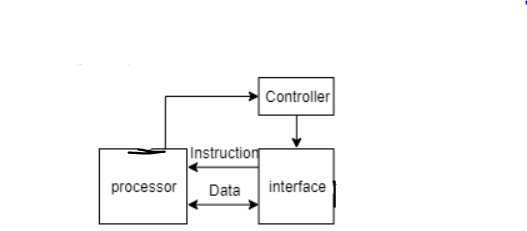
E94041107 鄧有敦

E94041220 郭濯瑀

E94046199 謝茹媛

電路方塊圖：



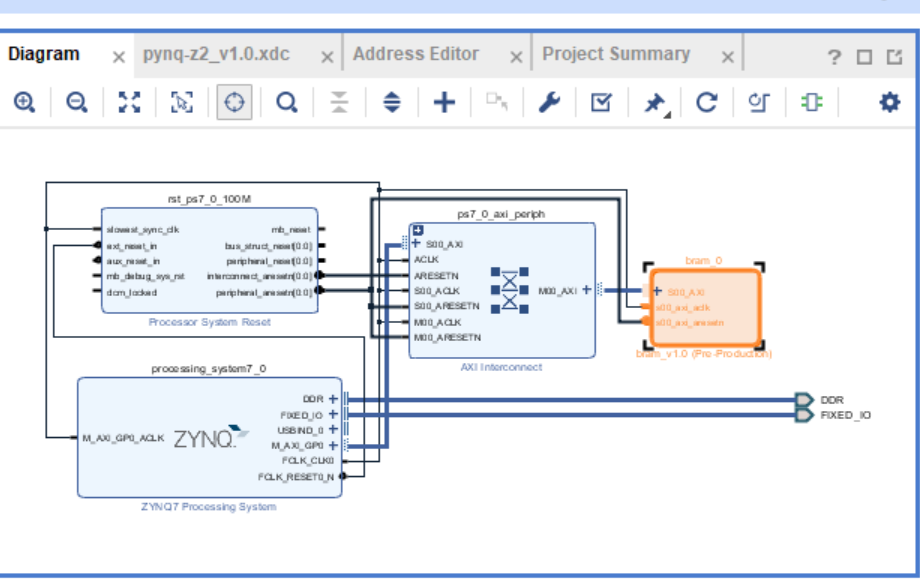
利用Lab6提供的ctrl.v和bram interface，我們自己設計了一個矩陣運算的processor，去做矩陣的各種運算，其中該運算單元並不會跟pynq的為處理器有接觸，所有的資料讀取都透過controller控制bram在兩邊的讀取行為，並且我們將所有的電路包成同一個ＩＰ（bram）。

IP簡介以及記憶體位址分配：

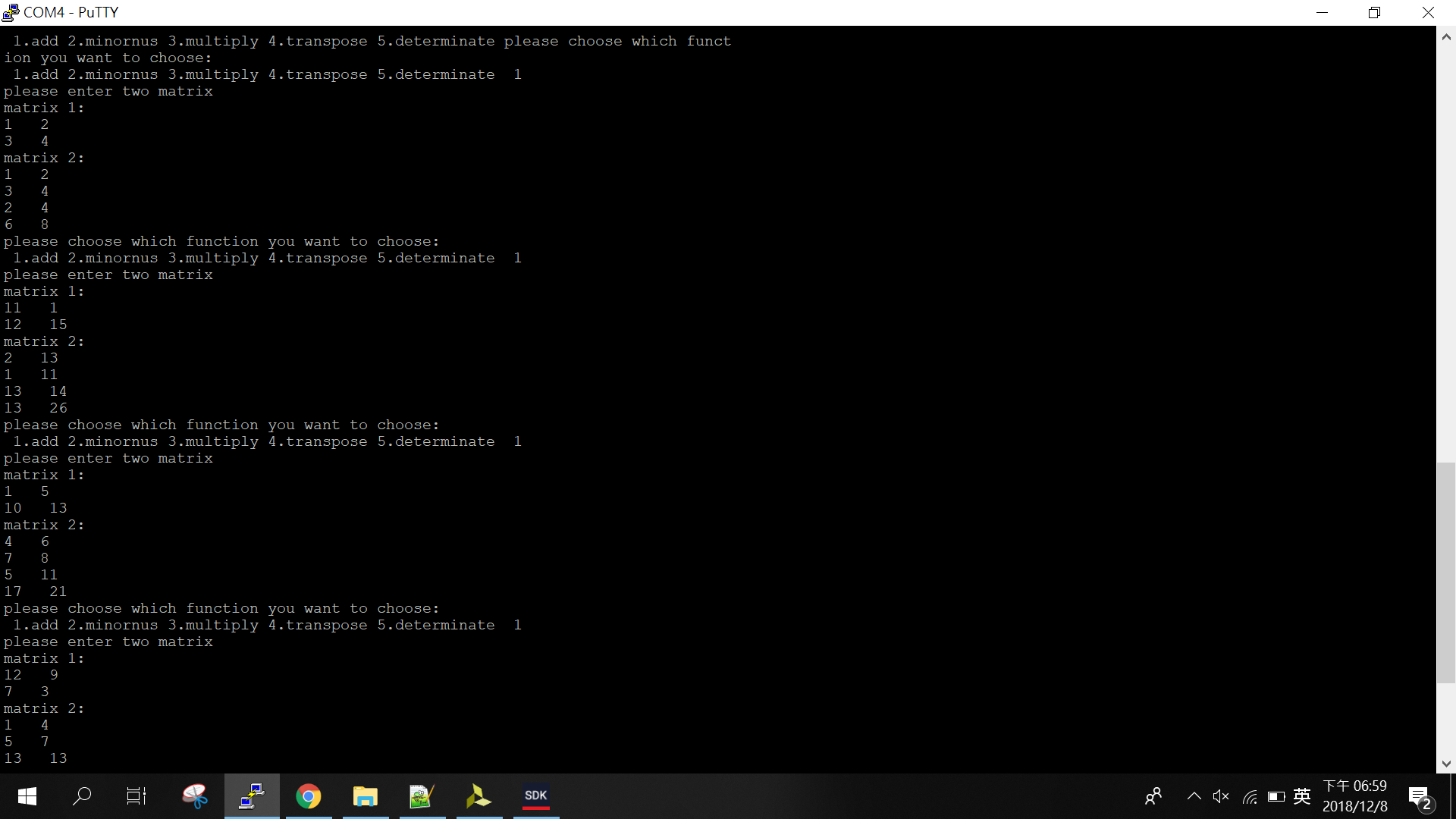
運算單元或是BRAM的設計都是時序電路，所以每次寫入或讀取的資料，都是一個2x2矩陣的元素，當讀取完所有資料，會傳入reset訊號使得電路被啟動並開始運算。

|  |  |
| --- | --- |
| 記憶體位址 | 資料 |
| 0x00000001 | 第一個矩陣第一個元素 |
| 0x00000002 | 第一個矩陣第二個元素 |
| 0x00000003 | 第一個矩陣第三個元素 |
| 0x00000004 | 第一個矩陣第四個元素 |
| 0x00000005 | 第二個矩陣第一個元素 |
| 0x00000006 | 第二個矩陣第二個元素 |
| 0x00000007 | 第二個矩陣第三個元素 |
| 0x00000008 | 第二個矩陣第四個元素 |
| 0x00000009 | 矩陣運算功能選取 |
| 0x0000000a | 回傳運算後矩陣 第一個元素 |
| 0x0000000b | 回傳運算後矩陣 第二個元素 |
| 0x0000000c | 回傳運算後矩陣 第三個元素 |
| 0x0000000d | 回傳運算後矩陣 第四個元素 |

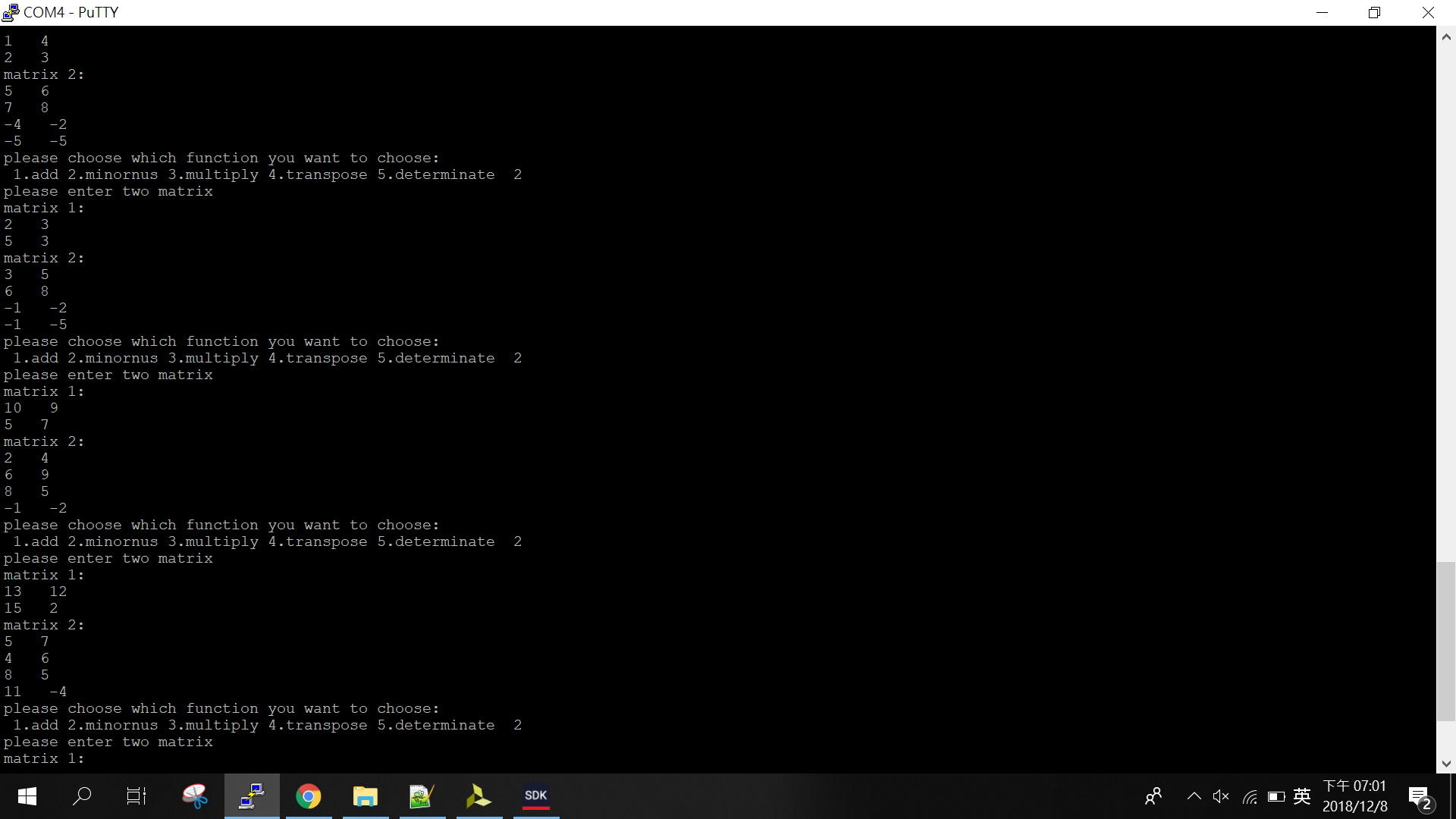
ＢＬＯＣＫ　ＤＥＳＩＧＮ：



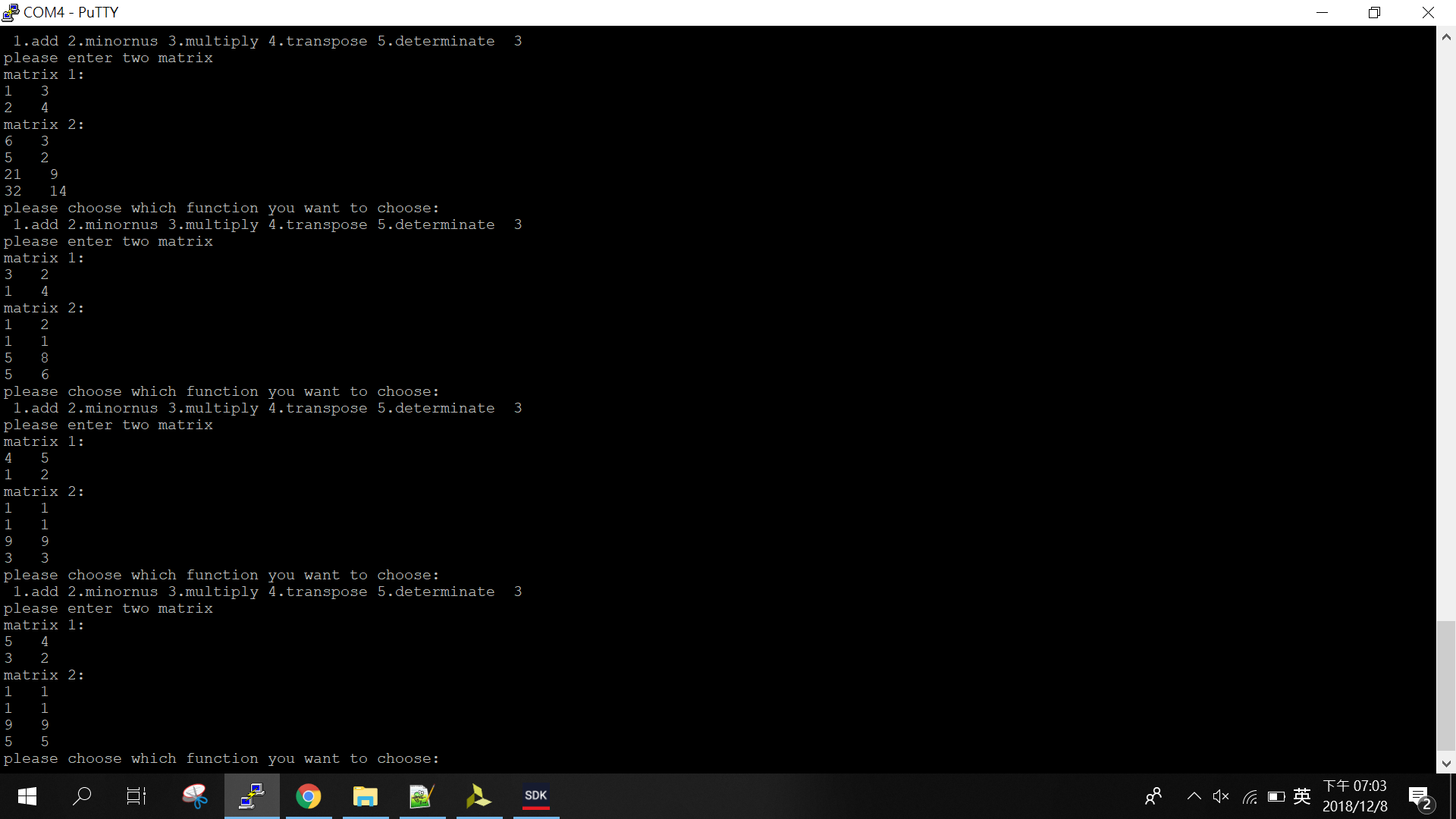
矩陣加法：



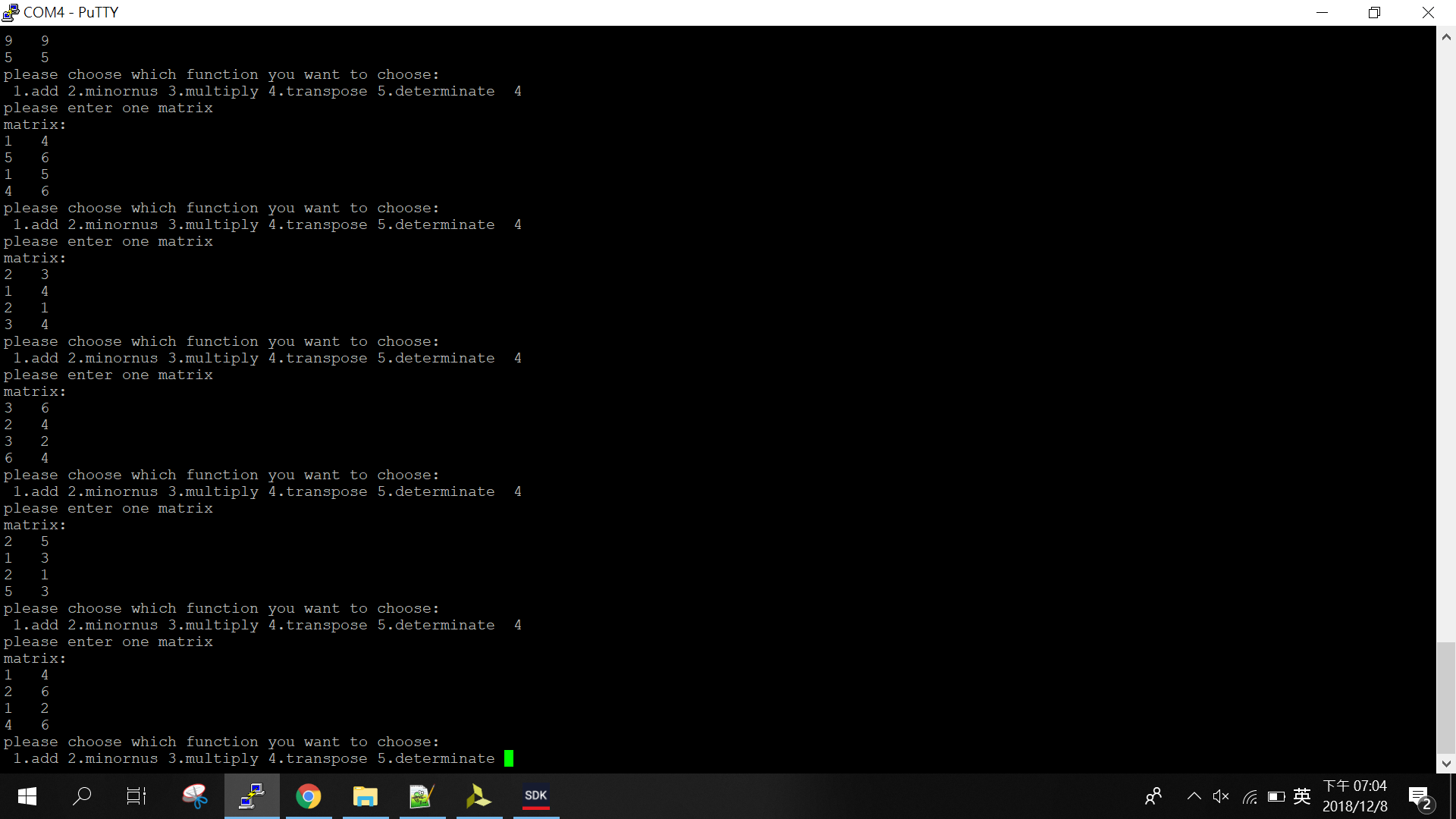
矩陣減法：



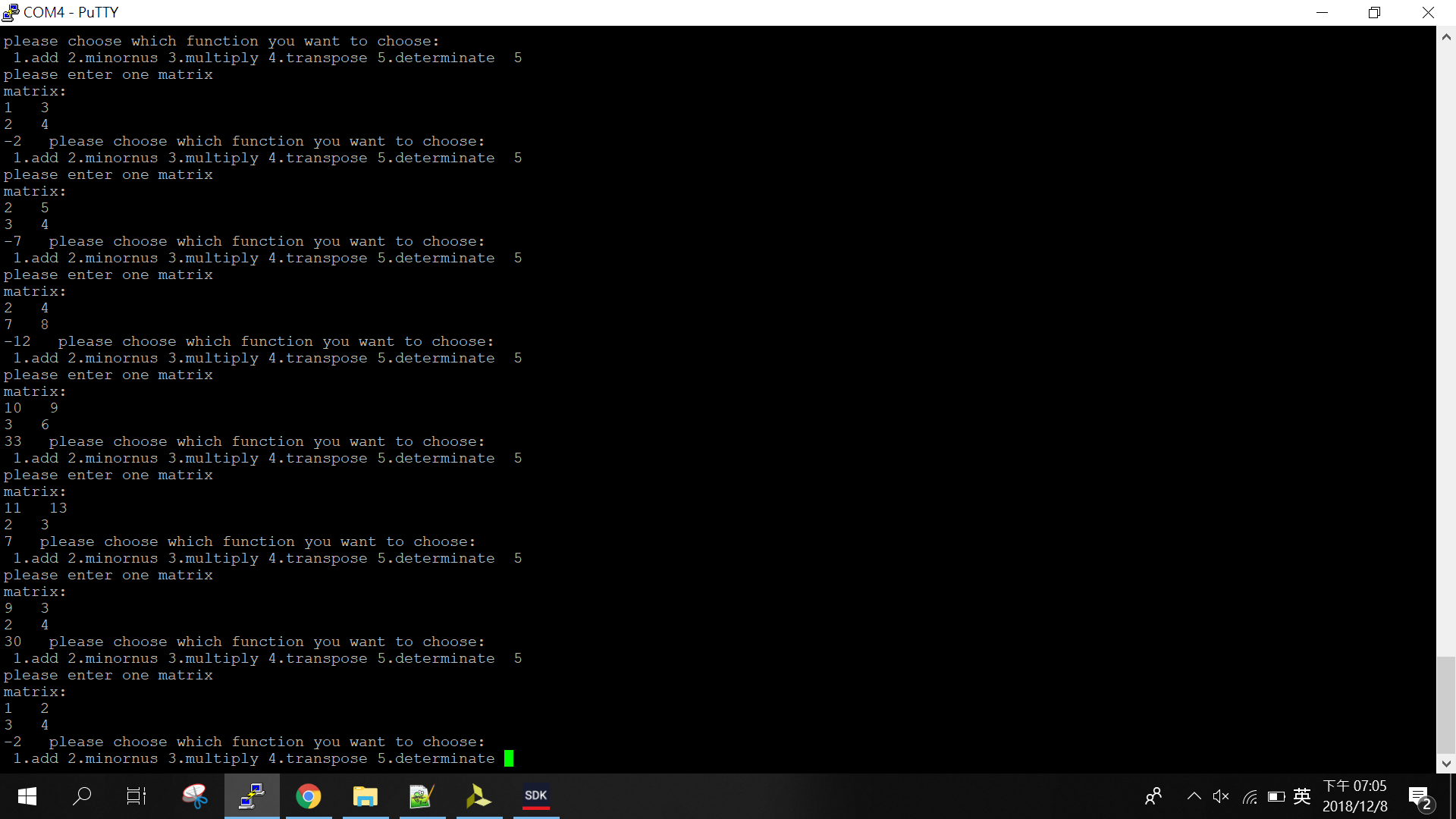
矩陣乘法：



矩陣轉置：



矩陣行列式：



問題與討論：

一開始運算單元跟controller的觸發時脈不同，導致產生資料讀取與運算的問題，後來發現後將作業完成的很順利。