

Description

This invention relates to integrated circuits and methods relating thereto.

Very Large Scale Integration (VLSI) is made possible by our technological ability to create electronic circuits having hundreds of thousands of transistors in a single chip. Efficient design of such VLSI circuits cannot be had without some automatic means for designing the interconnection patterns and for designing the layout of the circuits, including the creation of the active elements within the circuits. Not unexpectedly therefore, a number of products are available in the art that assist in the design of VLSI circuits. An integral part of such design tools is a means for evaluating the efficacy of the design. This includes means for measuring power consumption, speed of operation, area of the active elements, etc.

Timing analysis is one such tool, and is an important one because it is an essential element of any transistor sizing algorithm. Of course, the timing analysis is important in its own right to the designer of the circuit. In a paper titled "Synchronous Path Analysis in MOS Circuit Simulator," IEEE Proc. 19th Design Automation Conf., (1982), pp. 629-635, V. Agrawal describes an algorithm for estimating the time delay through a digital circuit. It is not the first such algorithm but it is an important one because it is able to discern the presence of latches (memory) in the circuit being investigated and is able to estimate the time delay from any input to any output of the combinatorial portion of the circuit. Two other timing analyzers are the Timing Verifier (TV) program by N. Jouppi, "Timing Analysis for NMOS VLSI," IEEE Proc. 20th Design Automation Conf., (June 1983), pp. 411-418, and the CRYSTAL program by J. Ousterhout, "Switch-Level Delay Models for Digital MOS VLSI," IEEE Proc. 21st Design Automation Conf., (June 1984) pp. 542-548. The CRYSTAL program is available commercially from the Regents of the University of California.

The CRYSTAL and TV programs differ from the Agrawal algorithm in that they use a more detailed timing estimation method, which takes into account differing gate delays for a single gate depending upon which input the signal arrives on. The drawback of both the TV and CRYSTAL programs is that they do not handle the presence of feedback in the integrated circuits, and that excludes almost all of the digital integrated circuits that are designed and built today.

Another important aspect of integrate circuit design is the transistor sizing approach that is employed. There are, of course, many deterministic sizing approaches. One simple approach, for example, is to create all transistors in one size that satisfies the circuit's overall speed requirement. In another, more sophisticated, approach the transistor size is related to how close the transistor is to the power (battery or ground) rail. There are also a number of algorithms that size the transistors within integrated circuits based on the specific circuit design. One of the earliest was the work of Ruehli, et al., described in "Analytical Power/Timing Optimization Technique for Digital System," IEEE Proc. 14th Design Automation Conf., (June 1977, pp. 142-146. They devised their own timing analyzer and with the aid of that analyzer they developed an algorithm for sizing transistors. Their timing analyzer, however, does not have the ability to recognize and deal with the presence of memory in the circuit. It also does not take into account the additional delay accruing from increasing the size of a transistor gate and thereby increasing the capacitance that needs to be driven. An algorithm that does take into account the increased gate capacitance was proposed by M. Matson in "Optimization of Digital MOS VLSI Circuits," Proc. Chapel Hill Conf. on VLSI, (University of North Carolina, May 1985), pp. 109-126. Matson's algorithm also uses a more accurate delay model for the active elements, which takes into account the slope of the input waveform.

The biggest drawback of Matson's algorithm and of all other known sizing algorithms is that they do not have the capability to automatically recognize and deal with feedback generally, and with memory elements in particular. This is a critical failing because essentially all digital integrated circuits are sequential circuits, and sequential circuits contain memory. Another drawback of these algorithms is their use of unnecessarily accurate optimization algorithms, which result in unacceptably slow analysis of large integrated circuit designs.

In 22nd ACM/IEEE Design Automation Conf., June 23-26, 1985, Las Vegas, Nevada US; pages 781-784; Kao et al.: "Algorithm for automatic transistor sizing in CMOS digital circuits", the sub-elements of the evaluated circuits are gates. These are: NAND, NOR, INVERTER, and PUSH-PULL gates, and PASS transistors. Those gates (and PASS transistors) have their characteristics stored in a "schematic database". In operation, the user selects a path. Once selected, a delay for the path is first calculated using the initial sizes given by the user. If the total circuit delay does not match the user-specified delay, a gate selection routine selects the gate with the largest circuit delay per unit area, and the size of that gate is increased. The steps of delay computation, selection of a gate and an increase in the size of the gate are the only automatic steps in the algorithm. To actually carry out a design process with the algorithm, the designer must try several runs for each path until satisfied. When a circuit has many paths, this results, of course, in a very burdensome process.

It is an object of our invention to overcome one or more of the limitations of the above prior methods.

According to one aspect of this invention there is provided an integrated circuit as claimed in claim 1.
According to another aspect of this invention there is provided a method as claimed in claim 2.

A subject design of an integrated circuit is iteratively analyzed and improved with each iteration, until a preselected constraint, such as speed of response, is met. The design improvement is realized by selecting a model for the delay through each active element of the circuit that is characterized by a convex function of the active element's size. Using the convex-function model, with each iteration a static timing analysis of the circuit is employed to identify the output that most grievously violates the specified constraint. With the output selected, an analysis of the path's timing structure identifies the active element in that path whose change in size would yield the largest improvement in performance. The size of that active element is adjusted accordingly, and the iteration is repeated.

For further improvement, at selected iterations, the interconnection pattern of subnetworks of the circuit is evaluated and rearranged to improve performance. Generally, that means that those active elements whose inputs arrive late are placed electrically closer to the output port of the subnetwork.

Brief Description of the Drawing

FIG. 1 is a generalized description of a digital circuit which is segmented into a combinatorial logic portion and a memory portion;

FIG. 2 illustrates the distributed RC model of an MOS transistor employed in connection with one embodiment of our invention;

FIG. 3 describes a three transistor "totem pole" interconnection, and the equivalent circuit that describes the delay through the circuit;

FIG. 4 is a flowchart providing a pictorial representation of the method employed in the system of our invention;

FIG. 5 illustrates a two transistor circuit arrangement where the input signals to the two transistors have different times of arrival;

FIG. 6 describes the general structure of CMOS integrated circuits, with a PFET network being employed as a pull-up circuit and an NFET network being employed as a pull-down circuit;

FIGS. 7 and 8 illustrate the need to sort elements within subnetworks of a circuit in order to optimize performance;

FIG. 9 depicts a system for performing the invention; and

FIG. 10 shows the layout of a portion of an integrated circuit embodying the invention.

Electronic circuits can be divided into purely combinatorial circuits, and circuits that contain feedback paths. In purely combinatorial circuits, signals flow in one direction only and combine in different ways to achieve a desired logical state. In circuits that contain feedback paths, memory can be realized and the presence of memory makes the circuit a sequential circuit. It is well known that a sequential circuit can always be divided into a section containing purely combinational logic, and a section containing nothing but a plurality of basic memory elements e.g., flip-flops. Such a circuit subdivision is depicted in FIG. 1.

Given a circuit as shown in FIG. 1 having a total number of N transistors (e.g., MOS transistors) of sizes $x_1, x_2, \dots, x_N$, our method improves circuit performance by adjusting the x_i values. Two figures of merit are of special interest. One is T , which is the minimum clock period at which the circuit will operate or, for unclocked paths, the propagation delay. The other is A , which is simply the sum of transistor sizes or, stated differently, the total area of the active elements. It turns out that the area, A , is positively correlated with a number of other attributes of the circuit that should be minimized or constrained. First, as A increases the circuit requires more silicon area. Second, since the total gate capacitance is proportional to A , the power required to charge or discharge capacitance increases as A increases. Third, for a static CMOS gate there is a brief period of time during transition when both the pull-up and pull-down networks are partially turned on. This short-circuit power also increases with A . Likewise, a ratioed-load NMOS gate consumes more power as the pull-down transistor sizes are increased. Fourth, and perhaps most important in many situations, the probability of a device fault in a chip depends strongly on the total active area of the chip. As the active area increases the probability of chip failure increases.

The critical aspect of any method of optimization is the form of the equation or function that is optimized and, more specifically, the nature or the behavior of that function.

We seek to select a function to be minimized that yields a good result and also assures that the result is reached effectively and expeditiously. That means that the result would be close to the absolute minimum, would be reached quickly, and the process would not get "stuck" at some local minimum of the

function. With integrated circuits having hundreds of thousands of transistors, an optimization process that achieves its end results expeditiously is not a luxury but an absolute necessity.

Thus, as a preliminary step, the attribute that is to be optimized and the constraint that is to be imposed must be selected. We consider three formulations:

- 1) minimize the area of the active elements in the integrated circuit subject to the constraint that the maximum delay at any output is less than a preselected constant, K;
- 2) minimize the maximum delay at any output subject to the constraint that the area of the active elements is less than a preselected constant; and
- 3) minimize the factor AT^K .

The first formulation is probably the most useful one because often the desired speed of operation (clock frequency) is given. In such a case, the first formulation can be used to fit the circuit within a system with a given clock period K. Also, unlike the other two formulations, two different circuits that are optimized separately with the same value of K are also optimal when considered jointly. This fact allows subcircuits to be optimized separately without loss of optimality in the final circuit.

The second formulation might be used when the circuit is to be made as fast as possible, subject to constraints on silicon area, power, or yield.

The third formulation represents a holistic approach in which both A and T are important, but have relative weights assigned to them. For example, it may be that T is important enough that one might be willing to pay for a one percent decrease in T with up to a five percent increase in A. In such a circumstance, one would set K to five.

The second and, perhaps, the more important factor to be considered in pursuing the minimization process is the model used in representing the active elements of the circuit. For the case of MOSFET integrated circuits we selected the model shown in FIG. 2, where 10, 11, and 12 represent the gate, drain, and source nodes of a MOSFET, respectively. Capacitor C_g (13) is the capacitance between the gate and ground, capacitor C_d (14) is the capacitance between the drain and ground, and capacitor C_s (16) is the capacitance between the source and ground. Switch 17 is closed when the FET is turned on, and resistor R_{ds} (15) is the resistance between the drain and the source when the FET is on. The gate, source, and drain capacitances are all directly proportional to the transistor size x (channel width), while the source-to-drain resistance is inversely proportional to x.

FIG. 3 illustrates the modeling of gate delay, in accordance with the distributed RC model of FIG. 2, for a "totem-pole" arrangement of three FETs. The Pentfield-Rubinstein distributed RC model implies that an upper bound for the discharge time of the FIG. 3 circuit is

$$(R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3, \quad (1)$$

where R_1 , R_2 , and R_3 are equal to the R_{ds} resistances of transistors x_1 , x_2 , and x_3 , respectively, C_1 is equal to C_d of transistor x_1 plus C_s of transistor x_2 , C_2 is equal to C_d of transistor x_2 plus C_s of transistor x_3 , and C_3 is C_d of transistor x_3 and C_L . It may be noted that C_1 is below the closing switch and it speeds the gate's operation rather than slowing it down upon closure of the switch. This can clearly be seen when it is observed that the initial current flow sinks into capacitor C_1 , which appears as a short circuit, and no current flows through resistor R_1 . Hence, in assessing an upper bound on the delay (as above), C_1 can be neglected and therefore C_1 does not appear in Equation (1).

When Equation (1) is expressed as a function of transistor sizes x_i , each R_i is proportional to $1/x_i$ and each C_i is some constant plus terms that are each proportional to some x_i . The constant term in C_i represents the wire capacitance, and the variable term represents the capacitance of the gate, drain, or source, as the case may be. Thus, Equation (1) can be written as:

$$\left(\frac{A}{x_1} + \frac{A}{x_2} \right) (Bx_2 + Bx_3 + C) + \left(\frac{A}{x_1} + \frac{A}{x_2} + \frac{A}{x_3} \right) (Bx_3 + D), \quad (2)$$

where A, B, C, and D are constants.

Recognizing that the delay through a path is the sum of delays encountered in the path, we reach the general form of a path delay in a circuit as follows:

$$\sum_{1 \leq i, j \leq N} a_{ij} \frac{x_i}{x_j} + \sum_{1 \leq i \leq N} \frac{b_i}{x_i}, (3)$$

where the a_{ij} and b_i are nonnegative constants that for any specific path in the circuit are mostly zero. It can be shown that the function described by Equation (3) is convex, which means that any straight line segment in $N+1$ -dimensional space whose end points lie in the graph of the function, is itself entirely on or above the graph. In addition, Equation (3) belongs to a special class of convex functions called posynomials.

As a consequence of the above, all three formulations of the constraints can be seen to be the minimization of a convex function, subject to an upper bound on another convex function. The convexity property yields enormous advantages that have not been possible to realize with the minimization functions employed in the prior art. The biggest advantage is that any point found to be a local minimum is also the global minimum.

The field of Convex Programming has been intensively explored over the last several decades and many computational techniques are available. See, for example, R. T. Rockafellar, Convex Analysis, (Princeton University Press 1970).

Since the desired optimization is that of a part of a chip, an entire VLSI chip or perhaps an entire system, the optimization method must be as efficient as possible; even perhaps at the expense of absolute accuracy in finding the optimum. Our process, which provides an efficient method for closely approximating the optimum point, arrives at the optimized solution quite quickly by proceeding generally in accordance with the following steps.

The process starts with all active elements in the VLSI circuit design being set at their minimum possible size (block 100 in FIG. 4). This results in an initial design that contains the minimum active area and is the most reliable (the larger the active area, the more probable are the chip defects in manufacturing).

Next, the circuit's description equations (specification) are arranged to conform to the FIG. 1 structure (block 100 in FIG. 4).

Following the above preliminary steps, an iterative process starts with a circuit analysis (from the specification resulting from step 2) to identify and select the output that fails most (block 110 in FIG. 4). By "fails most" we mean that, for example, when T is 1 microsecond, one output has a delay of 1.2 microseconds and another output has a delay of 1.5 microseconds, then the 1.5 microseconds output is selected as the output that fails most.

In the next step the causes of the delay in the selected output are investigated by examining, in turn, each transistor which could have an effect on that output (block 120 in FIG. 4). To perform the examination, two numbers are assigned to each electrical node: t_l (latest-time-to-go-low), and t_h (latest-time-to-go-high). In examining a delay "failure," when t_h causes the "failure," the pull-up circuitry is examined, while when t_l causes the "failure," the pull-down circuitry is examined. Such examination includes both the active element under consideration and its supporting elements. Among the supporting elements are, of course, all the active elements that contribute to that element's input signal. This creates a chain-like examination that may be characterized as a "walk" backwards which identifies the critical path that controls the delay in the selected output. In FIG. 3, for example, the investigation includes all three transistors in addition to the investigation of the critical path leading to the input that happens to be the slowest.

For each such transistor, i , examined in the previous step, the time savings accruing from incrementing x_i by a fixed amount is calculated. The one transistor in the critical path that offers the greatest such savings is incremented by that fixed amount (block 130 in FIG. 4).

The iterative loop is closed (in block 140 of FIG. 4) by returning to the point of output delay determination (i.e., to block 110 in FIG. 4), until either the imposed constraint is satisfied or no transformation is possible that reduces the quantity being minimized.

Since the number of paths through a circuit can be very large in comparison to the size of the circuit itself, our method realizes an advantage by never actually expanding the circuit into an enumeration of the paths. In addition, a change of a particular x_i can only change two gate delays: delay of the gate that includes the altered transistor, and the delay of the gate that drives the altered transistor. Therefore, when repeating the delay calculation to find the worst output delay we recalculate only those quantities that might change as a result of the single variable change.

As indicated by the above equations, the delay curve as a function of size follows the relationship $Ax + B/x + C$ where A , B , and C are greater than 0; which means that from some minimum value of x , as x is increased the delay is decreased until some minimum delay is reached. Beyond the size of minimum delay,

continuing to increase in size increases the delay.

Ordinarily, it would be expected that at no time would one wish to increase the size of a transistor beyond its minimum delay point. However, this is exactly what may be done. The reason for that lies in the observation that the size of a transistor affects not only its speed but also the speed of its neighbors, and in the observation (made earlier in connection with FIG. 3) that in a "totem-pole" arrangement, increasing the capacitance (i.e., size) of the transistor that is located closer to a power rail reduces the delay of the switching transistor with respect to the output of the "totem-pole." Thus a step is included in our method (within block 120) that analyzes all situations where more than one input affects an output, with an eye toward increasing the size of those transistors that affect the output earlier than the slowest input; even to the point of slowing down the signal on the gate of those transistors somewhat. It may be noted that when the size of a transistor, x , is small, the delay decreases rather rapidly with respect to increase in area. When x is large, the delay increases linearly with x , which is a relatively slower rate of increase in the delay.

Furthermore, a step is included in our method (block 150 in FIG. 4) that sorts the order of series subnetworks of a subcircuit within the VLSI circuit in order to improve the subnetworks' performance (speed). A specific example might illustrate this best and, to that end, FIG. 5 depicts a series subnetwork of two transistors where the input to transistor 20 arrives with a 7 nsec delay and the input to transistor 21 arrives with a 3 nsec delay. In the timing arrangement illustrated in FIG. 5, if each of the capacitors (in the distributed RC model) is taken to have the value C and each of the resistors is taken to have the value R , then the output-discharge time (controlled by the slower input) is $7\text{nsec} + R2C + 2R(C + C_{\text{load}})$. When we reverse the input leads and the slower input is applied to transistor 21, then the output-discharge time is $7\text{nsec} + 2R(C + C_{\text{load}})$. The difference between these two delays, $2RC$, represents the time that it takes transistor 20 to discharge the capacitance of the node between the two transistors. This discharge must take place in both configurations of the FIG. 5 circuit, but, in the latter configuration, it takes place in the "free" time interval between 3 nsec and 7 nsec, before the later input arrives.

Generalizing the above with reference to FIG. 6, a static CMOS complex gate consists of a pull-up network 25 and a pull-down network 26. The pull-up network contains only PFETS, and the pull-down network contains as many NFETS. The inputs to the two networks are identical, and the two networks (which are duals of each other) are connected to each other in series between V_{dd} and ground. The connection between the two networks forms the output of the gate. A network with two connection points and any number of inputs can be constructed from single transistors by any recursive composition of the following tree rules:

- 1) A single transistor is a network. In FIG. 2, for example, the source and drain are the two connection points, and the gate is the single input of the network.
- 2) Any number of networks can have their connection points connected in series to form another network (e.g., FIG. 5). The new network's inputs are the union of the inputs of the component networks. The new network's connection points are the two connection points of the components left free.
- 3) Any number of networks can have their connection points connected in parallel to form another network. The new network's inputs are the union of the inputs of the component networks. The new network's connection points are the connection points of the component network that have been connected.

Proceeding from the above, whenever a network is formed by the series composition rule from N subnetworks, the logic function computed by the network is unaffected by which of the $N!$ subnetwork orderings is chosen. However, as was shown in the simple example of FIG. 5, the timing performance of the gate is, in general, different for different orderings. To minimize delay, our improved method orders the subnetworks according to the latest-time-to-turn-on for each subnetwork. The subnetwork with the earliest such time is placed nearest the power rail (V_{dd} or ground, as the case may be), and the subnetwork with the latest such time is placed nearest the output. Given a subnetwork, this latest-time-to-turn-on is the maximum of the latest-time-to-turn-on values of the constituent transistors. The latest-time-to-turn-on of an NFET is the latest-time-to-go-high of its gate. The latest-time-to-turn-on of a PFET is the latest-time-to-go-low of its gate.

FIGS. 7 and 8 illustrate this process with a complex gate that contains 14 transistors. FIG. 7 is before sorting and FIG. 8 is after sorting. The latest-time-to-go-high and the latest-time-to-go-low of inputs A, B, C, and D are ordered $A_1 < B_1 < C_1 < D_1$ and $A_h < C_h < B_h < D_h$. In subnetwork 30, the three transistors in series are already in optimal order: A should be closest to V_{dd} , since its latest-time-to-go-low is earliest, followed by B and then C. However, the pull-up is composed of subnetworks 30 and 31 in series, and these networks are in their own order; subnetwork 31, which has a latest-time-to-turn-on that is earlier than that of subnetwork 30, should be closest to V_{dd} . Accordingly, in FIG. 8, the positions of subnetworks 30 and 31 are reversed. Meanwhile, in the pull-down subnetwork, 32, one reordering needs to be made; and that is in the three

transistors in series. Transistor B should be closest to the output followed by C, followed by A. This rearrangement within subnetwork 32 is also shown in FIG. 8.

Implementing the above method requires a combination of conventional elements that are all well known. The system implementation can be done in special purpose hardware or in a system comprising generalized circuits that are software controlled. With present day technology the latter approach may be easier to realize, particularly since a number of the steps called for are algorithmic in nature and are available in the art in software form.

The structure of a system for performing the invention is shown in FIG. 9. Block 151 is a timing analyzer which receives as input the specification of the circuit whose layout is to be optimized. It contains a model representing the active elements of the circuit which yields a convex delay function. With the aid of that model and under control of block 157, block 151 analyzes the specification (as taught, for example, by V. Agrawal) and develops a table that correlates the outputs of the circuit with approximated time delays. That table is stored in memory 156. Control then passes to block 152 which obtains the speed criteria to be satisfied from control block 157 and selects that output from the table created by block 151 which fails the speed criteria by the greatest amount. Control then passes to block 153, where a critical path associated with the selected output is identified. Each active element in the identified critical path is evaluated to determine its sensitivity to changes in size by evaluating the incremental change in delay for a given incremental change in size (from the existing size). It may be noted here that all active elements begin with some initial minimum size. The active element that is found to have the greatest incremental improvement in delay is selected, and its changed size specification is noted in another table within memory 156. Control then passes to block 155, where the effect on output delays caused by the change in size effected by block 155 is evaluated. The results of this evaluation are fed to block 157, where the resultant collection of circuit output delays and active element sizes (from the tables in memory 156) is compared to the operational criteria provided to block 157. Control is returned to block 152 until the criteria are met.

To provide further assistance, the following concentrates on the method steps that may be employed to carry out the invention.

The step of selecting the minimum size of transistors is taken simply on the advice of the integrated circuit manufacturer. The step of separating a circuit's specification into a combinatorial portion and a memory is well known and, indeed, the aforementioned Agrawal timing analyzer incorporates this step and therefore can be used in the implementation of our system and method.

The output of the timing analyzer is a table that correlates the circuits' outputs to the delay of those outputs. From this table, the worst output is selected and the "walk" backwards begins. Based on the specification of the subnetwork developing the worst output, with the aid of the distributed RC model, the effect of each transistor in that subnetwork is computed. Also, the input to the subnetwork that constitutes the critical input as far as delay is concerned is identified. The transistor in the subnetwork that is found to yield the largest improvement in performance is remembered and the process continues with the analysis of the subnetwork that feeds the critical input. With each subnetwork analysis, a transistor offering the largest improvement in performance is compared with the remembered transistor, and the one offering the larger improvement in performance is kept. As a consequence, when the entire critical path (of the worst output) has been analyzed, the transistor remembered is the one within the entire critical path that offers the largest improvement in performance. The size of that transistor is altered and an updated timing analysis of that output is performed and placed in the table. The process is then reiterated with a new worst output selected and analyzed.

At a predetermined rate ranging from every iteration to every M iterations, a sorting process is initiated. The sorting itself is completely conventional. Each logic subnetwork has a limited number of inputs which generally is relatively small (e.g., <10) and the number of possible permutations is therefore manageable. Each subnetwork in the critical path is evaluated and a sorting of the inputs is suggested based on the latest-time-to-turn-on of each subnetwork.

A simple "bubble-sort" algorithm is employed to effect the actual subnetwork sorting.

FIG. 10 illustrates a sample layout that results from application of our invention. Shown are MOS transistors with different sizes (channel widths) that were determined to yield, in combination, and the maximum speed is a given area.

Claims

1. An integrated circuit comprising memory elements, input ports, output ports and a plurality of active elements constructed in accordance with a layout arrangement, CHARACTERISED IN THAT at least one of said output ports is a critical output port that develops output signals with a delay that essentially



transistors in series. Transistor B should be closest to the output followed by C, followed by A. This rearrangement within subnetwork 32 is also shown in FIG. 8.

Implementing the above method requires a combination of conventional elements that are all well known. The system implementation can be done in special purpose hardware or in a system comprising generalized circuits that are software controlled. With present day technology the latter approach may be easier to realize, particularly since a number of the steps called for are algorithmic in nature and are available in the art in software form.

The structure of a system for performing the invention is shown in FIG. 9. Block 151 is a timing analyzer which receives as input the specification of the circuit whose layout is to be optimized. It contains a model representing the active elements of the circuit which yields a convex delay function. With the aid of that model and under control of block 157, block 151 analyzes the specification (as taught, for example, by V. Agrawal) and develops a table that correlates the outputs of the circuit with approximated time delays. That table is stored in memory 156. Control then passes to block 152 which obtains the speed criteria to be satisfied from control block 157 and selects that output from the table created by block 151 which fails the speed criteria by the greatest amount. Control then passes to block 153, where a critical path associated with the selected output is identified. Each active element in the identified critical path is evaluated to determine its sensitivity to changes in size by evaluating the incremental change in delay for a given incremental change in size (from the existing size). It may be noted here that all active elements begin with some initial minimum size. The active element that is found to have the greatest incremental improvement in delay is selected, and its changed size specification is noted in another table within memory 156. Control then passes to block 155, where the effect on output delays caused by the change in size effected by block 155 is evaluated. The results of this evaluation are fed to block 157, where the resultant collection of circuit output delays and active element sizes (from the tables in memory 156) is compared to the operational criteria provided to block 157. Control is returned to block 152 until the criteria are met.

To provide further assistance, the following concentrates on the method steps that may be employed to carry out the invention.

The step of selecting the minimum size of transistors is taken simply on the advice of the integrated circuit manufacturer. The step of separating a circuit's specification into a combinatorial portion and a memory is well known and, indeed, the aforementioned Agrawal timing analyzer incorporates this step and therefore can be used in the implementation of our system and method.

The output of the timing analyzer is a table that correlates the circuits' outputs to the delay of those outputs. From this table, the worst output is selected and the "walk" backwards begins. Based on the specification of the subnetwork developing the worst output, with the aid of the distributed RC model, the effect of each transistor in that subnetwork is computed. Also, the input to the subnetwork that constitutes the critical input as far as delay is concerned is identified. The transistor in the subnetwork that is found to yield the largest improvement in performance is remembered and the process continues with the analysis of the subnetwork that feeds the critical input. With each subnetwork analysis, a transistor offering the largest improvement in performance is compared with the remembered transistor, and the one offering the larger improvement in performance is kept. As a consequence, when the entire critical path (of the worst output) has been analyzed, the transistor remembered is the one within the entire critical path that offers the largest improvement in performance. The size of that transistor is altered and an updated timing analysis of that output is performed and placed in the table. The process is then reiterated with a new worst output selected and analyzed.

At a predetermined rate ranging from every iteration to every M iterations, a sorting process is initiated. The sorting itself is completely conventional. Each logic subnetwork has a limited number of inputs which generally is relatively small (e.g., <10) and the number of possible permutations is therefore manageable. Each subnetwork in the critical path is evaluated and a sorting of the inputs is suggested based on the latest-time-to-turn-on of each subnetwork.

A simple "bubble-sort" algorithm is employed to effect the actual subnetwork sorting.

FIG. 10 illustrates a sample layout that results from application of our invention. Shown are MOS transistors with different sizes (channel widths) that were determined to yield, in combination, and the maximum speed is a given area.

Claims

1. An integrated circuit comprising memory elements, input ports, output ports and a plurality of active elements constructed in accordance with a layout arrangement, CHARACTERISED IN THAT at least one of said output ports is a critical output port that develops output signals with a delay that essentially



is at a preselected maximum level, the or each critical output port has an associated critical path that is responsive to at least one of said memory elements and contains those of said active elements whose size variations affect the signal delay of said critical output port, and at least some of said active elements in the or each critical path are at sizes greater than a minimum size, said active element sizes being such that reducing the size of any active element in the or each critical path, without increasing the size of any other active element in that critical path by at least a corresponding amount, causes the signal delay of the associated critical output port to exceed said preselected maximum delay.

2. A method of altering a given layout of an integrated circuit having memory elements, input ports, output ports and a plurality of active elements constructed in accordance with a layout arrangement, the method comprising the following steps:
 - a) designating said given layout as the current layout,
 - b) assigning minimum size to all active elements
 - c) performing a timing analysis of said current layout using a model representing said active elements and having a function that is convex in the neighborhood of the operating point of said active elements to identify an output port of said integrated circuit that has a slower response than all other output ports of said integrated circuit,
 - d) performing a sensitivity analysis on the identified output port to locate an active element of said integrated circuit whose change in size improves the speed of response of said identified output port by the greatest amount,
 - e) automatically altering the size of said located active element, and designating the layout resulting from said altering as the current layout, and reiterating through the steps c) to e) of this method until a preselected maximum delay criterion is met.
3. A method as claimed in claim 2 wherein the step of performing a timing analysis comprises the steps of obtaining information of the output signal timing delay between all of said input ports and each of said output ports, and selecting an output port from said output ports that, in accordance with said step of obtaining information, develops an output signal that is delayed more than all other output signals of said output ports, and the step of performing a sensitivity analysis comprises investigating a subset of said active elements to determine the effect of altering the size of an active element in said subset on the delay of the output signal of said selected output port, and modifying that active element which, in accordance with said step of investigating, is determined to have the greatest effect on improving said delay of the output signal of said selected output port.

Patentansprüche

1. Integrierte Schaltung mit Speicherelementen, Eingangsports, Ausgangsports und einer Vielzahl von aktiven Elementen, die entsprechend einer Layout-Anordnung konstruiert sind, dadurch gekennzeichnet, daß wenigstens ein Ausgangsport ein kritischer Ausgangsport ist, der Ausgangssignale mit einer Verzögerung liefert, die im wesentlichen einen vorgewählten Maximalwert hat, daß der oder jeder kritische Ausgangsport einen zugeordneten kritischen Weg besitzt, der von wenigstens einem Speicherelement abhängt und diejenigen aktiven Elemente enthält, deren Größenänderungen die Signalverzögerung am kritischen Ausgangsport beeinflussen, daß wenigstens einige der aktiven Elemente auf dem oder jedem kritischen Weg eine Größe besitzen, die größer als ein Minimalwert ist und daß die Größe der aktiven Elemente so gewählt ist, daß die Verringerung der Größe eines aktiven Elements auf dem oder jedem kritischen Weg ohne Erhöhen der Größe jedes anderen aktiven Elements auf dem kritischen Weg um wenigstens entsprechend Betrag bewirkt, daß die Signalverzögerung für den zugeordneten kritischen Ausgangsport die vorgewählte Maximalverzögerung übersteigt.
2. Verfahren zur Änderung eines gegebenen Layouts einer integrierten Schaltung, die entsprechend einer Layout-Anordnung konstruierte Speicherelemente, Eingangsports, Ausgangsports und eine Vielzahl aktiver Elemente aufweist, mit den folgenden Verfahrensschritten:
 - a) Bezeichnen des gegebenen Layout als das augenblickliche Layout,
 - b) Zuordnen einer Minimalgröße für alle aktiven Elemente,
 - c) Ausführen einer Zeitanalyse für das augenblickliche Layout unter Verwendung eines Modells, das die aktiven Elemente darstellt und eine Funktion besitzt, die in der Umgebung des Arbeitspunktes



der aktiven Elemente konvex ist, um einen Ausgangsport der integrierten Schaltung zu identifizieren, der ein langsames Ansprechen als alle anderen Ausgangsports der integrierten Schaltung hat,
 d) Ausführen einer Empfindlichkeitsanalyse für den identifizierten Ausgangsport, um ein aktives Element der integrierten Schaltung zu lokalisieren, dessen Größenänderung die Ansprechgeschwindigkeit des identifizierten Ausgangsports um den größten Betrag verbessert,
 5 e) automatisches Ändern der Größe des lokalisierten aktiven Elements und Bezeichnen des sich durch die Änderung ergebenden Layouts als das augenblickliche Layout, und
 Wiederholen der Schritte c) bis e) des Verfahrens, bis ein vorgewähltes Kriterium für die maximale Verzögerung erfüllt ist.

- 10 3. Verfahren nach Anspruch 2, bei dem der Schritt für die Ausführung einer Zeitanalyse die Schritte aufweist: Gewinnen von Informationen für die Ausgangssignalverzögerung zwischen allen Eingangsports und jedem Ausgangsport und
 Auswählen eines Ausgangsports aus den Ausgangsports, der entsprechend dem Schritt für die Gewinnung von Informationen ein Ausgangssignal erzeugt, das stärker als alle anderen Ausgangssignale der Ausgangsports verzögert ist,
 15 und bei dem der Schritt der Ausführung einer Empfindlichkeitsanalyse die Schritte umfaßt:
 Untersuchen einer Untergruppe der aktiven Elemente, um den Einfluß einer Änderung der Größe eines aktiven Elements in der Untergruppe auf die Verzögerung des Ausgangssignals an dem gewählten
 20 Ausgangsport zu bestimmen, und
 Abändern des aktiven Elements, das entsprechend dem Untersuchungsschritt als dasjenige Element bestimmt worden ist, das den größten Einfluß für die Verbesserung der Verzögerung des Ausgangssignals an dem gewählten Ausgangsport besitzt.

25 Revendications

1. Un circuit intégré comprenant des éléments de mémoire, des accès d'entrée, des accès de sortie et un ensemble d'éléments actifs réalisés conformément à une configuration d'implantation, CARACTERISE
 30 EN CE QUE l'un au moins des accès de sortie est un accès de sortie critique qui produit des signaux de sortie avec un retard qui est fondamentalement égal à un niveau maximal présélectionné, l'accès de sortie critique, ou chacun d'eux, a un chemin critique associé qui fonctionne sous la dépendance de l'un au moins des éléments de mémoire et qui contient ceux des éléments actifs dont des variations de taille affectent le retard de signal de l'accès de sortie critique, et certains au moins de ces éléments actifs dans le chemin critique, ou dans chacun d'eux, ont des tailles supérieures à une taille minimale,
 35 les tailles d'éléments actifs étant telles qu'une réduction de la taille d'un élément actif quelconque dans le chemin critique, ou dans chacun d'eux, sans augmenter d'au moins une quantité correspondante la taille de n'importe quel autre élément actif dans ce chemin critique, conduit le retard de signal de l'accès de sortie critique associé à dépasser le retard maximal présélectionné.
- 40 2. Un procédé de modification d'une implantation donnée d'un circuit intégré comportant des éléments de mémoire, des accès d'entrée, des accès de sortie et un ensemble d'éléments actifs réalisés conformément à une configuration d'implantation, le procédé comprenant les étapes suivantes : a) on désigne l'implantation donnée comme étant l'implantation courante, b) on attribue une taille minimale à tous les éléments actifs, c) on effectue une analyse temporelle de la configuration courante, en utilisant un
 45 modèle qui représente les éléments actifs et qui a une fonction qui est convexe au voisinage du point de fonctionnement de ces éléments actifs, pour identifier un accès de sortie du circuit intégré qui a une réponse plus lente que tous les autres accès de sortie du circuit intégré, d) on effectue une analyse de sensibilité sur l'accès de sortie identifié, pour localiser un élément actif du circuit intégré dont un changement de taille améliore de la plus grande quantité la vitesse de réponse de l'accès de sortie identifié, e) on modifie automatiquement la taille de l'élément actif localisé, et on désigne pour
 50 l'implantation courante l'implantation qui résulte de cette modification, et on réitère les étapes c) à e) de ce procédé jusqu'à ce qu'un critère de retard maximal présélectionné soit satisfait.
- 55 3. Un procédé selon la revendication 2, dans lequel l'étape d'accomplissement d'une analyse temporelle comprend les étapes qui consistent à obtenir une information concernant le retard de signal de sortie entre tous les accès d'entrée et chacun des accès de sortie, et à sélectionner un accès de sortie parmi les accès de sortie précités qui, conformément à l'étape d'obtention d'information, produit un signal de sortie qui est retardé davantage que tous les autres signaux de sortie des accès de sortie, et l'étape



d'accomplissement d'une analyse de sensibilité comprend l'étude d'un sous-ensemble des éléments actifs pour déterminer l'effet de la modification de la taille d'un élément actif dans ce sous-ensemble sur le retard du signal de sortie de l'accès de sortie sélectionné, et la modification de l'élément actif qui a été déterminé par l'étape d'étude comme ayant le plus grand effet sur l'amélioration du retard du signal de sortie de l'accès de sortie sélectionné.

5

10

15

20

25

30

35

40

45

50

55