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: \_\_\_\_張正德\_\_ , \_林晉宇\_\_\_\_\_\_\_

Student ID: \_P76121089\_, \_P76121673

**Summary**

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

**Contribution**

|  |  |
| --- | --- |
| P76121089 | P76121673 |
| 50% | 50% |

**Hardware Design Description**

* System Block Diagram
* Interrupt mechanism description and flow chart
* DRAM wrapper FSM chart
* WDT & CDC circuit description and diagram

**System Block Diagram**

**Interrupt mechanism description and flow chart**

**DRAM wrapper FSM chart**

**WDT & CDC circuit description and diagram**

**Software & Firmware design description**

* Prog 1

使用insertion sort，並將指定陣列按照ascending排序。

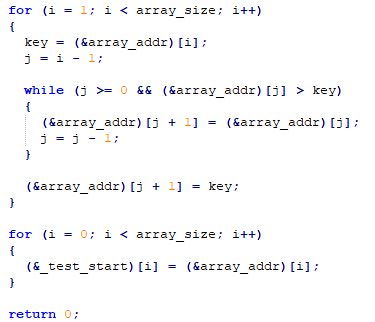
宣告外面變數 `array\_size` 表示陣列大小

宣告外部變數 `array\_addr` 表示陣列起始位址

宣告外部變數 `\_test\_start` 表示排序後陣列的儲存位址。



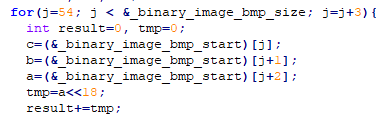
再利用for loop 進行insertion sort，最後存將結果的array依序存進test位址中。



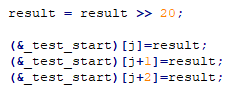
* Prog 2
  + 先將bmp中的header搬入test的開頭



* + 再將BGR資料分別取出，接著個別計算後再加總



* + 最後再將結果存回test中



* Booting

在main.c執行前，藉由boot.c來將DRAM中的資料搬到IM與DM之中。

一開始CPU會去讀取ROM中的指令，而ROM的指令為以下boot.c編譯而成。

**//設定將要搬移的指令數量**

**len = (&\_dram\_i\_end) – (&\_dram\_i\_start) + 1**

**//將dram中的指令搬移到IM中**

**for (i=0; i< len; i++)**

**(&\_imem\_start)[i] = (&\_dram\_i\_start)[i];**

**//設定將要搬移的資料數量**

**len = (&\_\_sdata\_end) – (&\_\_sdata\_start) + 1;**

**//將dram中的資料搬移到DM中**

**for (i=0; i<len; i++)**

**(&\_\_sdata\_start)[i] = (&\_\_sdata\_paddr\_start)[i];**

**//設定將要搬移的資料數量**

**len = (&\_\_data\_end) - (&\_\_data\_start) + 1;**

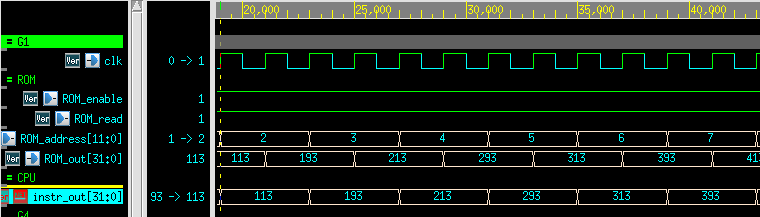
**//將dram中的資料搬移到DM中**

**for(i = 0; i < len; i++)**

**(&\_\_data\_start)[i] = (&\_\_data\_paddr\_start)[i];**

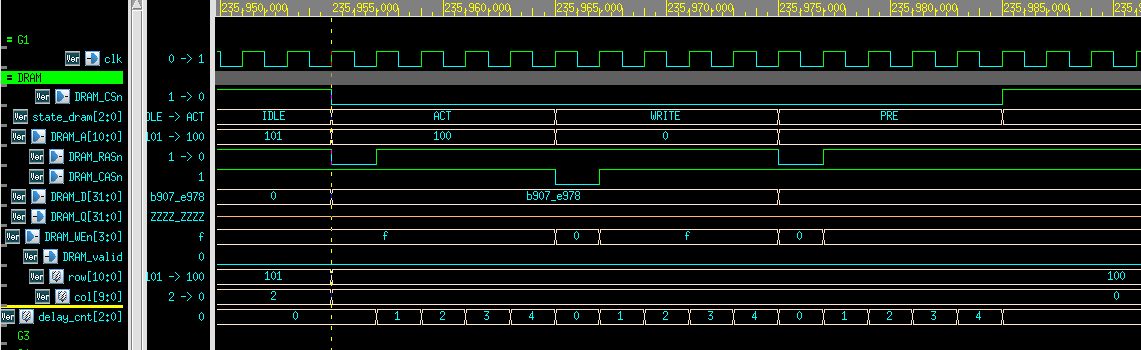
**Screen shot of wave forms and simulation results**

**ROM**

****

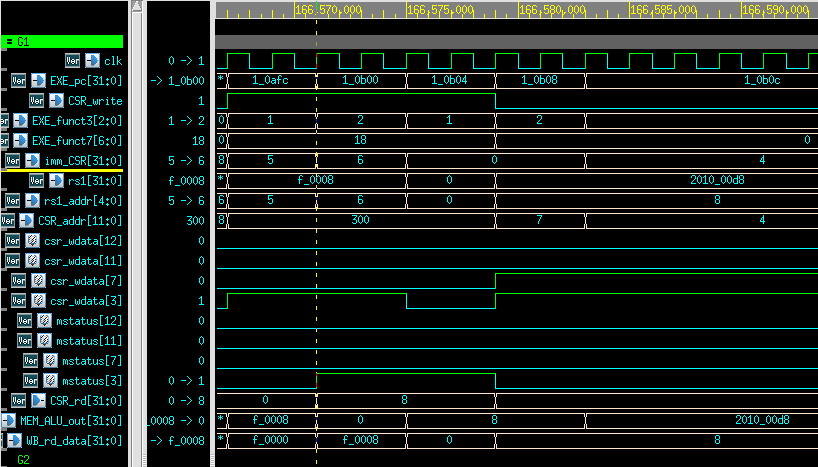
CPU的PC從address 0x0開始讀取ROM的資料到CPU的instr\_out，開始執行booting的程式。可以看到上圖中的ROM\_enable以及ROM\_read皆為high，正在被CPU讀取中。

**DRAM**

****

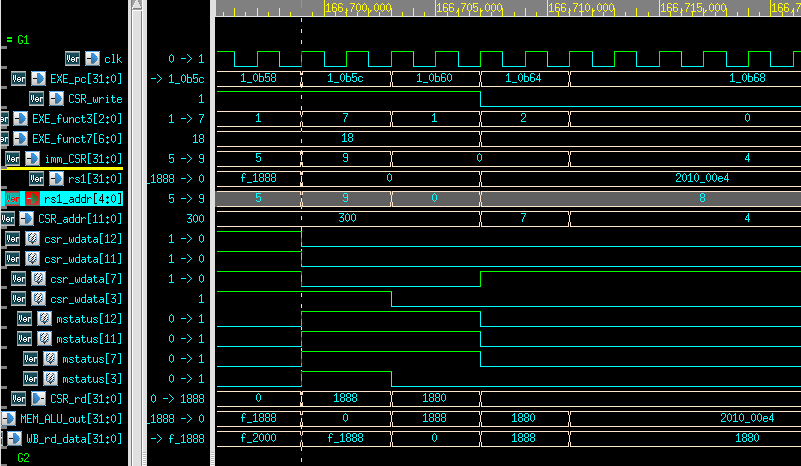
**CSR instructions:**

1. **CSRRS :**

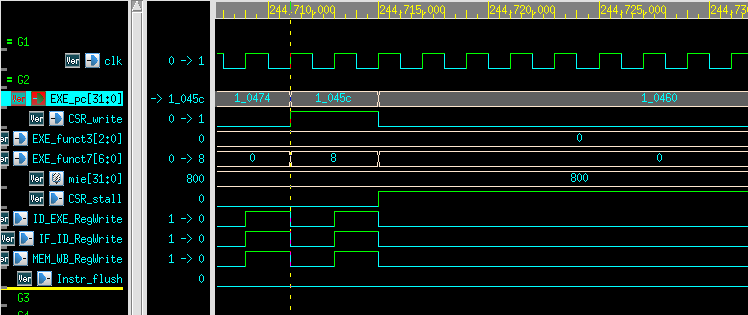
****

紅色框處EXE\_pc=1\_0b00，見編譯後的main.log可知指令為CSRRS ，

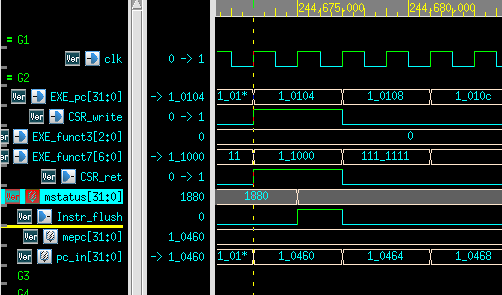
1. **CSRRCI:**

****

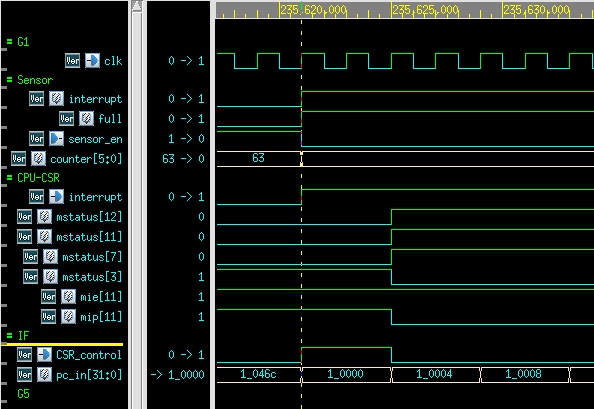
1. **WFI**

****

1. **MRET**

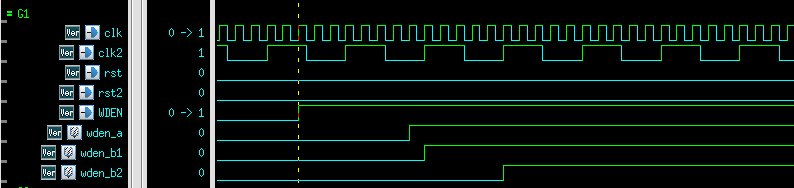
****

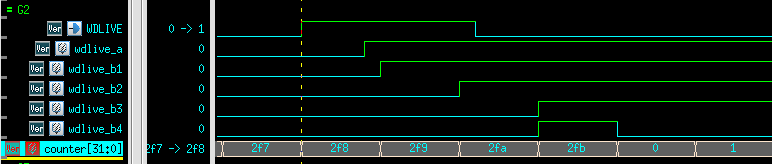
**Sensor Control and Interrupt**

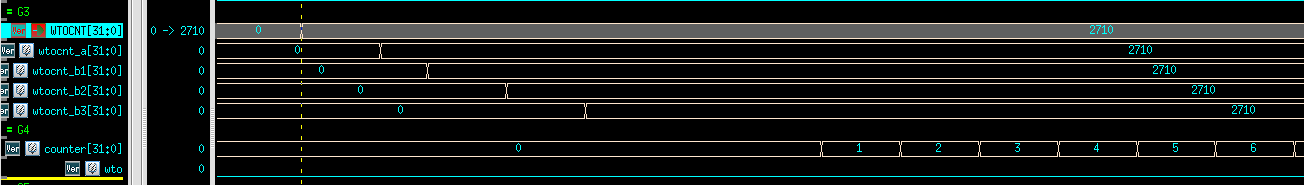
****

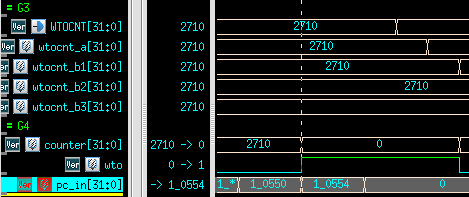
**Watch Dog Timer**

**Timeout**

****

****

****

****

**Spyglass CDC :**

**Lesson Learned**

**Problems to answer**

1. What is the deference between mcycle and timer? When is mcycle used?
2. What is “Potential Qualifier” in Spyglass?

**Lesson learned**