

### Ⅲ. シーケンス制御基本動作表現へのフローチャートの適用と問題

本章では、シーケンス制御における基本動作と考えられるものを取り上げ、この動作の表現にフローチャートを適用し、表現の特徴と問題点を分析する。シーケンス制御の基本動作として何を選定するかは重要な問題で、本来ならばこの基本動作の選定、分類、整理作業に基づいて、本章の基本動作項目を決定すべきであるが、今はそれができない。

ここで取り上げる基本動作は、一応教育・訓練の現場で経験的に、シーケンス制御実習の基本課題として用いられているものの内から選んである。これに、電気学会のシーケンス制御調査専門委員会及びシーケンス制御工学体系化調査専門委員会の各報告書<sup>(18)</sup>の制御の分類及び制御遷移条件のパターン分類を参考に、大雑把に、条件制御、プログラム制御、並列プログラム制御、条件+プログラム制御に分類し、その中で代表的動作を取り上げた。それは次に示すとおりである。

1. 条件制御
  - 1.1 自己保持動作
    - (a) 停止優先
    - (b) 起動優先
  - 1.2 インタロック付 F F 動作
2. プログラム制御
  - 2.1 間隔動作
  - 2.2 遅延動作
  - 2.3 繰り返し動作
3. 並列プログラム制御  
並列動作
4. 条件+プログラム制御  
インタロック付限時遷移動作

さて、訓練内容として選定された各課題 —— つまり、上記の基本動作

の各項目 — についてのフローチャート表現の特徴と問題点を分析するにあたって、比較対象が必要となるが、それは次のようにする。まず、PC固有の事項については、ラダー方式PCのプログラミングと、また、フローチャート表現一般については、マイクロコンピュータにおけるデータ処理、技術計算等のプログラム表現と比較することにする。

次に、使用するフローチャート方式PCの命令一覧表を示す(表3.1)。命令は機能から大きく3つに分類した。なお、本方式PCの取扱説明書では、①データ転送・演算命令及び②分枝判断・行先命令の命令群を基本命令、③サブルーチン処理命令をグループ命令と呼んでいる。

表3.1 使用命令一覧

① データ転送・演算命令

ニーモニック	オペラント	説明
AND	n n n n	論理AND
AND・NOT	n n n n	論理ANDの内容否定
OUT	n n n n	出力
OUT・NOT	n n n n	出力内容の否定
TIM	1 n n n	限時機能設定
	#n n n n	限時時間設定値
CNR・TIM	1 n n n	限時機能解除

② 分枝判断・行先命令

JMP LBL	1 n n n	ジャンプ
CJP LBL	1 n n n	条件ジャンプ
WAIT		条件成立待ち
LBL	1 n n n	ラベル

③サブルーチン処理命令 (nn ≤ 32)

GJ		グループへのジャンプ
GS	00 n n	グループの起動
GOFF	00 n n	グループの停止
GN	00 n n	グループ番号
GE	00 n n	グループ終了

### Ⅲ. 1 条件制御

#### Ⅲ. 1.1 自己保持動作

##### (a) 停止優先自己保持動作

動作機能についての説明を図 3.1 に示すような回路 (a) 展開接続図、(b) 論理回路図) を示すことで行う。以後の各動作とも、回路が入出力両端からみて目的とする動作の機能を表わしているのので、回路図を示すことで動作機能説明の代りとし、文章及びタイムチャート等による詳細な説明は原則として行わないことにする。

図 3.2 に、図 3.1 の停止優先保持回路と同一の動作機能をもつフローチャート表現と P C のプログラムリストとを示す。これについては、以下の特徴と問題がある。

- (1) フローチャート表現で停止優先自己保持動作にする最も自然な方法は次のとおりである。停止の押ボタンスイッチ B が優先されるように、図 3.2 の押ボタンの状態判断 (◇印記号) 箇所が、A より位置的に先にくるようにする。これにより A と B の両方同時に押された場合、B を優先して出力を OFF にすることができる。しかし、これが全てではなく A の状態判断を B より先に置く、図 3.3 のような方法も知っておかねばならない。なぜなら、後で問題となるが、後者の方法で停止優先動作を作らなければならない場合が現われるからである。
- (2) 図 3.2、図 3.3 に見られるように、シーケンス制御用プログラムのフローチャートは、その構造が全体として、プログラム開始点 (ラベル L 0) への無限ループ構造になっていることである。これは以下の各動作についても同様であって、データ処理及び技術計算プログラムでは、むしろ特殊なことである。また、図 3.3 に見るように、ラベル L 1 へのループ構造があるが、このループの戻り点 (ラベル L 1) が、制御構造上重要な意味を持つ。これを図 3.1 と同様に、ラベル L 0 へ戻したのでは目的とする動作とはならない。
- (3) 入力側の停止用押ボタンスイッチ B は、通常は安全上の配慮から、図 3.4 のように、b 接点で制御システムへ受け渡すようにする。回路表現の場合、B が b 接点で与えられることの表現は、例えば、同図の入

力負論理のバッファ記号（ $\text{---}\triangleleft\text{---}$ ）でできる。しかし、フローチャート表現の場合適格な表現がない（ただ、プログラム上では、図 3.2 (b) のアドレス 0001 の AND 1 を AND・NOT 1 とすることで対応可能）。また、出力側についても、実際は出力点 X は 1 箇所であるのに、フローチャート表現では ON と OFF のそれぞれについて 2 箇所の出力表示をすることになる。他の例からみてもわかるとおり、一般にフローチャートでの入出力機器と制御システムとの結合性は悪くなると言える。

- (4) 回路表現（図 3.1）においては、この動作の基本要素である保持（ラッチ）を作ることが、一つのポイントになっている。また、回路動作の理解においても、同様に大きなウエイトを占める。ところで、コンピュータ応用のフローチャート表現では、図 3.2 (a) のような制御構造にすることで、起動の保持機構にしている。また、停止は、停止すべき出力を指定して出力 OFF 信号を送るという方法をとることになる。このため、回路表現で使う自己保持の作り方とは、本質的に異なる保持表現になることへの認識の転換が求められる。つまり、従来の回路表現に慣れている人がフローチャート表現の学習をする場合、既に持っている知識の否定・とらえ直しが必要となることである。

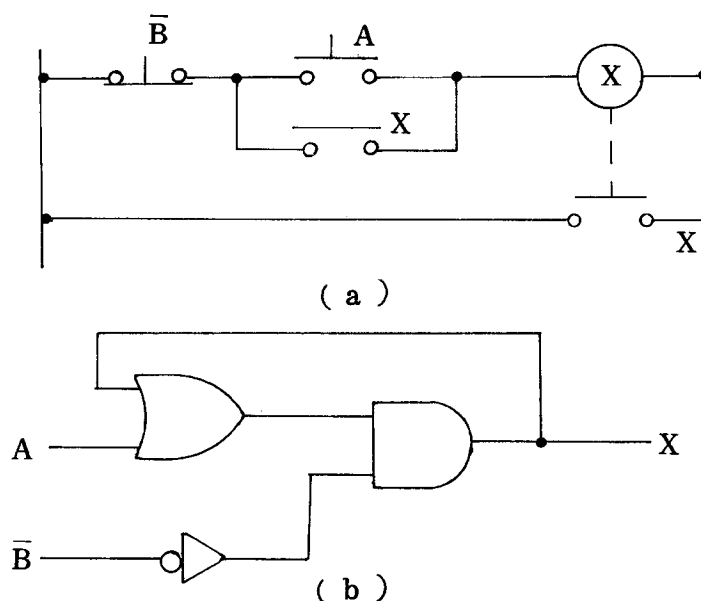
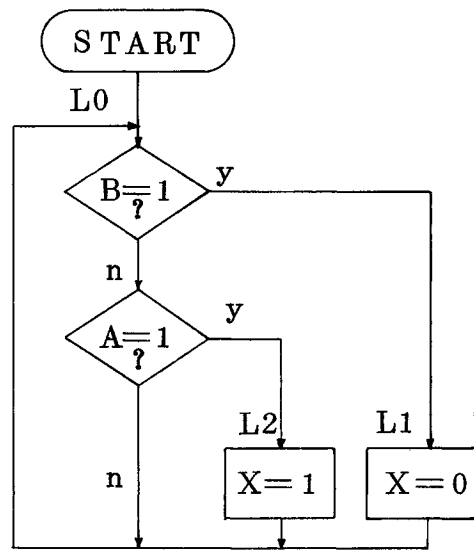


図 3.1 停止優先自己保持回路



(a)

(b) プログラムリスト

ADR.	LABEL	HNEMONIC	OPERAND	COMMENT
0000	LBL0:	AND	1	;(PB-B)
2		CJP	LBL1	
3		AND	0	;(PB-A)
4		CTP	LBL2	
5		JMP	LBL0	
6	LBL1:	OUT·NOT	200	;(X)
8		JMP	LBL0	
9	LBL2:	OUT	200	;(X)
10		JMP	LBL0	

図 3.2 停止優先自己保持動作(1)

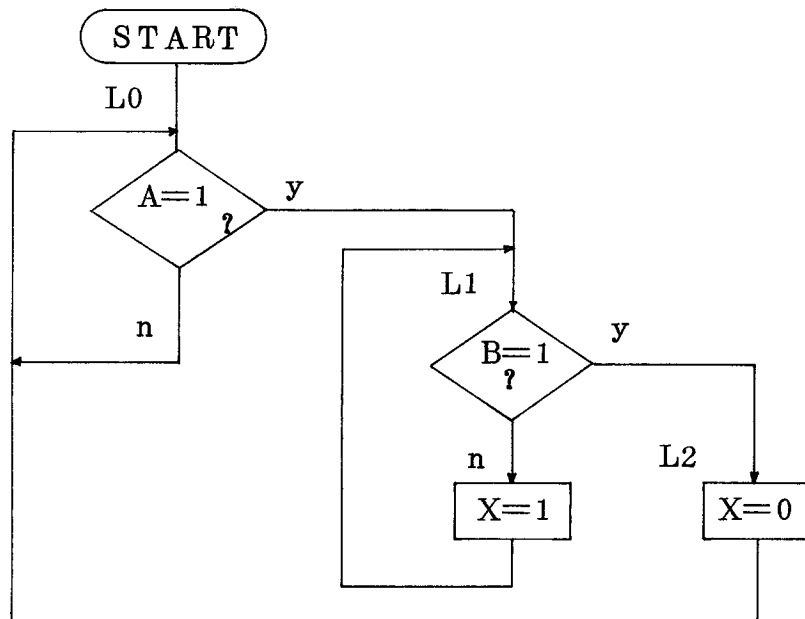


図 3.3 停止優先自己保持動作(2)

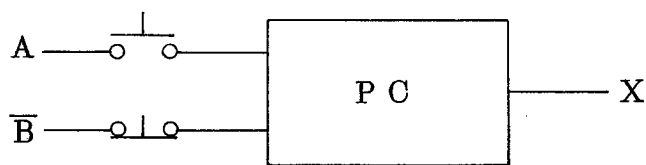
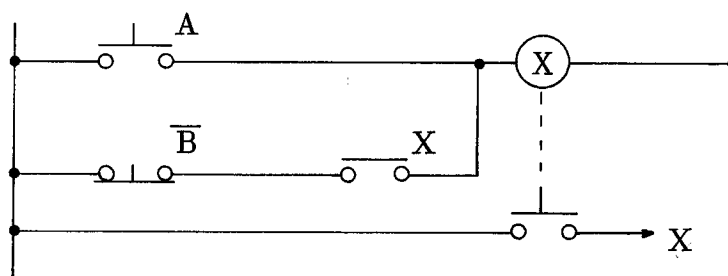


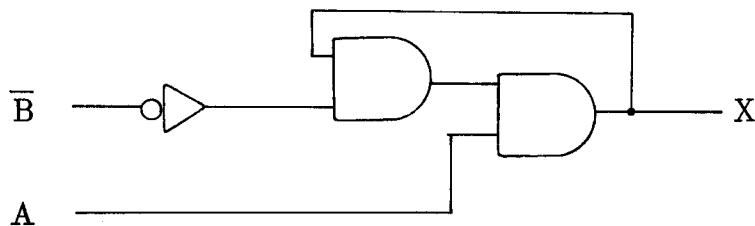
図 3.4 入力押ボタンスイッチの表現

(b) 起動優先自己保持動作

図 3.5 (a)、(b)に、本動作機能を有する回路を示す。図 3.6 は、図 3.5 と同一動作機能のフローチャート表現である。表現自体に格別の問題はなく、起動押ボタンスイッチ A の動作が優先されるように、位置的に A の判断を B の判断の前に置くことが自然であろう。ただ、先の(a)停止優先自己保持動作の表現(2) (図 3.3 ) と同一の形表現 (図 3.7 ) がとれないことに留意する必要がある。なぜならば、この表現では、押ボタンスイッチの操作に問題がでてしまうからである。即ち、このシステムに起動押ボタンスイッチ A から起動をかけようとする場合、必ず一旦停止用押ボタンスイッチ B を操作 ( 押 ) してからでないといけないという問題である。



(a)



(b)

図 3.5 起動優先自己保持回路

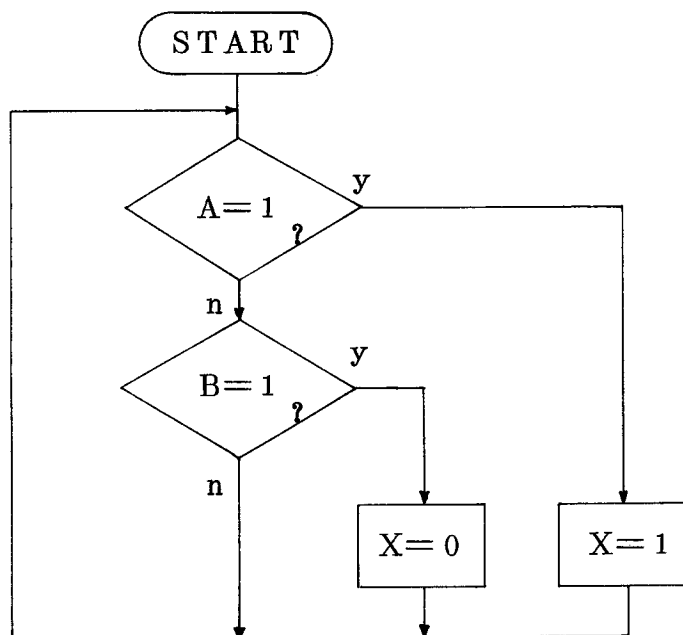


図 3.6 起動優先自己保持動作

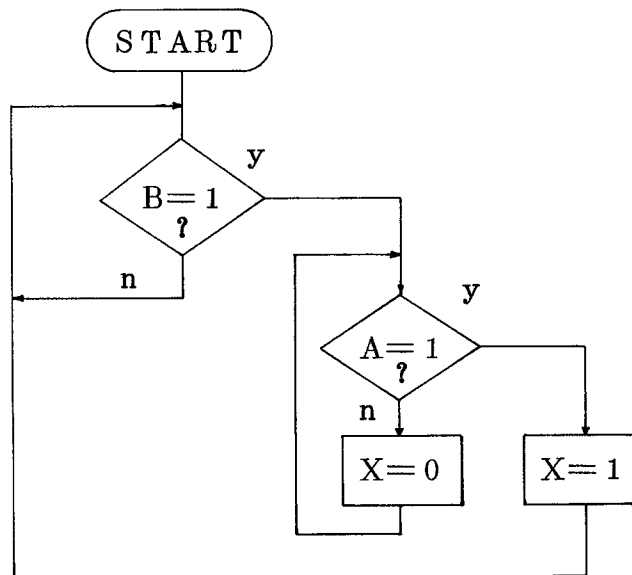


図 3.7 起動優先自己保持動作の問題ある表現



### Ⅲ. 1.2 インタロック付 F F 動作

図 3.8 (a)、(b)に、この動作の機能を有する回路を示す。図 3.9 は、同 3.8 と同一動作機能のフローチャート表現である。これについては、以下の特徴と問題がある。

- (1) 本動作では、インタロックが重要な役割を演じているので、これをフローチャート上に適確に表現しなければならない。つまり、F出力が ON 時には、R出力は ON しないようになっていなければならないし、R出力が ON の場合の F出力も同様でなければならない。この機構の表現は、図 3.9 の押ボタンスイッチ C の状態判断部のループ構造（ラベル L 3 の部分）でなされる。また、同図のフローチャートのラベル L 0 部分にある C の状態判断のループ構造も、同様にインタロック表現である。この機構は、C を操作し（押し）たまま、A、B のいずれかを操作することによって、出力が ON、OFF を繰り返して振動することを防止するものである。これは回路では起り得なかった問題に対するフローチャート特有の防止機構である。
- (2) また、上記のインタロック表現を高級プログラム言語の考え方で処理できる命令も用意されている。待ち命令（WAIT）がそれで、図 3.10 のような活用ができる。

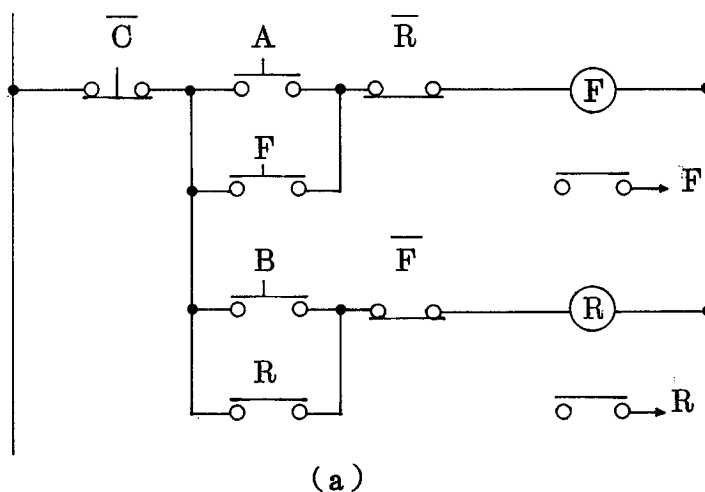
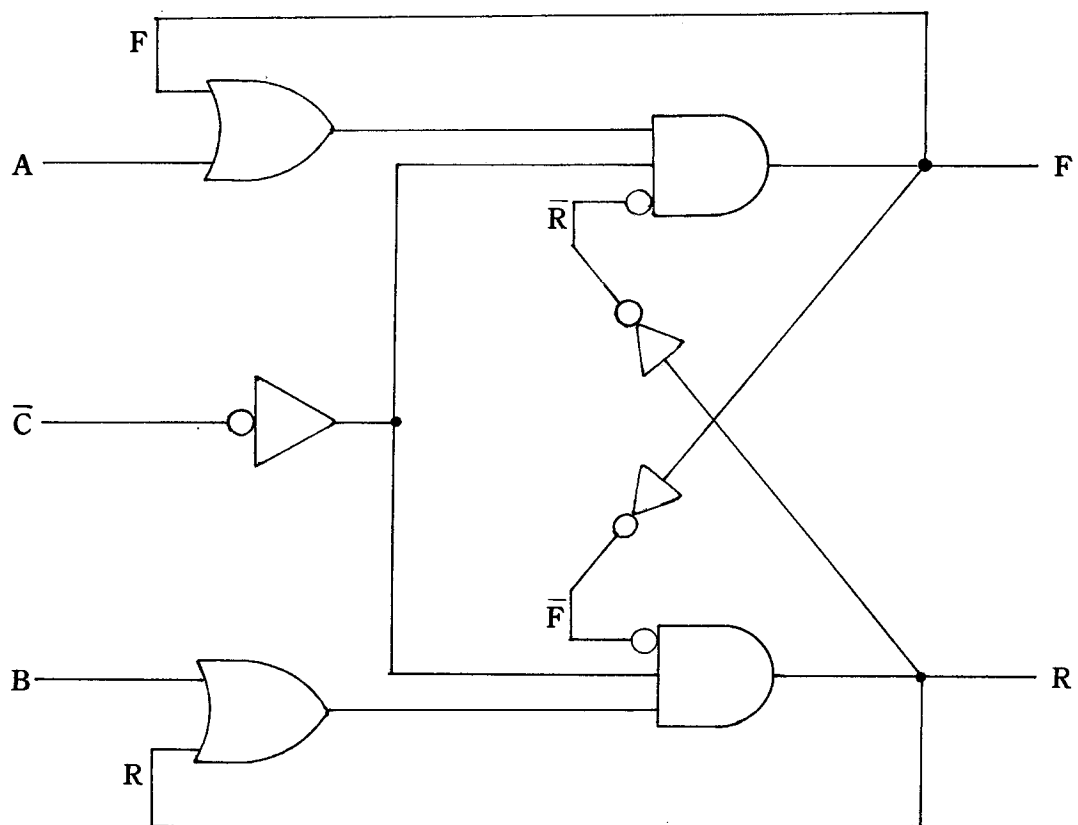


図 3.8 インタロック付 F F 動作回路



(b)

図 3.8 インタロック付 F F 動作回路

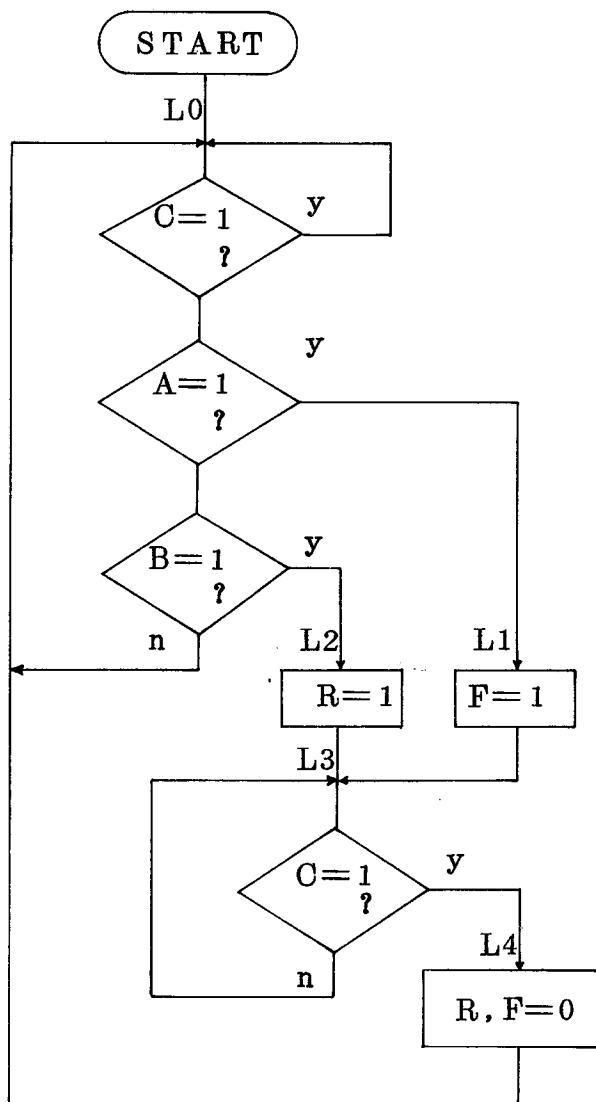


図 3.9 インタロック付 F F 動作

表 3.2 インタロック付 F F 動作  
プログラムリスト

```

0 0 0 0 LBL0: AND      2          ; (PB-C)
          CJP      LBL0
          AND      0          ; (PB-A)
          CJP      LBL1
          AND      1          ; (PB-B)
          CJP      LBL2
          JMP      LBL0
          LBL1: OUT      2 0 0
          JMP      LBL3
          LBL2: OUT      2 0 1
          JMP      LBL3
          LBL3: AND      2          ; (PB-C)
          CJP      LBL4
          JMP      LBL3
          LBL4: OUT·NOT  2 0 0
          OUT·NOT  2 0 1
          JMP      LBL0
    
```

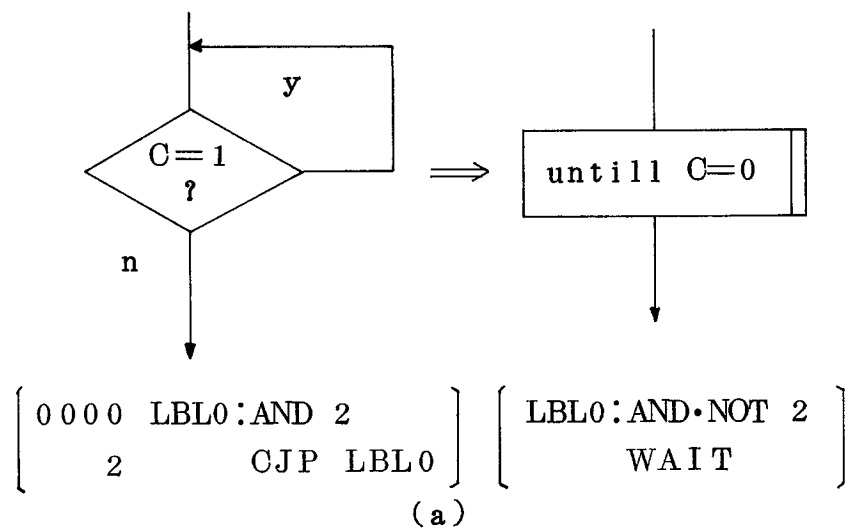
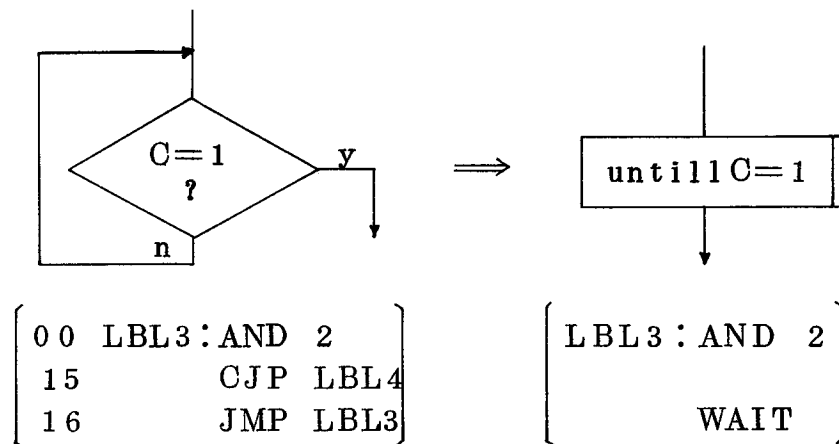


図 3.10 インタロック表現別法



(b)

図 3.1 0 インタロック表現別法

### Ⅲ. 2 プログラム制御

#### Ⅲ 2.1 間 隔 動 作

図 3.1 1 (a), (b) に間隔動作の機能を有する回路を示す。図 3.1 2 は、図 3.1 1 と同一動作機能のフローチャート表現である。この表現については、以下の特徴と問題がある。

- (1) 図 3.1 2 (a) のフローチャート表現では、タイマ時間値の設定は、位置的に押ボタンスイッチ B の状態判断の次になっている。これは、本方式 PC の命令が次のように作られているためである。すなわち、タイマ時間値の設定 ( `T: set` ) とタイマ動作状態 (  $t \stackrel{?}{=} T$ 、タイムアップしたかどうか) の判断とが 1 セットでコーディングしなければならない。本来のフローチャート表現からすると、図 3.1 3 に示すように、位置的に初めの所にタイマ時間値の設定をしておくべきものである。止むをえず、図 3.1 2 に示す表現にせざるをえないため、以下の問題が起きてくる。

このシステムに、押ボタンスイッチ A から起動をかける。そうしておいて、タイマがタイムアップしない内 ( 時間設定が、この場合 3 秒

だから3秒以内ということ)に、Bより停止をかけ、直ちに再度Aから起動をかける。こうした場合、設定値の3秒より少ない時間で停止するという事態が発生する。問題発生の原因は、次に示す本方式P C<sup>(19)</sup>固有のタイマ機能命令の作り方にある。

①タイマーは一たんセットされると、CPUがプログラムをスキャンしていても、タイマ自体は動作する。タイムアップになれば、動作が停止して、タイムアップになったことを記憶している(筆者注、記憶しているのみで出力等の処理は行わない)。

②タイマは減算式である。

③タイムアップしていないタイマを再度使用すると、タイマ設定値の残りの値を継承して減算してゆく。

本例では、一たん停止して再度起動することにより、タイムアップしていないタイマの設定値の残りの値を使用することにより、停止に至る時間が設定値より短くなるのである。

(2) この動作表現を、サブルーチンを活用して表現する方法がある

(図3.14)。起動・停止部を主ルーチンとして、間隔動作部をサブルーチンにしたものであるが、図3.12の表現に比べてすっきりしたわかり易いものになっている。ただ、この表現では、主ルーチンでの、グループジャンプ(GJ)、グループスタート(GS)、グループオフ(GOFF)及びサブルーチンでの、グループナンバ(GN)、グループエンド(GE)というサブルーチン活用の固有の命令を新たに追加して使用しなければならなくなる。また、この方法での正しい表現とするためには、プログラムの実行順序についての理解がなければならない。例えば、サブルーチン部に時間待ち命令(WAIT)があるが、これなどは命令の次のことへの理解を基に使用せねばならない。すなわち、WAIT命令の実行動作は、タイマがタイムアップするまでサブルーチン内でのみループしているのではなく、必ず主ルーチンに戻るように作られているのである。このため、このサブルーチンの動作を停止しようと思えばいつまでも主ルーチン側から停止できることな

どである。

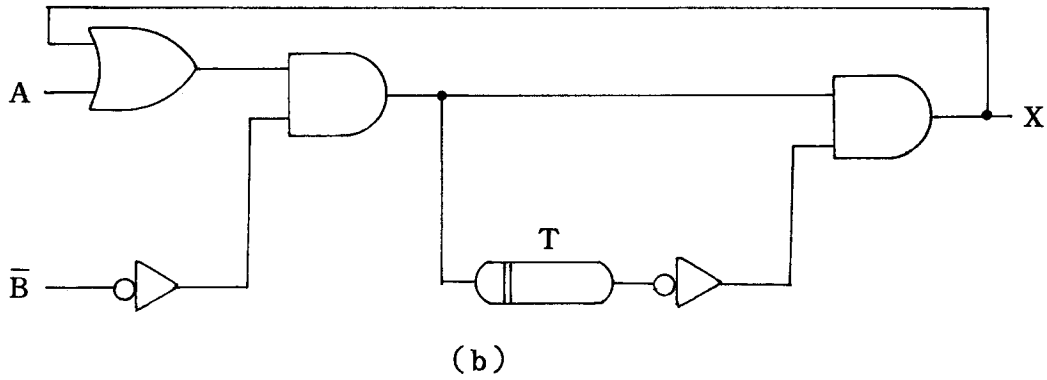
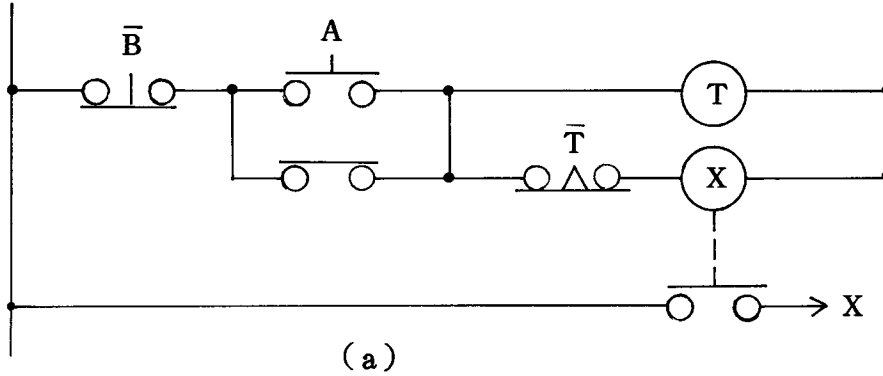
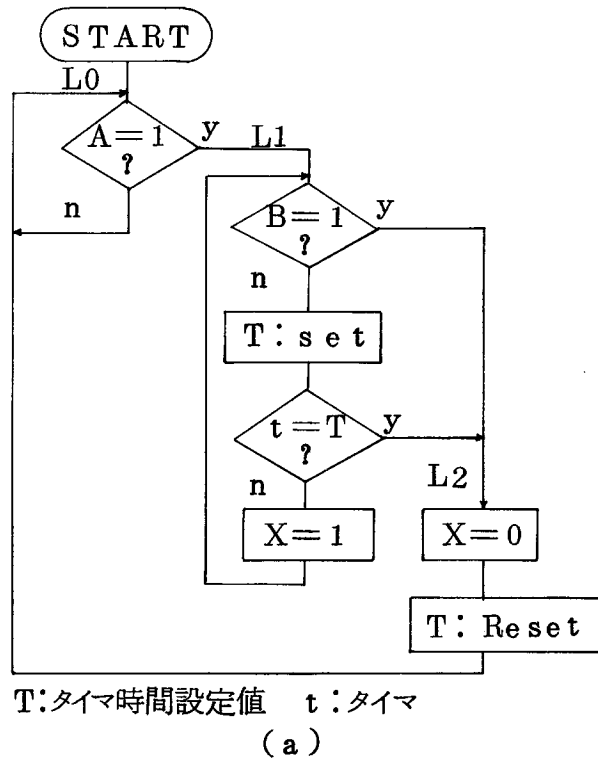


図 3.1 1 間隔動作回路



T: タイマ時間設定値 t: タイマ  
(a)

図 3.1 2 間隔動作

(b) プログラムリスト ( 間隔動作 )

```

0000 LBL0:  AND      0      ; (PB-A)
         CIP      LBL1
         JMP      LBL0
         LBL1:  AND      1      ; (PB-B)
         CJP      LBL2
         TIM      10
         #30
         CJP      LBL2
10      OUT      200     ; (X)
11      JMP      LBL1
12 LBL2:  OUT · NOT  200     ; (X)
14      CNR · TIM  10
15      JMP      LBL0
    
```

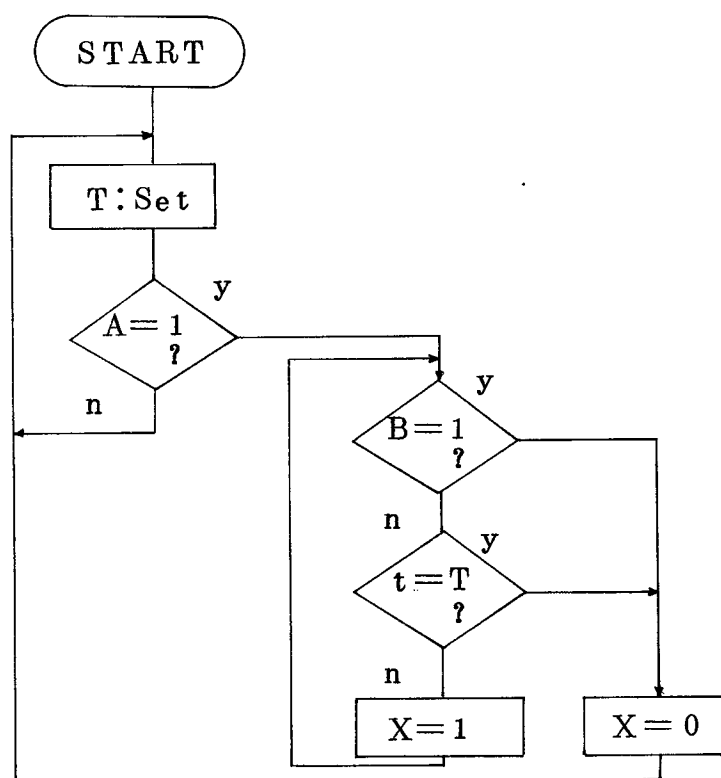
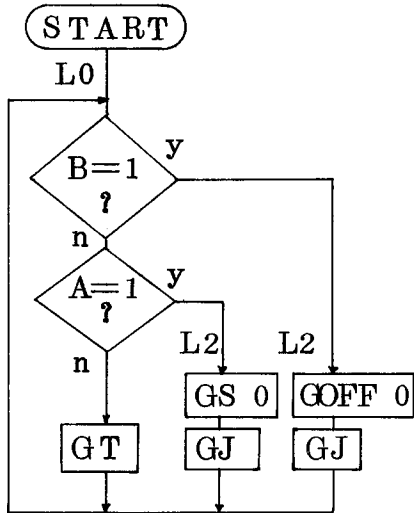


図 3.1 3 間隔動作のタイマ設定方法

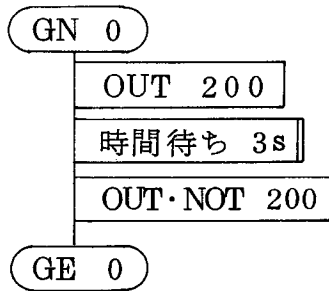


〔起動・停止部〕



0	LBL0:	AND	1
2		CJP	LBL1
3		AND	0
4		CJP	LBL2
5		GJ	
6		JMP	LBL0
7	LBL1:	G OFF	0
9		GJ	
10		JMP	LBL0
11	LBL2:	GS	0
13		GJ	
14		JMP	LBL0

〔間隔動作部〕



20		GN	0
21		OUT	200
22		TIM	10
23			#30
24		WAIT	
25		OUT·NOT	200
26		GE	0

図 3.1 4 間隔動作のサブルーチン活用表現

### Ⅲ 2.2 遅延動作

図 3.15 (a)、(b) に遅延動作の機能を有する回路を示す。図 3.16 は、図 3.15 と同一動作機能のフローチャート表現である。この表現については、以下の特徴と問題がある。

- (1) この動作も停止が優先されて動作しなければならないので、Ⅲ・1.1 (a) 停止優先自己保持動作で述べたように、最も自然なフローチャート表現は、図 3.17 に示すように、停止用押ボタンスイッチ B の状態判断部 (  $\text{◇} \begin{matrix} B=1 \\ ? \end{matrix} \text{◇}$  ) が A の判断部 (  $\text{◇} \begin{matrix} A=1 \\ ? \end{matrix} \text{◇}$  ) より、位置的に先にくるように作ることであろう。しかし、この動作表現には、上記の方法はとれないのである。表現の形としては、A の判断部が B の判断部より先にある図 3.16 に示すフローチャートの形のみになる。
- (2) もし、B の判断部が先にあるフローチャートの形をとるのであれば、図 3.18 に示すサブルーチン活用表現にすればよい。限時遷移動作的なプログラム制御であるから、タイマ命令 (TIM) と時間待ち命令 (WAIT) を活用することにより、先の間隔動作のサブルーチン活用表現と同様、容易に、わかり易い表現となる。

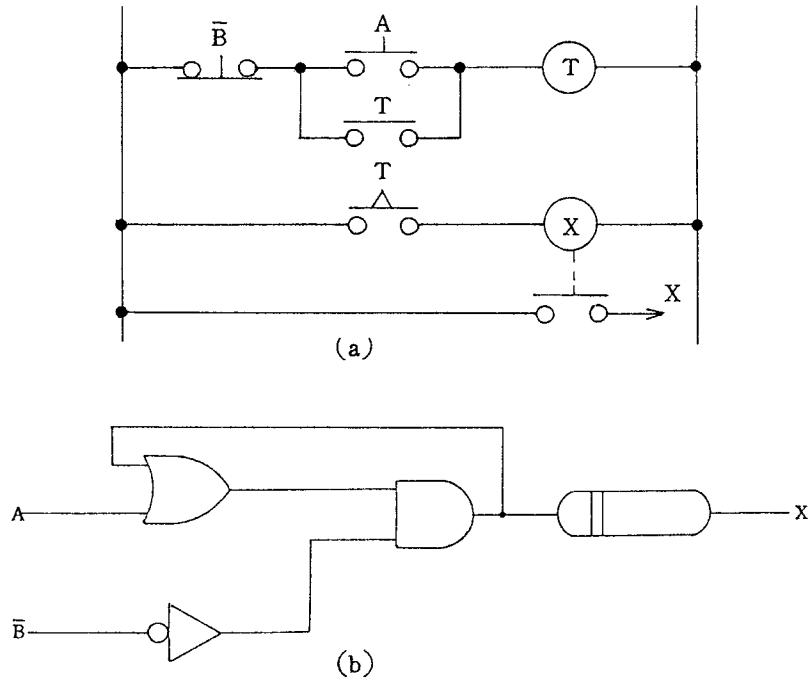
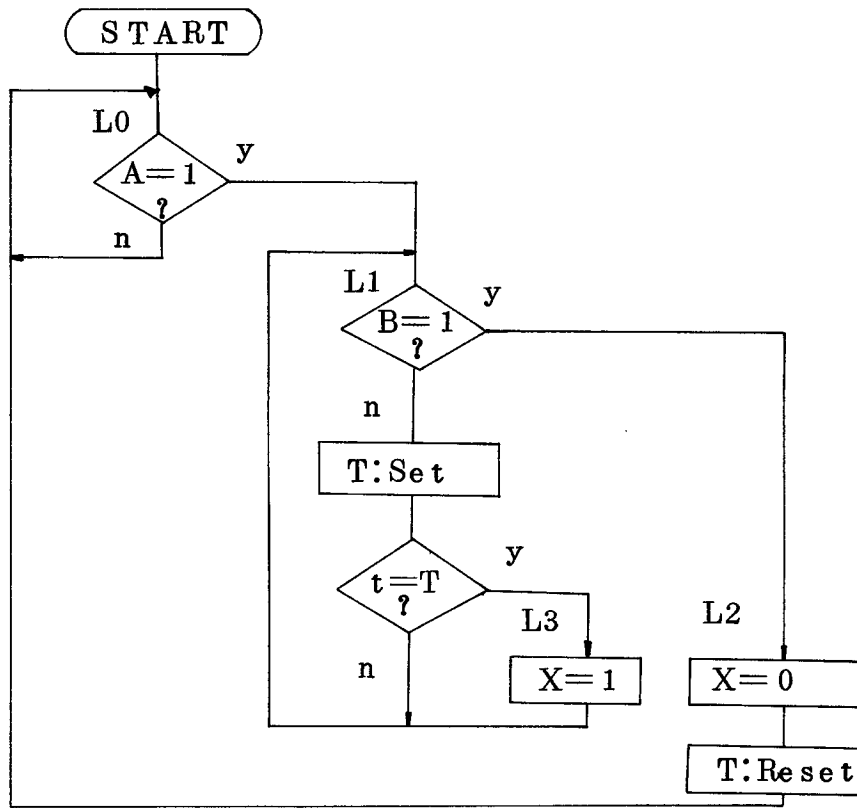


図 3.15 遅延動作回路



( a )

( b ) プログラムリスト

```

0000  LBL0:   AND      0          ; (PB-A)
        CJP      LBL 1
        JMP      LBL 0
        LBL1:  AND      1          ; (PB-B)
        CJP      LBL 2
        TIM      10
        #30      ; ( 3秒 )
        CJP      LBL 3
        LBL2:  OUT·NOT  200       ; (X)
        CNR·TIM  10
        JMP      LBL 0
        LBL3:  OUT      200       ; (X)
        JMP      LBL 1
16

```

図 3.16 遅延動作

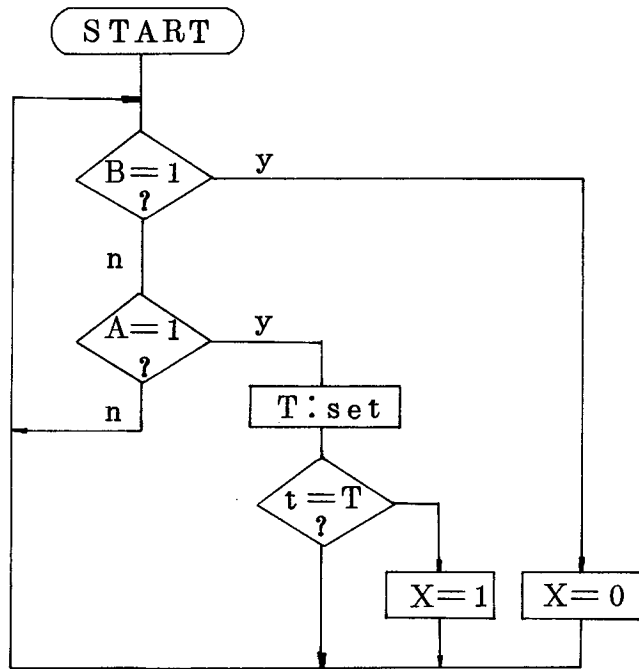


図 3.1 7 遅延動作の問題ある表現

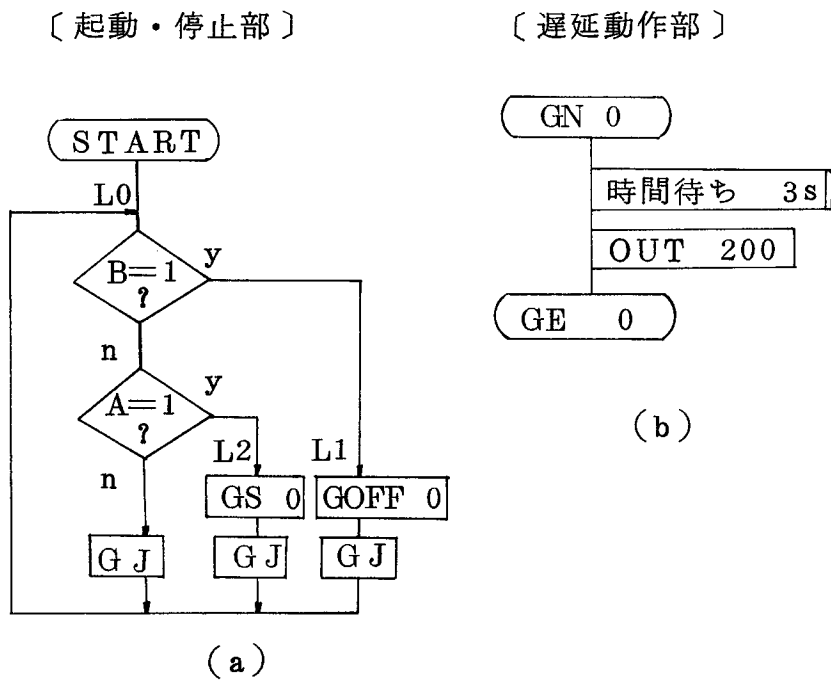


図 3.1 8 遅延動作のサブルーチン活用表現

### Ⅲ . 2 . 3 繰り返し動作

図 3.19 (a)、(b) に繰り返し動作の機能を有する回路を示す。図 3.20 は、図 3.19 と同一動作機能のフローチャート表現である。この表現については、以下の特徴を持つ。

- (1) この動作も、間隔、遅延動作と同様、限時遷移的なプログラム制御であるから、タイマ命令 (T I M) と時間待ち命令 (W A I T) を用いて、サブルーチンを活用した表現が適当である。一般に、このようなプログラム制御的性格の強い動作の表現には、フローチャートは作成上からも、また動作のわかり易さからしても、図 3.19 の回路表現と比較してみればわかるように、格段に優れている。

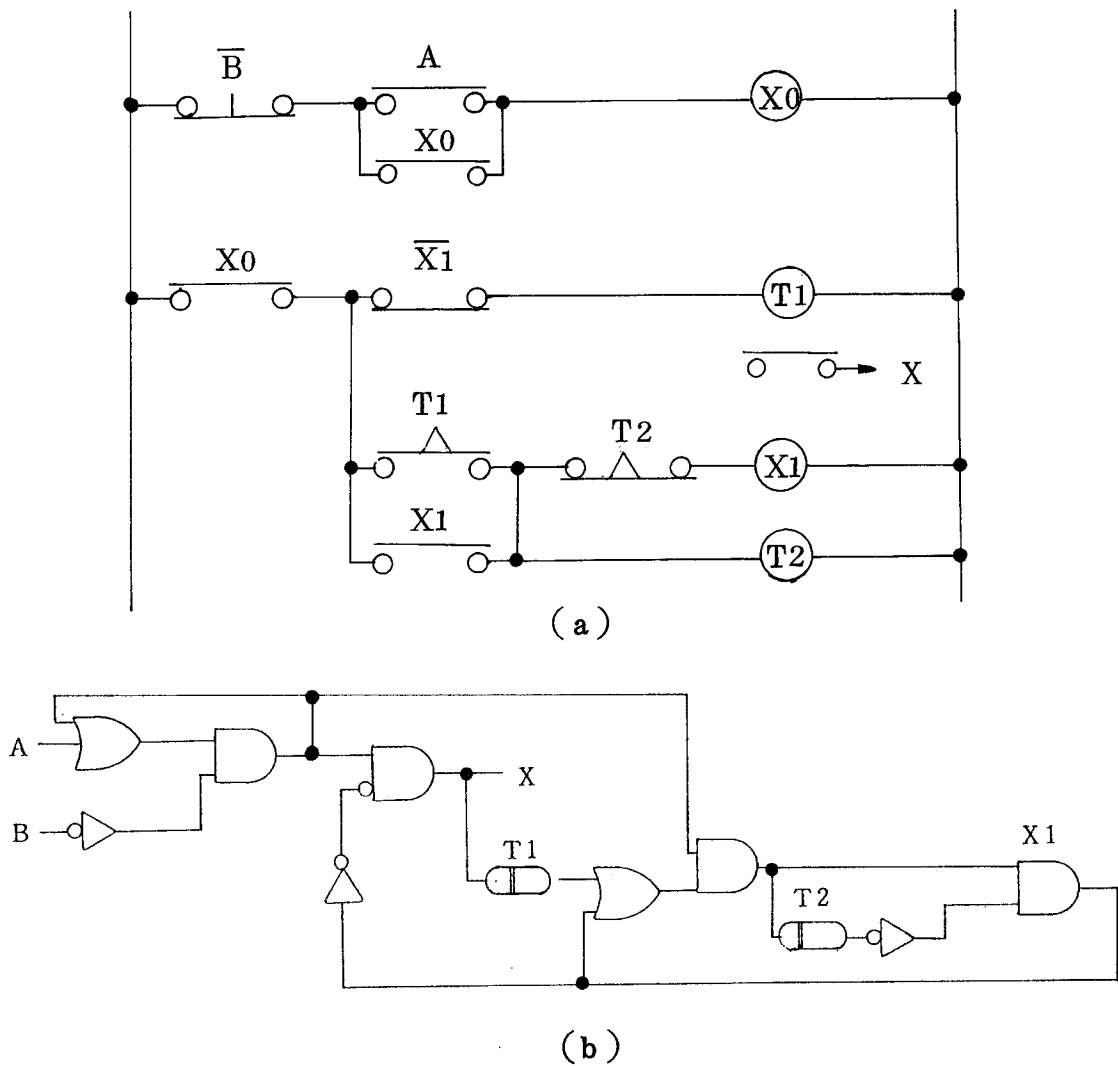
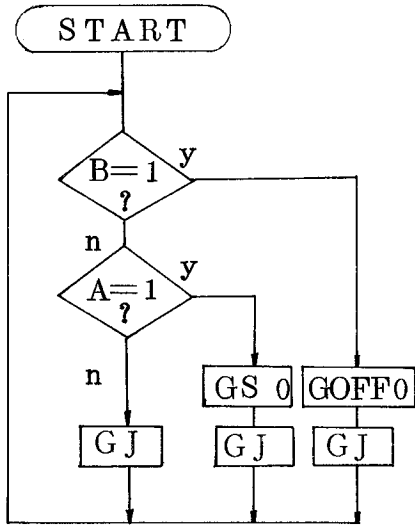


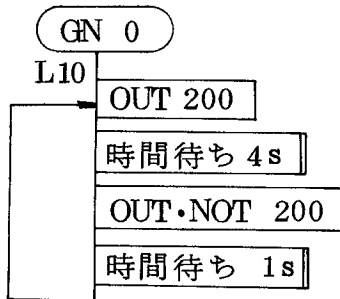
図 3.19 繰り返し動作回路

〔 起動・停止部 〕



0	LBL 0 :	AND	1
2		CJP	LBL 1
3		AND	0
4		CJP	LBL 2
5		GJ	
6		JMP	LBL 0
7	LBL 1 :	G OFF	0
9		G J	
10		JMP	LBL 0
11	LBL 2 :	GS	0
13		GJ	
14		JMP	LBL 0

〔 繰り返し部 〕



20		GN	0
21	LBL10 :	OUT	200
23		TIM	10
24			# 40
25		WAIT	
26		OUT・NOT	200
27		TIM	11
28			# 10
29		WAIT	
30		JMP	LBL 10

図 3.2 0 繰り返し動作

### Ⅲ． 3 並列プログラム制御

本例に限って、動作機能説明を回路で代用せず、図 3.3 1 に示す情報源によって動作の説明をする。この「並列プログラム制御」の実用制御システムの例として、「交通信号機制御システム」をとりあげる。本システムについて簡単に説明を加える。同図 (a) に示すように、“通りⅠ”と“通りⅡ”とにそれぞれ対の信号機Ⅰ、Ⅱがある。信号機の動作は、同図 (c) タイムチャートに示すとおりである。運転状態に入るには、同図 (b) に示す入出力構成の起動用押ボタンスイッチ A を押すことにより行い、停止するには、停止用押ボタンスイッチ B を押すことにより行う。図 3.2 2 が、本動作機能のフローチャート表現である。この表現については以下の特徴をもつ。

- (1) Ⅰ信号機とⅡ信号機との並列動作であるので、サブルーチンを、GN 0 と GN 1 の 2 つとし、これを主ルーチンから起動・停止制御するようにする。各サブルーチンの動作は限時遷移的なため、時間待ち命令 (WAIT) を活用して作る。このため、プログラム (表 3.3) は WAIT 命令の作用で、サブルーチン GN 0、GN 1 と主ルーチンとを時分割でスキャンし、見かけ上並列動作となっている。

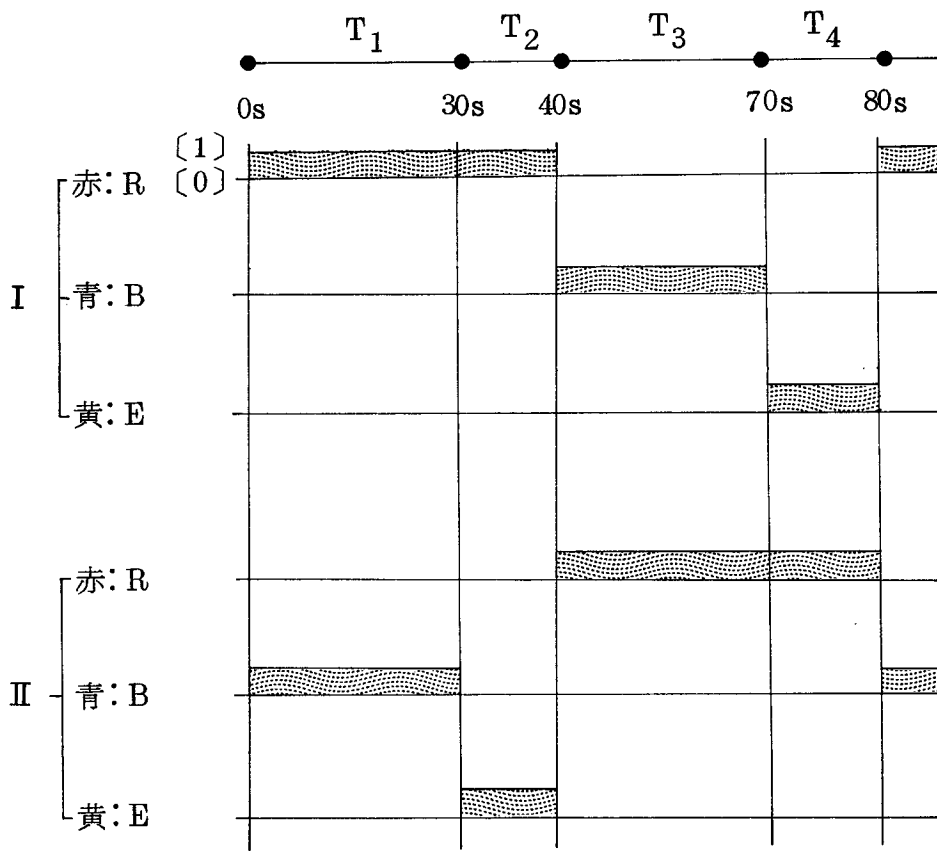
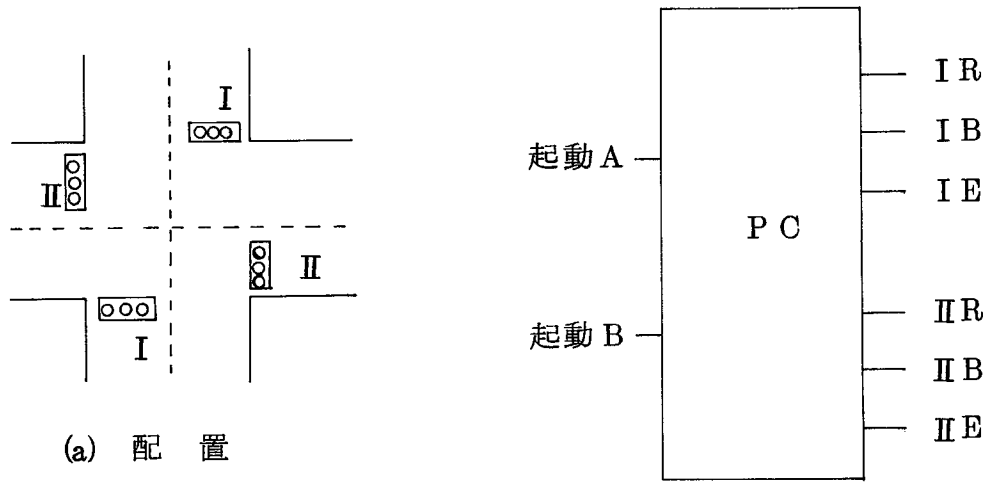
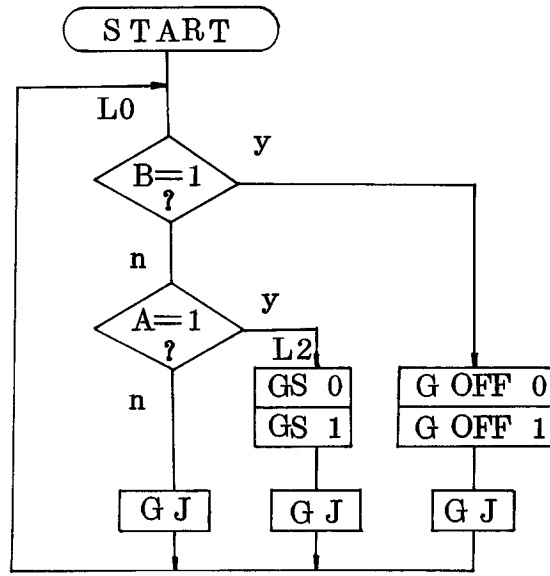


図 3.2 1 交通信号機動作のための情報



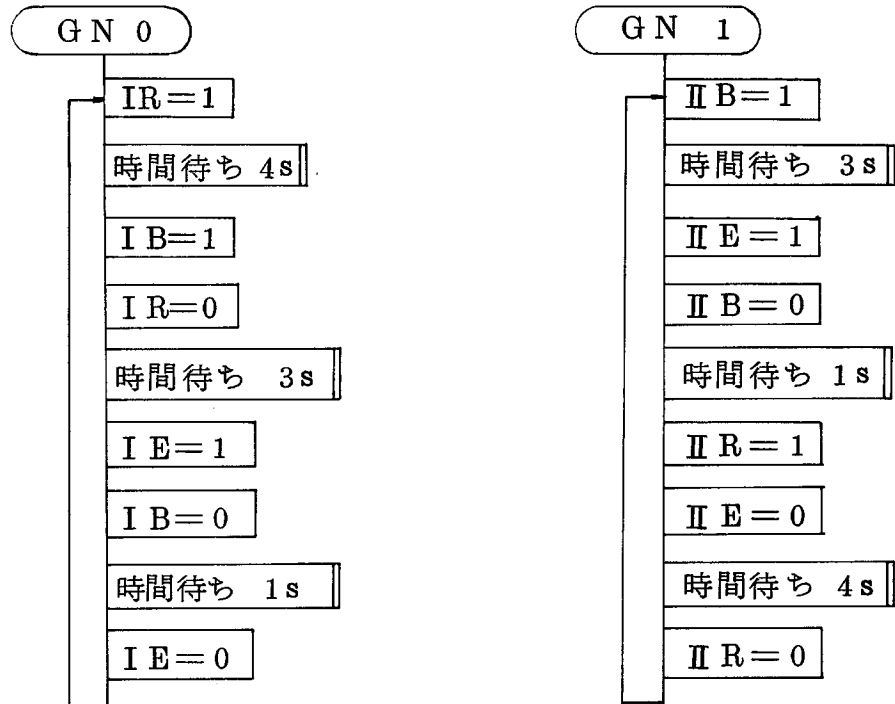
〔 起動・停止部 〕



(a)

〔 通り I 信号繰り返し部 〕

〔 通り II 信号繰り返し部 〕



(b)

図 3.2 2 交通信号機動作

表 3.3 プログラムリスト

(a) 起動・停止部 (主ルーチン)

0	LBL0:	AND	1
2		CJP	LBL1
3		AND	0
4		CJP	LBL2
5		GJ	
6		JMP	LBL0
7	LBL1:	G OFF	0
9		G OFF	1
10		GJ	
11		JMP	LBL0
12	LBL2:	GS	0
14		GS	1
15		GJ	
16		JMP	LBL0

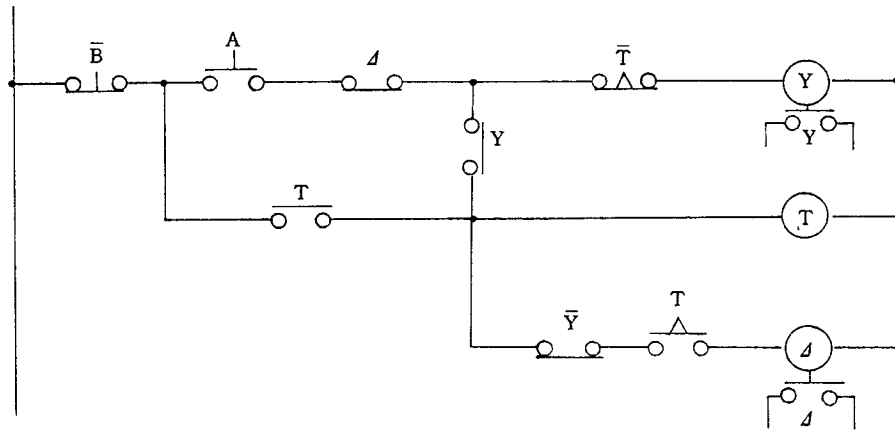
(b) 繰り返し部 (サブルーチン)

20	GN	0	40	GN	1		
21	LBL10:	OUT	200	41	LBL11:	OUT	204
23	TIM	10	43	TIM	20		
24		#40	44		#30		
25	WAIT		45	WAIT			
26	OUT	201	46	OUT	205		
27	OUT·NOT	200	47	OUT·NOT	204		
28	TIM	11	48	TIM	21		
29		#30	49		#10		
30	WAIT		50	WAIT			
31	OUT	202	51	OUT	203		
32	OUT·NOT	201	52	OUT·NOT	205		
33	TIM	12	53	TIM	22		
34		#10	54		#40		
35	WAIT		55	WAIT			
36	OUT·NOT	202	56	OUT·NOT	203		
37	JMP	LBL10	57	JMP	LBL11		

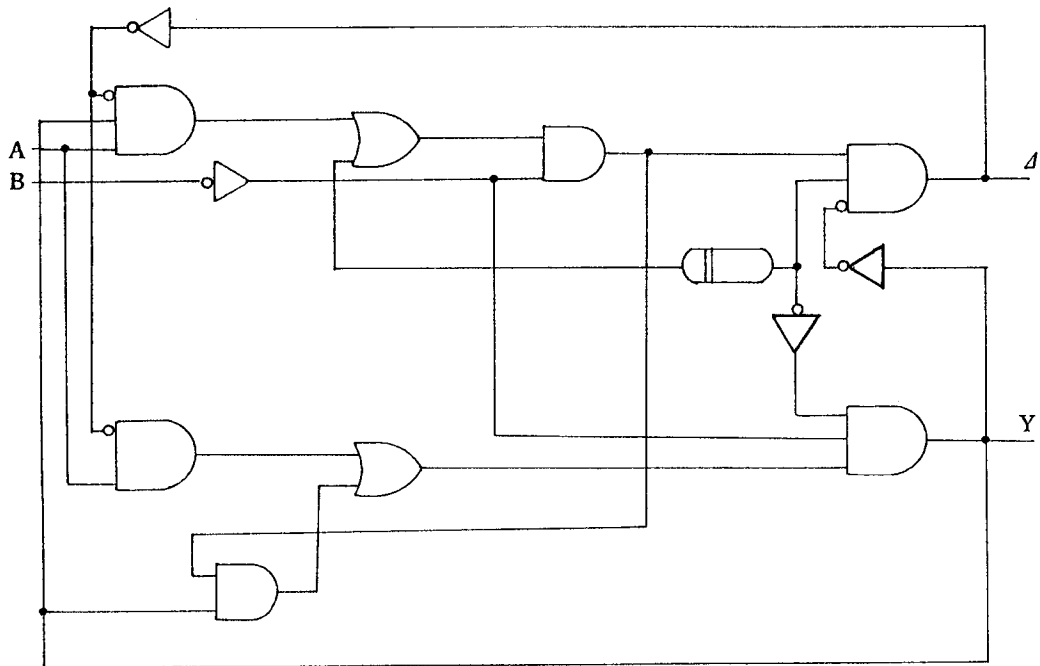
### Ⅲ．４ 条件+プログラム制御

この制御では、インタロック付限時遷移動作を取り上げる。実用化の例では、巻線形誘導電動機の Y- $\Delta$  起動制御がある。図 3.23 にこの動作機能を有する回路を、また、図 3.24 に同機能のフローチャート表現を示す。この表現については以下の特徴を持つ。

- (1) ソフトウェア工学におけるプログラム作成に関して、構造化定理があるが、これは、「プログラムが 1 つの入口と 1 つの出口を持つように設計されていれば、どのようなプログラムでも、3 種類の基本形の組合せにより作ることができる」というものである。3 種類の基本形とは、① 連接、② 反復、③ 選択 (図 3.25 参照) である。しかし、ここに示したフローチャートに関しては、この一部を図 3.25 の 3 種類の基本形を用いて書いた図 3.26 からわかるように、構造化定理の適用が困難となる。もちろん、このプログラム (フローチャート) 自体が、構造化定理の条件「1 つの入口と 1 つの出口を持つように設計する」を満足してはいない。しかし、ここに見るようにシーケンス制御用プログラムは、データ処理や技術計算のような普通のプログラムに比較して特殊な制御構造を持つといえる。
- (2) 図 3.27 に本動作のサブルーチン活用表現を示すが、システムの起動・停止部を主ルーチンに、インタロック付限時遷移部をサブルーチンにしている。サブルーチン中、条件制御的なインタロック機構を、待ち命令 (WAIT) を活用して表現 (図 3.28 (b) の `until  $\Delta=0$`  が相当) している。同一動作表現である先の図と比較すればわかるように、後者が明らかに制御構造が簡単でわかり易い。この理由は、待ち命令 (WAIT) に有効な機能を持たせたためである。



(a)



(b)

図 3.23 インタロック付限時遷移動作回路

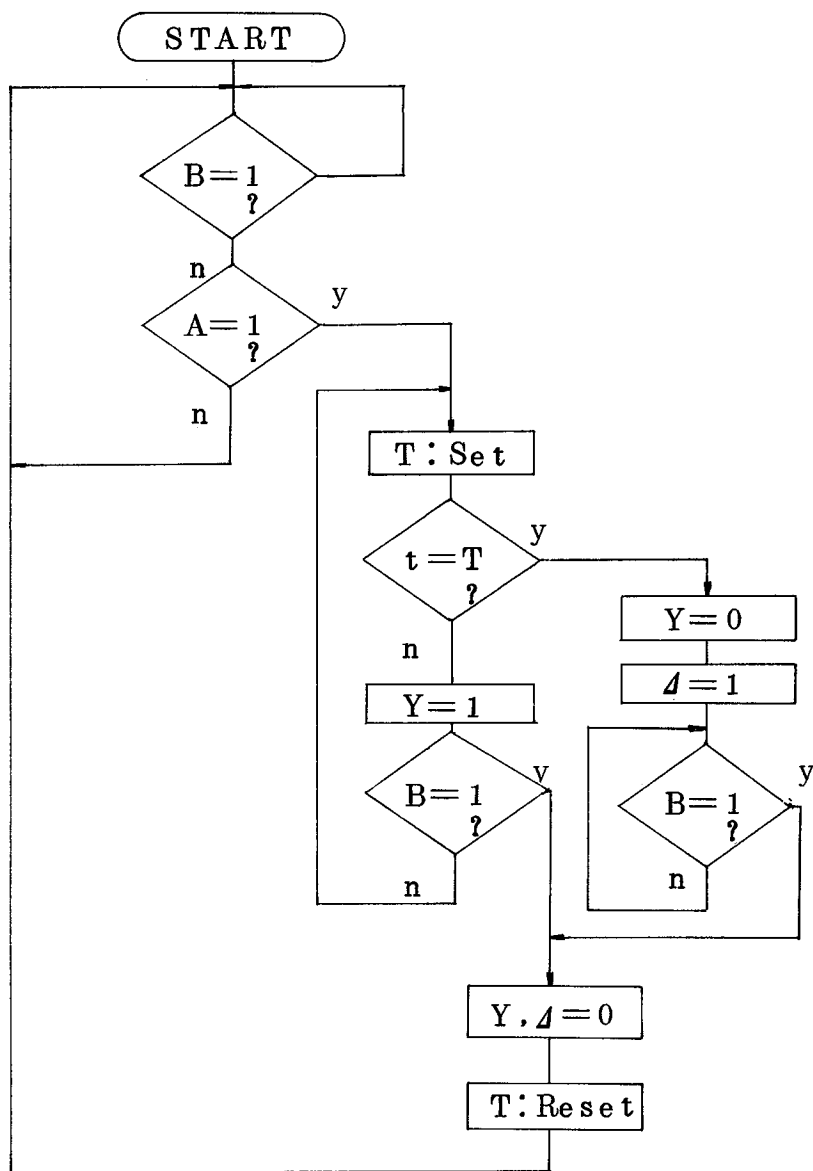


図 3.24 インタロック付限時遷移動作

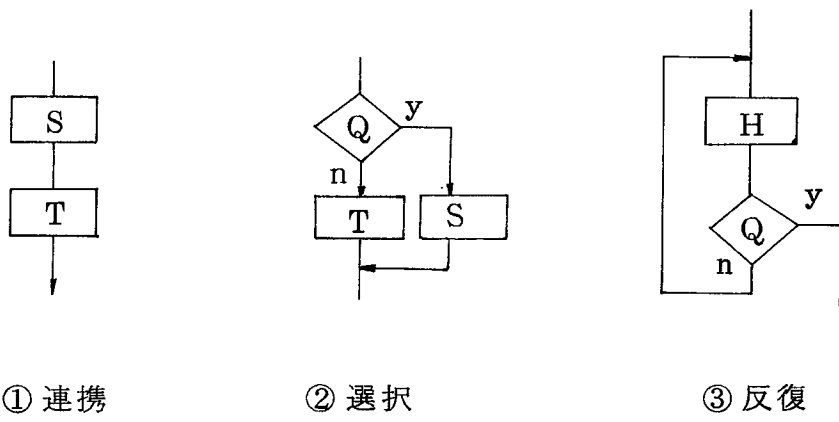


図 3.2 5 プログラムの基本形

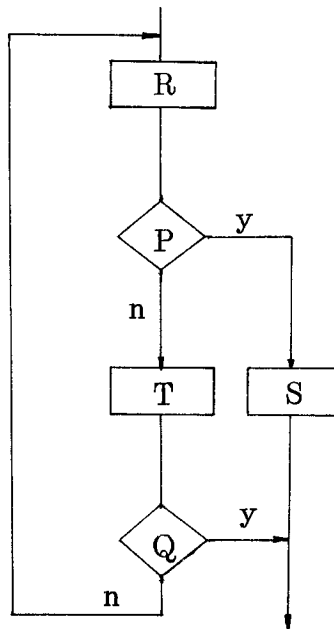
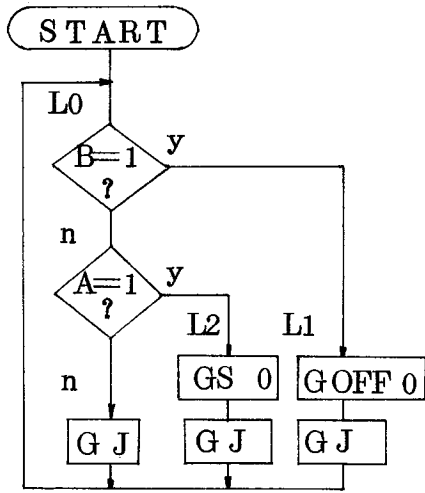
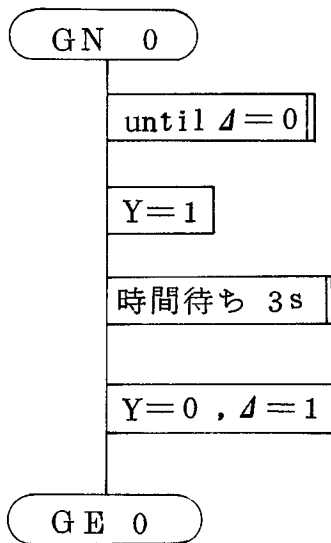


図 3.2 6 インタロック付限時遷移動作の制御構造



(a) 起動・停止部

1	LBL0:	AND	1
2		CJP	LBL1
3		AND	0
4		CJP	LBL2
5		GJ	
6		JMP	LBL0
7	LBL1:	G OFF	0
9		GJ	
10		JMP	LBL0
11	LBL2:	GS	0
13		GJ	
14		JMP	LBL0



20		GN	0
1		AND・NOT	201
2		WAIT	
3		OUT	200
4		TIM	10
5			#30
6		WAIT	
7		OUT・NOT	200
8		OUT	201
9		GE	0

(b) インタロック付限時遷移部

図 3.27 インタロック付限時遷移動作のサブルーチン活用表現