CO project 6 – Multi-core

*ID: 0340249*

*Name: 孫聖*

* Introduction

這個實驗中我們做一個兩核CPU，並寫出計算矩陣乘法的組合語言和機器語言。

* Procedure

先將Instruction\_Memory和CPU各做兩份出來，並將其放入TESTBENCH中，同時還要放入Data\_Memory，連線時指明是哪一個CPU，即用cpu.和cpu2.來表示。一開始，在Data\_Memory中初始化Memory的值，執行的過程中兩個Instruction\_Memory分別讀取不同的機器碼，交給兩個CPU來讀取。

* Results

能夠計算相應的矩陣乘法。

* Questions

**1.How do you separate the program for the two cores?**

由於第二個矩陣有兩列，所以第一個CPU計算第一列，第二個CPU計算第二列。但他們都要讀取第一個矩陣中所有的值。第一個CPU：它先讀取矩陣2的第一列及矩陣1的第一行，之後進行乘法和加法的運算，並把結果儲存。之後再依次讀取矩陣1的第二行和第三行，同樣進行運算並儲存。第二個CPU：由於兩個CPU不能夠同時讀取Memory，所以一開始它執行nop。之後它读取矩陣2的第二列及矩陣1的每一行進行運算並儲存。

**2. Assumed that programmers do not know the platform architecture (i.e. single core or multi-core) how can programmers manage their program partition?**

如果程序員有充分的時間進行開發，可以使用concurrency programming的方法，如果機器只有單核，那麼可以利用多個thread來執行。如果機器有多核，既可以利用多個thread，也可以利用編譯器來為我們分配程式，使得在多核的環境下更有效率。

**3. Assumed that each core has private cache. If core1 write a new data at address 0x123, how could core 2 get the new data from 0x123? (hint: coherence)**

Snooping：cache監測已緩存位置的地址線，當有對該位置進行寫入的操作時，cahce控制器使自己的那份位置無效。

Directory-based: 所有被分享的資料都被放在同一個資料夾中，每個processor必須經過申請才能夠獲取分享的資料。當某一個資料被修改時，分享的資料可能會被更新，也有可能是被無效化。

* Architecture

