

I nuovi controller PIC

Come in ogni settore dell'elettronica, anche nel campo dei microcontroller ogni giorno nascono delle novità. Ecco i nuovi componenti Microchip delle famiglie 12, 16 e 17xxx

Paolo Sbrana - 1ª parte

Quando nell'anno 1994 **Progetto** ha iniziato a parlare di microcontroller PIC, pochissimi in Italia li conoscevano, ad eccezione delle sole ditte ad alto livello tecnologico.

Abbiamo così approntato un primo corso sulla programmazione di questi microcontroller

RISC e successivamente si sono viste tutte le periferiche nate piano piano intorno alla CPU.

Ricordiamo che quando abbiamo trattato per la prima volta questi componenti, i tipi disponibili sul mercato italiano erano molto limitati: si potevano reperire i quattro PIC della famiglia base (PIC16C54,

PIC16C55, PIC16C56 e PIC16C57), l'unico PIC in tecnologia EEPROM (PIC16C84), l'unico PIC con un convertitore a bordo (PIC16-C71) e l'unico PIC della famiglia 17, ovvero il PIC17C42. Nel momento in cui scriviamo i tipi di chip disponibili sono già oltre 40, ognuno con caratteristiche proprie.

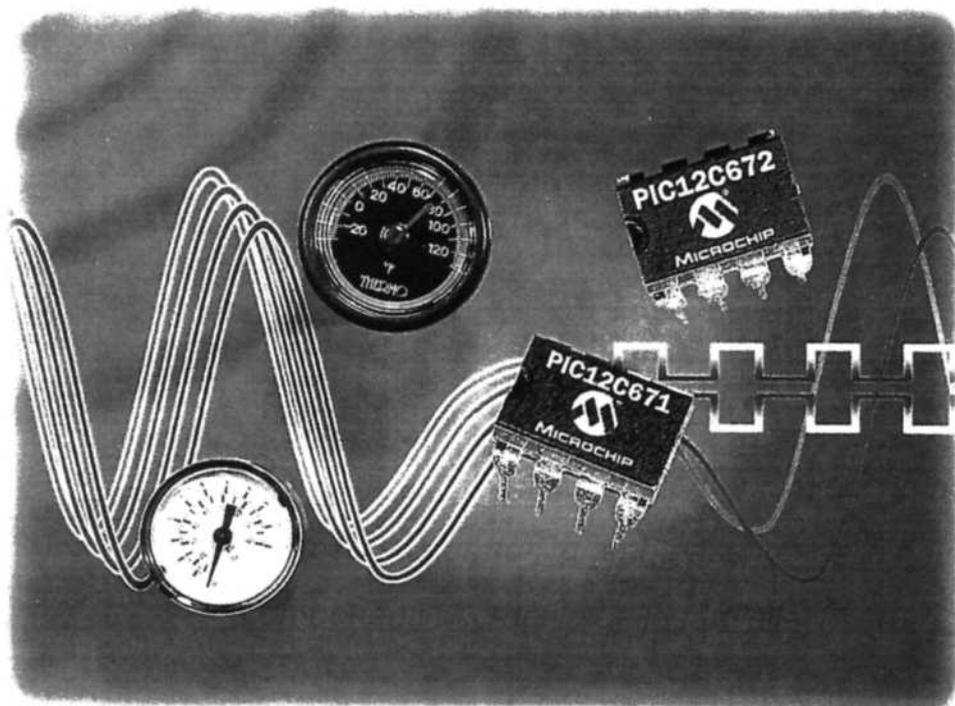
Nostra intenzione è di far conoscere ai lettori i modelli più recenti, analizzando attentamente famiglia per famiglia.

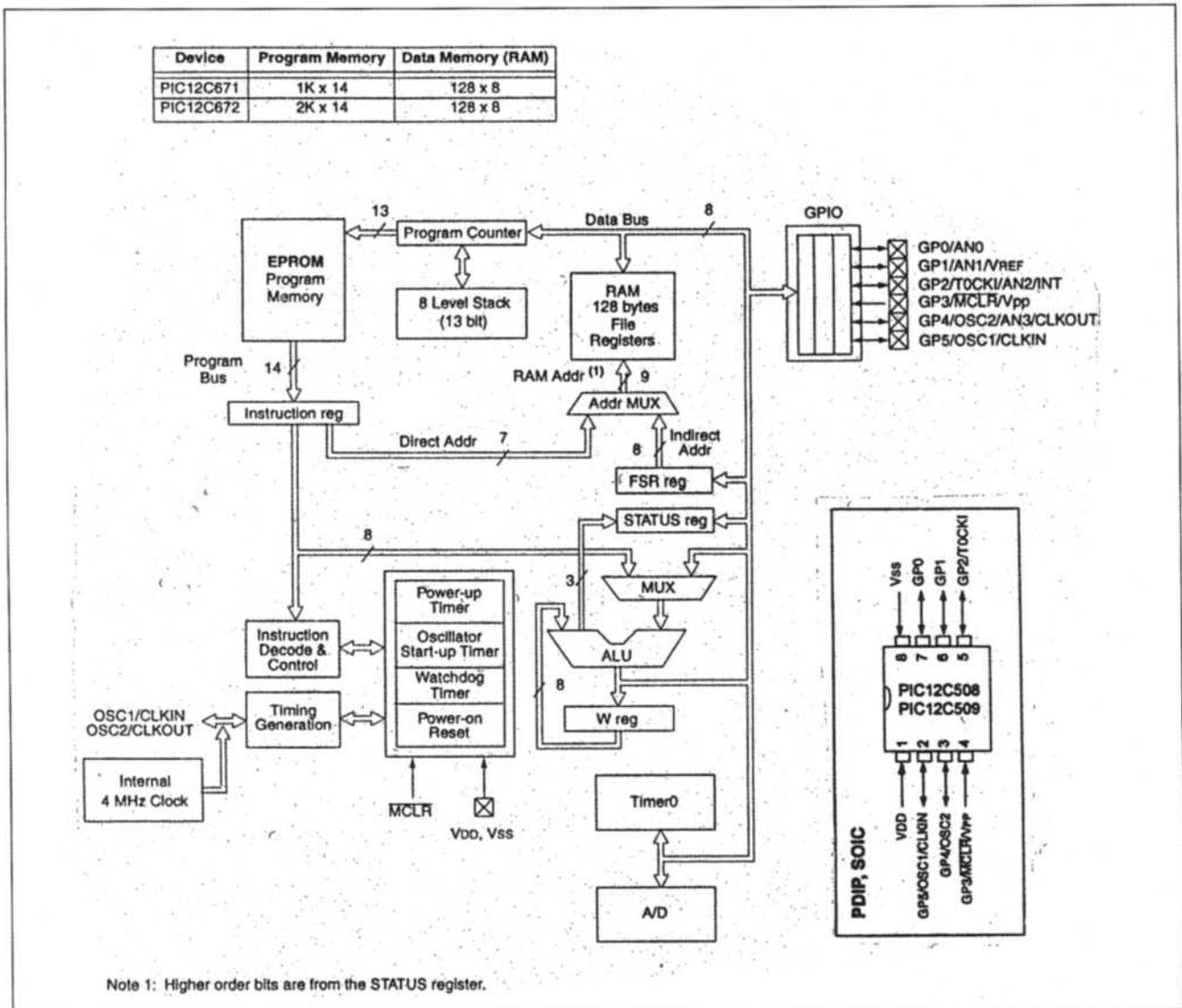
Questo mese inizieremo con i più recenti ed al tempo stesso i più piccoli PIC: la famiglia 12Cxx ad otto pin.

Con questa famiglia Microchip Technology entra in un mercato dove ancora pochi sono riusciti ad ottenere ottimi risultati, forse perchè gli otto pin hanno poche linee di I/O e questo può rivelarsi un handicap.

Le famiglie dei PIC12Cxxx

Iniziamo la nostra rassegna con i microcontroller più piccoli che possiamo trovare sul





mercato: le due famiglie 12C50x e 12C67x entrambe ad otto pin e disponibili perfino in versione SMD (con package detto SOIC per gli addetti ai lavori). Queste due famiglie hanno molto in comune, anche se vedremo che esistono delle differenze sostanziali per quanto riguarda le periferiche possedute e gli interrupt.

Nella tabella 1 valutiamo le caratteristiche principali: la memoria di programma spazia da 512 a 2048 locazioni, la RAM da 25 a 128 registri, le sorgenti di interrupt da 0 a 4, la tensione di alimentazione da 2,5 a 5,5 volt e la frequenza

di funzionamento da 0 a 10 MHz. Analizziamo quindi in dettaglio il diagramma a blocchi riportato in Figura 1, che vale sia per i 12C50x che per i 12C67x.

La struttura è quella tipica dei PIC delle famiglie mid-range, ovvero dei 16C6x e 16C7x, ma ovviamente le linee di I/O sono in numero minore (6).

Un modulo interno che però non è presente in alcun altro PIC è l'oscillatore interno a 4 MHz. Per non sprecare pin infatti, i progettisti hanno deciso di inserirne uno da circa 4 MHz all'interno, anche se poi è possibile applicare egualmente un

Figura 1.
Diagramma
a blocchi dei
PIC12Cxxx

oscillatore esterno per particolarissime applicazioni.

Vedremo inoltre che la frequenza di tale oscillatore è "regolabile" via software agendo su un ben preciso registro dedicato.

Si capisce subito la comodità di una scelta simile: per prima cosa si ha un componente in meno (o tre se si considerano anche i due condensatori) sul circuito con vantaggi economici e di dimensionamento, poi si liberano due pin altrimenti dedicati solo a questa ben definita funzione (che in un chip ad otto pin ne lascerebbero quindi solo 4 liberi).

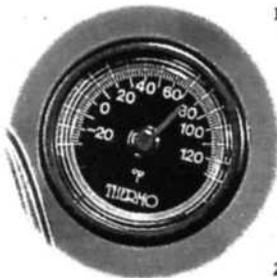
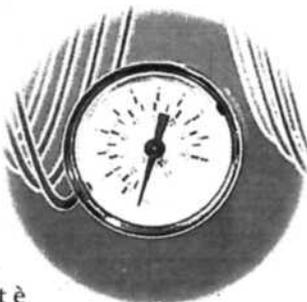
I livelli di stack sono rimasti due per la famiglia 12C-50x mentre sono aumentati a otto per la famiglia 12C67x.

Il modulo di conversione A/D ad 8 bit è presente solo nei 12C67x. Per il resto tutti gli altri moduli sono identici agli altri controller Microchip.

Ciò che cambia nei quattro modelli attualmente disponibili sono la quantità della memoria di programma (ROM) e quella dei registri RAM.

In Figura 2-a possiamo vedere la suddivisione della ROM nei 12C50x ed in Figura 2-b quella relativa ai 12C67x.

Per quanto riguarda la RAM invece, in Figura 3 vediamo le tre diverse pezzature rispettivamente per i 12C508, 12C509 e 12C67x.



Il primo tipo rientra in un banco solo, mentre i rimanenti necessitano di un secondo banco. Come si vede dalla Figura, i progettisti hanno fatto in modo di unificare gli indirizzi comuni di tutti i nuovi controller, sacrificando così locazioni che

non possono essere mai utilizzate (ad esempio come la serie di registri che nei 12C-67x va dall'indirizzo 0xd all'indirizzo 0x1d).

di stato dei 12C50x (Figura 4-a) con quello dei 12C67x (Figura 4-b).

I bit dallo 0 al 4 hanno lo stesso significato ovvero: il bit 0 è un flag che indica se dopo un'operazione si è verificato un riporto.

Il bit 1 ha lo stesso significato del bit 0 ma riferito a soli quattro bit.

Il bit 2 indica se dopo un'operazione si è ottenuto risultato zero o diverso da zero.

Il bit 3 serve per capire se il controller si è resettato dopo una caduta di alimentazione oppure dopo un'istruzione di sleep.

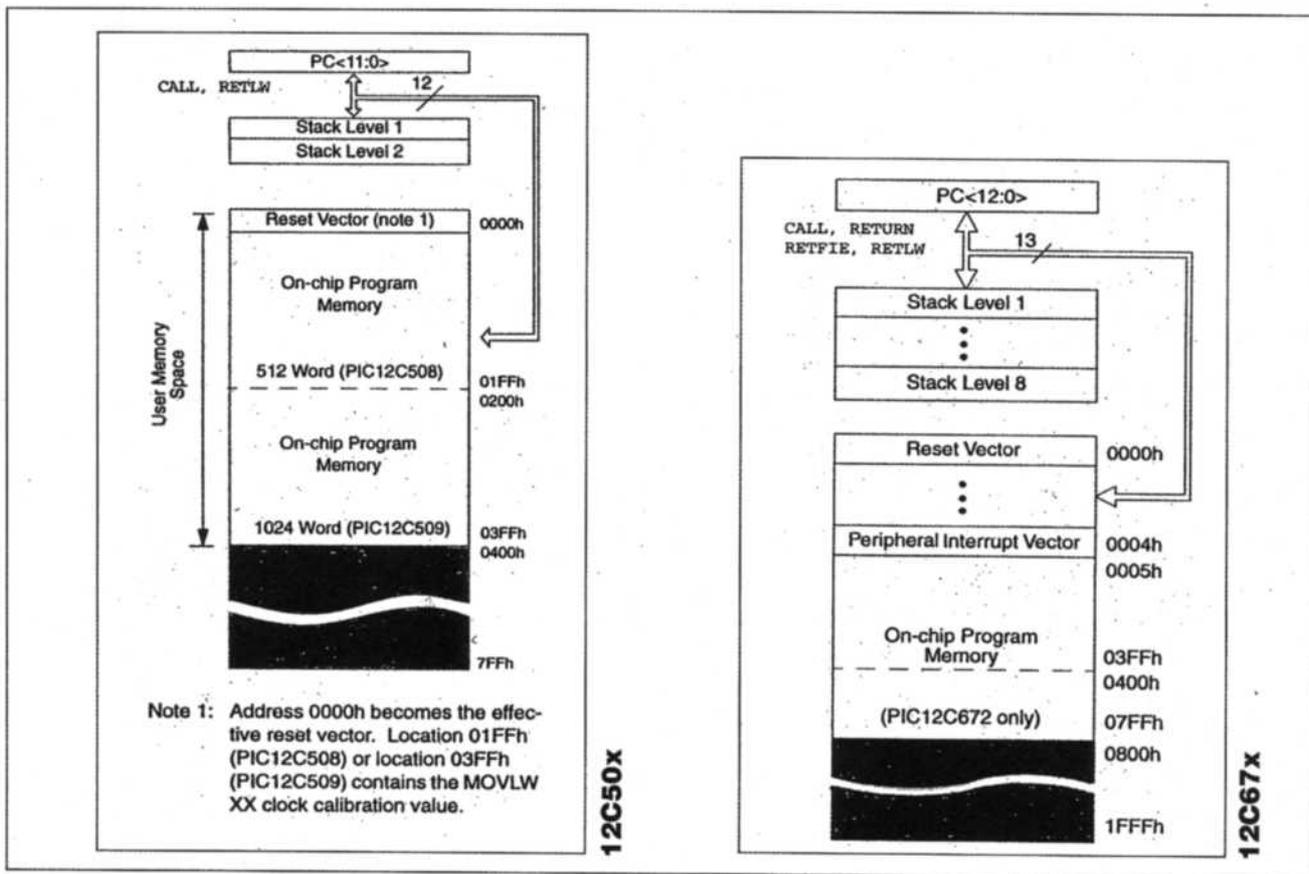
Il bit 4 invece indica se c'è stato il Time-out dall'watchdog interno.

Il bit 5 del registro di stato dei 12C50x serve per selezionare la pagina ROM desiderata (ovviamente solo per i 12C509) mentre il bit 6 non ha valore ed il bit

Figura 2a e2b.
Mapa della memoria ROM nelle famiglie 12C67x e 12C50x

Lo status register

Proprio per la diversità di implementazione, anche i registri interni hanno a volte delle piccole varianti, come si può vedere confrontando il registro



R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **IRP**: Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)
The IRP bit is reserved, always maintain this bit clear.

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h - 1FFh)
10 = Bank 2 (100h - 17Fh)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. The RP1 bit is reserved, always maintain this bit clear.

bit 4: **TO**: Time-out bit
1 = After power-up, CLRWD instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3: **PD**: Power-down bit
1 = After power-up or by the CLRWD instruction
0 = By execution of the SLEEP instruction

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)(for borrow the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred
Note: For borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

attenzione ai chip finestrati, poiché tale porzione di memoria viene cancellata ogni volta che si impiegano lampade a raggi ultravioletti per cancellare tutto il chip. In questi casi, conviene memorizzare tale valore in un file. L'oscillatore interno ha a disposizione 16 livelli di aggiustamento dalla frequenza propria principale, che essendo in realizzata con una rete RC può variare in base a diversi fattori sia in più che in meno rispetto ai 4 MHz prefissati.

Questi 16 livelli sono suddivisi in livelli di "aumento" della frequenza prodotta ed in livelli di "diminuzione" della frequenza prodotta.

A questo punto possiamo vedere il significato del registro OSCCAL: il bit 0 ed il bit 1 non vengono utilizzati.

Con i bit 2 e 3 diciamo all'oscillatore se deve aggiustare la sua frequenza aumentandola o diminuendola.

Con i quattro bit dal 4 al 7 si seleziona di quanto ci si deve discostare dalla frequenza di oscillazione principale. Con i 4 bit sono quindi disponibili 2 elevato alla quarta cioè 16 livelli (o step) di aggiustamento.

I registri delle interruzioni

Gli interrupt sono disponibili solamente per la famiglia 12C-67x ed in Figura 7 troviamo il registro INTCN che li gestisce completamente.

Figura 4-b.
Il registro di stato nei 12C67x



Il bit 0 è un flag che viene settato quando giunge un interrupt dal cambio di stato di uno dei tre pin GP0, GP1 e GP3. Il bit 1 è il flag che segnala l'interrupt dal pin GP2.

Il bit 2 invece segnala l'interrupt proveniente dal timer0. I bit 3, 4 e 5 sono i rispettivi bit di abilitazione al funzionamento delle sorgenti di interrupt appena viste.

Il bit 6 abilita gli interrupt da periferica (che in questo chip è rappresentata dal solo modulo convertitore A/D). Il bit 7 abilita o meno tutti i tipi di interrupt. La gestione dell'interrupt da conversione, viene effettuata attraverso i due registri dedicati PIR1 e PIE, visibili rispettivamente in Figura 8-a e Figura 8-b. Nel primo al bit 6 si ha il flag di avvenuto interrupt,

	W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
	GPWU	GPPU	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7:	6	5	4	3	2	1	bit0	

bit 7: **GPWU:** Enable wake-up on pin change (GP0, GP1, GP3)
1 = Disabled
0 = Enabled

bit 6: **GPPU:** Enable weak pull-ups (GP0, GP1, GP3)
1 = Disabled
0 = Enabled

bit 5: **T0CS:** Timer0 clock source select bit
1 = Transition on T0CKI pin
0 = Transition on internal instruction cycle clock, Fosc/4

bit 4: **T0SE:** Timer0 source edge select bit
1 = Increment on high to low transition on the T0CKI pin
0 = Increment on low to high transition on the T0CKI pin

bit 3: **PSA:** Prescaler assignment bit
1 = Prescaler assigned to the WDT
0 = Prescaler assigned to Timer0

bit 2-0: **PS2:PS0:** Prescaler rate select bits

Bit Value	Timer0 Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

W = Writable bit
U = Unimplemented bit
- n = Value at POR reset
Reference Table 4-1 for other resets.

mentre nel secondo, sempre al bit 6, si ha l'abilitazione o meno a quel tipo interruzione.

Tutti gli altri bit non vengono impiegati.

I registri del convertitore A/D

Passiamo adesso alla periferica del convertitore. Per capire a che cosa servono i bit dei due registri dedicati, vediamo prima in Figura 9 il diagramma a blocchi del convertitore e dei suoi possibili collegamenti con il mondo esterno.

Come si capisce dal grafico, al modulo arriva soltanto un segnale da convertire, gestito su 4 canali con un multiplex interno. Allo stesso modo, la tensione di riferimento Vref necessaria allo stadio convertitore, può sia essere presa direttamente dal positivo di alimentazione inter-

Figura 5.
Il registro delle opzioni nei 12Cxxx

no, sia prelevata esternamente dal pin GP1.

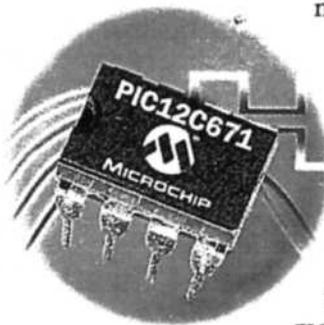
Se dopo questa premessa si passa alla Figura 10, si riesce a capire meglio il significato dei bit del registro ADCON0: il bit 0 serve per attivare o meno il modulo convertitore. Ovviamente, quando tale modulo

non serve, è inutile che assorba corrente restando acceso.

Il bit 1 è riservato. Il bit 2 fa partire la conversione e ci avvisa quando è terminata (unitamente all'eventuale interrupt se abilitato).

I due bit 3 e 4 consentono di selezionare il canale prescelto sul multiplex prima visto.

Chiaramente abbiamo due bit per consentire quattro possibili ingressi diversi.



Il bit 5 è riservato. Con i due bit 6 e 7 invece è possibile selezionare l'oscillatore relativo alla periferica del convertitore: si può avere la frequenza del clock divisa per 2, per 8 o per 32, oppure ottenere una frequenza da un oscillatore interno RC dedicato solo a questo compito.

Tale oscillatore viene spento quando il bit 0 di attivazione viene azzerato e viene riacceso quando tale bit viene settato.

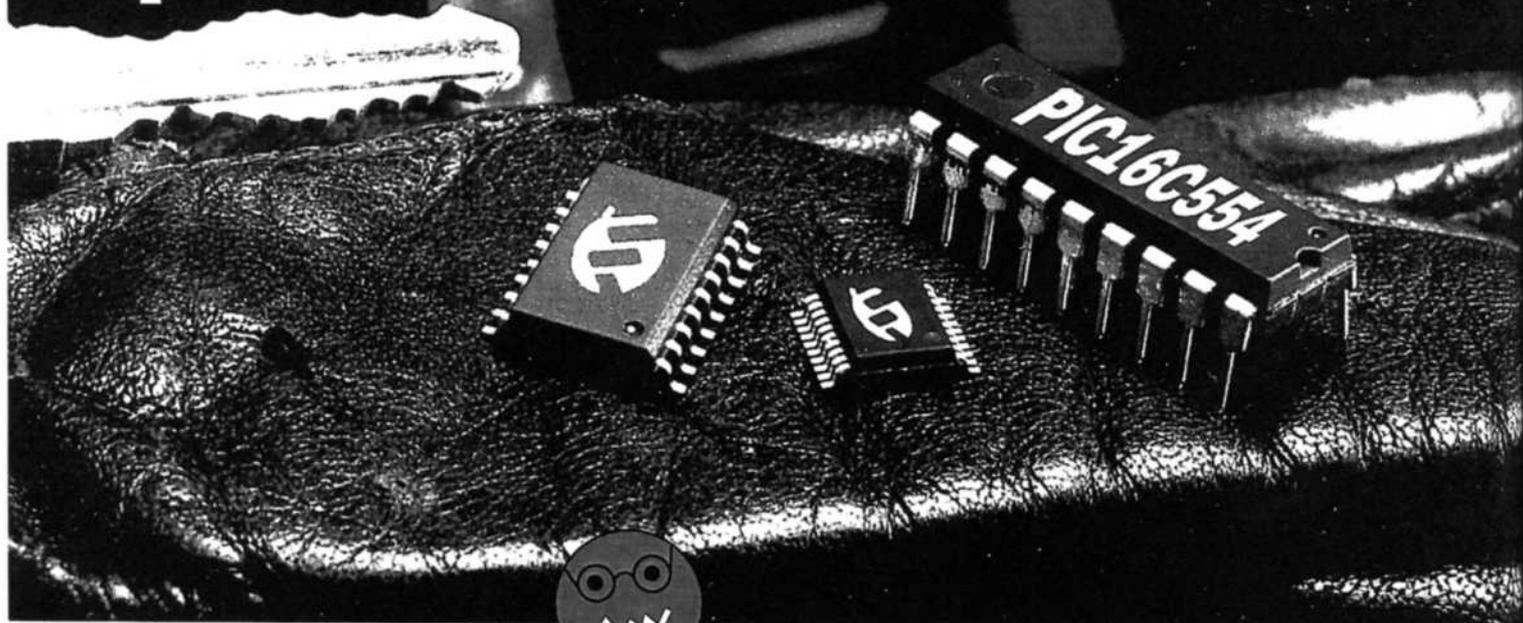
L'ultimo registro dedicato al modulo convertitore è ADCON1, visibile in Figura 11: con i primi tre bit è possibile impostare sul controller le configurazioni sui pin GP0, GP1, GP2 e GP4 in modo da selezionare il Vref voluto e gli ingressi analogici o digitali.

Poiché non sempre sono indispensabili gli otto bit di definizione relativi al convertitore, è possibile ottenere conversioni con una risoluzione più bassa, lavorando sul tempo di conversione. È sufficiente infatti bloccare il convertitore dopo un certo tempo, minore del tempo totale necessario ad una conversione completa ad otto bit. Così facendo, nel registro ADRES verranno settati solamente n bit con n inferiore ad 8.

Conoscendo per esempio il tempo di una conversione completa, interrompendola dopo tempo/2, si otterrà una risoluzione di 4 bit.

Per concludere, ricordiamo che questa periferica lavora con il metodo delle approssimazioni successive ovvero il segnale viene comparato con la metà della tensione di riferimento e, se maggiore, il bit MSB del risultato viene posto ad 1, viceversa viene azzerato. La comparazione successiva viene eseguita con la tensione di riferimento divisa per 4 e così via fino ad arrivare alla tensione di riferimento divisa per 256. ■

I microcontroller PIC più recenti



*Proseguiamo la nostra
camminata all'interno
dei microcontroller
Microchip esaminando
questo mese la famiglia
dei PIC16C55x*

Paolo Sbrana

Seconda parte

Fanno parte di questa recente famiglia tre microcontroller: il PIC16C554, il PIC16C556 ed il PIC16C558, tutti a 18 pin.

Il nome è molto simile ai 18 pin della famiglia base (16C5x), ed infatti nascono proprio per ampliare tale fascia inserendo al tempo stesso una serie di innovazioni. Cerchiamo di individuarle andando a vederne il diagramma a blocchi in figura 12.

Si nota subito che l'area della memoria di programma (EPROM) è diversa da chip a chip: i PIC16-C554 hanno 512 locazioni, i PIC16C556 hanno 1024 locazioni ed i PIC16C558 ne hanno 2048. Per quanto riguarda l'area di memoria dati (o utente o RAM) i PIC16C554 e PIC16C556 hanno a disposizione 80 registri, mentre i PIC16C558 ne hanno 128.

Un'altra sostanziale differenza rispetto alla famiglia base

sono i livelli disponibili per lo stack: otto contro due.

Ma la vera innovazione è la possibilità di lavorare con delle sorgenti di interrupt che, nei primi chip, non erano possibili.

Ultima peculiarità, la capacità di essere programmati con la programmazione in circuit ovvero con la ISP (In Serial Programming).

La gestione della memoria

Come già accennato, ogni microcontroller ha una propria gestione delle due aree di memoria (EPROM e RAM). In figura 13-a, 13-b e 13-c distinguiamo le diverse capacità.

Il PIC 16C554 ha il vettore di start alla locazione 000h, il vettore di interrupt alla locazione 4 e l'ultima locazione di memoria disponibile all'indirizzo 01FFh.

Il PIC 16C556 ha gli stessi indirizzi eccetto che per l'ultima locazione disponibile che si trova all'indirizzo 03FFh.

Il PIC16C558 invece ha l'ultima locazione all'indirizzo 07F6Fh. A differenza della famiglia base, non sono previste paginazioni della memoria EPROM, abbastanza complesse da gestire specie per i meno esperti.

Ad esempio confrontando il PIC16C58 con il corrispondente PIC16C558, si sa che il primo ha la memoria di programma suddivisa in quattro

banchi da 512 locazioni ciascuno. Ciò implica che per passare da una pagina all'altra si devono settare correttamente i due bit appositi e per tornare indietro dobbiamo riconfigurarli.

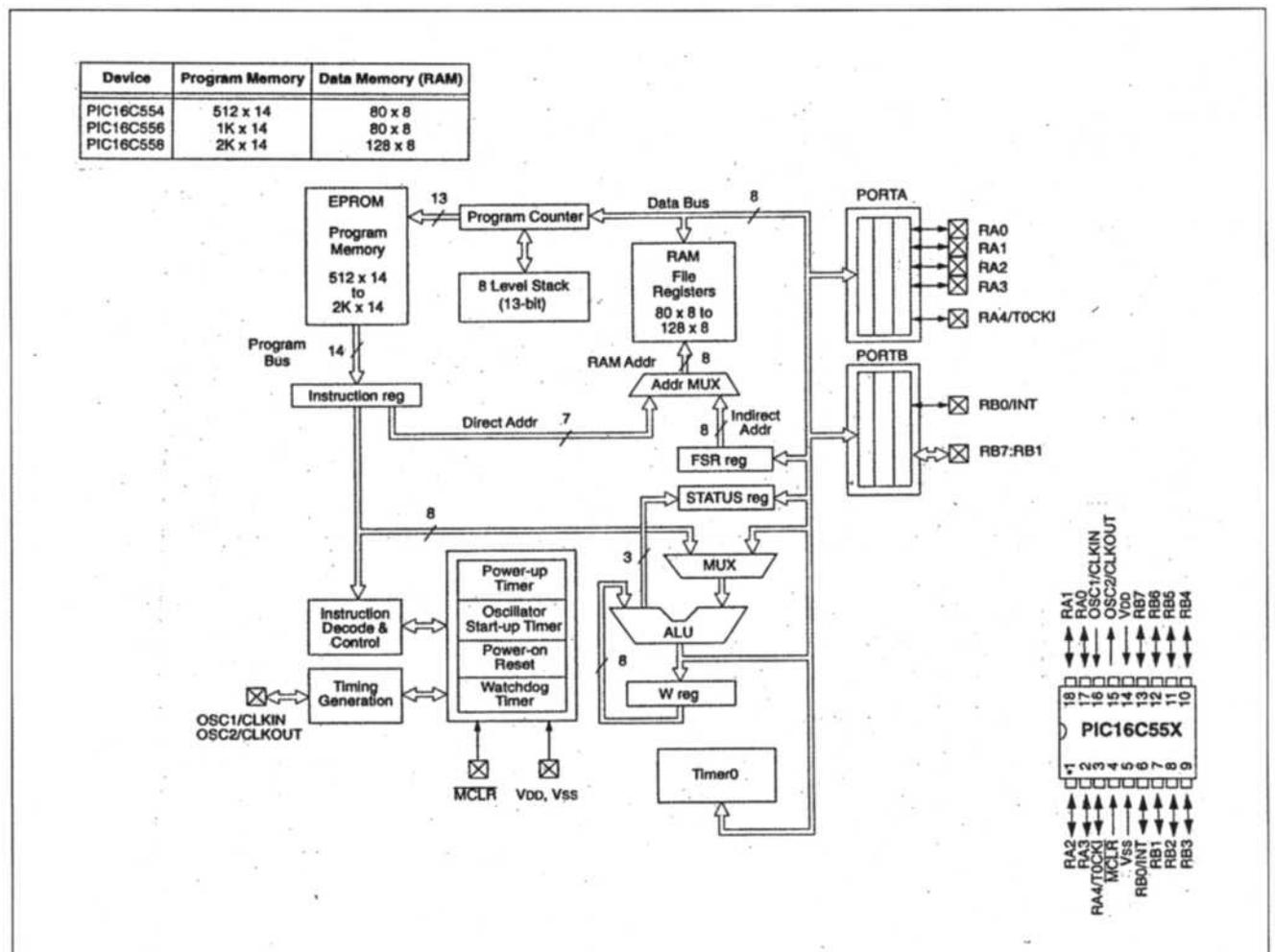
Se ci scordiamo di farlo il programma continuerà a lavorare nello stesso banco su locazioni casuali.

Il problema della paginazione della EPROM non sussiste invece nella famiglia 16C55x.

Viceversa, per uniformarsi alle nuove famiglie, sono paginati i registri specifici e quelli ad uso generale come si vede in figura 14-a e 14-b.

I registri speciali sono gli stessi per entrambi, mentre i registri ad uso generale (sfruttabili dal

Figura 12. Diagramma a blocchi della famiglia 16C55x



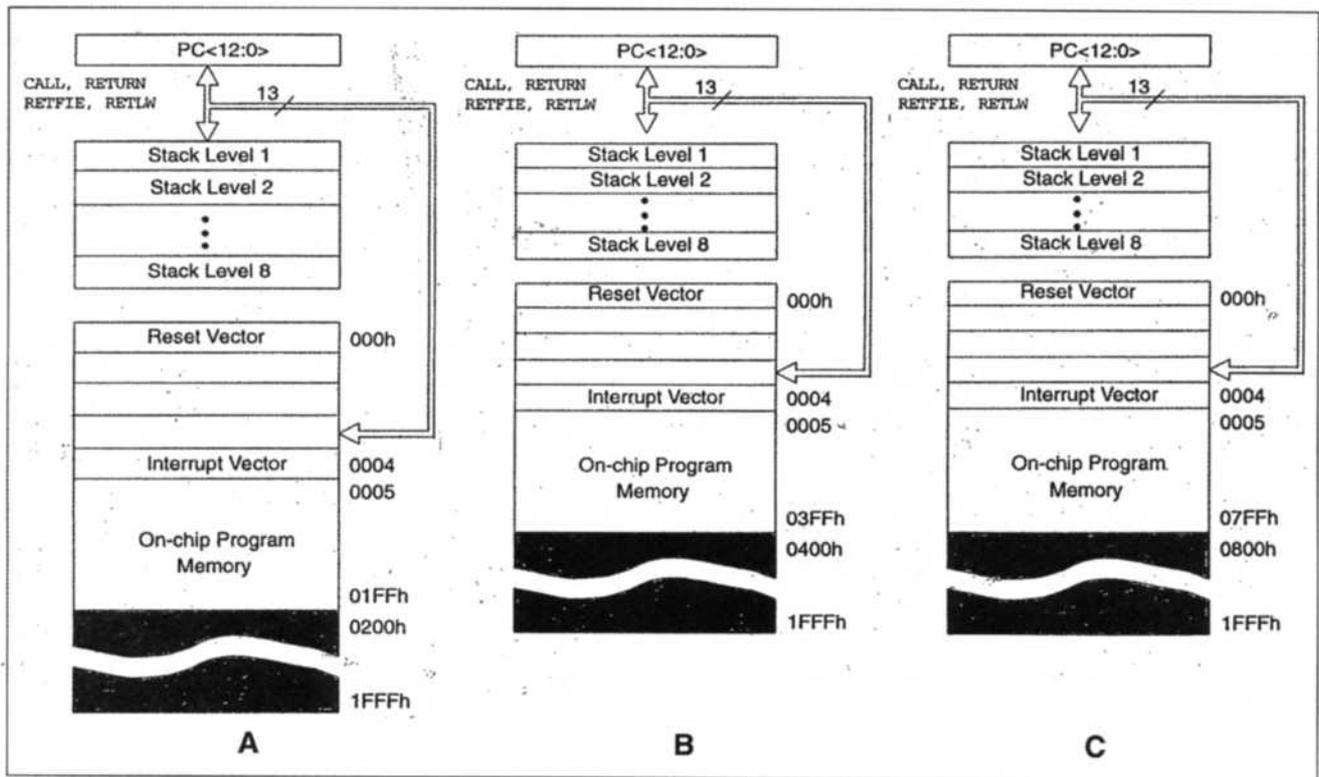


Figura 13.
Locazioni
EPROM nei
PIC16C554,
PIC16C555
e PIC16C558

programmatore) vanno dall'indirizzo 20h all'indirizzo 6Fh per i 16C554 e 16C556 mentre vanno dall'indirizzo 20h al 7Fh e dallo A0h al BFh per il 16C558.

Come si vede chiaramente, ci sono molte locazioni che non vengono utilizzate, come per esempio quelle dall'indirizzo 0Ch all'indirizzo 1Fh compreso.

Il motivo di tale decisione è che la Microchip ha voluto standardizzare tutte le nuove famiglie almeno a livello di indirizzi, quindi troveremo ad esempio che molti chip hanno la prima locazione disponibile di RAM all'indirizzo 20h ma che, a seconda delle famiglie e conseguentemente delle periferiche possedute, hanno dei registri speciali in locazioni qui non utilizzate.

I cosiddetti registri speciali (o dedicati) sono tutti quei registri che si trovano in un controller e che hanno un nome ben definito.

I registri speciali

Tra i registri speciali notiamo il TMR0 (TiMeR 0) che viene impiegato come un vero e proprio contatore ad 8 bit con o senza prescaler, il PCL (Program Counter Load) che è il registro di indirizzamento delle istruzioni, lo STATUS REGISTER dove si trovano i risultati delle operazioni ed alcune impostazioni del controller, il FSR (File Select Register) che permette di ottenere un indirizzamento indiretto, i registri PORTA e PORTB che sono le vere e proprie porte del microcontroller, il PCLATH che è la parte alta del registro PCL visto prima, il registro INTCON che vedremo tra poco gestisce le interruzioni, il registro OPTION che permette di impostare alcune opzioni di funzionamento, i due registri TRISA e TRISB che permettono di settare un pin come ingresso o come uscita ed infine il registro PCON il cui contenuto è visibile in figura 15.

L'unico bit utilizzato dal registro PCON è un flag che ci dice se il microcontroller si è risvegliato da uno stato di reset oppure dopo che gli è stata fornita alimentazione. In pratica ci serve per capire se si è resettato per un qualsiasi motivo software (stato di sleep, reset sul pin MCLR, watchdog) oppure perché semplicemente si è tolta e ridata alimentazione.

L'altro registro che nella famiglia base non è presente è quello della gestione degli interrupt rappresentato in figura 16 (INTCON). Il bit GIE (Global Interrupt Enable) abilita o meno il salto alla routine di interrupt.

Il bit TOIE (Timer0 Overflow Interrupt Enable) abilita o meno l'interrupt generato dall'overflow del Timer0.

Il bit INTE (RB0/INT External interrupt enable bit) abilita o meno l'interrupt esterno dal pin RB0.

Il bit RBIE (RB port change Interrupt Enable) abilita o meno

l'interrupt dal cambio di stato sulla porta B.

I rimanenti tre bit TOIF (Timer0 Overflow Interrupt Flag), INTF (RB0/INT interrupt enable Flag) e RBIF (RB Interrupt Flag) sono i corrispondenti flag che segnalano l'avvenuto tipo di interrupt.

Questi ultimi sono indipendenti e quindi possono trovarsi settati anche tutti e tre contemporaneamente.

Il registro di configurazione

L'ultimo registro dedicato che andiamo adesso a vedere è un registro che in genere non viene mai trattato, perché riguarda soltanto la fase di programmazione del micro, ma come vedremo è importante anch'esso.

In figura 17 notiamo il contenuto del registro di configurazione.

Tale registro non è collocato nella memoria disponibile all'utente, ma in una (o più di una a seconda dei casi) locazione adiacente (in genere dopo l'ultima locazione sfruttata dalla memoria di programma).

La caratteristica principale di questo registro è la non-programmabilità, ovvero come la memoria di programma si può scrivere una volta soltanto: vediamo il perché.

I primi due bit (FOSC1 e FOSC2) selezionano il tipo di oscillatore prescelto (che ricordiamo può essere di tipo LP, RC, XT o HS), il terzo bit (WDTE) decide se abilitare o meno il watchdog, il quarto bit (PWRTE) invece serve per ottenere all'accensione del micro un ritardo di alcuni millisecondi prima di far partire l'esecuzione del programma. Questa opzione non è presente nella famiglia base

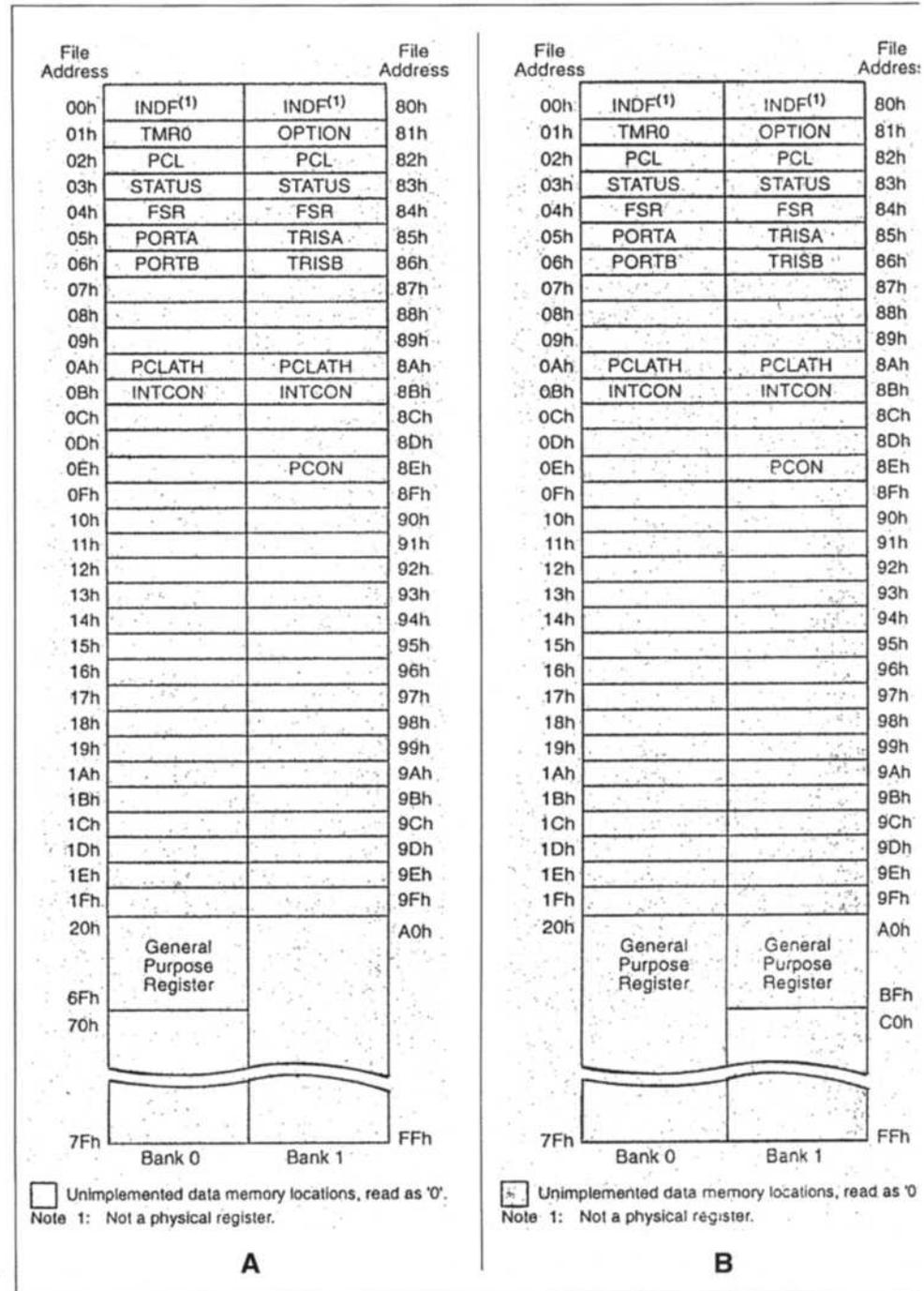
ma è molto utile in quei casi in cui il micro venga alimentato con una tensione ricavata dalla rete e che ha una pendenza molto lenta (sale da 0 a 5 volt molto lentamente) e quindi è consigliabile che il programma parta solo quando la tensione di alimentazione si sia ben stabilizzata. Gli ultimi bit permettono di definire il tipo di protezione

Figura 14.
Paginazione dei registri nei PIC16C554, 556 e PIC16C558

della memoria di programma per il micro, ovvero per non permettere a nessuno di rileggerlo, nemmeno al programmatore stesso.

Uno dei punti di forza di questi microcontroller infatti, è l'impossibilità di copiatura nel caso in cui siano protetti dalla lettura.

Vediamo che per questa famiglia, sono disponibili quattro



tipi di protezione: nessuna protezione, protezione di metà programma, protezione di 3/4 del programma e protezione totale.

Non è ancora chiaro il perché di una tale divisione: in genere i PIC in commercio si trovano o interamente protetti o interamente non protetti.

Le sorgenti hardware di reset

Come si sa i PIC possono essere resettati in vari modi, sia software che hardware.

In figura 18 abbiamo il diagramma a blocchi del reset nella famiglia 16C55x.

La prima sorgente di reset è hardware ed è causata dal pin di alimentazione: quando il PIC viene alimentato si genera sempre un reset che consente al program counter di essere inizializzato con l'indirizzo della prima locazione di memoria del programma. Sempre hardware

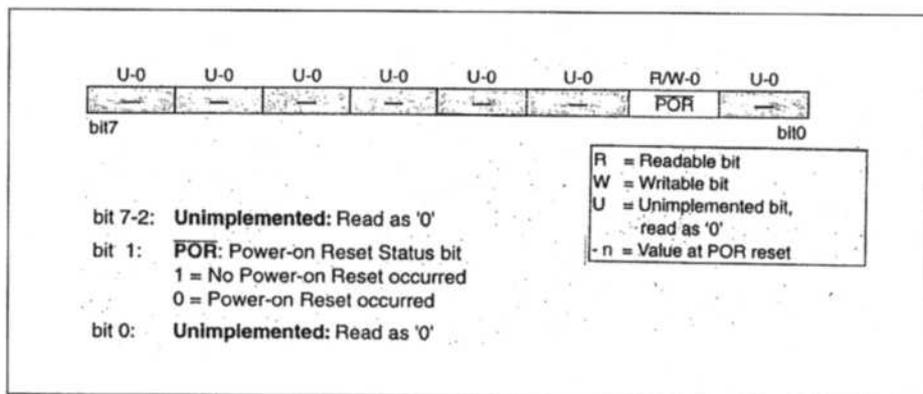


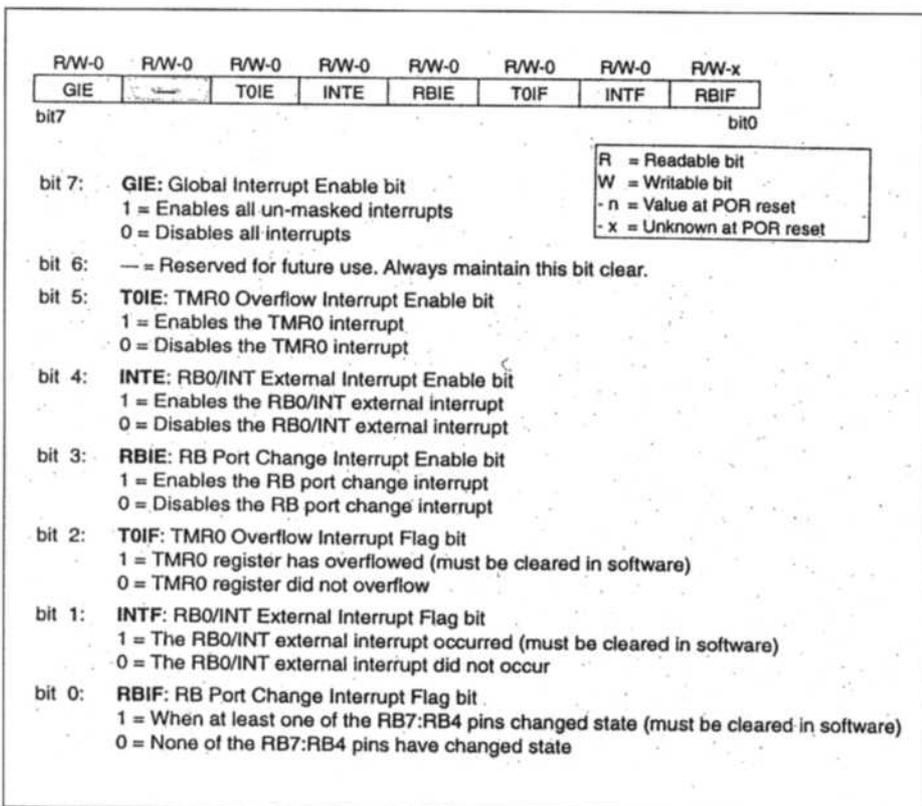
Figura 15.
Il registro PCON

è la sorgente di interrupt collegata al pin MCLR (Master CLeaR).

Portando a basso livello tale pin durante il normale funzionamento, si resetta il micro fino a quando il livello su tale pin non torna ad essere prossimo ai 5 volt.

Spesso questa caratteristica viene sfruttata per resettare il micro quando la tensione di alimentazione scende sotto un certo livello prefissato da un apposito circuito esterno detto "Brown-Out Detection".

Figura 16.
Il registro INTCON



Altra fonte di reset hardware è il PWRT (PoWer up on ReseT) che abbiamo già esaminato con il bit PWRT del registro di configurazione.

Il prescaler

Abbiamo precedentemente visto che i PIC della famiglia 16C55x hanno al loro interno un registro detto Timer0 ed un altro detto watchdog register (vedi figura 12).

Oltre a questi hanno anche un prescaler programmabile che però può essere abbinato soltanto ad uno dei due registri appena descritti.

Sul CD-ROM troviamo lo schema a blocchi della condivisione tra il prescaler ed il registro Timer0 ed il registro watchdog.

Innanzitutto, il prescaler deve essere assegnato ad uno dei due registri attraverso il settaggio del bit PSA.

Chiaramente, se il prescaler viene assegnato al watchdog e questo non è stato attivato in fase di programmazione del micro, il prescaler non verrà sfruttato.

Poi si devono settare i tre bit PS0, PS1 e PS2 che consentono di impostare i valori di divisione del prescaler: attenzione che a parità di configurazione i valori sono differenziati per i due registri fruitori.

In più, come si vede dal diagramma a blocchi, è possibile incrementare il Timer0 o con il clock del micro stesso, oppure con un segnale ad onda quadra da inviare sul pin TOKI.

La programmazione seriale

Tutti i nuovi chip della Microchip sono programmabili in circuit, ovvero sono compatibili ISP (In Serial Programming).

Per far ciò sono necessari solamente cinque fili: i due per l'alimentazione, uno per la tensione di programmazione, uno per il clock ed uno per i dati. Questo tipo di programmazione consente di programmare i chip anche già montati sullo stampato (ad esempio è comoda per gli SMD) con pochissimi collegamenti accessori. Non solo, ma è possibile anche realizzare un programmatore personale seguendo le specifiche

CP1	CP0	CP1	CP0	CP1	CP0	Reserved	CP1	CP0	PWRT	WDTE	FOSC1	FOSC0	
bit13												bit0	
bit 13-8: CP<1:0>: Code protection bits ⁽¹⁾													
5-4: 11 = Code protection off													
10 = Upper half of program memory code protected													
01 = Upper 3/4th of program memory code protected													
00 = All memory is code protected													
bit 7: Unimplemented: Read as '1'													
bit 6: Reserved: Do not use													
bit 3: PWRT: Power-up Timer Enable bit													
1 = PWRT disabled													
0 = PWRT enabled													
bit 2: WDTE: Watchdog Timer Enable bit													
1 = WDT enabled													
0 = WDT disabled													
bit 1-0: FOSC1:FOSC0: Oscillator Selection bits													
11 = RC oscillator													
10 = HS oscillator													
01 = XT oscillator													
00 = LP oscillator													
Note 1: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.													

presenti sul data-book fornito dalla Microchip.

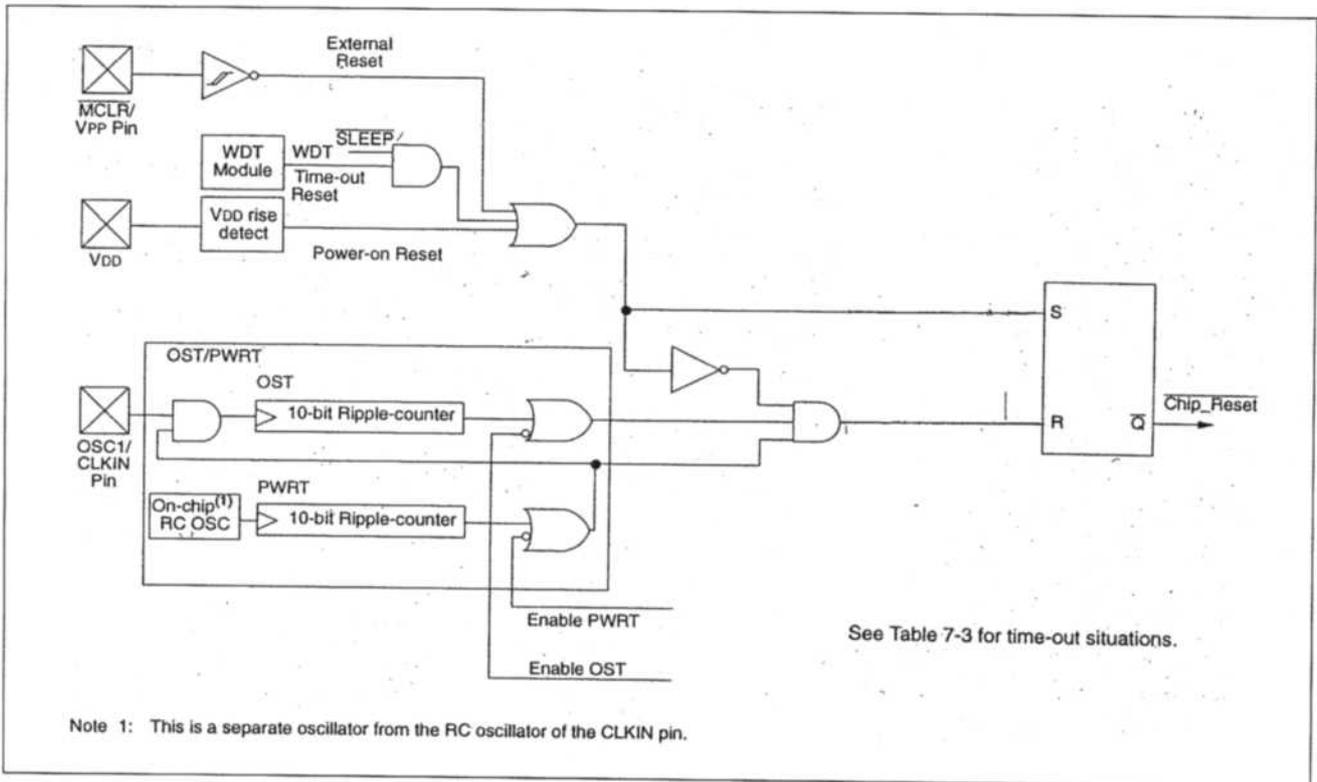
Il programmatore proposto su *Progetto PC Upgrade* di Marzo '98 lavora con questo principio. Purtroppo i micro della famiglia base (PIC16C5x) non supportano tale tipo di programmazione.

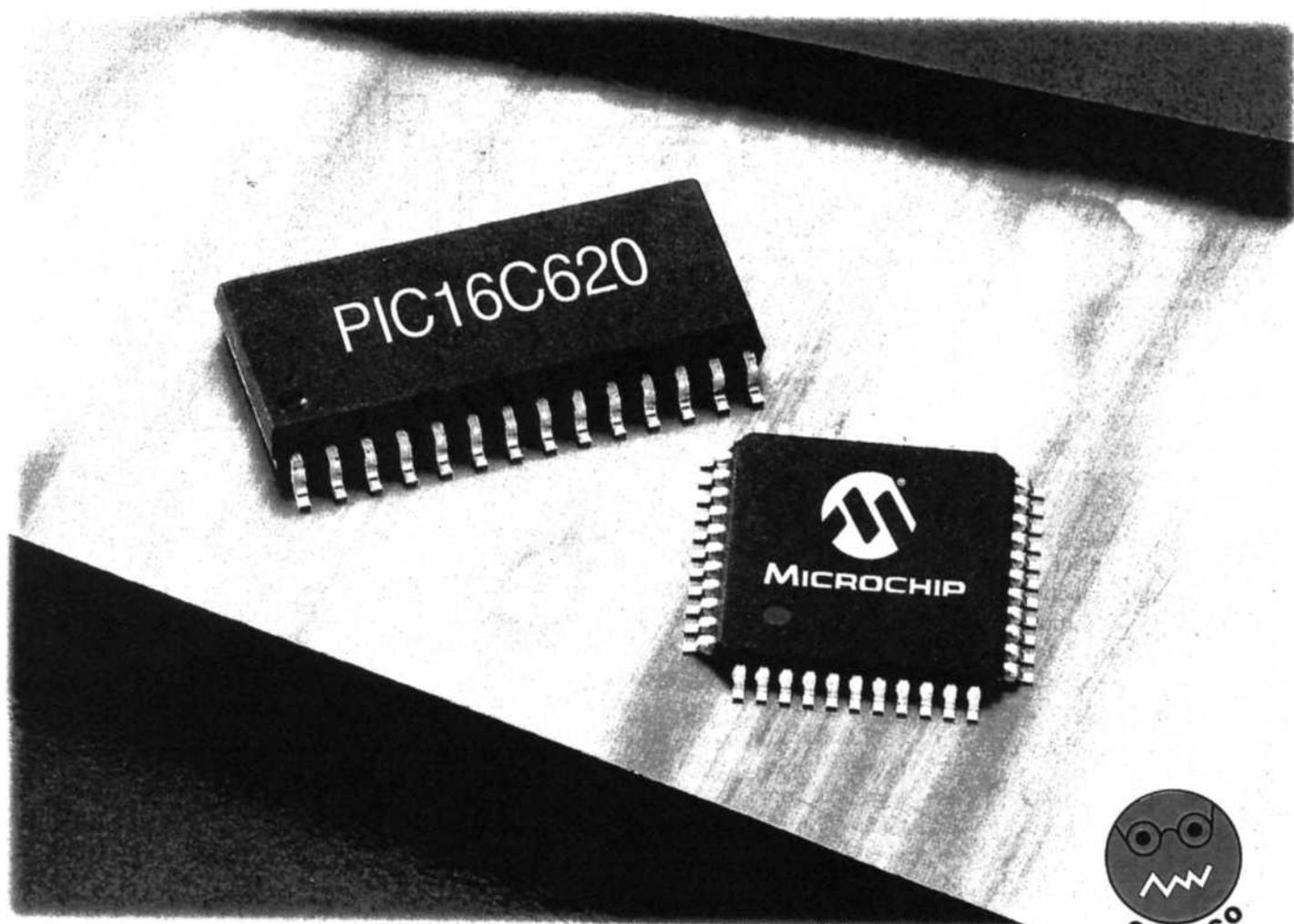
continua

Figura 17. Il registro di configurazione

Figura 18. Sviluppo di un reset nei 16C55x

Per motivi di spazio non è stato possibile inserire tutte le immagini dell'articolo. Le trovate in forma digitale sul CD-ROM di *Progetto PC Upgrade* nella directory D:\Pic. Sono state salvate in due diversi formati, le potete leggere e stampare con qualunque programma di grafica come Paint Shop Pro.





i controller PIC più attuali

Terza parte

Paolo Sbrana

Come in ogni settore dell'elettronica, anche nel campo dei microcontroller ogni giorno nascono delle novità. Esploriamo i nuovi chip della Microchip

Nei precedenti articoli abbiamo approfondito le famiglie 12Cxx e 16C-5xx valutandone le caratteristiche salienti ed i settori di applicazione più conosciuti. Passiamo adesso alla famiglia 16C62x.

Nel momento in cui scriviamo i modelli disponibili sono tre e tutti a 18 pin (vedi piedinatura in figura 21), ma sono già in test chip a 28 ed a 40 pin sempre della stessa famiglia. Per l'esattezza attualmente sono disponibili i PIC16C620, PIC16C621 ed i PIC16C622, e stanno per uscire i PIC16C642 (28 pin) ed i PIC16C662 (40 pin).

Che cosa accumuna questa famiglia ed in che cosa differiscono tra di loro questi controller? Diciamo subito che la peculiarità della famiglia sono due periferiche che hanno in esclusiva: una coppia di comparatori ed un modulo come riferimento di tensione.

In più, essendo abbastanza recenti, supportano tutti la programmazione in circuit e la periferica del Brown-Out-Detection, ovvero il circuito che controlla costantemente la tensione di alimentazione e produce un reset hardware quando questa scende sotto una determinata soglia. Sono presenti inoltre le sorgenti di interrupt più comuni come il fronte sul pin RB0, l'overflow del Timer0, l'overflow del watchdog, il cambio di stato sui pin RB7, RB6, RB5, RB4.

La struttura interna

Per vedere in modo migliore come sono strutturati i PIC16C62x all'interno, aiutiamoci con il diagramma a blocchi visibile in figura 22.

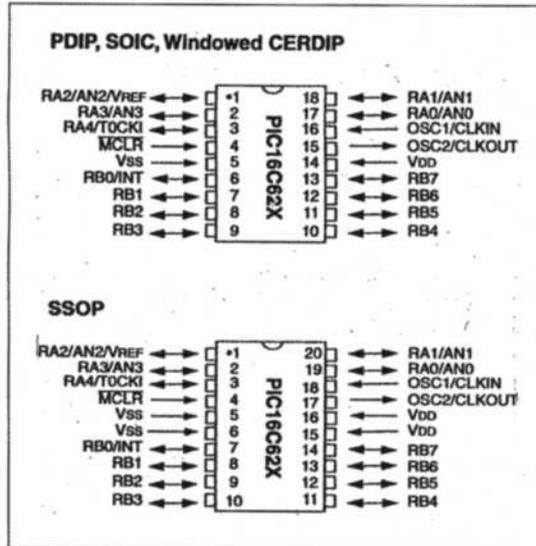
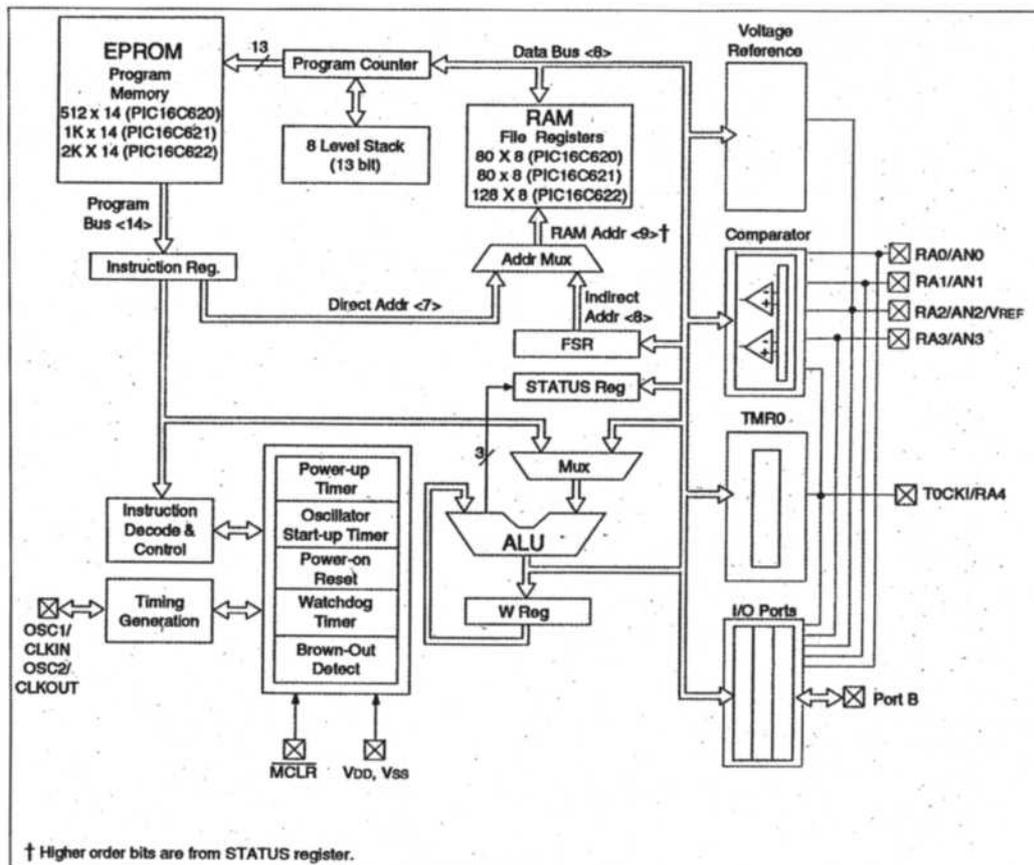


Figura 21.
Piedinatura dei PIC16C62x

I blocchi principali sono identici a tutti gli altri chip mid-range, mentre si nota subito l'aggiunta del blocco del Brown-Out-Detection, del modulo Voltage Reference (in alto a destra) e del modulo comparatore (sotto il precedente).

Figura 22.
Diagramma a blocchi dei PIC16C62x



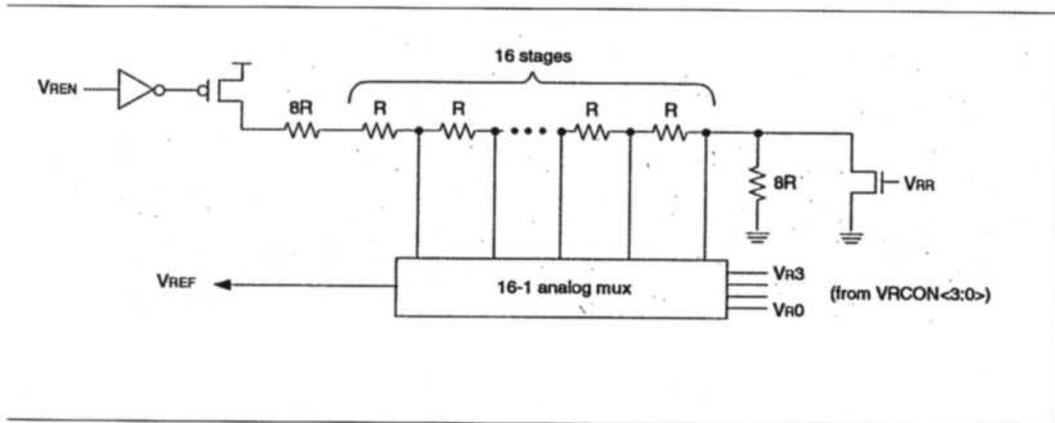
† Higher order bits are from STATUS register.

Si nota subito che entrambi questi ultimi moduli dialogano con il mondo esterno esclusivamente con la porta A.

Il primo è connesso con il pin RA2/VREF, mentre il secondo occupa interamente la porta. Di questo posizionamento si dovrà tener conto in fase di progettazione del circuito, in quanto se avessimo per esempio necessità di generare una tensione di riferimento, ci dovremo obbligatoriamente connettere al pin RA2/VREF. Conoscere questo tipo di informazioni è fondamentale per non essere poi costretti a riprogettare interamente l'hardware.

I tagli di memoria

Sempre in figura 22 si può notare che i tagli di memoria, ovvero sia la EPROM di programma sia la RAM dati, sono



diversi da chip a chip, ed in pratica queste sono le uniche differenze all'interno della stessa famiglia.

In figura 23 è mostrato lo schema concettuale del generatore (o riferimento) di tensione: è composto da 16 stadi resistivi e segmentato per offrire due range di tensione.

Inoltre ha il controllo on/off software per permetterne l'accensione e lo spegnimento da programma per preservare corrente quando non sia indispensabile. Abbinare a questa

Figura 23.
Il modulo di riferimento alla tensione

Figura 24.
Tagli di memoria EEPROM per i 16C620(A) 16C621(B) 16C622(C)

figura il registro VRCON vedi figura 28. In figura 24 vediamo i tre tagli relativi alla memoria EEPROM di programma. Il program counter è sempre di 13 bit (0..12) quindi offre la possibilità di indirizzare fino a 2K di locazioni.

I livelli dello stack, ovvero il numero di chiamate annidate a subroutine, sono sempre 8 come pure il reset vector è all'indirizzo 0000h ed l'interrupt vector è all'indirizzo 0004h.

L'ultima locazione disponibile per contenere istruzioni

di programma è invece la 01ffh per i PIC16C620, la 03ffh per i PIC16C621 e la 07ffh per i PIC16C622.

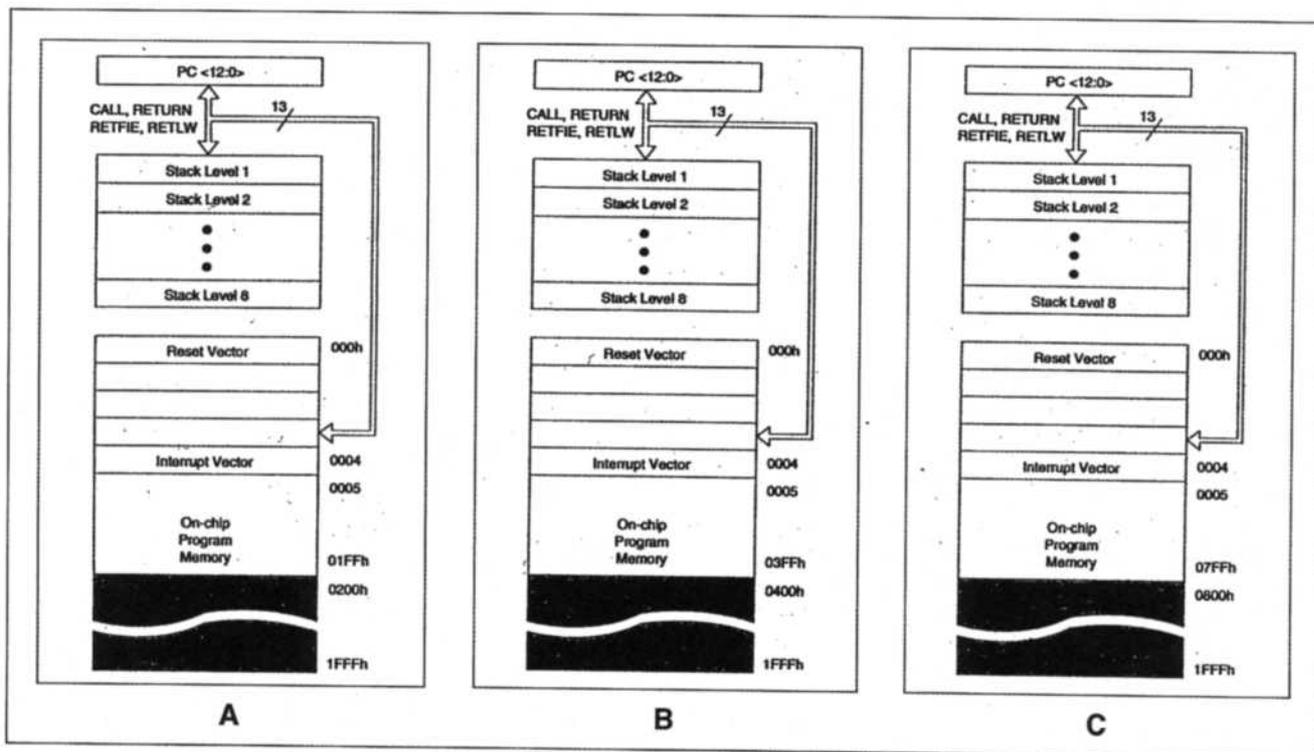
Poiché abbiamo già notizie delle caratteristiche dei nuovissimi 16C642 e 16C662, anticipiamo che entrambi avranno una capacità EPROM di 4K suddivisa in due banchi da 2K ciascuno.

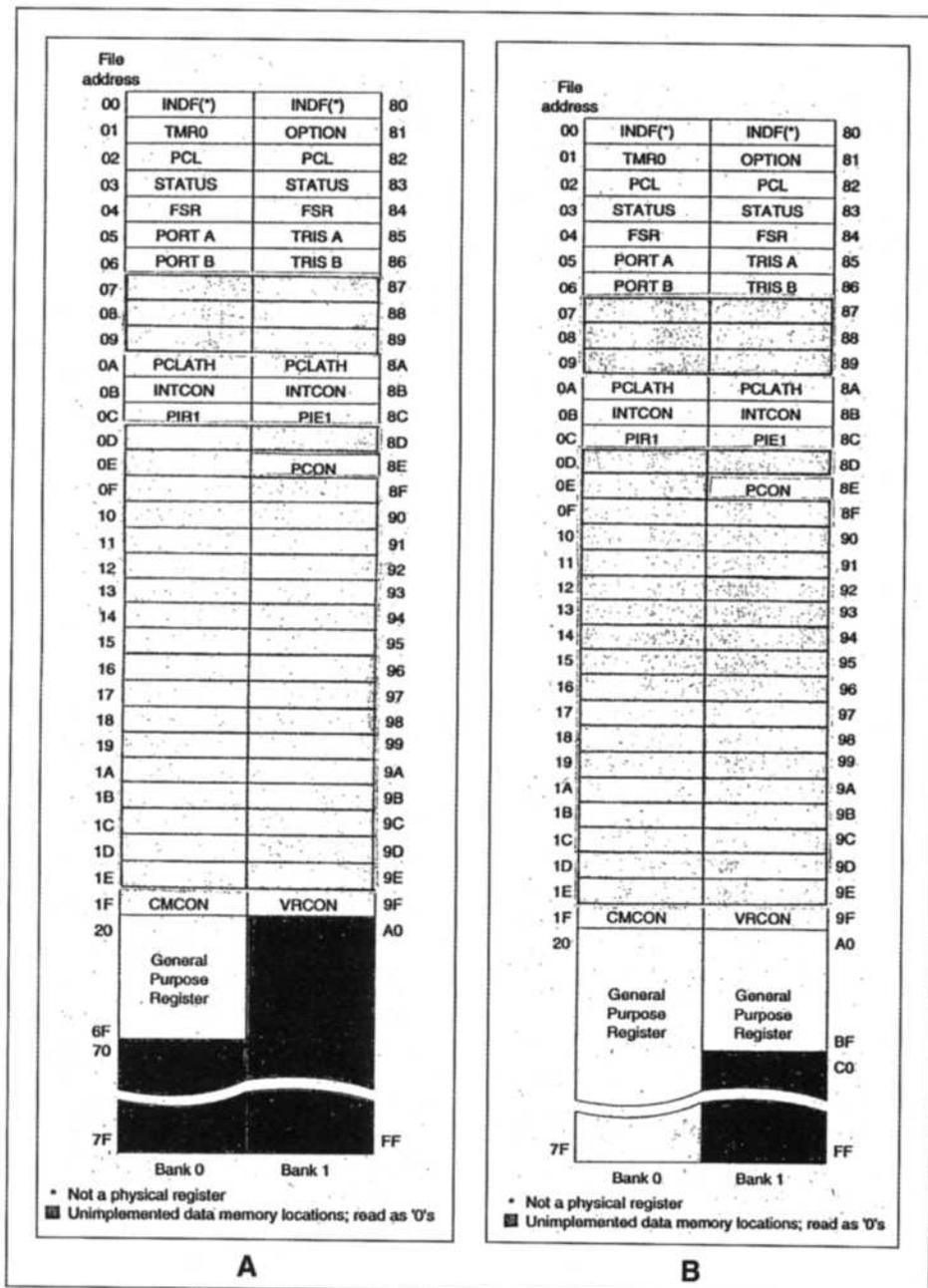
Per quanto riguarda la memoria RAM invece, ossia per la memoria disponibile per il mantenimento dei dati durante lo svolgimento del programma, in figura 25 vediamo le due possibilità: i PIC16C620 e 16C621 ne hanno 80 mentre i PIC16C622 ne hanno 128.

Dobbiamo precisare però alcune cose.

Innanzitutto dobbiamo distinguere tra registri RAM speciali (o dedicati) e registri RAM general purpose (ossia di uso generale).

I registri RAM indicati precedentemente sono i registri general purpose, cioè quelli che





il programmatore può adoperare per memorizzare i dati del programma, quindi quelli realmente liberi. Gli altri invece sono registri che il programmatore deve in ogni caso adoperare ma che non può sfruttare per memorizzare dati propri.

Per chiarire meglio, diciamo che i registri speciali sono il program counter, il Timer0, lo status register, i registri di settaggio delle porte, il registro di

puntamento indiretto ecc. Quindi se devo memorizzare il valore 023h, non posso certo sfruttare un registro speciale, poiché per esempio potrei variare la direzione della porta B agendo sul registro TRISB. Seconda questione da affrontare è la paginazione dei registri RAM. In figura 25 si vede che i registri sono suddivisi in due colonne. La prima corrisponde al banco 0, la seconda al banco 1.

Figura 25. Memoria RAM nei 16C620, 16C621(A) e 16C622(B) suddivisi su due colonne

Che cosa significa? Questo fatto implica che per agire su di un registro o su di un altro, devo essere sicuro che il bit che mi setta la pagina RAM corrente sia settato opportunamente.

Poiché ciò non è molto agevole, in generale i programmatori tendono a sfruttare i registri disponibili nel banco 0 il più possibile e passano agli altri solo nei casi indispensabili. Per completezza di informazione anticipiamo che, se Microchip non cambierà i dati già anticipati, i nuovissimi PIC16C642 e 16C-662 avranno la bellezza di 176 registri RAM.

Il modulo comparatore

La sezione relativa ai comparatori comprende due comparatori che possono essere configurati in otto modi diversi, a seconda delle differenti esigenze. In figura 26 vediamo le possibili soluzioni: la prima corrisponde allo stato di reset dei due comparatori.

La seconda consiste nel disporre di due comparatori che lavorano indipendentemente l'uno dall'altro.

La terza ha i due comparatori con l'ingresso V+ in comune.

La quarta permette di utilizzare un solo comparatore e di lasciare l'altro spento. La quinta mantiene spenti entrambi i comparatori (minimo assorbimento).

La sesta consente di avere quattro input multiplexati sui due comparatori.

La settima ha invece i due input V+ in comune con disponibili anche le due uscite ed infine l'ultima dispone in comune i due ingressi V+ ed

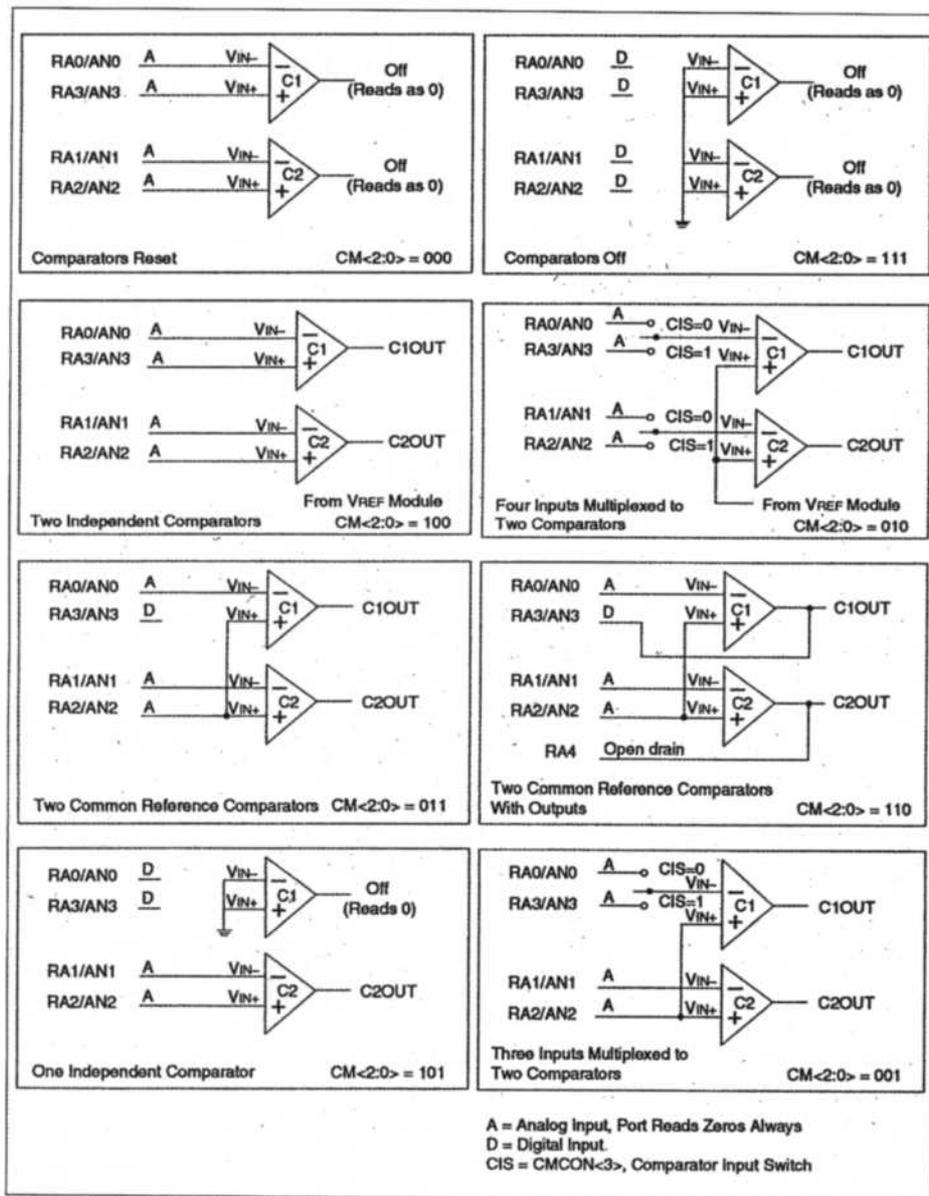


Figura 26.
Possibili
configurazioni
dei due
comparatori

offre tre input multiplexati. Per selezionare uno degli otto modi di funzionamento possibili, si deve agire sul registro COMCON (COMparator CONTROL) il cui significato è visibile in figura 27.

I primi tre bit meno significativi impostano una delle otto configurazioni permesse.

Il bit CIS (Comparator Input Switch) controlla la selezione degli ingressi multiplexati.

I due bit C1OUT e C2OUT invece sono di sola lettura e ci danno preziose informazioni

sullo stato di ingresso dei due comparatori.

Per esempio se la tensione al pin V+ del comparatore numero 1 è maggiore di quella al pin V-, tale bit varrà 1, viceversa varrà 0.

Quindi si potrà verificare l'uscita del comparatore semplicemente andando a testare tale bit nel caso in cui l'uscita fisica non sia disponibile.

Dobbiamo comunque precisare che quando viene letta la porta A, tutti i pin configurati come ingressi analogici

saranno sempre letti come "0". I pin invece configurati come ingressi digitali varranno "1" o "0" a seconda del valore del trigger di Schmitt interno.

Inoltre, inviare un segnale analogico su di un pin configurato come input digitale può far assorbire al buffer di input più corrente di quella specificata.

Per il modulo comparatore sono previste delle sorgenti di interrupt.

Nel caso in cui si decida di usarle, si dovrà settare il bit CMIE (CoMparator Interrupt Enable) ed il bit PEIE (Peripheral Interrupt Enable).

Inoltre dovrà essere settato il bit GIE (Global Interrupt Enable).

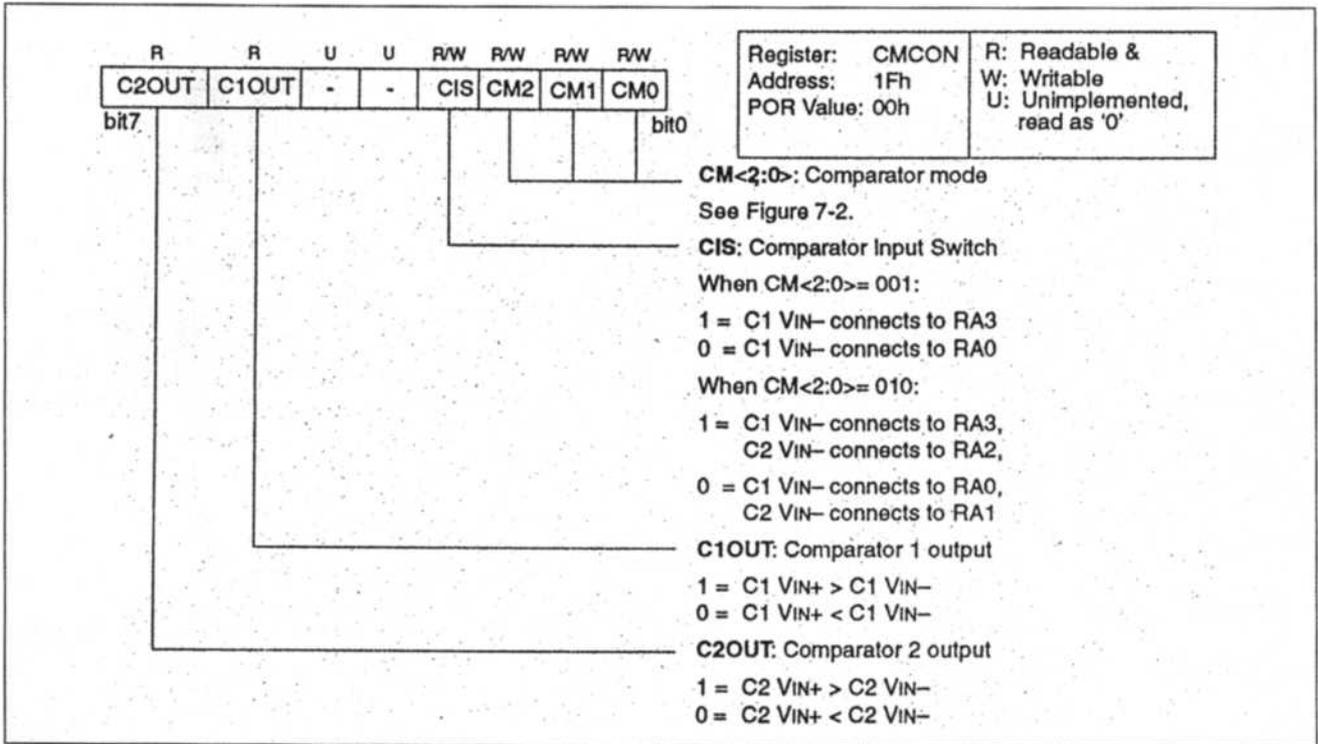
Poiché il flag di interrupt relativo al modulo comparatore è unico, si avrà generazione di interrupt anche nel caso in cui un solo comparatore modifichi il suo stato di uscita e quindi si demanda al programmatore l'analisi successiva per capire che cosa abbia generato l'interrupt.

Anche per questa periferica, è possibile far eseguire una comparazione dopo aver messo in SLEEP il microcontroller ed essere svegliati al termine della comparazione stessa.

Attenzione però perché poiché il modulo comparatore assorbe una certa corrente, se si decide di far entrare in SLEEP il controller e non sono necessarie comparazioni, è fortemente consigliabile disabilitare i due comparatori.

Un reset del controller forza il registro COMCON in uno stato di reset in cui i comparatori sono spenti.

Ciò assicura che tutti gli input siano di tipo analogico, limitando la corrente in fase di inizializzazione.



Il modulo di riferimento tensione

Poiché lo stadio comparatore, come dice il nome, deve comparare almeno due tensioni per fornirci un dato valido in uscita, può essere utile abbinargli un riferimento di tensione abbastanza preciso ed eventualmente

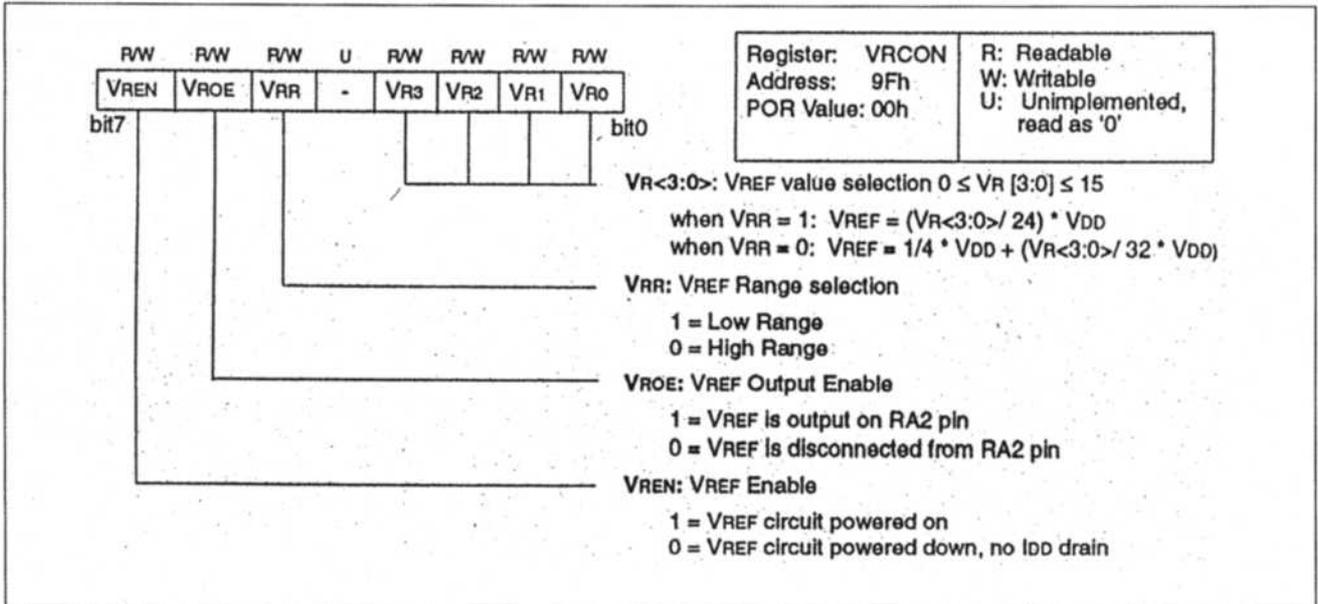
variabile. Per questo motivo nei PIC della famiglia 16C62x troviamo, oltre ai comparatori, anche un modulo che genera una tensione programmabile limitatamente ad un certo range. Abbinato a questo modulo abbiamo il registro VRCON (Voltage Reference Controller) il cui significato è disponibile in figura 28.

Figura 27.
Il registro COMCON

Figura 28.
Il registro VRCON

Con i bit assegnati è possibile attivare uno o l'altro range di funzionamento, collegare l'uscita Vref sul pin RA2 o meno ed accendere o spegnere tutto il modulo. Questo modulo è del tutto indipendente da quello del comparatore e quindi è attivabile anche autonomamente.

continua



i nuovi PIC

12/16/17xxx



Paolo Sbrana

Come in ogni settore dell'elettronica, anche nel campo dei microcontroller ogni giorno nascono delle novità. Esploriamo i nuovi integrati della Microchip

Quarta parte

Proseguiamo il nostro viaggio nel mondo dei nuovi microcontroller PIC esplorando una famiglia che pensiamo sia sconosciuta alla maggioranza degli hobbisti e cioè la famiglia dei PIC16C9xx.

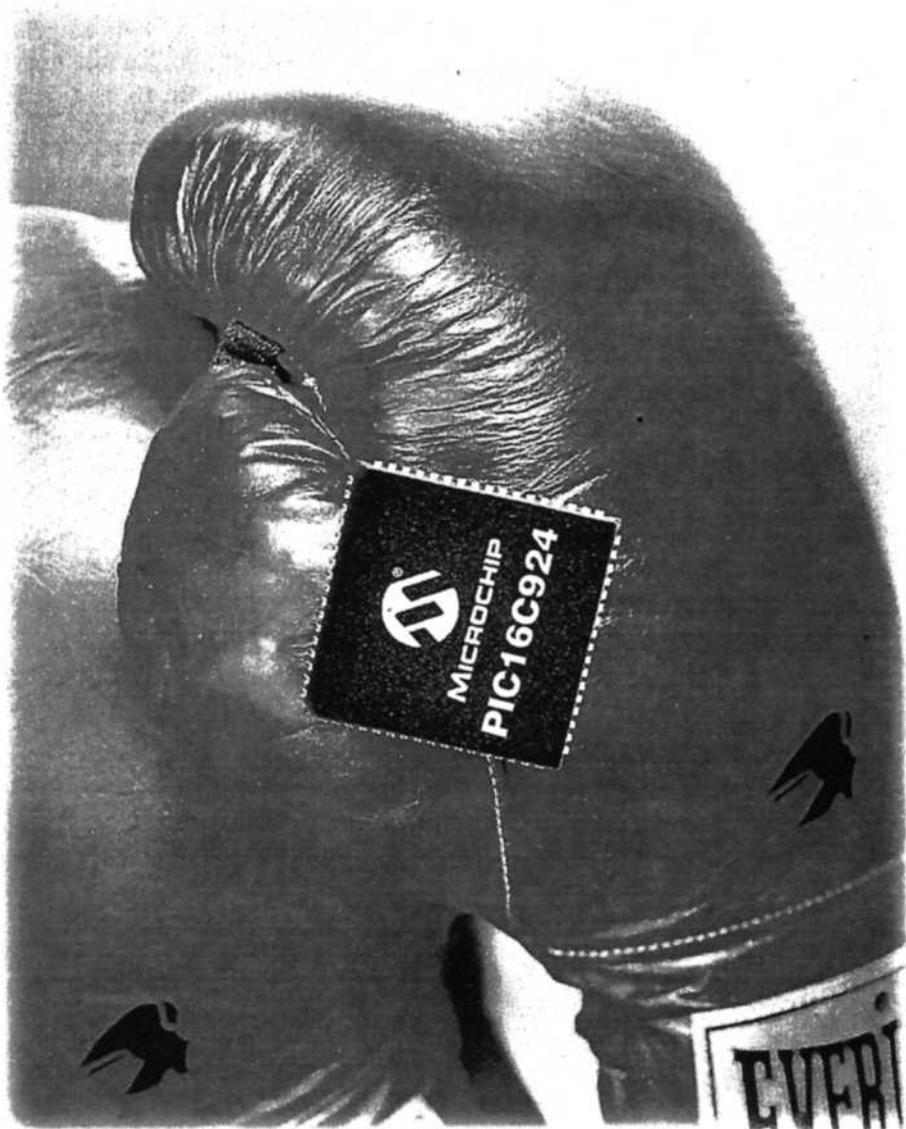
I motivi per cui tale famiglia sia sconosciuta sono abbastanza evidenti: primo tra tutti il package, ossia il tipo di contenitore di questi chip. Fino ad adesso abbiamo trattato controller dual in line oppure SMD con al massimo 64 pin.

La famiglia 16C923-24 supera questa cifra proponendo versioni PLCC con ben 68 pin.

La difficoltà di maneggiare chip così ricchi di piedini è notevole, a cominciare dalla realizzazione del circuito stampato.

Ma non potevamo non illustrare i 16C9xx perché differiscono da tutti gli altri per una periferica molto importante in funzione del lavoro da fargli svolgere e cioè il controller di cristalli liquidi.

Come sappiamo infatti, pilotare display LCD che non dispongano di controller non è



banale, e generalmente esistono micro già programmati che permettono tale funzione aggiungendo quindi complessità e costi al circuito.

I package disponibili

In figura 29, vediamo raffigurati tutti i tipi di package disponibili per la famiglia 16C9-xx.

Da questa figura si potrebbe pensare di avere a disposizione anche una versione Shrink PDIP a 750 mils, ovvero un package simile al dual-in-line comune ma con passo più stretto.

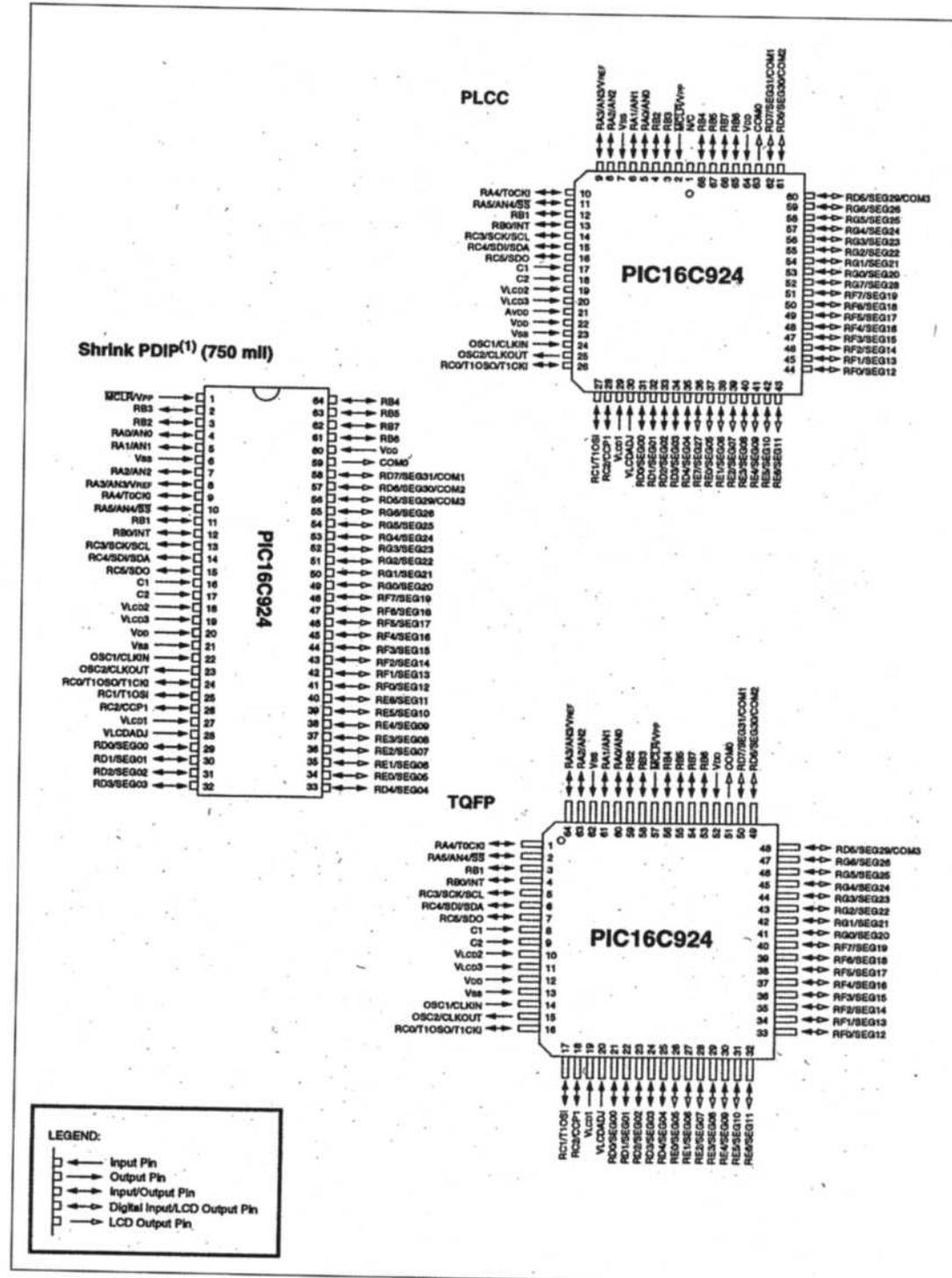
In realtà, nel momento in cui scriviamo questa non è ancora disponibile e crediamo che non verrà mai realizzata per le problematiche circuitali che ne potrebbero derivare.

Proprio per eliminare problematiche di questo tipo infatti, anche il "probe" ufficiale della Microchip per il sistema di sviluppo PICMASTER, è fornito in versione PLCC a 68 pin.

E proprio la versione PLCC è quella più movimentata perché più semplice da maneggiare utilizzando gli appositi zoccoli quadrati che si reperiscono abbastanza facilmente in commercio.

Anche la versione TQFP comunque è abbastanza movimentata, specie dove si realizzano schede completamente in SMD ed in grande quantità. Uno dei settori di maggior sbocco per questi chip con package TQFP è quello dei quadri elettronici automobilistici.

Come abbiamo modo di constatare in un numero sempre crescente di nuove automobili, i display a cristalli liquidi stanno



sostituendo via via i vecchi strumenti analogici e chi meglio dei PIC 16C9xx poteva gestire circuiti simili?

Questa breve introduzione sui package della famiglia 16C9xx è importante perché fa comprendere quanto il gestire

Figura 29. Tipi di package disponibili dei pic 16C924

tali chip sia leggermente più complesso di tutti gli altri, anche in mancanza di una versione JW, ovvero quella che la Microchip identifica come "finestrata" cioè cancellabile molte volte con lampade a raggi ultravioletti.

Tabella 2. Mappa della RAM nei 16C9xx

File Address		File Address		File Address		File Address	
Indirect addr. ⁽¹⁾	00h	Indirect addr. ⁽¹⁾	80h	Indirect addr. ⁽¹⁾	100h	Indirect addr. ⁽¹⁾	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTF	107h	TRISF	187h
PORTD	08h	TRISD	88h	PORTG	108h	TRISG	188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
	0Dh		8Dh	LCDSE	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	LCDPS	10Eh		18Eh
TMR1H	0Fh		8Fh	LCDDCON	10Fh		18Fh
T1CON	10h		90h	LCDD00	110h		190h
TMR2	11h		91h	LCDD01	111h		191h
T2CON	12h	PR2	92h	LCDD02	112h		192h
SSPBUF	13h	SSPADD	93h	LCDD03	113h		193h
SSPCON	14h	SSPSTAT	94h	LCDD04	114h		194h
CCPR1L	15h		95h	LCDD05	115h		195h
CCPR1H	16h		96h	LCDD06	116h		196h
CCP1CON	17h		97h	LCDD07	117h		197h
	18h		98h	LCDD08	118h		198h
	19h		99h	LCDD09	119h		199h
	1Ah		9Ah	LCDD10	11Ah		19Ah
	1Bh		9Bh	LCDD11	11Bh		19Bh
	1Ch		9Ch	LCDD12	11Ch		19Ch
	1Dh		9Dh	LCDD13	11Dh		19Dh
ADRES ⁽²⁾	1Eh		9Eh	LCDD14	11Eh		19Eh
ADCON0 ⁽²⁾	1Fh	ADCON1 ⁽²⁾	9Fh	LCDD15	11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register		General Purpose Register					
			EFh		16F		1EFh
		Mapped in Bank 0 70h-7Fh	F0h	Mapped in Bank 0 70h-7Fh	170	Mapped in Bank 0 70h-7Fh	1F0h
			FFh		17F		1FFh
Bank 0	7Fh	Bank 1		Bank 2		Bank 3	

Unimplemented data memory locations, read as '0'.
 Note 1: Not a physical register.
 Note 2: These registers are not implemented on the PIC16C923

il Timer 1 a 16 bit con il proprio controllo, il Timer2 con il relativo controllo, i registri della seriale sincrona, quelli dei moduli CAPTURE/COMPARE ed infine, solo per i 16C924, due registri per il convertitore.

Questi i registri RAM detti speciali perché dedicati alle funzioni interne del PIC, poi ce ne sono dall'indirizzo 0x20 all'indirizzo 0x7F di tipo generale, ovvero a disposizione del pro-

grammatore per lo svolgimento del programma.

Nel banco 1 invece, abbiamo il registro delle opzioni, i registri di controllo della direzione delle porte, il registro di abilitazione delle interruzioni da periferica, altri registri di controllo della seriale sincrona e l'ultimo controllo per il convertitore.

Anche questo banco poi rende disponibili dei registri generali RAM dall'indirizzo 0xA0

all'indirizzo 0xEF. Indirizzando invece i registri da 0xF0 a 0xFF, il programmatore, pur essendo sul banco 1, avrà a disposizione gli stessi registri che vanno da 0x70 a 0x7F nel banco 0.

Ciò è stato fatto per mettere in comune ai quattro banchi dei registri che potrebbero essere di uso comune oppure di scambio dati.

Nel banco 2 troviamo oltre ai soliti registri TMR0, PCL, STATUS ecc, anche quelli relativi al settaggio delle porte PORTB, PORTF e PORTG. Le porte F e G sono dedicate quasi esclusivamente alla gestione dei segmenti dei cristalli liquidi.

Se non si desidera sfruttarle in questa veste, è possibile tuttavia impiegarle esclusivamente come porte di input digitale.

Vedremo in dettaglio nei prossimi paragrafi come manipolare questi registri. Anche alla fine del banco 2 troviamo dall'indirizzo 0x170 all'indirizzo 0x17F i 16 registri mappati nel banco 0 e comuni a tutti i banchi.

Il quarto ed ultimo banco serve quasi esclusivamente per configurare la direzione delle porte PORTB, PORTF e PORTG, oltre ad avere i soliti 16 registri in comune dall'indirizzo 0x1F0 all'indirizzo 0x1FF.

Se facciamo due conti, vediamo che in sostanza i registri di RAM disponibili per il programmatore sono in totale 176, quindi in numero decisamente sufficiente per la maggior parte dei programmi che decideremo di inserire nel chip.

Le porte E, F e G

Poiché la struttura delle porte E, F e G è diversa dalle altre, cerchiamo di capire come funzionano e perché ad esempio possono essere configurate

solamente come ingressi. In figura 32 ne vediamo il diagramma a blocchi.

Sul pin di uscita leggiamo Digital Input/LCD Output. Si distinguono molto bene le due sezioni, una che gestisce il display LCD, l'altra che imposta la configurazione della porta e dialoga con il Data-Bus. Sul pin di I/O è collegato un buffer con abilitazione three-state.

Questo buffer riesce a pilotare un segmento lcd ma non potrà mai pilotare un'uscita digitale.

Il dato del segmento arriva all'ingresso di tale buffer e, se è presente il segnale LCD Segment Output Enable, viene trasferito sul pin di I/O.

Questa operazione avviene sempre, comunque sia configurata la porta.

Per poter sfruttare il pin come I/O digitale invece, vediamo che è necessario un segnale di enable chiamato LCDSE che arriva all'ingresso di una porta AND triggerata. Se questo segnale vale 0, la porta AND avrà sua uscita sempre uno zero logico.

Quando questo segnale vale 1 invece, la porta AND porrà sulla sua uscita lo stesso valore digitale che è presente sul pin di I/O.

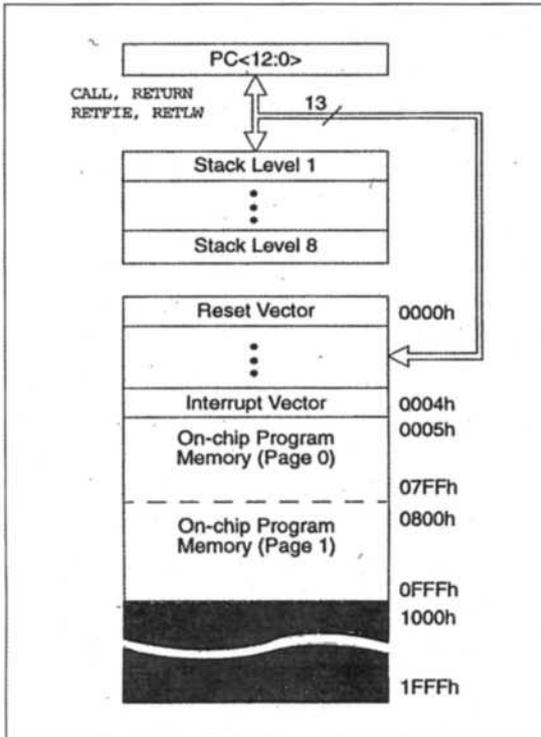
L'uscita della porta AND comunica il dato letto all'ingresso di un flip-flop di tipo D. L'abilitazione del flip-flop arriva dal

segnale RD PORT, che inoltre permette di far giungere il dato di uscita direttamente sul bus dati per mezzo di un buffer anch'esso pilotato in three-state.

Si noti che se la configurazione della direzione della porta è di output, in uscita sul bus dati si otterrà sempre un valore 1. La differenza sostanziale con le rimanenti porte rimane la non possibilità di essere impiegate come uscite ed in più, se usate come ingressi digitali, la presenza di una porta triggerata che consente di stabilire una ben precisa soglia (prefissata dal costruttore) per la determinazione del livello logico di ingresso.

Tabella 3. Caratteristiche dei 16C9xx in DC

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc Configuration XS osc configuration
D002	RAM Data Retention Voltage	VDR	-	1.5	-	V	Device in SLEEP mode
D003	VDD start voltage to Ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See Power-on Reset section for details
D010 D011 D012	Supply Current	IDD	- - -	2.7 13.5 3.5	5 30 7	mA μA mA	XT and RC osc configuration FOSC = 4 MHz, VDD = 5.5 V LP osc configuration, FOSC = 32 kHz, VDD = 4.0 V HS osc configuration FOSC = 8 MHz, VDD = 5.5 V
D20	Power-down Current	IPD	-	1.5	21	μA	VDD = 4.0 V
D021	Module Differential Current Watchdog Timer	ΔIWDT	-	6.0	20	μA	VDD = 4.0 V
D022	LCD internal RC osc enable	ΔILCDRC	-	6.0	20	μA	VDD = 4.0 V
D24	LCD voltage generation	ILCDVG	-	TBD	TBD	μA	VDD = 4.0 V
D025	Timer1 oscillator	ΔIT1OSC	-	10.6	17	μA	VDD = 4.0 V
D026	A/D Converters	ΔIAD	-	1.0	-	μA	A/D on, not converting



Le caratteristiche elettriche

Per quanto riguarda le caratteristiche elettriche in DC, questa famiglia non è molto diversa

Figura 31. Mappa della memoria EPROM dei 16C923-24

dalle altre, anche se una delle prerogative fondamentali dei PIC e cioè la velocità non è stata mantenuta: sono ad oggi disponibili solamente PIC16C9xx con al massimo 8MHz di clock (cioè che riescono ad eseguire una istruzione in 0,5us).

Inoltre la tensione di alimentazione non scende sotto i 4 volt come per altre famiglie.

Ma vediamo assieme le caratteristiche elettriche esposte in tabella 3.

La tensione di alimentazione può andare da un minimo di 4 volt ad un massimo di 6 volt. Per quel che riguarda invece la detenzione dei dati durante lo stato di SLEEP sono sufficienti 1,5 volt e da 1,5 a 21 µA di assorbimento.

L'assorbimento invece in condizioni di lavoro, compreso il pilotaggio degli LCD, va da 2,7 a 3,5 mA per frequenze di clock comprese tra 4 e 8 MHz. Se invece si impiega un cristallo a 32.768 Hz, l'assorbimento passa bruscamente a circa 15 µA.

Si deve ricordare però che l'assorbimento scende ma proporzionalmente scende anche la velocità di esecuzione: da 1µS a 4 MHz si passa a circa 122µS a 32.768 MHz.

Vediamo adesso l'assorbimento dei singoli moduli (se attivati).

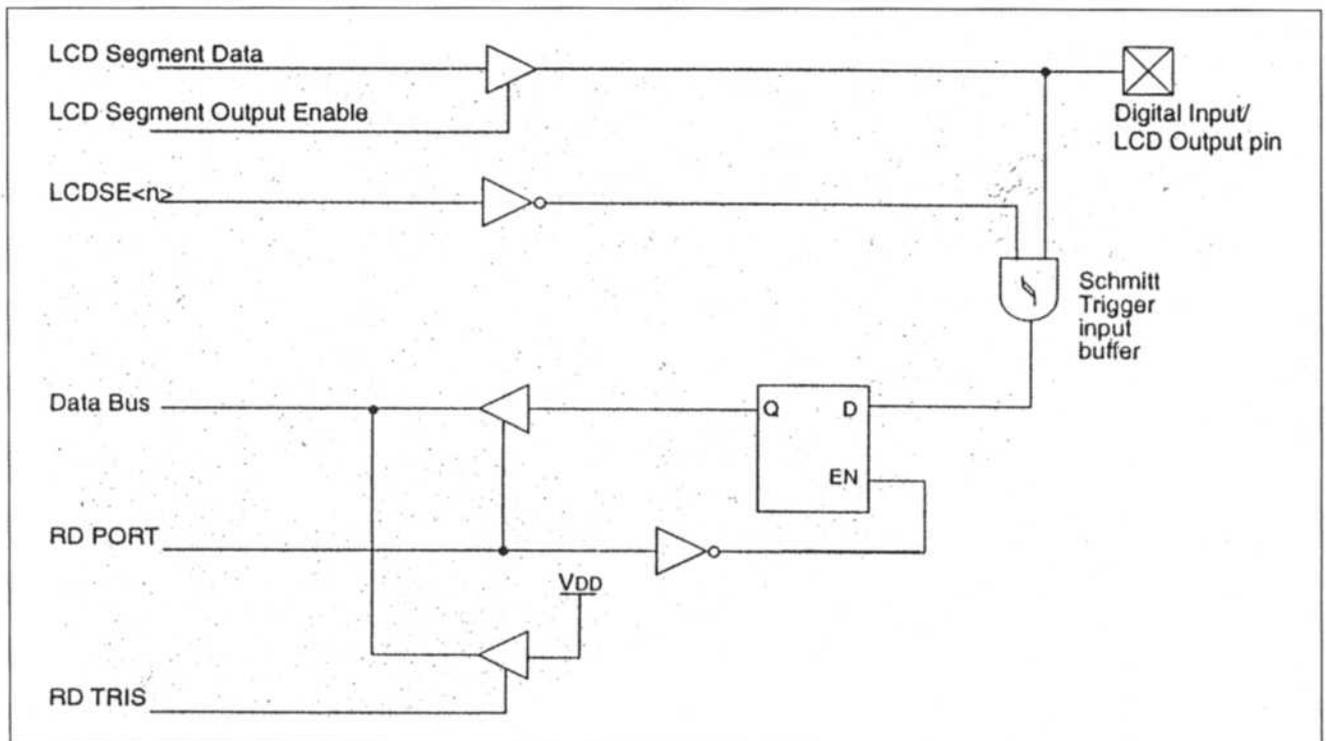
Il watchdog assorbe circa 6 µA, l'oscillatore per il pilotaggio degli LCD circa 6 µA, l'oscillatore del Timer1 circa 10 µA.

Il convertitore analogico/digitale, dove presente, assorbe circa 1 µA; ovviamente se non si trova nella fase di conversione.

Per adesso ci fermiamo dandoci appuntamento al prossimo mese dove andremo a vedere come funziona passo per passo la periferica che gestisce i display

continua

Figura 32. Diagramma a blocchi delle porte E, F e G utilizzabili solo come ingressi



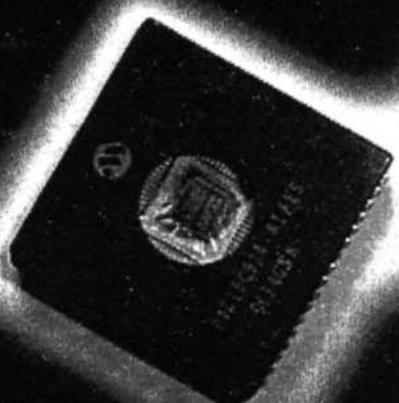
La famiglia PIC16C9xx



*Scopriamo insieme
quali sono le peculiarità
di questa famiglia di
microcontroller, l'unica
tra i prodotti Microchip
che gestisce
display LCD*

quinta parte

Paolo Pescioni



MICROCHIP
PIC16C923
-04/L
9637SAT

Nella puntata precedente si è visto come è strutturato un PIC della famiglia 16C9xx, a livello di memoria RAM, ROM, registri, periferiche varie, ecc. Adesso vedremo in dettaglio la periferica principale che contraddistingue questi microcontroller da tutti gli altri, ovvero la sezione che pilota qualsiasi display LCD a segmenti (ossia non quelli di tipo "intelligente" con a bordo un controllore, ma quelli simili ai display a 7 segmenti a led).

Possedere questo tipo di periferica non è banale: sul mercato non si trovano facilmente (C che abbiano a bordo la gestione di display LCD perché non semplice da implementare e comunque molto costosa rispetto al basso prezzo che deve mantenere un microcontroller.

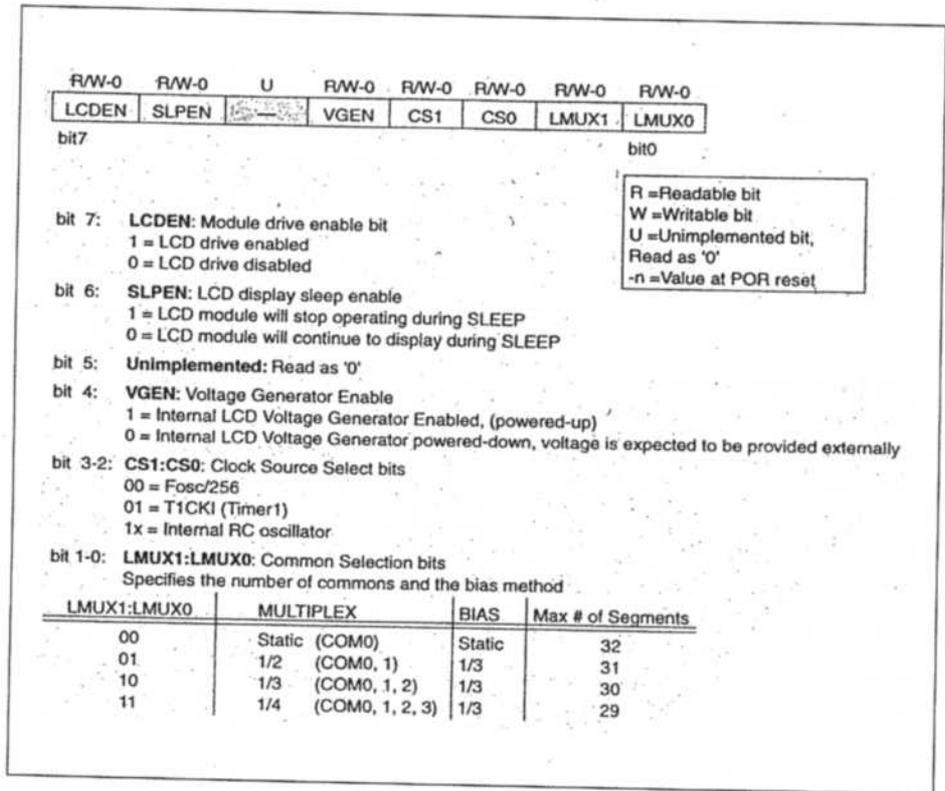
Le applicazioni più classiche che abbiamo trovato per questa famiglia sono quelle in campo automotive: sono presenti in diversi quadri di controllo di alcune vetture e di alcune motociclette di un certo valore, anche grazie alle altre periferiche incluse che abbiamo già visto come le seriali ed il convertitore analogico/digitale che viene impiegato per il

rilevamento di alcuni sensori come quello della benzina e della temperatura sia dell'acqua che dell'olio.

Ma applicazioni se ne trovano anche in tutti quei prodotti che devono costare poco, portatili, alimentati a batterie e che devono visualizzare su display diversi dati.

È importante sottolineare che con la gestione dei segmenti, è

Figura 33.
Il registro
LCDCON



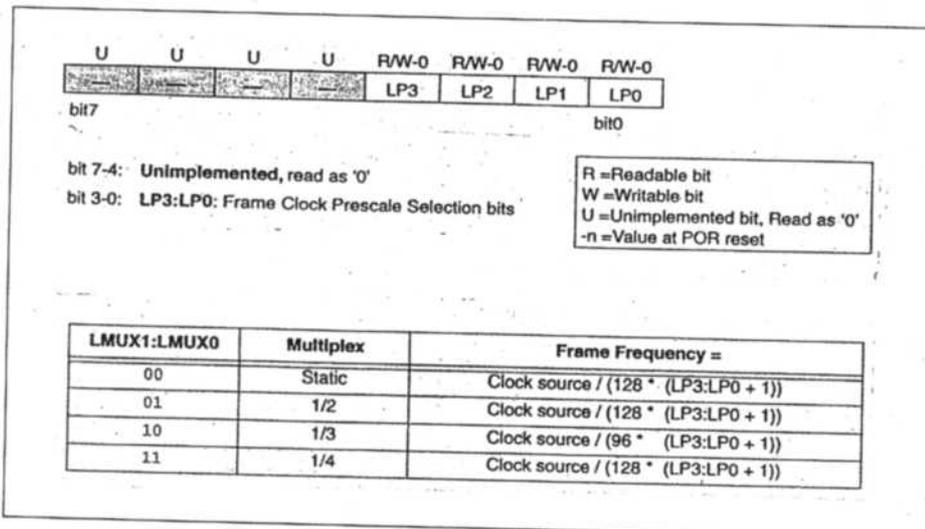
possibile anche pilotare display custom, dotati di segmenti sparsi su tutta la superficie e non allineati con la forma canonica del numero "8".

Veniamo allora alla descrizione completa della periferica in questione.

Il modulo LCD inserito nei PIC16C92X genera il segnale di clock con le corrette tempistiche per pilotare un segmento LCD sia statico che multiplexato (ovvero è possibile pilotare con continuità un solo segmento oppure più di un segmento con la tecnica della condivisione di tempo) fino a 32 segmenti con 4 comuni. Addirittura è previsto il controllo dei vari pixel.

L'interfacciamento al modulo LCD è supportato da tre registri dedicati (LCDCON, LCDSE e LCDPS) più altri sedici registri per la definizione dei dati (LCD00...LCD15) che rappresentano l'array dei pixel. All'avvio, l'informazione per

Figura 34.
Il registro
LCDPS



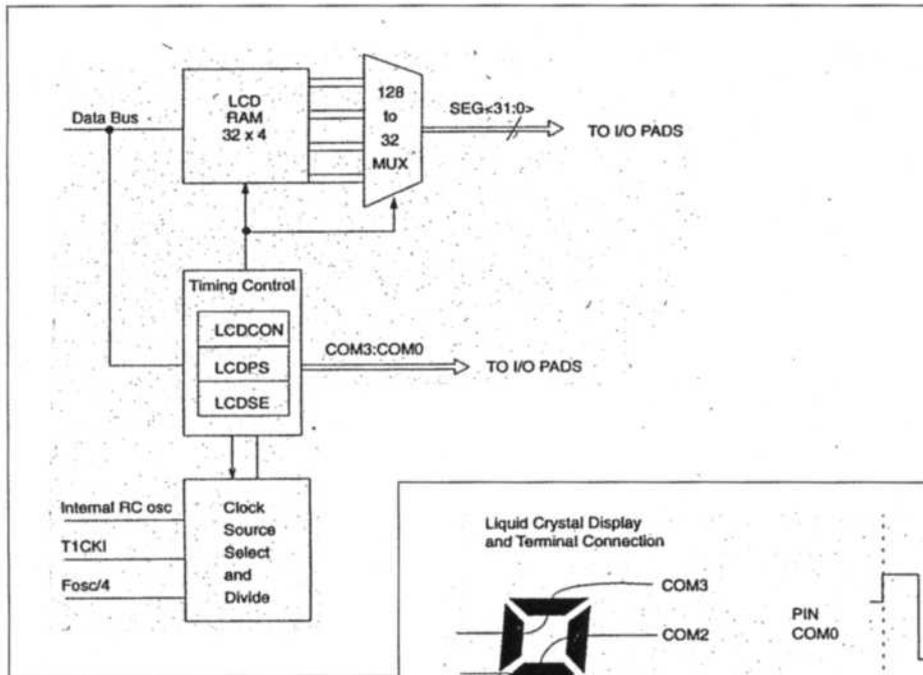


Figura 35. Diagramma a blocchi del modulo LCD

solo comune COM0: con questa soluzione il numero massimo di segmenti è ovviamente 32. Con la configurazione "01" invece si passa al multiplex su due comuni (COM0 e COM1) riducendo a 31 il numero massimo di segmenti. Con la configurazione "10" si mantiene il multiplex su tre comuni

l'inizializzazione del modulo consiste nel selezionare il numero di comuni richiesto dal display LCD e successivamente nello stabilire il clock rate per quel display.

Quando il modulo è stato inizializzato per quel determinato display LCD, i bit presenti nei registri LCD00...LCD15 vengono settati o azzerati a seconda che il corrispondente pixel debba essere rispettivamente visualizzato o meno.

Il registro LCDCON

Vediamo adesso quali sono le specifiche funzioni del registro LCDCON il cui significato è riportato in figura 33: i due bit meno significativi, ovvero i bit 0 e bit 1 decidono il sistema di pilotaggio dei segmenti.

Con la configurazione "00" si sceglie di comandare un segmento in modo statico con il

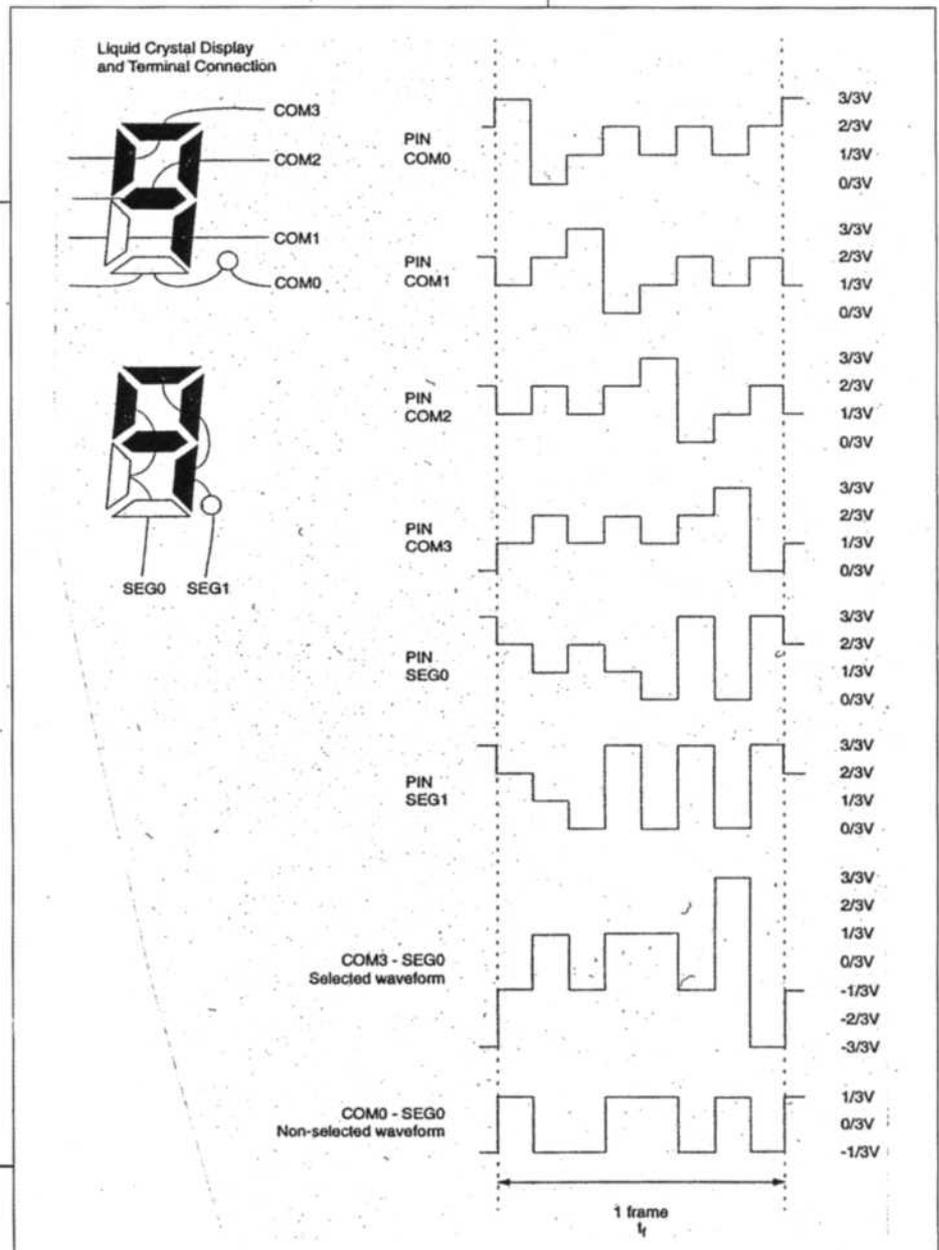


Figura 36. Forma d'onda per il pilotaggio con multiplex a 1/4

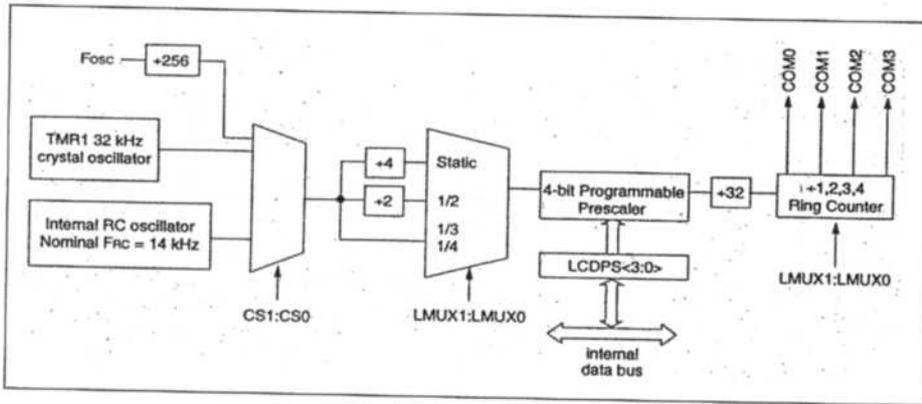


Figura 37. Diagramma a blocchi del generatore di clock

registri e successivamente ai segmenti) viene calcolata sulla base del valore precedentemente impostato con i due bit LMUX0 e LMUX1 del registro LCDCON e del valore impostato con i 4 bit LP0...LP3 di questo registro.

Per il calcolo della frequenza è necessario applicare una semplice formula illustrata nella figura 34: La sorgente di clock deve essere divisa per 128 e per il valore impostato da (LP0...LP3 + 1).

Vediamolo con un esempio: supponiamo di aver scelto la configurazione statica (LMUX0 e LMUX1 = "00").

Supponiamo inoltre di aver posto CS0 e CS1 a "00", ovvero di aver scelto la frequenza

(COM0, COM1 e COM2) riducendo a 30 il numero massimo di segmenti. Con la configurazione "11" infine, si porta il multiplex a quattro comuni (COM0, COM1, COM2 e COM4) riducendo a 29 il numero massimo di segmenti.

I successivi due bit (bit 2 e bit 3) stabiliscono la sorgente di clock: con la configurazione "00" si sfrutta il clock interno del microcontroller diviso per 256. Con la combinazione "01" invece, si impiega il Timer1 ed infine con la combinazione "10" oppure "11" si opta per l'oscillatore interno RC (dello stesso tipo usato per il watchdog e per il convertitore A/D).

Il bit 4 (VGEN = Voltage Generator ENable) indica alla periferica se la tensione per il pilotaggio dei segmenti LCD debba arrivare dal generatore di tensione interno (configurazione "1") oppure da un generatore esterno al micro (configurazione "0"). Il bit 5 non è stato ancora implementato.

Con il bit 6 (SLPEN = SLeep ENable) si dice al micro se far funzionare o meno la periferica LCD quando è in stato di SLEEP. Se posto a "0" continuerà a funzionare, viceversa si spegnerà. Il bit 7 (LCDEN = LCDENable) infine, abilita (se posto a "1") o disabilita (se posto a "0") il funzionamento di tutta la periferica.

Il registro LCDPS

In figura 34 troviamo il contenuto del registro LCDPS. Scopo di tale registro è di selezionare il tempo di clock per il pilotaggio dei segmenti LCD.

La frequenza dei frame (invece di tutti i pixel ai relativi

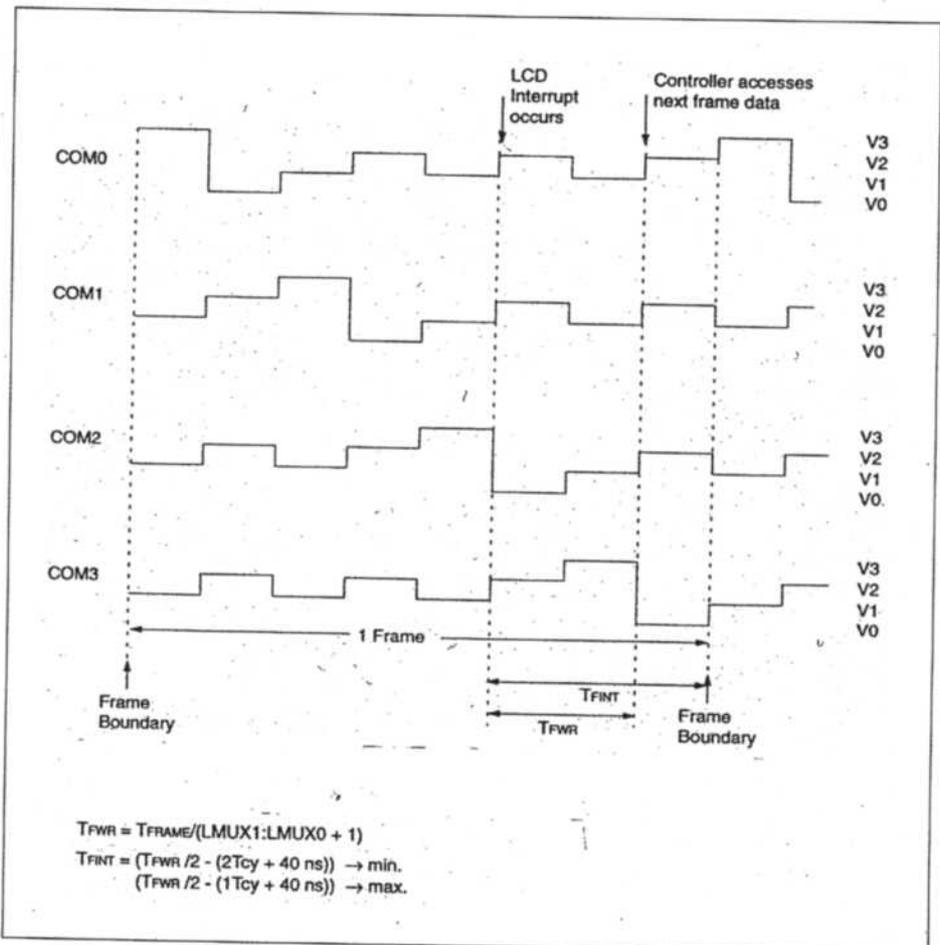


Figura 38. Forma d'onda nel multiplex a 1/4

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SEGs |
| COMc |

bit7 bit0

bit 7-0: **SEGsCOMc**: Pixel Data Bit for segment s and common c
 1 = Pixel on (dark)
 0 = Pixel off (clear)

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, Read as '0'
 -n = Value at POR reset

della sorgente di clock uguale a quella del quarzo diviso per 256 e che la frequenza del quarzo sia 4MHz. Allora la frequenza della sorgente di clock sarà di $4.000.000/256 = 15.625$ Hz. Supponiamo infine di aver posto LP3...LP0 a "0111".

Allora la frequenza dei frame sarà di $15.625/(128 * (7 + 1)) =$ circa a 15,5 Hz. Per comprendere meglio come lavora la periferica LCD, vediamo in figura 35 il diagramma a blocchi del modulo: in basso si trova la sezione che genera il clock necessario al funzionamento di tutto il sistema.

Poi sono presenti i tre registri sopra elencati ed infine abbiamo la memoria RAM dove vengono specificati valori di tutti i pixel da pilotare.

In figura 36 si può vedere a titolo d'esempio come viene pilotato un display con multiplex a 1/4: si nota subito l'obbligo di avere a disposizione 4 livelli di tensione esattamente ripartiti (0, 1/3, 2/3 e 1).

La generazione del clock

Cerchiamo di comprendere meglio come avviene la generazione del frame clock aiutandoci anche con il diagramma a blocchi di figura 37. Abbiamo già visto che la selezione del clock per la generazione delle tempistiche necessarie al modulo LCD

Tabella 4. Layout generico di un registro LCDD

Figura 39. Entrata e uscita dallo SLEEP con SLPEN=1 o CS1:CS0=00

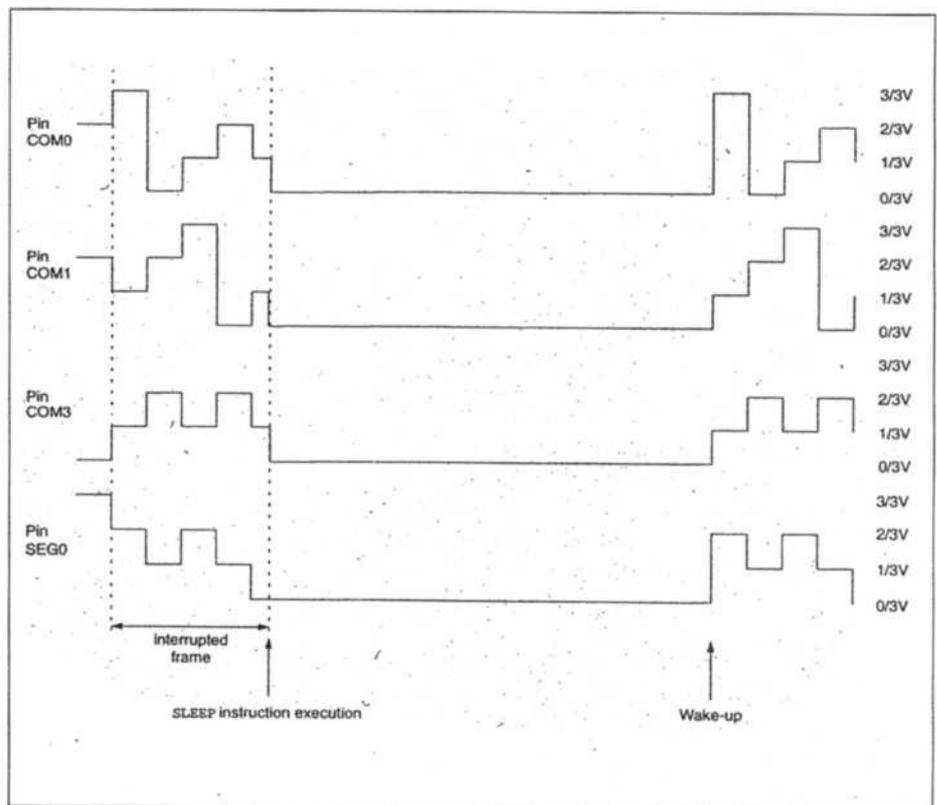
si imposta nel registro LCDCON e che è possibile avere l'oscillatore interno, l'oscillatore del Timer1 oppure la frequenza dell'oscillatore divisa per 256 come generatore principale.

Il generatore RC è un oscillatore RC interno che lavora ad una frequenza nominale di circa 14 KHz (in funzione della temperatura e della tensione di alimentazione). Con questo tipo di oscillatore è possibile mantenere l'LCD acceso anche quando il processore è in stato di SLEEP. È possibile disabilitare questo oscillatore o non

selezionandolo oppure disabilitando l'intero modulo LCD.

Il secondo generatore disponibile è quello ottenuto dal Timer1 abbinato ad un oscillatore esterno. Poiché è possibile applicare oscillatori a bassa frequenza (ad esempio 32,768 KHz) anche questo generatore si presta a far lavorare il modulo LCD anche quando il processore è in stato di SLEEP. Ci si deve ricordare però che per far funzionare tale oscillatore è necessario settare a "1" il bit 3 (T1OSCEN) del registro T1CON del PIC.

La terza sorgente è il clock di sistema diviso per 256. È stato scelto questo fattore di divisione per ottenere circa 15-30 kHz con oscillatori da 4-8 MHz. Il fattore di divisione è fisso e non programmabile. È possibile invece, come abbiamo già visto, agire sul registro LCDPS per variare il clock rate del frame.



| R/W-1 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SE29 | SE27 | SE20 | SE16 | SE12 | SE9 | SE5 | SE0 |

bit7 bit0

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, Read as '0'
 -n = Value at POR reset

bit 7: SE29: Pin function select RD7/COM1/SEG31 - RD5/COM3/SEG29
 1 = pins have LCD drive function
 0 = pins have digital Input function
 The LMUX1:LMUX0 setting takes precedence over the LCDSE register.

bit 6: SE27: Pin function select RG7/SEG28 and RE7/SEG27
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 5: SE20: Pin function select RG6/SEG26 - RG0/SEG20
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 4: SE16: Pin function select RF7/SEG19 - RF4/SEG16
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 3: SE12: Pin function select RF3/SEG15 - RF0/SEG12
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 2: SE9: Pin function select RE6/SEG11 - RE4/SEG09
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 1: SE5: Pin function select RE3/SEG08 - RE0/SEG05
 1 = pins have LCD drive function
 0 = pins have digital Input function

bit 0: SE0: Pin function select RD4/SEG04 - RD0/SEG00
 1 = pins have LCD drive function
 0 = pins have digital I/O function

bit in un registro LCDxx e il rispettivo comune e segmento.

Ogni bit o registro inutilizzato per la gestione del display può essere utilizzato come una comune locazione di memoria RAM.

Operazioni durante lo SLEEP

Si è già accennato al fatto che il modulo LCD lavora anche con il microcontroller in stato di SLEEP. La selezione è controllata con il bit SLPEN del registro LCDCON. Settare tale bit consente al modulo LCD di entrare in modalità SLEEP mentre azzerarlo blocca il funzionamento del modulo.

Se viene eseguita un'istruzione di SLEEP e SLPEN è uguale a 1, il modulo LCD cessa di funzionare ed entra in uno stato di bassissimo assorbimento. La cessazione è immediata e sui segmenti verrà inviata la più bassa tensione possibile.

La figura 39 mostra questa situazione.

Per essere sicuri che il modulo LCD completi il frame, l'istruzione di SLEEP dovrebbe essere eseguita immediatamente dopo la fine di un frame. In questi casi è utilissimo l'interrupt visto precedentemente.

LCD Interrupt e pixel control

Il clock analizzato precedentemente produce un interrupt che definisce il tempo dei frame. Tale interrupt può essere utilizzato per coordinare la scrittura dei pixel con l'invio di un nuovo frame. La scrittura coordinata dei pixel porta ad una migliore brillantezza del segmento.

Sempre lo stesso interrupt può essere impiegato anche per sincronizzare eventuali eventi esterni al display (fine di una conversione A/D, arrivo di un segnale di rinfresco, ecc).

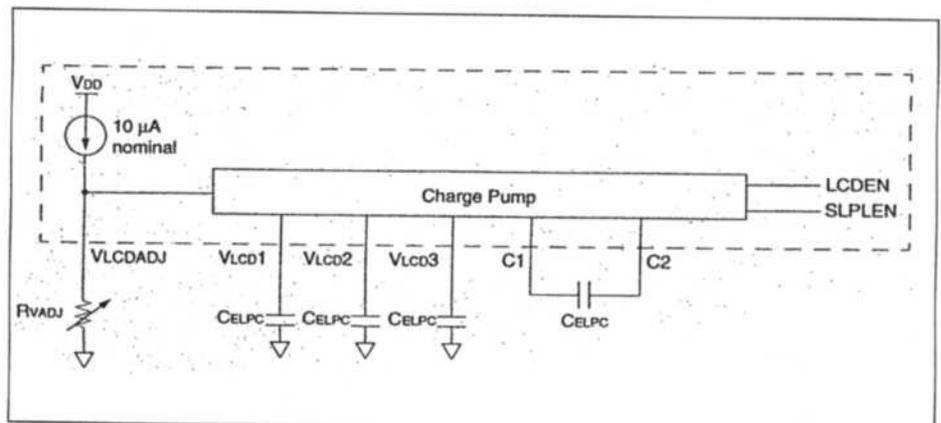
Un nuovo frame inizia al primo fronte del segnale COM0. L'interrupt viene immediatamente settato dopo che il controller LCD ha completato l'accesso a tutti i dati dei pixel richiesti per quel frame.

Queste transazioni sono visibili in figura 38. Il controller LCD inizierà l'accesso ai dati per il frame successivo dopo un tempo T_{fwr} successivo all'interrupt.

Abbiamo accennato che i registri LCDxx contengono le informazioni relative ai pixel da visualizzare. Ogni bit definisce un unico pixel. La tabella 4 mostra la correlazione tra ogni

Figura 40. Il registro LCDSE

Figura 41. Diagramma a blocchi del generatore di tensione



Se l'istruzione SLEEP viene eseguita quando SLPEN vale 0, il modulo continuerà a visualizzare il contenuto attuale dei registri LCDxx. Per consentire al modulo di funzionare però, ci si deve ricordare che la sorgente di clock non dovrà essere abbinata al clock del sistema ma all'oscillatore RC o al Timer1.

Quando il processore è in SLEEP, i dati sul display non potranno essere cambiati ed il consumo non si abbasserà al minimo.

Il registro LCDSE

L'ultimo registro che ci rimane da vedere è il LCDSE (LCD Segment Enable).

Questo registro, visibile in figura 40, è usato per selezionare la funzione dei pin a gruppi. Tale selezione permette ad ogni gruppo di pin di operare o come driver per LCD o come pin digitali. Per configurare un pin come porta digitale è necessario azzerare il bit corrispondente nel registro

Esempio 1:

```
BCF STATUS,RP0 ; Selezione banco 2
BSF STATUS,RP1 ;
BCF LCDCON,LMUX1 ; Selezione mux statico
BCF LCDCON,LMUX0 ;
MOVLW 0xff ; Porta D, E, F, G come pin LCD
MOVWF LCDSE ;
.... ; Configurazione delle altre funzioni
```

Esempio 2:

```
BCF STATUS,RP0 ; Selezione banco 2
BSF STATUS,RP1 ;
BSF LCDCON,LMUX1 ; Selezione 1/3 mux
BCF LCDCON,LMUX0 ;
MOVLW 0x87 ; Porta D<7:0> e E<6:0> come pin LCD
MOVWF LCDSE ;
.... ; Configurazione delle altre funzioni
```

LCDSE. Quando ciò avviene, la direzione del pin viene stabilita dal corrispondente registro TRISx.

Il registro LCDSE ha il controllo sui registri TRISx: se un bit viene settato in LCDSE, non viene più preso in considerazione il bit del registro TRISx in quanto è stato fatto un assegnamento del pin al modulo LCD.

Figura 42.
Connessione di resistori esterni al generatore di tensione

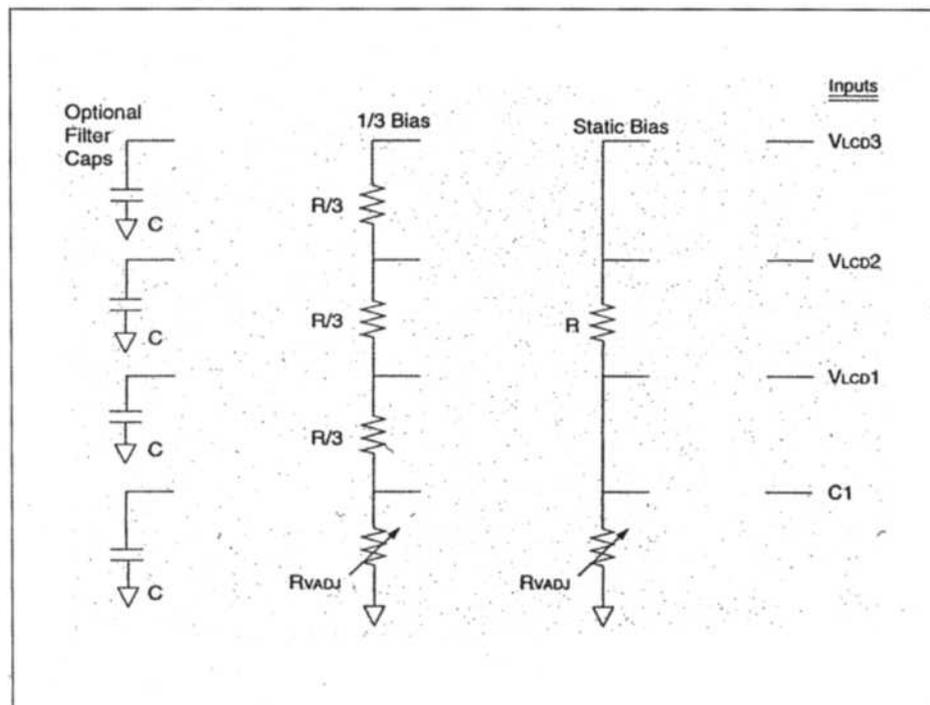
Sopra riportiamo due esempi per la configurazione rispettivamente di multiplex statico a 32 segmenti e di multiplex a 1/3 con 13 segmenti.

Il generatore di tensione

Ci sono due modi per generare la tensione di riferimento per i segmenti: il Charge Pump interno e la scala resistiva esterna. In figura 41 vediamo il Charge Pump. Questo regolatore è aggiustabile in range connettendo una resistenza esterna variabile da VLCDADJ verso massa. Il potenziometro servirà a regolare il contrasto per l'LCD. Questa tensione base è connessa a VLcd1 sul Charge Pump. Quando il Charge Pump non è attivo, VLcd3 viene portato al positivo di alimentazione.

È possibile però usare anche una scala di resistenze pesate. In figura 42 sono mostrate le connessioni esterne per il bias statico e per il multiplex a 1/3. Il bit VGEN del registro LCDCON deve essere azzerato per usare questo sistema.

continua



La famiglia PIC17C4x

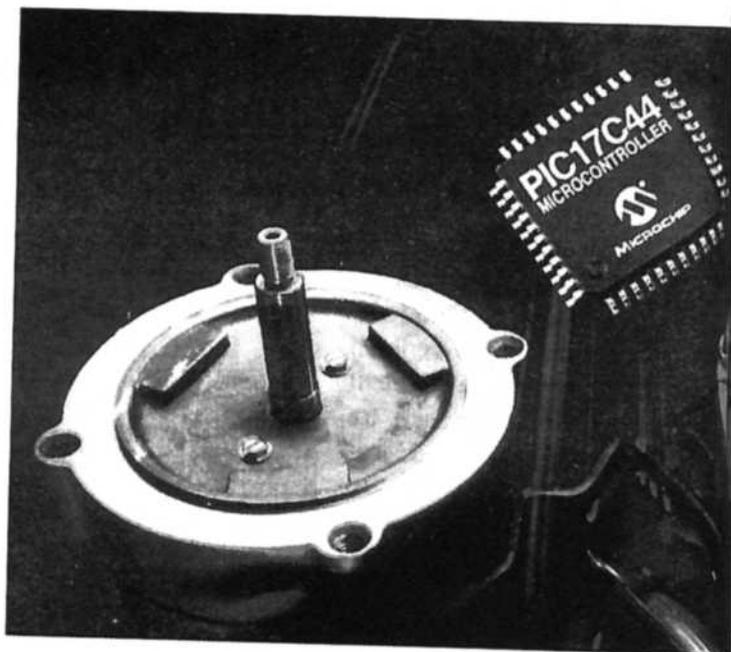


Questo mese prendiamo in esame la famiglia più completa e quindi anche più complessa che la Microchip offre nel campo dei microcontroller, anche se questo termine non è applicabile completamente a questi chip, in quanto viene meno una delle regole fondamentali dei microcontroller: la memoria di programma può non essere interna al chip, ma indirizzata esternamente.

Per le caratteristiche che andremo a vedere tra poco, la famiglia 17C4x si pone in competizione perfino con alcuni microprocessori abbastanza evoluti, e sempre più frequentemente viene preferita a quest'ultimi per il costo e la massima flessibilità d'uso sia hardware che software.

A questo proposito, nel corso della presentazione, vedremo che tutti i chip della famiglia 17Cxxx hanno un numero maggiore di

Scopriamo insieme quali sono le peculiarità di questa famiglia di microcontroller, l'unica tra i prodotti Microchip dotata di indirizzabilità di memoria esterna, di due periferiche seriali completamente indipendenti e di un set di istruzioni esteso pur rimanendo sempre nella categoria dei microcontrollori RISC



Parte sesta

istruzioni rispetto a tutte le altre famiglie, per velocizzare ulteriormente il programma da eseguire.

È opportuno accennare al fatto che, essendo questi chip molto più complessi degli altri, è praticamente quasi impossibile gestirli senza il necessario sistema di sviluppo: mentre con le famiglie 12Cxx o 16Cxx si poteva anche tentare di realizzare un programma funzionante utilizzando i modelli finestrati (quelli cioè che si cancellano con la luce a raggi ultravioletti), con questi nuovi prodotti è quasi obbligatorio l'impiego di un emulatore in tempo reale.

La ragione di ciò si vedrà chiaramente in seguito quando andremo ad analizzare la famiglia 17C4x in dettaglio: le periferiche a disposizione sono molte di più rispetto ai precedenti modelli e quindi anche la loro gestione sarà ovviamente più complessa.

Tanto per dare un'idea della maggiore complessità, diciamo che qui le seriali attivabili non sono soltanto una, ma possono essere anche due funzionanti in contemporanea!

Mentre quindi con i PIC più "piccoli" era sufficiente un finestrato ed una buona dose di pazienza, con la famiglia 17C4x si dovrà prendere in esame l'impiego di un buon emulatore real-time.

Apriamo qui una parentesi per rendere noto che emulatori per PIC sono disponibili sia presso la Microchip stessa, sia presso un buon rivenditore di terze parti, ovvero di accessori testati direttamente dalla Microchip prima di essere messi in commercio.

La differenza sostanziale dei prodotti Microchip contro quelli forniti da terze parti è che, ad oggi, non esiste una scadenza di garanzia, ovvero qualsiasi sistema venduto da Microchip, se dovesse guastarsi, viene riparato gratuitamente indipendentemente dalla data di acquisto e dal tipo di guasto.

Chiaramente questo stato di privilegio ha un costo che spesso è doppio rispetto a quello dei prodotti forniti dalle terze parti.

Ad ogni modo, qualunque sia il sistema di sviluppo scelto anche tra le terze parti, potrete essere sicuri che il prodotto sia stato testato direttamente dalla Microchip e che soddisfa determinati requisiti hardware e software.

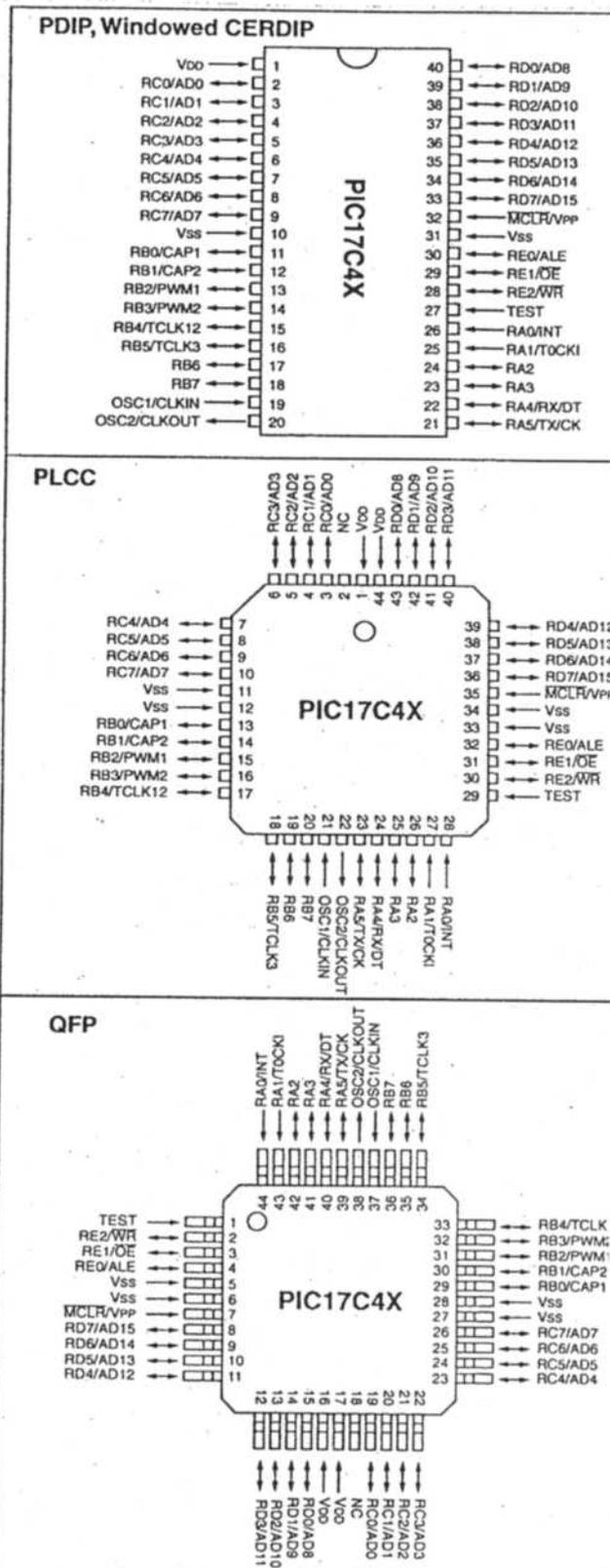
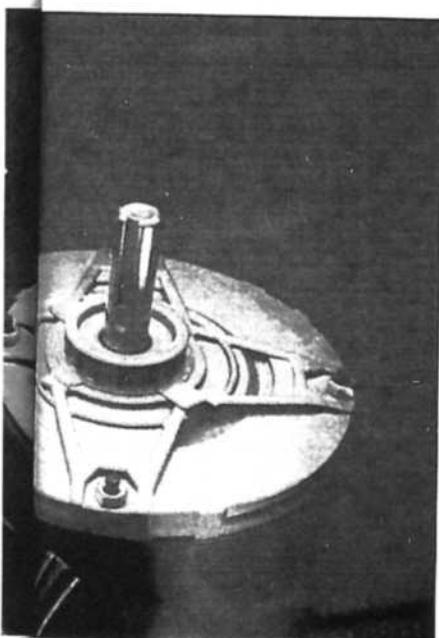


Figura 43. Piedinature disponibili per la famiglia 17C4x



Parlando quindi anche di software, anche per questi chip è disponibile il linguaggio "C", che al contrario delle altre famiglie, qui può essere ben sfruttato e sarà possibile ottenere degli ottimi risultati: nelle famiglie più piccole infatti, l'impiego del compilatore "C" è quasi sconsigliabile, poiché per quanto ottimizzato possa essere, non riuscirà mai a sfruttare in modo ottimale (come fa ad esempio un

buon programmatore in linguaggio assembler) le risorse del chip (memoria di programma e memoria RAM). Considerando invece che la famiglia 17C4x ha a disposizione, come vedremo, molta memoria sia RAM che di programma, la scelta del linguaggio "C" può essere adottata per una migliore visibilità del programma e quindi anche per la facilità, in tempi successivi allo sviluppo, di eseguire eventuali modifiche.

Oltre al linguaggio "C" è disponibile anche la logica Fuzzy per realizzazioni di controlli nel campo del continuo utilizzando variabili discrete. In pratica si tratta di ridurre, con formule matematiche, il numero di variabili di un determinato problema, facendo nascere più gruppi per variabili omogenee.

Alcune ditte americane, hanno recentemente prodotto anche altri compilatori per i PIC, relativi a

linguaggi "notoriamente conosciuti" come il BASIC o meno conosciuti come il FORTRAN. A nostro parere comunque, il linguaggio assembler è l'unico che consente di programmare il chip consentendoci sempre ed esattamente di sapere come internamente verrà svolta l'istruzione inviata: per fare un esempio, pensiamo di dover eseguire un loop di N volte in cui viene testato un determinato pin di ingresso.

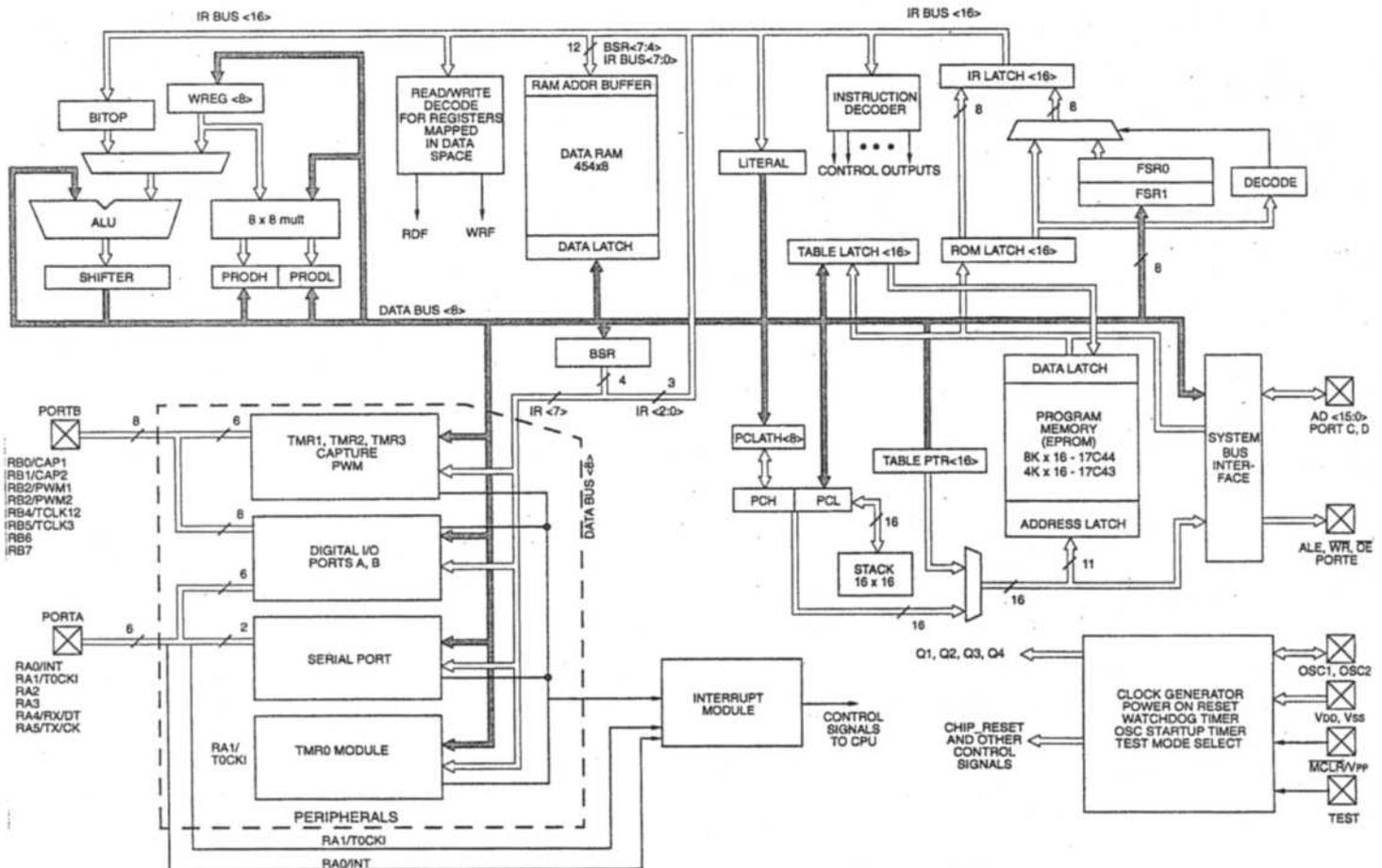


Figura 44. Architettura dei 17C4x

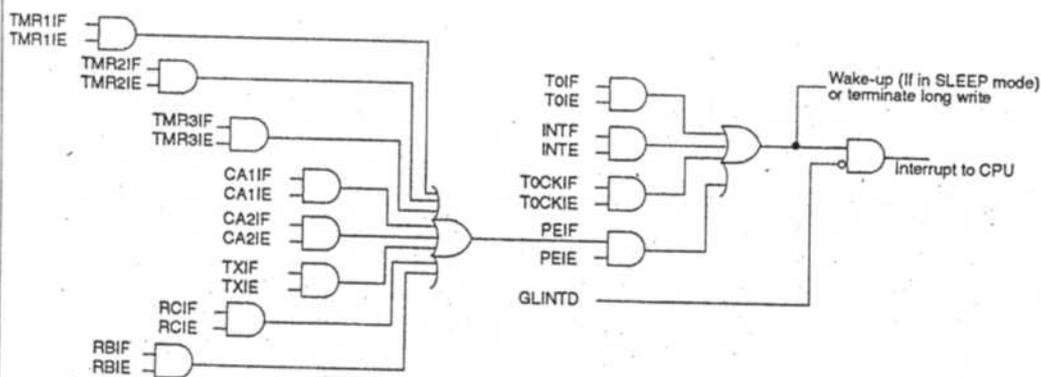


Figura 45. Diagramma degli interrupt

Le caratteristiche generali

Per prima cosa, vediamo che anche i chip della famiglia 17C4x sono disponibili in tutti i package possibili, ovvero in PDIP (OTP, ROM e finestrato), in PLCC ed in QFP, come si può notare in figura 43.

Il numero minimo è di 40 pin, esattamente come il modello 16C74, ma con un "cervello" decisamente molto più potente.

Il numero di istruzioni è aumentato a 58 ed anche per loro vale l'esecuzione in un ciclo macchina eccetto che per quelle di salto che ne richiedono due.

In assembler sappiamo esattamente come realizzarlo e quali saranno le risorse impegnate (i registri di conteggio, le istruzioni adoperate e, ultimo ma non meno importante, l'algoritmo impiegato) mentre in un qualsiasi altro linguaggio chi ci dice quale registro verrà usato e quale tipo di algoritmo verrà implementato?

Probabilmente per molti non avrà importanza il tipo di scelta, ma se si vuole ottenere il massimo dal microcontrollore, l'unica soluzione valida è la programmazione in assembler.

Per le caratteristiche che tra poco andremo a vedere, le applicazioni dei chip 17C4x sono generalmente diverse da quelle degli altri microcontrollori Microchip.

Data la loro velocità di esecuzione e l'istruzione di moltiplicazione hardware eseguibile in un ciclo macchina, è possibile impiegarli come dei veri e propri DSP (chiarmente per

frequenze non molto elevate): si trovano sul sito internet della Microchip (www.microchip.com) applicazioni in tal senso. Ma iniziamo a valutare le

caratteristiche di questa nuova famiglia, che è riuscita a conservare ed ampliare tutte le features positive delle altre, migliorandole pian piano.

Tabella 5. Confronto tra i chip della stessa famiglia

	PIC17C42	PIC17C43	PIC17C44
Maximum Frequency of Operation	25 MHz	25 MHz	25 MHz
Operating Voltage Range	4.5 - 5.5 V	2.5 - 6.0 V	2.5 - 6.0 V
On-chip Program Memory (16-bits wide)	2K	4K	8K
Data Memory (bytes)	232	454	454
Hardware Multiplier (8 x 8)	No	Yes	Yes
Timer0 (16-bit + 8-bit postscaler)	Yes	Yes	Yes
Timer1 (8-bit)	Yes	Yes	Yes
Timer2 (8-bit)	Yes	Yes	Yes
Timer3 (16-bit)	Yes	Yes	Yes
Capture inputs (16-bit)	2	2	2
PWM outputs (up to 10-bit)	2	2	2
Serial Communications Interface (SCI/USART)	Yes	Yes	Yes
Power-On Reset	Yes	Yes	Yes
Watchdog Timer	Yes	Yes	Yes
External Interrupts	Yes	Yes	Yes
Interrupt Sources	11	11	11
Program Memory Code Protect	Yes	Yes	Yes
I/O	33	33	33
I/O High Current Capability	Source	25 mA	25 mA
	Sink	25 mA ²	25 mA ²
Package Types		40-Pin DIP,	40-Pin DIP,
		44-pin PLCC	44-pin PLCC
		44-pin MQFP	44-pin TQFP

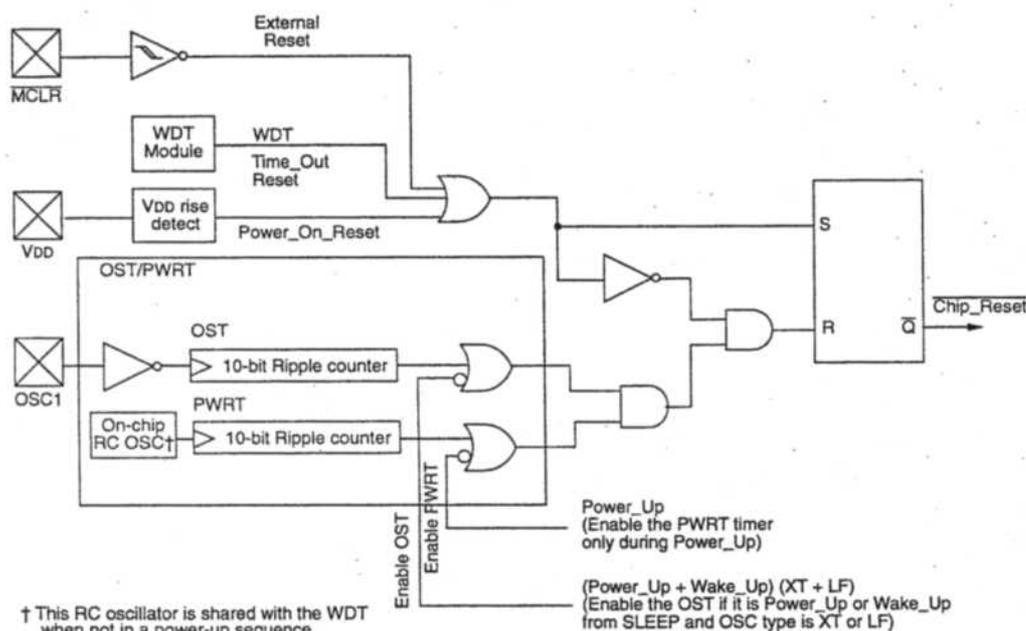


Figura 46. Diagramma del reset

La velocità di funzionamento è salita a 25MHz, per arrivare all'esecuzione di un'istruzione in soli 160ns.

Ma vediamo in tabella numero 5 le voci più interessanti: la tensione di alimentazione può variare tra 2,5 e 6 volt. La memoria RAM va da un minimo di 232 byte ad un massimo di 454 byte mentre quella di programma interna va da 2K a 8K word (16 bit).

Una tra le peculiarità di questa famiglia è l'avere a bordo un moltiplicatore 8x8 hardware, ovvero che esegue una moltiplicazione di due byte e pone il risultato di 2 byte in due registri in un solo ciclo macchina!

I timer invece sono ben 4. Il timer 0 è a 16 bit con un postscaler a 8 bit.

Il timer1 ed il timer2 sono entrambi ad 8 bit mentre il timer3 è a 16 bit.

Sono disponibili due input per la funzione CAPTURE identica alla famiglia 16C7x e due output per la generazione di segnali PWM con risoluzione fino a 10 bit. Si ritrova inoltre lo stesso modulo seriale sincrono/asincrono già visto nella famiglia 16C7x. Vedremo successivamente che ci sono chip della famiglia 17Cxxx che hanno a bordo ben due seriali e stadio convertitore A/D.

I livelli di stack sono passati da 8 a sedici mentre per gli interrupt ci sono più vettori con diverse priorità.

Lo spazio di indirizzabilità esterno è di 64Kx16. Si deve però sottolineare che quando il chip lavora con la memoria esterna, ovviamente non è possibile proteggere il programma da letture indesiderate. Sono poi disponibili due pin open-collector che riescono a pilotare carichi fino a 12 volt e 60 mA.

L'architettura interna

L'architettura dei 17C4x è derivata da quella Harvard, come per le altre famiglie, ovvero prevede due bus separati per i dati e per il programma, con conseguente separazione anche dei due banchi di memoria.

In figura 44 è visibile lo schema a blocchi dell'interno di un chip 17C4x.

Il bus delle istruzioni viene detto IR BUS (Instruction Register bus) ed è a 16 bit, mentre il bus dei dati viene detto DATA BUS ed è a 8 bit.

La RAM è interamente interna al micro ed ha un registro di indirizzamento ed uno di latch.

Il Working register è sempre detto "W" ed è anch'esso ad 8 bit.

Il modulo per le moltiplicazioni si chiama 8x8mult e si interfaccia con W e con il DATA BUS tramite due registri: il PRODH ed il PRODL, rispettivamente la parte alta e la parte bassa del risultato della moltiplicazione. In questa famiglia i File Select Register sono due invece di uno, molto comodi per indirizzare due aree diverse della memoria RAM.

Tabella 6. Interrupt: indirizzi e priorità

Address	Vector	Priority
0008h	External Interrupt on RA0/INT pin (INTF)	1 (Highest)
0010h	TMR0 overflow interrupt (TOIF)	2
0018h	External Interrupt on TOCKI (TOCKIF)	3
0020h	Peripherals (PEIF)	4 (Lowest)

Lo stack è profondo 16 livelli, cioè è possibile anidare fino a 16 chiamate a subroutine contro le 8 delle altre famiglie.

Si ricorda comunque che anche l'entrata in esecuzione di un interrupt viene considerata come una qualunque chiamata ad una subroutine.

A proposito di interrupt, in figura 45 si può vedere lo schema logico di generazione di interrupt: tutte le periferiche sono connesse da un OR logico, ma abilitate sia indipendentemente che complessivamente.

Le altre sorgenti sono anch'esse connesse in OR logico e l'uscita interrupt è divisa in due parti. Una serve per risvegliare la CPU se in SLEEP, l'altra ha la classica funzione dell'interrupt.

La memoria di programma può essere interna, interna/esterna o esterna, a seconda del tipo di funzionalità desiderato.

Per l'interfacciamento con memoria esterna, è disponibile un modulo che si chiama SYSTEM BUS INTERFACE e che si occupa di gestire indirizzi esterni, dati, segnali di controllo come l'ALE (Address Latch Enable), il WR (Write), l'OE (Output Enable).

Il modulo di generazione del clock si incarica anche di far funzionare il Power-on-reset, il timer del Watchdog e l'oscillatore di startup.

In figura 46 si vede

molto bene come lavora logicamente il reset del chip.

Questo può avvenire per i seguenti motivi: reset esterno dal pin MCLR (Master CLear), reset interno dal timer del Watchdog o dal rilevatore di tensione di alimentazione, reset interno dal modulo di

Power-up o dal modulo di Wake-up.

Un'altra novità è rappresentata dalla possibilità di gestire molto facilmente tabelle con istruzioni apposite e con due registri dedicati detti TABLEPTR (Puntatore di tabella) a 16 bit e TABLELATCH (Latch di tabella).

Si notano poi le varie periferiche disponibili in queste versioni di chip e cioè i tre timer per il CAPTURE ed il PWM, le porte di I/O, la porta seriale sincrona/asincrona ed il timer0.

L'organizzazione della memoria di programma

Nella famiglia 17C4x ci sono due aree di memoria distinte ed ognuna con il proprio bus: l'area della memoria di programma e l'area della memoria dei dati.

Abbiamo detto che il chip ha un program-counter di 16 bit quindi capace di indirizzare 64K locazioni di memoria.

Il vettore di reset è all'indirizzo 0000h, mentre i vettori di interruzione sono agli indirizzi 0008h, 0010h, 0018h e 0020h, come si vede in figura 47.

All'indirizzo 0008h si trova il vettore relativo all'interrupt generato esternamente sul pin INT.

All'indirizzo 0010h invece abbiamo il vettore relativo al timer0 ed all'indirizzo 0018h quello relativo al pin TOCKI.

Infine all'indirizzo 0020h si trova il vettore di interrupt relativo alle periferiche.

La soluzione di poter gestire diversamente i vari interrupt, permette di risparmiare il tempo necessario all'interrogazione software.

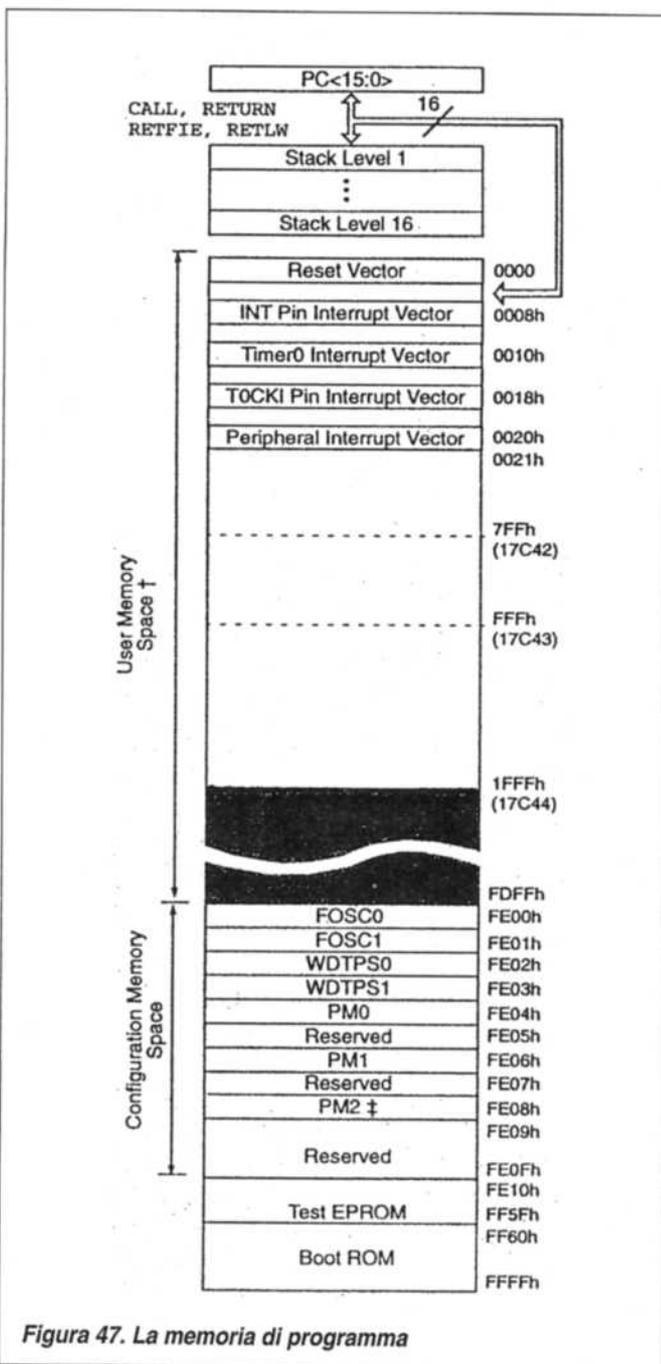
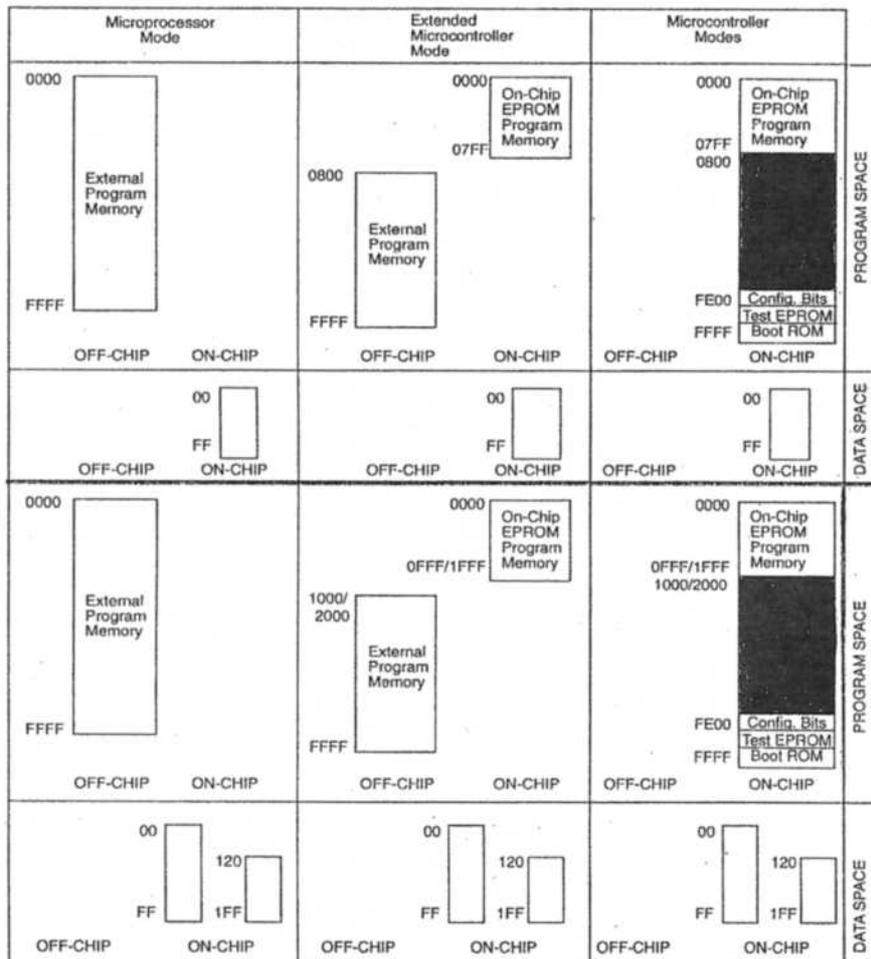


Figura 47. La memoria di programma

17C42

17C43/
17C44



Il chip gestisce automaticamente lo scambio di memoria commutando i due banchi.

Il modo microprocessore permette il solo utilizzo della memoria esterna, ignorando la memoria interna.

Nelle locazioni finali della memoria di programma sono inserite le informazioni riguardanti la configurazione del chip, come il modo di funzionamento, l'oscillatore, il watchdog, la modalità di test e la ROM di boot (che indica al chip da dove iniziare ad eseguire la prima istruzione).

In figura 48 è possibile individuare le possibili mappe della memoria in funzione del modo di funzionamento.

Quando il chip viene selezionato per il modo microprocessore o microcontrollore esteso, le porte "C", "D" ed "E" vengono configurate come SYSTEM BUS.

Inoltre ogni interrupt ha una diversa priorità: in tabella 6 se ne vede il valore.

I 17C4x possono operare in 4 modi possibili: microprocessori, microcontrollori, microcontrollori estesi e microcontrollori protetti.

Il secondo ed il quarto modo permettono solo istruzioni in memoria interna.

Il modo esteso consente di accedere sia alla memoria interna che a quella esterna.

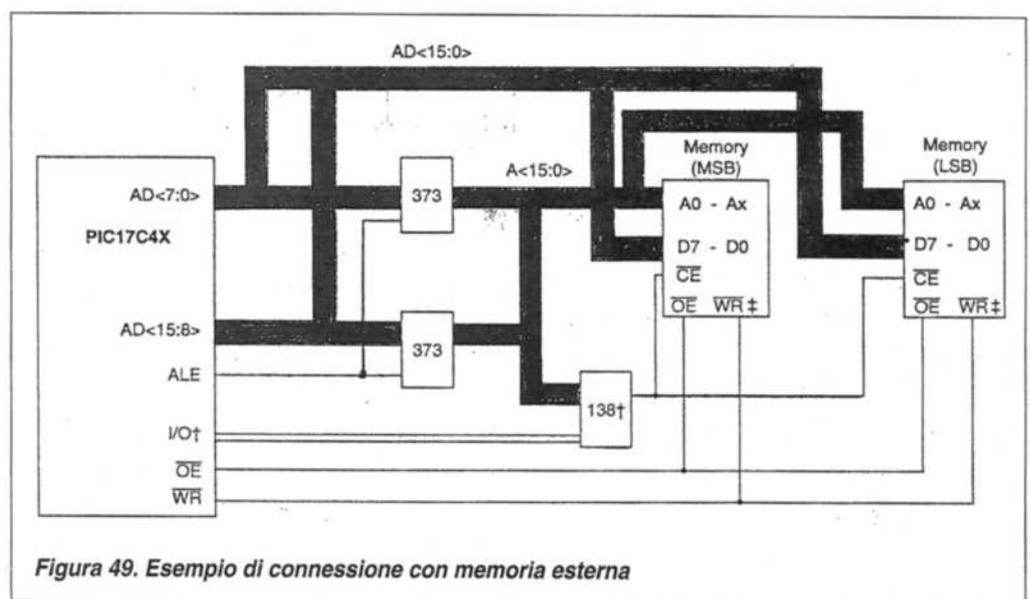


Figura 49. Esempio di connessione con memoria esterna

Le porte "C" e "D" vengono multiplexate come indirizzi/dati, mentre la porta "E" serve per i segnali di controllo.

Per demultiplexare i dati dagli indirizzi, sono necessari dei componenti esterni al chip. In figura 49 vediamo come questo può essere fatto: sono utilizzati dei latch 74xx373 ed un decoder 74xx138. Ovviamente, con questo tipo di soluzione, oltre a quanto già detto, si perde la velocità di esecuzione del programma, essendo necessario passare sullo stesso bus prima gli indirizzi e poi i dati (in pratica si torna ad una architettura di tipo Von Neumann).

La memoria dei dati

La memoria RAM è a sua volta sezionata in due parti. La prima si chiama GPR (General Purpose Register) e contiene i registri di uso generale. La seconda si chiama SFR (Special Function Register) e contiene tutti i registri cosiddetti "speciali".

Per entrambe queste aree, si ha una ripartizione in banchi per consentire una maggiore indirizzabilità di memoria. Tale ripartizione richiede l'uso di bit di controllo locati nel registro BSR (Bank Select Register). Se si tenta un accesso al di fuori della regione suddivisa in banchi (ad esempio da 00h a 0fh) il contenuto del BSR viene ignorato.

Per consentire scambi di dati tra le varie zone sono state aggiunte le due istruzioni MOVPF e MOVFP dove P è compreso nel range 01h-1Fh e F è compreso nel range 00h-FFh. Il range di P ha 8 locazioni in più rispetto ai registri periferici. Questo può essere utile in quelle applicazioni in cui alcune variabili devono essere copiate in altre locazioni di RAM ad uso generico come per esempio nel caso in cui si voglia trasferire il

contenuto del timer o di una porta di I/O in una locazione della memoria dati.

L'intera memoria dati può essere acceduta sia direttamente che indirettamente attraverso i due FSR (File Select Register). Per facilitare lo switching tra i banchi di memoria nell'area GPR, è stata aggiunta l'istruzione MOVLR bank. Questa istruzione copia la costante presente nel Working Register W nei 4 bit MSB del registro

BSR. Il nibble LSB invece viene lasciato inalterato. La MOVLR non è applicabile ai 17C42.

Nell'area SFR, è possibile individuare due gruppi di registri: quelli associati al funzionamento del "core" del chip e quelli associati alle varie periferiche. I primi risiedono nella zona non soggetta a "banchi", i secondi invece vengono suddivisi in banchi.

-continua-

Paolo Pescioni

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 ¹	Bank 2 ¹	Bank 3 ¹
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h				
19h				
1Ah				
1Fh				
20h				
	General Purpose RAM			
FFh				

Figura 50-a. Mappa dei registri nei 17C42

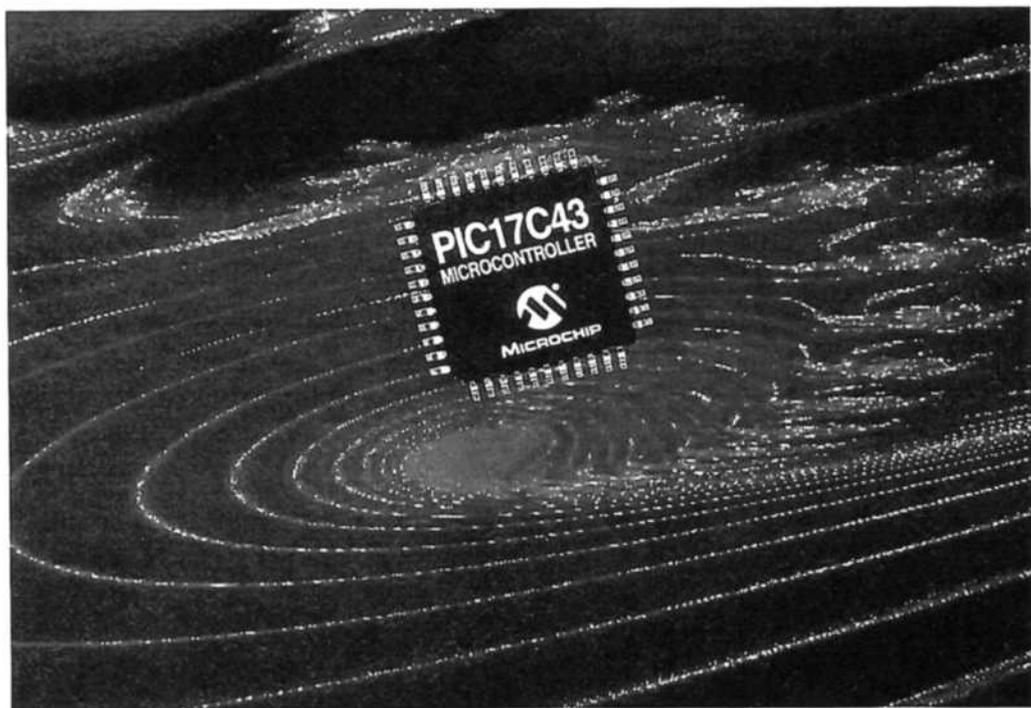
Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 ¹	Bank 2 ¹	Bank 3 ¹
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h	PRODL			
19h	PRODH			
1Ah				
1Fh				
20h				
	General Purpose RAM ²	General Purpose RAM ²		
FFh				

Figura 50-b. Mappa dei registri nei 17C43/4

La famiglia PIC17C4x



*Con la nuova famiglia
17C4x è adesso
possibile implementare
tabelle ed eseguire
moltiplicazioni
hardware con
istruzioni dedicate.
Vediamo come*



Settima parte

Indubbiamente la famiglia 17 dei microcontrollori PIC offre delle novità piuttosto consistenti per il programmatore, come la possibilità di collegare memoria di programma esterna o di spaziare tra un numero più elevato di istruzioni. Tra queste istruzioni ve ne sono alcune che sicuramente allieviscono la programmazione, snellendola e riducendo il numero di righe

del programma in modo tale da far effettivamente "girare" il programma con maggior velocità.

Ad esempio, una istruzione molto comoda è la BTG, che consente di togliere un bit in un solo ciclo macchina senza necessariamente dover passare per l'operazione di XOR.

Ma sono state implementate altre istruzioni che facilitano la gestione di tabelle: sono la TABLRD

(TABLE ReaD), la TABLWT (TABLE WriTe), la TLRD (Table ReaD) e la TLWT (Table WriTe). Queste quattro istruzioni consentono di spostare dati dallo spazio di memoria dati nello spazio di memoria di programma e viceversa.

Poiché l'unità base per la memoria di programma è lunga 16 bit mentre l'unità base della memoria dei dati è lunga 8 bit, sono richieste

due operazioni per il passaggio dei dati da un tipo di memoria all'altra.

Le due istruzioni TLWT e TABLWT vengono impiegate per scrivere dati dalla memoria dei dati nella memoria di programma. Le altre due istruzioni TLRD e TABLRD invece sono usate per scrivere dati dalla memoria di programma alla memoria dei dati.

Si ricorda che la memoria di programma può essere interna e/o esterna. Se quindi si prevede l'accesso alla memoria di programma esterna, si deve necessariamente configurare il PIC in modalità microcontrollore esteso oppure microprocessore. In figura 51 vediamo come effettivamente lavorano queste istruzioni.

Abbiamo già detto che le unità di allocazione sono diverse in funzione del tipo di memoria su cui si sta lavorando.

Per questo motivo si è reso necessario introdurre quattro registri di appoggio ciascuno ad 8 bit: i primi due sono TABLATH (TABLE LATCH High) e

TABLATL (TABLE LATCH Low). Questi registri offrono un temporaneo appoggio per le singole unità: supponiamo infatti di voler passare dalla memoria dei dati alla memoria di programma. Allora si dovranno inviare due byte dalla memoria dei dati per occupare una unità della memoria di programma che è di 16 bit.

In questo caso si usa dapprima la TLWT che, come si vede in figura 51-a, consente di riempire i due registri di latch. Attenzione però che per riempirli entrambi sarà necessario inviare i due byte con due istruzioni TLWT consecutive (perché è possibile passare solo 8 bit alla volta).

A questo punto, sarà possibile copiare il contenuto dei due registri di latch in una sola locazione della memoria di programma con la istruzione TABLWT, come si vede in figura 51-b. Ma come viene indicato l'indirizzo di destinazione? Attraverso i due registri dedicati TBLPTRH (TaBLE PoiTeR High) e TBLPTRL (TaBLE PoiTeR Low).

Quindi, prima di far eseguire questa istruzione, è necessario ricordarsi di caricare i due suddetti registri con l'indirizzo di memoria di programma voluto.

In definitiva, per passare 16 bit dalla memoria dei dati a quella di programma, sono necessarie almeno 5 istruzioni.

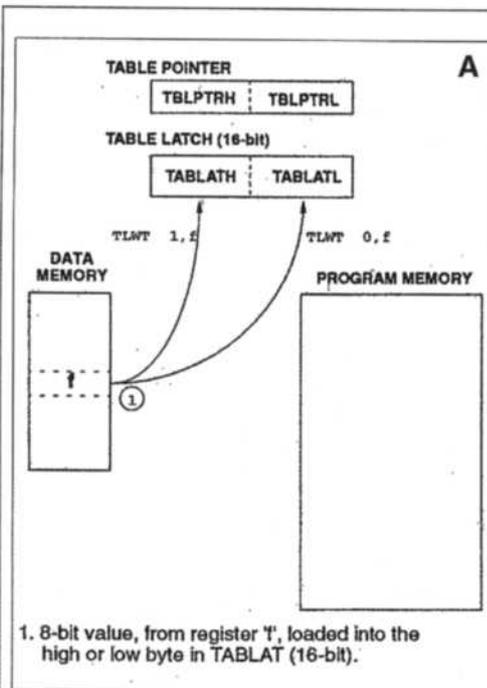


Figura 51-a. Istruzione TLWT

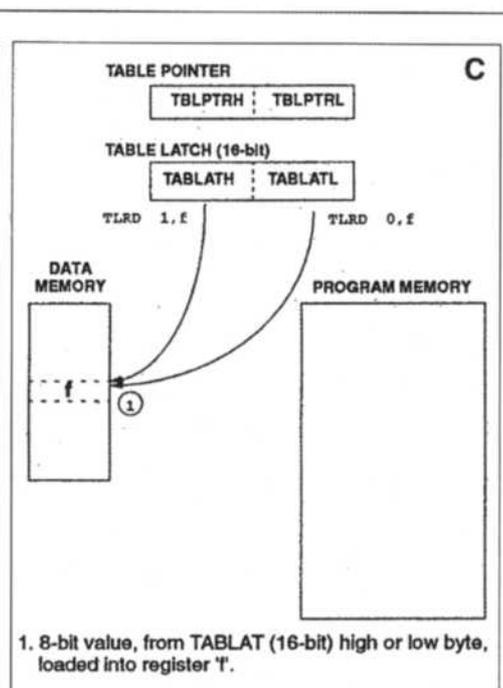


Figura 51-c. Istruzione TLRD

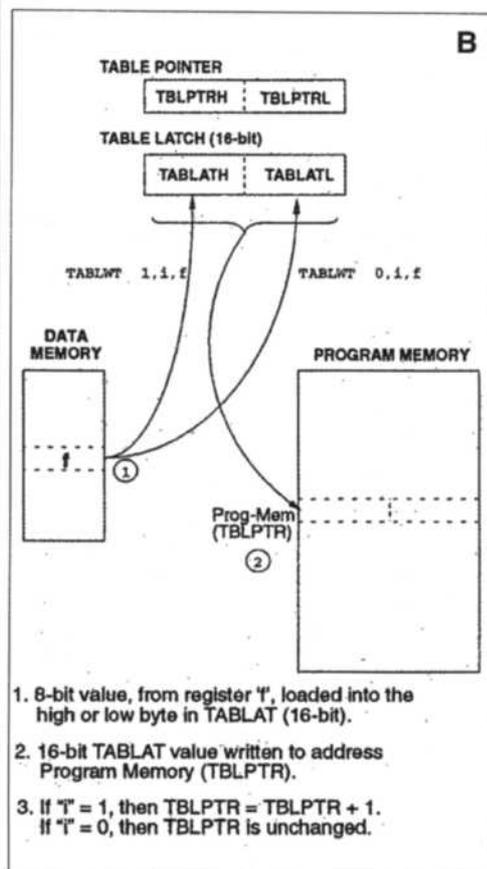


Figura 51-b. Istruzione TABLWT

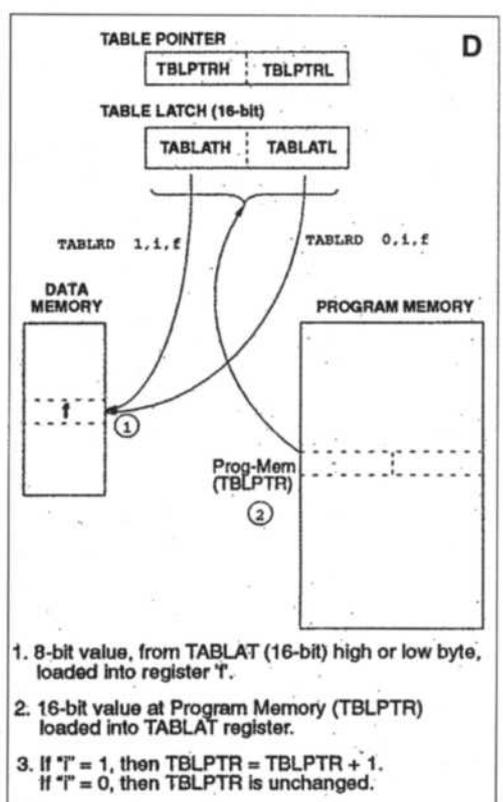


Figura 51-d. Istruzione TABLRD

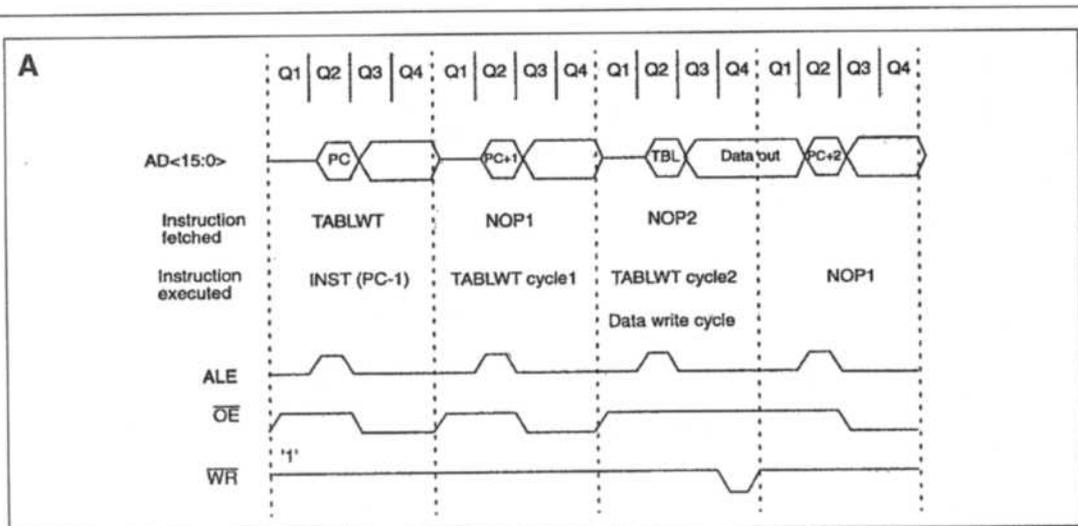


Figura 52-a. Diagramma dei tempi per l'istruzione TABLWT per 1 dato

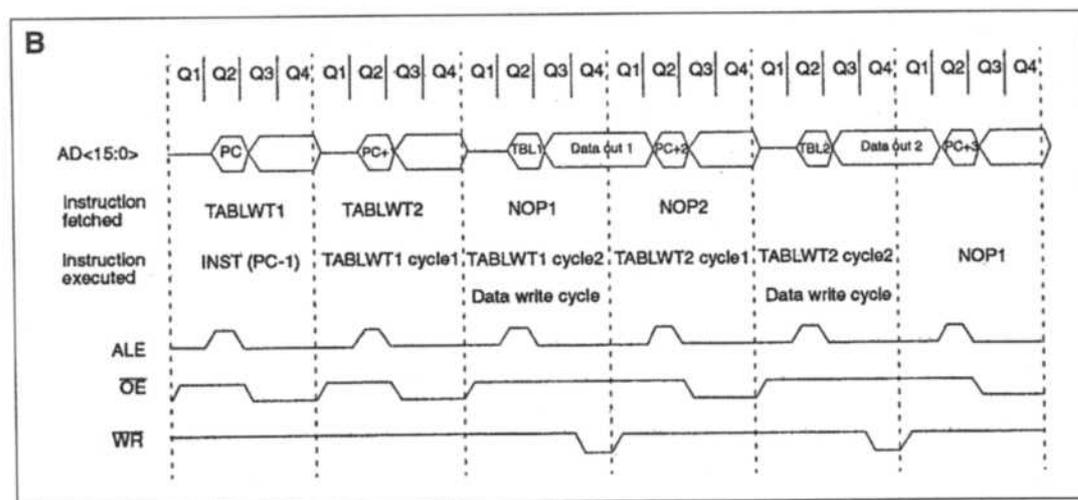


Figura 52-b. Diagramma dei tempi per l'istruzione TABLWT per più dati consecutivi

Vediamo adesso l'operazione inversa, ovvero come si fa per passare dei dati dalla memoria di programma alla memoria dei dati. Per prima cosa è necessario, come in precedenza, caricare nei registri puntatori TBLPTRH e TBLPTRL l'indirizzo della prima delle due locazioni di memoria dei dati su cui scrivere (ricordiamo che per la memoria di programma l'indirizzo è unico per i

16 bit, mentre per la memoria dei dati l'indirizzo è composto da due indirizzi consecutivi).

Per il passaggio inverso, sono disponibili due tipi di istruzione che agiscono in modo diverso: la prima è la TLRD illustrata in figura 51-c. Con questa istruzione, si ha il passaggio da un registro di latch alla cella specificata dal corrispondente indirizzo memorizzato nel registro puntatore.

Per passare tutti e due i byte allora sarà indispensabile inserire due istruzioni consecutive TLRD: la prima trasferirà il primo byte, la successiva il secondo.

La seconda istruzione disponibile è la TABLRD, illustrata in figura 51-d. Tale istruzione è molto utile, come vediamo adesso, per trasferire una serie di byte dalla memoria di programma alla memoria dei

dati senza ogni volta dover modificare i registri puntatori: l'unica condizione richiesta è che i dati siano memorizzati in celle consecutive.

Guardando infatti la figura 51-d ci accorgiamo che con la TABLRD vengono simultaneamente eseguiti due passaggi nell'ordine in cui li illustriamo: il primo prende i dati dai due registri di latch e li passa alla memoria dei dati, il secondo prende i dati della cella di memoria di programma successiva alla precedente e li pone nei registri di latch. In questo modo, non è necessario aggiornare i registri puntatori poiché viene fatto automaticamente dal controller via hardware e, così facendo, vengono risparmiate due istruzioni ad ogni passaggio di due byte.

Supponendo ad esempio di dover trasferire 20 celle (40 byte), il numero di istruzioni risparmiato sarà di $(20-1)*2 = 38$ istruzioni, che consentirà sicuramente una maggiore velocità di esecuzione del programma. Ripetiamo che però l'unica condizione da rispettare è la consecutività dei dati da trasferire e delle celle di destinazione.

Modalità di scrittura tabella

Un'operazione di scrittura tabella nella memoria interna causa, come già visto, un'operazione di scrittura detta "long", ovvero

lunga, per consentire la programmazione della memoria di tipo EPROM. Per essere sicuri che la locazione della cella EPROM è stata programmata correttamente, si deve attendere un certo tempo, che potrebbe anche essere gestito da interrupt.

La sequenza corretta per poter programmare una cella interna di memoria è la seguente:

- Disabilitare tutte le sorgenti di interrupt (perché qualsiasi interrupt giungesse durante il ciclo di scrittura lo terminerebbe in modo anomalo)
- Portare il pin MCLR/Vpp alla tensione di programmazione (circa 13,5 volt)
- Azzerare il contatore WATCHDOG
- Scrivere in tabella con le apposite istruzioni ed attendere l'intervallo di fine scrittura
- Verificare il contenuto della cella scritta eseguendo una riletta

Abbiamo detto che gli unici metodi per interrompere una scrittura su tabella sono un impulso di reset oppure l'arrivo di un qualsiasi interrupt. Se l'intervallo di una periferica viene sfruttato per terminare la scrittura di una tabella, i rispettivi bit di enable e di flag devono essere opportunamente settati.

Se il bit GLINTD viene azzerato prima dell'inizio della scrittura, allora al termine il programma salterà

all'indirizzo dell'intervallo vector. Viceversa, se il bit GLINTD viene settato prima dell'inizio della scrittura, allora al termine il programma non salterà all'indirizzo dell'intervallo vector.

Le operazioni che scrivono sulla memoria esterna sono sempre lunghe due cicli. Il secondo ciclo serve per scrivere il dato sulla cella di memoria esterna. La sequenza dei passi per la scrittura di una cella di

memoria esterna è identica a quella per la scrittura di una cella di memoria interna. Nella figura 52 abbiamo modo di vedere le tempistiche necessarie alla scrittura di un dato e di più dati consecutivi con le istruzioni TABLWT nel caso di memoria esterna. Si noti i cicli che non sono necessari per la memorizzazione di più dati consecutivi.

Le istruzioni di lettura tabellare, consentono anche

la lettura della memoria di programma. Ciò permette di memorizzare dati nella memoria di programma e di inviarli alla memoria dei dati quando occorre.

Sempre per questo motivo, si ricorda che quando il PIC viene configurato in modalità microprocessore oppure in modalità microcontrollore esteso, non sarà possibile proteggerlo dalla lettura del programma da parte di terzi.

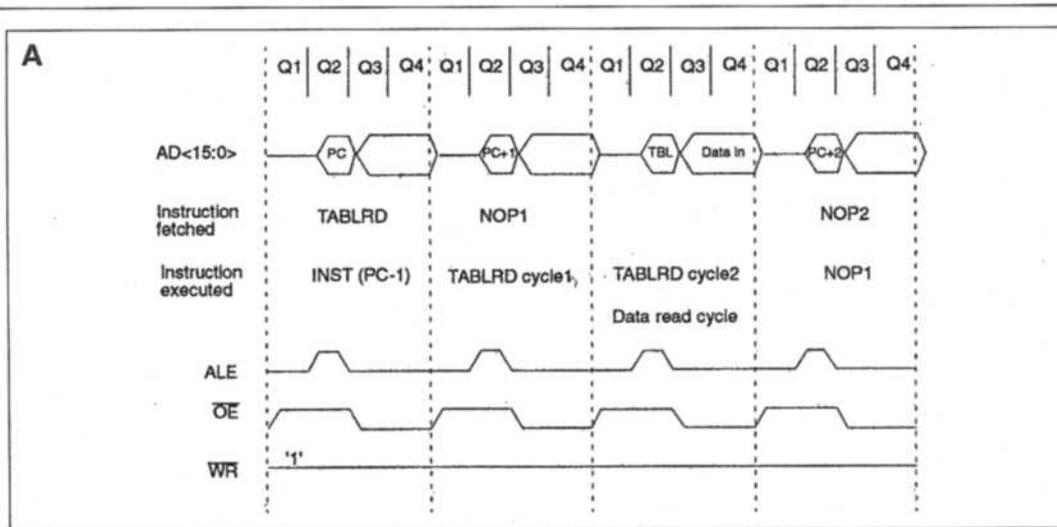


Figura 53-a. Diagramma dei tempi per l'istruzione TABLRD per 1 dato

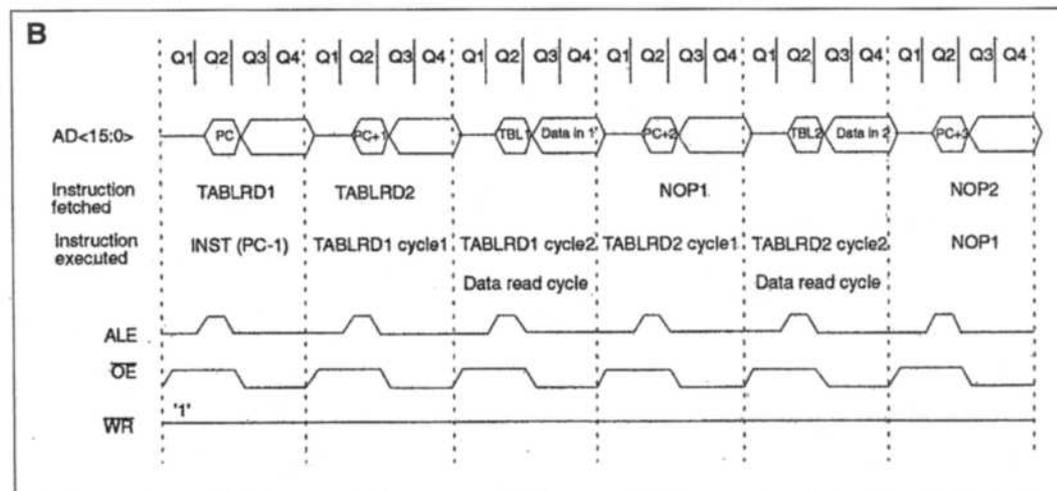
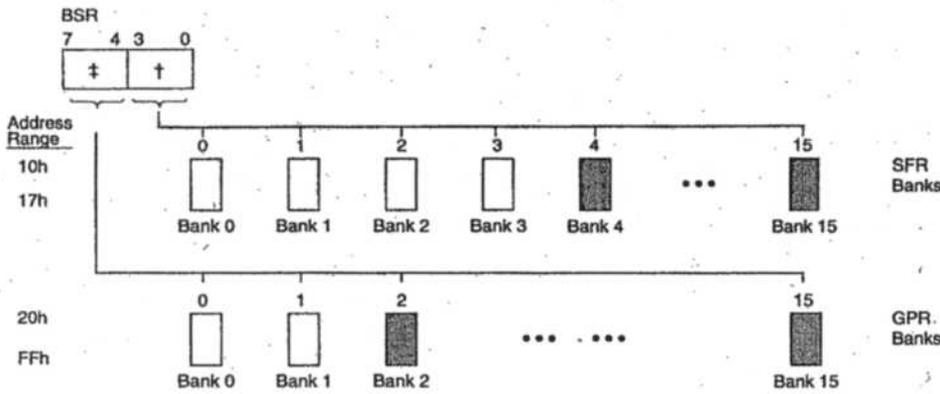


Figura 53-b. Diagramma dei tempi per l'istruzione TABLRD per più dati consecutivi



† Only Banks 0 thru Bank 3 are implemented. Selection of an unimplemented bank is not recommended. Bank 15 is reserved for Microchip use, reading of registers in this bank may cause random values to be read.
‡ Only Banks 0 and Bank 1 are implemented. Selection of an unimplemented bank is not recommended.

Figura 54. Operazioni sul registro BSR

di eseguire una moltiplicazione tra due dati ad 8 bit ciascuno con risultato di 16 bit. Tale modulo è incluso direttamente nella ALU (Arithmetic Logic Unit) ovvero unità logica aritmetica, che si occupa di gestire anche le operazioni di somma, sottrazione e tutte le altre operazioni di tipo algebrico come lo XOR, l'AND, ecc.

La moltiplicazione avviene tra un registro ed il working register W mentre il risultato viene collocato in due registri: PRODH e PRODL, uno per i bit più significativi, l'altro per quelli meno significativi. L'operazione

L'unica modalità che consente la crittografia della memoria di programma è quella detta "microcontrollore protetto". Da notare che, nel caso di scritture o letture multiple di dati, è possibile selezionare l'incremento o il decremento automatico dei registri puntatori TBLPTRH e TBLPTRL.

Nella figura 53 abbiamo modo di vedere le tempistiche necessarie alla lettura di un dato e di più dati consecutivi con le istruzioni TABLRD. Si noti anche in questo caso che leggendo più dati consecutivamente, si ha un risparmio sul tempo di lettura.

La moltiplicazione hardware

Nei PIC17C43 e 17C44 (escluso quindi il PIC-17C42) è presente un modulo hardware che consente

R/W - 1	R/W - 1	R/W - 1	R/W - 1	R/W - x	R/W - x	R/W - x	R/W - x
FS3	FS2	FS1	FS0	OV	Z	DC	C
bit7				bit0			

R = Readable bit
W = Writable bit
-n = Value at POR reset
(x = unknown)

- bit 7-6: **FS3:FS2: FSR1 Mode Select bits**
00 = Post auto-decrement FSR1 value
01 = Post auto-increment FSR1 value
1x = FSR1 value does not change
- bit 5-4: **FS1:FS0: FSR0 Mode Select bits**
00 = Post auto-decrement FSR0 value
01 = Post auto-increment FSR0 value
1x = FSR0 value does not change
- bit 3: **OV: Overflow bit**
This bit is used for signed arithmetic. It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit7) to change state.
1 = Overflow occurred for signed arithmetic, (in this arithmetic operation)
0 = No overflow occurred
- bit 2: **Z: Zero bit**
1 = The result of an arithmetic or logic operation is zero
0 = The results of an arithmetic or logic operation is not zero
- bit 1: **DC: Digit carry/borrow bit**
For ADDWF and ADDLW instructions.
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result
Note: For borrow the polarity is reversed.
- bit 0: **C: carry/borrow bit**
For ADDWF and ADDLW instructions.
1 = A carry-out from the most significant bit of the result occurred
Note that a subtraction is executed by adding the two's complement of the second operand. For rotate (RRCF, RLCF) instructions, this bit is loaded with either the high or low order bit of the source register.
0 = No carry-out from the most significant bit of the result
Note: For borrow the polarity is reversed.

Figura 55. Il registro ALUSTA

non muta i flag del registro di stato (ad esempio bit di zero o carry). Poter eseguire un'operazione di moltiplicazione ad 8x8 bit è molto importante per almeno due motivi: alta velocità di esecuzione del programma e riduzione del codice.

Poiché però questo modulo ha ancora un certo costo, la Microchip ha preferito non inserirlo, almeno per ora, nelle famiglie più piccole. Tanto per far capire come sia banale eseguire una moltiplicazione, vediamo quali sono le istruzioni necessarie per moltiplicare ARG1 x ARG2, entrambi ad 8 bit:

```
MOVFP ARG1,W ; Copia ARG1 in W
MULWF ARG2 ; Moltiplica W (che è = a ARG1) x ARG2
; Il risultato è in PRODH e PRODL
```

Se invece volessimo moltiplicare due numeri con segno, la routine diventerebbe:

```
MOVFP ARG1,W ; Copia ARG1 in W
MULWF ARG2 ; Moltiplica W (che è = a ARG1) x ARG2
BTFSC ARG2,SB ; Testa bit di segno
SUBWF PRODH,F ; PRODH = PRODH - ARG1
MOVFP ARG2,W ; Copia ARG2 in W
BTFSC ARG1,SB ; Testa bit di segno
SUBWF PRODH,F ; PRODH = PRODH - ARG2
```

Per avere un'idea della potenza di calcolo di questo modulo, in tabella 7 si può notare la comparazione tra i vari modelli di PIC e le quattro moltiplicazioni

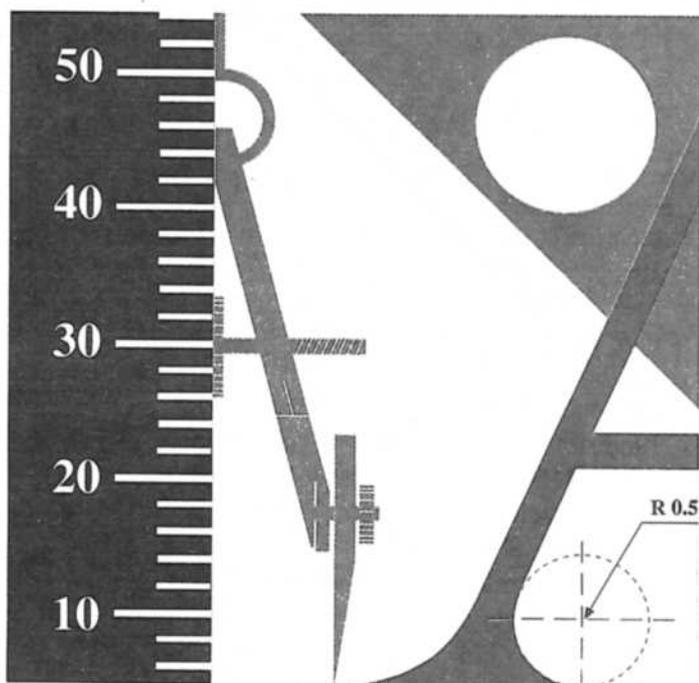
standard: senza e con segno ad 8 bit e senza segno o con segno a 16 bit. Si passa da un minimo di 160 nanosecondi a un massimo di 40,64 microsecondi.

Il Bank Select Register

Un altro registro "nuovo" per la famiglia 17C4x è il BSR ovvero il Bank Select Register. Tale registro viene impiegato per passare tra i vari banchi della memoria dei dati, come si vede in figura 54. Nel PIC17C42, è stato implementato solo il nibble più basso, cioè i primi 4 bit di peso inferiore, mentre nei PIC17C43 e 17C44 è stato utilizzato completamente.

Il nibble inferiore (si ricorda che nibble sta ad indicare un raggruppamento di 4 bit di un registro) viene

CAD ARCHITETTONICO CAD MECCANICO CAD STRUTTURALE MODELLAZIONE SOLIDA RENDERING FOTOGRAFICO AMBIENTAZIONI GRAFICA PROFESSIONALE PRE-PRESS CORSI DI FORMAZIONE RETI NETWORKING SOFTWARE HARDWARE ASSISTENZA APPLE SUN HP COMPAQ
 CAD ARCHITETTONICO CAD MECCANICO CAD STRUTTURALE MODELLAZIONE SOLIDA RENDERING FOTOGRAFICO AMBIENTAZIONI GRAFICA PROFESSIONALE PRE-PRESS CORSI DI FORMAZIONE RETI NETWORKING SOFTWARE HARDWARE ASSISTENZA APPLE SUN HP COMPAQ
 CAD ARCHITETTONICO CAD MECCANICO CAD STRUTTURALE MODELLAZIONE SOLIDA RENDERING FOTOGRAFICO AMBIENTAZIONI GRAFICA PROFESSIONALE PRE-PRESS CORSI DI FORMAZIONE RETI NETWORKING SOFTWARE HARDWARE ASSISTENZA APPLE SUN HP COMPAQ CA



AZIMUTH srl

VIA NAZARIO SAURO 48

20025 LEGNANO MI

T. 0331 540.178

FAX 0331 540.593

EMAIL: azimuth@betanet.it

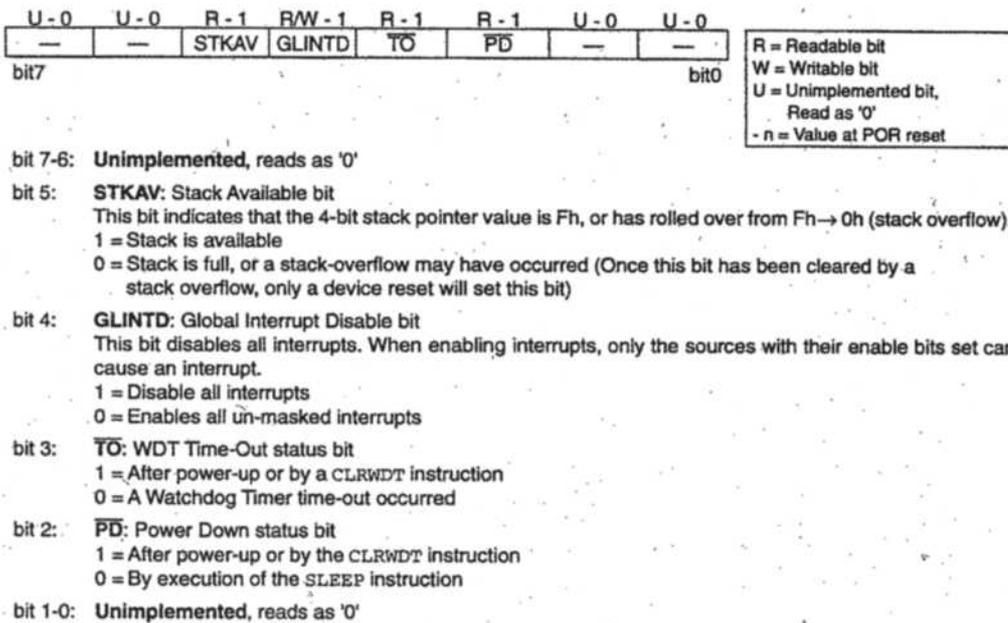


Figura 56. Il registro CPUSTA

usato per selezionare il banco dei registri delle periferiche, mentre il nibble superiore viene utilizzato per selezionare i banchi della memoria ad uso generale.

Tutti i registri ad uso speciale SFR (Special Function Register) sono mappati nello spazio della memoria dei dati. Per poterli sfruttare al meglio, è stato creato un sistema a banchi che va dall'indirizzo 0x10 all'indirizzo 0x17. Il nibble inferiore del BSR seleziona il banco corrente del "peripheral bank". Per muoversi tra i banchi, è stata inserita l'istruzione MOVLB (MOVE Literal to low nibble in Bsr) nel set di istruzioni.

Allo stesso modo, per muoversi nei banchi della memoria RAM, che vanno dall'indirizzo 0x20 all'indirizzo 0xFF, è stata inserita la nuova istruzione MOVLR

(MOVE Literal to high nibble in Bsr). I banchi della RAM vengono gestiti dal nibble superiore del BSR.

Se per caso vengono selezionati dei banchi che non sono stati implementati, la lettura di qualsiasi cella darà 0x00 come risultato. Nel banco 15 dell'area dei registri delle funzioni speciali inoltre, ci sono registri riservati alla Microchip. La lettura di questi registri potrà solo restituire valori casuali.

I registri di stato nei PIC17C4x

Imicrocontrollori della famiglia 17C4x non hanno un solo registro di stato, ma ne hanno due: il registro ALUSTA di indirizzo 0x04 ed il registro CPUSTA di indirizzo 0x06, entrambi nella porzione di memoria non bancata. Nelle figure 55 e 56 ne possiamo vedere i rispettivi contenuti. I primi tre bit del registro ALUSTA

corrispondono ai primi tre del vecchio registro di stato.

Il bit 3 ci indica se è avvenuto un overflow dopo un'operazione aritmetica.

I bit 4 e 5 selezionano l'incremento o il decremento automatico del registro FSR0 per l'indirizzamento indiretto con FSR0 mentre i bit 6 e 7 selezionano l'incremento o il decremento automatico del registro FSR1 per l'indirizzamento indiretto con FSR1. I primi due bit del registro CPUSTA non sono stati implementati.

Il bit 2 invece è il flag PD (Power Down) che ci dà informazioni sull'avvenuto reset del PIC mentre il bit 3 è il flag TO (watch dog Time Out) che indica quale sia stata la causa dell'entrata in funzione del timer dell'watch dog. Il bit 4 si chiama GLINTD (Global Interrupt Disable) e permette di disabilitare tutti gli interrupt.

Il bit 5 si chiama STKAV (Stack Available) ci indica se lo stack è disponibile oppure è pieno.

Infine i due bit 6 e 7 non sono stati implementati.

continua

Paolo Pescioni

Tabella 7. Comparazione tempi tra modelli di PIC ed operazioni

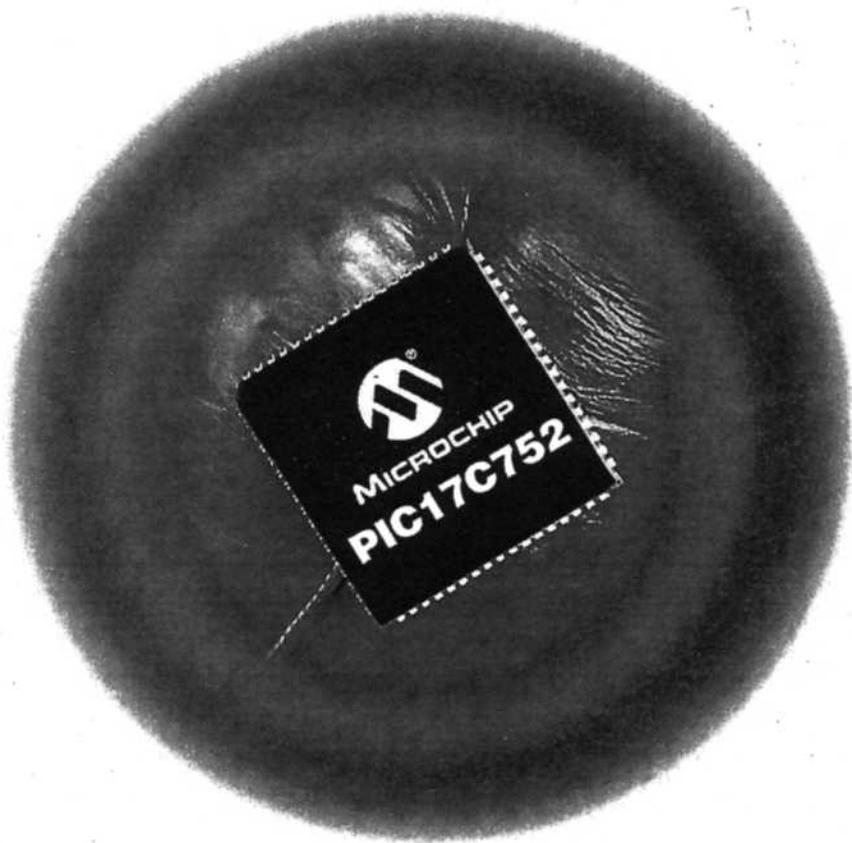
Routine	Device (Words)	Program Memory	Cycles (Max) (@ 25 MHz)	Time
8 x 8 unsigned	17C42	13	69	11.04 µs
	17C43 & 17C44	1	1	160 ns
8 x 8 signed	17C42	-	-	-
	17C43 & 17C44	6	6	960 ns
16 x 16 unsigned	17C42	21	242	38.72 µs
	17C43 & 17C44	24	24	3.84 µs
16 x 16 signed	17C42	52	254	40.64 µs
	17C43 & 17C44	36	36	5.76 µs

L'ultima generazione

Concludiamo questa rassegna dedicata ai nuovi PIC con la famiglia più completa: A/D converter a 10 bit, 3 PWM, 4 moduli Capture/Compare e clock di oltre 30 MHz

Ottava e ultima parte

di Paolo Pescioni



Fino ad ora, avevamo preso in esame microcontroller con al massimo 40 pin, partendo dagli ultimi nati ad 8 pin. Desideriamo ora portare a conoscenza anche la famiglia top della produzione Microchip, ovvero la 17C75x, caratterizzata da tutta una serie di migliorie e di periferiche speciali che tra poco andremo a valutare.

Innanzitutto, vediamo dalla figura 57 i package disponibili: si

passa dai 64 pin in versione Y-SHRINK DIP ai 68 pin in versione LCC. Come in ogni altro caso, l'aumento dei pin corrisponde all'aumento delle funzioni e delle opzioni offerte. Andiamo allora alla figura 58 per valutare quali siano queste maggiori funzioni rispetto ad un qualsiasi altro PIC delle altre famiglie.

Come si nota subito, l'architettura è assai più complessa delle altre e

si avvicina, a grandi linee, a quella della famiglia 17C4x vista in precedenza.

Le porte in questo caso sono 7, quasi tutte ad 8 pin. Ad ogni porta viene poi collegata una particolare periferica.

Le periferiche principali sono raggruppate in basso.

Si distinguono la bellezza di quattro timer, due porte seriali sincrone/asincrone, tre moduli PWM, quattro moduli Capture, una porta seriale sincrona (SPI, IIC-slave e IIC-master, Microwire), un gestore degli interrupt (che in questo micro sono ben 18!) ed infine un nuovissimo convertitore analogico/digitale a 12 canali e 10 bit di risoluzione.

La capacità dello stack, ovvero la possibilità di chiamare subroutine annidate, è di 16, mentre sia la RAM che la EPROM sono paginate come abbiamo visto per i 17C4x.

Anche per questi micro la ALU (Unità Logica Aritmetica) contiene un modulo per la moltiplicazione hardware 8x8 bit.

Le modalità di impiego sono 4: microprocessore, microcontroller, microcontroller esteso e microcontroller protetto.

È previsto quindi un bus su cui connettere sia RAM che EPROM esterne la cui capacità di indirizzamento arriva a ben 64K x 16.

La versione 17C752 ha 8K di EPROM e 454 byte di RAM, mentre la 17C756 ha 16 K di EPROM e 902 byte

di memoria RAM, con un ciclo di clock di 121 ns. Ovviamente sono presenti anche i moduli comuni a tutti gli altri chip come per esempio il POR (Power On Reset), il PWRT (PoWeRup Timer), l'OST (Oscillator Startup Timer) ed il WDT (WatchDog Timer).

L'organizzazione della memoria

Come nelle altre famiglie, anche nella 17C75x ci sono due sezioni separate di memoria: la memoria RAM e la memoria EPROM. In figura 59 troviamo la mappa della memoria di programma.

Si notano, in una zona esclusa dall'indirizzamento, il program counter a 16 bit ed i 16 registri che formano lo stack, sempre a 16 bit.

All'indirizzo 0000h è presente il vettore di reset, ovvero in quella locazione ci sarà la prima istruzione che il micro esegue dopo un qualsiasi evento di reset.

Per quanto riguarda invece il vettore degli interrupt, questo micro non ne ha uno solo come gli altri, ma ne possiede 4.

Il primo, all'indirizzo 000-8h è quello relativo ai pin degli interrupt esterni.

All'indirizzo 0010h invece abbiamo il vettore dell'interrupt del Timer0 mentre all'indirizzo 00-18h troviamo il vettore per interrupt generato dal pin TOCKI.

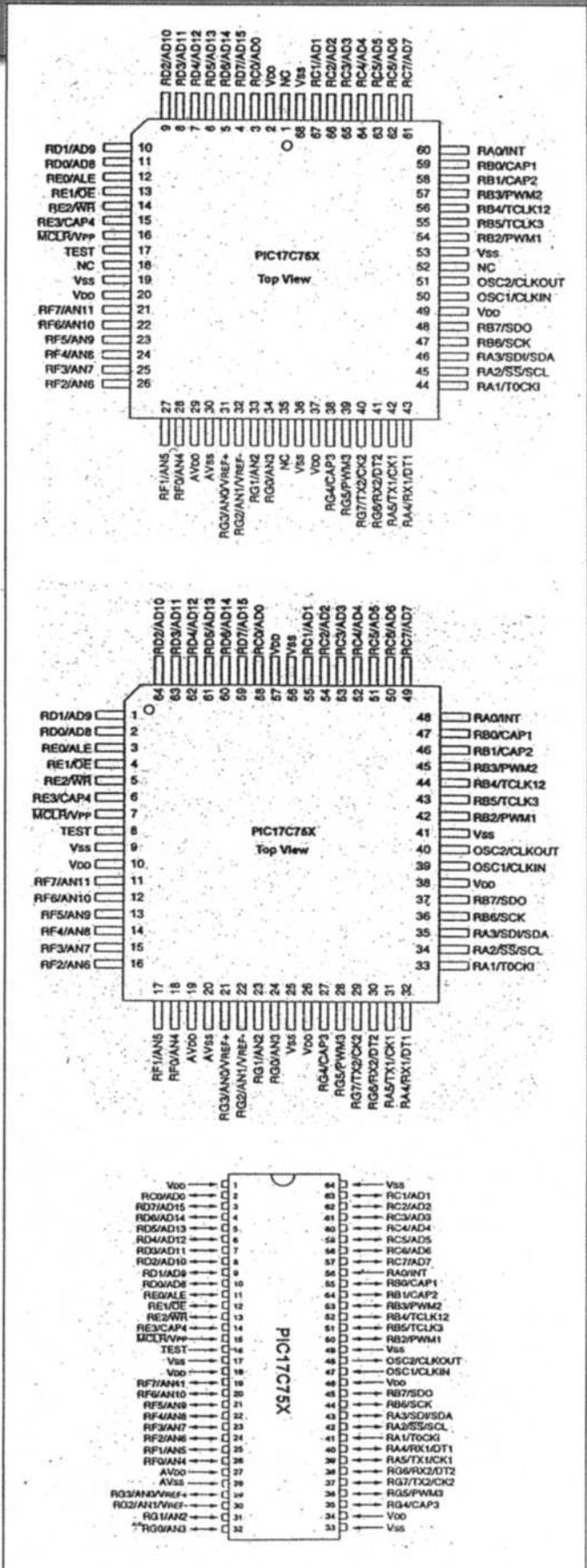


Figura 57. Package per la famiglia PIC 17C75x

Infine, all'indirizzo 0020h è collocato il vettore per gli interrupt provenienti da tutte le periferiche. A partire dall'indirizzo 0021h sono disponibili registri ad

uso generale fino all'indirizzo 1FFFh per il 17C752 e fino all'indirizzo 3FFFh per il 17C756. Dall'indirizzo FE00h invece si trovano alcuni registri per la

configurazione del micro ed altri per test Microchip.

I registri speciali sono invece paginati in "banchi" come avevamo già visto per i 17C4x.

Abbiamo già accennato al fatto che il micro può lavorare in 4 modalità differenti.

In figura 60 vediamo la mappa della memoria nei tre casi principali (ovvero escludendo quello protetto che è identico a quello microcontrollore).

Nel modo microcontroller, sono permesse soltanto le esecuzioni di istruzioni interne. Nella modalità estesa invece, sono consentiti accessi sia alla memoria interna che alla memoria esterna con un indirizzamento possibile di 64 K celle.

Nella modalità micro-processore infine, l'accesso è consentito alla sola memoria esterna (in pratica quella interna non viene utilizzata). Questa modalità è normalmente quella di default del micro.

Il convertitore A/D

Come abbiamo detto all'inizio, non tutti i moduli sono nuovi in questa famiglia e quindi ci preoccupiamo di analizzare soltanto le periferiche che possiedono caratteristiche diverse dalle precedenti.

Una di queste è senza ombra di dubbio il modulo del convertitore analogico/digitale.

Il numero dei canali a disposizione è aumentato a 12 e la definizione è migliorata di due bit, passando dagli 8 ai 10.

Tale salto potrebbe apparire insignificante, ma vediamo perché non lo è:

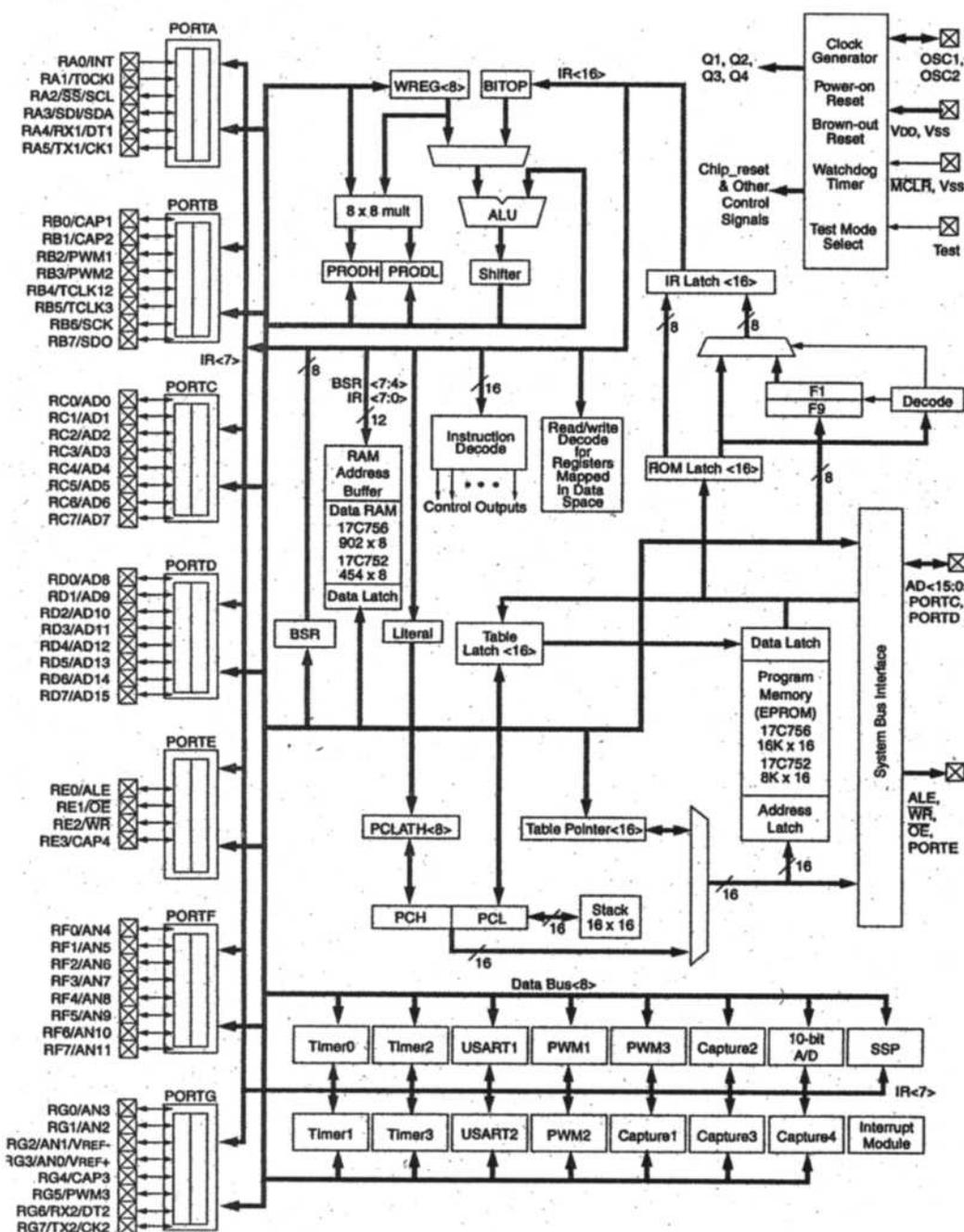


Figura 58. Diagramma a blocchi di un PIC17C75x

con una definizione di 8 bit sono possibili 256 livelli tra un minimo ed un massimo prestabiliti.

Supponiamo quindi di lavorare tra 0 e 5 volt: allora con 256 livelli possiamo ottenere una definizione di 5/256 che fa circa 20 mV.

In definitiva, se volessimo misurare una tensione con definizione di 10 mV non lo potremmo fare. Se invece i bit sono 10, allora i livelli passano a 2 elevato alla 10 che dà 1.024.

Considerando l'esempio precedente, 5/1.024 fa circa 5 mV. Allora otterremo una definizione di 5 mV su 5 volt, ovvero circa un millesimo.

Aumentando di soli due bit la conversione, si è ottenuto quindi una definizione migliorata di 4 volte rispetto a quella ottenuta con soli 2 bit.

Il diagramma a blocchi del convertitore è illustrato in figura 61. Come per i convertitori precedenti, il modulo è unico e viene eseguito un multiplexing tra i 12 canali.

I bit per la selezione del canale sono necessariamente passati da 3 a 4. In più rispetto alla famiglia 16C7x, questo modulo ha anche due riferimenti di tensione, uno positivo ed uno, ovviamente negativo.

Le procedure per l'inizio e la fine di una conversione sono identiche alle altre, ma questa volta cambia la destinazione del risultato.

Ricordiamo infatti che nella famiglia 16C7x è presente un registro ad 8 bit

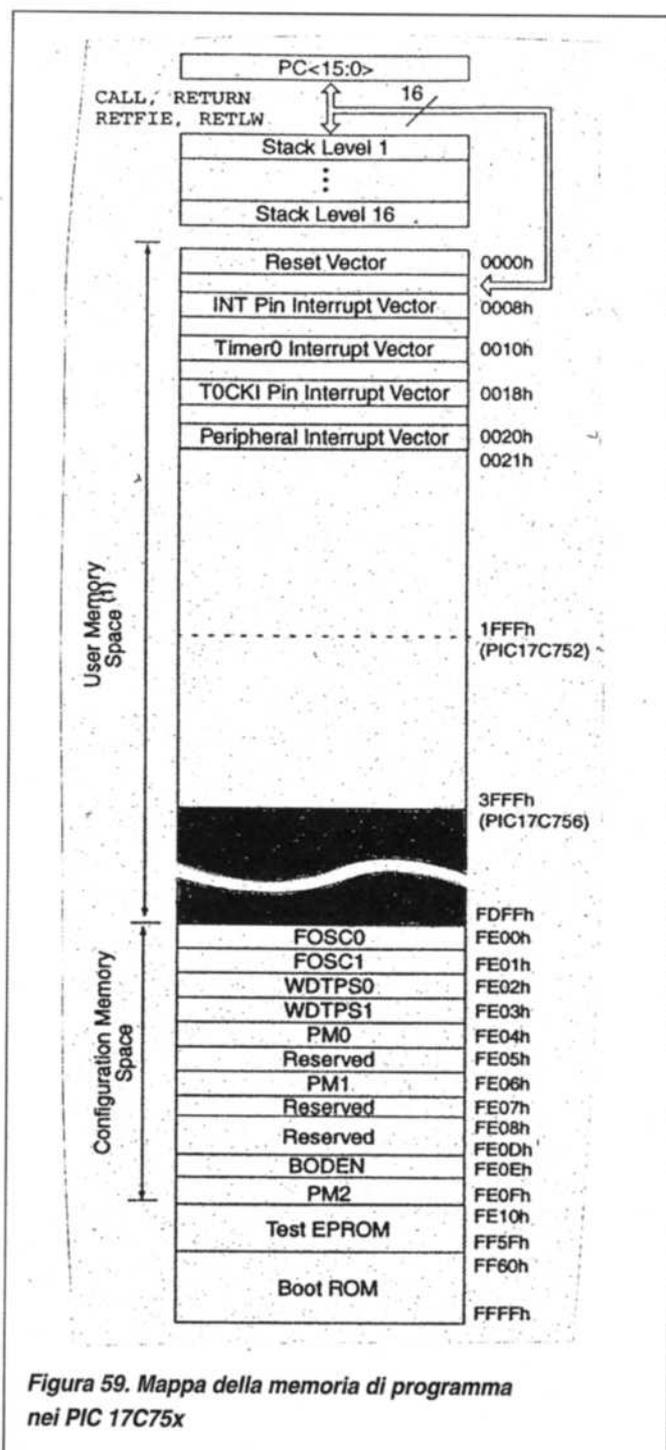


Figura 59. Mappa della memoria di programma nei PIC 17C75x

che si chiama ADRES e che il micro impiega per memorizzarci il risultato di una conversione.

Poiché i bit questa volta sono 10, un registro solo

non basta e quindi ne sono stati abbinati due, ADRESH e ADRESL.

La particolarità di questi due registri è che, modificando il valore del bit

ADFM, è possibile allineare a destra oppure a sinistra il risultato. In figura 62 possiamo vedere come ciò avviene.

Il metodo impiegato per eseguire la conversione è sempre lo stesso, ovvero quello delle approssimazioni successive, come può l'algoritmo per l'esecuzione delle istruzioni, ma ovviamente il tempo richiesto è inferiore alla famiglia 16C7x, dal momento che questi micro lavorano a 3 MHz contro i 25 MHz degli altri.

Il diagramma di flusso

In figura 63 troviamo il flow-chart delle operazioni necessarie al completamento della conversione: inizialmente si attende che il bit ADON sia settato a 1.

Poi si seleziona il canale da attivare tramite il multiplexer e si attende che il bit GO sia uguale a 1.

A questo punto il micro esegue un test per verificare se il clock per la conversione è prelevato dall'oscillatore RC interno ed in caso affermativo attende un ciclo di clock e fa partire la conversione.

Successivamente si testa se il micro era in SLEEP e, terminata la conversione, in caso affermativo decide se risvegliarsi dallo stato di SLEEP oppure tornare in quello stato.

Se inizialmente il micro non era in stato di SLEEP, si attende un certo

C17C752

C17C756

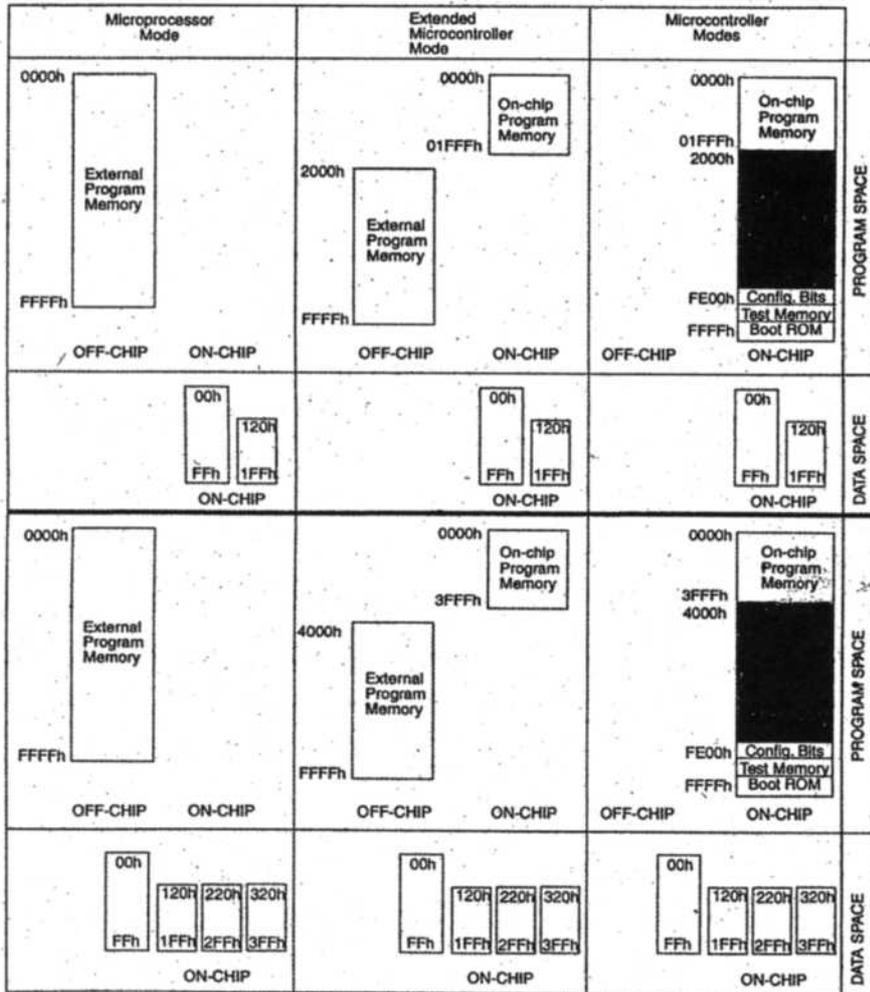


Figura 60. Mappa della memoria nei diversi modi operativi

tempo e poi la periferica "convertitore" è pronta a soddisfare un'altra richiesta. Se invece il tipo di oscillatore non era stato impostato come RC, si esegue un test sullo stato del micro: se questo era in SLEEP la conversione viene abortita e si torna in stato di SLEEP. Ciò perché la periferica del convertitore lavora con un clock e nel caso di un clock esterno il funzionamento si può

ottenere solamente quando il micro non è in stato di SLEEP, ovvero quando il suo oscillatore è in funzione.

La seriale sincrona

Un'altra periferica rivisitata è quella della seriale sincrona.

Con i modelli più anziani, non è infatti possibile avere un funzionamento in modalità IIC-bus

MASTER, ma soltanto IIC-bus SLAVE.

Ovviamente è sempre percorribile la strada dell'implementazione software, ma il dispendio di tempo e l'occupazione di memoria di programma ne sconsigliano l'uso.

Con queste nuove periferiche invece, il modulo IIC-bus MASTER è cablato direttamente in hardware e lavora sotto interrupt, lasciando così il micro libero

di eseguire altri lavori. Inoltre è consentita anche la modalità MULTIMASTER, ovvero la possibilità di collegare sullo stesso bus più di un componente MASTER senza preoccuparsi di dover gestire il protocollo di comunicazione e le eventuali collisioni.

In figura 64 troviamo il diagramma a blocchi della periferica SSP in IIC-MASTER mode.

Tutte le operazioni in modalità MASTER hanno inizio con l'invio di una cosiddetta condizione di START, di cui dopo vedremo un flow-chart chiarificatore. Quando il modulo riceve una condizione di START o di STOP, viene generato un apposito interrupt che interessa rispettivamente i bit S e P. Il MASTER può prendere il controllo del bus solo quando il bit P è settato a 1, oppure quando sia il bit P che il bit S sono azzerati.

Ovviamente le linee SDA (Serial Data) e SCL (Serial CLock) vengono gestite automaticamente dall'hardware visibile in figura 64. Si possono notare il registro buffer SSPBUF ed il registro che contiene effettivamente il byte da inviare (SSPSR).

Dal diagramma si vede anche che è presente un generatore autonomo di baud-rate: ciò consente di poter attivare il dialogo a 100 kHz, ovvero in modalità IIC standard, oppure a 400 kHz, ovvero in modalità FAST.

Nella modalità multi-master, la generazione di

interrupt sulle condizioni di START e di STOP permette di verificare se il bus è libero oppure no. Può però accadere che due master trovino contemporaneamente il bus libero e, sempre contemporaneamente, tentino di impossessarsene.

Questa situazione non è ottimale, ma è prevista dal protocollo del bus: esiste una procedura che si dice "arbitrazione" e che consente di stabilire quale tra tutti i master hanno iniziato la comunicazione, ha diritto di dialogo.

Condizioni tipiche in cui è possibile perdere il diritto al dialogo sono il trasferimento di indirizzo o dati, una condizione di start oppure di stop, una condizione di acknowledge.

Quando viene attivata la modalità master per la periferica settando e azzerando i bit appropriati nel registro SSPCON1, il programmatore può scegliere tra sei opzioni: avviare una condizione di START sul bus, avviare una condizione di RESTART, scrivere nel registro SSPBUF per iniziare a trasferire un dato oppure un indirizzo, generare una condizione di STOP sul bus, configurare la porta IIC per la ricezione di dati ed infine generare una condizione di acknowledge alla fine di un byte di dato ricevuto.

Ma la situazione più critica rimane la presa di possesso del bus generando una condizione di START.

In figura 65 troviamo ben dettagliato il diagramma a blocchi relativo all'esecuzione nel micro di una condizione di START.

Per prima cosa si setta a 1 il bit SSPEN e si passa in modalità IDLE.

Poi si testano le due linee SDA e SCL: se almeno una di queste è uguale a 0,

allora significa che si è verificata una collisione e quindi viene rilasciato il bus. Se viceversa entrambe queste linee sono a 1, allora il contatore del baud-rate viene

Figura 61. Diagramma a blocchi del convertitore A/D

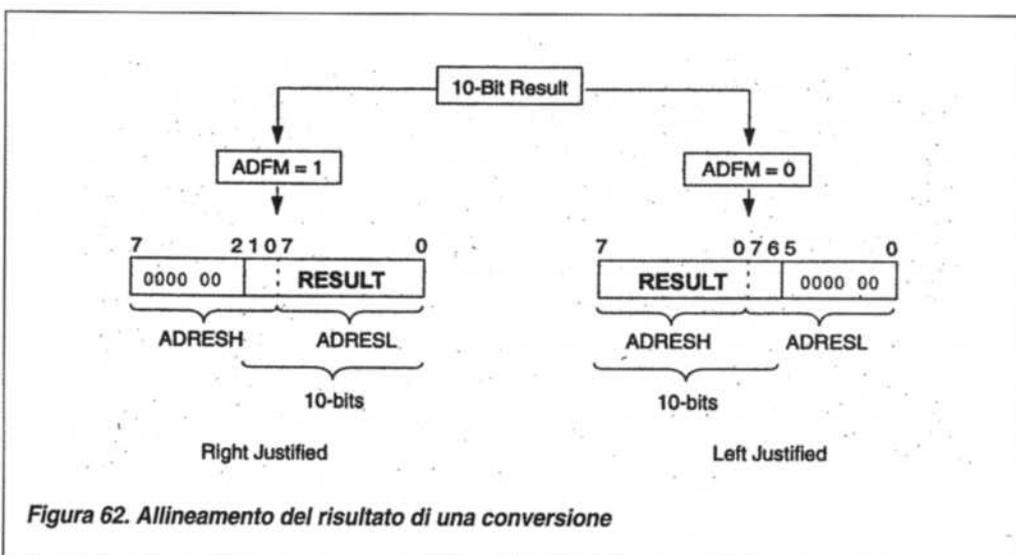
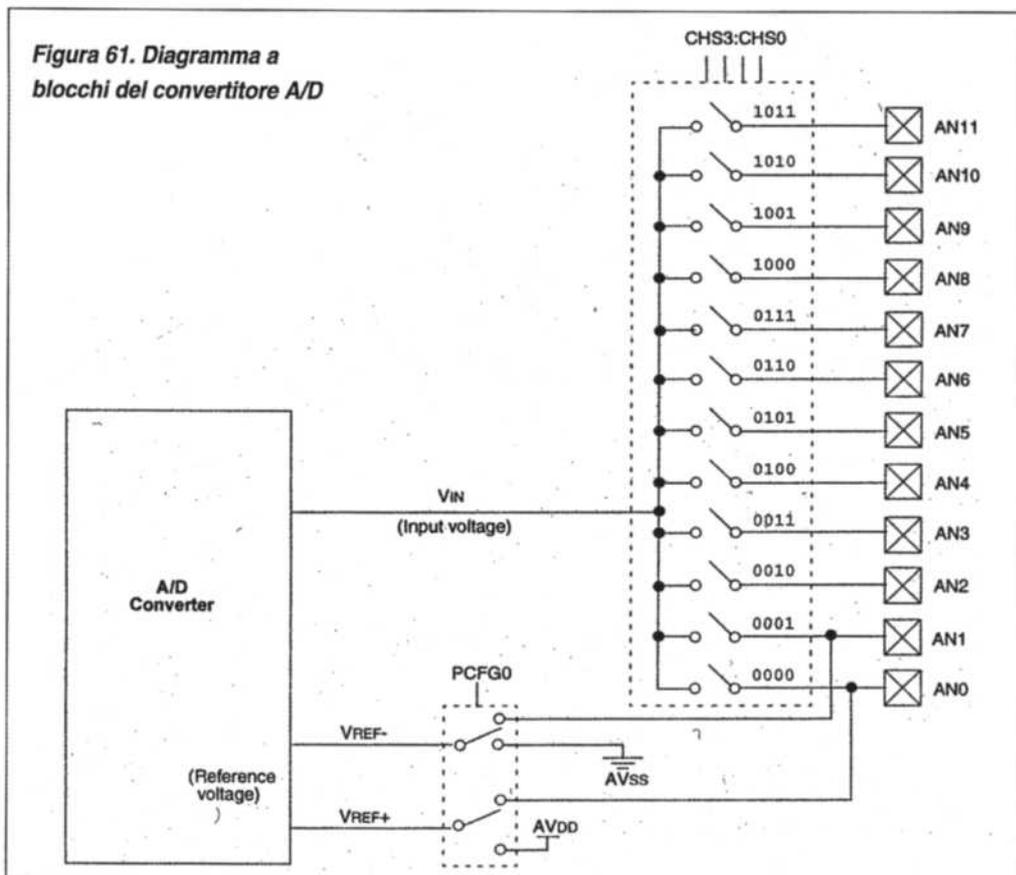


Figura 62. Allineamento del risultato di una conversione

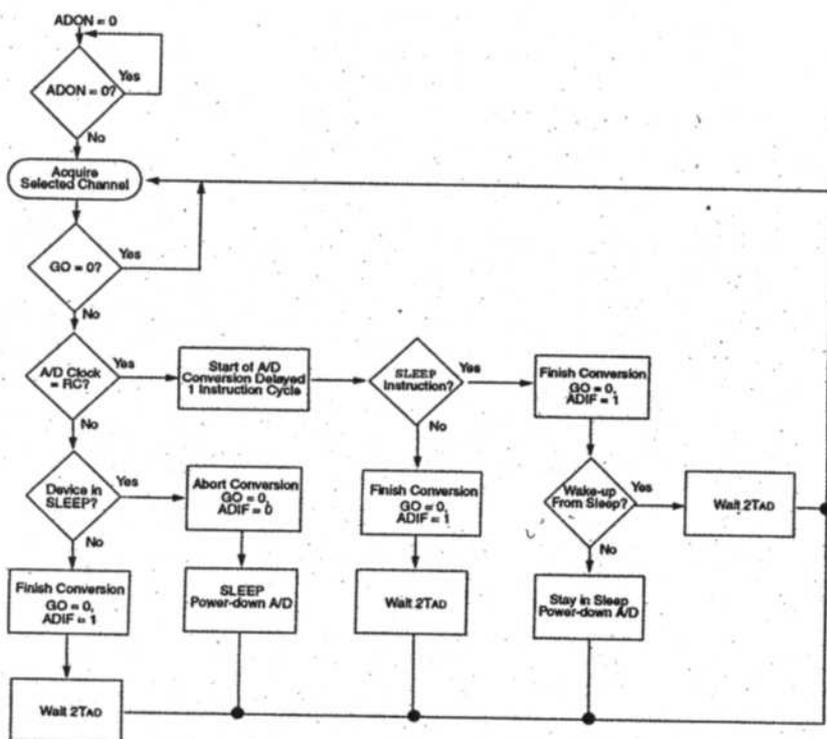


Figura 63. Flow-chart per una corretta conversione

caricato con il valore di SSPADDE e si attende l'overflow del BRG (Baud Rate Generator). Se ciò non accade e la linea SDA va a 1 e la linea SCL passa a 0, allora significa che siamo sempre in presenza di una collisione e quindi si torna all'inizio liberando il bus. Quando invece il BRG va in overflow, la linea SDA viene forzata a 0 ed il BRG stesso viene ricaricato con il valore precedente.

Si attende nuovamente l'overflow del BRG e, quando ciò accade, la linea SCL viene forzata a 0, segno che la condizione di START è stata effettuata correttamente.

Conclusioni

Siamo quindi giunti al termine della panoramica iniziata ben otto mesi fa sui nuovi microcontrollori Microchip e, nel momento in cui stiamo scrivendo, sul mercato stanno uscendo ancora nuovi prodotti.

La nostra intenzione è stata di mostrare tutte le famiglie di microcontroller disponibili e le loro caratteristiche principali, quelle cioè che all'inizio di una progettazione fanno decidere se preferire un tipo di chip oppure un altro. Per la reperibilità di eventuali data book, oppure per ricevere un CD-ROM con tutto il software Microchip, potete liberamente contattare direttamente la Microchip Arizona visitando il sito Internet:

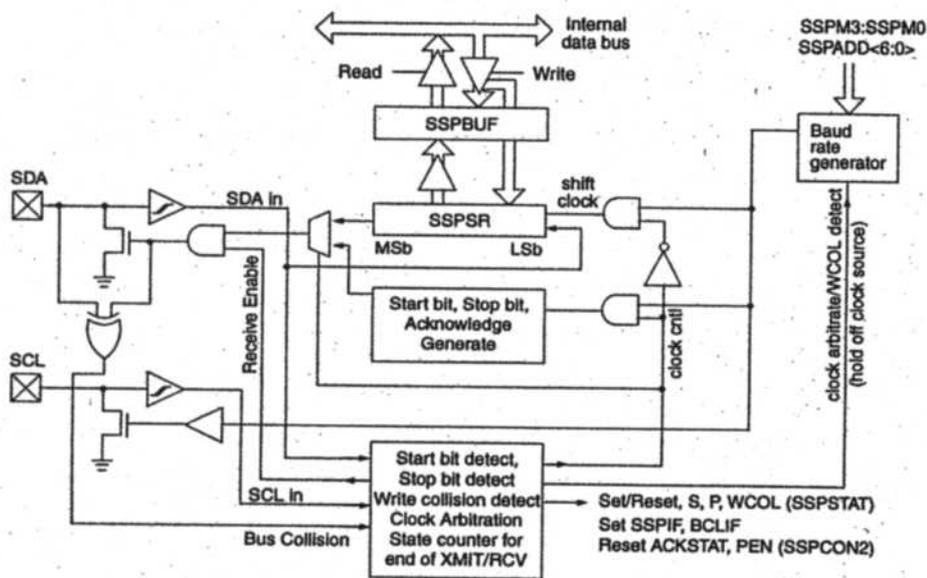


Figura 64. Schema a blocchi del modulo IIC in MASTER mode

<http://www.microchip.com> dove troverete anche gli indirizzi dei distributori locali e dei consulenti italiani.

Se volete reperire un sistema di sviluppo valido, nel sito suddetto troverete anche le terze parti, ovvero quei produttori di Hardware relativo ai PIC certificati e garantiti da Microchip.

Naturalmente tutto ciò che troverete sarà disponibile in lingua inglese. ■

Per chi vuole approfondire

Se desiderate imparare a programmare i microcontrollori della Microchip oppure progredire con le vostre conoscenze, sono disponibili due libri in lingua italiana con allegato un dischetto ciascuno che vi guideranno passo per passo alla programmazione avanzata di questi microcontrollori.

I libri sono reperibili al numero telefonico 0544/464059 presso la GPEKit di Ravenna.

Attualmente è in preparazione un altro libro su cui verranno pubblicati dieci progetti di grande interesse realizzati con i PIC, partendo dalla progettazione dell'hardware per giungere a quella del software. I programmi sorgente sono spiegati passo per passo e distribuiti su dischetto. Non è possibile stabilire l'esatta data di uscita in quanto ogni progetto deve prima essere sottoposto ad una lunga serie di test prima di essere approvato.

Sempre in questo periodo, sta per uscire presso la GPEKit di Ravenna un altro libro che spiega in modo chiaro il mondo dei microcontrollori in generale. Si parte dai concetti di base come i vari sistemi di numerazione, le conversioni tra tali sistemi, le operazioni booleane, l'architettura di un microcontrollore. Si prosegue poi alla simulazione su carta di un piccolo programma scritto per un PIC16C84, per far capire come convertire un problema reale in un circuito elettronico che lo risolve.

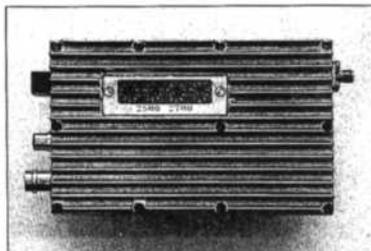
MODULI VIDEO-AUDIO 1000 - 2700 MHz

Modulatori, demodulatori e accessori per la costruzione di apparecchiature nei settori: sorveglianza, sicurezza, misura, CCTV, CATV, distribuzione segnali TV e radio, ecc.

MTV2500

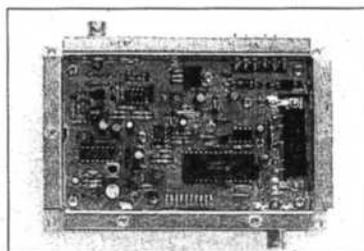
Modulatore FM Video+Audio sintetizzato

- 1000 - 2700 MHz in sottogamme da 100 - 200 MHz
- Impostazione frequenza esterna
- Potenza: 0,6 - 1,5 W in funzione della sottogamma scelta
- Modulazione portante video in frequenza ± 4 MHz
- Sottoportante audio: 5,5 - 7,5 MHz
- Alimentazione: 12 Vc.c.
- Dimensioni: 123 x 78 x 37 mm



TRV12 - TRV17 - TRV19

Modulatori FM Video+Audio sintetizzati

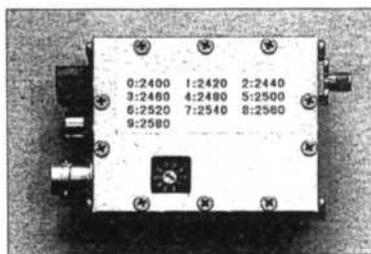


- TRV12: 1150 - 1350 MHz
- TRV17: 1650 - 1850 MHz
- TRV19: 1850 - 2000 MHz
- Potenza TRV12: 1,5 W
TRV17 e TRV19: 1 W
- Modulazione portante video in frequenza ± 4 MHz
- Sottoportante audio: 5,5 - 7,5 MHz
- Alimentazione: 12 Vc.c.
- Dimensioni: 147 x 99 x 37 mm

MTX2100

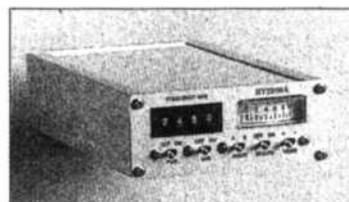
Modulatore FM Video+Audio sintetizzato

- 1000 - 2700 MHz in sottogamme da 100-200 MHz
- Impostazione frequenza esterna 10 canali
- Potenza: 100 mW
- Modulazione portante video in frequenza ± 4 MHz
- Sottoportante audio: 5,5 - 7,5 MHz
- Alimentazione: 12 Vc.c.
- Dimensioni: 52 x 64 x 23 mm



RV2000A

Demodulatore FM Video+Audio sintetizzato



- Ingresso: 950 - 2050 MHz (2050 - 2700 MHz con LNC)
- Uscita video: 1 Vpp - 75 Ω
- Uscita audio: 0 dBm
- Alimentazione: 12 Vc.c.
- Dimensioni: 116 x 55 x 150 mm
- Disponibile versione economica solo scheda mod. RV2000S

Produciamo inoltre: modulatori FM video+audio a 350 MHz, moduli a 1660 MHz per audio radiofonico, moduli bidirezionali audio, schede per telecomandi, sistemi audio per linee telefoniche dedicate, antenne, filtri e accessori.

BiTEL Microprogetti

Via Cadorna, 24/3 - 20059 Vimercate (MI) - Tel. 039/668849 - 668506
Fax 039/668849 - E-mail: micropro@pop.excalibur.it