行政院國家科學委員會專題研究計畫　期中進度報告

子計畫二:內容分類硬體加速器之 SoC 設計與製作(2/3)

計畫類別： 整合型計畫
計畫編號： NSC94-2213-E-009-036-
執行期間： 94 年 08 月 01 日至 95 年 07 月 31 日
執行單位： 國立交通大學電信工程學系(所)

計畫主持人： 李程輝

報告類型： 精簡報告
報告附件： 出席國際會議研究心得報告及發表論文
處理方式： 本計畫可公開查詢

中　華　民　國 95 年 5 月 30 日

一、摘要

ABSTRACT

The purpose of this project is to develop content inspection accelerator based on Xilinx FPGA platform. We have successfully integrated a parallel Bloom filter with the well-known Aho-Corasick (AC) pattern matching algorithm on the platform. To further improve system performance, we intend to generalize the AC algorithm so that multiple input symbols are processed in an operation cycle. Our goal is to achieve a throughput performance greater than 10Gbps. There are boundary conditions to be handled when multiple symbols are processed simultaneously. Currently we have derived the conditions for a pattern string to be matched. We will implement the generalized algorithm in the following year.

KEYWORDS: SOC, FPGA, String Matching, Network Security

中文摘要

隨著網路速度與技術不斷的提升，網路安全工具之工作量也越來越大。目前軟體的網路安全工具似乎已無法完全負擔沉重的工作量，尤其是網路內容過濾方面，包含病毒防護(Antivirus)、網頁內容過濾(Web/URL Content filter)和垃圾郵件過濾(Anti-Spam)等，這些均需要以字串比對(String Matching)的方式來達成；而字串比對(String Matching)正是最耗費計算資源的工作。本計畫之目的正是將字串比對(String Matching)的演算法實現於 FPGA，並配合嵌入式系統(Embedded System)以軟體與硬體互相整合以達到硬體加速軟體整合協調之目的。

關鍵字：網路安全、字串比對、嵌入式系統、FPGA

二、前言

網路技術日新月異，為工作、學術、生活等各方面帶來了莫大地幫助，但據統計，2000 年企業及政府部門遭受駭客或病毒攻擊的高達 85%，因此網路安全逐漸成為人們重視的問題。網際網路、企業網路等網路應用的頻寬需求急劇上升，傳輸、檢查、拆解、組合、搜尋、內容比對、轉遞等 IP 封包運算處理動作，以往可以靠軟體程式在一般微處理器上執行，搭配以網路卡做封包出入口。但是近年來這些封包的運算處理越來越複雜，將資料輸入處理單元，完後再將結果送往輸出單元，慢速處理造成的時間延遲會嚴重影響到資料吞吐量，無法滿足線速率的操作需求。一般所用的網路安全工具主要是防毒、防火牆等軟體，但是在網路傳輸速度不斷增長的現在，單純只靠軟體負責網路安全已不足夠勝任，若能將網路安全軟體中最耗費資源的部份轉移到硬體上來實現的話，必能大幅度地增加網路安全工具

的工作效率。

在大部分網路安全工具中，主要運算時間與資源是花費在字串比對(String Matching)上－將從網路上收到的資料與資料庫中(rule sets)的規則比對－因此要把此部分移至硬體上執行，以期達到加速的效果。而硬體開發最方便的方式就是採用 FPGA 的開發流程，而且若能配合現成的整合系統，就可解決軟體與硬體開發的困難處，並達成軟體硬體共同開發的效果。

三、設備器材

設備器材購買

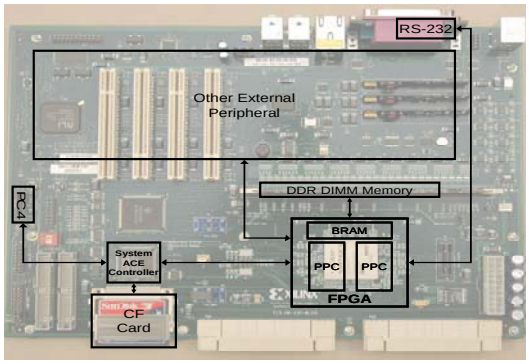
設備儀器名稱	數量
Xilinx Virtex-II Pro ML310	X 1

設備儀器簡介

簡介

1. **Xilinx Virtex-II Pro ML310：**

Xilinx Virtex-II Pro ML310(如下圖所示)擁有兩個 PowerPC 的處理器，,30816 個 Logic Cells 以及 2448kb 的 BRAM。晶片外還包含 DDR 記憶體,網路卡等 I/O 裝置及其介面以便連接,可使用提供之 IP 驅動這些週邊並可利用 PCI 匯流排外接其他裝置。板子中央銀白色的 FPGA 晶片，我們可以規劃此晶片以建立所需的硬體，2Mb 的 BRAM 作為目前儲存字串比對需要的資料,隨著比對字串的增加,2Mb 的 BRAM 不夠使用的時候，可以加入外部 DDR 記憶體。

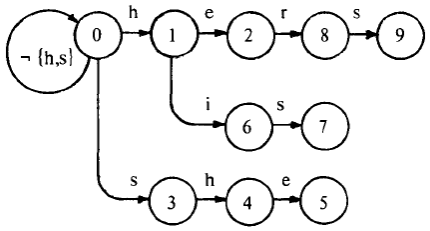


四、研究方法

字串比對－精確比對(Exact Matching)

不少 FPGA 與字串比對相關的論文，都是將網路安全的規則(rule)建立在 FPGA 的邏輯閘上，此舉雖然將最耗費時間的字串比對轉移到硬體上實現，提升了速度，使得整體的吞吐量(Throughput)能夠增加。但是 FPGA 上所能提供的邏輯閘數目、記憶體單元等硬體資源，總是有其限度的。侷限於此，在 FPGA 上能夠建立的規則(rule)數目實在是有限。所以在我們的設計上面，資源使用上的考量就更加注意。

在目前字串比對的許多演算法中，我們選擇 Aho-Corasick(AC)作為我們的基本做法。這是因為在 AC 的演算法中，可以保證快速而平穩的吞吐量，即使在最差的情況之下，也可以得到相當好的效能輸出，這樣可以避免攻擊者利用演算法的弱點去做攻擊。AC 是利用事先建立好一個有限狀態機(Finite State Machine)(如下圖所示)，在資料進來的時候，利用查表的方式作狀態轉換，在這狀態轉換過程中，部分狀態即代表有比對成功。因為使用查表的方式，所以可以保證穩定而快速的吞吐量，不過我們必須使用相當多的記憶體來提供查表的動作。

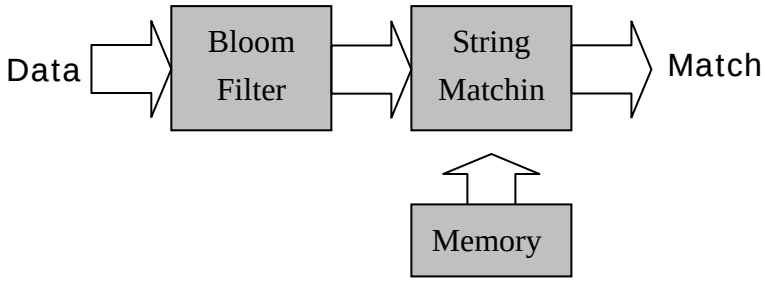


為了降低記憶體的使用，我們利用在 AC 的查表的資料是一個稀疏矩陣，使用 One-Band 的壓縮方式達到減少資料量的目的。我們原本表中每一列都會有 256 個資料，不過裡面會有許多資料都是指向初始狀態，所以我們將前後指向初始狀態的部分都不儲存，只儲存中間一段，這樣壓縮可以使資料量降低到原本的 1.45%。做法如下圖所示：

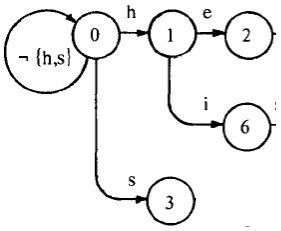
index	0	1	2	**3**	4	5	6	7	8	9	**10**	11	12	13	14	15	16	17	…	255
value	0	0	0	2	4	0	0	0	6	0	7	0	0	0	0	0	0	0	0	

BW	Start index	Band values							
8	3	2	4	0	0	0	6	0	7

為了提升字串比對速度，我們利用 Bloom-Filter 先去做過濾。資料進來的時候，先由 Bloom-Filter 去判斷哪一個位置可能會有字串出現，將不可能會有字串出現的部分跳過，有可能的部分在通知字串比對去做驗證的工作，如下圖所示。因為我們可以很快速的去過濾掉其他不可能的部分，所以在一般的封包傳送中，不會因為必須受到精確的字串比對而延遲，在通過 Bloom-Filter 之後，可以更快速的通過。在有攻擊出現時，我們的 AC 也可以保證穩定的吞吐量。



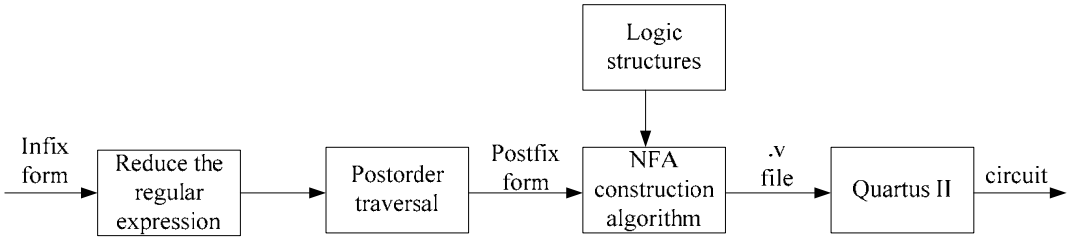
因為有先作過濾的動作，所以我們可以將我們的有限狀態機作簡化。在作驗證的時候，我們只要去驗證從這個位置開始是不是會有一個字串出現，不必考慮驗證過程中，如果失敗要再從哪一個狀態繼續做狀態轉換。在原本的 AC 的有許多狀態都是只有一條路徑可以走的，即代表只有一個字串有機會比對到，在這種情況之下，我們就只要將兩者的字元直接比較，而不用狀態轉換，以節省記憶體的使用，再搭配 One-Band 的壓縮，可以使資料量降低到原本的 0.31%。



字串比對－正規語言法比對(Regular Expression Matching)

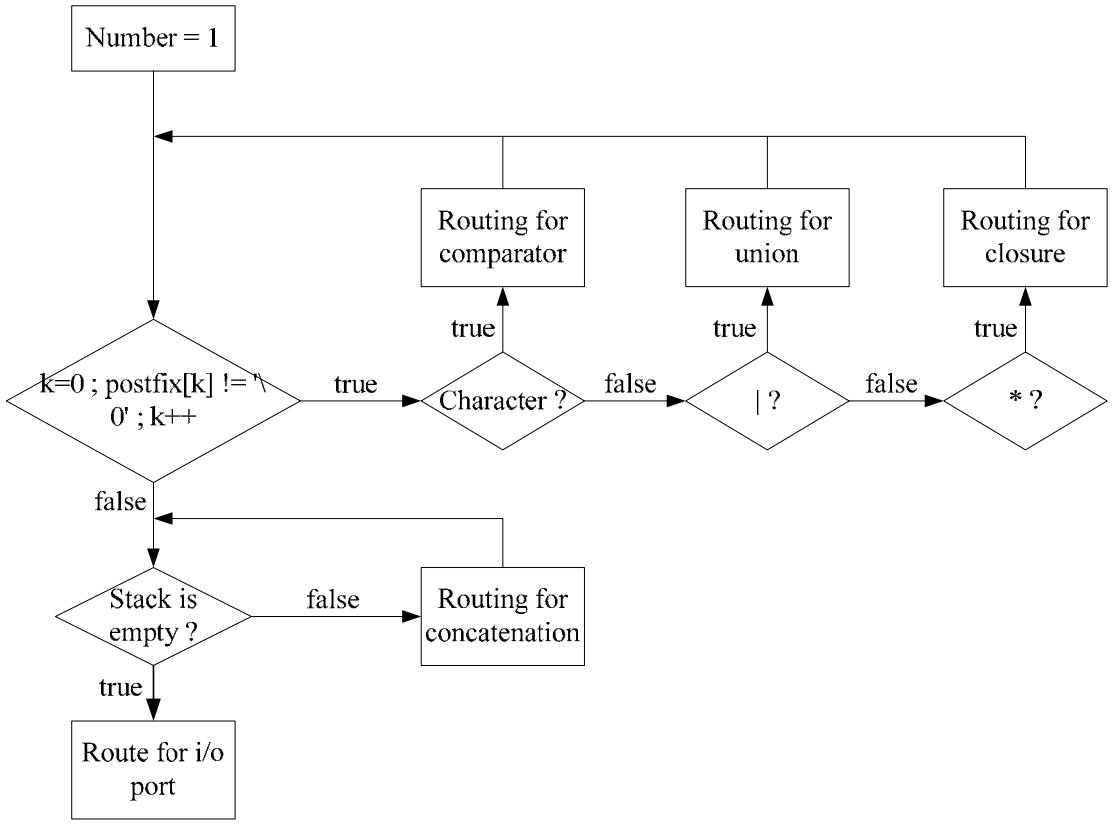
字串比對是一個很基本的問題，主要應用範圍包括文件搜尋、文法分析、DNA 序列搜尋…等。這些應用可以正規表示法(regular expression)當系統的輸入，系統再將正規表示法轉換成 DFA(deterministic finite automata)或 NFA(nondeterministic finite automata)。一般來說，使用軟體來實現通常都是將正規表示法轉成 DFA，例如 UNIX grep command、UNIX Lex(Lexical analyzer generator)和 Flex(Fast Lex) tools。使用硬體來實現通常都是將正規表示法轉成 NFA，我們提出方法屬於此類。

圖一是我們建構電路的流程圖。系統以中序式(infix form)的正規表示法為輸入，為了減低演算法的複雜度，先將中序式正規表示法轉換成後序式(postfix form)正規表示法。NFA 建造演算法(NFA construction algorithm)讀入後序式正規表示法和事先建立好的邏輯結構後，輸出 NFA 電路，此電路以 verilog 語言描述。以 verilog 語言描述的好處在於可適用各種型號的 FPGA，並且讓設計者有較好的彈性針對面積或是速度做最佳化(optimize)和調整電路。



圖一：建構電路流程圖

圖二是 NFA 建造演算法，演算法每次讀後序式正規表示法一個字元後，判斷正規表示法結束與否。若尚未結束則根據讀到的字元輸出相對應的電路(以 verilog 語言描述)，若已經讀完正規表示法則將上述獨立的電路區塊作連結，並且完成整個電路的輸入和輸出部分。



圖二：NFA 建造演算法

五、成果自評與未來展望

成果自評

在軟體方面：將修改 ClamAV，將其字串比對的部份分離出來，讓此部分交由硬體來負責。字串比對的方法部份，有精確比對(Exact Matching)與正規語言法比對(Regular Expression Matching)等兩種原型(Prototype)。在衡量過現有的 FPGA 規格後，利用 Verilog HDL 將這兩種原型(Prototype)實現在 FPGA 上，經過驗證之後，已確認 Function

是正確無誤的。目前在精確比對的部分中，搭配著 Bloom-Filter 以及前述的壓縮方式，將 ClamAV 中的 1000 個比對字串實作到板子上，其中字串比對需要的資料使用 1Mb 的 BRAM，在記憶體使用上面達到相當好的效能。

未來展望

接下來一年之中，在軟體方面：修改 ClamAV，並配合新演算法，讓軟硬體的效能更提升。在硬體方面，我們嘗試將 ClamAV 中的比對字串實作到板子上，因為數量的龐大，所以必須要使用到外部記憶體，接下來如何去有效的使用 BRAM 和外部記憶體將是將比對字串擴充的一項重點。還有就是建立 FPGA 與實驗母板兩者的通訊介面，使得軟體與硬體能夠溝通無礙，如何讓兩邊去分工合作而得到最佳的效能，達成兩者 Co-work 的目標。在演算法方面：目前在字串比對方面，在一個時脈內只會處理一個字元，將來以研究如何在有效的資源裡面，可以同時處理多個字元，以達到高速網路的要求。

六、參考文獻

[1] T. Kojm. ClamAV. www.clamav.net, 2004.

[2] Aho, A. V., and M. J. Corasick, “Efficient string matching: an aid to bibliographic search,” Communications of the ACM 18 (June 1975), pp. 333-340.

[3] Boyer R. S., and J. S. Moore, “A fast string searching algorithm,” Communications of the ACM 20 (October 1977), pp. 762-772.

[4] S. Wu and U. Manber. A fast algorithm for multi-pattern searching. Technical Report TR-94-17, University of Arizona, 1994.

[5] R. Sidhu and V. K. Prasanna, “Fast Regular Expression Matching using FPGAs,” in IEEE Sysposium on Field-Programmable Custom Computing Machines (FCCM), (Rohnert Park, CA, USA), Apr. 2001.

[6] Gokhale, M., Dubois, D., Dubois, A., Boorman, M., Poole, S., Hogsett, V.: Granidt: Towards gigabit rate network intrusion detection technology. In: Proceeding of 12th International Conference on Field Programmable Logic and Applications. (2002) France.

[7] Moscola, J., Lockwood, J., Loui, R. P., Pachos, M.: Implementation of a content-scanning module for an internet firewall. In: Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines. (2003) Napa, CA, USA.

[8] Young H. Cho, S.N., Mangione-Smith, W.: Specialized hardware for deep network packet filtering. In: Proceedings of 12th International Conference on Field Programmable Logic and Applications. (2002) France.

[9] F. Yu, R. H. Katz, and T. V. Laskhman, “Gigabit Rate Packet Pattern Matching with TCAM,” UCB technical report, UCB//CSD-04-1341, July 2004.

[10] I. Sourdis and D. Pnevmatikatos. Fast, Large-Scale String Match for a 10Gbps FPGA-Based

Network Intrusion Detection System. In Proceedings of FPL2003, 2003.
[11] Sarang Dharmapurikar, Praveen Krishnamurthy, Todd S. Sproull, John W. Lockwood: Deep Packet Inspection using Parallel Bloom Filters. IEEE Micro 24(1): 52-61 (2004).
[12] Nathan Tuck, Timothy Sherwood, Brad Calder, George Varghese: Deterministic Memory-Efficient String Matching Algorithms for Intrusion Detection. INFOCOM 2004.