シーケンス制御系の論理設計

小 山 昭 一*
三 浦 良 一*

The Logical Design of Sequential Control Systems

Shoichi KOYAMA
Ryoichi MIURA

Abstract

The purpose of this paper is to investigate systematically the practical method of the logical design of sequential control systems.

Recently, with the development of the theory of digital computer, the theory of finite automata has greatly developed, and many powerful design procedures have been investigated. But many of the design procedures of sequential control systems, although these sequential control systems can be regarded as special cases of the finite automata, still depend on direct inspection and are not systematical because of their specialities.

We have studied the specialities of the sequential control systems as special cases of finite automata, and consequently we have succeeded in simplifying the design procedures of finite automata for their applications to the logical design of sequential control systems.

1. 緒　　言

自動制御の分野におけるフィードバック制御の分野は，理論・実用の両面ともに，著しい発展をとげたことは周知の通りである。しかし，これと本質的に手法の異なったシーケンス制御の分野は，実用的には広く普及しているにもかかわらず，その設計法の多くは直観と視察による方法であり，合理的な設計理論はほとんど未開発の状態にとどまっている。

一方，近年ディジタル計算機理論の発達にともない，有限オートマトンの理論[1),4),6)]の発展は著しいものがあり，これ等有限オートマトンの解析・設計に関する有力な手法が非常に多く研究・開発されて来た。著者等は，これ等有限オートマトンの理論を検討した結果，シーケンス制御系は広い意味での有限オートマトンの一種と考えることができることに注目して，その適用を研究した。

しかし，従来の有限オートマトンの設計理論を，シーケンス制御系の論理設計にそのまま適用することは，シーケンス制御系のもつ特殊性から，実際に多くシーケンス制御系に適用さ

* 精密工学科

れている直観と視察による設計方法にくらべて，冗長かつ複雑に過ぎ，はなはだしく実際的ではない。

本研究では，有限オートマトンとしてのシーケンス制御系の検討を行ない，その結果として，シーケンス制御系のもつ特殊性を考慮して有限オートマトンの設計理論の簡略化を行ない，かつそれによるシーケンス制御系の体系的な論理設計法の確立を試みた。

本研究では，一貫して次の可識別性の原則に立ち，簡略化された遷移表による順序論理回路の設計法，及びそれにより得られる組合せ論理回路の簡単化を論ずる。

可識別性の原則

論理回路に要求される条件は，出力がオン（動作状態）であるべき入力条件と，出力がオフ（不動作状態）であるべき入力条件とを区別し得ることである。

2. 一般的考察

有限オートマトンの一種としてのシーケンス制御系は，系統的に分析すれば，一般に Fig. 1 のように構成されるものとみなすことができる。即ち，普通の意味でのシーケンス制御装置は，p 個の押しボタンその他の入力装置からの信号，及び制御対象に付けられた r 個の検出端からの信号の合計 $p+r$ の入力変数を持ち，また q 個の操作端への操作信号を出力として持つ。また，制御対象は操作端の動作を入力とし，制御対象の実際の挙動を出力とするが，これ等の出力は各検出端を通じて制御装置へフィードバックされるものであって，系全体としてディジタル信号を含む一種のフィードバック系と見られる。

ここで，系のシーケンスは次のように進行する。まず押しボタン等の入力装置から入力が与えられると，制御装置はその入力を判別し，かつ検出端を通じて検知される制御対象の状態と照合し，それ等の条件によって行なうべき操作を撰択する。一方，制御対象はその操作を受

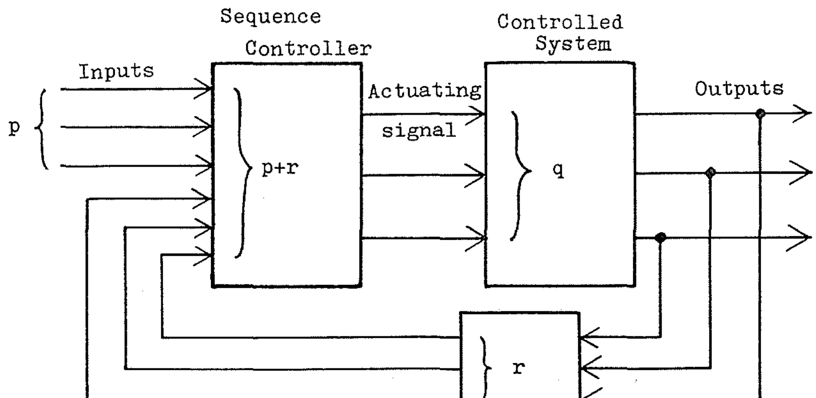

Fig. 1. Block Diagram of Sequencial Control System
Sequence controller

けることにより，次第にその状態を変化させるが，操作は制御対象の状態の変化が一定の条件に達するまで続けられる。これ等制御対象の状態の変化はまた，検出端を通じて制御装置に知らされ，その変化が一定の条件に達すると制御装置は次に行なうべき操作を撰択し，シーケンスが進行する。ここで特徴的なことは，シーケンスの進行は制御装置単独では遂行されず，制御対象自体がその一部分を分担していることである。即ち，制御対象は検出端と一体として考えるとき，それ自体で一種の記憶装置或いはタイマを含む論理回路の役割りをはたしていることである。このことにより，シーケンス制御系は，通常の記憶・論理回路に加えて，制御対象と云う特殊な記憶・論理回路を有する有限オートマトンであると云うことができる。そしてその中のシーケンス制御装置は，制御対象その他から与えられる一定のシーケンスに対し，出力シーケンスとして操作のシーケンスを与える有限オートマトンである。従って制御装置の設計においては，検出端の種類・位置等の制約を含めて，制御対象の記憶・論理回路としての性質，及び安定性等の解析が必要とされる。このことは，制御装置が，ある限られた入力シーケンスに対し，ある限られた出力シーケンスを与える有限オートマトンであることを意味し，そのためシーケンス制御装置は，一般の有限オートマトンにくらべて，冗長であるのが普通であるが，一方それ故に，設計理論が簡略化される可能性が含まれているとも云える。

以下これ等について，例を用いて考察を進めることにする。Fig. 2 は冷却槽のシーケンス制御の例であって，簡単でかつ一般性をもつものと考えられるので，原明弘氏の「シーケンス制御の設計」(オートメーション) から引用させていただいた。

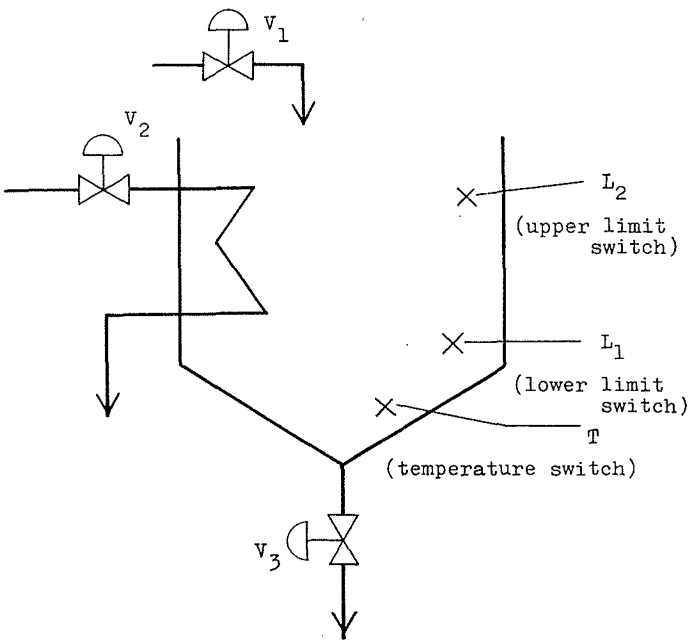

Fig. 2.　Schematic Flow Diagram of Cooling Tank

Table 1. Input-Output Tapes of Cooling Tank Sequence

a)

Steps	1	2	2′	3	4	5	6
Inputs L_1 L_2 T	0 0 0	1 0 0	0 0 1	1 0 1	1 1 1	1 1 0	1 0 0
Outputs V_1 V_2 V_3	1 0 0	1 0 0	1 0 0	1 0 0	0 1 0	0 0 1	0 0 1
Operations	Receiving. Operation				Cooling Opr.	Send-off Opr.	

b)

Steps	1	2	2′	3	4	5	6	7	8
Inputs L_1 L_2 T V_3	0 0 0 0	1 0 0 0	0 0 1 0	1 0 1 0	1 1 1 0	1 1 0 0	1 1 0 1	1 0 0 1	0 0 0 1
Outputs V_1 V_2 V_3	1 0 0	1 0 0	1 0 0	1 0 0	0 1 0	0 0 1	0 0 1	0 0 1	1 0 0
Operations	Receiving Operation				Cooling Opr.	Send-off Opr.			Re-ceiving Opr.

シーケンス動作は次の通りであるとする。

1) 液受け操作――バルブ V_1 を開け，液面上限スイッチ L_2 がオンになるまで前工程からの液を受け，オンになったら次の冷却操作に移る。この操作中 V_2, V_3 は閉。

2) 冷却操作――バルブ V_2 を開け，温度スイッチ T がオフになるまで冷却し，オフになったら排出操作に移る。この操作中 V_1, V_3 閉。

3) 排出操作――バルブ V_3 を開け，液面下限スイッチ L_1 がオフになるまで液を排出し，オフになったら始めの液受け操作にもどる。この操作中 V_1, V_2 は閉。

いま，すべてが正常に動作するものとして，制御装置の入出力のシーケンスの対を考えると，Tab. 1 a のようになる。ここで入力の可能な組合せは 7 通り，出力については 3 通りであり，6 ステップでシーケンスが一巡する。ただし，最初のステップ 1 から次のステップに移るときには，ステップ 2 及びステップ 2′ の 2 通りが考えられ，そのいずれも起り得るものとする。ここで注目すべきことは，ステップ 2 とステップ 6 において，同一の入力にもかかわらず出力が異なることである。このことは，検出端 L_1, L_2, T からの信号だけでは，3 つの操作を区別するのに十分なだけの情報が得られないことを意味している。従って，この場合は制御装置の内部（内部記憶）或いは外部（検出器）の変数を，少なくとも 1 個以上ふやさなければならない。Tab. 1 b は，外部変数として，バルブ V_3 にリミットスイッチを付加した場合の入出力テープである。このようにすると入出力の関係が多対 1 対応になって，一組の入力に対して必らずただ一組だけの出力が対応する。また，制御装置に内部記憶を付加する例としては，内部状態変数を 1 個 (S) 付加した場合を Tab. 2 に，また内部状態変数を 2 個 (S_1, S_2) 付加した場合を Tab. 3 に示す。それぞれ，a 表は入出力表であり，b 表は状態遷移表である。Tab. 2 の場合は前述の可識別性の原則に対する必要最小限度のものであり，[illegible] 1b の例における外部変数 V_3

Table 2.

a) Output Table

		Inputs (L_1 L_2 T)					
		0 0 0	1 0 0	0 0 1	1 0 1	1 1 1	1 1 0
States (S)	0	1 0 0	1 0 0	1 0 0	1 0 0	—	—
	1	—	0 0 1	—	—	0 1 0	0 0 1

Outputs (V_1 V_2 V_3)

b) State Table

		Inputs (L_1 L_2 T)					
		0 0 0	1 0 0	0 0 1	1 0 1	1 1 1	1 1 0
States $(S)^n$	0	0	0	0	0	1	—
	1	0	1	—	—	1	1

States $(S)^{n+1}$

Table 3.

a) Output Table

		Inputs (L_1 L_2 T)					
		0 0 0	1 0 0	0 0 1	1 0 1	1 1 1	1 1 0
States (S_1 S_2)	0 0	0 0 0	—	—	—	—	—
	1 0	1 0 0	1 0 0	1 0 0	1 0 0	—	—
	1 1	—	—	—	—	0 1 0	—
	0 1	—	0 0 1	—	—	—	0 0 1

Outputs (V_1 V_2 V_3)

b) State Table

		Inputs (L_1 L_2 T)					
		0 0 0	1 0 0	0 0 1	1 0 1	1 1 1	1 1 0
States $(S_1\ S_2)^n$	0 0	1 0	—	—	—	—	—
	1 0	1 0	1 0	1 0	1 0	1 1	—
	1 1	—	—	—	—	1 1	0 1
	0 1	0 0	0 1	—	—	—	0 1

States $(S_1\ S_2)^{n+1}$

Table 4.

Steps	1	2	3	4
Inputs A B	0 0	0 1	1 1	1 0
Outputs V_1 V_2 V_3	1 0 0	1 0 0	0 1 0	0 0 1
Operations	Receiving Opr.		Cooling Opr.	Send-off Opr.

に対応して，内部状態変数 S がステップ2とステップ6を区別している。これに対して Tab. 3 の場合には，冗長度は増すが各々の操作に対して制御装置の内部状態が1対1に対応している点で，それぞれ特徴を持っている。一方，逆に3通りの操作を区別できればよいと云うだけであれば，Tab. 4 の例のように2個の入力変数のみで十分なはずであるが，しかし，この場合には，そのために A, B のように制御対象のシーケンス動作に対しやや特殊な検出端等が要求されることが多い。

次に同一の出力に対応する入力について考察すると Tab. 1 の a の例では，ステップ 2, 2′ 及び3では，変数 L_2 が0であることだけが必要とされ，あとの情報は冗長である。またステップ5及び6では変数 L_1 及び T がそれぞれ1及び0であればよい。従って，これ等のことを考慮すると，Tab. 2 b 及び Tab. 3 b はそれぞれTab. 5 a 及び b の簡略化された遷移表に置き換えることができる。

Table 5.

a) Equivalent State Table of Tab. 2 b

		Inputs (L_1 L_2 T)			
		0 0 0	- 0 -	1 1 1	1 - 0
States $(S)^n$	0	0	0	1	—
	1	0	—	1	1

States $(S)^{n+1}$

b) Equivalent State Table of Tab. 3 b

		Inputs (L_1 L_2 T)				
		0 0 0	- 0 -	1 1 1	1 1 0	1 - 0
States $(S_1\ S_2)^n$	0 0	1 0	—	—	—	—
	1 0	—	1 0	1 1	—	—
	1 1	—	—	1 1	0 1	—
	0 1	0 0	—	—	—	0 1

States $(S_1\ S_2)^{n+1}$

以上の考察の結果，シーケンス制御系の特徴として，次の4点があげられる。

1) 制御対象それ自体が記憶・論理の機能を有している。

そのため検出端を通じてシーケンス制御装置に入る入力信号については，

2) シーケンスの経過中，一度も現われない入力信号の組合せ (inhibited condition) が多く存在する。

3) シーケンスの経過の中で現われるもののうち，直接シーケンスの進行には関係しない入力信号の組合せ (indifferent condition) が多く存在する。

かつまた，制御装置については，

4) 内部状態変数の数が一義的ではない。

2), 3) の性質から，シーケンス制御装置は必然的に扱う変数が多くなる。2) の inhibited condition は正常なシーケンス動作には無関係であるが，故障検知等において重要である。3) の indifferent condition については，シーケンス制御装置はこれにまったく無関心であってよく，従ってこの条件は設計の際に，あらかじめ取り除いても差支えない。筆者等は，主としてこれ等の条件に注目して設計法の簡略化を行なった。

3. シーケンス制御装置の論理設計

3-1 概　　説

シーケンス制御装置は，Fig. 3 に示すように，一般に順序制御回路と操作回路の 2 つの部分に分けられる。順序制御回路は，制御装置自体のシーケンスを制御する回路であり，これはまた，記憶回路と組合せ論理回路とに分けられる。記憶回路には，遅延素子或いはフリップ・フロップ等の記憶素子が用いられ，順序制御回路の内部状態の保持・記憶に用いられる。組合せ論理回路は内部状態の制御に用いられるもので，この回路の入力は，押しボタン等の入力装置からの外部入力信号，内部状態変数の現在の値，及び検出端からの出力信号，即ち外部状態変数の現在値であり，出力は内部状態変数の次の状態における値である。この回路は一般に信号だけを扱うものであり，その論理は記憶回路に用いられる記憶素子に依存する。

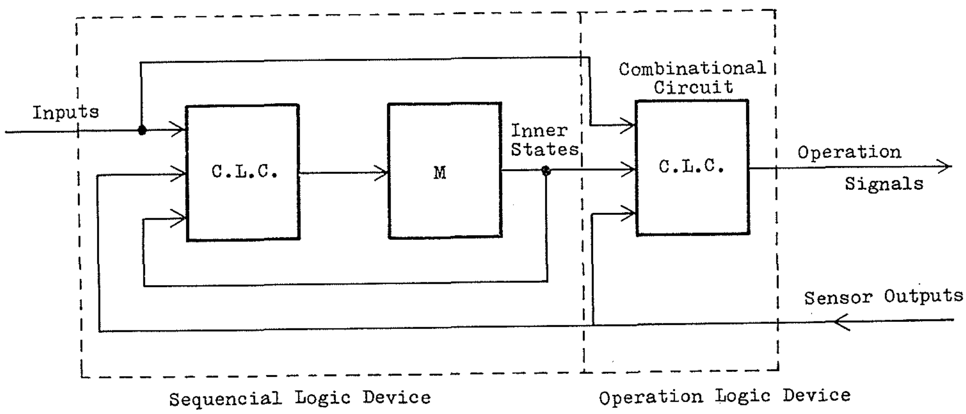

Fig. 3. Schematic Block Diagram of Sequence Controller
C. L. C.: Combinational logical circuit　M: Memory

また操作回路は，直接操作端に操作信号を送る回路で，いわば出力回路とも云うべきものであって，一般にパワーレベルの高い論理素子による組合せ論理回路である。この回路への入力は順序制御回路と同様，外部入力装置からの外部入力信号，順序制御回路の内部状態信号，及び検出端からの外部状態信号であり，出力は操作端への操作信号である。

前節でも述べた如く，制御対象自体が記憶・論理の機能を有するため，検出端の種類・数量・及びその配置等に制約がなければ，内部記憶は必らずしも必要ではない。しかしながら，シーケンスの進行のみに関してはそれで十分であっても，一般にシーケンス制御装置は，シーケンスの監視・故障時の処置等の，運転上或いは保安上等の仕様も満たさなければならず，従って内部記憶を有するべきか否か，またその数量等は，一義的ではないにせよ，仕様により決定されるべきものである。ただし，ここで重要なことは，操作回路において，各操作（出力変数値の組合せ）に対応する入力条件（入力変数値の組合せ）が互に重複しないだけの入力条件が

得られることである。

ここでは，状態変数の数はすでに決められているものとして，状態変数の定義，及び操作回路の設計法について簡単に述べ，さらに順序制御回路の設計法，特に遷移表の簡略化について述べることにする。

3-2 状態変数の定義及び操作回路の設計

一般にシーケンス制御系はかなり苛酷な条件の下で作動しなければならないことが多く，そのため，ノイズ等に対する安定性から，非同期式の回路が多く用いられる。非同期式では，まず hazard 及び race condition[4] が問題となるが，ここではそれに関する詳論は他の機会にゆずることにして，次の点に留意しながら，内部状態変数及びそのシーケンスを定義するのが得策であることを主張するにとどめる。即ち，ある状態から他の状態へ遷移するときには，常に唯一つの状態変数のみが変化するように，内部状態変数，及びそのシーケンスを定義する。

このことにより，少なくとも race condition はさけることができるが，hazard についてはその限りではない。例えば，Fig. 4 の例のように同一の入力条件の下で2段に遷移するときは，上の原則に従っても hazard がさけられない場合がある[註)]。しかしまた，この場合 continuous 接点或いは make-before-brake 接点等の特殊接点を用いることにより，hazard, race condition

States $(R_1R_2)^n$	Inputs AB: 0 0	1 0	0 1
0 0	0 0	1 0	0 0
1 0	1 0	1 0	1 1
1 1			0 1
0 1	0 0		0 1
	States $(R_1R_2)^{n+1}$		

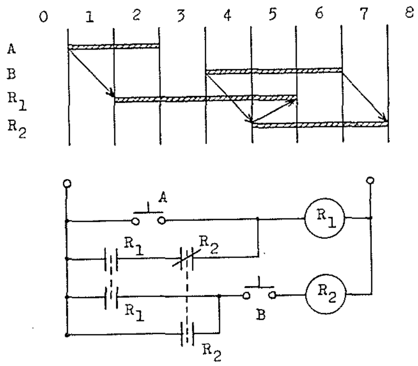

Fig. 4. Example of Hazard

Table 6. Truth Table for Operation Circuit.

a)

Inputs S_1	S_2	Outputs V_1	V_2	V_3	
0	0	0	0	0	
1	0	1	0	0	…Receiving opr.…
1	1	0	1	0	…Cooling opr.……
0	1	0	0	1	…Send-off opr.……

b)

	Inputs L_2	T	S_1	S_2	Opututs V_1	V_2	V_3
	—	—	0	0	0	0	0
	0	—	1	0	1	0	0
	—	—	1	1	0	1	0
	—	0	0	1	0	0	1
Inter-lock ………	1	—	—	—	0	—	—
	—	1	—	—	—	—	0

ともにさけることができる。

次に操作回路については，前述の如く，入力変数は押しボタン等の外部入力変数，順序制御回路の内部状態変数，及び検出端からの外部状態変数の三者であるが，出力を区別し得るためには，必ずしもその全部の入力変数を必要としない。例えば，内部状態と操作とが1対1対応をしていれば，もはや他の入力変数は必要ではなく，状態変数と操作変数とをそれぞれ入力及び出力とする真理値表を与えれば十分である。Tab. 6 a はその一例であり，前節の冷却槽のシーケンスの例において2個の内部状態変数を定義した場合であって，Tab. 3 a に対応している。

しかしながら，さらに，これに保安その他のためのインターロック等を考慮しなければならない場合には，そのインターロック等の条件を付加する。Tab. 6 b はその例である。ここでは，Tab. 6 a の真理値表で与えられる操作条件に加えて，内部状態変数を定義する記憶素子に故障があった場合の事故防止のために，液受け操作に対しては上限リミットスイッチ L_2 が動作していないこと，及び排出操作に対しては温度スイッチ T がオフであること等が，インターロック条件として付加されている。従って，真理値表では外部状態変数 L_2 及び T が入力変数として加えられているが，この場合，例えば入力変数 L_2 は，液受け操作 (出力 100) だけに関係して他の操作時には無関係であるので，液受け操作に対してのみ値 0 を入れ，他の操作に対する欄は空欄にする。これと同時に，変数 L_2 が 1 のときには無条件に出力変数 V_1 を 0 にするので，変数 L_2 が 1 のときは，他の入力変数の欄を空欄にし，また対応する出力変数も，V_1 のみが 0 で他の出力変数はこの条件のみでは決まらないので空欄となる。表中空欄には bar が入れられている。

以上，操作回路の論理設計に関しては，次のようにまとめることができる。まず真理値表の各変数については，

1) 少なくとも各出力に対し，それに対応する入力条件が互に重複しないようにとれるだけの入力変数を選ぶ。

2) ある出力に対し，それに無関係な入力変数の欄は空欄にする。

3) ある入力条件に対し，それだけでは決まらない出力変数の欄は空欄にする。

さらに，これ等の原則に従って，

4) まず，インターロックその他を考えない基本出力だけについて，それ等の出力を区別し得る最小限の真理値表を作成する。

5) 次に，インターロックその他の条件を付加する。このとき，これ等のインターロック条件とその他の入力条件が互に矛盾しないように，上の基本真理値表の部分にも修正が加えられる。

3·3 順序制御回路の設計

さて，次に順序制御回路は，前述の如く押しボタン等の外部入力信号，及び検出端を通じ

て知らされる制御対象の状態(外部状態)を入力として，制御装置自体のシーケンスを制御する一種の有限オートマトンである。従って，この順序制御回路の設計は，一般の有限オートマトンと同様に状態遷移表を用いて行なうことにするが,ここで先きに述べた indifferent condition 及びインターロックその他の付加条件に関係しない inhibited condition 等が非常に多く存在していることに注目し，これ等をあらかじめ取り除くようにすると，次に述べる簡略化された遷移表が得られる。

一般にシーケンス制御系における順序制御回路では，その内部状態の遷移を表す遷移表は，次の2つの条件を表記するものであると云える。即ち，

1) ある状態において，他の状態からその状態へ遷移する条件

2) ある状態において，その状態を保持するための条件

1)は，シーケンス制御系においては必ずサイクリックに制御が行なわれるため，各々の状態に対して，少なくとも1つはこの条件が必要である。また2)に関しては，同一の入力条件に対して数段にわたって遷移させることもあり得るので，必らずしも各状態のそれぞれに対して必要な条件であるとはかぎらないが，この条件をもつ状態が少なくとも1つは存在しなければ安定な順序制御回路は実現できない。また，同一の入力条件に対して数段にわたって状態の遷移が行なわれる場合には，前述の如く hazard に注意しなければならず，そのためのしかるべき工夫が必要である。

ここで，上の2)の条件について，前述の indifferent condition 及び inhibited condition に注目すると，簡略化された遷移表は次のようにして作られる。まず，遷移表は入力条件の欄，状態変数の現在の値の欄，及び状態変数の次の状態における値の欄の3つの欄で構成されるが，初めに入力条件の欄について次の規則に従うものとする。

規則 1　ある状態において，その状態から他の状態へ遷移するときには，その遷移に本質的に関係する入力変数の値だけを定義して，他の入力変数の欄は空欄とする。(例では，空欄の代りに bar を入れ，その変数の値の空欄の個所を示している。なおこの空欄の意味は，その個所における変数の値は，1であっても0であっても差支えないとの意味である)。

規則 2　ある状態において，その状態を保持するときには，その状態の保持に本質的な入力変数の値だけを定義し，他は空欄とする。

また，状態変数の現在の値の欄及び次の状態における値の欄については，

規則 3　ある状態及び入力条件に対して，ある状態変数の次の状態における値が定義される(保持を含む)ときは，その状態変数の定義に本質的な状態変数の現在の状態における値だけを定義して，他の状態変数については空欄とする。

規則 4　ある状態及び入力条件に対して，それ等によって定義される次の状態における状態変数の値が，シーケンスの進行に本質的ではないとき，及びすでに他の条件によって同等のものが定義されていることが明らかなときは空欄とする。

規則 5　ある状態及び入力条件に対して，それ等だけでは次の状態における値が定義されない状態変数は空欄とする。

ここで云う状態とは，ただ1個の状態をさすだけでなく，状態群をもさしている。例えば，状態変数の欄に空欄のある状態はその例である。また，このようにして作られる遷移表では，矛盾を生じないようにすることが重要である。

これ等簡略化された遷移表に関する諸規則を用いて，前述の冷却槽のシーケンスの遷移表を作ると Tab. 7 が得られる。Tab. 7 において，a 表はすべてが正常に動作することを仮定した場合であり，基本的なシーケンスのみを表わして居り，b 表はそれに多少のインターロックが付加されている場合である。いずれの場合も内部状態は，前述の Tab. 6 の a 表或いは b 表(これ等はいずれも同じで，ただ操作出力との関係だけが異なっている) で定義されるものとする。

Table 7.　Simplified Flow Tables of Cooling Tank Process.

a)

		Inputs (L_1 L_2 T)					
		0 - -	- 0 -	- 1 -	- - 1	- - 0	1 - -
States (S_1 S_2)^n	0 0	1 0	—	—	—	—	—
	1 0	—	1 0	1 1	—	—	—
	1 1	—	—	—	1 1	0 1	—
	0 1	0 0	—	—	—	—	0 1

States (S_1 S_2)$^{n+1}$

b)

		Inputs (L_1 L_2 T)					
		0 0 -	- 0 -	1 1 -	1 - 1	1 - 0	0 1 -
States (S_1 S_2)^n	0 0	1 0	—	—	—	—	—
	1 0	—	1 0	1 1	—	—	—
	1 1	—	—	—	1 1	0 1	—
	0 1	0 0	—	—	—	0 1	—
	- -	—	—	—	1 -	—	0 0

States (S_1 S_2)$^{n+1}$

まず a 表において，すべてが正常に動作するものと仮定すれば，状態 (1 0) (液受け操作) に遷移するためには，ここでは一旦休止の状態 (0 0) を経るが，状態 (0 1) (排出操作) において，下限リミットスイッチ L_1 がオフ即ち変数 L_1 が 0 になればよく，またこの状態 (1 0) の保持には上限リミットスイッチ L_2 がオフでありさえすればよい。従って，遷移表では入力 (0 - -) に対して，状態 (0 1) は (0 0) に，(0 0) は (1 0) にそれぞれ遷移すればよく，入力 (- 0 -) のときの状態 (1 0) に対して，次の状態は (1 0) となる。またその次の状態 (1 1) に対しては，(1 0) からの遷移には上限リミットスイッチ L_2 が，また保持には温度ス

イッチ T のみが本質的である。以下同様にして全シーケンスが定義される。

また b 表においては，上限リミットスイッチ L_2 がオンで，かつ下限リミットスイッチ L_1 がオフであることは故障時以外にはあり得ず，従ってこの場合は，停止することがインターロックの条件として与えられているものとすると，そのため状態（１０）への遷移条件は，インターロックとの矛盾をさけるため，状態（００）において入力が（００－）であることである。一方，状態（１０）を保持する条件（－０－）は，インターロック条件（０１－）と矛盾しないため変らない。また，現在の状態の欄に記入されている状態（－－）はすべての状態と云うことであり，ここではインターロック条件（０１－）がすべての状態に対して，停止するように定義されていることを意味する。なお，この遷移表にはもう一つのインターロック条件（１－１）が定義されている。この条件（１－１）の方は，正常な動作中にも現われる入力条件であるが，状態（０１）(排出操作）のときには，温度スイッチ T は必らずオフであり，もしこの温度スイッチ T がオフでなければ再び冷却操作にもどることを主張している。この場合，状態（－－）に対しては入力条件（１－１）だけでは状態変数 S_2 の値が定義できないので，上の規則５によって空欄となっている。

以上のようにして作られた遷移表は，順序制御回路全体としての順序動作を示す論理であって，前述の如く順序制御回路には記憶回路とそれを制御する組合せ論理回路があるので，この遷移表の論理は，次に個々の記憶回路及び組合せ論理回路の論理に変換されなければならない。しかし，記憶回路の論理は一般に用いる記憶素子に固有なものであり，ここで主要な問題はその記憶回路を制御する組合せ論理回路の論理である。

組合せ論理回路の論理は，用いられる記憶回路の性質によって異なるが，遷移表からその論理を求める基本的な方法は次の通りである。

まず，遷移表における入力条件及び現在の状態における状態変数の値を入力として，それに対応する次の状態における状態変数の値を出力とする組合せ論理回路の真理値表を作成する。これは，記憶素子が１ビットの遅延素子であるときの組合せ論理回路であって，リレー等を用いるときにはこの真理値表が対応する。Tab. 8 の出力 I に対する真理値表はその例である。しかしながら，その他の記憶回路を用いるときには，その用いる記憶回路の性質にあわせるために，さらに変換する必要がある。Tab. 8 の a 表における出力 II に対する真理値表は，記憶素子に S-R フリップ・フロップを用いたときの例であって，b 表の S-R フリップ・フロップの性質にあわせるために，c 表の関係を用いてさきの出力 I に対する真理値表を変換したものである。なお c 表の変換表は S-R フリップ・フロップに固有なものであり，他の記憶素子に対してはその記憶素子に固有な変換表が存在する。

4. 組合せ論理回路の簡単化

すでに見て来たように，本研究の方法による，シーケンス制御系の論理設計において多く

Table 8.

a) Truth Table derived from Tba. 7 b

Inputs					Outputs I		Outputs II			
L_1	L_2	T	S_1^n	S_2^n	S_1^{n+1}	S_2^{n+2}	S_{1s}	S_{1r}	S_{2s}	S_{2r}
0	0	—	0	0	1	0	1	0	0	1
0	0	—	0	1	0	0	0	—	0	1
—	0	—	1	0	1	0	—	1	0	—
1	1	—	1	0	1	1	—	1	1	0
1	—	1	1	1	1	1	—	1	—	0
1	—	1	—	—	1	—	1	0	—	—
1	—	0	1	1	0	1	0	1	—	0
1	—	0	0	1	0	1	0	—	—	0
0	1	—	—	—	0	0	0	1	0	1

b) Characteristics of S–R Flip-flop

		Inputs (S_s S_r)			
		0 1	0 -	1 0	- 0
States $(S)^n$	0	0	0	1	—
	1	0	—	1	1

States $(S)^{n+1}$

c)

S^n	S^{n+1}	S_s	S_r
0	0	0	—
0	1	0	1
1	0	1	0
1	1	—	0

みられる組合せ論理回路は，次の性質を持っている。

まず，入出力の条件については，

1) 入出力変数が非常に多い。

2) don't-care-condition が非常に多い。

また，与えられる真理値表に関しては，

3) 完全真理値表が与えられるのではなく，出力変数の値が1であるための入力条件 (on-condition) 及び出力変数の値が0であるための入力条件 (off-condition) のみであり，さらにそれ等は noncanonical な形で与えられる。

以上のような性質を持つ組合せ論理回路の簡単化の問題に対しては，Gavrilov の minimal term の概念[2] を用いる方法がもっとも適して居ること及びその実際的な計算の手法等に関しては，すでに報告[5] した通りであるので，ここでは，この minimal term による組合せ論理回路の簡単化の進め方を，実例にそって簡単に説明することにする。

初めに，minimal term は次のように定義される。

〈定義〉 minimal term：ある on-condition に関する minimal term とは，i) その on-condition に対して値1をとり，すべての off-condition に対して値0をとる入力変数の一次組合せ論理（これを on-term と呼ぶことにする）であって，ii) その変数のうちどの1つが無くても，もはや i) の条件，即ち on-term である条件が成立しないものを云う。

これは，論理回路においては on-condition と off-condition の区別さえつけばよい，と云うことを主張するものであって，minimal term はその最小形と云うことである。(on, off を入れかえると，同様にして off-condition に関する minimal term が得られる。) そしてさらに，この minimal term に関して次の Gavrilov の定理が成立し，その意義を明らかにしている。

〈**定理**〉 組合せ論理回路における，冗長でない論理関数は minimal term のみで構成される。

即ち，この定理は Quine の定理[7] と等価であって，この minimal term は Quine の方法における prime implicant に対応している。

さて，いま Tab. 9 の a 表のような真理値表が与えられたとして，その簡単化を行なうことにする。初めに，minimal term を求めるためには，on-condition と off-condition の比較を行なわなければならないので，便宜のため与えられた a 表の真理値表を b 表のような on-off-condition 表になおす。ここで横線より上は on-condition，及び下は off-condition である。そしてそれぞれの on-condition について，その各々の on-condition とすべての off-condition と

Table 9. Example of Simplification

a) Truth Table

Inputs					Outputs
L_1	L_2	T	S_1	S_2	S
0	0	—	0	0	1
0	0	—	0	1	0
—	0	—	1	0	1
1	1	—	1	0	1
1	—	1	—	—	1
1	—	0	1	1	0
1	—	0	0	1	0
0	1	—	—	—	0

b) On-off-condition Table

	L_2	L_2	T	S_1	S_2
On-conditions	0	0	—	0	0
	—	0	—	1	0
	1	1	—	1	0
	1	—	1	—	—
Off-conditions	0	0	—	0	1
	1	—	0	1	1
	1	—	0	0	1
	0	1	—	—	—

c)

	L_1	L_2	T	S_1	S_2
On	0	0	—	0	0
Off	0	0	—	0	1
	1	—	0	1	1
	1	—	0	0	1
	0	1	—	—	—

d) Simpliffed On-off-conditions

	L_1	L_2	T	S_1	S_2
On	—	0	—	—	0
	1	—	—	—	0
	1	—	1	—	—
	0	—	—	—	1
	—	—	0	—	1
	0	1	—	—	—

$$S=\bar{L}_2\cdot S_2+L_1\cdot\bar{S}_2+L_1\cdot T \quad \cdots\cdots\text{from On-conditions}$$
$$=(L_1+\bar{S}_1)(T+S_2)(L_1+\bar{L}_2)\cdots\text{from Off-conditious}$$

を c 表のように比較して，その on-condition を満足し，かつすべての off-condition と区別し得る最小のものを見出だせばよい。例えば on-condition (0 0 - 0 0) の minimal term は (- 0 - - 0) になる。このとき空欄 (表中では bar) は 1, 0 いずれの値もとり得るものとする。d 表は b 表の各々の on-condition について，それぞれ minimal term を求めた結果であり，off-condition からも同様にして off-condition に対する minimal term が得られるので，それも d 表に付記した。そして，on-condition の結果からは and/or 形の論理回路が得られ，また off-condition からは or/and 形の論理回路が得られる。一般にこれ等は Bool 代数的に等価な回路であるとは限らず，従って最終的にはその両者のうち使われる論理素子の少ない方を用いればよい。

また，この例では on 或いは off の各入力条件に対して，それぞれただ 1 つの minimal term しか存在しなかったが，一般には 1 つの入力条件に対して，複数個の minimal term が存在するのが普通である。従って，組合せ論理回路の簡単化の一般的方法は，次の 3 つの手順に従っ

て行なわれる。

手順 1　与えられた真理値表の各 on-condition (或いは off-condition) から minimal term を求める。

手順 2　得られた minimal term から冗長ではない minimal term の組 (irredundant cover) を求める。

手順 3　手順 2 で求めた minimal term の組のうち，用いられる論理素子のもっとも少ない組を求める。

このようにして得られるものが，もっとも簡単な論理回路である。

5. 例　　題

上に述べた手法を用いて，前述の冷却槽のシーケンスについて行なった総括的な設計例を次に示す。

操作回路は，Tab. 6 の b に示されるものとまったく同様であって，Tab. 10 はその真理値表及び on-off-condition 表であり，その結果を Tab. 11 に示した。

順序制御回路については，Tab. 7 b の遷移表をもとにして，それに起動・停止の条件が付加されている。即ち，P_1, P_2 はともに押しボタンであって，P_1 は起動用，P_2 は停止用である。停止はいわゆる順序停止にしてあり，シーケンス動作の途中で停止条件が生じても，そのシー

Table 10. Truth Table for Operation Circuit and its On-off Condition Tables

Inputs				Outputs		
L_2	T	R_1	T_2	V_1	V_2	V_3
—	—	0	0	0	0	0
0	—	1	0	1	0	0
—	—	1	1	0	1	0
—	0	0	1	0	0	1
1	—	—	—	0	—	—
—	1	—	—	—	—	0

	V_1				V_2				V_3			
	L_2	T	R_1	R_2	L_2	T	R_1	R_2	L_2	T	R_1	R_2
On	0	—	1	0	—	—	1	1	—	0	0	1
Off	—	—	0	0	—	—	0	0	—	—	0	0
	—	—	1	1	0	—	1	0	0	—	1	0
	—	0	0	1	—	0	0	1	—	—	1	1
	1	—	—	—					—	1	—	—

Table 11. Simplified On-off-condition Tables for Operation Circuit

	V_1				V_2				V_3			
	L_2	T	R_1	R_2	L_2	T	R_1	R_2	L_2	T	R_1	R_2
On	0	—	1	0	—	—	1	1	—	0	0	1
Off	—	—	0	—	—	—	—	0	—	—	—	0
	—	—	—	1	—	—	0	—	—	—	1	—
	1	—	—	—					—	1	—	—

$V_1 = \bar{L}_2 \cdot R_1 \cdot \bar{R}_2$
$V_2 = R_1 \cdot R_2$
$V_3 = \bar{T} \cdot \bar{R}_1 \cdot R_2$

Table 12. Flow Table for Sequential Logic Circuit

Inputs $P_1P_2L_1L_2T$

States $(R_1R_2R_3R_4)^n$	0----	1000-	-000-	---0-	--11-	--1-1	--1-0	-0---	-1---	--01-
0000	000-	0010								
0010			1010							
1010				10--	11--					
11--						11--	01--			
01--			00--				01--			
--1-								--1-		
-10-			000-							
----									--0-	---1
---0						---0	---0			
---1			00-1							

Fundamental Cycle

Start-stop Conditions

in Accident

States $(R_1R_2R_3R_4)^{n+1}$

Table 13. Truth Table and its On-off-Condition Table for Sequential Logic Circuit

Inputs									Outputs			
P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4	R_1	R_2	R_3	R_4
0	—	—	—	—	0	0	0	0	0	0	0	—
1	0	0	0	—	0	0	0	0	0	0	1	0
—	0	0	0	—	0	0	1	0	1	0	1	0
—	0	0	0	—	0	1	—	—	0	0	—	—
—	0	0	0	—	—	1	0	—	0	0	0	—
—	0	0	0	—	—	—	—	1	0	0	—	1
—	—	—	0	—	1	0	1	0	1	0	—	—
—	—	1	1	—	1	0	1	0	1	1	—	—
—	—	1	—	1	1	1	—	—	1	1	—	—
—	—	1	—	1	—	—	—	0	—	—	—	0
—	—	1	—	0	1	1	—	—	0	1	—	—
—	—	1	—	0	0	1	—	—	0	1	—	—
—	—	1	—	0	—	—	—	0	—	—	—	0
—	0	—	—	—	—	—	1	—	—	—	1	—
—	1	—	—	—	—	—	—	—	—	—	0	—
—	—	0	1	—	—	—	—	—	—	—	—	1

	R_1									R_2								
	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4
On	—	0	0	0	—	0	0	1	0	—	—	1	1	—	1	0	1	0
	—	—	—	0	—	1	0	1	0	—	—	1	—	1	1	1	—	—
	—	—	1	1	—	1	0	1	0	—	—	1	—	0	1	1	—	—
	—	—	1	—	1	1	1	—	—	—	—	1	—	0	0	1	—	—
Off	0	—	—	—	—	0	0	0	0	0	—	—	—	—	0	0	0	0
	1	0	0	0	—	0	0	0	0	1	0	0	0	—	0	0	0	0
	—	0	0	0	—	0	1	—	—	—	0	0	0	—	0	0	1	0
	—	0	0	0	—	—	1	0	—	—	0	0	0	—	0	1	—	—
	—	0	0	0	—	—	—	—	1	—	0	0	0	—	—	1	0	—
	—	—	1	—	0	1	1	—	—	—	0	0	0	—	—	—	—	1
	—	—	1	—	0	0	1	—	—	—	—	—	0	—	1	0	1	0

	R_3								
	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4
On	1	0	0	0	—	0	0	0	0
	—	0	0	0	—	0	0	1	0
	—	0	—	—	—	—	—	1	—
Off	0	—	—	—	—	0	0	0	0
	—	0	0	0	—	—	1	0	—
	—	1	—	—	—	—	—	—	—

	R_4								
	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4
On	—	0	0	0	—	—	—	—	1
	—	—	0	1	—	—	—	—	—
Off	1	0	0	0	—	0	0	0	0
	—	0	0	0	—	0	0	1	0
	—	—	1	—	1	—	—	—	0
	—	—	1	—	0	—	—	—	0

Table 14. Simplified On-off-Condition Tables and its Resultants

	R_1									R_2								
	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4
On	—	—	—	—	—	—	0	1	0	—	—	—	1	—	1	—	—	—
	—	—	1	—	1	1	—	—	—	—	—	1	—	—	—	1	—	—
Off	—	—	—	—	—	0	—	0	—	—	—	—	—	—	0	0	—	—
	—	—	0	—	—	—	1	—	—	—	—	0	—	—	—	—	—	—
	—	—	0	—	—	—	—	—	1	—	—	—	0	—	—	0	—	—
	—	—	—	—	0	—	1	—	—									

	R_3									R_4								
	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4	P_1	P_2	L_1	L_2	T	R_1	R_2	R_3	R_4
On	1	0	—	—	—	—	0	—	—	—	—	—	—	—	—	—	—	1
	—	0	—	—	—	—	—	1	—	—	—	0	1	—	—	—	—	—
Off	0	—	—	—	—	—	—	0	—	—	—	—	0	—	—	0	—	—
	—	—	—	—	—	—	1	0	—	—	—	1	—	—	—	—	—	—
	—	1	—	—	—	—	—	—	—									

$R_{1c} = \bar{R}_2 \cdot R_3 \cdot \bar{R}_4 + L_1 \cdot T \cdot R_1 \cdots\cdots (R_1+R_3)\cdot(L_1+\bar{R}_2)\cdot(L_1+\bar{R}_4)\cdot(T+\bar{R}_2)$

$R_{2c} = L_2 \cdot R_1 + L_1 \cdot R_2 \cdots\cdots\cdots\cdots\cdots (R_1+R_2)\cdot L_1 \cdot (L_2+R_2)$

$R_{3c} = P_1 \cdot \bar{P}_2 \cdot \bar{R}_2 + \bar{P}_2 \cdot R_3 = \bar{P}_2 \cdot (P_1 \cdot \bar{R}_2 + R_3) \cdots\cdots (P_1+R_3)\cdot(\bar{R}_2+R_3)\cdot \bar{P}_2$

$R_{4c} = R_4 + \bar{L}_1 \cdot L_2 \cdots\cdots\cdots\cdots\cdots\cdots \bar{L}_1 \cdot (L_2+R_4)$

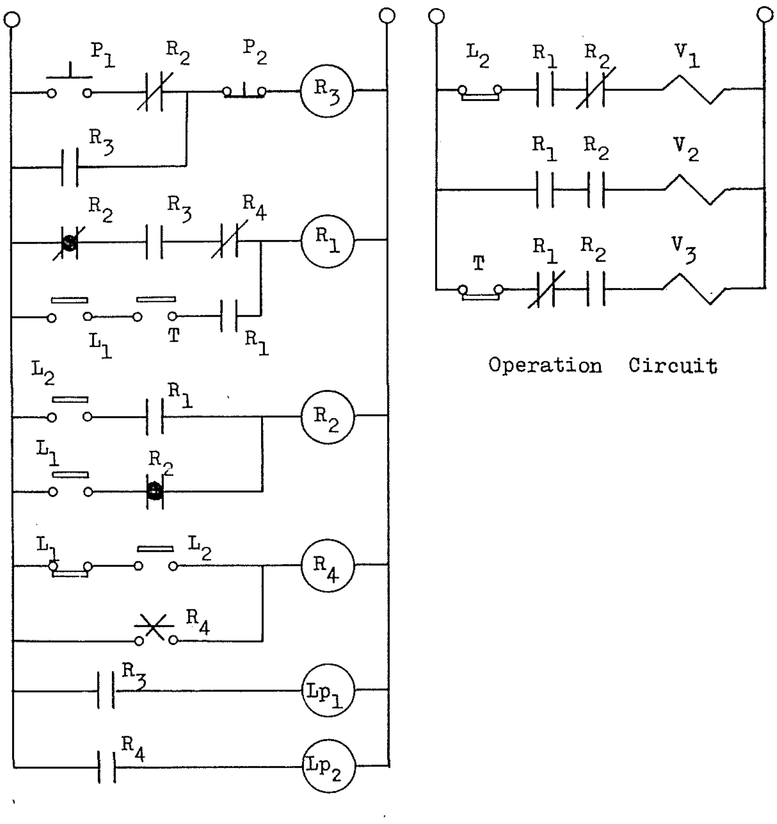

Fig. 5. An Example of Relay Circuit for Cooling Tank Process

ケンスの終了まで，操作は停止しない。次にリレー R_1, R_2 は各操作に対応するリレーであり，リレー R_3 は起動・停止の制御と動作中の指示，及びリレー R_4 は故障の指示と故障時の制御を行なう。また，故障時の制御は順序停止に準ずる。

これ等の動作は Tab. 12 の遷移表に示されており，Tab. 13 は Tab. 12 の遷移表によって得られる真理値表及びその on-off-condition 表であり，かつその結果は Tab. 14 で示されている。最終的に得られるリレー回路の接続図を Fig. 5 に示した。

6. 結　　言

以上，本研究では有限オートマトンとしてのシーケンス制御系の検討を行ない，その結果として，遷移表を簡単化する方法，及びそれを用いたシーケンス制御系の論理設計法を論じた。

これ等は，将来のシーケンス制御系の計算機による自動設計の問題等に重要な手がかりを与えるものと考える。

終りに，本研究に対して日ごろ熱心なご討論をいただいた北大田川遼三郎助教授，ならびに数々の有益なご助言をいただいた機械試験所研野和人博士に対し深謝の意を表します。

引 用 文 献

1) Aizerman, M. A. et al.: Automation and Remote Control, 21–3, 248–254 (1960).
2) Gavrilov, M. A.: Automation and Remote Control, 20–9, 1188–1207 (1960).
3) 原　明弘: オートメーション, 9–9, 55 (昭 39).
4) Huffman, D. A.: J. Franklin Institute, 257–3, 161–190 (1954).
5) 小山・田川・三浦: 第 4 回計測自動制御学会学術講演会予稿, No. 337 (昭 40).
6) Phister, Montogomery; 尾崎弘訳: ディジタル計算機の論理設計, 431 (昭 35), 朝倉書店.
7) Quine, W. V.: Amer. Math. Monthly, 59, 521–531 (1952).

註) リレー R_2 が動作するためにはリレー R_1 がオンでなければならないが，一方リレー R_1 はリレー R_2 が動作すると同時にオフになるので，このままではリレー R_2 が保持される前に動作回路が切れるため，結局リレー R_2 が動作しない可能性がある。この例ではリレー R_2 の接点を continuous 接点にすればこのことはさけられる。