シングルチップマイクロコンピュータ8048

知名定清（プロダクトマーケティング，インテルジャパン）

8048シングルチップマイクロコンピュータは、最新のNMOSテクノロジーを駆使して、ディジタル処理システムに必要とされる機能の全てを1つのシリコンチップに格納したものである。

限られたシリコンチップ内ではあるが、8ビット並列処理のプロセッサが採用された。シングルチップという限界からくる機能上の制約はあっても、汎用性を保持することが大きな設計目標の一つであったからである。

また、シングルチップ内の機能でおさまりきれないアプリケーションのためには、メモリやI/Oの拡張が容易であることも設計目標の一つであった。特に、I/Oラインの数は、40ピンという、ピン数の制限から24本に定められているため、アプリケーションが高度のものになると真先に拡張の必要性が生じてくるものと思われる。このために、I/Oラインの拡張のための専用のファミリーコンポーネントが用意された。

8048の概要

1、8ビットマイクロコンピュータである。
2、1Kバイトのプログラムメモリ（マスクROM）が内蔵されている。
3、64バイトのデータメモリが内蔵されている。
4、8ビットのポートが3ポート内蔵されている（計24ライン）。
5、プログラマブルなインタバルタイマ（イベントカウンタとしても使える）が内蔵されている。
6、クロック発生回路が内蔵されており、外部には、クリスタルかRC回路を接続すればよい。
7、割込み機能がある。

8048の性能

1、5ボルト単一電源で動作する。
2、40ピンDIPにパッケージされている。
3、90余の命令があり、全て1バイトか2バイトの命令である。
4、命令の実行時間は、1バイト命令で2.5μs，2バイト命令では5μsである。
5、シングルステップで命令の実行が可能。
6、8つのワーキングレジスタが2バンクあり、命令で切替えが可能。
7、8レベルのスタックレジスタがある。
8、パワーダウンモードでは、低電力でデータメモリの内容を保持することができる。

8748

8048 に内蔵されているマスクROMを起す以前に、デバッグの段階とか、あるいは少量で少しずつプログラムの異った物を数種類にわたってつくらなければならないような場合、8048 のEPROM版があれば非常に便利である。8748は、そのためのもので、8048と電気的、機械的に全くピンコンパチブルである。唯一つだけ異るのは、8748には、前節8項のパワーダウンモードがないことである。8748 の存在は、8048のアプリケーションシステムの開発を非常に容易にする。

8035

8035は、8048と全く同じであるが、内蔵の1Kバイトのプログラムメモリのないところだけが異る。これは、プログラムメモリが1Kバイトを越える場合、拡張用のメモリを外部に接続することになるが、外部メモリの容量によっては、内部メモリのない方が、容量を合わせやすいことがある。そのために8035が用意された。

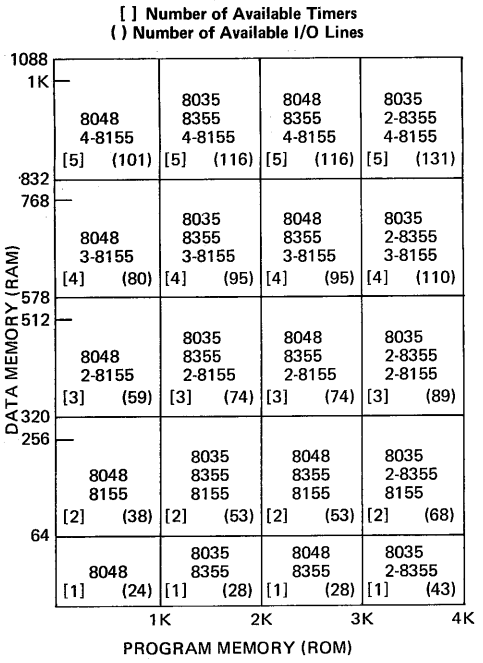

図1、拡張された8048システム

8048システムの拡張

図1は、8048(あるいは8035)システムの拡張を、横軸にプログラムメモリ、縦軸にデータメモリのサイズを取って示したものである。ここで示されている8355と8155は、8048のファミリーで、それぞれプログラムメモリと、データメモリの拡張のために使われる。

8355 の概要

1、2Kバイトのマスク ROM
2、16本のプログラマブルな I/O ラインを内蔵
3、5ボルト単一電源で動作
4、ROM のアクセスタイムは 400 ns
5、8355と全く電気的、機械的にピンコンパチブルな EPROM 版があり、これを8755と呼ぶ。

8155 の概要

1、256バイトのスタティック RAM
2、22本のプログラマブルな I/O ラインを内蔵
3、14ビットのプログラマブルなインタバルタイマを内蔵
4、5ボルト単一電源で動作
5、RAMのアクセスタイムは 400 ns

図2、8355 および 8155 でメモリの拡張を行った 8048（8748/8035）システム

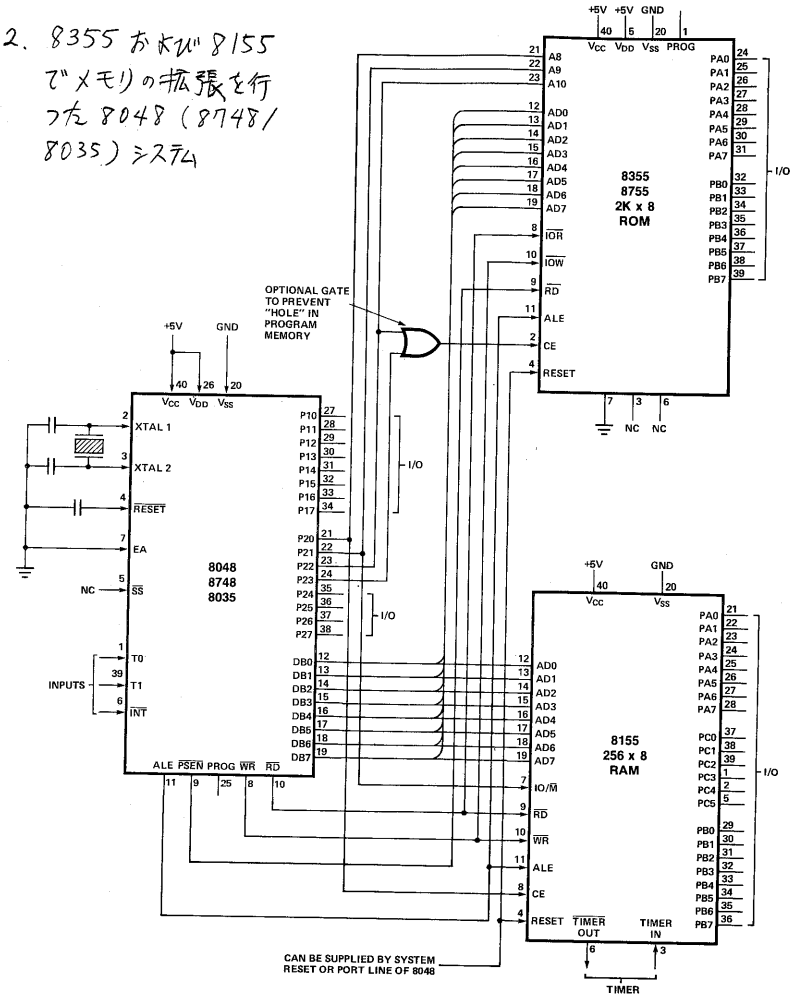

図2は、8048（あるいはその代りに8748, 8035でもよい）と8355（あるいは8755）と8155からなる3チップの拡張システムを示したものである。これら3チップの接続のためにインターフェース回路は全然必要としない。

8048のアーキテクチャー

図3は、8048のブロック図を示す。大きく分けて、プロセッサ（CPU）部、プログラムメモリ、データメモリ，I/Oポートから成っている。

CPUはさらに、(1) レジスタ群（アキュムレータ、プログラムカウンタ、命令レジスタ、ワーキングレジスタバンク，スタックレジスタ），(2) Arithmetic/Logic Unit (ALU) および(3)コントロールロジックから成っている。

アキュムレータ

アキュムレータは、通常、ALUによって処理されるべきデータを保持し、しばしば処理された後のデータの行先となる。

命令は、しばしば、ワーキングレジスタの一つを指定し、このレジスタの内容とアキュムレータの内容、あるいは、このレジスタの内容が指定するデータメモ

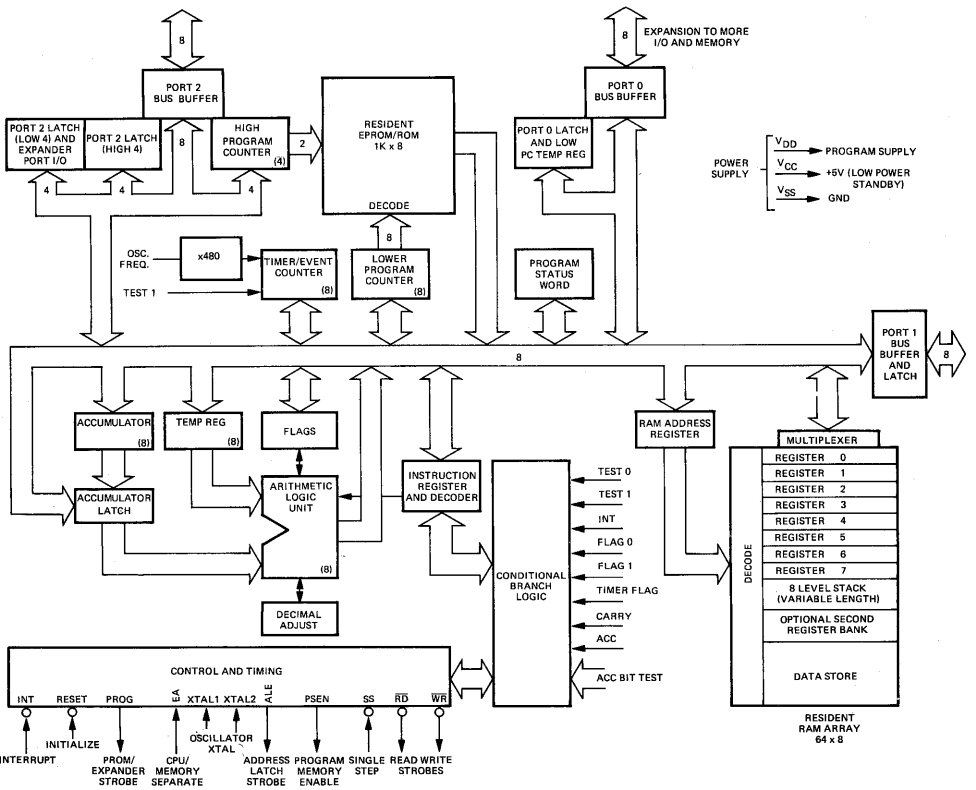

図3、8048のブロック図

りとアキュムレータの間の演算が行われる。

ALU

8048のALUは、1つあるいは2つのソースから8ビットのデータを受け、8ビットの結果を命令デコーダの制御の下に発生する。ALUの実行する機能は次のようなものである。

Add With/Without Carry
AND, OR, XOR
Increment/Decrement
Bit Complement
Rotate Left/Right
Swap Nibles
BCD Decimal Adjust

もしも演算結果が8ビット以上の値を発生する場合は、キャリーフラッグが、PSW(Program Status Word)の中にセットされる。

ポート1とポート2

ポート1とポート2はそれぞれ8ビットのカがあり、両者は全く同じ特性を持っている。出力ポートとして使われる場合は、データがラッチされて、次のデータが書きこまれるまでは、そのまま保持される。入出ポートとして使われる場合は、データがラッチされないので、入力データは、Input命令によってアキュムレータに読みこまれるまで保持されていなければならない。

図4に、ポート1および2の回路構成を示す。これから分るように、あるビットを入力として使う場合には、出力ラッチに予め、"1"をラッチしておかなくてはならない。そこでこのポートを準相方向性と呼ぶ。

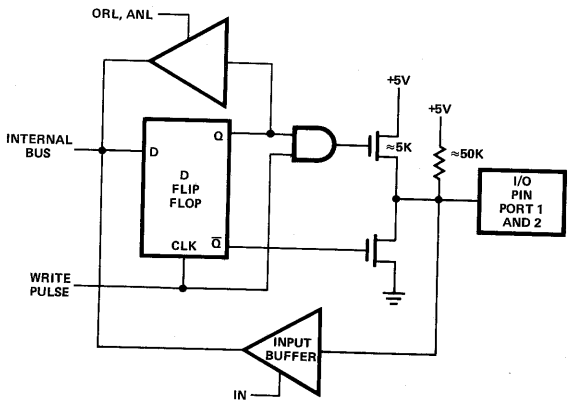

図4、ポート1/2の回路構成

バス（ポート0）

バスも8ビットのポートであるが、この回路は完全な相互向性になっている。相互向性で使わない時は、出力データはラッチされるが、入力データはラッチされない。また、バス上のビット毎に入力と出力を混ぜて使うことはできない。

相互向性のポートとしては、データの読み書きはMOVX命令を用いて行われる。ポートへのWriteではWR信号が発生され、出力データはWRの後縁で有効である。ポートへのReadでは、RD信号が発生され、読みこみデータは、RDの後縁で有効でなくてはならない。それ以外の時は、バスはハイインピーダンスの状態にある。

8048の命令

図15. **INSTRUCTION SET**

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A,R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{\text{INT}}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Movers	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENTO CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

8048 の応用例

図6および図7に 8048 の応用例を示す。

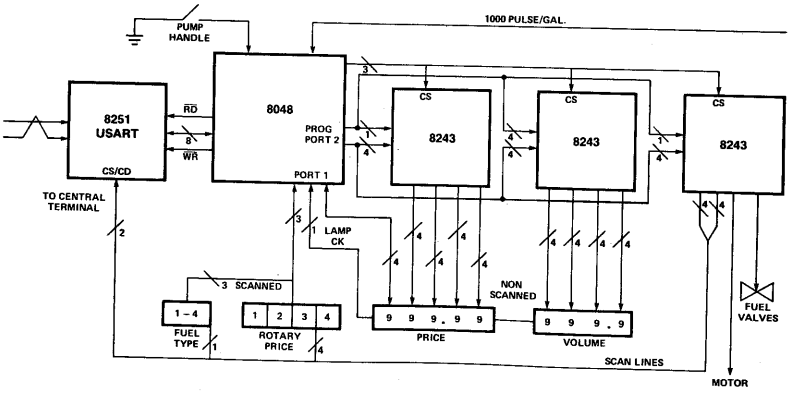

図6. 8048 のガソリンポンプコントローラへの応用例

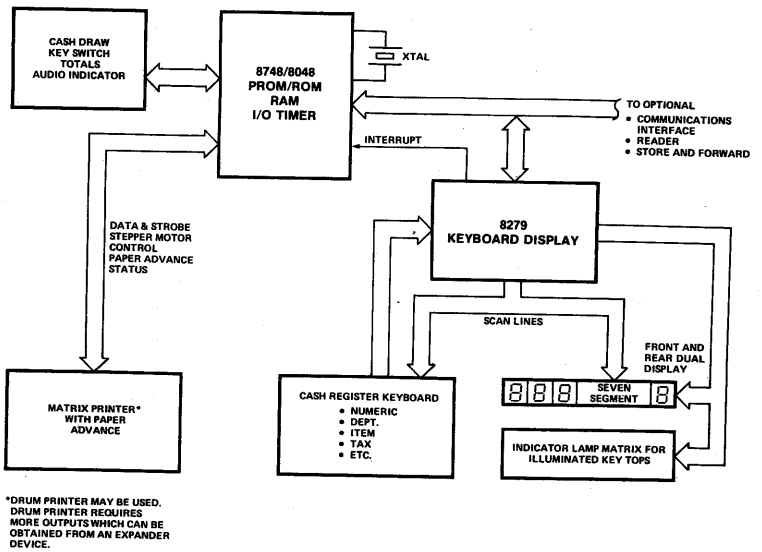

図7. 8048 のローコスト POS への応用例