

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: KỸ THUẬT MÁY TÍNH

ĐỒ ÁN MÔN HỌC

MÔN HỌC

HỆ THỐNG NHÚNG

Nhóm sinh viên: - Hà Anh Tuyên

- Đào Thị Vân

- Hà Văn Triều

Lớp: 43s

Giáo viên hướng dẫn: Tăng Cẩm Nhung

Thái Nguyên 2011

Nhận xét của giáo viên hướng dẫn:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, Ngày...Tháng...Năm 20.....

Giáo Viên hướng dẫn

(Ký ghi rõ họ tên)

Nhận xét của giáo viên chấm:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, Ngày...Tháng...Năm 20.....

Giáo Viên chấm

(Ký ghi rõ họ tên)

LỜI MỞ ĐẦU

Ngày nay, khoa học kỹ thuật ngày càng phát triển, nhu cầu của con người ngày càng tăng. Điện tử là một trong những ngành khoa học kỹ thuật có thể đáp ứng được nhu cầu của con người. Trong ngành điện tử thì có rất nhiều lĩnh vực khác nhau... nhưng vi điều khiển là một trong những lĩnh vực được ứng dụng rộng rãi trong đời sống để phục vụ cho con người. Vì nó là một vi mạch rất nhỏ nhưng giải quyết được rất nhiều vấn đề khó khăn, phức tạp.

Kinh tế ngày càng phát triển kéo theo đó con người càng phải tính toán nhiều, với những con số khổng lồ vì những nhu cầu thực tế đó máy tính đã ra đời phục vụ cho đời sống con người.

Thấy được vai trò quan trọng của máy tính trong đời sống nhóm em đã chọn chủ đề “Xây dựng mô hình máy tính cá nhân” làm đề tài của nhóm mình

Được sự hướng dẫn tận tình của các thầy cô giáo trong bộ môn và sự cố gắng của cả nhóm dựa trên lý thuyết học trên lớp và kiến thức thực tế đến nay đồ án của nhóm em đã hoàn thành.

Do thời gian và kiến thức còn hạn chế nên bài làm của nhóm em còn nhiều sai sót, em mong nhận được sự chỉ bảo tận tình của các thầy cô để chúng em nắm rõ hơn về trình tự thiết kế và để đồ án của chúng em hoàn thiện hơn.

Chúng em xin chân thành cảm ơn !

Nhóm sinh viên thực hiện

Hà Anh Tuyên

Hà Văn Triều

Đào Thị Vân

MỤC LỤC

	Trang
Lời mở đầu	4
Mục lục	5
 Chương 1 Khảo sát và phân tích bài toán	
1.1 Vai trò của máy tính	7
1.2 Một số máy tính trong đời sống	7
1.3 Xác định bài toán	9
 Chương 2 Thiết kế hệ thống	
2.1 Sơ đồ tổng thể	10
2.2 Lựa chọn giải pháp thiết kế	11
2.3 lựa chọn công nghệ	12
2.3.1 Khối điều khiển	13
2.3.2. Thiết bị hiển thị	14
2.3.3 Khối nhập dữ liệu	14
2.4. Tìm hiểu về các linh kiện được sử dụng trong hệ thống	15
2.4.1 PIC 16F877A	15
2.4.1.1 Đặc tính nổi bật của bộ vi xử lý	15
2.4.1.2 Mô tả sơ qua các chân của PIC 16F877A	17
2.4.2. LCD	24
2.4.3. KEYPAD 4x4	27
 Chương 3 Xây dựng hệ thống	
3.1. Sơ đồ nguyên lý tổng quát	29
3.1.1. Sơ đồ mạch nguyên lý	29
3.1.2. Sơ đồ thuật toán tổng quát	30
3.2. Sơ đồ đặc tả :	31
3.2.1. Khối nguồn	31

3.2.2. Khối xử lý trung tâm	32
3.2.3. Khối Reset	32
3.2.4. Khối tạo xung dao động ngoài	33
3.2.5. Khối hiển thị	33
3.2.6. Khối nhập dữ liệu	35
3.3 Chương trình	35
TÀI LIỆU THAM KHẢO	44

CHƯƠNG 1

KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN

1.1 VAI TRÒ CỦA MÁY TÍNH:

Bạn nghĩ chúng ta sẽ làm việc thế nào nếu không có máy tính?

Chúng ta sẽ mất rất nhiều thời gian để tính toán mà vẫn mắc rất nhiều sai lầm,

Thầy giáo sẽ phải ra đề kiểm tra như thế nào nếu học sinh không có máy tính”?”và học sinh sẽ làm bài thế nào”?”hay đất nước sẽ phát triển thế nào nếu không có máy tính”?” ?

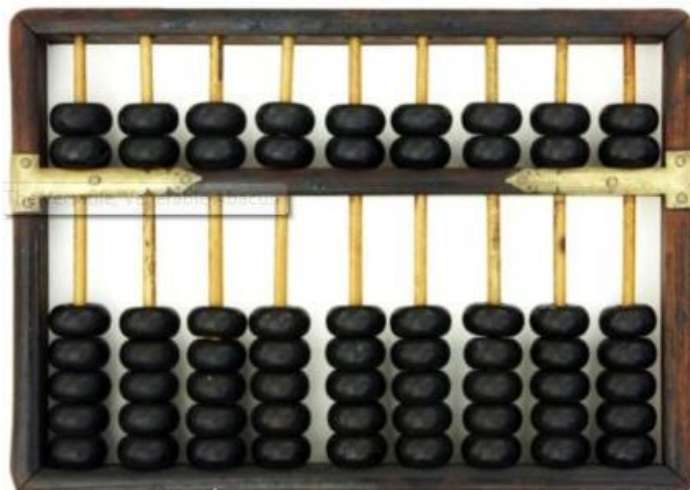
Từ xưa con người đã biết chế tạo ra : bàn phím số ở TRUNG QUỐC phục vụ cho việc tính toán

Đặc biệt trong nền sản xuất hiện đại, máy tính được coi là thiết bị quan trọng trong nhiều lĩnh vực như: trong các đơn vị hành chính sự nghiệp, các nhà máy, phân xưởng sản xuất, các khu công nghiệp...,

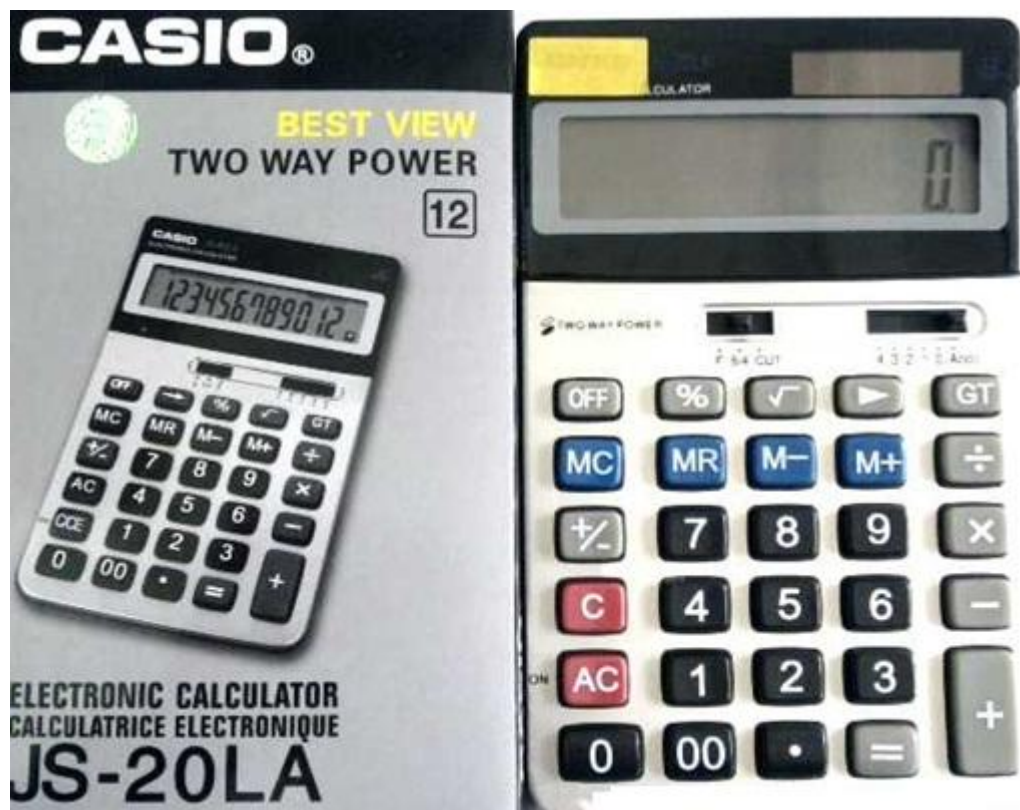
Máy tính giúp con người tính toán đơn giản đồng thời cho năng suất ,độ chính xác cao,tiết kiệm thời gian và không mắc phải những sai lầm đáng tiếc như việc tính nhầm hay tính bằng tay...

1.2 MỘT SỐ MÁY TÍNH TRONG ĐỜI SỐNG:

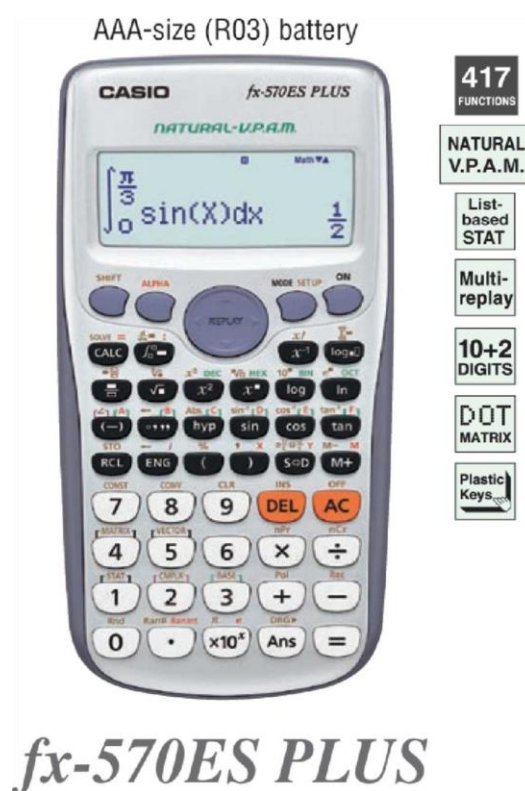
Bàn phím số:là dạng máy tính sơ khai,đơn giản



Máy tính cao cấp hơn: Máy tính Casio DF 120TM



Máy tính với nhiều chức năng hơn: casio FX 570ES



Máy tính để tính toán và in hóa đơn: Máy tính tiền Casio SE-S300



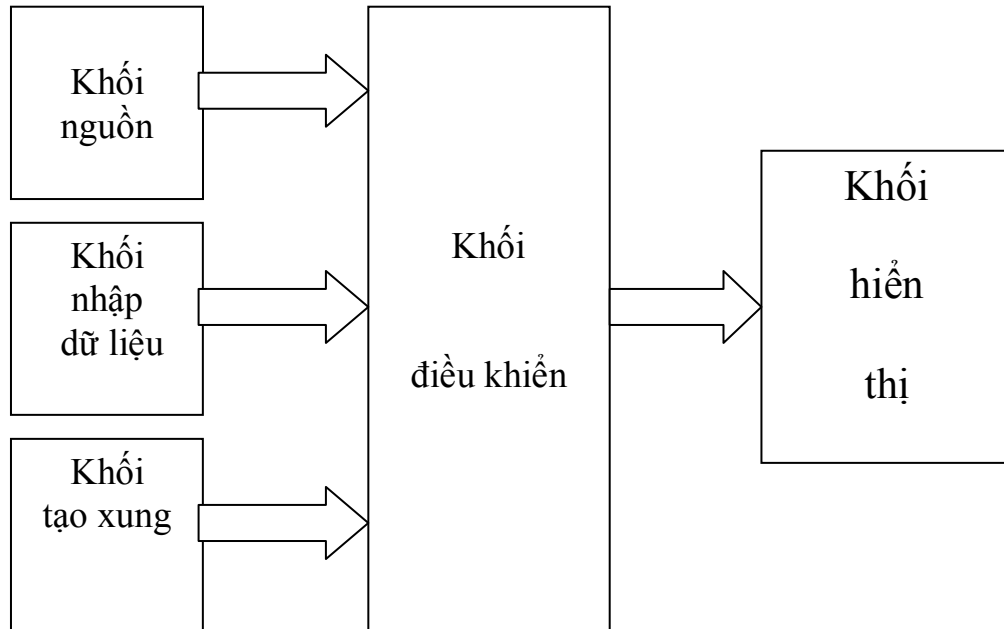
Hình 1: Một số máy tính trong thực tế

1.3 XÁC ĐỊNH BÀI TOÁN:

1. Hệ thống có khả năng thực hiện các phép tính đơn giản như (+, -, *, / , $\sqrt{}$, ...).
2. Thực hiện các phép tính trong phạm vi 8 chữ số.
3. Chất lượng hiển thị các kí tự trên khối hiển thị tốt, rõ ràng.
4. Có thể quan sát được kết quả hiển thị ở mọi điều kiện ánh sáng.
5. Hệ thống luôn làm việc ổn định khi nhiệt độ môi trường thay đổi (không phụ thuộc vào điều kiện môi trường)
6. Màu sắc của các kí tự khi hiển thị là đen trắng nhưng phải đảm bảo rõ ràng, dễ nhìn, dễ đọc kết quả.
7. Kích thước nhỏ gọn, đơn giản.....
8. thời gian tính toán nhanh.

CHƯƠNG 2 : THIẾT KẾ HỆ THỐNG

2.1 SƠ ĐỒ KHỐI TỔNG THỂ:



Chức năng của từng khối:

- + Khối nguồn: cung cấp điện áp chuẩn 5V cho PIC hoạt động.
- + Khối nhập dữ liệu: nhập số và các phép tính.
- + Khối tạo xung: tạo xung dao động cho hệ thống.
- + Khối điều khiển: Xử lý các thông tin nhập vào từ khối nhập dữ liệu mã hóa và đưa ra khối hiển thị.
- + Khối hiển thị: hiển thị các kết quả đưa ra từ khối điều khiển.

Nguyên lý hoạt động của sơ đồ:

Khi các khối tạo xung và khối nguồn được đảm bảo lúc đó hệ thống được cấp nguồn và xung dao động và bắt đầu hoạt động. Khối nhập dữ liệu là bàn phím gồm các nút bấm nên khi các nút này được bấm thì ngay khi đó các thông tin về phím bấm thông qua dạng tín hiệu điện được truyền đến khối điều khiển, tại đây các thông tin này được xử lý theo thuật toán được lập trình sẵn và đưa ra các kết quả xử lý ra khối hiển thị để hiển thị dạng số ra màn hình giúp người sử dụng có thể nhìn thấy được các kết quả này.

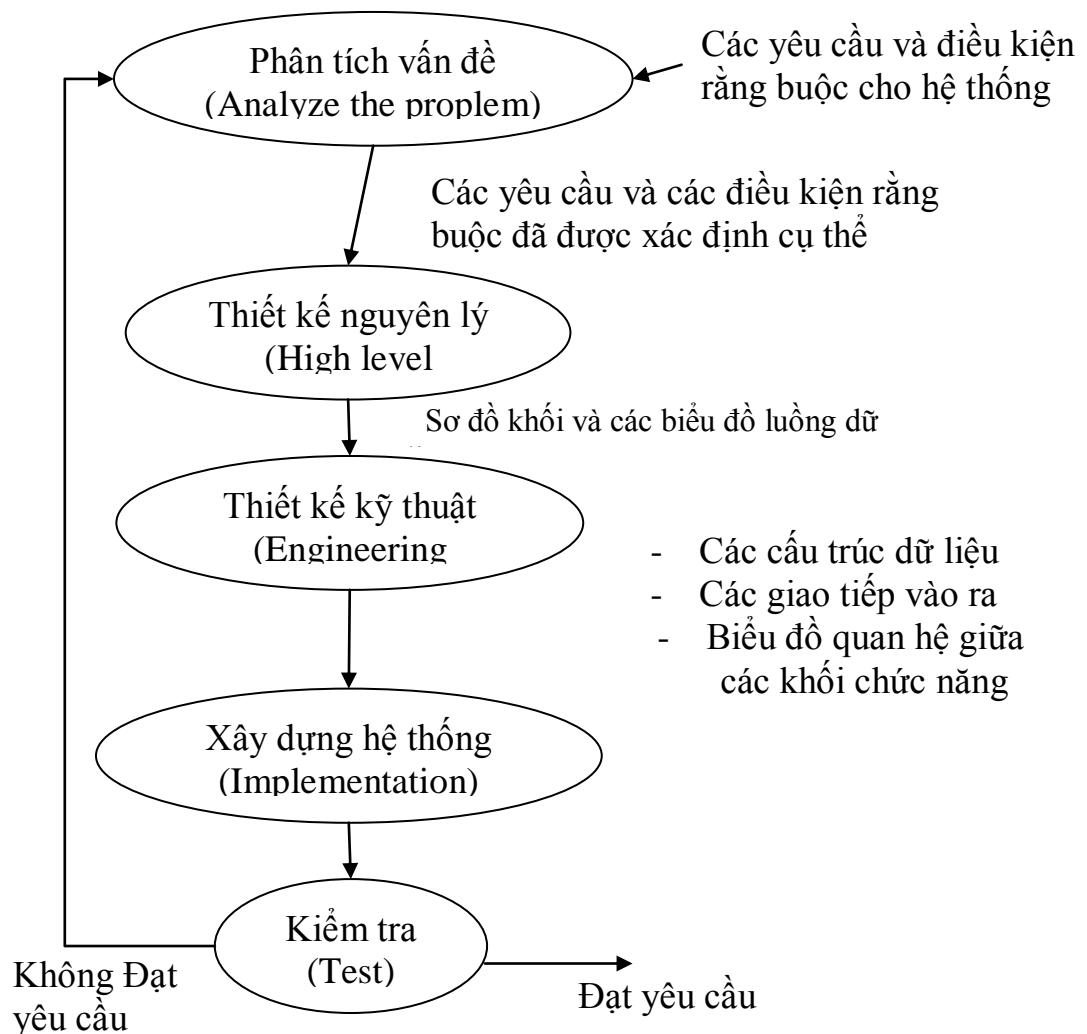
2.2 LỰA CHỌN GIẢI PHÁP THIẾT KẾ

Thiết kế hệ thống có vai trò rất quan trọng, chất lượng của phần mềm phụ thuộc rất nhiều vào bản thiết kế. Một bản thiết kế tốt còn giúp cho việc thực hiện các giai đoạn khác dễ dàng hơn, giúp cho người thực hiện hoàn thành chính xác hơn công việc của mình. Các quy trình thiết kế thường được sử dụng như: Top-Down, Bottom-Up hoặc kết hợp cả hai quy trình trên.

Quy trình Top-down: Quy trình này tiếp cận bài toán theo hướng xem xét bài toán từ các khía cạnh chi tiết và sau đó mới tổng quát lên. Quy trình Top-Down thường được áp dụng cho các bài toán đã có giải pháp công nghệ cả về phần mềm cũng như phần cứng. Các giải pháp này đã được phát triển trước đó ở các ứng dụng khác, và đã được kiểm định.

Trong thực tế chúng ta sẽ thấy, bản chất hay mấu chốt của quy trình là vấn đề tìm hiểu và xác định bài toán, làm sao để xác định được chính xác và đầy đủ nhất các yêu cầu cũng các ràng buộc mà hệ thống phải đạt được.

Sơ đồ khối quy trình kế top-down:



Quy trình Bottom-Up : Quy trình Bottom-Up trong thực tế thường áp dụng trong các bài toán chưa lựa chọn hay chưa tìm ra được giải pháp công nghệ. Mấu chốt của quy trình tập trung chủ yếu và quá trình thử nghiệm với hệ thống và tín hiệu thực, từ đó chọn ra giải pháp công nghệ và linh kiện phù hợp nhất cho bài toán. Sơ đồ tổng quát của quy trình như hình 5.

Quy trình Bottom-Up bắt đầu từ các ý tưởng đơn lẻ, sau đó xây dựng luôn thiết kế kỹ thuật. Như ta thấy quy trình hoàn toàn ngược so với Top-Down. Quy trình này thường áp dụng có các bài toán chưa lắm chắc về lời giải, người thiết kế mới chỉ có ý tưởng về một vấn đề nào đó và muốn tìm một giải pháp hoặc giải pháp tốt nhất để giải quyết vấn đề. Việc giải quyết các ý tưởng có thể 1 hoặc nhiều để có một sản phẩm hoàn chỉnh. Ở quy trình này ta cần chú ý có 2 khâu test nhằm kiểm định chính xác lại các thiết kế kỹ thuật và thiết kế nguyên lý trước khi lựa chọn 1 giải pháp tối ưu nhất.

Chính từ việc thí nghiệm và thiết kế thử hệ thống trước, sau đó mới có thể phân tích nguyên lý để chọn các đặc tính mới, ràng buộc mới cho một hệ thống mới. Với quy trình này khâu thiết kế kỹ thuật và Test sau khi xây dựng hệ thống là quan trọng nhất. Vì với Top-Down việc xây dựng một sản phẩm là theo nhu cầu của người dùng và môi trường đặt hệ thống. Còn với Bottom-Up có thể người ta còn chưa tìm ra cách để thiết kế ra sản phẩm đó, hoặc sản phẩm đó chưa hề có trên thị trường, khi đó cả người dùng và người thiết kế chưa thể có thông tin gì về các yêu cầu cho sản phẩm hay các đặc tính kỹ thuật của sản phẩm, vì vậy khâu thiết kế kỹ thuật và Test sau thực thi các kỹ sư phải tìm ra các đặc tính đó, nhằm xác định được các ưu việt cũng như các hạn chế của sản phẩm mới.

Qua những phân tích ở trên và qua thực tế bài toán chúng em nhận thấy việc lựa chọn quy trình Top- down để thiết kế hệ thống là hợp lý hơn cả.

2.3. LỰA CHỌN CÔNG NGHỆ.

Trong hệ thống ta có thể thiết kế với nhiều loại linh kiện khác nhau, và trong từng khối ta cũng có nhiều lựa chọn khác nhau:

2.3.1 Khối điều khiển:

Có thể sử dụng các loại VDK như AVR, 8051, ARM, PIC mà có khả năng đáp ứng cho việc thực hiện các phép tính toán (+ - * /...) đáp ứng yêu cầu của bài toán.

Tính năng của một số VDK điển hình :

Vi điều khiển AVR

AVR là VDK do hãng Atmel sản xuất, là chip VDK 8 bit với cấu trúc tập lệnh đơn giản. AVR có các tính năng cơ bản sau:

- + Có thể sử dụng xung clock lên đến 16MHz, hoặc sử dụng xung clock nội lên đến 8MHz.
- + Bộ nhớ chương trình Flash có thể lập trình lại nhiều lần và dung lượng lớn, có SRAM lớn, và đặc biệt có bộ nhớ lưu trữ lập trình được EEPROM.
- + Nhiều ngõ vào ra (I/O port) 2 hướng.
- + Các bộ chuyển đổi Analog – Digital với độ phân giải 10 bit.
- + Giao diện nối tiếp UART (thích hợp với chuẩn RS232)

Vi điều khiển 8051

Vi điều khiển 8051 do hãng intel sản xuất, ra đời đầu tiên vào năm 1980, nó là vi điều khiển đầu tiên của họ MCS-51.

- Các tính năng cơ bản của 8051 như sau:
- + 8kB bộ nhớ chương trình.
- + Dao động bên ngoài với thạch anh < 24MHz.
- + 4 port xuất nhập.
- + 3 Timer/counter 16 bit.
- + Nạp chương trình song song hoặc nối tiếp.

Vi điều khiển ARM

ARM là một loại cấu trúc VXL 32-bit kiểu RISC được sử dụng rộng rãi trong các thiết kế nhúng.

Tuy nhiên trong hệ thống này lựa chọn PIC 18F877A là tối ưu nhất vì :

- Họ vi điều khiển này có thể dễ dàng mua trên thị trường.

- Tính phổ biến cao.
- Giá thành hợp lí.
- Có đầy đủ các tính năng của một vi điều khiển hoạt động độc lập.
- Sự hỗ trợ của nhà sản xuất về trình biên dịch, các công cụ lập trình, ngôn ngữ lập trình khá phổ biến.
- Không quá phức tạp trong lập trình.
- Lập trình trên LCD đơn giản hơn nhiều so với 8051.
- ARM có giá thành cao và lãng phí tài nguyên vì ARM là loại chip cao cấp có khả năng thực hiện công việc lớn hơn như thế nhiều lần.

2.3.2 Thiết bị hiển thị

Có thể sử dụng rất nhiều thiết bị hiển thị có khả năng đáp ứng cho hệ thống này như: LED 7 thanh, LED ma trận, LED đơn, LCD, hay thậm chí là màn hình máy tính.

Đặc điểm và tính năng một số thiết bị hiển thị:

LED 7 thanh

Thiết bị này có thể hiển thị một cách rõ ràng, giá thành rẻ, tuy nhiên số kí tự hiển thị bị hạn chế. Nếu dùng cho hệ thống này thì phải cần rất nhiều LED và kích thước sẽ rất lớn.

LED đơn

Đây cũng là thiết bị hiển thị tuy nhiên không thể dùng chỉ hệ thống vì sẽ cần rất nhiều LED cho hệ thống và kích thước của hệ thống sẽ rất lớn.

LCD LM016L

Đây là thiết bị hiển thị có độ phân giải cao, hiển thị kết quả rõ ràng, giá thành không quá đắt. Vì thế nó là thiết bị đáp ứng cho hệ thống này là tốt nhất.

2.3.3 Khối nhập dữ liệu.

Có thể sử dụng các thiết bị mà có khả năng nhập dữ liệu từ bên ngoài như KEYPAD, ma trận nút bấm, hay bàn phím máy tính....

KEYPAD là một "thiết bị nhập" chứa các nút nhấn cho phép người dùng nhập các chữ số, chữ cái hoặc ký hiệu vào bộ điều khiển. KEYPAD không chứa tất

cả bảng mã ASCII như keyboard và vì thế KEYPAD thường được tìm thấy trong các thiết bị chuyên dụng.

Như vậy hệ thống sẽ sử dụng các linh kiện cơ bản sau:

- +PIC 16F877A
- +LCD1 LM016L
- + KEYPAD 4×4

Và một số linh kiện khác như: tụ điện, thạch anh, điện trở...

2.4 Tìm hiểu về các linh kiện được sử dụng trong hệ thống

2.4.1 PIC 16F877A



www.HVWTech.com

2.4.1.1 Đặc tính nổi bật của bộ vi xử lý

- +Sử dụng công nghệ tích hợp cao RISC CPU
- +Người sử dụng có thể lập trình với 35 câu lệnh đơn giản
- +Tốc độ hoạt động :-xung đồng hồ vào là DC-20MHZ
 - Chu kỳ lệnh thực hiện trong 200ns
- +Bộ nhớ chương trình Flash 8k×14 words
- +Bộ nhớ Ram 368x8 bytes
- +Bộ nhớ EPROM 256x8 bytes
- **khả năng của bộ vi xử lý này:**
- +Khả năng ngắt(lên tới 14 nguồn ngắt trong và ngắt ngoài)

- +Bộ tạo xung thời gian(PWRT) và bộ tạo dao động(OST)
- +Ngăn nhớ stack được chia làm 8 mức
- +Truy cập bộ nhớ bằng địa chỉ trực tiếp hoặc địa chỉ gián tiếp
- +Nguồn khởi động lại (POR)
- +Bộ đếm xung thời gian(WDT) với nguồn dao động trên chip hoạt động tin cậy
- +Có mã chương trình bảo vệ
- +Phương thức cắt giữ SLEEP
- +Có bảng lựa chọn dao động
- +Công nghệ CMOS FLASH/EEPROM nguồn mức thấp ,tốc độ cao
- +Thiết kế hoàn toàn tĩnh
- +Mạch chương trình nối tiếp có 2 chân
- +Dải thế điện áp rộng:2.0v đến 5.5v
- +Nguồn sử dụng tại 25mA
- +Công suất tiêu thụ thấp :
 - <0.6mA với 5V,4MHZ
 - 20mA với nguồn 3V ,32KHZ
 - <1mA nguồn dự phòng

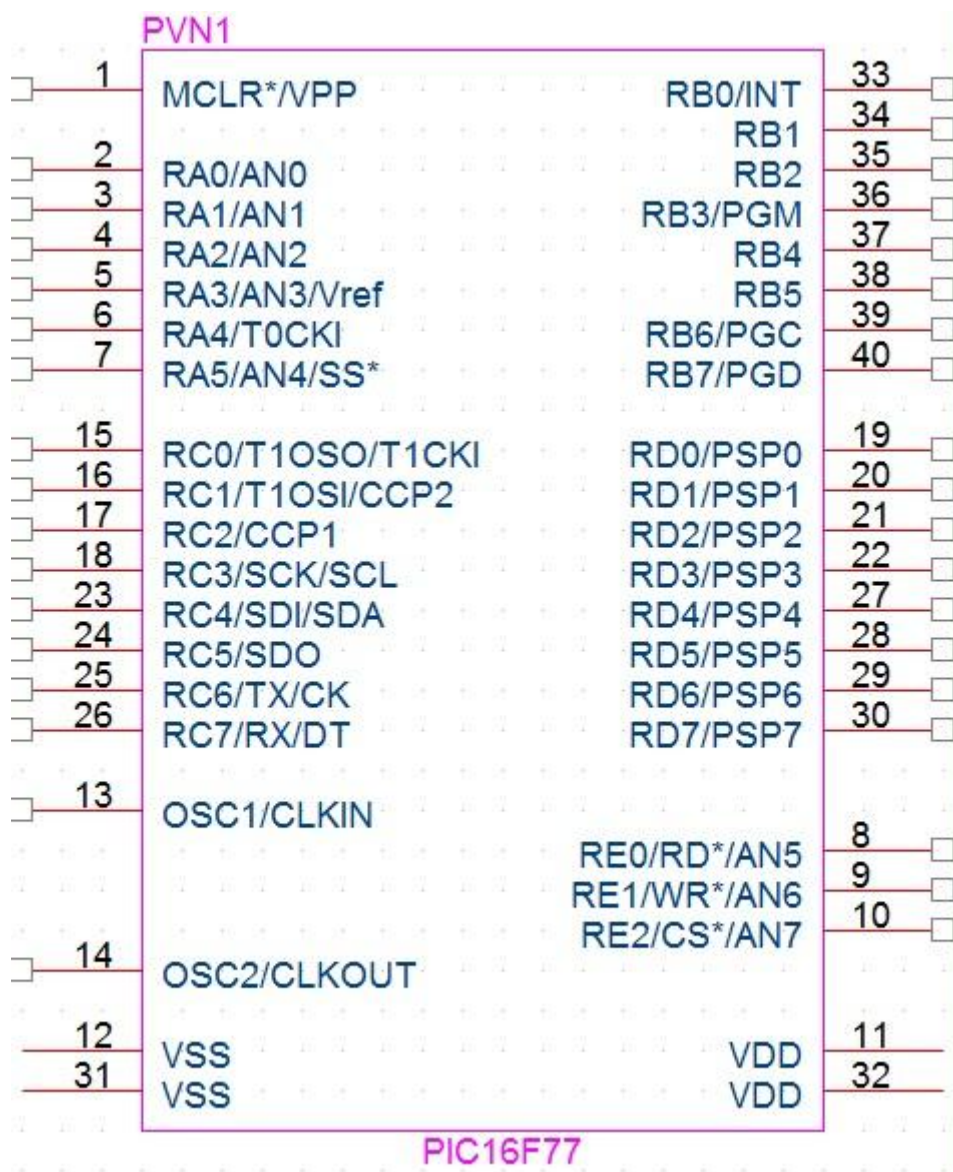
****các đặc tính nổi bật của thiết bị ngoại vi trên chip:**

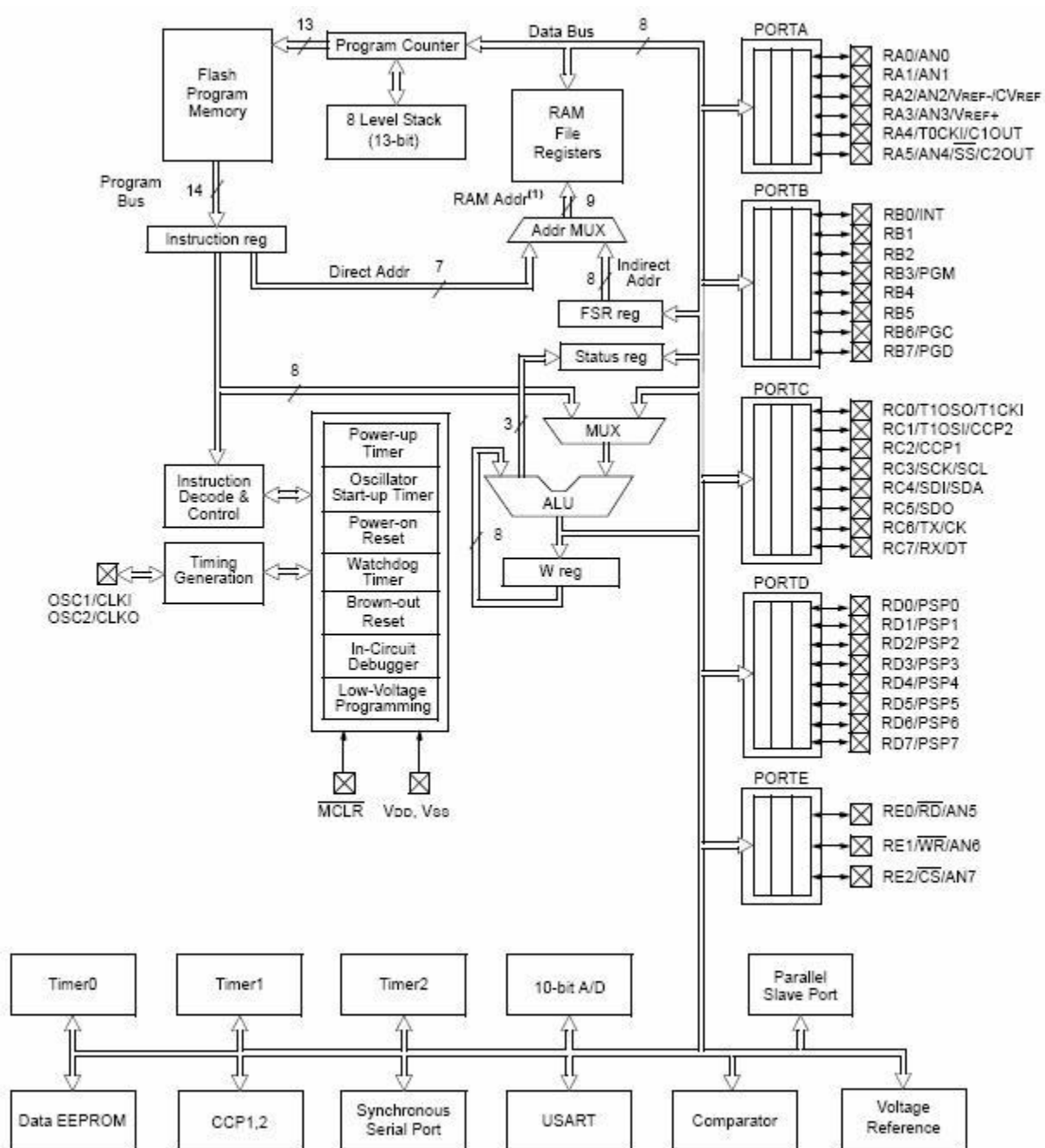
- +Timer 0: 8 bit của bộ định thời ,bộ đếm với hệ số tỉ lệ trước
- +Timer1 : 16 bit của bộ định thời ,bộ đếm với hệ số tỉ lệ trước,có khả năng trong chế độ SLEEP qua xung đồng bộ được cung cấp bên ngoài
- +Timer2: 8 bit của bộ định thời ,bộ đếm với 8bit hệ số tỉ lệ trước,hệ số tỉ lệ sau
- +Có 2 chế độ giữ ,so sánh ,điều chế độ rộng xung (PWM)
- +Chế độ giữ với 16 bit ,tốc độ 12.5ns,chế độ so sánh với 16bit,tốc độ giải quyết cực đại là 200ns chế độ điều chế độ rộng xung với 10 bit
- +Bộ chuyển đổi số sang tương tự với 10bit
- +Cổng truyền thông nối tiếp SSP với SPI phương thức chủ và I2C

- +Bộ truyền nhận thông tin đồng bộ ,dị bộ(USART/SCL) có khả năng phát hiện 9bit địa chỉ
- +Công phụ song song (PSP)với 8 bit mở rộng,với RD,WR và CS điều khiển

2.4.1.2 Mô tả sơ qua các chân của PIC 16F877A:

a. Sơ đồ chân của PIC 16F877A:



b. Sơ đồ khối của PIC16F877A

c. Các cổng xuất nhập của PIC16F877A**-PORT A**

PORTA (RPA) bao gồm 6 I/O

PORTA (RPA) bao gồm 6 I/O pin. Đây là các chân “hai chiều” (bidirectional pin), nghĩa là có thể xuất và nhập được. Chức năng I/O này được điều khiển bởi thanh ghi TRISA (địa chỉ 85h). Muốn xác lập chức năng của một chân trong PORTA là input, ta “set” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA và ngược lại, muốn xác lập chức năng của một chân trong PORTA là output, ta “clear” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA. Thao tác này hoàn toàn tương tự đối với các PORT và các thanh ghi điều khiển tương ứng TRIS (đối với PORTA là TRISA, đối với PORTB là TRISB, đối với PORTC là TRISC, đối với PORTD là TRISD và đối với PORTE là TRISE). Bên cạnh đó PORTA còn là ngõ ra của bộ ADC, bộ so sánh, ngõ vào analog ngõ vào xung clock của Timer0 và ngõ vào của bộ giao tiếp MSSP (Master Synchronous Serial Port).

-PORT B

PORTB (RPB) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISB.

Bên cạnh đó một số chân của PORTB còn được sử dụng trong quá trình nạp chương trình cho vi điều khiển với các chế độ nạp khác nhau. PORTB còn liên quan đến ngắt ngoại vi và bộ Timer0. PORTB còn được tích hợp chức năng điện trở kéo lên được điều khiển bởi chương trình.

Các thanh ghi SFR liên quan đến PORTB bao gồm:

PORTB (địa chỉ 06h, 106h) : chứa giá trị các pin trong PORTB

TRISB (địa chỉ 86h, 186h): điều khiển xuất nhập OPTION_REG (địa chỉ 81h, 181h) : điều khiển ngắt ngoại vi và bộ Timer0.

-PORT C

PORTC (RPC) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISC.

Bên cạnh đó PORTC còn chứa các chân chức năng của bộ so sánh, bộ Timer1, bộ PWM và các chuẩn giao tiếp nối tiếp I2C, SPI, SSP, USART.

Các thanh ghi điều khiển liên quan đến PORT:

- + PORTC (địa chỉ 07h) : chứa giá trị các pin trong PORTC
- + TRISC (địa chỉ 87h) : điều khiển xuất nhập

-PORT D

PORTD (RPD) gồm 8 chân I/O, thanh ghi điều khiển xuất nhập tương ứng là TRISD.

PORTD còn là cổng xuất dữ liệu của chuẩn giao tiếp PSP (Parallel Slave Port).

Các thanh ghi liên quan đến PORTD bao gồm:

- +Thanh ghi PORTD : chứa giá trị các pin trong PORTD.
- +Thanh ghi TRISD : điều khiển xuất nhập.
- +Thanh ghi TRISE : điều khiển xuất nhập PORTE và chuẩn giao tiếp PSP.

-PORT E

PORTE (RPE) gồm 3 chân I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISE.

Các chân của PORTE có ngõ vào analog. Bên cạnh đó PORTE còn là các chân điều khiển của chuẩn giao tiếp PSP.

Các thanh ghi liên quan đến PORTE bao gồm:

- +PORTE : chứa giá trị các chân trong PORTE.
- +TRISE : điều khiển xuất nhập và xác lập các thông số cho chuẩn giao tiếp PSP.

+ADCON1 : thanh ghi điều khiển khối ADC.

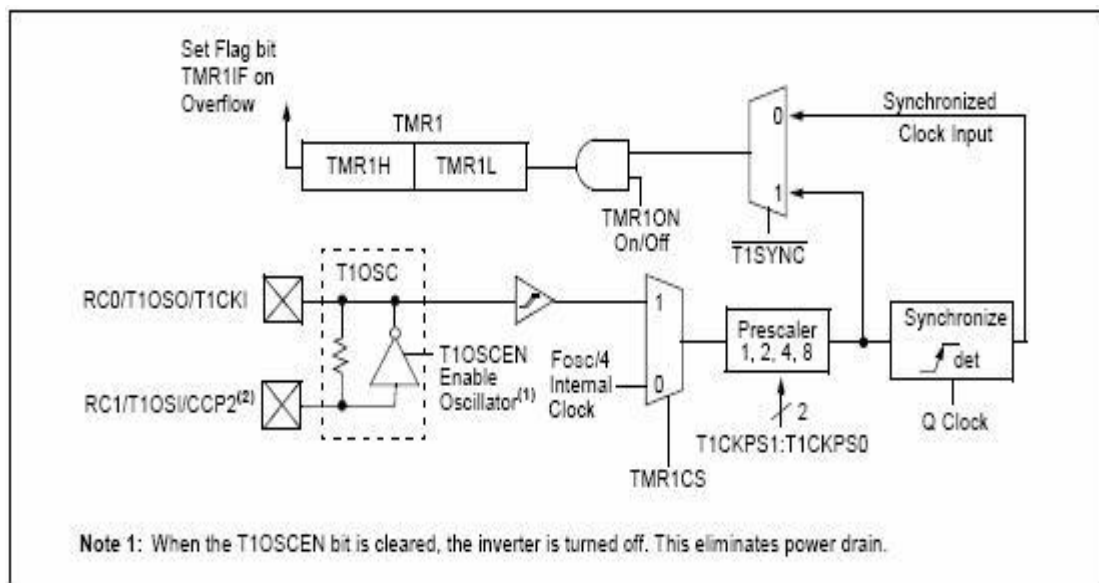
d. Các bộ Timer của PIC 16F877A

-Timer 0

Đây là một trong ba bộ đếm hoặc bộ định thời của vi điều khiển PIC16F877A. Timer0 là bộ đếm 8 bit được kết nối với bộ chia tần số (prescaler) 8 bit. Cấu trúc của Timer0 cho phép ta lựa chọn xung clock tác động và cạnh tích cực của xung clock. Ngắt Timer0 sẽ xuất hiện khi Timer0 bị tràn. Bit TMR0IE (INTCON<5>) là bit điều khiển của Timer0.

TMR0IE=1 cho phép ngắt Timer0 tác động, TMR0IF= 0 không cho phép ngắt Timer0 tác động.

Sơ đồ khối của Timer 0

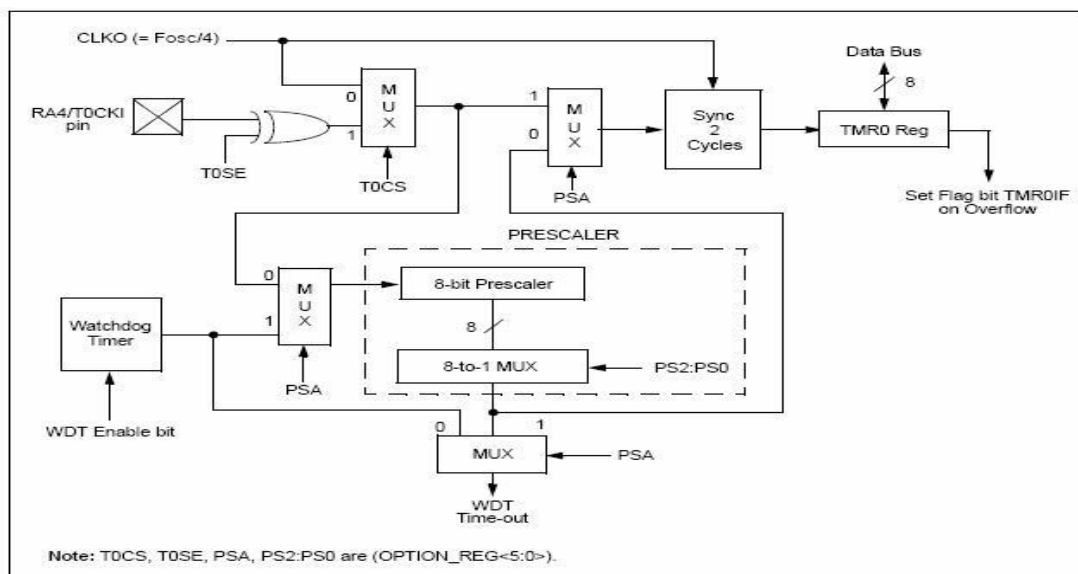


-Timer1

Sơ đồ khối của Timer 1:

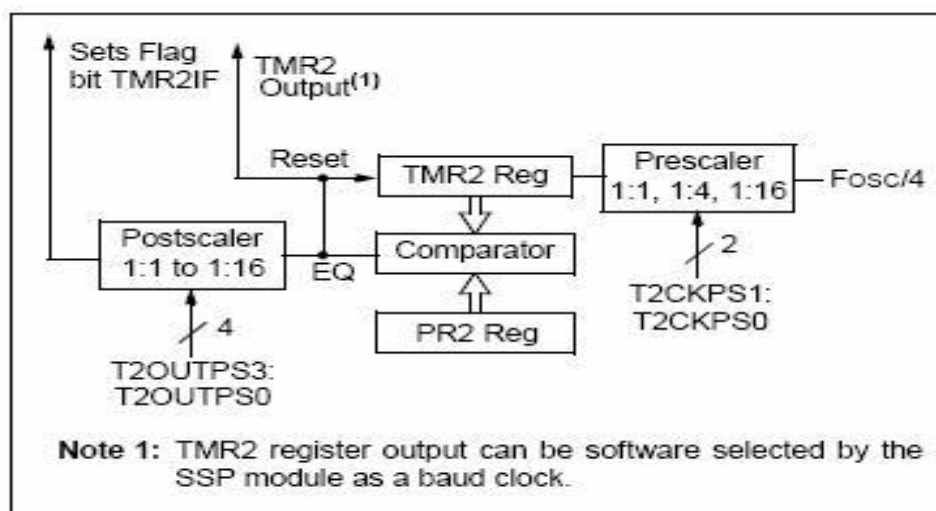
Timer1 là bộ định thời 16 bit, giá trị của Timer1 sẽ được lưu trong hai thanh ghi (TMR1H:TMR1L). Cờ ngắt của Timer1 là bit TMR1IF (PIR1<0>). Bit điều khiển của Timer1 sẽ là TMR1IE (PIE<0>). Tương

Sơ đồ khối của Timer 1:



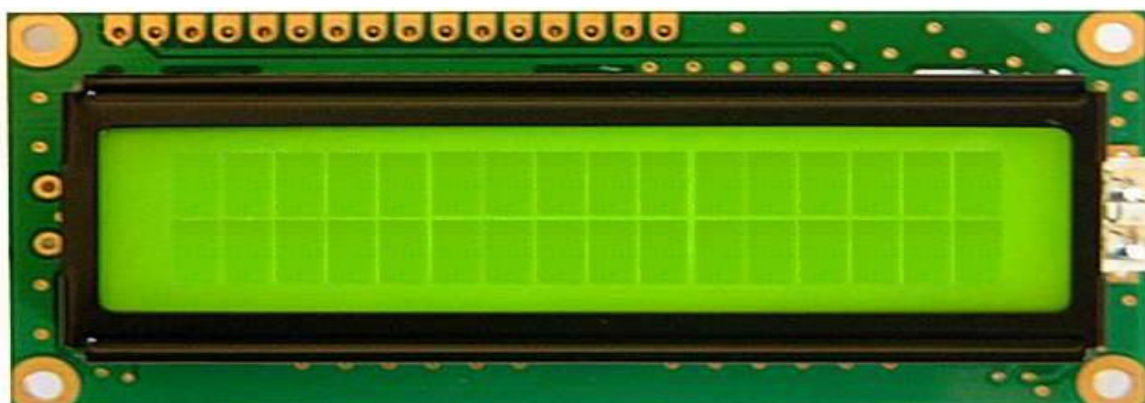
Timer2 là bộ định thời 8 bit và được hỗ trợ bởi hai bộ chia tần số prescaler và postscaler. Thanh ghi chứa giá trị đếm của Timer2 là TMR2. Bit cho phép ngắt Timer2 tác động là TMR2ON (T2CON<2>). Cờ ngắt của Timer2 là bit TMR2IF (PIR1<1>). Xung ngõ vào (tần số bằng ¼ tần số oscillator) được đưa qua bộ chia tần số prescaler 4 bit (với các tỉ số chia tần số là 1:1, 1:4 hoặc 1:16 và được điều khiển bởi các bit T2CKPS1:T2CKPS0 (T2CON<1:0>).

Sơ đồ khối của Timer 2:

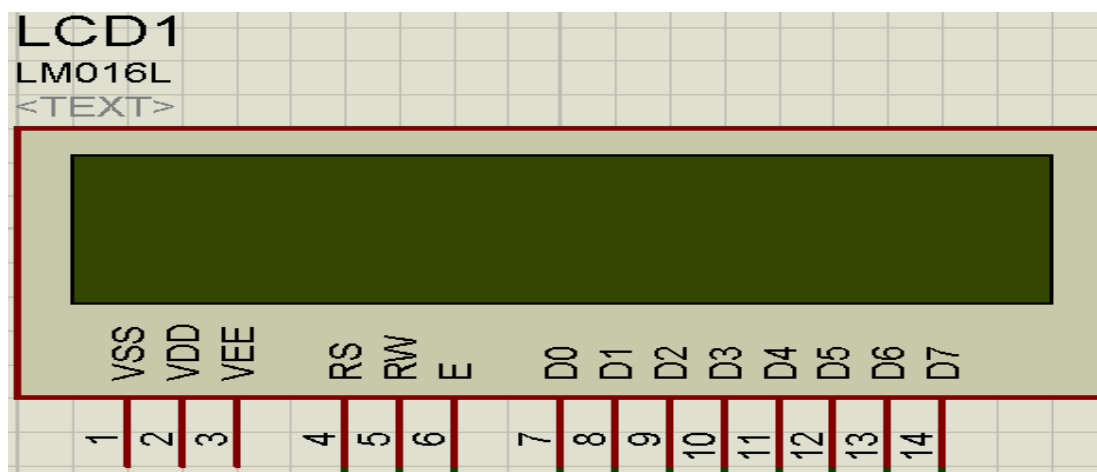


2.4.2. LCD

2.4.2.1. Hình ảnh thực tế



2.4.2.2. Sơ đồ chân LM016L:

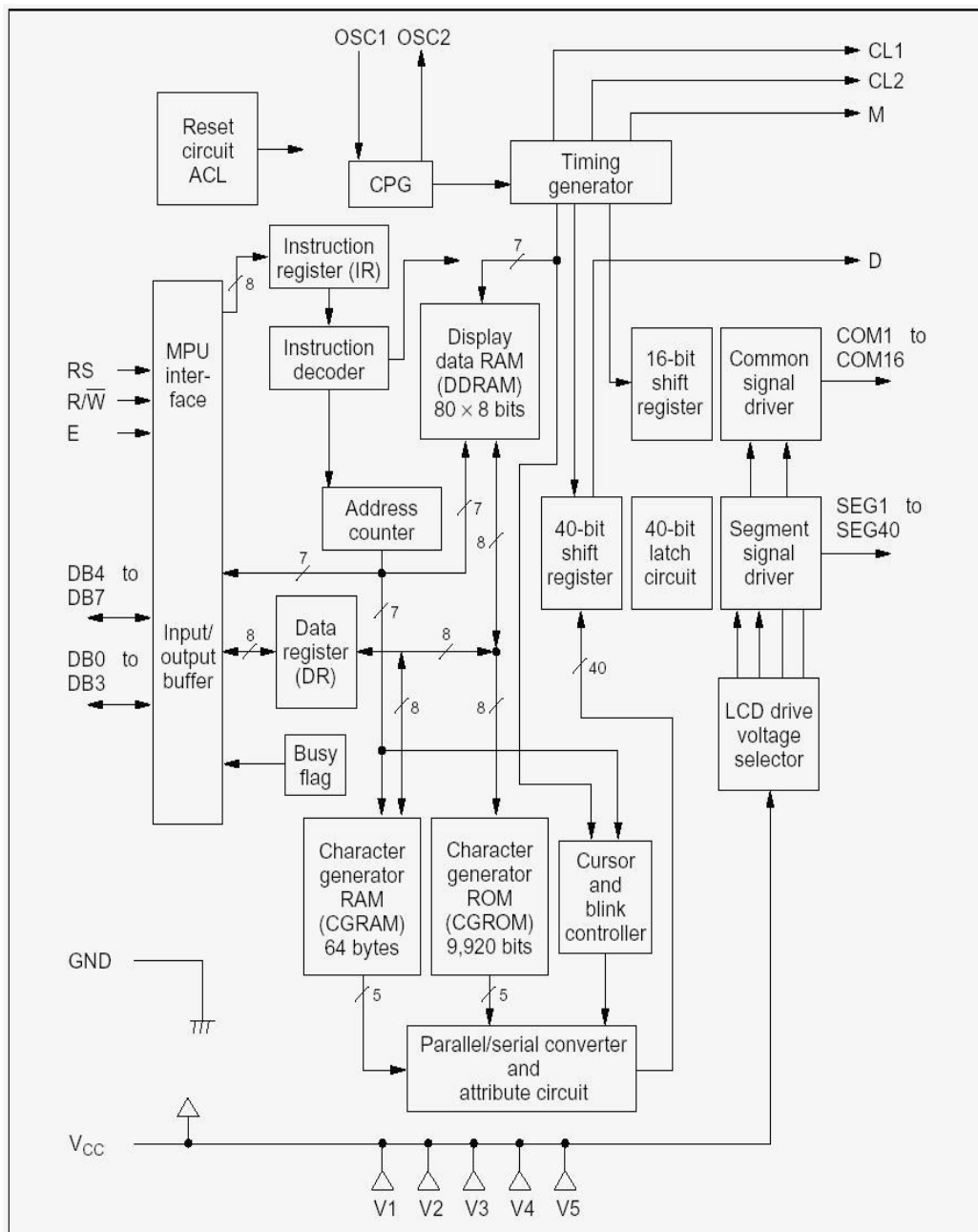


2.4.2.3. Chức năng các chân của LCD

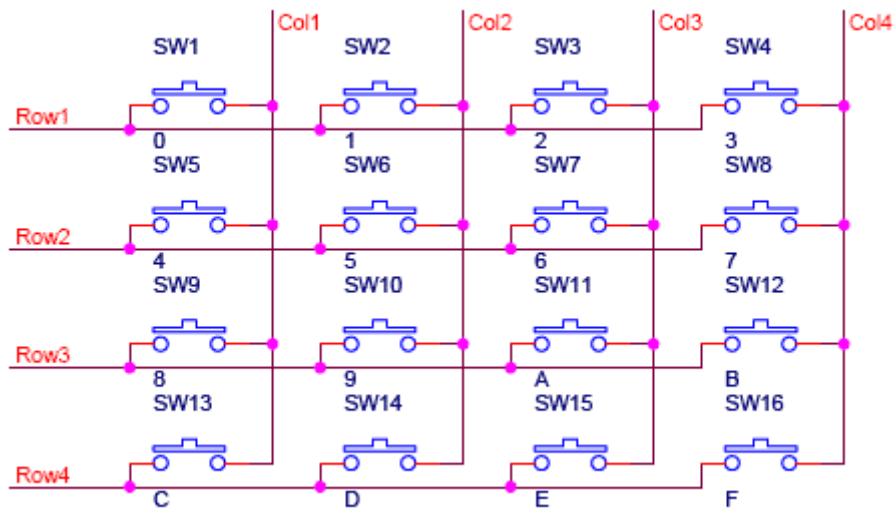
Số chân	Tên	Chức năng
1	GND	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VCC	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển
3	Vee	Chân này dùng để điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). 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 (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh 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	<p>Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :</p> <ul style="list-style-type: none"> + 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.
------	---------	---

2.4.2.4. Sơ đồ khối của LCD LM 016L

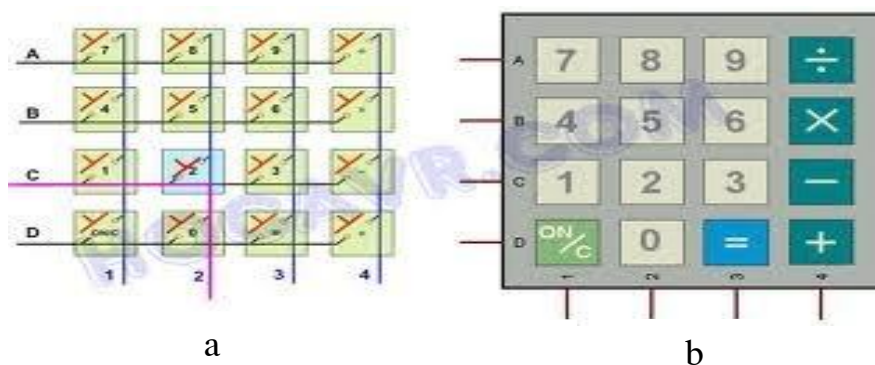


2.4.3. KEYPAD 4x4



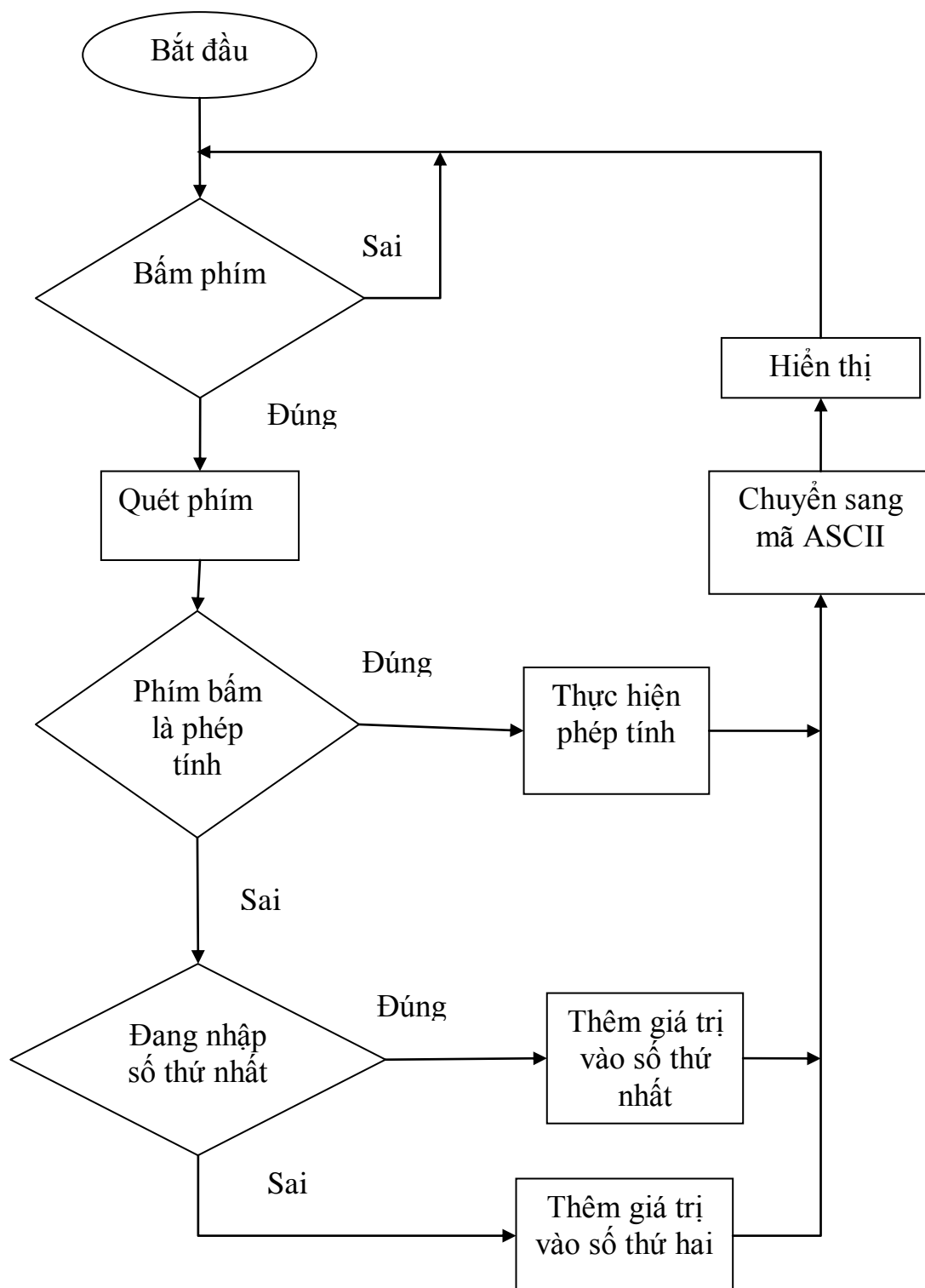
KEYPAD là một "thiết bị nhập" chứa các nút nhấn cho phép người dùng nhập các chữ số, chữ cái hoặc ký hiệu vào bộ điều khiển. KEYPAD không chứa tất cả bảng mã ASCII như keyboard và vì thế KEYPAD thường được tìm thấy trong các thiết bị chuyên dụng. Các nút nhấn trên các máy tính điện tử cầm tay là một ví dụ về KEYPAD. Số lượng nút nhấn của một KEYPAD thay đổi phụ thuộc vào yêu cầu ứng dụng. Trong bài này tôi giới thiệu cách điều khiển của một loại KEYPAD đơn giản, KEYPAD 4x4.

Gọi là KEYPAD 4x4 vì KEYPAD này có 16 nút nhấn được bố trí dạng ma trận 4 hàng và 4 cột. Cách bố trí ma trận hàng và cột là cách chung mà các KEYPAD sử dụng. Cũng giống như các ma trận LED, các nút nhấn cùng hàng và cùng cột được nối với nhau, vì thế với KEYPAD 4x4 sẽ có tổng cộng 8 ngõ ra (4 hàng và 4 cột). Mô hình KEYPAD 4x4 được thể hiện trong hình 1.



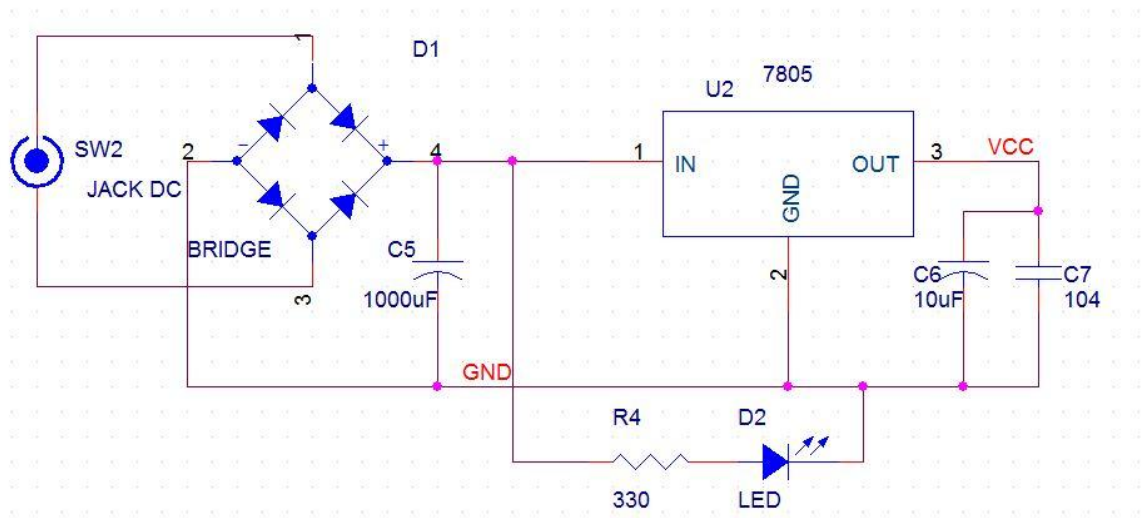
Hình 1 là mô hình thật của 1 KEYPAD 4x4 và hình a là cấu hình bên trong của nó. Bốn hàng của KEYPAD được đánh dấu là A, B, C và D trong khi 4 cột được gọi là 1, 2, 3 và 4.

Hoạt động của KEYPAD: Giả sử nút '2' được nhấn, khi đó đường C và 2 được nối với nhau. Giả sử đường 2 được nối với GND (mass, 0V) thì C cũng sẽ là GND. Tuy nhiên, câu hỏi đặt ra là bằng cách kiểm tra trạng thái đường C chúng ta sẽ có kết luận nút '2' được nhấn? Giả sử tất cả các đường 1, 2, 3, 4 đều nối với GND, nếu C= GND thì rõ ràng chúng ta không thể kết luận nút '1',= hay nút '2' hay nút '3' hay nút '-' được nhấn. Kỹ thuật để khắc phục vấn đề này chính là kỹ thuật "quét" KEYPAD.

3.1.2. Sơ đồ thuật toán tổng quát.

3.2. Sơ đồ đặc tả :

3.2.1. Khối nguồn :

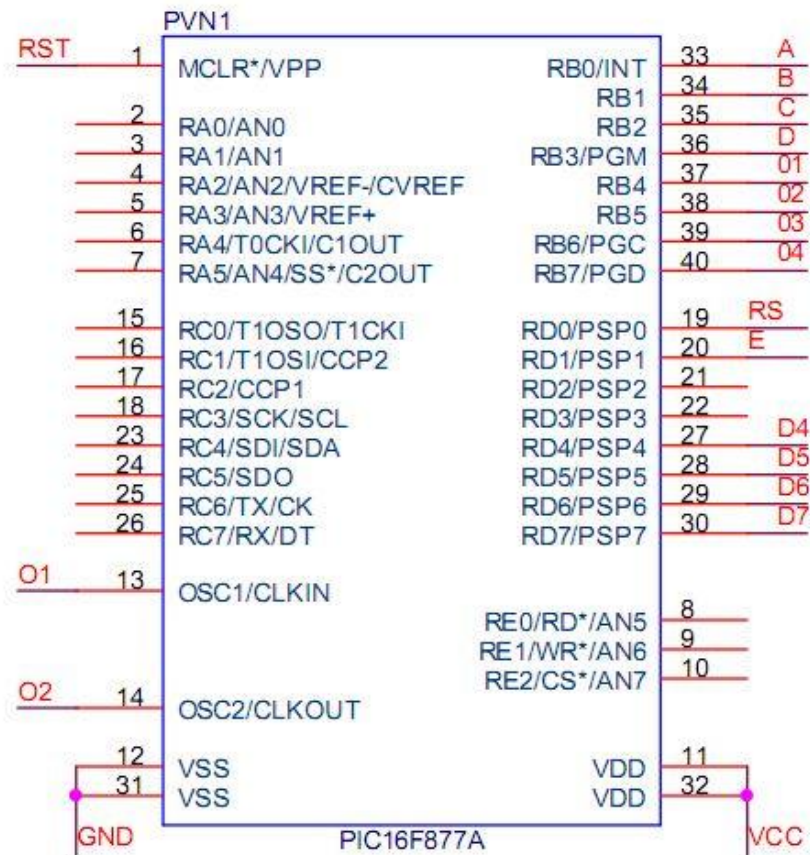


Đây là mạch dùng để tạo ra nguồn điện áp chuẩn +5V. Sử dụng IC7805.

Đầu vào qua JACK cắm DC là điện áp xoay chiều sau khi được biến đổi qua máy biến áp đưa vào bộ chỉnh lưu cầu không điều khiển để cho ra dòng điện một chiều (lúc này điện áp nằm trong khoảng từ 7->10V). Sau khi đi qua IC ổn áp 7805 sẽ tạo ra nguồn điện áp chuẩn +5V cung cấp cho mạch.

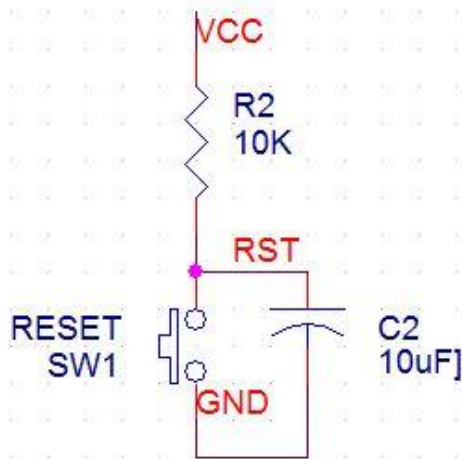
IC ổn áp 7805: Điện áp đầu vào > 7V, điện áp đầu ra 5V, 500 mA. Mạch ổn áp: Cần cho vi điều khiển vì nếu nguồn cho vi điều khiển không ổn định thì sẽ treo VDK, không chạy đúng hoặc reset liên tục thậm chí dẫn đến cháy VDK.

3.2.2. Khối xử lý trung tâm.



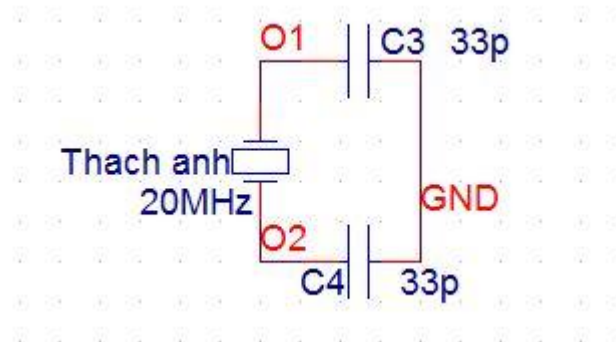
Khối xử lý trung tâm là nơi thực thi chương trình điều khiển mà ta nạp vào cho nó. Xử lý các thông tin nhập vào từ bàn phím, tính toán và xuất tín hiệu ra các chân RD4 đến RD7 để hiển thị ra LCD.

3.2.3. Khối Reset



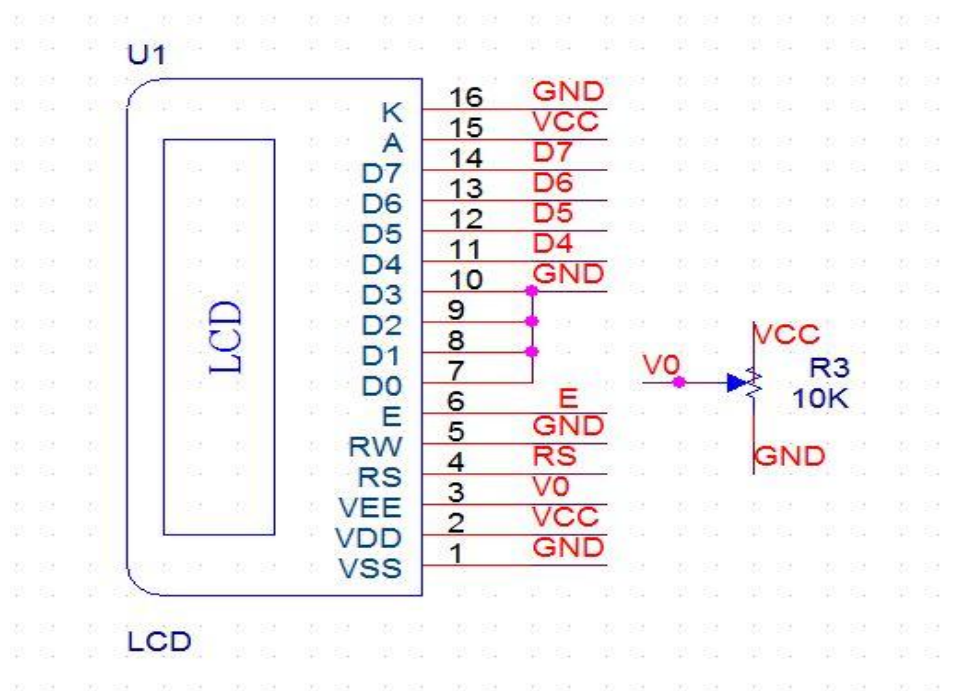
Khởi RESET có tác dụng đưa vi điều khiển về trạng thái ban đầu. Khi nút Reset được ấn điện áp +5V từ nguồn được nối vào chân Reset của vi điều khiển được chạy thẳng xuống đất lúc này điện áp tại chân vi điều khiển thay đổi đột ngột về 0, VĐK nhận biết được sự thay đổi này và khởi động lại trạng thái ban đầu cho hệ thống.

3.2.4. Khởi tạo xung dao động ngoài



Đây là bộ dao động thạch anh cấu tạo chính là thạch anh 20MHz có tác dụng tạo xung nhịp với tần số 20MHz cho PIC hoạt động làm cho mạch chạy ổn định hơn. Hai đầu O1 và O2 được nối vào 2 chân OSC1 và OSC2 của PIC.

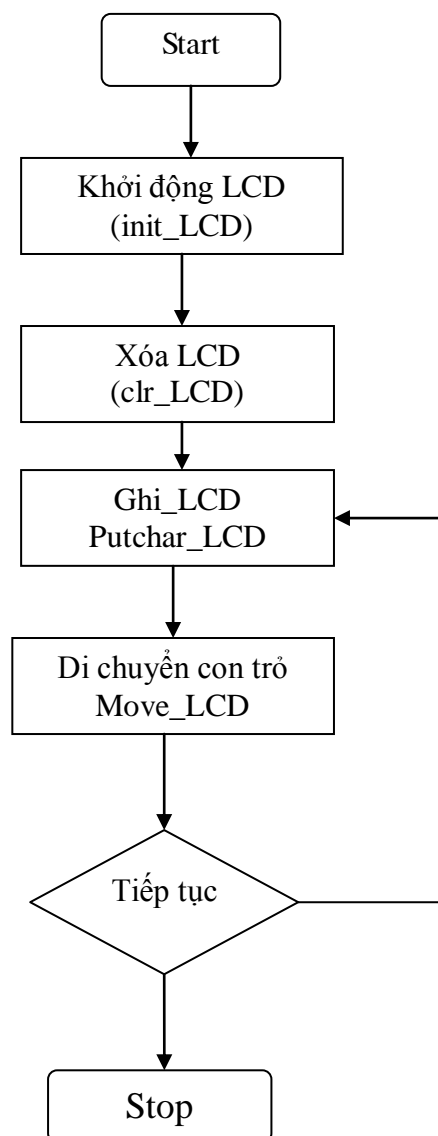
3.2.5. Khởi hiển thị



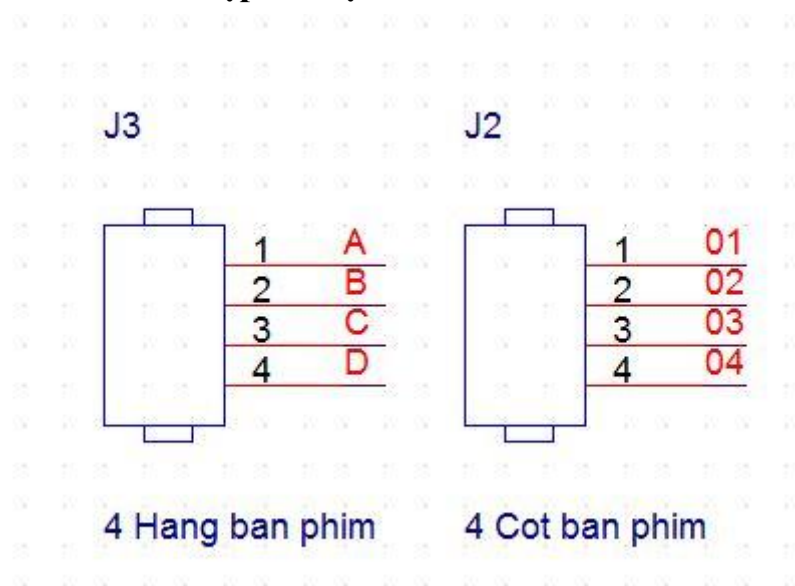
Khởi hiển thị là LCD 16x2 được dùng để hiển thị kết quả, được sử dụng ở chế độ 4bit vì vậy mà 4 bit cao được nối với PIC để nhận tín hiệu, còn 4 bit thấp được nối đất để chống nhiễu.

Chân V0 của biến trở được nối với chân Vee của LCD có tác dụng điều chỉnh độ tương phản.

Sơ đồ thuật toán LCD



3.2.6. Khối nhập dữ liệu



Khối nhập dữ liệu gồm các đầu ra bao gồm 4 đầu ra hàng của bàn phím là A B C D tương ứng với các chân RB0 đến RB3 của PIC và 4 đầu ra cột của bàn phím 01 02 03 04 tương ứng với các chân RB4 đến RB7 của PIC.

3.3. Chương trình

```
#include <16f877a.h>
#include <delay(clock=20M)>
#include <def_877a.h>
#include <lcd_4bit.c>
#include <math.h>
```

```
#bit cot1=portB.4
#bit cot2=portB.5
#bit cot3=portB.6
#bit cot4=portB.7
```

```
#bit hang1=portB.0
#bit hang2=portB.1
#bit hang3=portB.2
```

```
#bit hang4=portB.3
```

```
long kq,kq1,kq2,ok,pt,j;
```

```
//-----  
//-----Chương trình con-----  
//-----
```

```
int8 const tit0[]={ "*DO AN HTN*" };  
int8 const tit1[]={ "-T-T-V-" };  
int8 tit2[15];  
int8 tit3[15];  
int8 i;
```

```
void write_line1(long kq,pt){  
    lcd_putcmd(clear_scr);  
    lcd_putcmd(line_1);  
    i=0;j=kq;  
    while (j!=0)  
    {  
        if (j%10) tit2[i]=1;  
        else    tit2[i]=0;  
        j=j/10;  
        i++;  
    }  
    for (j=i-1;j>=0;j--) lcd_putchar(tit2[j]);  
    if (pt==1) lcd_putchar('+');  
    else  
        if (pt==2) lcd_putchar('-');  
    else  
        if (pt==3) lcd_putchar('x');  
    else  
        if (pt==4) lcd_putchar(':');
```

```
}  
void write_line2(long kq){  
    lcd_putcmd(line_2);  
    i=0;j=kq;  
    while (j!=0)  
    {  
        tit2[i]=j%10;  
        j=j/10;  
        i++;  
    }  
    for (j=i-1;j>=0;j--) lcd_putchar(tit2[j]);  
}  
  
#int_RB // Chuong trinh ngat Port B  
void RB_isr(void){  
    long kq=0, kq1=0, kq2=0;  
    if(hang1==1){  
        if(cot1==1){  
            if (ok==0)  
            {    kq1=kq1*10+7;write_line2(kq1); }  
            else {kq2=kq2*10+7;write_line1(kq1,pt);write_line2(kq2); }  
        }  
        else if(cot2==1){lcd_putcmd(clear_scr);  
            if (ok==0)  
            {    kq1=kq1*10+8;write_line2(kq1);}  
            else {kq2=kq2*10+8;write_line1(kq1,pt);write_line2(kq2);}  
        }  
        else if(cot3==1){lcd_putcmd(clear_scr);  
            if (ok==0)  
            {    kq1=kq1*10+9;write_line2(kq1);}  
            else {kq2=kq2*10+9;write_line1(kq1,pt);write_line2(kq2);}  
        }  
        else if(cot4==1){lcd_putcmd(clear_scr);
```

```
        if (ok==0)
        {
            pt=4;
            ok=1;
        }
        else
        {
            switch (pt)
            {
                case 1: kq=(kq1+kq2); kq1=kq; break;
                case 2: kq=kq1-kq2; kq1=kq; break;
                case 3: kq=kq1*kq2; kq1=kq; break;
                case 4: kq=kq1/kq2; kq1=kq; break;
            }
            pt=4;
        }
        write_line1(kq1,pt);
    }
}
else if(hang2==1){
    if(cot1==1){

        if (ok==0)
        {
            kq1=kq1*10+4;write_line2(kq1);}
        else {kq2=kq2*10+4;write_line2(kq1);}
    }
    else if(cot2==1){lcd_putcmd(clear_scr);
        if (ok==0)
        {
            kq1=kq1*10+5;write_line2(kq1);}
        else {kq2=kq2*10+5;write_line1(kq1,pt);write_line2(kq2);}
    }
    else if(cot3==1){lcd_putcmd(clear_scr);
        if (ok==0)
```

```
        {    kq1=kq1*10+6;write_line2(kq1);}
        else {kq2=kq2*10+6;write_line1(kq1,pt);write_line2(kq2);}
        }
    else if(cot4==1){lcd_putcmd(clear_scr);
        if (ok==0)
    {
        pt=3;
        ok=1;
        }
    else
    {
        switch (pt)
        {
            case 1: kq=(kq1+kq2); kq1=kq; break;
            case 2: kq=kq1-kq2; kq1=kq; break;
            case 3: kq=kq1*kq2; kq1=kq; break;
            case 4: kq=kq1/kq2; kq1=kq; break;
        }
        pt=3;
        }
        write_line1(kq1,pt);
    }
}
else if(hang3==1){
    if(cot1==1){lcd_putcmd(clear_scr);
        if (ok==0)
        {    kq1=kq1*10+1;write_line2(kq1);}
        else {kq2=kq2*10+1;write_line1(kq1,pt);write_line2(kq1);}
        }
    else if(cot2==1){lcd_putcmd(clear_scr);
        if (ok==0)
        {    kq1=kq1*10+2;write_line2(kq1);}
        else {kq2=kq2*10+2;write_line1(kq1,pt);write_line2(kq1);}
```

```
    }
    else if(cot3==1){lcd_putcmd(clear_scr);
if (ok==0)
{   kq1=kq1*10+3;write_line2(kq1);}
else {kq2=kq2*10+3;write_line1(kq1,pt);write_line2(kq1);}
    }
    else if(cot4==1){
        lcd_putcmd(clear_scr);
        if (ok==0)
        {
            pt=2;
            ok=1;
        }
        else
        {
            switch (pt)
            {
                case 1: kq=(kq1+kq2); kq1=kq; break;
                case 2: kq=kq1-kq2; kq1=kq; break;
                case 3: kq=kq1*kq2; kq1=kq; break;
                case 4: kq=kq1/kq2; kq1=kq; break;
            }
            pt=2;
        }
        write_line1(kq1,pt);
    }
}
else if(hang4==1){
    if(cot1==1){lcd_putcmd(clear_scr);
        if (ok==0)
        {   kq1=0;write_line2(kq1);}
        else {kq2=0;write_line1(kq1,pt);write_line2(kq1);}
    }
}
```



```
else if(cot2==1){lcd_putcmd(clear_scr);
    if (ok==0)
    {    kq1=kq1*10+0;write_line2(kq1);}
    else {kq2=kq2*10+0;write_line1(kq1,pt);write_line2(kq1);}
}
else if(cot3==1){lcd_putcmd(clear_scr);
    if (ok==0) kq=kq1;
    else
    {
        switch (pt)
        {
            case 1: kq=(kq1+kq2); kq1=kq; break;
            case 2: kq=kq1-kq2; kq1=kq; break;
            case 3: kq=kq1*kq2; kq1=kq; break;
            case 4: kq=kq1/kq2; kq1=kq; break;
        }
    }
    kq1=0;kq2=0;pt=0;ok=0;
    write_line2(kq);
}
else if(cot4==1){lcd_putcmd(clear_scr);
    if (ok==0)
    {
        pt=2;
        ok=1;
    }
    else
    {
        switch (pt)
        {
            case 1: kq=(kq1+kq2); kq1=kq; break;
            case 2: kq=kq1-kq2; kq1=kq; break;
            case 3: kq=kq1*kq2; kq1=kq; break;
```

```
        case 4: kq=kq1/kq2; kq1=kq; break;
        }
        pt=1;
    }
    write_line1(kq1,pt);
}

}

}

//-----
/*-----Chương trình chính-----*/
//-----

void main(){
    lcd_init();
    enable_interrupts(INT_RB);
    //cho phép ngắt (GLOBAL);

    trisb=0xF0;

    while(tit0[i]!=0){
        lcd_putchar(tit0[i]);
        i++;
        delay_ms(70);
    }
    lcd_putcmd(line_2);
    i=0;ok=0;pt=0;

    while(tit1[i]!=0){
        lcd_putchar(tit1[i]);
        i++;
        delay_ms(70);
    }
    delay_ms(2000);
    write_line1(1000,1);delay_ms(2000);
```

```
lcd_putcmd(clear_scr);

enable_interrupts(GLOBAL);
while(1){
    //quét hàng
    hang4=0; hang1=1; delay_ms(10);
        //Xuất tin hiệu 1 ra hàng 1, 0 ra hàng 4 để đo hàng,
    hang1=0; hang2=1; delay_ms(10);
        //Xuất tin hiệu 1 ra hàng 2, 0 ra hàng 1 để đo hàng
    hang2=0; hang3=1; delay_ms(10);
        //Xuất tin hiệu 1 ra hàng 3, 0 ra hàng 2 để đo hàng
    hang3=0; hang4=1; delay_ms(10);
        //Xuất tin hiệu 1 ra hàng 4, 0 ra hàng 3 để đo hàng
    }
}
```

TÀI LIỆU THAM KHẢO

- [1] Bài giảng Hệ Thống Nhúng – Bộ Môn Kỹ Thuật Máy Tính – ĐH Kỹ Thuật Công Nghiệp Thái Nguyên.
- [2] Giáo trình vi xử lý PIC.
- [3] “Kỹ Thuật Vi Xử Lý” tác giả Văn Thế Minh.
- [4] Nguồn tài liệu từ Internet :
WWW.PICVIETNAM.COM
WWW.WEBDIEN.COM
WWW.DIENTUVIETNAM.NET
Và một số tài liệu khác nữa.