那些讓電腦變快的方法

暫存器

快取:(L1-L3:SARM)

主記憶體(DRAM)

磁碟儲存

Cmos組成記憶體，不需要邏輯閘的想法，電路更精簡

記憶體是用暫存器做的 由於記憶體只有24K+1字組，容量很小只能作為嵌入式的小控制器

DRAM裡面是用電容

目的->用DRAM讓電腦記憶體變大，成本不能增加太多

HackCPU問題:沒有乘除法，沒有浮點數的硬體電路

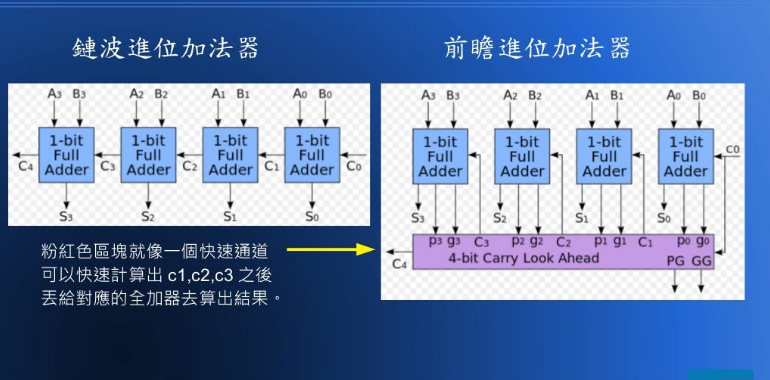
乘法器用加法器+移位運算

除法用減法器+移位運算

verilogHDL

HackCPU的ALU使用鏈波進位加法器，前一個算完後才開始下一個

只要改用前瞻進位加法器，速度會快數倍



常見加速方法

1. 快的材料取代慢的
2. 平行工作

1.

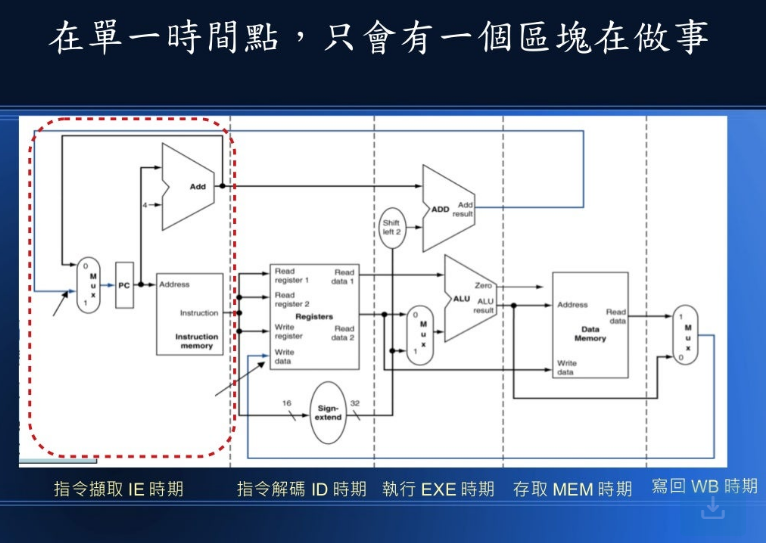
快取加速->加入適當容量快取(現代電腦來說特別重要)->增加快取命中率

區域性策略(一次載入一整塊)

記憶體速度>硬碟速度->盡量減少慢速裝置的存取

2.平行機制

管線pipeline機制:

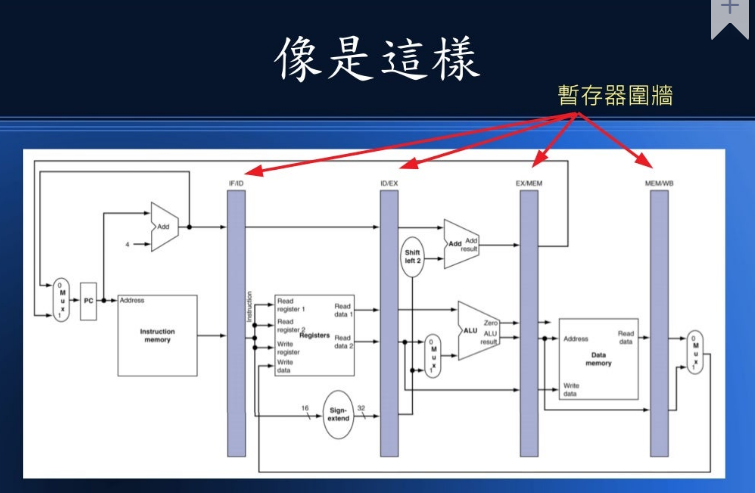
傳統

利用暫存器來讓CPU各區同時工作

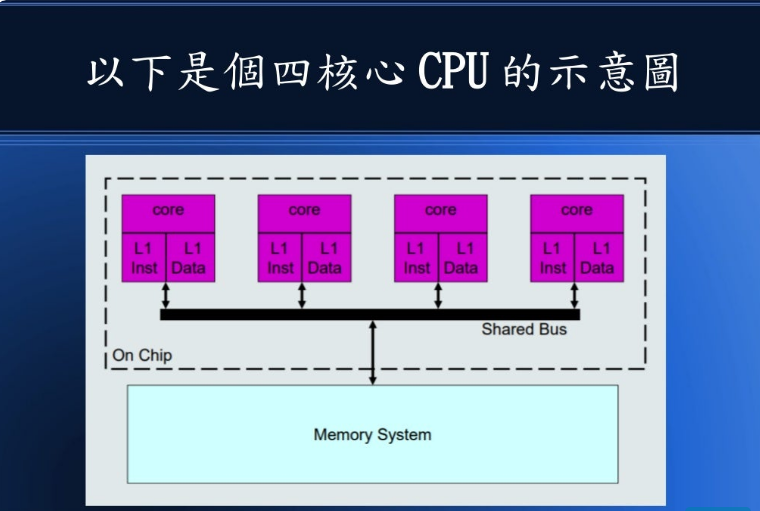
問題:

被跳躍指令打斷

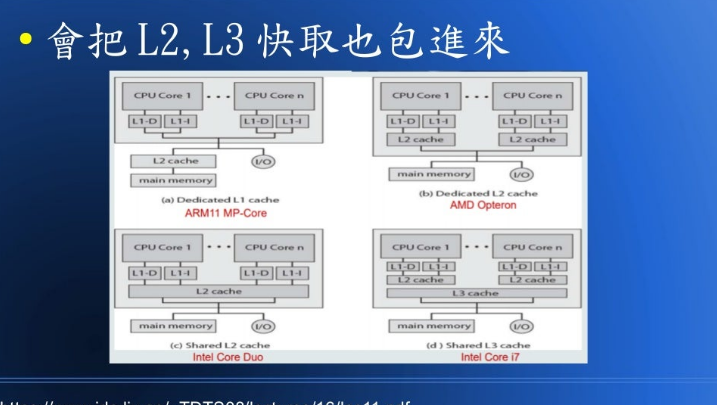
資料不在一級快取時，記憶體存取無法執行管線

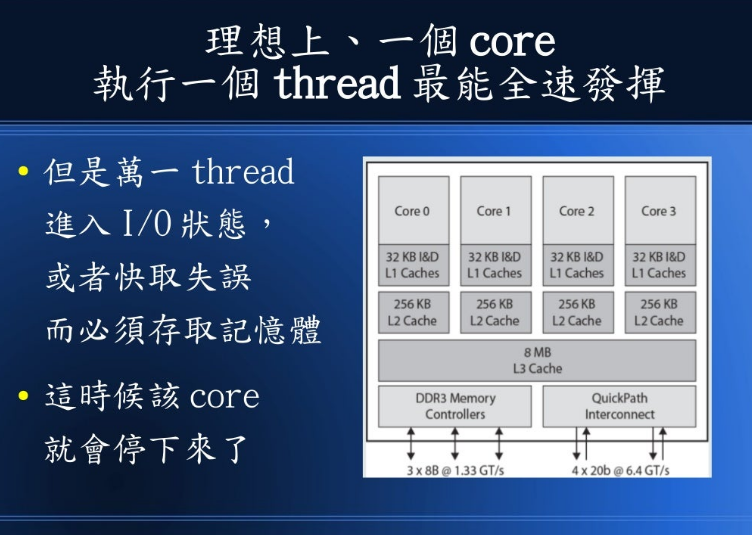


多核心+Hyper-Threading:

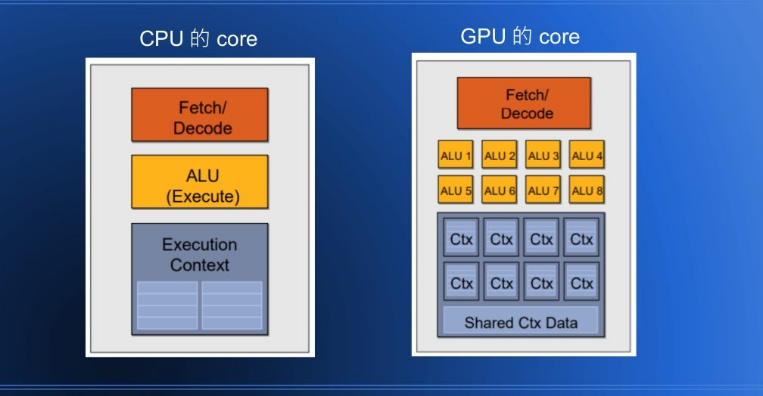


瓶頸:由於記憶體共用，同時間只有一個core能存取





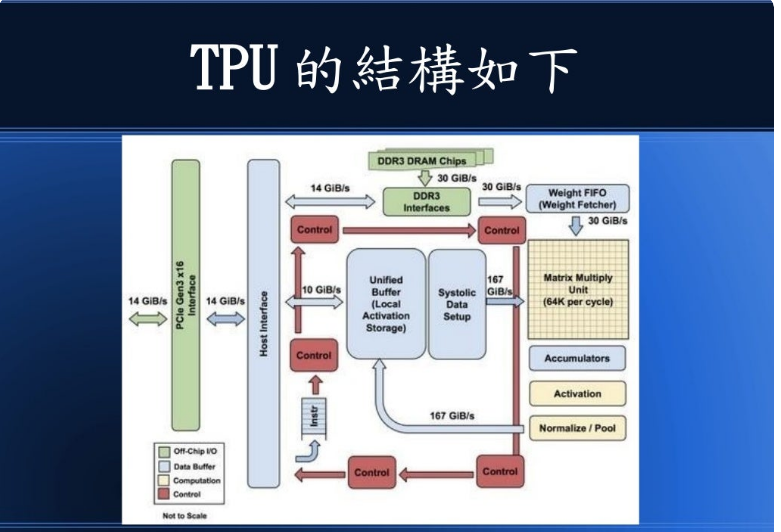
螢幕繪圖交給GPU



現在的電腦通常是CPU和GPU攜手合作

現代快速電腦舉例:

1. google深度學習TPU



TPU一次可以計算一整個矩陣相乘結果

1. 比特幣的挖掘機

使用SHA-256演算法快速計算

引用: https://www.slideshare.net/ccckmit/ss-85466673