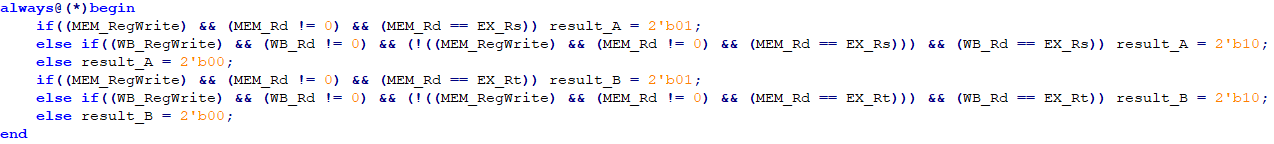
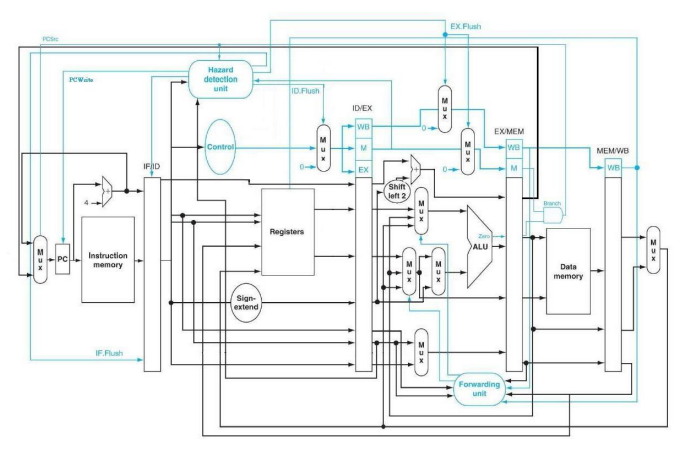
**Computer Organization Lab5**

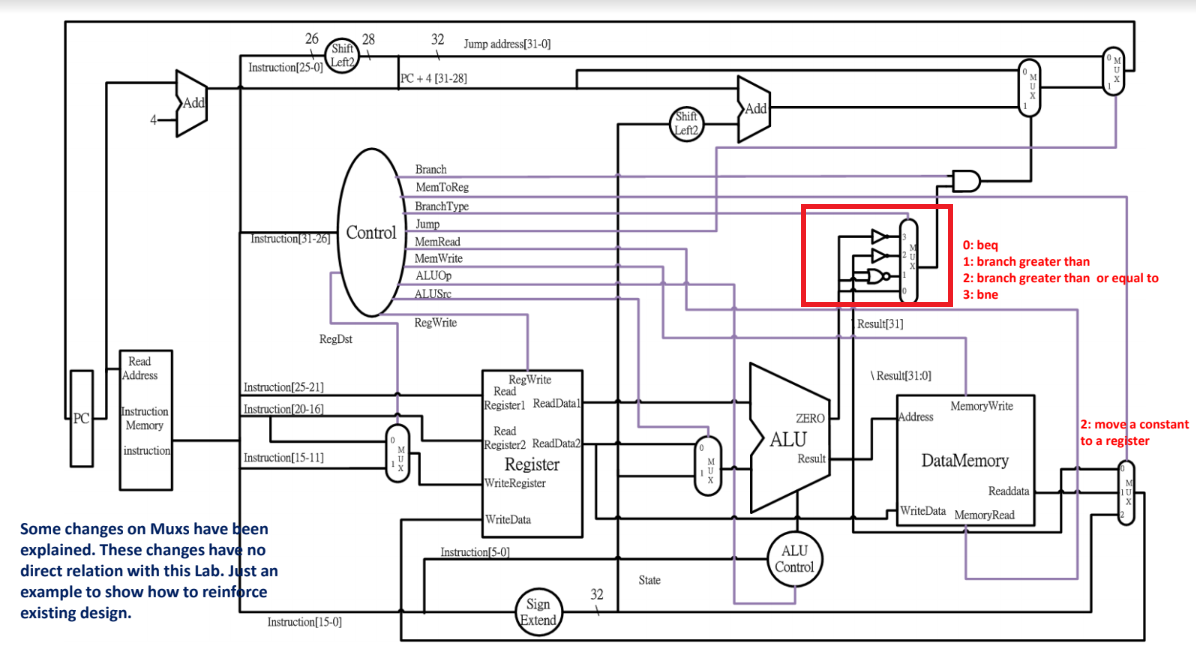
**Source code and the note:**

****

我認為這段在forwarding.v的程式碼是整個pipeline CPU的精髓，裡面粽重要的觀念是當有多個hazard發生時，要以最近的值來forward進行計算。例如：add $1,$1,$2 add $1,$1,$3 add $1,$1,$4，此時第三個add中的$1就要以第二個add中的$1運算結果為值，而並非第一個add中的$1。

**Architecture diagrams:**

****

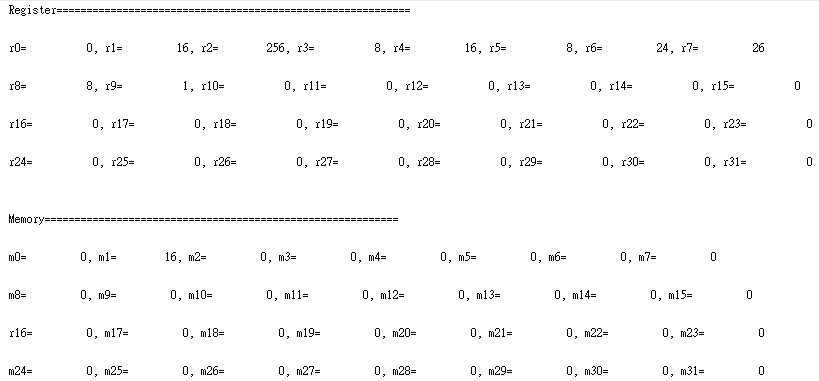
****

這次的電路圖設計是以上圖為基準，再在上圖的Branch之前加上下圖紅框內的部分以實作這次Advanced instruction的部分。

**Hardware module analysis:**

這次的架構與Lab4大致相同，多的部分只在forwarding以及hazard detection，forwarding使CPU能夠實現pipeline的功能，hazard detection則會在必要的時候stall instruction，使pipeline CPU能夠順利運作。

**Finished part:**

****