Chris Scott Graham (State Bar No. 114498)
chris.scott.graham@dechert.com
Michael N. Edelman (State Bar No. 180948)
michael.edelman@dechert.com
DECHERT LLP
1117 California Avenue
Palo Alto, CA 94304-1106
Telephone: 650.813.4800
Facsimile: 650.813.4848

Attorneys for Plaintiff and Counter-Defendant,
SYNOPSYS, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

SYNOPSYS, INC., a Delaware corporation,

Plaintiff and Counter-Defendant,

v.

MAGMA DESIGN AUTOMATION, a Delaware corporation,

Defendant and Counter-Claimant.

AND RELATED COUNTER-CLAIMS

Case No. C-04-03923 MMC {JCS}

DECLARATION OF MATTHEW HERNDON, PH.D., IN SUPPORT OF PLAINTIFF SYNOPSYS' OPPOSITION TO MAGMA DESIGN AUTOMATION, INC.'S MOTION FOR SUMMARY JUDGMENT AS TO THE SECOND THROUGH SIXTH CAUSES OF ACTION IN THE SECOND AMENDED COMPLAINT

Date: July 29, 2005
Time: 9:00 a.m.
Courtroom: 7, 19th Floor
Judge: Hon Maxine M. Chesney

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

I, MATTHEW HERNDON, declare:

1. I have been retained as an expert consultant by the law firm of Dechert LLP ("Dechert"), counsel of record for plaintiff Synopsys, Inc. ("Synopsys") in the above-captioned matter. The following declaration is based on my personal knowledge. If called upon to testify, I could testify competently as to the matters set forth herein.

I. Professional Background.

2. I am currently a Signal Integrity Engineer at Apple Computer, Inc. ("Apple"), located in Cupertino, California. I have worked for more than 20 years in the field of Electronic Design Automation ("EDA"), which generally encompasses the technology at issue. My qualifications and experience described below indicate my expertise in the EDA field. A true and correct copy of my resume is attached hereto as Exhibit A.

3. In 1976, I received a BSEE from the University of California, Santa Barbara. From 1976 to 1981, I attended Stanford University, where I received an MSEE (1977) and a Ph.D. in electrical engineering (1981). I remained at Stanford University until 1982 as part of the research staff.

4. From 1982 to 1983, I worked at Comsat General Integrated Systems, Inc. as a software engineer. I was responsible for developing software used in designing circuits employed in microwave applications. I gained valuable experience in developing software for circuit designers and further developed my understanding of the electrical engineering concepts involved in circuit design.

5. In 1983, I was one of the founding members of Analog Design Tools, Inc. (hereinafter "Analog"). At Analog, I developed one of the earliest mouse-driven Graphic User Interfaces (well known in the industry as "GUIs"). The product that implemented this technology rapidly then became an industry leader. Among other contributions, I successfully led a group of engineers and scientists to dramatically improve the performance of one of Analog's circuit design software products. As a result of my team's efforts, the software tool enjoyed a wide level of industry acceptance.

///

6. In 1990, I left Analog and, along with others, founded a company, Performance Processors, where I developed the next generation software for solving certain problems encountered during circuit design.

7. Having gained extensive expertise in developing software used in circuit design, I founded my own consulting company in 1994. For about the next year, I continued to successfully provide my expertise in software and circuit design technology implemented in a variety of applications.

8. In 1995, I joined Synopsys, Inc. ("Synopsys") as a Senior Software Engineer on a joint development effort with Hitachi (Japan) to develop a new circuit design tool. This tool integrated the circuit design concepts of two independent tools, one sold by Synopsys and the other sold by Hitachi. At Synopsys, my duties required an in-depth understanding of different types of design flow, including synthesis and other EDA concepts.

9. I left Synopsys in 1996 and joined Power Design Tools, Inc, which later changed its name to Transim Technologies, Inc. ("Transim,"). At Transim, I embarked on the then-novel idea of offering to circuit designers certain design software tools over the World Wide Web ("web"). I left Transim in 2001 to independently further develop circuit design software tools for a year. During this period, I developed software for the Apple computer that distributed circuit design capabilities over a network of servers.

10. Apple hired me in 2001 in my current position, where I work on next generation circuit design concepts and use advanced EDA software tools. Such concepts drive innovative integrated circuit and system designs which will be used in the manufacture of future generation Apple products.

11. In my current position, I have analyzed delays attributed to buffers, logic gates and other circuit components on integrated circuits and computer systems. Such analysis includes timing of electronic systems and is very analogous to the integrated circuit timing concepts, such as slack time and timing constraints, described in Synopsys draft patent application, which is discussed in greater detail below.

///

12. My extensive experience in the EDA industry has required expertise in advanced electrical engineering concepts and EDA software tools. I believe that my significant contributions have advanced the state of the art in EDA.

II. Summary of Opinions.

13. I have reviewed the Declaration of Dr. Carl Sechen in Support Of Magma's Motions For Summary Judgment And In Opposition To Synopsys' Motions For Partial Summary Judgment ("Sechen Declaration"), as well as the exhibits and the documents referenced therein. I have also reviewed Magma's provisional patent application no. 60/068,827 (the "'827 provisional"), which is attached hereto as Exhibit B.

14. I have reviewed the Declaration of Dr. Narendra Shenoy in Support Of Synopsys' Opposition to Magma's Motions For Summary Judgment ("Shenoy Declaration") and the Declaration of Dr. Robert Damiano in Support Of Synopsys' Opposition to Magma's Motions For Summary Judgment ("Damiano Declaration").

15. Based upon review of the above-identified documents and my expertise in EDA, I disagree with Dr. Sechen's analysis and conclusions in the following respects:[1]

A. Dr. Sechen's "decomposition" of the information in Synopsys' draft patent applications is erroneous.

B. Dr. Sechen's opinion that the disclosures made by Magma about its products to Synopsys before September 2000 disclose all of the information in Synopsys' draft patent applications is clearly erroneous. Rather, as properly identified, information contained in the Synopsys documents was not contained in Magma's disclosures.

C. From my review of Magma's disclosures, it would be unreasonable to contend, from the vantage point of a skilled EDA artisan or a reasonable Synopsys engineer, that Synopsys had reason to suspect any theft or misappropriation by Magma.

///

///

[1] Dr. Otten is wrong in his opinion that Otten's ICCAD tutorial discloses the concept of stretching. Based on my review of Otten's ICCAD tutorial, it is my opinion that this tutorial does not disclose the concept of stretching.

D. Under the circumstances, it would be reasonable for a skilled EDA artisan or a reasonable Synopsys engineer to believe, from reviewing Magma's disclosures, that Magma had independently developed its technology from the teachings in the public domain.

III. Analysis.

16. I was requested to review the Sechen Declaration and exhibits in order to analyze two different issues: (1) the extent to which Magma's disclosures to Synopsys about its products contained information taken from Synopsys' draft patent application, and (2) if so, whether one with ordinary skill in the art (either a skilled EDA artisan or a reasonable Synopsys engineer) would have reason to suspect, under all the circumstances, that Synopsys confidential information had been wrongfully taken by Magma. My analysis of these two issues follows.

A. Dr. Sechen's Identification of the Information in the Synopsys' Materials Is Erroneous.

17. Dr. Sechen's analysis is predicated on an identification of eleven "concepts" that he contends are contained in Synopsys' draft patent application. Sechen Declaration, ¶¶ 14-24. This identification is extremely flawed. I have identified two threshold problems with Dr. Sechen's declaration, which appear to infect the entirety of his analysis.

18. First, Dr. Sechen's comparison between the Synopsys patent application and the Magma materials assumes: (a) that the Magma disclosures should include the slide presentation of Patrick Groeneveld at the June 2000 DAC conference (Exhibit F to the Sechen Declaration); and (b) that the public disclosures should include a document authored by van Ginneken *et al.*, entitled "Size Independent Synthesis," (Exhibit VV to the Sechen Declaration). However, I do not believe that either of these materials are relevant.

19. Based on my review of the Shenoy Declaration and Damiano Declaration, I understand that the paper entitled "Size Independent Synthesis" was never submitted for review, much less published. I have not been presented with any information to indicate that this paper was published. Nor have I been presented with any information to indicate that Magma disclosed to Synopsys the fact that it would be using this paper to develop its technology. Accordingly, I do not believe this paper is relevant to the ultimate question Dr. Sechen is addressing, *i.e.*,

whether Magma's disclosures would have given Synopsys reason to suspect that Magma was using Synopsys' confidential information.

20. I further understand from the Damiano Declaration that, to the best of his knowledge, no one from Synopsys attended the Groeneveld slide presentation at the June 2000 DAC conference (Exhibit F to the Sechen Declaration). I have not been presented with any specific information to indicate that anyone from Synopsys was present at this particular slide presentation. Again, therefore, I do not believe that this presentation is relevant to the question Dr. Sechen is addressing.

21. With the above understanding, the proper comparison is between Synopsys' draft application (Exhibit TT to the Sechen Declaration) and the disclosures that were actually made to Synopsys about Magma's technology (Exhibits A, B, C, D, E, G and K to the Sechen Declaration) (referred to hereinafter as the "Magma Disclosures"). It should be noted that, for the reasons discussed below in Section E, my overall conclusion that a reasonable Synopsys engineer would not have reason to suspect theft would not change even if my understanding regarding Exhibits F and VV turned out to be incorrect.

22. Second, Dr. Sechen's analysis makes the critical mistake of wrongly defining the information contained in Synopsys' draft applications. Dr. Sechen's conclusions are predicated on the erroneous belief that "all of the information disclosed in the Synopsys materials can be decomposed into . . . 11 (eleven) different concepts and techniques." (Sechen Declaration, ¶ 13.) I do not understand how Dr. Sechen could arrive at this conclusion, and Dr. Sechen does not explain how he does so. Though Dr. Sechen states that he parsed the Synopsys draft patent application into these 11 isolated concepts, he does not explain how he did this parsing or how he decided which information from Synopsys' draft patent application to include in his identification. Further, Dr. Sechen does not explain the reason why he decided to lump huge portions of the application into eleven concepts, rather than specifically addressing all the portions of the application.

23. One of the major problems with Dr. Sechen's "parsing" approach is that it results in a series of concepts each being identified in isolation. However, the Synopsys draft application

does more than describe EDA concepts in isolation. Rather, the application describes specific integrations of a series of EDA concepts, with these integrations defining a specific flow for logic synthesis, placement and/or routing. In the flows discussed in the Synopsys application, the individual EDA concepts interrelate with each other in a novel way, which provides an integrated, automated solution to cell placement in the design of integrated circuits.

24. This point is recognized in Magma's U.S. Patent No. 6,453,446 (the "`446 Patent") and U.S. Patent No. 6,725,438 (the "'438 Patent"), significant portions of which were copied from the Synopsys draft patent application. In describing why the inventions in those claims were patentable, Magma repeatedly indicated that, although certain isolated concepts (such as constant delay) had been disclosed in the prior art, the inventions were novel because they placed those concepts together into an integrated, automatic solution for cell placement:

> In Sutherland and R. Sproull, "The Theory of Logical Effort: Designing for Speed on the Back of an Envelope", Advanced Research in VLSI, pp. 3-16, University of California, Santa Cruz, 1991, the delay is noted as being dependent on gain. A size independent formulation of the delay optimization problem is also presented, but the solution is intended for use as imprecise, scratch-pad type calculations, and not part of ***an overall integrated, automated solution to cell placement in the design of integrated circuits***.

(*See* Declaration of Peter Obstler in Support Of Magma's Motions For Summary Judgment And In Opposition To Synopsys' Motions For Partial Summary Judgment ("Obstler Declaration"), Exh. AAA, '446 Patent, at 2:57-65 (emphasis added).) Dr. Sechen's analysis completely overlooks the necessity of providing an integrated, automated solution to cell placement. By focusing only on isolated concepts, Dr. Sechen's analysis misses the novelty of the information contained in Synopsys' draft patent application.

25. For instance, Dr. Sechen's first identified category is "constant delay." However, in the '446 Patent, Magma itself acknowledged that the bare concept of constant delay was in the public domain. (Obstler Declaration, Exh. AAA, `446 Patent at col. 2, lines 31-39.) Synopsys' applications did not merely describe constant delay, but rather described a complex interrelationship between constant delay and a series of other EDA concepts.

///

B. A Proper Identification of Information in Synopsys' Application.

26. Since Dr. Sechen's identification of the information in Synopsys' draft patent application is erroneous, I have reviewed the application, and identify and discuss below the Synopsys information I have located in this application. My method in identifying this information was to carefully review the Synopsys documentation in order to locate that information that Synopsys apparently believed was novel or unique, *i.e.*, separate from information already existing in the public domain. To do this, I reviewed both the specification of the draft patent application, as well as the claims recited therein, and eliminated information (for instance, the bare concept of constant delay) that was not identified or could not fairly be identified in the Synopsys application as being something new.

It should be noted that the following identification is my own opinion based on my review to date of Synopsys' application; it is my understanding that Synopsys has not itself been asked to provide such a list of information to Magma for purposes of this litigation, and accordingly such a list may differ from mine.

27. **Integration of Concepts.** As described above, it is not merely a series of isolated concepts that were provided in the Synopsys' application, but also the way in which those concepts interact to define an overall integrated flow. The Synopsys draft patent application describes different integrations of EDA concepts to provide a specific flow – an automated solution to cell placement in the design of integrated circuits. For instance, the Synopsys draft patent application describes integration of the following concepts as part of an automated solution to cell placement:

1. The concept of size independent synthesis – the structuring of a digital network in a particular manner (*see* further description below).

2. The concept of size independent synthesis via optimal gain -- selecting an optimal delay gate in the context of size independent synthesis (*see* further description below).

3. The concept of continuous cell sizing in the context of size independent synthesis- using actual or idealized cells having sizes that may be adjusted continuously in the context of size independent synthesis (*see* further description below).

4. The concept of area estimation – computing a weight for each net and using those net weights and wire lengths to estimate circuit area in the context of size independent synthesis (*see* further description below).

5. The concept of stretching and compressing delays in the context of size independent synthesis – increasing and decreasing the delay for gates during logic or physical synthesis in the context of size independent synthesis (*see* further description below).

6. The concept of sizing gates iteratively – computing sizes of gates in an iterative manner to resolve the cyclic dependencies in a sequential circuit (*see* further description below).

7. The concept of boundary movement using constant delay – a "boundary move" transformation for reducing the number of levels in the logic in the mapped network using constant delay (*see* further description below).

8. The concept of target delay placement – a method for the placement and sizing of cells of a mapped digital network (*see* further description below).

9. The concept of selecting delay based on gain in the context of size independent synthesis.

10. The concept of selecting delay by considering input transition time in the context of size independent synthesis.

11. The concept of buffer insertion in the context of size independent synthesis.

12. The concept of buffer insertion for saving area in the context of size independent synthesis.

13. The concept of buffer insertion for saving area in the context of size independent synthesis, where area savings is estimated using net weights.

14. The concept of net weight calculation in the context of size independent synthesis.

15. The concept of stretching delay in the context of size independent synthesis.

16. The concept of compressing delay in the context of size independent synthesis.

17. The concept of stretching in the context of size independent synthesis, where cells are adjusted equally among the stages which have the same slack.

18. The concept of compressing in the context of size independent synthesis, where cells are adjusted equally among the stages which have the same slack.

19. The concept of stretching in the context of size independent synthesis, where the slack on each path become zero.

20. The concept of compressing in the context of size independent synthesis, where the slack on each path become zero.

21. The concept of area estimation in the context of size independent synthesis.

22. The concept of area estimation in the context of size independent synthesis and performed by traversing the network in the direction opposite to data flow.

23. The concept of area estimation in the context of size independent synthesis and performed by traversing the network in the direction opposite to data flow, where the network has loops.

24. The concept of area estimation in the context of size independent synthesis and performed by traversing the network in the direction opposite to data flow, where the traversal is started at primary outputs and registers.

25. The concept of using network area as another optimization goal in the context of size independent synthesis.

26. The concept of retiming and moving registers in the context of size independent synthesis.

27. The concept of mapping in the context of size independent synthesis, where the network is traversed twice to select the appropriate matching books.

28. The concept of performing placement and sizing in gradual steps in the context of target delay placement.

29. The concept of performing placement and sizing in gradual steps such that placement is performed by repeated partitioning in the context of target delay placement.

30. The concept of performing placement and sizing in gradual steps with placement optimization by repeatedly changing the location of one or two cells at a time in the context of target delay placement.

31. The concept of calculation of net length, capacitive load and sizes in the context of target delay placement.

32. The concept of calculation of sizes by traversing the network in the direction opposite to the data flow in the context of target delay placement.

33. The concept of calculation of sizes in traversing the network in the direction opposite to the data flow, where the network has loops in the context of target delay placement.

34. The concept of calculation of sizes in traversing the network in the direction opposite to the data flow, where traversal starts at primary outputs and registers in the context of target delay placement.

35. The concept of layout following target delay placement.

36. The concept of calculation of net length, capacitive load and sizes in the context of target delay placement that consists of selecting the most suitable size from a limited set of available sizes.

In my opinion, the integration of these concepts into an overall solution for cell placement was an aspect of the novelty of the information described in the Synopsys draft patent application.

28. **Representative Individual Concepts.** By way of example, I describe below a representative sample of the concepts disclosed in the Synopsys application that are integrated into the overall solution.

a. Size independent synthesis is the concept of structuring of a digital network, using network slack as an optimization goal and mapping of the cells of the network, wherein network slack is calculated assuming that the sizes of the cells are not fixed. This concept is reflected at a high level in the following passage:

> The present invention overcomes the problems of the conventional approach by not choosing a size for a cell at all. Rather then choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

(Sechen Declaration, Exh. TT at SY004570). The Synopsys draft patent application more particularly discloses this concept in the following section:

> In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:
>
> a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
>
> b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
>
> c) estimation of the area of the network based on net load

(Sechen Declaration, Exh. TT at SY004571).

b. Size independent synthesis via optimal gain is the concept of selecting an optimal delay for each gate in the context of size independent synthesis as described above. The Synopsys draft patent application discloses this concept in the following passage:

> Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S, which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is S*A and the delay of a cell is D = f(C/S). The library analysis will determine a good value for C/S for

> each cell in the library. Using this value, it determines a constant delay for each gate.

(Sechen Declaration, Exh. TT at SY004577). The Synopsys draft patent application also discloses this concept in the following passage:

> 5. The method of claim 3, where a parameter C/S for each book is chosen to have the largest possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the parameter C/S is chosen optimally after adding the buffer.

(Sechen Declaration, Exh. TT at SY004585-SY004586).

c. **Continuous cell sizing in the context of size independent synthesis** is the concept of using actual or idealized cells having sizes that may be adjusted continuously, in the context of size independent synthesis as described above. The Synopsys draft patent application discloses this concept in the following passage:

> Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S, which increases both the load driving capability of the cell and the area linearly.

(Sechen Declaration, Exh. TT at SY004577).

d. **Area estimation in the context of size independent synthesis** is the concept of computing a weight for each net and using those net weights and wire lengths to estimate circuit area, in the context of size independent synthesis and placement. The Synopsys draft patent application discloses this concept in the following passage:

> (net weights)
>
> Various algorithms, such as buffering and placement optimize the network by manipulating the loads in the network. Placement optimizes the net length, which directly related to the net load, and buffering reduces the load on a cell by adding extra delay. These algorithms can benefit from a more efficient calculation of the effect that changing the load of a cell has on sizing. We can do this by calculating single parameter per net, called the net weight, which represents the sensitivity of the total area of the network with respect to the load on that net. This net weight can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at the primary inputs, the net weight of the first (left most) gate is equal to its area per unit load. The

> net weights of the other cells can now be calculated with a recurrence relation traversing the network from left to right..... to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section. Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(Sechen Declaration, Exh. TT at SY004577).

e. Stretching and compressing delays in the context of size independent synthesis is the concept of increasing and decreasing the delay for gates during logic or physical synthesis, in the context of size independent synthesis as described above. The Synopsys draft patent application discloses this concept in the following passage:

> (stretching)
>
> The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "Stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.
>
> Note that this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process. The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate
>
> The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater then 0. (All other paths now have a slack of 0).

(Sechen Declaration, Exh. TT at SY004583-84).

///

///

f. **Sizing gates iteratively** is the concept of computing sizes of gates in an iterative manner to resolve the cyclic dependencies in a sequential circuit. The Synopsys draft patent application discloses this concept in the following passage:

> To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a levelized order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small. It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to be smaller then a preset maximum after a fixed number of iterations. The iteration does not converge if the network is an infeasible solution: The current network cannot be expected to work at this speed because its gain is too small. Changes need to be made to the network to increase the gain, which will usually mean increasing the delay of the network as well.

(Sechen Declaration, Exh. TT at SY004580-SY004581.)

g. **Boundary movement using constant delay** is the concept of a "boundary move" transformation for reducing the number of levels in the logic in the mapped network using constant delay. The Synopsys draft patent application discloses this concept in the following passage:

> (structuring boundary move)
>
> Using constant delay it is considerably easier to predict the effect of a change to the network then with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the

> transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.
>
> In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that is affects delay, is the change of the faning of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(Sechen Declaration, Exh. TT at SY004579-SY004580.)

h. Target delay placement is the concept of a method for the placement and sizing of cells of a mapped digital network, including: choosing a target delay for each cell, computing the network slack using the target delays, placing the cells, and sizing the cells such that the network meets the computed network slack. The Synopsys draft patent application discloses this concept in the following passage:

> 20. A method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:
>
> d) Choosing a target delay for each cell.
>
> e) Computing the network slack using the target delays.
>
> f) Placement of the cells of the network.
>
> g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

(Sechen Declaration, Exh. TT at SY004588.)

29. **Copied Information.**

There is a significant amount of other information that was copied from Synopsys' application to Magma's patents. This information discusses the background and understanding of various EDA concepts and why they are important to arriving at the inventions contained therein. This type of information can be very valuable for a competitor to have because it concentrates on information that could be utilized to file a patent application, or that could be utilized to quickly bring other engineers up to speed on the problems that certain techniques are intended to solve and therefore jumpstart a software development process. These background passages were copied at great length for Magma's patents. For instance, attached hereto as Exhibit C is a table of passages that were copied into the Magma patents, but which were not included in the Magma Disclosures.

C. The Magma Disclosures Do Not Indicate Use of the Information in Synopsys' Application.

30. After identifying the information from Synopsys' materials discussed above in Section B, I then reviewed the Magma Disclosures to determine whether that information was described in those materials. I was unable to find disclosure of this information in any of the Magma Disclosures. It is my opinion that the Magma Disclosures do not disclose these concepts. Further, even if the Magma Disclosures identified one or more of the individual concepts, a reasonable person in Synopsys' position would not have been suspicious of theft unless, at minimum, the integration of several of these concepts was presented as part of an integrated EDA flow.

31. In my opinion, the Magma Disclosures fail to disclose the concepts I identified in the Synopsys draft patent application (*see* section (B) above) primarily because these materials were designed as high level marketing presentations for Magma products. As such, it is not at all surprising that the Magma Disclosures are greatly lacking in the type of technical detail that would be expected in a technical publication. While the Magma Disclosures present some isolated concepts, it is nearly impossible to determine from these documents how or if Magma proposed to integrate these concepts into an integrated EDA flow. The Magma Disclosures contain, for instance, practically no information on the algorithms Magma intended to use for logical and physical synthesis. Also, because many of the individual concepts contained in the Magma Disclosures were found in the public domain (*e.g.*, "constant delay"), the isolated references in the Magma Disclosures to certain generic concepts fall far short of disclosing the automated solution to cell placement developed at Synopsys. In fact, even concepts taken in isolation are not disclosed in the Magma Disclosures.

32. For instance, the concept of size independent synthesis is not disclosed in the Magma Disclosures. There is no description anywhere in the Magma Disclosures of "structuring … and mapping of the cells of the network, wherein network slack is calculated assuming that the sizes of the cells are not fixed."

///

33. Similarly, the concept of target delay placement is not disclosed in the Magma Disclosures. There is no description anywhere in the Magma Disclosures of "computing the network slack using the target delays."

34. The concept of sizing gates iteratively is not present because the Magma Disclosures do not disclose computing sizes of gates in an iterative manner to resolve the cyclic dependencies in a sequential circuit.

35. The concept of boundary movement using constant delay is not present because the Magma Disclosures do not disclose a "boundary move" transformation for reducing the number of levels in the logic in the mapped network.

36. The concept of selecting delay by considering input transition time in the context of size independent synthesis is not present because the Magma Disclosures do not disclose that delay may be different for different input transition times.

37. The concept of buffer insertion for saving area in the context of size independent synthesis, where area savings is estimated using net weights, is not present because the Magma Disclosures do not disclose that area savings due to buffer insertion can be estimated using net weights.

38. The concept of stretching in the context of size independent synthesis, where cells are adjusted equally among the stages which have the same slack is not present because the Magma Disclosures do not disclose that the delay of each cell may be adjusted equally among stages which have the same slack. For the same reason, the concept of compressing in the context of size independent synthesis, where cells are adjusted equally among the stages which have the same slack, is also not present.

39. The concept of area estimation in the context of size independent synthesis being performed by traversing the network in the direction opposite to data flow is not present because the Magma Disclosures do not disclose that area estimation may be performed by traversing the network in the direction opposite to data flow.

40. The concept of area estimation in the context of size independent synthesis being performed by traversing the network in the direction opposite to data flow, where the traversal is

started at primary outputs and registers is not present because the Magma Disclosures do not disclose that the traversal is started at primary outputs and registers.

41. The concept of using network area as another optimization goal in the context of size independent synthesis is not present because the Magma Disclosures do not disclose that network area can be an optimization goal in the context of size independent synthesis.

42. The concept of mapping in the context of size independent synthesis, where the network is traversed twice to select the appropriate matching books, is not present because the Magma Disclosures do not disclose traversing a network more than once to select matching books.

43. The concept of performing placement and sizing in gradual steps in the context of target delay placement is not present because the Magma Disclosures do not disclose that sizing may be performed along with placement in gradual steps. For at least the same reason, the Magma Disclosures do not disclose the concepts of performing placement and sizing in gradual steps such that placement is performed by repeated partitioning in the context of target delay placement or performing placement and sizing in gradual steps with placement optimization by repeatedly changing the location of one or two cells at a time in the context of target delay placement.

44. The concept of calculation of sizes in traversing the network in the direction opposite to the data flow, where the network has loops in the context of target delay placement is not present because the Magma Disclosures do not take note of the fact that in a sequential network, there may be one or more loops, resulting a cyclic dependency. As a result, it is clear that the Magma Disclosures do not disclose this concept.

D. Even the Categories of Information Identified by Dr. Sechen Are Not Disclosed in The Magma Disclosures.

45. As explained above, Dr. Sechen's identification of the information in Synopsys' application is erroneous. However, I nevertheless conducted a review of the Magma Disclosures to determine whether, even if the identification was proper, this information could be found in the

///

Magma Disclosures and, if so, whether this information was presented in adequate detail such that it would apprise a reasonable Synopsys engineer of Magma's theft.

46. As explained below, the relevant Magma Disclosures either did not disclose the concepts identified by Dr. Sechen, or contained such a cursory disclosure compared to the detail of the Synopsys' application that it would not be reasonably possible for a reasonable Synopsys engineer to suspect theft.

a. With respect to Dr. Sechen's concept of "Constant Delay," it is my understanding there is no dispute between the parties that this bare concept was already in the public domain. (*See* Sechen Declaration at ¶ 52). Accordingly, I do not believe that the fact that this concept appears in the Magma Disclosures could possibly lead a reasonable Synopsys engineer to believe that Magma had misappropriated information from Synopsys.

b. With respect to Dr. Sechen's concept of "Constant Delay Synthesis," this concept was not disclosed in sufficient detail in the Magma Disclosures to lead a reasonable person in Synopsys' position to suspect that the origin of the information was Synopsys property. Indeed, the Magma Disclosures only disclose this concept, if it is disclosed at all, in isolated bits and pieces, and not as part of an overall, integrated solution for cell placement such as is set forth in the Synopsys' materials. As just one example, the Magma Disclosures do not disclose constant delay synthesis using network slack as an optimization criterion, as described in the Synopsys application. Without further details about how constant delay synthesis (as inadequately defined by Dr. Sechen) is utilized by Magma, I do not believe it would be reasonable to contend that a Synopsys engineer had reason to suspect theft.

c. With respect to Dr. Sechen's concept of "Buffer Insertion for Area Minimization," this concept was only mentioned in one line in a single slide. Sechen Declaration, Exh. A at p. 14. All this line stated was "inserting buffers saves area." *Id.* I do not believe that this is an adequate description of this concept. There is no description of the origin of any information pertaining to buffering, how buffering would save area, or how buffering would fit within a constant delay paradigm. Unless Synopsys was provided far more detailed information

/ / /

about how buffer insertion would work, I do not believe it would be reasonable to contend that a Synopsys engineer had reason to suspect theft.

d. With respect to Dr. Sechen's concept of "Sizing Driven Placement," this concept was not disclosed in sufficient detail in the Magma Disclosures to lead a reasonable person in Synopsys' position to suspect that Magma misappropriated information related to this concept from Synopsys. Rather, the Magma Disclosures only disclose this concept, if at all, in isolated sound bites, without any context provided which would be sufficient to match up Magma's information with Synopsys' patent application. As an example, one of the Magma Disclosures simply states "need sizing driven placement." (Sechen Declaration, Exh. A at 12.) This statement would lead a reasonable person to believe that Magma had yet to develop any particular algorithms for sizing driven placement, much less the particular integrated approach developed at Synopsys.

e. With respect to Dr. Sechen's concepts of "Net Weight Placement," "Continuous Gate Sizing," and "Discrete Gate Sizing," these concepts were not disclosed in detail in the Magma Disclosures. Unless Synopsys was provided far more detailed information about how these concepts at Magma would work, I do not believe it would be reasonable to contend that a Synopsys engineer had reason to suspect theft. Indeed, the Magma Disclosures provide so little information on these concepts that it would have been entirely reasonable for Synopsys to conclude that what Magma was developing with regard to these concepts was no different than what was in the public domain.

f. With respect to Dr. Sechen's concept of "Stretching Constant Delays," this concept was also not disclosed in detail in the Magma Disclosures. Unless Synopsys was provided far more detailed information about how this concept at Magma would work, I do not believe it would be reasonable to contend that a Synopsys engineer had reason to suspect theft. The passages in the Magma Disclosure that Dr. Sechen identifies in connection with this concept merely refer to "delay trimming." Sechen Declaration, ¶ 36, citing Exh. A. A reasonable person in Synopsys' position in 1998 would not have understood the vague reference to "delay trimming" to refer to the concept of "Stretching Constant Delays." Further, even if the bare

concept of stretching was introduced, the absence of any detail about how the stretching was to occur or the context in which it was to occur would be significant; in fact, the absence of such details would indicate to a reasonable engineer in Synopsys' position that Magma had not misappropriated Synopsys' information and therefore did not have access to such details.

47. For the reasons set forth above, even if Dr. Sechen's identification of the information in Synopsys' application was accepted, the Magma Disclosures either did not disclose these concepts or contained such a cursory disclosure compared to the detail of the Synopsys' application that it would not be reasonably possible for a reasonable engineer in Synopsys' position to suspect theft.

E. Synopsys Did Not Have Any Reason to Suspect Theft of Its Confidential Information Based On Information in the Public Domain.

48. In my expert opinion from my years of experience in the EDA field, it would be completely unreasonable to contend that Synopsys should have suspected that its confidential patent application was stolen or plagiarized by Magma. My understanding is that the determination of whether Synopsys should have suspected wrongful conduct is based on the perspective of a "reasonable person" in Synopsys' position. As someone who both had years of experience in EDA and has worked at Synopsys, I believe I am qualified to opine on what a reasonable engineer at Synopsys would or would not have had reason to suspect under these circumstances.

49. First, in my opinion, it would have been extraordinarily difficult for a reasonable engineer at Synopsys to have suspected theft, given the content of the Synopsys application and the Magma disclosures. The Synopsys application is very long, and was still in draft form. A comparison of this complicated document to Magma's disclosures simply to determine whether confidential information was present would have been an extremely difficult task. The Magma Disclosures did not contain verbatim copying of this document as did Magma's `446 and `438 patents, and therefore finding a reason to suspect theft, even if the Synopsys application was studied at length, would also be difficult.

///

50. For instance, here is a comparison of Claim 1 of the Synopsys draft patent application to the closest passage in Magma's literature that I could find:

Synopsys Application	Passage in Magma's Disclosure
A method for the structuring and mapping of an unmapped digital network comprising the following steps: a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network tis [sic] constant with respect to load. b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load. c) estimation of the area of the network based on net load.	Magma's No. 1999 white paper states: Turning around the conventional order of gate sizing and delay has additional advantages. Instead of a fluctuating delay, Magma's FixedTiming methodology changing cell sizes [sic]. Some cells will become larger (to drive a long wire), while others become smaller (if they only need to drive a neighboring input). These differences will cancel out, because area is additive. Timing, on the other hand, is determined by the worst case path delay across the entire circuit. Sechen Declaration, Exh. C at p. 5

These passages contain entirely different language, and it would not be apparent to even a studious observer that the passage on the right had been derived from any portion of the passage on the left. Even if it were theoretically feasible for a Synopsys engineer to perform claim interpretation on Synopsys' draft claim and determine that the invention described in the claim (recited in the left hand side of the above chart) was contained or suggested in Magma's disclosure (recited in the right hand side of the above chart), it would be completely unreasonable to contend that it would occur to an engineer to do so. As the above reflects, the language utilized in Magma's disclosures and the manner in which its technology was described was so different that Synopsys would have no reason to suspect theft, absent further evidence (such as verbatim copying) that could have triggered a suspicion.

51. Second, the manner in which the Magma Disclosures appeared would have immediately dissuaded a reasonable person at Synopsys from suspecting theft. The Magma Disclosures emphasized that Magma's constant delay technology had been independently developed by its own engineers. The Magma Disclosures also pointed out that Magma's approach would be different from Synopsys' approach. (*See, e.g.*, Sechen Declaration, Exh. C at

15 stating "Blast Fusion's patent pending FixedTiming methodology is significantly different from the approaches employed by Cadence! or Avant! (or even Synopsys' new design planning entry, Chip Architect).")

52. Third, the information presented in Magma's Disclosures repeatedly emphasized that the basic ideas that Magma started with came from the public domain. For instance, Magma's 1999 white paper makes clear that Magma's FixedTiming methodology was derived from the theory of logical effort, which was described by Ivan Sutherland in an article published in 1991. (Sechen Declaration, Exh. C at 5 and 17 (citing Sutherland and Sproull, "Logical Effort: Designing for Speed On the Back of an Envelope," invited paper, ICCAD, 1991).) Since the basic concept of constant delay was in the public domain, and I understand that Synopsys had itself used this concept to come up with its own approach, it would have been perfectly reasonable for Synopsys to believe that Magma had done so as well.

53. I do not understand how Dr. Sechen could disagree with this point, given the content of his declaration. If in fact Dr. Sechen is of the belief that the information contained in the Synopsys application was in the public domain, then there would be no reason for Synopsys to suspect that Magma came into that information by wrongful means. Rather, under Dr. Sechen's analysis, the only natural thing for Synopsys to think would be that Magma had developed its technology using information from the public domain. I do not understand how Dr. Sechen can state, on the one hand, that the information contained in the Synopsys application was in the public domain and then opine, on the other hand, that Magma's disclosure of their use of those concepts should have led Synopsys to believe that the information was stolen rather than taken from the public domain.

54. Fourth, the information in Magma's Disclosures did not include text copied from Synopsys' documents, as did Magma's patent applications, and did not provide a level of detail that might have been sufficient to give Synopsys a reason to suspect theft. For instance, below is a table showing: (1) text found in Synopsys' original draft patent application (Sechen Declaration, Exh. TT); (2) the same text that Magma copied into its own `827 Provisional Patent Application

///

(which is attached hereto as Exhibit B); and (3) corresponding catch-phrases and buzzwords that Magma used in the Magma Disclosures (Sechen Declaration, Exhs. A, B, C, D, E, G and K).

Synopsys Draft Patent Application (Sechen Declaration, Exh. TT)	'827 Provisional Patent Application (Exh. B)	Magma Disclosure
SYNOPSYS DRAFT PAT. APP. AT 15-16 (EMPHASIS ADDED.) In ***a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost*** cell. In this case the computation ***can start anywhere in the cycle, and*** repeats ***the cycle several times, until the capacitances converge*** and the error is ***sufficiently small.*** It is possible that this iteration will ***not converge and*** that the ***capacitance*** will ***increase*** in every iteration, ***by progressively larger amounts***. This situation is ***detected*** by requiring the increment to be smaller then ***a preset maximum after a fixed number of*** iterations. The iteration does ***not converge*** if the network is ***an infeasible solution***: The current network can***not*** be ***expected to work at*** this ***speed because*** its ***gain is too small. Changes*** need to be made to the network to ***increase the gain***, which ***will usually*** mean ***increasing*** the ***delay*** of the network as well.	**'827 PROV. PAT. APP. AT 39:16 - 40:17 (EMPHASIS ADDED.)** If the digital circuit is ***a sequential network*** (see, e.g., circuit 180 of FIG. 3), then ***there may be one or more loops*** (e.g., loop 182) which ***result*** in ***a cyclic dependency*** (i.e., ***there is no "rightmost"*** gate). Gate load calculation ***can start anywhere in the cycle, and*** calculation in ***the cycle*** is performed ***several times until the*** load ***capacitance*** values ***converge*** or have ***sufficiently small*** differences. However, a condition may exist when the load ***capacitance*** values do ***not converge and increase by progressively larger amounts*** every cycle calculation. This increase in load capacitance values can be ***detected*** if the calculated load values exceed ***a preset maximum*** value ***after a fixed number of*** cycle calculations. When the calculated load values do ***not converge***, then the particular circuit 180 has ***an infeasible solution***, which indicates that the digital circuit is ***not expected to work at*** the set ***speed because*** the circuit ***gain is too small. Changes*** are required to ***increase the*** circuit ***gain***, and these changes ***will usually*** lead to an increase in circuit ***delay***.	**Sechen Declaration, Exh. A at p. 12** "sizing driven placement"

Synopsys Draft Patent Application (Sechen Declaration, Exh. TT)	'827 Provisional Patent Application (Exh. B)	Magma Disclosure
SYNOPSYS DRAFT PAT. APP. AT 16 (EMPHASIS ADDED.) After the loads have been calculated the ***size*** can be calculated ***by dividing the actual load by the predetermined typical load***. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the ***size*** of the ***gate***. ***The size is a scale factor, which*** can be applied to the area of ***the gate***, to give ***the area of the sized gate***. ***The area of the*** network ***can be estimated*** as ***the sum total*** of the ***areas of the sized gates, plus the net area*** as ***estimated from the total length of all nets***.	**'827 PROV. PAT. APP. AT 40:18 – 41:12 (EMPHASIS ADDED.)** In the above example, the ***size*** S of a gate i is determined ***by dividing the actual load*** C_i ***by the predetermined typical load*** C/S of the gate i. The size S is a scale factor which is applied to all transistor channel widths of a gate in order to determine the area of the "***sized gate***". ***The size*** S ***is*** also ***a scale factor which*** is used to scale ***the gate***'s output load driving capability and its input pin loads. ***The area of the sized gate*** is determined by equation (5). area of sized gate=S*(area of gate) (5) ***The area of the*** mapped digital circuit ***can be estimated*** based on ***the sum*** of the ***total areas of the sized gates plus the net area*** (which is ***estimated from the total length of all nets*** in the circuit).	**Sechen Declaration, Exh. A at p. 12** "sizing driven placement"
SYNOPSYS DRAFT PAT. APP. AT 17-18 (EMPHASIS ADDED.) Next we have to calculate the reduction in load of this net, and check that ***area*** that is ***added*** by adding the ***buffer does not exceed the area saved by sizing down the source gate***. The ***area added by inserting the buffer is*** simply ***the area of the buffer*** times its ***size, where the size is determined by the load*** on the buffer ***divided by the typical load*** of ***the buffer. The area saved by*** **inser**ting the ***buffer*** can be calculated by first ***calculating the change in load due to the insertion*** of the buffer: ***some sinks are removed, the input load of the buffer is added, and the*** net load estimate ***may change*** as a result of the ***number of fanouts of the*** net ***chang***ing.	**'827 PROV. PAT. APP. AT 46:1-14 (EMPHASIS ADDED.)** In step 655, it is determined whether the ***added area*** due to ***buffer*** insertion ***does not exceed the area saved by sizing down the source gate***. The ***added area*** (***by inserting the buffer) is*** equal to ***the area of the buffer*** multiplied by the buffer ***size, where***in ***the buffer size is determined by the*** buffer ***load*** C ***divided by the typical load*** C/S on ***the buffer. The area saved by*** sizing down the source gate is determined by first ***calculating the change in net load due to the buffer insertion***. This ***net load change*** is due to the following: (1) ***some sinks*** (which sink currents***) are removed***, (2) ***the input load of the buffer is added, and*** (3) ***the number of fanouts of the gate may change.***	**Sechen Declaration, Exh. A at p. 14** "Inserting buffers saves area"

Synopsys Draft Patent Application (Sechen Declaration, Exh. TT)	'827 Provisional Patent Application (Exh. B)	Magma Disclosure
SYNOPSYS DRAFT PAT. APP. AT 18 (EMPHASIS ADDED.) The next step is the process of "***Stretch***ing" and "***Compress***ing" ***the delays of the individual gates to meet the*** timing ***constraints. Gates which are on long paths*** which do ***not meet the delay constraint are "compressed" until the path*** does ***meet the*** timing ***constraint. Gates on*** the ***short paths which easily meet the*** timing ***constraints are "Stretched". Gates with stretched delays require less area for the same load.*** In this step the ***delay*** of the ***gates*** is traded against the ***gain*** of the ***gates***. When ***the delay of*** a ***gate*** decreases so does the gain of the gate. It is important that there is enough gain in the network.	**'827 PROV. PAT. APP. AT 48:15 – 49:12 (EMPHASIS ADDED.)** Prior to cell placement, ***the delays of the individual gates*** may be ***stretch***ed or ***compress***ed ***to meet the*** delay ***constraints***, as shown in step 220 of FIG. 4. As shown in FIG. 9A, by compressing (decreasing) the delay of a given gate, the gate gain decreases. ***Gates which are on long paths not meet***ing ***the delay constraint***s ***are compressed*** (in delay) ***until the*** long ***paths meet the*** delay ***constraint***s. The ***delay*** of the ***gate***s (or gate) may be decreased as long as the minimum required gain requirements are met. By stretching (increasing) the delay of a given gate, the ***gate gain*** increases (see FIG. 9A). ***Gates on short paths which easily meet the*** delay ***constraints are stretched*** (in delay), since ***gates with stretched delays require less area for the same load. The delay of*** the gates (or ***gate***) in a path are stretched to the extent that timing constraints for the digital circuit are still met.	**Sechen Declaration, Exh. A at p. 15** "delay trimming"

As this table demonstrates, rather than making disclosures which might have given Synopsys a reason to suspect theft (as in the patent applications that Magma submitted to the PTO), Magma reworded the information in the Magma Disclosures and presented it at a much higher level, with much less detail, such that suspecting theft would have been practically impossible.

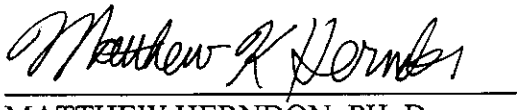
55. I strongly disagree with Dr. Sechen's statement that "Magma was very, very open with what they were working on, from presentations at conferences, trade shows, discussions with professors (such as myself)." Sechen Declaration, ¶ 55. My review of Magma's Disclosures indicate that Magma did not disclose that (a) its technology stemmed from material taken from Synopsys, (b) its technology was not independently developed by Magma engineers, (c) the ideas for its technology did not come from the foundation of public domain papers, (d) it had filed for a patent on constant delay concepts which were copied from Synopsys' documents, (e) it was using

a series of more detailed information (as described above) contained in the Synopsys application, and (f) it was using the overall, integrated solutions that were contained in Synopsys' application. In my view, Magma's disclosures omitted the information a reasonable Synopsys engineer would have needed to suspect theft, and instead included that information which would have dissuaded such an engineer from forming such a suspicion.

56. Fifth, the context in which the presentations were made would have, in my opinion, made it extremely unlikely that, even if Synopsys' information was included in the disclosures, Synopsys would have reason to suspect that the origin of that information was Synopsys property. According to Dr. Sechen, Magma disclosed eleven isolated EDA concepts, whose descriptions are strewn across seven different publications and slide presentations and which were presented at different times and in different forums before different people at Synopsys. In my opinion, it would be completely unreasonable to expect a reasonable Synopsys engineer to put all of these pieces together.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct.

Executed this 29th day of June, 2005.



MATTHEW HERNDON, PH. D.