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

ISO 9001:2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: ĐIỆN TỬ VIỄN THÔNG

Người hướng dẫn: Thạc sỹ Nguyễn Văn Dương
Sinh viên : Đinh Việt Đức

HẢI PHÒNG - 2010

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

THIẾT KẾ BỘ ĐẾM TẦN SỐ

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

Người hướng dẫn: Thạc sỹ Nguyễn Văn Dương
Sinh viên : Đinh Việt Đức

Hải Phòng - 2010

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

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Đinh Việt Đức Mã số : 101368.

Lớp : ĐT1001 Ngành: Điện tử viễn thông.

Tên đề tài : Thiết kế bộ đếm tần số.

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).

……..
……..
……..
……..
……..
……..
……..
……..
……..
……..

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

……..
……..
……..
……..
……..
……..
……..
……..
……..
……..

3. Địa điểm thực tập tốt nghiệp.

Công ty Thông Tin Điện Tử Hàng Hải Việt Nam

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên : Nguyễn Văn Dương.

Học hàm, học vị : Thạc sỹ.

Cơ quan công tác : Trường Đại học Dân lập Hải Phòng.

Nội dung hướng dẫn

:.....

…….....……

…..

…….....

…..

…….....……

…..

…….....……

…..

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 ..... tháng ..... năm 2010.

Yêu cầu phải hoàn thành xong trước ngày ..... tháng ..... năm 2010.

Đã nhận nhiệm vụ ĐTTN
Sinh viên

Đã giao nhiệm vụ ĐTTN
Người hướng dẫn

Hải Phòng, ngày ..... tháng ..... năm 2010.

HIỆU TRƯỞNG

GS.TS.NGƯT 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:

……..

……..

……..

……..

……..

……..

……..

……..

……..

……..

……..

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

……..

……..

……..

……..

……..

……..

……..

……..

……..

3. Cho điểm của cán bộ hướng dẫn (ghi cả số và chữ) :

……..

……..

……..

Hải Phòng, ngày ..... tháng ..... năm 2010.

Cán bộ hướng dẫn

PHẦN NHẬN XÉT TÓM TẮT CỦA NGƯỜI CHẤM PHẢN BIỆN

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.

……..

……..
……..
……..
……..
……..
……..
……..
……..
……..

2. Cho điểm của cán bộ phản biện. (Điểm ghi cả số và chữ).

……..
……..
……..
……..
……..
……..
……..
……..
……..

Hải Phòng, ngày ..... tháng ..... năm 2010.

Người chấm phản biện

MỤC LỤC

LỜI NÓI ĐẦU

Thế kỉ XXI là thế kỉ của sự bùng nổ công nghệ thông tin và sự phát triển vượt bậc của các ngành khoa học kĩ thuật. Kĩ thuật điện tử là một trong những ngành kĩ thuật như thế. Sự phát triển của kĩ thuật điện tử gắn liền với sự phát triển của kĩ thuật vi điều khiển.

Ngày nay, kĩ thuật vi điều khiển được ứng dụng rộng rãi trong các lĩnh vực kĩ thuật và đời sống xã hội, đặc biệt trong kỹ thuật tự động hóa và điều khiển từ xa.

Hiện tại, vi điều khiển (VĐK) đã rất phổ biến ở Việt Nam và được ứng dụng rất nhiều. Tuy nhiên, để có thể tìm hiểu rõ hơn về vi điều khiển và tìm hiểu một ứng dụng cụ thể của nó em đã thực hiện đề tài "**THIẾT KẾ BỘ ĐẾM TẦN SỐ** ".

Đồ án của em gồm 3 chương:

Chương 1. Tổng quan về VĐK và hiển thị LCD.

Chương 2. Thiết kế bộ đếm tần số.

Chương 3. Phần mềm điều khiển.

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 nhận được sự đóng góp ý kiến của thầy cô trong hội đồng và các bạn để đồ án tốt nghiệp của em được hoàn thiện hơn.

Chương 1

TỔNG QUAN VỀ VĐK VÀ HIỂN THỊ LCD

1.1. VI ĐIỀU KHIỂN

Thông thường có 4 họ vi điều khiển 8 bit chính là 6811 của Motorola, 8051 của Intel, Z8 của Xilog và Pic 16 của Microchip Technology. Mỗi một loại trên đây đều có một tập lệnh và thanh ghi riêng duy nhất, nên chúng thường không tương thích lẫn nhau. Ngoài ra cũng có những bộ vi điều khiển 16 bits và 32 bits được sản xuất bởi các hãng khác nhau. Với tất cả những bộ vi điều khiển khác nhau thì tiêu chuẩn để lựa chọn là:

*) Đáp ứng được nhu cầu tính toán của bài toán một cách hiệu quả, đầy đủ chức năng cần thiết và thấp nhất về mặt giá thành. 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 bits, 16 bits hay 32 bits có thể đáp ứng tốt nhất nhu cầu của bài toán một cách hiệu quả. Những tiêu chuẩn đó là:

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

- Kiểu đóng vỏ: Đóng vỏ kiểu DIP 40 chân hay QFP. Đây là yêu cầu quan trọng xét về không gian, kiểu lắp 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 nâng cấp cho hiệu suất cao hoặc giảm công suất tiêu thụ.

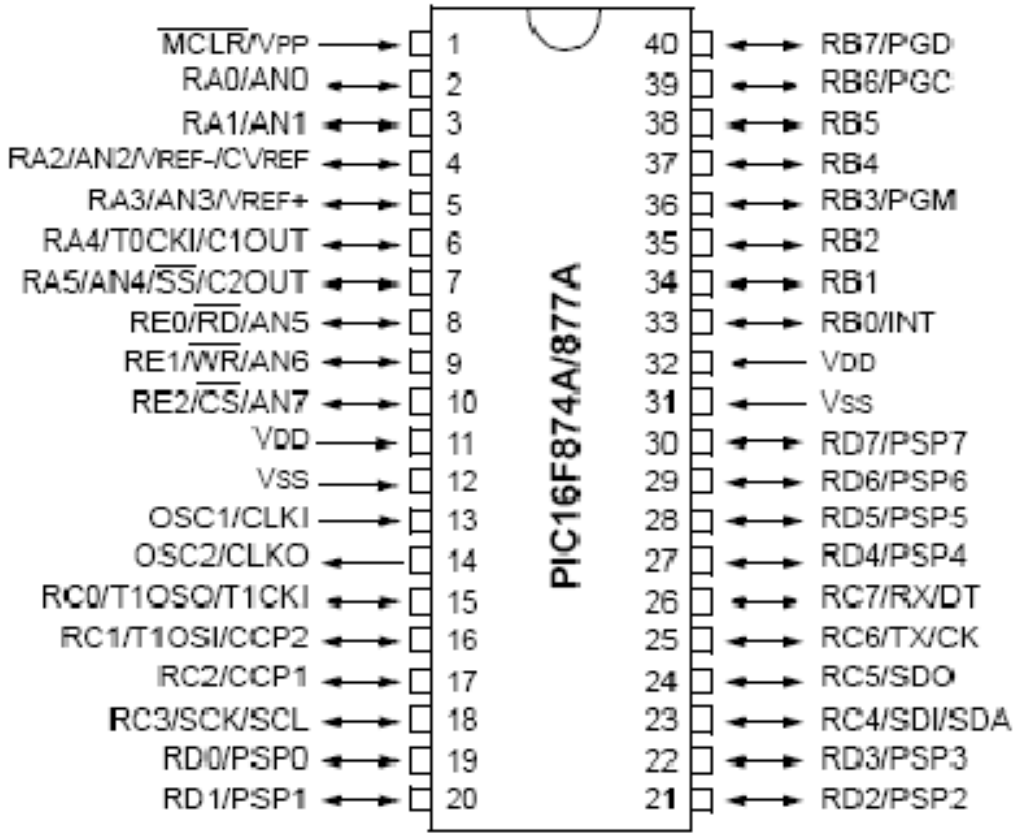
- Giá thành cho một đơn vị: Điều này quan trọng quyết định giá thành sản phẩm mà một bộ vi điều khiển được sử dụng.

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

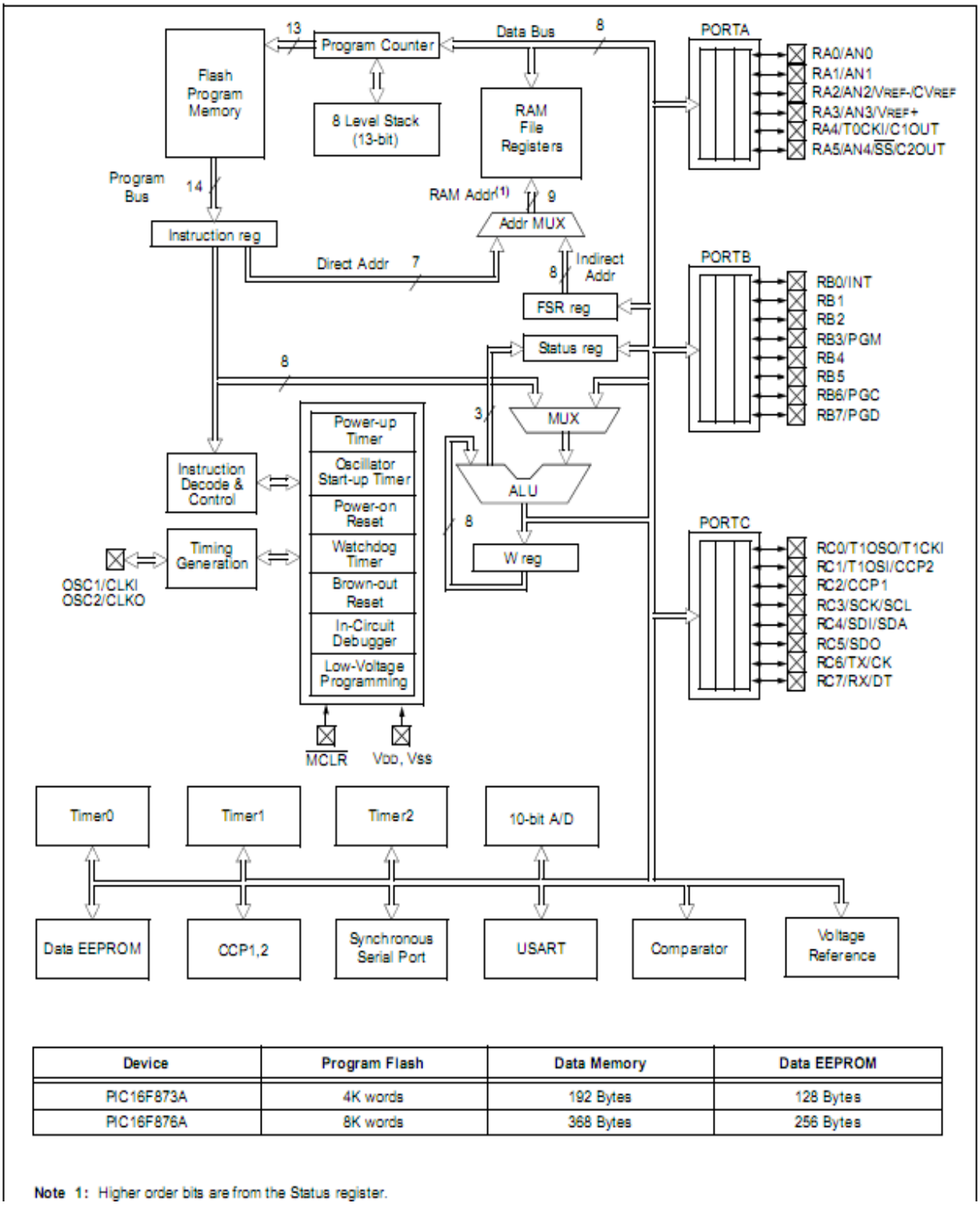
*) Nguồn các bộ vi điều khiển sẵn có nhiều và tin cậy. Khả năng sẵn sàng đáp ứng về số lượng trong hiện tại tương lai.

Hiện nay các bộ vi điều khiển 8 bits họ 8051 là có số lượng lớn nhất các nhà cung cấp đa dạng như Intel, Atmel, Philip… Nhưng về mặt tính năng và công năng thì có thể xem PIC vượt trội hơn rất nhiều so với 89 với nhiều module được tích hợp sẵn như ADC 10bits, PWM 10bits, PROM 256 Bytes, COMPARATER, VERF COMPARATER, một đặc điểm nữa là tất cả các vi điều khiển PIC sử dụng thì đều có chuẩn PI tức chuẩn công nghiệp thay vì chuẩn PC (chuẩn dân dụng). Ngoài ra PIC còn được rất nhiều nhà sản xuất phần mềm tạo ra các ngôn ngữ hỗ trợ cho việc lập trình ngoài ngôn ngữ Asembly ra còn có thể sử dụng ngôn ngữ C thì sử dụng CCSC, HTPIC hay sử dụng Basic thì có MirkoBasic… và còn nhiều chương trình khác nữa để hỗ trợ cho việc lập trình bên cạnh ngôn ngữ kinh điển là Asembler. Nên trong đề tài này tôi lựa chọn sử dụng vi điều khiển PIC làm bộ điều khiển chính, và ở đây là PIC16F877A.

1.1.1. Sơ đồ khối và bảng mô tả chức năng các chân của PIC16F877A



Hình 1.1. PIC 16F877A



Device	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the Status register.

Hình 1.2. Sơ đồ khối của PIC16F877A

Bảng mô tả chức năng các chân của PIC16F877A

Pin Name	DIP Pin#	PLCC Pin#	QFT Pin#	I/O/ P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	1	ST/CMOS (4)	Đầu vào của xung dao động thạch anh/ngõ vào xung clock ngoại
OSC2/CLKOUT	1	2	18	O	-	Đầu ra của xung 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.
$\overline{MCLR}/V_{pp}$	1	2	18	I/P	ST	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	3	19	I/O	TTL	PORTA là port vào ra hai chiều. RA0 có thể làm ngõ vào tương tự

						thứ 0.
RA1/AN1	3	4	20	I/O	TTL	RA1 có thể làm ngõ vào tương tự thứ 1
RA2/AN2/VREF –	4	5	21	I/O	TTL	RA2 có thể làm ngõ vào tương tự 2 hoặc điện áp chuẩn tương tự âm.
RA3/AN3/VREF +	5	6	22	I/O	TTL	RA3 có thể làm ngõ vào tương tự 3 hoặc điện áp chuẩn tương tự dương.
RA4/T0CKI	6	7	23	I/O	ST	RA4 có thể làm ngõ vào xung clock cho bộ định thời Timer0.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 có thể làm ngõ vào tương tự thứ 4
RB0/INT RB1 RB2	33 34 35	36 37 38	8 9 10	I/O I/O I/O	TTL/ST(1) TTL TTL	PORTB là port hai chiều. RB0 có thể làm chân ngắt ngoài
RB3/PGM	36	39	11	I/O	TTL	RB3 có thể làm ngõ vào của điện thế được lập trình ở mức thấp.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.

RB5 RB6/PGC RB7/PGD	38 39 40	42 43 44	15 16 17	I/O I/O I/O	TTL TTL/ST(2) TTL/ST(3)	Interrupt-on-change pin. Interrupt-on-change pin hoặc In-Crcuit Debugger pin . Serial programming clock. Interrupt-on-change pin hoặc In-Crcuit Debugger pin Serial programming data
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC là port vào ra hai chiều. RC0 có thể là ngõ vào của bộ dao động Timer1 hoặc ngõ xung clock cho Timer1
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 có thể là ngõ vào của bộ dao động Timer1 hoặc ngõ vào Capture2/ngõ ra compare2/ngõ vào

						PWM2.
RC2/CCP1	17	19	36	I/O	ST	RC2 có thể ngõ vào capture1/ngõ ra compare1/ngõ vào PWM1
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 có thể là ngõ vào xung
RC4/SDI/SDA	23	25	42	I/O	ST	Clock đồng bộ nội tiếp/ngõ ra trong cả hai chế độ SPI và I2C RC4 có thể là dữ liệu bên trong SPI(chế độ SPI) hoặc dữ liệu I/O(chế độ I^2C).
RC5/SDO	24	26	43	I/O	ST	RC5 có thể là dữ liệu ngoài SPI(chế độ SPI)
RC6/TX/CK	25	27	44	I/O	ST	RC6 có thể là chân truyền không đồng bộ USART hoặc đồng bộ với xung đồng hồ
RC7/RX/DT	26	29	1	I/O	ST	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	21 22 23 24 30 31 32 33	38 39 40 41 2 3 4 5	I/O I/O I/O I/O I/O I/O I/O I/O	ST/TTL(3) ST/TTL(3) ST/TTL(3) ST/TTL(3) ST/TTL(3) ST/TTL(3) ST/TTL(3) ST/TTL(3)	PORTD là port vào ra hai chiều hoặc là parallel slave port khi giao tiếp với bus của bộ vi xử lý.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	PORTE là port vào ra hai chiều. RE0 có thể điều khiển việc đọc parrallel slave port hoặc là ngoc vào tương tự thứ 5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 có thể điều khiển việc ghi parallel slave port hoặc là ngõ vào tương tự thứ 6.
RE2/$\overline{CS}$ /AN7	10	11	27	I/O	ST/TTL(3)	RE2 có thể điều khiển việc chọn parallel slave port hoặc là ngõ vào tương tự thứ 7

V_{ss} V_{DD}	12, 31 11, 32	13, 34 12, 35	7, 28 6, 29	P P		Cung cấp nguồn dương cho các mức logicvà những chân I/O.
NC		1, 17, 28, 40	12,13 33, 4			Những chân này không được nối bên trong và nó được để trống

Ghi chú: I = input; O = output; I/O = input/output; P = power

- = Not used; TTL = TTL input; ST = Schmitt Trigger input

1. Là vùng đệm có ngõ vào Trigger Schmitt khi được cấu hình như ngắt ngoài.

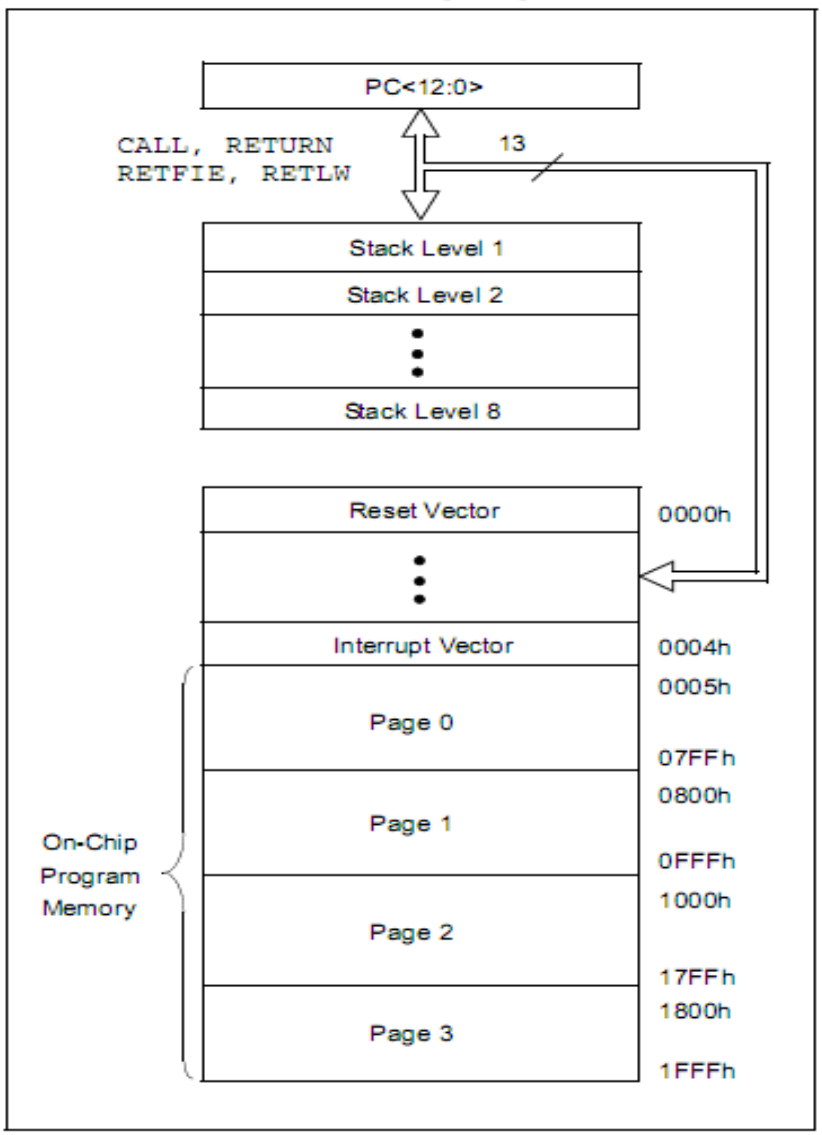
2. Là vùng đệm có ngõ vào Trigger Schmitt khi được sử dụng trong chế độ 9 Serial Programming.

3. Là vùng đệm có ngõ vào Trigger Schmitt khi được cấu hình như ngõ vào ra mục đích chung và là ngõ vào TTL khi sử dụng trong chế độ Parallel Slave Port (cho việc giao tiếp với các bus của bộ vi xử lý).

4. Là vùng đệm có ngõ vào Trigger Schmitt khi được cấu hình trong chế độ dao động RC và một ngõ vào CMOS khác.

1.1.2. Tổ chức bộ nhớ

Có 2 khối bộ nhớ trong các vi điều khiển họ PIC16F87X, bộ nhớ chương trình và bộ nhớ dữ liệu, với những bus riêng biệt để có thể truy cập đồng thời.



Hình 1.3. Ngăn xếp và bản đồ bộ nhớ chương trình PIC16F877A

1.1.2.1. Tổ chức của bộ nhớ chương trình

Các vi điều khiển họ PIC16F877A có bộ đếm chương trình 13 bits có khả năng định vị không gian bộ nhớ chương trình lên đến 8Kb. Các IC PIC16F877A có 8Kb bộ nhớ chương trình FLASH, các IC PIC16F873/874 chỉ có 4 Kb. Vectơ RESET đặt tại địa chỉ 0000h và vectơ ngắt tại địa chỉ 0004h.

1.1.2.2. Tổ chức bộ nhớ dữ liệu

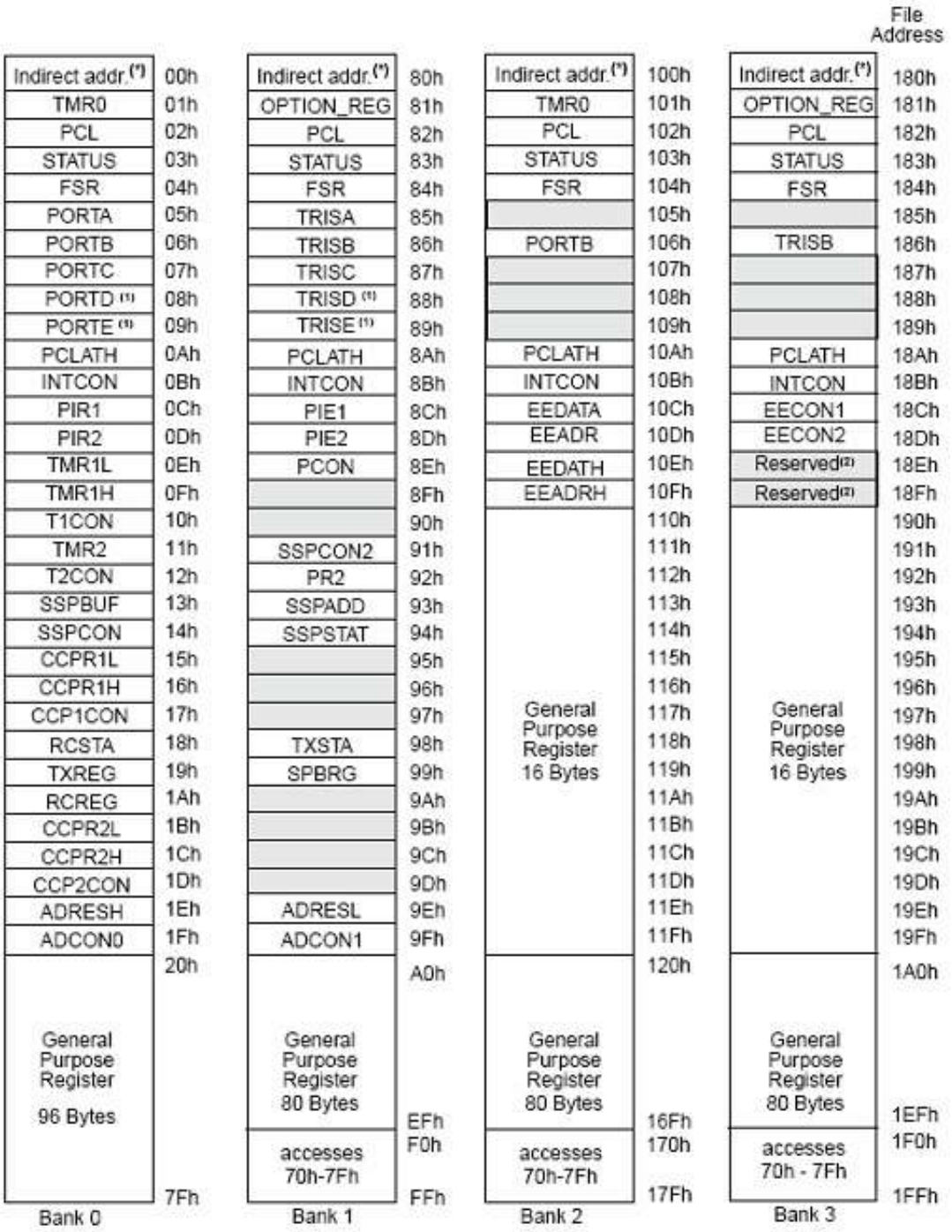
Bộ nhớ dữ liệu được chia thành nhiều dãy và chứa các thanh ghi mục đích chung và các thanh ghi chức năng đặc biệt. Bit RP1 (STATUS <6>) và RP0 (STATUS <5>) là những bits dùng để chọn các dãy thanh ghi.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Chiều dài của mỗi dãy là 7Fh (128 bytes). Phần thấp của mỗi dãy dùng để chứa các thanh ghi chức năng đặc biệt. Trên các thanh ghi chức năng đặc biệt là các thanh ghi mục đích chung, có chức năng như RAM tĩnh. Thường thì những thanh ghi đặc biệt được sử dụng từ một dãy và có thể được ánh xạ vào những dãy khác để giảm bớt đoạn mã và khả năng truy cập nhanh hơn.

1.1.2.3. Các thanh ghi mục đích chung

Các thanh ghi này có thể truy cập trực tiếp hoặc gián tiếp thông qua thanh ghi FSG (File Select Register).



Hình 1.4. Các thanh ghi của PIC16F877A

1.1.2.4. Các thanh ghi chức năng đặc biệt

Các thanh ghi chức năng đặc biệt (Special Function Resgister) được sử dụng bởi CPU và các bộ nhớ ngoại vi để điều khiển các hoạt động được yêu cầu của thiết bị. Những thanh ghi này có chức năng như RAM tĩnh. Danh sách những

thanh ghi nay được trình bày ở bảng dưới. Các thanh ghi chức năng đặc biệt có thể chia thành hai loại: phần trung tâm (CPU) và phần ngoại vi.

1.1.2.5. Các thanh ghi trạng thái

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h-1FFh)
10 = Bank 2 (100h-17Fh)
01 = Bank 1 (80h-FFh)
00 = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

bit 4 $\overline{TO}$: Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 $\overline{PD}$: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
(for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

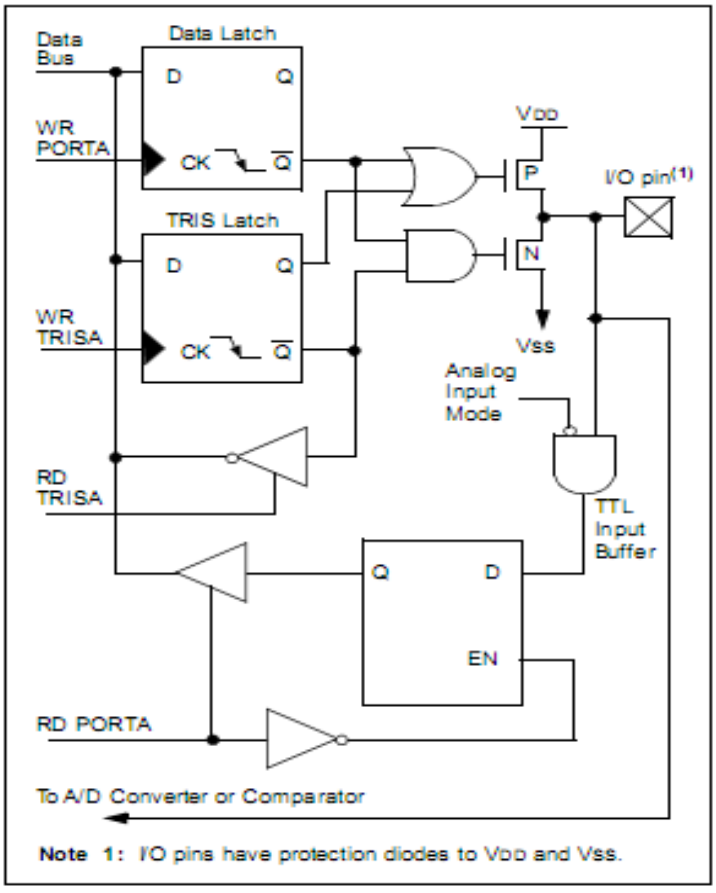
Hình 1.5. Thanh ghi trạng thái (địa chỉ 03h, 83h, 103h, 183h)

Thanh ghi trạng thái chứa các trạng thái số học của bộ ALU, trạng thái RESET và những bits chọn dãy thanh ghi cho bộ nhớ dữ liệu. Thanh ghi trạng thái có thể là đích cho bất kì lệnh nào, giống như những thanh ghi khác. Nếu thanh ghi trang thái là đích cho một lệnh mà ảnh hưởng đến các cờ Z, DC hoặc C, và sau đó những bit này sẽ được vô hiệu hoá. Những bit này có thể đặt hoặc xóa tuỳ theo trạng thái logic của thiết bị. Hơn nữa hai bit $\overline{TO}$ và $\overline{PD}$ thì không cho phép ghi, vì vậy kết quả của một tập lệnh mà thanh ghi trạng thái là đích có thể khác hơn dự định. Ví dụ, CLRF STATUS sẽ xóa 3 bit cao nhất và đặt bit Z.

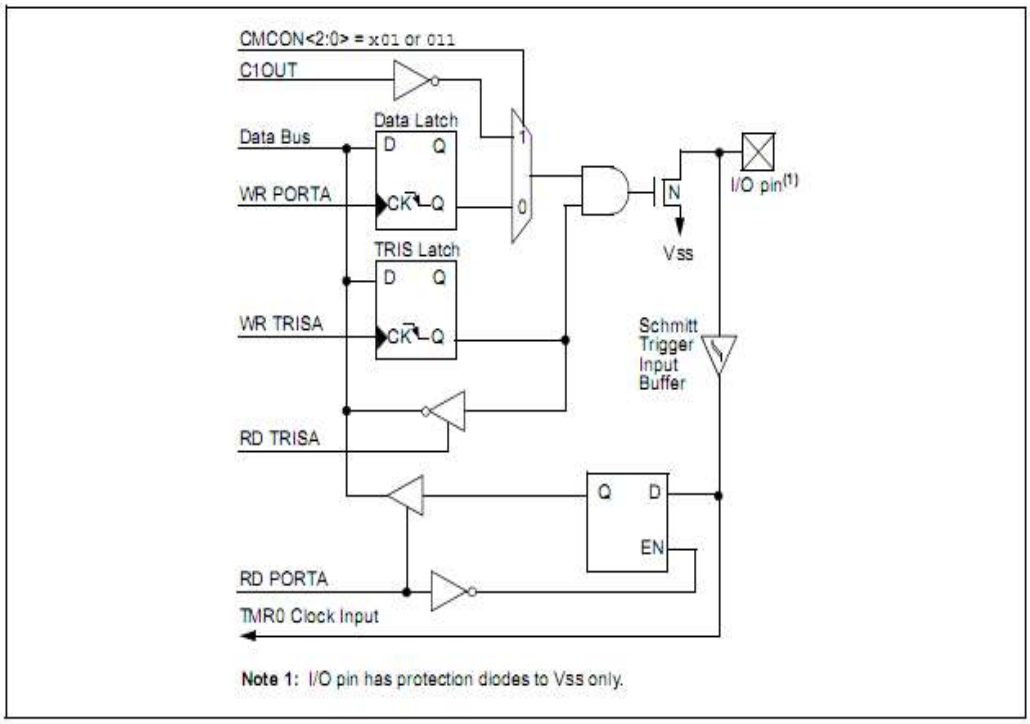
Lúc này các bits của thanh ghi trạng thái là 000u u1uu (u = unchanged). Chỉ có các lệnh BCF, BSF, SWAPF và MOVWF được sử dụng để thay đổi thanh ghi trạng thái, bởi vì những lệnh này không làm ảnh hưởng đến các bit Z, DC hoặc C từ thanh ghi trạng thái. Đối với những lệnh khác thì không ảnh hưởng đến những bits trạng thái này.

1.1.3. Các cổng của PIC 16F877A

1.1.3.1. PORTA và thanh ghi TRISA



Hình 1.6. Sơ đồ khối của chân RA3:RA0 và RA5



Hình 1.7. Sơ đồ khối của chân RA4/T0CKI

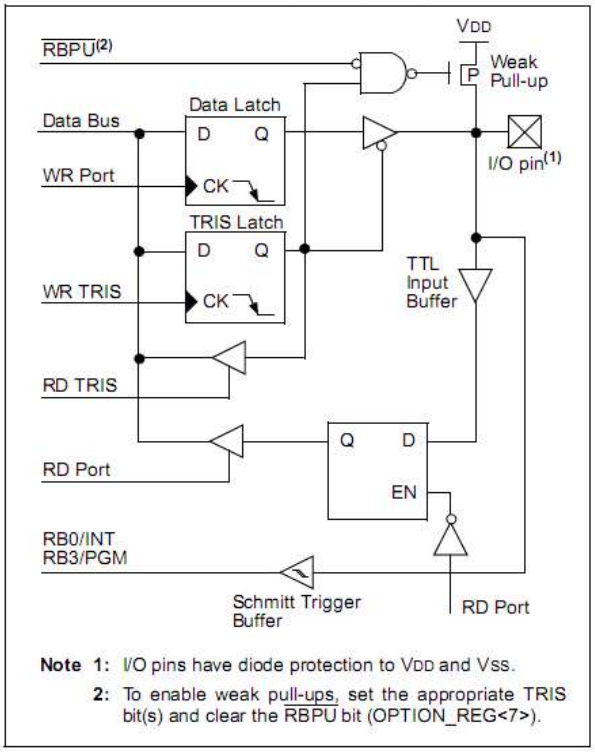
1.1.3.2. PORTB và thanh ghi TRISB

PORTB có độ rộng 8 bits, là port vào ra hai chiều. Ba chân của PORTB được đa hợp với chức năng lập trình mức điện thế thấp (Low Voltage Programming): RB3/PGM, RB6/PGC và RB7/PGD. Mỗi chân của PORTB có một điện trở kéo bên trong. Một bit điều khiển có thể mở tất cả những điện trở kéo này lên. Điều này được thực hiện bằng cách xoá bit $\overline{RBPU}$ (OPTION_REG<7>). Những điện trở này bị cấm khi có một Power-on Reset. Bốn chân của PORTB: RB7 đến RB4 có một ngắt để thay đổi đặc tính. Chỉ những chân được cấu hình như ngõ vào mới có thể gây ra ngắt này. Những chân vào (RB7:RB4) được so sánh với giá trị được chốt trước đó trong lần đọc cuối cùng của PORTB. Các kết quả không phù hợp ở ngõ ra trên chân RB7:RB4 được kết hợp hoặc với nhau để phát ra một ngắt Port thay đổi RB với cờ ngắt là RBIF (INTCON<0>). Ngắt này có thể đánh thức thiết bị từ trạng thái nghỉ

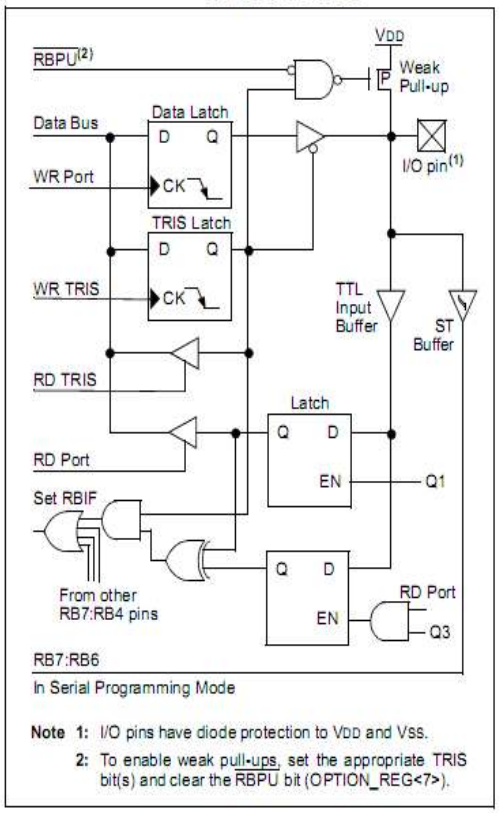
(SLEEP). Trong thủ tục phục vụ ngắt người sử dụng có thể xoá ngắt theo cách sau:

a) Đọc hoặc ghi bất kì lên PORTB. Điều này sẽ kết thúc điều kiện không hoà hợp.

b) Xoá bít cờ RBIF.



Hình 1.8. Sơ đồ chân RB3:RB0

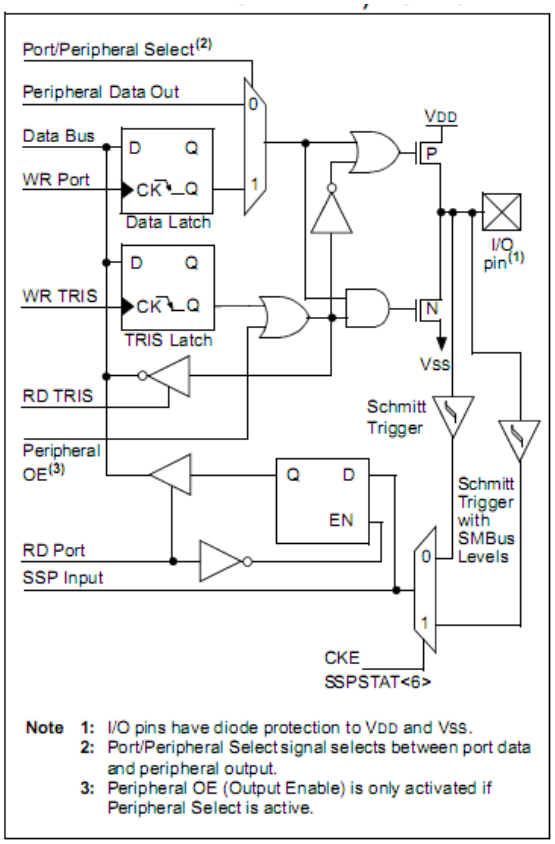


Hình 1.9. Sơ đồ chân RB7:RB4

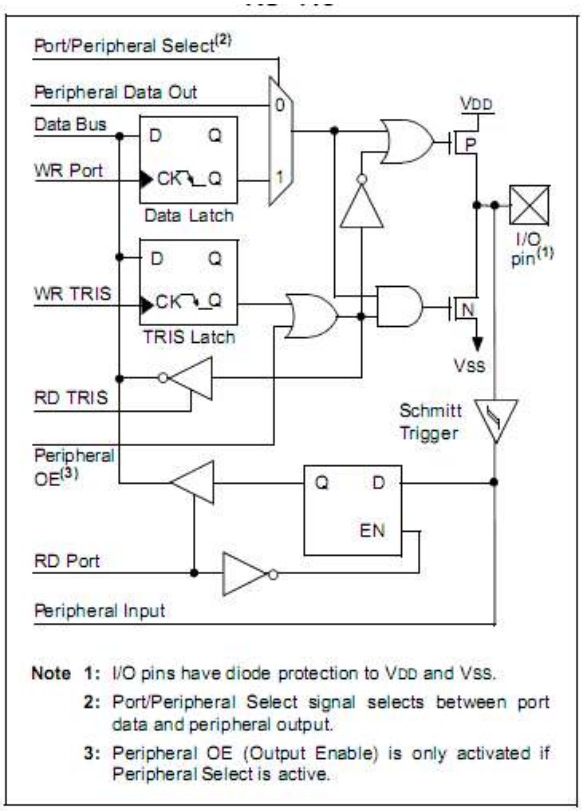
1.1.3.3. PORTC và thanh ghi TRISC

PORTC có độ rộng là 8 bits, là Port hai chiều. Thanh ghi dữ liệu trực tiếp tương ứng là TRISC. Cho tất cả các bit của TRISC là 1 thì các chân tương ứng ở PORTC là ngõ vào. Cho tất cả các bit của TRISC là 0 thì các chân tương ứng ở PORTC là ngõ ra. PORTC được đa hợp với vài chức năng ngoại vi, những chân của PORTC có đệm Trigger Schmitt ở ngõ vào. Khi bộ I2C được cho phép, chân 3 và 4 của PORTC có thể cấu hình với mức I2C bình thường, hoặc với mức

SMBus bằng cách sử dụng bit CKE (SSPSTAT<6>). Khi những chức năng ngoại vi được cho phép, chúng ta cần phải quan tâm đến việc định nghĩa các bits của TRIS cho mỗi chân của PORTC. Một vài thiết bị ngoại vi ghi đè lên bit TRIS thì tạo nên một chân ở ngõ ra, trong khi những thiết bị ngoại vi khác ghi đè lên bit TRIS thì sẽ tạo nên một chân ở ngõ vào. Khi những bit TRIS ghi đè bị tác động trong khi thiết bị ngoại vi được cho phép, những lệnh đọc thay thế ghi (BSF, BCF, XORWF) với TRISC là nơi đến cần phải được tránh. Người sử dụng cần phải chỉ ra vùng ngoại vi tương ứng để đảm bảo cho việc đặt TRIS bit là đúng.



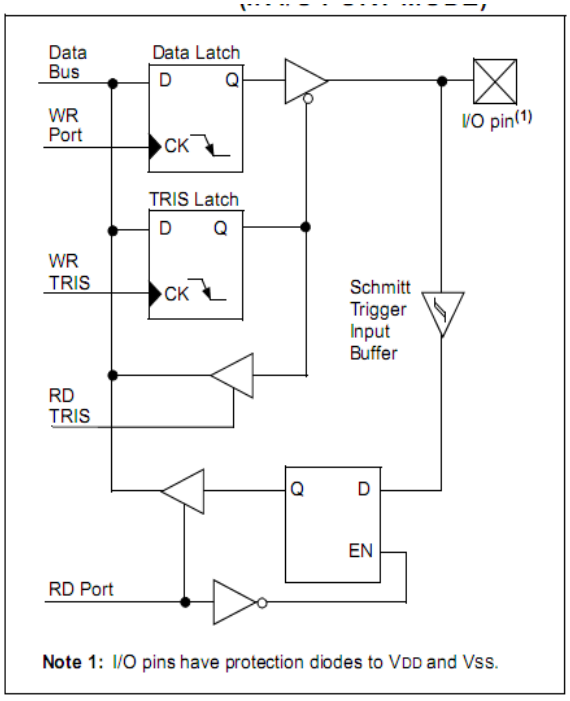
Hình 1.10. Sơ đồ chân RC4:RC3 RC7:RC5



Hình 1.11. Sơ đồ chân RC2:RC0,

1.1.3.4. PORTD và thanh ghi TRISD

PORTD là port 8 bits với đệm Trigger Schmitt ở ngõ vào. Mỗi chân có thể được cấu hình riêng lẻ như một ngõ vào hoặc ngõ ra. PORTD có thể được cấu hình như Port của bộ vi xử lý rộng 8 bits (parallel slave port) bằng cách đặt bit điều khiển PSPMIDE (TRISE <4>). Trong chế độ này, đệm ở ngõ vào là TTL.

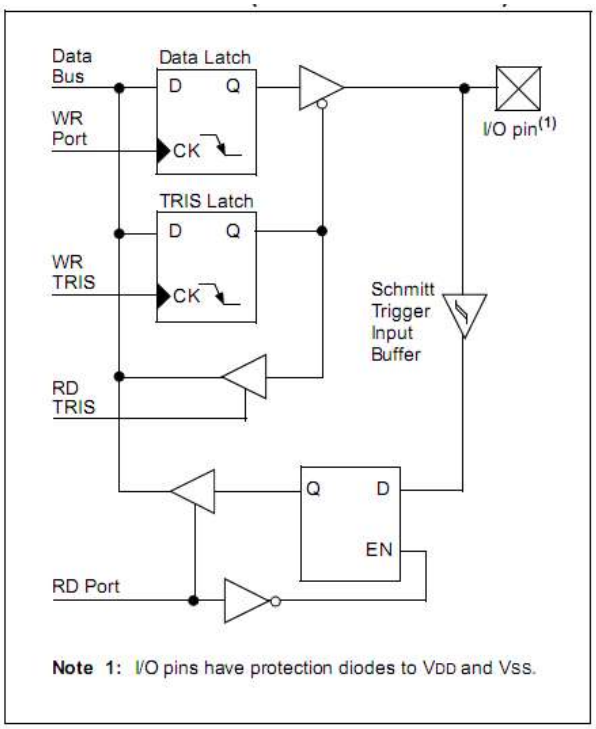


Hình 1.12. Sơ đồ khối của PORTD (trong chế độ là port I/O)

1.1.3.5. PORTE và thanh ghi TRISE

PORTE có ba chân (RE0/RD/AN5, RE1/WR/AN6, và RE2/CS/AN7) mỗi chân được cấu hình riêng lẻ như những ngõ vào hoặc những ngõ ra. Những chân này có đệm Trigger Schmitt ở ngõ vào. Những chân của PORTE đóng vai trò như những ngõ vào điều khiển vào ra cho Port của vi xử lý khi bit PSPMODE (TRISE <4>) được đặt. Trong chế độ này, người sử dụng cần phải chắc chắn rằng những bit TRISE <2:0> được đặt, và chắc rằng những chân này được cấu hình như những ngõ vào số. Cũng bảo đảm rằng ADCON1 được cấu hình cho vào ra số. Trong chế độ này, những đệm ở ngõ vào là TTL.

Những chân của PORTE được đa hợp với những ngõ vào tương tư, Khi được chọn cho ngõ vào tương tự, những chân này sẽ đọc giá trị "0". TRISE điều khiển hướng của những chân RE chỉ khi những chân này được sử dụng như những ngõ vào tương tự. Người sử dụng cần phải giữ những chân được cấu hình như những ngõ vào khi sử dụng chúng như những ngõ vào tương tự.



Hình 1.13. Sơ đồ khối của PORTE (trong chế độ I/O port)

1.1.4. Hoạt động của định thời

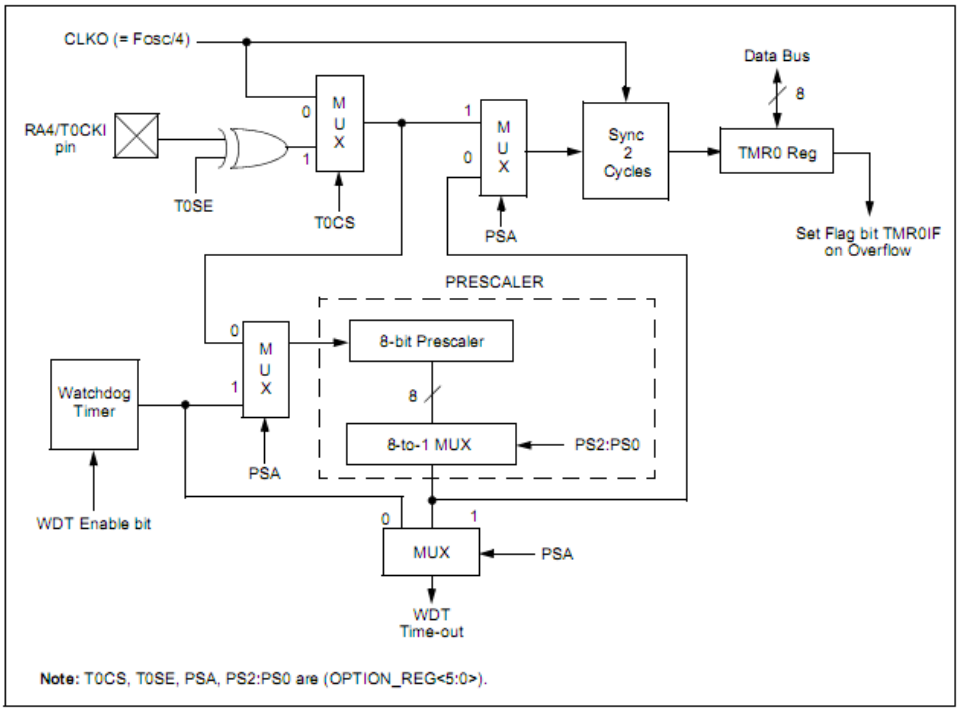
1.1.4.1. Bộ định thời TIMER0

Bộ định thời/bộ đếm Timer0 có các đặc tính sau:

- Bộ định thời/bộ đếm 8 bits
- Cho phép đọc và ghi
- Bộ chia 8 bits lập trình được bằng phần mềm
- Chọn xung clock nội hoặc ngoại
- Ngắt khi có sự tràn từ FFh đến 00h

- Chọn sườn cho xung clock ngoài

Sơ đồ khối của bộ định thời Timer0 và bộ chia dùng chung với WDT được đưa ra trong hình 1.14.



Hình 1.14. Sơ đồ bộ định thời Timer0 và bộ chia dùng chung với WDT

Chế độ định thời (Timer) được chọn bằng cách xoá bít T0CS (OPTION_REG<5>). Trong chế độ định thời, bộ định thời Timer0 sẽ tăng dần sau mỗi chu kì lệnh (không có bộ chia). Nếu thanh ghi TmR0 được ghi thì sự tăng sẽ bị ngăn lại sau hai chu kì lệnh.

Chế độ đếm (Counter) được chọn bằng cách xoá bit T0CS (OPTION_REG<5>). Trong chế độ đếm, Timer0 sẽ tăng dần ở mỗi cạnh lên xuống của chân RA4/T0CKI. Sự tăng sườn được xác định bởi bit Timer0 Source Edge Select, T0SE (OPTION_RE<4>). Bộ chia chỉ được dùng chung qua lại giữa bộ định thời Timer0 và bộ định thời Watchdog. Bộ chia không cho phép đọc hoặc ghi

Ngắt Timer0

Ngắt TMR0 được phát ra khi thanh ghi TMR0 tràn từ FFh đến 00h. Sự tràn này sẽ đặt bít T0IF (INTCON<2>). Ngắt này có thể được giấu đi bằng cách xóa bít T0IE (INTCON<5>). Bít T0IF cần phải được xóa trong chương trình bởi thủ tục phục vụ ngắt của bộ định thời Timer0 trước khi ngắt này được cho phép lại.

Sử dụng Timer0 với xung clock ngoại

Khi bộ chia không được sử dụng, clock ngoài đặt vào thì giống như bộ chia ở ngõ ra. Sự đồng bộ của chân T0CKI với clock ngoài được thực hiện bằng cách lấy mẫu bộ chia ở ngõ ra trên chân Q2 và Q4. Vì vậy thực sự cần thiết để chân T0CKI ở mức cao trong ít nhất 2 chu kỳ máy và ở mức thấp trong ít nhất 2 chu kỳ máy.

Bộ chia

Thiết bị PIC16F87X chỉ có một bộ chia mà được dùng chung bởi bộ định thời TIMER0 và bộ định thời Watchdog. Bộ chia có các 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.

1.1.4.2. Bộ định thời TIMER1

Bộ định thời TIMER1 là một bộ định thời/bộ đếm 16 bit gồm hai thanh ghi TMR1H (Byte cao) và TMR1L (byte thấp) mà có thể đọc hoặc ghi. Cặp thanh ghi này tăng số đếm từ 0000h đến FFFFh và báo tràn sẽ xuất hiện khi có sự chuyển số đếm từ FFFFh xuống 0000h. Ngắt, nếu được phép có thể phát ra khi có số đếm tràn và được đặt ở bit cờ ngắt TMR1IF. Ngắt có thể được phép hoặc cấm bằng cách đặt hoặc xóa bit cho phép ngắt TMR1IE.

Bộ định thời Timer1 có thể được cấu hình để hoạt động một trong hai chế độ sau:

- Định thời một khoảng thời gian (timer)
- Đếm sự kiện (Counter)

Việc lựa chọn một trong hai chế độ được xác định bằng cách đặt hoặc xóa bít điều khiển TMR1ON.

----	----	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

Bit7 Bit0

Bit 7, 6: Không được định nghĩa

Bit 5, 4: Bit chọn bộ chia clock cho timer1

Bit 3: Bit điều khiển cho phép bộ dao động Timer1

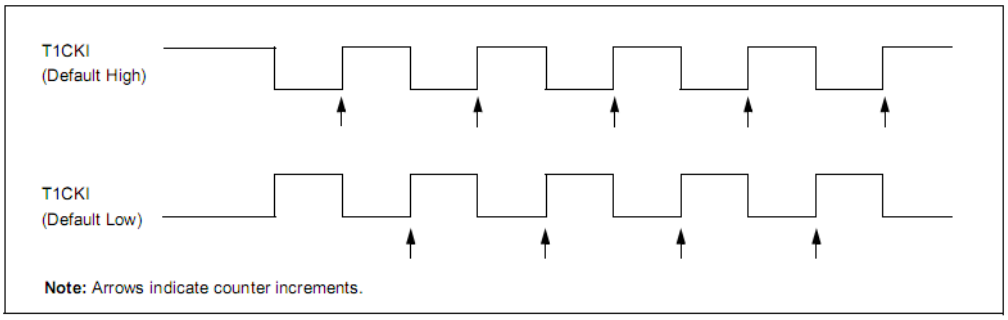
Bit 2: Bit điều khiển clock ngoài Timer

Bit 1: Bit chọn nguồn clock cho Timer1

Bit 0: Bit điều khiển hoạt động của Timer1

Chế độ Timer

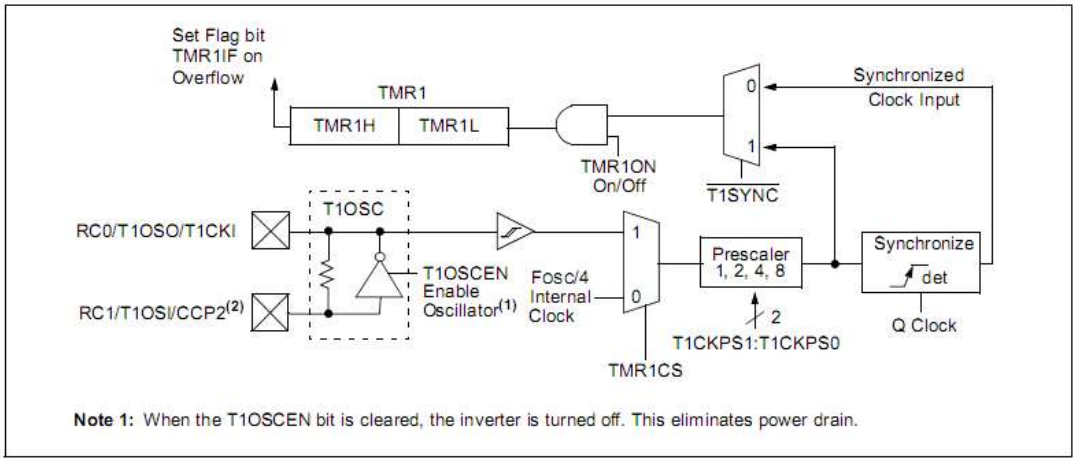
Chế độ Timer được chọn bằng cách xóa TMR1CS. Trong chế độ này, Nguồn clock đặt vào Timer là mạch dao động $F_{OSC}/4$. Bit điều khiển đồng bộ không bị tác động vì clock ngoài luôn luôn đồng bộ.



Hình 1.15. Sườn tăng timer1

Chế độ counter

Trong chế độ này, bộ định thời tăng số đếm qua clock ngoài. Việc tăng xảy ra sau mỗi sườn lên của xung clock ngoài. Bộ định thời phải có một sườn lên trước khi việc đếm bắt đầu.



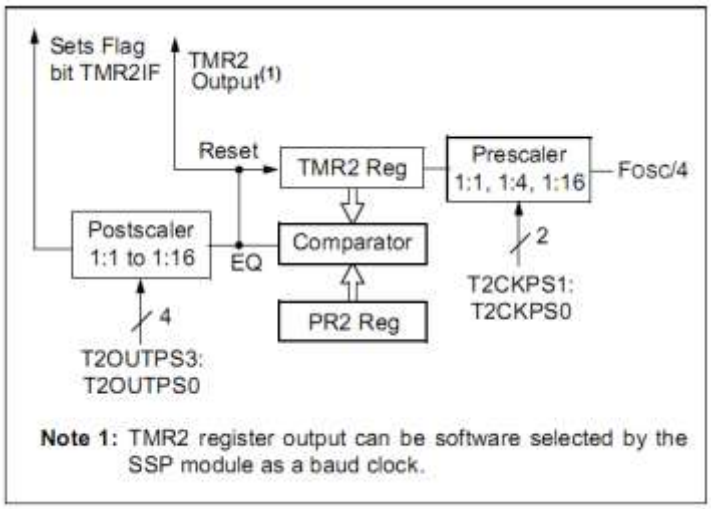
Hình 1.16. Sơ đồ khối bộ định thời timer1

1.1.4.3. Bộ định thời TIMER2

Bộ định thời TIMER2 là bộ định thời 8 bits với một bộ đếm và một bộ potscaler. Nó thường dùng chung với bộ CCP trong chế độ PWM (sẽ được đề cập ở phần sau). Thanh ghi TMR2 có thể đọc hoặc ghi và được xóa khi có bất kì tín hiệu reset nào của thiết bị.

Bộ định thời TIMER2 có một thanh ghi chu kỳ 8 bits, PR2. Bộ định thời tăng số đếm lên từ 00h đến giá trị được ghi trong thanh ghi TR2 và sau đó Reset lại giá trị 00h trong chu kỳ kế tiếp. PR2 là thanh ghi có thể đọc hoặc ghi.

Giá trị trùng hợp trong thanh ghi TMR2 được đi qua bộ postscaler 4 bits để phát ra một ngắt TMR2 (được đặt ở bit cờ ngắt TMR2IF). Bộ định thời TIMER2 có thể được tắt (không hoạt động) bằng cách xóa bít điều khiển TMR2ON để giảm thiểu công suất tiêu tán nguồn.



Hình 1.17. Sơ đồ khối của TIMER2

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

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
0010 = 1:3 postscale
•
•
•
1111 = 1:16 postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Hình 1.18. T2CON: Thanh ghi điều khiển Timer2 (địa chỉ 12h)

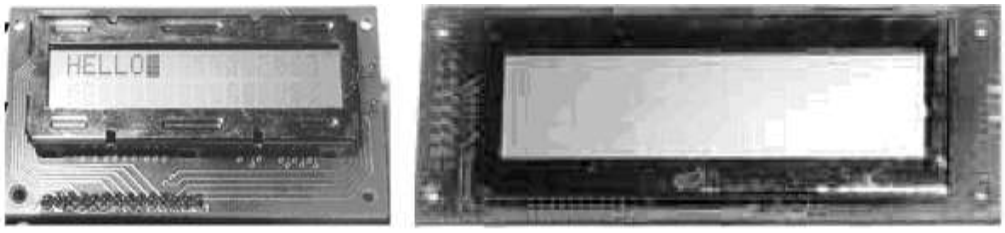
Một đặc điểm khác của vi điều khiển Pic16F877A là có bộ dao động chủ trên chip điều khiển, nó sẽ giúp tránh được những sai số không cần thiết trong việc tạo xung dao động, vi điều khiển Pic16F877A có khả năng tự Reset bằng bộ WDT, và có thêm 256 byte EEPROM. Nhưng giá thành của Pic đắt hơn so với 8051.

1.2. HIỂN THỊ LCD

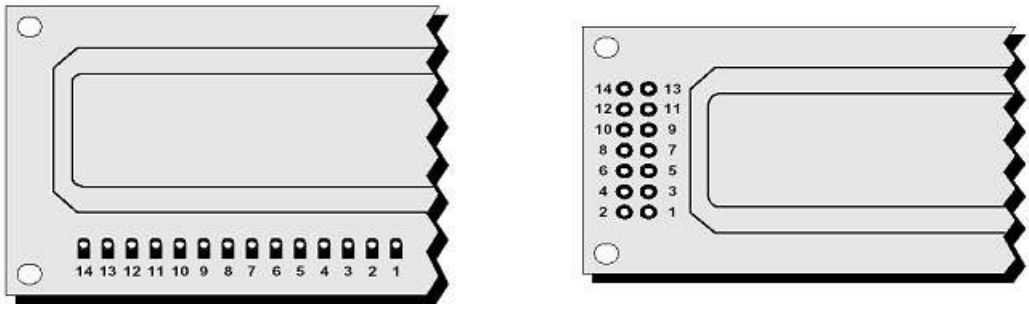
Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác như nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp 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.

1.2.1. Hình dáng kích thước.

Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau, trên hình 1.19. là hai loại LCD thông dụng.



Hình 1.19. Hình hai loại LCD thông dụng.



Hình 1.20. Sơ đồ chân của LCD

Hình 1.21. LCD loại DM 1602A.

Khi sản xuất LCD, nhà sản xuất đã tích hợp chíp 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 1.20.

1.2.2. Các chân chức năng.

Bảng 3.1. 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

1.2.3. Sơ đồ khối của HD44780.

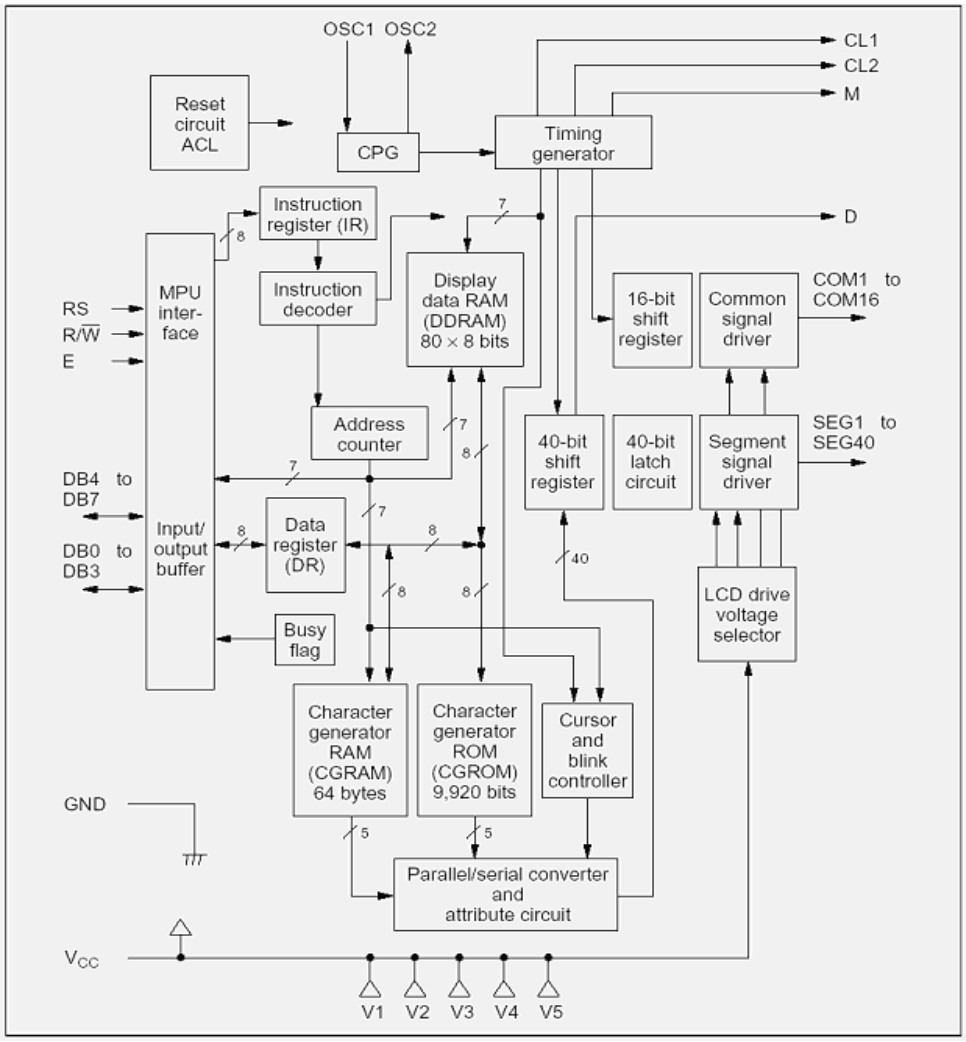
Để hiểu rõ hơn chức năng các chân và hoạt động của chúng, ta tìm hiểu sơ qua chíp HD44780 thông qua các khối cơ bản của nó.

- Các thanh ghi:

Chíp HD44780 có 2 thanh ghi 8 bits 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.

VD: Lệnh "hiển thị màn hình" có địa chỉ lệnh là 00001100 (DB7…DB0)



Hình 1.22. 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 3.2. 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):

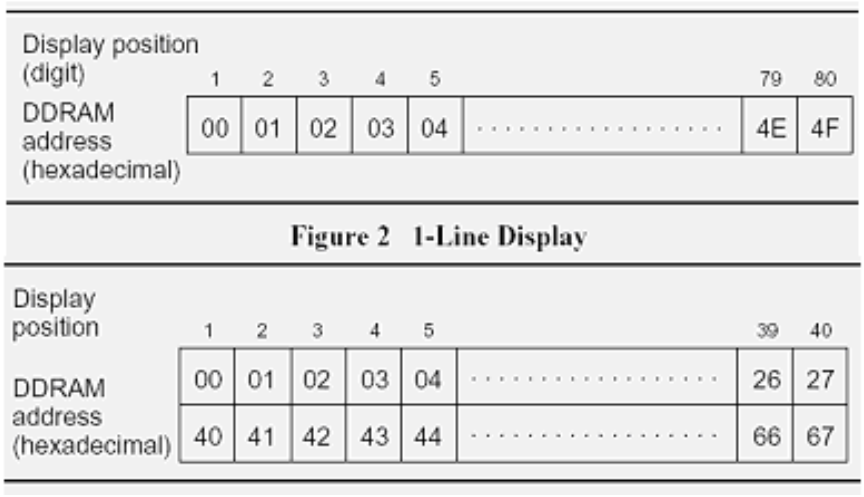
Khi thực hiện các hoạt động bên trong chíp, mạch nội bên trong cần một khoảng thời gian để hoàn tất. Khi đang thực thi các hoạt động bên trong chíp như thế, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF (thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để báo cho MPU biết nó đang "bận". Dĩ 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 t_{ADD} khoảng 4µS-5µS (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 bits, LCD sẽ hiển thị tại vị trí tương ứng trên màn hình một kí tự có mã 8 bits mà bạn đã cung cấp như hình 1.23.



Hình 1.23. Mối liên hệ giữa địa chỉ của DDRAM và vị trí hiển thị của LCD.

Vùng RAM này có 80x8 bits nhớ, nghĩa là chứa được 80 kí tự mã 8 bits. 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 bits. 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 1.24. 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 0000h 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 1.24.

1.2.4. 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 3.2)

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 (High … Low)	5 4 3 2 1 0 (High … Low)	7 6 5 4 3 2 1 0 (High … Low)
0 0 0 0 * 0 0 0	0 0 0	* * * 1 1 1 1 0 / 1 0 0 0 1 / 1 0 0 0 1 / 1 1 1 1 0 / 1 0 1 0 0 / 1 0 0 1 0 / 1 0 0 0 1 / * * * 0 0 0 0 0 — Character pattern (1); Cursor position
0 0 0 0 * 0 0 1	0 0 1	* * * 1 0 0 0 1 / 0 1 0 1 0 / 1 1 1 1 1 / 0 0 1 0 0 / 1 1 1 1 1 / 0 0 1 0 0 / 0 0 1 0 0 / * * * 0 0 0 0 0 — Character pattern (2); Cursor position
0 0 0 0 * 1 1 1	1 1 1	* * *

Hình 1.25. 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.

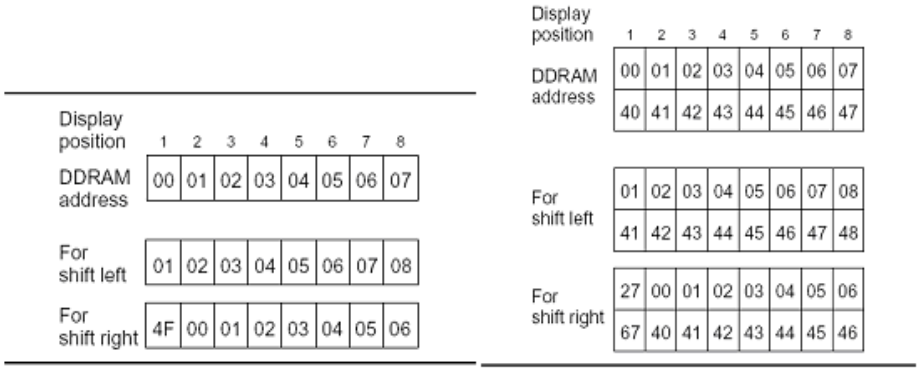
* Địa chỉ của RAM (AC) sẽ tự động tăng (giảm) 1 đơn vị, mỗi khi có lệnh ghi vào RAM. (Điều này giúp chương trình gọn hơn)

* 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 3.3. 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.	37µs

	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	
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

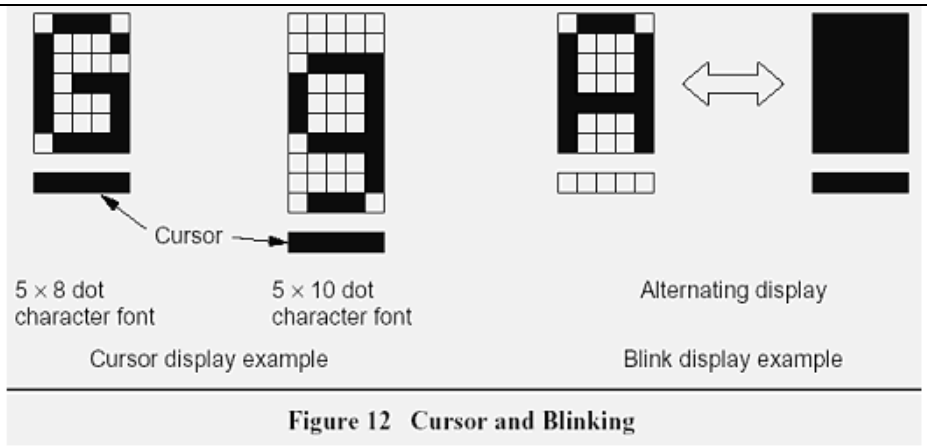
<table>
<tr><td></td><td> 

Figure 12 Cursor and Blinking
Hình 3.8. Kiểu con, kiểu kí tự và nhấp nháy kí tự</td><td></td></tr>
<tr><td>Cursor or display shift</td><td>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:
<table><tr><th>S/C</th><th>R/L</th><th>Hoạt động</th></tr><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></table></td><td>37µs</td></tr>
<tr><td>Function set</td><td>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</td><td></td></tr>
</table>

	được truyền/nhận 2 lần liên tiếp với 4 bit cao gửi/nhận trước, 4 bit thấp gửi/nhận sau. N: Thiết lập số hàng hiển thị. Khi N=0: hiển thị 1 hàng, N=1: hiển 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	37µs

	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.	
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	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	37µs tADD

or DDRAM	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.	4µs

1.2.5. 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 3.4. 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 Vcc = 4.5V đến 5.5V, T = -30 đến +75C).

Bảng 3.5. Miền làm việc bình thường.

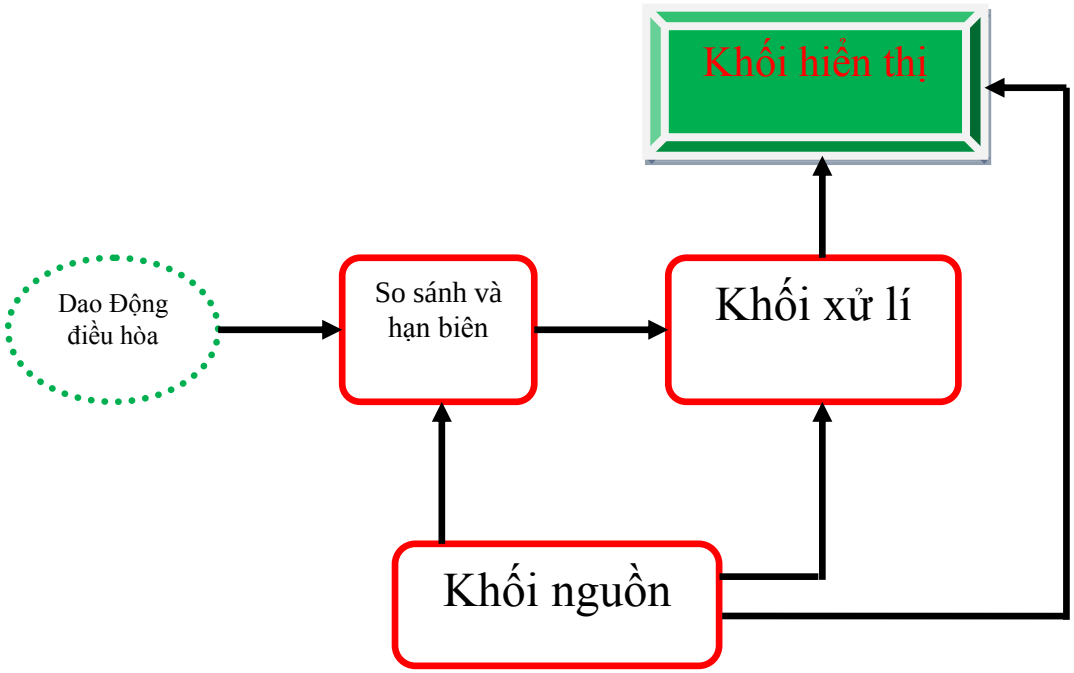
Chân cấp nguồn Vcc-GND	2.7V đến 5.5V
Điện áp vào mức cao V_{IH}	2.2V đến Vcc
Điện áp vào mức thấp V_{IL}	-0.3V đến 0.6V
Điện áp ra mức cao (DB0-DB7)	Min 2.4V (khi I_{OH} = -0.205mA)
Điện áp ra mức thấp (DB0-DB7)	Max 0.4V (khi I_{OL} = 1.2mA)
Dòng điện ngõ vào (input leakage current) I_{LI}	-1uA đến 1uA (khi V_{IN} = 0 đến Vcc)
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 2

THIẾT KẾ BỘ ĐẾM TẦN SỐ

2.1. SƠ ĐỒ KHỐI

Với mục đích của đề tài là thiết kế hệ bộ đếm tần số bằng cách sử dụng vi điều khiển và hiển thị kết quả trên LCD, ta có sơ đồ khối hình 2.1.



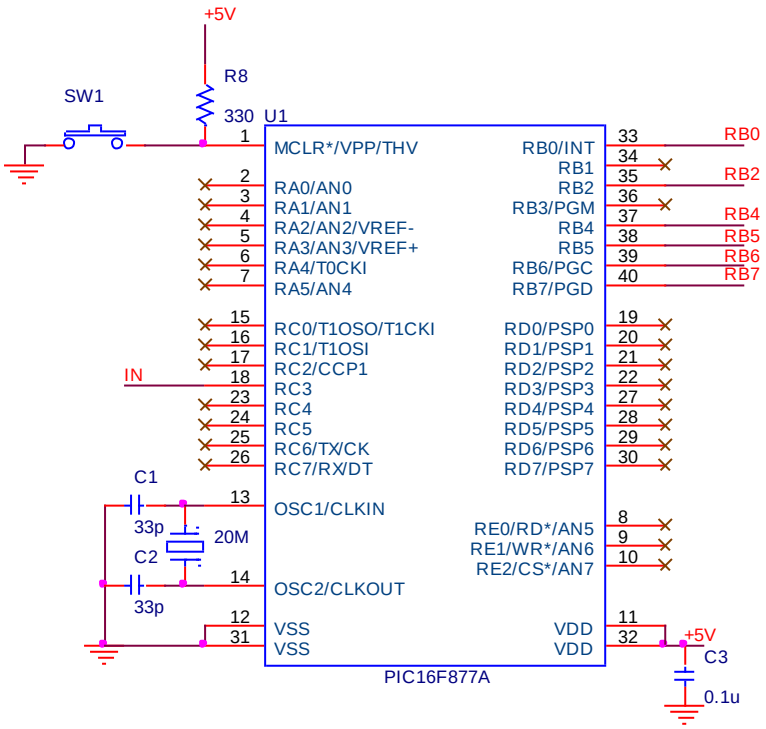
Hình 2.1. Sơ đồ khối bộ đếm tần số

Với sơ đồ này, thì dao động điều hòa (các điện áp hoặc dòng điện biến thiên tuần hoàn) là đối tượng cần đo, tín hiệu này được đưa vào bộ so sánh và hạn biên để tạo được xung vuông [0-5]V có cùng tần số ở đầu ra. Khối xử lý đếm số mẫu trong một chu kỳ của tín hiệu vào, từ đó tính ra tần số của dao động, đồng thời tính toán, điều khiển hiển thị kết quả trên LCD.

2.2. THIẾT KẾ CÁC KHỐI

2.2.1. Bộ xử lý

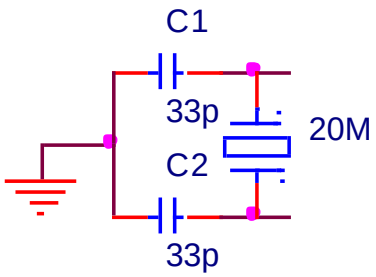
Bộ xử lý làm nhiệm vụ nhận tín hiệu từ mạch so sánh và hạn biên, đếm số mẫu trong một chu kỳ, từ đó tính ra tần số, hiển thị lên LCD. Như đã phân tích trong chương 1, ở đây em sử dụng vi điều khiển PIC16F877A. Đây là vi điều khiển có 40 chân, với 5 cổng vào ra là Port A (RA0÷RA5), Port B (RB0÷RB7), Port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



Hình 2.2. Sơ đồ bộ xử lý

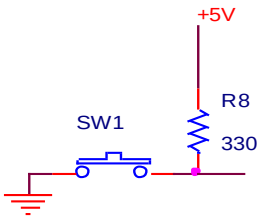
Mạch tạo xung dao động: Mạch tạo dao động được sử dụng để cung cấp xung đồng hồ cho vi điều khiển. Xung đồng hồ là cần thiết để vi điều khiển thực hiện các chu kỳ lệnh của chương trình phần mềm. Với mỗi loại vi điều khiển PIC hỗ trợ những kiểu mạch tạo dao động khác nhau như mạch dao động thạch anh (XT, HS), mạch dao động RC, mạch dao động nội, các nguồn dao động chuẩn bên ngoài khác. Ở đề tài này em sử dụng vi điều khiển PIC 16F877A nên sử dụng mạch dao động dùng thạch anh là thích hợp nhất.

Mạch dao động thạch anh có sơ đồ như hình 2.3. Mạch này là nguồn cung cấp xung đồng hồ chính cho CPU và tất cả các khối trong PIC. Hai chân OSC1 (chân 13) và OSC2 (chân 14) được mắc với mạch dao động thạch anh bên ngoài. Các tụ C1 và C2 với trị số 33pF. Mạch dao động thạch anh có ưu điểm là tạo ra xung đồng hồ có tần số cao và chính xác.



Hình 2.3. Mạch dao động thạch anh

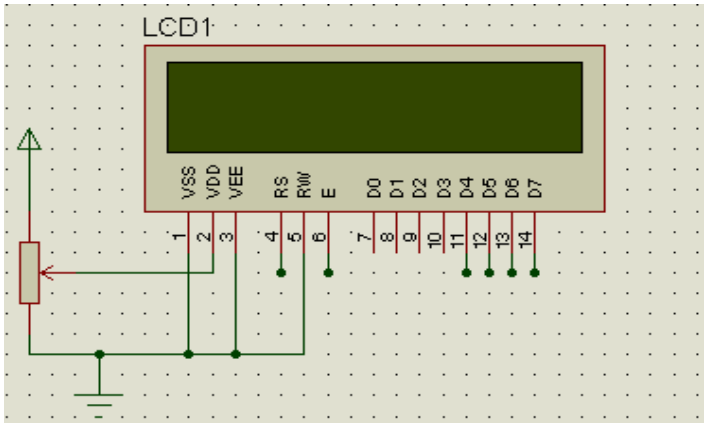
Mạch reset: Đặt lại hoạt động cho bộ xử lý, sơ đồ mạch trong hình 2.4. Bình thường chân MCLR có mức logic cao (được nối với nguồn 5V), khi SW1 đóng mạch (thực hiện reset) thì MCLR được nối với đất (có mức logic thấp).



Hình 2.4. Mạch Reset cho PIC

2.2.2. Khối hiển thị

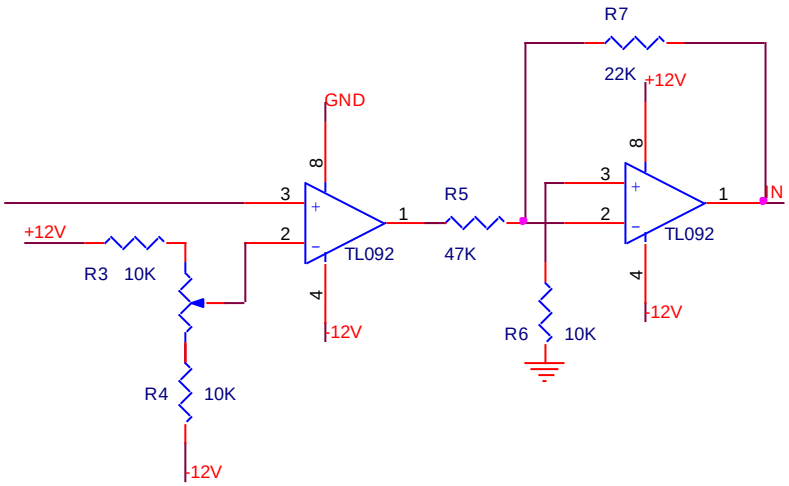
Để thuận tiện cho việc hiển thị kí tự, giá trị, trạng thái điều khiển, ở đây em sử dụng LCD_DM 1602A.



Hình 2.5. Sơ đồ nguyên lý của LCD1602

LCD1602 là loại 2 dòng, 16 kí tự, 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).

2.2.3. Mạch so sánh và hạn biên



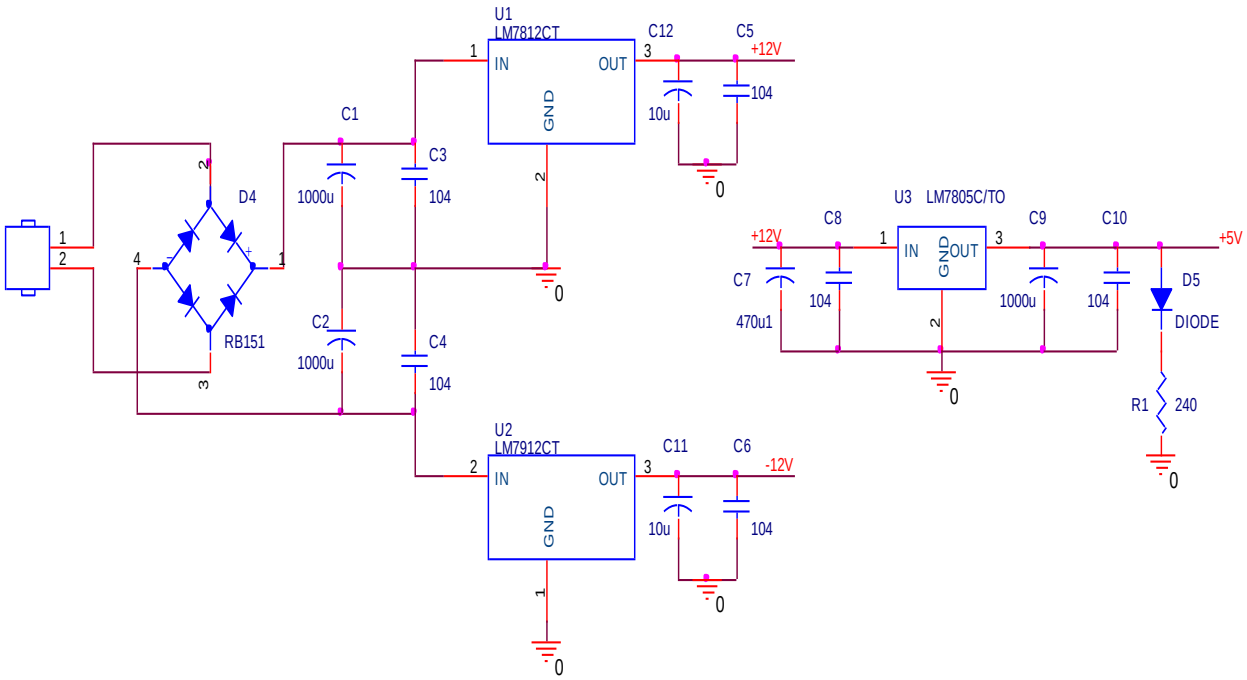
Hình 2.6. Mạch so sánh và hạn biên

Tín hiệu dao động điều hòa cần đo tần số được đưa vào chân IN + (chân 3) của IC TL092 và được so sánh với điện áp vào ở chân IN – (chân 2) của IC TL092. Đầu ra ta được xung vuông (biên độ [-10V; 0V]) có cùng tần số với tín hiệu vào, tiếp tục ta hạn biên về [0V; 5V] bằng mạch khuếch đại đảo pha có hệ số bằng -0.5.

Tín hiệu ra từ mạch so sánh và hạn biên được đưa vào PORT C (chân RC3) của vi điều khiển PIC16F877A (bộ xử lý).

2.2.4. Khối nguồn

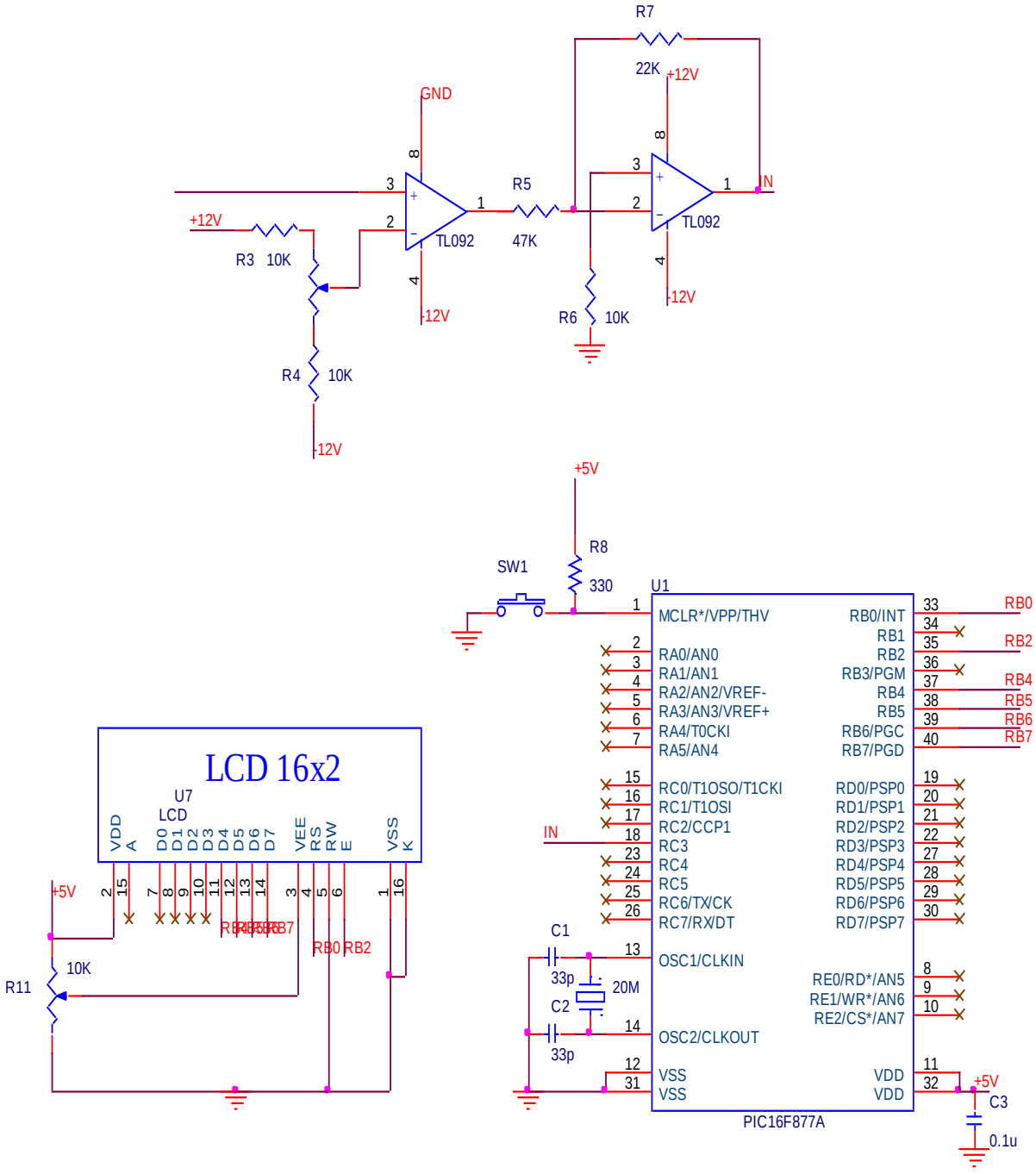
Để cung cấp nguồn nuôi cho toàn bộ hệ thống em đã sử dụng nguồn đối xứng +12V, -12V. Bộ ổn áp dùng IC 7812, 7912, 7805 để tạo các nguồn +12V, -12V, +5V ổn định cấp cho mạch hiển thị và bộ xử lí. Các tụ trong mạch có nhiệm vụ lọc nhiễu, diode D5 có nhiệm vụ báo nguồn. Sơ đồ mạch như hình 2.7.



Hình 2.7. Sơ đồ nguyên lý của bộ nguồn

2.3. SƠ ĐỒ MẠCH HỆ THỐNG

Sơ đồ mạch bộ đếm tần số trong hình 2.8. Tín hiệu vào (u_{IN}) là các dao động điều hòa có dạng bất kỳ được đưa vào chân (+) của mạch so sánh, chân (-) của mạch so sánh được chỉnh sao cho có mức điện áp nằm trong dải ($u_{INmin} \div u_{INmax}$), như vậy đầu ra của mạch so sánh ta được xung vuông, qua mạch hạn biên ta được xung vuông có mức logic theo chuẩn TTL đưa vào PORT C (chân RC3) của vi điều khiển PIC16F877A (bộ xử lý). Chương trình phần mềm trong bộ xử lý dựa vào tín hiệu nhận được xử lý đếm được số mẫu trong một chu kỳ dao động, ta nhân số mẫu đó với khoảng thời gian giữa hai mẫu, từ đó tính được tần số dao động của tín hiệu, đồng thời đưa dữ liệu hiển thị trên màn hình LCD trên các chân RB4÷RB7.

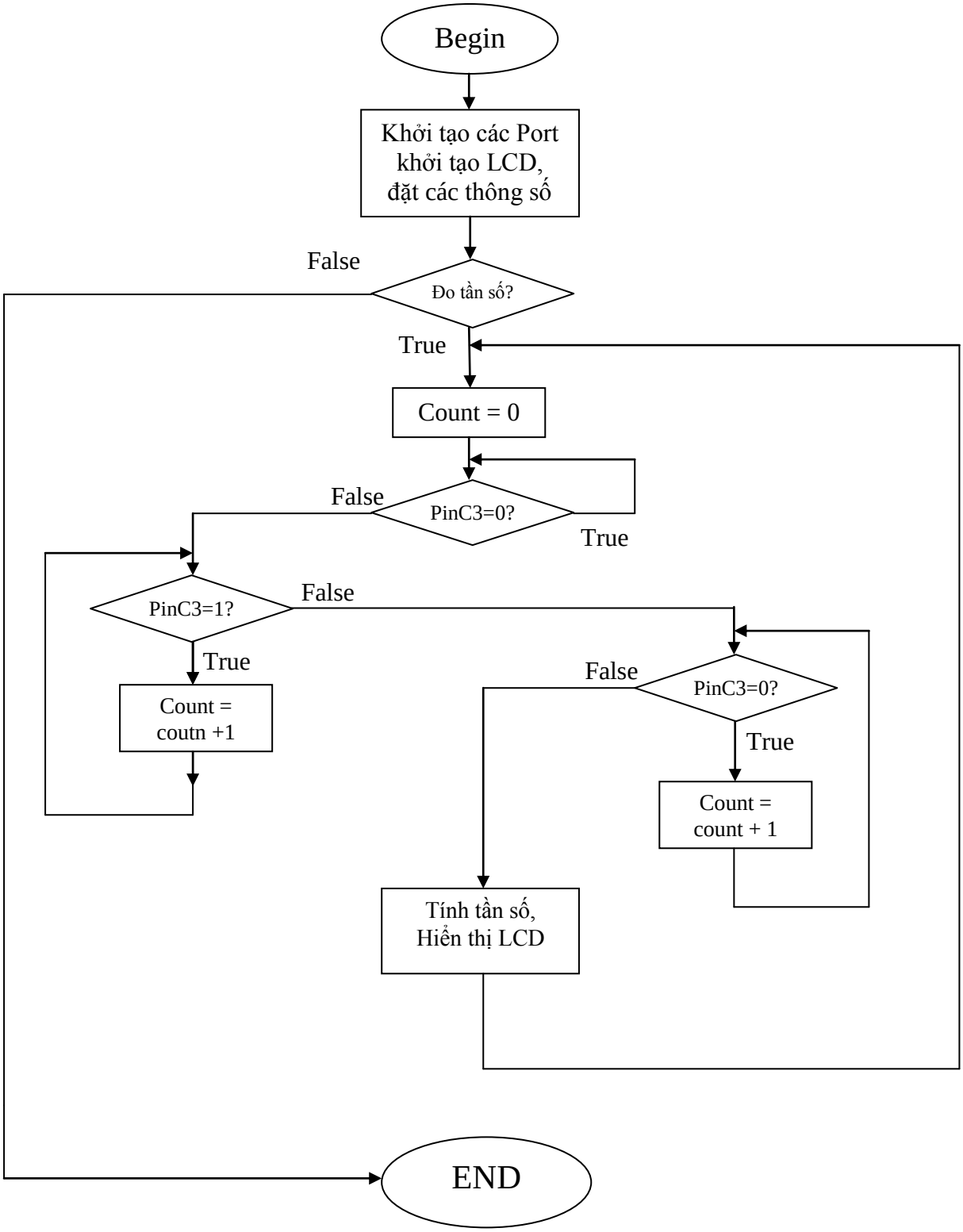


Hình 2.8. Sơ đồ nguyên lý mạch đếm tần số

Chương 3

PHẦN MỀM ĐIỀU KHIỂN

3.1. LƯU ĐỒ THUẬT TOÁN



3.2. CHƯƠNG TRÌNH

```
// Tên chương trình  : Thiết kế bộ đếm tần số
// Phần mềm dịch     : CCS
// Mô tả phần cứng   : Sử dụng PIC 16F877A – Thạch anh 20Mhz
//                     LCD HD44780 giao tiếp với Port B
//                     Chân RC3 là ngõ vào
//                     Tần số được hiển thị lên 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>
#use fast_io(b)
#use fast_io(c)
#INT_RB
void Convert_BCD(float x);
void OutLCD();
float Count, Val;
signed int8 Ts,In;
signed int8 V0, V1, V2, V3, V4, V5;
  void main()
  {
```

```
V0=0;V1=0;V2=0;V3=0;V4=0;V5=0;
LCD_init();
LCD_putcmd(0x83);
printf(LCD_putchar,"Freq Counter");
LCD_putcmd(0xC0);
printf(LCD_putchar,"Freq: ");
LCD_putcmd(0xC6);
LCD_putchar(V0 +0x30);
LCD_putcmd(0xC7);
LCD_putchar(V1 +0x30);
LCD_putcmd(0xC8);
LCD_putchar(V2 +0x30);
LCD_putcmd(0xC9);
printf(LCD_putchar,".");
LCD_putcmd(0xCA);
LCD_putchar(V3 +0x30);
LCD_putcmd(0xCB);
LCD_putchar(V4 +0x30);
LCD_putcmd(0xCC);
LCD_putchar(V5 +0x30);
LCD_putcmd(0xCD);
printf(LCD_putchar,"kHz");
delay_ms(500);
```

```
while(1)
{  Count=0;
   while(!input(PIN_C3));
   while(input(PIN_C3))
   {
      Count=Count+1;
   }
   while(!input(PIN_C3))
   {
      Count=Count+1;
   }
   Val=4000000.0/Count;
   Convert_BCD(Val);
   OutLCD();
   delay_ms(100);
}
}
 void Convert_BCD(float x)
 {
   V0=(int8)x/100;
   V1=(int8)(x/10)%10;
   V2=(int8)x%10;
   V3=(int8)(x*10)%10;
```

```
    V4=(int8)(x*100)%10;
    V5=(int8)(x*1000)%10;
  }
 void OutLCD()
  {
    LCD_putcmd(0xC0);
    printf(LCD_putchar,"Freq: ");
    LCD_putcmd(0xC6);
    LCD_putchar(V0 +0x30);
    LCD_putcmd(0xC7);
    LCD_putchar(V1 +0x30);
    LCD_putcmd(0xC8);
    LCD_putchar(V2 +0x30);
    LCD_putcmd(0xC9);
    printf(LCD_putchar,".");
    LCD_putcmd(0xCA);
    LCD_putchar(V3 +0x30);
    LCD_putcmd(0xCB);
    LCD_putchar(V4 +0x30);
    LCD_putcmd(0xCC);
    LCD_putchar(V5 +0x30);
  }
```

KẾT LUẬN

Sau một thời gian nghiên cứu và xây dựng thực nghiệm, em đã hoàn thành đề tài "THIẾT KẾ BỘ ĐẾM TẦN SỐ". Với hệ thống này ta có thể đo được tần số của các tín hiệu điều hòa tương đối chính xác, nhưng do tốc độ đáp ứng của các linh kiện có hạn, nên hiện tại mới chỉ đo được các tín hiệu có tần số thấp (nhỏ hơn vài kHz). Để khắc phục nhược điểm này ta có thể chọn các linh kiện có tốc độ đáp ứng cao hơn, bộ xử lý chọn vi điều khiển có tốc độ đáp ứng cao hơn (với xung nhịp tần số cao). Qua đồ án em cũng đã hiểu thêm được về phương pháp điều khiển thông qua vi điều khiển và ứng dụng quan trọng của vi điều khiển trong đo lường và điều khiển.

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 nhận được sự đóng góp ý kiến của thầy cô trong hội đồng và các bạn để đồ án tốt nghiệp của em được hoàn thiện hơn.

Sau cùng em xin trân thành bày tỏ lòng biết ơn của mình đối với thầy NGUYỄN VĂN DƯƠNG và các thầy cô trong khoa đã giúp đỡ em hoàn thành đồ án tốt nghiệp này.

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, Khoa Học và Kỹ Thuật*, Hà Nội.

5. Tống Văn On, *Họ vi điều khiển 8051*, Nhà xuất bản Lao Động – Xã Hội.

6. Các bạn có thể truy cập vào một số trang web sau.

www.dientuvietnam.net

www.picvietnam.com

www.dientuvienthong.net

www.vagam.dieukhien.net

www.duyphi.phpnet.us/index.htm