Semiconductors

UM10111

P89LPC904 User manual

Rev. 01 — 2 March 2005

User manual

Document information

Info	Content
Keywords	P89LPC904
Abstract	Technical information for the P89LPC904 device.

PHILIPS

Revision history

Rev	Date	Description
01	20050302	Initial version.

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

The P89LPC904 is a single-chip microcontroller designed for applications demanding high-integration, low cost solutions over a wide range of performance requirements. The P89LPC904 is based on a high performance processor architecture that executes instructions in two to four clocks, six times the rate of standard 80C51 devices. Many system-level functions have been incorporated into the P89LPC904 in order to reduce component count, board space, and system cost.

1.1 Pin configuration

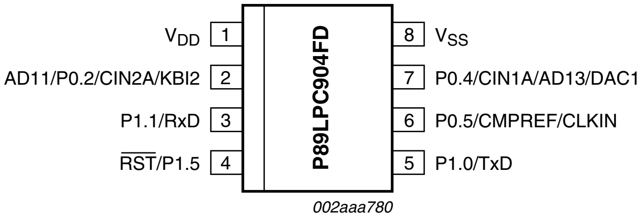


Fig 1. SO8 pin configuration.

Table 1: Pin description

Symbol	Pin	Type	Description
P0.0 to P0.6		I/O	**Port 0:** Port 0 is an I/O port with a user-configurable output type. During reset Port 0 latches are configured in the input only mode with the internal pull-up 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 Section 5.1 “Port configurations” 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:
	2	I/O	**P0.2** — Port 0 bit 2.
		I	**CIN2A** — Comparator 2 positive input A.
		I	**KBI2** — Keyboard input 2.
		I	**AD11** — ADC1 channel 1analog input.
	7	I/O	**P0.4** — Port 0 bit 4.
		I	**CIN1A** — Comparator 1 positive input A.
		I	**KBI4** — Keyboard input 4.
		I	**AD13** — ADC1 channel 3 analog input.
		I	**DAC1** — Digital-to-analog converter output 1.
	6	I/O	**P0.5** — Port 0 bit 5.
		I	**CMPREF** — Comparator reference (negative) input.
		I	**CLKIN** — External clock input.

Table 1: Pin description

Symbol	Pin	Type	Description
P1.0 to P1.5		I/O, I [1]	**Port 1:** Port 1 is an I/O port with a user-configurable output type, P1.5. During reset Port 1 latches are configured in the input only mode with the internal pull-up 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 Section 5.1 "Port configurations" for details. P1.5 is input only. All pins have Schmitt triggered inputs. Port 1 also provides various special functions as described below:
	5	I/O	**P1.0** — Port 1 bit 0.
		O	**TXD** — Transmitter output for the serial port.
	3	I/O	**P1.1** — Port 1 bit 1.
		I	**RXD** — Receiver input for the serial port.
	4	I	**P1.5** — Port 1 bit 5 (input only).
		I	**$\overline{\text{RST}}$** — External Reset input (if selected via FLASH configuration). 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. **When using an external clock frequency above 12 MHz, the reset input function of P1.5 must be enabled. An external circuit is required to hold the device in reset at power-up until V_{DD} has reached its specified level. When system power is removed V_{DD} will fall below the minimum specified operating voltage. When using an external clock frequency above 12 MHz, in some applications, an external brownout detect circuit may be required to hold the device in reset when V_{DD} falls below the minimum specified operating voltage.**
V_{SS}	8	I	**Ground:** 0 V reference.
V_{DD}	1	I	**Power Supply:** This is the power supply voltage for normal operation as well as Idle and Power-down modes.

[1] Input/Output for P1.0, P1.1. Input for P1.5.

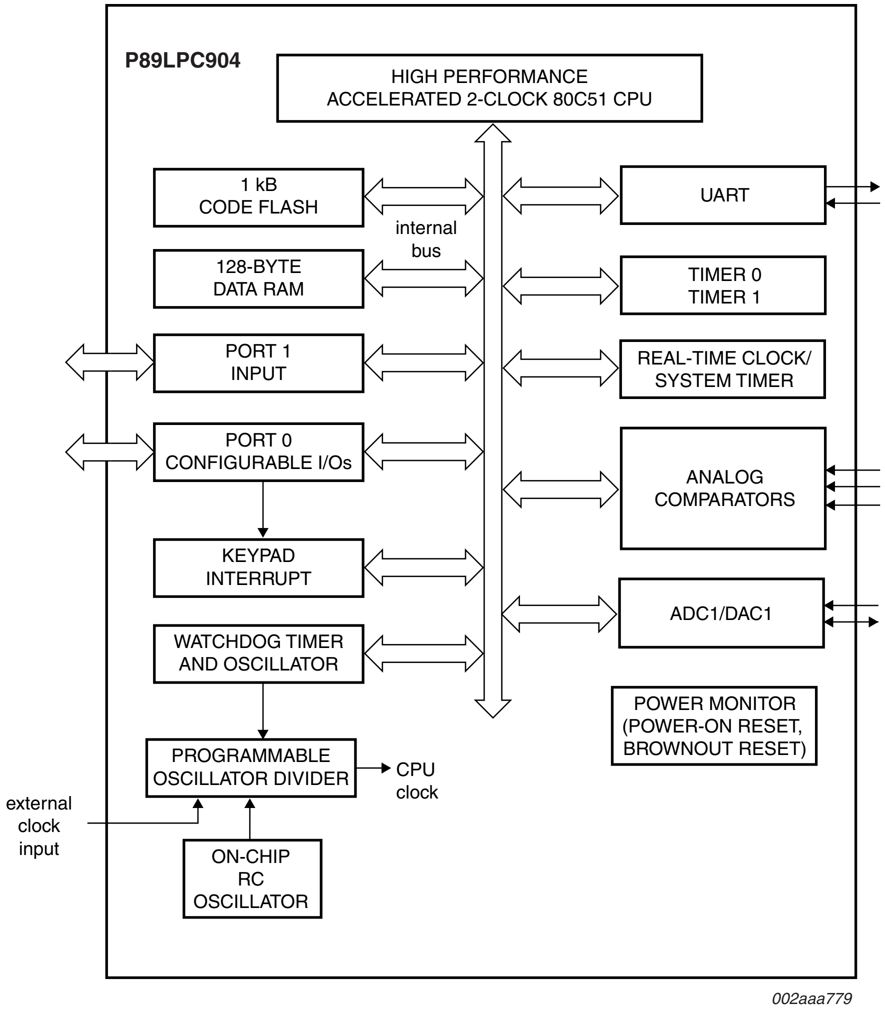


Fig 2. P89LPC904 block diagram.

1.2 Special function registers

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

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

Table 2: P89LPC904 Special function registers

** indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses								Reset value	
			MSB							LSB	Hex	Binary
		Bit address	E7	E6	E5	E4	E3	E2	E1	E0		
ACC*	Accumulator	E0H									00	00000000
ADCON1	A/D control register 1	97H	ENBI1	ENADCI1	TMM1	EDGE1	ADCI1	ENADC1	ADCS11	ADCS10	00	00000000
ADINS	A/D input select	A3H	ADI13	-	ADI11	-	-	-	-	-	00	00000000
ADMODA	A/D mode register A	C0H	BNDI1	BURST1	SCC1	SCAN1	-	-	-	-	00	00000000
ADMODB	A/D mode register B	A1H	CLK2	CLK1	CLK0	-	ENDAC1	-	BSA1	-	00	000x0000
AD1BH	A/D_1 boundary HIGH register	C4H									FF	11111111
AD1BL	A/D_1 boundary LOW register	BCH									00	00000000
AD1DAT0	A/D_1 data register 0	D5H									00	00000000
AD1DAT1	A/D_1 data register 1	D6H									00	00000000
AD1DAT2	A/D_1 data register 2	D7H									00	00000000
AD1DAT3	A/D_1 data register 3	F5H									00	00000000
AUXR1	Auxiliary function register	A2H	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS	00[1]	000000x0
		Bit address	F7	F6	F5	F4	F3	F2	F1	F0		
B*	B register	F0H									00	00000000
BRGR0[2]	Baud rate generator rate LOW	BEH									00	00000000
BRGR1[2]	Baud rate generator rate HIGH	BFH									00	00000000
BRGCON	Baud rate generator control	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00	xxxxxx00
CMP1	Comparator 1 control register	ACH	-	-	CE1	-	CN1	-	CO1	CMF1	00[1]	xx000000
CMP2	Comparator 2 control register	ADH	-	-	CE2	-	CN2	-	CO2	CMF2	00[1]	xx000000
DIVM	CPU clock divide-by-M control	95H									00	00000000
DPTR	Data pointer (2 bytes)											
DPH	Data pointer HIGH	83H									00	00000000
DPL	Data pointer LOW	82H									00	00000000
FMADRH	Program Flash address HIGH	E7H									00	00000000

Table 2: P89LPC904 Special function registers

** indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses								Reset value	
			MSB							LSB	Hex	Binary
FMADRL	Program Flash address LOW	E6H									00	00000000
FMCON	Program Flash control (Read)	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70	01110000
	Program Flash control (Write)	E4H	FMCMD. 7	FMCMD. 6	FMCMD. 5	FMCMD. 4	FMCMD. 3	FMCMD. 2	FMCMD. 1	FMCMD. 0		
FMDATA	Program Flash data	E5H									00	00000000
		Bit address	**AF**	**AE**	**AD**	**AC**	**AB**	**AA**	**A9**	**A8**		
IEN0*	Interrupt enable 0	A8H	EA	EWDRT	EBO	ES/ESR	ET1	EX1	ET0	EX0	00[1]	00000000
		Bit address	**EF**	**EE**	**ED**	**EC**	**EB**	**EA**	**E9**	**E8**		
IEN1*	Interrupt enable 1	E8H	EAD	EST	-	-	-	EC	EKBI	EI2C	00[1]	00x00000
		Bit address	**BF**	**BE**	**BD**	**BC**	**BB**	**BA**	**B9**	**B8**		
IP0*	Interrupt priority 0	B8H	-	PWDRT	PBO	PS/PSR	PT1	PX1	PT0	PX0	00[1]	x0000000
IP0H	Interrupt priority 0 HIGH	B7H	-	PWDRT H	PBOH	PSH/ PSRH	PT1H	PX1H	PT0H	PX0H	00[1]	x0000000
		Bit address	**FF**	**FE**	**FD**	**FC**	**FB**	**FA**	**F9**	**F8**		
IP1*	Interrupt priority 1	F8H	PAD	PST	-	-	-	PC	PKBI	PI2C	00[1]	00x00000
IP1H	Interrupt priority 1 HIGH	F7H	PADH	PSTH	-	-	-	PCH	PKBIH	PI2CH	00[1]	00x00000
KBCON	Keypad control register	94H	-	-	-	-	-	-	PATN _SEL	KBIF	00[1]	xxxxxx00
KBMASK	Keypad interrupt mask register	86H									00	00000000
KBPATN	Keypad pattern register	93H									FF	11111111
		Bit address	**87**	**86**	**85**	**84**	**83**	**82**	**81**	**80**		
P0*	Port 0	80H	-	-	CMPREF /CLKIN	CINIA	-	CIN2A/ KBI2				[1]
		Bit address	**97**	**96**	**95**	**94**	**93**	**92**	**91**	**90**		
P1*	Port 1	90H	-	-	$\overline{\text{RST}}$	-	-	-	RXD	TXD		[1]
P0M1	Port 0 output mode 1	84H	-	-	(P0M1.5)	(P0M1.4)	-	(P0M1.2)	-	-	FF	11111111
P0M2	Port 0 output mode 2	85H	-	-	(P0M2.5)	(P0M2.4)	-	(P0M2.2)	-	-	00	00000000
P1M1	Port 1 output mode 1	91H	-	-	-	-	-	-	(P1M1.1)	(P1M1.0)	D3[1]	xxxxxx11
P1M2	Port 1 output mode 2	92H	-	-	-	-	-	-	(P1M2.1)	(P1M2.0)	00[1]	xxxxxx00

Table 2: P89LPC904 Special function registers

** indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses								Reset value	
			MSB							LSB	Hex	Binary
PCON	Power control register	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00	00000000
PCONA	Power control register A	B5H	RTCPD	-	VCPD	ADPD	-	-	SPD	-	00[1]	00000000
Bit address			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	xx00x0xx
RSTSRC	Reset source register	DFH	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX		[3]
RTCCON	Real-time clock control	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60[1][6]	
RTCH	Real-time clock register HIGH	D2H									00[6]	00000000
RTCL	Real-time clock register LOW	D3H									00[6]	00000000
SADDR	Serial port address register	A9H									00	00000000
SADEN	Serial port address enable	B9H									00	00000000
SBUF	Serial Port data buffer register	99H									xx	xxxxxxxx
Bit address			9F	9E	9D	9C	9B	9A	99	98		
SCON*	Serial port control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00	00000000
SSTAT	Serial port extended status register	BAH	DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT	00	00000000
SP	Stack pointer	81H									07	00000111
Bit address			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer 0 and 1 control	88H	TF1	TR1	TF0	TR0	-	-	-	-	00	00000000
TH0	Timer 0 HIGH	8CH									00	00000000
TH1	Timer 1 HIGH	8DH									00	00000000
TL0	Timer 0 LOW	8AH									00	00000000
TL1	Timer 1 LOW	8BH									00	00000000
TMOD	Timer 0 and 1 mode	89H	-	-	T1M1	T1M0	-	-	T0M1	T0M0	00	00000000
TRIM	Internal oscillator trim register	96H	RCCLK	-	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0		[5] [6]
WDCON	Watchdog control register	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK		[4] [6]
WDL	Watchdog load	C1H									FF	11111111

Table 2: P89LPC904 Special function registers

** indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses		Reset value	
			MSB	LSB	Hex	Binary
WFEED1	Watchdog feed 1	C2H				
WFEED2	Watchdog feed 2	C3H				

[1] All ports are in input only (high-impedance) state after power-up.

[2] BRGR1 and BRGR0 must only be written if BRGEN in BRGCON SFR is logic 0. If any are written while BRGEN = 1, the result is unpredictable.

[3] The RSTSRC register reflects the cause of the P89LPC904 reset. Upon a power-up reset, all reset source flags are cleared except POF and BOF; the power-on reset value is xx110000.

[4] After reset, the value is 111001x1, i.e., PRE2-PRE0 are all logic 1, WDRUN = 1 and WDCLK = 1. WDTOF bit is logic 1 after Watchdog reset and is logic 0 after power-on reset. Other resets will not affect WDTOF.

[5] On power-on reset, the TRIM SFR is initialized with a factory preprogrammed value. Other resets will not cause initialization of the TRIM register.

[6] The only reset source that affects these SFRs is power-on reset.

1.3 Memory organization

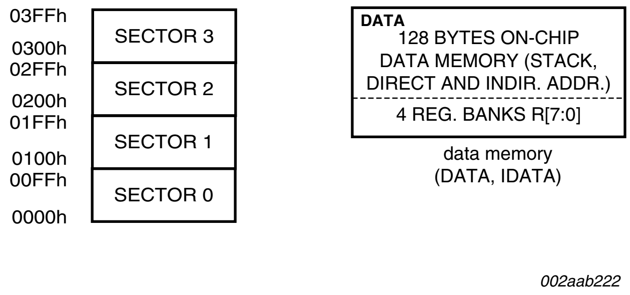


Fig 3. P89LPC904 memory map.

The various P89LPC904 memory spaces are as follows:

DATA — 128 bytes of internal data memory space (00h:7Fh) accessed via direct or indirect addressing, using instruction other than MOVX and MOVC. All or part of the Stack may be in this area.

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

CODE — 1 kB of Code memory space, accessed as part of program execution and via the MOVC instruction.

2. Clocks

2.1 Enhanced CPU

The P89LPC904 uses an enhanced 80C51 CPU which runs at six 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.

2.2 Clock definitions

The P89LPC904 device has several internal clocks as defined below:

OSCCLK — Input to the DIVM clock divider. OSCCLK is selected from one of four clock sources and can also be optionally divided to a slower frequency (see Figure 4 and Section 2.7 "CPU Clock (CCLK) modification: DIVM register"). **Note:** f_{osc} is defined as the OSCCLK frequency.

CCLK — CPU clock; output of the DIVM clock divider. There are two CCLK cycles per machine cycle, and most instructions are executed in one to two machine cycles (two or four CCLK cycles).

RCCLK — The internal 7.373 MHz RC oscillator output.

PCLK — Clock for the various peripheral devices and is CCLK/2.

2.2.1 Oscillator Clock (OSCCLK)

The P89LPC904 provides several user-selectable clock options. These options are configured when the FLASH is programmed and include an on-chip Watchdog oscillator, an on-chip RC oscillator, or an external clock source.

2.3 On-chip RC oscillator option

The P89LPC904 has a 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 data sheet 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.

Table 3: On-chip RC oscillator trim register (TRIM - address 96h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	RCCLK	-	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0
Reset	0	0	Bits 5:0 loaded with factory stored value during reset.					

Table 4: On-chip RC oscillator trim register (TRIM - address 96h) bit description

Bit	Symbol	Description
0	TRIM.0	Trim value. Determines the frequency of the internal RC oscillator. During reset, these bits are loaded with a stored factory calibration value. When writing to either bit 6 or bit 7 of this register, care should be taken to preserve the current TRIM value by reading this register, modifying bits 6 or 7 as required, and writing the result to this register.
1	TRIM.1	
2	TRIM.2	
3	TRIM.3	
4	TRIM.4	
5	TRIM.5	
6	-	Reserved.
7	RCCLK	when = 1, selects the RC Oscillator output as the CPU clock (CCLK)

2.4 Watchdog oscillator option

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

2.5 External clock input option

In this configuration, the processor clock is derived from an external source driving the P0.5/CMPREF/CLKIN pin. The rate may be from 0 Hz up to 18 MHz. **When using an external clock frequency above 12 MHz, the reset input function of P1.5 must be enabled. An external circuit is required to hold the device in reset at power-up until V_{DD} has reached its specified level. When system power is removed V_{DD} will fall below the minimum specified operating voltage. When using an external clock frequency above 12 MHz, in some applications, an external brownout detect circuit may be required to hold the device in reset when V_{DD} falls below the minimum specified operating voltage.**

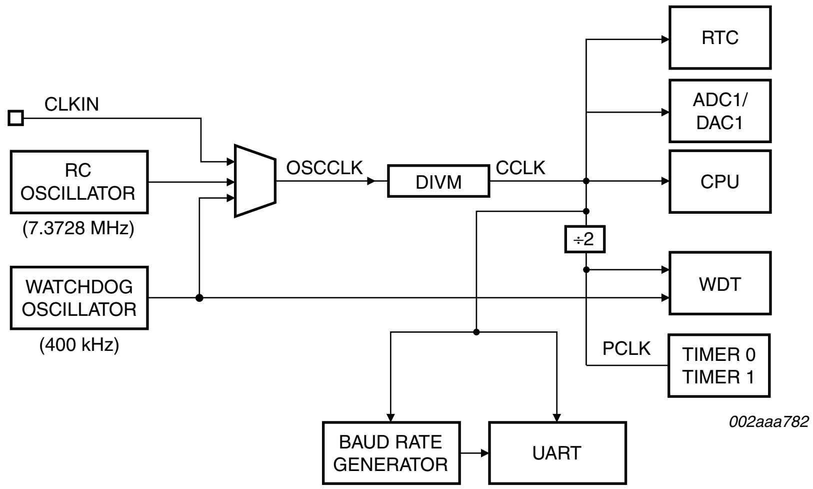


Fig 4. Block diagram of oscillator control.

2.6 Oscillator Clock (OSCCLK) wake-up delay

The P89LPC904 has an internal wake-up timer that delays the clock for 224 OSCCLK cycles plus 60 µs to 100 µs.

2.7 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.

2.8 Low power select

The P89LPC904 is designed to run at 18 MHz (CCLK) maximum. However, if CCLK is 8 MHz or slower, the CLKLP SFR bit (AUXR1.7) can be set to a logic 1 to lower the power consumption further. On any reset, CLKLP is logic 0 allowing highest performance. This bit can then be set in software if CCLK is running at 8 MHz or slower.

3. A/D converter

The P89LPC904 has an 8-bit, 2-channel, multiplexed successive approximation analog-to-digital converter module (ADC1) and one DAC module (DAC1). A block diagram of the A/D converter is shown in Figure 5. The A/D consists of a 2-input multiplexer which feeds a sample and hold circuit providing an input signal to one of two comparator inputs. The control logic in combination with the successive approximation register (SAR) drives a digital-to-analog converter which provides the other input to the comparator. The output of the comparator is fed to the SAR.

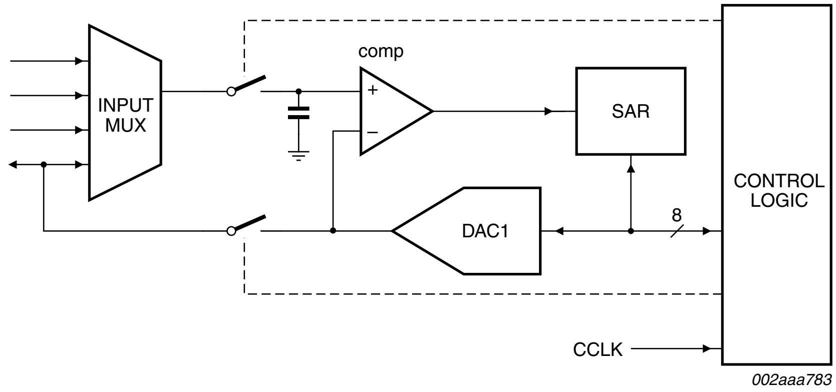


Fig 5. A/D converter block diagram.

3.1 Features

- An 8-bit, 2-channel, multiplexed input, successive approximation A/D converter
- Four A/D result registers
- Six operating modes
  - Fixed channel, single conversion mode
  - Fixed channel, continuous conversion mode
  - Auto scan, single conversion mode
  - Auto scan, continuous conversion mode
  - Dual channel, continuous conversion mode
  - Single step mode
- Two conversion start modes
  - Timer triggered start
  - Start immediately
- 8-bit conversion time of ≥ 3.9 μs at an ADC clock of 3.3 MHz
- Interrupt or polled operation
- Boundary limits interrupt
- DAC output to a port pin with high output impedance
- Clock divider
- Power-down mode

3.2 A/D operating modes

3.2.1 Fixed channel, single conversion mode

A single input channel can be selected for conversion. A single conversion will be performed and the result placed in the result register which corresponds to the selected input channel (See Table 5). An interrupt, if enabled, will be generated after the conversion completes. The input channel is selected in the ADINS register. This mode is selected by setting the SCAN1 bit in the ADMODA register.

Table 5: Input channels and Result registers for fixed channel single, auto scan single, and autoscan continuous conversion modes.

Result register	Input channel	Result register	Input channel
AD1DAT1	AD11	AD1DAT3	AD13

3.2.2 Fixed channel, continuous conversion mode

A single input channel can be selected for continuous conversion. The results of the conversions will be sequentially placed in the four result registers Table 6. An interrupt, if enabled, will be generated after every four conversions. Additional conversion results will again cycle through the four result registers, overwriting the previous results. Continuous conversions continue until terminated by the user. This mode is selected by setting the SCC1 bit in the ADMODA register.

Table 6: Result registers and conversion results for fixed channel, continuous conversion mode.

Result register	Contains
AD1DAT0	Selected channel, first conversion result
AD1DAT1	Selected channel, second conversion result
AD1DAT2	Selected channel, third conversion result
AD1DAT3	Selected channel, fourth conversion result

3.2.3 Auto scan, single conversion mode

Any combination of the two input channels can be selected for conversion by setting a channel’s respective bit in the ADINS register. The channels are converted from LSB to MSB order (in ADINS). A single conversion of each selected input will be performed and the result placed in the result register which corresponds to the selected input channel (See Table 5). An interrupt, if enabled, will be generated after all selected channels have been converted. If only a single channel is selected this is equivalent to single channel, single conversion mode. This mode is selected by setting the SCAN1 bit in the ADMODA register.

3.2.4 Auto scan, continuous conversion mode

Any combination of the two input channels can be selected for conversion by setting a channel’s respective bit in the ADINS register. The channels are converted from LSB to MSB order (in ADINS). A conversion of each selected input will be performed and the result placed in the result register which corresponds to the selected input channel (See Table 5). An interrupt, if enabled, will be generated after all selected channels have been converted. The process will repeat starting with the first selected channel. Additional conversion results will again cycle through the result registers of the selected channels, overwriting the previous results. Continuous conversions continue until terminated by the user. This mode is selected by setting the BURST1 bit in the ADMODA register.

3.2.5 Dual channel, continuous conversion mode

Any combination of two of the four input channels can be selected for conversion. The result of the conversion of the first channel is placed in the first result register. The result of the conversion of the second channel is placed in the second result register. The first channel is again converted and its result stored in the third result register. The second channel is again converted and its result placed in the fourth result register (See Table 7). An interrupt is generated, if enabled, after every set of four conversions (two conversions per channel). This mode is selected by setting the SCC1 bit in the ADMODA register.

Table 7: Result registers and conversion results for dual channel, continuous conversion mode.

Result register	Contains
AD1DAT0	First channel, first conversion result
AD1DAT1	Second channel, first conversion result
AD1DAT2	First channel, second conversion result
AD1DAT3	Second channel, second conversion result

3.2.6 Single step

This special mode allows ‘single-stepping’ in an auto scan conversion mode. Any combination of the four input channels can be selected for conversion. After each channel is converted, an interrupt is generated, if enabled, and the A/D waits for the next start condition. The result of each channel is placed in the result register which corresponds to the selected input channel (See Table 5). May be used with any of the start modes. This mode is selected by clearing the BURST1, SCC1, and SCAN1 bits in the ADMODA register.

3.2.7 Conversion mode selection bits

The A/D uses three bits in ADMODA to select the conversion mode. These mode bits are summarized in Table 8, below. Combinations of the three bits, other than the combinations shown, are undefined.

Table 8: Conversion mode bits.

BURST1	SCC1	Scan1	ADC1 conversion mode
0	0	0	single step
0	0	1	fixed channel,single
			auto scan, single
0	1	0	fixed channel, continuous
			dual channel, continuous
1	0	0	auto scan, continuous

3.3 Trigger modes

3.3.1 Timer triggered start

An A/D conversion is started by the overflow of Timer 0. Once a conversion has started, additional Timer 0 triggers are ignored until the conversion has completed. The Timer triggered start mode is available in all A/D operating modes. This mode is selected by the TMM1 bit and the ADCS11 and ADCS10 bits (See Table 10).

3.3.2 Start immediately

Programming this mode immediately starts a conversion. This start mode is available in all A/D operating modes. This mode is selected by setting the ADCS11 and ADCS10 bits in the ADCON1 register (See Table 10).

3.3.3 Boundary limits interrupt

The A/D converter has both a HIGH and LOW boundary limit register. After the four MSBs have been converted, these four bits are compared with the four MSBs of the boundary HIGH and LOW registers. If the four MSBs of the conversion are outside the limit an interrupt will be generated, if enabled. If the conversion result is within the limits, the boundary limits will again be compared after all eight bits have been converted. An interrupt will be generated, if enabled, if the result is outside the boundary limits. The boundary limit may be disabled by clearing the boundary limit interrupt enable.

3.4 DAC output to a port pin with high-impedance

The AD1DAT3 register is used to hold the value fed to the DAC. After a value has been written to AD1DAT3 the DAC output will appear on the DAC1 pin. The DAC1 output is enabled by the ENDAC1 bit in the ADMODB register (See Table 14).

3.5 Clock divider

The A/D converter requires that its internal clock source be in the range of 500 kHz to 3.3 MHz to maintain accuracy. A programmable clock divider that divides the clock from 1 to 8 is provided for this purpose (See Table 14).

3.6 I/O pins used with ADC functions

The analog input pins maybe be used as either digital I/O or as inputs to A/D and thus have a digital input and output function. In order to give the best analog performance, pins that are being used with the ADC should have their digital outputs and inputs disabled and have the 5V tolerance disconnected. Digital outputs are disabled by putting the port pins into the input-only mode as described in the Port Configurations section (see Table 20). Digital inputs will be disconnected automatically from these pins when the pin has been selected by setting its corresponding bit in the ADINS register and its corresponding A/D has been enabled

When used as digital I/O these pins are 5 V tolerant. If selected as input signals in ADINS, these pins will be 3V tolerant if the corresponding A/D is enabled and the device is not in power down. Otherwise the pin will remain 5V tolerant. Please refer to the *P89LPC904 data sheet* for specifications.

3.7 Power-down and idle mode

In idle mode the A/D converter, if enabled, will continue to function and can cause the device to exit idle mode when the conversion is completed if the A/D interrupt is enabled. In Power-down mode or Total Power-down mode, the A/D does not function. If the A/D is enabled, it will consume power. Power can be reduced by disabling the A/D.

Table 9: A/D Control register 1 (ADCON1 - address 97h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	ENBI1	ENADCI1	TMM1	-	ADCI1	ENADC1	ADCS11	ADCS10
Reset	0	0	0	0	0	0	0	0

Table 10: A/D Control register 1 (ADCON1 - address 97h) bit description

Bit	Symbol	Description
0	ADCS10	A/D start mode bits [11:10]:
1	ADCS11	**00 —** Timer Trigger Mode when TMM1 = 1. Conversions starts on overflow of Timer 0. Stop mode when TMM1 = 0, no start occurs. **01 —** Immediate Start Mode. Conversions starts immediately.
2	ENADC1	Enable A/D channel 1. When set = 1, enables ADC1. Must also be set for D/A operation of this channel.
3	ADCI1	A/D Conversion complete Interrupt 1. Set when any conversion or set of multiple conversions has completed. Cleared by software.
4	-	reserved
5	TMM1	Timer Trigger Mode 1. Selects either stop mode (TMM1 = 0) or timer trigger mode (TMM1 = 1) when the ADCS11 and ADCS10 bits = 00.
6	ENADCI1	Enable A/D Conversion complete Interrupt 1. When set, will cause an interrupt if the ADCI1 flag is set and the A/D interrupt is enabled.
7	ENBI1	Enable A/D boundary interrupt 1. When set, will cause an interrupt if the boundary interrupt 1 flag, BNDI1, is set and the A/D interrupt is enabled.

Table 11: A/D Mode Register A (ADMODA - address C0h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	BNBI1	BURST1	SCC1	SCAN1	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Table 12: A/D Mode Register A (ADMODA - address C0h) bit description

Bit	Symbol	Description
0:3	-	reserved
4	SCAN1	when = 1, selects single conversion mode (auto scan or fixed channel) for ADC1
5	SCC1	when = 1, selects fixed channel, continuous conversion mode for ADC1
6	BURST1	when = 1, selects auto scan, continuous conversion mode for ADC1
7	BNBI1	ADC1 boundary interrupt flag. When set, indicates that the converted result from ADC1 is outside of the range defined by the ADC1 boundary registers

Table 13: A/D Mode Register B (ADMODB - address A1h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	CLK2	CLK1	CLK0	-	ENDAC1	-	BSA1	-
Reset	0	0	0	0	0	0	0	0

Table 14: A/D Mode Register B (ADMODB - address A1h) bit description

Bit	Symbol	Description
0	-	reserved
1	BSA1	ADC1 Boundary Select All. When = 1, BNDI1 will be set if any ADC1 input exceeds the boundary limits. When = 0, BNDI1 will be set only if the AD10 input exceeded the boundary limits.
2	-	reserved
3	ENDAC1	When = 1 selects DAC mode for ADC1; when = 0 selects ADC mode.
4	-	reserved

Table 14: A/D Mode Register B (ADMODB - address A1h) bit description

Bit	Symbol	Description
5	CLK0	Clock divider to produce the ADC clock. Divides CCLK by the value indicated below. The resulting ADC clock should be 3.3 MHz or less. A minimum of 0.5 MHz is required to maintain A/D accuracy. A/D start mode bits: **CLK2:0 — divisor** **000** — 1 **001** — 2 **010** — 3 **011** — 4 **100** — 5 **101** — 6 **110** — 7 **111** — 8
6	CLK1	
7	CLK2	

Table 15: A/D Input Select register (ADINS - address A3h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	AIN13	-	AIN11	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Table 16: A/D Input Select register (ADINS - address A3h) bit description

Bit	Symbol	Description
0:4	-	reserved
5	AIN11	when set, enables the AD11 pin for sampling and conversion
6	-	reserved
7	AIN13	when set, enables the AD13 pin for sampling and conversion

4. Interrupts

The P89LPC904 uses a four priority level interrupt structure. This allows great flexibility in controlling the handling of the P89LPC904’s 10 interrupt sources.

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 received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are pending at the start of an instruction cycle, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used for pending requests of the same priority level. Table 18 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.

4.1 Interrupt priority structure

Table 17: 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

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 17.

Table 18: Summary of interrupts

Description	Interrupt flag bit(s)	Vector address	Interrupt enable bit(s)	Interrupt priority	Arbitration ranking	Power-down wake-up
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	TI and RI	0023h	ES/ESR (IEN0.4)	IP0H.4,IP0.4	13	No
Serial port Rx	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)	IP0H.0,IP0.0	8	Yes
Comparator 1	CMF1	0043h	EC (IEN1.2)	IP0H.0,IP0.0	11	Yes
Serial port Tx	TI	006Bh	EST (IEN1.6)	IP0H.0,IP0.0	12	No
ADC	ADCI1,BNDI1	0073h	EAD (IEN1.7)	IP1H.7,IP1.7	15 (lowest)	No

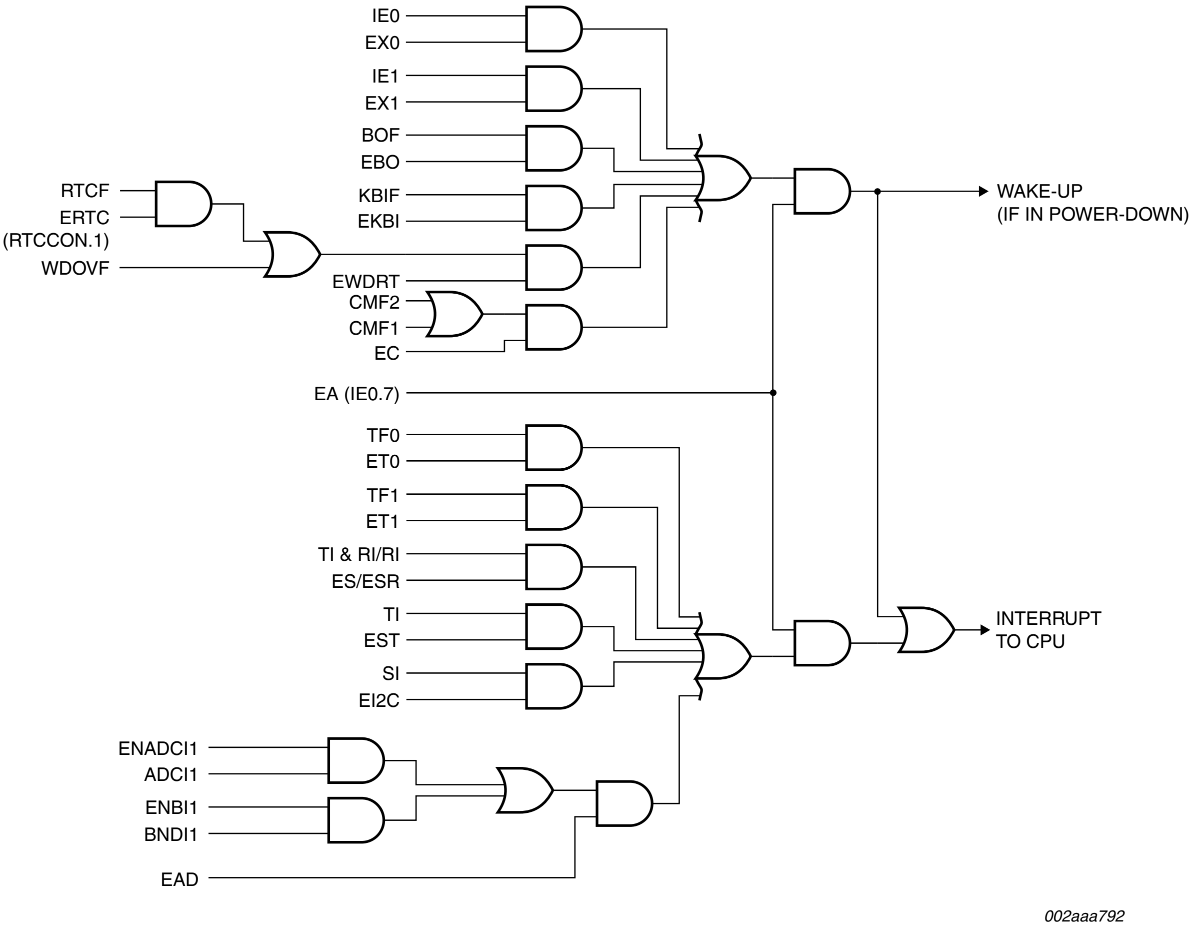


Fig 6. Interrupt sources, interrupt enables, and power-down wake up sources.

5. I/O ports

The P89LPC904 has two I/O ports: Port 0, and Port 1. The exact number of I/O pins available depends upon the clock and reset option chosen (see Table 19).

Table 19: Number of I/O pins available

Clock source	Reset option	Number of I/O pins
On-chip oscillator or Watchdog oscillator	No external reset (except during power up)	6
	External $\overline{\text{RST}}$ pin supported	5
External clock input	No external reset (except during power up)	5
	External $\overline{\text{RST}}$ pin supported[1]	4

[1] Required for operation above 12 MHz.

5.1 Port configurations

All but three I/O port pins on the P89LPC904 may be configured by software to one of four types on a pin-by-pin basis, as shown in Table 20. 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 ($\overline{\text{RST}}$) can only be an input and cannot be configured.

Table 20: 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

5.2 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 pull-up transistors in the quasi-bidirectional output that serve different purposes.

One of these pull-ups, called the ‘very weak’ pull-up, is turned on whenever the port latch for the pin contains a logic 1. This very weak pull-up sources a very small current that will pull the pin HIGH if it is left floating.

A second pull-up, called the ‘weak’ pull-up, 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 pull-up 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, the weak pull-up turns off, and only the very weak pull-up remains on. In order to pull the pin LOW under these conditions, the external device has to sink enough current to overpower the weak pull-up and pull the port pin below its input threshold voltage.

The third pull-up is referred to as the ‘strong’ pull-up. This pull-up 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 pull-up turns on for two CPU clocks quickly pulling the port pin HIGH.

The quasi-bidirectional port configuration is shown in Figure 7.

Although the P89LPC904 is a 3 V device most of the pins are 5 V-tolerant. If 5 V 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 5 V 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 *P89LPC904 data sheet, Dynamic characteristics* for glitch filter specifications).

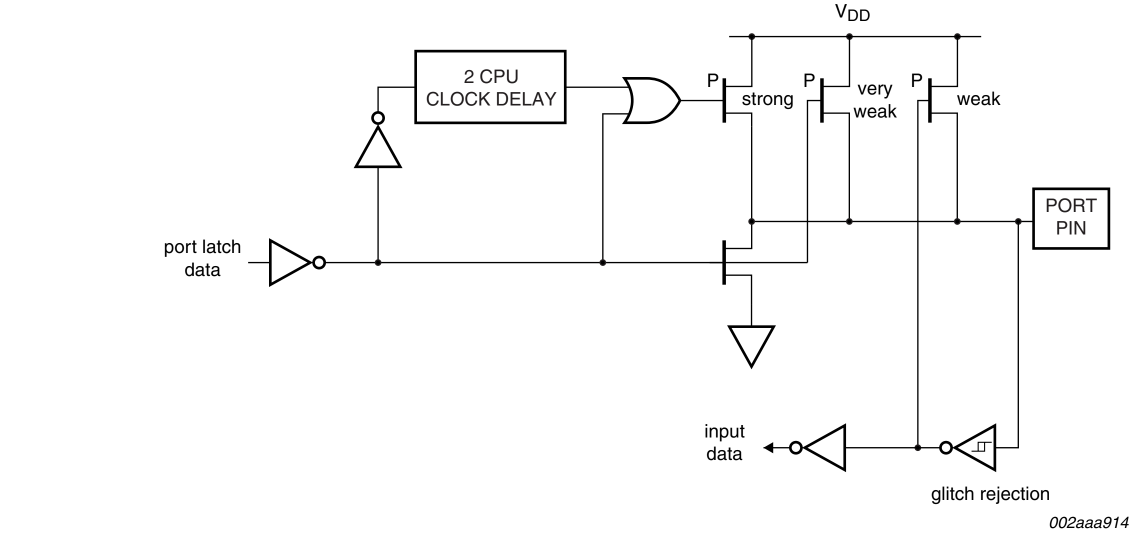


Fig 7. Quasi-bidirectional output.

5.3 Open drain output configuration

The open drain output configuration turns off all pull-ups and only drives the pull-down 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 pull-up, typically a resistor tied to V_{DD}. The pull-down for this mode is the same as for the quasi-bidirectional mode.

The open drain port configuration is shown in Figure 8.

An open drain port pin has a Schmitt triggered input that also has a glitch suppression circuit.

Please refer to the *P89LPC904 data sheet, Dynamic characteristics* for glitch filter specifications.

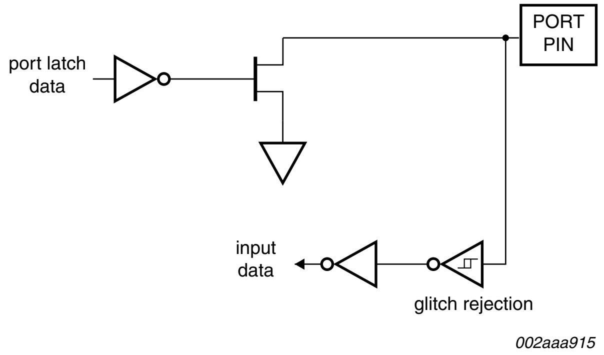


Fig 8. Open drain output.

5.4 Input-only configuration

The input port configuration is shown in Figure 9. It is a Schmitt triggered input that also has a glitch suppression circuit.

(Please refer to the *P89LPC904 data sheet, Dynamic characteristics* for glitch filter specifications).

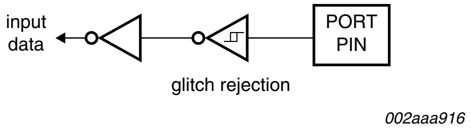


Fig 9. Input only.

5.5 Push-pull output configuration

The push-pull output configuration has the same pull-down structure as both the open drain and the quasi-bidirectional output modes, but provides a continuous strong pull-up 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 10.

A push-pull port pin has a Schmitt triggered input that also has a glitch suppression circuit.

(Please refer to the *P89LPC904 data sheet, Dynamic characteristics* for glitch filter specifications).

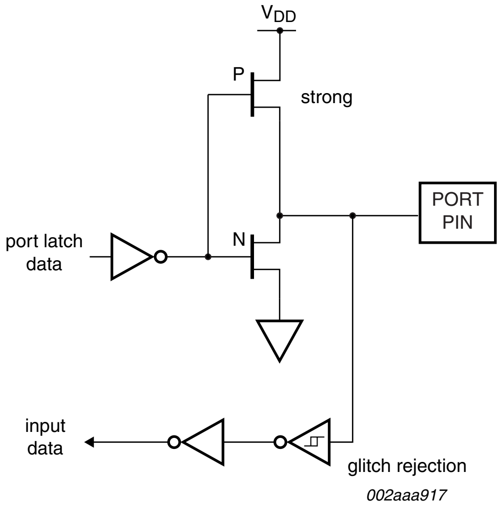


Fig 10. Push-pull output.

5.6 Port 0 analog functions

The P89LPC904 incorporates two Analog Comparators. 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 Figure 9).

Digital inputs on Port 0 may be disabled through the use of the PT0AD register. Setting the corresponding bit in PT0AD disables that pin's digital input. Port bits that have their digital inputs disabled will be read as logic 0 by any instruction that accesses the port.

On any reset, PT0AD bits default to logic 0s to enable the digital functions.

5.7 Additional port features

After power-up, all pins are in Input-Only mode. **Please note that this is different from most 80C51 family 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 P89LPC904 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 *P89LPC904 data sheet* 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.

Table 21: Port output configuration

Port pin	Configuration SFR bits			
	PxM1.y	PxM2.y	Alternate usage	Notes
P0.2	P0M1.2	P0M2.2	KBI2, CIN2A, AD11	Refer to section section "Port 0 analog functions" on page 25 for use as analog inputs.
P0.4	P0M1.4	P0M2.4	KBI4, CIN1A, AD13, DAC1	
P0.5	P0M1.5	P0M2.5	KBI5, CMPREF	
P1.0	P1M1.0	P1M2.0	TxD	
P1.1	P1M1.1	P1M2.1	RxD	
P1.5	P1M1.5	P1M2.5	$\overline{\text{RST}}$	

6. Power monitoring functions

The P89LPC904 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.

6.1 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 logic 0, indicating brownout detection is enabled on power-on if BOE is programmed.

If Brownout Detection is enabled, the brownout condition occurs when V_{DD} falls below the Brownout trip voltage, V_{BO} (see *P89LPC904 Static Characteristics*), and is negated when V_{DD} rises above V_{BO}. If the P89LPC904 device is to operate with a power supply that can be below 2.7 V, BOE should be left in the unprogrammed state so that the device can operate at 2.4 V, otherwise continuous brownout reset may prevent the device from operating.

If Brownout Detect is enabled (BOE programmed, PMOD1-0 ≠ ‘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 logic 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 data sheet for specifications.

Table 22: 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 2.4 V to 3.6 V.
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 2.4 V to 3.6 V. However, BOPD is default to logic 0 upon power-up.
		0 (brownout detect active)	0 (brownout detect generates reset)	X	X	Brownout reset enabled. V_{DD} operating range is 2.7 V to 3.6 V. Upon a brownout reset, BOF (RSTSRC.5) will be set to indicate the reset source. BOF can be cleared by writing logic 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 2.7 V to 3.6 V. Upon a brownout interrupt, BOF (RSTSRC.5) will be set. BOF can be cleared by writing logic 0 to the bit.
				0	X	Both brownout reset and interrupt disabled. V_{DD} operating range is 2.4 V to 3.6 V. However, BOF (RSTSRC.5) will be set when V_{DD} falls to the Brownout Detection trip point. BOF can be cleared by writing logic 0 to the bit.
				X	0	

[1] Cannot be used with operation above 12 MHz as this requires V_{DD} of 3.0 V or above.

6.2 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 logic 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.

6.3 Power reduction modes

The P89LPC904 supports three different power reduction modes as determined by SFR bits PCON.1-0 (see Table 23).

Table 23: 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 P89LPC904 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 VRAM. 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 VRAM, 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 begin execution after 256 CPU clocks. 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 logic 1 • Comparators (Note: Comparators can be powered down separately with PCONA.5 set to logic 1 and comparators disabled) • Real-time Clock/System Timer (and the crystal oscillator circuitry if this block is using it, unless RTCPD, i.e., PCONA.7 is logic 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 wake-up 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 wake-up options supported: • Watchdog timer if WDCLK (WDCON.0) is logic 1. Could generate Interrupt or Reset, either one can wake up the device • Keyboard Interrupt • Real-time Clock/System Timer (unless RTCPD, i.e., PCONA.7 is logic 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.

Table 24: Power Control register (PCON - address 87h) bit allocation

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

Table 25: Power Control register (PCON - address 87h) bit description

Bit	Symbol	Description
0	PMOD0	Power Reduction Mode (see Section 6.3)
1	PMOD1	
2	GF0	General Purpose Flag 0. May be read or written by user software, but has no effect on operation
3	GF1	General Purpose Flag 1. May be read or written by user software, but has no effect on operation
4	BOI	Brownout Detect Interrupt Enable. When logic 1, Brownout Detection will generate a interrupt. When logic 0, Brownout Detection will cause a reset
5	BOPD	Brownout Detect power-down. When logic 1, Brownout Detect is powered down and therefore disabled. When logic 0, Brownout Detect is enabled. (Note: BOPD must be logic 0 before any programming or erasing commands can be issued. Otherwise these commands will be aborted.)
6	SMOD0	Framing Error Location: • When logic 0, bit 7 of SCON is accessed as SM0 for the UART • When logic 1, bit 7 of SCON is accessed as the framing error status (FE) for the UART
7	SMOD1	Double Baud Rate bit for the serial port (UART) when Timer 1 is used as the baud rate source. When logic 1, the Timer 1 overflow rate is supplied to the UART. When logic 0, the Timer 1 overflow rate is divided by two before being supplied to the UART. (See Section 10)

Table 26: Power Control register A (PCONA - address B5h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	RTCPD	-	VCPD	-	I2PD	-	SPD	-
Reset	0	0	0	0	0	0	0	0

Table 27: Power Control register A (PCONA - address B5h) bit description

Bit	Symbol	Description
0	-	reserved
1	SPD	Serial Port (UART) power-down: When logic 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.
2	-	reserved
3	I2PD	I^{2}C power-down: When logic 1, the internal clock to the I^{2}C-bus is disabled. Note that in either Power-down mode or Total Power-down mode, the I^{2}C clock will be disabled regardless of this bit.
4	-	reserved
5	VCPD	Analog Voltage Comparators power-down: When logic 1, the voltage comparators are powered down. User must disable the voltage comparators prior to setting this bit.
6	-	reserved
7	RTCPD	Real-time Clock power-down: When logic 1, the internal clock to the Real-time Clock is disabled.

7. 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 overridden 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 *P89LPC904 data sheet, Static characteristics*) before power is reapplied, in order to ensure a power-on reset.

Note: When using an oscillator frequency above 12 MHz, the reset input function of P1.5 must be enabled. An external circuit is required to hold the device in reset at power-up until V_{DD} has reached its specified level. When system power is removed V_{DD} will fall below the minimum specified operating voltage. When using an oscillator frequency above 12 MHz, in some applications, an external brownout detect circuit may be required to hold the device in reset when V_{DD} falls below the minimum specified operating voltage.

Reset can be triggered from the following sources (see Figure 11):

- External reset pin (during power-on or if user configured via UCFG1)
- Power-on Detect
- Brownout Detect
- Watchdog timer
- Software reset
- UART break detect reset

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 logic 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.

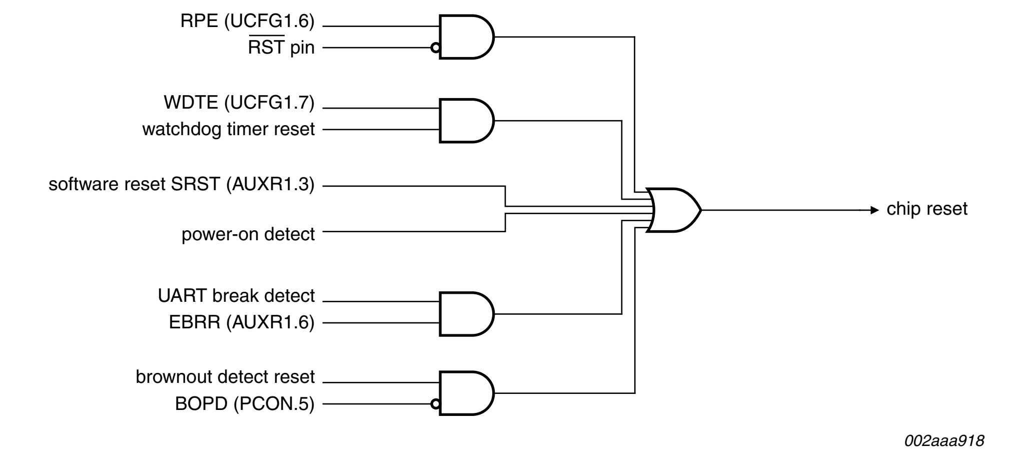


Fig 11. Block diagram of reset.

Table 28: Reset Sources register (RSTSRC - address DFh) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX
Reset[1]	x	x	1	1	0	0	0	0

[1] The value shown is for a power-on reset. Other reset sources will set their corresponding bits.

Table 29: Reset Sources register (RSTSRC - address DFh) bit description

Bit	Symbol	Description
0	R_EX	external reset Flag. When this bit is logic 1, it indicates external pin reset. Cleared by software by writing a logic 0 to the bit or a Power-on reset. If RST is still asserted after the Power-on reset is over, R_EX will be set.
1	R_SF	software reset Flag. Cleared by software by writing a logic 0 to the bit or a Power-on reset
2	R_WD	Watchdog timer reset flag. Cleared by software by writing a logic 0 to the bit or a Power-on reset.(NOTE: UCFG1.7 must be = 1)
3	R_BK	break detect reset. If a break detect occurs and EBRR (AUXR1.6) is set to logic 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 logic 0 to the bit or on a Power-on reset.
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 logic 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.)
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 logic 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.)
6:7	-	reserved

7.1 Reset vector

Following reset, the P89LPC904 will fetch instructions from either address 0000h or the Boot address. The Boot address is formed by using the Boot Vector as the HIGH byte of the address and the LOW byte of the address = 00h. The Boot address will be used if a

UART break reset occurs or the non-volatile Boot Status bit (BOOTSTAT.0) = 1, or the device has been forced into ISP mode. Otherwise, instructions will be fetched from address 0000H.

8. Timers 0 and 1

The P89LPC904 has two general-purpose timers which are compatible with the 80C51 Timer 0 and Timer 1, and are incremented every PCLK when running.

Timer 0 and Timer 1 have four operating modes (modes 0, 1, 2, and 3), which are selected by bit-pairs (TnM1, TnM0) in TMOD. Modes 0, 1, and 2 are the same for both Timers. Mode 3 is different. The operating modes are described later in this section.

Table 30: Timer Mode register (TMOD - address 89h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	T1M1	T1M0	-	-	T0M1	T0M0
Reset	0	0	0	0	0	0	0	0

Table 31: Timer Mode register (TMOD - address 89h) bit description

Bit	Symbol	Description
0	T0M0	Mode Select for Timer 0.
1	T0M1	
		The following timer modes are selected by timer mode bits T0M1.T0M0: **00** — 8048, Timer0 serves as a 5-bit prescaler. (Mode 0). **01** — 16-bit Timer, TH0 and TL0 are cascaded; there is no prescaler.(Mode 1). **10** — 8-bit auto-reload Timer. TH0 holds a value which is loaded into TL0 when it overflows. (Mode 2). **11** — 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).
2:3	-	reserved
4	T1M0	Mode Select for Timer 1.
5	T1M1	
		The following timer modes are selected by timer mode bits T1M1,T1M0: **00** — 8048 Timer1, TL1 serves as 5-bit prescaler. (Mode 0). **01** — 16-bit Timer, TH1 and TL1 are cascaded; there is no prescaler.(Mode 1). **10** — 8-bit auto-reload Timer. TH1 holds a value which is loaded into TL1 when it overflows. (Mode 2). **11** — 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).
6:7	-	reserved

8.1 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 12 shows Mode 0 operation.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all logic 1s to all logic 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 (Table 33).

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 12.

8.2 Mode 1

Mode 1 is the same as Mode 0, except that all 16 bits of the timer register (THn and TLn) are used. See Figure 13.

8.3 Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TLn) with automatic reload, as shown in Figure 14. 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.

8.4 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 15. TL0 uses the Timer 0 control bits: TR0 and TF0. TH0 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. With Timer 0 in Mode 3, an P89LPC904 device can look like it has three Timer/Counters.

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, or in any application not requiring an interrupt.

Table 32: Timer/Counter Control register (TCON) - address 88h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Reset	0	0	0	0	0	0	0	0

Table 33: Timer/Counter Control register (TCON - address 88h) bit description

Bit	Symbol	Description
0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/LOW level triggered external interrupts.
1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt 0 edge is detected. Cleared by hardware when the interrupt is processed, or by software.
2	IT1	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/LOW level triggered external interrupts.

Table 33: Timer/Counter Control register (TCON - address 88h) bit description

Bit	Symbol	Description
3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt 1 edge is detected. Cleared by hardware when the interrupt is processed, or by software.
4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
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, where it is cleared in hardware)
6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off
7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the interrupt is processed, or by software (except in mode 6, see above, when it is cleared in hardware).

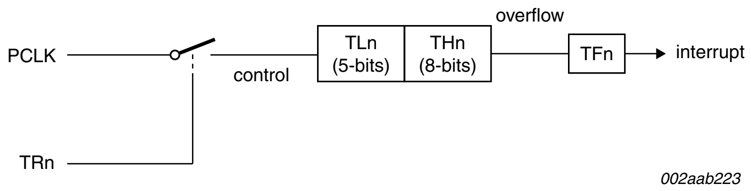


Fig 12. Timer/counter 0 or 1 in Mode 0 (13-bit counter).

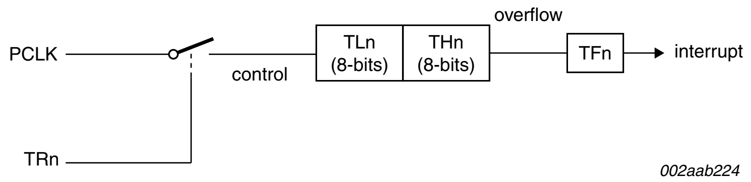


Fig 13. Timer/counter 0 or 1 in mode 1 (16-bit counter).

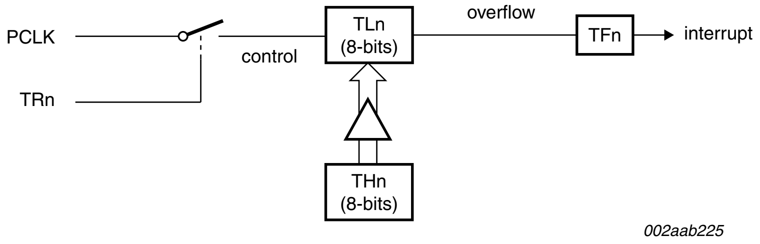


Fig 14. Timer/counter 0 or 1 in Mode 2 (8-bit auto-reload).

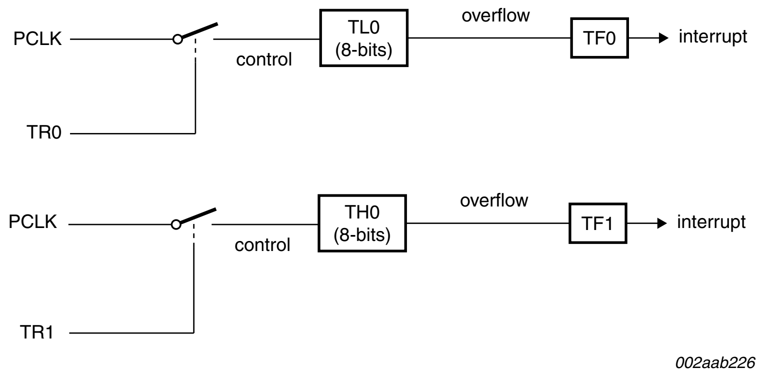


Fig 15. Timer/counter 0 Mode 3 (two 8-bit counters).

9. Real-time clock system timer

The P89LPC904 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 16).

The Real-time Clock is a 23-bit down counter. The clock source for this counter can be either the CPU clock (CCLK) or the XTAL1-2 oscillator, provided that the XTAL1-2 oscillator is not being used as the CPU clock. If the XTAL1-2 oscillator is used as the CPU clock, then the RTC will use CCLK as its clock source regardless of the state of the RTCS1:0 in the RTCCON register. 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 logic 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.

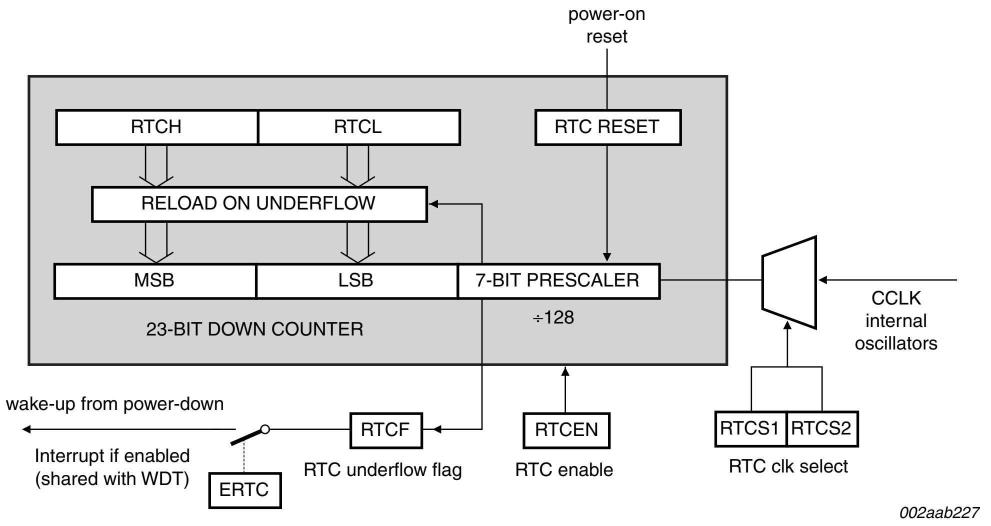


Fig 16. Real-time clock/system timer block diagram.

9.1 Real-time clock source

RTCS1-0 (RTCCON[6:5]) are used to select the clock source for the RTC if either the Internal RC oscillator or the internal WD oscillator is used as the CPU clock. If the internal crystal oscillator or the external clock input on XTAL1 is used as the CPU clock, then the RTC will use CCLK as its clock source.

9.2 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.

9.3 Real-time clock interrupt/wake-up

If ERTC (RTCCON.1), EWDRT (IEN1[6:0]) and EA (IEN0.7) are set to logic 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.

9.4 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.

Table 34: Real-time Clock/System Timer clock sources

FOSC2:0	RCCLK	RTCS1:0	RTC clock source	CPU clock source
000	x	xx	undefined	undefined
001				
010				

Table 34: Real-time Clock/System Timer clock sources

FOSC2:0	RCCLK	RTCS1:0	RTC clock source	CPU clock source
011	0	00	External clock input	Internal RC oscillator /DIVM
		01		
		10		
		11	Internal RC oscillator /DIVM	
	1	00	External clock input	Internal RC oscillator
		01		
		10		
		11	Internal RC oscillator	
100	0	00	External clock input	Watchdog oscillator /DIVM
		01		
		10		
		11	Watchdog oscillator /DIVM	
	1	00	External clock input	Internal RC oscillator
		01		
		10		
		11	Internal RC oscillator	
101	x	xx	undefined	undefined
110				
111	0	00	External clock input /DIVM	External clock input/DIVM
		01		
		10		
		11		External clock input /DIVM
	1	00	External clock input	Internal RC oscillator
		01		
		10		
		11	Internal RC oscillator	

Table 35: Real-time Clock Control register (RTCCON - address D1h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN
Reset	0	1	1	x	x	x	0	0

Table 36: Real-time Clock Control register (RTCCON - address D1h) bit description

Bit	Symbol	Description
0	RTCEN	Real-time Clock enable. The Real-time Clock will be enabled if this bit is logic 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.
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 logic 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.
2:4	-	reserved
5	RTCS0	Real-time Clock source select (see Table 34).
6	RTCS1	
7	RTCF	Real-time Clock Flag. This bit is set to logic 1 when the 23-bit Real-time Clock reaches a count of logic 0. It can be cleared in software.

10. UART

The P89LPC904 has 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 P89LPC904 does include an independent Baud Rate Generator. The baud rate can be selected from the oscillator (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 four modes, as described in the following sections.

10.1 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 CPU clock frequency.

10.2 Mode 1

10 bits are transmitted (through TxD) or received (through RxD): a start bit (logic 0), 8 data bits (LSB first), and a stop bit (logic 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 Section 10.6 “Baud Rate generator and selection”).

10.3 Mode 2

11 bits are transmitted (through TxD) or received (through RxD): start bit (logic 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logic 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.

10.4 Mode 3

11 bits are transmitted (through TxD) or received (through RxD): a start bit (logic 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logic 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 Section 10.6 “Baud Rate generator and selection” on page 40).

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.

10.5 SFR space

The UART SFRs are at the following locations shown in Table 37.

Table 37: UART SFR addresses

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

10.6 Baud Rate generator and selection

The P89LPC904 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 17). Note that Timer T1 is further divided by 2 if the SMOD1 bit (PCON.7) is set. The independent Baud Rate Generator uses CCLK.

10.7 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 logic 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 38: UART baud rate generation.

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)

Table 38: UART baud rate generation.

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)

Table 39: Baud Rate Generator Control register (BRGCON - address BDh) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	--	-	-	-	-	-	SBRGS	BRGEN
Reset	x	x	x	x	x	x	0	0

Table 40: Baud Rate Generator Control register (BRGCON - address BDh) bit description

Bit	Symbol	Description
0	BRGEN	Baud Rate Generator Enable. Enables the baud rate generator. BRGR1 and BRGR0 can only be written when BRGEN = 0.
1	SBRGS	Select Baud Rate Generator as the source for baud rates to UART in modes 1 and 3 (see Table 38 for details)
2:7	-	reserved

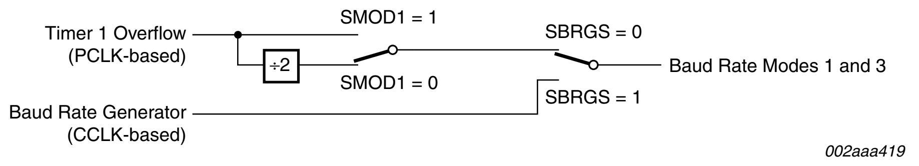


Fig 17. Baud rate generation for UART (Modes 1, 3).

10.8 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 logic 0.

10.9 Break detect

A break detect is reported in the status register (SSTAT). A break is detected when any 11 consecutive bits are sensed LOW. 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 and force the device into ISP mode by setting the EBRR bit (AUXR1.6).

Table 41: Serial Port Control register (SCON - address 98h) bit allocation

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

Table 42: Serial Port Control register (SCON - address 98h) bit description

Bit	Symbol	Description
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.
1	TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the stop bit (see description of INTLO bit in SSTAT register) in the other modes. Must be cleared by software.
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.
3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI 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.
6	SM1	With SM0 defines the serial port mode, see Table 43.
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 logic 0 - default mode on any reset.)

Table 43: Serial Port modes

SM0,SM1	UART mode	UART baud rate
00	Mode 0: shift register	CCLK/16 (default mode on any reset)
01	Mode 1: 8-bit UART	Variable (see Table 38)
10	Mode 2: 9-bit UART	CCLK/32 or CCLK/16
11	Mode 3: 9-bit UART	Variable (see Table 38)

Table 44: Serial Port Status register (SSTAT - address BAh) bit allocation

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

Table 45: Serial Port Status register (SSTAT - address BAh) bit description

Bit	Symbol	Description
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 logic 1.
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.
2	BR	Break Detect flag. A break is detected when any 11 consecutive bits are sensed LOW. Cleared by software.
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.
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 logic 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.
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 logic 0 to select combined interrupts.
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 logic 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.
7	DBMOD	Double buffering mode. When set = 1 enables double buffering. Must be logic 0 for UART mode 0. In order to be compatible with existing 80C51 devices, this bit is reset to logic 0 to disable double buffering.

10.10 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 18.

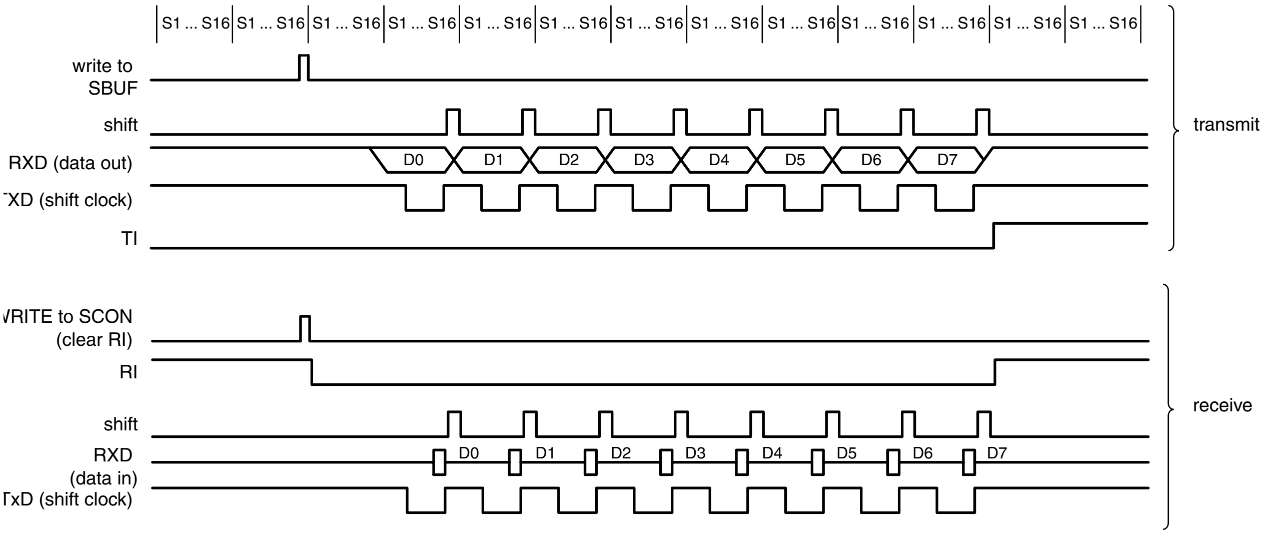


Fig 18. Serial Port Mode 0 (double buffering must be disabled).

10.11 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.

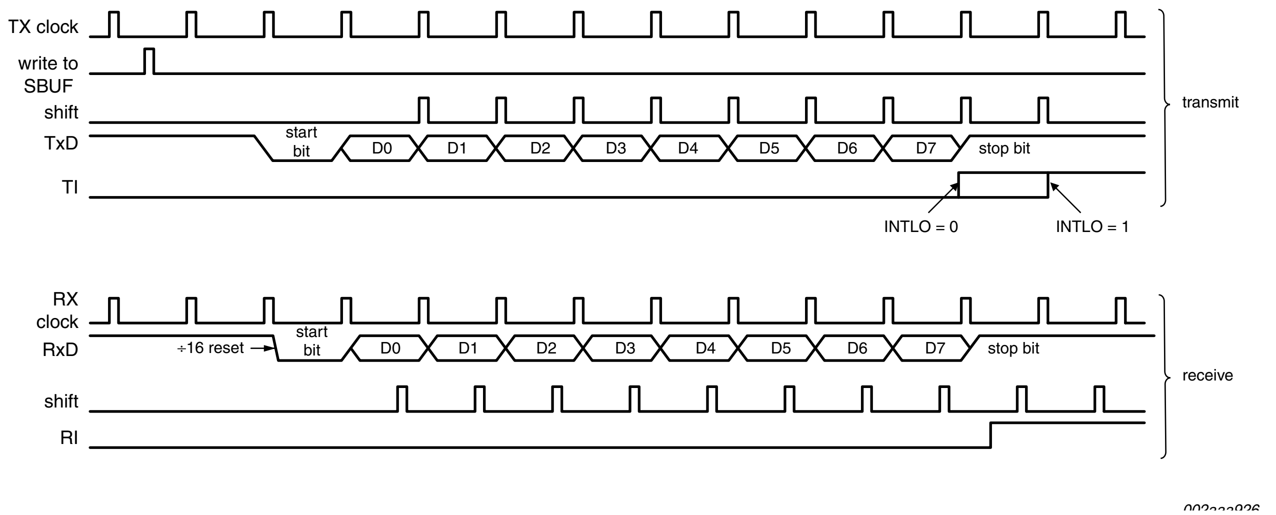


Fig 19. Serial Port Mode 1 (only single transmit buffering case is shown).

10.12 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.

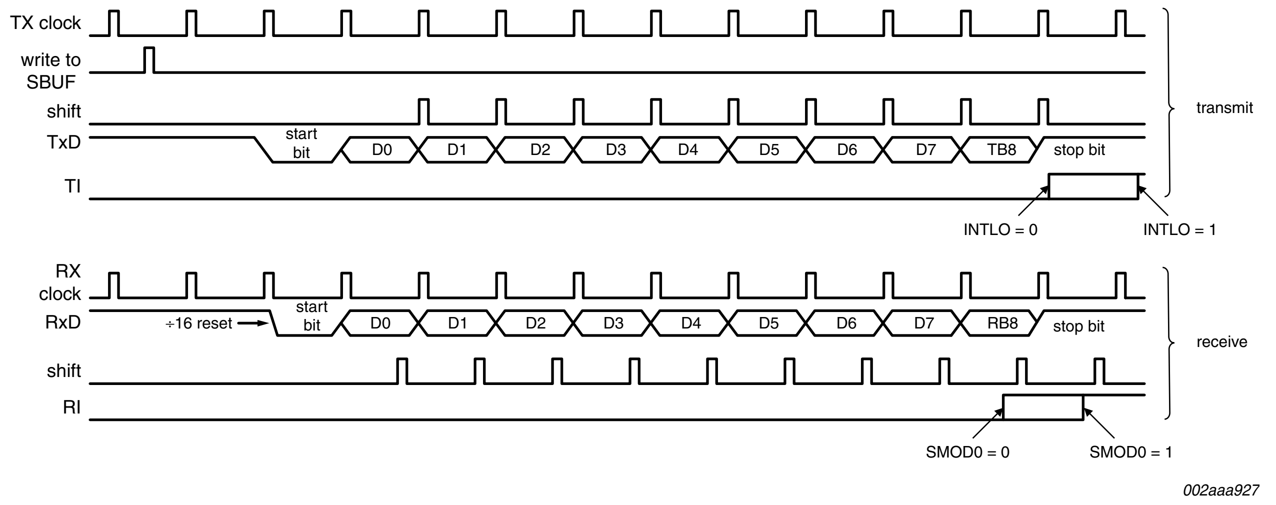


Fig 20. Serial Port Mode 2 or 3 (only single transmit buffering case is shown).

10.13 Framing error and RI in Modes 2 and 3 with SM2 = 1

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

Table 46: FE and RI when SM2= 1 in Modes 2 and 3.

Mode	PCON.6 (SMOD0)	RB8	RI	FE
2	0	0	No RI when RB8 = 0	Occurs during STOP bit
		1	Similar to Figure 20, 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 20, with SMOD0 = 1, RI occurs during STOP bit	Occurs during STOP bit

10.14 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 and 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 and force the device into ISP mode. This occurs if the UART is enabled and the the EBRR bit (AUXR1.6) is set and a break occurs.

10.15 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 SnBUF while the previous data is being shifted out.

10.16 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).

10.17 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.
5. If there is no more data, then:
   - If DBISEL is logic 0, no more interrupts will occur.

- If DBISEL is logic 1 and INTLO is logic 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 logic 1 and INTLO is logic 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).
- Note that if DBISEL is logic 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.

6. If there is more data, the CPU writes to SBUF again. Then:
   - If INTLO is logic 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 logic 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.

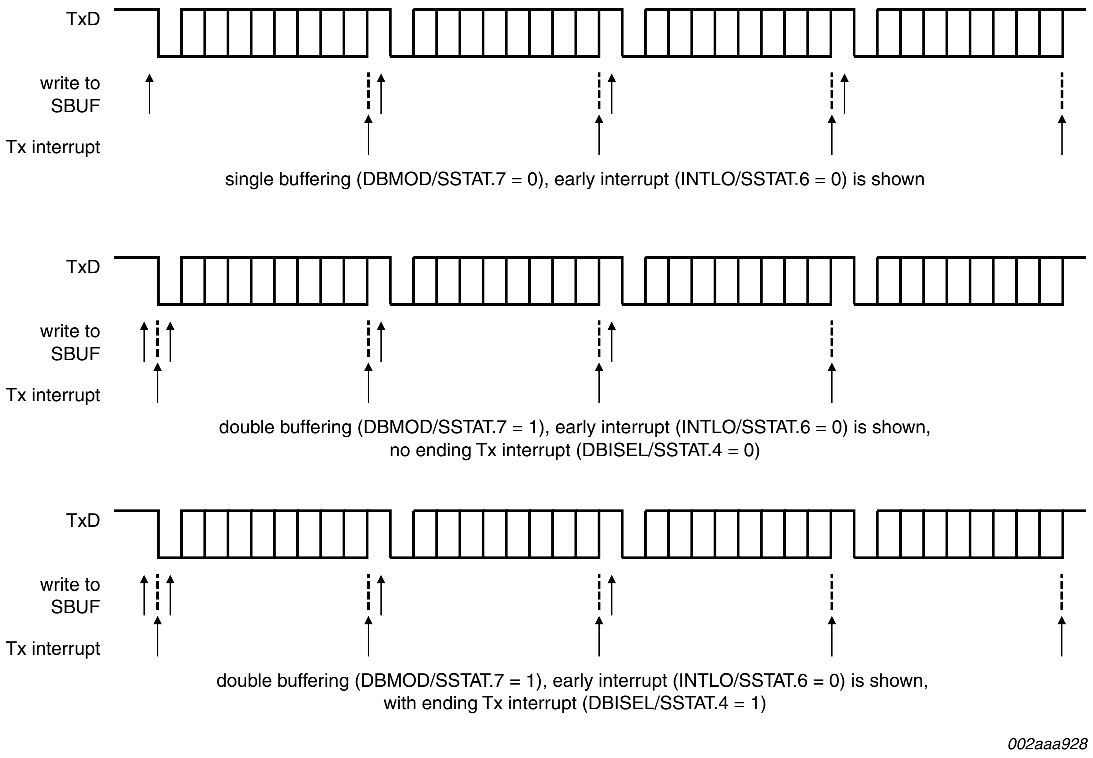

TxD
write to SBUF
Tx interrupt
single buffering (DBMOD/SSTAT.7 = 0), early interrupt (INTLO/SSTAT.6 = 0) is shown
TxD
write to SBUF
Tx interrupt
double buffering (DBMOD/SSTAT.7 = 1), early interrupt (INTLO/SSTAT.6 = 0) is shown, no ending Tx interrupt (DBISEL/SSTAT.4 = 0)
TxD
write to SBUF
Tx interrupt
double buffering (DBMOD/SSTAT.7 = 1), early interrupt (INTLO/SSTAT.6 = 0) is shown, with ending Tx interrupt (DBISEL/SSTAT.4 = 1)
002aaa928

Fig 21. Transmission with and without double buffering.

10.18 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 Section 10.17 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 logic 0, no more interrupt will occur.
   - If DBISEL is logic 1 and INTLO is logic 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 logic 1 and INTLO is logic 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 logic 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 logic 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.
9. Go to 4.
10. Note that if DBISEL is logic 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.

10.19 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 logic 0 in Mode 1.

10.20 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:

Table 47: Slave 0/1 examples

Example 1				Example 2			
Slave 0	SADDR	=	1100 0000	Slave 1	SADDR	=	1100 0000
	SADEN	=	1111 1101		SADEN	=	1111 1110
	Given	=	1100 00X0		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:

Table 48: Slave 0/1/2 examples

Example 1				Example 2				Example 3			
Slave 0	SADDR	=	1100 0000	Slave 1	SADDR	=	1110 0000	Slave 2	SADDR	=	1100 0000
	SADEN	=	1111 1001		SADEN	=	1111 1010		SADEN	=	1111 1100
	Given	=	1100 0XX0		Given	=	1110 0X0X		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.

11. Analog comparators

Two analog comparators are provided on the P89LPC904. Input options allow use of the comparators in different configurations. Comparator operation is such that the output is a logic 1 (which may be read in a register) 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.

11.1 Comparator configuration

Each comparator has a control register, CMP1 for comparator 1 and CMP2 for comparator 2. The control registers are identical and are shown in Table 50.

The overall connections to both comparators are shown in Figure 22. There are two possible configurations for each comparator, as determined by the control bits in the corresponding CMPn register. These configurations are shown in Figure 23.

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.

Table 49: Comparator Control register (CMP1 - address ACh, CMP2 - address ADh) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	CEn	-	CNn	-	COn	CMFn
Reset	x	x	0	0	0	0	0	0

Table 50: Comparator Control register (CMP1 - address ACh, CMP2 - address ADh) bit description

Bit	Symbol	Description
0	CMFn	Comparator interrupt flag. This bit is set by hardware whenever the comparator output COn changes state. This bit will cause a hardware interrupt if enabled. Cleared by software.
1	COn	Comparator output, synchronized to the CPU clock to allow reading by software.
2	-	reserved
3	CNn	Comparator negative input select. When logic 0, the comparator reference pin CMPREF is selected as the negative comparator input. When logic 1, the internal comparator reference, Vref, is selected as the negative comparator input.
4	-	reserved
5	CEn	Comparator enable. When set, the corresponding comparator function is enabled. Comparator output is stable 10 microseconds after CEn is set.
6	-	reserved
7	-	reserved

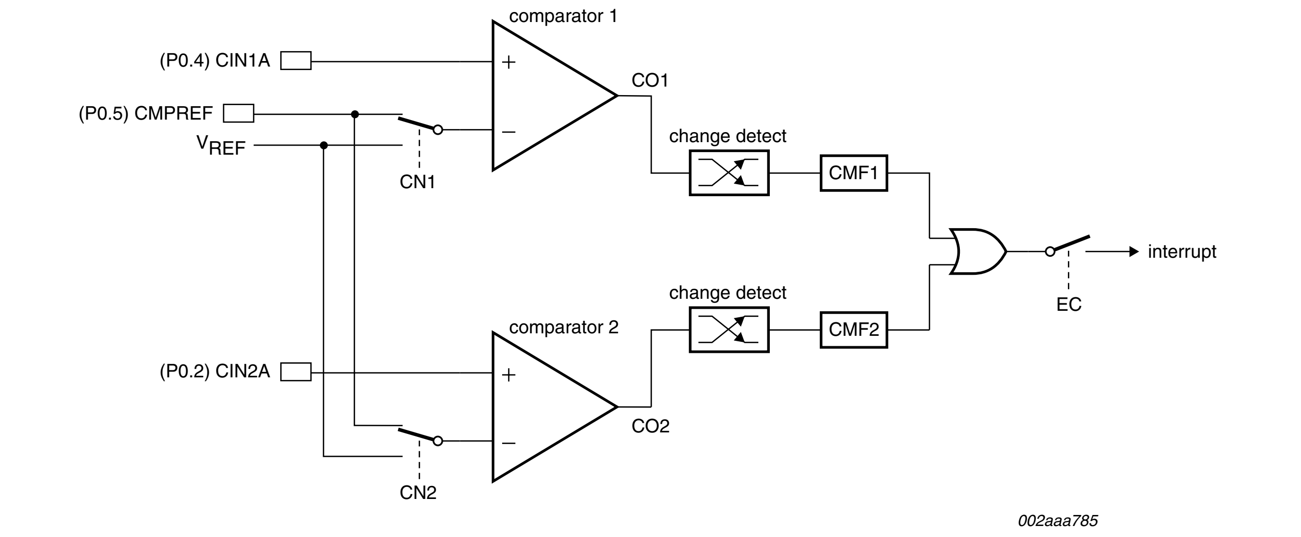


Fig 22. Comparator input and output connections.

11.2 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 *P89LPC904 data sheet* for specifications.

11.3 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.

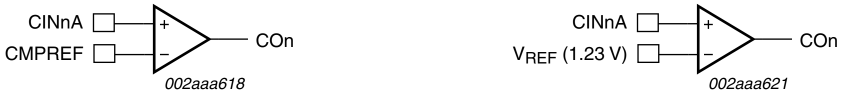
When a comparator is disabled the comparator's output, COx, goes HIGH. If the comparator output was LOW and then is disabled, the resulting transition of the comparator output from a LOW to HIGH state will set the comparator flag, CMFx. This will cause an interrupt if the comparator interrupt is enabled.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

11.4 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 logic 1, or simply putting the device in Total Power-down mode.



a. CNn = 0

b. CNn = 1

Fig 23. Comparator configurations.

11.5 Comparators 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 CIN1A, CMPREF.
 ANL   P0M2,#0CFh  ;Disable digital OUTPUTS on pins that are used
 ORL   P0M1,#030h  ;for analog functions: CIN1A, CMPREF.
 MOV   CMP1,#020h  ;Turn on comparator 1 and set up for:
                   ; - Positive input on CIN1A.
                   ; - Negative input from CMPREF pin.
 CALL  delay10us   ;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 Port 0 is equal to or not equal to a certain pattern. This function can be used for bus address recognition or 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), then any key connected to Port0 which is enabled by KBMASK register is 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 cause an interrupt, the pattern on Port 0 must be held longer than 6 CCLKs.

Table 51: Keypad Pattern register (KBPATN - address 93h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	KBPATN.7	KBPATN.6	KBPATN.5	KBPATN.4	KBPATN.3	KBPATN.2	KBPATN.1	KBPATN.0
Reset	1	1	1	1	1	1	1	1

Table 52: Keypad Pattern register (KBPATN - address 93h) bit description

Bit	Symbol	Access	Description
0:7	KBPATN.7:0	R/W	Pattern bit 0 - bit 7

Table 53: Keypad Control register (KBCON - address 94h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	PATN_SEL	KBIF
Reset	x	x	x	x	x	x	0	0

Table 54: Keypad Control register (KBCON - address 94h) bit description

Bit	Symbol	Access	Description
0	KBIF	R/W	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 logic 0.
1	PATN_SEL	R/W	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.
2:7	-	-	reserved

Table 55: Keypad Interrupt Mask register (KBMASK - address 86h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	KBMASK.7	KBMASK.6	KBMASK.5	KBMASK.4	KBMASK.3	KBMASK.2	KBMASK.1	KBMASK.0
Reset	0	0	0	0	0	0	0	0

Table 56: Keypad Interrupt Mask register (KBMASK - address 86h) bit description

Bit	Symbol	Description
0:1	-	Reserved, should always be written with a logic 0.
2	KBMASK.2	When set, enables P0.2 as a cause of a Keypad Interrupt.
3:7	-	Reserved, should always be written with a logic 0.

[1] The Keypad Interrupt must be enabled in order for the settings of the KBMASK register to be effective.

13. Watchdog timer (WDT)

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.

13.1 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 13.2). Additional bits in WDCON allow the user to select the clock source 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 Table 57 for details.

Figure 26 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 13.3).

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.

Table 57: 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 1 and cannot be cleared by software.

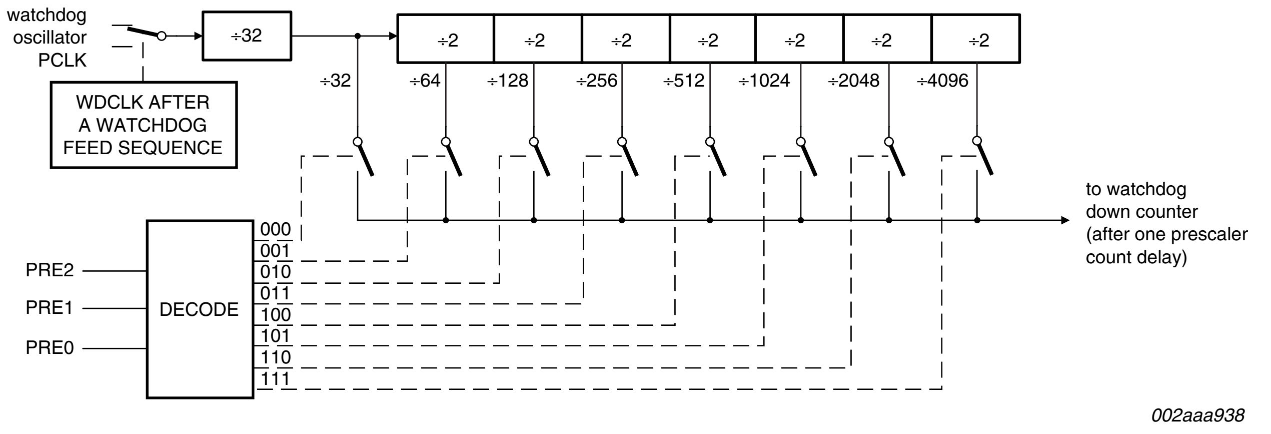


Fig 24. Watchdog prescaler.

13.2 Feed sequence

The Watchdog timer control register and the 8-bit down counter (See Figure 25) 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 P89LPC904 interrupt system is enabled and there is a possibility of an interrupt request occurring 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.

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

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

where:

PRE is the value of prescaler (PRE2 to PRE0) which can be the range 0 to 7, and;

WDL is the value of Watchdog load register which can be the range of 0 to 255.

The minimum number of tclks is:

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

The maximum number of tclks is:

$$tclks = (2^{(5+7)})(255 + 1) + 1 = 1048577 \tag{3}$$

Table 60 shows sample P89LPC904 timeout values.

Table 58: Watchdog Timer Control register (WDCON - address A7h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK
Reset	1	1	1	x	x	1	1/0	1

Table 59: Watchdog Timer Control register (WDCON - address A7h) bit description

Bit	Symbol	Description
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 13.5). (Note: If both WDTE and WDSE are set to 1, this bit is forced to 1.) Refer to Section 13.3 for details.
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 logic 0 to this bit in software.
2	WDRUN	Watchdog Run Control. The Watchdog timer is started when WDRUN = 1 and stopped when WDRUN = 0. This bit is forced to logic 1 (Watchdog running) and cannot be cleared to zero if both WDTE and WDSE are set to 1.
3:4	-	reserved
5	PRE0	Clock Prescaler Tap Select. Refer to Table 60 for details.
6	PRE1	
7	PRE2	

Table 60: Watchdog timeout vales.

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

13.3 Watchdog clock source

The Watchdog timer system has an on-chip 400 KHz oscillator. The Watchdog timer can be clocked from either the Watchdog oscillator or from PCLK (refer to Figure 24) 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 26, 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 two old clock source counts plus two new clock cycles.

Note: When switching clocks, it is important that the old clock source is left enabled for two 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 logic 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.

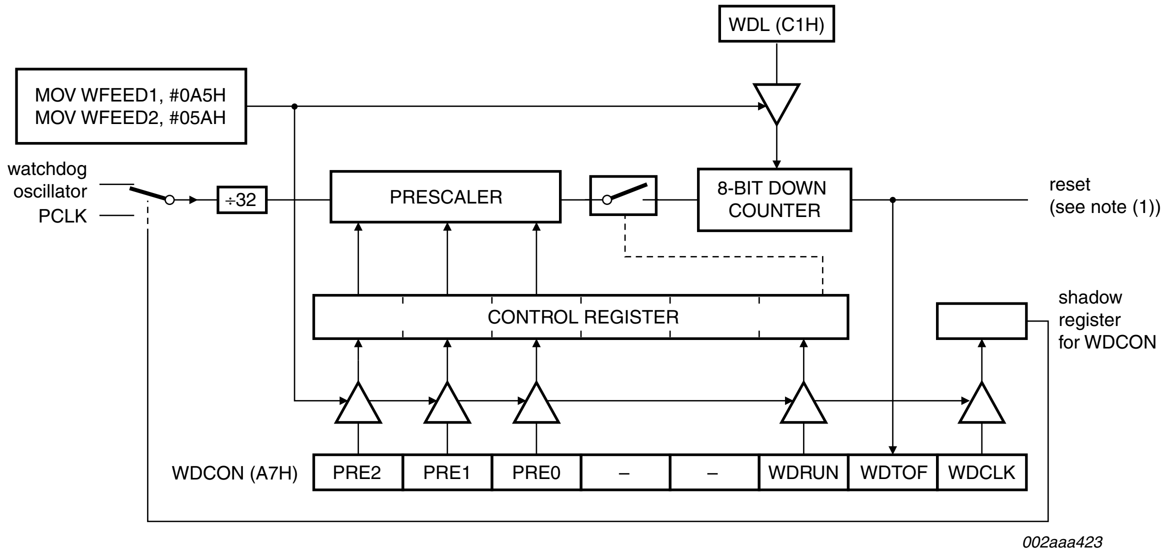


Fig 25. Watchdog timer in Watchdog mode (WDTE = 1).

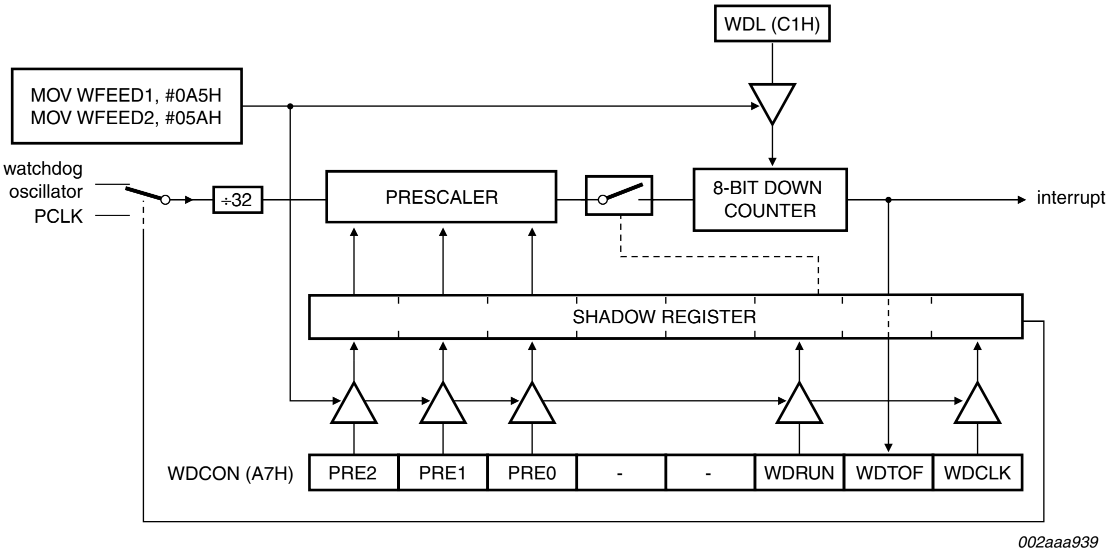


Fig 26. Watchdog timer in Timer Mode (WDTE = 0).

13.4 Watchdog timer in Timer mode

Figure 26 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 logic 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.

13.5 Power-down operation

The WDT oscillator will continue to run in power-down, consuming approximately 50 µA, 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 13.3). Power-down mode will also prevent PCLK from running and therefore the Watchdog is effectively disabled.

13.6 Periodic wake-up from power-down without an external oscillator

Without using an external oscillator source, the power consumption required in order to have a periodic wake-up is determined by the power consumption of the internal oscillator source used to produce the wake-up. The Real-time clock running from the internal RC oscillator can be used. The power consumption of this oscillator is approximately 300 µA. Instead, if the WDT is used to generate interrupts the current is reduced to approximately 50 µA. 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 Table 62.

Table 61: AUXR1 register (address A2h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS
Reset	0	0	0	0	0	0	x	0

Table 62: AUXR1 register (address A2h) bit description

Bit	Symbol	Description
0	DPS	Data Pointer Select. Chooses one of two Data Pointers.
1	-	Not used. Allowable to set to a logic 1.
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.
3	SRST	Software Reset. When set by software, resets the P89LPC904 as if a hardware reset occurred.
4	-	reserved
5	-	reserved
6	EBRR	UART Break Detect Reset Enable. If logic 1, UART Break Detect will cause a chip reset and force the device into ISP mode.
7	CLKLP	Clock Low Power Select. When set, reduces power consumption in the clock circuits. Can be used when the clock frequency is 8 MHz or less. After reset this bit is cleared to support up to 12 MHz operation.

14.1 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 logic 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.

14.2 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.

MOVC A, @A+DPTR — Move code byte relative to DPTR to the accumulator.

MOVX A, @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 P89LPC904 since the part does not have an external data bus. However, they may be used to access Flash configuration information (see Flash Configuration section) or auxiliary data (XDATA) memory.

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 memory

The P89LPC904 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 P89LPC904 Flash reliably stores memory contents even after 100,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. The P89LPC904 uses V_{DD} as the supply voltage to perform the Program/Erase algorithms.

15.1 Features

- In-Circuit serial Programming (ICP) with industry-standard commercial programmers.
- IAP-Lite allows individual and multiple bytes of code memory to be used for data storage and programmed under control of the end application.
- Boot vector allows user provided Flash loader code to reside anywhere in the Flash memory space, providing flexibility to the user.
- Programming and erase over the full operating voltage range.
- Read/Programming/Erase using IAP-Lite.
- Any flash erase/program operation in 4 ms.
- Programmable security for the code in the Flash for each sector.
- > 100,000 typical erase/program cycles for each byte.
- 10-year minimum data retention.

15.2 Using Flash as data storage: IAP-Lite

The Flash code memory array of this device supports IAP-Lite. 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. 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 programing 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 logic 0s 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 4 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 4 ms 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[3:0], 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.

Table 63: Flash Memory Control register (FMCON - address E4h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol (R)	-	-	-	-	HVA	HVE	SV	OI
Symbol (W)	FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0
Reset	0	0	0	0	0	0	0	0

Table 64: Flash Memory Control register (FMCON - address E4h) bit description

Bit	Symbol	Access	Description
0	OI	R	Operation interrupted. Set when cycle aborted due to an interrupt or reset.
	FMCMD.0	W	Command byte bit 0.
1	SV	R	Security violation. Set when an attempt is made to program, erase, or CRC a secured sector or page.
	FMCMD.1	W	Command byte bit 1
2	HVE	R	High voltage error. Set when an error occurs in the high voltage generator.
	FMCMD.2	W	Command byte bit 2.
3	HVA	R	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.
	FMCMD.3	W	Command byte bit 3.
4:7	-	R	reserved
4:7	FMCMD.4	W	Command byte bit 4.
4:7	FMCMD.5	W	Command byte bit 5.
4:7	FMCMD.6	W	Command byte bit 6.
4:7	FMCMD.7	W	Command byte bit 7.

An assembly language routine to load the page register and perform an erase/program operation is shown below.

```
;**************************************************
;* pgm user code   *
;**************************************************
;*                                                 *
;* 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 and 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
```

A C-language routine to load the page register and perform an erase/program operation is shown below.

```
#include <REG931.H>
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(0x00,0xC0);
}

bit PGM_USER (unsigned char page_hi, unsigned char page_lo)
      {
           #define LOAD 0x00// clear page register, enable loading
           #define EP 0x68// erase and 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 and 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);
}
```

15.3 In-circuit programming (ICP)

In-Circuit Programming is a method intended to allow commercial programmers to program and erase these devices without removing the microcontroller from the system. The In-Circuit Programming facility consists of a series of internal hardware resources to facilitate remote programming of the P89LPC904 through a two-wire serial interface. Philips has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area. The ICP function uses five pins (V_{DD}, V_{SS}, P0.5, P0.4, and $\overline{RST}$). Only a small connector needs to be available to interface your application to an external programmer in order to use this feature.

15.4 Power on reset code execution

The P89LPC904 contains two special Flash elements: the BOOT VECTOR and the Boot Status Bit. Following reset, the P89LPC904 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 logic 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 settings for these devices are show in Table 65, below.

Table 65: Boot loader address and default Boot vector

Product	Flash size	End address	Signature bytes			Sector size	Page size	Default Boot vector
			Mfg. id	Id 1	Id 2			
P89LPC904	1K × 8	03FFh	15h	DDh	21h	256 × 8	16 × 8	00h

15.5 Hardware activation of Boot Vector address

The device can be forced into using the Boot Vector address for code execution during a power-on sequence (see Figure 27). This is accomplished by powering up the device with the reset pin initially held LOW and holding the pin LOW for a fixed time after V_{DD} rises to its normal operating value. This is followed by three, and only three, properly timed LOW-going pulses. Fewer or more than three pulses will result in the device not entering ISP mode. Timing specifications may be found in the data sheet for this device.

This has the same effect as having a non-zero status bit. This allows an application to be built that will normally execute the user code starting at address 0000 but can be manually forced into using the Boot Vector address.

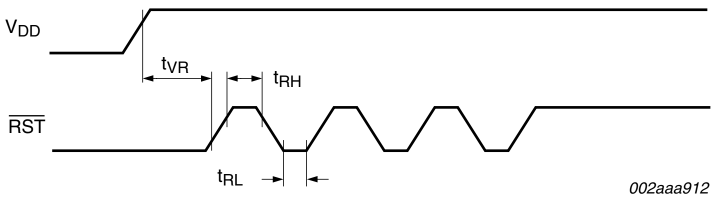


Fig 27. Forcing Boot Vector address execution.

15.6 Flash write enable

This device has hardware write enable protection. This protection applies to IAP-Lite mode and applies to both the user code memory space and the user configuration bytes (UCFG1, BOOTVEC, and BOOTSTAT). This protection does not apply to ICP mode. If the Activate Write Enable (AWE) bit in BOOTSTAT.7 is a logic 0, an internal Write Enable (WE) flag is forced set and writes to the flash memory and configuration bytes are enabled. If the Active Write Enable (AWE) bit is a logic 1 then the state of the internal WE flag can be controlled by the user.

The WE flag is SET by writing the Set Write Enable (08H) command to FMCON followed by a key value (96H) to FMDATA:

```
MOV FMCON,#08H
MOV FMDATA,#96H
```

The WE flag is CLEARED by writing the Clear Write Enable (0BH) command to FMCON followed by a key value (96H) to FMDATA, or by a reset:

```
MOV FMCON,#0BH
MOV FMDATA,#96H
```

15.7 Configuration byte protection

In addition to the hardware write enable protection, described above, the ‘configuration bytes’ may be separately write protected. These configuration bytes include UCFG1, BOOTVEC, and BOOTSTAT. This protection applies to IAP-Lite mode and does not apply to ICP mode.

If the Configuration Write Protect Writ (CWP) bit in BOOTSTAT.6 is a logic 1 writes to the configuration bytes are disabled. If the Configuration Write Protect (CWP) bit in BOOTSTAT.6 is a logic 0 writes to the configuration bytes are enabled. The CWP bit is set by programming the BOOTSTAT register. This bit is cleared by using the Clear Configuration Protection function in IAP-Lite.

The Clear Configuration Protection command can be disabled in IAP-Lite mode by programming the Disable Clear Configuration Protection (DCCP) bit in BOOTSTAT.7 to a logic 1. When DCCP is set, the CCP command may still be used in ICP mode. This bit is cleared by writing the Clear Configuration Protection command in ICP or mode.

15.8 IAP-Lite error status

It is not possible to use the Flash memory as the source of program instructions while programming or erasing this same Flash memory. During an IAP-Lite write operation the CPU enters a program-idle state. The CPU will remain in this program-idle state until the erase/program cycle is completed. These cycles are self timed. When the cycle is completed, code execution resumes. If an interrupt occurs during this cycle, the cycle will be aborted so that the Flash memory can be used as the source of instructions to service the interrupt. An IAP-Lite error condition will be indicated by reading status from FMCON.The status information returned is shown in Table 66. If the application permits interrupts during the IAP-Lite cycle, the user code should check this status after each operation to see if an error occurred. If the operation was aborted, the user's code will need to repeat the operation.

Table 66: FMCON error status

Bit	Flag	Description
0	OI	Operation Interrupted. Indicates that an operation was aborted due to an interrupt occurring during a program or erase cycle.
1	SV	Security Violation. Set if program or erase operation fails due to security settings. Cycle is aborted. Memory contents are unchanged. CRC output is invalid.
2	HVE	High Voltage Error. Set if error detected in high voltage generation circuits. Cycle is aborted. Memory contents may be corrupted.
3	HVA	High Voltage Abort due to interrupt or brownout disabled.
4:7	-	unused; reads as a logic 0

15.9 User configuration bytes

A number of user-configurable features of the P89LPC904 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 an Flash byte UCFG1 shown in Table 68.

Table 67: Flash User Configuration Byte (UCFG1) bit allocation

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

Table 68: Flash User Configuration Byte (UCFG1) bit description

Bit	Symbol	Description
0	FOSC0	CPU clock source select. See Section 2 “Clocks” on page 12 for additional information. Combinations other than those shown in Table 69 “Oscillator type selection” on page 68are reserved for future use should not be used.
1	FOSC1	
2	FOSC2	
3	-	reserved
4	WDSE	Watchdog Safety Enable bit. Refer to Table 69 for details.

Table 68: Flash User Configuration Byte (UCFG1) bit description

Bit	Symbol	Description
5	BOE	Brownout Detect Enable (see Section 6.1 “Brownout detection” on page 26).
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 overridden 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.
7	WDTE	Watchdog timer reset enable. When set = 1, enables the Watchdog timer reset. When cleared = 0, disables the Watchdog timer reset. The timer may still be used to generate an interrupt. Refer to Table 69 for details.

Table 69: Oscillator type selection

FOSC[2:0]	Oscillator configuration
111	External clock input on XTAL1.
100	Watchdog Oscillator, 400 kHz (+20/–30 % tolerance).
011	Internal RC oscillator, 7.373 MHz ± 2.5 %.
010	reserved
001	reserved
000	reserved

15.10 User security bytes

This device has three security bits associated with each of its eight sectors, as shown in Table 70

Table 70: Sector Security Bytes (SECx) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	EDISx	SPEDISx	MOVCDISx
Unprogrammed value	0	0	0	0	0	0	0	0

Table 71: Sector Security Bytes (SECx) bit description

Bit	Symbol	Description
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.
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-Lite, ICP) or a 'global' erase command (ICP).
2	EDISx	Erase Disable IAP. Disables the ability to perform an erase of sector ‘x’ in IAP-Lite mode. When programmed, this bit and sector x can only be erased by a 'global' erase command using a commercial programmer (ICP). This bit and sector x CANNOT be erased in IAP-Lite mode.
3:7	-	reserved

Table 72: 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 commands or an erase page command. Cycle aborted. Memory contents unchanged. Global erase is allowed.

15.11 Boot Vector register

Table 73: Boot Vector (BOOTVEC) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	BOOTV1	BOOTV0
Factory default value	0	0	0	0	0	0	1	1

Table 74: Boot Vector (BOOTVEC) bit description

Bit	Symbol	Description
0:1	BOOTV.1:0	Boot vector. If the Boot Vector is selected as the reset address, the P89LPC904 will start execution at an address comprised of 00h in the lower eight bits and this BOOTVEC as the upper eight bits after a reset.
2:7	-	reserved

15.12 Boot status register

Table 75: Boot Status (BOOTSTAT) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DCCP	CWP	AWP	-	-	-	--	BSB
Factory default value	0	0	0	0	0	0	0	1

Table 76: Boot Status (BOOTSTAT) bit description

Bit	Symbol	Description
0	BSB	Boot Status Bit. If programmed to logic 1, the P89LPC904 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 7.1 “Reset vector”).
1:4	-	reserved

Table 76: Boot Status (BOOTSTAT) bit description

Bit	Symbol	Description
5	AWP	Activate Write Protection bit. When this bit is cleared, the internal Write Enable flag is forced to the set state, thus writes to the flash memory are always enabled. When this bit is set, the Write Enable internal flag can be set or cleared using the Set Write Enable (SWE) or Clear Write Enable (CWE) commands.
6	CWP	Configuration Write Protect bit. Protects inadvertent writes to the user programmable configuration bytes (UCFG1, BOOTVEC, and BOOTSTAT). If programmed to a logic 1, the writes to these registers are disabled. If programmed to a logic 0, writes to these registers are enabled. This bit is set by programming the BOOTSTAT register. This bit is cleared by writing the Clear Configuration Protection (CCP) command to FMCON followed by writing 96H to FMDATA.
7	DCCP	Disable Clear Configuration Protection command. If Programmed to ‘1’, the Clear Configuration Protection (CCP) command is disabled during IAP-Lite mode. This command can still be used in ICP mode. If programmed to ‘0’, the CCP command can be used in all programming modes. This bit is set by programming the BOOTSTAT register. This bit is cleared by writing the Clear Configuration Protection (CCP) command in ICP mode.

16. Instruction set

Table 77: Instruction set summary

Mnemonic	Description	Bytes	Cycles	Hex code
	ARITHMETIC			
ADD A,Rn	Add register to A	1	1	28 to 2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26 to 27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38 to 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 to 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 to 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 to 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 to 0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06 to 07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18 to 1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16 to 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 to 5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56 to 57
ANL A,#data	AND immediate to A	2	1	54
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 to 4F
ORL A,dir	OR direct byte to A	2	1	45

Table 77: Instruction set summary

Mnemonic	Description	Bytes	Cycles	Hex code
ORL A,@Ri	OR indirect memory to A	1	1	46 to 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 to 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 to 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
Rotate A right	RR A	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 to EF
MOV A,dir	Move direct byte to A	2	1	E5
Move indirect memory to A	MOV A,@Ri	1	1	E6 to E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8 to FF
MOV Rn,dir	Move direct byte to register	2	2	A8 to AF
MOV Rn,#data	Move immediate to register	2	1	78 to 7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88 to 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 to 87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6 to F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6 to A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76 to 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

Table 77: Instruction set summary

Mnemonic	Description	Bytes	Cycles	Hex code
MOVX A,@Ri	Move external data(A8) to A	1	2	E2 to 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 to 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 to CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6 to C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6 to 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			
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

Table 77: Instruction set summary

Mnemonic	Description	Bytes	Cycles	Hex code
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠ 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 to BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6 to B7
DJNZ Rn,rel	Decrement register, jnz relative	2	2	D8 to DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5
	MISCELLANEOUS			
NOP	No operation	1	1	00

17. 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 licence 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.

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.

18. Contents

continued >>

Date of release: 2 March 2005

Published in the Netherlands