

2 J-11

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

北道 淳司

檜原 孝一

谷口 健一

大阪大学 基礎工学部 情報工学科

1. まえがき

ハードウェアの仕様記述言語では、その意味定義が明確であること、抽象度の高い機能設計レベルから具体的な論理設計レベルまでの任意の抽象レベルで書きたいことだけを自由に記述できること等は重要な特徴である。筆者らの研究グループでは、代数的言語ASLによる同期式順序回路の記述スタイルを定め、詳細化手法を提案し^[1, 2]、その評価を行ってきた。

従来、マイクロプログラム制御方式順序回路の正しさ(与えられた部品とそれらの入出力関係とマイクロプログラムのもとで、回路が動作するとき、要求通りに動作するかどうか)などに関する形式的な議論は行われていない。本論文では、我々が提案してきた手法のもとに、制御部にマイクロプログラム方式を適用する場合、どのような手順で設計を行うか、どのような検証を行えば正しい設計が行えるか、どのような部分が自動合成可能か等について、ソート用回路^[3]を例にして検討を行っている。これらについて報告する。

2. 順序回路の記述スタイル及び詳細化手法

順序回路の記述は、ASLで抽象的順序機械型のスタイルを用いて行う。回路の状態を表すデータタイプを導入し、回路の動作を状態遷移関数で、部品の値を状態成分関数で表す。また、状態遷移の実行順指定するために、遷移条件指定関数VALIDを導入し、状態sで遷移Tを実行すべきときVALID(T(s))が真となるように書く。

我々は、例題として、文献[3]におけるソートICの設計を行ってきた^[2](但、文献[3]には、制御方式にワイヤードロジック方式を用いたものと、マイクロプログラム方式を用いた2つの例があり、またバス構成や用いる部品も異なる。ここでは、前者をソート回路H、後者をソート回路Mと呼ぶ)。例えばソートICにおける要求仕様に相当するレベル(レベル1)では、以下のことを記述する。ソートを行う状態遷移SORTとソートされる要素を格納する状態成分E等を導入する。状態遷移SORT後の配列Eの各要素が昇順に並んでいなければいけない。さらに、SORTの実行前後で、配列の要素が欠落混入してもいけない。また、SORTが初期状態より一度だけ実行されるように、制御部の値を表す状態成分CONTROLを導入

し、VALID関数を記述する。

代数的手法による設計は、一段階の詳細化を行う次の手順(以下手順A)を繰り返して行う。

(1)下位レベルで導入する状態成分、状態遷移を決定し、それらの公理(状態遷移Tを行えば状態成分Fがどのような値となるか、または、どのような性質を満たせばよいか)を記述する。

(2)上位レベルの状態遷移と下位レベルの状態遷移(一般に系列)との対応を考案する。

(3)(必要ならば)手順(2)で考案した対応が正しいかを検証する。この検証には、我々のグループが開発した検証支援系を用いて行う。

(4)遷移の対応どおりに状態遷移が実行されるように、制御部の値を表すCONTROLを詳細化し、VALID関数を導く。

今までに、レベル1から、レベル7(論理設計レベル)まで段階的に設計を行い、ソート回路Hに対応する回路を得た^[2]。現在、ソート回路Mに相当する回路を得るまでの設計を行っているが、レベル5まではソート回路Hを得るまでの設計と同じ設計を行い、レベル5から、手順Aを用いた詳細化を2度行い、各状態遷移が1つのマイクロプログラムで実行できるレベル7'を得た。レベル7'から、制御部にマイクロプログラム方式を採用することを考慮した設計(以下手順B)を行う(図2)。

3. マイクロプログラム方式順序回路への詳細化例

レベル7'では、状態遷移としてmove_J_A(Jの値をAに転送する)などのように図3のダイアグラムでデータ転送を行うことを想定している。ただし、図3の部品や接続関係の記述自身は次のレベルで行う。レベル7'(上位レベル)から、次の手順(手順B)により、レベル8'(下位レベル)の設計を行う。

(1)制御部(図3の破線内)以外の部品の決定

例えば、16bitのアップダウンカウンタAの機能は、図4のように記述される。ctl_Aは2bitの変数でカウンタの制御入力を表し、data_Aは16bitの変数でロード実行時のデータ入力を表す。

```

A(CK_A(s_A,ctl_A,data_A))==
  if ctl_A = LL then A(s_A)
  else if ctl_A = LH then A(s_A) + 1
  else if ctl_A = HL then A(s_A) - 1
  else if ctl_A = HH then data_A (* ロード実行時 *)
  else A(s_A);

```

図4 用いる部品の記述例

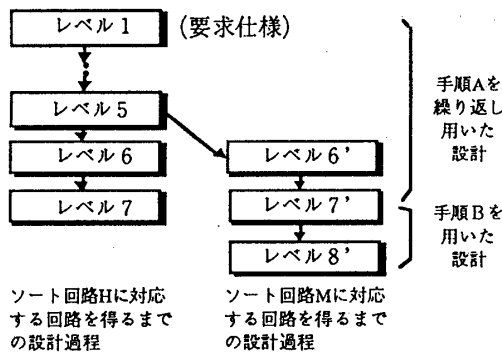


図2 ソートICの階層的設計

(2) 制御部以外の部品の入出力の接続関係等の指定

手順(1)で定義した各部品の入力に対し、どの部品の出力、どのバス、マイクロコードのどのフィールドがそれぞれ接続されるかを、指定する(図5)。例えば、部品Aの入力ctl_Aは、micro-codeの1bit目及び2bit目からなる2bitに接続され、入力data_AはALUの出力に接続される。

```
define architecture =
  [CK_A(s_A,ctl_A,data_A)
   CK_B(s_B,ctl_B,input1,input2)
   ;
  ]
  where ctl_A = [get(micro-code,1),get(micro-code,2)]
  where data_A = ALU
  where ctl_B = [get(micro-code,3),get(micro-code,4)]
  where ALU = ... ;
```

図5 各部品の接続関係

(3) 上位レベルの各状態遷移を実現するマイクロコードの考案

レベル7'の各状態遷移を実現するように、図5で定義される接続関係にどのようなmicro-codeを与える(図6)

```
move_J_A(s) == architecture
  where micro-code = [LL,HH,...];
inc_A(s) == architecture
  where micro-code = [LH,HH,...];
;
```

図6 状態遷移とマイクロコードの対応

(4) (必要ならば) (3)で考案したマイクロコードの検証

図4で与えられる部品に対し、図5, 6を用いて得られる各入力への値を与えたときに、上位レベルで要求される動作(データ転送)を行うかどうかを検証する。

(5) 上位レベルのCONTROL, VALID関数の最適化(改善)

(1)~(4)の手順で、下位レベルにおける1回の状態遷移(1マイクロプログラムの実行)でどのようなデータ転送が可能かが決定できる。状態遷移の実行手順を定義している上位レベルのCONTROL, VALID関数を単純化(CONTROLが表す有限状態名の削減, 実行ステップ数の削減等, いわゆるマイクロプログラムの最適化)を行う。

(6) 制御部の設計及びマイクロプログラムの合成

(a) 図3の破線内の制御部に用いる部品(ここでは μ PC

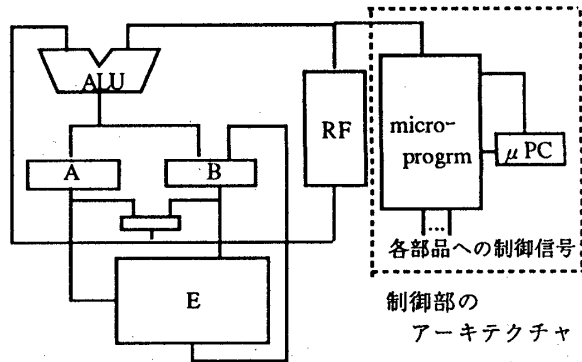


図3 レベル8'で導入されたダイアグラム

のみ)を決定する。(b)制御部の各部品の入出力の接続関係(マイクロプログラムのアドレス指定がどの部品(ここでは μ PCのみ, 一般には, 幾つかの部品の出力を組み合わせた値)で行われるかも含め)を決定する。(c)制御部の各部品へのマイクロコードのフィールド(ジャンプ先アドレスを書くフィールドや μ PCの制御フィールド)を決定する。(d)CONTROLが表す有限状態名に着目すると, 上位レベルのCONTROL, VALID関数は, いわゆる, 状態遷移図に相当する(節点が有限状態に, 枝が状態遷移にそれぞれ相当する)。この状態遷移図の各枝に数値(マイクロプログラムのアドレス)を割り付ける。一般には, 例えば, あるレジスタの値によってn通りの分岐を実現したい, という場合もあるが, 飛び先は1つの場合(すなわち, 状態遷移図において, 各節点からでる枝の数が高々2本)に限定すると, この割り付けは, 逐次的に行われる状態遷移系列に連続した数値を割り付けて, 遷移図全体をいくつかの「線分」で表すことに相当する。その数値(アドレス)のところに, その遷移のマイクロコードを配置する。(e)上位レベルのCONTROL, VALID関数で指定される順番どおりにマイクロプログラムが実行できるように(状態遷移図に従った分岐などができるように), 各マイクロコードの制御部の各部品用のフィールドの内容を決定し, マイクロプログラムを完成する。(f)(必要ならば)状態遷移図に従ってマイクロプログラムが実行されるかどうかを検証する。

以上の手順により, レベル8'が得られる。

4. あとがき

現在, 上述の手順Bの(6)の(d)及び(e)等の自動化アルゴリズムの実現を検討している。また, 手順Bの(5)の最適化についても, どのような最適化ルールなら適用することができるか, 最適化ルールを適用する場合どのような部分が自動化可能かなどの検討を行っている。

【参考文献】

- [1] 杉山, 北道, 谷口: “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”, 信学技報, Vol.88, No.52, SS88-7, (1988-06).
- [2] 北道, 杉山, 谷口: “クイックソートICの代数的記述と制御回路の自動生成について”, 情処学会研究会報告, Vol.89, No.14, DT46-13, (1989-02).
- [3] 社団法人 日本電子工業振興協会編: “LSI設計用記述言語の標準化に関する調査研究”, (1988-03).