

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。



# 論理設計

## 東野担当10回目

### 授業スライド

### 12月2日4限

基礎工学部情報科学科 東野輝夫





# 授業時間変更 & 中間試験のお知らせ

- 長谷川先生のご都合で、長谷川先生担当の水曜 3 限の計算機言語の授業と水曜 4 限の東野の論理設計の授業を下記のように交換して実施します.
- 1 1 月 2 5 日 (水)
  - 3 限 計算機言語 → 論理設計 の授業に変更
  - 4 限 論理設計 (この日は 2 コマ続けて論理設計の授業を実施)
- 1 2 月 9 日 (水)
  - 3 限 計算機言語 (この日は 2 コマ続けて計算機言語の授業を実施)
  - 4 限 論理設計 → 計算機言語 の授業に変更





# 授業計画の変更

- 授業計画：

1. ドントケアを含む論理関数の簡単化（6章）
2. フリップフロップとレジスタ（10章）
3. 同期式順序回路（Mealy型, Moore型順序回路）（11章）
4. カルノー図を用いた論理関数の簡単化（1章から5章の復習）
5. 組合せ論理回路設計、よく用いられる組み合わせ回路（7章, 8章）
6. 加減算器とALU、順序回路の簡単化（9章, 12章前半）
7. 演習
8. 順序回路の簡単化、カウンタ（12章後半, 13章）
9. 中間試験（1章～11章）
10. ICを用いた順序回路の実現（15章）
11. 演習
12. CPUの設計（付録）
13. CPUの設計, 演習
14. 乗算器と除算器（14章）
15. 期末試験（12章～15章, 付録）



# 質問について

- メールで随時問い合わせや質問にお答えしますので、何かあれば、[higashino@ist.osaka-u.ac.jp](mailto:higashino@ist.osaka-u.ac.jp) までメールで質問して下さい。
- また、時間を決めてZoomなどを用いて質問にお答えすることも可能ですので、まずはメールで疑問点や問い合わせ事項などを連絡して下さい。





# お願い

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

## 著作権保護

- この授業のテキスト（教科書）や授業スライド、授業ビデオの著作権保護に努めて下さい。
- この授業のビデオやスナップショットを録画したり，それらを他の人に転送したり，インターネット上で公開したりすることを禁止します。
- この授業で利用するスライドにはオーム社の教科書の図などが含まれているので，著作権保護の観点から，この授業スライドの公開につながる行為は謹んでください。
- 来年度は CLE を使ったメディア授業でなく，対面の授業ができることを期待していますが，今年度の演習課題の解答が事前に公開されたりすると，来年度の授業で同じ演習課題が使えなくなり，授業テキストの大幅な修正が必要になるため，協力をお願いします。



# 中間試験問題解答





# 問題 1 (25/100点)

学籍番号下一桁が  $X$  の人は,  $Z = X \bmod 4$  (下一桁の整数  $X$  を 4 で割った余り) を計算し, 「 $Z.$ 」の問題を解答して下さい. 例えば、学籍番号が 09B19090 の人は「0.」を 09B19097 の人は「3.」を解答して下さい.

- 下記の図の「0.」から「3.」のカルノー図に対して, 主項と必須主項 (必須項), 最簡積和形を求めよ. 最簡積和形が複数存在する場合は, それらの最簡積和形を全て列挙せよ.

$x_1x_0$	00	01	11	10
$x_3x_2$ 00				X
01		X	1	1
11		1		1
10				1

「0.」

$x_1x_0$	00	01	11	10
$x_3x_2$ 00				
01		X	1	
11		1		
10	X	1	1	1

「1.」

$x_1x_0$	00	01	11	10
$x_3x_2$ 00				X
01				1
11		1		1
10		X	1	1

「2.」

$x_1x_0$	00	01	11	10
$x_3x_2$ 00				
01			1	X
11				1
10	X	1	1	1

「3.」

- 解答は例えば, 右記のようにカルノー図を記載して, すべての主項に対してどのループがどのような主項に対応するかを図示し, 得られた主項から必須主項 (必須項) を選び, その後に最簡積和形を記載すること. 最簡積和形が複数存在する場合は, 全て列挙すること.

$x_1x_0$	00	01	11	10
$x_3x_2$ 00	X			X
01		X		1
11	1	1	X	1
10	X	1	X	X





# 問題 1 解答「1.」

学籍番号下一桁が  $X$  の人は,  $Z = X \bmod 4$  (下一桁の整数  $X$  を 4 で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば、学籍番号が 09B19090 の人は「0.」を 09B19097 の人は「3.」を解答して下さい.

- 下記の図の「1.」のカルノー図に対して, 主項と必須主項 (必須項), 最簡積和形を求めよ. 最簡積和系が複数存在する場合は, それらの最簡積和形を全て列挙せよ.

(解答)

- 主項

–  $\bar{X}_3 X_2 X_0$ ,  $X_3 \bar{X}_2$ ,  $X_2 \bar{X}_1 X_0$ ,  $X_3 \bar{X}_1 X_0$

- 必須主項 (必須項)

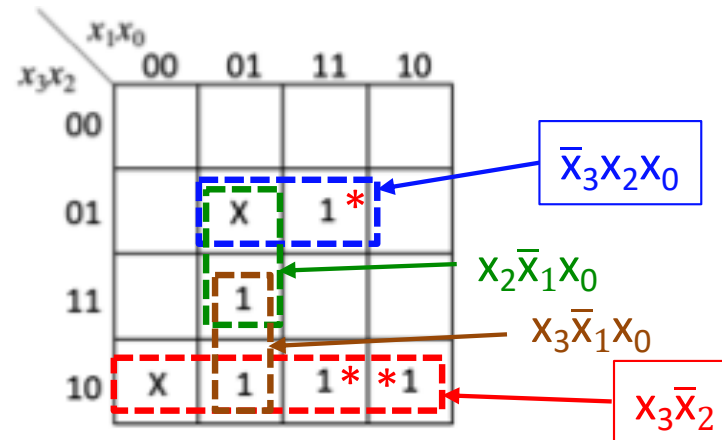
–  $\bar{X}_3 X_2 X_0$ ,  $X_3 \bar{X}_2$

- 最簡積和形

–  $f = \bar{X}_3 X_2 X_0 \vee X_3 \bar{X}_2 \vee X_2 \bar{X}_1 X_0$

または

$f = \bar{X}_3 X_2 X_0 \vee X_3 \bar{X}_2 \vee X_3 \bar{X}_1 X_0$



\*: 特異最小項





## 問題 2 (25/100点)

学籍番号下二桁が YX の人は  $Z=(Y+X) \bmod 4$  (二桁の整数 Y と X の和を4で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば, 学籍番号が 09B19093 の人は「0.」を 09B19095 の人は「2.」を解答して下さい.

- 下記の論理関数  $f(x_2, x_1, x_0)$  の主項をクワイン・マクラスキー法で求めよ. 解答は下記の図 6・6 のような表を書いて求めること (カルノー図を書いて主項を求めてはいけない. この問題はクワイン・マクラスキー法で解答できるかで採点します).
- 0. 「0.」の人は論理関数  $f$  として下記の  $f_0$  を選択すること
- 1. 「1.」の人は論理関数  $f$  として下記の  $f_1$  を選択すること
- 2. 「2.」の人は論理関数  $f$  として下記の  $f_2$  を選択すること
- 3. 「3.」の人は論理関数  $f$  として下記の  $f_3$  を選択すること

$x_2$	$x_1$	$x_0$	$f_0$	$f_1$	$f_2$	$f_3$
0	0	0	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	1	1	1	0	1
1	0	0	1	0	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	1	0	1	1	1

	第一段階のリスト						第二段階のリスト						第三段階のリスト					
	$x_3$	$x_2$	$x_1$	$x_0$	最小項		$x_3$	$x_2$	$x_1$	$x_0$	最小項		$x_3$	$x_2$	$x_1$	$x_0$	最小項	
第一グループ	0	0	0	0	(0)	v	0	0	—	0	(0,2)		0	—	1	—	(2,3,6,7)	
第二グループ	0	0	1	0	(2)	v	—	0	0	0	(0,8)		0	0	1	—	(2,3) v	
	1	0	0	0	(8)	v	0	0	1	—	(2,6) v			0	—	1	0	(2,6) v
第三グループ	0	0	1	1	(3)	v	1	0	0	—	(8,9)		0	—	1	1	(3,7) v	
	0	1	0	1	(5)	v	0	—	1	1	(3,7) v			0	1	—	1	(5,7)
	0	1	1	0	(6)	v	0	1	—	1	(5,7)		0	1	1	—	(6,7) v	
第四グループ	1	0	0	1	(9)	v												
	0	1	1	1	(7)	v												

図 6・6 例題 6.3 の主項の生成





# 問題 2 解答 「0.」

学籍番号下二桁が YX の人は  $Z=(Y+X) \bmod 4$  (二桁の整数 Y と X の和を4で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば, 学籍番号が 09B19093 の人は「0.」を 09B19095 の人は「2.」を解答して下さい.

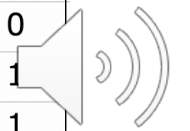
- 下記の論理関数  $f(x_2, x_1, x_0)$  の主項をクワイン・マクラスキー法で求めよ.
- 0. 「0.」の人は論理関数  $f$  として下記の **f0** を選択すること

(解答)

$$\begin{array}{llll} \bar{x}_2 x_0 & 0 & - & 1 \quad (1,3) \\ \bar{x}_1 x_0 & - & 0 & 1 \quad (1,5) \\ x_2 \bar{x}_1 & 1 & 0 & - \quad (4,5) \end{array}$$

	第一段階のリスト				第二段階のリスト			
	x2	x1	x0	最小項	x2	x1	x0	最小項
第一グループ								
第二グループ	0	0	1	(1)	0	-	1	(1,3)
	1	0	0	(4)	-	0	1	(1,5)
第三グループ	0	1	1	(3)	1	0	-	(4,5)
	1	0	1	(5)				
第四グループ								

x2	x1	x0	f0	f1	f2	f3
0	0	0	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	1	1	1	0	1
1	0	0	1	0	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	1	0	1	1	1





# 問題 3 (25/100点)

学籍番号下二桁が YX の人は  $Z = Y \bmod 4$  (10の位の Y を 4 で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば、学籍番号が 09B19089 の人は「0.」を 09B19093 の人は「1.」を解答して下さい.

- 3変数論理関数  $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  は  $(x_2, x_1, x_0)$  を 3 ビットの 2 進数 ( $x_2$  が上位ビット,  $x_0$  が下位ビット) と見なした際に, 次のように定義する. 解答する「0.」から「3.」に応じて,  $f_1$ ,  $f_2$  の定義が異なるので注意すること.
  - 「0.」 2 進数の値が 1, 4, 5 のとき  $f_1 = 1$  で、1, 3, 7 のとき  $f_2 = 1$  とする.
  - 「1.」 2 進数の値が 4, 5, 7 のとき  $f_1 = 1$  で、1, 3, 7 のとき  $f_2 = 1$  とする.
  - 「2.」 2 進数の値が 1, 4, 5 のとき  $f_1 = 1$  で、0, 1, 4 のとき  $f_2 = 1$  とする.
  - 「3.」 2 進数の値が 4, 5, 7 のとき  $f_1 = 1$  で、0, 1, 4 のとき  $f_2 = 1$  とする.
- このとき, 下記の①, ②に答えよ.
  - ①  $f_1$ ,  $f_2$  および  $f_1 \cdot f_2$  それぞれの主項と最簡積和形を求めよ.
  - ② 上の①で求めた主項を用いて、 $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  を同時に実現する最簡多出力論理関数をそれぞれ求めよ.
- 解答は  $f_1$ ,  $f_2$  および  $f_1 \cdot f_2$  それぞれのカルノー図を記載し, どの積項が主項になるかも明記すること.





# 問題3 解答「0.」

学籍番号下二桁が YX の人は  $Z = Y \bmod 4$  (10の位の Y を 4 で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば、学籍番号が 09B19089 の人は「0.」を 09B19093 の人は「1.」を解答して下さい.

- 3変数論理関数  $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  は  $(x_2, x_1, x_0)$  を 3 ビットの 2 進数 ( $x_2$  が上位ビット,  $x_0$  が下位ビット) と見なした際に, 次のように定義する. 解答する「0.」から「3.」に応じて,  $f_1$ ,  $f_2$  の定義が異なるので注意すること.

「0.」 2 進数の値が 1, 4, 5 のとき  $f_1 = 1$  で、1, 3, 7 のとき  $f_2 = 1$  とする.

- ①  $f_1$ ,  $f_2$  および  $f_1 \cdot f_2$  それぞれの主項と最簡積和形を求めよ.
- ② 上の①で求めた主項を用いて、 $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  を同時に実現する最簡多出力論理関数をそれぞれ求めよ.

		$x_1x_0$			
		00	01	11	10
$x_2$	0		1		
	1	1	1		

f1のカルノー図

		$x_1x_0$			
		00	01	11	10
$x_2$	0		1	1	
	1			1	

f2のカルノー図

		$x_1x_0$			
		00	01	11	10
$x_2$	0		1		
	1				

f1・f2のカルノー図

- $f_1$ の主項:  $x_2\bar{x}_1$ ,  $\bar{x}_1x_0$
- $f_2$ の主項:  $\bar{x}_2x_0$ ,  $x_1x_0$
- $f_1 \cdot f_2$ の主項:  $\bar{x}_2\bar{x}_1x_0$
- 最簡多出力論理関数:
- 最簡多出力論理関数:

$$f_1 = x_2\bar{x}_1 \vee \bar{x}_1x_0$$

$$f_2 = \bar{x}_2x_0 \vee x_1x_0$$

$$f_1 \cdot f_2 = \bar{x}_2\bar{x}_1x_0$$

$$f_1 = x_2\bar{x}_1 \vee \bar{x}_2\bar{x}_1x_0$$

$$f_2 = x_1x_0 \vee \bar{x}_2\bar{x}_1x_0$$



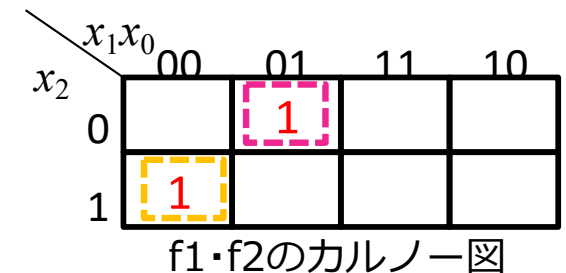
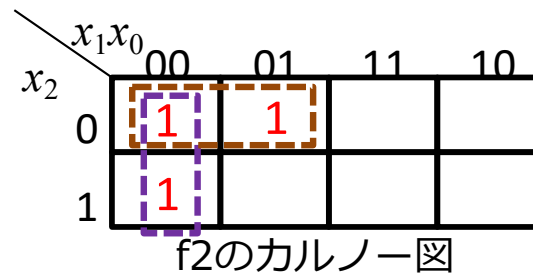
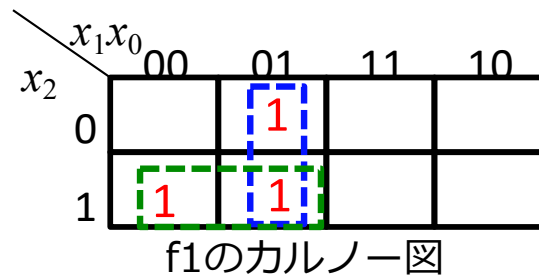


# 問題3 解答「2.」

学籍番号下二桁が YX の人は  $Z = Y \bmod 4$  (10の位の Y を 4 で割った余り) を計算し, 「Z.」の問題を解答して下さい. 例えば, 学籍番号が 09B19089 の人は「0.」を 09B19093 の人は「1.」を解答して下さい.

- 3変数論理関数  $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  は  $(x_2, x_1, x_0)$  を 3 ビットの 2 進数 ( $x_2$  が上位ビット,  $x_0$  が下位ビット) と見なした際に, 次のように定義する. 解答する「0.」から「3.」に応じて,  $f_1$ ,  $f_2$  の定義が異なるので注意すること.  
「2.」 2 進数の値が 1, 4, 5 のとき  $f_1 = 1$  で、0, 1, 4 のとき  $f_2 = 1$  とする.

- ①  $f_1$ ,  $f_2$  および  $f_1 \cdot f_2$  それぞれの主項と最簡積和形を求めよ.
- ② 上の①で求めた主項を用いて、 $f_1(x_2, x_1, x_0)$ ,  $f_2(x_2, x_1, x_0)$  を同時に実現する最簡多出力論理関数をそれぞれ求めよ.



–  $f_1$ の主項:  $x_2 \bar{x}_1$ ,  $\bar{x}_1 x_0$

$$f_1 = x_2 \bar{x}_1 \vee \bar{x}_1 x_0$$

–  $f_2$ の主項:  $\bar{x}_2 \bar{x}_1$ ,  $\bar{x}_1 \bar{x}_0$

$$f_2 = \bar{x}_2 \bar{x}_1 \vee \bar{x}_1 \bar{x}_0$$

–  $f_1 \cdot f_2$ の主項:  $\bar{x}_2 \bar{x}_1 x_0$ ,  $x_2 \bar{x}_1 \bar{x}_0$

$$f_1 \cdot f_2 = \bar{x}_2 \bar{x}_1 x_0 \vee x_2 \bar{x}_1 \bar{x}_0$$

– 最簡多出力論理関数:  $f_1 = x_2 \bar{x}_1 \vee \bar{x}_2 \bar{x}_1 x_0$  または  $f_1 = \bar{x}_1 x_0 \vee x_2 \bar{x}_1 \bar{x}_0$

– 最簡多出力論理関数:  $f_2 = \bar{x}_1 \bar{x}_0 \vee \bar{x}_2 \bar{x}_1 x_0$  または  $f_2 = \bar{x}_2 \bar{x}_1 \vee x_2 \bar{x}_1 \bar{x}_0$



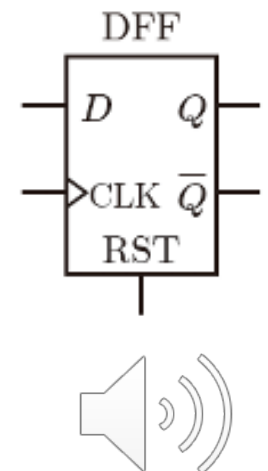




## 問題 4 (25/100点)

学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.
  - 「0.」  $N$  が 4 で割り切れるとき, 出力  $z$  の値が 1 になる
  - 「1.」  $N$  を 4 で割ったときの余りが 1 のとき, 出力  $z$  の値が 1 になる
  - 「2.」  $N$  を 4 で割ったときの余りが 2 のとき, 出力  $z$  の値が 1 になる
- このとき, 次の①, ②に答えよ.
  - ① この回路の状態遷移図を書け. 但し初期状態を  $s_0$  とすること.
  - ② この回路を右図のような 2 個の D フリップフロップ  $DFF_1, DFF_0$  と何個かの AND, OR, NOT ゲートを用いて実現せよ. 但し  $DFF_1, DFF_0$  の D 入力をそれぞれ  $D_1$  および  $D_0$  とし、 $DFF_1, DFF_0$  の出力を  $Q_1, Q_0$  とする ( $Q_1, Q_0$  の否定  $\bar{Q}_1, \bar{Q}_0$  も利用可能). 解答は, カルノー図を書いて,  $D_1, D_0$  および出力  $z$  を, それぞれ  $Q_1, Q_0$  ならびに入力  $x$  の論理式 (最簡積和形) で表せ (回路図を記載する必要はない). また, 状態  $s_0, s_1, s_2, s_3$  をそれぞれ  $\langle Q_1, Q_0 \rangle = \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 0 \rangle$  に割り当てること.





# 問題 4 解答

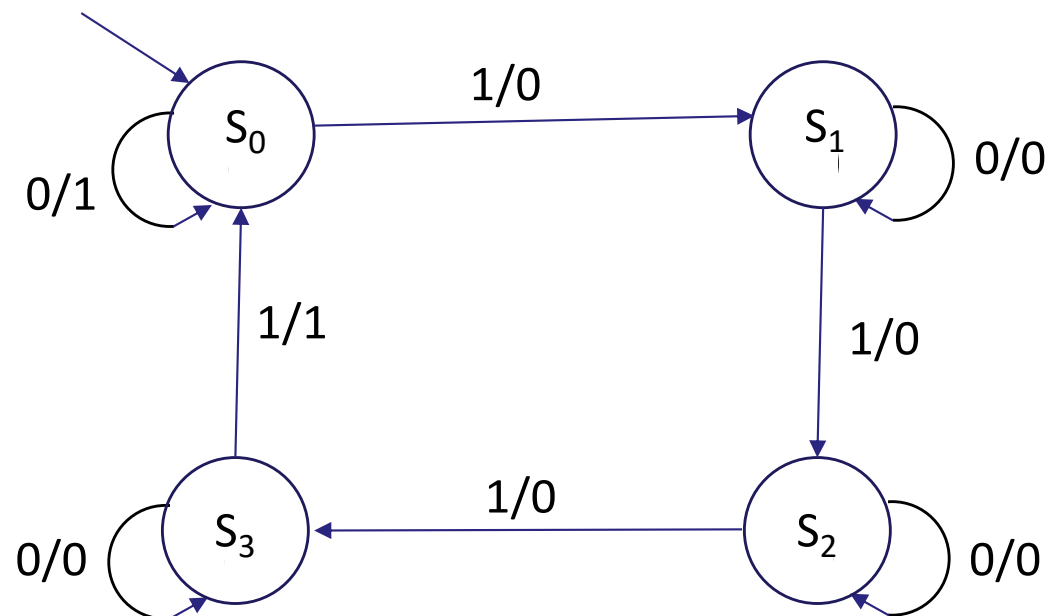
学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.

「0.」  $N$  が 4 で割り切れるとき, 出力  $z$  の値が 1 になる

(解答)

- この回路の状態遷移図を書け. 但し初期状態を  $s_0$  とすること.





# 問題 4 解答

学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.

「0.」  $N$  が 4 で割り切れるとき, 出力  $z$  の値が 1 になる

(解答)

- ② この回路を  $DFF_1, DFF_0$  を用いて実現せよ.  $D_1, D_0$  および出力  $z$  を, それぞれ  $Q_1, Q_0$  ならびに入力  $x$  の論理式で表せ. 状態  $s_0, s_1, s_2, s_3$  をそれぞれ  $\langle Q_1, Q_0 \rangle = \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 0 \rangle$  に割り当てること.

$$- D_1 = Q_1^+ = xQ_0 \vee \bar{x}Q_1$$

$$- D_0 = Q_0^+ = x\bar{Q}_1 \vee \bar{x}Q_0$$

$$- z = \bar{x}\bar{Q}_1\bar{Q}_0 \vee xQ_1\bar{Q}_0$$

$Q_1^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0			1	1
1		1	1	

$Q_0^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0		1	1	
1	1	1		

$z$ $x$	$Q_1Q_0$			
	00	01	11	10
0	1			
1				1







# 問題 4 解答

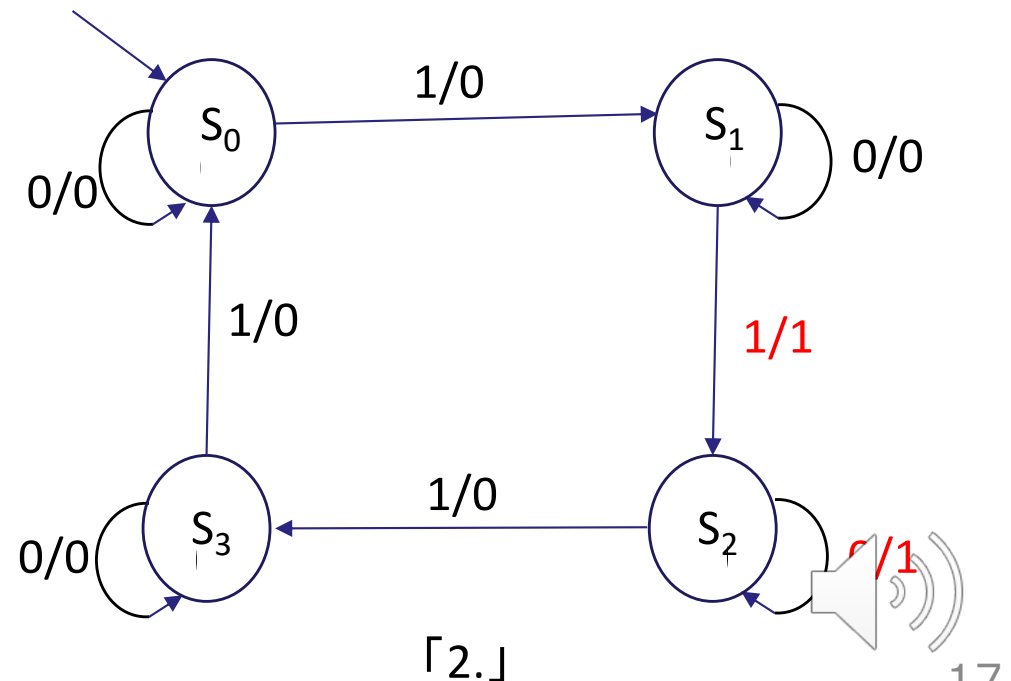
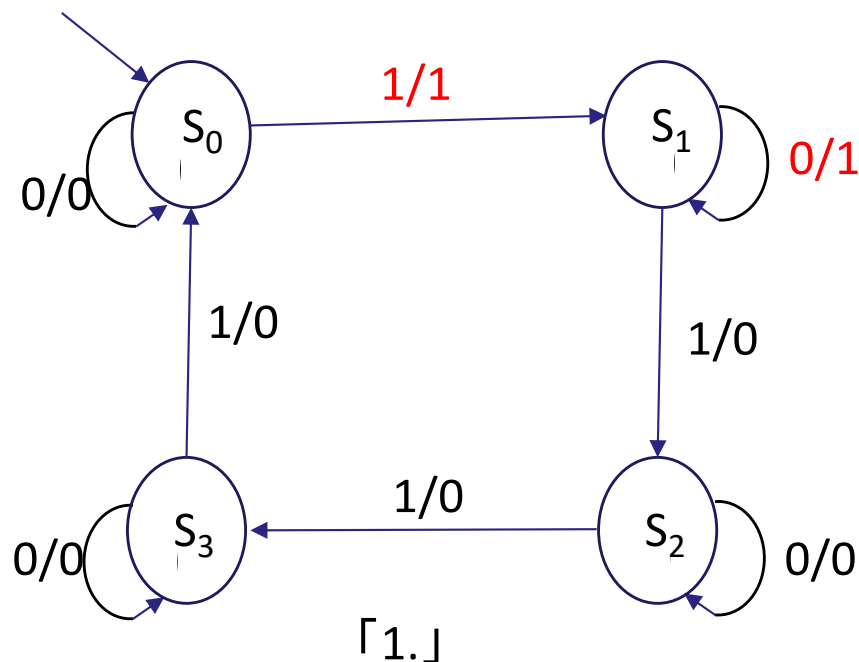
学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.

「0.」  $N$  が 4 で割り切れるとき, 出力  $z$  の値が 1 になる

「1.」  $N$  を 4 で割ったときの余りが 1 のとき, 出力  $z$  の値が 1 になる

「2.」  $N$  を 4 で割ったときの余りが 2 のとき, 出力  $z$  の値が 1 になる





# 問題 4 解答

学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.

「1.」  $N$  を 4 で割ったときの余りが 1 のとき, 出力  $z$  の値が 1 になる

(解答)

- ② この回路を  $DFF_1, DFF_0$  を用いて実現せよ.  $D_1, D_0$  および出力  $z$  を, それぞれ  $Q_1, Q_0$  ならびに入力  $x$  の論理式で表せ. 状態  $s_0, s_1, s_2, s_3$  をそれぞれ  $\langle Q_1, Q_0 \rangle = \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 0 \rangle$  に割り当てること.

$$- D_1 = Q_1^+ = xQ_0 \vee \bar{x}Q_1$$

$$- D_0 = Q_0^+ = x\bar{Q}_1 \vee \bar{x}Q_0$$

$$- z = \bar{x}\bar{Q}_1Q_0 \vee x\bar{Q}_1\bar{Q}_0$$

$Q_1^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0			1	1
1		1	1	

$Q_0^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0		1	1	
1	1	1		

$z$ $x$	$Q_1Q_0$			
	00	01	11	10
0		1		
1	1			





# 問題 4 解答

学籍番号下一桁  $X$  が  $X=0,5,9$  の人は「0.」の問題を,  $X=1,3,6$  の人は「1.」の問題を,  $X=2,4,7,8$  の人は「2.」の問題を解答して下さい. 例えば、学籍番号が 09B19093 の人は「1.」を 09B19097 の人は「2.」を解答して下さい.

- 次のような 1 入力 1 出力のMealy型同期式順序回路を作成したい. 入力  $x$  として動作開始時からその時点までに入力された 1 の個数 (その時点の入力も含める) が  $N$  であるとする. 「0.」から「2.」の出力  $z$  を次のように定義する.

「2.」  $N$  を 4 で割ったときの余りが 2 のとき, 出力  $z$  の値が 1 になる

(解答)

- ② この回路を  $DFF_1, DFF_0$  を用いて実現せよ.  $D_1, D_0$  および出力  $z$  を, それぞれ  $Q_1, Q_0$  ならびに入力  $x$  の論理式で表せ. 状態  $s_0, s_1, s_2, s_3$  をそれぞれ  $\langle Q_1, Q_0 \rangle = \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 1, 0 \rangle$  に割り当てること.

$$- D_1 = Q_1^+ = xQ_0 \vee \bar{x}Q_1$$

$$- D_0 = Q_0^+ = x\bar{Q}_1 \vee \bar{x}Q_0$$

$$- z = \bar{x}Q_1Q_0 \vee x\bar{Q}_1Q_0$$

$Q_1^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0			1	1
1		1	1	

$Q_0^+$ $x$	$Q_1Q_0$			
	00	01	11	10
0		1	1	
1	1	1		

$z$ $x$	$Q_1Q_0$			
	00	01	11	10
0			1	
1		1		





# 第15章 ICを用いた順序回路 の実現





# 第15章

## ICを用いた順序回路の実現

- この章のねらい

### 15章 ICを用いた順序回路の実現

本章では，メモリやレジスタ，加減算器，比較回路などの機能モジュールを組み合わせて，簡単な電卓や自動販売機の制御部の回路を同期式順序回路としてどのように設計するかについて説明する．





# 簡単な電卓の設計

- 簡単な電卓（加算のみ）の設計
  - 以下，レジスタや加減算器，比較回路などの機能モジュールを組み合わせて，同期式順序回路をどのように設計するかについて述べる．
  - LEDに表示される電卓の値を保持するレジスタR と加算の途中結果を保存する補助レジスタ（メモリ）M が用意されている。
  - ボタンとして C , + , = , 0 , … , 9 があり，クリアボタン C はレジスタ R の値（電卓のLED 表示）を 0 にするボタンであり， + , = はそれぞれ加算と合計を計算するボタンである．
  - 1, 2, 3, +, 4, 5, +, 1, 0, = がこの順に押されると，レジスタ R の値（電卓の LED 表示）が図15・1 のように変化する．

	<span>1</span>	<span>2</span>	<span>3</span>	<span>+</span>	<span>4</span>	<span>5</span>	<span>+</span>	<span>1</span>	<span>0</span>	<span>=</span>
レジスタ R	1	12	123	123	4	45	168	1	10	178
メモリ M	0	0	0	123	123	123	168	168	168	178

	<span>1</span>	<span>2</span>	<span>3</span>	<span>+</span>	<span>+</span>	<span>1</span>	<span>0</span>	<span>=</span>	<span>=</span>
レジスタ R	1	12	123	123	246	1	10	256	512
メモリ M	0	0	0	123	246	246	246	256	512

図 15・1 電卓の LED 表示





# 設計の手順

- 設計の手順
  - i. 回路構成と使用部品の概略設計：まず最初に実現したい同期式順序回路の回路構成と回路で利用する部品の概略設計を行う。
  - ii. 動作アルゴリズムの詳細設計：上記 i. で概略設計した回路や使用部品を用いて、与えられた問題の動作アルゴリズムの詳細設計を行う。場合によっては、考案した動作アルゴリズムで不足する部品を追加したり、回路構成を修正する。
  - iii. 回路で利用する部品の動作仕様の決定：上記 ii. で定めた動作アルゴリズムの実現に必要な各部品の動作仕様（制御信号の値とその動作）を決定する。
  - iv. 同期式順序回路としての実現：考案した動作アルゴリズムどおりに動作する同期式順序回路を設計し、設計した回路の各部品の制御信号の値を指定する。
- 同期式順序回路としての実現方法には、**Moore 型順序回路として実現する方法**と、**マイクロプログラム方式で実現する方法**の二つがある。以下、順に設計の手順の詳細を説明する。



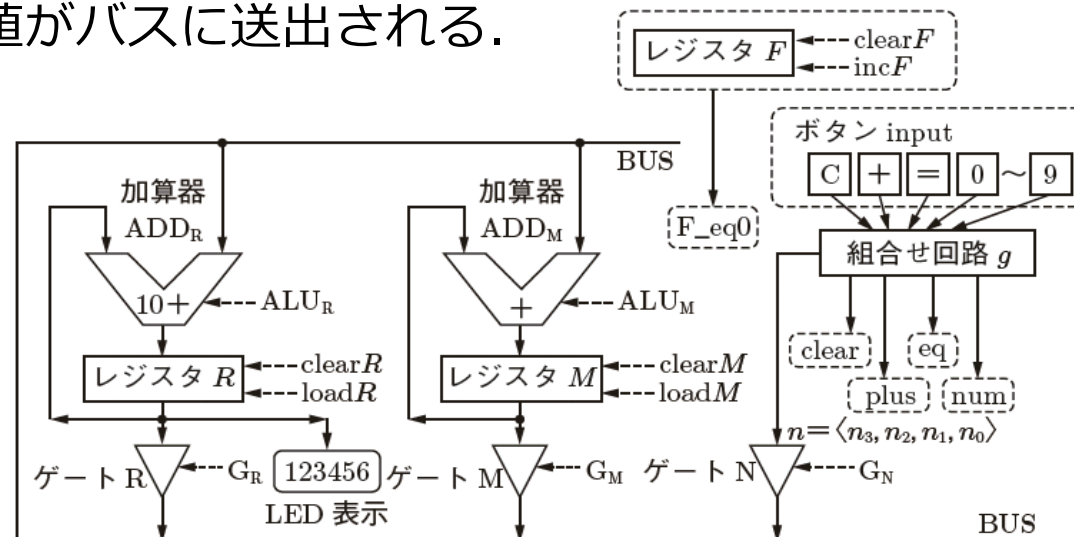


OSAKA UNIVERSITY

# 回路構成と使用部品の動作仕様の設計

## 回路構成の設計

- まず最初に実現したい同期式順序回路の回路構成と動作アルゴリズムを設計する. 例として 図15.2 のような電卓の回路構成を考える.
- 数字のボタン 0, ..., 9 が押されると, 組合せ論理回路  $g$  を介して, 押された数字  $n$  の 2 進数  $n = \langle n_3, n_2, n_1, n_0 \rangle$  がゲート  $N$  に送られる.
- 入力  $n$  やレジスタ間のデータの転送は **バス (BUS)** と呼ばれる共通の経路を介して行われる. ゲート  $R$ , ゲート  $M$ , ゲート  $N$  の制御信号  $G_R, G_M, G_N$  の値を排他的に 1 にすることで, レジスタ  $R, M$ , 入力  $n$  のいずれかの値がバスに送出される.



バス(BUS)は  
マルチプレクサ  
で実現される

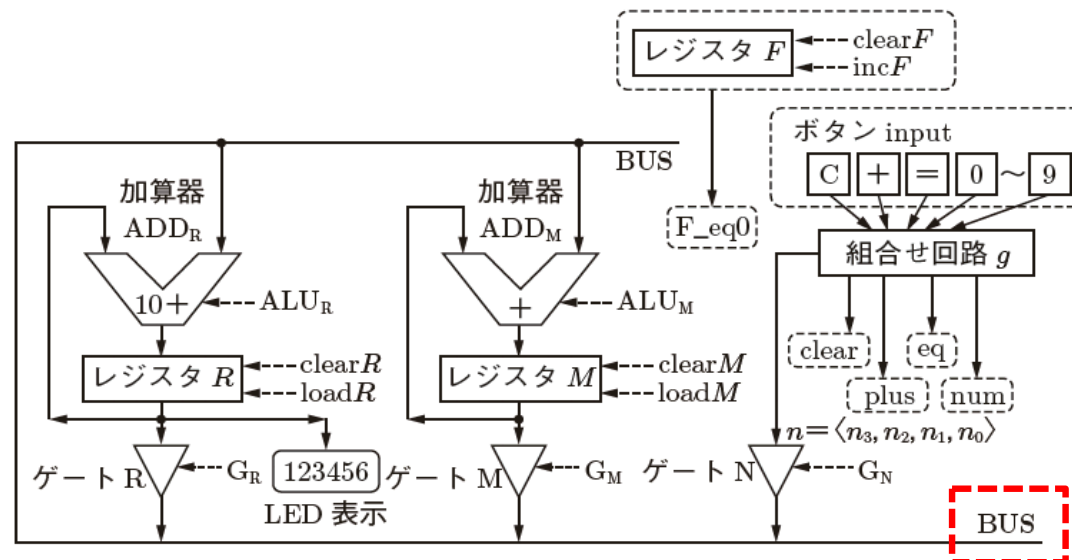
図 15.2 電卓の回路構成





# バス

- 同期式順序回路の内部には、加減算回路、レジスタ、メモリ、周辺回路などのさまざまなモジュールが存在する．あるモジュールから他のモジュールにデータを転送するには、モジュールの間にデータ転送路を設ける必要がある．
- データ転送は演算回路とメモリもしくは周辺回路の間で行われることが多い．このような場合には、複数の周辺回路から演算回路へのデータ転送要求があった場合に、その中から一つだけを選択して転送する仕組みが必要となる．このようなデータ転送路を**共有バス（common bus）**と呼ぶ．



バス(BUS)は  
マルチプレクサ  
で実現される

図 15・2 電卓の回路構成



# セクタ (マルチプレクサ)

- **マルチプレクサ (multiplexer)** は,  $n$  ビットの制御入力  $s_{n-1}, \dots, s_0$  の値によって,  $2^n$ 個のデータの入力  $i_{2^n-1}, \dots, i_0$  のいずれかを出力  $o$  に出力する回路である.
- マルチプレクサは **セクタ (selector)** とも呼ばれる.
- $n = 2$  とし,  $(s_1, s_0) = (0, 0), (0, 1), (1, 0), (1, 1)$  のときにそれぞれ  $i_0, i_1, i_2, i_3$  を  $o$  に出力するマルチプレクサの出力  $o$  の積和形論理式は (図8・3)

$$o = i_0 \bar{s}_1 \bar{s}_0 \vee i_1 \bar{s}_1 s_0 \vee i_2 s_1 \bar{s}_0 \vee i_3 s_1 s_0$$

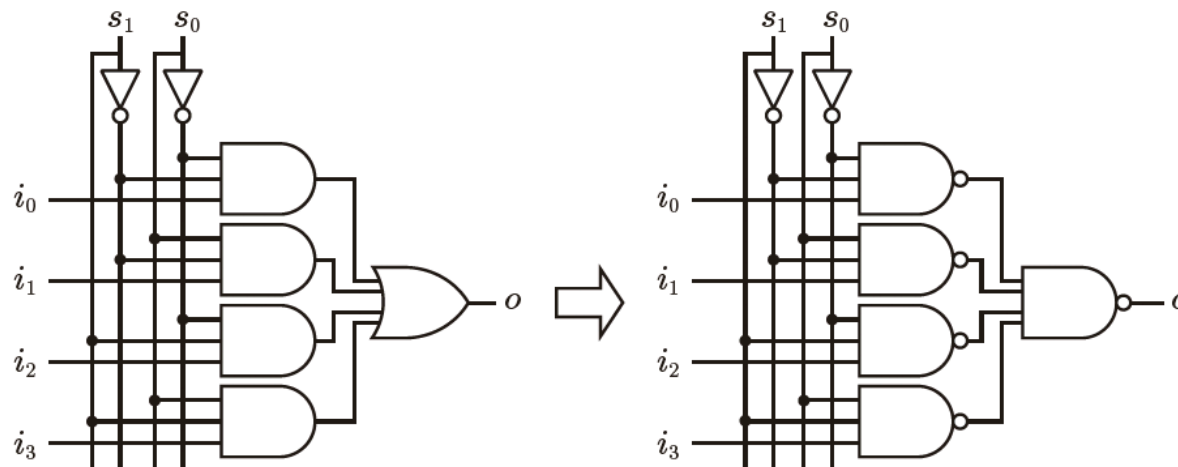


図 8・3 4 入力マルチプレクサの回路図



# 例題15・1

## ( $10 \times R + n$ ) の値を計算する 組合せ論理回路

- ( $10 \times R + n$ ) の値を計算する 組合せ論理回路
  - $Add_k(x, y)$  を  $k$  ビットの 2 進数  $x = (x_{k-1}, \dots, x_1, x_0)$ ,  $y = (y_{k-1}, \dots, y_1, y_0)$  の和  $x + y$  を計算する組合せ論理回路とする.
  - $Add_k(x, y)$  を組み合わせて ( $10 \times R + n$ ) の値を計算する組合せ論理回路  $ADD_R(R, n) = 10 \times R + n$  を実現せよ.
  - ただし, レジスタ  $R = (R_{k-1}, \dots, R_1, R_0)$ , BUSからの入力  $n = (n_{k-1}, \dots, n_1, n_0)$  とし ( $R_0, n_0$  が最下位ビット),  $10 \times R + n$  の値も  $k$  ビットで表せるものとする (桁あふれは生じないものとする).

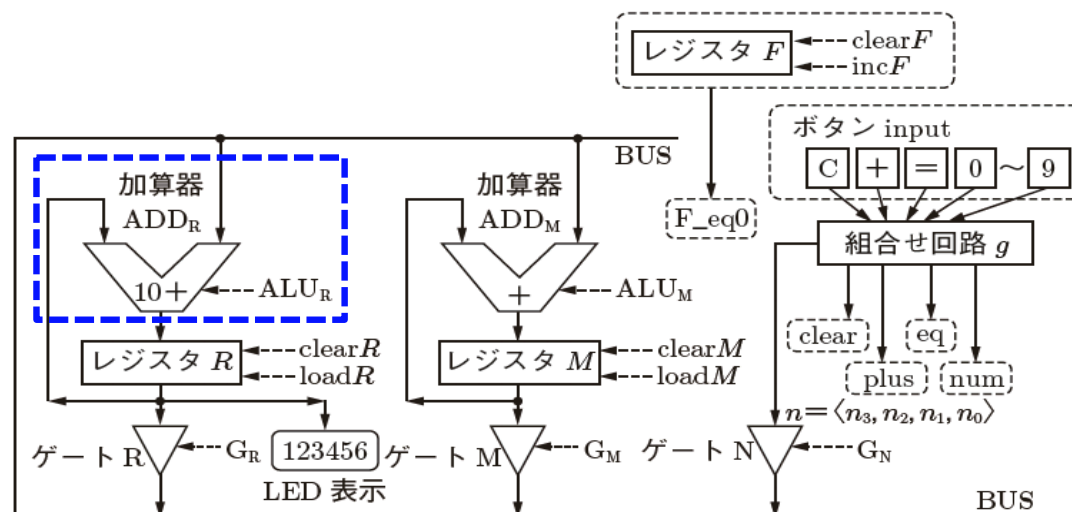


図 15・2 電卓の回路構成



# 例題15・1

## ( $10 \times R + n$ ) の値を計算する 組合せ論理回路

- $Add_k(x, y)$  を組み合わせて ( $10 \times R + n$ ) の値を計算する組合せ論理回路  $ADD_R(R, n) = 10 \times R + n$  を実現せよ.
  - $ADD_R(R, n) = 10 \times R + n = 8 \times R + 2 \times R + n$   
 $= Add_k(Add_k((R_{k-4}, \dots, R_0, 0, 0, 0), (R_{k-2}, \dots, R_0, 0)), (n_{k-1}, \dots, n_1, n_0))$
  - すなわち,  $R = (R_{k-1}, \dots, R_0)$  を 3 ビット左にシフトした値 ( $8 \times R$ ) と  $R$  を 1 ビット左にシフトした値 ( $2 \times R$ ) を加算し, それらの和と  $n$  を加算することで,  $ADD_R(R, n) = 10 \times R + n$  が実現できる.

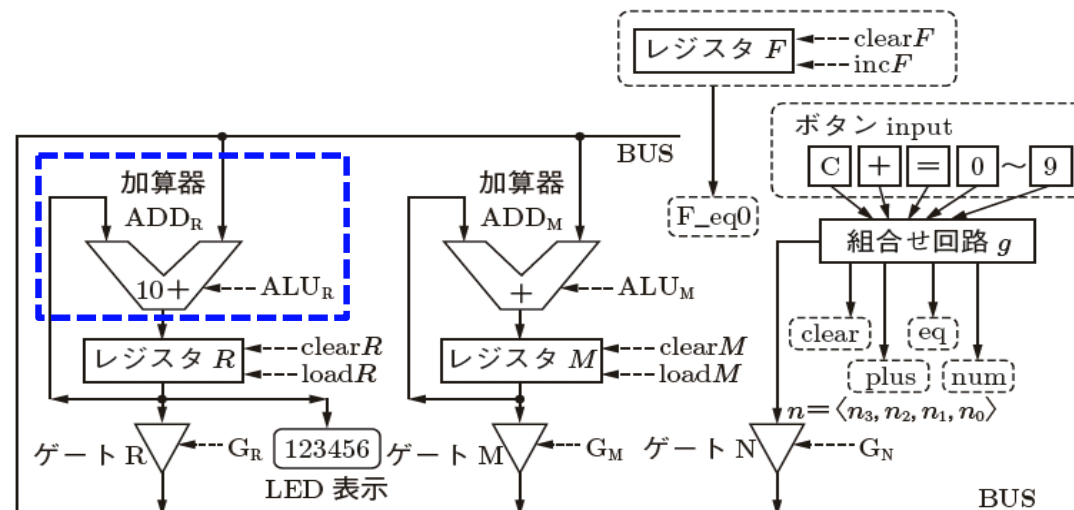


図 15・2 電卓の回路構成





OSAKA UNIVERSITY

# 使用部品の動作仕様の設計

## 使用部品の動作仕様の設計

- 次に図15・2 の回路で使われている部品の動作仕様，すなわち，部品の動作と制御信号の関係を定義する．
- 図15・2 では，各部品の制御信号は点線の矢印で表示されている．例えば，レジスタR は **clearR信号** を 1 にすることで R の値が 0 になり，**loadR信号** を 1 にすることで外部入力（加算器ADD<sub>R</sub>の出力）が R にセットされるものとする．

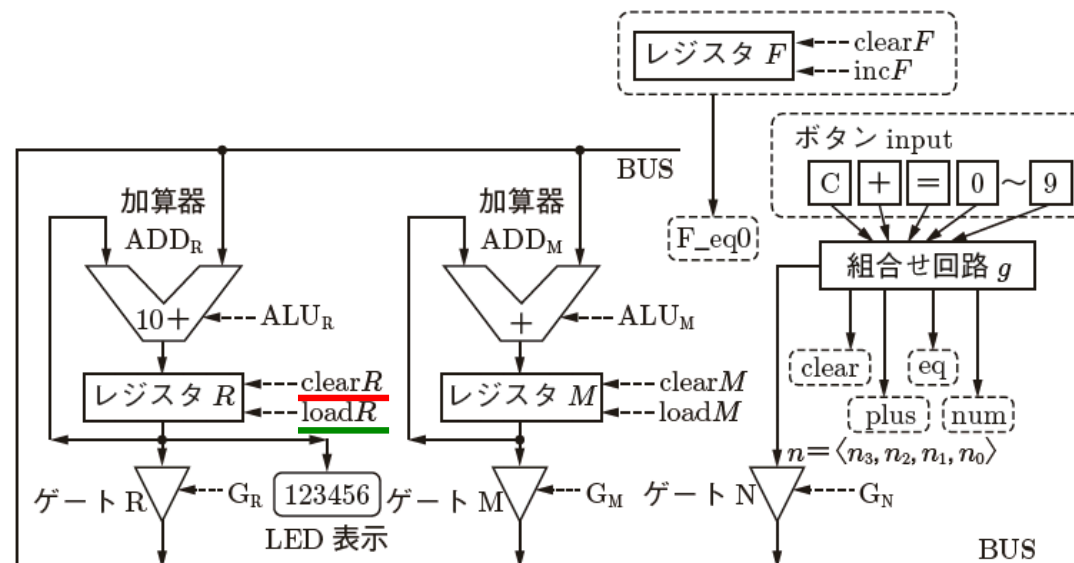


図 15・2 電卓の回路構成





OSAKA UNIVERSITY

## (a) レジスタR

- $\text{clearR} = 1, \text{loadR} = 0$  のとき, レジスタR の値を 0 にする ( $R \leftarrow 0$ )
- $\text{clearR} = 0, \text{loadR} = 1, \text{ALU}_R = 0, G_M = 1$  のとき, レジスタ M の値をレジスタR に転送する ( $R \leftarrow M$ )
- $\text{clearR} = 0, \text{loadR} = 1, \text{ALU}_R = 0, G_N = 1$  のとき, 入力N の値をレジスタR に転送する ( $R \leftarrow N$ )
- $\text{clearR} = 0, \text{loadR} = 1, \text{ALU}_R = 1, G_N = 1$  のとき,  $10 \cdot R + n$  の値をレジスタR に代入する ( $R \leftarrow 10 \cdot R + n$ )

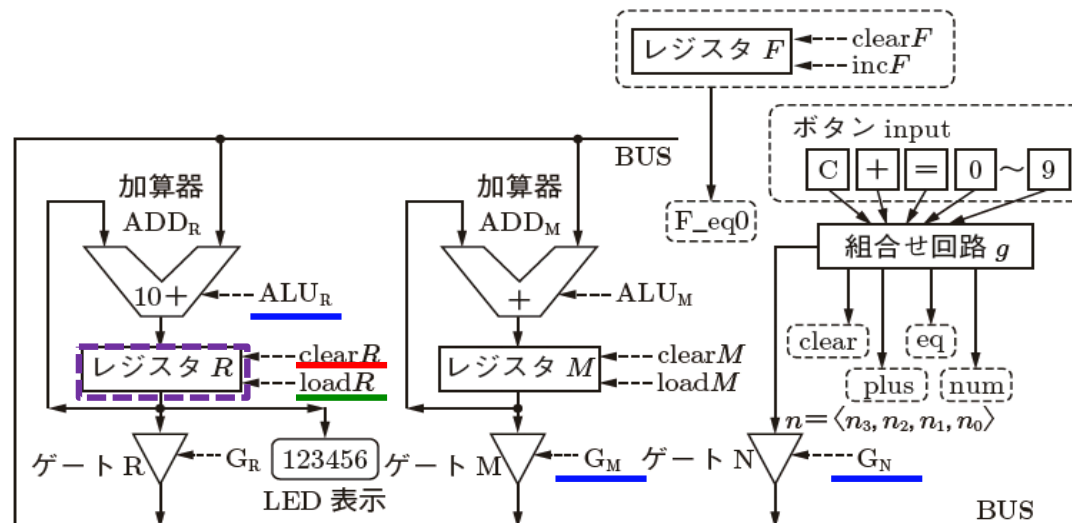


図 15・2 電卓の回路構成





# 使用部品の動作仕様

## (b) レジスタM

- $\text{clearM} = 1$ ,  $\text{loadM} = 0$  のとき, レジスタM の値を 0 にする ( $M \leftarrow 0$ )
- $\text{clearM} = 0$ ,  $\text{loadM} = 1$ ,  $\text{ALU}_M = 1$ ,  $G_R = 1$  のとき,  $M+R$  の値をレジスタM に転送する ( $M \leftarrow M+R$ )

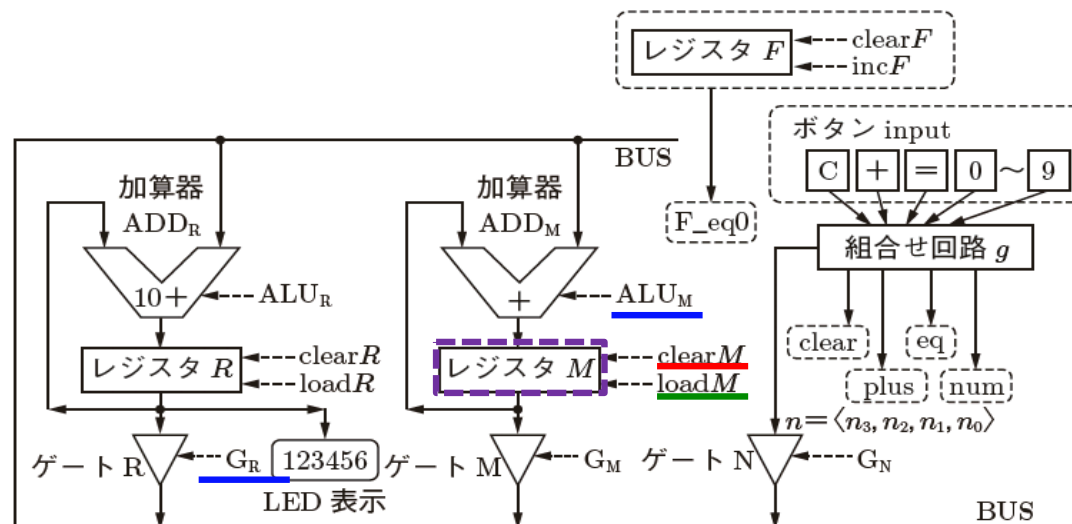


図 15・2 電卓の回路構成



OSAKA UNIVERSITY

## (c) レジスタF

- $\text{clear}F = 1, \text{inc}F = 0$  のとき, レジスタF の値を 0 にする ( $F \leftarrow 0$ )
- $\text{clear}F = 0, \text{inc}F = 1$  のとき, レジスタF の値を 1 増やす ( $F \leftarrow F + 1$ )
- レジスタF の値が 0 のとき,  $F\_eq0$  信号の値が 1 になる ( $F = 0$ )

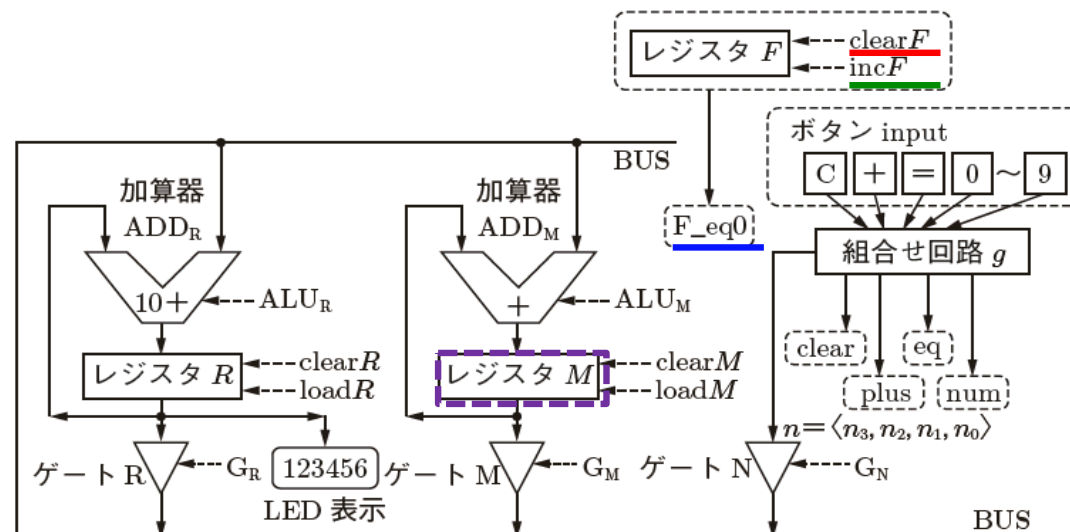


図 15・2 電卓の回路構成







OSAKA UNIVERSITY

## (d) 組合せ回路g

- ボタン C, +, =, 0, ..., 9 の入力を input とすると
- input = C のとき, 回路 g の出力信号 clear の値が 1 になる (clear=1)
- input = + のとき, 回路 g の出力信号 plus の値が 1 になる (plus = 1)
- input = = のとき, 回路 g の出力信号 eq の値が 1 になる (eq = 1)
- input = 0, ..., 9 のいずれかのとき, 回路 g の出力信号 num の値が 1 になり (num=1), 押された数字 n の 2 進数  $n = \langle n_3, n_2, n_1, n_0 \rangle$  がゲート N に送られる

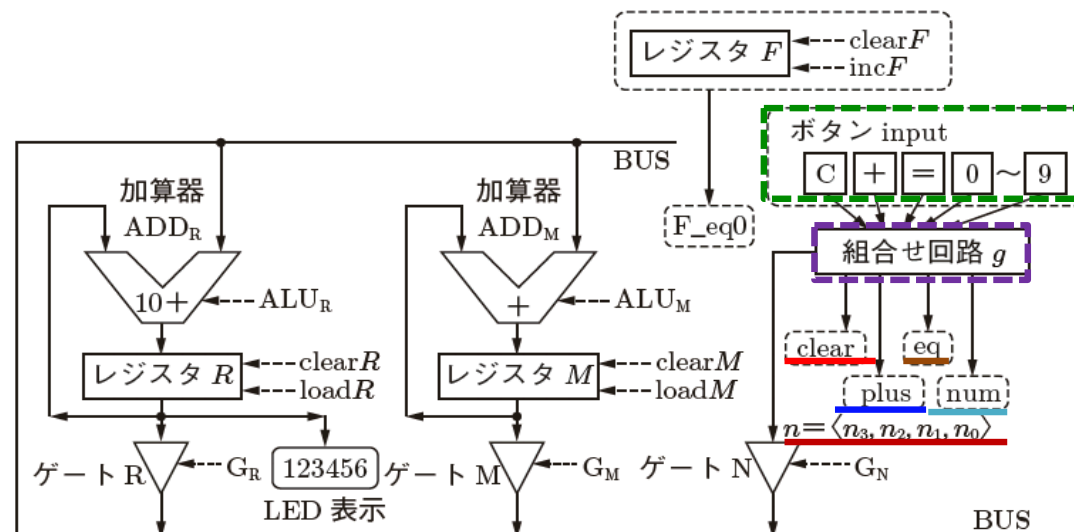


図 15・2 電卓の回路構成





OSAKA UNIVERSITY

## (d) 組合せ回路g

- input = C のとき, 回路 g の出力信号 clear の値が 1 になる (clear=1)
  - clear = C
- input = + のとき, 回路 g の出力信号 plus の値が 1 になる (plus = 1)
  - plus = +
- input = = のとき, 回路 g の出力信号 eq の値が 1 になる (eq = 1)
  - eq = =
- input = 0, ..., 9 のいずれかのとき, 回路 g の出力信号 num の値が 1 になり (num=1),
  - num = 1 v ... v 9
- $n = \langle n_3, n_2, n_1, n_0 \rangle$  の  
 $n_3, n_2, n_1, n_0$  はそれぞれ
  - $n_3 = 8 \vee 9$
  - $n_2 = 4 \vee 5 \vee 6 \vee 7$
  - $n_1 = 2 \vee 3 \vee 6 \vee 7$
  - $n_0 = 1 \vee 3 \vee 5 \vee 7 \vee 9$

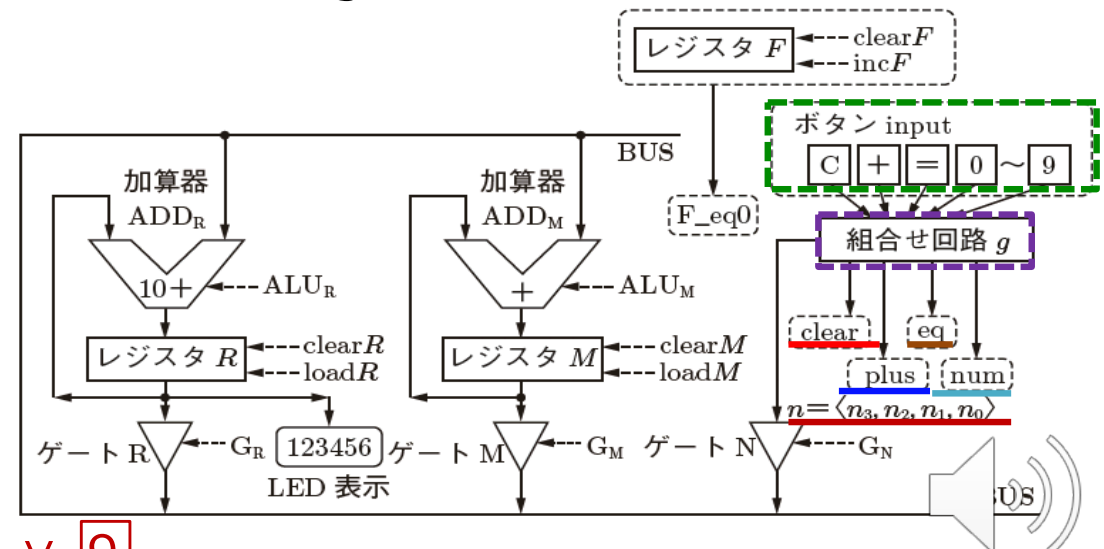


図 15・2 電卓の回路構成



# 電卓の動作アルゴリズム

- 図15.2 の回路構成や上述の部品の動作仕様をもとに，電卓の動作アルゴリズムのフローチャートを図15.3 のように作成する.
  - フローチャートはソフトウェアのフローチャートと類似しており，図15.3 の ①~⑧ のように条件分岐（図中の◇）と処理（図中の□）を組み合わせる記述する.
  - フローチャートの条件分岐（図中の◇）には，“clear = 1?” や “plus = 1 or eq = 1?”，“F = 0?” のような条件文を記述し，処理（図中の□）には，“R←0”，“R←M”，“M←M+R” のような代入文を記述する
  - 同時に実行可能な代入文については，同じ処理中（図中の□）に複数の代入文を記述してよい．これらの代入文は同時に実行される．ただし BUS につながっている複数のレジスタを同時に使用することはできない

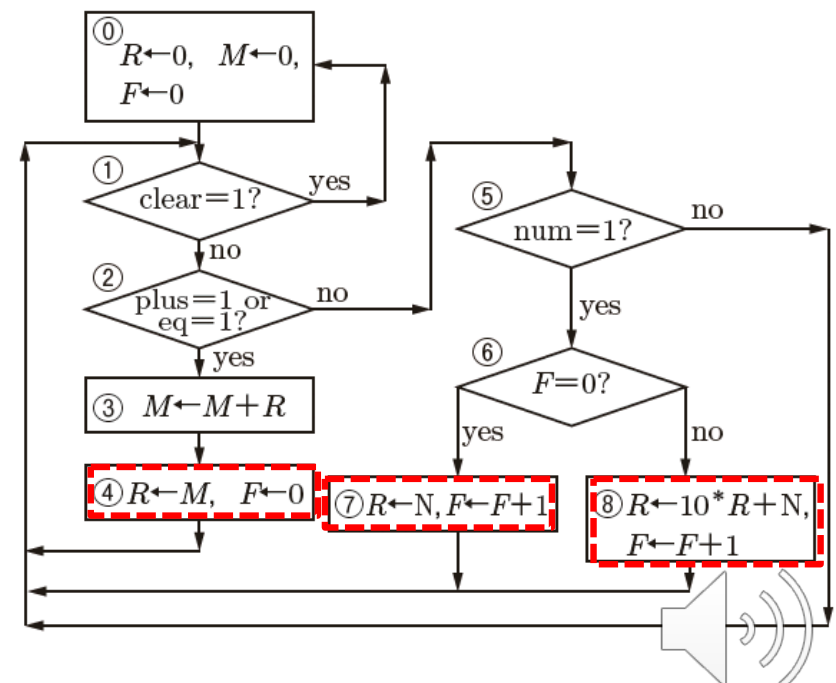


図 15・3



OSAKA UNIVERSITY

# Moore型順序回路としての実現

- 図15.3 のフローチャートの①~⑧の ◇ や □ をそれぞれ同期式順序回路の状態と考えると, 図15.2 の回路は 図15.4 のようなクロックを入力として各部品に **制御信号** を出力とする Moore 型の同期式順序回路とみなせる.
- ただしここでの状態は, ①~⑧ の9つの状態とレジスタR, M, Fの値, ならびに, 電卓のどのボタンが押されたかを表す組合せ論理回路  $g(\text{input})$  の出力群の組である.

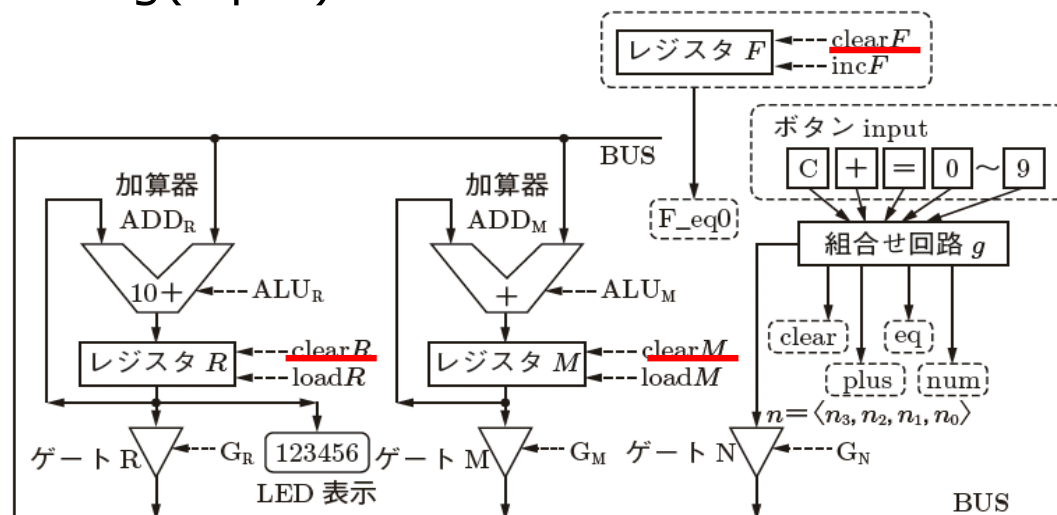


図 15・2 電卓の回路構成

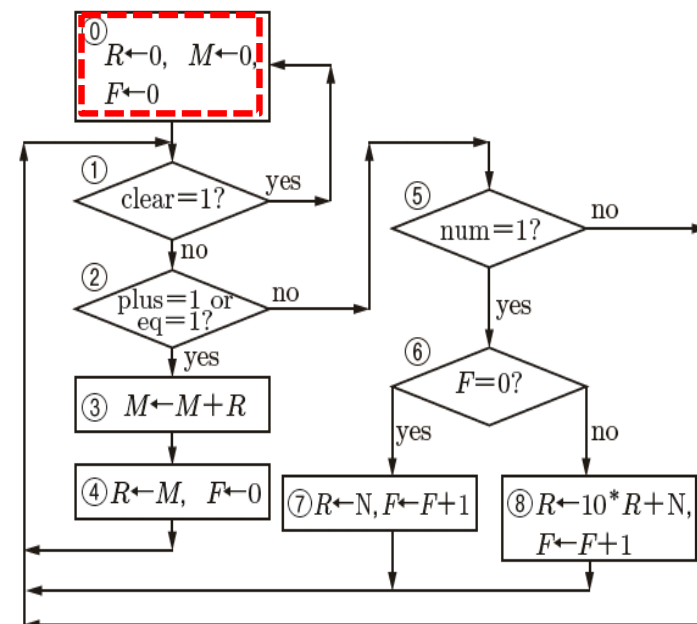


図 15・3 電卓の動作アルゴリズム

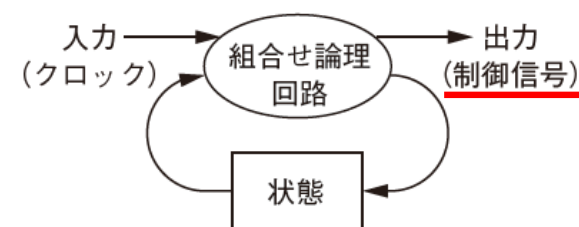


図 15・4 Moore 型の同期式順序回路としての実現



# Moore型順序回路としての実現

- 図15.3 のフローチャートの①~⑧の ◇ や □ をそれぞれ同期式順序回路の状態と考えると, 図15.2 の回路は 図15.4 のようなクロックを入力として各部件の **制御信号** を出力とする Moore 型の同期式順序回路とみなせる.
- ただしここでの状態は, ①~⑧ の9つの状態とレジスタR, M, Fの値, ならびに, 電卓のどのボタンが押されたかを表す組合せ論理回路  $a(\text{input})$  の出力群の組である.

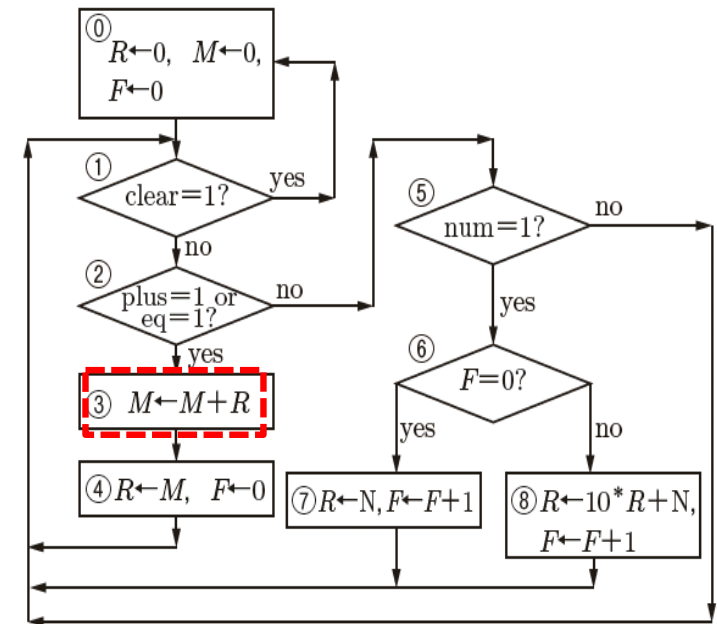


図 15・3 電卓の動作アルゴリズム

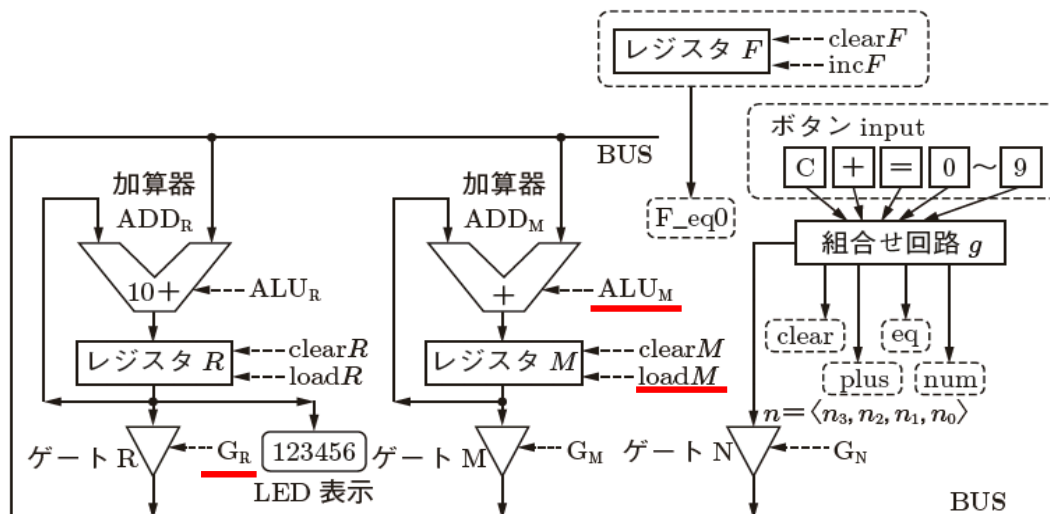


図 15・2 電卓の回路構成

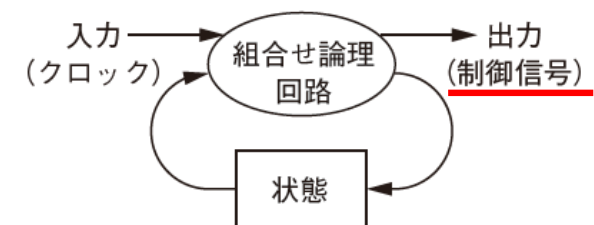


図 15・4 Moore 型の同期式順序回路としての実現



OSAKA UNIVERSITY

# Moore型順序回路としての実現

- ①～⑧の9つの状態を区別するには, 4つのフリップフロップ  $\langle d_3, d_2, d_1, d_0 \rangle$  を用いればよい.
- 各クロック入力に対して, 次の状態  $\langle d_3^+, d_2^+, d_1^+, d_0^+ \rangle$  の値は 11 章で説明したMoore型同期式順序回路の状態の実装法を用いて実現できる.
- レジスタ  $R, M, F$  の値は,  $R, M, F$  の各制御信号 ( $\text{clear}R, \text{load}R, \text{ALU}_R, \text{clear}M, \text{load}M, \text{ALU}_M, \text{clear}F, \text{inc}F, G_R, G_M, G_N$ ) の値を 0 または 1 に設定することで更新できる. これらの制御信号の値が図15・4のMoore型同期式順序回路の出力となる.

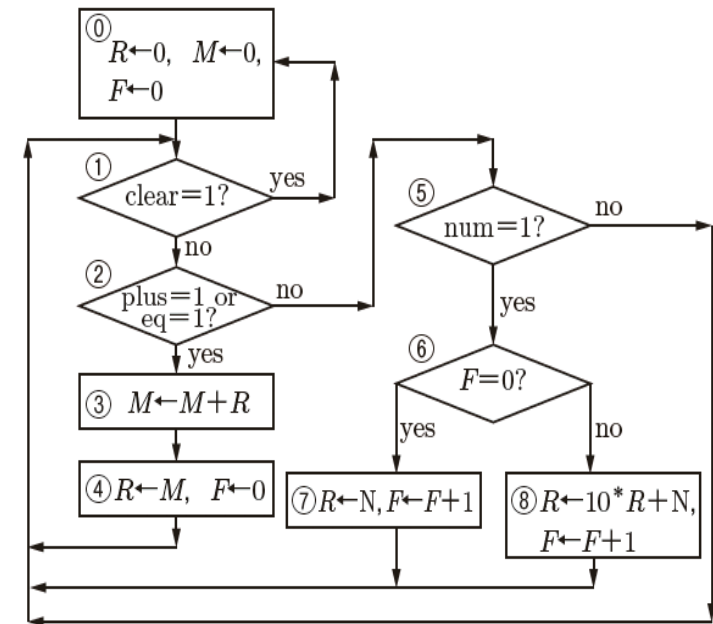


図 15・3 電卓の動作アルゴリズム

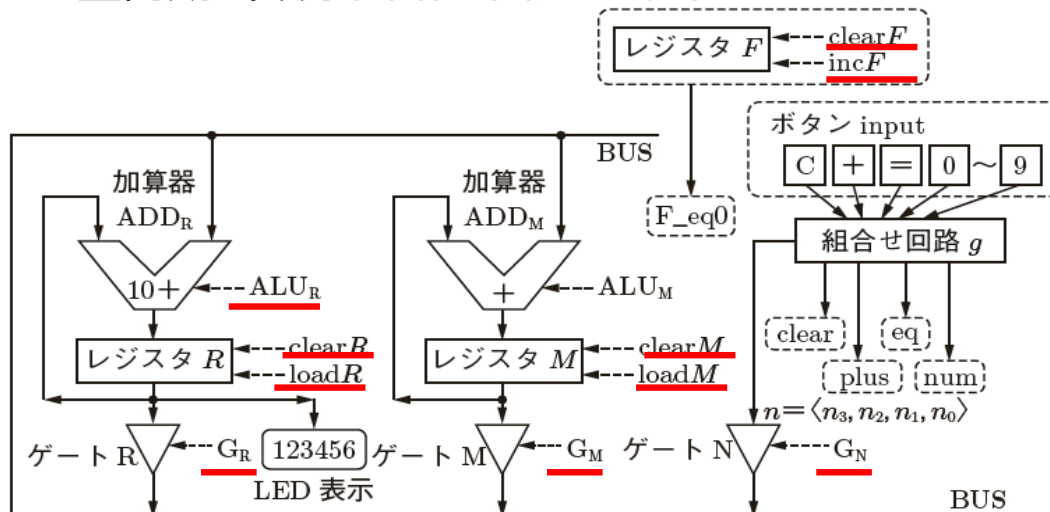


図 15・2 電卓の回路構成

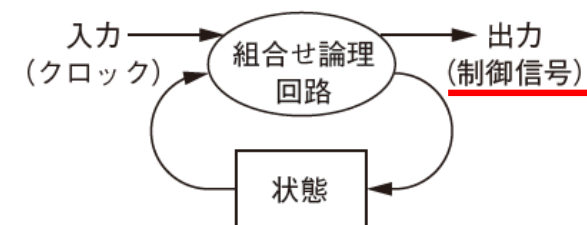


図 15・4 Moore 型の同期式順序回路として実現





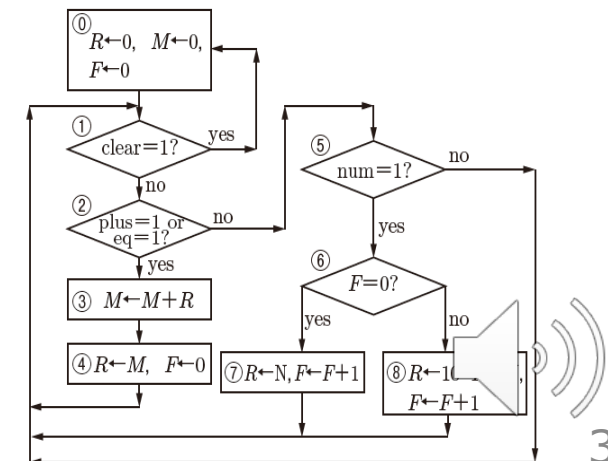
OSAKA UNIVERSITY

## 例題15・3

- 4つのフリップフロップの次の状態  $\hat{d}^+ = (d_3^+, d_2^+, d_1^+, d_0^+)$  の値を各フリップフロップの現在の値  $\hat{d} = (d_3, d_2, d_1, d_0)$  およびレジスタ F の値, 組合せ論理回路  $g(\text{input})$  の出力群  $\text{clear}, \text{plus}, \text{eq}, \text{num}$  を引数とする論理関数で表せ.

(解答)

- 図 15.3 のフローチャートの状態  $i$  が代入文で, 状態  $j$  に遷移する場合,  $\hat{d}^+ = \text{if } (\hat{d} = i) \text{ then } j$  となる. 同様に, 状態  $i$  が条件文で, 条件文  $C$  が真の場合に状態  $j$  に遷移し, 偽の場合に状態  $k$  に遷移する場合  $\hat{d}^+ = \text{if } (\hat{d} = i \wedge C) \text{ then } j \text{ else if } (\hat{d} = i \wedge \neg(C)) \text{ then } k$  となる.
- 図15.3 のフローチャートの 9 つの状態をすべて考慮すると, 4つのフリップフロップの次の状態  $\hat{d}^+ = (d_3^+, d_2^+, d_1^+, d_0^+)$  の値は次のようになる.
- $\hat{d}^+ = \text{if } (\hat{d} = \textcircled{0}) \text{ then } \textcircled{1}$   
 else if  $(\hat{d} = \textcircled{1}) \wedge (\text{clear} = 1)$  then  $\textcircled{0}$   
 else if  $(\hat{d} = \textcircled{1}) \wedge \neg(\text{clear} = 1)$  then  $\textcircled{2}$   
 else if  $(\hat{d} = \textcircled{2}) \wedge (\text{plus} = 1 \vee \text{eq} = 1)$  then  $\textcircled{3}$   
 else if  $(\hat{d} = \textcircled{2}) \wedge \neg(\text{plus} = 1 \vee \text{eq} = 1)$  then  $\textcircled{5}$   
 .....



# 例題15・3

(解答)

- 4つのフリップフロップの次の状態  $\hat{d}^+ = (d_3^+, d_2^+, d_1^+, d_0^+)$  の値は

$\hat{d}^+ = \text{if } (\hat{d} = \textcircled{0}) \text{ then } \textcircled{1}$

else if  $(\hat{d} = \textcircled{1}) \wedge (\text{clear} = 1)$  then  $\textcircled{0}$

else if  $(\hat{d} = \textcircled{1}) \wedge \neg(\text{clear} = 1)$  then  $\textcircled{2}$

else if  $(\hat{d} = \textcircled{2}) \wedge (\text{plus} = 1 \vee \text{eq} = 1)$  then  $\textcircled{3}$

else if  $(\hat{d} = \textcircled{2}) \wedge \neg(\text{plus} = 1 \vee \text{eq} = 1)$  then  $\textcircled{5}$

else if  $(\hat{d} = \textcircled{3})$  then  $\textcircled{4}$

else if  $(\hat{d} = \textcircled{4})$  then  $\textcircled{1}$

else if  $(\hat{d} = \textcircled{5}) \wedge (\text{num} = 1)$  then  $\textcircled{6}$

else if  $(\hat{d} = \textcircled{5}) \wedge \neg(\text{num} = 1)$  then  $\textcircled{1}$

else if  $(\hat{d} = \textcircled{6}) \wedge (F = 0)$  then  $\textcircled{7}$

else if  $(\hat{d} = \textcircled{6}) \wedge \neg(F = 0)$  then  $\textcircled{8}$

else if  $(\hat{d} = \textcircled{7})$  then  $\textcircled{1}$

else  $\textcircled{1}$  (\*  $\hat{d} = \textcircled{8}$ に相当\*)

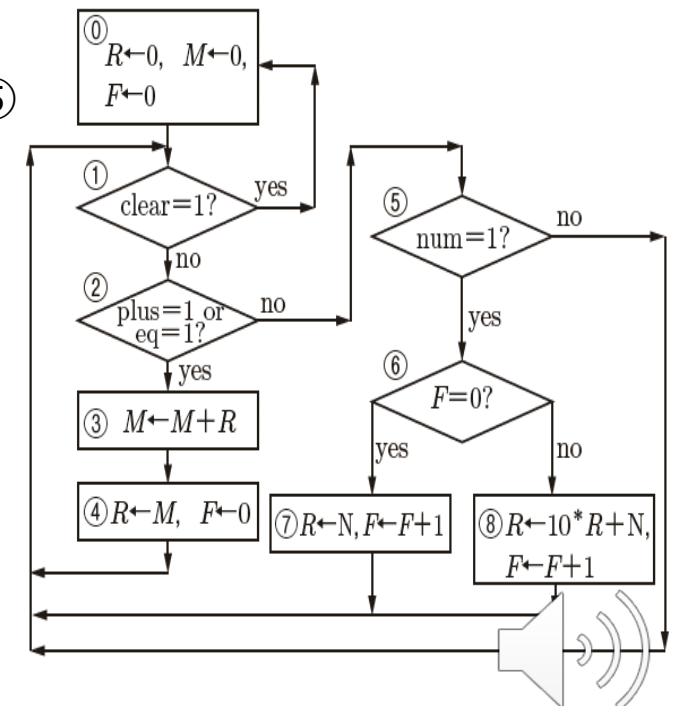


図 15・3 電卓の動作アルゴリズム



# 例題15・3

(解答)

- $\hat{d}^+ = (d_3^+, d_2^+, d_1^+, d_0^+)$  より
- $d_3^+ = (\hat{d}^+ = \textcircled{8})$   
 $= (\hat{d} = \textcircled{6}) \wedge \neg(F=0)$   
 $= ((\neg(d_3) \wedge d_2 \wedge d_1 \wedge \neg(d_0)) \wedge \neg(F=0))$
- $d_2^+ = (\hat{d}^+ = \textcircled{4}) \vee (\hat{d}^+ = \textcircled{5}) \vee (\hat{d}^+ = \textcircled{6}) \vee (\hat{d}^+ = \textcircled{7})$   
 $= \dots\dots\dots$   
 $= ((\neg(d_3) \wedge \neg(d_2) \wedge d_1 \wedge \neg(d_0))$   
 $\quad \wedge \neg(\text{plus}=1 \vee \text{eq}=1)$   
 $\vee ((\neg(d_3) \wedge \neg(d_2) \wedge d_1 \wedge d_0)$   
 $\vee ((\neg(d_3) \wedge d_2 \wedge \neg(d_1) \wedge d_0) \wedge (\text{num}=1)$   
 $\vee ((\neg(d_3) \wedge d_2 \wedge d_1 \wedge \neg(d_0)) \wedge (F=0))$
- $d_1^+, d_0^+$  の論理式も同様に求めることができる
- $d_3^+, d_2^+, d_1^+, d_0^+$  の論理式をそれぞれの D フリップフロップの入力として与えることで, 与えられた状態遷移を実現できる

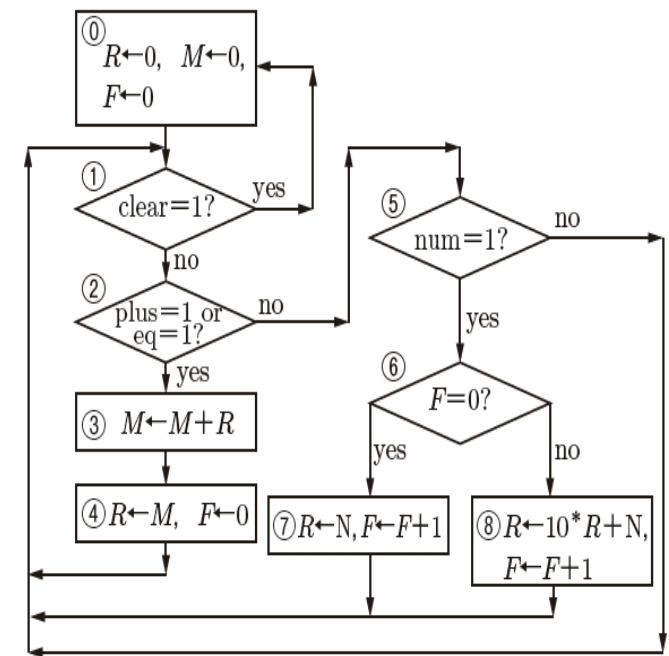


図 15・3 電卓の動作アルゴリズム



# 例題15・3

## 各部品制御信号の決定

- 各部品の制御信号 (clearR, loadR, ALU<sub>R</sub>, clearM, loadM, ALU<sub>M</sub>, clearF, incF, G<sub>R</sub>, G<sub>M</sub>, G<sub>N</sub>) の値をどのように求めるか?
  - 各フリップフロップの現在の値  $\hat{d} = (d_3, d_2, d_1, d_0)$
  - clearR を 1 にする必要があるのはレジスタR の値を 0 にする処理 (図15.3 中の  $R \leftarrow 0$ )
  - $\text{clearR} = (\neg(d_3) \wedge \neg(d_2) \wedge \neg(d_1) \wedge \neg(d_0))$
  - レジスタM に代入操作が行われるのは, フローチャートの③の  $M \leftarrow M + R$
  - $\text{loadM} = (\neg(d_3) \wedge \neg(d_2) \wedge d_1 \wedge d_0)$
  - ALU<sub>R</sub>=1 となるのは  $R \leftarrow 10 * R + N$  が実行される⑧に相当
  - $\text{ALU}_R = (d_3 \wedge \neg(d_2) \wedge \neg(d_1) \wedge \neg(d_0))$

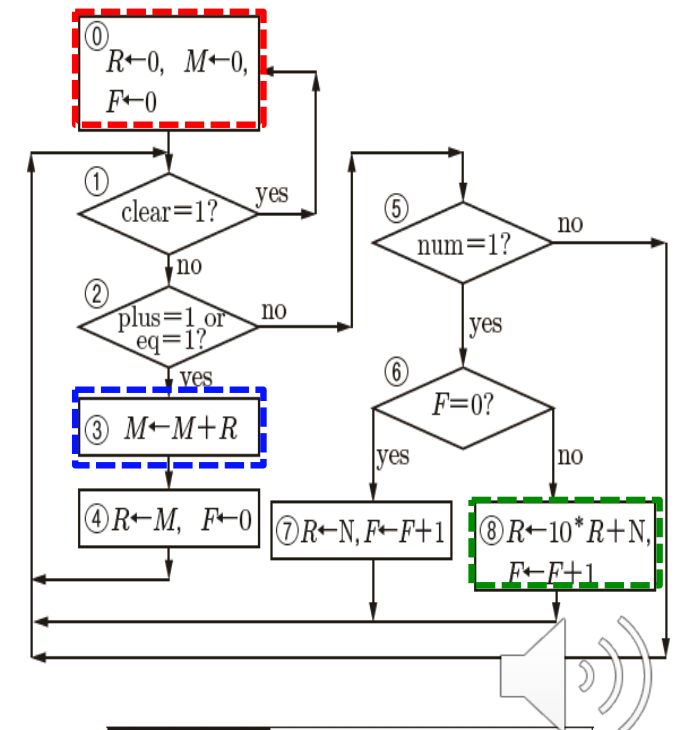


図 15・3 電卓の動作アルゴリズム



OSAKA UNIVERSITY

# 例題15・3 各部品制御信号の決定

- 各部品の制御信号 (clearR, loadR, ALU<sub>R</sub>, clearM, loadM, ALU<sub>M</sub>, clearF, incF, G<sub>R</sub>, G<sub>M</sub>, G<sub>N</sub>) の値をどのように求めるか?

- clearF, incF もそれぞれフローチャートの(⑥, ⑦)と(⑦, ⑧) に相当するので
- $\text{clearF} = (\neg(d_3) \wedge \neg(d_2) \wedge \neg(d_1) \wedge \neg(d_0)) \vee (\neg(d_3) \wedge d_2 \wedge \neg(d_1) \wedge \neg(d_0))$
- $\text{incF} = (\neg(d_3) \wedge d_2 \wedge d_1 \wedge d_0) \vee (d_3 \wedge \neg(d_2) \wedge \neg(d_1) \wedge \neg(d_0))$

- 3つのゲート (ゲートR, ゲートM, ゲートN) を 1 にするのは, フローチャートの代入文の右辺に R, M, N の値が使われる場合に限るので, それぞれ次のように表すことができる

- $G_R = (\neg(d_3) \wedge \neg(d_2) \wedge d_1 \wedge d_0) (*③*)$
- $G_M = (\neg(d_3) \wedge \neg(d_2) \wedge d_1 \wedge d_0) \vee (\neg(d_3) \wedge d_2 \wedge \neg(d_1) \wedge \neg(d_0)) (*③, ④*)$
- $G_N = (\neg(d_3) \wedge d_2 \wedge d_1 \wedge d_0) \vee (d_3 \wedge \neg(d_2) \wedge \neg(d_1) \wedge \neg(d_0)) (*⑦, ⑧*)$

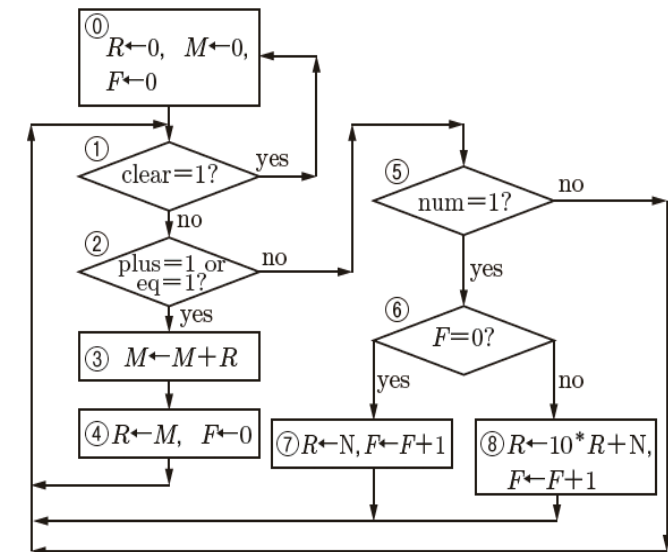


図 15・3 電卓の動作アルゴリズム

- 各部品の制御信号に上記の論理式を与えることで, フローチャートの①~⑧の9つの状態でそれぞれの処理に必要なゲートがオープンされ, ALU の演算内容に従った値がレジスタR, M に代入され, レジスタF の値が変化することになる



# マイクロプログラム方式 による実現

- マイクロプログラム方式では，順序回路の①～⑧の9つの状態を区別するため，図15・5に示すような状態レジスタ (state register) ，あるいはマイクロプログラムカウンタ ( $\mu PC$ ) と呼ばれるレジスタを用いる．
- マイクロプログラムカウンタはMoore型順序回路として実現する場合と同様に，9つの状態を区別するため，4ビットのレジスタを用い， $\mu PC = (d_3, d_2, d_1, d_0)$  と表記する．すなわち， $\mu PC$  の値が  $k$  なら，状態は  $(k)$  である．
- 図15・5に示すように，この  $\mu PC$  の値はマイクロプログラムと呼ばれる制御信号などのデータが格納されている ROM (以下  $\mu P\text{-ROM}$  と表記する) のアドレスを指定するために用いられる．

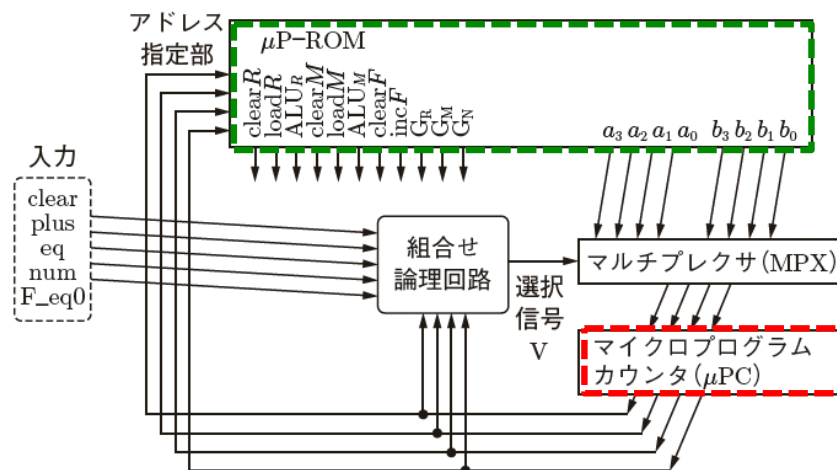


図 15・5 マイクロプログラム方式の制御部の回路構成

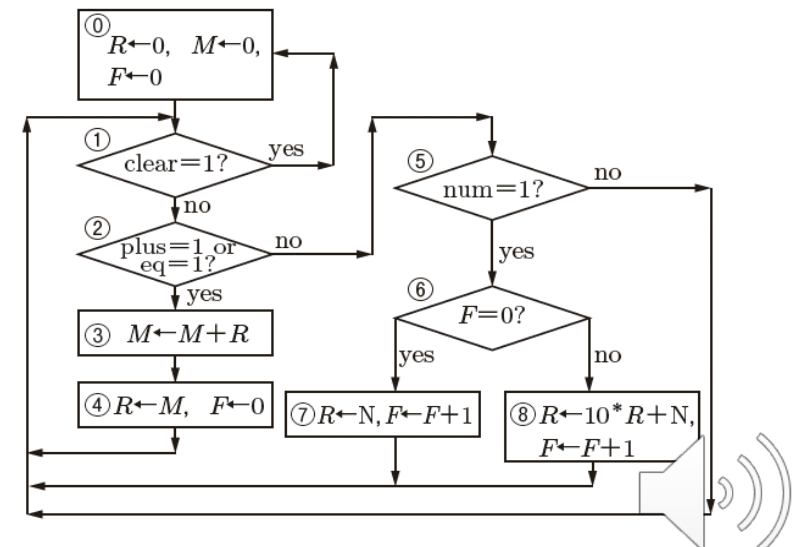
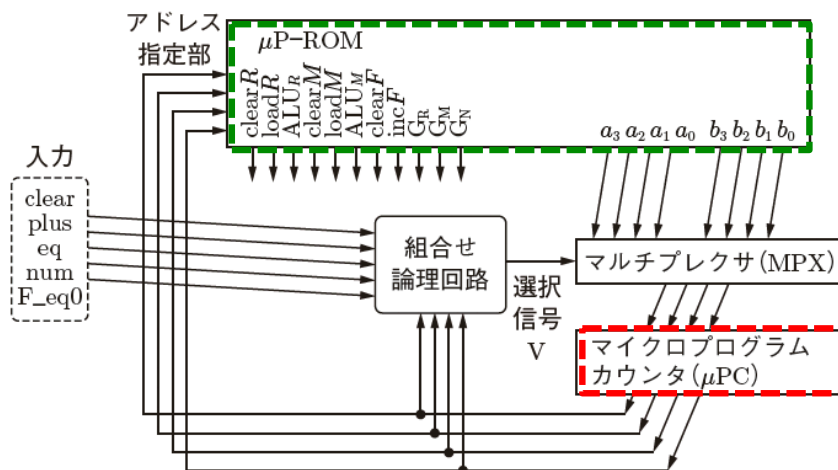


図 15・3 電卓の動作アルゴリズム



# マイクロプログラム方式 による実現

- 図15・5 に示すように、この  $\mu$ PC の値はマイクロプログラムと呼ばれる制御信号などのデータが格納されている ROM（以下  $\mu$ P-ROM と表記する）のアドレスを指定するために用いられる。
  - $\mu$ P-ROMの各番地  $k$  には、状態  $k$  の処理を実現するためのレジスタ  $R, M, F$ , 加算器  $ADD_R, ADD_M$ , ゲート  $G_R, G_M, G_N$  などの部品に対する制御信号の値と、この状態の処理が終わった際の次の遷移先の状態が書かれており、これらをマイクロ命令と呼び、状態①～⑧のマイクロ命令全体をマイクロプログラムと呼ぶ。
  - 電卓回路の場合、これらのマイクロ命令は図15・5 中の  $\mu$ P-ROM の 0 番地から 8 番地に格納される。マイクロ命令の内容は 図15・6 のような 19 ビットで表される。



ビット番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
マイクロ命令	clearR	loadR	ALU <sub>R</sub>	clearM	loadM	ALU <sub>M</sub>	clearF	incF	G <sub>R</sub>	G <sub>M</sub>	G <sub>N</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
番地 0 (①)	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	X	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
番地 2 (③)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1
番地 3 (④)	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	X	X	X	X
番地 4 (⑤)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	X	X	X	X
番地 5 (⑥)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
番地 6 (⑦)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
番地 7 (⑧)	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X
番地 8 (⑨)	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X

図 15・5 マイクロプログラム方式の制御部の回路構成

図 15・6 マイクロプログラムの内容



OSAKA UNIVERSITY

# マイクロプログラム方式 による実現

- 電卓の動作アルゴリズムのフローチャートの①ではレジスタR, M, F の値を 0 にしているので, 図15.6 の 0 番地ではビット番号 1, 4, 7 の **clearR, clearM, clearF** 信号を 1 にし, 次の遷移先の状態として 1 番地の 2 進数表現 (0,0,0,1) を **ビット12~15** の ( $a_3, a_2, a_1, a_0$ ) に記載している。
- 図15.6 の3 番地 ではレジスタ M と R の和をレジスタ M に代入するため, **ビット番号 5, 6, 9** の **loadM, ALU<sub>M</sub>, G<sub>R</sub>** 信号を 1 にし, 次の遷移先の状態として4 番地の 2 進数表現 (0,1,0,0) を**ビット12~15** の ( $a_3, a_2, a_1, a_0$ ) に記載している。

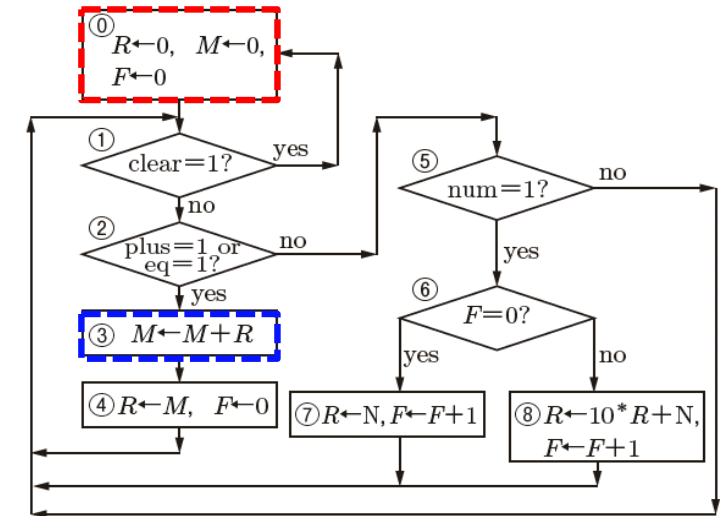


図 15・3 電卓の動作アルゴリズム

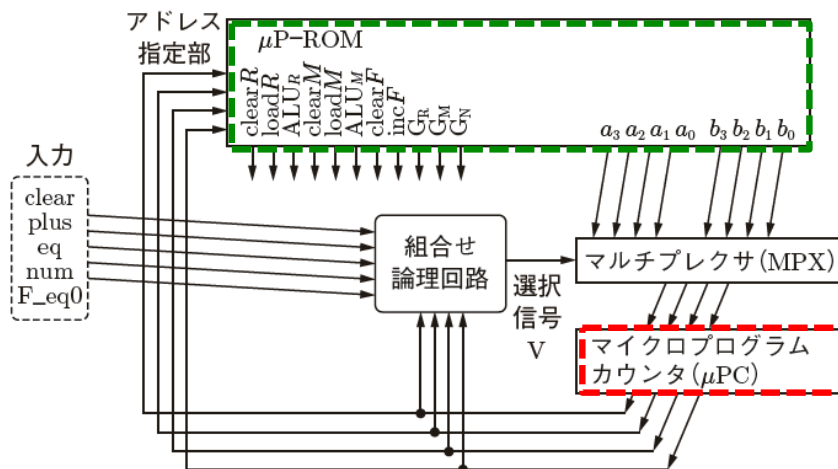


図 15・5 マイクロプログラム方式の制御部の回路構成

ビット番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
マイクロ命令	clearR	loadR	ALU <sub>R</sub>	clearM	loadM	ALU <sub>M</sub>	clearF	incF	G <sub>R</sub>	G <sub>M</sub>	G <sub>N</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
番地 0 (①)	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	X	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
番地 2 (③)	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	X	X	X	X
番地 3 (④)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	X	X	X	X
番地 4 (⑤)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
番地 5 (⑥)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
番地 6 (⑦)	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X
番地 7 (⑧)	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X

図 15・6 マイクロプログラムの内容





# マイクロプログラム方式 による実現（条件分岐）

- 図15.3 では①, ②, ⑤, ⑥の4つの分岐がある. これら4つの状態で条件が成立するときのみビット16~19の  $(b_3, b_2, b_1, b_0)$  が次の遷移先の状態に選ばれ, 条件が成立しないときは, ビット12~15の  $(a_3, a_2, a_1, a_0)$  が次の遷移先の状態に選ばれるように選択信号  $V$  の値を定めている. このため, 選択信号  $V$  の値は次のような論理関数で指定される. ただし  $\mu PC = (d_3, d_2, d_1, d_0)$ .

$$\begin{aligned}
 - V &= ((d_3, d_2, d_1, d_0) = (0, 0, 0, 1) \wedge (\text{clear} = 1)) \\
 &\vee ((d_3, d_2, d_1, d_0) = (0, 0, 1, 0) \wedge (\text{plus} = 1 \vee \text{eq} = 1)) \\
 &\vee ((d_3, d_2, d_1, d_0) = (0, 1, 0, 1) \wedge (\text{num} = 1)) \\
 &\vee ((d_3, d_2, d_1, d_0) = (0, 1, 1, 0) \wedge (F = 0))
 \end{aligned}$$

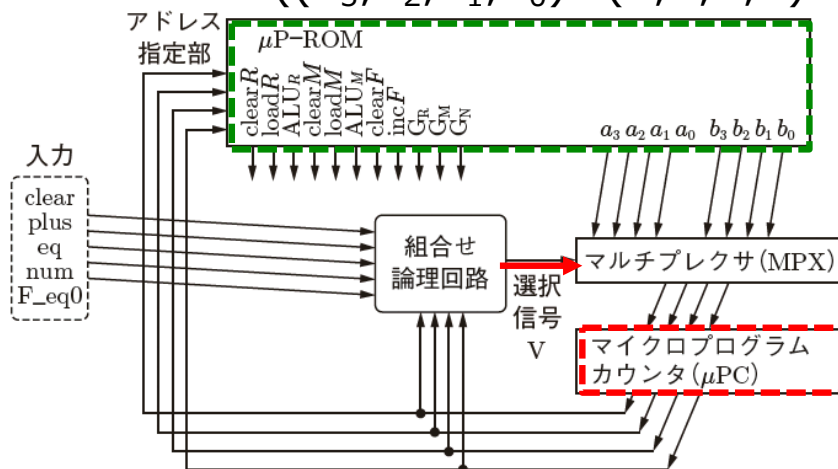


図 15.5 マイクロプログラム方式の制御部の回路構成

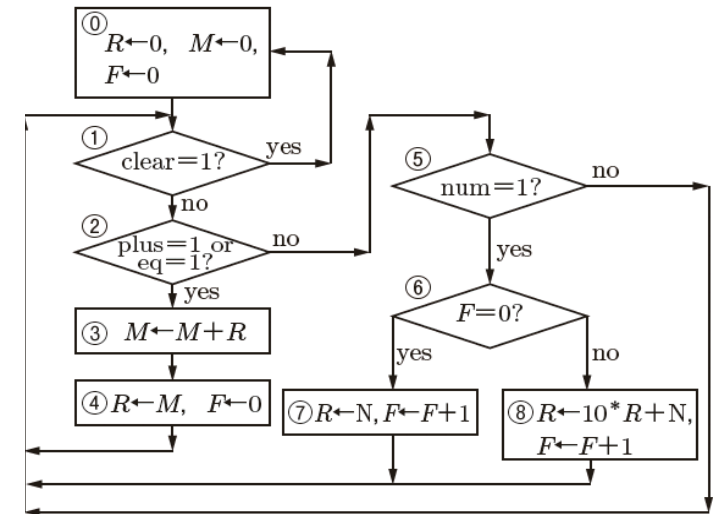


図 15.3 電卓の動作アルゴリズム

ビット番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
マイクロ命令	clearR	loadR	ALUR	clearM	loadM	ALUM	clearF	incF	GR	GM	GN	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
番地 0 (①)	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	X	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
番地 2 (③)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
番地 3 (④)	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	X	X	X	X
番地 4 (⑤)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	X	X	X	X
番地 5 (⑥)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
番地 6 (⑦)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
番地 7 (⑧)	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X
番地 8 (⑨)	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X

図 15.6 マイクロプログラムの内容





# マイクロプログラム方式 による実現（条件分岐）

## 選択信号 V の値

$$\begin{aligned}
 & - V = ((d_3, d_2, d_1, d_0) = (0, 0, 0, 1) \wedge (\text{clear} = 1)) \\
 & \vee ((d_3, d_2, d_1, d_0) = (0, 0, 1, 0) \wedge (\text{plus} = 1 \vee \text{eq} = 1)) \\
 & \vee ((d_3, d_2, d_1, d_0) = (0, 1, 0, 1) \wedge (\text{num} = 1)) \\
 & \vee ((d_3, d_2, d_1, d_0) = (0, 1, 1, 0) \wedge (F = 0)) \\
 & = ((\bar{d}_3 \cdot \bar{d}_2 \cdot \bar{d}_1 \cdot d_0) \wedge \text{clear}) \\
 & \vee ((\bar{d}_3 \cdot \bar{d}_2 \cdot d_1 \cdot \bar{d}_0) \wedge (\text{plus} \vee \text{eq})) \\
 & \vee ((\bar{d}_3 \cdot d_2 \cdot \bar{d}_1 \cdot d_0) \wedge \text{num}) \\
 & \vee ((\bar{d}_3 \cdot d_2 \cdot d_1 \cdot \bar{d}_0) \wedge F_{\text{eq}0})
 \end{aligned}$$

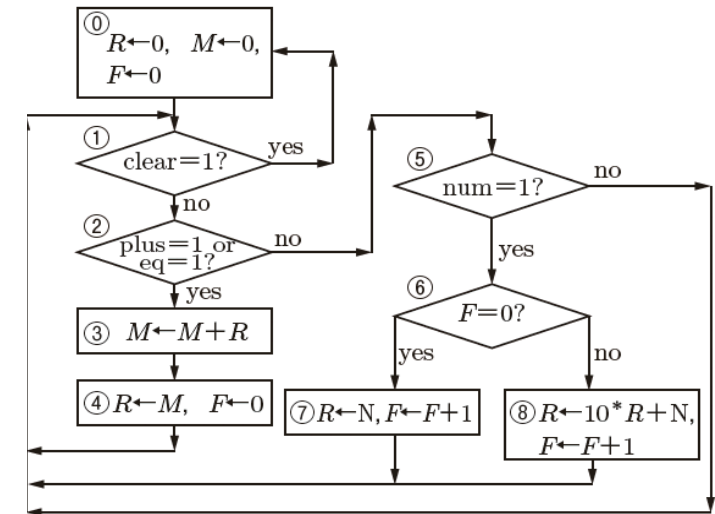


図 15・3 電卓の動作アルゴリズム

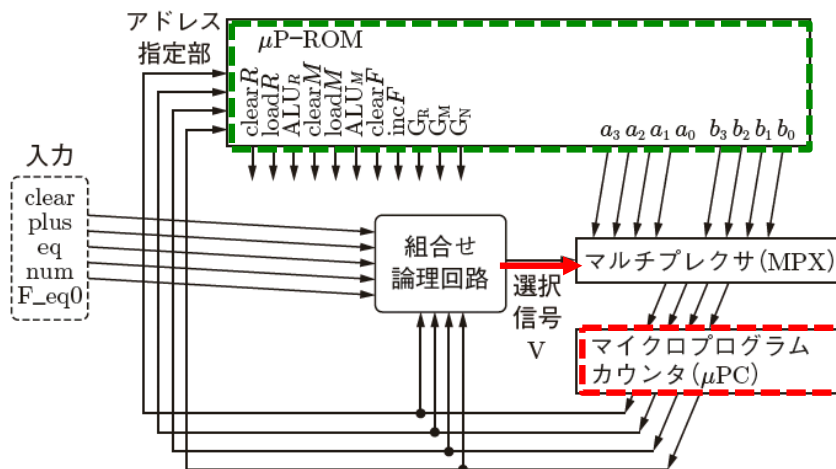


図 15・5 マイクロプログラム方式の制御部の回路構成

ビット番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
マイクロ命令	clearR	loadR	ALU <sub>R</sub>	clearM	loadM	ALU <sub>M</sub>	clearF	incF	G <sub>R</sub>	G <sub>M</sub>	G <sub>N</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
番地 0(①)	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	X	X	X	X
番地 1(②)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
番地 2(③)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
番地 3(④)	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	X	X	X	X
番地 4(⑤)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	X	X	X	X
番地 5(⑥)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
番地 6(⑦)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
番地 7(⑧)	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X
番地 8(⑨)	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X

図 15・6 マイクロプログラムの内容



# マイクロプログラム方式 による実現（条件分岐）

OSAKA UNIVERSITY

## 選択信号 V の値

- $V = ((d_3, d_2, d_1, d_0) = (0, 0, 0, 1) \wedge (\text{clear} = 1))$
- $\vee ((d_3, d_2, d_1, d_0) = (0, 0, 1, 0) \wedge (\text{plus} = 1 \vee \text{eq} = 1))$
- $\vee ((d_3, d_2, d_1, d_0) = (0, 1, 0, 1) \wedge (\text{num} = 1))$
- $\vee ((d_3, d_2, d_1, d_0) = (0, 1, 1, 0) \wedge (F = 0))$
- $= ((\bar{d}_3 \cdot \bar{d}_2 \cdot \bar{d}_1 \cdot d_0) \wedge \text{clear})$
- $\vee ((\bar{d}_3 \cdot \bar{d}_2 \cdot d_1 \cdot \bar{d}_0) \wedge (\text{plus} \vee \text{eq}))$
- $\vee ((\bar{d}_3 \cdot d_2 \cdot \bar{d}_1 \cdot d_0) \wedge \text{num})$
- $\vee ((\bar{d}_3 \cdot d_2 \cdot d_1 \cdot \bar{d}_0) \wedge F_{\text{eq}0})$

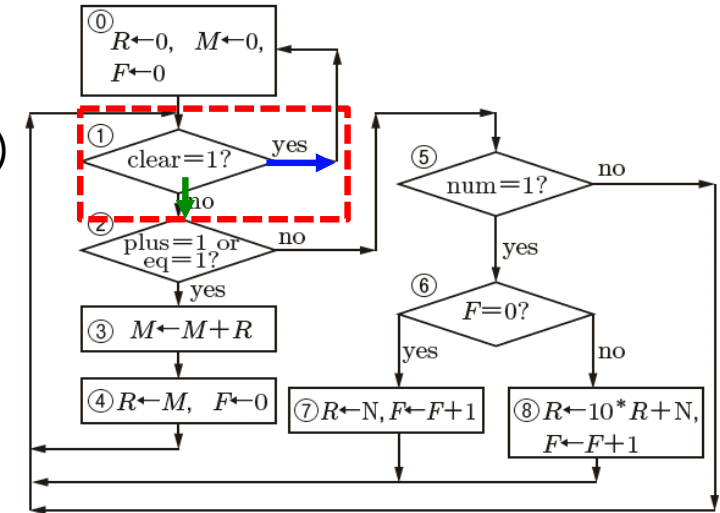


図 15・3 電卓の動作アルゴリズム

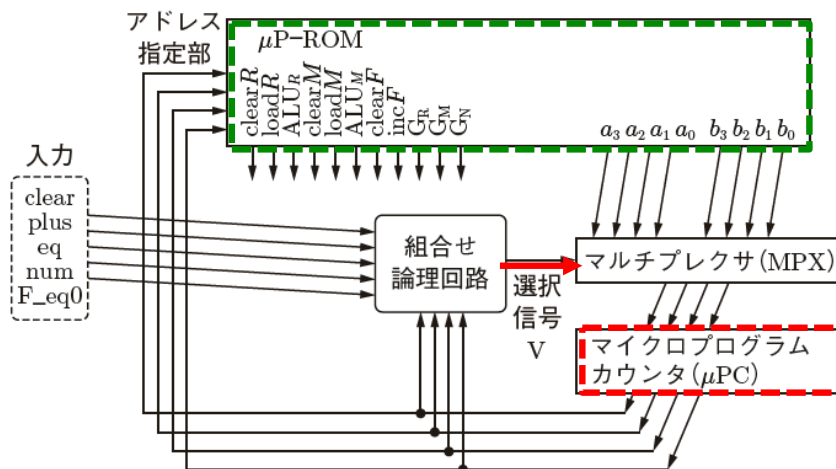


図 15・5 マイクロプログラム方式の制御部の回路構成

ビット番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
マイクロ命令	clearR	loadR	ALU <sub>R</sub>	clearM	loadM	ALU <sub>M</sub>	clearF	incF	G <sub>R</sub>	G <sub>M</sub>	G <sub>N</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
番地 0 (①)	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	X	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
番地 2 (③)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1
番地 3 (④)	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	X	X	X	X
番地 4 (⑤)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	X	X	X	X
番地 5 (⑥)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
番地 6 (⑦)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
番地 7 (⑧)	0	1	0	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X	X
番地 8 (⑨)	0	1	1	0	0	0	1	0	0	1	0	0	0	1	X	X	X	X	X

図 15・6 マイクロプログラムの内容



# 簡単な自動販売機の 制御部の設計

- 図15.7 のような回路を用いて自動販売機の制御を行う回路を同期式順序回路として実現
  - 外部からお金が投入されるとその金額が自動的にレジスタR に格納され、InputR 信号が ON になる。レジスタR の値は ClearR 信号をONにすると 0 にリセットされ、新たなお金が投入されるまで InputR信号 も OFF になる（InputR 信号が ON の間は次のお金が投入できないようになって いるものとする）。投入されたお金は順次 レジスタS に加算していくものとし、商品の料金（120 円）以上お金が投入されると比較器の出力 Comp が ON になる。Comp が ON になると、おつりを計算して釣り銭額をレジスタ Q に格納し、商品出力を指示するために レジスタP の値を ON にセットする。簡単のため、商品が出力されると自動的に レジスタQ の値が 0 に、レジスタP の値が OFF にリセットされるものとする。自動販売機はこれらの操作を繰り返し行うものとする。

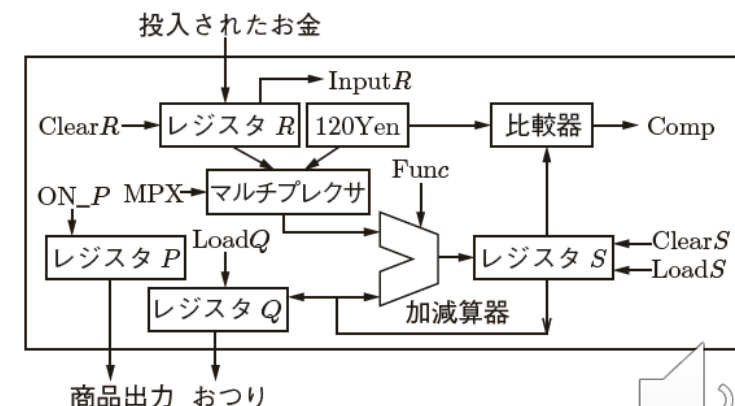


図 15・7 自動販売機の制御を行う同期式順序回路

# 例題15・4

## 自動販売機の制御部の 動作アルゴリズムのフローチャート

- 図15.7 の各部品は 図15.8 のように動作するものと仮定して，この自動販売機の制御部の動作アルゴリズムのフローチャートを作成せよ。

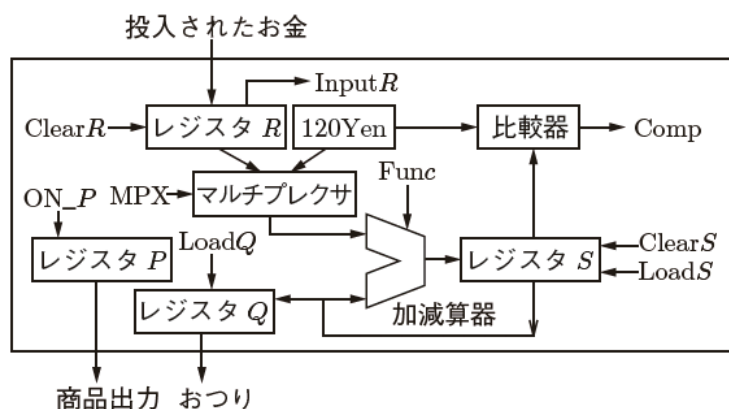


図 15・7 自動販売機の制御を行う同期式順序回路

- (イ) Clear $R$ =1 のとき  $R$  の値を 0 にし，Input $R$  信号を OFF にする ( $R \leftarrow 0$  と書く)  
Clear $R$ =0 のとき お金が投入されない限り  $R$  の値は不変，お金が投入されると投入されたお金が  $R$  にセットされ，Input $R$  信号が ON となる (Input $R$  信号が ON かどうかのチェックは (Input $R$ =ON) ? と書く)。
- (ロ) MPX=1 のとき レジスタ  $R$  の値がマルチプレクサから出力される。  
MPX=0 のとき 商品価格 (120 円) がマルチプレクサから出力される。
- (ハ) Func=1, Load $S$ =1, Clear $S$ =0 のとき  $S+R$  または  $S+120$  の値を  $S$  に代入する ( $S \leftarrow S+R$  または  $S \leftarrow S+120$  と書く)  
Func=0, Load $S$ =1, Clear $S$ =0 のとき  $S-R$  または  $S-120$  の値を  $S$  に代入する ( $S \leftarrow S-R$  または  $S \leftarrow S-120$  と書く)
- Load $S$ =0, Clear $S$ =1 のとき  $S$  の値を 0 にする ( $S \leftarrow 0$  と書く)  
Load $S$ =0, Clear $S$ =0 のとき  $S$  の値は不変
- (ニ) ON\_P=1 のとき  $P$  の値を ON にする ( $P \leftarrow \text{ON}$  と書く)  
ON\_P=0 のとき 商品出力後，自動的に OFF になる
- (ホ) Load $Q$ =1 のとき  $S$  の値を  $Q$  に代入する ( $Q \leftarrow S$  と書く)  
Load $Q$ =0 のとき 商品出力後，自動的に 0 になる
- (ヘ)  $S \geq 120$  のときかつそのときのみ Comp 信号が ON になる  
(Comp 信号が ON かどうかのチェックは ( $S \geq 120$ ) ? と書く)。

図 15・8 自動販売機の各部品の動作仕様



# 例題15・4

## 自動販売機の制御部の 動作アルゴリズムのフローチャート

- 図15.7 の各部品は 図15.8 のように動作するものと仮定して、この自動販売機の制御部の動作アルゴリズムのフローチャートを作成せよ。

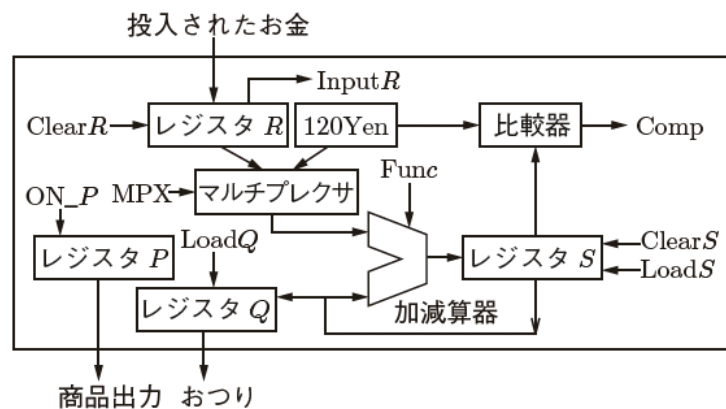


図 15・7 自動販売機の制御を行う同期式順序回路

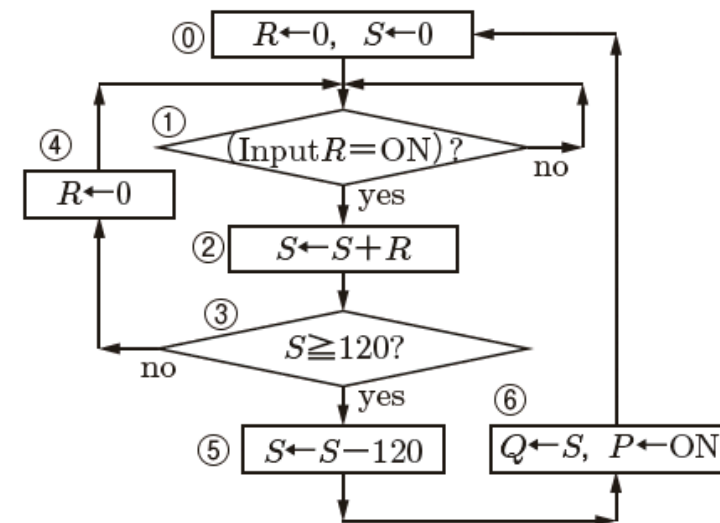


図 15・9 自動販売機の動作アルゴリズムのフローチャート





# 例題15・4

## 自動販売機の制御部を マイクロプログラム方式で実現する方法

各番地のマイクロ命令はフローチャートの各処理や条件分岐（① から⑥に相当）に対応

- 例えば、フローチャートの①の代入文 ( $R \leftarrow 0, S \leftarrow 0$ ) を実行するため、図15・10のマイクロ命令では、ClearR 信号とClearS 信号の値を 1 にセットしている。フローチャートの①の実行後の遷移先は①なので、図15・10のマイクロ命令では、 $\langle a_2, a_1, a_0 \rangle$  に対応する遷移先番地 (0,0,1) が記載されている。
- ここで、 $\langle a_2, a_1, a_0 \rangle, \langle b_2, b_1, b_0 \rangle$  はそれぞれマイクロ命令のアドレスである ( $a_0, b_0$  が最下位ビット)。
- マルチプレクサ MPX は選択信号  $V$  の値が 0 のとき  $\langle a_2, a_1, a_0 \rangle$  を出力し、 $V$  の値が 1 のとき  $\langle b_2, b_1, b_0 \rangle$  を出力する。選択信号  $V$  の値は、図15・10 中の

$$V = (T_{p1} \cdot f_1(\text{InputR}, \text{Comp}))$$

$$\vee (T_{p3} \cdot f_3(\text{InputR}, \text{Comp}))$$

- マイクロ命令に特別な制御信号  $T_{pk}$  ( $k=1,3$ ) を導入し、フローチャートの条件分岐①, ③に相当するマイクロ命令の 1,3 番地の制御信号  $T_{p1}, T_{p3}$  の値を 1 にすることで、その番地で  $f_k(\text{InputR}, \text{Comp})$  ( $k=1,3$ ) が成り立つとき、 $\langle b_2, b_1, b_0 \rangle$  が次の遷移先番地として選択されるように制御する

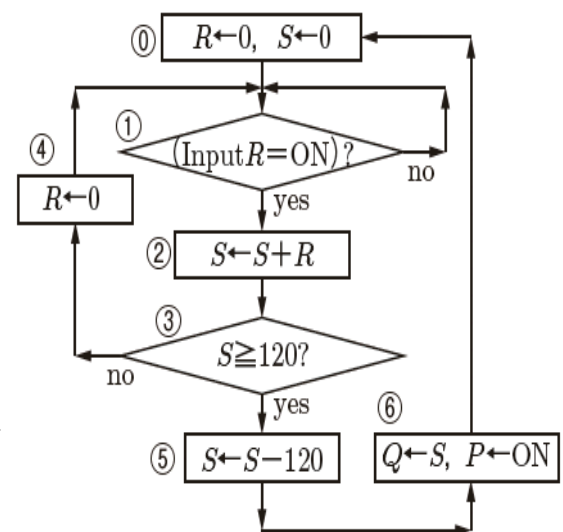


図 15・9

自動販売機の動作アルゴリズムのフローチャート

# 例題15・5

## 自動販売機の制御部を マイクロプログラム方式で実現する方法

1. この回路の選択信号  $V$  の内容 ( $f_1(\text{InputR}, \text{Comp})$ ,  $f_3(\text{InputR}, \text{Comp})$  の具体的内容) ?
2. 図15・10 の  $\mu\text{P-ROM}$  の内容で, 自動販売機の制御部が正しく動作することを確認せよ.
3. この回路を商品の定価が変わっても動作するようにするには回路をどう変更すべきか?

(解答)

1.  $V = (T_{p1} \cdot \text{InputR}) \vee (T_{p3} \cdot \text{Comp})$
2. (省略)
3. 図15・7 の120 円のレジスタを変更されたあとの商品の定価に置き換えればよい.

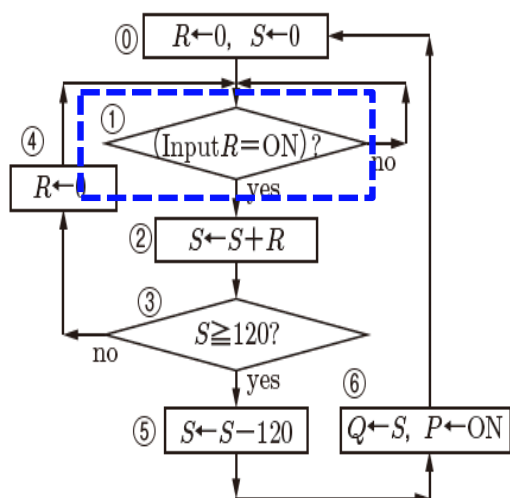


図 15・9 自動販売機の動作アルゴリズムのフローチャート

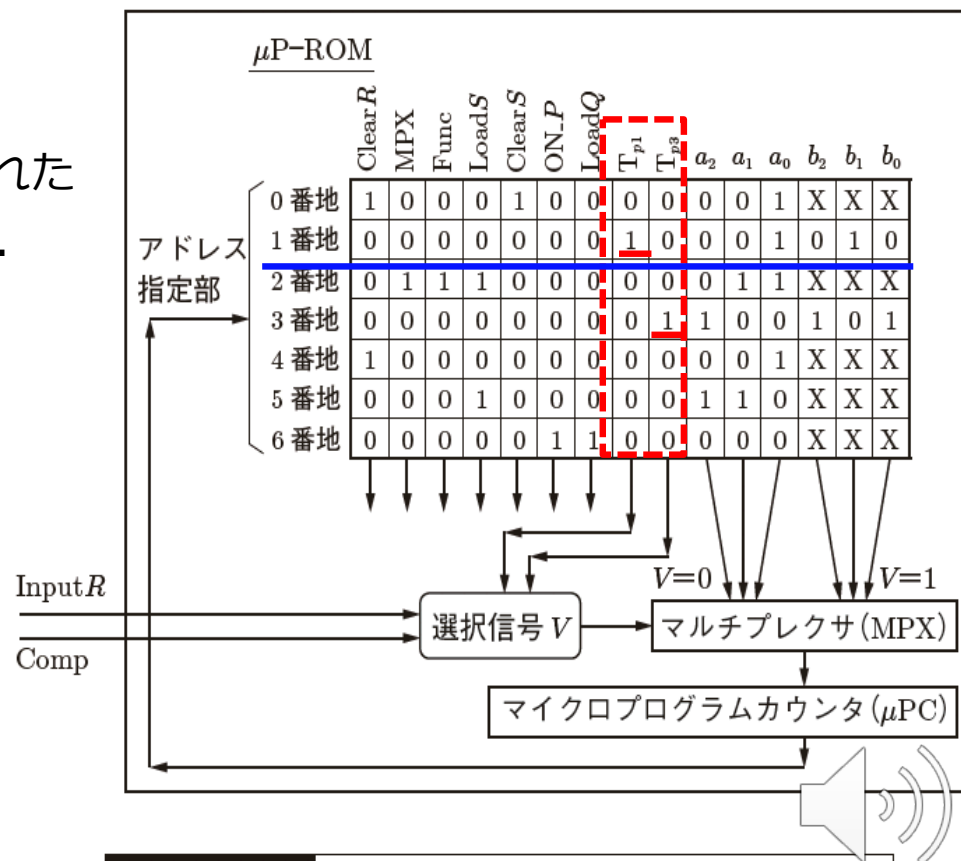


図 15・10 マイクロ命令のアドレスを決定する回路



# 例題15・5

## 自動販売機の制御部を マイクロプログラム方式で実現する方法

1. この回路の選択信号  $V$  の内容 ( $f_1(\text{InputR,Comp})$ ,  $f_3(\text{InputR,Comp})$  の具体的内容) ?
2. 図15・10 の  $\mu\text{P-ROM}$  の内容で, 自動販売機の制御部が正しく動作することを確認せよ.
3. この回路を商品の定価が変わっても動作するようにするには回路をどう変更すべきか?

(解答)

1.  $V = (T_{p1} \cdot \text{InputR}) \vee (T_{p3} \cdot \text{Comp})$
2. (省略)
3. 図15・7 の120 円のレジスタを変更されたあとの商品の定価に置き換えればよい.

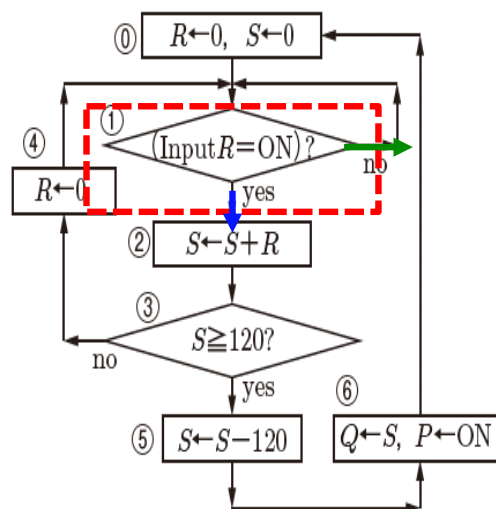


図 15・9 自動販売機の動作アルゴリズムのフローチャート

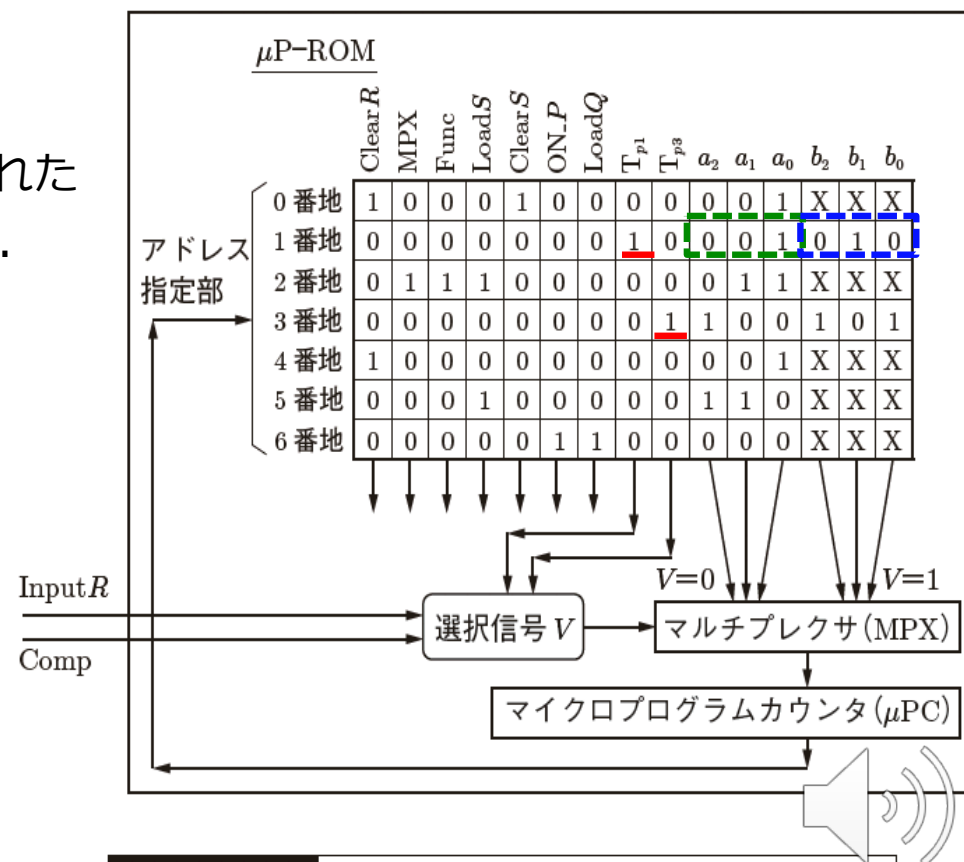


図 15・10 マイクロ命令のアドレスを決定する回路

# 例題15・5

## 自動販売機の制御部を マイクロプログラム方式で実現する方法

1. この回路の選択信号  $V$  の内容 ( $f_1(\text{InputR, Comp})$ ,  $f_3(\text{InputR, Comp})$  の具体的内容) ?
2. 図15・10 の  $\mu\text{P-ROM}$  の内容で, 自動販売機の制御部が正しく動作することを確認せよ.
3. この回路を商品の定価が変わっても動作するようにするには回路をどう変更すべきか?

(解答)

1.  $V = (T_{p1} \cdot \text{InputR}) \vee (T_{p3} \cdot \text{Comp})$
2. (省略)
3. 図15・7 の120 円のレジスタを変更されたあとの商品の定価に置き換えればよい.

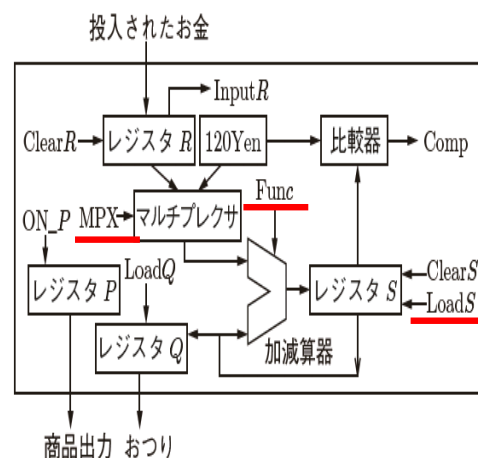
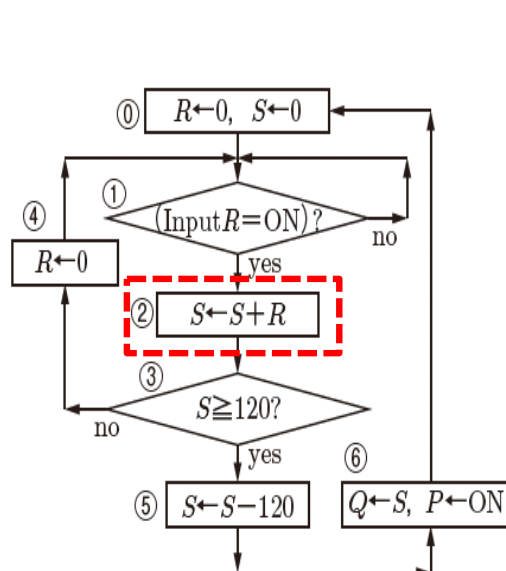


図 15・7 自動販売機の制御を行う同期式順序回路

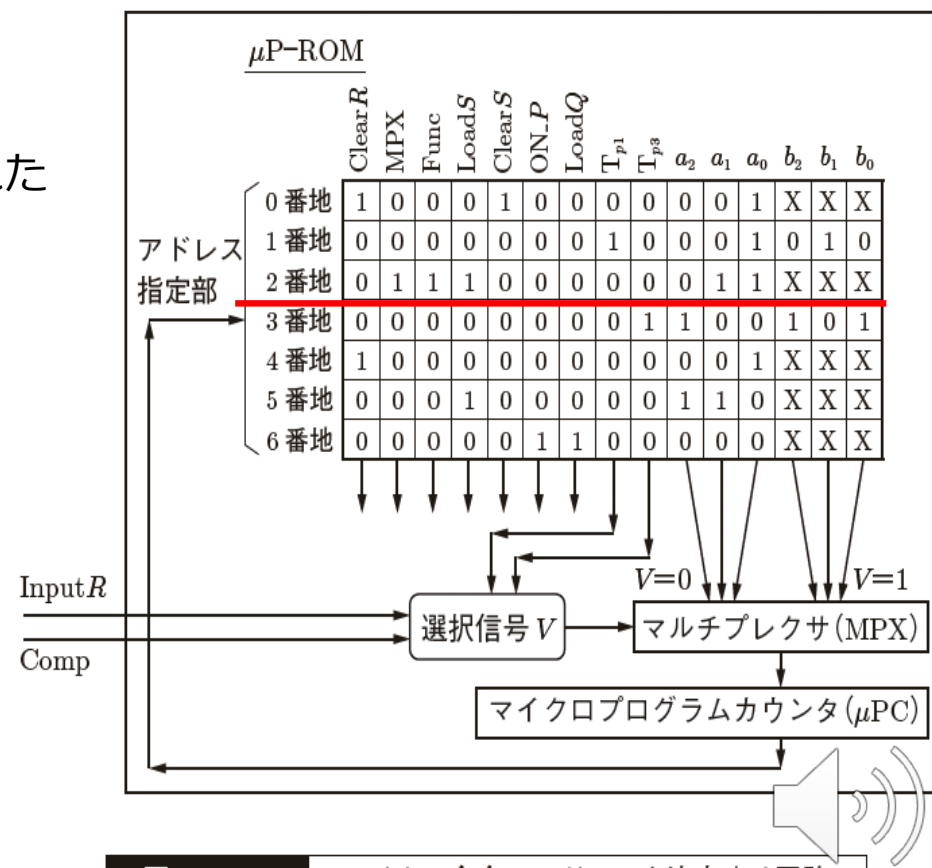


図 15・10 マイクロ命令のアドレスを決定する回路

# 例題15・5

## 自動販売機の制御部を マイクロプログラム方式で実現する方法

1. この回路の選択信号  $V$  の内容 ( $f_1(\text{InputR}, \text{Comp})$ ,  $f_3(\text{InputR}, \text{Comp})$  の具体的内容) ?
2. 図15・10 の  $\mu\text{P-ROM}$  の内容で, 自動販売機の制御部が正しく動作することを確認せよ.
3. この回路を商品の定価が変わっても動作するようにするには回路をどう変更すべきか?

(解答)

1.  $V = (T_{p1} \cdot \text{InputR}) \vee (T_{p3} \cdot \text{Comp})$
2. (省略)
3. 図15・7 の120 円のレジスタを変更されたあとの商品の定価に置き換えればよい.

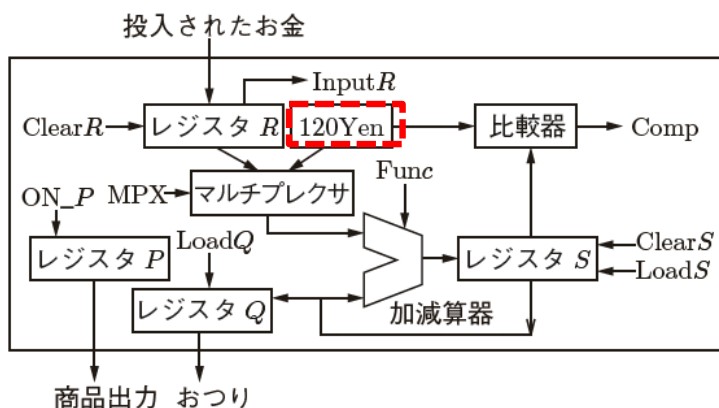


図 15・7 自動販売機の制御を行う同期式順序回路

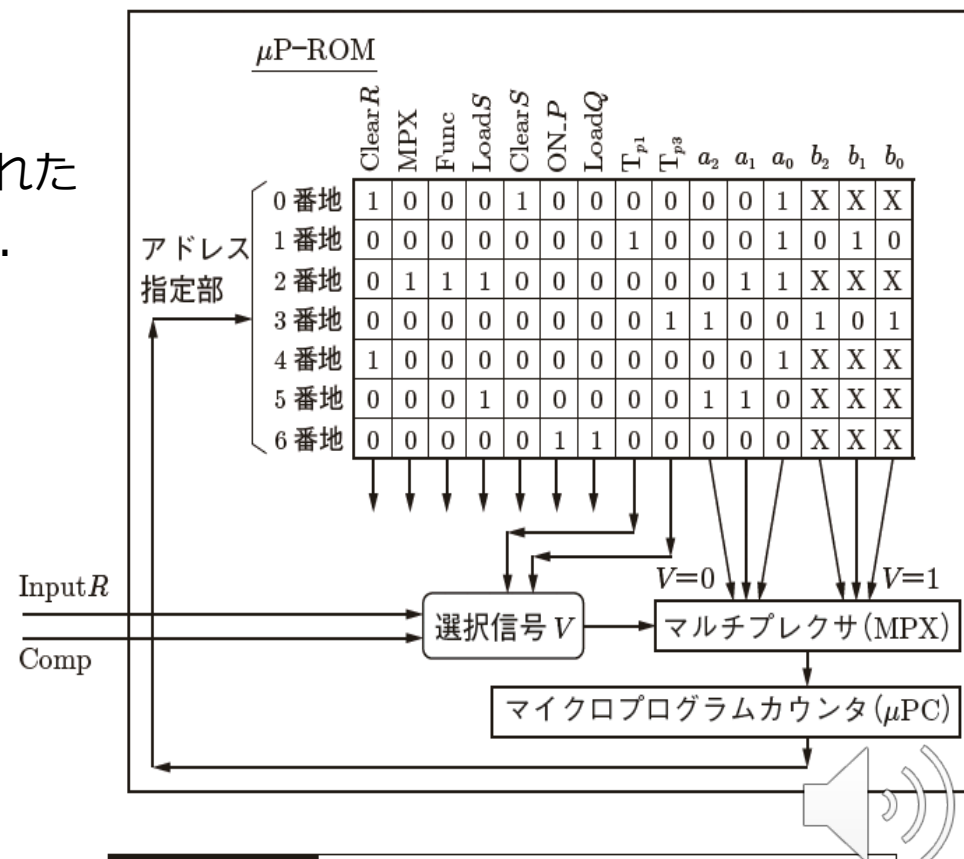


図 15・10 マイクロ命令のアドレスを決定する回路



# 10回目の授業終了





# 授業終了

皆さん

今日はレポート課題はありません

