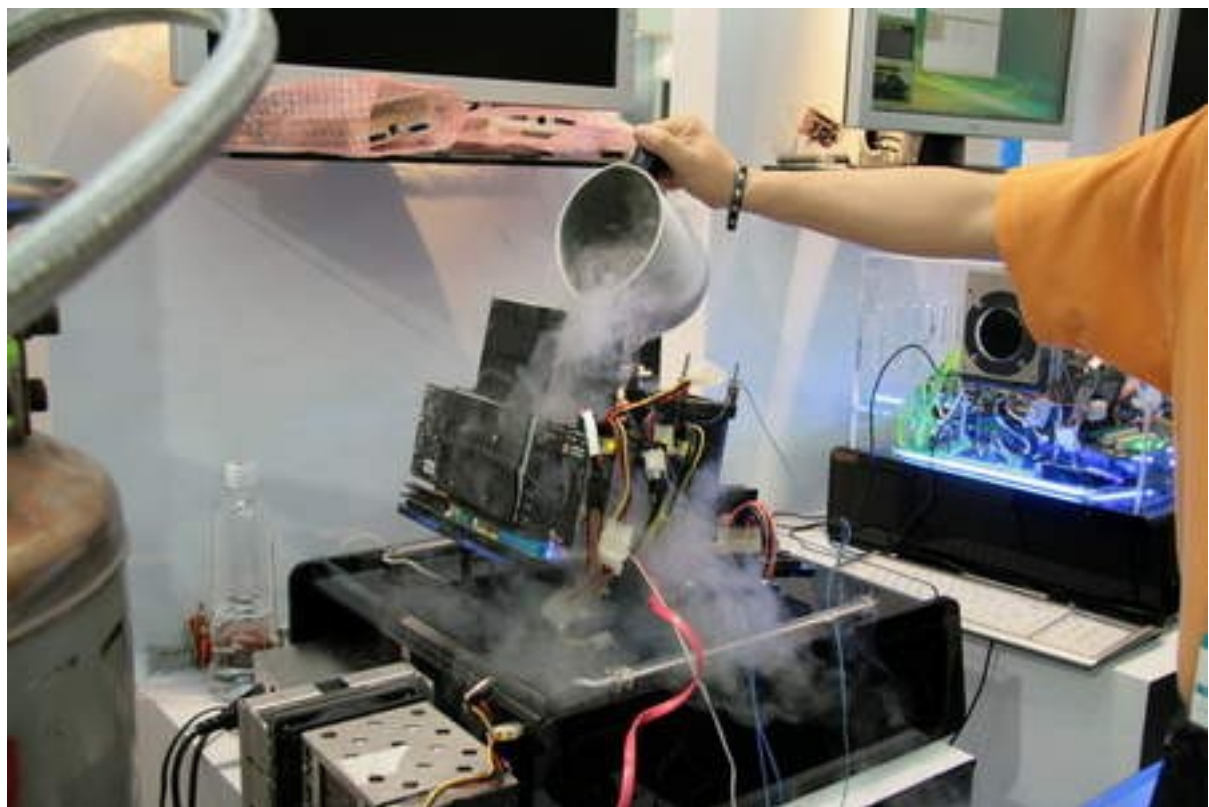


RASPBERRY pi 4 di

(overclocking)

Augusto Scatolini (webmaster@comunecampagnano.it) (a.scatolini@linux4campagnano.net)
Miniguia n. 175
Ver. 1.0 Settembre 2012



In *informatica* la pratica dell'overclocking consiste nel perseguire il miglioramento delle prestazioni di un componente elettronico (in genere una CPU) mediante l'aumento della frequenza di clock rispetto a quella prevista dal produttore, marchiata sul contenitore della CPU.

Nel caso delle CPU, di cui si parla in generale ai soli fini esemplificativi, ciò che viene alterato è il segnale di clock interno del sistema, che determina quanti cicli di operazioni la CPU del computer debba eseguire nell'unità di tempo. Cambiando la frequenza alla quale questi cicli vengono eseguiti, in alcuni casi, è possibile aumentare o diminuire il numero delle operazioni nell'unità di tempo eseguite dal componente coinvolto.

A livello pratico si tratta di una operazione perfettamente legale ma che invalida istantaneamente la garanzia del prodotto oggetto della manipolazione. L'overclocking può essere eseguito in modo più o meno spinto.

<http://it.wikipedia.org/wiki/Overclocking>

Le prestazioni del Raspberry pi sono quelle che sono, limitate ma sempre più che dignitose considerando le caratteristiche della scheda madre, del processore a 700 Mhz e della RAM (soli 250 MB). E' comunque possibile migliorare queste prestazioni con la tecnica dell'overclocking. (senza esagerare)

In termini più precisi, fare un overclocking significa aumentare la frequenza di lavoro di un qualsiasi componente che ne abbia una, come ad esempio un processore. La frequenza con cui lavora un moderno processore è determinata da due parametri: la velocità ed il tipo di bus, che potrebbe, ad esempio, essere di 200 MHz) ed il moltiplicatore di tale parametro, nel nostro caso esemplificativo, pari a x10. Dato l'esempio, la frequenza di lavoro del processore preso in esame sarà di 2000 MHz (bus * moltiplicatore).

In generale è possibile praticare l'overclock di tutti i componenti digitali in sistemi asincroni, composti per meglio dire da sottosistemi che operino quindi a velocità diverse tra loro e non uniformi. Nello specifico del nostro esempio, un qualsiasi sistema informatico ha le porte USB, la memoria di sistema, la memoria della scheda grafica, la CPU ed il bus di sistema che operano a velocità diverse: è l'architettura che procede poi a bufferizzare e gestire in modo del tutto trasparente le transizioni delle informazioni tra un componente e l'altro. In sistemi sincroni, invece, esiste una sola velocità di clock alla quale sono legate tutte le componenti del sistema: modificarla significherebbe agire sull'intero sistema, cosa che evidentemente non è possibile data la nativa eterogenea delle componenti in questione.

Tornando ad un personal computer, di solito si usa praticare overclock sulla memoria e la GPU della scheda video, nonché su memoria, bus e CPU del sistema, tutto ai fini di migliorare le prestazioni.

<http://it.wikipedia.org/wiki/Overclocking>

Installiamo sul Raspberry pi uno strumento come **bc** per fare un benchmark

GNU bc è un linguaggio algebrico interattivo a precisione arbitraria conforme alla bozza dello standard POSIX 1003.2 con parecchie estensioni come i nomi delle variabili multicarattere, il comando "else" ed espressioni booleane complete. GNU bc non necessita del programma GNU

<http://packages.debian.org/sid/bc>

sudo apt-get install bc

eseguiamo il benchmark con il comando

time echo "scale=3000;4*a(1)" | bc -l con i parametri di default che sono

CPU=700MHz RAM=400MHz GPU=250MHz

risultato:

real 1m23.064s *pari a 1 minuto, 23 secondi e 64 millesimi di secondo*
user 1m16.820s *pari a 1 minuto, 16 secondi e 820 millesimi di secondo*
sys 0m0.060s *pari a 0 minuti, 0 secondi e 60 millesimi di secondo*

modifichiamo i parametri aumentando CPU, RAM e GPU nel file `/boot/config.txt` come superutente

CPU – 700MHz → 900MHz
RAM – 400MHz → 500MHz
GPU – 250MHz → 275MHz

salviamo, riavviamo il sistema e ripetiamo il test

risultato:

real 1m23.064s → 1m2.868s *pari a 1 minuto, 2 secondi e 868 millesimi di secondo*
user 1m16.820s → 0m58.660s *pari a 0 minuti, 58 secondi e 660 millesimi di secondo*
sys 0m0.060s → 0m0.050s *pari a 0 minuti, 0 secondi e 50 millesimi di secondo*

è semplice verificare che c'è stata una riduzione di tempo **real** del 24,3%, **user** del 23,6% e **sys** del 16,7% e quindi un significativo aumento della velocità

real: è il tempo di clock che passa dall'inizio alla fine della chiamata

user: è il tempo della CPU trascorso in user-mode (fuori dal kernel)

sys: è il tempo della CPU trascorso dentro al kernel.

	A	B	C	D	E	F	G
1	real	1m23.064s	1m2.868s				
2		83,064	62,868	100	75,6862178561	24,3 %	
3							
4	user	1m16.820s	0m58.660s				
5		76,820	58,660	100	76,3603228326	23,6 %	
6							
7	sys	0m0.060s	0m0.050				
8		0,0600	0,0500	100	83,3333333333	16,7 %	
9							
10							
11							

FINE

Questo documento è rilasciato con licenza Copyleft
(tutti i rovesci sono riservati) altre miniguide su
<http://www.comunecampagnano.it/gnu/miniguide.htm>
oppure direttamente su <http://miniguide.tk>
sito consigliato: <http://www.linux4campagnano.net>
blog consigliato: <http://campagnano-rap.blogspot.com>