

4ビット・1チップマイクロコンピュータ MN1500シリーズ

坂 尾 隆 大 松 良 明 真 弓 和 昭

松下電器産業(株) 中央研究所

1. 緒 言

家電分野における1チップマイクロコンピュータの応用には著しいものがある。当社においても、先に4ビット・1チップマイクロコンピュータ MN1400 シリーズを発売し、テレビリモコン受信機、各種タイマ制御をはじめとして数多くの商品に利用している。しかしながら、割込み機能、入出力ポートの機能割当て、サブルーチンスタックの段数、処理速度等の問題により、MN1400 シリーズでは実現が困難な応用が出現している。

MN1500 シリーズは、この MN1400 シリーズでは実現困難な応用分野をカバーすることを目的に開発した高速、高機能 4 ビット・1 チップマイクロコンピュータである。

4 ビット・1 チップマイクロコンピュータの第1世代を T I 社の TMS1000 シリーズとすれば、MN1400 シリーズは第2世代に当たり、本 MN1500 シリーズは第3世代の 4 ビット・1 チップマイクロコンピュータと定義することが可能であろう。

2. MN1500シリーズの開発思想

2.1 開発方針

MN1500 シリーズの開発方針は、「使いやすい、高機能な 4 ビット・1 チップマイクロコンピュータを提供すること」という言葉に尽きる。

つぎに、アーキテクチャ設計時に留意した主要な点を述べる。

- 1) MN1500 シリーズの基本命令処理単位は 4 ビットであるが、データ転送、入出力および処理の一部は 8 ビット (バイト) 単位での処理を可能とした。8 ビットの処理を一部採用した理由としては、
 - a) 内蔵タイマカウンタおよびシフトバッファは 8 ビット構成であり、これらの処理を 1 命令で実行可能としたこと
 - b) RAM およびレジスタの構成をハードウェア的には 8 ビット構成にしたこと
 - c) 1 語 8 ビットで構成される命令語のテーブル参照命令を採用したこと

等があげられる。いいかえれば、命令が使用される処理内容に応じて命令の処理幅を定めたともいえる。すなわち、4 ビットのマイクロコンピュータに、8 ビットのタイマが内蔵されているとすると、従来のマイクロコンピュータでは、4 ビットの処理を 2 回実行することにより、8 ビットのタイマ処理をしていたが、プログラミングの立場からすると、1 命令で 8 ビット処理を行なった方が望ましいことは明らかである。このようにして、命令の処理幅をその処理内容によって定めた。

2) 割込み機能の充実

MN1500 シリーズは、内部割込み要因 2、外部割込み要因 2 (MN1599 ではプログラム開発時に使用するモニタ割込みが備えられているので、外部割込み要因 3) を有し、従来の 4 ビットマイクロコンピュータで

は困難であった複雑な制御を可能にした。各割込みは互いに異なる割込み開始番地を有し、外部割込みの 1 つ以外は、それぞれ独立な割込みイネーブルフラグを有している。

3) 入出力ポート割当ての自由度を上げる。

MN1400 シリーズのように、入出力ポートの割当てが固定的であると、応用例によっては入力数あるいは出力数の過不足の問題が生じやすい。そこで、MN1500 シリーズでは、入出力ポートはすべて双方向として、入出力の割当ては自由とした。この結果、応用例に最適な入出力の割振りが可能になった。

2.2 特徴

MN1500 シリーズは次の特徴を有する。

- ・命令サイクル 2 μ sec と高速
- ・全 RAM 領域の停電保護機能
- ・4 レベルの割込み (エバリュエータ 5 レベル)
- ・並列入出力 12 ポート (双方向)
- ・ディスクリット出力、バイト入出力可能
- ・直列データ転送可能 (8 ビットシフトバッファ)
- ・8 ビットタイマカウンタ (7 ビットプリスケラ付) 内蔵
- ・間接ジャンプ可能
- ・バイト処理可能 (内部データ転送、入出力、処理の一部)
- ・ROM 領域のテーブルルックアップ機能
- ・RAM 領域にレジスタの退避領域を有する
- ・全 RAM 領域直接アドレス指定演算可能
- ・ループカウンタ命令、1 バイトコール命令、10 進補正命令など効率のよい命令体系
- ・標準 124 種類の使いやすい豊富な命令
- ・サブルーチンスタックは RAM 領域を使用
- ・N チャネル LOCOS E/D MOS プロセス
- ・入出力は TTL / CMOS コンパチブル
- ・+5V 単一電源
- ・-30° ~ +70°C の広い動作温度範囲

3. MN1500シリーズのアーキテクチャ

3.1 基本構成

MN1500 シリーズには、現在、MN1542、MN1544、MN1562、MN1564 が用意されている。このほかに、エバリュエータとして MN1599、I/O エクスパンダとして MN1591 がある。

各チップの概略機能一覧を第 1 表に、MN1500 シリーズのブロック構成を第 1 図に、またこのブロック構成の機能説明を第 2 表に、MN1500 シリーズの内部バス構造図を第 2 図に示す。内部バスは 2 組の 4 ビットバスよりなり、8 ビットの転送を容易にしている。

3.2 タイミング

MN1500 シリーズは、クロック発振素子接続端子 OSC1、OSC2 を有しており、発振素子として、セラミックフィルタ、水晶振動子が使用可能である。また、チッ

3.3 データメモリ

第6図にRAMの構造を示す。MN1542/MN1562では152×4ビット、MN1544/MN1564では256×4ビットを有する。

RAM領域の中で、アドレスC0～FFはスタック領域として使用されるが、命令によってもアクセス可能である。SX(00)、SY(01)、SE(02)番地はそれぞれX、Y、Eレジスタの退避番地である。

第7図にスタック領域の格納状態を示す。

RAMは、4ビットのXレジスタ、Yレジスタによるレジスタ間接アドレス、および全領域に対する直接アドレスが可能である。スタック領域はスタックポインタSPによるレジスタ間接アドレスも可能である。

内蔵RAMには、データ保持制御線HOLDMおよび専用電源線V_{WM}が備えられており、低電力でRAMデータの保持を可能としている。

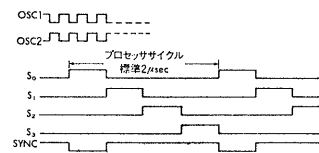
3.4 レジスタ、フラグ

内部レジスタとして、アキュムレータとして働くAレジスタ、汎用レジスタおよび拡張アキュムレータのEレジスタ、RAMのアドレスレジスタおよび汎用レジスタとして使用するXレジスタ、Yレジスタがある。

8ビットの演算、転送を実行するときには、EレジスタとAレジスタとで8ビットのレジスタを構成し処理を行なう。

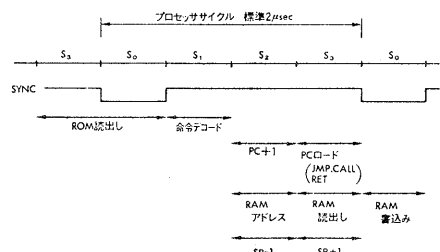
演算フラグとしては、キャリフラグ(CF)、ゼロフラ

グ(ZF)があり、これらのフラグには、演算、転送の結果がセットされる。



第4図 基本クロックのタイミング図

Fig. 4 Basic timing diagram.



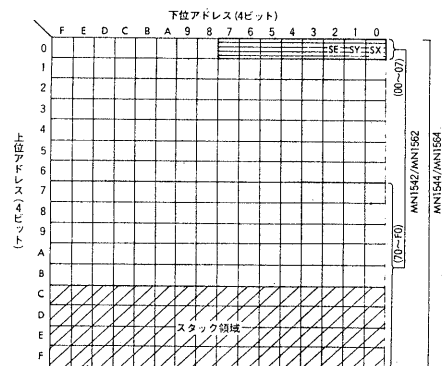
第5図 基本命令サイクル

Fig. 5 Basic instruction cycle.

第2表 機能説明

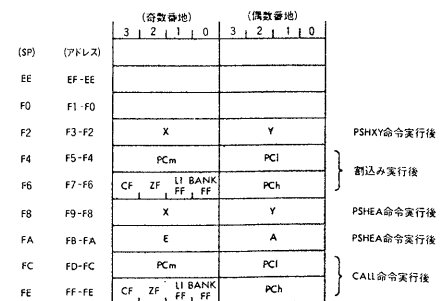
Table 2 Function explanation of symbols.

シンボル	機能説明
ALU	演算論理ユニット (4ビット並列処理)
A	アキュムレータ (4ビット)
E	アキュムレータ拡張レジスタ (4ビット) - Eレジスタ
CF	キャリフラグ / ボローフラグ
ZF	ゼロフラグ
RAM	データメモリ
X	Xレジスタ
Y	レジスタ / ディスクリット出力端子指定レジスタ
SX	Xレジスタ待避レジスタ RAM番地 (0, 0)
SY	Yレジスタ待避レジスタ RAM番地 (0, 1)
SE	Eレジスタ待避レジスタ RAM番地 (0, 2)
PCB/PCm/PCI	プログラムカウンタ (12ビット) PCb: 上位4ビット, PCm: 中位4ビット, PCI: 下位4ビット
IRU/IRL	インストラクションレジスタ (8ビット) IRU: 上位4ビット, IRL: 下位4ビット
SP	スタックポインタ (スタックはRAM領域)
IF	割込み受付フラグ
IF(SIRQ)	最優先割込み受付フラグ
IF(IRQ)	外部割込み受付フラグ
IF(TC)	タイマ / カウンタ割込み受付フラグ
IF(SB)	シフトバッファ割込み受付フラグ
IE	割込みイネーブル / ディスエーブルフラグ
IE(IRQ)	外部割込みイネーブル / ディスエーブルフラグ
IE(TC)	タイマ / カウンタ割込みイネーブル / ディスエーブルフラグ
IE(SB)	シフトバッファ割込みイネーブル / ディスエーブルフラグ
TCU/TCL	タイマ / カウンタ制御レジスタ (8ビット) TCU: 上位4ビット, TCL: 下位4ビット
TBU/TBL	タイマ / カウンタバッファレジスタ (8ビット) TBU: 上位4ビット, TBL: 下位4ビット
BCU/BCL	2進カウンタ (8ビット) BCU: 上位4ビット, BCL: 下位4ビット
SBU/SBL	シフトバッファレジスタ (8ビット) SBU: 上位4ビット, SBL: 下位4ビット
SBF	シフトバッファ・タイミングモードフラグ
LB	入出力ラッチバッファ



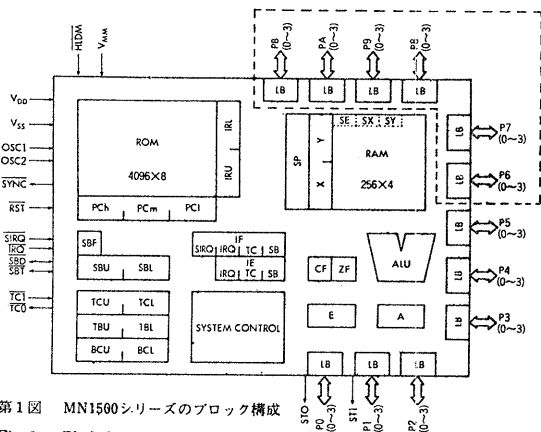
第6図 RAMの構造

Fig. 6 RAM map.



第7図 スタック領域の状態

Fig. 7 Stack area map.



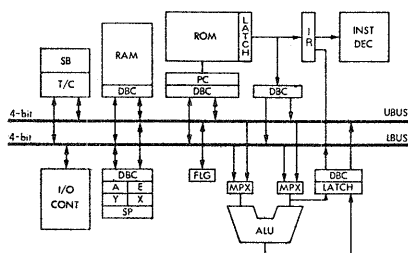
第1図 MN1500シリーズのブロック構成
Fig. 1 Block diagram of MN1500 series.

中に1/2分周器を内蔵しているので、3.58MHz、4.5MHzの安価なテレビ用水晶振動子またはセラミックフィルタを使用することが可能である。

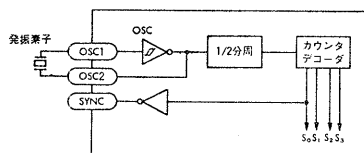
MN1500シリーズのプロセッササイクルは、オーバーラップしない4相クロック S_0, S_1, S_2, S_3 からなり、原発振が4MHzのとき2 μ secである。

第3図にクロック発生部のブロック図を、第4図に基

準クロックタイミング図を、また各クロックにおける動作を第5図の基本命令サイクルに示す。



第2図 内部バスの構成
Fig. 2 Internal bus structure.



第3図 クロック発生部のブロック図
Fig. 3 Circuit diagram of clock generator.

MN1500シリーズの一覧表

Table 1 MN1500 series.

		MN1542	MN1544	MN1562	MN1564	MN1599 (エバリュエータ)	MN1591 (I/Oエクスパンダ)	備 考
パ ッ ケ ー ジ		40ピンDIP プラスチック	40ピンDIP プラスチック	64ピンDIP プラスチック	64ピンDIP プラスチック	64ピンDIP セラミック	40ピンDIP プラスチック	
プ ロ セ ス		NMOS	NMOS	NMOS	NMOS	NMOS	NMOS	
電 源 電 圧		+5V	+5V	+5V	+5V	+5V	+5V	
命 令 サ イ ク ル		2 μ sec	2 μ sec	2 μ sec	2 μ sec	2 μ sec	—	
命 令 数		124	124	124	124	124	—	
インストラクション ROM	内 部	2048 $\times$ 8ビット	4096 $\times$ 8ビット	2048 $\times$ 8ビット	4096 $\times$ 8ビット	—	—	
	外 部	—	—	—	—	8192 $\times$ 8ビット	—	
デ ー タ R A M		152 $\times$ 4ビット	256 $\times$ 4ビット	152 $\times$ 4ビット	256 $\times$ 4ビット	256 $\times$ 4ビット	—	停電保護付
サブルーチンスタック		16レベル	16レベル	16レベル	16レベル	16レベル	—	RAM領域使用
割 込 み		4レベル (外部2レベル内部2レベル)	4レベル (外部2レベル内部2レベル)	4レベル (外部2レベル内部2レベル)	4レベル (外部2レベル内部2レベル)	5レベル (外部3レベル内部2レベル)	—	
タイマ/カウンタ		8ビット	8ビット	8ビット	8ビット	8ビット	—	7ビットプリスケアラ付 プリセット可
直列転送バッファ		8ビット	8ビット	8ビット	8ビット	8ビット	—	クロック 内部・外部選択可
入 出 力	並 列	4ビット $\times$ 6ポート	4ビット $\times$ 6ポート	4ビット $\times$ 12ポート	4ビット $\times$ 12ポート	4ビット $\times$ 6ポート	4ビット $\times$ 6ポート	バイト入出力・ビット入出力可
	出力ストローブ	2	2	2	2	2	2	
	直 列	2	2	2	2	2	—	
	タイマ/カウンタ	2	2	2	2	2	—	
	割 込 み	2	2	2	2	3	—	
イニシャルクリア		あり	あり	あり	あり	あり	—	
クロックジェネレータ		あり	あり	あり	あり	あり	—	発振素子：CR、水晶、 セラミックフィルタ
ソフトウェアサポート		あり	あり	あり	あり	あり	—	
ハードウェアサポート		あり	あり	あり	あり	あり	—	
システムエバリュエータ		MN1599	MN1599	MN1599 (+MN1591 $\times$ 2)	MN1599 (+MN1591 $\times$ 2)	—	—	

3.5 プログラムメモリ

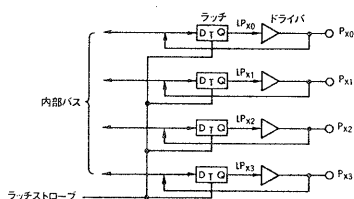
プログラムカウンタは12ビットからなり、下位8ビットはバイナリカウンタ、上位4ビットはレジスタで構成される。すなわち、MN1500 シリーズは1 ページを 256 語とする完全ページマシンであり、最大4096語のアクセスが可能である。

エバリュエータチップ MN1599 は12ビットのプログラムカウンタのほかに、BANK フラグを有しており、最大8192語アクセス可能である。

ROM は1 語 8 ビットで構成される。

3.6 入出力部

MN1500 シリーズの並列入出力部はすべて第8図に示す構造を有している。すなわち、出力ラッチ付の入出力ポートであり、入力動作は出力ラッチを“Hレベル”にセットして実行する、いわゆる疑似双方向回路構成をとっている。



第8図 入出力回路の構成
Fig. 8 Input/output circuit.

機能 \ ポート名	P0,P1	P2	P3	P4,P5
並列および ディスクリット入出力	○	○	○	○
Yレジスタ指定による ディスクリット入出力	○	○	○	
ストロブ出力	○			

MN1562, MN1564

機能 \ ポート名	P0,P1	P2	P3	P4,P5	P6~PB
並列および ディスクリット入出力	○	○	○	○	○
Yレジスタ指定による ディスクリット入出力	○	○	○		
ストロブ出力	○				

MN1599+MN1591×2

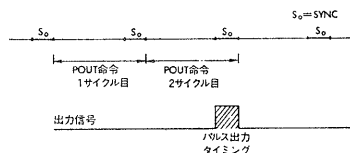
ポート名 機能	P0,P1	P2	P3	P4,P5	MN1599の入出力		
並列および ディスクリット入出力	○	○	MN1591拡張 制御信号入出力 力端子 (EXP端子 ローレベル)				
Yレジスタ指定による ディスクリット入出力	○	○					
ス ト ロ ー プ 出 力	○						
					P6~PB	PD	PE,PF
並列および ディスクリット入出力			○	○	○	○	○
Yレジスタ指定による ディスクリット入出力							
ス ト ロ ー プ 出 力							○

各ビットごとのセット、リセットも可能であり、8ビット並列の入出力も可能である。また、各ポートともパルス出力機能を有している。第9図にパルス出力命令の実行タイミングを示す。

ポート0とポート1はストロブ出力端子を備えている。第10図にストロブ出力タイミングを示す。

また、偶数番号ポートと奇数番号ポートの対による8ビット並列入出力の機能も有している。第3表に入出力ポートの構成を示す。

以上述べたように、MN1500 シリーズの入出力機能は非常に豊富であり、応用時に使いやすい構造になっている。



割込みが受理され、割込み処理プログラムを実行するまでに、割込み受理サイクルとして、2マシンサイクルを必要とする。割込み受理サイクルでは、次の1)～3)を実行する。

- 1) プログラムカウンタおよびプログラムステータスをスタック領域へ退避する。
 - 2) 受理したIFをリセットするとともに、IRQ、TC、SBに関しては該当するIEをリセットし、割込みをディセーブルする。
 - 3) 割込み要求の各処理開始番地をPCにセットする。
- なお、次に述べる命令サイクルでは、割込み受理は禁止される。

- 1) 2サイクル命令の1サイクル目
- 2) EDI 命令の2サイクル目
- 3) 割込み受理ルーチンの2サイクル目
- 4) 割込み開始番地および次の番地の実行サイクル

3.8 タイマカウンタ

タイマカウンタ部のブロック図を第12図に示す。

タイマカウンタは8ビットの制御レジスタ(TC)、2進アップカウンタ(BC)、タイマバッファ(TB)より構成される。タイマカウンタの動作モードは、タイマモード、イベントカウンタモード、パルス幅測定モードがあり、これらの動作モードはWTTC命令によって、タイマカウンタ制御レジスタへ制御モードを書き込むことによって設定できる。このWTTC命令実行により、TCO端子はHレベルとなり、BCがオーバフローすると共に反転する。

タイマカウンタの動作モードの割当てを第13図に示す。

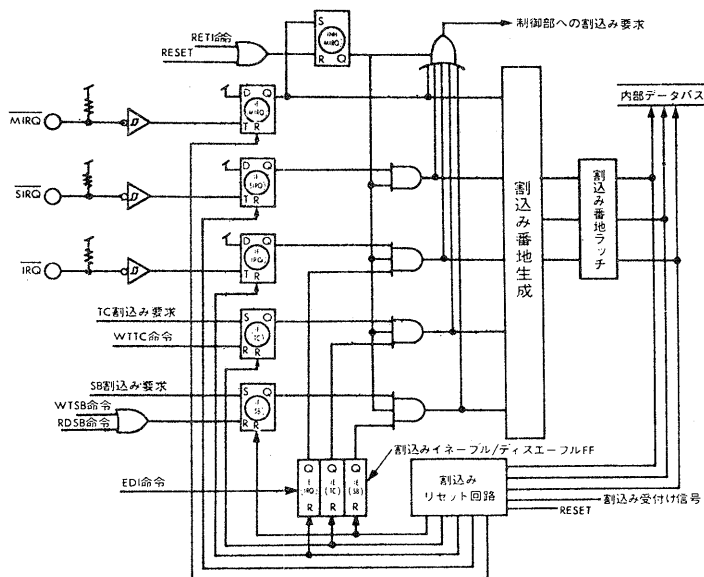
以下、上記の各モードの動作を簡単に説明する。

1) タイマモード

タイマモードでは、設定したい時間値をWTTB命令によりTBにセットする。つぎに、プリスケアラの分周比および動作モードをWTTC命令によりTCにセットしタイマを起動する。同時にWTTC命令により、TBをBCに転送し、初期値をセットし、BCがオーバフローすると、IF(TC)がセットされ、再びTBをBCにセットし、以上の動作を繰り返す。

2) パルス幅測定モード

パルス幅測定モードでは、TCI入力端子への入力信号のパルス幅を測定する。BCがカウントするのは、TCIの非有効レベルに続く、有効レベル間であり、カウント終



第11図 割込み制御回路

Fig. 11 Interrupt-control circuit.

了時点でIF(TC)をセットする。その他の動作は、タイマモードと同様である。

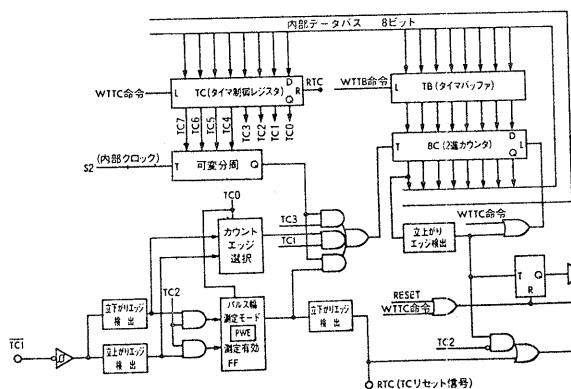
3) イベントカウンタモード

イベントカウンタモードでは、TCI端子に印加されるパルス数をカウントする。カウントの有効エッジは、TCのLSBで指定される。WTTC命令によってイベントカウンタモードの動作が開始すると、TB→BCを実行し、TCIへの印加信号の有効エッジを検出し、BCをインCREMENTする。BCがオーバフローすると、割込み要求信号を発生するとともに、再びTB→BCを実行し、以上の動作を繰り返す。

なお、イベントカウンタモードにおけるカウンタ外部入力の最高応答周期は2マシンサイクルである。

3.9 シフトバッファ部

シフトバッファレジスタ(SB)は、8ビット単位で直列入出力する機能を有する。シフトバッファはSBD端子でデータの入出力を、SBT端子でクロックの入出力を行なう。シフトバッファの構成を第14図に、その動作モードを第5表に示す。

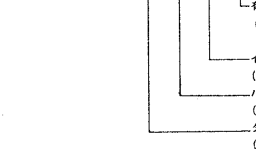


第12図 タイマカウンタの構成
Fig. 12 Timer/counter circuit.

TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
1/128	1/32	1/8	1/2	1/4	1/2	1/4	1/2

←プリスケアラ分周比→

←動作モード→

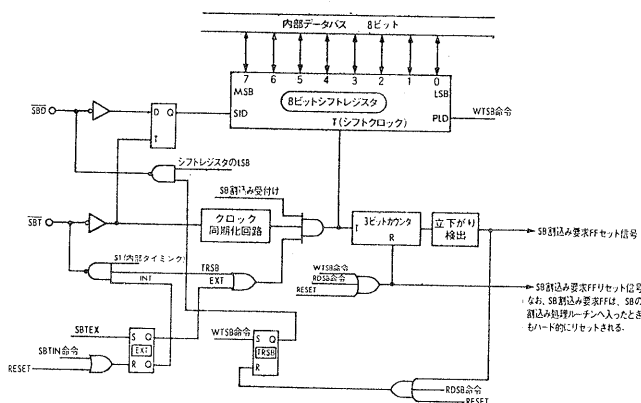


第13図 タイマカウンタの動作モード割当て
Fig. 13 Operation mode of timer/counter.

シフトバッファのデータ転送は送信、受信のいずれとも下位ビットから高位ビットの順に転送され、内部クロックモードにおいては、1マシンサイクルに1ビットずつのシフトクロックが生成される。また、外部クロックモードにおける最高クロック周期も、1マシンサイクルであり、SBTの立下がりがあり有効エッジになる。

3.10 リセット回路

リセット信号入力端子 (RST) をLレベルにするとプログラムカウンタ (PC)、アキュムレータ、Eレジスタ、Xレジスタ、Yレジスタ、CF、ZF、スタックポインタ (SP)、タイマ/カウンタ制御レジスタ (TC)、入出力ポートラッチがイニシャライズされる。また、電源電圧が動作点に達するまでは、内部リセット機能が働き、RSTをLレベルにした場合と同様に各レジスタが初期値にセットされる。



第14図 シフトバッファの構成
Fig. 14 Construction of shift buffer.

4. MN1500シリーズの命令セット

4.1 命令の種類

MN1500 シリーズの命令は、その機能から次の4種に分類される。

- 1) データ転送命令 レジスタ間、レジスタとメモリ間のデータ転送を実行
- 2) 入出力命令 レジスタと入出力ポート間のデータ転送を実行
- 3) 操作命令 算術演算、論理演算などの各種操作を実行。
- 4) コントロール命令 プログラムカウンタ、フラグなどの操作を実行

各命令は、1バイト命令または2バイト命令であり、1バイト命令の実行には、1マシンサイクルあるいは2マシンサイクルを必要とし、2バイト命令の実行にはすべて2マシンサイクルを必要とする。

4.2 命令セット

命令の形式は、命令コード中にオペランドの指定がないもの、命令コード中にオペランドの指定のあるもの、2バイト命令でオペランドの指定のあるものに大別される。その命令の形式を第15図に、MN1500シリーズの全命令セットを第6表に示す。

第5表 シフトバッファの動作モード

Table 5 Shift buffer operation mode.

TRSB _{FF}	EXT _{FF}	動作
L	L	何も動作しない
L	H	外部クロック受信モード
H	L	内部クロック送信モード
H	H	外部クロック送信モード

TRSB_{FF} セット信号: WTSB 命令

リセット信号: シフトカウンタオーバーフロー(8回)

: RDSB 命令

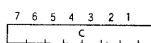
: リセット時

EXT_{FF} セット信号: SBTIN 命令

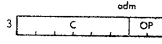
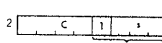
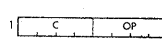
リセット信号: SBTIN 命令

: リセット時

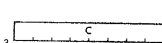
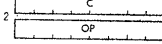
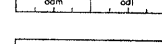
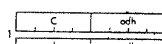
① 命令コード中にオペランドの指定のないもの。



② 命令コード中にオペランドの指定があるもの。



③ 2バイト命令でオペランドの指定のあるもの。



C: 命令操作部
OP: オペランド部
adn: アドレス上位部
adl: アドレス下位部
8+: ショートコールアドレス

第15図 命令の形式

Fig. 15 Instruction format.

5. 結 言

MN1500 シリーズは、複雑な制御を必要とする各種機器への応用を目指して設計した特徴ある構造を有し、新しいプロセス技術、回路技術によって製造される4ビット・1チップマイクロコンピュータである。

内蔵するシフトバッファを使用することにより、疎結合のマルチコンピュータ構造を比較的簡単に構成できるため、従来4ビットマイクロコンピュータの使用が困難であった複雑な機器制御や8ビットマイクロコンピュータを利用していた機器制御等、今後広範囲な応用が期待される。

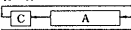
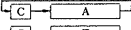
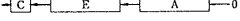
最後に、本マイクロコンピュータは多数の方々の協力によって完成したものであり、関係者各位に深く感謝します。

第6表 MN1500シリーズの命令セット

Table 6 Instruction set.

△論理積(AND) √論理和(OR) ✕排他的論理和(XOR)

ニ モ ニ ッ ク			命令コード (Hex)	影響される フ ラ グ	動 作	動 作 説 明
デ ィ タ 転 送 命 令	L	load	17	ZF	$A \leftarrow M(X, Y)$	メモリM(X, Y)をAにロードする
	LD	**load direct	1F, da	ZF	$A \leftarrow M(da)$	メモリM(da)をAにロードする
	LE	load from E	2F	ZF	$A \leftarrow E$	EをAにロードする
	LX	load from X	44	ZF	$A \leftarrow X$	XをAにロードする
	LY	load from Y	45	ZF	$A \leftarrow Y$	YをAにロードする
	LI	load immediate	Fn		$A \leftarrow n$	即値nをAにロードする
	LICY	*load, increment Y	21	ZF	$A \leftarrow M(X, Y), Y \leftarrow Y + 1$	メモリM(X, Y)をAにロードした後、Yをインクリメントする
	LDCY	*load, decrement Y	25	ZF	$A \leftarrow M(X, Y), Y \leftarrow Y - 1$	メモリM(X, Y)をAにロードした後、Yをデクリメントする
	LEAI	**load EA immediate	76, mn		$E \leftarrow m, A \leftarrow n$	即値mnをEAにロードする
	LXY	load from X, Y	5C		$E \leftarrow X, A \leftarrow Y$	X YをEAにロードする
	ST	store	57		$M(X, Y) \leftarrow A$	AをメモリM(X, Y)にストアする
	STD	**store direct	53, da		$M(da) \leftarrow A$	AをメモリM(da)にストアする
	STE	store to E	56		$E \leftarrow A$	AをEにストアする
	STX	store to X	54		$X \leftarrow A$	AをXにストアする
	STY	store to Y	55		$Y \leftarrow A$	AをYにストアする
	STICY	*store, increment Y	22	ZF	$M(X, Y) \leftarrow A, Y \leftarrow Y + 1$	AをメモリM(X, Y)にストアした後、Yをインクリメントする
	STDCY	*store, decrement Y	26	ZF	$M(X, Y) \leftarrow A, Y \leftarrow Y - 1$	AをメモリM(X, Y)にストアした後、Yをデクリメントする
	LBD	**load byte direct	48, da		$E \leftarrow M(da)$	メモリM(da)をEAにロードする
	LTBD	**store byte direct	49, da		$M(da) \leftarrow E$	EAをメモリM(da)にストアする
	STXY	store to X, Y	5D		$X \leftarrow E, Y \leftarrow A$	EAをXYにストアする
	EX	exchange	47		$A \leftrightarrow M(X, Y)$	AとメモリM(X, Y)を交換する
	EXD	**exchange direct	43, da		$A \leftrightarrow M(da)$	AとメモリM(da)を交換する
	EXE	exchange with E	46		$A \leftrightarrow E$	AとEを交換する
	EXICY	*exchange, increment Y	23	ZF	$A \leftrightarrow M(X, Y), Y \leftarrow Y + 1$	AとメモリM(X, Y)を交換した後、Yをインクリメントする
	EXDCY	*exchange, decrement Y	27	ZF	$A \leftrightarrow M(X, Y), Y \leftarrow Y - 1$	AとメモリM(X, Y)を交換した後、Yをデクリメントする
	STSE	store SE	52		$SE \leftarrow E$	EをSEにストアする
	EXSE	exchange SE	42		$E \leftrightarrow SE$	EとSEを交換する
	LMEI	**load and modify E	33, mn	ZF	$E \leftarrow E \wedge \bar{m} + n$	Eの即値mビット位置をリセットしEに即値nを加える
	STSX	store SX	50		$SX \leftarrow X$	XをSXにストアする
	EXSX	exchange SX	40		$X \leftrightarrow SX$	XとSXを交換する
	LMXI	**load and modify X	32, mn	ZF	$X \leftarrow X \wedge \bar{m} + n$	Xの即値mビット位置をリセットしEに即値nを加える
	LXSXI	**load X, SX immediate	75, mn		$X \leftarrow m, SX \leftarrow n$	即値mをXに、nをSXにロードする
	LYI	load Y immediate	Cn		$Y \leftarrow n$	即値nをYにロードする
	LXYI	**load X, Y immediate	77, mn		$X \leftarrow m, Y \leftarrow n$	即値mをXに、nをYにロードする
	STSY	store SY	51		$SY \leftarrow Y$	YをSYにストアする
	EXSY	exchange SY	41		$Y \leftrightarrow SY$	YとSYを交換する
	PSHEA	push E, A	68		$SP \leftarrow SP - 1, M(SP) \leftarrow EA$	SPをデクリメント後、EAをスタックエリアにストアする
	POPEA	pop E, A	6C		$EA \leftarrow M(SP), SP \leftarrow SP + 1$	スタックエリアをEAにロード後、SPをインクリメントする
	PSHXY	push X, Y	69		$SP \leftarrow SP - 1, M(SP) \leftarrow XY$	SPをデクリメント後、XYをスタックエリアにストアする
	POPXY	pop X, Y	6D		$XY \leftarrow M(SP), SP \leftarrow SP + 1$	スタックエリアをXYにロード後、SPをインクリメントする
	RMD	**reset memory direct	37, da		$M(da) \leftarrow 0$	メモリM(da)のデータを0にする
入 出 力 命 令	WTSB	write SB and start transmission	80		$SB \leftarrow EA$	EAをSBに転送し、シリアル転送を開始する
	RDSB	read SB	84		$EA \leftarrow SB$	SBをEAに転送する
	WTTC	write TC	81		$TC \leftarrow EA$	EAをTCに転送する
	WTTB	write TB	82		$TB \leftarrow EA$	EAをTBに転送する
	RDBC	read BC	85		$EA \leftarrow BC$	BCをEAに転送する
	WTSP	write SP	05		$SP \leftarrow EA$	EAをSPに転送する
	RDSP	read SP	04		$EA \leftarrow SP$	SPをEAに転送する
	RD_TBL	read table	0B		$EA \leftarrow ROM(PCh, EA)$	PCh, EAで指定されるROMをEAに転送する
	IN	**input	73, pn		$A \leftarrow PORT(p) \wedge n$	Pポートを即値nでマスクして入力する
	INEA	**input byte	71, q		$E \leftarrow PORT(q+1), A \leftarrow PORT(q)$	qポートをAに、q+1ポートをEにロードする
	OUT	**output	72, pn		$PORT(p) \leftarrow A \wedge n$	Aを即値nでマスクして出力する
	OUTEA	**output byte	70, q		$PORT(q+1) \leftarrow E, PORT(q) \leftarrow A$	Aをqポートに、Eをq+1ポートに出力する
	ROUT	**reset output	62, pn		$PORT(p) \leftarrow PORT(p) \wedge n$	Pポートの任意のビット(複数可)をリセットする
	SOUT	**set output	66, pn		$PORT(p) \leftarrow PORT(p) \vee n$	Pポートの任意のビット(複数可)をセットする
	ROUTY	reset output by Y	67		$PORT(Y(BP3 \sim 2)), Bn(Y(BP1 \sim 0)) \leftarrow 0$	Yの上位2ビットで指定されるポートの下位2ビットで指定されるビットをリセットする
	SOUTY	set output by Y	63		$PORT(Y(BP3 \sim 2)), Bn(Y(BP1 \sim 0)) \leftarrow 1$	Yの上位2ビットで指定されるポートの下位2ビットで指定されるビットをセットする
	POUT	**pulsed output	74, pn		$PORT(p) \leftarrow PORT(p) \vee n$	Pポートの任意のビット(複数可)の出力データをパルス幅サイクル間反転させる

	ニ モ ニ ッ ク		命令コード (Hex)	影響される フラッグ	動 作	動 作 説 明
操 作 命 令	A	add	10	CF,ZF	$A \leftarrow A + M(X, Y)$	AとメモリM(X,Y)を加算し、結果をAにストアする
	AI	add immediate	Dn	CF,ZF	$A \leftarrow A + n$	Aと即時値nを加算し、結果をAにストアする
	AD	**add direct	18,da	CF,ZF	$A \leftarrow A + M(da)$	AとメモリM(da)を加算し、結果をAにストアする
	AE	add E	28	CF,ZF	$A \leftarrow A + E$	AとEを加算し、結果をAにストアする
	AC	add with carry	11	CF,ZF	$A \leftarrow A + M(X, Y) + CF$	AとメモリM(X,Y)をキャリ付加算し、結果をAにストアする
	ACD	**add direct with carry	19,da	CF,ZF	$A \leftarrow A + M(da) + CF$	AとメモリM(da)をキャリ付加算し、結果をAにストアする
	ACE	add E with carry	29	CF,ZF	$A \leftarrow A + E + CF$	AとEをキャリ付加算し、結果をAにストアする
	AAC	add carry	01	CF,ZF	$A \leftarrow A + CF$	AとCFを加算し、結果をAにストアする
	S	subtract	12	CF,ZF	$A \leftarrow A - M(X, Y)$	AからメモリM(X,Y)を減算し、結果をAにストアする
	SD	**subtract direct	1A,da	CF,ZF	$A \leftarrow A - M(da)$	AからメモリM(da)を減算し、結果をAにストアする
	SE	subtract E	2A	CF,ZF	$A \leftarrow A - E$	AからEを減算し、結果をAにストアする
	SB	subtract with borrow	13	CF,ZF	$A \leftarrow A - M(X, Y) - CF$	AからメモリM(X,Y)をボロー付減算し、結果をAにストアする
	SBD	subtract direct with borrow	1B,da	CF,ZF	$A \leftarrow A - M(da) - CF$	AからメモリM(da)をボロー付減算し、結果をAにストアする
	SBE	subtract E with borrow	2B	CF,ZF	$A \leftarrow A - E - CF$	AからEをボロー付減算し、結果をAにストアする
	O	or	14	ZF	$A \leftarrow A \vee M(X, Y)$	AとメモリM(X,Y)のORをAにストアする
	OD	**or direct	1C,da	ZF	$A \leftarrow A \vee M(da)$	AとメモリM(da)のORをAにストアする
	OE	or E	2C	ZF	$A \leftarrow A \vee E$	AとEのORをAにストアする
	X	exclusive or	15	ZF	$A \leftarrow A \oplus M(X, Y)$	AとメモリM(X,Y)のXORをAにストアする
	XD	**exclusive or direct	1D,da	ZF	$A \leftarrow A \oplus M(da)$	AとメモリM(da)のXORをAにストアする
	XE	exclusive or E	2D	ZF	$A \leftarrow A \oplus E$	AとEのXORをAにストアする
	N	and	16	ZF	$A \leftarrow A \wedge M(X, Y)$	AとメモリM(X,Y)のANDをAにストアする
	ND	**and direct	1E,da	ZF	$A \leftarrow A \wedge M(da)$	AとメモリM(da)のANDをAにストアする
	NE	and E	2E	ZF	$A \leftarrow A \wedge E$	AとEのANDをAにストアする
	C	compare	0F	CF,ZF	$A - M(X, Y)$	AとメモリM(X,Y)を比較する
	CI	compare immediate	En	CF,ZF	$A - n$	Aと即時値nを比較する
	CD	**compare direct	0D,da	CF,ZF	$A - M(da)$	AとメモリM(da)を比較する
	CE	compare E	0E	CF,ZF	$A - E$	AとEを比較する
	CEAI	**compare byte immediate	0C,mn	CF,ZF	$E A - mn$	E Aと即時値mnを比較する
	DAA	decimal adjust add	06	CF,ZF	$A \leftarrow A + 6 \text{ if } A \geq 10 \text{ or } CF = 1$	Aが10以上またはCFがセットされているとき、6を加えCFをセットする
	DAS	decimal adjust subtract	0A	CF,ZF	$A \leftarrow A + 10 \text{ if } A \geq 10 \text{ or } CF = 1$	Aが10以上またはCFがセットされているとき、10を加えCFをセットする
	ICM	increment memory	60	CF,ZF	$M(X, Y) \leftarrow M(X, Y) + 1$	メモリM(X,Y)をインクリメントする
	ICMD	**increment memory direct	61,da	CF,ZF	$M(da) \leftarrow M(da) + 1$	メモリM(da)をインクリメントする
	DCM	decrement memory	64	CF,ZF	$M(X, Y) \leftarrow M(X, Y) - 1$	メモリM(X,Y)をデクリメントする
	DCMD	**decrement memory direct	65,da	CF,ZF	$M(da) \leftarrow M(da) - 1$	メモリM(da)をデクリメントする
	ICEA	**increment byte	59	CF	$E A \leftarrow E A + 1$	E Aをインクリメントする
	DCEA	*decrement byte	5A	CF	$E A \leftarrow E A - 1$	E Aをデクリメントする
	ICY	increment Y	20	ZF	$Y \leftarrow Y + 1$	Yをインクリメントする
	DCY	decrement Y	24	ZF	$Y \leftarrow Y - 1$	Yをデクリメントする
	CPL	complement	02	ZF	$A \leftarrow \bar{A}$	Aの補数を求める
	ROL	rotate left	08	CF,ZF		Aを左に回転する。MBSをCFに、CFをLSBにストアする
	ROR	rotate right	09	CF,ZF		Aを右に回転する。CFをMSBに、LSBをCFにストアする
	SLEA	*shift left byte	58	CF		E Aのデータを左にシフトする。LSBは0になる
	RBMD	**reset bit memory direct	3(8+b),da	ZF	$M(da, b) \leftarrow 0$	メモリM(da)のbによって指定されたビットをリセットする
	SBMD	**set bit memory direct	3(C+b),da	ZF	$M(da, b) \leftarrow 1$	メモリM(da)のbによって指定されたビットをセットする
コ ン ト ロ ル 命 令	NOP	no operation	00			ノーオペレーション
	WI	wait for interrupt	4A			割込みを待機する
	RC	reset carry	C3	CF	$CF \leftarrow 0$	CFをリセットする
	SC	set carry	07	CF	$CF \leftarrow 1$	CFをセットする
	EDI	**enable/disable interrupt.	5B,mn		$I E \leftarrow I E \wedge m \vee n$	3種の割込のイネーブル/ディスエーブルフラグおよびバンクフラグを指定する
	SBTIN	SB timing internal	30		$SB S \leftarrow 0$	シフトバッファのシフトクロックに内部タイミングを使う
	SBTEX	SB timing external	31		$SB F \leftarrow 1$	シフトバッファのシフトクロックに外部タイミングを使う
	JMP	**jump	Ah,ml		$PC \leftarrow hml$	アドレスhmlに無条件ジャンプする
	CALL	**call long	9h,ml		$SP \leftarrow SP - 2, M(SP) \leftarrow PC, PC \leftarrow hml$	SPをデクリメントし、アドレスhmlにあるサブルーチンをコールする
	CALS	*call short	8(8+s)		$SP \leftarrow SP - 2, M(SP) \leftarrow PC, PC \leftarrow l \leftarrow 0, PCm \leftarrow 8 + S$	0ページ(8+S)0にあるサブルーチンにコールする
	JMPEA	jump by EA	36		$PCm \leftarrow E, PCl \leftarrow A$	同一ページ内でE Aで指定される番地へジャンプする
	RET	*return	34		$PC \leftarrow M(SP), SP \leftarrow SP + 2$	サブルーチンよりリターンする
	RETI	*return from interrupt	35	CF,ZF	$CF/ZF/PC \leftarrow M(SP), SP \leftarrow SP + 2$	インタラプト処理ルーチンよりリターンする
	JBZ	**jump if bit zero	7(8+b),ml		$PCm/l \leftarrow ml \text{ if } (b) = 0$	Aのbで指定されるビットが0のとき、同一ページ内のmlにジャンプする
	JBNZ	**jump if bit non-zero	7(C+b),ml		$PCm/l \leftarrow ml \text{ if } A(b) = 1$	Aのbで指定されるビットが1のとき、同一ページ内のmlにジャンプする
	JZ	**jump if zero	6E,ml		$PCm/l \leftarrow ml \text{ if } ZF = 1$	ZFがセットされていれば同一ページ内のmlにジャンプする
	JNZ	**jump if non-zero	6A,ml		$PCm/l \leftarrow ml \text{ if } ZF = 0$	ZFがリセットされていれば同一ページ内のmlにジャンプする
	JC	**jump if carry	6F,ml		$PCm/l \leftarrow ml \text{ if } CF = 1$	CFがセットされていれば同一ページ内のmlにジャンプする
	JNC	**jump if non-carry	6B,ml		$PCm/l \leftarrow ml \text{ if } CF = 0$	CFがリセットされていれば同一ページ内のmlにジャンプする
	CYIJ	**compare Y and jump	Bn,ml	ZF	$PCm/l \leftarrow ml \text{ if } Y \neq n$	Yと即時値nを比較し、一致しなければmlにジャンプする
	ICEJ	**increment E and jump	5F,ml		$E \leftarrow E + 1, PCm/l \leftarrow ml \text{ if } E \neq 0$	Eをインクリメントし、0でなければ同一ページ内のmlにジャンプする
	ICYJ	**increment Y and jump	5E,ml		$Y \leftarrow Y + 1, PCm/l \leftarrow ml \text{ if } Y \neq 0$	Yをインクリメントし、0でなければ同一ページ内のmlにジャンプする

*1バイト2サイクル命令 (ROM1バイト使用 実行時間 4μsec) (標準) **2バイト2サイクル命令 (ROM2バイト使用 実行時間 4μsec) (標準) ***付きな命令はROM1バイト使用 実行時間 2μsec (標準))