

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

THIẾT KẾ HỆ THỐNG THƯỚC ĐO TUYẾN TÍNH

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: ĐIỆN TỬ VIỄN THÔNG**

HẢI PHÒNG - 2009

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

THIẾT KẾ HỆ THỐNG THƯỚC ĐO TUYẾN TÍNH

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: ĐIỆN TỬ VIỄN THÔNG**

**Người hướng dẫn : Th.S NGUYỄN VĂN DƯƠNG
Sinh viên : NGUYỄN THỊ PHƯƠNG THẢO**

HẢI PHÒNG - 2009

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Nguyễn Thị Phương Thảo

Mã số: 090246

Lớp : ĐT 901

Ngành : Điện tử viễn thông

Tên đề tài: Thiết kế hệ thống thước đo tuyến tính.

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Các số liệu cần thiết để thiết kế, tính toán.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Địa điểm thực tập tốt nghiệp.

Đài phát thanh và truyền hình Hải Phòng.....

.....

.....

.....

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên : Nguyễn Văn Dương

Học hàm, học vị : Thạc sĩ

Cơ quan công tác : Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn : Toàn bộ đề tài

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:.....

Nội dung hướng dẫn:

Đề tài tốt nghiệp được giao ngày 10 tháng 04 năm 2009

Yêu cầu phải hoàn thành xong trước ngày 10 tháng 07 năm 2009

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Nguyễn Thị Phương Thảo

Đã giao nhiệm vụ ĐTTN

Người hướng dẫn

Th.s Nguyễn Văn Dương

Hải Phòng, ngàytháng.....năm 2009

HIỆU TRƯỞNG

GS.TS.NGŨT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....
.....

Đánh giá chất lượng của đồ án (so với nội dung yêu cầu đã đề ra trong nhiệm vụ ĐTTN trên các mặt lý luận, thực tiễn, tính toán số liệu...):

.....
.....
.....
.....
.....
.....

Cho điểm của cán bộ hướng dẫn (điểm ghi bằng số và chữ)

.....
.....
.....
.....

Hải phòng, ngày.....tháng.....năm 2009

Cán bộ hướng dẫn

(Họ tên và chữ ký)

**NHẬN XÉT ĐÁNH GIÁ CỦA NGƯỜI CHẤM PHẢN BIỆN
ĐỀ TÀI TỐT NGHIỆP**

1. Đánh giá chất lượng đề tài tốt nghiệp về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh và bản vẽ, giá trị lý luận và thực tiễn đề tài.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cho điểm của cán bộ chấm phản biện (điểm ghi bằng số và chữ).

.....

.....

.....

.....

.....

.....

Hải phòng, ngày.....tháng.....năm 2009

Người chấm phản biện
(Họ tên và chữ ký)

MỤC LỤC

LỜI MỞ ĐẦU	11
Chương 1.....	13
TỔNG QUAN VỀ VI ĐIỀU KHIỂN.....	13
1.1. Sơ lược về vi xử lý và vi điều khiển.	13
1.2. Tổng quan về PIC16F877A	17
1.2.1 Hình dạng và bố trí chân của Pic16F877A.....	17
1.2.2. Đặc tính nổi bật của bộ xử lý.....	18
1.2.3. Sơ đồ khối bộ vi điều khiển Pic16F877A.....	18
1.2.4. Mô tả các chân chức năng của Pic16F877A.....	19
1.2.5. Tổ chức bộ nhớ.	22
1.2.6. Đọc và ghi vào bộ nhớ dữ liệu EEPROM.	28
1.2.7. Đọc và ghi chương trình FLASH.	28
1.2.8. Cổng vào ra.	28
1.2.9. Các bộ Timer của chip.	34
1.2.10. Bộ chuyển đổi tương tự sang số.	42
1.2.11. Các ngắt của PIC16F877A.	44
1.3. So sánh với Vi Điều Khiển 8051.	45
Chương 2.....	47
THIẾT BỊ HIỂN THỊ LCD.....	47
2.1. Hình dáng kích thước.....	47
2.2. Các chân chức năng.	48
2.3. Sơ đồ khối của HD44780.....	49
2.4. Tập lệnh của LCD.	54
2.5. Đặc tính của các chân giao tiếp.....	60
CHƯƠNG 3.	62

THIẾT KẾ HỆ THỐNG THƯỚC ĐO TUYẾN TÍNH	62
3.1. Sơ đồ khối của hệ thống:.....	62
3.2. Thiết kế các khối	63
3.2.1. Khối nguồn:.....	63
3.2.2 Hệ thống thước:	64
3.2.3. Bộ vi xử lý	65
3.2.4. Khối hiển thị	67
3.3. Sơ đồ mạch chi tiết:	68
3.4. Thuật toán.	68
3.5. Chương trình :	70
3.6. Sơ đồ mạch in:	72
Kết luận	73
Tài liệu tham khảo.....	74

LỜI MỞ ĐẦU

Ngày nay, kĩ thuật điện tử được áp dụng hết sức rộng rãi trong nhiều lĩnh vực khoa học công nghệ và đời sống, đặc biệt là trong việc tự động hóa. Việc cho phép xác định tọa độ để điều khiển định vị cho các đối tượng dịch chuyển, cụ thể được ứng dụng như trong hàn tự động, dùng trong việc sản xuất dây truyền như công nghệ đóng hộp các loại..., hay còn được sử dụng để đo mức, báo hiệu độ sâu, độ cạn của chất lỏng.

Qua đó chúng ta thấy được sự quan trọng của vi điều khiển, các loại cảm biến... trong đo lường. Nhờ việc sử dụng những linh kiện điện tử này chúng ta có thể thu thập được các đại lượng cần đo dễ dàng hơn và có thể xử lý ngay các đại lượng đó và đưa ra những kết quả mong muốn, đây cũng là nội dung mà em trình bày trong đề tài của mình.

Đề tài của em là “ Thiết kế hệ thống thước đo tuyến tính”. Trong hệ thống em sử dụng bộ vi điều khiển 8 bit PIC16F877A, cảm biến để xác định hướng chuyển động theo chu trình có cả tiến và lùi. Sau đó, kết quả được hiển thị lên LCD HD44780.

Đồ án của em gồm 3 chương:

Chương 1. Tổng quan về vi điều khiển.

Chương 2. Giới thiệu về LCD HD44780.

Chương 3. Thiết kế hệ thống thước đo tuyến tính.

Trong quá trình làm đồ án tốt nghiệp, do hạn chế về thời gian, tài liệu cũng như trình độ có hạn nên không tránh khỏi có thiếu sót. Em rất mong được sự giúp đỡ của nhà trường, của thầy cô trong hội đồng và các bạn để đồ án tốt nghiệp của em được hoàn thiện hơn.

Em xin gửi lời cảm ơn chân thành đến nhà trường cùng 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, ngày 10 tháng 07 năm 2009

Sinh viên

Nguyễn Thị Phương Thảo

Chương 1

TỔNG QUAN VỀ VI ĐIỀU KHIỂN

1.1. Sơ lược về vi xử lý và vi điều khiển.

Trong những thập niên cuối thế kỉ XX, từ sự ra đời của công nghệ bán dẫn, kĩ thuật điện tử đã có sự phát triển vượt bậc. Các thiết bị điện tử sau đó đã được tích hợp với mật độ cao và rất cao trong các diện tích nhỏ, nhờ vậy các thiết bị điện tử nhỏ hơn và nhiều chức năng hơn. Các thiết bị điện tử ngày càng nhiều chức năng trong khi giá thành ngày càng rẻ hơn, chính vì vậy điện tử có mặt khắp mọi nơi.

Bước đột phá mới trong công nghệ điện tử, công ty trẻ tuổi Intel cho ra đời bộ vi xử lý đầu tiên, tức là phần cứng chỉ đóng vai trò thứ yếu, phần mềm (chương trình) đóng vai trò chủ đạo đối với các chức năng cần thực hiện. Nhờ vậy vi xử lý có sự mềm dẻo hóa trong các chức năng của mình. Ngày nay vi xử lý có tốc độ tính toán rất cao và khả năng xử lý rất lớn.

Vi xử lý có các khối chức năng cần thiết để lấy dữ liệu, xử lý dữ liệu và xuất dữ liệu ra ngoài sau khi đã xử lý. Và chức năng chính của Vi xử lý chính là xử lý dữ liệu, chẳng hạn như cộng, trừ, nhân, chia, so sánh.v.v....Vi xử lý không có khả năng giao tiếp trực tiếp với các thiết bị ngoại vi, nó chỉ có khả năng nhận và xử lý dữ liệu mà thôi.

Để vi xử lý hoạt động cần có chương trình kèm theo, các chương trình này điều khiển các mạch logic và từ đó vi xử lý xử lý các dữ liệu cần thiết theo yêu cầu. Chương trình là tập hợp các lệnh để xử lý dữ liệu thực hiện từng lệnh được lưu trữ trong bộ nhớ, công việc thực hành lệnh bao gồm: nhận lệnh từ bộ nhớ, giải mã lệnh và thực hiện lệnh sau khi đã giải mã.

Để thực hiện các công việc với các thiết bị cuối cùng, chẳng hạn điều khiển động

cơ, hiển thị kí tự trên màn hình đòi hỏi phải kết hợp vi xử lý với các mạch điện giao tiếp với bên ngoài được gọi là các thiết bị I/O (nhập/xuất) hay còn gọi là các thiết bị ngoại vi. Bản thân các vi xử lý khi đứng một mình không có nhiều hiệu quả sử dụng, nhưng khi là một phần của một máy tính, thì hiệu quả ứng dụng của Vi xử lý là rất lớn. Vi xử lý kết hợp với các thiết bị khác được sử dụng trong các hệ thống lớn, phức tạp đòi hỏi phải xử lý một lượng lớn các phép tính phức tạp, có tốc độ nhanh. Chẳng hạn như các hệ thống sản xuất tự động trong công nghiệp, các tổng đài điện thoại, hoặc ở các robot có khả năng hoạt động phức tạp v.v..

Bộ Vi xử lý có khả năng vượt bậc so với các hệ thống khác về khả năng tính toán, xử lý, và thay đổi chương trình linh hoạt theo mục đích người dùng, đặc biệt hiệu quả đối với các bài toán và hệ thống lớn. Tuy nhiên đối với các ứng dụng nhỏ, tầm tính toán không đòi hỏi khả năng tính toán lớn thì việc ứng dụng vi xử lý cần cân nhắc. Bởi vì hệ thống dù lớn hay nhỏ, nếu dùng vi xử lý thì cũng đòi hỏi các khối mạch điện giao tiếp phức tạp như nhau. Các khối này bao gồm bộ nhớ để chứa dữ liệu và chương trình thực hiện, các mạch điện giao tiếp ngoại vi để xuất nhập và điều khiển trở lại, các khối này cùng liên kết với vi xử lý thì mới thực hiện được công việc. Để kết nối các khối này đòi hỏi người thiết kế phải hiểu biết tinh tường về các thành phần vi xử lý, bộ nhớ, các thiết bị ngoại vi. Hệ thống được tạo ra khá phức tạp, chiếm nhiều không gian, mạch in phức tạp và vấn đề chính là trình độ người thiết kế. Kết quả là giá thành sản phẩm cuối cùng rất cao, không phù hợp để áp dụng cho các hệ thống nhỏ.

Vì một số nhược điểm trên nên các nhà chế tạo tích hợp một ít bộ nhớ và một số mạch giao tiếp ngoại vi cùng với vi xử lý vào một IC duy nhất được gọi là Microcontroller-Vi điều khiển. Vi điều khiển có khả năng tương tự như khả năng của vi xử lý, nhưng cấu trúc phần cứng dành cho người dùng đơn giản hơn nhiều. Vi điều khiển ra đời mang lại sự tiện lợi đối với người dùng, họ không cần nắm vững một khối lượng kiến thức quá lớn như người dùng vi xử lý, kết cấu mạch điện dành cho người dùng cũng trở nên đơn giản hơn nhiều và có khả

năng giao tiếp trực tiếp với các thiết bị bên ngoài. Vi điều khiển tuy được xây dựng với phần cứng dành cho người sử dụng đơn giản hơn, nhưng thay vào lợi điểm này là khả năng xử lý bị giới hạn (tốc độ xử lý chậm hơn và khả năng tính toán ít hơn, dung lượng chương trình bị giới hạn). Thay vào đó, Vi điều khiển có giá thành rẻ hơn nhiều so với vi xử lý, việc sử dụng đơn giản, do đó nó được ứng dụng rộng rãi vào nhiều ứng dụng có chức năng đơn giản, không đòi hỏi tính toán phức tạp.

Vi điều khiển được ứng dụng trong các dây chuyền tự động loại nhỏ, các robot có chức năng đơn giản, trong máy giặt, ô tô v.v...

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 MCS-48. Độ phức tạp, kích thước và khả năng của Vi điều khiển tăng thêm một bậc quan trọng vào năm 1980 khi intel cho ra chip 8051, bộ Vi điều khiển đầu tiên của họ MCS-51 và là chuẩn công nghệ cho nhiều họ vi điều khiển được sản xuất sau này. Sau đó rất nhiều họ vi điều khiển của nhiều nhà chế tạo khác nhau lần lượt được đưa ra thị trường với tính năng được cải tiến ngày càng mạnh.

Các vi điều khiển thông dụng:

+ Họ vi điều khiển AMCC: do tập đoàn “Applied Micro Circuits Corporation” sản xuất. Tháng 5/2004, họ vi điều khiển này được phát triển và đưa ra thị trường bởi IBM, bao gồm:

403 PowerPC CPU

PPC 403GCX

405 PowerPC CPU

PPC 405EP

PPC 405GP/CR

PPC 405GPr

PPC NPe405H/L

440 PowerPC Book-E CPU

PPC 440GP

PPC 440GX

PPC 440EP/EPx/GRx

PPC 440SP/SPe

+ Họ vi điều khiển Atmel:

Dòng Atmel AT91 (Kiến trúc ARM THUMB)

Dòng AT90, Tiny & Mega – AVR (Atmel Norway design)

Dòng Atmel AT89 (Kiến trúc Intel 8051/MCS51)

Dòng MARC4

+ Họ vi điều khiển Freescale Semiconductor:

Năm 2004, những vi điều khiển này được phát triển và tung ra thị trường bởi Motorola.

Dòng 8-bit

68HC05 (CPU05)

68HC08 (CPU08)

68HC11 (CPU11)

Dòng 16-bit

68HC12 (CPU12)

68HC16 (CPU16)

Freescale DSP56800 (DSPcontroller)

Dòng 32-bit

Freescale 683XX (CPU32)

MPC500

MPC 860 (PowerQUICC)

MPC 8240/8250 (PowerQUICC II)

MPC 8540/8555/8560 (PowerQUICC III)

+ Họ vi điều khiển Intel

Dòng 8-bit

8XC42

MCS48

MCS51

8061

8xC251

Dòng 16-bit

80186/88

MCS96

MXS296

Dòng 32-bit

386EX

i960

+ Họ vi điều khiển Microchip

12-bit instruction PIC

14-bit instruction PIC

PIC16F84

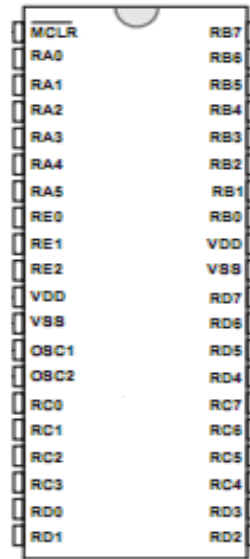
16-bit instruction PIC

Trong đó họ vi điều khiển Microchip được ứng dụng phổ biến nhất, đặc biệt là PIC16F877A được tích hợp thêm những thành phần mới như bộ chuyển đổi A/D 10 bits, và lập trình phần mềm điều khiển cũng đơn giản hơn, nên trong đề tài này em sử dụng PIC16F877A. Dưới đây em xin trình bày về PIC16F877A.

1.2. Tổng quan về PIC16F877A

PIC16F877A là chữ viết tắt của “Programmable Intelligent Computer” (máy tính khả trình thông minh), là dòng vi điều khiển phổ biến nhất được sử dụng, giá thành phù hợp, có nhiều chức năng, và hỗ trợ gần như toàn bộ chức năng của một bộ vi điều khiển hiện đại.

1.2.1 Hình dạng và bố trí chân của Pic16F877A.



Hình 1.1. Hình dạng Pic16F877A.

1.2.2. 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 là DC-20MHz.

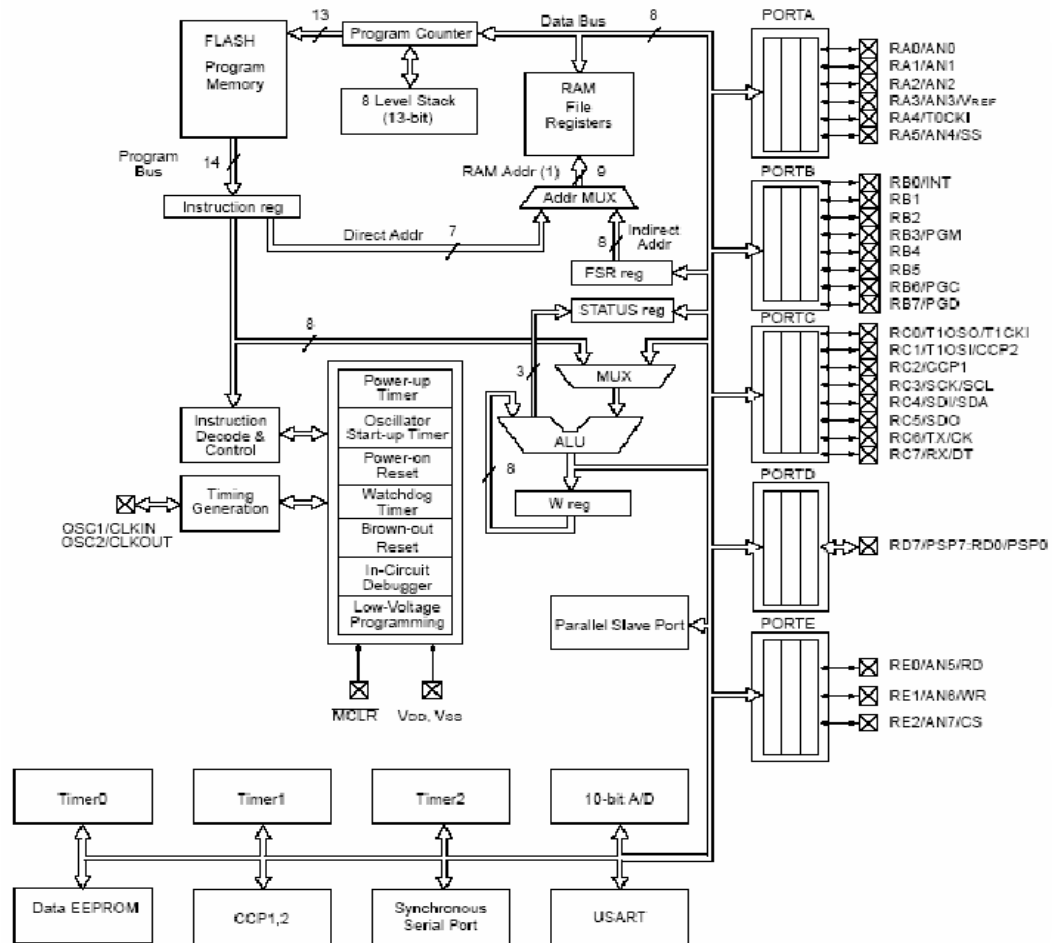
- Chu kì lệnh thực hiện trong 200ns.

Bộ nhớ chương trình Flash 8Kx14 Words.

Bộ nhớ Ram 368x8 bytes.

Bộ nhớ EPROM 256x8 bytes.

1.2.3. Sơ đồ khối bộ vi điều khiển Pic16F877A.



Hình 1.2. Sơ đồ khối của Pic16F877A.

1.2.4. Mô tả các chân chức năng của Pic16F877A.

Bảng 1.1. Bảng chân chức năng của Pic16F877A.

Tên chân	Chân số	Là chân	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.
RA0/AN0	2	I/O	PORTA là port vào ra hai chiều. 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/VRE F –	4	I/O	RA2 có thể làm ngõ vào tương tự thứ 2 hoặc điện áp chuẩn tương tự âm.
RA3/AN3/VRE F +	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.
RB0/INT	33	I/O	PORTB là port vào ra hai chiều. 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	
RC0/T1OSO/T1CKI	15	I/O	PORTC là port vào ra hai chiều. 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 hồ.
RC7/RX/DT	26	I/O	RC7 có thể là chân nhận không đồng bộ USART hoặc đồng bộ với dữ liệu.
RD0/PSP0	19	I/O	PORTD là port vào ra hai chiều hoặc là parallel slave port khi giao tiếp với bus của bộ vi xử lý.
RD1/PSP1	20	I/O	
RD2/PSP2	21	I/O	

RD3/PSP3	22	I/O	
RD4/PSP4	27	I/O	
RD5/PSP5	28	I/O	
RD6/PSP6	29	I/O	
RD7/PSP7	30	I/O	
RE0/ RD/AN5	8	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/WR /AN6	9	I/O	RE1 có thể điều khiển việc ghi parallel slave port hoặc là ngõ vào tương tự thứ 6.
RE2/CS /AN7	10	I/O	RE2 có thể điều khiển việc chọn parallel slave port hoặc là ngõ vào tương tự thứ 7.
VSS	11, 32	P	Mass
VDD	12, 31	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.

1.2.5. Tổ chức bộ nhớ.

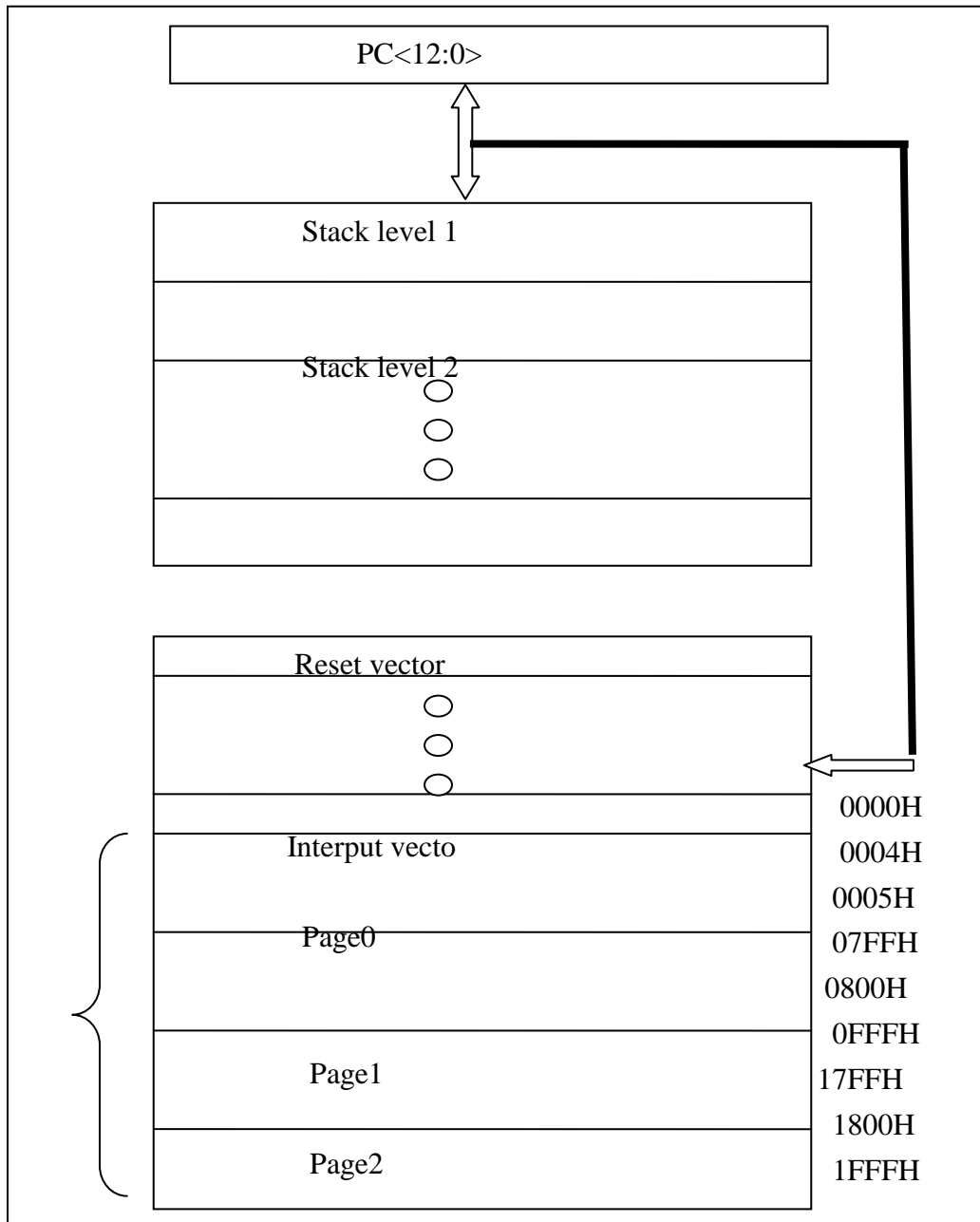
Pic16F877A có 3 khối bộ nhớ: Bộ nhớ chương trình Flash, bộ nhớ dữ liệu RAM, bộ nhớ EEPROM.

1.2.5.1. 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 xảy ra ngắt. Xem hình 1.3.



Hình 1.3. Bản đồ bộ nhớ chương trình và các ngăn xếp.

1.2.5.2. 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.

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 bit lựa chọn dãy thanh ghi cho bộ nhớ dữ liệu.

File Address	File Address	File Address	File Address
Indirect addr. (*) 00h	Indirect addr. (*) 80h	Indirect addr. (*) 100h	Indirect addr. (*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD (*) 08h	TRISD (*) 88h		
PORTE (*) 09h	TRISE (*) 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved (*) 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved (*) 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes 117h	General Purpose Register 16 Bytes 197h
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes 7Fh	General Purpose Register 80 Bytes A0h	General Purpose Register 80 Bytes 17Fh	General Purpose Register 80 Bytes 1A0h
Bank 0	Bank 1	Bank 2	Bank 3
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
	EFh FFh		1EFh 1FFh

Hình 1.4. Hình ảnh các Bank.

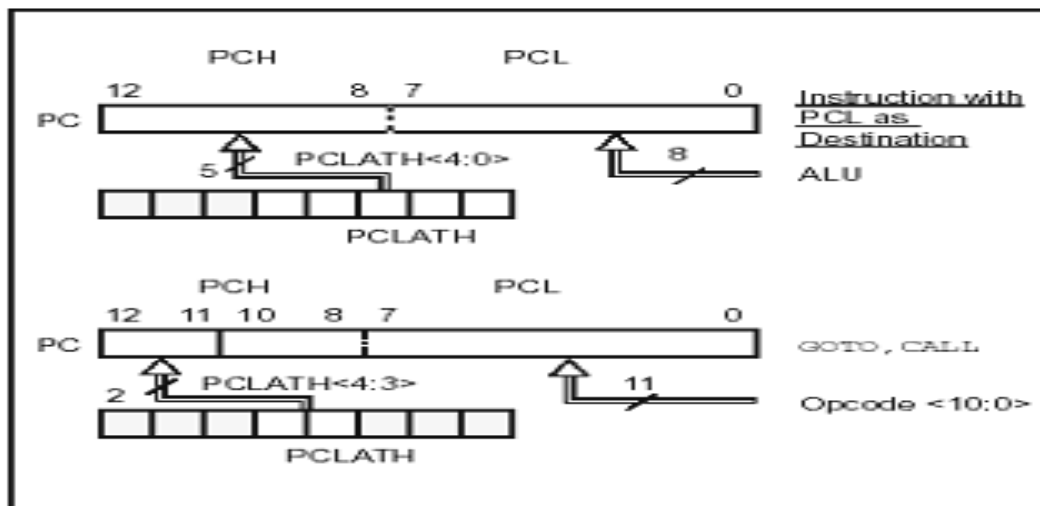
Thanh ghi trạng thái có thể là kết quả của một số lệnh như là với một số thanh ghi khác. Nếu thanh ghi trạng thái là kết quả bởi một lệnh mà tác động đến các bit Z, DC, C thì việc ghi vào các bit này là không thể.

Các thanh ghi lựa chọn OPTION_REG: có hai thanh ghi lựa chọn tại các địa chỉ 81h và 181h, các thanh ghi này có thể đọc hoặc ghi, nó chứa đựng nhiều bit 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 bit 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 bit 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 bit cờ cho phép phân biệt giữa việc Reset hệ thống (POR) để Reset MCLR ngoại với Reset WDT.



Hình 1.5. 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.

1.2.5.3 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 bit 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 bit 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.

1.2.6. 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 bit 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.

1.2.7. 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 bit 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.

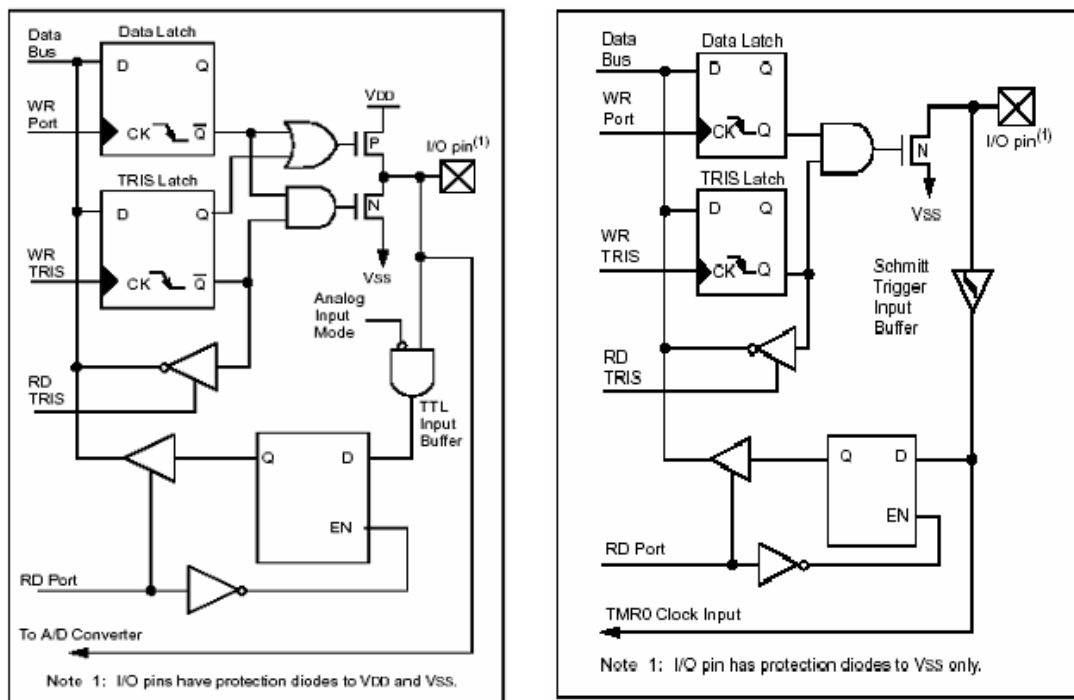
1.2.8. Cổng vào ra.

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.

1.2.8.1. 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. Nén đặt bit 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 bit 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 bit 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 1.6).

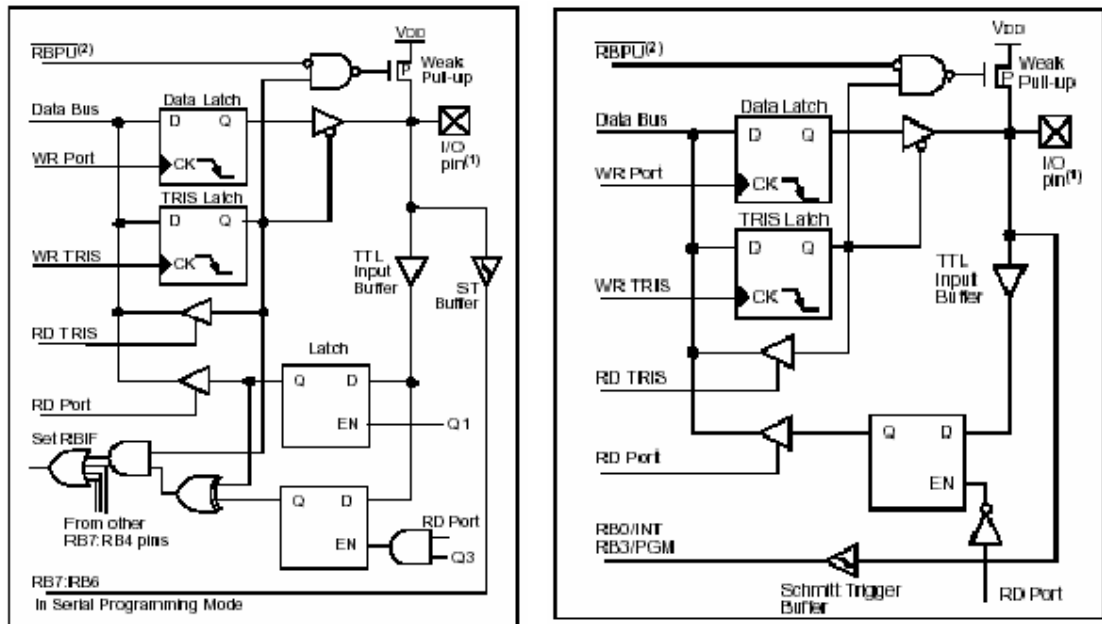


Hình 1.6 . Sơ đồ khối các chân của cổng A

1.2.8.2 Cổng B và thanh ghi TRISB.

Cổng B là cổng hai chiều với độ rộng đường truyền là 8 bit. Tương ứng với nó để điều khiển trực tiếp dữ liệu ta sử dụng thanh ghi TRISB. Nếu đặt bit TRISB=1 thì lúc này các chân của cổng B được định nghĩa là chân vào. Nếu xoá bit 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 bit RBPU (bit 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 thì 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 bit RBIF của thanh ghi INTCON. Ngắt này có thể khởi động thiết bị từ trạng thái SLEEP.

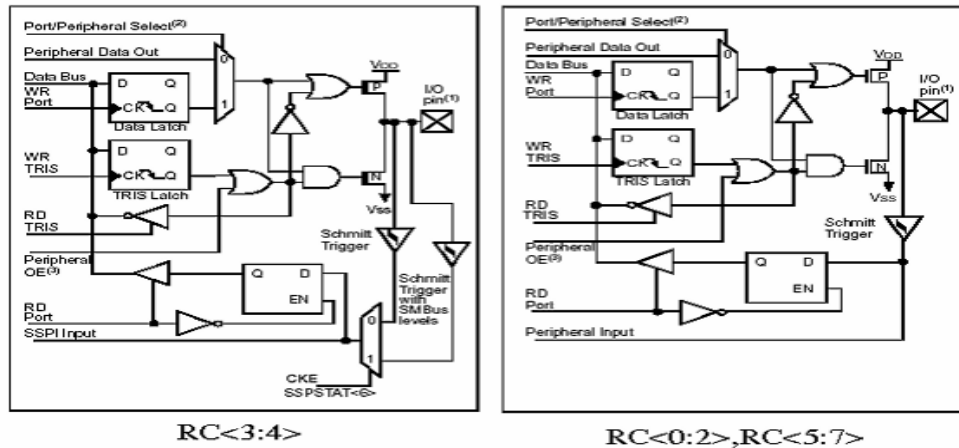


Hình 1.7. Sơ đồ khối của chân RB3 đến RB0, chân RB7:RB4 của cổng B.

1.2.8.3 Cổng C và thanh ghi TRISC.

Cổng C là cổng hai chiều với độ rộng đường truyền là 8 bit. Tương ứng với việc điều khiển nó là thanh ghi TRISC. Nếu đặt bit TRISC=1 thì tương ứng với chân của cổng C là chân vào. Nếu ta xoá bit 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 đã hợp với việc vận hành thiết bị ngoại vi. Chân của cổng C thông qua bộ đệm Schmitt Trigger đầu vào. Khi chế độ I²C hoạt động, thì các chân của cổng PORTC<4:3> có thể được sắp xếp với mức I²C thường hoặc với mức SMBUS bằng cách sử dụng bit CKE (SSPSTAT<6>) là bit 6 của thanh ghi SSPSTAT. Khi vận hành các thiết bị ngoại vi bằng việc xác định bit TRIS của mỗi chân cổng C. Một số phần phụ có thể ghi đè lên bit TRIS làm cho chân này sẽ trở thành chân ra, trong khi đó thì một số phần phụ khác lại ghi đè lên bit TRIS làm cho chân này trở thành chân vào. Từ khi những bit 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.

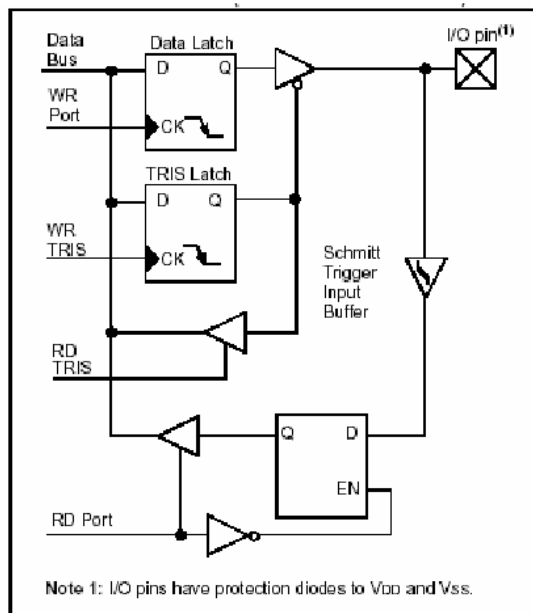
Người sử dụng nên đề cập tới việc phân chia kết nối các thiết bị ngoại vi cho việc đặt chính xác các bit TRIS.



Hình 1.8. Sơ đồ khối chân RC <0:2> RC <5:7> và chân RC <3:4> cổng C.

1.2.8.4 Cổng D và thanh ghi TRISD.

Cổng D có 8 bit 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 bit (cổng phụ song song) bằng việc đặt bit điều khiển PSPMODE (TRISE<4>) và trong chế độ này vùng đệm đầu vào là TTL.



Hình 1.9. Sơ đồ khối cổng D.

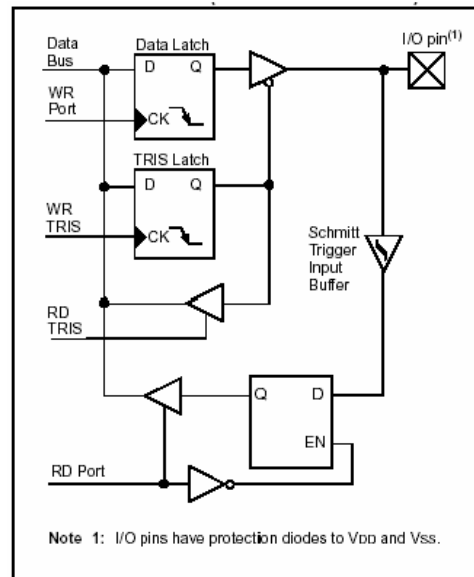
1.2.8.5. Cổng E và thanh ghi TRISE.

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 bit PSPMODE (TRISE<4>) được đặt. Và trong chế độ này phải chắc chắn rằng các bit 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 1.10. Sơ đồ khối của cổng E.

1.2.9. Các bộ Timer của chip.

Bộ vi điều khiển PIC16F87X có 3 bộ Timer đó là: Timer0, Timer1, Timer2

1.2.9.1. Bộ Timer0.

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

- + 8 bit 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 bit TOCS của thanh ghi OPTION_REG<5>.

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

Trong chế độ bộ định thời được lựa chọn bởi việc xoá bit T0CS (OPTION_REG<5>), nó sẽ được tăng giá trị sau một chu kỳ lên 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 bit T0CS (OPTION_REG<5>). Trong chế độ bộ đếm, nó sẽ được tăng ở xung đi xuống nếu xoá bit T0SE (OPTION_REG<4>) hoặc ở xung đi lên nếu đặt bit 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).

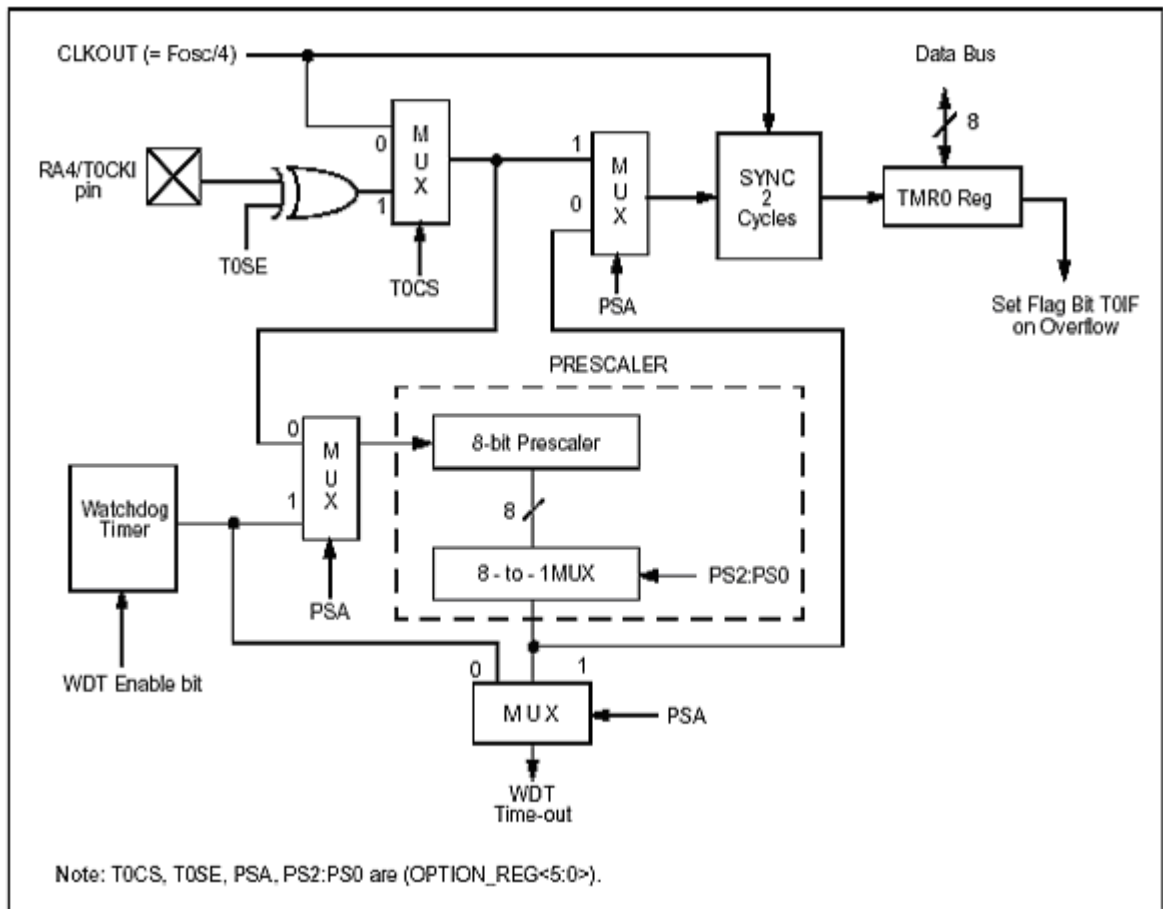
*) Các hệ số chia.

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 bit PSA của thanh ghi OPTION_REG<3>.

Những bit 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 bit T0IF của thanh ghi INTCON<2> sẽ được đặt. Bit này phải được xoá bằng phần mềm nếu cho phép ngắt bit 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 1.11. Sơ đồ khối của Timer0 và WDT.

OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP0	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Hình 1.12. Thanh ghi OPTION_REG.

Bít 5 TOCS lựa chọn nguồn clock

1 = Clock ngoài từ chân T0CKI

0 = Clock trong Fosc/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)

Bit 3 PSA gán bộ chia xung đầu vào

1 = gán bộ chia Prescaler cho WDT

0 = gán bộ chia Prescaler cho Timer 0

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

Bảng 1.2. 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

1.2.9.2. Bộ Timer1.

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

+ 16 bit 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 T1CON. Bộ Timer1 có hoạt động hay không hoạt động là nhờ việc đặt hoặc xoá bit TMR1ON (T1CON<0>).

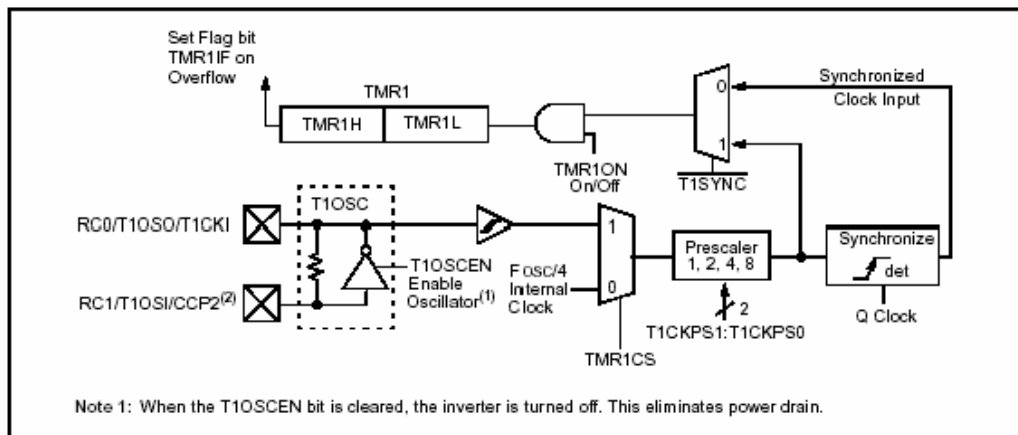
*) Hoạt động của bộ Timer1.

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

+ Là một bộ định thời 16 bit.

- + 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 bit TMR1CS (T1CON<1>). Ở chế độ bộ định thời, đầu vào là clock trong $F_{osc}/4$, bit 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 bit T1SYNC (T1CON<2>), không đồng bộ xung vào đặt bit T1SYNC (T1CON<2>) Timer1 tăng ở sườn khi xung đầu vào.



Hình 1.13. Sơ đồ khối 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 bit T1OSCEN xoá và xung vào phải đồng bộ với clock trong, ở chế độ này bộ đếm không tăng trong trạng thái SLEEP.

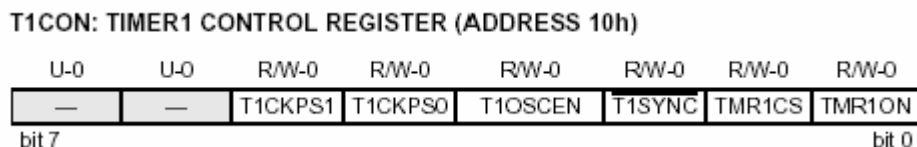
Ở chế độ bộ đếm không đồng bộ Timer1 tăng mỗi khi sườn lên ở chân RC0 hoặc ở chân RC1 nếu bit T1OSCEN xoá, ở chế độ này bộ đếm tiếp tục tăng

trong trạng thái SLEEP và có khả năng tràn gây ra ngắt 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.



Hình 1.14. Thanh ghi điều khiển 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 bit TMR1IF.

*) Thanh ghi điều khiển Timer1 T1CON:

Bit 7, 6 không sử dụng.

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

Bảng 1.3. Lựa chọn hệ số chia xung.

T1CKPS1÷T1CKPS0	
00	1:xõ lý ngÉú nhi ^a n æn ®Pnh
01	1:2
10	1:4
11	1:8

Bit 3 T1OSCEN bit điều khiển bộ dao động Timer1.

1 = Bộ dao động hoạt động.

0 = Bộ dao động không hoạt động.

Bit 2 bit điều khiển xung clock ngoài đồng bộ khi TMR1CS = 1.

Bit2 = 0 có đồng bộ clock ngoài.

Bit2 = 1 không đồng bộ clock ngoài khi TMR1CS = 0 bit này không có tác dụng.

Bit 1 TMR1CS lựa chọn nguồn xung clock vào.

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

Bit 0 bit bật tắt Timer.

1 = Timer 1 enable.

0 = Timer 1 disable.

1.2.9.3. Bộ Timer2.

Bộ Timer 2 có những đặc tính sau đây:

- + 8 bit cho bộ định thời (thanh ghi TMR2).
- + 8 bit 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.

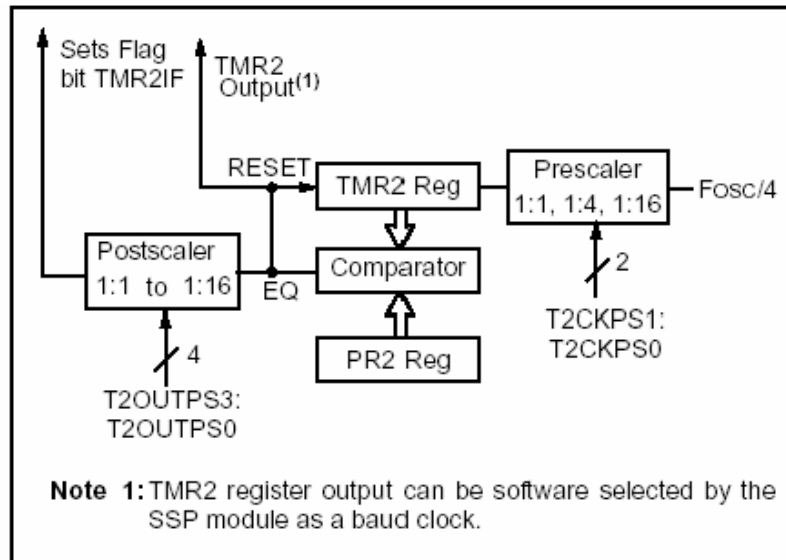
Chế độ SSP dùng đầu ra của TMR2 để tạo xung clock. Timer2 có một thanh ghi điều khiển đó là thanh ghi T2CON. Timer2 có thể tắt bằng việc xóa bit TMR2CON của thanh ghi T2CON

*) Hoạt động của bộ Timer2.

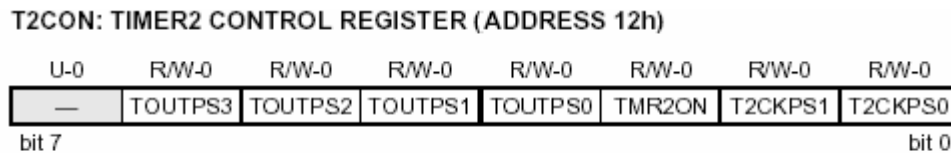
Timer2 được dùng chủ yếu ở phân điều chế xung của bộ CCP, thanh ghi TMR2 có khả năng đọc và viết, nó có thể xóa bằng việc reset lại thiết bị. Đầu vào của xung có thể chọn các tỷ lệ sau: 1:1, 1:2 hoặc 1:16 việc chọn các tỷ này có thể điều khiển các bit sau T2CKPS1 và bit T2CKPS0.

*) Ngắt của bộ Timer2.

Bộ Timer2 có 1 thanh ghi 8 bit 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 1.15. Sơ đồ khối bộ TIMER2.



Hình 1.16. Thanh ghi điều khiển timer2.

*) Thanh ghi TCON2.

Bit 7 không sử dụng.

Bit 6÷3 TOUTPS3÷TOUTPS0 bit lựa chọn hệ số đầu ra Timer2

0000 = 1:1

0001 = 1:2

0010 = 1:3

...

1111 = 1:16

Bit 2 TMR2ON bit bật tắt hoạt động Timer2

1 = enable

0 = disable

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

00 = 1:1

01 = 1:4

1x = 1:16

1.2.10. Bộ chuyển đổi tương tự sang số.

1.2.10.1 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 điện hình và giữ điện dung duy trì là đầu vào bộ chuyển đổi. Đầu ra bộ chuyển đổi A/D là 10 bit. 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).

Bit 7-6: ADCS1-ADCS0 Những bit 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)

Bit 5-3: CHS2-CHS0 Bit 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)

Bit 2: GO/DONE bit trạng thái chuyển đổi A/D.

Nếu ADON = 1 chuyển đổi A/D đang thực hiện (đặt bit 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 ADCON1 (địa chỉ 9Fh)

Bit 7 (ADFM): bit lựa chọn kết quả định dạng.

Bit 6-4: Người dùng định nghĩa.

Bit 3-0: Bit điều khiển sắp xếp công chuyển đổi A/D.

Thanh ghi ADRESH:ADRESL chứa đựng 10 bit 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. Bit GO/DONE (ADCON0<2>) được xoá và bit 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 đặt được trước khi chuyển đổi bắt đầu. Kênh vào tương tự phải có bit 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 bit ADIF.

+ Đặt bit ADIE.

+ Đặt bit PEIE.

- + Đặt bit GIE.
- 3. Đợi phụ thuộc thời gian đạt được.
- 4. Bắt đầu chuyển đổi.
- + Đặt bit GO/DONE (ADCON0).
- 5. Đợi cho chuyển đổi A/D hoàn thành.
- + Thăm dò bit 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 bit 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 T_{ad} . Giá trị nhỏ nhất đợi của 2 chu kỳ được quy định trước khi bắt đầu kế tiếp.

1.2.10.2. Lựa chọn tốc độ chuyển đổi.

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

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

Chú ý:

Nguồn RC có thời gian chu kỳ T_{ad} 4 μs nhưng có thể trong khoảng 2-6 μs .

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

1.2.11. Các ngắt của PIC16F877A.

PIC16F877A có 14 nguồn ngắt, thanh ghi INTCON là thanh ghi điều khiển các ngắt, mỗi ngắt có một bit cờ ngắt và một bit cho phép hoặc cấm ngắt. Bit GIE (INTCON<7>) điều khiển chung cho 14 ngắt khi bit này đặt thì các ngắt mới có tác dụng, khi bit GIE xoá thì tất cả các ngắt bị cấm.

Bit GIE bị xoá khi reset. Khi bit cờ ngắt thiết lập bit GIE thiết lập và bit PEIE thiết lập với ngắt ngoại vi đồng thời bit 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 bit 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 bit GIE.

Khi xảy ra ngắt PC luôn được nạp giá trị 00004h vì các ngắt được phân biệt bởi bit 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.

1.3. 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 kỹ 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 một đặc điểm hơn hẳn so với 8051 là có 10 bit 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.

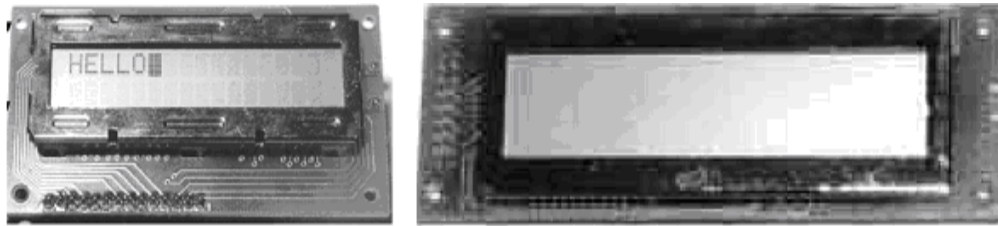
Chương 2

THIẾT BỊ 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 em sử dụng HD44780 của Hitachi, một loại thiết bị hiển thị LCD rất thông dụng ở nước ta.

2.1. Hình dáng kích thước.

Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau, được thể hiện trên hình 2.1 và hình 2.2.

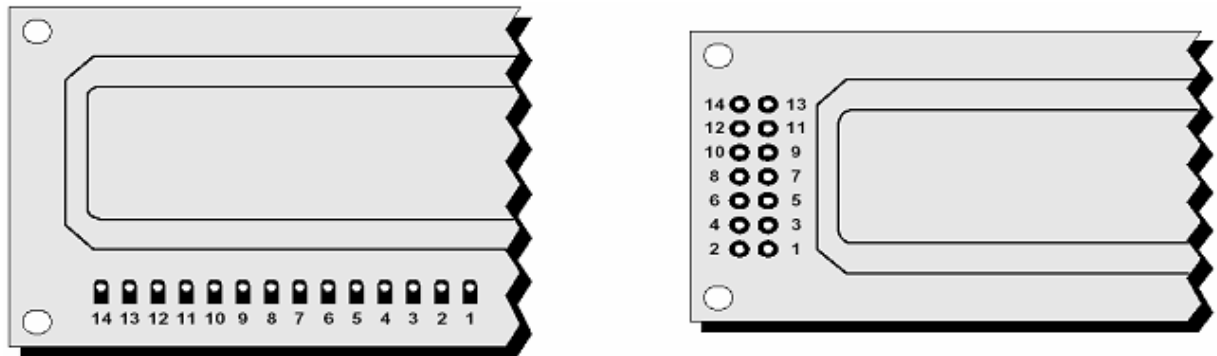


Hình 2.1. Hai loại LCD thông dụng.



Hình 2.2. LCD loại DM 1602A.

Khi sản xuất LCD, nhà sản xuất đã tích hợp chip điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết. Các chân này được đánh số thứ tự và đặt tên như hình 2.3



Hình 2.3. Hình dạng chân của LCD

2.2. Các chân chức năng.

Bảng 2.1. 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	<p>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.</p> <p>+ 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.</p> <p>+ 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.</p>
7÷ 14	DB0 ÷DB7	<p>8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này:</p> <p>+ Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.</p> <p>+ Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.</p>
15	A	15 là Catot, điện áp khoảng $U_{ak}=4,2V$
16	K	Chân nối đất của đèn Back light

2.3. Sơ đồ khối của HD44780.

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

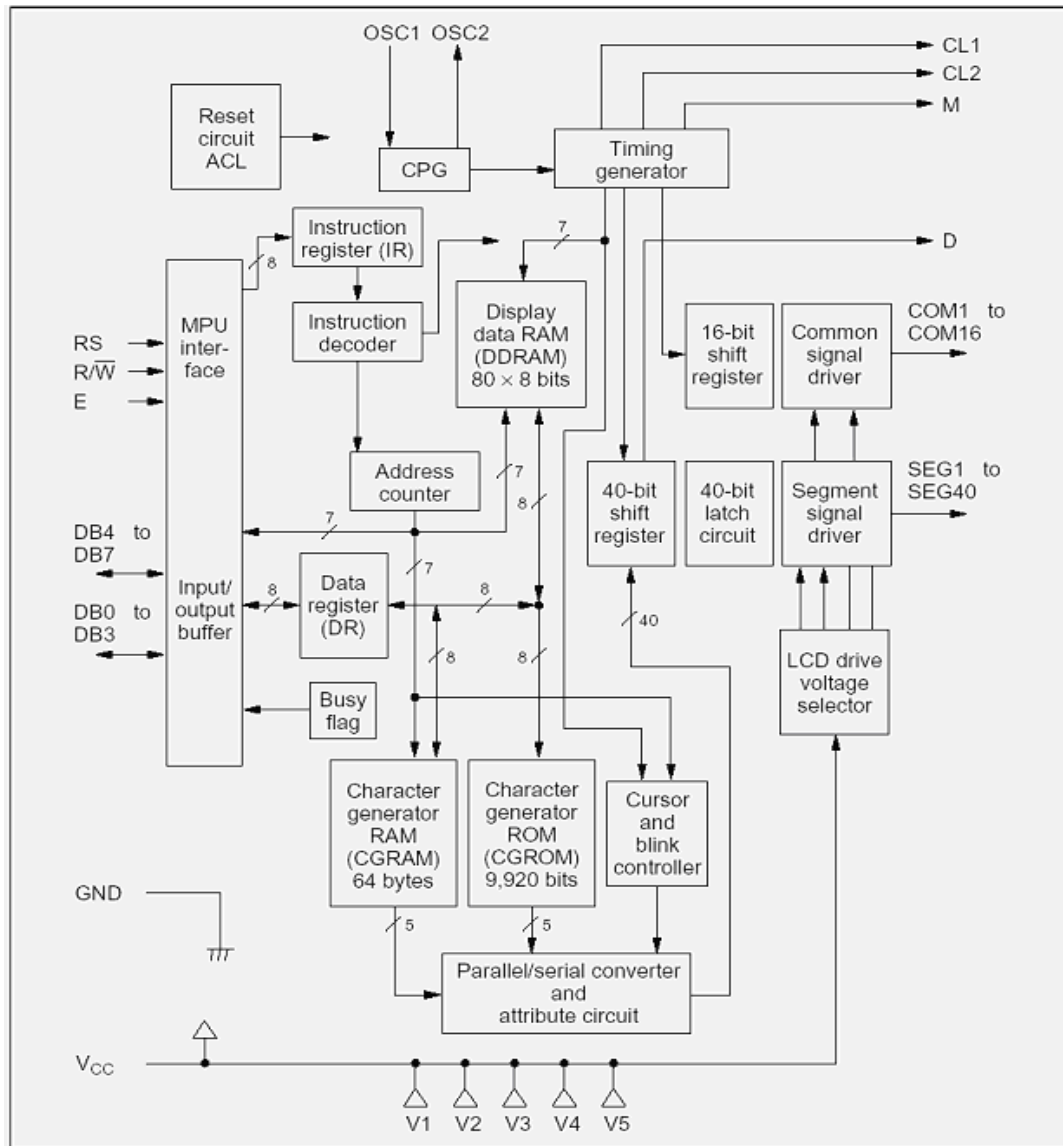
*) Các thanh ghi:

Chip 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, chip HD44780 sẽ tra bảng mã lệnh tại địa chỉ mà IR cung cấp và thực hiện lệnh đó.

VD : Lệnh “hiển thị màn hình” có địa chỉ lệnh là 00001100 (DB7...DB0)



Hình 2.4: 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 gửi ra cho MPU (ở chế độ đọc). Nghĩa là, khi MPU ghi thông tin

vào DR, mạch nội bên trong chip 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 giữa 2 thanh ghi này trong khi giao tiếp với MPU. Bảng 2.2. tóm tắt lại các thiết lập đối với hai chân RS và R/W theo mục đích giao tiếp.

Bảng 2.2. 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 chip, mạch nội bên trong cần một khoảng thời gian để hoàn tất. Khi đang thực thi các hoạt động bên trong chip 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 2.2). 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 4 μ S-5 μ S (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 2.5.

Display position (digit)	1	2	3	4	5	79	80
DDRAM address (hexadecimal)	00	01	02	03	04	4E	4F

Figure 2 1-Line Display

Display position	1	2	3	4	5	39	40
DDRAM address (hexadecimal)	00	01	02	03	04	26	27
	40	41	42	43	44	66	67

Figure 4 2-Line Display

Hình 2.5: 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. Thể hiện trên hình 2.6

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	O0
												1	0	0	0	0
												1	0	0	0	0
												1	0	1	1	0
												1	1	0	0	1
												1	0	0	0	1
												1	0	0	0	1
												1	1	1	1	0
0	1	1	0	0	0	1	0					0	0	0	0	0
												1	0	0	0	0
												1	0	0	1	0
												1	0	1	0	0
												1	0	1	1	0
												1	1	0	0	0
												1	1	0	1	0
												1	1	1	0	0
												1	1	1	1	0

← Cursor position

Hình 2.6. 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 2.7.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)								CGRAM Address								Character Patterns (CGRAM data)													
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0								
High				Low				High				Low				High				Low									
0 0 0 0 * 0 0 0								0 0 0								0	0	0	* * *					1	1	1	1	0	Character pattern (1)
																0	0	1	↑					1	0	0	0	1	
																0	1	0	↓					1	0	0	0	1	
																0	1	1	↑					1	1	1	1	0	
																1	0	0	↓					1	0	1	0	0	
																1	0	1	↑					1	0	0	1	0	
																1	1	0	↓					1	0	0	0	1	
0 0 0 0 * 0 0 1								0 0 1								0	0	0	* * *					1	0	0	0	1	Character pattern (2)
																0	0	1	↑					0	1	0	1	0	
																0	1	0	↓					1	1	1	1	1	
																0	1	1	↑					0	0	1	0	0	
																1	0	0	↓					1	1	1	1	1	
																1	0	1	↑					0	0	1	0	0	
																1	1	0	↓					0	0	1	0	0	
0 0 0 0 * 1 1 1								1 1 1								0	0	0	* * *					0	0	0	0	0	Cursor position
																0	0	1	↑										
																1	0	0	↓										
																1	0	1	↑										
																1	1	0	↓										
																1	1	1	↑										
																1	1	1	↓										

Hình 2.7. Mối liên hệ giữa địa chỉ của CGRAM, dữ liệu CGARM, và mã kí tự.

2.4. 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 giữa 2 thanh ghi này. (xem bảng 2.2)

* 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 2.3. Tập lệnh của LCD.

Tên lệnh	Hoạt động	Thời gian chạy
Clear Display	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 0 0 0 0 0 0 1</p> <p>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.</p>	
Return Home	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 0 0 0 0 0 1 *</p> <p>Lệnh Return home trả bộ đếm địa chỉ AC về 0, trả lại kiểu</p>	1.52 ms

	<p>hiển thị góc nếu nó bị thay đổi. Nội dung của DDRAM không thay đổi.</p>	
<p>Entry mode set</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 0 1 [I/D] [S]</p> <p>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.</p> <p>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.</p> <div style="text-align: center;"> <p>Figure 3 1-Line by 8-Character Display Example Figure 5 2-Line by 8-Character Display Example</p> </div>	<p>37μs</p>
<p>Display on/off control</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 1 [D] [C] [B]</p> <p>D: Hiển thị màn hình khi D=1 và ngược lại. Khi tắt hiển thị, nội dung DDRAM không thay đổi.</p> <p>C: Hiển thị con trỏ khi C=1 và ngược lại. Vị trí và hình</p>	

	<p>dạng con trỏ, xem hình 3.8.</p> <p>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.</p> <div data-bbox="505 501 1338 961" style="text-align: center;"> <p>Figure 12 Cursor and Blinking</p> </div>	37μs															
Cursor or display shift	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 1 [S/C] [R/L] * *</p> <p>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:</p> <table border="1" data-bbox="511 1581 1338 1814" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #ADD8E6;"> <th>S/C</th> <th>R/L</th> <th>Hoạt động</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).</td> </tr> <tr> <td>0</td> <td>1</td> <td>Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).</td> </tr> <tr> <td>1</td> <td>0</td> <td>Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.</td> </tr> </tbody> </table>	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.	37μs
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.															

<p>Function Set</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 0 1 [DL] [N] [F] * *</p> <p>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 gửi/nhận sau.</p> <p>N: Thiết lập số hàng hiển thị. Khi N=0: hiển thị 1 hàng, N=1: hiển thị 2 hàng.</p> <p>F: Thiết lập kiểu kí tự. Khi F=0: kiểu kí tự 5x8 điểm ảnh, F=1: kiểu kí tự 5x10 điểm ảnh.</p> <p>* Chú ý:</p> <ul style="list-style-type: none"> 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. 	<p>37μs</p>
<p>Set CGRAM Address</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx= 0 1 [ACG][ACG][ACG][ACG][ACG][ACG]</p> <p>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.</p>	<p>37μs</p>
<p>Set</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1</p>	

<p>DDRAM Address</p>	<p>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.</p>	<p>37μs</p>
<p>Read BF and address</p>	<p>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.</p>	<p>0μs</p>

<p>Write ata to CG or DDRAM</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = [Write data] (RS=1, R/W=0)</p> <p>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.</p>	<p>37μs tADD 4μs</p>
<p>Read data From CG or DDRAM</p>	<p>Mã lệnh: DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = [Read data] (RS=1, R/W=1)</p> <p>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.</p>	<p>37μs tADD 4μs</p>

2.5. Đặc tính của các chân giao tiếp.

LCD sẽ bị hỏng nghiêm trọng, hoặc hoạt động sai lệch nếu bạn vi phạm khoảng đặc tính điện sau đây:

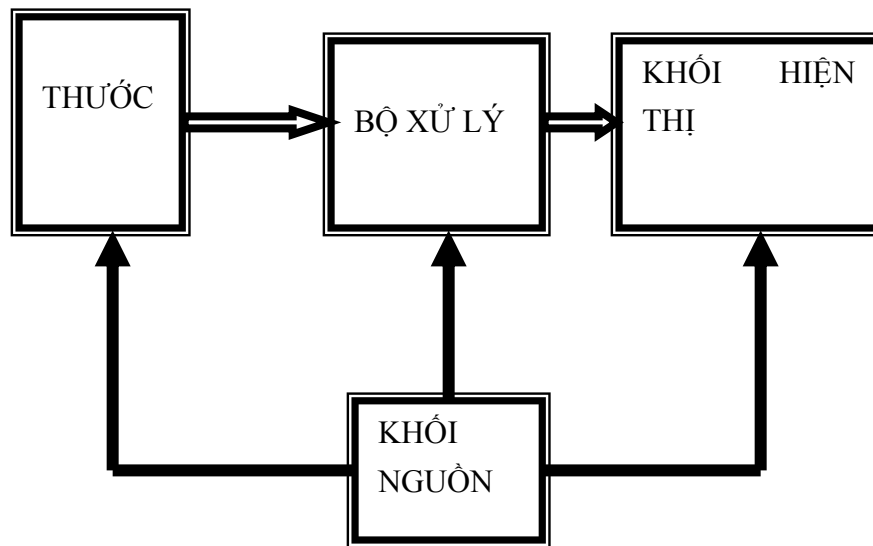
Bảng 2.4. Đặc tính điện làm việc điển hình.

Chân cấp nguồn (Vcc-GND)	Min:-0.3V , Max+7V
Các chân ngõ vào (DBx,E,...)	Min:-0.3V , Max:(Vcc+0.3V)
Nhiệt độ hoạt động	Min:-30C , Max:+75C
Nhiệt độ bảo quản	Min:-55C , Max:+125C

Đặc tính điện làm việc điển hình: (Đo trong điều kiện hoạt động $V_{cc} = 4.5V$ đến $5.5V$, $T = -30$ đến $+75C$).

Bảng 2.5. Miền làm việc bình thường.

Chân cấp nguồn V_{cc} -GND	2.7V đến 5.5V
Điện áp vào mức cao V_{IH}	2.2V đến V_{cc}
Điện áp vào mức thấp V_{IL}	-0.3V đến 0.6V
Điện áp ra mức cao (DB0-DB7)	Min 2.4V (khi $I_{OH} = -0.205mA$)
Điện áp ra mức thấp (DB0-DB7)	Max 0.4V (khi $I_{OL} = 1.2mA$)
Dòng điện ngõ vào (input leakage current) I_{LI}	-1uA đến 1uA (khi $V_{IN} = 0$ đến V_{cc})
Dòng điện cấp nguồn I_{CC}	350uA(typ.) đến 600uA
Tần số dao động nội f_{osc}	190kHz đến 350kHz (điển hình là 270kHz)

CHƯƠNG 3.**THIẾT KẾ HỆ THỐNG THƯỚC ĐO TUYẾN TÍNH****3.1. Sơ đồ khối của hệ thống:**

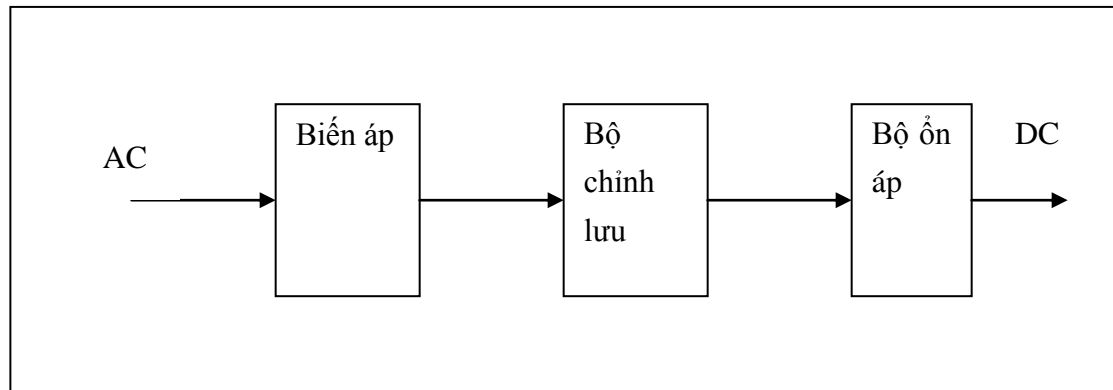
Hình 3.1. Sơ đồ khối của hệ thống

Tín hiệu nhận vào từ thước được đưa vào bộ vi xử lý, bộ vi xử lý làm nhiệm vụ biến đổi từ tín hiệu vào chuyển thành các giá trị đếm tiến hoặc lùi xác định tọa độ thước đo, sau đó đưa ra hiển thị tọa độ trên khối hiển thị.

3.2. Thiết kế các khối

3.2.1. Khối nguồn:

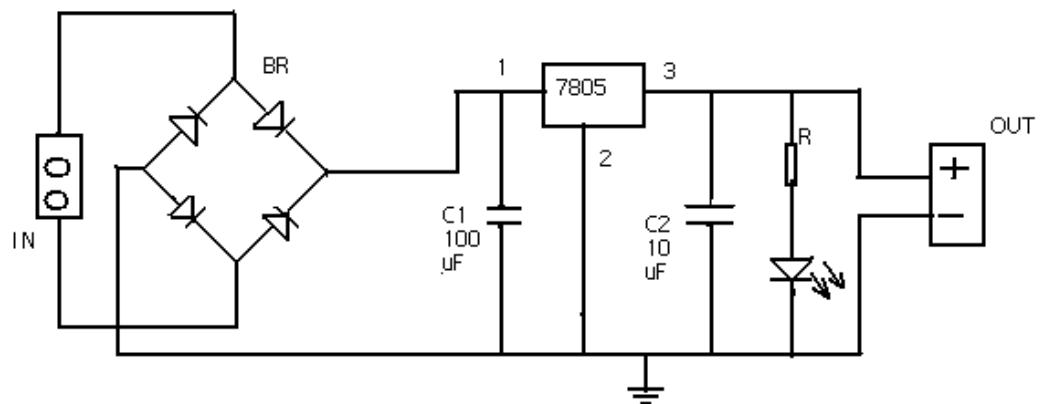
3.2.1.1. Sơ đồ khối



Hình 3.2. Sơ đồ khối mạch nguồn

Điện áp xoay chiều AC được đưa qua bộ biến áp để biến đổi thành điện áp một chiều. Sau đó qua bộ chỉnh lưu để cho ra điện áp một chiều duy nhất nhưng vẫn còn nhấp nhô. Khi đi qua bộ ổn áp, điện áp này được ổn định. Và đầu ra thu được điện áp một chiều DC.

3.2.1.2. Sơ đồ nguyên lý



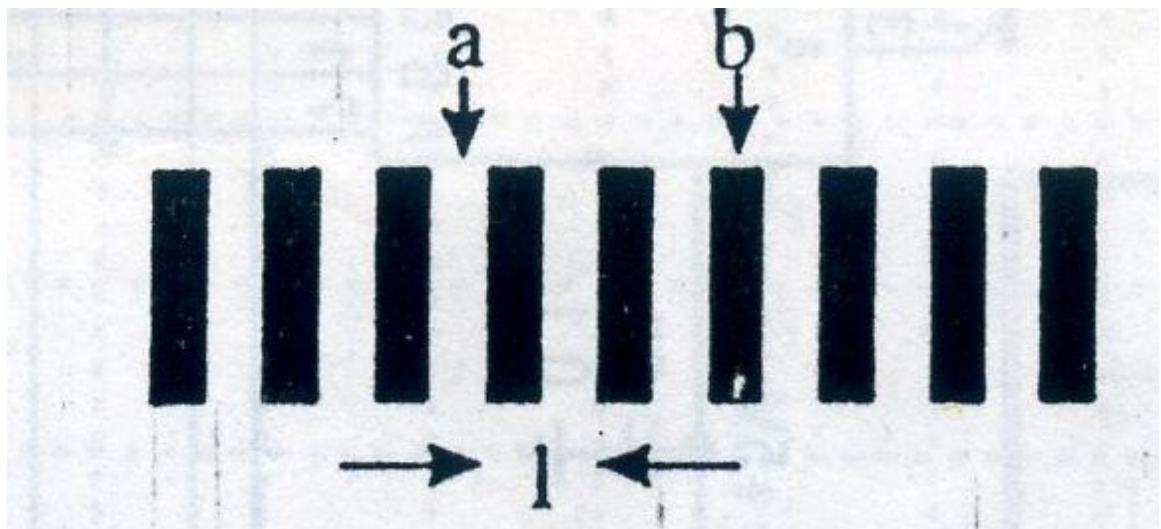
Hình 3.3. Sơ đồ nguyên lý

Nguồn ổn định cung cấp cho hệ thống là nguồn +5V.
Mạch trên lấy nguồn một chiều từ một máy biến áp với điện áp từ 0V đến 12V đưa vào ngõ vào IN, rồi cho qua một cầu diode để đảm bảo cực tính của nguồn cấp cho mạch theo một chiều duy nhất. Sau đó cho qua IC ổn áp 7805 để tạo ngõ ra OUT +5V ổn định cấp cho toàn mạch. Tụ điện đóng vai trò ổn định và chống nhiễu cho nguồn.

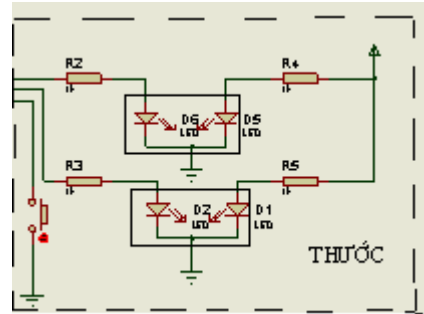
3.2.2 Hệ thống thước:

3.2.2.1. Đặc điểm của thước.

Thước bao gồm đầu đo có thể di động trên một thanh trượt có gắn một dải các vạch đen - trắng xác định tọa độ, vị trí a là các khe trong suốt – cho ánh sáng đi qua, còn vị trí b là các vạch đen – không cho ánh sáng truyền qua. Khi đầu đo dịch chuyển trên thanh trượt, bộ cảm biến gắn trong đầu đo sẽ phát hiện vị trí các vạch đen - trắng và hình thành tín hiệu cho bộ xử lý. Đầu đọc thu ánh sáng truyền qua từ đầu phát sẽ cho phép xác định số vạch khi dịch chuyển. Hình 3.4 thể hiện đặc điểm trên.



Hình 3.4. Thước đo tuyến tính kiểu vạch đen trắng



Hình 3.5. Cấu tạo bên trong của thước

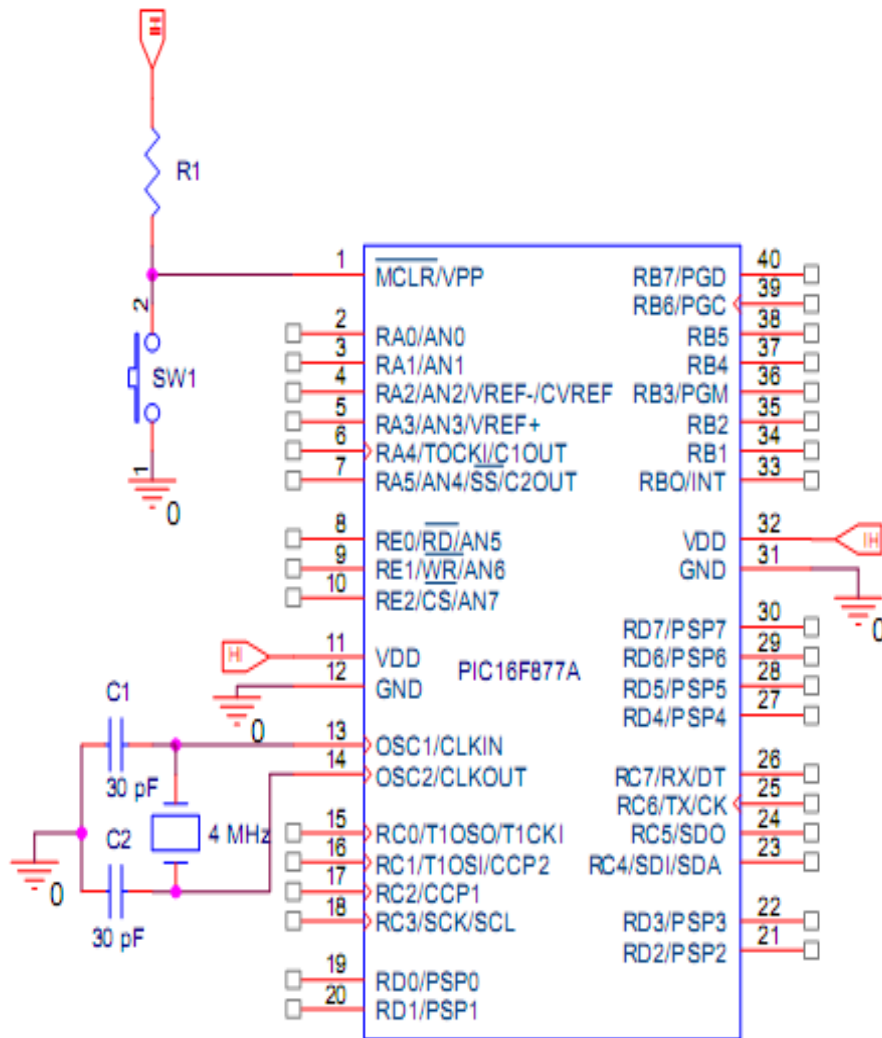
3.2.2.2. Nguyên tắc hoạt động:

Đầu đo có chứa hai bộ thu hồng ngoại độc lập A và B đặt cách nhau 1/4 chiều dài bước vạch. Đèn phát hồng ngoại và đèn thu đặt đối diện nhau ở hai băng vạch. Khi dịch tới vạch trong suốt, tín hiệu từ đèn phát tới được đèn thu và tạo mức ra - tín hiệu A (B) thấp (trạng thái 0). Ở vạch đen, tia hồng ngoại bị che chắn, tín hiệu ra là cao (trạng thái 1).

Khi dịch chuyển tiến, bộ thu A bị che chắn trước trong khi bộ thu B vẫn đang ở vị trí trắng. Khi dịch chuyển lùi, bộ thu B bị che chắn trước trong khi bộ thu A vẫn đang ở vị trí trắng.

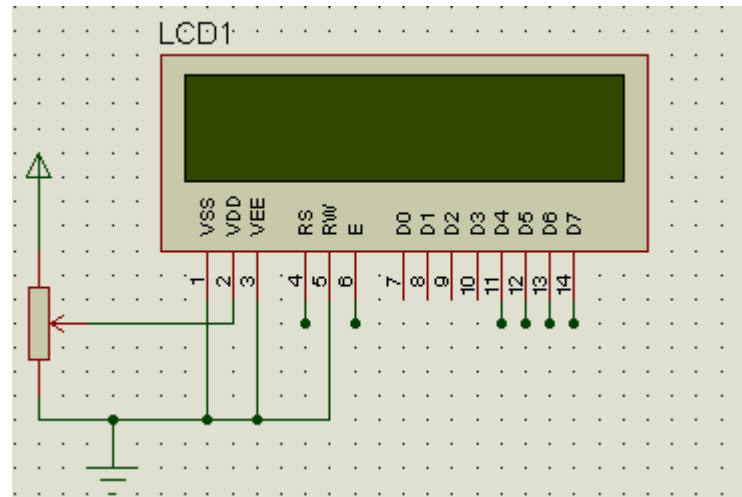
3.2.3. Bộ vi xử lý

Bộ xử lý làm nhiệm vụ nhận tín hiệu số từ rồi hiển thị số đếm trên màn hình LCD. Như đã phân tích trong chương 1, ở đây em sử dụng vi điều khiển PIC16F877A. Đây là vi điều khiển có 40 chân, với 5 cổng vào ra là Port A (RA0÷RA5), Port B (RB0÷RB7), Port C (RC0÷RC7), Port D (RD0÷RD7), Port E (RE0÷RE2). Nó có 8K Flash ROM và 368 Bytes RAM.



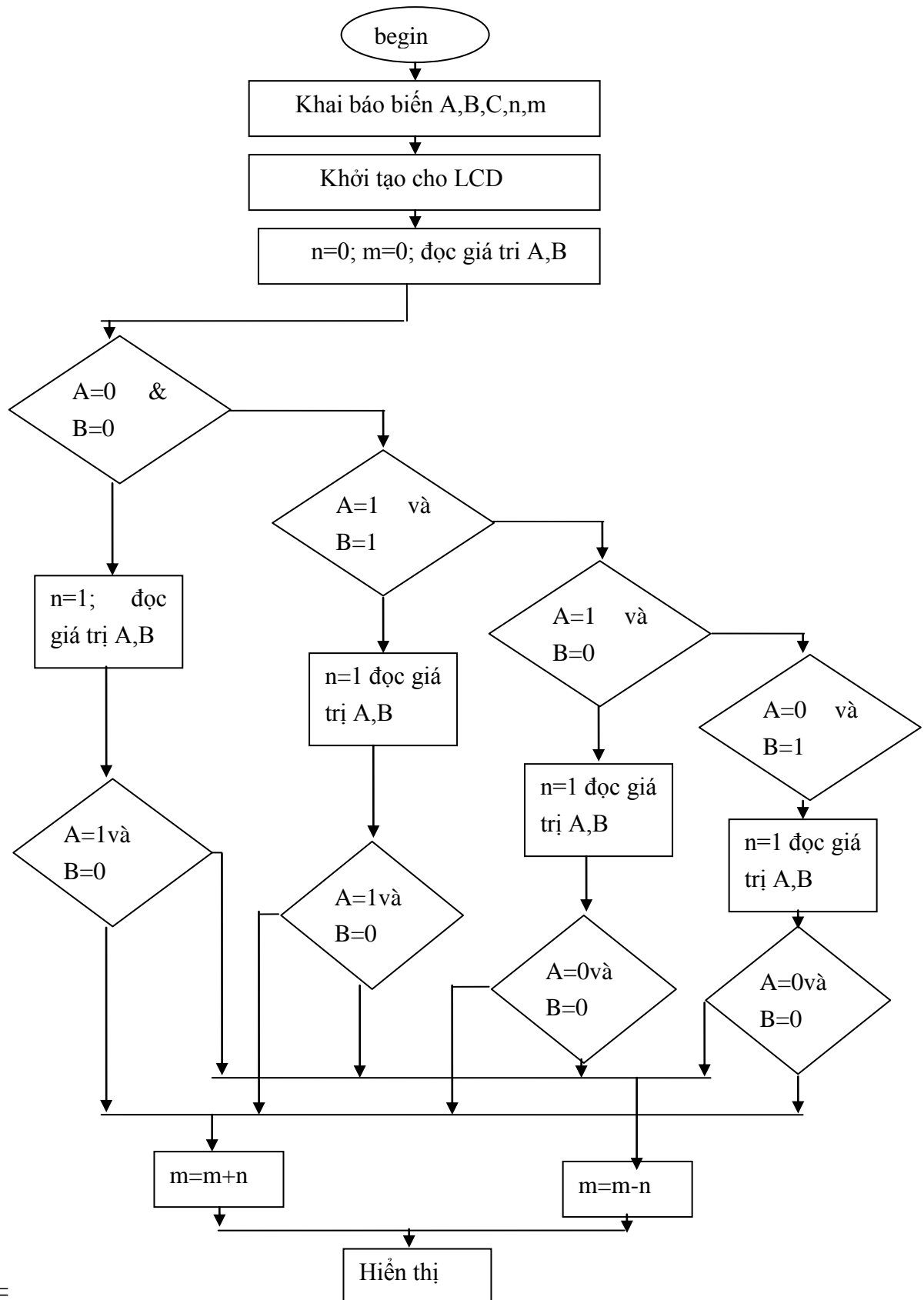
Hình 3.6. Sơ đồ nguyên lý của PIC16F877A trong mạch

3.2.4. Khởi hiển thị



Hình 3.8. Sơ đồ khởi hiển thị LCD

Khởi hiển thị LCD đã được giới thiệu ở chương 2. Trong chương này em sử dụng màn hình tinh thể lỏng LCD (Liquid Crystal Display) loại 2 dòng, 16 kí tự LCD1602. Màn hình LCD đã rất phổ biến trên thị trường và việc lập trình cho nó rất đơn giản thêm vào đó là nó có mặt thẩm mỹ rất cao. Sử dụng nguồn nuôi thấp (từ 2, 5 đến 5V). Có thể hoạt động ở hai chế độ 4 bit hoặc 8 bit (trong đề tài này em sử dụng chế độ 4 bit).



A=

3.5. Chương trình :

```

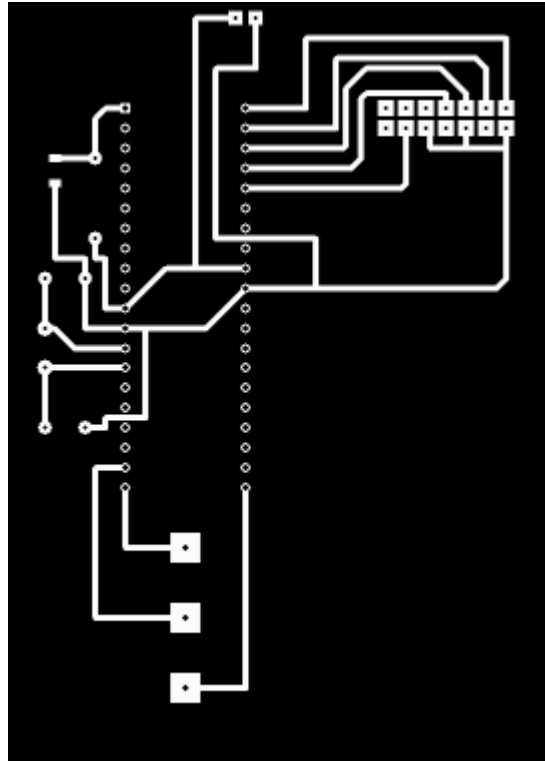
=====
// Ten chuong trinh      : Thiet ke he thong thước đo tuyến tính
// Phan mem              : Dung CCS
// Mo ta phan cung      : Dung pic16F877A - thach anh 10Mhz
//                        : LCD HD44780 giao tiep voi Port B qua cac chan
RB2-RB7
//                        : Cac chan RD0-RD2 la ngo vao
//                        : So dem duoc hien thi len LCD
=====

#include <16F877A.h>
#include <math.h>
#FUSES NOWDT, HS, NOPUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP, NOCPD,
NOWRT
#use delay(clock=20000000)
#include <lcd_lib_4bit.c>
int A,B,C,n,m;
void main()
{m=0;
  LCD_init();
  lcd_putcmd(0x80);
  printf(LCD_putchar,"thuoc do");
  lcd_putcmd(0xC0);
  printf(LCD_putchar,"tuyen tinh");
  LCD_putcmd(0x01);
  while(true)
  {n=0;
    A= input(pin_D0);
    B= input(pin_D1);
    C= input(pin_D2);
    if ((A==0) && (B==0))
    {n=1;
      while(true)
      { A= input(pin_D0);
        B= input(pin_D1);
        if ((A==1) && (B==0))
        {m=m+n;
          lcd_putcmd(0x80);
          printf(lcd_putchar,"dem tien");
          LCD_putcmd(0xC8);
          LCD_putchar(m +0x30);
          break;
        }
      }
    }
  }
}

```

```
    }
    if ((A==0) && (B==1))
    {
        m=m-n;
        lcd_putcmd(0x80);
        printf(lcd_putchar,"dem lui");
        LCD_putcmd(0xC8);
        LCD_putchar(m +0x30);
        break;}
    }
}
if ((A==1) && (B==1))
{
    n=1;
    while(true)
    {
        A= input(pin_D0);
        B= input(pin_D1);
        if ((A==1) && (B==0))
        {
            m=m-n;
            lcd_putcmd(0x80);
            printf(lcd_putchar,"dem lui");
            LCD_putcmd(0xC8);
            LCD_putchar(m +0x30);
            break;
        }
    }
    if ((A==0) && (B==1))
    {
        m=m+n;
        lcd_putcmd(0x80);
        printf(lcd_putchar,"dem tien");
        LCD_putcmd(0xC8);
        LCD_putchar(m +0x30);
        break;}
    }
}
}
```

3.6. Sơ đồ mạch in:



Kết luận

Sau thời gian nghiên cứu và làm đồ án, cùng với sự giúp đỡ tận tình của các thầy cô giáo và các bạn. Đặc biệt là thầy Nguyễn Văn Dương em đã hoàn thành nhiệm vụ đồ án của mình.

Qua đồ án em thấy được ứng dụng quan trọng của vi điều khiển trong đo lường và điều khiển, sử dụng vi điều khiển chúng ta thu thập được các đại lượng cần đo, xử lý các đại lượng đó và đưa ra kết quả mong muốn. Hiện nay vi điều khiển rất đa năng, nhỏ gọn, do đó áp dụng vi điều khiển vào trong cuộc sống là rất cần thiết.

Mặc dù rất cố gắng nhưng 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 góp ý, chỉ bảo của các thầy cô và các bạn để giúp em nâng cao kiến thức, chuyên môn phục vụ cho công việc sau này.

Em xin chân thành cảm ơn!

Tài liệu tham khảo

1. Nguyễn Tăng Cường, 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 Diên Tập, Vi Điều Khiển trong đo lường và điều khiển tự động, Nhà xuất bản Khoa Học và Kỹ Thuật, Hà Nội.
5. Họ Vi Điều Khiển 8051, Tống Văn ON, nhà Xuất bản Lao Động và Xã Hội.

Một số trang web tham khảo:

www.dientuvietnam.net

www.picvietnam.com

www.dientuvienthong.net

www.vagam.dieukhien.net

www.duyphi.phpnet.us/index.htm

