VLSI System Design (Graduate Level)

Fall 2023

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: \_\_\_張正德\_\_\_ \_\_\_\_林晉宇\_\_\_\_

Student ID: \_\_P76121089\_\_ \_\_P76121673\_\_

**Contribution**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Performance & Area | | | |
|  | rtl | syn | pr |
| Prog0 time | 4948780000 | 4949000000 | x |
| Prog1 time | 45000200000 | x | x |
| Prog2 time | 3715000000 | 3714020000 | x |
| Prog3 time | 10013420000 | 6962810000 | x |
| Area(um^2)  In APR | X | | |
| CPU cycle | 10ns | | |

Refer to the figure to fill in the simulation time !!

|  |  |  |
| --- | --- | --- |
|  | rtl | syn |
| Prog0 |  |  |
| Prog1 |  |  |
| Prog2 |  |  |
| Prog3 |  |  |

// Initiate your report from this point.

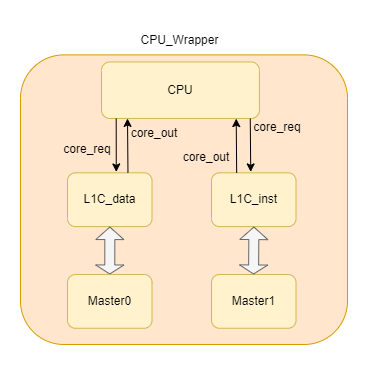
Block Diagrams

1. System Architecture

一張含有 文字, 螢幕擷取畫面, 圖表, 設計 的圖片

自動產生的描述

1. CPU\_wrapper – Cache - Master



1. FIFO

RA-channel:

一張含有 文字, 圖表, 方案 的圖片

自動產生的描述

RD-channel :

一張含有 文字, 圖表, 方案 的圖片

自動產生的描述

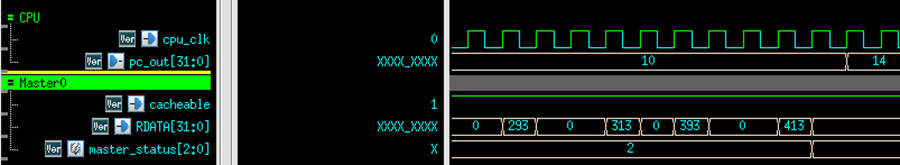
Waveform verification

1. Cache read miss

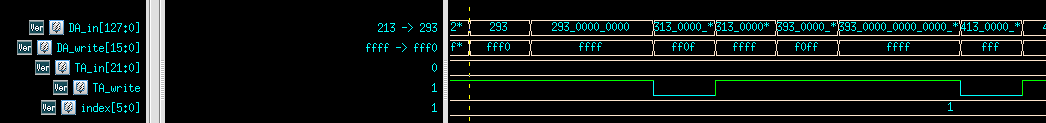
一張含有 螢幕擷取畫面, 文字, 軟體, 多媒體軟體 的圖片

自動產生的描述

在pc\_out為10時,CPU對L1C\_inst發出core\_req請求指令,core\_wait設為1直到正確資料回傳給CPU , core\_write為0因為此時為請求instruction指令並且core\_addr為10,所以是去ROM讀取指令,可以看到此時hit為0代表L1C\_inst中並沒有暫存這個指令,所以cache\_state = 2轉為read\_miss狀態,設定送往master0的資料來透過AXI讀取ROM,此時將read設為1因為是要讀取資料,將ARADDR設為10來讀取ROM,ARLEN設為3因為一個block有4個word需要讀出連續4筆資料傳回cache,ARVALID設為1開始透過AXI存取資料,cacheable一直設為1因為master0只會讀取ROM或IM並不會讀取sensor\_wrapper , master\_status = 2為讀取資料狀態



Master0等待4筆資料(293,313,393,413)回傳後將資料傳給cache存進Data\_array以及Tag\_array



依照DA\_in及DA\_write將四筆資料存入Data\_array後，一張含有 螢幕擷取畫面, 多媒體軟體, 軟體, 電路 的圖片

自動產生的描述

最後透過core\_out將指令傳給CPU中的IF-stage。

1. Cache read hit

一張含有 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

自動產生的描述

在經過pc\_out為10 的指令後,因為指令是連續讀取的,所以後面的三個指令都已經暫存在L1C\_cache中,所以可以看到後面三個指令hit都為1代表資料在cache中,不需要透過master0去讀取memory中的指令,所以會將I\_req設為0表示不需要去存取memory,可以看到core\_out直接將暫存的指令輸出給CPU中的IF-stage。

1. Cache write miss (write through)

一張含有 螢幕擷取畫面, 行 的圖片

自動產生的描述

當pc\_out為180時,MEM-stage要對DM\_addr為20fec寫入資料store\_data為1,所以CPU會發出core\_req並將core\_addr設為20fec , core\_in為要寫入的資料設為1,因為現在為hit所以必須對master1下指令去寫入AWADDR為20fec, WDATA為1

一張含有 螢幕擷取畫面, 文字, 軟體 的圖片

自動產生的描述

因為是hit所以cache中也有資料所以必須更新Data\_array中的資料。

1. FIFO
   1. Master0 to ROM channel

一張含有 螢幕擷取畫面, 文字, 鮮豔 的圖片

自動產生的描述

* 在PC\_out 為0時，master0會去ROM搬instruction回來CPU中，因此當整個系統開始時，是master0透過M0\_AR\_fifo再經由AXI給到AR\_S0\_fifo，上圖為其過程的波形截圖。
* 紅框處可以看到，在進入M0\_AR\_fifo之前，會先將data打包成一組資料，即{ARID, ARADDR, ARLEN, ARSIZE, ARBURST}，再等待wen拉起傳入fifo的wdata中。
* M0\_RA\_fifo的wen設定為ARVALID。
* M0\_RA\_fifo的ren設定為 ~RA\_S0\_fifo的wfull (黃框處)，因為當接收端的fifo寫滿時若M0\_RA\_fifo仍在讀出資料，可能會在AXI中遺失。(而接收端為哪一個slave則是透過decoder決定)
* 在上圖中可以看到RA\_S0\_rdata這條線，其是由RA\_S0\_fifo讀出，讀出後再依照{ARID, ARADDR, ARLEN, ARSIZE, ARBURST}的順序解開傳給ROM。
* 藍框處可以看到ROM (slave0)正確接收到由master0傳出的資料。
  1. ROM (slave 0) to Master0 channel

一張含有 螢幕擷取畫面, 文字, 軟體, 陳列 的圖片

自動產生的描述

* 紅框處可以看到，在進入S0\_R\_fifo之前，會先將data打包成一組資料，即{ RID, RDATA, RRESP, RLAST, R\_hs\_done}，再等待wen拉起傳入fifo的wdata中。
* S0\_R\_fifo的wen設定為RVALID。
* S0\_R\_fifo的ren設定為 ~R\_M0\_fifo的wfull。
* 藍框處可以看到Master0正確接收到由ROM傳出的資料。
* 此處的細節將在lesson learned提到。

IM/DM cache hit rate

1. IM cache hit rate:

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

上圖為L1C\_inst的片段程式碼用以計算IM cache hit rate。

當cache\_state為CHECK時會比對Tag以及查看是否valid，因此在此狀態下判斷cache\_hit是否加一，而一般讀取情況則將cache\_read加一，作為讀取IM的總次數。

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

查看prog0的波型可以看到在此program執行結束時的hit\_count=174704、read\_count=191803。

IM cache hit rate = 174704/191802 \* 100% = 91.08%

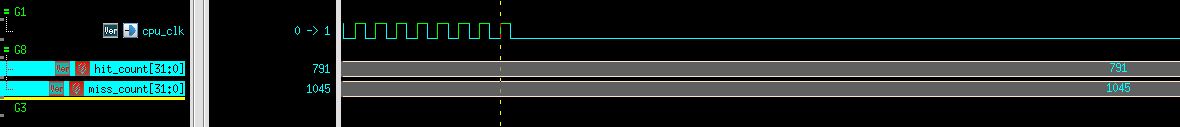
1. DM cache hit rate:

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

上圖為L1C\_data的片段程式碼用以計算DM cache hit rate。

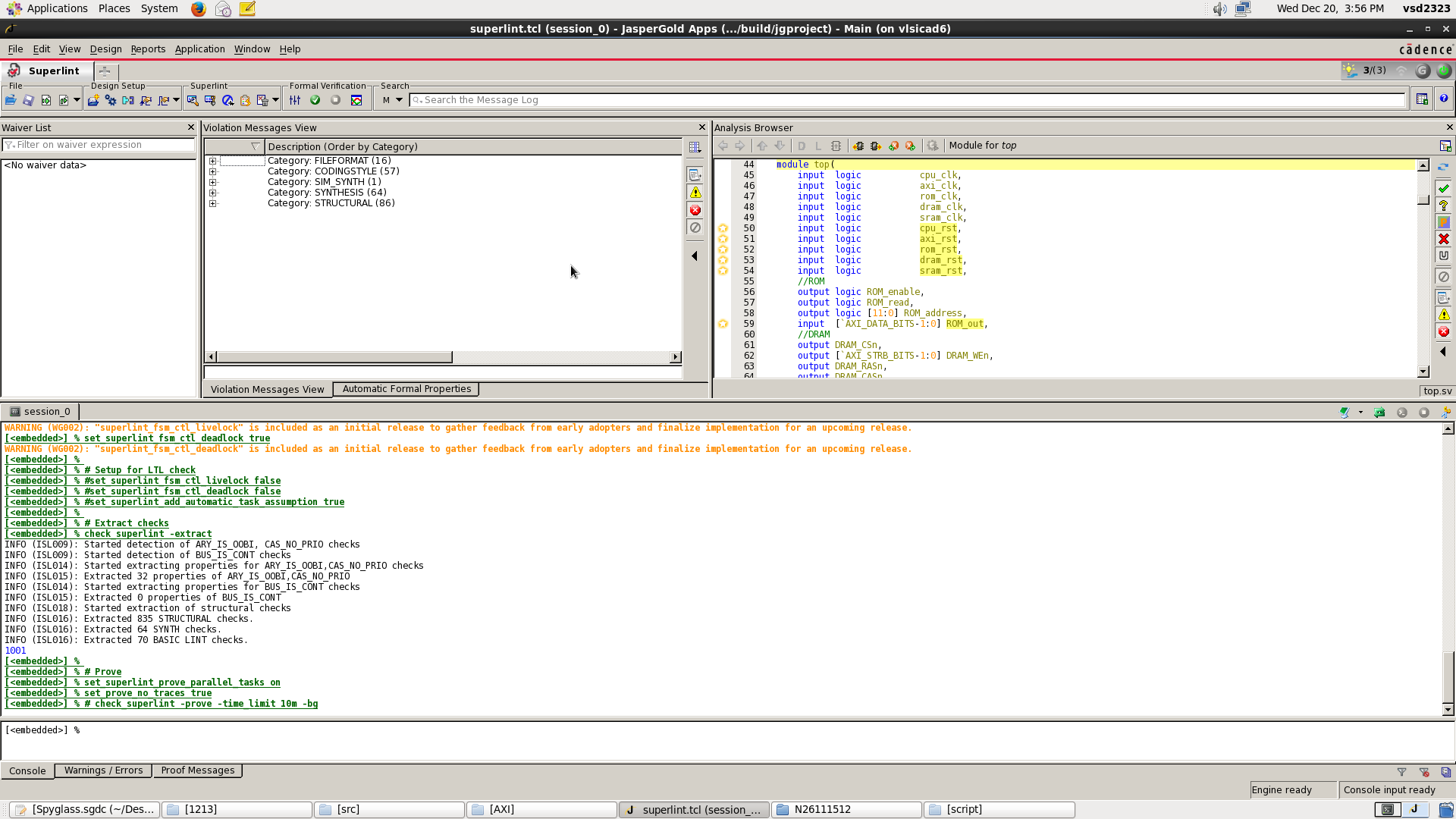
利用core\_addr[31:16]來判斷是否是要存取DM。

當cache\_state為STATE\_WRITE\_HIT以及STATE\_READ\_HIT時代表判斷為hit，而STATE\_WRITE\_MISS以及STATE\_READ\_MISS則是判斷為hit。  


查看prog0的波型可以看到在此program執行結束時的hit\_count=791、miss\_count=1045，total access count=1836。

DM cache hit rate = 791/1836 \* 100 % = 43.3%

Superlint



Total number of lines : 9316

100% - (224 / 9316)\*100% = 97.6%

Lesson learned

在這次作業中，我們遇到重重困難，一開始就遇到程式只能最多執行3個clk，就會壞掉，經過一天的實驗最後才發現問題在新加入的cache中。

另外在這次多個不同時域，加入FIFO後，可能會發生full以及empty的狀況，遇到最多問題的是在DRAM slave上，因為他本身有Timing 的規範，需要注意更多情形。

一張含有 螢幕擷取畫面, 多媒體軟體, 軟體 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, 軟體, 繪圖軟體 的圖片

自動產生的描述

因為時域不同的關係，在slave的地方address會等待handshake後才開始按照ARLEN的長度以及ARBURST的設定做往後加的動作，因此在本次作業雖然RVALID已經拉高，但是master端並不會直接接收到，因為當master端接收到RVALID後，也會相應的計算count來接收對應資料，但是因為時域不同，ROM端的資料送出較慢，而CPU端會有資料遺失的狀況。

因此我們在本次作業新增一條R\_hs\_done，由slave送出到RD channel中，新增(R\_hs\_done)?的條件在master接收RVALID之時。

因此可以看到上圖波形中master端的RVALID會和資料同時high以及同時low。