South Asian Journal of Engineering and Technology Vol.2, No.14 (2016) 1–6

ISSN No: 2454-9614

Application of Lattice Theory to Timing Analysis with Cross Talk

R.Viswanathan
Department of Mathematics
Kongu Engineering College,
Perundurai

*Corresponding Author: R.Jeetendra, *Department of Mathematics*
Kongu Engineering College, Perundurai

Received: 03/11/2015, Revised: 02/01/2016 and Accepted: 10/03/2016

Abstract

In this article, the application of Lattice Theory is discussed to establish a theoretical foundation for timing analysis with crosstalk. It is shown that the solution to the problem is to use x points on a complete lattice. Based on that, the convergence of any iterative approach is proved in general. It is also shown that, starting from different initial solutions, an iterative approach will reach different x points. To reach the least x point, it is needed to start from the best case solution and that which starts from the worst case solution, will always reach the greatest x point.

Key words: Lattice theory, Cross talk, Fix point, Multiple x points, Timing analysis, Solution space.

Reviewed by* **ICETSET'16 *organizing committee*

Introduction

Crosstalk is a major problem in IC design. Crosstalk can affect the behavior of a circuit in two ways:

i Introduce noise between adjacent wires;

ii Alter the delay of a switching transition.

This paper is organized as follows, In section 2, the general timing analysis is formulated as computing a x point of a mathematical transformation. Section3 discuss Sapatnekar (2000) which shows that multiple x points exist in the system. Section 4 establishes the fact that the solution space forms a complete lattice, proves the convergence of any iterative approach and characterizes the relation between initial solution and final fixpoint.

Fixpoint computation

Mathematically this means that it may be a scalar value, a vector, or even a function of time (representing the whole wave form). A variable x_i is used to represent each timing information on a point i. Furthermore X is used to represent the vector $(x_1,x_2\ldots x_n)$, that is, the timing information for the whole circuit.

Actually, the timing information on point i is only directly dependent on a subset of other points $(i_1, i_2,\ldots i_k)$. This means we can compute x_i by

$$x_i = t_i(x_{i1},x_{i2},\ldots x_{i3}),$$

where (t_i is any delay model used to derive the timing information on a point from that on its dependent points. When all these local transformations are put together, a transformation for the whole system is got which can be written as

$$X = T(X) \qquad (1)$$

A solution to the timing analysis is an X which satisfies the above that is a fixpoint of T.

When crosstalk effects are included in timing analysis, besides the fanins, the timing information on point i also depend on the timing information on coupled points. For the simplest coupling case shown in figure 2.1, we have

$$x_a = t_a(x_{ia}, x_b)$$

$$x_b = t_b(x_{ib}, x_a)$$

As it can be seen, a cycle is formed here because of the mutual dependence of x_a and x_b. Because of this, the transformation T becomes very complex in the presence of crosstalk.

For a complex transformation T, iterative method may be the only possible way to find its fixpoint. It works as follows. First, an initial solution X_0 is guessed, then the new solutions are iteratively computed from previous solutions $X_1 = T(X_0), X_2 = T(X_2), \ldots, Xn = T(X_{n-1})$ until we find a fixpoint. To the best of our knowledge, all previous work in the literature uses iterative methods to solve timing analysis with crosstalk.

Most of the previous works provide convergence proofs for the approaches. Gross et al (1998) show the convergence based on the approaches similarity to waveform relaxation. Lelarasmee et al (1962), Saptnekar (2000) and Arunachalam et al (2000) base their convergence arguments on the monotonic shrinking of the switching windows. However, none of them study the uniqueness of their solutions. As it will be shown in the next section, uniqueness is not guaranteed by convergence and simply finding one x point is not enough.

Multiple fixpoints

Since it is not possible to study an abstract transformation T, results of Saptnekar (2000) is used which is the simplest among existing work, as our study case.

Saptnekar (2000) considered the delay computation in the presence of crosstalk for a set of wires within a routing channel. For each driver, a switching window [T_{min}, T_{max}] signifying the range of switching time at the input of the driver, and a source resistance, R_d are specified. The intrinsic and coupling capacitances of a wire are computed from the routing. Then coupling capacitances are modeled by effective capacitances to the ground and delays are computed by Elmore delays. The value of an effective capacitance is dependent on the switching time of the two coupling wires. Given a coupling capacitance C_c between two wires, if they switch at the same time and in the opposite direction, then an effective capacitance of $2C_c$ is used; if they switch at the same time and in the same direction, then an effective capacitance of 0 is used; if they do not switch at the same time, then an effective capacitance of C_c is used. However, in static timing analysis, a range of switching time is computed. Thus the worst case analysis is used, which assumes that any switching within the range is possible. The algorithm to compute the wire delays works as follows. First, initialize a switching window on each wire such that the minimum and maximum time are computed by using 0 and C_c as effective capacitances, respectively. Then the maximum time for each wire is updated using effective capacitance of C_c or $2C_c$ based on whether switching windows are overlapping. Similarly, the maximum time for each wire is updated using 0 or C_c. These two updates are repeated in an alternative fashion until there is no further change.

An example is used to show that multiple fixpoints exist for this transformation. In the example, only two nets a and b are shown in Figure 1.

They couple with each other with a capacitance of 10 units. The nets are identical with the same driver of resistance of 10 units and the same load capacitance of 1 unit. Suppose the maximum and minimum arrival time for signal i_a, that is $T_{min}(i_a)$ and $T_{max}(i_b)$, be 0 unit and 1 unit respectively. Similarly let $T_{min}(i_b)$ = 10 units and $T_{max}(i_b)$ = 11 units. Elmore delay is used to compute the delays.

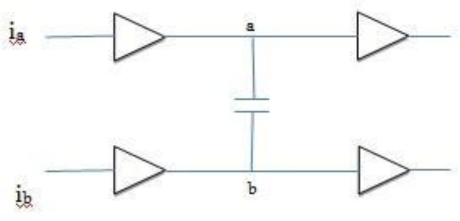

Figure 1: Timing Analysis with Cross Talk

According to the algorithm, the initial switching windows of wires are computed by using effective capacitance of 0 for minimum time and that of C_c for maximum time. That is $[T_{min}(a), T_{max}(a)]$ = [10, 11] and $[T_{min}(b), T_{max}(b)]$ = [20, 121]. Now their switching windows overlap, thus updates are needed. We get $[T_{min}(a), T_{max}(a)]$ = [10, 211] and $[T_{min}(b), T_{max}(b)]$ = [20, 221]. Since there is no update needed, the approach converges to this solution.

But if it is assumed that there is no switching window overlap at the beginning, C_c can be used as effective capacitance both for the minimum and maximum time. In this case $[T_{min}(a), T_{max}(a)]$ = [110, 111] and $[T_{min}(b), T_{max}(b)]$ = [120, 121] are used for initial solution. Since it is proved that no update is needed, thus it is also a converged solution.

This example shows that there are more than one fixpoints in Sapatnekars model, and starting from the different initial solution, different fixpoints may be got. A similar argument can be made to prove the existence of multiple fixpoints in other models.

Solution space from a complete lattice

As it is already observed from the previous section, there may be multiple fixpoints for a timing transformation T.

Since complete information is not used (functionality is not used), uncertainty is unavoidable in static timing analysis. This means that the timing information that is computed on each point is set representing the possible switching, instead of single switching. The result assures the users that the real (physical) switching is one in this set of switching's but which one is not known. A set of switching's can be represented by a switching window and having a slew in a range of slew rates such that any switching falling within the window and having a slew range is in the set. But other representations are also possible.

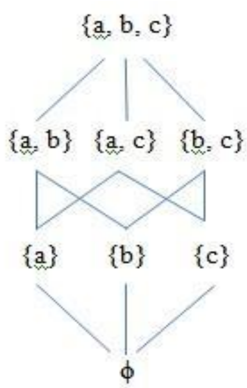

Figure 2: Subsets of a given set form a complete lattice

Now consider the family of all sets of switching at a point. The inclusion relation (that is) forms a partial order on the family. It is

i Reflexive: $A \subseteq A$;

ii Antisymmetric: $4A \subseteq B \wedge B \subseteq A \rightarrow A = B$;

iii Transitive: $A \subseteq B \wedge B \subseteq C \rightarrow A \subseteq C$

Actually, the sets of switching on a point are subsets of the whole switching set which consists of all possible switching. According to the lattice theory a partially ordered set forms a complete lattice if any subset has a least upper bound and a greatest lower bound for its elements. In fact, the family of all subsets of a given set with inclusion relation forms a complete lattice. Given a set S = {a, b, c}, the partial order of inclusion on its subsets can be represented by the Hase diagram shown in figure 2.

Here, the two sets within the inclusion relation are connected by an edge, and the lower set is included in the higher set. Timing information of a circuit is a vector of timing information on all points. The partial order on each point can be extended point wise to get a partial order on vectors : two vectors $A(A_1,A_2,\ldots, A_n)$ and $B = (B_1,B_2,\ldots,Bn)$ satisfy $A \subseteq B$ if and only if $A_i \subseteq B_i$ for all $1 < i < n$. It can be shown that the vectors with such a partial order also form a complete lattice.

Now consider a transformation T. In static timing analysis, it works on a complete lattice we defined above, that is, it transforms a vector of sets of switching to a vector of sets of switching. We say that T is a monotonic (or order preserving) transformation when, for any vector of subsets X and Y , if $X \subseteq Y$ then $T (X) \subseteq T (Y)$. The monotonicity of transformation T is based on the monotonicities of its member transformation $t_1, t_2,\ldots, t_n$ which must be true for any reasonable static timing analysis. Otherwise, it means that we can have fewer possible switching at a point while there are more possible switching at its fanins and coupling points.

Given a subset S of element in a complete lattice L, we use $\vee$ S and $\wedge$ S to represent the least upper bound and the greatest lower bound of elements in S, respectively. The existence of a x point in our system is guaranteed by the following theorem due to Knaster and Tarski (1990).

Theorem 4. (Knaster Tarski). Let L be a complete lattice and T : L $\rightarrow$ L an order preserving map. Then

$$\vee \{x \in L / x \subseteq T(x)\} \in fix(T),$$

where fix(T) is the set of fixpoints of T .

But it is not feasible to use the above theorem to compute a fixpoint since it is not feasible to compute the set $\{x \in L / x \subseteq T(x)\}$. Instead, iterative method is usually used (also called successive approximation) to find a x point. That is, an initial solution X_0 is first selected and

X_1 = T (X_0), X_2 = T (X_0), iteratively computed in the hope of finding an X_n such that X_n =T (X_n). But the hope cannot be fulled by starting from any initial point. Fortunately the bottom and the top elements are good candidates for that.

Following a tradition in lattice theory, $\perp$ and $\top$ are used to represent the bottom and the top elements of our complete lattice, respectively. That is, $\perp$ = {0, 0,…, 0} and $\top$ = {P_1, P_2, …, P_n }are available where P_i is the set of all possible switching on point i. Since T ($\top$) $\subseteq$ T , based on the monotonicity of T ,

$$T(\top) \subseteq T$$

$$T^2(\top) \subseteq T(\top)$$

$$T^3(\top) \subseteq T^2(\top)$$

Therefore a descending chain T $\supseteq$ T ($\top$) $\supseteq$ T^2($\top$)… is available. Similarly, starting with $\perp$ an ascending chain $\perp$ $\subseteq$ T ($\perp$) $\subseteq$ T^2($\perp$)… is available. If the chain has only finite elements, which is true on any finite solution space, the process will finally reach a x point. The property about $\top$ and $\perp$ that are used are : T ($\top$) $\subseteq$ T and $\perp$ $\subseteq$ T ($\perp$). Therefore, any solution X_0 such that either

X_0 $\subseteq$ T(X_0) or X_0 $\supseteq$ T(X_0) can be used as an initial solution to reach a x point. If x_0 $\subseteq$ T (X_0), we get an ascending chain; if X_0 $\supseteq$ T (X_0), we get a descending chain.

The case with infinite chains is more complex. When chains are in finite, stronger requirement are needed on T for them to converge to fixpoints.

Definition 4.2. A function T : L $\rightarrow$ L is or continuous if for any chain C, T ($\vee$ C) = $\vee$ {T (c) / c $\in$ C}, or equivalently T ($\vee$ C) = $\vee$ T (c). If T ($\wedge$ C) = $\wedge$ T (C), T is called and - continuous.

It is easy to check that an orcontinuous (or andcontinuous) function is order reversing.

Theorem 4.3. If T is or continuous, then the least fixpoint of T is $\vee_{n \geq 0} T^{n}(1)$, if T is and continuous, then the greatest fixpoint is $\wedge_{n \geq 0} T^{n}(T)$.

When a set of switching is represented by switching window, the continuity on the order coincides with the traditional continuity on real functions. That is, a continuous transformation will map a small change in input windows to a small change in output windows. This is also generally true for common transformations.

The following theorem shows that if x points are found through the above method, they must be the least and the greatest x points.

Theorem 4.4. Let L be a complete lattice, let T: L $\rightarrow$ L be an order-preserving map and define

$\alpha := \vee_{n \geq 0} T^{n}(\perp)$and $\beta := \wedge_{n \geq 0} T^{n}(T)$.

1. If $\alpha \in$ fix(T), then α is the least fixpoint;
2. If $\beta \in$ fix(T), then β is the greatest fixpoint;

Conclusion

The theory of posets and lattices has many practical applications in Timing Analysis with cross talk. Most of the result of Viswanthan et. al. (2006) is extended to many practical applications in distributed computing. For example, the concepts of Zeta Polynomial and Generating functions of posets, modular lattices, geometric lattices etc.

References

[1] B. Larose and L. Zadori, Finite posets and topological spaces in locally nite verieties, Algebra Universalis Vol. 52, no. 2, pp 119 - 136,2005.
[2] R.Viswanathan and P.Balasubramanie, Timing Analysis with cross talk, In proceeding of the National seminor on Fuzzy Mathematics and its application,pp 14,March 2006.
[3] R.Viswanathan and P.Balasubramanie and S.Ravi, Speech and Image coding using lattice based vector scalar quantizations scheme. In proceeding of the Inter National Conference on Resource Utilisation and Intelligent Systems, pp 69, January 2006.
[4] G.Gratzer,General Lattice theory. Academic Press, NewYork, NY,1978.
[5] B.A.Davey and H.A. Priestely.Introduction to Lattices and Order. Cambridge,1990.
[6] M.R. Garey and D.S. Johnson. Computers and Intractability. W.H. Freeman and Co., 1979.
[7] G.H. Golub and C.F. Van Loan. Matrix Computations. Johns Hopkins, 3rd edition, 1996.
[8] P.D.Gross, R.Arunachalam, K.Rajagopal, and L.T. Pileggi, Determination of worst-case ag-gressor alighment for delay calculation. In Proc. Intl. Conf. on computer Aided Design, San Jose, CA,pp 212 219,November 1998.
[9] E.Lelarasmee, A.E.Ruehli, and A.L. Sangiovanni Vincantelli.The waveform relaxation method for time domain analysis of large scale integrated circuits and systems, IEEE Transaction on Computer Aided Design, July 1982.
[10] S.S. Sapatnekar. A timing model incorporating the e ect of crosstalk on delay and its application to optimal channel routing. IEEE Transactions on Computer Aided Design, 2000.
[11] Y. Sasaki and K. Yano. Multi-aggressor relative window method for timing analysis including crosstalk delay degradation. In Custom Integrated Circuit Conference, pp 495 498, 2000.
[12] P.F. Tehrani, S.W. Chyou and U.Ekambaram.Deep sub-micron static timing analysis in presence of crosstalk.In international Symposiyum on quality Electronic Design, pp 505 512,2000.
[13] P.Chen, D.A Kirkpatrick, and K.K Keutzer. Switching window computation for static timing analysis in presence of crosstalk noise. In Proc. Intl. Conf. on Computer Aided Design, San Jose, CA, November 2000.
[14] R. Arunachalam, K.Rajagopal, and L.T.Pilleggi, Taco : Timing analysis with coupling.In Proc. Of the design Automation Conf., Los Angeles, Ca, pp 266 269, June 2000.