

デジタル機器の論理設計検査プログラム

GPLS-I について*

荒 牧 達** 河 村 佳 洋***

Abstract

An application program named GPLS-I (for General Purpose Logic Simulator Version I) has been developed to check the validity of logical design while it is still on the drawing board. GPLS-I can execute logic simulation of (1) synchronous logic circuits, (2) asynchronous logic circuits, and (3) mixtures of both types of circuits when basic logical design is over (AND, OR level), or when the logic circuit is expressed in terms of actual logic elements such as Texas Instruments' J-K flip-flops in dual-in-line packages, or if the designer finds it useful, when the circuit is a mixture of various stages of logical design as is often found in the designer's memos. The input data for GPLS-I are the connections of logic elements and input signals applied to the logic circuit. From these data GPLS-I prints out the static time chart of logical 0 or logical 1 at the arbitrarily specified terminals of arbitrarily specified logic elements. The times at which logical signals are to be printed out can also be predetermined. Signal voltage levels are treated as either high (logical 1) or low (logical 0). GPLS-I is best suited for the design check of medium to small scale industrial digital systems. Several cases of actual design check have proved that GPLS-I is a very useful and convenient tool. In this paper the outline of GPLS-I is explained and an example of design check by means of GPLS-I is given. Also the algorithm and program system structure are described in relative detail.

要 約

中小型の工業計測器型のデジタル論理装置が、所期の論理動作を行なうかどうかを設計図上で検査するためのアプリケーションプログラム GPLS-I (General Purpose Logic Simulator Version I) を開発した。このプログラムは同期式論理素子、非同期式論理素子、もしくは両者が混在する論理回路の設計検査を、(1) 基本論理設計終了 (AND, OR 等を用いて論理的な考え方を表現した段階)、(2) 論理回路設計終了 (IC のような市販の素子を用いて具体的に回路を構成した段階)、または (3) 上記 (1)、(2) の中間的な段階 (設計者のメモ) の任意の状態で行なうものである。入力データとしては論理素子の相互接続関係を

回路図上から読み取って与えるほか、外部からの入力信号を指定する。出力としては論理素子の各端子における信号レベル (1 または 0) のスタティックなタイムチャートが得られる。出力を観察したい論理素子、その端子および時刻は任意に指定することができる。信号レベルはすべて 0 または 1 として扱われる。

GPLS-I は中小型の工業計測器型デジタル論理装置の設計検査に適しており、適当に使用すれば手軽、かつ、強力な設計上の武器となることが、実際使用例からも確かめられた。本報告では、まず、GPLS-I の概略を説明し、ついで実用例について述べる。アルゴリズムおよび I/O を中心としたプログラム構成についても、比較的詳細に論じた。

1. まえがき

デジタル機器を設計し、製造に着手するに際しては、はたして完成機が所期どおり動作するかどうかをチェックする苦労は、対象が比較的小型機の場合でも

* Logic Simulator GPLS-I, by Itaru Aramaki (Research Division, Sumitomo Electric Industries, Ltd.) and Yoshihiro Kawamura (IBM Japan, Ltd.)

** 住友電気工業株式会社

*** 日本アイビーエム株式会社

言語を絶するものがある。GPLS-I は「一品料理」的な多品種少量生産の工業計測器型デジタル機器の設計者が論理設計検査を行なう際に、手軽で便利な道具として、使用コンピュータもコンピュータプログラムの知識もなしに使用しうるように開発した論理シミュレータである。その最大の特徴は、市販の IC パッケージをもまじえた 11 種類の論理素子を任意に接続した回路の論理設計検査を、同期式、非同期式にかかわらず行ないうる点にある。

2. GPLS-I について

GPLS-I (General Purpose Logic Simulator Version I) とは同期式、非同期式、および両者が混在する論理回路の設計の検査を、(1) 基本論理設計終了、(2) 論理回路設計終了、および(3) (1), (2) が混在する状態の任意の段階において行なうもので、工業計測器型のデジタル論理装置が、所期の論理動作を行なうかどうかを設計図上で検査するためのアプリケーションプログラムである。GPLS-I で取り扱う論

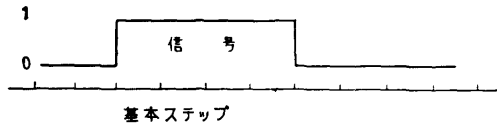
第1表 GPLS-I の対象となる論理素子

指定番号	コード	機能	端子番号	備考
12	SN 7470 N	J-K FLIP-FLOP		Texas Instruments 社製
13	SN 7472 N	J-K MASTER-SLAVE FLIP-FLOP		Texas Instruments 社製
24	DIF(+)	入力が0→1の変化をしたとき出力1		出力端子は 1. 2 は入力端子
25	DIF(-)	入力が1→0の変化をしたとき出力0		
26	DEL(n)	基本ステップ n 個分の遅延		
27	NÖR	$Z = \overline{A+B+\dots+S}$		出力端子は 1, 2~19 は入力端子 図において Z は端子 1 の出力
28	NAND	$Z = \overline{A \cdot B \cdot C \dots S}$		Z は端子 1 の出力 A, B, ..., S はそれぞれ端子 2, 3, ..., 19 の入力
29	ÖR	$Z = A+B+\dots+S$		
30	AND	$Z = A \cdot B \cdot C \dots S$		
31	NÖT	$Z = \overline{A}$		出力端子は 1, 2 は入力端子. 図において Z は端子 1 の出力, A は端子 2 の入力
32	RSFF	R-S FLIP-FLOP		セット入力は端子 3, 4, ..., 10 の入力のÖRとなる。また、リセット入力は端子 11, 12, ..., 18 の入力の ÖR となる

- (注) 1. 指定番号およびコードはコンピュータ入力データ作成時に使用する。
2. 論理は正論理 (Positive Logic) を用いている。
3. 使用しない端子の端子番号は無視する。

理素子を第1表に示す。GPLS-I は第1表中の素子がいかに混在した回路図でもシミュレートすることができる。第1表中 J-K flip-flop および J-K master-slave flip-flop は Texas Instruments 社の SN 7470N, および SN 7472 N と同じ機能と同じ端子配列を持っている。

GPLS-I においては外部よりの入力信号も、デジタル機器内部の信号も、出力信号も、すべて0または1 (logical 0 または logical 1) の矩形波であって、その時間ベースは GPLS-I 内部の基本ステップによっている。この基本ステップは任意に細かくできるので、非同期信号の位相も任意に細かく指定できる。



たとえば、上図の場合ならば、信号の長さは4 (units of simulator clock) である。

2.1 入力データ

GPLS-I を用いて論理設計検査を行なうためには、① プログラムコントロール、② 素子接続関係、③ プリントアウト、④ 外部入力信号 の4項目についてのデータをインプットしなければならない。以下に、これらの項目について第1図の回路図の論理検査

を行なう場合を例にとって説明する。まず、第1図のように論理素子、および外部入力端子に適当な番号をうつ。素子番号は各素子を区別するだけであるから、適当にうっておけばよい。ただし、各素子の端子(足)の番号は、第1表の端子番号に従う。

第1図をもとに入力データは、つぎのように作成される。

2.1.1 プログラム・コントロール

プログラムコントロールのうちの主要なものは、つぎの二つのコントロールである。

(1) 外部入力信号処理コントロール

「0」または「1」と指定する。

「0」と指定すると、外部入力信号が 150 ブロック* 以内であれば、基本ステップをどのようにでも細かくできる (最大 1,500 万ステップ)。

「1」と指定すると、外部入力信号の 0→1、もしくは 1→0 の変換点を 2,400 個までとりうるが、基本ステップも最大 2,400 ステップで制限される。

[*: 1ブロックとは、信号の山 (1)、または谷 (0) のことである。]

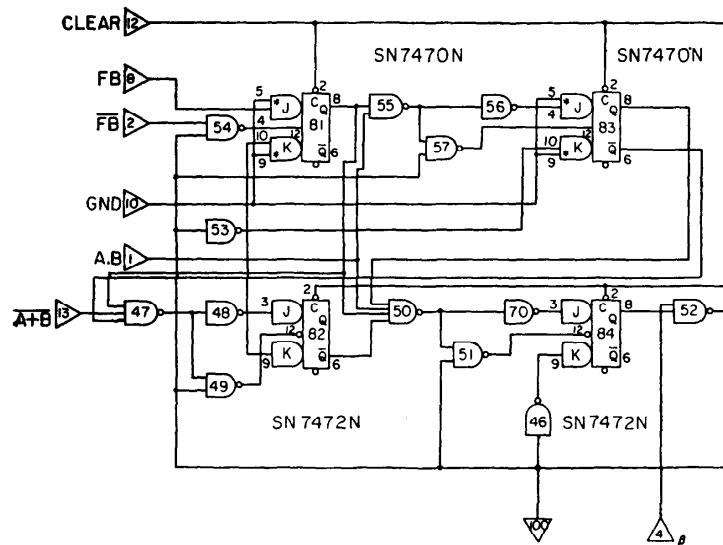
(2) チェックアウトコントロール

GPLS-I で論理演算を進める場合の計算内容の中間結果を打ち出すためのコントロールで、中間結果出しの詳しさに応じて6段階に指定しうる。

2.1.2 素子接続関係

対象となる回路図の論理素子の接続関係を指定する。回路図に現われる全論理素子の結線の行なわれている端子すべてについて、その端子から出ている線の行き先を指定する。この行き先が他の多くの素子にわたっている場合には、どの一つを選んでもよい。例として、第1図の回路の素子接続関係を指定する場合について説明する。

たとえば、素子 50 の端子 1 (出力端子) は、素子 70 の端子 2 および素子 51 の端子 2 にはいつている。これをデータシートに記入するとき、どちらか一方を書きおけばよい。



第1図 GPLS-I による設計検査の例

2.1.3 プリントアウト

(1) 信号波形をプリントアウトすべき素子, 端子番号指定信号波形を観察したい素子と, その端子番号を指定する.

(2) 基本ステップ数

GPLS-I で論理設計検査を行なう場合の回路の全動作時間に相当する.

(3) プリントアウトすべきステップの指定

基本ステップの何ステップ目で信号波形をプリント

アウトするかを指定する.

2.1.4 外部入力信号

設計検査を行なおうとしている回路の外部入力信号波形を数値的に表現する. Logical 1は「+」, logical 0は「-」で表現し, 信号の持続時間は, 基本ステップを単位に整数で表わす. また, 周期的に繰り返す信号に対しては「*」印を用い, 掛算の形で繰り返し回数を指定する. たとえば, 第1図の回路の外部入力端子

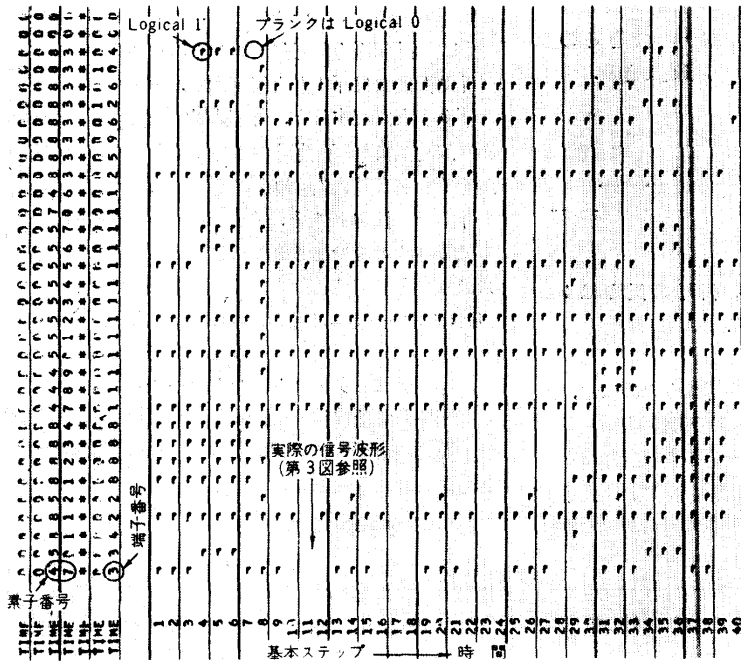
12 からの外部入力信号波形は, 第3図の上から4行目 (CLEAR) のようになるが, これを GPLS-I 式に表現すると, つぎようになる.

$$10-1*1+5-1*2+16-1*1+4-1*2+12-1+4-1+4-1+22/$$

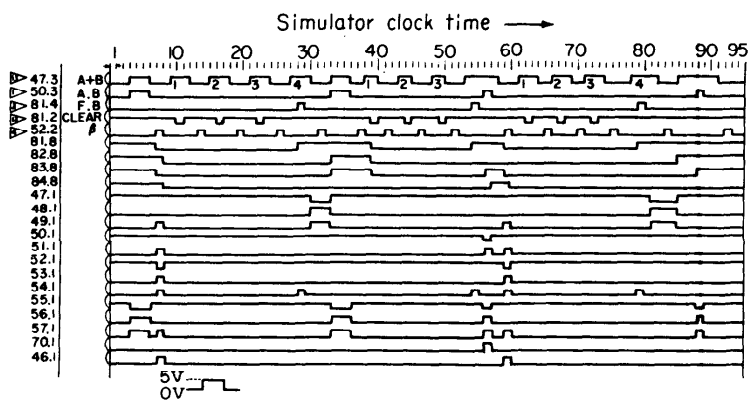
2.2 出力データ

第1図の論理回路の外部入力端子に第3図 A+B, A・B, FB, 等の外部入力信号を入れて GPLS-I によってタイムチャートを描かせると第2図のようになる. 第2図は第3図に示した基本ステップ95のうちの最初の40ステップの部分である.

出力データは, 60端子×40ステップを1単位としてプリントアウトされる. 端子数は1,500個, ステップ数は「外部入力信号処理コントロール」によって, 150ブロックもしくは2,400ステップまで延長可能である. 以下第2図を参照しながら出力データの見方を説明する. 第2図において0047*03, 0081*08等はそれぞれ素子47の端子3, 素子81の端子8等を示している. 左端の1, 2, 3……等の数字は基本ステップ番号, すなわち, 時間の進行を示している. 黒く「┆」印がでているところは logical 1, ブランクのところは logical 0を表



第2図 GPLS-Iによるタイムチャート作成例
第1図の回路に対応している



第3図 第1図の論理回路各部の電圧波形

わす。したがって、たとえば素子 47 の端子 8 の出力電圧波形はステップ 1, 2, 3 までは logical 1, ステップ 4, 5, 6 で logical 0, ステップ 7, 8, 9 で再び logical 1 等となっていることがわかる。

2.3 制限事項とエラーメッセージ

GPLS-I の主要な制限事項とエラーメッセージを以下に列挙しておく。

2.3.1 制限事項

- (1) 回路図を構成する論理素子の総数は最大 500 個とする。
- (2) 外部入力端子の総数は最大 100 個とする。
- (3) すべての素子の出力端子のファンアウト数は 10 とする。
- (4) 論理素子につける番号は 4 桁以内の正整数値

とする。

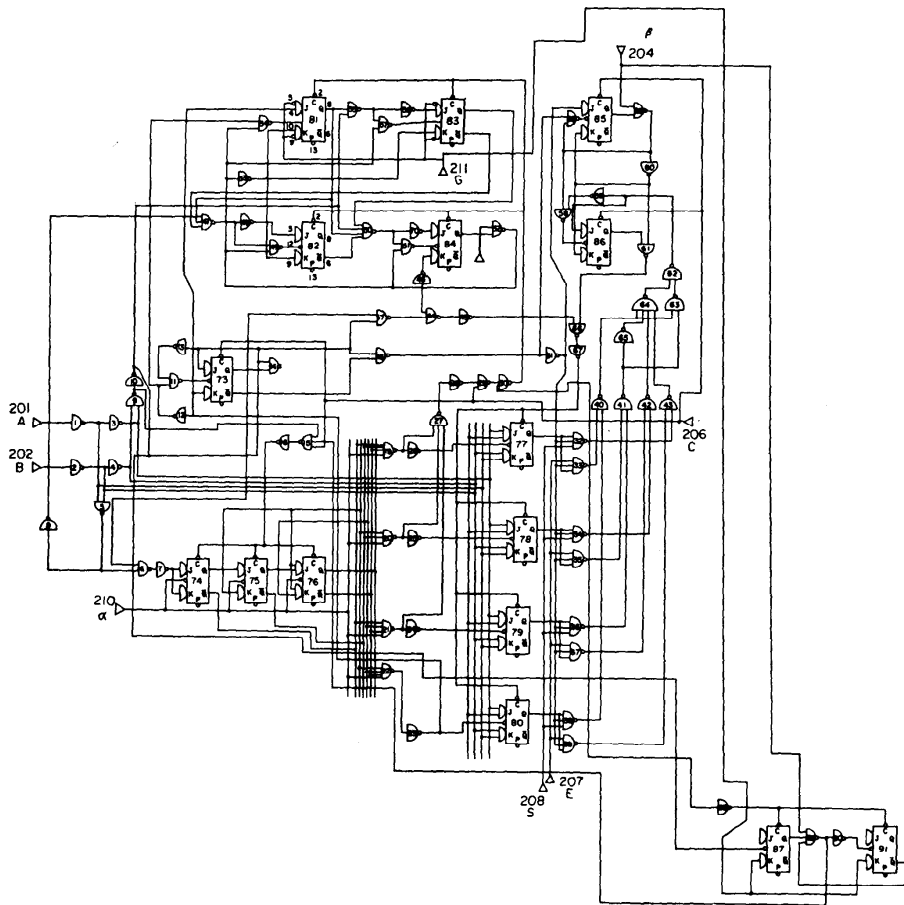
- (5) 基本ステップ (シミュレーションステップ) 数は最大 150 ブロック, または 2,400 ステップとする。

(ブロック指定の場合にはステップ数は最大 1,500 万となる。)

- (6) プリントアウト指定のできる端子数は最大 1,500 個である。

2.3.2 エラーメッセージ

GPLS-I では外部入力信号波形, 素子接続関係, 論理素子の三つの対象についてデータ記入の誤り, および論理動作上の明らかな誤りを監視している。第 2 表にエラーメッセージをプリントアウトする条件をかかげておく。エラーメッセージそのものはローマ字であ



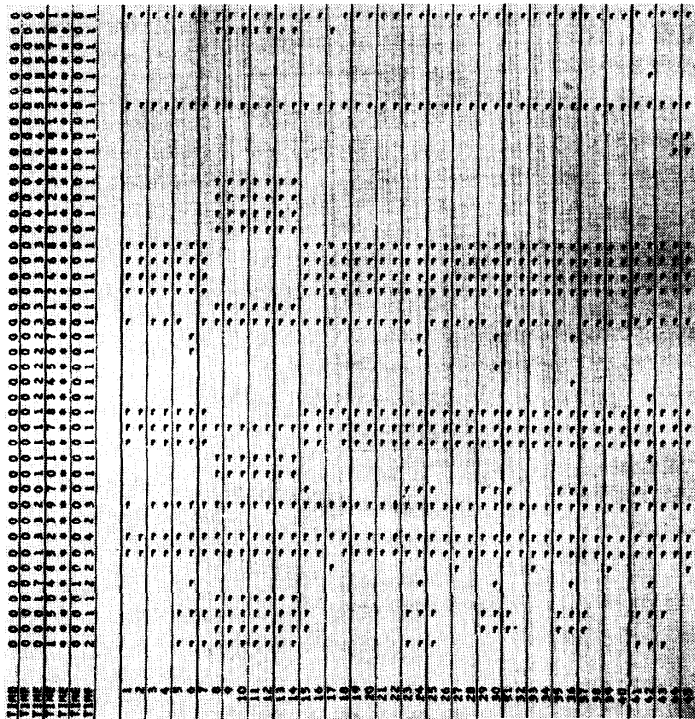
第 4 図 GPLS-I による論理回路検査実例

第2表 エラーメッセージ

エラーメッセージをプリントアウトする条件	計算機のとる処置
① 1ブロックが10万ステップ以上になる	計算続行
② 「+」「-」の符号が続く。例+6+5	計算続行 1個無視して計算
③ 同一符号が続く。例+, -, **	計算続行
④ 「/」スラッシュの後に「×」連続記号がある	「×」を無視して計算続行
⑤ 「/」スラッシュの後に文字がある	文字を無視して計算続行
⑥ 「/」スラッシュの後に文字と「×」連続記号がある	計算中止
⑦ 内容ブランクで「×」連続記号のみがある	計算続行 その文字をプリントアウトして計算
⑧ 数値, 「+」, 「-」, 「*」, 「×」, 「/」以外の文字がある	計算続行 外部入力信号は中断される
⑨ 基本ステップ数が外部入力信号に必要なステップ数に満たない	
接続関係	
① 外部入力端子、もしくは素子出力端子と全く無関係な結線がある	計算中止
② 接続関係の編集ができない	計算中止
論理素子	
① S-R FLIP-FLOP においてセット、リセット、入力ともに1となる	計算中止 当該のF/F番号 プリントアウト
② J-K, J-K MASTER-SLAVE FLIP-FLOP において CLEAR, RESET 端子がともに0になる	計算中止 当該のF/F番号 プリントアウト
③ ファンアウトが11以上になる	計算続行 当該の素子名 プリントアウト

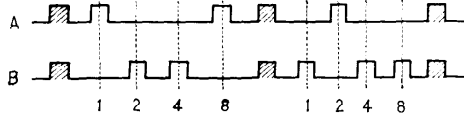
るが、容易に意味がわかるので、ここでは省略する。

3. GPLS-I による論理回路設計検査の例



第5図 論理回路検査例

第4図の論理回路の設計検査を行なった例について述べる。第4図において端子 A, B からは下図のような信号がはいる。



すなわち、端子Aからの信号はBCDの1を、また、端子Bからの信号はBCDの0を表わしている。各ディジットの区切りは端子A, Bの入力がともに1になることで判定される。

第4図の回路は端子A, Bの入力信号を読み取ると同時に、そのエラーチェックの一部を行なうように設計されている。たとえば、ディジットの区切りの間でA, B信号がともになくなって、1, 2, 4, 8のうちどれかのビットがかけてしまった場合、読取不能として出力はなしとする。入力信号が正常に端子A, Bからはいってくれば、各ディジットの値がBCDコードで第4図の素子(NAND) 40, 41, 42 および 43 の出力端子に現われる。

第5図に第4図の回路の論理設計検査の結果を示す。A, B 信号はこのタイムチャートの最下行 0001*02 および 0002*02 のごとく与えた。

シミュレータクロック 82 と 116 の間で、デジタル区切り間の1ビットがかけられているが、このとき素子 40, 41, 42 および 43 の出力端子、すなわち、0040*01, 0041*01, 0042*01, 0043*01 には出力が現われていない。

一方、正常な入力信号に対しては、読取りが正しく行なわれている。したがって、この検査により第4図の回路の設計は正しいと判定された。実物のハードウェアによる試験結果とも一致したことはもちろんである。

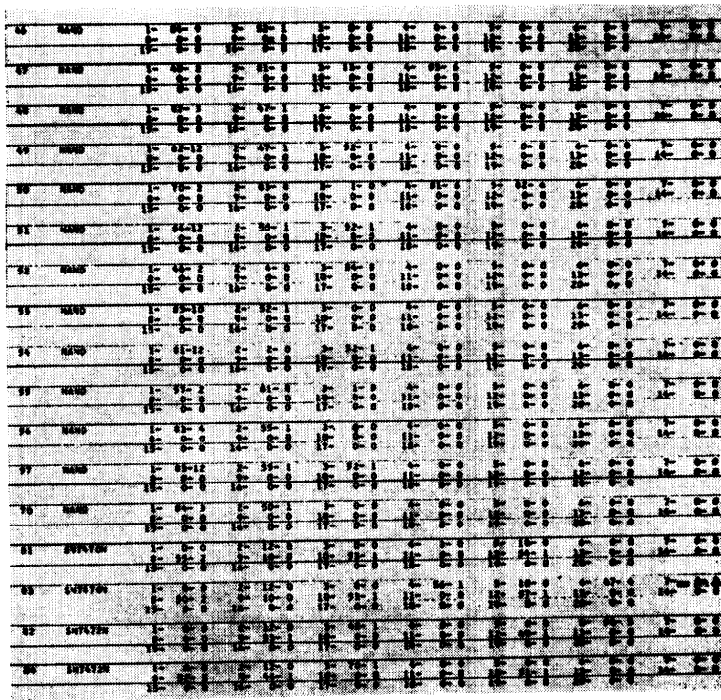
なお、第5図の結果を得るのに要した時間は約6分であった。現在はプログラム改良の結果、計算速度は約2倍になっている。

4. シミュレーションの方法

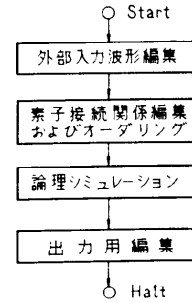
GPLS-I の論理シミュレーションは、右上の図のようなフローに従って行なわれる。

4.1 外部入力波形編集

外部入力端子からの入力信号は、第2章外部入力信



第6図 素子接続関係



号の項で述べたように、 $10-1*1+5*2+16\cdots$ のように、各端子ごとに文字として CPU に読み込まれ、数値に変換された後ストアされる。

4.2 素子接続関係編集およびオーダーリング

外部入力波形の編集が終了すると、論理シミュレーションの対象となる回路の論理素子の接続関係を入力データをもとに編集し、かつ、論理演算の順序づけ(オーダーリング)を行なう。以下第1図の回路を例にとって説明する。第1図の回路の素子の接続関係は、データ作成者が結線関係を適当にトレースして作成した未整理なデータとして CPU に読み込まれた後、電気的に等価な結線が整理されて第6図のようになる。

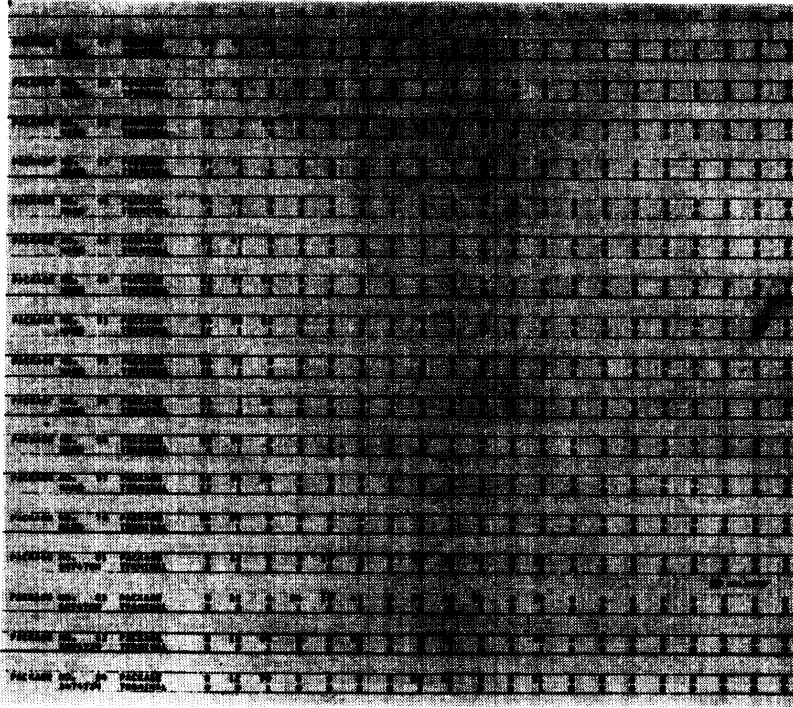
論理演算の順序づけは第6図の素子接続関係表をもとに行なわれる。まず、出力端子以外の端子がすべてフリップフロップ、もしくは外部入力端子に接続されている素子の出力が決定される。

第6図の場合にはこのような素子は、素子 47, 50, 52 および 55 の4個である。

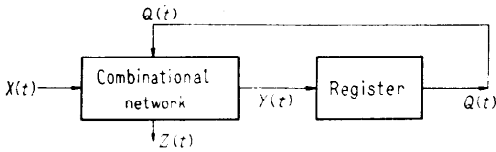
これら4個の素子の出力が決定された状態で再び第6図の接続表を調べる。以下同様の繰り返しを行なって論理演算の順序を決定すると、第6図の接続表を論理演算の順におきかえて、第7図のような順序付終了素子接続表を得る。以下の論理演算は第7図の表に従って行なわれる。

4.3 論理シミュレーションの方法

デジタルなシーケンシャルマシンは、一般に第8図のように



第7図 順序付終了素子接続表



第8図 A general form of sequential machines

$$\left. \begin{aligned} Y(n, m) &= A[X(n), Q(n, m)] \\ Q(n, m) &= B[Y(n, m-1)] \end{aligned} \right\} \quad (1)$$

(m=2, 3, 4, ...)

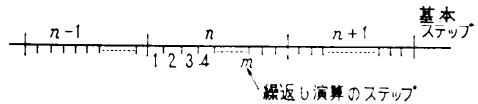
$$\left. \begin{aligned} Y(n, 1) &= A[X(n), Q(n, 1)] \\ Q(n, 1) &= B[Y(n-1)] \end{aligned} \right\} \quad (2)$$

(n=1)

ここで A, B は適当な論理演算の関数を表わしている。

表わすことができる。

第8図においてコンビネーションネットワークとは、記憶素子を含まない論理演算回路であり、レジスタとはフリップフロップによる記憶回路である。



$X(t), Y(t)$ 等は時刻 t における各部への入出力であって、外部入力端子が k 個あれば外部入力 $X(t)$ は $X(t) = (X_1(t), X_2(t), \dots, X_k(t))$ のように、ベクトル的に表現されているものとする。GPLS-I では第2章で述べたように、すべての信号をシミュレータ内部の基本ステップ(シミュレータクロック)に同期させて取り扱う。この基本ステップ(およびその番号)を n で表わし、かつ、1基本ステップ内部での繰返し演算のステップを m で表わすことにすると、(1)式、(2)式が成立する。

また、 $Y(n-1)$ は第 $(n-1)$ ステップ内で最終的に決定された Y の値、すなわち、 $Y(n-1, m) = Y(n-1, m+1)$ となるような m に対して、 $Y(n-1, m) = Y(n-1)$ としたものである。

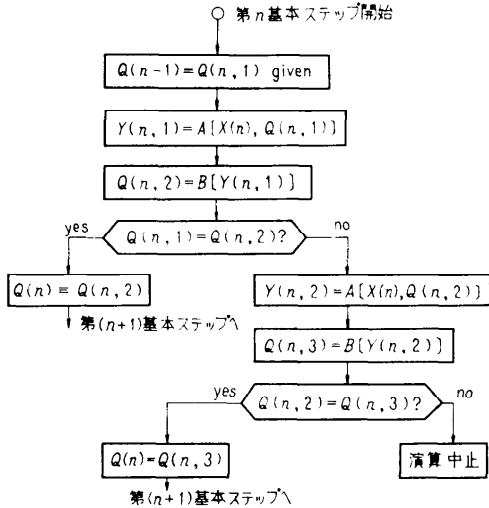
このような m が存在しない場合は、回路は不安定であって、論理検査は不合格ということになる。

1基本ステップ内で繰返し演算を行なうのは、第8図のレジスタに遅れの要素を含まないレーシングの可能性のある回路をも取り扱うためである。 $m=3$ で繰

返し演算を打ち切る場合のフローを第9図に示しておく。

4.4 出力用編集

以上のように、論理演算が1基本ステップ行なわれ



第9図 論理演算の方法

るごとに、第10図のような出力表を作成しコアにストアしておく。このような演算を40ステップ終了後、出力を第2図の形に編集し、プリントアウトする。

5. プログラム構成

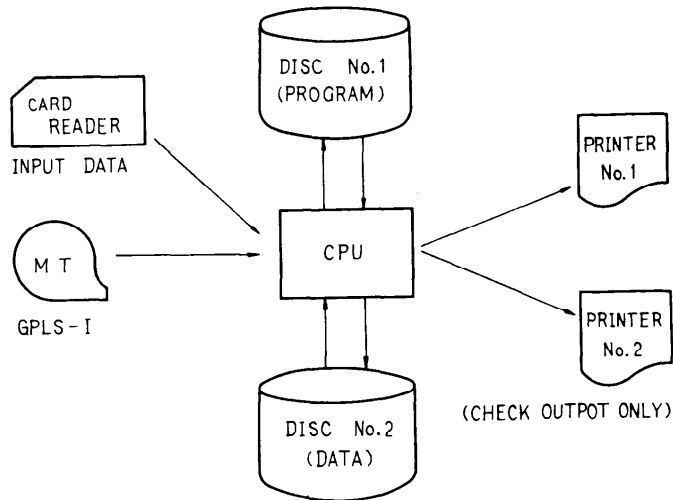
GPLS-I のプログラム構成の概略について述べる。

5.1 GPLS-I 使用時のコンピュータシステム構成

第11図に GPLS-I を使用する際のシステム構成を示す。使用コンピュータ IBM S/360 M-50(H) のCPU コアメモリ容量は 256 kbytes (=64 kW, 1 W=32 bits) である。GPLS-I は磁気テープに集録されており、使用時にはそのうちのメインルーチンのみが常時 CPU にあり、その他のルーチンは DISC No. 1 にストアされて、必要に応じて CPU に呼び出される。「コントロール」、「素子接続関係」、「プリントアウト」、および「外部入力信号」からなる入力データは、カードリーダーから CPU に読み込まれる。入力信号波形の編集、論理素子のオーダリング等の作業の際のデータエリアとして DISC No. 2 が使用される。論理演算の

TERMINAL NUMBER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PACKAGE NO. 47																				
PACKAGE NO. 50																				
PACKAGE NO. 53																				
PACKAGE NO. 55																				
PACKAGE NO. 44																				
PACKAGE NO. 48																				
PACKAGE NO. 49																				
PACKAGE NO. 51																				
PACKAGE NO. 52																				
PACKAGE NO. 54																				
PACKAGE NO. 56																				
PACKAGE NO. 57																				
PACKAGE NO. 58																				
PACKAGE NO. 59																				
PACKAGE NO. 60																				
PACKAGE NO. 61																				
PACKAGE NO. 62																				
PACKAGE NO. 63																				
PACKAGE NO. 64																				
PACKAGE NO. 65																				
PACKAGE NO. 66																				
PACKAGE NO. 67																				
PACKAGE NO. 68																				
PACKAGE NO. 69																				
PACKAGE NO. 70																				
PACKAGE NO. 71																				
PACKAGE NO. 72																				
PACKAGE NO. 73																				
PACKAGE NO. 74																				
PACKAGE NO. 75																				
PACKAGE NO. 76																				
PACKAGE NO. 77																				
PACKAGE NO. 78																				
PACKAGE NO. 79																				
PACKAGE NO. 80																				

第10図 出力表



第11図 GPLS-I を使用する際のシステム構成

結果はプリンタ No. 1 から打ち出され、中間結果をチェックするためのチェックアウトはプリンタ No. 2 より打ち出される。

5.2 メモリ配分、オーバーレイおよび I/O について

GPLS-I は FORTRAN IV によって書かれており大きく分けて、① メインルーチン、② 入力信号編集ルーチン、③ オーダリングルーチン、および④ シミュレータルーチンの四つの部分よりなっている（第 12 図参照）。それぞれのルーチンの役割は、つぎのとおりである。

① メインルーチン

GPLS-I のプログラムコントロールを行なう。

② 入力信号編集ルーチン

入力信号は $-3+3-27+\dots$ のように、文字として読み込まれる。これを通常の数字に変換して、各外部入力端子ごとに外部入力信号をストアしておく。

③ オーダリングルーチン

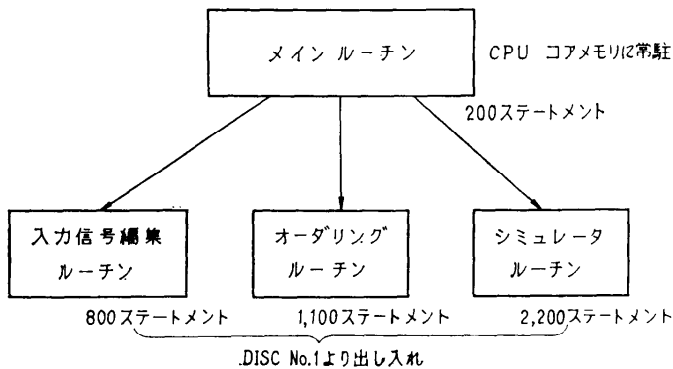
「素子接続関係」のデータをもとに接続関係を整理し、論理演算の順序を決定する。決定された演算順序は記憶され、繰り返し使用される。

④ シミュレータルーチン

②、③のルーチンによって整理されたデータに基づいて論理演算を行ない、結果をプリントアウトする。各論理素子の機能はそれぞれ独立のサブルーチンとして集録されて、シミュレータルーチンの一部となっている。

つぎに GPLS-I の演算の順序をメモリの配分、オーバーレイ、外部記憶装置との I/O の立場から簡単に述べる。

第 12 図および第 13 図を参照されたい。



第 12 図 GPLS-I のオーバーレイ

(1) 磁気テープにおさめられた GPLS-I のオブジェクトプログラムを CPU に読み込み、前記①、②、③、④のルーチンのうち、① メインルーチン、および② 入力信号編集ルーチンを CPU コアメモリに残して③、④のルーチンを DISC No. 1 にストアする。

(2) カードリーダーより「コントロール」、「素子接続関係」および「プリントアウト」入力データを CPU に読み込み、コアメモリにストアする。

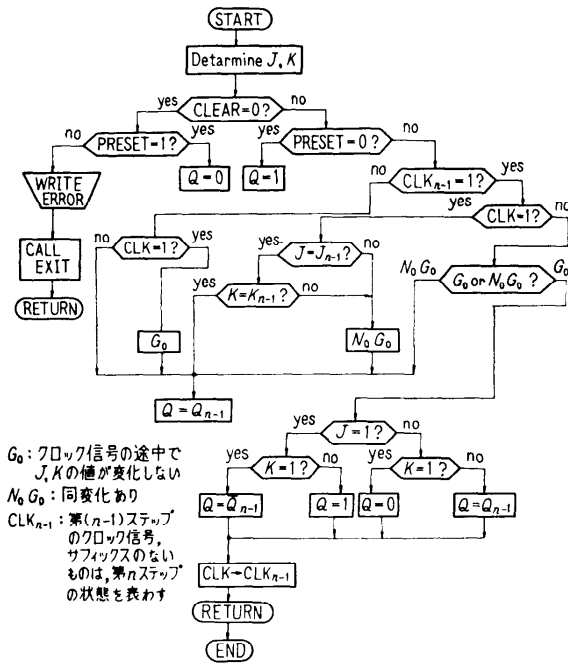
(3) カードリーダーより「外部入力信号」データを読み込む。外部入力信号データは、外部入力端子 1 個ごとに読み込まれ、編集され、CPU コアメモリにストアされる。外部入力端子 1 個あたりの入力信号は、たとえば $+28-1+25-1+24-1+15/$ のように文字として読み込まれ、数値（整数）に変換されて CPU コアメモリ内にストアされる。

(4) 全外部入力端子の入力信号が適当な形に編集されて CPU コアメモリにストアされると、その結果（15,000 half-words）および入力データ中オーダリングに不要なデータが、すべて DISC No. 2 へ移される。また、入力信号編集ルーチンがコアから DISC No. 1 へ移され、代わりに DISC No. 1 からオーダリングルーチンが取り出されてコアに入れられる。

(5) CPU コア内に素子接続関係のデータと GPLS-I メインルーチン、およびオーダリングルーチンがはいっている状態で、論理演算の順序づけを行なう。

(6) オーダリングを終了すると、オーダリングルーチンを CPU から取り出し DISC No. 1 に入れ、代わって DISC No. 1 よりシミュレータルーチンを取り出して、コアメモリに入れる。

入力信号編集後に DISC No. 2 に入れられた入力信号波形のデータ等論理演算に必要なデータ、および出力タイムチャート作成時に必要なヘッディング用データが、この時点で DISC No. 2 よりコアメモリに読み込まれる。ヘッディングはタイムチャートを作成する際に、出力を観測すべき素子名と端子名を見やすく編集するものである。ヘッディング終了後論理演算にはいるが、演算結果のタイムチャートは 40 ステップ×60 素子を 1 単位として、コアメモリにストアされた後、プリントアウトされる。



第 13 図 J-K master-slave flip-flop のフローチャート

5.3 論理素子のフローチャート例

各論理素子の論理演算は, それぞれ一つのサブルーチンを構成している. 一例として第 13 図に J-K master-slave flip-flop のフローを掲げておく. 図中 J, K 等の意味については第 1 表を参照されたい.

6. 結 言

中小型の工業計測器型デジタル論理装置のスタティックな論理設計検査を行なうためのコンピュータプ

ログラム GPLS-I について述べた. GPLS-I の主要な特徴を要約すると, つぎのとおりである.

(1) FORTRAN IV によって書かれており使用コンピュータは IBM S/360 M-50 (もしくは NEAC 2200/500) である.

(2) 対象とする論理素子の種類は, 通常よく使用される 11 種類の論理素子であり, 微分や遅れの素子の他, Texas Instruments 社の IC J-K flip-flop や J-K master-slave flip-flop をも含んでいる.

(3) 同期式論理素子のみならず, 非同期式論理素子も取り扱う. クロック信号そのものも外部入力信号の一種として扱われる.

(4) 多品種小量生産の工業計測器型の論理装置の設計検査に適しており, GPLS-I を使用する場合コンピュータの知識も, コンピュータプログラムの知識も全く不要である.

金物による実物の試作なしに論理設計検査を行なうことによって, 設計者の man-hour や資材の節減をはかりうる事が, 実際の使用例からも確認された.

終わりにあたりご指導をいただいた京都大学工学部電気工学教室長尾講師, 住友電工システム開発部中原部長, および計数課伊藤課長, 山崎氏に深甚なる謝意を表するものである.

参 考 文 献

Thomas C. Bartec, Irwin L. Lebow, and Irving S. Reed: Theory and design of digital machines, Mc Graw-Hill Book Company

(昭和 43 年 6 月 1 日受付)