Vol. 26 No. 6 情報処理学会論文誌 Nov. 1985

ショートノート

論理シミュレータにおける機能ブロック化について†

伊藤孝之†† 楠 菊信††

論理シミュレーションにおいて，あるまとまった機能をもつ回路の機能ブロック化を行い，入出力関係と遅延によりこれを表す三つの方法について提案する．そして，これらとゲートレベルの4方式について，処理時間，メモリ量，記述量の比較を行い，各方式の特徴とその適用領域について述べる．

1. まえがき

半導体の高集積化に伴い，設計ミスの許されないLSIチップの作製は著しく困難なものとなっている．これを解決するために，デバグ済みの回路を反復利用すること，これらを階層的に構成してより大きな機能ブロックを積み上げる階層化設計法等が講じられている．この考えを論理回路シミュレーションにも適用し，あるまとまった機能をもつ回路をまとめて機能ブロック化し，一つの素子として扱うことのできる論理シミュレータ LAP-S1 を作成した．LAP-S1 では，この機能ブロックをゲートレベルのままシミュレーションを行うことの他，これを入出力関係と遅延によって表す3種の機能ブロック化の方法をとり得る構成となっている．以下に，その概要と得失について述べる．

2. 本シミュレータの概要

LAP-S1 は回路記述プロセッサ，入力パターンプロセッサ，実行プロセッサ，および出力プロセッサからなる．本題に関する実行プロセッサの処理の流れを**図1**に示す．シミュレーションの方式はイベントドリブン方式である．シミュレータはシミュレーションにおける時間的経過を表す時計をもっている．始めに，現在の時刻において回路に与えられる信号を読みこみ，これをイベント（線の信号が変化すること）としてタイムテーブルの現在時刻の対応箇所に登録する．次に，タイムテーブルの現在時刻の対応箇所に登録してあるイベントを一つ取りだし，そのイベントが生じた線を入力としている素子の出力信号を求め，その信号が現在の信号の値と異なる場合，タイムテーブルの，その素子の遅延時間分だけ現在より後の時刻の対応箇所にイベントとして登録する．このイベント処理がすべて終ると，シミュレータの時計を1単位進めて再び同様の処理を行う．

実行プロセッサの処理の流れの中で回路中の素子の出力信号を求めるに際し，LAP-S1 では以下の4方式の一つを選択することができる．

（1） ゲートレベル方式

一般の論理シミュレータで行われているような，論理ゲートやフリップフロップなどの，動作があらかじめシミュレータに組みこまれている基本的な素子のみによって回路を記述し，ゲート単位で処理を進める．

（2） ハッシュ索引方式

図2に示すように，機能ブロックの入出力関係をあらかじめハッシュテーブルに登録しておき，シミュレーション実行時に入力信号によって出力信号を登録してある場所を求め，これを参照して出力信号の値を求める．この方式では出力信号をハッシュ探索によって求めるために高速に処理が行われる．しかし，探索方法の性質上 don't care の入力に対しては，これを1と0に直して別々に定義しなければならない．また，加算器のような，データ的な性質の信号を扱う素子について，取り得る何通りものデータ信号すべてに対して入出力関係を定義する必要があり，データが多量になる．これを避けるために，素子に入ってくると予想される信号のみについて定義するならば，ある回路のために定義した機能ブロックが他の回路ではそのまま使うことができない場合が生ずる．

（3） 2分木を用いる方式

図3に示すように，機能ブロックの入力信号を2分木で表し，シミュレーション実行時には，一つ一つの信号が，1であるか0であるかを辿ることにより，出

† Functional Block in Logic Simulator by TAKAYUKI ITO and KIKUNOBU KUSUNOKI (Department of Information Science, Faculty of Engineering, Toyohashi University of Technology).

†† 豊橋技術科学大学工学部情報工学系

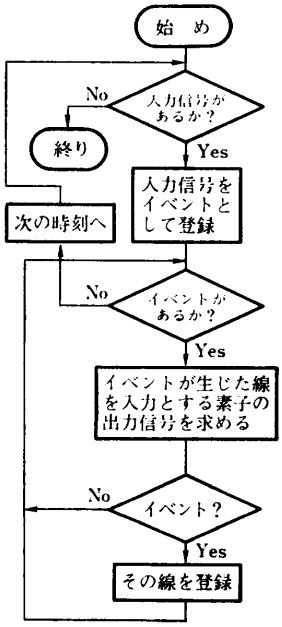

図 1 LAP-S1 の実行プロセッサの処理の流れ
Fig. 1 Flow of execution processor in LAP-S1.

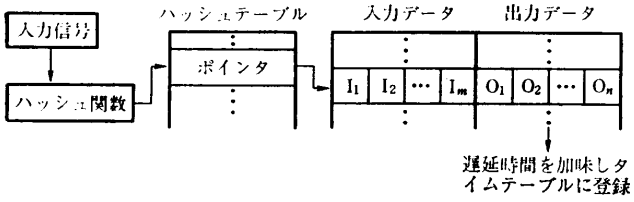

図 2 ハッシュ索引方式
Fig. 2 Method by using hash-table.

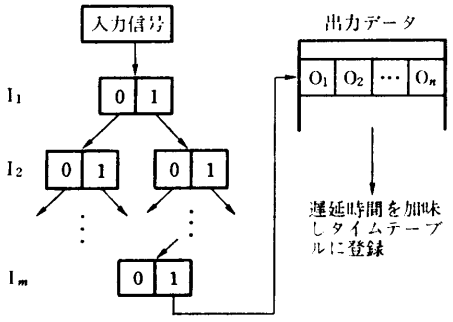

図 3 2分木を用いる方式
Fig. 3 Method by using binary tree.

力信号を求める．このとき，優先度の高い信号ほど，2分木の根に近いノードとする．順序回路は，機能ブロックの出力を入力にフィードバックさせて実現する．優先信号の例を**図4**に示す．CLK が最も優先度が高く，次に CLR, LOAD の順になる．優先信号を扱う素子では（1）の方式より簡潔に定義される．しかし，この方式でもデータを扱う素子では一般にデータ量が多くなり，他の回路への流用にも制限がある．

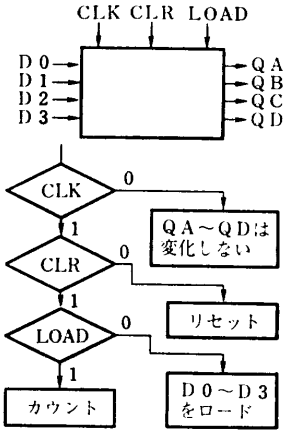

図 4 4ビットプリセッタブルカウンタ
Fig. 4 4 bit presettable counter.

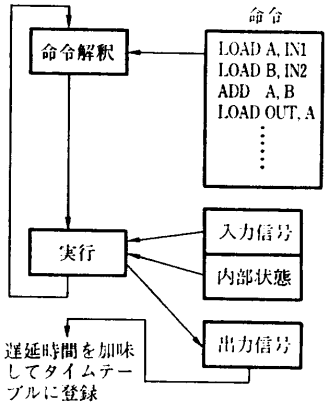

図 5 命令による解釈実行方式
Fig. 5 Method by interpreting instructions.

（4） 命令による解釈実行方式

図5に示すように，機能ブロックの入出力関係を，代入，加算，減算，AND, OR, NOT, シフト（左シフト，右シフト），ジャンプ，条件ジャンプ（二つの信号の等・不等，ある何ビットかの信号間の大小関係，クロック信号の変化の有無などの条件）の命令を用いて表現しておく．シミュレーション実行時には素子に入ってくる入力信号と素子の内部状態を表すレジスタの内容をもとに，命令を解釈実行していき，出力の値を求める（参考文献 3)).

3. 評　　価

特徴のある6回路に対してシミュレーションを行い，上記の4方式について，計算機の処理時間，ゲートレベル方式からのメモリ増加量，回路を記述するためのステップ数を求め，**表1**に示す．表1の(A)にお

表 1　シミュレーションに要した CPU 時間，メモリ量，記述量の比較
Table 1　Comparison of run-time, amount of storage and descriptions required for simulation.

(A)　順序回路
(i)　メモリ増加量：考え得るすべての入力パターン数を記述
(ii)　(2)はクロック入力の処理を非サポートのため除外

回路（優先信号あり）	①非同期16進カウンタ				②同期16進カウンタ			
入力パターン数	9				14			
方　式	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
CPU時間（秒）	1.2	—	0.8	1.2	3.1	—	1.4	2.1
メモリ増加量(バイト)	0	—	336	202	0	—	461	542
記述量（ステップ）	4	—	34	20	33	—	47	32

(B)　組合わせ回路（I）
(i)　メモリ増加量：考え得るすべての入力パターン数を記述

回路（優先信号あり）	③エンコーダ				④デコーダ			
入力パターン数	10				9			
方　式	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
CPU時間（秒）	1.8	1.3	0.9	1.7	1.9	0.9	1.0	1.7
メモリ増加量(バイト)	0	16896	122	488	0	2688	448	654
記述量（ステップ）	29	512	9	52	25	64	18	55

(C)　組合わせ回路（II）
(i)　メモリ増加量：考え得るすべての入力パターン数を記述

回路（*優先信号あり）	⑤エンコーダ*				⑥4ビット加算器			
入力パターン数	100				100			
方　式	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
CPU時間（秒）	11.7	8.6	8.6	9.6	9.0	4.9	4.8	5.1
メモリ増加量(バイト)	0	7000	122	488	0	17408	4248	76
記述量（ステップ）	29	101	9	52	40	512	527	10

注)　(1)ゲートレベル方式，　(2)ハッシュ索引方式，
(3)2分木を用いる方式，　(4)命令による解釈実行方式

いて回路①，②のゲート数はそれぞれ 5，34 である．機能ブロック化のオーバヘッドがあるため①のような小規模な回路では(1)が有利となる．表1の（B）から，(2)の方式は，考え得るすべての入力パターンに対する出力パターンを定義するため，膨大なメモリ増加量になる．しかし，優先信号や don't care がない組合わせ回路なら，回路④のように(2)の方式の有利な領域が存在する．回路③において(2)の方式が(3)の方式より遅くなっているが，これはハッシュテーブルを索引する際の衝突に対処するために要するオーバヘッドのためである．表1の(C)から，優先信号がある回路では，(3)の方式が CPU 時間，記述量の面で有利である．また，回路⑥のような入力信号の組合わせが大きくなるデータ信号を扱う回路では，(4)の方式が，記述量が非常に少なく総合的に有利である．

以上のことから一般的に，回路規模が小さいものでは(1)，優先信号，don't care がなく入力信号の組合わせが少ないものでは(2)，優先信号，don't care があり入力信号の組合わせが少ないものでは(3)，入力信号の組合わせが多いものでは(4)が適している．

4. むすび

階層化表現は回路規模が大きくなるほど効果的である（参考文献 1),2)）．本論文では，機能ブロックの回路の性格と規模によって，機能ブロック化の各方式にそれぞれ適用領域があることを明らかにした．

謝辞　本研究に関し，種々ご助言，ご協力いただいた NTT 研究開発本部丹羽昭男氏，神鋼電機(株)福岡宏明氏に感謝する．

参考文献

1) 土屋，楠：論理シミュレーションにおける機能ブロックの処理と評価，信学論（D），Vol. J 66-D, No. 6, pp. 757-758 (1983).
2) 土屋，楠：論理シミュレーションにおける機能ブロックの順序回路化とその評価，信学論(D)，Vol. J 66-D, No. 6, pp. 1249-1250 (1983).
3) Babacci, M. R.: Instruction Set Processor Specification (ISPS): the Notation and Its Applications, *IEEE Trans. Comput.*, Vol. C-30, pp. 24-40 (1981).

(昭和 60 年 2 月 21 日受付)
(昭和 60 年 6 月 20 日採録)