

Registri LAT nei Picxxxxx

I pin di I/O sono considerati le più semplici delle periferiche.

Sui PIC16 della serie Midrange (sono quindi esclusi quelli appartenenti alle serie Enhanced) abbiamo essenzialmente due soli registri associati alle funzioni di I/O: TRISx e PORTx (dove la x va sostituita con la lettera che identifica il banco di porte).

Su molti PIC10/12, invece, essendoci un unico banco di porte, il registro PORT diventa GPIO (General Purpose I/O) e il registro TRIS diventa TRISIO: solo nomi leggermenti differenti.

Su tutti PIC18 (ma anche sugli ultimi nati delle famiglie PIC10/12/16 Enhanced) oltre a questi due registri è stato aggiunto un registro LATx che per molti aspetti sembra svolgere le stesse funzioni del registro PORTx ma eliminando alcuni inconvenienti che analizzeremo dopo in dettaglio (per cui questo articolo non è destinato solo agli utilizzatori dei pic a 16bit ma un po' a tutti).

Sui pic a 16 bit (PIC24 e dsPIC) oltre questi 3 registri ce ne sono altri che permettono di settare altre funzionalità specifiche dei "semplici" I/O digitali.

Problemi del registro PORT per capire del perchè del registro LAT e introdurre quindi i nuovi registri presenti sui pic a 16bit.

Registri TRIS

I registri TRISx (TRISA, TRISB, TRISC ecc), o il registro TRISIO su molti PIC10/12, determinano il funzionamento di un pin di I/O come ingresso o come uscita.

Se il bit di una porta nel registro TRISx viene posto ad 1, il pin viene impostato come Ingresso (ovvero: il pin va in alta impedenza), se invece il bit viene posto a 0, il pin relativo funziona come uscita.

Ricordo ancora una volta che il funzionamento di un I/O come digitale è sempre e comunque influenzato dalla eventuale presenza di altre periferiche multiplexate sul pin: il pin funziona come I/O digitale se le periferiche aggiuntive (come ad esempio un convertitore A/D, un comparatore ecc) sono disattivate.

Un sistema semplice per ricordarsi di questo comportamento è associare il numero 1 alla lettera I (Input) e il numero 0 alla lettera O (Output).

Registri PORT

I registri PORT servono ad accedere al dato presente su un pin di I/O.

Quando un pin è configurato come ingresso, una lettura del bit ci indica il livello presente sul pin: se viene restituito un 1 vuol dire che il pin sta leggendo un livello logico alto, uno zero indica un livello logico basso.

Se un pin è configurato come uscita e sul bit relativo nel registro PORT scriviamo un 1, portiamo il pin a livello logico alto, se scriviamo uno zero portiamo il pin a livello logico basso.

Quando un pin viene utilizzato come uscita e quindi intendiamo impostare il suo livello logico di uscita, il dato che noi andiamo a scrivere viene prima posizionato in un latch, ovvero una cella elementare di memoria e quindi successivamente trasferito al pin (alla "cella" del pin), mentre la lettura del registro PORT viene eseguita direttamente sulla cella di uscita del pin.

Il valore in tensione dei livelli logici è in funzione della tensione di alimentazione del dispositivo. Generalmente il livello logico basso è identificato dal riferimento di 0V indicato anche come GND o Vss, mentre il livello logico alto è identificato dalla tensione di alimentazione Vdd del picmicro (in genere 5V per i pic10/12/16/18/dsPIC30 e 3.3V per i PIC24/dsPIC33).

C'è ovviamente un valore di soglia per il quale anche un livello logico più basso di Vdd viene identificato come alto, un valore di soglia per il quale anche un livello logico più alto di Vss viene identificato come basso e quindi un range "indeterminato" nel quale un valore di tensione può essere identificato come alto o basso e quindi nel quale non bisogna mai lavorare.

Un pin può essere configurato dinamicamente (a run time) sia come ingresso che come uscita e può anche passare da uno stato all'altro di continuo.

Ovviamente quando si esegue un'operazione del genere (passaggio da uscita a ingresso e viceversa) bisogna prendere delle precauzioni sia lato hardware per evitare cortocircuiti sia per evitare comportamenti indesiderati lato software.

Un esempio: supponiamo che un pin configurato come ingresso viene impostato come uscita; ad un tratto potrebbe verificarsi la presenza di un valore logico non voluto sul pin di uscita per cui generalmente si scrive prima il valore a cui impostare il pin nel registro PORT e solo successivamente si cambia il funzionamento del pin nel registro TRIS.

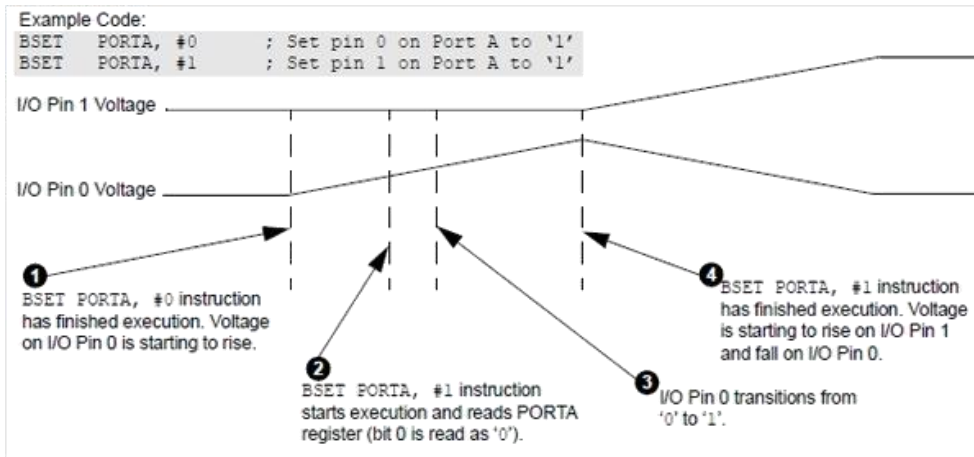
C'è da aggiungere una questione molto importante e spesso sottovalutata riguardante il funzionamento del registro PORT.

Quando un pin funziona come uscita, noi da software settiamo il livello di uscita (lo impostiamo a 1 o a zero).

Questa operazione viene eseguita in assembler mediante le istruzioni BSET (Bit Set => Setta un bit, ovvero lo imposta a 1) e BCLR (Bit Clear => Resetta un bit, ovvero lo imposta a zero).

Le istruzioni BSET e BCLR sono in realtà istruzioni di lettura-modifica-scrittura ovvero quando vengono eseguite viene dapprima letto lo stato di tutto il banco (viene letto lo stato di PORT e salvato in memoria ram), il valore del bit viene quindi modificato nella ram e solo alla fine il dato viene trasferito (scritto) nel latch di uscita. Impostare un bit ad un nuovo valore, quindi, comporta a livello hardware una prima operazione di lettura di tutto il banco. Una cosa che generalmente viene ignorata.

Quando eseguiamo operazioni di modifica del registro PORT mentre il pin è configurato come uscita, si potrebbero avere comportamenti indesiderati se c'è un elevato carico capacitivo sul pin o se il dispositivo funziona a frequenze di clock molto elevate e queste operazioni vengono eseguite di continuo, questo proprio a causa di questa operazione "nascosta" di lettura a cui non si pensa.



La prima istruzione eseguita è quella che prevede di portare a livello alto il pin 0 (la linea inferiore nello schema). Vediamo che dall'inizio fino al punto 1 l'istruzione viene caricata ed eseguita (la cpu legge lo stato di PORTA, ne carica il valore in memoria, lo modifica e quindi trasferisce il nuovo valore al latch di uscita del banco PORTA).

L'istruzione quindi termina in 1 e di conseguenza la tensione sul pin 0 comincia a salire per portarsi al livello che gli abbiamo specificato.

Questo accadrà in un certo tempo, molto molto breve.

Nel punto 2 comincia l'esecuzione dell'istruzione successiva. Abbiamo detto che prima di eseguire il Bit Set, il registro PORTA viene letto. Quindi nel punto 2 il processore legge tutto il banco A e leggerà il pin 0 come a livello logico zero in quanto la tensione non ha ancora raggiunto il valore di soglia. Ecco perchè specificavo che questo problema si ha soprattutto quando le frequenze di funzionamento sono molto elevate: ha fatto prima il processore a leggere lo stato della porta che non il pin a portarsi a livello logico alto! Questo può accadere anche nei casi in cui sui pin in uscita ci sono capacità molto elevate e che quindi rallentano l'operazione di passaggio di stato.

Nel punto 3 il pin 0 ha quindi raggiunto il valore di soglia e si trova pertanto a livello alto. Nel frattempo la seconda istruzione, però, non è ancora terminata!

Nel punto 4 finalmente anche la seconda istruzione termina il che vuol dire che il processore, dopo aver modificato il valore di PORTA che ha salvato in memoria, lo dovrà scrivere sul latch di uscita. Il valore di Pin 1 lo scrive correttamente perchè è l'ultimo che abbiamo impostato, il valore di pin 0 invece verrà scritto come zero in quanto è stato l'ultimo valore che ha letto prima della successiva modifica. Il risultato è che soltanto il pin 1 si troverà a livello alto mentre il pin 0 si troverà ancora a livello basso nonostante noi abbiamo eseguito l'operazione. Capite quindi che bel pasticcio quando andiamo a lavorare con dispositivi ad alta frequenza di clock. Microchip ha pensato a tutto e ha quindi introdotto il registro LAT.

Il registro LAT

Il registro LAT, sempre presente su tutti i pic dalla serie 18 in su e sui nuovi PIC10/12/16 serie enhanced, elimina i problemi relativi a queste operazioni di lettura-modifica-scrittura. Come?

L'operazione di lettura del registro LATx esegue la lettura del Latch di uscita anzichè del pin.

Il vantaggio è : se il pin non ha ancora cambiato lo stato logico (perchè il processore sta andando troppo veloce e ha letto prima che cambiasse o perchè c'è un elevato carico capacitivo che rallenta l'operazione "elettrica" di cambio stato) siamo sicuri che invece nel latch di uscita c'è sicuramente il valore che noi abbiamo impostato e che successivamente sarà trasferito al pin di uscita!

La scrittura in LATx, invece, è identica alla scrittura su PORTx, senza alcuna differenza.

Quindi, ricapitolando:

- Una scrittura in PORTx scrive nel latch di uscita
- Una scrittura in LATx pure scrive nel latch di uscita, quindi per le operazioni di scrittura, usare LAT o PORT è indifferente
- Una lettura di PORTx esegue la lettura dello stato in cui si trovano i pin
- Una lettura di LATx esegue la lettura del latch di uscita, il cui valore potrebbe essere differente da quello di PORTx o meglio: non subito uguale.

Per operazioni di scrittura potete usare indifferentemente uno dei due registri, per operazioni di lettura è più sicuro leggere il registro LAT (ma poi dipende dalla particolare applicazione).