

# **Chương I**

## **Tổng quan về Vi điều khiển**

cuu duong than cong . com

***Biên soạn : Lâm tăng Đức – Lê Tiến Dũng – Bộ môn TĐH***

cuu duong than cong . com

# **Chương I**

## **Tổng quan về Vi điều khiển**

### **1.1 Giới thiệu về các họ vi xử lý và các họ vi điều khiển thông dụng**

#### **1.1.1 Lịch sử phát triển của bộ vi xử lý và bộ vi điều khiển**

Sự ra đời và phát triển nhanh chóng của kỹ thuật vi điện tử mà đặc trưng là kỹ thuật vi xử lý đã tạo ra một bước ngoặt quan trọng trong sự phát triển của khoa học tính toán, điều khiển và xử lý thông tin. Kỹ thuật vi xử lý đóng một vai trò rất quan trọng trong tất cả các lĩnh vực của cuộc sống và khoa học kỹ thuật, đặc biệt là lĩnh vực Tin học và Tự động hóa.

Năm 1971, hãng Intel đã cho ra đời bộ vi xử lý (microprocessor) đầu tiên trên thế giới tên gọi là Intel-4004/4bit , nhằm đáp ứng nhu cầu cấp thiết của một công ty kinh doanh là hãng truyền thông BUSICOM. Intel-4004 là kết quả của một ý tưởng quan trọng trong kỹ thuật vi xử lý số. Đó là một kết cấu logic mà có thể thay đổi được chức năng của nó bằng chương trình ngoài chứ không phát triển theo hướng tạo ra một cấu trúc cứng chỉ thực hiện một số chức năng nhất định như trước đây.

Sau đó, các bộ vi xử lý mới liên tục được đưa ra thị trường và ngày càng được phát triển, hoàn thiện hơn trong các thế hệ sau :

Vào năm 1972, hãng Intel đưa ra bộ vi xử lý 8-bit đầu tiên với tên Intel-8008/8bit.

Từ 1974 đến 1975 , Intel chế tạo các bộ vi xử lý 8-bit 8080 và 8085A.

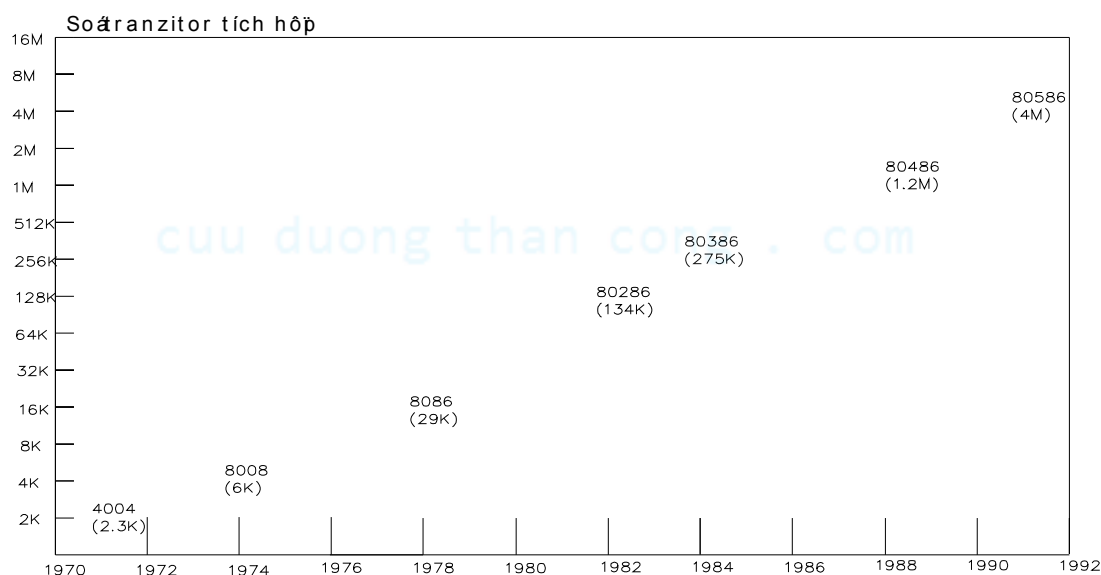
Cũng vào khoảng thời gian này, một loạt các hãng khác trên thế giới cũng đã cho ra đời các bộ vi xử lý tương tự như : 6800 của Motorola với 5000 tranzitor, Signetics 6520, 1801 của RCA, kế đến là 6502 của hãng MOS Technology và Z80 của hãng Zilog.

Vào năm 1976 Intel giới thiệu bộ vi điều khiển (microcontroller) 8748, một chip tương tự như các bộ vi xử lý và là chip đầu tiên trong họ vi điều khiển MCS-48. 8748 là một vi mạch chứa trên 17000 transistor, bao gồm một CPU, 1K byte EPROM, 64 byte RAM, 27 chân xuất nhập và một bộ định thời 8-bit. IC này và các IC khác tiếp theo của họ MCS-48 đã nhanh chóng trở thành chuẩn công nghiệp trong các ứng dụng hướng điều khiển (control-oriented application).

Năm 1978 xuất hiện Intel 8086 là loại bộ vi xử lý 16 bit với 29.000 tranzitor, Motorola 68000 tích hợp 70.000 tranzitor, APX 432 chứa 120.000 tranzitor. Bộ vi xử lý của Hewlet Pakard có khoảng 450.000 tranzitor. Từ năm 1974 đến 1984 số tranzitor tích hợp trong một chip tăng khoảng 100 lần.

Năm 1983, Intel đưa ra bộ vi xử lý 80286 dung trong các máy vi tính họ AT (Advanced Technology). 80286 sử dụng I/O 16 bit, 24 đường địa chỉ và không gian nhớ địa chỉ thực 16MB. Năm 1987, Intel đưa ra bộ vi xử lý 80386 32-bit. Năm 1989 xuất hiện xuất hiện bộ vi xử lý Intel 80486 là cải tiến của Intel 80386 với bộ nhớ ẩn và mạch tính phép toán đại số dấu phẩy động.

Năm 1992, xuất hiện Intel 80586 còn gọi là Pentium 64 bit chứa 4 triệu tranzitor.



**Hình 1.1** Số tranzitor tích hợp trong một chip của bộ vi xử lý Intel 8086

Độ phức tạp, sự gọn nhẹ về kích thước và khả năng của các bộ **vi điều khiển** được tăng thêm một bậc quan trọng vào năm 1980 khi Intel công bố chip 8051, bộ vi điều khiển đầu tiên của họ vi điều khiển MCS-51. So với 8048, chip 8051 chứa trên 60.000 transistor bao gồm 4K byte ROM, 128 byte RAM, 32 đường xuất nhập, 1 port nối tiếp và 2 bộ định thời 16-bit – một số lượng mạch đáng chú ý trong một IC đơn.

Từ các bộ vi xử lý ban đầu chỉ là các bộ xử lý trung tâm trong một hệ thống, không thể hoạt động nếu thiếu các bộ phận như RAM, ROM, bo mạch chủ... các hãng đã phát triển các bộ vi xử lý này lên thành các bộ vi điều khiển để phục vụ các mục đích riêng biệt, khác nhau trong công nghiệp. Một bộ vi điều khiển là một hệ vi xử lý thật sự được tổ chức trong một chip (trong một vỏ IC) bao gồm một bộ vi xử lý (microprocessor), bộ nhớ chương trình (ROM), bộ nhớ dữ liệu (RAM), tuy không bằng dung lượng RAM ở các máy vi tính nhưng đây không phải là một hạn chế vì các bộ vi điều khiển được thiết kế cho một mục đích hoàn toàn khác, ngoài ra trên chip còn có bộ xử lý số học-logic (ALU) cùng với các thanh ghi chức năng, các cổng vào/ra, cơ chế điều khiển ngắt, truyền tin nối tiếp, các bộ định thời... Hiện nay, các bộ vi điều khiển được sử dụng rất rộng rãi và ngày càng được chuẩn hóa để có thể sử dụng rộng rãi trong các ngành công nghiệp, có mặt trong nhiều máy móc, trong các hàng tiêu dùng.

**1.1.2 Ưu và khuyết điểm của các bộ vi điều khiển**

Các công việc được thực hiện bởi các bộ vi điều khiển thì không mới. Điều mới là các thiết kế hiện thực với ít thành phần hơn so với các thiết kế trước đó. Các thiết kế trước đó đòi hỏi phải vài chục hoặc vài trăm IC để hiện thực nay chỉ cần một ít thành phần trong đó bao gồm bộ vi điều khiển. Số thành phần được giảm bớt, hiệu quả trực tiếp của tính khả lập trình của các bộ vi điều khiển và độ

tích hợp cao trong công nghệ chế tạo vi mạch, thường chuyển thành thời gian phát triển ngắn hơn, giá thành khi sản xuất thấp hơn, công suất tiêu thụ thấp hơn và độ tin cậy cao hơn.

Vấn đề ở đây là tốc độ. Các giải pháp dựa trên bộ vi điều khiển không bao giờ nhanh bằng giải pháp dựa trên các thành phần rời rạc. Những tình huống đòi hỏi phải đáp ứng thật nhanh (cỡ nsec) đối với các sự kiện (thường chiếm thiểu số trong các ứng dụng) sẽ được quản lý tối khi dựa vào các bộ vi điều khiển.

Tuy nhiên trong vài ứng dụng, đặc biệt là các ứng dụng liên quan đến con người, các khoảng thời gian trễ tính bằng nsec, usec hoặc thậm chí msec là không quan trọng. Việc giảm bớt các thành phần là một điều lợi như đã đề cập, các thao tác trong chương trình điều khiển làm cho thiết kế có thể thay đổi bằng cách thay đổi phần mềm. Điều này có ảnh hưởng tối thiểu đến chu kỳ sản xuất. Do đó các bộ vi điều khiển có thể được ứng dụng rộng rãi trong các ứng dụng phục vụ con người.

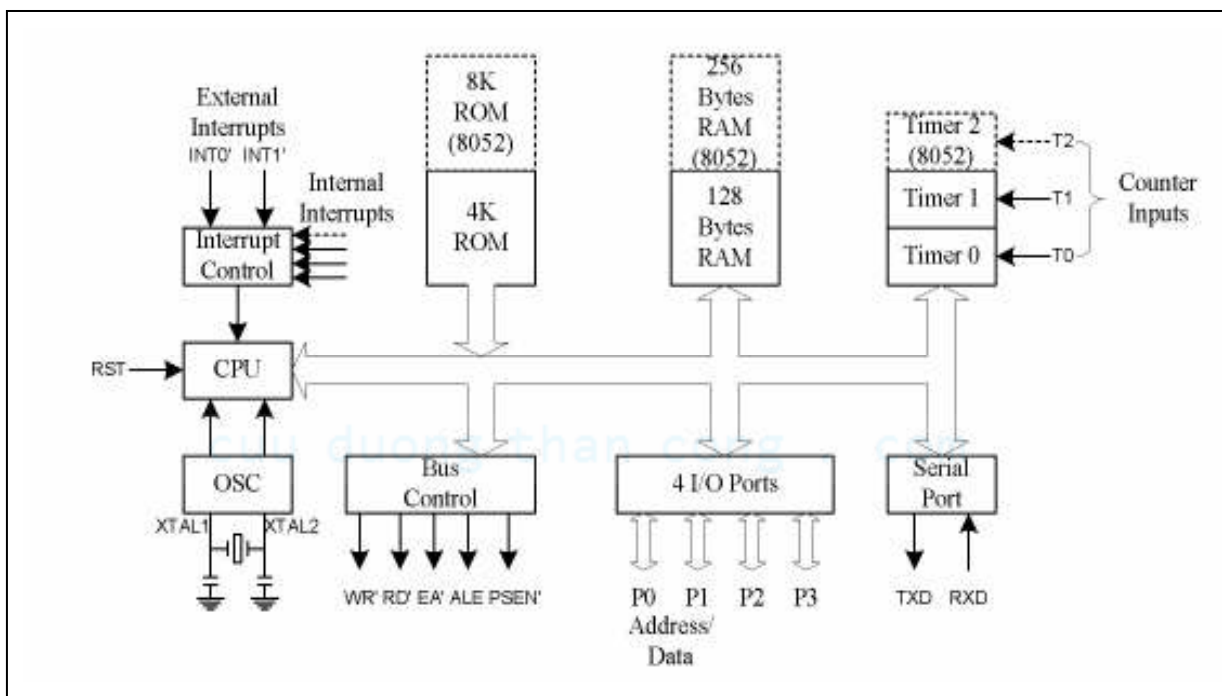
Để có thể hiểu rõ hơn về các bộ vi điều khiển, chúng ta sẽ tìm hiểu về một số các họ vi điều khiển của một số hãng điện tử điển hình đang được sử dụng rộng rãi trong khoa học kỹ thuật và đời sống.

### 1.1.3 Giới thiệu về họ vi điều khiển MCS-51

#### 1.1.3.1 Tóm tắt phần cứng

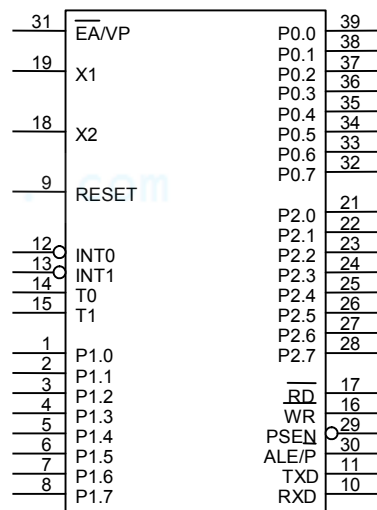
Họ vi điều khiển MCS-51 được hãng Intel cho ra đời vào năm 1980 với bộ vi điều khiển đầu tiên của nó là chip 8051.

Cấu trúc cơ bản của chip vi điều khiển 8051 được biểu diễn như **hình 1.2**.



**Hình 1.2-** Sơ đồ khối của chip 8051

**Hình 1.3** cho ta sơ đồ chân của chip 8051. Như ta thấy, 32 trong số 40 chân của 8051 có tác dụng xuất/nhập, hình thành 4 port 8-bit. Với các thiết kế yêu cầu một mức tối thiểu bộ nhớ ngoài hoặc các thành phần bên ngoài khác, ta có thể sử dụng các port này làm nhiệm vụ xuất/nhập, 8 đường cho mỗi port có thể được xử lý như là một đơn vị giao tiếp với các thiết bị song song như máy in, bộ biến đổi



**8051**

**Hình 1.3 -** Sơ đồ chân của 8051

D/A, v.v.. hoặc mỗi đường có thể hoạt động độc lập giao tiếp với một thiết bị đơn bit như chuyển mạch, LED, tranzistor, cuộn dây, động cơ...

### *1. Port 0*

Port 0 có 2 tác dụng. Trong các thiết kế ứng dụng tối thiểu, không có giao tiếp với các thành phần bên ngoài như RAM ngoài, ROM ngoài..., port 0 được sử dụng như một cổng xuất/nhập thông thường. Khi thiết kế hệ thống mà cần sử dụng bộ nhớ ngoài, port 0 được sử dụng là cổng địa chỉ và dữ liệu đa hợp. Port 0 nếu là cổng địa chỉ thì nó sẽ là phần byte thấp của bus địa chỉ.

### *2. Port 1*

Port 1 chỉ có một tác dụng là cổng xuất nhập. Nó chỉ dùng để giao tiếp với thiết bị ngoại vi khi có yêu cầu. Không có chức năng nào đặc biệt cho các chân của port 1. Tuy nhiên với các bộ vi điều khiển khác như 8052 thì các chân P1.0 và P1.1 còn được sử dụng làm các đường ngõ vào cho mạch định thời thứ ba.

### *3. Port 2*

Port 2 có hai tác dụng, hoặc làm nhiệm vụ là cổng xuất nhập, hoặc là phần byte cao của bus địa chỉ 16-bit cho các thiết kế hệ thống cần nhiều hơn 256 byte bộ nhớ ngoài.

### *4. Port 3*

Port 3 có hai tác dụng. Nó làm nhiệm vụ như một cổng xuất nhập thông thường. Khi không hoạt động xuất nhập, mỗi chân của port 3 đều có một chức năng riêng.

### *5. Chân cho phép bộ nhớ ngoài /PSEN - Program Store ENable*

Chân này thường được nối với chân cho phép xuất /OE (Output Enable) của EPROM (hoặc của ROM) để cho phép đọc các byte lệnh. Tín hiệu /PSEN ở logic '0' trong suốt thời gian tìm nạp lệnh. Các mã nhị phân của chương trình (opcode) được đọc từ EPROM, qua bus dữ liệu và được chốt vào thanh ghi lệnh IR của 8051 để được giải mã. Khi thực thi một chương trình trong ROM nội, /PSEN được duy trì ở logic không tích cực (logic 1).

### ***6. Chân cho phép chốt địa chỉ ALE - Address Latch Enable***

Là tín hiệu xuất ra để giải đa hợp bus địa chỉ và bus dữ liệu. Trong 1/2 chu kỳ đầu bộ nhớ, chân ALE xuất tín hiệu để chốt địa chỉ (byte thấp của bus địa chỉ 16 bit) vào thanh ghi ngoài khi ta sử dụng port 0 làm byte thấp địa chỉ. Trong 1/2 chu kỳ bộ nhớ còn lại port 0 sẽ xuất/nhập dữ liệu. Tín hiệu ALE có tần số bằng 1/6 tần số của mạch dao động bên trong chip vi điều khiển và có thể làm xung clock cho các phần còn lại của hệ thống (trường hợp ngoại lệ khi thực hiện lệnh MOVX, một xung ALE (và cả /PSEN) sẽ bị bỏ qua). Chân ALE còn được dùng để nhập xung ngõ vào lập trình cho EPROM hoặc Flash ROM trên chip đối với chip có loại ROM này.

### ***7. Chân truy xuất ngoài /EA - External Address***

Chân này được nối lên 5V khi thực thi chương trình trong ROM nội và được nối đất khi thực thi chương trình bộ nhớ ngoài. Chú ý đối với các chip không có ROM nội /EA phải được nối đất. Các chip họ 8051 có EPROM còn nhận chân /ALE làm chân nhận điện áp cấp điện 12V cho việc lập trình ( nạp) cho EPROM nội.

### ***8. Chân RESET (RST)***

Dùng để thiết lập lại trạng thái ban đầu của hệ thống hay gọi tắt là reset hệ thống khi được treo ở mức logic 1 ít nhất 2 chu kỳ máy. Các thanh ghi bên trong của 8051 được nạp các giá trị thích hợp cho việc khởi động lại hệ thống.

### ***9. Các chân XTAL1 và XTAL2***

Mạch dao động bên trong chip 8051 được ghép với thạch anh bên ngoài ở 2 chân XTAL1 và XTAL2. Thường tần số là 12MHz và các tụ ổn định có giá trị trong khoảng 30pF – 33 pF.



### 1.1.3.2 Tổ chức bộ nhớ của 8051

Các chip vi điều khiển dùng làm thành phần trung tâm trong các thiết kế hướng điều khiển. Bộ nhớ thường có dung lượng bé hơn nhiều so với một hệ vi xử lý. Nó còn không có ổ đĩa và hệ điều hành. Chương trình điều khiển phải thường trú trong ROM. Do vậy, chương trình vẫn được lưu giữ ngay cả khi mất điện.

Do lý do trên, chip 8051 có không gian bộ nhớ riêng cho chương trình và dữ liệu. Cả bộ nhớ chương trình và bộ nhớ dữ liệu đều nằm trong chip. Tuy nhiên ta có thể mở rộng bộ nhớ chương trình và bộ nhớ dữ liệu bằng cách sử dụng các chip nhớ bên ngoài với dung lượng tối đa là 64K cho bộ nhớ chương trình và 64K cho bộ nhớ dữ liệu.

#### ***Bộ nhớ chương trình (ROM)***

Bộ nhớ chương trình lưu giữ chương trình điều khiển chip 8051.

Sau khi RESET, CPU bắt đầu thực hiện chương trình từ địa chỉ 0000H. Khi chương trình lớn quá kích thước bộ nhớ chương trình bên trong chip, chương trình này phải được nạp vào bộ nhớ chương trình ngoài. Nếu chương trình nằm trong ROM nội, chân /EA của 8051 phải được treo lên 5V. Nếu chương trình ở ROM ngoài, chân /EA phải nối đất. Việc truy xuất chương trình ở bộ nhớ ngoài phải kết hợp với chân tín hiệu truy xuất bộ nhớ ngoài /PSEN.

#### ***Bộ nhớ dữ liệu (RAM)***

8051 có 128 byte RAM ở bên trong chip. Chúng được chia làm nhiều vùng khác nhau : vùng RAM đa mục đích, vùng RAM định địa chỉ bit, các dãy thanh ghi, và các thanh ghi chức năng đặc biệt. Ta hãy xem xét từng vùng RAM cụ thể.

- *Vùng RAM đa mục đích* : Có địa chỉ từ 30H đến 7FH (80 byte). Vùng RAM này có thể truy xuất bằng cách định địa chỉ trực tiếp hoặc định địa chỉ gián tiếp. Nó có thể dùng để chứa các biến trong chương trình hay dùng để định địa chỉ cho các cổng ngoại vi tùy theo mục đích của người sử dụng.

- *Vùng RAM định địa chỉ từng bit* : Có 128 bit chứa trong các byte ở địa chỉ từ 20H đến 2FH và 32 byte chứa các thanh ghi (00H đến 1FH).

- *Các dãy thanh ghi từ Bank 0 tới Bank 3* : Nằm ở 32 byte thấp nhất của vùng nhớ dữ liệu. Khi RESET hệ thống dãy thanh ghi mặc định là Bank 0. Có thể chọn Bank bởi thanh ghi từ trạng thái chương trình (PSW- Program Status Word) . Các lệnh sử dụng các thanh ghi từ R0 đến R7 là các lệnh ngắn và thực hiện nhanh hơn so với các lệnh tương đương sử dụng kiểu định địa chỉ trực tiếp. Các giá trị dữ liệu thường được sử dụng nên chứa ở một trong các thanh ghi này.

### ***Các thanh ghi chức năng đặc biệt (SFR- Special Function Register)***

Có 21 thanh ghi chức năng đặc biệt chiếm phần trên của RAM nội có địa chỉ từ 80H đến FFH. Các địa chỉ được định nghĩa trong vùng RAM này gồm các thanh ghi điều khiển của 8051, các thanh ghi đếm, và các cổng vào ra.

- Thanh ghi từ trạng thái **PSW - Program Status Word** : Có địa chỉ là D0H. Chứa các bit trạng thái có chức năng khác nhau.
- Thanh ghi chứa **ACC**: Có địa chỉ là A0H. Thường được dùng làm biến nhớ trung gian trong các phép tính toán số học.
- Thanh ghi **B**: Có địa chỉ là F0H. Thường được dùng chung với thanh ghi ACC trong các phép toán nhân chia.
- Các cổng giao tiếp song song : Đó là các cổng P0, P1, P2, P3. Được định địa chỉ tương ứng là 80H, 90H, A0H, B0H.
- Các thanh ghi phục vụ cho truyền thông nối tiếp **SCON (Serial port CONtrol)** và **SBUF (Serial data BUffer)**: Thanh ghi SCON dùng để thiết lập các thông số cho việc truyền thông nối tiếp. Còn thanh ghi SBUF dùng làm vùng nhớ đệm buffer cho việc truyền thông.
- Thanh ghi **PCON - Power CONtrol**: Đây là thanh ghi điều khiển nguồn cấp cho 8051. Có thể dùng thanh ghi này để đặt bộ vi điều khiển vào chế độ Power Down hoặc IDE.

- Các thanh ghi điều khiển ngắt **IP (Interrupt Priority)** và **IE (Interrupt Enable)** : Thanh ghi IE dùng để cho phép các ngắt hoạt động/không hoạt động. Còn thanh ghi IP dùng để xác định mức ưu tiên cho các ngắt của 8051.

- Các thanh ghi điều khiển bộ định thời : Đó là các thanh ghi **TMOD, TCON, TH0, TL0, TH1, TL1**. Các thanh ghi này được sử dụng để điều khiển bộ định thời 0 và 1.

- Thanh ghi **SP - Stack Pointer** : Thanh ghi con trỏ ngăn xếp (stack), SP chứa địa chỉ của dữ liệu hiện đang ở đỉnh stack. Nội dung mặc định của SP khi khởi động là 07H. Thao tác cất vào stack đầu tiên sẽ lưu dữ liệu vào vị trí nhớ có địa chỉ 08H vì cơ chế làm việc của chip 8051 là tăng nội dung SP lên 1 trước khi thực hiện lưu dữ liệu vào ngăn xếp.

- Con trỏ dữ liệu **DPTR - Data Pointer** : Được dùng để truy xuất bộ nhớ chương trình ngoài hoặc bộ nhớ dữ liệu ngoài. DPTR là thanh ghi 16 bit gồm 2 phần là DPH và DPL.

### 1.1.3.3 Các hoạt động chức năng chính của 8051

#### *Hoạt động định thời*

Các bộ định thời dùng để tạo ra các khoảng thời gian khác nhau, dùng để đếm sự kiện hoặc dùng để tạo tốc độ baud cho việc truyền thông nối tiếp. Trong 8051 có hai bộ định thời là T0 và T1. Việc lựa chọn chế độ hoạt động cho các Timer này nhờ vào thanh ghi TMOD. Còn việc điều khiển các Timer hoạt động nhờ vào thanh ghi TCON. Có 4 mode hoạt động cho các Timer. *Mode 0* là chế độ định thời 13-bit, *Mode 1* là chế độ định thời 16-bit, *Mode 2* là chế độ định thời tự nạp lại 8-bit, *Mode 3* là chế độ định thời chia sẻ và có hoạt động khác nhau cho từng bộ định thời.

#### *Hoạt động của port nối tiếp*

Chức năng cơ bản của port nối tiếp là chuyển đổi dữ liệu từ song song thành nối tiếp khi phát và từ nối tiếp thành song song khi thu. Dữ liệu được truyền đi hoặc nhận

về thông qua bộ đệm dữ liệu nối tiếp SBUF (Serial data BUffer). Khi truyền đi, dữ liệu được chuyển từ song song sang nối tiếp thông qua chân TxD. Khi thu, dữ liệu được chuyển từ nối tiếp sang song song thông qua chân RxD. Thanh ghi SBUF có địa chỉ 99H. Việc ghi dữ liệu cho SBUF tức là phát dữ liệu, việc đọc SBUF tức truy xuất dữ liệu nhận được (thu dữ liệu). Điều khiển port nối tiếp thông qua thanh ghi SCON có địa chỉ là 98H, thanh ghi này là thanh ghi định địa chỉ từng bit cho ta xác định được các chế độ làm việc của port nối tiếp (thông qua hai bit MS0 & MS1).

### ***Hoạt động của ngắt trong 8051***

Có 5 nguyên nhân để tạo ra ngắt trong 8051. Đó là 2 ngắt ngoài, hai ngắt do bộ định thời và một ngắt do port nối tiếp. Khi ta thiết lập trạng thái ban đầu (sau khi RESET), tất cả các ngắt đều bị vô hiệu hoá và sau đó chúng được cho phép riêng rẽ bằng phần mềm.

Khi chương trình đang thực hiện, nếu có ngắt với ưu tiên cao xuất hiện, trình phục vụ ngắt cho ngắt có mức ưu tiên thấp tạm dừng. Ta không thể tạm dừng một chương trình ngắt có mức ưu tiên cao hơn. Khi có 2 ngắt khác nhau xuất hiện đồng thời, ngắt có mức ưu tiên cao sẽ được phục vụ trước. Khi 2 ngắt có cùng mức ưu tiên xuất hiện đồng thời, **chuỗi vòng cố định** sẽ xác định ngắt nào được phục vụ trước. Chuỗi vòng này sẽ là ngắt ngoài 0, ngắt ngoài 1, ngắt do bộ định thời 0, ngắt do bộ định thời 1, ngắt do port nối tiếp, ngắt do bộ định thời 2 (đối với 8052).

Khi một ngắt được chấp nhận, giá trị được nạp cho bộ đếm chương trình được gọi là **véc tơ ngắt**. Véc tơ ngắt là địa chỉ bắt đầu của trình phục vụ ngắt của các ngắt tương ứng. Các **véc tơ ngắt** được cho ở bảng sau :

<i>Nguồn ngắt</i>	<i>Cờ</i>	<i>Địa chỉ vector ngắt</i>
Reset hệ thống	RST	0000H
Ngắt ngoài 0	IE0	0003H
Bộ định thời 0	TF0	000BH
Ngắt ngoài 1	IE1	0013H
Bộ định thời 1	TF1	001BH
Port nối tiếp	RI or TI	0023H
Bộ định thời 2	TF2 or EXF2	002BH

### ***Hoạt động RESET của 8051***

8051 được reset bằng cách giữ chân RST ở mức cao tối thiểu hai chu kỳ máy và sau đó chuyển về mức thấp. Trạng thái của tất cả các thanh ghi sau khi reset hệ thống như sau :

<i>Thanh ghi</i>	<i>Nội dung</i>
Bộ đếm chương trình PC	0000H
Thanh chứa A	00H
Thanh ghi B	00H
PSW	00H
SP	07H
DPTR	0000H
Port 0 đến port 3	FFH
IP	xxx00000B
IE	0xx00000B
Các thanh ghi định thời	00H
SCON	00H
SBUF	00H
PCON ( HMOS)	0xxxxxxxB
PCON ( CMOS)	0xxx0000B

Khi *reset* hệ thống thanh ghi PC được nạp địa chỉ 0000H, khi đó chương trình sẽ bắt đầu từ địa chỉ đầu tiên trong bộ nhớ chương trình. Nội dung của RAM trên chip không bị ảnh hưởng khi ta *reset* hệ thống.

### 1.1.4 Giới thiệu họ vi điều khiển 8 bit của Motorola

Các họ vi điều khiển của Motorola được kí hiệu bắt đầu bởi chữ cái MC. Vì cố gắng giữ lại phần mềm các nhà chế tạo linh kiện bán dẫn thường duy trì sự tương thích với các bộ vi xử lí đầu đời. Hãng Motorola đưa ra bộ vi điều khiển M6801 đầu tiên của hãng trên thị trường thế giới vào năm 1978 và bộ vi điều khiển này được chế tạo trên cơ sở sử dụng bộ vi xử lí M6800. Bộ vi điều khiển M6801 có bộ nhớ ROM và RAM ngoài lớn nhất lên đến 64KB với mạch giao diện vào/ra, truyền thông nối tiếp không đồng bộ đã bắt đầu phổ biến trong hệ thống điều khiển ô tô.

Sau đó vào năm 1985 Motorola đã phát triển bộ vi điều khiển 68HC11 tương thích với bộ vi xử lí M6800. Bộ vi điều khiển MC68HC11 tiêu thụ ít công suất hơn và cho phép tín hiệu nhiễu cao hơn so với các bộ vi xử lí đầu đời. MC68HC11 có nhiều phiên bản, ví dụ MC68HC11A8 và MC68HC11E9. Một số hãng như Mitsubishi, Toshiba cũng sản xuất các bộ vi điều khiển này theo công nghệ của Motorola.

Bảng dưới đây liệt kê các đặc điểm cơ bản của một số chip vi điều khiển trong họ 68HC11:

MC	ROM	RAM	EEPROM	ADC	I/O
68HC11A8	8KB	256 bytes	512 bytes	8 channels	38 pins
68HC811E2	0	256 bytes	2 KB	-	-
68HC11F1	0	1 KB	-	-	-
68HC11E9	12 KB	512 bytes	512 bytes	-	-
68HC11B8	8 KB	256 bytes	512 bytes	-	-
68HC11B0	0	256 bytes	0	-	-
68HC11B1	0	256 bytes	512bytes	-	-
68HC11D3	4 KB	192 bytes	-	-	40 pins
68HC11A0	0	256 bytes	0	-	22 pins

### 1.1.5 Giới thiệu các họ vi điều khiển 8 bit của hãng Microchip Technology

Hãng Microchip Technology là một trong những nhà cung cấp hàng đầu về các giải pháp điều khiển cho các hệ thống nhúng (Embedded control system). Hai dòng sản phẩm chính của hãng là các họ vi điều khiển 8-bit **PIC16/17** và các chip nhớ không bay hơi EEPROM nối tiếp, những sản phẩm này được đưa vào ứng dụng cho việc đưa ra các giải pháp thiết kế các hệ thống điều khiển nhúng của hãng cung cấp cho khách hàng.

Các họ vi điều khiển **PIC16Cxx** và **PIC17Cxx** của hãng Microchip Technology được sử dụng khá thông dụng và rộng rãi cho các ứng dụng điều khiển trong việc chế tạo các sản phẩm tiêu dùng, trong tự động hóa văn phòng, thiết bị ngoại vi của máy tính, các hệ thống tự động điều khiển ...

Dưới đây liệt kê các đặc điểm chính, các thành phần cơ bản của một số chip vi điều khiển trong các họ vi điều khiển 8-bit PIC16Cxx và PIC17Cxx.

#### Họ PIC16Cxx

PIC	(0)	Memory				Peripheral						Features	
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(12)	(13)
<b>16C61</b>	20	1K	–	36	–	1	–	–	–	–	–	3	13
<b>16C62</b>	20	2K	–	128	–	3	2	Yes	–	–	–	10	22
<b>16C63</b>	20	4K	–	192	–	3	2	Yes	–	–	–	10	22
<b>16C64</b>	20	2K	–	128	–	3	1	Yes	Yes	–	–	8	33
<b>16C65</b>	20	4K	–	192	–	3	2	Yes	Yes	–	–	11	33
<b>16C620</b>	20	512	–	80	–	1	–	–	–	–	2	4	13
<b>16C621</b>	20	1K	–	80	–	1	–	–	–	–	2	4	13
<b>16C622</b>	20	2K	–	128	–	1	–	–	–	–	2	4	13
<b>16C71</b>	20	1K	–	36	–	1	–	–	–	4	–	4	13
<b>16C73</b>	20	4K	–	192	–	3	2	Yes	–	5	–	11	22
<b>16C74</b>	20	4K	–	192	–	3	2	Yes	Yes	8	–	12	33
<b>16C84</b>	10	–	1K	36	64	1	–	–	–	–	–	4	13

Chú thích :

- (0) : Tần số cấp tạo xung Clock (MHz).
- (1) : Dung lượng bộ nhớ chương trình EPROM trong.
- (2) : Dung lượng bộ nhớ chương trình EEPROM trong.
- (3) : Dung lượng bộ nhớ dữ liệu RAM trong (bytes).
- (4) : Dung lượng bộ nhớ dữ liệu EEPROM trong (bytes).
- (5) : Số Module Timer.
- (6) : Số Capture, Module PWM.
- (7) : Cổng nối tiếp.
- (8) : Cổng song song.
- (9) : Số kênh chuyển đổi Analog – Digital (ADC).
- (10) : Số bộ so sánh (Comparator).
- (11) : Số nguồn ngắt (Interrupt source).
- (12) : Số đường vào/ra (I/O pins).

### Họ PIC17Cxx

PIC	(0)	Memory		Peripherals			Features			
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
17C42	25	2K	232	4	2	Yes	Yes	11	33	55
17C43	25	4K	454	4	2	Yes	Yes	11	33	58
17C44	25	8K	454	4	2	Yes	Yes	11	33	58

Chú thích :

- (0) : Tần số cấp tạo xung Clock (MHz).
- (1) : Dung lượng bộ nhớ chương trình EPROM bên trong.
- (2) : Dung lượng bộ nhớ dữ liệu RAM bên trong.



- (3) : Số Module Timer.
- (4) : Số Module PWM.
- (5) : Cổng nối tiếp.
- (6) : Ngắt ngoài (External interrupts).
- (7) : Số nguồn ngắt (Interrupt sources).
- (8) : Số đường vào/ra (I/O pins).
- (9) : Số lệnh điều khiển (Instructions).

### **Đánh giá các dòng PIC**

- Dòng PIC nhiều chân nhất là dòng dsPIC30Fxxxx và PIC18Fxxxx, có những con số chân lên đến 80 chân.
- Dòng PIC ít chân nhất là dòng PIC10Fxxx, chỉ có 6 chân
- Dòng PIC phổ biến nhất là dòng PIC16F877A (đủ mạnh về tính năng, 40 chân, bộ nhớ đủ cho hầu hết các ứng dụng thông thường)
- Dòng PIC mà được các chuyên gia đánh giá cao nhất là dòng PIC16F876A (28 chân, chức năng không khác gì so với PIC16F877A, nhưng nhỏ gọn hơn nhiều, và số chân cũng không quá ít như PIC16F88).
- Dòng PIC hỗ trợ giao tiếp USB là dòng PIC18F2550 và PIC18F4550
- Dòng PIC điều khiển động cơ mạnh nhất là dòng PIC18F4x31 và dòng dsPIC30F dùng điều khiển động cơ
- Dòng PIC hiện nay đã không còn được sản xuất nữa là dòng PIC17xxxxx.

\*(Các đánh giá về PIC được trích ở website [www.picvietnam.com](http://www.picvietnam.com))

### **1.1.6 Giới thiệu chip vi điều khiển thông minh PSoC**

PSoC là chữ viết tắt của *Programmable System on Chip*. Đó là một công nghệ chế tạo chip hiện đại nhất cho đến nay, cho phép tạo ra các chip hệ thống xử lý hỗn hợp (Mixed signal) với khả năng xử lý thuật toán mạnh.

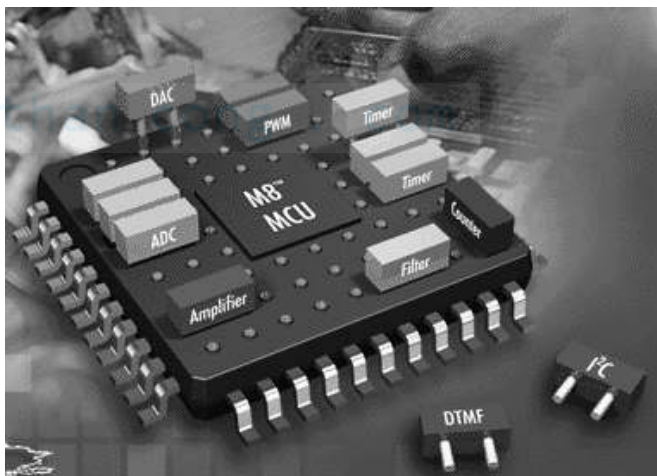
Đặc điểm của các thế hệ vi điều khiển chế tạo theo công nghệ trước đây là trên một chip vi điều khiển chỉ có thể thay đổi được chương trình phần mềm chứ không thể lập trình thay đổi được cấu trúc phần cứng. Giá thành cho việc sử dụng các chip vi điều

khuyến và mua các phần mềm hỗ trợ để phát triển các ứng dụng là rất đắt và mang lại hiệu quả không cao, khó có thể thông minh hóa việc điều khiển hệ thống.

Với chip vi điều khiển thông minh chế tạo theo công nghệ PSoC, chúng ta có thể lập trình trên chip để thay đổi không những cả phần mềm điều khiển mà cả cấu trúc phần cứng ngay khi hệ thống đang hoạt động. Việc ứng dụng chip vi điều khiển thông minh theo công nghệ PSoC mang lại hiệu quả cao về kỹ thuật và giá thành giảm đi nhiều lần so với việc sử dụng các chip vi điều khiển trước đây. Đối với các chip vi điều khiển thông minh, bên trong nó đã tích hợp đầy đủ các thành phần CPU, các bộ nhớ, các khối (block) tương tự và số nên một số thành phần ngoại vi bên ngoài chip trong hệ thống có thể giảm đi. Vì thế, giá thành và kích thước của hệ thống giảm đi nhiều so với hệ thống sử dụng các chip vi điều khiển thế hệ trước đây.

Hiện nay, có nhiều hãng điện tử trên thế giới quan tâm đầu tư cho việc nghiên cứu và áp dụng công nghệ chế tạo chip mới này. Trong đó có thể kể đến hãng Cypress Microchip đã áp dụng thành công công nghệ PSoC để cho ra đời các họ vi điều khiển thông minh CY8C24xxx, CY8C25xxx, CY8C26xxx và CY8C27xxx.

Một ví dụ cụ thể về khả năng lập trình thay đổi cả cấu hình phần cứng và chương trình phần mềm điều khiển là đề tài ứng dụng chip vi điều khiển thông minh công nghệ PSoC để chế tạo máy bán hàng thông minh tự động. Máy có chức năng bán hàng tự động vào ban ngày và ban đêm thì nó thay đổi cả cấu hình phần cứng và chương trình phần mềm để làm chức năng truyền thông.



**Hình 1.3** - Cấu hình của một chip PSoC

## 1.2 Giới thiệu các công cụ phát triển hệ vi xử lý và vi điều khiển

### 1.2.1 Công cụ phát triển cho họ MCS-51

#### 1.2.1.1 Phần mềm Keil của hãng Keil Software

Phần mềm Keil Vision2 là một chương trình biên dịch cho phép người sử dụng có thể viết được các chương trình điều khiển cho họ vi điều khiển MSC-51 bằng ngôn ngữ C và cả ASM. Nó có chứa hầu hết các hàm chuẩn của ANSI-C. Điều này giúp cho người sử dụng có thể tận dụng được những hàm chuẩn của C để viết chương trình cho vi điều khiển. Việc tổ chức chương trình thành các môđun nhỏ cũng rất dễ thực hiện. Chương trình biên dịch cho ngôn ngữ C của nó chính là Cx51 Compiler.

Cx51 Compiler hỗ trợ tất cả các kiểu dữ liệu chuẩn của C như : char, short, int, long, float. Ngoài ra nó còn có một vài kiểu dữ liệu đặc biệt tương thích với họ vi điều khiển MSC-51.

- *Kiểu bit* : đây là kiểu dữ liệu chỉ gồm 2 giá trị 0 và 1. Nó được sử dụng để gán giá trị cho các thanh ghi hay các địa chỉ có thể truy nhập được đến từng bit.

- *Kiểu sbit, sfr, sfr16* : đây là các kiểu dữ liệu dùng để định nghĩa cho các thanh ghi chức năng đặc biệt. Kiểu sfr dùng để định nghĩa cho các thanh ghi SFR 8-bit. Còn kiểu sfr16 dùng để định nghĩa cho các thanh ghi SFR 16-bit như thanh ghi RCAP2 dùng để lưu giữ giá trị nạp lại cho Timer 2. Kiểu sbit dùng để định nghĩa từng bit cho các bit trong các thanh ghi đặc biệt mà có thể định địa chỉ bit.

Ví dụ khai báo :

```
sfr P0    = 0x80;    /* Port 0 */
sfr SP    = 0x81;    /* Stack Pointer */
sbit P0_0 = 0x80;
sbit P0_1 = 0x81;
```

Về cách tổ chức bộ nhớ của vi điều khiển trong phần mềm Keil C51 :

- *Bộ nhớ chương trình* : Đây là vùng nhớ chỉ đọc trong vi điều khiển. Nó có thể có dung lượng tới 64KB. Trong phần mềm Keil C51 thì vùng nhớ này được định dạng bằng từ khoá **code**.

Ví dụ khai báo biến thuộc vùng **code** :

```
char code str[]={“DHBKDN”}; //Khai báo một mảng ký tự chứa trong vùng nhớ code
```

- *Bộ nhớ dữ liệu trong* : Đây là vùng nhớ có thể vừa ghi và đọc. Có 3 kiểu vùng nhớ dữ liệu ở bên trong vi điều khiển. Đó là các kiểu : **data**, **idata**, **bdata**. Vùng nhớ dữ liệu kiểu **data** là vùng nhớ chứa 128 byte dữ liệu đầu tiên của vi điều khiển. Vùng nhớ này có thể truy cập thông qua địa chỉ trực tiếp được. Vùng nhớ dữ liệu kiểu **idata** là vùng nhớ 256 byte dữ liệu truy nhập thông qua địa chỉ gián tiếp. Vùng nhớ dữ liệu kiểu **bdata** là vùng nhớ chứa 16 byte định địa chỉ bit từ địa chỉ 20H đến 2FH.

Ví dụ khai báo biến trong vùng nhớ dữ liệu trong :

```
unsigned char data v1; //Khai báo biến v1
```

```
unsigned int idata v2; //Khai báo biến v2
```

- *Bộ nhớ dữ liệu ngoài* : Có 2 kiểu dữ liệu ngoài ở trong Keil C51 là **XDATA** và **PDATA**. **XDATA** là vùng nhớ dữ liệu ngoài có thể định địa chỉ được đến 64 KB dữ liệu. **PDATA** là vùng nhớ chứa từng trang dữ liệu. Mỗi trang dữ liệu ngoài chứa được 256 byte. Để có thể truy cập được bộ nhớ dữ liệu ngoài, ta phải dùng từ khoá **\_at\_** để định địa chỉ cho các biến.

Ví dụ khai báo biến trong vùng nhớ dữ liệu ngoài :

```
float xdata a1 _at_ 0x2000; //Khai báo biến a1 có địa chỉ 2000H
```

```
unsigned long pdata a2 _at_ 0x2004; // Biến a2 có địa chỉ 2004
```

Con trỏ trong Keil C51 được khai báo giống như khai báo con trỏ trong ngôn ngữ C.

Ví dụ về khai báo con trỏ :

```
char *s; //Con trỏ của một string
```

```
int *p; //Con trỏ kiểu int
```

Con trỏ được sử dụng để truy cập tới bất cứ phần bộ nhớ nào của vi điều khiển. Có hai cách khai báo sử dụng con trỏ. Generic Pointers là kiểu con trỏ dùng chung. Khai báo con trỏ sẽ chiếm 3 byte của vùng nhớ dữ liệu mà nó trỏ đến.

Con trỏ còn có thể khai báo vùng nhớ dữ liệu cho con trỏ.

Ví dụ :

```
char data *str; /* ptr to string in data */  
int xdata *numtab; /* ptr to int(s) in xdata */  
long code *powtab; /* ptr to long(s) in code */
```

Ngoài ra, Keil C51 còn là chương trình mô phỏng chip rất tốt bằng chương trình tích hợp dScope. Nó cho phép người sử dụng mô phỏng chương trình viết trong C chạy trên nền chip giống như thật. Người sử dụng có thể dễ dàng theo dõi giá trị của Timer, trạng thái các ngắt, đầu ra cổng nối tiếp, đầu ra các port P0...P3.

#### **1.2.1.2 Phần mềm Borland Delphi 7.0 của hãng Borland**

Borland Delphi 7.0 là một công cụ mạnh dùng để xây dựng các ứng dụng 32 bit chạy trên nền Windows bao gồm cả Windows 9x và NT.

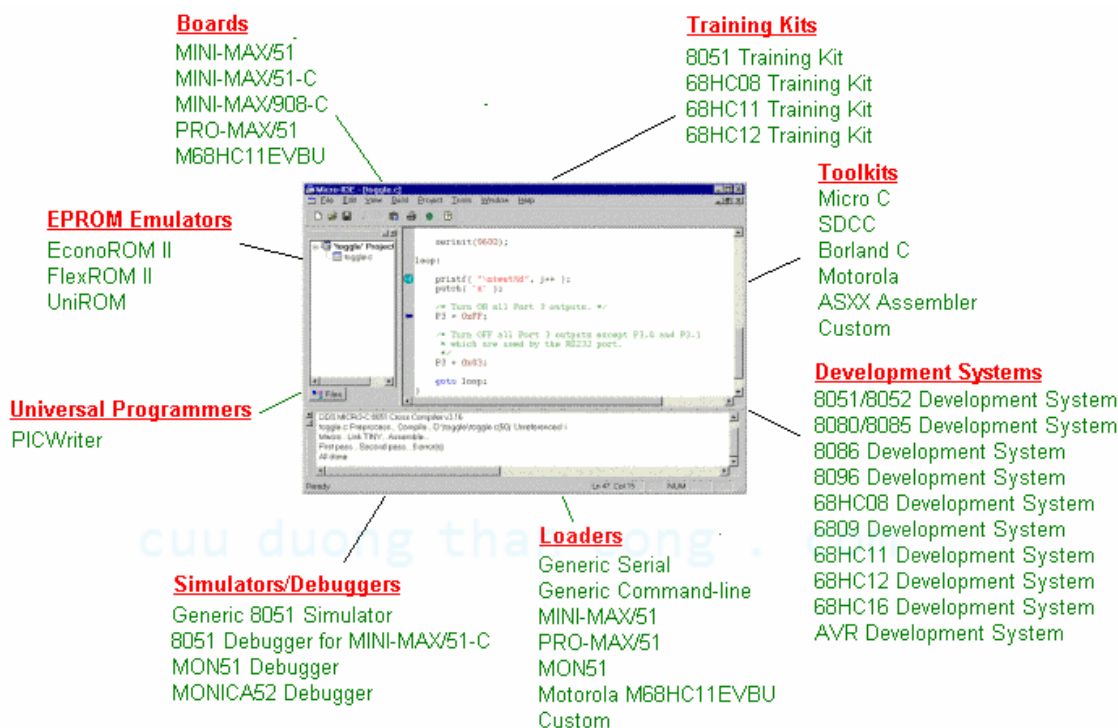
#### **1.2.3 Công cụ phát triển cho họ vi điều khiển MC68HC11**

Các chip vi điều khiển họ Motorola 68HC11 có thể được lập trình bằng ngôn ngữ Assembly hoặc C như các họ vi điều khiển khác. Chúng ta có thể tham khảo công cụ phát triển hỗ trợ cho việc lập trình, mô phỏng và phát triển các ứng dụng với chip vi điều khiển MC68HC11 sau đây.

#### **Phần mềm Micro-IDE**

Phần mềm Micro-IDE là một công cụ phát triển mạnh hỗ trợ cho việc lập trình, mô phỏng cho nhiều loại chip vi điều khiển khác nhau, chạy trên môi trường Windows. Giao diện người dùng của nó như hình vẽ, với phần mềm này ta có thể lựa chọn để làm việc với các chip trong họ 8051, M68HC11, M68HC12, M68HC16, 8085, 8086,... và các chip họ AVR. Các tính năng, đặc điểm của phần

mềm này tương tự như phần mềm Keil C51 nhưng ứng dụng của nó thì rộng hơn cho nhiều loại chip kể trên. Ngôn ngữ lập trình có thể soạn thảo và biên dịch trên môi trường của Micro-IDE có thể bằng C hoặc Assembly.



Hình 1.4 - Giao diện của phần mềm Micro-IDE

### 1.2.4 Công cụ phát triển cho họ vi điều khiển PIC16/17

Họ vi điều khiển PIC16/17 được hỗ trợ bởi đầy đủ các công cụ phát triển cả về phần cứng và phần mềm sau đây :

PICMASTER® Real-Time In-Circuit Emulator.

PROMATE™ Universal Programmer.

PICSTART® Low-Cost Prototype Programmer.

PICDEM-1 Low-Cost Demonstration Board.

PICDEM-2 Low-Cost Demonstration Board.

MPASM Assembler.

MPSIM Softwave Simulator.

C Compiler (MP-C).

Fuzzy logic development system (fuzzy TECH®-MP).

Dưới đây ta sẽ đi vào giới thiệu hai công cụ phát triển phần mềm thường được sử dụng là **MPASM Assembler** và **C Compiler (MP-C)**.

### **MPASM Assembler**

MPASM có hỗ trợ đầy đủ cho việc viết tạo ra các Macro trong chương trình mã ngữ, viết chương trình bằng ngôn ngữ Assembly, cung cấp các mã nguồn có sẵn và các định dạng file khác nhau. Nó tạo ra một môi trường cho phép soạn thảo mã nguồn, biên dịch cũng như gỡ rối các chương trình của người sử dụng lập trình cho các chip vi điều khiển khác nhau bao gồm cả họ PIC16Cxx, PIC 16Cxx và PIC16C5x.

MPASM có các đặc điểm chính sau đây trong việc hỗ trợ phát triển các ứng dụng của người dùng:

- + Cho phép biên dịch chương trình từ mã Assembly sang mã đối tượng cho tất cả các vi điều khiển của hãng Microchip.
- + Cho phép tạo ra các chương trình Macro.
- + Tạo ra tất cả các file cần thiết (Object, Listing, Symbol, và các dạng đặc biệt khác) khi biên dịch và gỡ rối cho các hệ vi điều khiển của Microchip.
- + Hỗ trợ việc sử dụng và tạo ra các file định dạng Hex, Decimal và Octal.

### **C Compiler (MP-C)**

Công cụ phát triển phần mềm MP-C là một trình biên dịch hoàn chỉnh và được tích hợp vào trong môi trường phát triển cho các chip vi điều khiển họ PIC16/17 của Microchip. Trình biên dịch này có nhiều khả năng mạnh mẽ và rất dễ sử dụng, điều này khó có thể tìm thấy ở một công cụ phát triển nào khác.

Để dễ dàng cho việc sử dụng và gỡ rối, trình biên dịch MP-C sử dụng các ký hiệu thông tin thích hợp với các ký hiệu của công cụ phát triển khác là PICMASTER® Universal Emulator.



Ngoài ra hiện nay còn có một số phần mềm khác rất linh hoạt, hỗ trợ viết cho nhiều dòng PIC, từ 16/17/18 đến dsPIC bằng C như CCS C, HT PIC, HT PIC 18 hay bằng Pascal như MikroPascal ...

### 1.2.5 Công cụ phát triển cho họ vi điều khiển thông minh PSoC

Phần mềm **PSoC Designer** là một công cụ hoàn chỉnh được tích hợp vào hệ thống hỗ trợ cho việc phát triển cả về mặt lập trình cấu hình phần cứng và chương trình phần mềm cho các chip vi điều khiển thông minh chế tạo theo công nghệ PSoC của hãng **Cypress Microchip**, ví dụ như họ **CY8C25-26xxx**.

Phần mềm PSoC Designer được xây dựng trên nền hệ điều hành Windows và nó có thể chạy được trên Windows98, Windows NT 4.0, Windows 2000, Windows Millennium (Me), or Windows XP.

PSoC Designer giúp người sử dụng lựa chọn một thuật toán cấu hình phần cứng cho vi điều khiển, lập trình phần mềm cho vi điều khiển, và gỡ rối. Hệ thống này cho phép quản lý cơ sở dữ liệu bằng các *project*, tích hợp phần gỡ rối với In-Circuit Emulator, cho phép lập trình ngay trên hệ thống (In-system programming), và hỗ trợ việc tạo ra các Macro CYASM cho chương trình bằng hợp ngữ.

PSoC Designer cũng hỗ trợ việc lập trình bằng ngôn ngữ bậc cao C và biên dịch, gỡ rối các chương trình đó cho các thiết bị của hãng Cypress Microchip.

## 1.3 KIT vi điều khiển

### 1.3.1 Mục đích

Kỹ thuật vi điều khiển là môn học cơ bản quan trọng đối với sinh viên các ngành Điều khiển, Tự động hóa, Điện tử, Tin học...v.... nhằm trang bị cho sinh viên những kiến thức cần thiết về một họ vi điều khiển, ở đây là họ 8051, như cấu trúc, nguyên tắc hoạt động cùng các mạch phụ trợ, tập lệnh, cách lập trình bằng hợp ngữ, các phương thức điều khiển vào ra và các phương pháp kết nối thiết bị ngoại vi. Dựa trên nền tảng kiến thức của môn học kỹ thuật vi điều khiển, sinh viên tiếp thu những môn học kỹ thuật chuyên ngành và tiếp cận dễ dàng hơn với những kỹ thuật, trang thiết bị hiện



đại. Đồng thời cho phép sinh viên có cái nhìn trực quan và kiểm chứng, khẳng định sự đúng đắn các kiến thức đã học. Do vậy việc đưa vào giáo trình phần xây dựng thiết kế Kit vi điều khiển là hết sức quan trọng và rất thiết thực đối với sinh viên chuyên ngành tự động hóa, điện tử viễn thông, đo lường ..... Các Kit vi điều khiển đều được thiết kế dựa trên các họ vi điều khiển mà sinh viên đang học trong giáo trình này. Sinh viên có thể dùng các Kit vi điều khiển này để điều khiển các quá trình nhiệt độ, điều khiển động cơ, điều khiển các đối tượng giao thông, điều khiển các quá trình sản xuất .....

Trên thế giới có rất nhiều nguồn cung cấp các mạch Kit nói trên như :

+ Từ các hãng của nước ngoài, ví dụ như bộ KIT MDA-EMS51 của hãng Midas của Hàn Quốc, KIT và phần mềm phát triển cho họ vi điều khiển PIC của MikroElektronika (Website <http://www.mikroelektronika.co.yu/>), hay một hãng nổi tiếng khác là OLIMEX (Website <http://www.olimex.com>) với các sản phẩm KIT phát triển cho các dòng AVR, PIC, MAXQ, MSP430, ARM... Đặc điểm của các sản phẩm này là có các tính năng mạnh, mẫu mã đẹp, phục vụ tốt nhu cầu thực hành của môn học. Tuy nhiên giá thành của thiết bị là rất đắt (giá khoảng 700\$ đối với KIT MDA-EMS51, và việc đặt mua không được thuận tiện.

+ Do các trung tâm nghiên cứu và các trường đại học trong nước tự chế tạo. Trong các trường đại học (Đại học BKĐN – bộ môn tự động hóa đã chế tạo và sản xuất hoàn chỉnh KIT vi điều khiển phục vụ đào tạo và sản xuất), các thiết bị thực hành môn học vi điều khiển có thể do các cán bộ giảng dạy môn học chế tạo hoặc do các sinh viên tự chế tạo dưới sự hướng dẫn của giáo viên. Ưu điểm của các sản phẩm chế tạo trong nước là giá thành rẻ mà chất lượng, mẫu mã không thua kém nước ngoài. Cấu hình của sản phẩm và nội dung thực hành phù hợp với mục đích, chương trình đào tạo, với điều kiện học tập của sinh viên cả nước nói chung và sinh viên ĐHBK Đà Nẵng nói riêng.

Xuất phát từ tình hình thực tế và sự cần thiết đáp ứng nhu cầu thực hành trong quá trình học tập môn học vi điều khiển nói trên, việc thiết kế, xây dựng một bộ KIT vi

điều khiển với đầy đủ các thành phần cơ bản (nêu ở phần yêu cầu), là vấn đề hết sức cần thiết đối với cá sinh viên ngành tự động hóa trong trường đại học BKĐN. Đây là vấn đề trọng tâm trong giáo trình này.

### **1.3.2 Yêu cầu thiết kế KIT vi điều khiển phục vụ thí nghiệm :**

+ *Phần cứng* : Bởi vì mục đích của sản phẩm là phục vụ cho việc đào tạo nên cấu hình phần cứng phải rõ ràng, dàn trải. Kích thước của bộ KIT gọn nhẹ, dễ dàng di chuyển và bảo quản. Các thành phần của bộ KIT được thể hiện rõ ràng, dễ nhận biết, một đặc điểm nữa là nó được phân thành các module nhỏ và liên kết với module trung tâm bằng BUS, điều này cũng giúp người học có thể nhìn thấy một cách trực quan các thành phần của bài thí nghiệm sẽ tiến hành. Đồng thời việc tổ chức thành các module nhỏ cũng tiện cho việc sửa chữa, thiết kế mở rộng, và có thể dễ dàng trong việc thương mại hóa sản phẩm sau này.

+ *Các thành phần của bộ KIT* : Bộ KIT có đầy đủ các thành phần cơ bản của một hệ vi điều khiển và các nguồn tài nguyên ngoại vi phong phú, cho phép người sử dụng có thể tự do phát triển các ứng dụng của mình :

Bộ vi điều khiển trung tâm.

Các bộ nhớ ROM (nội), RAM (nội/ngoại), EEPROM(nội/ngoại).

Các cổng vào/ra tương tự, vào/ra số, các cổng ghép nối ngoại vi mở rộng lập trình được.

Giao tiếp bàn phím.

Hiển thị LCD.

Hiển thị trạng thái bằng đèn LED .

Hiển thị LED 7 thanh.

Hiển thị ma trận LED (8x8).

Các đầu vào xung.

Mạch điều khiển động cơ một chiều, động cơ bước.

Giao tiếp với máy tính PC qua cổng truyền thông nối tiếp (COM) và song song (LPT).

+ *Các bài thí nghiệm thực hiện trên bộ KIT :*

*Các bài cơ bản :* Giúp sinh viên làm quen với hệ lệnh của chip vi điều khiển, nhập các lệnh và chạy chương trình để xem kết quả thực hiện của các lệnh đó.

*Các bài nâng cao :* Gồm các bài thí nghiệm phức tạp hơn thực hiện sử dụng các thành phần ngoại vi trên bộ KIT. Các bài thí nghiệm thực hiện việc điều khiển một quá trình tương tự hoặc số, ví dụ : điều khiển hiển thị đèn LED, LED 7 thanh, ma trận LED, điều khiển vào/ra số - tương tự, điều khiển động cơ bước...

+ *Thực hiện các bài thí nghiệm :*

*Các bài tập mẫu xây dựng trước :* Đi kèm với sản phẩm bộ KIT là một hệ thống các bài tập mẫu được xây dựng cho các ứng dụng trên bộ KIT như điều khiển các thành phần ngoại vi của KIT, điều khiển vào/ra tương tự - số... Sinh viên có thể đọc hướng dẫn sử dụng và thực hiện các bài tập mẫu này trên bộ KIT.

*Các bài tập do sinh viên tự lập trình :* Sau khi đã nắm rõ cấu hình của bộ KIT và sơ đồ bộ nhớ của các thành phần trên bộ KIT, sinh viên có thể tự lập trình các bài thí nghiệm trên bộ KIT theo ý định riêng của mình.

+ *Công cụ lập trình :*

*Bằng máy tính PC :* Có thể sử dụng ngôn ngữ ASSEMBLY với trình biên dịch hợp ngữ sẵn có của nhà sản xuất (ASM51.exe) hoặc các công cụ phát triển ví dụ như phần mềm Keil, ProView32, 8051IDE hay 8051 Bascom... để lập các chương trình cho bộ KIT từ máy tính bằng ngôn ngữ C, Assembly hay BASIC, sau đó biên dịch thành file dạng Hexa hoặc Binary rồi nạp xuống cho bộ KIT thực hiện các chương trình đó.

+ *Công cụ phát triển cho bộ KIT :*

*Chương trình MONITOR* : Chương trình Monitor điều hành hoạt động của cả bộ KIT và cho phép lập trình từ máy tính. Chương trình hỗ trợ chế độ gỡ rối bằng cách chạy từng dòng lệnh (STEP BY STEP) và quan sát câu lệnh vừa chạy (vd MOV A, #1) đồng thời cũng quan sát được giá trị của các thanh ghi đặc biệt SFR, các PORT, các ô nhớ RAM. Ngoài ra chương trình Monitor của bộ KIT còn cho phép sửa đổi nội dung của các thanh ghi đặc biệt hoặc nội dung các ô nhớ trong của chip vi điều khiển chính.

*Công cụ gỡ rối* : Trên giao diện máy tính có thể sử dụng các phần mềm phát triển ví dụ phần mềm Keil, ASM51, Proview32 để gỡ rối cho việc lập trình.

*(Các phần mềm có thể tìm thấy ở thư mục Programming tools trên CD đi kèm KIT)*

cuu duong than cong . com

cuu duong than cong . com