A Design of Asynchronous Control Circuits based on SDI Model

Pitchayapatchaya Srikram and Arthit Thongtak

Abstract— **This paper presents a Scalable-Delay-Insensitive (SDI) optimization which is an approach to implement asynchronous control circuits described in Signal Transition Graph (STG). This SDI circuits can be guaranteed their correct operations in the specified orders of signal transition. According to our method, we can reduce the overhead from the original STG without affected properties, such as complete state coding (CSC), persistency and consistency.**

Index Terms—**Signal Transition Graph (STG), Logic Synthesis, Asynchronous Control Circuits, Scalable-Delay-Insensitive Model (SDI model)**

I. Introduction

Unbounded Gate and Wire Delay models play an essential role in concurrent circuit design. The Asynchronous circuit is determined that all the signal transitions need their causal relations with each other. This property can guarantee the completion of signal transition which can be protected to incorrect circuit operation. On the other hand, if the asynchronous design based on delay model is too optimistic such as gambling on fixed or no delays, it may face to incorrect circuit operations. To solve the problem, [1] proposes the quasi-delay-insensitive (QDI) model has an "Isochronic-Fork" qualification but it is required additional signals or circuits to generate the completion of signals. Consequently the circuits become big and complex. SDI (Scalable-Delay-Insensitive Model) is developed from QDI [2] which assumes the relative delay ratio between any two components is bounded [3]. In the other word, it is significant that the one of signal transition which is completely changed before another signal by fan-out branch since it do not need a circuit to detect the completion signal.

However, the previous researches were investigated only Asynchronous Combination Circuits based on SDI model [4]. In this paper we propose an expansion, SDI optimization for synthesis asynchronous control circuit based on SDI model using Signal Transition Graph (STG). Our method focuses on the STG by relieving causal relations in a certain assumption. As the result, the SDI circuits become simplified but also operate correctly. However, STG need to be checked its properties at first as shown in fig.1. Otherwise STG cannot do SDI optimization. Then, the circuit Implementation based on S.Park method which is used on [4]. Experimental results are shown the improvements in terms of area and the number of non-primary input signal. The paper is organized in the following way. Section II discusses some basic concepts of STGs. In section III, we define Signal Transition Graph and Concurrent transition Model. Section IV discusses about our SDI optimization. Sections V discusses about the impact of STG properties from the SDI optimization. Experimental results and conclusions are given in Sections VI and VII, respectively.

Manuscript received January 8, 2014; revised January 27, 2014.

P. Srikram and A. Thongtak are with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand e-mail: Pitchayapatchaya.s@hotmail.com, Arthit.t@chula.ac.th.

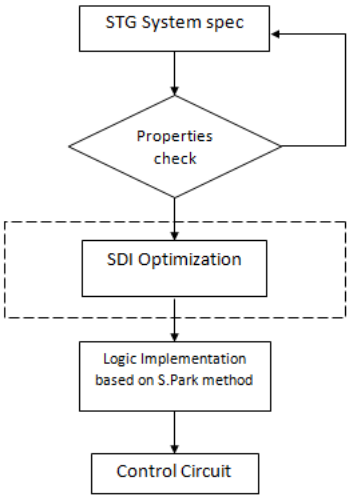

Fig.1. Design flow for SDI optimization

II. SIGNAL TRANSITION GRAPH

A design of Asynchronous circuits is described the characteristic by Signal Transition Graph (STG) [4], [5]. As the fig.3 shown, The signal transition is represented to vertices in which rising and falling transitions of the signals, respectively denoted by x^+ and x^- of signal x. x^* denote transition of signal x either x+ or x-.The causal relation between signals is represented to arc which a transition y^+ precedes a transition x^- (i.e. $y^+ \rightarrow x^-$) .This relation is also called order relation. The two signal transitions are conducted the coincident signals on STG, is represented to concurrent relation (i.e. $y^+ || z^+$, $y^- || z^-$) means that y^* and z^* are fire simultaneously or y^* is fired before z^* or in vice versa. Primary input signal is denoted the signal transition

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

with underline (i.e. $\underline{y}^*$) .However, non-primary input is denoted the signal transition without underline (i.e. x^*).The marking or place (●) is interpreted as enabled signal transitions.

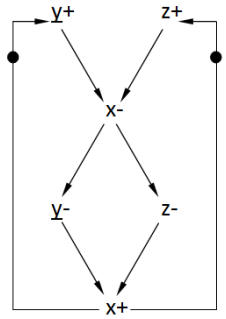

Fig.2. Simple STG

In this paper, we focus on three important properties of STG logic implementation as show below

-- Complete State Coding (CSC), an STG is said to be CSC if, every signal transition is produced the different binary code.

-- Consistency, an STG is consistent if, the signal transition is not risen or failed more than one time.

-- Persistency, an STG is persistent if, the signals are enabled in each other.

III. Scalable-Delay-Insensitive Model and Concurrent Transition Model

A. Scalable-Delay-Insensitive Model

Scalable Delay Insensitive model refers to a gate or an interconnection wire between two gates as shown in Fig.3, Both t1 and t2 can be caused by t0 but relative ratio between any two Components (Paths) is bounded by given K factor.

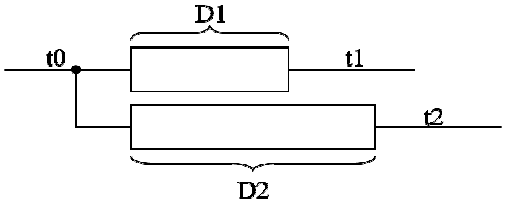

Fig.3. SDI Modeling

SDI model is given definition as below [7]; consider a component *C* in a circuit. Let *De* denote the delay that is estimated for *C* by the designer at design phase, and *Da* denote a delay that actually occurs for *C* at a specific time t during the system's lifetime. Ratio $R = Da/De$ is referred to as the scaling ratio at time t for component *C*. Next, consider any two components *C*1 and *C*2 in the circuit. Let *R*1and *R*2 denote the scaling ratios at time t for *C*1and *C*2, respectively. Ratio $V = R2/R1$ is referred to as the scaling variation at time t between *C*1and *C*2. Then, the SDI model assumes that the scaling variation *V* between any two components is bounded, that is $1/K \leq V \leq K$, though the system's lifetime, where *K* is a constant such that $K > 1$. Constant *K* is called the variation factor. An SDI circuit operates correctly on this model with the variation factor *K* appropriately specified. In addition to the SDI Modeling, asynchronous control circuits can be modelized as shown in fig.4, *t*0 is caused to *t*1 and *t*2. *t*1 denote a non-primary input signal. *t*2 denote a primary input. SDI model operate by handshaking between system and environment. The delay of primary input is included the delay from environment (i.e. *D*2), which is unknown bounded delay exactly. In this study is assumed that primary input is slower than non-primary input when asynchronous control circuit is took its input from primary input and non-primary input. Let *D*2*e* denote the estimate delay for primary input and *D*1*e* denotes and the estimate delay for non-primary input. Therefore, if we rewrite the inequality of SDI model between two type of signals which the estimate delay is $D1e < D2e$ and given the *K* value for guarantees to actual delay. There are expressed as: $K \bullet D1e < D2e$ and then actual delay will be $D1a < D2a$, i.e. non-primary input precedes primary input. However, SDI implementation is interpreted as STG characteristic by Concurrent Transition Model to do the SDI optimization as shown in fig.5

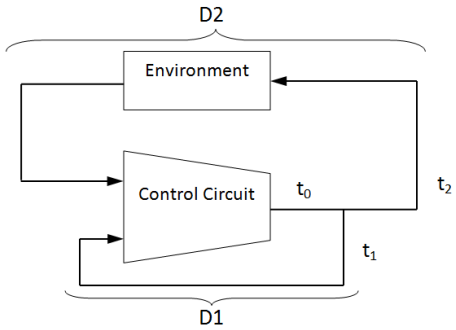

Fig.4. SDI Implementation

B. Concurrent Transition Model

Fig.5, Concurrent transition model is defined as a set of parent signal and concurrency between primary input signals and non-primary input signals. This work is specified to non-primary signals precede primary signals by *K* factor but is not estimated the each delay of the device in advance.

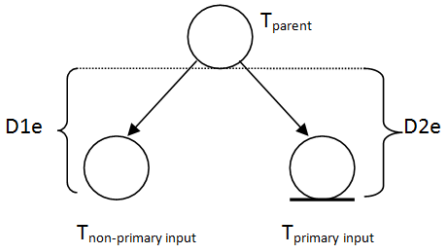

Fig.5. STG based on SDI Implementation

-- Parent transition signal denotes T_{parent}, which is $v \stackrel{def}{=} v\rightarrow (\bullet e_i \cup e_i \bullet)$.

-- Children transition signal denote e_i, which is primary input signal or non-primary input signal.

-- $T_{non\text{-}primary\ input}$ denotes e_{np}, thus the set of $e_{np(n)}$ is $\{e_{np(0)}, e_{np(1)},\ldots e_{np(n+1)} \mid n = 0,1,\ldots,n+1\}$.

-- $T_{primary\ input}$ denotes e_p, thus the set of $e_{p(n)}$ is $\{e_{p(0)}, e_{p(1)},\ldots e_{p(n+1)} \mid n = 0,1,\ldots,n+1\}$.

IV. SDI Optimization

In this section, we describe how SDI optimization is modified the path signal from the STG based on bounded

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

delay by SDI Optimization which consists of three steps: First step is determination beginning signal of the STG, Second step is STG reduction pattern based on SDI and third step is the heuristics used to repair STG as described in the following;

A. *Determination of an initial signal of STG*

Determine the signal be the parent transition in which the concurrent transition Model is guaranteed the delay between two signals by *K* factor. This is described in following Algorithm I:

Algorithm I

Input Original STG
Output Concurrent Transition Model
Do define initial parent signal transition(v)
Then estimate delay for e_i by K factor
if $[e_i \ni (e_{np(n)} \| e_{p(n)})]$
Concurrent Transition Model
else define the new parent transition (v′)

B. *STG reduction pattern based on SDI*

STG Reduction pattern based on SDI is reduced the slower signal that estimate from Concurrent Transition Model. Since, it was be lost the causal relation, to replenish with this relation by adding the new signal. This can be divided into two main categories as relation on primary input signal and non-primary input signal which is simultaneously fired.

1) Primary input signal is concurrent to non-primary input signal: This is described in Algorithm II and given as procedure example shown in fig.6 (a) original STG (b) STG on reduction pattern

Algorithm II

Input Concurrent Transition Model
Output STG reduction pattern
while Concurrent Transition Model
if [$v(v') \rightarrow e_i \,|\, e_i \in$ set of $\{e_{np} \| e_p\}$]
Then(1) reduce arc is set of $\{(v(v'),e_p) \,|\, v(v') \rightarrow e_p \} \equiv \{ v(v') - e_p \}$
Then(2) add arc is set of $\{(e_{np},e_p) | e_{np} \rightarrow e_p \}$
Then(3) Let $e_{np} := v'$

return (v′ ∈ Concurrent Transition Model)

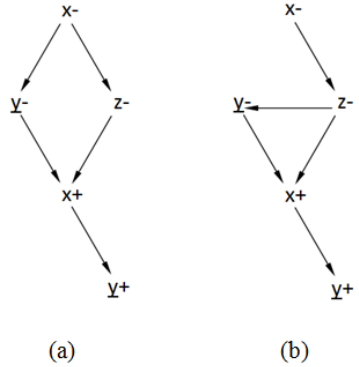

Fig.6 STG (a) initial (b) SDI Optimization

2) Primary input signals are concurrent to non-primary signals: The sum of any signals which is over or equal other signals would result in Algorithm III and be given a procedure example as primary input signals are concurrent to non-primary signal which sum of primary input signals and non-primary input signals are equal as shown in fig.7. Then, primary input signals are concurrent to non-primary signal ,the sum of non-primary input signals are more numerous than primary input signal as shown in fig.8

Algorithm III

Input Concurrent Transition Model
Output STG reduction pattern
while Concurrent Transition Model
if [$v(v') \rightarrow e_i \,|\, e_i \ni (e_{np(n)} \| e_{p(n)}) \wedge e_i \ni \#\sum e_{np(n)} \neq \#\sum e_{p(n)})$, $n = 0,1,..,n+1$]
or
[$v(v') \rightarrow e_i \,|\, e_i \ni (e_{np(n)} \| e_{p(n)}) \wedge e_i \ni (\#\sum e_{np(n)} = \#\sum e_{p(n)})$, $n = 0,1,..,n+1$]
Then(1) reduce arc is set of $\{(v(v'),\forall e_p) \,|\, v(v') \rightarrow e_p \} \equiv \{ v(v') - e_p \}$
Then(2) add arc is set of $\{(\forall e_{np} ,\forall e_p) \,|\, e_{np} \rightarrow e_p \}$
Then(3) Let $e_{np(n)} := v'$

return (v′ ∈ Concurrent Transition Model)

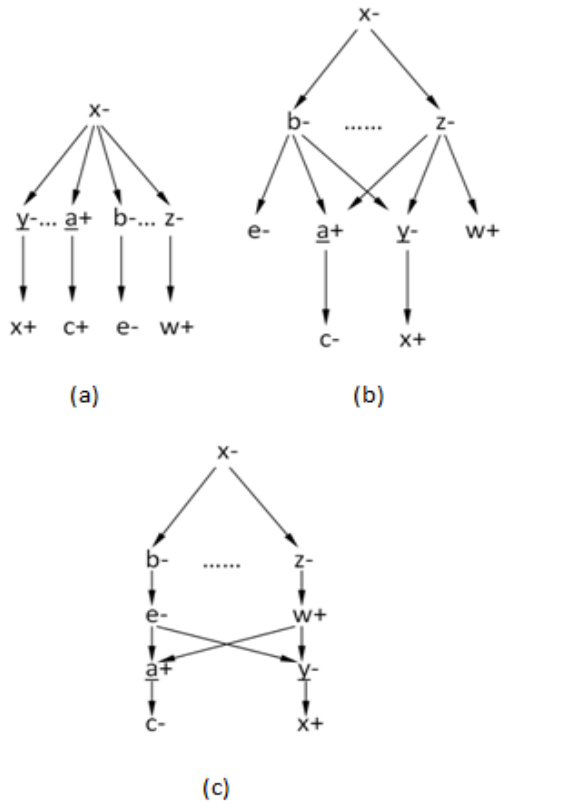

Fig.7 STG (a) initial (b) SDI Optimization and (c) Final SDI Optimization

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

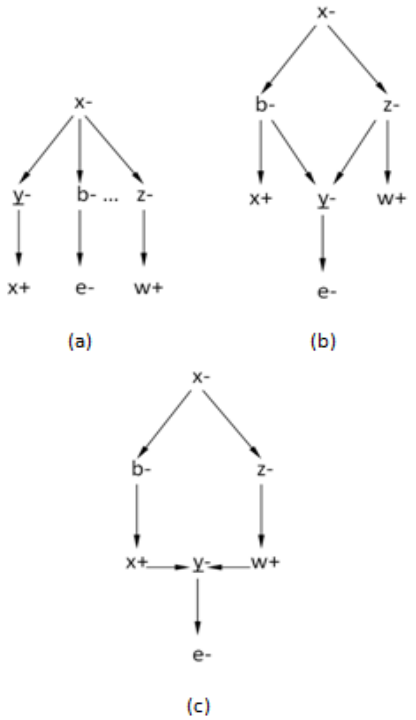

Fig.8 STG (a) initial (b) SDI Optimization and (c) Final SDI Optimization

C. The repairing STG

Since adding the new signal for handle the causal relation that it may be made the redundant signal on a part of STG. There is reduced the arc whether the signal transition path is minimization path from parent transition than another for the circuit speed might not be get slower. According to [8], the STG is behavior that x^+ and $\underline{y}^-$ are concurrent and both are caused by z^- signal transition to fire in the original STG as shown in fig.9(a). The adding arc of causal relation between z^- and $\underline{y}^-$ is caused by Algorithm II or Algorithm III (i.e. $z^- \rightarrow \underline{y}^-$). Since, we have to estimate the path from z^- to x^+. In this case, we can divide into two main paths. The first path is interleaved as $z^- \xrightarrow{l1} y^- \xrightarrow{l2} x^+$. The second path is ordered as $z^- \xrightarrow{l3} x^+$. Therefore, there is the shorter path than the first past which is reduced.

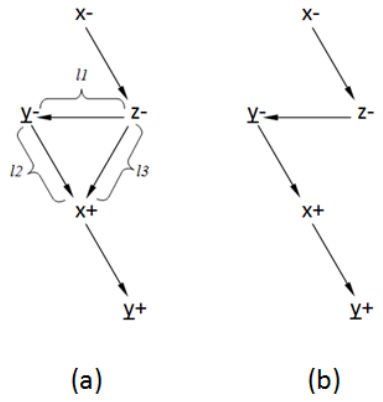

Fig.9 The part of STG (a) Estimate minimization path (b) Reduction the shorten path.

V. Impact of STG properties from SDI Optimization

STG properties are avoided an asynchronous circuit from hazard and deadlock. The study of SDI optimization was to examine whether the reduction and insertion the signal transition cannot affect STG properties. Indeed, a meaning of concurrent relation is fired simultaneously or one is fired before another in vice versa. If an original STG is correctly, to transformation STG is be not effect to STG properties. This section is described the reason in following the properties;

A. Consistent and Persistent

STG is persistent thereby it can be consistent; STG is non-persistent because the signal transition is concurrent to its trigger signal transition. STG is non-consistent because signal transition is not interleaved to risen signal then failed signal. i.e. risen signal is concurrent to failed signal. Figure10 (a) signal y^+ is a trigger signal of signal a^+ while signal a^+ is concurrent to y^-, thus signal a is non-persistent signal. This STG cannot be continued SDI optimization.

Our method is transformed the relation of signal transition from concurrent to order. It can avoid to consistent violent and persistent violent.

B. Complete State Coding (CSC)

The actual CSC violation is caused by pair $\{x^+,x^-\}$ of interleaved signal transition x which is concurrent to any signal transition as shown in fig.10(a) signal y^- is concurrent to pair of$\{z^-,z^+\}$. There is indicated the CSC conflict at State Graph domain which have the same binary representation as shown in fig 10(b).

Our method cannot modified the signal transition is a pair of interleaved signal transition, if it has this relation a precedent SDI Optimization.

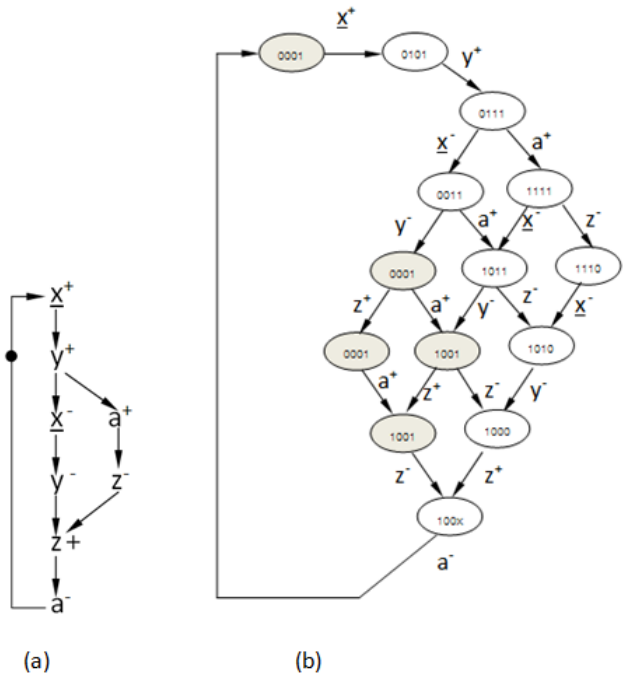

Fig.10 non-persistent, non-consistent and CSC violation (a) STG (b) State Graph

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. Experimental Result

In this section, we compared a result of STG based on SDI Optimization with STG based on QDI. Figure 11 shows the experimental result of Full Adder circuit at (a) STG based on QDI and (b) STG based on SDI Optimization. The number of arcs at STG based on SDI model might be decrease from STG based on QDI, thereby decreasing area of implementation circuit in fig.12 (b). Other STG benchmarks which result obtained from implementation circuit can be compared area of circuit in the Table I. The correlation between QDI implementation and SDI implementation is interesting because the SDI Optimization is effective with signal transition which is concurrent relation is cube as shown in fig.6(a).This behavior may decrease the arcs than the long chain as shown in fig.7(a) and 8(a).

Our method can only not affect I/O circuit. In the other hand, it cannot alternate the input of circuit to another, but also not affect the precedent STG properties.

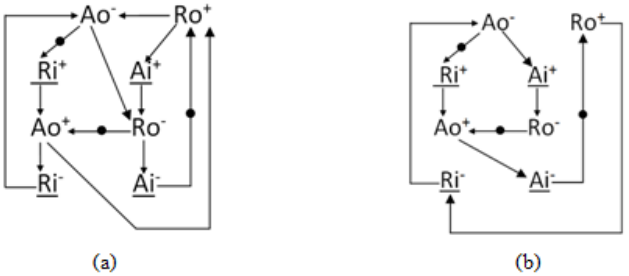

Fig11 (a) STG based on QDI (b) STG based on SDI

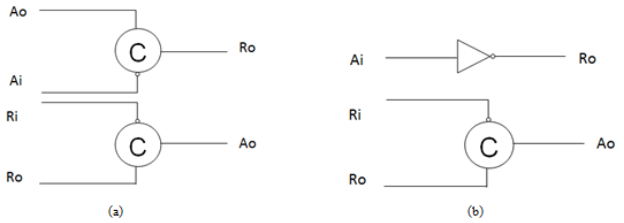

Fig.12 (a) its QDI Implementation (b) its SDI Implementation

Table I *The comparing with QDI Implementation*

Example	SDI implementation			QDI implementation		
	#non-primary input Signal	#Gate	#C-element	#non-primary input Signal	#Gate	#C-element
chu133	10	3	2	11	3	2
full adder	3	1	1	4	-	2
half adder	4	1	1	4	1	2
master-read	11	5	7	15	6	7
mmu	8	5	4	8	5	4
vbe5b	9	4	2	10	4	2
Vbe5c	2	1	3	3	2	3
Vbe6a	16	4	6	16	4	6

VII. Conclusion

This paper focuses on SDI Optimization for STGs which is single cycle signal, and generating a Scalable Delay Insensitive circuit. Our method is modified the concurrent relation to order relation between primary input signal and non-primary input signal by guaranteeing K factor. The results of this study shows that STG might be decreased the arcs. Consequently, its implementation might be decreased or equal with area.

Regarding of the role of SDI optimization, this approach will be a great help in STG conflict solution. In the future work, we plan to extend SDI optimization to generate logic circuits from STGs with multi-cycle signals, and free-choice STGs.

References

[1] A. Takamura, M. Imai, T. Fuji, M. Ozawa, I. Fukasaku, Y. Ueno and T. Nanya. "TITAC-2: An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive Model". Proceeding of ICCD IEEE, (October 1997). pp. 288-294.

[2] T. Nanya, A. Takamura, M. Kuwako, M. Imai, M. Ozawa, M. Ozcan, R. Morizawa and H. Nakamura. "Scalable-Delay-Insensitive Design: A High-performance Approach to Dependable Asynchronous Systems" (Invited paper). Proceedings of International Symposium on Future of Intellectual Integrated Electronics, (March 1999) .pp. 531-540.

[3] Stevens K.S, Ginosar R., Rotem S, "Relative timing", IEEE Transactions on Very Large Scale Integration (VLSI) Systems , (2003) .pp.129-140

[4] S.B. Park, "Synthesis of Asynchronous VLSI circuits from Signal Transition Graph Specifications", Phd. Thesis, Tokyo Institute of Technology, (1996)

[5] S. Hauck. "Asynchronous design Methodologies: An overview". Proceeding of IEEE 83 , (October 1997) , pp. 66-93.

[6] Chu,T.A. " Synthesis of self-timed VLSI Circuits from Graph-Theoretic Specifications", Doctor Dissertation, Massachusetts Institute of Technology, (1987)

[7] M. Imai, M. Ozcan, T. Nanya. "Evaluation of Delay Variation in Asynchronous Circuits based on the Scalable-Delay-Insensitive Model" Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems (ASYNC'04) , (April 2004) .pp.62-71

[8] M. Sahni, T. Nanya. "On the CSC property of Signal Transition Graph Specification for Asynchronous Circuit Design" Proceeding of Asia and South Pacific (ASP-DAC '98) , (February 1998). pp.183-189

[9] K.Jen Lin, J.Wen Kuo, C.Shang Lin. "Direct Synthesis of Hazard-Free Asynchronous Circuits from STGs Based on Lock Relation and MG-Decomposition" Proceeding of European Design and Test Conference (EDAC) , (1994).pp.178-183

[10] P.Vanbekbergen, G. Goossens, F. Catthoor, Hugo J. De Man. "Optimized Synthesis of Asynchronous Control Circuits from Graph-Theoretic Specifications" Proceeding of IEEE Transaction on Computer-Aided Design , (November 1992).pp.1426-1438

[11] K.Jen Lin, C.Shang Lin "Realization Algorithm of Asynchronous Circuit from STG" Proceedings of The European Conference on Design Automation , (March 1992).pp.322 - 326

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)