USER MANUAL

P89LPC912/913/914

8-bit microcontrollers with accelerated two-clock 80C51 core
1KB 3V Low-Power byte-eraseable Flash with 128 Byte RAM

2003 Dec 8

Philips
Semiconductors

PHILIPS

Table of Contents

List of Figures

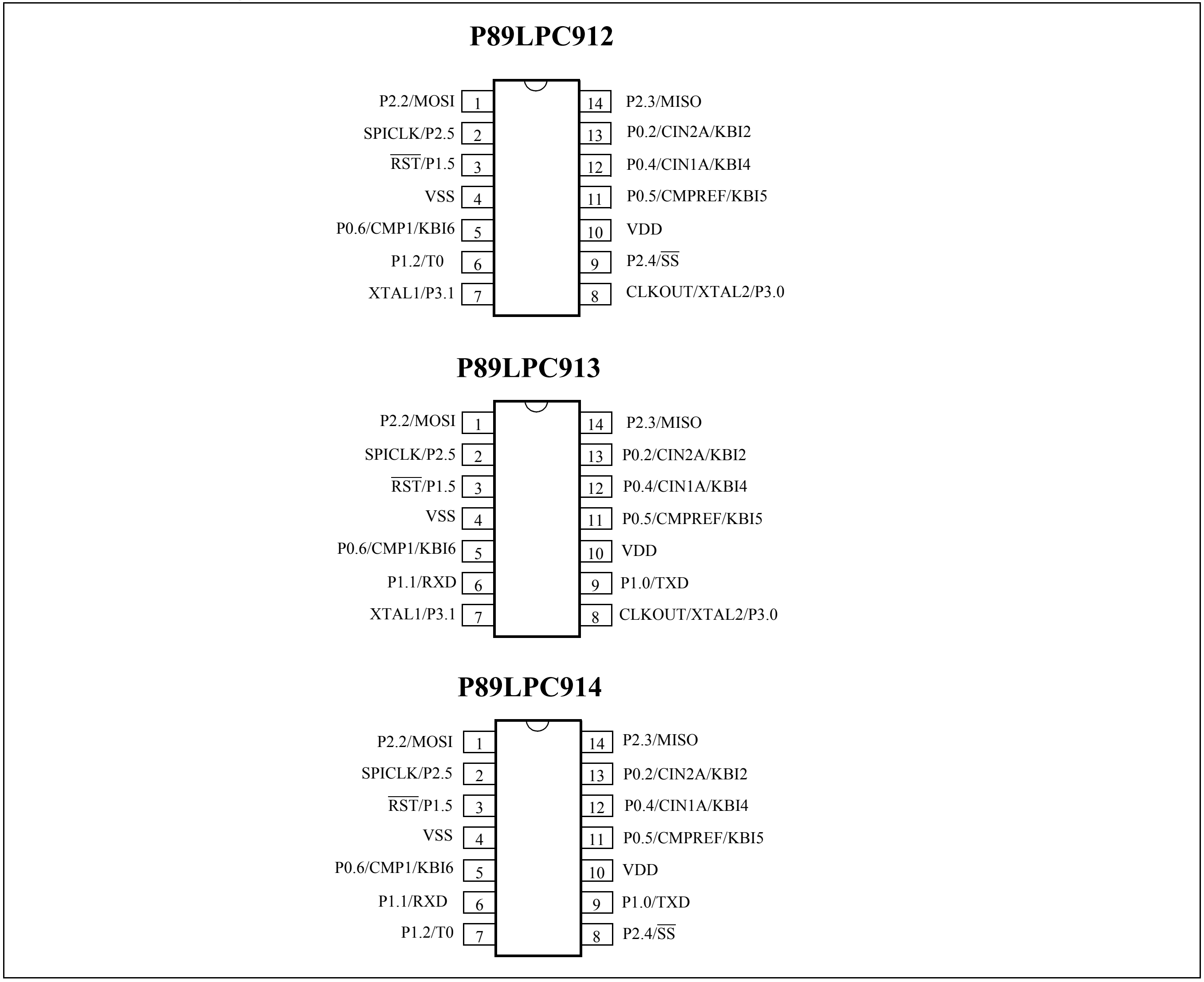
List of Figures

1. GENERAL DESCRIPTION

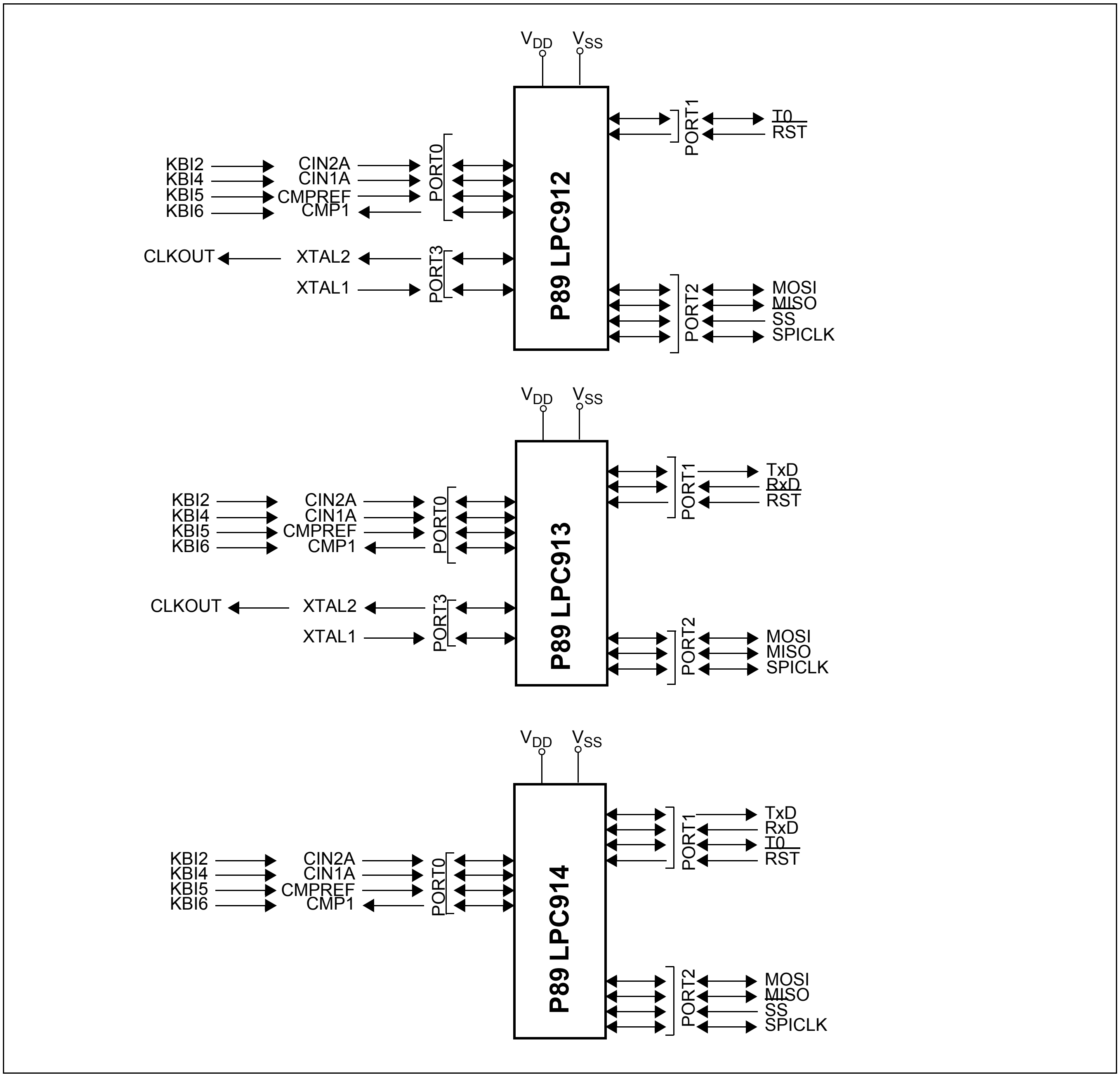
The **P89LPC912/913/914** is a single-chip microcontroller designed for applications demanding high-integration, low cost solutions over a wide range of performance requirements. The **P89LPC912/913/914** is based on a high performance processor architecture that executes instructions six times the rate of standard 80C51 devices. Many system level functions have been incorporated into the **P89LPC912/913/914** in order to reduce component count, board space, and system cost.

PIN CONFIGURATION

14-pin TSSOP package



LOGIC SYMBOL

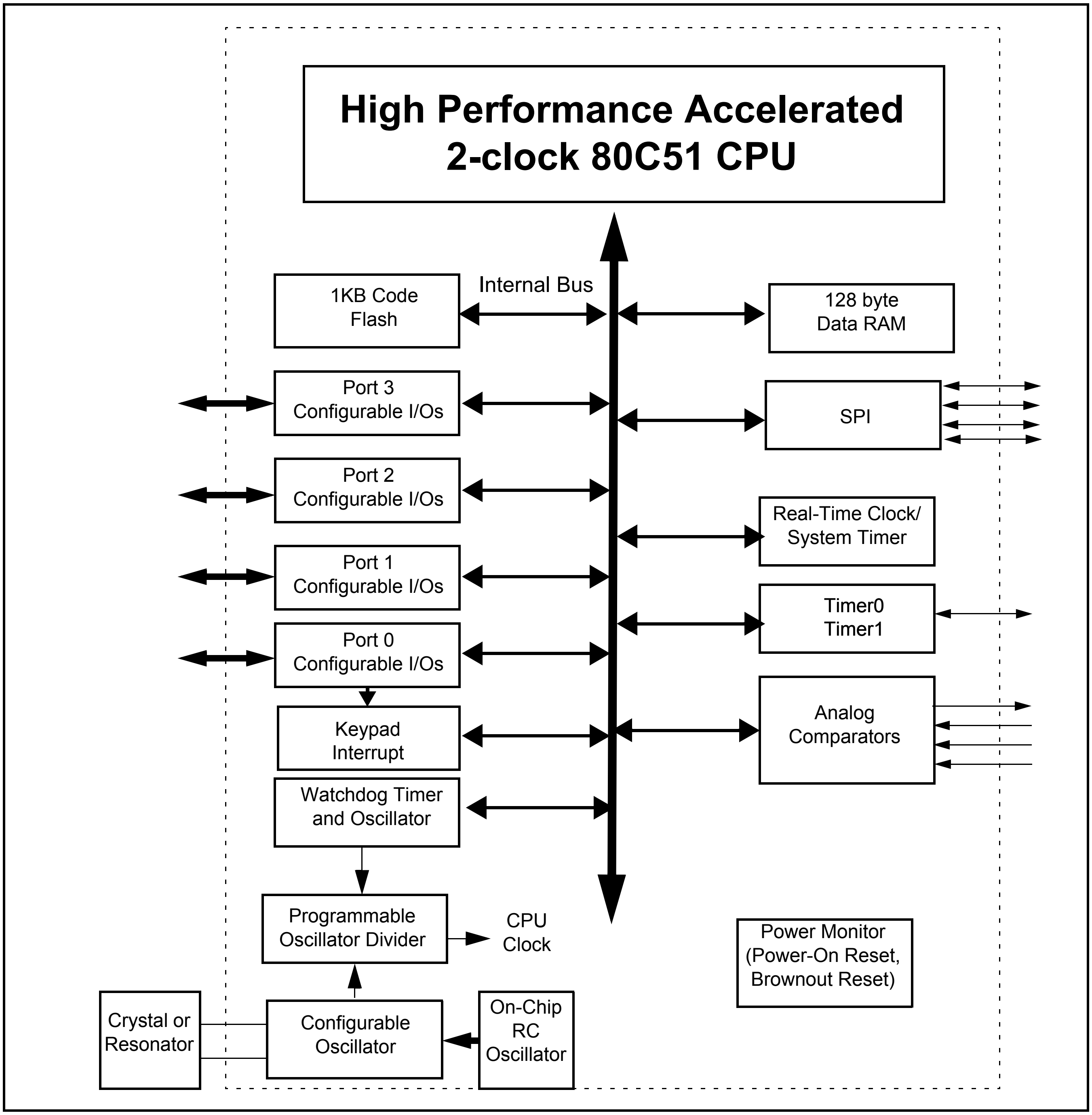


Product comparison

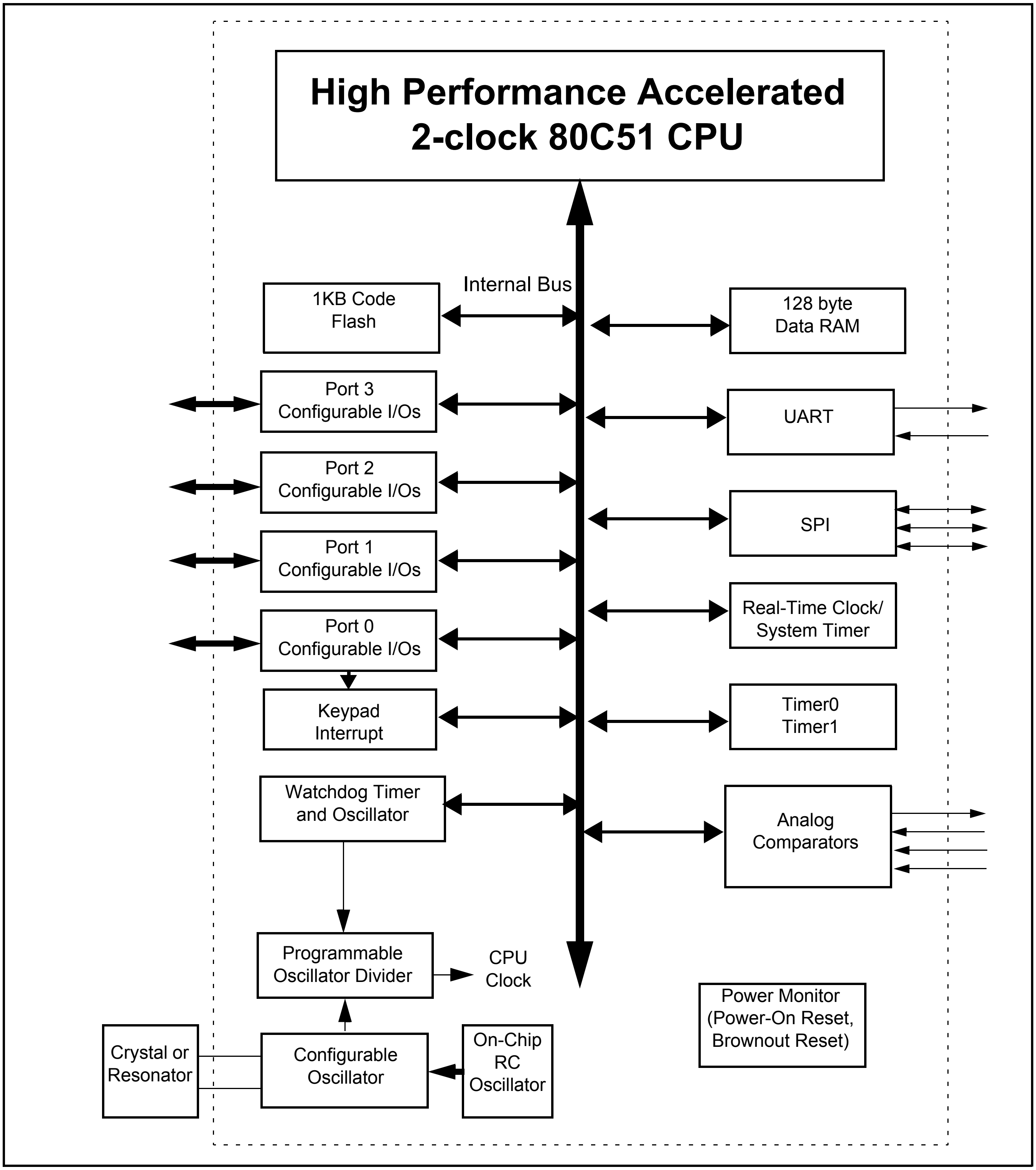
The following table highlights differences between these three devices.

Part number	Ext crystal pins	X2 clkout	T0 PWM output	SPI with SS pin	SPI without SS pin	UART	
						TxD	RxD
P89LPC912	Y	Y	Y	Y			
P89LPC913	Y	Y			Y	Y	Y
P89LPC914			Y	Y		Y	Y

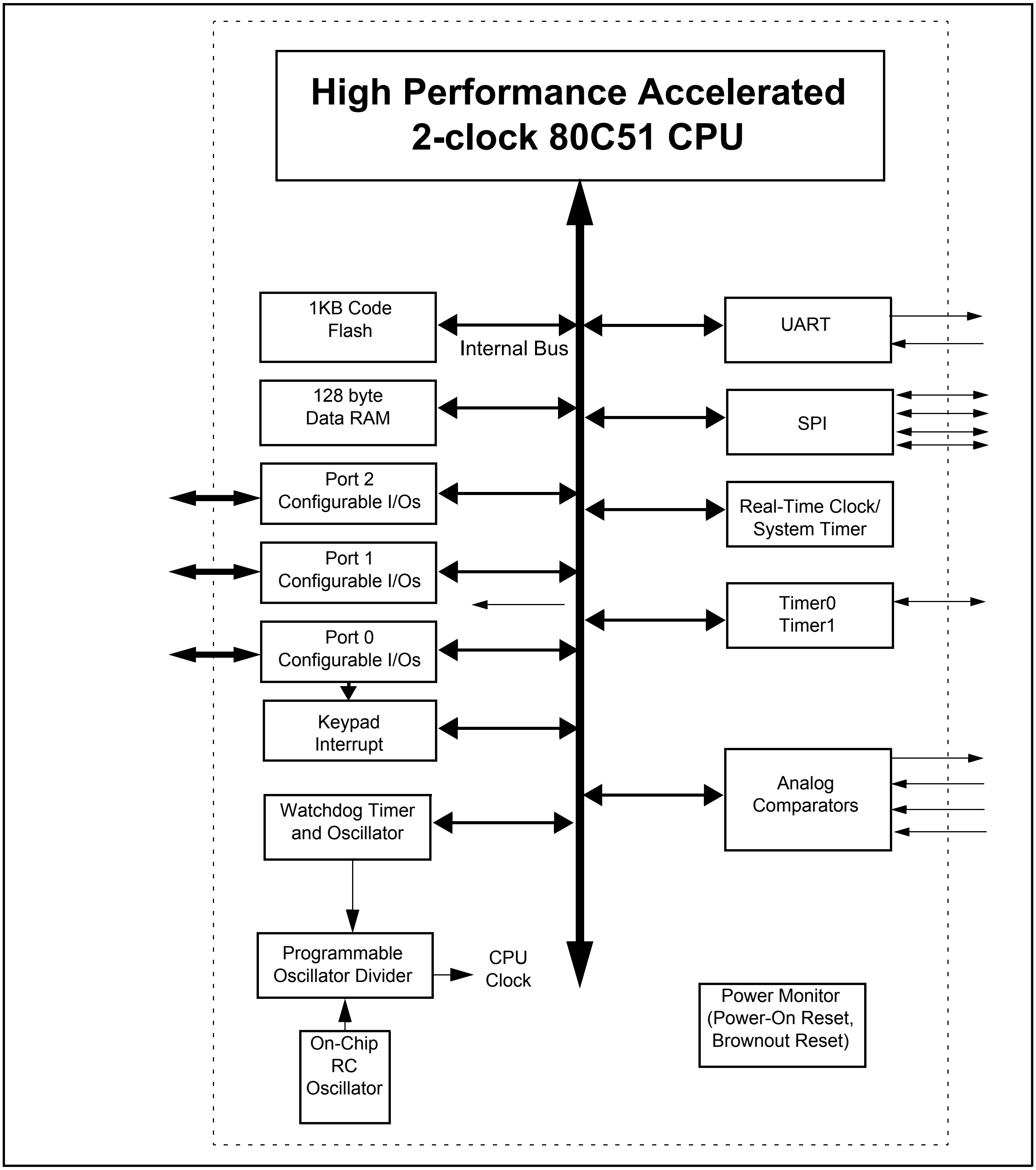
BLOCK DIAGRAM - P89LPC912



BLOCK DIAGRAM - P89LPC913



BLOCK DIAGRAM - P89LPC914



PIN DESCRIPTIONS - P89LPC912

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
P0.2,P0.4 - P0.6	5,11,12,13	I/O	**Port 0:** Port 0 is an 4-bit I/O port with a user-configurable output type. During reset Port 0 latches are configured in the input only mode with the internal pullup disabled. The operation of port 0 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. The Keypad Interrupt feature operates with port 0 pins. All pins have Schmitt triggered inputs. Port 0 also provides various special functions as described below.
	13	I/O I I	**P0.2** Port 0 bit 2. **CIN2A** Comparator 2 positive input A. **KBI2** Keyboard Input 2.
	12	I/O I I	**P0.4** Port 0 bit 4. **CIN1A** Comparator 1 positive input A. **KBI4** Keyboard Input 4.
	11	I/O I I	**P0.5** Port 0 bit 5. **CMPREF** Comparator reference (negative) input. **KBI5** Keyboard Input 5.
	5	I/O O I	**P0.6** Port 0 bit 6. **CMP1** Comparator 1 output. **KBI6** Keyboard Input 6.
P1.2,P1.5	3,6	I/O (P1.2); I (P1.5)	**Port 1:** Port 1 is an 2-bit I/O port with a user-configurable output type, except for three pins as noted below. During reset Port 1 latches are configured in the input only mode with the internal pullup disabled. The operation of the configurable port 1 pins as inputs and outputs depends upon the port configuration selected. Each of the configurable port pins are programmed independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. P1.2 is an open drain when used as an output. P1.5 is input only. All pins have Schmitt triggered inputs. Port 1 also provides various special functions as described below.
	6	I/O I/O	**P1.2** Port 1 bit 2. (Open-drain when used as an output) **T0** Timer/counter 0 external count input or overflow output. (Open-drain when used as output)
	3	I I	**P1.5** Port 1 bit 5. (Input only) **$\overline{\text{RST}}$** External Reset input during power-on or if selected via UCFG1. When functioning as a reset input a low on this pin resets the microcontroller, causing I/O ports and peripherals to take on their default states, and the processor begins execution at address 0. Also used during a power-on sequence to force In-System Programming mode.
P2.2 - P2.5	1,2,9,14	I/O	**Port 2:** Port 2 is a 4-bit I/O port with a user-configurable output type. During reset Port 2 latches are configured in the input only mode with the internal pullup disabled. The operation of port 2 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. All pins have Schmitt triggered inputs. Port 2 also provides various special functions as described below.
	1	I/O I/O	**P2.2** Port 2 bit 2. **MOSI** SPI master out slave in. When configured as master, this pin is output, when configured as slave, this pin is input.

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
	14	I/O	**P2.3** Port 2 bit 3.
		I/O	**MISO** SPI master in slave out. When configured as master, this pin is input, when configured as slave, this pin is output.
	9	I/O	**P2.4** Port 2 bit 4.
		I	$\overline{\mathbf{SS}}$ SPI Slave select.
	2	I/O	**P2.5** Port 2 bit 5.
		I/O	**SPICLK** SPI clock. When configured as master, this pin is output, when configured as slave, this pin is input.
P3.0 - P3.1	7,8	I/O	**Port 3:** Port 3 is an 2-bit I/O port with a user-configurable output type. During reset Port 3 latches are configured in the input only mode with the internal pullup disabled. The operation of port 3 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. All pins have Schmitt triggered inputs. Port 3 also provides various special functions as described below:
	8	I/O	**P3.0** Port 3 bit 0.
		O	**XTAL2** Output from the oscillator amplifier (when a crystal oscillator option is selected via the FLASH configuration).
		O	**CLKOUT** CPU clock divided by 2 when enabled via SFR bit (ENCLK - TRIM.6). It can be used if the CPU clock is the internal RC oscillator, watchdog oscillator or external clock input, except when XTAL1/XTAL2 are used to generate clock source for the Real-Time clock/system timer.
	7	I/O	**P3.1** Port 3 bit 1.
		I	**XTAL1** Input to the oscillator circuit and internal clock generator circuits (when selected via the FLASH configuration). It can be a port pin if internal RC oscillator or watchdog oscillator is used as the CPU clock source, AND if XTAL1/XTAL2 are not used to generate the clock for the Real-Time clock/ system timer.
V_{SS}	4	I	**Ground:** 0 V reference.
V_{DD}	10	I	**Power Supply:** This is the power supply voltage for normal operation as well as Idle and Power down modes.

PIN DESCRIPTIONS - P89LPC913

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
P0.2, P0.4 - P0.6	5, 11, 12, 13	I/O	**Port 0:** Port 0 is an 4-bit I/O port with a user-configurable output type. During reset Port 0 latches are configured in the input only mode with the internal pullup disabled. The operation of port 0 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. The Keypad Interrupt feature operates with port 0 pins. All pins have Schmitt triggered inputs. Port 0 also provides various special functions as described below.
	13	I/O I I	**P0.2** Port 0 bit 2. **CIN2A** Comparator 2 positive input A. **KBI2** Keyboard Input 2.
	12	I/O I I	**P0.4** Port 0 bit 4. **CIN1A** Comparator 1 positive input A. **KBI4** Keyboard Input 4.
	11	I/O I I	**P0.5** Port 0 bit 5. **CMPREF**Comparator reference (negative) input. **KBI5** Keyboard Input 5.
	5	I/O O I	**P0.6** Port 0 bit 6. **CMP1** Comparator 1 output. **KBI6** Keyboard Input 6.
P1.0, P1.1, P1.5	3,6,9	I/O(P1.0, P1.1,); I (P1.5)	**Port 1:** Port 1 is an 3-bit I/O port with a user-configurable output type, except for three pins as noted below. During reset Port 1 latches are configured in the input only mode with the internal pullup disabled. The operation of the configurable port 1 pins as inputs and outputs depends upon the port configuration selected. Each of the configurable port pins are programmed independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. P1.5 is input only. All pins have Schmitt triggered inputs. Port 1 also provides various special functions as described below.
	9	I/O O	**P1.0** Port 1 bit 0. **TxD** Transmitter output for the serial port.
	6	I/O I	**P1.1** Port 1 bit 1. **RxD** Receiver input for the serial port.
	3	I I	**P1.5** Port 1 bit 5. (Input only) $\overline{\textbf{RST}}$ External Reset input during power-on or if selected via UCFG1. When functioning as a reset input a low on this pin resets the microcontroller, causing I/O ports and peripherals to take on their default states, and the processor begins execution at address 0. Also used during a power-on sequence to force In-System Programming mode.
P2.2, P2.3, P2.5	13,14,16	I/O	**Port 2:** Port 2 is a 3-bit I/O port with a user-configurable output type. During reset Port 2 latches are configured in the input only mode with the internal pullup disabled. The operation of port 2 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. All pins have Schmitt triggered inputs. Port 2 also provides various special functions as described below.

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
	13	I/O I/O	**P2.2** Port 2 bit 2. **MOSI** SPI master out slave in. When configured as master, this pin is output, when configured as slave, this pin is input.
	14	I/O I/O	**P2.3** Port 2 bit 3. **MISO** SPI master in slave out. When configured as master, this pin is input, when configured as slave, this pin is output.
	16	I/O I/O	**P2.5** Port 2 bit 5. **SPICLK** SPI clock. When configured as master, this pin is output, when configured as slave, this pin is input.
P3.0 - P3.1	7,8	I/O	**Port 3:** Port 3 is an 2-bit I/O port with a user-configurable output type. During reset Port 3 latches are configured in the input only mode with the internal pullup disabled. The operation of port 3 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. All pins have Schmitt triggered inputs. Port 3 also provides various special functions as described below:
	8	I/O O O	**P3.0** Port 3 bit 0. **XTAL2** Output from the oscillator amplifier (when a crystal oscillator option is selected via the FLASH configuration). **CLKOUT** CPU clock divided by 2 when enabled via SFR bit (ENCLK - TRIM.6). It can be used if the CPU clock is the internal RC oscillator, watchdog oscillator or external clock input, except when XTAL1/XTAL2 are used to generate clock source for the Real-Time clock/system timer.
	7	I/O I	**P3.1** Port 3 bit 1. **XTAL1** Input to the oscillator circuit and internal clock generator circuits (when selected via the FLASH configuration). It can be a port pin if internal RC oscillator or watchdog oscillator is used as the CPU clock source, AND if XTAL1/XTAL2 are not used to generate the clock for the Real-Time clock/ system timer.
V_{SS}	4	I	**Ground:** 0 V reference.
V_{DD}	10	I	**Power Supply:** This is the power supply voltage for normal operation as well as Idle and Power down modes.

PIN DESCRIPTIONS - P89LPC914

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
P0.2, P0.4 - P0.6	5,11,12,13	I/O	**Port 0:** Port 0 is an 4-bit I/O port with a user-configurable output type. During reset Port 0 latches are configured in the input only mode with the internal pullup disabled. The operation of port 0 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. The Keypad Interrupt feature operates with port 0 pins. All pins have Schmitt triggered inputs. Port 0 also provides various special functions as described below.
	13	I/O I I	**P0.2** Port 0 bit 2. **CIN2A** Comparator 2 positive input A. **KBI2** Keyboard Input 2.
	12	I/O I I	**P0.4** Port 0 bit 4. **CIN1A** Comparator 1 positive input A. **KBI4** Keyboard Input 4.
	11	I/O I I	**P0.5** Port 0 bit 5. **CMPREF**Comparator reference (negative) input. **KBI5** Keyboard Input 5.
	5	I/O O I	**P0.6** Port 0 bit 6. **CMP1** Comparator 1 output. **KBI6** Keyboard Input 6.
P1.0 - P1.2,P1.5	3,6,7,9	I/O (P1.0-P1.2); I (P1.5)	**Port 1:** Port 1 is an 4-bit I/O port with a user-configurable output type, except for three pins as noted below. During reset Port 1 latches are configured in the input only mode with the internal pullup disabled. The operation of the configurable port 1 pins as inputs and outputs depends upon the port configuration selected. Each of the configurable port pins are programmed independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. P1.2 is an open drain when used as an output. P1.5 is input only. All pins have Schmitt triggered inputs. Port 1 also provides various special functions as described below.
	9	I/O O	**P1.0** Port 1 bit 0. **TxD** Transmitter output for the serial port.
	6	I/O I	**P1.1** Port 1 bit 1. **RxD** Receiver input for the serial port.
	7	I/O I/O	**P1.2** Port 1 bit 2. (Open-drain when used as an output) **T0** Timer/counter 0 external count input or overflow output. (Open-drain when used as outputs)
	3	I I	**P1.5** Port 1 bit 5. (Input only) **$\overline{\text{RST}}$** External Reset input during power-on or if selected via UCFG1. When functioning as a reset input a low on this pin resets the microcontroller, causing I/O ports and peripherals to take on their default states, and the processor begins execution at address 0. Also used during a power-on sequence to force In-System Programming mode.

MNEMONIC	PIN NO. for TSSOP14	TYPE	NAME AND FUNCTION
P2.2 - P2.5	1,2,8,14	I/O	**Port 2:** Port 2 is a 4-bit I/O port with a user-configurable output type. During reset Port 2 latches are configured in the input only mode with the internal pullup disabled. The operation of port 2 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. All pins have Schmitt triggered inputs. Port 2 also provides various special functions as described below.
	1	I/O I/O	**P2.2** Port 2 bit 2. **MOSI** SPI master out slave in. When configured as master, this pin is output, when configured as slave, this pin is input.
	14	I/O I/O	**P2.3** Port 2 bit 3. **MISO** SPI master in slave out. When configured as master, this pin is input, when configured as slave, this pin is output.
	8	I/O I	**P2.4** Port 2 bit 4. **$\overline{\text{SS}}$** SPI Slave select.
	2	I/O I/O	**P2.5** Port 2 bit 5. **SPICLK** SPI clock. When configured as master, this pin is output, when configured as slave, this pin is input.
V_{SS}	4	I	**Ground:** 0 V reference.
V_{DD}	10	I	**Power Supply:** This is the power supply voltage for normal operation as well as Idle and Power down modes.

SPECIAL FUNCTION REGISTERS

Note: Special Function Registers (SFRs) accesses are restricted in the following ways:

1. User must NOT attempt to access any SFR locations not defined.
2. Accesses to any defined SFR locations must be strictly for the functions for the SFRs.
3. 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.

SPECIAL FUNCTION REGISTERS TABLE - P89LPC912

Name	Description	SFR Address	Bit Functions and Addresses MSB							LSB	Reset Value Hex	Binary
			E7	E6	E5	E4	E3	E2	E1	E0		
ACC*	Accumulator	E0H									00H	00000000
AUXR1#	Auxiliary Function Register	A2H	CLKLP	-	-	ENT0	SRST	0	-	DPS	00H[1]	000000x0
			F7	F6	F5	F4	F3	F2	F1	F0		
B*	B Register	F0H									00H	00000000
CMP1#	Comparator 1 Control Register	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]	xx000000
CMP2#	Comparator 2 Control Register	ADH	-	-	CE2	-	CN2	-	CO2	CMF2	00H[1]	xx000000
DIVM#	CPU Clock Divide-by-M Control	95H									00H	00000000
DPTR	Data Pointer (2 bytes)											
DPH	Data Pointer High	83H									00H	00000000
DPL	Data Pointer Low	82H									00H	00000000
FMADRH#	Program Flash Address High	E7H									00H	00000000
FMADRL#	Program Flash Address Low	E6H									00H	00000000
FMCON#	Program Flash Control (Read)	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H	01110000
	Program Flash Control (Write)		FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0		
FMDATA#	Program Flash Data	E5H									00H	00000000
			AF	AE	AD	AC	AB	AA	A9	A8		
IEN0*	Interrupt Enable 0	A8H	EA	EWDRT	EBO	-	ET1	-	ET0	-	00H	00000000
			EF	EE	ED	EC	EB	EA	E9	E8		
IEN1*#	Interrupt Enable 1	E8H	-	-	-	-	ESPI	EC	EKBI	-	00H[1]	00x00000
			BF	BE	BD	BC	BB	BA	B9	B8		
IP0*	Interrupt Priority 0	B8H	-	PWDRT	PBO	-	PT1	-	PT0	-	00H[1]	x0000000
IP0H#	Interrupt Priority 0 High	B7H	-	PWDRTH	PBOH	-	PT1H	-	PT0H	-	00H[1]	x0000000

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
			FF	FE	FD	FC	FB	FA	F9	F8		
IP1*#	Interrupt Priority 1	F8H	-	-	-	-	PSPI	PC	PKBI	-	00H[1]	00x00000
IP1H#	Interrupt Priority 1 High	F7H	-	-	-	-	PSPIH	PCH	PKBIH	-	00H[1]	00x00000
KBCON#	Keypad Control Register	94H	-	-	-	-	-	-	PATN _SEL	KBIF	00H[1]	xxxxxx00
KBMASK#	Keypad Interrupt Mask Register	86H									00H	00000000
KBPATN#	Keypad Pattern Register	93H									FFH	11111111
			87	86	85	84	83	82	81	80		
P0*	Port 0	80H	-	CMP1/ KB6	CMPREF/ KB5	CIN1A/ KB4	-	CIN2A/KB2	-	-	Note 1	
			97	96	95	94	93	92	91	90		
P1*	Port 1	90H	-	-	$\overline{\text{RST}}$	-	-	T0	-	-	Note 1	
			A7	A6	A5	A4	A3	A2	A1	A0		
P2*	Port 2	A0H	-	-	SPICLK	$\overline{\text{SS}}$	MISO	MOSI	-	-	Note 1	
			B7	B6	B5	B4	B3	B2	B1	B0		
P3*	Port 3	B0H	-	-	-	-	-	-	XTAL1	XTAL2	Note 1	
P0M1#	Port 0 Output Mode 1	84H	-	(P0M1.6)	(P0M1.5)	(P0M1.4)	-	(P0M1.2)	-	-	FFH	11111111
P0M2#	Port 0 Output Mode 2	85H	-	(P0M2.6)	(P0M2.5)	(P0M2.4)	-	(P0M2.2)	-	-	00H	00000000
P1M1#	Port 1 Output Mode 1	91H	-	-	-	-		(P1M1.2)	-	-	D3H[1]	11x1xx11
P1M2#	Port 1 Output Mode 2	92H	-	-	-	-		(P1M2.2)	-	-	00H[1]	00x0xx00
P2M1#	Port 2 Output Mode 1	A4H	-	-	(P2M1.5)	(P2M1.4)	(P2M1.3)	(P2M1.2)	-	-	FFH	11111111
P2M2#	Port 2 Output Mode 2	A5H	-	-	(P2M2.5)	(P2M2.4)	(P2M2.3)	(P2M2.2)	-	-	00H	00000000
P3M1#	Port 3 Output Mode 1	B1H	-	-	-	-	-	-	(P3M1.1)	(P3M1.0)	03H[1]	xxxxxx11
P3M2#	Port 3 Output Mode 2	B2H	-	-	-	-	-	-	(P3M2.1)	(P3M2.0)	00H[1]	xxxxxx00
PCON#	Power Control Register	87H	-	-	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H	00000000
PCONA#	Power Control Register A	B5H	RTCPD	-	VCPD	-	-	SPPD	-	-	00H[1]	00000000
			D7	D6	D5	D4	D3	D2	D1	D0		
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H	00000000
PT0AD#	Port 0 Digital Input Disable	F6H	-	-	PT0AD.5	PT0AD.4	-	PT0AD.2	-	-	00H	xx00000x
RSTSRC#	Reset Source Register	DFH	-	-	BOF	POF	-	R_WD	R_SF	R_EX	Note 2	
RTCCON#	Real-Time Clock Control	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]	011xxx00
RTCH#	Real-Time Clock Register High	D2H									00H[5]	00000000
RTCL#	Real-Time Clock Register Low	D3H									00H[5]	00000000

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
SP	Stack Pointer	81H									07H	00000111
SPCTL#	SPI Control Register	E2H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	04H	00000100
SPSTAT#	SPI Status Register	E1H	SPIF	WCOL	-	-	-	-	-	-	00H	00xxxxxx
SPDAT#	SPI Data Register	E3H									00H	00000000
TAMOD#	Timer 0 and 1 Auxiliary Mode	8FH	-	-	-	-	-	-	-	T0M2	00H	xxx0xxx0
			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer 0 and 1 Control	88H	TF1	TR1	TF0	TR0	-	-	-	-	00H	00000000
TH0	Timer 0 High	8CH									00H	00000000
TH1	Timer 1 High	8DH									00H	00000000
TL0	Timer 0 Low	8AH									00H	00000000
TL1	Timer 1 Low	8BH									00H	00000000
TMOD	Timer 0 and 1 Mode	89H	-	-	T1M1	T1M0	T0GATE	T0C/$\overline{T}$	T0M1	T0M0	00H	00000000
TRIM#	Internal Oscillator Trim Register	96H	-	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	Notes 4,5	
WDCON#	Watchdog Control Register	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	Notes 3,5	
WDL#	Watchdog Load	C1H									FFH	11111111
WFEED1#	Watchdog Feed 1	C2H										
WFEED2#	Watchdog Feed 2	C3H										

SPECIAL FUNCTION REGISTERS TABLE - P89LPC913

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
			E7	E6	E5	E4	E3	E2	E1	E0		
ACC*	Accumulator	E0H									00H	00000000
AUXR1#	Auxiliary Function Register	A2H	CLKLP	EBRR	-	-	SRST	0	-	DPS	00H[1]	000000x0
			F7	F6	F5	F4	F3	F2	F1	F0		
B*	B Register	F0H									00H	00000000
BRGR0#§	Baud Rate Generator Rate Low	BEH									00H	00000000
BRGR1#§	Baud Rate Generator Rate High	BFH									00H	00000000
BRGCON#	Baud Rate Generator Control	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00H[%]	xxxxxx00
CMP1#	Comparator 1 Control Register	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]	xx000000
CMP2#	Comparator 2 Control Register	ADH	-	-	CE2	-	CN2	-	-	CMF2	00H[1]	xx000000
DIVM#	CPU Clock Divide-by-M Control	95H									00H	00000000
DPTR	Data Pointer (2 bytes)											
DPH	Data Pointer High	83H									00H	00000000
DPL	Data Pointer Low	82H									00H	00000000
FMADRH#	Program Flash Address High	E7H									00H	00000000
FMADRL#	Program Flash Address Low	E6H									00H	00000000
FMCON#	Program Flash Control (Read)	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H	01110000
	Program Flash Control (Write)		FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0		
FMDATA#	Program Flash Data	E5H									00H	00000000
			AF	AE	AD	AC	AB	AA	A9	A8		
IEN0*	Interrupt Enable 0	A8H	EA	EWDRT	EBO	ES/ESR	ET1	-	ET0	-	00H	00000000
			EF	EE	ED	EC	EB	EA	E9	E8		
IEN1*#	Interrupt Enable 1	E8H	-	EST	-	-	ESPI	EC	EKBI	-	00H[1]	00x00000
			BF	BE	BD	BC	BB	BA	B9	B8		
IP0*	Interrupt Priority 0	B8H	-	PWDRT	PBO	PS/PSR	PT1	-	PT0	-	00H[1]	x0000000
IP0H#	Interrupt Priority 0 High	B7H	-	PWDRTH	PBOH	PSH/PSRH	PT1H	-	PT0H	-	00H[1]	x0000000
			FF	FE	FD	FC	FB	FA	F9	F8		
IP1*#	Interrupt Priority 1	F8H	-	PST	-	-	PSPI	PC	PKBI	-	00H[1]	00x00000
IP1H#	Interrupt Priority 1 High	F7H	-	PSTH	-	-	PSPIH	PCH	PKBIH	-	00H[1]	00x00000

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
KBCON#	Keypad Control Register	94H	-	-	-	-	-	-	PATN _SEL	KBIF	00H[1]	xxxxxx00
KBMASK#	Keypad Interrupt Mask Register	86H									00H	00000000
KBPATN#	Keypad Pattern Register	93H									FFH	11111111
			87	86	85	84	83	82	81	80		
P0*	Port 0	80H	-	CMP1/ KB6	CMPREF/ KB5	CIN1A/ KB4	-	CIN2A/KB2	-	-	Note 1	
			97	96	95	94	93	92	91	90		
P1*	Port 1	90H	-	-	$\overline{\text{RST}}$	-	-	-	RxD	TxD	Note 1	
			A7	A6	A5	A4	A3	A2	A1	A0		
P2*	Port 2	A0H	-	-	SPICLK	-	MISO	MOSI	-	-	Note 1	
			B7	B6	B5	B4	B3	B2	B1	B0		
P3*	Port 3	B0H	-	-	-	-	-	-	XTAL1	XTAL2	Note 1	
P0M1#	Port 0 Output Mode 1	84H	-	(P0M1.6)	(P0M1.5)	(P0M1.4)	-	(P0M1.2)	-	-	FFH	11111111
P0M2#	Port 0 Output Mode 2	85H	-	(P0M2.6)	(P0M2.5)	(P0M2.4)	-	(P0M2.2)	-	-	00H	00000000
P1M1#	Port 1 Output Mode 1	91H	-	-	-	-	-	-	(P1M1.1)	(P1M1.0)	D3H[1]	11x1xx11
P1M2#	Port 1 Output Mode 2	92H	-	-	-	-	-	-	(P1M2.1)	(P1M2.0)	00H[1]	00x0xx00
P2M1#	Port 2 Output Mode 1	A4H	-	-	(P2M1.5)	-	(P2M1.3)	(P2M1.2)	-	-	FFH	11111111
P2M2#	Port 2 Output Mode 2	A5H	-	-	(P2M2.5)	-	(P2M2.3)	(P2M2.2)	-	-	00H	00000000
P3M1#	Port 3 Output Mode 1	B1H	-	-	-	-	-	-	(P3M1.1)	(P3M1.0)	03H[1]	xxxxxx11
P3M2#	Port 3 Output Mode 2	B2H	-	-	-	-	-	-	(P3M2.1)	(P3M2.0)	00H[1]	xxxxxx00
PCON#	Power Control Register	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H	00000000
PCONA#	Power Control Register A	B5H	RTCPD	-	VCPD	-	-	SPPD	SPD	-	00H[1]	00000000
			D7	D6	D5	D4	D3	D2	D1	D0		
PSW*	Program Status Wword	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H	00000000
PT0AD#	Port 0 Digital Input Disable	F6H	-	-	PT0AD.5	PT0AD.4	-	PT0AD.2	-	-	00H	xx00000x
RSTSRC#	Reset Source Register	DFH	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX	Note 2	
RTCCON#	Real-Time Clock Control	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]	011xxx00
RTCH#	Real-Time Clock Register High	D2H									00H[5]	00000000
RTCL#	Real-Time Clock Register Low	D3H									00H[5]	00000000
SADDR#	Serial Port Address Register	A9H									00H	00000000
SADEN#	Serial Port Address Enable	B9H									00H	00000000
SBUF	Serial Port Data Buffer Register	99H									xxH	xxxxxxxx
			9F	9E	9D	9C	9B	9A	99	98		
SCON*	Serial Port Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H	00000000

Name	**Description**	**SFR Address**	**Bit Functions and Addresses**								**Reset Value**	
			MSB							**LSB**	**Hex**	**Binary**
SSTAT#	Serial Port Extended Status Register	BAH	DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT	00H	00000000
SP	Stack Pointer	81H									07H	00000111
SPCTL#	SPI Control Register	E2H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	04H	00000100
SPSTAT#	SPI Status Register	E1H	SPIF	WCOL	-	-	-	-	-	-	00H	00xxxxxx
SPDAT#	SPI Data Register	E3H									00H	00000000
			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer 0 and 1 Control	88H	TF1	TR1	TF0	TR0	-	-	-	-	00H	00000000
TH0	Timer 0 High	8CH									00H	00000000
TH1	Timer 1 High	8DH									00H	00000000
TL0	Timer 0 Low	8AH									00H	00000000
TL1	Timer 1 Low	8BH									00H	00000000
TMOD	Timer 0 and 1 Mode	89H	-	-	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00H	00000000
TRIM#	Internal Oscillator Trim Register	96H	-	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	Notes 4,5	
WDCON#	Watchdog Control Register	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	Notes 3,5	
WDL#	Watchdog Load	C1H									FFH	11111111
WFEED1#	Watchdog Feed 1	C2H										
WFEED2#	Watchdog Feed 2	C3H										

SPECIAL FUNCTION REGISTERS TABLE - P89LPC914

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
			E7	E6	E5	E4	E3	E2	E1	E0		
ACC*	Accumulator	E0H									00H	00000000
AUXR1#	Auxiliary Function Register	A2H	-	EBRR	-	ENT0	SRST	0	-	DPS	00H[1]	000000x0
			F7	F6	F5	F4	F3	F2	F1	F0		
B*	B Register	F0H									00H	00000000
BRGR0#§	Baud Rate Generator Rate Low	BEH									00H	00000000
BRGR1#§	Baud Rate Generator Rate High	BFH									00H	00000000
BRGCON#	Baud Rate Generator Control	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00H[%]	xxxxxx00
CMP1#	Comparator 1 Control Register	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]	xx000000
CMP2#	Comparator 2 Control Register	ADH	-	-	CE2	-	CN2	-	CO2	CMF2	00H[1]	xx000000
DIVM#	CPU Clock Divide-by-M Control	95H									00H	00000000
DPTR	Data Pointer (2 bytes)											
DPH	Data Pointer High	83H									00H	00000000
DPL	Data Pointer Low	82H									00H	00000000
FMADRH#	Program Flash Address High	E7H									00H	00000000
FMADRL#	Program Flash Address Low	E6H									00H	00000000
FMCON#	Program Flash Control (Read)	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H	01110000
	Program Flash Control (Write)		FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0		
FMDATA#	Program Flash Data	E5H									00H	00000000
			AF	AE	AD	AC	AB	AA	A9	A8		
IEN0*	Interrupt Enable 0	A8H	EA	EWDRT	EBO	ES/ESR	ET1	-	ET0	-	00H	00000000
			EF	EE	ED	EC	EB	EA	E9	E8		
IEN1*#	Interrupt Enable 1	E8H	-	EST	-	-	ESPI	EC	EKBI	-	00H[1]	00x00000
			BF	BE	BD	BC	BB	BA	B9	B8		
IP0*	Interrupt Priority 0	B8H	-	PWDRT	PBO	PS/PSR	PT1	-	PT0	-	00H[1]	x0000000
IP0H#	Interrupt Priority 0 High	B7H	-	PWDRTH	PBOH	PSH/ PSRH	PT1H	-	PT0H	-	00H[1]	x0000000
			FF	FE	FD	FC	FB	FA	F9	F8		
IP1*#	Interrupt Priority 1	F8H	-	PST	-	-	PSPI	PC	PKBI	-	00H[1]	00x00000
IP1H#	Interrupt Priority 1 High	F7H	-	PSTH	-	-	PSPIH	PCH	PKBIH	-	00H[1]	00x00000

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
KBCON#	Keypad Control Register	94H	-	-	-	-	-	-	PATN_SEL	KBIF	00H[1]	xxxxxx00
KBMASK#	Keypad Interrupt Mask Register	86H									00H	00000000
KBPATN#	Keypad Pattern Register	93H									FFH	11111111
			87	86	85	84	83	82	81	80		
P0*	Port 0	80H	-	CMP1/ KB6	CMPREF/ KB5	CIN1A/ KB4	-	CIN2A/KB2	-	-	Note 1	
			97	96	95	94	93	92	91	90		
P1*	Port 1	90H			$\overline{\text{RST}}$			T0	RxD	TxD	Note 1	
			A7	A6	A5	A4	A3	A2	A1	A0		
P2*	Port 2	A0H	-	-	SPICLK	$\overline{\text{SS}}$	MISO	MOSI	-	-	Note 1	
P0M1#	Port 0 Output Mode 1	84H	-	(P0M1.6)	(P0M1.5)	(P0M1.4)	-	(P0M1.2)	-	-	FFH	11111111
P0M2#	Port 0 Output Mode 2	85H	-	(P0M2.6)	(P0M2.5)	(P0M2.4)	-	(P0M2.2)	-	-	00H	00000000
P1M1#	Port 1 Output Mode 1	91H	-	-	-	-	-	(P1M1.2)	(P1M1.1)	(P1M1.0)	D3H[1]	11x1xx11
P1M2#	Port 1 Output Mode 2	92H	-	-	-	-	-	(P1M2.2)	(P1M2.1)	(P1M2.0)	00H[1]	00x0xx00
P2M1#	Port 2 Output Mode 1	A4H	-	-	(P2M1.5)	(P2M1.4)	(P2M1.3)	(P2M1.2)	-	-	FFH	11111111
P2M2#	Port 2 Output Mode 2	A5H	-	-	(P2M2.5)	(P2M2.4)	(P2M2.3)	(P2M2.2)	-	-	00H	00000000
PCON#	Power Control Register	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H	00000000
PCONA#	Power Control Register A	B5H	RTCPD	-	VCPD	-	-	SPPD	SPD	-	00H[1]	00000000
			D7	D6	D5	D4	D3	D2	D1	D0		
PSW*	Program Status Wword	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H	00000000
PT0AD#	Port 0 Digital Input Disable	F6H	-	-	PT0AD.5	PT0AD.4	-	PT0AD.2	-	-	00H	xx00000x
RSTSRC#	Reset Source Register	DFH	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX	Note 2	
RTCCON#	Real-Time Clock Control	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]	011xxx00
RTCH#	Real-Time Clock Register High	D2H									00H[5]	00000000
RTCL#	Real-Time Clock Register Low	D3H									00H[5]	00000000
SADDR#	Serial Port Address Register	A9H									00H	00000000
SADEN#	Serial Port Address Enable	B9H									00H	00000000
SBUF	Serial Port Data Buffer Register	99H									xxH	xxxxxxxx
			9F	9E	9D	9C	9B	9A	99	98		
SCON*	Serial Port Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H	00000000
SSTAT#	Serial Port Extended Status Register	BAH	DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT	00H	00000000

Name	Description	SFR Address	Bit Functions and Addresses								Reset Value	
			MSB							LSB	Hex	Binary
SP	Stack Pointer	81H									07H	00000111
SPCTL#	SPI Control Register	E2H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	04H	00000100
SPSTAT#	SPI Status Register	E1H	SPIF	WCOL	-	-	-	-	-	-	00H	00xxxxxx
SPDAT#	SPI Data Register	E3H									00H	00000000
TAMOD#	Timer 0 and 1 Auxiliary Mode	8FH	-	-	-	-	-	-	-	T0M2	00H	xxx0xxx0
			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer 0 and 1 Control	88H	TF1	TR1	TF0	TR0	-	-	-	-	00H	00000000
TH0	Timer 0 High	8CH									00H	00000000
TH1	Timer 1 High	8DH									00H	00000000
TL0	Timer 0 Low	8AH									00H	00000000
TL1	Timer 1 Low	8BH									00H	00000000
TMOD	Timer 0 and 1 Mode	89H	-	-	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00H	00000000
TRIM#	Internal Oscillator Trim Register	96H	-	-	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	Notes 4,5	
WDCON#	Watchdog Control Register	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	Notes 3,5	
WDL#	Watchdog Load	C1H									FFH	11111111
WFEED1#	Watchdog Feed 1	C2H										
WFEED2#	Watchdog Feed 2	C3H										

Notes:

* SFRs are bit addressable.

\# SFRs are modified from or added to the 80C51 SFRs.

\- Reserved bits, must be written with 0's.

§ BRGR1 and BRGR0 must only be written if BRGEN in BRGCON SFR is '0'. If any of them is written if BRGEN = 1, result is unpredictable.

Unimplemented bits in SFRs (labeled '-') are X (unknown) at all times. Unless otherwise specified, ones 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.

1. All ports are in input only (high impendance) state after power-up.
2. The RSTSRC register reflects the cause of the P89LPC912/913/914 reset. Upon a power-up reset, all reset source flags are cleared except POF and BOF - the power-on reset value is xx110000.
3. After reset, the value is 111001x1, i.e., PRE2-PRE0 are all 1, WDRUN=1 and WDCLK=1. WDTOF bit is 1 after watchdog reset and is 0 after power-on reset. Other resets will not affect WDTOF.
4. On power-on reset, the TRIM SFR is initialized with a factory preprogrammed value. Other resets will not cause initialization of the TRIM register.
5. The only reset source that affects these SFRs is power-on reset.

MEMORY ORGANIZATION

The P89LPC912/913/914 memory map is shown in Figure 1-1.

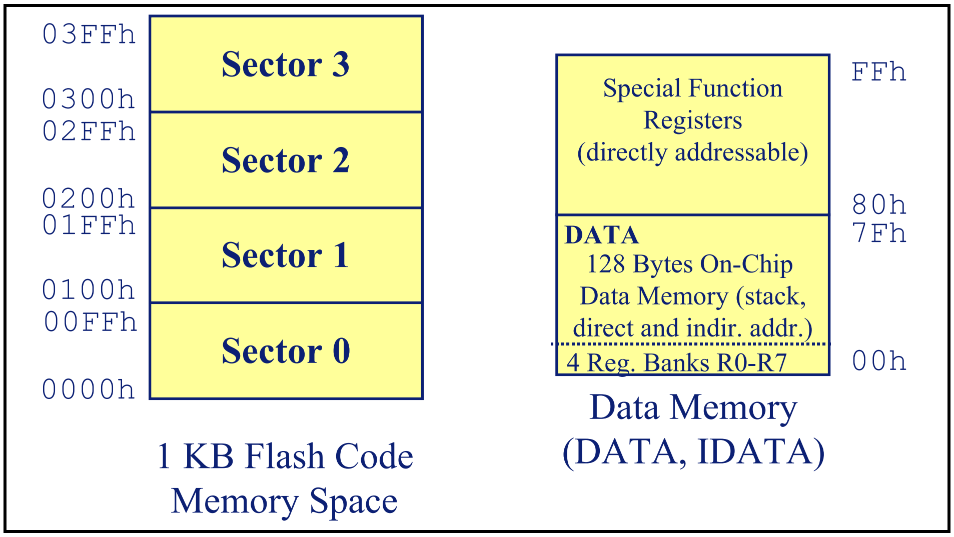


Figure 1-1: P89LPC912/913/914 Memory Map

The various P89LPC912/913/914 memory spaces are as follows:

DATA	128 bytes of internal data memory space (00h..7Fh) accessed via direct or indirect addressing, using instructions other than MOVX and MOVC.
SFR	Special Function Registers. Selected CPU registers and peripheral control and status registers, accessible only via direct addressing.
CODE	1KB of Code memory accessed as part of program execution and via the MOVC instruction.

2. CLOCKS

ENHANCED CPU

The **P89LPC912/913/914** uses an enhanced 80C51 CPU which runs at 6 times the speed of standard 80C51 devices. A machine cycle consists of two CPU clock cycles, and most instructions execute in one or two machine cycles.

CLOCK DEFINITIONS

The **P89LPC912/913/914** device has several internal clocks as defined below:

- OSCCLK - Input to the DIVM clock divider. OSCCLK is selected from one of the clock sources (see Figure 2-3,Figure 2-5,) and can also be optionally divided to a slower frequency (see section "CPU Clock (CCLK) Modification: DIVM Register"). **Note**: f_{OSC} is defined as the OSCCLK frequency.
- XCLK - Output of the crystal oscillator (**P89LPC912, P89LPC913**)
- CCLK - CPU clock .
- PCLK - Clock for the various peripheral devices and is CCLK/2

CPU CLOCK (OSCCLK)

The **P89LPC912/913** provides several user-selectable oscillator options. This allows optimization for a range of needs from high precision to lowest possible cost. These options are configured when the FLASH is programmed and include an on-chip watchdog oscillator, an on-chip RC oscillator, an oscillator using an external crystal, or an external clock source. The crystal oscillator can be optimized for low, medium, or high frequency crystals covering a range from 20 kHz to 12 MHz.

The **P89LPC914** device allows the user to select between an on-chip watchdog oscillator and an on-chip RC oscillator as the CPU clock source.

LOW SPEED OSCILLATOR OPTION - P89LPC912/913

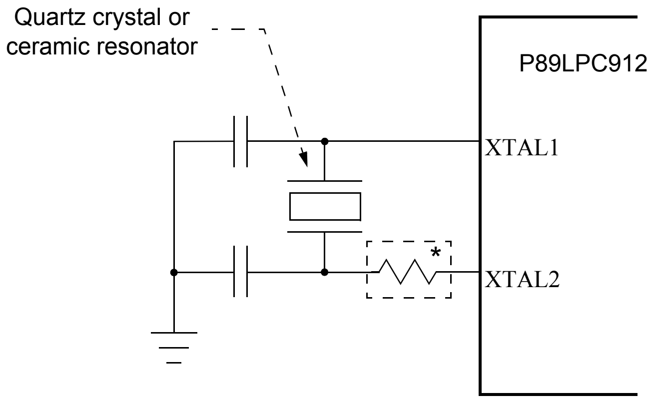
This option supports an external crystal in the range of 20 kHz to 100 kHz. Ceramic resonators are also supported in this configuration.

MEDIUM SPEED OSCILLATOR OPTION - P89LPC912/913

This option supports an external crystal in the range of 100 kHz to 4 MHz. Ceramic resonators are also supported in this configuration.

HIGH SPEED OSCILLATOR OPTION - P89LPC912/913

This option supports an external crystal in the range of 4MHz to 12 MHz. Ceramic resonators are also supported in this configuration. If CCLK is 8MHz or slower, the CLKLP SFR bit (AUXR1.7) can be set to '1' to reduce power consumption. On reset, CLKLP is '0' allowing highest performance access. This bit can then be set in software if CCLK is running at 8MHz or slower.



The oscillator must be configured in one of the following modes:

- Low Frequency Crystal
- Medium Frequency Crystal
- High Frequency Crystal

* A series resistor may be required to limit crystal drive levels. This is especially important for low frequency crystals.

Figure 2-1: Using the Crystal Oscillator - P89LPC912/913

OSCILLATOR OPTION SELECTION- P89LPC912/913

The oscillator option is selectable either by the FOSC2:0 bits in UCFG1 or by the RTCS1:0 bits in RTCCON. If the FOSC2:0 bits select an OSCCLK source of either the internal RC oscillator or the WDT oscillator, then the RTCS1:0 bits will select the oscillator option for the crystal oscillator. Otherwise, the crystal oscillator option is selected by FOSC2:0. See Table 6-1and Table 6-2.

CLOCK OUTPUT - P89LPC912/913

The **P89LPC912/913** supports a user selectable clock output function on the XTAL2 / CLKOUT pin when no crystal oscillator is being used. This condition occurs if another clock source has been selected (on-chip RC oscillator, watchdog oscillator, external clock input on X1) and if the Real-Time clock is not using the crystal oscillator as its clock source. This allows external devices to synchronize to the **P89LPC912/913**. This output is enabled by the ENCLK bit in the TRIM register. The frequency of this clock output is 1/2 that of the CCLK. If the clock output is not needed in Idle mode, it may be turned off prior to entering Idle, saving additional power. Note: on reset, the TRIM SFR is initialized with a factory preprogrammed value. Therefore when setting or clearing the ENCLK bit, the user should retain the contents of bits 5:0 of the TRIM register. This can be done by reading the contents of the TRIM register (into the ACC for example), modifying bit 6, and writing this result back into the TRIM register. Alternatively, the "ANL direct" or "ORL direct" instructions can be used to clear or set bit 6 of the TRIM register. Increasing the TRIM value will decrease the oscillator frequency.

ON-CHIP RC OSCILLATOR OPTION

The **P89LPC912/913/914** has a 6-bit field within the TRIM register that can be used to tune the frequency of the RC oscillator. During reset, the TRIM value is initialized to a factory pre-programmed value to adjust the oscillator frequency to 7.373 MHz, ±1%. (Note: the initial value is better than 1%; please refer to the datasheet for behavior over temperature). End user applications can write to the TRIM register to adjust the on-chip RC oscillator to other frequencies. Increasing the TRIM value will decrease the oscillator frequency.

If CCLK is 8MHz or slower, the CLKLP SFR bit (AUXR1.7) can be set to '1' to reduce power consumption. On reset, CLKLP is '0' allowing highest performance access. This bit can then be set in software if CCLK is running at 8MHz or slower.

TRIM

Address: 96h

Not bit addressable

7	6	5	4	3	2	1	0
-	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0

Reset Source(s): Power-up only

Reset Value: On power-up reset, ENCLK = 0, and TRIM.5-0 are loaded with the factory programmed value.

BIT	SYMBOL	FUNCTION
TRIM.7	-	Reserved.
TRIM.6	ENCLK	When ENCLK =1, CCLK/ 2 is output on the XTAL2 pin (P3.0) provided that the crystal oscillator is not being used. When ENCLK=0, no clock output is enabled.
TRIM.5-0		Trim value.

Note: on reset, the TRIM SFR is initialized with a factory preprogrammed value. When setting or clearing the ENCLK bit, the user should retain the contents of bits 5:0 of the TRIM register. This can be done by reading the contents of the TRIM register (into the ACC for example), modifying bit 6, and writing this result back into the TRIM register. Alternatively, the "ANL direct" or "ORL direct" instructions can be used to clear or set bit 6 of the TRIM register.

Figure 2-2: On-Chip RC Oscillator TRIM Register

WATCHDOG OSCILLATOR OPTION

The watchdog has a separate oscillator which has a nominal frequency of 400KHz. This oscillator can be used to save power when a high clock frequency is not needed.

EXTERNAL CLOCK INPUT OPTION - P89LPC912/913

In this configuration, the processor clock is derived from an external source driving the XTAL1 / P3.1 pin. The rate may be from 0 Hz up to 12 MHz. The XTAL2 / P3.0 pin may be used as a standard port pin or a clock output.

.

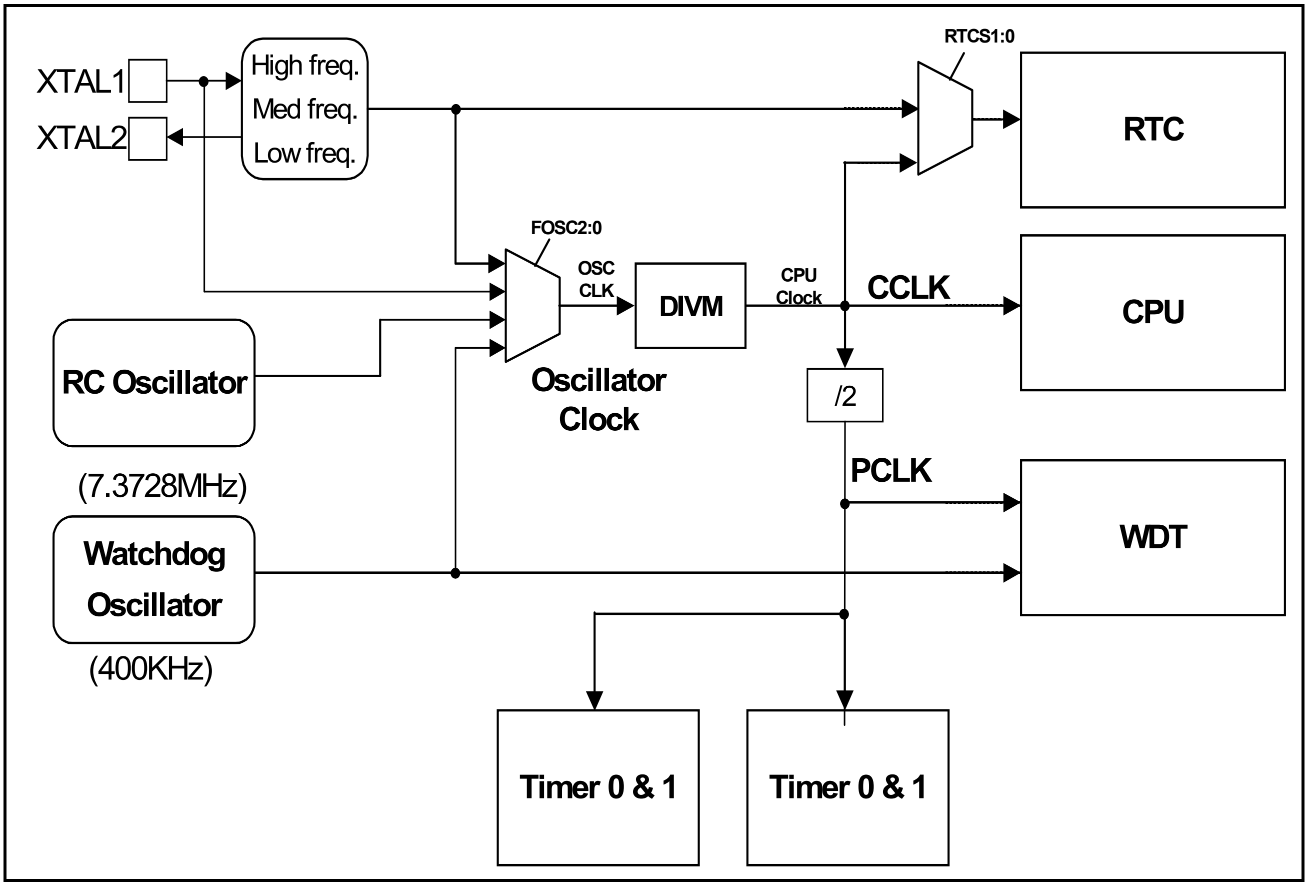


Figure 2-3: Block Diagram of Oscillator Control - P89LPC912

.

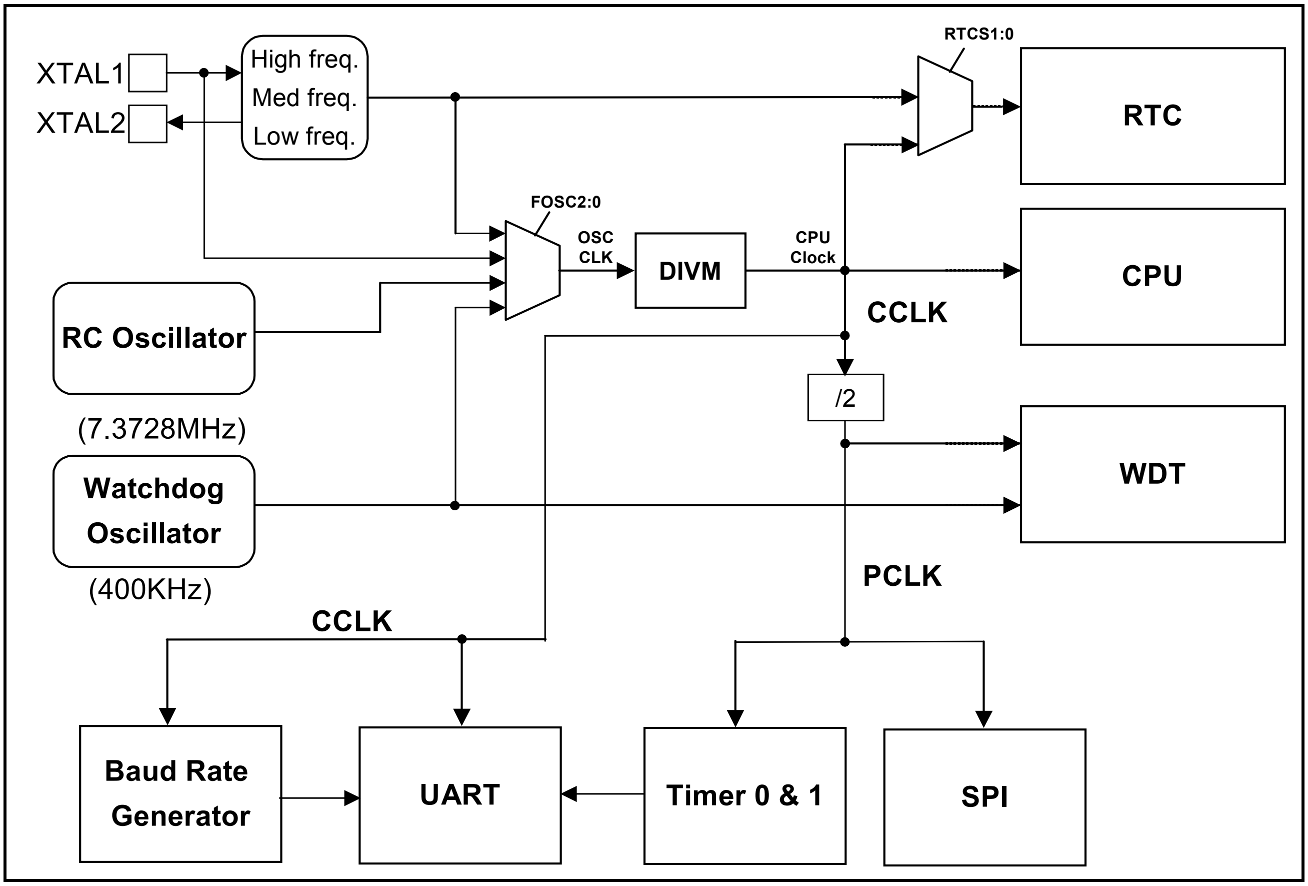


Figure 2-4: Block Diagram of Oscillator Control - P89LPC913

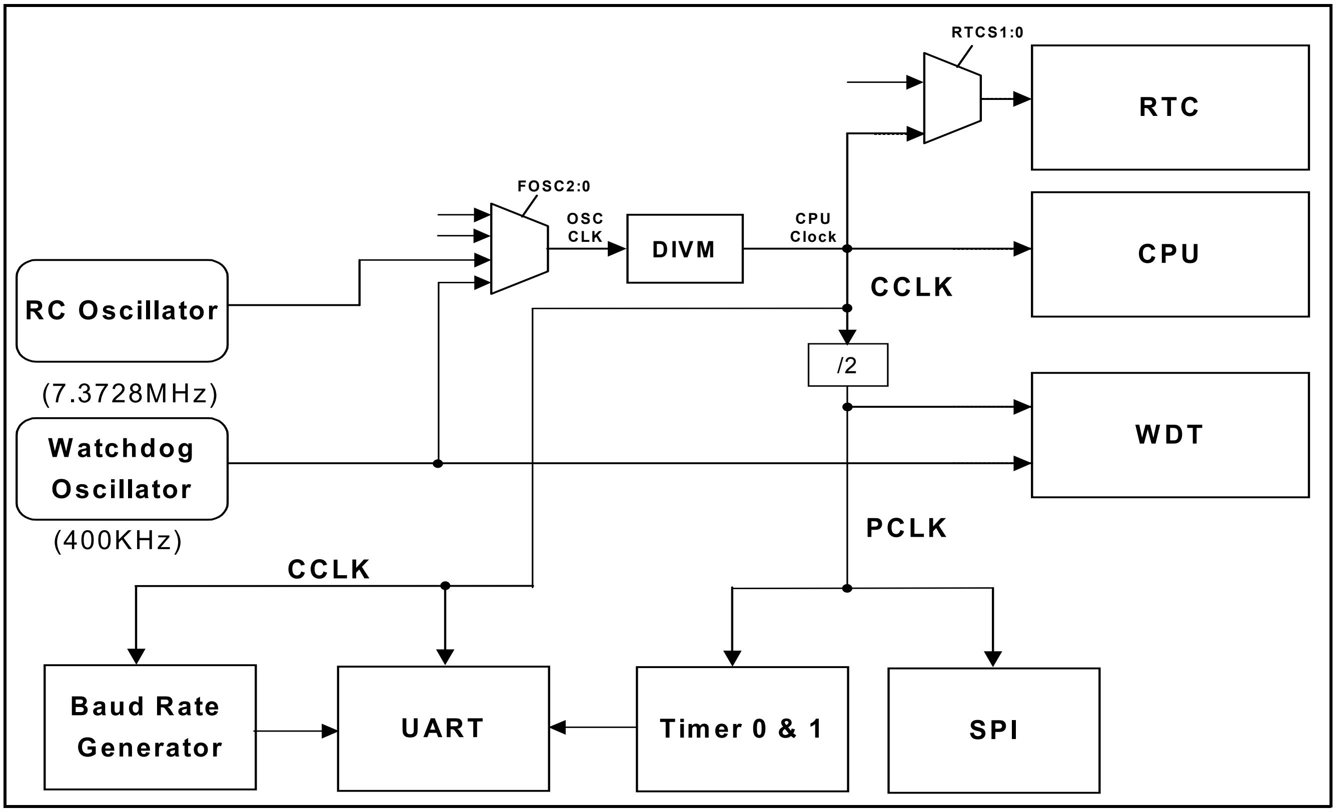


Figure 2-5: Block Diagram of Oscillator Control- P89LPC914

CPU CLOCK (CCLK) WAKEUP DELAY

The **P89LPC912/913/914** has an internal wakeup timer that delays the clock until it stabilizes depending to the clock source used. If the clock source is any of the three crystal selections (**P89LPC912/913**), the delay is 992 OSCCLK cycles plus 60-100µs. If the clock source is either the internal RC oscillator or the Watchdog oscillator, the delay is 224 OSCCLK cycles plus 60-100µs.

CPU CLOCK (CCLK) MODIFICATION: DIVM REGISTER

The OSCCLK frequency can be divided down, by an integer, up to 510 times by configuring a dividing register, DIVM, to provide CCLK. This produces the CCLK frequency using the following formula:

CCLK frequency = f_{OSC} / (2N)

Where: f_{OSC} is the frequency of OSCCLK

N is the value of DIVM.

Since N ranges from 0 to 255, the CCLK frequency can be in the range of f_{OSC} to f_{OSC}/510. (for N =0, CCLK = f_{OSC}) .

This feature makes it possible to temporarily run the CPU at a lower rate, reducing power consumption. By dividing the clock, the CPU can retain the ability to respond to events other than those that can cause interrupts (i.e. events that allow exiting the Idle mode) by executing its normal program at a lower rate. This can often result in lower power consumption than in Idle mode. This can allow bypassing the oscillator start-up time in cases where Power down mode would otherwise be used. The value of DIVM may be changed by the program at any time without interrupting code execution.

LOW POWER SELECT (P89LPC912/913)

The P89LPC912/913 is designed to run at 12MHz (CCLK) maximum. However, if CCLK is 8MHz or slower, the CLKLP SFR bit (AUXR1.7) can be set to a '1' to lower the power consumption further. On any reset, CLKLP is '0' allowing highest performance. This bit can then be set in software if CCLK is running at 8MHz or slower.

3. INTERRUPTS

The **P89LPC912/913/914** use a four priority level interrupt structure. This allows great flexibility in controlling the handling of the many interrupt sources. The **P89LPC913** and **P89LPC914** devices support 9 interrupt sources: timers 0 and 1,serial port Tx, serial port Rx, combined serial port Rx/Tx, brownout detect, watchdog/ realtime clock, keyboard, comparator, and SPI.

The **P89LPC912** supports 7 interrupt sources: timers 0 and 1, brownout detect, watchdog/ realtime clock, keyboard, comparator, and SPI.

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable registers IEN0 or IEN1. The IEN0 register also contains a global enable bit, EA, which enables all interrupts.

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority registers IP0, IP0H, IP1, and IP1H. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the start of an instruction, the request of higher priority level is serviced.

If requests of the same priority level are pending at the start of an instruction, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve pending requests of the same priority level.

Table summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from a Power down mode.

INTERRUPT PRIORITY STRUCTURE

There are four SFRs associated with the four interrupt levels: IP0, IP0H, IP1, IP1H. Every interrupt has two bits in IPx and IPxH (x = 0,1) and can therefore be assigned to one of four levels, as shown in Table .

Table 3-1: Interrupt priority level

Priority Bits		Interrupt Priority Level
IPxH	IPx	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

Table 3-2: Summary of Interrupts - P89LPC912

Description	Interrupt Flag Bit(s)	Table 3-3: V ector Address	Interrupt Enable Bit(s)	Interrupt Priority	Arbitration Ranking	Power down Wakeup
Timer 0 Interrupt	TF0	000Bh	ET0 (IEN0.1)	IP0H.1, IP0.1	4	No
Timer 1 Interrupt	TF1	001Bh	ET1 (IEN0.3)	IP0H.3, IP0.3	10	No
Brownout Detect	BOF	002Bh	EBO (IEN0.5)	IP0H.5, IP0.5	2	Yes
Watchdog Timer/Real-time Clock	WDOVF/ RTCF	0053h	EWDRT (IEN0.6)	IP0H.6, IP0.6	3	Yes
KBI Interrupt	KBIF	003Bh	EKBI (IEN1.1)	IP1H.1, IP1.1	8	Yes
Comparators 1/2 interrupt	CMF1/CMF2	0043h	EC (IEN1.2)	IP1H.2, IP1.2	11	Yes
SPI interrupt	SPIF	004Bh	ESPI(IEN1.3)	IP1H.3, IP1.3	14	No

Table 3-4: Summary of Interrupts - P89LPC913/914

Description	Interrupt Flag Bit(s)	Vector Address	Interrupt Enable Bit(s)	Interrupt Priority	Arbitration Ranking	Power down Wakeup
Timer 0 Interrupt	TF0	000Bh	ET0 (IEN0.1)	IP0H.1, IP0.1	4	No
Timer 1 Interrupt	TF1	001Bh	ET1 (IEN0.3)	IP0H.3, IP0.3	10	No
Serial Port Tx and Rx[1,3]	TI & RI	0023h	ES/ESR (IEN0.4)	IP0H.4, IP0.4	13	No
Serial Port Rx[1,3]	RI					
Brownout Detect	BOF	002Bh	EBO (IEN0.5)	IP0H.5, IP0.5	2	Yes
Watchdog Timer/Real-time Clock	WDOVF/ RTCF	0053h	EWDRT (IEN0.6)	IP0H.6, IP0.6	3	Yes
KBI Interrupt	KBIF	003Bh	EKBI (IEN1.1)	IP1H.1, IP1.1	8	Yes
Comparators 1/2 interrupt	CMF1/CMF2	0043h	EC (IEN1.2)	IP1H.2, IP1.2	11	Yes
SPI interrupt	SPIF	004Bh	ESPI(IEN1.3)	IP1H.3, IP1.3	14	No
Serial Port Tx[2]	TI	006Bh	EST (IEN1.6)	P1H.6, IP1.6	12	No

1. SSTAT.5 = 0 selects combined Serial Port (UART) Tx and Rx interrupt; SSTAT.5 = 1 selects Serial Port Rx interrupt only (Tx interrupt will be different, see Note 3 below).
2. This interrupt is used as Serial Port (UART) Tx interrupt if and only if SSTAT.5 = 1, and is disabled otherwise.
3. If SSTAT.0 = 1, the following Serial Port additional flag bits can cause this interrupt: FE, BR, OE

EXTERNAL INTERRUPT INPUTS

The **P89LPC912/913/914** have a Keypad Interrupt function (see Keypad Interrupt (KBI) on page 87). This can be used as an external interrupt input. If enabled when the **P89LPC912/913/914** is put into Power down or Idle mode, the keypad interrupt will cause the processor to wake up and resume operation. Refer to the section on Power Reduction Modes for details.

EXTERNAL INTERRUPT PIN GLITCH SUPPRESSION

All of the **P89LPC912/913/914** pins, except P1.2, have glitch suppression circuits to reject short glitches (please refer to the **P89LPC912/913/914** datasheet, AC Electrical Characteristics for glitch filter specifications) .

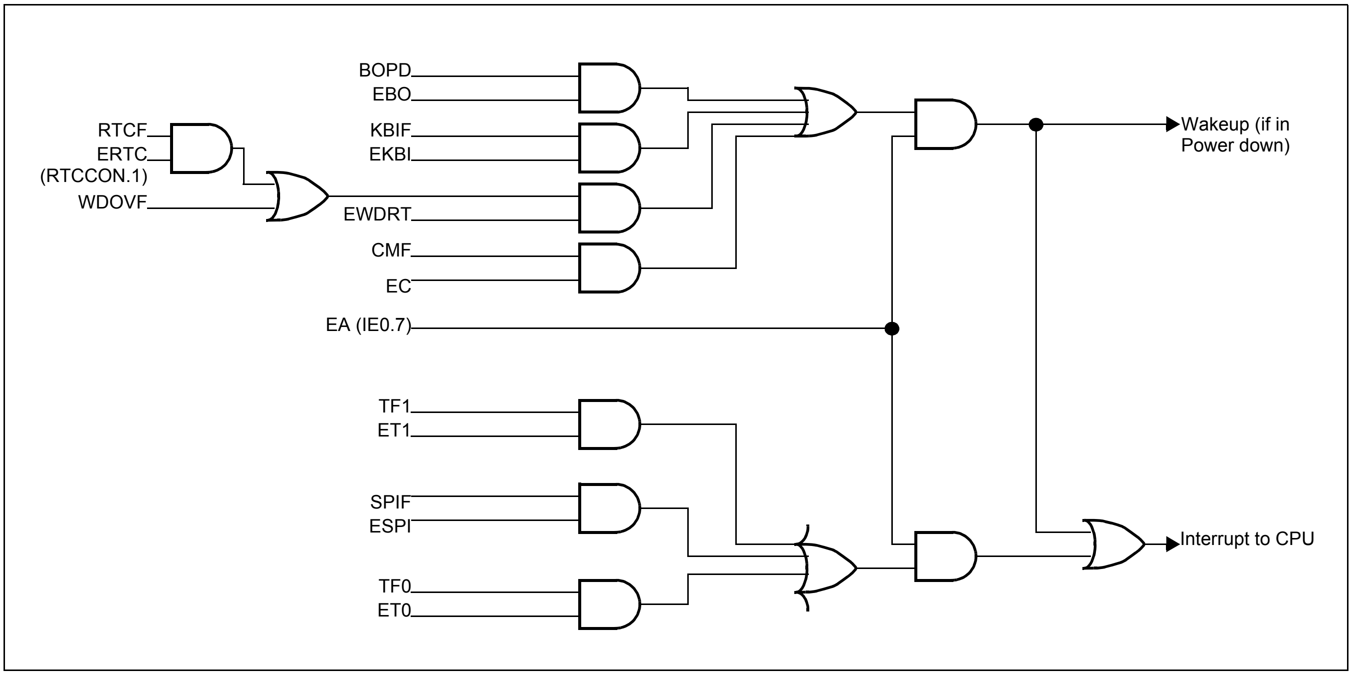


Figure 3-1: Interrupt sources, enables, and Power down Wake-up sources - P89LPC912

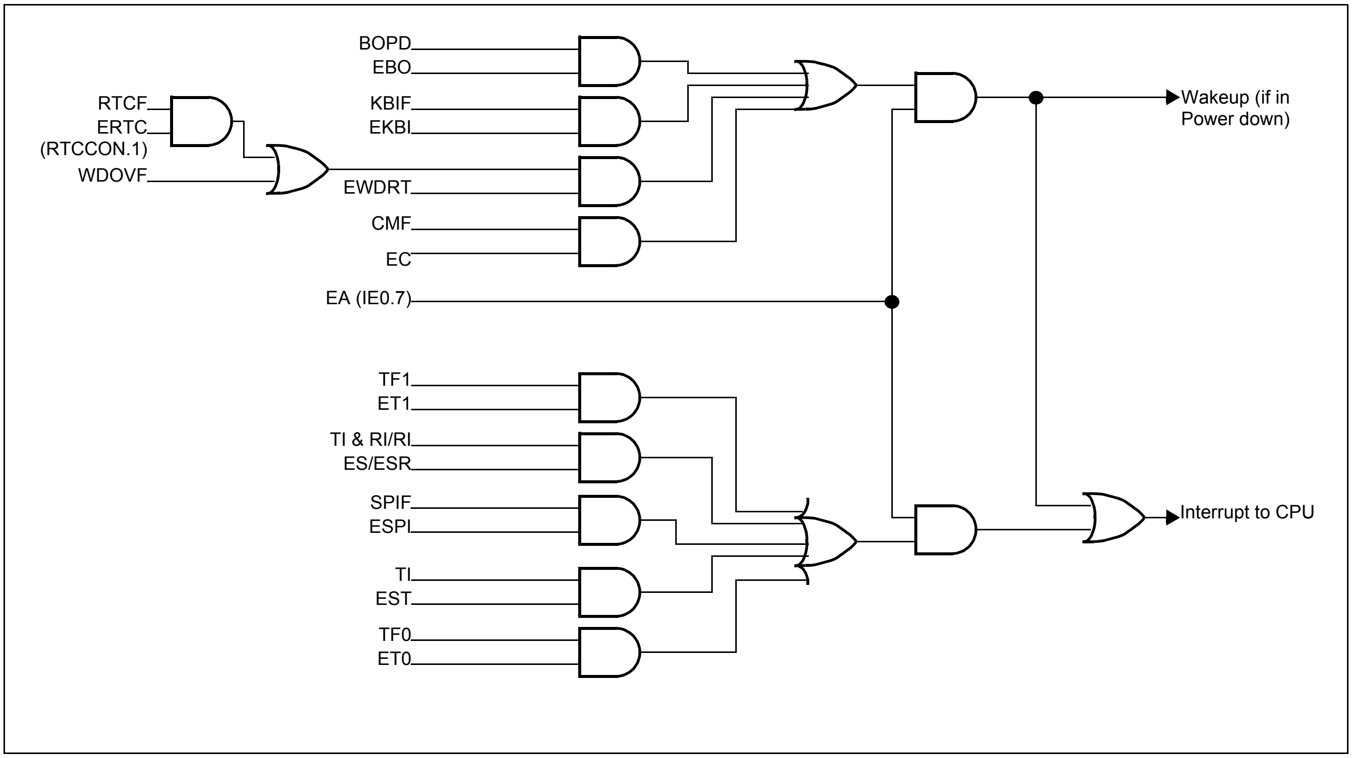


Figure 3-2: InterruptsSources, enables, and Power down Wake-up sources - P89LPC913,P89LPC914

4. I/O PORTS

The **P89LPC912/913/914** has between 9 and 12 I/O pins. The exact number of I/O pins available depends on the clock and reset options chosen:

Table 4-1: umber of I/O Pins Available

Clock Source	Reset Option	Number of I/O Pins
		14-Pin Package
On-chip oscillator or watchdog oscillator	No external reset(except during power-up)	12
	External $\overline{RST}$ pin supported	11
External clock input (P89LPC912/913)	No external reset(except during power-up)	11
	External $\overline{RST}$ pin supported	10
Low/medium/high speed oscillator (external crystal or resonator) (P89LPC912/913)	No external reset(except during power-up)	10
	External $\overline{RST}$ pin supported	9

PORT CONFIGURATIONS

All but one I/O port pin on the **P89LPC912/913/914** may be configured by software to one of four types on a pin-by-pin basis, as shown in Table . These are: quasi-bidirectional (standard 80C51 port outputs), push-pull, open drain, and input-only. Two configuration registers for each port select the output type for each port pin. P1.5 ($\overline{RST}$) can only be an input and cannot be configured.

Table 4-2: Port Output Configuration Settings

PxM1.y	PxM2.y	Port Output Mode
0	0	Quasi-bidirectional
0	1	Push-Pull
1	0	Input Only (High Impedance)
1	1	Open Drain

QUASI-BIDIRECTIONAL OUTPUT CONFIGURATION

Quasi-bidirectional outputs can be used both as an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is driven low, it is driven strongly and able to sink a large current. There are three pullup transistors in the quasi-bidirectional output that serve different purposes.

One of these pullups, called the "very weak" pullup, is turned on whenever the port latch for the pin contains a logic 1. This very weak pullup sources a very small current that will pull the pin high if it is left floating.

A second pullup, called the "weak" pullup, is turned on when the port latch for the pin contains a logic 1 and the pin itself is also at a logic 1 level. This pullup provides the primary source current for a quasi-bidirectional pin that is outputting a 1. If this pin is pulled low by an external device, this weak pullup turns off, and only the very weak pullup remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to overpower the weak pullup and pull the port pin below its input threshold voltage.

The third pullup is referred to as the "strong" pullup. This pullup is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port latch changes from a logic 0 to a logic 1. When this occurs, the strong pullup turns on for two CPU clocks quickly pulling the port pin high .

The quasi-bidirectional port configuration is shown in Figure 4-1.

Although the **P89LPC912/913/914** is a 3V device the pins are 5V-tolerant (except for XTAL1 and XTAL2). If 5V is applied to a pin configured in quasi-bidirectional mode, there will be a current flowing from the pin to V_{DD} causing extra power consumption. Therefore, applying 5V to pins configured in quasi-bidirectional mode is discouraged.

A quasi-bidirectional port pin has a Schmitt-triggered input that also has a glitch suppression circuit. Please refer to the **P89LPC912/913/914** datasheet, AC Characteristics for glitch filter specifications)

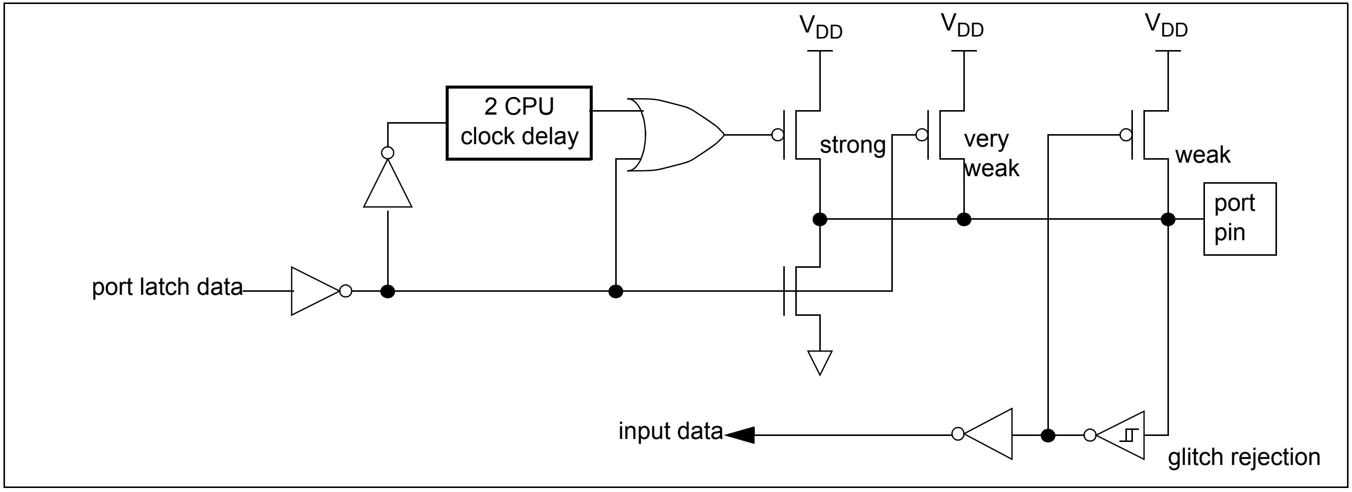


Figure 4-1: Quasi-Bidirectional Output

OPEN DRAIN OUTPUT CONFIGURATION

The open drain output configuration turns off all pullups and only drives the pulldown transistor of the port pin when the port latch contains a logic 0. To be used as a logic output, a port configured in this manner must have an external pullup, typically a resistor tied to V_{DD}. The pulldown for this mode is the same as for the quasi-bidirectional mode.

The open drain port configuration is shown in Figure 4-2.

An open drain port pin has a Schmitt-triggered input that also has a glitch suppression circuit (please refer to the **P89LPC912/913/914** datasheet, AC Characteristics for glitch filter specifications).

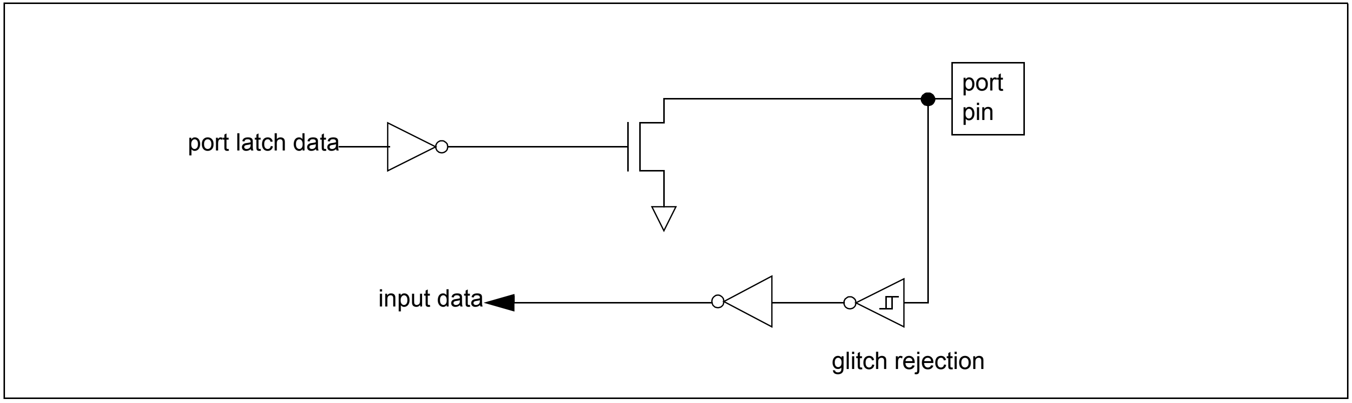


Figure 4-2: Open Drain Output

INPUT-ONLY CONFIGURATION

The input port configuration is shown in Figure 4-3. It is a Schmitt-triggered input that also has a glitch suppression circuit (please refer to the **P89LPC912/913/914** datasheet, AC Characteristics for glitch filter specifications)

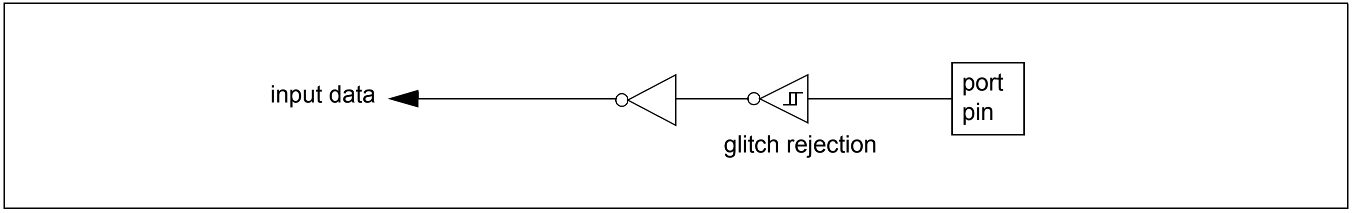


Figure 4-3: Input Only

PUSH-PULL OUTPUT CONFIGURATION

The push-pull output configuration has the same pulldown structure as both the open drain and the quasi-bidirectional output modes, but provides a continuous strong pullup when the port latch contains a logic 1. The push-pull mode may be used when more source current is needed from a port output.

The push-pull port configuration is shown in Figure 4-4.

A push-pull port pin has a Schmitt-triggered input that also has a glitch suppression circuit (please refer to the **P89LPC912/913/914** datasheet, AC Characteristics for glitch filter specifications)

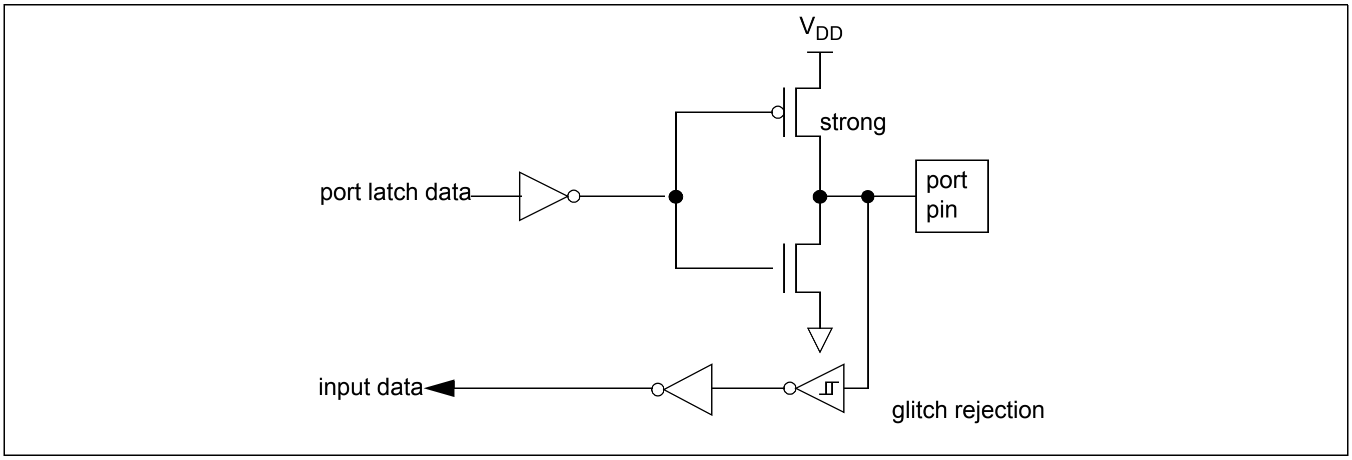


Figure 4-4: Push-Pull Output

PORT 0 ANALOG FUNCTIONS

The **P89LPC912/913/914** incorporates an analog comparator. In order to give the best analog performance and minimize power consumption, pins that are being used for analog functions must have both the digital outputs and digital inputs disabled.

Digital outputs are disabled by putting the port pins into the input-only mode as described in the Port Configurations section (see Table).

Digital inputs on Port 0 may be disabled through the use of the PT0AD register. On any reset, the PT0AD bits default to '0's to enable digital functions.

Port Pin	Configuration SFR Bits		Alternate Usage	Notes
	PxM1.y	PxM2.y		
P0.2	P0M1.2	P0M2.2	CIN2A,KBI2	
P0.4	P0M1.4	P0M2.4	KBI4,CIN1A	Refer to section "Port 0 Analog Functions" for usage as analog inputs CIN1A and CMPREF.
P0.5	P0M1.5	P0M2.5	KBI5,CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6,CMP1	
P1.0	P1M1.0	P1M2.0	TxD	
P1.1	P1M1.1	P1M2.1	RxD	
P1.2	P1M1.2	P1M2.2	T0	
P1.5	not configurable		$\overline{\text{RST}}$	Input only. Usage as general purpose input or $\overline{\text{RST}}$ is determined by User Configuration Bit RPD (UCFG1.6). Always a reset input during a power-on sequence.
P2.2	P2M1.2	P2M2.2	MOSI	
P2.3	P2M1.3	P2M2.3	MISO	
P2.4	P2M1.4	P2M2.4	$\overline{\text{SS}}$	
P2.5	P2M1.5	P2M2.5	SPICLK	
P3.0	P3M1.0	P3M2.0	XTAL2,CLKOUT	
P3.1	P3M1.1	P3M2.1	XTAL1	

Table 4-3: Port Output Configuration.

ADDITIONAL PORT FEATURES

After power-up, all pins are in Input-Only mode. **Please note that this is different from the LPC76x series of devices.**

- After power-up, all I/O pins except P1.5, may be configured by software.
- Pin P1.5 is input only.
- Every output on the **P89LPC912/913/914** has been designed to sink typical LED drive current. However, there is a maximum total output current for all ports which must not be exceeded. Please refer to the **P89LPC912/913/914** datasheet for detailed specifications.

All ports pins that can function as an output have slew rate controlled outputs to limit noise generated by quickly switching output signals. The slew rate is factory-set to approximately 10 ns rise and fall times.

5. TIMERS 0 AND 1

The **P89LPC912/913/914** have two general-purpose counter/timers which are similar to the 80C51 Timer 0 and Timer 1. Timer 0 of the **P89LPC912/914** can be configured to operate either as a timer or event counter (see Figure 5-1). An option to automatically toggle the T0 pin upon timer overflow has been added. Timer 1 of the **P89LPC912/914** and both Timer 0 and Timer 1 of the **P89LPC913** may only function as timers.

In the "Timer" function, the timer is incremented every PCLK.

In the "Counter" function, the Timer 0 register is incremented in response to a 1-to-0 transition on the external input pin, T0, which is sampled once during every machine cycle. When the pin is high during one cycle and low in the next cycle, the count is incremented. The new count value appears in the register during the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (4 CPU clocks) to recognize a 1-to-0 transition, the maximum count rate is 1/4 of the CPU clock frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

The "Timer" or "Counter" function is selected by control bit T0C/$\overline{T}$ in the Special Function Register TMOD. Timer 0 and Timer 1 of the **P89LPC913**, and Timer 1 of the **P89LPC912/P89LPC914** have four operating modes (modes 0, 1, 2, and 3), which are selected by bit-pairs (TnM1, TnM0) in TMOD. Modes 0, 1, 2 and 3 are the same for both Timers. Mode 3 is different. The operating modes are described later in this section. In addition to these modes, Timer 0 of the **P89LPC912/914** has mode 6. Additionally the T0M2 mode bit in TAMOD is used to specify modes with Timer 0 of the **P89LPC912/914**.

TMOD

Address: 89h

Not bit addressable

Reset Source(s): Any source

Reset Value: 00000000B

7	6	5	4	3	2	1	0
-	-	T1M1	T1M0	-	T0C/$\overline{T}$	T0M1	T0M0

BIT	**SYMBOL**	**FUNCTION**
TMOD.7	-	Reserved.
TMOD.6	-	Reserved.
TMOD.5, 4	T1M1,T1M0	Mode Select for Timer 1. These bits are used to determine the Timer 1 mode (see Figure 5-2).
TMOD.3	-	Reserved.
TMOD.2	T0C/$\overline{T}$	Timer or Counter Selector for Timer 0. Cleared for Timer operation (input from CCLK). Set for Counter operation (input from T0 input pin).**P89LPC912/914..**
TMOD.1, 0	T0M1,T0M0	Mode Select for Timer 0. These bits are used to determine the Timer 0 mode (see Figure 5-2). On the **P89LPC912/914** these bits are used with the T0M2 bit in the TAMOD register to determine the Timer 0 mode (see Figure 5-2).

Figure 5-1: Timer/Counter Mode Control register (TMOD)

TAMOD - P89LPC912/914

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	T0M2

Address: 8Fh

Not bit addressable

Reset Source(s): Any reset

Reset Value: xxx0xxx0B

BIT	**SYMBOL**	**FUNCTION**
TAMOD.7-1	-	Reserved for future use. Should not be set to 1 by user programs.
TAMOD.0	T0M2	Mode Select bit 2 for Timer 0. Used with T0M1 and T0M0 in the TMOD register to determine Timer 0 mode **(P89LPC912/914).**
	TnM2-TnM0	Timer Mode
	0 0 0	8048 Timer "TLn" serves as 5-bit prescaler. (Mode 0)
	0 0 1	16-bit Timer/Counter "THn" and "TLn" are cascaded; there is no prescaler. (Mode 1)
	0 1 0	8-bit auto-reload Timer/Counter. THn holds a value which is loaded into TLn when it overflows. (Mode 2)
	0 1 1	Timer 0 is a dual 8-bit Timer/Counter in this mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only, controlled by the Timer 1 control bits (see text). Timer 1 in this mode is stopped. (Mode 3)
	1 0 0	Reserved. User must not configure to this mode.
	1 0 1	Reserved. User must not configure to this mode.
	1 1 0	PWM mode (see section "Mode 6 - P89LPC912/914").
	1 1 1	Reserved. User must not configure to this mode.

Figure 5-2: Timer/Counter Auxiliary Mode Control register (TAMOD)

MODE 0

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 5-4 shows Mode 0 operation.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TFn. The count input is enabled to the Timer when TRn = 1. TRn is a control bit in the Special Function Register TCON (Figure 5-3).

The 13-bit register consists of all 8 bits of THn and the lower 5 bits of TLn. The upper 3 bits of TLn are indeterminate and should be ignored. Setting the run flag (TRn) does not clear the registers.

Mode 0 operation is the same for Timer 0 and Timer 1. See Figure 5-4.

Mode 1 is the same as Mode 0, except that all 16 bits of the timer register (THn and TLn) are used. See Figure 5-5.

MODE 2

Mode 2 configures the Timer register as an 8-bit Counter (TLn) with automatic reload, as shown in Figure 5-6. Overflow from TLn not only sets TFn, but also reloads TLn with the contents of THn, which must be preset by software. The reload leaves THn unchanged. Mode 2 operation is the same for Timer 0 and Timer 1.

MODE 3

When Timer 1 is in Mode 3 it is stopped. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate 8-bit counters. The logic for Mode 3 on Timer 0 is shown in Figure 5-7. TL0 uses the Timer 0 control bits: TR0 and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications that require an extra 8-bit timer.

Note: When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it into and out of its own Mode 3. It can still be used by the serial port as a baud rate generator (**P89LPC913,P89LPC914**), or in any application not requiring an interrupt.

MODE 6 - P89LPC912/914

In this mode, Timer 0 can be changed to a PWM with a full period of 256 timer clocks (see Figure 5-8). Its structure is similar to mode 2, except that:

- TF0 is set and cleared in hardware;
- The low period of the TF0 is in TH0, and should be between 1 and 254, and;
- The high period of the TF0 is always 256-TH0.
- Loading TH0 with 00h will force the T0 pin high, loading TH0 with FFh will force the T0 pin low.

Note that an interrupt can still be enabled on the low to high transition of TF0, and that TF0 can still be cleared in software as in any other modes.

TCON
Address: 88h
Bit addressable
Reset Source(s): Any reset
Reset Value: 00000000B

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	-	-	-	-

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer overflow. Cleared by hardware when the interrupt is processed, or by software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer 1 on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to the interrupt routine, or by software. (except in mode 6, see above, when it is cleared in hardware)
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
TCON.3	-	Reserved for future use. Should not be set to 1 by user programs.
TCON.2	-	Reserved for future use. Should not be set to 1 by user programs.
TCON.1	-	Reserved for future use. Should not be set to 1 by user programs.
TCON.0	-	Reserved for future use. Should not be set to 1 by user programs.

Figure 5-3: Timer/Counter Control register (TCON)

TIMERS 0 AND 1

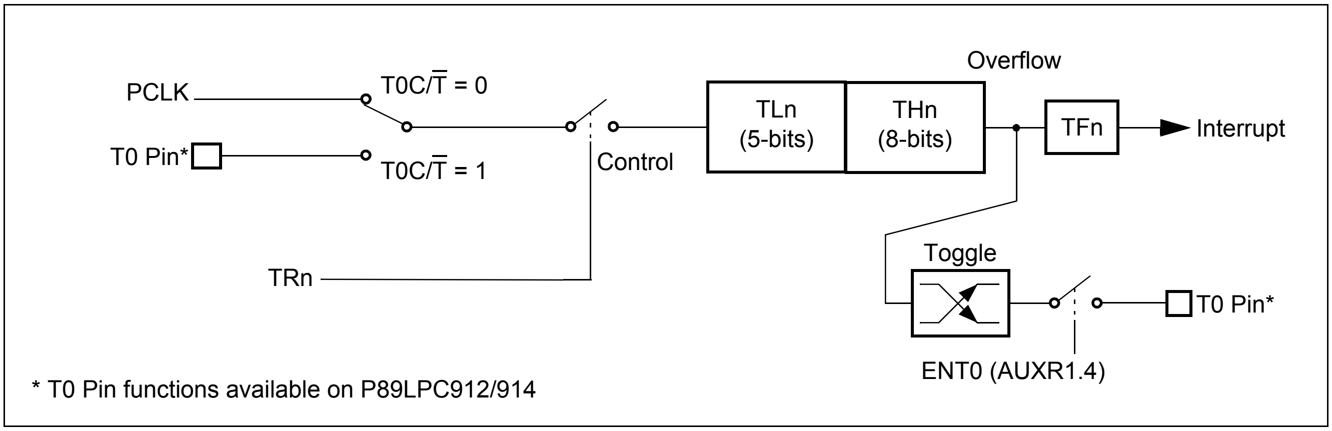


Figure 5-4: Timer/Counter 0 or 1 in Mode 0 (13-bit counter)

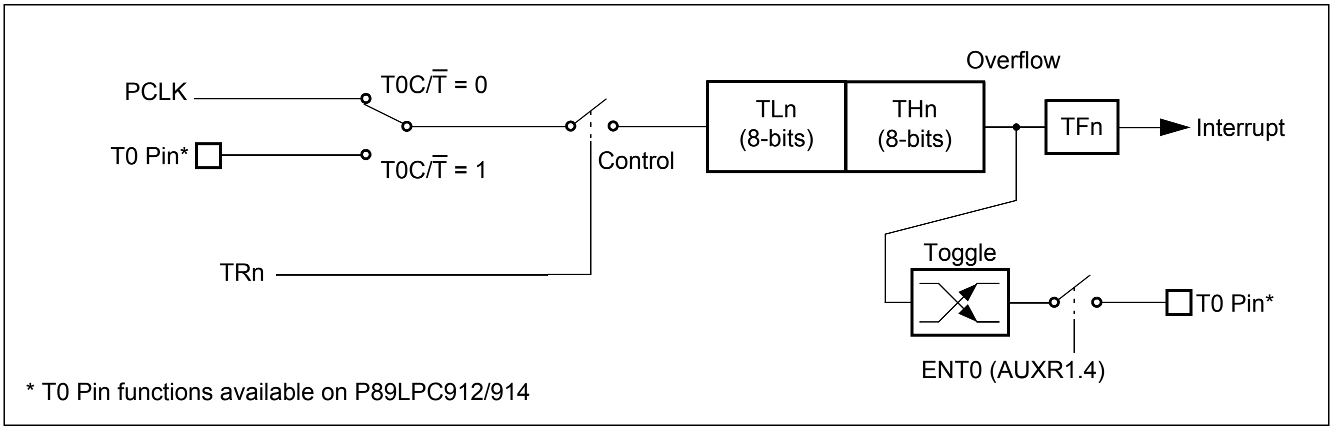


Figure 5-5: Timer/Counter 0 or 1 in Mode 1 (16-bit counter)

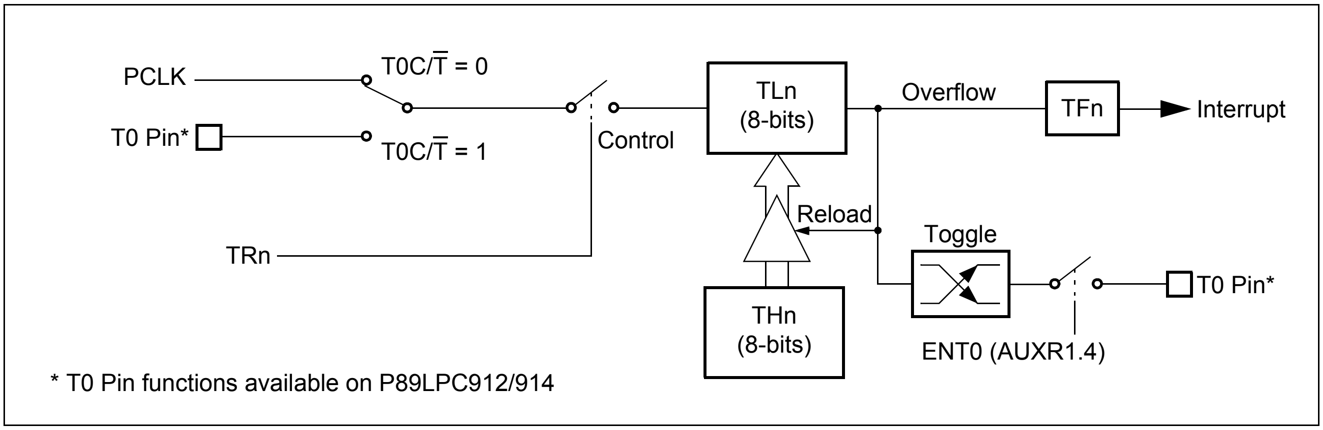


Figure 5-6: Timer/Counter 0 or 1 in Mode 2 (8-bit auto-reload)

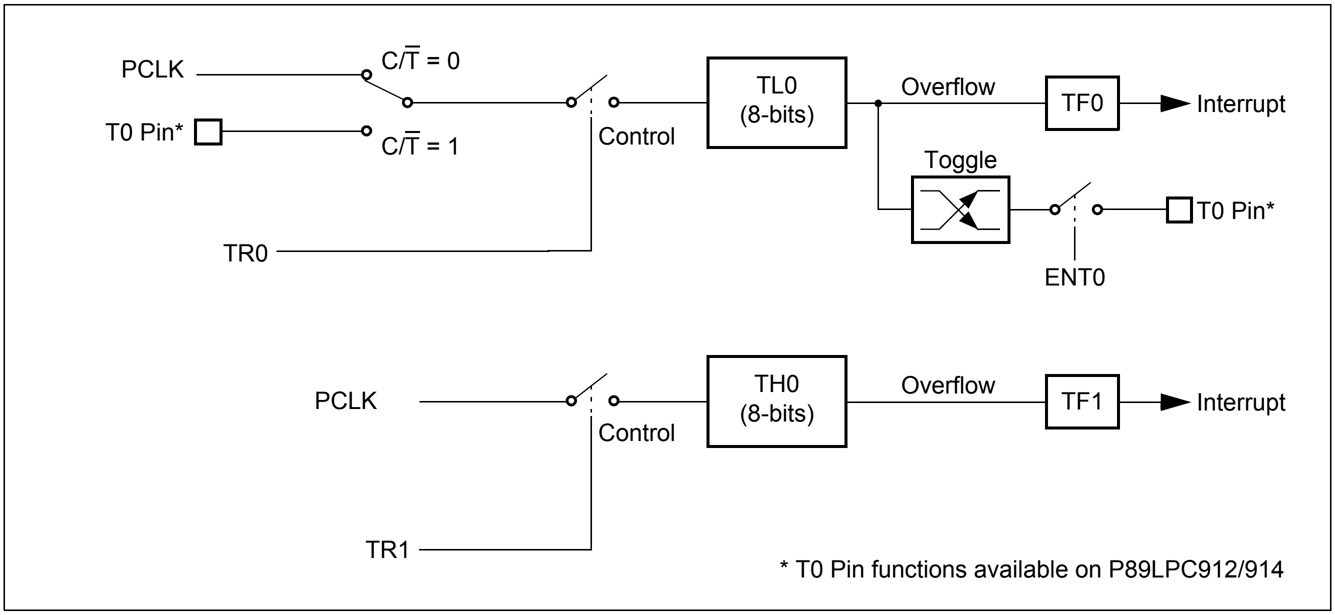


Figure 5-7: Timer/Counter 0 Mode 3 (two 8-bit counters)

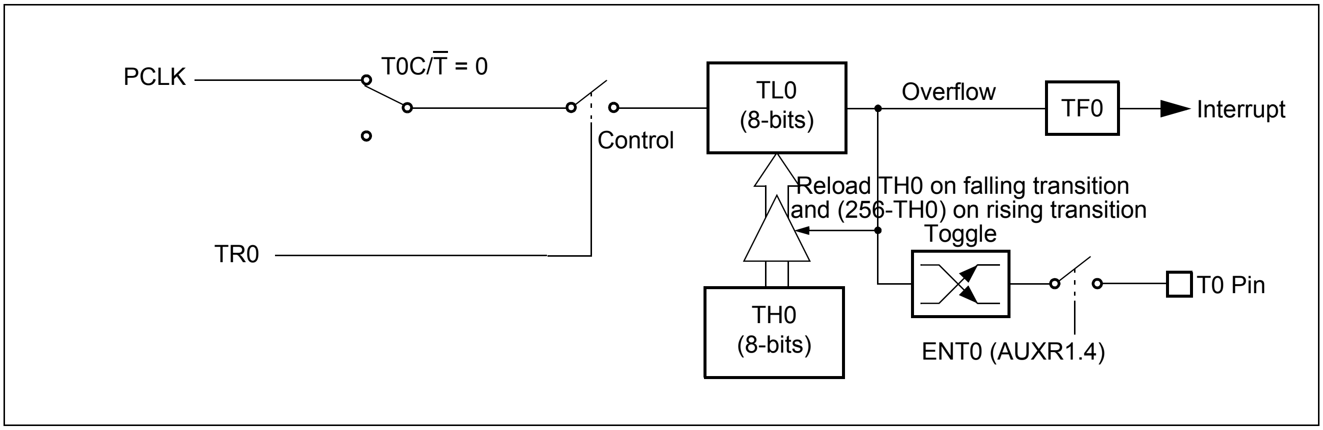


Figure 5-8: Timer/Counter 0 in Mode 6 (PWM auto-reload), P89LPC912/914.

TIMER OVERFLOW TOGGLE OUTPUT - P89LPC912/914

Timer 0 can be configured to automatically toggle the T0 pin whenever the timer overflow occurs. This function is enabled by control bit ENT0 in the AUXR1 register. The port output will be a logic 1 prior to the first timer overflow when this mode is turned on. In order for this mode to function, the T0C/T̄ bit must be cleared selecting PCLK as the clock source for the timer.

6. REAL-TIME CLOCK/SYSTEM TIMER

The **P89LPC912/913/914** has a simple Real-time clock/system timer that allows a user to continue running an accurate timer while the rest of the device is powered down. The Real-time clock can be an interrupt or a wake-up source (see Figure 6-1). The Real-time clock is a 23-bit down counter.

REAL-TIME CLOCK SOURCE

On the **P89LPC912/913** devices the clock source for this counter can be either CCLK or the XTAL1-2 oscillator (XCLK) . On the **P89LPC914** the clock source for this counter is CCLK. Please refer to Figure 2-3 "Block Diagram of Oscillator Control - P89LPC912" in section "Clocks" on page 29. CCLK can have either the XTAL1-2 oscillator, the internal RC oscillator, or the Watchdog oscillator as its clock source. If the XTAL1-2 oscillator is used for producing CCLK, the RTC will use either the XTAL1-2 oscillator’s output or CCLK as its clock source. The possible clocking combinations are shown in Table , below.

There are three SFRs used for the RTC:

- RTCCON - Real-time clock control.
- RTCH - Real-time clock counter reload high (bits 22-15).
- RTCL - Real-time clock counter reload low (bits 14-7).

The Real-time clock/system timer can be enabled by setting the RTCEN (RTCCON.0) bit. The Real-time clock is a 23-bit down counter (initialized to all 0’s when RTCEN = 0) that is comprised of a 7-bit prescaler and a 16-bit loadable down counter. When RTCEN is written with ’1’, the counter is first loaded with (RTCH,RTCL,’1111111’) and will count down. When it reaches all 0’s, the counter will be reloaded again with (RTCH,RTCL,’1111111’) and a flag - RTCF (RTCCON.7) - will be set.

Any write to RTCH and RTCL in-between the Real-time clock reloading will not cause reloading of the counter. When the current count terminates, the contents of RTCH and RTCL will be loaded into the counter and the new count will begin. An immediate reload of the counter can be forced by clearing the RTCEN bit to ’0’ and then setting it back to ’1’ .

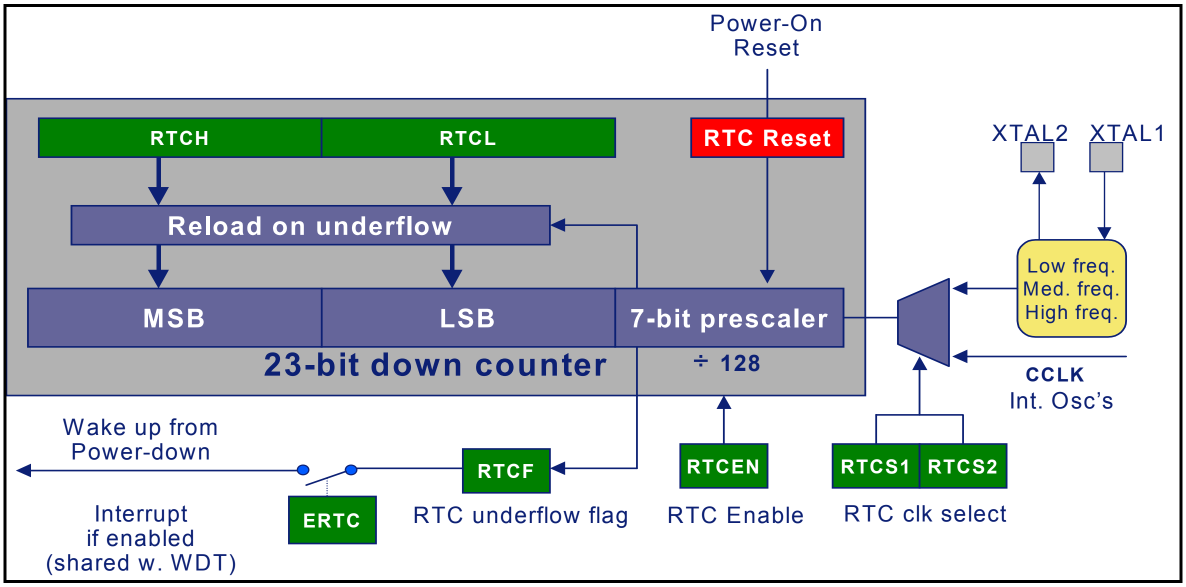


Figure 6-1: Real-time clock/system timer Block Diagram

Table 6-1: Real-time Clock/System Timer Clock Source - P89LPC912/913

FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTCS1:0	CCLK Frequency	RTC Clock Frequency
0	0	0	00	High frequency crystal/DIVM	High frequency crystal (XCLK)
			01		
			10		
			11		High frequency crystal/DIVM (CCLK)
0	0	1	00	Medium frequency crystal/DIVM	Medium frequency crystal (XCLK)
			01		
			10		
			11		Medium frequency crystal/ DIVM (CCLK)
0	1	0	00	Low frequency crystal/DIVM	Low frequency crystal (XCLK)
			01		
			10		
			11		Low frequency crystal/DIVM (CCLK)
0	1	1	00	RC Oscillator/DIVM	High frequency crystal (XCLK)
			01		Medium frequency crystal (XCLK)
			10		Low frequency crystal (XCLK)
			11		RC Oscillator/DIVM (CCLK)
1	0	0	00	WDT Oscillator/DIVM	High frequency crystal (XCLK)
			01		Medium frequency crystal (XCLK)
			10		Low frequency crystal (XCLK)
			11		WDT Oscillator/DIVM (CCLK)
1	0	1	xx	undefined	
1	1	0			
1	1	1	00	external clock/DIVM	external clock (XCLK)
			01		
			10		
			11		external clock/DIVM (CCLK)

Table 6-2: : Real-time Clock/System Timer Clock Source - P89LPC914

FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTCS1:0	CCLK Frequency	RTC Clock Frequency
0	0	0	x	undefined	
0	0	1			
0	1	0			
0	1	1	00	RC Oscillator/DIVM	undefined
			01		
			10		
			11		RC Oscillator/DIVM (CCLK)
1	0	0	00	WDT Oscillator/DIVM	undefined
			01		
			10		
			11		WDT Oscillator/DIVM (CCLK)
1	0	1	xx	undefined	
1	1	0			
1	1	1			

CHANGING RTCS1-0

RTCS1-0 cannot be changed if the RTC is currently enabled (RTCCON.0 =1). Setting RTCEN and updating RTCS1-0 may be done in a single write to RTCCON. However, if RTCEN = 1, this bit must first be cleared before updating RTCS1-0

REAL-TIME CLOCK INTERRUPT/WAKE UP

If ERTC (RTCCON.1), EWDRT (IEN0.6) and EA (IEN0.7) are set to '1', RTCF can be used as an interrupt source. This interrupt vector is shared with the watchdog timer. It can also be a source to wake up the device.

RESET SOURCES AFFECTING THE REAL-TIME CLOCK

Only power-on reset will reset the Real-time Clock and its associated SFRs to their default state.

RTCCON

Address: D1h

Not bit addressable

Reset Source(s): Power-up only

Reset Value: 011xxx00B

7	6	5	4	3	2	1	0
RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN

BIT	**SYMBOL**	**FUNCTION**
RTCCON.7	RTCF	Real-time Clock Flag. This bit is set to ’1’ when the 23-bit Real-time clock reaches a count of ’0’. It can be cleared in software.
RTCCON.6-5	RTCS1-0	Real-time Clock source select (see Table ,Table).
RTCCON.4-2	-	Reserved for future use. Should not be set to 1 by user programs.
RTCCON.1	ERTC	Real-time Clock interrupt enable. The Real-time clock shares the same interrupt as the watchdog timer. Note that if the user configuration bit WDTE (UCFG1.7) is ’0’, the watchdog timer can be enabled to generate an interrupt. Users can read the RTCF (RTCCON.7) bit to determine whether the Real-time clock caused the interrupt.
RTCCON.0	RTCEN	Real-time Clock enable. The Real-time clock will be enabled if this bit is ’1’. Note that this bit will not Power down the Real-time Clock. The RTCPD bit (PCONA.7) if set, will Power down and disable this block regardless of RTCEN.

Figure 6-2: RTCCON Register

7. POWER MONITORING FUNCTIONS

The **P89LPC912/913/914** incorporates power monitoring functions designed to prevent incorrect operation during initial power-on and power loss or reduction during operation. This is accomplished with two hardware functions: Power-on Detect and Brownout Detect.

BROWNOUT DETECTION

The Brownout Detect function determines if the power supply voltage drops below a certain level. The default operation for a Brownout Detection is to cause a processor reset. However, it may alternatively be configured to generate an interrupt by setting the BOI (PCON.4) bit and the EBO (IEN0.5) bit.

Enabling and disabling of Brownout Detection is done via the BOPD (PCON.5) bit, bit field PMOD1-0 (PCON.1-0) and user configuration bit BOE (UCFG1.5). If BOE is in an unprogrammed state, brownout is disabled regardless of PMOD1-0 and BOPD. If BOE is in a programmed state, PMOD1-0 and BOPD will be used to determine whether Brownout Detect will be disabled or enabled. PMOD1-0 is used to select the power reduction mode. If PMOD1-0 = '11', the circuitry for the Brownout Detection is disabled for lowest power consumption. BOPD defaults to '0', indicating brownout detection is enabled on power-on if BOE is programmed.

If Brownout Detection is enabled, the operating voltage range for V_{DD} is 2.7V-3.6V, and the brownout condition occurs when V_{DD} falls below the Brownout trip voltage, V_{BO} (see D.C. Electrical Characteristics), and is negated when V_{DD} rises above V_{BO}. If Brownout Detection is disabled, the operating voltage range for V_{DD} is 2.4V-3.6V. If the **P89LPC912/913/914** device is to operate with a power supply that can be below 2.7V, BOE should be left in the unprogrammed state so that the device can operate at 2.4V, otherwise continuous brownout reset may prevent the device from operating.

If Brownout Detect is enabled (BOE programmed, PMOD1-0 $\neq$ '11', BOPD = 0), BOF (RSTSRC.5) will be set when a brownout is detected, regardless of whether a reset or an interrupt is enabled, . BOF will stay set until it is cleared in software by writing '0' to the bit. Note that if BOE is unprogrammed, BOF is meaningless. If BOE is programmed, and a initial power-on occurs, BOF will be set in addition to the power-on flag (POF - RSTSRC.4).

For correct activation of Brownout Detect, certain V_{DD} rise and fall times must be observed. Please see the datasheet for specifications.

Table 7-1: Brownout Options

BOE (UCFG1.5)	PMOD1-0 (PCON.1-0)	BOPD (PCON.5)	BOI (PCON.4)	EBO (IEN0.5)	EA (IEN0.7)	Description
0 (erased)	XX	X	X	X	X	Brownout disabled. V_{DD} operating range is 2.4V-3.6V.
1 (programmed)	11 (total power down)	X	X	X	X	
	≠ 11 (any mode other than total power down)	1 (brownout detect powered down)	X	X	X	Brownout disabled. V_{DD} operating range is 2.4V-3.6V. However, BOPD is default to '0' upon power-up.
		0 (brownout detect active)	0 (brownout detect generates reset)	X	X	Brownout reset enabled. V_{DD} operating range is 2.7V-3.6V. Upon a brownout reset, BOF (RSTSRC.5) will be set to indicate the reset source. BOF can be cleared by writing '0' to the bit.
			1 (brownout detect generates an interrupt)	1 (enable brownout interrupt)	1 (global interrupt enable)	Brownout interrupt enabled. V_{DD} operating range is 2.7V-3.6V. Upon a brownout interrupt, BOF (RSTSRC.5) will be set. BOF can be cleared by writing '0' to the bit.
				0	X	Both brownout reset and interrupt disabled. V_{DD} operating range is 2.4V-3.6V. However, BOF (RSTSRC.5) will be set when V_{DD} falls to the Brownout Detection trip point. BOF can be cleared by writing '0' to the bit.
				X	0	

POWER-ON DETECTION

The Power-On Detect has a function similar to the Brownout Detect, but is designed to work as power initially comes up, before the power supply voltage reaches a level where the Brownout Detect can function. The POF flag (RSTSRC.4) is set to indicate an initial power-on condition. The POF flag will remain set until cleared by software by writing '0' to the bit. Note that if BOE (UCFG1.5) is programmed, BOF (RSTSRC.5) will be set when POF is set. If BOE is unprogrammed, BOF is meaningless.

POWER REDUCTION MODES

The **P89LPC912/913/914** supports three different power reduction modes as determined by SFR bits PCON.1-0 (see Table 7-2)

Table 7-2: :Power Reduction Modes

PMOD1 (PCON.1)	PMOD0 (PCON.0)	Description
0	0	Normal Mode (Default) - no power reduction.
0	1	Idle Mode. The Idle mode leaves peripherals running in order to allow them to activate the processor when an interrupt is generated. Any enabled interrupt source or reset may terminate Idle mode.
1	0	Power down mode: The Power down mode stops the oscillator in order to minimize power consumption. The **P89LPC912/913/914** exits Power down mode via any reset, or certain interrupts - brownout Interrupt, keyboard, Real-time clock (system timer), watchdog, and comparator trips. Waking up by reset is only enabled if the corresponding reset is enabled, and waking up by interrupt is only enabled if the corresponding interrupt is enabled and the EA SFR bit (IEN0.7) is set. In Power down mode the internal RC oscillator is disabled unless both the RC oscillator has been selected as the system clock AND the RTC is enabled In Power down mode, the power supply voltage may be reduced to the RAM keep-alive voltage V_{RAM}. This retains the RAM contents at the point where Power down mode was entered. SFR contents are not guaranteed after V_{DD} has been lowered to V_{RAM}, therefore it is recommended to wake up the processor via Reset in this situation. V_{DD} must be raised to within the operating range before the Power down mode is exited. When the processor wakes up from Power down mode, it will start the oscillator immediately and begin execution when the oscillator is stable. Oscillator stability is determined by counting 1024 CPU clocks after start-up when one of the crystal oscillator configurations is used, or 256 clocks after start-up for the internal RC or external clock input configurations. Some chip functions continue to operate and draw power during Power down mode, increasing the total power used during Power down. These include: • Brownout Detect • Watchdog Timer if WDCLK (WDCON.0) is '1'. • Comparator (Note: Comparator can be powered down separately with PCONA.5 set to '1' and comparator disabled); • Real-time Clock/System Timer (and the crystal oscillator circuitry if this block is using it, unless RTCPD, i.e., PCONA.7 is '1').
1	1	Total Power down mode: This is the same as Power down mode except that the Brownout Detection circuitry and the voltage comparators are also disabled to conserve additional power. Note that a brownout reset or interrupt will not occur. Voltage comparator interrupts and Brownout interrupt cannot be used as a wakeup source.The internal RC oscillator is disabled unless both the RC oscillator has been selected as the system clock AND the RTC is enabled. The following are the wakeup options supported: • Watchdog Timer if WDCLK (WDCON.0) is '1'. Could generate Interrupt or Reset, either one can wake up the device • Keyboard Interrupt • Real-time Clock/System Timer (and the crystal oscillator circuitry if this block is using it, unless RTCPD, i.e., PCONA.7 is '1'). • Note: Using the internal RC-oscillator to clock the RTC during Power down may result in relatively high power consumption. Lower power consumption can be achieved by using an external low frequency clock when the Real-time Clock is running during Power down.

PCON

Address: 87h

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

7	6	5	4	3	2	1	0
SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0

BIT	SYMBOL	FUNCTION
PCON.7	SMOD1	Double Baud Rate bit for the serial port (UART) when Timer 1 is used as the baud rate source. When 1, the Timer 1 overflow rate is supplied to the UART. When 0, the Timer 1 overflow rate is divided by 2 before being supplied to the UART. **P89LPC913/914**(See Figure 8-2).
PCON.6	SMOD0	Framing Error Location (**P89LPC913/914**): -When 0, bit 7 of SCON is accessed as SM0 for the UART. -When 1, bit 7 of SCON is accessed as the framing error status (FE) for the UART. This bit also determines the location of the UART receiver interrupt RI (see description on RI in Figure 8-3).
PCON.5	BOPD	Brownout Detect Power down. When 1, Brownout Detect is powered down and therefore disabled. When 0, Brownout Detect is enabled. (Note: BOPD must be '0' before any programming or erasing commands can be issued. Otherwise these commands will be aborted.)
PCON.4	BOI	Brownout Detect Interrupt Enable. When 1, Brownout Detection will generate a interrupt . When 0, Brownout Detection will cause a reset.
PCON.3	GF1	General Purpose Flag 1. May be read or written by user software, but has no effect on operation.
PCON.2	GF0	General Purpose Flag 0. May be read or written by user software, but has no effect on operation.

Figure 7-1: Power Control Register (PCON)

PCONA

Address: B5H

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

7	6	5	4	3	2	1	0
RTCPD	-	VCPD	-	-	-	SPD	-

BIT	SYMBOL	FUNCTION
PCONA.7	RTCPD	Real-time Clock Power down: When '1', the internal clock to the Real-time Clock is disabled.
PCONA.6	-	Not used. Reserved for future use.
PCONA.5	VCPD	Analog Voltage Comparator Power down: When '1', the voltage comparator is powered down. User must disable the voltage comparator prior to setting this bit.
PCONA.4	-	Not used. Reserved for future use.
PCONA.3	-	Not used. Reserved for future use.
PCONA.2	-	Not used. Reserved for future use.
PCONA.1	SPD	Serial Port (UART) Power down: When '1', the internal clock to the UART is disabled. Note that in either Power down mode or Total Power down mode, the UART clock will be disabled regardless of this bit (**P89LPC913/914**).
PCONA.0	-	Not used. Reserved for future use.

Figure 7-2: Power Control Register (PCONA)

8. UART (P89LPC913,P89LPC914)

The **P89LPC913** and **P89LPC914** devices have an enhanced UART that is compatible with the conventional 80C51 UART, except that Timer 2 overflow cannot be used as a baud rate source. The UART does include an independent Baud Rate Generator. The baud rate can be selected from the CCLK (divided by a constant), Timer 1 overflow, or the independent Baud Rate Generator.

In addition to the baud rate generation, enhancements over the standard 80C51 UART include Framing Error detection, break detect, automatic address recognition, selectable double buffering and several interrupt options. The UART can be operated in 4 modes:

MODE 0

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted or received, LSB first. The baud rate is fixed at 1/16 of the CCLK.

MODE 1

10 bits are transmitted (through TxD) or received (through RxD): a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). When data is received, the stop bit is stored in RB8 in Special Function Register SCON. The baud rate is variable and is determined by the Timer 1 overflow rate or the Baud Rate Generator (see "Baud Rate Generator and Selection" section).

MODE 2

11 bits are transmitted (through TxD) or received (through RxD): start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. When data is received, the 9th data bit goes into RB8 in Special Function Register SCON and the stop bit is not saved. The baud rate is programmable to either 1/16 or 1/32 of the CCLK frequency, as determined by the SMOD1 bit in PCON.

MODE 3

11 bits are transmitted (through TxD) or received (through RxD): a start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable and is determined by the Timer 1 overflow rate or the Baud Rate Generator (see "Baud Rate Generator and Selection" section).

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

SFR SPACE

The UART SFRs are at the following locations:

Table 8-1: SFR Locations for UARTs

Register	Description	SFR Location
PCON	Power Control	87H
SCON	Serial Port (UART) Control	98H
SBUF	Serial Port (UART) Data Buffer	99H
SADDR	Serial Port (UART) Address	A9H
SADEN	Serial Port (UART) Address Enable	B9H
SSTAT	Serial Port (UART) Status	BAH
BRGR1	Baud Rate Generator Rate High Byte	BFH
BRGR0	Baud Rate Generator Rate Low Byte	BEH
BRGCON	Baud Rate Generator Control	BDH

BAUD RATE GENERATOR AND SELECTION

The enhanced UART has an independent Baud Rate Generator. The baud rate is determined by a value programmed into the BRGR1 and BRGR0 SFRs. The UART can use either Timer 1 or the baud rate generator output as determined by BRGCON.2-1 (see Figure 8-2). Note that Timer T1 is further divided by 2 if the SMOD1 bit (PCON.7) is cleared. The independent Baud Rate Generator uses CCLK.

UPDATING THE BRGR1 AND BRGR0 SFRS

The baud rate SFRs, BRGR1 and BRGR0 must only be loaded when the Baud Rate Generator is disabled (the BRGEN bit in the BRGCON register is '0'). This avoids the loading of an interim value to the baud rate generator. **(CAUTION: If either BRGR0 or BRGR1 is written when BRGEN = 1, the result is unpredictable.)**

Table 8-2: Baud Rate Generation for UART

SCON.7 (SM0)	SCON.6 (SM1)	PCON.7 (SMOD1)	BRGCON.1 (SBRGS)	Receive/Transmit Baud Rate for UART
0	0	X	X	CCLK/16
0	1	0	0	CCLK/(256-TH1)64
		1	0	CCLK/(256-TH1)32
		X	1	CCLK/((BRGR1,BRGR0)+16)

SCON.7 (SM0)	SCON.6 (SM1)	PCON.7 (SMOD1)	BRGCON.1 (SBRGS)	Receive/Transmit Baud Rate for UART
1	0	0	X	CCLK/32
		1	X	CCLK/16
1	1	0	0	CCLK/(256-TH1)64
		1	0	CCLK/(256-TH1)32
		X	1	CCLK/((BRGR1,BRGR0)+16)

BRGCON

Address: BDh

Not bit addressable

Reset Source(s): Any reset

Reset Value: xxxxxx00B

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SBRGS	BRGEN

BIT	SYMBOL	FUNCTION
BRGCON.7-2	-	Reserved for future use. Should not be set to 1 by user programs.
BRGCON.1	SBRGS	Select Baud Rate Generator as the source for baud rates to UART in modes 1 & 3 (see Table for details)
BRGCON.0	BRGEN	Baud Rate Generator Enable. Enables the baud rate generator. BRGR1 and BRGR0 can only be written when BRGEN =0.

Figure 8-1: BRGCON Register

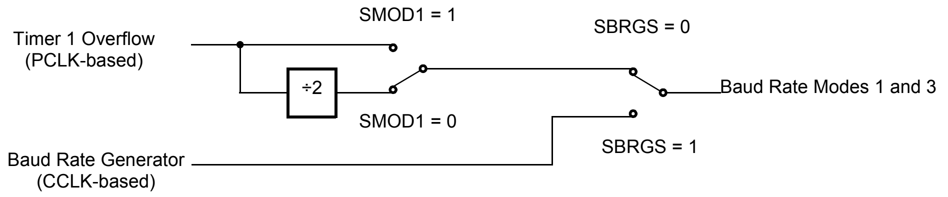


Figure 8-2: Baud Rate Generations for UART (Modes 1, 3)

FRAMING ERROR

A Framing error occurs when the stop bit is sensed as a logic ’0’. A Framing error is reported in the status register (SSTAT). In addition, if SMOD0 (PCON.6) is 1, framing errors can be made available in SCON.7. If SMOD0 is 0, SCON.7 is SM0. It is recommended that SM0 and SM1 (SCON.7-6) are programmed when SMOD0 is ’0’.

BREAK DETECT

A break is detected when any 11 consecutive bits are sensed low. A break detect is reported in the status register (SSTAT). Since a break condition also satisfies the requirements for a framing error, a break condition will also result in reporting a framing error. Once a break condition has been detected, the UART will go into an idle state and remain in this idle state until a stop bit has been received. The break detect can be used to reset the device by setting the EBRR bit (AUXR1.6).

A break detect reset will force the high byte of the program counter to be equal to the Boot Vector contents and the low byte cleared to 00h. The first instruction will be fetched from this address.

.

SCON

Address: 98h

Bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

BIT	SYMBOL	FUNCTION
SCON.7	SM0/FE	The use of this bit is determined by SMOD0 in the PCON register. If SMOD0 = 0, this bit is read and written as SM0, which with SM1, defines the serial port mode. If SMOD0 = 1, this bit is read and written as FE (Framing Error). FE is set by the receiver when an invalid stop bit is detected. Once set, this bit cannot be cleared by valid frames but is cleared by software. (Note: UART mode bits SM0 and SM1 should be programmed when SMOD0 is '0' - default mode on any reset.)
SCON. 6	SM1	With SM0, defines the serial port mode (see table below).

SM0, SM1	UART Mode	UART 0 Baud Rate
0 0	0: shift register	CCLK/16 (default mode on any reset)
0 1	1: 8-bit UART	Variable (see Table)
1 0	2: 9-bit UART	CCLK/32 or CCLK/16
1 1	3: 9-bit UART	Variable (see Table)

BIT	SYMBOL	FUNCTION
SCON.5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then Rl will not be activated if the received 9th data bit (RB8) is 0. In Mode 0, SM2 should be 0. In Mode 1, SM2 must be 0.
SCON.4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
SCON.3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
SCON.2	RB8	The 9th data bit that was received in Modes 2 and 3. In Mode 1 (SM2 must be 0), RB8 is the stop bit that was received. In Mode 0, RB8 is undefined.
SCON.1	TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the the stop bit (see description of INTLO bit in SSTAT register) in the other modes. Must be cleared by software.
SCON.0	RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or approximately halfway through the stop bit time in Mode 1. For Mode 2 or Mode 3, if SMOD0, it is set near the middle of the 9th data bit (bit 8). If SMOD0 = 1, it is set near the middle of the stop bit (see SM2 - SCON.5 - for exceptions). Must be cleared by software.

Figure 8-3: Serial Port Control Register (SCON)

SSTAT

Address: BAh

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

7	6	5	4	3	2	1	0
DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT

BIT	SYMBOL	FUNCTION
SSTAT.7	DBMOD	Double buffering mode. When set = 1 enables double buffering. Must be '0' for UART mode 0. In order to be compatible with existing 80C51 devices, this bit is reset to '0' to disable double buffering.
SSTAT.6	INTLO	Transmit interrupt position. When cleared = 0, the Tx interrupt is issued at the beginning of the stop bit. When set =1, the Tx interrupt is issued at end of the stop bit. Must be '0' for mode 0. Note that in the case of single buffering, if the Tx interrupt occurs at the end of a STOP bit, a gap may exist before the next start bit.
SSTAT.5	CIDIS	Combined Interrupt Disable. When set = 1, Rx and Tx interrupts are separate. When cleared = 0, the UART uses a combined Tx/Rx interrupt (like a conventional 80C51 UART). This bit is reset to '0' to select combined interrupts.
SSTAT.4	DBISEL	Double buffering transmit interrupt select. Used only if double buffering is enabled.This bit controls the number of interrupts that can occur when double buffering is enabled. When set, one transmit interrupt is generated after each character written to SBUF, and there is also one more transmit interrupt generated at the beginning (INTLO = 0) or the end (INTLO = 1) of the STOP bit of the last character sent (i.e., no more data in buffer). This last interrupt can be used to indicate that all transmit operations are over. When cleared = 0, only one transmit interrupt is generated per character written to SBUF. Must be '0' when double buffering is disabled. Note that except for the first character written (when buffer is empty), the location of the transmit interrupt is determined by INTLO. When the first character is written, the transmit interrupt is generated immediately after SBUF is written.
SSTAT.3	FE	Framing error flag is set when the receiver fails to see a valid STOP bit at the end of the frame. Cleared by software.
SSTAT.2	BR	Break Detect flag. A break is detected when any 11 consecutive bits are sensed low. Cleared by software.
SSTAT.1	OE	Overrun Error flag is set if a new character is received in the receiver buffer while it is still full (before the software has read the previous character from the buffer), i.e., when bit 8 of a new byte is received while RI in SCON is still set. Cleared by software.
SSTAT.0	STINT	Status Interrupt Enable. When set =1, FE, BR, or OE can cause an interrupt. The interrupt used (vector address 0023h) is shared with RI (CIDIS = 1) or the combined TI/RI (CIDIS = 0). When cleared = 0, FE, BR, OE cannot cause an interrupt. (Note: FE, BR, or OE is often accompanied by a RI, which will generate an interrupt regardless of the state of STINT). Note that BR can cause a break detect reset if EBRR (AUXR1.6) is set to '1'.

Figure 8-4: Serial Port Status Register (SSTAT)

MORE ABOUT UART MODE 0

In Mode 0, a write to SBUF will initiate a transmission. At the end of the transmission, TI (SCON.1) is set, which must be cleared in software. Double buffering must be disabled in this mode.

Reception is initiated by clearing RI (SCON.0). Synchronous serial transfer occurs and RI will be set again at the end of the transfer. When RI is cleared, the reception of the next character will begin. Refer to Figure 8-5 for timing.

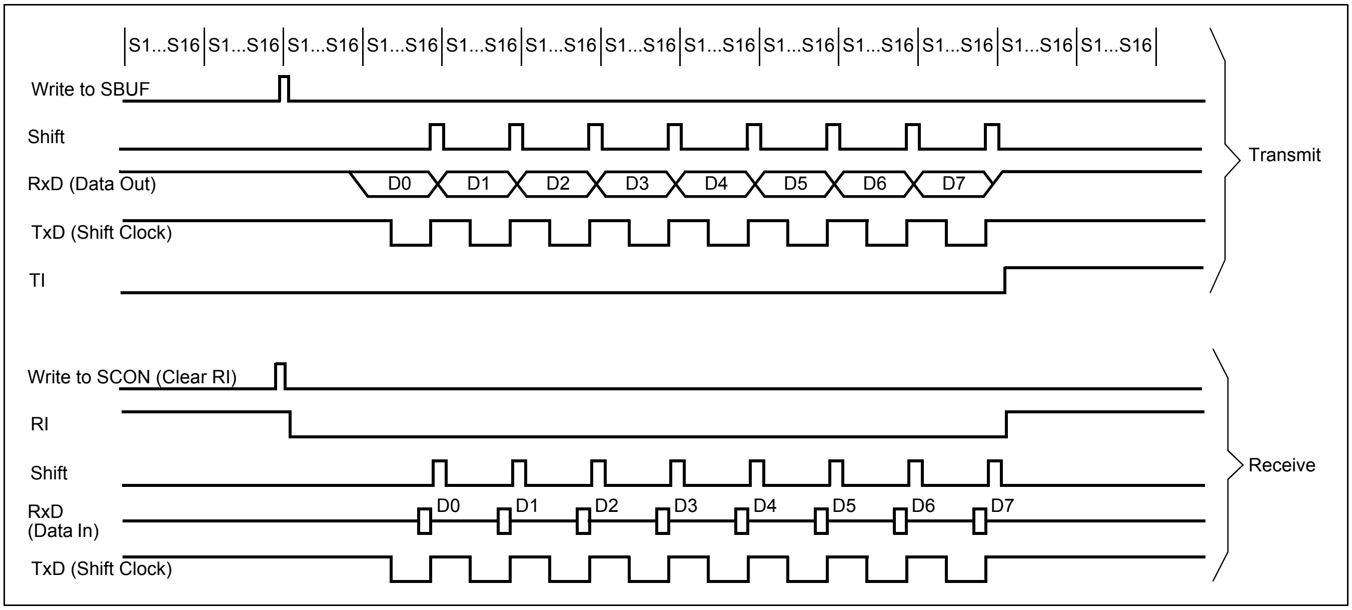


Figure 8-5: Serial Port Mode 0 (Double Buffering Must Be Disabled)

MORE ABOUT UART MODE 1

Reception is initiated by detecting a 1-to-0 transition on RxD. RxD is sampled at a rate 16 times the programmed baud rate. When a transition is detected, the divide-by-16 counter is immediately reset. Each bit time is thus divided into 16 counter states. At the 7th, 8th, and 9th counter states, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the receiver goes back to looking for another 1-to-0 transition. This provides rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated: RI = 0 and either SM2=0 or the received stop bit =1. If either of these two conditions is not met, the received frame is lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated.

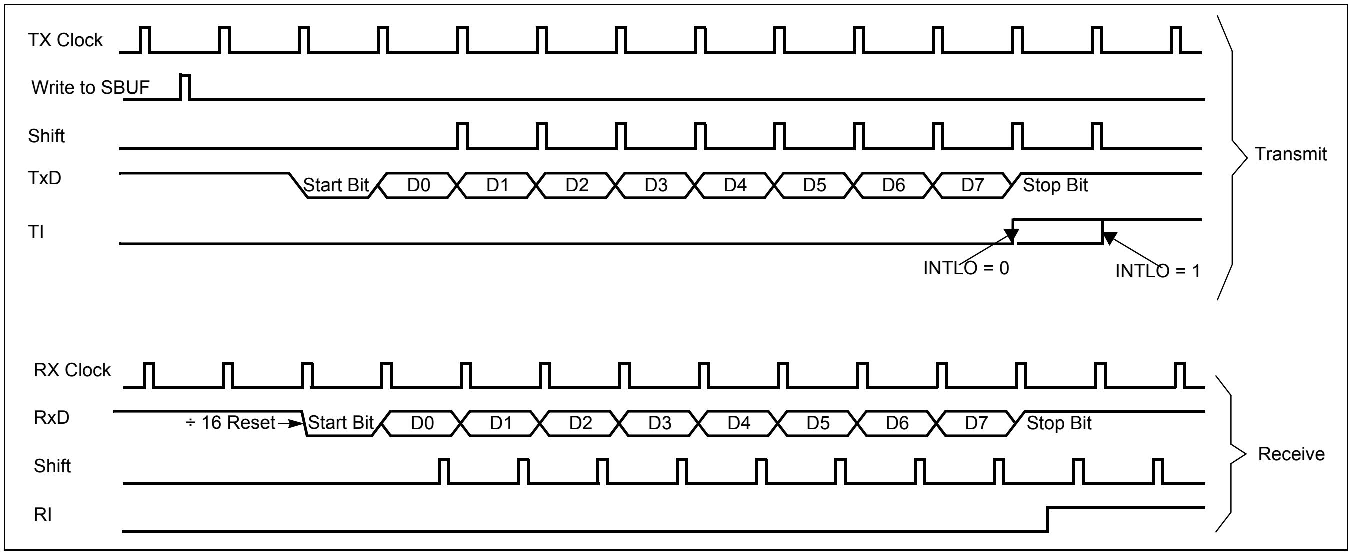


Figure 8-6: Serial Port Mode 1 (Only Single Transmit Buffering Case Is Shown)

MORE ABOUT UART MODES 2 AND 3

Reception is the same as in Mode 1.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated. (a) RI = 0, and (b) Either SM2 = 0, or the received 9th data bit = 1. If either of these conditions is not met, the received frame is lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF.

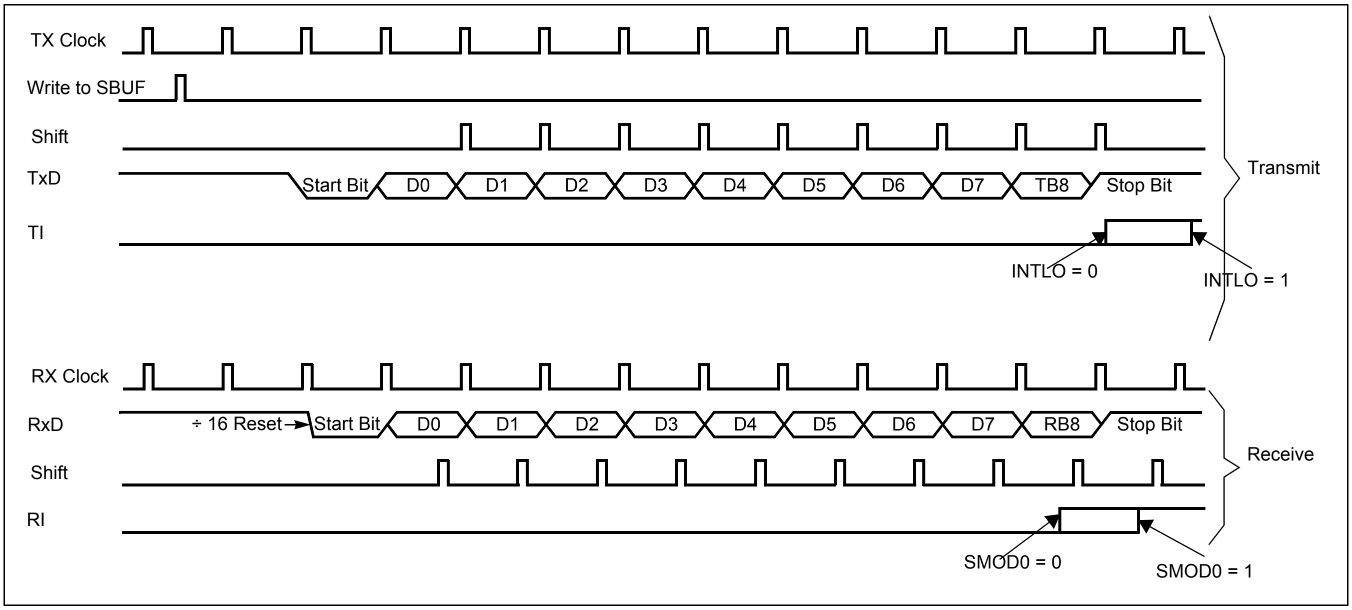


Figure 8-7: Serial Port Mode 2 or 3 (Only Single Transmit Buffering Case Is Shown)

FRAMING ERROR AND RI IN MODES 2 AND 3 WITH SM2 = 1

If SM2 = 1 in modes 2 and 3, RI and FE behave as in the following table.

Mode	PCON.6 (SMOD0)	RB8	RI	FE
2	0	0	No RI when RB8 = 0	Occurs during STOP bit
		1	Similar to Figure 8-7, with SMOD0 = 0, RI occurs during RB8, one bit before FE	Occurs during STOP bit
3	1	0	No RI when RB8 = 0	Will NOT occur
		1	Similar to Figure 8-7, with SMOD0 = 1, RI occurs during STOP bit	Occurs during STOP bit

Table 8-3: FE and RI when SM2 = 1 in Modes 2 and 3.

BREAK DETECT

A break is detected when 11 consecutive bits are sensed low and is reported in the status register (SSTAT). For Mode 1, this consists of the start bit, 8 data bits, and two stop bit times. For Modes 2 & 3, this consists of the start bit, 9 data bits, and one stop bit. The break detect bit is cleared in software or by a reset. The break detect can be used to reset the device. This occurs if the UART is enabled and the the EBRR bit (AUXR1.6) is set and a break occurs.

DOUBLE BUFFERING

The UART has a transmit double buffer that allows buffering of the next character to be written to SBUF while the first character is being transmitted. Double buffering allows transmission of a string of characters with only one stop bit between any two characters, provided the next character is written between the start bit and the stop bit of the previous character.

Double buffering can be disabled. If disabled (DBMOD, i.e. SSTAT.7 = 0), the UART is compatible with the conventional 80C51 UART. If enabled, the UART allows writing to SBUF while the previous data is being shifted out.

DOUBLE BUFFERING IN DIFFERENT MODES

Double buffering is only allowed in Modes 1, 2 and 3. When operated in Mode 0, double buffering must be disabled (DBMOD = 0).

TRANSMIT INTERRUPTS WITH DOUBLE BUFFERING ENABLED (MODES 1, 2 AND 3)

Unlike the conventional UART, when double buffering is enabled, the Tx interrupt is generated when the double buffer is ready to receive new data. The following occurs during a transmission (assuming eight data bits):

1. The double buffer is empty initially.
2. The CPU writes to SBUF.
3. The SBUF data is loaded to the shift register and a Tx interrupt is generated immediately.
4. If there is more data, go to 6, else continue on 5.
5. If there is no more data, then:
   - If DBISEL is ’0’, no more interrupts will occur.
   - If DBISEL is ’1’ and INTLO is ’0’, a Tx interrupt will occur at the beginning of the STOP bit of the data currently in the shifter (which is also the last data).
   - If DBISEL is ’1’ and INTLO is ’1’, a Tx interrupt will occur at the end of the STOP bit of the data currently in the shifter (which is also the last data).
6. If there is more data, the CPU writes to SBUF again. Then:
   - If INTLO is ’0’, the new data will be loaded and a Tx interrupt will occur at the beginning of the STOP bit of the data currently in the shifter.
   - If INTLO is ’1’, the new data will be loaded and a Tx interrupt will occur at the end of the STOP bit of the data currently in the shifter.

   Go to 3.

Note that if DBISEL is ’1’ and the CPU is writing to SBUF when the STOP bit of the last data is shifted out, there can be an uncertainty of whether a Tx interrupt is generated already with the UART not knowing whether there is any more data following.

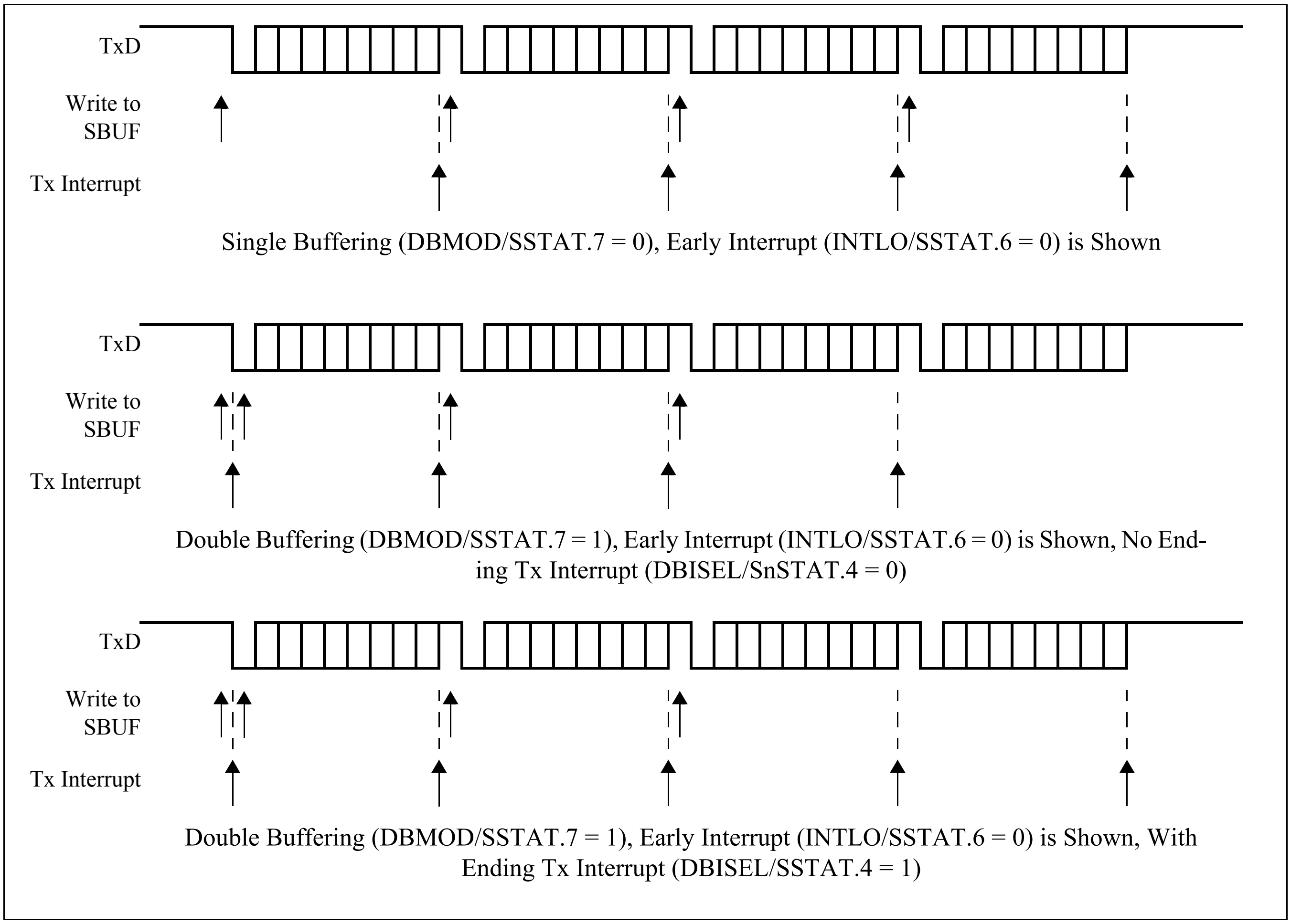


Figure 8-8: Transmission with and without Double Buffering

THE 9TH BIT (BIT 8) IN DOUBLE BUFFERING (MODES 1, 2 AND 3)

If double buffering is disabled (DBMOD, i.e. SSTAT.7 = 0), TB8 can be written before or after SBUF is written, provided TB8 is updated before that TB8 is shifted out. TB8 must not be changed again until after TB8 shifting has been completed, as indicated by the Tx interrupt.

If double buffering is enabled, TB8 MUST be updated before SBUF is written, as TB8 will be double-buffered together with SBUF data. The operation described in the section "Transmit Interrupts with Double Buffering Enabled (Modes 1, 2 and 3)" becomes as follows:

1. The double buffer is empty initially.
2. The CPU writes to TB8.
3. The CPU writes to SBUF.
4. The SBUF/TB8 data is loaded to the shift register and a Tx interrupt is generated immediately.
5. If there is more data, go to 7, else continue on 6.
6. If there is no more data, then:
   - If DBISEL is ’0’, no more interrupt will occur.
   - If DBISEL is ’1’ and INTLO is ’0’, a Tx interrupt will occur at the beginning of the STOP bit of the data currently in the shifter (which is also the last data).

- If DBISEL is '1' and INTLO is '1', a Tx interrupt will occur at the end of the STOP bit of the data currently in the shifter (which is also the last data).

7. If there is more data, the CPU writes to TB8 again.
8. The CPU writes to SBUF again. Then:
   - If INTLO is '0', the new data will be loaded and a Tx interrupt will occur at the beginning of the STOP bit of the data currently in the shifter.
   - If INTLO is '1', the new data will be loaded and a Tx interrupt will occur at the end of the STOP bit of the data currently in the shifter.

   Go to 4.

   Note that if DBISEL is '1' and the CPU is writing to SBUF when the STOP bit of the last data is shifted out, there can be an uncertainty of whether a Tx interrupt is generated already with the UART not knowing whether there is any more data following.

MULTIPROCESSOR COMMUNICATIONS

UART modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received or transmitted. When data is received, the 9th bit is stored in RB8. The UART can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. One way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that follow. The slaves that weren't being addressed leave their SM2 bits set and go on about their business, ignoring the subsequent data bytes.

Note that SM2 has no effect in Mode 0, and must be '0' in Mode 1.

AUTOMATIC ADDRESS RECOGNITION

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes (mode 2 and mode 3), the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0 SADDR = 1100 0000
SADEN = 1111 1101
Given = 1100 00X0

Slave 1 SADDR = 1100 0000
SADEN = 1111 1110
Given = 1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0 SADDR = 1100 0000
SADEN = 1111 1001
Given = 1100 0XX0

Slave 1 SADDR = 1110 0000
SADEN = 1111 1010
Given = 1110 0X0X

Slave 2 SADDR = 1110 0000
SADEN = 1111 1100
Given = 1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2. The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal. Upon reset SADDR and SADEN are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard UART drivers which do not make use of this feature.

9. SERIAL PERIPHERAL INTERFACE (SPI)

The **P89LPC912/913/914** provides another high-speed serial communication interface, the SPI interface. SPI is a full-duplex, high-speed, synchronous communication bus with two operation modes: Master mode and Slave mode. Up to 3 Mbit/s can be supported in either Master or Slave mode. It has a Transfer Completion Flag and Write Collision Flag Protection.

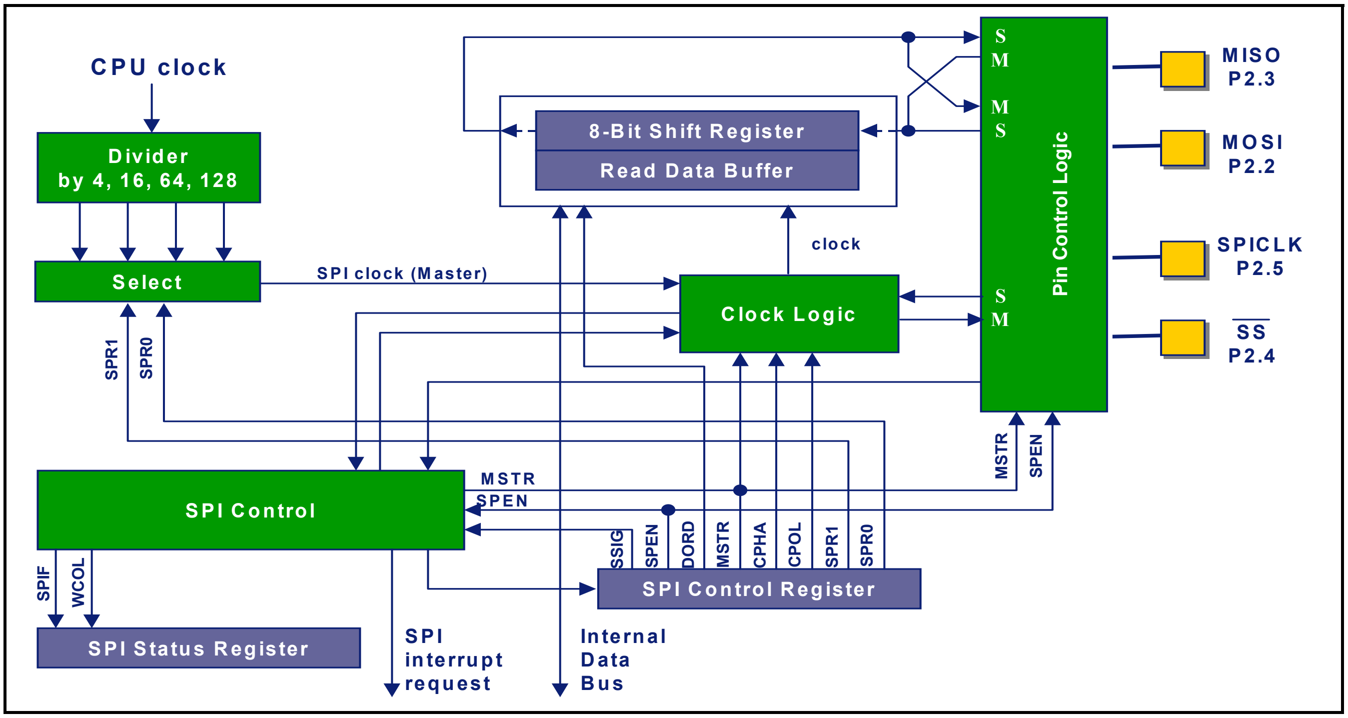


Figure 9-1: SPI block diagram

The SPI interface has four pins: SPICLK, MOSI, MISO and SS:

- SPICLK, MOSI and MISO are typically tied together between two or more SPI devices. Data flows from master to slave on the MOSI (Master Out Slave In) pin and flows from slave to master on the MISO (Master In Slave Out) pin. The SPICLK signal is output in the master mode and is input in the slave mode. If the SPI system is disabled, i.e. SPEN (SPCTL.6) = 0 (reset value), these pins are configured for port functions.
- SS is the optional slave select pin. In a typical configuration, an SPI master asserts one of its port pins to select one SPI device as the current slave. An SPI slave device uses its SS pin to determine whether it is selected. The SS is ignored if any of the following conditions are true:
  - If the SPI system is disabled, i.e. SPEN (SPCTL.6) = 0 (reset value)
  - If the SPI is configured as a master, i.e., MSTR (SPCTL.4) = 1, and P2.4 is configured as an output (via the P2M1.4 and P2M2.4 SFR bits);
  - If the SS pin is ignored, i.e. SSIG (SPCTL.7) bit = 1, this pin is configured for port functions.

  Note that even if the SPI is configured as a master (MSTR = 1), it can still be converted to a slave by driving the SS pin low (if P2.4 is configured as input and SSIG = 0). Should this happen, the SPIF bit (SPSTAT.7) will be set (see section "Mode change on SS").

Typical connections are shown in Figures 9-5 - 9-7.

- The **89LPC913** does not have the slave select pin, SS. The SPI interface is set to Master mode and an I/O pin may be used to implement the SS function. Typical connections are shown in Figure 9-5 and Figure 9-7.

SPCTL

Address: E2h

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000100B

7	6	5	4	3	2	1	0
SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

BIT	SYMBOL	FUNCTION
SPCTL.7	SSIG	$\overline{SS}$ IGnore. If set = 1, MSTR (bit 4) decides whether the device is a master or slave. If cleared = 0, the $\overline{SS}$ pin decides whether the device is master or slave. The $\overline{SS}$ pin can be used as a port pin (see Table 9-1).
SPCTL.6	SPEN	SPI Enable. If set = 1, the SPI is enabled. If cleared = 0, the SPI is disabled and all SPI pins will be port pins.
SPCTL.5	DORD	SPI Data ORDer. 1: The LSB of the data word is transmitted first. 0: The MSB of the data word is transmitted first.
SPCTL.4	MSTR	Master/Slave mode Select (see Table 9-1).
SPCTL.3	CPOL	SPI Clock POLarity (see Figures 9-8 - 9-11): 1: SPICLK is high when idle. The leading edge of SPICLK is the falling edge and the trailing edge is the rising edge. 0: SPICLK is low when idle. The leading edge of SPICLK is the rising edge and the trailing edge is the falling edge.
SPCTL.2	CPHA	SPI Clock PHAse select (see Figures 9-8 - 9-11): 1: Data is driven on the leading edge of SPICLK, and is sampled on the trailing edge. 0: Data is driven when SS is low (SSIG = 0) and changes on the trailing edge of SPICLK, and is sampled on the leading edge. (Note: If SSIG = 1, the operation is not defined.)
SPCTL.1-0	SPR1-SPR0	SPI Clock Rate Select:
	SPR1-SPR0	SPI Clock Rate
	0 0	CCLK/4
	0 1	CCLK/16
	1 0	CCLK/64
	1 1	CCLK/128

Figure 9-2: SPI Control register

SPSTAT

Address: E1h

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00xxxxxxB

7	6	5	4	3	2	1	0
SPIF	WCOL	-	-	-	-	-	-

BIT	SYMBOL	FUNCTION
SPSTAT.7	SPIF	SPI Transfer Completion Flag. When a serial transfer finishes, the SPIF bit is set and an interrupt is generated if both the ESPI (IEN1.3) bit and the EA bit are set. If $\overline{\text{SS}}$ is an input and is driven low when SPI is in master mode, and SSIG = 0, this bit will also be set (see section "Mode change on SS"). The SPIF flag is cleared in software by writing '1' to this bit.
SPSTAT.6	WCOL	SPI Write Collision Flag. The WCOL bit is set if the SPI data register, SPDAT, is written during a data transfer (see section "Write collision"). The WCOL flag is cleared in software by writing '1' to this bit.
SPSTAT.5-0	-	Reserved for future use. Should not be set to 1 by user programs.

Figure 9-3: SPI Status register definition

SPDAT

Address: E3h

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

7	6	5	4	3	2	1	0
MSB							LSB

BIT	SYMBOL	FUNCTION
SPD.7-0	-	Bit 7-0 of data transferred.

Figure 9-4: SPI Data register

TYPICAL SPI CONFIGURATIONS

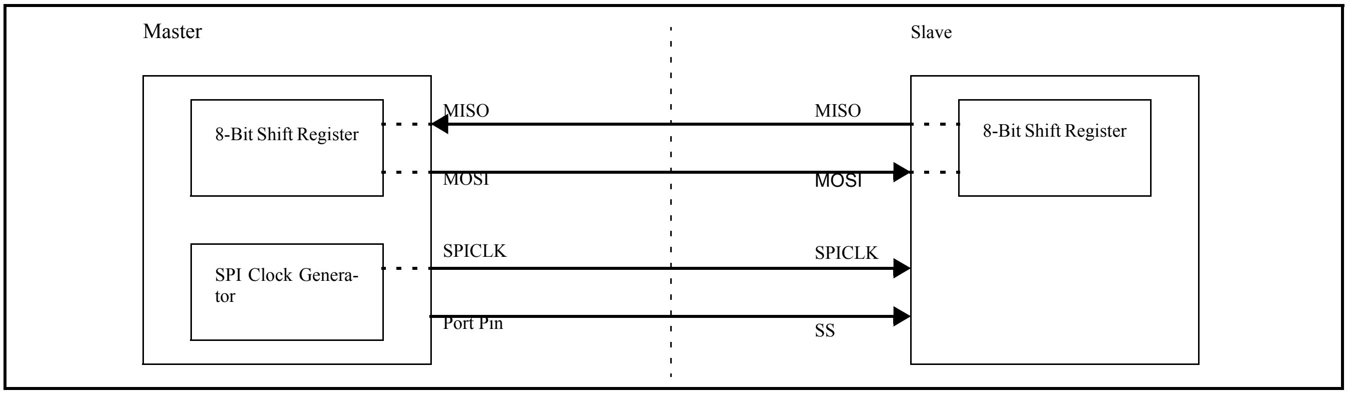


Figure 9-5: SPI single master single slave configuration

In Figure 9-5, SSIG (SPCTL.7) for the slave is '0', and $\overline{\text{SS}}$ is used to select the slave. The SPI master can use any port pin (including P2.4/$\overline{\text{SS}}$) to drive the $\overline{\text{SS}}$ pin.

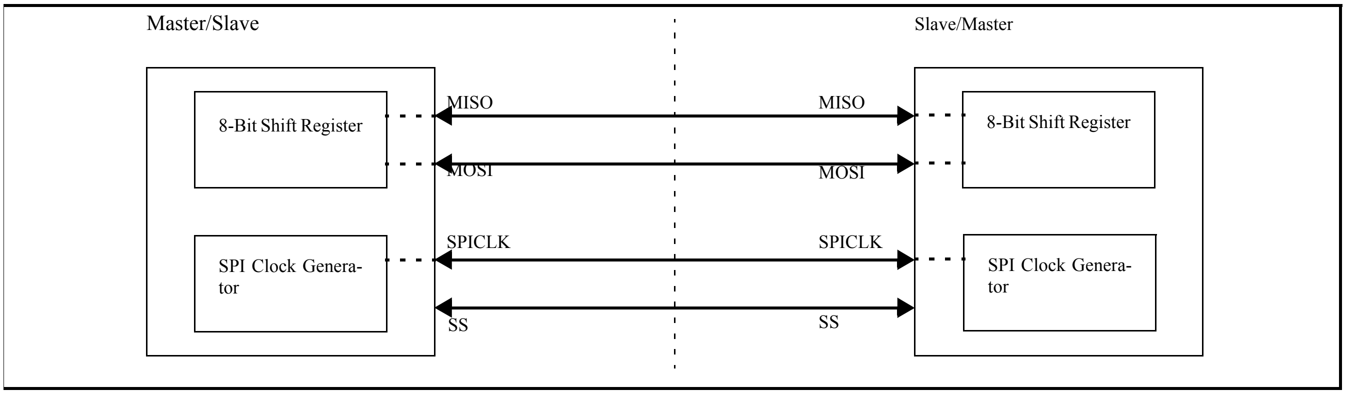


Figure 9-6: SPI dual device configuration, where either can be a master or a slave.

Figure 9-6 shows a case where two devices are connected to each other and either device can be a master or a slave. When no SPI operation is occuring, both can be configured as masters (MSTR = 1) with SSIG cleared to 0 and P2.4 ($\overline{SS}$) configured in quasi-bidirectional mode. When a device initiates a transfer, it can configure P2.4 as an output and drive it low, forcing a mode change in the other device (see section "Mode change on SS") to slave.

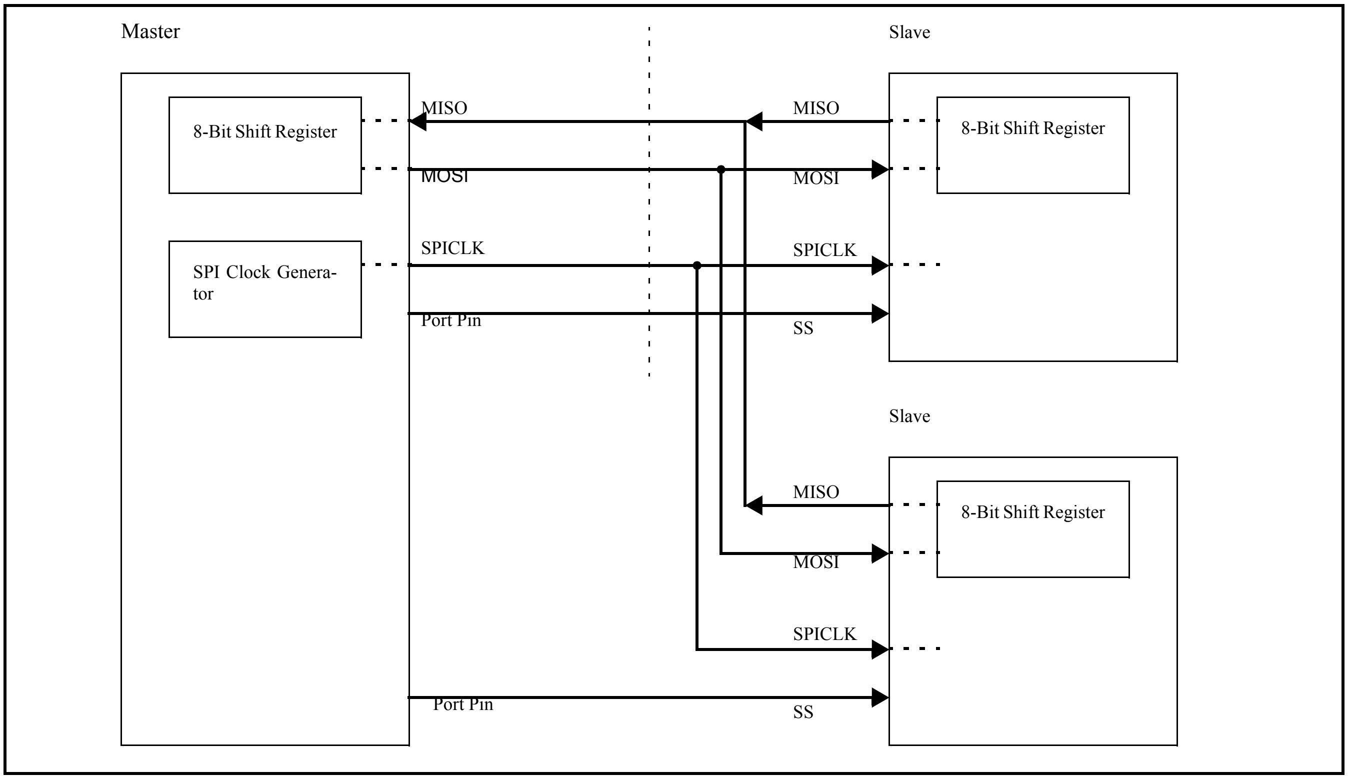


Figure 9-7: SPI single master multiple slaves configuration

In Figure 9-7, SSIG (SPCTL.7) bits for the slaves are '0', and the slaves are selected by the corresponding $\overline{SS}$ signals. The SPI master can use any port pin (including P2.4/$\overline{SS}$) to drive the $\overline{SS}$ pins.

CONFIGURING THE SPI

Table 9-1 shows configuration for the master/slave modes as well as usages and directions for the modes.

Table 9-1: SPI master and slave selection

SPEN (SPCTL. 6)	SSIG (SPCTL. 7)	$\overline{SS}$ Pin	MSTR (SPCTL. 4)	Master or Slave Mode	MISO	MOSI	SPICLK	Remarks
0	X	P2.4^1	X	SPI Disabled	P2.3^1	P2.2^1	P2.5^1	SPI disabled. P2.2, P2.3, P2.4, P2.5 are used as port pins.
1	0	0	0	Slave	output	input	input	Selected as slave.
1	0	1	0	Slave	Hi-Z	input	input	Not selected. MISO is high impedance to avoid bus contention.
1	0	0	1 (-> 0)^2	Slave	output	input	input	P2.4/$\overline{SS}$ is configured as an input or quasi-bidirectional pin. SSIG is 0. Selected externally as slave if $\overline{SS}$ is selected and is driven low. The MSTR bit will be cleared to '0' when $\overline{SS}$ becomes low.
1	0	1	1	Master (idle)	input	Hi-Z	Hi-Z	MOSI and SPICLK are at high impedance to avoid bus contention when the Master is idle. The application must pull-up or pull-down SPICLK (depending on CPOL - SPCTL.3) to avoid a floating SPICLK.
				Master (active)		output	output	MOSI and SPICLK are push-pull when the Master is active.
1	1	P2.4^1	0	Slave	output	input	input	
1	1	P2.4^1	1	Master	input	output	output	

1. Selected as a port function.
2. The MSTR bit changes to '0' automatically when $\overline{SS}$ becomes low in input mode and SSIG is 0.

ADDITIONAL CONSIDERATIONS FOR A SLAVE

When CPHA equals zero, SSIG must be '0' and the $\overline{SS}$ pin must be negated and reasserted between each successive serial byte. If the SPDAT register is written while $\overline{SS}$ is active (low), a write collision error results. The operation is undefined if CPHA is '0' and SSIG is '1'.

When CPHA equals one, SSIG may be set to '1'. If SSIG = 0, the $\overline{SS}$ pin may remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave driving the MISO data line.

ADDITIONAL CONSIDERATIONS FOR A MASTER

In SPI, transfers are always initiated by the master. If the SPI is enabled (SPEN = 1) and selected as master, writing to the SPI data register by the master starts the SPI clock generator and data transfer. The data will start to appear on MOSI about one half SPI bit-time to one SPI bit-time after data is written to SPDAT.

Note that the master can select a slave by driving the $\overline{\text{SS}}$ pin of the corresponding device low. Data written to the SPDAT register of the master is shifted out of the MOSI pin of the master to the MOSI pin of the slave, at the same time the data in SPDAT register in slave side is shifted out on MISO pin to the MISO pin of the master.

After shifting one byte, the SPI clock generator stops, setting the transfer completion flag (SPIF) and an interrupt will be created if the SPI interrupt is enabled (ESPI, or IEN1.3 = 1). The two shift registers in the master CPU and slave CPU can be considered as one distributed 16-bit circular shift register. When data is shifted from the master to the slave, data is also shifted in the opposite direction simultaneously. This means that during one shift cycle, data in the master and the slave are interchanged.

MODE CHANGE ON $\overline{\text{SS}}$

If SPEN = 1, SSIG = 0 and MSTR = 1, the SPI is enabled in master mode. The $\overline{\text{SS}}$ pin can be configured as an input (P2M2.4, P2M1.4 = 00) or quasi-bidirectional (P2M2.4, P2M1.4 = 01). In this case, another master can drive this pin low to select this device as an SPI slave and start sending data to it. To avoid bus contention, the SPI becomes a slave. As a result of the SPI becoming a slave, the MOSI and SPICLK pins are forced to be an input and MISO becomes an output.

The SPIF flag in SPSTAT is set, and if the SPI interrupt is enabled, an SPI interrupt will occur.

User software should always check the MSTR bit. If this bit is cleared by a slave select and the user wants to continue to use the SPI as a master, the user must set the MSTR bit again, otherwise it will stay in slave mode.

WRITE COLLISION

The SPI is single buffered in the transmit direction and double buffered in the receive direction. New data for transmission can not be written to the shift register until the previous transaction is complete. The WCOL (SPSTAT.6) bit is set to indicate data collision when the data register is written during transmission. In this case, the data currently being transmitted will continue to be transmitted, but the new data, i.e., the one causing the collision, will be lost.

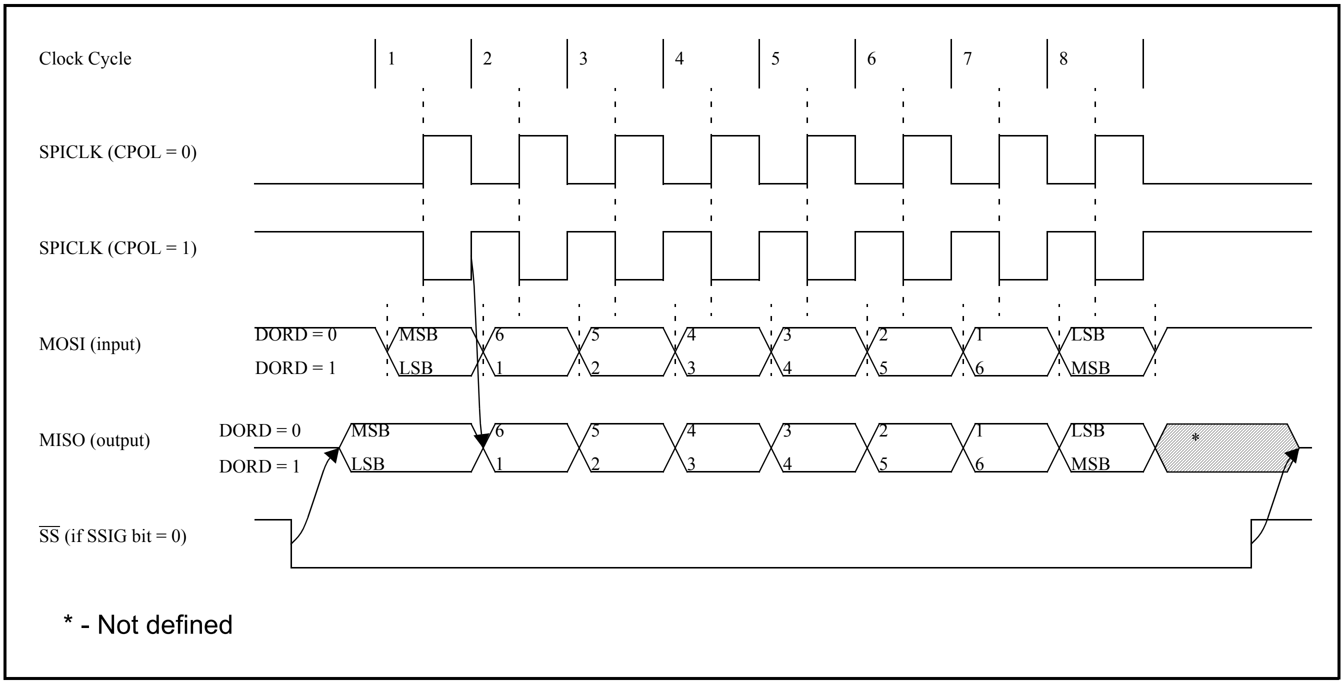
While write collision is detected for both a master or a slave, it is uncommon for a master because the master has full control of the transfer in progress. The slave, however, has no control over when the master will initiate a transfer and therefore collision can occur.

For receiving data, received data is transferred into a parallel read data buffer so that the shift register is free to accept a second character. However, the received character must be read from the Data Register before the next character has been completely shifted in. Otherwise. the previous data is lost.

WCOL can be cleared in software by writing ‘1’ to the bit.

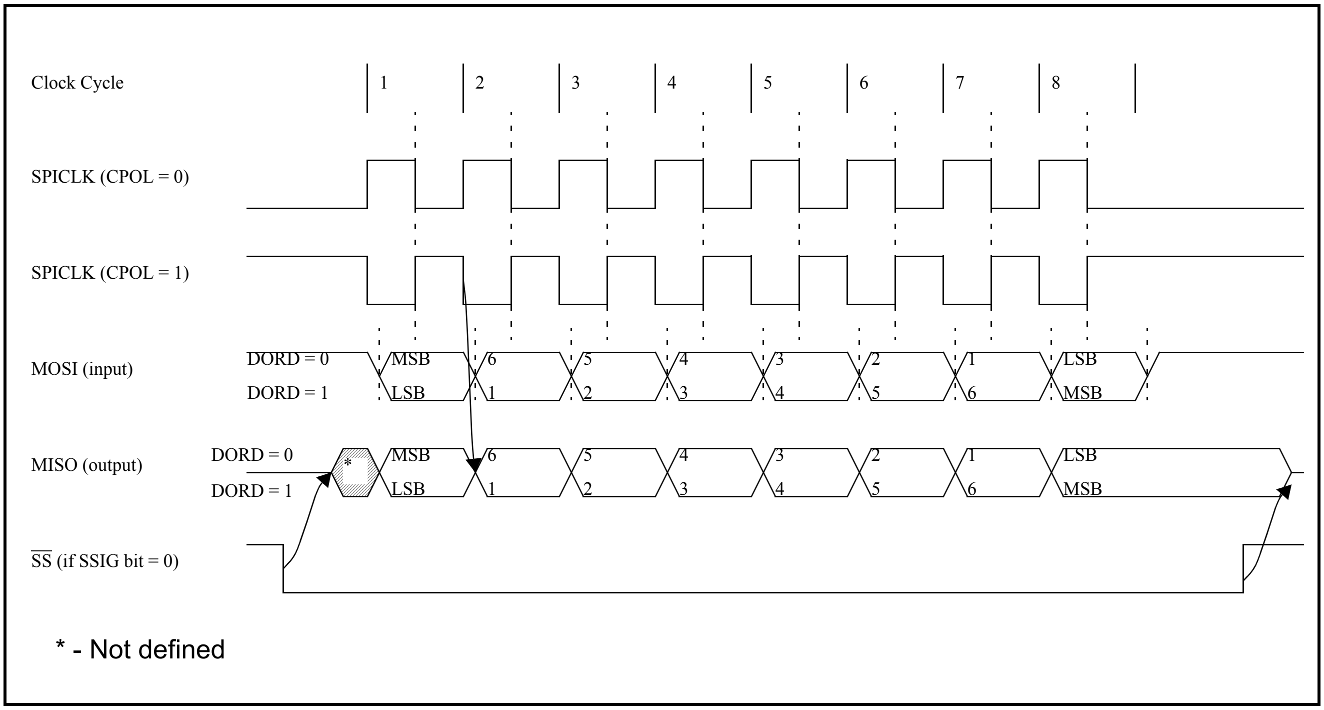
DATA MODE

Clock Phase Bit (CPHA) allows the user to set the edges for sampling and changing data. The Clock Polarity bi, CPOL, allows the user to set the clock polarity. Figures 9-8 - 9-11 show the different settings of Clock Phase bit CPHA.



* - Not defined

Figure 9-8: SPI slave transfer format with CPHA = 0



* - Not defined

Figure 9-9: SPI slave transfer format with CPHA = 1

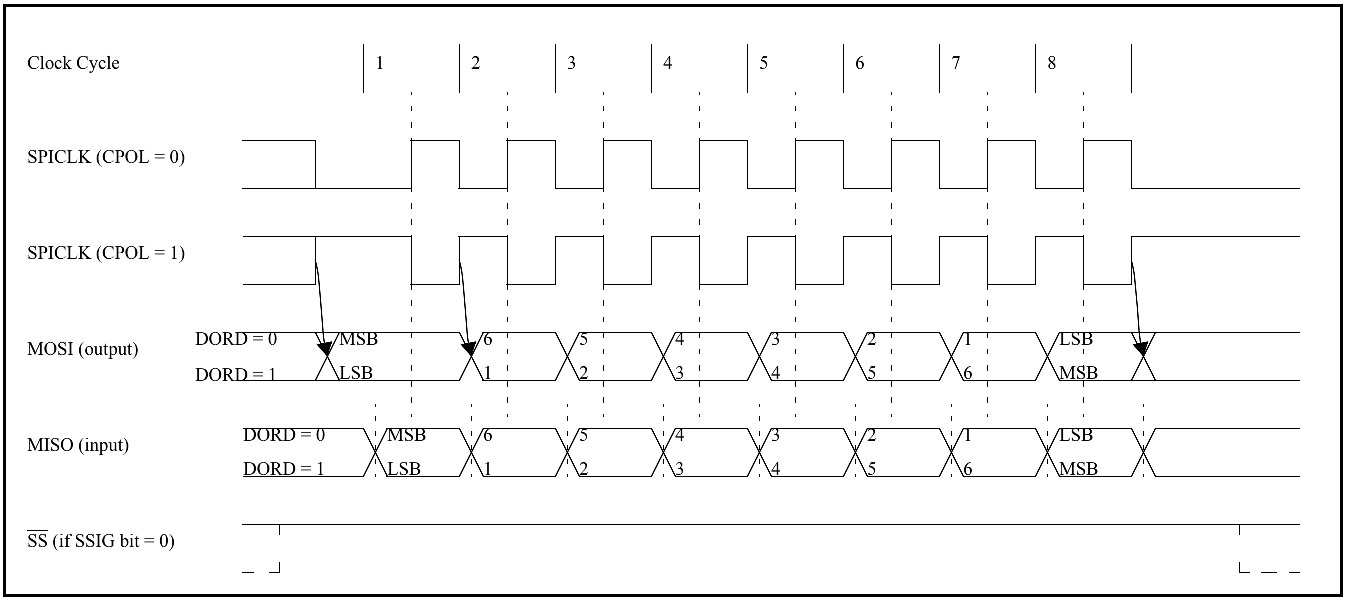


Figure 9-10: SPI master transfer format with CPHA = 0

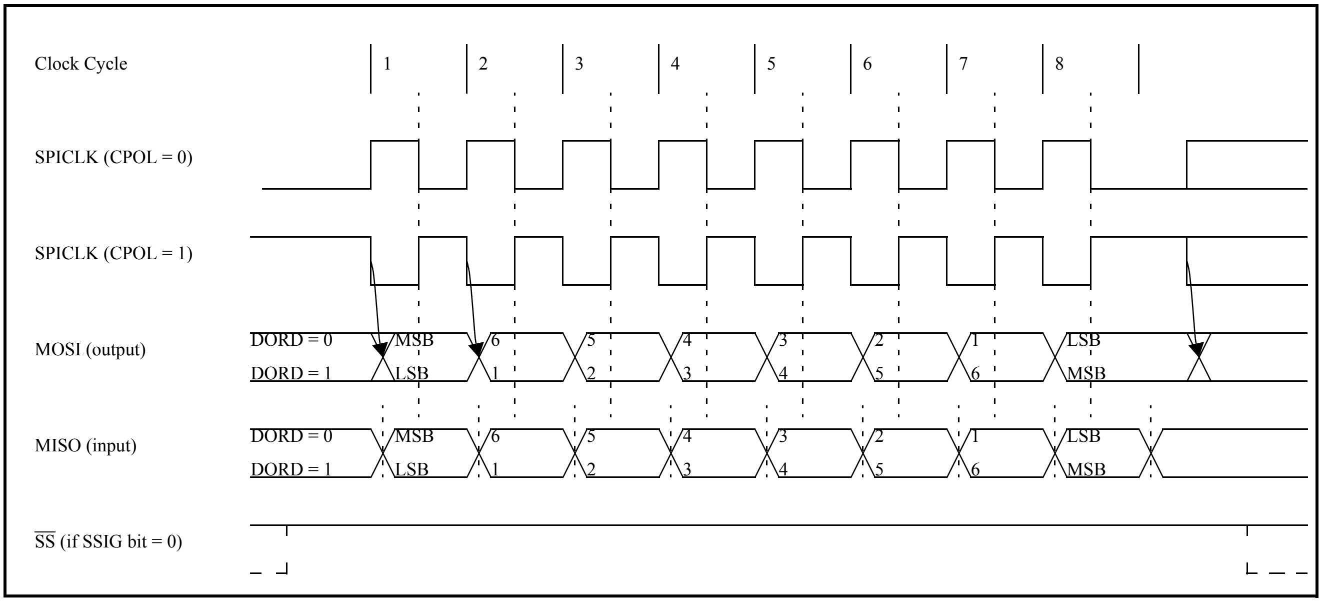


Figure 9-11: SPI master transfer format with CPHA = 1

SPI CLOCK PRESCALER SELECT

The SPI clock prescalar selection uses the SPR1-SPR0 bits in the SPCTL register (see Figure 9-2).

10. RESET

The P1.5/$\overline{\text{RST}}$ pin can function as either an active low reset input or as a digital input, P1.5. The RPE (Reset Pin Enable) bit in UCFG1, when set to 1, enables the external reset input function on P1.5. When cleared, P1.5 may be used as an input pin.

NOTE: During a power-on sequence, The RPE selection is overriden and this pin will always functions as a reset input. An external circuit connected to this pin should not hold this pin low during a Power-on sequence as this will keep the device in reset. After power-on this input will function either as an external reset input or as a digital input as defined by the RPE bit. Only a power-on reset will temporarily override the selection defined by RPE bit. Other sources of reset will not override the RPE bit.

NOTE: During a power cycle, V_{DD} must fall below V_{POR} (see "DC electrical characteristics" in the datasheet) before pwoer is reapplied, in order to ensure a power-on reset.

Reset can be triggered from the following sources (see Figure 10-1):

- External reset pin (during power-on or if user configured via UCFG1);
- Power-on Detect;
- Brownout Detect;
- Watchdog Timer;
- Software reset;
- UART break-character detect reset. (**P89LPC913/914**)

For every reset source, there is a flag in the Reset Register, RSTSRC. The user can read this register to determine the most recent reset source. These flag bits can be cleared in software by writing a '0' to the corresponding bit. More than one flag bit may be set:

- During a power-on reset, both POF and BOF are set but the other flag bits are cleared.
- For any other reset, any previously set flag bits that have not been cleared will remain set.

POWER-ON RESET CODE EXECUTION

The **P89LPC912/913/914** contains two special Flash elements: the BOOT VECTOR and the Boot Status Bit. Following reset, the device 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 one, the contents of the Boot Vector is used as the high byte of the execution address and the low byte is set to 00H. The factory default setting is 00H. A UART break-detect reset (**P89LPC913/914**) will have the same effect as a non-zero Status Bit.

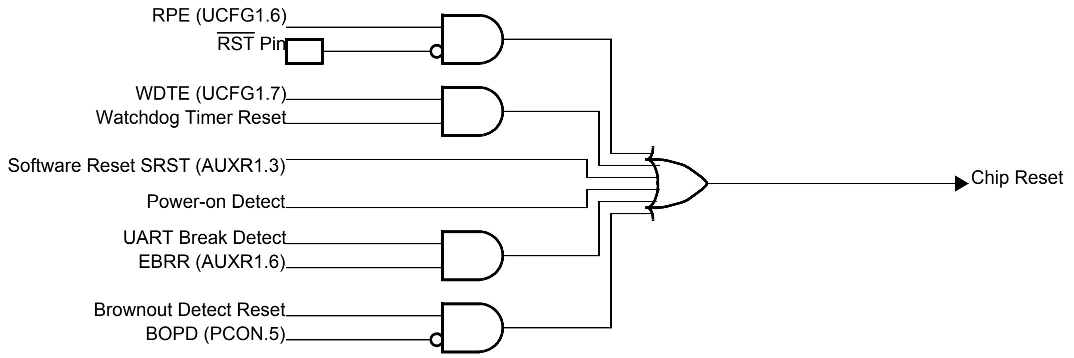


Figure 10-1: Block Diagram of Reset

RSTSRC

Address: DFH

Not bit addressable

7	6	5	4	3	2	1	0
-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX

Reset Sources: Power-on only

Reset Value: xx110000B (This is the power-on reset value. Other reset sources will set corresponding bits.)

BIT	**SYMBOL**	**FUNCTION**
RSTSRC.7-6	-	Reserved for future use. Should not be set to 1 by user programs.
RSTSRC.5	BOF	Brownout Detect Flag. When Brownout Detect is activated, this bit is set. It will remain set until cleared by software by writing a '0' to the bit. (Note: On a Power-on reset, both POF and this bit will be set while the other flag bits are cleared.)
RSTSRC.4	POF	Power-on Detect Flag. When Power-on Detect is activated, the POF flag is set to indicate an initial power-up condition. The POF flag will remain set until cleared by software by writing a '0' to the bit.. (Note: On a Power-on reset, both BOF and this bit will be set while the other flag bits are cleared.)
RSTSRC.3	R_BK	Break detect reset. If a break detect occurs and EBRR (AUXR1.6) is set to '1', a system reset will occur. This bit is set to indicate that the system reset is caused by a break detect. Cleared by software by writing a '0' to the bit or on a Power-on reset. (**P89LPC913/914)**
RSTSRC.2	R_WD	Watchdog Timer reset flag. Cleared by software by writing a '0' to the bit or a Power-on reset.(NOTE: UCFG1.7 must be = 1).
RSTSRC.1	R_SF	Software reset Flag. Cleared by software by writing a '0' to the bit or a Power-on reset.
RSTSRC.0	R_EX	External reset Flag. When this bit is '1', it indicates external pin reset. Cleared by software by writing a '0' to the bit or a Power-on reset. If $\overline{RST}$ is still asserted after the Power-on reset is over, R_EX will be set.

Figure 10-2: Reset Sources Register

11. ANALOG COMPARATORS

Two analog comparators are provided on the **P89LPC912/913/914**. Input and output options allow use of the comparators in a number of different configurations. Comparator operation is such that the output is a logical one when the positive input is greater than the negative input (selectable from a pin or an internal reference voltage).Otherwise the output is a zero.Each comparator may be configured to cause an interrupt when the output value changes. The comparator output can also be read in a register. In addition, the Comparator 1 output may be routed to a pin.

Comparator configuration

Each comparator has a control register, CMP1 for comparator 1 and CMP2 for comparator 2 and are shown in Figure 11-1 and Figure 11-2

The overall connections to both comparators are shown in Figure 11-3. There are four possible configurations for Comparator 1 and two for Comparator 2, as determined by the control bits in the corresponding CMPn register. These configurations are shown in Figure 11-4.

When each comparator is first enabled, the comparator output and interrupt flag are not guaranteed to be stable for 10 microseconds. The corresponding comparator interrupt should not be enabled during that time, and the comparator interrupt flag must be cleared before the interrupt is enabled in order to prevent an immediate interrupt service.

CMP1

Address: ACh

Not bit addressable

Reset Source(s): Any reset

Reset Value: xx000000B

7	6	5	4	3	2	1	0
-	-	CE1	-	CN1	OE1	CO1	CMF1

BIT	SYMBOL	FUNCTION
CMPn.7, 6,4	-	Reserved for future use. Should not be set to 1 by user programs.
CMPn.5	CE1	Comparator enable. When set, the comparator function is enabled. Comparator output is stable 10 microseconds after CE1 is set.
CMPn.3	CN1	Comparator negative input select. When 0, the comparator reference pin CMPREF is selected as the negative comparator input. When 1, the internal comparator reference, Vref, is selected as the negative comparator input.
CMPn.2	OE1	Output enable. When 1, the comparator output is connected to the CMP1 pin if the comparator is enabled (CE1 = 1). This output is asynchronous to the CPU clock.
CMPn.1	CO1	Comparator output, synchronized to the CPU clock to allow reading by software.
CMPn.0	CMF1	Comparator interrupt flag. This bit is set by hardware whenever the comparator output CO1 changes state. This bit will cause a hardware interrupt if enabled. Cleared by software.

Figure 11-1: Comparator 1 control registers (CMP1)

CMP2

Address: ADh

Not bit addressable

Reset Source(s): Any reset

Reset Value: xx000000B

7	6	5	4	3	2	1	0
-	-	CE2	-	CN2	-	CO2	CMF2

BIT	SYMBOL	FUNCTION
CMPn.7, 6,4,2	-	Reserved for future use. Should not be set to 1 by user programs.
CMPn.5	CE2	Comparator enable. When set, the comparator function is enabled. Comparator output is stable 10 microseconds after CE2 is set.
CMPn.3	CN2	Comparator negative input select. When 0, the comparator reference pin CMPREF is selected as the negative comparator input. When 1, the internal comparator reference, Vref, is selected as the negative comparator input.
CMPn.1	CO2	Comparator output, synchronized to the CPU clock to allow reading by software.
CMPn.0	CMF2	Comparator interrupt flag. This bit is set by hardware whenever the comparator output CO2 changes state. This bit will cause a hardware interrupt if enabled. Cleared by software.

Figure 11-2: Comparator 2 control register (CMP2)

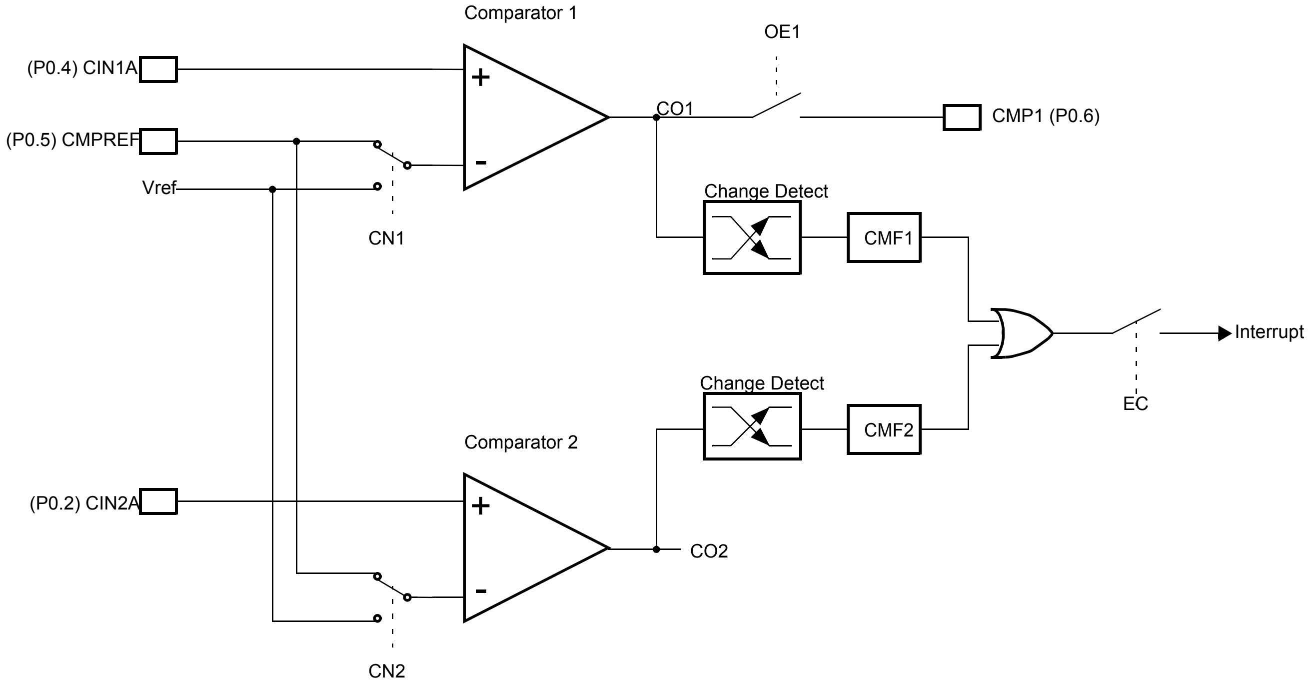


Figure 11-3: Comparator input and output connections

Internal reference voltage

An internal reference voltage, Vref, may supply a default reference when a single comparator input pin is used. Please refer to the Datasheet for specifications.

Comparator interrupt

Each comparator has an interrupt flag CMFn contained in its configuration register. This flag is set whenever the comparator output changes state. The flag may be polled by software or may be used to generate an interrupt. The two comparators use one common interrupt vector. The interrupt will be generated when the interrupt enable bit EC in the IEN1 register is set and the interrupt system is enabled via the EA bit in the IEN0 register. If both comparators enable interrupts, after entering the interrupt service routine, the user will need to read the flags to determine which comparator caused the interrupt.

When a comparator is disabled the comparator's output, COn, goes high. If the comparator output was low and then is disabled, the resulting transition of the comparator output from a low to high state will set the the comparator flag, CMFn. This will cause an interrupt if the comparator interrupt is enabled. The user should therefore disable the comparator interrupt prior to disabling the comparator. Additionally, the user should clear the comparator flag, CMFn, after disabling the comparator.

Comparators and power reduction modes

Either or both comparators may remain enabled when Power down or Idle mode is activated, but both comparators are disabled automatically in Total Power down mode.

If a comparator interrupt is enabled (except in Total Power down mode), a change of the comparator output state will generate an interrupt and wake up the processor. If the comparator output to a pin is enabled, the pin should be configured in the push-pull mode in order to obtain fast switching times while in Power down mode. The reason is that with the **oscillator** stopped, the temporary strong pull-up that normally occurs during switching on a quasi-bidirectional port pin does not take place.

Comparators consume power in Power down and Idle modes, as well as in the normal operating mode. This should be taken into consideration when system power consumption is an issue. To minimize power consumption, the user can Power down the comparators by disabling the comparators and setting PCONA.5 to '1', or simply putting the device in Total Power down mode.

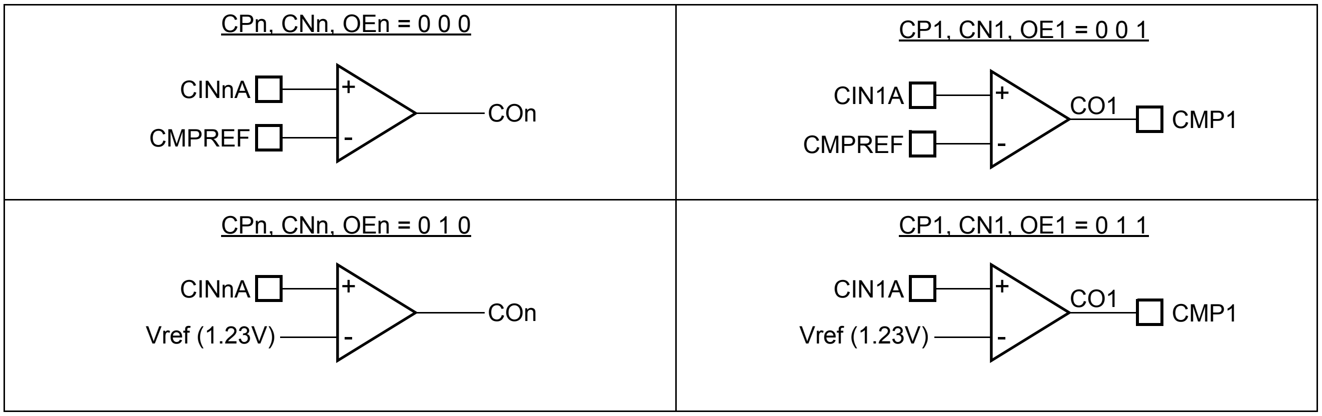


Figure 11-4: Comparator configurations

Comparator configuration example

The code shown below is an example of initializing one comparator. Comparator 1 is configured to use the CIN1A and CMPREF inputs, outputs the comparator result to the CMP1 pin, and generates an interrupt when the comparator output changes.

```
CMPINIT:
        MOV     PT0AD,#030h     ; Disable digital INPUTS on pins that are used for analog functions: CIN1A, CMPREF.
        ANL     P0M2,#0CFh      ; Disable digital OUTPUTS on pins that are used
        ORL     P0M1,#030h      ; for analog functions: CIN1A, CMPREF.
        MOV     CMP1,#024h      ; Turn on comparator 1 and set up for:
                                ; - Positive input on CIN1A.
                                ; - Negative input from CMPREF pin.
                                ; - Output to CMP1 pin enabled.
        CALL    delay10us       ; The comparator has to start up for at least 10 microseconds before use.
        ANL     CMP1,#0FEh      ; Clear comparator 1 interrupt flag.
        SETB    EC              ; Enable the comparator interrupt. The priority is left at the current value.
        SETB    EA              ; Enable the interrupt system (if needed).
        RET                     ; Return to caller.
```

The interrupt routine used for the comparator must clear the interrupt flag (CMF1 in this case) before returning.

12. KEYPAD INTERRUPT (KBI)

The Keypad Interrupt function is intended primarily to allow a single interrupt to be generated when the Port 0 bits are equal to or not equal to a certain pattern. This function can be used for keypad recognition. The user can configure the port via SFRs for different tasks.

There are three SFRs used for this function. The Keypad Interrupt Mask Register (KBMASK) is used to define which input pins connected to Port 0 are enabled to trigger the interrupt. The Keypad Pattern Register (KBPATN) is used to define a pattern that is compared to the value of Port 0. The Keypad Interrupt Flag (KBIF) in the Keypad Interrupt Control Register (KBCON) is set when the condition is matched while the Keypad Interrupt function is active. An interrupt will be generated if it has been enabled by setting the EKBI bit in IEN1 register and EA = 1. The PATN_SEL bit in the Keypad Interrupt Control Register (KBCON) is used to define equal or not-equal for the comparison.

In order to use the Keypad Interrupt as an original KBI function like in the 87LPC76x series, the user needs to set KBPATN = 0FFH and PATN_SEL = 0 (not equal). Any key connected to Port0 which is enabled by KBMASK register will cause the hardware to set KBIF = 1 and generate an interrupt if it has been enabled. The interrupt may be used to wake up the CPU from Idle or Power down modes. This feature is particularly useful in handheld, battery powered systems that need to carefully manage power consumption yet also need to be convenient to use.

In order to set the flag and and cause an interrupt, the pattern on Port 0 must be held longer than 6 CCLKs.

KBPATN

Address: 93h

Not bit addressable

Reset Source(s): Any reset

Reset Value: 11111111B

7	6	5	4	3	2	1	0
-	KBPATN.6	KBPATN.5	KBPATN.4	-	KBPATN.2	-	-

BIT	SYMBOL	FUNCTION
KBPATN.5,2	-	Pattern bit 6:4, bit 2

Figure 12-1: Keypad Pattern Register

KBCON

Address: 94h

Not bit addressable

Reset Source(s): Any reset

Reset Value: xxxxxx00B

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PATN_SEL	KBIF

BIT	SYMBOL	FUNCTION
KBCON.7-2	-	Reserved
KBCON.1	PATN_SEL	Pattern Matching Polarity selection. When set, Port 0 has to be equal to the user-defined Pattern in KBPATN to generate the interrupt. When clear, Port 0 has to be not equal to the value of KBPATN register to generate the interrupt.
KBCON.0	KBIF	Keypad Interrupt Flag. Set when Port 0 matches user defined conditions specified in KBPATN, KBMASK, and PATN_SEL. Needs to be cleared by software by writing "0".

Figure 12-2: Keypad Control Register

KBMASK

Address: 86h

7	6	5	4	3	2	1	0
-	KBMASK.6	KBMASK.5	KBMASK.4	-	KBMASK.2	-	-

Not bit addressable

Reset Source(s): Any reset

Reset Value: 00000000B

BIT	**SYMBOL**	**FUNCTION**
KBMASK.7	-	Reserved.
KBMASK.6	-	When set, enables P0.6 as a cause of a Keypad Interrupt.
KBMASK.5	-	When set, enables P0.5 as a cause of a Keypad Interrupt.
KBMASK.4	-	When set, enables P0.4 as a cause of a Keypad Interrupt.
KBMASK.3	-	Reserved.
KBMASK.2	-	When set, enables P0.2 as a cause of a Keypad Interrupt.
KBMASK.1,0	-	Reserved.

Note: the Keypad Interrupt must be enabled in order for the settings of the KBMASK register to be effective.

Bits positions KBMASK.7, KBMASK.3, KBMASK.1, and KBMASK.0 should always be written as a ’0’.

Figure 12-3: Keypad Interrupt Mask Register (KBM)

13. WATCHDOG TIMER

The watchdog timer subsystem protects the system from incorrect code execution by causing a system reset when it underflows as a result of a failure of software to feed the timer prior to the timer reaching its terminal count. The watchdog timer can only be reset by a power-on reset.

WATCHDOG FUNCTION

The user has the ability using the WDCON and UCFG1 registers to control the run /stop condition of the WDT, the clock source for the WDT, the prescaler value, and whether the WDT is enabled to reset the device on underflow. In addition, there is a safety mechanism which forces the WDT to be enabled by values programmed into UCFG1 either through IAP or a commercial programmer.

The WDTE bit (UCFG1.7), if set, enables the WDT to reset the device on underflow. Following reset, the WDT will be running regardless of the state of the WDTE bit.

The WDRUN bit (WDCON.2) can be set to start the WDT and cleared to stop the WDT. Following reset this bit will be set and the WDT will be running. All writes to WDCON need to be followed by a feed sequence (see section "Feed Sequence" on page 90). Additional bits in WDCON allow the user to select the clocksource for the WDT and the prescaler.

When the timer is not enabled to reset the device on underflow, the WDT can be used in "timer mode" and be enabled to produce an interrupt (IEN0.6) if desired.

The Watchdog Safety Enable bit, WDSE (UCFG1.4) along with WDTE, is designed to force certain operating conditions at power-up. Refer to the Table for details.

Table 13-1: Watchdog timer configuration

WDTE (UCFG1.7)	WDSE (UCFG1.4)	FUNCTION
0	x	The watchdog reset is disabled. The timer can be used as an internal timer and can be used to generate an interrupt. WDSE has no effect.
1	0	The watchdog reset is enabled. The user can set WDCLK to choose the clock source.
1	1	The watchdog reset is enabled, along with additional safety features: 1. WDCLK is forced to 1 (using watchdog oscillator) 2. WDCON and WDL register can only be written once 3. WDRUN is forced to 1and cannot be cleared by software.

Figure 13-3 shows the watchdog timer in watchdog mode. It consists of a programmable 13-bit prescaler, and an 8-bit down counter. The down counter is clocked (decremented) by a tap taken from the prescaler. The clock source for the prescaler is either PCLK or the watchdog oscillator selected by the WDCLK bit in the WDCON register. (Note that switching of the clock sources will not take effect immediately - see section "Watchdog Clock Source" on page 94).

The watchdog asserts the watchdog reset when the watchdog count underflows and the watchdog reset is enabled. When the watchdog reset is enabled, writing to WDL or WDCON must be followed by a feed sequence for the new values to take effect.

If a watchdog reset occurs, the internal reset is active for at least one watchdog clock cycle (PCLK or the watchdog oscillator clock). If CCLK is still running, code execution will begin immediately after the reset cycle. If the processor was in Power down mode, the watchdog reset will start the oscillator and code execution will resume after the oscillator is stable.

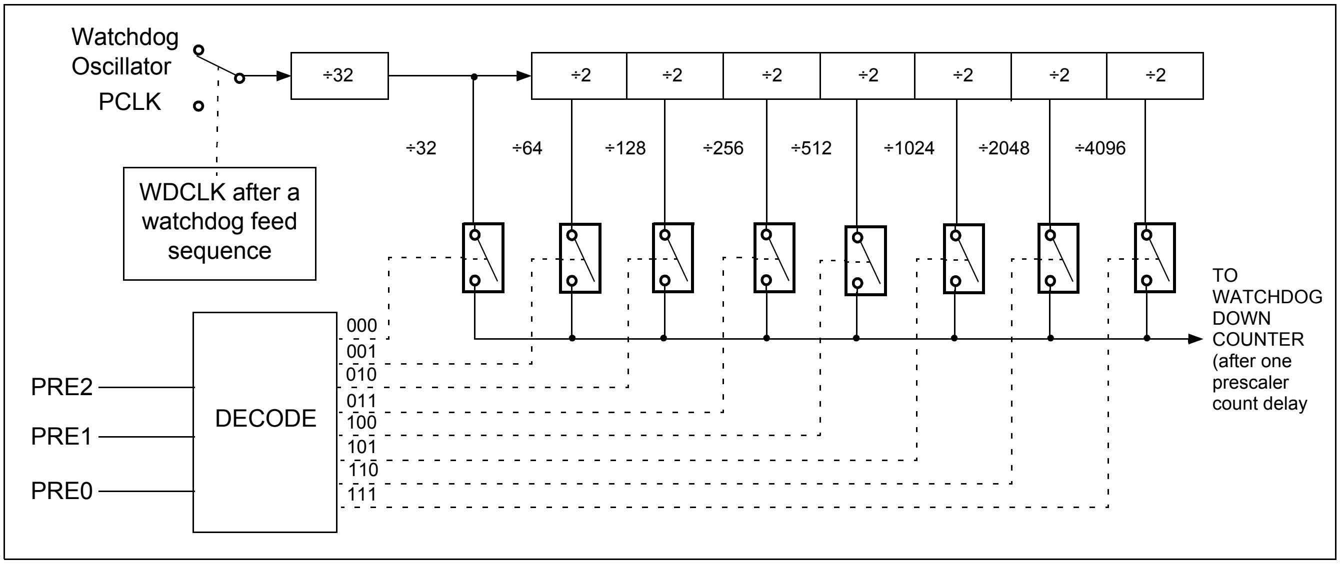


Figure 13-1: Watchdog Prescaler

FEED SEQUENCE

The watchdog timer control register and the 8-bit down counter (Figure 13-3) are not directly loaded by the user. The user writes to the WDCON and the WDL SFRs. At the end of a feed sequence, the values in the WDCON and WDL SFRs are loaded to the control register and the 8-bit down counter. Before the feed sequence, any new values written to these two SFRs will not take effect. To avoid a watchdog reset, the watchdog timer needs to be fed (via a special sequence of software action called the feed sequence) prior to reaching an underflow.

To feed the watchdog, two write instructions must be sequentially executed successfully. Between the two write instructions, SFR reads are allowed, but writes are not allowed. The instructions should move A5H to the WFEED1 register and then 5AH to the WFEED2 register. An incorrect feed sequence will cause an immediate watchdog reset. The program sequence to feed the watchdog timer is as follows:

```
CLR     EA                  ; disable interrupt
MOV     WFEED1,#0A5h        ; do watchdog feed part 1
MOV     WFEED2,#05Ah        ; do watchdog feed part 2
SETB    EA                  ; enable interrupt
```

This sequence assumes that the P89LPC912/913/914 interrupt system is enabled and there is a possibility of an interrupt request occuring during the feed sequence. If an interrupt was allowed to be serviced and the service routine contained any SFR writes, it would trigger a watchdog reset. If it is known that no interrupt could occur during the feed sequence, the instructions to disable and re-enable interrupts may be removed.

In watchdog mode (WDTE = 1), writing the WDCON register must be IMMEDIATELY followed by a feed sequence to load the WDL to the 8-bit down counter, and the WDCON to the shadow register. If writing to the WDCON register is not immediately followed by the feed sequence, a watchdog reset will occur.

For example: setting WDRUN = 1:

```
MOV     ACC,WDCON           ; get WDCON
SETB    ACC.2               ; set WD_RUN=1
MOV     WDL,#0FFh           ; New count to be loaded to 8-bit down counter
CLR     EA                  ; disable interrupt
MOV     WDCON,ACC           ; write back to WDCON (after the watchdog is enabled, a feed must occur
                            ; immediately)
```

```
MOV     WFEED1,#0A5h        ; do watchdog feed part 1
MOV     WFEED2,#05Ah        ; do watchdog feed part 2
SETB    EA                  ; enable interrupt
```

In timer mode (WDTE = 0), WDCON is loaded to the control register every CCLK cycle (no feed sequence is required to load the control register), but a feed sequence **is required** to load from the WDL SFR to the 8-bit down counter before a time-out occurs.

WDCON

Address: A7h

7	6	5	4	3	2	1	0
PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK

Not bit addressable

Reset Source(s): See reset value below

Reset Value: 111xx1?1B (Note: WDCON.7,6,5,2,0 - set to '1' any reset; WDCON.1 - cleared to '0' on Power-on reset, set to '1' on watchdog reset, not affected by any other reset)

BIT	SYMBOL	FUNCTION
WDCON.7-5	PRE2-PRE0	Clock Prescaler Tap Select. Refer to Table for details.
WDCON.4-3	-	Reserved for future use. Should not be set to 1 by user program.
WDCON.2	WDRUN	Watchdog Run Control. The watchdog timer is started when WDRUN = 1 and stopped when WDRUN = 0. This bit is forced to 1 (watchdog running) and cannot be reset to zero if both WDTE and WDSE are set to 1.
WDCON.1	WDTOF	Watchdog Timer Time-Out Flag. This bit is set when the 8-bit down counter underflows. In watchdog mode, a feed sequence will clear this bit. It can also be cleared by writing '0' to this bit in software.
WDCON.0	WDCLK	Watchdog input clock select. When set , the watchdog oscillator is selected. When cleared, PCLK is selected. (If the CPU is powered down, the watchdog is disabled if WDCLK = 0, see section "Power down operation"). (Note: If both WDTE and WDSE are set to 1, this bit is forced to 1.) Refer to section "Watchdog Clock Source" for details.

Figure 13-2: Watchdog Timer Control Register

The number of watchdog clocks before timing out is calculated by the following equations:

$$\text{tclks} = (2^{(5+\text{PRE})})(\text{WDL}+1)+1$$

where:

- PRE is the value of prescaler (PRE2-PRE0) which can be the range 0-7, and;
- WDL is the value of watchdog load register which can be the range of 0-255.

The minimum number of tclks is:

$$\text{tclks} = (2^{(5+0)})(0+1)+1 = 33$$

The maximum number of tclks is:

$$\text{tclks} = (2^{(5+7)})(255+1)+1 = 1{,}048{,}577$$

The following table shows sample P89LPC912/913/914 timeout values.

Table 13-2: P89LPC912/913/914 Watchdog Timeout Value

PRE2-PRE0	WDL in decimal)	Timeout Period (in watchdog clock cycles)	Watchdog Clock Source	
			400KHz Watchdog Oscillator Clock (Nominal)	12MHz CCLK (6MHz CCLK/2 Watchdog Clock)
000	0	33	82.5µs	5.50µs
	255	8,193	20.5ms	1.37ms
001	0	65	162.5µs	10.8µs
	255	16,385	41.0ms	2.73ms
010	0	129	322.5µs	21.5µs
	255	32,769	81.9ms	5.46ms
011	0	257	642.5µs	42.8µs
	255	65,537	163.8ms	10.9ms
100	0	513	.1.28ms	85.5µs
	255	131,073	327.7ms	21.8ms
101	0	1,025	2.56ms	170.8µs
	255	262,145	655.4ms	43.7ms
110	0	2,049	5.12ms	341.5µs
	255	524,289	1.31s	87.4ms
111	0	4097	10.2ms	682.8µs
	255	1,048,577	2.62s	174.8ms

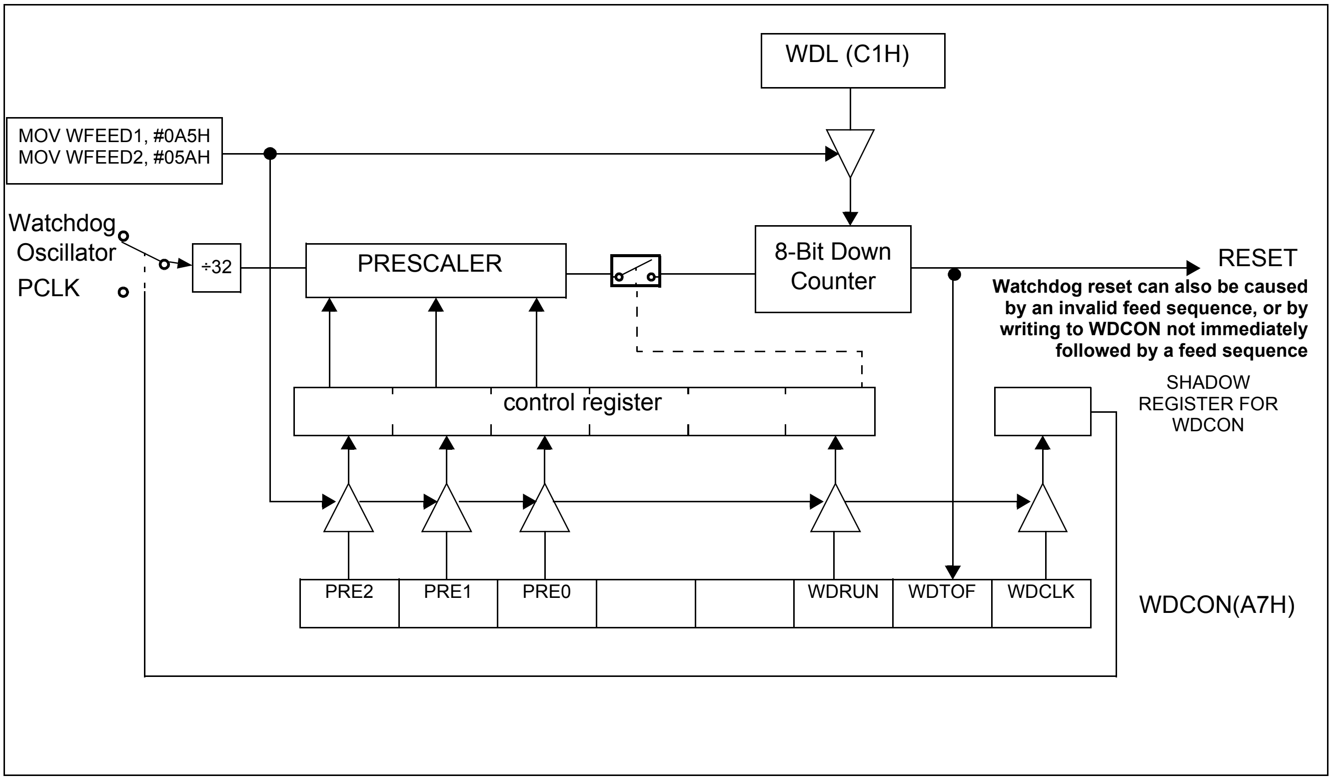


Figure 13-3: Watchdog Timer in Watchdog Mode (WDTE = 1)

WATCHDOG TIMER IN TIMER MODE

Figure 13-4 shows the Watchdog Timer in Timer Mode. In this mode, any changes to WDCON are written to the shadow register after one watchdog clock cycle. A watchdog underflow will set the WDTOF bit. If IEN0.6 is set, the watchdog underflow is enabled to cause an interrupt. WDTOF is cleared by writing a '0' to this bit in software. When an underflow occurs, the contents of WDL is reloaded into the down counter and the watchdog timer immediately begins to count down again.

A feed is necessary to cause WDL to be loaded into the down counter before an underflow occurs. Incorrect feeds are ignored in this mode.

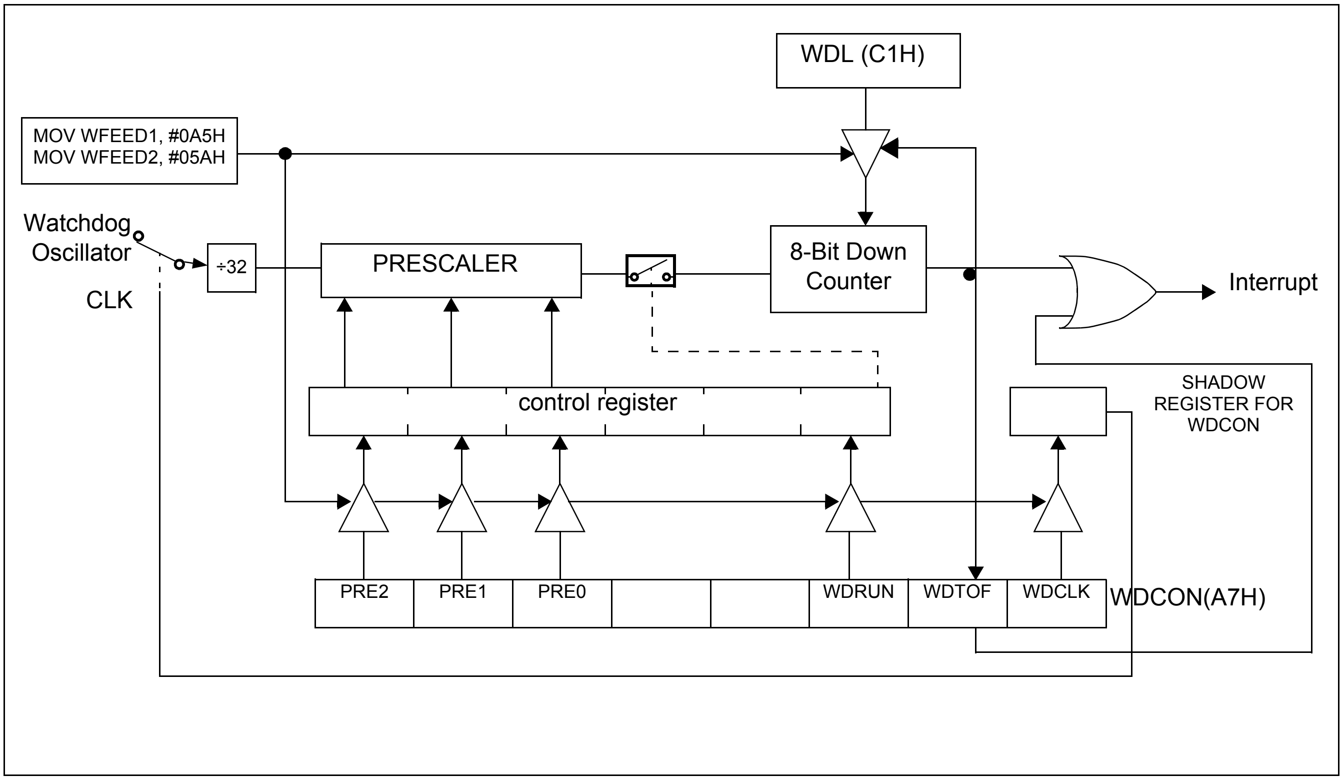


Figure 13-4: Watchdog Timer in Timer Mode (WDTE = 0)

POWER DOWN OPERATION

The WDT oscillator will continue to run in power down, consuming approximately 50uA, as long as the WDT oscillator is selected as the clock source for the WDT. Selecting PCLK as the WDT source will result in the WDT oscillator going into power down with the rest of the device (see section "Watchdog Clock Source", below). Power down mode will also prevent PCLK from running and therefore the watchdog is effectively disabled.

WATCHDOG CLOCK SOURCE

The watchdog timer system has an on-chip 400KHz oscillator. The watchdog timer can be clocked from either the watchdog oscillator or from PCLK (refer to Figure 13-1) by configuring the WDCLK bit in the Watchdog Control Register WDCON. When the watchdog feature is enabled, the timer must be fed regularly by software in order to prevent it from resetting the CPU.

After changing WDCLK (WDCON.0), switching of the clock source will not immediately take effect. As shown in Figure 13-3, the selection is loaded after a watchdog feed sequence. In addition, due to clock synchronization logic, it can take two old clock cycles before the old clock source is deselected, and then an additional two new clock cycles before the new clock source is selected.

Since the prescaler starts counting immediately after a feed, switching clocks can cause some inaccuracy in the prescaler count. The inaccuracy could be as much as 2 old clock source counts plus 2 new clock cycles.

Note: When switching clocks, it is important that the old clock source is left enabled for 2 clock cycles after the feed completes. Otherwise, the watchdog may become disabled when the old clock source is disabled. For example, suppose PCLK (WCLK=0) is the current clock source. After WCLK is set to '1', the program should wait at least two PCLK cycles (4 CCLKs) after the feed completes before going into Power down mode. Otherwise, the watchdog could become disabled when CCLK turns off. The watchdog oscillator will never become selected as the clock source unless CCLK is turned on again first.

PERIODIC WAKEUP FROM POWER DOWN WITHOUT AN EXTERNAL OSCILLATOR

Without using an external oscillator source, the power consumption required in order to have a periodic wakeup is determined by the power consumption of the internal oscillator source used to produce the wakeup.The Real-time clock running from the internal RC oscillator can be used. The power consumption of this oscillator is approximately 300uA. Instead, if the WDT is used to generate interrupts the current is reduced to approximately 50uA. Whenever the WDT underflows, the device will wake up.

14. ADDITIONAL FEATURES

The AUXR1 register contains several special purpose control bits that relate to several chip features. AUXR1 is described in Figure 14-1.

AUXR1

Address: A2h	7	6	5	4	3	2	1	0
	CLKLP	EBRR	-	-	SRST	0	-	DPS

Not bit addressable

Reset Source(s): Any reset

Reset Value: 000000x0B

BIT	SYMBOL	FUNCTION
AUXR1.7	CLKLP	Clock Low Power Select. When set, reduces power consumption in the clock circuits. Can be used when the clock frequency is 8MHz or less. After reset this bit is cleared to support up to 12MHz operation (**P89LPC912/913**).
AUXR1.6	EBRR	UART Break Detect Reset Enable. If '1', UART Break Detect will cause a chip reset (**P89LPC913/914**).
AUXR1.5	-	Reserved
AUXR1.4	-	Reserved
AUXR1.3	SRST	Software Reset. When set by software, resets the **P89LPC912/913/914** as if a hardware reset occurred.
AUXR1.2	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.
AUXR1.1	-	Not used. Allowable to set to a "1" .
AUXR1.0	DPS	Data Pointer Select. Chooses one of two Data Pointers.

Figure 14-1: AUXR1 Register

SOFTWARE RESET

The SRST bit in AUXR1 gives software the opportunity to reset the processor completely, as if an external reset or watchdog reset had occurred. If a value is written to AUXR1 that contains a 1 at bit position 3, all SFRs will be initialized and execution will resume at program address 0000. Care should be taken when writing to AUXR1 to avoid accidental software resets.

DUAL DATA POINTERS

The dual Data Pointers (DPTR) adds to the ways in which the processor can specify the address used with certain instructions. The DPS bit in the AUXR1 register selects one of the two Data Pointers. The DPTR that is not currently selected is not accessible to software unless the DPS bit is toggled.

Specific instructions affected by the Data Pointer selection are:

• INC DPTR	Increments the Data Pointer by 1.
• JMP @A+DPTR	Jump indirect relative to DPTR value.
• MOV DPTR, #data16	Load the Data Pointer with a 16-bit constant.
• MOVCA, @A+DPTR	Move code byte relative to DPTR to the accumulator.
• MOVXA, @DPTR	Move data byte the accumulator to data memory relative to DPTR.

- MOVX@DPTR, A Move data byte from data memory relative to DPTR to the accumulator.

Also, any instruction that reads or manipulates the DPH and DPL registers (the upper and lower bytes of the current DPTR) will be affected by the setting of DPS. The MOVX instructions have limited application for the P89LPC912/913/914 since the part does not have an external data bus. However, they may be used to access Flash configuration information (see Flash Configuration section).

Bit 2 of AUXR1 is permanently wired as a logic 0. This is so that the DPS bit may be toggled (thereby switching Data Pointers) simply by incrementing the AUXR1 register, without the possibility of inadvertently altering other bits in the register.

15. FLASH PROGRAM MEMORY

GENERAL DESCRIPTION

The **P89LPC912/913/914** Flash memory provides in-circuit electrical erasure and programming. The Flash can be read and written as bytes. On-chip erase and write timing generation contribute to a user-friendly programming interface. The cell is designed to optimize the erase and programming mechanisms. The **P89LPC912/913/914** uses V_{DD} as the supply voltage to perform the Program/Erase algorithms. Additionally, serial programming using commercially available programmers provides a simple inteface to achieve in-circuit programming.The **P89LPC912/913/914** Flash reliably stores memory contents after 100,000 erase and program cycles (typical).

FEATURES

- IAP-Lite allows individual and multiple bytes of code memory to be used for data storage.
- Programming and erase over the full operating voltage range
- Read/Programming/Erase using IAP-Lite
- Any flash program operation in 2 ms (4ms for erase/program)
- Serial programming with industry-standard commercial programmers allows in-circuit programming.
- Programmable security for the code in the Flash for each sector.
- >100,000 typical erase/program cycles for each byte.
- 256 byte sector size, 16 byte page size
- 10-year minimum data retention.

INTRODUCTION TO IAP-LITE

The Flash code memory array of this device supports IAP-Lite programming and erase functions. Any byte in a non-secured sector of the code memory array may be read using the MOVC instruction and thus is suitable for use as non-volatile data storage. In addition, the user's code may access additional flash elements. These include UCFG1, the Boot Vector, Status Bit, security bytes, and signature bytes. Access of these elements uses a slightly different method than that used to access the user code memory.

USING FLASH AS DATA STORAGE

IAP-Lite provides an erase-program function that makes it easy for one or more bytes within a page to be erased and programmed in a single operation without the need to erase or program any other bytes in the page. IAP-Lite is performed in the application under the control of the microcontroller's firmware using four SFRs and an internal 16-byte "page register" to facilitate erasing and programming within unsecured sectors. These SFRs are:

- FMCON (Flash Control Register). When read, this is the status register. When written, this is a command register. Note that the status bits are cleared to '0's when the command is written.
- FMDATA (Flash Data Register). Accepts data to be loaded into the page register.
- FMADRL, FMADRH (Flash memory address low, Flash memory address high). Used to specify the byte address within the page register or specify the page within user code memory.

The page register consists of 16 bytes and an update flag for each byte. When a LOAD command is issued to FMCON the page register contents and all of the update flags will be cleared. When FMDATA is written, the value written to FMDATA will be stored in the page register at the location specified by the lower 6 bits of FMADRL. In addition, the update flag for that location will be set. FMADRL will auto-increment to the next location. Auto-increment after writing to the last byte in the page register will "wrap -around" to the first byte in the page register, but will not affect FMADRL[7:4]. Bytes loaded into the page register do not have to be continuous. Any byte location can be loaded into the page register by changing the contents of FMADRL prior to writing to FMDATA. However, each location in the page register can only be written once following each LOAD command. Attempts to write to a page register location more than once should be avoided.

FMADRH and FMADRL[7:4] are used to select a page of code memory for the erase-program function. When the erase-program command is written to FMCON, the locations within the code memory page that correspond to updated locations in the page register will have their contents erased and programmed with the contents of their corresponding locations in the page register. Only the bytes that were loaded into the page register will be erased and programmed in the user code array. Other bytes within the user code memory will not be affected.

Writing the erase-program command (68H) to FMCON will start the erase-program process and place the CPU in a program-idle state. The CPU will remain in this idle state until the erase-program cycle is either completed or terminated by an interrupt. When the program-idle state is exited, FMCON will contain status information for the cycle.

If an interrupt occurs during an erase/programming cycle, the erase/programming cycle will be aborted and the OI flag (Operation Interrupted) in FMCON will be set. If the application permits interrupts during erasing-programming, the user code should check the OI flag (FMCON.0) after each erase-programming operation to see if the operation was aborted. If the operation was aborted, the user's code will need to repeat the process starting with loading the page register.

The erase-program cycle takes 4ms to complete, regardless of the number of bytes that were loaded into the page register.

Erasing-programming of a single byte (or multiple bytes) in code memory is accomplished using the following steps:

- Write the LOAD command (00H) to FMCON. The LOAD command will clear all locations in the page register and their corresponding update flags.
- Write the address within the page register to FMADRL. Since the loading the page register uses FMADRL[5:0], and since the erase-program command uses FMADRH and FMADRL[7:4], the user can write the byte location within the page register (FMADRL[3:0]) and the code memory page address (FMADRH and FMADRL[7:4]) at this time.
- Write the data to be programmed to FMDATA. This will increment FMADRL pointing to the next byte in the page register.
- Write the address of the next byte to be programmed to FMADRL, if desired. (Not needed for contiguous bytes since FMADRL is auto-incremented). All bytes to be programmed must be within the same page.
- Write the data for the next byte to be programmed to FMDATA.
- Repeat writing of FMADRL and/or FMDATA until all desired bytes have been loaded into the page register.
- Write the page address in user code memory to FMADRH and FMADRL[7:4], if not previously included when writing the page register address to FMADRL[3:0].
- Write the erase-program command (68H) to FMCON,starting the erase-program cycle.
- Read FMCON to check status. If aborted, repeat starting with the LOAD command.

An assembly language routine to load the page register and perform an erase/program operation is shown in Figure 15-2. A similar C-language routine is shown in Figure 15-3.

FMCON

Address: E4h

Not bit addressable

Reset Source(s): Any reset

Reset Value:

7	6	5	4	3	2	1	0
-	-	-	-	HVA	HVE	SV	OI

BIT	SYMBOL	FUNCTION
FMCON.7-4	-	Reserved.
FMCON.3	HVA	High voltage abort. Set if either an interrupt or a brown-out is detected during a program or erase cycle. Also set if the brown-out detector is disabled at the start of a program or erase cycle.
FMCON.2	HVE	High voltage error. Set when an error occurs in the high voltage generator.
FMCON.1	SV	Security violation. Set when an attempt is made to program, erase, or CRC a secured sector or page.

Figure 15-1: Flash Memory Control Register

```
;* Inputs:                                        *
;*    R3 = number of bytes to program (byte)      *
;*    R4 = page address MSB(byte)                 *
;*    R5 = page address LSB(byte)                 *
;*    R7 = pointer to data buffer in RAM(byte)    *
;* Outputs:                                       *
;*    R7 = status (byte)                          *
;*     C = clear on no error, set on error        *

LOAD  EQU   00H
EP    EQU   68H

PGM_USER:
      MOV   FMCON,#LOAD       ;load command, clears page register
      MOV   FMADRH,R4         ;get high address
      MOV   FMADRL,R5         ;get low address
      MOV   A,R7              ;
      MOV   R0,A              ;get pointer into R0
LOAD_PAGE:
      MOV   FMDAT,@R0         ;write data to page register
      INC   R0                ;point to next byte
      DJNZ  R3,LOAD_PAGE      ;do until count is zero
      MOV   FMCON,#EP         ;else erase & program the page

      MOV   R7,FMCON          ;copy status for return
      MOV   A,R7              ;read status
      ANL   A,#0FH            ;save only four lower bits
      JNZ   BAD               ;
      CLR   C                 ;clear error flag if good
      RET                     ;and return
BAD:
      SETB  C                 ;set error flag
      RET                     ;and return
```

Figure 15-2: Assembly language routine to erase/program all or part of a page

```
unsigned char idata dbytes[16];           // data buffer
unsigned char Fm_stat;              // status result

bit PGM_USER (unsigned char, unsigned char);
bit prog_fail;

void main ()

{
      prog_fail=PGM_USER(0x1F,0xC0);
}


bit PGM_USER (unsigned char page_hi, unsigned char page_lo)
      {
            #define LOAD      0x00  // clear page register, enable loading
            #define EP        0x68  // erase & program page
            unsigned char     i;    // loop count

      FMCON = LOAD;                 //load command, clears page reg
      FMADRH = page_hi;       //
      FMADRL = page_lo;       //write my page address to addr regs

      for   (i=0;i<16;i=i+1)
            {
                  FMDATA = dbytes[i];
            }
      FMCON = EP;             //erase & prog page command
      Fm_stat = FMCON;              //read the result status
      if ((Fm_stat & 0x0F)!=0) prog_fail=1; else prog_fail=0;
      return(prog_fail);
      }
```

Figure 15-3: C-language routine to erase/program all or part of a page

ACCESSING ADDITIONAL FLASH ELEMENTS

In addition to the user code array, the user's firmware may access additional flash elements. These include UCFG1, the Boot Vector, Status Bit, security bytes, and signature bytes. Access of these elements uses a slightly different method than that used to access the user code memory. Signature bytes are read-only. Security bytes may be erased only under certain conditions. IAP-Lite is performed in the application under the control of the microcontroller's firmware using four SFRs to facilitate erasing, programming, or reading. These SFRs are:

- FMCON (Flash Control Register). When read, this is the status register. When written, this is a command register. Note that the status bits are cleared to '0's when the command is written.
- FMDATA (Flash Data Register). Accepts data to be loaded into or from the flash element.
- FMADRL (Flash memory address low). Used to specify the flash element.

The flash elements that may be accessed and their addresses are shown in Table 15-1.

Table 15-1: Flash elements accesable through IAP-Lite

Element	Address	Description
UCFG1	00h	User Configuration byte 1.
Boot Vector	02h	Boot vector
Status Bit	03h	Status bit byte
Security byte 0	08h	Security byte, sector 0
Security byte 1	09h	Security byte, sector 1
Security byte 2	0Ah	Security byte, sector 2
Security byte3	0Bh	Security byte, sector 3
Mfgr Id	10h	Signature byte, manufacturer id
Id_1	11h	Signature byte,id 1
Id_2	12h	Signature byte,id 2

ERASE-PROGRAMMING ADDITIONAL FLASH ELEMENTS

The erase-program cycle takes 4ms to complete and is accomplished using the following steps:

- Write the address of the flash element to FMADRL.
- Write the CONF command (6CH) to FMCON.
- Write the data to be programmed to FMDATA.
- Read FMCON to check status. If aborted, repeat this sequence.

Writing the data to be programmed to FMDATA will start the erase-program process and place the CPU in a program-idle state. The CPU will remain in this idle state until the erase-program cycle is either completed or terminated by an interrupt. When the program-idle state is exited, FMCON will contain status information for the cycle.

If an interrupt occurs during an erase/programming cycle, the erase/programming cycle will be aborted and the OI flag (Operation Interrupted) in FMCON will be set. If the application permits interrupts during erasing-programming the user code should check the OI flag (FMCON.0) after each erase-programming operation to see if the operation was aborted. If the operation was aborted, the user's code will need to repeat the process.

READING ADDITIONAL FLASH ELEMENTS

The read cycle is accomplished using the following steps:

- Write the address of the flash element to FMADRL.
- Write the CONF command (6CH) to FMCON.
- Read the data from FMDATA

The read cycle completes in a single machine cycle and thus will not enter an idle state. It can be interrupted. However, there is no need to check status.

An assembly language routine to perform an erase/program operation of a flash element is shown in Figure 15-4. A similar C-language routine is shown in Figure 15-5. A C-language routine to read a flash element is shown in Figure 15-6.

r

```
;* Inputs:                                        *
;*    R5 = data to write(byte)                    *
;*    R7 = element address(byte)                  *
;* Outputs:                                       *
;*    None                                        *

CONF  EQU   6CH

WR_ELEM:
      MOV   FMADRL,R7         ;write the address
      MOV   FMCON,#CONF       ;load CONF command
      MOV   FMDAT,R5          ;write the data
      MOV   R7,FMCON          ;copy status for return
      MOV   A,R7              ;read status
      ANL   A,#0FH            ;save only four lower bits
      JNZ   BAD               ;see if good or bad
      CLR   C                 ;clear error flag if good
      RET                     ;and return
BAD:
      SETB  C                 ;set error flag if bad
      RET                     ;and return
```

Figure 15-4: Assembly language routine to erase/program a flash element

```
unsigned char     Fm_stat;                // status result

bit PGM_EL (unsigned char, unsigned char);
bit prog_fail;

void main ()

{
      prog_fail=PGM_EL(0x02,0x1C);
}

bit PGM_EL (unsigned char el_addr, unsigned char el_data)
      {
            #define CONF      0x6C        // access flash elements

      FMADRL      = el_addr;        //write element address to addr reg
      FMCON = CONF;           //load command, clears page reg
      FMDATA      = el_data;        //write     data and start the cycle
      Fm_stat = FMCON;              //read the result status
      if ((Fm_stat & 0x0F)!=0) prog_fail=1; else prog_fail=0;
      return(prog_fail);
      }
```

Figure 15-5: C-language routine to erase/program a flash element

```
#include <REG921.H>

unsigned char READ_EL (unsigned char);
unsigned char GET_EL;

void main ()

{
      GET_EL = READ_EL(0x02);
}

unsigned char READ_EL (unsigned char el_addr)
      {
            #define CONF      0x6C        // access flash elements
            unsigned char el_data;        // local for element data
      FMADRL      = el_addr;              //write element address to addr reg
      FMCON = CONF;                 //access flash elements command
      el_data     = FMDATA;               /read the element data
      return(el_data);
      }
```

Figure 15-6: C-language routine to read a flash element

USER CONFIGURATION BYTES

A number of user-configurable features of the **P89LPC912/913/914** must be defined at power-up and therefore cannot be set by the program after start of execution. These features are configured through the use of Flash byte UCFG1 shown in Figure 15-7.

UCFG1

Address: xxxxh

7	6	5	4	3	2	1	0
WDTE	RPE	BOE	WDSE	-	FOSC2	FOSC1	FOSC0

Default: 63h

BIT	**SYMBOL**	**FUNCTION**
UCFG1.7	WDTE	Watchdog timer reset enable. When set =1, enables the watchdog timer reset. When cleared = 0, dusables the watchdog timer reset.The timer may still be used to generate an interrupt. Refer to Table 13-1 for details.
UCFG1.6	RPE	Reset pin enable. When set =1, enables the reset function of pin P1.5. When cleared, P1.5 may be used as an input pin. NOTE: During a power-up sequence, the RPE selection is overriden and this pin will always functions as a reset input. After power-up the pin will function as defined by the RPE bit. Only a power-up reset will temporarily override the selection defined by RPE bit. Other sources of reset will not override the RPE bit.
UCFG1.5	BOE	Brownout Detect Enable (see section "Power Monitoring Functions" on page 55).
UCFG1.4	WDSE	Watchdog Safety Enable bit. Refer to Table for details.
UCFG1.3	-	Reserved (should remain unprogrammed at zero).
UCFG1.2-0	FOSC2-FSOC0	CPU oscillator type select. See section "Low Power Select (P89LPC912/913)" on page 35 for additional information. Combinations other than those shown below should not be used. They are reserved for future use.When FOSC2:0 select either the internal RC or Watchdog oscillators, the crystal oscillator configuration is controlled by RTCCON. See Table and Table . **Note: External clock input and crystal options are available on the P89LPC912/913.**

FOSC2-FOSC0	Oscillator Configuration
1 1 1	External clock input on XTAL1.
1 0 0	Watchdog Oscillator, 400KHz (+20/ -30% tolerance).
0 1 1	Internal RC oscillator, 7.373MHz ±2.5%.
0 1 0	Low frequency crystal, 20 kHz to 100 kHz.
0 0 1	Medium frequency crystal or resonator, 100 kHz to 4 MHz.
0 0 0	High frequency crystal or resonator, 4 MHz to 12 MHz.

Factory default value for UCFG1 is set for watchdog reset disabled, reset pin enabled, brownout detect enabled, and using the internal RC oscillator

Figure 15-7: Flash User Configuration Byte 1 (UCFG1)

USER SECURITY BYTES

There are four User Sector Security Bytes (SEC0, ..., SEC3), each corresponding to one sector and having the following bit assignments:

SECx

Address: xxxxh

7	6	5	4	3	2	1	0
-	-	-	-	-	EDISx	SPEDISx	MOVCDISx

Unprogrammed value: 00h

BIT	SYMBOL	FUNCTION
SECx.7-3	-	Reserved (should remain unprogrammed at zero).
SECx.2	EDISx	**Erase Disable x**. Disables the ability to perform an erase of sector "x" in IAP mode. When programmed, this bit and sector x can only be erased by a 'global' erase command using a commercial programmer . This bit and sector x CANNOT be erased in IAP mode.
SECx.1	SPEDISx	**Sector Program Erase Disable x**. Disables program or erase of **all or part of sector x**. This bit and sector x are erased by either a sector erase command (IAP or commercial programmer) or a 'global' erase command (commercial programmer).
SECx.0	MOVCDISx	**MOVC Disable**. Disables the MOVC command for sector x. Any MOVC that attempts to read a byte in a MOVC protected sector will return invalid data. This bit can only be erased when sector x is erased.

Figure 15-8: User Sector Security Bytes (SEC0 ... SEC3)

Table 15-2: Effects of Security Bits

EDISx	SPEDISx	MOVCDISx	Effects on Programming
0	0	0	None.
0	0	1	Security violation flag set for sector CRC calculation for the specific sector. Security violation flag set for global CRC calculation if any MOVCDISx bit is set. Cycle aborted. Memory contents unchanged. CRC invalid. Program/erase commands will not result in a security violation.
0	1	x	Security violation flag set for program commands or an erase page command. Cycle aborted. Memory contents unchanged. Sector erase and global erase are allowed.
1	x	x	Security violation flag set for program or erase commands. Cycle aborted. Memory contents unchanged. Global erase is allowed.

Boot Vector

BOOTVEC
Address: xxxxh

7	6	5	4	3	2	1	0
-	-	-	BOOTV4	BOOTV3	BOOTV2	BOOTV1	BOOTV0

Factory default value: 00h

BIT	SYMBOL	FUNCTION
BOOTVEC.7-5	-	Reserved (should remain unprogrammed at zero).
BOOTVEC.4-0	-	Boot Vector. If the Boot Vector is selected as the reset address, the **P89LPC912/913/914** will start execution at an address comprised of 00H in the lower eight bits and this BOOTVEC as the upper bits after a reset. (See section "Power-On reset code execution" on page 81).

Figure 15-9: Boot Vector (BOOTVEC)

Boot Status

BOOTSTAT
Address: xxxxh

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BSB

Factory default value: 00h

BIT	SYMBOL	FUNCTION
BOOTSTAT.7-1	-	Reserved (should remain unprogrammed at zero).
BOOTSTAT.0	BSB	Boot Status Bit. If programmed to ‘1’, the **P89LPC912/913/914** will always start execution at an address comprised of 00H in the lower eight bits and BOOTVEC as the upper bits after a reset. (See section "Power-On reset code execution" on page 81).

Figure 15-10: Boot Status (BOOTSTAT)

16. INSTRUCTION SET

Table 1: Instruction set summary

Mnemonic	Description	Bytes	Cycles	Hex code
	ARITHMETIC			
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4
	LOGICAL			
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54

Mnemonic	Description	Bytes	Cycles	Hex code
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13
	DATA TRANSFER			
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87

Mnemonic	Description	Bytes	Cycles	Hex code
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	94
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7
	BOOLEAN			
Mnemonic	Description	Bytes	Cycles	Hex code
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2
ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92
	BRANCHING			

Mnemonic	Description	Bytes	Cycles	Hex code
ACALL addr 11	Absolute jump to subroutine	2	2	116F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	016E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ¹ 0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5
	MISCELLANEOUS			
NOP	No operation	1	1	00

17. REVISION HISTORY

2003 Dec 8

Initial release.

REVISION HISTORY

18. INDEX

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products—including circuits, standard cells, and/or software—described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Contact information

For additional information please visit
http://www.semiconductors.philips.com. Fax: **+31 40 27 24825**

For sales offices addresses send e-mail to:
sales.addresses@www.semiconductors.philips.com.

© Koninklijke Philips Electronics N.V. 2003
All rights reserved. Printed in U.S.A.

Date of release: 12-03

Document order number: 9397 750 12492

Let's make things better.

Philips
Semiconductors

PHILIPS