於嵌入式網路開發平台實現具彈性之軟體運行環境

周厚沂[1] 蔡邦維[1] 羅孟彥[2] 楊竹星[1]
國立成功大學 電機工程學系暨電腦與通信工程研究所[1]
高雄應用科技大學 資訊工程系[2]
E-mail： csyang@ee.ncku.edu.tw[1]、myluo@cc.kuas.edu.tw[2]

摘要

在網路嵌入式元件設計平台之中，史丹佛大學所提出的NetFPGA開發平台能夠以FPGA元件實現自由程式化的硬體設計。提供網路開發人員更多的彈性去實現自訂的軟硬體功能。然而目前在 NetFPGA 上所開發的專案多數是基於史丹佛大學的參考設計架構，並沒有完整利用這張FPGA平台上的資源。由於嵌入式平台上的資源有限，在進行設計時須要考量運行效能與校調的彈性。本篇論文將提出一個新的軟硬體架構，充分利用 NetFPGA 開發板上FPGA內的PowerPC處理器，以及運行嵌入式的Linux作業系統，提供網路元件開發者一個更有彈性的開發環境，並讓 NetFPGA 上的硬體資源能更充分地被使用。

關鍵詞：NetFPGA、PowerPC處理器、嵌入式作業系統。

Abstract

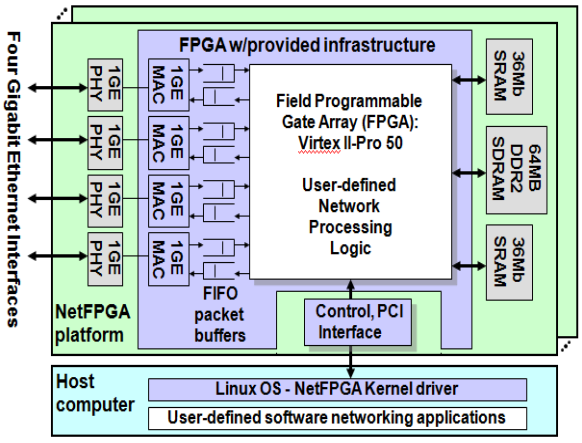
Among the embedded networking design platforms, NetFPGA, proposed by Stanford University, provides a freely programmable FPGA component to developers with a great flexibility to design custom hardware functionalities. However, most of the developed hardware projects are based on the Stanford reference design architecture, and they do not fully utilize the resources in the FPGA. On the other hand, due to the limited resources in embedded system, both performance and flexibility need to be concerned during design process. Therefore, in this paper, we propose a whole new architecture using the PowerPC processor in the FPGA on the NetFPGA board, and run an embedded Linux operating system as well. This not only provides developers a software execution environment with more flexibility, but also better utilizes the resources on the NetFPGA.

Keywords: NetFPGA, PowerPC processor, embedded operating system.

1. 前言

美國史丹佛大學的研究團隊於 2007 年提出了 NetFPGA 此一網路可程式化硬體開發平台[1]，上面搭載了賽靈思(Xilinx)公司生產的Virtex-II Pro 50 現場可程式化邏輯閘陣列晶片(Field-Programmable Gate Array，FPGA)，其內部包含了超過 50000 個邏輯單元(logic cell)供使用者實現自訂的硬體功能。除了具備可程式化晶片外，NetFPGA 開發版上還結合了豐富的周邊硬體資源，例如記憶體模組、PCI 介面、SATA 介面等。利用 NetFPGA，各種網路設備例如路由器(router)、交換器(switch)和網路介面卡(network interface card, NIC)等等得以實現，讓學生以及開發人員有一個良好的平台進行研究。截至 2011 年，世界各地已經有超過 150 個研究單位利用此平台開發各種具備網路功能之創新硬體。

在上述所提的 NetFPGA 開發板中，事實上還內嵌了兩顆 IBM PowerPC 405 嵌入式處理器。然而，目前已開發的專案主要都是基於史丹佛大學所提的參考設計架構[2]，如圖一所示，僅使用了部分硬體資源，而沒有採用其中任何一顆FPGA中的嵌入式處理器做開發。為了讓PowerPC處理器資源能被利用，本論文中的設計大幅修改了原先史丹佛的架構。但由於NetFPGA官方網站並沒有相關資源，硬體的供應商也沒有提供結合該處理器的系統配置檔案。因此本篇論文將根據 NetFPGA 的電路設計圖[3]，手動配置相關的硬體驅動控制器，再結合自行設計的硬體模組，來完成新的硬體架構。最後搭載嵌入式作業系統，以提供更具彈性的環境供網路開發者運行軟體並做測試。



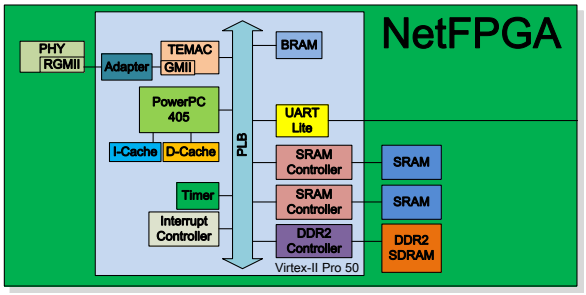
圖一、史丹佛大學提出之參考設計架構

2.硬體架構設計

新的硬體架構如圖二所示。其中包含了一個 PowerPC 405 處理器，具備指令快取(i-cache)與資料快取(d-cache)，運作時脈為 125MHz，以及靜態隨機存取記憶體控制器(SRAM controller)、動態隨機存取記憶體控制器(DRAM controller)、簡化的通用非同步收發傳輸器(UART Lite controller)、三態乙太網路媒體存取控制器(Tri-mode Ethernet Media Access Controller，TEMAC)、計時器、中斷控制器、FPGA 晶片內的區塊隨機存取記憶體(BRAM)與控制器。周邊硬體控制器是利用 Xilinx 公司所提供的整合開發工具 ISE Design Suite 10.1 之中的嵌入式設計工具(Embedded Design Kits，EDK)，配合電路板設計圖，輸入相關參數來產生。這部分需要考慮到 FPGA 晶片本身的支援性、周邊硬體的型號、與 FPGA 連結的針腳以及訊號定義，再交由 EDK 工具做電路合成。而為了簡化設計，所有的控制器與處理器均連結到單一的處理器匯流排(Processor Local Bus，PLB)。

2.1 UART Lite

架構中的 UART Lite controller 是為了方便除錯與驗證。UART Lite 為 UART 簡化版本，僅採用了 3 條訊號 TXD、RXD 及 GND，如表 1 所示，可減少 FPGA 邏輯單元的耗費，讓其他硬體元件有更多可用資源。其中，TXD 與 RXD 分別作為資料傳輸與接收，電路合成時會分配到 NetFPGA 開發板右上方的兩隻 Debug 針腳，再利用杜邦線接至電腦主機的 RS232 界面，即可透過終端機顯示及接收訊息，連接方式如圖三所示。



圖二、運用 PowerPC 處理器之硬體架構

2.2 SRAM / DRAM controller

記憶體控制器方面，SRAM 控制器需要根據外部晶片做參數設定，包括資料的寬度以及地址線的寬度，方可順利運作。但在 DRAM 控制器方面則需要下更多功夫。原因在於 NetFPGA 採用具備第二代雙倍資料率(DDR2)的同步動態隨機存取記憶體(SDRAM)，此技術在同一時脈的上升緣(rising edge)與下降緣(falling edge)都加以利用，進行資料傳輸，比起上一代可達到更快的速率，但這也意味著需要對時脈及其他控制訊號作更精確的控制。NetFPGA 所提供的開發原始碼中包含了利用 Xilinx Memory Interface Generator (MIG)工具所產生的控制器，其中便針對了控制訊號進行嚴格的時序與布線限制，然而該控制器並不具備與匯流排連結的介面，因此無法直接移植至本論文的架構設計中。

針對此一問題，本篇論文解決的方法是採用 EDK 生成的多埠口記憶體控制器(Multi-port Memory Controller，MPMC)，此元件具備與 PLB 匯流排連結的介面並且提供了 Static PHY 功能，能夠利用 Xilinx FPGA 中的數位時脈管理器(Digital Clock Manager，DCM)來自動對訊號進行校正，此動作只需在 NetFPGA 上電後進行一次，即可順利存取 DRAM。而校正程式則是透過 JTAG 於下載位元檔(bitfile)至 FPGA 時一併載入至晶片中的 BRAM(見圖二)並由處理器執行。

2.3 TEMAC

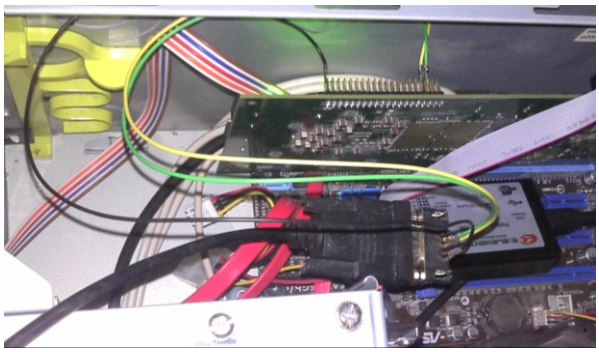
而關於網路部分，連結媒體存取控制層(MAC)與物理層(PHY)的介面可分為 MII、RMII、GMII、RGMII 等[4、5]。MII 與 RMII 介面用於 10/100 Mbps 乙太網路，而 GMII、RGMII 限於 Gigabit 乙太網路，又 RMII 與 RGMII 是為了簡化電路配置(layout)所提出之

精簡訊號線的介面。然而由 EDK 所產生的乙太網路控制器(TEMAC)，在 Virtex-II Pro 50 FPGA 上僅能採用 GMII 介面，但外部所連結的 Broadcom PHY controller[6]卻是採用 RGMII 的介面，為此本論文提出了一個硬體適配器模組(Adapter)的設計，以實現 GMII 與 RGMII 介面之間的訊號轉換。由於 NetFPGA 採用 Gigabit 級的 RMII 介面，運作時脈達 125MHz，因此適配器模組同樣必須對控制訊號做時序限制。這部分須利用 Xilinx 硬體函式庫所提供的 BUFG，將訊號利用 FPGA 內部的全局布線資源做布線，來確保訊號品質，以防延遲、擾動之影響而無法順利收送乙太網路訊框(frame)。

值得注意的是 BUFG 所衍生的多工器 BUFGMUX，在 Virtex-II Pro 中屬於稀有的資源，總數僅有 16 個，使用超過半數的 BUFGMUX 時 Xilinx ISE 10.1 版本的布局繞線工具(PAR)即會產生警告訊息，可能無法順利產生 bitfile。解決方案盡可能縮減 BUFG 的使用並手動進行精確的布局配置，不能純交由工具自動分配。故設計中參考了 Virtex-II Pro TEMAC 的技術文件[7]以及 NetFPGA 原始碼中的針對布局繞線作限制的.ucf 檔案，綜合了這些限制參數最終完成了適配器模組。

2.4 硬體資源總結

本論文中提出的新架構採用了 7000 多個邏輯單元、1 個 PowerPC 處理器、11 個 BUFGMUX 和 4 個 DCM 等，尚有 1 個 PowerPC 處理器及 40000 多個邏輯單元可用。未來將可用以實現雙核心處理器架構以及更多匯流排周邊硬體模組，如多埠口 TEMAC、PCI 橋接器等。



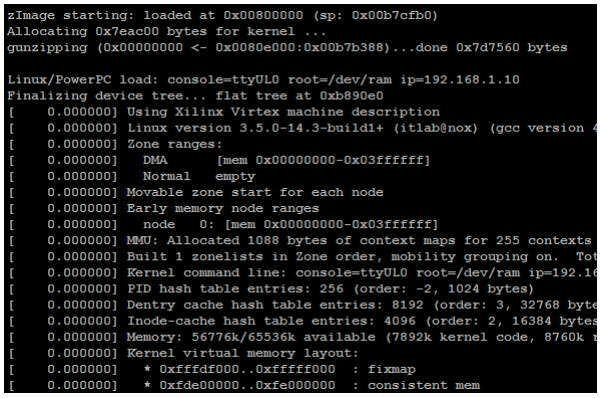
圖三、利用 NetFPGA 上的 Debug pins 連結電腦主機之 RS232 介面

表 1 UART 與 UART Lite 採用訊號之比較

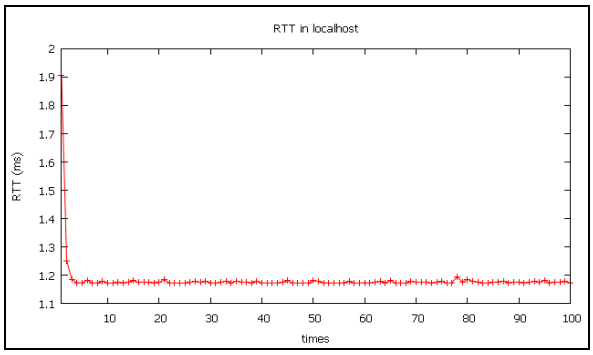
訊號	意義	UART	UART Lite
CD	載波偵測	V	
RXD	接收資料	V	V
TXD	傳送資料	V	V
DTR	資料就緒	V	
GND	接地	V	V
DSR	接收裝置就緒	V	
RTS	傳送請求	V	
CTS	接收就緒	V	
RI	鈴響指示	V	

3. 軟體環境設計

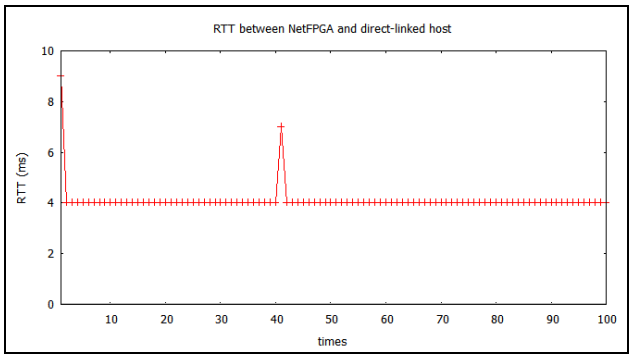
完成了具備處理器以及記憶體空間的硬體架構後，下一步即是搭載嵌入式作業系統，以作為軟體運行及開發之平台。本論文中移植的作業系統核心是採用是 Xilinx 公司所修改之 Linux 3.0 的版本[8]，此核心加入了 Xilinx 公司的多項硬體裝置驅動程式，包含 UART Lite、LL TEMAC 等。我們根據設計需求對核心進行了組態設定，並盡量縮減體積。檔案系統方面，則是利用 Buildroot [9]以及 BusyBox [10]來配置必要的目錄以及系統工具，包含網路工具 ifconfig、ping 和 tftp 等。完成核心與檔案系統的組態設定後，接著藉由交叉工具鏈(cross-toolchain)編譯、連結並壓縮，產生目標機器(PowerPC)的可連結與執行檔(Executable and Linkable file，elf)，而後再將.elf 檔案利用 GDB 工具透過 JTAG 傳輸至 NetFPGA 上的 Virtex-II Pro FPGA，就能經由內部 DRAM 控制器寫入 DRAM 記憶體空間，再於 GDB server 下達指令即可將核心解壓縮並運行。順利運行的系統畫面將呈現於終端機，如圖四所示。



圖四、核心啟動畫面



圖五、本機之封包傳輸 RTT



圖六、NetFPGA 與單一主機之間的 RTT

4. 實驗與測試結果

在測試部分，本章節將先介紹 NetFPGA 本機處理網路封包之效能量測結果，並透過網路線連結另一部主機，測試其網路傳輸的能力，最後則是記憶體傳輸效能的測試。

4.1 本機效能測試

NetFPGA 本機網路封包的測試如圖五所示，為 NetFPGA 載入包含網路工具之 Linux 作業系統後，利用 ping 指令對本機(127.0.0.1)送出 ICMP 請求 100 次後量測其回復時間之結果。對於本機的 ICMP 請求實際上不會透過網路介面發出，而是直接交由處理器處理並回覆，因此可做為衡量本機系統效能之參考指標。結果顯示出平均 RTT 約為 1ms，共計處理了 100 個封包，並無丟棄封包之行為。

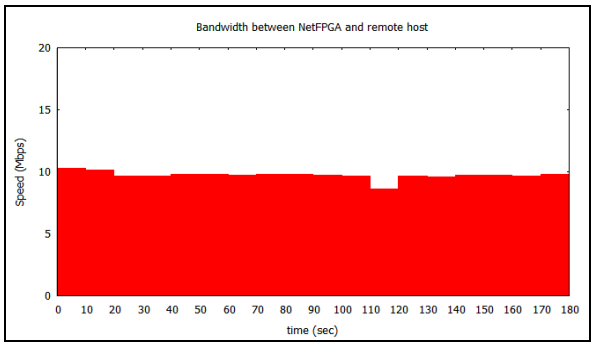
4.2 對外傳輸效能測試

第二部分將使用 CAT5e 等級以上的雙絞線(因為架構中的 TEMAC 控制器僅使用 1Gbps 傳輸速率，並不支援 CAT5 等級的 10/100 Mbps 速率)，聯結具備新架構的 NetFPGA 與另一台主機的網路孔介面，再利用 ping 工具測試了 RTT 時間與封包的丟失率，測試結果如圖六。圖六同樣為 NetFPGA 載入 Linux 作業系統之後，連結的主機利用 ping 工具對其發出 100 次的 ICMP 請求，再量測每次的回復時間所呈現之結果。圖中顯示出除了第 1 次與第 41 次的回應時間較長外，共計 100 次的請求均收到了回覆，封包遺失率為 0%，平均 RTT 約來到 4 毫秒。此部分證明了新架構之中的 TEMAC 控制器與本篇論文提出之 Adapter 模組之結合運作正確無誤，能順利傳送與接收乙太網路訊框(frame)，而不會有遺失之情況。

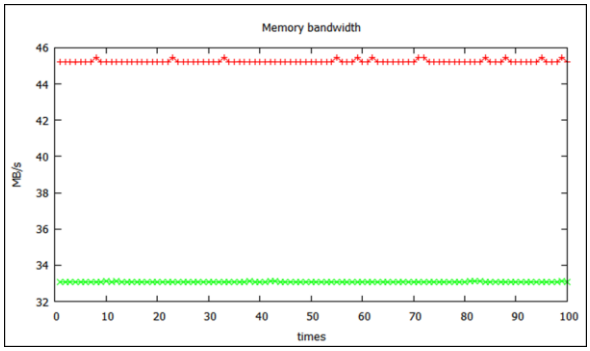
另一個測試則是利用 iperf 工具，此工具先透過交叉工具鏈編譯成 PowerPC 架構之版本，再透過 tftp 傳輸至 NetFPGA。NetFPGA 運行 iperf 伺服器端，連結主機則作為 iperf 客戶端，兩者進行資料傳輸，量測實際傳輸頻寬(throughput)，結果如圖七。測試的時間為 3 分鐘，以每 10 秒為間隔作記錄取平均。圖中顯示出了 NetFPGA 系統與連結主機之間所實際量測出之頻寬，約為理論頻寬的百分之一。此結果顯示新系統的效能有值得改善之處，推測效能的瓶頸在於處理器本身、PLB 匯流排以及 DRAM 傳輸。處理器運行在 125MHz 的時脈下，對於封包的處理緩慢，而處理完成後又須透過 PLB 匯流排傳輸至 DRAM 控制器，再到晶片外的記憶體模組進行資料存取，而後才是由 TEMAC 與記憶體之間的 DMA 傳輸，以上種種限制了對外傳輸的實際頻寬，此部分應當是未來改善的方向。

4.3 記憶體傳輸效能測試

為了研究系統效能的瓶頸，在實驗中最後量測了記憶體傳輸的頻寬。此部分利用著名的 benchmark 工具 LMbench[11] 進行測試。LMbench 中包含了各種效能測試工具，對於記憶體部分，本項實驗採用了 bw_mem_rd 以及 bw_mem_wr 兩種工具做為測試，分別量測記憶體讀取以及寫入的速率。LMbench 同樣利用交叉工具鏈作編譯，並以 tftp 傳輸至 NetFPGA。量測的結果如圖八所示，顯示了運行 LMbench 工具 100 次的結果，紅線代表記憶體讀取速率，約為 45MB/s，綠線代表記憶體寫入速率，約來到 33MB/s。證明了記憶體傳輸的效率不佳，這點會是未來需要亟待改進的部分。



圖七、NetFPGA 與單一主機之間的傳輸頻寬



圖八、NetFPGA 記憶體傳輸頻寬

5. 結論

本篇論文捨棄了史丹佛大學的原始設計，在 NetFPGA 上提出了一個新的軟硬體架構，運用到了 Virtex-II Pro 50 FPGA 內部的 PowerPC 處理器資源，並且結合各周邊控制器連結 PLB 匯流排，以及自行設計的硬體模組，協同運作。最終在上面運行了嵌入式作業系統，並做了實驗與測試，驗證了系統運作的正確性，可提供軟體開發者一個具彈性化的軟體運行環境。

6. 未來工作

在效能測試中顯示了記憶體讀寫速度緩慢，此部分的調整是未來工作的重點，雖然原先的構想是從彈性出發，讓不善於硬體設計的軟體開發者也能運用 NetFPGA 的資源進行網路設備的相關設計，但效能上作了過大的犧牲。最需要改善的地方在於 TEMAC 控制器的效能及記憶體的存取速率，希望能使網路傳輸達到 1 Gbps 的線速(line-speed)運作等級。另外，連結主機作通訊的 PCI 橋接器、雙核心處理器的設計、多埠口網路介面等也是未來硬體修改的方向。希望可藉此提升 NetFPGA 平台新架構的硬體效能，讓此開發平台具備更高的實用性。

致謝

本研究承蒙行政院國家科學委員會計畫 101-2219-E-006-004 及 100-2218-E-151-003-MY3 經費部分補助，特此感謝。

參考文獻

[1] NetFPGA - An Open Platform for Gigabit-rate Network Switching and Routing; John W. Lockwood, Nick McKeown, Greg Watson, Glen Gibb, Paul Hartke, Jad Naous, Ramanan Raghuraman, and Jianying Luo; MSE 2007, San Diego, June 2007.

[2] NetFPGA: Reusable Router Architecture for Experimental Research; Jad Naous, Glen Gibb, Sara Bolouki, and Nick McKeown; SIGCOMM PRESTO Workshop, Seattle, WA, August 2008.

[3] NetFPGA PCB Layout，http://netfpga.org/NetFPGA_PCB_r3.pdf

[4] IEEE Standard 802.3: CSMA/CD Access Method and Physical Layer Specifications, Section Two，http://standards.ieee.org/getieee802/download/802.3-2008_section2.pdf

[5] RGMII v2.0 spec，http://www.hp.com/rnd/pdfs/RGMIIv2_0_final_hp.pdf

[6] Broadcom BCM5464SR，http://zh-tw.broadcom.com/products/Physical-Layer/Gigabit-Ethernet-PHYs/BCM5464SR

[7] Xilinx XAPP 692 ，http://www.xilinx.com/support/documentation/application_notes/xapp692.pdf

[8] Linux kernel for Xilinx，git://git.xilinx.com/linux-xlnx.git

[9] Buildroot，http://buildroot.uclibc.org/

[10] BusyBox，http://www.busybox.net/

[11] LMbench，http://www.bitmover.com/lmbench/