

MỤC LỤC

LỜI MỞ ĐẦU	1
Chương I: TỔNG QUAN CÁC PHƯƠNG ÁN ĐIỀU KHIỂN	2
1.1.ĐẶT VẤN ĐỀ.....	2
1.2 ĐIỀU KHIỂN QUÉT	2
1.2.1. Sơ đồ khối:	2
1.2.2. Nguyên lý hiển thị:.....	2
1.2.3. Ưu điểm và nhược điểm:.....	3
1.3.ĐIỀU KHIỂN ĐIỂM	3
1.3.1 Sơ đồ khối:	3
1.3.2 Nguyên lý hiển thị:.....	4
1. 3.3 Ưu điểm và nhược điểm:.....	4
1.4. XÂY DỰNG PHƯƠNG ÁN TỐI ƯU.....	4
Chương II :THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG CỦA BẢNG THÔNG TIN ĐIỆN TỬ	5
2.1. GIỚI THIỆU LINH KIỆN	5
2.1.1. Khảo sát họ vi điều khiển 8051:	5
2.1.1.1. Cấu trúc bên trong của 8051:.....	6
2.1.1.2. Chức năng các chân vi điều khiển:	7
2.1.1.3. Các thanh ghi đặc biệt:.....	10
2.1.1.4. Khối tạo thời gian và bộ đếm (Timer/Counter)	18
2.1.1.5. Cơ chế ngắt trong On-chip 8051:.....	22
2.1.1.6. Bảo vệ chương trình.....	27
2.1.1.7. Tập lệnh của họ VDK 8051	27
2.1.2. IC 74HC595	54
2.1.2.1. Mô tả chung	54
2.1.2.2. Sơ đồ chân của IC 74HC595:	55

2.1.2.3. Bảng hoạt động của IC 74HC595:	55
2.1.2.4. Sơ đồ hoạt động:	56
2.1.2.5. Sơ đồ logic của IC 74HC595:	57
2.1.2.6. Nguyên tắc hoạt động:	58
2.1.3. IC 74HC573	58
2.1.3.1. Mô tả chung:	58
2.1.3.2. Sơ đồ chân:.....	58
2.1.3.3. Bảng hoạt động của IC 74HC573:.....	59
2.2 THIẾT KẾ MẠCH MASTER	59
2.2.1 Thiết kế mạch hoạt động cho 89c55:	59
2.2.2. Khối xuất dữ liệu hàng:.....	59
2.2.3. Khối ghi dịch và khối hiển thị:	61
2.3. SƠ ĐỒ NGUYÊN LÝ MẠCH MASTER	62
2.4. SƠ ĐỒ MẠCH IN MẠCH MASTER	63
2.5 SƠ ĐỒ MẠCH IN MODUL	64
Chương III : CHƯƠNG TRÌNH PHẦN MỀM.....	65
3.1. LƯU ĐỒ THUẬT TOÁN.....	65
3.2. CHƯƠNG TRÌNH PHẦN MỀM.....	66
KẾT LUẬN	69
TÀI LIỆU THAM KHẢO	70

LỜI NÓI ĐẦU

Kỹ thuật vi điều khiển với tốc độ phát triển nhanh đã và đang mang đến những thay đổi to lớn trong khoa học và công nghệ cũng như trong đời sống hàng ngày. Ngày nay, các thiết bị máy móc ngày càng trở nên thông minh hơn, các công việc được thực hiện với hiệu quả cao hơn, đó cũng là nhờ vi điều khiển.

Kỹ thuật vi điều khiển là kỹ thuật của tương lai, là chìa khóa đi vào công nghệ hiện đại. Đối với sinh viên chuyên ngành điện tử động công nghiệp, đây là một lĩnh vực hứa hẹn và mở ra nhiều triển vọng.

Với tính ưu việt của vi điều khiển, em thực hiện đề tài: “Nghiên cứu xây dựng hệ thống bảng thông tin điện tử”. Đề tài gồm những nội dung sau:

Chương 1: Tổng quan các phương án điều khiển

Chương 2: Thiết kế và thi công phần cứng của bảng thông tin điện tử

Chương 3: Thiết kế chương trình phần mềm

Những kiến thức học được cộng thêm hiểu biết từ các tài liệu tham khảo, tuy có thể hoàn thành cuốn đồ án này nhưng không thể tránh khỏi nhiều thiếu sót mong các thầy cô giáo khi xem cuốn đồ án này có thể thông cảm.

Để hoàn thành cuốn đồ án này, em đã nhận được sự chỉ bảo tận tình của thầy giáo hướng dẫn và sự giúp đỡ nhiệt tình của bạn bè. Cuối cùng em xin cảm ơn thầy giáo hướng dẫn Th.s Nguyễn Trọng Thắng và các thầy cô giáo đã dạy cho em những kiến thức cơ bản để em có thể hoàn thành cuốn đồ án này.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Lê Minh Hòa

Chương 1:

TỔNG QUAN CÁC PHƯƠNG ÁN ĐIỀU KHIỂN

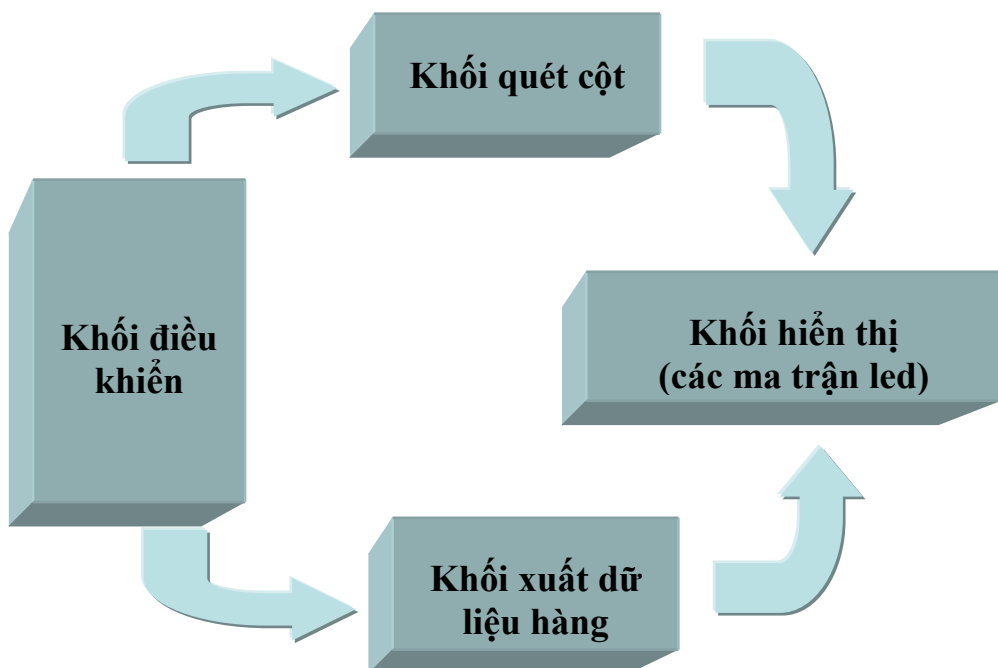
1.1.ĐẶT VẤN ĐỀ:

Trong thời đại hiện nay có rất nhiều hình thức thông tin quảng cáo.Một trong số những hình thức có hiệu quả nhất và đang được sử dụng rộng rãi đó là thông tin quảng cáo bằng bảng điện tử.Để thực hiện điều này người ta dùng ma trận led. Để điều khiển các ma trận led, ta có 2 phương án: Điều khiển quét và điều khiển điểm.

1.2 ĐIỀU KHIỂN QUÉT:

1.2.1. Sơ đồ khối:

Với phương án điều khiển quét thì một bảng quảng cáo điện tử thông thường gồm có bốn khối chính: Khối điều khiển, khối quét cột, khối xuất dữ liệu và khối hiển thị



1.2.2. Nguyên lý hiển thị:

Để hiển thị thông tin lên bảng quảng cáo trong một thời gian nhất định(đủ lâu để mắt người có thể lưu giữ hình ảnh)thì thực chất ta sẽ làm cho

các cột của bảng điện tử sáng lên trong 1 thời điểm nhỏ(quét cột).Tại một thời điểm đó,dữ liệu sẽ xuất phát từ vi điều khiển qua 2 con đường.Đường thứ nhất chứa thông tin về một cột cụ thể nào đó sẽ được phát sáng,qua khối quét cột làm cho cột đó phát sáng còn các cột còn lại sẽ tắt.Đường thứ 2 mang dữ liệu sẽ được phát sáng tại cột đó thông qua khối xuất dữ liệu hàng.Tất cả các thông tin đó tập trung tại bảng ma trận led và nó sẽ hiển thị thông tin trên một cột mà ta mong muốn.Các cột liên tục luân phiên phát sáng với tần số cao nhờ sự lưu ảnh của mắt mà ta thấy được các thông tin trên bảng điện tử một cách liên tục ,không bị chớp tắt

1.2.3. Ưu điểm và nhược điểm:

a. Ưu điểm:

- Do trong mỗi thời điểm chỉ có một cột sáng nên phương án quét tiết kiệm điện

- Xây dựng phần cứng dễ dàng

- Chi phí giá thành sản xuất rẻ

b. Nhược điểm:

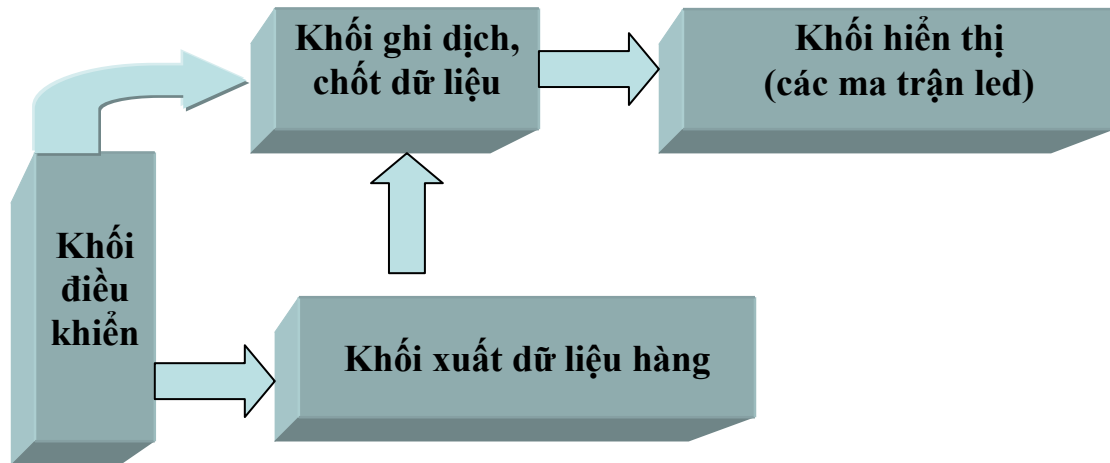
- Tuy rằng tiết kiệm điện năng nhưng phương án quét cho ta độ sáng của bảng điện tử khi đặt ngoài trời không được tốt nên ít sử dụng .

-Với những bảng có kích thước lớn xảy ra hiện tượng rò dòng(có những thông tin sáng lờ mờ không mong muốn xuất hiện trên bảng điện tử)do sự không đồng bộ giữa khối xuất dữ liệu và khối quét cột

1.3. ĐIỀU KHIỂN ĐIỂM

1.3.1 Sơ đồ khối:

Điều khiển điểm _ với tên gọi này ta có thể hiểu đây là phương án điều khiển từng con led trên bảng điện tử.Phương án này bao gồm những khối sau:Khối điều khiển,khối xuất dữ liệu hàng,khối ghi dịch,chốt dữ liệu và khối hiển thị



1.3.2 Nguyên lý hiển thị:

Tại thời điểm ban đầu ,khối xuất dữ liệu nhận dữ liệu từ khối điều khiển đưa đến khối ghi dịch.Đây là dữ liệu của một cột.Khối ghi dịch nhận dữ liệu và lưu dữ liệu này lại.Lúc này khối điều khiển phát xung đến khối ghi dịch,cho phép khối ghi dịch này đưa dữ liệu ra led.Việc điều khiển cứ tuần tự như vậy.

1. 3.3 Ưu điểm và nhược điểm:

a. Ưu điểm:

- Do điều khiển từng led nên phương án điều khiển điểm mang lại cho ta độ sáng tối đa của led

- Không xảy ra hiện tượng rò dòng
- Kích thước của bảng điện tử vô hạn

b. Nhược điểm:

- Tiêu thụ điện năng nhiều
- Phần cứng phức tạp

1.4. XÂY DỰNG PHƯƠNG ÁN TỐI ƯU

Ta nhận thấy rằng điều khiển điểm có tính năng ưu việt hơn so với điều khiển quét.nó thể hiện sự vượt trội về ánh sáng,mỹ thuật .Mặc dù vẫn có những nhược điểm nhưng chúng ta vẫn có thể hạn chế nếu xây dựng được phần cứng tối ưu.Trên cơ sở đó,em tìm hiểu và thiết kế sản phẩm là bảng thông tin điện tử dùng phương pháp điều khiển điểm

Chương 2 :

THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG CỦA BẢNG THÔNG TIN ĐIỆN TỬ

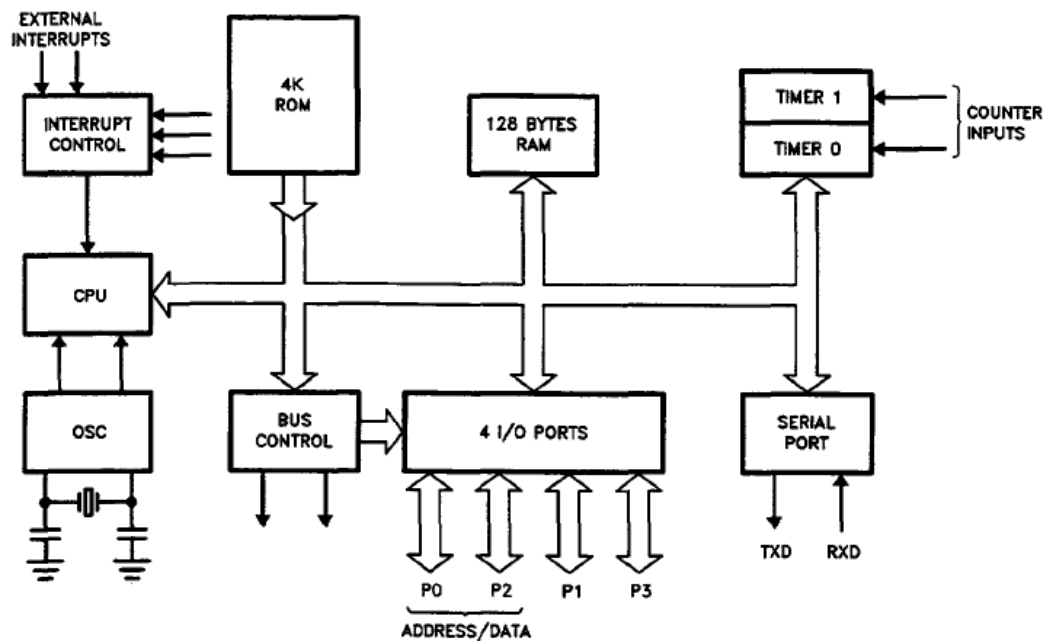
2.1. GIỚI THIỆU CÁC LINH KIỆN

2.1.1. Khảo sát họ 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 đầu nối giao diện nối tiếp, cũng như các đường ngắt dẫn ở bên ngoài

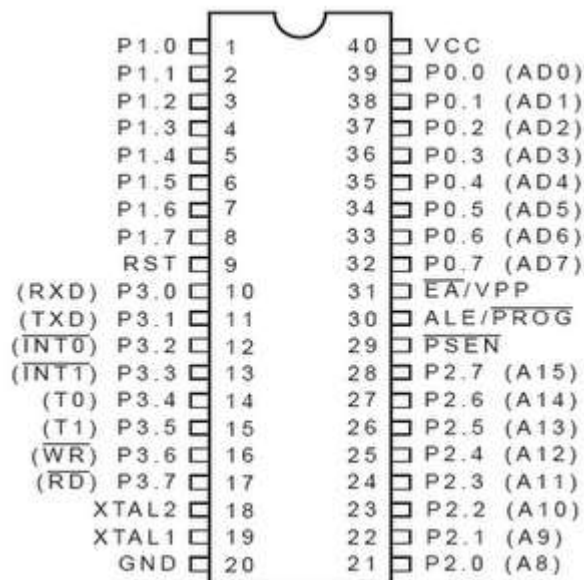
Giao diện nối tiếp có chứa một bộ truyền và một bộ nhận không đồng bộ, làm việc độc lập với nhau. Tốc độ truyền qua cổng nối tiếp có thể đặt trong dải rộng và được ấn định bằng một bộ định thời.

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 vi điều khiển:



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 chuyên 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.

i.Các ngõ vào bộ dao động trên chip : 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.

Figure 14. Using the On-Chip Oscillator

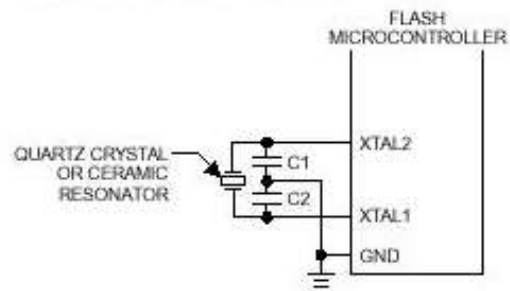
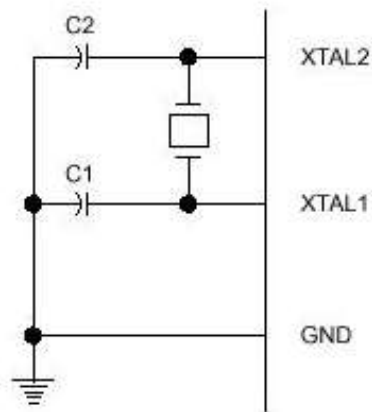


Figure 15. A: Oscillator Connections



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.

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

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

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

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

2.1.1.3.1. 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.

2.1.1.3.2. 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.

2.1.1.3.3. 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.

2.1.1.3.4. 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.

2.1.1.3.5. 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).

2.1.1.3.6. 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.

2.1.1.3.7. 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.

2.1.1.3.8. 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.

2.1.1.3.9. 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)

Bảng 2.5. Chọn băng thanh ghi

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

* 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.

* 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.

2.1.1.3.10. 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 công truyền tin nối tiếp được dùng bởi các kiểu 1, 2 hoặc 3.

* 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.

2.1.1.3.11. 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. 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.

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

* 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.

2.1.1.3.13. 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.

2.1.1.3.14. Thanh ghi TMOD: Thanh ghi điều khiển kiểu Timer/Counter

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

2.1.1.3.15. 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.

Bảng 2.6. Chọn Mode trong SCON

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)

* 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. Khối 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.

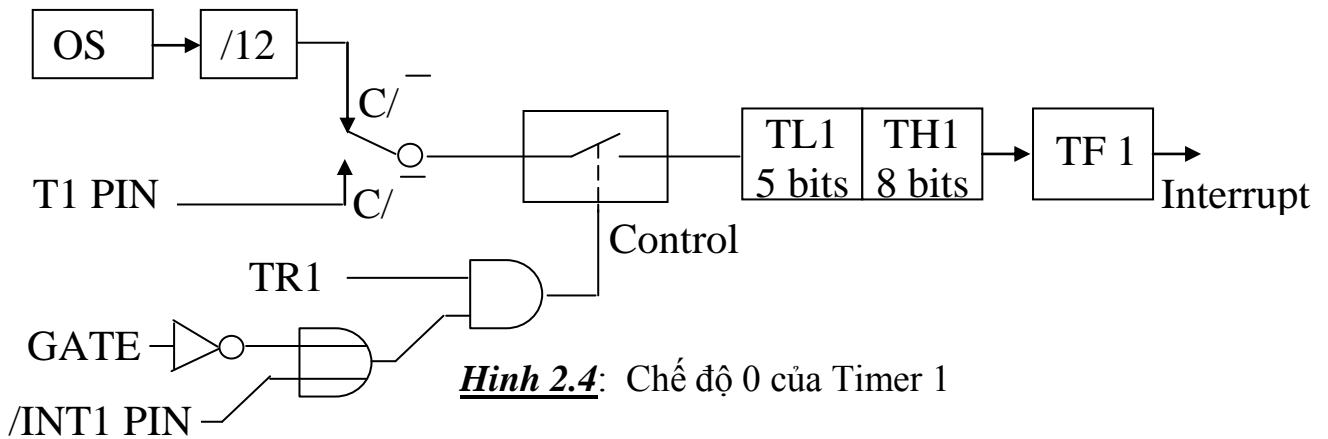
Do xung nhịp bên ngoài có tần số bất kỳ nên các bộ Timer (0 và 1) có 4 chế độ làm việc khác nhau để lựa chọn: (13 bit Timer, 16 bit Timer, 8 bit auto-reload, split Timer).

Timer 0 và Timer 1:

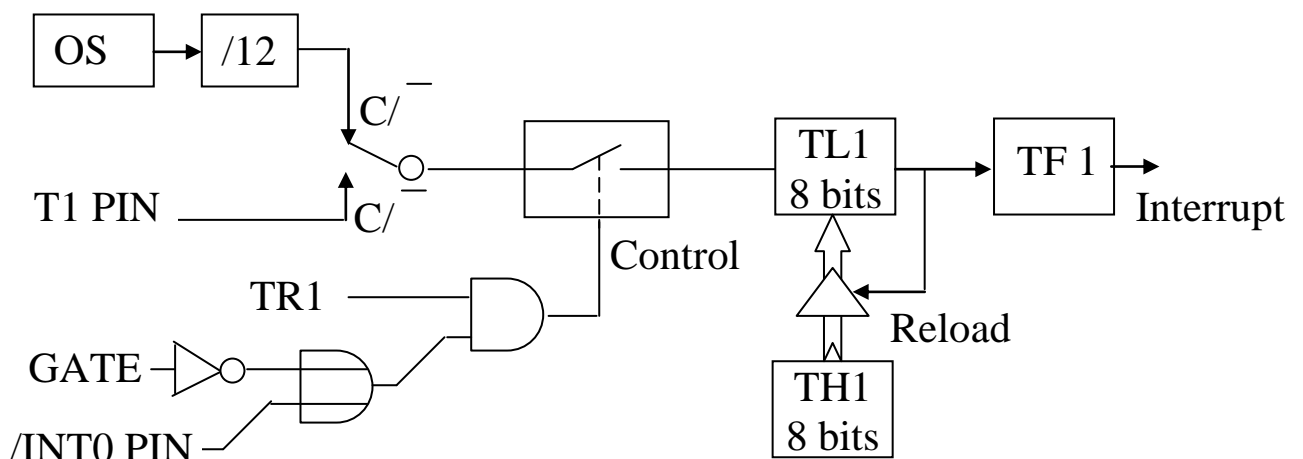
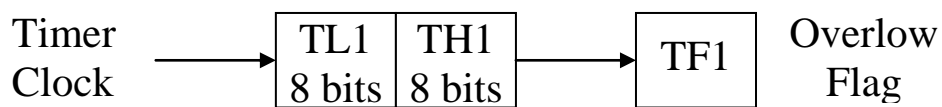
Trong AT89C51 và AT89C52 đều có các bộ Timer 0 và 1. Chức năng Timer hay Counter được chọn lựa bởi các bit điều khiển C/(/T) trong thanh ghi TMOD. Hai bộ Timer/Counter này có 4 chế độ hoạt động, được lựa chọn bởi cặp bit (M0, M1) trong TMOD. Chế độ 0, 1 và 2 giống nhau cho các chức năng Timer/Counter,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



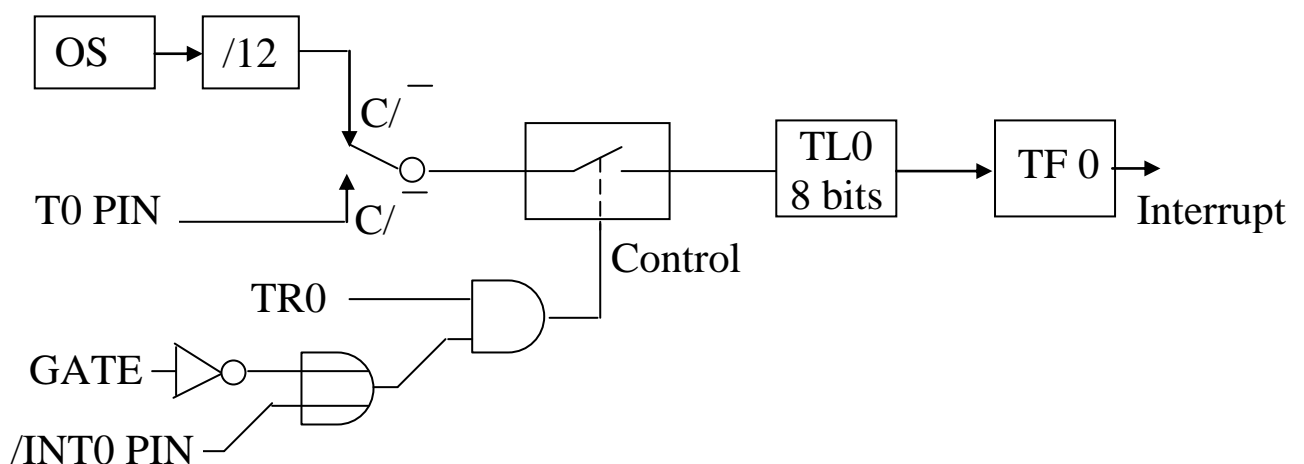
+ **Chế độ 1:** hoạt động tương tự như chế độ 0, chỉ khác là thanh ghi Timer/Counter được sử dụng cả 16 bit. Xung nhịp được dùng kết hợp với các thanh ghi thời gian byte thấp và byte cao (TH1 và TL1). Khi xung Clock được nhận, bộ Timer sẽ đếm tăng lên: 0000h, 0001h, 0002,... Khi hiện tượng tràn xảy ra, 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).

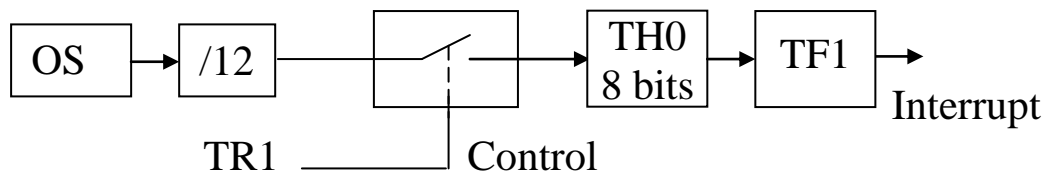


+ **Chế độ 2**: Chế độ này của thanh ghi Timer cũng hoạt động tương tự như 2 chế độ trên, nhưng nó được tổ chức như bộ đếm 8 bit (TL1) với chế độ tự động nạp lại, như hình 2.6. Khi xảy ra hiện tượng tràn ở TL1, không chỉ thiết lập bit TF1 mà còn tự động nạp lại cho TL1 bằng nội dung của TH1, đã được 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 chuyển mạch ngoài. Ở chế độ này, Timer 1 có thể được sử dụng bởi cổng 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:

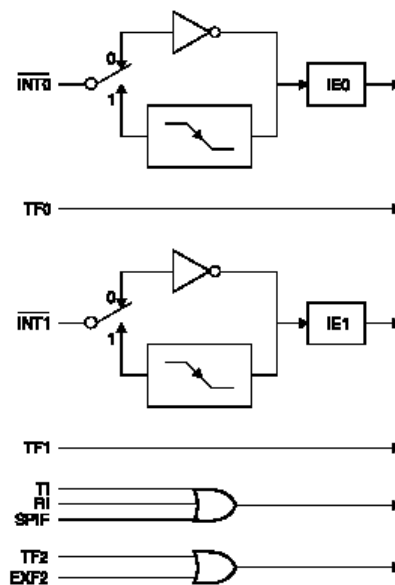
2.1.1.5.1. Phân loại ngắt trong On-chip:

Bộ 8051 có tất cả 5 Vectors ngắt bao gồm: 2 ngắt ngoài (/INT0 và /INT1), 2 ngắt của khối thời gian (Timer 0, 1), và ngắt cổng 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.

Hình 2.17. Các nguồn ngắt của 8051

Figure 10. Interrupt Sources



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 8051:

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:

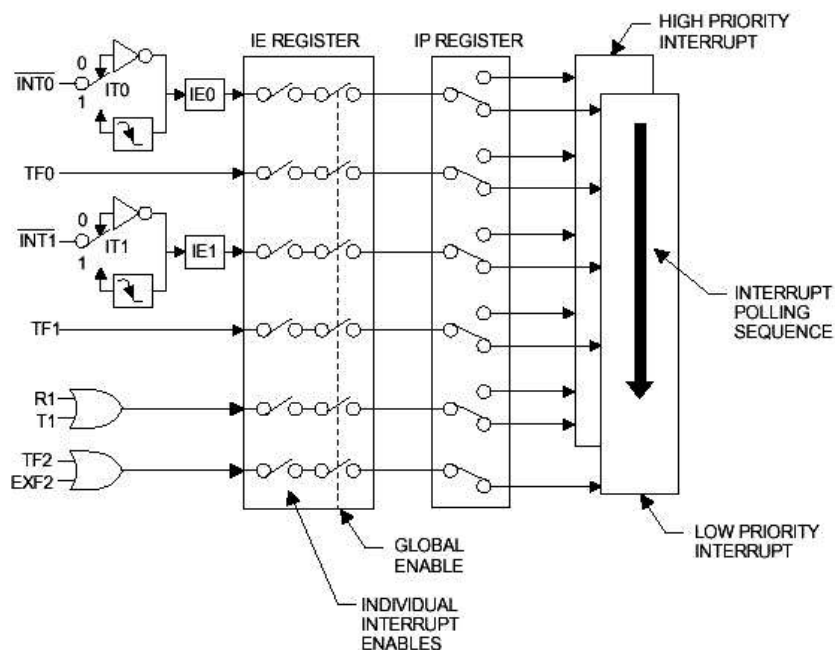
a- Đồng thời có ngắt với mức ưu tiên cao hơn hoặc bằng ngắt đang phục vụ. (Một ngắt có mức ưu tiên bằng hoặc cao hơn đang sẵn sàng để được phục vụ)

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 8051

Figure 20. AT89 Interrupt Control System



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:

Bảng 2.7. Địa chỉ véc tơ ngắt

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 (8051)	TF2 hoặc EXF2	002Bh
System Reset	RST	0000h

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.

a) 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 xóa 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 hủy yêu cầu đó trước khi thủ tục phục vụ ngắt hoàn thành, nếu không ngắt khác sẽ được tạo ra.

b) Vận hành Single-Step:

Cấu trúc ngắt 8051 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 8051:

Chế độ	LB1	LB2	LB	Loại bảo vệ
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. Tập lệnh của họ VĐK 8051

Bộ VĐK có tập lệnh được tối ưu hoá để ứng dụng trong các hệ thống điều khiển, đo lường 8 bit. Để tăng khả năng truy xuất RAM nội trên các dữ liệu nhỏ, các kiểu định địa chỉ đặc biệt đã được áp dụng. Ngoài ra tập lệnh của VĐK còn hỗ trợ các biến 1 bit, cho phép quản lý bit trực tiếp trong các hệ logic và điều khiển bit có yêu cầu xử lý bit. Do họ VĐK 8051 có các mã lệnh

8 bit, nên số lệnh có thể lên đến 256 lệnh (thực tế có 255lệnh, con 1 lệnh chưa được định nghĩa). Trong đó có 139 lệnh 1 byte, 92lệnh 2 byte, 24 lệnh 3 byte. Mỗi lệnh đều đặc trưng bởi mã lệnh (mã máy), mã gọi nhớ, số byte của lệnh và số chu kỳ máy cần để thực thi lệnh. Các lệnh của 8051 được chia thành 5 nhóm lệnh:

- Nhóm lệnh di chuyển dữ liệu
- Nhóm lệnh số học
- Nhóm lệnh logic
- Nhóm lệnh rẽ nhánh chương trình
- Nhóm lệnh điều khiển biến logic

Các quy ước trong câu lệnh và địa chỉ:

- Rn: Thanh ghi R0-R7 của bảng thanh ghi hiện hành đang được chọn để định địa chỉ thanh ghi

- Direct: Địa chỉ 8 bit của ô nhớ dữ liệu nội trú, nó có thể là ô nhớ trong RAM nội hoặc SFR (00h-FFh)

- @Ri: Ô nhớ 8 bit của RAM nội được định địa chỉ gián tiếp thông qua thanh ghi R0 hoặc R1.

- Source (Src): Toán hạng nguồn, có thể là Rn hoặc direct hoặc @Ri.

- Dest: Toán hạng đích, có thể là Rn hoặc direct hoặc @Ri.

- #Data: Hằng số 8bit chứa trong lệnh.

- #Data 16: Hằng số 16 bit chứa trong lệnh.

- Bit: Bit được định địa chỉ trực tiếp trong RAM nội trú hoặc SFR.

- Rel: Offset 8 bit có dấu (từ -128 đến +127). Nó được lệnh SJMP và các lệnh nhảy có điều kiện sử dụng.

- Addr11: Địa chỉ 11 bit của bộ nhớ chương trình, được lệnh ACALL và AMP sử dụng.

- Addr16: Địa chỉ 16 bit của 64Kb bộ nhớ chương trình, được lệnh LCALL và LJMP sử dụng.

Các ký hiệu dùng trong mô tả lệnh:

Ký hiệu	Ý nghĩa
<-	Được thay thế bởi...
()	Nội dung của...
(())	Dữ liệu được trở bởi...
Rrr	1 trong 8 thanh ghi (R0-R7) của các băng thanh ghi
dddddddd	Các bit dữ liệu
Aaaaaaaaa	Các bit địa chỉ
bbbbbbbb	Địa chỉ của 1 bit
I	Định địa chỉ gián tiếp
Eeeeeeee	Địa chỉ tương đối 8 bit

2.1.1.7.1. Nhóm lệnh di chuyển dữ liệu

a) Lệnh MOV dạng Byte:

Cú pháp câu lệnh: **MOV <dest-byte>, <src-byte>**

Chức năng: Sao chép nội dung của toán hạng nguồn vào toán hạng đích, nội dung của toán hạng nguồn không thay đổi. Lệnh này không làm ảnh hưởng tới các cờ và các thanh ghi khác.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
MOV A, Rn	1	1	11101rrr	(A)<-(Rn)
MOV A, direct	2	1	11100101 aaaaaaaaa	(A)<-(direct)
MOV A, @Ri	1	1	1110111i	(A)<-((Ri))
MOV A, #data	2	1	01110100 dddddddd	(A)<-#data
MOV Rn, A	1	1	11111rrr	(Rn)<-(A)
MOV Rn, direct	2	2	10101rrr aaaaaaaaa	(Rn)<-(direct)
MOV Rn, #data	2	1	01111rrr dddddddd	(Rn)<-#data
MOV direct, A	2	1	11110101 aaaaaaaaa	(direct)<-(A)
MOV direct, Rn	2	2	10001rrr aaaaaaaa	(direct)<-(Rn)
MOV direct, direct	3	2	10000101 aaaaaaaaa aaaaaaaa	(direct)<- (direct)
MOV direct, @Ri	2	2	1000011i aaaaaaaa	(direct)<-((Ri))
MOV direct, #data	3	2	01110101 aaaaaaaaa dddddddd	(direct)<-#data
MOV @Ri, A	1	1	1111011i	((Ri))<-(A)
MOV @Ri, direct	2	2	1010011i	((Ri))<-(direct)
MOV @Ri, #data	2	1	0111011i dddddddd	((Ri))<-#data

b) Lệnh MOV dạng Bit:

Cú pháp câu lệnh: **MOV <dest-bit>, <scr-bit>**

Chức năng: Chuyển bit dữ liệu ở dạng sao chép toán hạng nguồn vào toán hạng đích. Một trong 2 toán hạng phải là cờ nhớ (C), toán hạng còn lại sẽ

là bit bất kỳ được định địa chỉ trực tiếp. Lệnh không làm ảnh hưởng tới các thanh ghi khác hoặc các cờ khác.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
MOV C, bit	2	1	10100010 bbbbbbbb	(C)<-(bit)
MOV bit, C	2	2	10010010 bbbbbbbb	(bit)<-(C)

c) Lệnh MOV dạng Word:

Cú pháp câu lệnh: **MOV DPTR, #data16**

Chức năng: Giá trị 16 bit ở toán hạng thứ 2 trực tiếp trong câu lệnh được nạp vào thanh ghi DPTR. Hằng số 16 bit này được đặt ở byte 2 và byte 3 của lệnh. Byte 2 là byte cao được nạp cho thanh ghi DPH, byte 3 là byte thấp được nạp vào thanh ghi DPL. Lệnh này không làm ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
MOV DPTR, #data16	3	2	10010000 dddddddd ddddddd	(C)<-(bit)

d) Lệnh chuyển byte mã lệnh:

Cú pháp câu lệnh: **MOVC A, @A + <thanh ghi c⁻¹ s^e>**

Chức năng: Nạp cho thanh ghi tích lũy byte mã lệnh từ bộ nhớ chương trình. Địa chỉ của byte được nạp trong bộ nhớ là tổng nội dung của thanh ghi A 8 bit với nội dung của thanh ghi cơ sở 16 bit (có thể là DPTR hoặc PC – thanh ghi đếm chương trình). Trong trường hợp sau, PC được tăng để trở đến địa chỉ của lệnh tiếp theo ((PC)<-(PC+1)) trước khi được cộng với nội dung của thanh ghi A, còn thanh ghi DPTR không bị thay đổi. Lệnh không làm ảnh hưởng tới các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
MOVC A,@A+DPTR	1	2	10010011	(A)<-((A)+(DPTR))
MOVC A,@A+PC	1	2	10000011	(A)<-((A)+(PC))

e) Lệnh chuyển dữ liệu ra ngoài:

Cú pháp câu lệnh: **MOVX** <dest-byte>, <src-byte>

Chức năng: Chuyển dữ liệu giữa thanh ghi tích lũy với bộ nhớ ngoài.

Các lệnh này chia làm 2 loại, một loại cung cấp địa chỉ 8 bit và 1 loại cung cấp địa chỉ 16 bit.

Nếu dữ liệu được chuyển là 8 bit, nội dung của R0 hoặc R1 trong bảng thanh ghi hiện hành sẽ cung cấp địa chỉ 8 bit đa hợp với dữ liệu trên P0. 8 bit địa chỉ này đủ để mã hoá cho các cổng I/O mở rộng bên ngoài chip hoặc cho 1 dãy Ram kích thước tương đối nhỏ. Với các dãy RAM có kích thước lớn hơn một chút, một chân của cổng bất kỳ nào đó có thể được sử dụng để tạo ra các bit địa chỉ cao. Các chân này nên được điều khiển bởi 1 lệnh xuất đặt trước lệnh MOVX.

Nếu dữ liệu được chuyển là 16 bit, thì DPTR tạo ra địa chỉ 16 bit. P2 xuất 8 bit địa chỉ cao (nội dung của DPH), còn P0 xuất ra 8 bit địa chỉ thấp đa hợp với dữ liệu. Thanh ghi chức năng đặc biệt P2 duy trì nội dung trước đó trong khi các bộ đệm xuất của P2 đang phát các nội dung của DPH. Dạng này nhanh hơn và hiệu quả hơn khi truy xuất nhiều dãy dữ liệu rất lớn (lên đến 64 Kb) do ta không cần thêm để thiết lập các cổng khác.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
MOVX A, @Ri	1	2	11100011	(A)<-((Ri))
MOVX @Ri, A	1	2	11110011	((Ri))<(A)
MOVX A, @DPTR	1	2	11100000	(A)<-((DPTR))
MOVX @DPTR, A	1	2	11110000	((DPTR))<-(A)

f) Lệnh chuyển số liệu vào ngăn xếp:

Cú pháp câu lệnh: PUSH direct

Chức năng: Chuyển số liệu có trong câu lệnh vào ngăn xếp. Trước tiên, con trỏ r ngăn xếp (SP) được tăng lên 1, Sau đó số liệu sẽ được chuyển vào đỉnh của ngăn xếp mà địa chỉ đỉnh này được trỏ bởi SP. Ngăn xếp nằm ở RAM nội trú

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
PUSH direct	2	2	11000000 aaaaaaaaa	(SP)<-(SP+1) ((SP))<-(direct)

g) Lệnh chuyển số liệu ra khỏi ngăn xếp:

Cú pháp câu lệnh: POP direct

Chức năng: Chuyển nội dung của ngăn xếp ở RAM trong, có địa chỉ được SP trỏ tới đến nơi có địa chỉ trỏ tiếp trong câu lệnh. Sau đó, con trỏ ngăn xếp (SP) được giảm đi 1. Lệnh không ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
POP direct	2	2	11010000 aaaaaaaa	(direct)<--((SP)) (SP)<--(SP-1)

h) Hoán chuyển dữ liệu:

Cú pháp câu lệnh: XCH A, <byte>

Chức năng: Hoán chuyển nội dung giữa thanh ghi A với thanh ghi hoặc bộ nhớ có địa chỉ chứa trong toán hạng thứ 2 của câu lệnh. Toán hạng thứ 2 có thể được định địa chỉ kiểu thanh ghi, thanh ghi trực tiếp hoặc thanh ghi gián tiếp.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
XCH A, Rn	1	1	11001rrr	(A)<-->(Rn)
XCH A, direct	2	1	11000101 aaaaaaaa	(A) <-->(direct)
XCH A, @Ri	1	1	1100011i	(A) <-->((Ri))

i) Hoán chuyển 4 bit thấp:

Cú pháp câu lệnh: XCHD A,@Ri

Chức năng: Hoán chuyển 4 bit thấp nội dung trong thanh ghi A với ô nhớ của RAM bên trong, có địa chỉ được định gián tiếp qua thanh ghi được chỉ ra trong lệnh. Lệnh này không ảnh hưởng tới trạng thái các cờ và nửa cao của các thanh ghi trong lệnh

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
XCHD A, @Ri	1	1	1101011i	(A3-A0)<-->((Ri3-Ri0))

2.1.1.7.2. Nhóm lệnh tính toán số học

a) Lệnh thực hiện phép cộng.

Cú pháp câu lệnh: **ADD** **A, <scr-byte>**

Chức năng: Cộng giá trị 1 byte ở địa chỉ được chỉ ra ở câu lệnh với nội dung trong thanh ghi tích lũy, kết quả được lưu vào thanh ghi tích lũy. Nếu có nhớ từ bit số 7 hoặc bit số 3 thì cờ nhớ hoặc cờ nhớ phụ được thiết lập, Ngược lại các cờ nêu trên được xoá. Khi cộng 2 số nguyên không dấu mà bị tràn thì cờ nhớ cũng được thiết lập để cho ta biết phép toán bị tràn. Trường hợp thực hiện lệnh ADD mà có nhớ từ bit số 6 nhưng không có nhớ từ bit số 7, hoặc có nhớ từ bit số 7 nhưng không có nhớ từ bit số 6 thì cờ tràn sẽ được thiết lập, ngược lại thì OV bị xoá. Khi cộng 2 số nguyên có dấu mà tổng là 1 số âm thì OV được thiết lập

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ADD A, Rn	1	1	00101rrr	$(A) \leftarrow (A) + (Rn)$
ADD A, direct	2	1	00100101 aaaaaaaaa	$(A) \leftarrow (A) + (\text{direct})$
ADD A, @Ri	1	1	0010011i	$(A) \leftarrow (A) + ((Ri))$
ADD A, #data	2	1	00100100 dddddddd	$(A) \leftarrow (A) + \#data$

b) Lệnh cộng có nhớ

Cú pháp câu lệnh: **ADDC** **A, <scr-byte>**

Chức năng: Cộng đồng thời nội dung của 1 byte ở địa chỉ được chỉ ra trong câu lệnh với nội dung chứa trong thanh ghi tích lũy và cờ nhớ. Nếu có nhớ từ bit số 7 hoặc bit số 3 thì cờ nhớ hoặc cờ nhớ phụ được thiết lập bằng 1, ngược lại các cờ nêu trên bị xoá. Khi cộng các số nguyên không dấu mà bị tràn thì cờ nhớ cũng được thiết lập. Trường hợp thực hiện lệnh ADC mà có nhớ từ bit số 6 nhưng không nhớ từ bit số 7, hoặc có nhớ từ bit số 7 nhưng không

nhớ từ bit số 6 thì cờ tràn sẽ được thiết lập, ngược lại cờ này bị xoá. Khi cộng các số nguyên có dấu mà tổng là 1 số âm thì OV được thiết lập.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ADDC A, Rn	1	1	00110rrr	$(A) \leftarrow (A) + (C) + (Rn)$
ADDC A, direct	2	1	00110101 aaaaaaaa	$(A) \leftarrow (A) + (C) + (\text{direct})$
ADDC A, @Ri	1	1	0011011i	$(A) \leftarrow (A) + (C) + ((Ri))$
ADDC A, #data	2	1	00110100 dddddddd	$(A) \leftarrow (A) + (C) + \#data$

c) Lệnh Trừ có mượn.

Cú pháp câu lệnh: **SUBB** **A, <scr-byte>**

Chức năng: Trừ thanh ghi tích lũy cho toán hạng thứ 2 và cờ nhớ, kết quả được lưu vào thanh ghi tích lũy. Cờ nhớ được đặt bằng 1 nếu có số mượn được cần đến cho bit số 7, ngược lại thì cờ nhớ bị xoá. Cờ nhớ phụ được thiết lập nếu có nhớ cho bit số 3. Trường hợp thực hiện lệnh SUBB mà có số mượn được cần đến cho bit 7 (không phải cho bit 6), hoặc cho bit 6 (không phải cho bit 7) thì cờ tràn sẽ được thiết lập, ngược lại thì OV bị xoá. Khi trừ các số nguyên có dấu mà kết quả là 1 số âm thì OV được thiết lập.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
SUBB A, Rn	1	1	10011rrr	$(A) \leftarrow (A) - (C) - (Rn)$
SUBB A, direct	2	1	10010101 aaaaaaaa	$(A) \leftarrow (A) - (C) - (\text{direct})$
SUBB A, @Ri	1	1	1001011i	$(A) \leftarrow (A) - (C) - ((Ri))$
SUBB A, #data	2	1	10010100 dddddddd	$(A) \leftarrow (A) - (C) - \#data$

d) **Lệnh tăng lên 1 đơn vị**

Cú pháp câu lệnh: **INC** <byte>

Chức năng: Tăng giá trị của byte trong câu lệnh lên 1 đơn vị. Nếu giá trị ban đầu của byte là 0FFh, thì sau khi thực hiện lệnh INC nội dung của byte sẽ là 00h. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
INC A	1	1	00000100	(A)<- (A) + 1
INC Rn	1	1	00001rrr	(Rn)<- (Rn) + 1
INC direct	2	1	00000101 aaaaaaaa	(direct)<- (direct) + 1
INC @Ri	1	1	0000011i	((Ri))<- ((Ri)) + 1

e) **Lệnh giảm 1 đơn vị.**

Cú pháp câu lệnh: **DEC** <byte>

Chức năng: Giảm giá trị của byte trong câu lệnh xuống 1 đơn vị. Nếu giá trị ban đầu của byte là 00h, thì sau khi thực hiện lệnh DEC nội dung của byte sẽ là 0FFh. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
DEC A	1	1	00010100	(A)<- (A) - 1
DEC Rn	1	1	00011rrr	(Rn)<- (Rn) - 1
DEC direct	2	1	00010101 aaaaaaaa	(direct)<- (direct) - 1
DEC @Ri	1	1	0001011i	((Ri))<- ((Ri)) - 1

f) Lệnh tăng con trỏ đơn vị.

Cú pháp câu lệnh: INC DPTR

Chức năng: Tăng con trỏ dữ liệu lên 1 đơn vị. Khi byte thấp của con trỏ dữ liệu bị tràn, thì byte cao của con trỏ dữ liệu tăng lên 1 đơn vị. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
INC DPTR	1	2	10100011	(DPTR)<- (DPTR) + 1

g) Lệnh thực hiện phép nhân.

Cú pháp câu lệnh: MUL AB

Chức năng: Nhân các số nguyên không dấu 8 bit trong thanh ghi tích lũy với thanh ghi B. Byte thấp của kết quả 16 bit được lưu trong thanh ghi tích lũy, còn byte cao được lưu trong thanh ghi B. Nếu kết quả lớn hơn 0FFh thì cờ tràn được thiết lập, cờ nhớ luôn bị xoá.

Câu lệnh	Số byte	Số chu kỳ	Mã lệnh	Hoạt động
MUL AB	1	4	10100100	(B)<- byte cao của (A)x(B) (A)<- byte thấp của (A)x(B)

h) Lệnh thực hiện phép chia.

Cú pháp câu lệnh: DIV AB

Chức năng: Chia số nguyên không dấu 8 bit trong thanh ghi tích lũy cho số nguyên không dấu 8 bit trong thanh ghi B. Thương số được lưu trong thanh ghi tích lũy, còn số dư được lưu trong thanh ghi B. Cờ tràn và cờ nhớ bị xoá.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
DIV AB	1	4	10000100	(A)<- thương của (A)/(B) (B)<- số dư của (A)/(B)

i) Hiệu chỉnh số thập phân

Cú pháp câu lệnh: **DA A**

Chức năng: Hiệu chỉnh số thập phân nội dung 8 bit trong thanh ghi A sau khi thực hiện phép cộng.

Nếu 4 bit thấp trong thanh ghi A có giá trị lớn hơn 9 hoặc cờ nhớ phụ được thiết lập thì phải cộng thêm 6 vào thanh ghi A để cho chữ số thập phân được chính xác. Phép cộng này sẽ đặt cờ nhớ nếu số nhớ từ 4 bit thấp chuyển đến tất cả 4 bit cao, ngược lại phép toán không xoá cờ nhớ.

Nếu 4 bit cao trong thanh ghi A có giá trị lớn hơn 9 hoặc cờ nhớ (CF) được thiết lập, thì cũng phải cộng thêm 6 vào thanh ghi A.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh
DA A	1	1	11010100

Hoạt động:

- Nếu $[(A3-A0)>9]$ hoặc $[(AC)=1]$ thì $(A3-A0) \leftarrow (A3-A0) + 6$

- Nếu $[(A7-A4)>9]$ hoặc $[(C)=1]$ thì $(A7-A4) \leftarrow (A7-A4) + 6$

2.1.1.7.3. Nhóm lệnh tính toán logic

a) Lệnh AND cho các biến 1 byte.

Cú pháp câu lệnh: **ANL <dest-byte>, <src-byte>**

Chức năng: Thực hiện phép toán logic AND theo mức bit giữa các biến dài 1 byte đã cho, kết quả được lưu vào toán hạng đích. Toán hạng nguồn cho phép 6 chế độ địa chỉ hoá. Khi toán hạng đích là thanh ghi tích lũy thì toán hạng nguồn có thể là thanh ghi trực tiếp, thanh ghi gián tiếp hoặc tức thời. Khi toán hạng đích là địa chỉ trực tiếp thì toán hạng nguồn có thể là thanh ghi tích lũy hoặc dữ liệu tức thời. Lệnh này không làm ảnh hưởng tới các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ANL A, Rn	1	1	01011rrr	(A)<-(A) AND (Rn)
ANL A, direct	2	1	01010101 aaaaaaaa	(A)<-(A) AND (dir.)
ANL A, @Ri	1	1	0101011i	(A)<- (A) AND ((Ri))
ANL A, #data	2	1	01010100 dddddddd	(A)<- (A) AND #data
ANL direct, A	2	1	01010010 aaaaaaaa	(dir.)<-(dir.)AND (A)
ANL direct, #data	3	2	01010011 aaaaaaaa dddddddd	(dir.)<(dir.)AND#data

b) Lệnh AND cho các biến 1 bit

Cú pháp câu lệnh: ANL C, <src-bit>

Chức năng: Thực hiện phép tính logic AND cho các biến mức bit. Nếu giá trị logic của toán hạng nguồn bằng 0, thì cờ nhớ bị xoá. Dấu “/” đứng trước 1 toán hạng cho biết bit nguồn được lấy bù trước khi thực hiện AND với cờ nhớ nhưng giá trị của bit nguồn không bị thay đổi bởi thao tác lấy bù. Lệnh này không làm ảnh hưởng tới trạng thái các cờ khác. Toán hạng nguồn chỉ được sử dụng kiểu định địa chỉ trực tiếp.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ANL C, bit	2	2	10000010 bbbbbbbbb	(C)<-(C) AND (bit)
ANL C, /bit	2	2	10110000 bbbbbbbbb	(C)<-(C) AND NOT (bit)

c) Lệnh OR cho các biến 1 byte

Cú pháp câu lệnh: **ORL** <dest-byte>, <src-byte>

Chức năng: Thực hiện phép toán logic OR theo mức bit giữa các biến dài 1 byte đã cho, kết quả được lưu vào toán hạng đích. Toán hạng nguồn cho phép 6 chế độ địa chỉ hoá. Khi toán hạng đích là thanh ghi tích lũy thì toán hạng nguồn có thể là thanh ghi trực tiếp, thanh ghi gián tiếp hoặc tức thời. Khi toán hạng đích là địa chỉ trực tiếp thì toán hạng nguồn có thể là thanh ghi tích lũy hoặc dữ liệu tức thời. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ORL A, Rn	1	1	01001rrr	(A)<-(A) OR (Rn)
ORL A, direct	2	1	01000101 aaaaaaaaa	(A)<-(A) OR (dir.)
ORL A, @Ri	1	1	0100011i	(A)<- (A) OR ((Ri))
ORL A, #data	2	1	01000100 dddddddd	(A)<- (A) OR #data
ORL direct, A	2	1	01000010 aaaaaaaaa	(dir.)<-(dir.) OR (A)
ORL direct, #data	3	2	01000011 aaaaaaaaa ddddddd	(dir.)<(dir.) OR #data

d) Lệnh OR cho các biến 1 bit

Cú pháp câu lệnh: **ORL** C, <src-bit>

Chức năng:

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ORL C, bit	2	2	01110010 bbbbbbbbb	(C)<-(C) OR (bit)
ORL C, /bit	2	2	10100000 bbbbbbbbb	(C)<-(C) OR NOT (bit)

) Lệnh X-OR cho các biến 1 byte

Cú pháp câu lệnh: XRL <dest-byte>, <src-byte>

Chức năng: Thực hiện phép toán logic X-OR theo mức bit giữa các biến dài 1 byte đã cho, kết quả được lưu vào toán hạng đích. Toán hạng nguồn cho phép 6 chế độ địa chỉ hoá. Khi toán hạng đích là thanh ghi tích lũy thì toán hạng nguồn có thể là thanh ghi trực tiếp, thanh ghi gián tiếp hoặc tức thời. Khi toán hạng đích là địa chỉ trực tiếp thì toán hạng nguồn có thể là thanh ghi tích lũy hoặc dữ liệu tức thời. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
XRL A, Rn	1	1	01101rrr	(A)<-(A) XOR (Rn)
XRL A, direct	2	1	01100101 aaaaaaaa	(A)<-(A) XOR (dir.)
XRL A, @Ri	1	1	0110011i	(A)<- (A) XOR ((Ri))
XRL A, #data	2	1	01100100 dddddddd	(A)<- (A) XOR #data
XRL direct, A	2	1	01100010 aaaaaaaa	(dir.)<-(dir.)XOR (A)
XRL direct, #data	3	2	01100011 aaaaaaaa dddddddd	(dir.)<(dir.) XOR #data

f) Lệnh dịch trái thanh ghi A

Cú pháp câu lệnh: RL A

Chức năng: 8 bit trong thanh ghi A được dịch trái 1 bit. Bit 7 được quay đến vị trí của bit 0. Các cờ không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RL A	1	1	00100011	$(A_{n+1}) \leftarrow (A_n)$, với $n = 0 \dots 6$ $(A_0) \leftarrow (A_7)$

g) Lệnh dịch trái thanh ghi A cùng với cờ nhớ

Cú pháp câu lệnh: RLC A

Chức năng: 8 bit trong thanh ghi A và cờ nhớ cùng được dịch trái 1 bit. Bit 7 được chuyển đến cờ nhớ và trạng thái ban đầu của cờ nhớ được đưa về bit 0. Các cờ khác không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RLC A	1	1	00110011	$(A_{n+1}) \leftarrow (A_n)$, với $n = 0 \dots 6$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$

h) Lệnh dịch phải thanh ghi A

Cú pháp câu lệnh: RR A

Chức năng: 8 bit trong thanh ghi A được dịch phải sang 1 bit. Bit 0 được quay đến vị trí của bit 7. Các cờ không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RR A	1	1	00000011	$(A_n) \leftarrow (A_{n+1})$, với $n = 0 \dots 6$ $(A_7) \leftarrow (A_0)$

i) Lệnh dịch phải thanh ghi A cùng với cờ nhớ.

Cú pháp câu lệnh: RRC A

Chức năng: 8 bit trong thanh ghi A và cờ nhớ cùng được dịch phải 1 bit. Bit 0 được chuyển đến cờ nhớ và trạng thái ban đầu của cờ nhớ được đưa về bit 7. Các cờ khác không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RRC A	1	1	00010011	(An) <- (An+1), với n = 0...6 (A7) <- (C) (C) <- (A0)

j) Lệnh trao đổi nội dung hai nửa byte của A

Cú pháp câu lệnh: SWAP A

Chức năng: Trao đổi nội dung 2 nửa thấp và cao (mỗi nửa 4 bit) của thanh ghi A (các bit từ 0 đến 3 và các bit từ 4 đến 7). Thao tác này còn được hiểu là quay thanh ghi A 4 bit. Các cờ không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
SWAP A	1	1	11000100	(A3-A0) <- (A7-A4)

2.1.1.7.4. Nhóm lệnh rẽ nhánh chương trình

a) Lệnh gọi tuyệt đối

Cú pháp câu lệnh: ACALL addr11

Chức năng: Gọi không điều kiện một chương trình con đặt tại địa chỉ được chỉ ra trong câu lệnh. Lệnh này tăng bộ đếm chương trình thêm 2 đơn vị để PC chứa địa chỉ của lệnh kế lệnh ACALL, sau đó cất nội dung 16 bit của PC vào ngăn xếp (byte thấp cất trước) và tăng con trỏ ngăn xếp lên 2 đơn vị. Địa chỉ đích sẽ được hình thành bằng cách ghép 5 bit cao của thanh ghi PC (sau khi được tăng), 3 bit cao của byte mã lệnh và byte thứ 2 của lệnh. Do đó chương trình con được gọi phải nằm trong đoạn 2 Kbyte của bộ nhớ chương trình. Lệnh không làm ảnh hưởng tới các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
ACALL addr11	2	2	aaa10001 aaaaaaaa	(PC) <- (PC) + 2 (SP) <- (SP) + 1 ((SP)) <- (PC7-PC0) (SP) <- (SP) + 1 ((SP)) <- (PC15-PC8) (PC10-PC0) <- (page address)

b) Lệnh gọi dài.

Cú pháp câu lệnh: LCALL addr16

Chức năng: Gọi một chương trình con đặt tại địa chỉ được chỉ ra trong câu lệnh. Lệnh này tăng bộ đếm chương trình thêm 3 đơn vị để PC chứa địa chỉ của lệnh kế lệnh LCALL, sau đó cất nội dung 16 bit của PC vào ngăn xếp(byte thấp cất trước) và tăng con trỏ ngăn xếp lên 2 đơn vị. Tiếp theo nó sẽ chuyển byte thứ 2 và byte thứ 3 trong câu lệnh LCALL vào byte cao và byte thấp của PC. Việc thực thi chương trình tiếp tục với lệnh ở địa chỉ này. Như vậy chương trình con có thể bắt đầu bằng bất cứ nơi nào trong không gian bộ nhớ chương trình 64Kbyte. Lệnh không làm ảnh hưởng tới các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
LCALL addr16	3	2	00010010 aaaaaaaaa aaaaaaaaa	(PC) <- (PC) + 3 (SP) <- (SP) + 1 ((SP)) <- (PC7-PC0) (SP) <- (SP) + 1 ((SP)) <- (PC15-PC8) (PC) <- addr15-addr0

c) Lệnh quay trở lại từ chương trình con.

Cú pháp câu lệnh: RET

Chức năng: Trở về từ chương trình con. Lệnh này được thực hiện sau khi thực hiện xong lệnh ACALL hoặc LCALL. RET lấy lại byte cao và byte thấp của PC từ ngăn xếp, giảm SP đi 2 đơn vị. Chương trình tiếp tục được thực hiện với lệnh có địa chỉ ở trong PC. Các cờ không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RET	1	2	00100010	(PC15-PC8) <- ((SP)) (SP) <- (SP) - 1 (PC7-PC0) <- ((SP)) (SP) <- (SP) - 1

d) Lệnh quay trở lại từ ngắt.

Cú pháp câu lệnh: RETI

Chức năng: Trở về từ chương trình con. RETI lấy lại byte cao và byte thấp của PC từ ngăn xếp, phục hồi logic ngắt để có thể nhận các ngắt khác có cùng mức ưu tiên với ngắt được xử lý, sau đó giảm SP đi 2 đơn vị. Chương trình tiếp tục được thực hiện với lệnh trước khi xử lý ngắt với địa chỉ ở trong PC. Các cờ nhớ không bị ảnh hưởng.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
RETI	1	2	00110010	(PC15-PC8) <- ((SP)) (SP) <- (SP) - 1 (PC7-PC0) <- ((SP)) (SP) <- (SP) - 1

e) Lệnh nhảy gián tiếp

Cú pháp câu lệnh: JMP @A+DPTR

Chức năng: Cộng giá trị không dấu 8 bit của thanh ghi A với con trỏ dữ liệu 16 bit và nạp kết quả vào bộ đếm chương trình, kết quả này chính là địa chỉ để nạp lệnh kế tiếp. Việc cộng 16 bit được thực hiện: Số nhớ từ 8 bit thấp được truyền đến tất cả các bit cao. Cả 2 thanh ghi A và DPTR đều không bị thay đổi. Lệnh này không ảnh hưởng tới trạng thái các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JMP @A+DPTR	1	2	01110011	(PC)<-(A)+(DPTR)

f) Lệnh nhảy nếu 1 bit được thiết lập

Cú pháp câu lệnh: **JB** **bit, rel**

Chức năng: Nếu bit đã cho có giá trị bằng 1 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (tương đối) trong byte thứ 3 của lệnh với nội dung trong PC (sau khi được tăng đến địa chỉ của byte đầu tiên của lệnh kế tiếp). Bit được kiểm tra không bị thay đổi, lệnh không tới các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JB bit, rel	3	2	00100000 bbbbbbbbb eeeeeeee	(PC)<-(PC)+3 Nếu (bit)=1 thì: (PC)<-(PC) + rel

g) Lệnh nhảy nếu 1 bit không được thiết lập

Cú pháp câu lệnh: **JNB bit, rel**

Chức năng: Nếu bit đã cho có giá trị bằng 0 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (tương đối) trong byte thứ 3 của lệnh với nội dung trong PC (sau khi được tăng đến địa chỉ của byte đầu tiên của lệnh kế tiếp). Bit được kiểm tra không bị thay đổi, lệnh không ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JNB bit, rel	3	2	00110000 bbbbbbbbb eeeeeee	(PC)<-(PC)+3 Nếu (bit)=0 thì: (PC)<-(PC) + rel

h) Lệnh nhảy nếu 1 bit được thiết lập và xoá bit đó.

Cú pháp câu lệnh: **JBC bit, rel**

Chức năng: Nếu bit đã cho có giá trị bằng 0 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh và xoá bit này, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (trương đối) trong byte thứ 3 của lệnh với nội dung trong PC (sau khi được tăng đến địa chỉ của byte đầu tiên của lệnh kế tiếp). Lệnh không ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JBC bit, rel	3	2	00010000 bbbbbbbb eeeeeee	$(PC) \leftarrow (PC) + 3$ Nếu (bit)=1 th×: (bit) ← 0 $(PC) \leftarrow (PC) + rel$

i) Lệnh nhảy nếu cờ nhớ được thiết lập.

Cú pháp câu lệnh: **JC rel**

Chức năng: Nếu cờ CF có giá trị bằng 1 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh và xoá bit này, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (trương đối) trong byte thứ 2 của lệnh với nội dung trong PC (sau khi được tăng bởi 2). Lệnh không ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JC rel	2	2	01000000 eeeeeee	$(PC) \leftarrow (PC) + 2$ Nếu (C)=1 th×: $(PC) \leftarrow (PC) + rel$

j) Lệnh nhảy nếu cờ nhớ không được thiết lập

Cú pháp câu lệnh: **JNC rel**

Chức năng: Nếu cờ CF có giá trị bằng 0 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh và xoá bit nay, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (tương đối) trong byte thứ 2 của lệnh với nội dung trong PC (sau khi được tăng bởi 2). Lệnh không ảnh hưởng tới các cờ

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JNC rel	2	2	01010000 eeeeeeee	$(PC) \leftarrow -(PC) + 2$ Nếu $(C) = 0$ thì: $(PC) \leftarrow -(PC) + rel$

k) Lệnh nhảy nếu thanh ghi A bằng 0

Cú pháp câu lệnh: **JZ rel**

Chức năng: Nếu tất cả các bit của thanh ghi A có giá trị bằng 0 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (tương đối) trong byte thứ 2 của lệnh với nội dung trong PC (sau khi được tăng bởi 2). Lệnh không ảnh hưởng tới các cờ. Nội dung thanh ghi A không bị thay đổi

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JZ rel	2	2	01100000 eeeeeeee	$(PC) \leftarrow -(PC) + 2$ Nếu $(A) = 0$ thì: $(PC) \leftarrow -(PC) + rel$

l) Lệnh nhảy nếu thanh ghi A khác 0.

Cú pháp câu lệnh: JNZ rel

Chức năng: Nếu có 1 hoặc nhiều bit của thanh ghi A có giá trị bằng 1 thì nó nhảy tới địa chỉ đã xác định trong câu lệnh, ngược lại nó sẽ tiếp tục thực hiện lệnh tiếp theo. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu (trương đối) trong byte thứ 2 của lệnh với nội dung trong PC (sau khi được tăng bởi 2). Lệnh không ảnh hưởng tới các cờ. Nội dung thanh ghi A không bị thay đổi

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
JNZ rel	2	2	01110000 eeeeeeee	(PC) \leftarrow -(PC)+2 Nếu (A) < > 0 th×: (PC) \leftarrow (PC) + rel

m) Lệnh nhảy khi so sánh 2 toán hạng

Cú pháp câu lệnh: CJNE <dest-byte>, <src-byte>, rel

Chức năng: So sánh giá trị của toán hạng đầu tiên, nếu 2 toán hạng không bằng nhau thì chương trình được rẽ nhánh. Địa chỉ đích rẽ nhánh được tính bằng cách cộng độ lệch tương đối (có dấu) trong byte sau cùng của lệnh với nội dung của PC (sau khi nội dung của PC được tăng đến địa chỉ bắt đầu của lệnh kế tiếp CJNE). Cờ nhớ (CF) sẽ được thiết lập nếu như giá trị nguyên không dấu của toán hạng đích nhỏ hơn giá trị nguyên không dấu của toán hạng nguồn, ngược lại thì cờ này bị xoá. Lệnh này không là thay đổi giá trị của các toán hạng

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
CJNE A, direct, rel	3	2	10110101 aaaaaaaa eeeeeeee	(PC)←-(PC)+3 Nếu (A) < > (dir.) thì: (PC)← (PC) + offset Nếu (A) < (dir.) thì: (C) ← 1, ng-íc l'i: (C) ← 0
CJNE A, #data, rel	3	2	10110100 dddddddd eeeeeeee	(PC)←-(PC)+3 Nếu (A) < > #data thì: (PC)← (PC) + offset Nếu (A) < #data thì: (C) ← 1, ng-íc l'i: (C) ← 0
CJNE Rn, #data, rel	3	2	10111rrr dddddddd eeeeeeee	(PC)←-(PC)+3 Nếu (Rn) < > #data thì: (PC)← (PC) + offset Nếu (Rn) < #data thì: (C) ← 1, ng-íc l'i: (C) ← 0
CJNE @Ri, #data, rel	3	2	1011011i dddddddd eeeeeeee	(PC)←-(PC)+3 Nếu ((Ri)) < > #data thì: (PC)← (PC) + offset Nếu ((Ri)) < #data thì: (C) ← 1, ng-íc l'i: (C) ← 0

n) Lệnh giảm và nhảy.

Cú pháp câu lệnh: **DJNZ** <byte>, <rel-address>

Chức năng: Giảm ô nhớ đi 1 và nhảy tới địa chỉ cho bởi toán hạng thứ 2 nếu như kết quả khác 0. Nếu kết quả ban đầu là 00h thì nó chuyển qua 0FFh. Địa chỉ đích được tính bằng cách cộng thêm độ lệch có dấu trong byte lệnh cuối cùng với nội dung của PC (sau khi tăng PC tới byte đầu tiên của lệnh tiếp theo). Ngăn nhớ được giảm giá trị có thể là 1 thanh ghi hoặc 1 byte địa chỉ trực tiếp. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
DJNZ Rn, rel	2	2	11011rrr eeeeeeee	(PC)<-(PC)+2 (Rn)<- (Rn) - 1 Nếu (Rn) < > 0 th×: (PC) <- (PC) + rel
DJNZ Direct, rel	3	2	11010101 aaaaaaaa eeeeeeee	(PC)<-(PC)+2 (dir.)<- (dir.) - 1 Nếu (dir.) < > 0 th×: (PC) <- (PC) + rel

o) Lệnh tạm ngừng hoạt động.

Cú pháp câu lệnh: **NOP**

Chức năng: Tạm ngừng hoạt động khi có lệnh này và chương trình sẽ tiếp tục được thực hiện ở lệnh tiếp theo. Lệnh này không làm ảnh hưởng tới trạng thái các thanh ghi và các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
NOP	1	1	00000000	(PC)<-(PC)+2

2.1.1.7.5. Nhóm lệnh điều khiển biến logic.

a) Lệnh xoá bit

Cú pháp câu lệnh: **CLR bit**

Chức năng: Xoá bit được chỉ ra trong câu lệnh về 0. Lệnh này có thể thao tác trên cờ nhớ, hoặc trên 1 bit bất kỳ được định địa chỉ trực tiếp. Lệnh không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
CLR C	1	1	11000011	(C) <- 0
CLR bit	2	1	11000010 bbbbbbbb	(bit) <- 0

b) Lệnh xóa thanh ghi tích lũy

Cú pháp câu lệnh: **CLR A**

Chức năng: Xoá tất cả các bit của thanh ghi tích lũy về 0. Các cờ không bị ảnh hưởng

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
CLR A	1	1	11100100	(A) <- 0

c) Lệnh thiết lập bit

Cú pháp câu lệnh: **SETB bit**

Chức năng: Thiết lập bit được chỉ ra trong câu lệnh lên mức logic 1. Lệnh này có thể thao tác trên cờ nhớ, hoặc trên 1 bit bất kỳ được định địa chỉ trực tiếp. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
SETB C	1	1	11010011	(C) <- 1
SETB bit	2	1	11010010 bbbbbbbb	(bit) <- 1

d) Lệnh lấy bù của bit

Cú pháp câu lệnh: CPL <bit>

Chức năng: Lấy bù bit được chỉ ra trong câu lệnh. Một bit có giá trị 1 được đổi thành 0 và ngược lại. Lệnh này có thể thao tác trên cờ nhớ, hoặc trên 1 bit bất kỳ được định địa chỉ trực tiếp. Lệnh không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
CPL C	1	1	10110011	(C) <- NOT (C)
CPL bit	2	1	10110010 bbbbbbbb	(bit) <- NOT (bit)

e) Lệnh lấy bù của thanh ghi tích lũy.

Cú pháp câu lệnh: CPL A

Chức năng: Lấy bù tất cả các bit của thanh ghi A. lệnh này không làm ảnh hưởng tới trạng thái các cờ.

Câu lệnh	Số Byte	Số chu kỳ	Mã lệnh	Hoạt động
CPL A	1	1	11110100	(A) <- NOT (A)

2.1.2. IC 74HC595

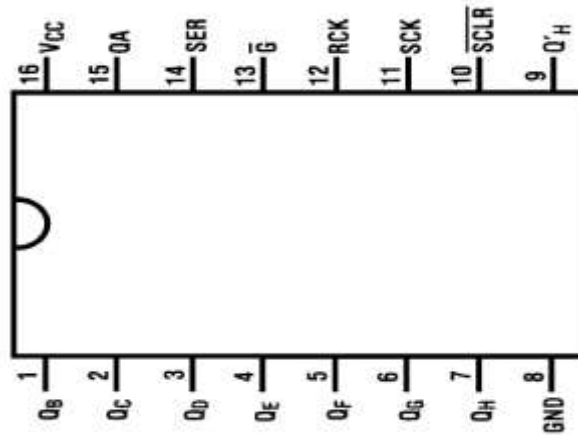
2.1.2.1. Mô tả chung

IC 74HC595 là IC ghi dịch 8bit có chốt đầu ra dùng công nghệ CMOS công silic tiên tiến. Thiết bị 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.

- Dòng vào thấp: 1uA max
- 8 bit đầu ra song song
- Phạm vi vận hành điện áp rộng
- Kết nối tầng nối tiếp

- Thanh ghi dịch có xóa trực tiếp
- Bảo đảm tần số chuyển dịch: DC tới 30MHz

2.1.2.2. Sơ đồ chân của IC 74HC595:

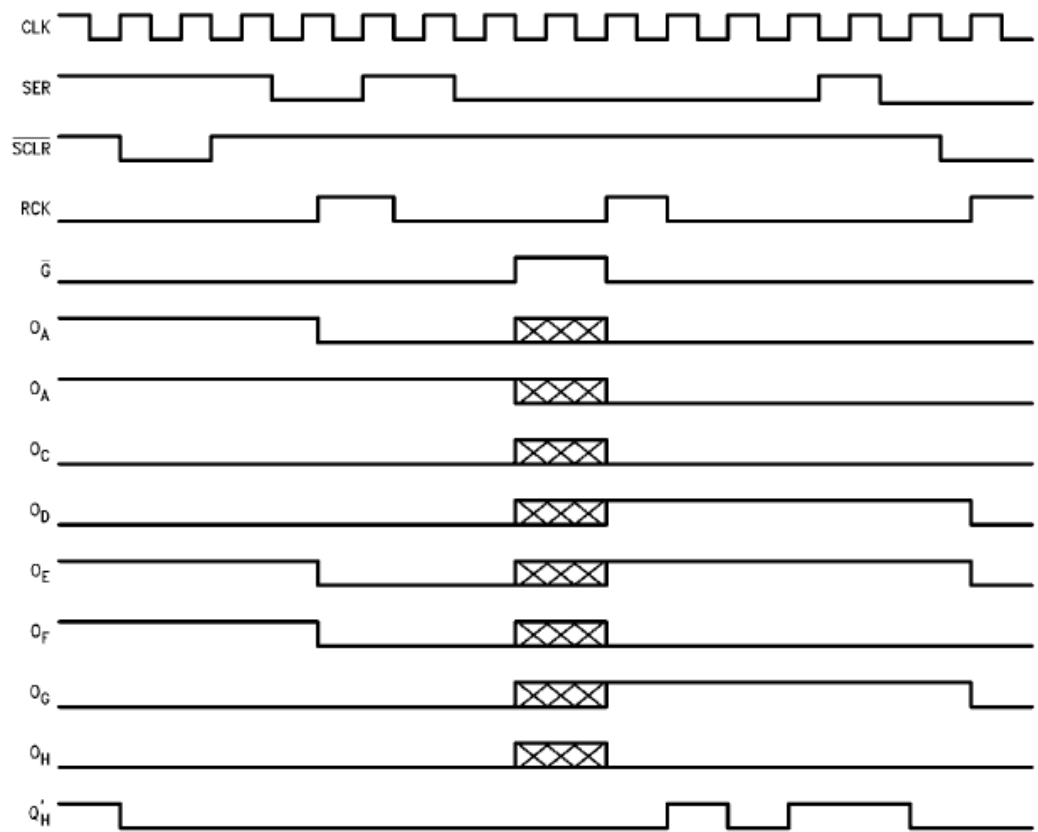


- Vcc, GND: dùng cấp nguồn cho IC hoạt động. Vcc được nối đến cực dương của nguồn (+5V), GND được nối đến cực âm của nguồn (0V)
- SER: Ngõ vào địa chỉ vào của IC 74HC595
- SCK: Chân clock
- RCK: Chân chốt địa chỉ
- $\overline{\text{SCLR}}$: Chân xóa dữ liệu đầu vào
- $\overline{\text{G}}$: Chân cho phép đầu ra của IC
- $O_A \rightarrow O_H$: Các ngõ ra của IC

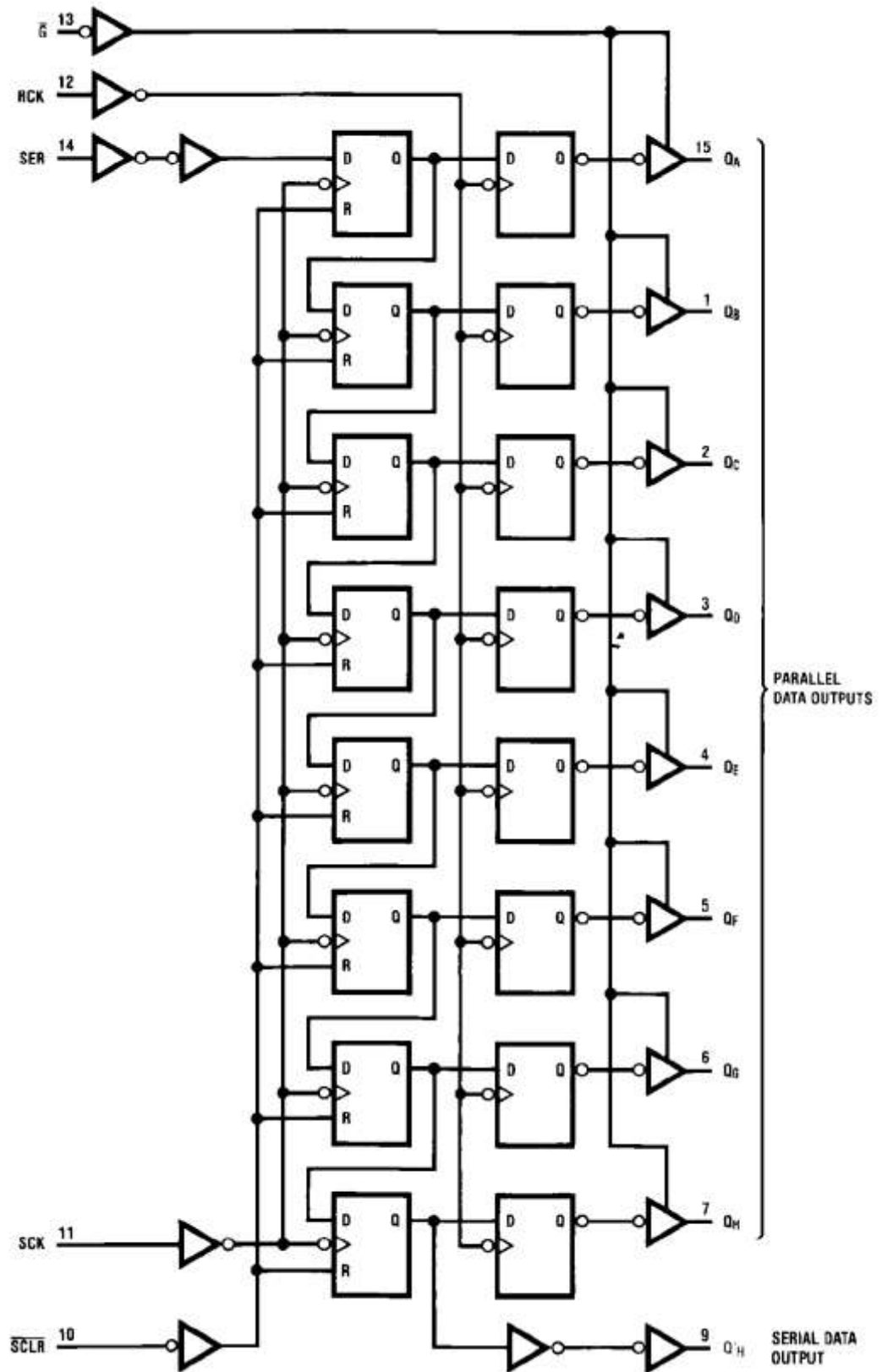
2.1.2.3. Bảng hoạt động của IC 74HC595:

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

2.1.2.4. Sơ đồ hoạt động:



2.1.2.5. Sơ đồ logic của IC 74HC595:



2.1.2.6. Nguyên tắc hoạt động:

IC 74HC595 hoạt động như sau

- Đưa chân \overline{G} xuống 0V, cho phép đầu ra của IC hoạt động
- Đưa chân \overline{SCLR} lên 5V, không cho phép xóa dữ liệu đầu vào
- Tạo một xung ở chân clock SCK
- Đưa dữ liệu vào chân dữ SER
- Tạo một xung vào chân chốt dữ liệu RCK

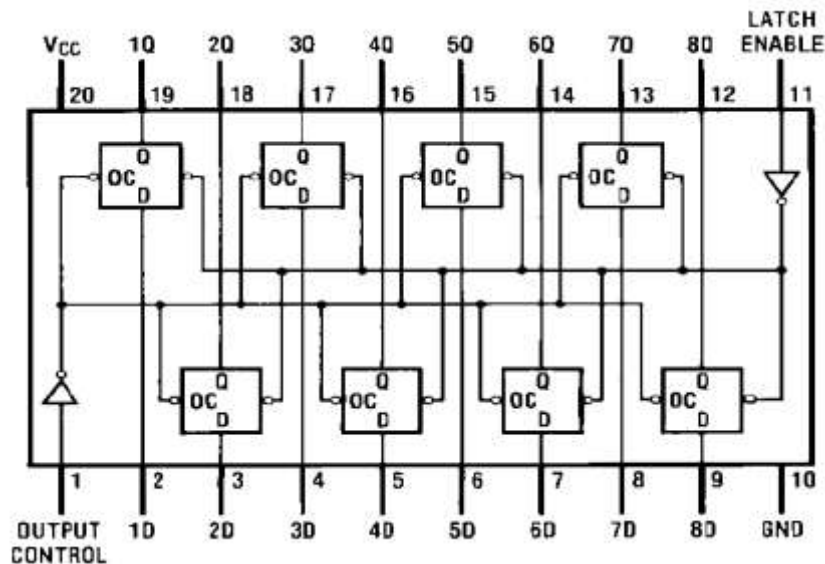
2.1.3. IC 74HC573

2.1.3.1. Mô tả chung:

IC 74HC573 là IC đệm có khả năng chốt đầu ra

- Thời gian chậm trễ trên đường truyền: 18 ns
- Phạm vi điện áp vận hành rộng: 2 -> 6 V
- Thích hợp với hệ thống Bus
- Đầu ra điều khiển: 15 LS-TTL tải

2.1.3.2. Sơ đồ chân:



- 1D -> 8D: Dữ liệu đầu vào
- Latch enable: Chốt đầu vào (tích cực mức cao)
- Output control: Điều khiển đầu ra tích cực mức thấp)
- 1Q -> 8Q: 8 đầu ra

- GND: mass (0V)
- Vcc: Điện áp dương (5V)

2.1.3.3. Bảng hoạt động của IC 74HC573:

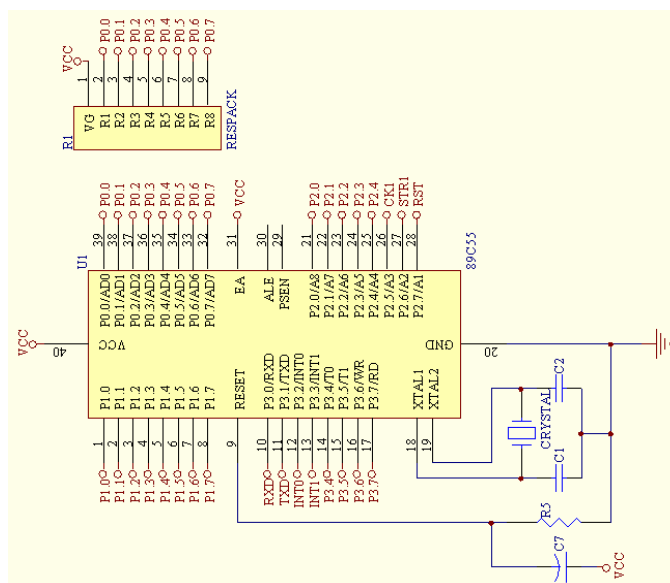
Output Control	Latch Enable	Data	Output
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

- H: Mức cao
- L: Mức thấp
- Q₀: Mức đầu ra trước đây dừng lại, nhập vào trạng thái thiết lập
- Z: Trở kháng cao
- X: Không quan tâm

2.2 THIẾT KẾ MẠCH MASTER

Mạch master bao gồm khối điều khiển, khối xuất dữ liệu hàng, bộ đệm. Khả năng mở rộng được tính đến gồm có : giao tiếp với max232 để kết nối với máy vi tính, kết nối với ROM ngoài để lưu trữ dữ liệu.

2.2.1 Thiết kế mạch hoạt động cho 89c55:



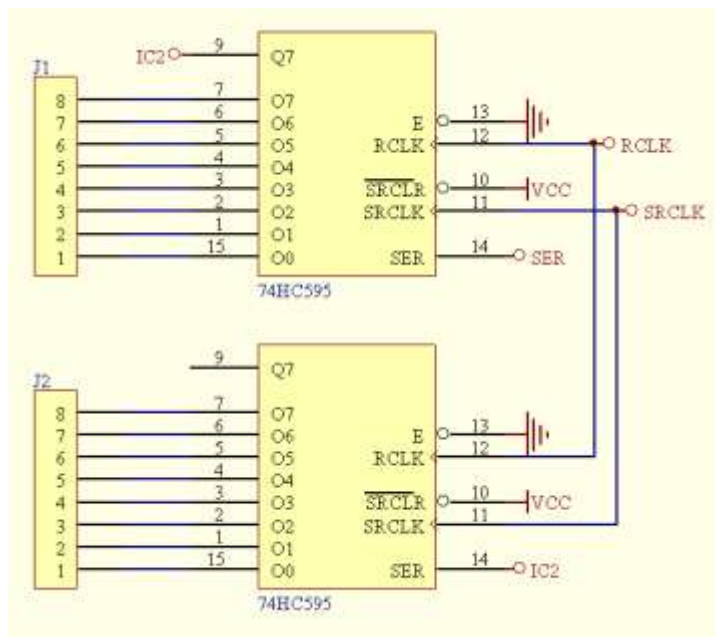
- Chân 40: Vcc(+5V)
- Chân 20: GND(0 V)
- Chân 31(EA): Vcc
- Chân 9 nối với mạch reset
- Chân 18,19 nối với mạch dao động
- Port 0 nối với trở thành 10k

Bộ dao động thạch anh có tác dụng tạo xung nhịp với tần số 12 MHz cho IC 89C55 hoạt động. Trở thành 10k được nối với port 0 (do cấu tạo của họ vi điều khiển 8051 thì port 0 phải có điện trở kéo)

2.2.2. Khối xuất dữ liệu hàng:

Đồ án này em tiến hành nghiên cứu làm bảng thông tin điện tử có kích thước 16 hàng*128 cột.

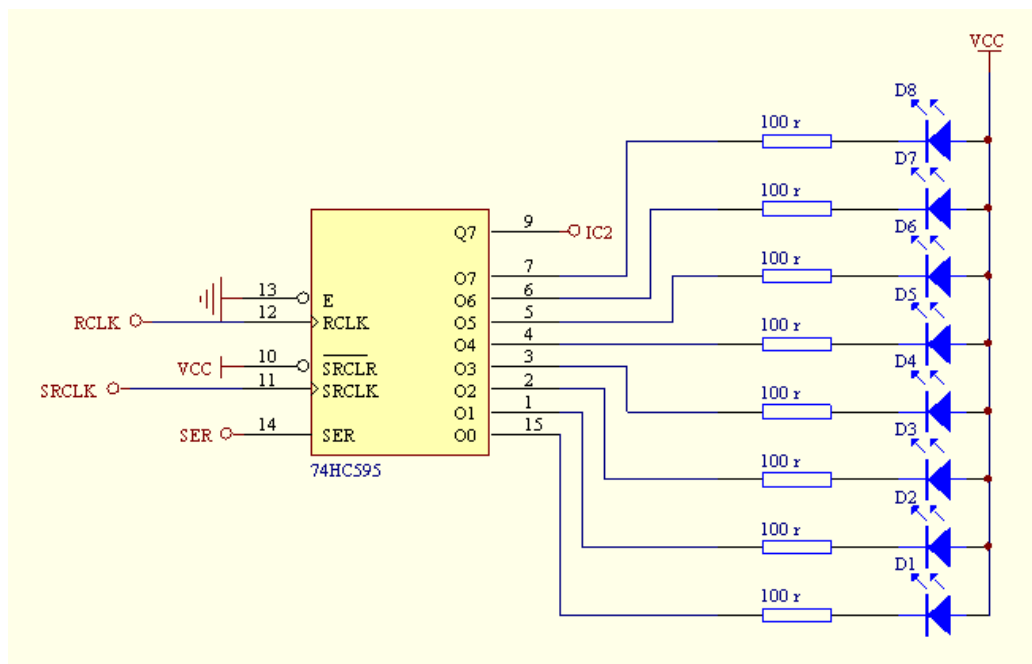
Để điều khiển bảng thông tin điện tử có 16 hàng thì ta phải sử dụng 2 port của vi điều khiển. Việc sử dụng này rất lãng phí tài nguyên sử dụng của vi điều khiển. Vì thế, ta sử dụng phương án truyền dữ liệu nối tiếp thay vì truyền dữ liệu song song. Để thực hiện truyền dữ liệu nối tiếp, ta dùng IC 74HC595. Với việc điều khiển 2 IC 74HC595, ta có thể truyền dữ liệu cho 16 hàng mà chỉ phải dùng có 3 chân của vi điều khiển.



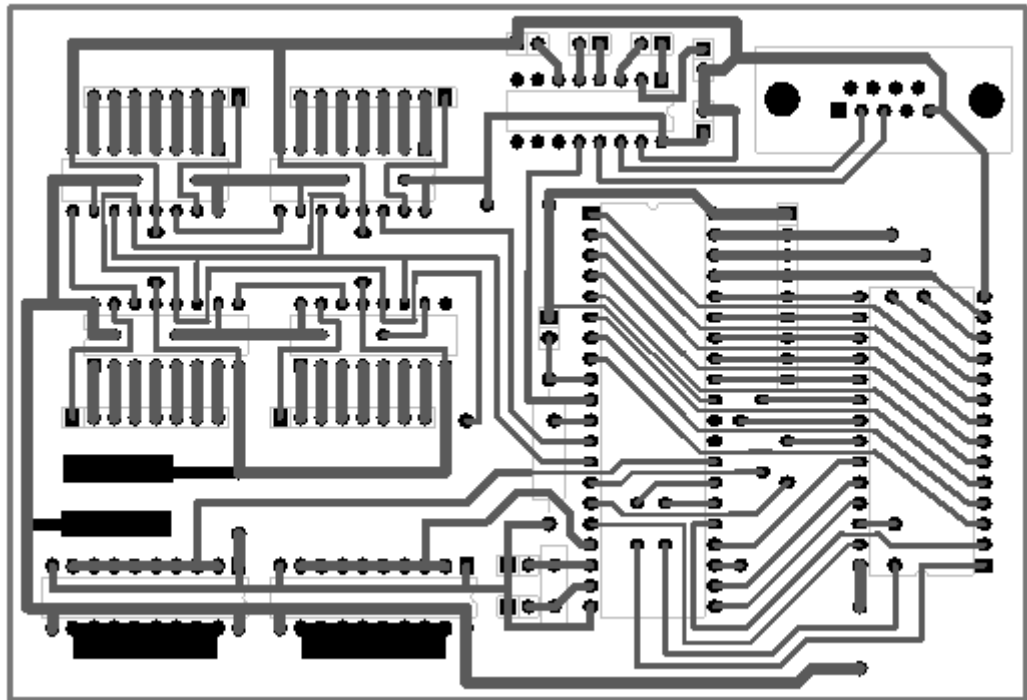
Nếu muốn làm bảng thông tin điện tử có kích thước rộng hơn (tăng số hàng), ta chỉ việc thực hiện ghép nối tiếp IC 74HC595

2.2.3. Khối ghi dịch và khối hiển thị:

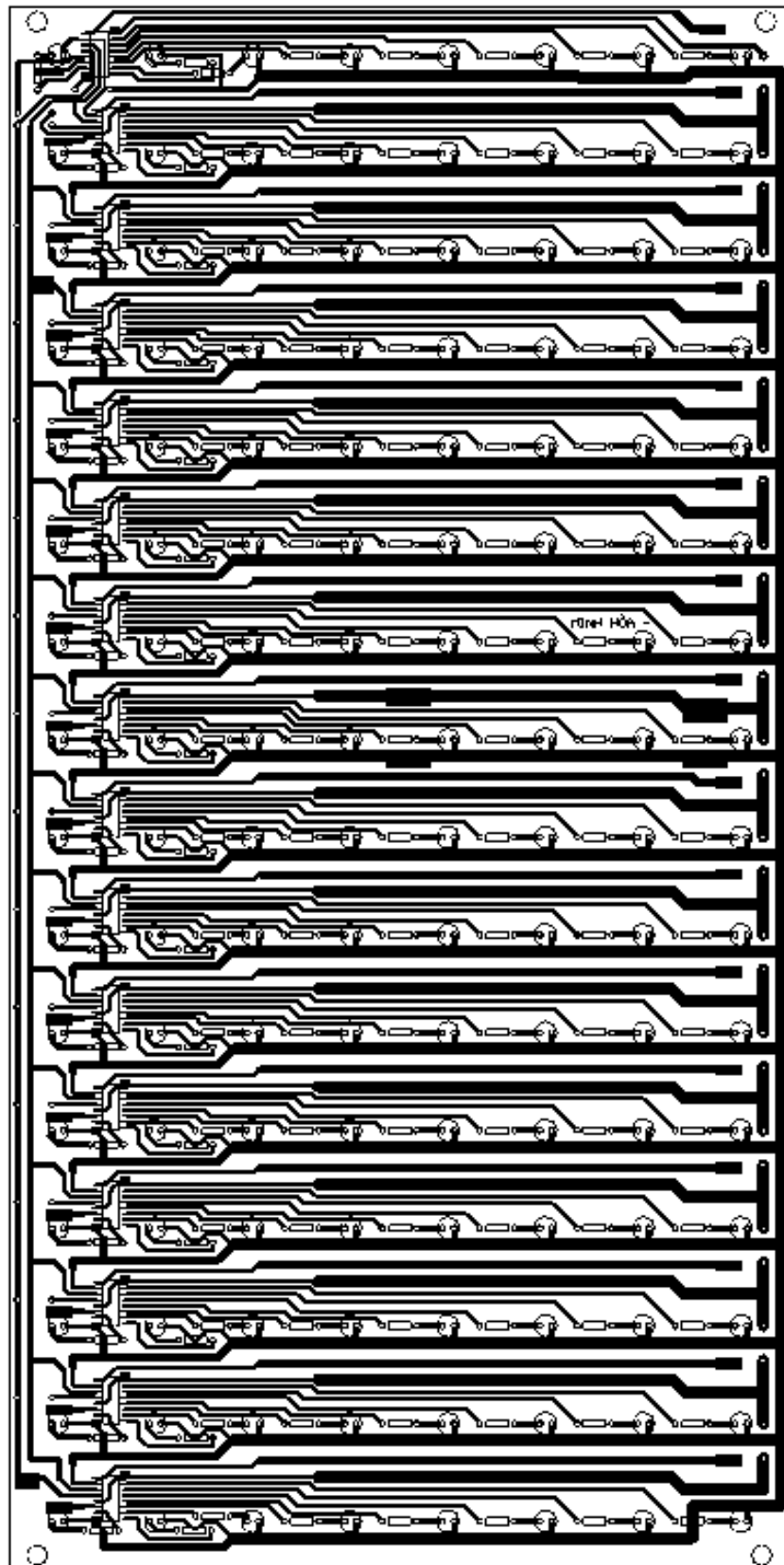
Em thiết kế khối ghi dịch và khối hiển thị vào cùng một modul có kích thước là 16 hàng* 8 cột. Với thiết kế như vậy, em có thể lắp ghép các bảng thông tin điện tử có kích thước khác nhau. Dưới đây là sơ đồ nguyên lý cho 1 hàng của 1 modul



2.4. SƠ ĐỒ MẠCH IN MẠCH MASTER



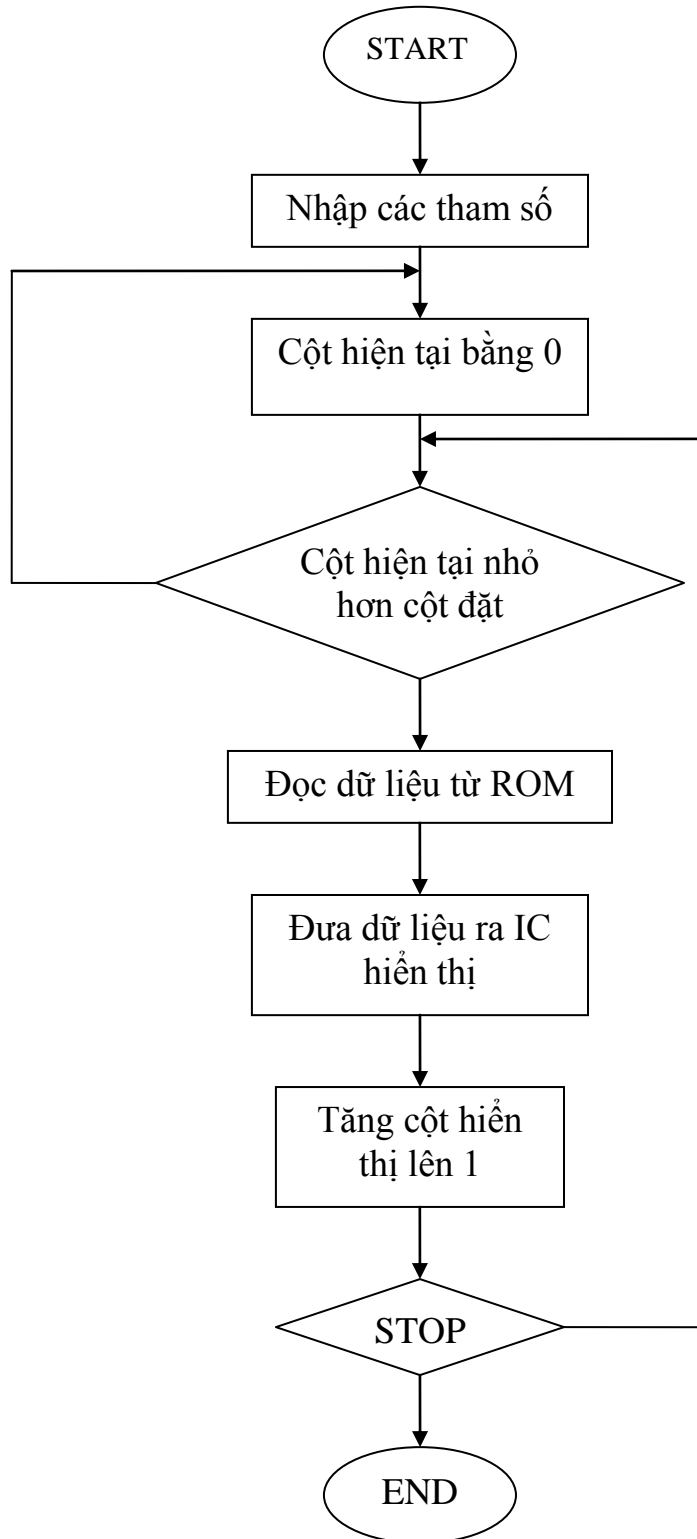
2.5 SƠ ĐỒ MẠCH IN MODUL



Chương 3 :

CHƯƠNG TRÌNH PHẦN MỀM

3.1. LƯU ĐỒ THUẬT TOÁN



3.2. CHƯƠNG TRÌNH PHẦN MỀM

```
$include(reg51.inc)
org 0000h
;
;
;diu khien ma tran 16*123
;89c55 truyen du lieu noi tiep qua 2 ic 74HC595
;16 tin hieu ra cua 2 ic 74HC595 (truyen du lieu) dua vao 16 chan data
cua 74HC595(ic dieu khien)
;
;
mov p3,#0ffh
dt bit p3.0
ck bit p3.1
rst bit p3.2; chot cua 2 ic truyen du lieu
rst1 bit p2.7;chot cua 16 ic dieu khien
ck1 bit p3.7;ck cua 16 ic dieu khien
start:
    mov dptr,#doantotnghiep
    mov R6,#192
    mov r5,#00
loop1: mov r7,#3
loop2: call hienthi
    djnz r7,2
    inc dptr
    inc dptr
    djnz R6,loop1
    inc r5
    cjne r5,#03,loop1
```

```

    jmp start
hienthi:
        push dph
        push dpl
    mov r1,#00
lap:
    setb ck1
    nop
    clr ck1
    ;;;;;;;;;;;;;;tao xung ck chi 16 ic dieu khien
        mov a,#0
        movc a,@a+dptr
        cpl a
        mov sbuf,a
        jnb ti,$
        clr ti
        ;inc r1
        inc dptr
    ;;;;;;;;;;;;;;truyen 8 du lieu dau cho ic 74595 dau tien
        mov a,#0
        movc a,@a+dptr
        cpl a
        mov sbuf,a
        jnb ti,$
        clr ti
        inc dptr
        inc r1

```

```

;;;;;;;;;;;;;truyen 8 du lieu cuoi cho ic 74595 tiep theo
setb p3.1
setb rst
nop
clr rst
;;;;;;;;;;;;;chot du lieu cua 2 ic truyen du lieu
call doi
;;;;;;;;;;;;;
CJNE r1,#128,lap
;;;;;;;;;;;;;
nop
setb ck1
setb rst1
nop
clr rst1
nop
pop dpl
pop dph
ret
doi:
    mov r0,#3
loop_doi:
    djnz r0,loop_doi
    RET
doantotnghiep:
dw .....
end

```

KẾT LUẬN

Được sự hướng dẫn nhiệt tình của thầy Th.s Nguyễn Trọng Thắng và sự nỗ lực của bản thân đến nay đồ án: “Nghiên cứu xây dựng hệ thống bảng thông tin điện tử” đã được em hoàn thành..

Chương 1: Tổng quan các phương án điều khiển

Chương 2: Thiết kế và thi công phần cứng của bảng thông tin điện tử

Chương 3: Thiết kế chương trình phần mềm

Đây là đồ án mang tính tổng hợp, kết hợp giữa kỹ thuật điện tử và kỹ thuật lập trình, qua đó em đã tích lũy một số kiến thức cơ bản về vi điều khiển, kỹ thuật số, thiết kế mạch và kỹ thuật lập trình. Đây là hành trang quý báu cho người sinh viên bước vào cuộc sống nghề nghiệp sau này.

Qua quá trình thực hiện đồ án, em đã tự đánh giá được phần nào còn hạn chế và ít nhiều bổ xung các kiến thức còn chưa biết trong thời gian học tại trường.

Em xin chân thành cảm ơn mọi sự giúp đỡ tận tình, quý báu của các thầy ,cô đã tạo nhiều điều kiện giúp em hoàn thành đồ án này. Một lần nữa rất mong sự đóng góp ý kiến của thầy, cô cùng với các bạn sinh viên tham khảo để đồ án hoàn thiện hơn

TÀI LIỆU THAM KHẢO

1. Nguyễn Tăng Cường – Phan Quốc Thắng(2004),**Cấu trúc và lập trình họ vi điều khiển 8051**, NXB Khoa học và kỹ thuật Hà Nội.
2. Ngô Diên Tập (2006) **Vi Điều Khiển Với Lập Trình C**, NXB Khoa học và kỹ thuật.
3. Tống Văn On (2006), **Thiết Kế Hệ Thống Với Họ 8051**, NXB Phương Đông
4. Tống Văn On – Hoàng Đức Hải (2008), **Họ Vi Điều Khiển 8051**, NXB Khoa học Xã Hội.