Lab4

1.

IF/ID

A picture containing text, whiteboard

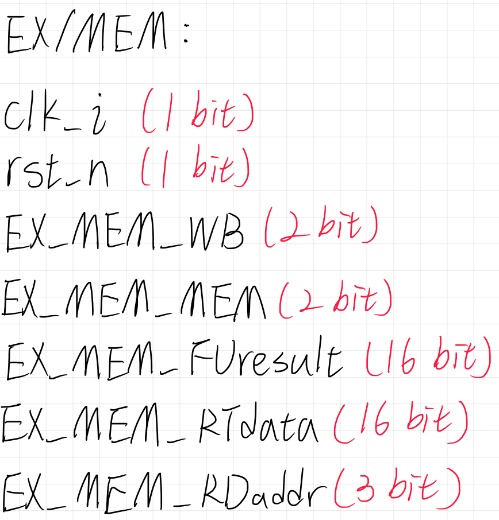
Description automatically generated

ID/EX

Text, letter

Description automatically generated

EX/MEM



MEM/WB

A picture containing text, whiteboard

Description automatically generated

2.

(a)

加入四個 pipeline registers ，重新安排線路，把RegDst移到

EX Stage，並把RDaddr傳遞到WB Stage。為了處理Data跟Control

Hazard 加入Data Hazard Detector與Control Hazard Detector。

(b)

為了用stall 處理Data Hazard，Data Hazard Detector 必須能暫停PC

與IF/ID，並清除ID/EX產生stall，因此加入

IF\_ID\_Hold , PC\_Hold, DATA\_ ID\_EX\_Flush 控制線，當偵測到Data Hazard 發生時，三條線路同時變成1，產生一個stall，避開data-dependency。PC\_Hold 會讓PC\_in從原本的加2變成加0，藉此維持不變，IF\_ID\_Hold 為1時IF/ID則不寫入，DATA\_ID\_EX\_Flush 為1時則全部寫入0。

(c)

為了用stall 處理Control Hazard，Control Hazard Detector 在EX Stage計算出Branch結果後，如果結果是跳，或是jump，則利用

IF\_ID\_Flush , Branch\_ID\_EX\_Flush 把IF/ID, ID/EX全部寫入0，把錯誤進入Pipeline的instruction洗掉。

3.

Graphical user interface

Description automatically generated

在執行中共產生 4個data hazard , 3個branch taken,

分別為 r2(instruction:1,3) , r3(instruction:2,3) , r5(instruction:3,4) , r4(instruction:5,6) , instruction 6 (beq) , instruction 7 (jump) , instruction 7(jump)。而透過detector偵測到hazard後，將datahazard訊號線 PC\_Hold , IF\_ID\_Hold , Data\_ID\_EX\_Flush 跟controlhazard訊號線IF\_ID\_Flush , Branch\_ID\_EX\_Flush變為1，產生stall。

4.

在將single cycle CPU 轉為 pipeline 的過程中，很常因為打錯線路名稱而產出錯誤訊號，為此偵錯了很久。還有改成forward 的過程中，必須把原本的線路再次安排，也是歷經一番波折。但是看到最後的結果是正確時，感到相當有成就感。透過這堂課才第一次學習到verilog，能完成各個Lab，感到相當開心，收穫頗多。

5.

新增了 Load\_Use\_Hazard , EX\_ForwardingUnit , MEM\_ForwardingUnit

刪除原本用stall 的Data Hazard Detector，但保持Control Hazard Detector 。為了EX forwarding加入mux3to1 : forward\_A跟forward\_B，為了MEM forwarding加入mux2to1 : mem\_forwarding。

而分別用Forward\_A , Forward\_B , MEM\_Forward控制線來控制。

6.

EX\_forward:

If ( EX\_MEM\_RegWrite &

(EX\_MEM\_RDaddr != 0) &

(ID\_EX\_RS\_reg == EX\_MEM\_RDaddr) ) ,

then { Forward\_A = 2 }

If ( ~( EX\_MEM\_RegWrite & (EX\_MEM\_RDaddr != 0) &

(ID\_EX\_RS\_reg == EX\_MEM\_RDaddr) ) &

MEM\_WB\_RegWrite &

(MEM\_WB\_RDaddr != 0) &

(ID\_EX\_RS\_reg == MEM\_WB\_RDaddr) ) ,

then{ Forward\_A = 1}

If ( EX\_MEM\_RegWrite &

(EX\_MEM\_RDaddr != 0) &

(ID\_EX\_RT\_reg == EX\_MEM\_RDaddr) ) ,

then { Forward\_B = 2 }

If ( ~( EX\_MEM\_RegWrite & (EX\_MEM\_RDaddr != 0) &

(ID\_EX\_RT\_reg == EX\_MEM\_RDaddr) ) &

MEM\_WB\_RegWrite &

(MEM\_WB\_RDaddr != 0) &

(ID\_EX\_RT\_reg == MEM\_WB\_RDaddr) ) ,

then{ Forward\_B =1 }

MEM\_forward:

If ( MEM\_WB\_MemtoReg &

EX\_MEM\_MemWrite &

( EX\_MEM\_RTaddr == MEM\_WB\_RDaddr)) ,

then{ MEM\_forward = 1 }

7.

A picture containing graphical user interface

Description automatically generated

結果正確，load\_use\_detetector 在clock cycle 7時，偵測到前一個instruction為lw，且r5有data dependency，故把三條訊號線(PC\_Hold , IF\_ID\_Flush , ID\_EX\_Flush)輸入1，使其產生一個stall。

8.

Graphical user interface

Description automatically generated

正確，在前面沒有被load\_use\_detector插入stall的情況下，MEM\_forwardongUnit偵測前一個指令為lw，而當前指令為sw，且r5有data dependency，故把訊號線MEM\_Forward輸入1，將lw讀出的data傳入MEM的WriteData中。