Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 2 The major problem with the ***conventional approach*** is that the net length and hence the cell delay is not known ***until after placement***.	'446 PATENT AT 1:46-47 Thus, under the ***conventional*** design ***approach***, timing closure is not certain ***until after placement***.
SYNOPSYS DRAFT PAT. APP. AT 2-3 Before placement, ***net length*** must be ***estimated***. This is usually done with ***an estimation function or table which gives the load of a net based on*** its ***fanout***. Experience has shown that it is very difficult to ***estimate*** the length of the nets ***accurately***.	'446 PATENT AT 1:37-40 While ***net lengths*** have been ***estimated*** prior to placement by use of ***an estimation function or table which gives the load value of a net based on*** the number of ***fanout*** gates, this ***estimation*** function is usually ***inaccurate***.
SYNOPSYS DRAFT PAT. APP. AT 3 The result is unpleasant surprises ***after placement*** step 105. ***Some nets turn out to be longer than expected***, and because of the ***longer delays***, the ***timing constraints*** are not met. ***Timing closure is not certain until after*** step 105.	'446 PATENT AT 1:41:46 This difficulty in accurately predicting net lengths leads to unpredictable delay effects ***after*** cell ***placement*** occurs. For example, ***some nets turn out to be longer in length than expected***. These longer nets cause ***longer delays*** which prevent satisfaction of ***timing constraints*** in the digital circuit. Thus, under the conventional design approach, ***timing closure is not certain until after*** placement.

[1] Note that page numbers do not appear on the Draft Patent Application.

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 3 If *timing closure* is not *achieved* the options the *designer* has are *expensive* and unreliable. He may choose to *fix* the *design manually, which is difficult and time consuming, because the automatically optimized network* is hard to *understand*. He may choose to *change* his *HDL specification and repeat the* synthesis *process*. Again *timing closure will not be certain until after placement*, which means that the entire *process* needs to be traversed *before the designer* knows *if* his *HDL changes were successful*.	'446 PATENT AT 1:48-60 Failure to *achieve timing closure* after placement leads to additional *expenses* and other problems for the *designer*. To correct for failure to achieve timing closure, the *designer* has the option of *fixing the design manually, which is difficult and time consuming because the automatically optimized* digital *network* is not easy *to understand*. As a second option, the designer may *change* the Hardware Description Language *(HDL) specification and repeat the* design *process*. However, *timing closure will* again *not be certain until after placement*. Thus, the design *process* must again be repeated *before the designer* can determine *if* the *HDL* specification *changes were successful* in enabling timing closure.
SYNOPSYS DRAFT PAT. APP. AT 3 *A common method* of *dealing with inaccurate net load estimates is* to use *net load estimates* which are *considerably larger than* accurate *estimates*. This *causes the sizes of the cells to be considerably larger then necessary* but reduces the *probability of* not *meeting* the *timing constraints after placement*. Clearly using cells with *sizes which are larger then necessary* is *wasteful in both silicon area and power consumption*. The *chips* thus synthesized will be *larger*, cost *more to produce and use more electrical power then necessary*.	'446 PATENT AT 1:61-2:3 *A common method* for *dealing with inaccurate net load estimates is* by *estimating* the *net load* at a *considerably larger* value *than* typically *estimated*. Although this method increases the *probability of meeting timing constraints after placement*, it *causes the sizes of the gates to be considerably larger than necessary*. Gates *which are larger than* the *necessary size* are *wasteful in both silicon area and power consumption*. This leads to *chips* which are *larger, more* expensive *to produce, and use more electrical power than necessary*.
SYNOPSYS DRAFT PAT. APP. AT 3 A second *problem with the conventional approach* is that the effect of synthesis decisions is hard to calculate. *Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems*.	'446 PATENT AT 2:4-9 Another *problem with the conventional* circuit design *approach* concerns the timing analysis required *during optimization* and during placement. The *timing analysis performed* throughout the conventional circuit design process *is very time consuming, and accounts for most of the run time of* a *conventional* circuit design *system*.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 4 In step 105 the placement program will *modify* the *net lengths*. *Depending on* which *location* was *chosen for each* cell, the *length* of *each net* can be different. *As* the *length* differs, *the capacitive load of the net changes*, and as a result, the delay of the cell driving the net changes. *Therefore the delays which were carefully optimized during the logic* synthesis, *are very different after placement*, and the *optimization* of the *network* is not very good.	'446 PATENT AT 2:12-19 *Depending on* the *location chosen for each* gate, each net *length* may be *modified*. *As each net length* is modified, *the capacitive load of the net* will *change*. *Therefore, the delays, which were carefully optimized during the logic* design, *are very different* in value *after* cell *placement*, thereby contributing to poor *network optimization*.
SYNOPSYS DRAFT PAT. APP. AT 4 *Much of the progress in the state of the art can be characterized as increased integration*. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It *has led to increasingly complex software systems which are slow and difficult to design and maintain.*	'446 PATENT AT 2:20-23 Additionally, *much of the progress in the state of the art for digital circuit design can be characterized as increased integration* which *has led to increasingly complex software systems which are slow, and difficult to design and maintain.*
SYNOPSYS DRAFT PAT. APP. AT 4 *Iterating between placement and sizing* has been especially hard to execute because placement programs are not sold by the same design automation software vendors as *logic synthesis programs*. Also they are not run by the same users: the logic synthesis program is often run by the *designer, who also wrote the HDL specification*. The *placement program is often* run *by the silicon chip manufacturer, after the design is* considered *complete*.	'446 PATENT AT 2:24-30 A further disadvantage with conventional design approaches is in the difficulty of *iterating between placement and sizing*, since the *logic synthesis program* is often operated by the logic *designer who also wrote the HDL specification*, but the *placement program is often* operated *by the silicon chip manufacturer, after the design is complete*.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 6 The *present invention maintains timing closure* after it has been achieved by *adjusting* the *size* of the cell *during or after placement. The adjustments compensate for the fact that the placement algorithm* can *assign different net lengths to different nets and that these lengths are difficult to predict* before *placement*.	'446 PATENT AT 16:23-29 According to the *present invention, timing closure* is *maintained* after placement occurs of cells 836. To *maintain timing closure*, the *size* of a particular gate may be *adjusted during or after placement.* This *adjustment compensates for the fact that placement algorithm* may *assign different net lengths to different nets, and that these* different net *lengths are difficult to predict* prior to the *placement* step.
SYNOPSYS DRAFT PAT. APP. AT 8 Wherever possible, the same *reference numbers* will be used throughout the *drawings* to *refer* to the same or like *parts*.	'446 PATENT AT 4:59-63 *Referring* in detail now to the *drawings* wherein similar *parts* or steps of the present invention are identified by like *reference numerals*, there is seen in FIG. 1 a schematic diagram of a host computer system 100 which is capable of implementing the present invention.
SYNOPSYS DRAFT PAT. APP. AT 9 The cells *can be combinational "gates"* 207, 208, 209, *whose function is represented as* an *expression* in the *Boolean* algebra, using *AND, OR and NOT operators*, or the cells *can be registers* 205, 206.	'446 PATENT AT 5:13-17 The gates *can be combinational gates whose function is represented as Boolean expression* based on, for example, the *operators AND, OR and NOT*. The gates *can* also *be registers*.
SYNOPSYS DRAFT PAT. APP. AT 9 *Each* cell (*e.g.*, 208) *has one or more inputs* 212, 213, *and a single output* 214.	'446 PATENT AT 5:18-19 *Each* gate (*e.g.*, gate j) *has one or more input* 155 *and a single output* 160.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 9 Cells *whose inputs are connected to the output of a* cell *are called the fanin of the latter* cell. Cells *whose inputs are connected to the output of a* cell *are called the fanout of the latter* cell.	'446 PATENT AT 5:26-32 Gates *whose outputs are connected to the inputs of a* gate *are* collectively *called the "fanin" of the latter* gate. Thus, the gate k is in the fanin of the gate i. Gates *whose inputs are connected to the output of a* gate *are* collectively *called the "fanout" of the latter* gate. Thus, the gate j is in the fanout of the gate i.
SYNOPSYS DRAFT PAT. APP. AT 9 *The digital* network *performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the* network on its **primary inputs** 201, 202, *and the result of the computation of the* network *function is presented at the *primary outputs** 203, 204, of the network. *The computation of the function* takes *one or more cycles. During each cycle the gate functions are calculated. The results are stored in* the *registers for use in the next cycle.*	'446 PATENT AT 5:33-41 *The digital* circuit 150 *performs a logic function by processing digital binary input data in a number of cycles. The input data is presented to the* digital circuit 150 at the *primary inputs* 170, *and the result of the computation of the* digital circuit *function is presented at the primary outputs* 175. Typically, *the computation of the* digital circuit *function* requires *one or more cycles. During each cycle, the gate functions are calculated,* and *the* calculation *results are stored in registers for use in the next cycle.*
SYNOPSYS DRAFT PAT. APP. AT 10 The *"arrival time" of the data at a gate is computed by taking the maximum arrival time of* its *fanin* cells each increased by *the delay from the input pin to the output pin.*	'446 PATENT AT 9:55-58 (An *arrival time of the data at a gate is computed by taking the maximum arrival time of* the *fanin* gates plus *the delay* measured *from the input pin to the output pin* of the gate).

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 11 The difference between the required time and the arrival time is the ***slack***. If the arrival time is smaller than the required time, ***the timing constraints are met***, and the ***slack is positive***. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. ***All slacks can be summarized*** as a ***single worst slack number***, called ***the *network slack***. ***Timing closure is achieved if the network slack is non-negative.***	'446 PATENT AT 13:27-34 This determination is made by subtracting the delay of the buffer from the "local ***slack***", to give the value of the predicted slack after buffer insertion. ***Slack is*** zero or ***positive*** if ***the timing constraints are met***. In addition, ***all slacks*** in the circuit ***can be summarized by the "network slack"*** which is the ***single "worst" slack number***. ***If the network slack is non-negative***, then the ***timing closure is achieved***.
SYNOPSYS DRAFT PAT. APP. AT 11 It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S, and ***the delay D is non-negative and*** monotonically ***increas***ing with ***C/S***.	'446 PATENT AT 6:38-43 The delay D of a gate can be approximated by equation (1): D=f(C/S) (1) ***The delay D is non-negative and increas***es as the ***C/S*** value increases.
SYNOPSYS DRAFT PAT. APP. AT 11 ***The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition.***	'446 PATENT AT 6:58-61 ***The delay*** D value ***may*** also ***be different for different inputs of the gate and it may*** also ***be different for the falling transition and rising transition*** of a signal propagating through the gate.
SYNOPSYS DRAFT PAT. APP. AT 12 ***The library analysis will determine a good value for C/S for each*** cell ***in the library***.	'446 PATENT AT 6:63-65 ***The library analysis will determine a "good" value for C/S for each gate in the library*** based on gain considerations.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 13 Mostly these optimizations ***change the structure of the*** network, ***and the Boolean functions of the*** cells, ***without changing the overall function of the*** network. The types of ***optimizations*** that should be performed are ***behavioral optimization such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal.*** These is a large amount of literature on how each of these ***classes of optimizations*** can be performed.	'446 PATENT AT 9:13-22 During this step, ***the structure of the*** circuit ***and the Boolean functions of the*** gates are ***changed*** to reduce the total number of connections, ***without changing the overall function of the*** circuit. Structural ***optimizations*** can include ***behavioral optimizations (such as resource sharing), sequential optimizations (such as retiming), algebraic optimizations (such as kernel extraction), and Boolean optimizations (such as redundancy removal).*** The ***classes of optimizations*** above are well known to those skilled in the art.
SYNOPSYS DRAFT PAT. APP. AT 13 Following the library independent optimizations, the network ***is mapped to a library of cells.*** This means that ***the logic functions of the*** cells ***are implemented with actual cells from the library.***	'446 PATENT AT 9:25-27 In step 210 (FIG. 4), the circuit ***is mapped to a library*** 209 ***of cells.*** Thus, ***the logic functions of the*** circuit gates ***are implemented with actual cells from the library*** 209.
SYNOPSYS DRAFT PAT. APP. AT 14 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.***	'446 PATENT AT 10:45-49 A local ***transformation*** is then ***used to reduce the number of levels in the logic in the*** gate chain circuit 550. The result of the ***transform***ation is shown as gate chain circuit 550' in FIG. 7B. ***The number of levels*** in the logic is ***reduced by bringing*** the gate 555 ***forward.***
SYNOPSYS DRAFT PAT. APP. AT 14 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.***	'446 PATENT AT 10:59-62 In order for the transformation shown in FIG. 7B to be valid, ***it is necessary that gates*** 555, 560, and 565 ***are fanout free. If*** the gates 555, 560, and 565 are ***not*** fanout free, then ***they*** are ***made fanout free*** through ***copy***ing logic.

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 14-15 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.*	'446 PATENT AT 11:4-13 Under *conventional logic* design, *copying logic will increase the load on* the *gates* whose outputs are connected to lines 575, 580, 585, and 590. In the example of FIG. 7B, the copying logic 555' *increases the load on the gates* whose outputs are connected to lines 575 and 580. *To predict* whether or not *the transformation improved delay, it* is *necessary to run a complete static timing analysis with accurate delay models. If the* transformation (from circuit 550 to 550') were *actually* harmful to *delay, then the* transformation *would* have to *be undone.*
SYNOPSYS DRAFT PAT. APP. AT 15 *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 fanin of gate* xxx. *The delay can* easily *be predicted by simple addition of gate delays.*	'446 PATENT AT 10:49-58 *In the constant delay model approach, the effect of this* transformation *can be easily predicted. Changes in the gate loads do not affect delay,* since delay is maintained as constant while gate size will be adjusted (during or after placement) to compensate for the load change. *The only change* which *affects delay* (of the gate chain circuit 550) *is the change of the fanin of gate* 555. *This delay* change *can be predicted by simple addition of gate delays* provided by the fanins connected at lines 590, 575, and 580 (see gate chain circuit 550' in FIG. 7B).
SYNOPSYS DRAFT PAT. APP. AT 15 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.*	'446 PATENT AT 11:26-30 The parameter w represents the *net* (wire) *load* for a given gate i (wherein the *net load can be estimated using a conventional net load model* such as the above-mentioned fanout-based model) *plus any other fixed load such as the load of* the *primary output* of the circuit implementation.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 15 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.*	'446 PATENT AT 11:48-52 If the digital circuit is *a combinational network* (see, e.g. circuit 150 in FIG. 2), then gate load calculation initiates *at the primary outputs* 175 *and traverses the* circuit *in a leveled order toward the primary inputs* 170.
SYNOPSYS DRAFT PAT. APP. AT 15-16 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.	'446 PATENT AT 11:53-12:4 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*.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 16 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***.	'446 PATENT AT 12:5-20 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).
SYNOPSYS DRAFT PAT. APP. AT 16-17 We can do this by ***calcula***ting 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.	'446 PATENT AT 12:22-30 Thus, the following discussion now turns to the ***calculation*** of "***net weights***." The ***net weight represents the sensitivity of the total area of*** a digital circuit ***with respect to the load*** of a particular ***net***. As an example, ***the net weight*** of a given ***gate***, which is immediately coupled to ***the primary inputs*** of a digital circuit, ***is equal to its area per unit load***. Using equation (6), ***the net weight of the other*** gates in the digital circuit are then ***calculated*** in a leveled order towards the primary outputs of the digital circuit.

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 17 The *buffering* algorithm works as follows: First it finds *locations in the* network *where a buffer can be added* without increasing the network delay. This is done *by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer* can be *inserted without increasing the* network *delay.*	'446 PATENT AT 13:23-37 The *buffering* step of 215 (FIG. 4) is discussed in further detail with reference to FIG. 8. In step 650, *locations in the* circuit are determined *where a buffer can be added* so that buffer insertion will still permit timing constraints to be met. This determination is made *by subtracting the delay of the buffer from the "local slack", to give the* value of the *predicted slack after buffer insertion.* Slack is zero or positive if the timing constraints are met. In addition, all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then timing closure is achieved. *If the predicted slack* calculated in step 650 *is larger than the network slack, then* it is possible to *insert* a *buffer without increasing the* circuit *delay.*
SYNOPSYS DRAFT PAT. APP. AT 17-18 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* inserting 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 *changing.*	'446 PATENT AT 13:37-48 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, wherein 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.*

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 18 *After the buffer* is *inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.*	'446 PATENT AT 13:53-57 *After the buffer* has been *inserted*, then in step 670 *the capacitance* values *need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.*
SYNOPSYS DRAFT PAT. APP. AT 18 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.	'446 PATENT AT 14:20-36 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 constraints are compressed* (in delay) *until the* long *path*s *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.
SYNOPSYS DRAFT PAT. APP. AT 18-19 *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.*	'446 PATENT AT 15:48-51 *For the purpose of stretching and compressing, registers* in the circuit are preferably *considered* as *part of a path* from *which they originate, but not part of* the *path* from which *they terminate.*
SYNOPSYS DRAFT PAT. APP. AT 19 The stretching algorithm considers the cells *on a path by path basis*, processing the *path* with the smallest slack *first.*	'446 PATENT AT 15:9-10 The invention operates *on a path-by-path basis* whereby the most critical *path* in a digital circuit 750 is evaluated *first.*

Synopsys Draft Patent Application[1]	Specification Of '446 Patent (Also Contained in '438 Patent)
SYNOPSYS DRAFT PAT. APP. AT 19 After a cell ***has been adjusted***, ***it becomes "locked"*** and its ***delay*** can***not be*** changed by the ***stretching*** algorithm.	'446 PATENT AT 15:21-25 After the gate 754 ***has been adjusted*** to meet the Path 2 timing constraints, ***it becomes "locked,"*** whereby the gate 754 delay will ***not be*** adjusted further for the remainder of the compression and ***stretching*** step.