

VỀ MỘT GIẢI PHÁP TRIỂN KHAI MÁY TÍNH NHÚNG SỬ DỤNG HỆ ĐIỀU HÀNH LINUX TRÊN NỀN TẢNG PHẦN CỨNG ZYNQ

Phạm Minh Thắng*, Nguyễn Văn Hiếu

Viện Điện tử, Viện Khoa học và Công nghệ quân sự.

*Email: thangpm1981@gmail.com

Nhận bài: 11/11/2022; Hoàn thiện: 15/12/2022; Chấp nhận đăng: 20/12/2022; Xuất bản: 28/02/2023.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.85.2023.167-170>

TÓM TẮT

Bài báo trình bày một số kết quả nghiên cứu về việc triển khai một máy tính nhúng sử dụng hệ điều hành Linux trên nền tảng phần cứng ZYNQ của hãng Xilinx. Các IC thuộc họ Zynq được phát hành trong 10 năm trở lại đây đã và đang định hình lại các nền tảng máy tính nhúng phổ biến, với khả năng phối hợp mềm dẻo giữa vi xử lý họ ARM và FPGA mang lại hiệu quả cao [1]. Trong bài báo này, một thiết kế GPIO (General Purpose Input Output) ở tốc độ cao được triển khai, trong đó phần cứng trên FPGA, driver trên hệ điều hành Linux và phần mềm điều khiển được phối hợp triển khai. Kết quả thử nghiệm đã chứng minh khả năng hoạt động của cổng GPIO tốc độ cao, tính linh hoạt khi được điều khiển thông qua phần mềm trên hệ điều hành Linux.

Từ khoá: Linux; ZYNQ; FPGA; Máy tính nhúng.

1. MỞ ĐẦU

Nhằm đáp ứng được yêu cầu đề ra trong việc xây dựng các hệ thống như Internet of Things, Edge Computing,... máy tính nhúng là một công cụ quan trọng [1]. Với ưu thế nhỏ gọn, linh hoạt, năng lực xử lý tiệm cận vượt trội khi so sánh với các hệ vi xử lý và tiệm cận PC thông thường, máy tính nhúng đã và đang đóng vai trò cốt lõi trong hầu hết các hệ thống tự động. Tuy nhiên, máy tính nhúng gặp nhiều khó khăn trong việc xử lý tín hiệu số thời gian thực, với các đầu vào dữ liệu tốc độ cao,... Những yếu điểm trên lại được coi là ưu điểm của các hệ thống FPGA, với tốc độ xử lý nhanh, các lõi IP (Intellectual Property) được cứng hóa rất phù hợp phát triển các ứng dụng xử lý tín hiệu số. Kết hợp một hệ thống máy tính nhúng có cả FPGA và lõi vi xử lý cho máy tính nhúng là một giải pháp khả thi mang lại hiệu quả ứng dụng cao. Trong một thập kỷ gần đây, nhiều hãng sản xuất IC như Xilinx, Altera, Gowin,... đã tích hợp FPGA và vi xử lý vào một chip duy nhất, trong đó, tiên phong là dòng chip ZYNQ của hãng Xilinx [4]. Bài báo này trình bày một giải pháp thiết kế phần mềm khi hệ điều hành Linux điều khiển lõi IP GPIO tốc độ cao trên FPGA thông qua một driver. Điều đó cho phép ứng dụng khả năng điều khiển và xử lý đa dạng trên hệ điều hành Linux kết hợp với phần cứng tốc độ cao trên FPGA, là tiền đề cho việc ứng dụng máy tính nhúng trong việc giải quyết các bài toán cần xử lý tín hiệu số tốc độ cao bằng máy tính nhúng.

2. TRIỂN KHAI MÁY TÍNH NHÚNG SỬ DỤNG NỀN TẢNG PHẦN CỨNG ZYNQ

2.1 Tổng quan về họ vi mạch ZYNQ-7000

Họ vi mạch Zynq-7000 là hệ thống SoC có khả năng lập trình toàn diện (cả phần cứng và phần mềm) đầu tiên trên thế giới. Trong mỗi một chip ZynQ-7000 đều đã tích hợp sẵn một hệ vi xử lý PS (Processing System) trên cơ sở bộ vi xử lý Cortex™-A9 lõi kép của hãng ARM và một kiến trúc logic khả trình PL (Programmable logic) sử dụng công nghệ 28 nm của Xilinx. Một kiến trúc như vậy cung cấp cho người dùng một nền tảng phần cứng với hiệu năng xử lý và công suất tiêu thụ tương đương một thiết kế ASIC (Application-Specific Integrated Circuit), trong khi vẫn đảm bảo tính linh hoạt trong thay đổi thiết kế phần cứng của FPGA, và tính dễ dàng trong việc tái cấu hình sản phẩm trên cả PS và PL. Các lõi IP mềm có thể được tạo ra trong phần PL và kết nối để mở rộng chức năng của PS. Chúng ta có thể sử dụng tổ hợp PS và PL để tạo ra các hệ

thống SoC phức tạp và hiệu năng cao [3]. Giao diện ghép nối giữa PS và PL dựa trên chuẩn AXI. Phiên bản cao nhất của AXI là AXI4 là một phần của chuẩn AMBA 3.0 do hãng ARM đề xuất. Ngoài ra, một vài kết nối từ PS có thể được định tuyến thông qua PL tới các ngoại vi ngoài chip thông qua giao diện EMIO (Endless Moments Input Output) [3].

2.2. Môi trường phát triển

Việc phát triển một ứng dụng cho máy tính nhúng cần thiết phải thực hiện thiết kế FPGA và phần mềm đồng thời trong đó có quan hệ chặt chẽ với nhau. Trong đó, phần thiết kế FPGA sẽ tạo ra các IP core với các lõi điều khiển thích hợp sẵn sàng cho việc phát triển phần mềm. Đối với phần phát triển phần mềm ứng dụng, công việc cần phải thực hiện nhiều hơn, yêu cầu nhà phát triển phải tích hợp được driver điều khiển của FPGA và hệ điều hành, dựa trên các driver đó sẽ viết các chương trình ứng dụng theo đúng mục đích yêu cầu trên ngôn ngữ lập trình C.

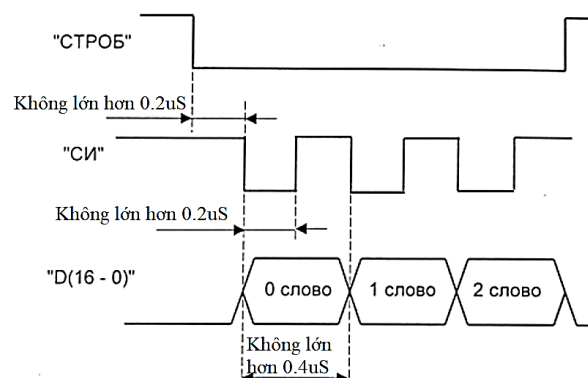
Để thực hiện những công việc đó, hãng Xilinx đã hỗ trợ các công cụ phát triển mạnh mẽ như Vivaldo và Vitis. Trong đó, Vivaldo cho phép xây dựng cấu hình hệ thống, phát triển các IP dựa trên ngôn ngữ lập trình phân cứng (Verilog hoặc VHDL), Vitis là công cụ phát triển phần mềm trên nền tảng lõi ARM cortex A9 sử dụng ngôn ngữ lập trình C. Khi máy tính nhúng sử dụng hệ điều hành Linux, Xilinx cũng cung cấp những công cụ phát triển mạnh mẽ dựa trên hệ thống mã nguồn mở Linux trong đó tập trung vào bản phân phối PetaLinux.

3. THIẾT KẾ MỘT ỨNG DỤNG GPIO TỐC ĐỘ CAO TRÊN MÁY TÍNH NHÚNG ZYNQ

Trong các ứng dụng GPIO tốc độ cao FPGA là sự lựa chọn tối ưu nhất, với ưu thế tốc độ cao, việc kiểm soát độ rộng xung, chu kỳ lặp lại,... có độ chính xác cao hơn so sánh với các giải pháp khác. Tuy nhiên, với một ứng dụng có tính phức tạp và độ mềm dẻo cao, việc sử dụng một ứng dụng chạy trên nền tảng hệ điều hành sẽ giúp quản lý, thay đổi cấu hình vào ra một cách đơn giản hơn là sử dụng các ngôn ngữ mô tả phân cứng. Đó là lý do chính để kết hợp FPGA và vi xử lý trong một hệ máy tính nhúng chuyên dụng.

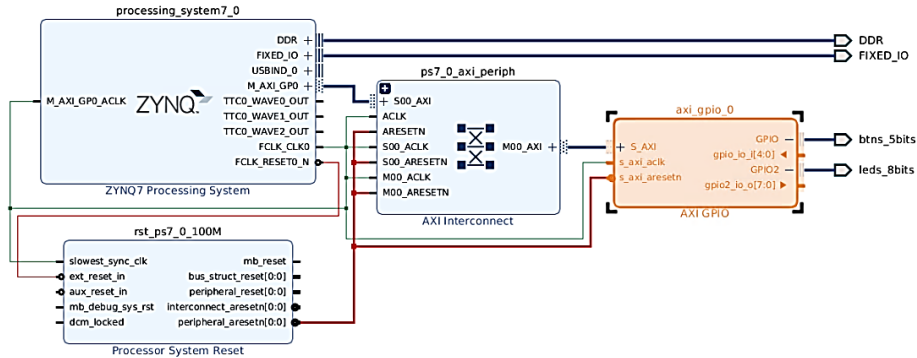
3.1. Thiết kế phần cứng

Phần cứng hệ thống là một bus dữ liệu song song tốc độ cao được mô tả như trên hình 1. Trong đó, bus dữ liệu gồm 19 cổng GPIO độc lập gồm tín hiệu báo truyền, xung clock có độ rộng xung 0.2 μ S và 17 bit dữ liệu song song.



Hình 1. Giản đồ thời gian GPIO.

Để thiết kế hệ thống này một lõi IP xử lý tín hiệu GPIO được thiết kế trên ngôn ngữ Verilog trên phần mềm Vivaldo như trên hình 2. Trong đó, các thành phần như lõi xử lý ARM Cortex A9, core IP GPIO đã viết và khối AXI Interconnect được kết nối với nhau. Kết quả của quá trình này cho phép xây dựng một cấu hình phân cứng sẵn sàng cho việc thiết lập hệ điều hành PetaLinux cùng với các driver tương ứng.



Hình 2. Kết nối phần cứng trên Vivado.

3.2. Thiết kế phần mềm

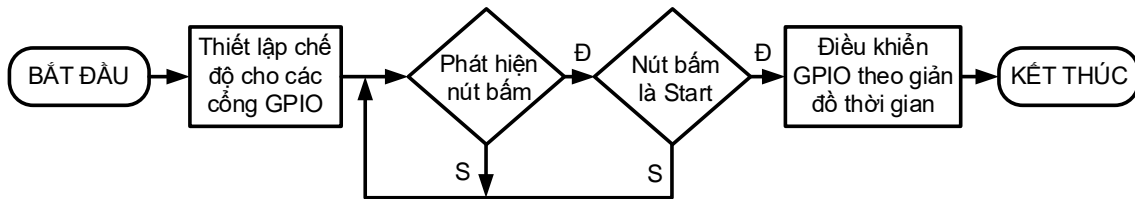
Việc thiết kế phần mềm điều khiển cổng GPIO cho máy tính nhúng Linux bao gồm hai bước chính gồm driver điều khiển và chương trình ứng dụng. Với driver điều khiển GPIO hệ điều hành Linux được hỗ trợ tương đối đầy đủ, mỗi driver đều có những ưu và khuyết điểm khác nhau. Trong đó, nổi bật nhất là trình điều khiển UIO (Userspace I/O), đây là một trình điều khiển đơn giản gần như nằm hoàn toàn trong không gian người dùng và hỗ trợ đầy đủ các ngắt. Đây cũng là trình điều khiển được thiết kế riêng cho các ứng dụng giao tiếp với FPGA trên kiến trúc Zynq sử dụng hệ điều hành PetaLinux.

```

--- Userspace I/O drivers
<> generic Hilscher CIF Card driver
<*> Userspace I/O platform driver with generic IRQ handling
<*> Userspace platform driver with generic irq and dynamic memory
<> AEC video timestamp device
<> Automata Sercos III PCI card driver
<> Generic driver for PCI 2.3 and PCI Express cards
<> Hilscher NetX Card driver
<> Texas Instruments PRUSS driver
<> Humusoft MF624 DAQ PCI card driver
<*> Xilinx AXI Performance Monitor driver
<> Xilinx AI Engine driver
    
```

Hình 3. Lựa chọn driver UIO.

Để triển khai UIO trên hệ điều hành PetaLinux, trong quá trình build hệ điều hành cần thiết phải cho phép driver này hoạt động bằng cách thông qua chương trình cấu hình kernel như trên hình 3.



Hình 4. Lưu đồ thuật toán phần mềm điều khiển GPIO.

Chương trình phần mềm điều khiển hoạt động của GPIO thông qua driver UIO có thể được thực hiện thông qua việc truy suất trực tiếp vào các vùng nhớ của thanh ghi điều khiển cổng GPIO thông qua lệnh “devmem”. Tương tự việc thiết đặt các cổng GPIO, việc thiết lập các ngắt cũng được thực hiện thông qua lệnh “devmem” trong đó, địa chỉ được trở lên là thanh ghi điều khiển ngắt. Trên cơ sở các lệnh điều khiển đã được thiết lập, một chương trình điều khiển cổng GPIO tốc độ cao được thực hiện thông qua điều khiển nút bấm với lưu đồ thuật toán như trên hình 4. Trong đó, một chương trình trên ngôn ngữ C++ được viết ngầm phát hiện các nút bấm đầu vào để khởi tạo và chấm dứt quá trình tạo xung theo đúng gián đồ thời gian như trên hình 4.

Khi nhận được nút bán Start thông qua ngắt đầu vào, dữ liệu bắt đầu được truyền, nội dung dữ liệu trong các xung được thiết lập bên trong chương trình, quá trình truyền là liên tục cho tới khi nhận được nút nhấn stop.

Chương trình sau khi được viết sẽ được biên dịch bằng trình biên dịch GCC, tệp tin nhị phân sau khi biên tập sẽ được copy xuống máy tính nhúng và được thiết lập luôn chạy khi khởi động máy tính. Khi đó, bằng cách sử dụng các nút bấm đơn giản có thể điều khiển một bus dữ liệu GPIO tốc độ cao.

4. KẾT LUẬN

Trên cơ sở triển khai một máy tính nhúng trên nền tảng phần cứng ZYNQ của hãng Xilinx, bằng cách sử dụng kết hợp giữa thiết kế phần cứng bằng ngôn ngữ mô tả phần cứng Verilog FPGA, và thiết kế phần mềm và driver điều khiển trên hệ điều hành Linux sử dụng bản phân phối PetaLinux, cho thấy khả năng triển khai đơn giản và mạnh mẽ. Bằng cách điều khiển một GPIO tốc độ cao đơn giản cho thấy được điểm mạnh trong việc kết hợp khả năng của lõi vi xử lý và FPGA. Dựa trên phần cứng tự thiết kế, máy tính nhúng dựa trên phần cứng ZYNQ cho phép làm chủ toàn bộ các ngoại vi cần thiết trong thiết kế, đảm bảo khả năng bảo mật của hệ thống, đồng thời với khả năng kết hợp chặt chẽ giữa FPGA và vi xử lý cho phép phát triển các hệ thống xử lý tín hiệu số có hiệu năng cao.

TÀI LIỆU THAM KHẢO

- [1]. Kang, Jian, Xiaojun Hei and Jianjian Song. "A Comparative Study of Zynq-Based OpenFlow Switches in a Software/Hardware Co-design." SpaCCS Workshops (2017).
- [2]. Pham Thinh Hung, Suhaib A. Fahmy, Ian Mcloughlin. "An End-to-End Multi-Standard OFDM Transceiver Architecture Using FPGA Partial Reconfiguration." IEEE Access 5, (2017).
- [3]. R. Vasanth and Krishna Prasad. "FPGA Based Design and Implementation of DUC/DDC Based OFDM for Data/Image Transmission." International Journal of Electrical and Electronic Engineering & Telecommunications, (2019).
- [4]. Gafsi, Mohamed, Nessrine Abbassi, Mohamed Ali Hajjaji, Jihen Malek and Abdellatif Mtibaa. "Xilinx Zynq FPGA for Hardware Implementation of a Chaos-Based Cryptosystem for Real-Time Image Protection." J. Circuits Syst. Comput. 30, (2021).

ABSTRACT

Embedded computer deployment solution using Linux operating system on ZYNQ

In this paper, some research results on deploying an embedded computer using Linux operating system on Xilinx's ZYNQ hardware platform are presented. ICs of the Zynq family released in the last 10 years have been reshaping popular embedded computing platforms, with flexible synergies between ARM and FPGA family of processors that deliver high efficiency first. In this paper, a high-speed GPIO design is implemented, in which the hardware on the FPGA, the driver on the Linux operating system, and the control software are jointly implemented. The test results have demonstrated the ability of the GPIO port to operate at high speed and flexibility when controlled via software on the Linux operating system.

Keywords: Linux; ZYNQ; FPGA; Embedded Computer.