

**UBND TỈNH HẢI DƯƠNG
TRƯỜNG CAO ĐẲNG HẢI DƯƠNG**

**GIÁO TRÌNH
MÔN HỌC VI ĐIỀU KHIỂN**

NGÀNH: Điện công nghiệp và dân dụng

TRÌNH ĐỘ: Trung cấp

Hải dương , năm 2017

Chương 1: **Bé Vi      khi  n AT89C51**

1.1Gi  i thi  u chung:

Vi      khi  n (V  K) l  m m  t “h  ” Vi x   lý (VXL)      t  c ch  c trong m  t chip. N  i bao g  m:

- B   VXL
- B   nh  i ch  ng tr  nh (ROM/EPROM/EEPROM/FLASH).
- B   nh  i d   li  u (RAM).
- C  c thanh ghi ch  c n  ng, c  c c  ng I/O, c  c ch  i     u khi  n ng  t v   truy  n tin n  i ti  p.
- C  c b   th  i gian d  ng trong l  nh v  c chia t  n v   t  o th  i gian th  c.

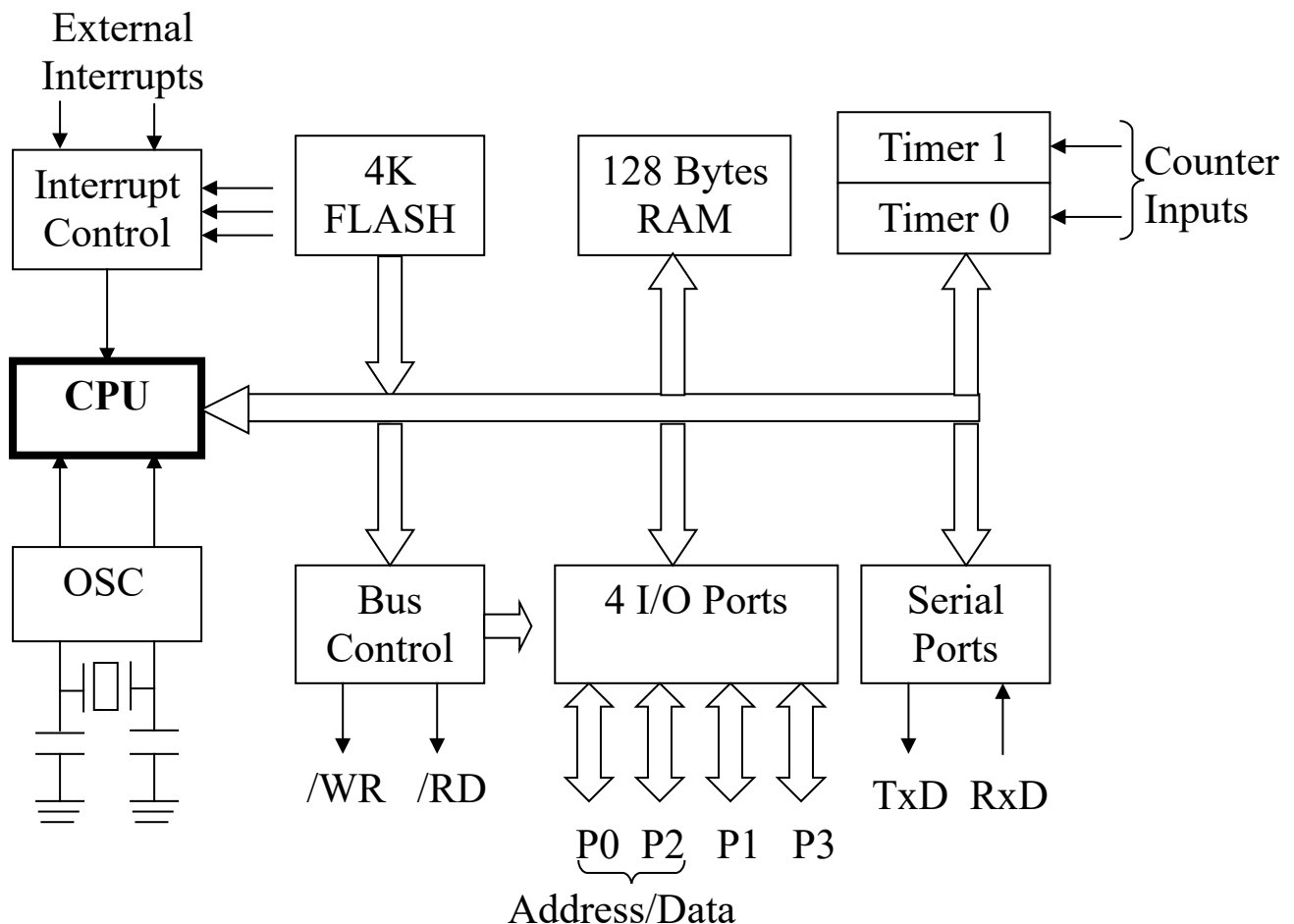
B   V  K c   th        l  p tr  nh          u khi  n c  c thi  t b  p th  ng tin, vi  n th  ng, thi  t b     o l  ng, thi  t b       u ch  nh c  ng nh c  c   ng d  ng trong c  ng ngh   th  ng tin v   k   thu  t     u khi  n t       ng. C   th   xem b   V  K nh m  t h   VXL On-chip,     i v  i h   AT89C51, n  i c       y      ch  c n  ng c  a m  t h   VXL 8 bit,     c     u khi  n b  i m  t h   l  nh, c   s   l  nh      m  nh, cho ph  p l  p tr  nh b  ng h  p ng   (Assembly).

1.2 S   kh  c nhau gi  a b   VXL v   b   V  K.

	VXL	V��K
Ph��n c��ng	CPU ����n ch��p.	CPU, RAM, ROM, Timers, SFR, m��ch giao ti��p, h�� th��ng ng��t v�� c��c ch��i ����u khi��n ng��t.....
T��p l��nh	S�� d��ng c��c t��p l��nh bao qu��t, m��nh v�� ki��u ����nh ����a ch��. C��c l��nh n��y c�� th�� truy xu��t d�� li��u l��n, th��c hi��n �� d��ng 1/2 Byte,	S�� d��ng c��c l��nh ����u khi��n xu��t nh��p, c�� th�� truy xu��t d�� li��u �� d��ng Bit ho��c Byte. C��c nh��m l��nh ch��nh: Chuy��n d�� li��u, ����u khi��n bi��n

	Byte, Word, Double Word.	logic, rĩ nh, nh ch→ng tr×nh, tĩnh to, n sè hãc vµ logic.
	VXL	VŞK
øng ðông	Trong c, c hÖ m, y vi tĩnh.	Trong c, c hÖ thèng ®iÖu khiÓn, ®o lêng vµ ®iÖu chØnh...

1.3. S→ ®ã khài.



H×nh 2.1. S→ ®ã khài hã VŞK
AT89C51

Bé V\$K 8 bit AT89C51 ho¹t ®éng ề tÇn sề 12 MHz, vớí bé nhí ROM 4Kbyte, bé nhí RAM 128 Byte c tró b^àn trong vµ cã thÓ mề réng bé nhí ra ngoµi. ề bé V\$K nµy cßn cã 4 cæng 8 bit (P0...P3) vµo/ ra 2 chiÒu ®Ó giao tiÕp vớí thiÕt bÞ ngo¹i vi. Ngoµi ra, nã cßn cã:

- 2 bé ®ình thêi 16 bit (Timer 0 vµ Timer 1)
- M¹ch giao tiÕp nèi tiÕp.
- Bé xõ lý bit (thao t_c trêñ c_c bit riêng rĩ).
- HÖ thèng ®iÒu khiÓn vµ xõ lý ng³/₄t.
- C_c k^ành ®iÒu khiÓn/ d÷ liÒu/ ®p^à chØ.
- CPU
- C_c thanh ghi chøc n^ëng ®Æc biÕt (SFR).

...

Tuy nhiêñ, tuú thuéc vµo tổng hã V\$K c^ña tổng h^àng s[¶]ñ xuÊt kh_c nhau mµ tÝnh n^ëng còng nh ph¹m vi øng dông c^ña mçi bé V\$K lµ kh_c nhau, vµ chóng ®ìc thÓ hiÕn trong c_c b[¶]ng thèng k^à sau:

Hã V\$K	ROM (bytes)	RAM (bytes)	Tèc ®é (MHz)	C _c ch [©] n I/O	Timer/ Counter	UART	Ngu ^ả n ng ³ / ₄ t
8051							
8031AH	ROMLESS	128	12	32	2	1	5
8051AH	4K ROM	128	12	32	2	1	5
8051AHP	4K ROM	128	12	32	2	1	5
8751H	4K EPROM	128	12	32	2	1	5
8751BH	4K EPROM	128	12	32	2	1	5
8052							
8032AH	ROMLESS	256	12	32	3	1	6
8052AH	8K ROM	256	12	32	3	1	6
8752BH	8K EPROM	256	12	32	3	1	6
80C51				32			
80C31BH	ROMLESS	128	12,16	32	2	1	5
80C51BH	4K ROM	128	12,16	32	2	1	5
80C31BHP	4K ROM	128	12,16	32	2	1	5
87C51	4K EPROM	128	12,16, 20,24	32	2	1	5

8xC52/54/58							
80C32	ROMLESS	256	12,16,20,24	32	3	1	6
80C52	8K ROM	256	12,16,20,24	32	3	1	6
87C52	8K EPROM	256	12,16,20,24	32	3	1	6
80C54	16K ROM	256	12,16,20,24	32	3	1	6
87C54	16K EPROM	256	12,16,20,24	32	3	1	6
Hä V\$K	ROM (bytes)	RAM (bytes)	Tèc ®é (MHz)	C, c ch©n I/O	Timer/Counter	UART	Nguån ng ^{3/4} t
80C58	32K ROM	256	12,16,20,24	32	3	1	6
87C58	32K EPROM	256	12,16,20,24	32	3	1	6
8xL52/54/58							
80L52	8K ROM	256	12,16,20	32	3	1	6
87L52	8K OTP ROM	256	12,16,20	32	3	1	6
80L54	16K ROM	256	12,16,20	32	3	1	6
87L54	16K OTP ROM	256	12,16,20	32	3	1	6
80L58	32K ROM	256	12,16,20	32	3	1	6
87L58	32K OTP ROM	256	12,16,20	32	3	1	6
...							

Bíng 1.1. C, c th«ng sè cña c, c hä V\$K thuéc h-ng Intel (MSC 51)

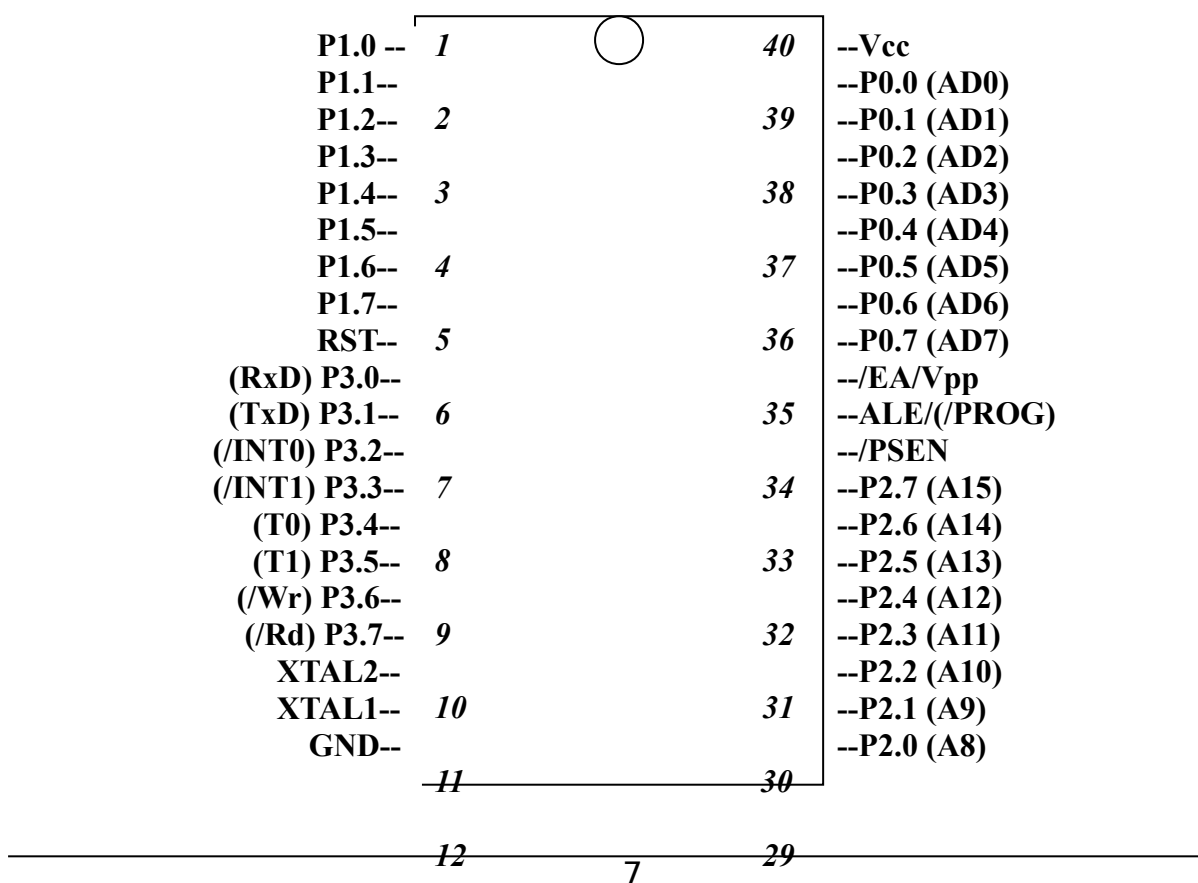
Hä V\$K	Bé nhí ch- -ng tr×nh(Byte s)	Bé nhí d÷ liÖu (Bytes)	Timer 16 bit	C«ng nghÖ
AT89C105	1K Flash	64 RAM	1	CMOS

1				
AT89C205	2K Flash	128 RAM	2	CMOS
1				
AT89C51	4K Flash	128 RAM	2	CMOS
AT89C52	8K Flash	256 RAM	3	CMOS
AT89C55	20K Flash	256 RAM	3	CMOS
AT89S825	8K Flash	256 RAM + 2K EEPROM	3	CMOS
2				
AT89S53	12K Flash	256 RAM	3	CMOS

Bảng 1.2. Các thông số của các họ VSK thuộc hãng Atmel

.Trong khuôn khổ tài liệu này, tôi sẽ tiếp trung trình bày cấu trúc phần cứng của họ VSK AT89C51 thuộc hãng Atmel.

1.4. Sơ đồ chân tín hiệu của 80C51/AT89C51.



14	27
15	26
16	25
17	24
18	23

Hình 1.2. IC 80C51⁴²/AT89C51

20	21
----	----

Chức năng của các chân tín hiệu như sau:

- P0.0 ® Ôn P0.7 là các chân của cổng 0.
- P1.0 ® Ôn P1.7 là các chân của cổng 1.
- P2.0 ® Ôn P2.7 là các chân của cổng 2
- P3.0 ® Ôn P3.7 là các chân của cổng 3
- RxD: Nhãn tín hiệu kiểu nối tiếp.
- TxD: Truy vấn tín hiệu kiểu nối tiếp.
- /INT0: Ngắt ngoại 0.
- /INT1: Ngắt ngoại 1.
- T0: Chân vào 0 của bộ Timer/Counter 0.
- T1: Chân vào 1 của bộ Timer/Counter 1.
- /Wr: Ghi dữ liệu bộ nhớ ngoại.
- /Rd: Đọc dữ liệu bộ nhớ ngoại.
- RST: Chân vào Reset, tích cực ở mức logic cao trong khoảng 2 chu kỳ máy.
- XTAL1: Chân vào mạch khuếch đại dao động
- XTAL2: Chân ra tải mạch khuếch đại dao động.
- /PSEN : Chân cho phép đọc bộ nhớ chương trình ngoại (ROM ngoại).
- ALE (/PROG): Chân tín hiệu cho phép chèn bộ nhớ truy cập bộ nhớ ngoại, khi On-chip xuất ra byte tiếp của bộ nhớ. Tín hiệu chèn để kích hoạt ở mức cao, tần số xung chèn = 1/6 tần số dao động của bộ VSK. Nó cũng dùng cho các bộ Timer ngoại hoặc cho mô-đun xung Clock. Nó cũng dùng làm chân nhả xung vào bộ nhớ chương trình cho Flash (hoặc EEPROM) bên trong On-chip khi nó ở mức thấp.
- /EA/Vpp: Cho phép On-chip truy cập bộ nhớ chương trình ngoại khi /EA=0, nếu /EA=1 thì On-chip sẽ làm việc với bộ nhớ chương trình nội bộ. Khi chân này để cấp nguồn để nạp 12V (Vpp)

th× On-chip ®¶m nhËn chc nng np chng trnh cho Flash bn trong n.

- Vcc: Cung cp dng ngun cho On-chip (+ 5V).
- GND: ni mt.

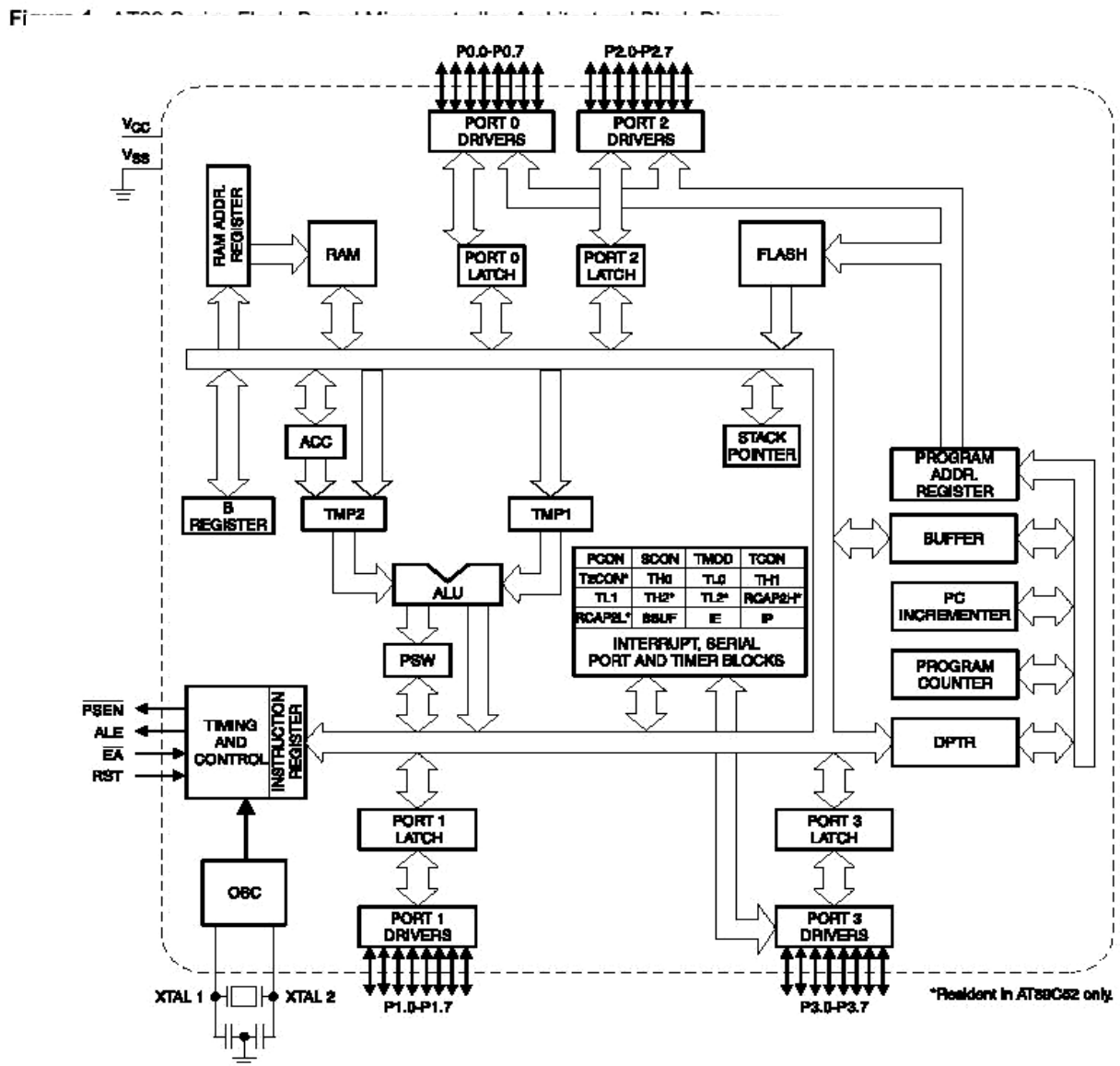
1.5. Chc nng cc thnh phn ca AT89C51:

1.5.1. Cc thanh ghi chc nng ®Æc bit.

SFR ®¶m nhim cc chc nng khc nhau trong On-chip. Chng nm  RAM bn trong On-chip, chim vng khng gian nh 128 Byte ®c ®nh ®a ch t 80h ®n FFh. Cu trc ca SFR bao gm cc chc nng th hin  bng 2.3 v bng 2.4.

Thanh ghi	Ni 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 1.3. Chức năng riêng của tổng thanh ghi trong SFR



Hình 1.3. Sơ đồ khối của AT89

Symbol	Name	Address	Reset Values
--------	------	---------	--------------

* ACC	Thanh ghi tÝch luÛ	0E0h	00000000b
* B	Thanh ghi B	0F0h	00000000b
* PSW	Tổ tr¹ng th,i ch-ng tr×nh	0D0h	00000000b
SP	Con trá ng"n xÕp	81h	00000111b
DP0L	Byte cao cña con trá d÷ liÖu 0	82h	00000000b
DP0H	Byte thÊp cña con trá d÷ liÖu 0	83h	00000000b
* P0	Cæng 0	80h	11111111b
* P1	Cæng 1	90h	11111111b
Symbol	Name	Address	Reset Values
* P2	Cæng 2	0A0h	11111111b
* P3	Cæng 3	0B0h	11111111b
* IP	TG ®iÖu khiÕn ng¾t u tiªn	0B8h	xxx00000b
* IE	TG ®iÖu khiÕn cho phÐp ng¾t	0A8h	0xx00000b
TMOD	§iÖu khiÕn kiÓu Timer/Counter	89h	00000000b
* TCON	TG ®iÖu khiÕn Timer/Counter	88h	00000000b
TH0	Byte cao cña Timer/Counter 0	8Ch	00000000b
TL0	Byte thÊp cña Timer/Counter 0	8Ah	00000000b
TH1	Byte cao cña Timer/Counter 1	8Dh	00000000b
TL1	Byte thÊp cña Timer/Counter 1	8Bh	00000000b
* SCON	Serial Control	98h	00000000b
SBUF	Serial Data Buffer	99h	indeterminate
PCON	Power Control	87h	0xxx0000b

* : cã thÓ ®Þnh ®Þa chØ bit, x: kh«ng ®Þnh nghÜa

B¶ng 1.4. §Þa chØ, ý nghÜa vµ gi, trÞ cña c,c SFR sau khi Reset

1.5.1.1. Thanh ghi ACC: Lưu 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 tệp lõi của On-chip, mã thông điệp quy định trên ghi của A.

1.5.1.2. Thanh ghi B : Thanh ghi này để dùng khi thực hiện các phép toán nhân và chia. Sẽ với các lõi khác, mã cả tho xem như lưu thanh ghi để tìm thời. Thanh ghi B dài 8 bits. Mã thông điệp dùng chung với thanh ghi A trong các phép toán nhân hoặc chia.

1.5.1.3. Thanh ghi SP: Thanh ghi con trỏ ngăn xếp dài 8 bit. SP chứa địa chỉ của địa chỉ hiện tại đang là địa chỉ ngăn xếp. Giá trị của mã để từ để tăng lên khi thực hiện lệnh PUSH trực khi địa chỉ để lưu trữ trong ngăn xếp. SP sẽ từ để giảm xuống khi thực hiện lệnh POP. Ngăn xếp mã tho để là bộ nhớ trong RAM on-chip, nhưng sau khi khi để tăng lên hoặc giảm xuống con trỏ ngăn xếp mã để sẽ trả lại địa chỉ của bộ nhớ 07h, và vậy ngăn xếp sẽ bắt đầu địa chỉ của bộ nhớ 08h. Ta cũng mã tho để phần con trỏ ngăn xếp tại địa chỉ của bộ nhớ mong muốn bằng các lõi di chuyển địa chỉ thông qua phần của bộ nhớ tốc thời.

1.5.1.4. Thanh ghi DPTR: Thanh ghi con trỏ địa chỉ (16 bit) bao gồm 1 thanh ghi byte cao (DPH-8bit) và 1 thanh ghi byte thấp (DPL-8bit). DPTR mã tho để dùng như thanh ghi 16 bit hoặc 2 thanh ghi 8 bit để để. Thanh ghi này để dùng để truy cập RAM ngoại.

1.5.1.5. Ports 0 to 3: P0, P1, P2, P3 là các chân của các cổng 0, 1, 2, 3 tương ứng. Mỗi chân gồm 8 bit. Khi ghi mức logic 1 vào một bit của chân, thì chân ra tương ứng của cổng là mức logic cao. Còn khi ghi mức logic 0 vào một bit của chân thì chân ra tương ứng của cổng là mức logic thấp. Khi các cổng để làm nhiệm vụ chức năng như các bộ vào và ra tương tự bên ngoài của các chân cổng sẽ để để là bit chân tương ứng. Tất cả 4 cổng của on-chip để lưu cổng I/O hai chiều, mỗi cổng để cả 8 chân ra, bên trong mỗi chân bit mã bé "Pullup-tương ứng" do để năng

cao khả năng nội bộ của các vi xử lý (các cổng giao tiếp với 4 cổng 8 bit loại TTL).

1.5.1.6. Thanh ghi SBUF: Tổng số dữ liệu nội bộ giao gồm 2 thanh ghi riêng biệt, một thanh ghi để ghi và một thanh ghi để đọc. Khi dữ liệu được chuyển tới SBUF, dữ liệu sẽ đi vào bộ ghi, và được gửi về bộ nhớ để chờ biến thành dạng truyền tin nội bộ. Khi dữ liệu được truyền đi từ SBUF, dữ liệu sẽ đi ra từ bộ đọc.

1.5.1.7. Các Thanh ghi Timer: Các cổng thanh ghi (TH0, TL0), (TH1, TL1) là các thanh ghi để đếm 16 bit đồng bộ với các bộ Timer/Counter 0 và 1.

1.5.1.8. Các thanh ghi điều khiển: Các thanh ghi chức năng để điều khiển: IP, IE, TMOD, TCON, SCON, và PCON bao gồm các bit trạng thái và điều khiển liên quan với hệ thống ngắt, các bộ Timer/Counter và các nội bộ. Chúng sẽ được mô tả tiếp theo sau.

1.5.1.9. Thanh ghi PSW: Tổ chức trạng thái chung của vi xử lý để điều khiển. PSW có 8 bit, mỗi bit đều có một chức năng cụ thể. Thanh ghi này cho phép truy cập đến các bit.

* CY: Carry. Trong các phép toán sẽ xảy ra, nếu cần thì phép cộng bit 7 hoặc cần sẽ tiến hành để bit 7 của CY để tiếp theo bit 1.

* AC: Carry flag (liên quan với mô-đem BCD). Khi cộng các giá trị thập phân BCD, nếu cần thì sẽ tiến hành để bit 3 chuyển sang bit 4 của AC để tiếp theo bit 1. Khi giá trị thập phân BCD, lệnh cộng phải liên quan đến việc tiếp theo theo bộ nhớ **DA A** (hiệu chỉnh tiếp theo thanh ghi A) để các kết quả liên quan 9 và giá trị tiếp theo.

* F0: Carry 0 (Liên quan với các mô-đem chung của các số nguyên)

* RS1: Bit 1 @iỜu khiỐn chẵn b"ng thanh ghi.

* RS0: Bit 0 @iỜu khiỐn chẵn b"ng thanh ghi.

Lu ý: RS0, RS1 @íc @Æt/xo, b»ng phÇn mÒm @Ó x,c @Pnh b"ng thanh ghi @ang ho¹t @éng (Chẵn b"ng thanh ghi b»ng c, ch @Æt tr¹ng th, i cho 2 bit nựy)

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

Bíng 1.5. Chẵn b"ng thanh ghi_

* OV: Cê trựn. Khi thùc hiỐn c,c phĐp to,n céng hoÆc trở mụ xuÊt hiỐn mét trựn sè hắc, th× OV @íc @Æt b»ng 1. Khi c,c sè cả dÊu @íc céng hoÆc @íc trở, phÇn mÒm cả thÓ kiỐm tra OV @Ó x,c @Pnh xem kỐt qu¶ cả n»m trong tÇm hay kh«ng. Vói phĐp céng c,c sè kh«ng dÊu, OV @íc bá qua. KỐt qu¶ lín h-n +128 hoÆc nhá h-n -127 sỉ @Æt OV=1.

* -: Bit dịnh cho ngêi sô đông tù @Pnh nghÜa(NỜu cÇn).

* P: Cê ch½n lĩ. §íc tù @éng @Æt/ xo, b»ng phÇn cøng trong mçi chu tr×nh lỜnh @Ó chØ thP sè ch½n hay lĩ cả bit 1 trong thanh ghi tÝch luũ. Sè c,c bit 1 trong A céng vói bit P lu«n lu«n lụ sè ch½n.

1.5.1.10. Thanh ghi PCON: Thanh ghi @iỜu khiỐn nguấn.

* SMOD: Bit t¹o tèc @é Baud gÊp @«i. NỜu Timer 1 @íc sô đông @Ó t¹o tèc @é baud vµ SMOD=1, th× tèc @é Baud @íc

tình trạng gấp ® khi cần truy cập nội tiếp ® để dừng bởi các kiểu 1, 2 hoặc 3.

* -: Khi số đồng, các bit này cả thảy ® để ở các bit VXL trong trạng thái. Khi số đồng không ® phép từ ® phân nghĩa cho các bit này.

* GF0, GF1: Chế độ cho các môc ® ý chung (®a môc ® ý).

* PD: bit nguồn gi¶m. §ét bit này ở mức tích cực ® ở mức thấp ® nguồn gi¶m trong AT89C51. Khi cả thảy ra khỏi ® ở mức Reset.

* IDL: bit chặn ® ở mức. §ét bit này ở mức tích cực ® ở mức thấp ® nguồn gi¶m (Chế độ không làm việc) trong AT89C51.

Lưu ý: Nếu PD và IDL cũng ® kích hoạt cũng 1 lúc ở mức tích cực, thì PD ® ưu tiên thực hiện trước. Khi ra khỏi ® ở mức 1 ng¾t hoặc Reset thì trở về.

1.5.1.11. Thanh ghi IE: Thanh ghi cho phép ng¾t

* EA: Nếu EA=0, không cho phép bất cứ ng¾t nào hoạt động. Nếu EA=1, mọi nguồn ng¾t riêng biệt ® phép hoặc không ® phép hoạt động ® ở mức các ®ét hoặc xóa bit Enable của nó.

* -: Khi dừng, khi số đồng không nên ® phân nghĩa cho Bit này, bởi vì cả thảy ® để ở các bit AT89 trong trạng thái.

* ET2: Bit cho phép hoặc không cho phép ng¾t Timer 2.

* ES: Bit cho phép hoặc không cho phép ng¾t cần nội tiếp (SPI và UART).

* ET1: Bit cho phép hoặc không cho phép ng¾t timer 1

* EX1: Bit cho phép hoặc không cho phép ng¾t ngoại 1.

* ET0: Bit cho phép hoặc không cho phép ngắt trừn bé Timer 0

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

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

* - : Không dùng, ngôi số đông không nên ghi "1" vào các Bit này.

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

* PS: Số phân chia mức ưu tiên của ngắt cần ngắt tiếp.

* PT1: Số phân chia mức ưu tiên của ngắt Timer 1.

* PX1: Số phân chia mức ưu tiên của ngắt ngoại 1.

* PT0: Số phân chia mức ưu tiên của ngắt Timer 0.

* PX0: Số phân chia mức ưu tiên của ngắt ngoại 0.

1.5.1.13. Thanh ghi TCON : Thanh ghi điều khiển bé Timer/Counter

* TF1: Chế độ trừn Timer 1. Đặt để bắt đầu cộng khi bé Timer 1 trừn.

Đặt xo, bắt đầu cộng khi bé vì xử lý hính tít ch—ng trxn con pho vô ngắt.

* TR1: Bit điều khiển bé Timer 1 ho¹t đặt. Đặt để/xo, bắt đầu mòm đặt điều khiển bé Timer 1 ON/OFF

* TF0: Chế độ trừn Timer 0. Đặt để bắt đầu cộng khi bé Timer 0 trừn. Đặt xo, bắt đầu cộng khi bé vì xử lý hính tít ch—ng trxn con pho vô ngắt.

* TR0: Bit điều khiển bé Timer 0 ho¹t đặt. Đặt để/xo, bắt đầu mòm đặt điều khiển bé Timer 0 ON/OFF.

* IE1: Chế độ ngắt ngoại 1. Đặt để bắt đầu cộng khi sên xung của ngắt ngoại 1 đặt ph, t hiÖn. Đặt xo, bắt đầu cộng khi ngắt đặt xử lý.

* IT1: Bit điều khiển ngắt 1 đặt t¹o ra ngắt ngoại. Đặt để/xo, bắt đầu mòm.

* IE0: Chế độ ngắt ngoại 0. Đặt để bắt đầu cộng khi sên xung của ngắt ngoại 0 đặt ph, t hiÖn. Đặt xo, bắt đầu cộng khi ngắt đặt xử lý.

* IT0: Bit @iÒu khiÓn ng^{3/4}t 0 @Ó t¹o ra ng^{3/4}t ngoµi. §íc @Æt/ xo_j bi phÇn mÒm.

1.5.1.14. Thanh ghi TMOD: Thanh ghi @iÒu khiÓn kiÓu Timer/Counter

* GATE: Khi TRx @íc thiÕt lp vµ GATE=1, b TIMER/COUNTERx ho¹t @ng chØ khi ch©n INTx  mc cao. Khi GATE=0, TIMER/COUNTERx s ho¹t @ng chØ khi TRx=1.

* C/(/T): Bit nµy cho phÐp chn chc nng lµ Timer hay Counter.

- Bit nµy @íc xo_j @Ó thùc hiÕn chc nng Timer

- Bit nµy @íc @Æt @Ó thùc hiÕn chc nng Counter

* M0, M1: Bit chn Mode, @Ó x_jc @Pnh tr¹ng th_ji vµ kiÓu Timer/Counter:

- M1=0, M0=0: Chn kiÓu b Timer 13 bit. Trong @ THx dµi 8 bit, cßn TLx dµi 5 bit.

- M1=0, M0=1: Chn kiÓu b Timer 16 bit. THx vµ TLx dµi 16 bit @íc ghÐp tÇng.

- M1=1, M0=0: 8 bit Auto reload. C_jc thanh ghi tù @ng n¹p l¹i mi khi bP trµn. Khi b Timer bP trµn, THx dµi 8 bit @- c gi÷ nguyn gi_j trP, cßn gi_j trP n¹p l¹i @íc @a vµo TLx.

- M1=1, M0=1: KiÓu ph©n chia b Timer. TL0 lµ 1 b Timer/Counter 8 bit, @íc @iÒu khiÓn b»ng c_jc bit @iÒu khiÓn b Timer 0, cßn TH0 chØ lµ b Timer 8 bit, @íc @iÒu khiÓn b»ng c_jc bit @iÒu khiÓn Timer 1.

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

1.5.1.15. Thanh ghi SCON:

SCON lµ thanh ghi tr¹ng th_ji vµ @iÒu khiÓn cng ni tiÕp. N khng nh÷ng cha c_jc bit chn chØ @, mµ cßn cha bit d÷ liÕu thø 9 dµnh cho viÕc truyÒn vµ nhn tin (TB8 vµ RB8) vµ cha c_jc bit ng^{3/4}t cng ni tiÕp.

* SM0, SM1: Lµ c_jc bit cho phÐp chn chØ @ cho cng truyÒn ni tiÕp.

SM	SM	Mode	§Æc @iÓm	Tc	@
----	----	------	----------	-----	----

0	1			Baud
0	0	0	Thanh ghi dPch	$F_{osc}/12$
0	1	1	8 bit UART	Cả thÓ thay ®æi (®íc ®Æt bëi bé Timer)
1	0	2	9 bit UART	$F_{osc}/64$ hoÆc $F_{osc}/32$
1	1	3	9 bit UART	Cả thÓ thay ®æi (®íc ®Æt bëi bé Timer)

Bíng1.6. Chän Mode trong SCON

* SM2: Cho phĐp truyÒn tin ®a xö lý, thÓ hiÖn ë Mode 2 vµ 3. ë chÕ ®é 2 hoÆc 3, nÕu ®Æt $SM2 = 1$ th× RI sÏ kh«ng ®íc kých ho¹t nÕu bit d÷ liÖu thø 9 (RB8) nhËn ®íc gi, trÞ b»ng 0. ë Mode 1, nÕu $SM2=1$ th× RI sÏ kh«ng ®íc kých ho¹t nÕu bit đồng cả hiÖu lúc ®· kh«ng ®íc nhËn. ë chÕ ®é 0, $SM2$ nªn b»ng 0

* REN: Cho phĐp nhËn nòi tiÕp. §íc ®Æt hoÆc xo, bëi phÇn mÒm ®Ó cho phĐp hoÆc kh«ng cho phĐp nhËn.

* TB8: Lµ bit d÷ liÖu thø 9 mµ sÏ ®íc truyÒn ë Mode 2 vµ 3. §íc ®Æt hoÆc xo, bëi phÇn mÒm.

* RB8: Lµ bit d÷ liÖu thø 9 ®· ®íc nhËn ë Mode 2 vµ 3. ë Mode 1, nÕu $SM2=0$ th× RB8 lµ bit đồng ®· ®íc nhËn. ë Mode 0, RB8 kh«ng ®íc sö dông.

* TI: Cê ng¾t truyÒn. §íc ®Æt bëi phÇn cøng t¹i cuèi thêi ®iÓm cña bit thø 8 trong Mode 0, hoÆc ®Çu thêi ®iÓm cña bit đồng trong c, c Mode kh, c. ë bÊt kú qu, tr×nh truyÒn nòi tiÕp nµo, nã còng ph¶i ®íc xo, b»ng phÇn mÒm.

* RI: Cê ng¾t nhËn. §íc ®Æt bëi phÇn cøng t¹i cuèi thêi ®iÓm cña bit thø 8 trong Mode 0, hoÆc ë gi÷a thêi ®iÓm cña bit đồng trong c, c Mode kh, c. ë bÊt kú qu, tr×nh nhËn nòi tiÕp nµo (trõ trêng híp ngo¹i lÖ, xem SM2), nã còng ph¶i ®íc xo, b»ng phÇn mÒm.

1.5.2. Khài t¹o thêi gian vư bé @Ổm (Timer/Counter).

On-chip AT89C51 cã 2 thanh ghi Timer/Counter dui 16 bit, @ã lư: Timer 0 vư Timer 1. Trong On-chip AT89C52, ngoi Timer 0 vư Timer 1 nã cãn cã thãm bé Timer 2. C¶ 3 bé Timer nưy @Ờu cã thÓ @ic @iỜu khiỐn @Ó thuc hiỐn chøc n'ng thêi gian hay bé @Ổm, th«ng qua thanh ghi TMOD.

Khi thanh ghi Timer/Counter lưm viỐc ẽ kiỐu Timer, thx sau mçi chu kú mưy néi dung trong thanh ghi @ic gia t'ng thãm 1 @-n vư. Vx vỄy thanh ghi nưy @Ổm sè chu kú mưy. Mét chu kú mưy cã 12 chu kú dao @éng, do @ã tẽc @é @Ổm cĩa thanh ghi lư 1/12 tCn sè dao @éng.

Khi thanh ghi Timer/Counter lưm viỐc ẽ kiỐu Counter, xung nhp bãn ngoi @ic @a vư @Ó @Ổm ẽ T0 hoÆc T1. Néi dung thanh ghi @ic t'ng lãn khi cã sù chuyỐn tr'ng th,i tũ 1 vỜ 0 t'i ch©n @Çu vư ngoi T0 hoÆc T1. Xung nhp ẽ c,c @Çu vư ngoi @ic lỄy mẾu t'i thêi @iỐm S5P2 cĩa mçi chu kú mưy. Khi qu, trxnh lỄy mẾu ph,t hiỐn ra møc cao ẽ 1 chu kú vư møc thÊp ẽ chu kú tiỐp theo, thx bé @Ổm @ic t'ng lãn. Gi, trp mĩ cĩa bé @Ổm xuÊt hiỐn trong thanh ghi t'i thêi @iỐm S3P1 cĩa chu kú mưy sau khi sù chuyỐn tr'ng th,i @· @ic ph,t hiỐn. Vx vỄy @Ó néi dung cĩa thanh ghi t'ng lãn 1 @-n vư ph¶i mÊt 2 chu kú mưy, nãn tẽc @é @Ổm tòi @a lư 1/24 tCn sè bé dao @éng. Kh«ng cã sù giĩ h'p sè vßng thuc hiỐn cĩa t'yn hiỜu ẽ @Çu vư ngoi, nhng nã sĩ gi÷ Ýt nhÊt 1 chu kú mưy @Çy @ñ @Ó @¶m b¶o ch³/₄c ch³/₄n r»ng mét møc @· cho @ic lỄy mẾu Ýt nhÊt 1 lCn n÷a tríc khi nã thay @æi.

Do xung nhp bãn ngoi cã tCn sè bÊt kú nãn c,c bé Timer (0 vư 1) cã 4 chỖ @é lưm viỐc kh,c nhau @Ó lừu chãn: (13 bit Timer, 16 bit Timer, 8 bit auto-reload, split Timer).

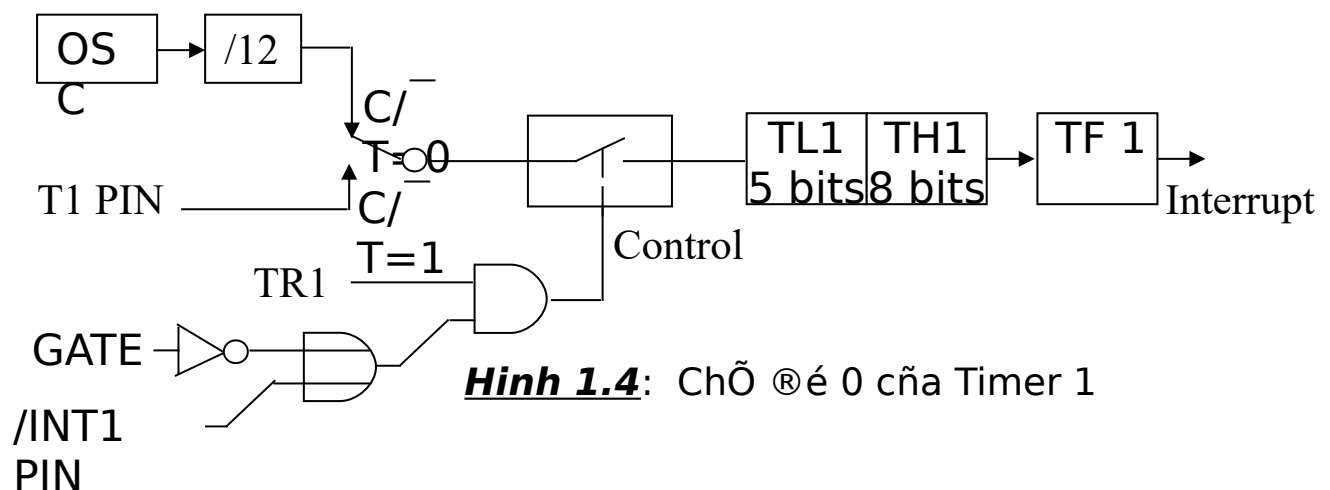
Timer 0 vư Timer 1:

Trong AT89C51 vư AT89C52 @Ờu cã c,c bé Timer 0 vư 1. Chøc n'ng Timer hay Counter @ic chãn lừa bễi c,c bit @iỜu

khiÓN C/(/T) trong thanh ghi TMOD. Hai bé Timer/Counter này cả 4 chỖ ®é ho¹t ®éng, ®íc lùa chän bëi cÆp bit (M0, M1) trong TMOD. ChỖ ®é 0, 1 vµ 2 giềng nhau cho c, c chøc n”ng Timer/Counter, nhng chỖ ®é 3 th× kh, c. Bèn chỖ ®é ho¹t ®éng ®íc m« t¶ nh sau:

+ **ChỖ ®é 0**: C¶ 2 bé Timer 0 vµ 1 ể chỖ ®é 0 cả cÆu h×nh nh mét thanh ghi 13 bit, bao gảm 8 bit cña thanh ghi THx vµ 5 bit thÊp cña TLx. 3 bit cao cña TLx kh«ng x, c ®pnh ch³/₄c ch³/₄n, nªn ®íc lµm ng–. Khi thanh ghi ®íc xo, vÒ 0, th× cê ng³/₄t thêi gian TFx ®íc thiỐt lỂp. Bé Timer/Counter ho¹t ®éng khi bit ®iÒu khiÓN TRx ®íc thiỐt lỂp (TRx=1) vµ, hoÆc Gate trong TMOD b»ng 0, hoÆc /INTx=1. NÕu ®Æt GATE=1 th× cho phÐp ®iÒu khiÓN Timer/ Counter b»ng ®êng vµo ngoµi /INTx, ®Ó dỖ dµng x, c ®pnh ®é réng xung.

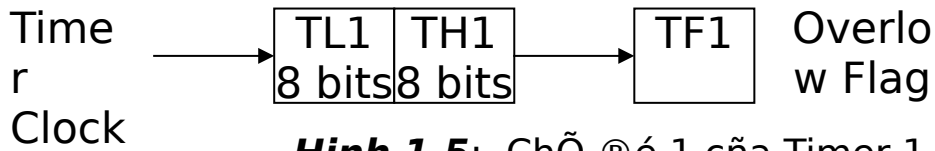
Khi ho¹t ®éng ể chøc n”ng thêi gian th× bit C/(/T)=0, do vÿy xung nhþp tổ bé dao ®éng néi, qua bé chia tỖn cho ra tỖn sê $f=f_{osc}/12$ ®íc ®a vµo ®Ó ®Ỗm trong



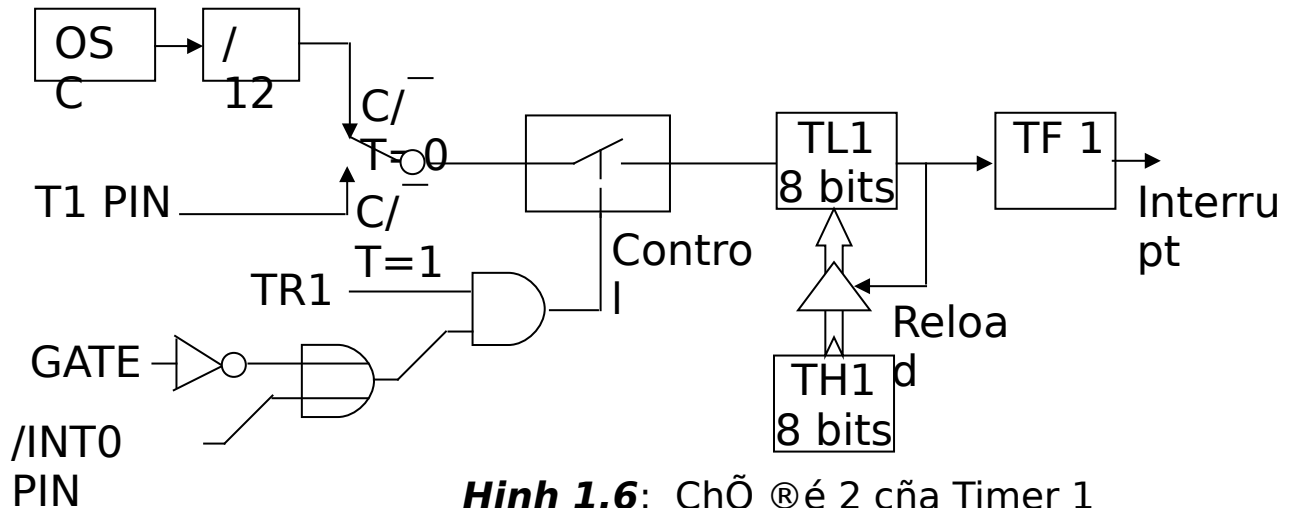
thanh ghi Timer/Counter. Khi ho¹t ®éng ể chøc n”ng bé ®Ỗm th× bit C/(/T)=1, lúc ®ã xung nhþp ngoµi ®a vµo sđ ®íc ®Ỗm.

+ **ChỖ ®é 1**: ho¹t ®éng t–ng tù nh chỖ ®é 0, chØ kh, c lµ thanh ghi Timer/Counter ®íc sđ dōng c¶ 16 bit. Xung nhþp ®íc dđng kỖt hđp vđi c, c thanh ghi thêi gian byte thÊp vµ byte cao (TH1 vµ TL1). Khi xung Clock ®íc nhËn, bé Timer sđ ®Ỗm t”ng

Đến: 0000h, 0001h, 0002,... Khi hiệN tăng trựn xEY ra, cE trựn sẽ chuyỐn FFFFh vO 0000h, vµ bE Timer tiỐp tOc Ồm. CЄ trựn cĩa Timer 1 lµ bit TF1 ẽ trong TCON, nã Ồĩc Ồắc hoÆc ghi bEi phCn mỒm, xem h×nh 2.5 (Timer/Counter 1 Mode 1: 16 bit Counter).



Hình 1.5: Chỗ ®é 1 cña Timer 1



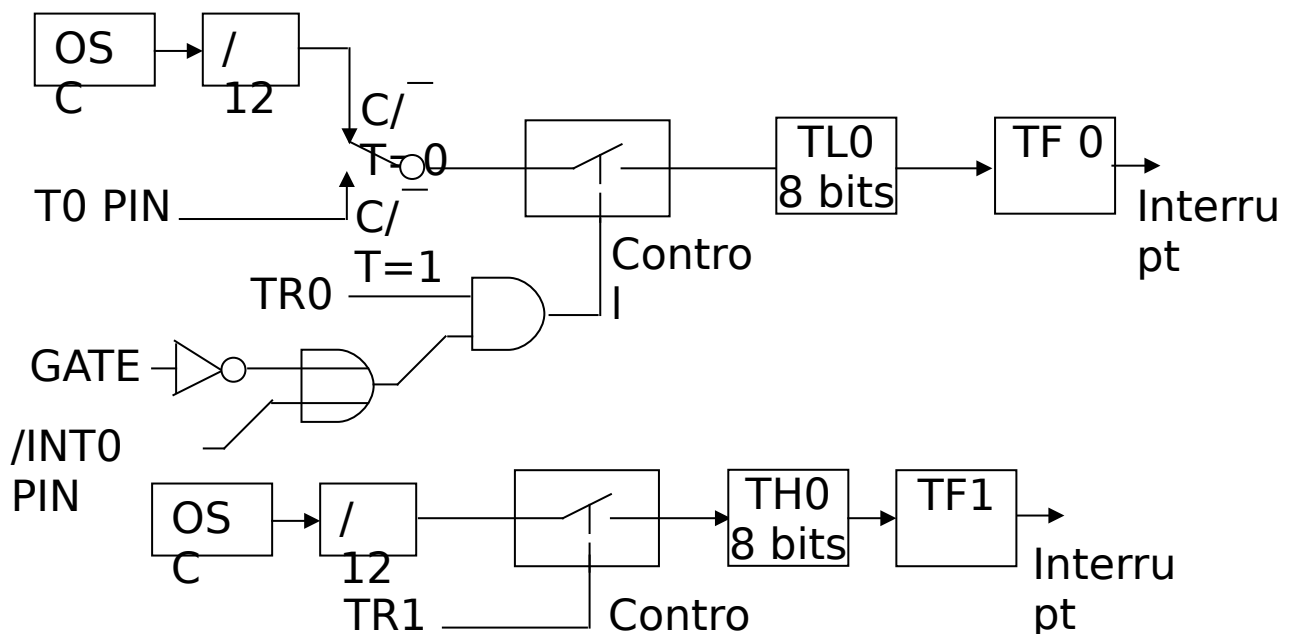
Hình 1.6: Chỗ ®é 2 cña Timer 1

+ **Chỗ 2:** Chỗ 2 này của thanh ghi Timer còn có 1 bit để chọn từ 1 trong 2 chỗ để lấy, nhưng lại có các bit để chọn 8 bit (TL1) với chỗ để chọn n1p 1i, như hình 2.6. Khi xảy ra hiện tượng tràn ở TL1, khi đó có thể thiết lập bit TF1 mà cần từ chọn n1p 1i cho TL1 bằng nội dung của TH1, để có thể thiết lập lại bằng phần mềm. Tuy nhiên n1p 1i cho phần nội dung của TH1 không thể thay đổi. Chỗ 2 của Timer/Counter 0 còn có 1 bit từ 1 trong Timer/Counter 1.

+ **ChỖ ®é 3**: ẽ chỖ ®é nựy, chۆc n`ng Timer/Counter 0 vự chۆc n`ng Timer/Counter 1 khực nhau. Bé Timer 1 ẽ chỖ ®é 3 chỖ chۆa chۆc n`ng ®Ỗm cữn ả, kỖt quự giềng khi ®Æt TR1=0. Bé Timer 0 ẽ chỖ ®é 3 thiỖt lẫp TH0, TL0 nh ỉm 2 bé ®Ỗm riẻng biỖt. Mứch Logic ®ềi vớ chỖ ®é 3 cữn Timer 0 thỖ hiỖn ẽ hựnh 2.7. Bé ®Ỗm TL0 ®ừ ®ừ ỏu khiỖn bễ cực bit:

C/(/T), GATE, TR0, /INT0 vµ khi ®Öm trµn nã thiÖt lÏp cê ng³/4t TF0. Bé ®Öm TH0 chØ ®íc ®iÖu khiÓn bëi bit TR1, vµ khi ®Öm trµn nã thiÖt lÏp cê ng³/4t TF1. VÏy, TH0 ®iÖu khiÓn ng³/4t Timer/Counter 1.

ChÖ ®é 3 thêng ®íc dïng khi yªu cÇu cÇn cã bé thêi gian hoÆc bé ®Öm ngoµi 8 bit. Ñèi vói Timer 0 ã chÖ ®é 3, AT89C51 cã thÓ cã 3 bé Timer/Counter, cßn AT89C52 cã thÓ cã 4 bé. Khi Timer 0 ho¹t ®éng ã chÖ ®é 3, th× Timer 1 cã thÓ ®íc bËt hoÆc t³/4t b»ng chuyÓn m¹ch ngoµi. ã chÖ ®é nµy, Timer 1 cã thÓ ®íc sö dông bëi cæng nèi tiÖp nh mét bé t¹o tèc ®é Baud, hoÆc trong bËt kú øng dông nµo mµ kh«ng yªu cÇu mét ng³/4t.



Hình 1.7: ChÖ ®é 3 cña Timer 0

1.5.3. Bé nhí ch¬ng tr×nh vµ bé nhí d÷ liÖu néi tró.

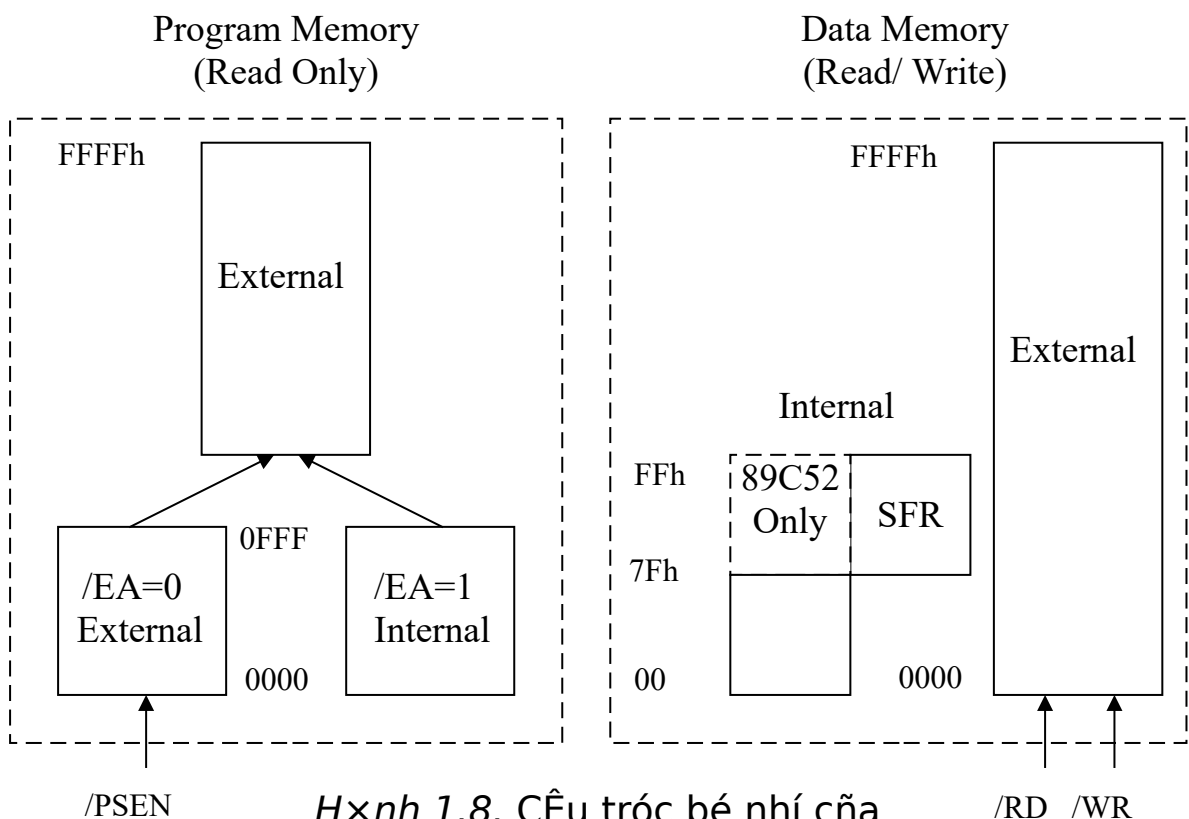
Têc c¶ c¸c bé Flash Microcontrollers cña Atmel ®Öu t¸c chøc c¸c vïng ®ãa chØ t¸ch biÖt ®èi vói bé nhí ch¬ng tr×nh vµ bé nhí

d÷ liÖu, ®íc m« t¶ ẽ h×nh dúi ®©y. C, c vïng nhí ch¬ng tr×nh vµ d÷ liÖu t, ch biÖt cho phÐp bé nhí d÷ liÖu ®íc truy cËp bëi ®Pa chØ 8 bit, cã thÓ ®íc lu tr÷ vói tèc ®é cao vµ ®íc vËn hïnh bëi mét bé CPU 8 bit. Tuy nhiªn, ®Pa chØ bé nhí d÷ liÖu 16 bit còng cã thÓ ®íc t¸o ra th«ng qua thanh ghi con trá d÷ liÖu (DPTR).

Bé nhí ch¬ng tr×nh cã thÓ chØ ®íc ®äc. Chóng cã thÓ lµ bé nhí ch¬ng tr×nh 64 Kbyte cã kh¶ n¨ng ®Þnh ®Pa chØ trùc tiÕp. §Ó ®äc néi dung tổ bé nhí ch¬ng tr×nh ngoµi, cÇn x, c ®Þnh tr¹ng th, i phï hîp cho ch©n /PSEN.

Bé nhí d÷ liÖu chiÖm 1 vïng ®Pa chØ riªng biÖt so vói bé nhí ch¬ng tr×nh. 64 Kbyte bé nhí ngoµi cã thÓ ®íc ®Þnh ®Pa chØ trùc tiÕp trong vïng bé nhí d÷ liÖu ngoµi. CPU t¸o ra tÝn hiÖu ®äc vµ ghi (/RD, /WR) ®Ó truy cËp bé nhí d÷ liÖu ngoµi.

Bé nhí ch¬ng tr×nh ngoµi vµ bé nhí d÷ liÖu ngoµi cã thÓ ®íc kÖt hîp bëi c, c tÝn hiÖu /RD vµ /PSEN ®Ó ®a vµo 1 cæng AND vµ s¸o d¸ng ®Çu ra cña cæng nµy ®Ó ®äc néi dung tổ bé nhí d÷ liÖu/ch¬ng tr×nh ngoµi.



H×nh 1.8. Cu tróc bé nhí cña AT89C51

1.5.3.1. Bé nhí ch–ng tr×nh néi tró.

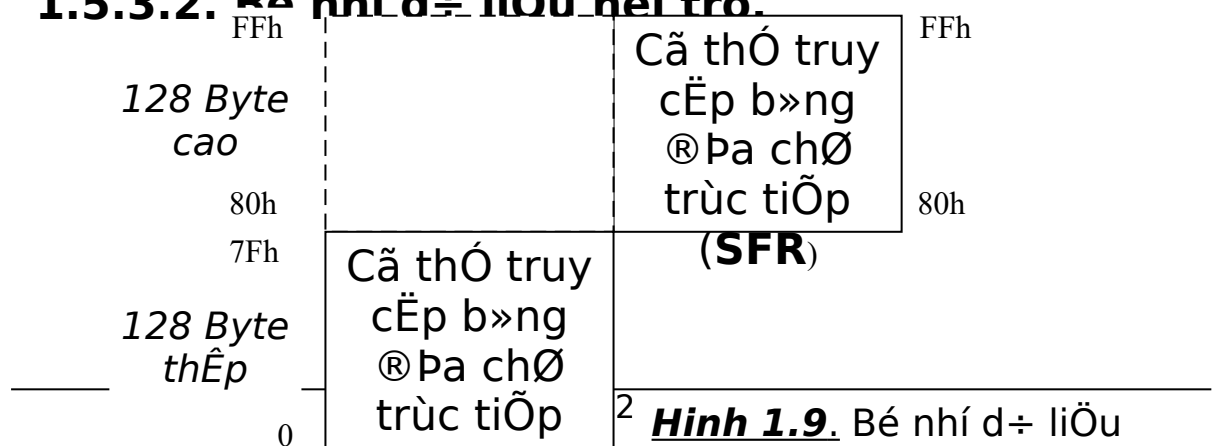
Bé nhí ch–ng tr×nh của AT89C51 ®íc tæ chøc nh thÓ hiÖn ẽ h×nh trªn. Kh«ng gian nhí cùc ®¹i của bé nhí nuy chiÖm 64 Kbyte, ®íc ®Þnh ®Þa chØ tã 0000h ®Õn FFFFh, trong ®ã cã 4 Kbyte Flash néi tró bªn trong nã vµ ®íc ®Þnh ®Þa chØ tã 0000h ®Õn 0FFFh. Do ®ã cã thÓ mẽ réng thªm 60 Kbyte bé nhí ch–ng tr×nh bªn ngoµi, ®íc ®Þnh ®Þa chØ tã 1000h ®Õn FFFFh. Tuy nhiªn bé VŞK nuy còng cã thÓ s dung toµn bé bé nhí ch–ng tr×nh ngoµi bao gm 64 Kbyte ®íc ®Þnh ®Þa chØ tã 0000h ®Õn FFFFh.

Còng tã h×nh trªn ta thÊy, th«ng qua viÖc chn mc logic cho bit /EA cã thÓ lµa chn ®Ó truy cËp bé nhí ch–ng tr×nh néi tró (4Kb), bé nhí ch–ng tr×nh mẽ réng ngo¹i tró (60Kb), hoÆc toµn bé bé nhí ch–ng tr×nh ngo¹i tró bªn ngoµi On-chip (64Kb). C thÓ, khi /EA = 1 th× bé VŞK s ®ång c¶ bé nhí ch–ng tr×nh néi tró vµ ngo¹i tró. Ngc l¹i, khi /EA = 0 th× bé VŞK chØ s ®ång bé nhí ch–ng tr×nh ngo¹i tró.

Mçi khi ®íc Reset, bé VŞK s truy cËp bé nhí ch–ng tr×nh t¹i ®Þa chØ khêi ®Çu lµ 0000h, sau ®ã nÕu c chØ ng³⁄⁄t ®íc s ®ång th× nã s truy cËp t¹i ®Þa chØ quy ®Þnh trong b¶ng vector ng³⁄⁄t.

Khi truy cËp bé nhí ch–ng tr×nh, bé VŞK s ®ång xung chn /PSEN ®Ó ®iÒu khiÓn. NÕu on-chip lµm viÖc vi bé nhí ch–ng tr×nh néi tró th× chn pht ra xung chn /PSEN kh«ng s ®ång. NÕu bé VŞK lµm viÖc vi bé nhí ch–ng tr×nh ngo¹i tró th× chn pht ra xung chn /PSEN ®íc s ®ång. Khi ®ã nÕu /PSEN = 0 th× cho phÐp bé VŞK ®c bé nhí ch–ng tr×nh ngoµi, ngc l¹i nÕu /PSEN = 1 th× bé VŞK chØ lµm viÖc vi bé nhí ch–ng tr×nh néi tró.

1.5.3.2. Bé nhí d÷ liÖu néi tró.



Hình 1.9. Bé nhí d÷ liÖu trong

AT89C51 cũng có bộ nhớ dữ liệu chi tiết một khoảng không gian bộ nhớ đặc biệt với bộ nhớ chương trình. Dung lượng của RAM nội bộ là 128 Byte, địa chỉ bộ nhớ từ 00h đến 7Fh. Phần vi địa chỉ từ 80h đến FFh dành cho SFR. Tuy nhiên bộ nhớ cũng có thể làm việc với RAM ngoài bộ nhớ cục bộ 64 Kbyte địa chỉ bộ nhớ từ 0000h đến FFFFh.

1.5.3.2.1. Vùng nhớ 128 Byte thƣp

Vùng nhớ 128 Byte thƣp địa chỉ bộ nhớ từ 00h đến 7Fh, địa chỉ chia thành 3 vùng con như thể hiện ở hình 2.10.

- Vùng nhớ đầu tiên có địa chỉ 32 Byte địa chỉ bộ nhớ từ 00h đến 1Fh bao gồm 4 bảng thanh ghi (bảng 0...bảng 3), mỗi bảng có 8 thanh ghi 8 bit. Các thanh ghi trong mỗi bảng cần giải mã từ R0 đến R7. Vùng RAM này địa chỉ truy cập bằng địa chỉ trực tiếp mặc định Byte, và quá trình chuyển đổi số dung bảng thanh ghi vào bộ đếm thuộc vào việc lựa chọn giá trị cho RS1 và RS0 trong PSW.

- Vùng nhớ 2 có địa chỉ 16 Byte địa chỉ bộ nhớ từ 20h đến 2Fh, cho phép truy cập trực tiếp bằng địa chỉ mặc định bit. Bộ nhớ cũng cấp các lệnh cần thiết để truy cập tới vùng nhớ 128 bit này (nếu truy cập ở dạng mặc định thì vùng này có địa chỉ địa chỉ bộ nhớ từ 00h đến 7Fh) ở mặc định. Ở vùng nhớ này, địa chỉ địa chỉ truy cập dưới dạng Byte hay Bit tùy vào lệnh cần thực hiện. Chẳng hạn, để đặt bit tại địa chỉ 5Fh về mặc định logic 1, ta thực hiện lệnh: **SETB 5Fh**. Sau khi thực hiện lệnh này, mặc định địa chỉ 5Fh là địa chỉ bit cao nhất trong Byte có địa chỉ 2Bh, nhưng sẽ không làm ảnh hưởng tới các bit khác trong Byte này. Trong khi đó, ở bộ nhớ VXL để thực hiện chức năng như trên cần dùng lệnh sau:

```
MOV    A,2Bh
ORL A,#10000000b
MOV    2Bh,A
```

§©y lụ u ®iÓm rầ nĐt cĩa c, c bé V§K khi thùc hiÖn viÖc truy xuÊt c, c bit riêng rĩ th«ng qua phÇn mÒm. C, c bit cã thÓ ®ù-c ®Æt, xo, , hay thùc hiÖn chøc n"ng AND, OR...chØ th«ng qua 1 lÖnh. Ngoµi ra c, c cæng xuÊt/nhËp còng cã thÓ ®íc ®Þnh ®Þa chØ d"ng bit, ®iÖu nşy lụm ®-n gi¶n viÖc giao tiÖp b»ng phÇn mÒm vớ c, c thiÖt bÞ xuÊt/nhËp ®-n bit.

- Vĩng nhí cßn l'i gãm 80 Byte cã ®Þa chØ tũ 30h ®Õn 7Fh ®íc dşnh riêng cho ngêi sũ dõng ®Ó lu tr÷ d÷ liÖu. §©y cã thÓ xem lụ vĩng RAM ®a míc ®Ých. Cã thÓ truy cËp vĩng nhí nşy b»ng ®Þa chØ trùc tiÖp hoÆc gi, n tiÖp th«ng qua c, c thanh ghi (R0 hoÆc R1) ẽ d"ng míc Byte.

1.5.3.2.2. Vĩng nhí dşnh cho SFR

Vĩng nhí nşy ®íc ®Þnh ®Þa chØ tũ 80h ®Õn FFh, vµ ®íc truy cËp b»ng ®Þa chØ trùc tiÖp.

1.5.3.2.3. C, c lÖnh truy cËp bé nhí d÷ liÖu néi tró.

- MOV A, <src>: ChuyÓn d÷ liÖu tũ to, n h"ng nguån (c, c « nhí, thanh ghi cã ®Þa chØ trùc tiÖp hoÆc gi, n tiÖp trong on chip, c, c gi, trÞ trùc h»ng chøa trong c©u lÖnh) vµo thanh ghi tÝch luũ.

- MOV <dest>, <src>: ChuyÓn d÷ liÖu tũ to, n h"ng nguån vµo to, n h"ng ®Ých (c, c « nhí, thanh ghi cã ®Þa chØ trùc tiÖp hoÆc gi, n tiÖp trong on chip).

- MOV <dest>, A : ChuyÓn d÷ liÖu tũ A vµo to, n h"ng ®Ých.

- MOV DPTR, #data16: ChuyÓn gi, trÞ h»ng 16 bit vµo thanh ghi con trá d÷ liÖu.

- PUSH <src>: ChuyÓn gi, trÞ tũ to, n h"ng nguån vµo ng"n xÕp.

- POP <dest>: ChuyÓn gi, trÞ tũ ng"n xÕp vµo to, n h"ng ®Ých.

- XCH A, <byte>: ChuyÓn ®æi d÷ liÖu gi÷a to, n h"ng nguån d"ng byte vớ A.

- XCHD A, @Ri: ChuyÓn ®æi nũa thÊp cĩa A vớ néi dung trong RAM t'i ®Þa chØ lụ néi dung cĩa Ri.

Byte Address	Bit Address
7F	General purpose RAM
30	
2F	
2E	
2D	
2C	
2B	
2A	
29	
28	
27	
26	
25	
24	
23	
22	
21	
20	
1F	Bank 3
1E	
1D	Bank 2
1C	
1B	Bank 1
1A	
0F	Default register bank for R0-R7
0E	
0D	
0C	
0B	
0A	
09	
08	
07	
06	
05	
04	
03	
02	
01	
00	

H×nh 1.10. 128 Byte thÊp cña RAM trong

Byte address**Bit address**

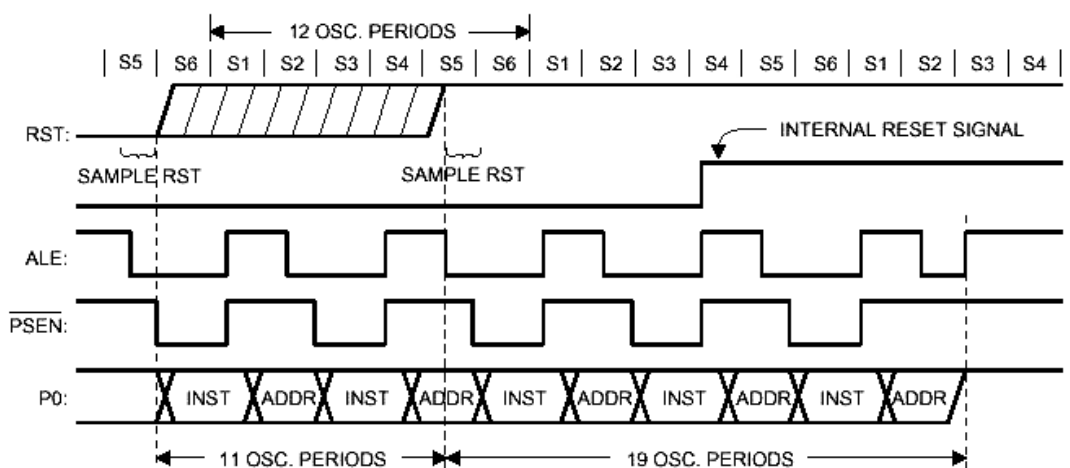
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	-	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	Not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	Not bit addressable								TH1
8C	Not bit addressable								TH0
8B	Not bit addressable								TL1
8A	Not bit addressable								TL0
89	Not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	Not bit addressable								PCON
83	Not bit addressable								DPH
82	Not bit addressable								DPL
81	Not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0

H×nh 1.11. Các thanh ghi chức năng
 ®Æc biÖt (SFR)

1.5.4. Nguyên lý khôi phục của On-chip AT89C51:

RST là chức năng Reset, chỉnh là bộ vào của Trigger Schmitt. Việc Reset thực hiện bằng cách giữ chức năng RST ở mức cao ít nhất 2 chu kỳ máy (24 chu kỳ dao động), trong khi bộ dao động đang làm việc. Khi đã on-chip sẽ thực hiện khôi phục lại (nhập reset bên trong này) theo biểu đồ thời gian (Hình 2.23. Reset Timing).

Figure 2.23. Reset Timing



Hình 1.12. Sơ đồ thời gian cho AT89C51

Tín hiệu khôi phục lại bên ngoài đưa vào chức năng RST không ảnh hưởng với xung Clock bên trong. Chức năng RST thực hiện mẫu tại thời điểm P2S5 của mỗi chu kỳ máy. Các chức năng của cổng sẽ giữ hoạt động hiện hành của chúng cho 19 chu kỳ dao động sau khi giữ trạng thái logic 1. Thực hiện mẫu ở chức năng RST.

Trong khi chức năng RST ở mức cao, ALE và PSEN thực hiện ở mức cao. Sau khi RST thực hiện xong, sẽ giữ 1 rồi 2 chu kỳ máy rồi với ALE và PSEN ở mức thấp khôi phục xung Clock. Vì lý do

Khi Reset cả hiÖu lùc th× gi_ trÞ cña c_ç SFR, ®íc m« t¶ ẽ (Table -Reset Values of SFRs). Trong ®ã thanh ghi SBUF, vµ mét sè bit cña PCON, T2MOD, IE, IP cũ gi_ trÞ bÊt ®Þnh. C_ç chèt cæng tã P0...P3 cũ gi_ trÞ FFh, SP cũ gi_ trÞ 07h. C_ç thanh ghi cßn l¹ cũ gi_ trÞ 00h.

Khěi ®éng l¹i cho On-chip cã thÓ ho¹t ®éng ẽ tr¹ng th₁i từ ®éng hoÆc b₁n từ ®éng .

AT89C51.

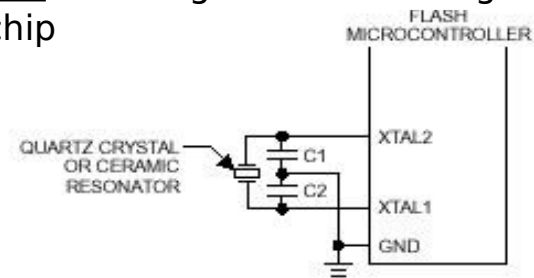
Hình 1.14. Khởi động bộ nhớ cho AT89C51.

Khởi động bộ nhớ cũng thực hiện ra khi cấp nguồn điện +Vcc cho on-chip bằng mạch điện RC. Sau khi cấp nguồn, mạch RC giới hạn cho chân RST về trạng thái cao trong thời gian đủ để nạp hàng sẽ thời gian của mạch RC. Số điện trở bằng khởi động điện on-chip, thời gian chân RST về trạng thái cao phải nhỏ hơn (Không lớn hơn 2 chu kỳ máy) để mạch dao động chuyển sang trạng thái dao động bình thường.

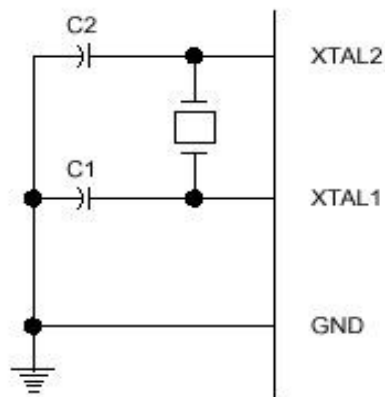
Nếu khởi động bộ nhớ, sau khi cấp nguồn mạch sẽ từ động Reset như khởi động bộ nhớ. Khi chân khởi động là phải nhận cùng tần số đồng bộ K, thời gian nhận cùng tần số chân RST phải về trạng thái cao.

1.5.5. Mạch dao động.

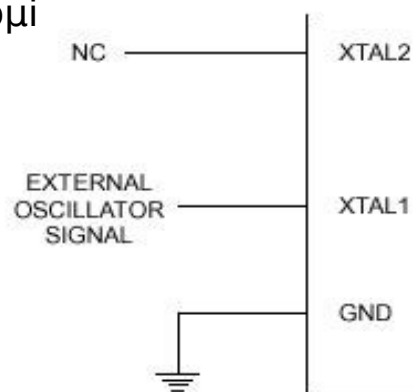
H×nh 2.15. Sơ đồ m¹ch dao ®éng trªn On chip



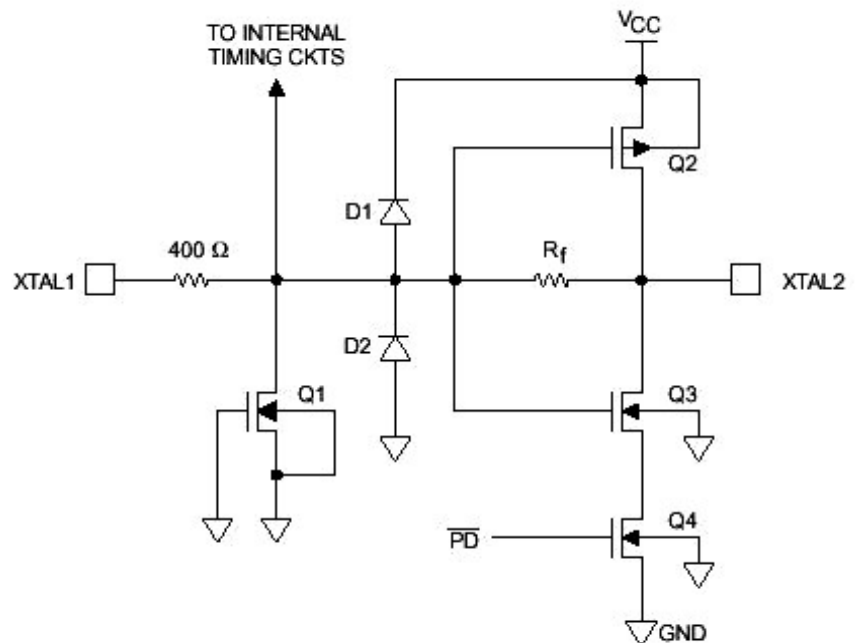
H×nh 2.16. Kĩt n÷i m¹ch dao ®éng



H×nh 1.17. Cúu h×nh nh÷n Clock tã bªn ngoµi



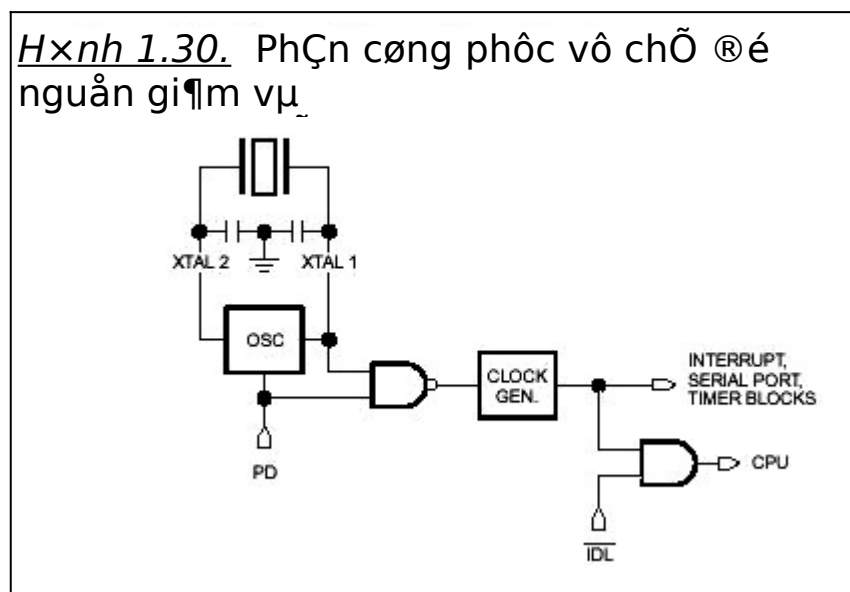
H×nh 1.18. M¹ch dao ®éng bñn trong On chip AT89C51



Note: In Atmel's CMOS microcontrollers the Oscillator Specification differs from that in NMOS versions.

1.5.6. Chỗ ®é nguån gi¶m vµ chỗ ®é nghØ

H×nh 1.30. PhÇn cøng phøc vø chỗ ®é nguån gi¶m vµ



1.5.7. Bảng vô hướng trục nh.

Hãng V&K	Các bit kho,
AT89C51	LB1, LB2, LB3
AT89C52	LB1, LB2, LB3
AT89C2051	LB1, LB2
AT89C1051	LB1, LB2

Kho, bé nhí ch→ng tr×nh cho hãng V&K AT89C51:

Chỗ ®é	LB1	LB2	LB3	Lo¹i bảng vô
1	U	U	U	Kh«ng cã ®Æc trng kho, ch→ng tr×nh.
2	P	U	U	Các l�nh MOVC ®�c th�c thi t� b� nh� ch→ng tr×nh ngo�i, kh«ng ®�c ph�p t�m n�p l�nh t� b� nh� n�i. EA ®�c l�y m�u v� ch�t khi reset. Vi�c l�p tr×nh tr�n Flash b� c�m.
3	P	P	U	Nh chỗ ®� 2, ngo�i ra c�n c�m vi�c ki�m tra ch→ng tr×nh.
4	P	P	P	Nh chỗ ®� 3, ngo�i ra c�n c�m vi�c th�c thi ch→ng tr×nh ngo�i.

Lưu ý: P=Programmed, U = Unprogrammed

Chương 2

T p l nh c n hãng V&K AT89/80C51

B  V&K c  t p l nh ® c t i u ho, ®   ng d ng trong các h  th ng ® i u khi n, ®  l ng 8 bit. Số t ng kh  n ng truy xu t RAM n i tr n các d  li u nh , các ki u ® nh ® p  ch  ® c bi t ®  ® c  p d ng. Ngo i ra t p l nh c n V&K c n h  tr  các bi n 1 bit, cho ph p qu n lý bit tr c ti p trong các h  logic v  ® i u khi n bit c  y u c u x  lý bit. Do hãng V&K AT89/80C51 c  các m  l nh 8 bit, n n s  l nh c  th  l n ® n 256 l nh (th c t  c  255 l nh, c n 1 l nh ch  ® c ® nh ngh a). Trong ®  c  139 l nh 1

byte, 92 l  nh 2 byte v   24 l  nh 3 byte. M  i l  nh                   tr  ng b  i m   l  nh ($m \cdot m_y$), m   g  i nh  , s   byte c  a l  nh v   s   chu k   m_y c  n       th  c thi l  nh. C  c l  nh c  a AT89/80C51       chia th  nh 5 nh  m l  nh:

- Nh  m l  nh di chuy  n d   li  u.
- Nh  m l  nh s   h  c.
- Nh  m l  nh logic.
- Nh  m l  nh r   nh  nh ch    ng tr  nh.
- Nh  m l  nh   i  u khi  n bi  n logic.

C  c quy   c trong c  u l  nh v       a ch  :

- + Rn: Thanh ghi R0-R7 c  a b  ng thanh ghi hi  n h  nh     ng       ch  n                 a ch   thanh ghi.
- + Direct:     a ch   8 bit c  a    nh   d   li  u n  i tr  , n   c   th    l      nh   trong RAM n  i ho  c SFR. (00h-FFh)
- + @Ri:    nh   8 bit c  a RAM n  i                 a ch   gi  n ti  p th    ng qua thanh ghi R0 h  c R1.
- + Source (Src): to  n h  ng ngu  n, c   th    l   Rn ho  c direct ho  c @Ri.
- + Dest: To  n h  ng      , c   th    l   Rn ho  c direct ho  c @Ri.
- + #Data: H  ng s   8 bit ch  a trong l  nh.
- + #Data16: H  ng s   16 bit ch  a trong l  nh.
- + Bit: Bit                 a ch   tr  c ti  p trong RAM n  i tr   ho  c SFR.
- + Rel: Offset 8 bit c   d  u (t   -128       +127). N         l  nh SJMP v   c  c l  nh nh  y c     i  u ki  n s   d  ng.
- + Addr11:     a ch   11 bit c  a b   nh   ch    ng tr  nh ,       l  nh ACALL v   AJMP s   d  ng.
- + Addr16:     a ch   16 bit c  a 64Kb b   nh   ch    ng tr  nh,       l  nh LCALL v   LJMP s   d  ng.

C  c k   hi  u d  ng trong m   t   l  nh:

Ký hiÖu	ý nghĩa
<-	§íc thay thÖ bëi...
()	Néi dung cña...
(())	D÷ liÖu ®íc trá bëi...
rrr	1 trong 8 thanh ghi (R0-R7) cña c, c bëng thanh ghi
dddddddd	C, c bit d÷ liÖu
aaaaaaaa	C, c bit ®pa chØ
bbbbbbbb	®pa chØ cña 1 bit
i	§pnh ®pa chØ gi, n tiÖp th«ng qua R0 hoÆc R1
eeeeeeee	§pa chØ t¬ng ®èi 8 bit

2.1. Nhãm lÖnh di chuyÖn d÷ liÖu

2.1.1. LÖnh MOV d¹ng Byte:

Có ph, p c©u lÖnh: **MOV <dest-byte>, <src-byte>**

Chøc năng: Sao chép néi dung cña to, n h¹ng nguån vµo to, n h¹ng ®Ých, néi dung cña to, n h¹ng nguån kh«ng thay ®æi. LÖnh này kh«ng lµm ¶nh hëng tí c, c cê vµ c, c thanh ghi kh, c.

C©u lÖnh	Sè by te	Sè chu kú	M- lÖnh	Ho¹t ®éng
MOV A, Rn	1	1	11101rrr	(A)<-(Rn)
MOV A, direct	2	1	11100101 aaaaaaaa	(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 aaaaaaaa	(Rn)<-(direct)
MOV Rn, #data	2	1	01111rrr dddddddd	(Rn)<-#data
MOV direct, A	2	1	11110101 aaaaaaaa	(direct)<-(A)

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
MOV direct, Rn	2	2	10001rrr aaaaaaaa	(direct)<-(Rn)
MOV direct, direct	3	2	10000101 aaaaaaaa aaaaaaaa	(direct)<-(direct)
MOV direct, @Ri	2	2	1000011i aaaaaaaa	(direct)<-((Ri))
MOV direct, #data	3	2	01110101 aaaaaaaa 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

2.1.2. LÖnh MOV d¹ng Bit:

Có ph, p c©u lÖnh: **MOV <dest-bit>, <scr-bit>**

Chøc n"ng: ChuyÖn bit d÷ liÖu ẽ d¹ng sao chÐp to, n h¹ng nguån vao to, n h¹ng ®Ých. Mét trong 2 to, n h¹ng ph¶i lµ cê nhí (C), to, n h¹ng cßn l¹i sñ lµ bit bÊt kú ®íc ®Þnh ®Þa chØ tróc tiÕp. LÖnh kh«ng lµm ¶nh hëng tíi c, c thanh ghi kh, c hoÆc c, c cê kh, c.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
MOV C, bit	2	1	10100010 bbbbbbbb	(C)<-(bit)
MOV bit, C	2	2	10010010 bbbbbbbb	(bit)<-(C)

2.1.3. LÖnh MOV d¹ng Word:

Có ph, p c©u lÖnh: MOV DPTR, #data16

Chøc n"ng: Gi, trÞ 16 bit ë to, n h"ng thø 2 trùc tiÖp trong c©u lÖnh ®íc n"p vµo thanh ghi DPTR. H»ng sè 16 bit nµy ®íc ®Æt ë byte 2 vµ byte 3 cña lÖnh. Byte 2 lµ byte cao ®íc n"p cho thanh ghi DPH, byte 3 lµ byte thÊp ®íc n"p vµo thanh ghi DPL. LÖnh nµy kh«ng ¶nh hëng tí c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
MOV DPTR, #data16	3	2	10010000 dddddddd ddddddd	(C)<- (bit)

2.1.4. LÖnh chuyÖn byte m· lÖnh:

Có ph, p c©u lÖnh: MOVC A, @A + <thanh ghi c- sè>

Chøc n"ng: N"p cho thanh ghi tÝch luü byte m· lÖnh tã bé nhí ch- ng tr×nh. §¸a chØ cña byte ®íc t×m n"p trong bé nhí lµ tæng néi dung cña thanh ghi A 8 bit vói néi dung cña thanh ghi c- sè 16 bit (cã thÓ lµ DPTR hoÆc PC - thanh ghi ®Öm ch- ng tr×nh). Trong trêng híp sau, PC ®íc t"ng ®Ó trá ®Ön ®¸a chØ cña lÖnh tiÖp theo ((PC)<-(PC+1)) tríc khi ®íc c«ng vói néi dung cña thanh ghi A, cßn thanh ghi DPTR kh«ng bÞ thay ®æi. LÖnh kh«ng ¶nh hëng tí c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
MOVC A, @A+DPTR	1	2	10010011	(A)<-((A)+ (DPTR))
MOVC A, @A+PC	1	2	10000011	(A)<-((A)+(PC))

2.1.5. LÖnh chuyÖn d÷ liÖu ra ngoµi:

Có ph, p c©u lÖnh: MOVX <dest-byte>, <src-byte>

2.1.7. Lệnh chuyển sẽ liệu ra khỏi ng"n x"p:

Có ph, p c"u l"nh: **POP direct**

Ch"c n"ng: Chuyển néi dung c"n ng"n x"p ề RAM trong, c"n "p" ch" "i" SP trá t"i "i"n n"i c"n "p" ch" tr"u" ti"p trong c"u l"nh. Sau "đ", con trá ng"n x"p (SP) "i" gi"m "i 1. L"nh kh"ng "nh h"ng t"i c, c c".

C"u l"nh	S" byt e	S" chu kú	M- l"nh	Ho"t "éng
POP direct	2	2	11010000 aaaaaaaa	(direct) <-- ((SP)) (SP) <-- (SP-1)

2.1.8. H"n chuyển d" liệu:

Có ph, p c"u l"nh: **XCH A, <byte>**

Ch"c n"ng: H"n chuyển néi dung gi"i"a thanh ghi A v"i thanh ghi ho"c bé nh"i c"n "p" ch" ch"i" trong to, n h"ng th" 2 c"n c"u l"nh. To, n h"ng th" 2 c"n th" "i" "p"nh "p" ch" ki"u thanh ghi, thanh ghi tr"u" ti"p ho"c thanh ghi gi, n ti"p.

C"u l"nh	S" by te	S" chu kú	M- l"nh	Ho"t "éng
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))

2.1.9. H"n chuyển 4 bit th"p:

Có ph, p c"u l"nh: **XCHD A, @Ri**

Ch"c n"ng: H"n chuyển 4 bit th"p néi dung trong thanh ghi A v"i « nh"i c"n RAM b"n trong, c"n "p" ch" "i" "p"nh gi, n ti"p qua thanh ghi "i" ch" ra trong l"nh. L"nh n"y kh"ng "nh h"ng t"i tr"ng th, i c, c c" v"i n"i" cao c"n c, c thanh ghi trong l"nh.

C©u lÖnh	Sè by te	Sè chu kú	M- lÖnh	Ho¹t ®éng
XCHD A, @Ri	1	1	110101 1i	(A3-A0)<-->((Ri3-Ri0))

2.2. Nhãm lÖnh tÝnh to,n sè häc.

2.2.1. LÖnh thùc hiÖn phÐp céng.

Có ph,p cña c©u lÖnh: ADD A, <scr-byte>

Chøc năng: Céng gi, trÞ 1 byte ë ®Þa chØ ®íc chØ ra ë c©u lÖnh víi néi dung trong thanh ghi tÝch luÛ, kÖt qu¶ ®íc lu vµo thanh ghi tÝch luÛ. NÕu cã nhÞ tö bit sè 7 hoÆc bit sè 3 th× cê nhÞ hoÆc cê nhÞ phô ®íc thiÖt lÖp, ngíc l¹i c,c cê nµu trªn ®íc xo,. Khi céng 2 sè nguyªn kh«ng ®éu mµ bÞ trËn th× cê nhÞ còng ®íc thiÖt lÖp ®Ó cho ta biÖt phÐp to,n bÞ trËn. Trêng hîp thùc hiÖn lÖnh ADD mµ cã nhÞ tö bit sè 6 nhng kh«ng cã nhÞ tö bit sè 7, hoÆc cã nhÞ tö bit sè 7 nhng kh«ng cã nhÞ tö bit sè 6 th× cê trËn sÏ ®íc thiÖt lÖp, ngíc l¹i th× OV bÞ xo,. Khi céng 2 sè nguyªn cã ®éu mµ tæng lµ 1 sè ©m th× OV ®íc thiÖt lÖp.

C©u lÖnh	Sè byt e	Sè chu kú	M- lÖnh	Ho¹t ®éng
ADD A, Rn	1	1	00101rrr	(A)<- (A) + (Rn)
ADD A, direct	2	1	00100101 aaaaaaaa	(A)<- (A) + (direct)
ADD A, @Ri	1	1	0010011i	(A)<- (A) + ((Ri))
ADD A, #data	2	1	00100100 dddddddd	(A)<- (A) + #data

2.2.2. LÖnh céng cã nhÞ.

Có ph,p cña c©u lÖnh: ADDC A, <scr-byte>

Chức năng: Cúng ®ăng thêi néi dung cña 1 byte ẽ ®pa chø ®íc chø ra trong c©u lÏnh vớ néi dung chøa trong thanh ghi tÝch luũ vµ cê nhí. NÕu cã nhí tã bit sè 7 hoÆc sè 3 th× cê nhí hoÆc cê nhí phô ®íc thiÕt lÏp b»ng 1, ngíc l¹i c,c cê nªu trªn bÞ xo_. Khi cúng c,c sè nguyªn kh«ng dÊu mụ bÞ trun th× cê nhí còng ®íc thiÕt lÏp. Trêng hîp thùc hiÕn lÏnh ADDC mụ cã nhí tã bit sè 6 nhng kh«ng nhí tã bit sè 7, hoÆc cã nhí tã bit sã 7 nhng kh«ng nhí tã bit sè 6 th× cê trun sã ®íc thiÕt lÏp, ngíc l¹i cê nuy bÞ xo_. Khi cúng c,c sè nguyªn cã dÊu mụ tæng lụ 1 sè ©m th× OV ®íc thiÕt lÏp.

C©u lÏnh	Sè byte	Sè chu kú	M-lÏnh	Ho¹t ®éng
ADDC A, Rn	1	1	00110rr r	(A) <- (A) + (C) + (Rn)
ADDC A, direct	2	1	001101 01 aaaaaaa a	(A) <- (A) + (C) + (direct)
ADDC A, @Ri	1	1	001101 1i	(A) <- (A) + (C) + ((Ri))
ADDC A, #data	2	1	001101 00 dddddd dd	(A) <- (A) + (C) + #data

2.2.3. LÏnh trã cã mîn.

Có ph,p cña c©u lÏnh: **SUBB A, <scr-byte>**

Chức năng: Trã thanh ghi tÝch luũ cho to,n h¹ng thø 2 vµ cê nhí, kÕt qu¶ ®íc lu vµo thanh ghi tÝch luũ. Cê nhí ®íc ®Æt b»ng 1 nÕu cã sè mîn ®íc cÇn ®Õn cho bit sè 7, ngíc l¹i th× cê nhí bÞ xo_. Cê nhí phô ®íc thiÕt lÏp nÕu cã nhí cho bit 3. Trêng hîp thùc hiÕn lÏnh SUBB mụ cã sè mîn ®íc cÇn ®Õn cho bit 7(kh«ng ph¶i cho bit 6), hoÆc cho bit 6 (kh«ng ph¶i cho bit 7) th× cê trun sã

®íc thiÖt lĒp, ngíc lĒi th× OV bÞ xoĴ. Khi trÖ cĴc sē nguyēn cā dĒu mÞ kÖt quĒ lÞ 1 sē ©m th× OV ®íc thiÖt lĒp.

C©u lÖnh	Sē byte	Sē chu kú	M· lÖnh	Ho¹t ®éng
SUBB A, Rn	1	1	10011rrr	(A)<- (A) - (C) - (Rn)
SUBB A, direct	2	1	1001010 1 aaaaaaaa	(A)<- (A) - (C) - (direct)
SUBB A, @Ri	1	1	1001011i	(A)<- (A) - (C) - ((Ri))
SUBB A, #data	2	1	1001010 0 ddddddd d	(A)<- (A) - (C) - #data

2.2.4. LÖnh tĒng lĒn 1 ®Ēn vÞ.

Có phÞp cĒa c©u lÖnh: **INC <byte>**

Chøc nĒng: TĒng giĴ trÞ cĒa byte trong c©u lÖnh lĒn 1 ®Ēn vÞ. NÖu giĴ trÞ ban ®Çu cĒa byte lÞ 0FFh, th× sau khi thùc hiÖn lÖnh INC néi dung cĒa byte sĒ lÞ 00h. LÖnh nÞy kh«ng lÞm ¶nh hĒng tí trĒng thĴi cĴc cē.

C©u lÖnh	Sē byte	Sē chu kú	M· lÖnh	Ho¹t ®éng
INC A	1	1	0000010 0	(A)<- (A) + 1
INC Rn	1	1	00001rrr	(Rn)<- (Rn) + 1
INC direct	2	1	0000010 1 aaaaaaaa	(direct)<- (direct) + 1
INC @Ri	1	1	0000011i	((Ri))<- ((Ri)) + 1

2.2.5. LÖnh gi¶m 1 ®Ēn vÞ.

Có phÞp cĒa c©u lÖnh: **DEC <byte>**

Chức năng: Giảm giá trị của byte trong bộ nhớ 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 cờ.

Câu lệnh	Số byte	Số chu kỳ	Mệnh	Hoạt động
DEC A	1	1	0001010 0	(A) <- (A) - 1
DEC Rn	1	1	00011rrr	(Rn) <- (Rn) - 1
DEC direct	2	1	0001010 1 aaaaaaaa	(direct) <- (direct) - 1
DEC @Ri	1	1	0001011i	((Ri)) <- ((Ri)) - 1

2.2.6. Lệnh tăng con trỏ di chuyển.

Có phép của câu lệnh: INC DPTR

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

Câu lệnh	Số byte	Số chu kỳ	Mệnh	Hoạt động
INC DPTR	1	2	1010001 1	(DPTR) <- (DPTR) + 1

2.2.7. Lệnh thực hiện phép nhân.

Có phép của câu lệnh: MULAB

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, các byte cao được lưu trong thanh ghi B. Nếu kết quả lớn hơn 0FFh thì cờ tràn được thiết lập, cờ nhớ được xóa.

C^ou l^onh	S^e byte	S^e chu kú	M- l^onh	Ho^t ®éng
MUL AB	1	4	101001 00	(B) <- byte cao của (A)x(B) (A) <- byte thấp của (A)x(B)

2.2.8. L^onh th^uc hi^on ph^op chia.

Có ph^op của c^ou l^onh: **DIV AB**

Ch^oc n^ong: Chia s^e nguy^an kh^ong d^uu 8 bit trong thanh ghi t^ych lu^u cho s^e nguy^an kh^ong d^uu 8 bit trong thanh ghi B. Th^ung s^e ®ⁱc lu trong thanh ghi t^ych lu^u, c^on s^e d^u ®ⁱc lu trong thanh ghi B. C^e tr^un v^u c^e nhⁱ b^o xo^o.

C^ou l^onh	S^e byte	S^e chu kú	M- l^onh	Ho^t ®éng
DIV AB	1	4	100001 00	(A) <- th ^u ng của (A)/(B) (B) <- s ^e d ^u của (A)/(B)

2.2.9. Hi^ou ch^onh s^e th^up ph^on.

Có ph^op của c^ou l^onh: **DA A**

Ch^oc n^ong: Hi^ou ch^onh th^up ph^on néi dung 8 bit trong thanh ghi A sau khi th^uc hi^on ph^op céng.

N^ou 4 bit th^up trong thanh ghi A c^a gi^o, tr^up lⁱn h^on 9 ho^ac c^e nhⁱ ph^o ®ⁱc thi^ot lⁱp th^x ph^oi céng th^am 6 v^uo thanh ghi A ®^o cho ch^o s^e th^up ph^on ®ⁱc ch^ynh x^oc. Ph^op céng n^uy sⁱ ®^oEt c^e nhⁱ n^ou s^e nhⁱ t^o 4 bit th^up chuy^on ®^on t^ut c^o 4 bit cao, ngⁱc lⁱi ph^op to^on kh^ong xo^o, c^e nhⁱ.

N^ou 4 bit cao trong thanh ghi A c^a gi^o, tr^up lⁱn h^on 9 ho^ac c^e nhⁱ (CF) ®ⁱc thi^ot lⁱp, th^x c^ong ph^oi céng th^am 6 v^uo thanh ghi A.

C^ou l^onh	S^e byte	S^e chu kú	M- l^onh
---	-------------------------------	---------------------------------	-------------------------------

DA A	1	1	110101 00
------	---	---	--------------

Ho¹t ®éng:

- NÕu $[(A3-A0)>9]$ hoÆc $[(AC)=1]$ th× $(A3-A0) \leftarrow (A3-A0) + 6$
- NÕu $[(A7-A4)>9]$ hoÆc $[(C)=1]$ th× $(A7-A4) \leftarrow (A7-A4) + 6$

2.3. Nhãm lÖnh tÝnh to,n logic.

2.3.1. LÖnh AND cho c,c biÖn 1 byte.

Có ph,p c©u lÖnh: **ANL** **<dest-byte>, <src-byte>**

Chøc năng: Thùc hiÖn phÐp to,n logic AND theo mæc bit gi÷a c,c biÖn dµi 1 byte ®· cho, kÖt qu¶ ®íc lu vµo to,n h¹ng ®Ých. To,n h¹ng nguån cho phÐp 6 chÖ ®é ®¸a chØ ho,. Khi to,n h¹ng ®Ých lµ thanh ghi tÝch luÛ th× to,n h¹ng nguån c¶ thÓ lµ thanh ghi, trùc tiÕp, thanh ghi-gi,n tiÕp hoÆc tæc thêi. Khi to,n h¹ng ®Ých lµ ®¸a chØ trùc tiÕp th× to,n h¹ng nguån c¶ thÓ lµ thanh ghi tÝch luÛ hoÆc d÷ liÖu tæc thêi. LÖnh nµy kh«ng lµm ¶nh hëng tí tr¹ng th,i c,c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
ANL A, Rn	1	1	01011rrr	$(A) \leftarrow (A) \text{ AND } (Rn)$
ANL A, direct	2	1	01010101 aaaaaaaa	$(A) \leftarrow (A) \text{ AND } (\text{dir.})$
ANL A, @Ri	1	1	0101011i	$(A) \leftarrow (A) \text{ AND } ((Ri))$
ANL A, #data	2	1	01010100 dddddddd	$(A) \leftarrow (A) \text{ AND } \#data$
ANL direct, A	2	1	01010010 aaaaaaaa	$(\text{dir.}) \leftarrow (\text{dir.}) \text{ AND } (A)$
ANL direct, #data	3	2	01010011 aaaaaaaa	$(\text{dir.}) \leftarrow (\text{dir.}) \text{ AND } \#data$

			ddddddddd	
--	--	--	-----------	--

2.3.2. Lệnh AND cho các biến 1 bit

Có phép cộng lệnh: **ANL C, <src-bit>**

Chức năng: Thực hiện phép tính logic AND cho các biến mức bit. Nếu giá trị logic của toàn hàng nguồn bằng 0, thì cờ nhớ bặt. Nếu "/" không tính 1 toàn hàng cho biết bit nguồn nào bị tính khi thực hiện AND với cờ nhớ nhưng **giá trị của bit nguồn không bị thay đổi bởi thao tác lấy bit**. Lệnh này không làm ảnh hưởng tới trạng thái các cờ khác. Toàn hàng nguồn chẵn lẻ số dòng kiểu phân chia trực tiếp.

Cộng lệnh	Số byte	Số chu kỳ	Mệnh	Họ tên
ANL C, bit	2	2	10000010 bbbbbbbb	(C) ← (C) AND (bit)
ANL C, /bit	2	2	10110000 bbbbbbbb	(C) ← (C) AND NOT (bit)

2.3.3. Lệnh OR cho các biến 1 byte

Có phép cộng lệnh: **ORL <dest-byte>, <src-byte>**

Chức năng: Thực hiện phép toán logic OR theo mức bit giá trị các biến dài 1 byte cho, kết quả lưu vào toàn hàng đích. Toàn hàng nguồn cho phép 6 chữ số phân chia. Khi toàn hàng đích lưu thành ghi tích lũy thì toàn hàng nguồn cả thảy lưu thành ghi, trực tiếp, thành ghi-giữ tiếp hoặc tác động. Khi toàn hàng đích lưu phân chia trực tiếp thì toàn hàng nguồn cả thảy lưu thành ghi tích lũy hoặc dữ liệu tác động. Lệnh này không làm ảnh hưởng tới trạng thái các cờ.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
ORL A, Rn	1	1	01001rrr	(A)<-(A) OR (Rn)
ORL A, direct	2	1	01000101 aaaaaaaa	(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 aaaaaaaa	(dir.)<-(dir.) OR (A)
ORL direct, #data	3	2	01000011 aaaaaaaa dddddddd	(dir.)<(dir.) OR #data

2.3.4. LÖnh OR cho c, c biÖn 1 bit

Có ph, p c©u lÖnh: **ORL C, <src-bit>**

Chøc n"ng: Thùc hiÖn phÐp tÝnh logic OR cho c, c biÖn mÛc bit. NÕu gi, trÞ logic cña to, n h¹ng nguån b»ng 1, th× cê nhí ®êc thiÖt lÖp. Dêu "/" ®øng tríc 1 to, n h¹ng cho biÖt bit nguån ®êc lÖy bÞ tríc khi thùc hiÖn OR vói cê nhí nhng **gi, trÞ cña bit nguån kh«ng bÞ thay ®æi bëi thao t, c lÖy bÞ**. LÖnh nuy kh«ng lµm ¶nh hëng tí tr¹ng th, i c, c cê kh, c.

C©u lÖnh	Sè byt e	Sè chu kú	M· lÖnh	Ho¹t ®éng
ORL C, bit	2	2	01110010 bbbbbbbb	(C)<-(C) OR (bit)
ORL C, /bit	2	2	10100000 bbbbbbbb	(C)<-(C) OR NOT (bit)

2.3.5. LÖnh X-OR cho c, c biÖn 1 byte

Có ph, p c©u lÖnh: **XRL <dest-byte>, <src-byte>**

Chøc n"ng: Thùc hiÖn phÐp to, n logic X-OR theo mÛc bit gi÷a c, c biÖn dui 1 byte ®· cho, kÕt qu¶ ®êc lu vµo to, n h¹ng

®Ých. To_n h¹ng nguấn cho phĐp 6 chỖ ®é ®Đa chỖ ho₂. Khi to_n h¹ng ®Ých lụ thanh ghi tÝch luũ th× to_n h¹ng nguấn cũ thÓ lụ thanh ghi, trùc tiÕp, thanh ghi-gi_n tiÕp hoÆc t¸c thêi. Khi to_n h¹ng ®Ých lụ ®Đa chỖ trùc tiÕp th× to_n h¹ng nguấn cũ thÓ lụ thanh ghi tÝch luũ hoÆc d÷ liÕu t¸c thêi. LÖnh nşy kh«ng lụm ¶nh hēng tí tr¹ng th_i c₂c cê.

C©u lÖnh	Sè by te	Sè chu kú	M- lÖnh	Ho¹t ®éng
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

2.3.6. LÖnh dÞch tr_i thanh ghi A

Có ph₂p c©u lÖnh: **RL A**

Ch¸c n"ng: 8 bit trong thanh ghi A ®íc dÞch tr_i 1 bit. Bit 7 ®íc quay ®Õn vÞ trÝ cũa bit 0. C₂c cê kh«ng bÞ ¶nh hēng.

C©u lÖnh	Sè by te	Sè chu kú	M- lÖnh	Ho¹t ®éng
RL A	1	1	001000 11	(An+1) <- (An), với n = 0... 6 (A0) <- (A7)

2.3.7. LÖnh dÞch tr_i thanh ghi A cēng vớ cê nhí

Có ph₂p c©u lÖnh: **RLC A**

Chức năng: 8 bit trong thanh ghi A và cờ nháy cũng được dịch trái 1 bit. Bit 7 được chuyển đổi cờ nháy và trạng thái ban đầu của cờ nháy được đặt vào bit 0. Các cờ khác không bị ảnh hưởng.

Câu lệnh	Số byte	Số chu kỳ	M- lệnh	Họ tác động
RLC A	1	1	001100 11	$(A_{n+1}) \leftarrow (A_n)$, với $n = 0 \dots 6$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$

2.3.8. Lệnh dịch phải thanh ghi A.

Có ph, p câu lệnh: **RR A**

Chức năng: 8 bit trong thanh ghi A được dịch sang phải 1 bit. Bit 0 được quay đổi và trở về của bit 7. Các cờ khác không bị ảnh hưởng.

Câu lệnh	Số byte	Số chu kỳ	M- lệnh	Họ tác động
RR A	1	1	000000 11	$(A_n) \leftarrow (A_{n+1})$, với $n = 0 \dots 6$ $(A_7) \leftarrow (A_0)$

2.3.9. Lệnh dịch phải thanh ghi A cũng với cờ nháy

Có ph, p câu lệnh: **RRC A**

Chức năng: 8 bit trong thanh ghi A và cờ nháy cũng được dịch phải 1 bit. Bit 0 được chuyển đổi cờ nháy và trạng thái ban đầu của cờ nháy được đặt vào bit 7. Các cờ khác không bị ảnh hưởng.

Câu lệnh	Số byte	Số chu kỳ	M- lệnh	Họ tác động
RRC A	1	1	000100 11	$(A_n) \leftarrow (A_{n+1})$, với $n = 0 \dots 6$ $(A_7) \leftarrow (C)$

			(C) <- (A0)
--	--	--	-------------

2.3.10. Lệnh trao đổi nội dung hai nửa byte của A

Có phép c@u 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 cần 1 cú lệnh quay thanh ghi A 4 bit. Các chế độ không bị ảnh hưởng.

C@u lệnh	Số byte	Số chu kỳ	M- lệnh	Hoạt động
SWAP A	1	1	11000100	(A3-A0) <- (A7-A4)

2.4. Nhãn lệnh rõ ràng trong chương trình.

2.4.1. Lệnh giải quyết điều kiện.

Có phép c@u lệnh: ACALL addr11

Chức năng: Giải quyết điều kiện khi kết thúc một chương trình con. Khi điều kiện thỏa mãn thì chế độ ra trong c@u lệnh. Lệnh này tăng bộ nhớ chương trình thêm 2 đơn vị và PC chứa địa chỉ của lệnh tiếp theo ACALL, sau đó đặt nội dung 16 bit của PC vào ngăn xếp (byte thấp đặt trước) và tăng con trỏ ngăn xếp lên 2 đơn vị. Địa chỉ đích sẽ được chuyển thành bằng cách ghép 5 bit cao của thanh ghi PC (sau khi điều kiện thỏa mãn), 3 bit cao của byte mệnh lệnh và byte tiếp theo của lệnh. Do đó chương trình con sẽ giải quyết nhanh chóng trong bộ nhớ 2 Kbyte của bộ nhớ chương trình chỉ ít phải chờ lệnh. Các tài liệu của chương trình con này. Lệnh không làm ảnh hưởng tới các chế độ.

C@u lệnh	Số byte	Số chu kỳ	M- lệnh	Hoạt động

ACALL addr11	2	2	aaa1000 1 aaaaaaaa	(PC) <- (PC) + 2 (SP) <- (SP) + 1 ((SP)) <- (PC7-PC0) (SP) <- (SP) + 1 ((SP)) <- (PC15-PC8) (PC10-PC0) <- (page address)
-----------------	---	---	--------------------------	--

2.4.2. Lệnh gọi dọi.

Có ph_hp c©u l nh: **LCALL addr16**

Ch c n ng: G i m t ch ng tr nh con   t t i   p  ch    c ch  ra trong c u l nh. L nh n y t ng b    m ch ng tr nh th m 3   n v     PC ch a   p  ch  c a l nh k  l nh LCALL, sau    c t n i dung 16 bit c a PC v o ng n x p (byte th p c t tr c) v  t ng con tr  ng n x p l n 2   n v . Ti p theo n  s  chuy n byte th  2 v  byte th  3 trong c u l nh LCALL v o byte cao v  byte th p c a PC. Vi c th c thi ch ng tr nh ti p t c v i l nh     p  ch  n y. Nh  v y ch ng tr nh con c  th  b t   u b ng b t c  n i n o trong kh ng gian b  nh  ch ng tr nh 64 Kbyte. L nh kh ng l m   nh h ng t i c c c .

C�u l�nh	S� by te	S� chu k�	M� l�nh	Ho�t ��ng
LCALL addr16	3	2	00010010 aaaaaaaa aaaaaaaa	(PC) <- (PC) + 3 (SP) <- (SP) + 1 ((SP)) <- (PC7-PC0) (SP) <- (SP) + 1 ((SP)) <- (PC15-PC8) (PC) <- addr15-addr0

2.4.3. L nh quay tr  l i t  ch ng tr nh con.

Có ph_hp c©u lÖnh: RET

Chøc n"ng: Trë vÒ tã ch-ng tr×nh con. LÖnh này ®íc thùc hiÖn sau khi thùc hiÖn xong lÖnh ACALL hoÆc LCALL. RET lÿi l¹i byte cao vµ byte thÊp cña PC tã ng"n xÕp, gi¶m SP ®i 2 ®-n vP. Ch-ng tr×nh tiÕp tôc ®íc thùc hiÖn vói lÖnh cũ ®ã chØ ẽ trong PC. C, c cê kh«ng bP ¶nh häng.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
RET	1	2	00100010	(PC15-PC8) <- ((SP)) (SP) <- (SP) - 1 (PC7-PC0) <- ((SP)) (SP) <- (SP) - 1

2.4.4. LÖnh quay trë l¹i tã ng³/4t.

Có ph_hp c©u lÖnh: RETI

Chøc n"ng: Trë vÒ tã ch-ng tr×nh con. RETI lÿi l¹i byte cao vµ byte thÊp cña PC tã ng"n xÕp, phòc hã logic ng³/4t ®Ó cũ thÓ nhÿn c, c ng³/4t kh, c cũ cïng mçc u tiªn ng³/4t vói ng³/4t ®íc xõ lý, sau ®ã gi¶m SP ®i 2 ®-n vP. Ch-ng tr×nh tiÕp tôc ®íc thùc hiÖn vói lÖnh tríc khi xõ lý ng³/4t vói ®ã chØ ẽ trong PC. C, c cê kh«ng bP ¶nh häng.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
RETI	1	2	00110010	(PC15-PC8) <- ((SP)) (SP) <- (SP) - 1 (PC7-PC0) <- ((SP)) (SP) <- (SP) - 1

2.4.5. LÖnh nh¶y gi, n tiÕp.

Có ph, p c©u lÖnh:
Chøc n"ng:

JMP <vị trí đích>

- **Trong đó:** <Vị trí đích> có thể là nhãn của một lệnh
- **Tác dụng:** Khi gặp lệnh này chương trình chuyển điều khiển (nhảy đến) đến thực hiện lệnh sau <Nhãn đích> mà không phụ thuộc vào bất kỳ điều kiện nào.

Khi đến lệnh này con trỏ dữ liệu của vi điều khiển sẽ đến địa chỉ trong câu lệnh và thực hiện nó

Céng gi, trÞ kh«ng dÊu 8 bit cña thanh ghi A vớ con trỏ d÷ liÖu 16 bit vµ n¹p kÖt qu¶ vµo bé ®Öm ch÷ng tr×nh, kÖt qu¶ nµy chÝnh lµ ®Þa chØ ®Ó n¹p lÖnh kÖ tiÖp. ViÖc céng 16 bit ®íc thùc hiÖn: Sè nhÞ tö 8 bit thÊp ®íc truyÒn ®Ön tÊt c¶ c, c bit cao. C¶ 2, thanh ghi A vµ DPTR ®Òu kh«ng bÞ thay ®æi. LÖnh nµy kh«ng ¶nh hëng tí tr¹ng th, i c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
JMP @A+DPTR	1	2	01110011	(PC)<-(A)+ (DPTR)

2.4.6. LÖnh nh¶y nÖu 1 bit ®íc thiÖt lÊp.

Có ph, p c©u lÖnh: **JB bit, <vị trí đích>**

Chøc n"ng: NÖu bit ®· cho cã gi, trÞ b»ng 1 th× nã nh¶y tí ®Þa chØ ®· x, c ®Þnh trong c©u lÖnh, ngâc l¹i nã sÏ tiÖp tục thùc hiÖn lÖnh tiÖp theo.

§Þa chØ ®Ých ®íc tÝnh b»ng c, ch céng thªm ®é lÖch cã dÊu (t÷ng ®èi) trong byte thø 3 cña lÖnh vớ nôi dung trong PC (sau khi ®íc t÷ng ®Ön ®Þa chØ cña byte ®Çu tiªn cña lÖnh kÖ tiÖp). Bit ®íc kiÓm tra kh«ng bÞ thay ®æi, lÖnh kh«ng ¶nh hëng tí c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
JB bit, rel	3	2	00100000 bbbbbbbb	(PC)<-(PC)+3 NÖu (bit)=1 th×:

			eeeeeeee	(PC)<-(PC) + rel
--	--	--	----------	------------------

2.4.7. Lệnh nháy nối 1 bit không thực thi Lệnh.

Có ph, p c@u l@nh: **JNB bit, rel**

Ch@c n"ng: N@u bit 0 cho c@ gi, tr@ b>ng 0 th× n@ nh@y t@i @pa ch@ 0 x,c @pnh trong c@u l@nh, ng@c l@i n@ s@ ti@p t@t th@ hi@n l@nh ti@p theo. @pa ch@ @Ých @ic tÝnh b>ng c, ch céng th@m @é l@ch c@ d@u (t-ng @èi) trong byte th@ 3 c@ l@nh v@i néi dung trong PC (sau khi @ic t"ng @@n @pa ch@ c@ byte @Çu ti@n c@ l@nh k@ ti@p). Bit @ic ki@m tra kh@ng b@ thay @æi, l@nh kh@ng @nh h@ng t@i c, c @.

C@u l@nh	S@ by te	S@ chu kú	M· l@nh	Ho¹t @éng
JNB bit, rel	3	2	00110000 bbbbbbbb eeeeeeee	(PC)<-(PC)+3 N@u (bit)=0 th×: (PC)<-(PC) + rel

2.4.8. Lệnh nháy nối 1 bit thực thi Lệnh v@ xo, bit @ã.

Có ph, p c@u l@nh: **JBC bit, rel**

Ch@c n"ng: N@u bit 1 cho c@ gi, tr@ b>ng 1 th× n@ nh@y t@i @pa ch@ 0 x,c @pnh trong c@u l@nh v@ xo, bit nuy, ng@c l@i n@ s@ ti@p t@t th@ hi@n l@nh ti@p theo. @pa ch@ @Ých @ic tÝnh b>ng c, ch céng th@m @é l@ch c@ d@u (t-ng @èi) trong byte th@ 3 c@ l@nh v@i néi dung trong PC (sau khi @ic t"ng @@n @pa ch@ c@ byte @Çu ti@n c@ l@nh k@ ti@p). L@nh kh@ng @nh h@ng t@i c, c @.

C@u l@nh	S@ by te	S@ chu kú	M· l@nh	Ho¹t @éng
JBC bit, rel	3	2	00010000 bbbbbbbb	(PC)<-(PC)+3 N@u (bit)=1 th×:

			eeeeeeee	(bit) <- 0 (PC) <- (PC) + rel
--	--	--	----------	-------------------------------------

2.4.9. Lệnh nhảy nếu cờ nhí ®ic thiÖt lËp.

Có ph, p c©u lÖnh: JC rel

Chøc n"ng: Nếu cờ CF cả gi, trÞ b»ng 1 th× nã nhÿy tÝ ®Þa chØ ®· x,c ®Þnh trong c©u lÖnh, ngíc l¹i nã sÝ tiÖp tØc thùc hiÖn lÖnh tiÖp theo. §Þa chØ ®Ých ®ic tÝnh b»ng c, ch céng thªm ®é lÖch cả dÊu (t¬ng ®èi) trong byte thø 2 cña lÖnh víi néi dung trong PC (sau khi ®ic t"ng bëi 2). LÖnh kh«ng ¶nh hëng tÝ c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
JC rel	2	2	01000000 eeeeeeee	(PC) <- (PC) + 2 Nếu (C) = 1 th×: (PC) <- (PC) + rel

2.4.10. Lệnh nhảy nếu cờ nhí kh«ng ®ic thiÖt lËp.

Có ph, p c©u lÖnh: JNC rel

Chøc n"ng: Nếu cờ CF cả gi, trÞ b»ng 0 th× nã nhÿy tÝ ®Þa chØ ®· x,c ®Þnh trong c©u lÖnh, ngíc l¹i nã sÝ tiÖp tØc thùc hiÖn lÖnh tiÖp theo. §Þa chØ ®Ých ®ic tÝnh b»ng c, ch céng thªm ®é lÖch cả dÊu (t¬ng ®èi) trong byte thø 2 cña lÖnh víi néi dung trong PC (sau khi ®ic t"ng bëi 2). LÖnh kh«ng ¶nh hëng tÝ c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
JNC rel	2	2	01010000 eeeeeeee	(PC) <- (PC) + 2 Nếu (C) = 0 th×: (PC) <- (PC) +

				rel
--	--	--	--	-----

2.4.11. Lệnh nhảy nếu thanh ghi A bằng 0.

Có ph, p c@u l@nh: JZ rel

Ch@c n"ng: Nếu tất c, c bit của thanh ghi A cả gi, trP bằng 0 th× ã nháy tới @pa ch@ @· x, c @Pnh trong c@u l@nh, ng-íc l@ ã sĩ tiếp tục thực hi@n l@nh tiếp theo. @pa ch@ @Ých @íc tÝnh bằng c, ch céng th@ m @é l@ch cả d@u (t-ng @èi) trong byte th@ 2 của l@nh với néi dung trong PC (sau khi @íc t"ng b@i 2). L@nh kh@ng @nh h@ng tới c, c cê. Néi dung thanh ghi A kh@ng bP thay @æi.

C@u l@nh	Sè by te	Sè chu kú	M· l@nh	Ho ¹ t @éng
JZ rel	2	2	01100000 eeeeeeee	(PC)<-(PC)+2 Nếu (A)=0 th×: (PC)<-(PC) + rel

2.4.12. Lệnh nhảy nếu thanh ghi A khác 0.

Có ph, p c@u l@nh: JNZ rel

Ch@c n"ng: Nếu cả 1 ho@c nhiều bit của thanh ghi A cả gi, trP bằng 1 th× ã nháy tới @pa ch@ @· x, c @Pnh trong c@u l@nh, ng-íc l@ ã sĩ tiếp tục thực hi@n l@nh tiếp theo. @pa ch@ @Ých @íc tÝnh bằng c, ch céng th@ m @é l@ch cả d@u (t-ng @èi) trong byte th@ 2 của l@nh với néi dung trong PC (sau khi @íc t"ng b@i 2). L@nh kh@ng @nh h@ng tới c, c cê. Néi dung thanh ghi A kh@ng bP thay @æi.

C@u l@nh	Sè by te	Sè chu kú	M· l@nh	Ho ¹ t @éng
JNZ rel	2	2	01110000 eeeeeeee	(PC)<-(PC)+2 Nếu (A) < > 0

				thx: (PC)← (PC) + rel
--	--	--	--	-----------------------------

2.4.13. Lệnh nhảy khi số, nh 2 to, n h¹ng.

Có ph, p c©u lÖnh: **CJNE** **<dest-byte>, <src-byte>, rel**

Chøc n"ng: So s, nh gi, trÞ cña 2 to, n h¹ng ®Çu tiªn, nÕu 2 to, n h¹ng kh«ng b»ng nhau th× ch÷ng tr×nh ®íc rñ nh, nh. §Pa chØ ®Ých rñ nh, nh ®íc tÝnh b»ng c, ch céng ®é lÖch t÷ng ®èi (cã dÊu) trong byte sau cïng cña lÖnh víi néi dung cña PC (sau khi néi dung cña PC ®íc t÷ng ®Ön ®Pa chØ b³/₄t ®Çu cña lÖnh kÖ tiÖp CJNZ). Cê nhí (CF) sÏ ®íc thiÖt lÖp nÕu nh gi, trÞ nguyªn kh«ng dÊu cña to, n h¹ng ®Ých nhá h÷n gi, trÞ nguyªn kh«ng dÊu cña to, n h¹ng nguån, ngíc l¹i th× cê nÿy bÞ xoÏ. LÖnh nÿy kh«ng lÿm thay ®æi gi, trÞ cña c, c to, n h¹ng

C©u lÖnh	Sè by te	Sè chu kú	M- lÖnh	Ho¹t ®éng
CJNE A, direct, rel	3	2	101101 01 aaaaaa aa eeeeee ee	(PC)←(PC)+3 NÕu (A) < > (dir.) th×: (PC)← (PC) + offset NÕu (A) < (dir.) th×: (C) <- 1, ngíc l¹i: (C) <- 0
CJNE A, #data, rel	3	2	101101 00 dddddd dd eeeeee ee	(PC)←(PC)+3 NÕu (A) < > #data th×: (PC)← (PC) + offset NÕu (A) < #data th×: (C) <- 1, ngíc l¹i: (C) <- 0
CJNE Rn, #data, rel	3	2	10111r rr	(PC)←(PC)+3 NÕu (Rn) < > #data

			dddddd dd eeeeeee ee	th×: (PC) <- (PC) + offset Nếu (Rn) < #data th×: (C) <- 1, ngic l ⁱ i: (C) <- 0
C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
CJNE @Ri, #data, rel	3	2	101101 li dddddd dd eeeeeee ee	(PC) <- (PC) + 3 Nếu ((Ri)) < > #data th×: (PC) <- (PC) + offset Nếu ((Ri)) < #data th×: (C) <- 1, ngic l ⁱ i: (C) <- 0

2.4.14. LÖnh gi¶m vµ nh¶y.

Có ph, p c©u lÖnh: **DJNZ <byte>, <rel-address>**

Chøc n"ng: Gi¶m « nhí ®i 1 vµ nh¶y tii ®pa chø cho bëi to, n h¹ng thø 2 nÖu nh kÖt qu¶ kh, c 0. NÖu kÖt qu¶ ban ®Çu lµ 00h th× nã chuyÖn qua 0FFh. §pa chø ®Ých ®íc tÝnh b»ng c, ch céng thãm ®é lÖch cã dÊu trong byte lÖnh cuèi cïng víi néi dung cña PC (sau khi t"ng PC tíi byte ®Çu tiªn cña lÖnh tiÖp theo). Ng"n nhí ®íc gi¶m gi, trÞ cã thÓ lµ 1 thanh ghi hoÆc 1 byte ®pa chø trùc tiÖp. LÖnh nµy kh«ng ¶nh hëng tíi tr¹ng th, i c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
DJNZ Rn, rel	2	2	11011r rr	(PC) <- (PC) + 2 (Rn) <- (Rn) - 1

			eeeeee ee	NỎu (Rn) < > 0 th×: (PC) <- (PC) + rel
DJNZ Direct, rel	3	2	110101 01 aaaaaa aa eeeeee ee	(PC)<-(PC)+2 (dir.)<- (dir.) - 1 NỎu (dir.) < > 0 th×: (PC) <- (PC) + rel

2.4.15. LỜnh t¹m ngỗng ho¹t ®éng.

Có ph₂p c©u lỜnh: **NOP**

Chợc n"ng: T¹m ngỗng ho¹t ®éng khi cả lỜnh nựy vµ ch÷ng tr×nh sĩ tiỜp tớc ®íc thùc hiỜn ề lỜnh tiỜp theo. LỜnh nựy kh«ng ¶nh hēng tí tr¹ng th₂i c₂c thanh ghi vµ c₂c cê.

C©u lỜnh	Sề by te	Sề chu kú	M· lỜnh	Ho¹t ®éng
NOP	1	1	000000 00	(PC)<-(PC)+2

2.5. Nhẫm lỜnh ®iỜu khiỜn biỜn logic.

2.5.1. LỜnh xo₂ bit

Có ph₂p c©u lỜnh: **CLR bit**

Chợc n"ng: Xo₂ bit ®íc chỖ ra trong c©u lỜnh vỜ 0. LỜnh nựy cả thỚ thao t₂c trªn cê nhí, hoÆc trªn 1 bit bÊt kú ®íc ®¶nh ®ãa chỖ trùc tiỜp. LỜnh kh«ng lµm ¶nh hēng tí tr¹ng th₂i c₂c cê.

C©u lỜnh	Sề by te	Sề chu kú	M· lỜnh	Ho¹t ®éng
CLR C	1	1	11000011	(C) <- 0
CLR bit	2	1	11000010 bbbbbbbb	(bit) <- 0

2.5.2. LỜnh xo₂ thanh ghi tỶch luũ

Có ph, p c©u lÖnh: **CLR A**

Chøc n"ng: Xo, tÊt c¶ c, c bit cña thanh ghi tÝch luÛ vÒ 0. C, c cê kh«ng b¶ ¶nh h-
ëng.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
CLR A	1	1	11100100	(A) <- 0

2.5.3. LÖnh thiÖt lËp bit

Có ph, p c©u lÖnh: **SETB bit**

Chøc n"ng: ThiÖt lËp bit ®íc chØ ra trong c©u lÖnh l²n mæc logic 1. LÖnh nµy c¶ thÓ thao t, c trªn cê nhí, hoÆc trªn 1 bit bÊt kú ®íc ®¶nh ®¶a chØ tróc tiÕp. LÖnh kh«ng lµm ¶nh hëng tí tr¹ng th, i c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
SETB C	1	1	11010011	(C) <- 1
SETB bit	2	1	11010010 bbbbbbbb	(bit) <- 1

2.5.4. LÖnh lÊy bĩ cña bit

Có ph, p c©u lÖnh: **CPL <bit>**

Chøc n"ng: LÊy bĩ bit ®íc chØ ra trong c©u lÖnh. Mét bit c¶ gi, trÞ 1 ®íc ®æi thµnh 0 vµ ngüc l¹i. LÖnh nµy c¶ thÓ thao t, c trªn cê nhí, hoÆc trªn 1 bit bÊt kú ®íc ®¶nh ®¶a chØ tróc tiÕp. LÖnh kh«ng lµm ¶nh hëng tí tr¹ng th, i c, c cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
CPL C	1	1	10110011	(C) <- NOT (C)
CPL bit	2	1	10110010 bbbbbbbb	(bit) <- NOT (bit)

2.5.5. LÖnh lÊy bĩ cña thanh ghi tÝch luÛ

Có ph_hp c©u lÖnh: CPL A

Chøc n"ng: LÊy bã tÊt c¶ c_hc bit cña thanh ghi A. LÖnh kh«ng lµm ¶nh hÖng tí tr¹ng th_hi c_hc cê.

C©u lÖnh	Sè by te	Sè chu kú	M· lÖnh	Ho¹t ®éng
CPL A	1	1	11110100	(A) <- NOT (A)

CHƯƠNG 3: Ứng dụng lập trình với họ IC AT89

I. Kiến thức cần thiết

Vì ®iều khiển chø xử lý ch÷ng tr×nh dñi dñng file Hex. V× vÿ cÇn ph¶i cã phÇn mÒm biªn dÞch ch÷ng tr×nh sang m· Hex. HiÖn nay, cã rÊt nhiÒu phÇn mÒm cã thÓ lÿp tr×nh vµ m« pháng: keil C, prog studio, reads 51, ride...V× u ®iÓm sô dōng dō dñng, m« pháng trùc quan nªn phÇn mÒm ride ®íc lùa chän giú thiÖu trong gi tr×nh nµy.

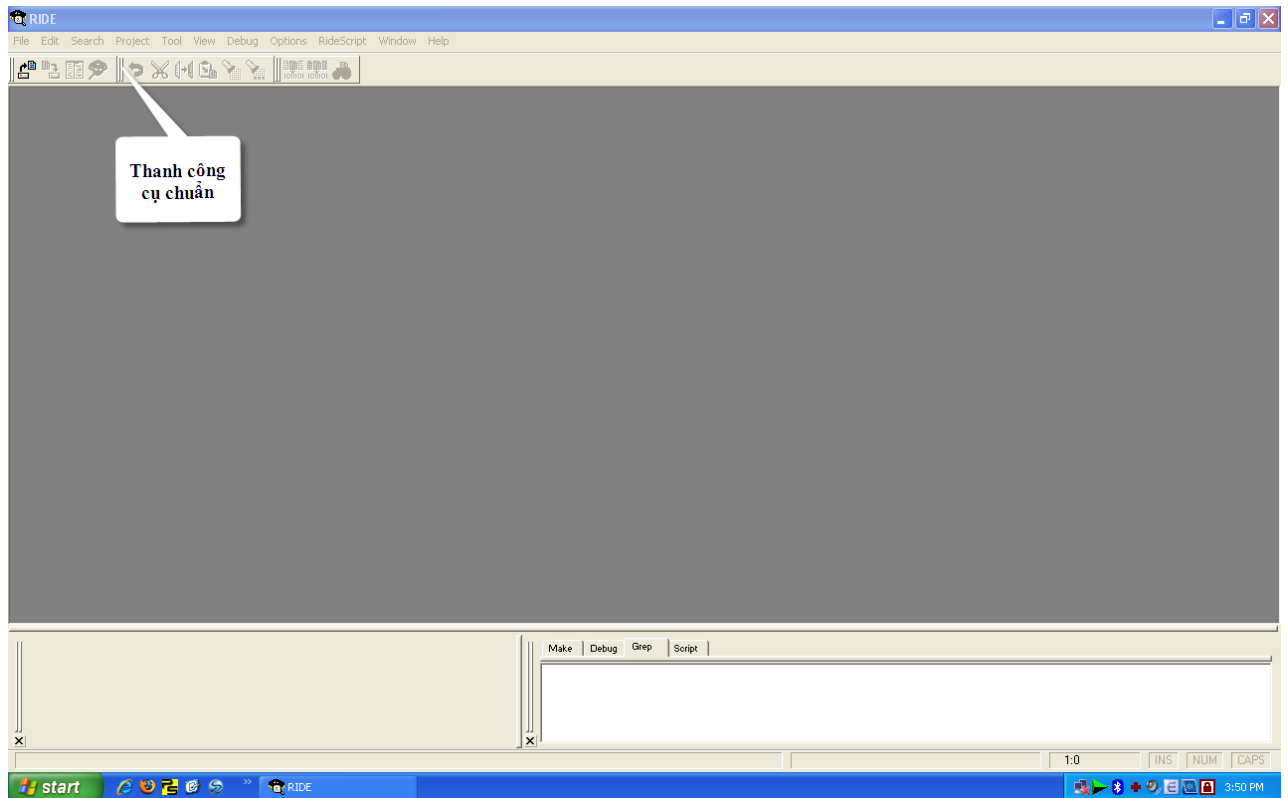
SỐ khêi ®éng ch÷ng tr×nh

- **Start > Programs > Raisonance Kit 6.1 > Ride IDE**



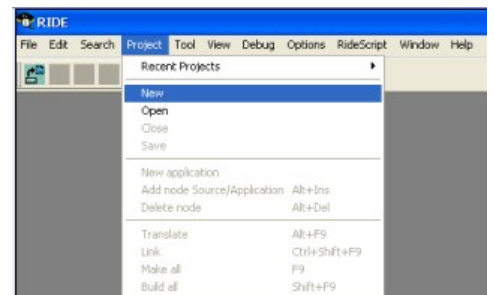
- **Double click** trªn Desktop

Cõa sã ch÷ng tr×nh sau khi khêi ®éng:

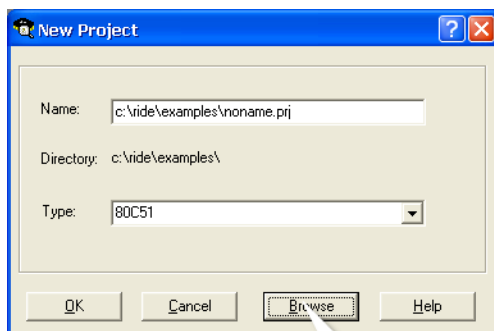


SỐ IẾp tr×nh vµ m« ph¸ng trªn phÇn n

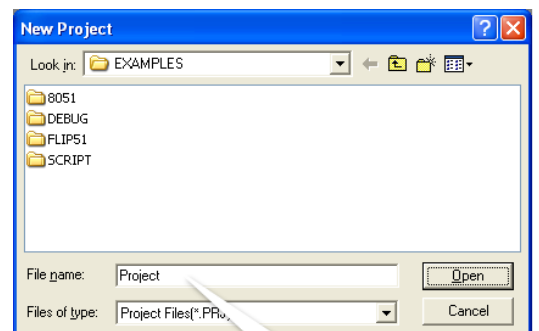
- **Bíc 1: T¹o Project**
+ Project > New



Mét hép tho¹i yu cÇu b¹n chØ ®êng dn, ®Æt tªn cho Project

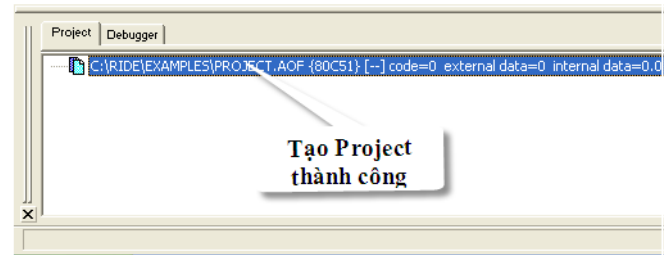


Lựa chọn file cất project

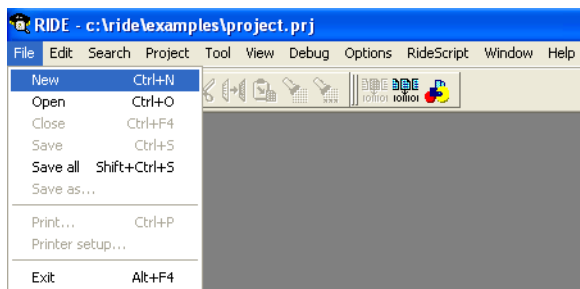


Đặt tên cho Project

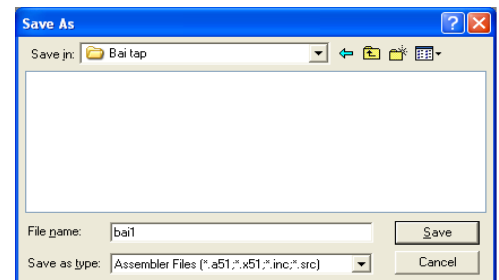
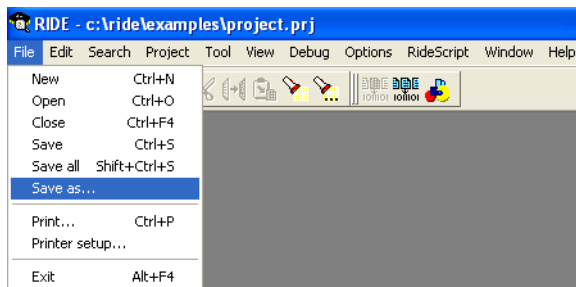
Như vậy bạn đã tạo Project thành công



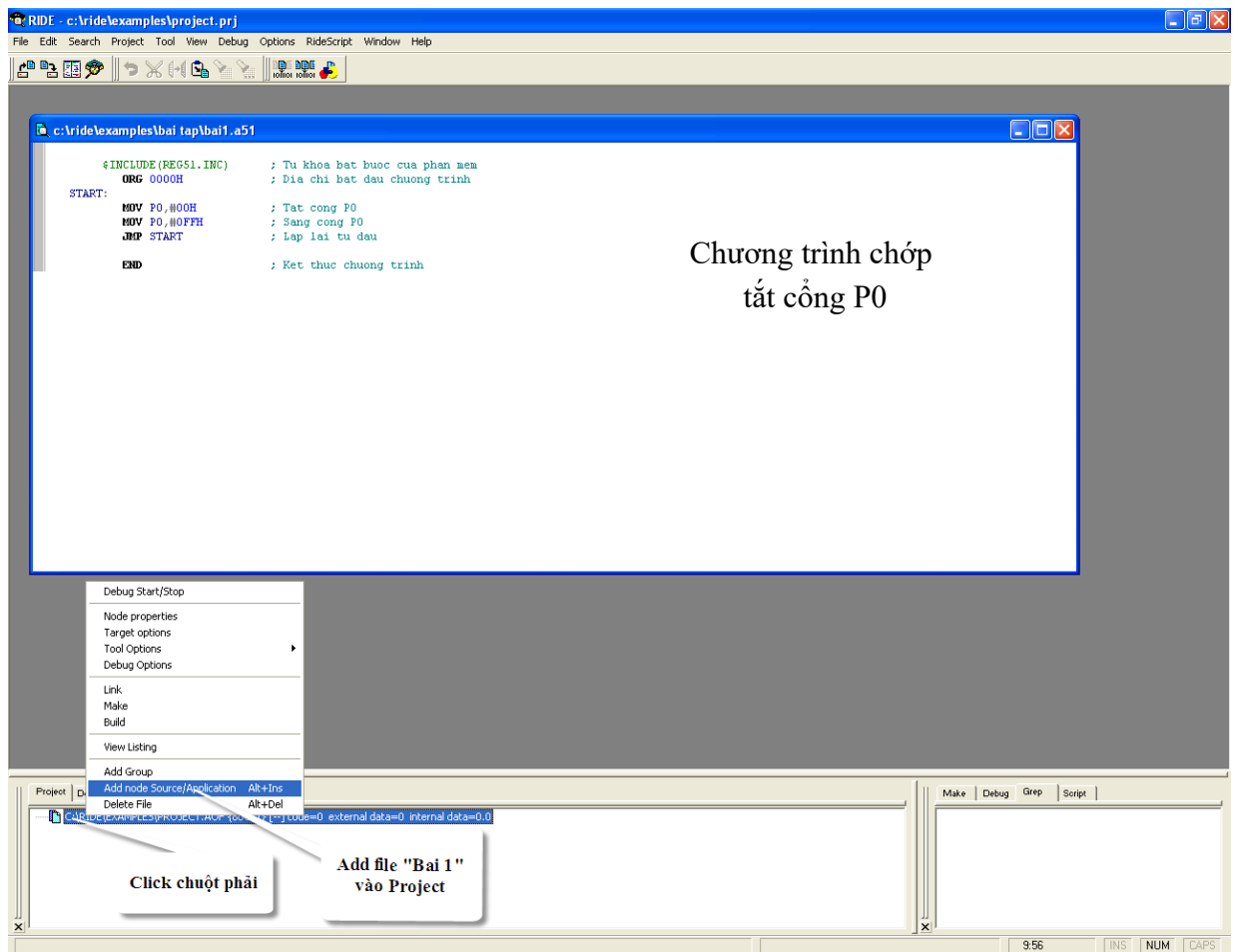
- **Bước 2: Tạo file tập hợp mã nguồn để lắp trình**
+ File > New > Assembler Files



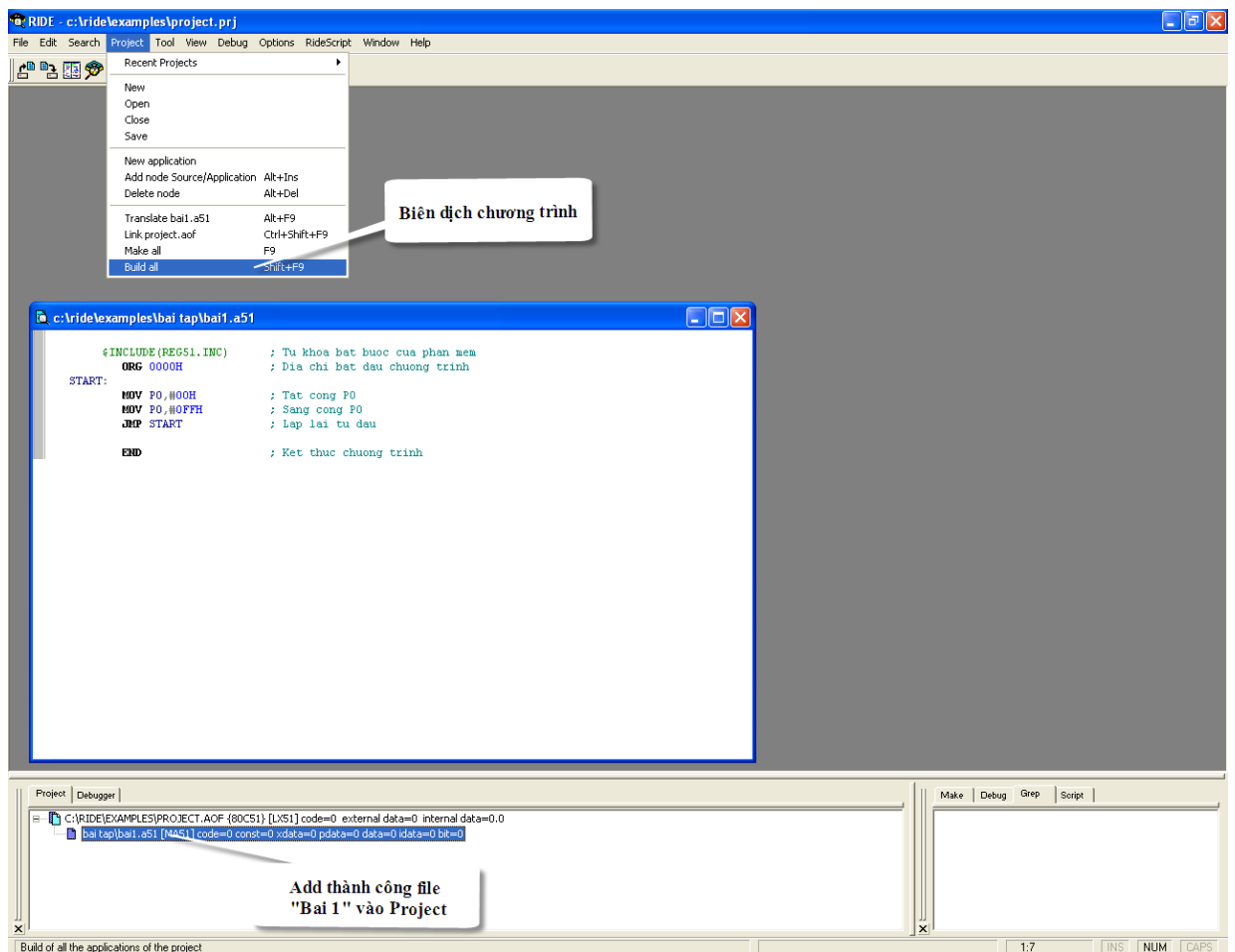
+ Lưu file Assembler với tên “Bài 1” trong th mục “Bài tập”



- **Bước 3: Viết chương trình chip 8086 bằng P0, add file “Bài 1” vào Project**

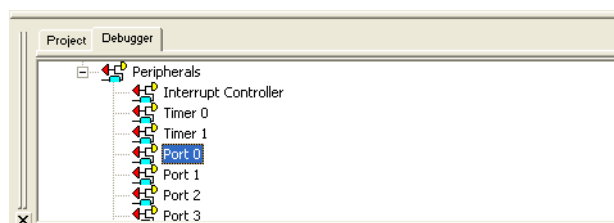
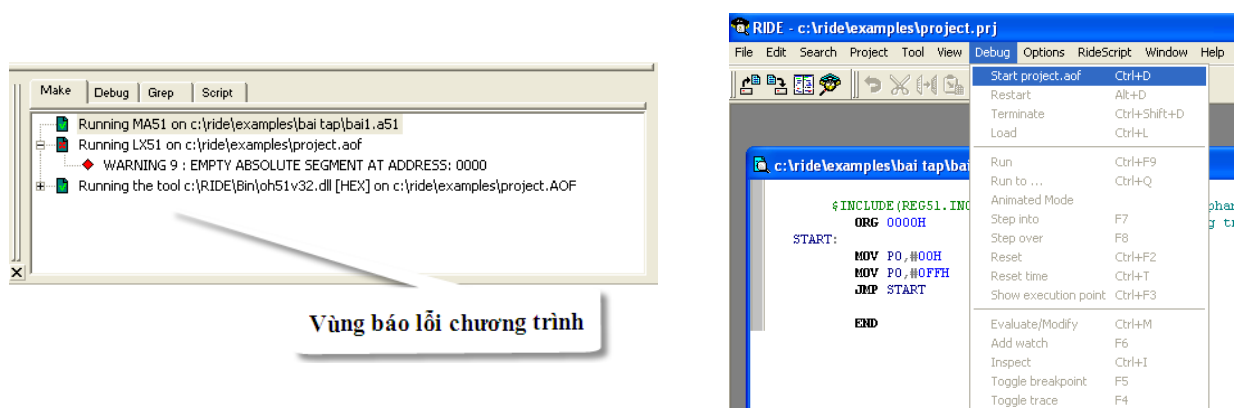


- **Bíc 4: Biªn dÞch ch¬ng tr×nh, s¸a l¹i vµ ch¹y m« ph¸ng**
+ Project > Build all (Shift+F9)



+ Chương trình không báo lỗi, thực hiện chạy mẫu phóng

Debug > Start project.aof (Ctrl+D)



- Click

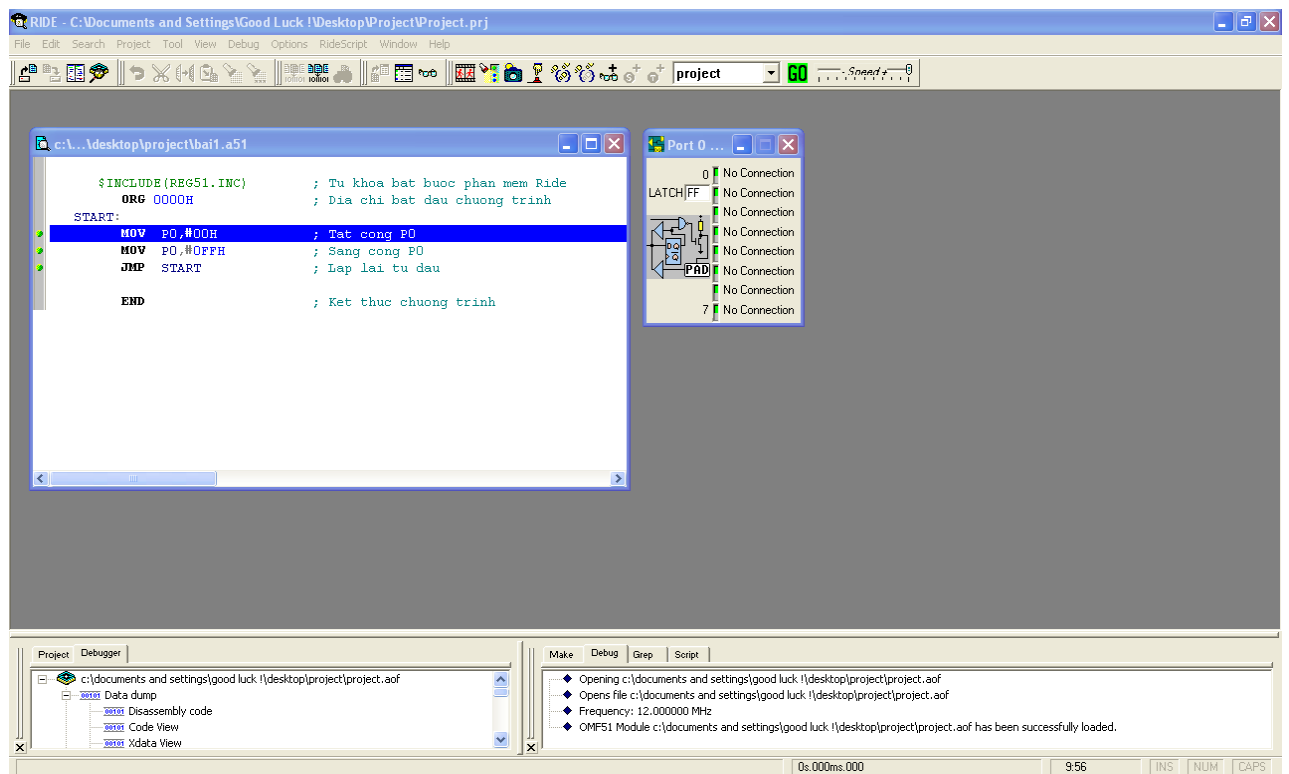


Double click P0

Sau @ ã Cli



+ Quan s_t kÕt qu¶ khi ch¹y ch¬ng tr×nh



3.1 Bài tập với nhóm lệnh di chuyển dữ liệu

Bài số 1.

D÷ kiÖn	Yêu cầu
- (30h) ã RAM néi = 40h. - (40h) ã RAM néi = 10h.	- ChuyÖn néi dung ti vÞ trÝ 30h ã RAM néi vµo R1 vµ A. Mov A, 30h Mov R1, 30h - ChuyÖn néi dung ti vÞ trÝ 40h ã RAM néi vµo B. Mov B, 40h - ChuyÖn néi dung 0CAh vµo vÞ trÝ 40h ã RAM néi vµo cæng P2. mov 40h, #0CAh mov P2, 40h - Xc ®Þnh gi trÞ ca thanh ghi R1 , thanh ghi A , B vµ néi dung ca RAM néi ti cc vÞ trÝ trªn b¶ng Main Register. R1 = 40h (là nội dung trong 30h) A= 40h B= 10h - XuÊt d÷ liÖu ca A vµ B ra cæng P1 vµ P3. Mov P1, A Mov P3, B

Bài số 2.

D÷ kiÖn	Yêu cầu
- (C) = 1 ; (P2) = 0C5h . - (P1) = 35h.	- Xc ®Þnh néi dung míi ca P1 sau khi thùc hiÖn cc lÖnh sau : MOV P1.3 , C MOV C , P2.3 MOV P1.2 , C

Bài số 3.

D÷ kiÖn	Yêu cầu
- (R0) = 11h ; (R1) = 23h . - (11h) ã RAM ngoµi = 35h.	- Sao chÐp néi dung ti vÞ trÝ 11h ã RAM ngoµi vµo A. - Sao chÐp néi dung 55h vµo vÞ trÝ 23h ca RAM ngoµi. - XuÊt d÷ liÖu ca A ra cæng P0 b»ng phng php ®Þnh ®Þa chØ trùc tiÖp.

3.2 Bài tập với nhóm lệnh tính toán logic

Bài tập 1.

D÷ kiÖn	Yêu cầu
Tổ lÖnh céng sau : MOV A , # 38h ADD A , # 2Fh <pre> 38 00111000 2F 00101111 ----- 67 01100111 </pre>	Xét ®Þnh trÞng thÞi c¸c bit cê CY , AC vµ P . Tr¶ lÊi : Cê CY = 0 v× kh«ng c¸nh¸i t¸ bit D7 Cê AC = 1 v× c¸nh¸i t¸ bit D3 sang D4 Cê P = 1 v× thanh ghi A c¸ 5 bit 1 lÊ .

Bài tập 2.

D÷ kiÖn	Yêu cầu
Tổ lÖnh céng sau : MOV A , # 9Ch ADD A , # 64h <pre> 9C 10011100 64 01100100 ----- 100 00000000 </pre>	Xét ®Þnh trÞng thÞi c¸c bit cê CY , AC vµ P . Tr¶ lÊi : Cê CY = 1 v× c¸nh¸i qua bit D7 Cê AC = 1 v× c¸nh¸i t¸ bit D3 sang D4 Cê P = 0 v× thanh ghi A kh«ng c¸ bit 1 ch½n.

Bài tập 3.

D÷ kiÖn	Yêu cầu
Tổ lÖnh céng sau : MOV A , # 38h ADD A , # 2Fh <pre> 88 10001000 93 10010011 ----- 11B 00011011 </pre>	Xét ®Þnh trÞng thÞi c¸c bit cê CY , AC vµ P . Tr¶ lÊi : Cê CY = 1 v× c¸nh¸i t¸ bit D7. Cê AC = 0 v× kh«ng c¸nh¸i t¸ bit D3 sang D4 Cê P = 0 v× thanh ghi A c¸ 4 bit 1 (ch½n) .

Bài sè 4.

D÷ kiÖn	Yêu cÇu
(A) = 0C3h . (R1) = 0AAh.	- Xúc ®pnh néi dung cña A, cê nhí phô AC , cê nhí CF , cê OV sau khi thùc hiÖn lÖnh : ADD A,R1

3.3 Bài tập với nhóm lệnh rẽ nhánh, so sánh, giảm và nhảy

Bài sè 1.

D÷ kiÖn	Yêu cÇu
(A) = 0C3h . (R0) = 55h.	- Xúc ®pnh (A) sau khi thùc hiÖn lÖnh : ANL A,R0

Bài sè 2.

D÷ kiÖn	Yêu cÇu
(P1.0) = 1 , ACC.7 =1 , OV = 0	- H·y thiÖt lÖp cê nhí b»ng lÖnh ANL.

Bài sè 3.

D÷ kiÖn	Yêu cÇu
(A) = 0C3h . (R0) = 55h.	- Xúc ®pnh (A) sau khi thùc hiÖn lÖnh : ORL A,R0

Bài sè 4.

D÷ kiÖn	Yêu cÇu
(P1.0) = 1 , ACC.7 =1 , OV = 0 .	- H·y thiÖt lÖp cê nhí b»ng lÖnh ORL.

3.4 Bài tập với nhóm lệnh điều khiển biến logic và các lệnh tổ hợp

Bài sè 1.

D÷ kiÖn	Yêu cÇu
Cho ®o¹n chñng trñnh sau : MOV R6 , # 25h MOV R1 , # 12h MOV R4 , # 0F3h PUSH 6	- H·y xúc ®pnh con trá ngñn xÖp (SP) ë ®o¹n chñng trñnh bñn. Gi¶ thiÖt ban ®Çu (SP) = 07h

PUSH 1 PUSH 4	
------------------	--

Bài số 2.

D÷ kiÖn	Yêu cầu
Cho trình tự xử lý sau xử lý ACC và sau đó đếm 3 vào ACC 10 lần: <pre> MOV A, # 0 MOV R2, # 10 Loop: ADD A, # 10h DJNZ R2, Loop MOV R5, A </pre>	- Hãy giải thích tổng cộng lệnh trong trình tự xử lý trên. - Hãy cho biết sẽ lấy ra được bao nhiêu dữ liệu trong trình tự xử lý trên?

Bài số 3.

D÷ kiÖn	Yêu cầu
	- Hãy viết trình tự xử lý để kiểm tra xem R5 đã chứa giá trị 0 hay chưa? Nếu không thì nhập vào R5 giá trị 55h.

Chương 1: Bé Vi 8-bit khi On AT89C51	1
1.1 Giới thiệu chung:	1
1.2 Sự khác nhau giữa bé VXL và bé VSK	1
1.3. Sự 8-bit khi	2
1.4. Sự 8-bit khi của 80C51/AT89C51	5
1.5. Chức năng của thanh phần của AT89C51:	6
1.5.1. Các thanh ghi chức năng 8-bit	6
1.5.2. Khi thời gian và bé 8-bit (Timer/Counter)	14
1.5.3. Bé khi chương trình và bé khi dữ liệu néi tró	17
1.5.4. Nguyên lý khi 8-bit của On-chip AT89C51:	23
1.5.5. Mạch dao 8-bit	25
1.5.6. Khi 8-bit nguồn giáng và khi 8-bit nghé	27
1.5.7. Bộ vi chương trình	27
Chương 2 Tiếp Lệnh của bé VSK AT89/80C51	28
2.1. Nhãn Lệnh di chuyển dữ liệu	29
2.1.1. Lệnh MOV dữ liệu Byte:	29
2.1.2. Lệnh MOV dữ liệu Bit:	30
2.1.3. Lệnh MOV dữ liệu Word:	30
2.1.4. Lệnh chuyển byte m. Lệnh:	30
2.1.5. Lệnh chuyển dữ liệu ra ngoài:	31
2.1.6. Lệnh chuyển dữ liệu vào ng. xỐp:	31
2.1.7. Lệnh chuyển dữ liệu ra khi ng. xỐp:	32
2.1.8. H. chuyển dữ liệu:	32
2.1.9. H. chuyển 4 bit thếp:	32
2.2. Nhãn Lệnh tính toán dữ liệu	33
2.2.1. Lệnh thực hiện phép cộng	33
2.2.2. Lệnh cộng cả khi	33
2.2.3. Lệnh trở cả mìn	34
2.2.4. Lệnh tính l²n 1 8-bit và	34
2.2.5. Lệnh giáng 1 8-bit và	35
2.2.6. Lệnh tính con tró dữ liệu	35

2.2.7. Lệnh thực hiện phép nhân.....	35
2.2.8. Lệnh thực hiện phép chia.....	36
2.2.9. Hiệu chỉnh sẽ tiếp nhận.....	36
2.3. Nhóm lệnh tính toán logic.....	37
2.3.1. Lệnh AND cho c, c biến 1 byte.....	37
2.3.2. Lệnh AND cho c, c biến 1 bit.....	37
2.3.3. Lệnh OR cho c, c biến 1 byte.....	38
2.3.4. Lệnh OR cho c, c biến 1 bit.....	38
2.3.5. Lệnh X-OR cho c, c biến 1 byte.....	38
2.3.6. Lệnh dịch trái thành ghi A.....	39
2.3.7. Lệnh dịch trái thành ghi A cũng với cờ nhí.....	39
2.3.8. Lệnh dịch phải thành ghi A.....	40
2.3.9. Lệnh dịch phải thành ghi A cũng với cờ nhí.....	40
2.3.10. Lệnh trao đổi nội dung hai nửa byte của A.....	40
2.4. Nhóm lệnh rẽ nhánh chọn trình.....	41
2.4.1. Lệnh giải quyết điều kiện.....	41
2.4.2. Lệnh giải quyết.....	41
2.4.3. Lệnh quay trở lại tổ chức trình con.....	42
2.4.4. Lệnh quay trở lại tổ chức ngắt.....	42
2.4.5. Lệnh nhảy gián tiếp.....	42
2.4.6. Lệnh nhảy nếu 1 bit điều kiện thiết lập.....	43
2.4.7. Lệnh nhảy nếu 1 bit không điều kiện thiết lập.....	43
2.4.8. Lệnh nhảy nếu 1 bit điều kiện thiết lập và xoá bit điều kiện.....	44
2.4.9. Lệnh nhảy nếu cờ nhí điều kiện thiết lập.....	44
2.4.10. Lệnh nhảy nếu cờ nhí không điều kiện thiết lập.....	45
2.4.11. Lệnh nhảy nếu thanh ghi A bằng 0.....	45
2.4.12. Lệnh nhảy nếu thanh ghi A khác 0.....	45
2.4.13. Lệnh nhảy khi số sinh 2 toán hạng.....	46
2.4.14. Lệnh giám sát nhảy.....	47
2.4.15. Lệnh tìm kiếm hoặc điều kiện.....	47

2.5. Nhảĩm lỜnh ®iỜu khiỐn biỐn logic.....	48
2.5.1. LỜnh xo, bit.....	48
2.5.2. LỜnh xo, thanh ghi tỶch luũ.....	48
2.5.3. LỜnh thiỐt lẾp bit.....	48
2.5.4. LỜnh lẦy bĩ cĩa bit.....	49
2.5.5. LỜnh lẦy bĩ cĩa thanh ghi tỶch luũ.....	49
CHƯƠNG 3: Ứng dụng lập trình với họ IC AT89.....	50
1. Kiến thức cần thiết.....	50
2. Bài tập ứng dụng.....	55