

NỘI DUNG THIẾT KẾ

Chương I : Tổng quan về hệ thống hiển thị thời gian thực

Chương II: Thiết kế và thi công phần cứng hệ thống hiển thị thời gian thực.

Chương III: Thiết kế chương trình phần mềm

.

Lời nói đầu

Ngày nay nhân loại đang trải qua những sự phát triển vượt về mọi mặt. Trong đó điện tử, tự động hoá đóng một vai trò không nhỏ. Điện tử góp phần vào quá trình tự động hoá mọi thứ giúp con người hiện đại hoá cuộc sống.

Sau hơn ba tháng không ngừng nghiên cứu, học hỏi, với đề tài được giao là: **“Thiết kế hệ thống hiển thị thời gian thực.”** Em đã hoàn thành. Đề tài được chia làm ba chương như sau:

Chương 1: Tổng quan về hệ thống hiển thị thời gian thực

Chương 2: Thiết kế và thi công hệ thống hiển thị thời gian thực

Chương 3: Thiết kế phần mềm cho hệ thống hiển thị thời gian thực

Để hoàn thành được đề án tốt nghiệp này, trước tiên em xin bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy cô Khoa Điện - Điện tử – trường Đại học Dân lập Hải Phòng đã hết lòng hướng dẫn, truyền đạt những kiến thức và kinh nghiệm cho em trong những năm tháng học tập.

Đặc biệt em xin cảm ơn thầy giáo Nguyễn Trọng Thắng đã luôn quan tâm, hướng dẫn, động viên và giúp đỡ em trong suốt quá trình thực hiện đề án tốt nghiệp này.

Sau cùng xin cảm ơn những người thân, bạn bè đã giúp đỡ để đề án này được hoàn thành tốt đẹp. Tuy nhiên do lần đầu tiên thiết kế hệ thống vi điều khiển, trình độ còn hạn chế, nắm bắt thông tin chưa kịp thời nên không tránh khỏi những sai sót. Em rất mong nhận được sự thông cảm, góp ý và chỉ dẫn thêm của thầy cô cùng các bạn

Em xin chân thành cảm ơn!

Hải phòng, tháng 7 năm 2010

Sinh viên thực hiện:

Vũ Hải Đăng

MỤC LỤC

Lời nói đầu	2
Mục lục	3
Chương 1: Giới Thiệu Chung	5
1.1 Sơ lược về vi điều khiển	5
1.2 Sơ đồ khối hệ thống hiển thị thời gian thực.....	5
Chương 2: Thiết kế và thi công phần cứng hệ thống hiển thị thời gian thực ...	6
2.1 Giới thiệu linh kiện	6
2.1.1. Khảo sát họ vi điều khiển 8051:	7
2.1.1.1 cấu trúc bên trong của 8051:.....	7
2.1.1.2.Chức năng các chân điều khiển	8
2.1.1.3. Các thanh ghi đặc biệt.....	9
2.1.1.4. Khối tạo thời gian và bộ đếm (Timer/Counter).	12
2.1.1.5. Cơ chế ngắt trong On-chip 8051:	20
2.1.1.6. Bảo vệ chương trình.....	24
2.1.1.7.Tra cứu nhanh tập lệnh	30
2.1.2 IC tạo nguồn ổn áp chuẩn 7805.	31
2.1.3 IC tạo thời gian thực DS1307.	38
2.1.4 IC Đệm ULN 2803.....	47
2.1.5 IC ghi dịch 74HC595	48
2.1.6 LED 7thanh.....	52
2.2 :Các khối mạch cần sử dụng và sơ đồ mạch hoàn chỉnh	53
2.2.1 Khối tạo nguồn.....	53

2.2.2	Khởi Reset.....	54
2.2.3	Khởi điều khiển.....	54
2.2.4	Khởi tạo xung dao động.....	55
2.2.5	Khởi hiển thị	56
2.2.6	Khởi tạo thời gian thực	56
2.3:	Sơ đồ mạch hoàn chỉnh.....	57
2.3.1	Sơ đồ mạch nguyên lý	57
2.3.2	Sơ đồ mạch in	58
Chương 3: Chương trình phần mềm		59
3.1	Sơ đồ thuật toán	59
3.2	Code chương trình cho AT89C51	60
Kết luận		74
Tài liệu tham khảo		75

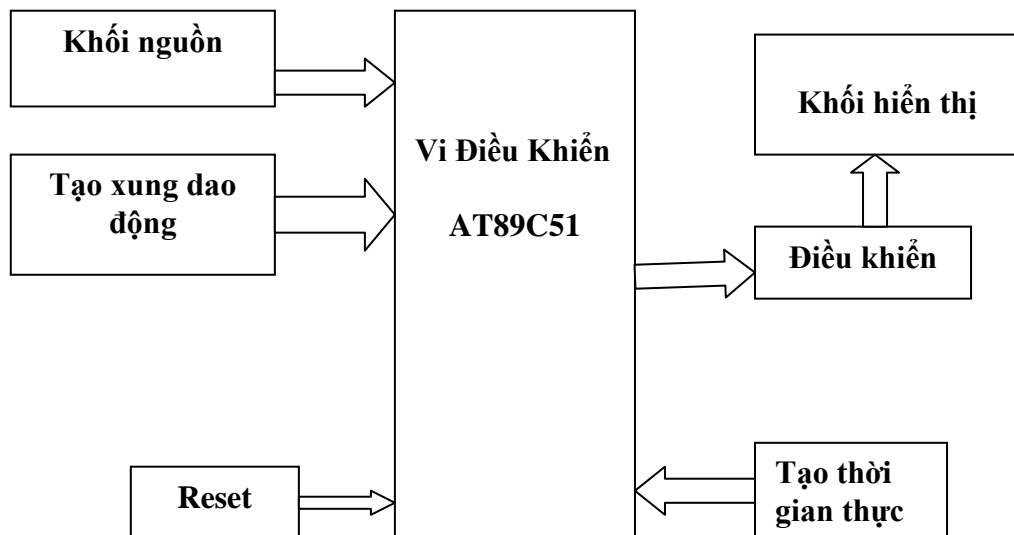
Chương I : GIỚI THIỆU CHUNG

1) *Sơ Lược về vi điều khiển:*

Trong công nghệ điện tử vi xử lý, vi điều khiển là một thành phần quan trọng không thể thiếu nó mang nhiều tính ưu việt: có thể thay thế một mạch điện phức tạp bằng một vi mạch nhỏ gọn với chi phí thấp hơn, nhưng ứng dụng lại đa dạng và linh hoạt hơn, tiết kiệm năng lượng hơn, tốc độ xử lý nhanh hơn,...

Để học tập tốt và hiểu sâu về môn học vi xử lý ngoài những kiến thức trên sách vở cần có những ứng dụng vào thực tế. Trên cơ sở đó em tìm hiểu và thiết kế sản phẩm là **HỆ THỐNG HIỂN THỊ THỜI GIAN THỰC** dùng vi điều khiển AT89C51. Có khả năng điều chỉnh và thay đổi được thời gian. So với những mạch đồng hồ dùng vi điều khiển AVR và PIC thì 8051 có ưu điểm hơn là đơn giản, rẻ, thông dụng, tập lệnh dễ và tường minh nhất.

2) *Sơ đồ khối hệ thống hiển thị thời gian thực:*



Chương II: THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG HỆ THỐNG HIỆN THỊ THỜI GIAN THỰC

2.1 giới thiệu linh kiện

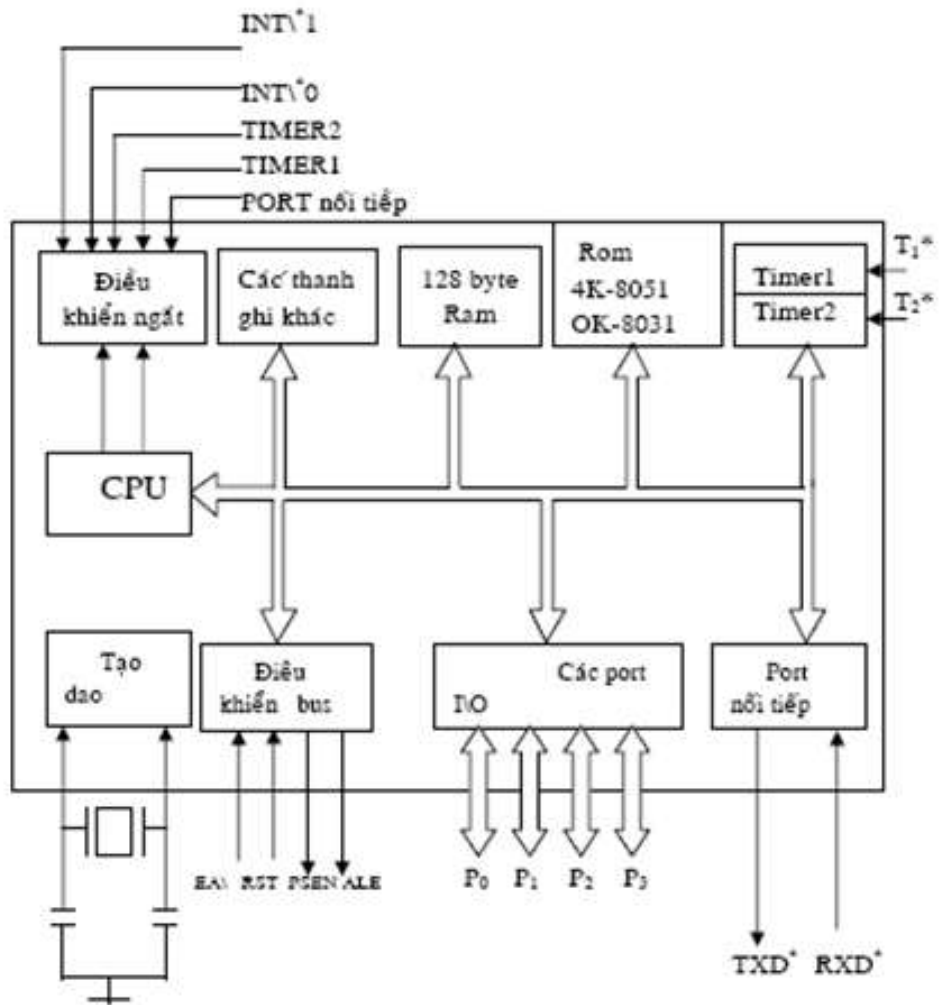
2.1.1 Khảo sát họ vi điều khiển 8051

*** Giới thiệu chung vi điều khiển 8051**

IC vi điều khiển 8051 thuộc họ MCS51 có đặc điểm sau:

- 4kb ROM(được lập trình bởi nhà sản xuất chỉ có ở 8051)
- 128 byte Ram
- 4 port I/O 8 bit
- 2 bộ định thời 16 bit
- Giao tiếp nối tiếp
- 64 kb không gian bộ nhớ chương trình mở rộng
- 64 kb không gian bộ nhớ dữ liệu mở rộng
- 1 bộ xử lý luận lí (thao tác trên các bit đơn)
- 210 bit nhận địa chỉ hóa
- bộ nhân / chia $4 \mu s$

2.1.1.1 cấu trúc bên trong của 8051:



Hình 2.1 : Sơ đồ khối 8051

Phần chính của vi điều khiển 8051 là bộ xử lý trung tâm (CPU: central processing unit) bao gồm :

- Thanh ghi tích lũy A
- Thanh ghi tích lũy phụ B, dùng cho phép nhân và chia
- Đơn vị logic học (ALU : Arithmetic Logical Unit)
- Từ trạng thái chương trình (PSW : Program Status Word)
- Bốn bảng thanh ghi
- Con trỏ ngăn xếp
- Ngoài ra còn có bộ nhớ chương trình , bộ giải mã lệnh, bộ điều khiển thời gian và logic

Đơn vị xử lý trung tâm nhận trực tiếp xung từ bộ giao động ,ngoài ra còn có khả năng đưa một tín hiệu giữ nhịp từ bên ngoài.

Chương trình đang chạy có thể cho dừng lại nhờ một khối điều khiển ngắt ở bên trong .Các nguồn ngắt có thể là : các biến cố ở bên ngoài ,sự tràn bộ đếm định thời hoặc cũng có thể là giao diện nối tiếp.

Hai bộ định thời 16 bit hoạt động như 1 bộ đếm.

Các cổng (port0,port1,port2,port3).Sử dụng vào mục đích điều khiển.Ở cổng 3 có thêm các đường dẫn điều khiển dùng để tra đổi với một bộ nhớ bên ngoài,hoặc để đấu nối giao diện nối tiếp,cũng như các đường ngắt dẫn ở bên ngoài

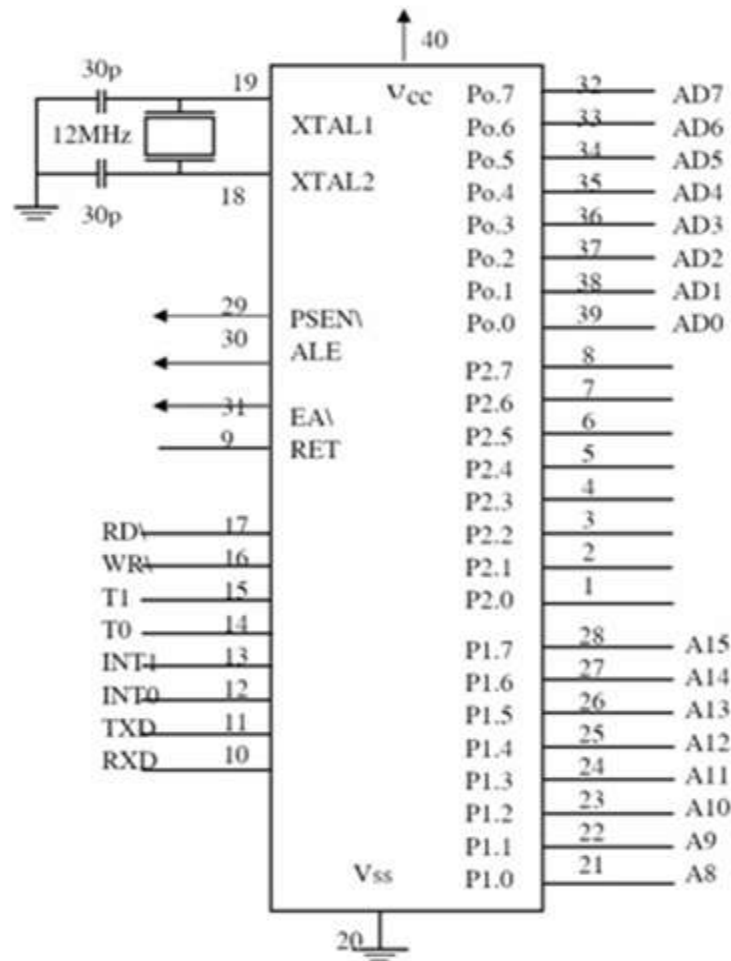
Giao diện nối tiếp có chứa một bộ truyền và một bộ nhận không đồng bộ,làm việc độc lập với nhau.Tốc độ truyền qua cổng nối tiếp có thể đặt trong dải rộng và được ấn định bằng một bộ định thời.

Trong vi điều khiển 8051 có 2 thành phần quan trọng khác đó là bộ nhớ và các thanh ghi:

Bộ nhớ gồm có bộ nhớ ram và bộ nhớ rom(chỉ có ở 8031) dùng để lưu trữ dữ liệu và mã lệnh.

Các thanh ghi sử dụng để lưu trữ thông tin trong quá trình xử lý .Khi CPU làm việc nó làm thay đổi nội dung của các thanh ghi.

2.1.1.2.Chức năng các chân điều khiển



Hình 2.1 sơ đồ chân 8051

a.port0 :là port có chức năng ở trên chân từ 32 đến 39 trong các thiết kế cỡ nhỏ (không dùng bộ nhớ mở rộng) có 2 chức năng như các đường I/O.Đối với các thiết kế cỡ lớn (với bộ nhớ mở rộng) nó được kết hợp kênh giữa các bus.

b.port1 : port1 là một port I/O trên các chân 1 – 8.các chân được kí hiệu p1.0,p1.1,p1.2...có thể dùng cho các thiết bị ngoài nếu cần.Port1 không có chức năng khác, vì vậy chúng ta chỉ được dùng trong giao tiếp với các thiết bị ngoài .

c.port2 : port2 là một port công cụ kép trên các chân 21 – 28 được dùng như các đường xuất nhập hoặc byte cao của bus địa chỉ đối với các thiết kế dùng bộ nhớ mở rộng.

d.port3 : port3 là một port công cụ kép trên các chân 10 -17.các chân của port này có nhiều chức năng , các công cụ chuyển đổi có liên hệ với các tính đặc biệt của 8051 như ở bảng sau:

Bit	Tên	Chức năng chuyển đổi
P3.0	RXD	Dữ liệu nhận cho port nối tiếp
P3.1	TXD	Dữ liệu phát cho port nối tiếp
P3.2	INT0	Ngắt 0 bên ngoài
P3.3	INT1	Ngắt 1 bên ngoài
P3.4	TO	Ngõ vào của timer/counter 0
P3.5	T1	Ngõ vào của timer/counter 1
P3.6	WR	Xung ghi bộ nhớ dữ liệu ngoài
P3.7	RD	Xung đọc bộ nhớ dữ liệu ngoài

e.Psen (program store enable) : 8052 có 4 tín hiệu điều khiển PSEN là tín hiệu ra trên chân 29.Nó là tín hiệu điều khiển để cho phép bộ nhớ chương trình mở rộng và thường được nối đến chân OE (output enable) của 1 EPROM để cho phép đọc các byte mã lệnh.

PSEN sẽ ở mức thấp trong thời gian lấy lệnh.Các mã nhị phân của chương trình được đọc từ EPROM qua bus và được chốt vào thanh ghi lệnh của 8051 để giải mã lệnh.Khi thi hành chương trình trong ROM nội (8051) PSEN sẽ ở mức thụ động (mức cao).

f.Ale (Address latch Enable):

Tín hiệu ra ALE trên chân 30 tương tự với các thiết bị làm việc với các xử lí 8585, 8088 ,8086 , 8051 dùng ALE một cách tương tự cho làm việc giải các kênh bus địa chỉ và dữ liệu khi port0 được dùng trong chế độ chuyển đổi của nó : Vừa là bus dữ liệu vừa là bus thấp của địa chỉ, ALE là tín hiệu để chốt địa chỉ vào một thanh ghi bên ngoài trong nửa đầu của chu kỳ bộ nhớ.Sau đó các đường port 0 dùng để xuất nhập dữ liệu trong nửa sau chu kỳ của bộ nhớ.

Các xung tín hiệu ALE có tốc độ bằng 1/6 lần tần số dao động trên chip và có thể được dùng là nguồn xung nhịp cho các hệ thống. Nếu xung trên 8051 là 12MHz thì ALE có tần số 2MHz. Chỉ ngoại trừ khi thi hành lệnh MOVX, 1 xung ALE sẽ bị mất. Chân này cũng được làm ngõ vào cho xung lập trình cho EPROM trong 8051.

g.Ea (External Access) :

Tín hiệu vào EA trên chân 31 thường được mắc lên mức cao (+5v) hoặc mức thấp (GND) .Nếu ở mức cao ,8051 thi hành chương trình từ ROM nội trong khoảng địa chỉ thấp (4k) .Nếu ở mức thấp, chương trình chỉ được thi hành từ bộ nhớ mở rộng.Nếu EA được nối mức thấp bộ nhớ bên trong, chương trình 8051 sẽ bị cấm và chương trình thi hành từ EPROM mở rộng. Người ta còn dùng chân EA làm chân cấp điện áp 12V khi lập trình cho EPROM trong 8051.

h.Rst (Reset):

Ngõ vào RST trên chân 9 là ngõ reset của 8051. Khi tín hiệu này được đưa lên mức cao (trong ít nhất 2 chu kỳ máy) ,các thanh ghi trong 8051 được tải những giá trị thích hợp để khởi động hệ thống.

Figure 14. Using the On-Chip Oscillator

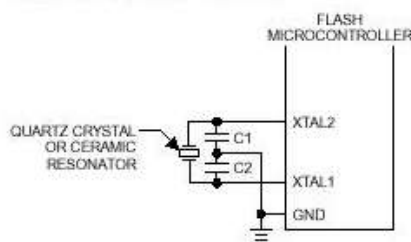
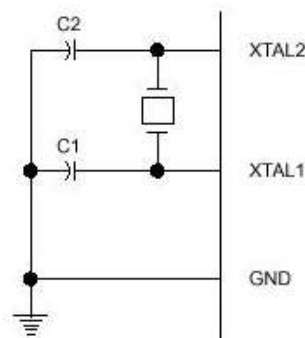


Figure 15. A: Oscillator Connections



i. Các ngõ vào bộ dao động trên chip :

Như đã thấy trong các hình trên , 8051 có 1 bộ dao động trên chip. Nó thường được nối với thạch anh giữa 2 chân 18 và 19. Các tụ giữa cũng cần thiết kể như đã vẽ. Tần số thạch anh thông thường là 12MHz.

j. Các chân nguồn :

Vcc: Cung cấp dương nguồn cho On-chip (+ 5V).

GND: nối mát.

2.1.1.3. Các thanh ghi đặc biệt

SFR đảm nhiệm các chức năng khác nhau trong On-chip. Chúng nằm ở RAM bên trong On-chip, chiếm vùng không gian nhớ 128 Byte được định địa chỉ từ 80h đến FFh. Cấu trúc của SFR bao gồm các chức năng thể hiện ở bảng 2.3 và bảng 2.4.

Thanh ghi	MSB Nội dung LSB							
<i>IE</i>	EA	-	ET2	ES	ET1	EX1	ET0	EX0
<i>IP</i>	-	-	PT2	PS	PT1	PX1	PT0	PX0
<i>PSW</i>	CY	AC	FO	RS1	RS0	OV	-	P
<i>TMOD</i>	GATE	C/(/T)	M1	M0	GATE	C/(/T)	M1	M0
<i>TCON</i>	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
<i>SCON</i>	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
<i>PCON</i>	SMOD	-	-	-	GF1	GF0	PD	IDL
<i>P1</i>	T2	T2EX			/SS	MOSI	MISO	SCK
<i>P3</i>	RXD	TXD	/INT0	/INT1	T0	T1	/WR	/RD

Bảng 2.3. Chức năng riêng của từng thanh ghi trong SFR

Symbol	Name	Address	Reset Values
* ACC	Thanh ghi tích lũy	0E0h	00000000b
* B	Thanh ghi B	0F0h	00000000b
* PSW	Từ trạng thái chương trình	0D0h	00000000b
SP	Con trỏ ngăn xếp	81h	00000111b
DP0L	Byte cao của con trỏ dữ liệu 0	82h	00000000b
DP0H	Byte thấp của con trỏ dữ liệu 0	83h	00000000b
* P0	Cổng 0	80h	11111111b
* P1	Cổng 1	90h	11111111b
* P2	Cổng 2	0A0h	11111111b
* P3	Cổng 3	0B0h	11111111b
* IP	TG điều khiển ngắt ưu tiên	0B8h	xxx00000b
* IE	TG điều khiển cho phép ngắt	0A8h	0xx00000b
TMOD	Điều khiển kiểu Timer/Counter	89h	00000000b
* TCON	TG điều khiển Timer/Counter	88h	00000000b
TH0	Byte cao của Timer/Counter 0	8Ch	00000000b
TL0	Byte thấp của Timer/Counter 0	8Ah	00000000b
TH1	Byte cao của Timer/Counter 1	8Dh	00000000b
TL1	Byte thấp của Timer/Counter 1	8Bh	00000000b
* SCON	Serial Control	98h	00000000b
SBUF	Serial Data Buffer	99h	indeterminate
PCON	Power Control	87h	0xxx0000b

* : có thể định địa chỉ bit, x: không định nghĩa

Bảng 2.4. Địa chỉ, ý nghĩa và giá trị của các SFR sau khi Reset

* **Thanh ghi ACC:** là thanh ghi tích lũy, dùng để lưu trữ các toán hạng và kết quả của phép tính. Thanh ghi ACC dài 8 bits. Trong các tập lệnh của On-chip, nó thường được quy ước đơn giản là A.

* **Thanh ghi B :** Thanh ghi này được dùng khi thực hiện các phép toán nhân và chia. Đối với các lệnh khác, nó có thể xem như là thanh ghi đệm tạm thời. Thanh ghi B dài 8 bits. Nó thường được dùng chung với thanh ghi A trong các phép toán nhân hoặc chia.

* **Thanh ghi SP:** Thanh ghi con trỏ ngăn xếp dài 8 bit. SP chứa địa chỉ của dữ liệu hiện đang ở đỉnh của ngăn xếp. Giá trị của nó được tự động tăng lên khi thực hiện lệnh PUSH trước khi dữ liệu được lưu trữ trong ngăn xếp. SP sẽ tự động giảm xuống khi thực hiện lệnh POP. Ngăn xếp có thể đặt ở bất cứ nơi nào trong RAM on-chip, nhưng sau khi khởi động lại hệ thống thì con trỏ ngăn xếp mặc định sẽ trở tới địa chỉ khởi đầu là 07h, vì vậy ngăn xếp sẽ bắt đầu từ địa chỉ 08h. Ta cũng có thể định con trỏ ngăn xếp tại địa chỉ mong muốn bằng các lệnh di chuyển dữ liệu thông qua định địa chỉ tức thời.

* **Thanh ghi DPTR:** Thanh ghi con trỏ dữ liệu (16 bit) bao gồm 1 thanh ghi byte cao (DPH-8bit) và 1 thanh ghi byte thấp (DPL-8bit). DPTR có thể được dùng như thanh ghi 16 bit hoặc 2 thanh ghi 8 bit độc lập. Thanh ghi này được dùng để truy cập RAM ngoài.

* **Ports 0 to 3:** P0, P1, P2, P3 là các chốt của các cổng 0, 1, 2, 3 tương ứng. Mỗi chốt gồm 8 bit. Khi ghi mức logic 1 vào một bit của chốt, thì chân ra tương ứng của cổng ở mức logic cao. Còn khi ghi mức logic 0 vào mỗi bit của chốt thì chân ra tương ứng của cổng ở mức logic thấp. Khi các cổng đảm nhiệm chức năng như các đầu vào thì trạng thái bên ngoài của các chân cổng sẽ được giữ ở bit chốt tương ứng. Tất cả 4 cổng của on-chip đều là cổng I/O hai chiều, mỗi cổng đều có 8 chân ra, bên trong mỗi chốt bit có bộ “Pullup-tăng cường” do đó nâng cao khả năng nối ghép của cổng với tải (có thể giao tiếp với 4 đến 8 tải loại TTL).

* **Thanh ghi SBUF:** Đệm dữ liệu nối tiếp gồm 2 thanh ghi riêng biệt, một thanh ghi đệm phát và một thanh ghi đệm thu. Khi dữ liệu được chuyển tới SBUF, nó sẽ đi vào bộ đệm phát, và được giữ ở đây để chế biến thành dạng truyền tin nối tiếp. Khi dữ liệu được truyền đi từ SBUF, nó sẽ đi ra từ bộ đệm thu.

* **Các Thanh ghi Timer:** Các đôi thanh ghi (TH0, TL0), (TH1, TL1) là các thanh ghi đếm 16 bit tương ứng với các bộ Timer/Counter 0 và 1.

* **Các thanh ghi điều khiển:** Các thanh ghi chức năng đặc biệt: IP, IE, TMOD, TCON, SCON, và PCON bao gồm các bit trạng thái và điều khiển đối với hệ thống ngắt, các bộ Timer/Counter và cổng nối tiếp. Chúng sẽ được mô tả ở phần sau.

* **Thanh ghi PSW:** Từ trạng thái chương trình dùng để chứa thông tin về trạng thái chương trình. PSW có độ dài 8 bit, mỗi bit đảm nhiệm một chức năng cụ thể. Thanh ghi này cho phép truy cập ở dạng mức bit.

* **CY:** Cờ nhớ. Trong các phép toán số học, nếu có nhớ từ phép cộng bit 7 hoặc có số mượn mang đến bit 7 thì CY được đặt bằng 1.

* **AC:** Cờ nhớ phụ (Đối với mã BCD). Khi cộng các giá trị BCD, nếu có một số nhớ được tạo ra từ bit 3 chuyển sang bit 4 thì AC được đặt bằng 1. Khi giá trị được cộng là BCD, lệnh cộng phải được thực hiện tiếp theo bởi lệnh *DA A* (hiệu chỉnh thập phân thanh chứa A) để đưa các kết quả lớn hơn 9 về giá trị đúng.

* **F0:** Cờ 0 (Có hiệu lực với các mục đích chung của người sử dụng)

* **RS1:** Bit 1 điều khiển chọn băng thanh ghi.

* **RS0:** Bit 0 điều khiển chọn băng thanh ghi.

Lưu ý: RS0, RS1 được đặt/xoá bằng phần mềm để xác định băng thanh ghi đang hoạt động (Chọn băng thanh ghi bằng cách đặt trạng thái cho 2 bit này)

	RS1	RS0
Bank 0	0	0
Bank 1	0	1
Bank 2	1	0
Bank 3	1	1

Bảng 2.5. Chọn băng thanh ghi

* OV: Cờ tràn. Khi thực hiện các phép toán cộng hoặc trừ mà xuất hiện một tràn số học, thì OV được đặt bằng 1. Khi các số có dấu được cộng hoặc được trừ, phần mềm có thể kiểm tra OV để xác định xem kết quả có nằm trong tầm hay không. Với phép cộng các số không dấu, OV được bỏ qua. Kết quả lớn hơn +128 hoặc nhỏ hơn -127 sẽ đặt OV=1.

* -: Bit dành cho người sử dụng tự định nghĩa(Nếu cần).

* P: Cờ chẵn lẻ. Được tự động đặt/ xoá bằng phần cứng trong mỗi chu trình lệnh để chỉ thị số chẵn hay lẻ của bit 1 trong thanh ghi tích lũy. Số các bit 1 trong A cộng với bit P luôn luôn là số chẵn.

* **Thanh ghi PCON:** Thanh ghi điều khiển nguồn.

* SMOD: Bit tạo tốc độ Baud gấp đôi. Nếu Timer 1 được sử dụng để tạo tốc độ baud và SMOD=1, thì tốc độ Baud được tăng lên gấp đôi khi cổng truyền tin nối tiếp được dùng bởi các kiểu 1, 2 hoặc 3.

* -: Không sử dụng, các bit này có thể được dùng ở các bộ VXL trong tương lai. Người sử dụng không được phép tự định nghĩa cho các bit này.

* GF0, GF1: Cờ dùng cho các mục đích chung (đa mục đích).

* PD: bit nguồn giảm. Đặt bit này ở mức tích cực để vận hành chế độ nguồn giảm trong AT89C51. Chỉ có thể ra khỏi chế độ bằng Reset.

* IDL: bit chọn chế độ nghỉ. Đặt bit này ở mức tích cực để vận hành kiểu Idle (Chế độ không làm việc) trong AT89C51.

Lưu ý: Nếu PD và IDL cùng được kích hoạt cùng 1 lúc ở mức tích cực, thì PD được ưu tiên thực hiện trước. Chỉ ra khỏi chế độ bằng 1 ngắt hoặc Reset lại hệ thống.

* **Thanh ghi IE**: Thanh ghi cho phép ngắt

* EA: Nếu EA=0, không cho phép bất cứ ngắt nào hoạt động. Nếu EA=1, mỗi nguồn ngắt riêng biệt được phép hoặc không được phép hoạt động bằng cách đặt hoặc xoá bit Enable của nó.

* -: Không dùng, người sử dụng không nên định nghĩa cho Bit này, bởi vì nó có thể được dùng ở các bộ AT89 trong tương lai.

* ET2: Bit cho phép hoặc không cho phép ngắt bộ Timer 2.

* ES: Bit cho phép hoặc không cho phép ngắt cổng nối tiếp (SPI và UART).

* ET1: Bit cho phép hoặc không cho phép ngắt tràn bộ Timer 1

* EX1: Bit cho phép hoặc không cho phép ngắt ngoài 1.

* ET0: Bit cho phép hoặc không cho phép ngắt tràn bộ Timer 0

* EX0: Bit cho phép hoặc không cho phép ngắt ngoài 0.

* **Thanh ghi IP**: Thanh ghi ưu tiên ngắt.

- : Không dùng, người sử dụng không nên ghi “1” vào các Bit này.

* PT2: Xác định mức ưu tiên của ngắt Timer 2.

* PS: Định nghĩa mức ưu tiên của ngắt cổng nối tiếp.

* PT1: Định nghĩa mức ưu tiên của ngắt Timer 1.

* PX1: Định nghĩa mức ưu tiên của ngắt ngoài 1.

* PT0: Định nghĩa mức ưu tiên của ngắt Timer 0.

* PX0: Định nghĩa mức ưu tiên của ngắt ngoài 0.

* **Thanh ghi TCON** : Thanh ghi điều khiển bộ Timer/Counter

* TF1: Cờ tràn Timer 1. Được đặt bởi phần cứng khi bộ Timer 1 tràn. Được xoá bởi phần cứng khi bộ vi xử lý hướng tới chương trình con phục vụ ngắt.

* TR1: Bit điều khiển bộ Timer 1 hoạt động. Được đặt/xoá bởi phần mềm để điều khiển bộ Timer 1 ON/OFF

* TF0: Cờ tràn Timer 0. Được đặt bởi phần cứng khi bộ Timer 0 tràn. Được xoá bởi phần cứng khi bộ vi xử lý hướng tới chương trình con phục vụ ngắt.

* TR0: Bit điều khiển bộ Timer 0 hoạt động. Được đặt/xoá bởi phần mềm để điều khiển bộ Timer 0 ON/OFF.

* IE1: Cờ ngắt ngoài 1. Được đặt bởi phần cứng khi sườn xung của ngắt ngoài 1 được phát hiện. Được xoá bởi phần cứng khi ngắt được xử lý.

* IT1: Bit điều khiển ngắt 1 để tạo ra ngắt ngoài. Được đặt/xoá bởi phần mềm.

* IE0: Cờ ngắt ngoài 0. Được đặt bởi phần cứng khi sườn xung của ngắt ngoài 0 được phát hiện. Được xoá bởi phần cứng khi ngắt được xử lý.

* IT0: Bit điều khiển ngắt 0 để tạo ra ngắt ngoài. Được đặt/xoá bởi phần mềm.

* **Thanh ghi TMOD**: Thanh ghi điều khiển kiểu Timer/Counter

* GATE: Khi TRx được thiết lập và GATE=1, bộ TIMER/COUNTERx hoạt động chỉ khi chân INTx ở mức cao. Khi GATE=0, TIMER/COUNTERx sẽ hoạt động chỉ khi TRx=1.

* C/(/T): Bit này cho phép chọn chức năng là Timer hay Counter.

- Bit này được xoá để thực hiện chức năng Timer

- Bit này được đặt để thực hiện chức năng Counter

* M0, M1: Bit chọn Mode, để xác định trạng thái và kiểu Timer/Counter:

- M1=0, M0=0: Chọn kiểu bộ Timer 13 bit. Trong đó THx dài 8 bit, còn TLx dài 5 bit.

- M1=0, M0=1: Chọn kiểu bộ Timer 16 bit. THx và TLx dài 16 bit được ghép tầng.

- M1=1, M0=0: 8 bit Auto reload. Các thanh ghi tự động nạp lại mỗi khi bị tràn. Khi bộ Timer bị tràn, THx dài 8 bit được giữ nguyên giá trị, còn giá trị nạp lại được đưa vào TLx.

- M1=1, M0=1: Kiểu phân chia bộ Timer. TL0 là 1 bộ Timer/Counter 8 bit, được điều khiển bằng các bit điều khiển bộ Timer 0, Còn TH0 chỉ là bộ Timer 8 bit, được điều khiển bằng các bit điều khiển Timer 1.

- M1=1, M0=1: Timer/Counter 1 Stopped

*** Thanh ghi SCON:**

SCON là thanh ghi trạng thái và điều khiển cổng nối tiếp. Nó không những chứa các bit chọn chế độ, mà còn chứa bit dữ liệu thứ 9 dành cho việc truyền và nhận tin (TB8 và RB8) và chứa các bit ngắt cổng nối tiếp.

* SM0, SM1: Là các bit cho phép chọn chế độ cho cổng truyền nối tiếp.

SM0	SM1	Mode	Đặc điểm	Tốc độ Baud
0	0	0	Thanh ghi dịch	$F_{osc}/12$
0	1	1	8 bit UART	Có thể thay đổi (được đặt bởi bộ Timer)
1	0	2	9 bit UART	$F_{osc}/64$ hoặc $F_{osc}/32$
1	1	3	9 bit UART	Có thể thay đổi (được đặt bởi bộ Timer)

Bảng 2.6. Chọn Mode trong SCON

* SM2: Cho phép truyền tin đa xử lý, thể hiện ở Mode 2 và 3. ở chế độ 2 hoặc 3, nếu đặt SM2 = 1 thì RI sẽ không được kích hoạt nếu bit dữ liệu thứ 9 (RB8) nhận được giá trị bằng 0. ở Mode 1, nếu SM2=1 thì RI sẽ không được kích hoạt nếu bit dừng có hiệu lực đã không được nhận. ở chế độ 0, SM2 nên bằng 0

* REN: Cho phép nhận nối tiếp. Được đặt hoặc xoá bởi phần mềm để cho phép hoặc không cho phép nhận.

* TB8: Là bit dữ liệu thứ 9 mà sẽ được truyền ở Mode 2 và 3. Được đặt hoặc xoá bởi phần mềm.

* RB8: Là bit dữ liệu thứ 9 đã được nhận ở Mode 2 và 3. Ở Mode 1, nếu SM2=0 thì RB8 là bit dừng đã được nhận. Ở Mode 0, RB8 không được sử dụng.

* TI: Cờ ngắt truyền. Được đặt bởi phần cứng tại cuối thời điểm của bit thứ 8 trong Mode 0, hoặc đầu thời điểm của bit dừng trong các Mode khác. Ở bất kỳ quá trình truyền nối tiếp nào, nó cũng phải được xoá bằng phần mềm.

* RI: Cờ ngắt nhận. Được đặt bởi phần cứng tại cuối thời điểm của bit thứ 8 trong Mode 0, hoặc ở giữa thời điểm của bit dừng trong các Mode khác. Ở bất kỳ quá trình nhận nối tiếp nào (trừ trường hợp ngoại lệ, xem SM2), nó cũng phải được xoá bằng phần mềm.

2.1.1.4. Khởi tạo thời gian và bộ đếm (Timer/Counter).

On-chip AT89C51 có 2 thanh ghi Timer/Counter dài 16 bit, đó là: Timer 0 và Timer 1. Trong On-chip AT89C52, ngoài Timer 0 và Timer 1 nó còn có thêm bộ Timer 2. Cả 3 bộ Timer này đều có thể được điều khiển để thực hiện chức năng thời gian hay bộ đếm, thông qua thanh ghi TMOD.

Khi thanh ghi Timer/Counter làm việc ở kiểu Timer, thì sau mỗi chu kỳ máy nội dung trong thanh ghi được gia tăng thêm 1 đơn vị. Vì vậy thanh ghi này đếm số chu kỳ máy. Một chu kỳ máy có 12 chu kỳ dao động, do đó tốc độ đếm của thanh ghi là 1/12 tần số dao động.

Khi thanh ghi Timer/Counter làm việc ở kiểu Counter, xung nhịp bên ngoài được đưa vào để đếm ở T0 hoặc T1. Nội dung thanh ghi được tăng lên khi có sự chuyển trạng thái từ 1 về 0 tại chân đầu vào ngoài T0 hoặc T1. Xung nhịp ở các đầu vào ngoài được lấy mẫu tại thời điểm S5P2 của mỗi chu kỳ máy. Khi quá trình lấy mẫu phát hiện ra mức cao ở 1 chu kỳ và mức thấp ở chu kỳ tiếp theo, thì bộ đếm được tăng lên. Giá trị mới của bộ đếm xuất hiện trong thanh ghi tại thời điểm S3P1 của chu kỳ máy sau khi sự chuyển trạng thái đã được phát hiện. Vì vậy để nội dung của thanh ghi tăng lên 1 đơn vị phải mất 2 chu kỳ máy, nên tốc độ đếm tối đa là 1/24 tần số bộ dao động. Không có sự giới hạn số vòng thực hiện của tín hiệu ở đầu vào ngoài, nhưng nó sẽ giữ ít nhất 1 chu kỳ máy đầy đủ để đảm bảo chắc chắn rằng một mức đã cho được lấy mẫu ít nhất 1 lần nữa trước khi nó thay đổi.

Do xung nhịp bên ngoài có tần số bất kỳ nên các bộ Timer (0 và 1) có 4 chế độ làm việc khác nhau để lựa chọn: (13 bit Timer, 16 bit Timer, 8 bit auto-reload, split Timer).

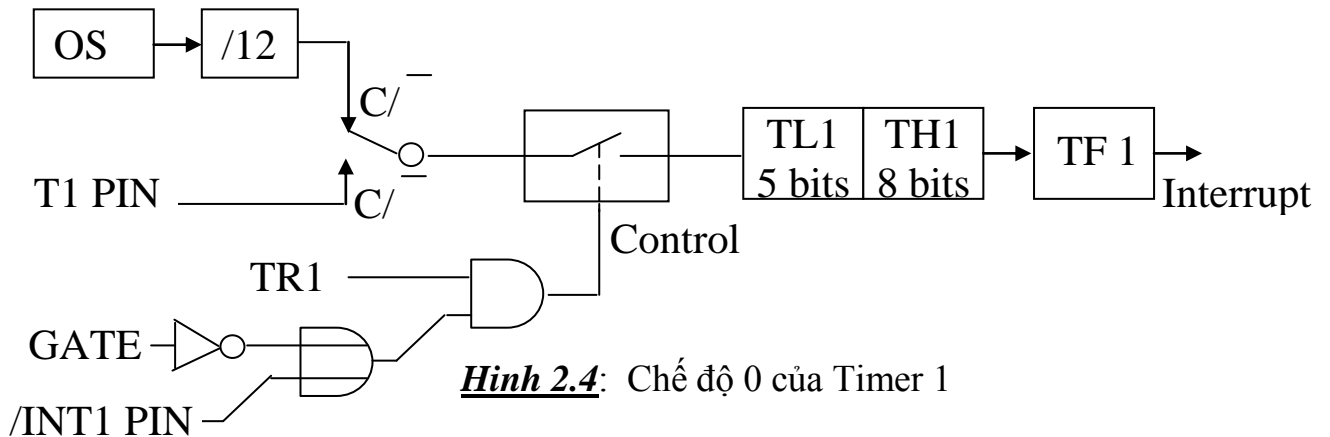
Timer 0 và Timer 1:

Trong AT89C51 và AT89C52 đều có các bộ Timer 0 và 1. Chức năng Timer hay Counter được chọn lựa bởi các bit điều khiển C/(/T) trong thanh ghi TMOD. Hai bộ Timer/Counter này có 4 chế độ hoạt động, được lựa chọn bởi cặp bit (M0, M1) trong TMOD. Chế độ 0, 1 và 2 giống nhau cho các chức năng Timer/Counter, nhưng chế độ 3 thì khác. Bốn chế độ hoạt động được mô tả như sau:

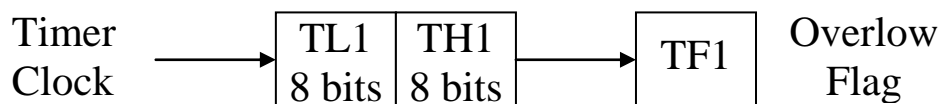
+ **Chế độ 0:** Cả 2 bộ Timer 0 và 1 ở chế độ 0 có cấu hình như một thanh ghi 13 bit, bao gồm 8 bit của thanh ghi THx và 5 bit thấp của TLx. 3 bit cao của TLx không xác định chắc chắn, nên được làm ngơ. Khi thanh ghi được xoá về 0, thì cờ ngắt thời gian TFX được thiết lập. Bộ Timer/Counter hoạt động khi bit điều khiển TRx được thiết lập (TRx=1) và, hoặc Gate trong TMOD bằng 0, hoặc /INTx=1. Nếu đặt GATE=1 thì cho phép điều khiển

Timer/ Counter bằng đường vào ngoài /INTx, để dễ dàng xác định độ rộng xung.

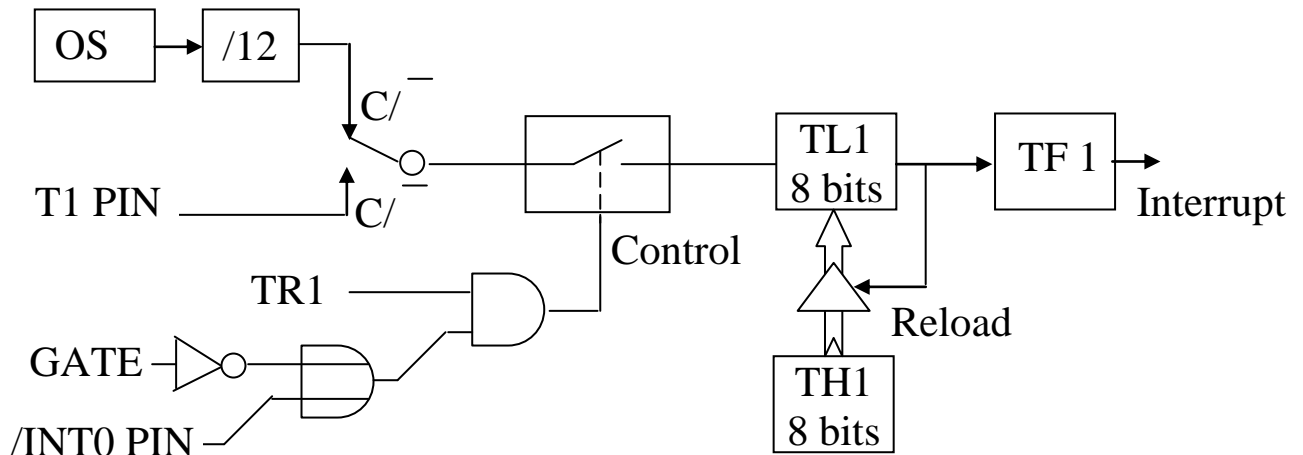
Khi hoạt động ở chức năng thời gian thì bit C/(/T)=0, do vậy xung nhịp từ bộ dao động nội, qua bộ chia tần cho ra tần số $f=f_{osc}/12$ được đưa vào để đếm trong thanh ghi Timer/Counter. Khi hoạt động ở chức năng bộ đếm thì bit C/(/T)=1, lúc đó xung nhịp ngoài đưa vào sẽ được đếm.



+ **Chế độ 1:** hoạt động tương tự như chế độ 0, chỉ khác là thanh ghi Timer/Counter được sử dụng cả 16 bit. Xung nhịp được dùng kết hợp với các thanh ghi thời gian byte thấp và byte cao (TH1 và TL1). Khi xung Clock được nhận, bộ Timer sẽ đếm tăng lên: 0000h, 0001h, 0002,... Khi hiện tượng tràn xảy ra, cờ tràn sẽ chuyển FFFFh về 0000h, và bộ Timer tiếp tục đếm. Cờ tràn của Timer 1 là bit TF1 ở trong TCON, nó được đọc hoặc ghi bởi phần mềm, xem hình 2.5 (Timer/Counter 1 Mode 1: 16 bit Counter).



Hình 2.5: Chế độ 1 của Timer 1



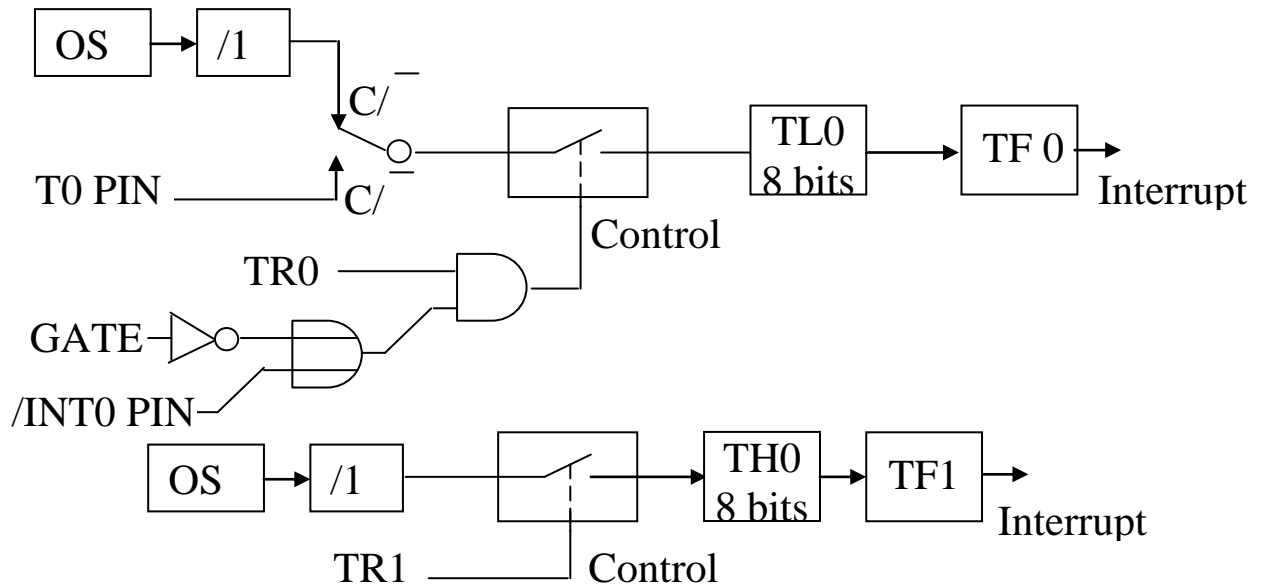
Hình 2.6: Chế độ 2 của Timer 1

+ **Chế độ 2:** Chế độ này của thanh ghi Timer cũng hoạt động tương tự như 2 chế độ trên, nhưng nó được tổ chức như bộ đếm 8 bit (TL1) với chế độ tự động nạp lại, như hình 2.6. Khi xảy ra hiện tượng tràn ở TL1, không chỉ thiết lập bit TF1 mà còn tự động nạp lại cho TL1 bằng nội dung của TH1, đã được thiết lập bởi phần mềm. Quá trình nạp lại cho phép nội dung của TH1 không bị thay đổi. Chế độ 2 của Timer/Counter 0 cũng tương tự như Timer/Counter 1.

+ **Chế độ 3:** Ở chế độ này, chức năng Timer/Counter 0 và chức năng Timer/Counter 1 khác nhau. Bộ Timer 1 ở chế độ 3 chỉ chứa chức năng đếm của nó, kết quả giống khi đặt TR1=0. Bộ Timer 0 ở chế độ 3 thiết lập TH0, TL0 như là 2 bộ đếm riêng biệt. Mạch Logic đối với chế độ 3 của Timer 0 thể hiện ở hình 2.7. Bộ đếm TL0 được điều khiển bởi các bit: C/(/T), GATE, TR0, /INT0 và khi đếm tràn nó thiết lập cờ ngắt TF0. Bộ đếm TH0 chỉ được điều khiển bởi bit TR1, và khi đếm tràn nó thiết lập cờ ngắt TF1. Vậy, TH0 điều khiển ngắt Timer/Counter 1.

Chế độ 3 thường được dùng khi yêu cầu cần có bộ thời gian hoặc bộ đếm ngoài 8 bit. Đối với Timer 0 ở chế độ 3, AT89C51 có thể có 3 bộ Timer/Counter, còn AT89C52 có thể có 4 bộ. Khi Timer 0 hoạt động ở chế độ 3, thì Timer 1 có thể được bật hoặc tắt bằng chuyên mạch ngoài. Ở chế độ

này, Timer 1 có thể được sử dụng bởi công nối tiếp như một bộ tạo tốc độ Baud, hoặc trong bất kỳ ứng dụng nào mà không yêu cầu một ngắt.



Hình 2.7: Chế độ 3 của Timer 0

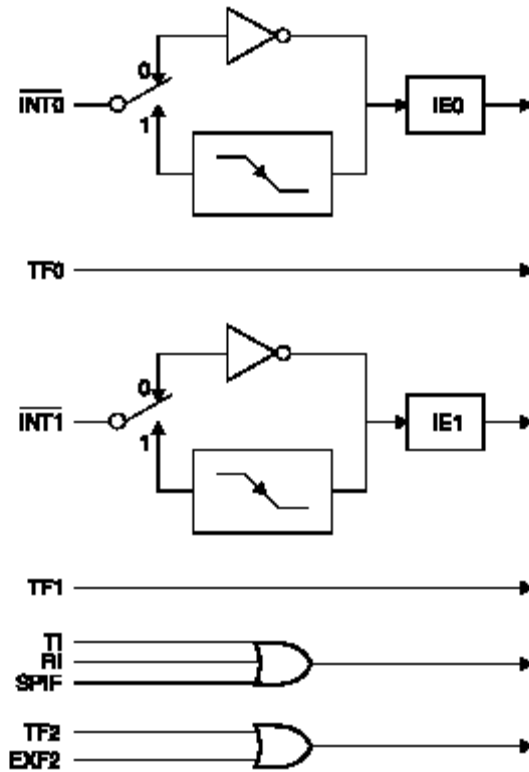
2.1.1.5. Cơ chế ngắt trong On-chip AT89C51:

2.1.1.5.1. Phân loại ngắt trong On-chip:

Bộ AT89C51 có tất cả 5 Vectors ngắt bao gồm: 2 ngắt ngoài ($\overline{\text{INT0}}$ và $\overline{\text{INT1}}$), 2 ngắt của khối thời gian (Timer 0, 1), và ngắt công truyền tin nối tiếp.

Mỗi nguồn ngắt có thể được kích hoạt hoặc không kích hoạt bằng cách đặt hoặc xoá Bit ở trong IE. IE cũng chứa bit có thể không cho tất cả các ngắt hoạt động EA (Nếu EA=0). Các ngắt ngoài có thể được kích hoạt theo mức hoặc theo sườn xung, tùy thuộc vào giá trị của các bit IT0, IT1 trong TCON. Ngắt ngoài có 2 cờ ngắt tương ứng là IE0, IE1 cũng nằm trong TCON. Khi một ngắt được thực hiện thì cờ ngắt tương ứng của nó bị xoá bằng phần cứng. Chương trình con phục vụ ngắt hoạt động chỉ khi ngắt được kích hoạt theo sườn xung. Nếu ngắt được kích hoạt theo mức thì nguồn yêu cầu ngắt từ bên ngoài điều khiển cờ ngắt.

Figure 10. Interrupt Sources



Hình 2.17. Các nguồn ngắt của AT89C51

Các ngắt trong, với ngắt Timer/Counter 0, 1 được phát sinh bởi cờ ngắt TF0, TF1. Hai cờ ngắt này được thiết lập khi thanh ghi Timer/Counter thực hiện quay vòng, tại thời điểm S5P2 của chu trình máy. Khi một ngắt được thực hiện thì cờ ngắt tương ứng phát sinh ra ngắt sẽ bị xoá bằng phần cứng trong On-chip.

Ngắt cổng nối tiếp được phát sinh bởi các ngắt RI, TI, SPIF thông qua phần tử Logic OR, khi chương trình con phục vụ ngắt được kích hoạt thì các cờ ngắt phát sinh tương ứng được xoá bằng phần mềm. Các ngắt trong có thể được phép hoặc không được phép kích hoạt bằng cách đặt hoặc xoá một bit trong IE.

2.1.1.5.2. Các bước thực hiện ngắt.

Theo đúng trình tự, để sử dụng các ngắt trong Flash Microcontroller, cần thực hiện các bước như sau:

- Đặt bit EA ở trong IE mức logic 1.
- Đặt bit cho phép ngắt tương ứng ở trong IE mức logic 1.

- Bắt đầu chương trình con phục vụ ngắt tại địa chỉ của ngắt tương ứng đó.

(Xem bảng địa chỉ Vector của các nguồn ngắt)

Ngoài ra, đối với các ngắt ngoài, các chân /INT0, /INT1 phải được đặt mức 1. Và tùy thuộc vào ngắt được kích hoạt bằng mức hay sườn xung, mà các bit IT0, IT1 ở trong TCON có thể cần phải đặt mức 1.

ITx=0: Kích hoạt bằng mức

ITx=1: Kích hoạt bằng sườn xung.

2.1.1.5.3. *Mức ngắt ưu tiên trong on-chip:*

Mỗi nguồn ngắt có thể được lập trình riêng cho 1 hoặc 2 mức ưu tiên bằng cách đặt hoặc xoá 1 bit trong IP của SFR. Mỗi ngắt ưu tiên ở mức thấp có thể được ngắt bằng ngắt ưu tiên ở mức cao hơn nhưng không thể ngắt bằng ngắt có mức ưu tiên ở mức thấp hơn được. Một ngắt ưu tiên ở mức cao có thể được ngắt bởi bất kỳ nguồn ngắt nào khác.

Nếu có yêu cầu ngắt của 2 mức ưu tiên cùng nhau (cùng 1 lúc), yêu cầu của mức ưu tiên cao hơn sẽ được phục vụ (Ngắt nào có mức ưu tiên cao hơn sẽ được phục vụ). Nếu các yêu cầu ngắt có cùng mức ưu tiên, thì thứ tự quay vòng bên trong sẽ quyết định ngắt nào được phục vụ.

Thứ tự ưu tiên ngắt từ cao xuống thấp của AT89C51 như sau:

IE0, TF0, IE1, TF1, RI hoặc TI.

2.1.1.5.4. *Nguyên lý điều khiển ngắt của AT89:*

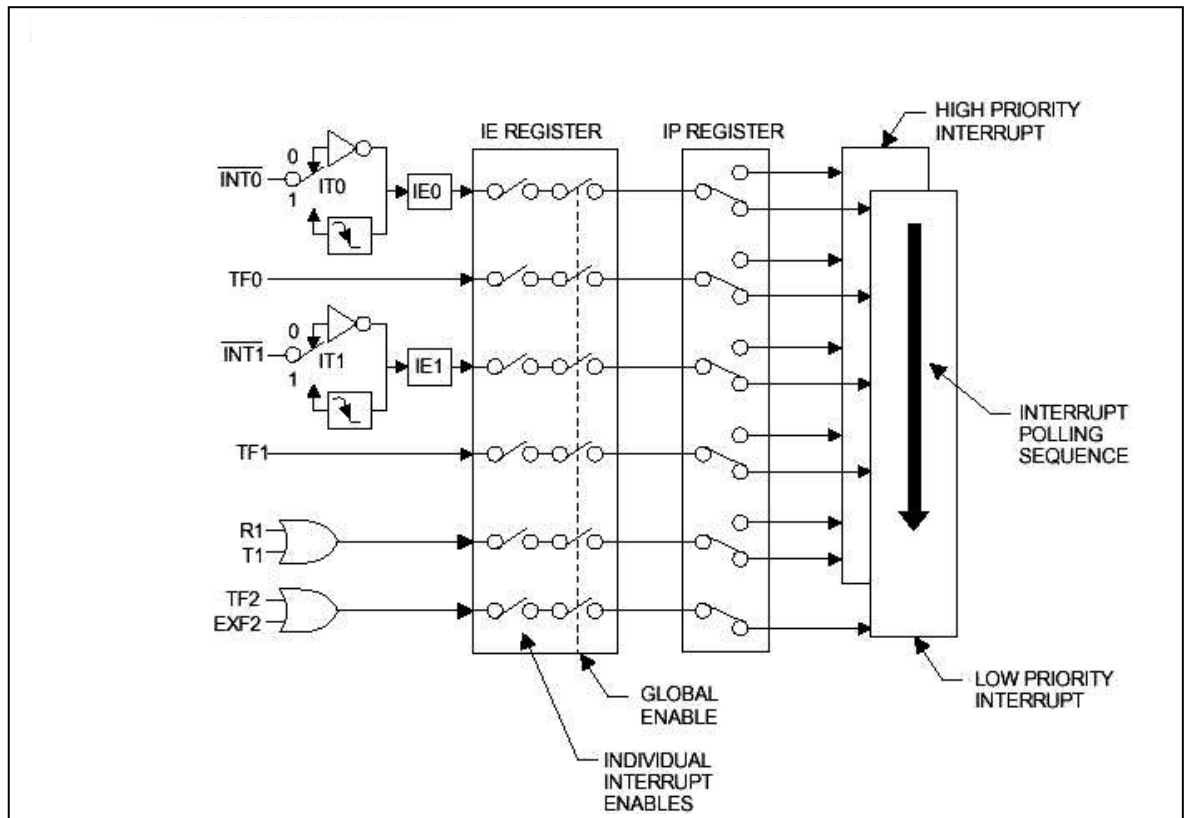
Các cờ ngắt được thiết lập tại thời điểm S5P2 của mỗi chu kỳ máy. Chu kỳ máy tiếp theo sau chu kỳ máy có cờ ngắt được thiết lập, thì chương trình con được thiết lập khi có lệnh gọi LCALL. Lệnh LCALL phát sinh nhưng lại bị cấm hoạt động khi gặp các tình huống sau:

a- Đồng thời có ngắt với mức ưu tiên cao hơn hoặc bằng ngắt đang phục vụ.

(Một ngắt có mức ưu tiên bằng hoặc cao hơn đang sẵn sàng để được phục vụ)

b- Chu kỳ máy hiện hành không phải là chu kỳ máy cuối cùng của lệnh đang thực hiện.

c- Lệnh đang thực hiện là RETI hoặc bất kỳ lệnh nào ghi vào thanh ghi IE hoặc IP.



Hình 2.18. Hệ thống ngắt của AT89

Bất kỳ một trong 3 điều kiện này xuất hiện sẽ cản trở việc tạo ra LCALL đối với chương trình phục vụ ngắt. Điều kiện 2 đảm bảo rằng, lệnh đang thực hiện sẽ được hoàn thành trước khi trở tới bất kỳ chương trình phục vụ nào. Điều kiện 3 đảm bảo rằng, nếu lệnh đang thực hiện là RETI hoặc bất kỳ sự truy cập nào vào IE hoặc IP, thì ít nhất một lệnh nữa sẽ được thực hiện trước khi bất kỳ ngắt nào được trở tới. Chu trình kiểm tra vòng được lặp lại với mỗi chu trình máy, và các giá trị được kiểm tra là các giá trị mà đã xuất hiện ở thời điểm S5P2 của chu trình máy trước đó. Nếu một chỉ thị ngắt có hiệu lực nhưng không được đáp ứng vì các điều kiện trên và nếu chỉ thị này

vẫn chưa có hiệu lực khi điều kiện cản trở được loại bỏ, thì ngắt bị từ chối này sẽ không được phục vụ nữa.

LCALL do phần cứng tạo ra sẽ chuyển nội dung của bộ đếm chương trình vào ngăn xếp (Nhưng không ghi vào PSW) và nạp lại cho PC một địa chỉ phụ thuộc vào nguồn gây ngắt đang được phục vụ, như bảng dưới đây:

Ngắt	Nguồn ngắt	Địa chỉ Véc tơ
External 0	IE0	0003h
Timer 0	TF0	000Bh
External 1	IE1	0013h
Timer 1	TF1	001Bh
Serial Port	RI hoặc TI	0023h
Timer 2 (AT89C52)	TF2 hoặc EXF2	002Bh
System Reset	RST	0000h

Bảng 2.7. Địa chỉ véc tơ ngắt

Lệnh RETI thông báo cho bộ VXL rằng thủ tục ngắt này đã kết thúc, sau đó lấy ra 2 Byte từ ngăn xếp và nạp lại cho PC để trả lại quyền điều khiển cho chương trình chính.

* .*Các ngắt ngoài:*

Vì các chốt ngắt ngoài được tạo mẫu mỗi lần trong mỗi chu trình máy, nên một giá trị cao hoặc thấp của đầu vào sẽ duy trì trong ít nhất là 12 chu kỳ xung nhịp của bộ dao động để đảm bảo tạo mẫu. Nếu ngắt ngoài được kích hoạt bằng sườn xung , thì nguồn ngắt ngoài phải duy trì ở chốt yêu cầu giá trị cao ít nhất 1 chu kỳ máy và sau đó duy trì giá trị thấp ít nhất 1 chu kỳ máy

nữa. Việc này được thực hiện để đảm bảo rằng quá trình chuyển tiếp cho thấy chỉ thị yêu cầu ngắt IEx sẽ được xác lập. IEx sẽ tự động được xoá bởi CPU khi thủ tục ngắt đáp ứng được gọi.

Nếu ngắt ngoài được kích hoạt theo mức, thì nguồn ngắt bên ngoài phải duy trì cho yêu cầu này có hiệu lực cho đến khi ngắt đã được yêu cầu thực sự được tạo ra. Sau đó nguồn ngắt ngoài phải huỷ yêu cầu đó trước khi thủ tục phục vụ ngắt hoàn thành, nếu không ngắt khác sẽ được tạo ra.

**. Vận hành Single-Step:*

Cấu trúc ngắt AT89C51 cho phép thực hiện các bước đơn với sự tham gia của rất ít phần mềm. Như đã lưu ý trước đây, một yêu cầu ngắt sẽ không được đáp ứng khi một ngắt khác có cùng mức ưu tiên vẫn đang hoạt động, nó cũng không được đáp ứng sau khi có lệnh RETI cho đến khi có ít nhất một lệnh khác đã được thực hiện. Do đó mỗi khi một thủ tục ngắt được đưa vào, thì nó không thể được đưa vào lần nữa cho đến khi ít nhất một lệnh của chương trình ngắt được thực hiện. Một cách để sử dụng đặc điểm này đối với hoạt động theo bước đơn lẻ là lập trình cho 1 trong những ngắt ngoài(chẳng hạn /INT0) được kích hoạt theo mức.

Nếu chân /INT0 được duy trì ở mức thấp, thì CPU sẽ chuyển ngay đến thủ tục ngắt ngoài 0 và dừng ở đó cho tới khi INT0 được nhận xung từ thấp lên cao rồi xuống thấp. Sau đó nó sẽ thực hiện lệnh RETI, trở lại nhiệm vụ chương trình, thực hiện một lệnh, và ngay sau đó nhập lại thủ tục ngắt ngoài 0 để đợi xung nhịp tiếp theo của P3.2. Mỗi bước của nhiệm vụ chương trình được thực hiện vào mỗi thời điểm chân P3.2 được nhận xung.

2.1.1.6. Bảo vệ chương trình.

Họ VĐK	Các bit khoá
AT89C51	LB1, LB2, LB3
AT89C52	LB1, LB2, LB3
AT89C2051	LB1, LB2
AT89C1051	LB1, LB2

Khoá bộ nhớ chương trình cho họ VĐK AT89C51:

<i>Chế độ</i>	<i>LB1</i>	<i>LB2</i>	<i>LB3</i>	<i>Loại bảo vệ</i>
1	U	U	U	Không có đặc trưng khoá chương trình.
2	P	U	U	Các lệnh MOVC được thực thi từ bộ nhớ chương trình ngoài, không được phép tìm nạp lệnh từ bộ nhớ nội. EA được lấy mẫu và chốt khi reset. Việc lập trình trên Flash bị cấm.
3	P	P	U	Như chế độ 2, ngoài ra còn cấm việc kiểm tra chương trình.
4	P	P	P	Như chế độ 3, ngoài ra còn cấm việc thực thi chương trình ngoài.

Lưu ý: P=Programmed, U = Unprogrammed

2.1.1.7 Tra cứu nhanh tập lệnh của 8051

Bảng 1 : Các lệnh toán học của bộ VĐK họ ATMEN:

Câu lệnh	Chức năng	Các kiểu định đ/chỉ				Thời gian thực hiện (us)
		Trực tiếp	Gián tiếp	Thanh ghi	Tức thời	
ADD A,<byte>	$A = A + \langle \text{byte} \rangle$	x	x	x	x	1
ADDC A,<byte>	$A = A + \langle \text{byte} \rangle + C$	x	x	x	x	1
SUBB A,<byte>	$A = A - \langle \text{byte} \rangle$	x	x	x	x	1
INC A	$A = A + 1$	Chỉ với A				1
INC <byte>	$\langle \text{byte} \rangle = \langle \text{byte} \rangle + 1$	x	x	x		1
INC DPTR	$DPTR = DPTR + 1$	Chỉ với DPTR				2
DEC A	$A = A - 1$	Chỉ với A				1
DEC <byte>	$\langle \text{byte} \rangle = \langle \text{byte} \rangle - 1$	x	x	x		1
MUL AB	$BA = A * B$	Chỉ với A&B				4
DIV AB	$A = \text{Int}(A/B);$ $B = \text{Mode}(A/B)$	Chỉ với A&B				4
DA A	Hiệu chỉnh số thập phân	Chỉ với A				1

Bảng 2. Các lệnh chuyển đổi dữ liệu để truy cập vùng nhớ dữ liệu

trong:

Câu lệnh	Chức năng	Các kiểu định địa chỉ				Thời gian Thực hiện(us)
		Trực tiếp	Gián tiếp	Thanh ghi	Tức thời	
MOV A,<scr>	A = <scr>	x	x	x	x	1
MOV <dest>,A	<dest> = A	x	x	x		1
MOV <dest>,<scr>	<dest> = <scr>	x	x	x	x	2
MOV <DPTR>,#data16	DPTR = h/số tức thời 16 bit				x	2
PUSH <scr>	INC SP; Mov “@SP“, <scr>	x				2
POP <dest>	Mov <dest>,”@SP“ ;DEC SP	x				2
XCH a,<byte>	Đổi dữ liệu giữa A&byte	x	x	x		1
XCHD A,@Ri	Đổi nửa bit thấp giữa A&@Ri		x			1

Bảng 3. Các lệnh số học:

Câu lệnh	Chức năng	Các kiểu định địa chỉ				Thời gian Thực hiện(us)
		Trực tiếp	Gián tiếp	Thanh ghi	Tức thời	
ANL A, <byte>	A=A AND <byte>	x	x	x	x	1
ANL <byte>,A	<byte>= <byte> AND A	x				1
ANL <byte>,#data	<byte>= <byte> AND #data	x				2
ORL A, <byte>	A=A OR <byte>	x	x	x	x	1
ORL <byte>,A	<byte>= <byte> OR A	x				1
ORL <byte>,#data	<byte>= <byte> OR #data	x				2
XRL A, <byte>	A=A XOR <byte>	x	x	x	x	1
XRL <byte>,A	<byte>= <byte> XOR A	x				1
XRL <byte>,#data	<byte>= <byte> XOR #data	x				2
CLR A	A = 00h	Chỉ với A				1
CPL A	A = NOT A	Chỉ với A				1
RL A	Dịch A sang trái 1 bit	Chỉ với A				1
RLC A	Dịch A sang trái thông	Chỉ với A				1

		qua C		
RR	A	Dịch A sang phải 1 bit	Chỉ với A	1
RRC	A	Dịch A sang phải thông qua C	Chỉ với A	1
SWAP	A	Đổi nửa bit trong A	Chỉ với A	1

Bảng 4. Các lệnh đại số:

Câu lệnh	Chức năng	Thời gian Thực hiện(us)
ANL C,bit	$C = C \text{ AND bit}$	2
ANL C,/bit	$C = C \text{ AND NOT bit}$	2
ORL C,bit	$C = C \text{ ORL bit}$	2
ORL C,/bit	$C = C \text{ ORL NOT bit}$	2
MOV C,bit	$C = \text{bit}$	1
MOV bit,C	$\text{Bit} = C$	2
CLR C	$C = 0$	1
CLR bit	$\text{Bit} = 0$	1
SETB C	$C = 1$	1
SETB bit	$\text{Bit} = 1$	1
CPL C	$C = \text{NOT } C$	1
CPL bit	$\text{Bit} = \text{NOT bit}$	1

JC	rel	Nhảy nếu C = 1	2
JNC	rel	Nhảy nếu C = 0	2
JB	bit,rel	Nhảy nếu bit = 1	2
JNB	bit,rel	Nhảy nếu bit = 0	2
JBC	bit,rel	Nhảy nếu bit=1, sau đó xoá bit	2

Bảng 5. Các lệnh chuyển đổi dữ liệu để truy cập RAM ngoài:

Độ lớn địa chỉ	Câu lệnh	Chức năng	Th.gian thực hiện (us)
8 bits	MOVX A,@Ri	Đọc RAM ngoài tại @Ri	2
8 bits	MOVX @Ri,A	Ghi vào RAM ngoài tại @Ri	2
16 bits	MOVX A,@DPTR	Đọc RAM ngoài tại @DPTR	2
16 bits	MOVX @DPTR,A	Ghi vào RAM ngoài tại @DPTR	2

Bảng 6. Các lệnh chuyển Byte mã lệnh:

Câu lệnh	Chức năng	Thời gian thực hiện
MOVC A,@A+DPTR	Đọc ROM tại (A+DPTR)	2 us
MOVC A,@A+PC	Đọc ROM tại (A+PC)	2 us

Bảng 7. Các lệnh nhảy không điều kiện trong Flash Microcontrollers:

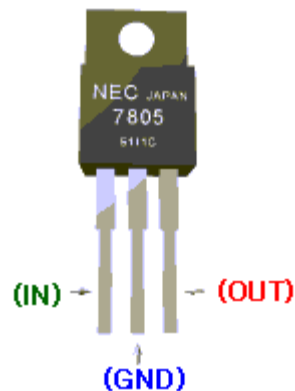
Câu lệnh	Chức năng	Thời gian thực hiện(us)
JMP addr	Nhảy tới addr.	2
JMP @A+DPTR	Nhảy tới A+DPTR.	2
CALL addr	Gọi C.trình con tại addr.	2
RET	Quay trở về từ C.trình con.	2
RETI	Quay trở về từ ngắt.	2

Bảng 8. Các lệnh nhảy có điều kiện:

Câu lệnh	Chức năng	Các kiểu định địa chỉ				Thời gian Thực hiện(us)
		Trực tiếp	Gián tiếp	Th/ghi	Tức thời	
JZ rel	Nhảy nếu A=0	Chỉ với A				2
JNZ rel	Nhảy nếu A≠0	Chỉ với A				2
DJNZ <byte>,rel	Giảm & nhảy nếu ≠ 0	x		x		2
CJNE A,<byte>,rel	Nhảy nếu A ≠ <byte>	x			x	2
CJNE <byte>,#data,rel	Nhảy nếu <byte> ≠ #data		x	x		2

2.1.2. IC tạo ổn áp 7805:(IC ổn áp 5v).

Với những mạch điện không đòi hỏi độ ổn định của điện áp quá cao, sử dụng IC ổn áp thường được người thiết kế sử dụng vì mạch điện khá đơn giản. Các loại ổn áp thường được sử dụng là IC 78xx, với xx là điện áp cần ổn áp. Ví dụ 7805 ổn áp 5V, 7812 ổn áp 12V. Việc dùng các loại IC ổn áp 78xx tương tự nhau, dưới đây là minh họa cho IC ổn áp 7805:



Sơ đồ phía dưới IC 7805 có 3 chân:

- * Chân số 1 là chân IN.
- * Chân số 2 là chân GND.
- * Chân số 3 là chân OUT.

Ngõ ra OUT luôn ổn định ở 5V dù điện áp từ nguồn cung cấp thay đổi. Mạch này dùng để bảo vệ những mạch điện chỉ hoạt động ở điện áp 5V (các loại IC thường hoạt động ở điện áp này). Nếu nguồn điện có sự cố đột ngột: điện áp tăng cao thì mạch điện vẫn hoạt động ổn định nhờ có IC 7805 vẫn giữ được điện áp ở ngõ ra OUT 5V không đổi.

Mạch trên lấy nguồn một chiều từ một máy biến áp với điện áp từ 7V đến 9V để đưa vào ngõ IN. Khi kết nối mạch điện, do nhiều nguyên nhân, người dùng dễ nhầm lẫn cực tính của nguồn cung cấp khi đấu nối vào mạch, trong trường hợp này rất dễ ảnh hưởng đến các linh kiện trên board mạch. Vì lí do đó một diode cầu được lắp thêm vào mạch, diode cầu đảm bảo cực tính của nguồn cấp cho mạch theo một chiều duy nhất, và người dùng cũng không cần quan tâm đến cực tính của nguồn khi nối vào ngõ IN nữa.

2.1.3. IC tạo thời gian thực DS1307:

a) Giới thiệu chung về DS1307:

IC thời gian thực là họ vi điều khiển của hãng dalat. DS1307 có một số đặc trưng cơ bản sau:

- DS1307 là IC thời gian thực với nguồn cung cấp nhỏ dùng để cập nhật thời gian và ngày tháng
- SRAM :56bytes
- Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều
- DS1307 có một mạch cảm biến điện áp dùng để dò các điện áp lỗi và tự động đóng ngắt với nguồn pin cung cấp 3V:

+ DS1307 có 7 byte dữ liệu nằm từ địa chỉ 0x00 tới 0x06, 1 byte điều khiển, và 56 byte lưu trữ (dành cho người sử dụng).

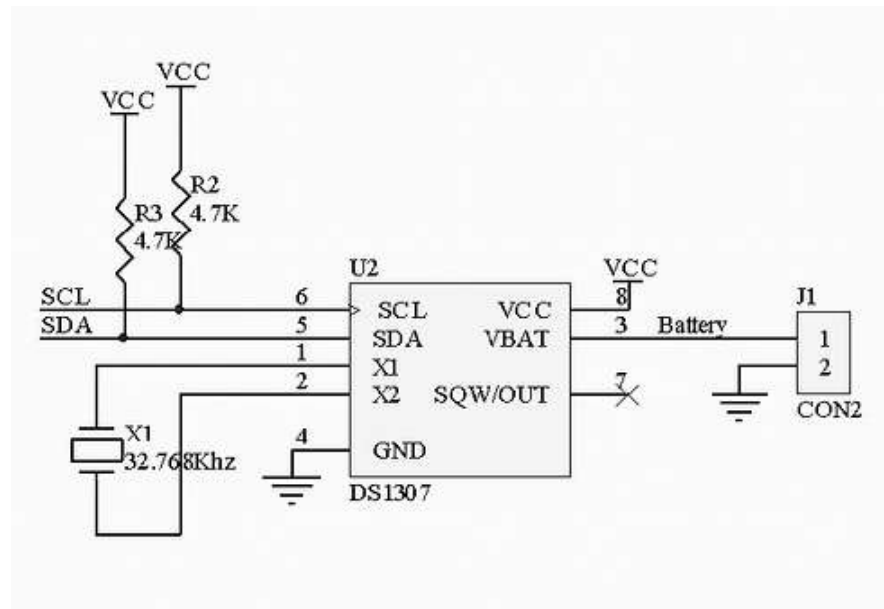
+ Khi xử lý dữ liệu từ DS1307, họ đã tự chuyển cho ta về dạng số BCD, ví dụ như ta đọc được dữ liệu từ địa chỉ 0x04 (tương ứng với Day- ngày trong tháng) và tại 0x05 (tháng) là 0x15, 0x11.

+ Lưu ý đến vai trò của chân SQW/OUT. Đây là chân cho xung ra của DS1307 có 4 chế độ 1Hz, 4.096Hz, 8.192Hz, 32.768Hz... các chế độ này được quy định bởi các bit của thanh ghi Control Register (địa chỉ 0x07).

+ Địa chỉ của DS1307 là 0xD0.

- Cơ chế hoạt động : DS1307 hoạt động với vai trò slave trên đường bus nối tiếp. Việc truy cập được thi hành với chỉ thị start và một mã thiết bị nhất định được cung cấp bởi địa chỉ các thanh ghi. Tiếp theo đó các thanh ghi sẽ được truy cập liên tục đến khi chỉ thị stop được thực thi.

b) Cơ chế hoạt động và chức năng của DS1307:



Vcc: nối với nguồn

X1,X2: nối với thạch anh 32,768 kHz

Vbat: đầu vào pin 3V

GND: đất

SDA: chuỗi data

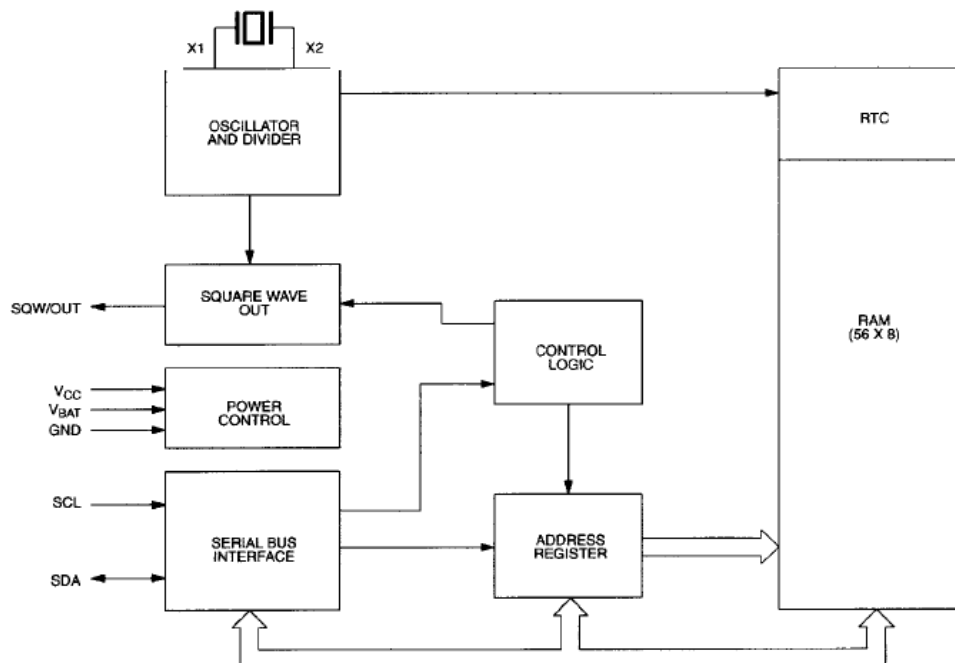
SCL: dãy xung clock

SQW/OUT: xung vuông/đầu ra driver

- DS1307 là một IC thời gian thực với nguồn cung cấp nhỏ, dùng để cập nhật thời gian và ngày tháng với 56 bytes SRAM. Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều. Nó cung cấp thông tin về giờ, phút, giây, thứ, ngày, tháng, năm. Ngày cuối tháng sẽ tự động được điều chỉnh với các tháng nhỏ hơn 31 ngày, bao gồm cả việc tự động nhảy năm. Đồng hồ có thể hoạt động ở dạng 24h hoặc 12h với chỉ thị AM/PM. DS1307 có một mạch cảm biến điện áp dùng để dò các điện áp lỗi và tự động đóng ngắt với nguồn pin cung cấp.

- DS 1307 hoạt động với vai trò slave trên đường bus nối tiếp. Việc truy cập được thi hành với chỉ thị START và một mã thiết bị nhất định được cung cấp bởi địa chỉ các thanh ghi. Tiếp theo đó các thanh ghi sẽ được truy cập liên tục đến khi chỉ thị STOP được thực thi.

Sơ đồ khối của DS1307:



***Mô tả hoạt động của các chân:**

- Vcc,GND: nguồn một chiều được cung cấp tới các chân này. Vcc là đầu vào 5V. Khi 5V được cung cấp thì thiết bị có thể truy cập hoàn chỉnh và dữ liệu có thể đọc và viết. Khi pin 3 V được nối tới thiết bị này và Vcc nhỏ hơn 1,25Vbat thì quá trình đọc và viết không được thực thi,tuy nhiên chức năng timekeeping không bị ảnh hưởng bởi điện áp vào thấp. Khi Vcc nhỏ hơn Vbat thì RAM và timekeeper sẽ được ngắt tới nguồn cung cấp trong (thường là nguồn 1 chiều 3V)

- Vbat: Đầu vào pin cho bất kỳ một chuẩn pin 3V . Điện áp pin phải được giữ trong khoảng từ 2,5 đến 3V để đảm bảo cho sự hoạt động của thiết bị.

- SCL(serial clock input): SCL được sử dụng để đồng bộ sự chuyển dữ liệu trên đường dây nối tiếp.

- SDA(serial data input/out): là chân vào ra cho 2 đường dây nối tiếp. Chân SDA thiết kế theo kiểu cực máng hở , đòi hỏi phải có một điện trở kéo trong khi hoạt động.

- SQW/OUT(square wave/output driver)- khi được kích hoạt thì bit SQWE được thiết lập 1 chân SQW/OUT phát đi 1 trong 4 tần số (1Hz,4kHz,8kHz,32kHz). Chân này cũng được thiết kế theo kiểu cực máng hở vì vậy nó cũng cần có một điện trở kéo trong. Chân này sẽ hoạt động khi cả Vcc và Vbat được cấp.

- X1,X2: được nối với một thạch anh tần số 32,768kHz. Là một mạch tạo dao động ngoài, để hoạt động ổn định thì phải nối thêm 2 tụ 33pF .

- Cũng có DS1307 với bộ tạo dao động trong tần số 32,768kHz, với cấu hình này thì chân X1 sẽ được nối vào tín hiệu dao động trong còn chân X2 thì để hở.

c) Sơ đồ địa chỉ RAM và RTC:

Seconds	00h
Minutes	
Hours	
Day	
Date	
Month	
Year	
Control	07h
Ram 58x8	08h
	3Fh

• Thông tin về thời gian và ngày tháng được lấy ra bằng cách đọc các byte thanh ghi thích hợp. thời gian và ngày tháng được thiết lập cũng thông qua các byte thanh ghi này bằng cách viết vào đó những giá trị thích hợp. nội dung của các thanh ghi dưới dạng mã BCD(binary coded decreaseimal). Bit 7 của thanh ghi seconds là bit clock halt(CH),khi bit này được thiết lập 1 thì dao động disable, khi nó được xoá về 0 thì dao động được enable. Chú ý: enable dao động trong suốt quá trình cấu hình thiết lập (CH=0).Thanh ghi thời gian thực được mô tả như sau:

	BIT7									BIT0
00H	CH	10 SECONDS			SECONDS				00-59	
	X	10 MINUTES			MINUTES				00-59	
	X	12 24	10 HR A/P	10 HR	HOURS				01-12 00-23	
	X	X	X	X	X	DAY				1-7
	X	X	10 DATE		DATE				01-28/29 01-30 01-31	
	X	X	X	10 MONTH	MONTH				01-12	
	10 YEAR				YEAR				00-99	
07H	OUT	X	X	SQWE	X	X	RS1	RS0		

- DS1307 có thể chạy ở chế độ 24h cũng như 12h. Bit thứ 6 của thanh ghi hours là bit chọn chế độ 24h hoặc 12h. khi bit này ở mức cao thì chế độ 12h được chọn. ở chế độ 12h thì bit 5 là bit AM/PM với mức cao là là PM. ở chế độ 24h thì bit 5 là bit chỉ 20h(từ 20h đến 23h).

- Trong quá trình truy cập dữ liệu, khi chỉ thị START được thực thi thì dòng thời gian được truyền tới một thanh ghi thứ 2, thông tin thời gian sẽ được đọc từ thanh ghi thứ cấp này, trong khi đó đồng hồ vẫn tiếp tục chạy. Trong DS1307 có một thanh ghi điều khiển để điều khiển hoạt động của chân SQW/OUT :

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUT	X	X	SQWE	X	X	RS1	RS0

- OUT(output control): bit này điều khiển mức ra của chân SQW/OUT khi đầu ra xung vuông là disable. Nếu SQWE = 0 thì mức logic ở chân SQW/OUT sẽ là 1 nếu OUT=1 và OUT = 0 nếu OUT = 0 .

- SQWE(square wave enable): bit này được thiết lập 1 sẽ enable đầu ra của bộ tạo dao động. Tần số của đầu ra sóng vuông phụ thuộc vào giá trị của RS1 và RS0.

RS1	RS0	tần số đầu ra SQW
0	0	1Hz
0	1	4,096kHz
1	0	8,192kHz
1	1	32,768kHz

DS1307 hỗ trợ bus 2 dây 2 chiều và giao thức truyền dữ liệu. thiết bị gửi dữ liệu lên bus được gọi là bộ phát và thiết bị nhận gọi là bộ thu. thiết bị điều khiển quá trình này gọi là master. thiết bị nhận sự điều khiển của master

gọi là slave. Các bus nhận sự điều khiển của master, là thiết bị phát ra chuỗi xung clock(SCL), master sẽ điều khiển sự truy cập bus, tạo ra các chỉ thị START và STOP.

d) Sự truyền nhận dữ liệu trên chuỗi bus 2 dây :

Tùy thuộc vào bit R/ w mà 2 loại truyền dữ liệu sẽ được thực thi:

- Truyền dữ liệu từ master truyền và slave nhận: Master sẽ truyền byte đầu tiên là địa chỉ của slave. Tiếp sau đó là các byte dữ liệu . slave sẽ gửi lại bit thông báo đã nhận được (bit acknowledge) sau mỗi byte dữ liệu nhận được. dữ liệu sẽ truyền từ bit có giá trị nhất (MSB).

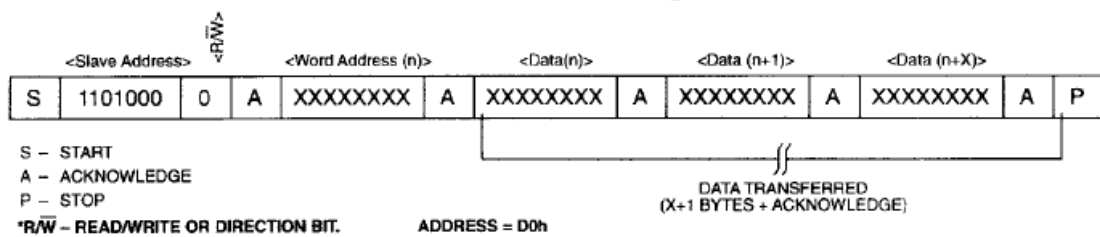
- Truyền dữ liệu từ slave và master nhận: byte đầu tiên (địa chỉ của slave) được truyền tới slave bởi master. Sau đó slave sẽ gửi lại master bit acknowledge. tiếp theo đó slave sẽ gửi các byte dữ liệu tới master. Master sẽ gửi cho slave các bit acknowledge sau mỗi byte nhận được trừ byte cuối cùng, sau khi nhận được byte cuối cùng thì bit acknowledge sẽ không được gửi.

Master phát ra tất cả các chuỗi xung clock và các chỉ thị START và STOP. sự truyền sẽ kết thúc với chỉ thị STOP hoặc chỉ thị quay vòng START. Khi chỉ thị START quay vòng thì sự truyền chuỗi dữ liệu tiếp theo được thực thi và các bus vẫn chưa được giải phóng. Dữ liệu truyền luôn bắt đầu bằng bit MSB.

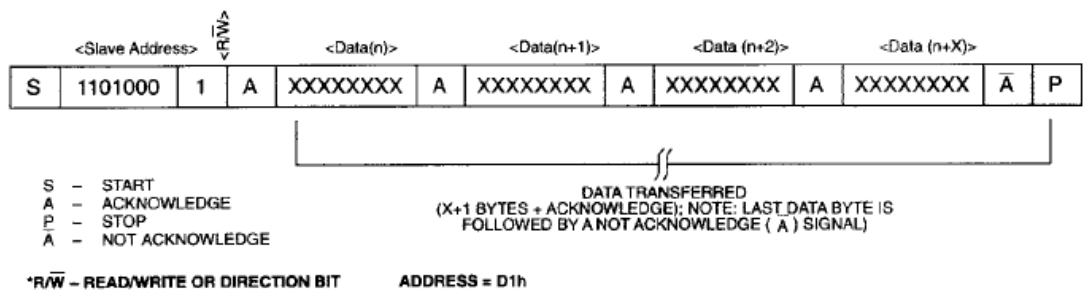
e) DS1307 có thể hoạt động ở 2 chế độ sau:

- Chế độ slave nhận(chế độ DS1307 ghi): chuỗi dữ liệu và chuỗi xung clock sẽ được nhận thông qua SDA và SCL. Sau mỗi byte được nhận thì 1 bit acknowledge sẽ được truyền. các điều kiện START và STOP sẽ được nhận dạng khi bắt đầu và kết thúc một truyền 1 chuỗi. nhận dạng địa chỉ được thực hiện bởi phần cứng sau khi chấp nhận địa chỉ của slave và bit chiều. Byte địa chỉ là byte đầu tiên nhận được sau khi điều kiện START được phát ra từ master. Byte địa chỉ có chứa 7 bit địa chỉ của DS1307, là 1101000, tiếp theo

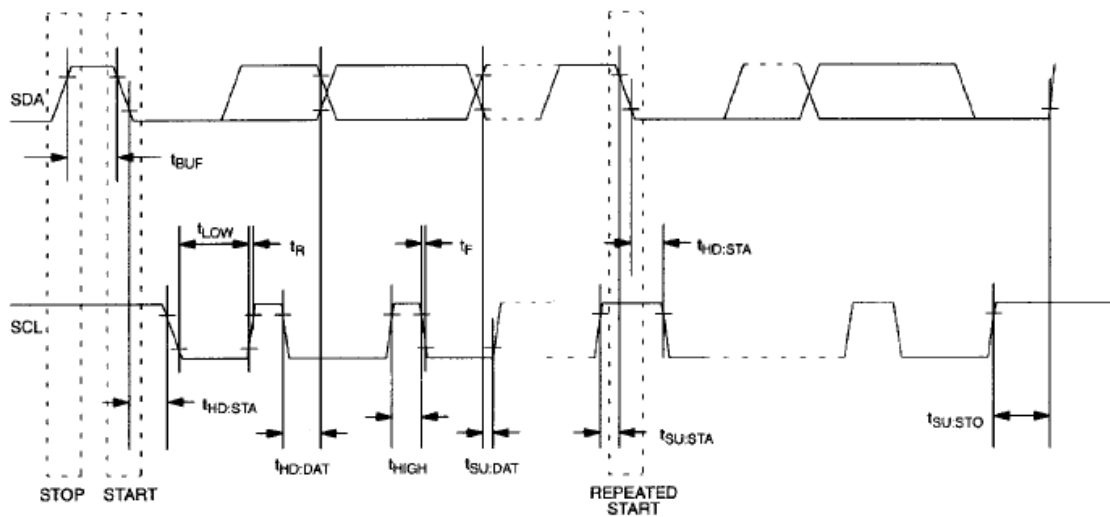
đó là bit chiều (R/ w) cho phép ghi khi nó bằng 0. sau khi nhận và giải mã byte địa chỉ thì thiết bị sẽ phát đi 1 tín hiệu acknowledge lên đường SDA. Sau khi DS1307 nhận dạng được địa chỉ và bit ghi thì master sẽ gửi một địa chỉ thanh ghi tới DS1307 , tạo ra một con trỏ thanh ghi trên DS1307 và master sẽ truyền từng byte dữ liệu cho DS1307 sau mỗi bit acknowledge nhận được. sau đó master sẽ truyền điều kiện STOP khi việc ghi hoàn thành.



- Chế độ slave phát (chế độ DS1307 đọc): byte đầu tiên slave nhận được tương tự như chế độ slave ghi. Tuy nhiên trong chế độ này thì bit chiều lại chỉ chiều truyền ngược lại. Chuỗi dữ liệu được phát đi trên SDA bởi DS 1307 trong khi chuỗi xung clock vào chân SCL. Các điều kiện START và STOP được nhận dạng khi bắt đầu hoặc kết thúc truyền một chuỗi. byte địa chỉ nhận được đầu tiên khi master phát đi điều kiện START. Byte địa chỉ chứa 7 bit địa chỉ của slave và 1 bit chiều cho phép đọc là 1. Sau khi nhận và giải mã byte địa chỉ thì thiết bị sẽ nhận 1 bit acknowledge trên đường SDA. Sau đó DS1307 bắt đầu gửi dữ liệu tới địa chỉ con trỏ thanh ghi thông qua con trỏ thanh ghi. Nếu con trỏ thanh ghi không được viết vào trước khi chế độ đọc được thiết lập thì địa chỉ đầu tiên được đọc sẽ là địa chỉ cuối cùng chứa trong con trỏ thanh ghi .DS1307 sẽ nhận được một tín hiệu Not Acknowledge khi kết thúc quá trình đọc. Đọc dữ liệu-chế độ slave phát.



- Thời gian thực hiện việc đọc, ghi dữ liệu của DS1307: sơ đồ đồng bộ:



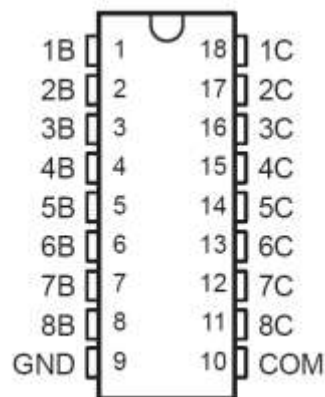
đặc tính và thời gian thực hiện:

(0°C to 70°C or -40°C to +85°C; V_{CC} = 4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	kHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			μ s	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			μ s	5
LOW Period of SCL Clock	t_{LOW}	4.7			μ s	
HIGH Period of SCL Clock	t_{HIGH}	4.0			μ s	
Set-up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			μ s	
Data Hold Time	$t_{HD:DAT}$	0			μ s	6, 7
Data Set-up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set-up Time for STOP Condition	$t_{SU:STO}$	4.7			μ s	
Capacitive Load for each Bus Line	C_B			400	pF	8
I/O Capacitance	C_{IO}		10		pF	
Crystal Specified Load Capacitance			12.5		pF	

2.1.4 IC Đệm ULN 2803

Sơ đồ chân của IC ULN2803:

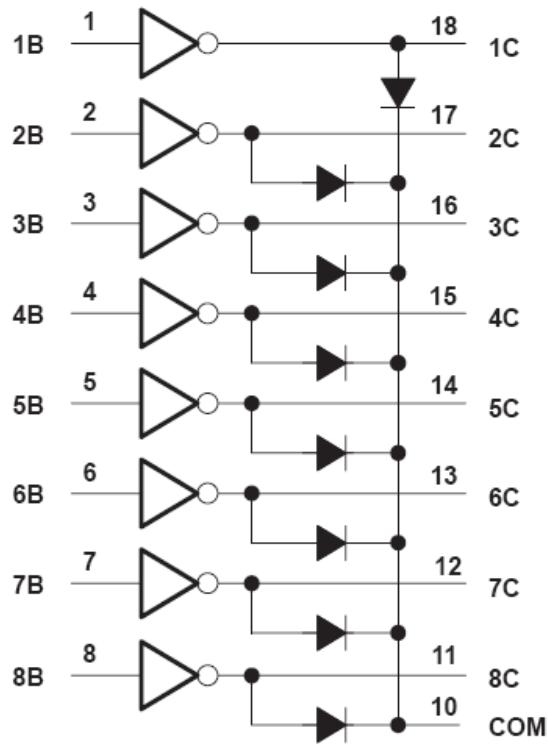


Đặc tính của ULN2803:

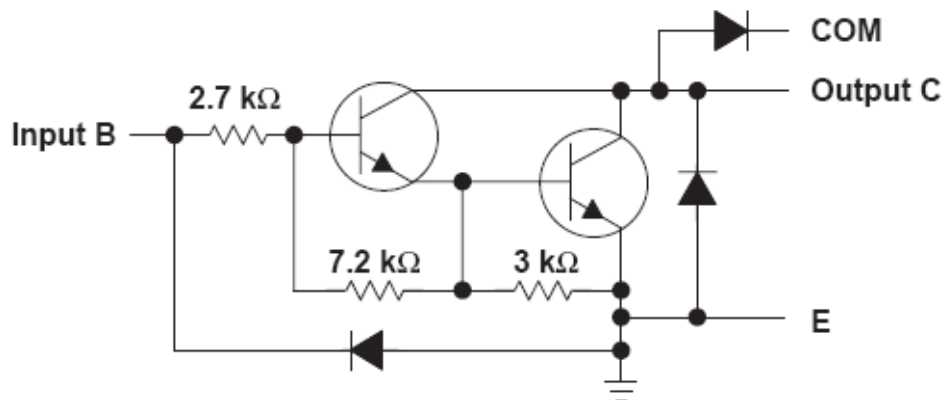
- Dòng đầu ra :500 mA
- Đầu ra điện áp cao: 50V
- Thích hợp với nhiều kiểu logic
- Ứng dụng điều khiển rơle

- Thích hợp với series ULN2800

- Sơ đồ logic:



- Sơ đồ nguyên lý cấu tạo trong của ULN2803: (1 cặp darlington)

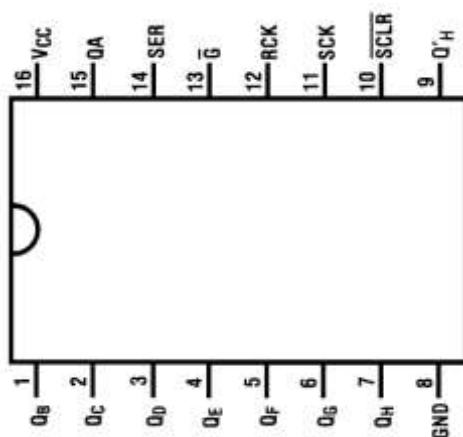


Đầu vào được mở với điện áp là 3.85V, dòng là 1.85mA; Đầu ra có thể đưa 500mA xuống mass

2.1.5 IC ghi dịch 74HC595

IC ghi dịch 74HC595 dùng công nghệ CMOS tiên tiến. IC này sở hữu sự chống nhiễu và tiêu thụ điện áp thấp. Thiết bị này chứa đựng 8 mẫu tuần tự song song kiểu D.

- Sơ đồ chân của IC 74HC595:



- Chức năng các chân của IC 74HC595:

- V_{CC} , GND: dùng cấp nguồn cho IC hoạt động. V_{CC} được nối đến cực dương của nguồn (+5V do là IC họ TTL), GND được nối đến cực âm của nguồn (0V)

- SER : Đường địa chỉ vào của IC 74HC595

- RCK: Chân chốt địa chỉ vào

- SCK : Chân clock

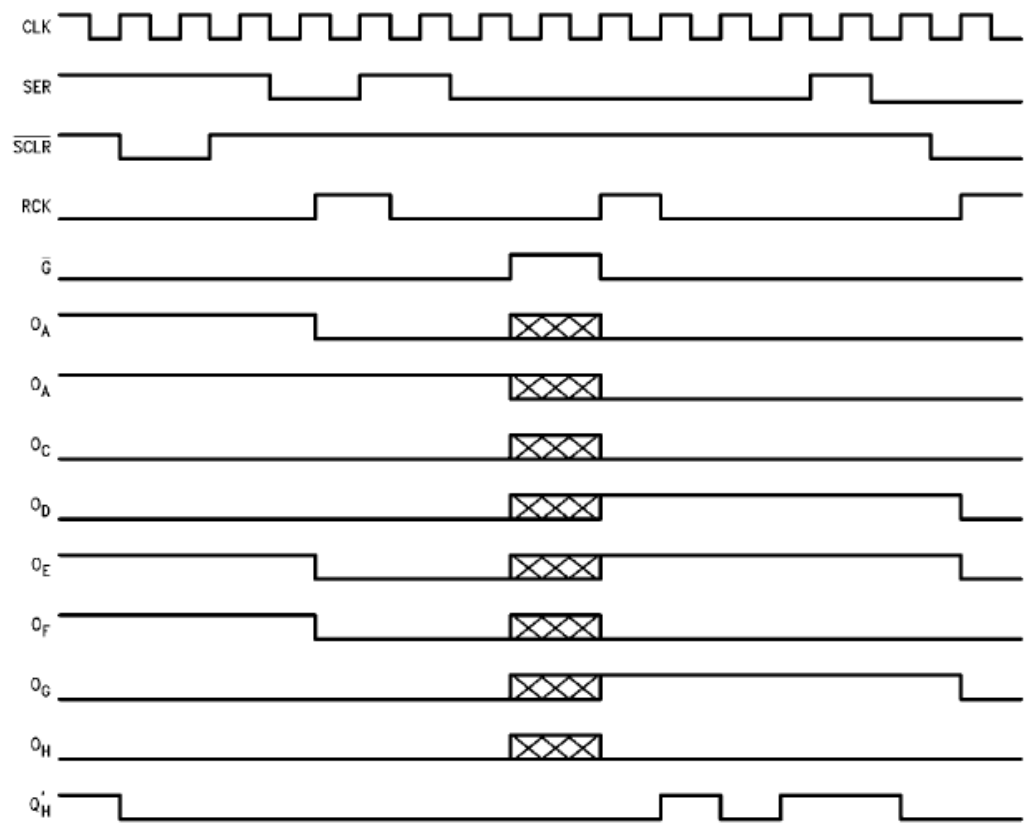
- \overline{SCLR} , \overline{G} : 2 chân vào điều khiển IC. IC sẽ được hoạt động ở trạng thái ghi dịch nếu \overline{SCLR} ở mức logic cao, \overline{G} ở mức logic thấp.

- $O_A \rightarrow O_H$: Các chân ra của IC

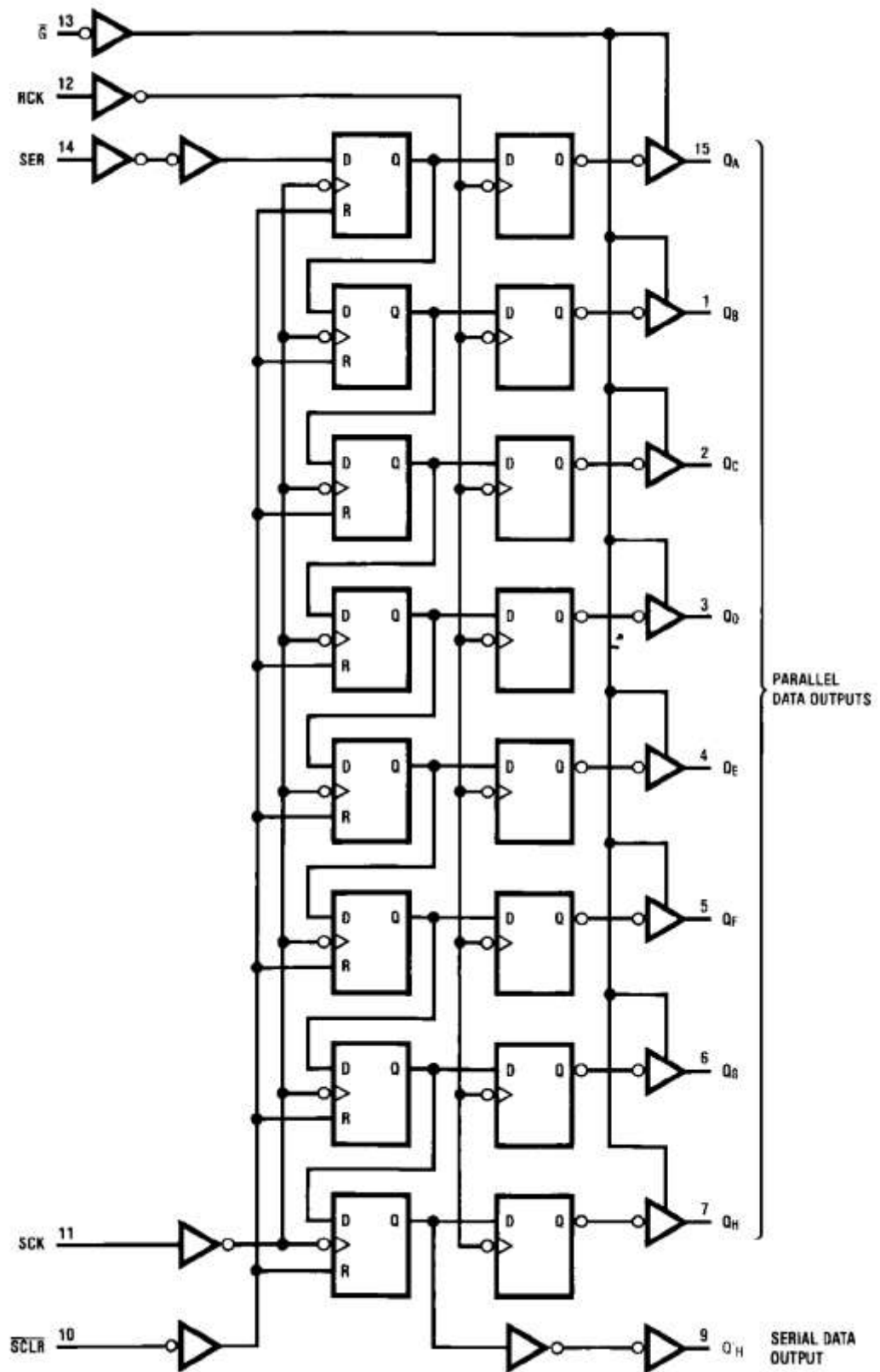
- Bảng hoạt động của IC 74HC595 :

RCK	SCK	\overline{SCLR}	\overline{G}	Chức năng
X	X	X	H	Đầu ra trạng thái trở kháng cao
X	X	L	L	$Q'_H = 0$
X	↑	H	L	Đem thanh ghi dịch $Q_N = Q_{n-1}, Q_0 = SER$
↑	X	H	L	Nội dung của trạng thái đầu ra

- Sơ đồ hoạt động:



- Sơ đồ logic của IC 74HC595:



-Nguyên tắc hoạt động:

Để thực hiện xuất dữ liệu ra led 7 đoạn,AT89C52 điều khiển 74HC595 như sau:

- Tạo một xung ở chân clock SCK
- Xuất 1 byte dữ liệu vào chân data SER
- Tạo một xung vào chân chốt dữ liệu RCK

Lúc này byte dữ liệu sẽ được đưa đến bảng đồng hồ thông qua 8 chân:OA,OB,OC,OD,OE,OF,OG,OH

Việc thực hiện xuất dữ liệu này mang lại cho ta rất nhiều ưu điểm:

-Sử dụng 3 chân của vi điều khiển để truyền dữ liệu nối tiếp thông qua IC74HC595 (so với 8 chân của vi điều khiển nếu thực hiện truyền dữ liệu song song)

-Dòng ra của IC 74HC595 là 7.8 mA, so với 10 uA của AT89C52

2.1.6 LED 7 thanh:

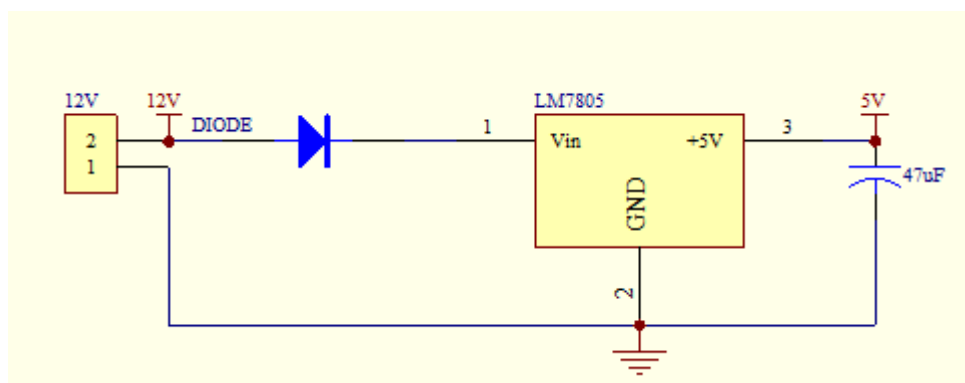
Led 7 thanh bao gồm nhiều loại led tích hợp bên trong các led được nối chung nhau 1 chân .Trong thực tế có 2 loại led 7 thanh là loại anot chung và loại katot chung. Trong chương trình này sử dụng loại anot chung , các led sẽ có chung nhau chân nguồn (chân dương) chân còn lại a,b,c,d,e,f,g của led nào được nối đất thì led đó sẽ sáng.Thay vì sử dụng loại led 7 thanh trên thị trường đang sử dụng,ở đây em thiết kế led 7 thanh được tích hợp bởi nhiều led đơn trong đó.Điện áp đưa vào anot chung là 12 VDC, I_{max} của led 7 thanh là khoảng 1,4A.

Đề ra được các con số tương ứng ta có bảng logic sau:

	a	b	c	d	e	f	G	h
0	0	0	0	0	0	0	1	1
1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	1	0	1
3	0	0	0	0	1	1	0	1
4	1	0	0	1	1	0	0	1
5	0	1	0	0	1	0	0	1
6	0	1	0	0	0	0	0	1
7	0	0	0	1	1	1	1	1
8	0	0	0	0	0	0	0	1
9	0	0	0	0	1	0	0	1

2.2 Các khối mạch cần sử dụng

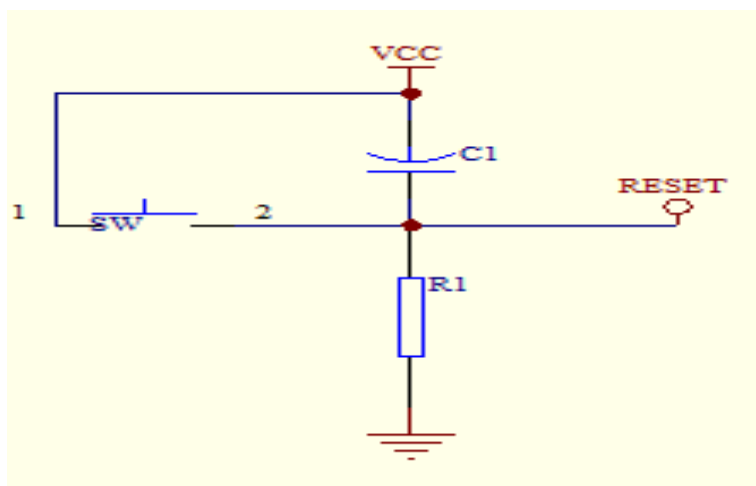
2.2.1 Khối tạo nguồn:



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

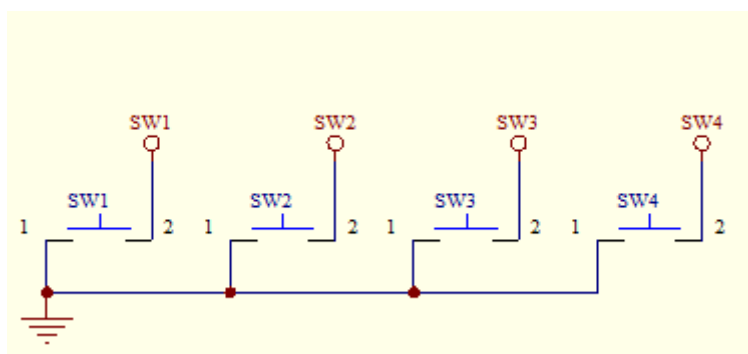
Đầu vào là điện áp 12v DC (Diot trong mạch có nhiệm vụ bảo vệ ngược dòng) sau khi đi qua IC ổn áp 7805 sẽ tạo ra nguồn điện áp chuẩn +5V cung cấp cho mạch.

2.2.2 Khối Reset:



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

2.2.3 Khối điều khiển:



Khối điều khiển gồm 4 Sw ,các chân 1 của các Sw được nối xuống mass,các chân 2 của Sw được nối vào VĐK (p2.0 – p2.3).Khi ta ấn một Sw thì bút được nối với Sw đó sẽ nhận được tín hiệu và thực hiện điều khiển

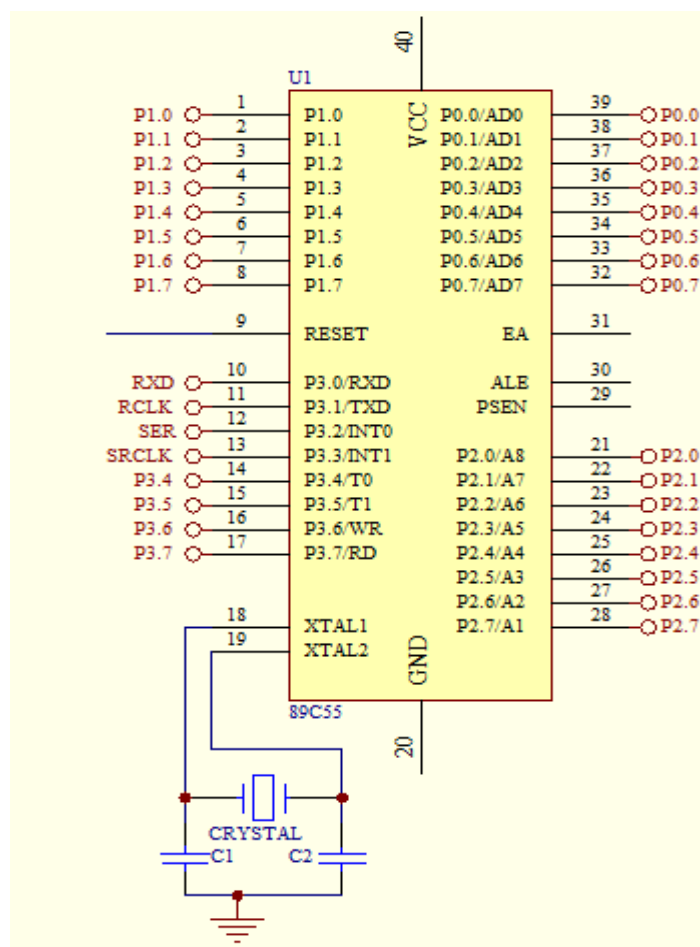
Sw1 : Dừng để chỉnh giờ

Sw2 : Dừng để chỉnh phút

Sw3 : Điều chỉnh tăng

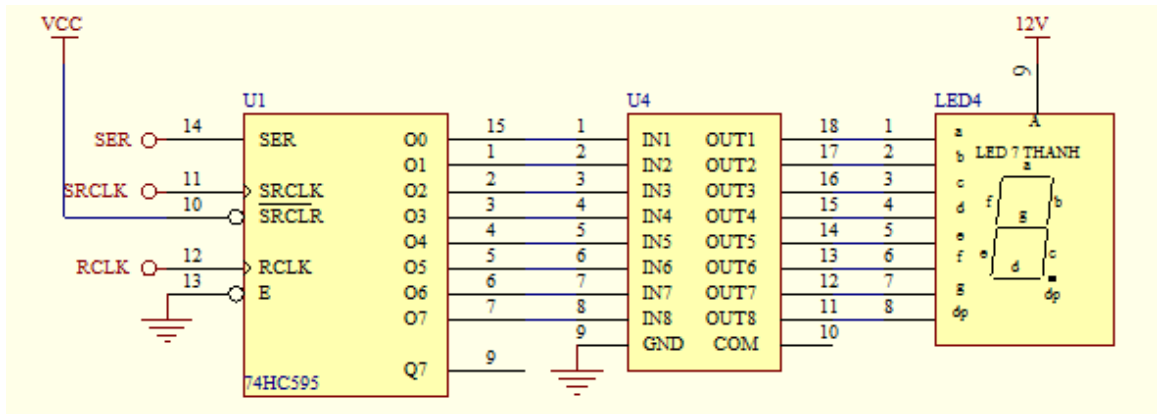
Sw4 : Điều chỉnh giảm

2.2.4 Khối tạo xung dao động:



Đây là bộ dao động thạch anh có tác dụng tạo xung nhịp với tần số 12MHz cho VĐK hoạt động. Hai đầu này được nối vào 2chân XTAL1 và XTAL2 của VĐK.

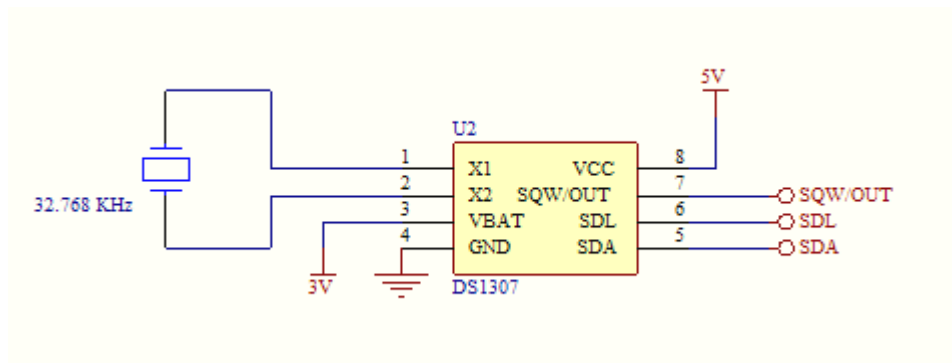
2.2.5 Khôi hiển thị:



Khôi hiển thị bao gồm các LED 7thanh(anode chung). Ở đây ta sử dụng phương pháp điều khiển tĩnh thay cho phương pháp quét thông thường. Ta sẽ điều khiển từng con led 7 thanh thông qua IC 74HC595. Do công suất của led rất lớn (I_{max} của 1 cạnh vào khoảng 200mA) nên ta dùng thêm IC ULN 2803. Với phương pháp điều khiển như vậy, thì mỗi một led 7 thanh ta sử dụng một IC 74HC595 và một IC ULN 2803. Phương pháp này mang lại cho ta rất nhiều ưu điểm :

- Đảm bảo độ sáng tốt của led 7 thanh.
- Không gây ra hiện tượng nhấp nháy như phương pháp quét.

2.2.6 Khôi tạo thời gian thực:

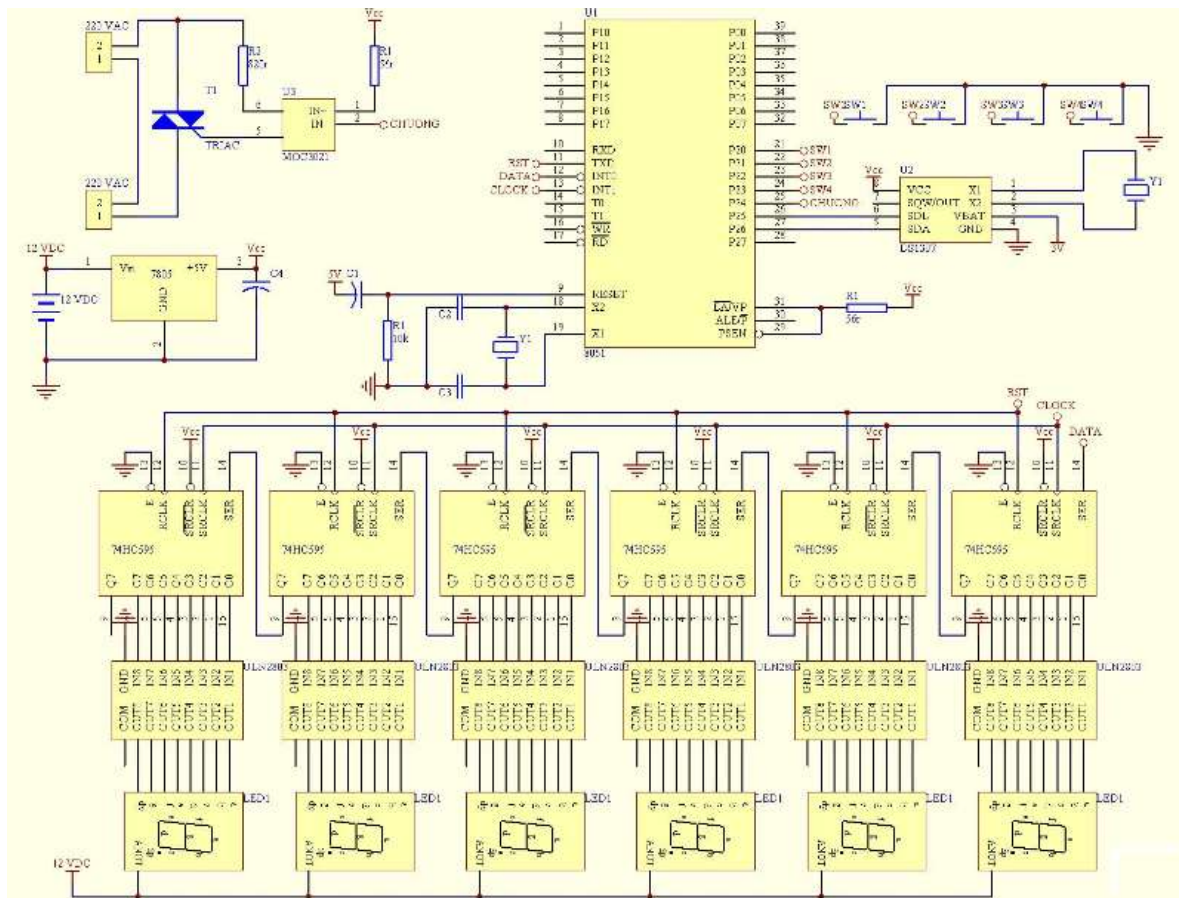


DS1307 là một IC thời gian thực với nguồn cung cấp nhỏ, dùng để cập nhật thời gian và ngày tháng với 56 bytes SRAM. Địa chỉ và dữ liệu được truyền nối tiếp qua 2 đường bus 2 chiều. Nó cung cấp thông tin về giờ, phút, giây, thứ, ngày, tháng, năm. Ngày cuối tháng sẽ tự động được điều chỉnh với các tháng nhỏ hơn 31 ngày, bao gồm cả việc tự động nhảy năm. Đồng hồ có thể hoạt động ở dạng 24h hoặc 12h với chỉ thị AM/PM.

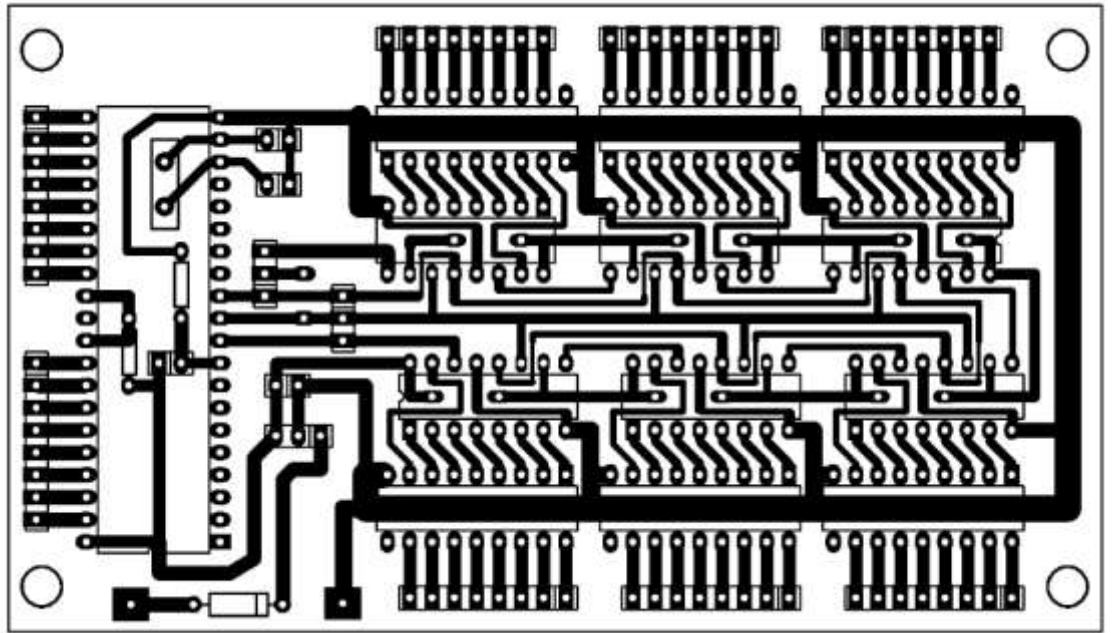
Để không phải điều chỉnh lại thời gian vào những lúc bị mất nguồn, có thể nối thêm 1 pin khoảng 3V vào chân số 3 của IC DS1307 (sao cho chân + của pin nối vào IC và chân - của pin nối xuống đất). Hai chân 1 và 2 của DS1307 được nối vào bộ dao động thạch anh có tần số 32,768KHz để tạo dao động cho IC hoạt động.

2.3 Sơ Đồ Mạch Hoàn Chỉnh

2.3.1 Sơ đồ mạch nguyên lý:

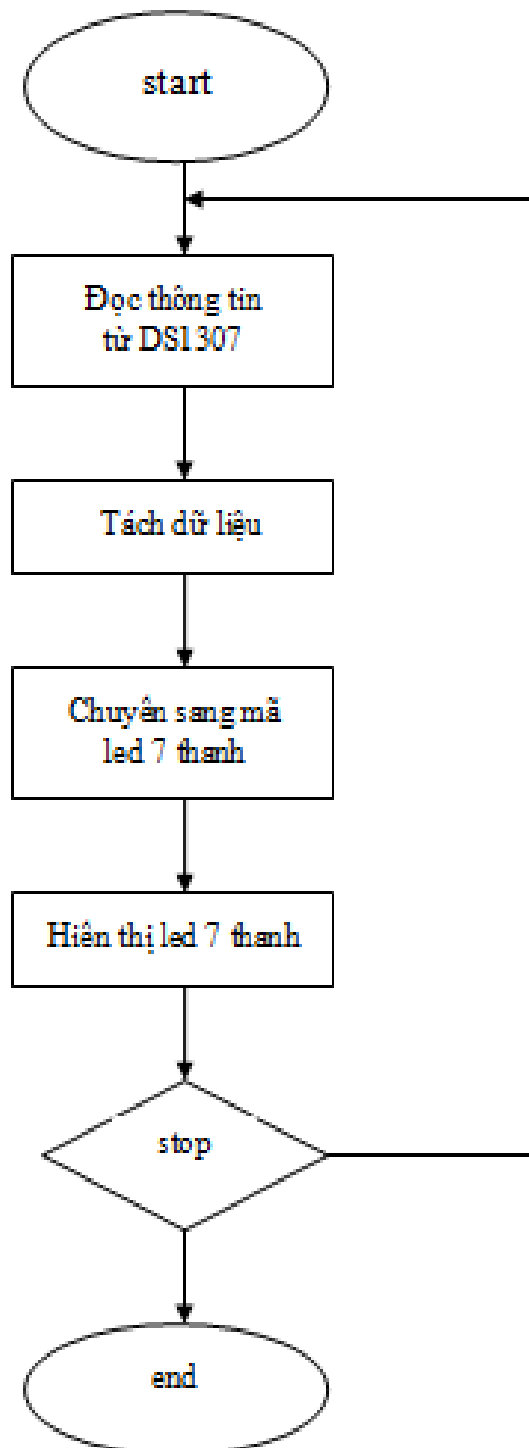


2.3.2 Sơ đồ mạch in:



Chương III: THIẾT KẾ PHẦN MỀM CHO HỆ THỐNG HIỂN THỊ THỜI GIAN THỰC

3.1 Lưu đồ thuật toán:



SQW BIT FLAGS.4
 ACK BIT FLAGS.5
 BUS_FAULT BIT FLAGS.6
 _2W_BUSY BIT FLAGS.7
 BITCOUNT DATA 21H
 BYTECOUNT DATA 22H
 BYTE DATA 23H

; int0 data,int1 clock

org 0000h

.....

.....

SJMP STAUP

ORG 000BH ;T0

AJMP T0ISR

ORG 0030H

STAUP:

MOV SP,#100

MOV TMOD,#01H

MOV TH0,#0

MOV TL0,#1

SETB SDA ; ENSURE SDA HIGH

LCALL SCL_HIGH ; ENSURE SCL HIGH

CLR ACK ; CLEAR STATUS FLAGS

CLR BUS_FAULT

CLR _2W_BUSY

OUT1:

```
RLC A
MOV DT,C
SETB CK
NOP
CLR CK
DJNZ R3,OUT1
;POP ACC
RET
```

;-----

TOISR:

```
CLR TR0
MOV TH0,#0
MOV TL0,#1
Setb 38.0
JNB 38.2,EXIT
DJNZ R7,EXIT
MOV R7,#6
CLR 38.2
```

EXIT:

```
SETB TR0
RETI
```

.....

;-----

; THIS SUB SENDS THE START CONDITION

```

;-----
SEND_START: ;
SETB _2W_BUSY ; INDICATE THAT 2WIRE OPERATION IN PROGRESS
CLR ACK ; CLEAR STATUS FLAGS
CLR BUS_FAULT
JNB SCL,FAULT ; CHECK FOR BUS CLEAR
JNB SDA,FAULT ; BEGIN START CONDITION
SETB SDA ;
LCALL SCL_HIGH ; SDA
CLR SDA
;
LCALL DELAY ; SCL ^START CONDITION
CLR SCL
RET
FAULT:
SETB BUS_FAULT ; SET FAULT STATUS
RET ; AND RETURN
;-----
; THIS SUB SENDS THE STOP CONDITION
;-----
SEND_STOP: ;
CLR SDA ; SDA
LCALL SCL_HIGH ;
SETB SDA ; SCL ^STOP CONDITION
CLR _2W_BUSY
RET
;-----
; THIS SUB SENDS ONE BYTE OF DATA TO THE DS1307
;-----

```



```

SEND_BYTE:
MOV BITCOUNT,#08H ; SET COUNTER FOR 8 BITS
SB_LOOP:
JNB ACC.7,NOTONE ; CHECK TO SEE IF BIT 7 OF ACC IS A 1
SETB SDA ; SET SDA HIGH (1)
JMP ONE
NOTONE:
CLR SDA ; CLR SDA LOW (0)
ONE:
LCALL SCL_HIGH ; TRANSITION SCL LOW-TO-HIGH
RL A ; ROTATE ACC LEFT ONE BIT
CLR SCL ; TRANSITION SCL HIGH-TO-LOW
DJNZ BITCOUNT,SB_LOOP ; LOOP FOR 8 BITS
SETB SDA ; SET SDA HIGH TO LOOK FOR ACKNOWLEDGE PULSE
LCALL SCL_HIGH ; TRANSITION SCL LOW-TO-HIGH
CLR ACK ; CLEAR ACKNOWLEDGE FLAG
JNB SDA,SB_EX ; CHECK FOR ACK OR NOT ACK
SETB ACK ; SET ACKNOWLEDGE FLAG FOR NOT ACK
SB_EX:
LCALL DELAY ; DELAY FOR AN OPERATION
CLR SCL ; TRANSITION SCL HIGH-TO-LOW
LCALL DELAY ; DELAY FOR AN OPERATION
RET
;-----
; THIS SUB READS ONE BYTE OF DATA FROM THE DS1307
;-----
READ_BYTE:
MOV BITCOUNT,#008H ; SET COUNTER FOR 8 BITS OF DATA
MOV A,#00H ;

```

```

SETB SDA ; SET SDA HIGH TO ENSURE LINE FREE

READ_BITS:

LCALL SCL_HIGH ; TRANSITION SCL LOW-TO-HIGH
MOV C,SDA ; MOVE DATA BIT INTO CARRY BIT \
RLC A ; ROTATE CARRY BIT INTO ACC.0
CLR SCL ; TRANSITION SCL HIGH-TO-LOW
DJNZ BITCOUNT,READ_BITS ; LOOP FOR 8 BITS
JB LASTREAD,ACKN ; CHECK TO SEE IF THIS IS THE LAST READ
CLR SDA ; IF NOT LAST READ SEND ACKNOWLEDGE BIT

ACKN:

LCALL SCL_HIGH ; PULSE SCL TO TRANSIMIT ACKNOWLEDGE
CLR SCL ; OR NOT ACKNOWLEDGE BIT

RET

```

```

;-----;
; THIS SUB SETS THE CLOCK LINE HIGH
;-----

```

```

SCL_HIGH:

SETB SCL ; SET SCL HIGH

JNB SCL,$ ; LOOP UNTIL STRONG 1 ON SCL

RET

```

```

;;;;; WRITE time

```

```

WRITE_TIME:

MOV R1,#40h ; POINT TO BEGINNING OF CLOCK DATA IN SCRATCHPAD
MEMORY

LCALL SEND_START ; SEND 2WIRE START CONDITION

MOV A,#DS1307W ; SEND DS1307 WRITE COMMAND

LCALL SEND_BYTE

```

```

MOV A,#00H ; SET DATA POINTER TO REGISTER 00H ON
LCALL SEND_BYTE ; THE DS1307

SEND_LOOP:

MOV A,@R1 ; MOVE THE FIRST BYTE OF DATA TO ACC
LCALL SEND_BYTE ; SEND DATA ON 2WIRE BUT
INC R1

CJNE R1,#48H,SEND_LOOP ; LOOP UNTIL CLOCK DATA SENT TO DS1307
LCALL SEND_STOP ; SEND 2WIRE STOP CONDITION
RET

```

```

;-----
READ_TIME:
READ_AGAIN:
MOV R1,#40H ; START OF CLOCK REG IN SCRATCHPAD
MOV BYTECOUNT,#00H ; COUNTER UP TO 8 BYTES FOR CLOCK
CLR LASTREAD ; FLAG TO CHECK FOR LAST READ
LCALL SEND_START ; SEND START CONDITION
MOV A,#DS1307W ; SET POINTER TO REG 00H ON DS1307
LCALL SEND_BYTE
MOV A,#00H
LCALL SEND_BYTE
LCALL SEND_STOP ; SEND STOP CONDITION
LCALL SEND_START ; SEND START CONDITION
MOV A,#DS1307R ; SEND READ COMMAND TO DS1307
LCALL SEND_BYTE
READ_LOOP:
MOV A,BYTECOUNT ; CHECK TO SEE OF DOING LAST READ
CJNE A,#07H,NOT_LAST
SETB LASTREAD ; SET LASTREAD FLAG

```

```

NOT_LAST:
LCALL READ_BYTE ; READ A BYTE OF DATA
MOV @R1,A ; MOVE DATA IN SCRATCHPAD MEMORY
MOV A,BYTECOUNT ; CHECK TO SEE IF READING SECONDS REG
CJNE A,#00H,NOT_FIRST
CLR OSC ; CLR OSC FLAG
MOV A,@R1 ; MOVE SECONDS REG INTO ACC
JNB ACC.7,NO_OSC ; JUMP IF BIT 7 OF IS A 0
SETB OSC ; SET OSC FLAG, BIT 7 IS A 1
CLR ACC.7 ; CLEAR BIT 7 FOR DISPLAY
; PURPOSES
MOV @R1,A ; MOVE DATA BACK TO SCRATCHPAD
NO_OSC:
NOT_FIRST:
INC R1 ; INC COUNTERS
INC BYTECOUNT
MOV A,BYTECOUNT
CJNE A,#08H,READ_LOOP ; LOOP FOR ENTIRE CLOCK REGISTERS
LCALL SEND_STOP ; SEND 2WIRE STOP CONDITION
RET
;-----
;-----;
; THIS SUB DELAY THE BUS
;-----
DELAY:
NOP ; DELAY FOR BUS TIMING
RET
;-----
; THIS SUB DELAYS 4 CYCLES

```

```

;-----
DELAY_4:
NOP ; DELAY FOR BUS TIMING
NOP
NOP
NOP
RET

;-----

READKEY:
    JNB KEY1,HOUR
    JNB KEY2,MINUT
    JNB KEY3,INCRE
    JNB KEY4,DECRE
    SJMP EXIT_KEY1
HOUR: SETB 38.1
    SJMP EXIT_KEY1
MINUT: CLR 38.1
    SJMP EXIT_KEY1
INCRE:JB 38.2,EXIT_KEY1 ;DELAY_KEY
    JB 38.1,I_HO
I_MI:INC 41
    SJMP EXIT_KEY
I_HO:INC 42
    SJMP EXIT_KEY
DECRE:JB 38.2,EXIT_KEY1
    JB 38.1,D_HO
D_MI:DEC 41
    SJMP EXIT_KEY

```

```
D_HO:DEC 42
      SJMP EXIT_KEY
```

```
EXIT_KEY:
      SETB 38.2
      MOV  R7,#6
      ACALL WRITE_TIME
      ACALL READ_TIME
```

```
EXIT_KEY1:
      RET
```

;------

```
::TACH BCD
```

```
TACH:
      PUSH ACC

      MOV A,40
      ANL A,#0Fh
      MOV 48,A
      MOV A,40
      ANL A,#0F0H
      SWAP A
      MOV 49,A

      MOV A,41
      ANL A,#0Fh
      MOV 50,A
      MOV A,41
      ANL A,#0F0H
```

SWAP A

MOV 51,A

MOV A,42

ANL A,#0Fh

MOV 52,A

MOV A,42

ANL A,#0F0H

SWAP A

MOV 53,A

POP ACC

RET

;=====

=====

DISPLAY1:

push ACC

mov a,54

ACALL OUT

mov a,55

ACALL OUT

mov a,56

ACALL OUT

mov a,57

ACALL OUT

mov a,58

ACALL OUT

mov a,59

ACALL OUT

setb p3.1

setb rst

nop

nop

clr rst

;;;chot du lieu cua 4 ic truyen du lieu

pop ACC

RET

;=====

=====

CONVER_LED:

MOV R6,#6 ;doi 16 bai ma

MOV R0,#48 ;dau vung ma HEX

MOV R1,#54 ;dau vung ma BCD

CONVER_LED_1:

MOV A,@R0

ADD A,#6 ;vuot qua 6bai lenh tiep theo

MOVC A,@A+PC ;de truy cap bang

MOV @R1,A

INC R0

INC R1

DJNZ R6,CONVER_LED_1

RET


```

                ;B7 B6 B5 B4 B3 B2 B1 B0
DB 3FH ;    0: 1  1  0  0  0  0  0  0
DB 06H ;    1: 1  1  1  1  1  1  0  0  1
DB 5BH ;    2: 1  0  1  0  0  0  1  0  0
DB 4FH ;    3: 1  0  1  1  0  0  0  0  0
DB 66H ;    4: 1  0  0  1  1  0  0  0  1
DB 6DH ;    5: 1  0  0  1  0  0  0  1  0
DB 7DH ;    6: 1  0  0  0  0  0  0  1  0
DB 07H ;    7: 1  1  1  1  1  1  0  0  0
DB 7FH ;    8: 1  0  0  0  0  0  0  0  0
DB 6FH ;    9: 1  0  0  1  0  0  0  0  0
DB 00H
DB 00H
DB 00H
end

```

Kết luận

Sau hơn 3 tháng làm đề tài tốt nghiệp, với sự nỗ lực học hỏi của bản thân và sự chỉ bảo tận tình của Thầy giáo Th.s Nguyễn Trọng Thắng, đề tài “**Thiết kế hệ thống hiển thị thời gian thực**” của em đã hoàn thành đúng thời gian và thực hiện thành công hệ thống hiển thị thời gian thực

Qua đề tài này em có những hiểu biết tốt hơn về lĩnh vực vi điều khiển cũng như cách ghép nối các linh kiện điện tử thành một sản phẩm hoàn thiện

Vì thời gian có hạn và kiến thức cùng kinh nghiệm của bản thân còn hạn chế nên không tránh khỏi những sai sót. Rất mong các thầy cô và các bạn xem xét và đóng góp những ý kiến quý báu để cuốn đồ án được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn các thầy cô giáo trong khoa Điện - Điện tử, trường Đại học Dân lập Hải Phòng đã hết lòng truyền đạt những kiến thức quý báu cho chúng em trong suốt những năm học qua. Đặc biệt, em xin cảm ơn sự chỉ bảo tận tình của thầy giáo Th.s Nguyễn Trọng Thắng cùng các bạn đã giúp em hoàn thành đồ án này.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 20 tháng 06 năm 2009

Sinh viên thực hiện:

Vũ Hải Đăng

TÀI LIỆU THAM KHẢO.

1. Ngô Diên Tập (2006) *Vi Điều Khiển Với Lập Trình C*, NXB Khoa học và kỹ thuật.
2. Tống Văn On (2006), *Thiết Kế Hệ Thống Với Họ 8051*, NXB Phương Đông
3. Tống Văn On – Hoàng Đức Hải (2008), *Họ Vi Điều Khiển 8051*, NXB Khoa học Xã Hội.
4. Nguyễn Tấn Phước (2005), *Giáo Trình Điện Tử Công Suất*, NXB TP Hồ Chí Minh