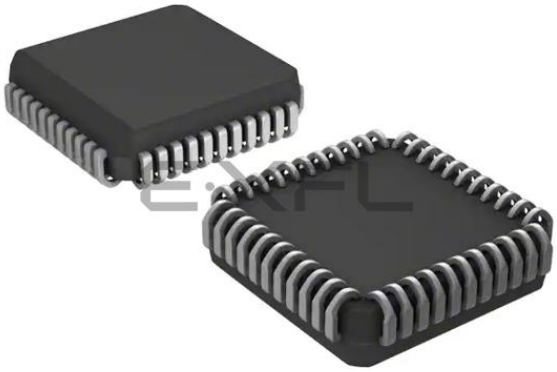

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p89v660fa-512

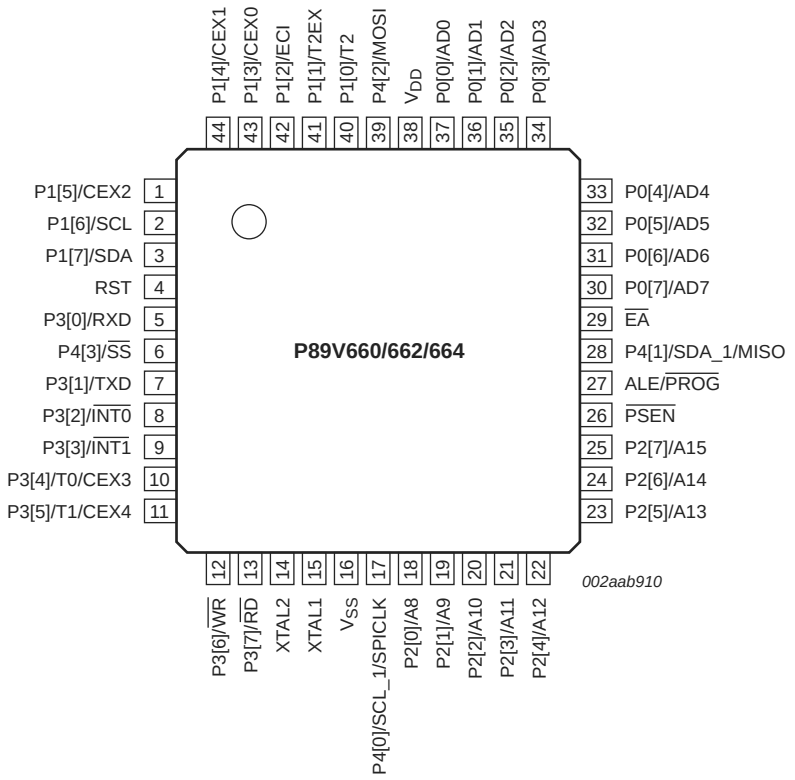

Fig 3. TQFP44 pin configuration

6.3.3 Boot block

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

6.3.4 Power-on reset code execution

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

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

Table 10. Default boot vector values and ISP entry points

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

6.3.5 Hardware activation of the bootloader

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

6.3.6 ISP

ISP is performed without removing the microcontroller from the system. The ISP facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the P89V660/662/664 through the serial port. This firmware is provided by NXP and embedded within each P89V660/662/664 device. The NXP ISP facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area. The ISP function uses five pins (V_{DD}, V_{SS}, TXD, RXD, and RST). Only a small connector needs to be available to interface your application to an external circuit in order to use this feature.

6.3.7 Using ISP

The ISP feature allows for a wide range of baud rates to be used in your application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts

6.3.8 IAP method

Several IAP calls are available for use by an application program to permit selective erasing, reading and programming of flash pages, security bits, security bits, Status bit, and device id. All calls are made through a common interface, PGM_MTP. The programming functions are selected by setting up the microcontroller's registers before making a call to PGM_MTP at FFF0H. The IAP calls are shown in Table 12.

Table 12. IAP function calls

IAP function	IAP call parameters
Read Id	**Input parameters:**
	R1 = 00H or 80H (WDT feed)
	DPH = 00H
	DPL = 00H = manufacturer id
	DPL = 01H = device id 1
	DPL = 02H = device id 2
	DPL = 03H = 6x/12x bit (bit 7 = 1 = 6x)
	DPL = 80H = ISP version number
	Return parameter(s):
	ACC = requested parameter
Erase 8 kB/16 kB code block	**Input parameters:**
	R1 = 01H or 81H (WDT feed)
	DPL = 00H, block 0, 0 kB to 8 kB
	DPL = 20H, block 1, 8 kB to 16 kB
	DPL = 40H, block 2, 16 kB to 32 kB
	DPL = 80H, block 3, 32 kB to 48 kB
	DPL = C0H, block 4, 48 kB to 64 kB
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail
Program User Code	**Input parameters:**
	R1 = 02H or 82H (WDT feed)
	DPH = memory address MSB
	DPL = memory address LSB
	ACC = byte to program
	Return parameter(s):
	ACC = 00 = pass
	ACC = !00 = fail
Read User Code	**Input parameters:**
	R1 = 03H or 83H (WDT feed)
	DPH = memory address MSB
	DPL = memory address LSB
	Return parameter(s):
	ACC = device data

Table 24. Slave Receiver mode *…continued*

Status code (S1STA)	Status of the I²C-bus hardware	Application software response					Next action taken by I²C-bus hardware
		to/from S1DAT	to S1CON				
			STA	STO	SI	AA	
88H	Previously addressed with own SLA address; Data has been received; NOT ACK has been returned.	Read data byte or	0	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address.
		read data byte or	0	0	0	1	Switched to not addressed SLA mode; Own SLA will be recognized; general call address will be recognized if S1ADR.0 = 1.
		read data byte or	1	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		read data byte	1	0	0	1	Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free.
90H	Previously addressed with General call; Data has been received; ACK has been returned.	Read data byte or	x	0	0	0	Data byte will be received and NOT ACK will be returned.
		read data byte	x	0	0	1	Data byte will be received and ACK will be returned.
98H	Previously addressed with General call; Data has been received; NOT ACK has been returned.	Read data byte	0	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or General call address.
		read data byte	0	0	0	1	Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1.
		read data byte	1	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		read data byte	1	0	0	1	Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free.

Table 28. TMOD - Timer/Counter mode control register (address 89H) M1/M0 operating mode *…continued*

M1	M0	Operating mode	
1	0	2	8-bit auto-reload Timer/Counter 'THx' holds a value which is to be reloaded into 'TLx' each time it overflows.
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

Table 29. TCON - Timer/Counter control register (address 88H) bit allocation
Bit addressable; Reset value: 0000 0000B; Reset source(s): any reset

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

Table 30. TCON - Timer/Counter control register (address 88H) bit description

Bit	Symbol	Description
7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to Timer 1 Interrupt routine, or by software.
6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off.
5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to Timer 0 Interrupt routine, or by software.
4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt 1 edge/LOW-state 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-state that triggers external interrupt 1.
1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt 0 edge/LOW-state is detected. Cleared by hardware when the interrupt is processed, or by software.
0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/LOW-state that triggers external interrupt 0.

6.5.1 Mode 0

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a fixed divide-by-32 prescaler. Figure 14 shows Mode 0 operation.

6.7.1 Mode 0

Serial data enters and exits through RXD and TXD outputs the shift clock. Only 8 bits are transmitted or received, LSB first. The baud rate is fixed at $\frac{1}{6}$ of the CPU clock frequency. UART configured to operate in this mode outputs serial clock on TXD line no matter whether it sends or receives data on RXD line.

6.7.2 Mode 1

10 bits are transmitted (through TXD) or received (through RXD): a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). When data is received, the stop bit is stored in RB8 in Special Function Register SCON. The baud rate is variable and is determined by the Timer $\frac{1}{2}$ overflow rate.

6.7.3 Mode 2

11 bits are transmitted (through TXD) or received (through RXD): start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or (e.g. 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, while the stop bit is ignored. The baud rate is programmable to either $\frac{1}{16}$ or $\frac{1}{32}$ of the CPU clock frequency, as determined by the SMOD1 bit in PCON.

6.7.4 Mode 3

11 bits are transmitted (through TXD) or received (through RXD): a start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). In fact, 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 $\frac{1}{2}$ overflow rate.

Table 37. SCON - Serial port control register (address 98H) bit allocation
Bit addressable; Reset value: 00H

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

Table 38. SCON - Serial port control register (address 98H) bit description

Bit	Symbol	Description
7	SM0/FE	The usage of this bit is determined by SMOD0 in the PCON register. If SMOD0 = 0, this bit is SM0, which with SM1, defines the serial port mode. If SMOD0 = 1, this bit is 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 can only be cleared by software. (Note: It is recommended to set up UART mode bits SM0 and SM1 before setting SMOD0 to '1'.)
6	SM1	With SM0, defines the serial port mode (see Table 39 below).
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 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be '0'.
4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.

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 irretrievably 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.

6.7.8 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 so 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 a way that the 9th bit is ‘1’ in an address byte and ‘0’ in the data byte. With SM2 = 1, no slave will be interrupted by a data byte, i.e. the received 9th bit is ‘0’. However, an address byte having the 9th bit set to ‘1’ will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed or not. The addressed slave will clear its SM2 bit and prepare to receive the data (still 9 bits long) 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.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit, although this is better done with the Framing Error flag. When UART receives data in mode 1 and SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

6.7.9 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 for the UART 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.

This device uses the methods presented in Figure 22 to determine if a Given or Broadcast address has been received or not.

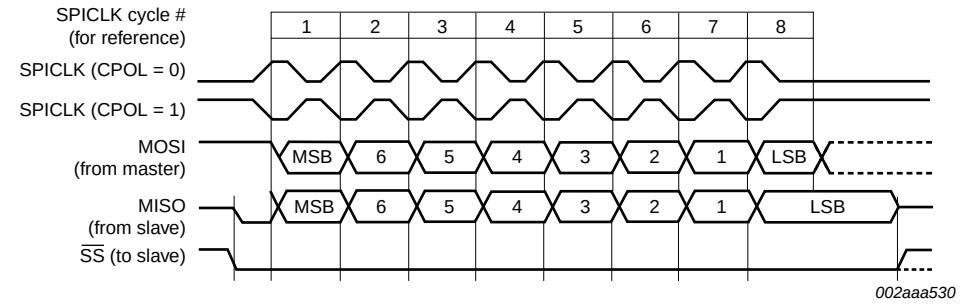

Fig 25. SPI transfer format with CPHA = 1

6.9 Watchdog timer

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 14-bit counter and the WatchDog Timer Reset (WDTRST) SFR. The WDT is disabled at reset. To enable the WDT, the user must write 01EH and 0E1H, in sequence, to the WDTRST SFR. When the WDT is enabled, it will increment every machine cycle while the oscillator is running and there is no way to disable the WDT, except through a reset (either hardware reset or a WDT overflow reset). When the WDT overflows, it will drive an output reset HIGH pulse at the RST pin.

When the WDT is enabled (and thus running) the user needs to reset it by writing 01EH and 0E1H, in sequence, to the WDTRST SFR to avoid WDT overflow. The 14-bit counter reaches overflow when it reaches 16383 (3FFFH) and this will reset the device.

The WDT's counter cannot be read or written. When the WDT overflows it will generate a output pulse at the reset pin with a duration of 98 oscillator periods in 6 clock mode or 196 oscillator periods in 12 clock mode.

6.10 PCA

The PCA includes a special 16-bit Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it. Module 0 is connected to CEX0, module 1 to CEX1, etc. Registers CH and CL contain current value of the free running up counting 16-bit PCA timer. The PCA timer is a common time base for all five modules and can be programmed to run at: $1/6$ the oscillator frequency, $1/2$ the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1[2]). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR (see Table 45 and Table 46).

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition.

The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

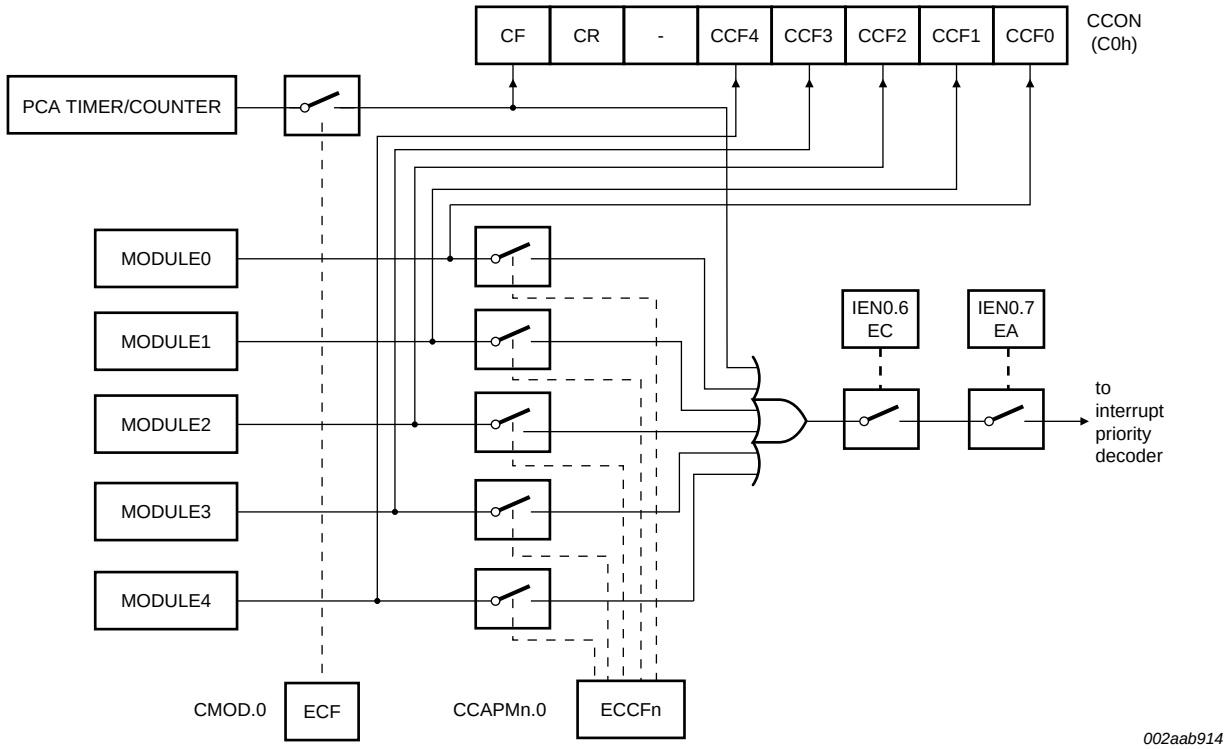

Fig 27. PCA interrupt system

Table 45. CMOD - PCA counter mode register (address C1H) bit allocation
Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF

Table 46. CMOD - PCA counter mode register (address C1H) bit description

Bit	Symbol	Description
7	CIDL	Counter Idle Control: CIDL = 0 programs the PCA Counter to continue functioning during Idle mode. CIDL = 1 programs it to be gated off during idle.
6	WDTE	WatchDog Timer Enable: WDTE = 0 disables watchdog timer function on module 4. WDTE = 1 enables it.
5 to 3	-	Reserved for future use. Should be set to '0' by user programs.
2 to 1	CPS1, CPS0	PCA Count Pulse Select (see Table 47 below).
0	ECF	PCA Enable Counter Overflow Interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function.

Table 47. CMOD - PCA counter mode register (address C1H) count pulse select

CPS1	CPS0	Select PCA input
0	0	0 Internal clock, f_{osc} / 6
0	1	1 Internal clock, f_{osc} / 6
1	0	2 Timer 0 overflow
1	1	3 External clock at ECI/P1[2] pin (max rate = f_{osc} / 4)

Table 48. CCON - PCA counter control register (address 0C0H) bit allocation

Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0

Table 49. CCON - PCA counter control register (address 0C0H) bit description

Bit	Symbol	Description
7	CF	PCA Counter Overflow Flag. Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
6	CR	PCA Counter Run Control Bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.
5	-	Reserved for future use. Should be set to '0' by user programs.
4	CCF4	PCA Module 4 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
3	CCF3	PCA Module 3 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
2	CCF2	PCA Module 2 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
1	CCF1	PCA Module 1 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
0	CCF0	PCA Module 0 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.

Table 50. CCAPMn - PCA modules compare/capture register (address CCAPM0 0C2H, CCAPM1 0C3H, CCAPM2 0C4H, CCAPM3 0C5H, CCAPM4 0C6H) bit allocation

Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

Table 51. CCAPMn - PCA modules compare/capture register (address CCAPM0 0C2H, CCAPM1 0C3H, CCAPM2 0C4H, CCAPM3 0C5H, CCAPM4 0C6H) bit description

Bit	Symbol	Description
7	-	Reserved for future use. Should be set to ‘0’ by user programs.
6	ECOMn	Enable Comparator. ECOMn = 1 enables the comparator function.
5	CAPPn	Capture Positive, CAPPn = 1 enables positive edge capture.
4	CAPNn	Capture Negative, CAPNn = 1 enables negative edge capture.
3	MATn	Match. When MATn = 1 a match of the PCA counter with this module’s compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.
2	TOGn	Toggle. When TOGn = 1, a match of the PCA counter with this module’s compare/capture register causes the CEXn pin to toggle.
1	PWMn	Pulse Width Modulation mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.
0	ECCFn	Enable CCF Interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.

Table 52. PCA module modes (CCAPMn register)

ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	Module function
0	0	0	0	0	0	0	no operation
x	1	0	0	0	0	x	16-bit capture by a positive-edge trigger on CEXn
x	0	1	0	0	0	x	16-bit capture by a negative-edge trigger on CEXn
x	1	1	0	0	0	x	16-bit capture by any transition on CEXn
1	0	0	1	0	0	x	16-bit software timer
1	0	0	1	1	0	x	16-bit high-speed output
1	0	0	0	0	1	0	8-bit PWM
1	0	0	1	x	0	x	watchdog timer

6.10.1 PCA capture mode

To use one of the PCA modules in the capture mode (Figure 28) either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module’s capture registers (CCAPnL and CCAPnH).

value in the module's CCAPnL SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPnL is reloaded with the value in CCAPnH. This allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

6.10.5 PCA watchdog timer

An on-board watchdog timer is available with the PCA to improve the reliability of the system without increasing chip count. Watchdog timers are useful for systems that are susceptible to noise, power glitches, or electrostatic discharge. Module 4 is the only PCA module that can be programmed as a Watchdog. However, this module can still be used for other modes if the Watchdog is not needed. Figure 31 shows a diagram of how the Watchdog works. The user pre-loads a 16-bit value in the compare registers. Just like the other compare modes, this 16-bit value is compared to the PCA timer value. If a match is allowed to occur, an internal reset will be generated. This will not cause the RST pin to be driven high.

user's software then must periodically change (CCAP4H,CCAP4L) to keep a match from occurring with the PCA timer (CH,CL). This code is given in the WATCHDOG routine shown above.

In order to hold off the reset, the user has three options:

1. Periodically change the compare value so it will never match the PCA timer.
2. Periodically change the PCA timer value so it will never match the compare values.
3. Disable the Watchdog by clearing the WDTE bit before a match occurs and then re-enable it.

The first two options are more reliable because the watchdog timer is never disabled as in option #3. If the program counter ever goes astray, a match will eventually occur and cause an internal reset. The second option is also not recommended if other PCA modules are being used. Remember, the PCA timer is the time base for **all** modules; changing the time base for other modules would not be a good idea. Thus, in most applications the first solution is the best option.

```
;CALL the following WATCHDOG subroutine periodically.
   CLR    EA           ;Hold off interrupts
   MOV    CCAP4L,#00   ;Next compare value is within 255 counts of
                        current PCA timer value
   MOV    CCAP4H,CH
   SETB   EA           ;Re-enable interrupts
   RET
```

This routine should not be part of an interrupt service routine, because if the program counter goes astray and gets stuck in an infinite loop, interrupts will still be serviced and the Watchdog will keep getting reset. Thus, the purpose of the Watchdog would be defeated. Instead, call this subroutine from the main program within 2^{16} count of the PCA timer.

Table 55. IEN0 - Interrupt enable register 0 (address A8H) bit allocation
Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	EA	EC	ES1	ES0	ET1	EX1	ET0	EX0

Table 56. IEN0 - Interrupt enable register 0 (address A8H) bit description

Bit	Symbol	Description
7	EA	Interrupt Enable Bit: EA = 1 interrupt(s) can be serviced, EA = 0 interrupt servicing disabled.
6	EC	PCA Interrupt Enable bit.
5	ES1	I²C-bus Interrupt Enable (primary).
4	ES0	Serial Port Interrupt Enable
3	ET1	Timer 1 Overflow Interrupt Enable.
2	EX1	External Interrupt 1 Enable.
1	ET0	Timer 0 Overflow Interrupt Enable.
0	EX0	External Interrupt 0 Enable.

Table 57. IEN1 - Interrupt enable register 1 (address E8H) bit allocation
Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	ES3	ES2	ET2

Table 58. IEN1 - Interrupt enable register 1 (address E8H) bit description

Bit	Symbol	Description
7 to 3	-	Reserved for future use. Should be set to ‘0’ by user programs.
2	ES3	SPI Interrupt Enable.
1	ES2	I²C-bus Interrupt Enable (secondary).
0	ET2	Timer 2 Interrupt Enable.

Table 59. IP0 - Interrupt priority 0 low register (address B8H) bit allocation
Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	PT2	PPC	PS1	PS0	PT1	PX1	PT0	PX0

Table 60. IP0 - Interrupt priority 0 low register (address B8H) bit description

Bit	Symbol	Description
7	PT2	Timer 2 Interrupt Priority Low Bit.
6	PPC	PCA Interrupt Priority Low Bit.
5	PS1	I²C-bus Interrupt Priority Low Bit.
4	PS0	Serial Port Interrupt Priority Low Bit.
3	PT1	Timer 1 Interrupt Priority Low Bit.
2	PX1	External Interrupt 1 Priority Low Bit.
1	PT0	Timer 0 Interrupt Priority Low Bit.
0	PX0	External Interrupt 0 Priority Low Bit.

Table 70. Static characteristics *…continued*
T_{amb} = −40 °C to +85 °C; V_{DD} = 4.5 V to 5.5 V; V_{SS} = 0 V

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{THL}	HIGH-LOW transition current	V_I = 2 V, ports 1, 2, 3, 4	[6]	-	-	−650	μA
I_{LI}	input leakage current	0.45 V < V_I < V_{DD} − 0.3 V, port 0		-	-	±10	μA
		0 V < V_I < 6 V, 0 V < V_{DD}< 5.5 V, SCL, SDA		-	-	10	μA
R_{pd}	pull-down resistance	on pin RST		40	-	225	kΩ
C_{iss}	input capacitance	@ 1 MHz, T_{amb} = 25 °C, V_I = 0 V	[7]	-	-	15	pF
$I_{DD(oper)}$	operating supply current	f_{osc} = 12 MHz		-	-	11.5	mA
		f_{osc} = 40 MHz		-	-	50	mA
		Programming and erase mode		-	-	70	mA
$I_{DD(idle)}$	Idle mode supply current	f_{osc} = 12 MHz		-	-	8.5	mA
		f_{osc} = 40 MHz		-	-	42	mA
$I_{DD(pd)}$	Power-down mode supply current	minimum V_{DD} = 2 V		-	-	90	μA

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

[2] Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
a) Maximum I_{OL} per 8-bit port: 26 mA
b) Maximum I_{OL} total for all outputs: 71 mA
c) If I_{OL} exceeds the test condition, V_{OH} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

[3] Capacitive loading on Ports 0 and 2 may cause spurious noise to be superimposed on the V_{OL} of ALE and Ports 1 and 3. The noise due to external bus capacitance discharging into the Port 0 and 2 pins when the pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE pin may exceed 0.8 V. In such cases, it may be desirable to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.

[4] Load capacitance for Port 0, ALE and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF.

[5] Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the V_{DD} − 0.7 specification when the address bits are stabilizing.

[6] Pins of Ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V_I is approximately 2 V.

[7] Pin capacitance is characterized but not tested. $\overline{EA}$ = 25 pF (max).

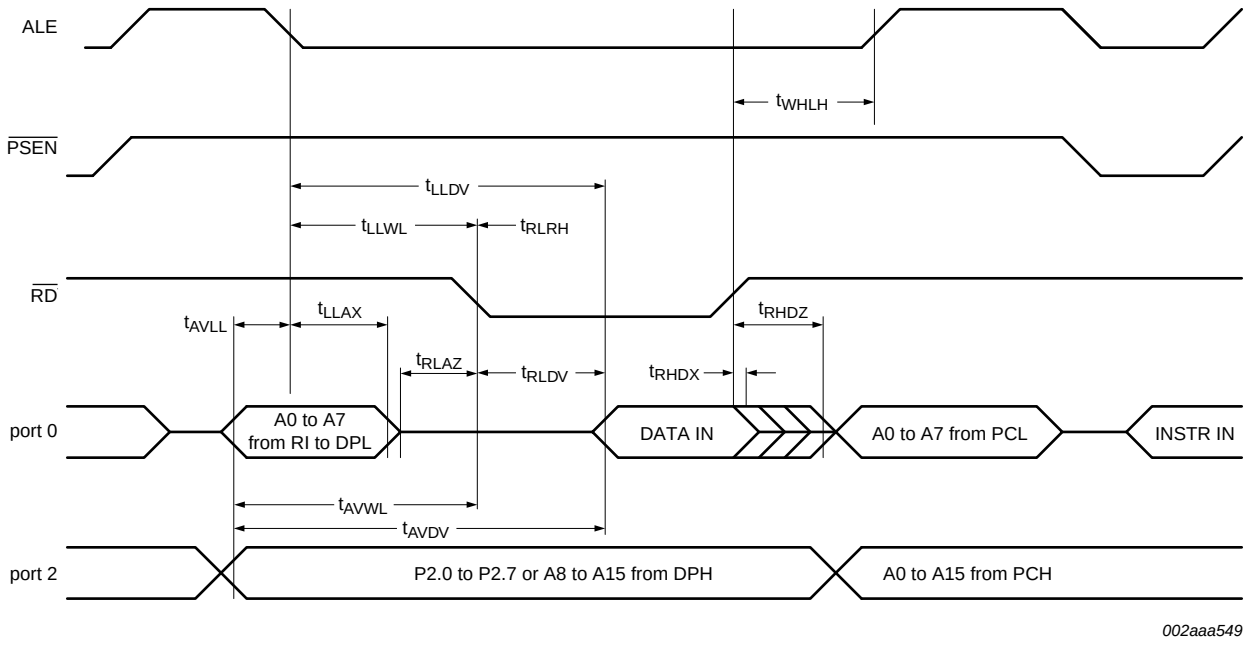

Fig 37. External data memory read cycle

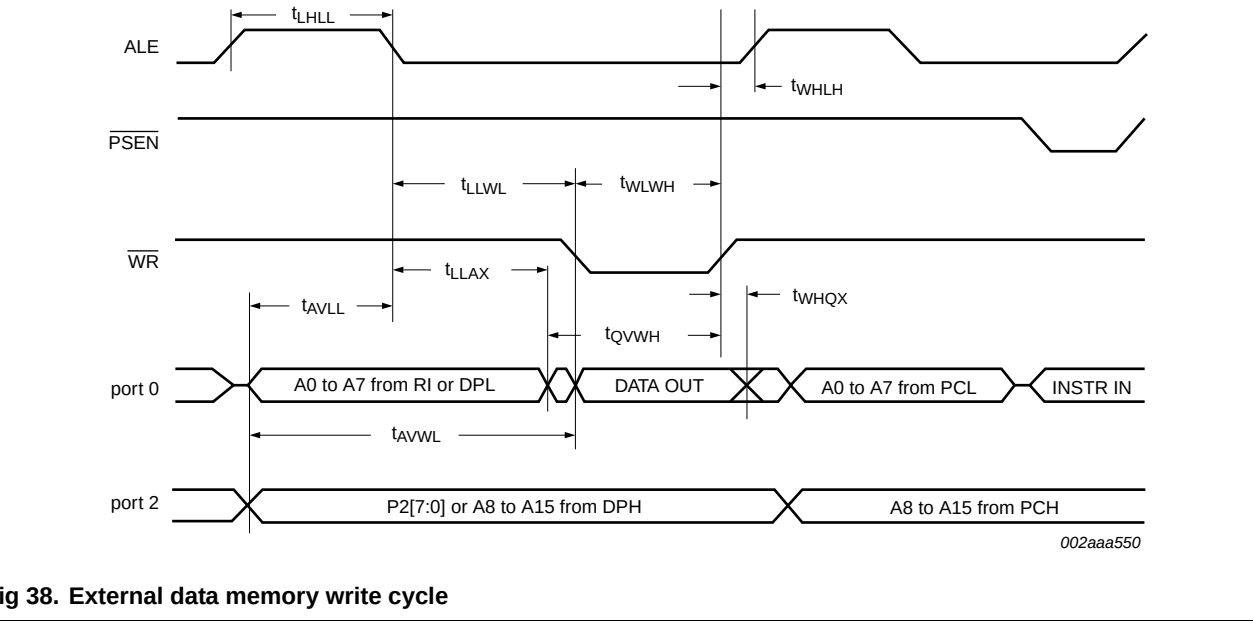

Fig 38. External data memory write cycle

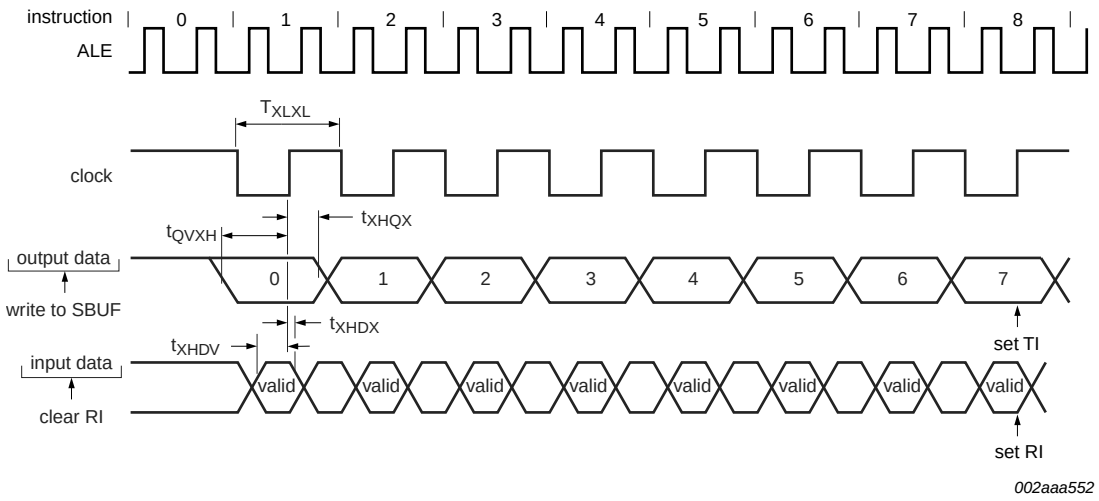

Fig 40. Shift register mode timing waveforms

Table 74. I²C-bus interface timing (12-clock mode)

Symbol	Parameter	Conditions	Input	Output	Unit
$t_{HD;STA}$	hold time (repeated) START condition		$\geq 14T_{cy(clk)}$	> 4.0[1]	µs
t_{LOW}	LOW period of the SCL clock		$\geq 16T_{cy(clk)}$	> 4.7[1]	µs
t_{HIGH}	HIGH period of the USCL clock		$\geq 14T_{cy(clk)}$	> 4.0[1]	µs
$t_{r(SCL)}$	SCL rise time		≤ 1	-[2]	µs
$t_{f(SCL)}$	SCL fall time		≤ 0.3	≤ 0.3[3]	µs
$t_{SU;DAT}$	data set-up time		≥ 250	$20T_{cy(clk)} - t_{r(SDA)}$	ns
t_{suDAT1}	data set-up time 1	before repeated START	≥ 250	> 1000[1]	ns
t_{suDAT2}	data set-up time 2	before STOP condition	≥ 250	$> 8T_{cy(clk)}$	ns
$t_{HD;DAT}$	data hold time		≥ 0	$> 8T_{cy(clk)} - t_{f(SCL)}$	ns
$t_{SU;STA}$	set-up time for a repeated START condition		$\geq 14T_{cy(clk)}$[1]	> 4.7[1]	µs
$t_{SU;STO}$	set-up time for STOP condition		$\geq 14T_{cy(clk)}$[1]	> 4.0[1]	µs
t_{BUF}	bus free time between a STOP and START condition		$\geq 14T_{cy(clk)}$[1]	> 4.7[1]	µs
$t_{r(SDA)}$	SDA rise time		≤ 0.3	≤ 0.3	µs
$t_{f(SDA)}$	SDA fall time		≤ 0.3	≤ 0.3[3]	µs

[1] At 100 kb/s. All other bit rates, this value is inversely proportional to the bit rate of 100 kb/s.

[2] Determined by the external bus capacitance and pull-up resistor. This must be < 1 µs.

[3] Spikes on SDA and SCL with a duration less than $3T_{cy(clk)}$ will be filtered out. Max capacitance on SDA and SCL = 400 pF.

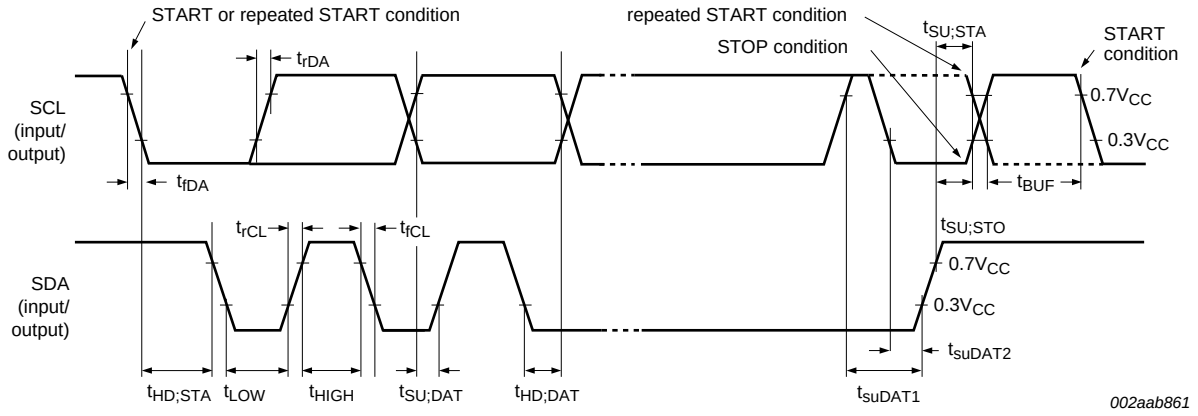

Fig 41. I²C-bus interface timing

Table 75. SPI interface timing

Symbol	Parameter	Conditions	Variable clock		f_{osc} = 18 MHz		Unit
			Min	Max	Min	Max	
f_{SPI}	SPI operating frequency		0	$T_{cy(clk)}$ / 4	0	10	MHz
T_{SPICYC}	SPI cycle time	see Figure 42, 43, 44, 45	$4T_{cy(clk)}$	-	222	-	ns
$t_{SPILEAD}$	SPI enable lead time	see Figure 44, 45	250	-	250	-	ns
t_{SPILAG}	SPI enable lag time	see Figure 44, 45	250	-	250	-	ns
$t_{SPICLKH}$	SPICLK HIGH time	see Figure 42, 43, 44, 45	$2T_{cy(clk)}$	-	111	-	ns
$t_{SPICLKL}$	SPICLK LOW time	see Figure 42, 43, 44, 45	$2T_{cy(clk)}$	-	111	-	ns
t_{SPIDSU}	SPI data set-up time	master or slave; see Figure 42, 43, 44, 45	100	-	100	-	ns
t_{SPIDH}	SPI data hold time	master or slave; see Figure 42, 43, 44, 45	100	-	100	-	ns
t_{SPIA}	SPI access time	see Figure 44, 45	0	80	0	80	ns
t_{SPIDIS}	SPI disable time	see Figure 44, 45	0	160	-	160	ns
t_{SPIDV}	SPI enable to output data valid time	see Figure 42, 43, 44, 45	-	111	-	111	ns
t_{SPIOH}	SPI output data hold time	see Figure 42, 43, 44, 45	0	-	0	-	ns
t_{SPIR}	SPI rise time	see Figure 42, 43, 44, 45					
	SPI outputs (SPICLK, MOSI, MISO)		-	100	-	100	ns
	SPI inputs (SPICLK, MOSI, MISO, $\overline{SS}$)		-	2000	-	2000	ns
t_{SPIF}	SPI fall time	see Figure 42, 43, 44, 45					
	SPI outputs (SPICLK, MOSI, MISO)		-	100	-	100	ns
	SPI inputs (SPICLK, MOSI, MISO, $\overline{SS}$)		-	2000	-	2000	ns

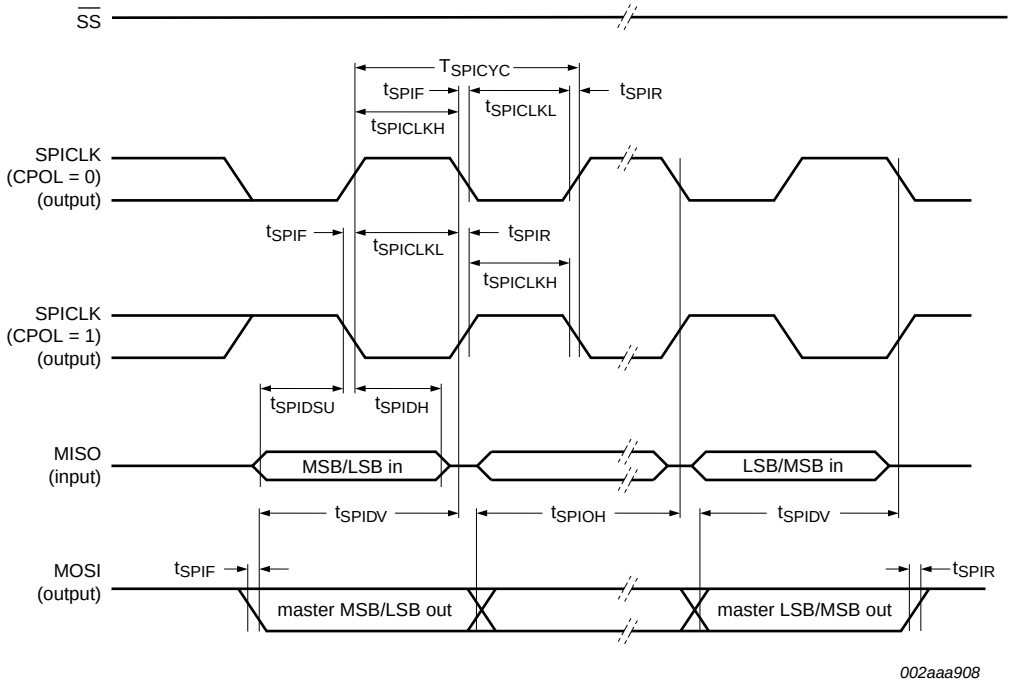

Fig 42. SPI master timing (CPHA = 0)

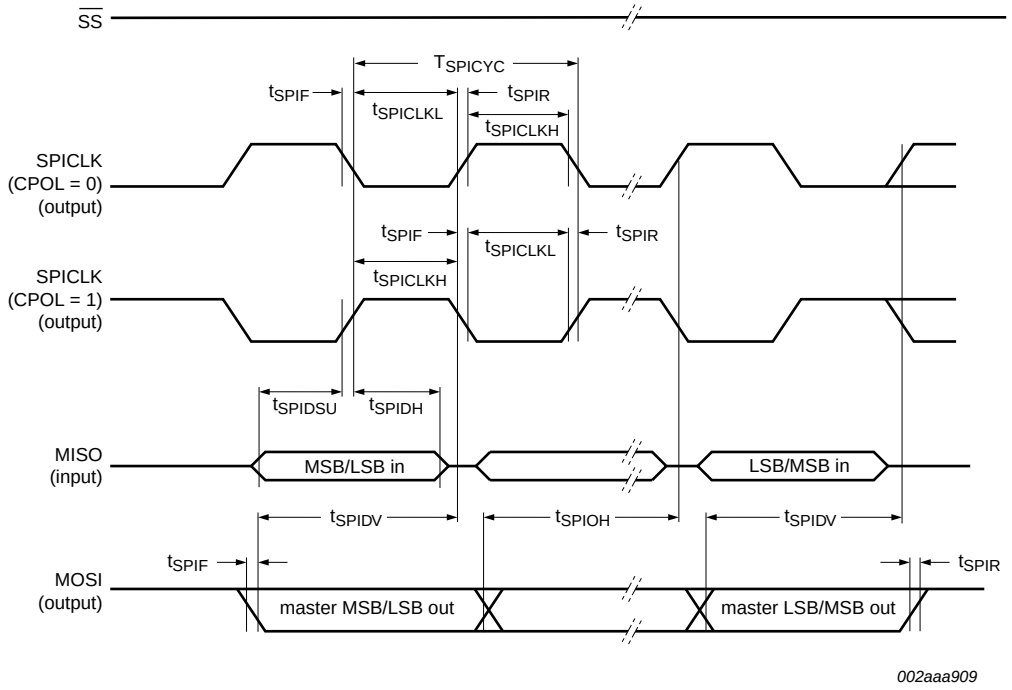

Fig 43. SPI master timing (CPHA = 1)

12. Revision history

Table 77. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
P89V660_662_664 v.3.1	20111017	Product data sheet	-	P89V660_662_664 v.3
Modifications:	• Table 3 “Pin description”: Added Table note 1. • Updated Equation 2 and Equation 3. • Removed type numbers P89V660FA and P89V660FBC from Table 1 and Table 2 due to EOL.			
P89V660_662_664 v.3	20081110	Product data sheet	-	P89V660_662_664 v.2
Modifications:	• Section 2.2 “Additional features”: corrected 6-clock/12-clock mode information.			
P89V660_662_664 v.2	20080129	Product data sheet	-	P89V660_662_664 v.1
P89V660_662_664 v.1	20070502	Product data sheet	-	-

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

13.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

14. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**