Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

P89LV51RB2/RC2/RD2

8-bit 80C51 3 V low power 16/32/64 kB flash microcontroller with 1 kB RAM

Rev. 05 — 15 December 2009 Product data sheet

1. General description

The P89LV51RB2/RC2/RD2 are 80C51 microcontrollers with 16/32/64 kB flash and 1024 B of data RAM.

A key feature of the P89LV51RB2/RC2/RD2 is its X2 mode option. The design engineer can choose to run the application with the conventional 80C51 clock rate (12 clocks per machine cycle) or select the X2 mode (six clocks per machine cycle) to achieve twice the throughput at the same clock frequency. Another way to benefit from this feature is to keep the same performance by reducing the clock frequency by half, thus dramatically reducing the EMI.

The flash program memory supports both parallel programming and in serial ISP. Parallel programming mode offers gang-programming at high speed, reducing programming costs and time to market. ISP allows a device to be reprogrammed in the end product under software control. The capability to field/update the application firmware makes a wide range of applications possible.

The P89LV51RB2/RC2/RD2 is also capable of IAP, allowing the flash program memory to be reconfigured even while the application is running.

2. Features

- 80C51 CPU
- 3 V operating voltage from 0 MHz to 33 MHz
- 16/32/64 kB of on-chip flash user code memory with ISP and IAP
- Supports 12-clock (default) or 6-clock mode selection via software or ISP
- SPI and enhanced UART
- PCA with PWM and capture/compare functions
- Four 8-bit I/O ports with three high-current port 1 pins (16 mA each)
- Three 16-bit timers/counters
- Programmable watchdog timer
- Eight interrupt sources with four priority levels
- Second DPTR register
- Low EMI mode (ALE inhibit)
- TTL- and CMOS-compatible logic levels

- Brownout detection
- Low power modes
  - Power-down mode with external interrupt wake-up
  - Idle mode
- PLCC44 and TQFP44 packages

3. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
P89LV51RB2BA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89LV51RC2FBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89LV51RD2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89LV51RD2BBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1

3.1 Ordering options

Table 2. Ordering options

Type number	Flash memory	Temperature range	Frequency
P89LV51RB2BA	16 kB	0 °C to +70 °C	0 MHz to 40 MHz
P89LV51RC2FBC	32 kB	−40 °C to +85 °C	
P89LV51RD2FA	64 kB	−40 °C to +85 °C	
P89LV51RD2BBC	64 kB	0 °C to +70 °C	

4. Block diagram

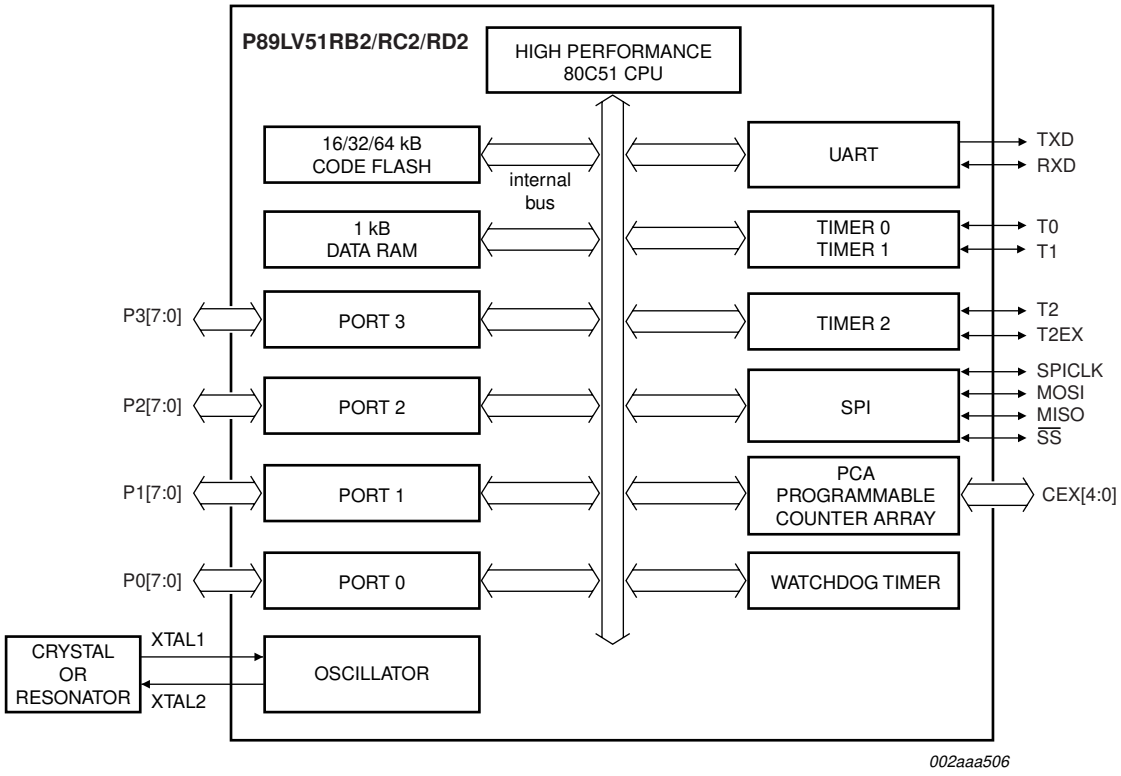

Fig 1. Block diagram

5. Pinning information

5.1 Pinning

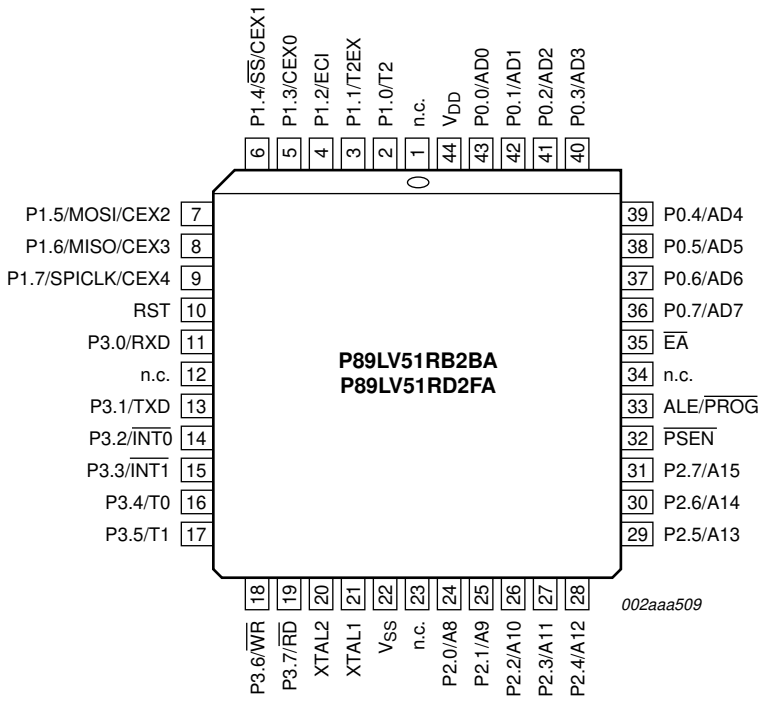

Fig 2. PLCC44 pin configuration

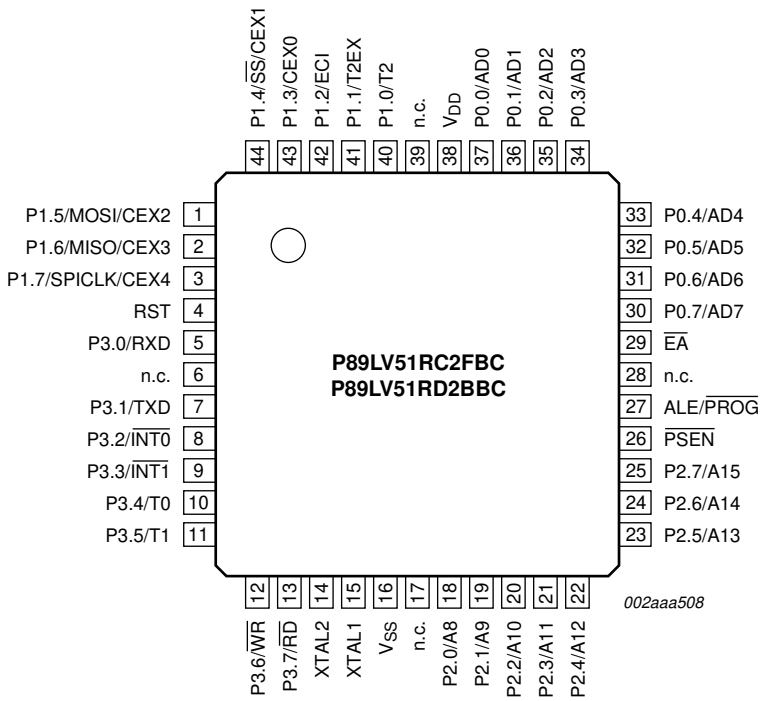

Fig 3. TQFP44 pin configuration

5.2 Pin description

Table 3. P89LV51RB2/RC2/RD2 pin description

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
P0.0 to P0.7			I/O	**Port 0:** Port 0 is an 8-bit open drain bidirectional I/O port. Port 0 pins that have '1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external code and data memory. In this application, it uses strong internal pull-ups when transitioning to '1's. Port 0 also receives the code bytes during the external host mode programming, and outputs the code bytes during the external host mode verification. External pull-ups are required during program verification or as a general purpose I/O port.
P0.0/AD0	37	43	I/O	**P0.0** — Port 0 bit 0.
			I/O	**AD0** — Address/data bit 0.
P0.1/AD1	36	42	I/O	**P0.1** — Port 0 bit 1.
			I/O	**AD1** — Address/data bit 1.
P0.2/AD2	35	41	I/O	**P0.2** — Port 0 bit 2.
			I/O	**AD2** — Address/data bit 2.
P0.3/AD3	34	40	I/O	**P0.3** — Port 0 bit 3.
			I/O	**AD3** — Address/data bit 3.
P0.4/AD4	33	39	I/O	**P0.4** — Port 0 bit 4.
			I/O	**AD4** — Address/data bit 4.
P0.5/AD5	32	38	I/O	**P0.5** — Port 0 bit 5.
			I/O	**AD5** — Address/data bit 5.
P0.6/AD6	31	37	I/O	**P0.6** — Port 0 bit 6.
			I/O	**AD6** — Address/data bit 6.
P0.7/AD7	30	36	I/O	**P0.7** — Port 0 bit 7.
			I/O	**AD7** — Address/data bit 7.
P1.0 to P1.7			I/O with internal pull-up	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 pins are pulled high by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 1 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. P1.5, P1.6, P1.7 have a high current drive of 16 mA. Port 1 also receives the low-order address bytes during the external host mode programming and verification.
P1.0/T2	40	2	I	**P1.0** — Port 1 bit 0.
			I/O	**T2** — External count input to Timer/counter 2 or Clock-out from Timer/counter 2.
P1.1/T2EX	41	3	I/O	**P1.1** — Port 1 bit 1.
			I	**T2EX**: Timer/counter 2 capture/reload trigger and direction control input.
P1.2/ECI	42	4	I/O	**P1.2** — Port 1 bit 2.
			I	**ECI** — External clock input. This signal is the external clock input for the PCA.

Table 3. P89LV51RB2/RC2/RD2 pin description *...continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
P1.3/CEX0	43	5	I/O	**P1.3** — Port 1 bit 3.
			I/O	**CEX0** — Capture/compare external I/O for PCA Module 0. Each capture/compare module connects to a Port 1 pin for external I/O. When not used by the PCA, this pin can handle standard I/O.
P1.4/$\overline{SS}$/CEX1	44	6	I/O	**P1.4** — Port 1 bit 4.
			I	$\overline{SS}$ — Slave port select input for SPI.
			I/O	**CEX1** — Capture/compare external I/O for PCA Module 1.
P1.5/MOSI/ CEX2	1	7	I/O	**P1.5** — Port 1 bit 5.
			I/O	**MOSI** — Master Output Slave Input for SPI.
			I/O	**CEX2** — Capture/compare external I/O for PCA Module 2.
P1.6/MISO/ CEX3	2	8	I/O	**P1.6** — Port 1 bit 6.
			I/O	**MISO** — Master Input Slave Output for SPI.
			I/O	**CEX3** — Capture/compare external I/O for PCA Module 3.
P1.7/SPICLK/ CEX4	3	9	I/O	**P1.7** — Port 1 bit 7.
			I/O	**SPICLK** — Serial clock input/output for SPI.
			I/O	**CEX4** — Capture/compare external I/O for PCA Module 4.
P2.0 to P2.7			I/O with internal pull-up	**Port 2**: Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 2 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 2 sends the high-order address byte during fetches from external program memory and during accesses to external Data Memory that use 16-bit address (MOVX@DPTR). In this application, it uses strong internal pull-ups when transitioning to '1's. Port 2 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P2.0/A8	18	24	I/O	**P2.0** — Port 2 bit 0.
			O	**A8** — Address bit 8.
P2.1/A9	19	25	I/O	**P2.1** — Port 2 bit 1.
			O	**A9** — Address bit 9.
P2.2/A10	20	26	I/O	**P2.2** — Port 2 bit 2.
			O	**A10** — Address bit 10.
P2.3/A11	21	27	I/O	**P2.3** — Port 2 bit 3.
			O	**A11** — Address bit 11.
P2.4/A12	22	28	I/O	**P2.4** — Port 2 bit 4.
			O	**A12** — Address bit 12.
P2.5/A13	23	29	I/O	**P2.5** — Port 2 bit 5.
			O	**A13** — Address bit 13.
P2.6/A14	24	30	I/O	**P2.6** — Port 2 bit 6.
			O	**A14** — Address bit 14.

Table 3. P89LV51RB2/RC2/RD2 pin description *...continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
P2.7/A15	25	31	I/O	**P2.7 —** Port 2 bit 7.
			O	**A15 —** Address bit 15.
P3.0 to P3.7			I/O with internal pull-up	**Port 3**: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 3 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 3 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0/RXD	5	11	I	**P3.0 —** Port 3 bit 0.
			I	**RXD —** Serial input port.
P3.1/TXD	7	13	O	**P3.1 —** Port 3 bit 1.
			O	**TXD —** Serial output port.
P3.2/$\overline{\text{INT0}}$	8	14	I	**P3.2 —** Port 3 bit 2.
			I	**$\overline{\text{INT0}}$ —** External interrupt 0 input.
P3.3/$\overline{\text{INT1}}$	9	15	I	**P3.3 —** Port 3 bit 3.
			I	**$\overline{\text{INT1}}$ —** External interrupt 1 input.
P3.4/T0	10	16	I/O	**P3.4 —** Port 3 bit 4.
			I	**T0 —** External count input to Timer/counter 0.
P3.5/T1	11	17	I/O	**P3.5 —** Port 3 bit 5.
			I	**T1 —** External count input to Timer/counter 1.
P3.6/$\overline{\text{WR}}$	12	18	O	**P3.6 —** Port 3 bit 6.
			O	**$\overline{\text{WR}}$ —** External data memory write strobe.
P3.7/$\overline{\text{RD}}$	13	19	O	**P3.7 —** Port 3 bit 7.
			O	**$\overline{\text{RD}}$ —** External data memory read strobe.
$\overline{\text{PSEN}}$	26	32	I/O	**Program Store Enable**: $\overline{\text{PSEN}}$ is the read strobe for external program memory. When the device is executing from internal program memory, $\overline{\text{PSEN}}$ is inactive (HIGH). When the device is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory. A forced HIGH-to-LOW input transition on the $\overline{\text{PSEN}}$ pin while the RST input is continually held HIGH for more than 10 machine cycles will cause the device to enter external host mode programming.
RST	4	10	I	**Reset**: While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. If the $\overline{\text{PSEN}}$ pin is driven by a HIGH-to-LOW input transition while the RST input pin is held HIGH, the device will enter the external host mode, otherwise the device will enter the normal operation mode.
$\overline{\text{EA}}$	29	35	I	**External Access Enable**: $\overline{\text{EA}}$ must be connected to V_{SS} in order to enable the device to fetch code from the external program memory. $\overline{\text{EA}}$ must be strapped to V_{DD} for internal program execution. The $\overline{\text{EA}}$ pin can tolerate a high voltage of 12 V.

Table 3. P89LV51RB2/RC2/RD2 pin description *...continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
ALE/$\overline{PROG}$	27	33	I/O	**Address Latch Enable:** ALE is the output signal for latching the low byte of the address during an access to external memory. This pin is also the programming pulse input ($\overline{PROG}$) for flash programming. Normally the ALE[1] is emitted at a constant rate of $\frac{1}{6}$ the crystal frequency[2] and can be used for external timing and clocking. One ALE pulse is skipped during each access to external data memory. However, if bit AO is set to '1', ALE is disabled.
n.c.	6, 17, 28, 39	1, 12, 23, 34	I/O	not connected
XTAL1	15	21	I	**Crystal 1**: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	14	20	O	**Crystal 2:** Output from the inverting oscillator amplifier.
V_{DD}	38	44	I	**Power supply**
V_{SS}	16	22	I	**Ground**

[1] ALE loading issue: When ALE pin experiences higher loading (> 30 pF) during the reset, the microcontroller may accidentally enter into modes other than normal working mode. The solution is to connect a pull-up resistor of 3 kΩ to 50 kΩ from pin ALE to V_{DD}.

[2] For 6-clock mode, ALE is emitted at $\frac{1}{3}$ of crystal frequency.

6. Functional description

6.1 Special function registers

Remark: SFR accesses are restricted in the following ways:

- User must **not** attempt to access any SFR locations not defined.
- Accesses to any defined SFR locations must be strictly for the functions for the SFRs.
- SFR bits labeled '-', '0' or '1' can **only** be written and read as follows:
  - '-' Unless otherwise specified, **must** be written with '0', but can return any value when read (even if it was written with '0'). It is a reserved bit and may be used in future derivatives.
  - '0' **must** be written with '0', and will return a '0' when read.
  - '1' **must** be written with '1', and will return a '1' when read.

Table 4. Special function registers

** Indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses							
			MSB							LSB
		Bit address	E7	E6	E5	E4	E3	E2	E1	E0
ACC*	Accumulator	E0H								
AUXR	Auxiliary function register	8EH	-	-	-	-	-	-	EXTRAM	AO
AUXR1	Auxiliary function register 1	A2H	-	-	-	-	GF2	0	-	DPS
		Bit address	F7	F6	F5	F4	F3	F2	F1	F0
B*	B register	F0H								
CCAP0H	Module 0 Capture HIGH	FAH								
CCAP1H	Module 1 Capture HIGH	FBH								
CCAP2H	Module 2 Capture HIGH	FCH								
CCAP3H	Module 3 Capture HIGH	FDH								
CCAP4H	Module 4 Capture HIGH	FEH								
CCAP0L	Module 0 Capture LOW	EAH								
CCAP1L	Module 1 Capture LOW	EBH								
CCAP2L	Module 2 Capture LOW	ECH								
CCAP3L	Module 3 Capture LOW	EDH								
CCAP4L	Module 4 Capture LOW	EEH								
CCAPM0	Module 0 Mode	DAH	-	ECOM_0	CAPP_0	CAPN_0	MAT_0	TOG_0	PWM_0	ECCF_0
CCAPM1	Module 1 Mode	DBH	-	ECOM_1	CAPP_1	CAPN_1	MAT_1	TOG_1	PWM_1	ECCF_1
CCAPM2	Module 2 Mode	DCH	-	ECOM_2	CAPP_2	CAPN_2	MAT_2	TOG_2	PWM_2	ECCF_2
CCAPM3	Module 3 Mode	DDH	-	ECOM_3	CAPP_3	CAPN_3	MAT_3	TOG_3	PWM_3	ECCF_3
CCAPM4	Module 4 Mode	DEH	-	ECOM_4	CAPP_4	CAPN_4	MAT_4	TOG_4	PWM_4	ECCF_4
		Bit address	DF	DE	DD	DC	DB	DA	D9	D8
CCON*	PCA Counter Control	D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
CH	PCA Counter HIGH	F9H								
CL	PCA Counter LOW	E9H								
CMOD	PCA Counter Mode	D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF
DPTR	Data Pointer (2 B)									
DPH	Data Pointer HIGH	83H								
DPL	Data Pointer LOW	82H								

Table 4. Special function registers *...continued*

** Indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses							
			MSB							LSB
FST	Flash Status Register	B6	-	SB	-	-	EDC	-	-	-
		Bit address	**AF**	**AE**	**AD**	**AC**	**AB**	**AA**	**A9**	**A8**
IEN0*	Interrupt Enable 0	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0
		Bit address	**EF**	**EE**	**ED**	**EC**	**EB**	**EA**	**E9**	**E8**
IEN1*	Interrupt Enable 1	E8H	-	-	-	-	EBO	-	-	-
		Bit address	**BF**	**BE**	**BD**	**BC**	**BB**	**BA**	**B9**	**B8**
IP0*	Interrupt Priority	B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0
IP0H	Interrupt Priority 0 HIGH	B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
		Bit address	**FF**	**FE**	**FD**	**FC**	**FB**	**FA**	**F9**	**F8**
IP1*	Interrupt Priority 1	F8H	-	-	-	PBO	-	-	-	-
IP1H	Interrupt Priority 1 HIGH	F7H	-	-	-	PBOH	-	-	-	-
FCF		B1H	-	-	-	-	-	-	SWR	BSEL
		Bit address	**87**	**86**	**85**	**84**	**83**	**82**	**81**	**80**
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Bit address	**97**	**96**	**95**	**94**	**93**	**92**	**91**	**90**
P1*	Port 1	90H	CEX4/ SPICLK	CEX3/ MISO	CEX2/ MOSI	CEX1/$\overline{SS}$	CEX0	ECI	T2EX	T2
		Bit address	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8
		Bit address	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TXD	RXD
PCON	Power Control Register	87H	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL
		Bit address	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P
RCAP2H	Timer2 Capture HIGH	CBH								
RCAP2L	Timer2 Capture LOW	CAH								
		Bit address	**9F**	**9E**	**9D**	**9C**	**9B**	**9A**	**99**	**98**
SCON*	Serial Port Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	Serial Port Data Buffer Register	99H								

Table 4. Special function registers *...continued*

** Indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses							
			MSB							LSB
SADDR	Serial Port Address Register	A9H								
SADEN	Serial Port Address Enable	B9H								
		Bit address	**87**[1]	**86**[1]	**85**[1]	**84**[1]	**83**[1]	**82**[1]	**81**[1]	**80**[1]
SPCTL	SPI Control Register	D5H	SPIE	SPEN	DORD	MSTR	CPOL	CPHA	PSC1	PSC0
SPCFG	SPI Configuration Register	AAH	SPIF	WCOL	-	-	-	-	-	-
SPDAT	SPI Data	86H								
SP	Stack Pointer	81H								
		Bit address	**8F**	**8E**	**8D**	**8C**	**8B**	**8A**	**89**	**88**
TCON*	Timer Control Register	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		Bit address	**CF**	**CE**	**CD**	**CC**	**CB**	**CA**	**C9**	**C8**
T2CON*	Timer2 Control Register	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T}$2	CP/RL2
T2MOD	Timer2 Mode Control	C9H	-	-	ENT2	-	-	-	T2OE	DCEN
TH0	Timer 0 HIGH	8CH								
TH1	Timer 1 HIGH	8DH								
TH2	Timer 2 HIGH	CDH								
TL0	Timer 0 LOW	8AH								
TL1	Timer 1 LOW	8BH								
TL2	Timer 2 LOW	CCH								
TMOD	Timer 0 and 1 Mode	89H	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0
WDTC	Watchdog Timer Control	C0H	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT
WDTD	Watchdog Timer Data/Reload	85H								

[1] Unimplemented bits in SFRs (labeled '-') are 'X's (unknown) at all times. Unless otherwise specified, '1's should not be written to these bits since they may be used for other purposes in future derivatives. The reset values shown for these bits are '0's although they are unknown when read.

6.2 Memory organization

The device has separate address spaces for program and data memory.

6.2.1 Flash program memory bank selection

There are two internal flash memory blocks in the device. Block 0 has 16/32/64 kB and is organized as 128/256/512 sectors, each sector consists of 128 B. Block 1 contains the IAP/ISP routines and may be enabled such that it overlays the first 8 kB of the user code memory. The overlay function is controlled by the combination of the Software Reset Bit (SWR) at FCF.1 and the Bank Select Bit (BSEL) at FCF.0. The combination of these bits and the memory source used for instructions is shown in Table 5.

Table 5. Code memory bank selection

SWR (FCF.1)	BSEL (FCF.0)	Addresses from 0000H to 1FFFH	Addresses above 1FFFH
0	0	boot code (in block 1)	user code (in block 0)
0	1	user code (in block 0)	
1	0		
1	1		

Access to the IAP routines in block 1 may be enabled by clearing the BSEL bit (FCF.0), provided that the SWR bit (FCF.1) is cleared. Following a power-on sequence, the boot code is automatically executed and attempts to autobaud to a host. If no autobaud occurs within approximately 400 ms and the SoftICE flag is not set, control will be passed to the user code. A software reset is used to accomplish this control transfer and as a result the SWR bit will remain set. **Therefore the user's code will need to clear the SWR bit in order to access the IAP routines in block 1.** However, caution must be taken when dynamically changing the BSEL bit. Since this will cause different physical memory to be mapped to the logical program address space, the user must avoid clearing the BSEL bit when executing user code within the address range 0000H to 1FFFH.

6.2.2 Power-on reset code execution

At initial power up, the port pins will be in a random state until the oscillator has started and the internal reset algorithm has weakly pulled all pins high. Powering up the device without a valid reset could cause the MCU to start executing instructions from an indeterminate location. Such undefined states may inadvertently corrupt the code in the flash. A system reset will not affect the 1 kB of on-chip RAM while the device is running, however, the contents of the on-chip RAM during power up are indeterminate.

When power is applied to the device, the RST pin must be held high long enough for the oscillator to start up (usually several milliseconds for a low frequency crystal), in addition to two machine cycles for a valid power-on reset. An example of a method to extend the RST signal is to implement a RC circuit by connecting the RST pin to V_{DD} through a 10 µF capacitor and to V_{SS} through an 8.2 kΩ resistor as shown in Figure 4. Note that if an RC circuit is used, provision should be made to ensure the V_{DD} rise time does not exceed 1 ms and the oscillator start-up time does not exceed 10 ms.

For a low frequency oscillator with slow start-up time the reset signal must be extended in order to account for the slow start-up time. This method maintains the necessary relationship between V_{DD} and RST to avoid programming at an indeterminate location, which may cause corruption in the Flash code. The power-on detection is designed to

work during initial power up, before the voltage reaches the brownout detection level. The POF flag in the PCON register is set to indicate an initial power up condition. The POF flag will remain active until cleared by software.

Following a power-on or external reset the P89LV51RB2/RC2/RD2 will force the SWR and BSEL bits (FCF[1:0]) to 00. This causes the boot block to be mapped into the lower 8 kB of code memory and the device will execute the ISP code in the boot block and attempt to autobaud to the host. If the autobaud is successful the device will remain in ISP mode. If, after approximately 400 ms, the autobaud is unsuccessful the boot block code will check to see if the SoftICE flag is set (from a previous programming operation). If the SoftICE flag is set the device will enter SoftICE mode. If the SoftICE flag is cleared, the boot code will execute a software reset causing the device to execute the user code from block 0 starting at address 0000H. Note that an external reset applied to the RST pin has the same effect as a power-on reset.

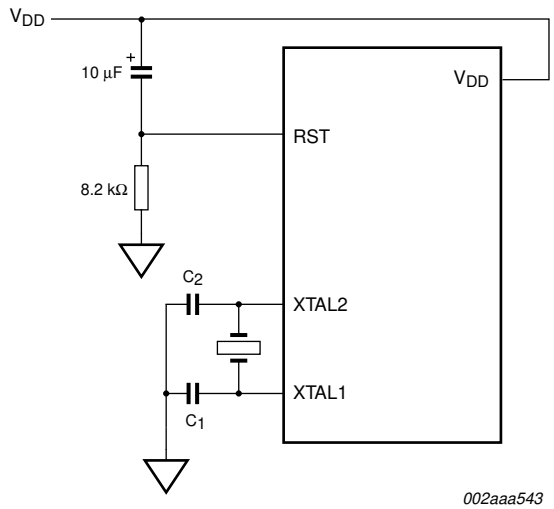

Fig 4. Power-on reset circuit

6.2.3 Software reset

A software reset is executed by changing the SWR bit (FCF.1) from '0' to '1'. A software reset will reset the program counter to address 0000H and force both the SWR and BSEL bits (FCF[1:0]) to 10. This will result in the lower 8 kB of the user code memory being mapped into the user code memory space. Thus the user's code will be executed starting at address 0000H. A software reset will not change bit WDTC.2 or RAM data. Other SFRs will be set to their reset values.

6.2.4 Brownout detect reset

The device includes a brownout detection circuit to protect the system from severe supply voltage fluctuations. The P89LV51RB2/RC2/RD2's brownout detection threshold is 2.35 V. When V_{DD} drops below this voltage threshold, the brownout detect triggers the circuit to generate a brownout interrupt but the CPU still runs until the supplied voltage returns to the brownout detection voltage V_{bo}. The default operation for a brownout detection is to cause a processor reset.

V_{DD} must stay below V_{bo} at least four oscillator clock periods before the brownout detection circuit will respond.

Brownout interrupt can be enabled by setting the EBO bit (IEN1.3). If EBO bit is set and a brownout condition occurs, a brownout interrupt will be generated to execute the program at location 004BH. It is required that the EBO bit is cleared by software after the brownout interrupt is serviced. Clearing EBO bit when the brownout condition is active will properly reset the device. If brownout interrupt is not enabled, a brownout condition will reset the program to resume execution at location 0000H. A brownout detect reset will clear the BSEL bit (FCF.0) but will not change the SWR bit (FCF.1) and therefore will not change the banking of the lower 8 kB of user code memory space.

6.2.5 Watchdog reset

Like a brownout detect reset, the watchdog timer reset will clear the BSEL bit (FCF.0) but will not change the SWR bit (FCF.1) and therefore will not change the banking of the lower 8 kB of user code memory space.

The state of the SWR and BSEL bits after different types of resets is shown in Table 6. This results in the code memory bank selections as shown.

Table 6. Effects of reset sources on bank selection

Reset source	SWR bit result (FCF.1)	BSEL bit result (FCF.0)	Addresses from 0000H to 1FFFH	Addresses above 1FFFH
External reset	0	0	Boot code (in block 1)	User code (in block 0)
Power-on reset				
Watchdog reset	x	0	Retains state of SWR bit. If SWR, BSEL = 00 then uses boot code. If SWR, BSEL = 10 then uses user code.	
Brownout detect reset				
Software reset	1	0	User code (in block 0)	

6.2.6 Data RAM memory

The data RAM has 1024 B of internal memory. The device can also address up to 64 kB for external data memory.

6.2.7 Expanded data RAM addressing

The P89LV51RB2/RC2/RD2 has 1 kB of RAM. See Figure 5 “Internal and external data memory structure” on page 19.

The device has four sections of internal data memory:

1. The lower 128 B of RAM (00H to 7FH) are directly and indirectly addressable.
2. The higher 128 B of RAM (80H to FFH) are indirectly addressable.
3. The special function registers (80H to FFH) are directly addressable only.
4. The expanded RAM of 768 B (00H to 2FFH) is indirectly addressable by the move external instruction (MOVX) and clearing the EXTRAM bit (see ‘Auxiliary function Register’ (AUXR) in Table 4 “Special function registers” on page 11).

Since the upper 128 B occupy the same addresses as the SFRs, the RAM must be accessed indirectly. The RAM and SFRs space are physically separate even though they have the same addresses.

Table 7. AUXR - Auxiliary register (address 8EH) bit allocation

Not bit addressable; reset value 00H.

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	EXTRAM	AO

Table 8. AUXR - Auxiliary register (address 8EH) bit descriptions

Bit	Symbol	Description
7 to 2	-	Reserved for future use. Should be set to '0' by user programs.
1	EXTRAM	Internal/External RAM access using MOVX-@Ri/@DPTR. When '0', core attempts to access internal XRAM with address specified in MOVX instruction. If address supplied with this instruction exceeds on-chip available XRAM, off-chip XRAM is going to be selected and accessed. When '1', every MOVX-@Ri/@DPTR instruction targets external data memory by default.
0	AO	ALE off: disables/enables ALE. AO = 0 results in ALE emitted at a constant rate of $\frac{1}{2}$ the oscillator frequency. In case of AO = 1, ALE is active only during a MOVX or MOVC.

When instructions access addresses in the upper 128 B (above 7FH), the MCU determines whether to access the SFRs or RAM by the type of instruction given. If it is indirect, then RAM is accessed. If it is direct, then an SFR is accessed. See the examples below.

Indirect access:

MOV@R0, #data; R0 contains 90H

Register R0 points to 90H which is located in the upper address range. Data in '#data' is written to RAM location 90H rather than port 1.

Direct access:

MOV90H, #data; write data to P1

Data in '#data' is written to port 1. Instructions that write directly to the address, write to the SFRs.

To access the expanded RAM, the EXTRAM bit must be cleared and MOVX instructions must be used. The extra 768 B of memory is physically located on the chip and logically occupies the first 768 B of external memory (addresses 000H to 2FFH).

When EXTRAM = 0, the expanded RAM is indirectly addressed using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. Accessing the expanded RAM does not affect ports P0, P3.6 ($\overline{WR}$), P3.7 ($\overline{RD}$), or P2. With EXTRAM = 0, the expanded RAM can be accessed as in the following example.

Expanded RAM access (indirect addressing only):

MOVX@DPTR, A DPTR contains 0A0H

DPTR points to 0A0H and data in 'A' is written to address 0A0H of the expanded RAM rather than external memory. Access to external memory higher than 2FFH using the MOVX instruction will access external memory (0300H to FFFFH) and will perform in the same way as the standard 8051, with P0 and P2 as data/address bus, and P3.6 and P3.7 as write and read timing signals.

When EXTRAM = 1, MOVX@Ri and MOVX@DPTR will be similar to the standard 8051. Using MOVX @Ri provides an 8-bit address with multiplexed data on Port 0. Other output port pins can be used to output higher order address bits. This provides external paging capabilities. Using MOVX@DPTR generates a 16-bit address. This allows external addressing up to 64 kB. Port 2 provides the high-order eight address bits (DPH), and Port 0 multiplexes the low-order eight address bits (DPL) with data. Both MOVX@Ri and MOVX@DPTR generates the necessary read and write signals (P3.6, - $\overline{WR}$ and P3.7, - $\overline{RD}$) for external memory use. Table 9 shows external data memory $\overline{RD}$, $\overline{WR}$ operation with EXTRAM bit.

The stack pointer (SP) can be located anywhere within the 256 B of internal RAM (lower 128 B and upper 128 B). The stack pointer may not be located in any part of the expanded RAM.

Table 9. External data memory $\overline{RD}$, $\overline{WR}$ with EXTRAM bit[1]

Register AUXR	MOVX-@DPTR, A or MOVX A, @DPTR		MOVX-@Ri, A or MOVX A, @Ri
	ADDR < 0300H	ADDR ≥ 0300H	ADDR = any
EXTRAM = 0	$\overline{RD}$/$\overline{WR}$ not asserted	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ not asserted
EXTRAM = 1	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ asserted

[1] Access limited to Expanded RAM address within 0 to 0FFH; cannot access 100H to 02FFH.

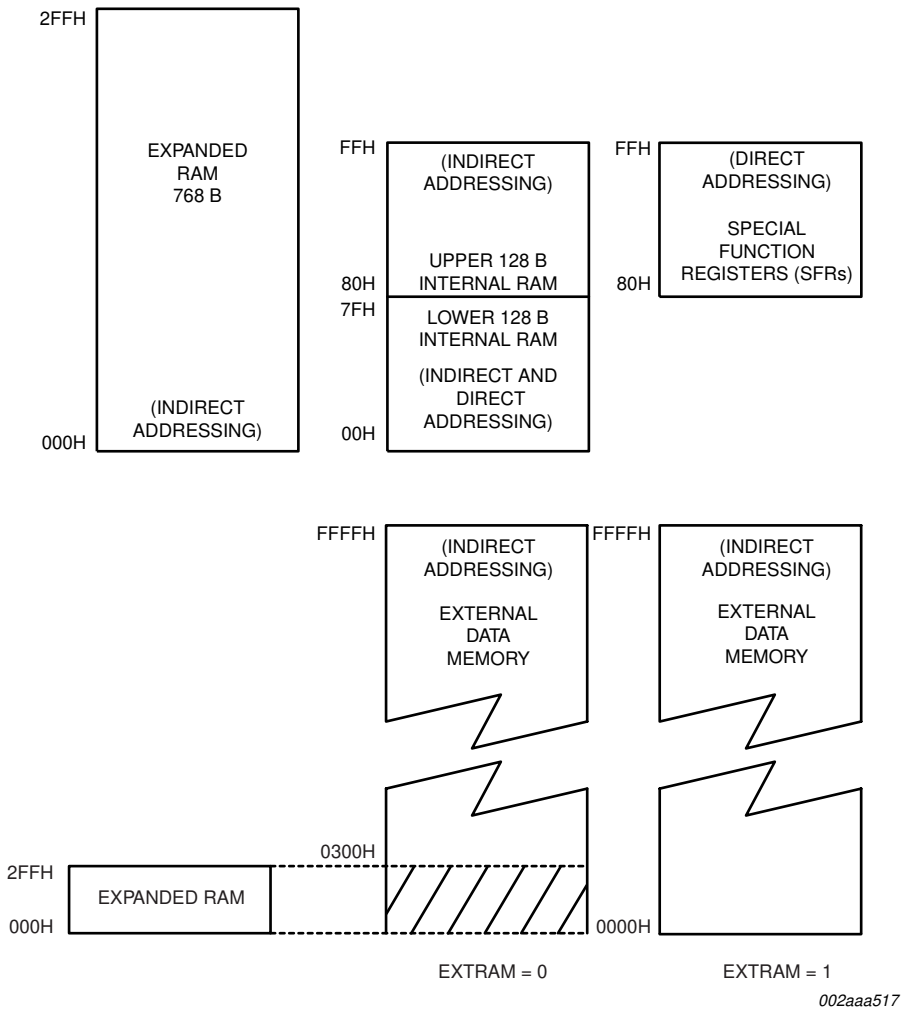

Fig 5. Internal and external data memory structure

6.2.8 Dual data pointers

The device has two 16-bit data pointers. The DPTR Select (DPS) bit in AUXR1 determines which of the two data pointers is accessed. When DPS = 0, DPTR0 is selected; when DPS = 1, DPTR1 is selected. Quickly switching between the two data pointers can be accomplished by a single INC instruction on AUXR1 (see Figure 6).

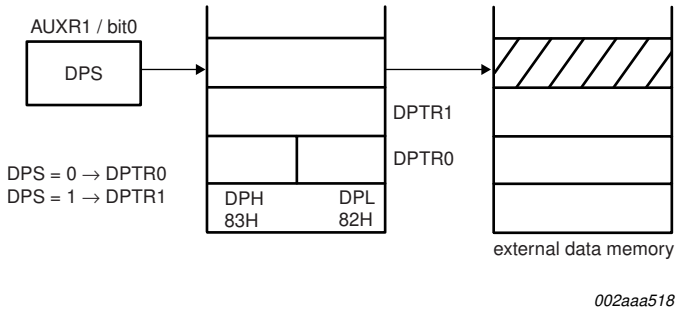

Fig 6. Dual data pointer organization

Table 10. AUXR1 - Auxiliary register 1 (address A2H) bit allocation

Not bit addressable; reset value 00H.

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	GF2	0	-	DPS

Table 11. AUXR1 - Auxiliary register 1 (address A2H) bit descriptions

Bit	Symbol	Description
7 to 4	-	Reserved for future use. Should be set to ‘0’ by user programs.
3	GF2	General purpose user-defined flag.
2	0	This bit contains a hard-wired ‘0’. Allows toggling of the DPS bit by incrementing AUXR1, without interfering with other bits in the register.
1	-	Reserved for future use. Should be set to ‘0’ by user programs.
0	DPS	Data pointer select. Chooses one of two Data Pointers for use by the program. See text for details.

6.3 Flash memory IAP

6.3.1 Flash organization

The P89LV51RB2/RC2/RD2 program memory consists of a 16/32/64 kB block. ISP capability, in a second 8 kB block, is provided to allow the user code to be programmed in-circuit through the serial port. There are three methods of erasing or programming of the flash memory that may be used. First, the flash may be programmed or erased in the end-user application by calling low-level routines through a common entry point (IAP). Second, the on-chip ISP bootloader may be invoked. This ISP bootloader will, in turn, call low-level routines through the same common entry point that can be used by the end-user application. Third, the flash may be programmed or erased using the parallel method by using a commercially available EPROM programmer which supports this device.

6.3.2 Boot block (block 1)

When the microcontroller programs its own flash memory, all of the low level details are handled by code that is contained in block 1. A user program calls the common entry point in the block 1 with appropriate parameters to accomplish the desired operation. Boot block operations include erase user code, program user code, program security bits, etc.

A chip-erase operation can be performed using a commercially available parallel programer. This operation will erase the contents of this boot block and it will be necessary for the user to reprogram this boot block (block 1) with the NXP-provided ISP/IAP code in order to use the ISP or IAP capabilities of this device. Go to http://www.nxp.com/support for questions or to obtain the hex file for this device.

6.3.3 ISP

ISP is performed without removing the microcontroller from the system. The ISP facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the P89LV51RB2/RC2/RD2 through the serial port. This firmware is provided by NXP and embedded within each P89LV51RB2/RC2/RD2 device. The NXP ISP facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area. The ISP function uses five pins (V_{DD}, V_{SS}, TXD, RXD, and RST). Only a small connector needs to be available to interface your application to an external circuit in order to use this feature.

6.3.4 Using ISP

The ISP feature allows for a wide range of baud rates to be used in your application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency. The ISP feature requires that an initial character (an uppercase U) be sent to the P89LV51RB2/RC2/RD2 to establish the baud rate. The ISP firmware provides auto-echo of received characters. Once baud rate initialization has been performed, the ISP firmware will only accept Intel Hex-type records. Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below:

:NNAAAARRDD..DDCC<crlf>

In the Intel Hex record, the ‘NN’ represents the number of data bytes in the record. The P89LV51RB2/RC2/RD2 will accept up to 32 data bytes. The ‘AAAA’ string represents the address of the first byte in the record. If there are zero bytes in the record, this field is often set to 0000. The ‘RR’ string indicates the record type. A record type of ‘00’ is a data record. A record type of ‘01’ indicates the end-of-file mark. In this application, additional record types will be added to indicate either commands or data for the ISP facility.

The maximum number of data bytes in a record is limited to 32 (decimal). ISP commands are summarized in Table 12. As a record is received by the P89LV51RB2/RC2/RD2, the information in the record is stored internally and a checksum calculation is performed. The operation indicated by the record type is not performed until the entire record has been received. Should an error occur in the checksum, the P89LV51RB2/RC2/RD2 will send an ‘X’ out the serial port indicating a checksum error. If the checksum calculation is found to match the checksum in the record, then the command will be executed. In most cases, successful reception of the record will be indicated by transmitting a ‘.’ character out the serial port.

Table 12. ISP hex record formats

Record type	Command/data function
00	Program User Code Memory :nnaaaa00dd..ddcc Where: nn = number of bytes to program aaaa = address dd..dd = data bytes cc = checksum Example: :10000000010203040500607080 9cc
01	End of File (EOF), no operation :xxxxxx01cc Where: xxxxxx = required field but value is a ‘don’t care’ cc = checksum Example: :00000001FF
02	Set SoftICE mode Following the next reset the device will enter the SoftICE mode. Will erase user code memory, and erase device serial number. :00000002cc Where: xxxxxx = required field but value is a ‘don’t care’ cc = checksum Example: :00000002FE

Table 12. ISP hex record formats *…continued*

Record type	Command/data function
03	Miscellaneous Write functions :nnxxxx03ffssddcc Where: nn = number of bytes in the record xxxx = required field but value is a 'don't care' ff = subfunction code ss = selection code dd = data (if needed) cc = checksum Subfunction code = 01 (Erase block 0) ff = 01 Subfunction code = 05 (Program security bit, Double Clock) ff = 05 ss = 01 program security bit ss = 05 program double clock bit Subfunction code = 08 (Erase sector, 128 B) ff = 08 ss = high byte of sector address (A15:8) dd = low byte of sector address (A7, A6:0]= 0) Example: :0300000308E000F2 (erase sector at E000H)
04	Display Device Data or Blank Check :05xxxx04sssseeeeffcc Where 05 = number of bytes in the record xxxx = required field but value is a 'don't care' 04 = function code for display or blank check ssss = starting address, MSB first eeee = ending address, MSB first ff = subfunction 00 = display data 01 = blank check cc = checksum Subfunction codes: Example: :0500000400001FFF00D9 (display from 0000H to 1FFFH)

Table 12. ISP hex record formats *…continued*

Record type	Command/data function
05	Miscellaneous Read functions :02xxxx05ffsscc Where: 02 = number of bytes in the record xxxx = required field but value is a 'don't care' 05 = function code for misc read ffss = subfunction and selection code 0000 = read manufacturer id 0001 = read device id 1 0002 = read boot code version 0700 = read security bit (00 SoftICE serial number match 0 SB 0 Double Clock) cc = checksum Example: :020000050000F9 (display manufacturer id)
06	Direct load of baud rate :02xxxx06HHLLcc Where: 02 = number of bytes in the record xxxx = required field but value is a 'don't care' HH = high byte of timer LL = low byte of timer cc = checksum Example: :02000006FFFFcc (load T2 = FFFF)
07	Reset serial number, erase user code, clear SoftICE mode :xxxxxx07cc Where: xxxxxx = required field but value is a 'don't care' 07 = reset serial number function cc = checksum Example: :00000007F9
08	Verify serial number :nnxxxx08ss..sscc Where: xxxxxx = required field but value is a 'don't care' 08 = verify serial number function ss..ss = serial number contents cc = checksum Example: :03000008010203EF (verify serial number = 010203)

Table 12. ISP hex record formats *…continued*

Record type	Command/data function
09	Write serial number :nnxxxx09ss..sscc Where: xxxxxx = required field but value is a 'don't care' 09 = write serial number function ss..ss = serial number contents cc = checksum Example: :03000009010203EE (write serial number = 010203)
0A	Display serial number :xxxxxx0Acc Where: xxxxxx = required field but value is a 'don't care' 0A = display serial number function cc = checksum Example: :0000000AF6
0B	Reset and run user code :xxxxxx0Bcc Where: xxxxxx = required field but value is a 'don't care' 0B = reset and run user code cc = checksum Example: :0000000BF5

6.3.5 Using the serial number

This device has the option of storing a 31 B serial number along with the length of the serial number (for a total of 32 B) in a non-volatile memory space. When ISP mode is entered, the serial number length is evaluated to determine if the serial number is in use. If the length of the serial number is programmed to either 00H or FFH, the serial number is considered not in use. If the serial number is in use, reading, programming, or erasing of the user code memory or the serial number is blocked until the user transmits a 'verify serial number' record containing a serial number and length that matches the serial number and length previously stored in the device. The user can reset the serial number to all zeros and set the length to zero by sending the 'reset serial number' record. In addition, the 'reset serial number' record will also erase all user code.

6.3.6 IAP method

Several IAP calls are available for use by an application program to permit selective erasing, reading and programming of flash sectors, security bit, configuration bytes, and device id. All calls are made through a common interface, PGM_MTP. The programming functions are selected by setting up the microcontroller's registers before making a call to PGM_MTP at 1FF0H. The IAP calls are shown in Table 13.