BỘ GIÁO DỤC VÀ ĐÀO TẠO

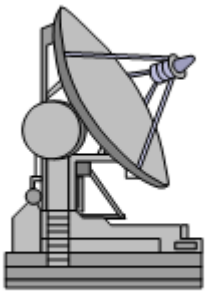
ĐẠI HỌC DÂN LẬP HẢI PHÒNG

KHOA ĐIỆN - ĐIỆN TỬ

NGÀNH ĐIỆN TỬ - VIỄN THÔNG

LUẬN VĂN TỐT NGHIỆP.

ĐỀ TÀI: Thiết kế và thi công hệ thống điều khiển nhiệt độ.



Sinh viên thực hiện: ĐỖ VĂN ĐẠT.

Giáo viên hướng dẫn: NGUYỄN VĂN DƯƠNG.

LỜI MỞ ĐẦU

Hiện nay, ngành kỹ thuật điện tử và công nghệ thông tin tiến bộ không ngừng. Chúng đang ngày càng phát triển và được ứng dụng trong tất cả các mặt của đời sống. Các thiết bị điện tử dùng Vi Điều Khiển được sử dụng rộng rãi khắp trong các ứng dụng tự động. Nó giúp chúng ta trong mọi công việc cũng như giải trí. Các bộ Vi Điều Khiển ngày càng hiện đại, tốc độ xử lý nhanh hơn, và có ứng dụng rộng hơn.

Một trong những ứng dụng quan trọng của Vi Điển Khiển đó là dùng trong đo lường và điều khiển. Nhờ các loại cảm biến, ứng dụng của đo lường bằng Vi Điều Khiển không chỉ giới hạn trong các đại lượng điện mà còn mở rộng ra các tín hiệu không phải điện. Sử dụng Vi Điều Khiển chúng ta thu thập các đại lượng cần đo dễ dàng hơn, có thể xử lý ngay các đại lượng đó và đưa ra được những kết quả như mong muốn.

Với tầm quan trọng của đo lường bằng Vi Điều Khiển nên, em đã nhận đề tài này làm đồ án tốt nghiệp để nghiên cứu, và hiểu biết thêm về Vi Điều Khiển và các ứng dụng hay của nó trong cuộc sống thường ngày của chúng ta.

Trong quá trình làm đồ án tốt nghiệp, do sự hạn chế về thời gian, tài liệu và trình độ có hạn nên không tránh khỏi có thiếu sót. Em rất mong được sự đóng góp ý kiến của thầy cô và các bạn để đồ án tốt nghiệp của em được hoàn thiện hơn.

Em xin gửi lời cảm ơn chân thành đến các thầy cô trong Khoa Điện-Điện tử, đặc biệt là thầy Nguyễn Văn Dương đã giúp đỡ em hoàn thành tốt đồ án này.

Hải phòng, 8 tháng 7 năm2009

Sinh viên thực hiện

ĐỖ VĂN ĐẠT

Mục lục

Chương 1: TỔNG QUAN VỀ VI ĐIỀU KHIỂN.....6

1.1. Bộ vi điều khiển 8051..... 7

1.2. Bộ vi điều khiển 8052..... 13

1.3. Bộ vi điều khiển 8031 13

Chương 2:VI ĐIỀU KHIỂN PIC16F877A 15

2.1. Tổng quan về thiết bị 15

2.1.1. Hình dạng và bố trí chân của Pic16F877A 15

2.1.2. Đặc tính nổi bật của bộ xử lý 15

2.1.3. Sơ đồ khối bộ vi điều khiển Pic16F877A 16

2.2. Mô tả các chân chức năng của Pic16F877A 17

2.3. Tổ chức bộ nhớ 19

2.3.1. Tổ chức bộ nhớ chương trình Flash 19

2.3.2. Tổ chức bộ nhớ dữ liệu RAM 21

2.3.3. Bộ nhớ dữ liệu EEPROM..... 24

2.3.4. Đọc và ghi vào bộ nhớ dữ liệu EEPROM.....16

2.3.5. Đọc và ghi chương trình FLASH 26

2.4. Cổng vào ra.....26

2.4.1. Cổng A và thanh ghi TRISA 27

2.4.2. Cổng B và thanh ghi TRISB.....28
2.4.3. Cổng C và thanh ghi TRISC.....29
2.4.4. Cổng D và thanh ghi TRISD.....31
2.4.5. Cổng E và thanh ghi TRISE.....31
2.5. Các bộ Timer của chip.....33
2.5.1. Bộ Timer0.....33
2.5.2. Bộ Timer1.....36
2.5.3. Bộ Timer2.....39
2.6. Bộ chuyển đổi tương tự sang số.....41
2.6.1. Bộ chuyển đổi tương tự sang số.....41
2.6.2. Lựa chọn tốc độ chuyển đổi.....43
2.7. Các ngắt của PIC16F877.....44
2.8. So sánh với Vi Điều Khiển 8051.....44
Chương 3:THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN NHIỆT ĐỘ....45
3.1. Sơ đồ khối tổng quát.....45
3.2. Khối cảm biến.....46
3.3. Khối chuyển đổi tương tự sang số.....47
3.4. Khối điều khiển.....52
3.5. Khối chuyển đổi số sang tương tự.....52
3.6. Khối điều khiển thyristor.....54
3.6.1. Sơ đồ cấu trúc.....54

3.6.2. Nguyên tắc điều khiển 55

3.6.3. Sơ đồ nguyên lý 56

3.7. Khối hiển thị LCD. 57

3.7.1. Các chân chức năng. 58

3.7.2. Sơ đồ khối của HD44780. 59

3.7.3. Tập lệnh của LCD. 63

3.8. Sơ đồ mạch hệ thống điều khiển nhiệt độ. 69

3.9. Phần mềm điều khiển 70

3.9.1. Lưu đồ thuật toán. 70

3.9.2. Chương trình. 72

Kêt luân 77

Tài liệu tham khảo 78

Chương 1

TỔNG QUAN VỀ VI ĐIỀU KHIỂN

Có 4 bộ vi điều khiển 8 bit chính. Đó là 6811 của Motorola, 8051 của Intel, z8 của Xilog và Pic 16 của Microchip Technology. Mỗi một kiểu loại trên đây đều có một tập lệnh và thanh ghi riêng duy nhất, nếu chúng đều không tương thích lẫn nhau. Cũng có những bộ vi điều khiển 16 bit và 32 bit được sản xuất bởi các hãng sản xuất chíp khác nhau. Với tất cả những bộ vi điều khiển khác nhau thì tiêu chuẩn để lựa chọn các bộ vi điều khiển là:

*) Đáp ứng được nhu cầu tính toán của bài toán một cách hiệu quả về mặt giá thành và đầy đủ chức năng có thể nhìn thấy được. Trong khi phân tích các nhu cầu của một dự án dựa trên bộ vi điều khiển chúng ta phải biết bộ vi điều khiển nào là 8 bit, 16 bit hay 32 bit có thể đáp ứng tôt nhất nhu cầu của bài toán một cách hiệu quả. Nhứng tiêu chuẩn đó là:

- Tốc độ: tốc độ lớn nhất mà vi điều khiển hỗ trợ lạ bao nhiêu.

- Kiểu đóng vỏ: Đóng vỏ kiểu DIP 40 chân hay QFP. Đây là yêu cầu quan trọng đối với yêu cầu về không gian, kiểu láp ráp và tạo mẫu thử cho sản phẩm cuối cùng.

- Công suất tiêu thụ: Điều này đặc biệt khắt khe đối với các sản phẩm dùng pin, ắc quy.

- dung lượng bộ nhớ Rom và Ram trên chíp.

- Số chân vào ra và bộ định thời trên chíp.

- Khả năng dễ dàng nângh cấp cho hiệu suất cao hoặc giảm công suất tiêu thụ.

- Giá thành cho một đơn vị: Điều này quan trọng quyết định giá thành sản phẩm mà một bộ vi điều khiển được sử dụng.

*) Cóp sẵn các công cụ phát triển phần mềm như các trình biên dịch, trình hợp ngữ và gỡ rối.

*) Nguồn các bộ vi điều khiển có sẵn nhiều và tin cậy. Khả năng sẵn sàng đáp ứng về số lượng trong hiện tại tương lai. Hiện nay các bộ vi điều khiển 8 bit họ 8051 là có số lượng lớn nhất các nhà cung cấp đa dạng như Intel, Atmel, Philip…

1.1. Bộ vi điều khiển 8051

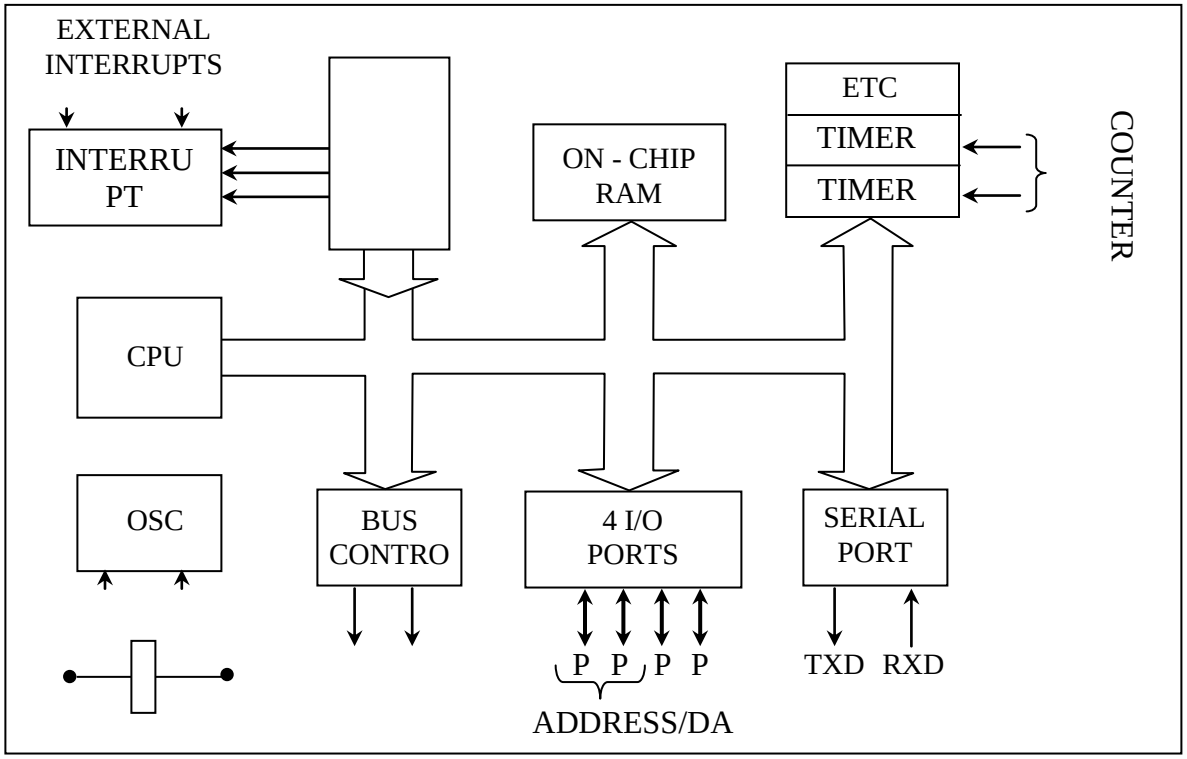
Vào năm 1981 hãng Intel giới thiệu một số bộ vi điều khiển được gọi là 8051. Bộ vi điều khiển này có 128 byte RAM, 4K byte ROM trên chíp, hai bộ định thời, một cổng nối tiếp và 4 cổng (đều rộng 8 bit) vào ra tất cả được đặt trên một chíp. Lúc ấy nó được coi là một 'hệ thống trên chíp'. 8051 là một bộ xử lý 8 bit có nghĩa là CPU chỉ có thể làm việc với 8 bit dữ liệu tại một thời điểm. Dữ liệu lớn hơn 8 bit được chia ra thành các dữ liệu 8 bit để cho xử lý. 8051 có tất cả 4 cổng vào ra I/O mỗi cổng rộng 8 bit (hình vẽ). Mặc dù 8051 có một ROM trên chíp cực đại là 64Kbyte, nhưng các nhà sản xuất lúc đó đã xuất xưởng chỉ với 4Kbyte Rom trên chíp.

8051 đã trở nên phổ biến sau khi Intel cho phép các nhà sản xuất khác nhau sản xuất và bán bất kỳ dạng biến thể nào của 8051 mà họ thích với điều kiện họ phải để lại mã tương thích với 8051. Điều này dẫn đến sự ra đời nhiều phiên bản của 8051 với các tốc độ khác nhau và dung lượng Rom trên chíp khác nhau. Điều này quan trọng là mặc dù có nhiều biến thể khác nhau của 8051 về tốc độ và dung lượng nhớ ROM trên chíp nhưng tất cả chúng đều tương thích với 8051 ban đầu về các lệnh. Điều này có nghĩa là nếu ta viết chương trình cho một phiên bản nào đó thì nó cũng sẽ chạy với mọi phiên bản bất kỳ khác mà không phân biệt nó từ hãng sản xuất nào.

Bảng 1.1. Các đặc tính của 8051 đầu tiên.

Đặc tính	Số lượng
ROM trên chíp	4Kbyte
RAM	128 byte
Bộ định thời	2
Các chân vào ra	32
Cổng nối tiếp	1
Nguồn ngắt	6

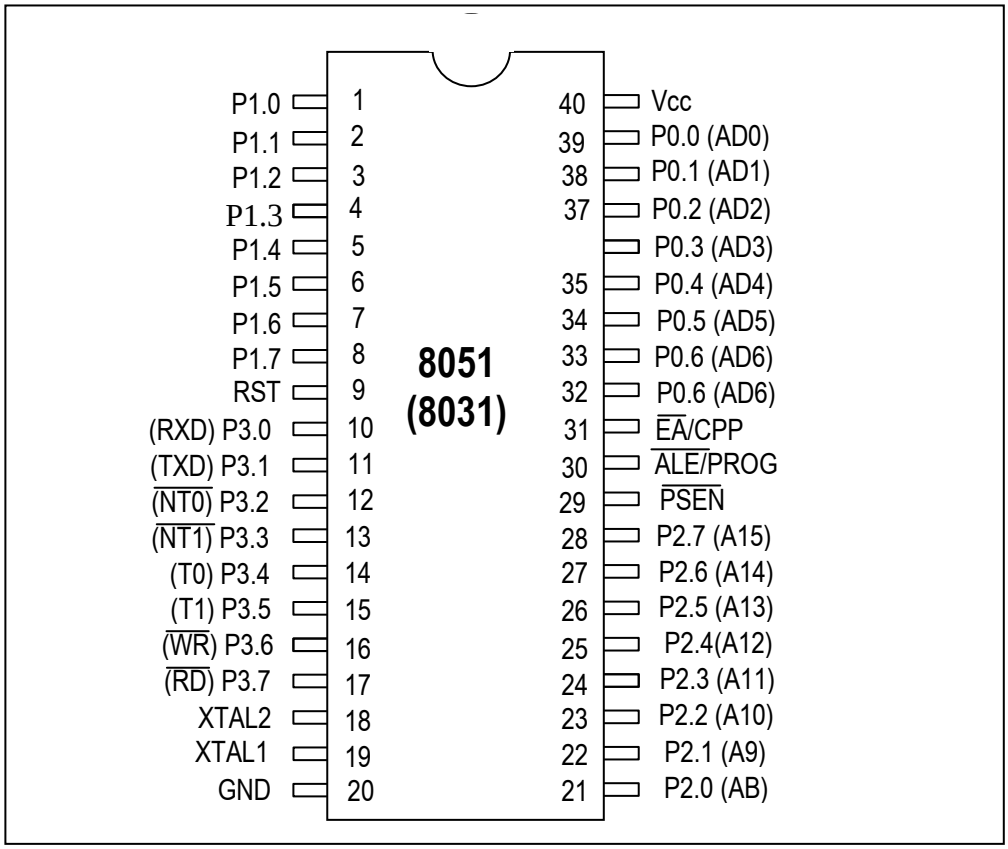
Bộ vi điều khiển 8051 là thành viên đầu tiên của họ 8051, hãng Intel ký hiệu nó là MSC51. Bảng trên là các đặc tính của họ 8051.



Hình 1.1. Bố trí bên trong của 8051

Mô tả chân của 8051như hình 1.2. Các thành viên của họ 8051(ví dụ 8751, 89C51, DS5000) đều có các kiểu đóng vỏ khác nhau, chẳng hạn như hai hàng chân DIP dạng vỏ dẹp vuông QFP và dạng chip không có chân đỡ LLC

thì chúng đều có 40 chân cho các chức năng khác nhau như vào ra I/O, đọc $\overline{RW}$, ghi $\overline{WR}$, địa chỉ, dữ liệu và ngắt. Cần lưu ý rằng một số hãng cung cấp phiên bản 8051 có 20 chân với số cổng vào ra ít hơn cho các ứng dụng yêu cầu thấp hơn. Tuy nhiên, vì hầu hết các nhà phát triển chính sử dụng chíp đóng vỏ 40 chân với hai hàng chân DIP nên ta chỉ tập chung mô tả phiên bản này.



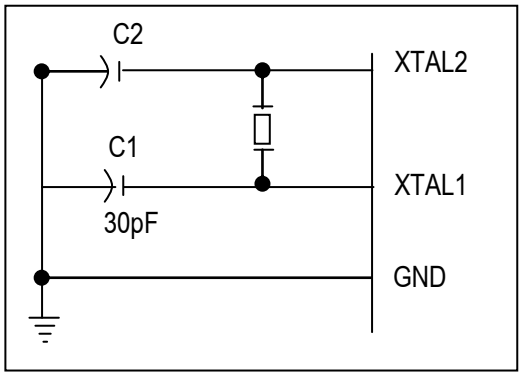
Hình 1.2. Sơ đồ chân của 8051

Từ hình 1.2. ta thấy trong 40 chân có 32 chân dùng cho các cổng P0, P1, P2, P3 với mỗi cổng có 8 chân. Các chân còn lại dành cho nguồn Vcc, đất GND, các chân dao động XTAL1 và XTAL2, khởi động lại RST

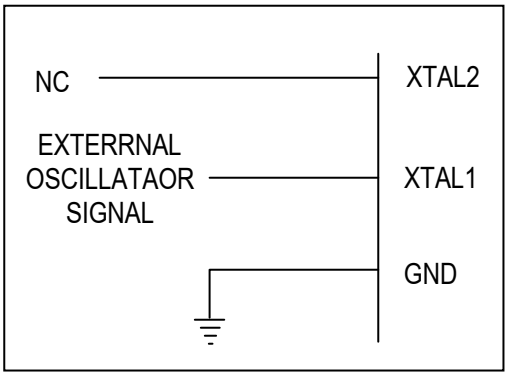
cho phép chốt địa chỉ ngoài $\overline{EA}$, cho ngắt cất chương trình $\overline{PSEN}$. Trong 8 chân này thì 6 chân Vcc, GND, XTAL1, XTAL2, RST và $\overline{EA}$ được các họ 8031 và 8051 sử dụng. Hay nói cách khác là chúng phải được nối để cho hệ thống làm việc mà không cần biết bộ vi điều khiển thuộc họ 8051 hay 8031. Còn chân $\overline{PSEN}$ và chân ALE được sử dụng trong các hệ thống dựa trên 8031.

- ✓ Chân Vcc và chân GND tương ứng với chân số 40 và chân số 20 cung cấp nguồn (+5V) và nối mass.
- ✓ Chân XTAL1 (chân 19) và XTAL2 (chân 18): 8051 có bộ dao động trên chíp nhưng nó yêu cầu có một xung đồng hồ ngoài để chạy nó. Bộ dao động thạch anh được nối với XTAL1 và XTAL2 cùng hai tụ điện có giá trị 30pF. Một phía tụ được nối xuống đất như hình 1.3.

  Cần phải lưu ý rằng có nhiều tốc độ khác nhau của họ 8051. Tốc độ được coi như là tần số cực đại của bộ giao động được nối tới chân XTAL. Ta có thể sử dụng một nguồn tần số khác dao động thạch anh chẳng hạn như bộ dao động TTL thì nó sẽ được nối tới chân XTAL1 còn chân XTAL2 để hở như hình 1.4.



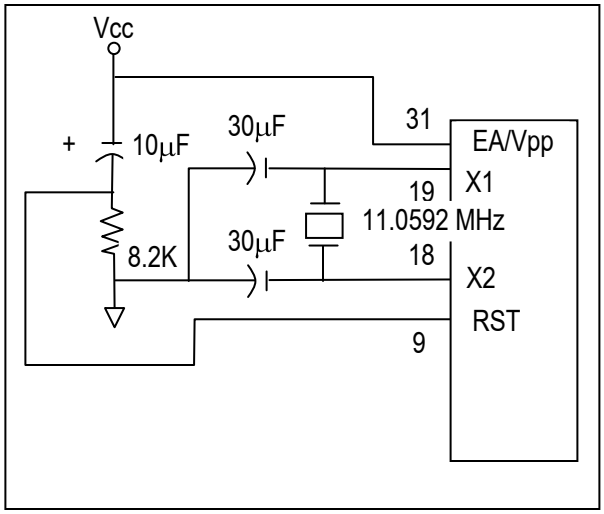
Hình 1.3. XTAL nối với 8051



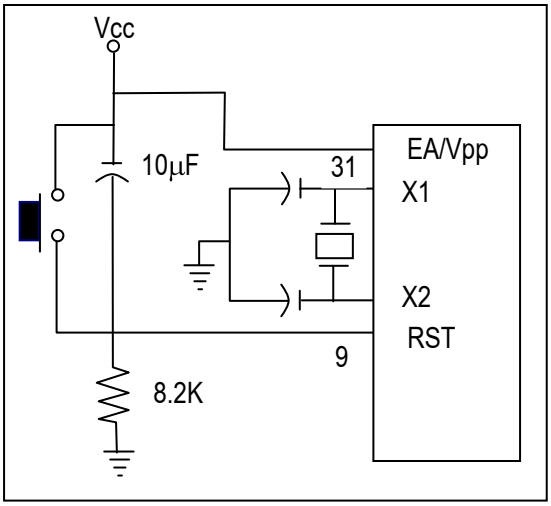
Hình 1.4. XTAL nối với dao đông ngoài

- ✓ Chân RST: Chân số 9 là chân tái lập RESET. Nó là chân đầu vào có mức tích cực cao. Khi cấp xung cao tới chân này thì bộ vi điều

khiển sẽ tái lập và kết thúc mọi hoạt động. Nó có thể coi như sự tái bật nguồn.



Hình 1.5. Mạch tái bật nguồn RESET



Hình 1.6. Mạch tái bật nguồn với Debounce.

Muốn mạch RESET làm việc có hiệu quả thì nó phải có tối thiểu 2 chu kì máy. Hay nói cách khác, xung cao phải kéo dài tối thiểu 2 chu kì máy trước khi nó xuống thấp.

- ✓ Chân $\overline{EA}$ (là chân IN): Truy cập bộ nhớ ngoài, chân số 31 trên vỏ chíp như 8751, 89C51 hoặc DS5000 thì chân $\overline{EA}$ được nối với nguồn Vcc. Trường hợp không có ROM trên chíp như 8031 và 8051 thì mã chương trình được lưu cất ở bộ nhớ ngoài, khi đó chân $\overline{EA}$ được nối đất. Như vậy chân này không bao giờ được để hở.
- ✓ Chân $\overline{\text{PSEN}}$ là chân có chức năng cho phép lưu chương trình. Ở hệ thống 8031, khi chương trình cất ở bộ nhớ ROM ngoài thì chân này được nối tới chân OE của ROM.

- ✓ ALE cho phép chốt địa chỉ là chân có mức tích cực cao. Khi nối 8031 tới bộ nhớ ngoài thì cổng 0 cũng được cấp địa chỉ và dữ liệu. Hay nói cách khác, 8031 dồn địa chỉ và dữ liệu qua cổng 0 để tiết kiệm số chân. Chân ALE được sử dụng để phân kênh địa chỉ và dữ liệu bằng cách nối tới chân G của của chíp 73LS373.
- ✓ Nhóm chân cổng vào ra I/O: bốn cổng P0, P1, P2, P3 đều có 8 chân và tạo thành cổng 8 bít. Tất cả các cổng khi RESET đều được cấu hình làm cổng ra. Để làm đầu vào thì cần được lập trình.

  Các cổng bình thường là cổng ra. Cổng P0 có thể vừa làm đầu ra, vừa làm đầu vào cổng P0 từ chân 32 đến 39 phải được nối với điện trở kéo 10K bên ngoài. Cổng P1 cũng có 8 chân, từ chân 1 đến chân 8, và có thể sử dụng làm đầu vào hoặc ra. Khác với cổng P0, cổng P1 không cần đến điện trở kéo bên ngoài vì nó đã có điện trở kéo bên trong. Cổng P2 cũng có 8 chân từ chân 21đến 28, và có thể sử dụng làm đầu vào hoặc ra. Cũng giống như cổng P1, cổng P2 không cần điện trở kéo vì bên trong đã có các điện trở kéo. Cổng P3 có 8 chân từ chân 10 đến chân 17. Cổng này có thể sử dụng làm đầu vào hoặc ra. Cũng như chân P1và P2, cổng P3 cũng không cần điện trở kéo.

  Bảng 1.2. Chức năng các chân cổng P3.

Bít cổng P3	Chức năng	Chân số
P3.0	Nhận dữ liệu (RXD)	10
P3.1	Phát dữ liệu (TXD)	11
P3.2	Ngắt 0(INT0)	12
P3.3	Ngắt 1(INT1)	13
P3.4	Bộ định thời 0 (TO)	14
P3.5	Bộ định thời 1(T1)	15
P3.6	Ghi (WR)	16
P3.7	Đọc (RD)	17

Có hai bộ vi điều khiển thành viên khác của họ 8051 là 8052 và 8031.

1.2. Bộ vi điều khiển 8052

Bộ vi điều khiển 8052 là thành viên khác của họ 8051, 8052 có tất cả các đặc tính chuẩn của 8051 ngoài ra nó có thêm 128 byte RAM và một bộ định thời nữa. Hay nói cách khác là 8052 có 256 byte RAM và 3 bộ định thời, nó cũng có 8K byte ROM trên chíp thay vì 4K byte như 8051.

Bảng 1.3. So sánh các đặc tính của các thành viên họ 8051.

Đặc tính	8051	8052	8031
ROM trên chíp	4K byte	8K byte	OK
RAM	128 byte	256 byte	128 byte
Bộ định thời	2	3	2
Chân vào - ra	32	32	32
Cổng nối tiếp	1	1	1
Nguồn ngắt	6	8	6

Qua bảng trên ta thấy thì 8051 là tập con của 8052, nên mọi chương trình viết cho 8051 đều chạy được trên 8052 nhưng điều ngược lại là không đúng.

1.3. Bộ vi điều khiển 8031

Một thành viên khác của 8051 là chíp 8031. Chíp này không có ROM trên chíp nên để sử dụng chíp này ta phải bổ sung ROM ngoài cho nó, ROM ngoài phải chứa chương trình mà 8031 sẽ nạp và thực hiện. So với 8051 mà chương trình được chứa trong ROM trên chíp bị giới hạn bởi 4K byte, còn ROM ngoài chứa chương trình được gắn vào 8031 thì có thể lớn đến 64K byte. Khi bổ xung cổng, như vậy chỉ còn lại hai cổng để thao tác. Để giải quyết vấn đề này ta có thể bổ xung cổng vào ra cho 8031 bằng cách phối ghép 8031 với bộ nhớ và cổng vào ra chẳng hạn với chíp 8255. Ngoài ra còn có các phiên bản khác nhau về tốc độ của 8031 từ các hãng sản xuất khác nhau.

Bảng 1.4. Các phiên bản của 8051 từ Atmel

Số linh kiện	ROM	RAM	Chân I/O	Timer	Ngắt	Vcc	Đóng vỏ
AT89C51	4K	128	32	2	6	5V	40
AT89LV51	4K	128	32	2	6	3V	40
AT89C1051	1K	64	15	1	3	3V	20
AT89C2051	2K	128	15	2	6	3V	20
AT89C52	8K	128	32	3	8	5V	40
AT89LV52	8K	128	32	3	8	3V	40

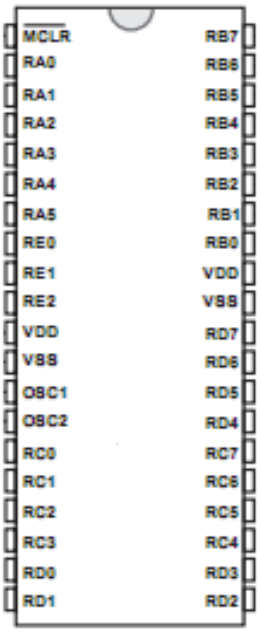
Chương 2.

VI ĐIỀU KHIỂN PIC16F877A

Ngày nay, các bộ điều khiển đang có ứng dụng rộng rãi trong các lĩnh vực khoa học kỹ thuật và đời sống xã hội, đặc biệt là trong tự động hoá và điều khiển. Giờ đây với nhu cầu chuyên dụng hoá, tối ưu hoá về thời gian không gian giá thành, bảo mật, tính chủ động trong công việc, ... ngày càng đòi hỏi khắt khe. Và dòng vi điều khiển Pic đã đáp ứng tốt các yêu cầu đó.

2.1. Tổng quan về thiết bị

2.1.1. Hình dạng và bố trí chân của Pic16F877A



Hình 2.1. Hình dạng Pic16F877A

2.1.2. Đặc tính nổi bật của bộ xử lý.

- ✓ Sử dụng công nghệ tích hợp cao RICSC CPU
- ✓ Người sử dụng có thể lập trình với các câu lệnh đơn giản

✓ Tất cả các câu lệnh thực hiện trong 1 chu kì ngoại trừ một số lệnh rẽ nhánh thực hiện trong 2 chu kì.

✓ Tốc độ hoạt động là : - Xung đồng hồ vào la DC-20MHz

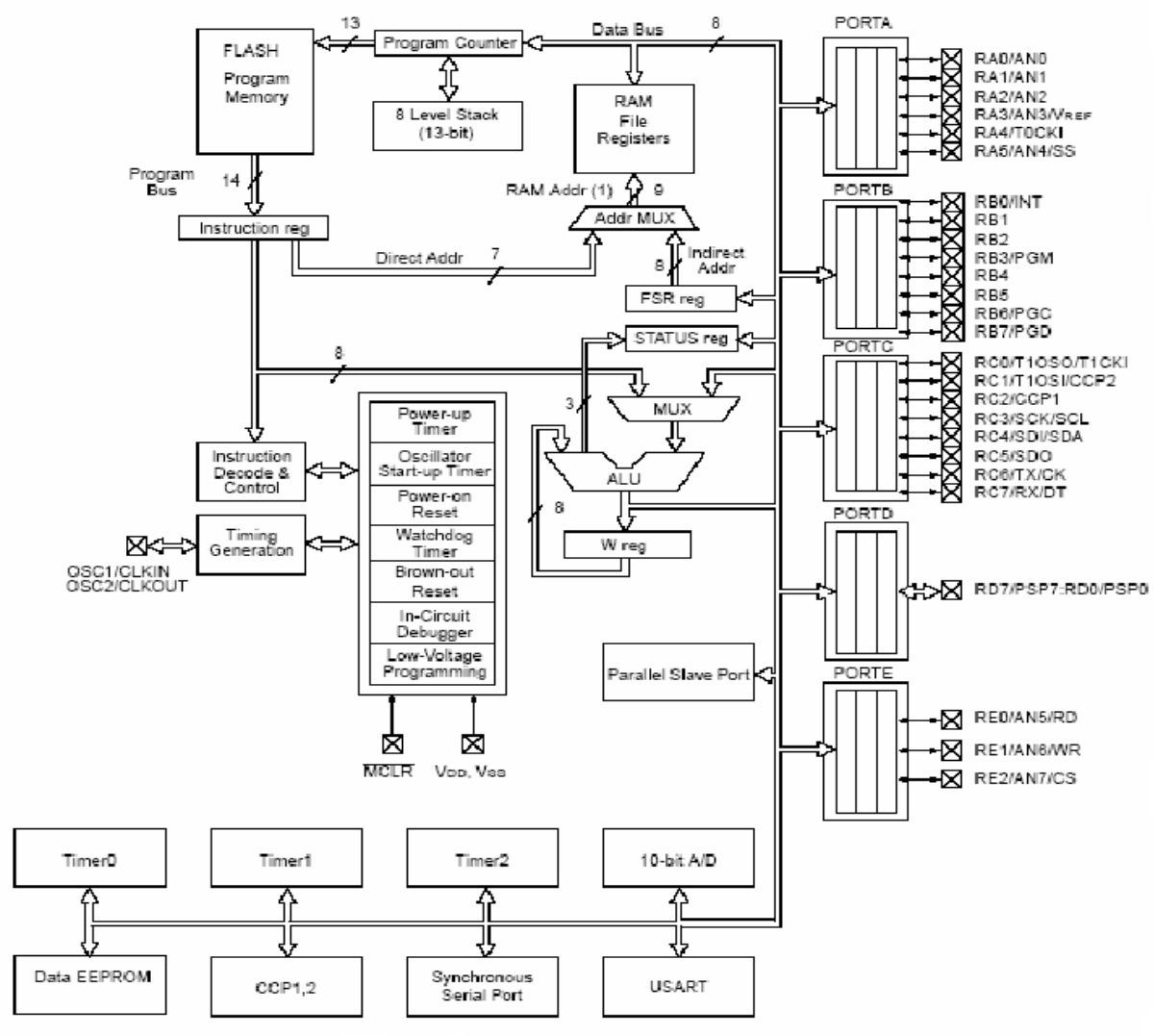
- Chu kì lệnh thực hiện trong 200ns

✓ Bộ nhớ chương trình Flash 8Kx14 Words

✓ Bộ nhớ Ram 368x8 bytes

✓ Bộ nhớ EFPROM 256x8 bytes.

2.1.3. Sơ đồ khối bộ vi điều khiển Pic16F877A



Hình 2.2. Sơ đồ khối của Pic16F877A

2.2. Mô tả các chân chức năng của Pic16F877A

Bảng 2.1. Bảng chân chức năng của Pic16F877A

Tên chân	Chân số		Chức năng của chân
OSC1/CLKIN	13	I	Đầu vào của dao động thạch anh/ngõ vào xung clock ngoại.
OSC2/CLKOUT	14	O	Đầu ra của bộ dao động thạch anh. Nối với thạch anh hay cộng hưởng trong chế độ dao động của thạch anh. Trong chế độ RC, ngõ ra của chân OSC2
MCLR /VPP	1	I/P	Ngõ vào của Master Clear (Reset) hoặc ngõ vào điện thế được lập trình. Chân này cho phép tín hiệu RESET thiết bị tác động ở mức thấp.
			PORTA là port vào ra hai chiều.
RA0/AN0	2	I/O	RA0 có thể làm ngõ vào tương tự thứ 0.
RA1/AN1	3	I/O	RA1 có thể làm ngõ vào tương tự thứ 1.
RA2/AN2/VREF –	4	I/O	RA2 có thể làm ngõ vào tương tự thứ 2 hoặc điện áp chuẩn tương tự âm.
RA3/AN3/VREF +	5	I/O	RA3 có thể làm ngõ vào tương tự thứ 3 hoặc điện áp chuẩn tương tự dương.
RA4/T0CKI	6	I/O	RA4 có thể làm ngõ vào xung clock cho bộ định thời Timer0. Hoặc làm đầu ra.
RA5/ SS /AN4	7	I/O	RA5 có thể làm ngõ vào tương tự thứ 4 hoặc làm đầu ra.
			PORTB là port vào ra hai chiều.

RB0/INT	33	I/O	RB0 có thể làm chân ngắt ngoài.
RB1	34	I/O	
RB2	35	I/O	
RB3/PGM	36	I/O	
RB4	37	I/O	
RB5	38	I/O	
RB6/PGC	39	I/O	
RB7/PGD	40	I/O	
			PORTC là port vào ra hai chiều.
RC0/T1OSO/T1CKI	15	I/O	RC0 có thể là ngõ ra của bộ dao động Timer1 hoặc ngõ vào xung clock cho Timer1.
RC1/T1OSI/CCP2	16	I/O	RC1 có thể là ngõ vào của bộ dao động Timer1 hoặc ngõ vào Capture2/ngõ ra compare2/ngõ ra PWM2.
RC2/CCP1	17	I/O	RC2 có thể là ngõ vào Capture1/ngõ ra compare1/ngõ vào PWM1.
RC3/SCK/SC	18	I/O	RC3 có thể là ngõ vào xung clock đồng bộ nối tiếp/ngõ ra trong cả hai chế độ SPI và I2C.
RC4/SDI/SDA	23	I/O	RC4 có thể là dữ liệu bên trong SPI (chế độ SPI) hoặc dữ liệu I/O (chế độ I2C).
RC5/SDO	24	I/O	RC5 có thể là dữ liệu ngoài SPI (chế độ SPI).
RC6/TX/CK	25	I/O	RC6 có thể là chân truyền không đồng bộ USART hoặc đồng bộ với xung đồng

RC7/RX/DT	26	I/O	hồ. RC7 có thể là chân nhận không đồng bộ USART hoặc đồng bộ với dữ liệu.
RD0/PSP0 RD1/PSP1 RD2/PSP2 RD3/PSP3 RD4/PSP4 RD5/PSP5 RD6/PSP6 RD7/PSP7	19 20 21 22 27 28 29 30	I/O I/O I/O I/O I/O I/O I/O I/O	PORT là port vào ra hai chiều hoặc là parallel slave port khi giao tiếp với bus của bộ vi xử lý.
RE0/ RD/AN5 RE1/WR /AN6 RE2/CS /AN7	8 9 10	I/O I/O I/O	PORTE là port vào ra hai chiều. RE0 có thể điều khiển việc đọc parallel slave port hoặc là ngõ vào tương tự thứ 5. RE1 có thể điều khiển việc ghi parallel slave port hoặc là ngõ vào tương tự thứ 6. RE2 có thể điều khiển việc chọn parallel slave port hoặc là ngõ vào tương tự thứ 7.
VSS	12,31	P	mass
VDD	11,32	P	Cung cấp nguồn dương cho các mức logic và những chân I/O.

Các kí hiệu: I: input O: output I/O:input/output P: power

2.3. Tổ chức bộ nhớ

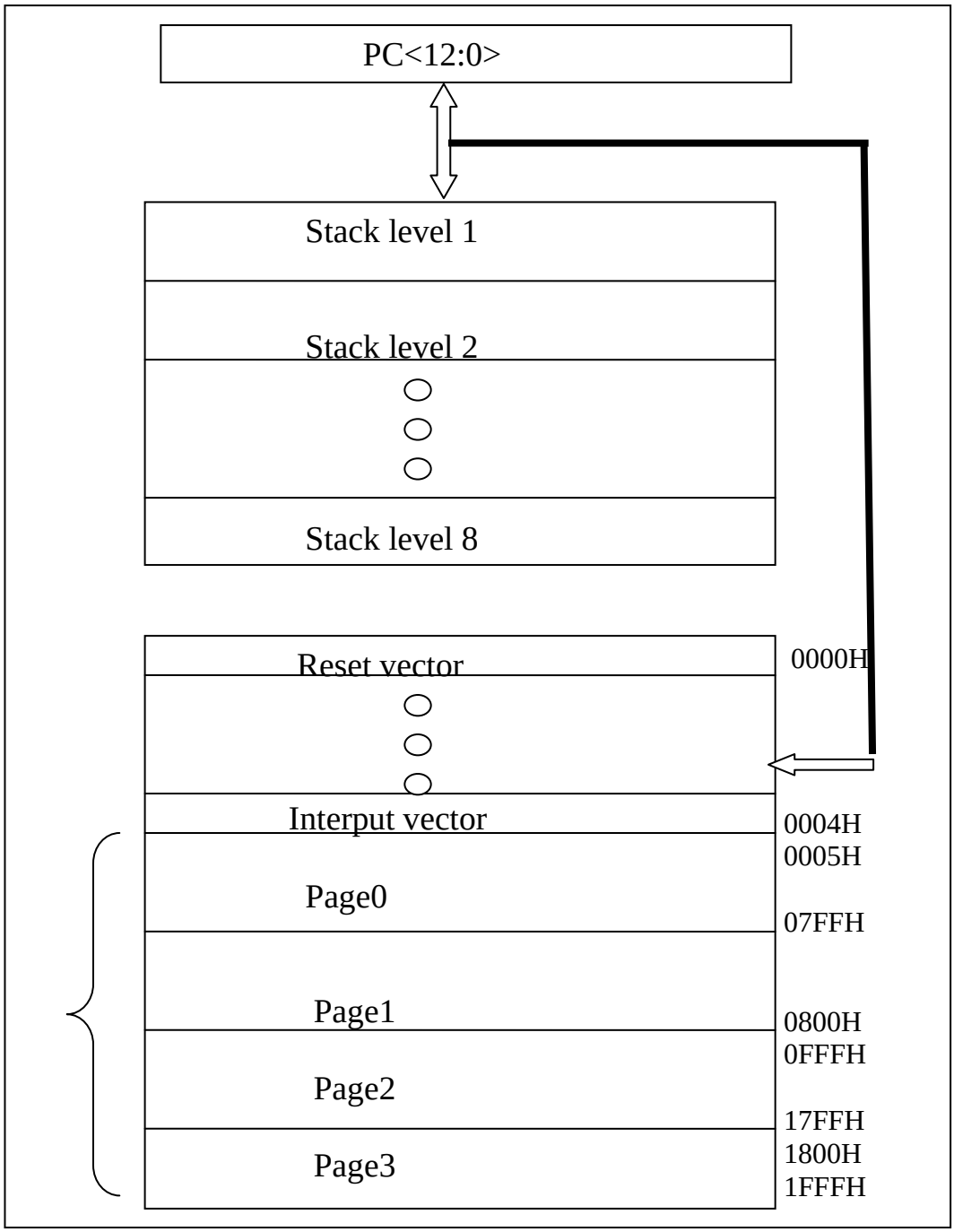
Pic16F877A có 3 khối bộ nhớ: Bộ nhớ chương trình Flash, bộ nhớ dữ liệu RAM, bộ nhớ EEPROM.

2.3.1. Tổ chức bộ nhớ chương trình Flash

Vi điều khiển Pic16F877A có bộ nhớ chương trình 13 bit và có 8Kx14 từ mã của bộ nhớ chương trình Flash, được chia thành 4 trang mỗi trang 2Kx14 từ mã.

Khi Reset địa chỉ bắt đầu thực hiện chạy là 0000h, vector ngắt bắt đầu từ 0004h.

Stack có 8 mức dùng để lưu địa chỉ lệnh thực hiện tiếp theo sau lệnh CALL và khi xẩy ra ngắt.

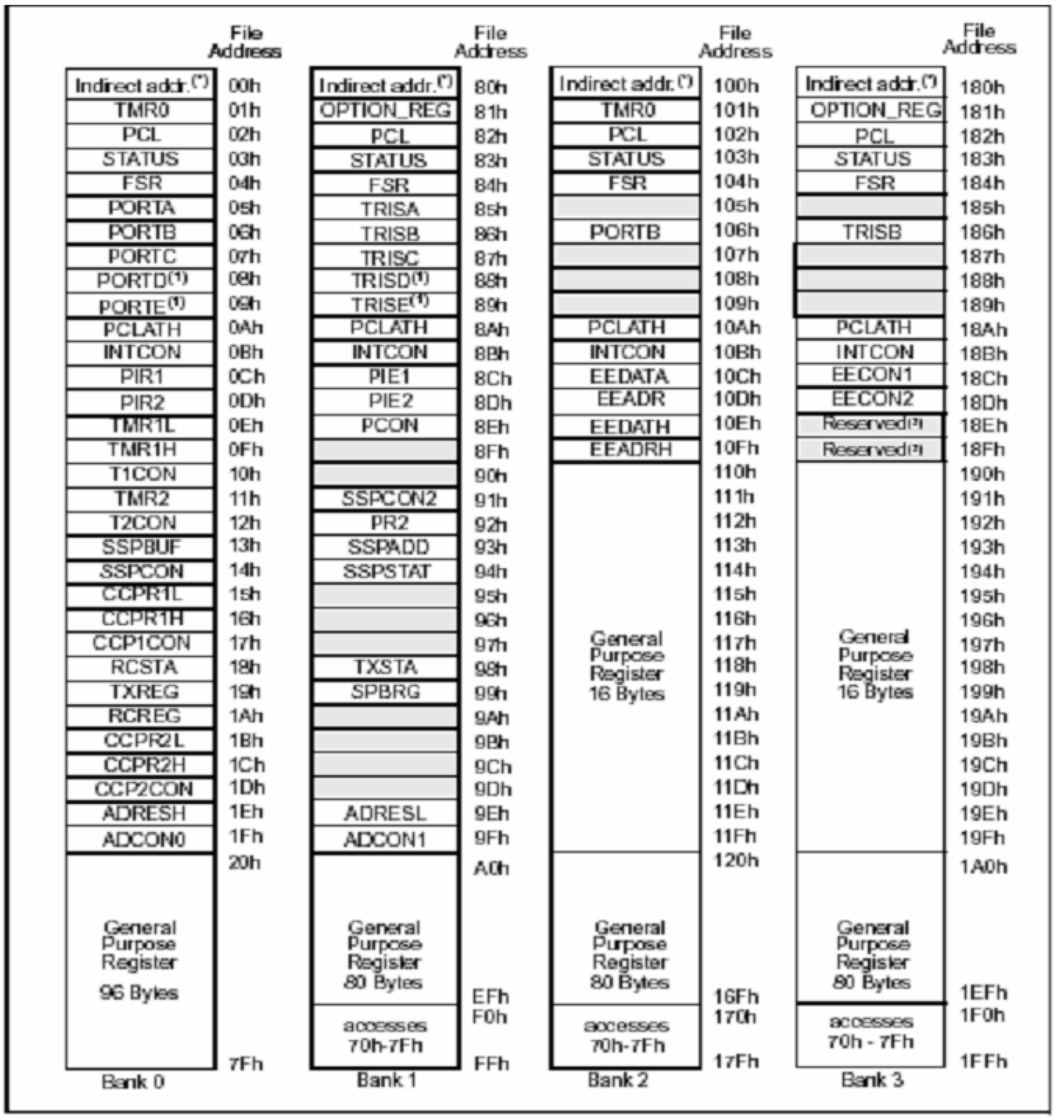


Hình 2.3. Bản đồ bộ nhớ chương trình và các ngăn xếp

2.3.2. Tổ chức bộ nhớ dữ liệu RAM

RAM là bộ nhớ có thể đọc/ghi, nó không lưu dữ liệu khi mất điện, bộ nhớ RAM của Pic16F877A có 4 Bank, mỗi Bank có dải địa chỉ 0-7FH (128

byte) trên các Bank những thanh ghi đa mục đích, nó hoạt động như một RAM tĩnh và những thanh ghi chức năng đặc biệt ở vùng địa chỉ thấp.



Hình 2.4. Hình ảnh các Bank

Các Thanh ghi đa mục đích (General Purpose Register), các thanh ghi này được truy cập bằng cả hai cách trực tiếp hoặc gián tiếp qua thanh ghi FSR, tổng cộng có 368 bytes.

Các thanh ghi chức năng đặc biệt: các thanh ghi này được dùng bởi CPU và các khối ngoại vi để điều khiển sự hoạt động theo yêu cầu của thiết bị. Các thanh ghi này có thể phân loại vào bộ phận trung tâm (CPU) và ngoại vi.

Các thanh ghi trạng thái STATUS: có 4 thanh ghi trạng thái trên 4 dãy, tại các địa chỉ 03h, 83h, 103h, 108h. Các thanh này cho biết trạng thái của phần tử logic toán học ALU, trạng thái Reset, trạng thái của các bít lựa chọn dãy thanh ghi cho bộ nhớ dữ liệu.

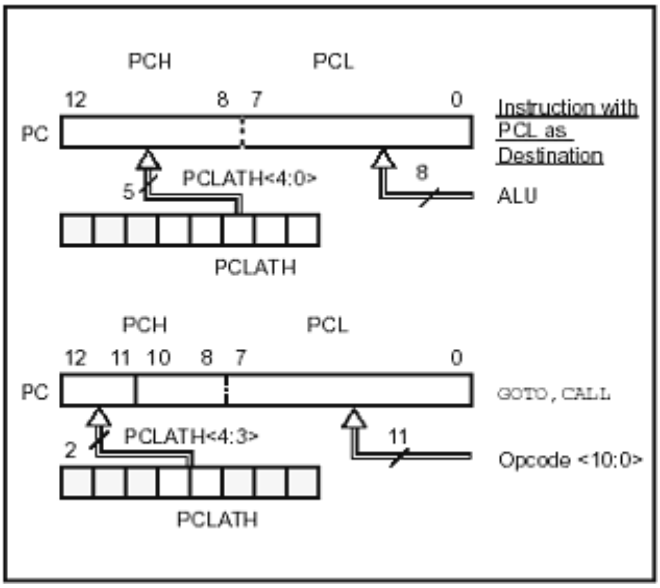
Thanh ghi trạng thái có thể là kết quả của một số lệnh như là với một số thanh ghi khác. Nếu thanh ghi trạng thái là kết quả bởi một lệnh mà tác động đến các bít Z, DC, C thì việc ghi vào các bit này là không thể.

Các thanh ghi lựa chọn OPTION_REG: có hai thanh ghi lựa chọn tai các địa chỉ 81h và 181h, các thanh ghi này có thể đọc hoặc ghi, nó chứa đựng nhiều bits điều khiển khác nhau để xác định hệ số định trước TMR0, hệ số định sau WDT, ngắt ngoài INT, TMR0, các điện áp treo cổng B.

Các thanh ghi INTCON: có 4 thanh ghi INTCON tại địa chỉ 0Bh, 8Bh, 10Bh, 18Bh, các thanh ghi này có thể đọc và ghi, nó chứa đựng nhiều sự cho phép và các bits cờ cho việc tràn thanh ghi TMR0, các ngắt thay đổi cổng RB và chân ngắt ngoài RB0/INT.

Thanh ghi PIE1: tại địa chỉ 8Ch chứa đựng các bít cho phép riêng lẻ cho các ngắt ngoại vi CCP2, ngắt xung đột tuyến SSP và EEPROM ghi các hoạt động ngắt.

Thanh ghi PCON (Power Control): chứa bít cờ cho phép phân biệt giữa việc Reset hệ thống (POR) để Reset MCLR ngoại với Reset WDT.



Hình 2.5. Hình ảnh nạp PCLATH tới PC

PLC và PCLATH: chương trình đếm chỉ rõ địa chỉ của lệnh tiếp theo được thực hiện. PC có độ rộng 13 bit, byte thấp được gọi là thanh ghi PLC, thanh ghi này có thể đọc hoặc ghi toàn bộ sự cập nhật của nó thông qua thanh ghi PCLATH.

2.3.3. Bộ nhớ dữ liệu EEPROM

Các bộ nhớ này có thể đọc và ghi trong khi các hoạt động vẫn diễn ra một cách bình thường. Bộ nhớ dữ liệu không trực tiếp sắp xếp dữ liệu trên các thanh ghi dữ liệu còn trống. Thay vì đó là ghi các địa chỉ gián tiếp qua các thanh ghi chức năng đặc biệt. Có 6 thanh ghi SFR dùng để đọc và ghi bộ nhớ chương trình và bộ nhớ dữ liệu EEPROM đó là các thanh ghi:

EECON 1	EEDATH
EECON 2	EEADR
EEDATA	EEADRH

Bộ nhớ dữ liệu EEPROM cho phép đọc và ghi các byte. Khi có tác động tới khối bộ nhớ dữ liệu. Thanh ghi EEDATA giữ 8 bít dữ liệu để đọc/ghi và thanh ghi EEADR giữ địa chỉ vị trí của EEPROM được truy cập. Các thanh ghi EEDATH và EEADRH không được sử dụng để truy cập dữ liệu EEPROM. Các thiết bị này có tới 256 byte của dữ liệu EEPROM với địa chỉ từ 00h tới FFh.

Bộ nhớ chương trình cho phép đọc và ghi các ký tự. Khi tác động đến khối chương trình nhớ, các thanh ghi EEDATH, EEDATA có dạng 2 byte ký tự giữa 14 bit dữ liệu để đọc/ghi và các thanh ghi EEADRH, EEADR có dạng hai bit từ mã với 13 bit địa chỉ của vị trí EEPROM được truy cập. Nhưng thiết bị này có thể có tới 8K từ mã của chương trình EEPROM với một địa chỉ giới hạn từ 0h tới 3FFh.

Thanh ghi địa chỉ có thể đánh địa chỉ lớn nhất là 256 byte của dữ liệu EEPROM hoặc lớn nhất là 8K ký tự của chương trình FLASH. Khi lựa chọn giá trị một địa chỉ được ghi tới thanh ghi EEADR.

Các thanh ghi EECON1 và EECON2:

EECON1 là thanh ghi điều khiển cho việc nhập dữ liệu bộ nhớ.

EECON2 không phải là thanh ghi vật lý. Khi đọc thanh ghi EECON2 sẽ đọc toàn bộ là 0. Thanh ghi EECON2 được sử dụng dành riêng cho việc ghi một cách trình tự vào bộ nhớ.

Bit điều khiển EEPGD xác định nếu việc nhập dữ liệu sẽ là nhập một chương trình hoặc nhập một bộ nhớ dữ liệu. Khi xoá, một số hoạt động tiếp theo sẽ hoạt động trên bộ nhớ dữ liệu. Khi đặt, một số hoạt động tiếp theo sẽ hoạt động trên bộ chương trình.

Các bít điều khiển RD và RW kích hoạt các hoạt động đọc và ghi theo thứ tự. Trong phần mềm những bit này không thể bị xoá, chỉ được đặt. Chúng bị xoá trong phần cứng khi mà hoạt động ghi/đọc được hoàn thành. Việc không

thể xoá bit RW trong phần mềm ngăn ngừa sự kết thúc bất ngờ hoặc kết thúc sớm của hoạt động ghi.

2.3.4. Đọc và ghi vào bộ nhớ dữ liệu EEPROM.

Để đọc một vị trí bộ nhớ dữ liệu, ta phải ghi địa chỉ vào thanh ghi EEADR xoá bít điều khiển EEPGD (EECON1<7>) sau đó đặt bit điều khiển RD (EECON1<0>). Dữ liệu có thể được đọc bởi lệnh tiếp theo. EEDATA sẽ giữ giá trị này cho tới khi có hoạt động đọc dữ liệu khác hoặc tới khi được ghi.

Ghi vào bộ nhớ dữ liệu EEPROM thì đầu tiên địa chỉ phải được ghi vào thanh ghi EEADR và dữ liệu ghi vào thanh ghi EEDATA.

2.3.5. Đọc và ghi chương trình FLASH.

Đọc một vị trí bộ nhớ chương trình có thể thực hiện bởi việc ghi 2 byte địa chỉ vào thanh ghi EEADR và EEADRH, đặt bit điều khiển EEPGD (EECON1<7>) và sau đó đặt bít điều khiển RD (EECON1<0>). Chỉ khi bit điều khiển đọc được đặt, vi xử lý sẽ sử dụng chu trình lệnh thứ hai để đọc dữ liệu.

Dữ liệu đó sẽ có trong chu trình thứ 3, trong các thanh ghi EEDATA và EEDATH, do đó nó có thể được đọc là 2 byte trong các lệnh tiếp theo. Dữ liệu có thể được đưa ra ngoài của EEDATH, EEDATA bắt đầu với lệnh thứ 3 sau lệnh BSF EECON1, RD. Và thanh ghi EEDATA và EEDATH sẽ giữ giá trị này cho tới khi có hoạt động đọc một giá trị khác hoặc có hoạt động ghi.

Ghi một vị trí bộ nhớ chương trình có thể được thực hiện bởi việc ghi thanh ghi 2 byte địa chỉ vào các thanh ghi EEADR và EEADRH, ghi dữ liệu 13 bit vào thanh ghi EEDATA và EEDATH.

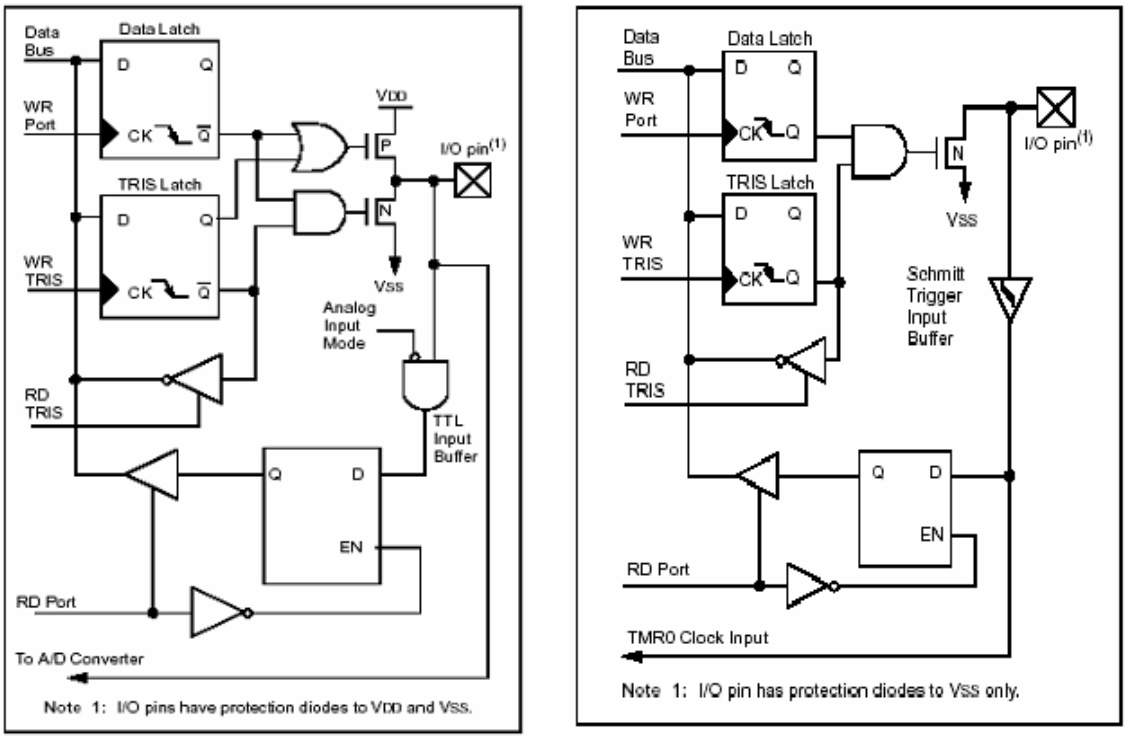
2.4. Cổng vào ra

Một số chân của các cổng vào/ra được tích hợp với những thiết bị ngoại vi. Nhìn chung khi thiết bị ngoại vi hoạt động, các chân có thể không sử dụng với mục đích làm chân vào ra.

2.4.1. Cổng A và thanh ghi TRISA

Cổng A là cổng hai chiều với độ rộng đường truyền là 6 bit. Để điều khiển việc truy xuất dữ liệu người ta dùng thanh ghi TRISA. Nếu đặt bít TRISA=1 thì lúc này cổng A sẽ có các chân là chân vào. Và ngược lại sẽ là các chân xuất. Việc đọc cổng A chính là đọc trạng thái các chân, trong đó việc xuất phải qua việc xuất các chốt của cổng. Các chân của cổng A chủ yếu được sử dụng với mục đích chính là nhận tín hiệu tương tự hoặc làm chân vào/ra. Riêng chân RA4 có thể đa hợp với chân vào bộ Timer0 và khi đó nó trở thành chân RA4/TOCKI. Chân này như một đầu vào Schmitt Trigger và nó mở một đầu ra. Các chân khác của cổng A là chân vào với bộ TTL. Việc điều khiển các chân này thông qua việc đặt hay xoá các bít của thanh ghi ADCON1. Thanh ghi TRISA điều khiển trực tiếp các chân của cổng A, khi sử dụng các chân này để nhận tín hiệu tương tự vào ta phải chắc chắn rằng các bít của thanh ghi TRISA đã được đặt rồi.

Sơ đồ khối chân RA3÷RA0, chân RA5 và của chân RA4/TOCKI của cổng A:



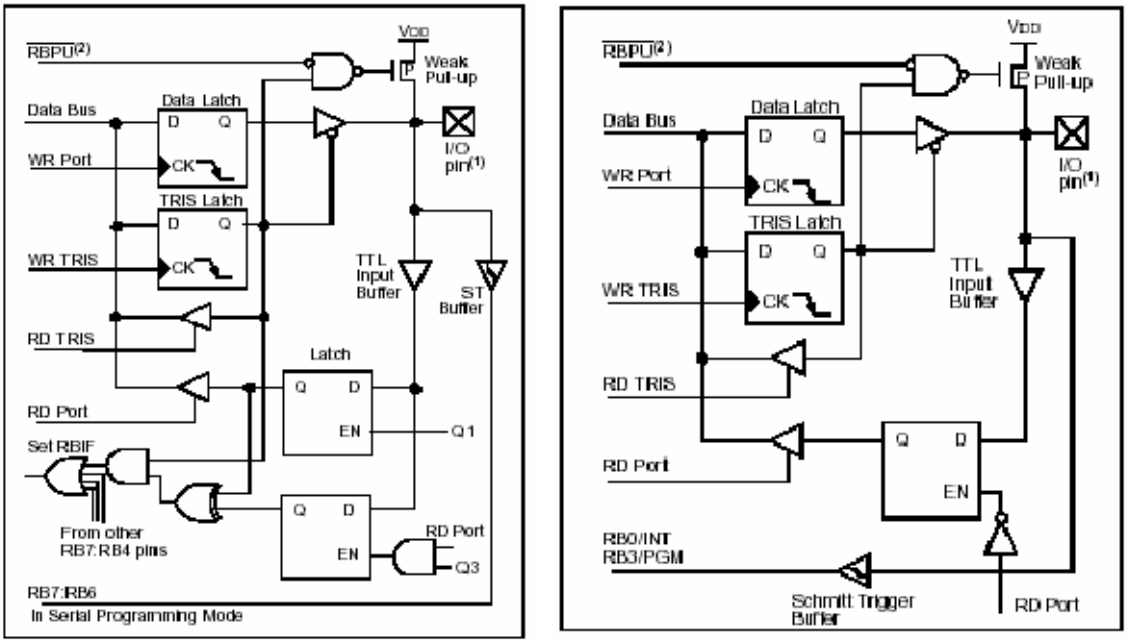
Hình 2.6. Sơ đồ khối chân cổng A

2.4.2. Cổng B và thanh ghi TRISB

Cổng B là cổng hai chiều với độ rộng đường truyền là 8 bít. Tương ứng với nó để điều khiển trực tiếp dữ liệu ta sử dụng thanh ghi TRISB. Nếu đặt bít TRISB=1 thì lúc này các chân của cổng B được định nghĩa là chân vào. Nếu xoá bít TRISB=0 thì lúc này các chân của cổng B được định nghĩa là chân ra. Nội dung của chốt ra có thể chọn trên mỗi chân.

Các chân của cổng B có thể đa hợp với các chương trình vận hành bằng điện áp thấp. Đó là các chân sau: RB3/PGM, RB6/PGC, RB7/PGD. Sự thay đổi hoạt động của những chân này được miêu tả ở trong phần đặc tính nổi bật. Mỗi chân của cổng B sẽ có một khả năng dừng bên trong nhưng yếu. Điều này được trình bày ở việc xoá bít RBPU (bít 7 của thanh ghi OPTION_REG). Khả năng dừng này sẽ tự động tắt đi khi các chân của cổng được định nghĩa là chân

ra. Khả năng dừng này sẽ tự động mất khi ta RESET. Bốn chân của cổng B, từ RB7 đến RB4 có đặc tính là ngắt khi thay đổi trạng thái. Chỉ những chân được định dạng là những chân vào thì ngắt này mới tồn tại. Một vài chân RB7÷RB4 được định dạng như chân ra, nó thi hành ngắt trên sự thay đổi so sánh. Chân vào RB7÷RB4 được so sánh với giá trị cũ của chốt ở lần đọc cuối cùng của cổng B. Sự ghép đôi không khớp chân ra của RB7÷RB4 bằng lệnh OR làm phát ra ngắt với cờ bít RBIF của thanh ghi INTCON. Ngắt này có thể khởi động thiết bị từ trạng thái SLEEP.



Hình 2.7. Sơ đồ khối của chân RB3 đến RB0, chân RB7:RB4 của cổng B

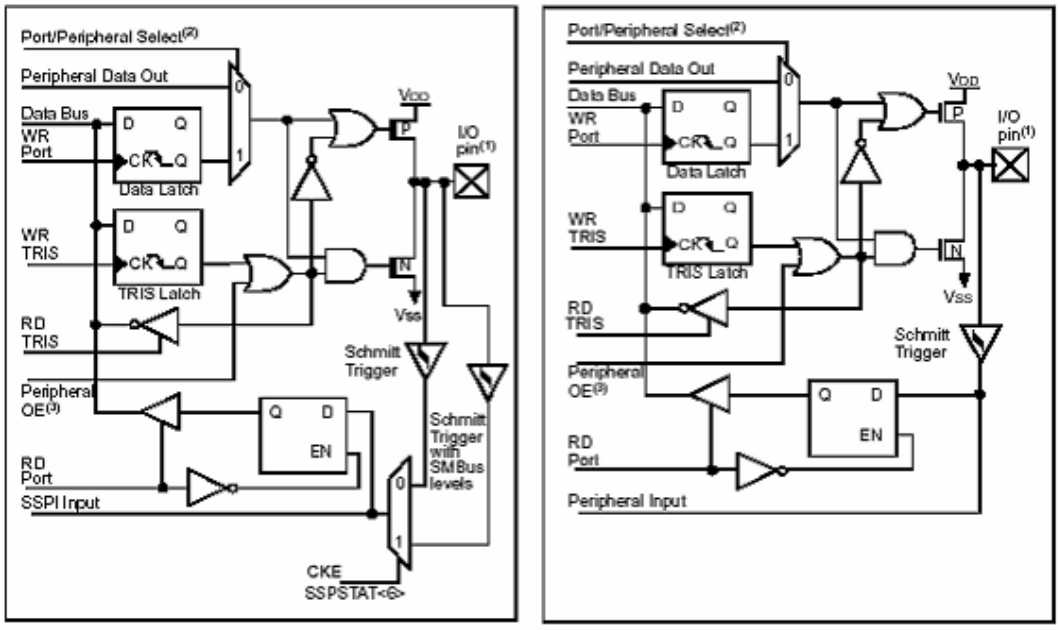
2.4.3. Cổng C và thanh ghi TRISC

Cổng C là cổng hai chiều với độ rộng đường truyền là 8 bít. Tương ứng với việc điều khiển nó là thanh ghi TRISC. Nếu đặt bít TRISC=1 thì tương ứng với chân của cổng C là chân vào. Nếu ta xoá bít TRISC=0 thì tương ứng với nó chân của cổng C là chân ra. Đặt nội dung của chốt ra có thể đặt trên chân chọn.

Cổng C đa hợp với việc vận hành thiết bị ngoại vi. Chân của cổng C thông qua bộ đệm Schmitt Trigger đầu vào.

Khi chế độ I^2C hoạt động, thì các chân của cổng PORTC<4:3> có thể được sắp xếp với mức I^2C thường hoặc với mức SMBUS bằng cách sử dụng bít CKE (SSPSTAT<6>) là bít 6 của thanh ghi SSPSTAT.

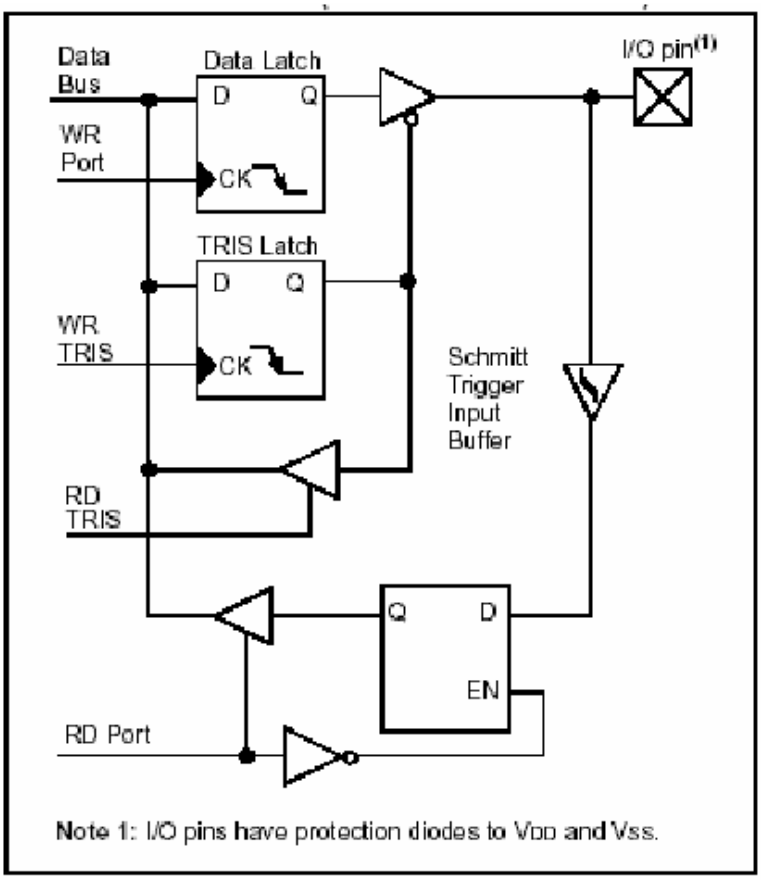
Khi vận hành các thiết bị ngoại vi bằng việc xác định bít TRIS của mỗi chân cổng C. Một số phần phụ có thể ghi đè lên bít TRIS làm cho chân này sẽ trở thành chân ra, trong khi đó thì một số phần phụ khác lại ghi đè lên bít TRIS làm cho chân này trở thành chân vào. Từ khi những bít TRIS ghi đè thì trong việc tác động trong các thiết bị ngoại vi là có thể, những lệnh đọc - sửa - ghi (BSF, BCF, XORWF) với thanh ghi TRISC như là nơi gửi tới sẽ được tránh. Người sử dụng nên đề cập tới việc phân chia kết nối các thiết bị ngoại vi cho việc đặt chính xác các bít TRIS.



Hình 2.8. Sơ đồ khối chân RC <0:2> RC <5:7> và chân RC <3:4> cổng C

2.4.4. Cổng D và thanh ghi TRISD

Cổng D có 8 bít có bộ đệm đầu vào Schmitt Trigger. Mỗi chân được sắp xếp riêng lẻ như đầu vào hoặc đầu ra. Cổng D cũng có thể được sắp xếp như là một cổng vi xử lý 8 bít (cổng phụ song song) bằng việc đặt bít điều khiển PSPMODE (TRISE<4>) và trong chế độ này vùng đệm đầu vào là TTL.



Hình 2.9. Sơ đồ khối cổng D

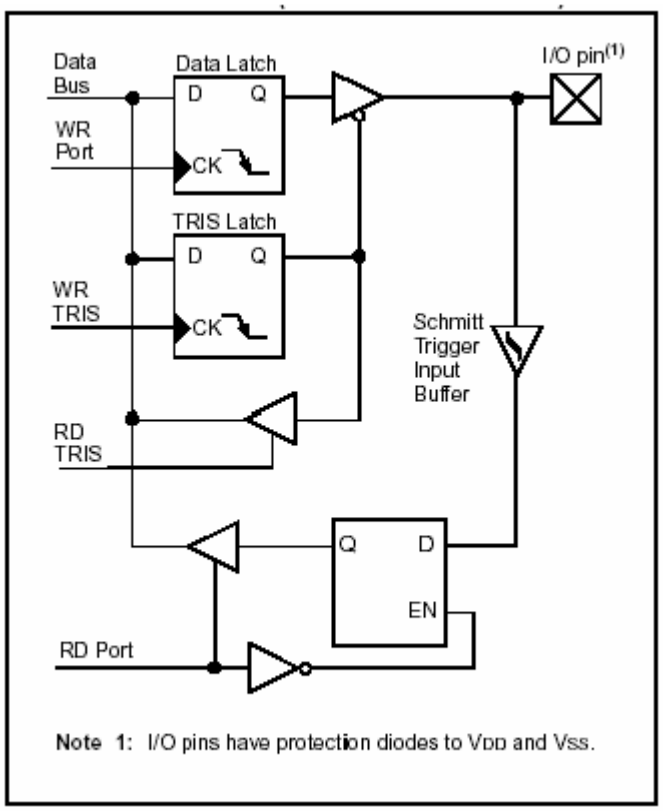
2.4.5. Cổng E và thanh ghi TRISE

Cổng E có 3 chân là RE0/RD/AN5, RE1/WR/AN6, RE2/CS/AN7. Các chân này có thể sắp xếp riêng lẻ là các đầu vào hoặc đầu ra, và các chân có vùng đệm đầu vào là các mạch Schmitt Trigger.

Cổng vào/ra E trở thành đầu vào điều khiển cho cổng vi xử lý khi bít PSPMODE (TRISE<4>) được đặt. Và trong chế độ này phải chắc chắn rằng các bít TRISE<2:0> được đặt (các chân được định dạng là các đầu vào số), thanh ghi ADCON1 phải được định dạng cho việc số vào/ra và vùng đệm đầu vào là TTL.

Các chân cổng E cũng được tích hợp với các đầu vào tương tự và trong trường hợp này các chân sẽ đọc là "0".

Thanh ghi TRISE điều khiển trực tiếp các chân RE, ngay cả khi chúng được dùng là các đầu vào tương tự.



Hình 2.10. Sơ đồ khối của cổng E

2.5. Các bộ Timer của chip.

Bộ vi điều khiển PIC16F87X có 3 bộ Timer đó là: Tmer0, Tmer1, Tmer2

2.5.1. Bộ Timer0

Là bộ định thời hoặc bộ đếm có những ưu điểm nổi bật sau:

+ 8 bít cho Timer hoặc bộ đếm
+ Có khả năng đọc và viết
+ Có thể dùng đồng hồ bên trong hoặc bên ngoài
+ Có thể chọn sườn xung của xung đồng hồ
+ Có hệ số chia cho xung đầu vào có thể lập trình lại bằng phần mềm
+ Ngắt tràn

Hoạt động của Timer0:

Timer 0 có thể hoạt động như một bộ định thời hoặc một bộ đếm. Việc chọn bộ định thời hoặc bộ đếm có thể được xác lập bằng việc xoá hoặc đặt bít TOCS của thanh ghi OPTION_REG<5>.

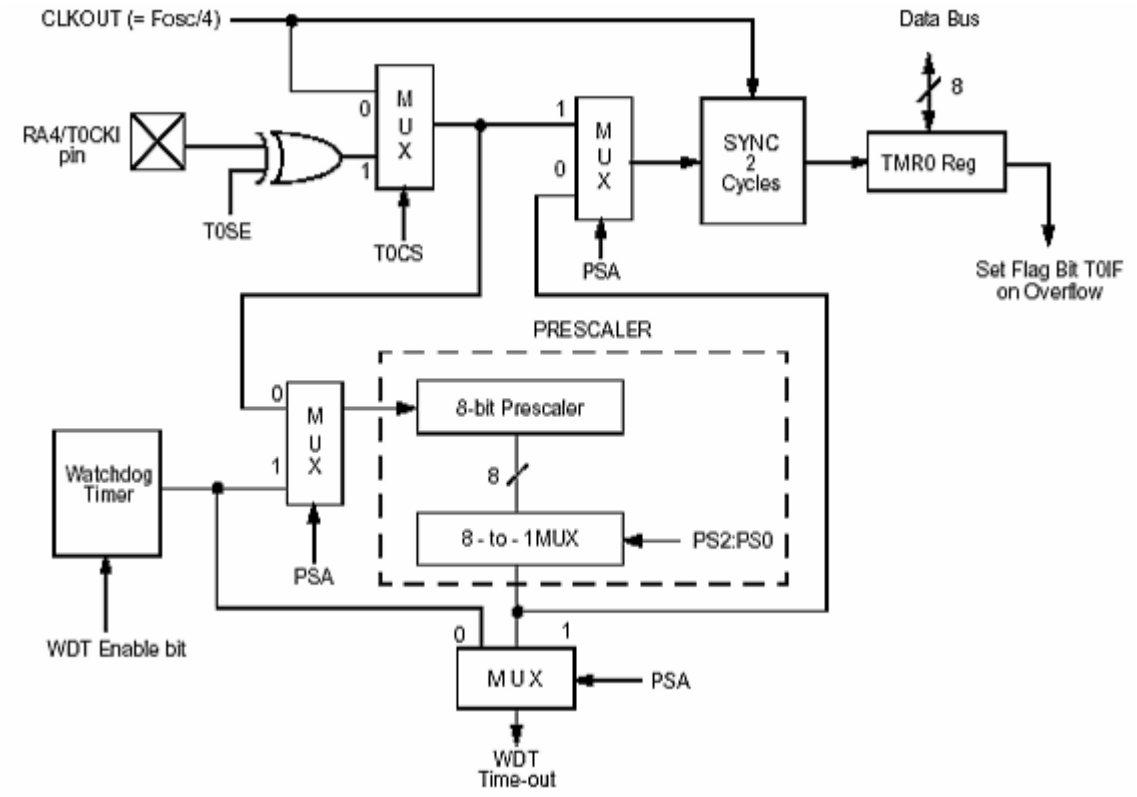
Nếu dùng hệ số chia xung đầu vào thì xoá bít PSA của thanh ghi OPTION_REG<3>.

Trong chế độ bộ định thời được lựa chọn bởi việc xoá bít T0CS (OPTION_REG<5>), nó sẽ được tăng giá trị sau một chu kỳ lện nếu không chọn hệ số chia xung đầu vào. Và giá trị của nó được viết tới thanh ghi TMR0.

Chế độ đếm được lựa chọn bởi việc đặt bít T0CS (OPTION_REG<5>). Trong chế độ bộ đếm, nó sẽ được tăng ở xung đi xuống nếu xoá bít T0SE (OPTION_REG<4>) hoặc ở xung đi lên nếu đặt bít T0SE. Và giá trị của nó được viết tới thanh ghi TMR0.

Khi dùng xung clock bên ngoài cho bộ định thời Timer0 và không dùng hệ số chia clock đầu vào Timer0 thì phải đáp ứng các điều kiện cần thiết để có

thể hoạt động đó là phải bảo đảm xung clock bên ngoài có thể đồng bộ với pha xung clock bên trong (Tosc).



Hình 2.11. Sơ đồ khối của Timer0 và WDT:

*) Các hệ số chia

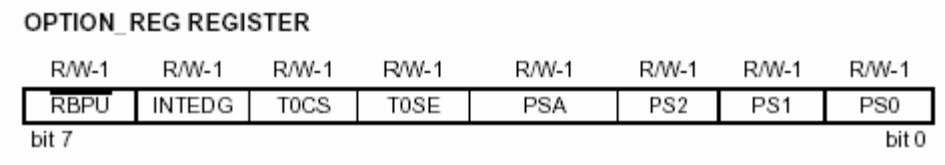
Hệ số chia dùng cho Timer0 hoặc bộ WDT. Các hệ số này không có khả năng đọc và khả năng viết. Để chọn hệ số chia xung vào Timer0 hoặc cho bộ WDT ta tiến hành xoá hoặc đặt bít PSA của thanh ghi OPTION_REG<3>.

Những bít PS2, PS1, PS0 của thanh ghi OPTION_REG<2:0> dùng để xác lập các hệ số chia.

*) Ngắt của bộ Timer 0

Ngắt của bộ Timer 0 được phát sinh ra khi thanh ghi TMR0 bị tràn tức từ FFh quay về 00h. Khi đó bít T0IF của thanh ghi INTCON<2> sẽ được đặt. Bít này

phải được xoá bằng phần mềm nếu cho phép ngắt bít T0IE của thanh ghi INTCON<5> được đặt. Timer0 bị dừng hoạt động ở chế độ SLEEP ngắt Timer 0 không đánh thức bộ xử lý ở chế độ SLEEP.



Hình 2.12. Thanh ghi OPTION_REG

Bít 5 TOCS lựa chọn nguồn clock

1 = Clock ngoài từ chân T0CKI

0 = Clock trong Focs/4

Bít 4 T0SE lựa chọn sườn xung clock

1 = Timer 0 tăng khi chân T0CKI từ cao xuống thấp(sườn xuống)

0 = Timer 0 tăng khi chân T0CKI từ thấp lên cao(sườn xuống)

Bít 3 PSA gán bộ chia xung đầu vào

1 = gán bộ chia Prescaler cho WDT

0 = gán bộ chia Prescaler cho Timer 0

Bít 2÷0 PS2÷PS1 lựa chọn hệ số chia xung vào theo bảng sau

Bảng 2.2. Lựa chọn hệ số chia xung

PS2÷PS0	Timer0	WDT
000	1:2	1:1
001	1:4	1:2

010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

2.5.2. Bộ Timer1

Bộ Timer1 có thể là bộ đếm hoặc bộ định thời với ưu điểm sau:

+ 16 bít cho bộ đếm hoặc bộ định thời (gồm hai thanh ghi TMR1H, TMR1L).

+ Có khả năng đọc và viết

+ Có thể chọn xung đồng hồ bên trong hoặc bên ngoài

+ Có thể ngắt khi tràn FFFFh về 0000h

Timer1 có một thanh ghi điều khiển, đó là thanh ghi T1C0N. Bộ Timer1 có hoạt động hay không hoạt động là nhờ việc đặt hoặc xoá bít TMR1ON (T1CON<0>).

*) Hoạt động của bộ Timer1

Nó có thể hoạt động ở một trong các chế độ sau:

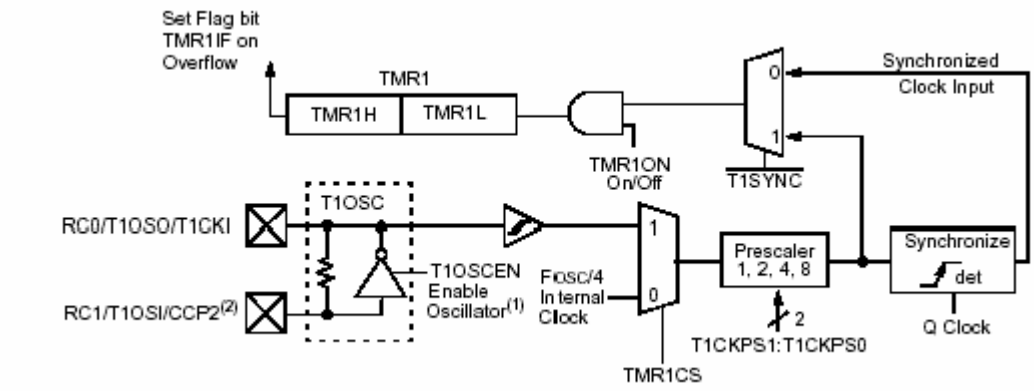
+ Là một bộ định thời 16 bít.

+ Là một bộ đếm có đồng bộ.

+ Là một bộ đếm không có đồng bộ.

Phương thức hoạt động của bộ này được xác định bởi việc chọn nguồn xung vào Timer1. Nguồn xung đồng hồ được chọn bởi việc đặt hoặc xoá bít TMR1CS (T1CON<1>). Ở chế độ bộ định thời, đầu vào là clock trong Fosc/4, bít đồng bộ T1SYNC (T1CON<2>) không có tác dụng vì clock trong luôn đồng bộ. Chế độ bộ đếm hoạt động hai chế độ: Có đồng bộ xung vào xoá bít

T1SYNC (T1CON<2>), không đồng bộ xung vào đặt bít T1SYNC (T1CON<2>) Timer1 tăng ở sườn khi xung đầu vào.



Hình 2.13. Sơ đồ khối bộ timer1

Khi bộ dao động Timer1 cho phép hoạt động thì các chân RC/T1OSI/CCP2, RC0/T1OSO/T1CKI trở thành chân vào.

Ở chế độ đếm có đồng bộ, bộ đếm tăng mỗi khi sườn lên ở chân RC0 hoặc ở chân RC1 nếu bít T1OSCEN xoá và xung vào phải đồng bộ với clock trong, ở chế độ này bộ đếm không tăng trong trong trạng thái SLEEP.

Ở chế độ bộ đếm không đồng bộ Timer1 tăng mỗi khi sườn lên ở chân RC0 hoặc ở chân RC1 nếu bít T1OSCEN xoá, ở chế độ này bộ đếm tiếp tục tăng trong trạng thái SLEEP và có khả năng tràn gây ra ngát khi đó bộ xử lý được đánh thức.

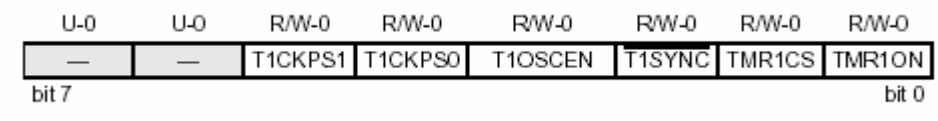
*) Dao động của Timer1

Mạch dao động thạch anh được xây dựng giữa 2 chân T1OSI và T1OS0. Khi dao động được cung cấp ở chế độ công suất thấp thì tần số cực đại của nó sẽ là 200kHz và ở chế độ SLEEP nó cung cấp ở tần số 32kHz.

*) Ngắt của bộ Timer1

Cặp thanh ghi TMR1H và TMR1L tăng từ giá trị 0000h đến giá trị FFFFh đến giá trị này tiếp tục tăng thì tràn và quay lại giá trị 0000h. Và ngắt

xuất hiện khi tràn quá giá trị FFFFh khi này cờ ngắt TMR1IF sẽ được đặt. Ngắt có thể hoạt động hoặc không hoạt động nhờ việc đặt xoá bít TMR1I.



Hình 2.14. Thanh ghi điều khiển Timer1

*) Thanh ghi điều khiển Timer1 T1CON:

Bít 7, 6 không sử dụng

Bít 5, 4 T1CKPS1÷T1CKPS0 lựa chọn hệ số chia xung vào

Bảng 2.3. Lựa chọn hệ số chia xung

T1CKPS1÷T1CKPS0	
00	1:1
01	1:2
10	1:4
11	1:8

Bít 3 T1OSCEN bít điều khiển bộ dao động Timer1

1 = Bộ dao động hoạt động

0 = Bộ dao động không hoạt động

Bít 2 bít điều khiển xung clock ngoài đồng bộ khi TMR1CS = 1

Bít2 = 0 có đồng bộ clock ngoài

Bít2 = 1 không đồng bộ clock ngoài khi TMR1CS = 0 bít này không có tác dụng

Bít 1 TMR1CS lựa chọn nguồn xung clock vào

TMR1CS = 1 clock từ chân RC0/T1OSO/T1CKI(sườn lên)

Bít 0 bít bật tắt Timer

1 = Timer 1 enable

0 = Timer 1 disable

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

2.5.3. Bộ Timer2

Bộ Timer 2 có những đặc tính sau đây:

+ 8 bít cho bộ định thời (thanh ghi TMR2)

+ 8 bít vòng lặp (thanh ghi PR2)

+ Có khả năng đọc và viết ở cả hai thanh ghi nói trên

+ Có khả năng lập trình bằng phần mềm tỷ lệ trước

+ Có khả năng lập trình bằng phần mềm tỷ lệ sau

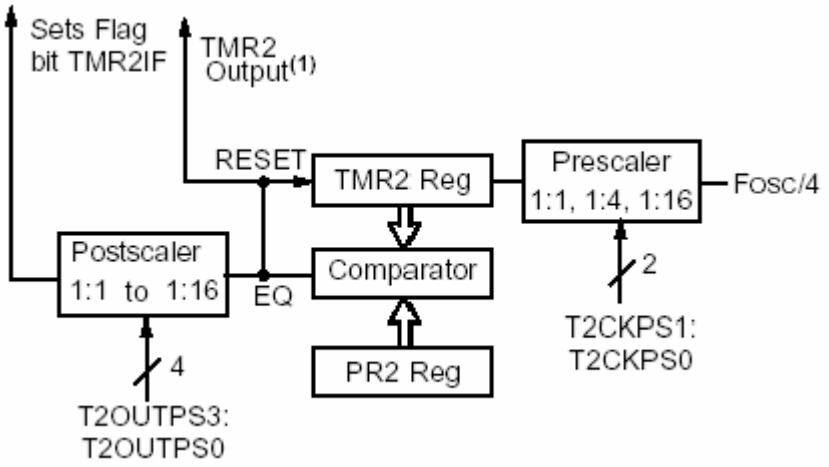
Chế độ SSP dùng đầu ra của TMR2 để tạo xung clock. Timer2 có một thanh ghi điều khhiển đó là thanh ghi T2CON. Timer2 có thể tắt bằng việc xoá bít TMR2CON của thanh ghi T2CON

*) Hoạt động của bộ Timer2

Timer2 được dùng chủ yếu ở phần điều chế xung của bộ CCP, thanh ghi TMR2 có khả năng đọc và viết, nó có thể xoá bằng việc reset lại thiết bị. Đầu vào của xung có thể chọn các tỷ lệ sau: 1:1, 1:2 hoặc 1:16 việc chọn các tỷ này có thể điều khiển các bít sau T2CKPS1 và bít T2CKPS0.

*) Ngắt của bộ Timer2

Bộ Timer2 có 1 thanh ghi 8 bít PR2. Timer2 tăng từ giá trị 00h cho đến khớp với PR2 và tiếp theo nó sẽ reset lại giá trị 00h và lệnh kế tiếp thực hiện. Thanh ghi PR2 là một thanh ghi có khả năng đọc và khả năng viết. Thanh ghi PR2 bắt đầu từ giá trị FFh đầu ra của TMR2 là đường dẫn của cổng truyền thông đồng bộ, nó được dùng để phát các xung đồng hồ.



Hình 2.15 Sơ đồ khối bộ timer2

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Hình 2.16. Thanh ghi điều khiển timer2

*) Thanh ghi TCON2

Bít 7 không sử dụng

Bít 6÷3 TOUTPS3÷TOUTPS0 bít lựa chọn hệ số đầu ra Timer2

0000 = 1:1

0001 = 1:2

0010 = 1:3

…

1111 = 1:16

Bít 2 TMR2ON bít bật tắt hoạt động Timer2

1= enable

0 = disable

Bít 1-0 T2CKPS1-T2CKPS0 chọn hệ chia đầu vào

00 = 1:1

01 = 1:4

1x = 1:16

2.6. Bộ chuyển đổi tương tự sang số.

2.6.1. Bộ chuyển đổi tương tự sang số

Bộ chuyển đổi tương tự sang số có 8 kênh (với Pic16F877A).

Tín hiệu tương tự được nạp vào bộ nạp và giữ điện dung. Tín hiệu ra điển hình và giữ điện dung duy trì là đầu vào bộ chuyển đổi. Đầu ra bộ chuyển đổi A/D là 10 bít. Bộ chuyển đổi A/D có sự chuyển điện thế cao và thấp đầu vào được lựa chọn trong phần mềm để có sự kết hợp của Vdd, Vss, RA2, RA3.

Bộ chuyển đổi A/D có 4 thanh ghi. Đó là những thanh ghi:

A/D thanh ghi kết quả cao(ADRESH)

A/D thanh ghi kết quả thấp (ADRESL)

Thanh ghi điều khiển chuyển đổi A/D (ADCON0)

Thanh ghi điều khiển chuyển đổi A/D (ADCON1)

*) Thanh ghi 8.1: thanh ghi ADCON0 (địa chỉ 1Fh)

Bít 7-6: ADCS1-ADCS0 Những bít lựa chọn đồng hồ chuyển đổi A/D

00 = Fosc/2

01 = Fosc/8

10 = Fosc/32

11 = Frc(đồng hồ xuất phát từ bên trong bộ chuyển đổi A/D dao động RC)

Bít 5-3: CHS2-CHS0 Bít chọn kênh tương tự.

000 = kênh 0(RA0/AN0)

001 = kênh 1(RA1/AN1)

010 = kênh 2(RA1/AN2)

011 = kênh 3(RA3/AN3)

100 = kênh 4(RA5/AN4)

101 = kênh 5(RE0/AN5)

110 = kênh 6(RE1/AN6)

111 = kênh 7(RE2/AN7)

Bít 2: GO/DONE bít trạng thái chuyển đổi A/D

Nếu ADON = 1 chuyển đổi A/D đang thực hiện (đặt bít này để bắt đầu quá trình chuyển đổi)

ADON=0 chuyển đổi A/D tắt và ngừng hoạt động.

*) Thanh ghi 8.2: thanh ghi ADCONN1 (địa chỉ 9Fh)

Bít 7 (ADFM): bít lựa chọn kết quả định dạng.

Bít 6-4: Người dùng định nghĩa.

Bít 3-0: Bít điều khiển sắp xếp cổng chuyể đổi A/D.

Thanh ghi ADRESH:ADRESL chứa đựng 10 bít kết quả của chuyển đổi A/D. Khi chuyển đổi A/D là hoàn thành kết quả được nạp vào thanh ghi kết quả chuyển đổi A/D. Bít GO/DONE (ADCON0<2>) được xoá và bít cờ ngắt chuyển đổi A/D là ADIF được đặt.

Sau đó bộ chuyển đổi A/D được sắp xếp như mong muốn. Lựa chọn kênh phải đạt được trước khi chuyển đổi bắt đầu. Kênh vào tương tự phải có bít TRIS tương ứng được lựa chọn như là đầu vào.

Những bước cần làm khi thực hiện chuyển đổi A/D:

1. Lựa chọn cấu hình A/D.

+ Đặt cấu hình tương tự cho chân vào A/D

+ Lựa chọn kênh vào chuyển đổi A/D (ADCON0).

+ Lựa chọn đồng hồ chuyển đổi A/D.

+ Bật bộ chuyển đổi A/D (ADCON0).

2. Lựa chọn cấu hình ngắt cho A/D.

+ Xoá bít ADIF.

+ Đặt bít ADIE.

+ Đặt bít PEIE.

+ Đặt bít GIE.

3. Đợi phụ thuộc thời gian đạt được.
4. Bắt đầu chuyển đổi.

+ Đặt bít GO/DONE (ADCON0).

5. Đợi cho chuyển đổi A/D hoàn thành.

+ Thăm dò bít GO/DONE để xoá (với thực hiện ngắt) hoặc đợi cho ngắt chuyển đổi A/D.

6. Đọc kết quả chuyển đổi trên cặp thanh ghi (ADRESH:ADRESL) xoá bít ADIF nếu quy định.
7. Cho chuyển đổi kế tiếp, thực hiện bước 1 hoặc bước 2 theo quy định. Tốc độ chuyển đổi A/D qui định như là chu kỳ Tad. Giá trị nhỏ nhất đợi của 2 chu kỳ được quy định trước khi bắt đầu kế tiếp.

2.6.2. Lựa chọn tốc độ chuyển đổi.

Tốc độ chuyển đổi là được định như là Tad. Quy định thời gian chuyển đổi A/D nhỏ nhất 12 Tad cho 10 bít chuyển đổi. Nguồn của thời gian chuyển đổi lựa chọn trong phần mềm. Có thể lựa chọn một trong các giá trị sau: 2Tosc, 8Tosc, 32Tosc, dao động RC trong bộ chuyển đổi A/D (2 đến 6 µs).

Để cho việc chuyển đổi đúng, thời gian chuyển đổi Tad phải được lựa chọn để chắc chắn Tad nhỏ nhất 1.6 µs.

Chú ý:

Nguồn RC có thời gian chu kỳ Tad 4µs nhưng có thể trong khoảng 2-6µs.

Khi tần số thiết bị lớn hơn 1MHz bộ chuyển đổi A/D nguồn đồng hồ khởi tạo cho SLEEP hoạt động.

2.7. Các ngắt của PIC16F877

PIC16F877 có 14 nguồn ngắt, thanh ghi INTCON là thanh ghi điều khiển các ngắt, mỗi ngắt có một bít cờ ngắt và một bít cho phép hoặc cấm ngắt. Bít GIE (INTCON<7>) điều khiển chung cho 14 ngắt khi bít này đặt thì các ngắt mới có tác dụng, khi bít GIE xoá thì tất cả các ngắt bị cấm.

Bít GIE bị xoá khi reset. Khi bít cờ ngắt thiết lập bít GIE thiết lập và bít PEIE thiết lập với ngắt ngoại vi đồng thời bít cho phép ngắt của ngắt đó cho phép thì ngắt đó xảy ra.

Khi một ngắt xảy ra bộ đếm chương trình PC được nạp giá trị 0004h và bít GIE bị xoá để cấm sự chồng ngắt, khi chỉ lệnh RETFIE thực hiện trả lại địa chỉ cho PC nơi xảy ra ngắt, đồng thời thiết lập lại bít GIE.

Khi xảy ra ngắt PC luôn được nạp giá trị 00004h vì các ngắt được phân biệt bởi bít cờ ngắt của ngắt đó.

Ngắt ngoài từ chân RB0/INT, và ngắt từ sự thay đổi trạng thái các chân RB4÷RB7 có thể đánh thức bộ xử lý từ chế độ SLEEP.

Các thanh ghi PIE1, PIR1, PIE2, PIR2 điều khiển các ngắt ngoại vi

Khi một ngắt xảy ra chỉ có PC được lưu trong stack do đó người sử dụng phải lưu các thanh ghi W, STATUS, PCLATH, khi xảy ra ngắt.

2.8. So sánh với Vi Điều Khiển 8051

Đặc điểm có thể thấy ngay đầu tiên mà Pic16F877A đem lại và nổi bật so với vi điều khiển 8051 là dòng Pic16F877A có những đặc tính kĩ thuật cao hơn hẳn so với bộ vi điều khiển 8051 thể hiện ở những điểm sau:

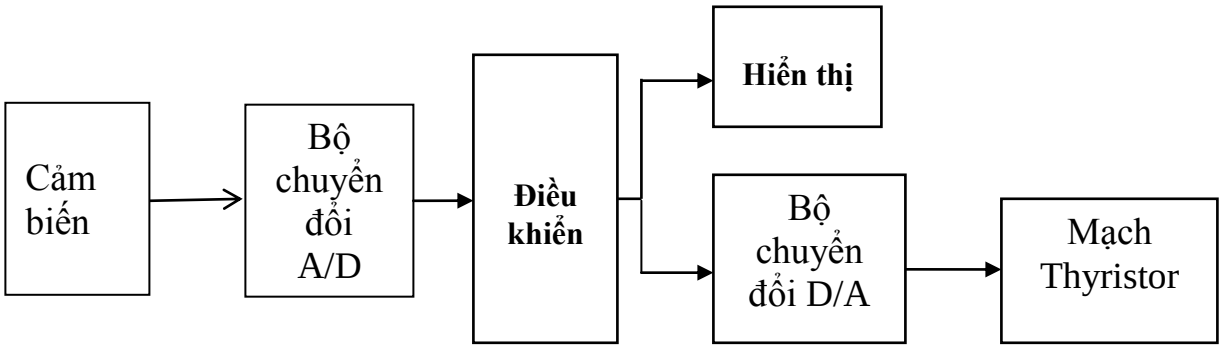
Vi điều khiển 8051		Pic 16F877A	
Đặc tính	số lượng	Đặc tính	số lượng
ROM trên chip	4K byte	ROM trên chip	4K byte
RAM	128 byte	RAM	368 byte
Bộ định thời	2	Bộ định thời	3
Các chân vào ra	32	Các chân vào ra	40
Cổng nối tiếp	1	Cổng nối tiếp	2
Nguồn ngắt	6		14

Ngoài những đặc điểm trên thì bộ vi điều khiển Pic16F877A còn có môt đặc điểm hơn hẳn so với 8051 là có 10 bít chuyển đổi A/D, điều này sẽ giúp chúng ta không phải sử dụng bộ chuyển đổi ngoài làm cho nối dây trở nên phức tạp.

Một đặc điểm nữa mà vi điều khiển pic16F877A có bộ dao động chủ trên chip điều này sẽ giúp tránh được những sai số không cần thiết trong việc tạo xung dao động, vi điều khiển Pic16F877A có khả năng tự Reset bằng bộ WDT, và có thêm 256 byte EEPROM. Nhưng giá thành của Pic đắt hơn so với 8051.

Chương 3.
THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN NHIỆT ĐỘ

3.1. Sơ đồ khối tổng quát.

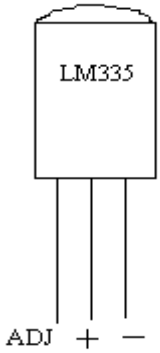


3.1. Sơ đồ khối hệ thống điều khiển nhiệt độ.

Hệ thống điều khiển nhận nhiệt độ từ đối tượng bằng cảm biến. Bộ cảm biến này sẽ chuyển nhiệt độ này thành mức điện áp tương ứng mức điện áp này ở dạng tín hiệu tương tự. Sau đó tín hiệu tương tự này sẽ được chuyển về dạng số bằng bộ chuyển đổi tương tự sang số trước khi đưa vào bộ điều khiển. Bộ điều khiển này nhận được nhiệt độ đo, kiểm tra nhiệt độ xem đã đạt hay chưa, nếu nhiệt độ chưa đủ thì điều khiển tăng hoặc ngược lại, nhiệt độ cao hơn thì điều khiển giảm. Quá trình điều khiển bằng điện áp xuất ra, qua bộ chuyển đổi DAC được điện áp tương tự, đưa đến mạch điều khiển thyristor. Mạch điều khiển thyristor sẽ tạo ra xung để mở thyristor phù hợp với yêu cầu tăng hoặc giảm nhệt độ.

3.2. Khối cảm biến.

Có rất nhiều loại cảm biến đo nhiệt độ trên thị trường nhưng dễ sử dụng và thông dụng nhất vẫn là LM335.



3.2. Cấu tạo chân

Các tham số của LM335:

- Có độ biến thiên theo nhiệt độ là:10mv/K^0.
- Có sự ổn định cao, chỉ sai số khoảng 1%.
- Hàm điện áp biến thiên tuyến tính khi đo trong khoảng -40–100^{0}C
- Tiêu tán công suất thấp.
- Dòng điện làm việc từ 4.10^{-4}A đến 5.10^{-3}A.
- Dòng ngược 15mA.
- Dòng thuận 10mA.
- Biến thiên của điện áp theo nhiệt độ: Vout=2.73+0.01xT^0C.

3.3. Khối chuyển đổi tương tự sang số.

Gọi tín hiệu tương tự là U_A, thì tín hiệu số là U_D được biểu diễn dưới dạng mã nhị phân như sau:

$$U_D=b_{n-1}.2^{n-1}+b_{n-2}.2^{n-2}+\ldots+b_0 2^0$$

Trong đó các hệ số b_k là các bit của số nhị phân. Bit b_{n-1} được gọi là bit có ý nghĩa lớn nhất (MSB), mỗi biến đổi giá trị của MSB tương ứng với sự biến đổi của tín hiệu là nửa dải làm việc. Bit b_0 gọi là bit có ý nghĩa nhỏ nhất (LSB), mỗi biến đổi giá trị của LSB tương ứng với một mức lương tử .

Với một mạch biến đổi có N bit đầu ra thì mỗi bước của bậc thang tương ứng một giá trị:

$$Q = U_{LSB} = \frac{U_{Am}}{2^N - 1}$$

Trong đó, U_{AM} là giá trị cực đại cho phép cho phép của điện áp đầu vào ADC. Q hoặc U_{LSB} gọi là mức lượng tử. Do tín hiệu số là rời rạc nên trong quá trình biến đổi AD xuất hiện một sai số gọi là sai số lượng tử hóa được xác định như sau:

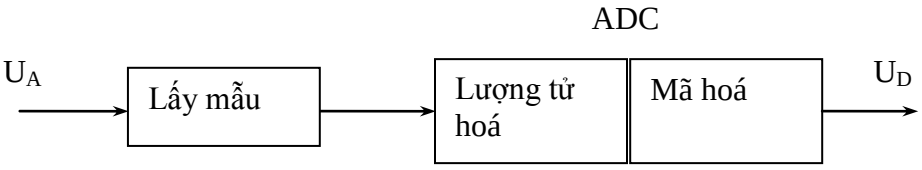
$$\Delta Q = \frac{1}{2} Q$$

Khi chuyển đổi AD phải thực hiện lấy mẫu tín hiệu tương tự. Để đảm bảo khôi phục lại tín hiệu một cách trung thực thì tần số lấy mẫu f_M phải thỏa mãn điều kiện:

$f_M \geq 2.f_{Amax}$

Trong đó, f_{Amax} là tần số cực đại của tín hiệu đầu vào

Quá trình biến đổi A/D gồm 3 bước: lấy mẫu, lượng tử hóa và mã hóa.



Hình 3.3. Sơ đồ khối quá trình chuyển đổi A/D.

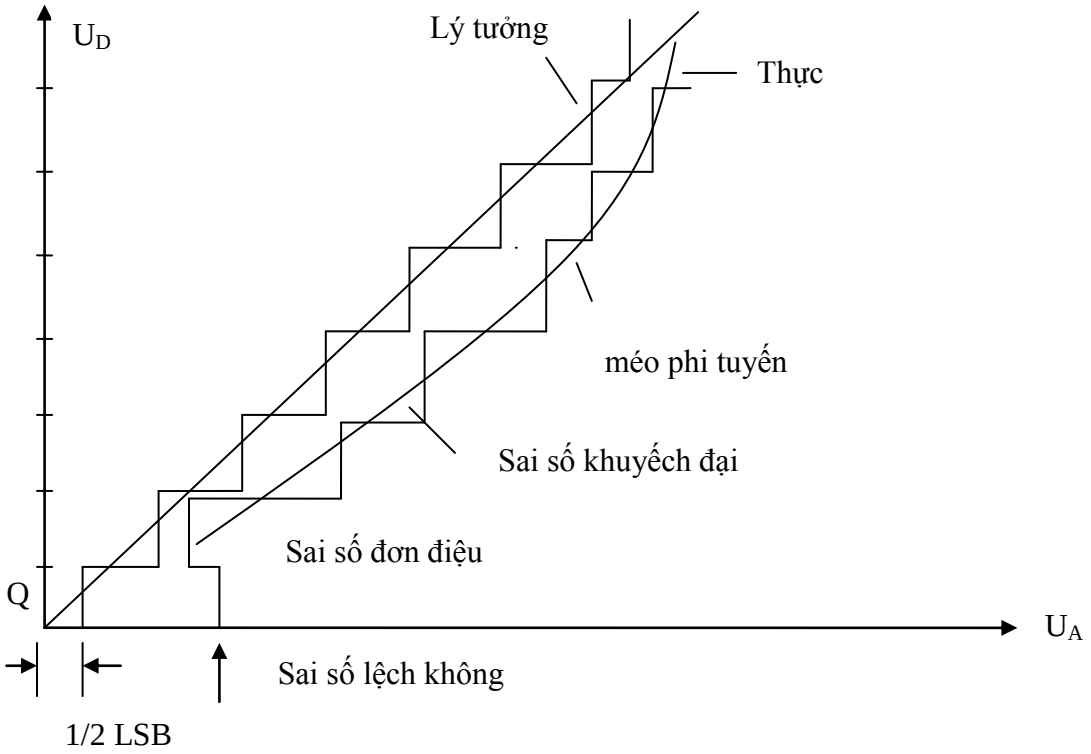
- Lấy mẫu: mạch lấy mẫu có hai nhiệm vụ:

+ Lấy mẫu tín hiệu tương tự tại các thời điểm khác nhau và cách đều nhau (rời rạc hóa về mặt thời gian).

+ Giữ cho biên độ điện áp tại các thời điểm lấy mẫu không đổi trong suốt quá trình chuyển đổi tiếp theo.

- Lượng tử hoá: là quá trình rời rạc các mẫu về biên độ. Chia khoảng biên độ thành các mức rời rạc gọi là các mức lượng tử, biên độ của các mẫu được làm tròn về các mức lượng tử đó.

- Mã hoá: Mã hoá các mẫu sau khi được lượng tử hoá thành các bit số.



Hình 4.4. Đặc tuyến truyền đạt lý tưởng và thực của bộ chuyển đổi **A/D.**

Tổng quát ta có công thức chuyển đổi A/D đối với mỗi mẫu tín hiệu tương tự:

$$U_{Di} = Round\left[\frac{U_{Ai}}{U_{ref}}.2^{N}\right]$$

U_{Ai}: Điện áp tương tự của mẫu thứ i.

U_{ref}: Điện áp tham chiếu (điện áp chuẩn cố định), dùng để so sánh với U_{Ai} tạo điện áp số.

U_{Di}: Điện áp số ứng với mẫu U_{Ai}.

N: Số bit của bộ chuyển đổi

ở đây yêu cầu $U_A \leq U_{ref}$, nên ta phải lựa chọn U_{ref} thích hợp với mỗi tín hiệu U_A.

Với mỗi giá trị N, thì U_{ref} càng lớn thì sai số lượng tử càng lớn. Với mỗi giá trị U_{ref} thì N càng lớn thì sai số lượng tử càng nhỏ.

Các tham số chính của bộ chuyển đổi A/D

- Dải biến đổi của điện áp tín hiệu tương tự ở đầu vào: là khoảng điện áp mà bộ chuyển đổi A/D có thể thực hiện chuyển đổi được. Khoảng điện áp đó có thể lấy trị số từ 0 đến một trị số dương hay âm nào đó hoặc có thể là điện áp hai cực tính.

- Độ chính xác của bộ chuyển đổi A/D: Tham số đầu tiên đặc trưng cho độ chính xác của một ADC là độ phân biệt. Trên đầu ra mỗi bộ ADC là các giá trị số được sắp xếp theo quy luật của một loại mã nào đó. Số các số hạng của mã số ở đầu ra (số bit trong mã nhị phân) tương ứng với dải biến đổi của điện áp vào cho biết mức chính xác của phép chuyển đổi. Một ADC có N bit đầu ra thì nó có thể phân biệt được 2^N mức trong dải biến đổi của nó. Độ phân biệt của một ADC là Q, nó chính là giá trị của một mức lượng tử hóa hoặc còn gọi là 1 LSB. Trong thực tế thường dùng số bit N ở đầu ra để đặc trưng cho độ chính xác với cùng một dải điện áp vào số các số hạng của mã số ở đầu ra càng lớn thì độ chính xác càng cao.

Ngoài ra đặc trưng cho tính chính xác của ADC còn có các tham số khác, đó là :

+ Đường đặc tuyến có sai số lệch không, nghĩa là nó không xuất phát tại giá trị tương ứng là 1/2 LSB. Nó là hình bậc thang không đều do ảnh hưởng của các sai số.

+ Sai số khuyếch đại là sai số giữa độ dốc trung bình của đường đặc tuyến thực với độ dốc trung bình của đường đặc tuyến lý tưởng.

+ Sai số phi tuyến được đặc trưng bởi sự thay đổi độ dốc đường trung bình của đặc tuyến thực trong dải biến đổi của của điện áp vào. Sai số này làm cho đặc tuyến chuyển đổi có dạng hình bậc thang không đều.

+ Sai số đơn điệu thực chất cũng do tính phi tuyến của đường đặc tính biến đổi gây ra, nhưng nó làm cho độ dốc đường trung bình biến thiên không đơn điệu, thậm chí mất một vài mã số.

- Tốc độ chuyển đổi: cho biết số mẫu chuyển đổi trong 1 giây, được gọi là tần số chuyển đổi f_c. Cũng có thể dùng tham số thời gian chuyển đổi T_c để đặc trưng cho tốc độ chuyển đổi. Vì giữa các lần chuyển đổi còn có một khoảng thời gian cần thiết để cho ADC phục hồi lại trạng thái ban đầu, nên thường $f_c<1/T_c$, ở đây với một bộ ADC tốc độ cao thì phải trả giá bằng độ chính xác giảm hoặc ngược lại.

Các phương pháp biến đổi số tương tự.

Có nhiều cách phân loại các phương pháp biến đổi số tương tự. Trong đó có cách phân loại theo quá trình chuyển đổi về mặt thời gian, theo cách phân loại này có bốn phương pháp biến đổi A/D:

- Biến đổi song song: trong phương pháp này, tín hiệu được so sánh cùng một lúc với nhiều giá trị chuẩn. Do đó, tất cả các bit được xác định đồng thời và đưa đến đầu ra.

- Biến đổi nối tiếp theo mã đếm: quá trình so sánh được thực hiện lần lượt từng bước theo quy luật của mã đếm. Kết quả chuyển đổi được xác định bằng cách đếm số lượng giá trị chuẩn có thể chứa được trong giá trị tín hiệu tương tự cần chuyển đổi.

- Biến đổi nối tiếp theo mã nhị phân: quá trình so sánh được thực hiện lần lượt từng bước theo qui luật của mã nhị phân. Các đơn vị chuẩn dùng để so sánh lấy các giá trị giảm dần theo quy luật của mã nhi phân. Do đó các bit được xác định lần lượt theo từ bit có ý nghĩa lớn nhất đến bit có ý nghĩa nhỏ nhất.

- Biến đổi song song - nối tiếp kết hợp: trong phương pháp này, qua mỗi bước so sánh có thể xác định được tối thiểu là hai bit đồng thời.

Sử dụng bộ chuyển đổi trong hệ thống.

Trong hệ thống ở đây ta sử dụng bộ chuyển đổi A/D 10 bit tích hợp trong bộ vi điều khiển Pic 16F877A.

3.4. Khối điều khiển.

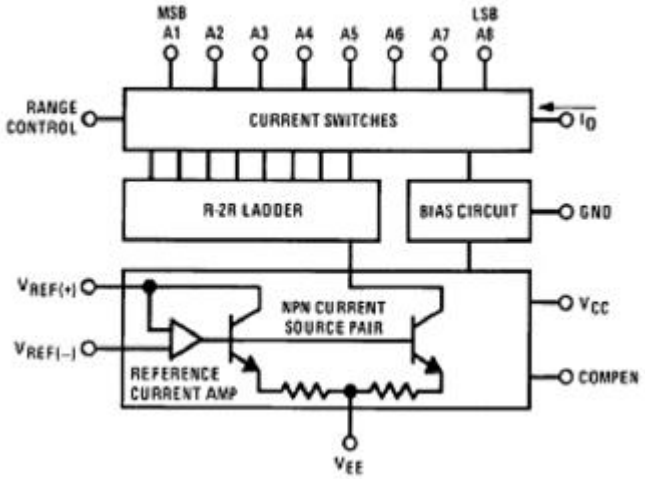
Hiện nay vi điều khiển phát triển rất mạnh điển hình trong số đó là pic có rất nhiều tính năng ưu việt hơn so với vi điều kiển khác đặc biệt nhất là tính thông dụng của nó. Vì những lý do đó mà em sử dụng pic16F877A cho khối điều khiển trong đề tài này.

3.5. Khối chuyển đổi số sang tương tự.

Với yêu cầu ở đây đường điều khiển đưa ra là 8 bits, và điều khiển nhiệt độ không yêu tốc độ đáp ứng nhanh nên để đơn giản ở đây em chọn IC DAC sẵn có là DAC0808. DAC0808 là một bộ chuyển đổi 8 bits số sang tương tự đầu ra có đặc tính thời gian đúng bằng kích thước của tín hiệu vào trong khoảng 150ns với công suất tiêu thụ là 33mW khi điện áp cung cấp là ± 5V. Không cần phải điều chỉnh dòng điện IREF cho tất cả các ứng dụng, từ đó đầu ra hiện tại là ±1LBS của 255 (IREF/256). Nguồn cung cấp của DAC0808 độc lập với “bit code” và đưa ra những đặc điểm nổi bật của thiết bị phụ thộc vào mức điện áp vào.

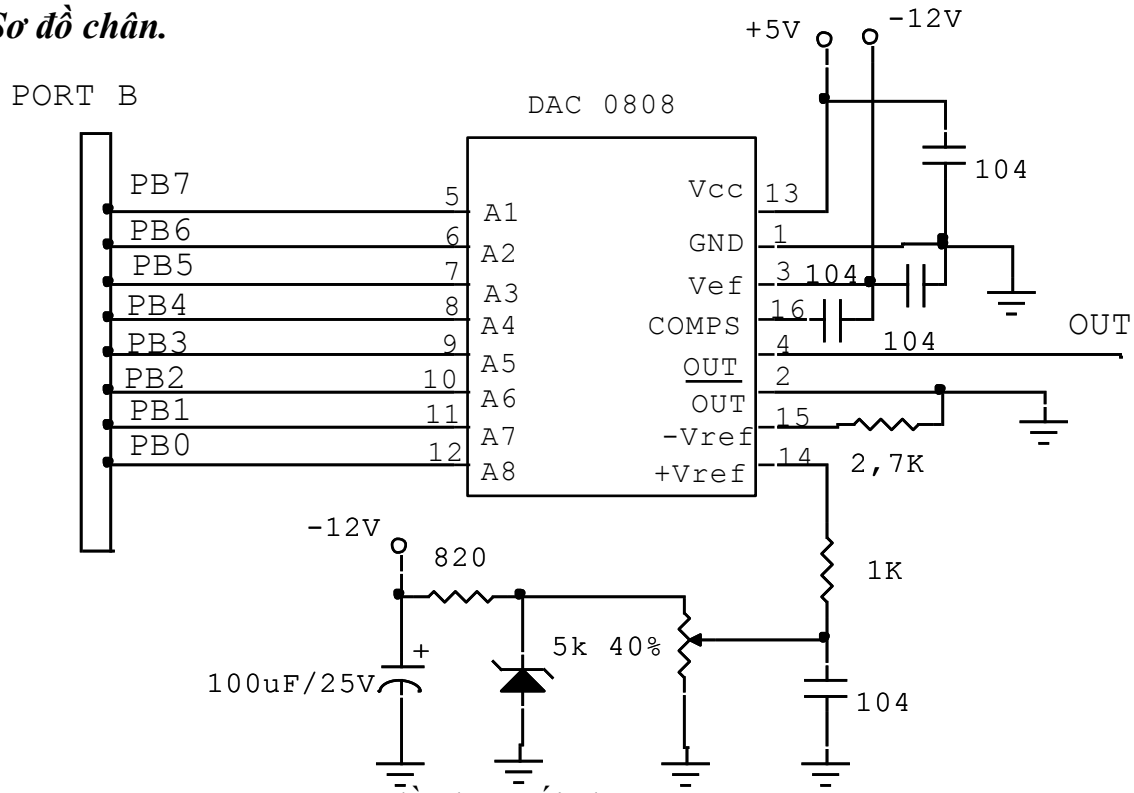
DAC0808 giao tiếp trực tiếp với TTL, DTL hay CMOS ở mức logic, và dùng thay thế cho MC1580/MC1408.

Cấu tạo bên trong DAC



4.5. Sơ đồ cấu tạo bên trong DAC0808.

Sơ đồ chân.



4.6. Sơ đồ ghép nối chân của DAC0808.

3.6. Khối điều khiển thyristor.

Thyristor chỉ mở cho dòng điện chạy qua khi có điện áp dương đặt lên anôt và xung điện áp dương đặt lên cực điều khiển. Sau khi Thyristor đã mở thì xung điều khiển không còn tác dụng, dòng điện chảy qua Thyristor do thông số của mạch động lực quyết định.

3.6.1. Sơ đồ cấu trúc

Sơ đồ khối mạch điều khiển Thyristor như hình 4.7.

Mạch điều khiển có các chức năng sau:

- Điều chỉnh được vị trí xung điều khiển trong phạm vi nửa chu kỳ dương của điện áp trên anôt- catôt của Thyristor.

- Tạo ra được các xung có đủ điều kiện mở được Thyristor. Xung điều khiển thường có biên độ từ 2 đến 10V, độ rộng xung tx= 20-100µs đối với thiết bị chỉnh lưu hoặc cặp Thyristor đấu song song ngược.

Độ rộng xung được xác định theo biểu thức:

$$t_x = \frac{I_{dt}}{d_i / d_t}$$

Trong đó:

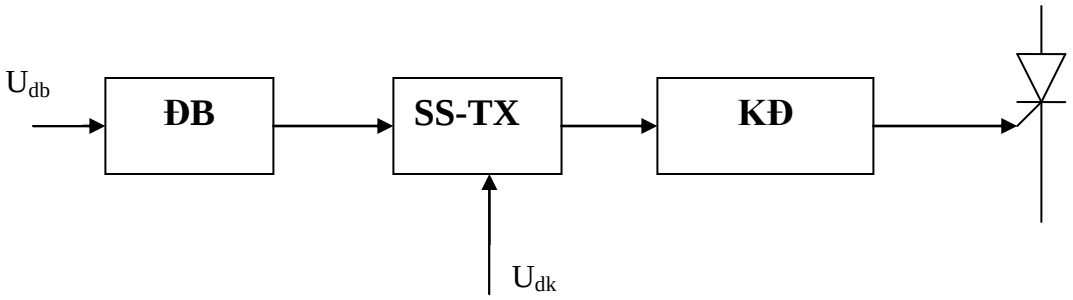
- I_{dt} là dòng duy trì của Thyristor;
- di/dt là tốc độ tăng trưởng của dòng tải.

Cấu trúc của một mạch điều khiển Thyristor gồm 3 khâu chính sau đây:

- Khâu đồng bộ (ĐB): tạo tín hiệu đồng bộ với điện áp anôt-catôt của Thyristor cần mở. Tín hiệu này là điện áp xoay chiều, thường lấy từ biến áp có sơ cấp nối song song với Thyristor cần mở.

- Khâu so sánh - tạo xung (SS-TX): làm nhiệm vụ so sánh giữa điện áp đồng bộ thường đã được biến thể với tín hiệu điều khiển một chiều để tạo ra xung kích mở Thyristor.

- Khâu khuếch đại xung (KĐ): tạo ra xung mở có đủ điều kiện để mở Thyristor.



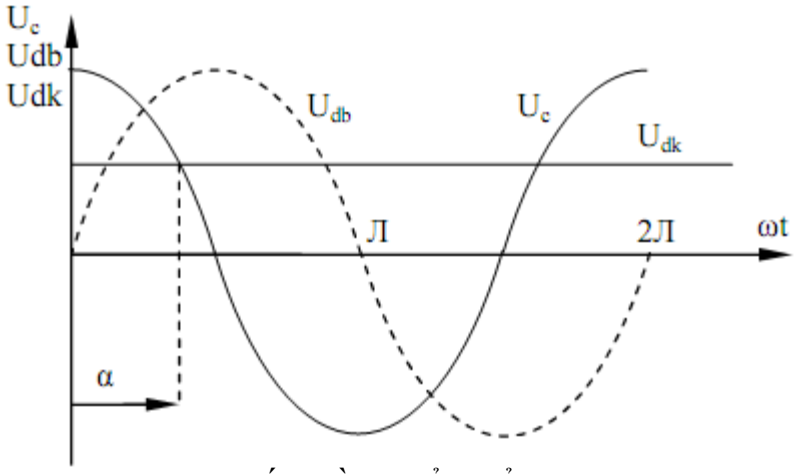
4.7. Sơ đồ khối mạch điều khiển thyristor.

Khi thay đổi giá trị điện áp một chiều Uđk thì góc mở α sẽ thay đổi.

3.6.2. Nguyên tắc điều khiển

Sử dụng nguyên tắc điều khiển thẳng đứng "arccos" như hình 2 để thực hiện điều chỉnh vị trí đặt xung trong nửa chu kỳ dương của điện áp đặt trên Thyristor. Theo nguyên tắc này, ở khâu so sánh có hai điện áp đặt vào:

- Điện áp đồng bộ sin, sau khi ra khỏi khâu ĐB được tạo thành tín hiệu cos
- Điện áp điều khiển là áp một chiều có thể biến đổi được



4.8. Nguyên tắc điều khiển thẳng đứng "arccos" .

Điện áp uđb= Um sinωt thì: Uc = Um cosωt

Giá trị α được tính theo phương trình sau: Umcosα = Uđk

Do đó: α = arccos(Udk/Um)

- khi Udk = Um thì α = 0

- khi Udk = 0 thì α =Л/2

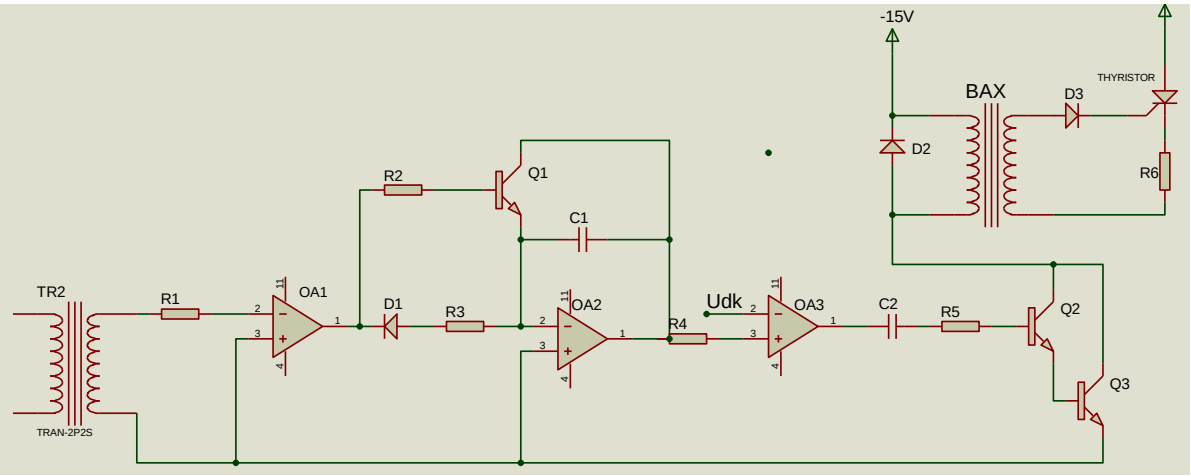
- khi Udk = -Um thì α = Л

Như vậy, khi điều chỉnh Udk từ trị -Um đến +Um, ta có thể điều chỉnh được góc α từ 0 đến Л.

3.6.3. Sơ đồ nguyên lý

Trong phần này trình bày một sơ đồ điều khiển Thyristor một kênh như hình 4.10 đã được thiết kế và lắp ráp thực tế. Sơ đồ làm việc theo nguyên tắc điều khiển thẳng đứng tuyến tính trong đó khâu tạo xung tam giac và khâu so sánh sử dụng OA loại TL084 so sánh theo kiểu hai tín hiệu cùng dấu.

Khâu khuếch đại xung sử dụng một Transistor và biến áp xung. Khâu khuếch đại có thể tính chọn khác nhau tuỳ thuộc vào Thyristor được chọn.



Hình 4.10. Sơ đồ hệ thống điều khiển Thyristor.

Sơ đồ gồm kênh kích mở cho Thyristor. Kênh gồm có 4 khâu: khâu tạo xung vuông, khâu tạo xung tam giác khâu so sánh, khâu khuếch đại, khâu tạo xung kim.

- Khâu tạo vuông gồm: OP1, R1. Đầu vào của khâu là tín hiệu hình sin qua khâu này tín hiệu xung sin bị cắt thành xung vuông.

- Khâu tạo xung tam giác bao gồm: OA2, R2, R3, C1, D1, Q1. Khâu tạo xung tam giác thực chất là mạch tích phân dùng để biến đổi xung vuông từ đầu ra của khối tạo xung vuông để chuyển thành xung tam giác.

- Khâu so sánh gồm R4 và OP3. Tín hiệu ở đầu ra của mạch tích phân sẽ được so sánh với điện áp điều khiển. Đầu ra của khâu so sánh tạo ra xung vuông được điều khiển bằng Udk.

- Khâu khuếch đại xung gồm: R5, Q2, Q3. Dùng để khuếch đại tín hiệu xung vuông ở đầu ra của bộ so sánh để được điện áp theo yêu cầu.

- Khâu tạo xung kim gồm: D2, BAX. Khối này tạo ra các xung kim từ các sườn của xung vuông ở đầu ra của bộ khuếch đại để đưa vào điều kiển mở thyristor theo yêu cầu.

3.7. Khối hiển thị LCD.

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác như nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ ... Trong đề tài này tôi sử dụng HD44780 của Hitachi, một loại thiết bị hiển thị LCD rất thông dụng ở nước ta.

3.7.1. Các chân chức năng.

Bảng 4.1. Các chân chức năng của HD44780.

Chân số	Tên	Chức năng
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển.
2	Vdd	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với 5V của mạch điều khiển.
3	Vo	Chân này dùng để điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic "0" (GND) hoặc logic "1" (Vcc) để chọn thanh ghi. + Logic "0": Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ "ghi" - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ "đọc" - read) + Logic "1": Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	RW	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic "0" để LCD hoạt động ở chế độ ghi, hoặc nối với logic "1" để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (low-to-high transition) của tín hiệu chân E.

		+ Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện sườn lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7÷14	DB0÷DB7	8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.
15	A	15 là Catot, điện áp khoảng Uak=4,2V
16	K	Chân nối đất của đèn Back light

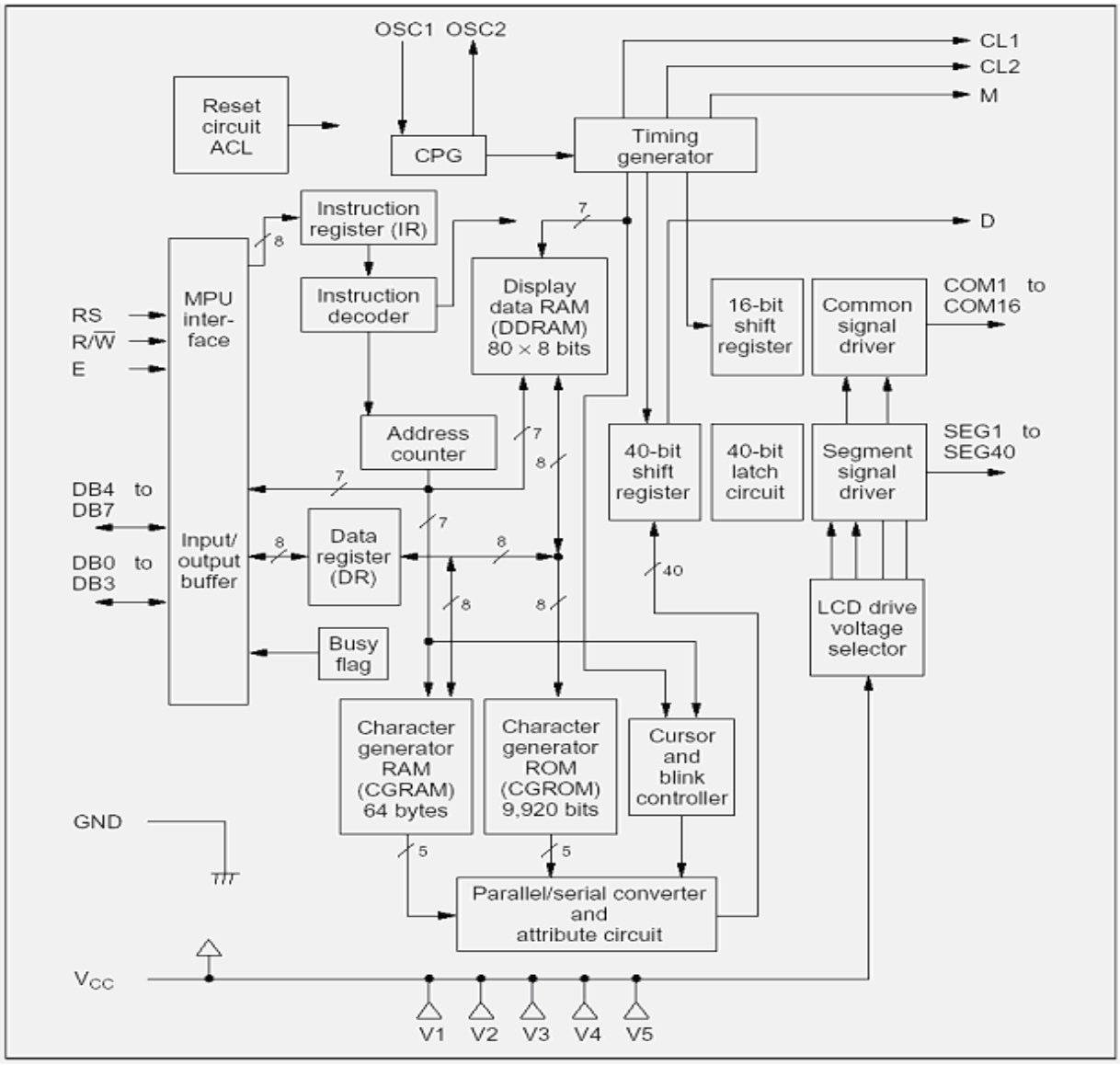
3.7.2. Sơ đồ khối của HD44780.

Để hiểu rõ hơn chức năng các chân và hoạt động của chúng, ta tìm hiểu sơ qua chíp HD44780 thông qua các khối cơ bản của nó.

*) Các thanh ghi:

Chíp HD44780 có 2 thanh ghi 8 bit quan trọng là: Thanh ghi lệnh IR (Instructor Register) và thanh ghi dữ liệu DR (Data Register).

- Thanh ghi IR: Để điều khiển LCD, người dùng phải "ra lệnh" thông qua tám đường bus DB0-DB7. Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR. Nghĩa là, khi ta nạp vào thanh ghi IR một chuỗi 8 bit, chíp HD44780 sẽ tra bảng mã lệnh tại địa chỉ mà IR cung cấp và thực hiện lệnh đó.



Hình 4.11 Sơ đồ khối của HD44780.

- Thanh ghi DR: Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM, DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gởi ra cho MPU (ở chế độ đọc). Nghĩa là, khi MPU ghi thông tin vào DR, mạch nội bên trong chíp sẽ tự động ghi thông tin này vào DDRAM hoặc CGRAM. Hoặc khi thông tin về địa chỉ được ghi vào IR, dữ liệu ở địa chỉ này trong vùng RAM nội của HD44780 sẽ được chuyển ra DR để truyền cho MPU. Vậy bằng cách điều khiển chân RS và R/W chúng ta có thể

chuyển qua lại giữ 2 thanh ghi này trong khi giao tiếp với MPU. Bảng 3.2. tóm tắt lại các thiết lập đối với hai chân RS và R/W theo mục đích giao tiếp.

Bảng 4.2. Bảng chức năng chân RS và R/W theo mục đích sử dụng.

RS	RW	Ý nghĩa
0	0	Ghi vào thanh ghi IR để ra lệnh cho LCD (VD: cần display clear, …)
0	1	Đọc cờ bận ở DB7 và giá trị của bộ đếm địa chỉ ở DB0-DB6
1	0	Ghi vào thanh ghi DR
1	1	Đọc dữ liệu từ DR

*) Cờ báo bận BF (Busy Flag):

Khi thực hiện các hoạt động bên trong chíp, mạch nội bên trong cần một khoảng thời gian để hoàn tất. Khi đang thực thi các hoạt động bên trong chíp như thế, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF (thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để báo cho MPU biết nó đang "bận". Dĩ nhiên, khi xong việc, nó sẽ đặt cờ BF lại mức 0.

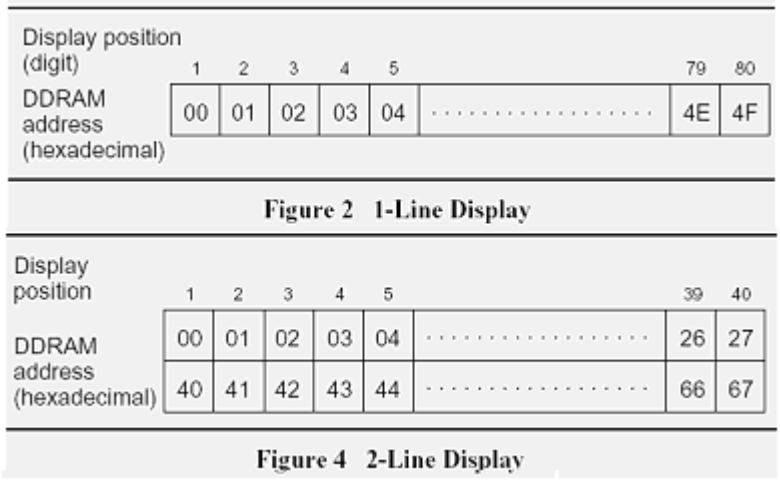
*) Bộ đếm địa chỉ AC (Address Counter):

Như trong sơ đồ khối, thanh ghi IR không trực tiếp kết nối với vùng RAM (DDRAM và CGRAM) mà thông qua bộ đếm địa chỉ AC. Bộ đếm này lại nối với 2 vùng RAM theo kiểu rẽ nhánh. Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM nhưng việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh. Sau khi ghi vào (hoặc đọc từ) RAM, bộ đếm AC tự động tăng lên (hoặc giảm đi) 1 đơn vị và

nội dung của AC được xuất ra cho MPU thông qua DB0-DB6 khi có thiết lập RS=0 và R/W=1 (xem bảng 3.2). Lưu ý: Thời gian cập nhật AC không được tính vào thời gian thực thi lệnh mà được cập nhật sau khi cờ BF lên mức cao (not busy), cho nên khi lập trình hiển thị, bạn phải delay một khoảng tADD khoảng 4uS-5uS (ngay sau khi BF=1) trước khi nạp dữ liệu mới.

*) Vùng RAM hiển thị DDRAM (Display Data RAM):

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình và khi bạn ghi vào vùng RAM này một mã 8 bit, LCD sẽ hiển thị tại vị trí tương ứng trên màn hình một kí tự có mã 8 bit mà bạn đã cung cấp như hình 3.3.



Hình 4.12 Mối liên hệ giữa địa chỉ của DDRAM và vị trí hiển thị của LCD.

Vùng RAM này có 80x8 bit nhớ, nghĩa là chứa được 80 kí tự mã 8 bit. Những vùng RAM còn lại không dùng cho hiển thị có thể dùng như vùng RAM đa mục đích. Lưu ý là để truy cập vào DDRAM, ta phải cung cấp địa chỉ cho AC theo mã HEX.

*) Vùng ROM chứa kí tự CGROM (Character Generator ROM):

Vùng ROM này dùng để chứa các mẫu kí tự loại 5x8 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit. Tuy nhiên, nó chỉ có 208 mẫu kí tự 5x8 và 32 mẫu kí tự kiểu 5x10 (tổng cộng là 240 thay vì 256 mẫu kí tự). Người dùng không thể thay đổi vùng ROM này.

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)

EPROM Address												Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0	
								0	0	0	0	1	0	0	0	0	
								0	0	0	1	1	0	0	0	0	
								0	0	1	0	1	0	1	1	0	
								0	0	1	1	1	1	0	0	1	
								0	1	0	0	1	0	0	0	1	
								0	1	0	1	1	0	0	0	1	
								0	1	1	0	1	1	1	1	0	
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	← Cursor position
								1	0	0	0	0	0	0	0	0	
								1	0	0	1	0	0	0	0	0	
								1	0	1	0	0	0	0	0	0	
								1	0	1	1	0	0	0	0	0	
								1	1	0	0	0	0	0	0	0	
								1	1	0	1	0	0	0	0	0	
								1	1	1	0	0	0	0	0	0	
								1	1	1	1	0	0	0	0	0	

Hình 4.13. Mối liên hệ giữa địa chỉ của ROM và dữ liệu tạo mẫu kí tự.

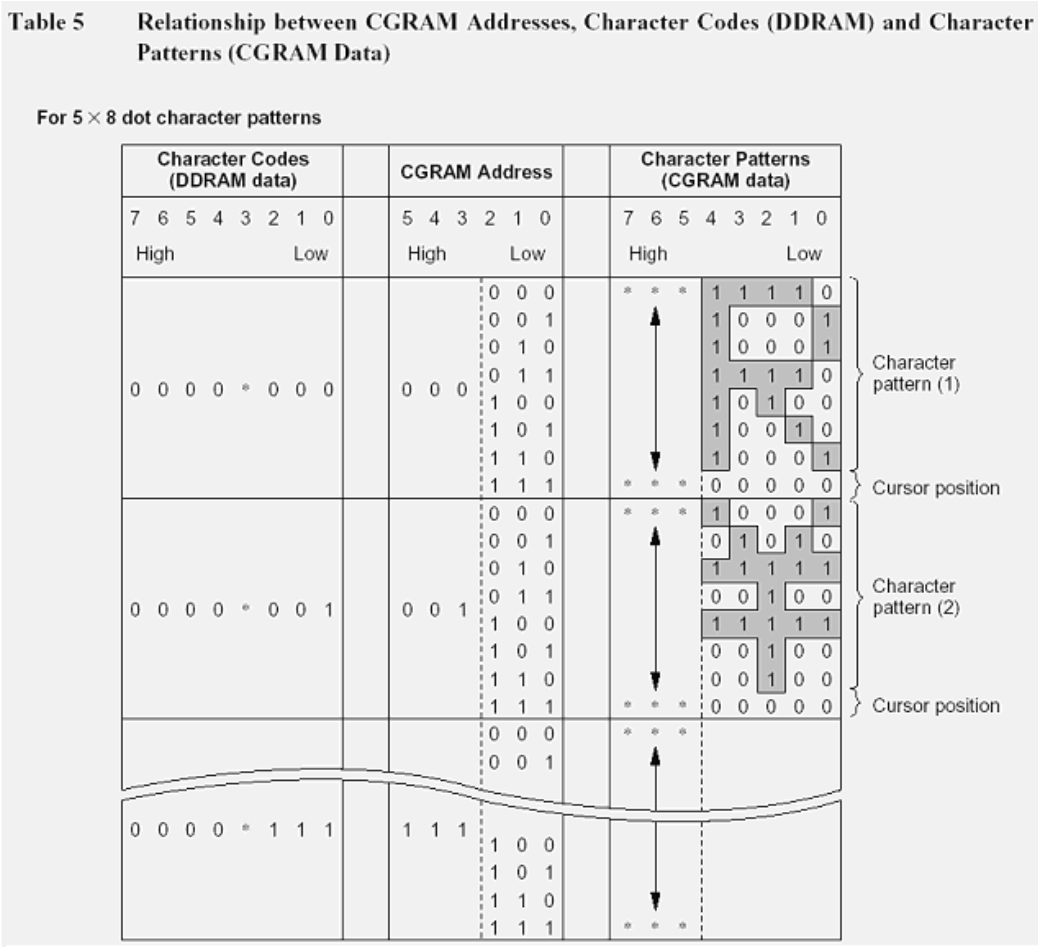
*) Vùng RAM chứa kí tự đồ họa CGRAM (Character Generator RAM):

Như trên bảng mã kí tự, nhà sản xuất dành vùng có địa chỉ byte cao là 0000 để người dùng có thể tạo các mẫu kí tự đồ họa riêng. Tuy nhiên dung lượng vùng này rất hạn chế: Ta chỉ có thể tạo 8 kí tự loại 5x8 điểm ảnh, hoặc 4 kí tự loại 5x10 điểm ảnh. Để ghi vào CGRAM, xem hình 3.6.

3.7.3. Tập lệnh của LCD.

Trước khi tìm hiểu tập lệnh của LCD, sau đây là một vài chú ý khi giao tiếp với LCD:

* Tuy trong sơ đồ khối của LCD có nhiều khối khác nhau, nhưng khi lập trình điều khiển LCD ta chỉ có thể tác động trực tiếp được vào 2 thanh ghi DR và IR thông qua các chân DBx, và ta phải thiết lập chân RS, R/W phù hợp để chuyển qua lại giữ 2 thanh ghi này. (xem bảng 3.2)



Hình 4.14. Mối liên hệ giữa địa chỉ của CGRAM, dữ liệu CGARM, và mã kí tự.

* Với mỗi lệnh, LCD cần một khoảng thời gian để hoàn tất, thời gian này có thể khá lâu đối với tốc độ của MPU, nên ta cần kiểm tra cờ BF hoặc đợi (delay) cho LCD thực thi xong lệnh hiện hành mới có thể ra lệnh tiếp theo.

* Địa chỉ của RAM (AC) sẽ tự động tăng (giảm) 1 đơn vị, mỗi khi có lệnh ghi vào RAM. (Điều này giúp chương trình gọn hơn)

* Các lệnh của LCD có thể chia thành 4 nhóm như sau:

• Các lệnh về kiểu hiển thị. VD : Kiểu hiển thị (1 hàng / 2 hàng), chiều dài dữ liệu (8 bit / 4 bit), …

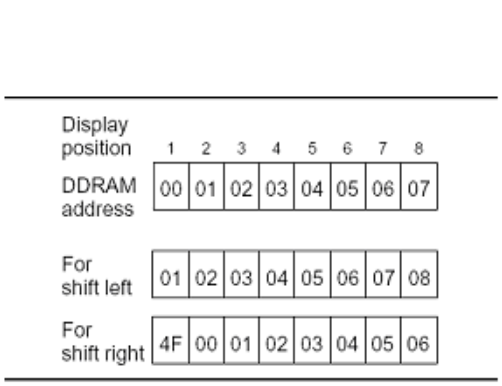
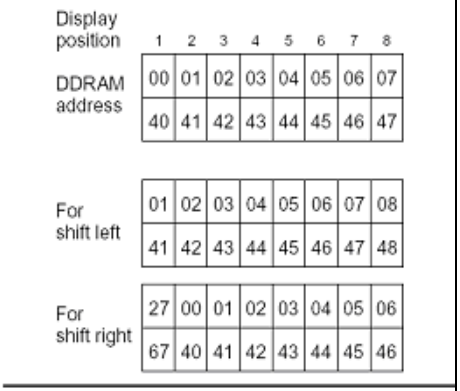
• Chỉ định địa chỉ RAM nội.

• Nhóm lệnh truyền dữ liệu trong RAM nội.

• Các lệnh còn lại .

Bảng 4.3. Tập lệnh của LCD.

Tên lệnh	Hoạt động	Thời gian chạy
Clear Display	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 0 0 0 0 0 1 Lệnh Clear Display (xóa hiển thị) sẽ ghi một khoảng trống (mã hiển thị kí tự 20H) vào tất cả ô nhớ trong DDRAM, sau đó trả bộ đếm địa chỉ AC=0, trả lại hiển thị gốc nếu nó bị thay đổi, nghĩa là: Tắt hiển thị, con trỏ dời về góc trái (hàng đầu tiên), chế độ tăng AC.	
Return home	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 0 0 0 0 1 * Lệnh Return home trả bộ đếm địa chỉ AC về 0, trả lại kiểu hiển thị gốc nếu nó bị thay đổi. Nội dung của DDRAM không thay đổi.	1.52 ms
Entry mode set	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 0 0 0 1 [I/D] [S] I/D: Tăng (I/D=1) hoặc giảm (I/D=0) bộ đếm địa chỉ hiển thị AC 1 đơn vị mỗi khi có hành động ghi hoặc đọc vùng DDRAM. Vị trí con trỏ cũng di chuyển theo sự tăng giảm này. S: Khi S=1 toàn bộ nội dung hiển thị bị dịch sang phải (I/D=0) hoặc	

	sang trái (I/D=1) mỗi khi có hành động ghi vùng DDRAM. Khi S=0: không dịch nội dung hiển thị. Nội dung hiển thị không dịch khi đọc DDRAM hoặc đọc/ghi vùng CGRAM.  Figure 3 1-Line by 8-Character Display Example  Figure 5 2-Line by 8-Character Display Example Hình 3.7. Hoạt động dịch trái và dịch phải nội dung hiển thị	37µs
Display on/off control	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 0 0 1 [D] [C] [B] D: Hiển thị màn hình khi D=1 và ngược lại. Khi tắt hiển thị, nội dung DDRAM không thay đổi. C: Hiển thị con trỏ khi C=1 và ngược lại. Vị trí và hình dạng con trỏ, xem hình 3.8. B: Nhấp nháy kí tự tại vị trí con trỏ khi B=1 và ngược lại. Xem thêm hình 8. về kiểu nhấp nháy. Chu kì nhấp nháy khoảng 409,6ms khi mạch dao động nội LCD là 250kHz.	37µs

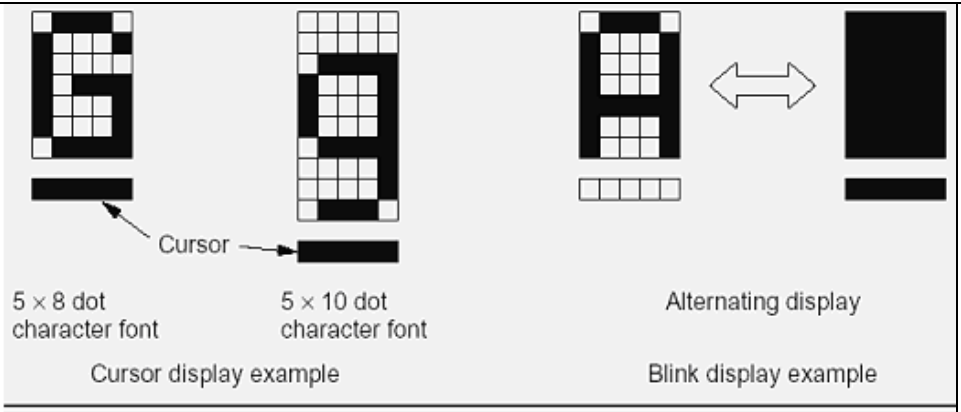


Figure 12 Cursor and Blinking

Hình 3.8. Kiểu con, kiểu kí tự và nhấp nháy kí tự

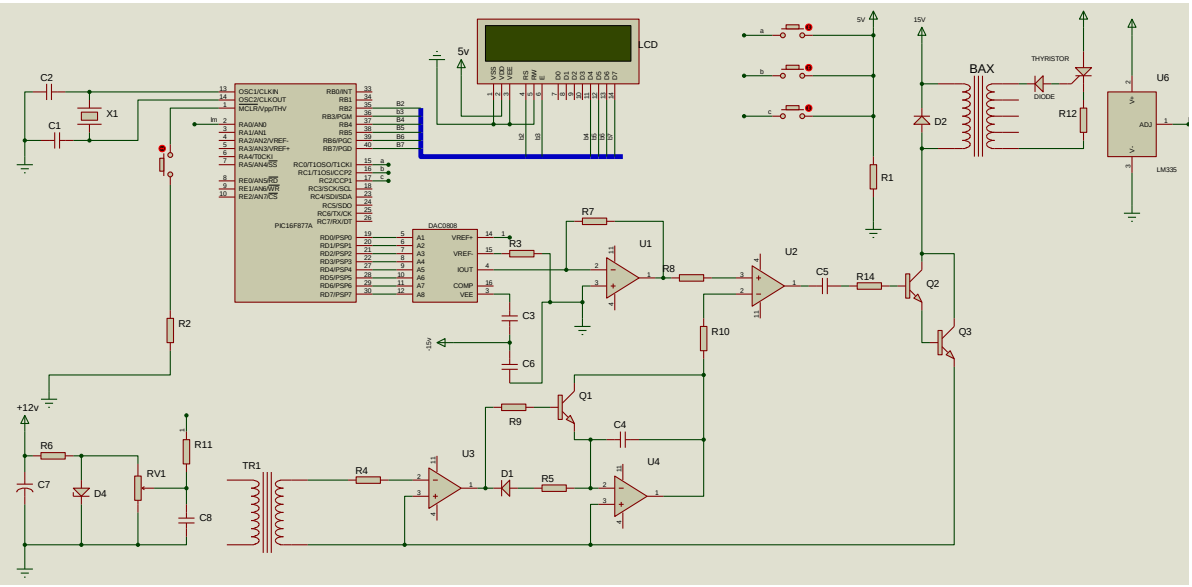
Cursor or display shift	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 0 1 [S/C] [R/L] * * Lệnh Cursor or display shift dịch chuyển con trỏ hay dữ liệu hiển thị sang trái mà không cần hành động ghi/đọc dữ liệu. Khi hiển thị kiểu 2 dòng, con trỏ sẽ nhảy xuống dòng dưới khi dịch qua vị trí thứ 40 của hàng đầu tiên. Dữ liệu hàng đầu và hàng 2 dịch cùng một lúc. Chi tiết sử dụng xem bảng sau:	37µs
Function set	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 0 0 1 [DL] [N] [F] * * DL: Khi DL=1, LCD giao tiếp với MPU bằng giao thức 8 bit (từ bit DB7 đến DB0). Ngược lại, giao thức giao tiếp là 4 bit (từ bit DB7 đến bit DB0). Khi chọn giao thức 4 bit, dữ liệu được truyền/nhận 2 lần liên tiếp với 4 bit cao gửi/nhận trước, 4 bit thấp gởi/nhận sau. N: Thiết lập số hàng hiển thị. Khi N=0: hiển thị 1 hàng, N=1: hiển	

S/C	R/L	Hoạt động
0	0	Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).
0	1	Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).
1	0	Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.
1	1	Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.

	thị 2 hàng. F: Thiết lập kiểu kí tự. Khi F=0: kiểu kí tự 5x8 điểm ảnh, F=1: kiểu kí tự 5x10 điểm ảnh. * Chú ý: • Chỉ thực hiện thay đổi Function set ở đầu chương trình. Và sau khi được thực thi 1 lần, lệnh thay đổi Function set không được LCD chấp nhận nữa ngoại trừ thiết lập chuyển đổi giao thức giao tiếp. • Không thể hiển thị kiểu kí tự 5x10 điểm ảnh ở kiểu hiển thị 2 hàng.	37µs
Set CGRAM address	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 0 1 [ACG][ACG][ACG][ACG][ACG][ACG] Lệnh này ghi vào AC địa chỉ của CGRAM. Kí hiệu [ACG] chỉ 1 bit của chuỗi dữ liệu 6 bit. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ CGRAM tại địa chỉ đã được chỉ định.	37µs
Set DDRAM address	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = 1 [AD] [AD] [AD] [AD] [AD] [AD] [AD] Lệnh này ghi vào AC địa chỉ của DDRAM, dùng khi cần thiết lập tọa độ hiển thị mong muốn. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ DDRAM tại địa chỉ đã được chỉ định. Khi ở chế độ hiển thị 1 hàng, địa chỉ có thể từ 00H đến 4FH. Khi ở chế độ hiển thị 2 hàng, địa chỉ từ 00h đến 27H cho hàng thứ nhất, và từ 40h đến 67h cho hàng thứ 2.	37µs
Read BF and address	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= [BF] [AC] [AC] [AC] [AC] [AC] [AC] [AC] (RS=0, R/W=1) Như đã đề cập trước đây, khi cờ BF bật, LCD đang làm việc và lệnh tiếp theo (nếu có) sẽ bị bỏ qua nếu cờ BF chưa về mức thấp. Cho nên, khi lập trình điều khiển, bạn phải kiểm tra cờ BF trước khi ghi dữ liệu vào LCD. Khi đọc cờ BF, giá trị của AC cũng được xuất ra các bit [AC]. Nó là địa chỉ của CG hay DDRAM là tùy thuộc vào lệnh trước đó.	0µs

Write ata to CG or DDRAM	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = [Write data] (RS=1, R/W=0) Khi thiết lập RS=1, R/W=0, dữ liệu cần ghi được đưa vào các chân DBx từ mạch ngoài sẽ được LCD chuyển vào trong LCD tại địa chỉ được xác định từ lệnh ghi địa chỉ trước đó (lệnh ghi địa chỉ cũng xác định luôn vùng RAM cần ghi). Sau khi ghi, bộ đếm địa chỉ AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode. Lưu ý là thời gian cập nhật AC không tính vào thời gian thực thi lệnh.	37µs tADD 4µs
Read data from CG or DDRAM	Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx = [Read data] (RS=1, R/W=1) Khi thiết lập RS=1, R/W=1,dữ liệu từ CG/DDRAM được chuyển ra MPU thông qua các chân DBx (địa chỉ và vùng RAM đã được xác định bằng lệnh ghi địa chỉ trước đó). Sau khi đọc, AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode, tuy nhiên nội dung hiển thị không bị dịch bất chấp chế độ Entry mode.	37µs tADD 4µs

3.8. Sơ đồ mạch hệ thống điều khiển nhiệt độ.



Hình 4.15. Sơ đồ chi tiết mạch điều khiển nhiệt độ.

- Đo nhiệt độ tại đối tượng thông qua sensor nhiệt LM335. LM335 là sensor đo nhiệt độ với đầu ra là 10mV/^{0}K, do đó để đo độ C ta cần có công thức chuyển đổi giá trị từ độ K sang độ C. Vì ta dùng ADC của PIC là 10 bit nên giá trị số lớn nhất là 1023. Vref=Vcc, giả thiết là VCC=5V nên tại 0^{o}C hay 273^{o}K thì đầu ra của LM335 có giá trị là 2.73V. Như vậy khi muốn tính toán ra độ C ta cần phải trừ đi mức điện áp là 2.73V.

Ví dụ: Nhiệt độ là 30^{o}C = 303^{o}K, mức điện áp tương ứng là

out = 303 x 10mV/^{o}K =3.03V.

Ta tính toán giá trị đọc được từ ADC.

- Với ADC 10 bit (V_in là điện áp đưa vào chân ADC của PIC):

V_in = 5V => ADC_value = 1023

V_in = 2.73V => ADC_value = (1023/5)x2.73=558.6 (tương ứng 0^0)

mặt khác do V_ref = VCC = 5V nên ADC_value = 1 tương ứng với 5/1023=4.9mV÷5mV. Trong khi đó LM335 cho ra điện áp là 10mV/1^{o}K nên để giá trị ADC thay đổi 1 đơn vị thì nhiệt độ phải thay đổi là 0.5^{o}K (hay gần 5mV) Từ đó ta có công thức đầy đủ sau để tính giá trị ^{o}C:

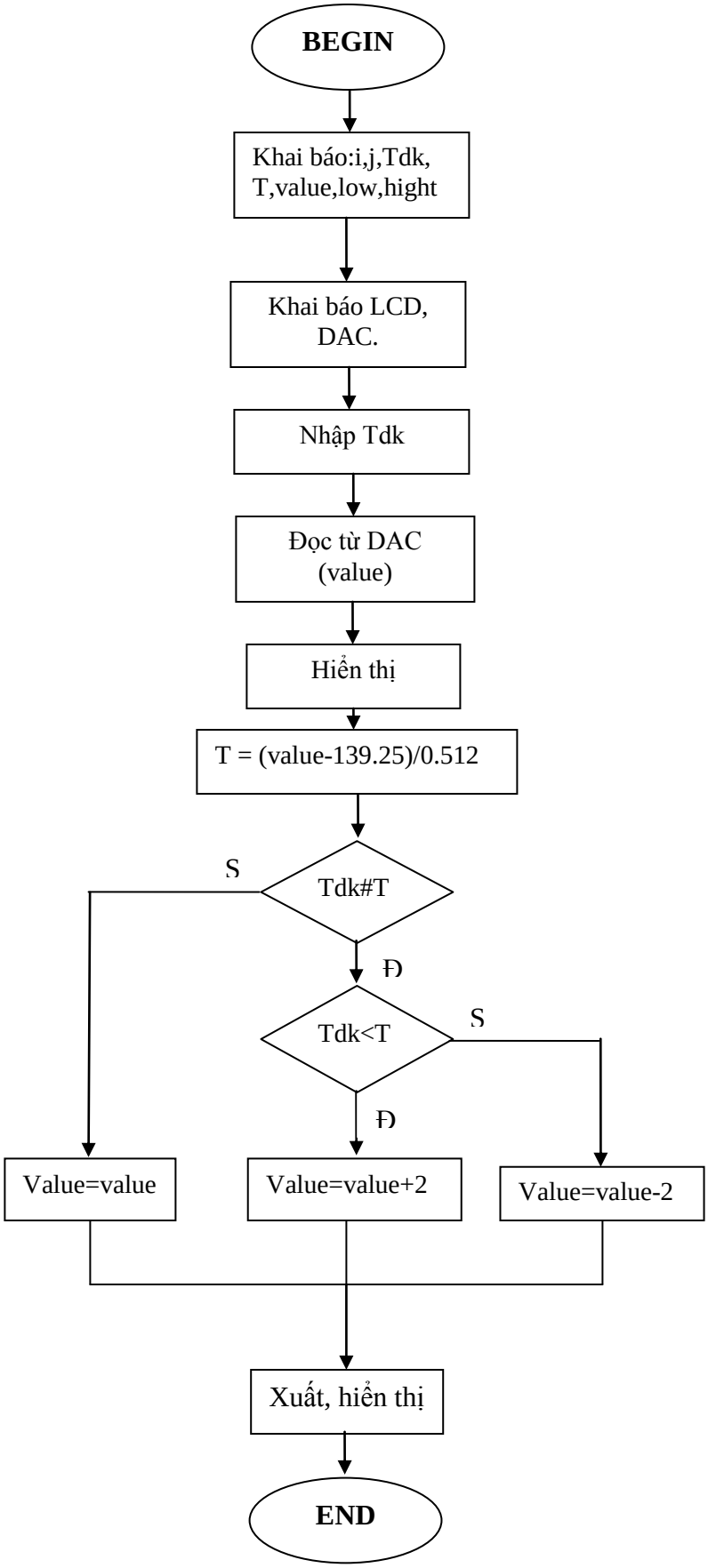
$$T[C] = \frac{(ADC\_value - 558.6).5V}{1023x10mV} \qquad (4.6)$$

Vậy ta có công thức rút gọn là:

$$T[C] = \frac{(ADC\_value - 558.6)}{2.046} \qquad (4.7)$$

3.9. Phần mềm điều khiển

3.9.1. Lưu đồ thuật toán.


BEGIN
Khai báo:i,j,Tdk, T,value,low,hight
Khai báo LCD, DAC.
Nhập Tdk
Đọc từ DAC (value)
Hiển thị
T = (value-139.25)/0.512
Tdk#T
S
Đ
Tdk<T
S
Đ
Value=value
Value=value+2
Value=value-2
Xuất, hiển thị
END

3.9.2. Chương trình.

Chương trình điều khiển nhiệt độ

```
#include <16F877A.h>
#device *=16 adc=8
#FUSES NOWDT, HS, NOPUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP, NOCPD, NOWRT
#use delay(clock=20000000)
int8 high,low,i,x,j,value;  khai báo biến
float T,Tdk;
void convert_bcd(int8 x); khai báo hàm convẻt_bcd
void xuatlcd();
void main()
  {
 i=0 ;
 j=0 ;
trisa = 0xFF;
trisb = 0x01;
trisd = 0x00;
setup_adc_ports(AN0);
setup_adc(ADC_CLOCK_INTERNAL);
delay_us(10);
    LCD_init() ; bắt đầu LCD
    printf(lcd_putchar,"   DAT - DT901   ");
    LCD_putcmd(0xC0);
```

```
printf(lcd_putchar," nhap nhiet do:  ");
delay_ms(50);
 lcd_putcmd(0xc0);
 printf(lcd_putchar,"   T =   ");
lcd_putchar(i+0x30);
 lcd_putchar(j+0x30);
 printf(lcd_putchar," C    ");
  while(true)
    {
       If(input(pin_C0))
         {
          i=i+1;
          delay_ms(10);
          if(i==10)
          {i=0;
          lcd_putcmd(0xc9);
          lcd_putchar(i+0x30);
          delay_ms(10);
          }
          else
          {LCD_putcmd(0xC9);
         lcd_putchar(i +0x30);
          }
         }
```

```
            if(input(pin_c1))
              {
                j=j+1;
                delay_ms(10);
               if(j==10)
                {
                  j=0;
                 lcd_putcmd(0xca);
                   lcd_putchar(j+0x30);
                   delay_ms(10);
                }
              else{  LCD_putcmd(0xCa);
                lcd_putchar(j+0x30);
                }
              }
            if(input(pin_c2))
               Break ;
          }
       Tdk=i*10+j;
       LCD_init();
       printf(lcd_putchar,"please wait.... ");
while(true)
  {
    value = read_adc();
```

```
      delay_ms(10);
       T = (value-139.25)/0.512;
       convert_bcd((int8)T);
       delay_us(50);
       T=high*10+low;
       LCD_init();
      printf(lcd_putchar,"  nhiet do do  :  ");
      xuatlcd();
      delay_us(50);
    if(T!=Tdk)
       if(T<Tdk)
           value=value+2;
       else
           value=value-2;
   else
           value=value;
           output_b(value);
           xuatlcd();
  }
 }
 void convert_bcd(int8 x)
{
  low=x%10; lấy hàng đơn vị nhiệt độ
  high=x/10; lấy hang chục và hang trăm
```

```
low=low+0x30;
high=high+0x30;
}
void xuatlcd()
{
   LCD_putcmd(0xC0);
   printf(LCD_putchar,"   T =   ");
   lcd_putchar(high+0x30);
   lcd_putchar(low+0x30);
   printf(LCD_putchar," C    ");
}
```

KẾT LUẬN

Sau ba tháng nghiên cứu và tìm hiểu, với sự h- ớng dẫn của thầy Nguyễn Văn D- ơng và sự giúp đỡ của các thầy cô trong khoa Điện- Điện tử, em đã hoàn thành đ- ợc đồ án tốt nghiệp của mình.

Qua đồ án này em đã thu đ- ợc những kết quả sau

- Hiểu đ- ợc các ph- ơng pháp đo l- ờng thông qua Vi Điều Khiển, đặc biệt là đo l- ờng thông qua PIC16F877A.
- Biết đ- ợc ph- ơng pháp lập trình bằng C phục vụ cho lập trình Vi Điều Khiển.
- Tìm hiểu đ- ợc các loại cảm biến thông dụng dùng trong đo l- ờng.
- Xây dựng đ- ợc một hệ thống đo l- ờng cơ bản.

Tuy nhiên đề tài vẫn còn có những hạn chế:

Mở rộng đề tài chúng ta có thể thiết kế hệ thống và00 hiển thị nhiệt độ trên LED 7 đoạn hoặc thiết kế hệ thống điều khiểu nhiệt độ lò công nghiệp...

Do hạn chế về kiến thức, kinh nghiệm và tài liệu nên đề tài không tránh khỏi những thiếu sót. Em rất mong thầy cô và các bạn giúp đỡ chỉ bảo để em có thể học hỏi đ- ợc nhiều hơn nữa.

TÀI LIỆU THAM KHẢO

1. Nguyễn Tăng C- ờng, Phan Quốc Thắng, Cấu trúc và lập trình họ Vi Điều khiển 8051, Nhà xuất bản khoa học và Kỹ Thuật.

2 Nguyễn Mạnh Giang, Cấu trúc, lập trình ghép nối và ứng dụng của Vi Điều Khiển, nhà xuất bản Lao Động – Xã Hội.

3. Phạm Minh Hà(2004), Kỹ thuật mạch điện tử, Nhà xuất bản Khoa học và kỹ thuật.

4. Ngô Diện Tập, Vi Điều Khiển trong đo l- ờng và điều khiển tự động, Nhà xuất bản Khoa Hoc và Kỹ Thuật, Hà Nội.

5. Họ Vi Điều Khiển 8051, Tống Văn ON, nhà Xuất bản Lao Động và Xã Hội.

6. Các bạn có thể truy cập các trang Web rất hay của Việt Nam nh- :

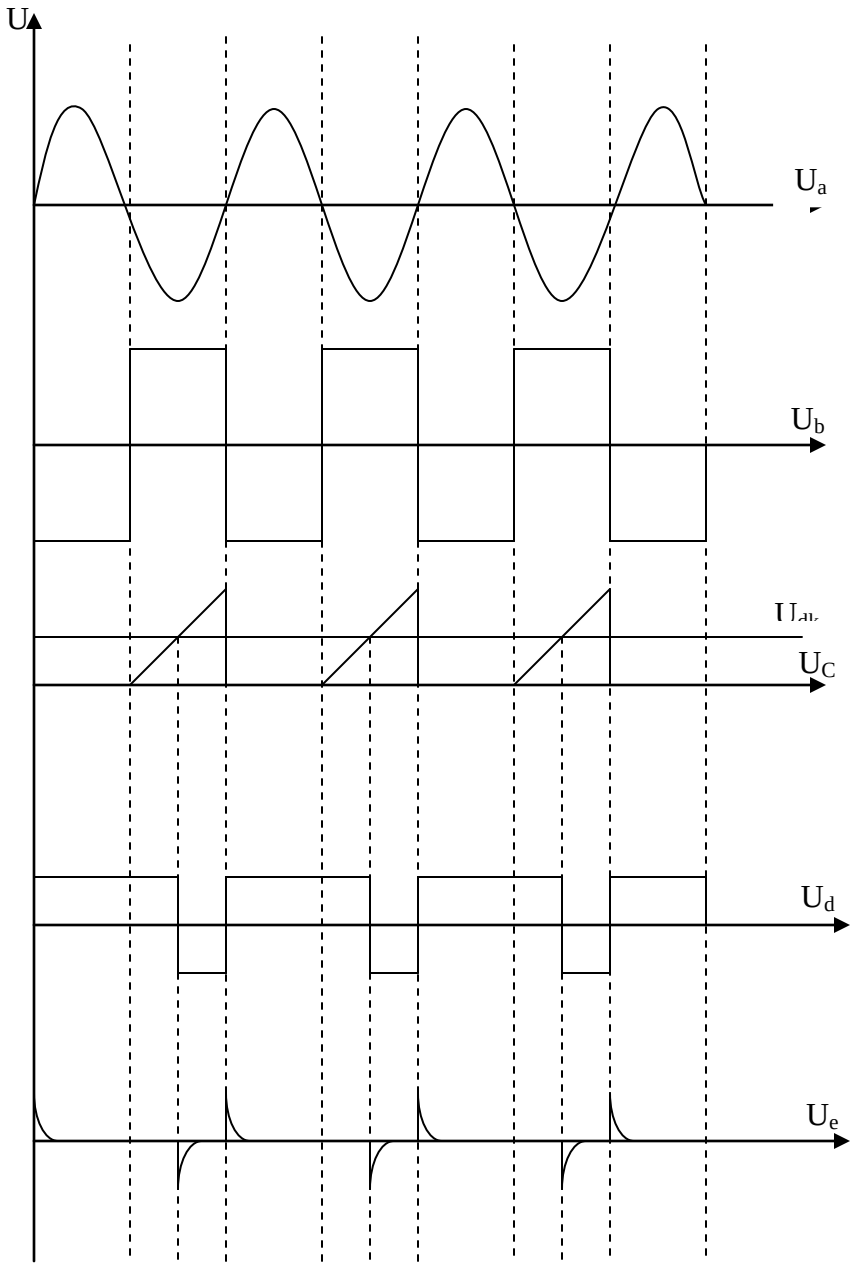
www.dientuvietnam.net

www.picvietnam.com

www.dientuvienthong.net

www.vagam.dieukhien.net

www.duyphi.phpnet.us/index.htm


U
Ua
Ub
Uđk
UC
Ud
Ue