

BỘ VI ĐIỀU KHIỂN ĐA NĂNG

Trần Thanh Hùng, Trần Hữu Danh¹

ABSTRACT

This research develops a multipurpose microcontroller system (MPC) to help technicians and students in study and design control systems. Based on an 8951 microcontroller, the MPC is built to satisfy almost basic needs of control. Using the MPC, technicians and students can develop control applications easily and quickly. Based on an efficient algorithm, the MPC has ability to run program by steps or by blocks, thing that can not be done by 8951. This is very important in designing control systems because users can stop application programs at any points to look for errors not only in software but also in hardware. With such abilities, MPC is very helpful for everyone in studying microcontroller technique and designing control systems.

Keyword: MPC- multipurpose microcontroller, step, block, AutoRun, ISR.

Title: Multipurpose Microcontroller

1 GIỚI THIỆU

Để đáp ứng nhu cầu điều khiển trong thực tế, các hãng đã tung ra nhiều loại chip vi điều khiển khác nhau. Ở Việt Nam hiện nay phổ biến nhất vẫn là vi điều khiển họ 89 của hãng Atmel (8951, 52, 55, 892051,...). Sử dụng các chip này, những người hoạt động trong lĩnh vực điện tử đã tạo nên được các sản phẩm đáp ứng tốt các yêu cầu về điều khiển trong thực tế. Tuy nhiên khi đứng trước các yêu cầu đặt ra, mỗi người phải đi thiết kế một bộ vi điều khiển riêng, đều phải trải qua các công đoạn vẽ mạch, làm mạch in, ráp mạch, viết chương trình, chạy thử... Như vậy rất mất thời gian và công sức, hơn nữa việc thiết kế sẽ chậm trễ, không đáp ứng được yêu cầu của khách hàng,...

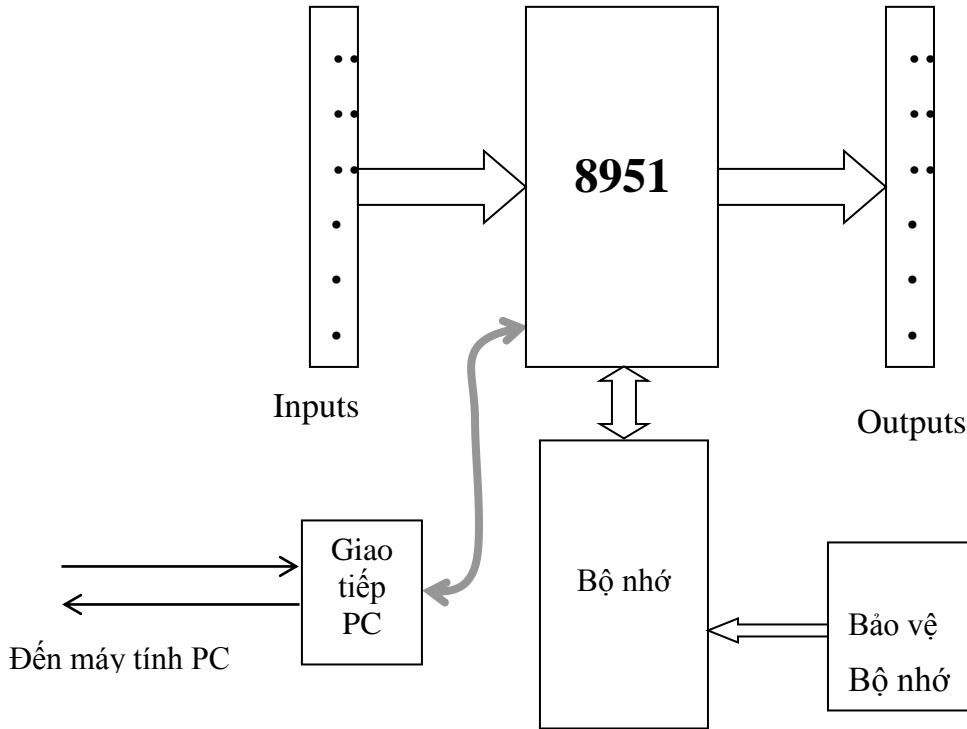
Một vấn đề khó khăn nữa mà một người thiết kế hay gặp phải là các lỗi. Ít có hệ thống nào được thiết kế lần đầu mà hoàn chỉnh, chạy được. Thường có rất nhiều lỗi xảy ra. Lỗi có thể phát sinh từ phần cứng, ví dụ như đường mạch bị đứt, bị chạm, mối hàn không tốt, IC bị hư một phần hoặc toàn bộ... hoặc xuất phát từ phần mềm (vô số lỗi có thể xảy ra) hoặc kết hợp cả hai. Việc sửa lỗi không phải là việc dễ dàng và thường chiếm rất nhiều thời gian trong quá trình thiết kế cũng như trong thời gian vận hành của thiết bị sau này. Đặc biệt việc sửa chữa thiết bị đang trong quá trình vận hành (nối kết với các thiết bị khác) lại càng khó khăn hơn.

Nghiên cứu này nhằm mục đích giải quyết các vấn đề trên. Dựa trên chip vi điều khiển 8951, một bộ vi điều khiển đa năng (Multipurpose microcontroller-MPC) được xây dựng để đáp ứng được hầu hết các nhu cầu về điều khiển tự động thường gặp trong thực tế. Từ đó tạo cơ sở ban đầu giúp các ứng dụng được điều khiển được triển khai dễ dàng và nhanh chóng hơn. Các phần mềm hỗ trợ, giúp hệ thống có những khả năng mạnh như chạy từng bước, chạy từng đoạn lệnh,... cũng được phát triển.

2 THIẾT KẾ PHẦN CỨNG

Để có thể thực hiện được các mục tiêu trên, đòi hỏi phải có một phần cứng được thiết kế hợp lý (Hình 1). Về cơ bản, phải có thêm bộ nhớ RAM bên ngoài để chứa chương trình người dùng (download từ máy tính PC). Bộ nhớ này phải được bảo vệ cẩn thận để không bị mất dữ liệu khi bị cúp điện. Ngoài ra phải có thêm các ngõ vào / ra vì khi đã sử dụng bộ nhớ bên ngoài thì 8951 chỉ còn lại 8 ngõ vào/ra.

¹ Khoa Công Nghệ Thông Tin



Hình 1: Sơ đồ khối tổng quát của bộ điều khiển đa năng MPC

2.1 Thiết kế bộ nhớ

Bên trong 8951 đã tích hợp sẵn 4 KB ROM (thuộc loại Flash memory) và 128 byte RAM. Trong đó ROM dùng để chứa chương trình điều khiển và RAM để chứa dữ liệu. Bộ nhớ này là khá nhỏ nhưng vẫn đáp ứng được các yêu cầu điều khiển. Tuy nhiên khi ứng dụng đòi hỏi chương trình lớn hoặc dữ liệu nhiều thì bộ nhớ nội không đáp ứng được. Khi đó 8951 cho phép mở rộng thêm bộ nhớ bên ngoài, có thể gắn thêm tối đa 64 KB ROM và 64 KB RAM [1,2].

Trong thực tế khi thiết kế bộ nhớ, người ta thường căn cứ vào nhu cầu và linh kiện có sẵn (hoặc thông dụng) để thiết kế. Để hệ thống tổng quát, đa năng, cấu hình bộ nhớ cần thiết kế phải linh hoạt, cho phép cắm nhiều loại IC nhớ có dung lượng khác nhau trên một đế cắm, không phân biệt ROM và RAM.

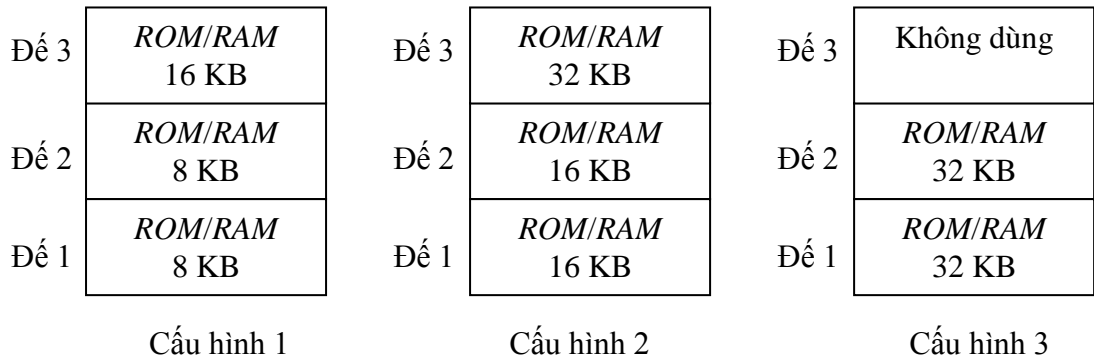
Trên thị trường hiện nay, đối với IC ROM có các loại thông dụng 8KB, 16 KB và 32 KB, đối với RAM là 8 KB và 32 KB [3]. Các loại này có đều có 28 chân và có sơ đồ khá giống nhau nên có thể cắm trên cùng một đế được.

Từ thực tế trên, bộ nhớ được thiết kế với 3 đế cắm IC nhớ, có thể cắm được các IC nhớ theo cấu hình ở hình 2.

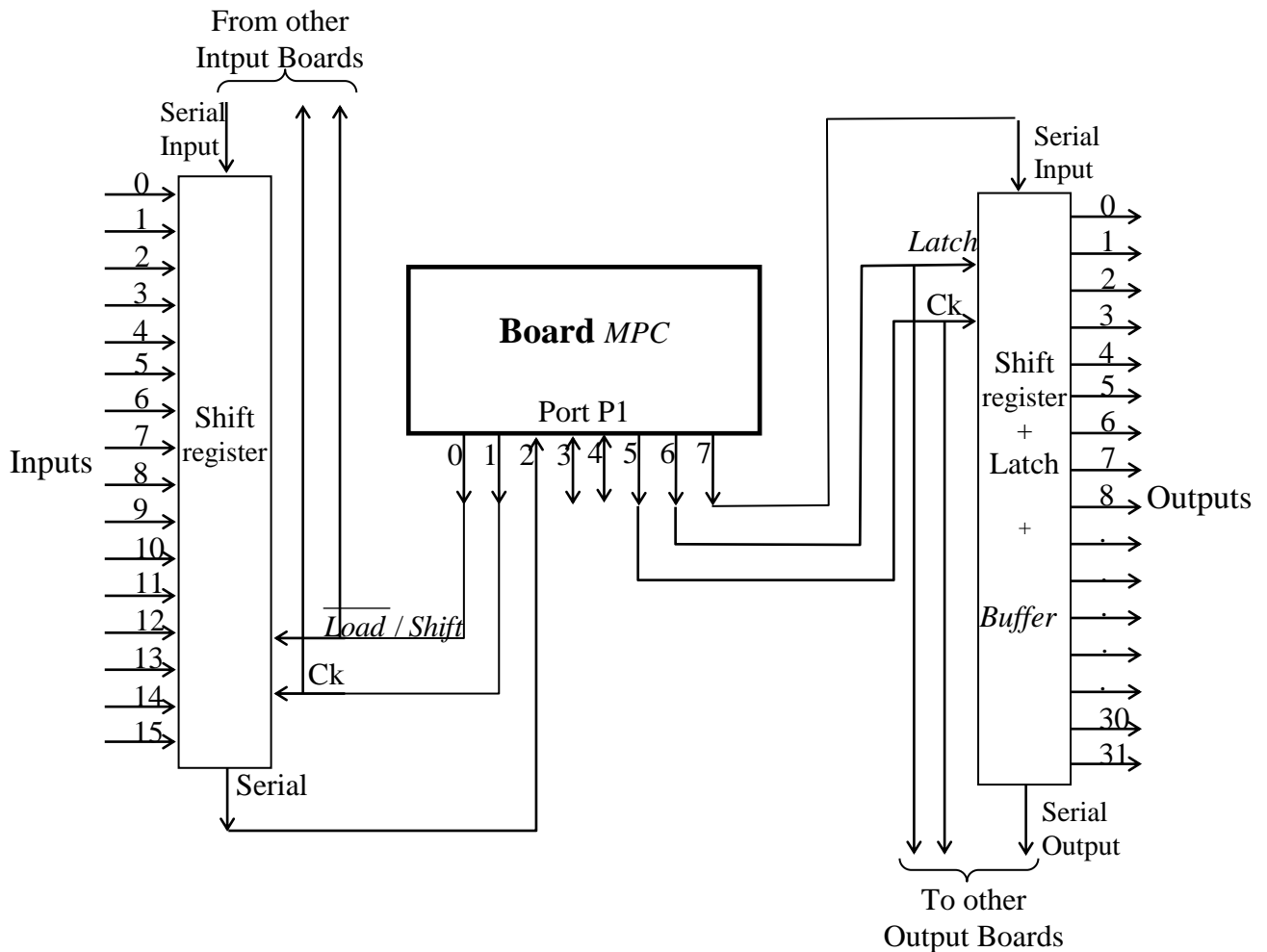
Cấu hình bộ nhớ trên được thực hiện bằng một mạch giải mã bộ nhớ và một bộ Jumper chọn cấu hình.

Bộ nhớ này được bảo vệ bằng một mạch phát hiện cúp điện. Ngay tại thời điểm điện thế cung cấp vừa tụt xuống (trước khi CPU ngưng hoạt động), mạch này phát hiện và khoá bộ nhớ lại, không cho CPU truy xuất nữa (để tránh trường hợp CPU ghi nhầm vào bộ nhớ

trong lúc quá độ do cúp điện). Sau đó dòng điện từ pin backup sẽ tiếp tục lưu giữ dữ liệu trong bộ nhớ cho đến khi có điện trở lại.



Hình 2: Các cấu hình bộ nhớ của bộ vi điều khiển đa năng (MPC)



Hình 3: Mở rộng ngõ vào / ra

2.2 Mở rộng ngõ vào / ra

8951 cung cấp 32 ngõ vào/ra lập trình được (port P0, P1, P2 và P3). Tuy nhiên khi mở rộng bộ nhớ ngoài, 16 ngõ của port P0 và P2 đã sử dụng cho bus địa chỉ và dữ liệu (đa hợp), 8 ngõ của port P3 được sử dụng cho mục đích khác (ngắt, counter, port nối tiếp,...)

[1,2]. Như vậy chỉ còn lại 8 ngõ vào/ra (P1) dành để điều khiển thiết bị. Do đó nhất thiết phải mở rộng ngõ các ngõ vào/ra để có thể điều khiển các hệ thống lớn.

Việc mở rộng phải thoả mãn điều kiện là có thể cắm thêm các các ngõ vào/ra dễ dàng mà không làm thay đổi hệ thống. Để có thể thực hiện điều này, phương pháp được lựa chọn là biến đổi song song thành nối tiếp (đối với ngõ vào) và từ nối tiếp sang song song (đối với ngõ ra) (Hình 3).

Board ngõ vào thực chất là một mạch ghi dịch vào song song ra nối tiếp. Trạng thái các ngõ vào được chốt lại khi có tín hiệu Load tác động (mức 0). Sau đó tín hiệu này trở lại mức cao để chuẩn bị dịch (Shift). Mỗi khi có 1 xung Ck thì một bit được dịch vào MPC. Các bit ứng với trạng thái các ngõ vào sẽ được chuyển thành song song bên trong MPC để xử lý. Mỗi board có 16 ngõ vào. Có thể mắc nối tiếp các board lại để có nhiều ngõ vào hơn nữa.

Board ngõ ra gồm mạch ghi dịch vào nối tiếp ra song song, mạch chốt và mạch buffer (driver). Dữ liệu cần xuất ra được đưa ra từng bit nối tiếp ứng với từng xung Ck. Các bit này được dịch vào thanh ghi dịch và chuyển thành dữ liệu song song. Khi thanh ghi dịch đã dịch xong tất cả các bit, tín hiệu chốt sẽ chốt dữ liệu đó vào mạch chốt và đưa ra các ngõ ra qua mạch buffer. Mạch buffer nâng dòng mỗi ngõ ra lên khoảng 350 mA với điện thế tối đa 70V DC để đủ khả năng điều khiển các thiết bị. Mỗi board có 32 ngõ ra, có thể ghép nối tiếp các board lại để có nhiều ngõ ra hơn nữa.

Với cách mở rộng ngõ vào / ra như trên, số lượng các ngõ vào / ra có thể mở rộng không giới hạn. Tuy nhiên khi số lượng ngõ vào /ra quá nhiều thì tốc độ đọc và xuất dữ liệu sẽ chậm hơn. Rất nhiều các ứng dụng điều khiển thực tế không cần tốc độ nhanh lắm (ví dụ thao tác bấm nút - điều khiển thì việc chậm hơn vài mili giây không thành vấn đề), do đó hệ thống vẫn đáp ứng tốt. Với những ứng dụng đòi hỏi tốc độ nhanh thì có thể sử dụng port P1 (8 ngõ vào/ ra) để điều khiển. Nếu không đủ, có thể mở rộng port theo kiểu vào /ra song song (ví dụ dùng 8255- 3 port, hoặc đơn giản là sử dụng các mạch chốt 74535).

3 XÂY DỰNG PHẦN MỀM

Trên phần cứng đã thiết kế, cần phải xây dựng một phần mềm điều khiển. Phần mềm này hoạt động như một hệ điều hành, có chức năng download chương trình người dùng (user program) và điều khiển việc thực hiện chương trình đó. Ngoài ra còn phải cung cấp các tiện ích cần thiết khác. Phần mềm có các tính năng mạnh sau đây:

- Soạn thảo và biên dịch chương trình người dùng.
- Download chương trình đã biên dịch xuống hệ thống.
- Thực hiện chương trình đã download (liên tục).
- Thực hiện từng lệnh chương trình.
- Thực hiện từng khối chương trình.
- Hiện thị tất cả các thông tin: giá trị các thanh ghi, ô nhớ nội, ô nhớ ngoại,... khi chạy từng bước (từng lệnh/ khối). Cho phép thay đổi nội dung thanh ghi hoặc ô nhớ.
- Hiện thị nội dung bộ nhớ.
- Từ PC có thể reset (mềm) được hệ thống dù đang chạy bất cứ chương trình nào.
- Có thể đặt chế độ tự động chạy user program mỗi khi bật điện (khi sử dụng MPC để điều khiển hệ thống thực tế).
- Cung cấp các hàm / chương trình con tiện ích khác,...

Phần mềm được thiết kế gồm có hai phần: Phần chạy trên máy tính PC và phần chạy trên hệ thống MPC. Hai phần này phải bắt tay nhau rất chặt chẽ để bảo đảm không có bất kì lỗi nào xảy ra. Dữ liệu được truyền qua lại giữa PC và MPC thông qua cổng COM1 (hoặc COM2). Tốc độ truyền dữ liệu được chọn là 9600bps, parity chẵn, bảo đảm truyền rất ít lỗi với tốc độ tương đối nhanh.

3.1 Phương pháp chạy từng bước

Khác với các loại CPU khác như Z80 [4,5], 8086,..., 8951 (và họ hàng của nó) không hỗ trợ việc điều khiển chạy từng bước bằng phần cứng [1,2]. Vì vậy để bắt 8951 chạy từng lệnh, chỉ có cách giải quyết duy nhất là dùng thủ thuật phần mềm. Để làm được việc này, cần phải tìm hiểu đáp ứng của 8951 khi một ngắt xảy ra.

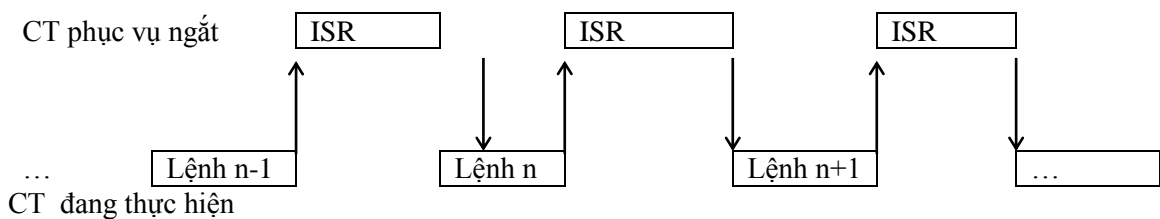
8951 có thể đáp ứng 6 tín hiệu ngắt: 2 ngắt bên ngoài INT0 và INT1, 2 ngắt timer0 và timer1 (ngắt khi timer tràn), hai ngắt truyền nhận nối tiếp (ngắt khi truyền xong và khi nhận xong 1 byte). Mỗi ngắt có thể chọn 2 mức ưu tiên. Có thể cho phép (Enable) hoặc cấm (Disable) từng ngắt [1,2].

Khi có một ngắt xảy ra và nếu được cho phép, 8951 sẽ thực hiện các công việc sau:

- Thực hiện xong lệnh hiện tại (đang thực hiện dở dang).
- Lưu giá trị thanh ghi con trỏ lệnh (PC) vào ngăn xếp.
- Lưu trạng thái các ngắt hiện tại.
- Khoá các ngắt có cùng mức ưu tiên ngắt với ngắt đó (các ngắt có cùng mức ưu tiên với ngắt đang được chấp nhận không thể gọi ngắt khi chưa phục vụ xong ngắt hiện tại).
- Địa chỉ của chương trình phục vụ ngắt (ISR-Interrupt Service Routine) được đặt vào thanh ghi PC.
- Chương trình phục vụ ngắt được thực hiện.

Sau khi đáp ứng xong yêu cầu của ngắt, chương trình phục vụ ngắt kết thúc bằng lệnh RETI (Return from Interrupt). Lệnh này sẽ lấy lại giá trị của PC trong ngăn xếp và phục hồi lại trạng thái các ngắt. Chương trình (CT) đang thực hiện (trước khi ngắt) được tiếp tục với lệnh kế tiếp. Nếu có ngắt xảy ra thì 8951 sẽ tiếp tục đáp ứng ngắt.

Như vậy 8951 sẽ liên tục nhảy qua nhảy lại giữa CT đang thực hiện và CT phục vụ ngắt nếu ngắt xảy ra nhiều lần. Nếu ngắt xảy ra liên tục (phục vụ ngắt vừa xong hoặc chưa xong thì đã có ngắt xảy ra nữa), thì 8951 vẫn thoát khỏi CT phục vụ ngắt ISR khi phục vụ xong, trở về CT đang thực hiện để thực hiện 1 lệnh của CT đó, sau đó mới thực hiện ISR (Hình 4).



Hình 4: Quá trình thực hiện lệnh của 8951 nếu có ngắt liên tục

Hình 4 chính là cơ chế chạy từng bước bằng thủ thuật phần mềm. Ở phần cứng, chân ngắt INT0 được nối ở mức luôn tác động. Khi nào cần chạy từng bước chương trình người dùng (user program), chỉ cần cho phép ngắt INT0. Khi đó, cứ mỗi khi thực hiện xong một lệnh của user program, 8951 sẽ nhảy đến thực hiện CT phục vụ ngắt ISR. Chương trình này sẽ lập vòng để chờ sự điều khiển từ máy tính. Khi nào nhận được yêu cầu chạy

tiếp lệnh kế thì thoát khỏi ISR. Như thế từng lệnh của user program sẽ được thực hiện theo sự điều khiển của người sử dụng (từ máy tính PC).

Để chạy từng đoạn lệnh, ISR phải xét xem lệnh kế tiếp (lệnh sắp thực hiện của user program) có phải là điểm dừng không (đã đặt trước, được qui định là lệnh NOP- no operation). Nếu không phải thì thoát ngay khỏi ISR. Nếu phải thì đợi sự điều khiển từ máy tính. Như vậy đối với người điều khiển, 8951 ‘hình như’ dừng user program tại những điểm mong muốn.

Để hệ thống thực sự tiện lợi, trong quá trình chạy từng bước (từng lệnh/ từng đoạn lệnh), phải hiển thị tất cả các nội dung cần thiết như giá trị các thanh ghi, ô nhớ, port vào/ra, các bit trạng thái,... Ngoài ra người sử dụng còn có thể muốn thay đổi nội dung một thanh ghi hay ô nhớ nào đó, tiếp tục chạy từng bước hay thoát khỏi user program,... Toàn bộ các công việc này sẽ được giải quyết bên trong chương trình phục vụ ngắt ISR (gọi là chương trình ControlRun). Lưu đồ và giao thức cơ bản của công việc điều khiển chạy từng bước được thể hiện trên hình 7.

3.2 Chương trình chính trên PC và trên MPC

Hai chương trình này giao tiếp nhịp nhàng với nhau, điều khiển hệ thống thực hiện tất cả các thao tác mà người sử dụng cần (Hình 5, 6). Người dùng có thể soạn thảo chương trình, biên dịch, download xuống MPC. Có thể chạy thử chương trình bằng các chức năng chạy từng lệnh, từng đoạn lệnh. Người sử dụng có thể tìm ra lỗi phần mềm bằng cách quan sát dữ liệu lúc chạy từng bước. Nếu dùng MPC để điều khiển thiết bị, phần cứng khác, có thể tìm lỗi phần cứng bằng cách cho chương trình dừng lại ở những điểm thích hợp và kiểm tra phần cứng tại những điểm dừng đó. Khi tất cả đều đã hoạt động tốt, để đưa hệ thống vào vận hành chính thức, chỉ cần chọn chế độ tự động chạy chương trình.

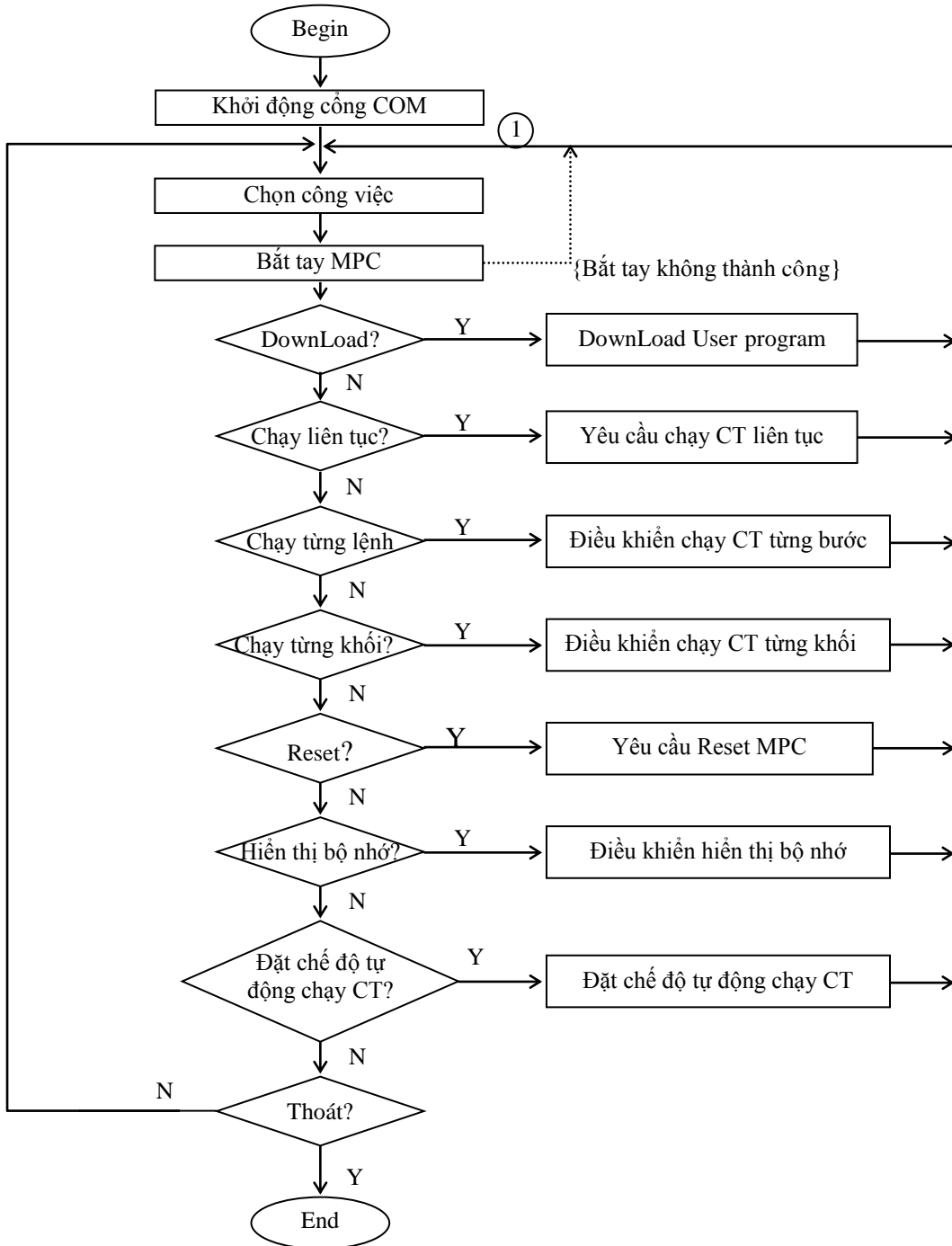
4 KẾT LUẬN

Hệ thống MPC cho phép những người thiết kế các ứng dụng điều khiển có thể phát triển các ứng dụng rất nhanh. Về cơ bản, phần cứng của MPC đáp ứng được hầu hết các yêu cầu về điều khiển thường gặp. Các ngõ vào/ra có thể mở rộng gần như vô hạn. Bộ nhớ cấu hình linh hoạt, dung lượng khá lớn cho các ứng dụng điều khiển, được bảo vệ tốt nên có thể lưu trữ lâu dài.

Chạy chương trình từng bước trên vi điều khiển 8951 là điều không thể thực hiện được (so với Z80 hay 8086). Tuy nhiên với giải pháp phần mềm, điều này đã thực hiện được rất tốt. Điều này rất quan trọng vì nó cho phép người thiết kế nhanh chóng tìm ra các lỗi ở phần mềm cũng như ở phần cứng, giảm thời gian thiết kế đi rất nhiều. Nó cũng rất hữu dụng với những người muốn tìm hiểu về vi điều khiển, đặc biệt là các sinh viên, vì họ có thể học và thực hành một cách trực quan trên hệ MPC.

TÀI LIỆU THAM KHẢO

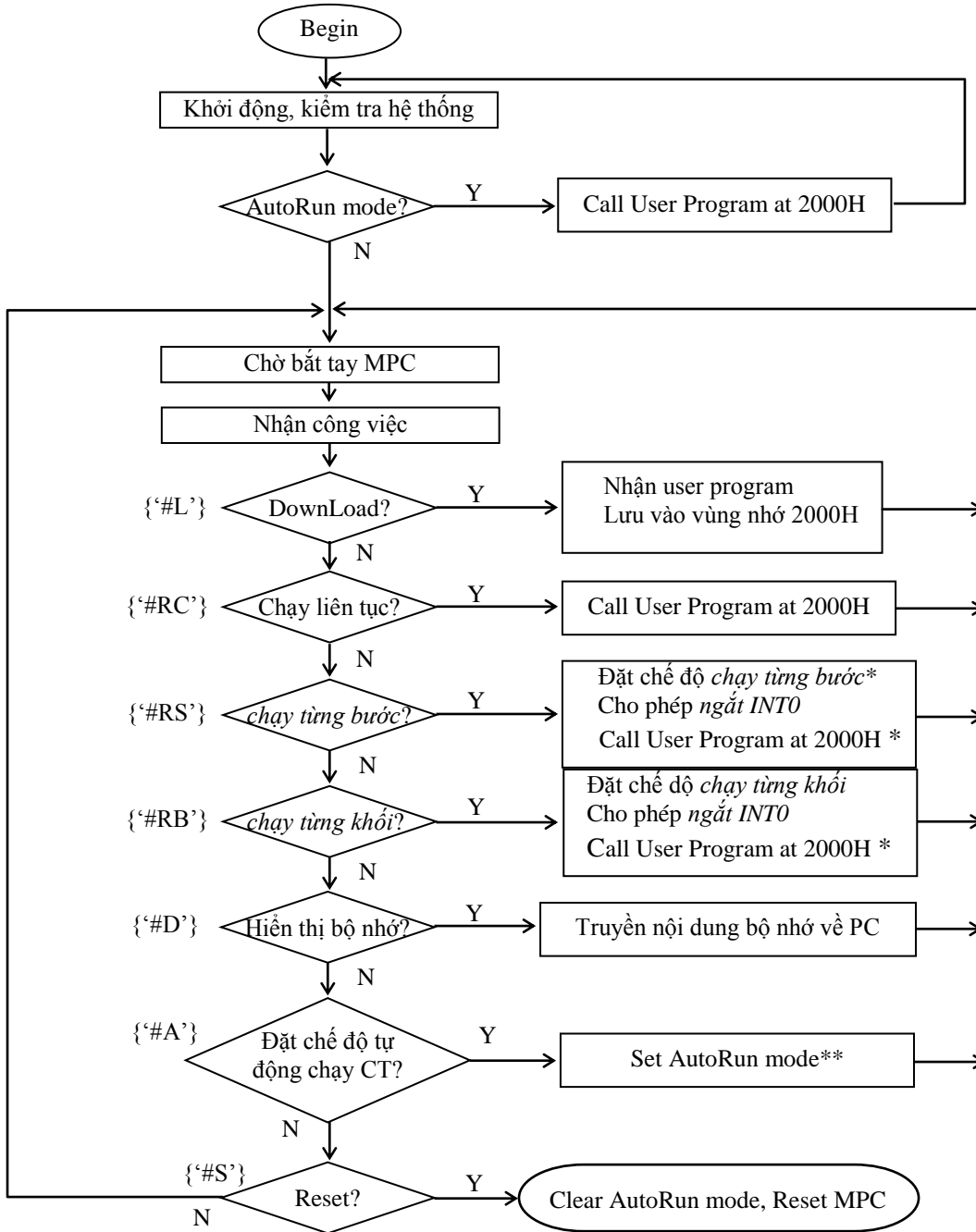
- Scott MacKenzie University of Guelph, THE 8051 MICROCONTROLLER FAMILY, PRENTICE HALL, 1997
- AT8951 Datasheet, http://www.atmel.com/dyn/resources/prod_documents/doc0265.pdf, truy suất 12/2001
- Dương Minh Trí, SỔ ĐỒ CHÂN LINH KIỆN, nhà xuất bản Khoa Học Kỹ Thuật, năm 2001
- Lương Văn Sơn, GIÁO TRÌNH VI XỬ LÝ, Đại học Cần Thơ, năm 2001
- Lương Văn Sơn, GIÁO TRÌNH THỰC HÀNH VI XỬ LÝ, Đại học Cần Thơ, năm 2001
- Trần Thanh Hùng, Trần Hữu Danh, Báo cáo tổng kết đề tài nghiên cứu: BỘ VI ĐIỀU KHIỂN ĐA NĂNG, Đại học Cần Thơ, 2003



Hình 5: Lưu đồ chương trình chính trên PC

QUI ƯỚC CÁC LỆNH YÊU CẦU TỪ PC

Khi PC (Personal Computer) muốn MPC thực hiện một công việc nào đó, nó sẽ truyền xuống MPC một số kí tự mã hóa công việc bắt đầu bằng kí tự '#'.
 '#S': reSet MPC
 '#D': Dump Memory
 '#A': Autorun user program when power up
 '#L': downLoad user program
 '#RC':Run user program Continuous
 '#RS':Run user program by Step
 '#RB':Run user program by Block

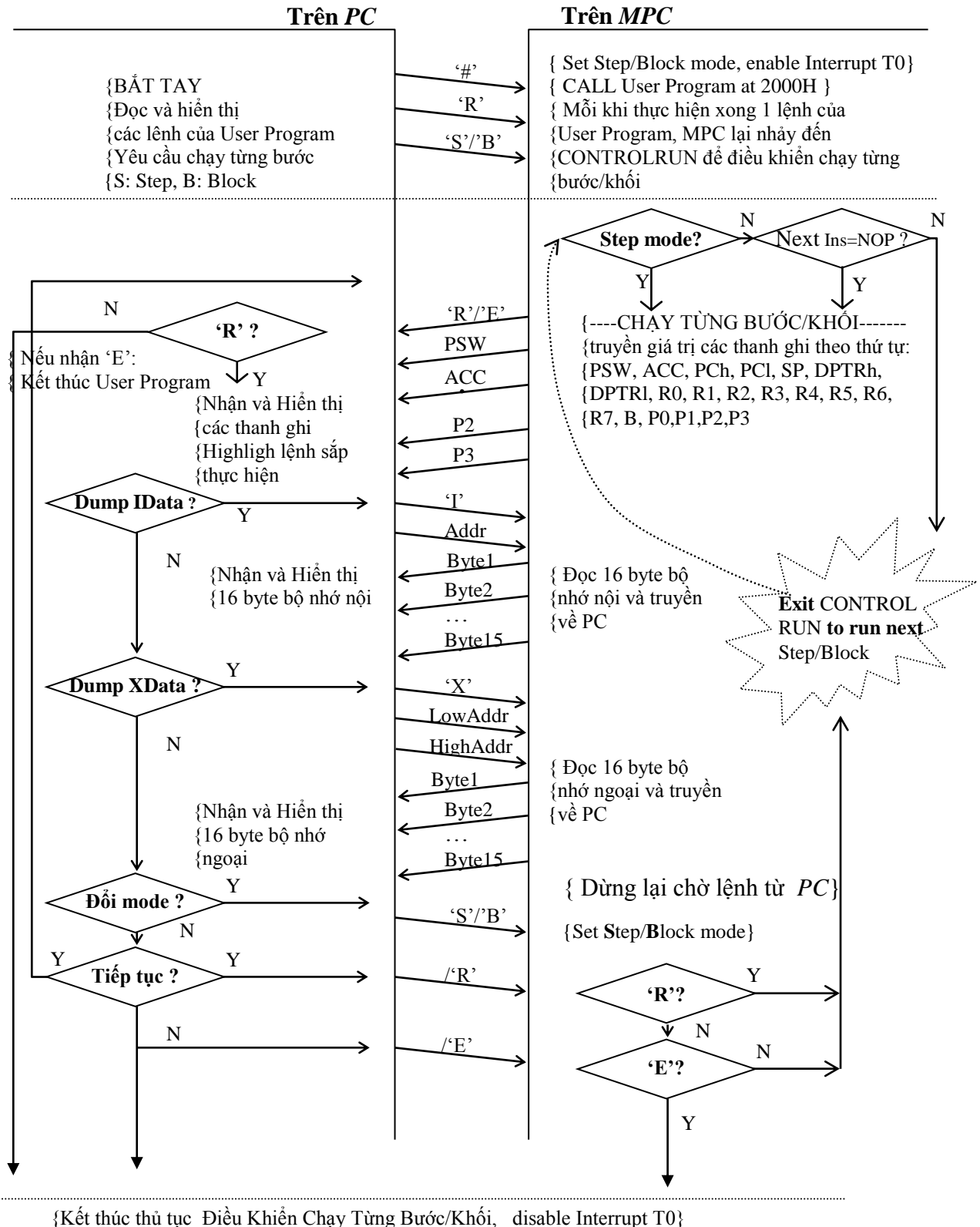


Hình 6: Lưu đồ chương trình chính trên MPC

Ghi chú:

* Nguyên tắc chạy từng bước: Chân ngắt INT0 ở phần cứng được nối ở mức luôn tác động. Mỗi khi muốn chạy từng bước, chỉ cần cho phép ngắt INT0. Sau mỗi khi thực hiện xong một lệnh của chương trình đang chạy (User Program), MPC sẽ nhảy đến chương trình phục vụ ngắt INT0 (CT CONTROLRUN). Chương trình này sẽ truyền về PC giá trị các thanh ghi và ô nhớ mà PC cần, sau đó dừng lại chờ PC. Nếu PC ra lệnh chạy tiếp lệnh kế, MPC sẽ thoát khỏi CT phục vụ ngắt để thực hiện lệnh kế tiếp của User Program.

** Đặt chế độ tự động chạy User Program (đã download trước đó) mỗi khi bật nguồn. Chỉ có thể xoá chế độ này bằng cách reset mền.



Hình 7: Lưu đồ và giao thức điều khiển chạy từng bước (ControlRun)

Ghi chú:

- Dump: Hiển thị, IData: Internal data- Bộ nhớ nội, XData: External Data- Bộ nhớ ngoại
- Addr: Address- Địa chỉ ô nhớ, LowAddr: byte thấp địa chỉ, HighAddr: byte cao địa chỉ
- Next Ins: Next Instruction- Lệnh kế tiếp, lệnh NOP: no operation- lệnh không làm gì cả