when the voltage controlling unit 1 issues a power shutdown control signal, it does not instruct the external variable voltage supply 101 to disable power supply to the PLL circuit 11 and clock generating unit 2. Further, as discussed in in **Section X11.E.2** above, a POSITA would modify the processor 100 in Shikata to ensure that the PLL 11 and clock generation unit 2 continue to receive power even though the processor 100 is entering the standby mode.

2. Motivation to combine Shikata and Crump with Chung

340. In my opinion, at the time of the invention of the '575 Patent, a POSITA would understand that there are two primary approaches to inhibiting the supply of clock to a circuit module: first, disabling the clock generator and associated PLL circuits by removing power so that clocks are not generated; and second, inhibiting supply of clock to the circuit module by using a tristate gate or an AND gate, though the clock generator and the PLL continue to generate the clock. A POSITA would understand that the latter approach is more suitable for circuits that frequently transition into and out of standby mode. A POSITA would understand that when power is supplied again to a clock generator and associated PLL circuits, it takes time for the PLL circuits to latch on to the incoming clock signal and to then generate stable clock

signals that the circuit modules can use safely. This introduces a delay in restarting the processor that was placed in the shutdown mode because digital circuits cannot operate without a steady and reliable clock signal. On the other hand, if the supply of the clock was inhibited using the tristate gate or an AND gate, then resupply of clock signals simply requires changing input to these gates. The second approach shortens the time frame needed to restart when exiting out of the standby mode. A POSITA would understand that continuing to supply power to the clock generator and PLL circuits would incur additional power consumption. However, the power consumption by PLL is typically quite small compared to power consumption of a processor. Moreover, if the processor can enter and exit the standby mode quickly, it is more likely to leverage the standby mode more often. This would reduce power consumption further, and such savings would be sufficient to overcome the increased consumption from the clock generator and the PLL circuits.

341. As I discussed in **Sections VII.B.2** (discussing use of clocks) and **VII.B.3** (discussing voltage and clock scaling) above, use of clocks in digital design was well-known and well understood by the time of the invention of the '575 Patent. In addition, in my opinion, a POSITA would have appreciated the tradeoffs associated with turning off and on power supply to a PLL circuits and a clock generator. As a result, modification of processor 100 in Shikata to incorporate techniques of disabling

and enabling of clock supply, as disclosed in Chung, would be a combination of known elements in prior art and a simple substitution of one element for another to yield predictable results. In addition, the proposed modification of processor 100 disclosed in Shikata would be based on applying known techniques (e.g., managing clock supply) to a known device (e.g., the processor 100) to improve the device. As I discussed in **Section VII.B.2** (discussing use of clocks) above, use of clocks is indispensable in digital designs, such as the processor 100 of Shikata and circuits disclosed in Crump and Chung. Thus, a POSITA would be motivated by the knowledge of the skilled artisan to choose from a finite number of identifiable and predictable solutions. Finally, a POSITA would have a reasonable expectation of success in incorporating the abovementioned features disclosed in Chung to implement and improve the processor 100 disclosed by Shikata, as tri-stating clock signals was recognized as a routine and standard procedure.

I. Alleged Secondary Considerations

342. I reserve the right to address any other alleged evidence of secondary considerations of non-obviousness that Patent Owner presents later.

XIII. CONCLUSION

343. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and

Declaration of Dr. M. Annavaram
U.S. Patent No. 7,930,575

further 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.

Respectfully submitted,



Dated: May 20, 2025

Murali Annavaram, Ph.D.

APPENDIX A: THE CHALLENGED CLAIMS OF '395 PATENT

What is claimed is:

1. A microcontroller comprising:
 - a CPU;
 - a power supply unit arranged between the CPU and a power supply device
for supplying power to the CPU;
 - a power supply control unit for controlling the CPU and the power supply
unit;
 - an information holding unit for holding information evacuated from the
CPU, the information being necessary in proceeding with a program;
 - and
 - a clock generator; wherein:
 - the power supply control unit outputs a shutdown request signal to the CPU
in response to an occurrence of a power shutdown factor,
 - the CPU, upon receiving the shutdown request signal, executes a power
shutdown microprogram, evacuates the information necessary in
proceeding with the program to the information holding unit, and
outputs an evacuation completed signal to the power supply control
unit after evacuation is completed,

the power supply control unit, upon receiving the evacuation completed signal, outputs a power shutdown control signal to the power supply unit,

the power supply unit shuts down power supply to the CPU upon receiving the power shutdown control signal from the power supply control unit,

the power supply control unit, upon receiving the evacuation completed signal, outputs a clock stop control signal to the clock generator,

the power supply control unit outputs a power supply control signal to the power supply unit in response to an occurrence of a power supply restoration factor,

the power supply unit, upon receiving the power supply control signal, starts to supply power to the CPU,

the power supply control unit outputs a restoration request signal to the CPU and outputs a clock supply control signal to the clock generator when a supplied power supply voltage is stabilized,

the clock generator, upon receiving the clock supply control signal, restarts to supply an operation clock, and

the CPU, upon receiving the restoration request signal, executes a power supply restoration microprogram, restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown, and then branches a process to an address indicated by a program counter to continue program execution from a shutdown state.

2. The microcontroller according to claim 1, wherein the clock generator selects one of operation performed after stopping supply of an operation clock and operation stop performed without stopping the supply of the operation clock as operation to be performed upon receiving the clock stop control signal from the power supply control unit.

3. The microcontroller according to claim 1, wherein the power supply control unit limits a period in which shutdown is possible.

5. The microcontroller according to claim 1, wherein the power supply restoration factor uses an interrupt system of the microcontroller.

8. The microcontroller according to claim 1, wherein one of the power shutdown microprogram and the power supply restoration microprogram is selectively executed in an alternative way, according to the generated power shutdown factor or a type of the selected power shutdown factor.

11. The microcontroller according to claim 1, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information holding when the power shutdown factor occurs.

12. The microcontroller according to claim 1, further comprising a circuit for loading an inverted value to a register in a circuit portion in a power shutdown range.

15. The microcontroller of claim 1, wherein the power supply unit includes a voltage regulator.

16. The microcontroller of claim 1, wherein the power supply unit does not shut down the power supply at least to the clock generator upon receiving the power shutdown control signal from the power supply control unit.

APPENDIX B: LIST OF MATERIALS CONSIDERED

Paper	Description
1	Petition for Inter Partes Review of U.S. Patent No. 7,930,575 (filed concurrently)

Exhibit	Description
1001	U.S. Patent No. 7,930,575 (the “’575 Patent”)
1002	File History of U.S. Patent No. 7,930,575 (“File History”)
1005	U.S. Patent No. 7,293,183 (“Lee”)
1006	U.S. Patent No. 7,664,970 (“Jahagirdar”)
1007	U.S. Patent No. 7,219,248 (“Shikata”)
1008	U.S. Patent No. 5,530,879 (“Crump”)
1009	U.S. Patent No. 7,167,991 (“Higashida”)
1010	U.S. Patent No. 5,502,689 (“Peterson”)
1011	U.S. Patent No. 7,369,815 (“Kang”)
1012	U.S. Patent No. 5,390,350 (“Chung”)
1013	Petition for <i>Inter Partes</i> Review, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-01523, Paper 2 (P.T.A.B. Sep. 16, 2019) (“Jahagirdar IPR Petition”)
1014	Patent Owner’s Preliminary Response, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-01523, Paper 9 (P.T.A.B. Jan. 3, 2020) (“Jahagirdar POPR”)
1015	Decision Denying Institution of <i>Inter Partes</i> Review, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-01523, Paper 18 (P.T.A.B. Mar. 27, 2020) (“Jahagirdar Institution Decision”)
1016	Petition for <i>Inter Partes</i> Review, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-0152, Paper 2 (P.T.A.B. Sept. 16, 2019) (“Chung IPR Petition”)
1017	Patent Owner’s Preliminary Response, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-01524, Paper 9 (P.T.A.B. Jan. 3, 2020) (“Chung POPR”)
1018	Decision Denying Institution of <i>Inter Partes</i> Review, <i>Advanced Micro Devices, Inc. v. Collabo Innovations Inc.</i> , IPR2019-

Exhibit	Description
	01524, Paper 13 (P.T.A.B. Mar. 30, 2020) (“Chung Institution Decision”)
1019	Complaint, <i>Collabo Innovations, Inc. v. Advanced Micro Devices, Inc.</i> , No. 1:18-cv-552, ECF 1 (W.D. Tex. July 2, 2018) (“AMD Complaint”)
1020	Claim Construction Order, <i>Collabo Innovations, Inc.</i> , No. 1:18-cv-552, ECF 93 (W.D. Tex. July 5, 2023) (“AMD Claim Construction”)
1021	Michael J. Flynn, <i>Computer Architecture: Pipelined and Parallel Processor Design</i> (1995) (“Flynn Textbook”)
1022	Excerpts from <i>The Modern Dictionary of Electronics</i> (7th Ed. 1999)
1023	M.V. Wilkes, <i>Microprogramming</i> , AIEE-ACM-IRE: Papers and Discussions Presented at the Dec. 3-5, 1958 Easter Joint Computer Conference (1958) (“Wilkes Paper”)
1024	M.V. Wilkes, <i>The Growth of Interest in Microprogramming: A Literature Survey</i> , 1 Computing Surveys 139 (1969) (“Wilkes Survey”)
1025	David A. Patterson, <i>Microprogramming</i> , 248 Sci. Am. 50 (1983)
1026	Mutunori Igarashi et al., <i>A Low-power Design Method Using Multiple Supply Voltages</i> , in <i>Proceedings of 1997 International Symposium on Low Power Electronics and Design</i> , Monterey, CA 36 (1997) (“Igarashi”)
1027	Jui-Ming Chang & Massoud Pedram, <i>Energy Minimization Using Multiple Supply Voltages</i> , 5 IEEE Transactions on Very Large Scale Integration (VLSI) Systems 436 (1997) (“Chang”)
1028	U.S. Patent No. 7,174,472 (“Hill”)
1029	Zhigang Hu et al., <i>Microarchitectural Techniques for Power Gating of Execution Units</i> , in <i>Proceedings of 2004 International Symposium on Low Power Electronics and Design</i> 32 (2004) (“Hu”)
1030	Shekhar Borkar, <i>Design Challenges of Technology Scaling</i> , 19 IEEE Micro 23 (July-Aug. 1999) (“Borkar”)
1031	Lawrence T. Clark et al., <i>Standby Power Management for a 0.18μm Microprocessor</i> , in <i>Proceedings of 2002 International</i>

Exhibit	Description
	<i>Symposium on Low Power Electronics and Design 7</i> (2002) (“Clark”)
1032	Luca Benini et al., <i>A Survey of Design Technologies for System-Level Dynamic Power Management</i> , 8 IEEE Transactions on Very Large Scale Integration (VLSI) Systems 299 (2000) (“Benini”)
1033	Resve Saleh et al., <i>System-on-Chip: Reuse and Integration</i> , 94 Proceedings of the IEEE 1050 (2006) (“Saleh”)
1034	Raj Nair & Bao Tang, <i>Low Cost On-Die CMOS Distributed Voltage Regulation for Microcontrollers</i> , in <i>Proceedings of the 24th European Solid-State Circuits Conference</i> 188 (1998) (“Nair”)
1035	L. Richard Carley & Akshay Aggarwal, <i>A Completely On-Chip Voltage Regulation Technique for Low Power Digital Circuits</i> , in <i>Proceedings of 1999 International Symposium on Low Power Electronics and Design</i> 109 (1999) (“Carley”)
1036	Sape J. Mullender, <i>Interprocess Communication: Computer Networks</i> , Ch. 1, 47 (1992) (“Mullender”)
1037	Howard Lee Morgan, <i>An Interrupt Based Organization for Management Information Systems</i> , 13 Communications of the ACM 734 (1970) (“Morgan”)
1038	A.C. Fischer-Cripps, <i>Newnes Interfacing Companion</i> (2002) (“Fischer-Cripps”)
1039	B Holdsworth & R.C. Woods, <i>Digital Logic Design</i> (2003) (“Holdsworth”)
1040	Jurij Silc et al., <i>Processor Architecture: From Dataflow to Superscalar and Beyond</i> (Springer 1999) (“Processor Architecture”)
1041	Preliminary Infringement Contentions, <i>Collabo Innovations, Inc. v. Qualcomm Inc.</i> , No. 6:24-cv-00472 (W.D. Tex. Dec. 6, 2024) (“Collabo’s Preliminary Infringement Contentions”)
1042	U.S. Patent No. 6,141,762 to Nicol et al. (“Nicol”)
1043	U.S. Patent No. 5,784,628 to Reneris (“Reneris”)
1044	U.S. Patent No. 6,804,632 B2 to Orenstien et al. (“Orenstien”)
1045	Excerpts from <i>The Authoritative Dictionary of IEEE Standard Terms</i> (7th Ed. 2000)

Exhibit	Description
1046	Excerpts from <i>Microsoft Computer Dictionary</i> (4th Ed. 1999)
1047	Joint Claim Construction Statement, <i>Collabo Innovations, Inc. v. Qualcomm Inc.</i> , No. 6:24-cv-00472-ADA (W.D. Tex.)
1048	Opening Claim Construction Brief, <i>Collabo Innovations, Inc. v. Qualcomm Inc.</i> , No. 6:24-cv-00472-ADA, Dkt. 32 (W.D. Tex.)
1049	Responsive Claim Construction Brief, <i>Collabo Innovations, Inc. v. Qualcomm Inc.</i> , No. 6:24-cv-00472-ADA, Dkt. 33 (W.D. Tex.)
1050	James Leiterman, <i>32/63-BIT 80x86 Assembly Language Architecture</i> , 2005 (“Leiterman”)
1051	U.S. Patent Application Publication No. 2004/0257898 (“Ozawa”)
1052	U.S. Patent No. 6,715,090 (“Totsuka”)