Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

P89V660/662/664

8-bit 80C51 5 V low power 16 kB/32 kB/64 kB flash microcontroller with 512 B/1 kB/2 kB RAM, dual I^{2}C-bus, SPI

Rev. 3.1 — 17 October 2011 Product data sheet

1. General description

The P89V660/662/664 are 80C51 microcontrollers with 16 kB/32 kB/64 kB flash and 512 B/1 kB/2 kB of data RAM. These devices are designed to be drop-in and software compatible replacements for the P89C660/662/664 devices. Both the In-System Programming (ISP) and In-Application Programming (IAP) boot codes are upward compatible.

Additional features of the P89V660/662/664 devices when compared to the P89C660/662/664 devices are the inclusion of a secondary 100 kHz byte-wide I^{2}C-bus interface, an SPI interface, four addition I/O pins (Port 4), and the ability to erase code memory in 128-byte pages.

The IAP capability combined with the 128-byte page size allows for efficient use of the code memory for non-volatile data storage.

2. Features and benefits

2.1 Principal features

- Dual 100 kHz byte-wide I^{2}C-bus interfaces
- 128-byte page erase for efficient use of code memory as non-volatile data storage
- 0 MHz to 40 MHz operating frequency in 12x mode, 20 MHz in 6x mode
- 16 kB/32 kB/64 kB of on-chip flash user code memory with ISP and IAP
- 512 B/1 kB/2 kB RAM
- SPI (Serial Peripheral Interface) and enhanced UART
- PCA (Programmable Counter Array) with PWM and Capture/Compare functions
- Three 16-bit timers/counters
- Four 8-bit I/O ports, one 4-bit I/O port
- WatchDog Timer (WDT)

2.2 Additional features

- 30 ms page erase, 150 ms block erase
- Support for 6-clock (default) or 12-clock mode selection via ISP or parallel programmer
- PLCC44 and TQFP44 packages
- Ten interrupt sources with four priority levels
- Second DPTR register
- Low EMI mode (ALE inhibit)
- Power-down mode with external interrupt wake-up

- Idle mode

2.3 Comparison to the P89C660/662/664 devices

- **SPI interface**. The P89V660/662/664 devices include an SPI interface that was not present on the P89C660/662/664 devices.
- **Dual I2C-bus interfaces**. The P89V660/662/664 devices have two I^{2}C-bus interfaces. The P89C660/662/664 devices have one.
- **More I/O pins**. The P89V660/662/664 devices have an additional four-bit I/O port, Port 4.
- The **6x/12x mode** on theP89V660/662/664 devices is **programmable** and erasable **using ISP** and IAP as well as parallel programmer mode. The P89C660/662/664 devices could only be switched using parallel programmer mode.
- **Smaller block sizes**. The smallest block size on the P89C660/662/664 devices was 8 kB. The P89V660/662/664 devices have a page size of 128 B. These small pages can be erased and reprogrammed using IAP function calls making use of the code memory for non-volatile data storage practical. Each page erase is 30 ms or less. The IAP and ISP code in P89V660/662/664 devices support these 128-byte page operations. In addition, the IAP and ISP code uses multiple page erase operations to emulate the erasing of the larger block sizes (8 kB and 16 kB to maintain firmware compatibility).
- **Status bit versus Status byte**. The P89V660/662/664 devices used a Status byte to control the automatic entry into ISP mode following a reset. On the P89V660/662/664 devices this has changed to a single Status bit. Since the ISP entry was based on the zero/non-zero value of the Status byte this is an almost identical operation on the P89V660/662/664 devices.
- **Faster block erase**. The erase time for the entire user code memory of the P89V660/662/664 devices is 150 ms.

3. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
P89V662FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V662FBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89V664FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V664FBC	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
P89V662FA	32 kB	−40 °C to +85 °C	0 MHz to 40 MHz
P89V662FBC	32 kB		
P89V664FA	64 kB		
P89V664FBC	64 kB		

4. Block diagram

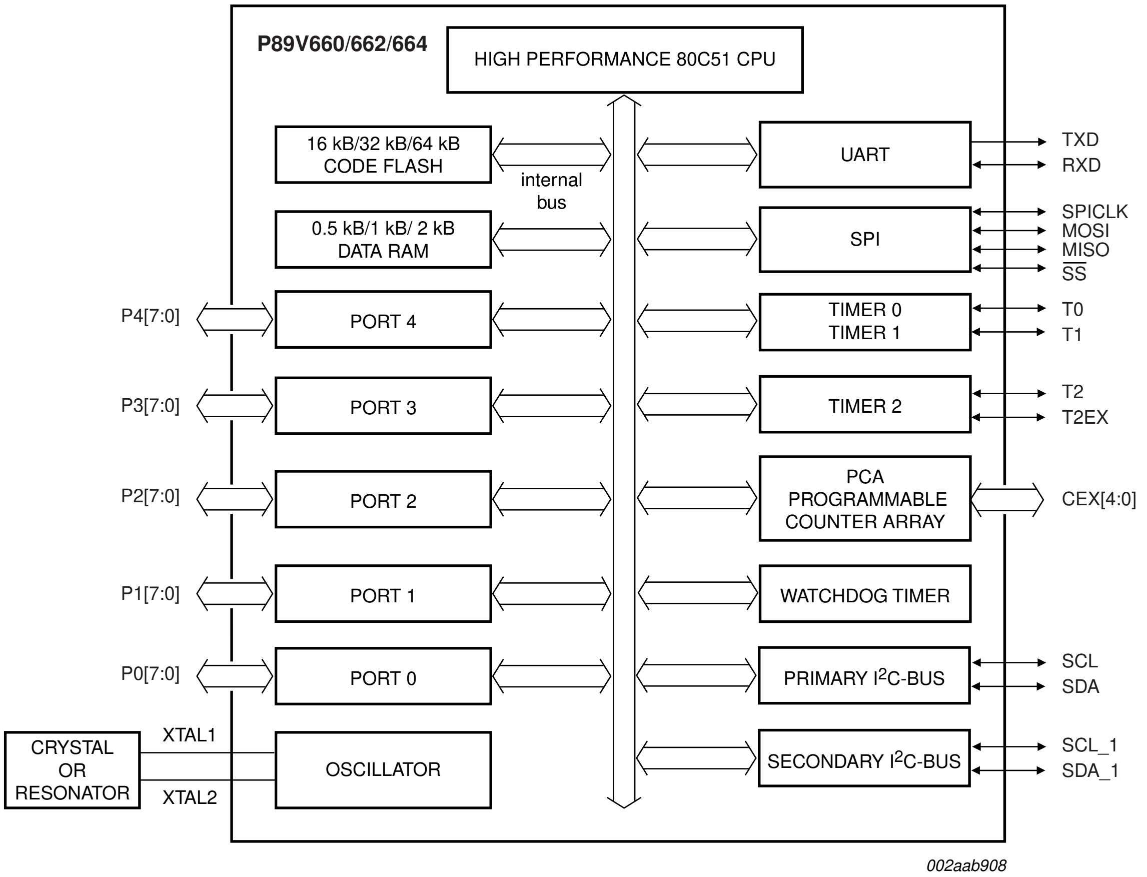

Fig 1. Block diagram

5. Pinning information

5.1 Pinning

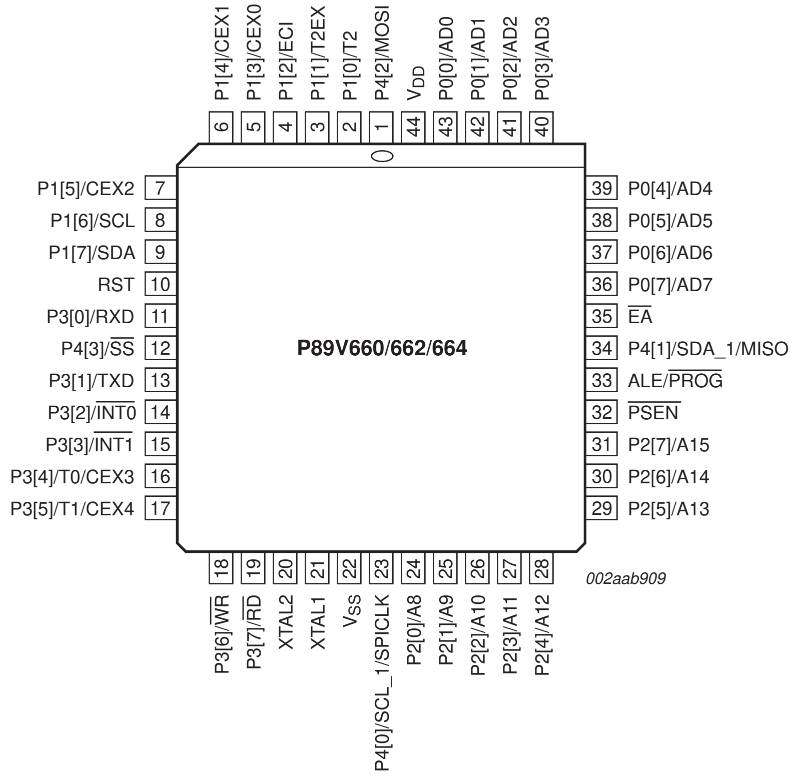

Fig 2. PLCC44 pin configuration

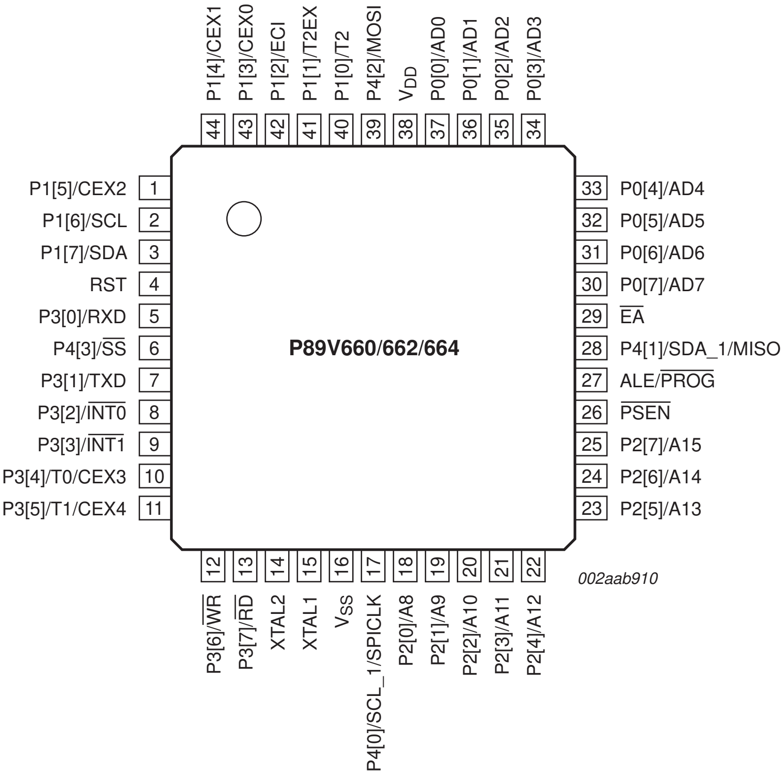

Fig 3. TQFP44 pin configuration

5.2 Pin description

Table 3. 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 making the transition to ‘1’s. External pull-ups are required 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][1]			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 high current drive of 16 mA.
P1[0]/T2	40	2	I/O	**P1[0]** — Port 1 bit 0.
			I	**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
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.
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.

Table 3. Pin description *...continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
P1[4]/CEX1	44	6	I/O	**P1[4]** — Port 1 bit 4.
			I/O	**CEX1** — Capture/compare external I/O for PCA Module 1
P1[5]/CEX2	1	7	I/O	**P1[5]** — Port 1 bit 5.
			I/O	**CEX2** — Capture/compare external I/O for PCA Module 2
P1[6]/SCL	2	8	I/O	**P1[6]** — Port 1 bit 6.
			I/O	**SCL** — I^{2}C-bus serial clock input/output
P1[7]/SDA	3	9	I/O	**P1[7]** — Port 1 bit 7.
			I/O	**SDA** — I^{2}C-bus serial data input/output
P2[0] to P2[7][1]			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 making the transition to '1's.
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.
P2[7]/A15	25	31	I/O	**P2[7]** — Port 2 bit 7.
			O	**A15** — Address bit 15.
P3[0] to P3[7][1]			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.
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.

Table 3. Pin description *...continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
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/CEX3	10	16	I/O	**P3[4]** — Port 3 bit 4.
			I	**T0** — External count input to Timer/Counter 0.
			I/O	**CEX3** — Capture/compare external I/O for PCA Module 3.
P3[5]/T1/CEX4	11	17	I/O	**P3[5]** — Port 3 bit 5.
			I	**T1** — External count input to Timer/Counter 1
			I/O	**CEX4** — Capture/compare external I/O for PCA Module 4
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.
P4[0] to P4[3][1]			I/O with internal pull-up	**Port 4**: Port 4 is a 4-bit bidirectional I/O port with internal pull-ups. Port 4 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 4 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups.
P4[0]/SCL_1/ SPICLK	17	23	I/O	**P4[0]** — Port 4 bit 0.
			I/O	**SCL_1** — Second I^2C-bus serial clock input/output
			I/O	**SPICLK** — Serial clock input/output for SPI
P4[1]/SDA_1/ MISO	28	34	I/O	**P4[1]** — Port 4 bit 1.
			I/O	**SDA_1** — Second I^2C-bus serial data input/output
			I/O	**MISO** — Master input/slave output for SPI
P4[2]/MOSI	39	1	I/O	**P4[2]** — Port 4 bit 2.
			I/O	**MOSI** — Master output/slave input for SPI
P4[3]/$\overline{\text{SS}}$	6	12	I	**P4[3]** — Port 4 bit 3.
			I	**$\overline{\text{SS}}$** — Slave select input for SPI
$\overline{\text{PSEN}}$	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.
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.
$\overline{\text{EA}}$	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.

Table 3. Pin description *…continued*

Symbol	Pin		Type	Description
	TQFP44	PLCC44		
ALE/$\overline{\text{PROG}}$	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 ($\overline{\text{PROG}}$) for flash programming. Normally the ALE[2] is emitted at a constant rate of $\frac{1}{6}$ the crystal frequency[3] and can be used for external timing and clocking. One ALE pulse is skipped during each access to external data memory. However, if AO is set to ‘1’, ALE is disabled.
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] Port 1, 2, 3, and 4 enter the bidirectional state (except the I^2C pins) with a weak pull-up after reset. In this state, the pins can be used as inputs or outputs. See the *80C51 Family Hardware Description* for details of the port structure.

A reset does not assert the strong pull-up for two clock cycles for these ports which normally occurs when the port transitions from a LOW to a HIGH state. You must first write a zero, then a logic one to enable the strong pull-up for two clock cycles.

[2] 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 add a pull-up resistor of 3 kΩ to 50 kΩ to V_{DD}, e.g., for ALE pin.

[3] 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 Special Function Registers (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	C2H	-	ECOM_0	CAPP_0	CAPN_0	MAT_0	TOG_0	PWM_0	ECCF_0
CCAPM1	Module 1 mode	C3H	-	ECOM_1	CAPP_1	CAPN_1	MAT_1	TOG_1	PWM_1	ECCF_1
CCAPM2	Module 2 mode	C4H	-	ECOM_2	CAPP_2	CAPN_2	MAT_2	TOG_2	PWM_2	ECCF_2
CCAPM3	Module 3 mode	C5H	-	ECOM_3	CAPP_3	CAPN_3	MAT_3	TOG_3	PWM_3	ECCF_3
CCAPM4	Module 4 mode	C6H	-	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	C0H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
CH	PCA Counter HIGH	F9H								
CL	PCA Counter LOW	E9H								
CMOD	PCA Counter mode	C1H	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 Special Function Registers (SFRs) that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses							
			MSB							LSB
		Bit address	AF	AE	AD	AC	AB	AA	A9	A8
IEN0*	Interrupt Enable 0	A8H	EA	EC	ES1	ES0	ET1	EX1	ET0	EX0
		Bit address	EF	EE	ED	EC	EB	EA	E9	E8
IEN1*	Interrupt Enable 1	E8H	-	-	-	-	-	ES3	ES2	ET2
		Bit address	BF	BE	BD	BC	BB	BA	B9	B8
IP0*	Interrupt Priority 0	B8H	PT2	PPC	PS1	PS0	PT1	PX1	PT0	PX0
IP0H	Interrupt Priority 0 HIGH	B7H	PT2H	PPCH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0H
		Bit address	FF	FE	FD	FC	FB	FA	F9	F8
IP1*	Interrupt Priority 1	91H	-	-	-	-	-	-	PS3	PS2
IP1H	Interrupt Priority 1 HIGH	92H	-	-	-	-	-	-	PS3	PS2
		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	SDA	SCL	CEX2	CEX1	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	$\overline{RD}$	$\overline{WR}$	CEX4/T1	CEX3/T0	$\overline{INT1}$	$\overline{INT0}$	TXD	RXD
P4	Port 4	A1H	-	-	-	-	$\overline{SS}$	MOSI	MISO/ SDA_1	SPICLK/ SCL_1
PCON	Power Control Register	87H	SMOD1	SMOD0	-	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
S0CON*	Serial Port Control	98H	SM0/FE_	SM1	SM2	REN	TB8	RB8	TI	RI
S0BUF	Serial Port Data Buffer Register	99H								
SADDR	Serial Port Address Register	A9H								

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

** indicates Special Function Registers (SFRs) that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses							
			MSB							LSB
SADEN	Serial Port Address Enable	B9H								
		Bit address	87[1]	86[1]	85[1]	84[1]	83[1]	82[1]	81[1]	80[1]
SPCR	SPI Control Register	D5H	SPIE	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
SPSR	SPI Configuration Register	AAH	SPIF	WCOL	-	-	-	-	-	-
SPDAT	SPI Data	86H								
SP	Stack Pointer	81H								
S1DAT	I^{2}C-bus data register	DAH								
S1ADR	I^{2}C-bus slave address register	DBH	S1ADR.6	S1ADR.5	S1ADR.4	S1ADR.3	S1ADR.2	S1ADR.1	S1ADR.0	S1GC
S1STA	I^{2}C-bus status register	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0
S1CON*	I^{2}C-bus control register	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
S2DAT	I^{2}C-bus data register	E2H								
S2ADR	I^{2}C-bus slave address register	E3H	S2ADR.6	S2ADR.5	S2ADR.4	S2ADR.3	S2ADR.2	S2ADR.1	S2ADR.0	S2GC
S2STA	I^{2}C-bus status register	E1H	SC24	SC23	SC22	SC21	SC20	0	0	0
S2CON*	I^{2}C-bus control register	F8H	CR22	ENS21	STA2	STO2	SI2	AA2	CR21	CR20
		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	GATE	C/T	M1	M0	GATE	C/T	M1	M0
WDTRST	WatchDog Timer Reset	A6H								

[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 various P89V660/662/664 memory spaces are as follows:

- DATA

  128 B of internal data memory space (00H:7FH) accessed via direct or indirect addressing, using instructions other than MOVX and MOVC. All or part of the Stack may be in this area.

- IDATA

  Indirect Data. 256 B of internal data memory space (00H:FFH) accessed via indirect addressing using instructions other than MOVX and MOVC. All or part of the Stack may be in this area. This area includes the DATA area and the 128 B immediately above it.

- SFR

  Special Function Registers. Selected CPU registers and peripheral control and status registers, accessible only via direct addressing.

- XDATA

  'External' Data or Auxiliary RAM. Duplicates the classic 80C51 64 kB memory space addressed via the MOVX instruction using the DPTR, R0, or R1. The P89V660/662/664 have 256/768/1792 B of on-chip XDATA memory.

- CODE

  64 kB of Code memory space, accessed as part of program execution and via the MOVC instruction. The P89V660/662/664 have 16/32/64 kB of on-chip Code memory.

6.2.1 Expanded data RAM addressing

The P89V660/662/664 have 512 B/1 kB/2 kB of RAM. See Figure 4.

To access the expanded RAM, the EXTRAM bit must be set and MOVX instructions must be used. The extra memory is physically located on the chip and logically occupies the first bytes of external memory (addresses 000H to 0FFH/2FFH/6FFH).

Table 5. 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

When EXTRAM = 1, 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] ($\overline{\text{WR}}$), P3[7] ($\overline{\text{RD}}$), or P2. With EXTRAM = 1, the expanded RAM can be accessed as in the following example.

Expanded RAM Access (Indirect Addressing only):

```
MOVX@DPTR, A; DPTR contains 0A0H
```

The DPTR points to location 0A0H and the data in the accumulator is written to address 0A0H of the expanded RAM rather than off-chip external memory. Access to EXTRAM addresses that are not present on the device (above 0FFH for the 89V660, above 2FFH for the 89V662, above 6FFH for the 89V664) will access external off-chip memory 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.

Table 6. AUXR - Auxiliary register (address 8EH) bit description

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 ‘1’, accesses internal XRAM with address specified in MOVX instruction. If address supplied with this instruction exceeds on-chip available XRAM, off-chip RAM is accessed. When ‘0’, every MOVX instructions targets external data memory by default.
0	AO	ALE off: disables/enables ALE. AO = 0 results in ALE emitted at a constant rate of $^1\!/_2$ the oscillator frequency. In case of AO = 1, ALE is active only during a MOVX or MOVC.

When EXTRAM = 0, 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 the 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 7 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 7. External data memory $\overline{RD}$, $\overline{WR}$ with EXTRAM bit

AUXR	MOVX @DPTR, A or MOVX A, @DPTR		MOVX @Ri, A or MOVX A, @Ri
	ADDR < 0100H (89V660)	ADDR ≥ 0100H (89V660)	ADDR = any
	ADDR < 0300H (89V662)	ADDR ≥ 0300H (89V662)	
	ADDR < 0700H (89V664)	ADDR ≥ 0700H (89V664)	
EXTRAM = 0	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ asserted
EXTRAM = 1	$\overline{RD}$/$\overline{WR}$ not asserted	$\overline{RD}$/$\overline{WR}$ asserted	$\overline{RD}$/$\overline{WR}$ not asserted

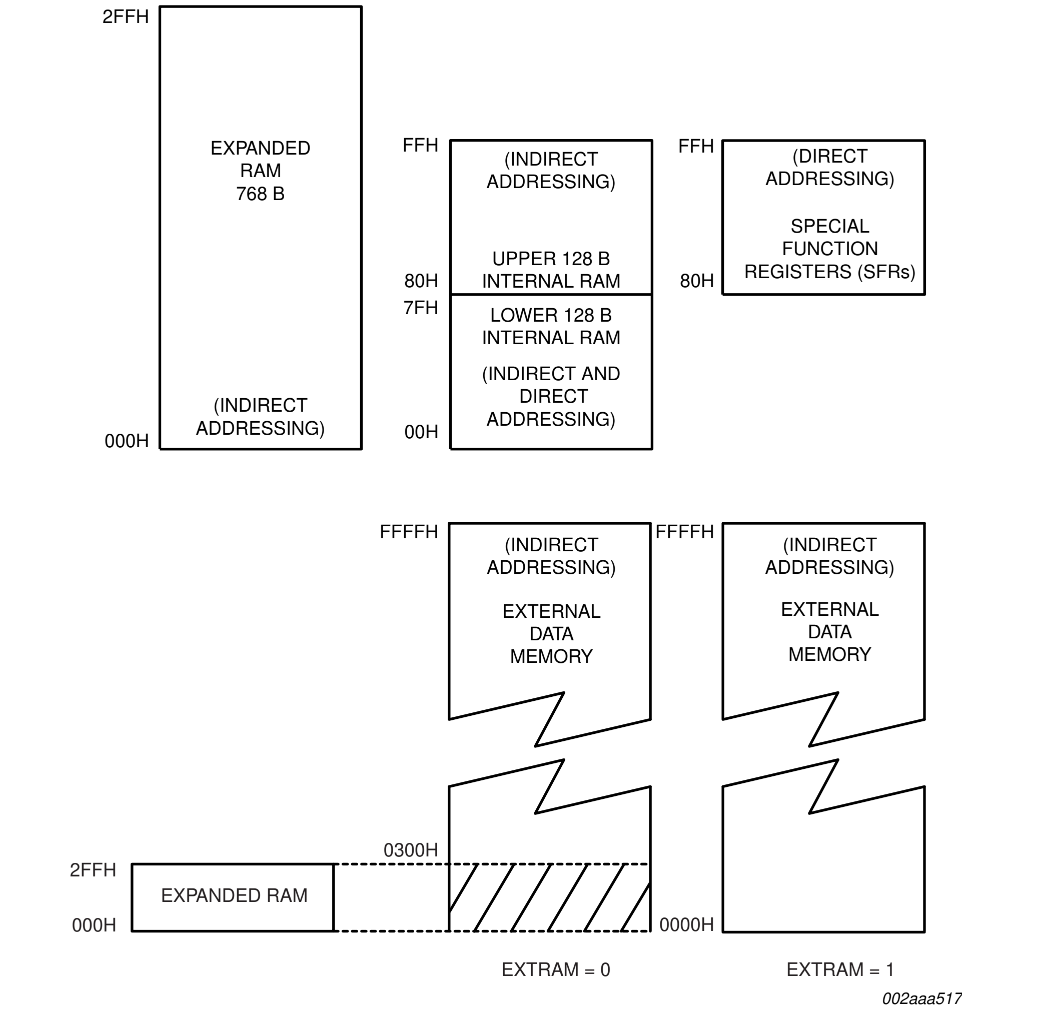

Fig 4. Internal and external data memory structure

6.2.2 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 5).

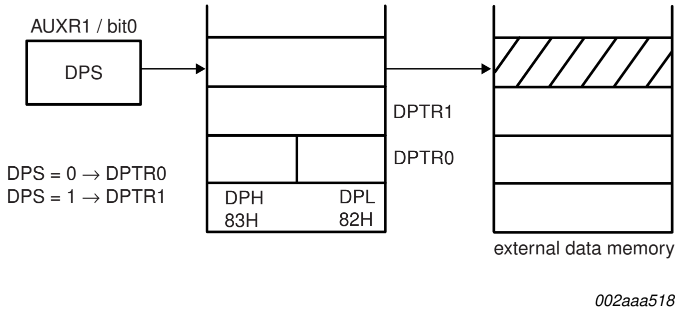

Fig 5. Dual data pointer organization

Table 8. 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 9. AUXR1 - Auxiliary register 1 (address A2H) bit description

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.

6.2.3 Reset

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 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 6.

During initial power 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 reset condition, under normal conditions, the MCU will start executing code from address 0000H in the user’s code memory. However if either the $\overline{\text{PSEN}}$ pin was low when reset was exited, or the Status Bit was set = 1, the MCU will start executing code from the boot address. The boot address is formed using the value of the boot vector as the high byte of the address and 00H as the low byte.

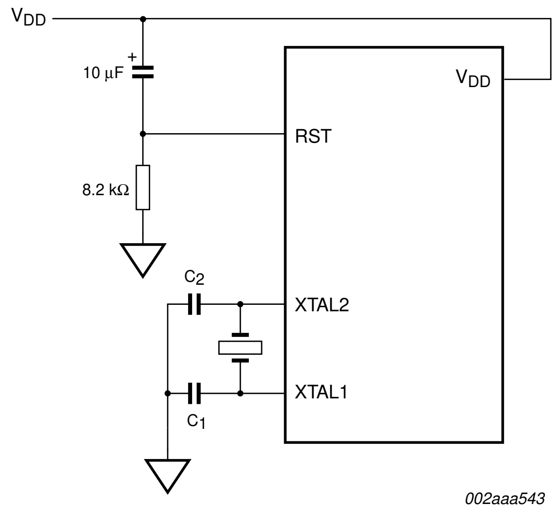

Fig 6. Power-on reset circuit

6.3 Flash memory

6.3.1 Flash organization

The P89V660/662/664 program memory consists of a 16/32/64 kB block for user code. The flash can be read or written in bytes and can be erased in 128 pages. A chip erase function will erase the entire user code memory and its associated security bits. There are three methods of erasing or programming the flash memory that may be used. First, the flash may be programmed or erased in the end-user application by calling LOW-state routines through a common IAP entry point. Second, the on-chip ISP bootloader may be invoked. This ISP bootloader will, in turn, call LOW-state 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 Features

- Flash internal program memory with 128-byte page erase.
- Internal Boot block, containing LOW-state IAP routines available to user code.
- Boot vector allows user-provided flash loader code to reside anywhere in the flash memory space, providing flexibility to the user.
- Default loader providing ISP via the serial port, located in upper end of program memory.
- Programming and erase over the full operating voltage range.
- Read/Programming/Erase using ISP/IAP.
- Programming with industry-standard commercial programmers.
- 10000 typical erase/program cycles for each byte.
- 100 year minimum data retention.

6.3.3 Boot block

When the microcontroller programs its own flash memory, all of the low level details are handled by code (bootloader) that is contained in a Boot block. A user program calls the common entry point in the Boot block with appropriate parameters to accomplish the desired operation. Boot block operations include erase user code, program user code, program security bits, chip erase, etc. The Boot block logically overlays the program memory space from FC00H to FFFFH, when it is enabled. The Boot block may be disabled on-the-fly so that the upper 1 kB of user code is available to the user's program.

6.3.4 Power-on reset code execution

The P89V660/662/664 contains two special flash elements: the Boot Vector and the Boot Status bit. Following reset, the P89V660/662/664 examines the contents of the Boot Status bit. If the Boot Status bit is set to zero, power-up execution starts at location 0000H, which is the normal start address of the user's application code. When the Boot Status bit is set to a value other than zero, the contents of the Boot Vector are used as the high byte of the execution address and the low byte is set to 00H

Table 10 shows the factory default Boot Vector setting for this device. A factory-provided bootloader is pre-programmed into the address space indicated and uses the indicated boot loader entry point to perform ISP functions.

Table 10. Default boot vector values and ISP entry points

Device	Default boot vector	Default bootloader entry point	Default bootloader code range
P89V660/662/664	FCH	FC00H	FC00H to FFFFH

6.3.5 Hardware activation of the bootloader

The bootloader can also be executed by forcing the device into ISP mode during a power-on sequence. This has the same effect as having a non-zero status byte. This allows an application to be built that will normally execute user code but can be manually forced into ISP operation. If the factory default setting for the boot vector (FCH) is changed, it will no longer point to the factory pre-programmed ISP bootloader code. After programming the flash, the status byte should be programmed to zero in order to allow execution of the user's application code beginning at address 0000H.

6.3.6 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 P89V660/662/664 through the serial port. This firmware is provided by NXP and embedded within each P89V660/662/664 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.

6.3.7 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 P89V660/662/664 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 P89V660/662/664 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 11. As a record is received by the P89V660/662/664, 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 P89V660/662/664 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 11. 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: :090000000102030405060708090CA
01	End of File (EOF), no operation :xxxxxx01cc Where: xxxxxx = required field but value is a 'don't care' cc = checksum Example: :00000001FF
02	not used

Table 11. 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 Blocks) ff = 01 ss = block code, as shown below block 0, 0k to 8k, 00H block 1, 8k to 16k, 20H block 0, 16k to 32k, 40H block 0, 32k to 48k, 80H block 0, 48k to 64k, C0H Subfunction code = 04 (Erase Boot vector and Status Bit) ff = 04 ss = don’t care Subfunction code = 05 (Program security bits) ff = 05 ss = 00 program security bit 1 ss = 01 program security bit 2 ss = 02 program security bit 3 Subfunction code = 06 (Program Status bit, Boot vector, 6x/12x bit) ff = 06 dd = data (for Boot vector) ss = 00 program Status bit ss = 01 program Boot vector ss = 02 program 6x/12x bit Subfunction code = 07 (Chip Erase) Erases code memory and security bits, programs default Boot vector and Status bit ff = 07 Subfunction code = 08 (Erase page, 128 B) ff = 08 ss = high byte of page address (A[15:8]) dd = low byte of page address (A[7:0]) Example: :0300000308E000F2 (erase page at E000H)

Table 11. ISP hex record formats *…continued*

Record type	Command/data function
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)
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 device id 2 0003 = read 6x/12x bit (bit 7 = 1 is 6x, bit 7 = 0 is 12x) 0080 = read boot code version 0700 = read security bits 0701 = read Status bit 0702 = read Boot vector 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 T2 LL = low byte of timer T2 cc = checksum Example: :02000006FFFFcc (load T2 = FFFF)

All information provided in this document is subject to legal disclaimers.

6.3.8 IAP method

Several IAP calls are available for use by an application program to permit selective erasing, reading and programming of flash pages, security bits, security bits, Status bit, 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 FFF0H. The IAP calls are shown in Table 12.

Table 12. IAP function calls

IAP function	IAP call parameters

Read Id	**Input parameters:**
	R1 = 00H or 80H (WDT feed)
	DPH = 00H
	DPL = 00H = manufacturer id
	DPL = 01H = device id 1
	DPL = 02H = device id 2
	DPL = 03H = 6x/12x bit (bit 7 = 1 = 6x)
	DPL = 80H = ISP version number
	Return parameter(s):
	ACC = requested parameter

Erase 8 kB/16 kB code block	**Input parameters:**
	R1 = 01H or 81H (WDT feed)
	DPL = 00H, block 0, 0 kB to 8 kB
	DPL = 20H, block 1, 8 kB to 16 kB
	DPL = 40H, block 2, 16 kB to 32 kB
	DPL = 80H, block 3, 32 kB to 48 kB
	DPL = C0H, block 4, 48 kB to 64 kB
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

Program User Code	**Input parameters:**
	R1 = 02H or 82H (WDT feed)
	DPH = memory address MSB
	DPL = memory address LSB
	ACC = byte to program
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

Read User Code	**Input parameters:**
	R1 = 03H or 83H (WDT feed)
	DPH = memory address MSB
	DPL = memory address LSB
	Return parameter(s):
	ACC = device data

Table 12. IAP function calls *…continued*

IAP function	IAP call parameters

Erase Status bit and Boot vector	**Input parameters:**
	R1 = 04H or 84H (WDT feed)
	DPL = don't care
	DPH = don't care
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

Program Security bits	**Input parameters:**
	R1 = 05H or 85H (WDT feed)
	DPL = 00H = security bit 1
	DPL = 01H = security bit 2
	DPL = 02H = security bit 3
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

Program Status bit, Boot vector, 6x/12x bit	**Input parameters:**
	R1 = 06H or 86H (WDT feed)
	DPL = 00H = program Status bit
	DPL = 01H = program Boot vector
	DPL = 02H = 6x/12x bit
	ACC = Boot vector value to program
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

Read Security bits, Status bit, Boot vector	**Input parameters:**
	ACC = 07H or 87H (WDT feed)
	DPL = 00H = security bits
	DPL = 01H = Status bit
	DPL = 02H = Boot vector
	Return parameter(s):
	ACC = 00 SoftICE S/N-match 0 SB 0 DBL_CLK

Erase page	**Input parameters:**
	R1 = 08H or 88H (WDT feed)
	DPH = page address high byte
	DPL = page address low byte
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail

6.4 I^{2}C-bus interface

The I^{2}C-bus uses two wires, Serial Clock (SCL) and Serial Data (SDA) to transfer information between devices connected to the bus, and has the following features:

- Bidirectional data transfer between masters and slaves

- Multimaster bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- The I^{2}C-bus may be used for test and diagnostic purposes

A typical I^{2}C-bus configuration is shown in Figure 7. Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I^{2}C-bus:

- Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
- Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned. The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I^{2}C-bus will not be released.

The P89V660/662/664 device provides two byte-oriented I^{2}C-bus interfaces. For simplicity, the description in this text is written for the primary interface. However, unless otherwise noted, the description applies to the secondary I^{2}C-bus interface with consideration given to the SFR's addresses for the secondary interface. Please note that the secondary I^{2}C-bus interface uses quasi-bidirectional I/O pins instead of open-drain pins. The interface has four operation modes: Master Transmitter mode, Master Receiver mode, Slave Transmitter mode and Slave Receiver mode

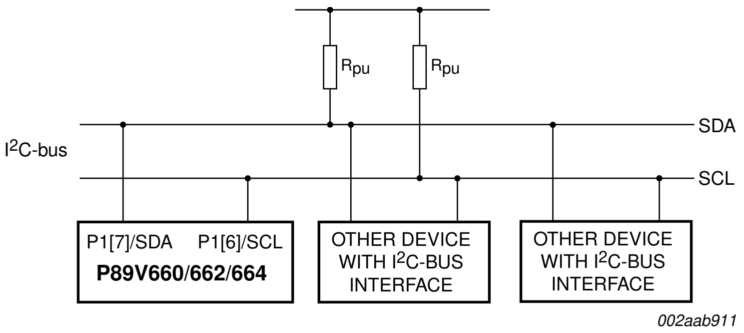
Rpu
Rpu
I2C-bus
SDA
SCL
P1[7]/SDA
P1[6]/SCL
P89V660/662/664
OTHER DEVICE WITH I2C-BUS INTERFACE
OTHER DEVICE WITH I2C-BUS INTERFACE
002aab911

Fig 7. I^{2}C-bus configuration

The P89V660/662/664 CPU interfaces with the I^{2}C-bus through four Special Function Registers (SFRs): S1CON (primary I^{2}C-bus Control Register), S1DAT (primary I^{2}C-bus Data Register), S1STA (primary I^{2}C-bus Status Register), and the S1ADR (primary I^{2}C-bus Slave Address Register).