

小華的部落格

將自己踏入BIOS領域中所學習到的知識做一些心得整理，像是Legacy BIOS、EFI BIOS、Windows Driver...etc. ※版權與智慧財產權聲明:保留所有法律權利。我在寫文章時如果有引用到其他人的地方我會盡量說明參考出處，如果有遺漏的地方請告訴我，我會馬上註明! 而轉貼我的文章時也請您註明出處!

搜尋

首頁

About Me

總網頁瀏覽量



1 7 7 6 2 4
5

訂閱小華的部落格

發表文章



留言



訂閱

請輸入您的email address:

星期四, 9月 27, 2007

[我所知道的BIOS]->[Jumpless] 4

上一篇所提到的是:BIOS entry point. 它是第一個被 CPU 所抓取並執行指令之處.每家BIOS都有其相應的 file,要牢記.

在本文中,要提及: Jumpless ! 在此要先說明Jumpless的意義.它代表:"用 s/w方式來 config clockgen generator,使之產生正確的 頻率s 來供應系統運作". 重要性在於:假如某個 frequency錯了,則將導致 system abnormal or malfunction...

有哪些 frequencies由 clockgen 提供?

- CPU clock (在此指的是 : host clock, Ex. FSB800 CPU 需被 supply 200MHz clock)
- DRAM clock(depends on chipset design; Ex. DDR 533要被 supply 266MHz clock)

更正 : DRAM clock現在都是由NB產生 . 外部的最多是 buffer而已. 並非 clock generator. 原因是為了保持較良好的 host clock & DRAM clock的同步.

補充:我舉的例子是以以前的chipset design. 現在我做過的 chipset 其 DRAM clock的確由北橋來推. 北橋中有一會 register定義: CPU-DRAM ratio.

Ex. CPU clock = 100MHz. Ratio = 3:5,則可得 DRAM clock = 166MHz

訂閱電子報

EZMAIL提供

Translate

选择语言 | ▼

網誌存檔

- ▶ [2020](#) (1)
- ▶ [2019](#) (2)
- ▶ [2018](#) (3)
- ▶ [2016](#) (2)
- ▶ [2015](#) (1)
- ▶ [2014](#) (8)
- ▶ [2013](#) (3)
- ▶ [2012](#) (12)
- ▶ [2011](#) (19)
- ▶ [2010](#) (20)
- ▶ [2009](#) (11)
- ▶ [2008](#) (35)
- ▼ [2007](#) (59)
 - ▶ [12/23 - 12/30](#) (1)
 - ▶ [12/16 - 12/23](#) (4)
 - ▶ [12/02 - 12/09](#) (1)

- AGP clock: 標準 is 66.6MHz (= 1/2 PCI clock)
- PCI clock: 33.3MHz
- SATA clock: 100MHz(see SATA spec,似乎還要更精確...)
- Link clock: 即假如NB/SB中間有 link,亦由clockgen提供
- USB 1.1 clock: 12MHz

更正: USB 的 clock 是 48MHz

補充:我們 chipset的 design是: 由 clockgen提供 12MHz clock(或由 external crystal提供),被 PLL吃進去後,會產生 48MHz給 USB 1.1(你說的應該是這部分). 至於 USB 2.0,則是 supplied other frequencies !

- SIO clock...etc

Clock Generator的架構

=> ClockGen基本組成為 PLL(Phase Locked Loop) + Div. PLL 的輸入為 14.318MHz, 利用 迴授方式產生出 frequency F;之後 經過 Div 除頻,便得到 desired frequency,見下列圖示:

14.318MHz -> PLL -> F -> Div -> f

Ex. F = 400MHz, 為 PLL振出來的;若 Div = 4,則 f = 100MHz

ClockGen的 spec中,會列出許多 registers,好比是 Function 的 "contact window",透過這些 registers可以設定此 clockgen !

更正:其實PLL 不是單純一個的 div. 鎖相迴路的參數是兩的divider 組成 M(VCO divider)/N(REF divider)

補充:觀念上來說是 PLL所振出的頻率經 div後可得到 desired frequency. 但在 clockgen中:

1. 有 2 bytes(即所謂的M,N)來決定 VCO
2. 有幾個 bits來決定 div;一些主要的 frequency都有相對應的div
3. $VCO / div = frequency$

How to access ClockGen

=>現今存取 clockgen的方式,都是透過 SMBUS來達成. SMBUS 有分 1.1 and 2.0,不過大同小異. SMBUS是 2-wire protocol;由 SMBUS master(即內建在南橋內部的 SMBUS controller)來下 command 給 SMBUS device(Ex. Clockgen);Command中常用的有 Block Read/Write & Byte Read/Write

不同家 chipset 的 SMBUS controller implement方式不同,自然下command的方式亦不同;主要的步驟是:

- pre-init(clear status or ...)
- put slave address(要 access 的對象)
- decide Read or Write
- put command(Protocol type, Ex. Block or Byte access)
- put offset(要存取 自 clockgen中的哪一個 Byte開始...)

► 11/25 - 12/02 (2)

► 11/18 - 11/25 (1)

► 11/11 - 11/18 (3)

► 11/04 - 11/11 (4)

► 10/28 - 11/04 (4)

► 10/21 - 10/28 (2)

► 10/14 - 10/21 (2)

► 10/07 - 10/14 (7)

▼ 09/23 - 09/30 (6)

[我所知道的BIOS]->
[Chipset Config] 5

[我所知道的BIOS]->
[Jumpless] 4

[我所知道的BIOS]->
[PowerOn Sequence
& BIOS Entry] 3

[我所知道的BIOS]->[系統
架構] 2

[我所知道的BIOS]->[前
言] 1

PCI IRQ Routing Table
Specification

► 08/26 - 09/02 (2)

► 07/29 - 08/05 (3)

► 07/08 - 07/15 (1)

► 07/01 - 07/08 (3)

► 06/17 - 06/24 (2)

► 05/27 - 06/03 (3)

- put data if (Write device)
- start transaction !!!
(... processing ...)
- get "Complete" status to check if transaction is done successfully
(...wait for system reset...)

* 當 start transaction後, SMBUS controller便會將 programmer所 prepare的資訊將之轉成 SMBUS command打給 device;完成後應該在 controller端有 status register可以檢查,看看是否 transaction ok or failed,或是有其他 error conditions.

透過這樣的方法,programmer便可以 config clockgen使之產生正確的 frequencies.

[2 Programming ways for clockgen]

=> 現今的 ClockGen有提供兩種方式 for S/W config. 一為 table mode,一為linear mode. Table mode指的是: ClockGen spec 中會有一個 table,裡面說明: 要產生 ? frequency 要填 ? 值.因此,這種方式只需program 1 byte to Clockgen就行了. 若是 linear mode,則是坊間所說的無段變頻,即可以微調frequency. Ex. table mode只提供 100/133/166/200MHz,但 linear mode則可以提供類似 102.6/136.3/168.7..等的頻率. 此時,要program的 byte數就很多了.....除此之外, linear mode也可以config 所謂的 spread spectrum(%),板廠測EMI時會用到

【Notes】

1. For NB platform power-on CPU frequency 由 2 pins來決定;DT platform會由 Jumper決定.不管如何,之後做 Jumpless都可以 override掉原 clock settings

更正: 不是只有NB如此.不管DT 或server. CPU FSB 也是 pin(由於FSB頻率比NB多.所以是 3 pins)來決定的. 不是jumper. 即使要做超頻設計會對這三pin做手腳, 也不會使用jumper

2. 有些 clockgen所提供的 SS 都是 center spread. 若需要 Down spread則可以由 center spread搭配 改變後的 frequency(類似直流準位)來達成

給我正確的頻率,其餘免談 ...^_^...

=====
Q&A
=====

>前輩我有幾個問題要發問!

>1. 你這邊說的Table方式頻率是指供應給CPU頻率嗎? Intel網站的CPU說明, 他說是固定的不可調的? 所以這不是HW固定住(Latch)設定值了嗎?

>

- 05/06 - 05/13 (5)
- 04/29 - 05/06 (1)
- 04/22 - 04/29 (2)
- 2006 (1)

逛逛不一樣的地方

演算法 (影像處理, 資料結構, 智慧型視訊分析, 人工智慧)

平凡的幸福

相關資訊

流浪小築

旅遊美食~

小君君的祕密花園

繼續閱讀懶人加強版

幸福雅痞~

懷舊系列~

標籤

一些筆記 (10)

分享 (2)

心情分享 (3)

生活運用 (1)

>2. 可以解釋一下Down Spread/Center Spread /...所代表的意思嗎? 還有用途?

>

>3. 我目前也在嘗試Program ClcokGen(新手上路), 很多專有名詞還沒搞懂, 另外目前ClockGen不是可以輸出不同頻率嗎? 為什麼Table方式只需要Program 1 byte就好 ???還是說各家廠商設定方式不同, 有些廠商只要利用1 Byte 就已經選擇好所有的頻率了(33MHZ/48MHZ/DOT96MHZ/LCD27MHZ...)?

[For #1]

=> 我所謂的 "Table"方式指的是: 為了方便設定,clockgen designer會專門提供一個byte register(8-bit);此 byte的某些 bit fields(Ex. bit[4:0])可以讓programmer填一組值,例如 01001b. 此值 write 至 clockgen成功後, CPU/DRAM/AGP/PCI/...etc 都會是固定的頻率. Ex. 100/133/66/33/...etc. 所以才說只填 1 byte only !

此值一旦填至 clockgen, clockgen便依據該 byte(可參考 spec前幾頁,會有一張表說明: bit[4:0] <-> frequency combinations)來產生出 frequencies.

不管用哪一種方式(table or linear mode),都會 supply CPU frequency !!!不過,此時的 CPU frequency is CPU的外頻 (CPU 總頻率(core) = (外頻)*(倍頻 <-multiplier))

Ex. 外頻 200MHz(Merom CPU, FSB 800),倍頻 = 12 -> core freq = 2.4GHz

還有,BIOS可以調整 外頻 & 倍頻 ! 沒問題 ! 不過,若遇到鎖頻的CPU(Ex. 量產版的CPU),倍頻可能無法調整.但外頻還是可以靠 config clockgen來達到 !!!

For #2

=> 假設要達到 100MHz 且 SS = 0.5% center spread.則:
(...想像以前學的直流與交流的訊號...)

100MHz 相當於 直流的部分,其值不隨時間而變; 0.5% center spread好比弦波訊號(振幅是 $100M * 0.005 * (1/2)$); 所以,兩者的合成便是: 以 100MHz為中心,其值隨時間而變,振幅 is $100M * 0.005 * (1/2)$ 的訊號 !!!

我 '猜' 要這樣測試的原因是:所產生的 clock一定不會是 perfect的 100MHz,有可能因為元件或是外部干擾而被 "改變".為了模擬這樣的情況,我們會將 clock的 spread spectrum enable,產生 "隨時間而改變"的頻率,來測試系統的穩定度...等

Down spread 0.5% 指的是:距離 水平 100MHz, 最低的值是 $100M - 100M * 0.005$,最高值則是 100M本身(等同於: 水平值 $99.5 + \text{center spread } 0.5\%$, right ?)

For #3

=> 前面提過,只設 1 byte 就可以產生所有頻率是因為: clockgen公司已經把設定各頻率的功能"濃縮"在一個 byte 裡面 ; programmer只要 config this byte 就可以了...

其它 (9)

思念 (1)

音樂分享 (1)

音樂歌詞 (1)

組合語言Assembly (4)

軟體工具 (12)

網路遊戲 (2)

攝影 (1)

AD (2)

BIOS 開發 (6)

BIOS相關 (21)

C 語言相關知識 (9)

EDK2 (1)

EDKII (1)

EFI BIOS相關知識 (23)

EFI教學 (2)

IA32 相關基礎知識 (27)

Windows 程式相關 (22)

* 關於 EMI 測試部分,有錯請指正. (...我有用到 "猜" 字眼...^_^)

[[EMI 測試的知識]]

由於 對clock信號來說. 鋒(peak)值是能量最強的地方. 所以peak的能量就會不斷累積, 發射出來. 因此在頻譜上就會出現能量很強的頻率. EMI 工程師的工作就是要想辦法屏閉這些電磁波的干擾.

為什麼需要 spread spectrum? 由於PLL 十分精準. 也就是說 100MHz的clock, 其 peak & peak之間的距離一致(波長固定). 所以能量十分驚人. 使用展頻可以使 peak 的距離改變(由於頻率不斷的在一個範圍內改變, 這個可以用示波器累積clock信號來看) 所以累積的能量會比較低. 這樣EMI 可以比較好處理.

至於為什麼會有要求BIOS一開始就把展頻開著, 以確保板子的穩定? 很簡單. 不開展頻clock很穩定. 自然比較不會有問題. 展頻一開. 由於頻率是浮動的. 很多設計欠佳的板子就會不穩定. 及早發現才能及早修改.

[[補充資料~~運作原理]]

CPU本身有預設值的BSEL[2:0]值, 這個值是與Clockgen所產生的BCLK是對應的, CPU本身Spec會有一個Table定義這個值, 然後CPU透過BSEL[2:0](輸出pin) 連接到Clockgen與NB(這兩個都是輸入pin), 所以當Power on的時候, Clcokgen會依照CPU預設的BSEL[2:0]去產生BCLK給CPU, 而FSB 頻率也是經由CPU接到NB的BSEL[2:0] 來同步(一般都是1/4 CLK 提取一次資料, 所以假設BCLK=200MHZ, 而Clockgen輸出到NB的Host Clk=200MHZ, 一般都會同步, 所以 FSB=Host clk*4 = 800MHZ)

※ Clockgen 分別透過BCLK與HostClk pins輸出頻率給CPU與NB

然後CPU開始執行BIOS Code, 當執行到Program Clockgen時, BIOS可以透過Clockgen內的暫存器的某個Byte 內的FSC/FSB/FSA 這3個bit 來改變BCLK輸出頻率(應該可看作override BSEL[2:0]設定), 當CPU接收到新的BCLK頻率後可能的反應有:

1. CPU本身不支援這個BCLK所以系統當機或是Shutdown
2. CPU 可工作在這個BCLK頻率, 依照這個頻率對照Table, 然後改變BSEL[2:0]的設定

另外CPU 是否能工作在266Mhz or 200Mhz or both, 這部份依照每顆CPU不同而有所不同, 有些CPU只能工作在266Mhz, 所以當BLCK=200Mhz時, CPU就不會工作.

假設CPU可工作在266Mhz與200Mhz, 則當BCLK從200MHZ-->266Mhz時, CPU一樣可以工作.

還有CPU內部的調整ratio的MSR暫存器可分成測試版的CPU跟出貨版CPU的不同:

- a)測試版CPU的Ratio可調, 如何調要看Spec
 - b)出貨版CPU不可調, 也就是鎖倍頻
-

1. Clockgen 會送兩組 differential-pair clocks給 CPU & NB. 目前這兩組 clocks是相同的頻率.

2. CPU有兩 pins 稱 BCLK[1:0]. Input.

NB HostClk[1:0]. Input

* bit[1:0] 是代表: differential-pair clock都會有 2 clocks,分別是 true clock and complementary clock

3. 開機時的 behavior 是:

1) CPU內部自己會產生一組 default BSEL[2:0] 給 Clockgen的 FSC/B/A (依據經驗,這組 default 值 可以由 CPU MSR 得到 ; 也等於這顆CPU所能支援的最大外頻,BCLK)

2) Clockgen據此 FSC/B/A會產生對應的頻率,送給 CPU 的 pins BCLK[1:0]

Ex. 266MHz

3) 之後,若有需要, BIOS可以透過 config FSC/B/A 或是 M/N programming(透過clockgen內部的暫存器)來 "override"掉原來的 clock setting以達到改變 BCLK的目的.

CPU 工作頻率=外頻 (BCLK)* 倍頻(MSR調整)

FSB 頻寬= CPU Spec說由 BSEL[2:0] "同步決定", 但是看Clockgen電路有分成, BCLK與HostClk(兩者應該輸出同樣頻率) 所以應該是北橋拿HostClk當參考, 每1/4 Clk提取一次Data, 所以應該是HostClk * 4

以前FSB頻寬=FSB工作頻率, 所以有人把CPU工作頻率計算方式看作 " FSB頻率 * 倍頻 ", 但是由於FSB頻寬目前已經不等於FSB 工作頻率, 且Clockgen 分別拉了兩組pins(BCLK與HostClk)給CPU以及北橋, 所以計算CPU工作頻率的方式才會變成目前的 "BCLK * 倍頻"

加註:

1. FSB1066指的是: bandwidth ! 不是頻率. 一般不會說 FSB266,而會說 host clock = 266MHz !

2. *4 應該是因為 (clock的上下緣都可以觸發) + (differential-pair) 的關係

我先解釋一下這三pin,就我印象中除了之前 NV chip + K7 和 SiS的chipset以外.

CPU 的FS0, FS1, FS2會接到clockgen (FSA,B,C)和NB (BSEL0,1,2)的對應pin.

CPU 是負責產生準位, clockgen & NB 會在自己的powerok ready的時候把資料latch進去.

clockgen 除了利用這三pin來產生對應頻率的CPU clock外, 另外也會根據這個頻率來產生合適的PCI-E, PCI clock..., 而北橋則是利用來產生正確的 DRAM clock(host/dram ratio),對intel來說也有PCI-E的同步關係.

至於超頻跟這三pin 的關係, 我稍微解說一下.一般而言要超頻的話, MB 必須把到NB & clockgen的路徑斷開, 由另外設計的線路來取代CPU的FS0,1,2.也就是說NB & clockgen 一開機必須收到一個由這個控制線路決定的FSB. 這樣才能作所謂的 133-> 200 or 200 -> 266的大幅度超頻.因為此時NB 和 clockgen 才會正確的分配. 別忘了NB, SB, PCI-E, PCI, DRAM是沒有CPU這麼強的超頻力. 直接利用軟體改變clockgen會造成很多不該變化的頻率跟著變.

Note:此超頻線路, default是一個所有CPU都能開的安全值, 一但BIOS完成設定, 就會latch住資料

此外還必須配合watchdog 來處理過度超頻不能開機的問題. linear mode只用來處理小超頻的部份

以前遇到的 clockgen 是 CPU & DRAM clocks 由同一個 PLL決定,其餘 clocks由另一個PLL決定. 當時有寫一個 OS下的小程式,可以微調 CPU頻率(以1MHz為單位). 不過,微調CPU的同時也會微調到DRAM clock(<-同一個 PLL),只是微調值不同(depends on divider).

現在遇到的 clockgen + chipset design 是: CPU由一PLL供給,而DRAM clock由北橋推.同樣的,微調CPU clock一樣牽動DRAM clock.

至於其它 frequencies是否會跟著變, depends on Clockgen design;現在的clockgen幾乎都有 2以上的PLLs;所以類似PCI/AGP/SATA clocks都會有專門的 PLL控制. 而CPU則由另一個PLL控制.

有興趣及時間的人可以試試看 M/N programming(小微調). 看著 (Ex. FSB800 CPU)由200MHz 操到 210MHz,也是一種快樂 ^_^

Q: 這邊剛好有提到watchdog...想請問一下前輩這個東西在PC系統上是何時開始使用? 用途為何或是應用為何? 以前學校沒看到過這個東西, 不過我在嵌入式系統常看到一些應用, 另外在南橋Spec有看到過, 而BIOS端需要如何去應用以及設定它? 另外使用 watchdog 來處理過度超頻不能開機是怎樣的一個方式? 頻率不對, CPU已經無法work, 連BIOS code都不能執行了? 所以他是靠HW去處理嗎?

A:我曾用過關於watchdog timer 的經驗是 => 在 config clockgen完後, 需要 system發 PCI reset ! 此時會需要 watchdog timer...

因為, write clockgen完後,有可能系統當掉 !!! (<-frequency settings被改變了...)因此,我知道的作法是: 在 write clockgen之前,先 enable watchdog timer ! (當然,一定會指明要"多少時間後"才發 PCI reset !) — enable完,便匆匆去 write clockgen(當然,這段時間要夠長,以免 write clockgen沒完成,PCI reset就出來了 >_<).

一旦 write clockgen完畢,code會寫 "jmp \$"...然後等 watchdog timer 時間到,然後PCI reset出來 reset system...

你所說當機時的 watchdog timer,我沒做過 ! 不過,watchdog timer 不一定要在 SB內啊!(或是即使做在 SB內,也是"獨立的"...)做成外部的電路就不怕 chipset hang時 timer會失效了...^_^

一般我知道的 watchdog timer control 大概有:

- enable/disable (=> 開始記時...)
- time to bomb (=>多少時間後爆炸 ?)
- event routing (=> 時間到要幹嘛 ?)

[[WatchDog]]

watchdog是個硬體機制.

watchdog 在嵌入式應用中主要就是用來做系統自動回復用的.通常用在機器處理很critical 的任務, 不能當機超過一定時間的事務. 一般的用法是, 系統的軟體要固定時間去重設timer. 例如: watchdog timer 設定10秒以後會重置系統, 那麼系統軟體可以每5秒有一個task 去重設timer. 萬一系統當機了. 由於軟體無法重設timer. timer 數完了以後就會發出reset , 讓系統重新恢復狀態. 在MB 中superI/O 有watchdog, clockgen 也有支援watchdog. 以超頻來說, 往往設定完clock generator的瞬間, 系統可能會當機. 這時有兩種狀況. 1. 因為clock 變化造成, 這種狀況只要一個PCI reset. 系統就可以繼續執行. 這時BIOS 可以借用 watchdog來發PCI reset. 只要在定clock之前先設好 watchdog timer, 然後在一段時間後關掉watchdog timer. 這樣就可以解決這個問題. 一種是 clock 增加太多造成 chip 死當. 這種就需要依靠 clock gen 裡面的watch dog來讓clock 回復成default value(Note:要視clock gen的設計, 有些只是發reset而已).

另外一種我知道的應用是雙 BIOS的主板. 利用watchdog來切換BIOS.

張貼者: 小華的部落格



標籤: [BIOS相關](#)

5 則留言:

匿名 提到...

本身非BIOS相關,但在系統中也用了MS Debug去寫一些basic tool或查看一些GPR..看了你的blog之後,真是受益良多!

自己也想和你分享一下有關FSB concept.

你的Blog提到下述:

>加註:

2. *4 應該是因為 (clock的上下緣都可以觸發) + (differential-pair) 的關係

就我了解FSB是Source synchronous timing, 所以最後是以data strobe 來抓資料的,流程應該是 Clock gen. issue CPU CLK+/CLK- to CPU and HCLK+/HCLK- to North Bridge, then their internal Clock Distribution(PLL) will produce the new bus clock($200\text{MHz} \times 4 = 800\text{MHz}$). The CPU/chip internal use bus clock to trigger the Data and Data Strobe.然後receiver用Strobe來latch data, Strobe 需加delay circuit to guarantee the Setup/Hold time. Delay一般會加在比DCLK 延遲half cycle處,所以看起來有一點像是Bus clock的falling edge來觸發, 但事實上它是因Delay circuit所致.

Summary: BCLK = System clock = 200MHz

DCLK = Bus clock = $200 \times 4 = 800\text{MHz}$

It also called as "Quad Pumped"

FSB是單邊觸發 也就是Only Rising or Falling edge, 上下緣都觸發應該是DDR Pre-fetch 的 concept, 而differential-pair 只是讓Clock進入chip之後能有好的waveform, (CLK- CLK-) it will produce 2倍的 Amplitude, 觀念就像為何DDR2 667之後就可能要求要Differential Strobe (DQS+/DOS- pair)是一樣的.

煩請指教!!

3月 16, 2009 9:09 下午

Unknown 提到...



我試著去掃clockgen的值, 但是我調整FSB的值好像會造成clockgen的值的改變, 是我有什麼地方疏忽了嗎? 另外我查了ICH的spec, clock generator似乎不是SMBus控制的, 愈看愈不懂的感覺@@

1月 11, 2010 2:57 上午

Unknown 提到...



你好, 我試著掃CLOCKGEN值, 但是我改變FSB時, CLOCKGEN的值不會跟著改變 (我確定程式是沒有問題的, 因為可以看到其它SLAVE的值); 另外我查了ICH的SPEC, CLOCK GENERATOR似乎不是由SMBUS控制的, 可以幫我解惑一下嗎, 謝謝。

1月 11, 2010 3:00 上午

Grant 提到...



我現在在開發一個平台
是AMD L325+AMD R780+AMD SB 710 (Clock 為 ICS9LPRS471C)
不管我插667/800Mhz DDR2
AMD CPU振出來的Dram Clock都只有300MHZ(是AMD CPU 限制)

我想超頻

把AMD CPU振出來的Dram Clock 超頻至333

該如何調整Clock的M/N 值??

我不懂超頻, 也不懂Clock Gen M/N值的調法

如果您懂Clock Gen 的M/N 公式 或設定方法, 請告訴我這笨笨的 軟體 BIOS engineer

不懂硬體的Grant留

3月 11, 2010 1:21 下午



Grant 提到...

上面的問題補充

因為AMD CPU振出來的Dram clock只有300MHz(Limitation)
目前DDR2 667/800 Ram 不穩

AMI BIOS engineer建議

1. 降頻跑533Mhz
2. 將CPU Dram超頻至333/400Mhz.

但我不知道第二項如何Implement?

這次希望大家不吝回覆

Grant

3月 11, 2010 1:52 下午

[張貼留言](#)

[較新的文章](#)

[首頁](#)

[較舊的文章](#)

訂閱: [張貼留言 \(Atom\)](#)