Funzioni avanzate di IO digitale sui PIC32.
La gestione "atomica" dei bit.

Caratteristica presente unicamente sui pic32: l'**Atomic Bit Manipulation**.

Sui PIC32 abbiamo (relativamente alla gestione degli IO): **PORTx, LATx, TRISx, ODCx, CNCONx, ANSELx** e funzionalita' **PPS.**

In più abbiamo questa nuova caratteristica che permette di agire sui singoli bit di un registro (da qui la denominazione "atomic", si va ad agire direttamente sul singolo bit) consentendo di eseguire le operazioni di cambio stato dei singoli bit in una sola operazione.

In particolare i registri PORT, LAT e TRIS hanno i corrispondenti registri SET, CLR e INV.

Cosa significa?

Un registro SET serve per settare (ovvero impostare a 1) i singoli bit, un registro CLR al contrario serve a resettare (porre a zero) i singoli bit e infine un registro INV serve ad invertirne lo stato.

Quindi ulteriori 9 registri: PORTxSET, PORTxCLR, PORTxINV, LATxSET, LATxCLR, LATxINV, TRISxSET, TRISxCLR e TRISxINV.

In realtà ce ne sono molti di più perchè i registri SET, CLR e INV sono disponibili anche per i registri di altre periferiche.

Il vantaggio è notevole.

Supponiamo, ad esempio, di voler fare il toggle di un singolo I/O (ovvero invertirne lo stato).

Ad esempio con RA0.

Un sistema potrebbe essere il seguente:

```
PORTA ^= 0x0001;
```

Questa operazione, però, prevede parecchi passaggi anche se in realtà noi lo stiamo scrivendo su una sola riga!

Il valore di PORTA viene prima letto, trasferito in ram, viene fatta l'inversione del bit sempre in ram ed infine il risultato viene trasferito al registro PORTA.

Sui PIC32 questa operazione possiamo continuare ad eseguirla così, nessuno ce lo vieta.

Stiamo lavorando con dispositivi che possono lavorare a 80MHz, e con un ciclo istruzioni uguale al ciclo di clock (FCY=FOSC), quindi condizioni pesanti, e allora sfruttiamolo.

L'inversione di un bit può essere fatta sui PIC32 sfruttando il registro INV che ci interessa:

```
PORTAINV=0x0001;
```

Questa istruzione viene eseguita in un sol colpo ed è quindi molto più rapida ed efficiente.

L'istruzione non fa altro che dire: inverti il bit 0 del registro PORTA.

Allo stesso modo operano i registri CLR e SET:

```
PORTASET=0x0002; // metti a 1 il bit 1 del registro PORTA
TRISBSET=0x0005; // metti a 1 il bit 4 del registro TRISB
LATCINV=0x9000; // inverti i bit 15 e 12 del registro LATC
PORTBCLR=0x0041; // metti a 0 i bit 0 e 6 del registro PORTB
```