

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Mark Henrik Sandstrom
U.S. Patent No.: 11,150,948 Attorney Docket No.: 42342-0120IP1
Issue Date: October 19, 2021
Appl. Serial No.: 17/212,903
Filing Date: March 25, 2021
Title: MANAGING PROGRAMMABLE LOGIC-BASED PROCESSING
UNIT ALLOCATION ON A PARALLEL DATA PROCESSING
PLATFORM

DECLARATION OF DR. HENRY H. HOUH
REGARDING U.S. PATENT NO. 11,150,948

I, Dr. Henry H. Houh, of Burlington, MA, declare that:

I. Introduction

1. I have been retained on behalf of Microsoft Corp. (“Microsoft” or “Petitioner”) to offer technical opinions related to U.S. Patent No. 11,150,948 (“the ’948 patent”) (Microsoft-1001). I understand that Microsoft is requesting that the Patent Trial and Appeal Board (“PTAB” or “Board”) institute an *inter partes* review (“IPR”) proceeding of the ’948 patent. Specifically, I have been asked to consider the validity of claims 1–15 of the ’948 Patent (the “Challenged Claims”) in view of prior art, anticipation and obviousness considerations from the perspective of a person of ordinary skill in the art at the time of the invention (“POSITA”) as it relates to the ’948 Patent.

2. I am being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I incur during the course of this work.

My compensation is not contingent upon the results of my study, the substance of my opinions, or the outcome of any proceeding involving the challenged claims. I have no financial interest in the outcome of this matter or on the pending litigation between Petitioner and Patent Owner.

3. My analysis here is based on my years of education, research and experience, as well as my investigation and study of relevant materials, including those cited herein.

4. I may rely upon these materials, my knowledge and experience, and/or additional materials to rebut arguments raised by the Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

5. My analysis of the materials produced in this proceeding is ongoing and I will continue to review any new material as it is provided. This declaration represents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

II. Qualifications

6. I am over the age of 18 and am competent to write this declaration. I have personal knowledge, or have developed knowledge of these technologies

based upon education, training, or experience, of the matters set forth herein.

7. My qualifications and professional experience are described in my curriculum vitae, a copy of which is attached as Appendix A.

8. I received a Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (“MIT”) in 1998. Beforehand, I received a Master of Science degree in Electrical Engineering and Computer Science in 1991, a Bachelor of Science degree in Electrical Engineering and Computer Science in 1989, and a Bachelor of Science degree in Physics in 1990, all from MIT.

9. I am currently self-employed as an independent technical consultant. I am also president of a company that provides supplemental science, technology, engineering, and mathematics (“STEM”) education to children of all ages.

10. I first entered telecommunications in 1987 when I worked as a summer intern at AT&T Bell Laboratories as part of a five-year dual degree program at MIT. During my first summer, I worked designing digital hardware including using programmable logic devices. I continued to work at AT&T Bell Laboratories as part of this MIT program. While at MIT, I was a teaching assistant (“TA”) in the Electrical Engineering and Computer Science Department’s core Computer Architectures course. I first was a TA as a senior for a role typically reserved for graduate students. I later became head TA. The course covered

various topics in computer architectures, including programmable logic arrays, multiprocessor systems and parallel processing. As a TA, I helped write homework assignments, lab assignments, and exams. I also taught in the recitation sections.

11. Later, as part of my doctoral research at MIT from 1991-1998, I was a research assistant in the Telemedia Network Systems (“TNS”) group at the Laboratory for Computer Science. The TNS group built a high-speed gigabit network and created applications that ran over the network. Example applications included ones for remote video capture, processing, and display of video on computer terminals. In addition to working on the design of core network components, designing and building the high-speed links, and designing and writing the device drivers for the interface cards, I also helped design the switch, which was composed of an array of FPGAs at the core.

12. I also set up the group’s web server and helped to build the web pages that initiated the above-mentioned video sessions via a web interface. Vice President Al Gore visited our group in 1996 and received a demonstration of—and remotely drove—a radio-controlled toy car with a wireless video camera mounted on it that was built by our group. This toy car device received commands transmitted over a network from a remote computer, and video data from the toy car was transmitted wirelessly then over a computer network back to the user controller. On occasion, we allowed users visiting our web site to drive the toy car

from their remote computer while they watched the video on their computer. The video stream was encoded by TNSdesigned hardware, streamed over the TNS-designed network, and displayed using TNS-designed software.

13. I defended and submitted my Ph.D. thesis, titled “Designing Networks for Tomorrow’s Traffic,” in January 1998. As part of my thesis research, I analyzed local area and wide area flows to show a more efficient method for routing packets in a network, based on traffic patterns at the time.

14. From 1997 to 1999, I was a Senior Scientist and Engineer at NBX Corporation, a start-up that made business telephone systems for streaming packetized audio over data networks instead of using traditional telephone lines. NBX was later acquired by 3Com Corporation and the phone system is still used today by numerous businesses.

15. As part of my work at NBX, I designed the core audio reconstruction algorithms for the telephones, as well as the packet transmission algorithms. I also designed and validated the core packet transport protocol used by the phone system. The protocol was used for all signaling in the phone system, including for the setup of conference calls. The NBX system also featured a computer interface for initiating phone calls, which could also initiate conference calls. The NBX system also supported the Telephony Application Programming Interface (“TAPI”) that allowed other computer programs to integrate with our system telephony

features. We obtained U.S. Patent No. 6,697,963, entitled “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” as part of this work.

16. From 1999-2004, I was employed by Empirix or its predecessor company, Teradyne. Empirix was a leader in test tools for telecommunications protocols and systems, providing functional testing tools as well as load testing tools. From 2000-2001, I conceived and built a test platform for testing Voiceover-IP (VoIP) based on highly parallel network processors. The first application on this new test platform was a cloud emulator for simulating the effects of transmitting VoIP over a busy network. The platform was based on a multi-core high performance network processor chip.

17. In 2006, as part of my role at BBN Technologies, I helped found PodZinger Inc., now known as RAMP Inc. PodZinger utilized BBN’s speech recognition algorithms to search through the spoken words in audio and video segments. While I was Vice President of Operations and Technology, PodZinger followed its initial prototype with a full streaming audio and video search solution. I also created a social networking web site, which BBN sold to a venture-funded startup company. In the process of creating the web site, I designed and specified the authentication and authorization protocols.

18. I have been awarded several United States patents, and I have several

1. patent publications including the following examples:

- U.S. Patent No. 7,975,296, “Automated security threat testing of web pages”;
- U.S. Patent No. 7,877,736, “Computer language interpretation and optimization for server testing”;
- U.S. Patent No. 7,801,910, “Method and apparatus for timed tagging of media content”;
- U.S. Patent 7,590,542, “Method of Generating Test Scripts Using a Voice-Capable Markup Language”;
- U.S. Patent No. 6,967,963, “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data”;

19. Additional details concerning my professional qualifications, experience, and publications are set forth in my curriculum vitae (“CV”), which is attached as Appendix A.

III. Level of Ordinary Skill in the Art

20. In rendering the opinions set forth in this declaration, I was asked to consider the patent claims and the prior art through the eyes of a POSITA at the time of the alleged invention, which I understand is asserted to be November 4, 2011—the earliest possible priority date of the ’948 Patent. I understand that the

factors considered in determining the ordinary level of skill in a field of art include the level of education and experience of persons working in the field; the types of problems encountered in the field; the teachings of the prior art, and the sophistication of the technology at the time of the alleged invention. I understand that a POSITA is not a specific real individual, but rather is a hypothetical individual having the qualities reflected by the factors above. I understand that a POSITA would also have knowledge from the teachings of the prior art, including the art cited below.

21. Taking these factors into consideration, on or before November 4, 2011, a POSITA relating to the technology of the '948 Patent would have had at least a combination of a Bachelor's Degree (or a similar Master's Degree, or higher degree) in an academic area emphasizing computer science, or a related field, and two or more years of work experience with parallel processing technologies. Additional education or industry experience may compensate for a deficit in one of the other aspects of the requirements stated above.

22. Before November 4, 2011, my level of skill in the art was at least that of a POSITA. I am qualified to provide opinions concerning what a POSITA would have known and understood at that time, and my analysis and conclusions herein are from the perspective of a POSITA as of that date.

IV. Materials Considered and Relied Upon

23. In reaching the conclusions described in this declaration, I have relied on the documents and materials cited herein as well as those identified in this declaration, including the '948 Patent, the prosecution history of the '948 Patent, and prior art references cited herein. These materials comprise patents, related documents, and printed publications. Each of these materials is a type of document that experts in my field would have reasonably relied upon when forming their opinions.

24. I have also relied on my education, training, research, knowledge, and personal and professional experience in the relevant technologies and systems that were already in use prior to, and within the timeframe of the earliest possible priority date of the claimed subject matter in the '948 Patent, which is November 4, 2011.

- EX1001: U.S. Patent No. 11,150,948 to Sandstrom (“the '948 patent”)
- EX1002: Excerpts from the Prosecution History of the '948 patent (“the Prosecution History”)
- EX1004: U.S. Patent Application Publication No. 2009/0187756A1 to Nollet *et al.* (“Nollet”)
- EX1005: Vu Manh Tuan. “A Study on a Multitasking Environment for Dynamically Reconfigurable Processors.” (2009) (“Tuan”)
- EX1006: Euiseong Seo *et al.*, *Energy Efficient Scheduling of Real-Time Tasks on Multicore Processors*, 19 IEEE Transactions On Parallel and Distributed Systems 1540 (Nov. 2008) (“Seo”)

- EX1007: Redaelli, Francesco, et al. “Reconfiguration aware task scheduling for multi-fpga systems.” 4th HiPEAC Workshop on Reconfigurable Computing (2010) (“Redaelli”)
- EX1008: Ewerson Luiz de Souza Carvalho *et al.*, *Dynamic Task Mapping for MPSoCs*, 27 IEEE Design & Test of Computers 26 (Nov. 2010)
- EX1009: Jose Luis Nunez-Yanez *et al.*, *Dynamic Voltage Scaling in a FPGA-Based System-On-Chip*, 2007 International Conference on Field Programmable Logic and Applications, 459-462 (Aug. 2007)
- EX1010: U.S. Pub. No. 2012/0303809 to Parveen Patel *et al.* (“Patel”)
- EX1011: Excerpts from Kai Hwang, Zhiwei Xu, SCALABLE PARALLEL COMPUTING – TECHNOLOGY, ARCHITECTURE PROGRAMMING (1998)
- EX1013: Kuen Hung Tsoi and Wayne Luk, *Axel: A Heterogeneous Cluster with FPGAs and GPUs*, FPGA '10: Proceedings of the 18th annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 115-124 (Feb. 2010)
- EX1014: Amir H. Hormati, *Flexstream: Adaptive Compilation of Streaming Applications for Heterogeneous Architectures*, 2009 18th International Conference on Parallel Architectures and Compilation Techniques (Sep. 2009)

V. Legal Standards

25. I am not a lawyer and do not provide any legal opinions, but I have been advised that certain legal standards are to be applied by technical experts in forming opinions regarding meaning and validity of patent claims. I have applied the legal standards described below, which were provided to me by counsel for the

Patent Owner.

26. It is my understanding that assessing the validity of a U.S. patent based on a prior art analysis involves two steps. First, one must assess what meaning one of ordinary skill in the art would have given the terms. Second, after assessing the meaning of the claim terms, one may then assess validity by comparing a patent claim to the “prior art.” I understand that the teaching of the prior art is viewed through the eyes of a POSITA at the time the invention was made. My analysis as to what constitutes a relevant POSITA is set forth above.

A. Legal Standards for Prior Art

27. I understand that a patent or other publication must first qualify as prior art before it can be used to invalidate a patent claim. For purposes of this declaration, counsel for Petitioner has instructed me to assume that each prior art reference cited in the combinations described below (Nollet, Tuan, Miller and Seo) qualify as prior art to the '948 Patent.

B. Legal Standard for Priority Date

28. I understand that the “priority date” (or “earliest effective filing date” or “Critical Date”) of a patent is the date on which it is filed, or the date on which an earlier application was filed if the patentee properly claims the benefit of the earlier application’s filing date. For purposes of my analysis in this declaration, I have assumed that the '948 Patent is entitled to a priority date of November 4,

2011.

C. Legal Standard for Obviousness

29. My understanding is that a patent claim is invalid as obvious only if the subject matter of the claimed invention “as a whole” would have been obvious to a POSITA at the time the invention was made. To determine the differences between a prior art reference (or a proposed combination of prior art references) and the claims, the question of obviousness is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. Also, obviousness grounds cannot be sustained by mere conclusory statements. Rather, it is necessary to provide some articulated reasoning with rational underpinning to support the legal conclusion of obviousness.

30. I understand that a patent claim that comprises several elements is not proved obvious by simply showing that each of its elements was independently known in the prior art. In my evaluation of whether any claim of the '948 Patent would have been obvious, I considered whether the Petition, or any evidence submitted in this proceeding, presented an articulated reason with a rational basis that would have motivated a POSITA to combine the elements or concepts from the prior art in the same way as in the claimed invention.

31. It is my understanding that there is no single way to define the line

between true inventiveness on one hand—which is patentable—and the application of common sense and ordinary skill to solve a problem on the other hand—which is not patentable. For instance, factors such as market forces or other design incentives may be the source of what produced a change, rather than true inventiveness.

32. I understand that the decision-maker may consider whether the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness. And, the decision-maker may also consider whether there is some teaching or suggestion in the prior art to make the modification or combination of elements recited in the claim at issue. Also, the decision-maker may consider whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way. The decision-maker may also consider whether the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art.

33. I have been instructed by counsel that if any of these considerations are relied upon to reach a conclusion of obviousness, the law requires that the analysis of such a consideration must be made explicit. I understand that the decision-maker must be careful not to determine obviousness using the benefit of

hindsight and that many true inventions might seem obvious after the fact. I understand that the decision-maker should consider obviousness from the position of a POSITA at the time the claimed invention was made, and that the decision-maker should not consider what is known today or what is learned from the teaching of the patent.

34. I understand that in order to determine whether a patent claim is obvious, one must make certain factual findings regarding the claimed invention and the prior art. Specifically, I understand that the following factors must be evaluated to determine whether a claim is obvious: the scope and content of the prior art; the difference or differences, if any, between the claim of the patent and the prior art; the level of ordinary skill in the art at the time the claimed invention was made; and, if available, the objective indicia of non-obviousness, also known as “secondary considerations.”

35. I understand that the secondary considerations include: commercial success of a product due to the merits of the claimed invention; a long felt need for the solution provided by the claimed invention; unsuccessful attempts by others to find the solution provided by the claimed invention; copying of the claimed invention by others; unexpected and superior results from the claimed invention; acceptance by others of the claimed invention as shown by praise from others in the field or from the licensing of the claimed invention; teaching away from the

conventional wisdom in the art at the time of the invention; independent invention of the claimed invention by others before or at about the same time as the named inventor thought of it; and other evidence tending to show obviousness.

36. I understand that, to establish a secondary consideration, the evidence must demonstrate a nexus between that secondary consideration and the claimed invention.

VI. Overview of the '948 Patent

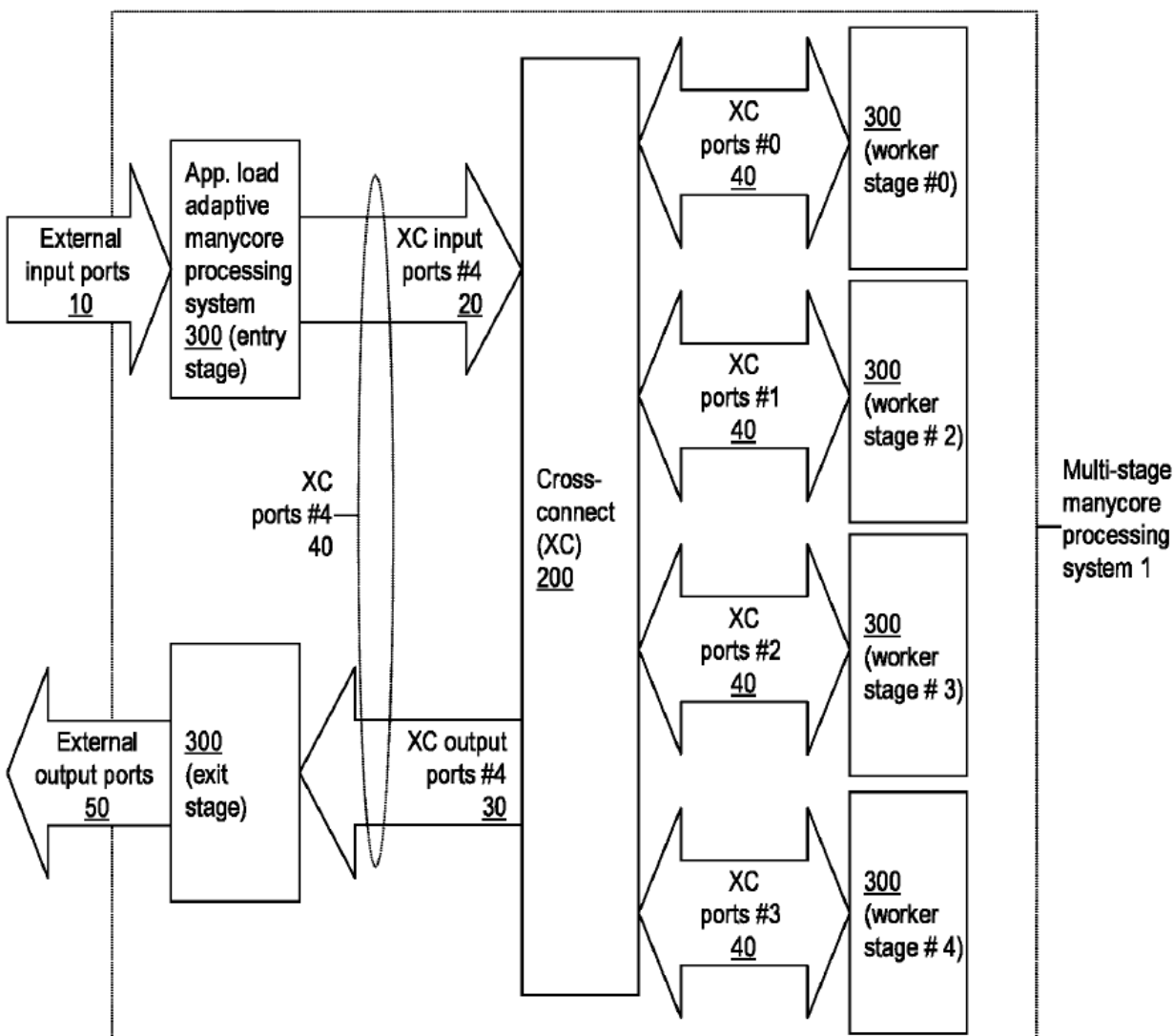
A. Subject Matter Overview

37. The '948 patent is directed to “ techniques for connecting tasks of parallelized programs running on multi-stage manycore processor with each other as well as with external parties with high resource efficiency and high data processing throughput rate.” EX1001, 1:66-2:4.

38. In its discussion of related art, the '948 patent identifies two well understood objectives in advancing computing technology: (1) “maximizing the processing speed of one given computationally intensive program running on a dedicated hardware comprising a large number of parallel processing elements” and (2) “most efficiently shar[ing] a given pool of computing hardware resources among a large number of user application programs.” *Id.*, 2:8-18. In doing so, the '948 patent recognizes within the prior art “a need for a new manycore processor system data flow and IO architecture whose operation, while providing IO data

throughout performance, causes little or no overhead in terms of usage of the computation units of the processor.” *Id.*, 3:18-22. The ’948 patent contends however that “[t]here thus exists a need for a new parallel computing architecture, which, at the same time, enables increasing the speed of executing application programs, including through execution of a given application in parallel across multiple processor cores, as well as improving the utilization of the computing resources available, thereby maximizing the collective application processing throughput for a given cost budget.” *Id.*, 3:46-53.

39. To allegedly address these issues, the ’948 patent describes the “multi-stage manycore processor system” shown in the functional block diagram of Figure 1 (reproduced below). *Id.*, 6:19-21. The block diagram includes “a set of application processing load adaptive manycore processing stages interconnected by a packet destination app-task-inst controlled cross connect.” *Id.*, 8:44-48.

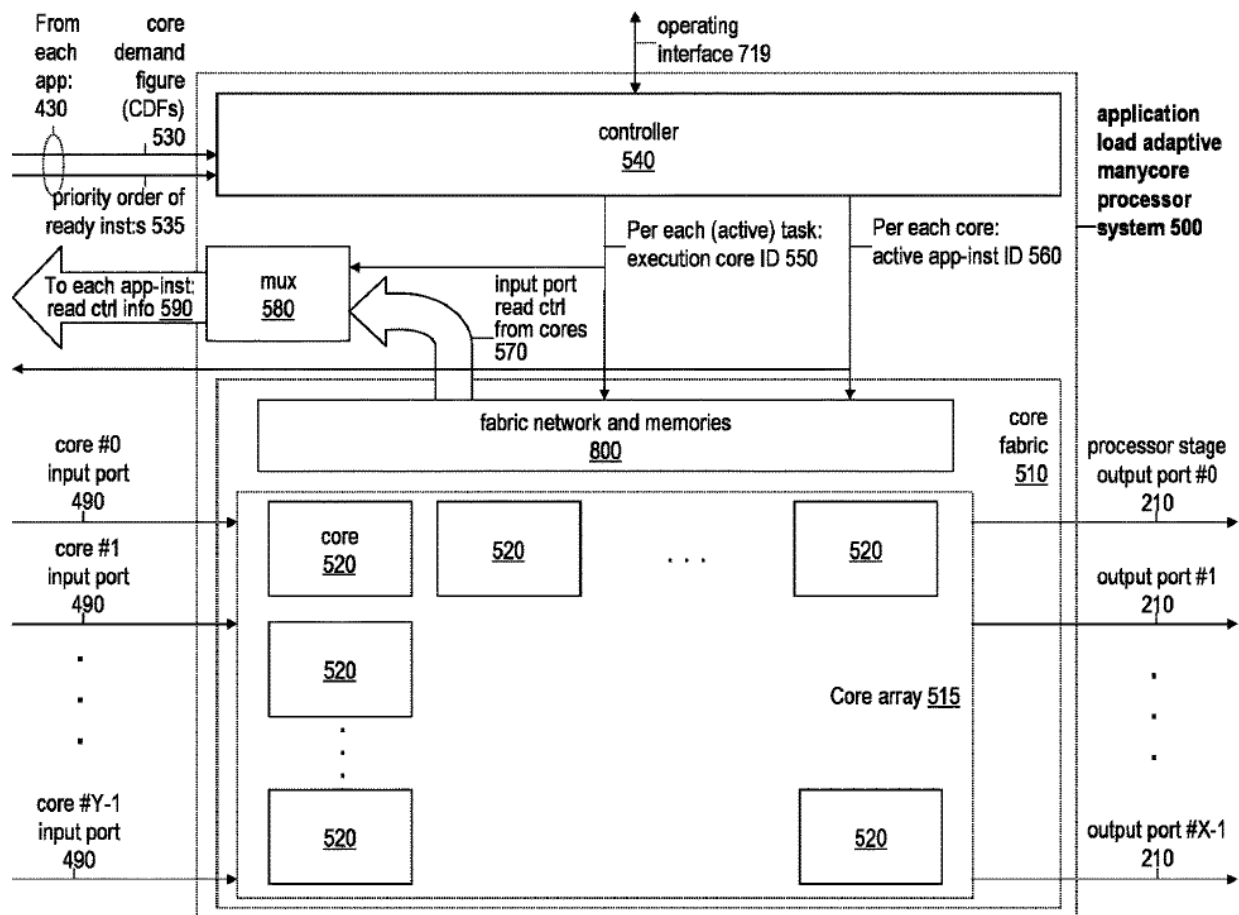


'948 Patent (EX1001), FIG. 1

40. In Figure 1, the “system 1 provides data processing services to be used by external parties (e.g. client portions of programs whose server portions run on the system 1) over networks,” “receives data units (e.g. messages, requests, data packets or streams to be processed) from its users through its RX network ports 10, and transmits the processing results to the relevant parties through its TX network ports 50.” *Id.*, 8:55-64. Data can be passed in the system through the cross-

connect (XC) 200, which include first-in first out buffers (FIFOs) 260, as illustrated in Figure 2. *Id.*, 10:45-11:12, FIG. 2.

41. Figure 5 (reproduced below) shows a functional block diagram for the application load adaptive parallel data processing subsystem 500 for a given manycore processing system 300. *Id.*, 6:33-37. The processor system 500 “comprises an array 515 of processing cores 520, which are dynamically shared among a[sic] the locally hosted tasks of a set of application programs configured to run on the system 1.” *Id.*, 15:12-15. The demand information 430 issued to write “associated applications’ capacity demand indicators 530,” also referred to as “core-demand-figures (CDFs),” to the controller. *Id.*, 15:16-22. The CDFs “express how many cores 520 their associated app-task is presently able to utilize for its ready to execute instances.” *Id.*



'948 Patent (EX1001), FIG. 5

42. “A hardware logic based controller module 540 within the processor system 500, through a periodic process, allocates and assigns the cores 520 of the processor 500 among the set of applications 610 (FIG. 6) and their instances, at least in part based on the CDFs 530 of the applications.” *Id.*, 15:33-37.

B. Brief Summary of Prosecution History

43. The '948 patent issued October 19, 2021, from U.S. Patent Application No. 17/212,903 (“the '903 application”), filed March 25, 2021. EX1001, Cover page. The '903 application was filed as a continuation of several

patent applications and claims priority to several provisional applications dating back to November 4, 2011. EX1002, 410-418.

44. Substantively, the '948 patent prosecution history involves no prior art rejections. *See, generally, id.* Applicant filed the '903 application with one claim that Applicant canceled in a subsequently-filed amendment, adding new claims 2-16. *Id.*, 393, 402, 503. Applicant also filed a Track One request, which the Office granted. *Id.*, 368-369, 505.

45. In a non-final Office action, the Office indicated that claims 2-11 were allowable while claims 12-16 were rejected as directed to non-eligible subject matter. *Id.*, 318-322. Following an examiner interview, the applicant amended independent claim 12 to add “one or more processors executing” computer-executable instructions, as discussed in the interview. *Id.*, 298-306.

46. The Office subsequently issued a Notice of Allowance with no substantive comments concerning the allowance. *Id.*, 279-283. After payment of the issue fee, the applicant filed a petition to withdraw from issuance to cite prior art references, which was granted. *Id.*, 252, 268, 274-278. The Office again issued a Notice of Allowance with no substantive comments concerning the second allowance. *Id.*, 6-7.

47. While the Examiner initialed an IDS that cited Nollet, neither Tuan nor Seo were ever presented to or cited by the Examiner during prosecution.

Rather, the only reference cited by the Examiner during prosecution was U.S. Pub. No. 2012/0303809 (Patel). EX1002, 322. The Patel reference was listed as “prior art of record and not relied upon,” and Patel was never specifically discussed by either the Examiner or Applicant. *Id.* Even so, Patel relates to the communication infrastructure (e.g., packet handling) used for load balancing in a distributed network-based computing system. *See* EX1010, Abstract, claim 1. This is an entirely different context than found in Nollet, Tuan, and Seo, which all relate to parallel processing using multicore/multinode processors in mobile systems.

48. I am not aware of any prior art of record—particularly none that was specifically highlighted in the prosecution history—that relates to dynamic repartitioning and scaling in the context of mobile multicore processing systems, as described by Seo. Thus, the prior art and combinations that I describe below are distinct of any previously considered by the Patent Office in relation to the '948 Patent.

VII. Overview of the Cited References

A. Nollet (EX1004)

49. Nollet discloses a multi-processor computing platform that includes an instruction set processor (ISP) for running “software tasks” and a reconfigurable device (e.g., a Field Programmable Gate Array (FPGA)) for running “hardware tasks.” EX1004, [0057]-[0061]. Such a multi-processor computing platform is

said by Nollet to be particularly useful for “multimedia applications, e.g., on portable devices like a personal digital assistant (PDA) or a mobile phone” *Id.*, [0080]; *see also id.*, [0005], [0011], [0107].

50. Nollet’s multi-processor system relies on an operating systems (OS) that can “provide an environment where several tasks can run concurrently, with minimal of interference between them, but with support for safe data sharing.” *Id.*, [0006]. “The operating system schedules tasks, both on a programmable processor such as an Instruction Set Processor (ISP) and on the reconfigurable logic device and supports a uniform communication framework, which allows tasks to send/receive messages, regardless of their execution location.” *Id.*, [0060]. The ISP can be a standard programmable CPU. *See id.*, [0079]. Thus, the problem addressed by the OS is “how to decide what task to run and where, in case there are multiple processors [] available,” and the OS can base these decisions on “parameters like task priority, processor affinity, etc.” *Id.*, [0082].

51. Nollet’s multi-tasking environment is “capable of executing at least one application, composed of a plurality of tasks.” *Id.*, [0054]-[0055]. This environment includes a “reconfigurable hardware device, compris[ing] a plurality of small configurable logic blocks and an interconnect structure for interconnecting the configurable logic blocks,” where the hardware device “can be a logic gate array, e.g. a Field Programmable Gate Array (FPGA).” *Id.*, [0057]. When the

hardware is reconfigured, the “configuration or function of the logic blocks and/or the interconnection between the configurable logic blocks” are changed. *Id.*

52. The logic blocks are typically arranged in an array and referred to as “tiles” that are “placed edge-to-edge in a tiled arrangement.” *Id.*, [0058]-[0059]; *see also id.*, [0013] (“a reconfigurable matrix (for instance a gate array, especially an FPGA)”). Each tile is “configurable in a variety of configurations,” and each configuration is “capable of running a hardware component or task” so that the “functionality of the hardware reconfigurable device” can be modified from running a first task in a first configuration to running a “second (different) task” in a “second (different) configuration.” *Id.*, [0059].

53. An example of Nollet’s architecture is illustrated in FIG. 1, where various tasks/processes of one or more applications are executed simultaneously on each of the ISP and the tiles of the FPGA-based reconfigurable hardware:

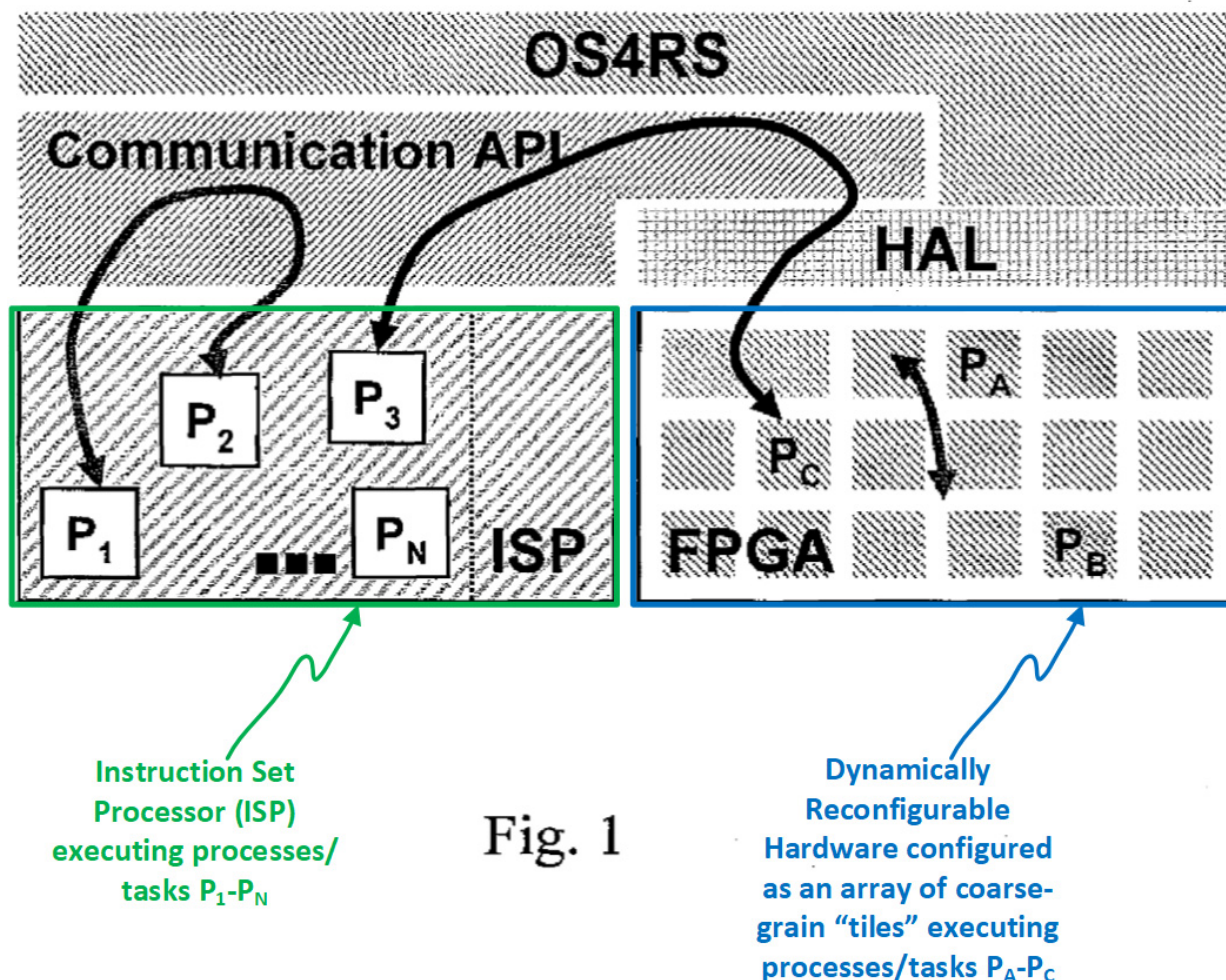


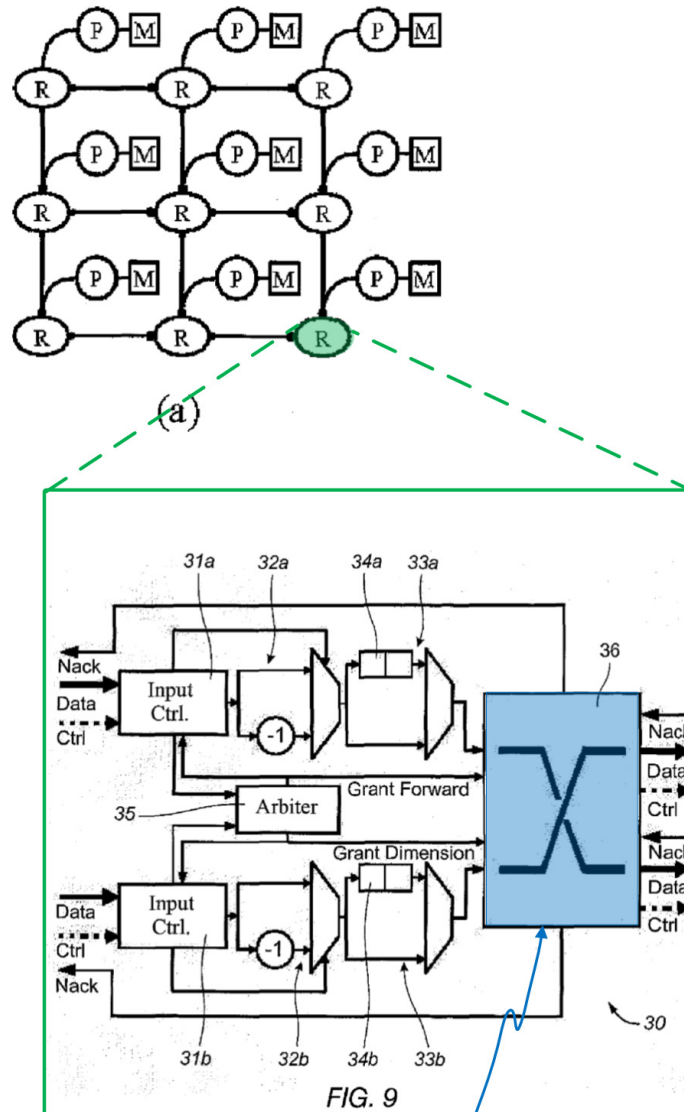
Fig. 1

Nollet (EX1004), Annotated FIG. 1

54. Nollet describes that these various tasks are able to communicate with each other. *Id.*, [0028]. Tasks are able “to send/receive messages regardless of whether they are executed as hardware or as software tasks.” *Id.* To facilitate the execution of the tasks and the communication between them, each tile may act as a “processor,” and the processors are connected via an interconnection network (ICN). *See id.*, [0035], [0103]-[0104]. The ICN can take “various forms of architectures, e.g. k-ary n-cubes, hypercubes, butterflies, crossbars and routing

policies, e.g. virtual cut-through, wormhole, mad-postman switching.” *Id.*, [0104].

In one example, the network can be composed of 1D-routers, like those shown in FIG. 9, interconnected in a mesh topology, as illustrated in FIGS. 7a and 7b. *Id.*, [0104].



Cross-Bar Switch

Nollet (EX1004), Annotated FIGS. 7a and 9

55. Each router may include an “output controller 36 [] composed of a 2-input ***crossbar-switch*** and the NACK.” *Id.*, [0114].¹ Nollet describes that the network can itself “partly comprise logic blocks,” and is thus reconfigurable with the entire device. *Id.*, [0076]. During operation, though, Nollet describes that the interconnection network is “fixed” in order to provide for dynamic partial reconfiguration of the tiles. *See id.*, [0097]-[0100]. And still, Nollet contemplates a crossbar-switch that is itself “reconfigurable” in the sense that it “provide[s] ***dynamic*** connections between all (source, destination) pairs,” and can be “set on or off ***dynamically*** upon program control.” EX1011, 86-88.

56. In an example of task switching and hardware reconfiguration, Nollet describes a scenario where a device user wishes to run two separate applications simultaneously. EX1004, [0107]. The user first starts playing a movie on a movie player, causing a movie decoder to begin performing hardware tasks. *Id.* “When the user is starting his movie decoder the first time, the reconfigurable hardware is free and completely available” so that “the movie player can be started on it, as a hardware task,” and the “computational power of the reconfigurable hardware allows playing the movie at full quality.” *Id.* Later, the user decides to play a 3D game on the device while still playing the movie, but “there are not enough

¹ I’ve added emphasis throughout this declaration, unless otherwise noted.

resources available on the reconfigurable hardware to run both applications.” *Id.* As such, “the movie can be downscaled in resolution and frame rate, resulting in reduced computational requirements for the player,” which “frees up some hardware resources and allows the 3D game to be started as a hardware task.” *Id.* Because these applications are run on reconfigurable hardware, this problem can be solved without requiring different architectures for different multimedia content. *Id.*

57. To manage the different applications running on the hardware/software multitasking platform, Nollet teaches an operating system that manages the various tasks. *Id.*, [0086]. Each application on the platform includes several tasks run on an available hardware/software resource. *Id.* “The operating system preferably keeps track of the available resources, meaning location of all the tasks running. For example, the OS can keep track of which tile of an FPGA is being used to execute a task.” *Id.* In this way, Nollet provides a way of keeping track of where each task is being executed (*i.e.*, which task is being run on which FPGA tiles). *Id.*

58. The system (e.g., via middleware running on top of the OS) is configured to provide Quality of Service (QoS) for the applications running on the system. *Id.*, [0080], [0093]. The system is able to reschedule tasks by “determining the best tradeoff in terms of QoS for the different applications that

run simultaneously on the platform.” *Id.*, [0096]. Thus, the system is able to vary the resources allocated to a given application and its tasks to satisfy the QoS levels required by the various applications. *Id.* “If a task requires, for example, for QoS reasons, to be executed in hardware, one of the less critical tasks running on a hardware tile will be reschedule onto an ISP, providing a free resource for the new task.” *Id.*, [0099]. However, in providing the necessary QoS, Nollet teaches that its system balances the computational power of the system with its power consumption. *Id.*, [0086] (“When a new task is started or an old task is deleted, some of the existing tasks might have to be rescheduled onto another resource (e.g. to provide more computational power to the new task or to decrease the power consumption).”) In this way, Nollet’s system can be “powerful, flexible, *energy conscious* and inexpensive.” *Id.*, [0081].

59. However, Nollet does not constrain the scheduling mechanisms/techniques employed by the OS and middleware to any specific algorithm. Rather, Nollet leaves such implementation details and choices to the POSITA. Tuan and Seo provide examples of the types of task scheduling mechanisms/techniques that a POSITA would have found it obvious to employ when implementing Nollet’s system.

B. Tuan (EX1005)

60. Like the ’948 patent and Nollet, Tuan is directed to “multitasking on

dynamically reconfigurable processors” and a parallel processing hardware architecture. EX1005, 2, 25-26, 102-103, FIG. 6.1. Like Nollet, Tuan recognizes that “[i]mplementing applications on a reconfigurable fabric allows [sic] to improve performance while minimizing power consumption especially for stream applications.” EX1005, 15; EX1004, [0080]-[0081]. Tuan also recognizes that “reconfigurable platforms are not efficient for control and load/store operations; instead they are often used for accelerating certain parts of an application like data-flow kernels.” EX1005, 29; EX1004, [0080]-[0081]. Thus, like Nollet, Tuan describes coupling the reconfigurable platform with a certain type of microprocessor acting as a general-purpose processor (GPP). *Id.* In other words, the SoC architecture contemplated by Tuan is very similar—if not exactly the same—as that described in Nollet.

61. Like the '948 patent, Tuan considers “how to improve the throughput in data flow drive applications by using an appropriate application model and a mapping algorithm to map a target application onto a[n] underlying coarse-grained reconfigurable device consisting of multiple hardware executing units.” EX1005, 2. It also considers “how to extend the multi-thread execution model in a tile-based architecture into a multicore architecture.” *Id.*

62. For example, when describing methods for mapping tasks onto tiles and handling task preemption, Tuan uses the architecture of NEC’s DRP-1. *Id.*,

64, 72, 93. The tile-based architecture of the DRP-1 is illustrated in FIG. 6.2, where each reconfigurable tile (i.e., “DRP core”) is connected to a mesh network of routers. This architecture is very similar to that shown in Nollet’s FIG. 7a. Indeed, like Nollet, the routers in the NEC DRP-1 architecture comprise crossbar switches. EX1005, 103.

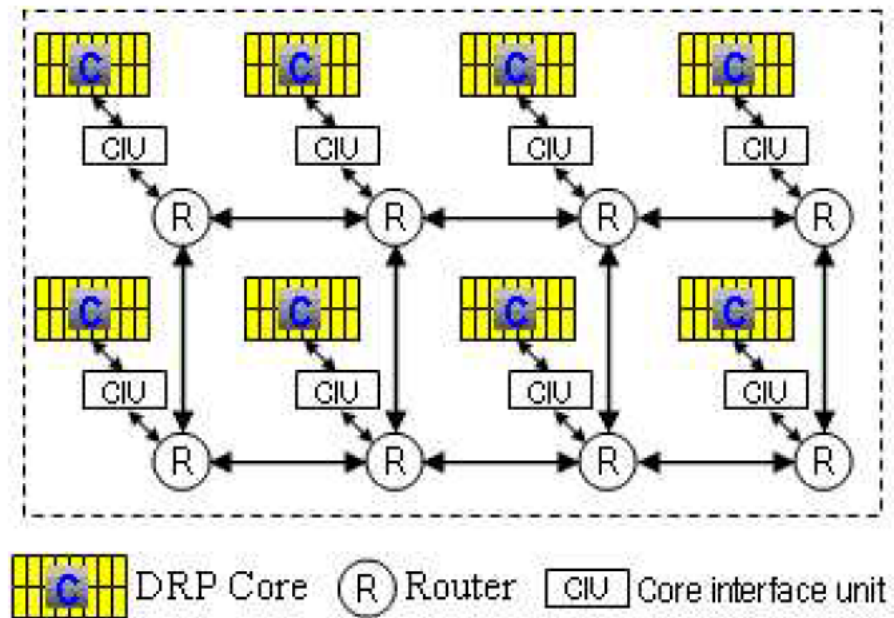


Fig. 6.2: Multicore architecture

Tuan (EX1005), Fig. 6.2

63. With respect to this type of architecture, Tuan describes that tasks can be mapped to a tile, and further explains that the tasks assigned to the tiles can be switched. *Id.*, 30-33, FIG. 2.9. In particular, referencing Figure 2.9 (reproduced below), Tuan shows “task-switching” by the reconfigurable array, explaining that when tasks “finish or are preempted, their correspondent cores are freed” and new

tasks “are assigned to these cores.” *Id.*, 33.

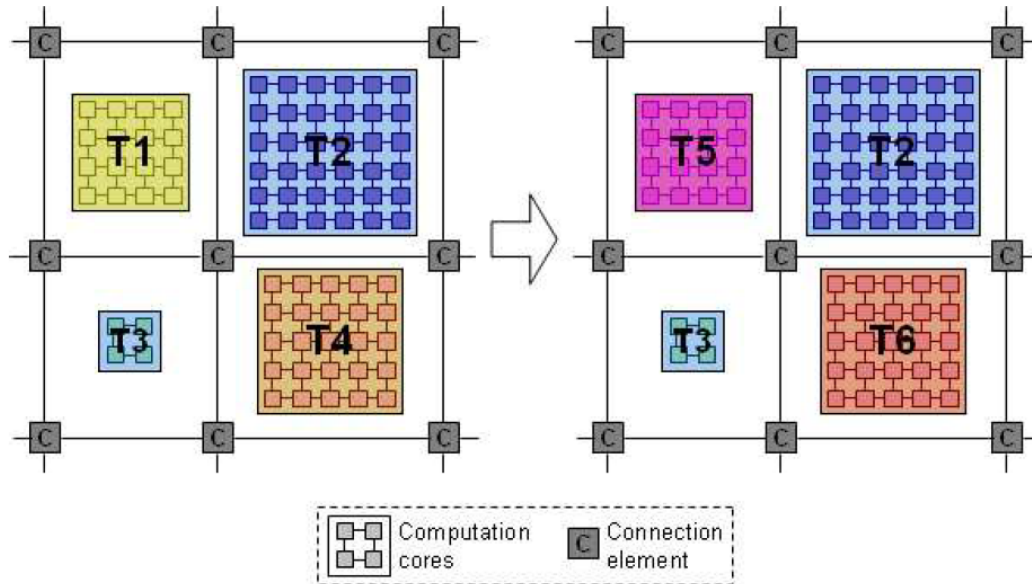


Fig. 2.9: Time-sharing, multicore partitioning model

Tuan (EX1005), FIG. 2.9

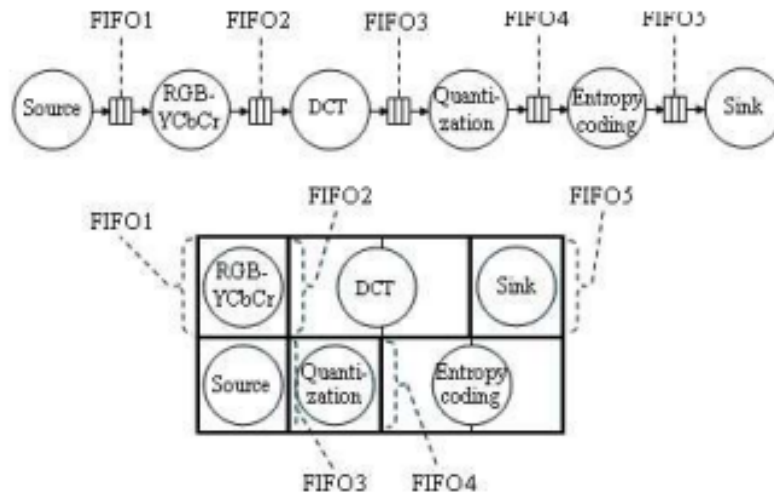
64. Referencing Figure 2.9, Tuan explains:

[A]t the beginning, four tasks T1, T2, T3, and T4 are allocated for execution according to their sizes. When tasks T1 and T4 finish or are preempted, their correspondent cores are freed and tasks T5 and T6 are assigned these cores.

Id., 33. Task switching occurs when task T1 is preempted and removed and task T5 occupies the processing unit—that is, task T1 is switched out for task T5. *Id.*

65. To implement its parallel processing platform, Tuan specifies a mapping step where target applications are “mapped onto a candidate architecture for evaluation.” EX1005, 65. In this mapping process, “a total job” of an application “is represented with multiple processing tasks that can be executed in a

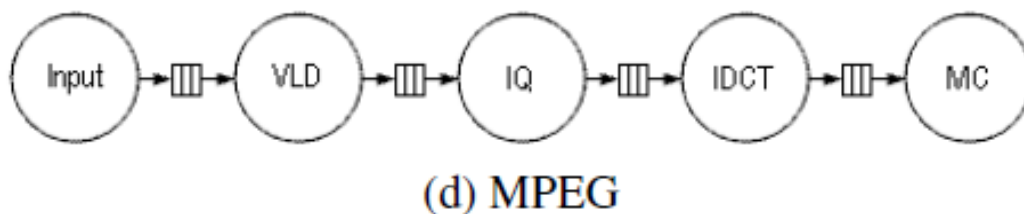
pipelined manner.” *Id.*, 66. For example, a JPEG encoder is mapped in Figure 4.3 (below) to include multiple processing tasks: a source stage, a sink stage, and multiple intermediate processing stages, including RGB-YCbCr conversion, two-dimensional DCT, quantization, and entropy coding stages. *Id.*, 67, 73-74. Each task (stage) is executed by a set of reconfigurable logic-based processing units (e.g., tiles), and passes data blocks that are being processed through a set of FIFOs. *Id.*, 73, 75.



EX1005, FIG. 4.3

66. In another mapping example, Tuan maps an MPEG decoder—a type of video decoder—in Figure 4.9(d) (reproduced below). *Id.*, 74-75, FIG. 4.9. Like the JPEG encoder, the MPEG decoder is mapped to multiple processing tasks: “input, variable length decode (VLD), inverse quantization (IQ), inverse discrete cosine transform (IDCT), and motion compensation (MC),” each of which is

separated from a neighboring task by a FIFO. *Id.*, 75.



Tuan (EX1005), FIG. 4.9

67. Tuan teaches that the execution time of a task can be improved by allocating more resources to the task. EX1005, 68, 76. Depending on the amount of parallelism involved in a task, assigning more tiles to perform a given task can lower the tasks execution time, and increase its performance. *Id.* However, at a certain point, allocating more tiles to perform a task no longer improves execution time and may actually degrade performance. *Id.*, 76-77. “Moreover, since using larger numbers of Tiles with low usage will increase the leakage and the clock distribution power, an optimum number of tiles should be carefully selected for each implementation.” *Id.* In other words, like Nollet, Tuan teaches the need to consider power consumption when mapping tasks to tiles. EX1005, 76; EX1004, [0086]. Thus, Tuan suggests that task mapping/scheduling should be a balance between execution performance and power consumption. *Id.*

68. However, Tuan does not limit how the task mapping/scheduling mechanism in the system implements this balance between execution performance and power consumption. Seo provides one explanation for how to implement this

balance in a multicore processor.

C. Seo (EX1006)

69. Seo recognized that high-throughput multicore processors were gaining popularity in mobile devices. EX1006, 1. However, as recognized in Nollet and Tuan, Seo also recognized that energy efficiency in mobile devices is an “important issue.” *Id.* To account for this problem, Seo describes a real-time task scheduling mechanism for a multicore processor that attempts to optimize power consumption while maintaining the necessary performance to satisfy the demands of the running applications. *Id.*

70. The algorithm presented in Seo utilizes dynamic voltage scaling (DVS), which “adjusts the clock frequency and operating voltage on the fly to meet changes in the performance demand.” *Id.* Seo’s algorithm has two parts: (1) dynamic repartitioning, and (2) dynamic core scaling. *Id.*, 2. Dynamic repartitioning “tries to keep the *performance demands of each core* balanced by migrating tasks from the core with the *highest demand* to the one with the *lowest demand*.” *Id.* Dynamic core scaling “decides on *the optimal number of cores for the current performance demand* and tries to meet this criterion as far as all deadlines are guaranteed.” *Id.* “While under high task loads, having more cores is always more efficient; below a certain task load, having more cores actually consumes more energy.” *Id.*, 7. Thus, dynamic core scaling can “save a

considerable amount of leakage power in low load periods.” *Id.*

71. With respect to the dynamic core scaling portion of the algorithm, Seo describes:

determin[ing] the optimal core number for given [load] L , which is newly updated with each change in the task state. If the optimal number is smaller than the current number of active cores, *try_to_shrink* will be called to reduce the number of active cores. If the optimal number exceeds the current number of active cores, the *Activate* function will be called to increase the number of active cores.

EX1006, 9. Seo’s algorithm determines the optimal core number by considering “the relationship between task set utilization, the number of active cores, and the total power consumption.” *Id.*, 8. Seo calls this the “power consumption expectation function X ,” and it “can be used to determine the number of active cores n that minimizes X for a given [load] L .” *Id.*

72. Seo defines the equation for the dynamic utilization L_n (i.e., load) of a given core C_n in equation (9). *Id.*, 5. Load L_n is based on the most recent actual execution time cc_i of all finished tasks and the Worst Case Execution Time (WCET) w_i of all unfinished tasks, where w_i “is defined as the maximum execution time required to complete [task] τ_i at the highest possible processor frequency.” *Id.*, 4-5. Thus, in order to determine the load L_n of a given core C_n , Seo’s algorithm must monitor at least two characteristics for each core: the most recent actual execution time cc_i of all finished tasks assigned to that core and the Worst

Case Execution Time (WCET) w_i of all unfinished tasks assigned to that core.

VIII. Analysis

73. For the reasons articulated in detail below, and based on my review of the '948 Patent, the file history, and the prior art references cited here, it is my opinion that a POSITA would have readily understood that the Nollet, Tuan, and Seo references—as a whole and in an obvious combination of their total teachings—provide all the elements of claims 1-15.

A. Overview of the Nollet-Tuan Combination

74. As I described above, Nollet and Tuan both relate to parallel processing on dynamically-reconfigurable hardware, including assigning individually-programmable tiles to tasks of an application. EX1004, Abstract, [0017], [0058], FIG. 1; EX1005, 2, 25-26, 30-32, 73, 102-103, FIG. 6.1. The hardware architecture leveraged in each has common features, which are apparent from examination of the following Figures. EX1004, 7a; EX1005, FIG. 6.2.

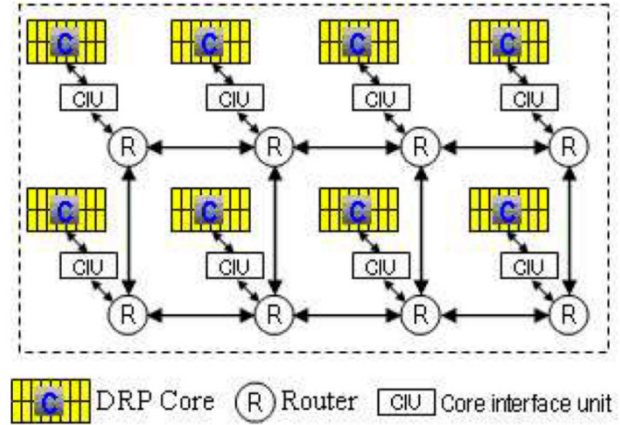
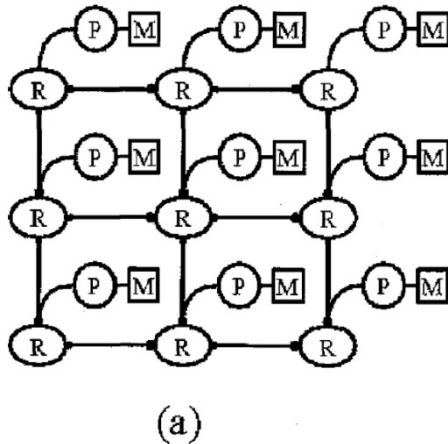


Fig. 6.2: Multicore architecture

Nollet (EX1004), FIG. 7(a)

Tuan (EX1005), FIG. 6.2

75. In my opinion, a POSITA would not understand Nollet to constrain the scheduling mechanisms/techniques employed by its OS and middleware to any specific algorithm. Rather, Nollet leaves such implementation details and choices to the POSITA. Tuan and Seo provide examples of the types of task scheduling mechanisms/techniques that a POSITA would have found it obvious to employ when implementing Nollet's system.

76. Specifically, Nollet describes "allowing a designer to make a representation of an application as communicating threads." EX1004, [0068]. This is accomplished by breaking the application into a plurality of communicating tasks, which can each be mapped on either the reconfigurable matrix/FPGA (denoted hardware tasks) or the ISP (denoted software tasks). EX1004, [0014], [0082], [0086]-[0087], [0102], [0132]-[0133], FIG. 1. And yet, Nollet leaves open

to a POSITA how those tasks would actually be mapped onto the reconfigurable matrix and the specific details of that mapping. *See* EX1004, FIG. 2a-2b, [0144]-[0145].

77. Tuan explicitly teaches representing the processes of an application as multiple stages or tasks and provides example implementation details about how to map those tasks onto a reconfigurable matrix. *See generally* EX1005, 66-79, FIGS. 4.3, 4.9. A POSITA would have found it obvious to implement the reconfigurable multitasking computing platform of Nollet with the task mapping architecture, network manager, and memory of Tuan. A system based on the teachings of both Nollet and Tuan would thus facilitate mapping tiles to tasks of an application by implementing the tasks of an application in a pipelined manner, which “makes it easier to partition an application into a set of parallel communicating tasks,” which is “suitable to be implemented on reconfigurable arrays,” such as Nollet’s reconfigurable device. EX1005, 66.

78. The combined system would further include FIFOs, network manager, and memory to facilitate Nollet’s teaching of inter-task communication and to help manage “information about where tasks are located” after tasks are mapped to tiles or cores. EX1004, [0129]-[0133], FIG. 1; EX1005, 15-16, 27, 28-29, 66, 72-73, 75, 101-111. A POSITA would have been motivated to provide such a combination to implement a mapping approach “to improve throughput and reduce

fragmentation” and to “support[] inter-task communication and input/output operations.” EX1005, 28-29. Indeed, Tuan provides specific implementation details of how to map tasks communicating via switches with FIFO buffers onto a tile-based reconfigurable hardware system utilizing a switch/router-based interconnection network, like the reconfigurable hardware architecture shown in FIG. 7 of Nollet. *See* EX1005, 103-111.

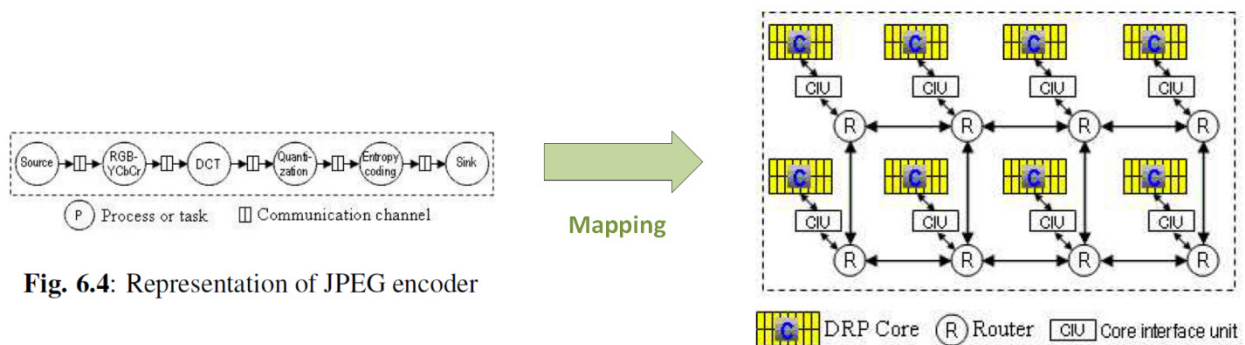


Fig. 6.4: Representation of JPEG encoder

Fig. 6.2: Multicore architecture

Tuan (EX1005), Annotated FIGS. 6.2 and 6.4

79. Tuan further explains that its reconfigurable logic-based processing units are “*often used as an accelerator* for a host processor.” *Id.*, 38 (“A system consisting of a microprocessor and a DRPA could achieve both the high performance with hardware-accelerated execution on the reconfigurable array and the flexibility with software execution on the GPP.”), 29 (“reconfigurable platforms are not efficient for control and load/store operations; instead, they are *often used for accelerating* certain parts of an application like data-flow kernels”). A POSITA would have thus recognized the benefits of using the reconfigurable

logic-based processing units of Nollet as an accelerator to the execution of various programs such as a movie player or a 3D game.

80. Combining the teachings of Nollet with Tuan would have merely involved combining prior art elements according to known methods to yield predictable results. Nollet and Tuan are directed to similar systems that assign reconfigurable tiles/cores of a hardware device to tasks of an application. A POSITA would have been motivated to modify Nollet's platform with Tuan's explicit inclusion of FIFOs, a network manager, and memory to facilitate communications between hardware tasks, keep track of the locations of tiles where each application task is assigned, and managing distribution of tasks to hardware resources. EX1005, 15-16, 27-29, 66, 72-73, 75, 101-111. The use of these components in Tuan would align with Nollet's teaching or suggestion that its platform supports each of these functions. EX1004, [0081] (discussing "manag[ing] the applications by distributing the different tasks on the available resources"), [0086] (discussing "keep[ing] track of which tile of an FPGA is being used to execute a task"), FIG. 1 (showing communication paths between tasks, suggesting multi-stage programs). A POSITA would have understood Nollet's teaching of communication paths between tasks as suggesting the inclusion of multi-stage programs and naturally looked to related references such as Tuan that describe how programs can be partitioned into multiple "tasks that can be executed

in a pipelined manner,” which “allows for parallelism and pipelined processing, which are suitable to be implemented on reconfigurable arrays,” such as those taught in Nollet. EX1005, 66. Moreover, a POSITA would have understood Nollet’s discussion of keeping track of which FGPA tile is being used to execute a task as requiring the use of a system for managing this information and thus naturally looked to related references such as Tuan that describe how “information about where tasks are located” for execution by FPGA tiles is stored and updated using switches for inter-task communication, a network manager, and memory. EX1005, 15-16, 27-29, 66, 72-73, 75, 101-111.

B. Overview of the Nollet-Tuan-Seo Combination

81. As I described in my overviews of Nollet and Tuan, each of Nollet and Tuan teach the need to consider power consumption when mapping tasks to tiles. EX1004, [0086]; EX1005, 76. Together and alone, they suggest that task mapping/scheduling should balance between execution performance and power consumption. *Id.* Seo explains how this balance might be implemented in a multicore processor. Specifically, Seo describes a two-part algorithm for scheduling tasks on a multicore processor, and the algorithm reduces power consumption while maintaining the required deadlines of the tasks. EX1006, 1-2.

82. A POSITA would have found it obvious to utilize Seo’s approach to dynamic repartitioning and core scaling in the context of the Nollet-Tuan multicore

reconfigurable hardware matrix architecture in order to “optimize power consumption during execution” and “reduce leakage power consumption under low load conditions.” EX1006, 1-2. Utilizing the concepts of Seo’s tasking scheduling algorithm would have been consistent with Nollet’s teaching of “manag[ing] the available hardware resources in a consistent, efficient and fair way.” EX1004, [0006].

83. According to Seo, dynamic task scheduling algorithm can be useful for multicore processors that utilize dynamic voltage scaling (DVS). EX1006, 1. While Seo does not specifically mention that the multicore processor can be implemented using reconfigurable hardware, like the tiled reconfigurable fabric described in Nollet and Tuan, POSITAs would have known that dynamic task scheduling algorithms that utilize DVS and are able to activate/deactivate tiles/cores were applicable to reconfigurable fabric, and would have been motivated to do so in order to reduce wasteful energy consumption generally, and because a POSITA would have known that heat buildup is a problem with high performance chips, and taking the opportunity to reduce heat generation to start with would be helpful in controlling excess heat generation. *See, e.g.*, EX1007, 73 (teaching that “it is possible to save power in situations of a low workload: the FPGAs that are not needed can be turned off” and “[t]his same principle can be applied when certain devices have reached high operating temperatures”). In fact,

a POSITA would have known that Nollet describes a type of reconfigurable fabric for use in its system that employs DVS. *See, e.g.*, EX1009, 1 (describing that Virtex-4 Xilinx FPGAs include dynamically reconfigurable clock synthesis technology); EX1004, [0098] (describing the use of Xilinx Virtex™ series hardware for Nollet's FPGA).

C. The Results of the Nollet-Tuan-Seo Combination Would Have Been Predictable to a POSITA

84. The combination of Nollet, Tuan, and Seo would have been predictable and foreseeable because (1) Nollet and Tuan both teach parallel processing on dynamically-reconfigurable hardware arrays by assigning individually-reconfigurable tiles to tasks of an application, (2) Nollet, Tuan, and Seo each seek to schedule tasks in a way that considers “efficiency” and “power consumption,” and (3) the combination merely involves incorporating known features that are explicitly disclosed in Tuan and/or Seo and would have been understood by a POSITA as likely present but not explicitly disclosed by Nollet. *See, e.g.*, EX1004, [0087], [0159], FIG.1. The combination is predictable, at least in part, because elements of the combined system would perform similar functions that had been known to perform prior to the combination. For example, the computing platform of Nollet would still perform the same or similar functions, including assigning tasks of applications to FPGA tiles, as it did prior to the combination with Tuan and Seo. Applying and combining these concepts would

have been obvious from a POSITA's perspective, and a POSITA would have had a reasonable expectation of success in doing so. The systems disclosed in each of Nollet, Tuan, and Seo are determinative computing systems where the operation of the systems and their behavior under varying inputs is predictable.

D. Application of Nollet-Tuan-Seo Combination to the Claims of the '948 Patent

i. Claim 1

85. Claim 1 of the '948 Patent recites:

1. A method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program, the method comprising:

forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first multi-stage program instance is configured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance, wherein the second multi-stage program instance is configured to accelerate the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and

in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

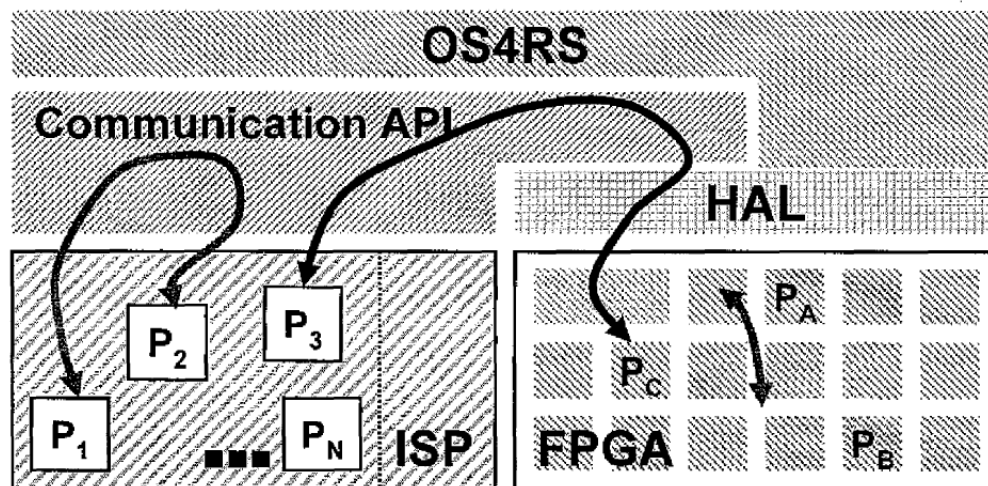
86. It is my opinion that the combination of Nollet, Tuan, and Seo renders

this claim obvious.

[1.0] “A method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program, the method comprising:”

87. Nollet is directed to “multitasking computing systems and methods adapted for reconfigurable computing platforms as well as reconfigurable hardware devices and software, such as operating systems, middleware, or drivers adapted for supporting and executing the methods.” EX1004, [0003]. Nollet indicates that its parallel data processing platform comprises a “reconfigurable matrix [that] can be subdivided into smaller processing units or ‘tiles.’” *Id.*, [0017]. Nollet explains that, “[p]referably, the reconfigurable device, comprises a plurality of tiles, each tile being configurable in a variety of configurations, each of the tile configurations being capable of running a hardware task.” *Id.*, [0020]. Figure 1 (reproduced below) shows an exemplary hardware architecture comprising an FPGA array having a plurality of reconfigurable tiles. *Id.*, [0037], FIG. 1. Because each tile is “configurable in a variety of configurations” and is based on a Field ***Programmable Gate*** Array (FPGA), a POSITA would have understood that each tile is a programmable logic-based processing unit, because a gate is a type of logic. *See id.*, [0081] (“The provided platform is thus a heterogeneous multi-processor platform, containing one or more instruction set processors (ISP) and one or more ***reconfigurable*** matrices (for instance a ***logic gate array***, especially an

FPGA).”). Nollet also discloses a first program (3D game) and a second program (movie player). *Id.*, [0107].



Nollet (EX1004), FIG. 1

88. A POSITA would have understood that the FPGA array of Nollet would act to accelerate a service (e.g., programs running on the device) as FPGAs are well known to do. *See, e.g.*, EX1013, 2 (“In the UNNS approach, each node has a single type of PE such that the installation and management of hardware *accelerators (e.g. FPGA and GPU)* are simpler in a segment of the cluster.”). Indeed, Nollet teaches that the “introduction of the reconfigurable hardware device provides low power consumption for *high computation power by its hardware characteristics* on the one hand, and flexibility by its reconfiguration aspects on the other hand.” EX1004, [0081].

89. For example, Nollet discusses an MP3 player and a movie player. EX1004, [0080]. The MP3 player would necessarily include an MP3 decoder, and

the movie player would similarly include a multimedia decoder (e.g., movie decoder). *Id.*, [0107]; *see also* EX1005, 15, 43, 54, 74 (Table 4.2), 95 (Table 5.1). The FPGA array of Nollet would be used to accelerate the particular decoder (e.g., MP3 decoder or movie decoder) of the respective program. EX1004, [0107] (“The computational power of the reconfigurable hardware allows playing the movie at full quality.”). Moreover, the combination of Nollet and Tuan teaches that using the FPGA array to accelerate the programs on the device would have been obvious. *Id.*

90. Like Nollet, Tuan discloses a parallel processing hardware architecture. EX1005, 25-26, 101-111, FIG. 6.2. In particular, Figure 6.2 of Tuan discloses a “computational cores for a multicore architecture” where “tasks are dynamically assigned into and removed from cores” and “the exact place of a certain task . . . is often changed from time to time either when the system is defragmented or when the task is removed and later resumed.” EX1005, 101-111, FIG. 6.2. Tuan further explains that its reconfigurable logic-based processing units are “often used as an accelerator for a host processor.” *Id.*, 38 (“A system consisting of a microprocessor and a DRPA could achieve both the high performance with **hardware-accelerated execution** on the reconfigurable array and the flexibility with software execution on the GPP.”), 29 (“reconfigurable platforms are not efficient for control and load/store operations; instead, they are

often used for *accelerating certain parts of an application* like data-flow kernels”).

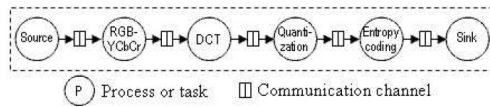


Fig. 6.4: Representation of JPEG encoder

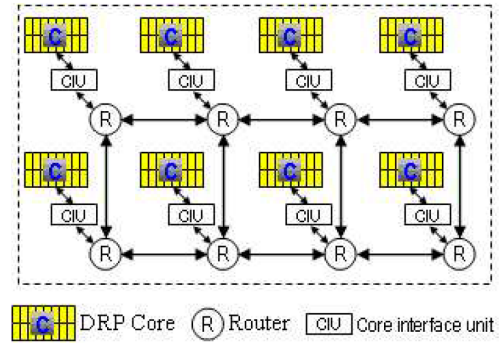


Fig. 6.2: Multicore architecture

Tuan (EX1005), Annotated FIGS. 6.2 and 6.4

[1.1] forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance,

91. Nollet provides “a method for execution of *an application* or a plurality of applications, the execution requiring running substantially simultaneously a *plurality of tasks* (multi-tasking), on the platform.” EX1004, [0017]. This method provides for dynamic instantiation of hardware tasks “by exploiting partial reconfiguration” by subdividing the reconfigurable device into *tiles that are individually programmable*. *Id.* That is, Nollet’s platform allows individual logic-based tiles of a hardware device to be programmed to perform different hardware tasks of one or more applications. *Id.*, [0017], FIG. 1 (showing different tiles executing different hardware tasks P_A , P_B , and P_C on an FPGA).

92. In an example scenario, Nollet discloses programming a first set of

FPGA tiles to execute a first multi-stage program instance (e.g., a multimedia decoder) when a user starts watching a movie. *Id.*, [0107]. Nollet also discloses forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units (e.g., FGPA tiles/cores) to change the configuration or function of the logic units. *Id.*, [0057]. Nollet further explains that “[m]essages 5 between tasks, both scheduled in hardware 8, are routed inside the interconnection network,” where example communication paths between tasks are shown in Figure 1. *Id.*, [0132], FIG. 1. A POSITA would have understood or at least found it obvious that, because the hardware tasks (e.g., the constituent tasks of a multimedia decoder) communicate, the tasks are part of a multi-stage program instance. For example, Nollet discloses an example video decoder where “a send thread 100 passes the coded frames one by one to the decoder thread 102, 106,” which “decodes the frames and sends them, one macroblock at a time, to a receive thread 104 that reconstructs the images and displays them.” EX1004, [0148]-[0149]. As such, a POSITA would have understood the multi-stage nature of the program instances in Nollet. EX1004, [0148]-[0149].

93. Beyond Nollet’s teachings regarding its movie decoder being a multi-stage program instance, Tuan further describes the use of multi-stage program instances in reconfigurable systems. Tuan discloses forming a first set of inter-task communication paths that connect a first set of programmable tiles into a multi-

stage program instance by reconfiguring the communication paths. EX1005, 66, 101-111, FIGS. 4.3, 4.9, 6.2, 6.4, 6.6. For example, Tuan shows that a JPEG encoder in Figure 6.4, which includes a source stage, a sink stage, and multiple intermediate processing stages, including RGB-YCbCr conversion, two-dimensional DCT, quantization, and entropy coding stages. *Id.*, 67, 73-74, 104-105. Each of these stages is executed by a set of programmable logic-based processing units (e.g., cores). *Id.*, 105 (“With the multicore architecture, tasks are mapped onto cores . . . , [and] all tasks belong[ing] to a target application are arranged for executing in a pipelined manner.”). A POSITA would have understood that the “cores” described by Tuan are very similar in function and nature to the “tiles” described by Nollet, and that Tuan’s teachings regarding its “cores” are almost all, if not all, applicable to the “tiles” of Nollet.

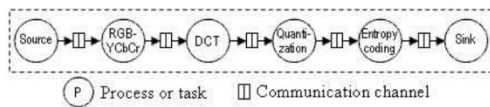


Fig. 6.4: Representation of JPEG encoder



Mapping

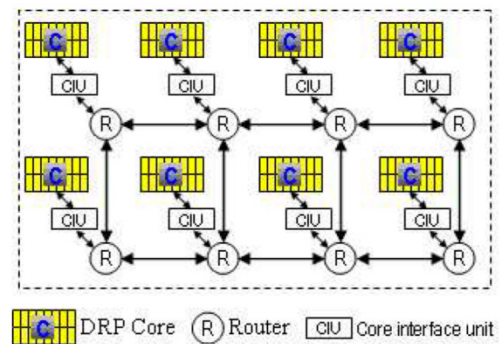
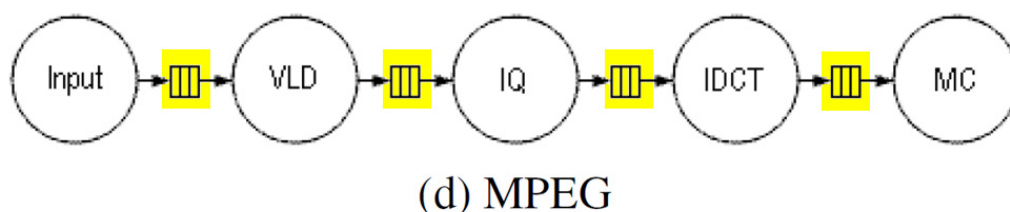


Fig. 6.2: Multicore architecture

Tuan (EX1005), Annotated FIGS. 6.2 and 6.4

94. In another example of mapping programmable tiles to applications—like JPEG encoding, Tuan discusses movie decoders. *Id.*, 73-75. In particular,

Figure 4.9(d) of Tuan (reproduced below) shows an example of a movie decoder—like the multimedia decoder of Nollet—in the form of an MPEG decoder. *Id.*, 74-75, FIG. 4.9. Like the JPEG encoder, the MPEG decoder of Tuan is a multi-stage program instance that includes multiple computation stages including “input, variable length decode (VLD), inverse quantization (IQ), inverse discrete cosine transform (IDCT), and motion compensation (MC).” *Id.*, 75. Thus, a POSITA would have understood that Nollet’s movie decoder is a multi-stage program instance, at least based on Tuan’s more detailed teachings involving the multiple tasks of movie decoders.



Tuan (EX1005), FIG. 4.9 (highlighted)

95. The Nollet-Tuan-Seo combination further discloses that the linking of the processing units is performed through a first set of inter-task communication paths. EX1005, 28-29, 32-33, 35-36, 103-105, FIGS. 4.9, 6.2, 6.4. In particular, the MPEG decoder includes a number of stages, and though not explicitly labeled in Figure 4.9(d), each of the stages of the MPEG decoder is connected to an adjacent stage via a FIFO (highlighted above), through which communication is performed. *Id.*, 66-67, 73, 75, FIG. 4.9(d). While Nollet (and several alternative

embodiments within Tuan) utilizes a mesh network to communicate between tasks, as opposed to fixed FIFOs that directly connect two communicating tasks, a POSITA would have known or at least found it obvious how to port the communication mechanism between these architectures. Indeed, Tuan devotes an entire chapter (Chapter 6) to this very concept—i.e., the alternative use of a Network-on-Ship (NoC) for inter-task communication, as opposed to fixed FIFOs. *See generally* EX1005, 99-111. In the context of Nollet’s mesh network architecture and Tuan’s similar NoC, a POSITA would have understood or at least found it obvious that mechanisms to facilitate inter-task communication and even message buffering would be implemented within each tile as part of the interface between the tile and the routers, or alternatively within the router network itself. For example, Tuan describes a core interface unit (CIU) for each core within the reconfigurable architecture, and the CIU is responsible for facilitating communications between its core and other cores, and thus communications between the tasks that are dynamically assigned to each core at runtime. *See* EX1005, 104-105.

[1.2] wherein the first multi-stage program instance is configured to accelerate the first program, and

96. Based on my review, I do not believe the ’948 patent explicitly describes how a program instance—as opposed to, *e.g.*, the hardware on which the instance executes—is configured to accelerate a program. *See, generally*, EX1001.

To the extent that it discusses accelerating a program, the '948 patent states the following in its description of related art:

Thus, in effect, one branch of computing technology advancement effort has been seeking to **effectively use a large number of parallel processors to accelerate execution of a single application program**, while another branch of the effort has been seeking to efficiently share a single pool of computing capacity among a large number of user applications to improve the utilization of the computing resources.

Id., 2:18-25.

There thus exists a need for a new parallel computing architecture, which, at the same time, **enables increasing the speed of executing application programs, including through execution of a given application in parallel across multiple processor cores**, as well as improving the utilization of the computing resources available, thereby maximizing the collective application processing throughput for a given cost budget.

Id., 2:46-53.

97. The combination of Nollet, Tuan, and Seo combination discloses at least the same type of acceleration of execution of an application program by parallel processors as described in the '948 Patent. In particular, as I discussed above for element [1.0], a POSITA would have understood or at least found it obvious that the FPGA array of Nollet accelerates a service (e.g., programs running on the device) as FPGAs are well known to do. EX1013, 2. In fact, Tuan teaches

that, “[a]s [] mentioned in many research, reconfigurable platforms are not efficient for control and load/store operations; instead, they are often used for **accelerating certain parts of an application**, like data-flow kernels.” EX1005, 29. Moreover, the combination of Nollet and Tuan teaches that using the FPGA array to accelerate the programs on the device would have been obvious. *Id.*; EX1005, 38 (“A system consisting of a microprocessor and a DRPA could achieve both the high performance with hardware-accelerated execution on the reconfigurable array and the flexibility with software execution on the GPP.”), 29 (“reconfigurable platforms are not efficient for control and load/store operations; instead, they are often used for accelerating certain parts of an application like data-flow kernels”).

98. To the extent that [1.2] is interpreted to mean that the first multi-stage program instance itself accelerates the first program, the combination of Nollet and Tuan renders obvious that the first multi-stage program instance accelerates execution of the first program. One reason a POSITA would have found this obvious is because, as described with respect to [1.1], utilizing a multi-stage program instance breaks a program or portion of a program into a set of discrete tasks, some of which can be executed in parallel. *See, e.g.*, EX1005, 64-69 (“... throughput can be enhanced by parallel processing with number of processing elements.”); EX1014, 1 (“Many-core processors provide a lot of flexibility in that they can potentially *speed up the execution* of individual applications (*because of*

increased parallelism), while also having the ability to run many applications at the same time.”), 4 (describing the use of “data parallel actors” to allow work to be evenly distributed across processors). Thus, by employing a multi-stage program instance in the manner known to a POSITA (e.g., as taught in Tuan), a POSITA would have recognized or at least found it obvious that the first multi-stage program instance is configured to accelerate the first program.

[1.3] wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

99. With respect to [1.3], the '948 patent states that its system includes “a hardware implemented data packet switching cross-connect (XC) connecting data packets from an output port of a processing stage to an input port of a given processing stage if a destination software program task of the data packet is hosted at the given processing stage.” EX1001, 3:40-59.

100. Nollet discloses such a reconfigurable cross-connect for interconnecting programmable logic-based processing units. EX1005, [0110]-[0112]. Specifically, Nollet discloses a “reconfigurable hardware device, compris[ing] typically a plurality of small configurable logic blocks and *an interconnect structure for interconnecting the configurable logic blocks*,” and states that “[r]econfiguring the hardware device means changing the configuration or function of the logic blocks and/or the *interconnection between the configurable logic blocks*.” *Id.*, [0057]. Nollet also describes a specific

implementation of its reconfigurable multitasking computing platform that includes FPGAs as the reconfigurable hardware device “having a packet-switched ***interconnection network***” provided by “a 2D torus using wormhole routing,” for example. *Id.*, [0110]; *see also id.*, [0034] (“The communication network can be adapted for supporting packet-switched interconnection network, in particular for supporting message passing communication between tasks, executed by the tiles. ... The communication network may comprise a plurality of routers. Interfaces, for decoupling the tasks from the network may be provided.”), [0035] (“A routing structure for interconnecting the configurable logic blocks (CLB’s) may be provided. A communication network defined on the routing structure for providing communication between the tiles can also be provided. The communication network can comprise a plurality of routers.”).

101. Figure 6 of Nollet further illustrates such a reconfigurable cross-connect. Nollet explains that “[o]n a 2D torus, rows are equivalent to columns, therefore the torus can be decomposed into rows of 1D toruses connected to columns of 1D toruses as shown in FIG. 6.” *Id.*, [0113]. “Two simple 1D routers can be chained to obtain a 2D router and produce a more modular network.” *Id.*

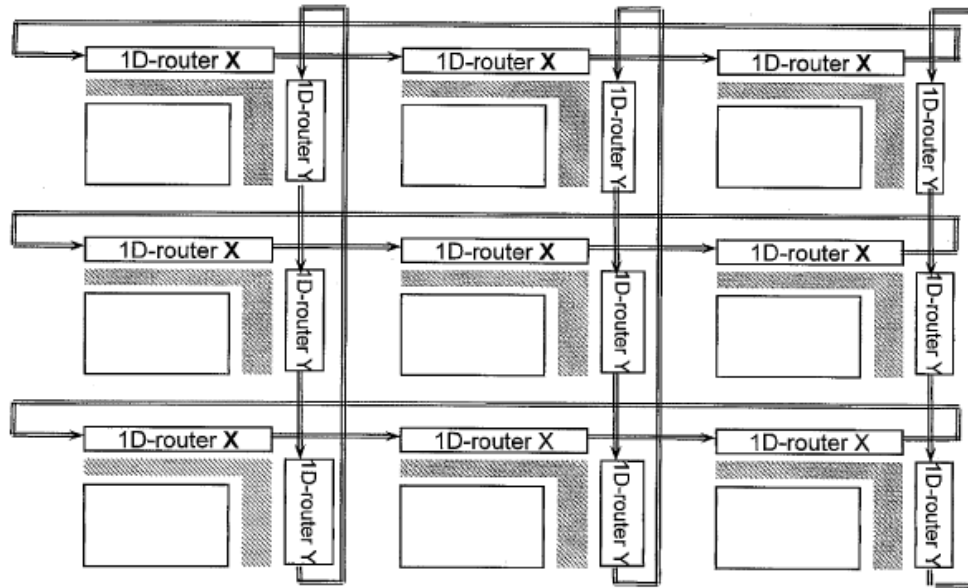


Fig. 6

Nollet (EX1004), FIG. 6

102. Similarly, Tuan describes mapping a multi-stage program onto a multicore processing architecture utilizing switches to communicate between cores/tasks—an architecture that operates in the same manner as that disclosed in Nollet. EX1005, 99-105, FIGS. 6.2, 6.4; EX1004, [0035], [0103]-[0104], [110]-[114]. Thus, a POSITA would have understood that the stages (tasks) of the MPEG decoder would be connected via the first set of inter-task communication paths (i.e., the 2D switch-based interconnection network described in each of Nollet and Tuan) in view of Tuan’s description with respect to FIGS. 6.2 and 6.4. EX1004, FIGS. 7a-7b (showing communication paths between processors/cores/tiles); EX1005, 99-105.

103. Nollet describes that each router may include an “output controller 36

[] composed of a 2-input ***crossbar-switch*** and the NACK.” EX1004, [0114]. Tuan similarly describes its router architecture as relying upon a ***crossbar*** switch.

EX1005, 103-104, FIG. 6.3. As noted above, Nollet describes that the network can itself “partly comprise logic blocks,” and is thus reconfigurable with the entire device. *Id.*, [0076]. This is consistent with Tuan, which describes how the same NEC DRP-1 chip is able to be reconfigured to operate as either a tile-based architecture or a multicore architecture that relies upon a “fixed” interconnection network based on a 2D mesh network of switches. EX1005, 101-104.

104. During operation, Nollet describes that the interconnection network is “fixed” in order to provide for dynamic partial reconfiguration of the tiles. EX1004, [0097]-[0100]. However, a crossbar-switch is itself “reconfigurable” in the sense that it “provide[s] ***dynamic*** connections between all (source, destination) pairs,” and can be “set on or off ***dynamically*** upon program control.” EX1011, 86-88. In other words, one or more crossbar switches are addressable to effectively provide a reconfigurable connection between cores/tiles. This type of “reconfigurable” crossbar switch-based network operates in a very similar manner to cross-connect (XC 830) described in the ’948 Patent, which shows the XC 830 relying upon addressable multiplexors 910. *See* EX1001, 18:20-27, FIG. 9. A POSITA would have found it obvious that the crossbar-switches described in Nollet and Tuan could similarly be configured with addressable multiplexors to

facilitate the dynamic connections.

[1.4] forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance,

105. As I described above with respect to [1.1], Nollet discloses connecting a second set of programmable logic-based processing units (e.g., FPGA tiles) to a second multi-stage program instance of a second program (e.g., a 3D game). EX1004, [0017], [0107]. Specifically, Nollet teaches “dynamically reconfiguring a computing platform in a process for executing ***at least one application*** on the platform.” EX1004, [0019]. For example, Nollet discloses reconfiguring one or more processing units to run 3D gaming tasks of a 3D game. *Id.*, [0105], [0107] (“This frees up some hardware resources and allows the 3D game to be started as a hardware task.”).

106. Tuan discloses a number of different applications that are partitioned and modeled as multi-stage program instances. EX1005, 72-75. For example, Tuan discloses a Viterbi decoder, a JPEG encoder, an MPEG decoder, a Turbo encoder, a G721 encoder, and a G721 decoder. *Id.*, 73-75. It would have been obvious to a POSITA to partition and model the 3D gaming tasks of Nollet as multi-stage program instances on dynamically reconfigurable processors, as taught in Tuan, at least because such an approach “enables efficient sharing of resources due to adequate task scheduling, allows dynamic changes of the application due to

introduction of new tasks during run-time and greatly simplifies development since tasks can be developed quite independent from each other.” *Id.*, 16.

107. The Nollet-Tuan-Seo combination further discloses that the linking of the processing units is performed through forming a second set of inter-task communication paths that connect the processing units. EX1005, 66-67, 73-75, FIGS. 4.3 and 4.9. Indeed, it was well known that multicore systems like those taught in Nollet and Tuan were used to execute multiple applications in parallel by assigning each of the programs and their constituent tasks to a subset of the cores, and that these assignments could be updated dynamically at runtime in accordance with the needs of the applications/tasks. *See, e.g.*, EX1008; EX1014. In particular, the 3D gaming tasks of the Nollet-Tuan-Seo combination include a number of stages, where each of the stages is connected to an adjacent stage via a communication pathway through which communication would be performed. *Id.*, 66-67, 73, 75, 101-110, FIGS. 4.9(d), 6.4. Tuan further describes mapping these multi-stage programs onto the multicore architecture described with respect to FIG. 6.2 such that each stage is executed by a set of programmable logic-based processing units (e.g., cores). *Id.*, 105 (“With the multicore architecture, tasks are mapped onto cores . . . , [and] all tasks belong[ing] to a target application are arranged for executing in a pipelined manner.”). A POSITA would have understood that the stages of the 3D gaming tasks would be connected via a second

set of inter-task communication paths (i.e., the 2D switch-based interconnection network described in each of Nollet and Tuan) in view of Tuan's description with respect to FIGS. 6.2 and 6.4. I described this interconnection network and its re-configurability above with respect to [1.3], and I incorporate discussion here with respect to [1.4]. See EX1005, 99-105.

[1.5] wherein the second multi-stage program instance is configured to accelerate the second program, and

108. Because a POSITA would have understood or at least found it obvious that each of the program instances executed in the Nollet-Tuan-Seo combination are multi-stage and execute on an FPGA-based reconfigurable platform, the second multi-stage program instance is configured to accelerate the second program for the same reasons that I described above with respect to [1.2].

[1.6] wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

109. Because a POSITA would have understood or at least found it obvious that each of the program instances executed in the Nollet-Tuan-Seo combination are multi-stage and are capable of executing in parallel on the same reconfigurable architecture, the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects for the same reasons that I described above with respect to [1.3].

[1.7] in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the

first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and

110. As discussed above for elements [1.1]-[1.6], Nollet discloses dynamically reconfiguring a computing platform for executing at least two programs, including an example where a user is watching a movie (second multi-stage program instance) on a portable device until he decides he wants to play a 3D game (first multi-stage program instance) while a scene in the movie is playing. EX1004, [0019], [0107]. When the user first starts watching the movie, the movie decoder starts and reconfigurable hardware is used for this task. *Id.* “When the user decides to start the 3D game, there are not enough resources available on the reconfigurable hardware to run both [the movie and 3D game] applications,” but “the movie can be downscaled in resolution and frame rate” to reduce computational requirements, “free[ing] up some hardware resources and allow[ing] the 3D game to be started as a hardware task.” *Id.*, [0107].

111. Nollet suggests that, when the user decides to start the 3D game, the movie may be entirely removed from the reconfigurable hardware and instead executed on the software processor. *Id.* However, a POSITA would have known or at least found it obvious that the execution of both the multi-stage program instances of each of the movie and the 3D game could be dynamically updated to

share the reconfigurable hardware such that all or at least some of the tasks of each multi-stage program instance would be run in parallel by the reconfigurable hardware. For example, a POSITA would have known that:

Applications (such as multimedia and networking) running in [multiprocessor systems on a chip (MPSoCs)] generally present a dynamic task workload. This implies a varying number of tasks running at any given moment, possibly exceeding the available hardware resources. Thus, it is necessary to control task operation and system resource use, including dynamic management of task-loading actions.

EX1009, 1. There were various known ways of scaling a program instance to allow for parallel execution on limited resources. One aspect of this process involves the amount of parallel processes employed by a given program instance, as a certain amount of parallel “actors” (e.g., tasks) were known to improve throughput, but excessive parallelism was also known to degrade from the “optimal” solution. EX1014, 3-4. Furthermore, a system may be configured to employ dynamic adaptation “that modifies the active schedule based on the available resources in the architecture.” EX1014, 2. There were several known approaches to dynamic adaptation, and Seo provides one example described in greater detail below. Based at least on these known techniques and the broader teachings of Nollet, Tuan, and Seo, a POSITA would have found it obvious to configure Nollet’s reconfigurable hardware to execute multiple tasks from multiple

program instances at once.

112. More generally, Nollet discloses reallocating at least one processing unit “based on parameters like task priority and processor affinity.” *Id.*, [0008] (“how to decide what task to run and where”), [0082]. Nollet further discloses task preemption, or reallocation of at least one processing unit, and explains that the applications comprise “multiple, possible independent, tasks running in parallel on the reconfigurable matrix.” *Id.*, [0082]. When preemption occurs, “the context of the current task can be saved, in order for this task to continue whenever additional computing resources are assigned to it.” *Id.* Nollet also discloses monitoring the state of the computing resource through a number of variables, which “mainly includes *the load of the computing unit.*” *Id.*, [0160]. In the example above, when the processing load in the FPGA is too high to accommodate a first program instance (3D game), the movie tasks are reallocated to the CPU to make room for the 3D gaming tasks that have just been started, thus experiencing increased demand. *Id.*, [0107]. That is, in response to the detected increase in demand determined by monitoring the load on the processing units, at least one or more programmable logic-based processing units is allocated or added to the first program instance (3D game). *Id.*, [0107], [0160].

113. As I noted in my brief overview of Nollet, Nollet does not constrain the scheduling mechanisms/techniques employed by the OS and middleware to any

specific algorithm. Rather, Nollet leaves such implementation details and choices to the POSITA. Seo provides an example of a task scheduling algorithm that a POSITA would have found it obvious to employ when implementing Nollet's system.

114. As I explained in my brief overview of Seo, Seo describes a real-time task scheduling mechanism for a multicore processor that attempts to optimize power consumption while maintaining the necessary performance to satisfy the demands of the running applications. EX1006, 1. One part of Seo's scheduling algorithm, called dynamic core scaling, "decides on *the optimal number of cores for the current performance demand*" and tries to meet this criterion as far as all deadlines are guaranteed." *Id.*, 2. "While under high task loads, having more cores is always more efficient; below a certain task load, having more cores actually consumes more energy." *Id.*, 7. Thus, dynamic core scaling can "save a considerable amount of leakage power in low load periods." *Id.*

115. Seo's dynamic core scaling algorithm is based on the "power consumption expectation function X ." *Id.*, 8. Seo's dynamic core scaling algorithm attempts to minimize the power consumption expectation function X by determining the optimal core number n for given load L . *Id.*, 8-9. Thus, Seo's dynamic core scaling algorithm requires that the multicore system track the dynamic utilization L_n (i.e., load) of every core C_n in the system. *Id.*

116. Seo defines the equation for the dynamic utilization L_n (i.e., load) of a given core C_n in equation (9). *Id.*, 5. Load L_n is based on the most recent actual execution time cc_i of all finished tasks and the Worst Case Execution Time (WCET) w_i of all unfinished tasks, where w_i “is defined as the maximum execution time required to complete [task] τ_i at the highest possible processor frequency.” *Id.*, 4-5. Thus, in order to determine the load L_n of a given core C_n , Seo’s algorithm must monitor at least two different characteristics for each core: the most recent actual execution time cc_i of all finished tasks assigned to that core and the Worst Case Execution Time (WCET) w_i of all unfinished tasks assigned to that core. A POSITA would have recognized that each of cc_i (first characteristic) and w_i (second characteristic) are characteristics corresponding to the cores running the tasks associated with each of the executing applications (which, in the context of Nollet’s reconfigurable hardware would be both the first and second set of programmable logic-based processing units”) and that their use in evaluating L_n means that these characteristics are used to determine increases in demand for the applications (i.e., “program instances”) associated with the tasks executing on each of the cores.

117. Seo’s algorithm then determines “the *optimal* core number for given [load] L , which is newly updated with each change in the task state.” EX1006, 9. In other words, Seo’s dynamic core scaling algorithm is “an adaptive optimized

resource allocation policy,” as claimed. Specifically:

If the ***optimal*** number is smaller than the current number of active cores, *try_to_shrink* will be called to reduce the number of active cores. If the ***optimal*** number exceeds the current number of active cores, the *Activate* function will be called to increase the number of active cores.

Id. In this way, the dynamic core scaling algorithm “***adjusts*** the number of active cores to reduce leakage power consumption under low load conditions.” *Id.*, 1.

118. When applied to the context of Nollet’s reconfigurable hardware, a POSITA would have found it obvious to utilize Seo’s dynamic core scaling algorithm to determine the optimal number of active tiles/cores in Nollet’s dynamically reconfigurable hardware. If the demand on the tiles/cores by the active tasks exceeds a threshold, the dynamic core scaling algorithm would add an active core/tile (i.e., adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units).

[1.8] in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

119. As I discussed above with respect to [1.7], Nollet discloses dynamically reconfiguring a computing platform for executing at least two

programs, including an example where a user is watching a movie (second multi-stage program instance) on a portable device until he decides he wants to play a 3D game (first multi-stage program instance) while a scene in the movie is playing. EX1004, [0019], [0107].

120. Seo's dynamic core scaling algorithm must monitor at least two different characteristics for each core: the most recent actual execution time cc_i of all finished tasks assigned to that core and the Worst Case Execution Time (WCET) w_i of all unfinished tasks assigned to that core. *See* EX1006, 4-5, 8-9. A POSITA would have recognized that each of cc_i (first characteristic) and w_i (second characteristic) are characteristics corresponding to the cores running the tasks associated with each of the executing applications (which, in the context of Nollet's reconfigurable hardware would be both the first and second "set of programmable logic-based processing units") and that their use in evaluating L_n means that these characteristics are used to determine increases in demand for the applications (i.e., "program instances") associated with the tasks executing on each of the cores.

121. When applied to the context of Nollet's reconfigurable hardware, a POSITA would have found it obvious to utilize Seo's dynamic core scaling algorithm to determine the optimal number of active tiles/cores in Nollet's dynamically reconfigurable hardware. If the demand on the tiles/cores by the

active tasks exceeds a threshold, the dynamic core scaling algorithm would add an active core/tile (i.e., adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units).

ii. Claim 2

122. Claim 2 of the '948 Patent recites:

2. The method of claim 1, wherein the allocation policy relates to balancing load among programmable logic-based processing units.

123. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

124. Specifically, Seo describes that:

An activation of a slept core occurs when the current active core number is less than the power-optimal number. The decision of whether or not to expand is made whenever L increases. It is also done when a task was newly released, and it is unable to be scheduled in C_{min} [i.e., the core with the lowest L].

EX1006, 9. After the activation operation is complete, Seo describes calling “the repartitioning function,” which “balances the L values of the cores” by moving “the earliest unfinished deadline task in C_{max} to the C_{src} .” *Id.* Thus, Seo’s dynamic core scaling algorithm also relates to balancing load among programmable logic-based processing units, as recited in [2.0].

125. Moreover, as explained in Section V.A.3, there is another aspect of Seo's task assignment algorithm called dynamic repartitioning. EX1006, 2. Dynamic repartitioning "tries to keep the *performance demands of each core* balanced by migrating tasks from the core with the *highest demand* to the one with the *lowest demand*." *Id.* Thus, the dynamic repartitioning portion of Seo's algorithm relates to balancing load among programmable logic-based processing units, as recited in [2.0].

iii. Claim 3

126. Claim 3 of the '948 Patent recites:

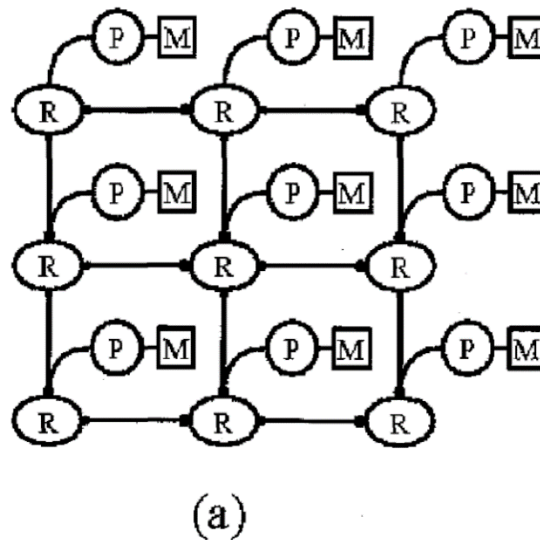
3. The method of claim 1, wherein each of the first set of programmable logic-based processing units comprises a field programmable gate array (FPGA).

127. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

128. Specifically, Nollet describes that its "apparatus comprises a plurality of resources, more in particular at least of one programmable processor and a reconfigurable hardware device." EX1004, [0056]. The "reconfigurable hardware device can be a logic gate array, e.g. a Field Programmable Gate Array (FPGA)." EX1004, [0057]. Nollet describes that the "the operating system is aware of the tile usage at any moment," and that, as a result, "it can spawn a new task without placement overhead by replacing the tile content through partial reconfiguration of

the FPGA.” EX1004, [0065]. “For example, the OS can keep track of which tile of an FPGA is being used to execute a task.” EX1004, [0086].

129. As shown in FIG. 7(a), “[e]mbodiments of the invention are implemented as a whole network with all the processors on the same FPGA.” EX1004, [0104].



Nollet (EX1004), FIG. 7(a)

130. In other words, each of the first set of programmable logic-based processing units comprises a field programmable gate array (FPGA), because a portion of a field programmable gate array is itself a field programmable gate array.

iv. Claim 4

131. Claim 4 of the '948 Patent recites:

4. The method of claim 1, wherein each of the second set of programmable logic-based processing units comprises a field programmable gate array (FPGA).

132. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

133. As I described above with respect to [1.7], a POSITA would have found it obvious to configure Nollet's reconfigurable hardware to execute multiple tasks from multiple program instances at once. Because both the first and second multi-stage programming instances (e.g., a movie and 3D) would have operated on first and second subsets of the tiles of Nollet's reconfigurable hardware, claim 4 is obvious for the same reasons as claim 3.

v. Claim 5

134. Claim 5 of the '948 Patent recites:

5. The method of claim 1, wherein the first characteristic or the second characteristic is load or utilization.

135. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

136. As I discussed with respect to [1.7], Seo defines the equation for the **dynamic utilization** L_n (i.e., load) of a given core C_n in equation (9). EX1006, 5. Load L_n is based on the most recent actual execution time cc_i of all finished tasks and the Worst Case Execution Time (WCET) w_i of all unfinished tasks, where w_i

“is defined as the maximum execution time required to complete [task] τ_i at the highest possible processor frequency.” *Id.*, 4-5. Thus, in order to determine the load L_n of a given core C_n , Seo’s algorithm must monitor at least two different characteristics for each core: the most recent actual execution time cc_i of all finished tasks assigned to that core and the Worst Case Execution Time (WCET) w_i of all unfinished tasks assigned to that core. *See id.* Thus, the first and second characteristics are constituent parts of ***dynamic utilization*** L_n (i.e., load) of a given core C_n . *See id.*

vi. Claim 6

137. Claim 6 of the ’948 Patent recites:

6. A non-transitory computer program product comprising instructions configured to perform a method parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program different than the first program, the method comprising:

- forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first multi-stage program instance is configured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

- forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance, wherein the second multi-stage program instance is configured to accelerate the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

- in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and

- in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

138. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

[6.0] A non-transitory computer program product comprising instructions configured to perform a method parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program different than the first program, the method comprising:

139. Nollet explains that the “operating system software can be supplied as a computer program product which includes code segments which when executed on a computing device execute methods in accordance with one aspect of the invention.” EX1004, [0088]. Moreover, as I discussed above with respect to [1.0], the Nollet-Tuan-Seo combination discloses a method on a parallel data processing platform comprising a plurality of programmable logic-based processing units for accelerating a service comprising at least a first program and a second program, different than the first program.

[6.1] forming a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance,

140. For the same reasons as I discussed above with respect to [1.1], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.2] wherein the first multi-stage program instance is configured to accelerate the first program, and

141. For the same reasons as I discussed above with respect to [1.2], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.3] wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

142. For the same reasons as I discussed above with respect to [1.3], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.4] forming a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance,

143. For the same reasons as I discussed above with respect to [1.4], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.5] wherein the second multi-stage program instance is configured to accelerate the second program, and

144. For the same reasons as I discussed above with respect to [1.5], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.6] wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

145. For the same reasons as I discussed above with respect to [1.6], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.7] in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, based on an adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units; and

146. For the same reasons as I discussed above with respect to [1.7], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[6.8] in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic different from the first characteristic, corresponding to the second set of programmable logic-based processing units, based on the adaptive optimized resource allocation policy, adding at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

147. For the same reasons as I discussed above with respect to [1.8], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

vii. Claim 7

148. Claim 7 of the '948 Patent recites:

7. The non-transitory computer program product of claim 6, wherein the allocation policy relates to balancing load among programmable logic-based processing units.
--

149. For the same reasons as I discussed above with respect to claim 2, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

viii. Claim 8

150. Claim 8 of the '948 Patent recites:

8. The non-transitory computer program product of claim 6, wherein each of the first set of programmable logic-based processing units comprises a field programmable gate array (FPGA).
--

151. For the same reasons as I discussed above with respect to claim 3, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

ix. Claim 9

152. Claim 9 of the '948 Patent recites:

9. The non-transitory computer program product of claim **6**, wherein each of the second set of programmable logic-based processing units comprises a field programmable gate array (FPGA).

153. For the same reasons as I discussed above with respect to claim 4, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

x. Claim 10

154. Claim 10 of the '948 Patent recites:

10. The non-transitory computer program product of claim **6**, wherein the first characteristic or the second characteristic is load or utilization.

155. For the same reasons as I discussed above with respect to claim 5, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

xi. Claim 11

156. Claim 11 of the '948 Patent recites:

11. A system comprising:

- a plurality of programmable logic-based processing units, each processing unit being configurable in at least a first configuration and a second configuration, different from the first configuration;
- a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance, wherein the first-multi-stage program instance is configured to accelerate the first program, and wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;
- a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance configured to execute the second program, and wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects; and
- a hardware logic and/or one or more processors executing computer-executable instructions configured to
 - (1) add, based on an allocation policy, in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, at least one more programmable logic-based processing unit to the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units, and
 - (2) add, based on the allocation policy, in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic, different from the first characteristic, corresponding to the second set of programmable logic-based processing units, at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

157. It is my opinion that the combination of Nollet, Tuan, and Seo renders this claim obvious.

[11.0] A system comprising:

158. As I discussed above with respect to [1.0], the Nollet-Tuan-Seo combination discloses a parallel data processing platform, which is a system. In fact, Nollet is titled “***System*** and Method for Hardware-Software Multitasking on a Reconfigurable Computing Platform,” and describes that an aspect of its “invention [is] to provide means to (re)schedule a task either in hardware or software in a reconfigurable ***system***-on-chip, provided with a uniform communication scheme, more in particular tools incorporated within a suitable object-oriented design environment permitting the implementation and management of hardware/software relocatable tasks.” EX1004, (54), [0018].

[11.1] a plurality of programmable logic-based processing units, each processing unit being configurable in at least a first configuration and a second configuration, different from the first configuration;

159. For the same reasons as I discussed above for [1.1], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.2] a first set of inter-task communication paths connecting a first set of programmable logic-based processing units of the plurality of programmable logic-based processing units into a first multi-stage program instance,

160. For the same reasons as I discussed above for [1.1], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.3] wherein the first-multi-stage program instance is configured to accelerate the first program, and

161. For the same reasons as I discussed above for [1.2], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.4] wherein the first set of programmable logic-based processing units are interconnected using reconfigurable cross-connects;

162. For the same reasons as I discussed above for [1.3], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.5] a second set of inter-task communication paths connecting a second set of programmable logic-based processing units of the plurality of programmable logic-based processing units to a second multi-stage program instance configured to execute the second program, and

163. For the same reasons as I discussed above for [1.4] and [1.5], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.6] wherein the second set of programmable logic-based processing units are interconnected using reconfigurable cross-connects; and

164. For the same reasons as I discussed above for [1.6], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.7] a hardware logic and/or one or more processors executing computer-executable instructions configured to

165. For the same reasons as I discussed above for [6.0], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.8] (1) add, based on an allocation policy, in response to a first increased demand for the first multi-stage program instance determined by monitoring a first characteristic corresponding to the first set of programmable logic-based processing units, at least one more programmable logic-based processing unit to

the first set of programmable logic-based processing units from the plurality of programmable logic-based processing units, and

166. For the same reasons as I discussed above for [1.7], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

[11.9] (2) add, based on the allocation policy, in response to a second increased demand for the second multi-stage program instance determined by monitoring a second characteristic, different from the first characteristic, corresponding to the second set of programmable logic-based processing units, at least one more programmable logic-based processing unit to the second set of programmable logic-based processing units from the plurality of programmable logic-based processing units.

167. For the same reasons as I discussed above for [1.8], the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

xii. Claim 12

168. Claim 12 of the '948 Patent recites:

12. The system of claim 11, wherein the allocation policy relates to balancing load among programmable logic-based processing units.

169. For the same reasons as I discussed above with respect to claim 2, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

xiii. Claim 13

170. Claim 13 of the '948 Patent recites:

13. The system of claim 11, wherein each of the first set of programmable logic-based processing units comprises a field programmable gate array (FPGA).

171. For the same reasons as I discussed above with respect to claim 3, the

Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

xiv. Claim 14

172. Claim 14 of the '948 Patent recites:

14. The system of claim 11 , wherein each of the second set of programmable logic-based processing units comprises a field programmable gate array (FPGA).
--

173. For the same reasons as I discussed above with respect to claim 4, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

xv. Claim 15

174. Claim 15 of the '948 Patent recites:

15. The system of claim 11 , wherein the first characteristic or the second characteristic is load or utilization.
--

175. For the same reasons as I discussed above with respect to claim 5, the Nollet-Tuan-Seo combination discloses this element or at least renders it obvious.

ADDITIONAL REMARKS

176. I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

177. I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Dated: 4/17/23

By: Henry H. Houh.
Henry H. Houh, Ph.D.

APPENDIX A

Henry H. Houh

15 Ballard Ter.
Lexington, MA 02420
henry.houh@gmail.com
cell: 781-325-8273

Education

Massachusetts Institute of Technology, Cambridge, MA

- PhD in Electrical Engineering and Computer Science, February 1998. “Designing Networks for Tomorrow's Traffic,” thesis supervised by Professor David Tennenhouse and Professor John Guttag. GPA 4.7/5.0
- Master of Science in Electrical Engineering and Computer Science, February 1991. “Demonstration of a laser repetition rate multiplier,” thesis. GPA 4.5/5.0
- Bachelor of Science in Electrical Engineering and Computer Science, June 1989. “Boundary element analysis of arbitrarily shaped dielectric structures,” thesis. GPA 4.7/5.0
- Bachelor of Science in Physics, February 1990. GPA 4.7/5.0

MITx, Cambridge, MA

- September 2020 – present: Enrolled in Statistics and Data Science MicroMasters program, a sequence of five MIT-offered graduate-level courses in statistics and machine learning, culminating in a MicroMasters certificate, expected 2022. Have completed four of five required courses.

Experience

BlocksCAD Inc. (H3XL spin-out)

- 2014 - present: Founder and member of Board of Directors. BlocksCAD is a visual Computer-Aided Design (CAD) program, designed to teach children mathematics and geometric visualization. BlocksCAD is being used in U.S. schools and worldwide in maker spaces, with 350,000+ users. BlocksCAD was selected to be part of the LearnLaunch 2017 Boost Accelerator, the MIT Play Labs 2018 Summer Accelerator, and the MassChallenge 2018 Boston Cohort. Recipient of DARPA DSO grant in 2014, and SBIR grant from the USDA in 2019.

H3XL Inc. d/b/a Einstein's Workshop (formerly Lexington Robotics)

- 2009 - present: Founder and President. Started local league providing science and engineering education programs based on LEGO Mindstorms, LEGO WeDo, and FIRST LEGO League. In 2012, grew program into a full science, technology, engineering and math enrichment program and creative/maker space, in 7,000 square feet of space. Serve 2,000+ kids and families annually. As of end of 2021, have delivered an estimated 220,000 student-instruction-hours of

STEM courses. Principle Investigator for 2-year DARPA grant to improve 3D Computer Aided Design tool developed in-house for the purposes of teaching 8 year olds and up 3D CAD.

Houh Consulting Inc. / Independent Consultant

- 2009 - present: Technical consultant specializing in Social Networking, Web 2.0, Web Site Development, Data Networking, Optical Networking, Telecommunications, Media Streaming and Voice Over IP. Clients include: Akin Gump Strauss Hauer & Feld LLP, Arnold & Porter Kaye Scholer LLP, BBN, Baker Botts LLP, Barnes & Thornburg LLP, BBN Technologies, Beus Gilbert McGroder PLLC, Bracewell LLP, Christensen O'Connor Johnson Kindness PLLC, Cooley LLP, Covington & Burling LLP, Cozen O'Connor P.C., DLA Piper, Davis Polk & Wardwell LLP, Desmarais LLP, Duane Morris LLP, Erise IP, P.A., Faegre Baker Daniels LLP, Finnegan, Henderson, Farabow, Garrett & Dunner, LLP, Fish & Richardson P.C., Goldman Ismail Tomaselli Brennan & Baum LLP, Haynes and Boone, LLP, Kaye Scholer, Kellogg Huber Hanson Todd Evans & Figel PLLC, Kirkland & Ellis LLP, Klarquist Sparkman, LLP, McCarthy Tetrault LLP, McGuireWoods LLP, Morgan, Lewis & Bockius LLP, Munger Tolles & Olson LLP, Orrick, Herrington & Sutcliffe LLP, O'Melveny & Myers LLP, Orrick, Herrington & Sutcliffe LLP, Paul Hastings LLP, Paul, Weiss, Rifkind, Wharton & Garrison LLP, Perkins Coie LLP, Quinn Emanuel Urquhart & Sullivan, LLP, Sidley Austin LLP, Smith Baluch LLP, Sterne, Kessler, Goldstein & Fox PLLC, Troutman Sanders LLP, Venable LLP, Weil, Gotshal & Manges LLP, Wiley Rein LLP, Wilmer Cutler Pickering Hale and Dorr LLP, and Winston & Strawn LLP.

Eons

- 2008 - 2009: Chief Technology Officer. Created product that Eons acquired from BBN Technologies. Integrated BBN product with Eons social networking platform and significantly increased the Eons group creation rate. Helped evaluate advertisement platform offerings and rolled out the "Boom Network" advertising network. Eons raised \$32 million from General Catalyst Partners, Charles River Ventures, Sequoia Capital, and Intel Capital.

BBN Technologies

- 2007 - 2008: Delta Division, Vice President of Technology. Grew "Boomerang" counter-sniper project engineering team and significantly de-risked \$10 million worth of product deliveries. Boomerang was a significant asset leading to the acquisition of BBN by Raytheon in 2009. Created new business plan and grew team; launched new fully-featured social networking web site in 5 months. Served as lead expert witness in patent infringement lawsuit, resulting in \$58 million jury award to client; verdict for patents I testified on were upheld on appeal which resulted in a \$120 million settlement.
- 2004 - 2007: Delta Division, Director of Technology, responsible for commercializing IP and creating new businesses. Hired and grew division's initial engineering team. Wrote three business plans, two of which are funded and active. For call center business plan, acted as general manager, hiring and managing engineering team, inside sales team, and identifying and recruiting a new general manager. Identified and recruited other key employees to Delta Division, including senior members of team leading to successful internal sales growth and spin-outs of projects. Contracted by BBN to BBN spin-out PodZinger as VP of Operations and

Technology. Identified sales team for counter-sniper system, leading to \$10 million dollars of new sales within 6 months and \$100+ million in additional orders in the following two years.

PodZinger Inc. (BBN spin-out, also known as EveryZing and now RAMP)

- 2006: Vice President of Operations and Technology. Significantly upgraded capability of consumer-facing search site and redeployed web site from company to co-location facility. Identified key portions of infrastructure for upgrading and cost reduction. Helped write business plan, evaluating advertisement models of revenue. Hired in operations replacement and phased back to BBN.

Commonwealth Capital

- 2004: Entrepreneur-in-residence (informally), performed technical due diligence on business plans, brainstormed ideas for new businesses with venture partner. With venture partner, left for portfolio company BBN to form core of commercialization team.

Empirix, Inc./Teradyne, Inc.

- 2001 - 2004: Chief Technologist, Engineering Manager, Web Application Test Group. Researched potential new product areas; developed product plan and prototype. Responsible for three new and existing products. Managed off-shore development team. Chief architect for all web testing products. Re-architected core testing product, helped write javascript interpreter. Provided technical vision for core product.
- 2000 - 2001: Chief Technologist, Communication Infrastructure Test Group. Responsible for incorporating new technology internally, tracking new technologies, technical evaluation of partnerships and potential acquisitions. Helped develop division strategy. Developed plans which formed core capabilities for successful new products introduced in 2004-5.
- 2000 - 2001: Engineering Manager, Communications Infrastructure Test Group. Execution of new product plan developed in prior role. Grew team from four existing engineers to team of over 30 on immediate team and over 40 on project. Delivered new platform in one year. Platform and derivatives accounted for large portion of booked products for the division within 2 years and is currently (2008) a key portion of new product offerings.
- 2000: Empirix was formed as a spin-out of Teradyne in January 2000. Reported to CEO in carve-out of Empirix from Teradyne.
- 1999: (Teradyne) Director of Business Development, Software Test Units. Reported directly to Chairman of the Board/Founder and then to general manager of software test unit (6 divisions of Teradyne). Evaluated and researched acquisition and partnership candidates. Internally assessed technology position in market and gaps in product lines. Worked with senior division staff to develop new product strategies. Attended internal Teradyne divisional board meetings. Chairman served as my mentor.

3Com Corporation/NBX Corporation

- 1999: Software Engineer 5. Continued work after 3Com acquisition of NBX. Built cross-division relationships for new products and research directions. NBX was acquired by 3Com in March 1999.

- 1997 - 1999: (NBX) Senior Scientist and Engineer. Work in IP Telephony. Architected next-generation product. Protocol design and validation for core protocol now used tens of millions of times daily. Led team in integration of IP protocols into current product. Designed audio reconstruction algorithms. Developed applications for bug analysis and diagnosis of system problems. Implementation of network simulator. Work on collaborative projects with external partners. Worked to identify gaps in product. Representative at numerous trade shows. Innovated novel methods of using product.

MIT Department of Electrical Engineering and Computer Science, Cambridge, MA

- 1991 - 1998: Research assistant, Telemedia Network Systems Group. Design, development, and implementation of Gigabit ATM network for distributed multimedia system. Studied host interface design issues. ATM network simulation.
- Spring 1989, Fall 1990, Spring 1995: Teaching assistant, Computation Structures digital systems course. (Spring 1995 Head TA)
- 1988 - 1989: Head laboratory teaching assistant for Computation Structures. Responsibilities included writing and revising lab assignments, and maintaining the lab.
- 1987: Laboratory teaching assistant for Computation Structures.
- 1987: Design, construction, and programming of 16-bit computer.

Agora Technology Group, Incorporated

- 1994 - 1996: Founder and CEO. Conceived and oversaw development of targeted advertising-supported Web sites. Responsible for company's vision and direction. Attended the first two WWW Conferences, presenting a workshop and paper at the first, and appearing on the "Commercialization and Economics of the Web" panel and chairing the "Where Commercial Services and the Web Are Headed" panel at the second. Sold company intact; is currently an operating stand-alone company.

AT&T Bell Laboratories, Holmdel, NJ

- 1989 - 1990: Implementation of cascadable all-optical fiber logic gate. Modelocking of all-fiber erbium laser. Construction of modelocked laser repetition rate booster. Strong optics laboratory and fiber optic experience.
- Summer 1988, 1987: Research in integrated optics. Analysis of rectangular waveguides using microwave modeling. Fabrication of integrated optical components.

Honors

- MIT Alumni Association Great Dome Award, 2010, Baker 60th Anniversary Reunion Co-Chair (highest group award given by MIT Alumni Association)
- MIT Alumni Association Presidential Citation Award (now known as Great Dome), 2008, Member of MIT Chairman's Salon committee
- MIT Alumni Association Bronze Beaver Award, 2007 (highest individual award given by MIT Alumni Association)
- MIT Alumni Association Volunteer Honor Roll, February 2004
- MIT Alumni Association Lobdell Award, 1999

- Boston Museum of Science Gold Pin for 1000 hours of Volunteer Service, April 1999
- MIT Alumni Association Presidential Citation Award (now known as Great Dome), 1997, Member of Alumni Online Communications Committee

Patents and Patent Publications

- U.S. Patent #9,697,231, H. Houh, J. Stern, R. Spina, M. Meteer, “Methods and apparatus for providing virtual media channels based on media search,” July 4, 2017.
- U.S. Patent #9,697,230, H. Houh, J. Stern, “Methods and apparatus for dynamic presentation of advertising, factual, and informational content using enhanced metadata in search-driven media applications,” July 4, 2017. See also WO2007056485.
- U.S. Patent #7,975,296, L. Apfelbaum, H. Houh, T. Mayberry and G. Friedman, “Automated security threat testing of web pages,” July 5, 2011. See also US20030159063, WO2003067405.
- U.S. Patent #7,877,736, H. Houh and J. N. Stern, “Computer language interpretation and optimization for server testing,” January 25, 2011. See also US20050138104, WO2005043300.
- U.S. Patent #7,801,910, H. Houh and J. N. Stern, “Method and apparatus for timed tagging of media content,” September 21, 2010. See also US20070112837, US20090222442, WO2007056535.
- U.S. Patent #7,590,542, D. C. Williams, W. C. Hand, H. Houh, A. R. Seeley, “Method of Generating Test Scripts Using a Voice-Capable Markup Language,” September 15, 2009. See also EP1530869, US20030212561, WO2003096663.
- U.S. Patent #6,967,963, H. H. Houh, P. Anderson, C. Gadda, “Telecommunication method for ensuring on-time delivery of packets containing time-sensitive data,” November 22, 2005. See also EP1060400, WO2000033092, CA2318774.
- U.S. Patent #5,144,375, M. C. Gabriel, H. H. Houh, N. A. Whitaker, “Sagnac Optical Logic Gate,” September 1, 1992. Also issued as European Patent # EP0456422, July 23, 1997, German Patent #DE69126913, August 28, 1997
- U.S. Patent Application Publication No. 20020015387, “Voice Traffic Packet Capture and Analysis Tool for a Data Network” (Abandoned in 2007)
- U.S. Patent Application Publication No. 20020016708, “Method and Apparatus for Utilizing a Network Processor as Part of a Test System” (Abandoned in 2007)
- U.S. Patent Application Publication No. 20020016937, “Method and Apparatus for Utilizing a Network Processor as Part of a Test System.” (Abandoned in 2007) See also WO2002011413.
- U.S. Patent Application Publication No. 20070106646, “User-directed navigation of multimedia search results” (Abandoned in 2009)
- U.S. Patent Application Publication No. 20070106660, “Method and apparatus for using confidence scores of enhanced metadata in search-driven media applications” (Abandoned in 2009)
- U.S. Patent Application Publication No. 20070106685, “Method and apparatus for updating speech recognition databases and reindexing audio and video content using the same” (Abandoned in 2015)
- U.S. Patent Application Publication No. 20070118873, “Methods and apparatus for merging media content” (Abandoned in 2015)
- U.S. Patent Application Publication No. 20090222442, “User-directed navigation of multimedia search results” (Abandoned in 2016)

- U.S. Patent Application 11/395,732, “Search snippet creation for audio and video data” (Abandoned in 2009)
- U.S. Patent Application 11/774,931, “Methods and apparatus for managing a social networking web site”
- U.S. Patent Application 11/774,947, “Methods and apparatus for organizing media files”
- U.S. Patent Application 11/774,956, “Methods and apparatus for managing an online event”
- U.S. Provisional Patent Application 61/086,909, “Measuring and ranking relationship activity”
- U.S. Provisional Patent Application 61/086,914, “Detecting media object commonality”
- U.S. Provisional Patent Application 61/086,904, “Message categorization based on message characteristics”
- U.S. Provisional Patent Application 61/086,905, “Photo tagging to request action”

Trials, Testimony and Depositions

- Case No. 1:06CV682 (CMH/BRP), Verizon vs. Vonage, U.S. District Court for the Eastern District of Virginia, filed expert report, was deposed and testified at trial.
- Case No. 1:08CV157 (CMH/TRJ), Verizon vs. Cox, U.S. District Court for the Eastern District of Virginia, was deposed and testified at trial.
- Case No. 5:09-cv-476, Two-Way Media vs. AT&T, U.S. District Court for the Western District of Texas, filed expert report, testified at trial.
- Case No. 2:10-cv-248 (RAJ/FBS), ActiveVideo Networks vs. Verizon, U.S. District Court for the Eastern District of Virginia, filed expert report and was deposed as an expert witness.
- Case No. 1:11-cv-00880-TSE-JFA, Bear Creek Technologies, Inc. vs. Verizon Services Corp., et al, U.S. District Court for the Eastern District of Virginia, was deposed as an expert witness.
- Case No. 3:10-CV-298-BBC, AlmondNet, Inc. vs. Microsoft Corp., U.S. District Court for the Western District of Wisconsin, filed expert report.
- Case No. 6:10-cv-00597, Guardian Media Technologies, Ltd. Vs. AT&T Operations, Inc. et al., U.S. District Court for the Eastern District of Texas, Tyler Division, filed expert report.
- Case No. ESCV2010-02282C, The Octopus Solution LLC v. Gary Brown et al., Essex, MA Superior Court, testified at trial.
- Investigation No. 337-TA-882, In the matter of Certain digital media devices, including televisions, Blu-ray disc players, home theater systems, tablets and mobile phones, components thereof and associated software, U.S. International Trade Commission, filed expert reports, was deposed and testified at hearing.
- Investigation No. 337-TA-995, In the matter of Certain communications or computing devices, and components thereof, U.S. International Trade Commission, filed expert reports, and was deposed.
- Case No. 8:12-cv-122-LES-TDT, Prism Technologies LLC v. AT&T Mobility LLC, U.S. District Court for the District of Nebraska, filed expert report, was deposed and testified at trial.
- Case No. 11-2684 (JWL), Sprint Communications Co., L.P., v. Comcast Cable Communications, LLC, et al., U.S. District Court for the District of Kansas, filed expert report and was deposed.
- Case No. 11-2686 (JWL), Sprint Communications Co., L.P., v. Time Warner Cable, Inc., et. al., LLC, et al., U.S. District Court for the District of Kansas, filed expert report, was deposed, and testified at trial.
- Case No. 12-487 (SLR), Cox Communications, Inc., et al., v. Sprint Communications Company L.P., et al., U.S. District Court for the District of Delaware, filed expert report and was deposed.

- Case IPR 2014-00039, Microsoft Corporation v. B.E. Technology LLC, U.S. Patent No. 6,628,314. Submitted declaration and cross-examined in deposition.
- Case IPR 2014-00086, Apple, Inc. v. Evolutionary Intelligence, LLC, U.S. Patent No. 7,010,536. Submitted declaration and cross-examined in deposition.
- Case No. 6:11-CV-421, Stragent, LLC v. Intel Corporation, U.S. District Court for the Eastern District of Texas, Tyler Division, deposited and testified at trial as a fact witness.
- IPR2014-01366, IPR2014-01367, IPR2014-01368, Samsung et al. v. Straight Path, submitted IPR declarations and was deposited.
- IPR2015-1006, IPR2015-1007, Cisco v. Straight Path, submitted IPR declarations.
- IPR2014-01457, IPR2014-01459, Microsoft v. Biscotti, submitted IPR declaration and was deposited.
- IPR2016-00302, Apple v. Nonend, submitted IPR declaration.
- IPR2014-00812, Yelp and Twitter v. Evolutionary Intelligence, submitted IPR declaration.
- IPR2015-00307, Cisco v. AIP, submitted IPR declaration.
- Microsoft v. Acacia, submitted IPR declaration.
- IPR2015-00342, AT&T Mobility LLC and Cellco Partnership d/b/a Verizon Wireless v. Solocron, submitted IPR declaration.
- Mediatube v. Bell Canada, Canadian Federal Court, submitted expert report and testified at trial.
- CBM2016-00036, Google v. At Home Bondholder's Liquidating Trust, submitted CBM declaration.
- IPR2016-01198, IPR2016-01201, Apple, Inc. v. VoIP-Pal.com, Inc., submitted IPR declaration and was deposited.
- Case No. 3:14-cv-2824-VC, Trend Micro Incorporated v. Rpost Holdings, Inc., et al, submitted expert report and was deposited
- Akamai v. Limelight Networks, IPR2016-01631, IPR2016-01711, IPR2017-00249, IPR2017-00349, submitted IPR declarations and was deposited
- Hyperlync Technologies v. Verizon Sourcing, No. 650151/2015 New York Supreme Court, submitted declaration and was deposited
- Cascades Streaming Technologies v. Big Ten Networks, submitted expert reports and was deposited
- Klaustech, Inc. v. Google LLC., Case No. 4:10-CV-05899-JSW (N.D. Cal.), submitted expert report and was deposited
- Cisco v. Uniloc, IPR2017-00198, IPR2017-00058, submitted IPR declarations and was deposited.
- Cisco v. Uniloc, IPR2017-00597, submitted IPR declaration and was deposited
- Twitter v. Youtoo Technologies, LLC, IPR2017-01133, IPR2017-01131, submitted IPR declarations and was deposited
- Taser International Inc. (now Axon Enterprise, Inc.) v. Digital Ally Inc., IPR2017-00375, submitted IPR declaration and was deposited
- Taser International Inc. (now Axon Enterprise, Inc.) v. Digital Ally Inc., IPR2017-00376, IPR2017-00515, IPR2017-00775, submitted IPR declarations
- HTC Corporation v. IPA Technologies, IPR2018-00306, IPR2018-00307, submitted IPR declaration
- Hulu v. Soundview, IPR2018-00864, IPR2018-01023, submitted IPR declarations and was deposited
- XpertUniverse v. Cisco, 3:17-cv-3848, U.S. District Court for Northern District of California, submitted declaration in support of claim construction and was deposited
- Improved Search v. Microsoft, C.A. No. 16-cv-650-JFB-SRF (Del.), submitted declaration in support of claim construction

- Apple v. Uniloc, IPR2018-00361, IPR2018-00394, IPR2018-00395, IPR2019-00700, IPR2019-00701, IPR2019-01667, submitted IPR declarations
- LG Electronics v. Uniloc, submitted IPR declaration
- SRC Labs and Saint Regis Mohawk Tribe v. Microsoft, 2:18-cv-00321-JLR, U.S. District Court for Western District of Washington at Seattle, submitted declaration in support of claim construction and was deposed
- Verizon Wireless v. Barkan Wireless IP Holdings LP, submitted IPR declarations
- Verizon Wireless and Samsung Electronics America Inc. v. Barkan Wireless IP Holdings LP, IPR2019-199, IPR2019-200, IPR2019-234, IPR2019-631, IPR2019-632, submitted IPR declarations
- Samsung Electronics America Inc. v. Barkan Wireless IP Holdings LP, IPR2019-100, submitted IPR declaration
- Cisco Systems Inc. v. Meetrix IP LLC, IPR2019-539, IPR2019-540, IPR2019-541, IPR2019-542, IPR2019-543, IPR2019-544, submitted IPR declarations
- GMG Products LLC v. Traeger Pellet Grills LLC, PGR2019-24, PGR2019-34, PGR2019-35, PGR2019-36, submitted PGR declaration and was deposed
- Uniloc 2017 LLC v. Google LLC, 2:18-cv-00502-JRG-RSP, U.S. District Court for the Eastern District of Texas, Marshall Division, submitted declaration in support of claim construction
- Microsoft Corporation v. Uniloc 2017 LLC, IPR2020-00102, IPR2020-00103, submitted IPR declarations
- Microsoft Corporation and HP Inc. v. Synkloud Technologies, LLC, IPR2020-00316, IPR2020-01031, IPR2020-01032, IPR2020-01269, IPR2020-01270, IPR2020-01271, submitted IPR declarations
- Seven Networks, LLC v. Apple Inc, 2:19-cv-115-JRG, U.S. District Court for the Eastern District of Texas, Marshall Division, submitted declaration in support of claim construction
- Unified Patents Inc. v. Ortiz & Associates Consulting, LLC, IPR2019-00743, submitted IPR declaration
- Uniloc 2017 LLC v. Apple Inc., 3:19-cv-01905-JD, U.S. District Court for the Northern District of California, San Francisco Division, submitted declaration in support of claim construction and was deposed
- ClearPlay Inc. v. Dish Network LLC et al.; 2:14-cv-00191, U.S. District Court for the District of Utah, Central Division, submitted expert report and re-examination declarations
- AMC Networks, Inc. et al v. Sound View Innovations, LLC, IPR2020-00482, submitted IPR declaration
- Case No. 1:17-cv-01734-RGA, Sprint Communications Co., L.P., v. Charter Communications, Inc., et al., U.S. District Court for the District of Delaware, filed expert report and was deposed.
- CommScope Technologies LLC et al. v. Barkan Wireless IP Holdings LP et al., IPR2020-00827, IPR2020-00829, IPR2020-00831, IPR2020-00833, IPR2020-00835, IPR2020-00838, submitted IPR declarations
- Cisco Systems Inc. v. Monarch Networking Solutions LLC, IPR2020-01226, submitted IPR declaration
- CoolTV Network.com v. Blackboard, Inc., Facebook, Inc., International Business Machines Corporation, Kaltura, Inc., Limelight Networks, Inc., Microsoft Corporation, Ooyala, Inc., Snap, Inc., and Trapelo Corp., submitted declaration in support of claim construction and was deposed
- Facebook, Inc. v. Onstream Media Corporation, IPR2020-01507, IPR2020-01508, IPR2020-01525, IPR2020-01527, IPR2020-01528, submitted IPR declarations
- Hisense Co LTD. v. Maxell, LTD, IPR2020-01598, IPR2020-01600, submitted IPR declarations
- Microsoft Corp. v. O'Brien, IPR2021-00015, submitted IPR declaration and was deposed

- Sable Networks Inc. v. Cisco Systems, Inc., WA:20-CV-00288-ADA, U.S. District Court for the Western District of Texas, Waco Division, submitted declaration in support of claim construction
- Peloton Interactive, Inc. v. Icon Health and Fitness Inc et al., IPR2021-00342, submitted IPR declaration
- Hulu, LLC v. SITO Mobile R&D IP, LLC et al., CBM2020-00028, IPR2020-00158, IPR2020-00206 , IPR2020-00219 , IPR2020-00265 , IPR2020-00298 , IPR2020-00304 , IPR2020-00308 , submitted CMB and IPR declarations and was deposited
- Twitter, Inc. and Google LLC v. B.E. Technology, LLC, IPR2021-00482, IPR2021-00483, IPR2021-00484, IPR2021-00485, submitted IPR declarations
- MG Freesites LTP v. ScorpCast, LLC, IPR2021-00511, submitted IPR declaration
- Avionte, LLC v. Sandeep Acharya, Samar Basnet, and Aqore LLC, Arbitration, Hon. Jeffrey Keyes (Ret.)
- Microsoft Corporation v. WSOU Investments, LLC d/b/a Brazos Licensing and Development, IPR2021-00751, submitted IPR declaration
- SAP America, Inc. v. Express Mobile, Inc., IPR2021-01144, IPR2021-01145, IPR2021-01146, submitted IPR declarations
- StratosAudio Inc. v. Hyundai Motor America, 6:20-cv-01125-ADA, U.S. District Court for the Western District of Texas, Waco Division, submitted claim construction declaration and was deposited
- Peloton Interactive, Inc. v. Ifit Health and Fitness Inc, IPR2022-00029, IPR2022-00030, submitted IPR declarations and was deposited
- Investigation No. 337-TA-1263, In the matter of Certain televisions, remote controls, and components thereof, U.S. International Trade Commission, filed expert reports and was deposited
- Alexander Stross v. Netease, Inc., et al., U.S. District Court (Central District of California), Case 2:20-cv-00861-AB(JC), filed expert reports and was deposited
- Ralph Ledergerber v. Netease, Inc., et al., U.S. District Court (Central District of California), Case 2:20-cv-00862-AB(JC), filed expert reports and was deposited
- John Walmsley v. Netease, Inc., et al., U.S. District Court (Central District of California), Case 2:20-cv-00863-AB(JC), filed expert reports and was deposited
- Bernhard Kuhmstedt v. Netease, Inc., et al., U.S. District Court (Central District of California), Case 2:20-cv-02044-AB(JC), filed expert reports and was deposited
- F45 Training Pty Ltd. v. Body Fit Training USA Inc., C.A. No. 20-1194 (LPS) (D. Del.), filed expert report

Publications

- “IP switching: server driven flow classification,” H. H. Houh, *Proceedings of the Washington University Workshop on Integration of IP and ATM* , November 1996.
- “Aurora at MIT,” D. D. Clark, H. H. Houh, and D. L. Tennenhouse, Editors, *MIT Laboratory for Computer Science Technical Report 673*, December 1995.
- “ViewStation Applications: Implications for Network Traffic,” C. J. Lindblad, D. Wetherall, W. Stasior, J. F. Adam, H. H. Houh, M. Ismert, D. Bacher, B. Phillips, and D. L. Tennenhouse, *IEEE Journal of Selected Areas in Communications*, 1995.
- “The VuNet Desk Area Network: Architecture, Implementation, and Experience,” H. H. Houh, J. F. Adam, M. Ismert, C. J. Lindblad, and D. L. Tennenhouse, *IEEE Journal of Selected Areas in Communications*, 13 (4), May, 1995.
- “Reducing the Complexity of ATM Host Interfaces,” H. H. Houh and D. L. Tennenhouse, *Hot Interconnects II Symposium Proceedings*, Stanford, August 11-12, 1994.

- “Media-intensive data communications in a ‘desk-area’ network,” J. F. Adam, H. H. Houh, M. Ismert, and D. L. Tennenhouse, IEEE Communications, August 1994.
- “ViewStation Applications: Intelligent Video Processing Over A Broadband Local Area Network,” C. J. Lindblad, D. J. Wetherall, W. Stasior, B. Phillips, D. Bacher, J. Adam, H. Houh, M. Ismert, and D. L. Tennenhouse, Proceedings of the 1994 USENIX Symposium on High-Speed Networking, Oakland, CA, August 1994.
- “The Media Gateway: Live Video on the World Wide Web,” H. H. Houh, C. J. Lindblad, J. Soo, D. L. Tennenhouse, and D. J. Wetherall, Workshop at the 1994 World Wide Web Conference, Geneva, Switzerland, May 1994.
- “Active Pages: Intelligent Nodes on the World Wide Web,” H. H. Houh, C. J. Lindblad, and D. J. Wetherall, Proceedings of the 1994 World Wide Web Conference, Geneva, Switzerland, May 1994.
- “Wavelength Division vs. Code Division Access Methods for Optical Networks,” H. H. Houh, Area Exam Paper, May 1993.
- “Experience with the VuNet: A Network Architecture for a Distributed Multimedia System,” J. F. Adam, H. H. Houh, D. L. Tennenhouse, Proceedings of the 18th Conference on Local Computer Networks, pp. 70-76, September 1993
- “The VudBoard: A Simple DMA Interface,” H. H. Houh, Proceedings of the 4th Gigabit Minijam, January 1994.
- “A Software-Oriented Approach to the Design of Media Processing Environments,” D. L. Tennenhouse, J. Adam, D. Carver, H. Houh, M. Ismert, C. Lindblad, W. Stasior, D. Weatherall, D. Bacher, and T. Chang., submitted to the International Conference on Multimedia Computing and Systems, May 1994.
- “A Network Architecture for Distributed Multimedia Systems,” J. F. Adam, H. H. Houh, M. Ismert, and D. L. Tennenhouse, submitted to the International Conference on Multimedia Computing and Systems, May 1994.
- “The Viewstation Collected Papers,” D. L. Tennenhouse, J. Adam, C. Compton, A. Duda, D. Gifford, H. Houh, M. Ismert, C. Lindblad, W. Stasior, R. Weiss, D. Wetherall, D. Bacher, D. Carver, and T. Chang, MIT Laboratory for Computer Science Technical Report, MIT/LCS/TR-590, November 1993.
- “A System's Perspective of the Sagnac Fiber Logic Gates and Their Possible Applications,” A. Huang, N. Whitaker, C. Gabriel, H. Avramopoulos, P. M. W. French, H. H. Houh, and I. Chuang, Applied Optics, September 10, 1994
- “Complete Switching in a Three-Terminal Sagnac Switch,” H. Avramopoulos, P. M. W. French, M. C. Gabriel, H. H. Houh, N. A. Whitaker, T. Morse, IEEE Phot. Tech. Lett. **3** (3), 235
- “Complete Switching in a Three-Terminal Sagnac Switch,” H. Avramopoulos, P. M. W. French, M. C. Gabriel, H. H. Houh, N. A. Whitaker, IEEE/LEOS Annual Meeting, Paper PDP-13, November 1990
- “All-optical phase-locked oscillator,” N. A. Whitaker, Jr., H. H. Houh, H. Avramopoulos, T. F. Morse, IEEE/LEOS Annual Meeting, Paper ELT2.4/MOO3, November 1990
- “Passive modelocking of an all-fiber erbium laser,” H. Avramopoulos, H. H. Houh, N. A. Whitaker, M. C. Gabriel, T. F. Morse, IEEE/LEOS Conference on Optical Amplifiers and their Applications, Paper PDP-8, August 1990
- “Transverse modes, waveguide dispersion, and 30ps recovery in submicron GaAs/AlAs microresonators,” J. L. Jewell, S. L. McCall, A. Scherer, H. H. Houh, N. A. Whitaker, A. C. Gossard, and J. H. English, Appl. Phys. Lett. **55** (1), July 3, 1989

Leadership, Activities and Interests

- Leadership
 - Discovery Museums (Acton, MA)
 - Science and Technology Advisory Council, 2012 - present
 - MIT Alumni Association Board of Directors
 - National Selection Committee, 3 year term, 2020 – 2023
 - K-12 STEM Initiatives Co-chair, 2013 – 2017
 - Awards Committee Chair, 2012 – 2014
 - Awards Committee, 3 year term, 2011 – 2014
 - Vice President, 2 year term, 2004 – 2006
 - Alumni Ad-Hoc Committee, Chair, 2005-2006
 - Board Member, 2 year term, 1997 – 1999
 - MIT Club of Boston
 - Board of Directors, 2006 - 2011
 - K-12 Initiatives Chair, 2009 - 2012
 - VP of Communications, MIT Club of Boston, 2003 - 2006
 - Past-President, MIT Club of Boston, 2002 - 2003
 - President, MIT Club of Boston, 2001 - 2002
 - President-Elect, MIT Club of Boston, 2000 - 2001
 - VP of Programs, MIT Club of Boston, 1999 - 2000
 - Activities Super-Chair, MIT Club of Boston, 1998 - 1999
 - MIT Enterprise Forum of Cambridge, Inc.
 - Past Chair, 2009 – 2011
 - In-NOW-vation Co-chair, 2010
 - Chair, 2007 - 2009
 - Vice Chair, 2005 - 2007
 - Executive Board Member, 2002 - 2011
 - Winter Workshop Co-Chair, February 2007 - conceived idea for conference, which sold-out and produced largest attendance numbers in recent memory
 - Spring Workshop Co-Chair, Spring 2004
 - Membership Committee Chair, Fall 2003 - 2006
 - 25th Anniversary Dinner Chair, Fall 2003
 - As Membership Chair and Board Member, started Special Interest Groups in 2004; a SIG won the MIT Presidential Citation award, the MIT Alumni Association's highest award for organizations, in 2006
 - Estabrook Elementary School PTA
 - Advisory committee to the superintendent on PCB issue, 2010-2011
 - 4th Grade after-school science program co-organizer, 2010-2012
 - 4th and 5th Grade before-school Math Olympiad co-organizer, 2009-2013
 - 5th Grade BBQ and Yearbook Committee, 2011, 2013
 - Family Math Night volunteer, 2008-2012
 - Tau Beta Pi National Engineering Honor Society
 - Advisor, MA B Chapter at MIT, 2003 - present
 - District Director (National Officer), Tau Beta Pi, New England Area, 1991 - 2003
 - President, MA B Chapter at MIT, Fall 1988 - Spring 1989
 - Laureate award, 1989

- MIT Class of 1989
 - Secretary, five consecutive 5 year terms, 1989 - 2014
 - 30-year Reunion Committee, 2019
 - 25-year Reunion Committee, 2014
 - 20-year Reunion Committee and Gift Committee, 2009
 - 15-year Reunion Committee and Gift Committee, 2004
 - 10-year Reunion Committee and Gift Committee, 1999
 - 5-year Reunion Committee, 1994
 - Interim Treasurer, 1993 - 1994
 - Instituted annual senior class career fair, now raising over \$100,000 annually for senior class activities, Fall 1988
- Strong, consistent record of leadership dating to high school
- Acting
 - '21' (Sony Pictures), credited as "Chinatown Dealer," 2007, Kevin Spacey's movie about the MIT Blackjack Team inspired by "Bringing Down the House" by Ben Mezrich, opened nationwide on March 28, 2008. 21 was the number one movie in US for two weeks and number one globally for one week. 21 also topped the DVD sales, Blu-ray sales and DVD rental charts.
 - Spring Lake Theater Company, first New York-area off-broadway production of "A Chorus Line," played role of Mark, Summer 1990
- Former member of the MIT Blackjack team
- Producer for 10,000 Maniacs' 2013 album "Music from the Motion Picture"
- Executive Producer for 10,000 Maniacs' 2015 album "Twice Told Tales"
- Violist, violist, harpist, guitarist, singer, actor: played in many amateur/semi-professional groups including Reading Symphony Orchestra, MIT Summer Philharmonic Orchestra, Freisinger Chamber Orchestra, Merrimack Valley Philharmonic, Longwood Symphony, MIT Symphony, and Somerville Community Chorus
- Violist, "Coming Home" album by Karen Phillips, 2014
- Google Scholar citations: 1914, h-index: 21, i10-index: 26 (since 2016: 649/13/13)
- Erdos-Bacon number of 7
- Erdos-Bacon-Sabbath number of 11