UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NXP USA, INC. and QUALCOMM INCORPORATED,

Petitioners,

v.

REDSTONE LOGICS LLC,

Patent Owner.

Case: IPR2025-00485
U.S. Patent No. 8,549,339

DECLARATION OF MURALI ANNAVARAM, Ph.D.

TABLE OF CONTENTS

TABLE OF CONTENTS 2
I. INTRODUCTION 4
II. PROFESSIONAL BACKGROUND 4
III. COMPENSATION 10
IV. LEGAL CONSIDERATION 10
A. OBVIOUSNESS 10
B. CLAIM INTERPRETATION 12
V. OVERVIEW OF TASK & BASIS FOR OPINIONS 13
VI. PERSON OF ORDINARY SKILL IN THE ART 14
VII. A POSITA'S UNDERSTANDING OF THE FIELD OF TECHNOLOGY 15
A. MULTI-CORE PROCESSORS 15
B. CLOCK GENERATION 17
C. DYNAMIC VOLTAGE FREQUENCY SCALING (DVFS) 21
D. PROCESSOR CACHE MEMORY AND CACHE COHERENCY ..25
VIII. OVERVIEW OF THE '339 PATENT 28
IX. IEEE PUBLICATION PRACTICES RELEVANT TO THE WILLIAMS PAPER 32
X. GROUNDS CHALLENGING THE '339 PATENT 33
A. GROUND 1: WHITE AND TALWAR RENDER OBVIOUS CLAIMS 1-3, 5, 8, 10-11, 14, AND 21 33
1. White 33
2. Talwar 38
3. Motivation to Combine 42
4. Modification of White in View of Talwar 45
5. Element-by-Element Analysis 49
B. GROUND 2: WHITE, TALWAR, AND CHO RENDER OBVIOUS CLAIMS 5, 14 67
1. Cho 67
2. Motivation to Combine White/Talwar with Cho 68

3. Modification of the White/Talwar Combination in View of Cho 70
4. Element-by-Element Analysis 72
C. GROUND 3: WHITE, TALWAR, AND VAJDA RENDER OBVIOUS CLAIMS 8-9.....73
1. Vajda 73
2. Motivation to Combine White/Talwar with Vajda.....73
3. Modification of the White/Talwar Combination in View of Vajda 74
4. Element-by-Element Analysis 75
D. GROUND 4: ELGEBALY AND OZER RENDER OBVIOUS CLAIMS 1-3, 5, 8, 10-11, 14, AND 21 76
1. Elgebaly 76
2. Ozer.....83
3. Motivation to Combine Elgebaly and Ozer 87
4. Modification of Elgebaly in View of Ozer 89
5. Element-by-Element Analysis 92
E. GROUND 5: ELGEBALY, OZER, AND VAJDA RENDER OBVIOUS CLAIMS 8-9.....110
1. Motivation to Combine Elgebaly, Ozer, and Vajda 110
2. Modification of Elegbaly/Ozer in View of Vajda 111
3. Element-by-Element Analysis 111
F. GROUNDS 6-7: GROUNDS 4-5 AND WILLIAMS RENDER OBVIOUS CLAIMS 1-3, 5, 8-11, 14, AND 21 112
XI. CONCLUSION.....113

I. INTRODUCTION

1. My name is Dr. Murali Annavaram, Ph.D. I have been retained by the Petitioners as an independent expert consultant in this inter partes review ("IPR") proceeding before the United States Patent and Trademark Office ("PTO") regarding US Patent No. 8,549,339 ("the '339 Patent").

II. PROFESSIONAL BACKGROUND

2. All of my opinions stated in this Declaration are based on my own personal knowledge and professional judgment. 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 EX1003. The following provides a brief overview of some of my experience that is relevant to the matters set forth in this Declaration.

4. I am an expert in the field of computer architecture, including power efficient multi-core computer architecture and design. 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 Gopalakrishnachar 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 power efficient hardware 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 only one of 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 only one of 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 multi-core 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 is 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, 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, 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, GPGPUsim, for power and performance evaluations.

III. COMPENSATION

21. 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 CONSIDERATION

22. I am not an attorney. My understanding of the law is based on information provided by counsel for Petitioners.

A. Obviousness

23. 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 that the invention of the ’339 Patent (a “POSITA”), which I have been asked to treat as February 26, 2010.

24. 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. 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.

25. 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. 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. 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 inventor of the ’339 Patent or any assignee of the ’339 Patent that any secondary considerations tend to rebut the obviousness of the ’339 Patent.

B. Claim Interpretation

26. 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.

V. OVERVIEW OF TASK & BASIS FOR OPINIONS

27. I have been asked to review the ’339 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 February 26, 2010, 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 paragraphs 31-32 below.

28. In preparing this declaration, I have considered this patent and its prosecution history as well as the general knowledge of those familiar with the field of technology relating to multi-core processors, which includes techniques for generating and managing distinct clock signals, power management techniques such as dynamic voltage frequency scaling, intra-chip communication techniques, efficient memory hierarchy designs such as caches and cache coherency protocols, and common semiconductor die layouts.

29. 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 Exhibit List included with the Petition, and any other items that I reference in my below analysis.

30. 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.

VI. PERSON OF ORDINARY SKILL IN THE ART

31. A person of ordinary skill in the art ("POSITA") pertinent to the '339 Patent would have had at least (1) a bachelor's degree in electrical engineering, computer engineering, computer science, or a similar field, and (2) at least two years of academic or industry experience designing or analyzing electronic circuits, semiconductors, processors, or power management, and related firmware and software, or the equivalent. The prior art also evidences this level of ordinary skill. Individuals with different education and additional experience could still be of ordinary skill in the art if that experience compensates for a deficit in their education stated above. Additional education might substitute for some experience, and vice versa.

32. Here, the background technology described in Section VII and the prior art described in Section X demonstrate that a POSITA would have been familiar with the underlying principles of multi-core processor design, including known techniques for generating and managing distinct clock signals, power management techniques such as dynamic voltage frequency scaling, intra-chip communication techniques, efficient memory hierarchy designs such as caches and cache coherency protocols, and common semiconductor die layouts.

VII. A POSITA'S UNDERSTANDING OF THE FIELD OF TECHNOLOGY

A. Multi-core Processors

33. Multi-core processors were well known prior to the priority date of the '339 Patent. EX1019, 52-53/219. By 2008, multi-core processors were already "becom[ing] the predominant general-purpose, high-performance microprocessor platform." EX1019, 52/219. For example, dual and quadcore processors were commercially available as early as 2006 and processors with 8, 16, and 32 cores were being designed and studied throughout the 2000s. *See e.g.,* EX1023, 2/8; EX1007, FIG. 1, [0014]; EX1020, 1:16-20; EX1021, 2:53-57, FIG. 1; EX1022, FIG. 1. Multi-core processors "are typically used to gain higher performance by arranging the different processor cores to execute respective data processing operations in parallel" by "shar[ing] access to shared memory." EX1013, [0004].

34. By 2010, "[i]ndependent DVFS for each core," cluster or "domain"-based DVFS, and "per-chip DVFS" had already become well known. EX1019, 52-53/219; EX1006, 2:15-44; EX1004, [0032]. These options all presented known trade-offs: per-core DVFS enables the most precise adjustments and most power and heat efficiencies, but "per-core DVFS is considered more expensive; it requires more than one power/clock domain per chip, and synchronizer circuits are required to avoid metastability between domains." EX1019, 52-53/219. Balancing costs of component and limited die space against the desire to set operating conditions

granularly, it was well known to group cores into a "cluster" or "domain" that "share a common clock and/or a common supply voltage that may be jointly set for these cores." EX1007, [0011]; EX1004, [0033].

35. Multi-core processors feature a number of known design constraints that must be balanced to create an effective design, including power consumption, high temperatures that create overheating risks, high system complexity that limits performance, and limited chip space that restricts the number of components that may be cost-effectively included on any given chip. *See e.g.,* EX1006, 1:14-26 (limited chip space); EX1029, 4 (same); EX1006, 1:27-52 (overheating limiting performance); EX1011, [0063] (same); EX1021, 1:32-40 (same); EX1007, [0002]-[0004] (desire to minimize power consumption); EX1004, [0006] (same); EX1013, [0010] (same).

36. Of particular relevance to this Petition, it was known in the art that circuitry to support "the regulation of operating voltage for integrated circuits, such as microprocessors" requires "large values of capacitance which may consume inordinate amount of real estate on an IC die." EX1006, 1:14-26; *see also* EX1024, 1/8 (describing how "conventional" voltage regulator designs would require too much space on the motherboard to be "practical" for 2000s-era microprocessors requiring high currents and well-regulated voltages). In part because of the amount of space required for voltage regulator circuitry, multi-core processors often "don't

have the same number of individual voltage sources [as] processors." EX1007, [0003]. However, restricting the number of voltage sources to a single source is also sub-optimal since that would force all the components on a chip to operate at a single voltage. *See also* EX1007, [0004], EX1019, 52-53/219. It is well known in art that a small number of heterogeneous mix of voltages (and correspondingly a few frequencies) would yield the best design. *See e.g.,* EX1007, [0011], FIG. 1, EX1019, 53/219 (describing the benefits of independently scaling the supply voltage and frequency of each core).

B. Clock Generation

37. The basic design principles for processor clock signals, including their generation, distribution, and synchronization, were widely known by 2010. The basic design to generate a single clock signal uses an external reference oscillator (e.g., crystal oscillator) paired with a phase locked loop (PLL) to multiply or divide the reference oscillator's signal to reach the desired clock frequency. EX1028, 2:61-3:11. Other designs are also common. For example, a clock signal may be provided via one or more of: (a) an external reference oscillator (e.g., crystal oscillator), (b) a PLL, and (c) "clock trees or other circuitry configured to split or divide a clock signal into multiple individual signals." *Id.*, EX1006, 7:23-45.

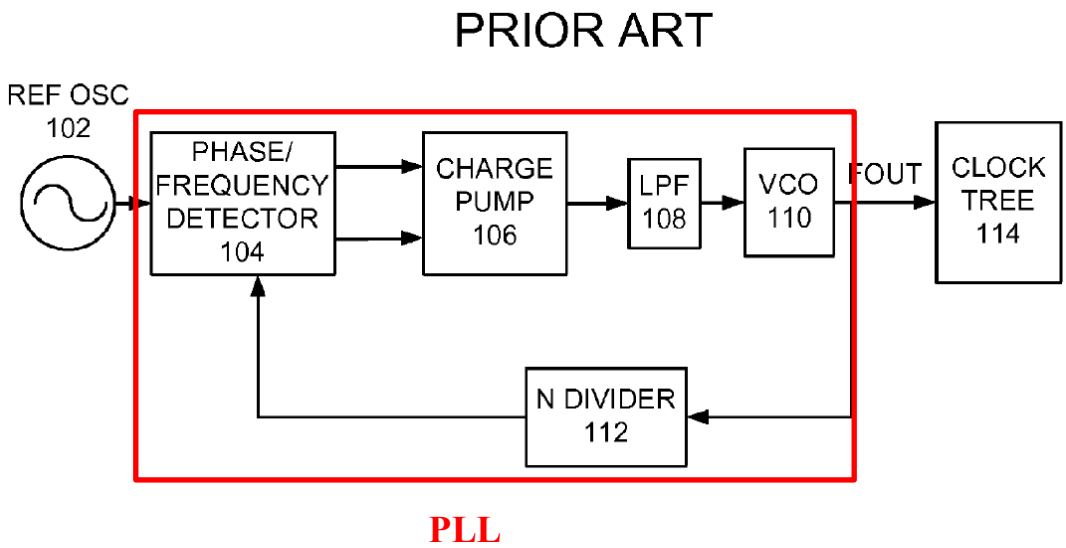


FIG. 1

EX1028, Figure 1 (annotated). As shown in the figure above, the key components of a PLL are a phase/frequency detector, charge pump, low-pass filter (LPF), voltage-controlled oscillator (VCO), and N divider that work together to align the output signal's phase with the phase of the reference crystal oscillator. *Id.* The basic PLL clock design generates a clock signal at a single frequency.

38. In addition to the basic PLL clock design, it was well known in the art that multiple PLLs, multiple crystal oscillators, or a combination of both may be used to create multiple clock signals at different frequencies. *See e.g.,* EX1004, [0028]; EX1006, 7:23-45; EX1008, 1-2; EX1027, [0005]-[0010]. For example, it was known that "[t]he simplest way to implement multiple clock signals is to provide multiple oscillators." EX1027, [0005]. Such a design is useful, for example, in a

processor where different functional units run at different speeds using different clock frequencies, also known as operating in different clock domains. *See* EX1008, 1-2. It was well known in the art that it was beneficial for some components of the processor to operate in one clock domain, while other components operate in a different clock domain. *Id.* The use of multiple clock domains within a processor allows certain components, such as memory, to run at lower frequencies and consume less power, while other components, such as adders or other computational units to run at higher frequencies for improved performance.

39. An example of a known design disclosed in Williams to generate two independent clock domains is shown below:

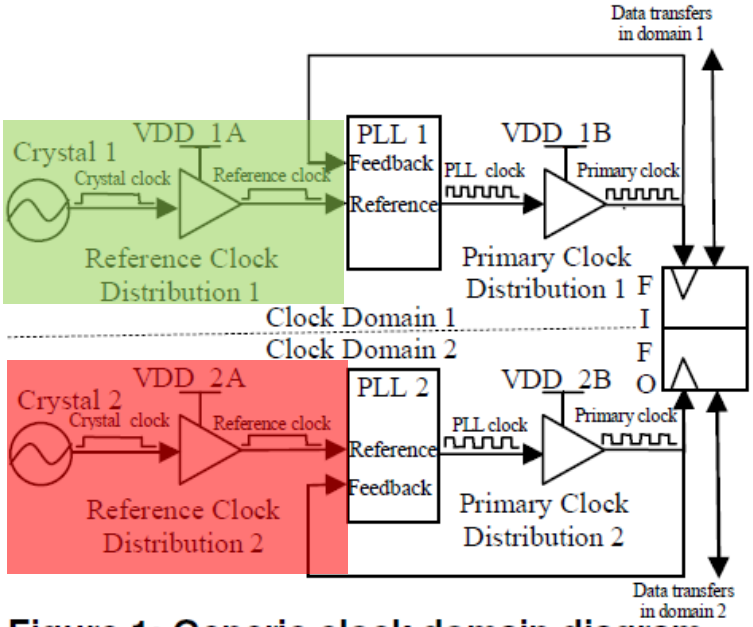


Figure 1: Generic clock domain diagram

Id., Figure 1. As shown in the figure, the two clock domains are "clocked by different PLLS . . . [that] run off reference clocks produced by different clock

generators and different crystal oscillators." *Id.*, 1. Specifically, the clock signal in "Clock Domain 1" is derived from the [green] "Reference Clock Distribution 1" circuit that includes "Crystal 1" and the independent clock signal in "Clock Domain 2" is derived from the [red] "Reference Clock Distribution 2" circuit that includes "Crystal 2." *Id.*

40. In addition to the multiple crystal oscillator, multiple PLL design above, it was also known in the art that clock signals at different frequencies may be generated using a clock tree or other circuitry configured to split or divide a single clock source into multiple signals at different frequencies. *See e.g.,* EX1006, 7:23-45; EX1027, [0027], FIG. 3. An example of this kind of configuration is depicted below:

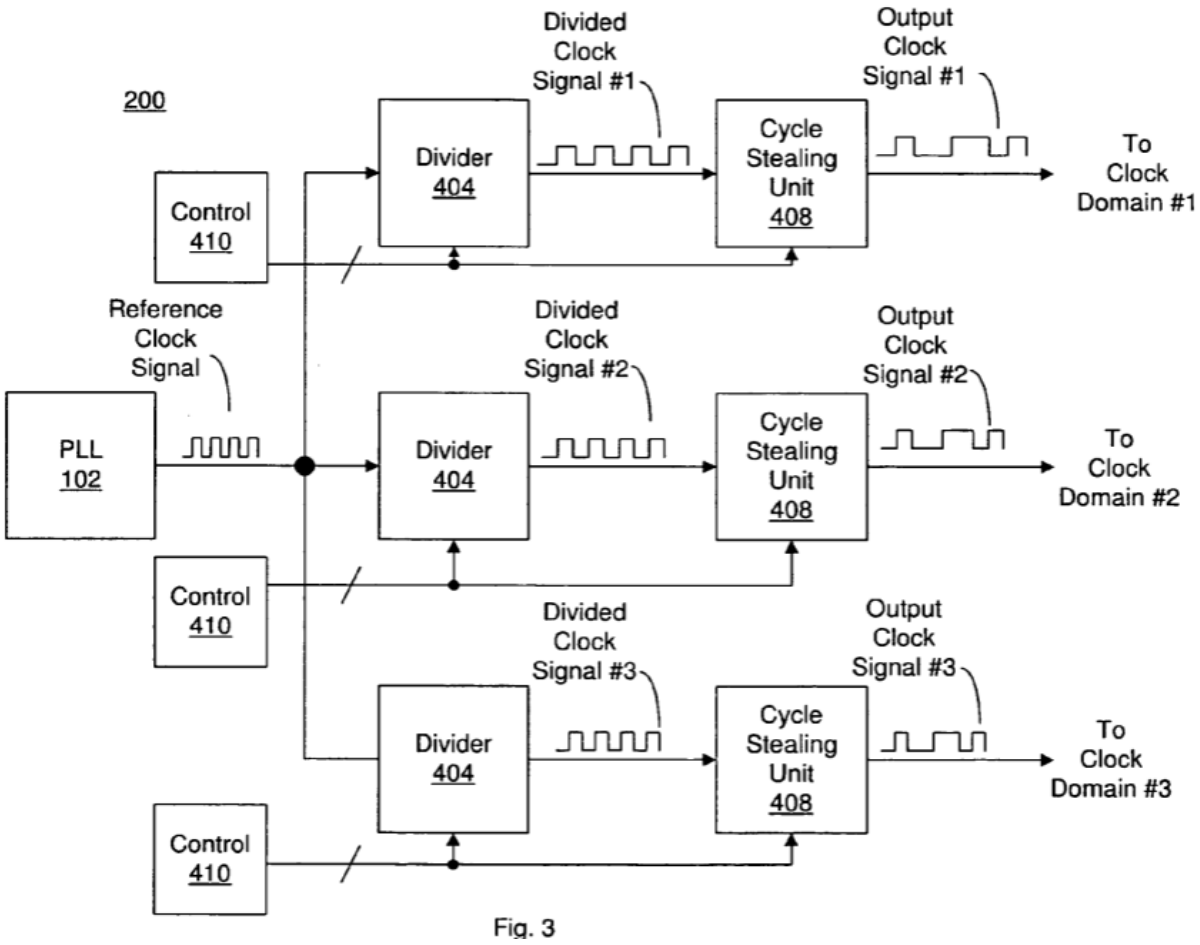


Fig. 3

EX1027, FIG. 3.

41. As explained further in the following section on DVFS, it was also known that each frequency has an ideal supply voltage to sustain that frequency and hence multiple clock domains may also benefit from having multiple such voltage sources.

C. Dynamic Voltage Frequency Scaling (DVFS)

42. Dynamic voltage and frequency scaling, or DVFS, describes a broad range of well-known prior art methods to reduce microprocessor power consumption

using "dynamic adjustments to supply voltage, clock frequency, or both." EX1019, 35/219. The amount of work a processor core can accomplish in a given period of time depends on clock rate, with higher clock rates resulting in more work. But a higher clock rate comes with costs; voltage must be increased to support the clock rate, increasing both power drain and heat. DVFS, "dynamic adjustments to supply voltage, clock frequency, or both," was well known to increase clock rate and voltage when needed for higher workloads, but also decrease them when possible to reduce power consumption and heat. EX1019, 35/219. By 2010, DVFS was "[t]he most popular technique for CPU power management." EX1007, [0002].

43. Fundamentally, DVFS techniques are derived from the basic power equation for processors: $P \propto CV^2Af$, where f is processor frequency, V is voltage, C is capacitance, and A is an activity factor related to how often transistors switch on a particular processor. EX1019, 35/219. This equation demonstrates that, "[i]f we can reduce voltage by some small factor, we can reduce power by the square of that factor," while keeping activity and capacitance unchanged. *See id.* However, "reducing supply voltage often slows transistors such that reducing the clock frequency is also required." *Id.*, *see also* EX1007, [0002] ("the frequency of the CPU is roughly linear in voltage"). Reducing a processor's clock frequency "may also linearly degrade performance," which can cause programs to "run[] at lower power dissipation levels, but for longer durations." EX1019, 35/219. As a result of

this trade-off, DVFS techniques focus on ways "to reduce clock frequency without having the workload experience a proportional reduction in performance." *Id.*, 36/219.

44. In multi-core processors using DVFS techniques, it was known to divide cores and other components into multiple voltage and/or clock domains that are adjusted independently. EX1019, 52-53/219; EX1006, 2:15-44; EX1004, [0032]. By 2010, "[i]ndepedent DVFS for each core," "per-chip DVFS," and "a mixture chip-wide DVFS and core allocation," were all known configurations of DVFS for multi-core processors. EX1019, 52-53/219. Additionally, it was known that "synchronizer circuits are required to avoid metastability between domains" operating at different clock frequencies." EX1019, 52-53/219; *see e.g.,* EX1032, FIGs. 1-3D, 1:55-2:4.

45. In addition to such a "[clock] synchronisation module," it was well known that a "voltage level shifter" is required "to deal with clock synchronisation issues and the different supply voltage levels (voltage [signaling] levels) between the two domains." EX1025, [0048]; *see e.g.,* EX1031, FIGs. 4-5, 5:45-57, EX1039, 1-2/4, Figures 2(a), 2(b), 3. "A logic-swing level shifter is a key circuit component" in DVFS implementations both because, often, "I/O blocks are operated under fixed V_{DD}" and each DVFS block on a chip could be operating under different voltages at any given point in time. EX1039, 1/4. Thus, it is possible that a high voltage level

signal of a first voltage domain may be recognized as a low voltage level by a second voltage domain, and vice versa, resulting in communication errors. Level shifter circuits address this problem. An example of a conventional level shifter circuit that was known in 2005 is shown below:

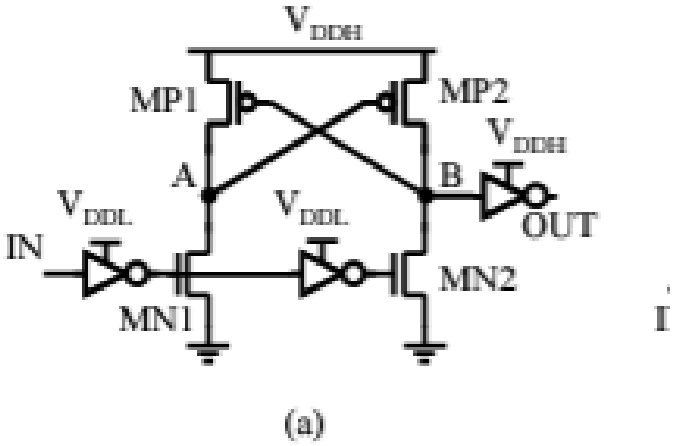


Figure 2 (a) Conventional level shifter

Id., 2/4. The conventional design shown in Figure 2(a) operates as follows: "[a]ssuming that initially input signal, IN, is '[Low, or a digital 0]' MN1 is turned on and provides a conducting path to ground while MN2 is cut off. Therefore, node A is pulled down to ground, which makes MP2 on and node B is pulled up to '[High, or a digital 1].' Thus, the output signal, OUT, becomes '[Low, or a digital 0].'" *Id.*, 1/4. If IN is a digital 1 in the first voltage domain (i.e., V_{DDL}), then node B is pulled down to ground, before being inverted to V_{DDH} so that OUT reflects a digital 1 in the second voltage domain (i.e., V_{DDH}). *See id.* Thus, voltage signals used by the circuits

in the first voltage domain can be understood by circuits in the second voltage domain. Similar designs were well known in the art and an essential feature of chip designs that use multiple voltage domains.

D. Processor Cache Memory and Cache Coherency

46. Processor designers have long used memory storage hierarchies, otherwise known as caches, to compensate for slow memory access speeds which may limit the performance of applications that have to access the data in slower memory. *See* EX1026, 1/7; EX1030, 1/14. By 2010, it was well known that multiple levels of caches was an optimal design, balancing the trade-off between memory speed and memory size. *See e.g.,* EX1006, FIGs. 1, 8. For example, to keep up with ever increasing processor clock speeds and "[t]o further improve speed of access to data within such a multi-processing system, it is known to provide each of the processing units with its own local cache." EX1013, [0005]. An example of this cache design, also known private caching, is shown in the figure below:

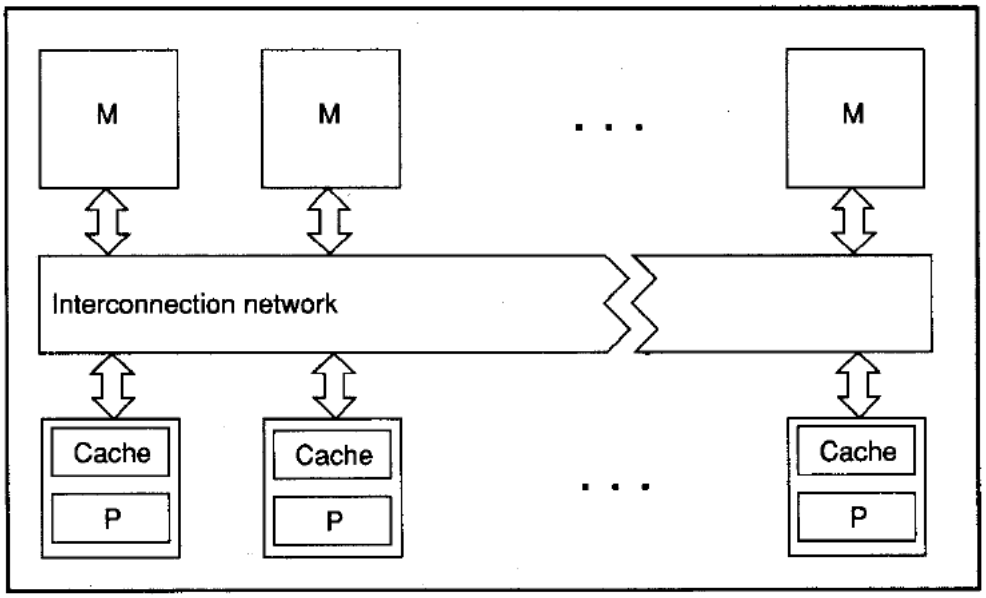


Figure 2. An example of a multiprocessor with private caches.[1]

EX1030, Figure 2. As shown in the figure, each core may have a local cache that is unique to each processor core (P) and not directly accessible by other cores, hence the local cache is also termed as a private cache. EX1030, 1/14; *see also* EX1013, FIG. 1A. In addition, to allow the multiple processor cores to execute data processing operations in parallel, it was known for the local caches to "share access to shared memory." EX1013, [0004], *see e.g.,* EX1023 5/8; EX1006, FIGs. 1, 8. Shared memory has the added benefit that it allows multiple, "low-cost microprocessors [to be] economically interconnected with shared memory modules," making this design particularly "cost-effective." EX1030, 1/14.

47. Because prior art multi-core processors included both shared main memory and dedicated local caches, it became "important to ensure that those processors will access the up-to-date data when seeking to access the associated

address in shared memory." EX1013, [0006]; EX1030, 1/14. The process of ensuring all processors operate using up-to-date data is known as cache coherency and is an essential feature of processor designs that use a shared memory. *See* EX1013, [0006]-[0009]; EX1030, 1/14.

48. Cache coherency schemes for multi-core processors have been well known since at least 1990. *See* EX1030, 1-2/14. Hardware-based cache coherency protocols were also well known since at least 1990, including "write-invalidate" and "write-update" schemes. EX1030, 3/14. In a write-invalidate scheme, when a first processor writes to a memory address, any copies of that memory address stored in the other processors' local caches are invalidated. *Id.* In a write-update scheme, by contrast, instead of invalidating the copies held in the other processors' local caches, the system updates them to reflect the modified value written to the memory address by the first processor. *Id.* For example, the cache of the processor that writes to the shared data may send the updated copy of the data to any other cache that holds a local copy of that data along a bus.

49. Both write-invalidate and write-update protocols "require that cache invalidation and update commands . . . be sent to at least those caches having copies of the [data] block." *Id.*, 4/14. One such approach to transmitting these invalidation and update commands is to use protocols called "snoopy cache protocols because every cache 'snoops' on the network for every incoming consistency command."

Id., see also EX1013, [0009]. A bus on the microprocessor is one example of a common "network" used to implement snooping operations. EX1030, 4/14; *see e.g.,* EX1013, FIG. 1A. Alternate "network" configurations are also well known, such as mesh, ring, and hyper-cube topologies.

VIII. OVERVIEW OF THE '339 PATENT

50. The '339 Patent relates to "power management for a multi-core processor" and related "techniques for handling communication between processor cores." EX1001, 1:58-60, Abstract. Specifically, the '339 Patent teaches that power consumption in a multi-core processor can be reduced using "dynamic supply voltage and clock speed control" by "operat[ing] at high power and high clock frequency when needed and at low power when the computing requirements are reduced." *Id.*, 1:10-14. Further, dynamic voltage and clock speed control can be applied separately to different groups of processor cores, or "stripes," (green, red) within the multi-core processor. *Id.*, 2:24-31, FIG. 1. For example, "[b]ased on the measured operational information (e.g., supply voltage and local temperature) and the performance data, the power control block 108 may then be configured to select a supply voltage for each strip." *Id.*, 2:50-54.

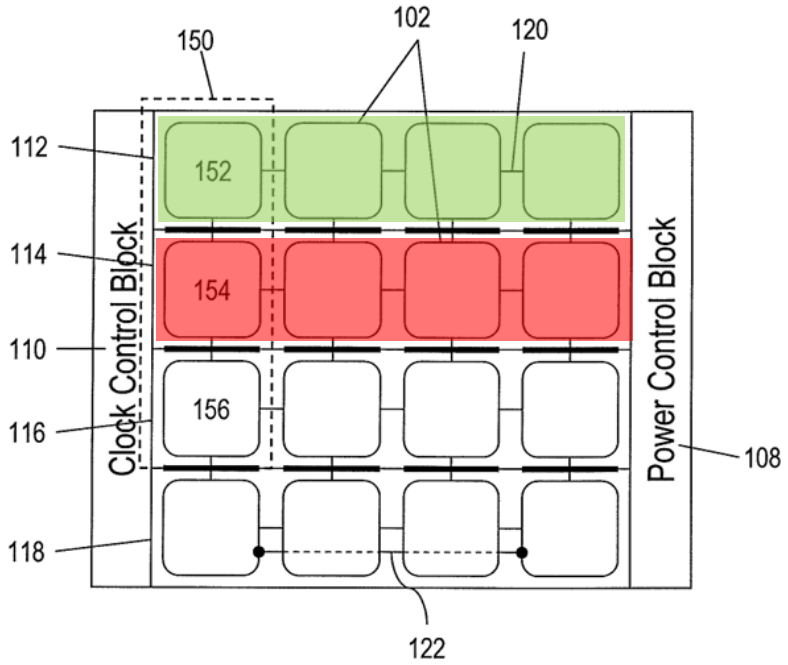


FIG. 1

Id., FIG. 1 (annotated). Figure 1 depicts a multi-core processor divided into rows of processor cores, or "stripes 112 [green], 114 [red], 116, and 118" that "may be associated with an independent power profile." *Id.*, 2:25-27. By contrast, the '339 Patent asserted that "[e]ach processor core in a conventional multi-core processor generally shares the same supply voltage and clock signal to simplify the interfaces between the processor cores." *Id.*, 1:7-10.

51. The '339 Patent associates each stripe with an "independent power profile" to address this perceived problem in the prior art. EX1001, 2:26-27, 1:10-14. A "power profile may include . . . power-supply voltages . . . clock

rates . . . clock multipliers . . . power throttling . . . and/or sleep state cycles of the core processor." *Id.*, 1:62-2:3. For example, each "stripe," or set of processors "may be powered by a supply voltage received from a power control block 108 and/or may be associated with an independent clock domain defined by a clock signal received from a clock control block 110." *Id.*, 2:28-31. Each stripe's power profile "may be determined based on the computational requirements of the tasks assigned to the processor cores in the stripe." *Id.*, 2:41-43. Thus, the '339 Patent's goal of "operat[ing] at high power and high clock frequency when needed and at low power when the computing requirements are reduced" can be achieved on a stripe-by-stripe basis. *Id.*, 1:10-14.

52. The '339 Patent also discloses an "interface block" to facilitate communications between processor cores operating at different voltage levels and using different clocks. *Id.*, FIGs. 2-3. By passing inter-processor communications through an "interface block," signals from different processors operating at different supply voltages and driven by independent clock signals can be "translate[d] . . . such that each of the processor cores operates correctly." *Id.*, 3:47-52, Abstract. This process is depicted in the figures below:

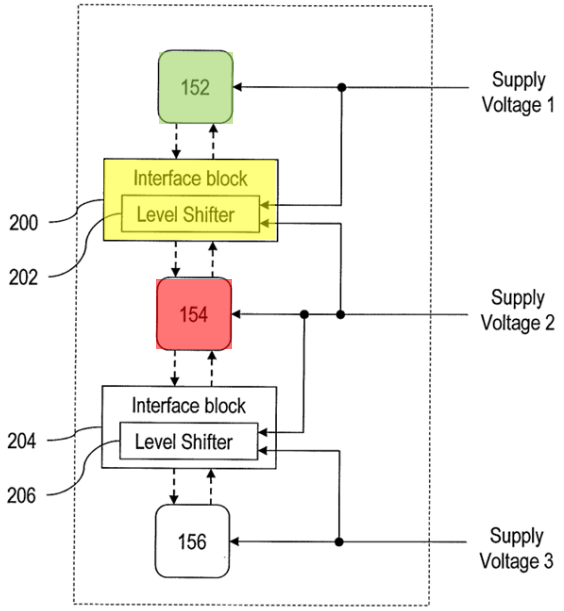


FIG. 2

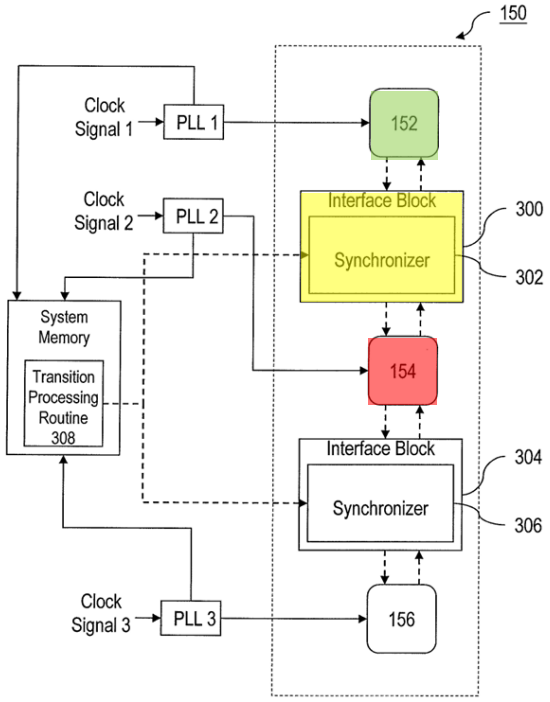


FIG. 3

EX1001, FIGs. 2, 3. In these figures, "the processor core 152 belongs to the [green] stripe 112; the processor core 154 belongs to the [red] stripe 114; and the processor core 156 belongs to the stripe 116." *Id.*, 3:21-26.

53. FIG. 2 (above) depicts a process for translating communications from cores operating a different voltage levels. As shown in the figure, "the level shifter 202 may be adapted to translate first logic levels associated with the [green] stripe 112 to second logic levels associated with the [red] stripe 114." *Id.*, 3:52-54. This translation allows each processor core to "properly interpret the voltages as valid logic levels even though [the] processor cores are powered by different supply voltages." *Id.*, 3:50-52.

54. FIG. 3 (above) depicts a process for translating communications from cores operating using different, independent clock signals. As described above, each stripe is "associated with an independent clock domain." *Id.*, 2:29-30. Being in different stripes, each processor core is clocked by the output of a PLL whose input is a different, independent clock signal: "clock signal 1," "clock signal 2," and "clock signal 3." *Id.*, 4:4-10. To "facilitate communication between two processor cores," the "[yellow] interface blocks" include "synchronizer 302" and "synchronizer 306." *Id.*, 3:21-24, 4:1-10, Abstract.

IX. IEEE PUBLICATION PRACTICES RELEVANT TO THE WILLIAMS PAPER

55. In addition to my review of the '339 Patent and the Petition, I have been asked to attest to my experience with the Institute of Electrical and Electronics Engineers (IEEE) and their publication practices. I have also been asked to review the Williams paper as downloaded from IEEE's website and its related metadata.

56. I am a Fellow of IEEE and my papers have been published through IEEE dozens of times, ranging from my first publication in 1996 to today. Many of these papers were presented publicly at a conference, before being published either on IEEE's website, in print, or both. In most cases, conference papers are publicly available on IEEE's website shortly (e.g., within a month) after the conference occurs.

57. I have reviewed the IEEE website page for the Williams paper (EX1008) and can confirm that it lists the paper as having been "added to IEEE *Xplore*" on March 26, 2007. *See also* EX1038, 1/2. In my experience, this means that the paper was publicly available on IEEE's website as of that date.

58. In addition, I have downloaded the Williams paper from the same page and reviewed the paper's PDF file metadata using Adobe Acrobat's File>Properties feature. This metadata shows that the paper was last modified on March 16, 2007 at 4:56 AM, two days after the end of the conference at which it was presented. *See* EX1040, 1/1. A screenshot of my review of the metadata is included with the Petition as Exhibit 1040. *See id.*

X. GROUNDS CHALLENGING THE '339 PATENT

A. Ground 1: White and Talwar Render Obvious Claims 1-3, 5, 8, 10-11, 14, and 21

1. White

59. White discloses a multi-core processor in which each core "may be configured to operate at frequencies and voltages independently of the other logic cores" based on various considerations including power and temperature management. EX1006, 2:15-44.

60. FIG. 2A illustrates the disclosed structure.

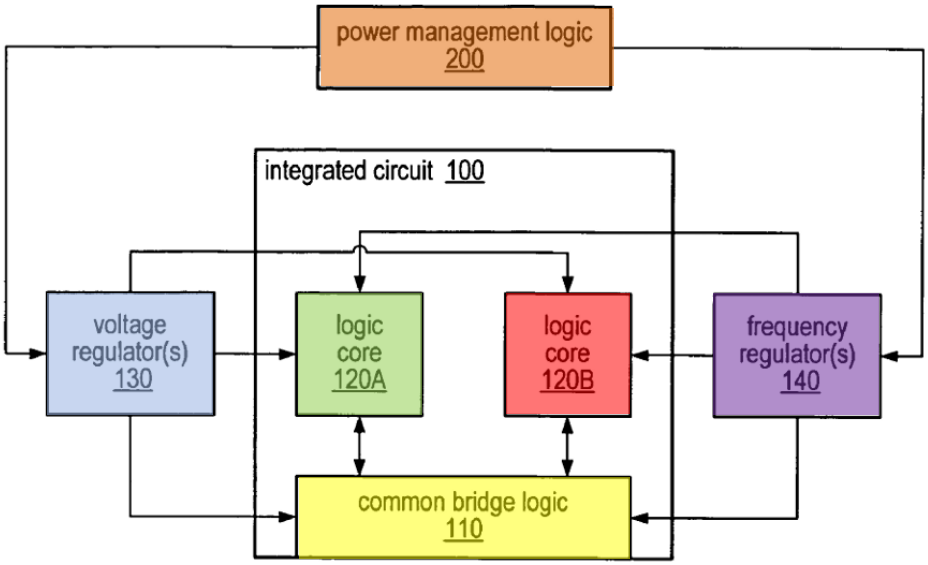


FIG. 2A

Id., FIG. 2A (annotated). "[Orange] power management logic 200" "control[s] the operating characteristics, such as the operating voltage and/or operating frequency, of each of multiple logic cores 120." *Id.*, 8:59-65. "[Blue] voltage regulator 130 . . . may be configured to supply independent voltages to each logic core," each of which can be "different from the voltage levels at which other logic cores are operated," as illustrated by the two separate lines extending from [blue] voltage regulator 130 to [green, red] logic cores 120A, 120B, respectively. *Id.*, 7:23-45, FIG. 2A. Similarly, "[purple] frequency regulator 140" may "supply multiple, independent clock signals" to the [green, red] logic cores 120A and 120B to allow "each to operate at frequencies and voltages independent of the frequencies or voltages that the other logic cores, or other components, may be operating." *Id.*, 6:41-49, 7:23-45. "[Orange] power management logic 200 may be configured to

communicate . . . with [blue] voltage regulator 130 to instruct voltage regulator 130 to supply a particular voltage to [green] logic core 120A" and "with [purple] frequency regulator 140 to instruct frequency regulator 140 to supply a particular frequency or clock signal to [green] logic core 120A." *Id.*, 10:11-24. Further, "as with power management logic 200, described above, temperature control logic 210 may, in some embodiments be configured to communicate or interface with voltage regulator 130 and frequency regulator 140 in order to adjust the voltages and frequencies supplied to logic cores 120A and 120B." *Id.*, 13:38-43. Accordingly, each logic core is arranged to receive an independent voltage and clock signal.

61. White discloses multiple options for generating and changing clock signals using a combination of elements at both frequency regulator 140 and the cores. Frequency regulator 140 includes "various sorts of clock sources, such as an oscillator or phase locked loop (PLL) to generate a reference clock" and uses those "multiple clock sources in order to supply multiple, independent clock signals of different frequencies." *Id.*, 7:38-45. Frequency regulator 140 may also include "clock trees or other circuitry configure to split or divide a clock signal into multiple individual signals." *Id.*, 7:40-43. Additionally, each logic core may also "generate its internal clock using a phase-locked loop (PLL) locked to an externally provided clock signal, such as may be provided by frequency regulator 140." *Id.*, 9:62-65. In other words, White discloses a clock system in which each core includes a PLL that

receives a different, independently-generated clock signal. The Petition describes in more detail a number of the configurations disclosed by White. Each of these configurations are discussed below.

62. In the First Configuration, frequency regulator 140 includes "multiple clock sources in order to supply multiple, independent clock signals of different frequencies." *Id.*, 7:40-43. Based on White's designation of "oscillators" as a clock source, a POSITA would understand these passages to disclose frequency regulator 140 using multiple external oscillator crystals "in order to supply multiple, independent clock signals of different frequencies" (*id.*, 7:40-45), each feeding "a phase-locked loop (PLL)" at one of logic cores 120A or 120B (*id.*, 9:62-65).

63. In the Second Configuration, frequency regulator 140 includes "multiple clock sources in order to supply multiple, independent clock signals of different frequencies" in the form of multiple PLLs, the output of each feeding "a phase-locked loop (PLL)" at one of logic cores 120A or 120B, or directly feeding the cores themselves. *See id.*, 7:43-45, 9:62-65. Based on White's designation of a "PLL" as another kind of clock source, a POSITA would understand these passages to disclose a configuration of the frequency regulator in which a separate PLL is used to supply each "independent clock signal" to each core. *Id.*, 7:38-40. Because PLLs require input clock signals to operate, a POSITA would further understand these passages to teach the use of a separate reference oscillator to generate the input

for each PLL in the frequency regulator, as was known in the art. *See e.g.*, EX1008, FIG. 1, *see also* Section VII.B. Fundamentally, PLLs cannot operate without a reference clock signal as an input.

64. In a Third Configuration, frequency regulator 140 includes "clock trees or other circuitry configured to split or divide a clock signal into multiple individual signals," each feeding a "phase-locked loop (PLL)" at one of logic cores 120A or 120B, or directly feeding the cores themselves. EX1006, 7:40-43, 9:62-65. A POSITA would understand these passages to disclose the well-known PLL and clock tree configurations in which a single external oscillator crystal feeds one or more PLLs along with a subsequent clock tree that includes dividers and selection circuitry, as needed, to generate multiple, independent clock signals of different frequencies.

65. White also discloses the use of "common bridge logic, such as a Northbridge,"—a well-known means for communicating between the cores of a multi-core processor and its memory—"configured to communicate with the logic cores as well as with other elements or peripherals . . . such as a memory subsystem." *Id.*, 7:1-7, 17:29-32, FIGS. 2A, 2B. Because the logic cores may be operating at different voltage levels and frequencies, White discloses that the "[c]ommon bridge logic 110" is "supplied with multiple voltages and/or frequencies" so that it can "interface with a logic core using the same logic levels

and/or clock timing as the logic core." *Id.*, 7:67-8:10. In addition, because different voltage levels and frequencies are generally incompatible with one another, common bridge logic further includes an interface "to translate the communication, such as the signals on one or more control lines, between the different logic levels and/or between the different frequencies." *Id.*, 8:21-26, FIG. 3. A POSITA would understand this to mean that the common bridge logic includes one of the known designs for voltage-level shifters and clock synchronization blocks to translate signals between the voltage levels and clock domains of each logic core. *See* Section VII.C, *see also* EX1031, FIGs. 4-5, 5:45-57, EX1039, 1-2/4, Figures 2(a), 2(b), 3

66. Finally, while FIG. 2A illustrates two logic cores, White explains that its teachings "may be applied to systems or devices that include any number of logic cores configured to operate at independent voltages and/or frequencies…[where] a processor may include multiple logic cores, or processor cores, configured to operate at frequencies and voltages independently of each other." *Id.*, 6:56-60; *see also* 6:63-67, Claims 1, 25 ("wherein at least one of the logic cores is configured to operate at frequency independent of a frequency at which another of the plurality of logic cores operates").

2. Talwar

67. Talwar discloses a multi-core processor design in which "[t]he cores and voltage sources are divided into clusters, whereby multiple cores in a cluster

receive power from a single voltage source" and "[e]ach cluster may have a different cluster frequency," but "all the cores in a cluster [have] the same frequency." EX1007, [0011], [0018]-[0019]. By providing a "single voltage source" to "multiple cores in a cluster," "the frequency of all cores in a cluster" and the single voltage source can be jointly adjusted "[d]uring power management." *Id.*, [0011]-[0012], [0018]-[0019], [0021]-[0025], [0030] ("The cores in a cluster, which receive power in one power domain, can be independently controlled from other clusters.").

68. Similar to both the '339 Patent and White, Talwar discloses the use of "[d]ynamic voltage frequency scaling (DVFS)" to reduce CPU power consumption, which is "[t]he most popular technique used for CPU power management." *Id.*, [0002]. DVFS "reduces the frequency [of the CPU] when the CPU utilization is below a certain threshold." *Id.*, [0003]. Additionally, as was well known in the art, "if the CPU frequency is reduced, the required voltage is reduced" because "the frequency of the CPU is roughly linear in voltage." *Id.*, [0002]. Talwar recognized, however, that this approach is "not as efficient in multi-core systems" because "they don't have the same number of individual voltage sources" as processor cores. *Id.*, [0004]. For example, if "one processor's frequency is scaled down, the voltage to the processor doesn't change because the other processor [on the same chip] is still running at a higher frequency and needs the higher voltage." *Id.*, [0004].

69. Talwar's design solved the problem that "in current multi-core systems, all the processors use a single voltage source which renders frequency scaling [power management] technique[s] often inefficient." *Id.*, [0003]-[0004]. By grouping the cores into "clusters," Talwar's design gains the power efficiency of multiple DVFS groups of cores per chip without requiring a unique voltage source for every single core. Additionally, Talwar's design scales to "any number of voltage sources and cores," while still requiring "less voltage sources than cores on the chip." *Id.*, [0014]. Providing one voltage (and clock) source per cluster enables efficiently performing DVFS on each cluster in "stark contrast to conventional multi-core chips" with only a single voltage source. *Id.*, [0012].

70. FIG. 1 depicts a "multi-core computer system 100" according to one embodiment of Talwar, including "multi-core chip 110" with "clusters (i.e., volt-cpu-sets) 111a-n." *Id.*, [0014], FIG. 1.

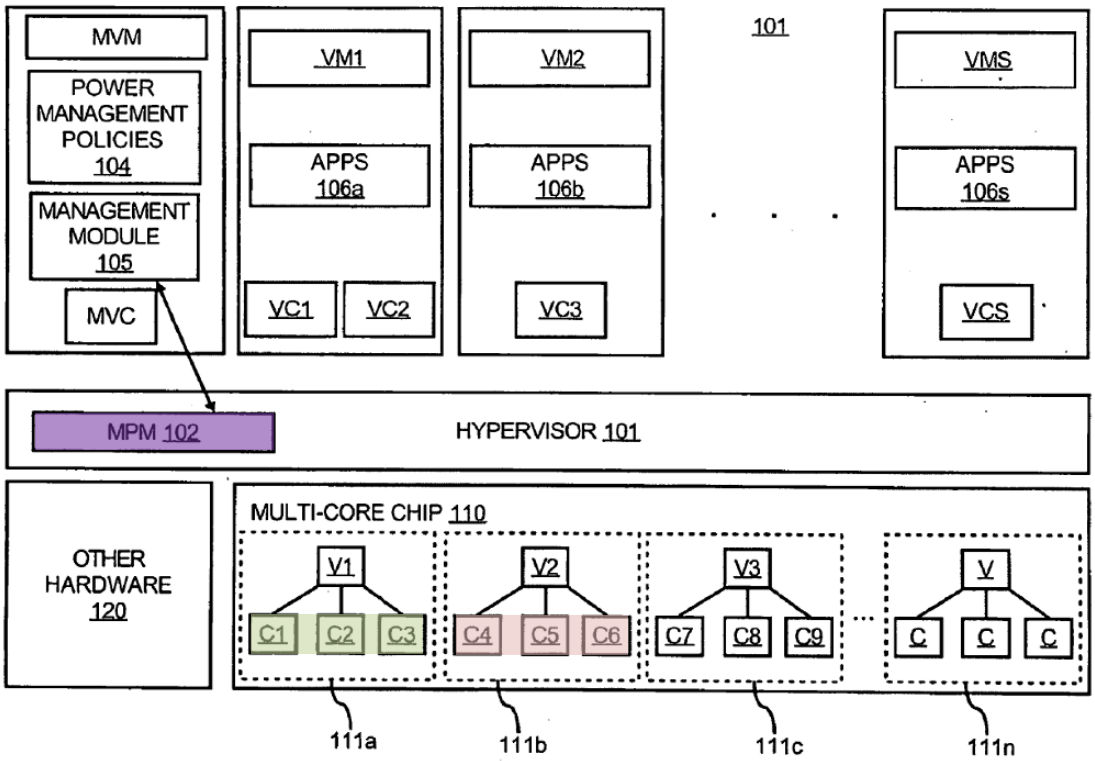


FIG. 1

Id., FIG. 1 (annotated). As shown in the figure, "cluster 111a includes voltage source V1 and cores C1-C3 [green]" and "cluster 111b includes voltage source V2 and cores C4-C6 [red]." *Id.*, [0014]. The "multi-core power module (MPM) [purple]" "maintains all the cores in a cluster at the same frequency" and "[e]ach cluster may have a different cluster frequency." *Id.*, [0018]-[0019]. By using the same supply voltage and frequency for each cluster, "maximum power savings" can be achieved (i.e., some clusters can be run at higher frequencies and voltages, while others can be dynamically set to lower frequencies and voltages). *Id.*, [0021].

3. Motivation to Combine

71. A POSITA would be motivated to apply Talwar's DVFS "clustering" of processor cores to White's DVFS implementation in order to achieve the goal common to White and Talwar—DVFS in a multicore processor below the chip level—while meeting practical chip space constraints. Specifically, a POSITA would be motivated to combine White with a reference that clearly discloses the use of more than two cores, like Talwar, because (1) White contemplates the use of "any number of cores," *see* EX1006, 6:50-60, and (2) when designing a chip with "any number of cores," such as the quadcore processors already commercially available before 2010, a POSITA would be motivated to add Talwar's DVFS "clusters," or groupings of processor cores to White's DVFS implementation to save valuable chip space and achieve improved power performance. *See also* EX1023, 2/8.

72. First, a POSITA would be motivated to combine White and Talwar because they address similar design considerations and Talwar provides additional disclosure on the optimization of additional cores. White teaches that "one or more of the cores of a processor may be adjusted to operate at a lower frequency and/or voltage in order to save power in certain situations." EX1006, 2:26-28. Similarly, Talwar teaches that "[d]ynamic voltage frequency scaling (DVFS)" can be used to reduce CPU power consumption in multi-core processors. EX1007, [0002]. White recognized that cores in prior art multi-core processors generally "operate at a

common voltage level." EX1006, 1:14-26. White proposed a multi-core processor in which "each logic core . . . operate[s] at a particular voltage and frequency." *Id.*, 2:19-22. And Talwar recognized that "multiple clusters . . . for performing power management . . . is a stark contrast to conventional multi-core chips and conventional DVFS." EX1007, [0012]. In other words, while White disclosed an example of individually controlling each core, Talwar explicitly recognized benefits—in terms of both efficiency and chip area—of controlling cores in clusters. Additionally, Talwar explicitly discloses an embodiment that includes up to 12 cores organized into independently controllable clusters. *See id.*, FIG. 1. For these reasons, a POSITA would have naturally looked to Talwar to supplement White in a multi-core processor with more than two cores.

73. Second, a POSITA would have understood that White's in-depth exemplar design—if applied with a distinct voltage regulator for each individual core—would need to be adapted as the number of cores increases due to chip space constraints associated with adding an additional voltage source for each additional core. By 2010, it was well known in the field of multi-core processor design that chip space was an increasingly restrictive design constraint, driven primarily by increasingly complex designs. *See supra* Section VII.A. For example, in 2008, with respect to off-chip voltage regulators like the one disclosed in White, it was known that "due to cost and system board area constraints, it may not be practical to

implement multiple power domains using off-chip voltage regulators" because the switching voltage regulators "better suited for loads employing DVFS" are "bulky and consume large area." EX1029, 3-4. Further, independently-regulated voltages for each core would require separate "bulky" regulator circuits for each independently-regulated core. *See id.*, 3-4. A POSITA therefore would have been motivated to find a solution that would allow for more cores without requiring a separate voltage regulator for each core. *See also* EX1019, 52-53 (surveying 2000s-era research into varying granularities of DVFS from "per-chip" to "per-core").

74. Additionally, by 2010, processors with sixteen or more cores were being contemplated, studied, and designed. *See e.g.,* EX1020, 1:16-21, 1:52-53, EX1021, 2:54-55, FIG. 1. Thus, a POSITA would have understood White's disclosure of a processor with "any number of cores" to realistically apply to multi-core processors with at least as many as 16 cores, each with a separate voltage and clock source. This number of voltage and frequency regulators is not an optimal design in most, if not all multi-core processors. As explained in Section VII.A, there is a design trade off between DVFS flexibility and chip space that must be considered.

75. Talwar recognized that multi-core processors often "don't have the same number of individual voltage sources" as processors. EX1007, [0004]. Working within this extra design consideration (i.e., fewer voltage sources than

processors), Talwar's clusters allow for cluster-by-cluster DVFS, while still enabling "maximum power savings" by ensuring all cores associated with a particular voltage source "run at the same frequency." *Id.*, [0021]. Talwar's "clustering" also fits White's explicitly teaching that its design is not limited to per-core voltage and clock signals, but may be applied more generally "to any integrated circuit or component that includes multiple areas or sections of circuitry configured to operate at independent operating voltages and/or frequencies" EX1006, 8:44-49, 3:37-52, 6:63-67.

76. Accordingly, for the reasons described above, a POSITA designing a multi-core processor for a power-sensitive and area-sensitive system based on White would look to Talwar's method of clustering cores to supplement White's design at higher number of cores per chip, in which chip space is a more scarce resource and, therefore, a more important design constraint. *See supra* Section VII.A.

4. Modification of White in View of Talwar

77. As discussed above, a POSITA would have been motivated to add Talwar's clustering of logic cores to White's dynamic voltage and frequency power management system. The combination would implement a multi-core processor following the block diagram of White, but would replace the individual logic cores with groups, or clusters of cores. Thus, the system would perform the same power management, or DVFS methods as White, but would do so on a cluster-by-cluster

basis. White's FIG. 2 embodiment, including logic cores 120A-120B can be modified to include four logic cores 120A-120D. Then, applying Talwar's clustering disclosure, [green] logic cores 120A and 120C constitute a first cluster and [red] logic cores 102B and 120D constitute a second cluster as shown below:

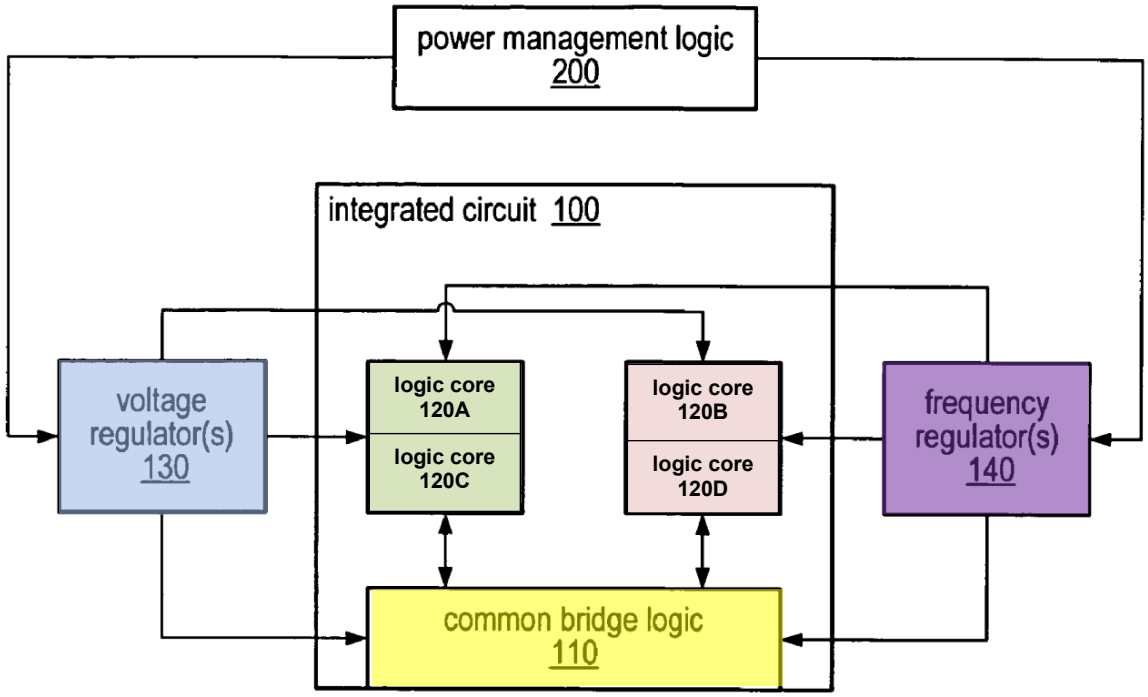


FIG. 2A

See EX1006, FIG. 2A (modified); EX1007, FIG. 1. In the combination, voltage regulator 130 provides one voltage level to cores 120A and 120C (added) and an independent voltage level to cores 120B and 120D (added). Frequency regulator 140 provides one external clock signal to cores 120A and 120C and an independent external clock signal to cores 120B and 120D.

78. Because each core in a cluster operates at the same voltage, there is a corresponding ideal frequency for the entire cluster due to the inherent relationship between a CPU's supply voltage and achievable clock frequency. *See* EX1007, [0003], [0019], [0021]; *supra* Section VII.C. Talwar teaches that "reducing the frequency of one core in a cluster provides insignificant power savings unless all the cores in the cluster have their frequency reduced. EX1007, [0012]. Therefore, a POSITA would have understood that the internal clock signal for each core, as disclosed by White, should be shared among the cluster of cores operating at the same voltage level to avoid the need for additional, redundant clock circuitry. *See also* EX1006, 9:62-65; EX1007, [0012]. Thus, in the combination depicted above, frequency regulator 140 provides one clock signal to a local PLL shared by cores 120A and 120C and an independent clock signal to a separate, local PLL shared by cores 120B and 120D. *See* EX1006, 9:62-65. Each PLL generates the internal clock signal, as taught by White, for a cluster of cores that all operate at the same frequency, as taught by Talwar. *See* EX1006, 9:62-65; EX1007, [0021]. In short, a POSITA would have understood that each cluster has one, shared PLL to derive the shared clock for the cores in the cluster.

79. In the combination, White's common bridge logic need only be expanded to add a connection to each additional logic core. White discloses both that the common bridge logic "may include multiple ports or interfaces, each of

which may communicate with a different one of multiple logic cores," and that there may be "any number of logic cores" in the multi-core processor. EX1006, 4:11-30, 6:56-60. Further, Talwar's clustering does not increase the complexity of the common bridge logic by requiring it to interface with cores operating at additional voltages and frequencies because the cores within each cluster share a common voltage and frequency (i.e., the combination has two voltage and clock domains, the same number as in White's disclosed embodiments).

80. White's power management logic remains the same in the combination. White provides multiple teachings that its design is compatible with power management across groups of processor components, such as clusters of cores. *See e.g.*, EX1006, 3:37-52 (when adjusting the voltage and frequency of one logic core, "the operating voltage and/or frequency for others of the logic cores may also be reduced, either at the same time or subsequently."), 6:63-67("an integrated circuit may include two or more logic cores that each may be configured to operate at frequencies and voltages independent of the frequencies or voltages that other logic cores . . . may be operating."), 8:44-49 ("the systems, methods, and techniques described herein for independently controlling operating voltages and/or frequencies may be applied to any integrated circuit or component that includes multiple areas or sections of circuitry configured to operate at independent operating voltages and/or frequencies.").

5. Element-by-Element Analysis

81. The analysis of each element below incorporates the above descriptions of each reference, the motivation to combine the references, and the resulting modification.

a) Claim 1

[1.pre] A multi-core processor, comprising:

82. In my opinion, White/Talwar renders obvious Limitation [1.pre]. As discussed in Section X.A.1, White discloses methods for independently adjusting supply voltages and clock frequencies in "a multi-core integrated circuit," or "[a] processor or other integrated circuit [that] may include multiple logic cores." EX1006, 2:15-22, 4:31-35, 6:41-46, 8:41-49, 14:39-54, 15:42-64.

[1.1] a first set of processor cores of the multi-core processor,

83. In my opinion, White/Talwar renders obvious Limitation [1.1]. As discussed in Section X.A.4 and shown in modified Figure 2A, below, the proposed White/Talwar combination implements Talwar's teaching that "[t]he cores and voltage sources are divided into clusters, whereby multiple cores in a cluster receive power from a single voltage source." EX1007, [0011], Claims 1, 25.

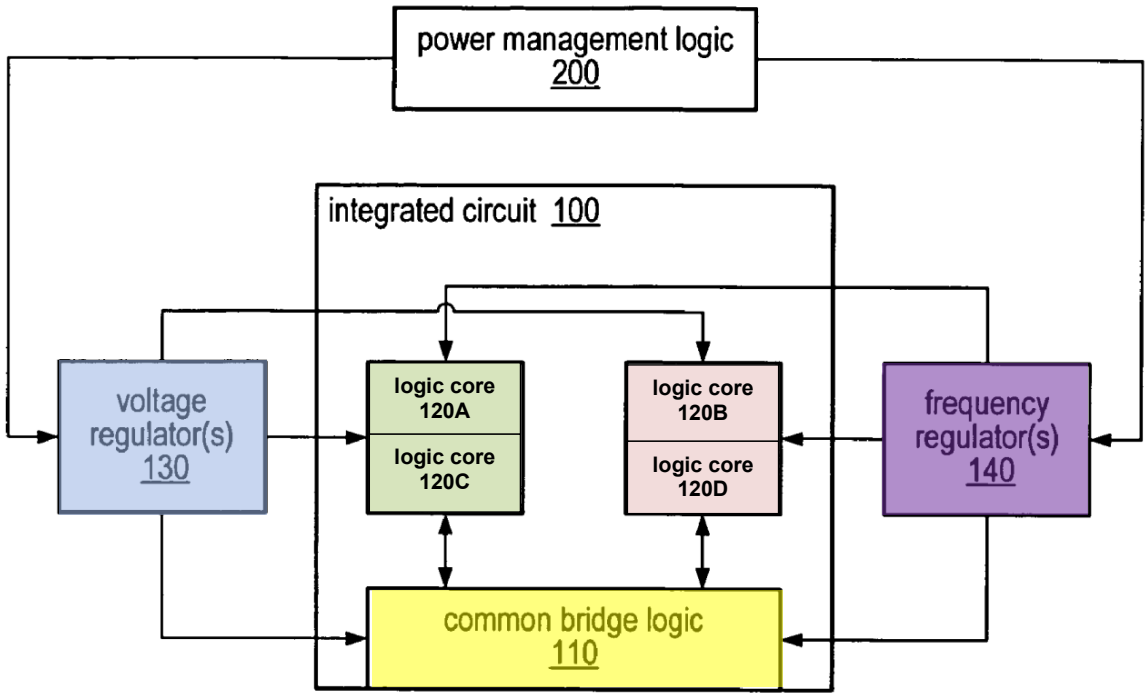


FIG. 2A

84. As shown in modified Figure 2A of White above, cores 120A and 120C of integrated circuit 100 (i.e., the multi-core processor) are grouped into a first cluster that is a first set of processor cores of the multi-core processor and cores 120B and 120D are grouped into a second cluster that is a second set of processor cores of the multi-core processor. *See* EX1006, FIG. 2A (modified).

[1.1.1] wherein each processor core from the first set of processor cores is configured to dynamically receive a first supply voltage and a first output clock signal of a first phase lock loop (PLL) having a first clock signal as input;

85. In my opinion, White/Talwar renders obvious Limitation [1.1.1]. As discussed in Section X.A.1, White teaches that independent supply voltages and

clock signals may be provided to separate processor cores in a multi-core processor. EX1006, 10:11-11:23. White also teaches that the supply voltages and clock signal frequencies should be "adjusted" as needed, for example to support "power management" or "temperature management" requirements. EX1006, 2:15-44. White further teaches that internal clocks for each clock domain may be generated "using a phase-locked loop (PLL) locked to an externally provided clock signal, such as may be provided by frequency regulator 140," which provides "multiple, independent clock signals" to the system. EX1006, 7:40-45, 9:62-65.

86. White discloses that the independent clock signals sent from the frequency regulator to the PLLs at the cores may be generated according to both the First, Second, and Third Configurations discussed above in Section X.A.1. As discussed, those Configurations are:

87. **First Configuration**: Frequency regulator 140 includes "multiple clock sources" in the form of multiple crystal "oscillators" "in order to supply multiple, independent clock signals of different frequencies" (EX1006, 7:40-45), each feeding "a phase-locked loop (PLL)" at one of the clusters.

88. **Second Configuration**: Frequency regulator 140 includes "multiple clock sources" in the form of multiple PLL's, each fed by a corresponding crystal "oscillator" "in order to supply multiple, independent clock signals of different frequencies" (EX1006, 7:40-45), each feeding "a phase-locked loop (PLL)" at one

of the clusters, or directly feeding the cores themselves (*id.*, 9:62-65). EX1008, FIG. 1, Section VII.B.

89. **Third Configuration**: Frequency regulator 140 includes "clock trees or other circuitry configured to split or divide a clock signal into multiple individual signals" (EX1006, 7:40-43), each feeding "a phase-locked loop (PLL)" at one of the clusters, or directly feeding the cores themselves (*id.*, 9:62-65).

90. With respect to the First and Second Configurations, White discloses the use of multiple clock sources within the frequency regulator to generate the "multiple, independent clock signals," where "clock source" is defined to include "an oscillator or phase locked loop (PLL)." EX1006, 7:23-45. Therefore, White discloses multiple oscillators or PLLs within the frequency regulator to generate the "multiple, independent clock signals" provided to the PLLs at each core that generate each core's internal clock. *See id.* As explained above, this means that the "multiple, independent clock signals" originate from two different external crystal oscillators. With respect to the Third Configuration, White also discloses that "clock trees or other circuitry" may be used to "split or divide a clock signal into multiple individual signals." EX1006, 7:40-43. In the Third Configuration, the signals provided to the PLLs at each core are derived from "clock trees or other circuitry configured to split or divide a clock signal into multiple individual signals." *Id.*

Thus, while the clock signals in the Third Configuration share a common source that they both depend on, they are still different signals.

91. Each of the above Configurations satisfy Limitation [1.1.1]. The First and the Second Configurations satisfy this Limitation because they disclose a first input clock signal to a PLL for the first cluster of cores that is independent of a second input clock signal to a PLL for the second cluster of cores, in the sense that there is no dependency between the two input clock signals. The Third Configuration satisfies this Limitation to the extent that "independent" is understood simply to mean different clock signals that can depend from the same reference oscillator.

92. Additionally, as explained in Section X.A.4 and shown in modified Figure 2A, the White/Talwar combination implements White's teachings of dynamically providing supply voltages and clock frequencies with Talwar's clustering of cores, which dynamically receive the same. The proposed combination reduces the amount of on-chip circuitry required to support independent voltage regulation and clock generation for each core, while preserving the power management flexibility provided by White's multiple voltage and clock domains.

93. Thus, the White/Talwar combination discloses this Limitation as follows: each processor core in each cluster "dynamically receive[s]" a common supply voltage ("first supply voltage") and common clock signal ("first output clock

signal"), the common clock signal being derived from a separate PLL for each cluster of cores ("of a first phase lock loop (PLL)"), the PLL being phase locked to the first of the multiple, independent clock signals provided by the frequency regulator in accordance with any of the Configurations discussed above ("having a first clock signal as input").

[1.2] a second set of processor cores of the multi-core processor,

94. In my opinion, White/Talwar renders obvious Limitation [1.2]. As discussed in Limitation [1.1], the White/Talwar combination implements Talwar's teaching that "[t]he cores and voltage sources are divided into clusters, whereby multiple cores in a cluster receive power from a single voltage source." EX1007, [0011], Claims 1, 25. As shown in modified Figure 2A, cores 120B and 120D of integrated circuit 100 (i.e., the multi-core processor) are grouped into a second cluster that is "a second set of processor cores of the multi-core processor."

[1.2.1] wherein each processor core from the second set of processor cores is configured to dynamically receive a second supply voltage and a second output clock signal of a second PLL having a second clock signal as input,

95. In my opinion, White/Talwar renders obvious Limitation [1.2.1]. As discussed in Limitation [1.1.1], the White/Talwar combination includes multiple sets, or clusters of cores, wherein each cluster shares a supply voltage and clock signal that is independent from the supply voltages and clock signals of other clusters. *See* EX1006, 10:11-11:23; EX1007, [0011]-[0012]. Therefore, for the

same reasons described in Limitation [1.1.1], the second cluster of cores "dynamically receive[s]" a second common supply voltage ("second supply voltage") and second common clock signal ("second output clock signal"), the second common clock signal being derived from a separate PLL for the second cluster of cores ("of a second PLL") the second PLL being phase locked to the second of the multiple, independent clock signals provided by the frequency regulator in accordance with any of the Configurations discussed in Limitation [1.1.1] and Section X.A.1 ("having a second clock signal as input").

[1.2.2] wherein the first supply voltage is independent from the second supply voltage,

96. In my opinion, White/Talwar renders obvious Limitation [1.2.2]. As explained in Sections X.A.1 and X.A.2, both White and Talwar teach that the supply voltages to individual cores (White) or clusters of cores (Talwar) can be independently controlled. EX1006, 10:11-11:23; EX1007, [0011]-[0012]. Therefore, in the White/Talwar combination, the first supply voltage to the first cluster is independent from the second supply voltage to the second cluster.

[1.2.3] and the first clock signal is independent from the second clock signal; and

97. In my opinion, White/Talwar renders obvious Limitation [1.2.3]. A POSITA would have understood that the Configurations disclosed by White disclose multiple, independent clock signals as inputs to the PLLs at each core. *See*

Limitation [1.1.1]. A POSITA would have understood this based both on White's explicit teachings that the frequency regulator "may include multiple clock sources in order to supply multiple, independent clock signals of different frequencies" and the well-known prior art method of using multiple, independent reference oscillators to generate independent clocks for separate clock domains. EX1006, 7:23-45; *see e.g.,* EX1008, 1-2, Figure 1, *supra* Section VII.B.

[1.3] an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores,

98. In my opinion, White/Talwar renders obvious Limitation [1.3]. As discussed in Section X.A.1, White's common bridge logic block is "an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores." As shown in modified Figure 2A and explained in White, the common bridge logic is "configured to communicate with the logic cores as well as with other elements or peripherals . . . such as a memory subsystem." EX1006, 7:1-7.

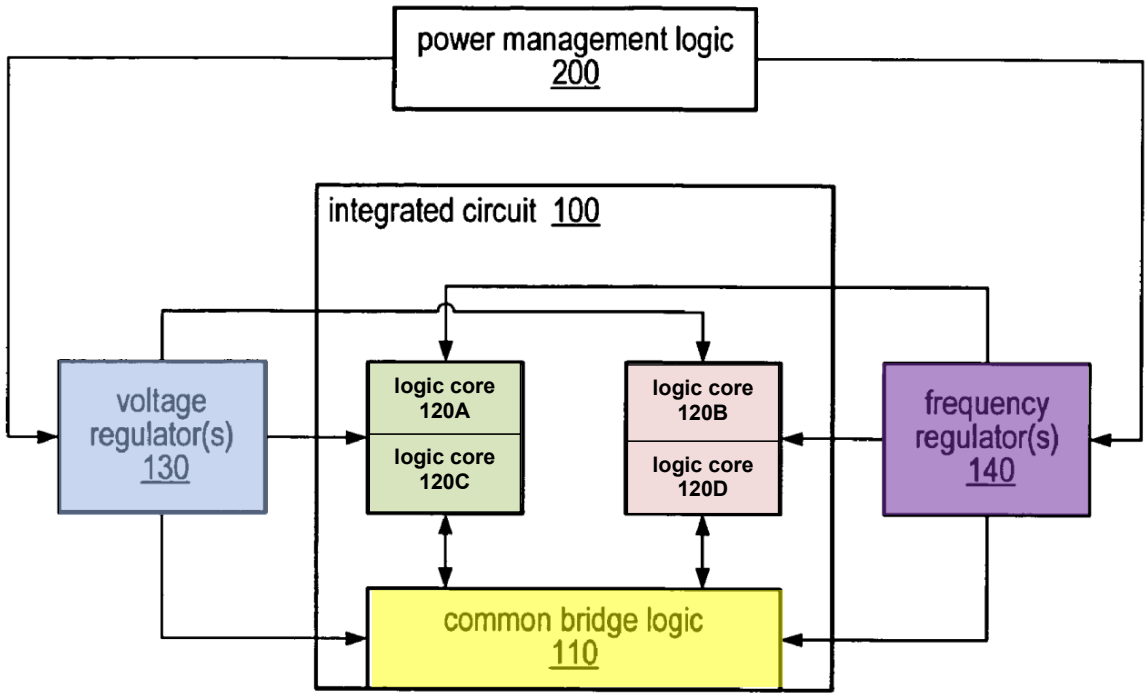


FIG. 2A

EX1006, FIG. 2A (modified). As shown in the figure, the common bridge logic facilitates communication with both the first and second sets of processor cores. Further, a POSITA would have understood that each processor core must be separately connected to the common bridge logic to allow each processor to interface with "other elements and peripherals" of the system, as each processor executes a separate thread or application. *See* EX1006, 7:1-7. As was known in the art, it is also preferred that each of logic cores 120A-D are separately connected to the common bridge logic to facilitate coherence related communication between the cores.

99. Based on White's explicit reference to a "memory subsystem" and the knowledge of a POSITA in 2010 that cores may read and write to a common memory space through shared memory and cache coherency (Section VII.D), a POSITA would understand that each of cores of the first and second sets of processor cores may read and write to common space in the memory subsystem, assisted by cache coherency, via the common bridge logic. Accordingly, the common bridge logic, memory subsystem, and/or cache coherency logic are the claimed interface block.

[1.3.1] wherein the interface block is configured to facilitate communication between the first set of processor cores and the second set of processor cores.

100. In my opinion, White/Talwar renders obvious Limitation [1.3.1]. In addition to the disclosure described with respect to Limitation [1.3], White teaches that the common bridge logic "may be configured to translate the communication, such as the signals on one or more control lines, between the different logic levels and/or between the different frequencies." EX1006, 8:13-40, 14:55-15:17.

101. As discussed for Limitation [1.3], a POSITA would understand that each of cores of the first and second sets of processor cores may read and write to common space in the memory subsystem, assisted by cache coherency, via the common bridge logic. Section VII.D. Shared memory (i.e., the memory subsystem enabling cores from the first and second set of processor blocks to write to a common memory), as well as the use of cache coherency, facilitate communication between the first and second set of processor cores. As was well known in the art, multi-core

processors can gain the performance benefits of parallel processing using shared memory and by executing operations on shared data in parallel. This sort of parallelism requires coordination between the cores with respect to the shared memory. *See* Section VII.D.

102. Further, White teaches that the common bridge logic "may be configured to translate the communication, such as the signals on one or more control lines, between the different logic levels and/or between the different frequencies." EX1006, 8:13-40, 14:55-15:17. Because it is the processor cores that are operating at different logic levels and different frequencies, a POSITA would have understood this teaching to mean that the common bridge logic translates signals either directly between processor cores, indirectly between processor cores via the use of a shared memory, or both, which also facilitates communication between the first and second set of processor cores.

b) Claim 2

[2.1] The multi-core processor of claim 1, wherein the interface block further comprises a first level shifter that is referenced to the second supply voltage and adapted to translate first logic levels associated with the first set of processor cores to second logic levels associated with the second set of processor cores for a first signal traveling from the first set of processor cores to the second set of processor cores.

103. In my opinion, White/Talwar renders obvious Limitation [2.1] for reasons discussed above in Limitation [1.3.1]. As discussed in Section VII.C, it was well known in the art that multi-core processors implementing DVFS required

mechanisms for translating logic levels between chip components operating at different voltages. A POSITA would have understood White's teaching that the common bridge logic "may be configured to translate the communication … between the different logic levels" to mean that the common bridge logic includes one or more level shifters that enable the bridge to translate between the logic levels used by the first set of processor cores and the second set of processor cores. *See* EX1006, 8:13-40, 14:55-15:17, FIG. 3. As discussed, level shifters are an essential feature of chip designs that use multiple voltage domains. *See* Section VII.C. Further, level shifter designs were well known in the art. *See e.g.,* EX1031, FIGs. 4-5, 5:45-57, EX1039, 1-2/4, Figures 2(a), 2(b), 3. Additionally, a POSITA would have understood that the level shifter that translates between the logic level of the first set of processor cores to the logic level of the second set of processor cores must be referenced to the second supply voltage to know at what logic level the second set of processor cores is operating. *See also* EX1039, 1/4 ("each block on a chip could be operated under either high V_{DD} (V_{DDH}) or low V_{DD} (V_{DDL})"), Figure 2(a) (depicting a "conventional level shifter" referenced to both V_{DDH} and V_{DDL} to translate signals between the two logic levels).

c) Claim 3

[3.1] The multi-core processor claim 1, wherein the interface block further comprises a second level shifter that is referenced to the first supply voltage and adapted to translate second logic levels associated with the second set of processor cores to first logic levels associated with the first set of processor cores for a second signal traveling from the second set of processor cores to the first set of processor cores.

104. In my opinion, White/Talwar renders obvious Limitation [3.1] for reasons discussed above in Limitation [2.1].

d) Claim 5

[5.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a periphery of the multi-core processor.

105. In my opinion, White/Talwar renders obvious Limitation [5.1]. White discloses that the "logic circuitry or firmware configured to control the operating voltage and frequency of logic cores" may "reside in the processor itself." EX1006, 5:17-24; *see also* 13:7-26 ("logic circuitry…internal…to integrated circuit"), 14:32-38 ("temperature control logic [] and power management logic [] may…be part of a single set of logic."), FIGs. 2A, 2B. Thus, the logic circuitry within the processor may include the "power management logic" that sends control signals to the processor cores by "writ[ing]to one or more control registers" within each processor core. EX1006, 10:7-9. Further, a POSITA would have understood that circuitry design involves layout decisions that balance performance, area, and thermal considerations, and that placement of control logic circuitry along the perimeter of

the multi-core processor or substantially central to processor cores is an obvious design choice. *See also* EX1033, FIG. 8, [0056], [0103]-[0107] (depicting chip "floor plan" with "a controller for signal transmission from/to storage control units SC" that are "located at the side nearer to the SC [storage control units]," which is substantially central to the processor units.). For example, the central placement of shared components, such as control logic, typically shortens the total length of wires between the cores and the control logic, shortening the delay between the cores and the control logic. On the other hand, a POSITA would also understand that it can be more efficient under some circumstances to place the control blocks, such as the power management controller, around the periphery of core clusters, rather than substantially central to the core clusters. That way the core clusters can be placed closed to each other and may have shorter inter-cluster communication wires.

e) Claim 8

[8.1] The multi-core processor of claim 1, wherein the first set of processor cores are located in a first region of the multi-core processor, and the second set of processor cores are located in a second region of the multi-core processor.

106. In my opinion, White/Talwar renders obvious Limitation [8.1]. As discussed in Section X.A.4, White/Talwar includes a first cluster of logic cores 120A and 120C (the first set of logic cores) that share a voltage signal, clock signal, and cluster-level clock control circuitry and a second cluster of logic cores 120B and 120D (the second set of logic cores) that share a different voltage signal, clock

signal, and cluster-level clock control circuitry. In addition, it would have been obvious to a POSITA to arrange logic cores 120A and 120C in a first row constituting a first region and logic cores 120B and 120D in a second row constituting a second non-overlapping region. First, arranging logic cores in rows was a well-known approach to physical layout of a multicore processor. *See e.g.,* EX1033, FIG. 8, EX1022, 2:8-10, FIGs. 1-2, EX1036, 4:53-57, FIG. 2, EX1011, FIG. 3. Cores and other components must be organized on the chip efficiently to meet any placement needs, and it was well known that rows, or an array of cores was a common organization touted for its simplicity.

107. In addition to simplicity, arranging the first cluster in one row and second cluster in a second row enables efficient signal routing to route the shared clock signal and voltage signal to each core in a cluster. It was well known that optimal chip layouts may sometimes involve minimizing the distance between shared components and the cores that share them. *See* EX1033, [0124], FIG. 8.

108. Similarly, placing the clusters in non-overlapping regions can enable a floor plan design that allows cluster-level clock control circuitry to be placed in between the non-overlapping regions so that control data can be sent to different clusters efficiently. Again, the placement of shared components near to the cores that share them was a well-known design that minimizes wire length and increases logic speed.

f) Claim 10

[10.1] The multi-core processor of claim 8, wherein the first region and the second region are non-overlapping regions of the multi-core processor.

109. In my opinion, White/Talwar renders obvious Limitation [10.1] for reasons discussed in Limitation [8.1].

g) Claim 11

[11.1] The multi-core processor of claim 8, wherein the first region corresponds to a first row of the multi-core processor, and wherein the second region corresponds to a second row of the multi-core processor.

110. In my opinion, White/Talwar renders obvious Limitation [11.1] for reasons discussed in Limitation [8.1].

h) Claim 14

[14.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a common region that is substantially central to the first set of processor cores and the second set of processor cores

111. In my opinion, White/Talwar renders obvious Limitation [14.1] for reasons discussed in Limitation [5.1].

i) Claim 21

[21.pre] A multi-core processor, comprising:

112. In my opinion, White/Talwar renders obvious Limitation [21.pre] for reasons discussed in Limitation [1.pre].

[21.1] a first set of processor cores of the multi-core processor,

113. In my opinion, White/Talwar renders obvious Limitation [21.1] for reasons discussed in Limitation [1.1].

[21.1.1] wherein each processor core from the first set of processor cores is configured to dynamically receive a first supply voltage from a power control block and a first output clock signal from a first phase lock loop (PLL) having a first clock signal as input in a clock control block;

114. In my opinion, White/Talwar renders obvious Limitation [21.1.1] for reasons discussed in Limitation [1.1.1]. As explained in Section X.A.4 and shown in modified Figure 2A, in White/Talwar each processor in the first / second cluster receives a first / second voltage signal (the first / second supply voltage) from voltage regulator 130. A POSITA would have understood White's "voltage regulator 130" to be a power control block.

115. In the First and Third Configurations of the White/Talwar combination, the internal clocks for each cluster are generated "using a phase-locked loop (PLL) locked to an externally provided clock signal." *See* EX1006, 9:62-65. A POSITA would have understood this passage to disclose that the region on the chip encompassing these local PLLs is a clock control block.

116. Alternatively, the Second Configuration of White's frequency regulator described in Section X.A.1 discloses PLLs in the frequency regulator itself. Thus, in the Second Configuration, a POSITA would also have understood the frequency regulator to be a clock control block.

[21.2] a second set of processor cores of the multi-core processor,

117. In my opinion, White/Talwar renders obvious Limitation [21.2] for reasons discussed in Limitation [1.2].

[21.2.1] wherein each processor core from the second set of processor cores is configured to dynamically receive a second supply voltage from the power control block and a second output clock signal from a second PLL having a second clock signal as input in the clock control block,

118. In my opinion, White/Talwar renders obvious Limitation [21.2.1] for reasons discussed in Limitations [1.2.1] and [21.1.1].

[21.2.2] wherein the first supply voltage is independent from the second supply voltage, and

119. In my opinion, White/Talwar renders obvious Limitation [21.2.2] for reasons discussed in Limitation [1.2.2].

[21.2.3] the first clock signal is independent from the second clock signal; and

120. In my opinion, White/Talwar renders obvious Limitation [21.2.3] for reasons discussed in Limitation [1.2.3].

[21.3] an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores,

121. In my opinion, White/Talwar renders obvious Limitation [21.3] for reasons discussed in Limitation [1.3].

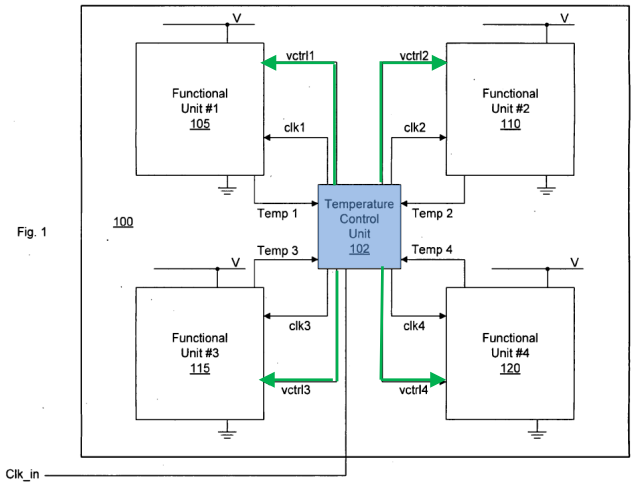
[21.3.1] wherein the interface block is configured to facilitate communication between the first set of processor cores and the second set of processor cores.

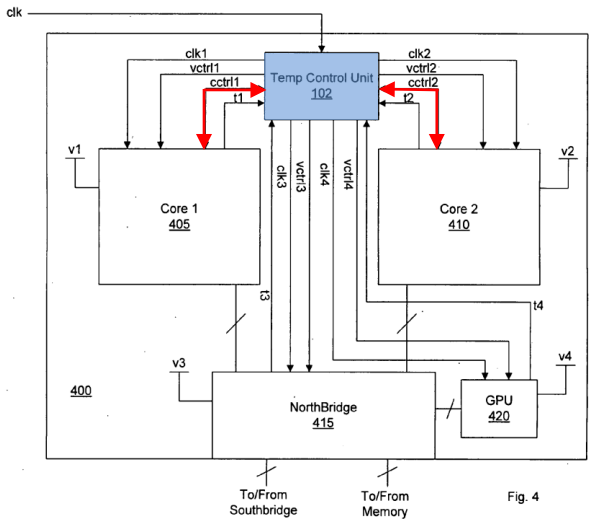
122. In my opinion, White/Talwar renders obvious Limitation [21.3.1] for reasons discussed in Limitation [1.3.1].

B. Ground 2: White, Talwar, and Cho Render Obvious Claims 5, 14

1. Cho

123. Cho discloses an "integrated circuit" that includes a plurality of "functional units…implemented on a die of the integrated circuit." EX1010, [0007]. The functional units may each represent a "processor core." *Id.*, [0038], Claim 17. In addition, a "temperature control unit (TCU)" is coupled to each of the functional units and "is configured to perform temperature control actions on each, independently of any temperature control actions performed on other ones of the functional units [e.g., processor cores]." *Id.*, [0038], [0047]-[0048]. Specifically, Cho explains that "in addition to the temperature signals, the clock signals, and the voltage control signals [green], processor cores [] may also be coupled to TCU [] by corresponding core control signals cctrl1 and ccrtl2 [red], respectively." *Id.*, [0042], Figs. 4. "These control signals may be bi-directional and/or may be transmitted via multiple signal lines" and through the use of these signals, the "TCU [] may also perform temperature control actions that involve re-allocating the processing load between processor cores [] in an effort to control their respective temperatures." *Id.*; *see also id.*, [0047]-[0048].





EX1010, Figures 1, 4 (annotated). In addition, as depicted above in Figures 1 and 4, the control block that provides these control signals (i.e, temperature control unit [blue]) may be located within the multi-core processor substantially central to the functional units (e.g., processor cores) as well as in the periphery of the multi-core processor. *Id.*

2. Motivation to Combine White/Talwar with Cho

124. A POSITA would have been motivated to further combine White/Talwar with Cho to achieve the goal of controlling processor core temperatures that is common to White and Cho. A POSITA would be motivated to include Cho's temperature control functionality because it achieves temperature control in a different way and is, thus, additive and adds extra flexibility. White discloses "[i]f temperature control logic 210 determines that the operating

temperature of integrated circuit 100 is too high, it may "automatically adjust the operating voltage and/or frequency of one or more logic cores." EX1006, 12:27-31.

125. Rather than adjusting operating voltage and/or frequency, Cho teaches that, when a processor core "is under a heavy processing workload and reports a temperature that exceeds a predetermined threshold," the "TCU [] may attempt to reduce the temperature of processor core 405 by re-allocating a portion of its workload…" EX1010, [0042]. Thus, these "core control signals" are used to communicate with the processor cores and "perform temperature control actions through workload allocation in addition to controlling clock frequency and temperature." *Id.* This approach, re-allocating a portion a core's workload, supplements the teachings of White. A POSITA would have been motivated to add this additional flexibility to White.

126. Additionally, these stated benefits of Cho align with White's temperature management goals. White recognized that processor cores within an integrated circuit "may be adjusted to operate at a lower frequency and/or voltage in order to lower the operating temperature of the processor." *See e.g.,* EX1006, Abstract, 2:22-26, *see also supra* Section VII.A. A POSITA would naturally look to combine the White/Talwar combination with a reference like Cho to gain the added flexibility of adjusting workloads to lower core temperatures as taught by Cho. EX1010, [0047]. By incorporating the ability to re-allocate workloads through

direct control signaling to the cores in the manner taught by Cho, the processor "may optimize both performance as well as thermal control." EX1010, [0047]. Therefore, a POSITA would be motivated to combine White, Talwar, and Cho to gain the predictable benefits disclosed in Cho.

3. Modification of the White/Talwar Combination in View of Cho

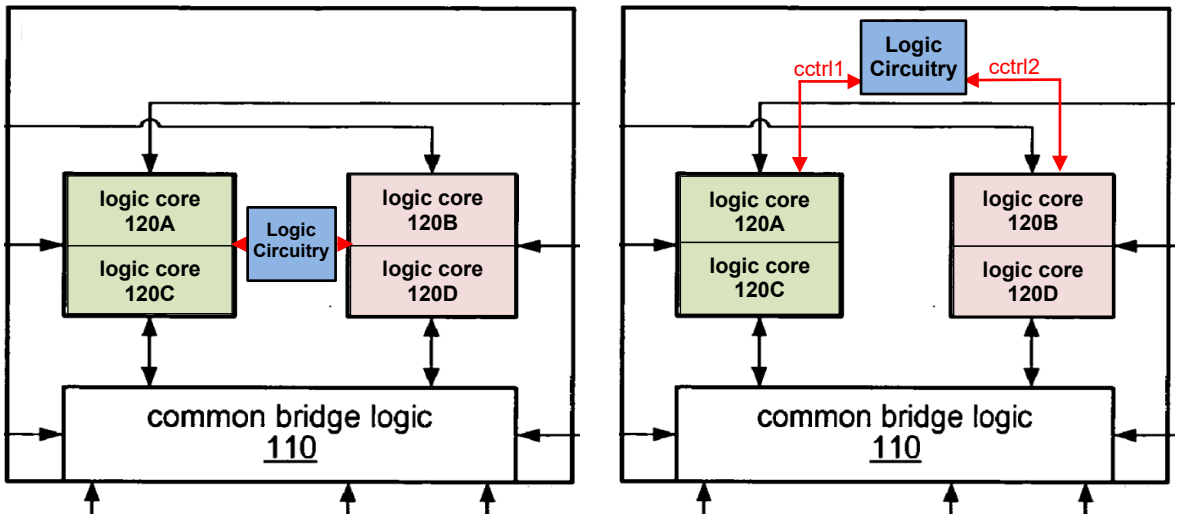
127. The functionality of Cho's TCU 102 may be incorporated into White's existing control logic circuitry. White explains that the logic circuitry (i.e., the "temperature control logic 210" and "power management logic 200") "configured to control the operating voltage and frequency of logic cores," "may reside in the processor itself" (EX1006, 5:17-20) and "may both be part of a single set of logic, implemented in hardware, firmware, software, or a combination." *Id.*, 14:32-38.

128. Cho provides additional guidance on the location of the control circuitry beyond White's disclosure of "in the processor" (i.e., integrated circuit 100), illustrating the control circuitry as alternatively substantially central to the functional units (e.g., processor cores) (EX1010, FIG. 1) or in the periphery of the multi-core processor *(id.*, FIG. 4). A POSITA would have understood both of these locations to represent acceptable known design choices. *See* EX1033, FIG. 8, [0056], [0103]-[0107], *see also* EX1036, FIG. 5, 6:39-47.

129. Finally, Cho's "core control signals cctrl1 and ccrtl2" that couple the "TCU" to "processor cores" follow the existing control path signals White describes

to enable "power management logic" to "write to one or more control registers" in the processing cores. EX1006, 10:4-8.

130. Accordingly, as depicted below in the modified portions of Figure 2B, the "control signals cctrl1 and ccrtl2" [red] as taught by Cho could be integrated within the power/temperature management set of logic (i.e., logic circuitry [blue]) in the White/Talwar combination and may be located within the multi-core processor as taught by both White and Cho. *See* EX1010, Fig. 2B (modified below).



Substantially Central Periphery

EX1006, FIG. 2B (modified). A POSITA would have understood that this single set of logic circuitry [blue] is a control block within the processor that could provide "control signals cctrl1 and ccrtl2," to registers within the processor cores instructing them to re-allocate workloads. The specific implementation of the logic circuitry within White/Talwar combination would not change the operation of White's power

management and temperature control within the multi-core processor. *See* EX1006, Abstract, 5:17-20, 13:7-26, 14:32-38, 16:58-63.

4. Element-by-Element Analysis

a) Claim 5

[5.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a periphery of the multi-core processor.

131. In my opinion, White/Talwar/Cho renders obvious Limitation [5.1]. White discloses that the "logic circuitry or firmware configured to control the operating voltage and frequency of logic cores" may "reside in the processor itself." EX1006, 4:49-5:17-24, 13:7-26. Further, as explained in Section X.B.3, the logic circuitry [blue] provides control signals to the processor cores and is located along the perimeter of the multi-core processor as shown in Fig. 4. *See also* EX1010, Fig. 4, [0038]-[0040], [0042], [0047]-[0048].

b) Claim 14

[14.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a common region that is substantially central to the first set of processor cores and the second set of processor cores

132. In my opinion, White/Talwar/Cho renders obvious Limitation [14.1] for reasons discussed in Limitation [5.1]. Further, as explained in Section X.B.3, in addition to locating logic circuitry [blue] along the periphery of a multi-core

processor, a POSITA would have understood that an obvious design choice would be placement of logic circuitry that provides control signals from a common region that is substantially central to the functional units (e.g., processor cores). *See* EX1010, [0018]-[0019], [0021]-[0022].

C. Ground 3: White, Talwar, and Vajda Render Obvious Claims 8-9

1. Vajda

133. Vajda discloses "a multi-core chip 230" that includes a number of "processor cores 231 that can execute at higher frequencies" and "other processor cores 232." EX1011, [0062]. Vajda provides guidance for "the geometry" (i.e., layout of the individual cores) "of the multi-core chip 230." *Id.* Specifically, Vajda proposes that "the processor cores 231 that can execute at higher frequencies may be spread out as evenly as possible and surrounded by other processor cores 232, running at lower frequencies" because such a layout "may be an advantage primarily in order to avoid hot spots and over-heating of the multi-core chip 230." *Id.*

2. Motivation to Combine White/Talwar with Vajda

134. A POSITA would have been motivated to apply Vajda's core layout guidance in defining the physical layout of cores in White/Talwar to obtain Vajda's stated benefit of a layout that may "avoid hot spots and over-heating." *See* EX1011, [0062]. As discussed in Section X.A.4, White/Talwar includes a first cluster of logic cores 120A and 120C and a second cluster of logic cores 120B and 120D, each having independent voltage and clock signals that enable operating one cluster at a

different higher frequency—producing more heat—than the other. As explained in Section VII.A, it was well known that temperature management is a key design consideration for multi-core processors. *See e.g.,* EX1006, 1:27-52, EX1011, [0063], EX1021, 1:32-40. Thus, the problem Vajda seeks to address where a layout includes "processor cores 231 that can execute at higher frequencies" and "other processor cores 232, running at lower frequencies" was a common design consideration and Vada's solution would be applicable to White/Talwar, such that a POSITA would be motivated to apply Vajda's layout guidance to "avoid hot spots and over-heating." *See* EX1011, [0062].

3. Modification of the White/Talwar Combination in View of Vajda

135. As discussed in Section X.A.4, White/Talwar includes a first cluster of logic cores 120A and 120C and a second cluster of logic cores 120B and 120D, each having independent voltage and clock signals that enable operating one cluster at a different higher frequency than the other. In accordance with Vajda's disclosure that "the processor cores 231 that can execute at higher frequencies may be spread out as evenly as possible and surrounded by other processor cores 232, running at lower frequencies" (EX1011, [0062]), a POSITA would be motivated to arrange the cores of the first cluster and the second cluster in a checkerboard pattern (e.g., a first row of logic core 120A followed by 120B and a second row of logic core 120D followed

by 120C) in order to "spread out" cores of one cluster from each other and avoid any hot spots created by grouping of cores of one cluster operating at a higher frequency.

4. Element-by-Element Analysis

a) Claim 8

[8.1] The multi-core processor of claim 1, wherein the first set of processor cores are located in a first region of the multi-core processor, and the second set of processor cores are located in a second region of the multi-core processor.

136. In my opinion, White/Talwar/Vajda renders obvious Limitation [8.1]. As discussed in Section X.C.3, White/Talwar/Vajda includes logic cores 120A and 120C (the first set of processor cores) and logic cores 120B and 120D (the second set of processor cores) arranged in a checkerboard pattern (e.g., a first row of logic core 120A followed by 120B and a second row of logic core 120D followed by 120C). The region containing the first set of processor cores is the first region and the region containing the second set of processor cores is the second region. Such a checkerboard pattern spreads out the heat generated by any one cluster of cores that is operating at a high frequency and voltage. As a result, the cluster may be able to continue operating at a higher frequency and voltage for a longer period of time before it must be limited by the temperature management logic.

b) Claim 9

[9.1] The multi-core processor of claim 8, wherein the first region and the second region are overlapping regions of the multi-core processor.

137. In my opinion, White/Talwar/Vajda renders obvious Limitation [9.1] for reasons discussed in Limitation [8.1]. The first and second regions are an overlapping region because the checkerboard pattern of cores results in overlapping regions containing each cluster. A POSITA would understand that such a design would have the benefit of spreading the heat across the entire chip.

D. Ground 4: Elgebaly and Ozer Render Obvious Claims 1-3, 5, 8, 10-11, 14, and 21

1. Elgebaly

138. Elgebaly discloses "[t]echniques for adaptively scaling supply voltage" and generating "[a] target clock frequency for the processing core" "based on computational requirements for the core." EX1004, Abstract, [0009], FIGs. 1-2. While Elgebaly's Figures 1 and 2 each show an ASIC (120) with two processor cores, Elgebaly expressly states that ASIC (120) may include "any number of processor cores" and "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." *Id.*, [0030], [0033]. Based on this disclosure, a POSITA would have understood Elgebaly to extend at least to an ASIC 120 with four cores, as shown in Modified Figure 1 below with an additional [green] Processor Core 3 and an additional [red] Processor Core 4.

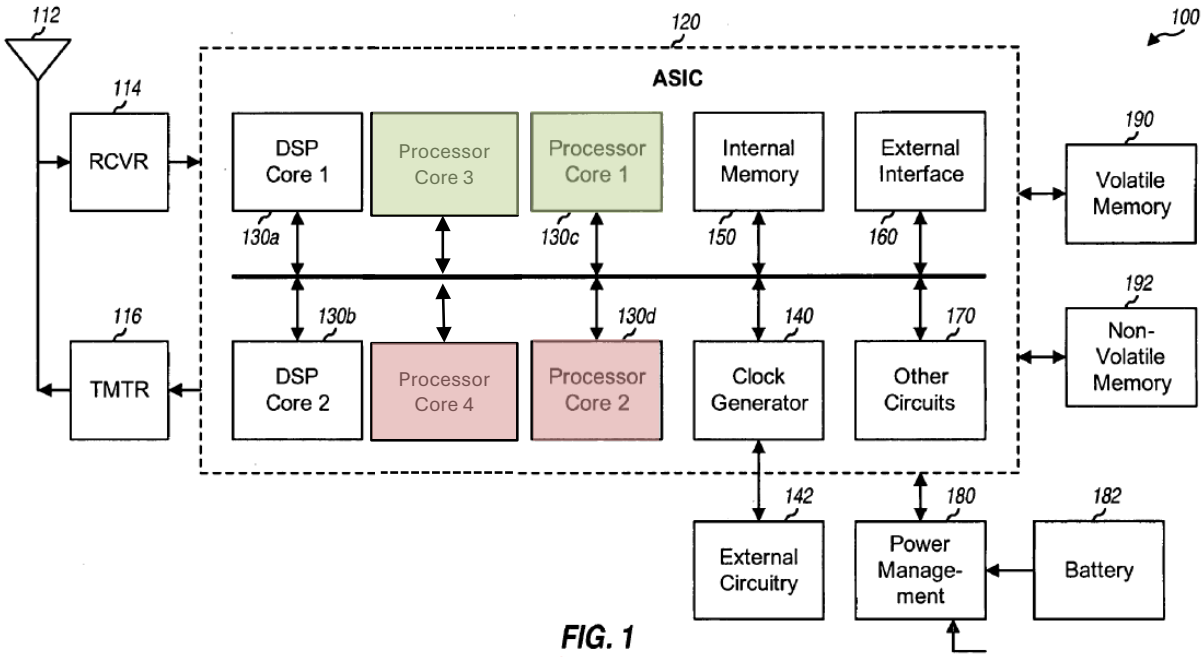


FIG. 1
(MODIFIED)

See EX1004, FIG. 1, [0033]. Consistent with Elgebaly's teaching and as described in Section VII.B, multi-core processors containing four cores were also well known in the art and available commercially well before 2010. *See e.g.,* EX1023, 2/8, EX1004, [0030], [0033].

a) Independent clock signals

139. Elgebaly's Background explains that "[a]n effective method for reducing power consumption when operating in an active mode is to scale or adjust the supply voltage." *Id.*, [0006]. As was known in the art, this "may be achieved by

identifying a critical signal path in an integrated circuit (IC), e.g., the signal path with the longest delay, and adjusting the supply voltage such that the critical signal path meets timing requirements." *Id.*, [0007]. Elgebaly also recognized that "the critical signal path may vary from die to die based on IC process and temperature variations," and also based on the current supply voltage and target clock frequency. *Id.*, [0007], [0009]. Elgebaly discloses the use of "adaptive voltage scaling (AVS)" units, where "one AVS unit 230 is provided for each processing core 130 and is used to adaptively scale the supply voltage for that core." *Id.*, [0031].

140. Figure 2 of Elgebaly shows chip using (blue) AVS units to independently set the supply voltages to each (green, red) processor core:

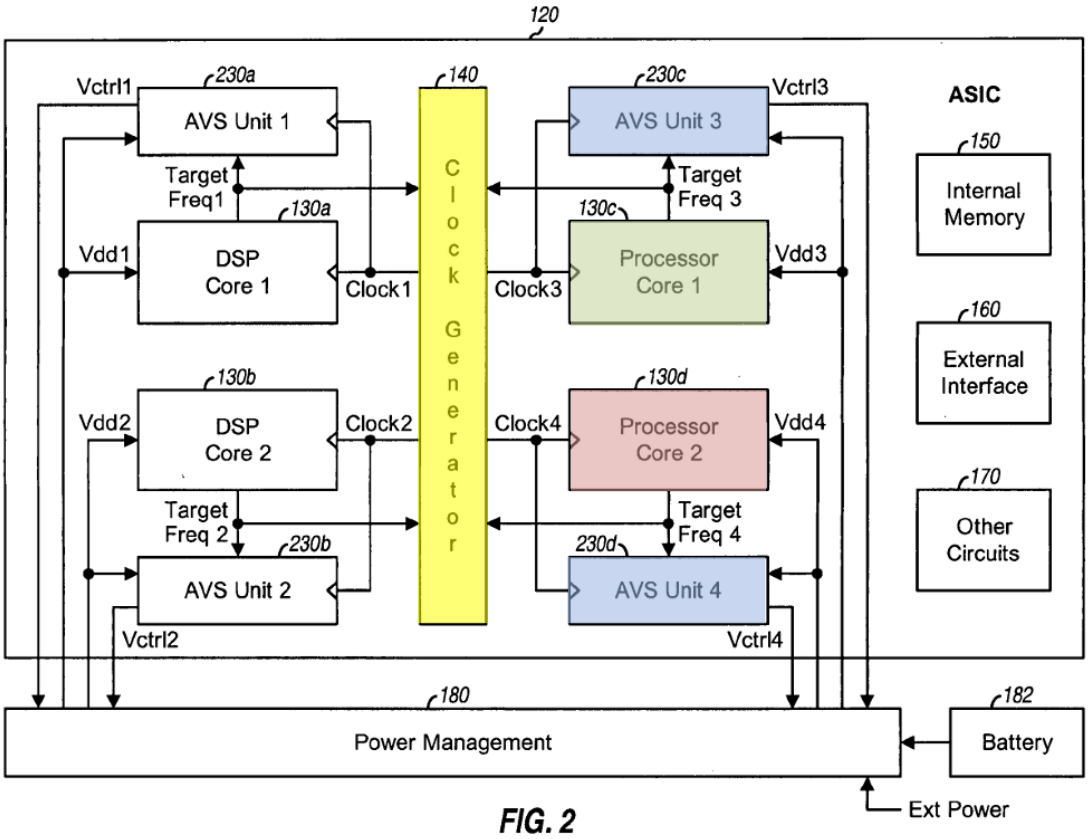


FIG. 2

Id., FIG. 2 (annotated). As shown in the figure above, "[yellow] clock generator 140 generates the clock at the target frequency and provides this clock to the [processor] core." *Id.*, [0032]. The "target frequency may be selected based on the processing load and computational requirements for the core." *Id.*

141. To generate clock signals, the clock generator 140 uses "one or more phase locked loops (PLLs) that control the oscillators" to generate corresponding clock signals. *Id.*, [0028]. The clock generator also "may couple to external circuitry, which may include ***crystals***." *Id.* For the embodiment of Figure 2, in which the clock for each core "may be set independently," a POSITA would have understood Elgebaly's disclosure of one or more PLLs that control the oscillators and external circuitry including plural "crystals" to include known clock distribution channels; specifically, two PLLs each receiving a distinct clock signal from a different external oscillator crystal and, in turn, outputting one of the desired clock signals.

142. Such a configuration was also an obvious design choice that would have been considered by a POSITA. Multiple external oscillators driving multiple PLLs is a simple way to implement multiple clock signals that may be set independently and thereby achieve the flexibility and potential power savings benefits disclosed by Elgebaly. Such a configuration would have been desirable to maximize independence and flexibility between the clock signals sent to AVS Unit 1 and AVS

Unit 3 to maximize the benefits Elgebaly espouses from the ability to independently control clock signals and voltages. Indeed, such a configuration using separate oscillators and PLLs was known, as explained in Section VII.C above. *See* EX1008, Figure 1.

143. Once the processor-specific clock signal is generated, "[t]he supply voltage for each processing core 130 is set by the associated AVS unit 230 such that the core can operate at the target clock frequency." EX1004, [0032]. Thus, "the clock for each processing core 130 may be set independently by clock generator 140, and the supply voltage for each processing core 130 may be set independently by power management unit 180." *Id.*, [0033]. Elgebaly may be extended to "multiple processing cores" in which the clock signal and supply voltage are "jointly set for these cores." *Id.* Further, "ASIC 120 may include . . . any number of processor cores." *Id.*, [0030].

b) Sets of cores

144. As noted above, Elgebaly explains that "ASIC 120 may include … any number of processor cores." *Id.*, [0030]. As such, Elgebaly at least renders obvious an ASIC 120 with four cores, if not 8 or 16 cores, as were already being designed by 2010. Egebaly also discloses that "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." EX1004, [0033]. Elgebaly therefore both discloses and renders obvious a chip with

four processor cores in which a first two of the processor cores share a first common clock and supply voltage, and in which the other two of the processor cores share a second common clock and supply voltage, as shown in Modified Figure 2, inset below.

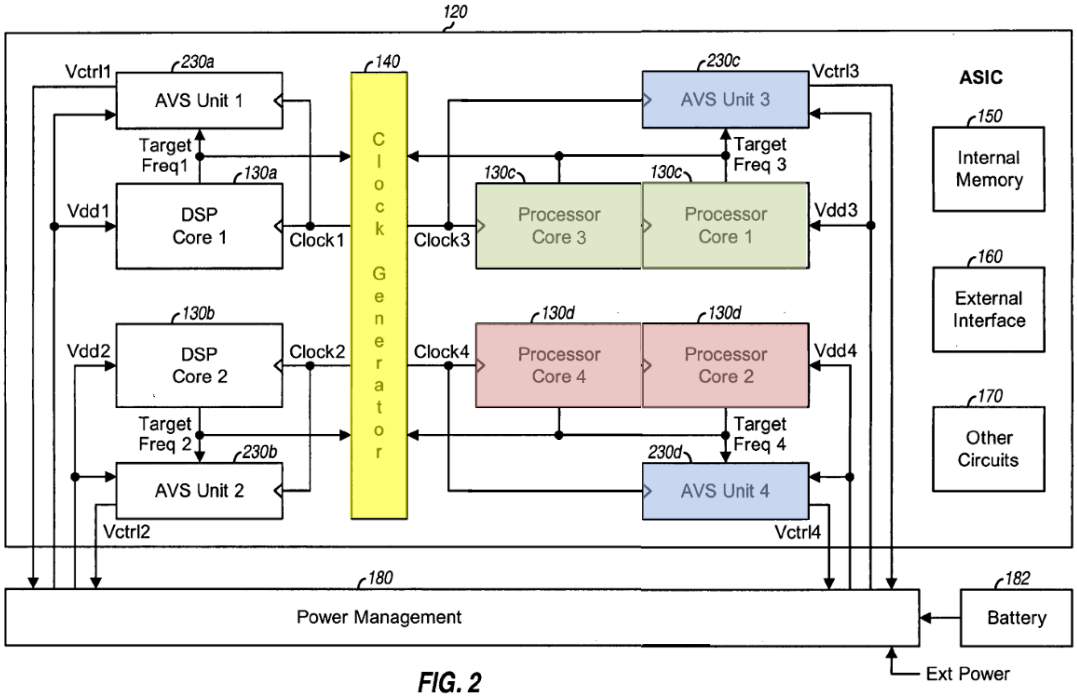


FIG. 2
(MODIFIED)

See id., FIG. 2.

145. A POSITA would have also understood that this type of grouping was a well-known approach to balancing competing design factors. For example, a POSITA would have understood that chip space is limited and, as the number of cores increases, adding individual power generation and control circuitry, AVS units and clock generation circuitry for each core becomes more "expensive" both in terms of space and cost. *See* Section VII.A. Additionally, each additional processor core

provides a roughly linear increase in processing power, whereas additional AVS units and clock generation circuitry provide voltage and frequency flexibility that may give diminishing returns at the cost of valuable chip space. Accordingly, a POSITA would have understood both from this knowledge in the art and Elgebaly's teaching of grouping processor cores into sets, each with a common clock and voltage, that it would have been obvious in a chip with four or more processor cores to group the processor cores into two sets, such that each set of processor cores is managed by one of Elgebaly's AVS units that receives a corresponding clock signal from one clock generator 140 as disclosed in Elgebaly. *See* EX1004, FIG. 2; *see also* EX1007, [0011]-[0012] (disclosing that clusters of cores were known in the art).

c) Interface block configured to facilitate communication between sets of processor cores

146. Elgebaly also discloses an internal memory used by the ASIC's processors and a related communication interface. EX1004, [0027]-[0028], FIG. 1. Figure 1, below, shows an interface, which a POSITA would understand to be a bus, between the processor cores and a [yellow] internal memory that "stores data and program codes used by the processing units within ASIC 120." *Id.*, [0028].

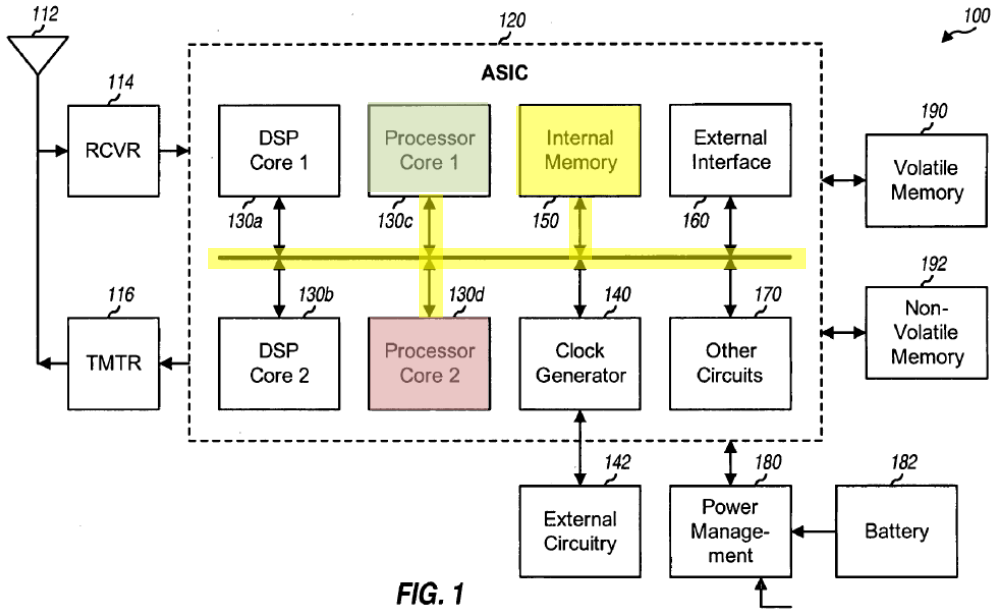


FIG. 1

Id., FIG. 1 (annotated). A POSITA would have understood this figure and the related disclosure to mean that this shared, internal memory facilitates communication between processor cores via the transfer of "data and program codes used by the processing units" (e.g., 130c and 130d).. *See* EX1004, FIG. 1, [0028]; *see also id.*, FIG. 2, [0033]; *supra* Section VII.D.

2. Ozer

147. Ozer discloses a cache coherency method for a processor "compris[ing] a plurality of processor cores [], each provided with its own local cache circuitry." EX1013, [0068]. Each processor core's local cache is updated by a "cache coherency fabric" associated with a "snoop bus" that work in tandem to "ensure that when one processor updates data in its local cache, any copies of that data in the caches associated with the other processors will either be appropriately updated or

invalidated in those caches to ensure coherency." *Id.*, [0072]. The specific actions performed to update each local cache "will then vary dependent on the cache coherency protocol being used." *Id.*, [0072]-[0073]. For example, and as explained in Section VII.D above, a write-update scheme requires local caches to send the updated value of the data to each processor that holds an old copy of the data in its local cache. Thus, an embodiment of Ozer that implements a write-update scheme would send a copy of the modified data from the processor that modified the data to other processors that hold a local copy of the data to "appropriately update[the shared data] … to ensure coherency." *See id.*, [0072].

148. Figure 1A of Ozer depicts an example of the multi-core processor cache coherency system described above:

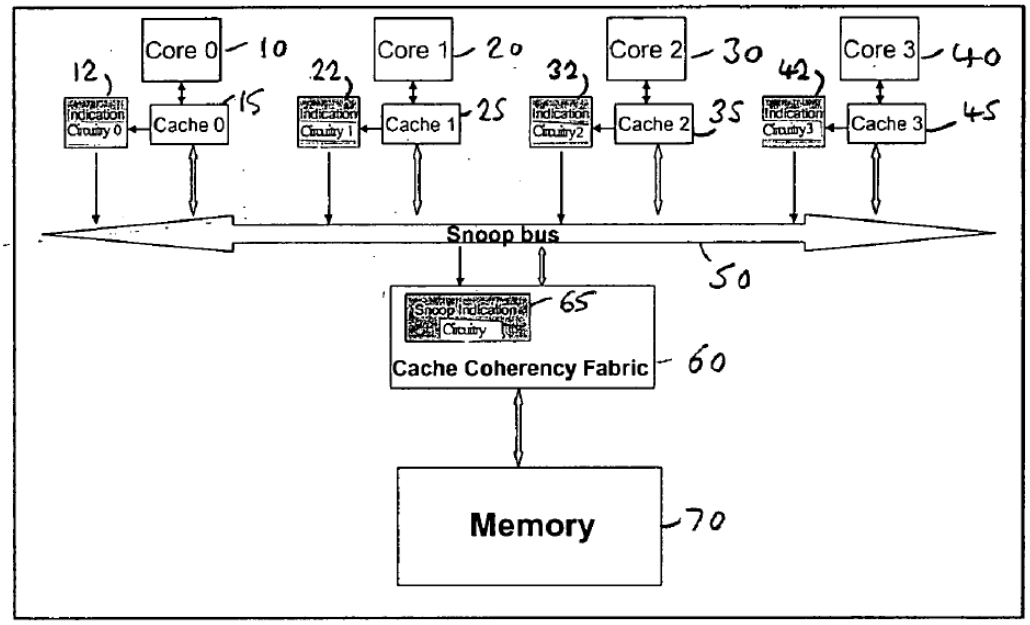


FIG. 1A

Id., FIG. 1A. As shown in the figure above, the multi-core processor includes "a plurality of processor cores 10, 20, 30, 40, each provided with its own local cache circuitry 15, 25, 35, 45," respectively. *Id.*, [0068]. The "cache circuitry may comprise a single cache or may comprise multiple caches." *Id.* Each of the local caches stores data from "memory 70," which "may be entirely formed as shared memory," or include "a number of memory regions, with at least one of those regions being a shared memory region in which the processors 10, 20, 30, 40 can store data that is sharable amongst those processors." *Id.*, [0071]. Since the multiple processor cores share data, "it needs to be ensured that the copy maintained by the local cache

circuitry is up-to-date." *Id.* This coherency is maintained by the "cache coherency fabric 60" using "coherency operations." *Id.*, [0072]-[0073].

149. During the normal operation of each core, "[w]hen a processor core wishes to access data in memory 70, it issues an access request specifying an address of that data in memory." *Id.*, [0069]. Then, as described in Section VII.D with respect to memory caching, generally, "a lookup operation is typically performed in the cache circuitry associated with that processor core in order to determine whether the data can be accessed locally in the cache." *Id.* In response to "certain type[s] of access requests, the cache coherency fabric 60 will initiate a coherency operation." *Id.*, [0073]. Typically, coherency operations are initiated "each time a cache miss occurs on a read request to shared data, or whenever a write access request is issued in respect of shared data." *Id.* When a coherency operation is required, "notifications," or "snoop requests" are "issued to each of the other caches in the system to cause those caches to perform a snoop operation in order to determine whether any of those caches store data at the address specified by the access request." *Id.* The actions taken in response to the snoop request "will then vary dependent on the cache coherency protocol being used." *Id.*

150. Additionally, Ozer aims to reduce chip power consumption by filtering out unneeded snoop operations before they are broadcast along the snoop bus. *Id.*, [0022]. Ozer's cache coherency circuitry uses "segment filtering data" to determine

whether the data subject to a possible coherency operation is "definitely not stored" or is "potentially stored" in a particular segment, which "is used to influence the lookup procedure" performed for each other local cache. *Id.*, [0018]. This lookup procedure determines which other local caches "require subjecting to a snoop operation." *Id.* As a result, the system need not initiate a snoop operation for every other local cache responsive to every access request, which "filter[s] out a significant number of cache snoop operations, thereby giving significant energy savings." *Id.*, [0022].

3. Motivation to Combine Elgebaly and Ozer

151. A POSITA would have understood that the use of a local cache at each processor core was an obvious design choice in a multi-core processor like Elgebaly's. *See* EX1013, FIG. 1A. It has long been known in the art of processor design that the time required to access main memories is a barrier to faster processor speeds. *See e.g.,* EX1026, 1/7. Further, it was well known by 2010 that an effective solution to minimize processor idle time while waiting on memory accesses is a hierarchy of caches, ranging from a small, fast level 1 (L1) local cache, to a medium-sized, medium-speed shared level two (L2) cache that may be on or off chip, to a larger, slower bulk storage or level 3 (L3) cache. *See supra* Section VII.D. A POSITA would have understood Elgebaly's disclosure of an internal memory, in addition to off-chip volatile memory and non-volatile memory to contemplate a

memory hierarchy of this sort. *See also* EX1004, FIG. 1, [0028]-[0029]. Consistent with contemporary designs, a POSITA would also have expected each of the processor cores to include its own local cache. Local caches were common in the art at the time to achieve faster memory access speeds for memory blocks that are repeatedly accessed in a short time period.

152. Relatedly, Ozer explains that "[t]o further improve speed of access to data within such a multi-processing system, it is known to provide each of the processing units with its own local cache in which to store a subset of the data held in the shared memory." EX1013, [0005]. Ozer further explains that using local caches to improve access speed may also "complicate[] the issue of data coherency. *Id.*, [0005]. But cache coherency schemes to manage the coherence problem where known in the art long before Ozer and Ozer also provides one such method for managing cache coherency. *Id.*, Abstract, *see also* Section VII.D.

153. A POSITA therefore would have been motivated to implement certain of Ozer's teachings with multi-core processors of Elgebaly to "improve speed of access to data" and manage cache coherency to realize the benefits of that improved speed. *See* EX1013, [0004]-[0006]; EX1004, [0058] (recognizing memory access delays exist and vary for different types of memory).

154. Additionally, a POSITA would look to Ozer's specific cache coherency implementation because it aligns with Elgebaly's goal of reducing power

consumption. *See* EX1004, [0003], [0008]; EX1013, [0018]-[0021]. Ozer achieves both the increased processor speeds from a hierarchy of caches, while "significantly reducing unnecessary snoop operations, thereby producing significant energy savings." EX1013, [0036]. Therefore, a combination of Elgebaly and Ozer would have the well-known memory latency benefits that come with a local cache for each processor, while also reducing the power consumption of the circuitry required to manage these local caches, which aligns with Elgebaly's broader design goals.

155. Particularly given Elgebaly's and Ozer's shared focus of improving multi-core processors, a POSITA would have understood that such local caches and cache coherency methods would predictably improve the speed of memory access to Elgebaly's internal memory, which "stores data and program codes used by the processing units" and experiences "memory access delays." EX1004, [0028], [0058].

4. Modification of Elgebaly in View of Ozer

156. As discussed above, a POSITA would have been motivated to modify Elgebaly to add a local cache and coherency circuitry to each processor and implement Ozer's cache coherency technique using the on-chip bus depicted in Figure 1 of Elgebaly. By 2010, local processor caches, such as the ones disclosed in Ozer were well known in the art. *See* Section VII.D. Thus, Ozer tells a POSITA ***how*** to implement a well-known feature in the art that is tangential to Elgebaly's

invention, but was commonplace in the art of multi-core processor design in 2010. The specific implementation of the memory hierarchy and cache coherency scheme in Elgebaly would not change the operation of Elgebaly's invention, which is related to adaptive voltage and clock control and is compatible with any common memory hierarchy and cache coherency scheme. *See* EX1004, FIGs. 1-3.

157. For the reasons explained in Sections X.D.1-3 above, at least one obvious combination of Elgebaly and Ozer (the "Elgebaly/Ozer Processor") would have included the structure shown in the modified versions of Elgabaly's Figures 1 and 2, below, in which (a) clock generator 140 includes separate PLLs, each receiving a signal from a respective one of two external oscillators, to generate distinct clock signals to each of AVS Unit 3 and AVS Unit 4, (b) each of Processor Cores 1-4 include local caches, (c) cache coherency protocols are implemented in ASIC 120 to facilitate sharing of data between and for use by Processor Cores 1-4 as detailed in Ozer.

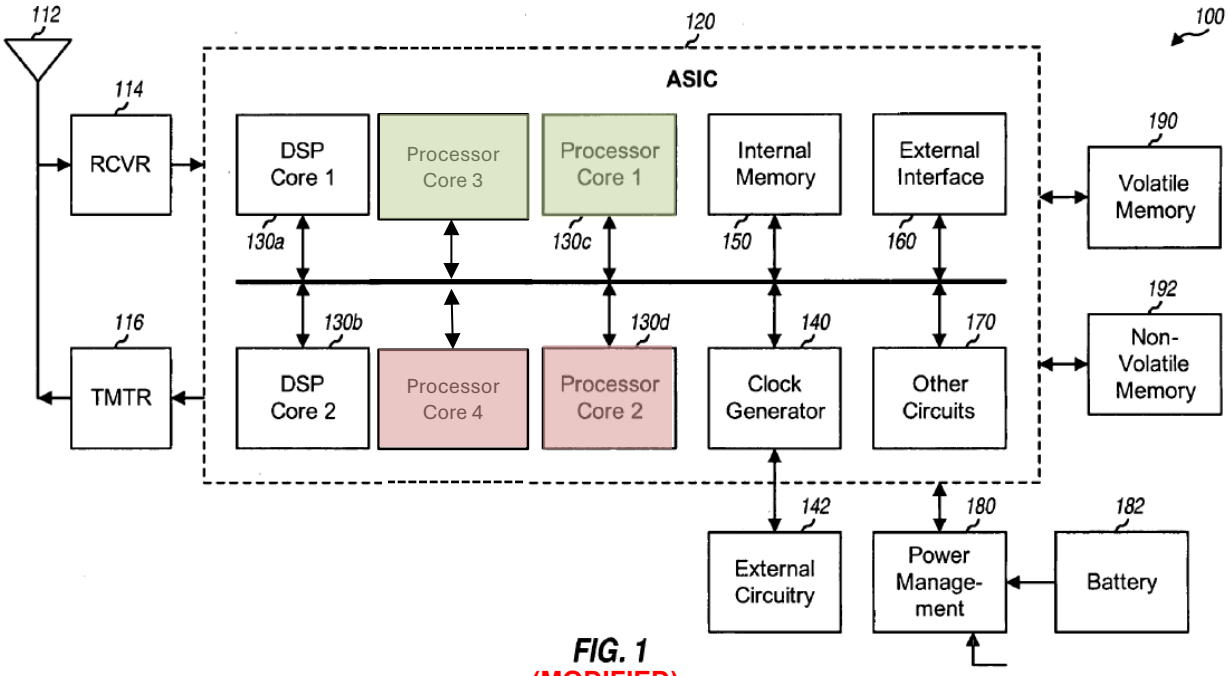


FIG. 1
(MODIFIED)

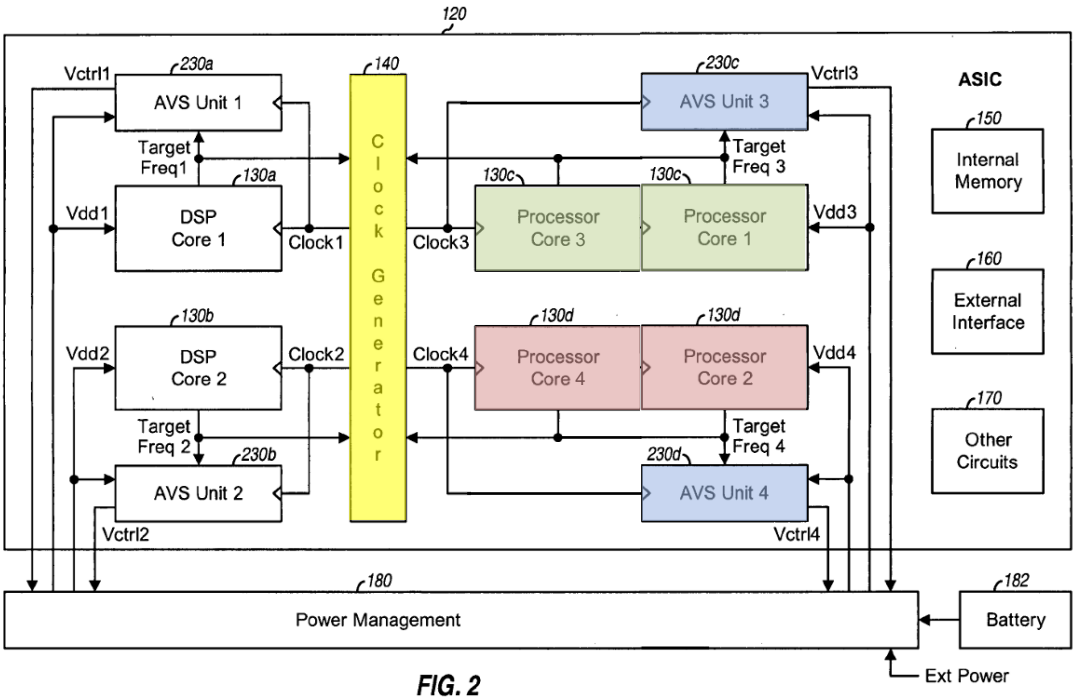


FIG. 2
(MODIFIED)

5. Element-by-Element Analysis

a) Claim 1

[1.pre] A multi-core processor, comprising:

158. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.pre]. As discussed in Section X.D.1, Elgebaly discloses an "ASIC 120 [that] includes various processing units that support monitoring and/or communication with one or more communication systems." EX1004, [0027], FIG. 1. In the embodiment of Elgebaly shown in Figure 1, the ASIC includes two processor cores: "processor cores 130c and 130d." *Id.* In addition, "ASIC 120 may include … any number of processor cores." *Id.*, [0030]. Thus, Elgebaly's ASIC is a multi-core processor.

[1.1] a first set of processor cores of the multi-core processor,

159. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.1]. Elgebaly teaches that its multi-core processor "may include …any number of processor cores." *Id.*, [0030]. A POSITA would have understood this disclosure to mean that Elgebaly's voltage and clock frequency control methods could be duplicated to support adaptive voltage and frequency control of up to N processors on an ASIC chip, where N is limited only by other chip design constraints. Elgebaly also discloses that "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." *Id.*, [0033]. For the reasons explained in Sections X.D.1-4 above, the Elgebaly/Ozer Processor depicted in modified Figure 2 would include a first set of [green] processor cores ("Processor

Core 1" and "Processor Core 3"), and a second set of [red] processor cores ("Processor Core 2" and "Processor Core 4").

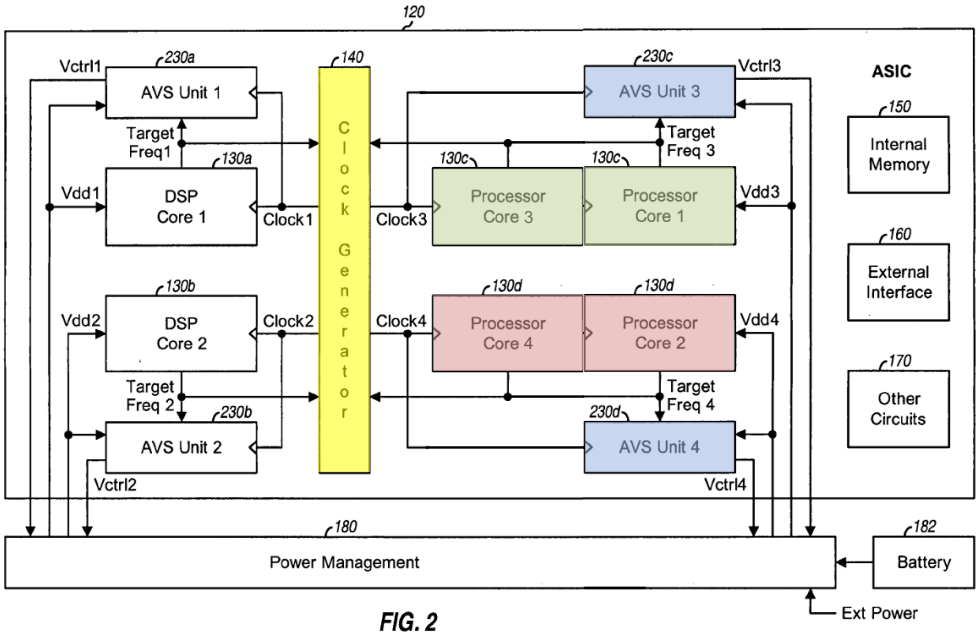


FIG. 2
(MODIFIED)

See id., FIG. 1, [0033]. Consistent with Elgebaly's teaching and as described in Section VII.A and throughout the Petition, multi-core processors containing four cores were well known in the art and available commercially well before 2010. *See e.g.,* EX1023, 2/8, EX1004, [0030], [0033]. Ozer also discloses an embodiment of a multi-core processor that includes four cores and four corresponding local caches. EX1013, FIG. 1A. A POSITA would have understood Elgebaly to teach that its methods of voltage and frequency control could be implemented on an ASIC with four cores, such as the one depicted in modified Figure 1.

160. A POSITA would have further understood that the chip space required to accommodate individual AVS units and clock generation circuitry for each core as the number of cores increases is a costly design choice. Each additional processor core provides a roughly linear increase in processing power, while additional AVS units and clock generation circuitry provide voltage and frequency flexibility that may give diminishing returns at the cost of valuable chip space. Elgebaly understood this trade off, further teaching that "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." EX1004, [0033]. Accordingly, a POSITA would have understood that the processor cores could be grouped into two sets, such that each group of processor cores is managed by one of the AVS units explicitly disclosed in Elgebaly and the clocks for each set share a clock derived from the same clock generator explicitly disclosed in Elgebaly. *See* EX1004, FIG. 2; *see also* EX1007, [0011]-[0012] (disclosing that clusters of cores were known in the art). A POSITA would have added two or more additional processor cores to increase the processing power of the ASIC without requiring additional AVS and clock generation circuitry, consistent with Elgebaly's teaching that cores may share a "common clock and/or a common supply voltage. *See* EX1004, [0033].

161. Thus, the Elgebaly/Ozer combination renders obvious a first set of processor cores.

[1.1.1] wherein each processor core from the first set of processor cores is configured to dynamically receive a first supply voltage and a first output clock signal of a first phase lock loop (PLL) having a first clock signal as input;

162. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.1.1]. As discussed in Section X.D.1.a, Elgebaly discloses that "the clock for each processing core 130 may be set independently by clock generator 140, and the supply voltage for each processing core 130 may be set independently by power management unit 180." EX1004, [0033]. And, as explained in Section X.D.1.b, Elgebaly also discloses that a set of "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." EX1004, [0033]. Thus, for the reasons explained in Sections X.D.1-4 above, in the Elgebaly/Ozer Processor depicted in modified Figure 2, the cores of the first set of [green] processor cores ("Processor Core 1" and "Processor Core 3") would be configured to receive a first supply voltage ("Vdd3") and a first output clock signal ("Clock 3"), and the cores of the second set of [red] processor cores ("Processor Core 2" and "Processor Core 4") would be configured to receive a second supply voltage ("Vdd4") and a first output clock signal ("Clock 4").

163. Additionally, Elgebaly teaches that the supply voltages to the cores should be "adaptive," meaning that the processor "adjusts the supply voltage to a level that satisfies performance requirements," which may change over time, such as when the device enters an "idle mode." *Id.*, [0034], [0006]. When the supply

voltage is "jointly set for [each set of] cores," as described in Elgebaly and further in Limitation [1.1] above, the Elgebaly/Ozer combination discloses the first set of processor cores being configured to dynamically receive the first supply voltage. *See id.*, [0033]. Thus, in the Elgebaly/Ozer Processor, the cores in the first set of processor cores are configured to dynamically receive the first supply voltage, and the cores in the second set are configured to dynamically receive a second supply voltage. *See also* EX1004, [0033].

164. Elgebaly's cores are also configured to dynamically receive respective clock signals. "Each core 130 provides to clock generator 140 and to its AVS unit 230 a target frequency for its clock." EX1004, [0032]. "For each processing core 130, clock generator 140 generates the clock at the target frequency and provides this clock to the core as well as to the associated AVS unit 230." *Id.*

165. Finally, as explained in Section X.D.1.a, Elgebaly discloses or at least renders obvious that clock generator 140 includes a plurality of PLLs each receiving a first clock signal from an external oscillator, with a first of such PLLs generating a first output clock signal to AVS Unit 3 for the first set of cores (Processor Cores 1, 3). *See id.*, [0028].

166. Thus, Elgebaly/Ozer renders obvious each processor core in the first set of processor cores being configured to dynamically receive a first supply voltage

and a first output clock signal of a first phase lock loop (PLL) having a first clock signal as input.

[1.2] a second set of processor cores of the multi-core processor,

167. Elgebaly/Ozer renders obvious Limitation [1.2] for the reasons discussed in Limitation [1.1]. Elgebaly teaches that its multi-core processor "may include … any number of processor cores." EX1004, [0030]. Egebaly also discloses that "multiple processing cores may share a common clock and/or a common supply voltage that may be jointly set for these cores." EX1004, [0033]. For the reasons explained in Sections X.D.1-4 above, the Elgebaly/Ozer Processor depicted in modified Figure 2 would include a second set of [red] processor cores ("Processor Core 2" and "Processor Core 4"). Thus, the Elgebaly/Ozer combination renders obvious a second set of processor cores.

[1.2.1] wherein each processor core from the second set of processor cores is configured to dynamically receive a second supply voltage and a second output clock signal of a second PLL having a second clock signal as input,

168. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.2.1] for reasons discussed in Limitation [1.1.1]. Elgebaly discloses that separate, independent supply voltages and clock signals are provided to each core (or set of cores in the proposed combination). *See* EX1004, [0033]. Further, a POSITA would have understood that separate PLLs must be used to generate the clock signals for each set of cores because Elgebaly teaches that "oscillators," plural, are used to

generate the clock signals for the cores and each PLL contains only one oscillator. *See id.*, [0028]. In addition, the use of separate, independent PLLs for each independent clock signal on a chip was an easily-implemented design that was well known in the art. *See supra* Section VII.B.

[1.2.2] wherein the first supply voltage is independent from the second supply voltage,

169. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.2.2] for reasons discussed in Limitation [1.1.1] and [1.2.1]. Specifically, Elgebaly teaches that "the supply voltage for each processing core 130 may be set independently by power management unit 180." EX1004, [0033]. Similarly, in the Elgebaly/Ozer combination in which one AVS unit is used for each set of cores, a POSITA would understand that the supply voltage for each set of cores would be set independently.

[1.2.3] and the first clock signal is independent from the second clock signal; and

170. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.2.3] for the reasons discussed in Limitations [1.1.1] and [1.2.1]. As explained in Limitation [1.1.1], Elgebaly teaches a clock generator that includes multiple PLLs, one for each independent clock signal. Additionally, Elgebaly's clock generator "may couple to external circuitry, which may include crystals." EX1004, [0028]. A POSITA would have understood this disclosure to mean that one or more external crystal oscillators may supply the input clock signals to each PLL in Elgebaly's clock generator. Thus,

consistent with Elgebaly's disclosure, a POSITA would have found it obvious that Elgebaly's clock generator could be implemented using the multiple crystal oscillator design that was known in the art. *See e.g.,* EX1008, Figure 1, EX1027, [0005]. Specifically, it would have been obvious to a POSITA that the input for each PLL could be generated from separate crystal oscillators that are external to the ASIC, thus driving the PLLs in Elgebaly's clock generator with independent input clock signals (a "first clock signal" and "second clock signal"). *See also supra* Section VII.B.

[1.3] an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores,

171. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.3]. Modified Figure 1 of Elgebaly, above, depicts a communication bus between the components of Elgebaly's multi-core processor that is an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores. *See supra* Limitation [1.1]. Ozer further discloses a snoop bus that enables cache coherency between the local caches of each processor core in the Elgebaly/Ozer combination. *See* EX1013, [0072]. As explained in Section X.D.6, a POSITA would have found it obvious to combine Ozer's snoop bus and the bus depicted in Figure 1 of Elgebaly, rendering obvious an interface block (e.g., cache coherency fabric, snoop bus) couple to the sets of processor cores in the Elgebaly/Ozer Processor.

[1.3.1] wherein the interface block is configured to facilitate communication between the first set of processor cores and the second set of processor cores.

172. In my opinion, Elgebaly/Ozer renders obvious Limitation [1.3.1]. As explained in Sections X.D.3-4, it was obvious to combine Ozer's teachings with those of Elgebaly, rendering obvious at least the Elgebaly/Ozer Processor that included an interface block (e.g., cache coherency fabric, snoop bus) configured to facilitate communication between the sets of processor cores.

173. For example, Ozer explains that per "a typical cache coherency protocol, certain accesses performed by a processor will require a coherency operation to be performed." EX1013, [0007]. "The coherency operation will cause a notification ([aka] … snoop request) to be sent to the other processors identifying the type of access taking place and the address …," which "will cause those other processors to perform certain actions …, and may also in certain instances result in certain information being fed back from one or more of those processors to the processor initiating the access …." EX1013, [0007].

174. Ozer also explains that when a core wishes to read or write to a data location in memory, its cache coherency fabric 60 selectively circulates a cache coherency notification to "other caches in the system to cause those caches to perform a snoop operation in order to determine whether any of those caches store data at the address specified by the access request." EX1013, [0073]. Each processor that receives a snoop request then takes the proper action "dependent on

the cache coherency protocol being used." *Id.*. Under a write-invalidate scheme, as was known in the art, a POSITA would have understood that the processor receiving the snoop request would mark that data in its cache as invalid. *See supra* Section VII.D. Under a write-update scheme, as was also known in the art, a POSITA would have understood that the modified data is sent on the snoop bus from the processor that issued the cache coherency notification to any processors that hold an old copy of the data in their local cache. *See id.*

175. Because the bus interface in the Elgebaly/Ozer combination facilitates communication between cores by passing cache coherency operations and updated data between processor cores, the bus interface is configured to facilitate communication between the first set of processor cores and the second set of processor cores.

b) Claim 2

[2.1] The multi-core processor of claim 1, wherein the interface block further comprises a first level shifter that is referenced to the second supply voltage and adapted to translate first logic levels associated with the first set of processor cores to second logic levels associated with the second set of processor cores for a first signal traveling from the first set of processor cores to the second set of processor cores.

176. In my opinion, Elgebaly/Ozer renders obvious Limitation [2.1] for reasons discussed above in Limitation [1.3.1]. As discussed in Section VII.C, it was well known in the art that multi-core processors implementing multiple voltage domains required mechanisms for translating logic levels between chip components

operating at different voltages. A POSITA would have understood Figure 1 of Elgebaly, which depicts communication between processor cores operating at independent voltage levels, to require a voltage level shifter as part of the bus shown in the center of the ASIC. As discussed, level shifters are an essential feature of chip designs that use multiple voltage domains. *See* Section VII.C. Further, level shifter circuits were well known in the art. *See e.g.,* EX1031, FIGs. 4-5, 5:45-57, EX1039, 1-2/4, Figures 2(a), 2(b), 3. Additionally, a POSITA would have understood that the level shifter that translates between the logic level of the first set of processor cores to the logic level of the second set of processor cores must be referenced to the second supply voltage to know at what logic level the second set of processor cores is operating. *See* EX1039, Figure 2(a). For these reasons, it would have been obvious to a POSITA, based on Elgebaly's disclosure, to implement a first level shifter that is referenced to the second supply voltage and adapted to translate first logic levels associated with the first set of processor cores to second logic levels associated with the second set of processor cores for cache coherency operations between the first and second set of processor cores (i.e., "a first signal traveling from the first set of processor cores to the second set of processor cores").

c) Claim 3

[3.1] The multi-core processor claim 1, wherein the interface block further comprises a second level shifter that is referenced to the first supply voltage and adapted to translate second logic levels associated with the second set of processor cores to first logic levels associated with the first set of processor cores for a second signal traveling from the second set of processor cores to the first set of processor cores.

177. In my opinion, Elgebaly/Ozer renders obvious Limitation [3.1] for reasons discussed above in Limitation [2.1].

d) Claim 5

[5.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a periphery of the multi-core processor.

178. In my opinion, Elgebaly/Ozer renders obvious Limitation [5.1]. Elgebaly discloses "other circuits 170" onboard the ASIC, which "may include a power control unit that controls power to various processing units." EX1004, [0028]. Elgebaly further discloses that the [orange] "other circuits 170," including the power control unit, may be located in a periphery of the ASIC. *Id.*, FIG. 1.

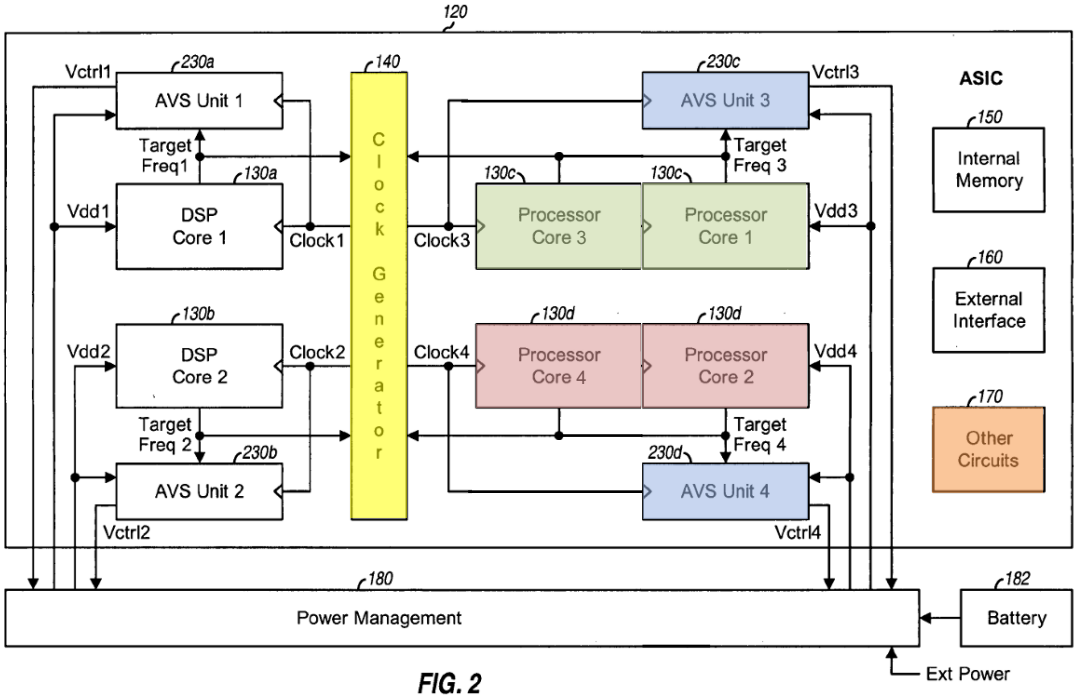


FIG. 2
(MODIFIED)

Id., FIG. 2 (modified). A POSITA would have understood Figure 2 to disclose power control unit—which controls power consumption of the processor cores by sending signals to each core—to be in a periphery of the multi-core processor. Further, as explained in Ground 1 for Limitation [5.1], it was obvious to a POSITA to place control logic circuitry along the perimeter of the multi-core processor. *See also* EX1033, FIG. 8, [0056], [0103]-[0107]. Thus, a POSITA would have understood Elgebaly to disclose—or at least render obvious—a control block that supplies control signals to each processing core being located in a periphery of the multi-core processor.

e) Claim 8

[8.1] The multi-core processor of claim 1, wherein the first set of processor cores are located in a first region of the multi-core processor, and the second set of processor cores are located in a second region of the multi-core processor.

In my opinion, Elgebaly/Ozer renders obvious Limitation [8.1]. As discussed in Section X.D.1, Elgebaly/Ozer includes a first set of processor cores 1 and 3 that share a voltage signal, clock signal, and AVS Unit 3 and a second set of processor cores 2 and 4 that share a different voltage signal, clock signal, and AVS Unit 4. It would have been obvious to a POSITA to arrange processor cores 1 and 3 in a first row constituting a first region and processor cores 2 and 4 in a second row constituting a second non-overlapping region. An example is shown in Modified Figure 2 of Elgebaly below:

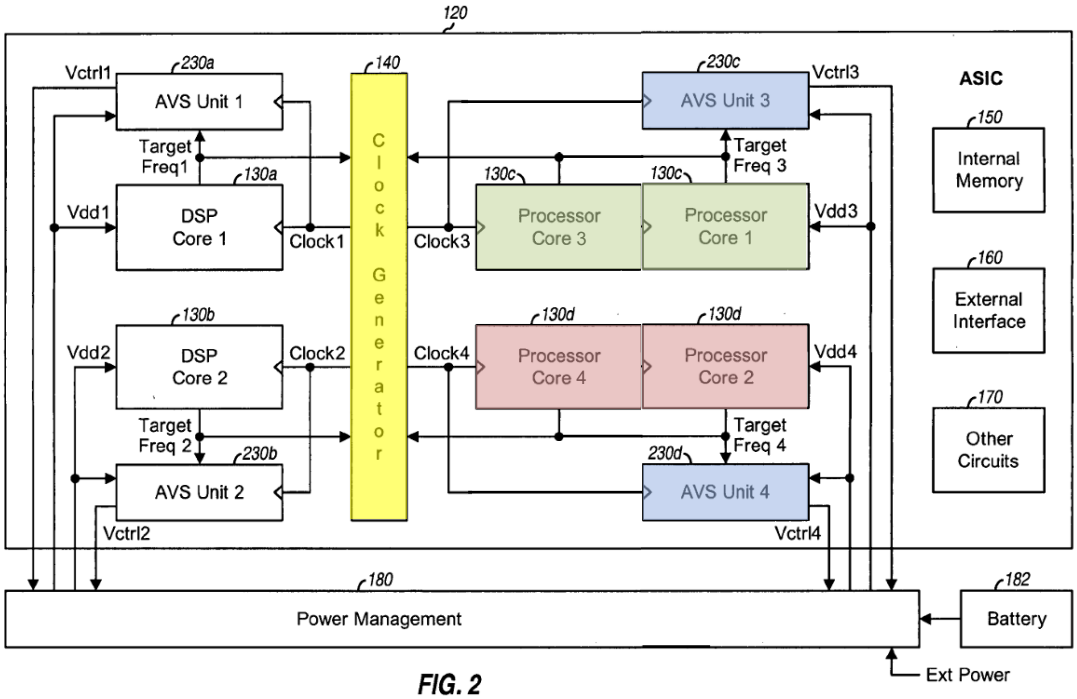


FIG. 2
(MODIFIED)

EX1004, FIG. 2 (modified).

179. First, arranging processor cores in rows was a well-known approach to physical layout of a multicore processor. *See e.g.,* EX1033, FIG. 8, EX1022, 2:8-10, FIGs. 1-2, EX1036, 4:53-57, FIG. 2, EX1011, FIG. 3 Second, as explained with respect to the White/Talwar combination above, arranging the first set of cores in one row and the second set of cores in the second row enables efficient signal routing to route the shared clock signal and voltage signal to each core in a set. Third, as also explained with respect to the White/Talwar combination, placing the cores of each set together and, thus, in non-overlapping regions enables the shared components (i.e., the shared AVS Unit) to be placed close to all cores in the set. Accordingly, a POSITA would have understood that organizing sets of cores in rows was one obvious design choice of the possible choices for the physical layout of cores on the ASIC.

f) Claim 10

[10.1] The multi-core processor of claim 8, wherein the first region and the second region are non-overlapping regions of the multi-core processor.

180. In my opinion, Elgebaly/Ozer renders obvious Limitation [10.1] for reasons discussed in Limitation [8.1].

g) Claim 11

[11.1] The multi-core processor of claim 8, wherein the first region corresponds to a first row of the multi-core processor, and wherein the second region corresponds to a second row of the multi-core processor.

181. Elgebaly/Ozer renders obvious Limitation [11.1] for reasons discussed in Limitation [8.1].

h) Claim 14

[14.1] The multi-core processor of claim 1, wherein the first set of processor cores and the second set of processor cores are configured to receive one or more control signals from one or more control blocks located in a common region that is substantially central to the first set of processor cores and the second set of processor cores.

182. In my opinion, Elgebaly/Ozer renders obvious Limitation [14.1]. Ozer discloses a "cache coherency fabric 60 [that] is located centrally" and "issue[s] snoop requests to any such identified caches identifying the segments to be subjected to the snoop operation." EX1013, [0077], FIG. 1A. Ozer also discloses that "the cache coherency fabric 60 will ensure that when one processor updates data in its local cache, any copies of that data in the caches associated with the other processors will either be appropriately updated or invalidated in those caches to ensure coherency." *Id.*, [0072]. As depicted in Figure 1A of Ozer, the cache coherency fabric is located between the snoop bus and the memory shared by the processors as it interfaces directly with both components. EX1013, FIG. 1A. A POSITA would have understood this to mean that Ozer discloses a control block located substantially central to the first set of processor cores and the second set of processor

cores. As explained in Sections X.D.3-X.D.4, a POSITA would have been motivated to incorporate Ozer's cache coherency fabric as a control block within Elgebaly's multi-core processor to send control signals to each of the processor cores to ensure coherency between the local caches at each core. *See also* EX1004, [0028]-[0028], [0048]; EX1013, [0004]-[0006], [0072], FIG. 1A. In addition, both Ozer and Elgebaly teach that the interface between the on-chip bus and the internal memory is substantially central to the processor cores. EX1004, FIG. 1, [0028]; EX1013, [0077].

i) Claim 21

[21.pre] A multi-core processor, comprising:

183. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.pre] for reasons discussed in Limitation [1.pre].

[21.1] a first set of processor cores of the multi-core processor,

184. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.1] for reasons discussed in Limitation [1.1].

[21.1.1] wherein each processor core from the first set of processor cores is configured to dynamically receive a first supply voltage from a power control block and a first output clock signal from a first phase lock loop (PLL) having a first clock signal as input in a clock control block;

185. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.1.1] for reasons discussed in Limitation [1.1.1].

[21.2] a second set of processor cores of the multi-core processor,

186. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.2] for reasons discussed in Limitation [1.2].

[21.2.1] wherein each processor core from the second set of processor cores is configured to dynamically receive a second supply voltage from the power control block and a second output clock signal from a second PLL having a second clock signal as input in the clock control block,

187. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.2.1] for reasons discussed in Limitation [1.2.1].

[21.2.2] wherein the first supply voltage is independent from the second supply voltage, and

188. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.2.2] for reasons discussed in Limitation [1.2.2].

[21.2.3] the first clock signal is independent from the second clock signal; and

189. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.2.3] for reasons discussed in Limitation [1.2.3].

[21.3] an interface block coupled to the first set of processor cores and also coupled to the second set of processor cores,

190. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.3] for reasons discussed in Limitation [1.3].

[21.3.1] wherein the interface block is configured to facilitate communication between the first set of processor cores and the second set of processor cores.

191. In my opinion, Elgebaly/Ozer renders obvious Limitation [21.3.1] for reasons discussed in Limitation [1.3.1].

E. Ground 5: Elgebaly, Ozer, and Vajda Render Obvious Claims 8-9

1. Motivation to Combine Elgebaly, Ozer, and Vajda

192. Similar to the White/Talwar/Vajda combination in Ground 3, a POSITA would have been motivated to apply Vajda's core layout guidance in defining the physical layout of cores in Elgabaly/Ozer to obtain Vajda's stated benefit of a layout that may "avoid hot spots and over-heating." *See* EX1011, [0062]. As discussed in Section X.D.1, Elgebaly/Ozer includes a first set of processor cores 1 and 3 that share voltage and clock signals and a second set of processor cores 2 and 4 that share voltage and clock signals that are "set independently" from other voltage and clock signals, thus enabling operating one set at a different higher frequency—producing more heat—than the other. Thus, the problem Vajda seeks to address where a layout includes "processor cores 231 that can execute at higher frequencies" and "other processor cores 232, running at lower frequencies" is applicable to Elgebaly/Ozer, and a POSITA would be motivated to apply Vajda's layout guidance to "avoid hot spots and over-heating." *See* EX1011, [0062].

2. Modification of Elegbaly/Ozer in View of Vajda

193. As discussed in Section X.D.1, Elgebaly/Ozer includes a first set of processor cores 1 and 3 that share voltage and clock signals and a second set of processor cores 2 and 4 that share voltage and clock signals that are "set independently" from other voltage and clock signals. In accordance with Vajda's disclosure that "the processor cores 231 that can execute at higher frequencies may be spread out as evenly as possible and surrounded by other processor cores 232, running at lower frequencies" (EX1011, [0062]), a POSITA would be motivated to arrange the cores of the first set and the second set in a checkerboard pattern (e.g., a first row of processor core 1 followed by 2 and a second row of processor core 4 followed by 3) in order to "spread out" cores of one cluster from each other and avoid any hot spots created by grouping of cores of one cluster operating at a higher frequency.

3. Element-by-Element Analysis

a) Claim 8

[8.1] The multi-core processor of claim 1, wherein the first set of processor cores are located in a first region of the multi-core processor, and the second set of processor cores are located in a second region of the multi-core processor.

194. In my opinion, Elgebaly/Ozer/Vajda renders obvious Limitation [8.1]. As discussed in Section X.E.2, Elgebaly/Ozer/Vajda includes processor cores 1 and 3 (the first set) and processor cores 2 and 4 (the second set) arranged in a

checkerboard pattern (e.g., a first row of processor core 1 followed by 2 and a second row of processor core 4 followed by 3). The region containing the first set of processor cores is the first region and the region containing the second set of processor cores is the second region. As explained with respect to the White/Talwar/Vajda combination, a checkerboard pattern that separates cores in the same set running at the same voltage and frequency may allow a set to maintain a high voltage and frequency for a longer period of time before the set must be limited due to overheating.

b) Claim 9

[9.1] The multi-core processor of claim 8, wherein the first region and the second region are overlapping regions of the multi-core processor.

195. Elgebaly/Ozer/Vajda renders obvious Limitation [9.1] for reasons discussed in Limitation [8.1]. The first and second regions are an overlapping region because the checkerboard pattern of cores results in overlapping regions containing each set of cores.

F. Grounds 6-7: Grounds 4-5 and Williams Render Obvious Claims 1-3, 5, 8-11, 14, and 21

196. In addition to the obvious combinations discussed in Grounds 4 and 5, above, it would have been obvious to a POSITA to explicitly modify Elgebaly with the clock generation technique disclosed in Williams.

197. Specifically, a POSITA would look to Williams to supplement Elgebaly's disclosure related to clock generation. Williams explains how to design

multi-core processor clock systems, such as Elgebaly's clock generator, a topic tangential to Elgebaly's disclosure. It would have been obvious to a POSITA to combine Williams' disclosure of multiple crystal oscillators that generate independent clock signals for independent clock domains within a chip, with Elgebaly's chip architecture that uses multiple clock domains. *See* EX1004 [0028], FIG. 2. A POSITA would also have known that the use of multiple crystal oscillators is a simple, well-known method to generate multiple, independent clock signals to feed independent clock domains on a multi-core processor. *See* EX1027, [0005]. Such a design is in line with Elgebaly's teaching that the clock generator "may couple to external circuitry 142, which may include crystals, inductors, capacitors, and so on." EX1004, [0028].

198. Relying on this teaching from Elgebaly and further in view of Williams, a POSITA would have implemented Elgebaly's clock generator with the multiple oscillator structure disclosed in Figure 1 of Williams. *See* EX1008, Figure 1. Such a modification would result in the same structure as discussed in Grounds 4-5, above. Accordingly, Grounds 6-7 render the challenged claims obvious for the same reasons as discussed in Grounds 3-4, respectively.

XI. CONCLUSION

199. All statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and these

statements were made with the knowledge that willful false statement and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,

Dated: January 21, 2025

Murali Annavaram, Ph.D.