

論理回路における閉路の切断について

神戸大学工学部 木村 晋二 (Shinji Kimura)

神戸大学工学部 羽根田 博正 (Hiromasa Haneda)

1. まえがき

論理回路動作のシミュレーションは、論理回路の設計検証などに広く用いられている^{(1),(2)}。論理回路の動作のシミュレーションは、通常離散時間の経過を表わす系列に対してなされるが、系列の集合に対して動作のシミュレーションを行う手法を考えることもできる⁽³⁾。このとき、シミュレーション対象の論理回路が閉路をもつ場合には、系列の集合に対してシミュレーションを行うことができない。なぜならば、閉路の部分に現れる系列集合は、シミュレーションを終了しないと求められないが、シミュレーション時にもその系列集合を必要とするからである。

ここでは、そのような閉路を切断して、入力に適当な系列集合を与えた場合の論理回路の動作について考察し、閉路を切断した出力側に現れる系列集合が、入力に加えたものと等しいときには、閉路を切断しない回路と同じ動作を行うという結果を導いている。このことから、閉路を持つ論理回路に対しても、組合せ論理回路と同じ位容易に、動作のシミュレーションを行うことができる。この場合、閉路に現れる系列集合を推測する必要があるが、設計検証などの応用では、フリップフロップなどのように、閉路上の値を仕様として与えることが多く、実際上は問題にならない。また、この結果は閉路以外の部分に関しても成立するので、大規模論理回路の動作のシミュレーションを、分割して行うこともできる。

本報告の2章では、準備として論理関数や論理回路の遅延時間について述べる。3章では、論理回路における閉路の切断定理およびその拡張について述べる。4章では、論理回路の閉路の切断定理の応用として、系列集合に対する論理回路のシミュレーションについて述べる。5章はむすびとする。

2. 準備

ここでは、論理関数、論理回路、論理回路の遅延時間などの定義を行う。

記号の有限集合をアルファベットと呼ぶ。アルファベット Σ 上の n 入力論理関数は、 Σ^n から Σ への関数とする。 Σ としては、通常 $\{0, 1\}$ や $\{0, 1, X\}$ などがとられる。

論理回路 M は、有向グラフ (V, E) 、 V から論理関数の集合への関数 F 、 V から自然数 N への関数 D からなる 3 項組 $((V, E), F, D)$ であると定義する。ここで V は節点の集合を、 E は $(V \times V)$ の部分集合で有向枝の集合を表わす。また、 F は各節点に対応する論理関数を、 D は各節点に対応する遅延時間を表わす。

V の要素の内、入射する枝のない節点を外部入力、射出する枝のない節点を外部出力と呼ぶ。 V の要素 v に対して、 $I(v)$ を $\{v' \in V \mid (v', v) \in E\}$ と定義し、 v の入力と呼ぶ。

V の要素の並び $v_1, v_2, \dots, v_n$ において、1 以上 $n-1$ 以下の任意の i で $(v_i, v_{i+1}) \in E$ であるとき、この並びを道と呼ぶ。道 $v_1, v_2, \dots, v_n$ の遅延時間を $D(v_1) + D(v_2) + \dots + D(v_n)$ と定義する。また、道 $v_1, v_2, \dots, v_n$ において、 $v_n = v_1$ であるとき、この道を(有向)閉路と呼ぶ。

論理回路 M の節点の集合 V と N およびアルファベット Σ に対し、関数 $Val: V \rightarrow (N \rightarrow \Sigma)$ で、以下の条件を満たすものを M における動作と呼ぶ。

(条件) $\forall v \in V$ および $\forall t \in N$ で、もしも $|I(v)| \neq 0$ ならば、 $D(v) = d$ としたとき、

$Val(v)(t) = F(v)(Val(I(v))(t), Val(I(v))(t-1), \dots, Val(I(v))(t-d))$ である。□

以下では、 $Val(v)(t)$ を単に $v(t)$ と表わす。

有向閉路のない論理回路を組合せ論理回路とよぶ。組合せ論理回路における道の遅延時間の最大値をその組合せ論理回路の最大遅延時間と呼ぶ。

3. 論理回路の閉路の切断定理

ここでは、論理回路における有向閉路の切断定理について述べる。

[定理 1]

図 1(a) に有向閉路を持つ論理回路 M を示す。ただし、図中の C は組合せ論理回路、

$I=(i_1, i_2, \dots, i_m)$ はMの外部入力並び、 $O=(o_1, o_2, \dots, o_n)$ はMの外部出力並び、 $Y=(y_1, y_2, \dots, y_k)$ はMの閉路の始点となる節点並びとする。また図1(b)には、Mのすべての閉路Yを切断し、外部入力 $Y_I=(y_{I1}, y_{I2}, \dots, y_{Ik})$ および外部出力 $Y_O=(y_{O1}, y_{O2}, \dots, y_{Ok})$ (y_{Oj} と y_{Ij} がMにおいて閉路を構成していたとする)とした組合せ論理回路M'を示す。

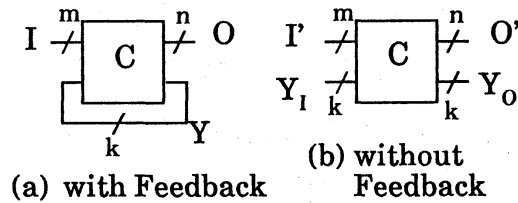


図1. 閉路の切断

このとき、以下の条件が満たされるならば、任意の時間 $t(\geq d_{\max}+1)$ において、 $(O, Y)(t)=(O', Y_O)(t)$ である。ただし、 $(O, Y)(t)$ および $(O', Y_O)(t)$ は、MおよびM'の動作で各回路のとり値である。組合せ論理回路Cの最大遅延時間を $d_{\max}$ と表わす。

(条件1) M'において Y_I から Y_O までの遅延時間は0よりも大きい。

(条件2) 任意の時間 t で、 $I'(t)=I(t)$ である。

(条件3) 0以上 $d_{\max}$ 以下の任意の時間 t で、 $Y_I(t)=Y(t)$ である。

(条件4) $d_{\max}+1$ 以上の任意の時間 t で、 $Y_I(t)=Y_O(t)$ である。 □

この定理は、論理回路における閉路を切断し、閉路の入力側に適当な入力を加えて出力が入力と等しくなったならば、そのときの動作がもとの閉路を含む論理回路と等しいことを述べたものである。

(証明)

ここでは、時間 $t(>d_{\max})$ より小さい任意の t' において、 $Y_I(t')=Y(t')$ であるか $Y(t')=Y_O(t')$ であるとき、時間 t においても $Y(t)=Y_O(t)$ であることを t に関する数学的帰納法により示す。組合せ論理回路Cの最大遅延時間が $d_{\max}$ であることおよび条件1より、M'の出力 $(O', Y_O)(t)$ は、論理関数を f として、

$$(O', Y_O)(t)=f(I'(t), I'(t-1), \dots, I'(t-d_{\max}), Y_I(t-1), Y_I(t-2), \dots, Y_I(t-d_{\max}))$$

と表わすことができる。つぎに条件2より、

$$(O', Y_0)(t) = f(I(t), I(t-1), \dots, I(t-d_{\max}), Y_1(t-1), Y_1(t-2), \dots, Y_1(t-d_{\max}))$$

である。条件4より、 $t-i \leq d_{\max}$ ならば $Y_1(t-i) = Y(t-i)$ である。また、条件3より $t-i > d_{\max}$ ならば $Y_1(t-i) = Y_0(t-i)$ でありかつ帰納法の仮定より $Y_0(t-i) = Y(t-i)$ である。よっていずれの場合でも $Y_1(t-i) = Y(t-i)$ が言えるので、

$$(O', Y_0)(t) = f(I(t), I(t-1), \dots, I(t-d_{\max}), Y(t-1), Y(t-2), \dots, Y(t-d_{\max}))$$

となり、 $(O', Y_0)(t) = (O, Y)(t)$ が言える。また、帰納の初期段階として、条件4により、 $t = d_{\max} + 1$ より小さい任意の時間で $Y_1 = Y$ であることを保証している。

ゆえに、 $d_{\max} + 1$ 以上の任意の時間で定理1が成立する。 $\square$

論理値 'X' に関して考察する。論理関数が論理値 'X' について以下のような条件を満たすならば、定理1における条件を緩めることができる。

(条件) アルファベット $\{0, 1, X\}$ 上の n -入力論理関数 f において、 $f(x_1, x_2, \dots, x_i = X, \dots, x_n) \neq X$ のとき、 $f(x_1, x_2, \dots, x_i = X, \dots, x_n) = f(x_1, x_2, \dots, x_i = 0, \dots, x_n) = f(x_1, x_2, \dots, x_i = 1, \dots, x_n)$ である。 $\square$

この条件は、以下に示すような通常の and, or, not、およびこれらの組み合わせにより構成される論理関数に対して成立する。

and(a, b) = if a = 0 then 0,
 else if b = 0 then 0,
 else if a = b = 1 then 1,
 else X.

or(a, b) = if a = 1 then 1,
 else if b = 1 then 1,
 else if a = b = 0 then 0,
 else X.

not(a) = if a = 0 then 1,
 else if a = 1 then 0,
 else X.

[系1]

定理1における条件3および条件4を以下のように変更する。このとき、任意の $j(1 \leq j \leq k)$ に対し、 $y_{Oj}(t) \neq X$ であるならば、 $y_{Oj}(t) = y_j(t)$ である。

(条件3') 0以上 $d_{\max}$ 以下の任意の時間 t で、任意の $j(1 \leq j \leq k)$ に対し、 $y_{Ij}(t) \neq X$ ならば $y_{Ij}(t) = y_j(t)$ である。

(条件4') $d_{\max} + 1$ 以上の任意の時間 t で、任意の $j(1 \leq j \leq k)$ に対し、 $y_{Ij}(t) \neq X$ ならば $y_{Ij}(t) = y_{Oj}(t)$ である。 □

この系より、閉路上に生じる値でわからないものは 'X' としておけば良いことが言える。これは、閉路上に生じる値の推測において有効である。

4. 閉路の切断の例

閉路の切断定理の応用として、系列集合を用いた論理回路のシミュレーションの例を示す。図2に、JK flip flopの仕様記述の例を示す。図2(b)の動作仕様および図2(c)に示すタイミングチャートより、JK flip flopのClear動作に関して図3に示す

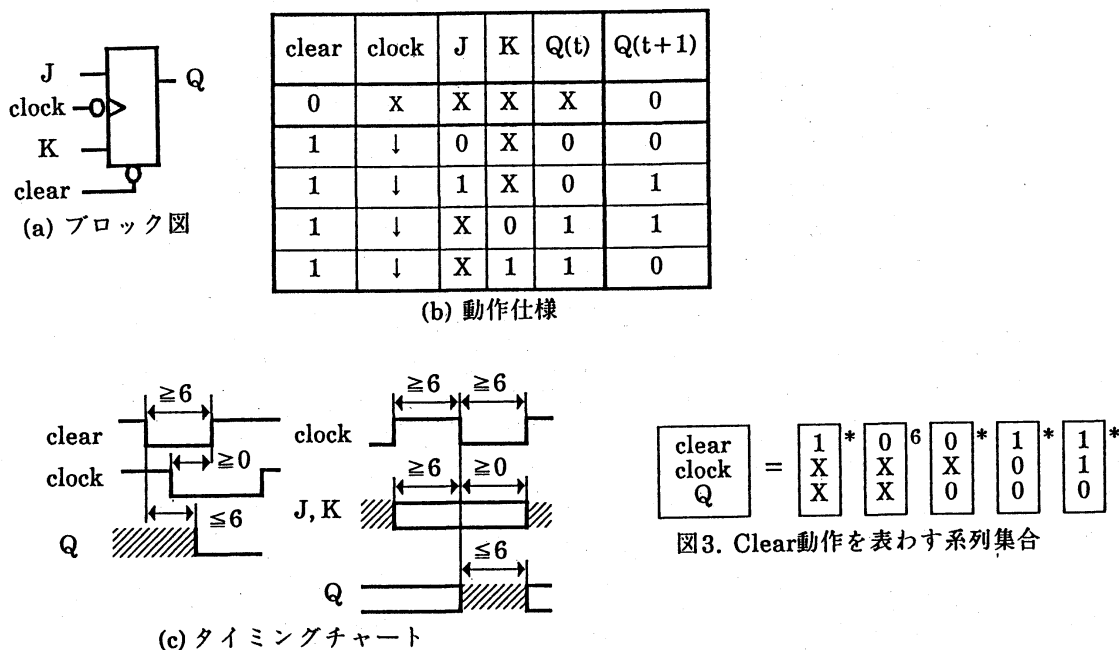


図2. JK flip flopの動作仕様

ような系列集合が得られる。この系列集合に対して、図4に示す実現をシミュレーションしてゆく。ただし、図4の回路に先の定理を適用し、図5に示す閉路を切断した回路で考える。

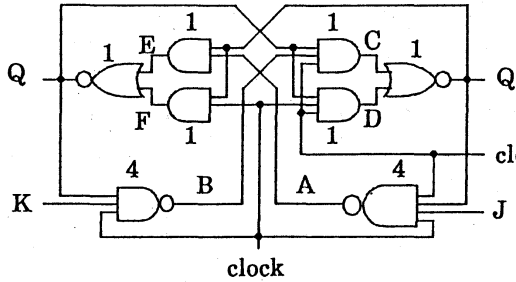


図4. JK F.F.の実現

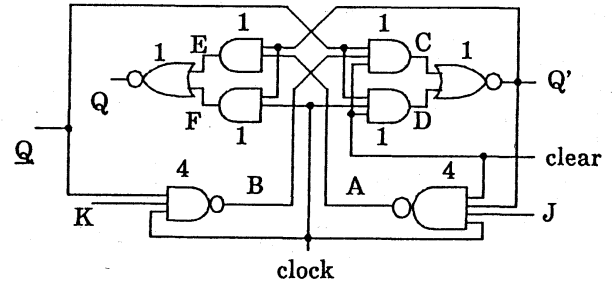


図5. JK F.F.の検証用の回路

図5は組合せ論理回路となっているので、各ゲートの出力に現れる系列集合を順次求めてゆけば良い。これを図6に示す。例えば、図6の $(A, \text{clear}, \text{clock}, Q)^T$ は、clearが0になってから4単位時間遅れてAが1になる系列集合を表わしている。図

$$\begin{aligned}
 \begin{matrix} A \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 4 & 1 & 2 & 1 & * & 1 & * & 1 & 4 & X & * \\ 1 & & 0 & & 0 & & 0 & & 1 & & 1 & & 1 & \\ X & & X & & X & & X & & 0 & & 1 & & 1 & \\ X & & X & & X & & 0 & & 0 & & 0 & & 0 & \end{bmatrix} \\
 \begin{matrix} C, D \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 0 & 0 & 5 & 0 & * & 0 & 0 & * & 0 & 0 & * \\ 1 & & X & & X & & 0 & & 0 & & 1 & & 0 & \\ X & & X & & X & & X & & 0 & & 1 & & 1 & \\ X & & X & & X & & 0 & & 0 & & 0 & & 0 & \end{bmatrix} + \begin{bmatrix} 0 & 0 & * & 0 & * & 0 & 0 & 1 & 1 & * & 0 & 0 & * \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 \begin{matrix} Q' \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 2 & 1 & 4 & 1 & * & 1 & 1 & * & 1 & 1 & * \\ 1 & & X & & 0 & & 0 & & 1 & & 1 & & 1 & \\ X & & X & & X & & X & & 0 & & 0 & & 0 & \\ X & & X & & X & & 0 & & X & & 0 & & 0 & \end{bmatrix} + \begin{bmatrix} 1 & 1 & * & 1 & * & 1 & 1 & 1 & 1 & * & 1 & 1 & * \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & * & 1 & 1 & * & 1 & 1 & 1 & * & 1 & 1 & * \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 \begin{matrix} E \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 5 & 1 & 1 & * & 1 & 1 & * & 1 & 5 & X & * \\ 1 & & X & & 0 & & X & & 0 & & 1 & & 1 & \\ X & & X & & X & & X & & X & & 0 & & 1 & \\ X & & X & & X & & 0 & & X & & 0 & & 0 & \end{bmatrix} + \begin{bmatrix} 1 & 1 & * & 1 & 1 & * & 1 & 1 & 1 & * & 1 & 1 & * \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & * & 1 & 1 & * & 1 & 1 & 1 & * & 1 & 1 & * \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 \begin{matrix} F \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 3 & X & 3 & X & * & X & 0 & 0 & 1 & * & X & * \\ 1 & & X & & 0 & & X & & X & & X & & 1 & \\ X & & X & & X & & X & & X & & 0 & & 1 & \\ X & & X & & X & & 0 & & X & & 0 & & 0 & \end{bmatrix} + \begin{bmatrix} X & 0 & * & 0 & 1 & * & 0 & 1 & 1 & * & X & 1 & * & X & * \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} + \begin{bmatrix} X & 1 & * & X & 1 & * & X & 1 & 1 & * & X & 1 & * & X & * \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} \\
 \begin{matrix} Q \\ \text{clear} \\ \text{clock} \\ Q \end{matrix} &= \begin{bmatrix} X & * & X & 6 & 0 & * & 0 & 0 & 0 & * & 0 & 5 & 0 & * & 0 & 5 & 0 & * \\ 1 & & X & & 0 & & 0 & & X & & 0 & & 0 & & 1 & & 1 & \\ X & & X & & X & & X & & 0 & & 0 & & 0 & & 1 & & 1 & \\ X & & X & & X & & 0 & & X & & 0 & & 0 & & 0 & & 0 & \end{bmatrix} + \begin{bmatrix} 0 & 0 & * & 0 & 1 & * & 0 & 1 & 1 & * & 0 & 1 & 1 & * & 0 & 1 & 1 & * \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & * & 0 & 1 & * & 0 & 1 & 1 & * & 0 & 1 & 1 & * & 0 & 1 & 1 & * \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

図6. JK F.F.のclear動作の検証

6で各素子の出力を求めるときには、(clear, clock, Q)の部分でタイミングを合わせている。図6の(Q , clear, clock, Q)により、任意の時間で出力の Q と、入力として与えた Q が等しいことがわかる。よって、ここで与えた系列集合は、もとの回路(図4)で生じるものであり、タイミング仕様が満たされていることがわかる。

5. むすび

本報告では、論理回路における閉路の切断に関する定理について述べた。また、この定理の応用として、閉路を含む論理回路の系列集合に関するシミュレーションについて述べた。論理回路の系列集合に関するシミュレーションは設計検証の手段として有効であり、ここで述べた定理の応用範囲も広い。また、この定理を拡張すれば、論理回路を任意の位置で分割することが可能となり、大規模な論理回路の検証に有効であると考えられる。

謝辞

日頃から御討論戴く京都大学矢島脩三教授および安浦寛人博士に心から感謝いたします。なお、本研究は一部文部省科学研究費補助金による。

参考文献

- (1) M. A. Breuer and A. D. Friedman : Diagnosis & Reliable Design of Digital Systems, Computer Science Press, 1976.
- (2) 樹下, 浅田, 唐津 : VLSIの設計II, 岩波書店, 1985.
- (3) 木村, 矢島 : “論理回路の入力制約および入出力仕様の記述とその検証,” 信学論(D), Vol. J69-D, No. 4, pp. 502-513 (1986-04).