

時間論理分割を用いた順序回路の合成方式

4N-3

戸次 圭介, 浜田 巨曼, 清水 嗣雄†, 平塚 憲晴†, 島 和正††

(株)日立製作所 日立研究所, †中央研究所, ††システム開発研究所

1. はじめに

状態遷移図で表現された動作を実現する論理回路の自動合成アルゴリズムの研究は古くから行なわれており、合成される論理回路が最小となるように状態を符号化する手法を確立することが最も重要な技術課題となっている。状態を適切な2進数の数値に符号化するには膨大な設計空間を探索しなければならないが、状態遷移を複数の部分状態遷移に分割して合成することによりこの膨大な探索を回避することができる。そのため状態遷移の分割手法を確立してゆくことは極めて重要な課題である。^{1),2)} マイクロプロセッサ等の論理回路を解析すると遅延素子(Dフリップフロップ)を用いて順序回路を簡単に実現していることが多い。本報告では与えられた状態遷移図を遅延素子を用いて構成する部分と、従来の手法で構成する部分に分割する方法について述べる。

2. 時間論理分割手法

図1は、順序回路の動作を5状態の状態遷移図によって表現したものである。ここでは状態遷移図の取り扱いを簡単にするため、ブール数を要素とするブーリアン行列を導入することにより、状態遷移図を2つの行列方程式で表現する。図1の状態遷移図は次の2つの行列方程式で記述することができる。

$$\begin{pmatrix} s1(n) \\ s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} = z^{-1} \begin{pmatrix} \alpha & \alpha & 0 & 0 & 0 \\ 0 & 0 & \alpha & \alpha & \alpha+\beta \\ 0 & 0 & \beta & \beta & 0 \\ \beta+\chi & \beta+\chi & 0 & 0 & 0 \\ 0 & 0 & \chi & \chi & \chi \end{pmatrix} \begin{pmatrix} s1(n) \\ s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} \quad \text{---(1)}$$

$$\begin{pmatrix} o1(n) \\ o2(n) \\ o3(n) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s1(n) \\ s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} \quad \text{---(2)}$$

$$\begin{aligned} \alpha &= \delta(a=0 \ \& \ b=0), \\ \beta &= \delta(a=0 \ \& \ b=1), \\ \chi &= \delta(a=1) \end{aligned}$$

ここでs1(n)~s5(n)は時刻nのときの状態を表現するブール変数であり、値として1をもつ変数に対応する状態が発火してい

ることを意味する。δはクロネッカのデルタ関数であり、例えばδ(a=1)は信号aの値が0のとき1となり、信号aの値が1のとき0となるブーリアン関数である。z⁻¹は時間を1つシフトする演算子であり、z⁻¹s1(n) = s1(n-1)が成立する。(1)式は現状態と次状態の関係を表現する式であり状態方程式と呼ぶことにし、また(2)式は現状態と出力との関係を表現する式であり出力方程式と呼ぶことにする。以下ではこの2つの行列方程式を等価変換することによって、状態遷移情報を分割する手法について述べる。

定理1 出力方程式の等価変換

m状態の状態方程式及び1次元の出力方程式を次式に示す。

$$s(n) = z^{-1} A s(n) \quad \text{---(3)}$$

$$o(n) = B s(n) \quad \text{---(4)}$$

ただしs(n)=[s1(n) ... sm(n)], o(n)=[o1(n) o2(n) o1(n)]とする。

ここでm x mのブーリアン行列A及び1 x mのブーリアン行列Bを下式のように表現しよう。

$$A = \begin{pmatrix} a1 \\ a2 \\ \vdots \\ am \end{pmatrix}$$

$$B = \begin{pmatrix} b11 & b12 & \dots & b1m \\ b21 & b22 & \dots & b2m \\ \vdots & \vdots & \dots & \vdots \\ b11 & b12 & \dots & b1m \end{pmatrix}$$

このときi<j, 1 ≤ k ≤ m に対してaki = akj が成立するとき、出力方程式は次式のように等価変換することができる。

$$\begin{aligned} & b_{ki} = b_{kj} \text{のとき} \\ o_k(n) &= [b_{k1} \ b_{k2} \ \dots \ b_{km}] s(n) \\ & b_{ki} \neq b_{kj} \text{のとき} \\ o_k(n) &= \{ [b_{k1} \dots b_{ki-1} \ 0 \ b_{ki+1} \dots b_{kj-1} \ 0 \ b_{kj+1} \dots b_{km}] \\ & \quad + b_{ki} z^{-1} a_i + b_{kj} z^{-1} a_j \} s(n) \end{aligned}$$

この出力方程式の等価変換によりsiの状態で出力する信号とsjの状態で出力する信号を等しくすることができる。なお証明は省略する。

この定理を用いて(1)式及び(2)式で表現される状態遷

Temporal Logic Decomposition of Finite State Machine

Keisuke Bekki, Nobuhiro Hamada, Tsuguo Shimizu,

Noriharu Hiratsuka, Kazumasa Shima Hitachi Ltd.

移表現を変換すると次式を得る。

$$\begin{pmatrix} o1(n) \\ o2(n) \\ o3(n) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ z^{-1}\alpha & z^{-1}\beta & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s1(n) \\ s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} \quad \text{----(5)}$$

補題 1 識別不可能の十分条件

(3) 式の状態方程式及び (4) 式の出力方程式で表現される状態遷移表現があるとする。ここで **A** 及び **B** を次のように行列に分解して表現する。

$$A = [a1 \ a2 \ \dots \ am], \quad B = [b1 \ b2 \ \dots \ bm]$$

このとき $1 \leq i < j \leq m$ に対して $a_i = a_j, b_i = b_j$ が成立するとき状態 s_i と状態 s_j は識別不可能な状態対である。この補題が成立することは Paul-Unger の方式より明らかである。

補題 1 を用いると (1) 式と (5) 式で表現される状態遷移表現は以下に示すように 1 つ状態数の少ない状態遷移表現を得ることができる。

$$\begin{pmatrix} s1(n)+s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} = z^{-1} \begin{pmatrix} \alpha & \alpha & \alpha & \alpha+\beta \\ 0 & \beta & \beta & 0 \\ \beta+\chi & 0 & 0 & 0 \\ 0 & \chi & \chi & \chi \end{pmatrix} \begin{pmatrix} s1(n)+s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} \quad \text{----(6)}$$

$$\begin{pmatrix} o1(n) \\ o2(n) \\ o3(n) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ z^{-1}\alpha & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s1(n)+s2(n) \\ s3(n) \\ s4(n) \\ s5(n) \end{pmatrix} \quad \text{----(7)}$$

ここに示したように定理 1 及び補題 1 を繰り返し適用することにより少ない状態数で状態遷移を記述することができるようになる。例えば (6) 式及び (7) 式に対しさらにこの処理を適用してゆくと、最終的に以下の表現が導出できる。

$$\begin{pmatrix} s1(n)+s2(n) \\ s3(n)+s4(n) \\ s5(n) \end{pmatrix} = z^{-1} \begin{pmatrix} \alpha & \alpha & \alpha+\chi \\ \beta+\chi & \beta & 0 \\ 0 & \chi & \chi \end{pmatrix} \begin{pmatrix} s1(n)+s2(n) \\ s3(n)+s4(n) \\ s5(n) \end{pmatrix} \quad \text{----(8)}$$

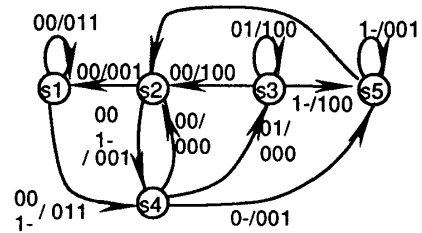
$$\begin{pmatrix} o1(n) \\ o2(n) \\ o3(n) \end{pmatrix} = \begin{pmatrix} 0 & z^{-1}\beta & 0 \\ z^{-1}\alpha & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} s1(n)+s2(n) \\ s3(n)+s4(n) \\ s5(n) \end{pmatrix} \quad \text{----(9)}$$

(1) 式および (2) 式の状態遷移を表現するためには 5 つの状態が必要であったが、定理 1 と補題 1 の処理を繰り返し適用することにより 3 つの状態に表現できるようになることがわかる。この省略できた 2 つの状態は (9) 式の出力方程式に現われた 2 つの時間シフト演算子に等価変換されたのである。なおこの時間シフト演算子は遅延素子を用いて簡単に実現できる。従って (8) 式の状態方程式を実現する回路は従来の手法で実

現する部分であり、(9) 式の出力方程式を実現する部分は遅延素子を用いて実現するのに適した部分となる。図 2 は (8) 式及び (9) 式から生成した論理回路を示すものである。ここで示した 2 つの素子は遅延素子であり、(9) 式に現われた 2 つの時間シフト演算子をそれぞれ実現する。

3. おわりに

本報告では状態遷移表現を遅延素子を用いて合成する部分と状態割付を行なって合成する部分に分割する手法について述べた。また 10 状態で表現できる状態遷移表現を用いて本手法の評価を行なった結果、本手法を適用した状態遷移表現を分割して生成した論理回路は、従来の手法で生成したものと比較してハードウェア量が小さくなる傾向があることを確認した。しかしながら遅延素子を用いて合成する部分の最適性やその適用範囲等、さらに検討を進めてゆく必要がある。



入力 a, b
出力 o1, o2, o3
分岐条件及び出力は ab/o1 o2 o3 の順で記述

図 1 順序回路の動作を記述する状態遷移図の例

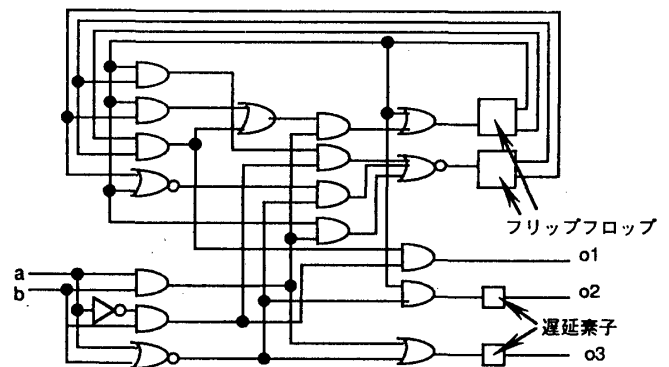


図 2 論理分割の結果から生成した順序回路の例

参考文献

- 1) S. Devadas: Decomposition and Factorization of Sequential Finite State Machines: IEEE, Trans on CAD, Vol. 8, No.11, 1989
- 2) G. De Micheli, et al: Optimal State Assignment for Finite State Machines: IEEE, Trans. on CAD, Vol. 4, No. 3, 1985