

# THỰC HIỆN LỌC SỐ THỜI GIAN THỰC DÙNG FPGA GIAO TIẾP VỚI MÁY TÍNH QUA BUS PCI

Huỳnh Hữu Thuận, Nguyễn Hữu Phương

Khoa Vật lý, Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

(Bài nhận ngày 17 tháng 8 năm 2005, hoàn chỉnh sửa chữa ngày 29 tháng 11 năm 2005)

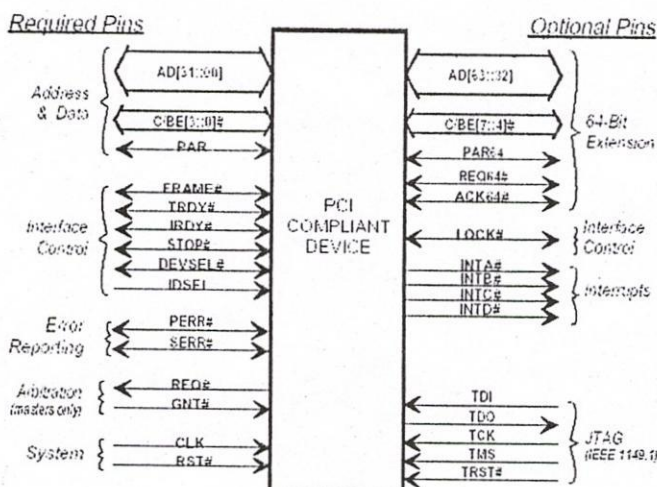
**TÓM TẮT:** Để xử lý thời gian thực tín hiệu (âm thanh, hình ảnh ...) trên máy tính ta phải kết hợp nhiều yếu tố như phải thêm mạch xử lý (tiền xử lý) ở bên ngoài, nhập dữ liệu vào máy tính qua Bus tốc độ cao (USB và nhất là PCI), chọn ngôn ngữ và kỹ thuật lập trình phù hợp, dùng xử lý song song (máy đa bộ xử lý)... Trong bài báo này, chúng tôi trình bày hai khâu: (1) Lọc số thời gian thực có hệ số lọc thay đổi tùy ý thực hiện trên FPGA, thay vì dùng bộ DSP như mạch tiền xử lý, (2) tạo giao tiếp qua Bus PCI giữa bo mạch và máy tính. Chúng tôi đã thu được các kết quả tốt.

## 1. GIỚI THIỆU

Để xử lý các tín hiệu của thế giới bên ngoài máy tính như tiếng nói, hình ảnh, đo lường ... thường ta nhập chúng vào máy tính thông qua một bo mạch A/D hoặc DAQ. Các bo mạch này có tốc độ từ thấp đến rất cao được lựa chọn phù hợp với đặc tính tần số của tín hiệu. Tuy nhiên khi xử lý thời gian thực các tín hiệu ta phải kết hợp thêm nhiều biện pháp khác [1], trong đó có hai vấn đề chúng tôi trình bày trong bài báo này:

- Trong những năm gần đây, người ta có khuynh hướng dùng các linh kiện (thiết bị) khả trình CPLD và nhất là FPGA [2], [3] cho các xử lý chính bên ngoài thay vì dùng các bộ DSP hay ASIC, để được chủ động, linh hoạt hơn. Cho mục đích này, chúng tôi đã cài đặt lọc số FIR, IIR thời gian thực có các hệ số thay đổi được tùy ý trên Kit FPGA Altera Stratix EP1S10 (hoặc EP1S25) [4]. Chọn lọc số là để minh họa và cho nhu cầu sử dụng ở phòng thí nghiệm của chúng tôi. Việc thêm các mạch chuyển đổi A/D, D/A hay cài một giải thuật xử lý nào khác (cho âm thanh, hình ảnh...) chủ yếu là do nhu cầu.

- Mạch ta tự thiết kế và lắp ráp và nhiều Kit hoặc bo phát triển (Development Board) mua không có sẵn giao tiếp Bus PCI [5], [6] rất cần thiết cho việc trao đổi dữ liệu tốc độ cao (kể cả cơ chế DMA). Trường hợp Kit Stratix nêu trước cũng vậy. Do đó chúng tôi phải thực hiện truyền thông này, đồng thời cũng thảo luận một số vấn đề về PCI (rất ít được đề cập đến trên các tạp chí KH-CN trong nước).

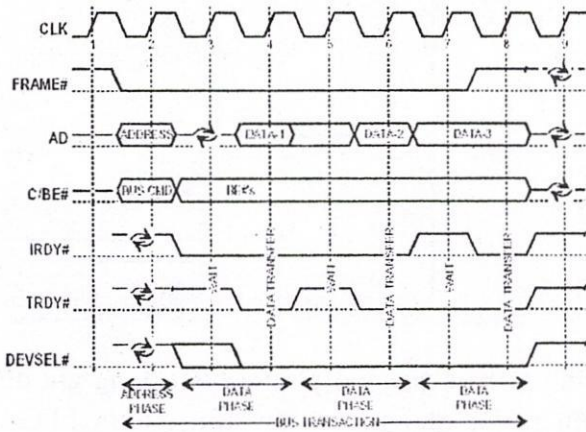


Hình 1: Các tín hiệu PCI trình bày theo nhóm chức năng

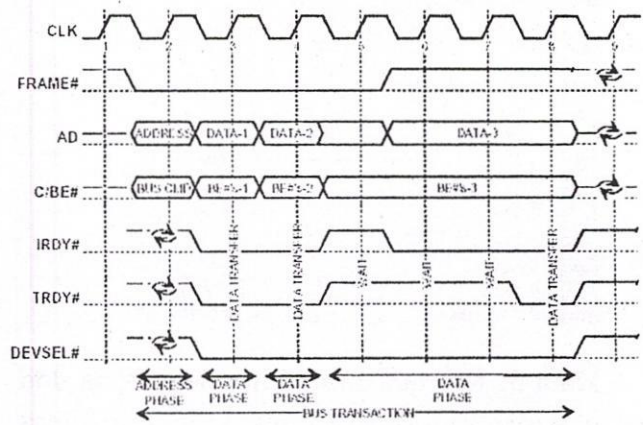
## 2. CƠ SỞ THIẾT KẾ

### 2.1 Thiết kế Card PCI

Giao tiếp PCI đòi hỏi tối thiểu 47 chân cho các thiết bị Target (hay Slave – là các thiết bị được đánh địa chỉ bởi bộ khởi tạo quá trình truyền dữ liệu, lúc đó bộ khởi tạo quá trình truyền dữ liệu gọi là Master) và 47 chân cho các thiết bị Master để điều khiển quá trình truyền dữ liệu, điều khiển việc giao tiếp và các chức năng hệ thống khác. Các chân tín hiệu PCI được trình bày theo từng nhóm trong Hình 1. Quá trình đọc, ghi dữ liệu trên Bus PCI được trình bày trong Hình 2 và Hình 3 tương ứng.

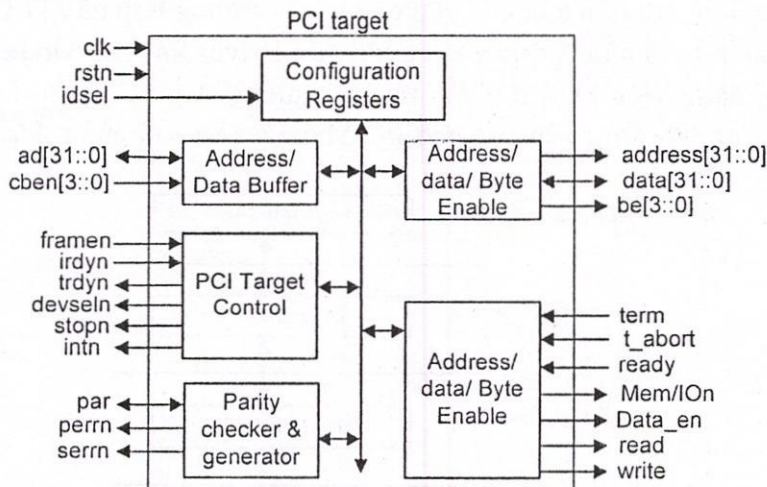


Hình 2: Hoạt động đọc trên Bus PCI.



Hình 3: Hoạt động ghi trên Bus PCI.

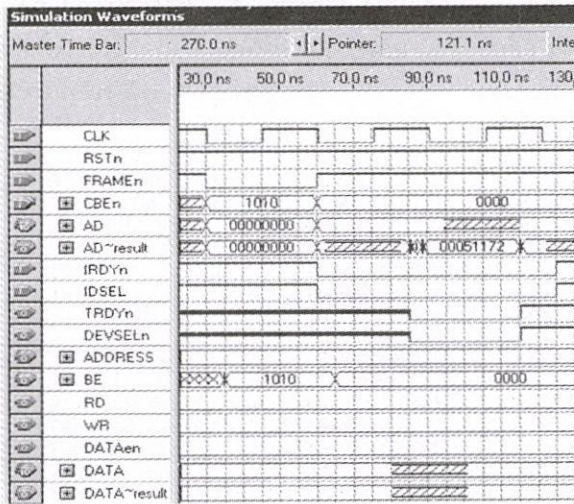
Chúng tôi đã phát triển một lõi giao tiếp với Bus PCI, lõi được viết bằng ngôn ngữ mô tả phần cứng và có thể được nhúng vào bất kỳ FPGA nào. Cấu trúc của lõi này được trình bày trong Hình 4.



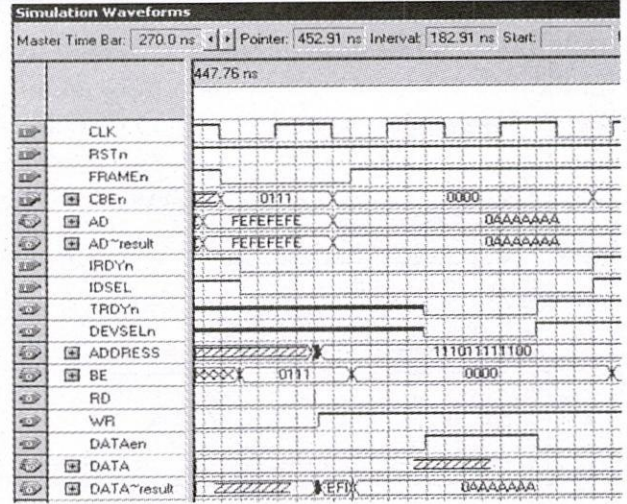
Hình 4: Cấu trúc của một PCI Target.

Các thanh ghi cấu hình cho phép phần mềm dò tìm, khởi tạo và cấu hình thiết bị PCI. Ngoài ra, các thanh ghi trong không gian cấu hình này còn hỗ trợ các chức năng điều khiển lõi. Khối PCI Target Control điều khiển toàn bộ hoạt động của hệ thống gồm: Đáp ứng với các lệnh trên Bus PCI (thông qua các tín hiệu cben[3:0]) và thực hiện việc tạo ra các tín hiệu bắt tay tương ứng theo quy định của chuẩn PCI [5]. Ngoài ra, khối này còn tạo ra các Logic bắt tay với phần ứng dụng trong khối Address/ Data/ Byte Enable. Khối Parity Checker & Generator thực hiện quá trình kiểm tra/ tạo Pariry, tạo ra tín hiệu báo lỗi khi

phát hiện sai. Lỗi được mô phỏng trên FPGA Stratix EP1S10F672C6ES với tài nguyên sử dụng: 450/10.570 Logic Element, 0/920.448 memory bits, 0/48 DSP Block 9-bit. Các kết quả mô phỏng cho hoạt động đọc cấu hình (trong tình huống này là đọc Vendor ID và Device ID) được trình bày ở Hình 5 và ghi dữ liệu ra bộ nhớ ở Hình 6. Hoạt động đọc/ ghi này có thể so sánh với chuẩn PCI ở Hình 2 và Hình 3.

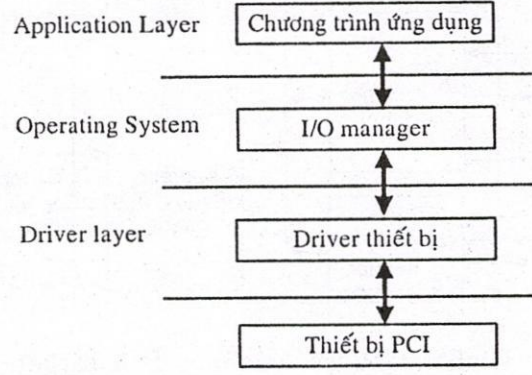


Hình 5: Kết quả mô phỏng hoạt động đọc cấu hình trên Bus PCI thực hiện trên Altera FPGA EP1S10F672C6ES.



Hình 6: Kết quả mô phỏng hoạt động ghi dữ liệu ra bộ nhớ thực hiện trên Altera FPGA EP1S10F672C6ES.

Để sử dụng thiết kế trên hệ điều hành Windows 2000 (hoặc Windows XP) thì Driver cho Card cũng phải được phát triển. Chức năng của Driver trong các ứng dụng được trình bày ở Hình 7. I/O Manager nhận các thông điệp từ các chương trình ứng dụng (ở Application Layer) ở dạng các mã điều khiển (có thể kèm theo dữ liệu tham số hoặc không) và nó xác định nơi đến của các yêu cầu (trong trường hợp này là Driver của thiết bị PCI), kiểm tra tính hợp lệ của chúng và chuyển đến Driver thiết bị. Driver thiết bị đáp ứng với mỗi yêu cầu bằng việc khởi tạo các tín hiệu tương ứng trên giao tiếp PCI và trả về trạng thái tương ứng với từng yêu cầu cho I/O Manager và sau cùng đưa đến chương trình ứng dụng.



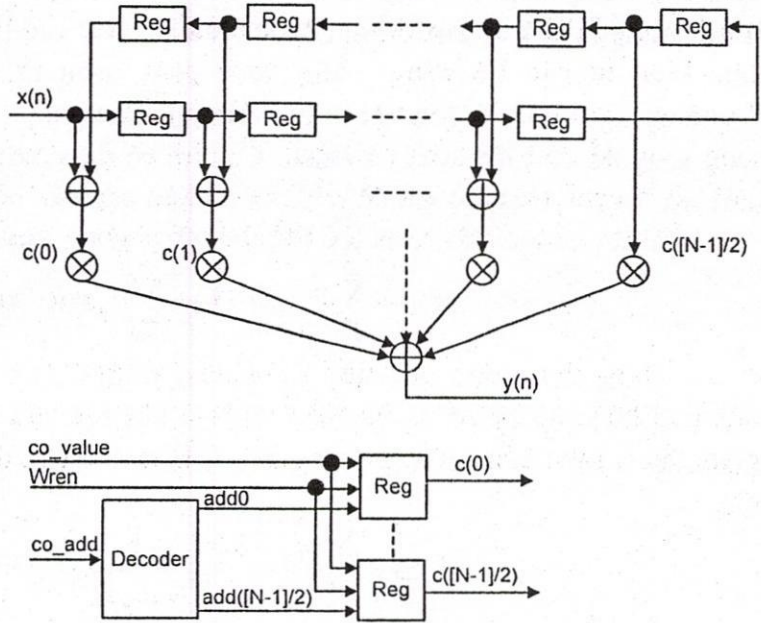
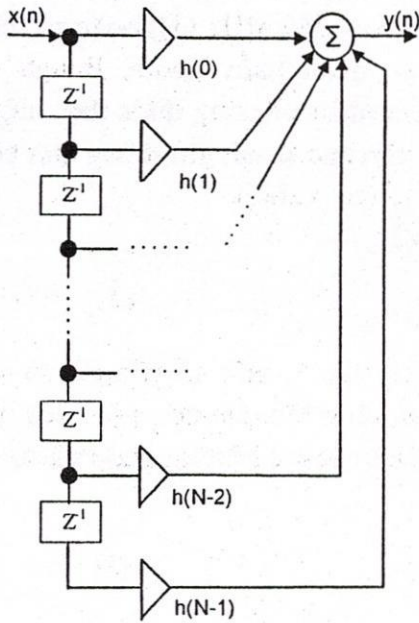
Hình 7: Chức năng của Driver [7].

2.2. Thiết kế lọc số có hệ số thay đổi

Thiết kế Lọc FIR: Phương trình của lọc FIR tuyến tính, bất biến thời gian và nhân quả cho bởi (1):

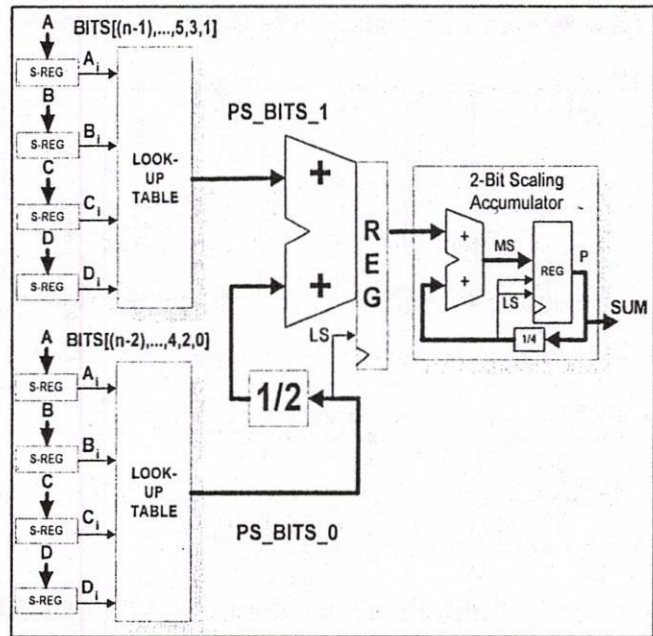
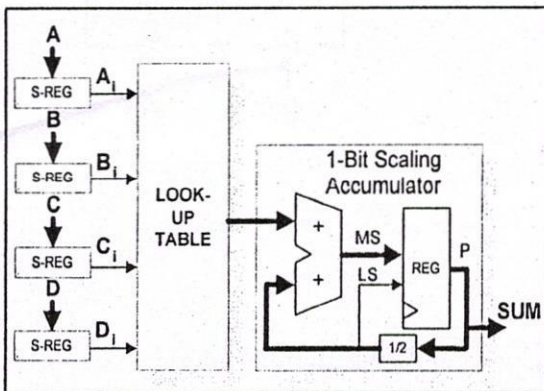
$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{1}$$

và có cấu tạo như Hình 8.



Hình 8: Cấu tạo của một lọc FIR N hệ số.

Hình 9: Thực hiện lọc FIR thời gian thực với N hệ số lẻ, đối xứng và thay đổi được.



Hình 10: Thuật toán SDA dựa trên LUT cho MAC gồm bốn tích số [8].

Hình 11: Thuật toán PDA hai bit dựa trên LUT cho MAC gồm bốn tích số [8].

Ưu điểm của lọc FIR là sự ổn định không điều kiện và đáp ứng pha tuyến tính nhưng bất lợi rất lớn là chiếm nhiều tài nguyên FPGA nên một số kỹ thuật tính toán phân bố đã được nghiên cứu nhằm giảm số tài nguyên tối thiểu tùy theo hiệu suất mong muốn [8], đáng kể nhất là hai thuật toán SDA (Serial Distributed Arithmetic) được trình bày trong Hình 10 và thuật toán PDA (Parallel Distributed Arithmetic) ở Hình 11. Tuy nhiên, các thuật toán này phụ thuộc nhiều vào kiến trúc đặc thù của FPGA, hơn nữa, việc thực hiện các phép toán dựa vào các LUT làm cho thuật toán này không phù hợp với hệ thống có các

hệ số lọc thay đổi khi đang hoạt động. Thuật toán được sử dụng trong bài báo này được trình bày trong Hình 9, trong đó các bộ nhân và cộng có thể được thực hiện bằng các DSP Block trong FPGA Stratix để đạt được hiệu suất rất cao (khoảng 250 MHz [3]) hoặc có thể thực hiện từ các bộ cộng, nhân được phát triển từ các thuật toán Booth, Baugh – Woodley... với tốc độ đến vài trăm MHz [9]. Các phép toán nhân và cộng được thực hiện song song để có hiệu suất cao nhất. Các hệ số được nạp vào các thanh ghi thông qua bộ giải mã địa chỉ, tín hiệu ghi (Wren), và dữ liệu nạp vào hệ số (co\_value).

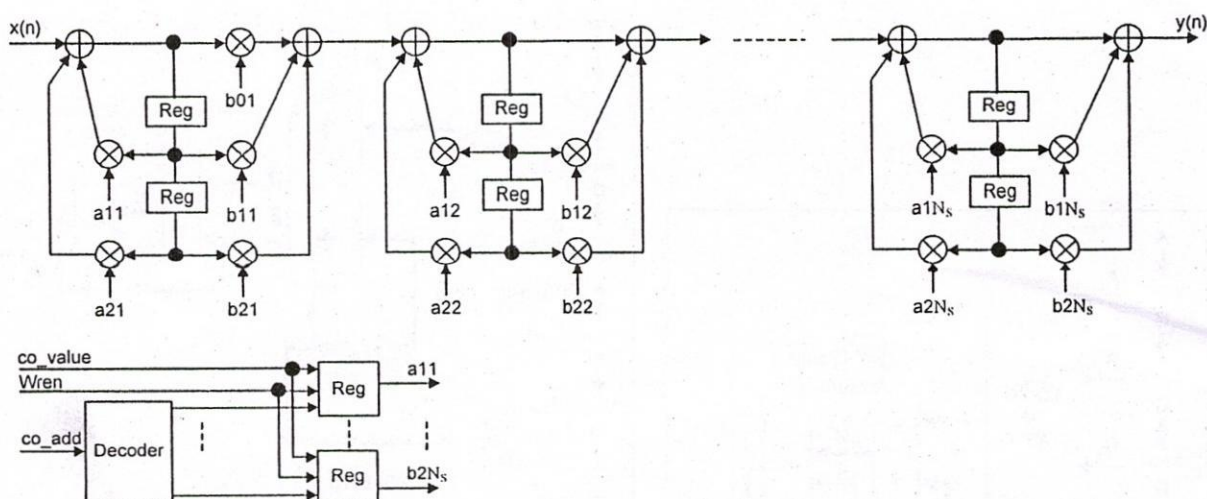
Thiết kế lọc IIR: Xem lọc IIR cho bởi phương trình (2):

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{m=0}^M b_m x(n-m) \quad (2)$$

Dạng thực hiện trực tiếp từ phương trình (2) có thể gây ra một số vấn đề do sự lượng tử hóa các hệ số và tín hiệu, nhất là khi bậc của lọc tăng lên. Do đó, một cách để giảm thiểu ảnh hưởng của sự lượng tử hóa là tách lọc ra thành các lọc bậc hai ghép nối tiếp (3):

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \quad (3)$$

Dạng thực hiện của phương trình này được trình bày trong Hình 12. Lọc bậc hai đầu tiên có chứa số hạng  $b_{01}$  khác 1 do đó hệ số nhân này được thực hiện song song với các phép nhân tương ứng với các hệ số  $b_{11}$ ,  $b_{21}$ . Từ lọc bậc hai thứ hai trở đi, hệ số nhân  $b_{0k} = 1$  nên phép nhân này được loại bỏ. Các hệ số lọc được nạp thông qua bộ giải mã địa chỉ, tín hiệu Wren và co\_value chứa trị giá cần nạp cho hệ số tương ứng, như đã nói ở trước.



Hình 12: Thực hiện lọc IIR thời gian thực với các hệ số thay đổi được.

### 3. THỰC NGHIỆM VÀ KẾT QUẢ

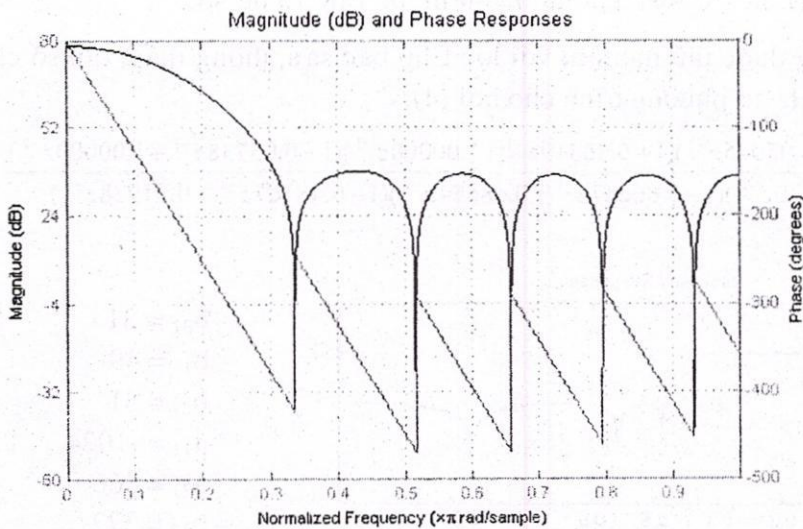
#### 3.1 Thực hiện Card PCI

Có nhiều giải pháp để thực hiện giao tiếp FPGA với PCI Bus như dùng một số vi mạch giao tiếp ví dụ PLX PCI 9052, 9054 . . . hoặc dùng các Card trên thị trường. Chúng tôi cũng đã sử dụng thành công Card PCI-DIO24 của hãng Measurement Computing (Mỹ) mà không dùng Driver và thư viện lập trình kèm theo của Kit (vì được bán riêng). Để có hiệu suất cao hơn và linh hoạt hơn, một giải pháp khác là sử dụng FPGA với lõi giao tiếp PCI như đã mô tả ở mục 2.1 hoặc các lõi PCI phát triển bởi các hãng sản xuất FPGA (Xilinx, Altera[11], Quicklogic...) rồi sau đó viết Driver. Chúng tôi cũng đã thực hiện

thành công thiết kế trên PCI Development Board của hãng Memec [10] hoạt động ở tần số 33MHz, 32 bit. Driver sử dụng cho bo mạch được phát triển bằng Windows 2000 DDK (Driver Development Kit) và Visual C 6.0 của hãng Microsoft. Bên cạnh đó, chúng tôi còn thử nghiệm Driver tạo ra bởi phần mềm Windriver và kết quả hoạt động tốt trên cả hai hệ điều hành Windows 2000 và Windows XP.

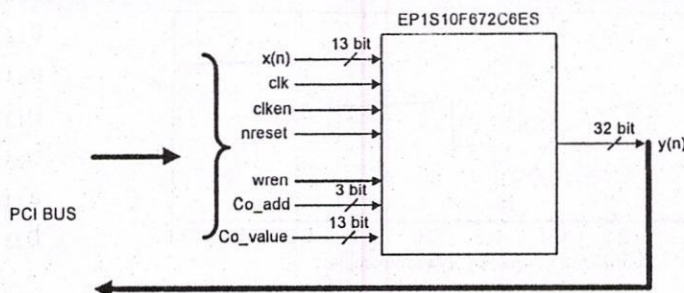
### 3.2. Thực hiện lọc số

Lọc FIR 15 hệ số, thông thấp, tần số cắt 1KHz ở tần số lấy mẫu 32KHz được thực hiện dựa trên các cơ sở đã trình bày ở trước. Đáp ứng tần số của lọc với độ chính xác 13 bit được trình bày ở Hình 13. Sơ đồ khối thử nghiệm cho hệ thống được trình bày ở Hình 14, các tín hiệu điều khiển, dữ liệu vào, kết quả sau khi lọc, các hệ số lọc...đều được điều khiển thông qua card PCI. Ngoài ra, để tăng hiệu suất thực hiện của lọc, ngõ ra của các bộ cộng và nhân được đệm bằng thanh ghi để thực hiện Pipeline.



- h0 = h14 = 65
- h1 = h13 = 111
- h2 = h12 = 242
- h3 = h11 = 445
- h4 = h10 = 685
- h5 = h9 = 911
- h6 = h8 = 1072
- h7 = 1130

Hình 13: Đáp ứng biên độ và pha của lọc FIR thử nghiệm với 15 hệ số.

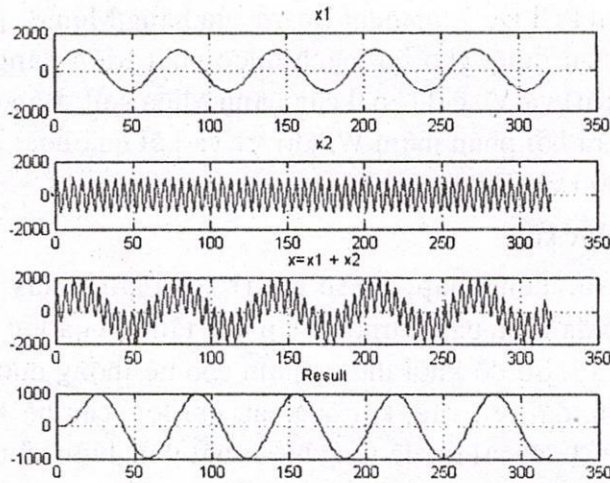


Hình 14: Sơ đồ khối thử nghiệm lọc FIR và IIR.

Kết quả sau khi tổng hợp thiết kế trên FPGA EP1S10F672C6ES:

- Tài nguyên sử dụng: 466/10.570 Logic Element, 0/920.448 memory bits, 16/48
- DSP Block 9-bit.
- Hiệu suất: 201.69 MHz.
- Latency: 4 Clock Cycles.

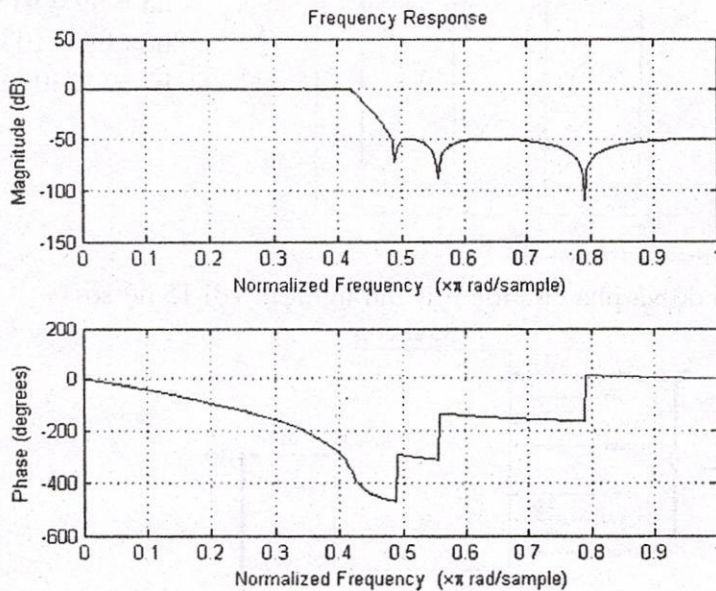
Hình 15 trình bày kết quả trong trường hợp tín hiệu vào, x, là tín hiệu tổng hợp của hai tín hiệu sin x1 có tần số 500Hz, x2 có tần số 6KHz (với tần số lấy mẫu đều là 32 KHz). Kết quả rất tốt khi tín hiệu 6 KHz bị loại bỏ hoàn toàn.



Hình 15: Kết quả thử nghiệm lọc FIR 15 hệ số.

Lọc số IIR cũng được thử nghiệm với lọc Elip bậc sáu, thông thấp, tần số cắt 5KHz ở tần số lấy mẫu 24 KHz có phương trình cho bởi (4):

$$\frac{(0.03005 + 0.04759z^{-1} + 0.03045z^{-2})(1 + 0.36319z^{-1} + 1.00000z^{-2})(1 - 0.05738z^{-1} + 1.00000z^{-2})}{(1 - 100021z^{-1} + 0.35730z^{-2})(1 - 0.66647z^{-1} + 0.68654z^{-2})(1 - 0.48167z^{-1} + 0.91928z^{-2})} \quad (4)$$



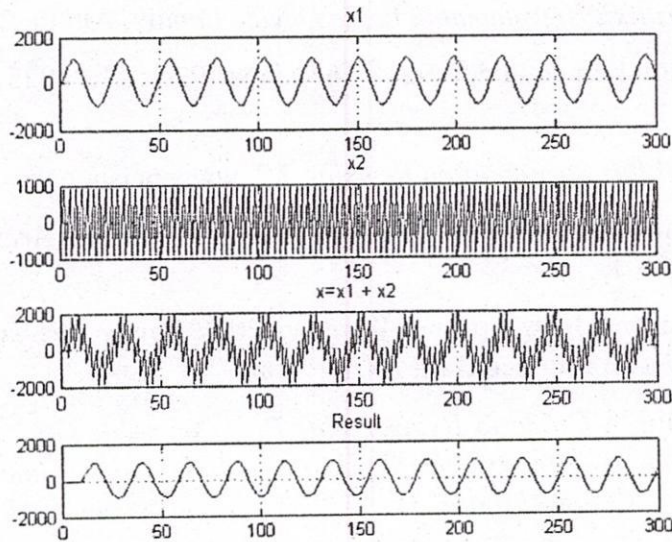
- $b_{01} = 31$
- $b_{11} = 49$
- $b_{21} = 31$
- $a_{11} = -1024$
- $a_{21} = 366$
- $b_{12} = 372$
- $b_{22} = 1024$
- $a_{12} = -682$
- $a_{22} = 703$
- $b_{13} = -59$
- $b_{23} = 1024$
- $a_{13} = -493$
- $b_{23} = 941$

Hình 16: Đáp ứng tần số và pha của lọc Elip bậc 6 thử nghiệm trong hệ thống.

Đáp ứng biên độ và pha của lọc IIR này với độ chính xác 10 bit được trình bày ở Hình 16. Sơ đồ khối thử nghiệm hệ thống được thực hiện tương tự như Hình 14. Kết quả sau khi tổng hợp thiết kế trên FPGA EP1S10F672C6ES:

- Tài nguyên sử dụng: 455/10.570 Logic Element, 0/920.448 memory bits, 36/48 DSP Block 9-bit.
- Hiệu suất: 63.22 MHz.
- Latency: 6 Clock Cycles.

Hình 17 trình bày kết quả trong trường hợp tín hiệu vào, x, là tín hiệu tổng hợp của hai tín hiệu sin x1 có tần số 1KHz, x2 có tần số 8KHz (với tần số lấy mẫu 24 KHz). Kết quả rất tốt khi tín hiệu 8 KHz bị loại bỏ hoàn toàn.



Hình 17: Kết quả thử nghiệm lọc IIR, Elip bậc 6, thử nghiệm trong hệ thống.

#### 4. KẾT LUẬN

Chúng tôi đã trình bày hai vấn đề liên quan đến các ứng dụng xử lý thời gian thực tín hiệu trên máy tính: Lập trình FPGA, thay vì dùng DSP hay ASIC, để xử lý tín hiệu trên cơ sở phần cứng, và giao tiếp FPGA với máy tính qua Bus PCI. Về nội dung đầu chúng tôi đã tạo lọc FIR và IIR thời gian thực có các hệ số lọc thay đổi tùy ý. Trên cơ sở thành công này chúng tôi có thể thay đổi loại lọc hoặc những giải thuật xử lý nào khác. Về giao tiếp PCI chúng tôi đã thực hiện theo nhiều cách, tất cả đã thành công, để cung cấp các tài liệu tham khảo cho ai cần sự trao đổi dữ liệu tốc độ cao với máy tính.

### IMPLEMENTING REAL-TIME FILTERS ON FPGA AND COMMUNICATING WITH PERSONAL COMPUTER USING PCI BUS

Huynh Huu Thuan, Nguyen Huu Phuong

Faculty of Physics, University of Natural Sciences, VNU-HCM

**ABSTRACT:** To process digital signals (representing sound, image...) efficiently in real time on a PC we must combine several measures: adding an external preprocessing board, inputting data into the PC through high speed Bus (USB or, especially, PCI), choosing appropriate programming language and technique, using parallel processing (on multiprocessor computers). In this paper, two problems are presented: implementing real time digital filters on FPGA (instead of using a DSP) as a preprocessor, and then interfacing it to PC using PCI Bus. Good results have been obtained.

#### TÀI LIỆU THAM KHẢO

- [1]. Paul M. Embree, *C Algorithms for Real-Time DSP*, Prentice Hall, 1995.
- [2]. Ashok K. Sharma, *Programmable Logic Handbook-PLD, CPLDs, & FPGA*, McGraw-Hill, 1998.



- [3]. *Altera Stratix Programmable Logic Device Family*, Altera, 2002.
- [4]. Các tài liệu liên quan đến Stratix 672 SmartPack – EP1S25 (EP1S10), Parallax, 2002.
- [5]. *PCI Local Bus Specification Revision 2.2*, [www.pcisig.com](http://www.pcisig.com).
- [6]. MindShare Inc, Tom Shanley, *PCI-X System Architecture*, Addison Wesley, 2001.
- [7]. Art Baker and Jerry Lozano, *Windows 2000 Device Driver Book: A Guide for Programmers*, Prentice Hall, 2001.
- [8]. Ray Goslin, *A Guide to Using Fields Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance*, Xilinx Corporate Applications Group, 1995.
- [9]. High Performance Multipliers in QuickLogic FPGAs, [www.quicklogic.com](http://www.quicklogic.com).
- [10]. Các tài liệu liên quan đến PCI Development Board, [www.legacy.memec.com](http://www.legacy.memec.com).
- [11]. Simplify Your PCI Designs: PCI Compiler Is Now SOPC Builder – Ready, [www.altera.com](http://www.altera.com).