VLSI System Design (Graduate Level)

Fall 2023

HOMEWORK III

REPORT

Must do self-checking before submission:

Compress all files described in the problem into one tar

All SystemVerilog files can be compiled under SoC Lab environment

All port declarations comply with I/O port specifications

Organize files according to File Hierarchy Requirement

No any waveform files in deliverables

Student name: \_蔡明翰\_ \_曾柏硯\_

Student ID: \_N26120579\_ \_M16121027\_

**Summary**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Hardware | | | | | | | | |
|  | | | | | | RTL | | synthesis |
| Top | CPU\_wrapper | | CPU | | | V | | V |
| New instructions | | | V | | V |
| SRAM\_wrapper (IM & DM) | | | | | V | | V |
| ROM\_wrapper | | | | | V | | V |
| DRAM\_wrapper | | | | | V | | V |
| AXI | | | | | V | | V |
| Sensor control wrapper | | | | | V | | V |
| Watch Dog Timer | | | | | V | | V |
| Synthesis result | | | | | | | | |
| Area(um2) | | | | | Clock cycle(ns) | | | |
| 3143077 | | | | | 10 | | | |
| Firmware & Software | | | | | | | | |
|  | | RTL pass | | syn pass | | | Execution time(ns) | |
| Booting | | V | | V | | | - | |
| Prog 0 | | V | | V | | | 1496030 | |
| Prog 1 | | V | | V | | | 7712020 | |
| Prog 2 | | V | | V | | | 33414420 | |
| Prog 3 | | V | | V | | | 15735950 | |
| Prog 4 | | V | | V | | | 9110130 | |
| Prog 5 | | V | | V | | | 9112200 | |
| Spyglass summary(number of inline messages) | | | | | | | | |
| Information | | Warning | | Error | | | Fatal | |
| 33 | | 1 | | 0 | | | 0 | |
| Superlint(number of inline messages) | | | | | | | | |
| Total lines | | Warning | | Error | | | coverage(%) | |
| 3965 | | 19 | | 0 | | | 99.52 | |

**Contribution**

|  |  |
| --- | --- |
| 蔡明翰 50% | 曾柏硯 50% |
| CDC電路設計、CSR指令說明、WDT、Superlint、Waveform、sensor\_ctrl wrapper | AXI(VIP)、DRAM\_wrapper、Spyglass、Firmware & Softwar解釋、ROM\_wrapper |

目錄

1. **Hardware Design Description**
2. **Software & Firmware design description**

**三、 Screen shot of wave forms & simulation results**

四、 **Synthesize**

五、 **JasperGold**

**六、 Superlint**

**七、 Spyglass**

**八、 Problems to answer**

**九、 Lesson Learn**

1. **Hardware Design Description**

* System Block Diagram

一張含有 文字, 圖表, 行, 平行 的圖片

自動產生的描述

新增Sensor Control、ROM、DRAM、WDT和對應wrapper，並擴充AXI。

1. **AXI**: 擴充AXI為M0~M1、S0~S5，並根據CPU讀寫的地址，透過Mem mapping決定儲存的位址，同時重新設計arbitrator排程每個Memory。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

2. **Sensor Control**: Sensor每1024 cycles會產生一筆新的資料，並儲存至sensor control的local memory，當記憶體滿時，sensor control會使sensor\_en=0停止request data，同時向CPU發出interrupt訊號，CPU會暫停目前執行指令並儲存暫停時的PC位址。之後將sensor control的資料搬至DRAM，結束後返回暫停時之PC位址。

另外CPU會分別透過寫入0x1000\_0100和0x1000\_0200去拉起stcrl\_en(使sensor資料能存入sensor control)和sctrl\_clear訊號(清空sensor control)。

3. **ROM wrapper**: CPU只能讀取ROM的資料，之後再寫入SRAM或DRAM，故只需要上次所設計之sram wrapper讀取的部分。

4. **SRAM wrapper**: 與上次作業相同。

5. **CPU**: 新增CSR指令和CSR暫存器，前者需要修改ID stage的decoder架構；後者我們是將CSR regfile放置在EXE stage，這樣的設計可以避免WFI、MRET指令和jumpbranch同時處理的狀況。另外csr指令的結果僅跟register有關，所以以csr為目標暫存器的結果會不經過registers而直接存入csr reg file，不需要forwarding，以達到簡化設計。

6. **WDT**: 內部會設置一timer不斷計數，計數至一定數值會發出timer interrupt給CPU，使CPU中斷執行，並跳至設定之PC位址。CPU會透過寫入0x1001\_0200以拉起WDLIVE訊號，以清空該timer。但若CPU因為陷入迴圈、執行錯誤或執行某指令等等花費太多cycle時，就會無法及時發出WDLIVE訊號，WDT就會發出timer interrupt訊號。但WDT的clock domain與CPU之clock domain並不相同，直接傳送資料可能會有setup time和hold time的問題，造成資料metastable，所以需要透過CDC設計將另一個clock domain的計數資料能成功對到CPU的clk domain。

* Interrupt mechanism description and flow chart

(1)Sensor interrupt: 當sensor interrupt拉起時，會檢查interrupt enable訊號MIE和MEIE是否拉起，若已進入WFI則僅需檢查MEIE，若拉起則進入trap，將mstatus: MIE<=0, MPIE<=MIE, MPP<=2’b11，拉起MEIP表示interrupt掛起，將PC存入mepc(若同時發生WFI則存入PC+4)，CPU之pc跳至{mtvec[31:2],2’b0}，此次作業僅direct mode為0x0001\_0000。

進入trap後，直到MRET發生，CPU才會返回interrupt

發生之pc位址即mepc所存之值，並使mstatus: MIE<=MPIE, MPIE<=1’b1, MPP<=2’b11; mip: MEIP<=0, MTIP<=0; mie: MEIE<=1, MTIE<=1，即清空interrupt pending拉起interrupt enable號。

(2)Timer interrupt: 當timer interrupt拉起時，檢查MIE和MTIE是否拉起，若已進入WFI則檢查MTIE即可，CPU之pc跳至指定pc位址，拉起timer interrupt pending訊號MTIP，後續大致動作同上。

* WDT & CDC circuit description and diagram

Watch dog timer和CDC的設計，這次CDC要傳輸的訊號是慢到快且是週期性的counter訊號，所以我們使用的方法是將clk2對齊至clk1後，在clk1 domain進行計數，可以避免多bits訊號傳輸的問題，比起gray code所需要的電路面積更小。

首先產生一週期為clk2兩倍之D0訊號，接著將D0訊號接著在clk1正緣觸發產生D1(對其clk1)，D1再延遲1 clk1產生D2(對齊clk1)，D2再延遲1個clk產生D3(對齊clk1)，接著將D2和D3進行xor就可以產生一個對齊clk1且頻率跟clk2相同的clk3訊號。透過clk3訊號就可以在clk1 domain進行clk2頻率的計數，完成CDC設計，clk有dealy下，功能仍正常運作。一張含有 螢幕擷取畫面, 多媒體軟體, 電腦 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 多媒體軟體, 鮮豔, 行 的圖片

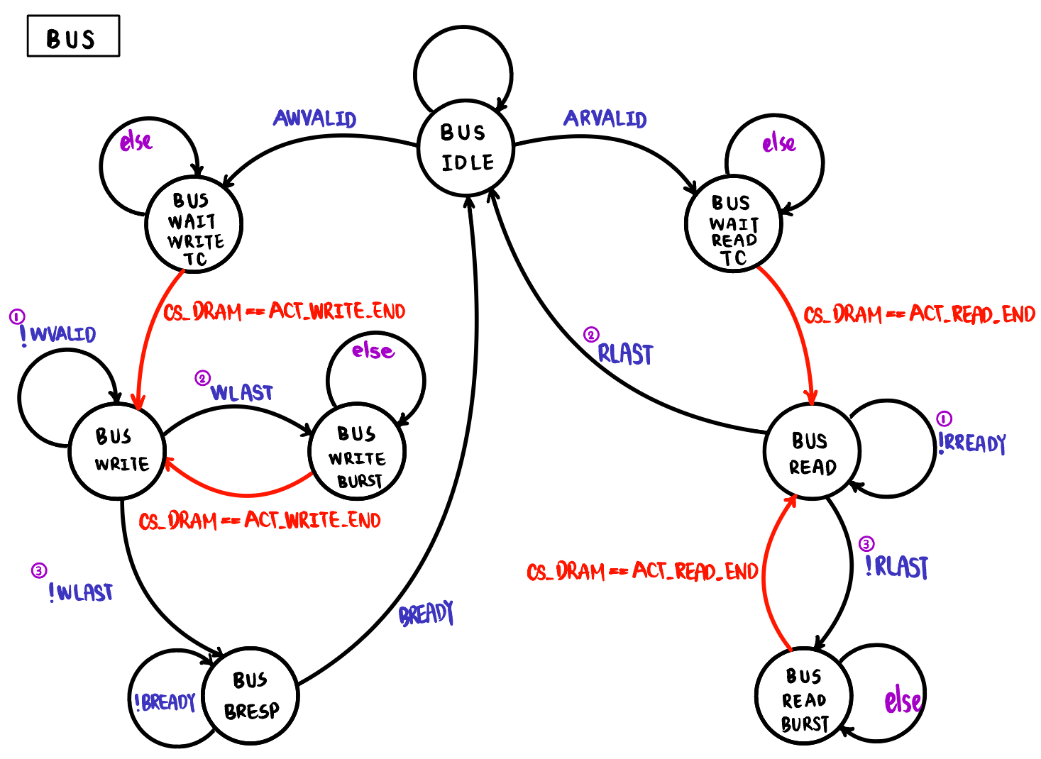
自動產生的描述

一張含有 文字, 圓形, 圖表, 螢幕擷取畫面 的圖片

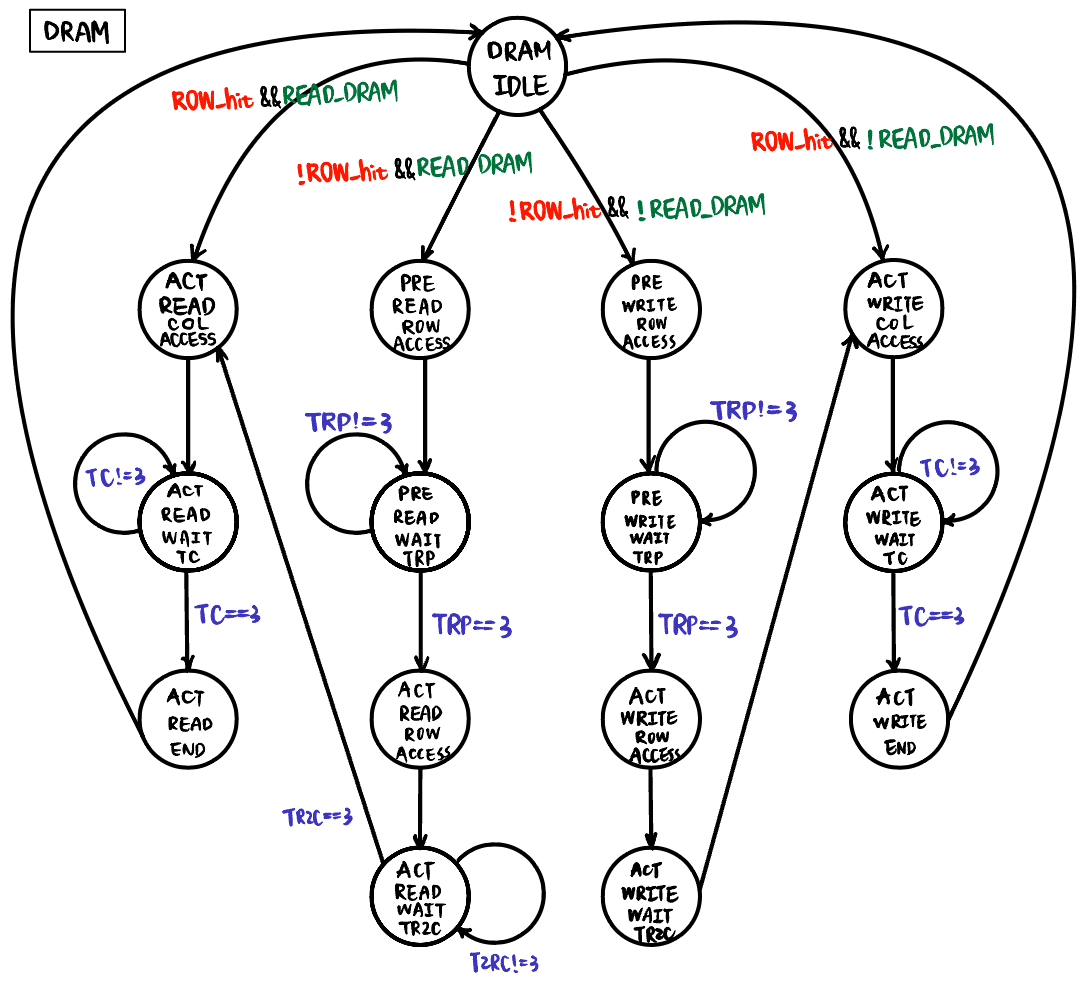
自動產生的描述

7. **DRAM\_wrapper**

在DRAM\_wrapper中設計兩個Finite state machine以方便進行整體控制。



首先是BUS State machine，reset後進入BUS IDLE，並且等待ARVALID/AWVALID，分別進入相應的READ/WRITE WAIT\_TC狀態，TC代表的是DRAM中column access所需要的延遲時間，等到對應DRAM狀態已經達到column access time後即可透過BUS傳輸資料，並且當Burst length不為1時，會進入READ/WRITE burst狀態，等待DRAM READ/WRITE END結束反覆回到BUS READ/WRITE狀態確認LAST訊號，再分別回到IDLE或者是BRESP🡪IDLE。



DRAM access的方式為先讀取row再讀取column，如果Row miss則需要先對原本的row進行precharge，再讀取新的row；若Row hit則可以不用

進行precharge，並跳過讀取row的時間，直接對col進行讀取。reset後進入DRAM IDLE狀態並判斷是否為Row hit以及是進行READ/WRITE操作判斷進入哪一個狀態，若為Row miss則進入Precharge row，等待TRP(DRAM precharge delay)後開始Activate新的row，等待TR2C(DRAM row to column delay)完成後進入column access，等待TC(DRAM column delay)後完成資料的讀取或寫入，回到DRAM IDLE。

**二、 Software & Firmware**

1. Prog1

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

【說明】**:**宣告兩個外部變數array\_size array\_addr表示這些變數在其他地方定義，const short\* array = &array\_addr建立一個指標array，指向 array\_addr 的位置，並且這裡的array是指向常數的指標，所指向的內容**不可被修改**。

extern short \_test\_start宣告一個\_test\_start 的外部變數。

short\* dest = &\_test\_start建立一個指標 dest，指向 \_test\_start 的位置。

在 insertionSort() 函式中：

使用插入排序演算法對 dest 指向的陣列進行排序。

透過array\_size控制排序的範圍，使用 temp 暫存要插入的元素。

透過雙重迴圈，將元素依序從後往前比較，將比較大的元素往右移，直到找到合適的位置插入 temp。

2. Prog2

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

【說明】**:** 宣告3個外部變數(extern)

透過讀取 \_binary\_image\_bmp\_start 的特定位置(偏移了2、3、4、5個位置)來計算圖像大小 binary\_image\_bmp\_size。這裡使用了四個uint8\_t變數來組合成一個 unsigned int，從圖像的第54個位元組開始進行處理(BMP圖像的像素資料)，迴圈將RGB色彩分量提取出來，並且依據指定的權重計算出灰度值 gray，將計算得到的灰度值寫回 \_test\_start 陣列，並且在輸出時將原本的 RGB 資料替換為灰度值，使用加權平均來計算灰度值，公式為：gray = 0.11 \* blue + 0.59 \* green + 0.3 \* red，之後將得到的灰度值寫回 \_test\_start 陣列中對應的位置，同時將原本的三個色彩通道替換為計算得到的灰度值

3. **boot**

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

【說明】**:** d1、d2 和 d3 是計算不同區段的資料大小。

接著使用三個迴圈根據計算出的大小(d)，從DRAM中的起始位址開始，將資料逐一複製到IM或DM中。

1. **Screen shot of waveform & simulation result**
2. **Pre-sim** (clk period: 10ns)

Prog0:299196ns

一張含有 文字, 螢幕擷取畫面, 藝術, 設計 的圖片

自動產生的描述

Prog1:1542394ns

一張含有 文字, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

Prog2:3341438ns

**一張含有 文字, 螢幕擷取畫面, 摩天大樓, 藝術 的圖片

自動產生的描述**

Prog3:3147180ns

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Prog4:1722008ns

一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述

Prog5:1722218ns

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

1. **Post-sim** (clk period: 10ns)

Prog0: 1496030ns

一張含有 螢幕擷取畫面, 文字, 摩天大樓 的圖片

自動產生的描述

Prog1: 7712020ns

一張含有 文字, 螢幕擷取畫面, 陳列 的圖片

自動產生的描述

Prog2: ns

一張含有 文字, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

Prog3: 15735950ns

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

Prog4: 9110130ns

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Prog5: 9112200ns

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

1. **Waveform**
2. Waveforms

(1)CSRRW

一張含有 螢幕擷取畫面, 行 的圖片

自動產生的描述

Pc 10a14: csrrw mstatus t0, csr\_rd=0x300, csr\_alu\_out=t0=f\_0008, 因為mstatus只能寫入MIE、MEIE、MPIE, mstatus<=8。

(2)CSRRWI

一張含有 螢幕擷取畫面, 行, space 的圖片

自動產生的描述

Pc 10aac: csrrwi mstatus 9, csr\_alu\_out=uimm=9, mstatus<=8。

(3)CSRRS

一張含有 螢幕擷取畫面, 行 的圖片

自動產生的描述

Pc 10b00: csrrs t0 mstatus t1, csr\_alu\_out=csr|rs1=f\_008, mstatus<=8。

(4)CSRRSI

一張含有 螢幕擷取畫面, 行, 多媒體軟體 的圖片

自動產生的描述

Pc 10b1c: csrrsi t0 mstatus 9, csr\_alu\_out=csr|uimm=0|9=9, mstatus<=8。

(5)CSRRC

一張含有 螢幕擷取畫面, 行, 多媒體軟體, space 的圖片

自動產生的描述

Pc 10b40: csrrc t0 mstatus t1, csr\_alu\_out=csr&~rs1=1888&~f\_1880=8, mstatus<=8

(6)CSRRCI

一張含有 螢幕擷取畫面, 行 的圖片

自動產生的描述

Pc 10b5c:csrrcit0mstatus 9,csr\_alu\_out=csr&~uimm=1888&~9=1880,mstatus<=1880

(7)MRET

一張含有 螢幕擷取畫面, 鮮豔 的圖片

自動產生的描述

pc 10104: MRET，MRET訊號拉起，返回mepc儲存pc位址，currentpc<=mepc, mstatus<=1888, mie<=800, mip<=0。

(8)WFI

一張含有 螢幕擷取畫面, 行, 鮮豔, 電路 的圖片

自動產生的描述

Pc 1045c: WFI訊號拉起，pc位址設為0，並拉起WFI\_reg持續使EXE級PC歸0。直到發生interrupt時(圖中為sensor interrupt)，WFI\_reg歸0，PC跳轉至mtvec儲存之pc 1\_0000，mstatus<=1800, mie<=0,mip<=800，表示interrupt掛起，正在處理中斷。

(9)sensor

一張含有 螢幕擷取畫面, 行, 鮮豔 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 文字, 行, 軟體 的圖片

自動產生的描述

CPU 寫入地址 0x1000\_0100，使sensor\_en拉起，開始接收sensor資料，並透過counter計數。

一張含有 螢幕擷取畫面, 鮮豔, 行 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 多媒體軟體, 繪圖軟體 的圖片

自動產生的描述

寫滿之後，發出sctrl interrupt，並使sensor\_en為0，停止接收sensor訊號。直到CPU寫入地址1000\_0200，拉起sctrl\_clear訊號，結束sctrl interrupt並重新拉起sensor\_en，開始接收sensor訊號。

(10)WDT

一張含有 螢幕擷取畫面, 文字, 行, 平行 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 多媒體軟體, 鮮豔, 行 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 多媒體軟體, 鮮豔, 行 的圖片

自動產生的描述

CPU寫入地址0x1001\_0100，拉起WDEN，開始計數，透過產生週期跟clk2相同且和clk1對齊之CL2作為timer觸發訊號。且CDC設計要能在clk有delay的情況下，仍能正常運作。

一張含有 螢幕擷取畫面, 行, 鮮豔 的圖片

自動產生的描述

CPU透過寫入地址0x1001\_0200，拉起WDLIVE訊號，使WDT counter清空，避免產生WDT interrupt。

一張含有 螢幕擷取畫面, 行, 鮮豔 的圖片

自動產生的描述

若因為CPU當機或執行指令花費太多cycles，在WDT timer計數至2710後，拉起WDT interrupt訊號，並清空WDT timer重新計數。

1. **Synthesize**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock Period(ns)** | **Area(um2)** | **Power(mw)** | **Data Require time(ns)** |
| **10.0** | **6143077** | **136.6560** | **10.79/100.81** |

1. **Timing**

(1) clk domain I: 10.79ns

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

(2) clk domain II : 100.81ns

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

1. **Area: 6143077um2**

**一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

1. **Power: 136.6560 mW**

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

**五、 Jaspergold**

vip\_b: all pass

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

**六、 Supelint**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

Total lines: **3965**

Code Coverage = 100%

**七、 Spyglass**

一張含有 文字, 電子產品, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

**八、 Problem to answer**

1. **What is the deference between mcycle and timer? When is mcycle used?**

mcycle: 計算運行的cycle數，通常在CPU內部；timer: 同樣會進行計數，但可能在CPU以外的module或外部設備，所以clock domain也許和CPU不同，功能是測量或控制時間訊號，透過timer可以安排事件或實現延遲。

若需要精確的執行時間相關操作時，就可以讀取在CPU內部的mcycle暫存器數值。或是要對CPU內部進行執行時間量測、性能分析並對其進行優化時，mcycle的數值就是很好的指標。

1. **What is “Potential Qualifier” in Spyglass?**

Potential Qualifier是一個指示符號，通常用來表示潛在的問題或警告，表示電路設計中的某些部分可能存在問題，但不代表真的有錯誤，需要設計者進一步的檢查。

(1)時序問題：當設計中的某些元件或訊號可能會在特定條件下出現時序問題時，Spyglass可能會標記為Potential Qualifier。這可能是因為信號傳輸時間過長、時脈偏移或者時序constraint設定不好。

(2)組合邏輯問題：如果設計中存在組合邏輯電路，在某些情況下可能產生未預期的行為，Spyglass可能會標記這些部分為Potential Qualifier。可能是因為不完整的P&R、訊號延遲或電路的設計錯誤。

(3)可能的功耗優化：當Spyglass檢測到設計中存在可以改進功耗的可能性時，它會將這些部分標記為Potential Qualifier，可能包括冗餘邏輯、功耗過高的電路或者可以進行電源管理的部分。

(4)可靠性問題：當設計中的某些部分可能會導致硬件故障或可靠性問題時，Spyglass可能會標記為Potential Qualifier，包括時脈和重要電路之間的冗餘。

**九、 Lesson Learn**

**蔡明翰**: 這次的作業一次增加ROM、DRAM、sensor和WDT的module，一開始原本以外wrapper使可以使用之前的sram wrapper，解果除了ROM以外，幾乎都需要重新思考架構和狀態機，而DRAM wrapper也並不是原本所想只需要增加的row hit和column hit那麼簡單。WDT的設計是之前上過很多次相關課程的CDC設計，原本是打算使用2 flip-flop的設計，但因為是多bits的傳輸，設計會變很複雜。後來的解決方法是將clk2 domain的訊號，保持同樣頻率的狀況下，對其clk1 domain。結果花最多時間的問題是在CPU wrapper的部分，因為這次的執行指令次數非常多，竟然驗到之前沒測到的錯誤條件，挖了很多時間確認其他元件都正確後，仔細對照每個訊號來源，才找到錯誤訊號的原因，算是強化了debug的能力吧。這次作業學習了CDC設計和考慮更多MEM訊號要如何排程以及DRAM如何運作，更重要的是養成了更好的除錯能力。

**曾柏硯**: 這次的作業加入了DRAM以及CDC電路等設計，除了需要新增AXI BUS裡面Slave之外，還需要注意每一個IP定義的記憶體位置做一些相應的判斷調整，而我這次最主要的工作是負責設計DRAM wrapper，透過這次的作業我學習到DRAM運作的過程，DRAM每一次access都需要先判斷是否為Row hit，若與前一次使用的Row相同則可以不用進行Precharge，若為Row miss則需要經過Precharge再Activate，也透過這次作業了解到DRAM真的為餘記憶體階層比較下層的位置，所以每次要到DRAM搬運資料都要花費比較多個時脈週期，控制的邏輯與狀態的設計也會比on chip memory和SRAM還要再複雜一些。