

Speed-Independent 制御方式と Semi-modular 回路の設計手順

長崎大 医 中村 剛
早大 理工 宇都宮公訓

§1. 序

電子計算機の論理設計において、セミモジュー回路の理論が適用される部分及び方法を明らかにする為に、ユークザ例をみることにします。本格的な非同期制御方式計算機として有名なイリフックIIの論理のうち主に数値計算部分制御についての概要を報告している[1]に従って

述べてみます。図1はイリフックIIの数値計算部分のあるブロックです。A, S, Q, R, Mは数字用のレジスターで(フリップフロップ及び論理素子からなっている)各レジスターに付随しているG1, G2, G3, G4, G5はレジスターに情報が入るのを制御する為のゲートと呼ばれるもので、現在貯まられているレジスターの内容が使用中であれば、このゲートは閉じられており又開いているときには新しい情報がこのレジスターに入れられます。レジスターに入る情報の源、

を選択するのが $S_1, S_2, S_3, S_4, S_5, S_6$ と記されているセレクトターと呼ばれるものです。例えば四則演算等がよく使われる右シフト及び左シフトを実行するさいには、まづシフトが正しい方向に行われるようにセレクトターをセットし次にゲートを開いて情報を転送することになります。ゲートとセレクトターに信号を送り数値計算部分を制御するのが非同期制御部門([1]では *delayed control*)であり、これはいくつかの制御ステップからなるフローチャートと呼ばれるものの集合と考えられる。各制御ステップではいくつかの制御命令が平行して実行されるが、二つ以上の異なるフローチャートが同時に実行されることはない。フローチャートは次の三つのことを明示する。

- (1) どのゲート及びセレクトターがセットされるか。
 - (2) 次にくる制御ステップは何か
 - (3) 次にくる制御ステップの選択に使われる条件は何か。
- 図2に[2]で与えられているフローチャートのうちの2制御ステップが書かれている。上側の制御ステップCP1での記号 gS と gR はレジスター S と R のゲートに信号を発すること、 RS と SR はその R, S のセレクトターを右シフトにセットすること、又 $\overline{DL}$ は記憶素子 DL を0にセットすることの意味している。CP2での記号 gA, gQ はレジスター

A と Q のゲートに信号を送ることを、 $SDS AQ$ はその A , Q に直接つながっているレジスターから A , Q に数値が転送されるべくセレクターをセットすること、又 KgA はキャリーをフロッパーに加える為のゲートを開くことを意味している。これらフローチャートで示される一連の制御命令を実行する論理機能が非同期回路で構成されており、特にごく一部を除いて Speed-Independent 設計がなされている。即ちその回路の論理機能が素子の速度に無関係に正しく実行されるよう設計されている。制御用フリップフロップの操作等のごく一部にはそれらにつづく組み合わせ回路が安定状態になるのに要する時間だけ次の制御命令を遅らせる方式がとられている。それでは [2] に従ってフローチャートを Speed-Independent 回路で置き換えていく実例をみることにする。図3は [2] のなかの一頁を丸写ししたものです。右上の回路ブロックが左上のフローチャートを実現していることが下段の回路の状態遷移表をみることにより示されている。このフローチャートはゲート A , B , C が順々に操作されることを意味している。回路図の中で用いられている $\textcircled{A}$ は AND-NOT を $\begin{array}{|c|} \hline 1 \\ \hline 0 \end{array}$ は図4で示されるフリップフロップを示しており $\textcircled{\Delta}$ はゲートのロックを表現している。実際のゲートには多くの組み合わせ回路や増幅器を含む

まれているが、制御部門から見れば一つの屋延素子としてと
 してこしつかえない。図3下の状態遷移表をみれば、オ
 1段目でゲートAを開く信号0が発せられているがまだゲー
 トからの受信信号 gAr は0になっていない。オ二段目で
 受信信号が返って来てオ7段目でゲートAを閉じる信号1が
 発せられオ8段目で受信信号が返って来てオ9段目でゲート
 Bを開く信号が発せられている。この他にも勿論多くの複
 雑なフローチャートが合成されているわけであるが、それら
 の紹介は本論からはみれるので省くことにする。[2]又は
 [4]にも詳しい説明がなされています。イリフックIIの数
 値計算部門は6000条のトランジスターを含む非同期回路で
 構成されその論理設計は全て発見的になされ厳重なテストで
 確められて完成した([3]による)。計算機の完成にとも
 なってイリフックIIとしての論理設計問題は完結したわけだ
 らその後[5, 6, 7, 8, 9]等を経て Speed-Independent
 回路の理論的発見がなされ数学的基礎が確立された。そこ
 で扱われているチェーンチャート及びスタートチャートの概
 念とフローチャートとの関係を大づかみに述べてみます。
 図5aはCP1でゲート1とゲート2の開閉を平行して行い、
 完了を待ってゲート3の開閉を行うことを示したフローチャ
 ートであり、それをチェーンチャートに翻訳すると図5bの

ようになりす。 $[1, 1]$ はゲート 1 につながっている信号線への状態を 1 回変化させる (従ってもし最初 0 であれば 1 になる) ことを意味し、 $[1, 2]$ は同様に 2 回目の状態の変化を意味して一般に $[i, j]$ はゲート i につながっている信号線の j 回目の状態変化を意味している。 $[i, j] \rightarrow [k, l]$ は $[i, j]$ が完了してから $[k, l]$ を実行することを示しています。一方 $[1, 1]$ と $[2, 1]$ には $\rightarrow$ の関係がないので $[1, 1]$ と $[2, 1]$ は平行して実行されることになりす。図 5 b のチェーンチャートで指定された様に回路の素子が信号を変化したときの、回路全体の状態の変化をみる為に、ゲート 1, 2, 3 につながっている信号線の信号変化の数をそれぞれ第 1, 2, 3 成分に書いた 3 次元ベクトルを考えると図 5 c の様に書かれる。例えば $(110) \begin{matrix} \nearrow (210) \\ \rightarrow (220) \\ \searrow (120) \end{matrix}$ は 1 番目と 2 番目の信号線が 1 回ずつ状態変化した後で 1 番目と 2 番目の信号線の状態のどちらでもあるいは同時にでも 2 回目の状態変化をすることを意味している。この様にフローチャートからチェーンチャートとしてスタートチャートと変換することが出来るが一般には逆はできません。スタートチャートからチェーンチャートへの変換ができる必要条件是スタートチャートが分配的のときであり ([5])、又チェーンチャート $[1, 1] \rightarrow [2, 1] \rightarrow$

$[1, 2] \rightarrow [2, 2] \rightarrow \dots$ に対応するフローチャートは存在しない。更に重要なことは一般にフローチャートを変換してできたまでのチェンクチャートあるいはスタートチャートは不完全である。実際図5cのスタートチャートに従って回路の状態遷移図を書いてみると図5dの様になり、状態(000)の次にくるべき状態が決定不可能であり、このままでは論理設計の困難さはフローチャートの時と少しも違っていません。不完全なスタートチャートを拡張(extension)して完全な(digital)スタートチャートを得ることを合成(synthesis)するといいい[10, 11, 12, 13]でその手順が与えられた。完全なスタートチャートに従って状態遷移図を書くと、 n 次元 $\{0, 1\}$ -ベクトルを頂点としてもつ特殊なグラフ(デジタルグラフ)になり、特にスタートチャートがセミモジューラーの時にはグラフもセミモジューラーになる([6])。セミモジューラーなデジタルグラフが与えられた時に、それに従って Speed-Independent 回路を設計する為の手順がこれから述べる論理設計アルゴリズムです。非常に長い前あきになりしたが、これから述べるアルゴリズムには前あきの内容は用いておりません。

§2. 前半は[14]で示されたアルゴリズムの若干変形されたものを具体例に照らし合わせて述べ、後半は未だ証明は

完了してないので不変法はありますが、新しいアルゴリズムについて述べます。アルゴリズムの正当性の証明等は省略します。ディジタルグラフは以下で定義される n 次元 $\{0,1\}$ -ベクトルを頂点としたグラフの特殊なものです。

I を空でない有限集合とする。とくに断らない限り $I = \{1, 2, \dots, n\}$ とする。 I の各要素はノードといわれる。また、 $\{0,1\}^I$ で n 次元 $\{0,1\}$ -ベクトル $x = (x_1, \dots, x_n)$ の全体を表わす。ただし、すべての $i \in I$ について $x_i \in \{0,1\}$ である。

[定義 1] $\{0,1\}^I$ の 2 点 x, y に対して $\{0,1\}^I$ の部分集合 $[x, y]$ を $[x, y] = \{z \in \{0,1\}^I ; \text{すべての } i \in I \text{ に対して、} \min\{x_i, y_i\} \leq z_i \leq \max\{x_i, y_i\}\}$ と定義する。

$\{0,1\}^I$ の部分集合 G と、 G から $\{0,1\}$ への $|I|$ 個の写像 $\{f_i\}$ を導入し、 G の各点 x に関して $x^* = (f_1(x), \dots, f_n(x))$ としたとき $[x, x^*] \subset G$ を考える。このとき、3-組 $I, G, \{f_i\}$ に対して、ディジタルグラフ (G) を次のように定義する。

[定義 2] $x, y \in G$ に対して $(x, y) \in (G)$ であるのは

- (1) $y \in [x, x^*]$ でありかつ
- (2) $x \neq x^*$ のときは $y \neq x$ である。

例 1. $I = \{1, 2\}$, $G = \{0,1\}^I$, $f_1(x) = f_2(x) \equiv 1$ とす

ると、ディジタルグラフ $\textcircled{G}$ は図 6 a のようになり、 $I = \{1, 2, 3\}$, $G = \{0, 1\}^I$, $f_1(x) \equiv 1$, $f_2(x) = \overline{x_1} \overline{x_3} + x_1 \overline{x_2}$, $f_3(x) = \overline{x_1} x_2 + x_1 x_3$ とするとディジタルグラフ $\textcircled{G}$ は図 6 b のようになる。

[定義 3] ディジタルグラフ $\textcircled{G}$ がセミモジューであるとは $(b, c) \in \textcircled{G}$ でかつある $i \in I$ に対して $c_i \neq b_i^*$ であれば $b_i^* = c_i^*$ であるときをいう。これは次の条件と同値である。すなわち、 $(b, c) \in \textcircled{G}$ であれば $b^* \in [c, c^*]$ である。図 6 a のグラフはセミモジューでないが図 6 b はセミモジューである。

それでは [14] のアルゴリズムに従って図 6 b のグラフを合成してみる。まず各 $i \in I$ について $\{0, 1\}^I$ の部分集合 $M(i, 0, 1) = \{(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m); b_i = 0 \text{ かつ } b_i^* = 1\}$
 $M(i, 1, 0) = \{(b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_m); b_i = 1 \text{ かつ } b_i^* = 0\}$ をつくる。即

$$M(1, 0, 1) = \{(10), (11), (00), (01)\}$$

$$M(1, 1, 0) = \emptyset$$

$$M(2, 0, 1) = \{(00), (10), (11)\}$$

$$M(2, 1, 0) = \{(01), (10), (11)\}$$

$$M(3, 0, 1) = \{(01), (10), (11)\}$$

$$M(3, 1, 0) = \{(00), (10), (11)\} \quad \text{である。}$$

つぎに各部分集合について同期式同路で良く行われるように

cubical representation をする

$$M(1, 0, 1) = [(00), (11)]$$

$$M(1, 1, 0) = \phi$$

$$M(2, 0, 1) = [(00), (10)] \cup [(10), (11)]$$

$$M(2, 1, 0) = [(11), (10)] \cup [(01), (11)]$$

$$M(3, 0, 1) = [(01), (11)] \cup [(10), (11)]$$

$$M(3, 1, 0) = [(00), (10)] \cup [(10), (11)]$$

するとただちに各1毎に単位回路が作られて図7の様になる。
^{回は0発信器}

。 図中 AND は AND , OR は OR を示し $\begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \end{array}$ はフ

リップフロップでその論理特性は上側と下側の入力とが必要な
 るときに下側の入力信号が1の側の出力に、上側の入力信号
 が下側の出力になり、一致しているときは状態変化はないも
 のである。 又1側の出力信号の変化と0側のそれとの時間
 差は無視できるものとする。 これらの単位回路を自然に連
 結して図8の回路をうる。 これに各信号線の上にふられた
 初期状態を与え、各フリップフロップの1側の出力の変化
 をみていれば図6bで与えられたグラフに従っておりしかも
 適当に素子の遅れの大小関係を仮定すれば図6bでの任意の
 状態変化が可能である。 こうして紙の上では任意のセミマ
 ジュラスラートチャートの論理設計が可能になったが、使
 用されているフリップフロップにおける仮定及び実際にこの

回路でゲートを制御することと考えると種々の問題が生じるであろう。フリックフロップを使わずに 1-output line の論理素子だけを使って単位回路を作ることが望ましいわけであるが、上記の回路を改良するべく考察された以下の回路が Speed-Independent に動くかどうか検討中である。

単位回路の一般形は図9のようになる。図中各信号線にそれぞれの果す役割にふさわしい名前がふられている。図中①は多数決の論理機能をもった素子であり、 は AND-NOT、 は OR-NOT を示している。この回路では図10の振にして容易に任意個のゲートをとう入できる。しかし使用される素子の種類と数が増え又それとともに Speed-Independent であることを証明するのが困難である。

3.3 結論

R. E. Miller が [4] においてチェーンチャート及びスタートチャートからの Speed-Independent Design へのアプローチについて指摘した不利な面の一つに条件付フローチャートが表現困難なことと完全なスタートチャートからの論理設計が困難なことがある。前者はこれから解決していかなければならない問題であるが、後者に対する一つの答がここで述べた論理設計手法である。[15] で説明されてい

回路設計における諸問題や、fan-in fan-out 等のこと
 がこには触れられなかったが、これらも考慮に入れた設計
 手法が開発される必要がある。

図 1

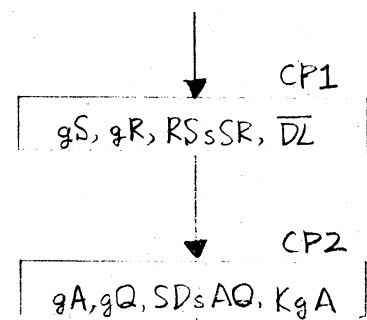
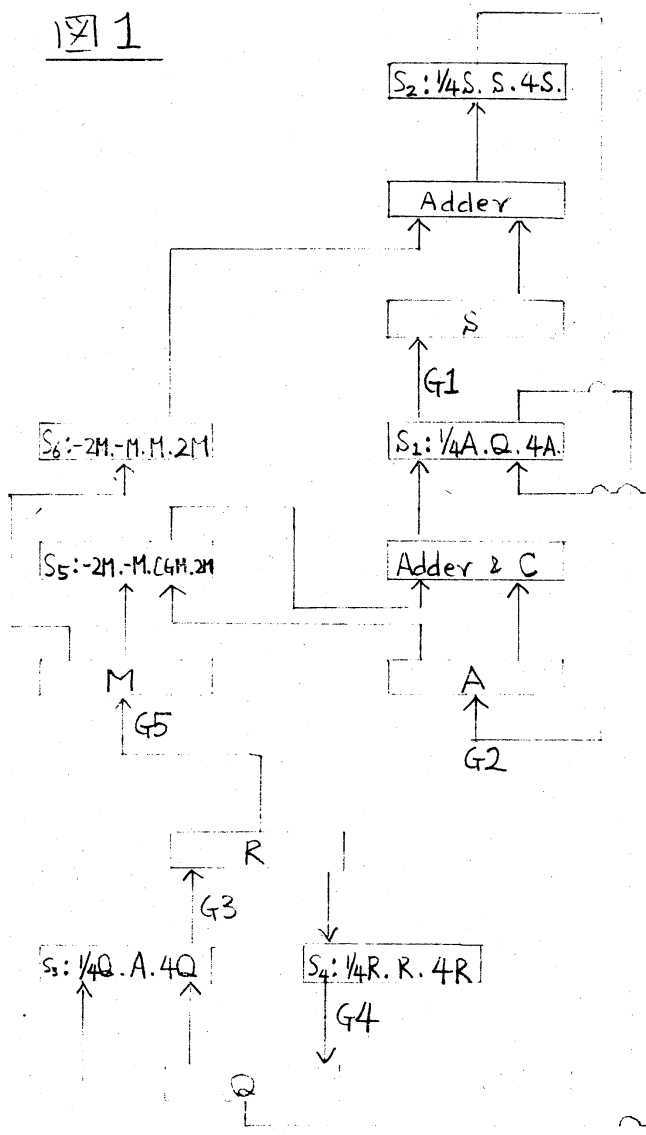
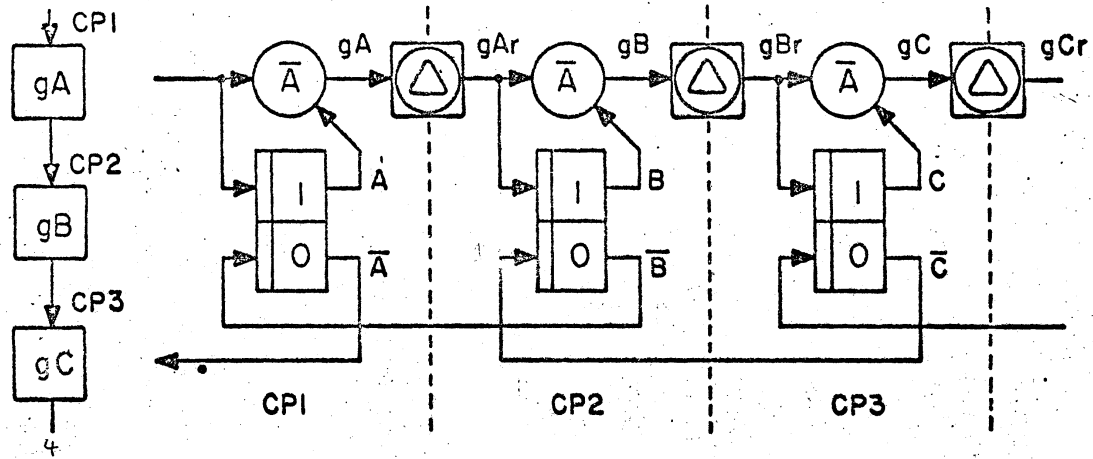


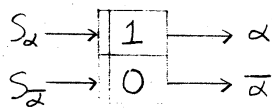
図 2

FLOW CHART

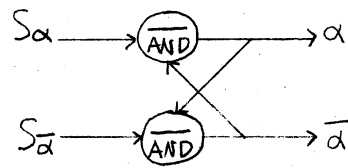


A	$\bar{A}$	gA	gAr	B	$\bar{B}$	gB	gBr	C	$\bar{C}$
1	0	0	1	0	1	1	1	0	1
1	0	0	0	0	1	1	1		
1	0	0	0	1	1	1	1		
1	0	0	0	1	0	1	1		
1	1	0	0	1	0	1	1		
0	1	0	0	1	0	1	1		
0	1	1	0	1	0	1	1		
0	1	1	1	1	0	1	1		
0	1	1	1	1	0	0	1	0	1
				1	0	0	0	0	1
				1	0	0	0	1	1

Fig. 3 - AN EXAMPLE OF A SEQUENCING CONTROL



フリップフロップ



ロジック

S_α	$S_{\bar{\alpha}}$	α	$\bar{\alpha}$
1	1	Memory	
0	1	1	0
1	0	0	1
0	0	1	1

真理値表

図 4

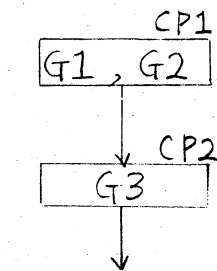
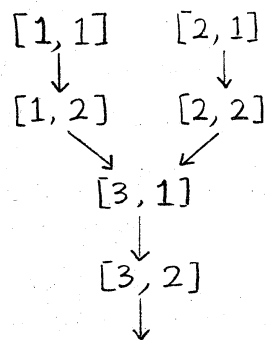
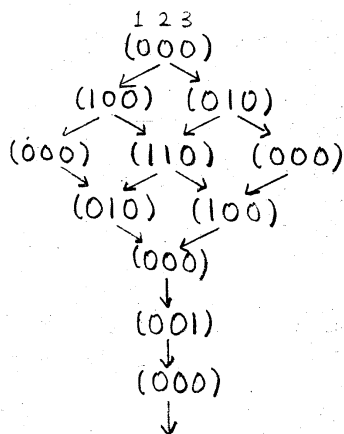
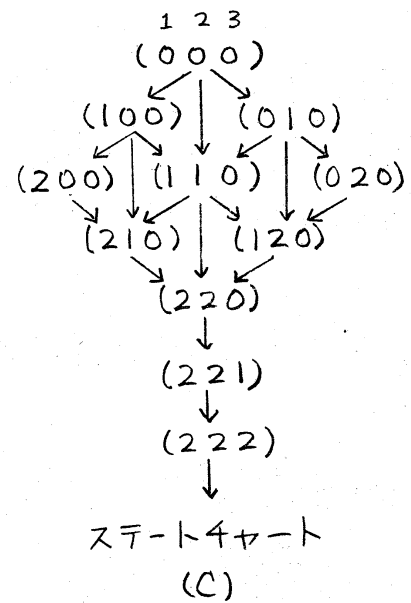
フローチャート
(a)チェーンチャート
(b)状態遷移図
(d)スタートチャート
(c)

図 5

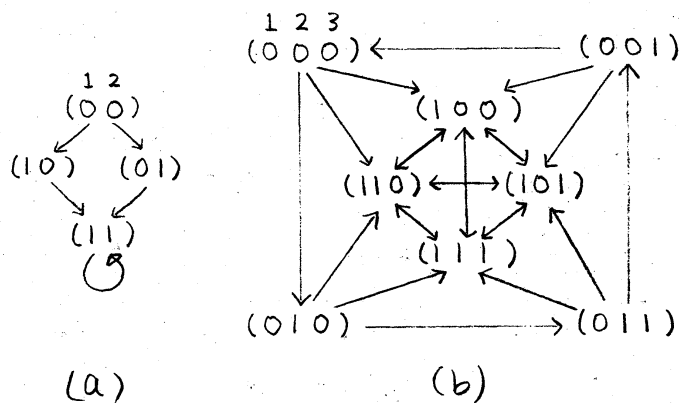


图 6

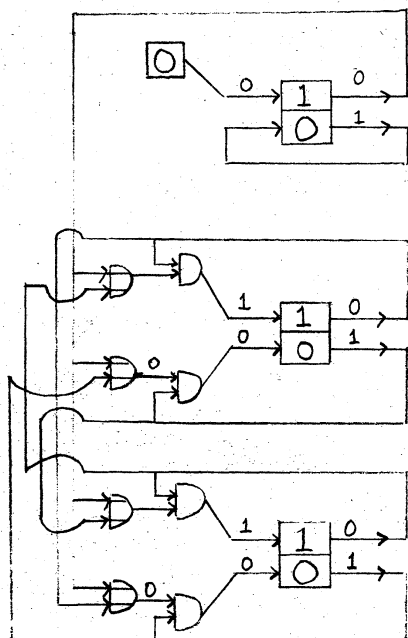
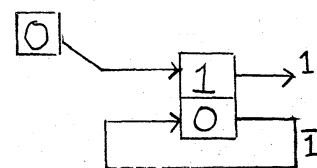
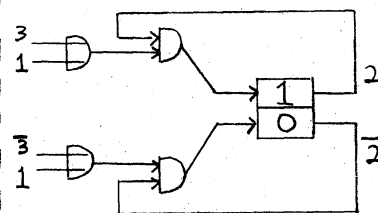


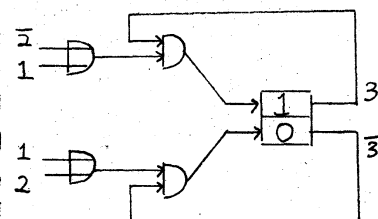
图 8



1 单位回路



2 单位回路



3 单位回路

图 7

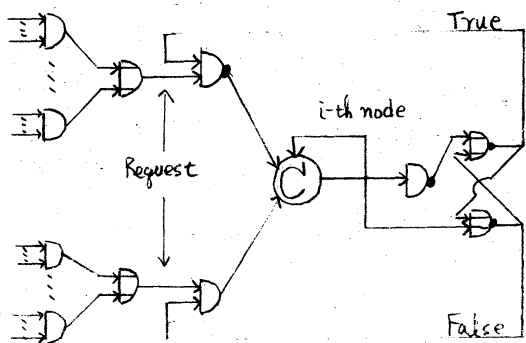


图 9

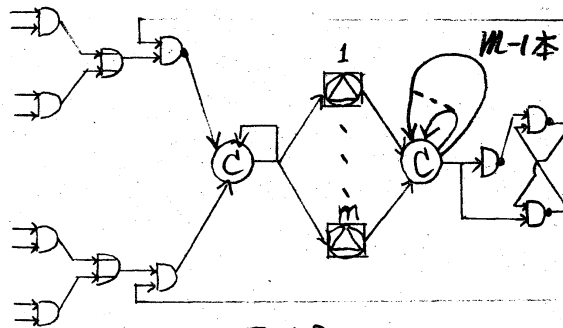


图 10