

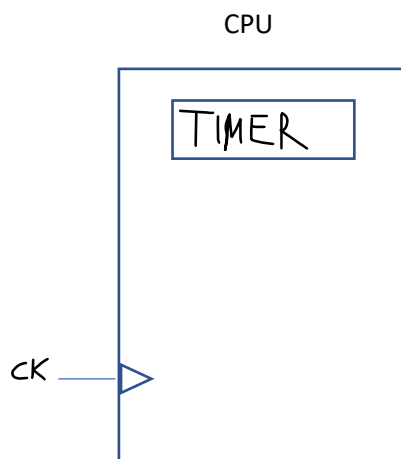
Rappresentazione del tempo all'interno dei sistemi di calcolo

Posso usare il clock del processore?

Arriverei a una misurazione del tempo precisa?

L'idea è quella di avere un contatore e ad ogni colpo di clock incrementare di 1 il contatore

Possiamo, dentro la nostra CPU, che ha l'ingresso clock, mettere un registro TIMER che verrà incrementato di 1 ad ogni ciclo di clock



Così, quando vado a leggere il contenuto del registro, so quanto tempo è passato dall'inizio del conteggio (so quanti cicli di clock sono passati)

Ma non in tutti i processori c'è questo registro dedicato, ci sono processori che non ce l'hanno, come facciamo ad implementare il conteggio del tempo anche su sistemi di calcolo con processori del genere? Dobbiamo usare delle tecniche alternative.

Inoltre un inconveniente dell'usare il clock del processore è che in diverse macchine (con processori moderni) la frequenza di clock non rimane costante: ad esempio ci sono macchine a batteria che, sotto una certa soglia di autonomia fanno in modo che il processore "viaggi" meno velocemente (quindi ad una diversa frequenza di clock) così da garantire di risparmiare energia, invece che garantire il massimo delle prestazioni (più veloce funziona il processore, più velocemente vengono eseguiti i programmi e quindi più velocemente si scarica la batteria).

Normalmente viene adottata un'altra soluzione infatti:
avere un dispositivo esterno che funzioni da "orologio".

Tale dispositivo viene chiamato RTC (real time clock)

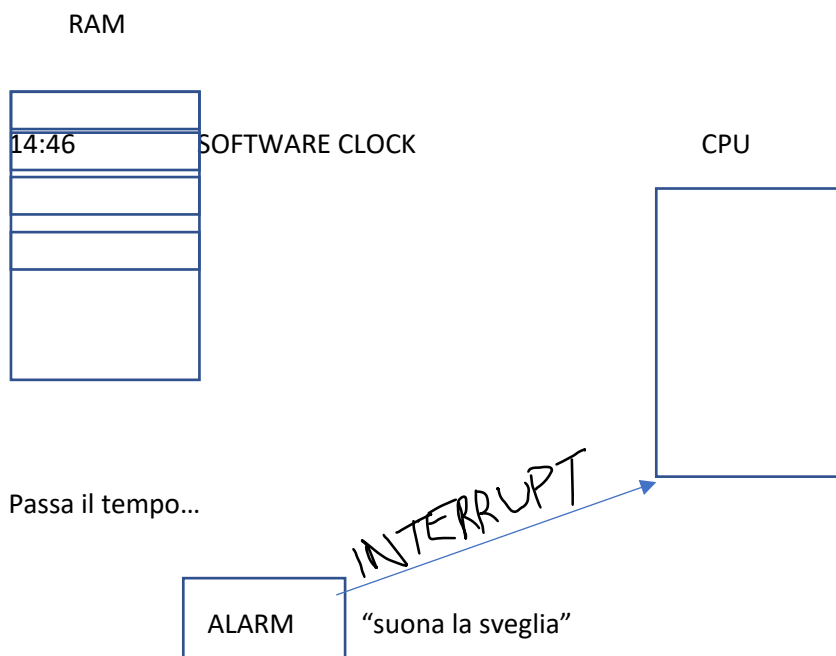
L'RTC è un normalissimo orologio digitale, simile ad un orologio da polso (senza display ovviamente perché non ci serve a niente): c'è l'oscillatore al quarzo, ci sono i contatori, ma l'informazione sui conteggi (sul tempo) rimane codificata in forma binaria all'interno dei contatori (non dobbiamo trasmetterla su alcun display, bensì usarla nel nostro sistema di calcolo).

Normalmente all'interno di un sistema di calcolo questi dispositivi sono dotati di alimentazione autonoma (così che si possa continuare a contare il tempo anche quando il sistema è spento).

Per leggere l'ora all'interno del nostro sistema di calcolo dobbiamo leggere il contenuto dell'RTC.

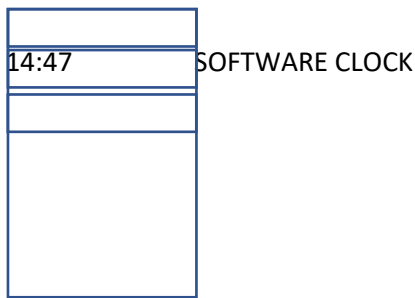
L'RTC non è un dispositivo collegato direttamente al BUS di sistema, ci sarà un considerevole ritardo nella lettura del suo contenuto, quindi se inizio a leggere al tempo x (ad esempio 12:20) ma finisco al tempo y (ad esempio 12:25) sto leggendo 12:20 ma in realtà sono già le 12:25.

Per questo motivo all'interno di un sistema di calcolo è previsto che ci sia anche un software clock, esso è un contatore che viene inserito ad un certo indirizzo della RAM, tale contatore viene aggiornato sulla base di un "temporizzatore", l'alarm: ipotizziamo che, nell'istante in cui lavoriamo, sul software clock (nella RAM) ci sia 14:46, settiamo l'alarm in modo appropriato (a un po' meno di un minuto), quando "suona" il timer andiamo ad incrementare il nostro software clock, scrivendoci 14:47.



Il software clock viene incrementato:

RAM



L'ALARM continuerà a mandare interruzioni a frequenze prestabilite e continuerò a incrementare l'orologio appena "arriva" l'interruzione

L'RTC allora a cosa serve?

A gestire accensione e spegnimento; nel momento in cui accendo la macchina faccio una volta sola la lettura dell'RTC: avendo letto il contenuto so cosa mettere all'interno del software clock: ci metto l'ora/data che ho letto dal RTC.

Quindi sappiamo che il tempo viene rappresentato attraverso un contatore (che sia nel RTC, nel software clock o nel registro timer della CPU, sempre di contatore si tratta), come do un significato a questo contatore? Da cosa parte? A che frequenza bisogna incrementare?

EPOCH è un punto di riferimento che mi dice "che ore sono" quando il mio contatore (software clock) segna 0, quindi mi permette di far partire il mio contatore in un momento ben definito nel tempo.

Esistono due standard principali per la rappresentazione del tempo:

nei sistemi UNIX like il contatore è formato da due contatori:

uno che conta i secondi e l'altro che conta i nanosecondi

32 bit

32 bit

In complemento a 2 (bit più significativo interpretato come bit di segno)

L'EPOCH è 1/1/1970

Quindi il valore 0 in entrambi i contatori significa le ore 00:00 del 1/1/1970.

Essendo a 32 bit vuol dire che il massimo valore positivo rappresentabile è 2^{31}

(1° gennaio 1970 + 2^{31} è la massima data rappresentabile: 1° gennaio 2038, dopodiché mandiamo in overflow il contatore, e ciò non ci permetterebbe più di rappresentare le date: prima di arrivare al 2038 si prevede di aggiornare lo standard passando a rappresentazioni a 64 bit, dopodiché non dovremmo più avere problemi siccome la massima data rappresentabile sarebbe qualcosa intorno al 1970 + 9,223,372,036,854,775,808 ossia 9 trilioni 223 miliardi 372 bilioni 36 miliardi 854 milioni 775 mila 808 secondi, quindi 285,388,127,856 anni + a questo punto gli insignificanti 1970 anni dell'inizio: una data in cui

probabilmente non esisterà più il sistema solare per come lo conosciamo a causa dello spegnimento del sole per esaurimento di combustibile).

Quindi il valore 0 in entrambi i contatori significa le ore 00:00 del 1/1/1970

A cosa serve il bit di segno? Permette di rappresentare date precedenti al 1970.

Altro standard: UTC (implementato anche dal Network Time Protocol)

L'UTC funziona in maniera simile allo standard UNIX: c'è un'epoca, ci sono rappresentazioni a 32 bit per sapere quanto tempo è passato dall'epoca fino al momento x, quali sono le differenze con UNIX standard?

- La rappresentazione dei secondi NON è in complemento a 2, bensì unsigned (sono solo numeri positivi, non bisogna considerare il bit più significativo come bit di segno, quindi effettivamente abbiamo un bit in più per la rappresentazione del valore massimo, quindi pur partendo da una data antecedente rispetto a UNIX (1900 vs 1970) si arriva più o meno alla stessa data massima)
- L'epoca di UTC è 00:00 del 1/1/1900

Ovviamente anche qui dopo il 2038 bisognerà passare a una rappresentazione a 64 bit.

Ma perché sia nello standard UNIX che nell'UTC si hanno secondi e nanosecondi?

Per sapere quale è l'ora devo leggere come leggersi un orologio (ore e minuti->secondi e nanosecondi: se trovassi 5s e 3 ns significherebbe 5,000000003)

Rappresentazione broken down: suddividere in anni, mesi, giorni, ore, minuti, secondi, frazioni di secondo

Per ciascuna di queste quantità avremo una rappresentazione binaria.

Prendo il valore rappresentato e lo divido per il numero di secondi in un anno, così so quanti anni sono passati, il numero di secondi in un anno posso dividerlo per 12, per sapere quanti secondi ci sono in un mese, poi per 28,29,30 o 31, poi ci sono le ore (dividere per 24), i minuti (dividere per 60), secondi (dividere per 60), frazioni di secondo (dividere per 10, 100 o qualsiasi frazione vogliamo).

La libreria POSIX ci offre delle funzioni predefinite che prendono la rappresentazione in formato s ns e le trasformano in una rappresentazione broken down.

Dalla rappresentazione broken down si arriva a quella testuale.

Funzioni predefinite che useremo:

- la funzione che mi permette di andare a recuperare il valore del software clock
- funzioni di conversione dal formato s ns UNIX STANDARD al formato broken down
- funzioni di conversione da broken down a testuale GMT