How do you implement this Pipelined CPU.

我們依照投影片最後一張圖，先分別將 modle算出來有多少，每個人認領部分。完成 modle後，再去分 Stage接線。

–

Explain the implementaHon of each module.

sign\_extend:

將輸入的16bits，用最後面的第16bit 重複放在 17-32的位置上。

mux: mux1, mux2, mux8, mux6, mux7, mux4, mux3, mux5

在 mux的部分，我依照三種不同的輸入輸出，將 mux寫成三種形式分別為兩個輸入(1,2,4,5,8)、三個輸入(6,7)、五個bits 的兩個輸入(3)。

forwarding unit

依照作業投影片P.7上的 pseudocode，改成 verilog。

EX/MEM & MEM/WB

幫每一個 output先設定預設值都為 0,之後將輸入接近輸出。

PC & Add\_PC & IF\_ID

依照投影片給的圖(p.12)寫成verilog。

Instruction\_Memory

利用前一次作業的內容。

Control

主要還是參考各個intruction的Truth Table依照Logic Design的方式決定intruction類型，再計算出要output的Control Signal

–

Problems and solution of this project.

PC能夠正常運作，但從 register中輸出的 read data1 & read data2 都會是 32‘h0。

在debug的時候發現設定各個reg跟wire的時候有些缺陷，只有考慮到1,0而忽略了x的可能，導致第一次跑的時候許多值都出不來，後來經過修改後才有所改善。