数理解析研究所講究録
第522巻 1984年 266-279

入力制約を用いた論理回路の形式的検証について

京都大・工　木村　晋二　(Shinji Kimura)
安浦　寛人　(Hiroto Yasuura)
矢島　脩三　(Shuzo Yajima)

1．まえがき

集積回路技術の進歩に伴い、大規模な論理回路の設計および検証の研究の重要性が増してきた。大規模論理回路の設計に際しては、回路を幾つかの部分回路に分割して設計を進める構造化論理設計が一般的であるが、構造化論理設計においては、部分回路間のインタフェースで、信号の送受のタイミングを誤るなどの設計ミスを生じるので、各部分回路の論理的な正しさばかりでなく、部分回路間のタイミングの正しさを保つことにも注意を払わなければならない。タイミングの正しさを保証しながら設計を進めることは、論理回路の設計者にとって、回路の論理的正しさを保証すること以上に困難である。本報告では、タイミングの検証について考察する。

現在、VLSI等の設計において、論理回路のタイミングの誤りを検査する手法として用いられているものとしては、論理シミュレータを用いたタイミングシミュレーションによるもの（文献[1])と、遅延解析プログラムを用いて組合せ論理回路の最大／最小遅延を求める方法（文献[2])とが一般的である。しかし、これらの方法は、あくまで、設計者が設計

ミスを発見するのを支援するする手法であり、論理回路の検証を行う方法としては不十分である。このようなことから、形式的な検証手法の確立が望まれている。

本報告では、入力制約を用いたタイミング検証の手法について考察する。ここで述べる入力制約は、我々が以前開発した会話型論理設計検証支援システムISS（文献[3]）において導入された入力制約の一般化である。論理回路は一般に全ての入力に対して動作が定義されているのではなく、ある特定の入力に対してのみ動作が定義されている。入力制約は論理回路に加えても良い入力を表わしている。例えば、メモリ素子におけるセットアップタイム、ホールドタイムなどに基づく制約は入力制約の例である。論理回路が幾つかの部分回路から構成されている場合には、各々の部分回路の入力制約は常に満たされていなければならない。以下では入力制約の検証について考察する。

入力制約の検証において重要なのは次に示す二点である。

（1） 入力制約の記述法

（2） 検証のための形式的体系

入力制約の記述法としては、本報告においては形式的タイミングチャートを提案する。形式的タイミングチャートは、実際の論理設計において広く用いられているタイミングチャートを形式的に定義したものであり、直感的なわかりやすさを持ち、しかも厳密である。本報告では、更に、形式的タイミングチャートにより表わされた制約の意味を、各時刻に端子に現れる値の系列の集合により与え、制約が満たされてい

ることの検証を系列集合に対する演算を用いて行う。

入力制約の検証は、入力制約を満たす系列集合と実際に入力される系列の集合が与えられるなら、これら二つの集合の包含判定に帰着される。以下に示す方法は、系列集合を受理する有限オートマトンの性質を用いたものであり、部分回路の数をnとした場合、nの指数に比例した計算時間でこれらの入力制約の検証ができる。

2章では、準備として、系列集合の記述法及び系列集合を受理する有限オートマトン等の定義を行う。3章では、論理回路の入力制約及び入力制約の記述法について述べる。4章では、入力制約を用いた論理回路の検証手法について述べる。

2．準備

系列、正則表現、有限オートマトン等については参考文献[4]に譲る。アルファベットA上の系列 $x = ab\cdots c, y = de\cdots f$ $(a,b,\ldots c,d,e,\ldots,f \in A)$ に対し、xとyの選択的連接 $x \circ y$ を以下のように定義する。

(1) $x \circ y = ab\cdots ce\cdots f$; if $c = d$.

$\perp$; if $c \neq d$.

(2) $x \circ \varepsilon = \varepsilon \circ x = x$.

(3) $\perp \circ x = x \circ \perp = \perp$.

ここでεは長さ0の系列を表わし、$\perp$ は未定義の系列を表わす。アルファベットA上の系列集合BとCに対し、A上の系列集合 $\{x \circ y \mid x \in B$ かつ $y \in C\}$ をBとCの選択的連接と呼び、$B \circ C$ と表わす。非負整数i及び系列集合Xに対

し、 X°° i を X のi個の選択的連接 X。X。・・・。X と定義する。特に、 X°° 0 は {ε} とする。 又、二つの非負整数 i, j(i<j) に対し、 X°° i,j を X°° i U X°° i+1 U ・・U X°° jと定義する。X° を X°° 0,∞ により定義し、Xの選択的閉包と呼ぶ。

[補題 1]

B, C が正則集合である時、 B。C 及び B° は正則集合である。

[証明] B , C を受理する非決定性有限オートマトン (NDFA) を各々 (Q1, A, δ 1, q01, F1), (Q2, A, δ 1, q02,F2) とする。この時、 B。C 及び B° を受理する NDFA を以下のように構成することができる。 B。C を受理する NDFA を M 、 B° を受理する NDFA を M' とする。

M は (Q1 U Q2, A, δ , q01, F2)である。ここで、δ は以下のように定義される。

(1) δ (q, a) = δ 1(q, a) ; if q∈ Q1, a∈ A, δ 1(q,a)∩ F1 = {}.

(2) δ (q, a) = δ 1(q, a) U δ 2(q, a) ; if q∈ Q1,a∈ A, δ 1(q, a) ∩ F1 ≠ {}.

(3) δ (q, a) = δ 2(q, a) ; if q∈ Q2, a∈ A.

M' は (Q1 U Q2, A, δ ', q0, F1 U {q0}) である。ここで、δ 'は以下のように定義される。

(1) δ '(q0, a) = δ 1(q01, a).

(2) δ '(q, a) = δ 1(q, a) ;

if q∈ Q1, a∈ A, δ 1(q, a) ∩ F1 = {}.

(3) $\delta'(q, a) = \delta 1(q, a) \cup \delta 1(q0, a)$;

if $q \in Q1$, $a \in A$, $\delta 1(q, a) \cap F1 \neq \{\}$.

これらが正しい構成法を与えていることは選択的連接及び選択的閉包の定義より明らかである。 Q.E.D.

３． 入力制約を用いた検証

本章においては、入力制約の形式的定義及び入力制約を用いた論理回路の形式的検証について考察する。

３．１． 論理回路の設計検証

ここでは論理回路のモデルについて考察する。論理回路の外部とのインタフェースは、入力及び出力端子であるので、先ず、端子の定義を行う。

端子 T は、２つ組 (N, A) により定義される。Nは端子の名前を表わす。Aは、アルファベットであり、Tに現れる値の集合を表わす。T(t)はA の要素であり、時刻 t において端子 T の取る値を表わしている。t は非負の整数値のみを取ると仮定する。A上の系列 T(0)T(1)T(2)･･･をTの履歴と呼ぶ。

時刻 t において端子Tで変化が生じるとは、 $T(t) \neq T(t+1)$ を意味している。又、２つ組 (T(t), T(t+1)) により変化を表わす。

論理回路 M は、５つ組 (Ti, To, Ci, Co, Cio) により定義される。Tiは、入力端子の集合 {Ti1, Ti2, ..., Tip} であり、各 Tij を (Nij, Aj) とする。 To は、出力端子の集合 {To1, To2, ..., Toq} であり、各 Toj を (Noj, Bj) と

する。(A1× A2× ･･･× Ap)* の要素を入力系列と呼ぶ。又、(B1× B2× ･･･× Bq)* の要素を出力系列と呼ぶ。Ci は(A1× A2× ･･･× Ap)* の部分集合であり、入力制約と呼ばれる。Co は (B1× B2× ･･･× Bq)* の部分集合であり、出力制約と呼ばれる。Cio は (A1× A2× ･･･× Ap× B1× B2× ･･･× Bq)* の部分集合であり、入出力制約と呼ばれる。論理回路の動作は入力系列から出力系列への関数として与えられるので、Ciは動作の定義域、Co は動作の値域、 Cio は動作そのものを表わしている。例えば、メモリのアクセスタイムなどによるメモリを使用する上での条件は、入力制約の例である。又、フリップ・フロップなどにおける Q, Q' 出力の関係は出力制約の例である。これら三つの制約を総称してタイミング制約と呼ぶ。

論理回路 M は次の条件を満たす時正しいと呼ばれる。

(1) (A1× A2× ･･･× Ap)* の要素 x が Ci に含まれているならば、(B1× B2× ･･･× Bq)* のある要素 y(y の長さは x の長さに等しい)に対し、(x, y)が Cio に含まれている。(x,y) はデカルト積を表わす。

(2) (A1× A2× ･･･× Ap)* の要素 x及び(B1× B2× ･･･× Bq)* の要素 y に対し、 (x, y) が Cio に含まれており、かつ x が Ci に含まれているならば、 y は Co に含まれている。

上記の(1) は、入力制約を満たす入力に対しては必ず出力が定義されることを表している。又、(2) は出力と成り得るものは必ず Co に含まれていることを表わしている。

論理回路 M の設計は、３つ組 (M, V, E) により定義され

る。Mは論理回路を、Vは論理回路の集合を、Eは結線を表わしている。Vの要素をMの部分回路と呼ぶ。Vを{M1, M2, ..., Mn}とし、各Mjを(Tij, Toj, Cij, Coj, Cioj)とする。又、Mを(Ti0, To0, Ci0, Co0, Cio0)とする。Eは、{(T,T') | (T, T')は(Tc U Ti0 U To1 U To2 U ･･･U Ton)×(To0 U Ti1 U Ti2 U ･･･ U Tin)の要素であり、T, T'が(N, A), (N', A')であるとすると、AはA'に含まれる。}の部分集合であり、出力から入力への結線を表わしている。ここでTcは、アルファベットの要素数が1である端子の集合であり、固定入力を表わす。

論理回路Mの設計は、以下の条件が成立する時、正しいと呼ばれる。

(1) 論理回路Mの出力制約Co0が常に満されている。
(2) 論理回路Mの入出力制約Cio0が常に満されている。
(3) 各部分回路Mjの入力制約Cijが満されている。

論理回路の設計検証は、ここに示した条件が成立するかどうかの判定である。

３．２．形式的タイミングチャート

ここでは、入力制約等の記述法として、形式的タイミングチャートを提案する。まず形式的タイミングチャートの記述法について述べる。更に、形式的タイミングチャートが表わす入力制約を正則表現を用いて定義する。形式的タイミングチャートにより任意の入力制約が記述できるわけではないが、文献[5]のTTLデータブックに記載されているものにつ

いては全て記述でき、実用上は十分であると思われる。

形式的タイミングチャートは、1端子のタイミングチャート及び2端子のタイミングチャートから成る。1端子のタイミングチャートは、4つ組 (T, [0, n], L, α) により定義される。Tは端子 (N, A) である。[0, n] は整数集合 {0, 1, 2, ... ,n} を表わす。Lは、[0, n] から Aへの関数を表わす。αは、Lの変化点の対から P × (P U {∞})への関数である。ここで Lの変化点とは、{ i | i∈[0, n-1] かつ L(i)≠L(i+1)} の要素のことであり、Pは非負の整数の集合を表わす。又、∞は無限を表わし、Pのどの要素よりも大きい。図1に1端子のタイミングチャートの例を示す。図中、太い線はLを、"←→"と数字の対はαを表わしている。

2端子のタイミングチャートは、6つ組(T1, T2, [0, n], L1, L2, α) により定義される。T1 及び T2は端子(N1, A), (N2, B) を表わす。[0, n] は整数集合 {0, 1, 2, ... ,n} を表わす。L1, L2 は、[0, n] から A もしくは Bへの関数を表わす。αは、L1 もしくは L2 の変化点の対から P × P U {∞} への関数である。L1 及び L2 の変化点は1端子の場合と同様に定義される。図2に2端子のタイミングチャートの例を示す。

αは原始タイミング制約を表わす。原始タイミング制約は7つ組(T1, chg1, T2, chg2, min, max) により定義される。T1, T2 は端子 (N1, A), (N2, B)を表わす。chg1, chg2は各々 T1, T2 上での変化を表わす。すなわち、chg1 (chg2)は {(v, v') | v, v'∈A (B) かつ v≠v'} の要素である。min

はPの要素であり、変化 chg1 が生じてから、 chg2 が生じるまでに少なくとも minの間がなければならないことを表わす。 maxはPの要素もしくは∞であり、変化 chg1 が生じてから、 chg2 が生じるまでが高々 max であることを表わす。原始タイミング制約を満たす系列集合は以下のように定義される。但し、chg1, chg2 を (a, b), (c, d) とする。

(A× B·A× B - {a}× B·{b}× B)° 。

({a}× B·{b}× B。

((A× B·A× B-{a}× B·{b}× B-A× {c}·A× {d})° +

(A× B·A× B-{a}× B·{b}× B-A× {c}·A× {d})°° min-1,max-1。

(A× {c}·A× {d} + ε))°

ここで" × "は、デカルト積を表わす。又、 A·B は A と B の連接を表わす。

形式的タイミングチャートが表わすタイミング制約は、原始タイミング制約を全て満たす系列集合である。すなわち、全ての原始タイミング制約を満たす系列集合の共通集合である。

4. 形式的検証法

本章では、3章で定義した検証を行うための手法について述べる。

論理回路の設計検証は次の二段階にわかれる。

(1) 設計された回路の各端子に生じる系列の集合を求める。

(2) 系列集合が制約を満していることを示す。

論理回路の設計(M, V, E)において、各端子に生じる値の系列は、常に以下の制約を満している。

(1) Mの入力制約

(2) 各部分回路の出力制約

(3) 各部分回路の入出力制約

これら全ての制約を満たす系列集合が、第一段階で求めるべき集合である。各制約を満たす系列は正則表現で与えられるので、これらを受理する有限オートマトンの上で議論する。

[補題2]

原始タイミング制約を満たす系列集合を受理する決定性有限オートマトンが存在する。

[証明]

受理すべき系列集合 Sは

(A× B·A× B - {a}× B·{b}× B)° 。

({a}× B·{b}× B。

((A× B.A× B - {a}× B·{b}× B - A× {c}·A× {d})° +

(A× B·A× B-{a}× B·{b}× B-A× {c}·A× {d})°° min-1,max-1。

A× {c}·A× {d})°

である。 V + Wにおいて、V ∩ W = {} ならば+において非決定性は生じない。 {} は空集合を表わす。又、 V。Wにおいて、V ∩ V。W = {}ならば、選択的連接において非決定性は生じない。Sはこの条件を満しているので、Sを受理する決定性有限オートマトンが構成できる。状態数 3× max + 6の決定性有限オートマトンを構成する構成法が存在する。

(文献 [6])

Q.E.D.

V, Wが決定性有限オートマトンにより受理される系列の集

合であるとすると、 V ∩ W を受理する決定性有限オートマトンが構成できる。(文献 [4]) このとき V ∩ W を受理する決定性有限オートマトンの状態数は、 V,W を受理する決定性有限オートマトンの状態数の積である。

各制約の共通集合をとる時には、制約の定義されているアルファベットを揃える必要がある。例えば、原始タイミング制約(T1, c1, T2, c2, min1, max1),(T3, c3,T2, c4, min2, max2) に対しては、 Tj を (Nj, Aj)として (j = 1, 2, 3)、各々 A1× A2 及び A3× A2 上で定義されているが、これらの共通集合をとる場合には、A1× A2× A3 上で再定義しなければならない。受理オートマトンの場合、この操作を行っても、状態数は変化しない。

M の入力制約、各部分回路の出力制約、各部分回路の入出力制約全てを満たす系列集合を求める場合には、全端子のアルファベットのデカルト積上の系列集合を考えれば良い。この系列集合を受理する決定性有限オートマトンの状態数を評価する。 n を部分回路の数、Imaxを M もしくは部分回路の入力端子の数の最大、Omax を M もしくは部分回路の出力端子の数の最大、 T を原始タイミング制約 (T1, c1, T2, c2, min, max) の max の最大、Amax をアルファベットの要素数の最大とする。この時、原始タイミング制約の数は、変化対の数が高々 Amax·(Amax-1)であることと、端子対の数が高々 n·(Imax·Imax + Omax·Omax + Imax·Omax) であることから、高々 Amax·(Amax-1) · n·(Imax·Imax + Omax·Omax + Imax·Omax) である。各原始タイミング制約に対して、これを満た

す系列集合の受理オートマトンの状態数は、高々 3·T + 6である。よって、これら全部の共通集合を受理する決定性有限オートマトンの状態数は、高々(3·T + 6)**(Amax·(Amax-1)·n·(Imax·Imax + Omax·Omax +Imax·Omax))である。但し、**は指数乗を表わす。

次に制約が満されていることの判定について述べる。この判定は、二つの系列集合の包含判定に帰着される。すなわち第一段階で求めた系列集合 Sと検証すべき制約を満たす系列集合 S' とが与えられた時に、S が S' に含まれていれば、設計は正しい。S'が S を含むことの判定は、 S, S' がアルファベット A 上の系列集合であるとして、 S ∩ (A^x -S') が空であることの判定に等しい。但し"-"は差集合を表わす。S'を受理する決定性有限オートマトン M があるならば、(A^x - S')を受理する決定性有限オートマトンはMと同じ状態数で構成できる。よって、 S ∩ (A^x - S')を受理する決定性有限オートマトンの状態数は S, S'を受理する決定性有限オートマトンの状態数の積になる。決定性有限オートマトンの受理する系列集合が空かどうかを判定する、状態数に比例した計算量のアルゴリズムがあるので、各原始タイミング制約の検証に必要な計算量は、 高々 (3·T +6)**(Amax·(Amax-1)·n·(Imax·Imax + Omax·Omax + Imax·Omax))× (3·T + 6)である。よって、検証全体としては、 (3·T + 6)**(Amax·(Amax-1)·n·(Imax·Imax+ Omax·Omax +Imax·Omax)+1)× (Amax·(Amax-1)·n·(Imax·Imax+Omax·Omax + Imax·Omax))に比例した計算量でできる。

５．おわりに

本報告では、論理回路の正しさを入力制約等のタイミング制約を用いて定義し、正しさの検証を、系列集合を受理する決定性有限オートマトンを用いて検証する方法について考察を加えた。又、タイミング制約の記述法として、形式的タイミングチャートを提案した。ここで示した方法は、論理回路を構成する部分回路の数 nに対して、O(n･2**n) に比例した計算量を必要とする。

今後の問題点としては、検証に必要な計算量の削減が挙げられる。このためには、系列集合の記述法に関する研究が必要であると考えている。

謝辞

日頃、御討論頂く京都大学工学部上林弥彦助教授、平石祐実博士はじめ矢島研究室の皆様に感謝します。

参考文献

[1] McWilliams, T. M., Widdoes Jr, L. C., "SCALD: Structured Computer-Aided Logic Design," Proc. 15th Design Automation Conf., pp. 271-277, 1978.

[2] 小原，木村，" VLSI遅延解析プログラム，" 昭和５８年度電子通信学会情報・システム部門全国大会講演論文集［分冊２］，p. 552, 1983.

[3] Sakai, T., Tsuchida, Y, et al.,"An Interactive

Simulation System for Structured Logic Design-ISS," Proc. 19th Design Automation Conf., pp.747-755.

[4] ホップクロフト，ウルマン，"言語理論とオートマトン，" サイエンス社，1971.

[5] "The TTL Data Book for Design Engineers :Second Edition," Texas Instrument Incorporated,1976.

[6] Kimura, S., " Formal Timing Verification of Hardware with Timing Constraints," Master Thesis of Kyoto University, 1984.

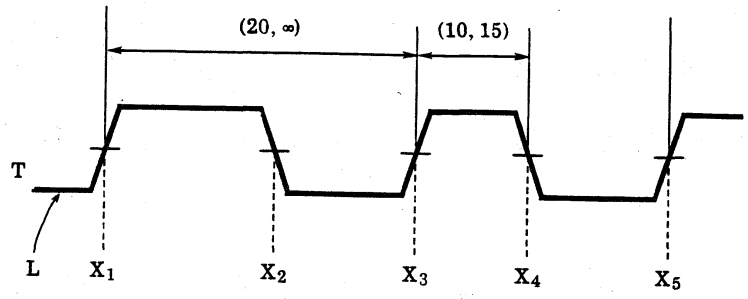

図１．１端子のタイミングチャート

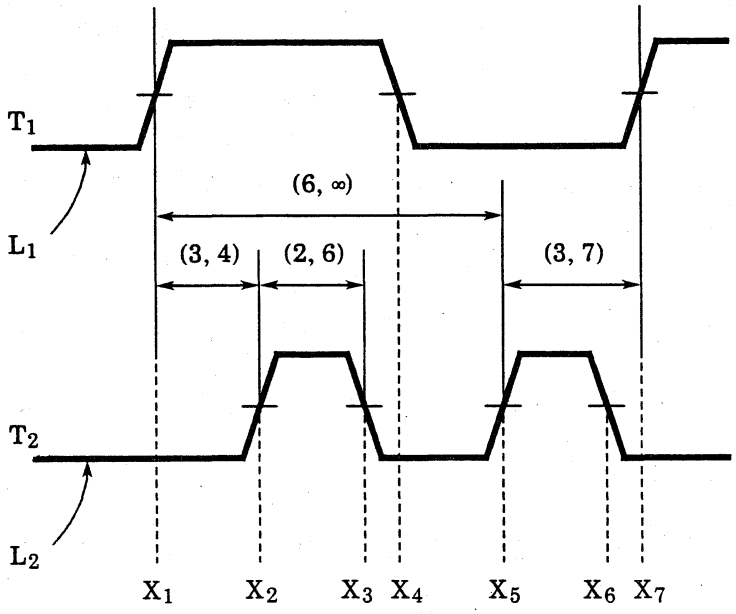

図２．２端子のタイミングチャート