

SỬ DỤNG MAINBOARD MÁY TÍNH 386, 486 VÀO VIỆC GIẢNG DẠY VÀ ỨNG DỤNG KỸ THUẬT

Nguyễn Hữu Phương - Nguyễn Văn Thiện

Trường Đại Học Khoa Học Tự Nhiên

(*Bài nhận ngày 10/02/1999*)

TÓM TẮT : Do đặc tính của bộ vi xử lý và các đặc điểm của bo mạch chính hay bo chính, thường để nâng cấp máy vi tính một cách đáng kể chúng ta phải đổi bo chính trong lúc các thành phần khác của máy (vỏ và bộ nguồn, card video, màn hình, v.v) vẫn được sử dụng tiếp tục. Các bo chính vẫn còn đó rất nhiều khả năng nhất là cho các mục đích sử dụng ở cấp thấp hơn là máy tính nguyên bộ.

Do đó cần nghiên cứu sử dụng lại các bo chính 386, 486 mà gần như đồ phế thải khi chúng ta chuyển sang dùng các bo cao cấp hơn. Chúng ta thiết kế thêm phần cứng và viết phần mềm cần thiết để chuyển bo chính thành các bộ Kit tương đối đa năng cho các ứng dụng giảng dạy và kỹ thuật. Kết quả đã đạt được như mong muốn.

Bài báo tóm lược đề tài nghiên cứu cấp Bộ mã số B96-18-18 đã được nghiệm thu.

I. MỞ ĐẦU:

Sự phát triển nhanh chóng của công nghệ vi mạch điện tử và công nghệ thông tin khiến chúng ta phải nâng cấp hoặc đổi mới máy vi tính khá thường xuyên. Thành phần chính của máy vi tính là bo mạch chính hay **bo chính** (Mainboard) còn gọi bo mẹ hay bo hệ thống. Bộ phận trung tâm của bo chính là **bộ vi xử lý (VXL)** mà cũng được gọi bộ xử lý. Khi chúng ta thay đổi bo chính lên cấp cao hơn thì các bo 386, 486 vẫn còn có thể sử dụng cho giảng dạy cấu trúc VXL, phần nào đó cấu trúc máy tính, và cho các ứng dụng kỹ thuật (đo lường, điều khiển, v.v.). Việc giảng dạy cấu trúc VXL và lập trình ngôn ngữ máy trên một Kit 386 hay 486 có nhiều thuận lợi hơn là trên Kit Z80 hay 6802 vì 386, 486 nằm trong họ VXL 80x86 rất phổ biến của hãng Intel, vì chúng là các VXL 16 bit, 32 bit, v.v.

Bài này trình bày việc nghiên cứu sử dụng lại bo chính 386, 486 cho các mục đích nêu trên. Chúng ta không sửa đổi gì trên bo chính mà chỉ thiết kế thêm phần cứng ở dạng các Card cắm vào **khe mở rộng** (Expansion Slot) và viết các phần mềm giao tiếp và điều khiển cần thiết. Thật ra trong phạm vi bài này chúng ta sử dụng các bo chính 386, 486 gần như là bo mạch 16 bit 8086 nhưng dĩ nhiên với tốc độ nhanh hơn nhiều và rất nhiều khả năng tiềm tàng khác (Bus dữ liệu 32 bit, không gian bộ nhớ ảo, đồng xử lý toán, v.v.).

Vì cấu trúc của 486 bao gồm 386 nên sau đây chúng ta chỉ nói VXL 486 và bo chính 486. Bộ Kit xây dựng từ bo chính 486 được gọi **CKIT-486**.

II. CHỨC NĂNG ROM SCAN CỦA ROM BIOS.

Đối với bo chính dùng VXL của Intel, khi được cấp điện các VXL hoạt động ở chế độ thực (Real mode), chương trình được thực hiện từ ô nhớ 0F000h ÷ 0FFF0h nằm trong ROM BIOS. Tại đây một lệnh **JMP FAR** sẽ chuyển quyền điều khiển đến một đoạn chương trình

định sẵn trong ROM. Đoạn chương trình này kiểm tra VXL, các thanh ghi v.v., nếu có lỗi máy sẽ dừng lại mà không thông báo lỗi. Nếu việc kiểm tra tốt, máy sẽ so sánh để tìm lỗi trong ROM. Bộ điều khiển DMA, bộ điều khiển ngắt, bộ nhớ hệ thống (DRAM), ... sẽ được kiểm tra và khởi tạo. Các biến BIOS và bảng Vectơ ngắt mặc định cũng được khởi tạo. Đây là quá trình POST.

Tiếp theo đoạn chương trình trong BIOS kiểm tra các ROM ngoài (tức các ROM gắn trên các Card cắm vào khe mở rộng của bo chính). Đây là chức năng **ROM Scan** của ROM BIOS. Các ROM ngoài có thể hiện hữu giữa các địa chỉ 0C000h ÷ 0000h đến 0E000h ÷ 0000h. Chương trình trong Rom BIOS sẽ quét các khối 2 Kbyte nhằm tìm ra một ROM có giá trị để trao quyền điều khiển cho nó. Một Rom có giá trị được định nghĩa như sau:

Offset Kích thước Nội dung

+0 1 Byte (Byte 0) 055h

+1 1 Byte (Byte 1) 0AAh

+2 1 Byte (Byte 2) Dung lượng của ROM/ 512

+3? Các Byte sau là đoạn mã thi hành (thông thường là một lệnh nhảy gần đến một đoạn mã khởi tạo).

Chương trình BIOS nhận ra các giá trị trên, nó sẽ kiểm tra tổng kiểm (Checksum) bằng cách cộng giá trị của tất cả các Byte nằm trong ROM. Giá trị này phải là bội số của 100h. Nếu thỏa, chương trình BIOS sẽ xác nhận đây là một ROM hợp lệ và trao quyền điều khiển cho nó bằng một lệnh CALL FAR đến Offset 0003. Tại đây đoạn chương trình trong ROM sẽ được thực thi. Chúng ta có thể gọi ROM ngoài này là **ROM Boot**.

Sau khi thực hiện xong đoạn chương trình trong ROM Boot chúng ta có thể trả quyền điều khiển lại cho BIOS bằng một lệnh RET FAR hoặc có thể giành quyền điều khiển hoàn toàn và chạy **chương trình Monitor** tự viết nạp chung vào ROM Boot để điều khiển các hoạt động xuất nhập trên các mạch ta lắp vào khe mở rộng mà không cần thay đổi bất cứ gì trên bo chính về phần cứng cũng như phần mềm. Đây chính là cách mà chúng ta sẽ làm.

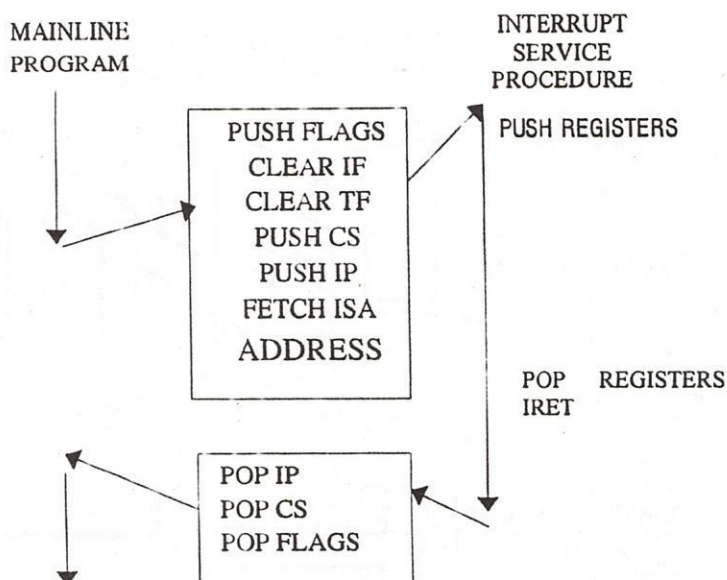
III. NGẮT.

Trong CKIT-486 bộ VXL giao tiếp với các bộ phận ngoại vi theo phương pháp ngắt (Interrupt) nên phần này trình bày tóm lược về ngắt.

Có hai loại ngắt: ngắt mềm đến từ các chỉ thị INT hoặc một vài điều kiện lỗi (Error) trong chương trình, và ngắt cứng đến từ một tín hiệu điện bên ngoài. Trên bo chính đã tích hợp sẵn các vi mạch điều khiển các quá trình ngắt. Một số đường ngắt được sử dụng cho chính hệ thống (bộ đếm, bàn phím, v.v.), còn một số đường ta có thể dùng thông qua các khe mở rộng của bo chính.

Để đáp ứng một ngắt, VXL ngừng thi hành chương trình hiện hành của nó và gọi một chương trình con phục vụ ngắt. Một chỉ thị IRET tại cuối chương trình con sẽ trả quyền điều khiển lại cho chương trình chính và thực hiện tiếp chương trình này.

Ở cuối mỗi chu kỳ chỉ thị, VXL kiểm tra nếu có một ngắt được yêu cầu nó sẽ đáp ứng bằng cách thực hiện các bước ở hình 1. Khi VXL thực hiện một chỉ thị CALL FAR đến một chương trình con, nó đặt giá trị mới vào thanh ghi Code Segment (CS) và thanh ghi Instructions Pointer (IP) từ 4 ô nhớ. KByte đầu của bộ nhớ từ 00000h đến 003FFh là nơi cất các địa chỉ bắt đầu của chương trình con phục vụ ngắt. Từ KByte đó, 4 Byte chứa địa chỉ của chương trình con phục vụ ngắt sẽ được truy cập và cất vào CS và IP. Địa chỉ (4 Byte tức 2 Word) bắt đầu của một chương trình con phục vụ ngắt thường được

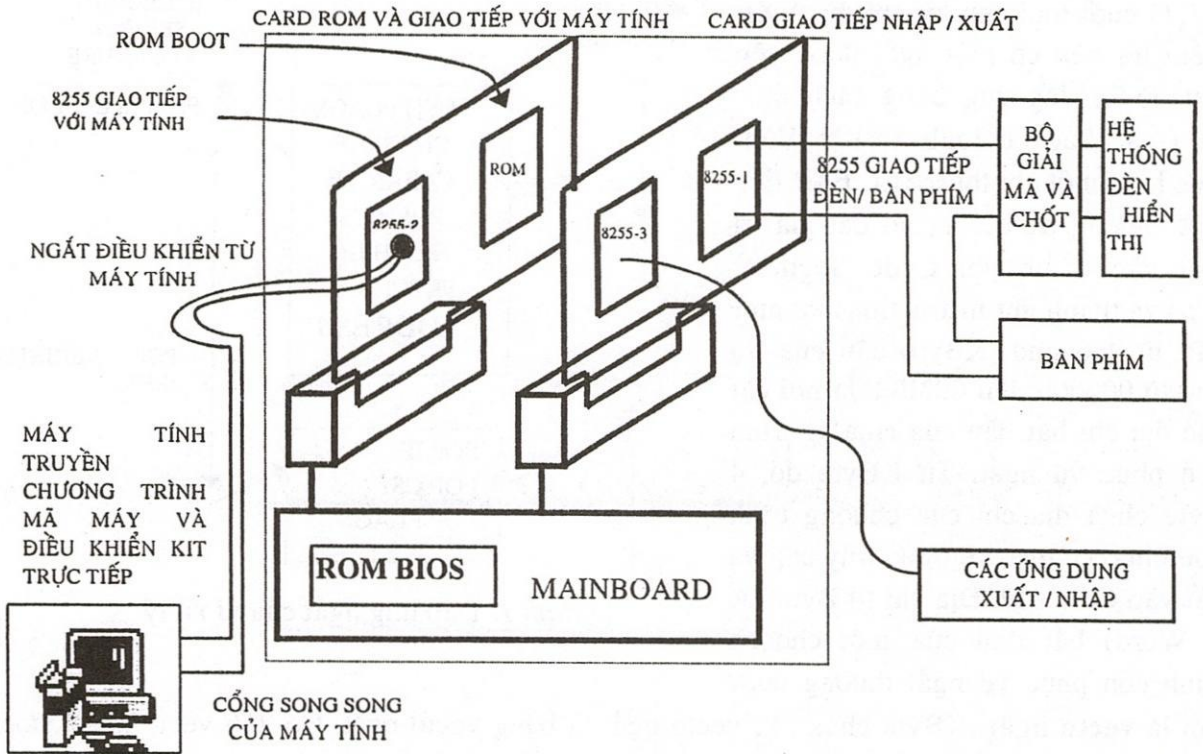


Hình 1: Đáp ứng ngắt của vi xử lý

gọi là **vectơ ngắt**. KByte chứa các vectơ ngắt là bảng vectơ ngắt. Có 256 vectơ ngắt được đánh số từ 0 đến 255. Hãng Intel gọi chúng là các **kiểu ngắt** (Interrupt Type) gồm: ngắt chia cho không – kiểu 0, ngắt bước đơn (Single Step) – kiểu 1, ngắt không che được (Nonmaskable) – kiểu 2, ngắt điểm dừng (Breakpoint) – kiểu 3, ngắt tràn (Overflow) – kiểu 4, các ngắt INTR có kiểu từ 0 đến 255.

IV. CÁC MẠCH CỦA CKIT-486.

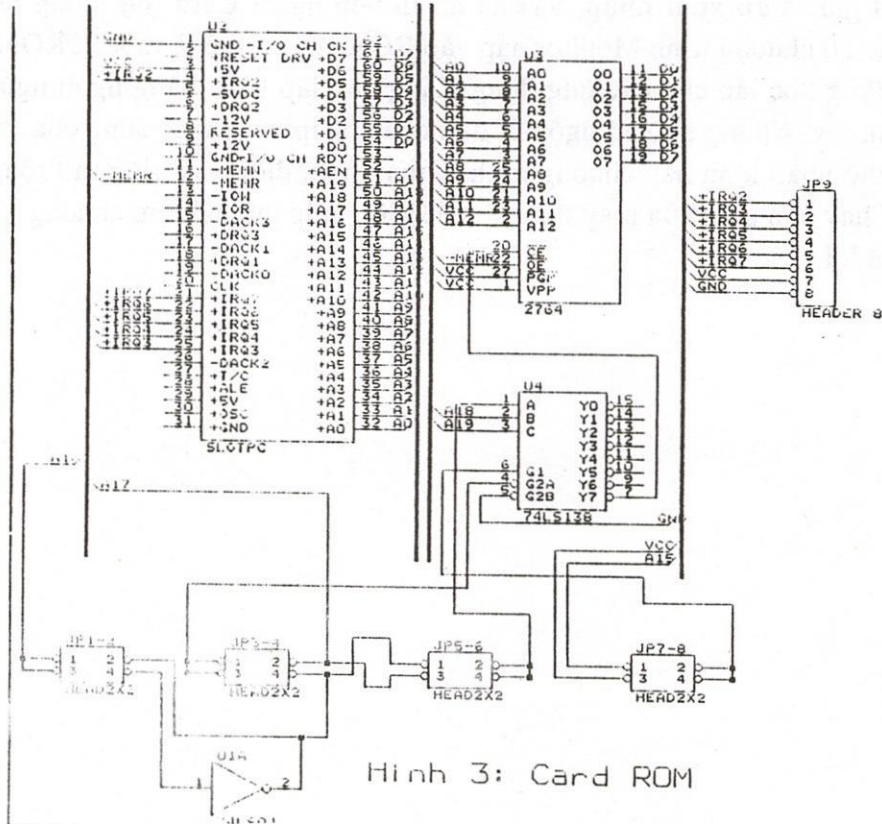
Hình 2 là sơ đồ khối của CKIT-486. Chúng ta thấy có **card ROM** và **giao tiếp với máy tính, card giao tiếp xuất/nhập**, và các mạch bên ngoài Card (hệ thống đèn hiển thị, bàn phím). Nhờ có chương trình Monitor nạp vào ROM mà cụ thể là một EPROM 2764, CKIT có thể hoạt động độc lập cho các ứng dụng giảng dạy lập trình bằng ngôn ngữ máy, đo lường, điều khiển, v.v. Nhưng nhờ có ngõ nối đến cổng nhập/xuất song song của một máy tính nên CKIT có thể nhận lệnh hay chương trình từ máy tính, điều này giúp mở rộng tầm hoạt động của CKIT hay bản thân của máy tính và, đặc biệt, giúp thử nghiệm chương trình Monitor một cách thuận lợi.



Hình 2: Sơ đồ khối CKIT-486

Card ROM.

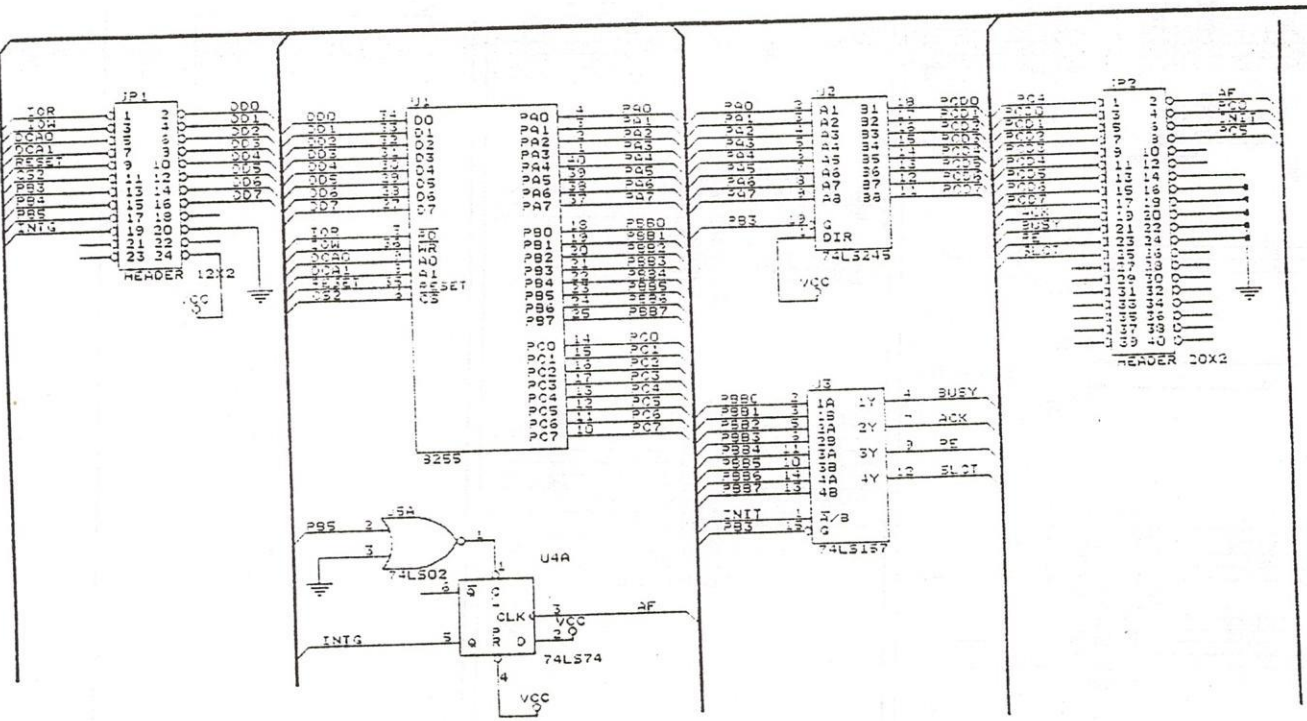
Card Rom giao tiếp với bo mạch chính qua SLOTPC (hình 3). Vi mạch chính trên card là EPROM 2764 (8KBx8) chứa chương trình Boot máy và chương trình Monitor điều khiển kit. EPROM phải nằm đúng ngay một trong các địa chỉ mà quá trình ROM Scan sẽ tìm tới. Việc giải mã địa chỉ được đảm nhiệm bởi vi mạch 74LS138. Trên các ROM này còn có vi mạch PPI 8255-2 để giao tiếp với cổng song song của máy tính (hình 2) mà sẽ được trình bày tiếp theo đây.



Hình 3: Card ROM

Mạch giao tiếp với cổng song song của máy tính.

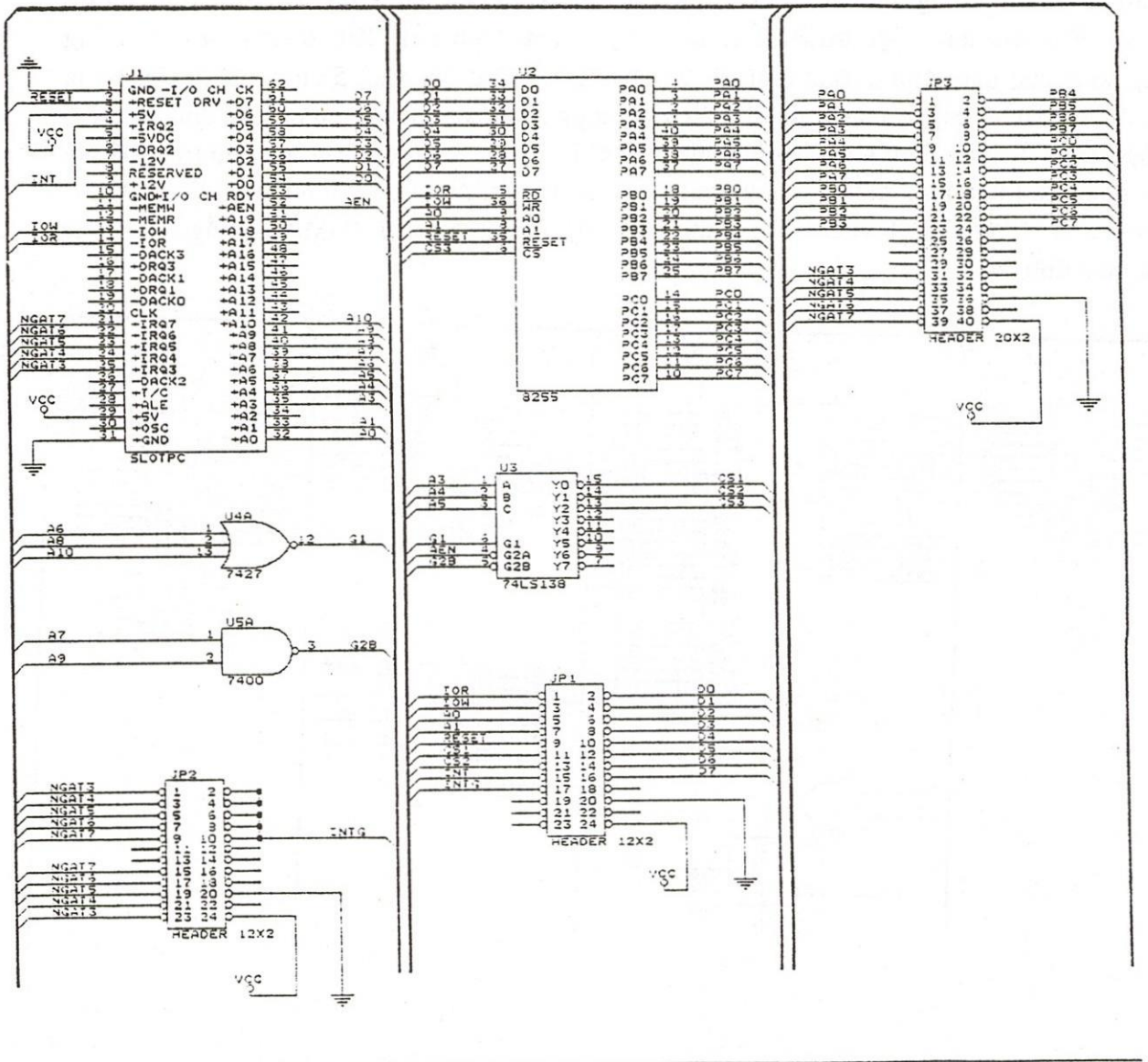
Khi sử dụng CKIT-486 độc lập với máy tính thì nếu muốn thực thi một chương trình chúng ta phải gõ từng địa chỉ và mã máy trên bàn phím của Kit. Như vậy sẽ mất nhiều thì giờ và dễ nhầm lẫn. Việc truyền File mã máy từ máy tính sang Kit sẽ tiện lợi hơn. Cổng song song của máy tính là một đầu nối 25 chân gồm 12 chân xuất, 5 chân nhập, còn lại là các đường đất. Trong vi mạch điều khiển cổng song song có 3 thanh ghi để xác định các ngõ nhập/xuất của cổng. Sau khi xác định được địa chỉ của cổng, chúng ta có thể lập trình để gửi hay nhận tín hiệu bằng một số chỉ thị của ngôn ngữ lập trình ví dụ với hợp ngữ là IN, OUT, ..., với C là INPORT, OUTPORT Hình 4 là mạch giao tiếp card ROM với cổng song song của máy tính.



Hình 4: Mạch giao tiếp với cổng song song của máy tính.

Card giao tiếp nhập/xuất.

Card này (hình 2) một mặt giao tiếp với bo chính qua SLOTPC một mặt giao tiếp với các mạch bên ngoài (hệ thống đèn hiển thị, bàn phím, v.v.). Trên card chứa các vi mạch: 74LS138 giải mã địa chỉ cho các PPI 8255 trong Kit, PPI 8255-1 để giao tiếp với hệ thống đèn hiển thị và bàn phím, PPI 8255-3 cho các ứng dụng nhập/xuất (trong đo lường, điều khiển). Hình 5 chỉ một phần của Card này (SLOTPC, 8255-1, 74LS138, v.v.).



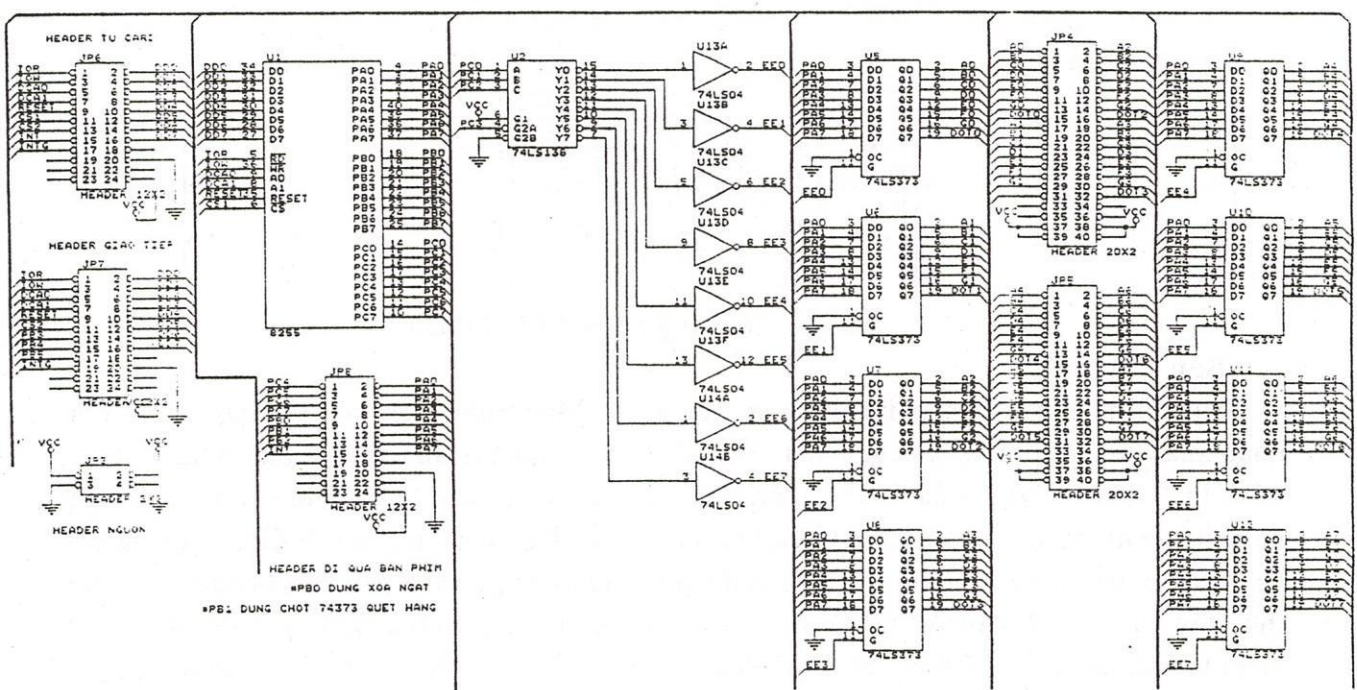
Hình 5: Một phần của Card giao tiếp nhập/xuất.

Hệ thống đèn hiển thị.

Hình 6 và 7 là một khối nhưng được tách làm hai phần. Hình 6 là mạch giải mã cho hệ thống đèn LED 7 đoạn để hiển thị địa chỉ, dữ liệu và chỉ thị. Các vi mạch 74LS373 là các chốt. Hình 7 là mạch gắn 8 đèn LED 7 đoạn, các chân của đèn được đưa vào hai Connector JP1 và JP2 để đến mạch hình 6.

Mỗi ngõ ra của chốt điều khiển một đoạn của LED 7 đoạn sáng hay tắt. LED là loại Anod chung nên muốn đoạn sáng ngõ ra Q của chốt phải ở mức thấp, ngược lại là mức cao.

Muốn LED 7 đoạn hiển thị một số hay ký tự chúng ta phải đặt lên các chân $Q_0 \div Q_7$ của chốt mã sáng của đèn. Để điều khiển việc chốt/truyền dữ liệu ở các chốt chúng ta dùng mạch giải mã 3 sang 8 đường 74LS138.

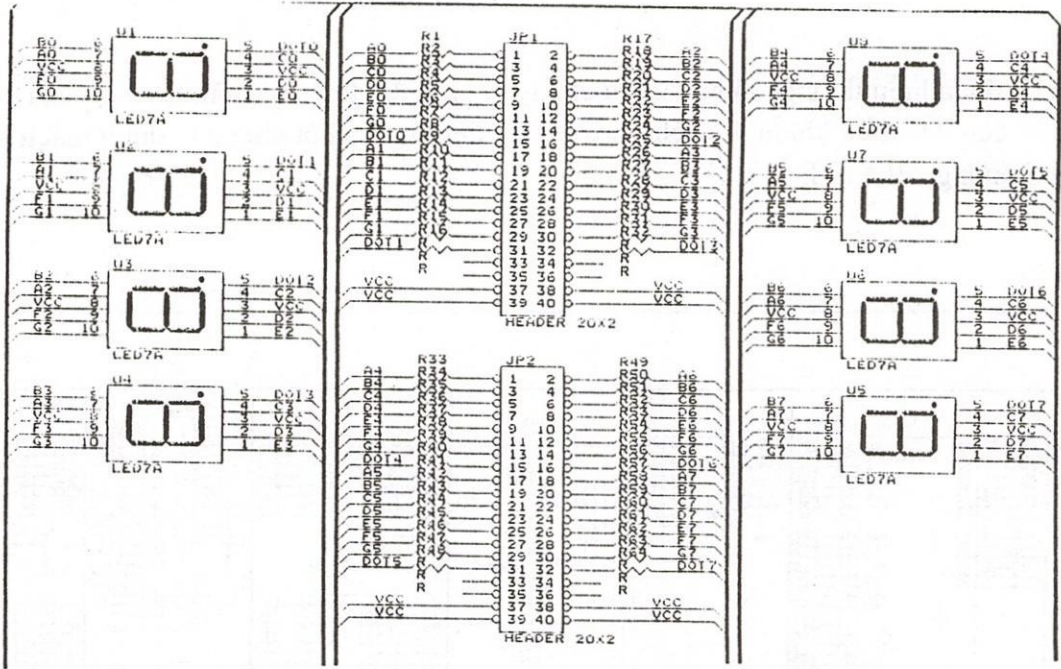


Hình 6 : Mạch giải mã hệ thống đèn LED 7 đoạn để hiển thị

Muốn gửi dữ liệu đến một đèn chúng ta làm các bước sau:

- Gửi mã sáng của đèn ra Port A của 8255-1.
- Đặt các bit lên các ngõ vào A, B, C của 74LS138 để đưa ngõ ra Y_x xuống thấp để chọn đèn ($Y_0 = 0$ chọn đèn LED 0, $Y_1 = 0$ chọn đèn LED 1, v.v.). Thủ tục WRITE-CHAR trong chương trình Monitor (xem sau) phục vụ việc gửi dữ liệu ra giàn đèn với thanh ghi AH chứa mã sáng của đèn và thanh ghi AL chứa vị trí của đèn muốn sáng.

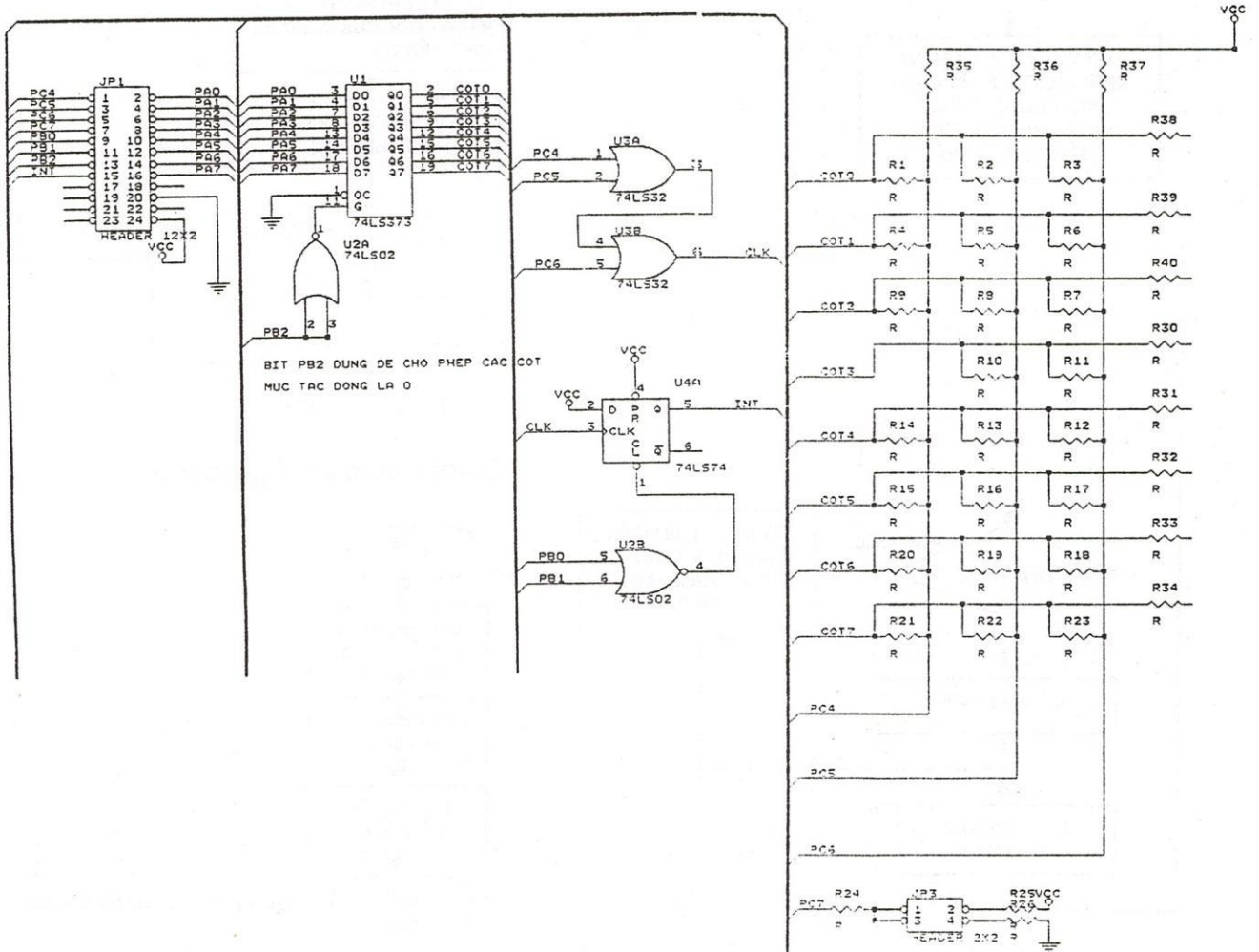
Việc dùng nhiều mạch chốt trong mạch giải mã hệ thống đèn làm mạch rườm rà tuy rằng về mặt giải thích hoạt động cho mục đích giảng dạy thì đơn giản vì mạch chỉ dùng các linh kiện quen thuộc. Cách khác hiện đại và có thể gọn hơn là dùng một bộ vi điều khiển (Microcontroller) như 8051 để phụ trách chức năng giải mã hệ thống đèn.



Hình 7: Hệ thống 8 đèn LED 7 đoạn

Bàn phím.

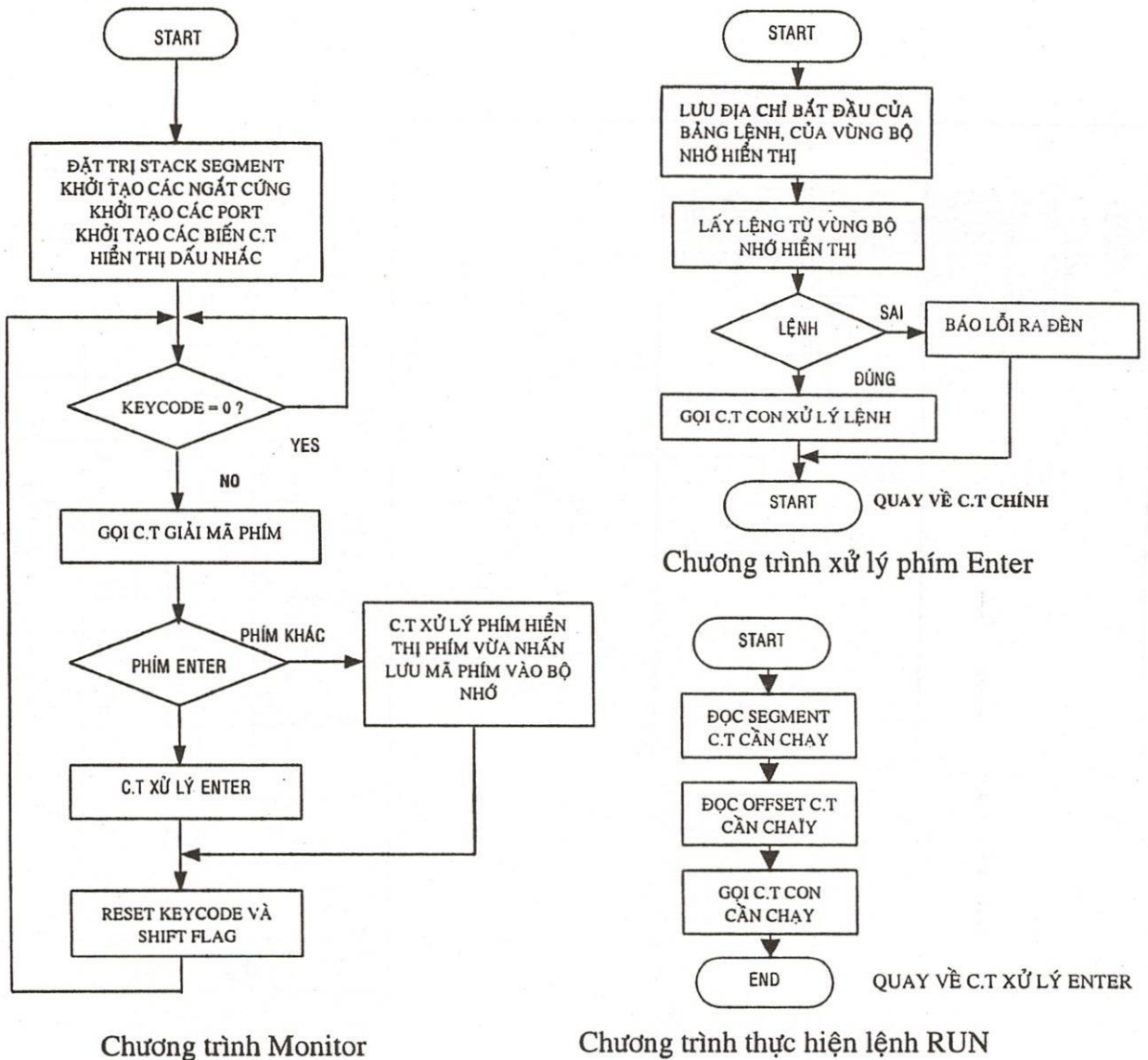
Hình 8 là mạch bàn phím dùng cơ chế ngắt. Mạch này liên kết với card giao tiếp nhập/xuất (hình 5) qua Connector JP1. Khi có một phím được nhấn thì 1 trong 3 hàng (PC4, PC5, PC6 thuộc Port C của 8255-1) lên mức cao, tạo một xung từ thấp lên cao ở ngõ ngắt cứng ở khe mở rộng của bo chính (ở đây sử dụng ngắt IRQ2), lúc bấy giờ VXL sẽ gọi chương trình con phục vụ ngắt. Nhiệm vụ của chương trình này là giải mã phím vừa nhấn là ở hàng thứ mấy và cột thứ mấy. Vị trí hàng và cột sẽ được chương trình con chuyển đổi thành mã của phím đã ấn, nếu mã là ký tự thì sẽ được hiển thị lên màn hình, còn nếu là chức năng thì chương trình Monitor sẽ thực hiện chức năng đã được định sẵn. Sau khi thực hiện xong ngắt, tại mỗi chương trình con phục vụ ngắt sẽ có lệnh gửi ra Port C của 8255-1 các giá trị thấp tại PB0 và PB1 và chờ một ngắt khác.



Hình 8: Bàn phím dùng cơ chế ngắt

V. CÁC CHƯƠNG TRÌNH.

CKIT-486 gồm phần cứng như mô tả ở trên và chương trình Monitor nạp vào EPROM để điều khiển Kit. Tuy nhiên Kit cũng có thể được lập trình và điều khiển từ một máy tính nhờ qua cổng song song.



Hình 9: Lưu đồ chương trình Monitor và một số chương trình con.

Chương trình Monitor

Hình 9 là lưu đồ chương trình Monitor. Ngoài ra còn nhiều chương trình con có các chức năng riêng như xử lý phím ENTER, RUN để thi hành, CLRS xóa các đèn, CLRS_ để xóa các đèn nhưng hiển thị dấu nhắc các dòng, WRITE_CHAR viết một ký tự ra đèn, WRITE_STRING viết một chuỗi ra đèn, KEY_INT phục vụ ngắt bàn phím, v.v.

Hình 10 là lưu đồ chương trình để máy tính điều khiển và truyền File đến Kit. Ngoài ra trong Kit còn có chương trình con phục vụ ngắt đến từ máy tính.

VI. KẾT QUẢ.

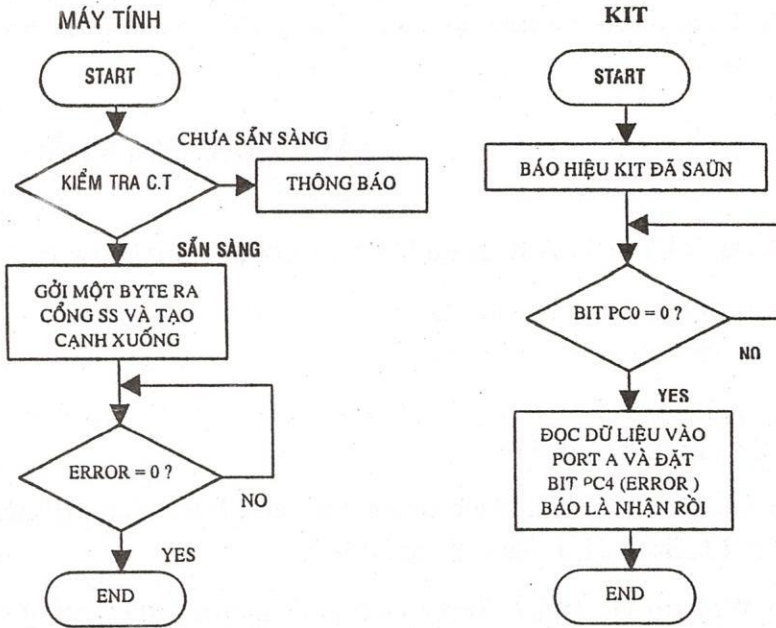
Mục đích của đề tài nghiên cứu là tìm cách sử dụng lại các bo mạch chính 386, 486. Để đạt mục đích này chúng tôi đã thực hiện ba nội dung:

- Tìm hiểu cấu trúc của các bo mạch chính 386, 486.
- Thiết kế thêm phần cứng và viết chương trình Monitor để chuyển đổi bo mạch chính thành bộ Kit dùng để giảng dạy lập trình tập lệnh của các vi xử lý họ Intel 80x86 và cấu trúc vi xử lý. Trong ứng

dụng này phương pháp **hỏi vòng** (Polling) được dùng để tiết kiệm chi phí lắp mạch, ngược lại việc lập trình ứng dụng bị một số hạn chế. Vì khuôn khổ giới hạn của bài báo, phần này không được trình bày.

➢ Thiết kế thêm phần cứng và viết chương trình Monitor để chuyển đổi bo chính thành một bộ Kit để giảng dạy cấu trúc vi xử lý và một phần nào đó cấu trúc máy tính, đồng thời cho các ứng dụng kỹ thuật (đo lường, điều khiển, tự động) như một Microcontroller hoặc PLC mạnh. Phương pháp **ngắt** (Interrupt) được dùng khiến mạch phức tạp hơn nhưng bộ CKIT-486 này có khả năng giao tiếp và mở rộng chương trình tốt hơn. CKIT-486 còn có thể giao tiếp với một máy tính nguyên bộ qua cổng song song khiến việc thử nghiệm hoặc cập nhật chương trình điều khiển rất thuận lợi.

Phần cứng và phần mềm của CKIT-486 thiết kế thêm đã được thực hiện và hoạt động tốt. Tùy mục đích sử dụng (giảng dạy cấu trúc vi xử lý, cấu trúc máy tính, đo lường hay điều khiển tự động) chúng ta có thể thêm phần cứng và phần mềm phù hợp.



Hình 10: Lưu đồ chương trình truyền từ máy trình sang CKIT-486

USING DISCARDED 386, 486 MAINBOARDS FOR TRAINING AND TECHNICAL APPLICATIONS

Nguyen Huu Phuong - Nguyen Van Thien

ABSTRACT : Due to the properties of microprocessors and features of mainboards, when upgrading substantially a computer we often have to change its mainboard whereas the other constituents (case and power supply, video card, monitor, etc.) remain in use. However the mainboard still has many capabilities especially for the purposes at the level lower than a complete computer set. This article presents our research result to reuse the 386, 486 mainboards which are almost discarded when we upgrate our computers. We designed

additional hardware and wrote appropriate programmes to turn the mainboards into relatively versatile microprocessor kits for training and technical applications. We have obtained good results.

TÀI LIỆU THAM KHẢO

1. Daniel Tabak, Advanced Microprocessors. McGraw Hill, New York (1995).
2. Barry B. Brey. The Intel 232-bit Microprocessors (80386, 80486 and Pentium). Prentice Hall, New Jersey (1995).
3. Intel Corporation. Intel 486 Microprocessor Family Programmer's Reference Manual. Intel Corporation (1993).
4. Douglas V. Hall. Microprocessors and Interfacing (Programming and Hardware), 2nd edition. McGraw Hill, New York (1992).
5. William H. Rigby, Terry Dalby. Computer Interfacing (A Practical Approach to Data Acquisition and Control). Prentice Hall, New Jersey (1995).