

## 代数的手法を用いた回路設計支援システムにおける 状態図簡約機能とその評価

3 E-2

榎原孝一 北道淳司 東野輝夫 谷口健一

大阪大学 基礎工学部

## 1. まえがき

我々は、代数的手法を用いた順序回路の仕様記述法及び段階的設計法を提案している<sup>[1]-[4]</sup>。この設計法の特徴は、回路の要求仕様を代数的言語ASLを用いて記述し、それからのトップダウン設計(用いる動作アルゴリズム、バス構成、部品等)を形式的に記述できること、設計検証を行えること等が挙げられる。

現在、我々は提案する設計法を支援するシステムを作成している。支援系は、下位レベルの制御部、入力論理式及びマイクロプログラムの合成機能、冗長な状態遷移の除去やアーキテクチャに応じた状態図の変形等の状態図に対する操作を行う状態図簡約機能、設計検証を行うための検証支援機能等からなる。今までに行った回路の設計例から状態図に対する操作を抽出した<sup>[4]</sup>。本論文では、それらをもとに状態図簡約機能をプログラム化し、機能の有用性等について調べた結果について述べる。

## 2. 代数的手法を用いた順序回路設計法と回路設計支援システムの概要

同期式順序回路の仕様は、動作内容の定義Dと動作の実行制御の記述Cからなる(図1)。各部品の値を状態成分

関数で表し、回路の動作を状態遷移関数で表す。回路の動作内容の定義Dは、各状態遷移を行うと状態成分の値がどの様に変化するかを公理の形で記述する(図1では公理を略記している。公理及び文法の書き方などの詳細は文献[1]参照)。動作内容は、配列の中から最大値を取り出すような抽象的な動作や、抽象データタイプの値を保持する部品間のデータ転送等を定義することもできる。動作の実行制御Cでは、有限制御部の値(以下、状態名と呼ぶ)がどのような値で、どのような分岐条件が成り立つときにどの状態遷移を行うか(状態遷移の実行条件)、またその状態遷移を行うと状態名はどのような値になるかを記述する。これらは、状態名に着目すると状態図で表される。但し、状態図を代数的言語ASLで記述するため、2つの状態成分関数CONTROL, VALIDを導入し、公理の形で記述する。

上位レベルの仕様 $t=<D,C>$ が与えられた時、アーキテクチャを考慮しない抽象度の高いレベルでは、設計者は、

(a1)下位レベルで用いる各動作内容D'を記述する。

(a2)上位レベルの状態遷移と下位レベルの状態遷移系列との対応関係M(いわゆるスケジューリング)を記述する。対応関係に用いる分岐条件にはアーキテクチャを考慮する

ことなく自由な論理式を与えることができる。

(a3)Mが正しいか(Mで指定された順にD'を実行すればDの要求を満たすか)を検証支援機能(既に開発されているASL検証支援系を利用する)を用いて形式的に証明する。証明に失敗した時は、D'やMの修正を行い、成功した時は、

(a4)状態図CとMから、下位レベルの状態図C'を合成し、下位レベルの仕様 $t'=<D',C'>$ を得る(合成機能)。  
(a5)必要ならば、C'を簡単化する。得られた状態図をC''とし、D'に新たな状態遷移の追加・削除を行って得られたものをD''とする。このレベルの回路を、 $t'=<D'',C''>$ とする(状態図簡約機能)。

この手順を繰り返すことにより、各動作内容が導入するアーキテクチャにおいて数クロック(または1マイクロ命令)で実現できるレベルまで回路を具体化する。次に、以下の手順で部品の機能やバス構造を含む接続関係、入力論理式やマイクロプログラムからなる具体的な回路を求める。

(b1)下位レベルで用いる(制御部を除く)部品の機能及びバス構造を含む入出力関係を記述し(これらにより1クロックや1マイクロ命令で行えるデータ転送が定義できる)、これらの記述をD'とする。

(b2)上位レベルの動作内容とそれを実現する各部品への制御入力系列(または1マイクロ命令)との対応

関係Mを記述する。D'を限定すれば正しい対応関係Mを機械的に求めることもできる。

(b3)設計者がMを与えた場合、Mが正しいかどうかを検証支援機能を用いて証明する。証明に失敗した時は、D'やMの修正を行い、成功した時は、

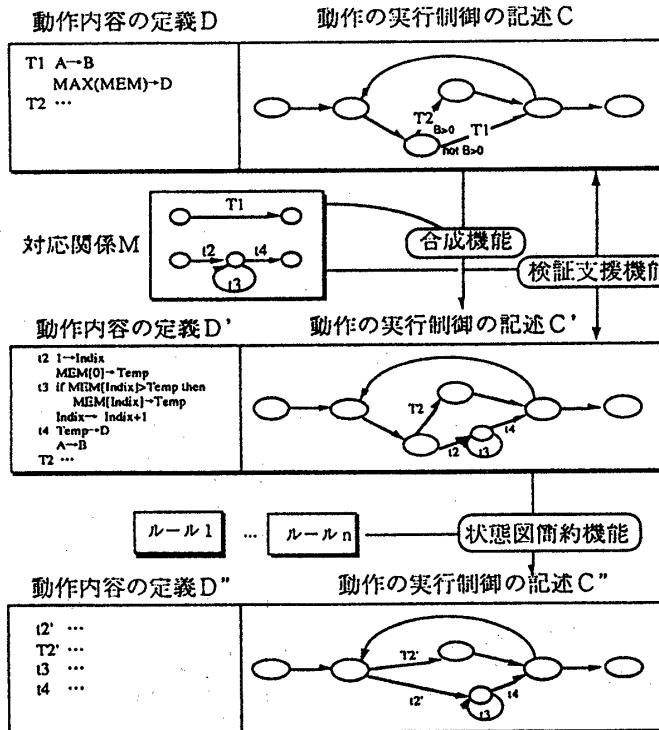


図1 回路の仕様、設計手順及び設計支援システムの概略

State Reduction Facility in Hardware Design Support System based on Algebraic Method

Kouichi Narahara Junji Kitamichi Teruo Higashino Kenichi Taniguchi  
Department of Information and Computer Sciences,  
Faculty of Engineering Science, Osaka University  
Toyonaka-shi, 560 Japan

(b4) 状態図CとMから、下位レベルの状態図C'を合成し、C'の各分岐部における条件判定が、与えたD'で実現できるように分岐部の展開を行い、C''を得る。マイクロプログラムを求める場合には、1つの状態から出射している各状態遷移の分岐条件に従った実行内容を、1マイクロ命令の実行に対応させられるように、C'を変更する。必要ならば、C''を簡略化する（状態図簡約機能）。

(b5) C''及びMから、制御レジスタと各部品への入力論理式やマイクロプログラムを合成する（図1では省略している）。

以上の手順により、要求仕様を満たす回路を得る。

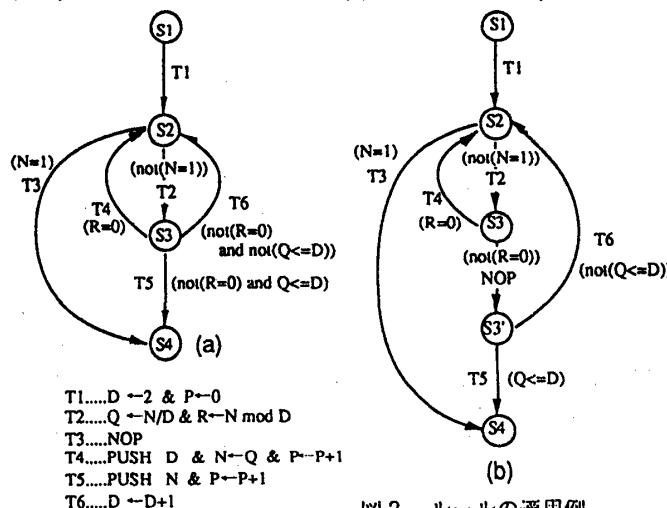
### 3. 状態図簡約機能とその評価

状態図簡約機能は、2. で述べた設計法において、手順(a5)及び(b5)の支援を行うものである。提案する設計法を幾つかの事例に適用し、それをもとに状態図に対する操作等を10個程度のルールとして抽出した<sup>[4]</sup>。状態図簡約機能を用いて、状態図におけるルールの適用箇所の指定、指定箇所における適用可能ルールの検出、指定したルールの適用条件の判定及び適用実行等が行える。

手順(b4)におけるルールの適用を簡単な例（与えられた整数を素因数分解し、全素因数をメモリ内に格納する回路）で説明する。図2(a)が、ある程度詳細化されて得られた状態図である。この回路を、マイクロプログラム制御回路で実現する。但し、プログラムの分岐はレジスタAの内容(A=0等)による条件付ジャンプによってのみ実現されるようなアーキテクチャとし、各状態遷移で行われるデータ転送は1マイクロ命令で実行できるものとする。

各分岐条件がand, orもしくは、Q=D=0等の形の論理式からなる（または変形できる）ことを確認し、2本以上の枝が出射する状態S3に分岐及び分岐条件を展開するルールを適用する（図2(a)から(b)）。

各分岐条件が判定できるように、レジスタAに必要なデータを格納させる。例えばS3においてAの内容がそれ以降の状態遷移で参照されていなければ、Rの内容をAに転送する状態遷移を挿入するルールを適用する（図2(b)から(c)）。Aの内容がそれ以降の状態遷移で参照される時は、Aの内容を退避できるレジスタやメモリを探し、それに一時的に退避し分岐後にその内容を再びAに転送するルール群を適用する。退避できるレジスタやメモリ等が無い場合は、そのアーキテクチャでは実現できないので、レジ



スカを増やす等の再設計を行う（手順(b1)に戻る）。

手順(b4)において、この例では、1つの状態から出射している2つの状態遷移の分岐条件に従った実行を、1マイクロ命令の実行に対応させられるように、状態図を変更する。例えば図2(c)の状態S5において、分岐条件A=0に従ってT4かNOPを実行することを1マイクロ命令で実現できないときは、「S5とT4間にNOPを挿入する」ルールを適用する。

これ以後、上述のルールの適用により挿入されたNOP等の冗長な遷移を取り除くルール群を繰り返し適用する。上述の例題及びソートIC<sup>[2,3]</sup>等の例題に対し、ルールの適用回数及び時間(CPUはSun SparcIPC)等を表1に示す。

	変換前の状態数	変換後の状態数	適用ルール数	適用時間(秒)
素因数分解IC	4	12	13	10.7
ソートIC レベル6 アーキテクチャの導入版	53	38	25	29.4
ソートIC レベル7 アーキテクチャの導入版	38	49	48	173.3

表1 簡約機能の入力として用いた回路の規模と適用結果

ソートICのレベル6の回路は、アーキテクチャを考慮しないレベルのものなので、冗長な状態遷移を削除するルールのみ適用した。ソートICのレベル7及び素因数分解の例では、アーキテクチャを決定しマイクロプログラムを導出できるようするため、分岐部の展開等を行うことにより状態数は増加している。ソートICは、人手により書き下したアーキテクチャとマイクロプログラム(サイズ66)を参考に、それと同一のアーキテクチャを採用し、また、そのマイクロプログラムを参考にしたのでそれ程試行錯誤なくルールの適用が行え、サイズ49のマイクロプログラムが得られた。簡略化の方針としてプログラムサイズを小さくすること、実行効率を良くすることなどが挙げられるが、このデータは、実行効率は若干悪いがプログラムサイズを小さくする方針によるものである。

### 4. あとがき

実際にシステムを運用した結果、システムの可視化やルール適用戦略（どのような条件が成り立つときにどのような順でルールを組み合わせて適用するか）に従って、ルールを自動的に適用する機能が有効であると思われ、実現法などの検討を行っている。今後、検証機能等の機能も拡張し、統合的な設計支援系を構築し、その評価を行おう予定である。

### 文献

- [1] 杉山裕二、北道淳司、谷口健一：「代数的手法を用いた順序回路の記述とその詳細化について」、電子情報通信学会技術研究報告 COMP88-7, pp.61-70(1988-05).
- [2] 北道淳司、杉山裕二、谷口健一：「クイックソートICの代数的手法と制御回路の自動生成について」、情報処理学会研究報告 DT46-13, pp.95-102(1989-02).
- [3] 北道淳司、谷口健一：「代数的手法を用いたマイクロプログラム制御方式順序回路の階層的設計」、情報処理学会第42回(平成3年前期)全国大会講演論文集, pp.6-170-171(1991-03).
- [4] 横原孝一、北道淳司、東野輝夫、谷口健一：「代数的手法を用いたマイクロプログラム制御方式順序回路の設計例」、情報処理学会研究報告 DA59-14, pp.99-106(1991-10).