

図的言語を用いた装置機能設計支援法[†]

小林一夫^{††}

大規模な論理装置の VLSI 化にともない、論理設計の高信頼度化と省力化とが強く望まれるため、機能設計の段階でハードウェアを記述し、それからゲート回路レベルに自動展開する手法が各所で研究されている。その一環として、図的言語による直感的なわかりやすさをねらった機能的記述法がいくつか提案されているが、いずれも制御回路の動作の時間的な条件を表現する手段が不十分である。制御回路の動作の時間的条件は、ハードウェアの処理性能を決定する重要な要因であり、機能設計の段階から明確に表せる必要がある。本稿では、一種の状態遷移図にもとづいて制御回路の動作を記述できる図的言語と、それからのゲート回路合成手順を述べる。ここでは、時間条件の明確な記述を可能とするため、信号作成タイミングや信号の持続時間のような時間経緯と複数(多相)のクロックを表せる記法を提案した。さらに、多相クロック制御方式をもちいた回路の合成を可能にするため、状態遷移の自動補正手順と多相クロック用マクロ回路を考案した。本手法を電子交換機の入出力制御装置(約 2k ゲート)を構成する 4 種類の制御回路(合計で約 300 ゲート)に適用した例では、2 相クロックで動作する制御回路を、人手によるゲート回路レベルの設計にくらべ、約 8 割の記述量と約 1.5 倍強程度のゲート回路規模で合成できた。

1. まえがき

大規模な論理装置の設計では、設計を体系的におこなうため、ハードウェアの仕様を決める方式設計、その仕様を実現する機能を決める機能設計、各機能をゲート回路の組合せで具体化する論理設計および製造情報を作成する実装設計という手順がとられる。一方、集積回路技術の飛躍的な発達により、数 10 k ゲート以上の超高集積回路(VLSI)が製造可能となり、また VLSI のレイアウトパタン設計の自動化も進んできた。そのため、論理装置への VLSI の適用が図られるようになってきた。VLSI を適用した装置の設計では、VLSI の回路に設計誤りがあった場合、従来の中規模以下の集積回路による装置とくらべ修正に桁違いの期間と費用を要する。このため、高い設計品質が確保できる設計手法の必要性が高まってきている。さらに、機能ならびに論理の設計に大半の工数を費やすようになるため、設計の省力化が強く望まれている。

このような背景のもとに、機能設計の段階からの自動化をめざした設計支援の研究が活発におこなわれるようになり、ハードウェア機能記述言語や、それにもとづくゲート回路合成手法がいくつか提案されている^{1), 2)}。このうち、機能記述言語の最近の研究では、高度な図形処理機能が安価に利用できるようになった

ことを生かして、従来のテキスト形式の言語に代わる図的言語の実現が検討されるようになってきた³⁾⁻⁵⁾。図的言語をもちいれば、機能ブロック図や状態遷移図のように論理のつながりや制御の順序を二次元で表現することが可能となり、複数の機能が独立に働いたり、相互作用をもったりする並列性を自然に記述でき、また設計内容を直感的にわかりやすく記述できる。

しかし、機能を動作として表現するこれまでの図的言語には、制御動作の時間的な条件の表現法にまだ問題が残されている。すなわち動作の相対的な順序のみに着目しているため、制御信号の持続時間のような制御タイミングや、多相クロックを用いた制御を明確に記述できない。制御動作の時間的条件は、ハードウェアの処理性能を決定する重要な要因であり、機能設計の段階から明確に表す必要がある。

本稿では、このような問題を解決するため、制御タイミングや多相クロックのような時間的条件が記述できる図的言語と、図的言語にもとづくゲート回路合成手順を提案する。さらに、図的言語とゲート回路合成手順を装置に適用した結果から本手法の効果を明らかにする。

2. ハードウェア機能記述言語

2.1 機能記述法

ハードウェアの機能を表す方法には、機能の論理的なつながり(構造)を記述する方法と、実行すべき動作を記述する方法がある。これらの方法については、記述しやすさや、わかりやすさの観点から適用分野を

[†] Hardware Design Support with Graphical and Functional Description by KAZUO KOBAYASHI (Switching Control Systems Section, NTT Musashino Electrical Communication Laboratories).

^{††} 日本電信電話(株)武蔵野電気通信研究所処理方式研究室

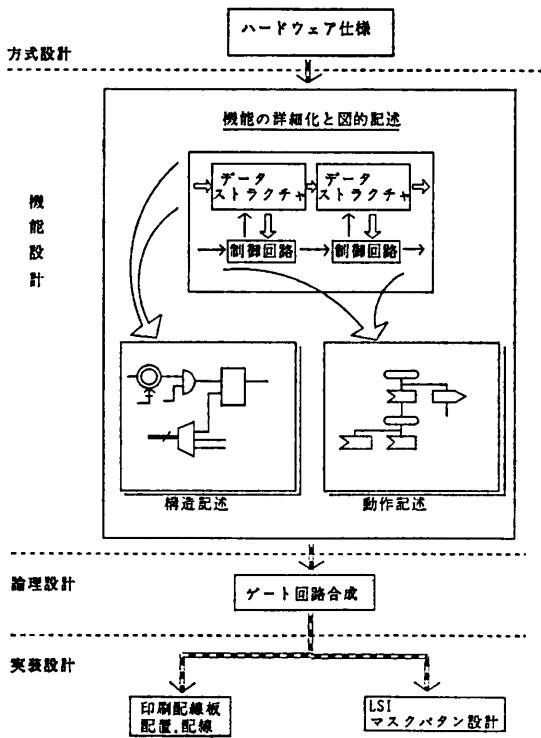


図 1 設計手順
Fig. 1 Design procedure.

分けることができる。すなわちレジスタや演算回路で構成されるデータストラクチャ回路のようにデータの転送される道筋に着目する機能には、論理ブロック相互の接続関係が明確である必要があるため、構造を記述する方法が適する。一方、データ転送を制御する回路のように制御手順が重要な機能には、動作の遷移がわかりやすい必要があるため、動作を記述する方法が適する。

ここでは、このような構造記述と動作記述の特徴を考慮して、図1に示すように、両者を使いわけて機能を表現する方法をとった⁶⁾。なかでも、制御回路については、厳密な時間の規定と複雑な制御の論理を記述できることが要求されるため、次のような条件を考慮して、図的な動作記述言語を考案した。

2.2 動作記述言語の必要条件

機能設計支援における記述言語は、次の条件を満たす必要がある。

- 1) 計算機処理のため、形式的であること。
- 2) 分担して設計する他の設計者との情報交換が容易なように、記述の内容がわかりやすいたこと。

これらに加えて、動作を記述するためには、次が必要となる。

3) 複雑な制御論理の整理に役立ち、それを簡明に記述できること。

これまでの機能設計では、内容の理解が容易なタイムチャートや動作フローチャート等の設計図面を用いてきたが、これらの図的な表現法は、計算機処理のための形式性に欠けていた。そのため、次の特徴が活用できる交換システム仕様記述言語 (SDL⁷⁾: functional Specification and Description Language) をもちいて動作を記述する方法をとることにした。

(1) 制御の動作を、制御回路と被制御回路との間の信号のやりとりとして形式的に、かつ、簡明に表せる。

(2) 一種の状態遷移図記法であり、動作を図的に表せるため、ハードウェアに多い制御機能の並列性や、動作内容が直感的にわかりやすい。

しかし、SDLは、本来、システム仕様レベルの記法であり、そのままでは、必ずしもハードウェアの機能を記述するのに十分とはいえない。そのため、時間的条件を明確に記述できるように、つぎの記法をあらたに加えて機能設計への適用を図った。

2.3 時間的条件の記述の導入

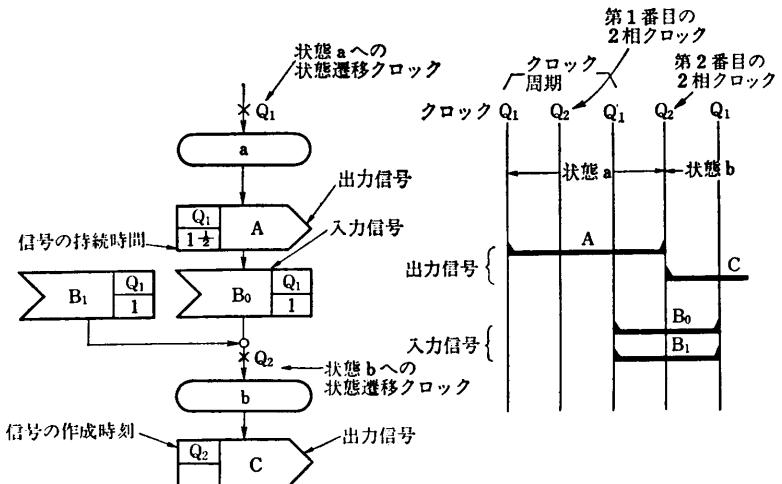
ハードウェアの処理速度は、データがレジスタからレジスタへ動く時間や、外部からの信号に応答する時間などのような動作時間に左右される。したがって、データ転送や信号応答等を制御する機能では、制御信号をいつ作成し、どの期間有効とするかが重要である。そのような機能を記述するには、信号の作成順序および信号作成タイミングや信号の持続時間のような時間経緯を明確に表す必要がある。

また、高速な制御を実現する方法には、処理の遅延に応じて同期タイミングを適当に選べるように複数の種類(多相)のクロックを用いる制御方式がある。この制御方式の記述が可能なように、多相クロックを扱える必要がある。

これらの時間遷移や多相クロックのような時間的条件を記述可能とするため、つぎの記法をあらたに設けた。

(1) 同期クロック

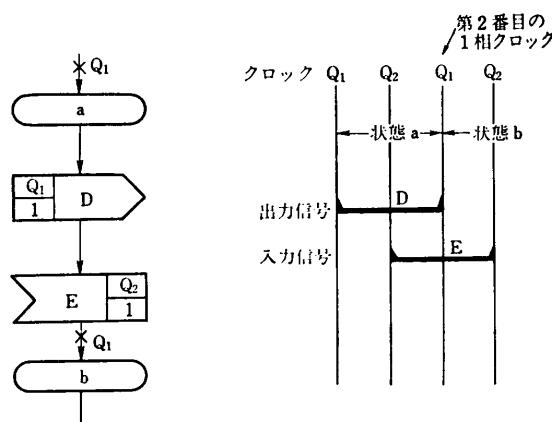
時間的条件は、以下の(2), (3)に示すように、同期するクロックをもとに表す。その際に、多相クロックを表すため、クロック名を Q_n (n は第 n 番目の相) と表示する。図2, 図3に示すように、一つの状態からつぎの状態への遷移の途中および制御信号の図記号の中に記述する。



(a) 動作記述
(a) Behavior description using extended
SDL (functional specification and
description language).

(b) タイムチャートによる(a)の説明
(b) Explanation of (a) using
time-chart.

図 2 時間条件の記述 (その 1)
Fig. 2 Example of description of a signal holding time that
is 1.5 clock cycle.



(a) 動作記述
(a) Behavioral description
using extended SDL.
(b) タイムチャートによる
(b) Explanation of (a)
using time-chart.

図 3 時間条件の記述 (その 2)
Fig. 3 Example of description of a signal holding
time that is 1 clock cycle.

(2) 制御信号の持続時間

制御信号は、一つの状態からの出力信号として表す。出力信号の任意の持続時間を簡明に表すために、クロック周期をもとにした持続時間を出力信号に付随して記述するようにした。たとえば、出力信号 A の持続時間は $1\frac{1}{2}$ (図 2 (a) 参照) で、クロックの 1 周期半であることを表している。一方、出力信号 D

の持続時間は 1 (図 3 (a) 参照) で、クロックの 1 周期であることを表している。

(3) 制御信号の作成時刻

本記法では、入力信号 (他の制御回路の出力信号) の組合せで状態遷移の条件を表し、それらの入力信号の持続時間内に来た最初のクロックでつぎの状態に移ることを表す。すなわち制御信号が有効となる時刻 (作成時刻) は、ある状態からその制御信号をもつ状態に遷移するときの同期クロックで表すようにした。図 2 では、状態 a への入力信号 B₀ または B₁ が到来したあとの最初の Q₂ クロック (第 2 番目の Q₂) のとき、状態 b に遷移し、出力信号 C が有効になる。

以上をまとめて動作記述の図記号を表 1 に示す。『状態』、『入力信号』、『出力信号』では、SDL の本来の図記号に、用法欄の * 付きの記法 (作成時刻、持続時間など) を追加した。『判定』、『状態遷移時刻』は、あらたに追加した図記号である。『判定』は、用法欄に示すように真理値表の形式で入力信号の組合せと状態遷移先とを一括して表す。『状態遷移時刻』は、図記号に付随したテキストで状態遷移を起動する同期クロックを表す。また、『状態』の用法欄の状態遷移名は、状態遷移図の初期状態に付記されて、その状態遷移全体を代表する名称である。

表 1 動作記述のおもな図記号
Table 1 SDL extended to describe hardware behavior.

分類	図記号	意味	用法
状態	○	一つの状態の始まり	ADCTL 状態遷移名*
入力信号	>	状態遷移を引起す信号の入力	SIG Q1 作成時刻* 持続時間*
出力信号	→	制御信号の出力	@OUT Q1 作成時刻* 持続時間*
判定*	□	入力信号による状態遷移の条件を一括して表現する	A0 A1 X Y 遷移条件確定 A0 A1 X Y 入力論理値 X Y 出力名
状態遷移時刻*	*	状態遷移を起動する同期クロック	*Q3 状態遷移クロック AD3

注) *付きの記法はSDLに追加した仕様

3. ゲート回路合成法

3.1 ゲート回路合成上の考慮

制御回路は、現在の状態を記憶する状態回路、現在の状態と入力信号から出力信号を決める出力回路およびつぎの状態を決める状態選択回路とで構成できる。機能記述から、このような制御回路を合成する場合、従来の手順⁸⁾では、おもに単相クロックによる同期動作を前提としていた。それに対して、制御信号の保持時間や多相クロックなどの時間的条件をふくむ機能記述から、コンパクトな制御回路を合成するには、つき

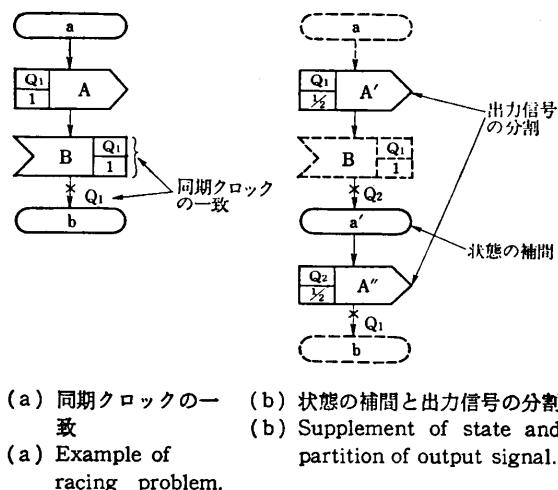


図 4 同相論理の回避
Fig. 4 Method to solve racing problem.

のような考慮が必要である。

(1) 同相論理の回避

構造の最も簡単なディレイ形のフリップフロップ(FF)を状態回路に適用するためには、回路の動作が不安定にならないように、入力信号の同期クロックと状態遷移の同期クロックとが一致しないようにする必要がある。たとえば、図4(a)のように、入力信号が無効となる時刻(作成時刻から持続時間だけ過ぎたあと)と状態遷移の同期クロックが一致する場合(同相論理)、FFの駆動に要する時間的な余裕が少ないため、合成した回路の実装条件によって信号がFFに到達する時刻が異なると、所期の時刻でFFが駆動されないことがある。

(2) 指定の時間長の信号作成

2章で述べたように、本記法では、出力信号の持続時間を、たとえば $1^{1/2}$ のように(図2(a)参照)信号に付随して指定するため、クロック周期の $n^{p/q}$ (ここで、n, p, qは0でない正の整数)倍の時間長の信号を作る出力回路の合成が必要である。

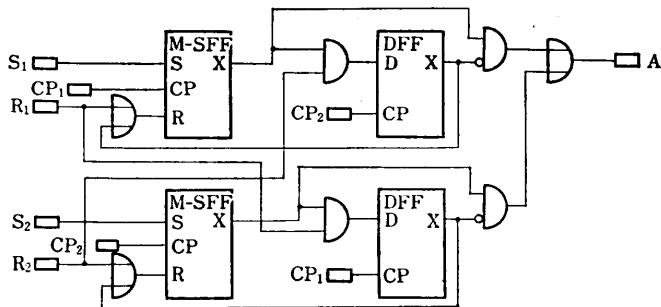
(3) クロックおよび信号の分離

デバッグや詳細なタイミング解析には、論理がわかりやすい単純な回路構成が望まれる。わかりやすい構成のゲート回路を合成するためには、クロックと信号を明確に分離する必要がある。さらに、状態回路の各FFについては、一つのFFは多相のクロックのなかの唯一のクロックで動作するように合成する必要がある。

これらを考慮して、つぎのようなゲート回路合成手順とした。

① 動作記述から状態遷移の条件を作成する。

現在の状態からつぎの状態への遷移は、制御回路への入力信号と状態遷移の同期クロックとの組合せ(遷移入力条件)で決まる。そのため、まず状態と遷移入力条件とからなる状態遷移条件を動作記述から抽出する。同相論理となるのを防ぐため、同期クロックの一一致している遷移入力条件があれば、対象の状態と次に遷移する状態の間にあらたな状態を追加し、これを介した遷移に変更する(図4(b)参照)。追加する状態としては、同期クロックが入力信号のものと異なる状態を選ぶ。さらに、持続時間が1クロック周期以上(たとえば $n^{p/q}$)の出力信号をもつ状態に対しても、



注) 本図は2相クロック用であり、状態値の1要素を表す。
 $CP_i (i=1, 2)$ は、同期クロック Q_i に対するクロック端子。
 $S_i (i=1, 2)$ は、同期クロック Q_i に対するセット端子。
 $R_i (i=1, 2)$ は、同期クロック Q_i に対するリセット端子。
M-S FF は、マスタスレーブ形 FF, DFF は、ディレイ形 FF.

図 5 多相クロック用 FF マクロ回路

Fig. 5 Macro-circuits to compose control-circuits using multi-phase clock (example of using 2 phase clock).

持続時間が1クロック周期の状態を n 個と p/q クロック周期の状態とに分割するように状態を追加し、指定の時間長の信号を作る。

② 状態値を割当てる。

各状態に、それぞれユニークになるように2進数で表現した値（状態値）を割当てる。ここで、状態値の初期値は、状態遷移名が付記された状態に割当てる。

③ 制御回路に変換する（構造変換）。

①, ②で得られた状態遷移条件と状態値から、マクロ回路を用いた制御回路を作る。一般にマクロ回路とは、FF やセレクタなどの標準的な回路で、特定の回路テクノロジに依存しない回路をいう。前記のように論理のわかりやすい回路を合成するために、図 5 の多相クロック用 FF マクロ回路を導入した。図 5 は2相クロックではたらく制御回路用のマクロ回路である。冗長な回路が残らないようにするために、マクロ回路の中で使われなかった端子と回路は、最後に削除する。

④ マクロ回路をゲート回路に変換する。

マクロ回路の論理構成は、ゲート回路によって設計されて設計ライブラリに登録されている。この設計ライブラリを参照してマクロ回路をゲート回路に変換する。このとき、否定回路の削除などの回路簡略化とファンアウト数の調整のような電気条件への適合化をおこなう^{9), 10)}。

以下では、時間的条件を考慮した ① 遷移入力条件の抽出、と ③ 構造変換、の手順の詳細を述べる。

3.2 ゲート回路合成手順

3.2.1 状態遷移条件の作成

状態 S_i から状態 S_j ($i \neq j$, $i, j = 1 \sim n$, n は状態

数) に遷移するための入力条件 ($C_i (S_i \rightarrow S_j)$) を動作記述から抽出して、つぎのように入力信号と状態遷移の同期クロックとの符号ベクトル (a_p : 入力ベクトル, $p=1 \sim q$, q は遷移入力条件となる入力ベクトルの数) で表す。

$$C_i (S_i \rightarrow S_j) = a_1^{\wedge} a_2^{\wedge} \cdots = (a_{11} a_{12} \cdots a_{1m} q_{11} \\ \cdots q_{1t})^{\wedge} (a_{21} a_{22} \cdots a_{2m} q_{21} \cdots q_{2t})^{\wedge} \cdots$$

ここで、 $a_p (r=1 \sim m, m$ は入力信号数) および $q_{pt} (u=1 \sim t, t$ は状態遷移の同期クロックの相数) は、入力信号と状態遷移の同期クロックの論理値を表し、『1』、『0』または『*』（無効）で表現される。

たとえば、図 6において、状態 A から状態 ID への遷移入力条件はつぎのように表す。

$$C_i (A \rightarrow ID) = (**1 * 1 *)^{\wedge} (**1 1 *).$$

ただし、上記の入力ベクトル $a_p = (CS\ DS\ NG\ AK\ Q_1\ Q_2)$ では、CS, DS, NG, AK が入力信号で、 Q_1, Q_2 が状態遷移の同期クロックである。すなわち状態 A から状態 ID への遷移は、NG または AK が『1』のときに Q_1 クロックを同期としておこなわれる。

このような遷移入力条件に、つぎの手順で状態を補間し、同相論理を防ぐ。

Step 1 : 状態 S_i から S_j への遷移入力条件のなかから、入力信号の同期クロックと状態遷移の同期クロック (q_0) とが一致する入力ベクトル（これを a_0 とする）を一つ選ぶ。なければ、終了する。

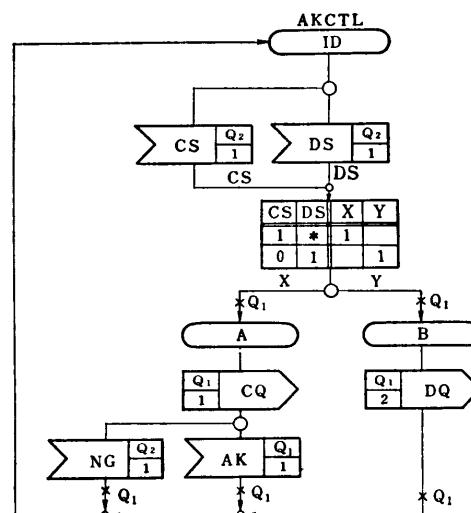


図 6 動作記述モデル
Fig. 6 Behavioral description model for explanation of synthesizing gate level circuits.

Step 2: 状態 S'_i を S_i と S_j の間に追加し、 a_o による $S_i \rightarrow S_j$ の遷移を $S_i \rightarrow S'_i \rightarrow S_j$ に変更する。それに伴い、各状態遷移条件を以下のように修正する。

a) a_o において、有効(『1』または『0』)であるどの入力信号の同期クロックとも一致しない同期クロックを一つ選ぶ。これを q_{oo} とする。なければ終了する(この場合、同期クロックの一一致を避けるように、動作記述を変える必要がある)。

b) a_o において、状態遷移の同期クロックを q_o から q_{oo} に置換えた入力ベクトル(これを a'_o とする)を S_i から S_j への遷移入力条件にふくめる。

すなわち、 $C_o(S_i \rightarrow S_j) = a'_o \wedge \dots$

c) q_o のみを有効な要素としてもつ入力ベクトルを、 S'_i から S_j への遷移入力条件にふくめる。すなわち、 $C_o(S_i \rightarrow S_j) = (* \dots * q_o * \dots *) \wedge \dots$

Step 3: 未処理の遷移入力条件があれば、Step 1へ戻る。なければ終了する。

図 6 の記述をもとに上記の手順で作成した状態遷移条件を図 7 に示す。同図(a)は、動作記述から抽出した状態遷移条件である。第一行は、遷移入力条件を表

す。第二行以下では、最左端の列が遷移前の状態、第二列より右では、各列の遷移入力条件によって遷移する先の状態を表す。先に例示した $C_o(A \rightarrow ID)$ のうち、第一番目の入力ベクトル($\ast\ast 1 \ast 1 \ast$)では、入力信号 AK と状態遷移の同期クロック Q_1 が一致する。すなわち $a_o = (\ast\ast 1 \ast 1 \ast)$, $q_o = Q_1$ 。そのため、状態 A' を追加して、 $C_o(A \rightarrow A') = (\ast\ast 1 \ast \ast 1)$ 。これは、状態遷移の同期クロックを Q_1 から、AK の同期クロックと一致しない Q_2 に換えたものである。さらに、 $C_o(A' \rightarrow ID) = (\ast\ast\ast\ast 1 \ast)$ 。一方、同図(b)では、状態 B から ID への遷移にも、間に B' を追加し、 $B \rightarrow B' \rightarrow ID$ と修正している。これは、出力信号 DQ の持続時間がクロック 2 周期分のため、状態 B と B' とでクロック 1 周期分ずつ信号を作成するようにしている。

3.2.2 構造変換

ある入力ベクトルに対して、遷移の前後で値の変化する状態値の要素(3.2 節で述べたように状態値は2進数で表現されており、その1ビットをいう。これを f_i ($i=1 \sim n$, n はすべての状態を表すのに必要な最小のビット数)とする)をみつける。この入力ベクトルを、該当の要素の駆動条件とよぶ。とくに、値が『0』から『1』に変化する場合をセット駆動条件、『1』から『0』に変化する場合をリセット駆動条件とよぶ。状態値の要素と駆動条件とから、多相クロックで動作する制御回路を、つきの手順により合成する。

Step 1: 状態値の各要素のセット駆動条件、リセット駆動条件をえる。

		CDANQQ SSKG12	1***1*	01***1*	**1*1*	***11*	****1*
ID	A	B					
A			ID	ID			
B						ID	

(a) 動作記述から抽出した状態遷移条件

(a) Condition of state transition extracted from behavioral description.

		CDANQQ SSKG12	1***1*	01***1*	**1*1*	***11*	****1*
ID	A	B					
A			A'	ID			
B					B'		
A'						ID	
B'						ID	

(b) 状態を補間した状態遷移条件

(b) Condition of state transition modified for matching time condition.

図 7 状態遷移条件
Fig. 7 Condition of state transition.

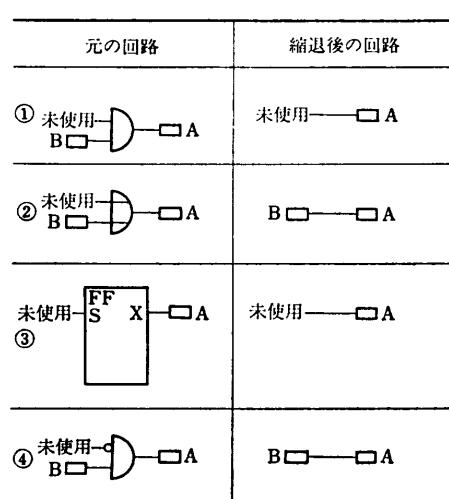


図 8 FF マクロ回路の縮退ルール
Fig. 8 Reduction rules of FF macro-circuits.

状態	状態値		
	f_1	f_2	f_3
I D	0	0	0
A	1	0	0
B	0	1	0
A'	0	0	1
B'	1	1	0

(a) 状態値割当て (a) Status values assigned.				
入力信号	元の状態 (デコード した値)	次の状態		
		f_1 S R	f_2 S R	f_3 S R
CDAN QQ	0	1		
SSKG 12	2	1		
**** 1*	4		1	
*** 1*	4		1	
**** 1*	6		1	
01** 1*	0		1	
**** 1*	6		1	
*** 1*	4			1
**** 1*	1			1

(b) FF 駆動条件 (b) Condition of driving flip-flops.			
状態 (デコード した値)	出力信号		CQ DQ
	4	1	
1		1	
2			1
6			1

(c) 出力信号条件 (c) Condition of output signals.			
状態 (デコード した値)	出力信号		CQ DQ
	4	1	
1		1	
2			1
6			1

図 9 構造変換の条件
Fig. 9 Condition of translation from behavioral description into macro-circuits.

Step 2: 要素 f_i のセット駆動条件の同期クロックが Q_i の場合, f_i のセット駆動条件のうちの入力信号を, f_i を実現する FF マクロ回路の Q_i に対応するセット端子の条件とする(図 5 参照). 同様にリセット駆動条件のうちの入力信号を, Q_i に対応するリセット端子の条件とする.

Step 3: すべての要素の駆動条件を対応づけたあと, 図 8 の縮退ルールをもちいて, マクロ回路の未使用端子と回路を削除する.

図 7 の状態遷移条件をもとに作成した構造変換のための条件を図 9 に示す. 図 9 (a) は, 状態遷移条件にふくまれる五つの状態に状態値を割当てた結果である. 同図(b) は, 状態値を表す FF の駆動条件であり, 前記の状態値と状態遷移条件をもとに, 上記の Step 1 の処理で作成する. たとえば第一行は, 状態値が $(f_1 \ f_2 \ f_3) = (0 \ 0 \ 0)$ (図では, 10 進数に直した値で示す) のとき, 入力ベクトル (CS DS AK NG Q1

$Q_2) = (1 \ * \ * \ * \ 1 \ *)$ によって, f_1 がセットされることを表す. 同図(c) は, 先の状態値と動作記述から作られる出力信号の作成条件である. 図 10 は, さらに, 上記の Step 2, 3 より作成される回路である. 図 3 の縮退ルールの適用例を f_1 について示した. 点線が縮退された回路で, 付記した ①~③ は適用するルールを示す.

4. 適用結果

表 2 に, 本手法を電子交換機の入出力制御装置の制御回路の機能設計に試行した例を示す. この入出力制御装置は, 全体の回路規模が約 2k ゲートあり, このうちの約 300 ゲートが制御機能である. 中央制御装置や入出力装置との間の転送制御を行うインターフェース制御回路などの 4 種類の制御回路により構成されている.

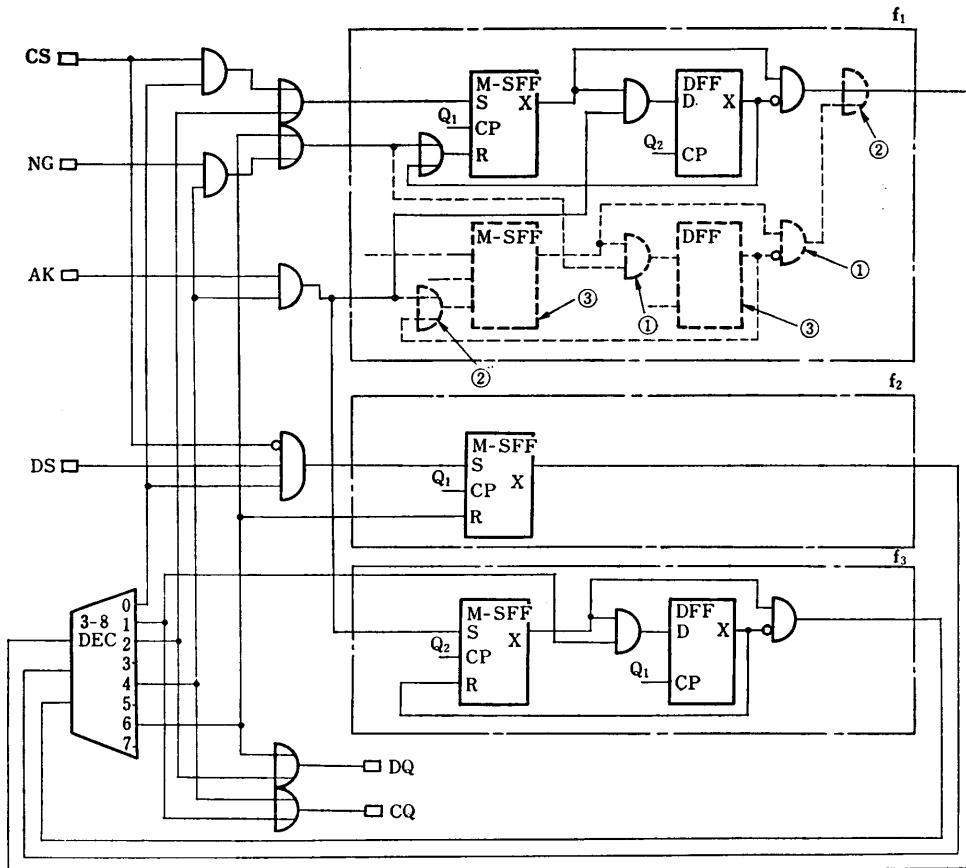
この適用例では, 2 相クロックで動作する制御回路を合成した. その際に, とくに記述性に関する問題ではなく, 記述量(ゲート回路図, 動作図および構造図における図記号の数)は, ゲート回路レベルまで人手で設計した場合にくらべて約 8 割に削減された. 一方, 合成されるゲート回路規模は, 人手設計によるものより約 55% 多くなかった.

5. むすび

ハードウェア機能設計支援のために, 制御回路の機能仕様を動作図をもつて図的に記述し, ゲート回路を自動合成する手法を述べた. 本手法によれば, 時間的条件を明確にした機能記述が可能で, 多相クロックをもちいた制御方式の回路を合成できる.

電子交換機の入出力制御装置(約 2k ゲート)を構成する 4 種類の制御回路(合計で約 300 ゲート)に適用した例では, とくに記述性に関する問題ではなく, 記述量はゲート回路レベルにくらべ約 8 割に削減できた. さらに, 人手設計より約 5 割強程度の回路規模の増加で 2 相クロックで動作する制御回路が合成できた.

なお, ゲート回路合成に要する時間は, 一時に処理する動作記述のなかの状態数が増えると急激に増加することが予想される. それにたいしては, ここでは動作記述の対象となる機能を適当な規模にわけて記述することにより実用的な処理時間で合成できると考えられる. 今後, 本記述法にもとづく機能検証法などと組合わせた総合的な機能設計支援システムの構築が課題



注) 一点鎖線で囲んだ f_i ($i=1 \sim 3$) が、状態値の各要素を表す FF マクロ回路。 f_i については、縮退した回路を点線で、そのときのルールを ①～③ (図 8 参照) で示してある。

図 10 モデル回路の生成例

Fig. 10 Example of circuits synthesized from description of Fig. 6 that also shows reduced circuits with dotted lines.

表 2 記述量と合成される回路規模

Table 2 Experimental results of behavioral description and of synthesizing gate level circuits from its description.

比較項目	対象回路				
	インタフェース制御回路	保守制御回路	バス制御回路	状態表示制御回路	計
記述量 (個)	本手法の適用結果	75	16	51	26
	人手設計	92	19	64	27
ゲート回路規模 (ゲート)	本手法の適用結果	194	30	102	50
	人手設計	109	25	83	26

注) 記述量は図記号の数。[] 内は人手設計を 1 としたときの本手法の比。人手設計はゲート回路レベルを対象とする。

である。

謝辞 最後に、本検討に当たり、ご指導、ご助言を賜った、日本電信電話株式会社 武藏野電気通信研究所 前処理装置研究室、丹羽昭男室長、松浦洋征調査員および室員各位に深謝します。

参考文献

- 1) vanCleempunt, W. M.: Computer Hardware Description Languages and Their Applications, 16 th DA Conf., pp. 554-560 (1979).
- 2) Thomas, D. E.: The Automatic Synthesis of Digital Systems, Proc. of the IEEE, Vol. 69, No. 10, pp. 1200-1211 (1981).
- 3) Parasch, G. J. and Pricle, R. L.: Development and Application of a Design Oriented Cyclic Simulator, 13 th DA Conf., pp. 48-53 (1976).

- 4) 西川, 寺田, 浅田他: 図的表現によるデータ駆動型並列処理記述言語の一提案, 昭和 56 年度信学会情報・システム部門別全国大会予稿集, p. 2-250 (1981).
- 5) 藤田, 田中, 元岡: 入出力に图形表示を用いた機能レベル論理設計支援システム, 情処学会第 24 回全国大会予稿集, 6 J-4 (1982).
- 6) 小林: 動作・構造記述を用いた装置機能設計サポートシステムの構想, 情処第 25 回全国大会予稿集, 6 N-3 (1982).
- 7) CCITT Sixth Plenary Assembly: Orange

- Book, Vol. VI. 4, Programming Language for Stored-Program Control Exchanges (1976).
- 8) 当麻: 順序回路論, p. 273, 昭晃堂, 東京 (1980).
- 9) 増尾, 小林, 松浦他: 機能設計サポートにおける回路簡単化手法, 昭和 57 年度信学会全国大会予稿集, p. 6-135 (1982).
- 10) 若林, 小林, 増尾: 機能設計サポートにおけるマクロ展開法の考察, 情処第 26 回全国大会予稿集, 3K-10 (1983).

(昭和 59 年 6 月 4 日受付)
(昭和 60 年 1 月 17 日採録)