

論理回路学習システムの開発*

堀内泰輔**・堀内征治***

Some Systems for Learning of Logical Circuit

Taisuke HORIUCHI and Seiji HORIUCHI

In education of information processing, especially for mechanical or electrical engineer, it is important to teach not only programming technique but also designing of digital circuit. From this point of view, we have developed some systems for experiment with digital circuit, in the past 10 years.

Firstly, orthodox logical circuit trainer was developed. This system is one-board type trainer, and user can make digital circuit using patch cords. Secondly, unit type logical trainer was made. And thirdly, we developed logical circuit simulator. This system is operated on a personal computer, and user uses a mouse to draw circuit diagram on the CRT display.

These systems have merits respectively, but it has been appreciated that the logical circuit simulator is the best system for students.

1. はじめに

情報処理教育はとかくソフトウェアに重点が置かれがちであるが、機械系・電気系の科にあっては、メカトロニクス教育に立脚したハードウェア教育も重視されなければならない。そのなかでも論理回路の学習は最も基本的なものであり、学生が早くからこの考え方と意義を知っておくことは重要である。

われわれはこの立場から、過去10年間に渡り、情報処理教育の一環としての論理回路の基礎実習を行ってきた。これに用いた学習システムはすべて自作であり、初期はICを用いてパッチコードにより論理回路を作成するハード的なものであったが、現在ではパソコンとマウスを用いて画面上で回路を描画してその実行をシミュレートするというソフト的なシステムに変わってきている。

本論では、これらの開発、実践の経緯を述べるとともに、学習の効果について評価することとした。

2. 論理回路学習システム開発の経緯

表1はこれまでに開発し学生の実習に供してきた本システム群の概要を示す。

* 昭和63年8月 全国高等専門学校情報処理研究協議会、情報処理教育研究発表会において発表
 ** 機械工学科 講師
 *** 機械工学科 助教授
 原稿受付 昭和63年9月30日

表中、(A)は一般に「ロジックトレーナ」と呼ばれている最もオーソドックスな論理回路実習システムであり、パッチコードを用いて任意の論理回路を構成し、実際に電圧を与えてLEDの点灯/消灯によって結果を知るものである。当時市販のものが1台十数万円以上もし、購入できたとしても個別実習は到底無理であったため、基本的なゲート(NOT, NAND, NOR)を搭載した簡易ロジックトレーナを8台自作し、1台当り1~2学生が占有できるようにシステムを構築した。

表1 システムの経緯

	システム名	開発時期	主要システム構成パーツ
(A)	一体型 ロジックトレーナ	1979	IC 電源内蔵本体 パッチコード
(B)	ユニット型 ロジックトレーナ	1985	IC 各ユニット群 電源 パッチコード
(C)	ロジック シミュレータ	1987	パソコン マウス シミュレートソフトウェア

(A)は学習システムとしては十分機能したが、複雑な回路になると配線状態が把握しにくくなり、誤配線をしやすい欠点が発見された。また、使用できるゲートの数が限定されるためあまり複雑なものは実習できないという悩みがあった。このことから、(B)のユニット型ロジックトレーナの開発となった(写真1, 2)。

これは、原理は(A)と同じであるが、各ゲートや入力、出力、電源などを完全に分離しそれぞれのユニットにまとめたものである。実験回路に応じて必要なユニットを選択し、それらを電源ラインを含めてパッチ配線することで、どんなに複雑な回路でも実験でき、配線も比較的容易であるメリットを持つ。表2には各ユニットの機能と構成を示す。なお、(A)では実習が面倒だったフリップフロップを専用のユニット(J/Kフリップフロップ)として新たに用意した。

(A)および(B)システムは機能的には十分であったが、パッチコードの抜き差しという物理的操作を伴うために、パッチコードの断線およびICやジャックの接触不良が増え、実習の進行に支障をきたすことが多くなってきた。

このことから、(C)の「ロジックシミュレータ」という全く新しい考え方によるシステムを開発した。これは、パソコンの画面上で論理回路のシミュレートを行うもので、マウスによる論理回路作図の部分と、入力端子に任意の論理値を入れ動作をシミュレートして結果を出力端子に表示する部分から成り立っている。このシステムは画面上で配線をするので、自分

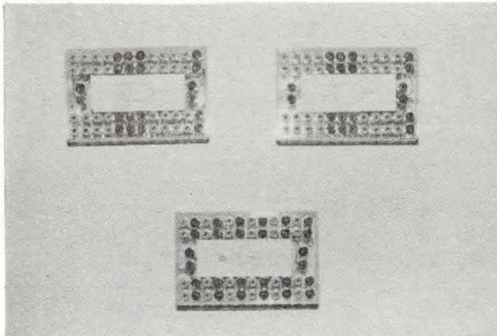


写真1 ユニットの一例



写真2 ユニット型ロジックトレーナの実習風景

表2 ユニット一覧表

ユニット名	1ユニット内の ゲート数など	1実験セットの ユニット数	備 考
NOTゲート	12ゲート	2	SN7404
4入力 NANDゲート	4 //	2	SN7420
4入力 NORゲート	4 //	2	SN7425
J-K フリップフロップ	4 //	1	SN74LS73
電源スイッチ	—	1	Vcc/GND端子兼用
入力スイッチ	8 個	1	
出力表示(LED)	8 //	1	
中継ボックス	12 //	1	配線分岐用

の描いた論理回路図と実際の配線が完全に対応し、実習の進行、学生の理解度とも3つのシステムのなかでは最良と思われる。以下には、このシステムについて概説したあと、これらのシステムの総合的な評価を行う。なお、写真3には(C)による実習風景を示す。

3. ロジックシミュレータによる論理回路学習システム

3-1 開発思想

授業という限られた時間内での有効な学習システムを構築する最大条件は、システム自体の使い勝手が良いことである。本システムでは、入力装置としてマウスを採用し、キーボードはほとんど不要となるよう設計した。また、ペーパーマニュアルを見ながらの操作をなくすために、機能やゲートをアイコン化し、画面だけを見ながら配線からシミュレートまでを行えるように配慮した。

なお、開発のターゲットマシンは、本校電子計算機センターにパソコン端末として設置されているFM16βFDⅡ(45台)であり、ソフトの開発にはC言語(Lattice C)を用いた。サイズはソースコードで約1200行、実行プログラムは38KBである。

3-2 機能の概要

配置可能な論理ゲートや入出力端子のシンボルを表3に示す。これらをマウスにより選択し、マウスカーソルを所定の位置に持って行ってボタンをクリックすることで、画面上の任意の場所にこれらを配置できる。

次に、これらのシンボルを配線することになるが、自分の描いた論理回路図にでき



写真3 ロジックシミュレータによる実習風景

表3 シンボル一覧表

ゲートの種類	入力数	ゲート図
NOT	1	
AND	2	
	3	
	4	
	2	
OR	3	
	4	
	2	
NAND	3	
	4	
	2	
NOR	3	
	4	
入力	0	
出力	1	

表4 配線形態の種類

種類	内容
	開始点のX座標と終了点のY座標の位置が一致するように入力端子を折る。この場合、配線が複雑になる。
例	
種類	内容
	開始点と終了点のX軸中点で直角に折れる。この場合、配線が複雑になる。下の例のように右から左に戻るような場合等に用いる。
例	
種類	内容
	開始点と終了点のY軸中点で直角に折れる。この場合、配線が複雑になる。下の例のように右から左に戻るような場合等に用いる。
例	

るだけ近づけるため、また配線を見やすくするために、表4のような4種類の配線形態を用意した。

また、配線の修正のために、ゲートの削除、配線の削除、画面全体の消去の各機能を設けた。ただし、システムを簡略にするため、ゲートの移動機能は省略した。このほか、回路図をファイルに保管し、任意の時点で呼び出すことができるようにした。以上の諸機能をすべてアイコン化したため、従来のソフトウェアに見られるようなキーボードからのコマンド入力は一切不要である。図1には描画中の画面の一例を示した。

次に回路のシミュレートであるが、これには実行のアイコンをクリック後、各入力端子に0または1の論理値をアイコン入力する方式を採った。全入力端子への入力が終了した時点でシミュレートを行い、論理値0、1をそれぞれ青、赤の2色に対応させて、配線部と出力端子に表示する。これにより学習者は出力のみならず、回路の各部での論理の状態を一見して理解できる。図2には実行の一例を示した。

3-3 システム概要

本システムは、マウス入力による回路の描画部分と、描画された回路のシミュレート部分に大別される。

3-3-1 マウス入力と回路図の描画

回路図の描画は、ゲートアイコンの選択と画面への配置を繰り返した後、各ゲートを結線するという手順になる。このうち、ゲートアイコンの選択では、アイコンとして表示できるゲートの個数が限られるため、2入力ゲートのみをアイコンとして配置し、アイコン選択直後ポップアップメニュー形式で3~4入力ゲートを表示し、最終的にアイコン選択を行うようにした。

次に画面へのゲートの配置であるが、描画の自由度を上げるため、指定枠内であれば任意の位置に配置できるようにした。また、各ゲートの結線の手続きでは、出力どうしは結線できないような論理チェックを施した。

最終的な回路図が完成した段階では、これをシミュレートできるように特定のデータ構造

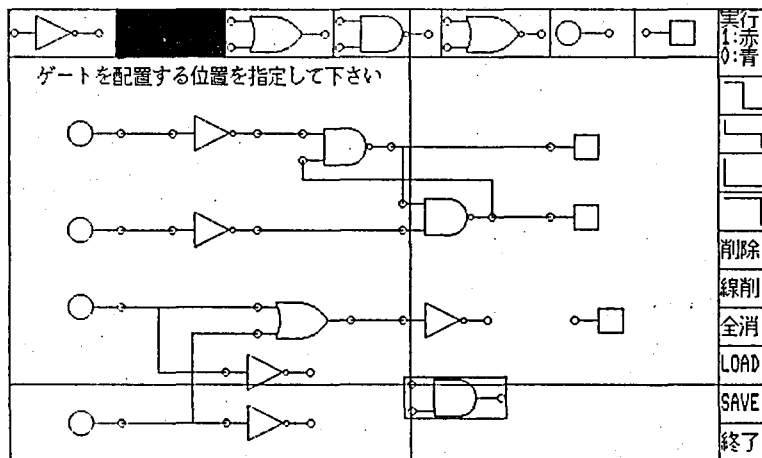


図1 描画の一例

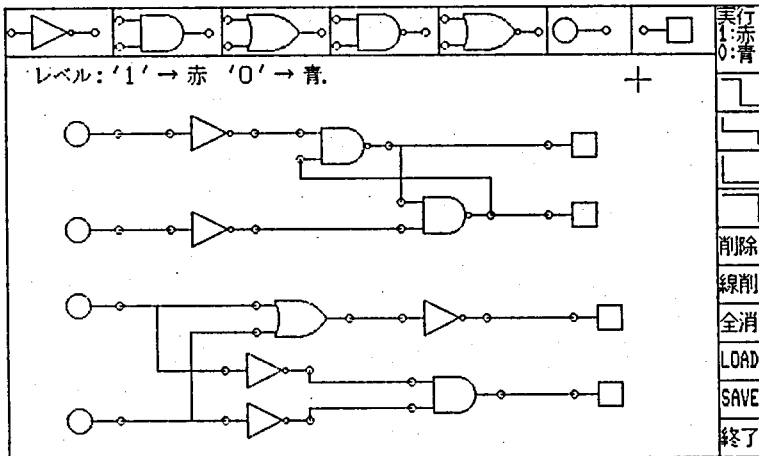


図2 図1の回路の実行例

da[i][0]	ゲートコード
da[i][1]	入力端子のX座標
da[i][2]	入力端子①のY座標
da[i][3]	入力端子②のY座標
da[i][4]	入力端子③のY座標
da[i][5]	入力端子④のY座標
da[i][6]	出力端子のX座標
da[i][7]	出力端子のY座標

(i=1, 総ゲート数)

(a) ゲートデータ配列

d1[i][0]	配線コード
d1[i][1]	始点のX座標
d1[i][2]	始点のY座標
d1[i][3]	終点のX座標
d1[i][4]	終点のY座標
d1[i][5]	始点のゲート→配列番号
d1[i][6]	終点のゲート→配列番号
d1[i][7]	折れ曲がる部分の座標

(i=1, 総ゲート数)

(b) 配線データ配列

図3 回路図データ配列

を用いて内部表現がなされなければならない。本システムでは、図3のようなデータ構造を用いた。これによれば、シミュレートのみならず、ゲートの削除、再配置、結線の変更なども可能となる。

3-3-2 描画した回路のシミュレート

シミュレートに先立ち、描画が完成した回路図中の各入力端子に、論理0または論理1のいずれかを入力する必要がある。これには2つのアイコンを用意し、マウスから入力するようにした。入力端子にはいずれかの論理値が必要となるが、入力端子が多い回路のとき、特定の入力のみを変更してシミュレートする機会が多いことを考慮して、前回の入力をそのまま用いることを指示するためのアイコンも用意した。全ての入力値が決定した時点で、回路

のシミュレートが行われる。

前述のゲートデータ配列ならびに配線データ配列をもとにシミュレートが行われるが、最初にこれら进行处理しやすいデータ構造に変換する。これをシミュレート用配列と称し、図4に示す。実際のシミュレートでは、この配列と各ゲートごとに用意された出力バッファが用いられる。

シミュレートに際しては、まず各入力端子の入力値をそれに接続されている各ゲートの入力端子に供給する。次に、当該ゲート固有の論理演算を行い、出力バッファに状態を記憶した後、次段のゲートにその出力を供給することを繰り返す。

一致回路や加算器など、入力のみにより出力が決定される組合せ回路の場合には、入力値がすべて決定したゲートから順次出力を求めていき、すべてのゲートのシミュレートが終了した時点で全体のシミュレートを終えることができる。これに対し、フリップフロップに代表される順序回路では、フィードバックが掛かっているため、通常、各ゲート当り1回のみのシミュレートでは出力が決定できない。このため、このようなシミュレートの終了はユーザがマウスを用いて明示的に指示するまでは、シミュレートを続行する仕様とした。なお、シミュレートの

g[i][0]	ゲートコード
g[i][1]	出力論理値(0, 1, 2) <2は未定>
g[i][2]	入力端子数(1~4)
g[i][3]	前段のゲート配列番号①
g[i][4]	前段のゲート配列番号②
g[i][5]	前段のゲート配列番号③
g[i][6]	前段のゲート配列番号④

(i=1, 総ゲート数)

図4 シミュレート配列

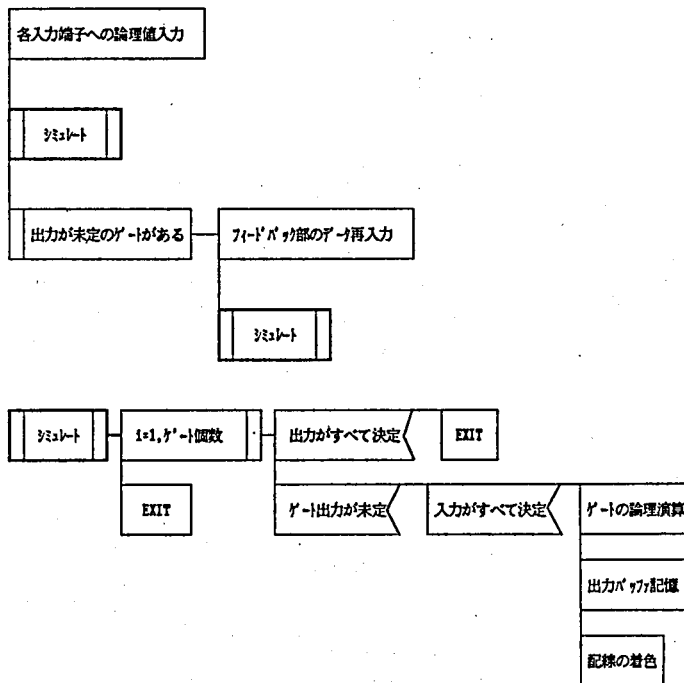


図5 シミュレートのアルゴリズム

アルゴリズムを図5のPADに示した。

シミュレートの経過ならびに結果は、画面上の各配線の色により表現することとし、論理1の場合は赤、論理0の場合は青とした。シミュレート開始時点の配線の色は黄色としたため、シミュレートが進んでいく様子が画面の配線の色の変化により認識できる。

4. 各システムの評価と課題

以上の各システムは開発時期の相違はあるものの、それぞれ特徴を持ったものである。表5は、これまでに実習を行った学生のレポート中の感想などをもとにして作成した評価表である。

この結果、ロジックシミュレータは操作性、実習事項の理解度などで最良と考えられる。ただし、実体（IC、スイッチ、LEDなど）が見えないため、実際の応用製品（パソコンなど）との関係がつかみにくく、机上の理論に留まりがちであるという欠点を持つ。

このことから、ロジックシミュレータとユニット型ロジックトレーナと組み合わせることにより両者の長所を最大に引き出せることになり、これが最良の利用方法と考えられる。

なお、現状のロジックシミュレータについて、ユーザからの要望が多かったのは、画面のハードコピー機能、ゲートの大きさの縮小機能、画面のスクロール（上下/左右）機能などのサポートである。これらのほか、任意のICのシミュレートをユーザレベルで増設可能にすること、XYプロッタに出力できること、タイミングチャートの自動作成等を今後の課題として、よりよい学習システムにして行きたい。

表5 各システムの評価

評価項目	(A)	(B)	(C)	評価項目	(A)	(B)	(C)
回路の拡張性	△	◎	◎	誤配線のしにくさ	△	○	◎
各ゲートの理解のしやすさ	○	○	◎	配線後の実験のしやすさ	○	○	◎
配線のしやすさ	△	○	◎	部品のメンテナンスの必要性	×	△	◎
配線チェックのしやすさ	△	○	◎	回路のリアルさ	◎	◎	△

(A)……………一体型ロジックトレーナ

(C)……………ロジックシミュレータ

(B)……………ユニット型ロジックトレーナ

5. おわりに

本論では、各種の自作した論理回路学習システムを紹介し、それぞれの特徴を持ちながらも、シミュレータ方式によるシステムが、学習効率からも有効であることを述べた。パソコンのコストが極度に低下している現在、従来のハード一辺倒の実習教材を見直すことも有効な教育改善となりうると思われる。

おわりに、シミュレータの開発に当たっては、本校機械工学科情報研究室に在籍した畑山昌也（現東京電機大学）、宮川良一（現富士通）、大池明、各君の努力が大きい。ここに深く感謝の意を表す。

参 考 文 献

- (1) デジタルICの基礎, 白土義男, 東京電機大学出版局, 1980
- (2) Lattice C Compiler ユーザーズマニュアル, ライフポート
- (3) FM C関数ライブラリ使用手引書, 富士通