

SUPPLEMENTAL DECLARATION OF DR. STANLEY SHANFIELD

Re: U.S. Pat. No: 6,938,177

For Proceeding: IPR2023-00195

TABLE OF CONTENTS

I.	Materials Considered	3
A.	Original Materials Considered.....	3
B.	Additional Materials Considered.....	6
II.	Person of Ordinary Skill in the Art.....	9
III.	Summary of Opinions Regarding Claim Construction	16
A.	The Dispute	16
B.	Approach to Claim Construction	16
C.	Past Relevant Constructions and Arguments (Not Involving Xilinx).....	17
D.	Summary of Conclusions.....	18
IV.	Details of Supplemental Opinions.....	19
A.	Ground 1: The Prior Art Provides a “gate array ... configured to run independent processes in parallel with the processor.”	19
1.	The Prosecution History Focused on “Multiprocessing”, Not Clocking or Control Line Direction.....	22
2.	“in parallel”	25
3.	“process”	26
4.	“independent processes”	30
5.	The Challenged Claims 1 and 4 Are Obvious Under Any of these Definitions.....	44
B.	Ground 4: The Prior Art Provides the “Power Converter” Element	65
C.	Sentient’s Miscellaneous Arguments About Dehkordi	67
D.	Sentient’s Miscellaneous Arguments About Halverson, Kostarnov, and Other Art	69

I, Dr. Stanley Shanfield, declare that:

1. I submit this supplemental declaration to my original declaration EX1002 in the matter IPR2023-00195 in support of the Petitioner Xilinx, Inc. in this proceeding.

2. I am being compensated at my ordinary and customary consultant rate for my work. My compensation is in no way contingent on the outcome of this proceeding or any other proceeding. I have no other interest in this proceeding or any other proceeding relating to the '177 patent.

3. I set forth my supplemental opinions and the bases for my opinions below.

I. Materials Considered

4. I have considered the background knowledge/technologies that were commonly known to persons of ordinary skill in this field during the time before the alleged invention date for the '177 patent.

A. Original Materials Considered

5. I have read and considered the publications and materials including the following:

- U.S. Patent No. 6,938,177 (“’177 patent”) (EX1001)
- U.S. Patent No. 6,148,399 to James C. Lyke (“Lyke”) (EX1003)
- P. Dehkordi, T. Powell and D. Bouldin, *Development of a DSP/MCM subsystem assessing low-volume, low-cost MCM prototyping for universities*, Proceedings 1996 IEEE Multi-Chip Module Conference (Cat. No. 96CH35893), 1996, pp. 89-94 (“Dehkordi”) (EX1004)
- G. A. Frantz, K. Lin, J. B. Reimer and J. Bradley, “The Texas Instruments TMS320C25 Digital Signal Microcomputer,” in IEEE Micro, vol. 6, no. 6, pp. 10-28, Dec. 1986 (“Frantz”) (EX1005)

- J. Faura, C. Horton, P. van Duong, J. Madrenas, M. A. Aguirre and J. M. Inerser, *A novel mixed signal programmable device with on-chip microprocessor*, Proceedings of CICC 97 - Custom Integrated Circuits Conference, 1997, pp. 103-106 (“Faura”) (EX1006)
- U.S. Patent No. 5,663,734 to Norman F. Krasner (“Krasner”) (EX1007)
- U.S. Patent No. 4,279,020 to Ronald W. Christian et al (“Christian”) (EX1008)
- U.S. Patent No. 5,675,824 to Randy C. Steele (“Steele”) (EX1009)
- File Wrapper of the ’177 Patent (EX1010)
- Claim Construction Order in *Sentient Sensors v. Cypress Corp.*, 19-1868 (MN), (Jan. 28, 2021, D. Delaware) (D.I. 89) (EX1012)
- Data Sheet for Motorola 96002 Digital Signal Processor (EX1013)
- Data Sheet for Xilinx XC4000E and XC4000X Series (including Xilinx XC4010) (EX1014)
- K. Avery and D. Alexander, *Recent progress in space-based micro-controller design*, 16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings, 1997, pp. 2.1-7 (“Avery”) (EX1015)
- T. Goforth, S. Connon, J. Lyke, *Space Testing of the Advanced Instrument Controller*, Proceedings of the 13th AIAA/USU Conference on Small Satellites, Aug. 23–26, 1999, Logan UT, SSC99-XI-3 pp. 1-11 (“Goforth”) (EX1016)
- L. Muller, M. H. Hecht, L. M. Miller, H. K. Rockstad and J. C. Lyke, *Packaging and qualification of MEMS-based space systems*, Proceedings of Ninth International Workshop on Micro Electromechanical Systems, 1996, pp. 503-508 (“Muller”) (EX1017)
- Blemel, Kenneth G., *Reactive Prognostic Management of Corrosion With Autonomous Sensors*. Paper presented at the CORROSION 2000, Orlando, Florida, March 2000 (EX1018)
- Hauck, Scott, *The Roles of FPGAs in Reprogrammable Systems*, Proceedings of the IEEE, Vol. 86, No. 4, pp. 615-639, April 1998 (EX1019)
- U.S. Patent No. 5,867,535 to William C. Phillips (“Phillips”) (EX1023)
- U.S. Patent No. 6,178,494 to Steve M. Casselman (“Casselman”) (EX1024)
- U.S. Patent No. 5,574,930 to Richard P. Halverson (“Halverson”) (EX1025)
- U.S. Patent No. 5,872,960 to Oved Oz (“Oz”) (EX1026)

- T. Nishitani, I. Kuroda, Y. Kawakami, H. Tanaka and T. Nukiyama, *Advanced single-chip signal processor*, ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1986, pp. 409-412 ("Nishitani") (EX1027)
- P. R. Christensen, et al. (2001), *Mars Global Surveyor Thermal Emission Spectrometer experiment: Investigation description and surface science results*, J. Geophys. Res., 106 (E10), 23823–23871 ("Christensen") (EX1028)
- U.S. Patent No. 5,936,562 to Todd L. Brooks ("Brooks") (EX1029)
- U.S. Patent No. 5,861,831 to Franklin M. Murden ("Murden") (EX1030)
- U.S. Patent No. 5,638,092 to Tommy K. Eng ("Eng") (EX1031)
- J. Faura, C. Horton, P. van Duong, J. Madrenas, M. A. Aguirre and J. M. Inerser, *A novel mixed signal programmable device with on-chip microprocessor*, Proceedings of CICC 97 - Custom Integrated Circuits Conference, 1997, pp. 103-106 ("FauraMoreno") (EX1032)
- Y. Tian, J. Ying, X. Ye, W. Gu, *DDGIPS: a general image processing system in robot vision*; Proc. SPIE 4212, Reconfigurable Technology: FPGAs for Computing and Applications II, Oct. 6, 2000 ("Tian") (EX1033)
- U.S. Patent No. 5,874,834 to Bernard J. New ("New") (EX1034)
- Marchal, Pierre, *Field-programmable gate arrays*, Communications of the ACM, Vol. 42, No. 4 (1999), p. 57-59 ("Marchal") (EX1035)
- I. Kostarnov, S. Morley, J. Osmany, C. Solomon, *A reconfigurable approach to low cost media processing*. In: Luk, W., Cheung, P.Y.K., Glesner, M. (eds) Field-Programmable Logic and Applications. FPL 1997. Lecture Notes in Computer Science, vol 1304. Springer, Berlin, Heidelberg ("Kostarnov") (EX1036).
- U.S. Patent No. 5,757,690 to Scott H. R. McMahon ("McMahon") (EX1037)
- Analog Devices, ADuC824, MicroConverter, Dual-Channel 16-/24-Bit ADCs with Embedded FLASH MCU, 2001 (EX1038)
- Analog Devices, ADuC824 FAQ, January 2001 (EX1039)
- U.S. Patent No 5,892,962 to Jocelyn Cloutier ("Cloutier") (EX1040)
- U.S. Patent Publication No. to 2002/0129191 to Berham DaCosta ("DaCosta") (EX1041)

- de Lima, F.G., *Designing A Radiation Hardened 8051-like Micro-Controller*, Proceedings 13th Symposium on Integrated Circuits and Systems Design (Cat. No.PR00843), 18-24 September 2000 (“de Lima”) (EX1042)
- J. J. Wang et al., *SRAM based re-programmable FPGA for space applications*, in IEEE Transactions on Nuclear Science, vol. 46, no. 6, pp. 1728-1735, Dec. 1999 (EX1043)
- U.S. Patent No. 5,936,971 to Ellyahou Harari (“Harari”) (EX1044)
- U.S. Patent No. 5,261,109 to Sudarshan B. Cadambi (“Cadambi”) (EX1045)
- T. Griffen, *Multichip Modules Including Processing: A Literature Survey*, Army Research Laboratory publication ARL-MR-257, August 1995 (EX1046)
- J. Nedeau, D. King, D. Lanza, K. Hunt and L. Byington, *32-bit radiation-hardened computers for space*, 1998 IEEE Aerospace Conference Proceedings (Cat. No.98TH8339), 1998, pp. 241-253 vol.2 (EX1047)

B. Additional Materials Considered

6. In addition to those materials, I have read and considered the publications and materials including the following:

- Institution Decision (Paper 12 in this case)
- Sentient’s Patent Owner Response (Paper 18 in this case)

I have also reviewed and considered exhibits entered by the Board:

- Microsoft dictionary extract (EX3002)
- *Stallings* textbook extract (EX3003)
- *Hennessy and Patterson* textbook extract (EX3004)

I have also reviewed and considered exhibits submitted by Sentient:

- USAF Cooperative Research and Development Agreement between Air Force Research Laboratory, Space Vehicles Directorate and MSI (November 24, 1998 (EX2001) (Public Version)

- JTAG Article FPGA4fun(1) (EX2002)
- TAP Article FPGA4fun(2) (EX2003)
- Xilinx JTAG Programmer Guide 1991-1999 (EX2004)
- Xilinx XC4000 TAP Application Note (EX2005)
- USPTO Office Action (EX2006)
- Examiner Prior Art Searches (EX2007-EX2014)
- Purported Unsigned Letter (EX2017) (Public Version)
- Bus Arbitration In Computer Organization Article (EX2018)
- J. York, T. Powell P. Dehkordi, and D. Bouldin, *Enhancement of MCM Testability Using Embedded Reconfigurable FPGA*, Proceedings of the 1997 IEEE Multi-Chip Module Conference (Cat. No. 0-7803-4275), 1997, pp. 165-173 (“D2”) (EX2019)
- Declaration of John Peck (with Curriculum Vitae) (EX2020)
- Ch. 11 in A New Kind of Science. Champaign, IL: Wolfram Media, pp. 637-714, 2002. (EX2021)
- Sentient’s District Court Identification of claim constructions (EX2022)
- Xilinx’s District Court Identification of claim constructions (EX2023)
- Definition of computation (EX2024)

7. I have also reviewed and considered these additional materials submitted with my declaration:

- Joint Claim Construction Brief in *Sentient Sensors, LLC v. Cypress Semiconductor Corp.*, ECF No. 63 in 19-cv-01868-MN (D. Del.) (EX1051)
- NEW IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS, IEEE Press, 1993 (highlighted definition: “parallel”) (EX1052)
- Wikipedia entry: microprocessor (December 18, 2023 version) (EX1053)
- U.S. Patent No. 5,317,696 to Rolf B. Hilgendorf (“Hilgendorf”) (EX1054)
- Image of Intel 4004 (EX1055)

- D.A. Doane, P. D. Franzon, eds. *Multichip Module Technologies and Alternatives: The Basics*, Springer Science, 1993 (“Doane”) (highlighted) (EX1056)
- M. Alderighi, F. Casini, S. D’Angelo, D. Salvi and G. R. Sechi, *A fault-tolerance strategy for an FPGA-based multi-stage interconnection network in a multi-sensor system for space application*," Proc. 2001 IEEE Int’l Symposium on Defect and Fault Tolerance in VLSI Systems, San Francisco, CA, USA, 2001, pp. 191-199. (“Alderighi”) (EX1057)
- Diessel, O., H ElGindy, M. Middendorf, H. Schmeck, B. Schmidt, *Dynamic Scheduling of Tasks on Partially Reconfigurable FPGAs*, IEEE Proceedings-Computers and Digital Techniques, vol. 147, no. 3, pp. 181-188, ISSN 1350-2387 (2000) (“Diessel”) (EX1058)
- J. Emmert, C. Stroud, B. Skaggs and M. Abramovici, *Dynamic fault tolerance in FPGAs via partial reconfiguration*, Proc. 2000 IEEE Symposium on Field-Programmable Custom Computing Machines, Napa Valley, CA, USA, pp. 165-174 (2000) (“Emmert”) (EX1059)
- W. Fornaciari, V. Piuri, *Virtual FPGAs: Some steps behind the physical barriers*, In: Rolim, J. (eds) Parallel and Distributed Processing. IPPS 1998. Lecture Notes in Computer Science, vol 1388. Springer, Berlin, Heidelberg (1998) (“Fornaciari”) (EX1060)
- L. Levinson, R. Manner, M. Sessler and H. Simmler, *Preemptive multitasking on FPGAs*, Proc. 2000 IEEE Symposium on Field-Programmable Custom Computing Machines, Napa Valley, CA, USA, pp. 301-302 (2000) (“Levinson”) (EX1061)
- M. Ade, R. Lauwereins and J. A. Peperstrate, *Implementing DSP applications on heterogeneous targets using minimal size data buffers*, Proc. 7th IEEE International Workshop on Rapid System Prototyping. Shortening the Path from Specification to Prototype, Thessaloniki, Greece, pp. 166-172 (1996) (“Ade”) (EX1062)
- Microsoft Computer Dictionary, 5th Edition (2002) (highlighted definitions: “MIMD”, “multiprocessing”) (EX1063)
- W. Stallings, COMPUTER ORGANIZATION AND ARCHITECTURE: DESIGNING FOR PERFORMANCE (Prentice Hall 4th ed. 1996) (highlighted) (EX1064)
- Deposition Transcript of John C. Peck (EX1065)

- U.S. Patent No. 5,978,260 to Stephen M. Trimberger *et al.* (“Trimberger”) (EX1066)
- Oxford English Dictionary (<http://www.oed.com>: “computational”) (EX1067)

II. Person of Ordinary Skill in the Art

8. I understand there may be a dispute about the proper definition of the person of ordinary skill in the art (POSITA) for the '177 patent, which I described in my original declaration, EX1002 ¶¶ 25-26. As I wrote there, for purposes of assessing this level of ordinary skill in the art, I have considered the types of problems encountered in the art, the prior solutions to those problems, e.g., as found in prior art references, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field. As previously described, I have reviewed and understand the '177 patent, and I understand these factors. I give additional examples below.

9. As I noted in my original declaration, ¶¶ 7-11, I have personal knowledge of these factors and these issues. In my past, I served as the division leader responsible for much of the design, layout, fabrication, and testing of Draper Lab’s multi-chip module products, and participated directly in multiple MCM development programs. The MCM products were used as controllers for space and commercial satellites, sensor data collection, instrument controllers for terrestrial navigation, tracking and communication devices, and specialized government applications.

10. For example, as to the types of problems encountered in the art in December 2001 at the time of the creation of the '177 patent, I considered the descriptions describing multi-chip modules (MCMs) for space applications that I included in my original declaration, including “commercial off the shelf” solutions, similar to the prior art MCMs described in the '177 patent. *See* EX1002 ¶¶19, 24-25, 36-38. Not only am I personally familiar with the issues around constructing such MCMs in the late 1990s and early 2000s through my own work and knowledge, but I also reviewed prior art by others describing these problems, and I included reference to these papers in my original declaration. For example, EX1015, by Avery, describes “Recent progress in space-based micro-controller design.” This prior art identifies challenges of minimizing “the size, weight, and power” of MCMs used as instrument controllers by putting together parts designed by others. The paper presents a “typical micro-controller” system with a CPU, power, memory, connected with off-the-shelf parts, or challenges in building same while remaining economical, small, and power efficient. The paper describes several exemplary systems, including MAPLE I & II, DEEP SPACE II (which is describing Lyke’s AIC), and ETB. As I noted in my original declaration, e.g., EX1002, ¶65 (quoting Avery, EX1015), the problems addressed in this area included the need to build small, low power, inexpensive instrument controllers by integrating the right set of parts to remain low cost, small, and power efficient. *See, e.g.,* EX1015, 3, identifying problems of MCMs being too “large ... power hungry ... and

expensive.” I also considered EX1016, describing “space testing of the advanced instrument controller” which set out the problems for obtaining “lower performance (<1MIPS), low power, and small size end.” EX1016, 1. *See also* EX1003 (Lyke), 1:25-30, which describes “combin[ing] in a small MCM a number of functions not presently possible in a single integrated circuit.” Similarly, systems like Dehkordi, EX1004, are seeking to explore “low-volume, low-cost MCM prototyping for universities” where basic off-the-shelf components are put together into a MCM. Each of these systems are fairly straightforward implementations of MCMs; as such the skills needed to address the problems are found in the arts of electrical engineering or computer engineering, or a related field, with knowledge of MCMs for use as instrument controllers.

11. I considered the prior solutions to those problems that were known at the time of the creation of the ’177 patent and as noted in my original declaration. EX1002 ¶¶25-26, 36-38, 53-54, 65-66. These were found in prior art references, including EX1015, EX1016, EX1017, EX1003, EX1004, and the technical solutions presented. For example, EX1015 describes the applicant admitted prior art system also described in Lyke, and the AIC that integrates components designed by others into a normal MCM. Again, this suggests the skills needed to address the problems are found in the arts of electrical engineering or computer engineering, or a related field, with knowledge of MCMs for use as instrument controllers.

12. I also considered the rapidity with which innovations are made, and the sophistication of the technology. EX1002 ¶¶25-26, 35-38. Interestingly, the MCM systems being designed to address prior art problems and in the Lyke prior art, and the '177 patent itself are not the “cutting edge” systems sometimes found in patents, in that a common concern of designers was that much of the available technology was “too” advanced to use. EX1016, 1 (“This gap often forces the designer to use high-performance or high current processors for low complexity tasks”). For the applications at hand, such as in Lyke, the prior art cited above, and even in the '177 patent, the instrument controllers would have used parts known to basic engineering students (e.g., off the shelf RAM, CPU, EEPROMs, FPGAs, etc.) and been accessible to someone with a basic electrical engineering or computer engineering background, with modest exposure to MCM design such as described in the prior art cited above.

13. Similarly, I have considered the educational level of active workers in the field. EX1002 ¶¶25-26. The work performed here to design a MCM using off the shelf parts, and the concepts and prior art of the '177 patent, would have been in the ken of junior electrical engineers or computer engineers. For example, I know that the digital design needed to make a multi-chip module as described in the '177 patent with off the shelf parts (such as an FPGA, CPU, memory, etc.) was taught in electrical engineering classes in colleges in the late 1990s and it would have been solidly in the grasp of a EE/CE university graduate with two years of experience related to the design or

development of multi-chip modules operating as instrument controllers. *See also, e.g.*, EX1056, xxvii (text on MCM technology “for advanced undergraduate ... students”); EX1004 (university project to build a low-cost MCM for universities). This is another reason why I believe the relevant level of skill for would be akin to having an undergraduate degree in electrical engineering or computer engineering, or something similar, and a few years of experience related to the design or development of multi-chip modules operating as instrument controllers.

14. For these reasons, and having reviewed the ’177 patent and its file history prior to my original declaration and again on review of Sentient’s response, I agree with the Board’s adoption in ID (and similar to my opinion in EX1001 ¶26) that a POSITA would have had a Bachelor’s degree in electrical engineering, computer engineering, or a related field, and two years of experience¹ related to the design or development of multi-chip modules operating as instrument controllers. Further, as noted in EX1001

¹ In my original opinion, I said “at least two years of experience” and see the Board in its Institution Decision omitted the “at least.” I do not think that statement changes what I intended in my original statement of the POSITA. But just for clarity, my opinions in EX1002 and in my supplemental opinion here do not differ under the version of the POSITA adopted in the Institution Decision.

¶26, additional education could substitute for some experience, and substantial experience could substitute for some of the educational background.

15. I have considered Mr. Peck’s statement of the POSITA (at EX2020 ¶22) that the POSITA would have had “a Bachelor’s degree in Electrical or Computer Engineering or an equivalent degree with at least four years of experience in FPGA and processor systems design including but not limited to system on a chip ‘SoC’ and vehicle sensor systems or related technologies. Additional education may substitute for lesser work experience and vice-versa.”

16. My opinions in EX1002 and below do not change if Mr. Peck’s definition were adopted. In particular, we are both agreed that a Bachelor’s degree in Electrical or Computer Engineering (or similar) is the correct field of education. Mr. Peck would impose a larger number of years of experience focused on FPGA and processor system design, but something obvious to an engineer with less experience will also be obvious to an engineer with more experience. I have considered my opinions in view of Mr. Peck’s definition and if the Board adopted Mr. Peck’s definition instead of what it did in the Institution Decision, my opinions would not change.

17. Practically speaking, the ’177 patent is not directed to a particularly complex invention. Mr. Peck does not explain why “at least four years of experience in FPGA and processor systems design” is appropriate where the ’177 patent is neither about designing an FPGA nor a processor system; instead it is about designing a multi-

chip module (MCM). *See* EX1001, 3:26-30 (“Fig. 1 is a functional schematic block diagram of a preferred embodiment of the MCM Controller of the present invention.”), 3:40-46 (“Preferably, the MCM of the present invention...”), 1:14-15 (“The present invention generally relates to multi-[c]hip modules (MCM).”)

18. Nor is the focus on system-on-a-chip “SoC” or vehicle sensor systems. Instead, the ’177 patent distinguishes itself from system-on-a-chip. The ’177 patent says its MCM has some SoC characteristics, but is a MCM, not an SoC: “the invention **is similar to** a “system-on-a-chip” and appears similar to a chip in size, weight and physical configuration, **but through a tightly coupled MCM implementation,** **transcends** the functional capability of a single integrated circuit.” EX1001, 3:55-60. Given the description of the prior art as “prior art MCM controllers” (see EX1001, 2:43) the POSITA would know about those MCM controllers, and that experience would be reflected in the definition of the POSITA.

19. Further, the focus of the patent is not on “electronic vehicle sensor systems.” Instead, the ’177 patent discusses instrument controllers for “test devices and modules” that are used in outer space payloads such as satellites, EX1001, 1:24-30; *see also* 1:44-51 (“space-based test devices”). The ’177 patent gives a number of other examples of instrument controllers as well, including nuclear control systems (“safety interlock protocol management”), “credit card processors,” “miniature weapons computers,” and “wearable computers.” EX1001, 9:8-18. Mr. Peck does not explain

why he focuses on “electronic vehicle sensor systems” but the ’177 patent does not reflect that (or a subset of FPGA and processor systems design) as the proper focus for the POSITA of the ’177 patent.

III. Summary of Opinions Regarding Claim Construction

A. The Dispute

20. I understand that there may be a dispute between the parties as to two claim terms: (1) “gate array ... configured to run independent processes in parallel with the processor” at issue in Ground 1 of the Petition, and (2) “power converter” at issue in Ground 4 of the Petition. I further comment on these terms and Patent Owner’s arguments below, as well as my opinions about the plain and ordinary meaning of these terms.

B. Approach to Claim Construction

21. I understand that for purposes of my analysis in this *inter partes* review proceeding the terms appearing in the patent claims should be interpreted according to their “ordinary and customary meaning” (absent any special explicit definition given by the inventor in the intrinsic record, which is not the case here.) In determining the ordinary and custom meaning, the words of a claim are first given their plain meaning that those words would have had to a POSITA at the time of the purported invention date.

22. As I noted in my original report, I also understand that the claims themselves, the patent itself including the specification and images, and the file history

also may be used to better construe a claim insofar as the plain meaning of the claims cannot be understood. I have followed this approach in my analysis. I also understand that the words of the claims should be interpreted as they would have been interpreted by a POSITA at the time of the filing of the patent (December 19, 2001, i.e., not today).

C. Past Relevant Constructions and Arguments (Not Involving Xilinx)

23. As I wrote in my original report, I also understand that the '177 patent was the subject of an unrelated proceeding (involving Sentient, but **not** involving Xilinx) in the District of Delaware, in which Sentient made certain arguments, and in which the Court made certain rulings.

24. In particular, the term “configured to run independent processes” was construed by a judge in that matter based upon an agreed definition between Sentient and the unrelated party that “‘configured to run independent processes’ shall have its plain and ordinary meaning.” EX1012, 1.

25. Further, I understand that the prior court judge ruled that “‘power converter’ shall have its plain and ordinary meaning, specifically, a power converter converts power.” EX1012, 2.

26. I also understand that Sentient took certain positions as to these two terms, and I have read Sentient’s positions in EX1051 (pages 47-54 for “power converter” and pages 71-75 for “configured to run independent processes”).

D. Summary of Conclusions

27. The Institution Decision in this case observes that it appears that the parties have a different understanding as to the term “independent processes.” It is thus helpful to discuss the plain and ordinary meaning of the terms “process” and what is meant by “independent” processes in the patent.

28. I also understand that Sentient tries to re-define the meaning of “power converter” and discuss this below.

29. First, as to the disputed terminology relating to **“gate array ... configured to run independent processes in parallel with the processor,”** for the reasons I explain below, a summary of my conclusions is that:

- a. The plain and ordinary meaning of a **“process”** includes “a segment of code that may be run independently,” or “a program or part of a program; a coherent sequence of steps undertaken by a program.”
- b. The plain and ordinary meaning of an “independent process” includes “a process that runs separately from another process.” (Thus, logically, the plain meaning of **“independent processes”** includes processes that run separately from each other.)
- c. Though seemingly not in dispute, the plain and ordinary meaning of **“in parallel”** is “at the same time.”

30. Second, as to the disputed terminology “**power converter**” I have used the plain meaning that is the same that a district court judge has already found in an unrelated proceeding, i.e., a “‘power converter’ shall have its plain and ordinary meaning, specifically, a power converter converts power.” EX1012, 2.

31. Third, as I explain below, the asserted prior art in this IPR proceeding provides these claim elements and renders the asserted claims invalid under either the plain and ordinary meaning I have described above, or under Sentient’s proposed constructions. That is, even if the Board were to adopt Sentient’s view of the claim terms, the prior art asserted in Ground 1 or Ground 4 would still render the asserted claims invalid.

32. Fourth, I address some miscellaneous issues below raised by in the Patent Owner Response, including to rebut the argument that Dehkordi was incomplete or hypothetical, or that the POSITA wouldn’t have known how to build its teachings.

IV. Details of Supplemental Opinions

A. Ground 1: The Prior Art Provides a “gate array ... configured to run independent processes in parallel with the processor.”

33. I have read Sentient’s Patent Owner Response and Sentient’s expert Mr. John Peck’s declaration in support of same. EX2020. Mr. Peck’s declaration appears largely to be a copy of the Patent Owner Response (“POR”), and so I primarily focus on (and cite) the arguments that Patent Owner itself is presenting to the Board in its POR.

34. I see that Sentient is focused primarily on whether the prior art, i.e., Lyke, EX1003 or Dehkordi, EX1004, provide the “independent processes” limitation. In describing “independent processes” I see that Patent Owner appears primarily focused on clocking, with arguments like:

- a. POR, 14 (“A POSA would understand that the designers of D1 intended the control bus to allow the processor to control the FPGA with operational instructions, including providing a clock signal. Thus, for normal operation, the designers selected the FPGA to be dependent on and under the control of the processor.”)
- b. POR, 33 (“First, to the extent that there is any question whether the ‘399 Patent shows any independent parallel processes, the clear answer is no. As the ‘177 specification correctly notes, the ’399 Patent runs from a clock controlled by the processor, making computing circuitry dependent on that processor-controlled clock”);
- c. POR, 38 (“both components relying on the same clock, and which a POSA would understand were synchronized. As such, the FPGA is dependent on the processor, and they are necessarily in a dependent configuration. Consequently, no independent processes are possible during normal operation”)

- d. POR, 49 (“As discussed above, D1 in view of D2 does not run independently or in parallel with the processor despite being externally configured. This is also true for Halverson and Marchal. The same is true for Lyke because despite external configuration – it still operates on a processor-controlled clock”)

35. I also see Patent Owner citing the existence of a “control bus” in Dehkordi to argue that Dehkordi does not teach a FPGA that executes independent processes, with arguments like:

- a. POR, 38 (“DSP processor and the FPGA are connected through a *control bus* (highlighted in yellow) used to control the FPGA”) (emphasis in original)
- b. POR, 39-40 (“As annotated FIG. 1 of D2 shows, in normal operation (the “design for functionality” configuration - highlighted in orange) the processor controls the entire subsystem including the FPGA, the memory and the other resources necessary to operate the system. Alternatively, in the special testing mode (the “design for testability” configuration - highlighted in green) it is the FPGA that controls the entire subsystem including the processor, the memory and the other resources.”)
- c. POR, 41 (“The FIG. 2 architecture receives instructions (in orange) only from the CPU indicating that it is a unidirectional (one-way) path.”)

- d. POR, 42 (“As is clearly apparent from the above [figures showing a control line], D1-D2 operates under the control of the processor, meaning that its FPGA neither operates independently or in parallel during normal operation”)

36. I have reviewed the intrinsic and extrinsic evidence and believe that Patent Owner’s analysis is not correct for the reasons and support that I provide below.

1. *The Prosecution History Focused on “Multiprocessing”, Not Clocking or Control Line Direction*

37. I have examined the intrinsic record, including the prosecution history, to determine if there was any commentary made by the applicant or examiner that would shed light on this dispute.

38. First, the prosecution record indicates that the claim term at issue was **not** the reason that the patent examiner allowed the claims that eventually issued. Instead, the examiner **rejected** a prior version of the patent claim with the claim limitation “gate array ... configured to run independent processes in parallel with the processor” and explained that this element was not novel over the prior art. More particularly, the examiner in the prosecutor record rejected the earlier version of the claim due to the Cloutier reference (EX1040). The examiner wrote (see EX1010, 58):

Cloutier teaches a multiprocessor based on a multidimensional array of field programmable gate arrays (FPGAs) (column 2, lines 9-13, FIG. 1) adapted to function as a parallel processor (column 2, lines 65-67). In Cloutier, the multiprocessor provides improved data processing for applications requiring large number of operations (column 1, line 32-35). It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to use the FPGA-based processing elements, as suggested by Cloutier for the multi-chip module instrument controller disclosed by Lyke in order to run independent processes in parallel with the processor.

39. I see that the patent applicant did not dispute this rejection or say that Cloutier taught something different. Instead, the applicant cancelled the affected claims. Here is what the applicant submitted in response to the patent office. EX1010, 44:

The Rejection of Claims 1, 2, 11, and 12 Based on the Lyke Patent, the Clotier Patent, and the DaCosta Publication

On pages 3-6 of the Office Action, independent claims 1 and 11, and dependent claims 2 and 12, were rejected under 35 U.S.C. § 103(a), as allegedly obvious over the Lyke patent in view of the Clotier patent and the DaCosta publication. Applicant has canceled independent claims 1 and 11, and amended claims 2 and 12 to depend from allowable claims 6 and 13, respectively. Thus, Applicant believes this rejection to claims 1, 2, 11, and 12 is moot.

40. The above suggests that the examiner and applicant were focused on the “multiprocess[ing]” aspects disclosed in Cloutier. This is also apparent to me by looking at what the examiner was citing, in column 2 above. The paragraph in Cloutier talks about how the FPGA “could can be programmed in a wide variety of ways to perform a

wide variety of applications [and] function as a parallel processor.” EX1010, 58 (citing EX1040, 2:65-67). That paragraph gives examples, including that the “FPGA may be programmed to operate as one or more P[rocessing] E[lement]s [including] to operate as a multiple-instruction, multiple-data (MIMD) ... processor.” *See* EX1040, 2:67-3:14.

41. Having multiple processing elements operate as a MIMD processor meant having those processing elements work independently, usually on separate programs, at the same time. *See, e.g.*, EX1064, 599 (defining MIMD as “a set of processors **simultaneously** execute **different instruction sequences** on different data sets”), EX1063, 341 (defining MIMD as “...**parallel processing** in which central processing units **independently** fetch instructions and operate on data.”)).

42. Further, the examiner’s rejection based on a “multiprocessor” disclosure suggests that the examiner was focused on the “multiprocessing” aspect, which means separate processes running at the same time. *See also* EX1063, 354 (defining “multiprocessing” as “A mode of operation in which **two or more ... processing** units carry out one or more **processes (programs or sets of instructions) in tandem**”)).

43. It does not appear that a common clock or control was a consideration by the examiner in making this rejection, or by the applicant when responding to this rejection. In toto, the above reveals that this portion of the examination was focused on multiprocessing, and there is no evidence the rejection was based on processor clocking or control.

2. “in parallel”

44. For the reasons I explain below, I believe the plain meaning of “in parallel” as used in the ’177 patent claims is “at the same time” (or “simultaneously” to more grammatically fit with the claim language.)

45. While I do not think there is a dispute as to the meaning of “in parallel” as it appears in the claim language, I also do see the Patent Owner made a number of unsupported and unexplained statements about “in parallel” in the prior art references. *See, e.g.*, POR, 42 (“Thus, in light of D2, D1 is not operating independently or in parallel.”). Just for clarity, I want to comment on the plain meaning of “in parallel” in this claim.

46. I have read the claim construction briefing in a prior dispute involving Sentient and this patent, but not involving Xilinx. In that prior proceeding Sentient wrote that “the plain and ordinary meaning of the phrase in parallel, is “at the same time.” EX1051, 71. This is a reasonable statement of the plain and ordinary meaning of “in parallel” here, because it is in line with how the ’177 patent specification uses that term in the specification. *See, e.g.*, EX1001, 7:45-50 (“The FPGA provides the ability to use internal **parallel** processing. For example, the FPGA could be performing [tasks] **at the same time.**”). This simultaneity aspect also accords with the common dictionary definition of “in parallel.” *See* EX1052, 914 (IEEE Standard Dictionary of Electrical

and Electronics terms defining “parallel” as including “pertaining to the simultaneity of two or more processes.”).

3. “process”

47. As I explain below, I believe that the plain and ordinary meaning of “process” includes “a segment of code that may be run independently,” or a “program or part of a program; a coherent sequence of steps undertaken by a program.” These are quotes from extrinsic reference materials already in this case, as noted below.

48. The Institution Decision raises a question asking the parties to address what is meant by a process. ID, 14-15. There is no explicit definition of a “process” in the ’177 patent or the prosecution record and the term is somewhat general, akin in this context to describing a task that is running on a processing unit (e.g., a CPU, DSP, FPGA, etc.)

49. The Board added several pieces of extrinsic evidence to the record, which I have reviewed. The Institution Decision notes that a contemporaneous dictionary, the *Microsoft Computing Dictionary* defines “process” as “[a] program or part of a program; a coherent sequence of steps undertaken by a program.” EX3002, 423. In my opinion that is one reasonable understanding of the term process, where the reference itself is consistent with both my understanding and with other references such as EX3004.

50. Similarly, the record evidence includes an extract from a classic textbook *Computer Architecture: A Quantitative Approach*, which states that states further that “a

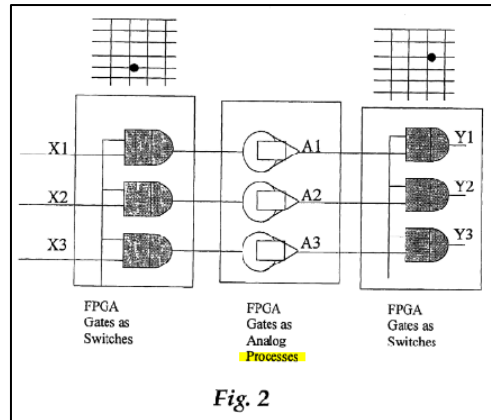
process is a segment of code that may be run independently.” EX3004, 530. That is another reasonable understanding of the term process. That definition goes on to describe additional details of a process (such as that “in a multiprogrammed environment, where the processors may be running independent tasks, each process is typically independent of the processes on other processors.”) Again, this is true and a reasonable statement of a “process.”

51. There can be added complexity to a “process” depending on the context. For example, in operating systems, the term “process” can take on additional meaning. For example, EX3003, 234-235 is talking about the meaning of a process in the context of MULTICS, which was a MIT mainframe timesharing operating system. But at the most basic well-understood level, the concept of a “process” is simply akin to a task that a processing unit (e.g., processor, FPGA) performs.

52. The ’177 patent does not add a special limitation to the broad concept of a process. For example, the ’177 patent gives an example of a process:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

EX1001, 8:18-26. *See also* Fig. 2:



53. The concept of “process” as including “a segment of code that may be run independently,” EX3004, 530; or “[a] program or part of a program; a coherent sequence of steps undertaken by a program,” EX3002, 423 is compatible with the ’177 patent’s example of the three programs running on the depicted FPGA gate partitions. More specifically, the ’177 patent describes “processing” on an FPGA in terms of how the FPGA is configured to perform various tasks (such as the “FPGA Gates as Analog Processes” caption above. EX1001, 4:45-49).

54. While a POSITA would typically speak of the code or program on the FPGA as the FPGA’s “configuration” or “configuration bitstream” (see e.g., EX1035 (“configured via a string of bits”)) the term “process” was also sometimes used and would have been understood as referring to the individual pieces of “program” or “code” (i.e., the relevant portion of the configuration bitstring), which in an FPGA are the partitions of gates configured to perform a task. As one example of a FPGA reference referring to the code or program configuration running on the FPGA as a process, *see, e.g.*, EX1066, 34:25-26 (referring to “a plurality of [FPGA gate] partitions, i.e.

processes”); 34:55-65 (talking about the various tasks that are running on the FPGA as “processes”); *id.* at 37:60-63 (same)).

55. For these reasons, I believe that the plain and ordinary meaning of “process” to include all of the meanings noted by the Board from extrinsic evidence, including that a process includes “a segment of code that may be run independently,” EX3004, 530; or “[a] program or part of a program; a coherent sequence of steps undertaken by a program,” EX3002, 423.

56. I have also reviewed Sentient’s proposal for “process” to be “a computational process programmed into the FPGA gates.” I think that definition is not particularly helpful for several reasons:

57. First, the definition of a “process” is defined as a “process” albeit a computational one, and limited to a set of FPGA gates. But it’s not clear to me what the Patent Owner is trying to do in adding the word computational. The ’177 patent is, of course about a computational system, just as is the prior art.

58. The specialized dictionaries in the record do not contain a definition for “computational” perhaps because that term is so generic. I note that the Oxford English Dictionary defines “computational” as “relating to, or used for computation; relating to or making use of computers,” which adds nothing in the context of the patent. EX1067. I do not see any language in the ’177 patent that explains any further limitation on what is meant by computational at the cites provided by Patent Owner. POR, 28 (citing

EX1001, 8:16-32 & 9:19-35). Neither the word “computational” nor the word “computation” appears in the ’177 patent.

59. Second, a “process” generally does not have to run only on FPGA gates, and by restricting a process to something “programmed into the FPGA gates” it excludes processes run on regular computer processors, for example, which have no such FPGA gates. Moreover, I note that the use of the term “process” in the patent claim is already specified as running on “a plurality of gates arranged in a field programmable gate array.” See EX1001, claim 1 at 10:4-7. Overall, Sentient’s construction is not consistent with what a POSITA would have understood as the plain meaning for the reasons I have just described.

4. “independent processes”

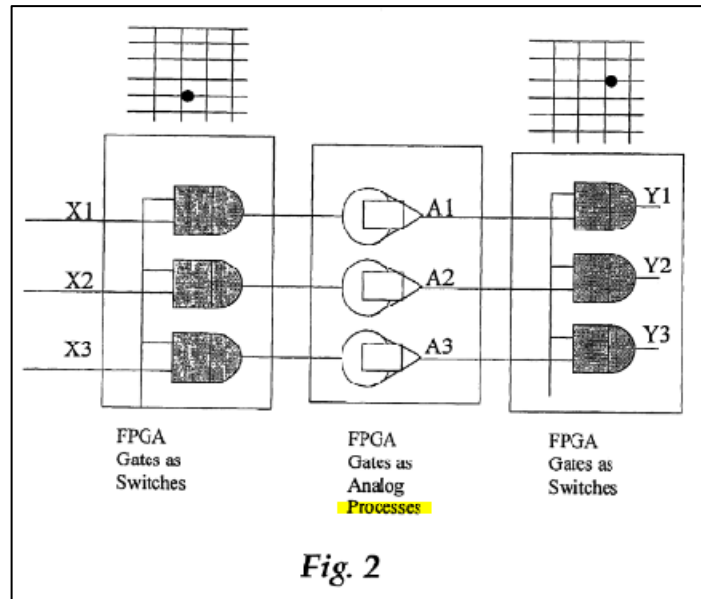
60. For the reasons I explain below, a POSITA at the time of the creation of the ’177 patent would have understood the plain meaning of an “independent process” to include “a process that runs separately from another process.” (Thus, logically, the plain meaning of “independent processes” includes processes that run separately from each other.)

61. The ’177 specification gives an example of “independent processes” and depicts such processes in the patent’s Figure 2. EX1001, 8:16-32, Fig. 2.:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the

FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

EX1001, 8:18-26. *See also* Fig. 2:



62. What is described and depicted above are three processes that are running separately—of independently—from another on the FPGA. As the patent describes, these three processes are “independent configurations of a subset of gates within the FPGA.” EX1001, 8:18-26. The way this works is that different partitions or sets of physical FPGA gates (e.g., as depicted in Marchal, EX1035, Figure 1) are physically configured to do different things. Each separate set of gates performs a process. Each process takes input, performs a “function”, and produces that process’s output, by itself. EX1001, 8:18-26.

63. One good reason to use a FPGA to implement redundant independent processes, especially in the context of the outer space environment that the ’177 patent

targeted, is to be able to run such “independent processes” in a fault-tolerant manner. In that setting, if one process fails due to a high energy particle interaction, it is important that failure is isolated and doesn’t affect another process.

64. In space, for example, ionizing radiation can change the state or damage some of the physical FPGA gates while other sets of gates can operate unaffected. *See, e.g.,* EX1057, 1 (describing “a fault tolerance strategy for an FPGA implementation of a redundant multistage interconnection network ... for a space multi-sensor system.”); EX1059, 165-166 (“Our approach is specifically targeted toward faults [such as] cosmic radiation, a-particles, etc.” and describing using replication to obtain FPGA fault tolerance); *see also* EX1042 (“Fault-tolerance has always been an essential attribute of [space] systems, to keep them operational in such hostile environment.”).

65. By keeping the processes independent from each other, for example by running each process’s programming in a disjoint partitions of the physical FPGA chip, then the system has an increased level of fault tolerance, which is what the ’177 patent also explains:

When the analog functions represented by A1-A3 are identical, then a form of functional redundancy is achieved, giving an increased fault tolerance to the signal monitoring function.

EX1001, 8:26-29.

66. Further, an FPGA is useful to run independent processes (i.e., processes that are separate from each other) even when the goal of system is not fault tolerance.

Instead, the FPGA's basic ability to run different programs at the same time without affecting each other (e.g., many different "processing elements" as Cloutier describes, and which the examiner noted as multiprocessor functionality), increases the capacity and functionality of a multi-chip module. The '177 patent also makes clear that the "independent processes" it contemplates do not have to be the same, and can "differ":

When the analog functions A1-A3 differ, then different analog processing functions are selectively applied to the analog signal being monitored.

EX1001, 8:29-32.

67. The ability to run multiple separate processes at the same time is a basic characteristic of a FPGA, which was well known to the POSITA by the time of the invention of the '177 patent: This fact is illustrated by contemporaneous references describing just such an operation, further illustrating my point that it was common knowledge in the field to configure distinct partitions (sets or portions) of FPGA gates to run separate and distinct tasks from each other, wherein each task would run a different algorithm. Examples include, for example, EX1040, EX1058, EX1060, 1061, discussed below.

68. Using separate parts of the FPGA to run separate programming was a well-known feature of an FPGA (sometimes referenced as spatial partitioning), the idea being that once in the system, it was "free" to add additional separate functionality into the FPGA that would run simultaneously (i.e., in parallel) with the other programming. This

is the concept of having many “processing elements” in a single FPGA, described in Cloutier, EX1040, and referenced above when I discussed the prosecution history and rejection around multiprocessing.

69. An additional example demonstrating this common knowledge known to the POSITA includes Diessel, EX1058, 2 which describes “partitioning the FPGA logic resource so that each function or task obtains as much resource as it needs and executes **independently of all others** as if it were the sole application executing on a chip”).

70. An additional example demonstrating this common knowledge known to the POSITA includes Fornaciari, EX1060, 3 (describing using the “FPGA to perform specific (usually, **independent and unrelated**) algorithms in hardware”).

71. An additional example demonstrating this common knowledge known to the POSITA includes Levinson, EX1061, 301 (describing as background, using FPGAs “to **execute several algorithms**, that do not require all FPGA resources, **in parallel in one FPGA, i.e., true multitasking**”).

72. I have also considered Sentient’s statements about “independent processes.” At page 28 of the POR, Sentient first says “**the proper construction** of ‘independent processes’ is ‘without computational assistance from another.’” Sentient then says at “**the plain meaning** of ‘independent processes’ are independent of each other and the processor.” The below appears to be the total argument and the support (or, more

accurately, a lack thereof) for this construction, and Mr. Peck’s declaration at ¶52 says the same thing, with the prefix “In my opinion.”

The proper construction of “independent processes” is “without computational assistance from another” which is consistent with Patent Owner’s previous litigation definition and the specification. See, for example, EX. 1001 2:9-18; 2:48-55. The plain meaning of “independent processes” are independent of each other and the processor. This is because grammatically speaking “independent” modifies “process” which is its plain meaning, and, as the Board points out - the specification supports both. Institution Decision at 16. EX 2020 ¶52.

73. I believe that Sentient’s first proposal, “without computational assistance from another” suggests Sentient is intending to add hidden limitations through the word “computational” as I discussed at ¶¶56-58 in my declaration above.

74. I believe that Sentient’s second proposal, that “‘independent processes’ are independent of each other *and* the processor” (emphasis added) is too limiting because of implicit arguments that Sentient is building into this definition, and because the examples shown in the patent for an “independent process” do not explicitly reference the processor. Perhaps more importantly, the addition of “and the processor” goes too far based on Sentient’s arguments as to what Sentient thinks that this extra limitation actually means.

75. I note again that Sentient’s support for this claim construction is a citation of its expert declaration, which repeats this text, adding no extrinsic support. *See*

EX2020 ¶52. The sole unsupported reasoning given there (in both the POR and in the declaration) is:

[In my opinion,] the plain meaning of “independent processes” are independent of each other and the processor. This is because grammatically speaking “independent” modifies “process” which is its plain meaning, and, as the Board correctly opines - the specification supports both.

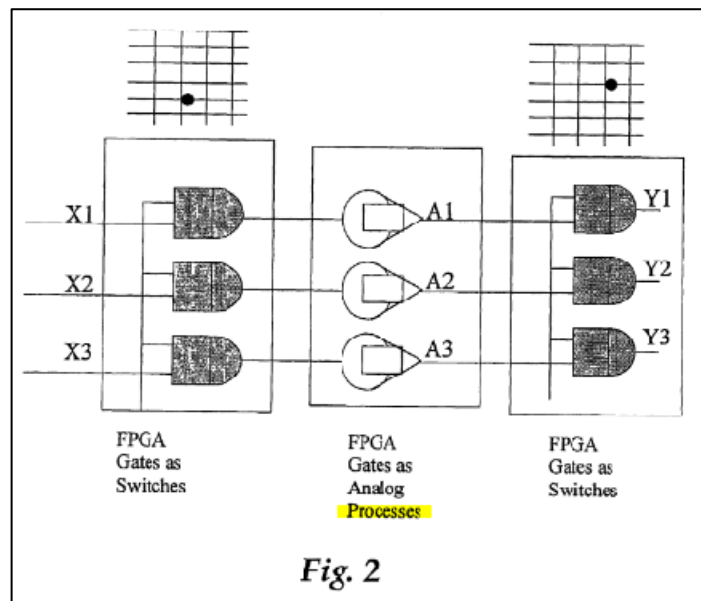
76. But that the “specification supports both” is not support for Sentient’s insertion of the “**and**” in the phrase independent processes’ are independent of each other **and** the processor” (as opposed to an “**or**”).

77. More specifically, I have reviewed the Institution Decision, where I see at 16 that the Board noted “at least two possibilities” and that “both interpretations find support in the ’177 patent.” ID, 16. Or, at ID, 17, the Board states: “The parties should clarify what the processes run on the FPGA must be independent of, e.g., each other or the processor, and support their respective positions with evidence.” Sentient’s argument seems to turn the Board’s “or”/“at least two possibilities” into an “and”; but neither Sentient nor Mr. Peck explain why it must be both.

78. As I noted above, the ’177 patent’s example of “independent processes” in Figure 2 and text describing that Figure concerns an independent process with respect to another process; it doesn’t mention the processor (or clocking, or control, which are how Sentient interprets “the processor” independence). *See* EX1001, 8:16-32, Fig. 2).:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

EX1001, 8:18-26. *See also* Fig. 2:



79. Within the '177 specification I also see an additional mention of “independent processes” when the '177 patent purportedly describes EX1003, Lyke.

There, the '177 patent states:

Lyke’s AIC is not capable of performing parallel independent processes. Although the AIC does include a specialized ASIC, it is controlled and clocked by the microprocessor and cannot run independently of the microprocessor. Accordingly, independent parallel processes are not possible.

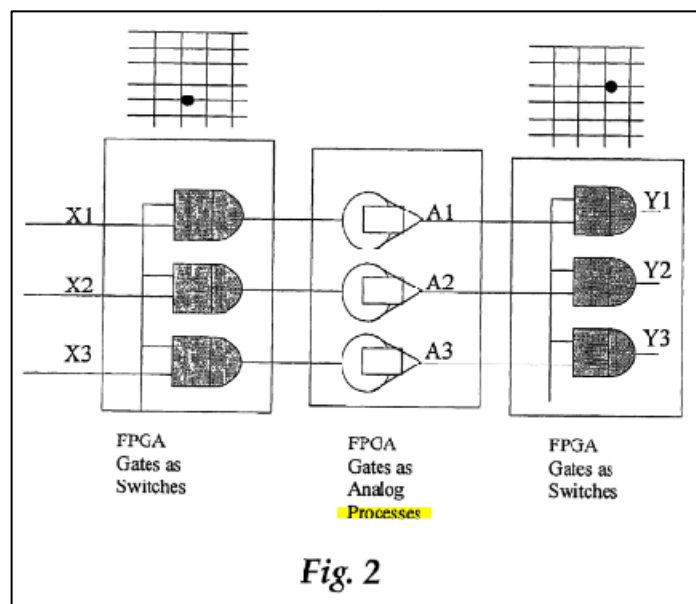
EX1001, 2:9-15.

80. The inventor is not describing Lyke accurately in this passage—this is apparent at least because the '177 patent inventor appears to have missed the disclosure of Lyke's FPGA completely.

81. For example, "independent processes" of exactly the type described in the '177 patent *would* be possible with Lyke's teaching of an FPGA running in the system. That is, had Lyke's FPGA been used, then exactly the same text in the '177 patent would describe Lyke's FPGA:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

EX1001, 8:18-26. *See also* Fig. 2, which would equally apply to Lyke's FPGA:



82. Nothing in that passage relates to clocking or control.

83. Further, the passage at EX1001, 2:9-15 refers to a single “specialized IC” in Lyke when there are two ASICs. I agree with the reading of the Board in the institution decision that “[t]he ’177 patent attributes Lyke’s alleged lack of independent processing to Lyke’s ‘need for a resistive ASIC.’ [EX1001] at 2:47–54.” ID, 4. That language reads:

Additionally, the circuitry design of the present invention **eliminates the need for a resistive ASIC. Instead**, the present invention adds a separately controllable FPGA that acts as a parallel processor with internal or separate external clock.

EX1001, 2:47-54. I also see that the patent applicant made the same statement to the examiner during prosecution. EX1010, 43.

84. I agree with the Patent Owner that Lyke’s “‘resistive’ ASIC is merely a set of ‘dumb resistors’ such as a custom resistor IC that are passive circuit elements and whose only purpose is to conserve space.” POR, 34. A bank of resistors would not typically have been considered to be a “process” at all in the sense of being able to “run independent processes” demanded by the claim language. A resistor is a passive circuit element made of a resistive material that limits the flow of electrons. A set of resistors cannot run “a segment of code that may be run independently,” or a “program or part of a program; a coherent sequence of steps undertaken by a program.” So I agree that Lyke’s resistive ASIC wouldn’t have been able to run independent processes.

85. As to control, I note further that the presence of control lines, or the direction of them, does not seem helpful in resolving whether there are “independent

processes” running on the FPGA. In fact, we know that what Patent Owner points to in Dehkordi (POR, 38), and the “D2” reference (POR, 42) as having control lines **does not mean** that the FPGA isn’t “configured to run independent processes in parallel with the processor”:

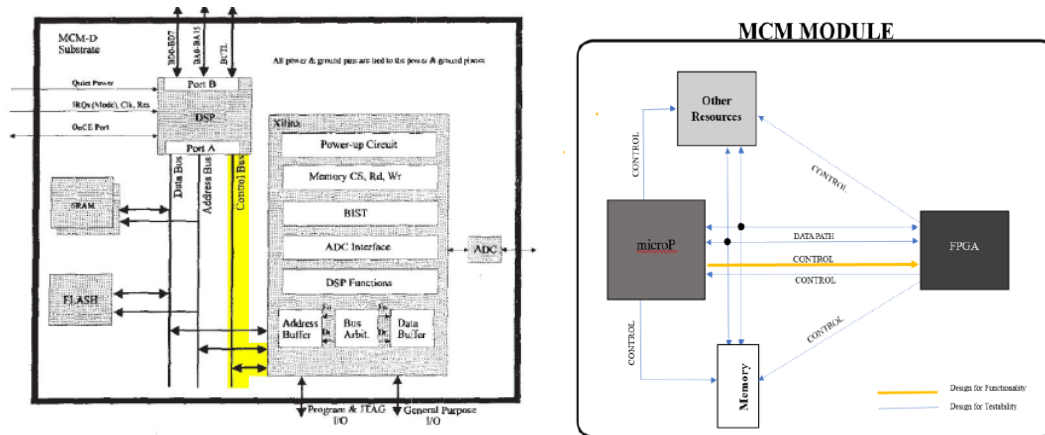
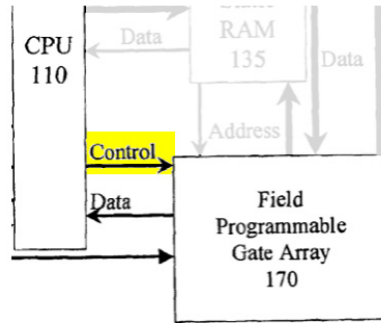


Figure 1. The Block Diagram of the DSP Sub-system.

Figure 1.1: Designing for Testability for a Typical MCM.

86. This is because we know that the '177 patent did not intend a control line to indicate whether there was a “gate array ... configured to run independent processes in parallel with the processor” from the '177 patent itself. More specifically, the '177 patent describes Figure 1 as shown below as “a preferred embodiment of the MCM Controller of the present invention.” EX1001, 3:34-37. Figure 1 shows a control line from the processor to the FPGA, and yet this is the same FPGA that is claimed as running independent processes:



EX1001, Fig. 1 (extract, emphasis added)

87. That is, the '177 patent's only disclosure of an example that practices the "independent processes" language has such a control line from the CPU to the FPGA. Sentient's focus on "control lines" seems misplaced, where the presence of a lower level control line does not mean the FPGA cannot run "independent processes" in the way that term was intended by the '177 patent, as the '177 patent itself shows in Figure 1 (and further, the presence of that control line has no effect on the discussion of "independent processes" around Figure 2 quoted above).

88. As to clocking, I further note that the '177 patent gives four examples of how the disclosed FPGA could be clocked, and does not comment that any of these four ways affects whether the FPGA "configured to run independent processes in parallel with the processor". See EX1001, 4:49-53 ("As shown, the FPGA may be [1] clocked using the same on-board VCO as the microprocessor; or it may be [2] clocked separately via an external clock 175. Alternatively, where [3] no clock is used, the FPGA will operate at gate speeds"); EX1001, 7:64-8:1 ("The invention provides for either external or [4] internal input of a clock signal from a high speed oscillator for controlled

sequencing of multiple processors formed in the FPGA for parallel processing.

Otherwise, if no clock signal is provided, then the FPGA operates at gate ripple speed”) (bracketed numbers added).

89. For example, the ’177 patent describes its FPGA as potentially being “clocked using the same on-board VCO as the microprocessor” EX1001, 4:49-50 without an effect on “independent processes.” That makes sense, where the ’177 patent’s affirmative example of “independent processes” has nothing to do with the source of the clocking signal. EX1001, 8:18-26.

90. In fact, the only example of “independent processes” running in the FPGA presumably do share the same clock. To see this, consider the teaching of the ’177 patent that one use of independent processes is to reach a result at the same time in order to provide redundancy of a calculation:

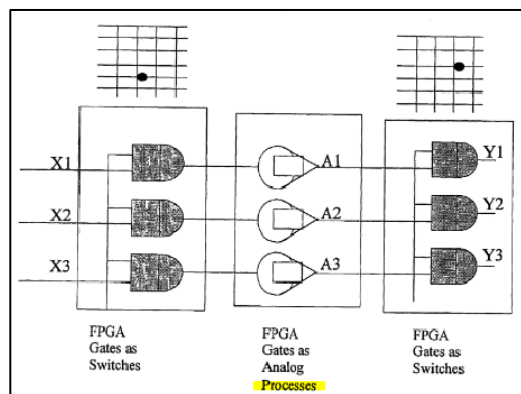
As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA.

Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B. When the analog functions represented by A1-A3 are identical, then a form of functional redundancy is achieved, giving an increased fault tolerance to the signal monitoring function.

EX1001, 8:18-29.

91. The passage that “when ... A1-A3 are identical, then a form of functional redundancy is achieved, giving an increased fault tolerance” suggests that the same

clock is used in all three processes, because the idea behind that “redundancy” is for these three processes to reach a simultaneous and (if all is working correctly) equal result and then for the system to check the result between the processes. Under normal operation of a deterministic process, the three processes would reach an identical result based on an identical input (e.g., to run a computation on a particular signal at a certain moment in time.) However, the “fault tolerance” comes into play because one of the processes residing in a portion of the FPGA could be erroneous—say from cosmic radiation bit flipping effects (see, e.g., EX1057, 2; EX1059, 165-166; EX1042)—and thus the computation is “bad.” Still, if two out of the three processes run in parallel on the same data and agree, then there is a basic form of redundancy by taking the two-out-of-three “majority” result, just as the patent depicts. This simple redundant computation would have been executed at the same time, as implied by the “AND” gates in the diagram below () – the idea being that the data from inputs X1, X2, X3 are “stepping” through to become outputs Y1, Y2, Y3 through the banks of switches/gates to be evaluated at the same time:



92. Based on this reasoning, even if the “independent processes” A1-A3 have a shared clock, that the common clock doesn’t detract from whether the processes are “independent”. In other words, “independent processes” is a different concept from having a shared clock, as the affirmative example of “independent processes” in the ’177 patent illustrates.

93. I have also considered the plain language of the claim and further observe that the language has no requirement about “independent hardware” or “independent clocking” or “independent control.” The patent applicant chose “independent processes” and not some different recital of what the purported invention was. Further, this focus on where the independence is would have been meaningful to the POSITA. At the level of a “process” (i.e., “segments of code,” or “programs or parts of programs”), a process typically will not have visibility into how it is clocked or initiated, but the processes can still be ‘independent’ in the way that term would be commonly understood by the POSITA, illustrate, e.g., as depicted in EX1061, 301 (describing “execut[ing] several algorithms...in parallel in one FPGA, i.e., true multitasking). Such details are at “lower level” than the task being executed by the process, meaning that at the level of a process, lower level hardware concerns (such as clocking, power, etc.) are not visible or relevant to a typical process task.

5. *The Challenged Claims 1 and 4 Are Obvious Under Any of these Definitions*

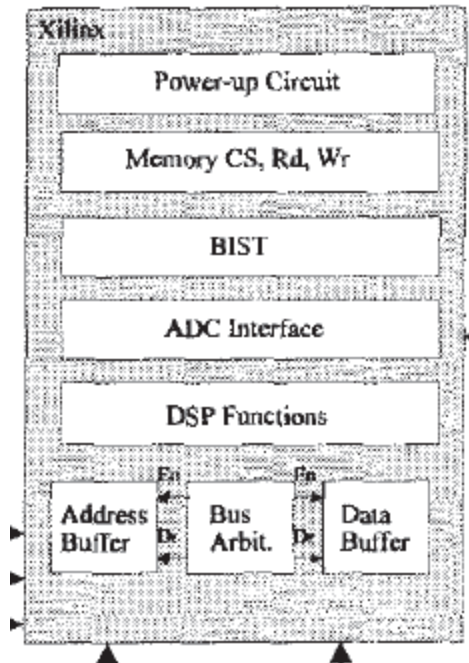
94. In this section, I explain why the above dispute is somewhat academic because the prior art provides the challenged claim elements under any of the definitions or plain meanings described above. Thus, while I think my opinions about claim construction are correct and supported by the intrinsic and extrinsic evidence, even if the Patent Board were to adopt one of the other meanings, my opinions that the prior art would have provided the asserted claims do not change.

(a) *The Prior Art Provides “Independent Processes” Under Xilinx’s Plain Meaning as “a process that runs separately from another process.”*

95. In this section, I focus first on the arguments based on Dehkordi, and then on the arguments based on Lyke. I note that the POSITA would have all the common knowledge in the art that I laid out in my first declaration EX1002 as well as in this supplement when reading the teachings of Dehkordi and Lyke.

Dehkordi-Based Arguments

96. First, **Dehkordi** provided a description of multiple independent processes that run separately, i.e., “independently” from each other process. Dehkordi shows such processes as blocks that are separated within the “Xilinx” FPGA in Dehkordi’s Figure 1 block diagram. See EX1004, Fig 1:



97. The POSITA would have seen this depiction and the text describing the internals of the FPGA (e.g., “**A portion of the FPGA is configured** for multi-processing bus arbitration” at EX1004, 2); (The FPGA is **primarily** a hardware pre-processor of incoming data **but also provides** other generic logic functions”, EX1004, 1) as describing one of the basic, well known ideas of an FPGA configured so that multiple different tasks occupy different parts of the FPGA and run independently at the same time; this fundamental idea is well supported by descriptions of FPGAs of earlier prior art: For example, Diessel, EX1058, 2, describes “partitioning the FPGA logic resource so that each function or task obtains as much resource as it needs and executes **independently of all others** as if it were the sole application executing on a chip”).

98. An additional example demonstrating this common knowledge known to the POSITA includes Fornaciari, EX1060, 3 (describing using the “FPGA to perform specific (usually, **independent and unrelated**) algorithms in hardware”).

99. An additional example demonstrating this common knowledge known to the POSITA includes Levinson, EX1061, 301 (describing as background, using FPGAs “to **execute several algorithms**, that do not require all FPGA resources, **in parallel in one FPGA, i.e., true multitasking**”).

100. This is one of the key advantages of an FPGA: the gates can be configured in basically any fashion subject to the design constraints of the FPGA, meaning that the FPGA can, act as many “processing elements” each performing their own task (i.e., simultaneous independent processes). *See, e.g.*, EX1040, Cloutier, 2:67-3:14 (“FPGA may be programmed to operate as one or more P[rocessing] E[lement]s [including] to operate as a multiple-instruction, multiple-data (MIMD) ... processor.”) This was common knowledge to the POSITA, even having its own short-hand (like MIMD) to describe it. *Id.*; *See also* Diessel, EX1058, 2; Levinson, EX1061, 301; Fornaciari, EX1060, 3, etc.

101. Moreover, in Dehkordi the Xilinx FPGA is running these multiple simultaneous independent processes (that do not overlap with each other) simultaneously with the process(es) running on the DSP. EX1004, 89 (“The [MCM] module was designed to exploit the multi-processing capability of the 96002 [processor].”) As I laid

out in my initial report, the Motorola DSP in Dehkordi is running other tasks while the FPGA runs. EX1002, ¶149:

Dehkordi teaches its particular DSP is a Motorola 96002 processor. EX1004, 1 (“[Multichip module] consists of a complete DSP subsystem based on the Motorola 96002 32-bit floating point DSP processor as shown in Figure 1.”). A POSITA would know that Motorola 96002 has its own built-in embedded “bootstrap ROM” and “program memory,” as shown in the 96002 data sheet, which control the operation of the DSP. EX1013, 1. The microprocessor thus runs the program in own program memory, separate from the FPGA.

That the microprocessor processes are running separately and simultaneously with the FPGA processes is further reinforced by the reference that Petitioner calls “D2,” EX2019. That reference, which Patent Owner claims is a “highly relevant reference [that] clarifies the operation of the MCM subsystem [of Dehkordi]” (POR, 17) states that “The Motorola DSP processor has a built-in boot algorithm that allows the loading of the 4K instruction set from a byte-wide memory source.” EX2019, 171.

102. As I noted in my original declaration, further confirming the microprocessor’s independent and parallel operation is Dehkordi’s depiction that “[t]he DSP has a dual data and address port architecture,” which a POSITA would have understood that the DSP can process data completely inaccessible to the FPGA, through “Port B” in the figure. EX1004, 3 & 1 (showing “Port B”). This is especially so in the context of Dehkordi’s description that “the [MCM] module was designed to exploit the multi-processing capability of the 96002 [processor].” EX1004, 89. In other words, this

suggests again that the DSP is intended to operate on its set of tasks, while meanwhile the FPGA operates on a separate set of tasks. Thus, even if “independent processes” must be separate and simultaneous from the processes running on the processor, Dehkordi provides this reading of the term.

103. Dehkordi depicts multiple “independent processes” from each other (and the processes running on the Motorola DSP). The various separate blocks and the disclosure that “the FPGA is configured for multi-processing” would have suggested to the POSITA to run “independent processes in parallel with the processor,” and the examples given in Dehkordi Figure 1 are not limited to “initialization” or “power up sequences.” For example, Dehkordi provides other processes including “DSP Functions” and “Bus Arbitration”: EX1004, 3 (“a portion of the FPGA is configured for multi-processing bus arbitration.” *id.*, 1 (“provid[ing] other generic logic functions required for implementation of the entire subsystem”); *id.*, 1 (“pre-processor of incoming data”).

104. Additionally, EX2019 that Patent Owner says “is “a highly relevant reference” to understanding Dehkordi, POR, 39, also shows that the POSITA would have known that Dehkordi provided independent processes in parallel with the Motorola processor, where the FPGA had separate processes for “an address decoder, an interactive user I/O interface and the means for reconfigurable computing,” EX2019, 170, as well as a “data monitoring” process. The “data monitoring” process is

particularly interesting here, because it clearly is running separately and simultaneously with the processor, such that if the processor fails, the data monitoring process on the FPGA is able to detect that failure, and optionally report that failure: If a “fault is detected [the monitoring process on the FPGA can] and notify its environment (PCB - LEDs) the probable cause of the error/fault. ... [I]t can flag the DSP as the fault cause. ... [I]t can flag the EEPROM as the cause of the error.” Meanwhile, this monitoring process is described as occurring in parallel to the other processes running by the DSP (e.g., that takes actions that might cause the monitoring process to determine the DSP is not operating correctly) and on the FPGA (including providing the disclosed routing of signals). EX2019, 172; see also EX2019, 171 (“The PCB is designed to allow the FPGA to be the main interface between the MCM and its environment.”)

Lyke-Based Arguments

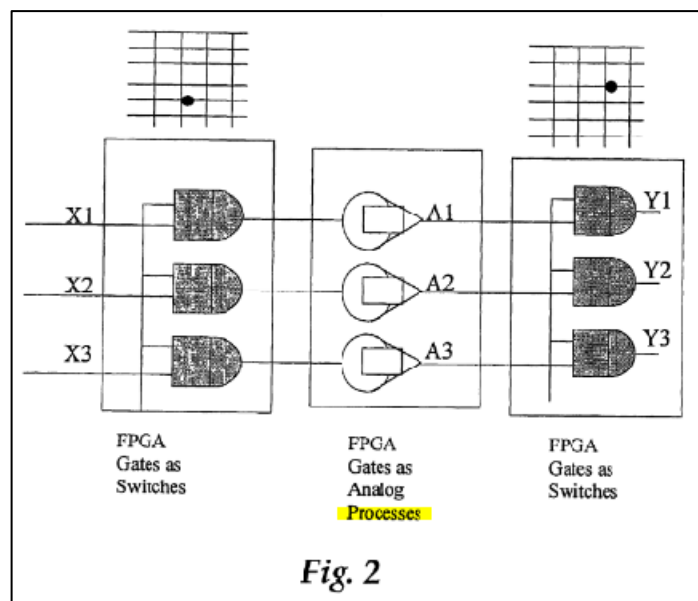
105. The prior art combination also provides the “gate array ... configured to run independent processes in parallel with the processor” element through Lyke. Lyke teaches an FPGA that ran a program downloaded from “non-volatile memory.” EX1003, 8:55-59. But the ’177 patent seems to ignore Lyke’s teaching about an FPGA entirely. For many of the same reasons above, a POSITA would have known that one of the fundamental features of an FPGA was the ability to run separate simultaneous processes. *See, e.g.*, Diessel, EX1058, 2 (describing “partitioning the FPGA ... so that each function or task ... executes independently of all others”); Fornaciari, EX1060, 9

(“independent and unrelated ... algorithms in hardware”); Levinson, EX1061, 301 (“to execute several algorithms ... in parallel in one FPGA, i.e., true multitasking”).

106. And if Lyke’s FPGA were used, then the same “independent processes” as described in ’177 patent would be directly possible. This is because the same description in the ’177 patent as to what an independent process is applies directly to what the POSITA would have recognized Lyke’s FPGA to provide as well:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

EX1001, 8:18-26. *See also* Fig. 2:



107. This is simply use of the configured FPGA for its intended purpose as laid out above, e.g., Diessel, EX1058, 2 (describing “partitioning the FPGA ... so that each function or task ... executes independently of all others”); Fornaciari, EX1060, 9

(“independent and unrelated ... algorithms in hardware”); Levinson, EX1061, 301 (“execute several algorithms ... in parallel in one FPGA, i.e., true multitasking”).

108. Further, Lyke’s disclosure (that was not noted by the applicant or examiner) of a “RAM-based field-programmable gate array(s) interfaced internally within the AIC and configured automatically through ... access of the non-volatile memory” (EX1003, 8:55-59) means that the FPGA was running its own programming distinct from what is run on the processor. The “program” that runs on an FPGA is a configuration string as to how to arrange gates within the FPGA. EX1035, 1. In contrast, the “program” that runs on a processor is something very different, i.e., assembly code that executes specific operands (like ADD, MOVE, etc.) that are called by a process. EX1013, 91 (showing sample assembly language code). Any process running on the FPGA is different and distinct from whatever is simultaneously running on the processor.

109. Thus, the POSITA would have understood Lyke’s FPGA to be running “independent processes in parallel with the processor” pursuant to the meaning of independence taught by the ’177 patent (EX1001 at 8:20-24), as well as in the meaning of processes running on the FPGA that would be executing at the same time as the processor.

- (b) *The Prior Art Also Provides “Independent Processes” Under Sentient’s Proposal that “‘independent processes’ [means] ‘without computational assistance from another.’”*

110. I see that Sentient makes two statements as to how Sentient would construe “independent processes.” POR, 28. First, Sentient writes that “independent processes” means “without computational assistance from another” and then Sentient writes the phrase means “independent of each other and the processor.” I address both in turn in this subsection, and the next subsection.

111. Taken literally, Sentient’s proposal of “without computational assistance from another” is initially similar (if one puts aside Sentient’s implicit additional restrictions it applies to this language) to what I just discussed above, with my understanding as “a process that runs separately from another process.” Specifically, the processes that are shown and described in Dehkordi each operate on their own, in a space partitioned manner on the FPGA. None of the processes (e.g., bus arbitration, etc.) are receiving “computational assistance” from another process, and are independent in the same way shown in Figure 2, describing “independent processes.”

112. For example, the “portion of the FPGA is configured for multi-processing bus arbitration” is separate from the processes providing “DSP functions.” This reflects a common way that the FPGA array of gates is partitioned to run several different tasks in parallel. For example, see Dehkordi’s teachings ((e.g., “**A portion of the FPGA is configured for multi-processing bus arbitration**” at EX1004, 2); (The FPGA is primarily a hardware pre-processor of incoming data **but also provides** other generic logic

functions”, EX1004, 1)); see also my discussion of Diessel (EX1058, 2) Fornaciari (EX1060, 3), and Levinson (EX1061, 301) described above.

113. Similarly, as to Lyke, the presence of the “RAM-based field-programmable gate array(s) interfaced internally within the AIC and configured automatically through ... access of the non-volatile memory” (EX1003, 8:55-59) means that the FPGA was running its own programming distinct from what is run on the processor. As noted just above, in the last section, the POSITA would have known that the programs simultaneously running on the FPGA were separate from the CPU, not least by their separate (and incompatible with each other) literal programming. *Compare* EX1035, 1 (discussing the configuration string on an FPGA) *with* EX1013, 91 (showing sample assembly language code converted into binary and run on a processor).

114. I further incorporate my discussion above where my reasoning with respect to the plain meaning as “a process that runs separately from another process” applies to Sentient’s “without computational assistance from another.”

(c) *The Prior Art Also Provides “Independent Processes” Under Sentient’s Proposal that “‘independent processes’ [means] ‘independent of each other and the processor.’”*

115. For all of the reasons discussed in the prior two sections, the asserted art in this case provides processes that are “independent of each other.” For example, the ’177 patent provides the clearest meaning of “independent processes” in its discussion that “three independent processes A1-A3. ... are independent configurations of a subset of

gates within the FPGA.” EX1001, 8:18-26. The processes disclosed in the prior art are physically separate from each other, and also separate from any process running on either Dehkordi or Lyke’s CPU.

116. Further, even if the Board were to adopt Sentient’s definition that “independent processes” have no processor control or clocking, the prior art provides the “independent processes” teachings under that meaning too.

(a) **clocking**

117. If found to be a requirement by the Board, the prior art teaches this element. For example, the POSITA would have known Dehkordi’s “**particular FPGA has an internal system clock**,” EX2019, 172. That quote is from the D2 reference, EX2019, that Patent Owner describes as “a highly relevant reference” as to what the POSITA would know about Dehkordi. POR, 39. Dehkordi discloses use of a particular FPGA – a “Xilinx 4010 FPGA” – which is an off-the-shelf physical part whose characteristics were known. EX1014, 23-24 (“**XC4000 Series devices include an internal oscillator...** Therefore, if the primary oscillator output is running at the nominal 8 MHz, the **user has access to an 8 MHz clock**”).

118. This internal clock is not just “used as the source clock in configuration modes in which the FPGA is master,” ID, 37, but is also in *every* configuration mode when the OSC4 module is included in a process’s design. EX1014, 24 (“These [clock] signals can be accessed by placing the OSC4 library element in a schematic or in HDL

code.”) The POSITA would have known (just as D2 reflects) that Dehkordi’s use of a Xilinx 4010 FPGA was thus teaching running processes with the FPGA’s clock, such as the detailed monitoring process described in D2.

119. Such monitoring processes, like the bus arbitration process and other processes in Dehkordi, ran on their own, separate (and not under the control) from what was running on the processor: For example, EX2019 describes how the FPGA had the ability to signal to the outside world when the “monitoring” process detected an error, regardless of what the processor was doing. *Id.*, 171-72. What that means is that even if the processor went into an error or frozen condition, the FPGA processes would not stop—and indeed would continue operating—such as by running bitstream code to turn on LED segments as described in EX2019, 171-72 (“Since this particular FPGA has an internal system clock, it can stop operations once an error or fault is detected and notify its environment (PCB - LEDs) the probable cause of the error/fault. [The] FPGA can flag the DSP as the fault cause.”).

120. Dehkordi itself further taught “fault isolation” and that “[t]he FPGA can easily be reconfigured to route these lines to the module pins for possible probing and debugging during testing.” To the POSITA, this would have further suggested the need to run a separate process for such fault isolation in the manner later described in detail in D2. That is, to isolate a fault and “probe” a MCM, the FPGA would need to “sit above the fray” (so to speak), i.e., able to monitor the chips in the system independently, with

its own independent process running off its own clock, so that it would not interfere with the other chips in the system, and could detect and report on errors even if other chips were not working. EX1004, 3. *See also* D2, EX2019, 167:

The FPGA can be **configured as a pseudo built-in logic analyzer that can monitor core dies within the MCM in real time**. This allows testing of not only the hardware aspect, but also allows **monitoring of software execution** in a limited manner. Moreover, a FPGA can be configured as a self-tester for the module by **creating stimuli and monitoring results**. If the FPGA has access to the external I/O of the module itself, **testing results can be sent as output and observed as they become available**.

121. Thus, regardless of whether the processor was operating correctly, the monitoring process (running simultaneously, and on an FPGA with other processes such as “multi-processing bus arbitration” (EX1004, 3)) would be able to detect and report out “testing results ... as they become available” as that process was running on the FPGA’s internal clock. D2, EX2019, 167.

122. As a further aside, the above discussion shows why the Patent Owner’s unsupported arguments about a POSITA’s understanding are wrong: Without support, the Patent Owner argues about Dehkordi for example (at POR, 38-39) that:

As such, the FPGA is *dependent* on the processor, and they are necessarily in a dependent configuration. Consequently, no independent processes are possible during normal operation and neither the Petition nor D1 suggest otherwise. EX. 2020

¶68.

To the contrary, a POSITA would have known of the independent clocking features generally of such FPGAs, including as shown in the Xilinx reference manual, EX1014, as well as D2, EX2019. These references, which were both known to the POSITA, support my opinion that the POSITA would have known that an FPGA has a separate internal clock—which completely undermines the supported argument above that the FPGA is “necessarily [run] in a dependent configuration.”

123. In sum, if the Board were to find that a clock separate from the processor’s clock was actually required for “independent processes”, for the reasons detailed above, the POSITA would have known by the teachings of Dehkordi’s “Xilinx 4010 FPGA” that Dehkordi provided that element.

124. D2, EX2019, is further evidence of what was known to the POSITA in interpreting Dehkordi, EX1004 as to the POSITA’s understanding of the prior art. *See, e.g.*, EX1014, 6-46 (describing multiple modes of operation and internal clocking); 23 (noting that “XC4000 Series FPGAs include an internal oscillator.”).

(b) control

125. If found to be a requirement by the Board, the POSITA would have known that the processes operating on the FPGAs in the prior art (and in particular, Dehkordi) operate without processor control.

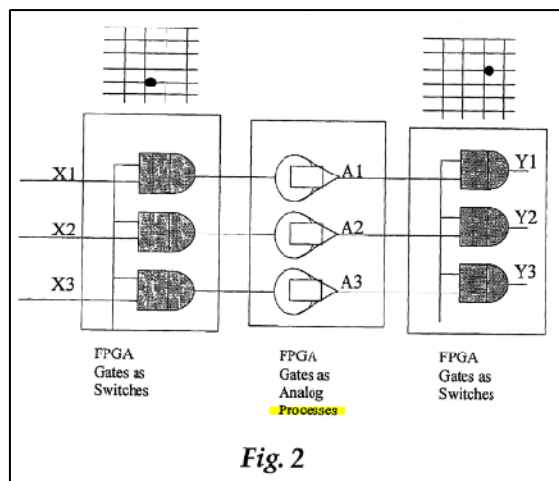
126. As a basic point, each of the processes shown in EX1004, Figure 1 running on the FPGA would be its own code, not provided by the processor. Dehkordi explicitly

taught the source of such FPGA configuration code “could come from outside the module,” i.e., would be supplied from a source other than the processor, EX1004, 3, which was a typical way of loading a configuration (*see e.g.*, EX1014, 3, discussing loading configurations from external memories or devices). Further, when running, the code would be independent: The “bus arbitration” process, for example, is not described as operating under processor control, but instead is running on its own accord. EX1004, 3 (“A portion of the FPGA is configured for multi-processing bus arbitration and is connected to the internal data and address bus on the module”).

127. This is further seen by the definition of “independent processes” in the ’177 patent:

As shown [in FIG. 2], gates X1, X2 and X3 are configured to route the input at node A to anyone of three independent processes A1-A3. The processes are independent configurations of a subset of gates within the FPGA. Switches Y1, Y2 and Y3 are configured to selectively route the outputs from these processes to the output node B.

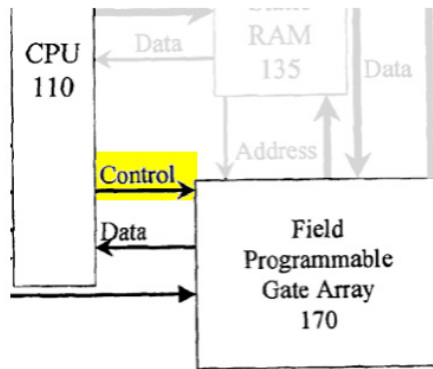
EX1001, 8:18-26. *See also* Fig. 2:



128. In the above description, the fact that data is provided to the analog processes A1-A3 from another source on the '177 patent's MCM does not render the *processes* as non-independent.

129. I further understand one of Sentient's primary attacks is to rely on pictures showing "control" lines or busses, saying that the process controls the FPGA. *See* POR, 41-42 ("As is clearly apparent from the above, D1-D2 operates under the control of the processor, meaning that its FPGA neither operates independently or in parallel during normal operation. Thus, in light of D2, D1 is not operating independently or in parallel.").

130. But this argument fails to undermine what the POSITA would have known on reading Dehkordi for several reasons. First, the presence of control lines or control bus is not what determines whether the prior art is teaching "independent processes." The presence of a hardware control lines (or the a control bus) does not really speak to whether the *processes* running on the FPGA are independent from the processor (i.e., what is required by the '177 patent claims). We can observe this with confidence for several reasons, not least of which is that the '177 patent itself depicts a single control line from the processor to the FPGA, that the '177 patent claims as running independent processes:



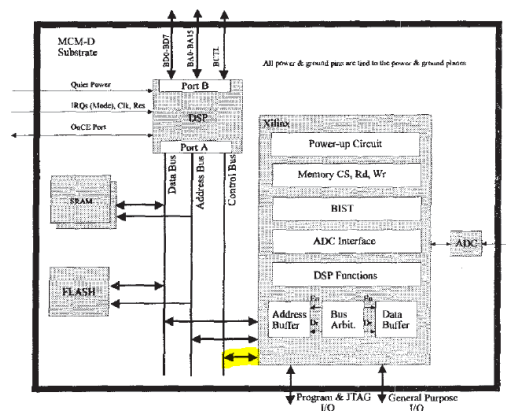
EX1001, Fig. 1 (extract, emphasis added)

131. That is, the “MCM Controller of the present invention” (EX1001, 3:29-30) shows a single control line (highlighted above), which is never discussed as having anything to do with “independent processes.” The fact that there is a control line doesn’t restrict the “independent processes” running on the FPGA in any way.

132. This further makes sense when we look at the limited amount of data is that is sent over control lines. As shown in the technical manuals for the Xilinx FPGA disclosed in Dehkordi, there is nothing in the control lines operating at a “process” level. *See, e.g.*, EX1014, 7 (Xilinx FPGA). Low level control lines, like the control signals disclosed on the Xilinx 4000 series FPGA present in Dehkordi are not about *process* control. Thus, the Patent Owner’s unsupported statement that “A POSA would understand that the designers of D1 intended the control bus to allow the processor to control the FPGA with operational instructions” is not credible given (1) what the Xilinx FPGA discusses as control lines, (2) what Dehkordi shows (including independent processes, such as bus arbitration, that would not be taking “operational instructions”

from the processor) and (3) what D2 further shows in the way of the “monitoring” process that further describes what the POSITA would have known coming to Dehkordi’s discussion of a need for “fault isolation” techniques.

133. Next, to the extent control lines *were* meaningful as to “independent processes” (which I do not think is the case, given the ’177 patent’s own counter example in its Figure 1, EX1001 Fig. 1), the POR tries to disprove Dehkordi’s teaching of a bi-directional control line with D2’s Figure 1. But D2’s Figure 1 shows an abstract “[t]ypical MCM” and not the Dehkordi MCM. Dehkordi’s MCM has many more elements than what is shown in D2’s Figure 1. I take Sentient’s main point about D2’s Figure 1 is that it doesn’t show a double arrow on the “control” line for “design for functionality,” such a line and teaching *is* present in Dehkordi, as highlighted below:



134. In fact, Figure 4 is what is described as “show[ing] a block diagram of an MCM designed at the University of Tennessee,” and then cites Dehkordi. EX2019, 5:

3. Prototype MCM

Figure 4 shows a block diagram of an MCM designed at the University of Tennessee [11].

...

[11] P. Dehkordi, T. Powell, and D. Bouldin, "Development of a DSP/MCM Subsystem Assessing Low-volume, Low-cost MCM Prototyping for Universities", in *Proceedings of 1996 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 6-7 1996, pp. 360-370.

EX2019, 169, 173. Notably, there is no control connection at all between the DSP and the Xilinx chip in that figure: Instead, the control bus to the FPGA is from off-MCM, and the FPGA also shows control of the ADC through the "ADC Control" lines:

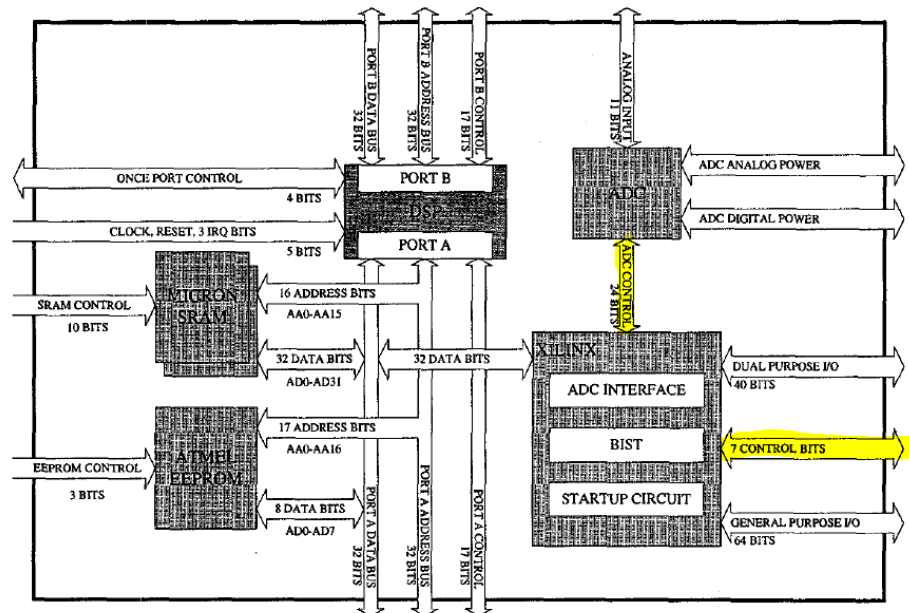


Figure 4: Block (Flow) Diagram of DSP MCM.

EX2019, 169. To the extent the Board would find the control lines matter to “independent processes” (and again, I don’t think they do given the unidirectional control line in the ’177 patent Figure 1), if anything, the lack of control lines in the

“flow” diagram of the MCM stresses the independence of processes running on the FPGA from the processor.

135. I also see that D2 expands the processes that a POSITA would know would be run on Dehkordi. D2 additionally provides that “the FPGA provides logic for an address decoder,” and “an interactive user I/O interface.” Those again are separate processes from the processor, including as depicted in the figure above. EX2019, 169. D2 also describes how “the FPGA could be used to test the MCM system.” *Id.*, 170. “The FPGA can be configured as a pseudo built-in logic analyzer that can monitor core dies within the MCM in real time.” *Id.*, 167. D2 details a particular monitoring process, and how that process running on the FPGA has characteristics, including that the process: (A) is independent from whatever process is running on the processor (e.g., the FPGA runs a passive monitor of the entire MCM, determining if any chips in the system are defective); and (B) is running simultaneously to detect faults, i.e., in parallel to the program that is running on the Motorola DSP processor.)

136. In sum, it is my supported opinion—for the reasons and evidence that I cite and describe above—that if the Board were to find that “independent processes” required independent clocking and control from the processor that the prior art (and specifically Dehkordi) clearly provided this requirement.

B. Ground 4: The Prior Art Provides the “Power Converter” Element

137. I also see from Sentient’s Patent Owner Response and Sentient’s expert Mr. John Peck’s declaration that Sentient disputes whether the prior art Steele reference provides the “power converter” element.

138. First of all, my understanding is that Sentient is now proposing a claim construction that has already been rejected once in a different district court proceeding that did not involve Xilinx. *See* EX1012.

139. In that proceeding, Sentient argued that a power converter is “a converter which changes a voltage level to another voltage level.” EX1012, 7. I understand that this is the same proposal that Sentient advances again now. *See* POR, 26. *See also* EX1012, 7 (district court rejecting that very same construction).

140. Instead, I understand that the district court judge in the unrelated prior case construed “power converter” as having “its plain and ordinary meaning, specifically, a power converter converts power.”

141. Neither Sentient nor its expert Mr. Peck provides any reasoning to explain why the plain and ordinary meaning is incorrect, or to address (or even acknowledge) that the as found by the prior district court judge.

142. In my opinion, the “power converter” term needs no special construction and I used in my earlier opinion the plain meaning issued by the prior court, i.e., that a

“power converter converts power.” For that analysis, see my original declaration, EX1002 ¶¶115-120.

143. At the same time, even if the Board were to go in a different direction and adopt Patent Owner’s meaning, then the prior art still provides that element for the reasons I give below.

144. Even if “a power converter is ‘a converter which changes a voltage level to another voltage level.’” then Steele provides that teaching: “Yet another object of the present invention is to provide an output signal voltage level **which is derived from** that signal’s input power supply.” EX1009 at 2:40-43 (emphasis added); *See also* EX1009 at 4:41-42 (“Meanwhile, output signals **generated** by CFB1 are at a 3 volt level. These 3 volt output signals can then be used by those devices operating on signals at a 3 volt reference.”) (emphasis added); 4:50-52 (“Note that even though the PLD, itself, is powered by 5 volts, a user has the ability **to choose the voltage level for signals output from the CFBs.**”). *See also* EX1002 ¶¶47-48, 114-120 (citing EX1009, 4:50-58).

145. Steele’s teachings include a power converter in which the supplied voltage is 5 volts and the output voltage is variable, including 3 volts in a preferred embodiment. I see the Board noted this as well when issuing its decision instituting this trial. *See also* ID at 40 (noting that “[Steele’s] FPGA can be configured so that different CFBs output different voltages on their respective V_{CCO} pins. *Id.* at 2:35–47, 2:54–65, 4:7–43. In an

exemplary embodiment, either 3 V or 5 V is output on the various V_{CCO} pins. *Id.* at 2:66–67.”).

C. Sentient’s Miscellaneous Arguments About Dehkordi

146. I also understand Sentient has claimed that Dehkordi should be ignored or disregarded purportedly because “at the time of its publication in 1996, the DSP system was still in fabrication, with its operation and configuration still theoretical and therefore not enabled.” That’s not correct. Dehkordi states that those university authors had built “a total of 5 MCMs” and reported on same. EX1004, 1 (“The complete subsystem is interconnected on a 37 mm by 37 mm MCM-D substrate and packaged in a 320-pin ceramic quad flat pack [CQFP]”); *id.*, 3 (“**We have prototyped a total of 5 MCMs** which each MCM costing about \$ 2,780. ... The substrate cost of \$1,480 per MCM included the fabrication and test of the substrate, assembly of the dies onto the substrate and the CQFP package.”) A POSITA would have known how to build the system in Dehkordi, which was a routine collection of off-the-shelf parts arranged in the way they were intended into a multi-chip module. Moreover, EX2019, the D2 paper, shows a labeled “photograph” of the Dehkordi MCM:

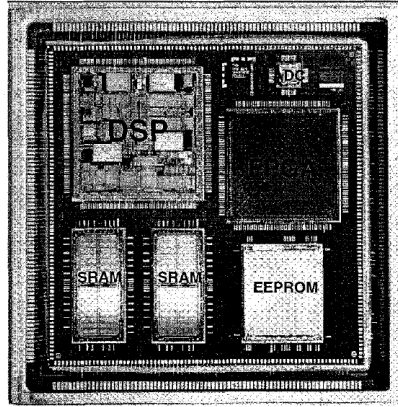


Figure 5: Photograph of the DSP MCM.

EX2019, 170. In my opinion, there is no question that the teachings of Dehkordi were sufficiently described such that a POSITA with that POSITA's background knowledge in 2001 would have understood and been able to build the system described in Dehkordi, and to take and apply the teachings that I have explained above and in my original opinion. As Dehkordi itself said about the MCM technology in its paper: "This technology is no longer considered to be an exotic technology."). Instead, the focus of that paper was on taking something well known, accessible, and low cost and showing that it could be built in a university setting, without the resources of industry. EX1004, 1 ("Our experience shows that low-volume MCM prototyping is **achievable** ... for universities.") (emphasis is original in Dehkordi). The POSITA certainly could have made Dehkordi from the disclosures in that reference and her or his background knowledge.

D. Sentient’s Miscellaneous Arguments About Halverson, Kostarnov, and Other Art

147. Finally, the Patent Owner attacks my reliance on other prior art to show what was known to the POSITA. Generally, I rely on references like Halverson, Marchal, and Kostranov to illustrate points about what was known in the art, and common knowledge to the POSITA.

148. Rather than just provide “because I say so” *ipse dixit* solely based on my knowledge in this area, I use these references to illustrate my points and support my opinions about the main grounds and art like Lyke and Dehkordi. When Dehkordi discloses a particular Xilinx XC4000 series FPGA in its MCM, for example, that would have meant something to the POSITA. I know this because of my own background, experience, and knowledge of how a POSITA would have approached the references—and I can illustrate this by citing exemplary background art. References like Kostranov (which describes modes of operation on a Xilinx XC4000 series FPGA, EX1036, 80) support my views on the common knowledge that the POSITA would have brought in understanding Dehkordi’s disclosure (which also includes a Xilinx XC4000 series FPGA). Similarly, when Lyke discloses an RAM-based field programmable gate array, the POSITA would have had a base knowledge as to what that disclosure meant. Again, references like Marchal, EX1035, and Halverson, EX1025, support my opinions about what the POSITA would have understood by reading Lyke and Dehkordi’s teachings.

Sentient tries to downplay the knowledge of the POSITA, but such knowledge is relevant when assessing the art.

149. As to Halverson, the POR claims that Halverson “does not operate independently.” POR, 32, 49 (“Halverson does not disclose independent parallel processing but discloses just parallel processing.”) But Halverson notes the advantages of “**processing unrelated instructions** simultaneously within the same interval of time.” EX1025, 1:18-20. Halverson further describes: “Traditional parallel data processors execute simultaneously with a main processor **instructions that are unrelated**, that is, not dependent on the result of the other.” EX1025, 1:27-30. “The present invention provides ‘expression level’ parallel processing capabilities to computers through the use of field programmable gate arrays (FPGAs) that are dynamically configured to perform the computations associated with application programs.” Id., 2:18-40.

150. Similarly, in Kostarnov, EX1026, the POR (at 40-41) wrongly focuses on just one of the modes of operation of a FPGA/CPU, without engaging with the remainder of the references teachings, such as the datapath mode, which Kostranov describes thusly: “The CPU and FPGA execute as two independent execution units.” EX1026, 82. *See also id.*, 89 (“Datapath mode offers the best potential performance ... as the CPU and FPGA execute as two independent units”); *id.*, 80 (“Crucial to the

effective utilisation of the FPGA is the consideration given to ... exploitation of **parallelism both within the FPGA** as well as **between CPU and FPGA.**”):

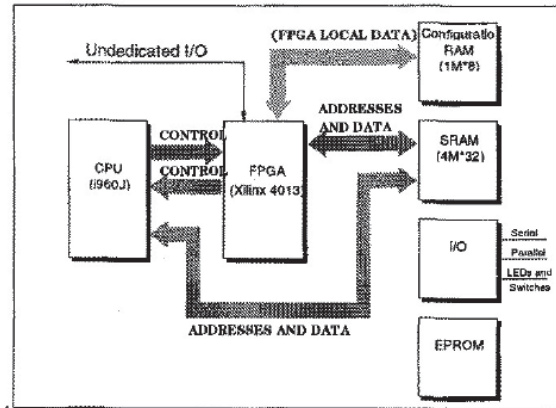


Fig. 4: Datapath mode

151. The do see that the Board already noted Sentient’s “selective” recitation of Sentient’s arguments. ID, 34:

In so doing, Patent Owner selectively reproduces two figures in Kostarnov that illustrate two modes of operation—i.e., functional unit mode and lockstep mode—in which instructions flow from the CPU to the FPGA. Id.; Ex. 1036, 3–4. However, Patent Owner omits a third disclosed mode of operation—i.e., datapath mode—which is the mode of operation upon which Dr. Shanfield’s testimony relies, as discussed above. Ex. 1002 ¶ 83 (citing Ex. 1036, 4). Because Patent Owner’s arguments neglect the teachings upon which Dr. Shanfield and Petitioner rely, and instead are based on disclosure of unrelated modes of operation, the arguments are not responsive and do not undermine either Petitioner or Dr. Shanfield.

152. In fact, I see that Sentient concedes that Kostarnov illustrates exactly the independence that I described in my earlier report. POR, 51. The paper was published

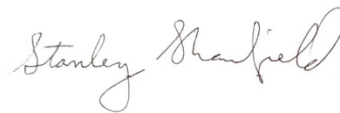
by Springer, a respected long-time publisher of the vein of IEEE or ACM, in its Lecture Notes in Computer Science series and was in the same area as the '177 patent invention. This paper reflects common knowledge of the POSITA in December, 2001 and would have been available and relevant as the POSITA considered the disclosures in the asserted art.

ADDITIONAL REMARKS/CONCLUSION

153. I currently hold the opinions expressed in this declaration. But as I understand briefing has not finished in this matter, my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

154. 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.

3/21/24



Dated: _____

By: _____