13th GOLDEN SILICON AWARDS

# D13-073

**An Efficient SoC Debug Platform**

高效能之系統單晶片除錯平台


**隊伍名稱**

蟲蟲獵殺 / Bug Hunter

**隊長**

張欽堯　成功大學電機工程研究所

**隊員**

林倬民　成功大學電機工程研究所
留國凱　成功大學電機工程研究所
吳承融　成功大學電機工程研究所


## 作品摘要


在產品開發過程中，除錯與驗證往往是最耗時的環節，特別是在先進製程中，即使電路設計者已在模擬階段進行各種驗證，仍可能因為某些原因無法保證整體電路下線製造後的正確性。例如模擬驗證時所採用的模擬模型（Simulation Models）無法完全地模擬實際製程元件之特性，或是驗證樣本（Verification Testbench）過多，造成模擬所有樣本所需的模擬時間過長，因此只能選擇部分較具代表性的樣本來進行驗證。上述的原因都可能導致即使在模擬驗證階段，驗證的結果都是正確的，但實際下線製造出來的晶片卻仍有設計上的錯誤（Bug）或因製程參數變異而導致錯誤的發生。

基於上述動機，我們提出了一個有效且系統化的系統單晶片除錯平台（SoC Debug Platform），此平台建構於一個多核心及多重時脈的系統單晶片環境，能對存在於此單晶片中的矽智財電路進行除錯。本平台提供三種方法以觸發除錯機制來幫助使用者對矽智財電路進行除錯。第一種方法為基於時脈週期觸發之除錯方法，其利用矽智財所運作的時脈作為觸發訊號，可在任意週期將系統暫停下來，以觀察軟體或硬體狀態，此方法並可配合執行單步時脈週期（Single Step）和跳躍到指定的時脈週期方式，對矽智財電路進行除錯。第二種方法為事件觸發之除錯方法，該方法在一段時脈週期區間內將數個矽智財電路內掃描鍊上正反器的訊號狀態設為觸發條件，而此方法亦提供相同條件重複觸發和即時重新設定觸發條的功能，對矽智財電路進行除錯。第三種方法為軟硬體交互觸發之除錯方法，可與運行於處理器上之軟體應用程式進行交互觸發，當軟體應用程式進入中斷狀態時，本除錯平台也會使矽智財電路進入除錯狀態（Debug State），反之當矽智財電路進入除錯狀態時，也會觸發軟體應用程式進入中斷狀態。本整合平台對於矽智財電路不論是在產品開發階段或產品下線後的功能驗證，均可大幅提高使用者的測試與除錯效率。


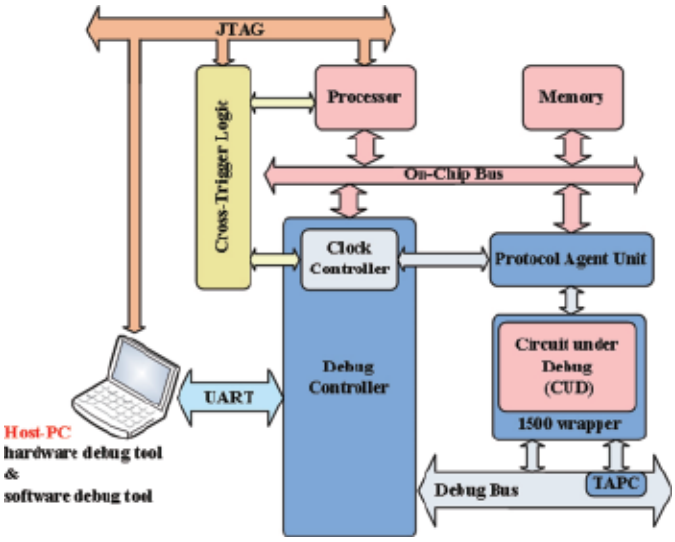


圖1 > 除錯平台系統圖

**指導教授**
**李昆忠 / 成功大學電機工程學系**

分別於 1981 年、1986 年及 1991 年，取得臺灣大學電機系學士、美國愛荷華大學電機及電腦工程系碩士、美國南加州大學電機工程系博士學位。1991 年進入成功大學電機系，1997 年升任教授，期間擔任成功大學晶片系統研發中心主任（2008-2010 年）、台灣積體電路設計學會理事長（2008-2010 年）、工研院晶片系統技術研發中心（現為資通所）諮諮委員（2008 年迄今）、NSOC 國家型計畫專案召集人（2011 年）、國科會智慧電子國家型計畫專案召集人（2013 年迄今）、Steering Committee Member, IEEE Asian Test Symposium（2000 年迄今）、Vice Chair of Steering Committee, IEEE Asian Test Symposium（2010 年迄今）、Executive Committee Member, IEEE VLSI-DAT Symposium（2009 年迄今）、Board member, IEEE Tainan Chapter（2008 年迄今）。

**研究領域**
超大型積體電路設計、系統單晶片測試與除錯、電腦輔助設計、計算機演算法。

## Abstract

Verification and debug are usually the most time-consuming stage in chip development. Even designers try to verify their designs thoroughly, the correctness of first-cut chips still cannot be guaranteed. For example, if process elements are not perfectly modeled by simulation models, character of real chips may be different from the models. Another example is that the amount of verification testbench may be too much to complete the whole verification. Both the above examples may cause that designs are correct during verification but fail after manufacturing.

We thus propose an efficient and systematic SoC debug platform to help designs fix bugs. The SoC debug platform is built on a multi-core system with multiple clock domains, and it is able to figure out when and where bugs occur for IP cores in SoC designs. This SoC debug platform supports three debug methods. The first one is the cycle-based debug. With this method, we can suspend the system during its application at the cycle that we want and observe the internal states of IP cores. After the suspension and observation, the system is able to resume its state and continue its application. This method can also be used for single step debug. The second is the event trigger debug. In this method, if some signals of IP cores under debug match the assigned values in a time period, the system will be suspended. Signals and assigned values can be re-configured during the debug process. The third one is the hardware/software cross trigger debug. With this method, a debug signal requested from hardware designers can trigger software programs, and a debug event from software programs can trigger hardware IP cores as well. In this way, hardware and software designs can be suspended at the same time, and hardware and software designers consequently can debug together. The proposed three debug methods along with the SoC debug platform can verify silicon chips and FPGA prototypes to reduce the debug effort.

Fig.2 > SoC Debug Platform