

GIẢI PHÁP THIẾT KẾ VÀ THI CÔNG MÁY TÍNH NHÚNG THỜI GIAN THỰC TÍCH HỢP CÔNG NGHỆ FPGA

DESIGN AND IMPLEMENTATION SOLUTIONS FOR REAL-TIME EMBEDDED COMPUTER INTEGRATED FPGA TECHNOLOGY

Trần Hoàng Vũ¹, Nguyễn Văn Thọ², Đỗ Thành Bảo Ngọc², Chữ Đức Hoàng³

¹Trường Đại học Sư phạm Kỹ thuật – Đại học Đà Nẵng; thvu@ute.udn.vn

²Trường Đại học Duy Tân; vanthodn@gmail.com, baongocdt@gmail.com

³Quỹ Đổi mới công nghệ quốc gia - Bộ Khoa học & Công nghệ; hoangcd@most.gov.vn

Tóm tắt - Máy tính nhúng đã và đang làm nên làn sóng đổi mới trong lĩnh vực công nghệ thông tin và đặc biệt là trong lĩnh vực tự động hóa. Với ưu thế nhỏ gọn, linh hoạt và giá thành rẻ; máy tính nhúng đã và đang đóng vai trò cốt lõi trong hầu hết các hệ thống tự động. Tuy nhiên, do hạn chế về tốc độ xử lý, nó gặp nhiều khó khăn trong việc xử lý các bài toán cần tốc độ xử lý nhanh như các ứng dụng thời gian thực, xử lý ảnh, xử lý video... Trong bài báo này chúng tôi đề xuất giải pháp thiết kế máy tính nhúng có tích hợp FPGA nhằm giải quyết hạn chế trên. Với thiết kế này các công việc phức tạp cần tốc độ xử lý có thể được chuyển sang thực thi trên FPGA. Một hệ xử lý truyền thông thời gian thực cũng được thực thi thử nghiệm trên máy tính nhúng để đánh giá hiệu năng của thiết kế.

Từ khóa - Máy tính nhúng; hệ thống thời gian thực; FPGA; PCB tốc độ cao; vi xử lý.

Abstract - Embedded computers are making waves in innovation in the field of information technology and automation. It is playing a core role in most automated systems because of the compact, flexibility and low cost. However, due to limitations in processing speed, it is difficult to handle problems that need fast processing speed such as real-time applications, image processing, video processing ... To solve this limitation, we propose an embedded computer design which has an integrated FPGA. As such, complex tasks can be built on FPGA. A real-time information transmission system has been implemented on our embedded computer to assess its performance.

Key words - Embedded computer; realtime system; FPGA; high speed PCB; microprocesor.

1. Giới thiệu

Những năm gần đây, sự phát triển nhanh chóng của thông tin vô tuyến đã và đang ảnh hưởng ngày càng to lớn đến nhiều lĩnh vực khác nhau trong đời sống kinh tế xã hội. Điều này dẫn đến vấn đề bảo mật trong thông tin vô tuyến lại trở thành tâm điểm của nhiều nhà nghiên cứu trong và ngoài nước, đặc biệt là trong các lĩnh vực quan trọng đòi hỏi tính bảo mật cao như an ninh, quốc phòng, tài chính...

Ngày nay, công nghệ hệ thống nhúng đã phát triển mạnh mẽ, thiết kế đóng vai trò quan trọng trong việc phát triển hệ thống. Sự phổ biến ngày càng tăng của các hệ thống nhúng đã làm cho nhiều nhà khoa học quan tâm nghiên cứu cải thiện, tối ưu hóa kiến trúc của các hệ thống nhúng. Tối ưu hóa các hệ thống nhúng bao gồm các khía cạnh như hiệu suất [1], [2], [3], tiêu thụ điện năng [4], [5], [6], [7], chi phí, kích thước [8], [9], [10], ...

Các hệ thống thời gian thực mà điển hình là các hệ thống xử lý nhận thức hỗ trợ bộ vi xử lý cho các công nghệ có thể cho phép các máy tính cảm nhận thế giới theo cách con người. Nó nhằm mục tiêu các lĩnh vực như phát hiện đối tượng, nhận dạng và theo dõi, nhận dạng giọng nói và cử chỉ và khả năng đa phương thức như đọc môi để hỗ trợ nhận dạng giọng nói. Các ứng dụng cho việc xử lý nhận thức đều rất lớn và đa dạng. Ngày càng có nhiều thiết bị máy tính đang được nhúng vào vô hình trong môi trường sống của chúng ta và chúng ta nhận thấy sự tồn tại của chúng chỉ khi chúng ngừng phục vụ chúng ta. Các môi trường nhúng di động trong tương lai cần ít nhất hỗ trợ các ứng dụng tính vi như nhận dạng giọng nói, nhận dạng tính năng trực quan, kết nối mạng không dây an toàn và xử lý phương tiện nói chung.

Bởi các ứng dụng thời gian thực rất phong phú trong

các hệ thống nhúng di động như PDA thông minh, robot không người lái và các thiết bị giả cho người khiếm thị và khiếm thính. Một vấn đề cơ bản gây ra các ứng dụng này là chúng yêu cầu hiệu năng đáng kể hơn so với các bộ xử lý nhúng hiện tại có thể phân phối. Hầu hết các bộ vi xử lý nhúng và công suất thấp, chẳng hạn như Intel XScale, không có tài nguyên phần cứng và hiệu suất cần thiết để hỗ trợ trình nhận dạng giọng nói đầy đủ tính năng. Ngay cả các bộ vi xử lý hiệu năng cao hiện đại cũng hầu như không thể theo kịp các yêu cầu thời gian thực của các ứng dụng nhận thức tính vi.

Với tốc độ phát triển vi xử lý theo định luật Moore, vấn đề hiệu suất không phải là vấn đề quan trọng. Tuy nhiên hai vấn đề quan trọng vẫn còn. Thứ nhất, mức tiêu thụ năng lượng đi kèm với mức hiệu suất yêu cầu thường có cường độ vượt quá mức năng lượng nhúng điển hình. Hơn nữa, các yêu cầu về năng lượng của các bộ vi xử lý hiệu năng cao mới ngày càng tăng. Kết luận là quy mô công nghệ một mình không thể giải quyết vấn đề này. Thứ hai, nhận thức và giao diện bảo mật là do tự nhiên luôn hoạt động. Điều này giới hạn tính khả dụng của bộ xử lý đối với các tác vụ tính toán khác, chẳng hạn như hiệu được những gì đã được nhận thức. Giải pháp thông thường để giảm tiêu thụ điện trong khi tăng hiệu suất là sử dụng ASIC. Với sự phức tạp và luôn luôn về bản chất của nhiệm vụ nhận thức, một cách tiếp cận có liên quan hơn sẽ là sử dụng ASIC như một bộ xử lý kết hợp với bộ xử lý máy chủ công suất thấp.

Trong giai đoạn đầu của nghiên cứu này, một bộ xử lý ASIC cho một trong những giai đoạn thống trị của hệ thống nhận dạng giọng nói Sphinx CMU đã được nghiên cứu [11]. Nỗ lực này củng cố quan điểm cho rằng ASIC là tốn kém và không linh hoạt. Chi phí chế tạo cao của họ cùng

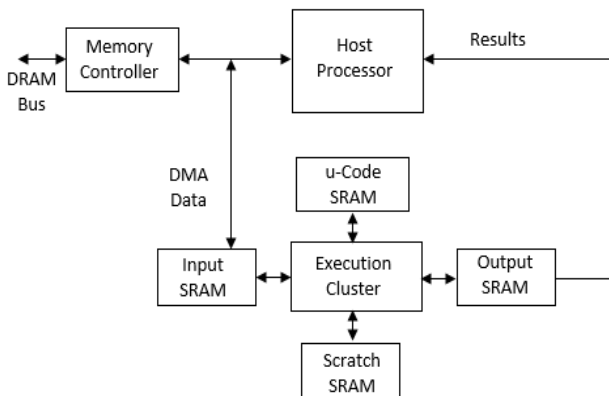
với các chi phí liên quan đến một chu kỳ thiết kế kéo dài rất khó để khấu hao trừ trường hợp sản xuất với quy mô rất lớn. Chuyên môn ASIC vốn có làm cho nó vô cùng khó khăn để hỗ trợ nhiều ứng dụng, phương pháp mới, hoặc thậm chí cải tiến thuật toán tiến hóa.

Các ứng dụng những phát triển nhanh chóng và các hệ thống nhúng là cực kỳ nhạy cảm với chi phí, những vấn đề này thúc đẩy việc tìm kiếm một cách tiếp cận mục đích chung hơn. Việc sử dụng các thiết bị logic và cấu hình lại có thể cấu hình lại là một cách tiếp cận phổ biến khác [12]. Khả năng cấu hình lại vốn có của FPGA cung cấp một mức độ chuyên môn hóa trong khi vẫn giữ được tính tổng quát đáng kể. Tuy nhiên, FPGA có một bất lợi đáng kể cả về hiệu năng và công suất khi so sánh với các chức năng logic ASIC hoặc CPU.

Xuất phát từ các phân tích trên, nhằm mục đích cân bằng giữa các mục tiêu về hiệu năng, chi phí và tính linh hoạt; chúng tôi đề xuất giải pháp thiết kế hệ thống máy tính nhúng tích hợp FPGA. Khối FPGA tích hợp vào giúp máy tính khắc phục các nhược điểm về hiệu suất tính toán trong khi vẫn giữ được tính linh hoạt và bảo đảm chi phí thấp.

Phần còn lại của bài báo được sắp xếp như sau: Phần 2 giới thiệu kiến trúc hệ thống máy tính nhúng đề xuất. Phần 3 trình bày các kết quả về thiết kế layout. Phần 4 trình bày kết quả thực nghiệm và đánh giá hiệu quả của thiết kế đề xuất. Cuối cùng phần 5 là phần kết luận.

2. Kiến trúc hệ thống



Hình 1. Tổ chức thực hiện của vi xử lý

Hình 1 thể hiện tổ chức thực hiện của một hệ vi xử lý. Bộ xử lý đưa dữ liệu vào hoặc ra bộ xử lý thông qua các bộ đệm SRAM. Các thuật toán xử lý thời gian thực có xu hướng được định hướng luồng, tức là, chúng xử lý một chuỗi các bản ghi dữ liệu tương tự, nơi các bản ghi dữ liệu có thể là các gói hoặc khối tín hiệu rời rạc, khung video hoặc đầu ra của các quy trình xử lý luồng khác. Mỗi gói dữ liệu đầu vào được xử lý bởi một thuật toán tương đối đơn giản và thường xuyên thường đề cập đến một số bảng trạng thái hoặc lịch sử cục bộ hạn chế để tạo ra một gói đầu ra. Các thuật toán thường được định hướng vòng lặp với các thành phần chi phối được lồng vào nhau cho các vòng với các cơ quan phụ thuộc vào dòng chảy. Một số các nghiên cứu đã đề xuất kiến trúc vi xử lý được tối ưu hóa cho kiểu tính toán này được gọi là bộ xử lý luồng [13].

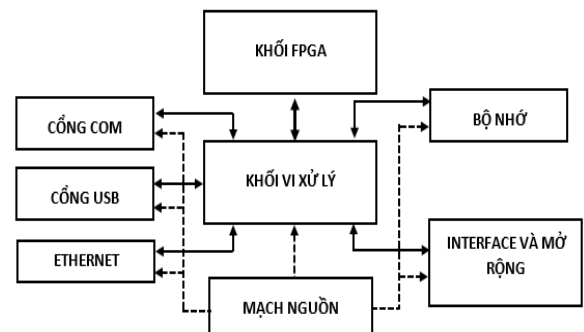


Hình 2. Luồng xử lý dữ liệu

Thay vì thiết kế một kiến trúc vi xử lý chuyên biệt để tối ưu hóa việc xử lý các ứng dụng thời gian thực, chúng tôi thêm vào một khối chức năng logic tùy biến. Luồng xử lý được thể hiện ở Hình 2. Các gói dữ liệu cần xử lý sẽ được chuyển sang cho khối Logic Function. Khối này được tùy biến để có thể giải quyết các thuật toán xử lý dữ liệu khác nhau bằng mô tả phần cứng HDL, dữ liệu sau khi được xử lý sẽ được trả lại cho vi xử lý ra quyết định.

Trong thực tế, có rất nhiều cấu hình tương ứng với kiến trúc của hệ thống nhúng. Các cấu hình này khác nhau về tốc độ CPU, kích thước bộ nhớ, chiều rộng Bus, cổng I/O... [1], [2]. Chúng tôi xây dựng các chức năng khách quan để đánh giá cấu hình của kiến trúc hệ thống nhúng trên cơ sở cân bằng giữa các mục tiêu tối ưu.

Hình 3 là kiến trúc máy tính nhúng được chúng tôi thiết kế để đánh giá hiệu năng của thiết kế, vi xử lý ARM được lựa chọn cho máy tính nhúng và khối chức năng tùy biến là FPGA. Chúng tôi lựa chọn dòng Spartan 3E cho hệ thống thử nghiệm của mình.



Hình 3. Sơ đồ khối của máy tính nhúng thời gian thực

Trong kiến trúc này, chúng tôi tích hợp vào máy tính nhúng một khối FPGA nhằm tùy biến phần cứng và thực thi những nhiệm vụ đòi hỏi hiệu năng cao và vi xử lý không đủ tài nguyên để giải quyết. Khối FPGA được kết nối trực tiếp để dễ dàng điều khiển và tùy biến.

Mặt khác một giao diện tùy biến FPGA với ngoại vi được tổ chức riêng biệt và cổng nạp JTAG cho FPGA độc lập hoàn toàn cho phép trong trường hợp cần thiết hệ thống có thể làm việc như một kit phát triển FPGA đơn thuần, độc lập với vi xử lý của máy tính nhúng.

3. Kết quả mô phỏng PCB

Các ứng dụng thời gian thực như xử lý ảnh, xử lý video thường có khối lượng dữ liệu rất lớn, dòng dữ liệu giữa vi xử lý và FPGA thường có tốc độ rất cao, do đó việc thiết kế bo mạch in của máy tính nhúng để đảm bảo chất lượng tín hiệu là vấn đề rất quan trọng trong thiết kế máy tính nhúng.

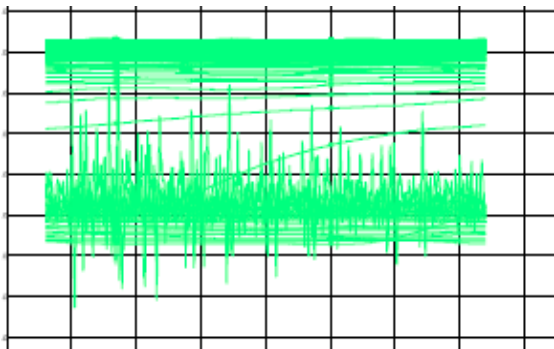
Thiết kế mạch in tốc độ cao trên đặt ra nhiều thách thức đối với người thiết kế trong việc đảm bảo chất lượng tín hiệu truyền dẫn như nhiễu xuyên âm, bức xạ trường điện từ, phối hợp trở kháng và đảm bảo cân bằng độ trễ truyền

dẫn giữa các đường tín hiệu. Các khó khăn thách thức này được gọi theo một khái niệm tiêu chuẩn là tích hợp tín hiệu (SI: Signal Integrity). Nếu quá trình thiết kế không có sự hỗ trợ mô phỏng và phân tích bởi các công cụ mô phỏng tiêu chuẩn công nghiệp, thì việc kiểm tra chất lượng tín hiệu sau khi chế tạo mạch in PCB (Printed Circuit Board) sẽ vô cùng khó khăn, đồng thời chất lượng tín hiệu truyền dẫn sẽ không được đảm bảo (hay còn gọi là khả năng tích hợp tín hiệu không được đảm bảo), dẫn đến phải làm lại mạch in nhiều lần mà không tìm được ra nguyên nhân chính xác. Điều này không chỉ làm tăng chi phí, mà còn kéo dài thời gian đưa sản phẩm ra thị trường.

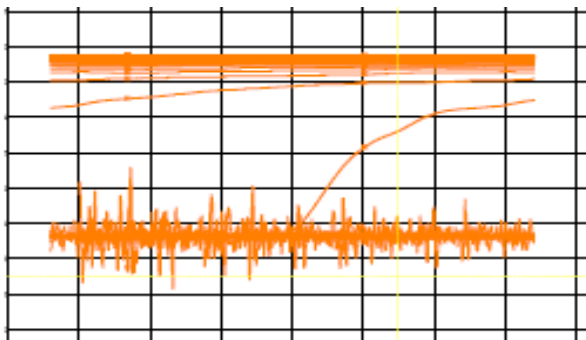
Khi một đường dẫn truyền dẫn tín hiệu, giữa nó và các đường truyền dẫn kề cận sẽ xuất hiện dung kháng tương hỗ kết hợp và cảm kháng tương hỗ kết hợp. Lý thuyết về tương hỗ kết hợp cho thấy rằng, các dòng điện tín hiệu trở về của một đường dẫn truyền dẫn tín hiệu sẽ tạo ra các trường từ. Các trường từ này tiếp tục cảm ứng ra các điện áp trong các đường dẫn khác nằm trong vùng trường từ này. Các điện áp cảm ứng tỉ lệ thuận với sự biến đổi của tín hiệu truyền. Các điện áp này ảnh hưởng càng lớn khi thời gian tăng (rise time) của tín hiệu càng ngắn (tần số càng cao), có nghĩa là khi tốc độ truyền dẫn càng cao. Điện áp cảm ứng này gọi là nhiễu xuyên âm.

Trong các mạch số, đặc biệt là các mạch tốc độ cao trên 1 GHz, xuyên âm do cảm kháng luôn lớn hơn nhiễu xuyên âm do dung kháng. Vì vậy, các phương pháp giảm xuyên âm luôn tập trung vào việc giảm xuyên âm do cảm kháng.

a. Dạng sóng đầu phát



b. Dạng sóng đầu thu



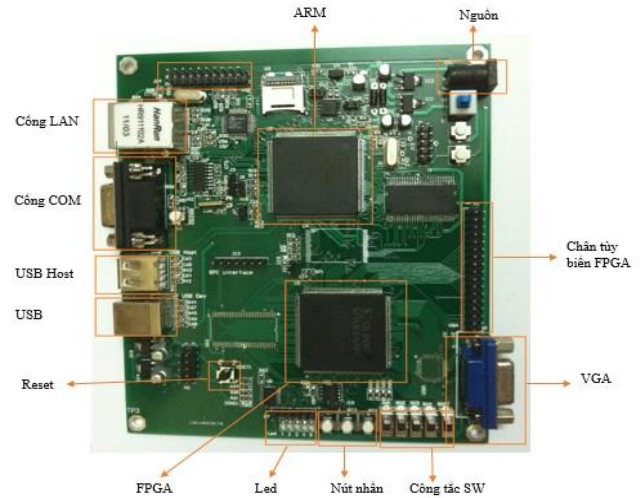
Hình 4. Kết quả mô phỏng ảnh hưởng nhiễu xuyên âm

Để đánh giá chất lượng thiết kế mạch in PCB chúng tôi sử dụng công cụ HyperLynx của MentorGraphics để đánh giá ảnh hưởng của nhiễu xuyên âm đến chất lượng tín hiệu kể với tốc độ truyền dữ liệu 1GHz. Kết quả mô phỏng ảnh hưởng của nhiễu xuyên âm được thể hiện ở Hình 4. Theo

kết quả mô phỏng ta thấy mắt mờ to, không thấy ảnh hưởng của nhiễu giữa 2 đường truyền. Biên độ nhiễu rất thấp.

4. Kết quả thực thi và thử nghiệm

4.1. Thi công phần cứng máy tính nhúng



Hình 5. Máy tính nhúng thời gian thực thử nghiệm

Chúng tôi đã thiết kế và thi công thử nghiệm máy tính nhúng với vi xử lý được lựa chọn là AT91SAM9260, bộ nhớ SRAM 256MB, khối FPGA được chọn là XC3S500E của Xilinx. Đây là chip FPGA được sử dụng trong các Kit Spartan nổi tiếng của Xilinx. XC3S500E có mật độ tích hợp cổng tương đối cao và đủ các đặc tính cho thiết kế phần cứng nhiều chức năng. Nó được hỗ trợ bởi các công cụ phần mềm của Xilinx. Các thông số kỹ thuật cụ thể của XC3S500E được mô tả ở Bảng 1.

Bảng 1. Thông số kỹ thuật của XC3S500E

System Gate	Logic cells	Total CLBs	Total Slices	Maximum User I/O	Maximum Differential I/O pairs
500.000	10.476	1164	4656	232	92

Hình 5 là sản phẩm hoàn thiện. Ngoài khối phần cứng tùy biến FPGA, máy tính nhúng có đầy đủ các cổng chức năng cơ bản như VGA, USB, Ethernet....

4.2. Nạp hệ điều hành

Sau khi đã thi công hoàn chỉnh phần cứng máy tính nhúng, chúng tôi tiến hành nạp hệ điều hành Linux cho máy tính nhúng.

Trình tự Bootloader của AT91SAM9260 được thể hiện ở Hình 6. Đầu tiên, nó khởi tạo các cổng đơn vị gỡ lỗi nối tiếp (DBGU) và cổng thiết bị USB.

Sau đó chương trình DataFlash Boot được thực thi. Nó tìm kiếm một chuỗi tám vector ARM hợp lệ trong một DataFlash kết nối với SPI. Tất cả các vector phải là nhánh B hoặc LDR lấy giá trị thanh ghi trừ vector thứ sáu. vector này được sử dụng để lưu trữ các kích thước của hình ảnh để tải về. Nếu một chuỗi hợp lệ được tìm thấy, code được tải về vào SRAM nội bộ. Tiếp theo là remap và nhảy tới địa chỉ đầu tiên của SRAM.

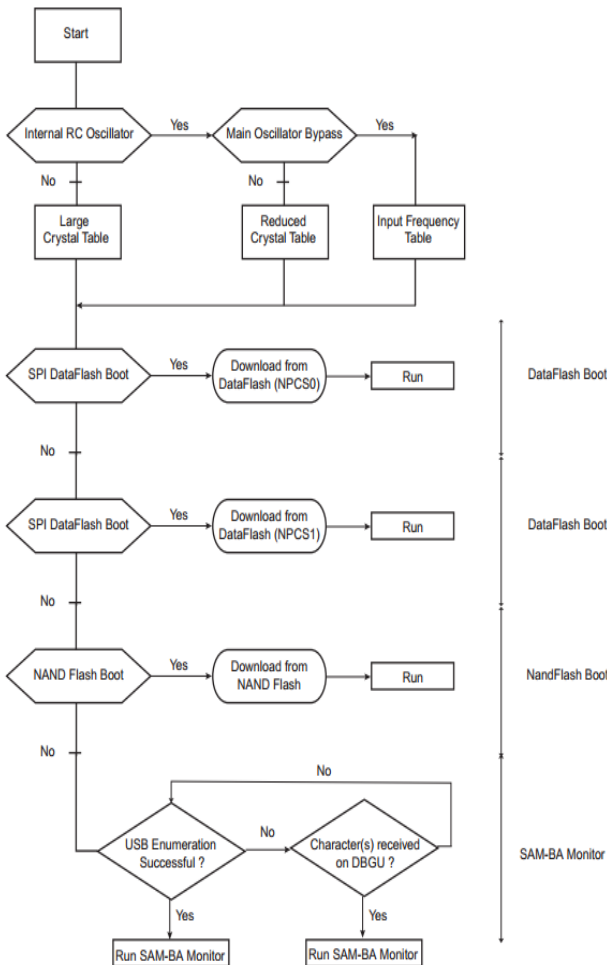
Nếu không có chuỗi vector ARM hợp lệ được tìm thấy, các chương trình DataFlash Boot được thực thi trên chip

thứ hai lựa chọn.

Nếu không có trình tự vector ARM hợp lệ được tìm thấy, chương trình NAND Flash Boot sau đó được thực hiện.

Các chương trình NAND Flash Boot sẽ tìm kiếm chuỗi tám vector ARM hợp lệ. Nếu một trình tự như vậy được tìm thấy, code được tải về vào SRAM nội bộ. Tiếp theo là remap và nhảy tới địa chỉ đầu tiên của SRAM.

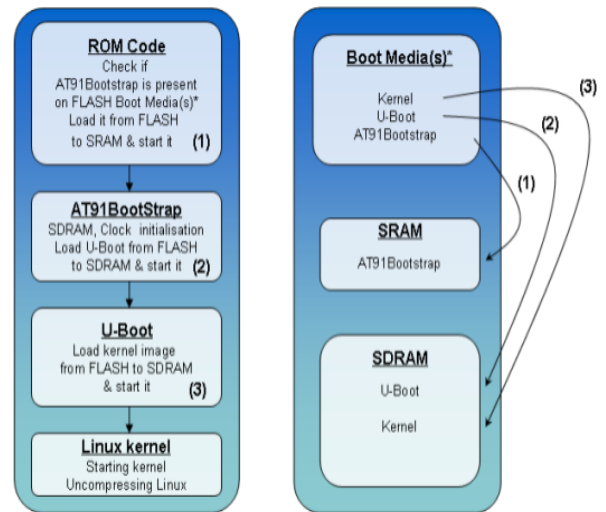
Nếu không có trình tự vector ARM hợp lệ được tìm thấy, Màn hình SAM-BA sau đó được thực hiện. Nó chờ đợi cho các giao đổi giữa các thiết bị USB, hoặc trên cổng nối tiếp DBGU.



Hình 6. Trình tự BootLoader của máy tính nhúng

Hình 7 thể hiện quá trình Boot Linux của máy tính nhúng. Máy tính nhúng của chúng tôi thiết kế thực thi ROM boot, load AT91AT91BootStrap.bin từ AT45DB041D-SU (SPI NPCS1) vào SRAM, thực hiện lệnh nhảy và thực thi chương trình AT91BootStrap. Sau đó nó thực hiện khởi động PLL cho system clock, khởi động SDRAM controller, load u-boot.bin từ AT45DB041D-SU tại offset 0x8400 vào địa chỉ 0x21F00000 của SDRAM. Sau đó thực hiện lệnh nhảy vào vùng SDRAM để chạy chương trình U-Boot.

U-Boot load kernel ulmage từ nhiều nguồn khác nhau (TFTP, NAND FLASH...) chép vào địa chỉ 0x20000000 trên SDRAM, sau đó thực hiện lệnh boot hệ điều hành Linux.

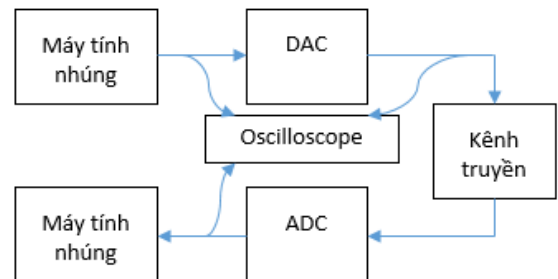


Hình 7. Trình tự Boot kernel Linux của máy tính nhúng

4.3. Thử nghiệm

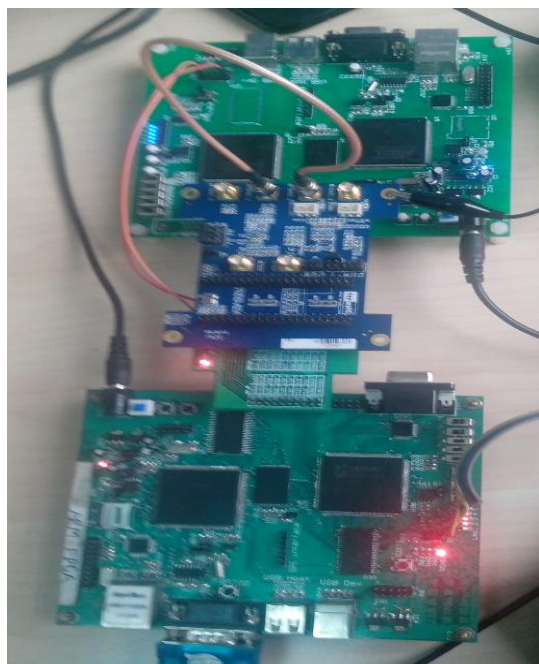
Để đánh giá hiệu năng của máy tính nhúng thời gian thực chúng tôi thực thi thử nghiệm một hệ thống truyền thông thời gian thực giữa 2 máy tính theo mô hình như Hình 8. Trong mô hình này, bên phát là một máy tính nhúng có nhiệm vụ điều chế PSK tín hiệu số trước khi đưa lên kênh truyền. Bên thu cũng là một máy tính nhúng thực hiện giải điều chế để nhận lại dữ liệu. Trong thử nghiệm này chúng tôi xây dựng một phần mềm chạy trên máy tính nhúng bên phát có nhiệm vụ sinh dữ liệu và chuyển sang khối FPGA. Một HDL được nạp vào khối FPGA thực hiện điều chế PSK và đưa ra DAC. Một chương trình khác được nạp lên máy tính nhúng bên thu có nhiệm vụ nhận dữ liệu từ khối FPGA. Một HDL giải điều chế được nạp vào khối FPGA. Một máy dao động ký được sử dụng để đo đặc và so sánh dạng sóng bên thu và bên phát.

Trong mô hình này công việc thực hiện điều chế/ giải điều chế PSK - đòi hỏi đáp ứng thời gian thực do khối FPGA thực hiện. Đây là công việc và vi xử lý trong các máy tính nhúng không thể thực hiện được.

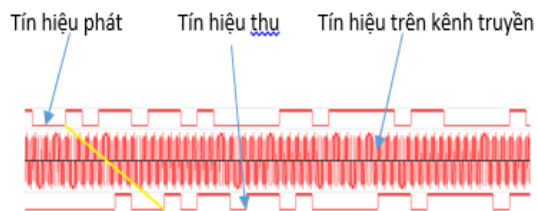


Hình 8. Mô hình thử nghiệm máy tính nhúng

Hình 9 là hình ảnh kết nối thử nghiệm thực tế của chúng tôi. Chúng tôi sử dụng 2 máy tính nhúng tích hợp FPGA do chúng tôi thiết kế và 1 module AD/DA. Kênh truyền là cáp đồng trục kết nối đầu ra của DAC với đầu vào ADC. Hình 10 là kết quả đo đạc từ Oscilloscope số. Kết quả thử nghiệm cho thấy hệ thống truyền thông thực hiện chính xác và đảm bảo chất lượng thời gian thực. Kết quả đo trên dao động ký cho thấy hệ thống đảm bảo cả về sự đúng đắn của dữ liệu truyền và yếu tố thời gian thực.



Hình 9. Hình ảnh thử nghiệm thực tế



Hình 10. Kết quả đo tín hiệu trên máy dao động ký.

5. Kết luận

Trong bài báo này chúng tôi đề xuất một kiến trúc máy tính nhúng phục vụ cho các ứng dụng thời gian thực trên cơ sở tích hợp một khối FPGA vào để có thể thực hiện các công việc có độ phức tạp cao và cần tốc độ xử lý nhanh. Kiến trúc đề xuất vừa đảm bảo cân bằng giữa hiệu năng cho các ứng dụng thời gian thực và các yếu tố mức năng lượng cũng như giá thành của hệ thống. Một phiên bản thử nghiệm đã được chế tạo và thử nghiệm thành công một ứng dụng truyền thông thời gian thực chứng tỏ hiệu quả của thiết kế.

Lời cảm ơn

“Nghiên cứu này được tài trợ bởi Quỹ Phát triển Khoa học và Công nghệ Đại học Đà Nẵng trong đề tài mã số B2017-ĐN06-09”

TÀI LIỆU THAM KHẢO

- [1] A. Asaduzzaman, N. Limbachiya, I. Mahgoub, F. Sibai. Evaluation of ICACHE Locking Technique for Real-Time Embedded Systems, Proc. IEEE Int.Conf. on Innovations in Info. Technology (IIT'07), 2007.
- [2] Mehdi Alipour et al, “Cache Power and Performance Tradeoffs for Embedded Applications”, 011 International Conference on Computer Applications and Industrial Electronics (ICCAIE 2011).
- [3] A. Asaduzzaman, F.N.Sibai “Investigating Cache Parameters and Locking in Predictable and Low Power Embedded Systems”, 22nd International Conference on Microelectronics
- [4] Guo, Hui, “A Structural Customization Approach for Low Power Embedded Systems Design” 2010 IEEE/ACM International Conference on Green Computing and Communications
- [5] Jong Wook Kwak; Ju Hee Choi; " Selective Access to Filter Cache for Low-Power Embedded Systems" 43rd Hawaii International Conference on System Sciences (HICSS), pp. 1-8, 2010
- [6] Ji Gu, Hui Guo and Patrick Li, “ROBTIC: An On-Chip Instruction Cache Design for Low Power Embedded Systems”, 2009 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.
- [7] Zhou, Yu and Guo Hui, “Application Specific Low Power ALU Design”, 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing
- [8] T.S. Rajesh Kumar, C.P. Ravikumar, R. Govindarajan “Memory Architecture Exploration Framework for Cache Based Embedded SoC”, 21st International Conference on VLSI Design
- [9] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit design. In Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), pages 201{204, June 2000.
- [10] J. Christman and E. Alley, A Hands-on Approach to Demonstrating Hardware/ Software”, In ASEE 2011 Annual conference & Exposition, Vancouver, BC, 2011.
- [11] B. Mathew, A. Davis, and Z. Fang. A low-power accelerator for the Sphinx 3 speech recognition system. In Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '03), pages 210{219, October 2003.
- [12] A. DeHon. FPGA-coupled microprocessors: Commodity ICs for the early 21st century. In D. A. Buell and K. L. Pocek, editors, IEEE Workshop on FPGAs for Custom Computing Machines, pages 31{39, Los Alamitos, CA, 1994. IEEE Computer Society Press.
- [13] W. Wolf and J. Madsen, “Embedded systems education for the future”, Proc. IEEE, Vol.88, no.1,pp. 23-30, Jan. 2000.

(BBT nhận bài: 05/10/2018, hoàn tất thủ tục phản biện: 25/10/2018)