# **515152電子設計自動化演算法與實作 Electronic Design Automation Algorithms and Implementation**

**ICCAD contest Problem D:**

**APB Transaction Recognizer**

**組員1: (109261008)李謝昀**

**組員 2: v(113065537)黃奕立**

##### 題目簡介

設計並實作一套 VCD（Value Change Dump）解析器，能夠即時解析 APB 匯流排交易，並針對交易過程中的多種錯誤情境進行偵測與統計。透過此工具，可協助設計者驗證 APB （Advanced Peripheral Bus）匯流排的交易合法性、Completer 連線狀態、交易效率與錯誤分析。

競賽網站公告的題目介紹中嘗試引導介紹驗證電路的三種常見型態，基於模擬（Simulation-Based Verification）、形式驗證（Formal Verification）以及基於測試平台（Testbench-Based Verification）的三種驗證流程，但實際上題目的要求被限縮在觀察AMBA（Advanced Microcontroller Bus Architecture）中的APB協議，並且主要以識別其中三種APB Protocol為任務目標，分別為UART、GPIO、SPI Master。

任務目標為透過識別APB交易過程中六個主要訊號，即PCLK(時脈)、PADDR(地址)、PWRITE(讀/寫標誌)、PENABLE(始能訊號)、PSEL(外設選擇訊號)、PWDATA/PRDATA(數據)，識別APB Transaction、從Bus使用率交易數量週期解析Bus行為，最後驗證並且記錄限定的六種APB交易錯誤。

##### 方法流程圖以及pseudo code

(a) 流程

VCD 檔案讀取

│

▼

逐個Timeframe解析 VCD 變化

│

▼

餵入 Transaction Analyzer (詳細請看以下pseudo code)

│

├── 判斷交易起始 (Setup Phase)

│

├── 判斷 Access Phase

│

├── 判斷 Timeout、Overlap、Mirroring、Corruption

│

├── 判斷 Completer 類型 (UART / GPIO / SPI)

│

▼

交易完成 → 記錄交易資訊

│

▼

統計交易結果、錯誤類型

│

▼

輸出報告

(b) Pseudo code of Transaction Analyzer

由於錯誤類型包含時序上的判斷，需要一個類似學習的過程來判斷錯誤，反之前面有被判斷過的錯誤也能用在後續訊號的判讀，因此六種錯誤的判斷位置會在 Transaction Analyzer的不同階段。

**主流程Pseudo code**

#1 //每個timestamp都會被餵進此函式，分析訊號的變化

函式 feed(change)

- 讀取每個 signal 並更新 signal\_state

- 取得目前時間 timestamp = now

如果尚未進入 transaction 且 (PSEL=1 且 PENABLE=0)

→ 進入 Setup Phase

→ 初始化一筆新的 transaction 記錄 current

如果 in\_transaction 且 經過超過 1000000 ps

→ 標記為 Timeout（超時），儲存並退出

如果 in\_transaction 且尚未進入 Access Phase 且 PENABLE=1

→ 標記 Access Phase 已進入

如果 in\_transaction 且類型為 WRITE，但此時發現 READ setup 嘗試開始

→ 記錄為 Read-Write Overlap Error，儲存至 overlap\_errors

如果 in\_transaction 且 PREADY=1（交易完成）

→ 記錄交易時間、判斷是否有 Wait state

→ 擷取最終 data，計算 Completer ID 與錯誤檢查

→ 呼叫 finalize() 儲存結果

#2 //當有完整交易發生時呼叫finalize儲存結果

函式 finalize(txn)

- 推測 Completer ID

- 建立該 Completer 的浮接錯誤檢查器 (地址/數據)

- 若為 WRITE，將該地址與資料記錄下來

- 檢查 Out-of-Range 錯誤

- 檢查 Data Mirroring 錯誤

- 將 txn 加入 transactions[] 紀錄

**錯誤檢測Pseudo code(Timeout,overlap,corruption已經在Feed時檢查過)**

#3

函式 checkCorruptionBetweenPhases(txn, addr\_access, data\_access)

- 僅針對 WRITE 檢查，若 orig\_data 或 access data 不同

- 對應 PADDR 錯誤：標記 addr\_corrupted 並存 detector

- 對應 PWDATA 錯誤：標記 data\_corrupted 並存 detector

#4

函式 checkOutOfRange(txn)

- 將 binary PADDR 轉為 uint32\_t

- 檢查是否落在 UART/GPIO/SPI 的合法地址範圍  
 - 若落在合法區域且 data 為 X，標記 completer\_unconnected  
 - 若不在合法範圍 → 標記為 out\_of\_range

#5

函式 checkMirroring(txn)

若 txn 是 WRITE

- 將該 (addr, data) 記入 mirror\_map

- 若 data 是首次出現，記入 mirrored\_write\_values

若 txn 是 READ 且該地址從未被寫入

- 若該 data 是之前某處合法寫入的值

- 標記為 data\_mirrored，並記錄來源地址

##### 方法介紹

1. VCD 檔案解析

使用 VCDStreamer 讀取並解析 VCD 檔案，建立符號表與即時訊號變化 紀錄。主要流程：

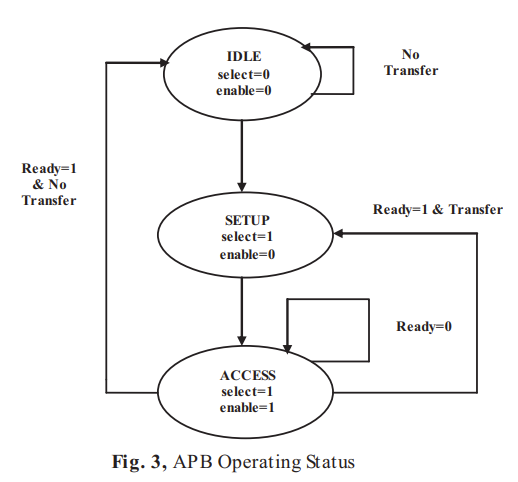
建立 symbol → signal name 映射表。

逐行解析時間點與訊號變化。

透過 callback 將每筆變化傳遞給 TransactionAnalyzer。

2. Transaction 分析

題目需解析的VCD是單一時序的序項解析過程，解析後的映射表不需要再經過複雜的資料結構預處裡。



(合法APB 操作過程的State Diagram ，按參考資料一Fig. 3.)

使用 TransactionAnalyzer 進行交易解析，若該筆交易是合法的(但有可能 是錯誤的)並即時偵測以下六種錯誤：

Timeout：超過預設週期未完成。

Mirroring：讀取到其他地址曾寫入的資料。

Overlap：新的讀交易與尚未完成的寫交易重疊。

Address/Data Corruption：交易過程中地址或資料異常改變。

Out-of-Range：PADDR 不屬於合法 Completer 範圍。

3. Completer 判斷

透過 PADDR 範圍自動對應 Completer：

UART：0x1A100000 ~ 0x1A100FFF

GPIO：0x1A101000 ~ 0x1A101FFF

SPI Master：0x1A102000 ~ 0x1A102FFF

4. 錯誤報告與統計

輸出以下統計項目：

各類交易數量（讀取/寫入、含等待週期/不含等待週期）

Bus 使用率

平均交易週期

Timeout 次數

Out-of-Range 次數

Mirror 次數

Overlap 次數

Completer 連線狀態

Address/Data Corruption 位元浮接分析

1. 可能的調整方向

競賽的得分方式，主要在於APB 交易解析的正確性，同分比序才會加入Cpu程式執行時間做為參照，因此針對測題D所限定的測資範圍，未來可以嘗試導入File I/O Buffering，針對不同錯誤類別設計個別模組化以實現並行處理來加速。

1. 組員貢獻內容

李謝昀完成主程式骨幹以及測資的產生、以及期末報告，黃奕立完成期末報告整理，目前modularization尚未完成。

##### TestCase 介紹表格

官方原先公告會提供六種錯誤的測資，但目前看不到測資的更新(Date: 6/19)。關於併行化處理，競賽題目中引用的文獻VCD file 大小可以達到200GB 的數量級[2]，目前官方公告最大的Testcase 大小為2MB。

目前官方有提供筆測資，但測資均為無錯誤，因此只能驗證 : 各類交易數量（讀取/寫入、含等待週期/不含等待週期），Bus 使用率，平均交易週期

是否正確。

透過官方提供的資料以及QA，我們自行建立五筆測資，其中分別有Timeout

,Mirroring,Overlap,Address/Data Corruption,Out-of-Range錯誤，並進行測試。

##### 實驗結果比較表格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error type | Timeout | Mirroring | Overlap | Address/Data Corruption | Out-of-Range |
| 是否偵測 | Yes | Yes | Yes | Yes | Yes |

##### 待改進處 :

在錯誤偵測實驗中，由於測資都是手動生成，可能會有比賽的測資無法檢測出錯誤，需要alpha test的結果來驗證。另外效能也有較大的改善空間，4000個transaction會需要0.05s，相比於官方給的結果慢了10倍以上，考慮到錯誤檢測中的corruption會有迴圈檢查哪裡有短路bit，在多達一百萬筆Transaction時可能會讓效能大幅降低。

改善方法 : 改進資料結構，加強錯誤檢測的邏輯。

##### 

##### Reference

[1] K.V. Kumar and R. C. Rao, "Design and Verification of Advanced Microcontroller Bus Architecture-Advanced Peripheral Bus (AMBA-APB) Protocol," 2021 International Conference on Innovative Computing and Communications (ICICV), 2021, pp. 1-5, doi:10.1109/ICICV50876.2021.9388549.

[2] R. V. Anam, P. S. Deogade, and S. S. Shingare, "Accelerated Analysis of Simulation Dumps through Parallelization on Multicore Architectures," 2021 24th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2021, pp. 1-6, doi:10.1109/DDECS52668.2021.9417048.

1. **報名截圖、繳交證明**

**報名截圖**



**繳交證明**

