大規模回路用汎用論理シミュレータ

村上　道郎　　西　謙二　　小沢　康明

1. はじめに

設計された論理回路の論理機能検証法としては，物理的に回路を実現する基板検証法と，論理シミュレータにより模擬的に回路を実現し検証する方法があるが，後者より有効である事は周知の事実である。

ところで，論理シミュレータによる検証法には二つの目的がある。一つは時間依存性をまったく無視し単に論理値計算のみを行なう事であり，もう一つは回路を構成する各素子の遅れ時間，さらに閾値の変化による遅れ時間の変動をも考慮して，回路動作の正確な時間依存性を知る事である。

本稿による論理シミュレータは，上述の二目的をダイナミック回路，スタティック回路等，いわゆる汎用の大規模回路に対して適用可能とするべく開発されたものである。

2. 特長

本システムは以下の特長を持つ。

① ファイルオリエンテッドなシステムである。

シミュレートに必要な回路接続情報，シミュレート系列，シミュレート結果をファイル（4参照）に登録し，必要に応じ取出し，追加，消去，修正を容易に行なえる。したがって，あるシミュレート結果に対しさまざまの角度から論理値を調べる事が可能であり，また，同一回路に対し異なるシミュレート系列で実行させる事も容易である。その結果，回路の誤りを発見すれば修正個所のみを入力としてファイルを更新すれば良い。

② マクロな素子定義が可能である。

ある論理機能を持った複数個の素子の集合をマクロ素子（3参照）として定義し，ファイルに登録しライブラリー化しておく事により，回路接続情報を容易に入力する事が可能である。マクロ素子内で他のマクロ素子の使用も許される。

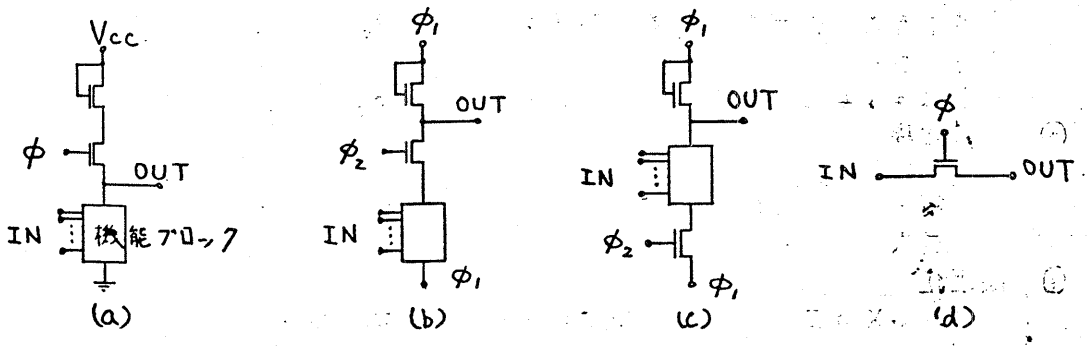


図1. ダイナミック回路用基本素子

③ 基本素子の追加定義が可能である（3参照）。
素子の論理機能をFORTRAN文で記述する事により基本素子の追加定義が可能である。

④ ダイナミック回路のシミュレートが可能である。
前頁図1に示す基本素子[1]がシステムに登録されており，これを使用してダイナミック回路のシミュレートが可能である。

⑤ シミュレート方法を任意に選択出来る。
シミュレート方法として，単純論理値計算を行なう，遅れ時間を含む論理値計算を行なうの二種類用意されており，ユーザーはそれらを任意に選択出来る。また，遅れ時間要素として，最小ターンオン，オフ時間（T_{on}, T_{off}）とそれらのバラツキ等による変動分，すなわち不確定時間（$T_{on\text{-}A}$, $T_{off\text{-}A}$）の四種類を各ゲート独立に定義出来，より正確なシミュレートが可能となっている。

3．機能

本システムは以下の機能を持つ。

① 対象回路
ダイナミック，スタティック，また，RAM，ROM，ランダムロジック等のいわゆる汎用論理回路を対象とする。

② 対象素子
ブラックボックス的に記述可能な論理機能を持つものを「素子」と定義する。素子には，「基本素子」，「マクロ素子」，「追加基本素子」の3種類がある。

(イ) 基本素子
システム中に基本的に組込まれている素子を「基本素子」と定義する。基本素子には，AND，OR等のゲート，外部入出力端子，電源，グランド，RAM，ROM，各種フリップフロップ，ダイナミック素子等がある。

(ロ) 追加基本素子
素子の論理機能，素子名等をFORTRAN文で記述しシステムに組込む事により，その素子をあたかも基本素子として使用出来る。これを「追加基本素子」と定義する。

(ハ) マクロ素子
基本素子，追加基本素子，登録済のマクロ素子のみで定義可能な素子を「マクロ素子」と定義する。マクロ素子はマスタ，サブファイル（4参照）に登録する事によりライブラリー化が可能である。

③ 回路規模
ゲート換算で

標準	5,000 ゲート
最大	100,000 ゲート

④ 論理値
0，1，Xの三値である。ただしXは不確定値である。

4．構成

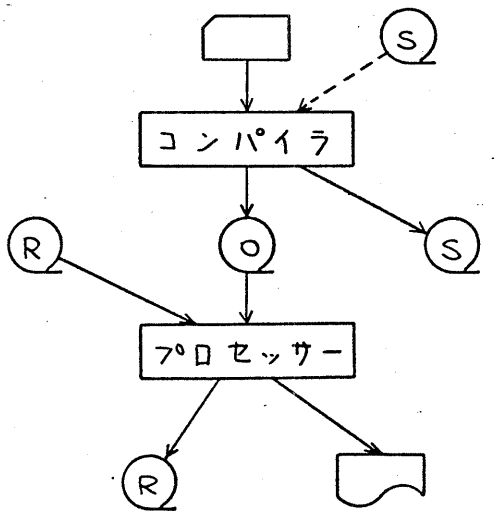
本システムは，コンパイラー，独立な六つのプロセッサー，三つのファイルで構成されており，以下のように定義される。

コンパイラー　入力データ（ソースモジュール）を翻訳し，各プロセッサの入力となるオブジェクトモジュールを作成する。また，ソース，オブジェクトモジュールをファイルに登録する。

プロセッサー　オブジェクトモジュールに従って，実作業を行なう。
マクロ素子定義データよりマクロ素子を合成する「ＭＡＣＲＯ」，全体回路定義データより回路を合成する「ＣＯＮＮＥＣＴ」，シミュレート系列定義データよりシミュレート系列を合成する「ＰＡＴＴＥＲＮ」，実際にシミュレートを行なう「ＳＩＭＵＬＡＴＥ」，シミュレート結果をプリントする「ＴＲＡＣＥ」，ファイルのメインテナンスを行なう「ＦＩＬＥ」の六種があり，前四者の実行結果（リザルトモジュール）は指定されたファイルに登録される。

ファイル　永久記憶ファイル二つ，一時記憶ファイル一つがあり，前者には各ユーザーの共通データを登録しておく「マスタ」，個別データ登録用の「サブ」の二種類がある。

結局，ユーザーがコーディングした入力データはコンパイルされオブジェクトモジュールとなり，プロセッサーによりリザルトモジュールの作成，結果のプリントが行なわれる（図２）。全体流れ図を図３に示す。



S：ソースモジュール
O：オブジェクトモジュール
R：リザルトモジュール
---→：登録済データを修正する場合のみ必要

図２．プログラム流れ図

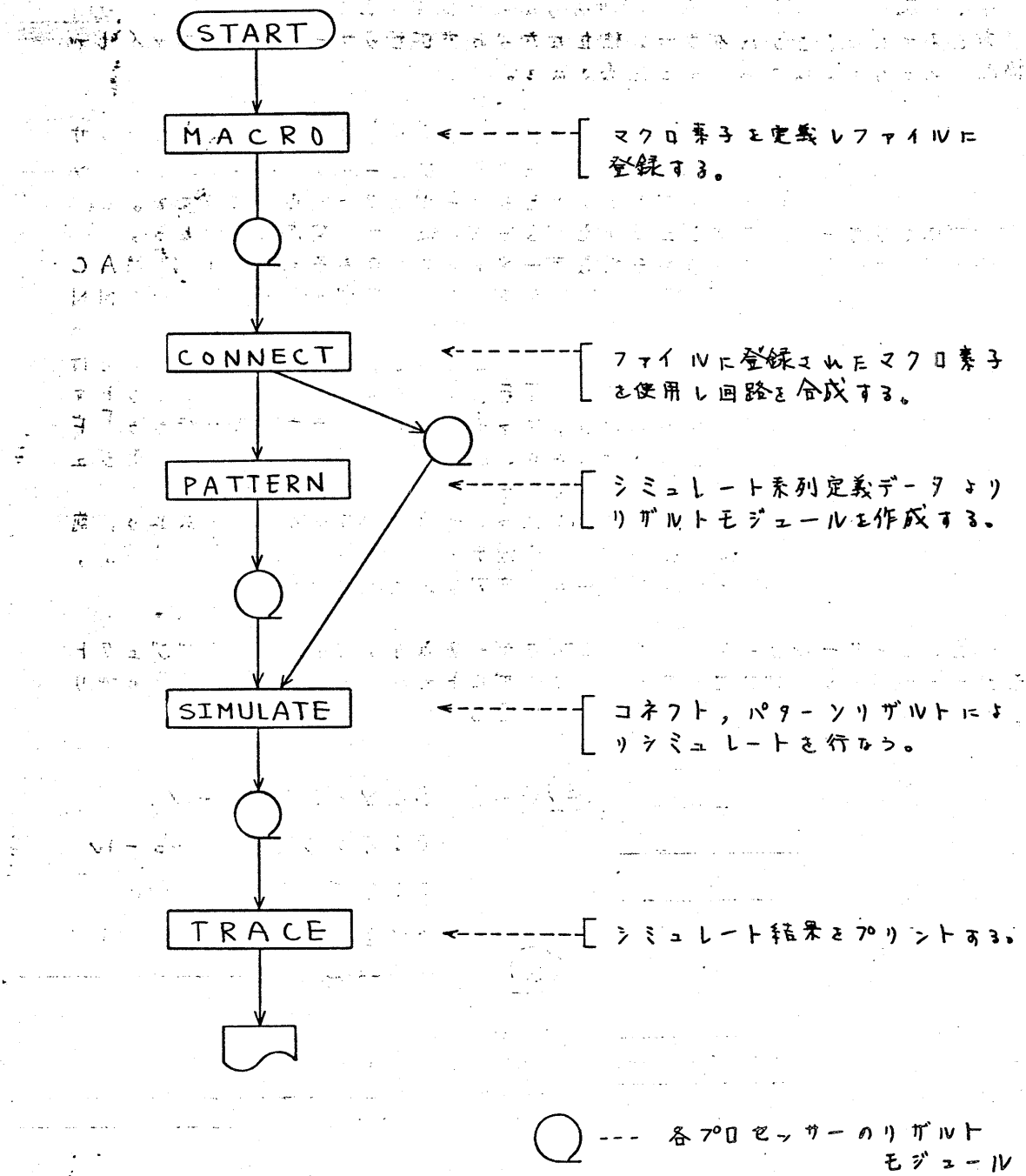

START
MACRO
マクロ素子を定義しファイルに登録する。
CONNECT
ファイルに登録されたマクロ素子を使用し回路を合成する。
PATTERN
シミュレート系列定義データよりリザルトモジュールを作成する。
SIMULATE
コネクト，パターンリザルトによりシミュレートを行なう。
TRACE
シミュレート結果をプリントする。
--- 各プロセッサーのリザルトモジュール

図 3. システムフロー

5．シミュレート手法

本システムでは，イベントデイレクテッド法（Event-Directed Method：以後ED法と略す。）[2]，タイムマッピング法（Time-Mapping Method：以後TM法と略す。）[3]を採用している。ここで遅れ時間を無視したシミュレート手法にはレベル割当法（以後LA法と略す。），ED法があるが，我々は両者の実験プログラムを作成し比較検討した結果，後者がシミュレート時間の面からより有効であると判断し本システムで採用した。以下にその要旨とTM法の手法を記す。

① LA法，ED法

LA法とは回路内の全素子に外部入力が最も低く，出力に向うにつれ高くなるようにレベル値を与え，このレベル値の低順に全素子の論理値を計算する手法である。すなわちレベル値は外部入力を1，内部素子のそれを「その素子の入力レベルの最大値＋1」と定義する。したがって当然の事ながらフィードバックループ（Feed Back Loop：以後ループと略す。）の存在する場合，レベル付けが困難になる欠点を持つ。

ED法とは，ある状態で安定している回路に新シミュレート系列を与えたとき，各素子の論理値変化をイベントと定義しそれを外部入力から外部出力へ伝搬させ，全素子の安定状態を求める手法である。したがってレベル付けの必要がなく，ループの有無にかかわらずシミュレートが可能である。

② ED法の有効性

ED法，LA法の実験式を求め両者を比較する。

LA法でのシミュレート時間は与えられた回路により一義的に定まり，シミュレート前の回路状態，入力系列には依存しない。いま一素子あたりの論理値計算時間をt_g，素子数をG，各系列のシミュレートのための始終処理時間をt_{LC}とすると一系列あたりのシミュレート時間T_{LA}はGに比例し，

$$T_{LA} \simeq t_g \cdot G + t_{LC} \qquad (1)$$

ここで実験プログラムでの60～200素子の回路の結果（図4）よりt_g，t_{LC}を求め代入すると

$$T_{LA} \simeq 0.3\,G + 19.5 \quad (m\,sec) \qquad (2)$$

ED法でのシミュレート時間は回路状態，シミュレート系列に依存し，決定要素として

i) シミュレート系列を与えてから回路が安定する迄に発生するイベント数
ii) イベントスタックのスワップイン，アウトの繰返数
iii) 素子数（始終処理部が依存する。）

を考えて次式を得る。

$$T_{ED} \simeq t_e \cdot E + t_f \cdot G^S + t_{EC} \qquad (3)$$

ただし t_e：一イベントの処理時間
E：一系列あたりのイベント数

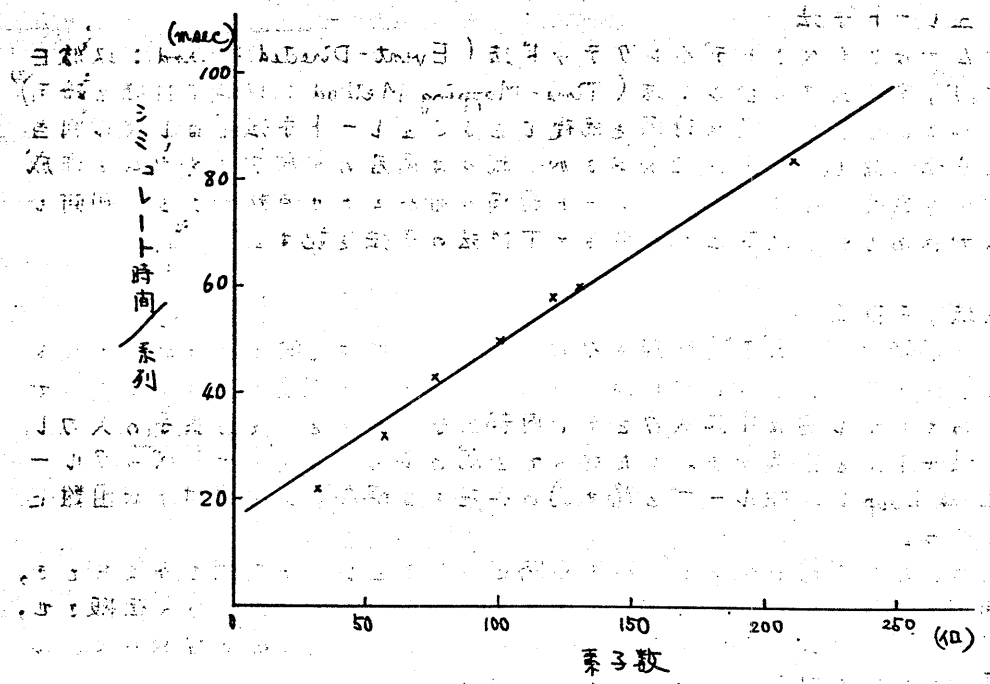


図4. LA法の実行結果

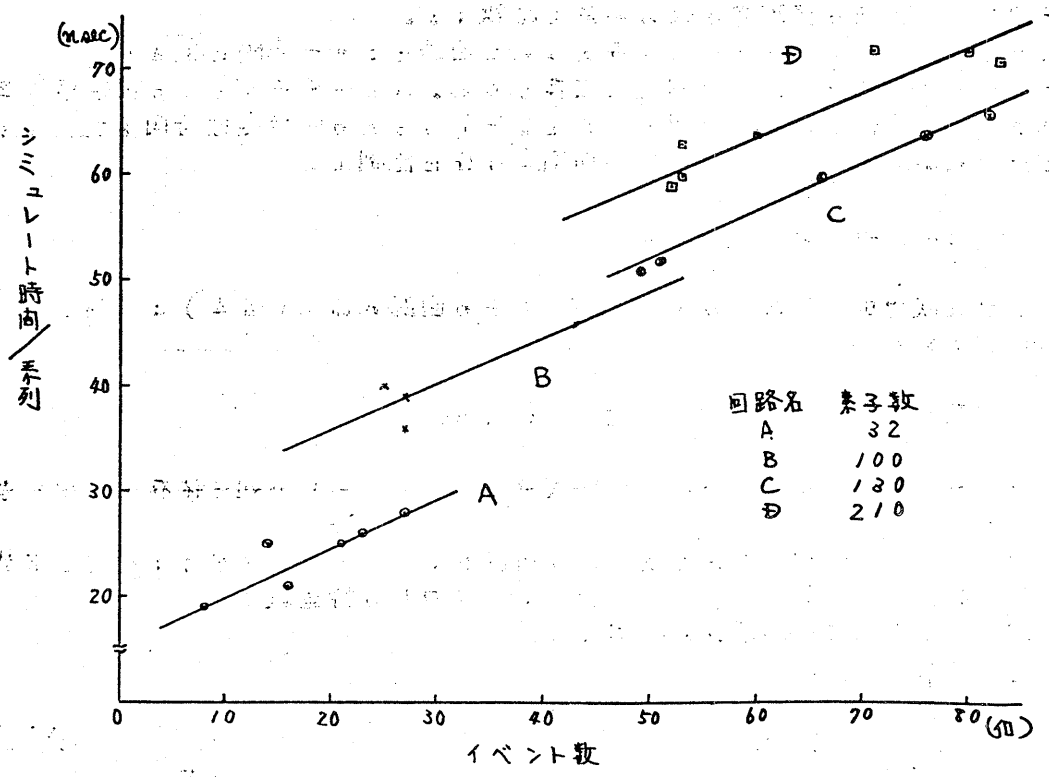


図5. ED法の実行結果

t_f：始終処理の一素子あたりの時間
G：素子数
s：始終処理における素子依存性
t_{EC}：$t_f \cdot G^s$ 以外の始終処理時間

実験プログラムでの結果をイベント数対シミュレート時間特性として図5に示す。これより t_e, t_f, s, t_{EC} を近似的に求め (3) 式に代入すると

$$T_{ED} \simeq 0.46E + 4.1\,G^{0.41} \quad (msec) \qquad (4)$$

ここで両者を比較するために全素子数に対するイベント数の割合を α (＝妻) とし，$T_{LA} = T_{ED}$ の場合の α を G の関数として求めると

$$\alpha = \frac{t_a}{t_e} - \frac{t_f}{t_e} \cdot G^{s-1} + \frac{(t_{LC} - t_{EC})}{t_e} \cdot G^{-1} \qquad (5)$$

各実験値を代入して

$$\alpha = 0.65 - 8.9\,G^{-0.59} + 42\,G^{-1} \quad (msec) \qquad (6)$$

となる。また

$$T_{ED} = 0.5\,T_{LA} \qquad (7)$$

の場合の α と (6) 式の α の特性曲線を図6に示す。同図より中，大規模回路に関して次の事が理解される。

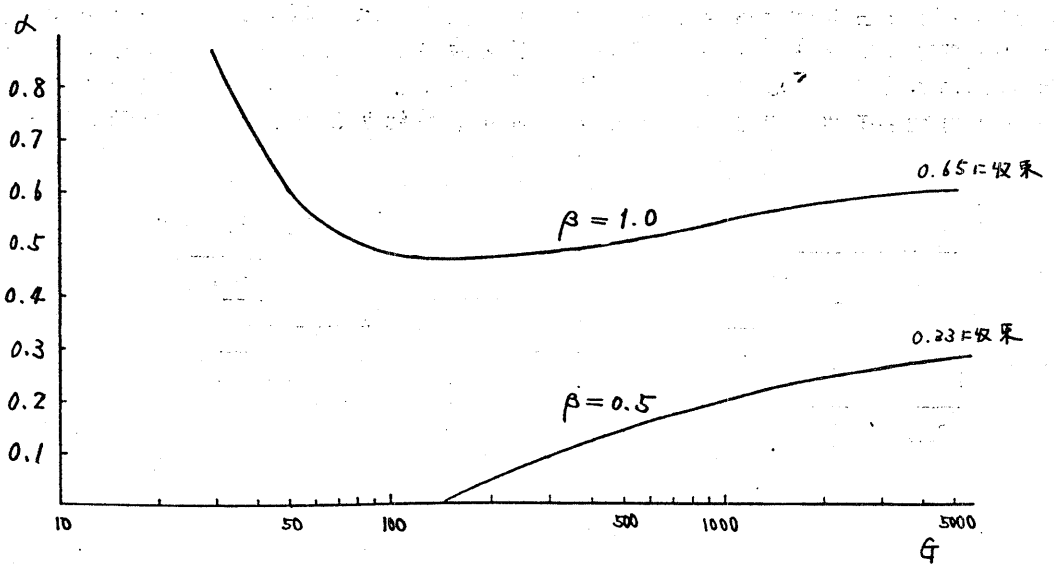


図6. α特性曲線

i) $T_{ED} = I_{LA}$ となるのは $\alpha \simeq 0.6$，すなわち全素子数に対するイベントの割合が60％以下であればED法が有効である。

ii) $T_{ED} = 0.5\ T_{LA}$ となるのは $\alpha \simeq 0.3$，すなわち全素子数に対するイベントの割合が30％以下であればED法はLA法の倍の速さを持つ。

一般には大規模回路では $\alpha < 0.1$ 程度であるためED法がLA法の数倍の速さを持つ事が判明し，我々は遅れ時間を無視したシミュレート手法にED法を採用した。

③ TM法

遅れ時間を含むシミュレート手法にはTM法を採用した。遅れ時間要素として最小ターンオン，オフ時間（T_{on}, T_{off}）とそれらのバラツキによる不確定時間（$T_{on\text{-}A}$, $T_{off\text{-}A}$）を考慮し，これらを各素子独立に定義する事により正確なシミュレートを可能にしている。しかし，これら四要素を独立に与えると図7に示すように，幅の短かい入力パルスに対し出力が応答しない場合が生じる。これをシミュレータの見地に立てば「既にスケジュールされているイベントの消去」を意味し，正しくシミュレートするにはイベントの消去手法を確立する必要がある。そこで各イベントに素子リンク，時刻リンクを設け，これらをたどる事による追加，消去の規則を確立した。図8にデータ構造を示す。素子リンクとは同一素子のイベントを時刻の早順にリンクしたものであり，時刻リンクとは同一時刻に活性化するイベントをリンクしたものである。

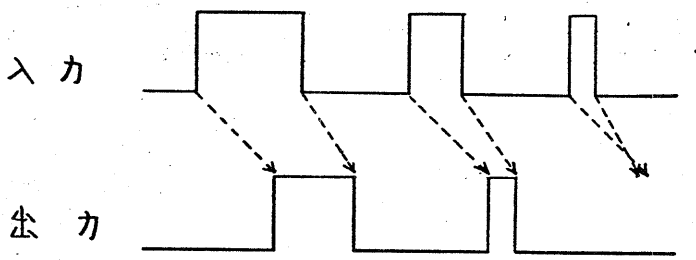


図7. パルス応答例

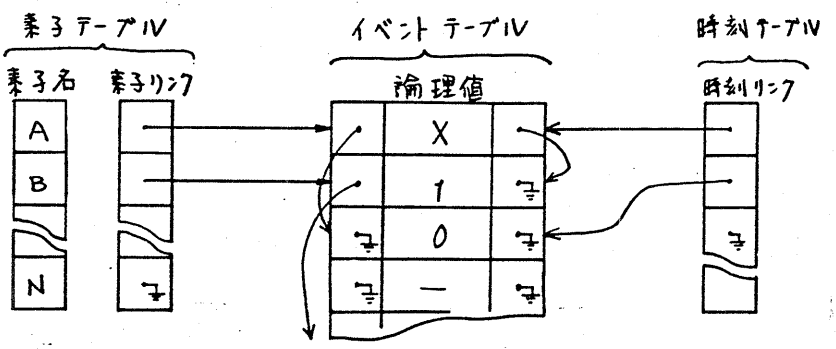


図8 データ構造

「イベント追加消去規則」

同一素子に対し複数個のイベントが存在するとき，素子リンク中の各イベントはスケジュールする時刻により順序づけられリンクされているものと仮定する。このイベントが既にスケジュールされている素子に新たなイベントを追加する場合，以下の規則に基づき処理する。

i) 素子リンクの最後に追加の場合は単純追加を行なう。

ii) 素子リンクの途中に挿入の場合は，

イ) 追加イベントの論理値が0，1のとき，以後に続くイベントを消去し追加イベントは直前のイベントの論理値と異なるときに行なう。

ロ) 追加イベントの論理値がXのとき，

a) 直前のイベントの論理値もXであれば以後に続くイベントを消去し，追加は行なわない。

b) 直前のイベントの論理値が0，1であれば，直後のイベントの論理値をXに変更しそれ以後のイベントを消去する。

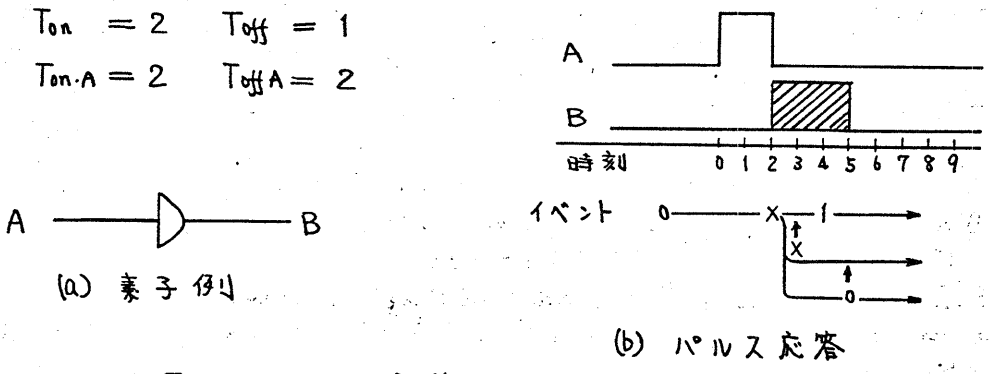


図9. パルス応答例

たとえば図9(a)の単純ゲートに0時刻で(b)に示す入力パルスを与える場合を考える。$T_{on} = 2$，$T_{on\text{-}A} = 2$ だから2，4時刻にそれぞれX，1がスケジュールされる。さて2時刻で入力パルスは0となるため T_{off}，$T_{off\text{-}A}$ に従ってX，1をそれぞれ3，5時刻にスケジュールするのだが，まずXは前述の規則(ii)・(ロ)・(a)に相当し，1のイベントが消去される。次に0だがこれは(i)に相当し単純に追加される。

6．おわりに

本システムはUNIVAC-1106を主計算機（Host Computer）とし，言語はFORTRANV，ステップ数は20K，コアーサイズ140KWである。いくつかの実行結果を表1に示す。Dはマイクロプロセッサーの主要部で1300素子であるが，シミュレート時間は短かく実用的である。これはED法を採用した事の正当性を示していると言える。

今後の課題としては，①パフォーマンスの向上，②並列シミュレーション，③故障シミュレーション等を考えている。

	素子数	系列数	手法	全実行時間	シミュレート時間/系列	備考
A	15	20	ED	37 秒	0.049 秒/系列	4相ダイナミック回路
		13	TM	80	0.252	
B	260	10	ED	80	0.070	8ビットALU
		10	TM	150	1.450	
C	150	10	ED	91	0.078	64ビットRAM
		10	TM	113	0.830	
D	1300	23	ED	687	0.021	マイクロプロセッサ

表1. 結果

(参考文献)

1) Y.T. Yen, "A Mathematical Model Characterizing Four-Phase MOS Circuits for Logic Simulation", IE^3 Trans. Computers VOL C-17 No.9 ('68) PP 822-826

2) E.B. Eichelberger, "Hazard Detection in Combinational and Sequential Switching Circuits", IBM J. March ('65) PP 90-99

3) S. G. Chappell 他, "Simulation of large asynchronous logic circuits using an ambiguous gate model", FJCC 1971 PP 651-661