

# 教育用論理回路シミュレータ LOCUS について†

吉岡 信夫†† 的場 進††

## Abstract

LOCUS is an interactive logic simulator that has been developed to assist the education of logic circuit.

It can be concurrently used by several students. Each student describes logic circuit that he has designed, exercises it by commands, traces nodal values that are printed out at every clock time. If necessary, he can modify the circuit at any time.

Interactive conversation between LOCUS and terminals, and extensive error diagnostics make LOCUS a very useful tool in the education.

## 1. まえがき

情報処理教育への補助に電子計算機を用いるのには種々の方法があると考えられる。一般的な CAI (Computer Aided Instruction) システムをこの目的に使用することもできるが、CAI システムの開発、試験、保守はかなり大がかりなものとなり、個々の分野の教育にちょうど合うシステムを作るには適当でないところもあると考えられる。さらに大学における教育を対象とすれば、一般的な CAI システムはそのステップが細かすぎるきらいがあり、また学生が自ら積極的に取り組んでいくのにも適していないと考えられる。これにたいし、個々の分野にちょうど適合した教育用ソフトウェアを開発し、種々の具体的分野での実用的なソフトウェアを積み上げていき、全体としてすぐれた教育用システムを作る方法が考えられる。本稿ではこの方法における具体的な分野でのソフトウェアの 1 つとして作成した教育用の論理回路シミュレータ LOCUS (Logic Circuit Simulator)<sup>1),2)</sup>について述べる。

LOCUS は、学生が構成した小規模な同期式論理回路を学生自身が動作させて試験し、修正するための道具として設計されており、実用的な論理回路の設計に用いられるシミュレータ<sup>3)</sup>とは異なった回路規模、使用者を対象としている。その主な特長は多端末にたい

してサービスできることと、インタラクティブな動作を行なうことにある。これらの機能により教育用の論理回路シミュレータとしての LOCUS は、多人数を対象として個別教育が行なえると同時に、学生が会話形式で論理回路のシミュレーションを実行し、その結果によりすぐに回路の組立て、修正ができるという特長を持つ。LOCUS は PDP-11/20 用に作成したものであり、現在は 3 端末からの同時使用が可能となっている。以下にその利用方法、構成、特長などについて述べる。

## 2. LOCUS による論理回路のシミュレーション

LOCUS は論理素子の組合せにより構成されている論理回路をすこし表現を変えて、簡単に会話形式でシミュレートするために作成されたものである。以下に LOCUS との会話方式、回路記述やコマンドの形式および機能を述べる。

### 2.1 LOCUS との会話

LOCUS がある端末から起動されるとその端末にメッセージが出力される。これ以後各端末から LOCUS への 1 行分の入力(以下単に入力と呼ぶ)にたいして、その結果が LOCUS からそれぞれの端末へ出力(以下単に出力と呼ぶ)されるという会話形式でシミュレーションが行なえる。入力が受付られる状態にはモード 0 とモード 1 の 2 つがあり、モード 0 では回路記述とそれに関連したコマンドが受け付けられ、モード 1 ではシミュレーションの実行に関連したコマンドが受

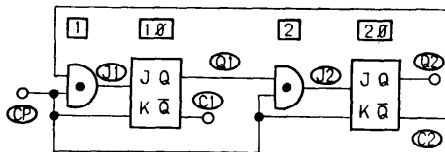
† LOCUS, Simulator for logic circuit education, by Nobuo Yoshioka, Susum Matoba (Faculty of Engineering Science, Osaka University)

†† 大阪大学基礎工学部情報工学科

付けられる。LOCUS がモード 0 かモード 1 かはそれぞれ \* または # の出力によって示される。このようにモードを分けたので回路の不用意な破壊が防がれており、誤りの検出も容易になっている。

LOCUS への端末からの入力行単位に受け付けられる。その各行はスラッシュ (/) で区切られたフィールドから成っており、それはさらにコンマ (,) またはセミコロン (;) で区切られたサブフィールドから成っている。そして各行の終りは CR (carriage return) で示される。なお、LOCUS がある行を受け付ける以前 (CR を押す以前) には、特殊キイを用いてその行全体またはその行の 1 文字を取消することができるので、行単位の入力の訂正は簡単に行なえる。

入力の各行は、論理回路の構成を素子単位に表現する回路記述、または論理回路のシミュレーションに関する操作を示すコマンド、のいずれかであり、その形式、機能はつぎに述べるようになっている。なお以下では例として図 1 に示す 3 進カウンタおよびそれをシミュレートしたときのリスト (図 2) を用いる。



□ : 素子番号      ○ : 端子名

図 1 3 進カウンタ

Fig. 1 Ternary counter

2.2 回路記述 (図 2②)

論理回路を表現する方式として LOCUS では節点表現 (すなわち 1 つの端子と直接接続されているすべての端子を 1 つの節点とみなして同一の名前をつける方式) を用いている。端末装置としてはテレタイプを利用しているため、この方式を用いてタイプインする入力の量の減少を図っている。

論理回路を記述するためにまず端子に英数字 2 文字以内の名前を付け、論理素子に数字 2 桁以内の素子番号を付ける (図 1)。LOCUS で表現できる論理素子は 9 種類とし、その素子記号を表 1 のように定める。(なお素子記号は最初の 2 文字に省略してもよい。) これらの素子番号、素子記号、端子名などを用いて表 1 の形式で素子単位で回路記述を入力する。この回路記述はモード 0 のときのみ受け付けられる。回路記述にたいし LOCUS は、すでに入力された素子が同じ素子番

① LOCUS V001A

```

*1/AND/CP,C2/J1
*2/AND/CP,Q1/J2
② *10/FJK/J1,CP/Q1,C1/0
   *20/FJK/J2,C1/Q2,C2/0
③ *ED/20/DF
   *20/FJK/J2,CP/Q2,C2/0
④ *NI/CP
⑤ *NO/J1,J2,Q1,Q2,C1,C2
⑥ *FC
⑦ #IN/CP;1111
⑧ #PR/4
   CP/Q1,Q2//J1,J2,Q1,Q2,C1,C2
   1 /0 ,0 //1 ,0 ,1 ,0 ,0 ,1
   1 /1 ,0 //1 ,1 ,0 ,1 ,1 ,0
   1 /0 ,1 //0 ,0 ,0 ,0 ,1 ,1
   1 /0 ,0 //1 ,0 ,1 ,0 ,0 ,1
⑨ #MO/J1,Q1
   J1,Q1
   1 ,1
⑩ #IZ
⑪ #SE/Q1;1
   #IN/CP;1
   #PR/1
   CP/Q1,Q2//J1,J2,Q1,Q2,C1,C2
   1 /1 ,0 //1 ,1 ,0 ,1 ,1 ,0
⑫ #NO/C1,C2/DF
⑬ #AU
   CP/Q1,Q2//J1,J2,Q1,Q2
   0 /0 ,0 //0 ,0 ,0 ,0
   1 /0 ,0 //1 ,0 ,1 ,0
   0 /1 ,0 //0 ,0 ,1 ,0
   1 /1 ,0 //1 ,1 ,0 ,1
   0 /0 ,1 //0 ,0 ,0 ,1
   1 /0 ,1 //0 ,0 ,0 ,0
   0 /1 ,1 //0 ,0 ,1 ,1
   1 /1 ,1 //0 ,1 ,0 ,0
⑭ #SA
   TYPE "P" OR SPACF
   1 /AN/CP,C2/J1
   2 /AN/CP,Q1/J2
   10/FJ/J1,CP/Q1,C1/0
   20/FJ/J2,CP/Q2,C2/0
   NI/CP
   NO/J1,J2,Q1,Q2
⑮ #RE
   LOCUS V001A

```

\*

— は LOCUS への入力を示す。

図 2 3 進カウンタのシミュレーションリスト

Fig. 2 Simulation list of ternary counter.

表1 素子記号および回路記述の形式

Table 1 Element symbol and circuit description format

(1) 組合せ論理素子
(i) AND, NAND, OR, NOR, XOR (exclusive OR), INV (inverter)
(ii) 素子番号/素子記号/入力端子1, 入力端子2, ... /出力端子
(2) フリップフロップ (FF)
(i) FRS (R-S FF), FJK (J-K FF), FD (D FF)
(ii) 素子番号/素子記号/R (または J, D) 入力端子, S (または K) 入力端子/Q 出力端子, $\bar{Q}$ 出力端子 /Q 端子初期値
(i) は素子記号, (ii) は回路記述の形式を示す。

号を持っているか, その出力端子が他の素子の出力端子に接続されているかなどを調べて, もしそうならエラーメッセージを出力する。

### 2.3 コマンド

LOCUS にコマンドを入力することにより, 論理回路のシミュレーションに必要な種々の操作を行なう。コマンドの一般的な形式はまずコマンド略号があり, 必要に応じて以後にパラメータを持ち, コマンドによっては最後のフィールドに DELETE (DE) スイッチを持つこともできるようになっている。各コマンドの名称, 形式は表2に示すとおりであるが以下にその機能について述べる。

#### (1) ED コマンド (図2③)

LOCUS をモード 0 にする。また DE スイッチが付いているときは指定した素子番号の素子を回路記述から削除。回路記述に誤りや変更のある場合にこのコマンドを用いて素子を削除してから正しい回路記述を入力すればよい。

#### (2) NI コマンド (図2④)

端子を外部入力端子に指定する。また DE スイッチが付いているときは, その指定を取消す。この端子は回路記述が終了した段階では他の素子の出力端子でないということにより簡単にわかるが, 誤り検出を容易にするためにこのコマンドにより指定する必要があるものとしている。

#### (3) NO コマンド (図2⑤, ⑫)

端子を出力観測端子に指定する。また DE スイッチが付いているときはその指定を取消す。この指定を受けた端子の値がシミュレーションの実行時に出力される。

#### (4) EC コマンド (図2⑥)

表2 コマンド一覧表

Table 2 Command list

コマンド名	機能	使用モード		形式
		0	1	
EDIT	モードの変更 {素子の取消}	○	○	ED{/素子番号, .../DE}
NODE FOR INPUT	外部入力端子の 指定(取消)	○	×	NI/外部入力端 子, .../DE}
NODE FOR OUTPUT	出力観測端子の 指定(取消)	○	○	NO/端子, ... /DE}
END OF CIRCUIT	回路記述の終了	○	×	EC
INPUT VALUE	入力値系列の指 定	×	○	IN/外部入力端 子: 入力値系列 /...
PROCESS	シミュレーショ ンの実行	×	○	PR/クロック数
MONITOR	端子値の観測	×	○	MO/端子, ...
INITIALIZE	回路の初期化	×	○	IZ
SET FF	FF の状態セッ ト	×	○	SE/Q 端子: 値 /...
AUTOMATIC	入出力状態遷移 表の自動作製	×	○	AU
SAVE	回路記述の出力	○	○	SA
RESTART	LOCUS の初期 化	○	○	RE

{ } は選択項目を示す。

回路記述の終了を LOCUS に示す。このコマンドを受け取ると, LOCUS は回路の中の各素子の論理演算を実行する順序を決定し, FF の出力端子の値を初期値にセットし, シミュレーションの実行に関連したコマンドを受付けるモード1になる。演算順序の決定はつぎの方法を用いている。まず外部入力端子と FF の出力端子の値を既知とする。このときすべての入力端子の値が既知であるような素子をさがし, その素子の演算順序を1としその素子の出力端子の値を既知とする。この操作をすべての素子の出力端子の値が既知となるまで繰返し, 素子の演算順序を決める。もし入力端子の値が1つでも既知でない素子が残っていれば, その素子番号を示すエラーメッセージを出力し, モード0にとどまる。

#### (5) IN コマンド (図2⑦)

シミュレーションの実行時に外部入力端子に加える論理値 (1または0) の系列を指定する。入力値系列は1行にタイプインできるものなら任意の長さのものを用いることができる。

#### (6) PR コマンド (図2⑧)

与えられたクロック数だけ論理回路のシミュレーションを実行する。LOCUS はまず端子名を

外部入力端子/FF の Q 端子//出力観測端子の形式で出力し, 1クロックごとにシミュレーション

を実行し、各端子の値を上述の形式で出力する。ただし FF の出力端子の値は / の左側ではクロックが加えられる以前の値を示し、右側では以後の値を示す。IN コマンドにより与えられた入力値の系列の長さより PR コマンドのクロック数が大きいときは、入力値系列の最後の値が引き続いてその外部入力端子に加えられているものとみなされる。したがって、値の変化しない端子には 1 クロック分の論理値を IN コマンドにより指定すればよく、入力値の指定が簡単になる。なお R-S FF において R および S 入力端子の値が同時に 1 になるときはシミュレーションの実行が切られ、エラーメッセージが出力される。

#### (7) MO コマンド (図 2⑨)

指定した端子のそのときの値を出力する。シミュレーションの途中で一時的に端子の値を観測するのに用いる。

#### (8) IZ コマンド (図 2⑩)

FF の状態 (FF の出力端子の値) を回路記述のときに指定した初期値にリセットし、外部入力端子に加えられている入力値系列を取消す。すなわち、回路記述が終了し EC コマンドの実行を終ったときと同じ状態にする。

#### (9) SE コマンド (図 2⑪)

FF の Q 端子を指定された値にする。(Q 端子はその値の否定の値にセットされる。)このコマンドにより FF の状態を任意にセットできる。

#### (10) AU コマンド (図 2⑬)

入出力状態遷移表を自動的に作製する。LOCUS は外部入力端子、FF の Q 端子の値を 1 行に並べて左方が下位の桁である 2 進数とみなしたときその値が 0 であるように端子の値をセットし、1 クロック分のシミュレーションを実行し、その結果を PR コマンドによる出力と同じ形式で出力する。つきにこの 2 進数を 1 だけ増加させた値になるように端子の値をセットし、この操作を繰り返す。たとえば図 2⑬ において外部入力端子 CP、FF の Q 端子 Q1、Q2 の値をそれぞれ (0, 0, 0) にセットしてシミュレーションし、つきに (1, 0, 0) にセットしシミュレーションする。この 2 進数の各桁がすべて 1 になるまで行なうので、すべての外部入力端子の値および FF の状態の組合せについてそのときの出力観測端子の値が表として出力される。FF の Q 端子を NO コマンドで出力観測端子に指定すれば、この表は入出力状態遷移表になる。なおこの表は組合せ論理回路にたいしては真理値表と

なっている。

#### (11) SA コマンド (図 2⑭)

これまで LOCUS に入力された回路記述、外部入力端子指定および出力観測端子指定を出力する。その形式は LOCUS への入力に用いられる形式と同じなので、紙テープへ出力すれば回路を保存できる。その後ふたたびその回路についてシミュレーションするときはこの紙テープを入力すればよく、簡単に再試行ができる。また回路をいくつかの部分回路に分けてくわしくシミュレーションし、それぞれの部分回路を紙テープに保存しておき、のちに全体をつないで動作を調べることもできる。

#### (12) RE コマンド (図 2⑮)

これまで LOCUS に入力されたすべての回路記述、内部の表などを消す。LOCUS は完全な初期状態に戻り、ふたたび新しい回路を受付ける。

### 3. LOCUS の構成と内部における処理

#### 3.1 構成

会話形式で使用するためと回路表現やコマンドの誤り検出を容易にするために LOCUS ではインタプリタ方式を用いており、端末から入力した 1 行のコマンド (回路記述を含む) をただちに解釈してシミュレーションに必要な各種の表を作成し、つきにそれらの表を解釈、実行して端末に結果を出力するようになっている。

使用した PDP-11/20 システムのハードウェアのうち LOCUS に直接関連した部分は CPU、24kW コアメモリ (16ビット/W)、テレタイプ 3 台、1.2MW ディスク装置から構成されている。ここでディスク装置は、LOCUS 自身を格納するために用いられており、オペレーティングシステムにたいして RUN コマンドを入力すればただちに LOCUS を利用することができる状態になる。

LOCUS それ自身は、PDP-11/20 のアセンブリ言語 (PAL-11 R) で書かれている数個のモジュールから構成されており、そのおもなモジュールのメモリ内での配置は図 3 に示すようになっている。LOCUS を利用する場合はすべてのモジュールはメモリ内に常駐している。各端末から共通に使用するプログラム領域と共通データ領域の大きさは合計約 5k ワードであり、各端末専用のデータ領域の大きさは各 2k ワードである。このとき各端末から取扱うことのできる論理回路は 60 素子、120 端子程度の大きさのものである。

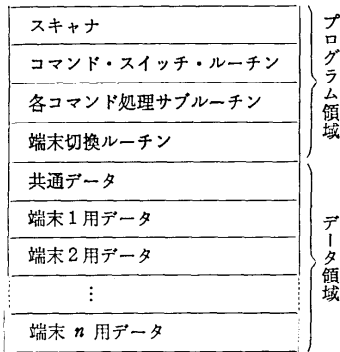


図3 LOCUS モジュールのメモリ配置  
Fig. 3 LOCUS modules in memory

現在3端末しかないので LOCUS 全体の大きさは 11 k ワードである。なお端末装置およびメモリを適当に増設すれば利用可能な端末数を増すことができる。つぎに各モジュールの機能、構造およびそこにおける処理などを入力の解釈、処理関係、回路の内部表現、シミュレーションの実行、端末切替に分けてすこしくわしく述べる。

3.2 入力の解釈、処理

端末から入力された文字はまずスキャナによって調べられ、ついでコマンド・スイッチ・ルーチンによってどのコマンドか判定され、さらに各コマンドに応じてそれぞれのコマンド処理サブルーチンによって処理される。各ルーチンの機能はつぎのようになっている。

スキャナは端末から入力される文字を順次バッファに入れて CR (carriage return) までの 1 行をたくわえてから、その 1 行を左からスキャンしてデリミタ (/) で区切られた各フィールドの要素およびフィールドの中のデリミタ ( , と ; ) で区切られたサブフィールドの要素を取出す。そしてこの 1 行が表 1、表 2 に示す回路記述あるいはコマンドに共通な形式に合っているかどうかを調べる。

コマンド・スイッチ・ルーチンはスキャナによって取出された行の最初のフィールドを調べてそれが数字であればその行を回路記述とみなしその処理ルーチンに分岐させ、それが英字であればその行をコマンドとみなしどのコマンドであるかを調べて適切なコマンド処理サブルーチンへ分岐させる。

コマンド処理サブルーチンは 2 つのフェイズからなっていて、入力はずまずフェイズ 0 で処理される。フェイズ 0 ではその入力の意味を調べ、またすでに入力された他のコマンドとの関係も調べる。そして入力が

正しいコマンドであるかどうかを判定し、もし正しければフェイズ 1 に進む。そこでは各種の表への書込み、変更を行ない、さらにそれらの表を解釈してシミュレーションを実行する。このように 2 つのフェイズに分かれていて、コマンドに誤りがあるとフェイズ 1 に進まず表も変更されないで、誤りからの回復が容易となり、これ以降のコマンドの実行に影響を与えることもない。

3.3 回路の内部表現

回路記述およびそれに関連したコマンドなどにより入力された論理回路は LOCUS 内部では図 4 に示す 3 個の表に変換して表現され、各端末用のデータ領域内にたくわえられる。その表のうち素子に関する表は、各列が固定長の素子番号表 (ELT) と、各列が可変長の素子接続表 (ENT) に分けられている。

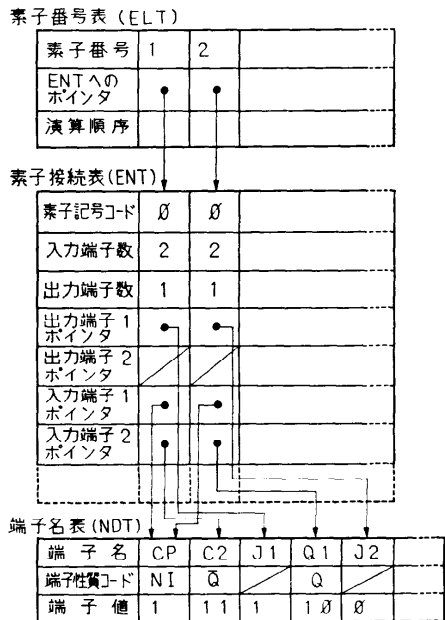


図4 回路の内部表現

Fig. 4 Internal representation of logic circuit

素子番号表 (ELT) の各列は 1 つの素子に対応しており、ENT へのポインタによって ENT の 1 つの列とも対応している。演算順序の行は回路記述が終了したときに書込まれる行で、この素子のつぎに演算されるべき素子を指している。

素子接続表 (ENT) は 1 つの素子に対応してその種類のコード、入出力端子の端子名を指すポインタなど

からなっている。素子記号コードの行は各素子記号に対応した数 (AND 素子ならば 0) が入っている。入力端子の数は組合せ論理素子 (INV を除く) にたいして 2 以上いくらかでもよいので、この表の入力端子ポインタの行の数が列によって異なり、NET の各列は可変長となる。

端子名表 (NDT) の各列は 1 つの端子に対応している。この各列は ENT の中の入出力端子ポインタによって指されており、これらのポインタが同一の列を指すことによりそれらの素子の入出力端子が接続されていることを示している。また端子性質コードは回路記述などにより指定される端子の性質 (NI は外部入力端子、Q、 $\bar{Q}$  は FF の Q 端子、 $\bar{Q}$  端子を示す) をコード化した数であり、誤りの検出などのために用いられる。端子値の行は現在の端子の値を示す。なお Q、 $\bar{Q}$  端子の場合この値の上の桁は現在の値を示し、下の桁は 1 クロック前の値を示す。

### 3.4 シミュレーションの実行

PR コマンドによりシミュレーションの実行が行なわれるが、そのときの内部の動作と各種の表への参照、書込みはつきのように行なわれる。なお図 4 の NDT の端子値は図 2⑧ の PR コマンドの第 1 クロック終了後の状態を示している。

まず IN コマンドで与えられた論理値を外部入力端子の NDT の端子値に書込み、FF の出力端子の初期値あるいは現在の値を 1 クロック前の値のところへ書込む。最初の演算すべき素子の ELT、ENT を見て各入力端子の値を NDT より取り出す。そして素子記号コードに応じた論理演算サブルーチンに飛び、出力値を得てその素子の出力端子の端子値 (FF の出力端子の場合は現在の値) に書込む。そしてつぎに演算すべき素子を ELT の演算順序の行により見つける。この操作をすべての素子について行なうことにより 1 クロック分のシミュレーションの実行を終了し、その結果を端末に出力する。

なおシミュレーションの実行速度は、1 クロックごとに行なわれる結果のテレタイプへの出力速度に比べて非常に速い。

### 3.5 端末の切換

多人数が同時に LOCUS を利用する場合に多端末間の切換のコントロールは端末切換ルーチンによって行なわれる。このルーチンは端末への入出力待ちのためのプログラム割込み、および端末への割当て時間 (0.1 秒) が経過したことを示すクロック割込みにより

起動される。

切換を簡単にするために、各端末用のデータ (各種の表および変数) は各端末に対応した一連の領域に格納されている。そして特定のレジスタを、各端末用のデータ領域の底を指すベースレジスタとして固定して割当てることにより、各端末用のデータを参照することができるようになっている。

端末切換ルーチンは現在活動中の端末のデータ領域を指すベースレジスタを含む 8 個のレジスタをその端末のデータ領域に退避し、つぎに活動する予定の端末のデータ領域に退避されているレジスタを復旧することにより端末を切換えている。なお割込みが起こったときにハードウェアによってプログラムカウンタなどが自動的に退避される機能<sup>4)</sup>を用いているので、このルーチンは非常に簡単なものとなっている。

## 4. LOCUS の特長

(1) インタラクティブに同期式論理回路のシミュレーションを行ない、その結果はただちに出力される。さらに回路記述形式や入力信号の指定が簡単であるので、回路の作製、修正、シミュレーションの実行が容易であり使いやすい。

(2) 種々の誤りを早く検出し、わかりやすく指摘する。まず入力された回路記述あるいはコマンドを 1 行ごとにその形式、意味について調べ、さらにすでにシステムに入力されたデータと相互にチェックしてその正しさを調べているので誤りの発見が早い。また誤りの個所をわかりやすいメッセージで的確に指摘するので誤りの発見、訂正が容易である。検出される誤りとそのときのエラーメッセージの例は付録に示されている。

付録のような誤りを検出するほかに、内部の表の大きさが限界をこえてしまうときも誤りとして検出し、LOCUS が破壊されないようにしている。そしてこれらの誤りが検出されたときはエラーメッセージを出力するとともにその入力を無視するのでそれ以降のコマンドの実行に影響を与えない。

(3) 多端末にたいしてサービスできるように作成されているので端末を適当に増設すれば多人数が同時に利用可能である。なお端末の切換による待ち時間の増加、切換のためのオーバーヘッドはテレタイプによる入出力時間にくらべて十分小さく問題とはならない。

(4) 以下に述べるような便利な機能を持つ。

(i) 真理値表または入出力状態遷移表を自動的に作成できる。この機能は小さな回路についてはかなり有用であると考えられる。

(ii) 順序回路の状態を任意にセットできる。

(iii) 回路記述を再入力可能な形式で紙テープに出力できる。これにより回路を数個の部分に分けて調べることや回路の保存、再試行が簡単にできる。

(5) 節点表現を用いて回路記述の入力量を減少させて誤りの発生を防いでいる。

## 5. むすび

LOCUS は教育用の論理回路シミュレータとして種々の特長を持ち、実用上は十分な機能を備えている。今後さらに追加していくのが望ましい機能としてはつぎのようなものが考えられる。

(1) 数個の素子を結合したブロックを1つの単位として回路を組み立てることができる。

(2) 回路記述やブロックの定義、シミュレーションの結果などをディスクファイルに出し入れできる。

(3) ディスプレイを通じて入出力を行ない接続状態を見て確かめながらシミュレートする。これはテレタイプを用いて入力を行なうと回路の接続誤りが生じたときその発見が容易でない場合があるのでそれを防ぐためである。

(4) CAI 的機能の追加、すなわち問題をシステムから与え、論理回路の構成やシミュレーションの結果を調べて端末に知らせるとともに端末の記録をとって使用者のふるまいの統計分析を行ない教育効果の向上をねらう。

なお機能をあまり拡張しすぎると、システム自身が大きなメモリを占有し端末用のデータ領域が小さくなって取り扱える回路が小さくなってしまふ。さらに使用者がシステムを使いこなすのにも時間がかかりすぎるので、適切な機能を選択することが必要であると考えられる。

最後に本システム作成に際して種々御助力いただいた大阪大学基礎工学部嵩研究室の方々に深謝する。

### 付録 誤りとエラーメッセージの例

— は LOCUS への入力を示す。  
左端のフィールドを  $\emptyset$  としている。

(1) 文法誤り (syntax error).

```
*1/AND/A,C//C
SYNTAX ERR
ERR AT FIELD 3
*
```

(2) 回路記述誤り (illegal element)

```
*f/AND/A,B,C
ILL ELM
FIELD NUMBER ERR
*
```

(3) 同じ素子番号をもつ素子がある。

```
*1/AND/A,B/C
*1/AND/D,E/F
ILL ELM
ERR AT FIELD 0
SAME ELM 1
*
```

(4) 素子の出力端子同士が接続されている。

```
*1/AND/A,B/C
*2/AND/D,E/C
ILL ELM
ERR AT FIELD 3
SAME OUTPUT NODE C
*
```

(5) 演算順序の決らない素子がある。

```
*1/AND/A,B/B
*NI/A
*EC
LOOP CIRCUIT
NO VALUE ELM
1
*
```

(6) 外部入力端子に値が与えられていない。

```
*1/AND/A,B/C
*NI/A,B
*NO/C
*EC
#PR/1
NO VALUE INPUT NODE
A,B
#
```

(7) R-S FF の R 端子, S 端子が同時に 1 になる。

```
*1/FRS/R,S/Q1,Q2/1
*NI/R,S
*NO/Q1,Q2
*EC
#IN/R;1/S;1
#PR/1
R,S/Q1//Q1,Q2
UNSTABLE RS-FF 1
#
```

### 参考文献

- 1) 蔭, 吉岡, 的場: ミニコンによる教育用論理回路シミュレータ, 昭 47 電気関係学会関西連大, G7-14.
- 2) 吉岡, 的場: 多端末からアクセス可能な教育用

- 論理回路シミュレータ, 昭 47 情報処理学会大会予稿集, 72.
- 3) A. H. Boyce, et al.: Simulation of binary logic circuits by digital computers, The Marconi Review, Second quater, 1971.
- 4) PDP 11 Handbook: Digital equipment corporation, 1969.

(昭和 48 年 3 月 13 日受付)