

μCOM 42, 43

1チップマイクロコンピュータ

高井 祖 (日本電気)

§ 1. はじめに

マイクロコンピュータが広範囲の産業分野に応用されようとする時、夫々の応用分野に於る機能を有し、システム的に使い易く、しかも経済性に優れたプロセッサが要求される。しかも現在の我々の周囲には4ビットのデータ処理で制御可能なものは多い。従って4ビットのマイクロコンピュータは、その応用分野、及びそのユーザーは千差万別となり、各応用分野に対応出来しかも使い易い事がさらに重要な課題となって来ている。

この思想に基づき、先ず応用分野を大きく、'演算用' '制御用'と2つに分けて開発された専用化マイクロプロセッサがμCOM 42及びμCOM 43であり、μCOM 42は演算を主とするシステムを、μCOM 43は制御を主とするシステムを、夫々ねらった1チップ4ビットマイクロコンピュータである。

又、マイクロコンピュータの1チップ化は応用するシステムの経済性ばかりではなく、信頼性を向上させ、各分野毎の専用化はハードウェアの機能、ソフトウェアの使い易さを提供してくれるものである。

以下、μCOM 42, μCOM 43の順にそのアーキテクチャの概要とその特徴について述べる。

§ 2. μCOM 42のアーキテクチャとその特徴

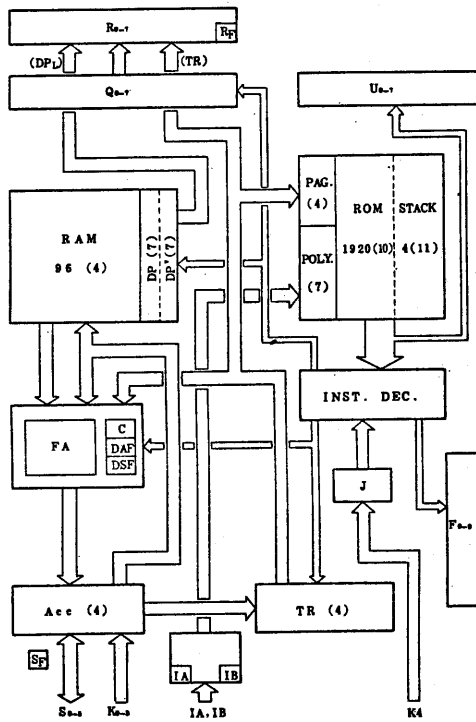
2-1 プログラムメモリとデータメモリの分離

1チップで経済性に優れた実をより有効にする為L S Iの生産コストを下げる意味で、使用されるL S I生産プロセスの考慮が必要になる。この実でμCOM 42には生産が安定しているPチャンネルAlゲートE/D-MOSプロセスを用いている。しかし命令実行速度の実では、Pチャンネルプロセスは不利になる。

上記の実を考慮し1チップマイクロコンピュータではバス分離方式を採用している。つまり、プログラムメモリ(ROM)とデータメモリ(RAM)を分離し、ROMアドレスはプログラムカウンタで、RAMアドレスはデータポイントで指定し、夫々が同時処理が可能であるようなハードウェアを構っている。又、ROM出力はインストラクションバスに、RAM出力は4ビットバスに、夫々分離されている。この様な構成をすると、プログラムメモリとデータメモリの制御系が分離されて、マイクロオードの制御が容易になり、処理速度が高速化され、複数の命令の同時実行が可能になるのである。

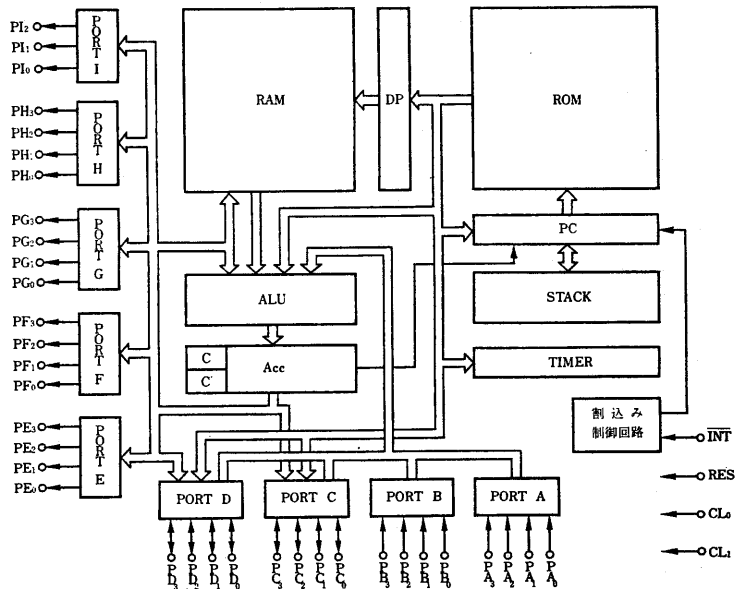
しかし、プログラムメモリとデータメモリを分離すると、データメモリの中

UCOM-42の機能ブロック図



- POLY : ホリミナル・カウンタ
- FA : フル・アダー
- TR : テンポラリ・レジスタ
- IA : 割込みポート
- IB : 割込みポート
- DAF : 10進加算モード F/F
- DSF : 10進減算モード F/F
- C : キャリー F/F
- Acc : アキュムレータ
- DP : データ・ポインタ
- DP' : 退避用データ・ポインタ

UCOM-43の機能ブロック図



- DP : データ・ポインタ
- ALU : 演算論理ユニット
- Acc : アキュムレータ
- PC : プログラム・カウンタ
- C : キャリー F/F
- C' : キャリー退避 F/F

にプログラムを入れたり、データによるプログラムの分岐が不可能となるが、 $\mu\text{COM}42$ では、アキュムレータの内容と、ジャンプアドレスの下4ビットとの論理和をとる方式のアキュムレータジャンプ命令を命令セットの中に設けてこの不利を回避している。

次にプログラムメモリとデータメモリの容量に関しては、ソフトウェア処理による補助的な使い方ではあるが、C-MOS-RAMを外付け出来るので、(I/Oポートの形態の項参照)プログラムメモリの容量に重きを置いた設計になっており約2Kステップのプログラム領域を持っている。(ハードウェアの容量としては2096ステップであるがテスト用として128ステップを使用している(テスト方法の項参照))。

2-2 命令体系

$\mu\text{COM}42$ は演算向きであることから、4ビットのデータ処理を容易に出来る命令に重きを置いているが、(命令表参照)前項で述べたようにプログラムメモリの容量との兼ね合いで、プログラムメモリの有効活用を計り、同じメモリサイズでもより多くの処理が出来るように、又、より多くの分野のユーザーに使い易くという点を考慮し、1語を10ビットとして、全ての命令を1語命令を用意してプログラムメモリの効率を上げるようにした。

プログラムメモリとデータメモリの分離により実現可能となった複数の命令の同時処理に関しては、いわゆるミニコンピュータのオートインクリメント・スキップに相当するデータメモリ操作命令を有しており、その一例として下記に示すXMI命令が掲げられる。

- XMI
- (1) アキュムレータの内容とデータポインタで示されるメモリの内容の交換。
 - (2) データポインタの上位3ビットを命令の一部でEOR修飾する。
 - (3) データポインタの下位4ビットをインクリメントする。
 - (4) データポインタの下位4ビットが H_H になるとSTATUSをセットし次の命令をSKIPする。

その他の複合命令の例としては出力ポートの操作の例でも示される。演算型のシステムには必要であるダイナミックセグメント方式の表示もある場合に使うOUI命令の例を下記に示す。

U出力ポートはセグメント信号に用い、R出力ポートは桁信号に用いQレジスタはあらかじめ次の桁信号を設定しておく際のレジスタである。

- OIU
- (1) プログラムメモリの下位8ビットをUポートに出力する。
 - (2) Qレジスタの内容をRポートを構成するRレジスタに並列転送する。

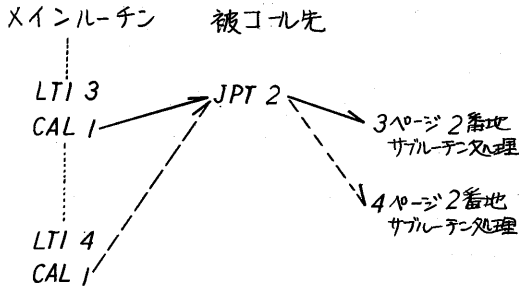
μCOM-42 インストラクション活用表

命令群	ニーモニック	命令コード	マシン #(7%)	オペレーション	スキップ 条件
アキムレニア操作命令	CMA	000000 0001	1	Acc←(Acc)	
	CIA	000011 0000	1	Acc←(Acc)+1	
	INA	000011 0001	½	Acc←(Acc)+1	Carry=1
	DEA	000011 1111	½	Acc←(Acc)-1	Borrow*1
	RFC	000111 0010	1	C←0	
	SFC	000111 0011	1	C←1	
演算命令	DSM	000001 1101	1	Decimal Subtract Mode	
	DAM	000001 1100	1	Decimal Add Mode	
	AD	000001 0000	½	Acc←(Acc)+(DP)	Carry=1
	ADC	000001 0010	1	Acc←(Acc)+(DP)+(C)	
	ADI	000011 1111	½	Acc←(Acc)+1	Carry=1
	ロード・ストア命令	LM	000101 0 M ₂ M ₁ M ₀	1	Acc←(DP) DP _n ←(DP _n)∨M ₂ M ₁ M ₀
XM		000101 1 M ₂ M ₁ M ₀	1	(Acc)←(DP) DP _n ←(DP _n)∨M ₂ M ₁ M ₀	
XMI		000100 0 M ₂ M ₁ M ₀	½	(Acc)←(DP) DP _n ←(DP _n)∨M ₂ M ₁ M ₀ DP _n ←(DP _n)+1	(DP _n)=8 or (DP _n)=0
XMD		000100 1 M ₂ M ₁ M ₀	½	(Acc)←(DP) DP _n ←(DP _n)∨M ₂ M ₁ M ₀ DP _n ←(DP _n)-1	(DP _n)=F or (DP _n)=7
LI*		001111 1 L ₂ L ₁ L ₀	1	Acc←L ₂ L ₁ L ₀	
LDI*		001111 0 L ₂ L ₁ L ₀	1	DP←L ₂ L ₁ L ₀	
アークボウイング操作命令	IND	000110 0001	½	DP _n ←(DP _n)+1	(DP _n)=8 or (DP _n)=0
	DED	000110 1001	½	DP _n ←(DP _n)-1	(DP _n)=F or (DP _n)=7
	XDP	000001 1111	1	(DP) _n ←(DP) _n	
	ZAC	000001 1110	1	000DP _n ←(DP)	
	XTA	000110 0000	1	(Acc)←(TR)	
レジスタ操作命令	LTI*	001110 1 L ₂ L ₁ L ₀	1	TR←L ₂ L ₁ L ₀	
	QS1	000111 0001	1	Q _{n+1} ←Q _n , Q _n ←1	
	QS0	000111 0000	1	Q _{n+1} ←Q _n , Q _n ←0	
	ビット操作命令	SB	000111 1 B ₂ B ₁ B ₀	1	(DP, B ₂ , B ₁ , B ₀)←1
RB		000111 0 B ₂ B ₁ B ₀	1	(DP, B ₂ , B ₁ , B ₀)←0	
SBT		000111 0 1 B ₂ B ₁ B ₀	½	Skip if (DP, B ₂ , B ₁ , B ₀)=1	指定ビット=1
スキップ命令		SC	000110 1000	½	Skip if (C)=1
	SEM	000001 0001	½	Skip if (Acc)=(DP)	(Acc)=(DP)
	SEI	000010 1111	½	Skip if (Acc)=L ₂ L ₁ L ₀	(Acc)=L ₂ L ₁ L ₀
	SK4	000001 0011	½	Skip if K4=1	K4=1
ジャンプ命令	JPT	1 0 1 R ₂ R ₁ R ₀ P ₂ P ₁ P ₀	1	PC←(TR), P _n ←0	
	JPA	0 1 1 R ₂ R ₁ R ₀ P ₂ P ₁ P ₀	1	PC _n ←P _n -4 PC ₂₋₀ ←P ₂₋₀ +V (Acc)	
	JCP	0 1 0 R ₂ R ₁ R ₀ P ₂ P ₁ P ₀	1	PC _n ←P _n -4	
制御命令	CAL	1 0 0 R ₂ R ₁ R ₀ P ₂ P ₁ P ₀	1	(PC) _n ←(STACK) 1000P ₂ P ₁ P ₀ P ₂ P ₁ P ₀ →PC	
	RT	000000 1110	1	PC←(STACK)	
	RTS	000000 1111	2	PC←(STACK) PC←PC+1	無条件
周辺制御命令	EIA	000000 1000	1	Enable IA port	
	DIA	000000 1001	1	Disable IA port	
	EIB	000110 1110	1	Enable IB port	
	DIB	000110 1111	1	Disable IB port	

*: 同種の命令が続けて実行されると、2番目以降の命令はNOPになります。

命令群	ニーモニック	命令コード	マシン #(7%)	オペレーション
入	OIU	1 1 1 L ₂ L ₁ L ₀ L ₂ L ₁ L ₀	1	U ₇₋₀ ←L ₂ →R ₇₋₀ R ₇₋₀ ←Q ₇₋₀
	ERO	000001 0111	1	Enable R port
	DRO	000001 0110	1	Disable R port
	OQR	000001 1000	1	R←(Q)
	OTR	000001 1001	1	R ₂₋₀ ←(TR), R ₇₋₀ ←(DP _n)
	SFS	000001 0101	1	S←(Acc)
	RFS	000001 0100	1	S port Input Mode
	IS	000001 1011	1	Acc←S
	IK	000001 1010	1	Acc←K
	出力	RF1	000000 0010	1
SF1		000000 0011	1	F ₁ ←1
RF2		000000 0100	1	F ₂ ←0
SF2		000000 0101	1	F ₂ ←1
RF3		000000 0110	1	F ₃ ←0
SF3		000000 0111	1	F ₃ ←1
RF4		000000 1010	1	F ₄ ←0
SF4		000000 1011	1	F ₄ ←1
RF5		000000 1100	1	F ₅ ←0
SF5		000000 1101	1	F ₅ ←1
力命令	RF6	000110 0010	1	F ₆ ←0
	SF6	000110 0011	1	F ₆ ←1
	RF7	000110 0100	1	F ₇ ←0
	SF7	000110 0101	1	F ₇ ←1
	RF8	000110 0110	1	F ₈ ←0
	SF8	000110 0111	1	F ₈ ←1
	RF9	000110 1010	1	F ₉ ←0
	SF9	000110 1011	1	F ₉ ←1
	RF0	000110 1100	1	F ₀ ←0
	SF0	000110 1101	1	F ₀ ←1
その他	NOP	000000 0000	1	No Operation

全てが1語命令になっているが、組み合わせを必要とする命令の1つは、ページ外ジャンプ命令(JPT)である。これはテンポラリレジスタTで示されるページ(LTI命令で指定される)にジャンプするものである。この方式を採用した為に、コール命令(CAL)は128通りしか飛ばずサブルーチンの数としてはその半である64ヶしか設定出来無いが、ページ指定をした後にコール命令をする事で擬似的に全プログラムメモリアドレス空間をコールする事が出来る。



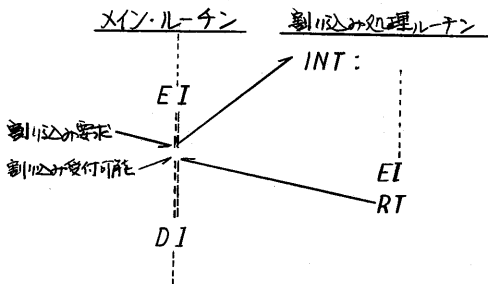
2-3 割り込み機能

割り込みの方式は、ハードウェアによる方法かソフトウェアによる方法、固定アドレスの方法かスタートアドレス指定の方法、割り込要求信号の変化による方法か信号レベルによる方法、と各種有るが、μCOM42では系統的に使い易く効率の良い方式であるところのハードウェアによる固定番地の立ち上がりエッジトリガ方式を用いている。

割り込みとして INT-A と、INT-B の2ポートを携っており、両者の間には下記の如く優先度が有り、A、B 夫々別の固定番地が割り当ててある。

- (1) Aが入るとA、B共に割り込み禁止となる。
- (2) Bが入るとBのみ割り込み禁止となる。
- (3) A、B両者が同時に入るとAが優先され(1)と同様の動作となる。

なお割り込み処理の速りには特別な命令を携っていないので、(RT命令を用いる)割り込み許可命令(EI命令)は1命令実行後に割り込み受付可能状態になるようにし、スタックオーバーを防いでいる。



又、割り込みレベルはデータポインタが2系統なので1レベルの制限となっているが、データポインタが破壊されても良いプログラムに使用する場合は2レベルが許される。

これら割り込み機能は、主対象をインパクトタイアのプリンタとしており、プリンタのタイミング信号をBに、リセット信号をAに、夫々割り当てている。

2-4 ALUの形態

μCOM42は演算型のマイクロコンピュータを目的とし、論理制御的な機能よりもむしろ有効な演算専用機能を盛り込んである。

その一つは10進演算の機能であり、プログラムを容易にする為に10進補正のハードウェアを備えている。10進加算の例をとると、10進加算モード命令DAM (DECIMAL ADD MODE) の次にADC命令 (ADD WITH CARRY) をプログラムする事で自動的に10進補正を実行するものである。このDAM命令は演算のみならず次に続く命令の種類を選択する事によりモード変換に利用する事が出来る。

なおビット操作に関してはRAMで行い、データポインタで示されるデータメモリのビットテスト、セット、リセットが可能であり、演算中のフラグ操作を意識している。

演算に関して、基本演算桁数(ワード数)は8桁としてあり、データポインタにハードウェアのストップを持っていて、ソフトウェア処理をすれば桁の制限はない。

2-5 I/Oポートの形態

専用化という観点からμCOM42はその入力及び出力ポートはかなり使用目的を考慮した設計を取り入れている。

(1) 表示、プリンタ関係ポート

表示関係のポートとしてはDポート及びRポートを備えており命令体系の項で述べたように、いわゆるセグメントデコードをハードウェアで持たずに、プログラムメモリ内にデコード機能を持たせ、デコードすべきコードに自由度を与え、論理上256通りのコードを出力可能にしている。又、Rポートは1命令でマスク可能な出力ポートであり、8ビット直列接続の命令によりシフトセット、シフトリセット可能なQレジスタに並列に接続されているので表示府信号及びプリンタのハンニマドライブ信号を自由に発生する事が出来るものである。

(2) キースキャン及びスイッチ関係入出力ポート

キースキャン信号としては表示に用いたDポート又はRポートが使用可能であり、その入力としては、アキュムレータと接続された4ビットKポート及びKテストポートを備えている。

(3) RAM拡張用ポート

内蔵されたデータメモリ領域では不十分な場合、又は電源断時に於いてもデータの保護の必要な場合にC-MOS-RAMを増設出来るように、データ授受の出来る入出力ポートを備え、アドレス信号としては前述のRポートを使用する。この場合、Rポートは内部のデータポインタと接続されているので互いにデータ転送をするプログラムは容易に出来る。

(4) ディスクリートポート

1ビット毎に独立したセット、リセット可能な10ヶの出力ポートを設けてあり、ランプ、アランジヤ、リレー等の制御が自由に出来るように考慮されている。

§3. $\mu\text{COM}43$ のアーキテクチャとその特徴

3-1 プログラムメモリとデータメモリの分離

$\mu\text{COM}43$ に於ても $\mu\text{COM}42$ と同一の思想からROMとRAMの分離を計っており、データによる分岐に関しては、プログラムカウンタ下位6ビットのうち上位4ビットにアキュムレータの内容をロードし、下位2ビットに0をロードする事でデータによるプログラム分岐を実現している。

ROMとRAMの容量に関しては、ROMは約2Kステップ(ハードウェアの容量としては2096ステップであるがテスト用として96ステップを使用している)を持っている。

応用分野によってかなり異なっているが、1つの方向としてROM、RAM共に容量の少なくして良い小さいシステムの需要に対処出来るように $\mu\text{COM}44$ 、 $\mu\text{COM}45$ を開発している。

3-2 命令体系

前項の同一思想から命令体系に於ても $\mu\text{COM}42$ と同様、XMI命令で代表される複合命令を備えている。但し $\mu\text{COM}43$ は1語を8ビットで構成しているので、IO-外ジマン、コール等は2語構成をしているが、4ステップ毎にダブルチンの先頭アドレスを指定出来るコール命令も備え、プログラムの効率化を計っている。

3-3 割り込み機能

$\mu\text{COM}43$ の場合は、割り込みの使用目的が広範囲であるので、ハードウェアにある固定番地の立ち下がりエッジトリガ方式のものを1入力用意してあるが、割り込み処理中でも割り込み要求信号を記憶するフリックフロップを持っており、これは命令によりテストリセット可能であるのでソフトウェア処理にあるレベル割り込みも可能とされている。

割り込み処理から復帰する場合は $\mu\text{COM}42$ と同じ処理方法を採用している。

3-4 ALUの形態

$\mu\text{COM}43$ は制御型のマイクロコンピュータとして、4ビットの数値データ以外に、1ビットのデータを処理する機会が多く、ビット操作が豊富にしかも容易出来るように考慮されている。

アキュムレータのローテーション、アキュムレータとメモリ内容の同一ピン

μCOM-43 インストラクション活用表

命令群	ニーモニック	命令コード			ワイト ビット	オペレーション	スキップ条件
		D ₇ D ₆ D ₅	D ₄ D ₃ D ₂	D ₁ D ₀			
アキユムレック操作命令	CLA	1001	0000	1 1	Acc←0		
	CMA	0001	0000	1 1	Acc←(Acc)		
	CIA	0001	0001	1 1	Acc←(Acc)+1		
	INC	0000	1101	1 1/2	Acc←(Acc)+1 skip if Carry	Carry	
	DEC	0000	1111	1 1/2	Acc←(Acc)-1 skip if Borrow	Borrow	
	CLC	0000	1011	1 1	C←0		
	STC	0001	1011	1 1	C←1		
	XC	0001	1010	1 1	C←(C)		
	RAR	0011	0000	1 1	(Acc←)←(Acc) C←(Acc), (Acc)←(C)		
	増減乗除命令	INM	0001	1101	1 1/2	((DP))←((DP))+1 skip if ((DP))=0	((DP))=0
DEM		0001	1111	1 1/2	((DP))←((DP))-1 skip if ((DP))=F	((DP))=F	
演算命令	AD	0000	1000	1 1/2	Acc←(Acc)+((DP)) skip if Carry	Carry	
	ADS	0000	1001	1 1/2	Acc, C←(Acc)+((DP))+C skip if Carry	Carry	
	ADC	0001	1001	1 1	Acc, C←(Acc)+((DP))+C		
	DAA	0000	0110	1 1	Acc←(Acc)+6		
	DAS	0000	1010	1 1	Acc←(Acc)-10		
	EXL	0001	1000	1 1	Acc←(Acc)∨((DP))		
	LI	1001	1011	1 1	Acc←IsIsIsIs		
	S	0000	0010	1 1	((DP))←(Acc)		
	L	0011	1000	1 1	Acc←((DP))		
	LM	0011	10M ₆ M ₅	1 1	Acc←((DP)) DP _n ←(DP _n)∨0M ₁ M ₀		
ロード・ストア命令	X	0010	1000	1 1	(Acc)←((DP)) DP _n ←(DP _n)∨0M ₁ M ₀		
	XM	0010	10M ₆ M ₅	1 1	(Acc)←((DP)) DP _n ←(DP _n)∨0M ₁ M ₀		
	XD	0010	1100	1 1/2	(Acc)←((DP)) DP _L ←(DP _L)-1 skip if (DP _L)=F	(DP _L)=F	
	XND	0010	11M ₆ M ₅	1 1/2	(Acc)←((DP)) DP _n ←(DP _n)∨0M ₁ M ₀ DP _L ←(DP _L)-1 skip if (DP _L)=F	(DP _L)=F	
	XI	0011	1100	1 1/2	(Acc)←((DP)) DP _L ←(DP _L)+1 skip if (DP _L)=0	(DP _L)=0	
	XM1	0011	11M ₆ M ₅	1 1/2	(Acc)←((DP)) DP _n ←(DP _n)∨0M ₁ M ₀ DP _L ←(DP _L)+1 skip if (DP _L)=0	(DP _L)=0	
	アキユムレック操作命令	LDI	0001 0IsIsIs	0101 IsIsIsIs	2 2	DP←Is←Is	
		LDZ	1000	IsIsIsIs	1 1	DP _n ←0 DP _L ←IsIsIsIs	
		DED	0001	0011	1 1/2	DP _L ←(DP _L)-1 skip if (DP _L)=F	(DP _L)=F
		IND	0011	0011	1 1/2	DP _L ←(DP _L)+1 skip if (DP _L)=0	(DP _L)=0
TAL		0000	0111	1 1	DP _L ←(Acc)		
TLA		0001	0010	1 1	Acc←(DP _L)		
ワイト・アキユムレック操作命令	XHX	0100	1111	1 2	(X)←(DP _n)		
	XLY	0100	1110	1 2	(Y)←(DP _n)		
	THX	0100	0111	1 2	X←(DP _n)		
	TLY	0100	0110	1 2	Y←(DP _n)		

命令群	ニーモニック	命令コード			ワイト ビット	オペレーション	スキップ条件
		D ₇ D ₆ D ₅	D ₄ D ₃ D ₂	D ₁ D ₀			
ワイト・アキユムレック操作命令	XAZ	0100	1010	1 2	(Z)←(Acc)		
	XAW	0100	1011	1 2	(W)←(Acc)		
	TAZ	0100	0010	1 2	Z←(Acc)		
	TAW	0100	0011	1 2	W←(Acc)		
	XHR	0100	1101	1 2	(R)←(DP _n)		
	XLS	0100	1100	1 2	(S)←(DP _L)		
	SMB	0111	10B ₆ B ₅	1 1	((DP, B ₆ B ₅))←1		
	RMB	0110	10B ₆ B ₅	1 1	((DP, B ₆ B ₅))←0		
	TMB	0101	10B ₆ B ₅	1 1/2	skip if ((DP, B ₆ B ₅))=1	((DP, B ₆ B ₅))=1	
	TAB	0010	01B ₆ B ₅	1 1/2	skip if (Acc(B ₆ B ₅))=1	(Acc(B ₆ B ₅))=1	
ビット操作命令	CMB	0011	01B ₆ B ₅	1 1/2	skip if (Acc(B ₆ B ₅))=1 = (DP, B ₆ B ₅)	(Acc(B ₆ B ₅))=1 = ((DP, B ₆ B ₅))	
	SFB	0111	11B ₆ B ₅	1 2	FLAG (B ₆ B ₅)←1		
ワイト・アキユムレック操作命令	RFB	0110	11B ₆ B ₅	1 2	FLAG (B ₆ B ₅)←0		
	FBT	0101	11B ₆ B ₅	1 2	skip if (FLAG (B ₆ B ₅))=1	(FLAG (B ₆ B ₅))=1	
	FBF	0010	00B ₆ B ₅	1 2	skip if (FLAG (B ₆ B ₅))=0	(FLAG (B ₆ B ₅))=0	
	CM	0000	1100	1 1/2	skip if (Acc)←((DP))	(Acc)←((DP))	
比較命令	CI	0001 1100	0111 IsIsIsIs	2 2	skip if (Acc)=IsIsIsIs	(Acc)=IsIsIsIs	
	CLI	0001 1110	0110 IsIsIsIs	2 2	skip if (DP _L)=IsIsIsIs	(DP _L)=IsIsIsIs	
スキップ命令	TC	0000	0100	1 1/2	skip if (C)=1	(C)=1	
	TIT	0000	0011	1 1/2	skip if (INT F/F)=1 INT F/F←0	(INT F/F)=1	
ジャンプ命令	JCP	11RR	RRRR	1 1	PC _n ←A←R←R←R←R		
	JMP	1010	0P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀	2 2	PC←P ₆ ←P ₅ ←P ₄ ←P ₃ ←P ₂ ←P ₁ ←P ₀		
	JPA	0100	0001	1 2	PC _n ←A←A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀		
戻り命令	EI	0011	0001	1 1	INTE F/F←1		
	DI	0000	0001	1 1	INTE F/F←0		
スタック命令	CZP	1011	P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀	1 1	STACK←(PC) PC←0000 P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀		
	CAL	1010	1P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀	2 2	STACK←(PC) PC←P ₆ ←P ₅ ←P ₄ ←P ₃ ←P ₂ ←P ₁ ←P ₀		
	RT	0100	1000	1 2	PC←(STACK)		
	RTS	0100	1001	1 3-4	PC←(STACK) PC←(PC)+1, 2	無条件	
タイマ命令	STM	0001	0100 10IsIsIs	2 2	TM F/F←0 TIMER←Is←Is		
	TTM	0000	0101	1 1/2	skip if (TM F/F)=1	(TM F/F)=1	
入出力命令	SEB	0111	01B ₆ B ₅	1 2	PORT E (B ₆ B ₅)←1		
	REB	0110	01B ₆ B ₅	1 2	PORT E (B ₆ B ₅)←0		
	SPB	0111	00B ₆ B ₅	1 1	PORT (DP _L , B ₆ B ₅)←1		
	RPB	0110	00B ₆ B ₅	1 1	PORT (DP _L , B ₆ B ₅)←0		
	TPA	0101	01B ₆ B ₅	1 1/2	skip if (PORT A (B ₆ B ₅))=1	(PORT A (B ₆ B ₅))=1	
	TPB	0101	00B ₆ B ₅	1 1/2	skip if (PORT (DP _L , B ₆ B ₅))=1	(PORT (DP _L , B ₆ B ₅))=1	
命令	OE	0100	0100	1 2	PORT E←(Acc)		
	OP	0000	1110	1 1	PORT (DP _L)←(Acc)		
	OCD	0001	1110 IsIsIsIs	2 2	PORT C, D←Is←Is		
	IA	0100	0000	2 2	Acc←(PORT A)		
命令	IP	0011	0010	1 1	Acc←(PORT (DP _L))		
	NOP	0000	0000	1 1	No Operation		

ト比較，アキュムレータのビットテスト等が可能で，これらの思想は次に述べる I/Oポートにも取り入れられている。

3-5 I/Oポートの形態

前述の如く，ビット単位の操作に重点が置かれ，プランジヤー，リレー，ランプ等の制御が自由に出来るよう考慮されている。

又，少数桁の表示が実現可能であるように ROM のデータを直接出力するポートと，キー読み込みが出来る 4 ビット単位の入力ポートを備えており，これらの機能を下記の表に示す。

○印：機能あり

入出力 命令	機能	ポート								
		樹 DPL A	B	C	D	E	F	G	H	*出力 I
		0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8
SEB	ビット単位のセット					*○				
REB	ビット単位のリセット					*○				
SPB	ビット単位のセット					○	○	○	○	○
RPB	ビット単位のリセット					○	○	○	○	○
TPA	ビット単位のテスト	*○								
TPB	ビット単位のテスト	○	○	○	○					
OE	4 ビット単位の出力					*○				
OP	4 ビット単位の出力			○	○	○	○	○	○	**○
OCD	イミディエト・データ出力			*○	*○					
IA	4 ビット単位の入力	*○								
IP	4 ビット単位の入力	○	○	○	○					

注：ポート指定の DPL コード（上は 2 進，下は 1 6 進）。ビット指定は命令の 2 ビット（B: B₀）で行う。

*：DPL で指定しなくてよい。

**：出力ポート I は 8 ビット構成。他のポートはすべて 4 ビット構成。

なお，μcom 43 にはプログラマブルタイマを設け時間的制御を必要とする分野への考慮がなされている。このプログラマブルタイマは 12 ビットで構成され上位 6 ビットがソフトプログラマブルになっており，標準クロックサイクル時に 630 μs ~ 40.32 ms の時間設定が可能である。

§4 テスト方法とプログラム開発

マイクロコンピュータが1チップで構成されると、従来の複数チップで構成される場合と異なり、プログラム開発と、LSIのテスト方法が問題になって来る。しかも両者共マイクロコンピュータを開発するに当ってそのアーキテクチャに少なからず影響を与えるものである。

4-1 テスト方法

1チップマイクロコンピュータをテストという観点から考える時、そのハードウェアは大きく (A)ROM (B)ランダム(RAMを含む)の2つに分ける事が出来る。ここで問題となるのは従来の複数チップ構成のマイクロコンピュータではROMから出力される命令を自由にCPUチップに挿入してランダム部分をテストする事が出来たが、1チップマイクロコンピュータでは、命令を出力すべきROMが同一チップ内に存在するという事で、何らかのテスト用ハードウェアを設けてこれに対処しなければならぬ。

μcom 42/43では

- (1) ROMの内容を自動的にシーケンシャルに特定端子に出力する。
- (2) 特定端子からインストラクションを挿入する。
- (3) ROMの一部領域をテスト用として使い(1),(2)で実行不可能な命令をテストする。

以上(1),(2),(3)を実現するテスト用ハードウェアを持つ事でLSIのテストを完全なものとしている。

4-2 プログラム開発

1チップマイクロコンピュータを使用するシステムのシミュレーション方法としては、ディスクリート部品によるシミュレータ、上位機種のコンピュータによるエミュレータ開発等が考えられるが、製造プロセス、量産性を考慮して64ピンセラミックDIPケースによるROM外付ハードウェアシミュレーションチップを開発した。これを基に、プログラム開発が容易出来るように設計されたPR0Mベースの開発キット(EVA-KIT)及びソフトウェアサポートとしてアセンブラを用意しプログラム開発に対処した。

謝辞 μcom 42/43を開発するに当たり、プロセス及び試作を担当していただきました集積回路専業部、山本宏彦主任、武川藤次郎氏、奥村孝一郎氏、ならびにシステム開発を指導していただきました高島二郎主任に深謝致します。