

Metastable-robust Self-timed Circuit Synthesis from Live Safe Simple Signal Transition Graphs

Edwin C.Y. Chung and Lindsay Kleeman

Department of Electrical & Computer Systems Engineering
Monash University, Clayton, Victoria 3168, Australia

Abstract

A technique for the synthesis of metastable-robust self-timed digital circuits from Live Safe Simple Signal Transition Graphs (LSS STGs) is described. In this technique, the set of non-input choices and their related signals are first identified. Metastable behaviour of circuit elements implementing these signals is next contained using Differential Threshold Buffers (DTB), thus preventing circuit malfunctions. An LSS STG is implemented as an example.

1. Introduction

Advances in VLSI technology over the past decades have resulted in a significant scaling down in the feature sizes of integrated circuits. An effect of this is the lowering of the transit time of transistors, giving us faster devices. However, since scaling entails an increase in wire delays relative to transit time [1], scaling also aggravates clock skew as designs require faster and larger circuits. Synchronous systems with asynchronous inputs are also inherently susceptible to synchronisation failure due to the metastable failure of storage devices [2-4]. This further limits the maximum speed at which a synchronous circuit can be operated reliably. To circumvent these timing difficulties and to exploit the superior speed of scaled devices, designers are now turning towards self-timed system design methodology.

Currently, a number of techniques for the synthesis of self-timed circuits are available. Amongst these is the graph-theoretic approach using interpreted *live safe free-choice Petri Nets* known as *Signal Transition Graphs* (STGs) [5]. In this technique, the behaviour of a self-timed circuit is first described using an STG. An

implementation of the circuit is then derived using a sequence of steps briefly summarised as follows:

- State graph derivation
- Karnaugh map generation for non-input signals
- Derivation of logic expressions from the Karnaugh maps

Note that these are the principle steps for the synthesis of a self-timed circuit from its STG description. Variations of these along with procedures to eliminate circuit hazards [6, 7], state assignment problem [8] etc are frequently included.

To restrict an STG description to circuit behaviour that is both realistic and implementable, a number of restrictions have been previously employed. One of these restrictions is that only transitions of *input signals* can be specified at the output of a place with more than one output transition. This restricts an STG description to circuit behaviour involving only sequential ordering, concurrency and *input free choices*, excluding the description of circuit behaviour involving *non-input controlled choices* such as an arbitration circuit. To allow for the description of these circuits, these restrictions are weakened as follows:

- The property of *free-choice* [9] restricts a transition to at most one shared input place and these transitions to have no other input places (see Fig. 1a). This restriction is weakened to allow *simple* places (see Fig. 1b), where transitions with multiple input places can have up to one shared input place.
- The restriction that transitions with a shared input place must be transitions of *input* signals only is relaxed to include transitions of *non-input* signals. However, transitions sharing a common input place must all be of the same type (i.e., all input or all non-input and not a mixture). A transition disabled by the firing of a

conflicting transition must also remain disabled until all its input places are again non-empty (i.e., the same transition must not be enabled elsewhere when it has been disabled by the firing of a conflicting transition).

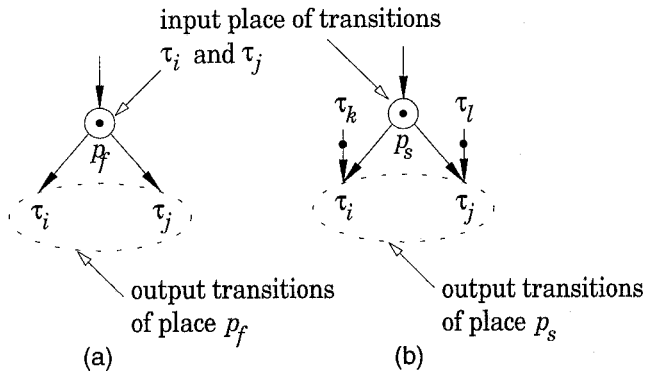


Figure 1: (a) A free-choice place, p_f , with output transitions τ_i and τ_j describing a free-choice between the firing of transitions τ_i and τ_j . (b) A simple place, p_s , specifying a controlled choice between the firing of τ_i and τ_j dependent on the firing of τ_k and τ_l .

These changes may result in metastable behaviour [2, 3, 4, 10] in the implemented circuits due to the firing of mutually exclusive signal transitions implementing non-input choices. To prevent circuit malfunctions due to metastability, a technique is presented that implements metastable-robust self-timed digital circuits from the broader class of *live safe simple* STGs (LSS STGs) using *differential threshold buffers* (DTBs).

This paper is organised as follows: In the next section, an overview of the STG notation used in this paper is presented. This is followed by a discussion of the use of simple places in STGs and the general circuit configuration associated the implementation of these descriptions. In section 4, a CMOS differential threshold buffer configuration and its characteristics are presented. The technique for the synthesis of metastable-robust self-timed digital circuits from LSS STGs is then presented in section 5, followed by an example in section 6 and conclusions in section 7.

2. Formal STG notation: an overview.

An STG description is formally denoted by $\Sigma_J = \langle P, T, F, M_0 \rangle$ where:

- J is the set of all *network signals* described in the STG, Σ_J . The set of *input* and *non-input*

signals are denoted as J_I and J_{NI} respectively with $J = J_I \cup J_{NI}$.

- P is the set of *places* in Σ_J ,
- T , the set of *signal transitions* in Σ_J ($T = J \times \{+, -\}$),
- F , the *flow relation* ($F \subseteq (P \times T) \cup (T \times P)$) such that $\text{dom}(F) = \text{range}(F) = P \cup T$ and
- M_0 is the *initial token marking* on Σ_J .

Similar to its Petri Net predecessor, the set of output transitions for a given place $p \in P$ is denoted by $p \cdot$. Likewise, the set of token markings reachable from the initial marking, M_0 , is denoted by $[M_0]$. For a given marking $M \in [M_0]$, a transition $\tau \in T$ and place $p \in P$, $M[\tau]$ denotes that transition τ is enabled in marking M , while $M(p)$ denotes the number of tokens in place p at the marking M .

The notation, $j\pm$, is used in this paper to denote either a rising or a falling transition of a signal $j \in J$. Additionally, the cardinality of a set, r , is denoted in this paper as $\#(r)$. Note that in Fig. 1, $p_f \cdot = p_s \cdot = \{\tau_i, \tau_j\}$ and $\#(p_f \cdot) = \#(p_s \cdot) = 2$.

3. The General Configuration of Circuits Implementing Non-input Choices.

In this section, the use of simple places in STGs and their restrictions are examined. The configuration of circuits implementing these descriptions are also examined.

Unlike its free-choice counterpart, a simple place does not always imply a non-deterministic choice¹. An example is the description of a linearly ordered set of transitions using a simple place in Fig. 2a. In general, a *controlled choice* specified using a set of n signal transitions sharing a simple input place describes only an m -way non-deterministic choice where $0 \leq m \leq n$. Choices between a transition, τ , and both the rising and the falling transitions of another signal can also be specified using a simple place as shown in Fig. 2b.

To contain the complexity of these descriptions to choices that can be implemented reliably, the description of choices (both free and controlled) are restricted as follows. In the case of input choices, note that these choices are implemented by the environment external to the circuit. Since it is not possible to

¹ Choices whose outcome is not deterministic can lead to metastability if these choices are non-input choices.

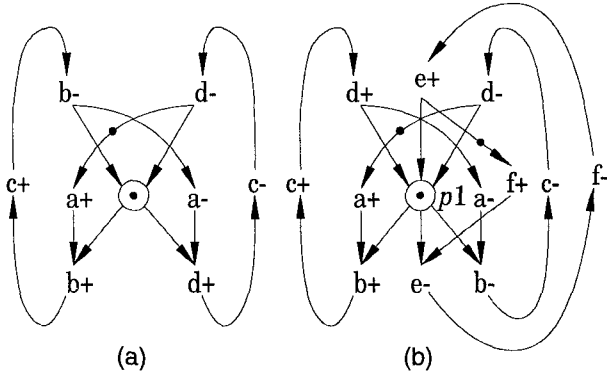


Figure 2: (a) The description of a linearly ordered set of transitions using a simple place and (b) an LSS STG containing the description of two 2-way choices, one between transition e- and transition b+ and the other between transition e- transition b-.

deduce the structure of the external circuit from the structure of the input choice described in an STG, meaningful restrictions on the description of input choices cannot be derived. The description of input choices can hence be used to simplify an STG description. However, care should be taken to ensure that the description is consistent with the behaviour of these input signals.

In the case of non-input choices, note that the firing of mutually exclusive signal transitions implementing non-input choices can result in metastability as the firing of these transitions disable one another. (Consider the firing of e- and b- in Fig. 2b) To identify the set of signals which are susceptible to metastability, we defined the M -set of an STG as follows:

Definition: M -set

Given an LSS STG, $\Sigma_J = \langle P, T, F, M_0 \rangle$, the set $\mathcal{M} = \{ \langle J_i, p_i \rangle \mid J_i \subseteq J_{NI}, p_i \in P : j \in J_i \Leftrightarrow j^\pm \in p_i \cdot \}$ is called the M -set of Σ_J if and only if $\forall \langle J_i, p_i \rangle \in \mathcal{M}$,

- $\#(p_i \cdot) > 1$ and
- $\exists M \in [M_0]$ such that at least two transitions, τ_1 and $\tau_2 \in p_i \cdot$, are enabled in marking M .

In other words, it is a collection of places describing non-input choices in the STG and the corresponding set of signals whose transitions are specified at the output of these places. As an example, notice that the M -set of the STG in Fig. 2b is $\{ \langle \{e, b\}, p1 \rangle \}$ if both e and b are

non-input signals; while the M -set of the STG in Fig. 2a is an empty set since it does not describe a choice.

To contain the complexity of circuits implementing non-input choices to cross-coupled configurations² the following restrictions on the description of non-input choices are employed.

Non-input choice restriction

Given an LSS STG, $\forall \langle J_i, p_i \rangle \in \mathcal{M}$

- (1) $\#(J_i) = \#(p_i \cdot)$
(i.e., the set of transitions specified at the output of a place describing a non-input choice are all transitions of different signals)
- (2) $j \in J_i \Rightarrow \nexists \langle J_k, p_k \rangle \in \mathcal{M} : i \neq k \text{ and } j \in J_k$
(i.e., each signal may appear in an M -set only once)
- (3) $\forall \tau_1, \tau_2 \in p_i \cdot$
 $\exists M \in [M_0] : M[\tau_1] \wedge M[\tau_2] \Rightarrow$ the firing of transition τ_1 and τ_2 are **mutually exclusive** (i.e., $\forall M' \in [M]$ such that $M[\sigma]M'$ for some $\sigma \in T^*$, $\tau_1, \tau_2 \in \sigma$ if and only if $\exists \sigma_1, \sigma_2 \in T^*$ such that $\sigma = \sigma_1 \sigma_2$ and $M[\sigma_1]M''$ where $M''(p_i) \neq 0$). In other words, a transition of a signal disabled by the firing of a conflicting transition sharing a common input place p_i can only be enabled after place p_i becomes non-empty again.

To appreciate the significance of these restrictions on the configuration of circuits implementing non-input choices, consider the following theorems.

Theorem 1

Given an LSS STG, $\Sigma_J = \langle P, T, F, M_0 \rangle$, if $\exists p \in P : p \cdot = \{ \chi_1^\pm, \chi_2^\pm \}$ and $\exists M \in [M_0] : M(p) \neq 0 \wedge M[\chi_1^\pm] \wedge M[\chi_2^\pm]$, χ_1 and χ_2 can be implemented using a cross-coupled configuration if the firing of transitions χ_1^\pm and χ_2^\pm are **mutually exclusive**.

Proof:

Firstly, note that place p may either be a free-choice or a simple place describing a choice between the firing

² The implementation for a set of signals x_1, x_2, \dots, x_n is said to have a *cross-coupled configuration* if the implementation of

$$\text{each signal has the structure } x_i = \prod_{\substack{k=1 \\ k \neq i}}^n (x_k + m_k) \cdot f_d \text{ or}$$

$$x_i = \sum_{\substack{k=1 \\ k \neq i}}^n x_k \cdot m_k + f_d \text{ where } m_d \text{ is either 1 or 0 and } f_d \text{ is the}$$

remaining expression of x_i .

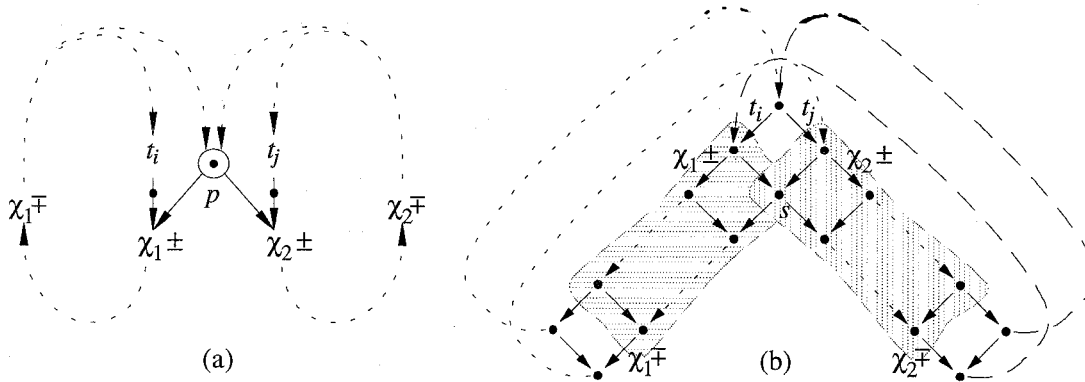


Figure 3: (a) The mutual exclusive firing of χ_1^\pm and χ_2^\pm at place p and (b) its corresponding non-semi-modular (defined in [12]) state graph structure. In the case where place p corresponds to a free-choice place, its state graph structure can be derived from the structure in (b) by collapsing all paths $s \xrightarrow{t_i} \chi_1^\pm \xrightarrow{\chi_1^\mp} s$ and $s \xrightarrow{t_j} \chi_2^\pm \xrightarrow{\chi_2^\mp} s$ to state s .

of transitions χ_1^\pm and χ_2^\pm . Translated to the state graph level, this corresponds to a non-semi-modular³ structure as depicted in Fig. 3b where the firing of χ_1^\pm and χ_2^\pm at state s disable the firing of one another. In the case where the firing of transitions χ_1^\pm and χ_2^\pm are mutually exclusive, the logical state of χ_1 (χ_2) will remain unchanged throughout the sequence of transitions between the firing of χ_2^\pm (χ_1^\pm) to the firing of its reverse transition χ_2^\mp (χ_1^\mp) as shaded in Fig. 3b. Considering that transitions χ_1^\pm and χ_2^\pm , may either be a rising or a falling transition, the structure in Fig. 3 can be further classified into three different cases, namely:

- Case 1: both transitions χ_1^\pm and χ_2^\pm are rising transitions
- Case 2: both transitions χ_1^\pm and χ_2^\pm are falling transitions
- Case 3: one of transitions χ_1^\pm and χ_2^\pm is a rising transition while the other a falling transition

Case 1:

In the case where both transitions χ_1^\pm and χ_2^\pm are rising transitions, χ_1 (χ_2) will remain unchanged at logic 0 throughout the sequence of transitions from χ_2^+ (χ_1^+) to χ_2^- (χ_1^-). Considering the mapping of a Karnaugh map from its state graph [5, 13], this indicates that the implied values of χ_1 (χ_2) are 1s only in the region within the Karnaugh map where the logical state of χ_2 (χ_1) is 0. The implementation for

both χ_1 and χ_2 derived in the sum of products (SOP) form will hence have the canonical form

$$\begin{aligned}\chi_1 &= \overline{\chi_2} \cdot f_{d1} \\ \chi_2 &= \overline{\chi_1} \cdot f_{d2}\end{aligned}$$

where f_{d1} and f_{d2} are the remaining expressions for χ_1 and χ_2 and are free from the literals χ_2 and χ_1 respectively. Using De Morgan's theorem, these expressions can be rearranged into the following form corresponding to a cross-couple NOR structure depicted in Fig. 4a.

$$\begin{aligned}\chi_1 &= \overline{\chi_2 + f_{d1}} \\ \chi_2 &= \overline{\chi_1 + f_{d2}}\end{aligned}$$

Case 2:

In the case where both transitions χ_1^\pm and χ_2^\pm are falling transitions (the reverse of case 1 above), the implied values of χ_1 (χ_2) are 0s only in the region within the Karnaugh map where the logical state of χ_2 (χ_1) is 1. Similar to case 1 above, the implementation for both χ_1 and χ_2 derived in the product of sums (POS) form will hence have the canonical form

$$\begin{aligned}\chi_1 &= \overline{\chi_2 + f_{d1}} \\ \chi_2 &= \overline{\chi_1 + f_{d2}}\end{aligned}$$

Likewise, these expressions can be rearranged into form

$$\begin{aligned}\chi_1 &= \overline{\chi_2 \cdot f_{d1}} \\ \chi_2 &= \overline{\chi_1 \cdot f_{d2}}\end{aligned}$$

³ The term *semi-modular* refers to the property where an enabled transition cannot be disabled by the firing of another transition [12]. Note that this definition differs from [11].

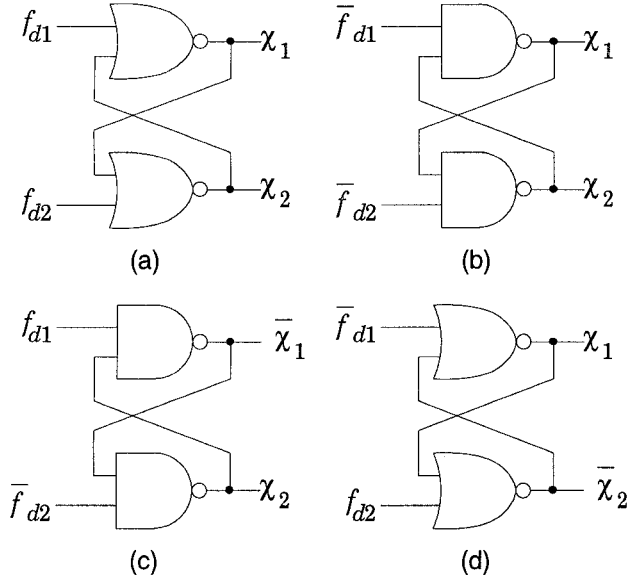


Figure 4: The different cross-coupled configurations implementing χ_1 and χ_2 where the firing of χ_1^\pm and χ_2^\pm are mutually exclusive.

corresponding to a cross-coupled NAND structure depicted in Fig. 4b.

Case 3:

In the case where one of transitions χ_1^\pm and χ_2^\pm is a rising transition and the other a falling transition, say transitions χ_1^+ and χ_2^- . The mutual exclusion between these transitions will result in χ_1 (χ_2) having implied values of 1s (0s) in the region within the Karnaugh map where the logical state of χ_2 (χ_1) is 1(0). The expressions for χ_1 and χ_2 derived in the SOP and POS form respectively will have the canonical form

$$\begin{aligned}\chi_1 &= \chi_2 \cdot f_{d1} \\ \chi_2 &= \chi_1 + f_{d2}\end{aligned}$$

Similar to the cases above, these expressions can be rearranged into

$$\begin{aligned}\overline{\chi_1} &= \overline{\chi_2 \cdot f_{d1}} & \text{or} & & \overline{\chi_1} &= \overline{\chi_2 + f_{d1}} \\ \chi_2 &= \overline{\chi_1 \cdot f_{d2}} & & & \chi_2 &= \chi_1 + f_{d2}\end{aligned}$$

corresponding to the cross-coupled NAND and cross-coupled NOR depicted in Fig. 4c and Fig. 4d respectively.

From the above analysis, note that irrespective of the nature of transitions χ_1^\pm and χ_2^\pm , due to the *mutual exclusion* between the firing of these transitions, χ_1 and χ_2 can be implemented using a cross-coupled gate configuration proving the theorem. ■

Note that this theorem is also valid for places specifying the firing of more than 2 mutually exclusive transitions. However, since the above derivation is only valid if the logical state of χ_1 (χ_2) remains unchanged throughout the sequence between the firing of χ_2^\pm (χ_1^\pm) to the firing of its reverse transition χ_2^\mp (χ_1^\mp), this theorem is not valid in places specifying the mutual exclusive firing of a transition τ with both the rising and the falling transition of another signal as depicted in Fig. 2b.

Theorem 2

Given an LSS STG, $\Sigma_J = \langle P, T, F, M_0 \rangle$, and its M -set, \mathcal{M} , the implementation for the set of signals in J_i for some $\langle J_i, p_i \rangle \in \mathcal{M}$ has a cross-coupled gates configuration if the non-input choice restriction above is satisfied.

Proof:

Firstly, notice that for a pair of $J_i \in J_{NI}$ and $p_i \in P$ such that $\langle J_i, p_i \rangle \in \mathcal{M}$:

- (1) $\#(J_i)$ denotes the number of signals whose transitions are at the output of p_i . Hence, $\#(J_i) = \#(p_i \cdot)$ implies that \exists a signal $j \in J_i$ such that both transitions j^+ and j^- are elements of $p_i \cdot$. Since $\langle J_j, p_j \rangle$ corresponds to a choice between the firing of transitions in $p_i \cdot$, this also implies that $\forall j' \in J_i$, \exists a signal $j \in J_i$ such that the firing of j'^\pm and j^+ , and the firing of j'^\pm and j^- are mutually exclusive at place p_i .
- (2) If $j \in J_i \Rightarrow \exists \langle J_j, p_j \rangle \in \mathcal{M} : j \in J_j$, this will restrict the set of $J_i \forall \langle J_j, p_j \rangle \in \mathcal{M}$ to a disjoint sets. In other words, $\forall j \in J_i$, the firing of the transition for signal j is only mutually exclusive with the transition of other signals $j' \in J_i$.

Collectively, these imply that, $\forall \langle J_i, p_i \rangle \in \mathcal{M}$, the transition of a signal $j \in J_i$ is mutually exclusive of the transition of another signal $j' \in J_{NI}$ if and only if $j' \in J_i$. Additionally, there will not exists a signal $j \in J_i$ whose transition is mutually exclusive of both the rising and the falling transition of another signal $j' \in J_i$. Using Theorem 1 above, due to the mutual exclusion between the firing of their transitions, the implementation for the set of signals J_i can hence be implemented using a cross-coupled gate configuration. ■

Theorems 1 and 2 hold for m -way non-input choices where $m \geq 2$. For reasons of clarity and simplicity, we

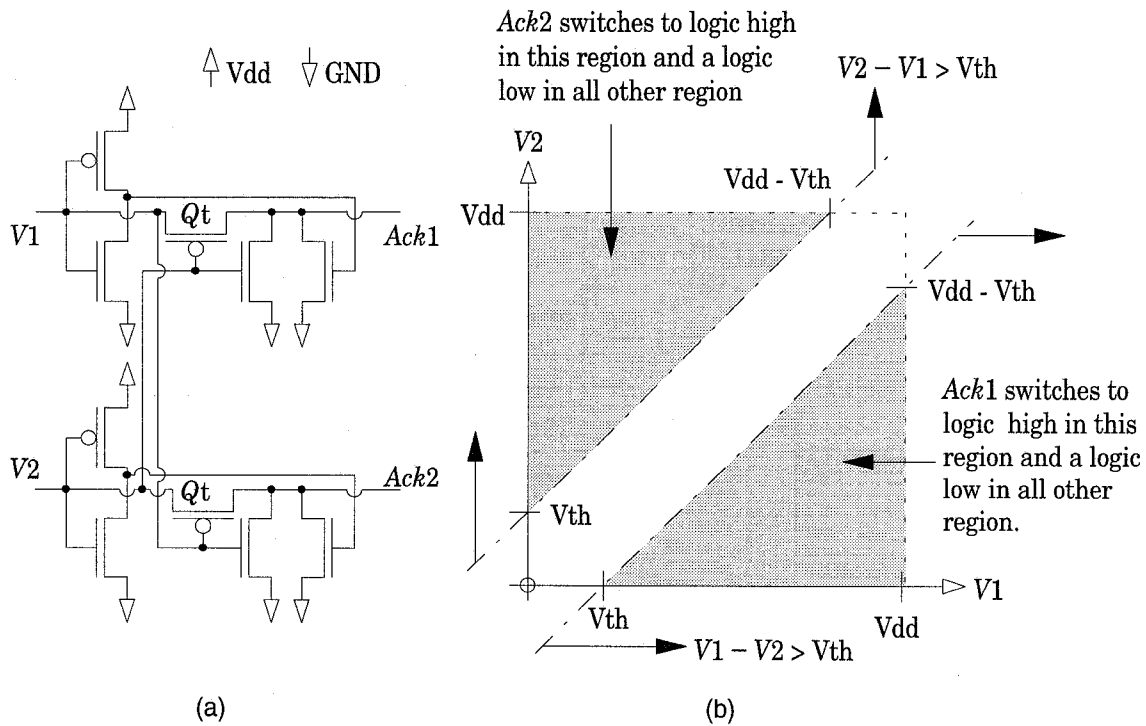


Figure 5: (a) A two input CMOS DTB and (b) its characteristic.

present examples and circuits for 2-way choices ($m = 2$).

4. CMOS Differential Threshold Buffer

The implementation of mutually exclusive non-input choices using the cross-coupled gate configuration can result in metastable behaviour. As an example, consider the concurrent firing of transitions χ_{1+} and χ_{2+} at state s of Fig. 3b in the configurations in Fig. 4 above. Due to the mutual exclusion between the firing of these transitions, note that the firing of χ_{1+} will disable the firing of χ_{2+} and vice versa. This may drive the cross-coupled into a metastable state and may also result in malfunctions if the metastable behaviour of χ_1 and χ_2 is allowed to influence other parts of the circuit. To isolate the metastable behaviour of the cross-coupled from the rest of the circuit, and hence avoid circuit malfunctions, we have developed a CMOS *Differential Threshold Buffer* (DTB) as depicted in Fig. 5a. The characteristic of the DTB is briefly summarised as follows:

- (1) Both Ack_1 and Ack_2 remain at logic 0 while V_1 and V_2 are within V_{th} (threshold voltage of transistor Q_t) of each other, $|V_1 - V_2| < V_{th}$.
- (2) Ack_1 becomes logic 1 while Ack_2 remains at logic 0 in the region where $V_1 - V_2 > V_{th}$.
- (3) Ack_2 becomes logic 1 while Ack_1 remains at logic 0 in the region where $V_2 - V_1 > V_{th}$.

For an active low configuration, the dual circuit is shown in Fig. 6. Note that these DTB configurations can be easily extended to more than two inputs but with a corresponding deterioration of speed. See the Appendix for details. An nMOS DTB configuration can be found in [1].

Here, it is assumed that the cross-coupled is designed so that the metastable region is extremely unlikely to be re-entered, due to circuit noise, once the outputs differ by more than the threshold V_{th} of the MOS transistors. To achieve reliable performance, the cross-coupled stage should be designed with the DTB as one unit.

5. A Technique for Implementing Live Safe Simple STGs.

Utilising the above theorems and restricting the description of non-input choices in an LSS STG as specified earlier, it has been shown that circuits implementing these choices have cross-coupled gates configuration. Due to the non-semi-modular structure of these choices, the cross-coupled configurations implementing non-input choices may result in metastability while the firing of mutually exclusive

transitions disables each other. However, a metastable-robust implementation can be derived if the metastable characteristic of the cross-coupled is contained within the cross-coupled using DTBs. A technique for the synthesis of metastable-robust self-timed digital circuit from LSS STGs is developed along this approach.

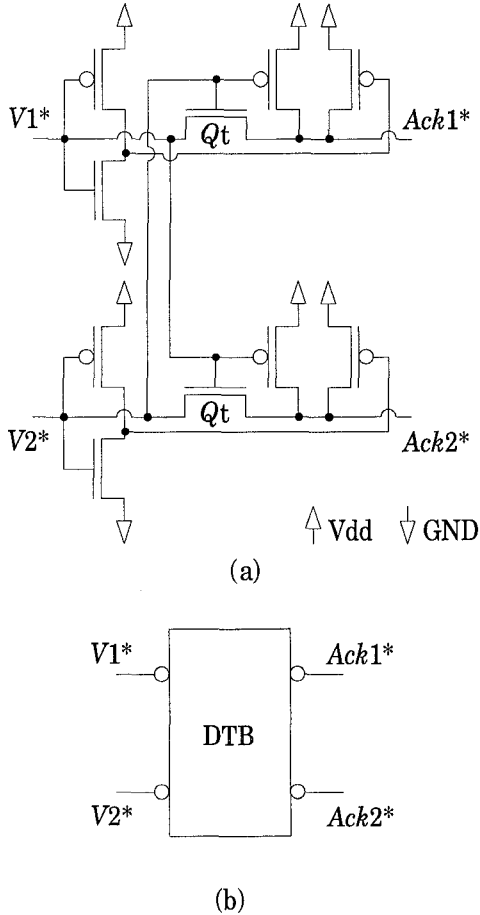


Figure 6: (a) The dual DTB configuration of Fig. 5a and (b) its block representation.

Procedure: *Metastable-robust implementation of LSS STGs.*

Given an LSS STG, $\Sigma_J = \langle P, T, F, M_0 \rangle$, satisfying the above non-input choice restriction

- (1) *Determine its M-set, \mathcal{M}*
- (2) *Implement all the non-input signals.*
- (3) *For all $\langle J_i, p_i \rangle \in \mathcal{M}$, rearrange the logic expressions of the set of signals J_i into an equivalent cross-coupled form.*

- (4) *Implement the set of signals J_i using an appropriate cross-coupled/DTB combination with the output of the DTB now taken as the set of clean signals implementing J_i .*

6. An Example.

As an example, consider the LSS STG description in Fig. 7a. Note that the M-set for the STG is $\mathcal{M} = \{\langle A1, A2 \rangle, p1\}$. Using a previously published procedure [13], the logic expressions derived for all the non-input signals are:

$$\begin{aligned} A1 &= \overline{A2} + \overline{R1} \\ A2 &= \overline{x} \cdot (\overline{z} + y) + \overline{A1} \\ x &= \overline{A2} \cdot z \\ z &= z \cdot \overline{y} + \overline{R2} \end{aligned}$$

Rearranging the expressions for A1 and A2 into the cross-coupled form

$$\begin{aligned} A1 &= \overline{A2} \cdot \overline{R1} \\ A2 &= \overline{A1} \cdot \overline{x} \cdot (\overline{z} + y) \end{aligned}$$

and using the DTB configuration in Fig. 6, this leads to a metastable-robust implementation of the LSS STG in Fig. 7b.

7. Conclusions

A technique for the synthesis of metastable-robust self-timed digital circuits from *live safe simple signal transition graphs* using *CMOS differential threshold buffers* has been described. In this technique, the description of non-input choices is restricted to contain the complexity of these implementations to cross-coupled configurations. Metastable characteristic of these configurations is isolated from the rest of the circuit using an appropriate *differential threshold buffer*. To achieve reliable performance, the cross-couple/DTB configuration should be designed as one unit.

8. References.

- [1] Seitz, C.L., "Chapter 7: System Timing" in Mead, C. and Conway, L. *Introduction to VLSI Systems*, Addison-Wesley, Reading, Massachusetts, 1980, pp. 218-262.

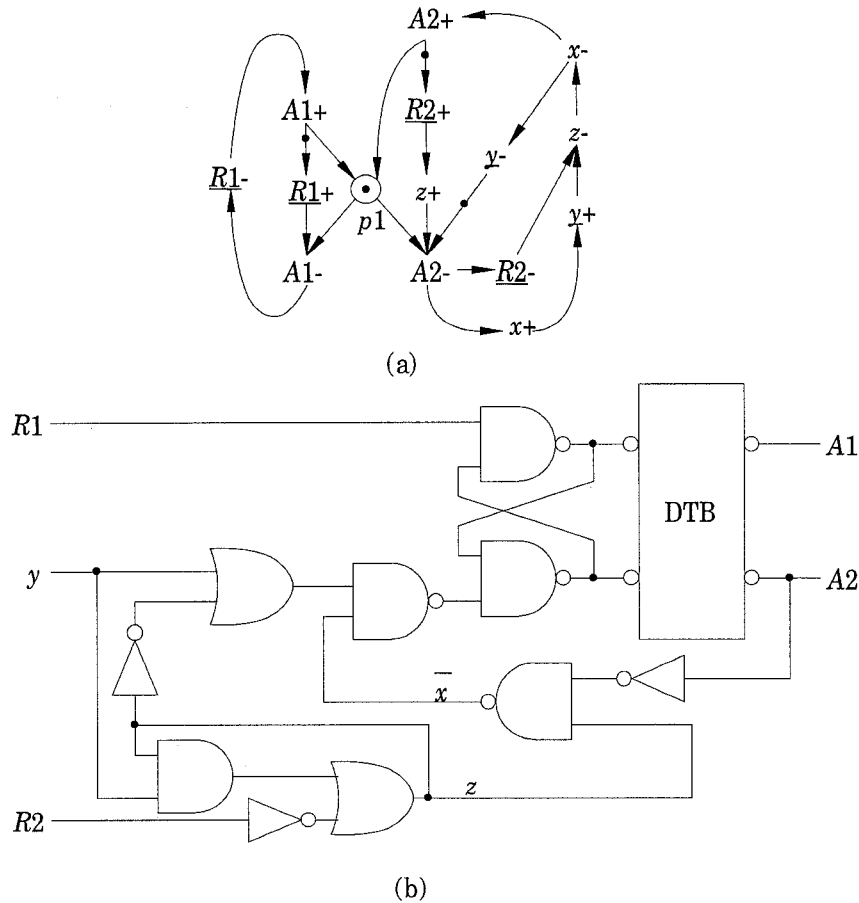


Figure 7: (a) An LSS STG and (b) its metastable-robust implementation using an active low DTB.

- [2] Chaney, T.J. and Molnar, C.E.: "Anomalous Behaviour of Synchronizer and Arbiter Circuits," *IEEE Trans. on Comp.*, 1973, C-22(4), pp. 421-422.
- [3] Kleeman, L. and Cantoni, A.: "Metastable Behaviour in Digital Systems," *IEEE Design & Test of Comp.*, 1987, 4(6), pp. 4-19.
- [4] Kleeman, L. and Cantoni, A.: "On the Unavoidability of Metastable Behaviour in Digital Systems," *IEEE Trans. on Comp.*, 1987, C-36(1), pp. 109-112.
- [5] Chu, T.A.: "Synthesis of Self-timed VLSI Circuits from Graph-theoretic Specifications," Ph.D. dissertation Department of Electrical and Computer Science, Massachusetts Institute of Technology, June 1987.
- [6] Lavagno, L., Keutzer, K. and Sangiovanni-Vincentelli, A., "Algorithms for Synthesis of Hazard-free Asynchronous Circuits," *Proc. 28th ACM/IEEE Design Automation Conference*, San Francisco, California, USA, June 17-21, 1991, pp. 302-325.
- [7] Chung, E.C.Y. and Kleeman, L., "Avoiding Hazards in Self-timed Digital Circuits Derived from Signal Transition Graphs", *Technical Report MECSE-94-1*, Department of Electrical and Computer Systems Engineering, Monash University, February 1994.
- [8] Lavagno, L., *et al.*, "Solving the State Assignment Problem for Signal Transition Graphs," *Proc. 29th ACM/IEEE Design Automation Conference*, Anaheim, California, USA, June 8-12, 1992, pp.568-572.
- [9] Thiagarajan, P.S. and Voss, K.: "A Fresh Look at Free Choice Nets," *Information and Control*, 1984, 61(2), pp. 85-113
- [10] Kleeman, L.: "Service and Metastability performance of Arbiters", Ph.D. dissertation Dept. of Electrical and Computer Engineering, University of Newcastle, Australia, August 1986.

[11] Meng, T.H.-Y., Brodersen, R.W. and Messerschmitt, D.G., "Automatic Synthesis of Asynchronous Circuits from High-level Specification," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 8, No. 11, November 1989, pp. 1185-1205.

[12] Miller, R.E., *Switching Theory Volume II: Sequential Circuits and Machines*, John Wiley & Sons, New York, 1965, Chapter 10.

[13] Chung, E.C.Y. and Kleeman, L.: "An Optimal Approach to Implementing Self-timed Logic Circuits from Signal Transition Graphs," *Australian Telecommunication Research*, 1993, 27(2), pp. 41-56.

Appendix

The basic component of the 2 input DTB configuration in Fig. 4a and its characteristics are summarised in Fig. 8. Note that the equivalent 3 input configuration is depicted in Fig. 9.

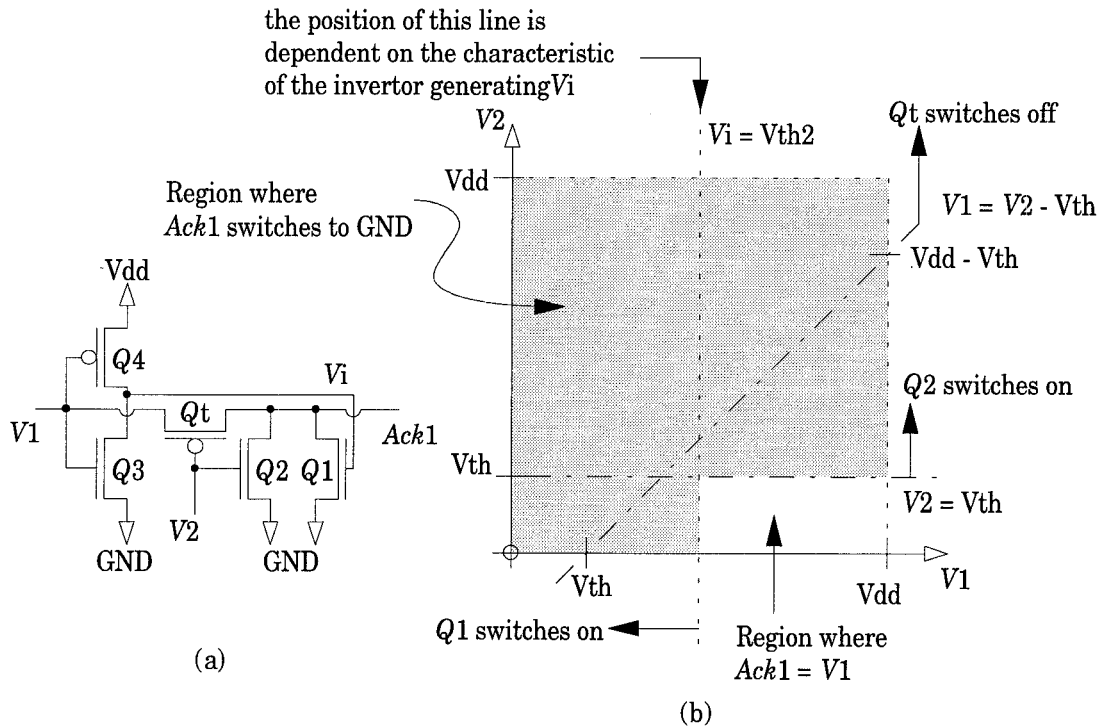


Figure 8: (a) The basic component of the DTB configuration in Fig. 5a – one is required for each input and (b) its characteristics.

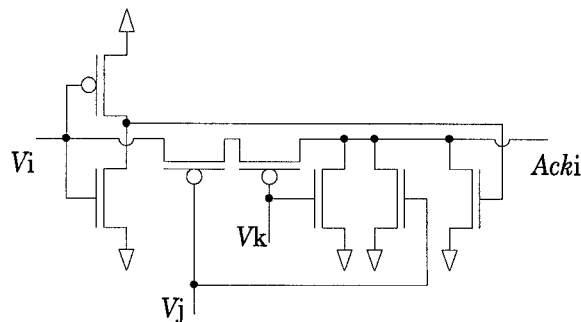


Figure 9: The basic component of a 3 input CMOS DTB – three of these are required with Ack1, Ack2 and Ack3 obtained by substituting $i j k = 1 2 3, 2 3 1, 3 1 2$. respectively.