

## 論理回路シミュレータ GPLS-II とその応用\*

荒牧 達\*\* 柴田 潤\*\*

## Abstract

General Purpose Logic Simulator GPLS-II is unique in that it uses the concept of simulation steps and phases in order to handle synchronous and asynchronous logic circuits as well as mixtures of both types of circuits. The fundamental unit of time in GPLS-II is called the "step". Each logical state within the step which corresponds to a state of the register is called the "phase". Thus there may be a number of phases within a step when the logic circuit contains loops without delays, elements or asynchronous elements. As an example of GPLS-II in action, design check of a terminal in an area traffic control system was carried out, and a possibility of error caused by a gate enabled instantaneously was detected. This paper plays the role of a supplement to and an application of "Logic Simulator GPLS-I," a paper appeared in Vol. 10, No. 1 of the Journal of the Information Processing Society of Japan.

**要約** 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設計図上で行なうための論理シミュレータ GPLS-II (General Purpose Logic Simulator Version II) は論理シミュレーションにステップとフェーズの概念を導入している点に特徴がある。

ステップとは GPLS-II における単位時間であって, 外部からの入力信号はすべてステップに同期させて取り扱う。

フェーズとは論理回路内のレジスタの一つの値の組に対応して論理演算を行なう場合の状態であって, 回路の中に非同期式の部分を含む場合や, 時間遅れの素子を含まないループを含む場合には, 一つのステップの中に複数個のフェーズが含まれる。

応用例として, 広域交通制御用の端末機の設計に GPLS-II を用いた結果, ゲートの条件がいっしゅんそろうために発生する細いパルスによってフリップフロップが誤動作する可能性のあるケースが検出された。

本論文は先般発表した「デジタル機器の論理設計検査プログラム GPLS-I について」(末尾参考文献参照) の補遺および応用編の役割をなすものである。

## 1. ま え が き

同期式, 非同期式にかかわらず, 中小型のデジタル機器の論理設計検査を行なうための論理シミュレータとして筆者らは, かつて GPLS-I を発表した。

その後, 実際の使用をとおしてアルゴリズムの改良を行なうとともに, 使用素子に D フリップフロップを追加するなどの変更を加えた結果, GPLS-I は新しく GPLS-II として生まれ変わった。

本論文では GPLS-II のアルゴリズムを詳細に述べるとともに, 広域交通制御用の端末機器の論理設計に GPLS-II を用いた例を報告する。

GPLS-II (General Purpose Logic Simulator Version II) 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行なうためのアプリケーションプログラムである。GPLS-II で取扱う論理素子を第 1 表に示す。

GPLS-II においては外部からの入力信号も, デジタル機器内部の信号も, 出力信号もすべて logical 0 または logical 1 の方形波であって, その時間の単位は GPLS-II 内部のステップ (およびフェーズ) により定められている。

論理回路のデータは回路図から論理素子の接続関係を読み取って, たとえば「素子 29 の端子 1 は素子 22 の端子 2 へ」というように入力する。GPLS-II は, このようにして与えられた論理回路図と, 別途指定さ

\* Logic Simulator GRLS-II and its Application by Itaru Aramaki, Yosimi Fujita and Jun Shibata (Sumitomo Electric Industries, Ltd.)

\*\* 住友電気工業株式会社

第 1 表 GPLS-II 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
15	SN 7474 N	D 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	NOR	$Z = A + B + \dots + S$		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, A, B, ... S はそれぞれ端子 2, 3, ... 19 の入である。
28	NAND	$Z = A \cdot B \cdot C \cdot \dots \cdot S$		
29	OR	$Z = A + B + \dots + S$		
30	AND	$Z = A \cdot B \cdot C \cdot \dots \cdot S$		
31	NOT	$Z = \bar{A}$		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 の入力である。
32	RSFF	R-S FLIP-FLOP		セット入力は端子 3, 4, ... 10 の入力の OR となる。また, リセット入力は端子 11, 12, ... 18 の入力の OR 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。

れた外部入力信号とから回路図内各部の信号レベル (logical 0 または logical 1) を計算し、タイムチャートとして出力する。

## 2. GPLS-II のアルゴリズム

GPLS-II の対象となる論理回路は一般に第 1 図のように表わすことができる。

第 1 図において組み合わせ回路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や遅れ素子のような記憶素子よりなる回路である。

$X(t)$ ,  $Y(t)$  などは時刻  $t$  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。

レジスタの出力  $Q(t)$  および外部入力  $X(t)$  をレベル 0 とし、組み合わせ回路内の素子のレベルを  $Q(t)$ ,  $X(t)$  より決定される素子群はレベル 1, その次に決定される素子群はレベル 2,

.....

のように論理演算が後段になるにつれてレベルが高くなるようにする (第 2 図参照)。

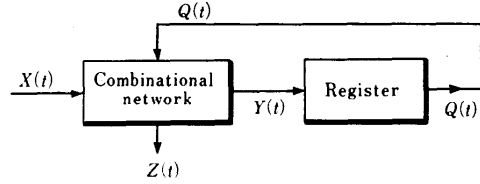
組み合わせ回路からの出力  $Y(t)$ ,  $Z(t)$  は  $X(t)$ ,  $Q(t)$  をもとに第 2 図に示すようなレベルに従って順次計算することにより決定される。組み合わせ回路内の論理演算の順序は、素子の接続関係によって一意的に決定されるが、レジスタ内部のフリップフロップや遅れ素子の演算順序は、GPLS-II ではこれらの素子に対して回路図上で与えられた素子番号によって決定される。

第 1 図の論理回路は必ずしも同期式であるとはかぎらず、また  $Y(t)$  から  $Q(t)$  へのループの中に時間遅れが含まれているとはかぎらないので、GPLS-II ではステップとフェーズの概念を導入した。

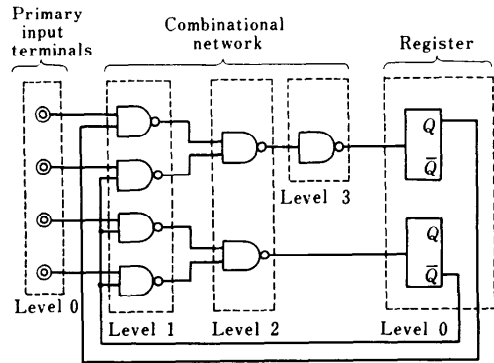
ステップとは GPLS-II における単位時間であって、外部からの入力信号  $X(t)$  はすべてステップに同期させてとりあつかう。

フェーズとはレジスタの一つの値の組に対応して第 1 図の回路の論理演算を行なう場合の状態であって、第 1 図の回路に非同期式の部分を含む場合や、 $Y(t)$  から  $Q(t)$  へのループの中に時間遅れが含まれない場合には一定の外部入力  $X(t)$  に対して  $Y(t)$ ,  $Q(t)$  がなかなか落ち着かず、一つのステップの中に多数のフェーズが含まれる可能性がある。

ステップの番号を  $n(n=1, 2, \dots)$ , フェーズの番号



第 1 図 GPLS-II の対象となる論理回路



第 2 図 論理演算レベルの決定

を  $m(m=1, 2, \dots)$  で表わし、ステップ  $n$ , フェーズ  $m$  のときの  $X(t)$ ,  $Y(t)$ ,  $Q(t)$  および  $Z(t)$  の値をそれぞれ  $X(n, m)$ ,  $Y(n, m)$ ,  $Q(n, m)$  および  $Z(n, m)$  と書くことにすると、第 1 図の回路について (1), (2) 式が成立する。

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

$(m=2, 3, 4, \dots)$

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

$(m=1)$

ただし  $C, R$  はそれぞれ組み合わせ回路、レジスタを表わす論理関数であり、 $Y(n-1)$  は第  $(n-1)$  ステップ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  $Y(n-1, m) = (n-1)$  としたものである。

このような  $m$  が存在しない場合に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。非同期式回路では  $m$  は最悪の場合、回路内の非同期素子の数に等しくなる。これを防ぐため回路図内で記憶素子に番号をつけるとき信号の流れに従って番号を大きくするようにすればよい。

### 2.1 J-K フリップフロップの取り扱い

GPLS-II で取り扱う J-K フリップは、第 1 表に示

すように Texas Instrument 社の SN 7470 N, もしくはこれに準ずるものである. SN 7470 N の J, K 入力の設定アップタイムは標準 10 ns, 最大 20 ns である. またホールドタイムは標準 0 ns, 最大 5 ns である. 一方, 同社の NAND ゲート, たとえば SN 7400 N の logical 0 レベルへのプロパゲーションディレイ時間は標準 8 ns, 最大 15 ns であり, また logical 1 レベルへのプロパゲーションディレイ時間は標準 18 ns, 最大 29 ns である (第 3 図参照).

したがって第 4 図 (b) の場合にはクロック入力は J, K 入力に対して標準 26 ns, 最大 44 ns 程度の遅れを持つことになり, 第 4 図 (a), (b) の J-K フリップフロップの動作は第 5 図 (a), (b) のように異なる.

第 4 図 (b) のようにゲートによってクロック入力を遅らせるばかりでなく, コンデンサを用いて図 (b) と等価な動作を行なわせる場合もある.

GPLS-II ではこのような J-K フリップフロップのエッジトリガーの特性を表わすために, J-K フリップフロップの演算に際して次のようなコントロールを行なう.

(1) **J, K, クロック入力のレベルの決定**  $J_1, J_2, J^*, K_1, K_2, K^*$  の入力を構成する組み合わせ回路内の論理素子のレベルの最大値を,

$$(JK)_{level}$$

とする. またクロック入力を構成する組み合わせ回路内の論理素子のレベルをクロックの入力レベルとして,

$$(CLK)_{level}$$

と書く. クロックの入力が同期回路の場合のように外部入力端子と直結されている場合には, 外部入力端子のレベルは 0 であるから,

$$(CLK)_{level} = 0$$

となる. J, K 入力がオープンとなっている場合にはオープン端子はレベル 0 として,

$$(JK)_{level} = 0$$

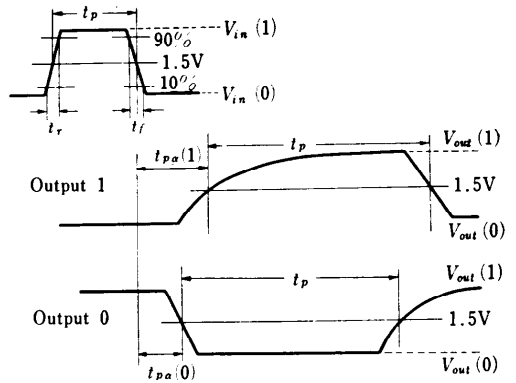
となる.

(2) **J-K フリップフロップの選別** (1) で決定した  $(JK)_{level}$  および  $(CLK)_{level}$  を比較して,

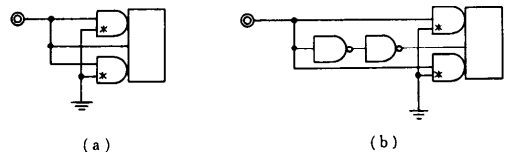
$$(JK)_{level} < (CLK)_{level} \quad (3)$$

ならば, 第  $n$  ステップの外部入力と, 第  $n$  ステップ内第  $(m-1)$  フェーズの演算結果である  $Q(n, m-1)$  を用いて計算した  $J(n, m), K(n, m)$  の値をもって  $Q(n, m)$  を計算する. このような J-K フリップフロップを  $(JKFF)_m$  と書く.

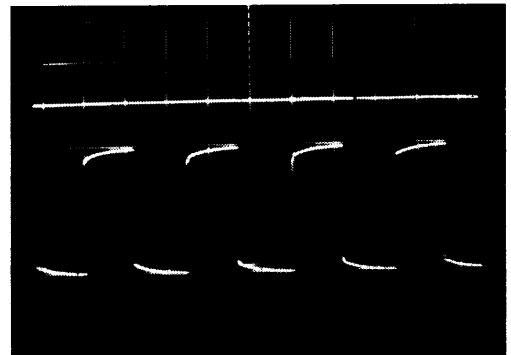
もし,



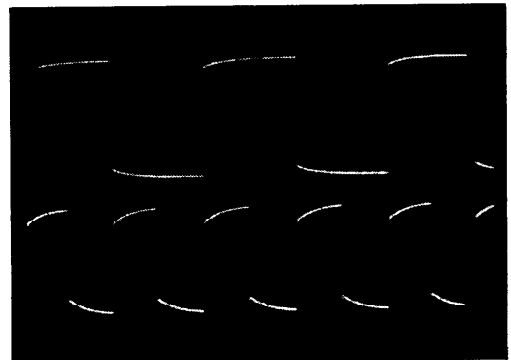
第 3 図 プロパゲーションディレイ時間



第 4 図 J-K フリップフロップの接続



(a) 第 4 図 (a) に対応



(b) 第 4 図 (b) に対応

第 5 図 フリップフロップの動作

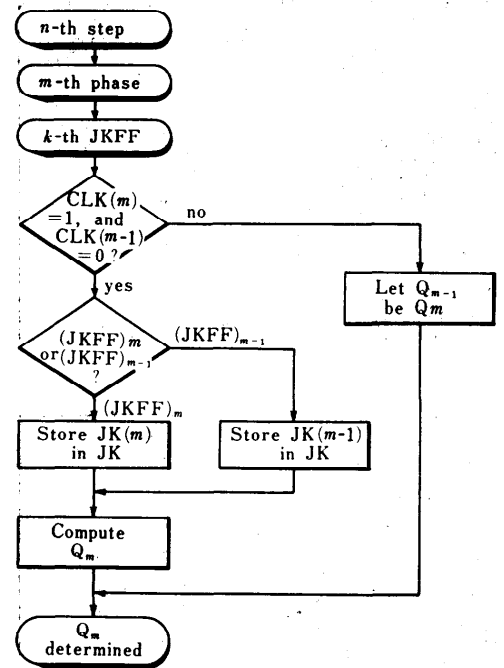
$$(JK)_{level} \geq (CLK)_{level} \quad (4)$$

ならば、第  $n$  ステップの外部入力と、 $Q(n, m-1)$  を決定する際に用いた  $J(n, m-1)$ ,  $K(n, m-1)$  の値をもって  $Q(n, m)$  を計算する。このような J-K フリップフロップを  $(JKFF)_{m-1}$  と書く。

以上のように J-K フリップフロップを  $(JKFF)_m$  と  $(JKFF)_{m-1}$  に選別してから論理演算を第 6 図のように行なう。第 6 図においては第  $n$  ステップ内の演算を問題にしているので  $Q(n, m) = Q_m$  などと略記している。

まずクロック入力  $CLK(n, m) = 1$ ,  $CLK(n, m-1) = 0$  であるかどうかを調べる。もしそうであれば第  $(m-1)$  フェーズから第  $m$  フェーズに移ったときに、クロック入力が上がったことになるので、当該 J-K フリップフロップが  $(JKFF)_m$  であるか  $(JKFF)_{m-1}$  であるかによって  $J(n, m)$ ,  $K(n, m)$  もしくは  $J(n, m-1)$ ,  $K(n, m-1)$  の組を用いて  $Q(n, m)$  を計算する。クロック入力第  $m$  フェーズで logical 0 から logical 1 に上がっていないときはもちろん、

$$Q(n, m-1) = Q(n, m)$$



k-th JKFF storage

CLK (m)	CLK (m-1)	$Q_{m-1}$	$(JKFF)_m$ or $(JKFF)_{m-1}$
J (m)	J (m-1)	K (m)	K (m-1)

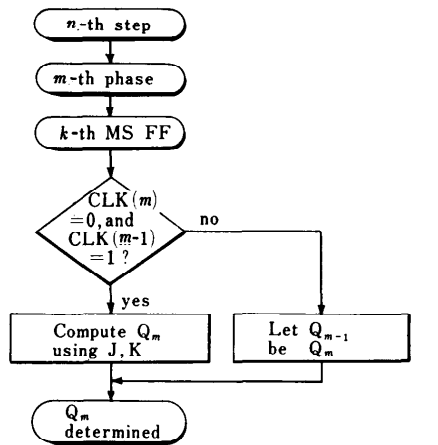
第 6 図 J-K フリップフロップの演算方式

である。

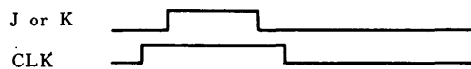
以上 J-K フリップフロップについて述べたが、エッジトリガーの機能は D フリップフロップについても同様であり、 $(JKFF)_m$ ,  $(JKFF)_{m-1}$  に対応して  $(DFF)_m$ ,  $(DFF)_{m-1}$  の区別を行なう。D フリップフロップの場合には、データ入力とクロック入力の問題になる。

**2.2 J-K MASTER-SLAVE フリップフロップの取り扱い** GPLS-II で扱う J-K MASTER-SLAVE フリップフロップ (以下 M-S フリップフロップと呼ぶ) は第 1 表に示すように、Texas Instruments 社の SN 7472 N, もしくはこれに準ずるものである。

第 7 図に M-S フリップフロップの演算方式を示す。



Step-----	n-2	n-1	n
Phase-----	1 2 ---	1 2 3 ---	1 ... m-1 m



k-th MS FF storage		
CLK (m)	CLK (m-1)	$Q_{m-1}$
J	K	

第 7 図 M-S フリップフロップの演算方式

第 7 図においてクロック入力  $CLK(n, m)$  は第  $n$  ステップの外部入力信号と第  $(m-1)$  フェーズの演算結果であるレジスタの値  $Q(n, m-1)$  をもとに組み合わせ回路の演算の結果として決定される。

また  $J, K$  は第  $(m-1)$  フェーズ以前の  $J, K$  の値およびクロック入力の値によって決定される値である。M-S フリップフロップは  $CLK(n, m) = 0$ ,  $CLK(n, m-1) = 1$  のときに第  $m$  フェーズで動作する。このと

き  $Q(n, m)$  決定のために使用する  $J, K$  の値はつぎのように決定する。

クロック入力が第  $n$  ステップ、第  $m-1$  フェーズ以前に連続的に logical 1 である間に 1 フェーズの間でも  $J$  入力に logical 1 になれば  $Q(n, m)$  は  $J=1$  として計算する。  $K$  入力についても同様である。

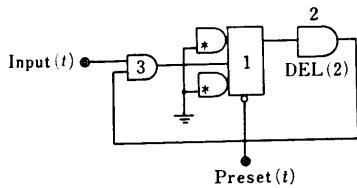
なおプリセットもしくはクリアの信号が M-S フリップフロップに加えられると無条件に  $Q(n)=1$  または  $Q(n)=0$  となるが、このとき上に述べた  $J, K$  の記憶もクリアしていったん  $J=K=0$  としなければならない。

**2.3 GPLS-II による「ひげ」の検出 第8図(a)** は J-K フリップフロップ、AND 素子、DELAY 素子よりなる論理回路の一例である。

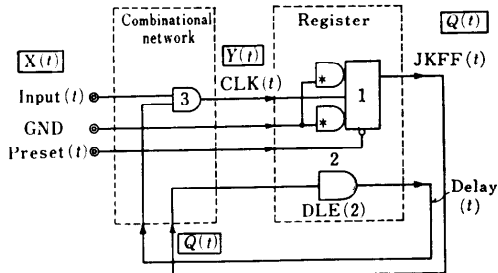
第8図の DELAY 素子は2ステップ時間遅れを与えるものとしてある。第8図(a)を第1図の標準形にかきなおしたものが第8図(b)である。

第8図(b)において組み合わせ回路への入力端子は常時 logical 0 の GND 端子、J-K フリップフロップのプリセット端子への入力である PRESET 端子および信号用の INPUT 端子よりなっている。組み合わせ回路の素子は AND 素子1個だけであり、レジスタは J-K フリップフロップと DELAY 素子よりなっている。したがって、この場合には、

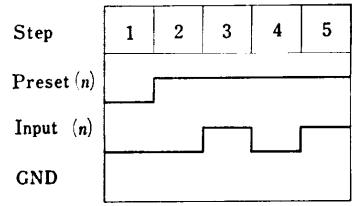
$$\left. \begin{aligned} X(t) &= [\text{PRESET}(t), \text{INPUT}(t), \text{GND}], \\ Y(t) &= [\text{PRESET}(t), \text{CLK}(t), \end{aligned} \right\} \quad (5)$$



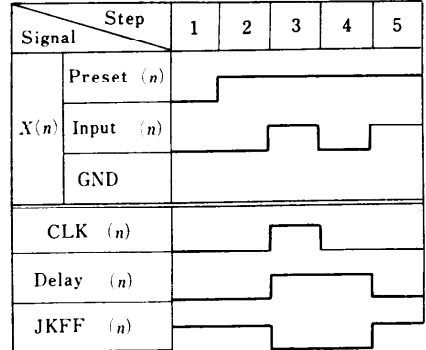
(a) 回路図



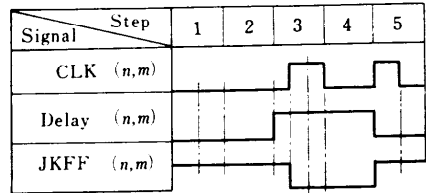
(b) 第1図の形に書きなおしたもの  
第8図 「ひげ」の検出



第9図 第8図(b)の入力信号  $X(t)$



(a) 通常の出力タイムチャート



(b) 詳細な出力タイムチャート

第10図 第8図の回路の GPLS-II による出力タイムチャート

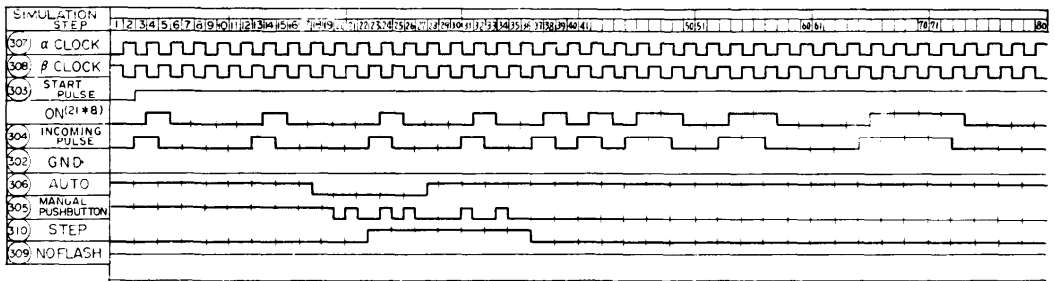
$$\left. \begin{aligned} \text{JKFF}(t), \text{GND}], \\ Q(t) &= [\text{JKFF}(t), \text{DELAY}(t)] \end{aligned} \right\}$$

となる。ただし  $\text{CLK}(t)$ ,  $\text{JKFF}(t)$  などの記号については第8図(b)を参照されたい。

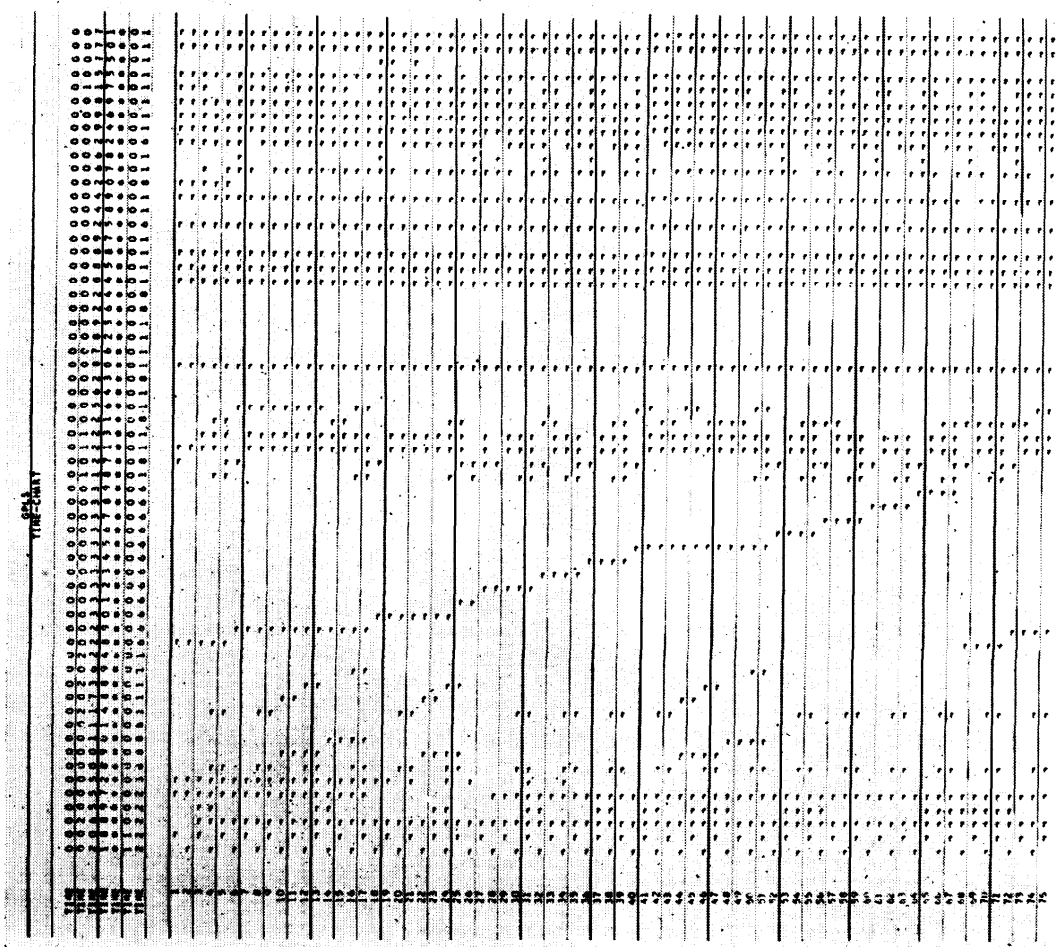
いま  $X(t)$  に第9図のような信号を与える。このとき GPLS-II のアルゴリズムに従って  $Y(n, m)$ ,  $Q(n, m)$  を逐次計算し、結果をプロットしたものが第10図(b)である。

第10図(b)を見ると第5ステップでクロック入力に logical 0 のままであるにもかかわらず、J-K フリップフロップが logical 0 から logical 1 に反転している。これは INPUT 端子からの入力信号の立上がり時に DELAY 素子の出力の立下がり時にいっしょにクロック CLK を logical 1 とする条件がそろったためである。このとき J-K フリップフロップを反転させる CLK 信号は、いわゆる「ひげ」と呼ばれるもの





第 12 図 SSC-4 制御部への外部入力信号



第 13 図 SSC-4 制御部第 1 回



一種であって、回路の誤動作の原因となる。

第10図(b)には CLK ( $n, m$ ) などの値をプロットしておいた。これによると第5ステップで J-K フリップフロップが反転した理由が一目でわかる。

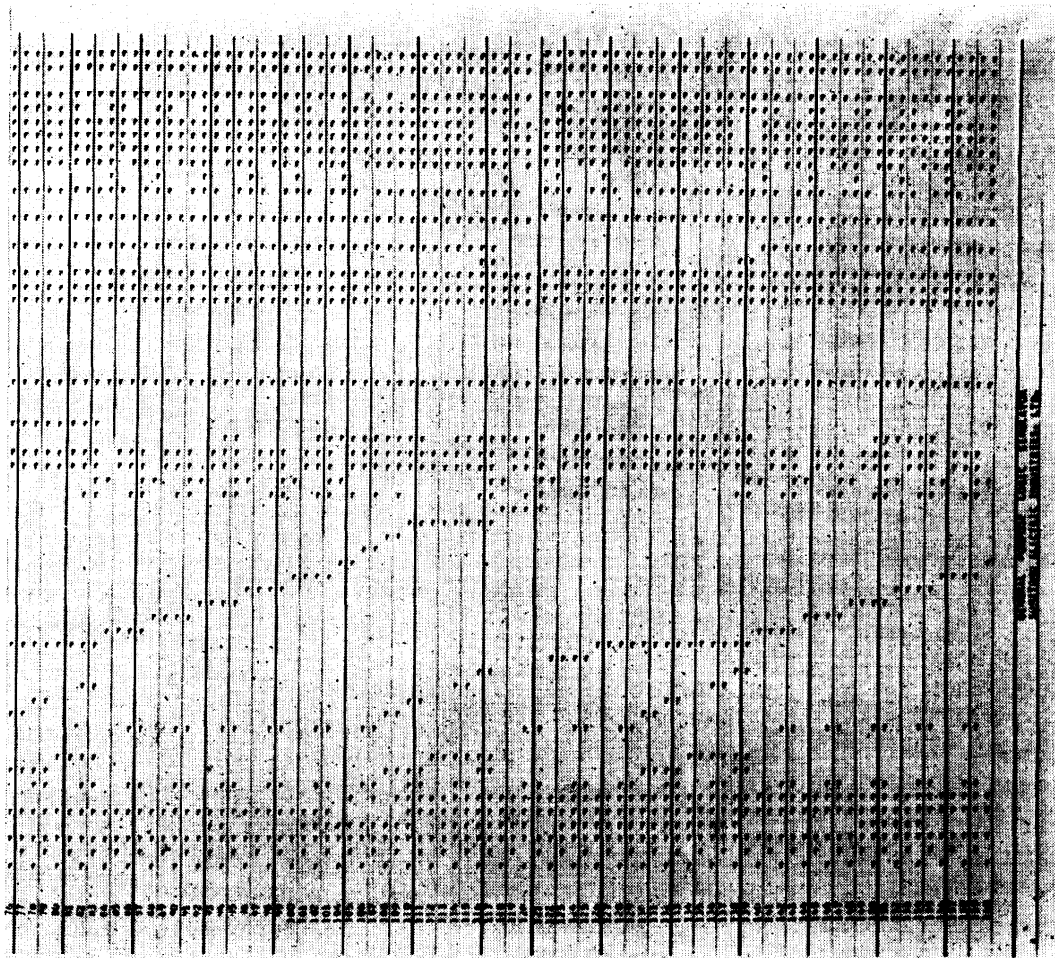
GPLS-II では第10図(b)に相当する出力も指定すればタイプアウトさせることができる。

以上述べたような「ひげ」検出の能力は、安全サイドで論理設計をする場合に有効な GPLS-II の特徴の一つである。

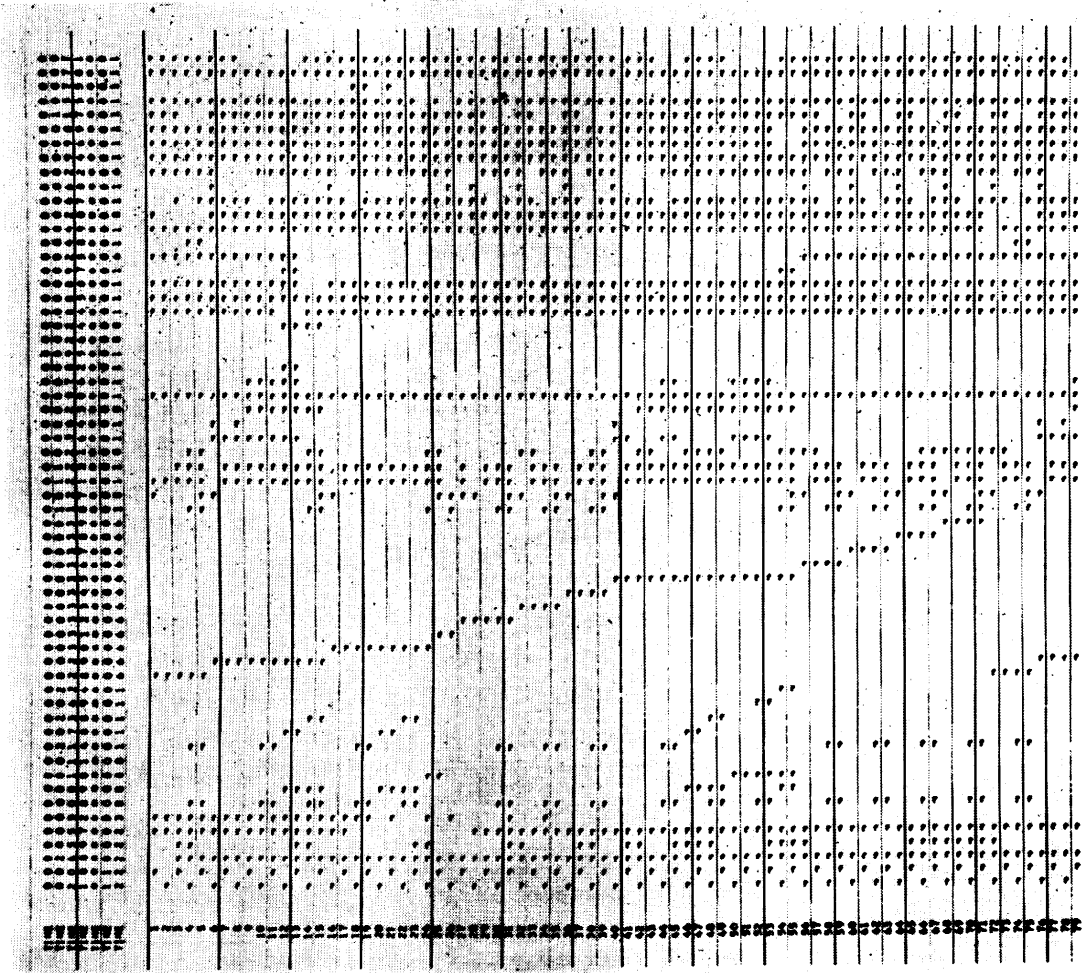
### 3. GPLS-II を用いた広域交通制御端末機的设计

2. では GPLS-II のアルゴリズムについて述べた。本章では GPLS-II を用いて中小型のデジタル機器を設計した例について述べる。

設計例としては広域交通制御用の端末機を選んだ。本機は都市道路網の自動車交通の制御を中央に設置されたコンピュータで行なう、いわゆる広域交通制御シ



論理シミュレーション



第 14 図

テスムの制御用端末として使用されるもので、従来の信号機にかわって交差点に設置される。

GPLS-II で取り扱った端末機は SSC-4 型と呼んでいる。

第 11 図の SSC-4 制御部の外部入力端子 302~310 に第 12 図のような入力信号を加えて GPLS-II による論理シミュレーションを行なった結果を第 13 図に示す。

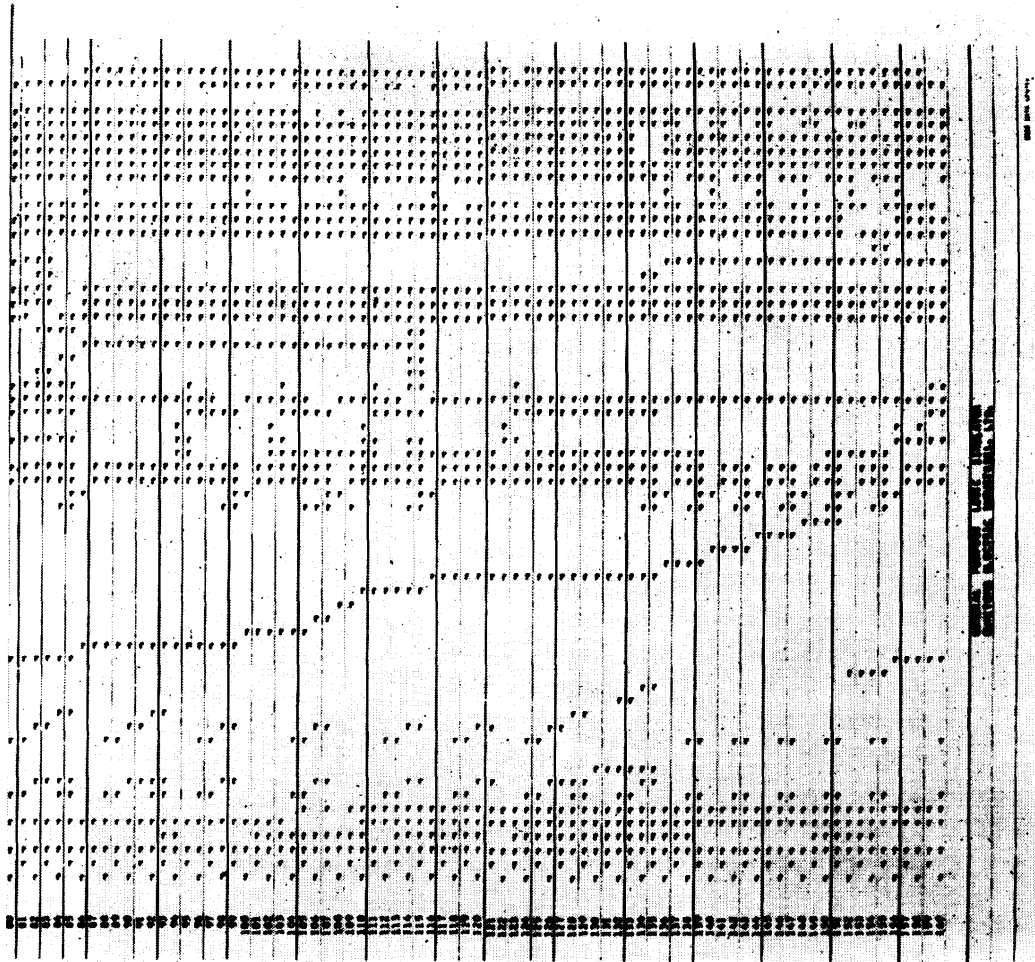
第 13 図において縦軸の数字は素子の番号と端子の番号を表わす。たとえば 0021\*12 は素子 21 の端子 12, すなわち  $\alpha$  クロックを表わしている。横軸の数字はステップ番号を表わしている。NAND 素子の出力端子は必ず 1, 入力端子は 2, 3 の順に番号がつけられている。

MSFF 20 への入力 は NAND 48, 49 であるから

48\*1, 49\*1 をみると 48\*1 は logical 0, 49\*1 は logical 1 のレベルになったままである。入力の変化がないのに、第 6 ステップにおいて 20\*8 が logical 1 から logical 0 に変化しているので、2.3 で述べた「ひげ」が発生した可能性がある。

NAND 49 への入力 は STEP 1 と DSHIFT であるから、28\*6 と 121\*1 をみると両者とも信号のレベルが第 6 ステップで変動している。STEP 1 の logical 1 のレベルと DSHIFT パルスの logical 1 のレベルが第 6 ステップの当初で一瞬一致したために NAND 49 の出力がこの瞬間 logical 0 になり、このため 20\*8 が第 6 ステップで logical 0 になってしまったものと考えられる。

対策として第 2 ステップで強制的に STET INT の信号レベルを logical 1 にしてやることにして MSFF



20 のプリセット端子 13 に 29\*8 を接続した。29\*8 の信号は第 2 ステップの間中 logical 0 になっている。このような変更を施した後の SSC-4 制御部の動作を第 14 図に示した。第 14 図の結果では第 13 図で発見した不具合は解消している。

#### 4. 結 言

同期式であるか、非同期式であるかをとわず、中小型のデジタル機器の論理設計検査を行なうための論理シミュレータ GPLS-II のアルゴリズムについて述べた。

また GPLS-II を広域交通制御システムの端末機的设计に用いた例について説明した。

GPLS-II の今後の課題はプログラムそのものの高性能化による時間短縮、論理素子として LSI をも考

慮することなどである。

おわりにあたり、ご指導をいただいた京都大学工学部長尾助教授、住友電気システム開発部中原部長およびご協力をいただいた日本アイビーエム河村氏、大阪大学基礎工学部川畑氏、住友電気システム開発部有本氏、同計数部太田氏らの方々に深甚なる謝意を表する。

#### 参 考 文 献

- 1) Thomas C. Bartee, et al.: Theory and design of digital machines, Mcgraw-Hill Book Company.
- 2) 荒牧, 河村: デジタル機器の論理設計検査プログラム GPLS-1 について, 情報処理, 10, No. 1 (1969).

(昭和 44 年 11 月 19 日受付)