VLSI System Design (Graduate Level)

Fall 2022

HOMEWORK IV

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 waveform files in deliverables

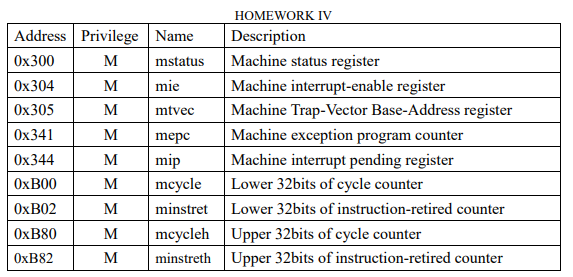
Student name: \_\_黃冠予\_\_\_\_ \_\_\_俞杉麒\_\_\_\_

Student ID: \_\_M16111064\_\_\_ \_\_\_P78111519\_\_\_\_

1. **Summary**

在這次作業中，完成了以下事項:

* 1. 在5-stage pipeline CPU中增加下列CSRs(Control & State Registers)並且實現中斷功能(Interrupt)



* 1. 新增下列Instructions(只需要實現Machine Mode)

一張含有 文字, 櫃子, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 桌 的圖片

自動產生的描述

* 1. 新增SENSOR\_controller\_wrapper 使SENSOR\_controller和AXI Bus 相接
  2. 新增WDT(Watch Dog Timer)和WDT\_wrapper，並解決CDC 問題。
  3. 修改AXI Bus Slave configuration

一張含有 桌 的圖片

自動產生的描述

d. 完成APR(Automatic Place and Route)

e.

1. Performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prog0** | **Prog1** | **Prog2** | **Prog3** | **Prog4** | **Area** |
| 1138212500 ps | 15097437500  ps | 1068562500 ps | 7440062500 ps | 7441325 ns | 6639610.16 |

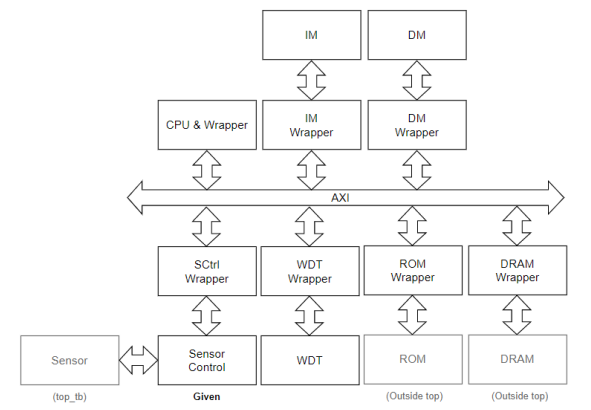
1. Simulation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RTL0 | RTL1 | RTL2 | RTL3 | RTL4 |
| pass | pass | pass | pass | pass |
| Syn0 | Syn1 | Syn2 | Syn3 | Syn4 |
| pass | pass | pass | pass | pass |
| PR0 | PR1 | PR2 | PR3 | PR4 |
| pass | pass | pass | pass | pass |

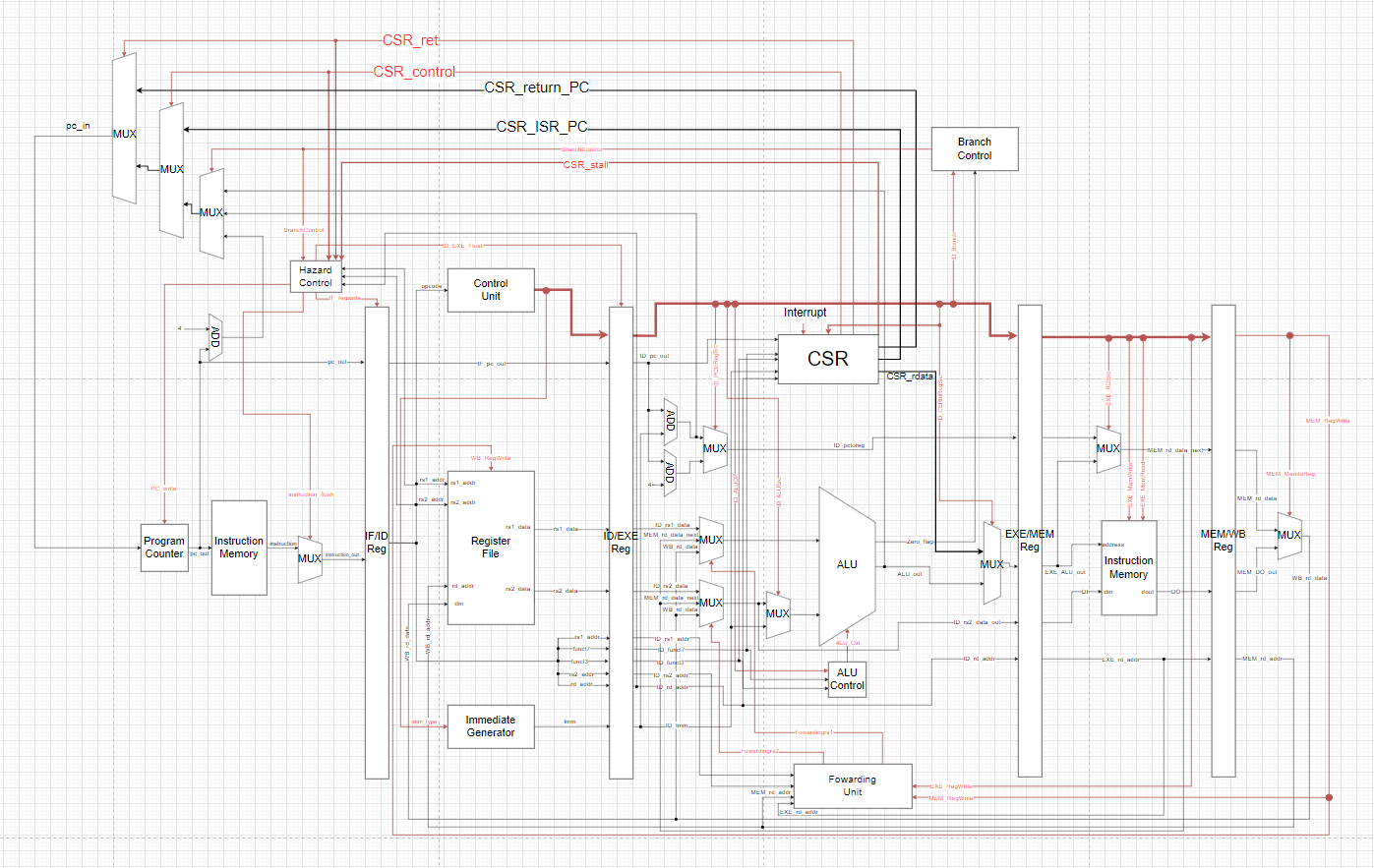
f.貢獻度

|  |  |
| --- | --- |
| 黃冠予 | 俞杉麒 |
| M16111064 | P78111519 |
| 50% | 50% |

1. **Block diagram**
   1. System Block overview



* 1. CPU Block overview



* + 1. CSR module 中包含了

mstatus,mie,mtvec,mepc,mip,mcycle,minstret,mcycleh,minstreth等Register。

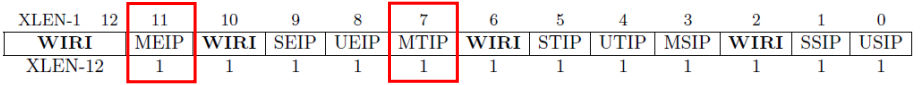
* + - 1. mstatus(Machine status register)

一張含有 桌 的圖片

自動產生的描述

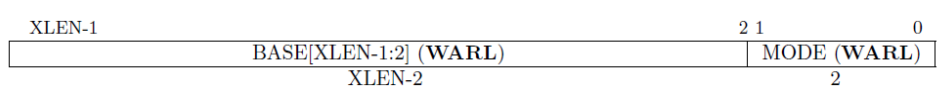
紅框外全部hardwire成0。

* + - 1. mie(Machine interrupt-enable register)



紅框外全部hardwire成0。

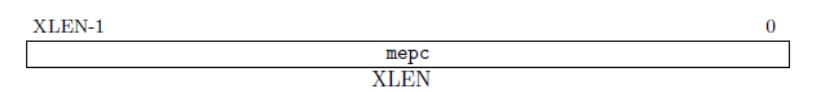
* + - 1. mtvec(Machine Trap-Vector Base-Address Register)



此次只實做Direct Mode(mode == 2’b00)，

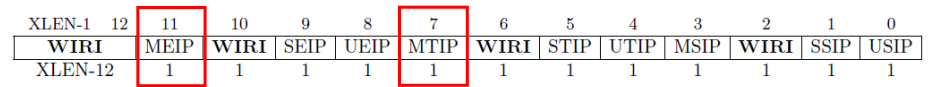
且mtvec hardwire成0x0001\_0000。

* + - 1. mepc(Machine Exception Program Counter)



當中斷發生在WFI執行後，則存入pc+4，如果沒有則存入pc。

* + - 1. mip(Machine interrupt-pending register)



紅框外全部hardwire成0。

* + - 1. mcycleh & mcycle

一張含有 桌 的圖片

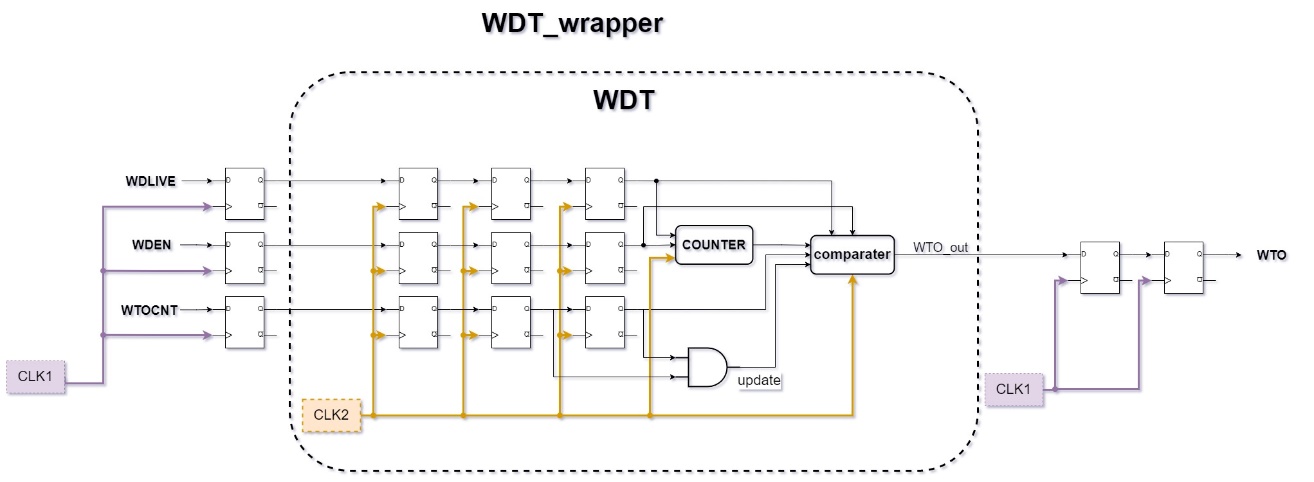
自動產生的描述

* + - 1. minstreth & minstret

一張含有 桌 的圖片

自動產生的描述

* 1. WDT Block overview



1. CPU發送WTOCNT設定WDT觸發WTO(timeout)所需的時間

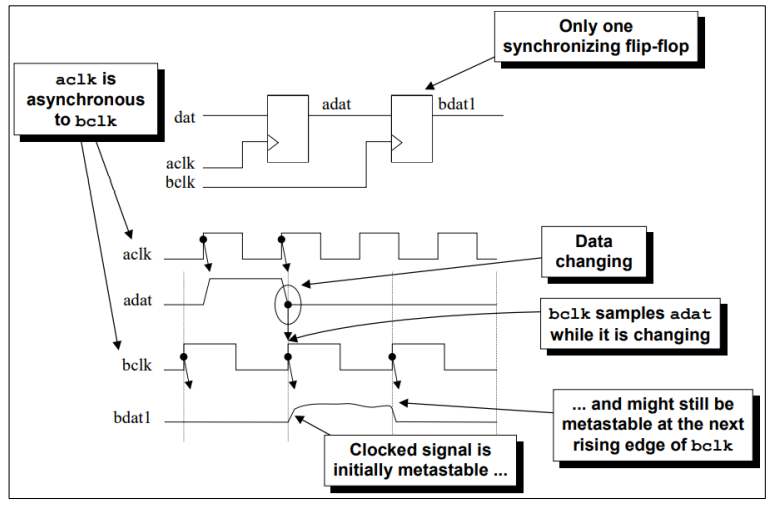
ii. CPU拉起WDEN訊號使WDT中的counter開始計數

iii. CPU發出WDLIVE訊號刷新counter，使counter歸零。

iv. 當CPU沒有發出WDLIVE刷新counter，使counter值大於WTOCNT時，WDT拉起WTO訊號給CPU

，發生timeout interrupt。

v. 在不同clock domain中傳遞訊號可能會產生metastable的問題，如下圖



因此在設計中使用兩層Flipflop當作單bits的 synchronizer，降低metastabe發生的機率。

而在多bits中使用兩層Flipflop依然會造成訊號出現錯誤的值(2Flipflop synchronizer的Delay具隨機性，可能一個周期或2個週期才同步成功)。但在本次設計中考慮到WTOCNT並不會時常更新，並且觸發WTO的時機通常較遠，因此仍使用2Flipflop當作synchronizor，同時使用AND邏輯閘來當作判定WTOCNT訊號已經穩定，更新Counter內的WTOCNT。

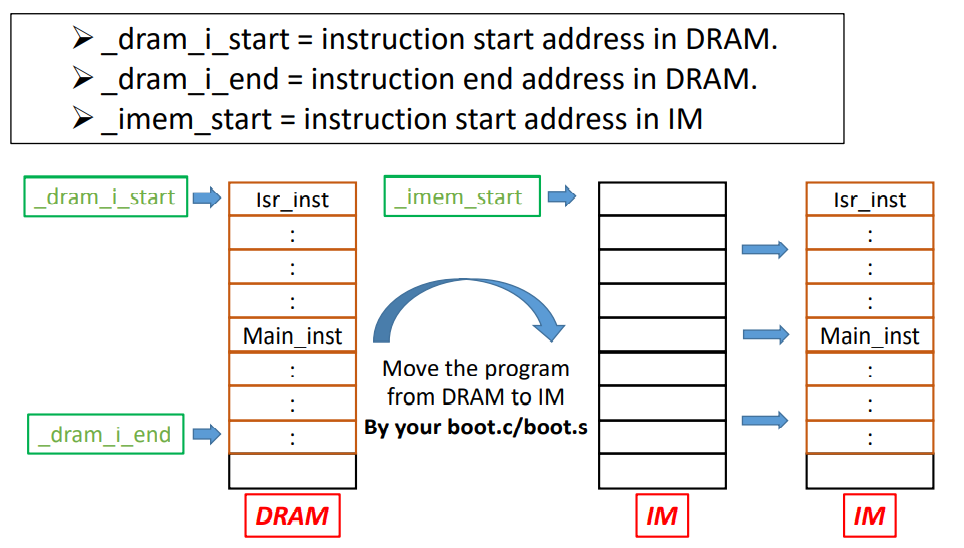
1. **Boot.c explaination**

在CPU剛開始運作時，會先執行存在ROM中的Boot.c，而在Boot.c中所記載的資訊為將存放在外部的instruction和data搬進內部memory的程式碼。

一張含有 文字 的圖片

自動產生的描述

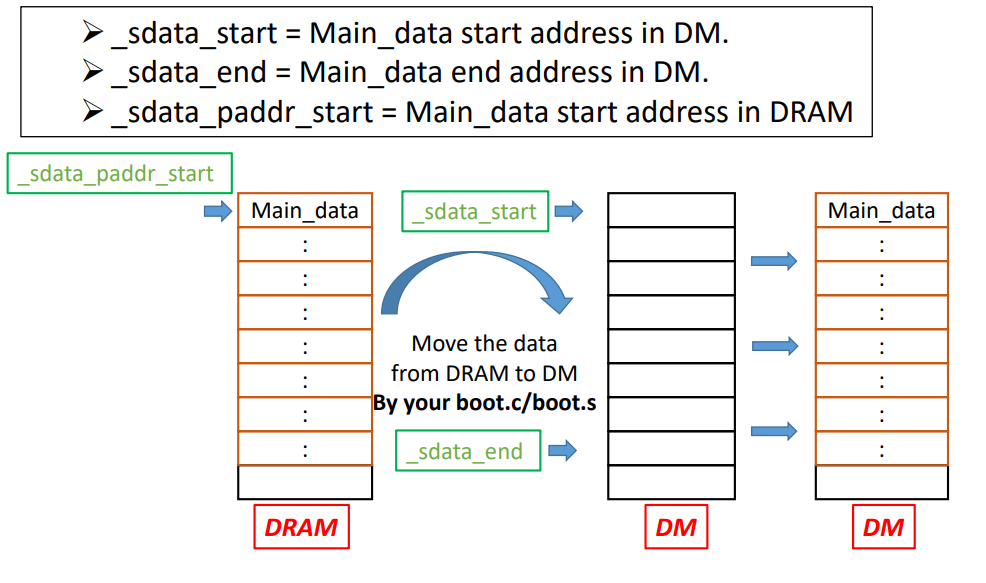
如同上圖程式碼，先計算存在DRAM中instruction的長度，接著使用for迴圈將instruction從DRAM搬進Instruction Memory. (如下圖所示)



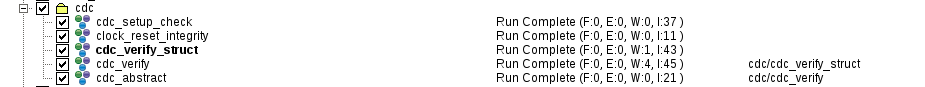
同上述做法，接續著將DRAM中的DATA搬進Data Memory. (如下二圖所示)

一張含有 桌 的圖片

自動產生的描述



1. **Spyglass CDC report**
2. Spyglass CDC result



ii. CDC\_verify\_struct



AC\_conv01: Reports signals from the same domain that are synchronized in the same destination domain and converge after any number of sequential elements

隨然有發生聚合問題的warning，但由於有在WDT\_wrapper中實現延長輸入pulse的電路，因此就算產生聚合問題而造成經過Synchronized的訊號不同步

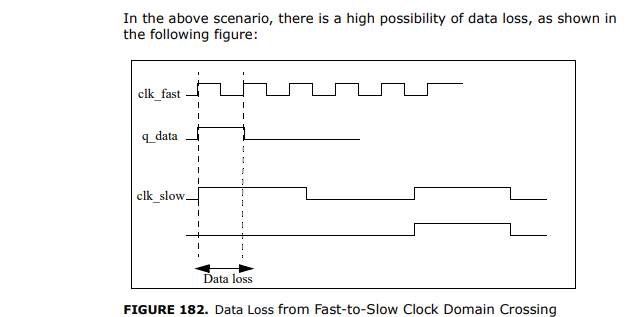
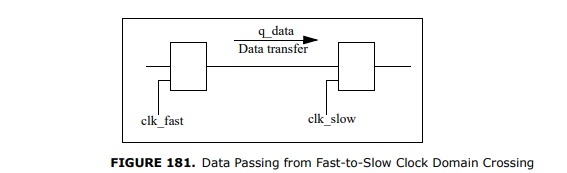
，到最後也會穩定，並且不會使WTO拉起而造成不預期的中斷產生。

iii. CDC\_verify

一張含有 文字 的圖片

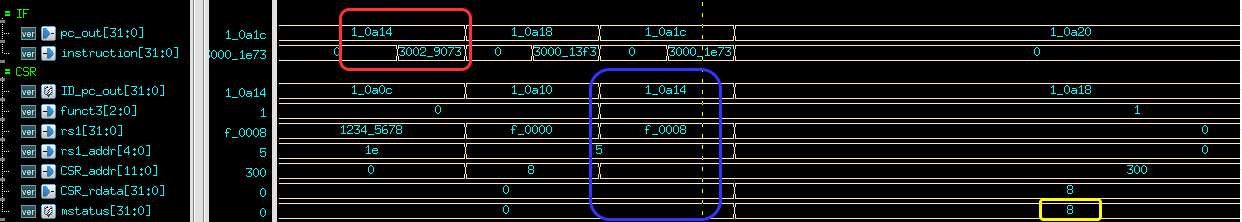
自動產生的描述

Ac\_cdc01a :Checks data loss for multi flop or sync cell or qualifier synchronized clock domain crossings

Example: 

即使發生Ac\_cdc01a的warning，但由於有Hold住input的pulse，因此即使是一個pulse也能讓其延長至被clk2的FLIPFLOP給栓鎖進去。

1. **Waveform explanation**
2. CSRRW



1. 在紅色框中，pc=1\_0a14,instruction為0x3002\_9073

經過decode後可得知此assembly code為

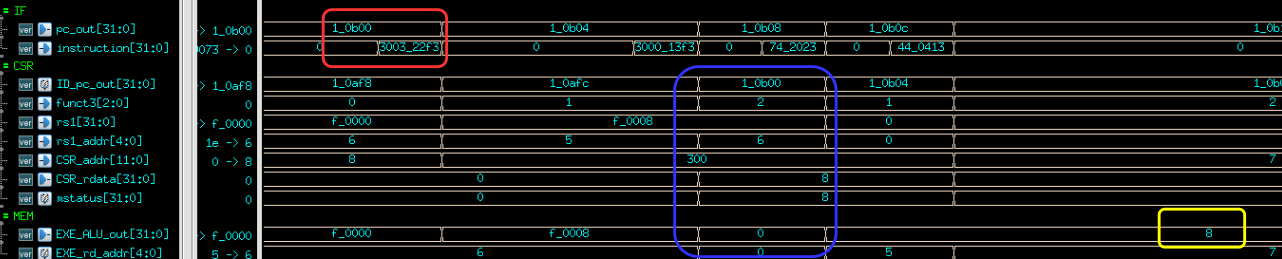
“csrrw zero,mstatus,t0”。

1. 藍色框中，可發現t0的值為0xf\_0008，由於在這次的實作中限定除了mstatus中第[12:11],[7],[3]等bits(即MPP,MPIE,MIE)都必須hardwire成0，因此可得知mstatus中的MPP=0,MPIE=0,MIE=1。

而RD為zero，因此無法將mstatus寫入register中。

1. 經過一個clk，成功將數值寫入mstatus中。(黃色框)

ii. CSRRS



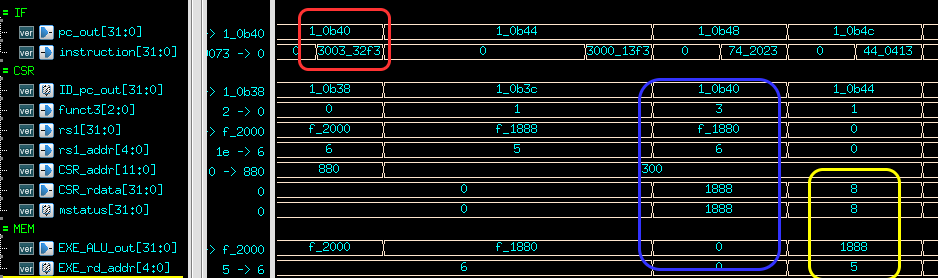
1. 在紅色框中，pc=1\_0b00,instruction為0x3003\_22f3

經過decode後可得知此assembly code為

“csrrs t0,mstatus,t1”。

1. 藍色框中，可發現t1的值為0xf\_0008，因此將 mstatus “OR” t1後，傳給mstatus，並且將mstatus中的值傳給t0。
2. 在下一個clk可發現傳出的值為8，成功確認mstatus值被傳出。(黃色框)

iii.CSRRC



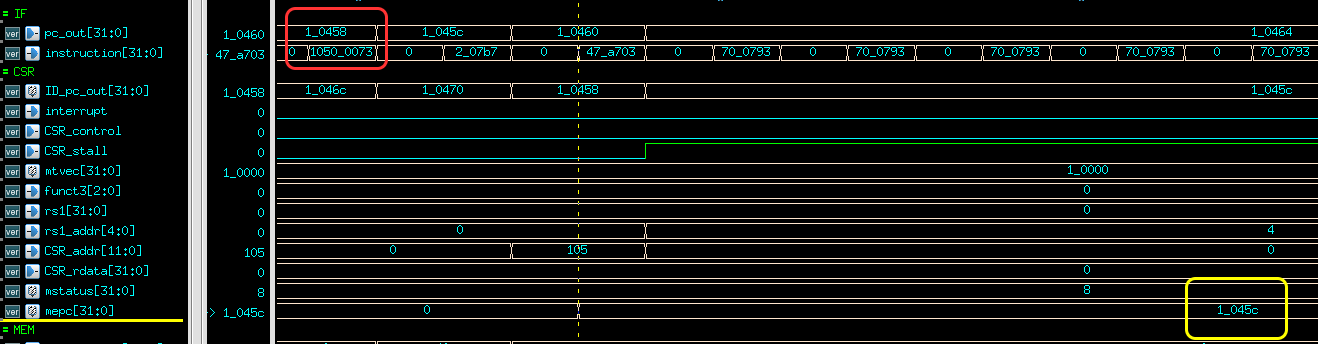
1. 在紅色框中，pc=1\_0b40,instruction為0x3003\_32f3

經過decode後可得知此assembly code為

“csrrc t0,mstatus,t1”。

1. 藍色框中，可發現t1的值為0xf\_1880，因此將 mstatus “AND” ~t1後，傳給mstatus，並且將mstatus中的值傳給t0。
2. 在下一個clk可發現傳出的值為0x1888，成功確認mstatus值被傳出，並且mstatus值更新為0x8。(黃色框)

iv.WFI



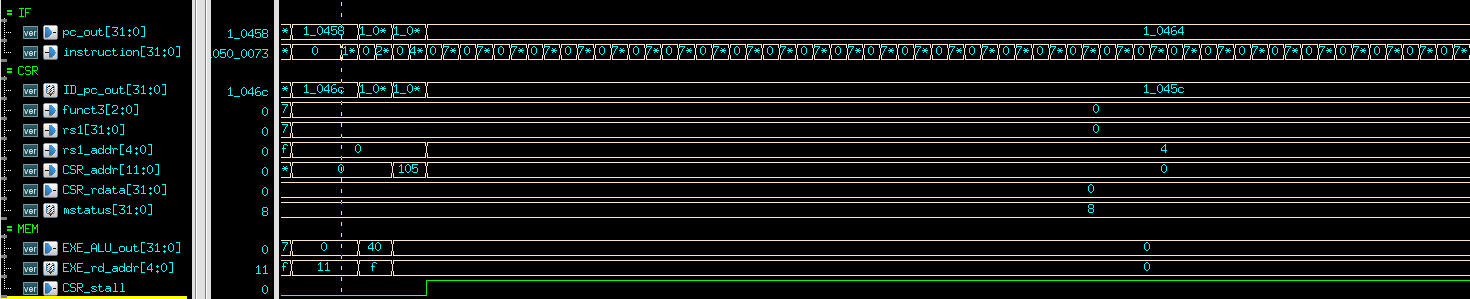
1. 在紅色框中，pc=1\_0458,instruction為0x1050\_0073

經過decode後可得知此assembly code為

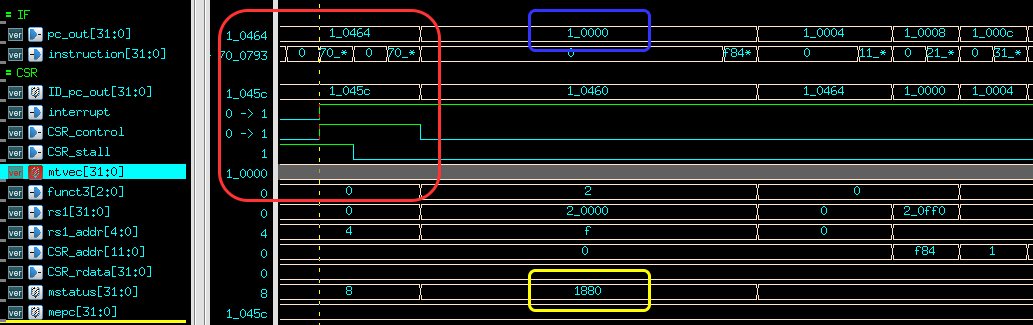
“WFI”。

1. 當EXE stage接收到WFI instruction ，便會觸發CSR\_stall，讓CPU暫停等待Interrupt。(如下圖)

3.同時將pc+4寫入mepc(即0x10458+4=0x1045c)



v.Interrupt occurred



* 1. 在紅色框中可以發現中斷被拉起，因此CSR\_stall拉下且CSR\_control拉起，準備跳去mtvec所儲存的位置(0x1\_0000)
  2. 成功跳入01\_0000(藍色框處)
  3. 當中斷發生時，MPIE<=MIE , MIE<=0,MPP<=2’b11(Machine Mode),

因此mstatus從0x08轉變為0x1880。

vi.MRET

一張含有 文字, 電子用品, 電腦 的圖片

自動產生的描述

1. 在紅色框中，pc=1\_0104,instruction為0x3020\_0073

經過decode後可得知此assembly code為

“mret”。

1. 藍色框中，可發現在EXE stage中的CSR接收到mret，拉起CSR\_ret，使PC在下一個clk變成中斷前所儲存的位置。
2. 成功跳回中斷前所儲存的位置(0x1045c)。(橘色框)
   1. 當結束中斷，將MPIE<=1 , MIE<=MPIE,MPP<=2’b11(Machine Mode),

因此mstatus從0x80轉變為0x1888。

1. **Synthesize /APR result**
2. Synthesize (CLK1 = 12.5ns , CLK2 = 100ns)
   * 1. AREA

一張含有 文字 的圖片

自動產生的描述

AREA = 6639610.16

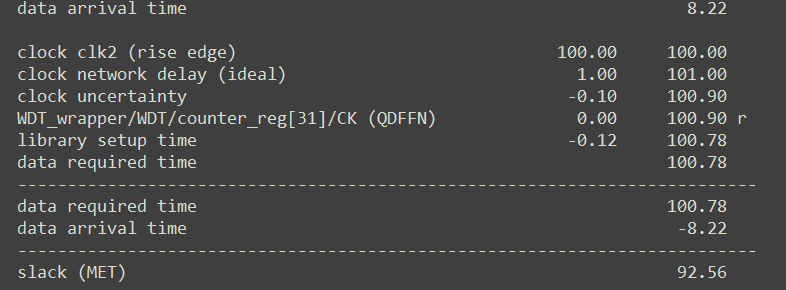
* + 1. Timing

a.CLK1 (12.5ns)

一張含有 文字 的圖片

自動產生的描述

b.CLK2 (100ns)



ii. APR (CLK1 = 12.5ns , CLK2 = 100ns)

1. DRC Verify:

一張含有 桌 的圖片

自動產生的描述

1. CONNECTIVITY Verify

一張含有 文字, 報紙, 收據 的圖片

自動產生的描述

1. ANTENNA Verify

一張含有 文字 的圖片

自動產生的描述

1. AREA

一張含有 文字 的圖片

自動產生的描述

1. POWER

一張含有 桌 的圖片

自動產生的描述

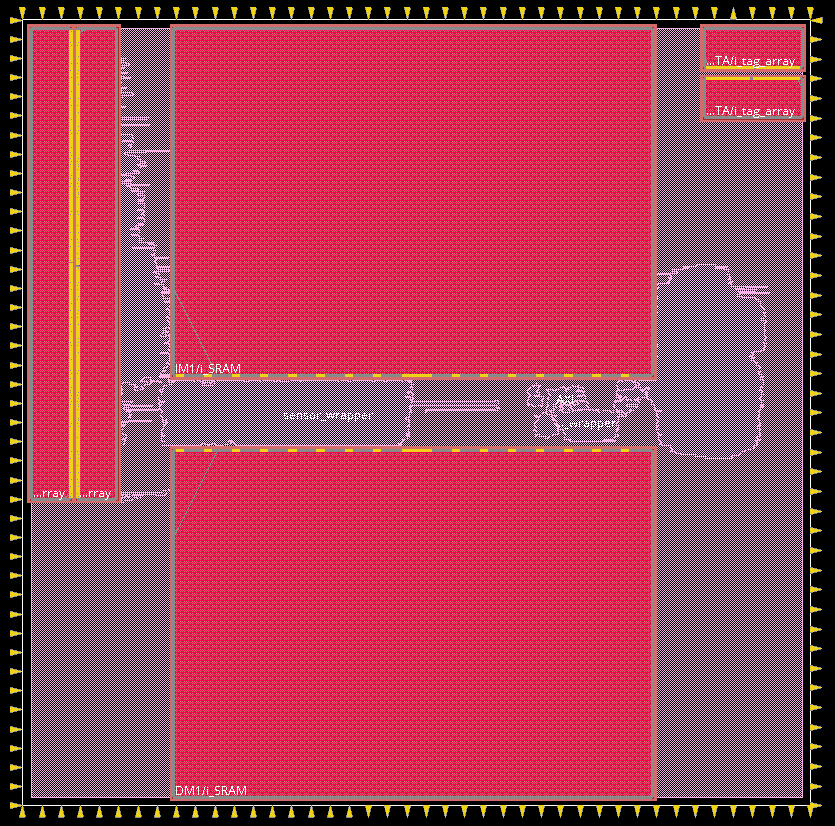
Total power = 175.06mW

1. Floorplan View

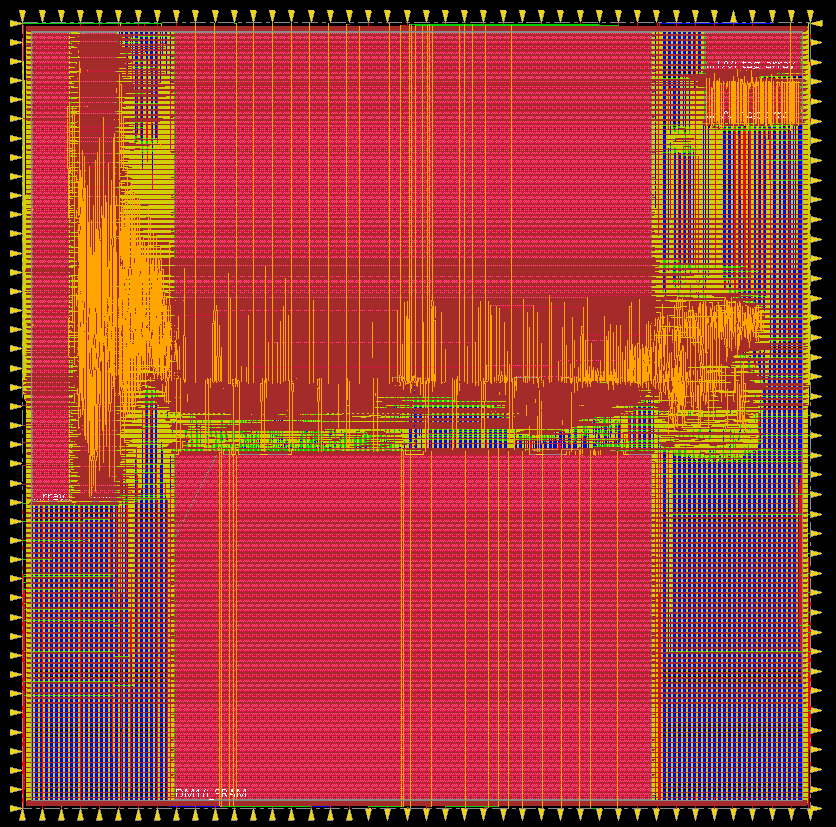
一張含有 文字, 電子用品 的圖片

自動產生的描述

1. AMOEBA View



1. Physical View



1. **Superlint**
2. Superlint Result

一張含有 文字 的圖片

自動產生的描述

ii. Total number of line of RTL code



1. **Simulation Result**
   * + 1. **RTL0: 1138212500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **RTL1: 15097437500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **RTL2: 1068562500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **RTL3: 7440062500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **RTL4 : 7441325 ns**

**一張含有 文字 的圖片

自動產生的描述\**

* + - 1. **SYN0: 1138212500 ps**

一張含有 文字 的圖片

自動產生的描述

* + - 1. **SYN1: 15097437500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **SYN2 : 1068562500 ps**

**一張含有 文字 的圖片

自動產生的描述**

* + - 1. **SYN3: 7440062500 ps**

**一張含有 文字 的圖片

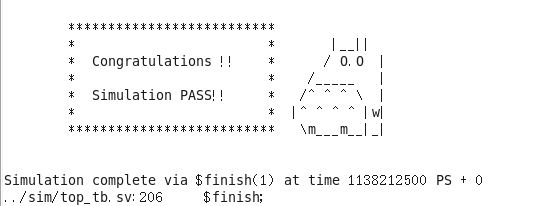
自動產生的描述**

* + - 1. **SYN4: 7441325 ns**

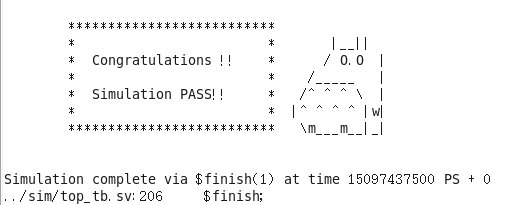
**一張含有 文字 的圖片

自動產生的描述**

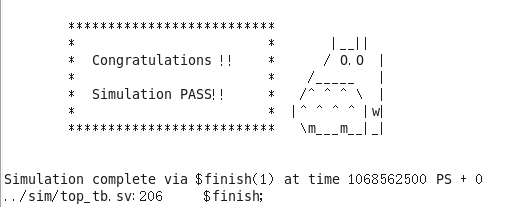
* + - 1. **PR0 : 1138212500 ps**

****

* + - 1. **PR1: 15097437500 ps**



* + - 1. **PR2: 1068562500 ps**

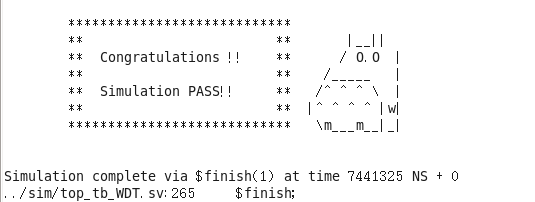
****

* + - 1. **PR3: 7440062500 ps**

**一張含有 桌 的圖片

自動產生的描述**

* + - 1. **PR4: 7441325 ns**

****

1. **Lesson Learned**

在這次作業當中，學到了中斷是如何執行以及實作，由於在之前對於中斷的執行過程不太熟悉，因此花費了很多的時間在了解中斷的原理。除此之外，遇到了在過去未曾碰見過的CDC(Cross Domain Clock)問題也是一大挑戰，在這過程中了解到許多CDC為何會造成問題的原因以及各種影響的分類。

而在實作過程中，其中有遭遇到模擬解結果皆正確，卻在Synthesize時Slack為負，由於本次作業要求CLK上限為20ns，因此無法將CLK往上調，花費了許多時間在改善電路設計，用pipeline將Critial path給縮小，使設計能夠通過合成並且繼續APR流程。