Delay Hazards in Complex Gate Based Speed Independent VLSI Circuits

Nozar Tabrizi, Michael J. Liebelt, Kamran Eshraghian‡,
Department of Electrical and Electronic Engineering
University of Adelaide
Adelaide, SA, 5005,
Australia

‡Faculty of Science and Technology
Edith Cowan University
Joondalup Drive, Joondalup, WA, 6027,
Australia

Abstract

Although speed independent[1] VLSI circuit design is supported by rich theory at higher levels, it suffers from the lack of an area efficient robust transistor level implementation technique. In this paper we introduce safe cells based on which well-formed STGs can be implemented free of (delay) hazards with no unrealistic assumptions about physical gates. Although this technique still compromises chip area for the sake of preventing hazards, we show that it may achieve a significant area gain in comparison with the two-phase RS-implementation method, which is one of the few true speed independent implementation techniques that we are aware of so far. Delay hazards are then analysed in complex gate based speed independent circuits and hence theorems are developed to identify a subclass of delay hazards.

Index terms- *Asynchronous circuits, hazards, isochronic forks, signal transition graphs (STGs), speed independent circuits (SICs).*

1. Introduction

The traditional difficulties of asynchronous design methodology, that is race and hazard problems, are eliminated in Muller's speed independent circuit theory [9] provided that the inputs comply with the ordered sequence prescribed by the state transition diagram. This theory assumes that: 1- logic operators are modelled with an instantaneous decision element followed by an arbitrary unbounded but finite lumped inertial delay, 2- all signals are absorbed in the fan out points simultaneously, that is an isochronic fork [7][14], except the small group of delay insensitive (sub)circuits [7].

The unrealistic atomic gate model (in the first assumption) for many logic operators has been the motivation for much research work to replace the unavailable gates with a combination of gates, at a wide range of cost:

Moon [10] has proposed a simple gate based design methodology in which combinational hazards are identified and possibly removed from the implementations of different nodes in the network, although this technique still suffers from delay hazards.

Beerel [1][2] has also proposed a simple gate based implementation for *distributive circuits* in which a simple gate with as many inverted inputs as required is considered an atomic element.

Martin [6] has used complex gates in his speed independent design methodology with some limited parallelism. Furthermore, it may not result in speed independence if there is over one instance of a signal transition in the compiled program sequence.

Chu [3] and Meng [8] have also used complex gates but again with as many inverted inputs as required with a negligible delay along the inverters.

Kishinevsky, et al. [5] have introduced the two-phase RS-implementation. Although this technique may suffer

1. In this article by speed independent [9] we mean, more technically speaking, semi-modular [9], as the former seems more widely used in the asynchronous circuits literature to mean no implementation deviation from specification, independent of delays of the different gates realizing the intended circuit under the unbounded gate delay model, although the original notion of speed independence is weaker.

0-8186-7502-0/96 $5.00

from some speed and area penalty due to doubling the nodes in order to introduce modelling nodes and also the modulo-2 operations for extraction of the original signals, it has two interesting features apart from its robustness against gate delays: 1-the state assignment problem is automatically solved by doubling the nodes, 2- using a simple algorithm, the logic equation for each node is easily determined with *no* need for the state transition diagram.

The second restriction of the Muller speed independent circuit theory entails negligible delay difference between different branches of a fork interconnecting a node to its multiple fan-out transistor gates, except the limited class of delay insensitive circuits [7]. The delay due to one individual branch of a fork has two major components: 1- the RC delay caused by that branch, and 2- the turn-on/ turn-off delay time resulting from (different) threshold voltages of transistors sitting at the ends of the fork and from non-zero rise/fall time of the corresponding signal transitions.

The contribution of this paper is as follows: Combinational hazards in complex gates are analysed and then safe cells are introduced based on which well-formed STGs can be implemented free of delay hazards with no unrealistic assumptions. It is further concluded that safe cells can be mixed with normal gates resulting in a flexible implementation. Finally, delay hazards are analysed in complex gate based speed independent circuits and hence a subclass of delay hazards is identified.

2. Complex logic gates as speed independent operators

A complex AND-OR-NOT gate as shown in Figure 1-a, reasonably satisfies requirements for speed independence, as no matter how close together two sequential inputs occur, they affect the circuit in the same order. Note that the only factors jeopardizing its robustness are the difference between the threshold voltages of the transistors participating in the sequential inputs and between the rise and fall times of the corresponding input

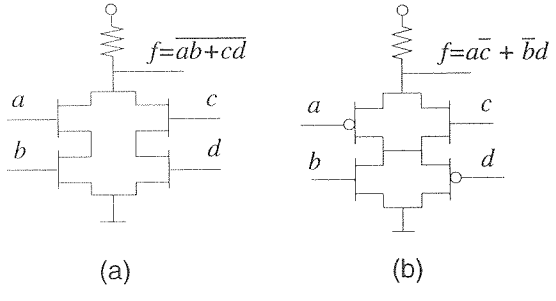

Figure 1: Two complex gates realized with (a) real, (b) ideal transistors.

signals. These differences are usually assumed to be within some tolerable extremes including the time gap between two sequential inputs, as considering the causal relationship, no two sequential inputs may occur simultaneously. Otherwise a simple NAND gate would not be a speed independent gate either.

Every gate, including safe cells (to be discussed later) may have two visible complementary outputs:

Definition 1: The *primary output* (*PO*) of a gate is the visible output generated first.

Definition 2: The complement of a primary output is called the *secondary output* (*SO*).

Complex AND-OR-NOT gates provide a complete collection of speed independent gates as long as all input signals are unipolar. However, as soon as an input gets inverted, the basic speed independent theory assumption is violated, as the inverted signal transition no longer reaches all the fan-out points simultaneously, unless the implementation process allows the use of complementary transistors in the pull down trees as shown in Figure 1-b. Moreover, inverted inputs may result in static hazards as discussed in the following section.

3. Combinational hazards in complex logic gates

Different types of hazards have been studied extensively in two stage combinational logic [13]. In this section complex gates are analysed and shown to be more robust against hazards than their normal two stage simple gate counterparts. Although complex gates have been used by some researchers as mentioned earlier, we are not aware of a systematic hazard analysis in this family so far. Furthermore, in complex gate based design strategies the delay along the inverters is assumed negligible [3] [8] or the implementation has some limitations [6] as mentioned in the introduction.

Notice that the delay model assumed in this article is the *inertial gate delay* or *speed independent model* [9], and the logic circuits considered are assumed in the form of *sum of product (SOP)* (unless otherwise specified) possibly with some inverted inputs, although taking proper duality into consideration, our discussions apply to *product of sum* implementations as well. It is also assumed that all transitions are free of function hazards.

Lemmas 1, 3 and 5 are adapted mainly from [13][11], while the rest which deal with complex gates, have been developed in this work. The proofs of lemmas are not presented here for the sake of brevity.

Lemma 1: A single input transition in simple gate SOPs is free of static 1-hazard iff both vertices are covered by a single product term (p-term).

Lemma 2: A low to high single *PO* type input transition

may not have a static 1-hazard in complex gates. A high to low transition of a *PO* type input, however, has a 1-hazard in this family iff both (on-set) vertices are not covered by a single p-term.

Lemma 3: A multiple input transition in simple gate SOPs is free of static 1-hazards iff the transition cube is covered by one p-term. ◊

In a 1-1 multiple input change, an originally disabled p-term which may be enabled during the transition, has one or more *enabling inputs*:

Lemma 4: A multiple input transition is free of static 1-hazards in complex gates iff each vertex in the transition cube belongs to an initially enabled p-term, or a p-term with no *SO* type enabling inputs.

Lemma 5: A multiple input transition in simple gate SOPs is free of 0-1 dynamic hazards iff no product term in the cover intersects the transition cube unless it also covers the final vertex as well[1].

Lemma 6: Multiple input transitions in complex gates are free of dynamic hazards. ◊

Specifically speaking, we mean *real* dynamic hazards in Lemma 6. See [12] for different types of dynamic hazards.

4. Safe cells

Definition 3: A *Petri net* is a four-tuple $G = <P, T, F, M_0>$, where P is the set of places or conditions, T represents the set of transitions, F is the flow relation: $F \subseteq (P \times T) \cup (T \times P)$ and M_0 is the initial marking.

Definition 4: In a *free choice* Petri net if a place has more than one fan-out transition, then it is the only fan in place for all its fan-out transitions.

Definition 5: A *signal transition graph* (*STG*) [3] is a free choice Petri net in which each transition is interpreted as a physical signal transition on some node of a circuit.

A STG is called *well-formed* if it is live and its corresponding state transition diagram satisfies the complete state coding property[2].

Having shown that complex gates may easily be made free of combinational hazards by introducing some redundancy, and considering the discussion in section 2, a major problem with these gates with inverted inputs is the possible undesirable change of the precedence between ordered inputs due to an inverter in a branch of an interconnecting fork or a delay hazard. For example if two sequential inputs x and y ($x^+ \rightarrow y^-$)[3] are applied to an operator z (with x inverted), then z may see the rise of x after y has gone down and this may result in a hazard and hence non-speed independence. On the other hand, a safe cell, introduced in this subsection, is so realized that its output (the effect) cannot change unless both its true and complemented input transitions (that is, the cause) have already been stabilized and absorbed. Now in the above example if y were a safe gate, then z would never see y^- before $\sim x^-$ occurs, as y would have never been pulled down if both x and $\sim x$ had not been absorbed yet in the fan out points.

A safe cell is realized with two complementary trees, *T1* and *T2*, and one 2-input Muller C-element[4] as shown in Figure 2 in which $\forall i,\ 1 \le i \le n,\ b_i = \sim a_i$.

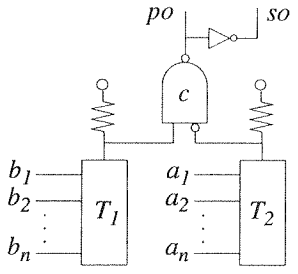

Figure 2: A safe cell.

Notice that the operation of safe cells is not dual rail coding based. Instead, they are driven by and generate normal single rail logic levels. Therefore, these cells can easily be mixed with normal complex gates or even simple gates as far as logic levels are concerned.

Definition 6: In a STG if $a^* \rightarrow b^*$, then b and a are called *immediate successor* (*IS*) and *immediate predecessor* (*IP*), respectively, of a and b, respectively.

A variable may have more than one *IS* and one *IP* in a STG.

Definition 7: A node is *driven* by the input literals applied to the logic gate realizing that node.

Definition 8: If the output of a gate is expected to change when an input transition occurs, then the output is called *excited*.

Corollary 1: Any node is necessarily driven by the *PO* or *SO* of its *IP ('s)*.

Considering this introduction the following theorem is

1. We show in [12] that multiple input transitions are free of 1-0 dynamic hazards under the speed independent delay model for SOP circuits.
2. Refer to [10] for the definition of liveness and CSC property.
3. This shows a causal relationship between x^+ (cause) and y^- (effect), where x^+ and x^- mean a low to high and high to low transition, respectively, of signal x. A signal transition on node x with an unknown direction is shown as x^*.
4. A Muller C-element behaves like an AND or OR gate if all its inputs are at logic 1 or 0 respectively. Otherwise the gate holds its previous output.

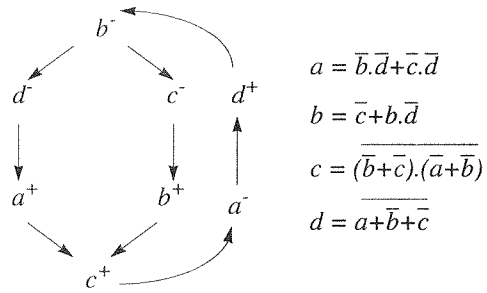

Figure 3: A STG and one corresponding group of logic equations.

proved:

Theorem 1: In order to remove delay hazards resulting from the inversion of a *PO*, *a*, it is sufficient to make all its *IS*'s, like *b*, safe if *b* is excited by *PO (a)*.

Example: In this section the STG in Figure 3 taken from [1][5][10], is implemented using both the 2-phase RS technique and safe cells, and it is shown that the second method results in a considerable area gain.

In the 2-phase RS method two modelling signals, x_1 and x_2, are introduced for every original one, *x*. Considering the equivalent two phase STG (not shown here), one consisting of all positive and the other one consisting of all negative modelling variables, the logic equation for these variables can be readily determined as follows using a simple algorithm described in [5]. The table below shows the set and reset functions[1] for the RS flip-flops realizing different nodes, as depicted in Figure 4, for node a_2 as an example.

Modelling Variables	a_1	a_2	b_1	b_2	c_1	c_2	d_1	d_2
Set	d_1	c_2	$\overline{d}_2$	c_1	b_1	$a_1.b_2$	b_1	a_2
Reset	$\overline{d}_1$	$\overline{c}_2$	d_2	$\overline{c}_1$	$\overline{b}_1$	$\overline{a}_1.\overline{b}_2$	$\overline{b}_1$	$\overline{a}_2$

Then each original signal is determined using a modulo-2 function as shown in Figure 5 for the variable *a*, as an example.

Now, the same circuit is implemented using safe cells. According to the logic equations shown in Figure 3, all *PO's* are inverted. The *IS table* in Figure 6 shows all immediate successors of the inverted *PO's* from the STG.

Next, it is checked whether any *PO* (from the above list) excites any of its own immediate successors. Such cases

1. As shown in Figure 4, the original signal corresponding to the IP ('s) of the modelling signal being generated has to be considered in the R and S functions as well. See [5] for details.

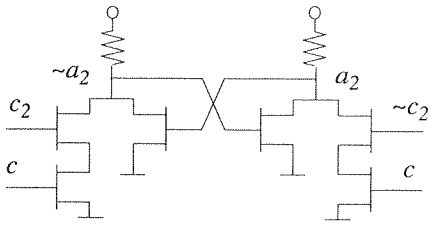

Figure 4: A modelling node, a_2, implemented as two cross coupled complex gates.

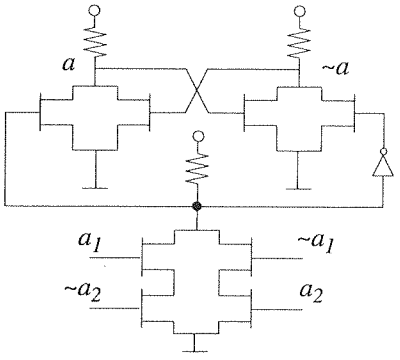

Figure 5: Modulo 2 function to extract original variables from modelling variables.

are shown with a × in the *SI (Speed Independent) table* in Figure 6. Notice that here *b* is the corresponding *PO*. Referring to Theorem 1, the SI table shows that only node *c* may need to be implemented as a safe cell.

IS Table

PO	*IS*
a	*c, d*
b	*c, d*
c	*a, b*
d	*a, b*

SI Table

IS \ *PO*	*a*	*b*	*c*	*d*
a			✓	✓
b			✓	✓
c	×	✓		
d	✓	✓		

Figure 6.

The above implementation needs only one safe cell, showing a significant area gain over the corresponding two-phase RS-implementation.

As a part of the ongoing work, we are optimizing the resulting asynchronous network to achieve the least number of safe cells required for some different sizes of the selected past history of the signal transitions as pointed out in the following section.

5. Delay hazards analysis and verification

In this section it is assumed that a node in a network

may only be affected by the change of precedence of only two consecutive transitions, that is one transition and its *IS* ('s). Based on this assumption different types of signal transitions in sequential STGs are analysed and hence the hazardous transitions are identified.

Definition 9: A STG in which all transitions belong to PO's, we call a *complete* STG. ◊

Hereafter all partial STGs are assumed to be complete unless otherwise specified.

For each pair of signal transitions like $a^* \rightarrow b^*$, two possible groups of nodes, that is, *IS* and *non-IS* nodes, possibly suffering from delay hazards, are considered in Theorem 2 and Theorem 3, respectively. Notice that delay hazards manifest themselves as dynamic and static hazards, respectively, in these two groups.

The following theorem identifies a necessary condition for a 3-stage partial STG to incur a delay hazard at the third node.

Theorem 2: The partial STG $a^* \rightarrow b^* \rightarrow c^*$ in which c is driven by both a and b, is free of delay hazards for node c if a and b do not undergo low to high and high to low transitions, respectively, where a and b are the corresponding *PO*'s.

Proof: For the (complete) partial STG $a^* \rightarrow b^*$ there are four different possible signal transition sequences shown in Figure 7, where a, b and c are *PO*'s.

	PO transitions	*Literal transitions*
1-	$a^+ \rightarrow b^+$	$a^+ \rightarrow \sim a^- \rightarrow b^+ \rightarrow \sim b^-$
2-	$a^- \rightarrow b^-$	$a^- \rightarrow \sim a^+ \rightarrow b^- \rightarrow \sim b^+$ $\nearrow \sim a^-$
3-	$a^+ \rightarrow b^-$	$a^+ \rightarrow b^- \rightarrow \sim b^+$ $\nearrow \sim a^+$
4-	$a^- \rightarrow b^+$	$a^- \rightarrow b^+ \rightarrow \sim b^-$

Figure 7: Four possible signal transitions and corresponding literal transition graphs.

On the other hand it has been shown in [10] that at least one *0-hazard* and one *high to low* (*low to high*) type product term are necessary and sufficient to have a dynamic *1-0* (*0-1*) hazard. A high to low (low to high) type product term is necessarily provided on node c by its *IP*, that is the cause of c^*. Therefore, what is required to introduce a dynamic hazard, as a result of a delay hazard, is a static 0-hazard in one of the branches. This type of hazard may only be generated through a low to high transition overtaking a high to low transition, that is an unwanted change of precedence. Notice that since only the two very last transitions are considered here, delay hazards in our case may occur only when one of the transitions b^+ or $\sim b^+$ overtakes one of the transitions a^- or $\sim a^-$. Since a^- may obviously not be overtaken, only case 3 in Figure 7, in which a and b undergo low to high and high to low transitions, respectively, may introduce hazardous transitions.◊

Theorem 3: Suppose that x is driven by the transitions $a^* \rightarrow b^*$, but is not an IS of b. Furthermore, b is excited by *PO (a)*. Two different cases are considered according to the initial state of x:

If *PO (x)* is initially at logic high, then it suffers from a delay hazard (in the form of a static 1-hazard) iff *PO (a)* undergoes a low to high transition and both signals a and b are applied in *SO* form to the same branch of node x and this branch is not disabled by another signal.

If *PO (x)* is initially at logic low, then it suffers from a delay hazard (in the form of a static 0-hazard) iff *PO (a)* undergoes a high to low transition and both signals a and b are applied in *SO* form to two different branches of node x and the p-term including b is the only enabled p-term.

Proof: Suppose that a and b are the corresponding *PO*'s. The possible sequences seen by x are as tabulated in Figure 7. The first two transition sequences in which b is excited by *SO (a)* are free of delay hazards, as there is only one choice for each of them. The next two sequences in that figure including parallel transitions, however, may cause a delay hazard as demonstrated below.

1- First consider the sequence 3 in Figure 7. The literal transitions for this case may be seen as one of the following sequences:

1.1- $a^+ < b^- < \sim a^- < \sim b^{+1}$, where the only change in the precedence is between b^- and $\sim a^-$. This, however, does not cause any hazards, as both of these transitions are in the same direction and hence tending to stabilize the output at the same logic level.

1.2- $a^+ < b^- < \sim b^+ < \sim a^-$, where the precedence change between $\sim b$ and $\sim a$ may cause a delay hazard: If $\sim b$ and $\sim a$ are applied to two different branches, the output does not experience a transition. However, if these two transitions are in the same p-term, the output might undergo a spurious logic low pulse only if it was initially at logic high.

2- Now consider the sequence 4 in Figure 7. The literal transitions for this case may be seen as one of the following sequences:

2.1- $a^- < b^+ < \sim a^+ < \sim b^-$, in which the only change in the precedence is between b^+ and $\sim a^+$. This, again, does not cause any hazards, as both of these transitions are in the

1. $a^* < b^*$ indicates a temporal relationship meaning that a^* occurs before b^*.

same direction and hence tending to stabilize the output at the same logic level.

2.2- $a^- < b^+ < \sim b^- < \sim a^+$, where the precedence change between $\sim b$ and $\sim a$ may cause a delay hazard: If $\sim b$ and $\sim a$ are applied to the same branch the output again does not experience a transition. However, if these two transitions are applied to two different branches of node x, the output may undergo a spurious logic high pulse only if the output was initially at logic low and the p-term including $\sim b$ was the only asserted p-term. ◊

Example: An incomplete STG and the corresponding logic circuit[1] for a L/R element [6] are shown in Figure 8. In this example l_o is driven by $r_i^+ \rightarrow x^+$. Notice that apart from the r_i input, the output of the Muller- C element, x, is in fact inverted as well. Also note that according to our convention l_o in this figure is an inverted output which is at logic low level. Now, considering that this AND branch is not disabled by any other signal, case 1 of Theorem 3 applies to this example and hence this transition pair is hazardous for node l_o.

Now consider node r_o which is driven by transition $l_i^- \rightarrow x^-$. Since the input l_i is not inverted, a necessary condition does not exist any more and hence the transition is free of delay hazards for node r_o.

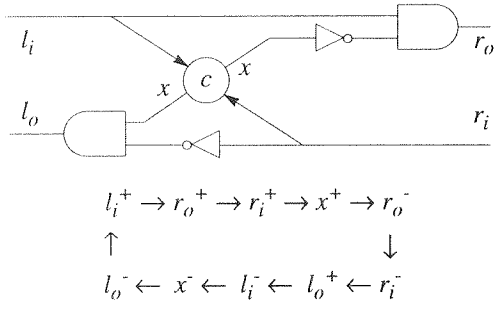

Figure 8: L/R element: logic circuit and STG.

6. Conclusion

Combinational hazards in complex AND-OR-NOT gates were discussed and it was shown that this logic family is more immune than simple gate AND-OR logic to those hazards. Therefore, delay hazards caused by the inverters in interconnection forks are one of the two[2] major drawbacks of complex gate based speed independent VLSI circuits. With this motivation safe cells were introduced and shown to be robust against delay hazards. Then, some sufficient conditions were identified to determine those nodes in a network which need to be realized as a safe cell. This method was compared with the two phase RS-implementation technique in an example and it was concluded that safe cells might result in a significant chip area gain.

In order to exclude unlikely delay hazards caused by unrealistic assumptions on different delays, we studied delay hazards in complex gate based speed independent circuits and identified a subclass of delay hazards in different nodes of the network.

1. The procedure of logic equation extraction is not discussed here. The interested reader may refer to, say, [3].
2. The other major problem is the limit on the number of stacked transistors using today's VLSI technologies [15][4].

References

[1] P. A. Beerel and T. H-Y. Meng, "Automatic Gate-Level Synthesis of Speed Independent Circuits", In Proceedings of the International Conference on Computer-Aided Design, November 1992.

[2] P. A. Beerel and T. H-Y. Meng, "Gate-Level Synthesis of Speed Independent Asynchronous Control Circuits", In Proceedings of the ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU), March 1992.

[3] T. A. Chu. "Synthesis of Self-Timed VLSI Circuits from Graph-theoretic Specifications", PhD thesis, MIT, June 1987.

[4] K. Eshraghian. "Fundamentals of Ultra High Speed Systems", Prentice Hall, to be published.

[5] M. Kishinevsky, A. Kondratyev, A. Taubin, V. Varshavsky, "Concurrent Hardware, The Theory and Practice of Self-Timed Design", John Wiley & Sons, 1994.

[6] A. J. Martin, "Programming in VLSI, From Communicating Processes to Delay-Insensitive Circuits", Chapter one, pp. 1-64, Developments in Concurrency and Communication, Addison-Wesly 1990.

[7] A. J. Martin, "Limitations to Delay Insensitivity in Asynchronous Circuits", In proceedings of the Conference on Advanced Research in VLSI, April 1990.

[8] T. Meng. "Asynchronous Design for Digital Signal Processing Architectures", PhD thesis, U.C. Berkeley, November 1988.

[9] R. E. Miller, "Switching Theory", Volume 2, Chapter 10, John Wiley and Sons, 1965.

[10] C. W. Moon, "Synthesis and Verification of Asynchronous Circuits from Graphical Specifications", PhD thesis, University of California at Berkeley, 1992.

[11] S. M. Nowick and D. L. Dill, "Exact Two Level Minimization of Hazard Free Logic with Multiple-Input Change", In Proceedings of the International Conference on Computer-Aided Design, November 1992.

[12] N. Tabrizi, M. J. Liebelt and K. Eshraghian "Dynamic Hazards and Speed Independent Delay Model", In Proceedings of the Second International Symposium on Advanced Research in Asynchronous Circuits and Systems, March 1996.

[13] S. H. Unger, "Asynchronous Sequential Switching Circuits", Wiley-Interscience, 1969.

[14] K. Van Berkel, "Beware the isochronic fork", Integration, the VLSI journal, 13 (2):103-128, June 1992.

[15] N. H. E. Weste and K. Eshraghian, "Principles of CMOS VLSI Design", Addison-Wesley, 1993.