

教材用基本的計算機の試作

福間 彰^{*}・駅田 省吾^{**}

Akira FUKUMA and Shōgo EKIDA
Trial Manufacture of Fundamental Computer
for Training

1. ま え が き

1971年末に現れた最初のマイクロコンピュータ MC S・4以後約10年、各界に与えたマイクロコンピュータの衝撃は大きく、研究室を含む各職域へのパーソナルコンピュータの進出が著しい昨今である。計算機を利用する立場からいえば、ベーシックのような言語を理解し、ソフトウェアを学習するだけでも十分に広い分野での活用が期待できる。しかしながら、ひとたび自らの手でマイコンシステムの拡充や新しい周辺機器の接続等を考えた場合、インタフェースなどハードに関する基本的知識が必要となってくる。

一方、電子計算機の教科書には、基本的計算機の機能が記述されているが、初心者が計算機の基本を真に理解するには机上の学習のみでなく体験学習が大切であり、理論と実際を結び付ける装置が必要となってくる。

また、マイクロプロセッサから出発したマイコン学習装置は種々市販もされているが、CPUの中身にまで踏込んだより基本的な学習教材は少ないようである。

ここでは後者の立場に立ち、現在、国民一般のものとなってきたコンピュータのしくみを、ハードの面からも理解できるよう、富崎新氏の ATOM-8¹⁾を起点とし、より広い用途に供しうるよう拡張した基本的計算機を設計製作したのでこれについて報告する。

2. 設 計

2.1 概要

基本的電子計算機 MIC-II のデータの流れを図1に示す。単位語長8ビット、直列処理方式を採用し、処理の様子を1ビットずつモニターでき、制御のタイムン

グも設計し易く、ATOM-8に次の機能を新しく加えた。これによりモデル化と汎用性を増すことができた。

- 1) 処理速度の選択……MAIN CLOCK 20Hz または 1kHz。
- 2) 1ステップモード……一命令だけの実行が可能。
- 3) 入出力ポートの新設……キーボードや表示装置との接続を考慮して、入力、出力各4ポート用意。
- 4) インデックスレジスタの新設……8ビット×4個
- 5) 記憶容量の拡大……32語×8ページ
- 6) スタックレジスタの新設……8ビットのスタックレジスタにより PC (プログラムカウンタ) の内容を一時退避でき、1レベルのサブルーチンの処理が可能。
- 7) PC をパネルに用意したデータスイッチによりランダムにアクセスできるため、任意の番地の内容を読み出したり (DEBUG)、書き込んだり (STORE)、任意の番地からプログラムをスタート (RUN) させることができる。

2.2 入出力ポート

入力素子には並列入力、直列出力用 TTL SN 74165 を用い、アキュムレータ (ACC) 用クロック信号 ACC ϕ と同位相のクロックパルスにより入力信号を ACC に送る。出力素子には、直列入力、並列出力用 SN74164 を用い、さらに、シフトの際に不必要なデータが出力されるのを防ぐため、シフトの終りでラッチして出力するよう4ビットのラッチ SN7475 を2個ずつ用意した。また各ポートにはハンドシェイクが可能のようにフラグを1ビットずつ備えてあり、これらフラグの状態は、FR (FLAG REGISTER) を介し RFR 命令 (READ FR) によりモニターできる。ゆえに、入力の有無、あるいは出力データのレシーブの有無は、この FR をモニターすることにより知ることができる。

2.3 インデックスレジスタ

* 島根大学教育学部技術研究室

** 島根県飯石郡赤来町立赤来中学校

カウント操作も命令一つでできる汎用のレジスタで、次の機能を持つ。(1)インクリメント (INC 命令)、(2)デクリメント (DEC 命令)、(3)クリア (CIX 命令)、(4)書き込み (WIX 命令)、(5)読み出し (RIX 命令)、(6) INDEX 修飾によるメモリの内容の読み出し (REI 命令)、と書き込み (WRI 命令)。この場合、読み書きされるメモリの番地は INDEX REGISTER (8ビット) によって示される。

2.4 記憶容量の拡大

オペランドで示される番地は最大5ビットのため、一度にアクセス可能なメモリは $2^5=32$ (0番地~31番地) であり、これではプログラムもかなりの制約を受ける。そこで、単位語長の8ビットに合わせ、さらに3ビットを追加し、1ページ32バイト (0~31番地)、 $8(=2^3)$ ページという概念の下に、メモリの増設を計った。ただし5ビットのオペランドでは、このページはアクセスできないので、実行ページ以外のページをアクセスするときは BR (BASE PAGE REGISTER) にページ数をロードしてやる必要がある (SPL 命令)。

2.5 サブルーチンの1レベルまでの処理

JMP (JUMP) 命令でプログラムの分岐を起こす場合、その次の番地を SR (STACK REGISTER) に退避させ、元のルーチンに戻るときは、RTS 命令 (RETURN SUBROUTINE) で PC (PROGRAM COUNTER) に SR の内容を書き込むことにより1レベルの分岐を可能にしている。

2.6 命令コード

MIC-II のモニクな命令とその機械コードを表1に示す。このうち JMP 命令は、アキュムレータの MSB (D7) が1の時、オペランド5ビットが指定する

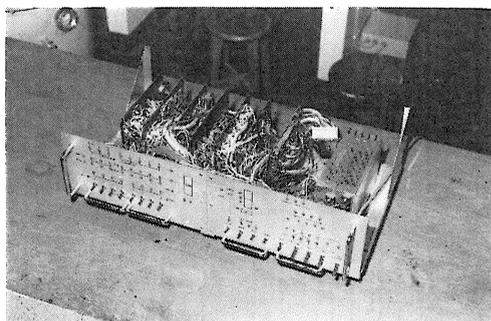


図2 MIC-II

表1

No	MNEMONIC	DESCRIPTION	MACHIN CODE							
			D7	D6	D5	D4	D3	D2	D1	DO
1	STP	STOP MAIN CLOCK	0	0	0	0	0	0	0	0
2	RSH	RIGHT SHIFT ××× TIMES	0	0	1	0	0	×	×	×
3	LSH	LEFT SHIFT ××× TIMES	0	0	1	0	1	×	×	×
4	BPL	BASE PAGE REGISTER LOAD	0	0	1	1	0	×	×	×
5	INP	(INPUT PORT) →ACC	0	0	1	1	1	0	×	×
6	OUT	(ACC) →OUTPUT PORT	0	0	1	1	1	1	×	×
7	SUB	(ACC) - (MEM) →ACC	0	1	0	×	×	×	×	×
8	ADD	(ACC) + (MEM) →ACC	0	1	1	×	×	×	×	×
9	RED	(MEM) →ACC	1	0	0	×	×	×	×	×
10	WRT	(ACC) →MEMORY	1	0	1	×	×	×	×	×
11	JMP	JUMP IF D7(ACC) IS '1'	1	1	0	×	×	×	×	×
12	RTS	(STACK REGISTER) →PC	1	1	1	0	0	0	0	0
13	RFR	(FLAG REGISTER) →ACC	1	1	1	0	0	0	0	1
14	CFR	CLEAR FLAG REGISTER	1	1	1	0	0	0	1	0
15	REI	(MEM IND INX) →ACC	1	1	1	0	0	1	×	×
16	WRI	(ACC) →MEM IND INDEX	1	1	1	0	1	0	×	×
17	CIX	CLEAR INDEX REGISTER	1	1	1	0	1	1	×	×
18	INC	(INX) + 1 → INX	1	1	1	1	0	0	×	×
19	DEC	(INX) - 1 → INX	1	1	1	1	0	1	×	×
20	RIX	(INX) →ACC	1	1	1	1	1	0	×	×
21	WIX	(ACC) → INX	1	1	1	1	0	1	×	×

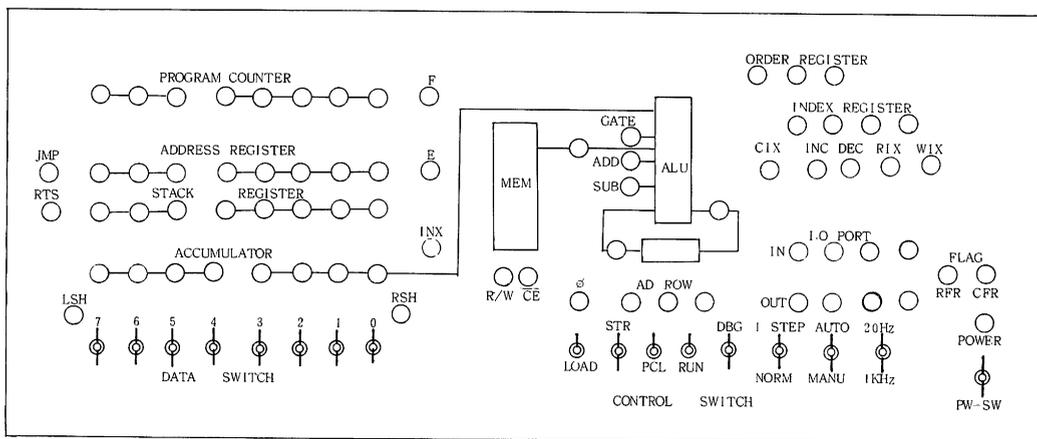


図9 パネル構成

番地へ分岐させる命令である。

3. 製作

3.1 回路

MIC-II は図2および次に示すように、6枚の基板と1枚の補助基板、パネルおよび電源部より構成されている。(1)ACC・ALU・CLOCK(図3), (2)MEMORY(図4), (3)CONTROL-1 (ORDER DECODER) (図5), (4)CONTROL-2 (ORDER DECODER) (図6), (5)INDEX REGISTER (図7), (6)I. O PORT(図8)

3.2 パネル

図9および図2に示すように、垂直パネル上にDATA SWITCH 8個, CONTROL SWITCH 8個(Load, STORE, PC-LOAD, RUN, DEBUG, 1-STEP/NORMAL, AUTO/MANUAL, 20 Hz/1kHz) および電源 SWITCH を設け、発光ダイオードを用いたコード等表示部には、PROGRAM COUNTER, ADDRESS REGISTER, STACK REGISTER, ACCUMULATOR 各8ビット, ORDER REGISTER 3ビットを備え、命令表示としては、JMP, RTS, LSH, RSH, ADD, SUB, CIX, INC, DEC, RIX, WIX, RFR, CFR があり、状態表示にはこの他 F (FETCH), E (EXECUTE), INX, R/W, \overline{CE} , GATE, ϕ , AD ROW, INDEX REGISTER, IN-PORT, OUT-PORT, POWER がある。

LOAD.....MANU で、DATA SWITCH (DS) の状態を ACCUMULATOR (ACC) にロードする。

PCL.....MANU で、DS の状態を PROGRAM COUNTER (PC) にロードする。

STR.....MANU で、ACC の内容を PC の示す番地にストアする。

RUN.....AUTO, NORM/1-STEP の状態で、20 Hz/1kHz のメインクロックで、PC の示す番地からプログラムに従って命令を読み出し実行する。

DBG.....PC が示す番地の内容を ACC に示す。

4. プログラム例

4.1 キー入力

キーから ASCII コードのデータを受取ると、1 バイトごとにデータを組み立て、INDEX REGISTER 2 で示す番地へ順次格納してゆくプログラムのフローチャートとルーチンを夫々図10および表2に示す。ここでは INX0 をキーコードのデータ用に、INX-1 は1バイトの HEX DATA のカウント用に、INX-2 はデータ

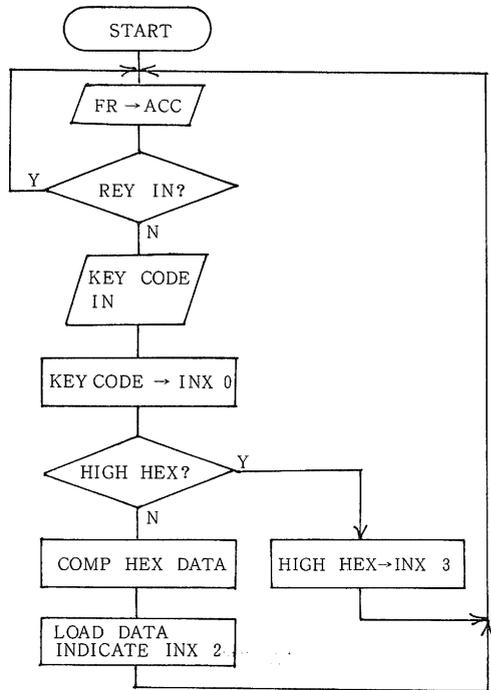


図10 ロード

表2

P	LOC	LABEL	MNI	OPERAND	MACHIN CODE
0	0	SRCH KEY	RFR		1 1 1 0 0 0 0 1
1	1	LSH	LSH (7)		0 0 1 0 1 1 1 1
2	2	DL 0	JMP	KEY IN 0	1 1 0 0 0 1 0 1
3	3		RED	DL 0	1 0 0 0 0 0 1 0
4	4		JMP	SRCH KEY	1 1 0 0 0 0 0 0
5	5	KEY IN 0	BPL (1)		0 0 1 1 0 0 0 1
6	6		RED	DL 1	1 0 0 0 1 1 0 0
7	7		JMP	KEY IN	1 1 0 0 0 0 0 0
1	0	KEY IN	INP (0)		0 0 1 1 1 0 0 0
1	1		WIX (0)		1 1 1 1 1 1 0 0
2	2		RIX (1)		1 1 1 1 1 0 0 1
3	3		SUB	01	0 1 0 1 1 0 0 1
4	4		JMP	LOAD HIGH HEX	1 1 0 1 0 0 1 1
5	5	LOAD LOW HEX	RIX (0)		1 1 1 1 1 0 0 0
6	6		SUB	30	0 1 0 1 1 0 1 0
7	7		WRT	LOW HEX	1 0 1 1 1 0 1 1
8	8		RIX (3)		1 1 1 1 1 0 1 1
9	9		ADD	HIGH HEX	0 1 1 1 1 0 1 1
A	A		WIX (3)		1 1 1 1 1 1 1 1
B	B		CIX (0)		1 1 1 0 1 1 1 0
C	C	DL 1	CIX (1)		1 1 1 0 1 1 0 1
D	D	LOAD HEX	RIX (3)		1 1 1 1 1 0 1 1
E	E		WRI (2)		1 1 1 0 1 0 1 0
F	F		JNC (2)		1 1 1 1 0 0 1 0
10	10	TO SRCH KEY	BPL 0		0 0 1 1 1 0 0 0
11	11		RED	DL 0	1 0 0 0 0 0 1 0
12	12		JMP	SRCH KEY	1 1 0 0 0 0 0 0
13	13	LOAD HIGH HEX	RIX (0)		1 1 1 1 1 0 0 0
14	14		LSH (4)		0 0 1 0 1 1 0 0
15	15		WIX (3)		1 1 1 1 1 1 1 1
16	16		INC (1)		1 1 1 1 0 0 0 1
17	17		RED	DL 1	1 0 0 0 1 1 0 0
18	18		JMP	TO SRCH KEY	1 1 0 1 0 0 0 0
19	19	01			0 0 0 0 0 0 0 1
1A	1A	30			0 0 1 1 0 0 0 0
1B	1B	LOW HEX			0 0 0 0 0 0 0 0

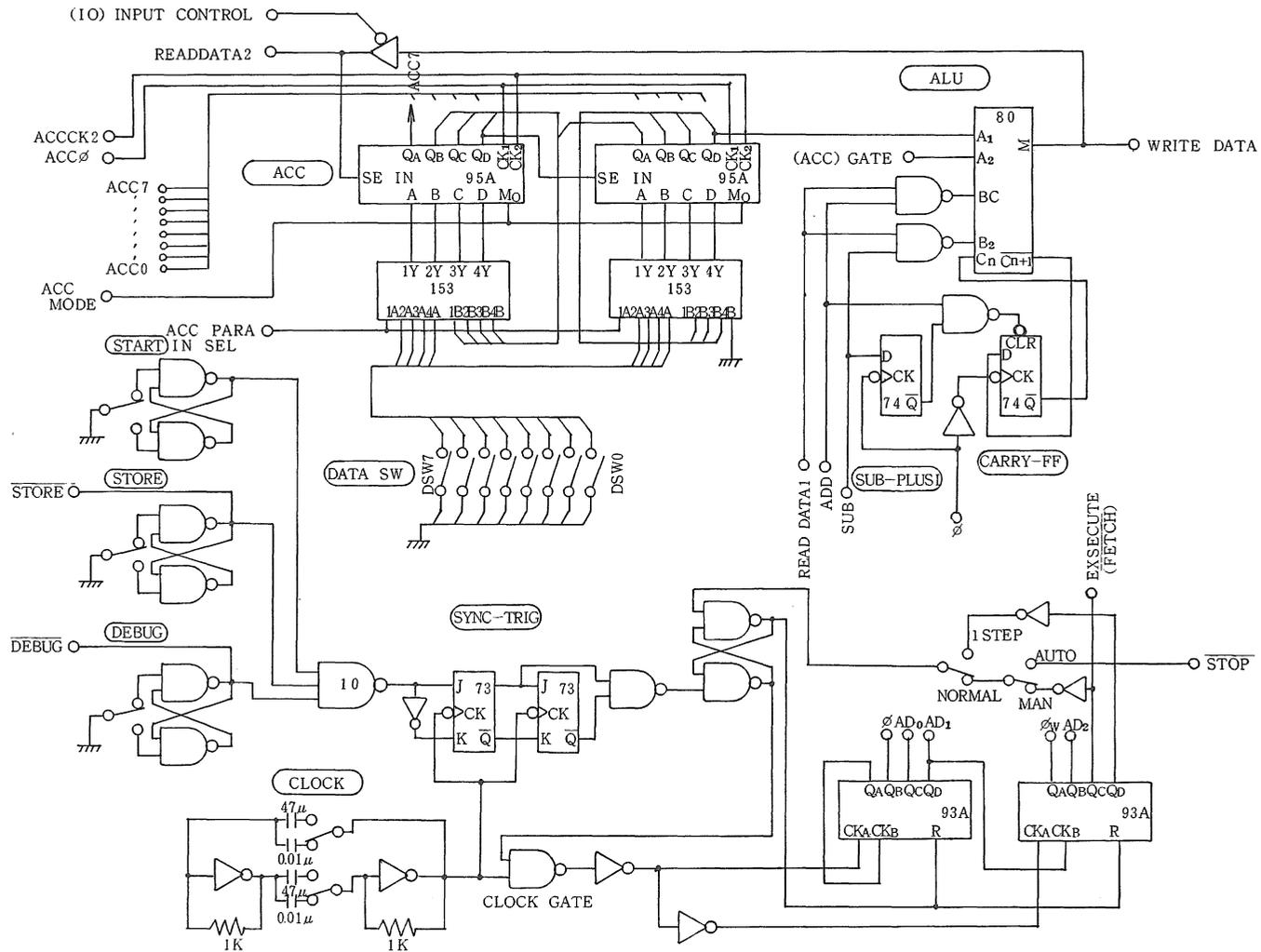


图3 ACC·ALU·CLOCK

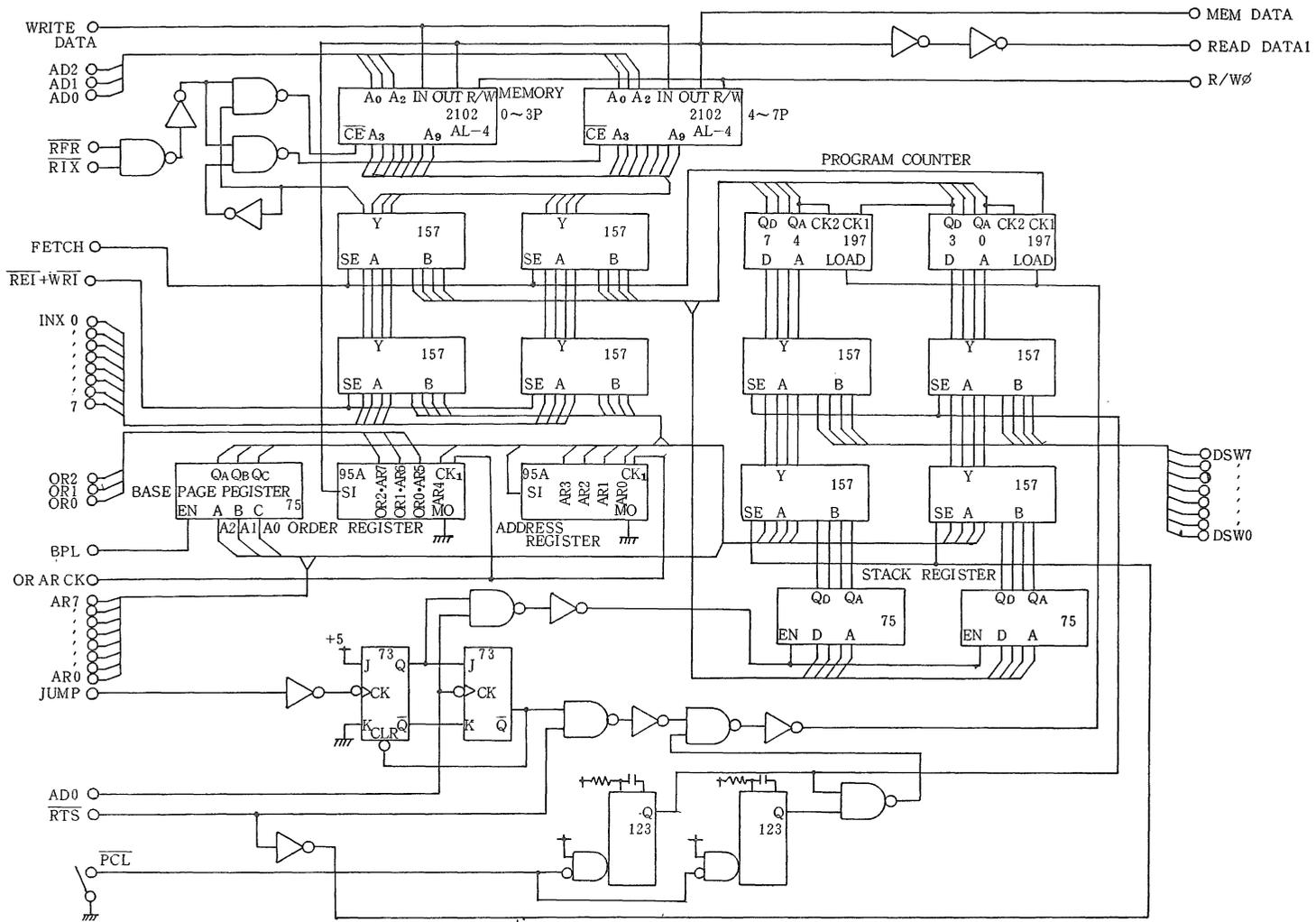


図4 MEMORY

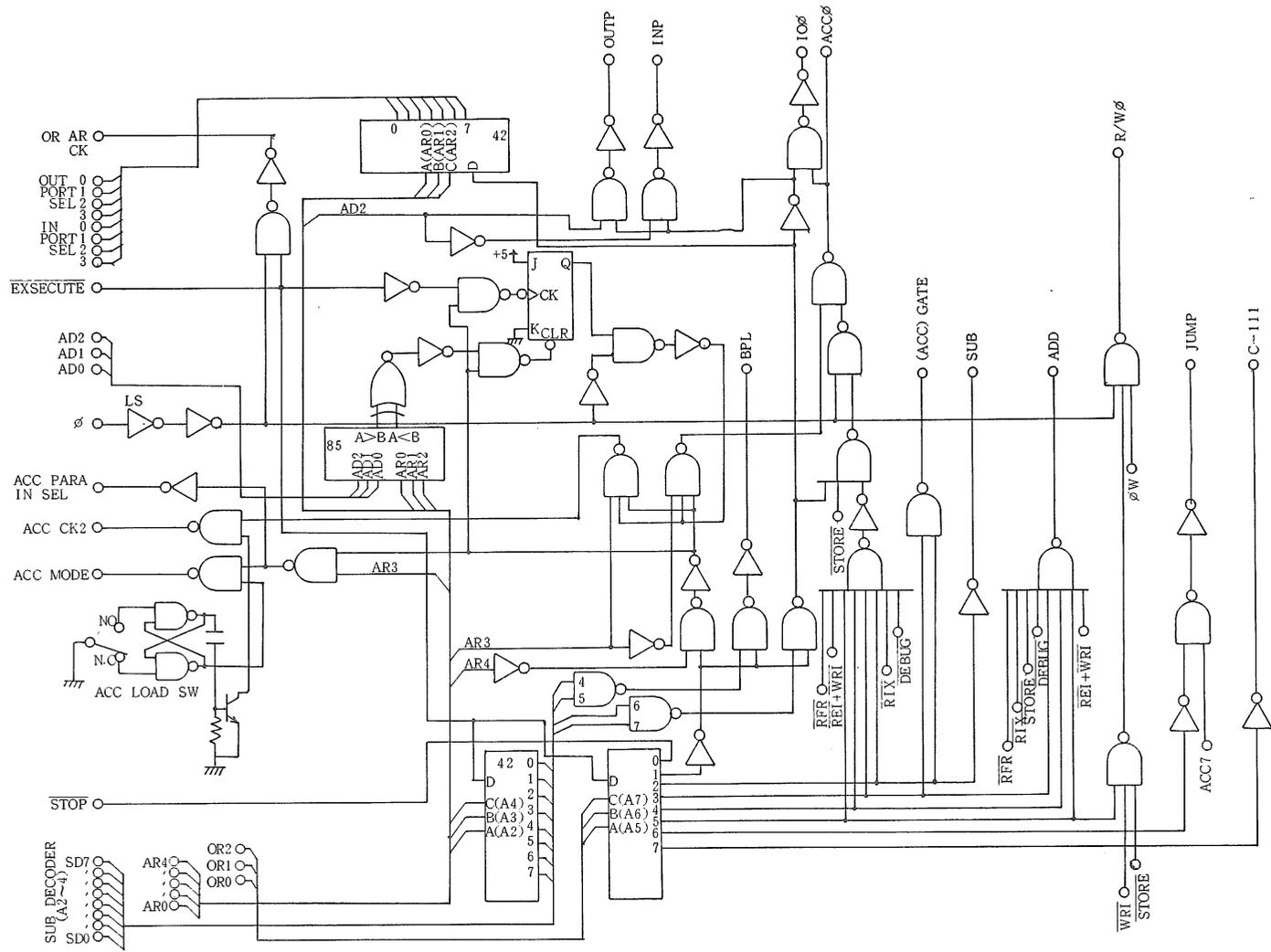


図5 CONTROL 1. (ORDER DECODER)

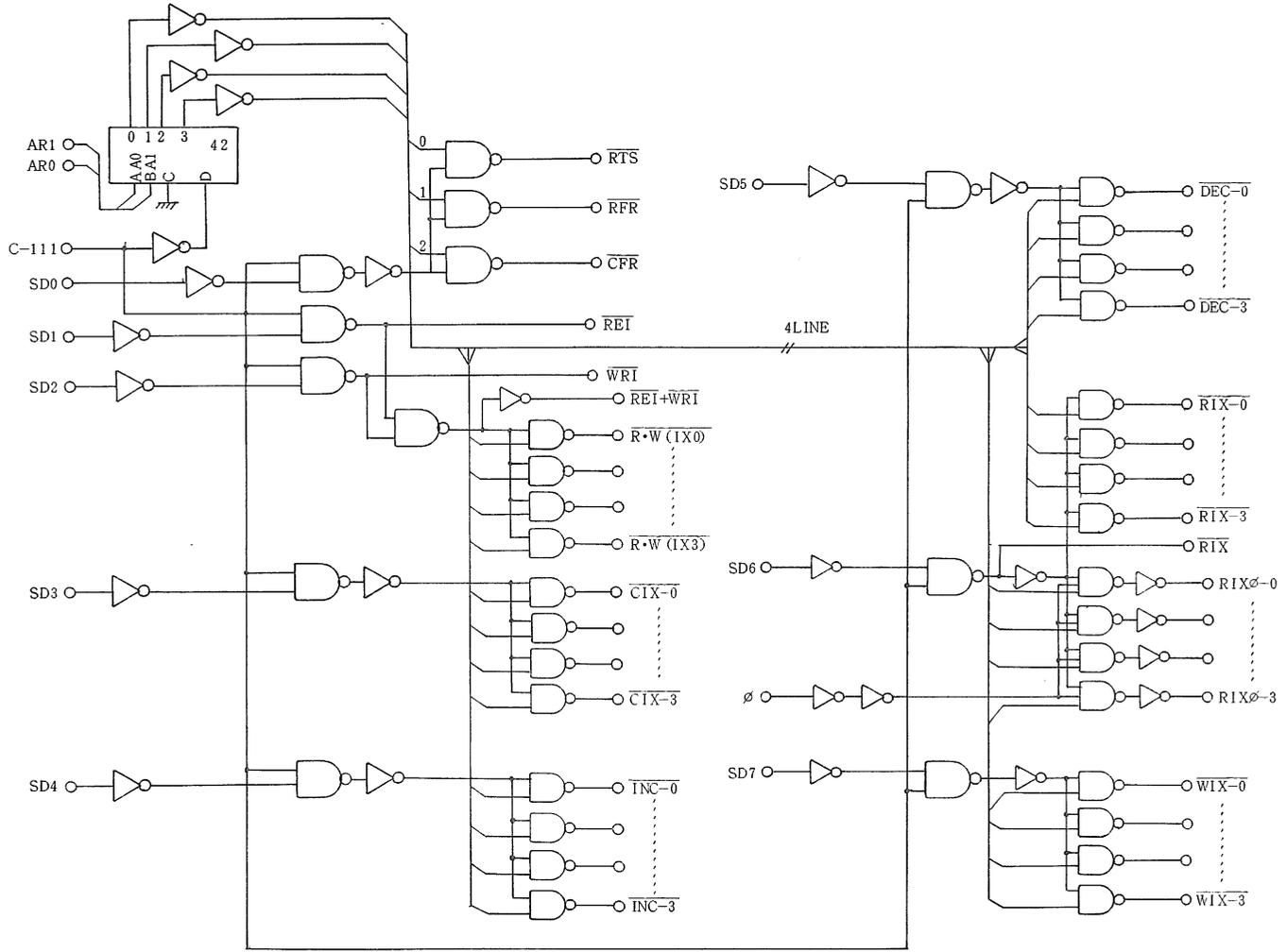


図6 CONTROL 2. (ORDER DECODER)

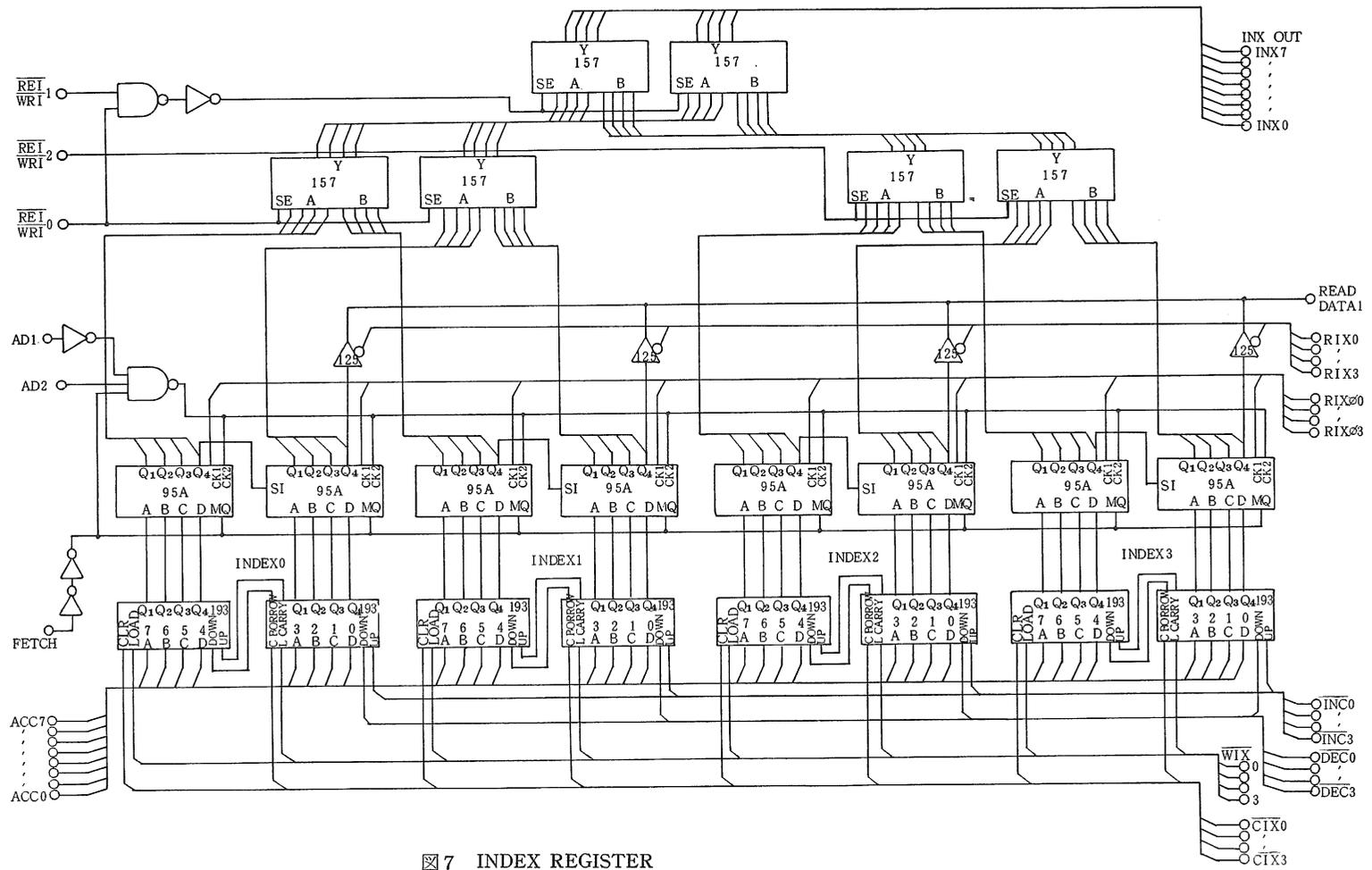


図 7 INDEX REGISTER

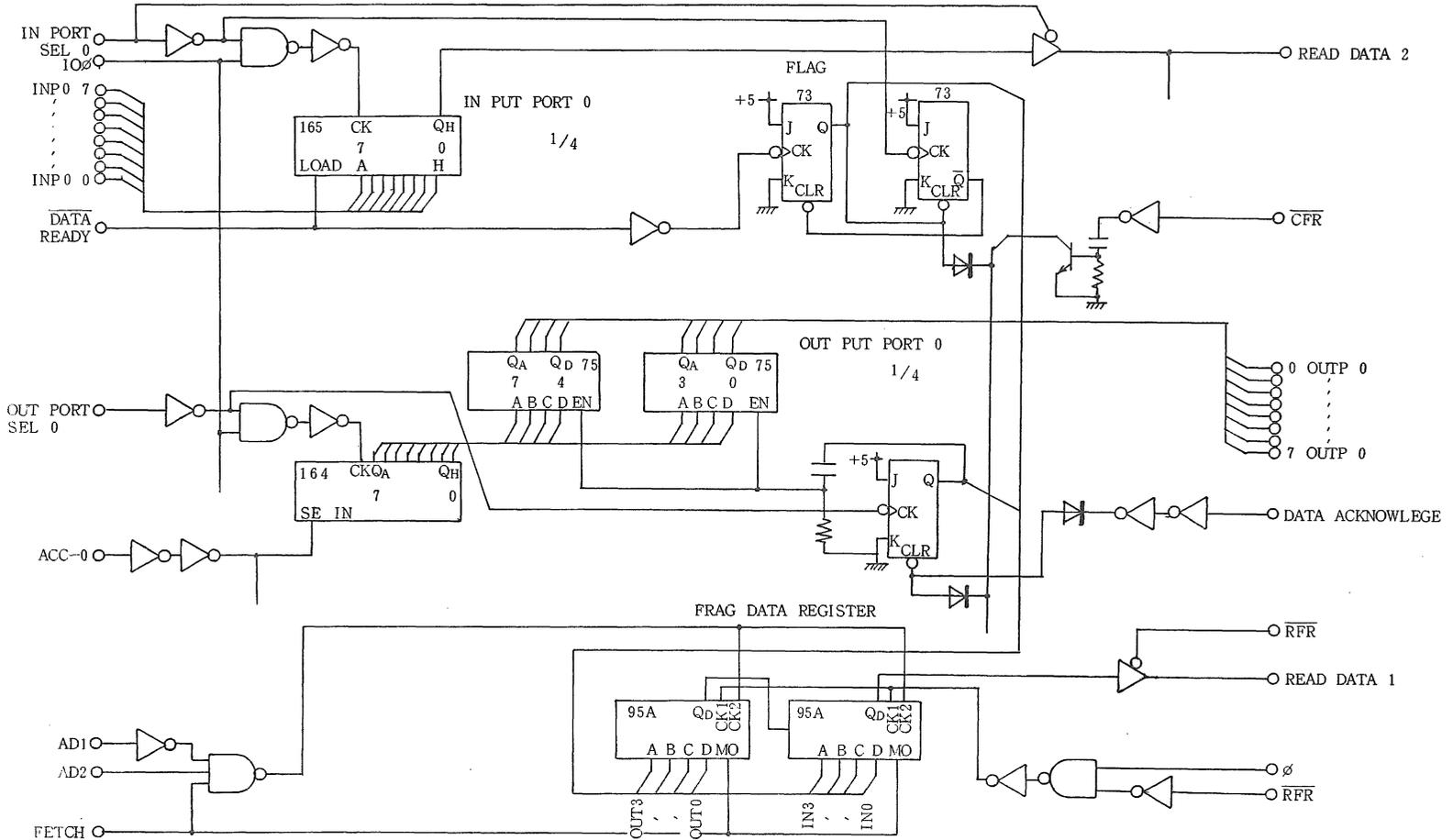


図8 I. O PORT

表3 キーボードによる加減算プログラム

0-PAGE				4-PAGE							
LOC	LABEL	MNTR	OPERAND	MACHIN	CODE	LOC	LABEL	MNTR	OPERAND	MACHIN	CODE
0	SRCH KEY	RSH	(7)	1	1 1 1 0 0 0 1	0	CALC DL 4	RIX	(3)	NUMB DATA	1 1 1 1 0 0 1
1	DL 0	JMP	KEY IN	1	1 0 0 0 0 1 0	1		WRT	(1)	COPY NUMB DATA 1	1 0 1 1 0 0 1
2		RED	DL 0	1	1 0 0 0 0 1 0	2		RIX	(1)	FUNK N DATA FLAG	1 1 1 1 0 0 1
3		JMP	SRCH KEY	1	1 0 0 0 0 0 0	3		LSH	(4)		0 0 1 0 1 0 0
4	KEY IN	WIX	(0)	1	1 1 1 1 0 0 0	4		SUB	(2)		1 0 0 1 0 0 0
5		RSH	(4)	1	0 0 1 0 0 1 0	5		JMP	PLUS	1 0 0 0 1 0 0	
6		WRT	KEY DATA 0	1	0 0 1 0 0 1 0	6		RIX	(2)	COPY NUMB DATA 1	1 1 1 1 0 0 1
7		WRT	KEY DATA	1	0 0 1 0 0 1 0	7		SUB	(2)		1 1 1 1 0 0 1
8		WIX	(1)	1	1 1 1 1 1 1 1	8		DEC	(1)		1 1 1 1 0 0 1
9		RSH	(4)	1	0 0 1 0 0 1 0	9		WRT	DL 6	1 0 0 0 0 0 0	
A		JMP	TO NUMB KEY	1	1 0 0 1 0 0 1	10		JMP	TO CHECK SIGN	1 1 0 0 0 0 0	
B		RED	KEY DATA	1	0 0 1 1 1 1 1	11		RPL	(6)		1 0 0 0 0 0 0
C		SUB	(5)	1	0 0 1 1 1 1 0	12		ADD	(2)	COPY NUMB DATA 1	1 1 1 1 0 0 1
D		JMP	TO FUNC KEY	1	1 0 0 0 0 0 0	13		WIX	(2)		1 1 1 1 1 1 0
E	TO CONT KEY	BPL	4	1	0 0 1 1 0 1 0	14		RED	DL 4	1 0 0 0 0 0 0	
F		RED	CONT KEY	1	1 0 0 0 0 0 0	15		JMP	TO CHECK SIGN	1 1 0 0 0 1 0	
G		BPL	1	1	0 0 1 1 0 0 0	16		RPL	(6)		0 0 1 0 0 0 0
H		RED	DL 1	1	1 0 0 0 0 0 0	17		JMP	TO CLAR KEY	1 1 0 1 1 0 1	
I		RPL	(1)	1	1 0 0 0 0 0 0	18		RPL	3		0 0 1 0 0 1 1
J		RED	NUMB KEY	1	1 0 0 0 0 0 0	19		IA	=KEY	1 0 0 0 0 0 0	
K		BPL	3	1	0 0 1 1 0 0 1	20		RPL	7		0 0 1 0 0 1 1
L		JMP	DL 3	1	1 0 0 0 0 0 0	21		IA	DL 7	1 0 0 0 0 0 0	
M		RPL	(1)	1	1 0 0 0 0 0 0	22		JMP	CLAR	1 0 0 0 0 1 1	
N		JMP	FUNK KEY	1	1 1 0 0 1 1 1	23				0 0 0 0 0 1 0	
O						24				1 1 1 1 1 0 0	
P						25				0 0 1 0 1 0 1	
Q						26				1 1 0 1 1 0 1	
R						27				0 0 1 0 0 0 0	
S						28				1 0 0 0 0 0 0	
T						29				1 1 1 1 1 0 0	
U						30				0 0 1 0 0 0 0	
V						31				1 1 1 1 1 0 0	
W						32				1 1 1 1 1 0 0	
X						33				1 1 1 1 1 0 0	
Y						34				1 1 1 1 1 0 0	
Z						35				1 1 1 1 1 0 0	
01						36				1 1 1 1 1 0 0	
02						37				1 1 1 1 1 0 0	
03						38				1 1 1 1 1 0 0	
04						39				1 1 1 1 1 0 0	
05						40				1 1 1 1 1 0 0	
06						41				1 1 1 1 1 0 0	
07						42				1 1 1 1 1 0 0	
08						43				1 1 1 1 1 0 0	
09						44				1 1 1 1 1 0 0	
10						45				1 1 1 1 1 0 0	
11						46				1 1 1 1 1 0 0	
12						47				1 1 1 1 1 0 0	
13						48				1 1 1 1 1 0 0	
14						49				1 1 1 1 1 0 0	
15						50				1 1 1 1 1 0 0	
16						51				1 1 1 1 1 0 0	
17						52				1 1 1 1 1 0 0	
18						53				1 1 1 1 1 0 0	
19						54				1 1 1 1 1 0 0	
20						55				1 1 1 1 1 0 0	
21						56				1 1 1 1 1 0 0	
22						57				1 1 1 1 1 0 0	
23						58				1 1 1 1 1 0 0	
24						59				1 1 1 1 1 0 0	
25						60				1 1 1 1 1 0 0	
26						61				1 1 1 1 1 0 0	
27						62				1 1 1 1 1 0 0	
28						63				1 1 1 1 1 0 0	
29						64				1 1 1 1 1 0 0	
30						65				1 1 1 1 1 0 0	
31						66				1 1 1 1 1 0 0	
32						67				1 1 1 1 1 0 0	
33						68				1 1 1 1 1 0 0	
34						69				1 1 1 1 1 0 0	
35						70				1 1 1 1 1 0 0	
36						71				1 1 1 1 1 0 0	
37						72				1 1 1 1 1 0 0	
38						73				1 1 1 1 1 0 0	
39						74				1 1 1 1 1 0 0	
40						75				1 1 1 1 1 0 0	
41						76				1 1 1 1 1 0 0	
42						77				1 1 1 1 1 0 0	
43						78				1 1 1 1 1 0 0	
44						79				1 1 1 1 1 0 0	
45						80				1 1 1 1 1 0 0	
46						81				1 1 1 1 1 0 0	
47						82				1 1 1 1 1 0 0	
48						83				1 1 1 1 1 0 0	
49						84				1 1 1 1 1 0 0	
50						85				1 1 1 1 1 0 0	
51						86				1 1 1 1 1 0 0	
52						87				1 1 1 1 1 0 0	
53						88				1 1 1 1 1 0 0	
54						89				1 1 1 1 1 0 0	
55						90				1 1 1 1 1 0 0	
56						91				1 1 1 1 1 0 0	
57						92				1 1 1 1 1 0 0	
58						93				1 1 1 1 1 0 0	
59						94				1 1 1 1 1 0 0	
60						95				1 1 1 1 1 0 0	
61						96				1 1 1 1 1 0 0	
62						97				1 1 1 1 1 0 0	
63						98				1 1 1 1 1 0 0	
64						99				1 1 1 1 1 0 0	
65						100				1 1 1 1 1 0 0	
66						101				1 1 1 1 1 0 0	
67						102				1 1 1 1 1 0 0	
68						103				1 1 1 1 1 0 0	
69						104				1 1 1 1 1 0 0	
70						105				1 1 1 1 1 0 0	
71						106				1 1 1 1 1 0 0	
72						107				1 1 1 1 1 0 0	
73						108				1 1 1 1 1 0 0	
74						109				1 1 1 1 1 0 0	
75						110				1 1 1 1 1 0 0	
76						111				1 1 1 1 1 0 0	
77						112				1 1 1 1 1 0 0	
78						113				1 1 1 1 1 0 0	
79						114				1 1 1 1 1 0 0	
80						115				1 1 1 1 1 0 0	
81						116				1 1 1 1 1 0 0	
82						117				1 1 1 1 1 0 0	
83						118				1 1 1 1 1 0 0	
84						119				1 1 1 1 1 0 0	
85						120				1 1 1 1 1 0 0	
86						121				1 1 1 1 1 0 0	
87						122				1 1 1 1 1 0 0	
88						123				1 1 1 1 1 0 0	
89						124				1 1 1 1 1 0 0	
90						125				1 1 1 1 1 0 0	
91						126				1 1 1 1 1 0 0	
92						127				1 1 1 1 1 0 0	
93						128				1 1 1 1 1 0 0	
94						129				1 1 1 1 1 0 0	
95						130				1 1 1 1 1 0 0	
96						131				1 1 1 1 1 0 0	
97						132				1 1 1 1 1 0 0	
98						133				1 1 1 1 1 0 0	
99						134				1 1 1 1 1 0 0	
100						135				1 1 1 1 1 0 0	
101						136				1 1 1 1 1 0 0	
102						137				1 1 1 1 1 0 0	
103						138					

を入れる番地の記憶に用いている。

4.2 キーボードによる加減算

1桁の数の加減算をキーボードからの入力で行い、出力に接続したセグメントに表示するプログラムを表3に示す。またフローチャートを図11に示す。

5. あとがき

MIC-II は基本的計算機として、ハード・ソフト両面にわたり、計算機に関する幅広い学習に応えるだけの機能があり、今後、工業高校乃至大学における実習教材として、或いはまた中学校におけるクラブ活動等の場において十分な活用が期待できる。なお MIC-II は MIC-

2) I の機能を更に拡充したものであり、卒業研究³⁾の一部として MC-II の製作に従事した立石昌弘君の労に感謝したい。

文 献

- 1) 富崎新：1101 RAM を用いた ATOM-8, トランジスタ技術別冊, CQ 出版, 昭和51年 5月10日
- 2) 駅田省吾, 成相友之：マイクロコンピュータとその入出力装置の製作, 島根大学, 教育学部, 卒業研究, 昭和52年 1月
- 3) 立石昌弘：マイクロコンピュータの応用, 島根大学教育学部, 卒業研究, 昭和54年 1月

図11 キーボードによる加減算

