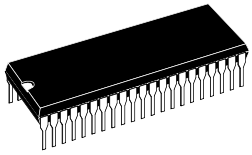
ST72E121
ST72T121

8-BIT MCU WITH 8 TO 16K OTP/EPROM, 384 TO 512 BYTES RAM, WDG, SCI, SPI AND 2 TIMERS

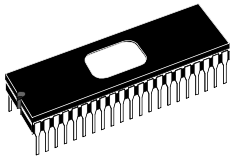
DATASHEET

- User Program Memory (OTP/EPROM): 8 to 16K bytes
- Data RAM: 384 to 512 bytes including 256 bytes of stack
- Master Reset and Power-On Reset
- Low Voltage Detector (LVD) Reset option
- Run and Power Saving modes
- 32 multifunctional bidirectional I/O lines:
  - 9 programmable interrupt inputs
  - 4 high sink outputs
  - 13 alternate functions
  - EMI filtering
- Software or Hardware Watchdog (WDG)
- Two 16-bit Timers, each featuring:
  - 2 Input Captures [1)]
  - 2 Output Compares [1)]
  - External Clock input (on Timer A)
  - PWM and Pulse Generator modes
- Synchronous Serial Peripheral Interface (SPI)
- Asynchronous Serial Communications Interface (SCI)
- 8-bit Data Manipulation
- 63 basic Instructions and 17 main Addressing Modes
- 8 x 8 Unsigned Multiply Instruction
- True Bit Manipulation
- Complete Development Support on DOS/ WINDOWS™ Real-Time Emulator
- Full Software Package on DOS/WINDOWS™ (C-Compiler, Cross-Assembler, Debugger)

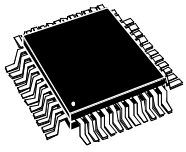
Note: 1. One only on Timer A.



PSDIP42



CSDIP42W



TQFP44

(See ordering information at the end of datasheet)

Device Summary

Features	ST72T121J2	ST72T121J4
Program Memory - bytes	8K	16K
RAM (stack) - bytes	384 (256)	512 (256)
Peripherals	Watchdog, Timers, SPI, SCI and optional Low Voltage Detector Reset	
Operating Supply	3 to 5.5 V	
CPU Frequency	8MHz max (16MHz oscillator) - 4MHz max over 85°C	
Temperature Range	- 40°C to + 125°C	
Package	TQFP44 - SDIP42	
OTP/EPROM Devices	ST72T121J4/ST72E121J4	

Note: The ROM versions are supported by the ST72124 family.

Table of Contents

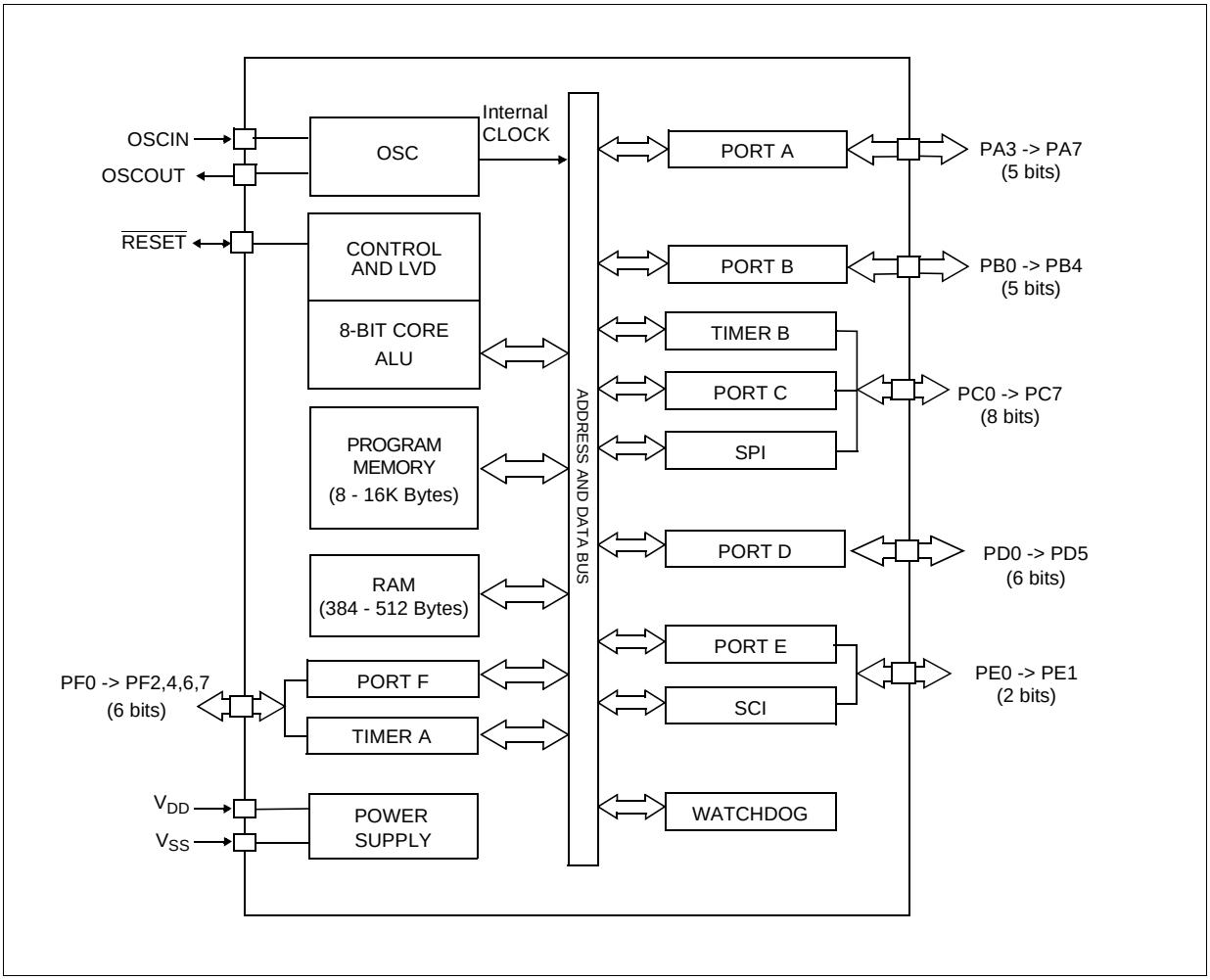
Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

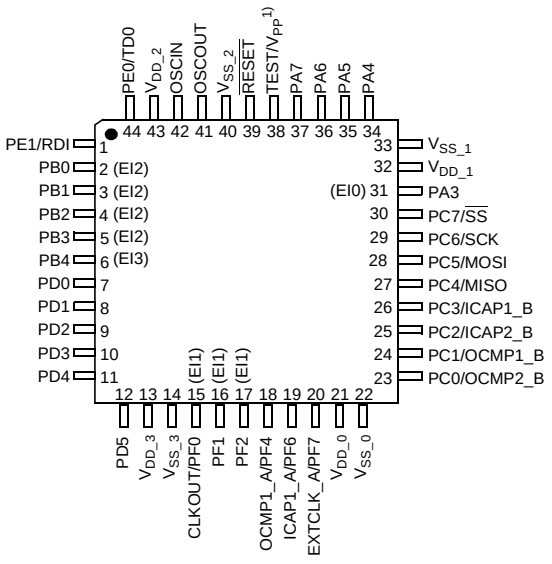
The ST72T121 HCMOS Microcontroller Unit (MCU) is a member of the ST7 family. The device is based on an industry-standard 8-bit core and features an enhanced instruction set. The device is normally operated at a 16 MHz oscillator frequency. Under software control, the ST72T121 may be placed in either Wait, Slow or Halt modes, thus reducing power consumption. The enhanced instruction set and addressing modes afford real programming potential. In addition to standard 8-bit data management, the ST72T121 features true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes on the whole memory. The device includes a low consumption and fast start on-chip oscillator, CPU, program memory (OTP/EPROM versions), RAM, 32 I/O lines, a Low Voltage Detector (LVD) and the following on-chip peripherals: industry standard synchronous SPI and asynchronous SCI serial interfaces, digital Watchdog, two independent 16-bit Timers, one featuring an External Clock Input, and both featuring Pulse Generator capabilities, 2 Input Captures and 2 Output Compares (only 1 Input Capture and 1 Output Compare on Timer A).

Figure 1. ST72T121 Block Diagram



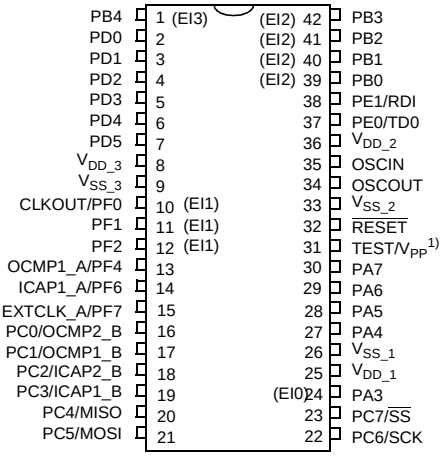
1.2 PIN DESCRIPTION

Figure 2. 44-Pin Thin QFP Package Pinout



1. V_{PP} on EPROM/OTP only

Figure 3. 42-Pin Shrink DIP Package Pinout



1. V_{PP} on EPROM/OTP only

Table 1. ST72T121Jx Pin Description

Pin n° QFP44	Pin n° SDIP42	Pin Name	Type	Description	Remarks
1	38	PE1/RDI	I/O	Port E1 or SCI Receive Data In	
2	39	PB0	I/O	Port B0	External Interrupt: EI2
3	40	PB1	I/O	Port B1	External Interrupt: EI2
4	41	PB2	I/O	Port B2	External Interrupt: EI2
5	42	PB3	I/O	Port B3	External Interrupt: EI2
6	1	PB4	I/O	Port B4	External Interrupt: EI3
7	2	PD0	I/O	Port D0	
8	3	PD1	I/O	Port D1	
9	4	PD2	I/O	Port D2	
10	5	PD3	I/O	Port D3	
11	6	PD4	I/O	Port D4	
12	7	PD5	I/O	Port D5	
13	8	V_{DD_3}	S	Main Power Supply	
14	9	V_{SS_3}	S	Ground	
15	10	PF0/CLKOUT	I/O	Port F0 or CPU Clock Output	External Interrupt: EI1
16	11	PF1	I/O	Port F1	External Interrupt: EI1
17	12	PF2	I/O	Port F2	External Interrupt: EI1
18	13	PF4/OCMP1_A	I/O	Port F4 or Timer A Output Compare 1	
19	14	PF6/ICAP1_A	I/O	Port F6 or Timer A Input Capture 1	
20	15	PF7/EXTCLK_A	I/O	Port F7 or External Clock on Timer A	
21		V_{DD_0}	S	Main power supply	
22		V_{SS_0}	S	Ground	
23	16	PC0/OCMP2_B	I/O	Port C0 or Timer B Output Compare 2	
24	17	PC1/OCMP1_B	I/O	Port C1 or Timer B Output Compare 1	
25	18	PC2/ICAP2_B	I/O	Port C2 or Timer B Input Capture 2	
26	19	PC3/ICAP1_B	I/O	Port C3 or Timer B Input Capture 1	
27	20	PC4/MISO	I/O	Port C4 or SPI Master In / Slave Out Data	
28	21	PC5/MOSI	I/O	Port C5 or SPI Master Out / Slave In Data	
29	22	PC6/SCK	I/O	Port C6 or SPI Serial Clock	
30	23	PC7/$\overline{SS}$	I/O	Port C7 or SPI Slave Select	
31	24	PA3	I/O	Port A3	External Interrupt: EI0
32	25	V_{DD_1}	S	Main power supply	
33	26	V_{SS_1}	S	Ground	
34	27	PA4	I/O	Port A4	High Sink
35	28	PA5	I/O	Port A5	High Sink
36	29	PA6	I/O	Port A6	High Sink
37	30	PA7	I/O	Port A7	High Sink
38	31	TEST/V_{PP} [1)]	S	Test mode pin. In the EPROM programming mode, this pin acts as the programming voltage input V_{PP}.	This pin must be tied low in user mode
39	32	$\overline{RESET}$	I/O	Bidirectional. Active low. Top priority non maskable interrupt.	
40	33	V_{SS_2}	S	Ground	
41	34	OSCOUT	O	Input/Output Oscillator pin. These pins connect a parallel-resonant crystal, or an external source to the on-chip oscillator.	
42	35	OSCIN	I		
43	36	V_{DD_2}	S	Main power supply	
44	37	PE0/TDO	I/O	Port E0 or SCI Transmit Data Out	

Note 1: V_{PP} on EPROM/OTP only.

1.3 EXTERNAL CONNECTIONS

The following figure shows the recommended external connections for the device.

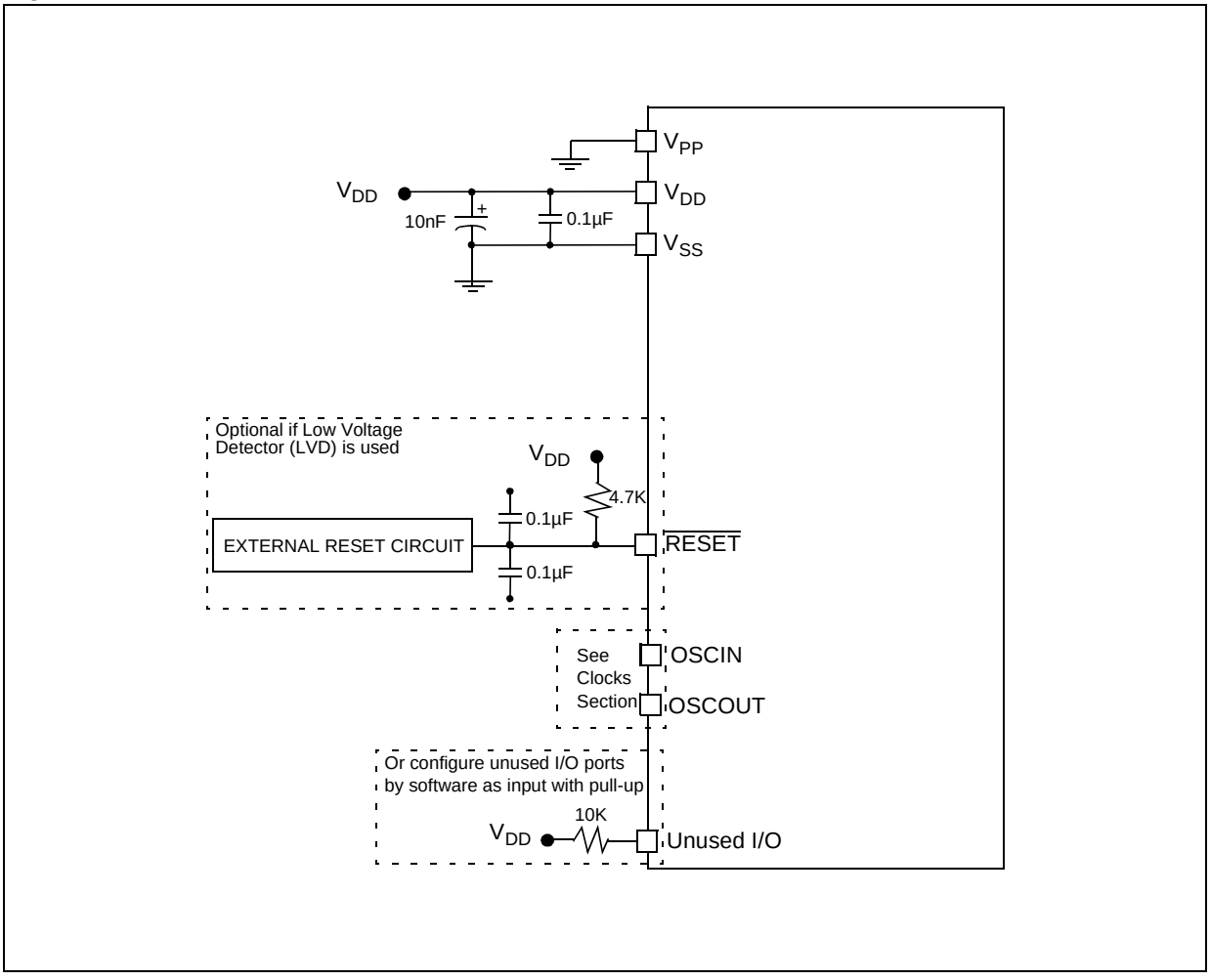
The V_{PP} pin is only used for programming OTP and EPROM devices and must be tied to ground in user mode.

The 10 nF and 0.1 µF decoupling capacitors on the power supply lines are a suggested EMC performance/cost tradeoff.

The external reset network is intended to protect the device against parasitic resets, especially in noisy environments.

Unused I/Os should be tied high to avoid any unnecessary power consumption on floating lines. An alternative solution is to program the unused ports as inputs with pull-up.

Figure 4. Recommended External Connections



1.4 MEMORY MAP

Figure 5. Program Memory Map

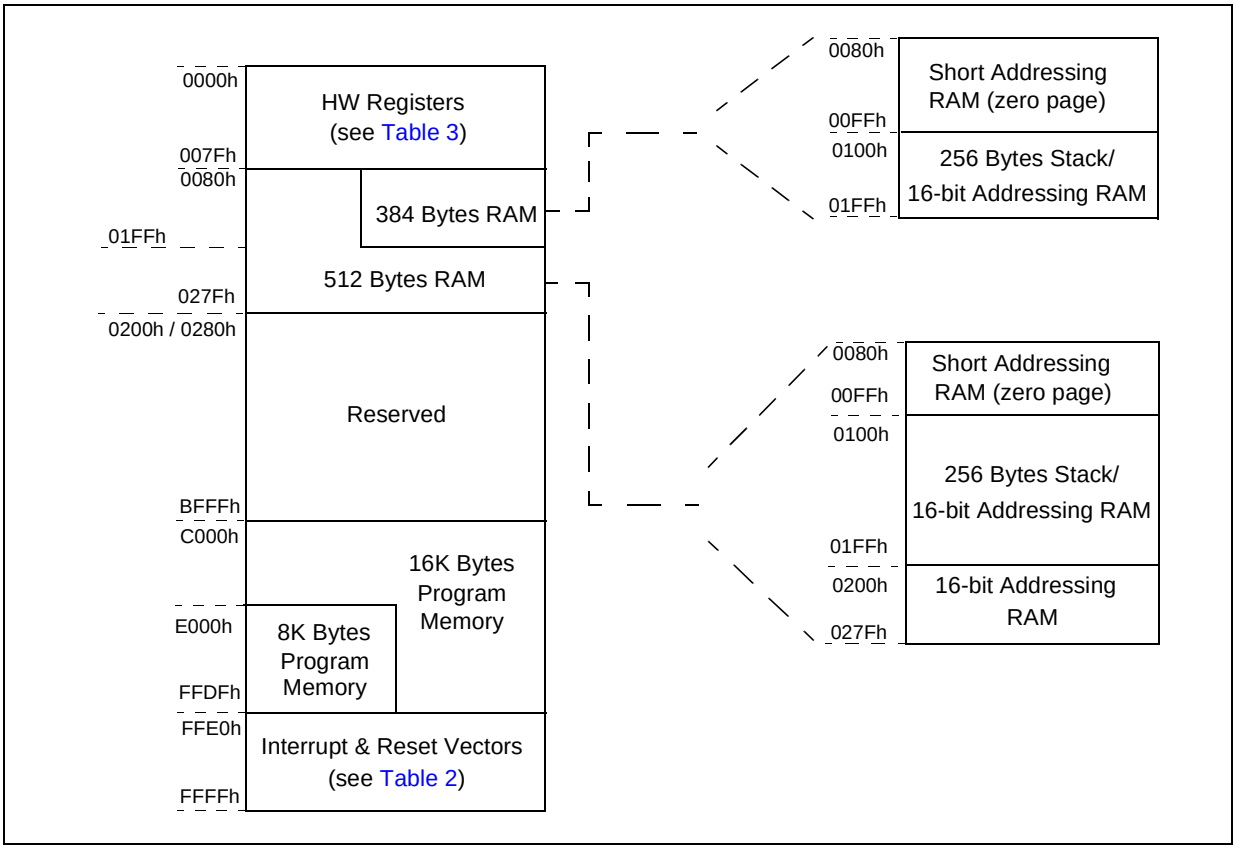


Table 2. Interrupt Vector Map

Vector Address	Description	Remarks
FFE0-FFE1h	Not Used	
FFE2-FFE3h	Not Used	
FFE4-FFE5h	Not Used	Internal Interrupt
FFE6-FFE7h	SCI Interrupt Vector	Internal Interrupt
FFE8-FFE9h	TIMER B Interrupt Vector	Internal Interrupt
FFEA-FFEBh	TIMER A Interrupt Vector	Internal Interrupt
FFEC-FFEDh	SPI interrupt vector	Internal Interrupt
FFEE-FFEFh	Not Used	
FFF0-FFF1h	External Interrupt Vector EI3 (PB4)	External Interrupt
FFF2-FFF3h	External Interrupt Vector EI2 (PB0:PB3)	External Interrupt
FFF4-FFF5h	External Interrupt Vector EI1 (PF0:PF2)	External Interrupt
FFF6-FFF7h	External Interrupt Vector EI0 (PA3)	External Interrupt
FFF8-FFF9h	Not Used	
FFFA-FFFBh	Not Used	
FFFC-FFFDh	TRAP (software) Interrupt Vector	CPU Interrupt
FFFE-FFFFh	RESET Vector	

Table 3. Hardware Register Memory Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h	Port A	PADR	Data Register	00h	R/W
0001h		PADDR	Data Direction Register	00h	R/W
0002h		PAOR	Option Register	00h	R/W [1]
0003h	Reserved Area (1 byte)				
0004h	Port C	PCDR	Data Register	00h	R/W
0005h		PCDDR	Data Direction Register	00h	R/W
0006h		PCOR	Option Register	00h	R/W
0007h	Reserved Area (1 byte)				
0008h	Port B	PBDR	Data Register	00h	R/W
0009h		PBDDR	Data Direction Register	00h	R/W
000Ah		PBOR	Option Register	00h	R/W [1]
000Bh	Reserved Area (1 byte)				
000Ch	Port E	PEDR	Data Register	00h	R/W
000Dh		PEDDR	Data Direction Register	00h	R/W
000Eh		PEOR	Option Register	0Ch	R/W [1]
000Fh	Reserved Area (1 byte)				
0010h	Port D	PDDR	Data Register	00h	R/W
0011h		PDDDR	Data Direction Register	00h	R/W
0012h		PDOR	Option Register	00h	R/W [1]
0013h	Reserved Area (1 byte)				
0014h	Port F	PFDR	Data Register	00h	R/W
0015h		PFDDR	Data Direction Register	00h	R/W
0016h		PFOR	Option Register	28h	R/W [1]
0017h to 001Fh	Reserved Area (9 bytes)				
0020h		MISCR	Miscellaneous Register	00h	
0021h	SPI	SPIDR	SPI Data I/O Register	xxh	R/W
0022h		SPICR	SPI Control Register	xxh	R/W
0023h		SPISR	SPI Status Register	00h	Read Only
0024h to 0029h	Reserved Area (6 bytes)				
002Ah	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh		WDGSR	Watchdog Status Register	00h	R/W [3]
002Ch to 0030h	Reserved Area (5 bytes)				

Address	Block	Register Label	Register Name	Reset Status	Remarks
0031h		TACR2	Control Register2	00h	R/W
0032h		TACR1	Control Register1	00h	R/W
0033h		TASR	Status Register	xxh	Read Only
0034h-0035h		TAIC1HR	Input Capture1 High Register	xxh	Read Only
		TAIC1LR	Input Capture1 Low Register	xxh	Read Only
0036h-0037h		TAOC1HR	Output Compare1 High Register	80h	R/W
		TAOC1LR	Output Compare1 Low Register	00h	R/W
0038h-0039h	Timer A	TACHR	Counter High Register	FFh	Read Only
		TACLR	Counter Low Register	FCh	Read Only
003Ah-003Bh		TAACHR	Alternate Counter High Register	FFh	Read Only
		TAACLR	Alternate Counter Low Register	FCh	Read Only
003Ch-003Dh		TAIC2HR	Input Capture2 High Register	xxh	Read Only[2]
		TAIC2LR	Input Capture2 Low Register	xxh	Read Only[2]
003Eh-003Fh		TAOC2HR	Output Compare2 High Register	80h	R/W[2]
		TAOC2LR	Output Compare2 Low Register	00h	R/W[2]
0040h	Reserved Area (1 byte)				
0041h		TBCR2	Control Register2	00h	R/W
0042h		TBCR1	Control Register1	00h	R/W
0043h		TBSR	Status Register	xxh	Read Only
0044h-0045h		TBIC1HR	Input Capture1 High Register	xxh	Read Only
		TBIC1LR	Input Capture1 Low Register	xxh	Read Only
0046h-0047h		TBOC1HR	Output Compare1 High Register	80h	R/W
		TBOC1LR	Output Compare1 Low Register	00h	R/W
0048h-0049h	Timer B	TBCHR	Counter High Register	FFh	Read Only
		TBCLR	Counter Low Register	FCh	Read Only
004Ah-004Bh		TBACHR	Alternate Counter High Register	FFh	Read Only
		TBACLR	Alternate Counter Low Register	FCh	Read Only
004Ch-004Dh		TBIC2HR	Input Capture2 High Register	xxh	Read Only
		TBIC2LR	Input Capture2 Low Register	xxh	Read Only
004Eh-004Fh		TBOC2HR	Output Compare2 High Register	80h	R/W
		TBOC2LR	Output Compare2 Low Register	00h	R/W
0050h		SCISR	SCI Status Register	C0h	Read Only
0051h		SCIDR	SCI Data Register	xxh	R/W
0052h		SCIBRR	SCI Baud Rate Register	00x----xb	R/W
0053h	SCI	SCICR1	SCI Control Register 1	xxh	R/W
0054h		SCICR2	SCI Control Register 2	00h	R/W
0055h		SCIERPR	SCI Extended Receive Prescaler Register	00h	R/W
0056h			Reserved	---	Reserved
0057h		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W
0058h to 007Fh	Reserved Area (40 bytes)				

Notes:
1. The bits corresponding to unavailable pins are forced to 1 by hardware, this affects the reset status value.
2. External pin not available.
3. Not used in versions without Low Voltage Detector Reset.

1.5 OPTION BYTE

The user has the option to select software watchdog or hardware watchdog (see description in the Watchdog chapter). When programming EPROM or OTP devices, this option is selected in a menu by the user of the EPROM programmer before burning the EPROM/OTP. The Option Byte is located in a non-user map. No address has to be specified. The Option Byte is at FFh after UV erasure and must be properly programmed to set desired options.

OPTBYTE

7							0
-	-	-	-	b3	b2	-	WDG

Bit 7:4 = Not used

Bit 3 = Reserved, must be cleared.

Bit 2 = Reserved, must be set on ST72T121N devices and must be cleared on ST72T121J devices.

Bit 1 = Not used

Bit 0 = **WDG** *Watchdog disable*
0: The Watchdog is enabled after reset (Hardware Watchdog).
1: The Watchdog is not enabled after reset (Software Watchdog).

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

2.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- 8 MHz CPU internal frequency
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

2.3 CPU REGISTERS

The 6 CPU registers shown in Figure 6 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

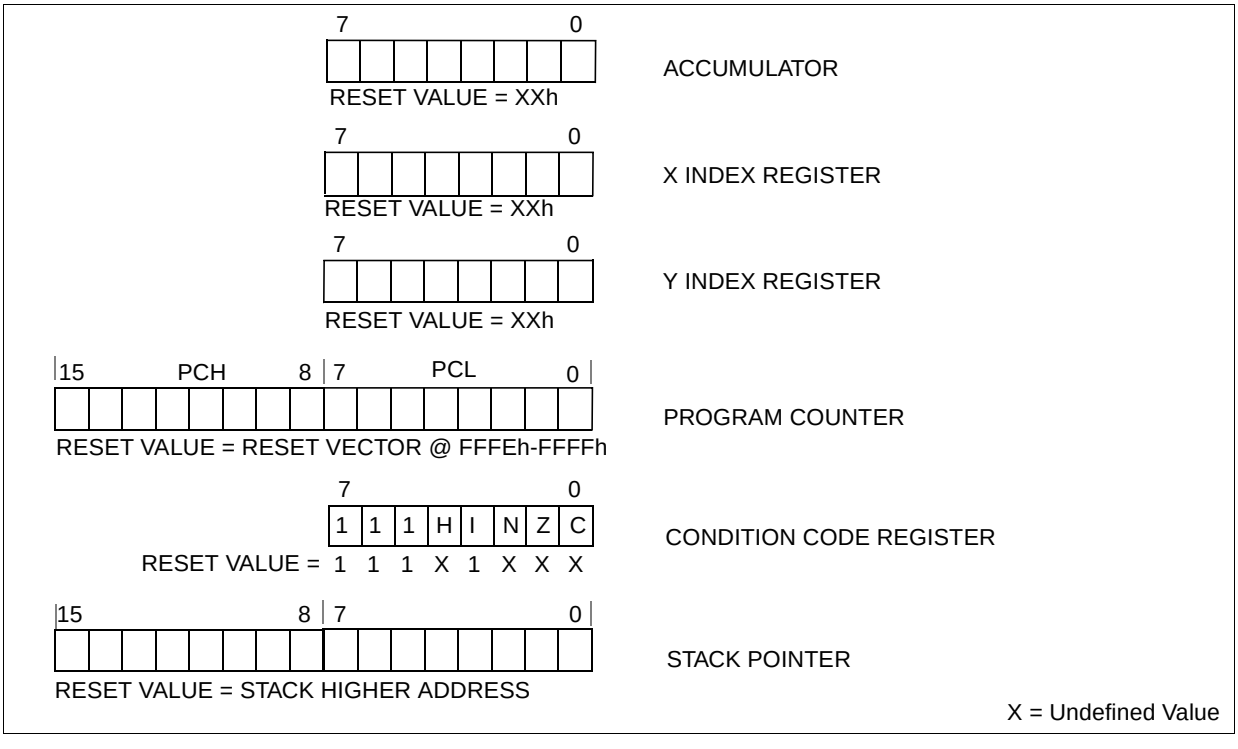
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 6. CPU Registers



CENTRAL PROCESSING UNIT (Cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = **H** *Half carry*.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.
0: No half carry has occurred.
1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 3 = **I** *Interrupt mask*.

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
0: Interrupts are enabled.
1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptable because the I bit is set by hardware when you enter it and reset by the IRET instruction at the end of the interrupt routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = **N** *Negative*.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow.*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01FFh

15							8
0	0	0	0	0	0	0	1

7							0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 7).

Since the stack is 256 bytes deep, the 8th most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

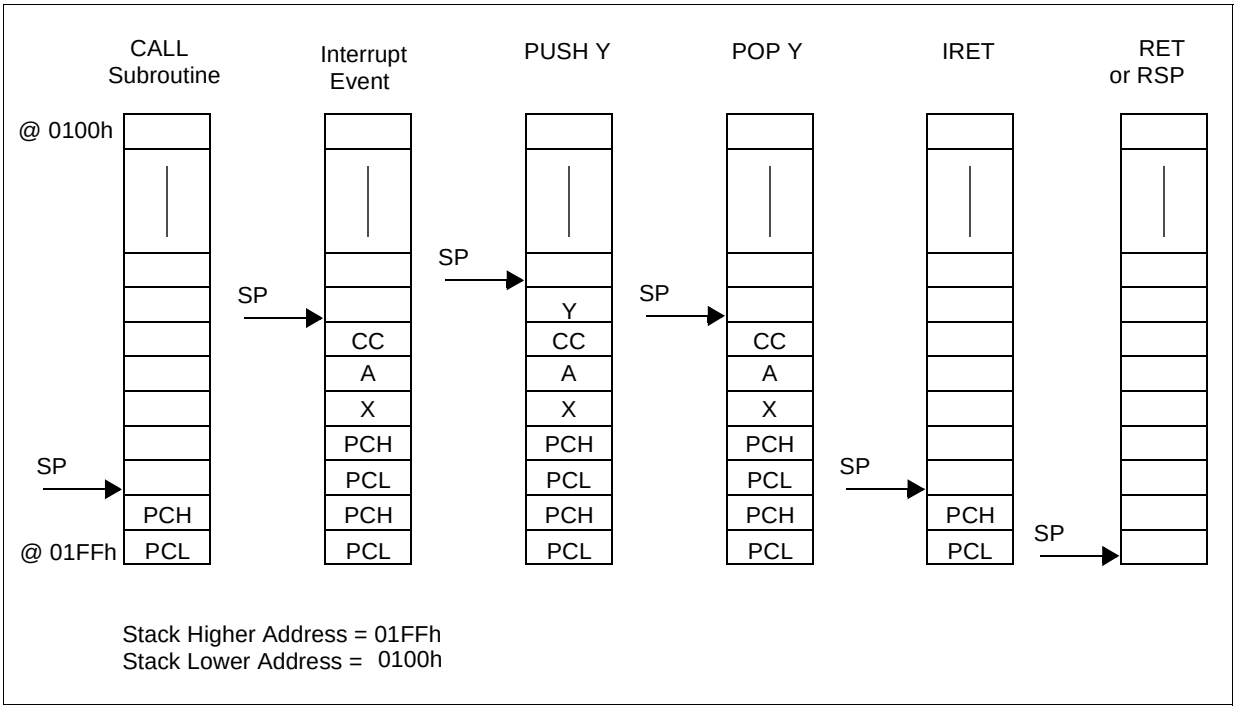
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 7.

– When an interrupt is received, the SP is decremented and the context is pushed on the stack.

– On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 7. Stack Manipulation Example



3 CLOCKS, RESET, INTERRUPTS & POWER SAVING MODES

3.1 CLOCK SYSTEM

3.1.1 General Description

The MCU accepts either a crystal or ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}). The external Oscillator clock is first divided by 2, and an additional division factor of 2, 4, 8, or 16 can be applied, in Slow Mode, to reduce the frequency of the f_{CPU}; this clock signal is also routed to the on-chip peripherals. The CPU clock signal consists of a square wave with a duty cycle of 50%.

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc}. The circuit shown in Figure 9 is recommended when using a crystal, and Table 4 lists the recommended capacitance and feedback resistance values. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilisation time.

Use of an external CMOS oscillator is recommended when crystals outside the specified frequency ranges are to be used.

3.1.2 External Clock

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected, as shown on Figure 8.

Table 4 Recommended Values for 16 MHz Crystal Resonator (C_0 < 7pF)

R_{SMAX}	40 Ω	60 Ω	150 Ω
C_{OSCIN}	56pF	47pF	22pF
C_{OSCOUT}	56pF	47pF	22pF

R_{SMAX}: Parasitic series resistance of the quartz crystal (upper limit).

C_0: Parasitic shunt capacitance of the quartz crystal (upper limit 7pF).

C_{OSCOUT}, C_{OSCIN}: Maximum total capacitance on pins OSCIN and OSCOUT (the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device).

Figure 8. External Clock Source Connections

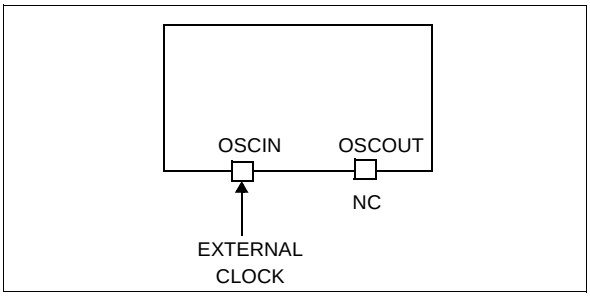


Figure 9. Crystal/Ceramic Resonator

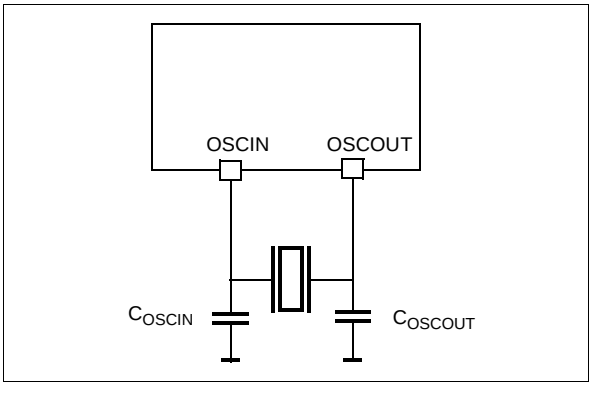
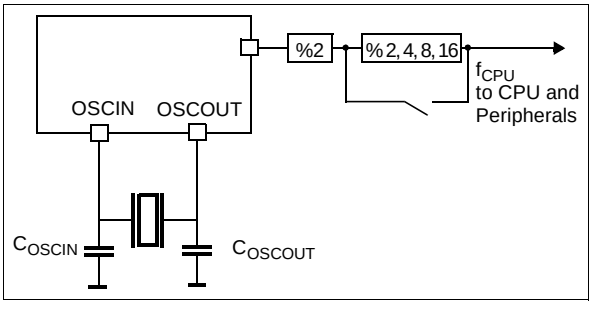


Figure 10. Clock Prescaler Block Diagram



3.2 RESET

3.2.1 Introduction

There are four sources of Reset:

– $\overline{\text{RESET}}$ pin (external source)

– Power-On Reset (Internal source)

– WATCHDOG (Internal Source)

– Low Voltage Detection Reset (internal source)

The Reset Service Routine vector is located at address FFFEh-FFFFh.

3.2.2 External Reset

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated pull-up resistor. When one of the internal Reset sources is active, the Reset pin is driven low for a duration of t_{RESET} to reset the whole application.

3.2.3 Reset Operation

The duration of the Reset state is a minimum of 4096 internal CPU Clock cycles. During the Reset state, all I/Os take their reset value.

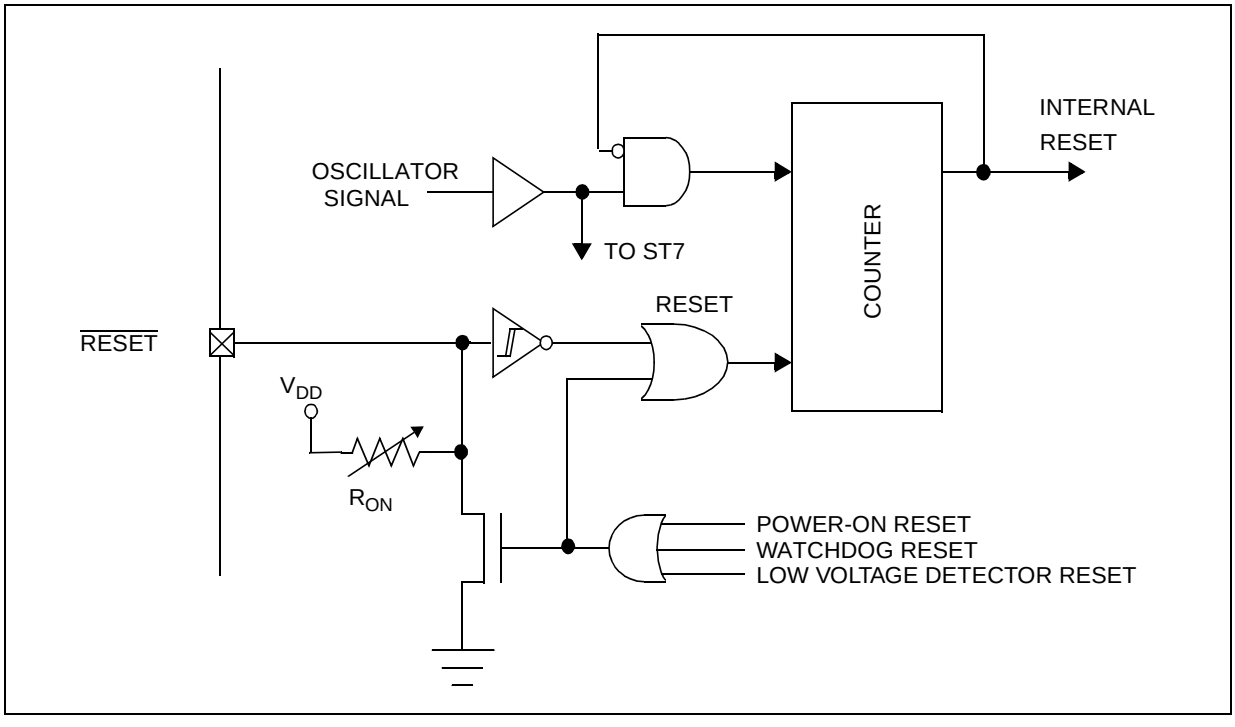
A Reset signal originating from an external source must have a duration of at least t_{PULSE} in order to be recognised. This detection is asynchronous and therefore the MCU can enter Reset state even in Halt mode.

At the end of the Reset cycle, the MCU may be held in the Reset state by an External Reset signal. The $\overline{\text{RESET}}$ pin may thus be used to ensure V_{DD} has risen to a point where the MCU can operate correctly before the user program is run. Following a Reset event, or after exiting Halt mode, a 4096 CPU Clock cycle delay period is initiated in order to allow the oscillator to stabilise and to ensure that recovery has taken place from the Reset state.

In the high state, the $\overline{\text{RESET}}$ pin is connected internally to a pull-up resistor (R_{ON}). This resistor can be pulled low by external circuitry to reset the device.

The $\overline{\text{RESET}}$ pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to use the external connections shown in Figure 4.

Figure 11. Reset Block Diagram



RESET (Cont'd)

3.2.4 Low Voltage Detector Reset

The on-chip Low Voltage Detector (LVD) generates a static reset when the supply voltage is below a reference value. The LVD functions both during power-on as well as when the power supply drops (brown-out). The reference value for a voltage drop is lower than the reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD Reset circuitry generates a reset when V_{DD} is below:

V_{LVDUP} when V_{DD} is rising

$V_{LVDDOWN}$ when V_{DD} is falling

Provided the minimun V_{DD} value (guaranteed for the oscillator frequency) is above $V_{LVDDOWN}$, the MCU can only be in two modes:

- under full software control or

- in static safe reset

In this condition, secure operation is always ensured for the application without the need for external reset hardware.

During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

In noisy environments, the power supply may drop for short periods and cause the Low Voltage Detector to generate a Reset too frequently. In such cases, it is recommended to use devices without the LVD Reset option and to rely on the watchdog function to detect application runaway conditions.

Figure 12. Low Voltage Detector Reset Function

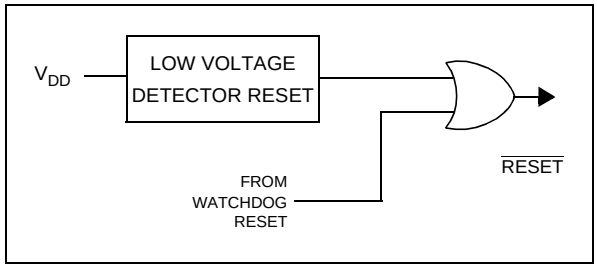
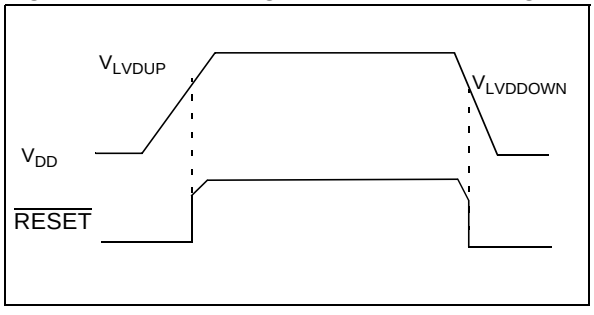
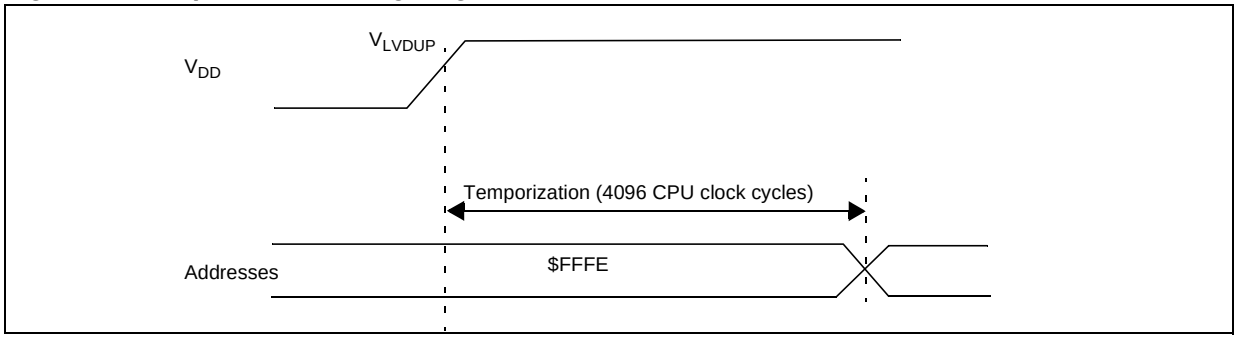


Figure 13. Low Voltage Detector Reset Signal



Note: See electrical characteristics for values of V_{LVDUP} and $V_{LVDDOWN}$

Figure 14. Temporization timing diagram after an internal Reset



3.3 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in the Interrupt Mapping Table and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 15.
The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

– Normal processing is suspended at the end of the current instruction execution.

– The PC, X, A and CC registers are saved onto the stack.

– The I bit of the CC register is set to prevent additional interrupts.

– The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping Table for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority management

By default, a servicing interrupt can not be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping Table).

Non Maskable Software Interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on Figure 15.

Interrupts and Low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific mentioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT“ column in the Interrupt Mapping Table).

External Interrupts

External interrupt vectors can be loaded in the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the Halt low power mode.

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

If several input pins, connected to the same interrupt vector, are configured as interrupts, their signals are logically ANDed before entering the edge/level detection block.

Warning: The type of sensitivity defined in the Miscellaneous or Interrupt register (if available) applies to the EI source. In case of an ANDed source (as described on the I/O ports section), a low level on an I/O pin configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

Peripheral Interrupts

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both:

– The I bit of the CC register is cleared.

– The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by:

– writing “0” to the corresponding bit in the status register or

– an access to the status register while the flag is set followed by a read or write of an associated register.

Note: the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being enabled) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

Figure 15. Interrupt Processing Flowchart

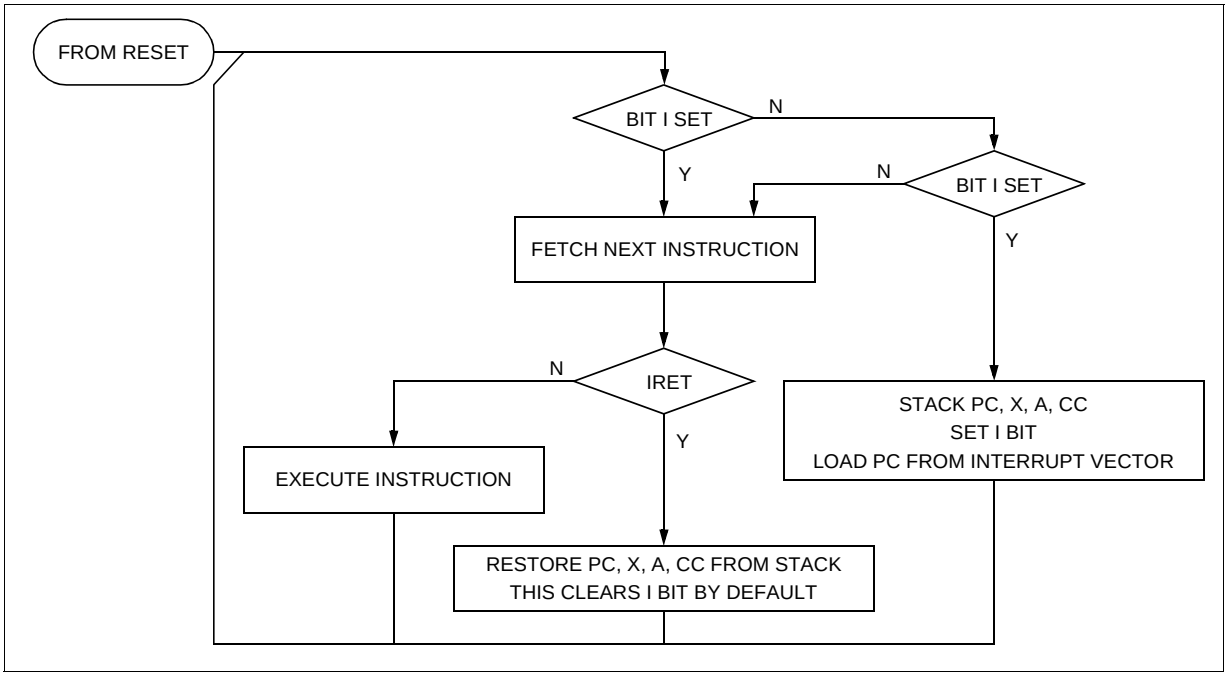


Table 5. Interrupt Mapping

Source Block	Description	Register Label	Flag	Exit from HALT	Vector Address	Priority Order
RESET	Reset	N/A	N/A	yes	FFFEh-FFFFh	Highest Priority
TRAP	Software	N/A	N/A	no	FFFCh-FFFDh	
NOT USED					FFFAh-FFFBh	
NOT USED					FFF8h-FFF9h	
EI0	Ext. Interrupt (Ports PA0:PA3)	N/A	N/A	yes	FFF6h-FFF7h	
EI1	Ext. Interrupt (Ports PF0:PF2)	N/A	N/A		FFF4h-FFF5h	
EI2	Ext. Interrupt (Ports PB0:PB3)	N/A	N/A		FFF2h-FFF3h	
EI3	Ext. Interrupt (Ports PB4:PB7)	N/A	N/A		FFF0h-FFF1h	
NOT USED					FFEEh-FFEFh	
SPI	Transfer Complete	SPISR	SPIF	no	FFECh-FFEDh	
	Mode Fault		MODF			
TIMER A	Input Capture 1	TASR	ICF1_A		FFEAh-FFEBh	
	Output Compare 1		OCF1_A			
	Input Capture 2		ICF2_A			
	Output Compare 2		OCF2_A			
	Timer Overflow		TOF_A			
TIMER B	Input Capture 1	TBSR	ICF1_B		FFE8h-FFE9h	
	Output Compare 1		OCF1_B			
	Input Capture 2		ICF2_B			
	Output Compare 2		OCF2_B			
	Timer Overflow		TOF_B			
SCI	Transmit Buffer Empty	SCISR	TDRE		FFE6h-FFE7h	
	Transmit Complete		TC			
	Receive Buffer Full		RDRF			
	Idle Line Detect		IDLE			
	Overrun		OR			Lowest Priority
NOT USED					FFE4h-FFE5h	
NOT USED					FFE2h-FFE3h	
NOT USED					FFE0h-FFE1h	

3.4 POWER SAVING MODES

3.4.1 Introduction

There are three Power Saving modes. Slow Mode is selected by setting the relevant bits in the Miscellaneous register. Wait and Halt modes may be entered using the WFI and HALT instructions.

3.4.2 Slow Mode

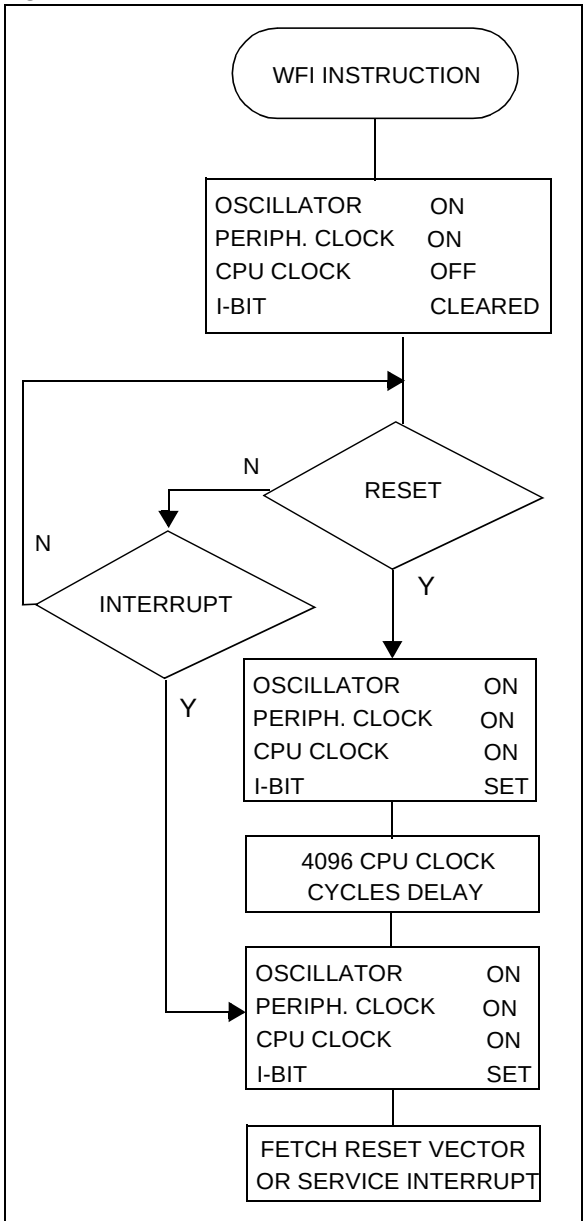
In Slow mode, the oscillator frequency can be divided by a value defined in the Miscellaneous Register. The CPU and peripherals are clocked at this lower frequency. Slow mode is used to reduce power consumption, and enables the user to adapt clock frequency to available supply voltage.

3.4.3 Wait Mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU. All peripherals remain active. During Wait mode, the I bit (CC Register) is cleared, so as to enable all interrupts. All other registers and memory remain unchanged. The MCU will remain in Wait mode until an Interrupt or Reset occurs, whereupon the Program Counter branches to the starting address of the Interrupt or Reset Service Routine.
The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to Figure 16 below.

Figure 16. WAIT Flow Chart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

POWER SAVING MODES (Cont'd)

3.4.4 Halt Mode

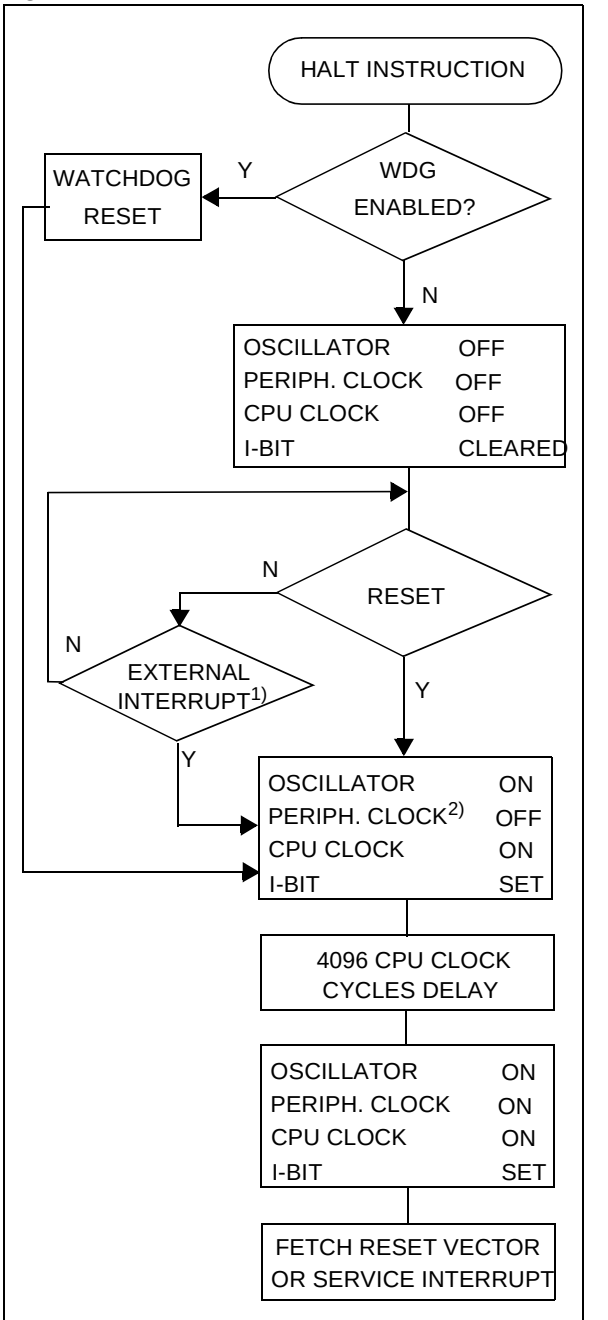
The Halt mode is the MCU lowest power consumption mode. The Halt mode is entered by executing the HALT instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals. The Halt mode cannot be used when the watchdog is enabled, if the HALT instruction is executed while the watchdog system is enabled, a watchdog reset is generated thus resetting the entire MCU.

When entering Halt mode, the I bit in the CC Register is cleared so as to enable External Interrupts. If an interrupt occurs, the CPU becomes active.

The MCU can exit the Halt mode upon reception of an interrupt or a reset. Refer to the Interrupt Mapping Table. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

Figure 17. HALT Flow Chart



1) or some specific interrupts
2) if reset PERIPH. CLOCK = ON ; if interrupt PERIPH. CLOCK = OFF

Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

3.5 MISCELLANEOUS REGISTER

The Miscellaneous register allows to select the SLOW operating mode, the polarity of external interrupt requests and to output the internal clock.

Register Address: 0020h — Read/Write

Reset Value: 0000 0000 (00h)

7							0
PEI3	PEI2	MCO	PEI1	PEI0	PSM1	PSM0	SMS

Bit 7:6 = **PEI[3:2]** *External Interrupt EI3 and EI2 Polarity Options*.

These bits are set and cleared by software. They determine which event on EI2 and EI3 causes the external interrupt according to Table 6.

Table 6. EI2 and EI3 External Interrupt Polarity Options

MODE	PEI3	PEI2
Falling edge and low level (Reset state)	0	0
Falling edge only	1	0
Rising edge only	0	1
Rising and falling edge	1	1

Note: Any modification of one of these two bits resets the interrupt request related to this interrupt vector.

Bit 5 = **MCO** *Main Clock Out*
This bit is set and cleared by software. When set, it enables the output of the Internal Clock on the PPF0 I/O port.
0 - PF0 is a general purpose I/O port.
1 - MCO alternate function (f_{CPU} is output on PF0 pin).

Bit 4:3 = **PEI[1:0]** *External Interrupt EI1 and EI0 Polarity Options*.
These bits are set and cleared by software. They determine which event on EI0 and EI1 causes the external interrupt according to Table 7.

Table 7. EI0 and EI1 External Interrupt Polarity Options

MODE	PEI1	PEI0
Falling edge and low level (Reset state)	0	0
Falling edge only	1	0
Rising edge only	0	1
Rising and falling edge	1	1

Note: Any modification of one of these two bits resets the interrupt request related to this interrupt vector.l

Bit 2:1 = **PSM[1:0]** *Prescaler for Slow Mode.*
These bits are set and cleared by software. They determine the CPU clock when the SMS bit is set according to the following table.

Table 8. f_{CPU} Value in Slow Mode

f_{CPU} Value	PSM1	PSM0
f_{OSC} / 4	0	0
f_{OSC} / 16	0	1
f_{OSC} / 8	1	0
f_{OSC} / 32	1	1

Bit 0 = **SMS** *Slow Mode Select*
This bit is set and cleared by software.
0: Normal Mode - $f_{CPU} = f_{OSC}/ 2$ (Reset state)
1: Slow Mode - the f_{CPU} value is determined by the PSM[1:0] bits.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

4.1.1 Introduction

The I/O ports offer different functional modes:

– transfer of data through digital inputs and outputs

and for specific pins:

– analog signal input (ADC)

– alternate signal input/output for the on-chip peripherals.

– external interrupt generation

An I/O port is composed of up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

4.1.2 Functional Description

Each port is associated to 2 main registers:

– Data Register (DR)

– Data Direction Register (DDR)

and some of them to an optional register:

– Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, for specific ports which do not provide this register refer to the I/O Port Implementation Section 4.1.3. The generic I/O block diagram is shown on Figure 19.

4.1.2.1 Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Notes:

1. All the inputs are triggered by a Schmitt trigger.

2. When switching from input mode to output mode, the DR register should be written first to output the correct value as soon as the port is configured as an output.

Interrupt function

When an I/O is configured in Input with Interrupt, an event on this I/O can generate an external Interrupt request to the CPU. The interrupt polarity is given independently according to the description mentioned in the Miscellaneous register or in the interrupt register (where available).

Each pin can independently generate an Interrupt request.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see Interrupts section). If several input pins are configured as inputs to the same interrupt vector, their signals are logically ANDed before entering the edge/level detection block. For this reason if one of the interrupt pins is tied low, it masks the other ones.

4.1.2.2 Output Mode

The pin is configured in output mode by setting the corresponding DDR register bit.

In this mode, writing “0” or “1” to the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Note: In this mode, the interrupt function is disabled.

4.1.2.3 Digital Alternate Function

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over standard I/O programming. When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin has to be configured in input mode. In this case, the pin’s state is also digitally readable by addressing the DR register.

Notes:
1. Input pull-up configuration can cause an unexpected value at the input of the alternate peripheral input.
2. When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

***Warning*:** The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

I/O PORTS (Cont'd)

4.1.2.4 Analog Alternate Function

When the pin is used as an ADC input the I/O must be configured as input, floating. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

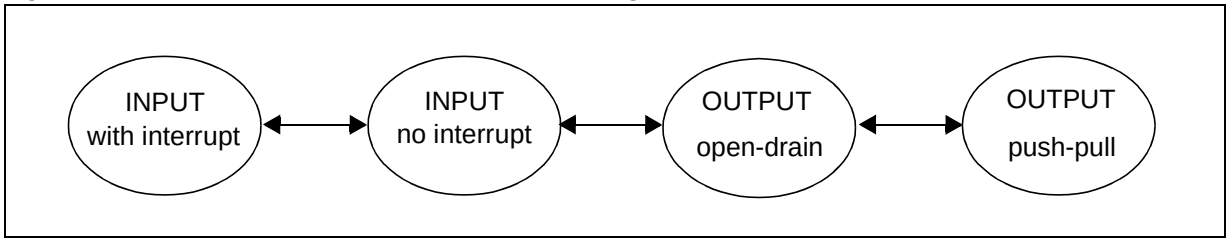
It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

Warning: The analog input voltage level must be within the limits stated in the Absolute Maximum Ratings.

4.1.3 I/O Port Implementation

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port such as ADC Input (see Figure 19) or true open drain. Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 18. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 18. Recommended I/O State Transition Diagram



I/O PORTS (Cont'd)

Figure 19. I/O Block Diagram

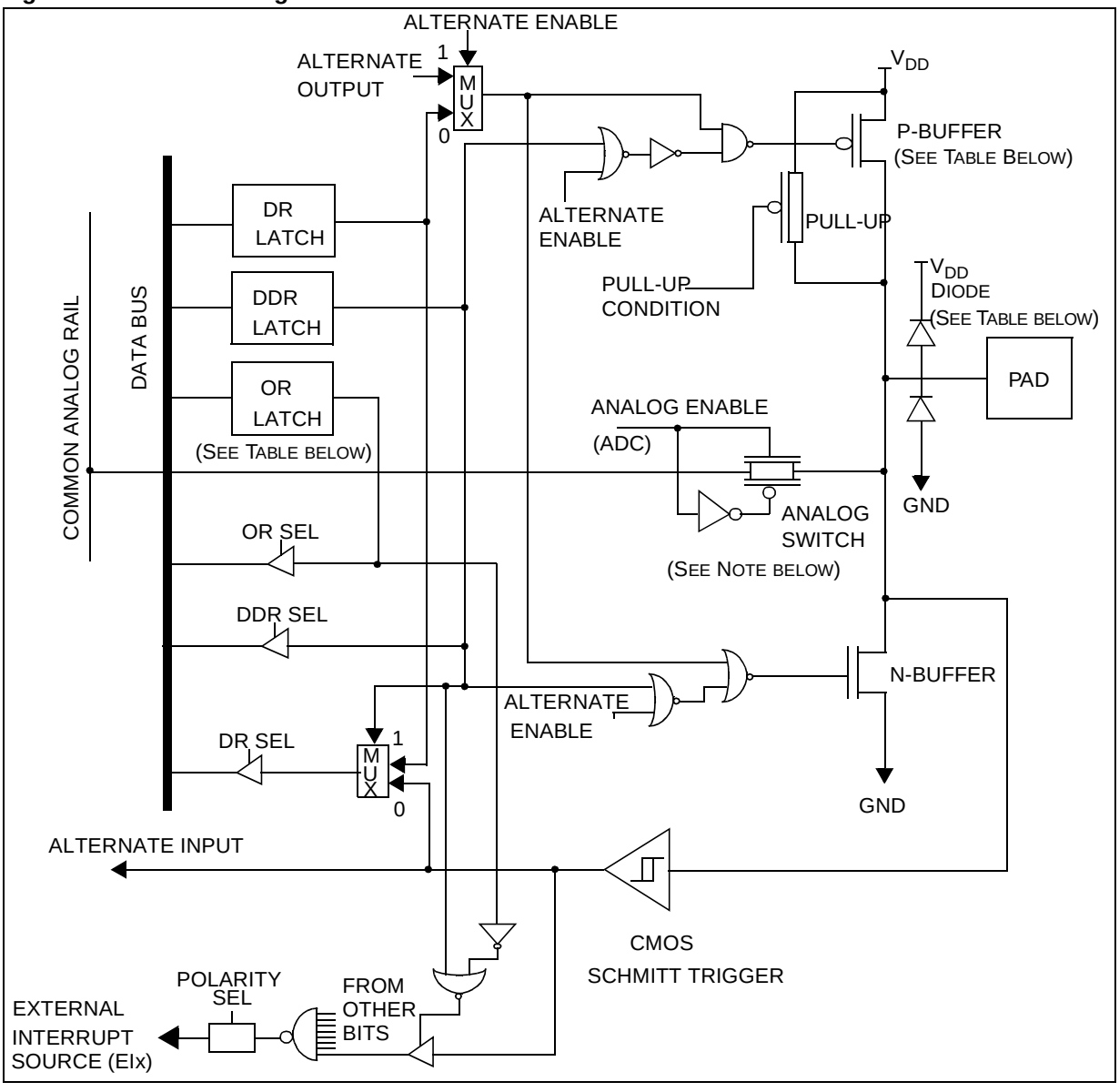


Table 9. Port Mode Configuration

Configuration Mode	Pull-up	P-buffer	V_{DD} Diode
Floating	0	0	1
Pull-up	1	0	1
Push-pull	0	1	1
True Open Drain	not present	not present	not present
Open Drain (logic level)	0	0	1

Legend:
0 - present, not activated
1 - present and activated

Notes:
– No OR Register on some ports (see register map).
– ADC Switch on ports with analog alternate functions.

I/O PORTS (Cont'd)

Table 10. Port Configuration

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)	
		OR = 0	OR = 1	OR = 0	OR =1
Port A	PA3	floating*	pull-up with interrupt	open-drain	push-pull
	PA4:PA7	floating*		true open drain, high sink capability	
Port B	PB0:PB4	floating*	pull-up with interrupt	open-drain	push-pull
Port C	PC0:PC7	floating*	pull-up	open-drain	push-pull
Port D	PD0:PD5	floating*	pull-up	open-drain	push-pull
Port E	PE0:PE1	floating*	pull-up	open-drain	push-pull
Port F	PF0:PF2	floating*	pull-up with interrupt	open-drain	push-pull
	PF4, PF6, PF7	floating*	pull-up	open-drain	push-pull

* Reset state (The bits corresponding to unavailable pins are forced to 1 by hardware, this affects the reset status value).

Warning: All bits of the DDR register which correspond to unconnected I/Os must be left at their reset value. They must not be modified by the user otherwise a spurious interrupt may be generated.

I/O PORTS (Cont'd)

4.1.4 Register Description

4.1.4.1 Data registers

Port A Data Register (PADR)

Port B Data Register (PBDR)

Port C Data Register (PCDR)

Port D Data Register (PDDR)

Port E Data Register (PEDR)

Port F Data Register (PFDR)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = D7-D0 *Data Register 8 bits.*

The DR register has a specific behaviour according to the selected input/output configuration. Writing the DR register is always taken in account even if the pin is configured as an input. Reading the DR register returns either the DR register latch content (pin configured as output) or the digital value applied to the I/O pin (pin configured as input).

4.1.4.2 Data direction registers

Port A Data Direction Register (PADDR)

Port B Data Direction Register (PBDDR)

Port C Data Direction Register (PCDDR)

Port D Data Direction Register (PDDDR)

Port E Data Direction Register (PEDDR)

Port F Data Direction Register (PFDDR)

Read/Write
Reset Value: 0000 0000 (00h) (input mode)

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = DD7-DD0 *Data Direction Register 8 bits.*

The DDR register gives the input/output direction configuration of the pins. Each bits is set and cleared by software.

0: Input mode
1: Output mode

4.1.4.3 Option registers

Port A Option Register (PAOR)

Port B Option Register (PBOR)

Port C Option Register (PBOR)

Port D Option Register (PBOR)

Port E Option Register (PBOR)

Port F Option Register (PFOR)

Read/Write
Reset Value: see Register Memory Map Table 3

7							0
O7	O6	O5	O4	O3	O2	O1	O0

Bit 7:0 = O7-O0 *Option Register 8 bits.*

The OR register allow to distinguish in input mode if the interrupt capability or the floating configuration is selected.

In output mode it select push-pull or open-drain capability.

Each bit is set and cleared by software.

Input mode:

0: floating input
1: input pull-up with interrupt

Output mode:

0: open-drain configuration
1: push-pull configuration

I/O PORTS (Cont'd)

Table 11. I/O Port Register Map

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0000h	**PADR**	D7	D6	D5	D4	D3	D2	D1	D0
0001h	**PADDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
0002h	**PAOR**	O7	O6	O5	O4	O3	O2	O1	O0
0004h	**PCDR**	D7	D6	D5	D4	D3	D2	D1	D0
0005h	**PCDDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
0006h	**PCOR**	O7	O6	O5	O4	O3	O2	O1	O0
0008h	**PBDR**	D7	D6	D5	D4	D3	D2	D1	D0
0009h	**PBDDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
000Ah	**PBOR**	O7	O6	O5	O4	O3	O2	O1	O0
000Ch	**PEDR**	D7	D6	D5	D4	D3	D2	D1	D0
000Dh	**PEDDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
000Eh	**PEOR**	O7	O6	O5	O4	O3	O2	O1	O0
0010h	**PDDR**	D7	D6	D5	D4	D3	D2	D1	D0
0011h	**PDDDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
0012h	**PDOR**	O7	O6	O5	O4	O3	O2	O1	O0
0014h	**PFDR**	D7	D6	D5	D4	D3	D2	D1	D0
0015h	**PFDDR**	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
0016h	**PFOR**	O7	O6	O5	O4	O3	O2	O1	O0

4.2 WATCHDOG TIMER (WDG)

4.2.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

4.2.2 Main Features

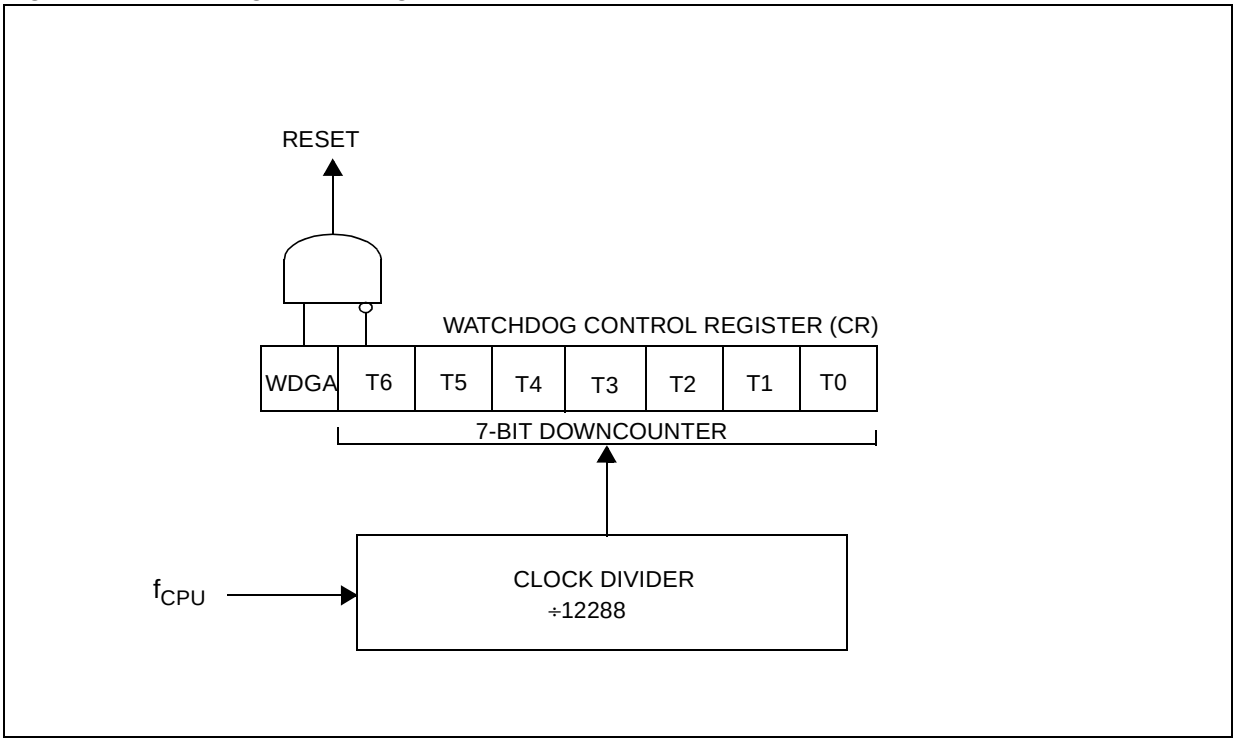
- Programmable timer (64 increments of 12288 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) after a HALT instruction or when the T6 bit reaches zero
- Hardware Watchdog selectable by option byte
- Watchdog Reset indicated by status flag (in versions with Safe Reset option only)

4.2.3 Functional Description

The counter value stored in the CR register (bits T[6:0]), is decremented every 12,288 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

Figure 20. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 12):

– The WDGA bit is set (watchdog enabled)

– The T6 bit is set to prevent generating an immediate reset

– The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 12.Watchdog Timing (f_{CPU} = 8 MHz)

	CR Register initial value	WDG timeout period (ms)
Max	FFh	98.304
Min	C0h	1.536

Notes: Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

4.2.4 Hardware Watchdog Option

If Hardware Watchdog Is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the device-specific Option Byte description.

4.2.5 Low Power Modes

Mode	Description
WAIT	No effect on Watchdog.
HALT	Immediate reset generation as soon as the HALT instruction is executed if the Watchdog is activated (WDGA bit is set).

4.2.6 Interrupts

None.

4.2.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = **WDGA** *Activation bit.*
This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.
0: Watchdog disabled
1: Watchdog enabled

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bit 6:0 = **T[6:0]** *7-bit timer (MSB to LSB).*
These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

STATUS REGISTER (SR)

Read/Write

Reset Value*: 0000 0000 (00h)

7							0
-	-	-	-	-	-	-	WDOGF

Bit 0 = **WDOGF** *Watchdog flag.*
This bit is set by a watchdog reset and cleared by software or a power on/off reset. This bit is useful for distinguishing power/on off or external reset and watchdog reset.
0: No Watchdog reset occurred
1: Watchdog reset occurred

* Only by software and power on/off reset

Note: This register is not used in versions without LVD Reset.

Table 13. WDG Register Map

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
2A	**WDGCR**	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1
2B	**WDGSR**	-	-	-	-	-	-	-	WDOGF
	Reset Value	0	0	0	0	0	0	0	0

4.3 16-BIT TIMER

4.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

4.3.2 Main Features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 21.

***Note:** Some external pins are not available on all devices. Refer to the device pin out description.

When reading an input signal which is not available on an external pin, the value will always be '1'.

4.3.3 Functional Description

4.3.3.1 Counter

The principal block of the Programmable Timer is a 16-bit free running increasing counter and its associated 16-bit registers:

Counter Registers

- Counter High Register (CHR) is the most significant byte (MSB).
- Counter Low Register (CLR) is the least significant byte (LSB).

Alternate Counter Registers

- Alternate Counter High Register (ACHR) is the most significant byte (MSB).
- Alternate Counter Low Register (ACLR) is the least significant byte (LSB).

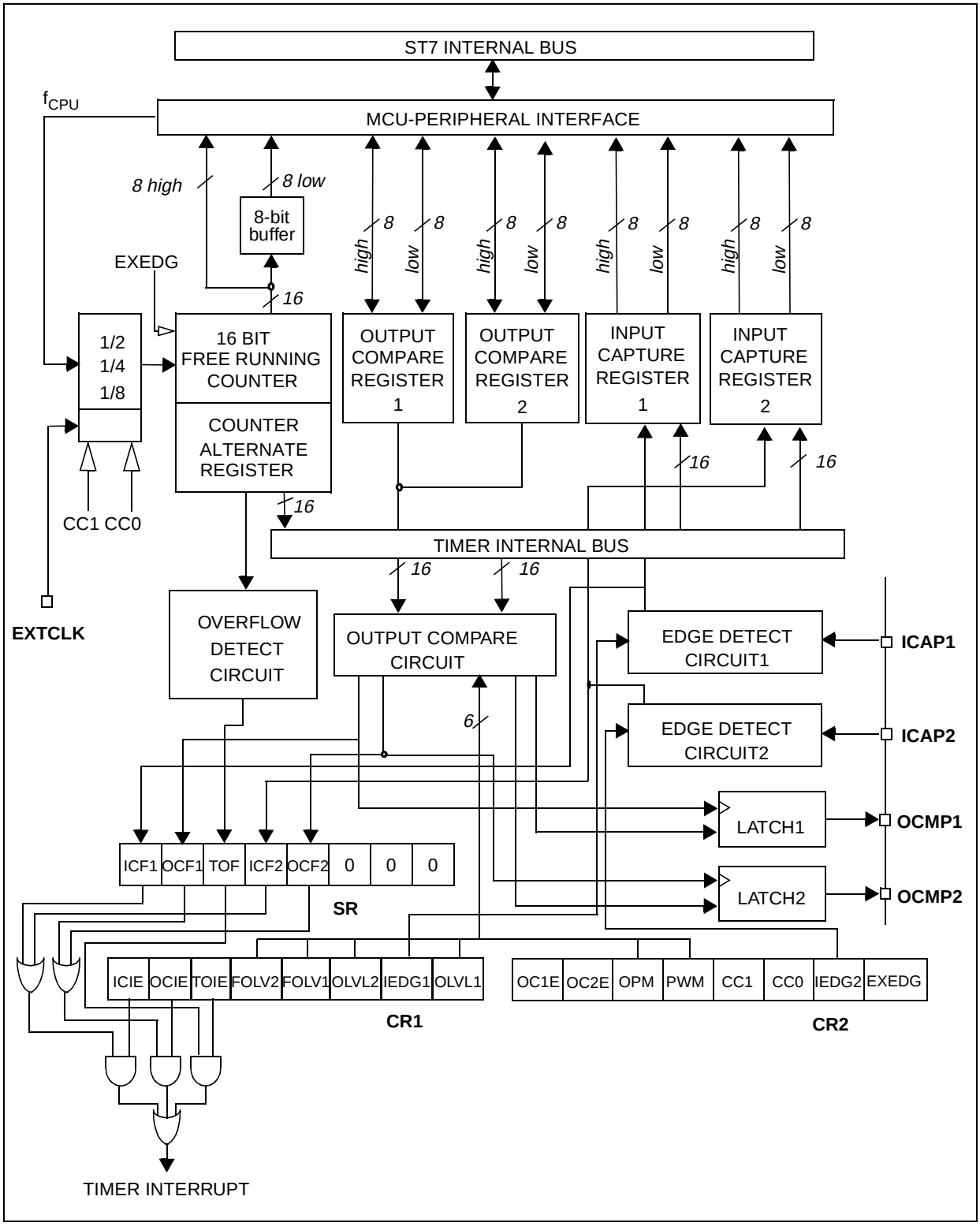
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (overflow flag), (see note at the end of paragraph titled 16-bit read sequence).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 14. The value in the counter register repeats every 131.072, 262.144 or 524.288 internal processor-clock cycles depending on the CC1 and CC0 bits.

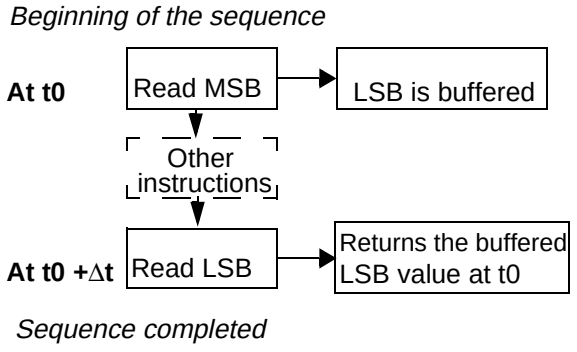
16-BIT TIMER (Cont'd)

Figure 21. Timer Block Diagram



16-BIT TIMER (Cont'd)

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the MSB first, then the LSB value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, one pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h then:

– The TOF bit of the SR register is set.

– A timer interrupt is generated if:

  – TOIE bit of the CR1 register is set and

  – I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Notes: The TOF bit is not cleared by accesses to ACLR register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

4.3.3.2 External Clock

The external clock (where available) is selected if CC0=1 and CC1=1 in CR2 register.

The status of the EXEDG bit determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronised with the falling edge of the internal CPU clock.

At least four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

16-BIT TIMER (Cont'd)

Figure 22. Counter Timing Diagram, internal clock divided by 2

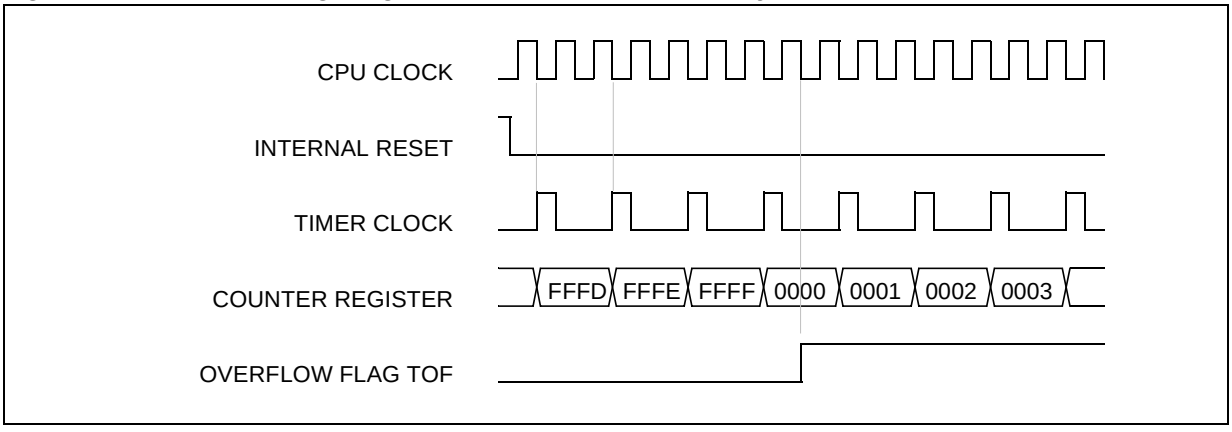


Figure 23. Counter Timing Diagram, internal clock divided by 4

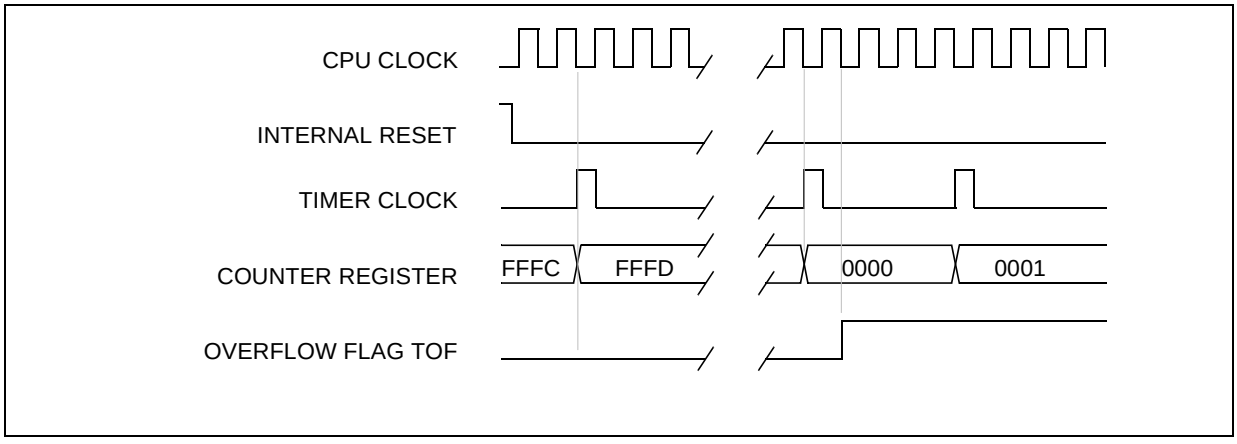
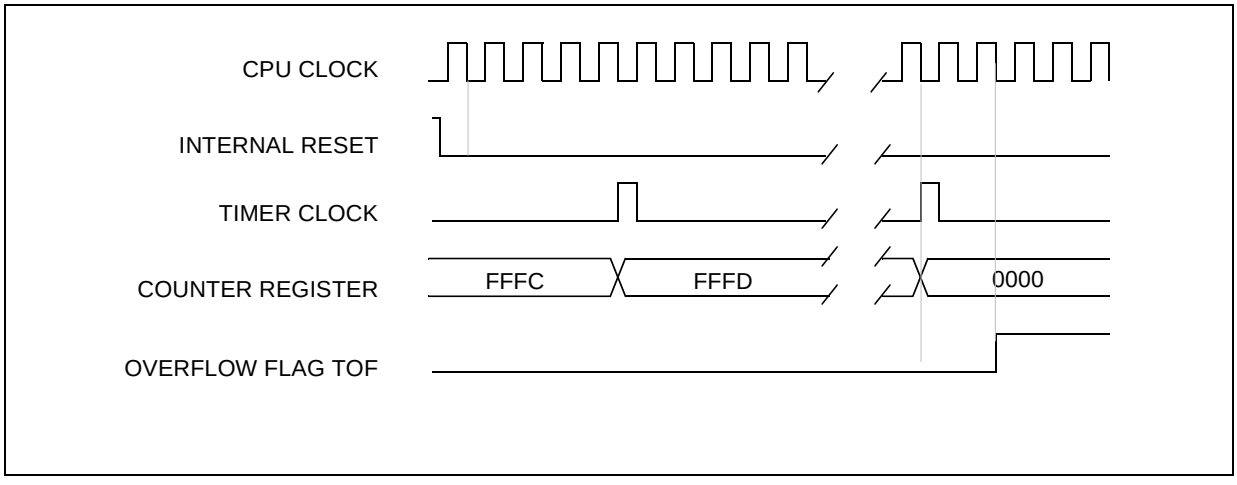


Figure 24. Counter Timing Diagram, internal clock divided by 8



16-BIT TIMER (Cont'd)

4.3.3.3 Input Capture

In this section, the index, *i*, may be 1 or 2.

The two input capture 16-bit registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition detected by the ICAP*i* pin (see figure 5).

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

IC*i* register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of the Control Register (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/(CC1.CC0)).

Procedure:

To use the input capture function select the following in the CR2 register:

– Select the timer clock (CC1-CC0) (see Table 14).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input).

And select the following in the CR1 register:

– Set the ICIE bit to generate an interrupt after an input capture coming from both the ICAP1 pin or the ICAP2 pin

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input).

When an input capture occurs:

– ICF*i* bit is set.

– The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see Figure 26).

– A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

Notes:

1. After reading the IC*i*HR register, transfer of input capture data is inhibited until the IC*i*LR register is also read.
2. The IC*i*R register always contains the free running counter value which corresponds to the most recent input capture.
3. The 2 input capture functions can be used together even if the timer also uses the output compare mode.
4. In One pulse Mode and PWM mode only the input capture 2 can be used.
5. The alternate inputs (ICAP1 & ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture process.
6. Moreover if one of the ICAP*i* pin is configured as an input and the second one as an output, an interrupt can be generated if the user toggle the output pin and if the ICIE bit is set.
7. The TOF bit can be used with interrupt in order to measure event that go beyond the timer range (FFFFh).

16-BIT TIMER (Cont'd)

Figure 25. Input Capture Block Diagram

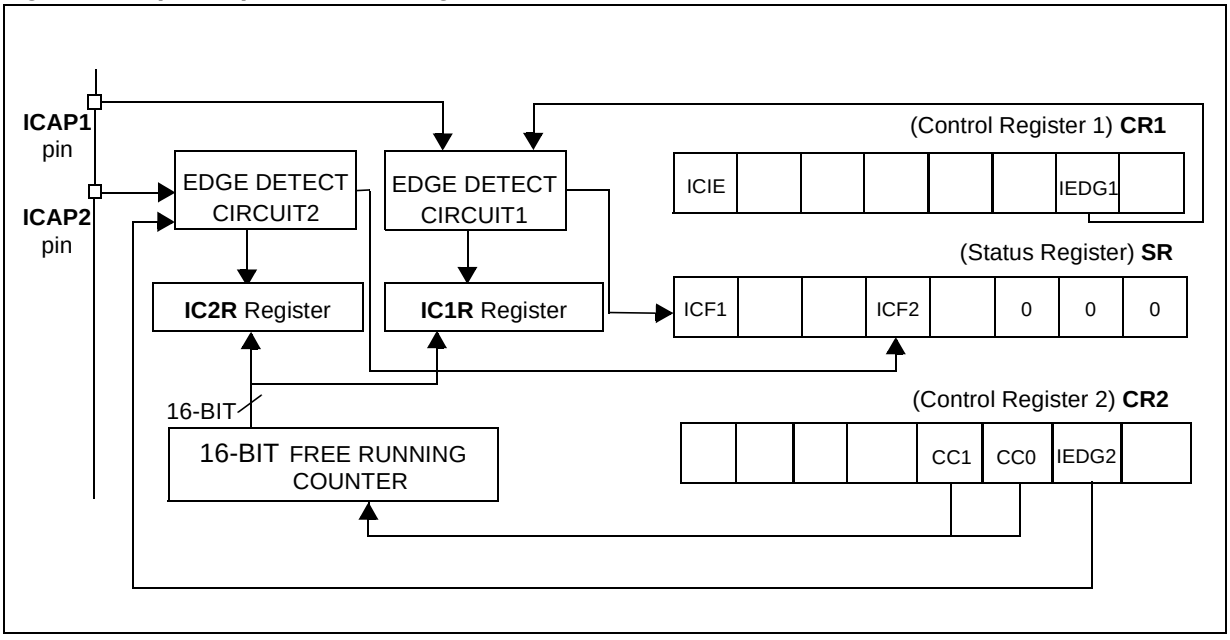
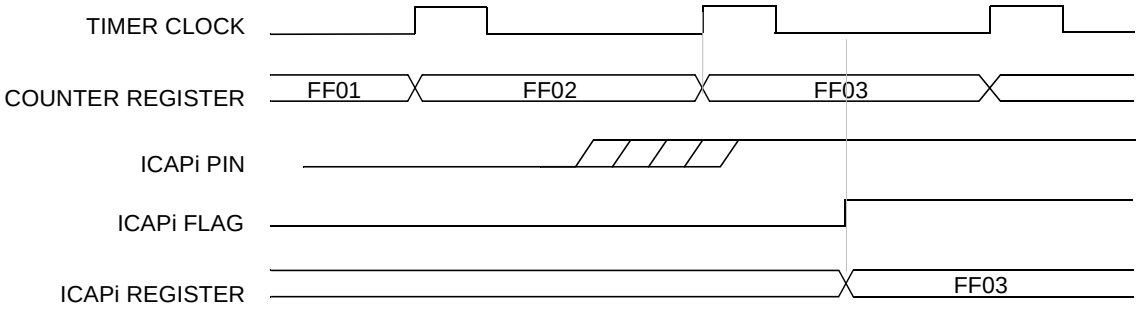


Figure 26. Input Capture Timing Diagram



Note: Active edge is rising edge.

16-BIT TIMER (Cont'd)

4.3.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2.

This function can be used to control an output waveform or indicating when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

– Assigns pins with a programmable value if the OCIE bit is set

– Sets a flag in the status register

– Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the free running counter each timer clock cycle.

	MS Byte	LS Byte
OC*i*R	OC*i*HR	OC*i*LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: ($f_{CPU/(CC1.CC0)}$).

Procedure:

To use the output compare function, select the following in the CR2 register:

– Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* function.

– Select the timer clock (CC1-CC0) (see Table 14).

And select the following in the CR1 register:

– Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.

– Set the OCIE bit to generate an interrupt if it is needed.

When a match is found:

– OCF*i* bit is set.

– The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset and stays low until valid compares change it to a high level).

– A timer interrupt is generated if the OCIE bit is set in the CR2 register and the I bit is cleared in the CC register (CC).

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

Δt = Desired output compare period (in seconds)

f_{CPU} = Internal clock frequency

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC1-CC0 bits, see Table 14)

Clearing the output compare interrupt request is done by:

1. Reading the SR register while the OCF*i* bit is set.
2. An access (read or write) to the OC*i*LR register.

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

– Write to the OC*i*HR register (further compares are inhibited).

– Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).

– Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

Notes:

1. After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.
3. When the clock is divided by 2, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value (see Figure 28, on page 40). This behaviour is the same in OPM or PWM mode.
   When the clock is divided by 4, 8 or in external clock mode, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value plus 1 (see Figure 29, on page 40).
4. The output compare functions can be used both for generating external events on the OCMP*i* pins even if the input capture mode is also used.
5. The value in the 16-bit OC*i*R register and the OLV*i* bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

16-BIT TIMER (Cont'd)

Figure 27. Output Compare Block Diagram

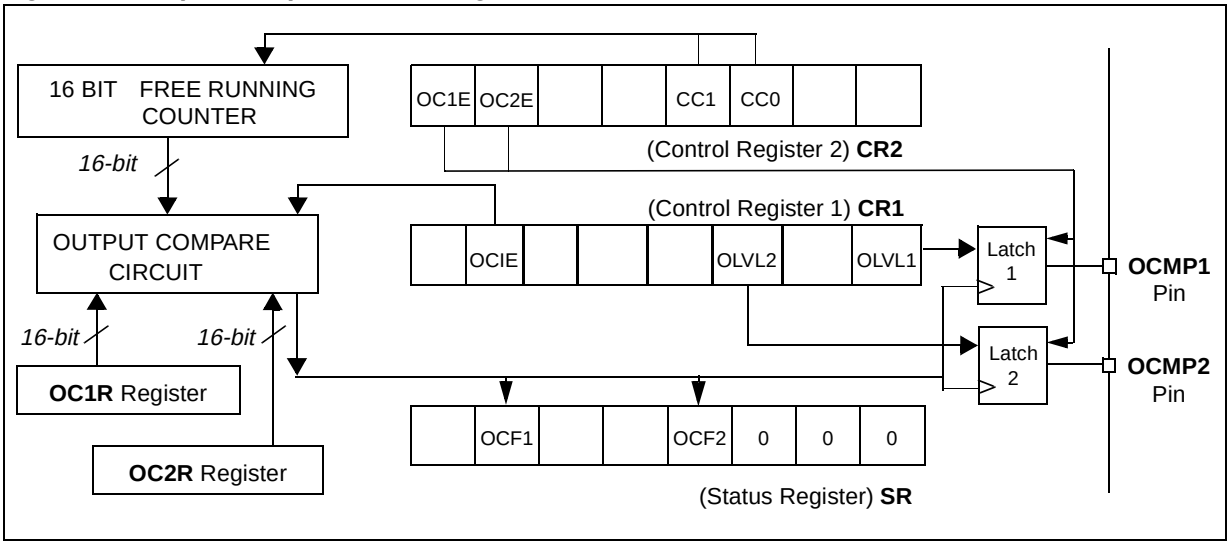


Figure 28. Output Compare Timing Diagram, Internal Clock Divided by 2

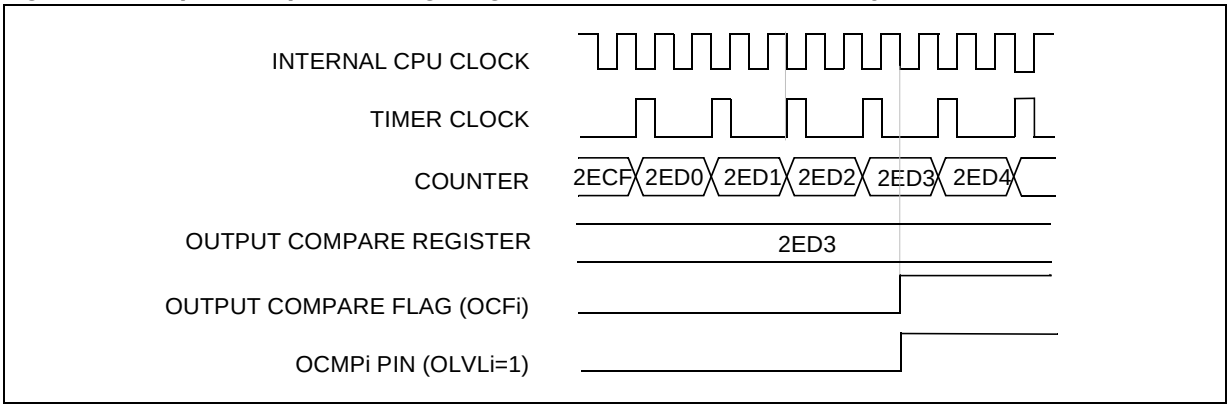
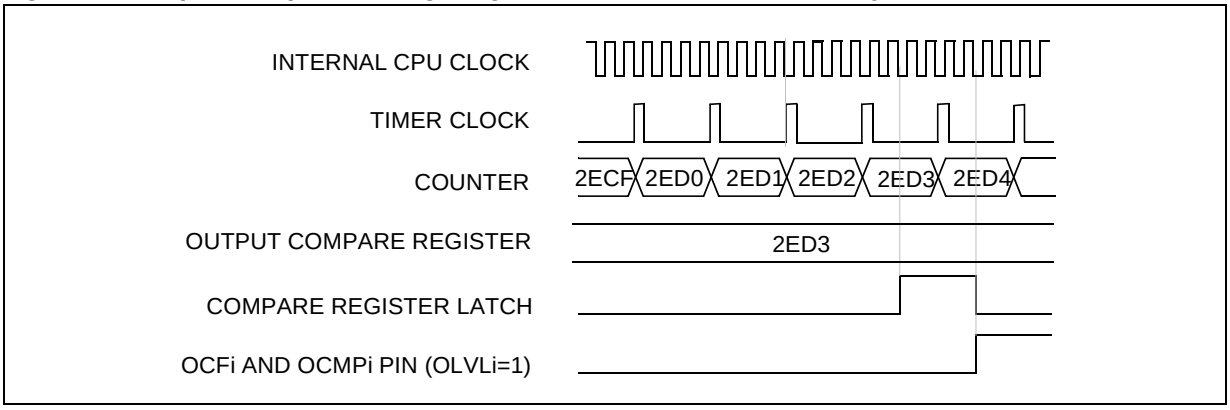


Figure 29. Output Compare Timing Diagram, Internal Clock Divided by 4



16-BIT TIMER (Cont'd)

4.3.3.5 Forced Compare

In this section *i* may represent 1 or 2.

The following bits of the CR1 register are used:

			FOLV2	FOLV1	OLVL2		OLVL1

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit=1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

4.3.3.6 One Pulse Mode

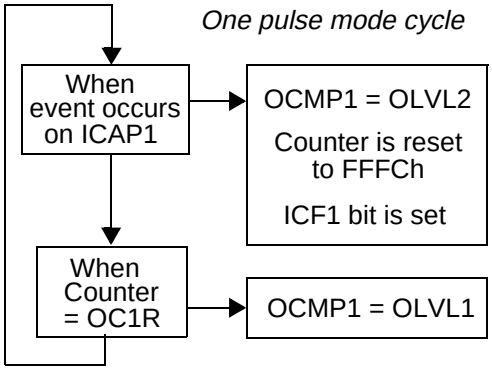
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use one pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in Section 4.3.3.7).
2. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
   - Set the OPM bit.
   - Select the timer clock CC1-CC0 (see Table 14).



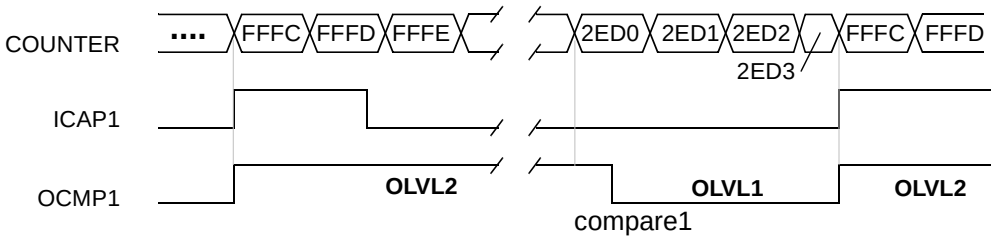
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 30).

Notes:

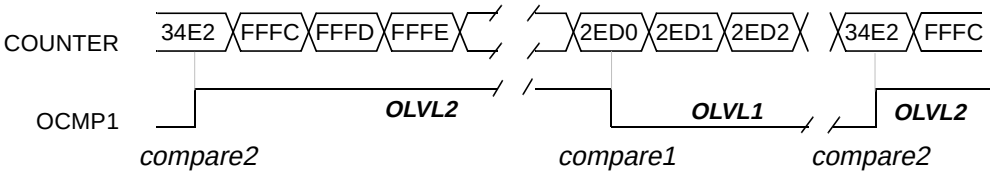
1. The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.
2. The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.
3. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.
4. If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.
5. The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.
6. When the one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 30. One Pulse Mode Timing Example



Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1

Figure 31. Pulse Width Modulation Mode Timing Example



Note: OC1R=2ED0h, OC2R=34E2, OLVL1=0, OLVL2= 1

16-BIT TIMER (Cont'd)

4.3.3.7 Pulse Width Modulation Mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

The pulse width modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so these functionality can not be used when the PWM mode is activated.

Procedure

To use pulse width modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal.
2. Load the OC1R register with the value corresponding to the length of the pulse if (OLVL1=0 and OLVL2=1).
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC1-CC0) (see Table 14).

If OLVL1=1 and OLVL2=0 the length of the positive pulse is the difference between the OC2R and OC1R registers.

If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\text{OC}i\text{R Value} = \frac{t * f_{CPU}}{PRESC} - 5$$

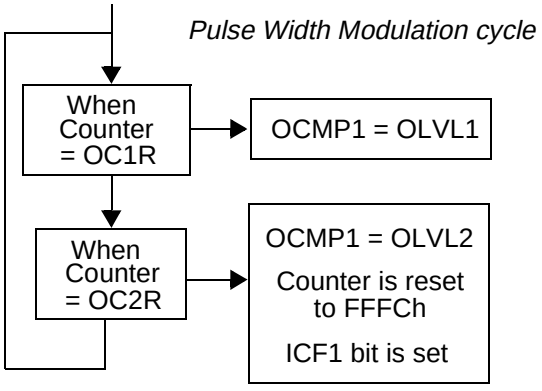
Where:

t = Desired output compare period (in seconds)

f_{CPU} = Internal clock frequency

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC1-CC0 bits, see Table 14)

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 31).



Notes:

1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
   Therefore the Input Capture 1 function is inhibited but the Input Capture 2 is available.
2. The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.
3. The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.
4. In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected to the timer. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each period and ICF1 can also generates interrupt if ICIE is set.
5. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

16-BIT TIMER (Cont'd)

4.3.4 Low Power Modes

Mode	Description
WAIT	No effect on 16-bit Timer. Timer interrupts cause the device to exit from WAIT mode.
HALT	16-bit Timer registers are frozen. In HALT mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the MCU is woken up by an interrupt with "exit from HALT mode" capability or from the counter reset value when the MCU is woken up by a RESET. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the MCU is woken up by an interrupt with "exit from HALT mode" capability, the ICF*i* bit is set, and the counter value present when exiting from HALT mode is captured into the IC*i*R register.

4.3.5 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Input Capture 1 event/Counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2		Yes	No
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE	Yes	No
Output Compare 2 event (not available in PWM mode)	OCF2		Yes	No
Timer Overflow event	TOF	TOIE	Yes	No

Note: The 16-bit Timer interrupt events are connected to the same interrupt vector (see Interrupts chapter). These events generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

16-BIT TIMER (Cont'd)

4.3.6 Register Description

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = **OCIE** *Output Compare Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = **TOIE** *Timer Overflow Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = **FOLV2** *Forced Output Compare 2.*
This bit is set and cleared by software.
0: No effect on the OCMP2 pin.
1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.

Bit 3 = **FOLV1** *Forced Output Compare 1.*
This bit is set and cleared by software.
0: No effect on the OCMP1 pin.
1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.

Bit 2 = **OLVL2** *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse Mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

16-BIT TIMER (Cont'd)

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the Output Compare 1 function of the timer remains active.
0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP1 pin alternate function enabled.

Bit 6 = **OC2E** *Output Compare 2 Enable.*
This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the Output Compare 2 function of the timer remains active.
0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP2 pin alternate function enabled.

Bit 5 = **OPM** *One Pulse Mode.*
0: One Pulse Mode is not active.
1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation.*
0: PWM mode is not active.
1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

Bit 3, 2 = **CC1-CC0** *Clock Control.*

The value of the timer clock depends on these bits:

Table 14. Clock Control Bits

Timer Clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2	0	1
f_{CPU} / 8	1	0
External Clock (where available)	1	1

Bit 1 = **IEDG2** *Input Edge 2.*
This bit determines which type of level transition on the ICAP2 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin EXTCLK will trigger the free running counter.
0: A falling edge triggers the free running counter.
1: A rising edge triggers the free running counter.

16-BIT TIMER (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0

Bit 7 = **ICF1** *Input Capture Flag 1.*
0: No input capture (reset value).
1: An input capture has occurred or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow.*
0: No timer overflow (reset value).
1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*
0: No input capture (reset value).
1: An input capture has occurred.To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2-0 = Reserved, forced by hardware to 0.

INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the SR register clears the TOF bit.

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to SR register does not clear the TOF bit in SR register.

7							0
MSB							LSB

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

7							0
MSB							LSB

INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).

7							0
MSB							LSB

16-BIT TIMER (Cont'd)

Table 15. 16-Bit Timer Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
TimerA: 32	**CR1**	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
TimerB: 42	Reset Value	0	0	0	0	0	0	0	0
TimerA: 31	**CR2**	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
TimerB: 41	Reset Value	0	0	0	0	0	0	0	0
TimerA: 33	**SR**	ICF1	OCF1	TOF	ICF2	OCF2	-	-	-
TimerB: 43	Reset Value	0	0	0	0	0	0	0	0
TimerA: 34	**IC1HR**	MSB	-	-	-	-	-	-	LSB
TimerB: 44	Reset Value	-							-
TimerA: 35	**IC1LR**	MSB	-	-	-	-	-	-	LSB
TimerB: 45	Reset Value	-							-
TimerA: 36	**OC1HR**	MSB	-	-	-	-	-	-	LSB
TimerB: 46	Reset Value	1	0	0	0	0	0	0	0
TimerA: 37	**OC1LR**	MSB	-	-	-	-	-	-	LSB
TimerB: 47	Reset Value	0	0	0	0	0	0	0	0
TimerA: 3E	**OC2HR**	MSB	-	-	-	-	-	-	LSB
TimerB: 4E	Reset Value	1	0	0	0	0	0	0	0
TimerA: 3F	**OC2LR**	MSB	-	-	-	-	-	-	LSB
TimerB: 4F	Reset Value	0	0	0	0	0	0	0	0
TimerA: 38	**CHR**	MSB							LSB
TimerB: 48	Reset Value	1	1	1	1	1	1	1	1
TimerA: 39	**CLR**	MSB							LSB
TimerB: 49	Reset Value	1	1	1	1	1	1	0	0
TimerA: 3A	**ACHR**	MSB							LSB
TimerB: 4A	Reset Value	1	1	1	1	1	1	1	1
TimerA: 3B	**ACLR**	MSB							LSB
TimerB: 4B	Reset Value	1	1	1	1	1	1	0	0
TimerA: 3C	**IC2HR**	MSB	-	-	-	-	-	-	LSB
TimerB: 4C	Reset Value	-							-
TimerA: 3D	**IC2LR**	MSB	-	-	-	-	-	-	LSB
TimerB: 4D	Reset Value	-							-

4.4 SERIAL COMMUNICATIONS INTERFACE (SCI)

4.4.1 Introduction

The Serial Communications Interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

4.4.2 Main Features

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Dual baud rate generator systems
- Independently programmable transmit and receive baud rates up to 250K baud.
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- Two receiver wake-up modes:
  - Address bit (MSB)
  - Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- Three error detection flags:
  - Overrun error
  - Noise error
  - Frame error
- Five interrupt sources with flags:
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error detected

4.4.3 General Description

The interface is externally connected to another device by two pins (see Figure 33):

– TDO: Transmit Data Output. When the transmitter is disabled, the output pin returns to its I/O port configuration. When the transmitter is enabled and nothing is to be transmitted, the TDO pin is at high level.

– RDI: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

Through this pins, serial data is transmitted and received as frames comprising:

– An Idle Line prior to transmission or reception

– A start bit

– A data word (8 or 9 bits) least significant bit first

– A Stop bit indicating that the frame is complete.

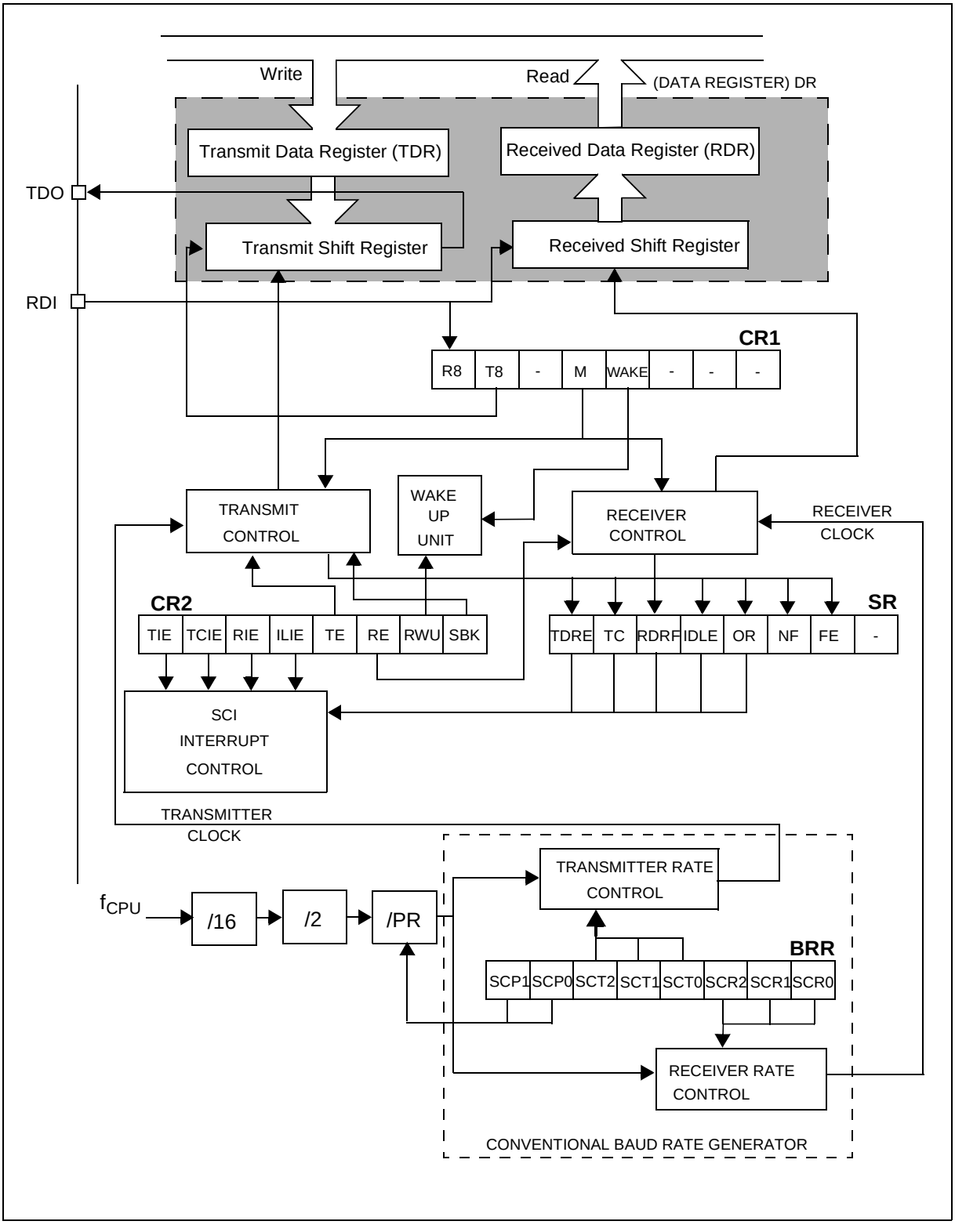
This interface uses two types of baud rate generator:

– A conventional type for commonly-used baud rates,

– An extended type with a prescaler offering a very wide range of baud rates even with non-standard oscillator frequencies.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 32. SCI Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 32. It contains 6 dedicated registers:

– Two control registers (CR1 & CR2)

– A status register (SR)

– A baud rate register (BRR)

– An extended prescaler receiver register (ERPR)

– An extended prescaler transmitter register (ETPR)

Refer to the register descriptions in Section 4.4.7for the definitions of each bit.

4.4.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the CR1 register (see Figure 32).

The TDO pin is in low state during the start bit.

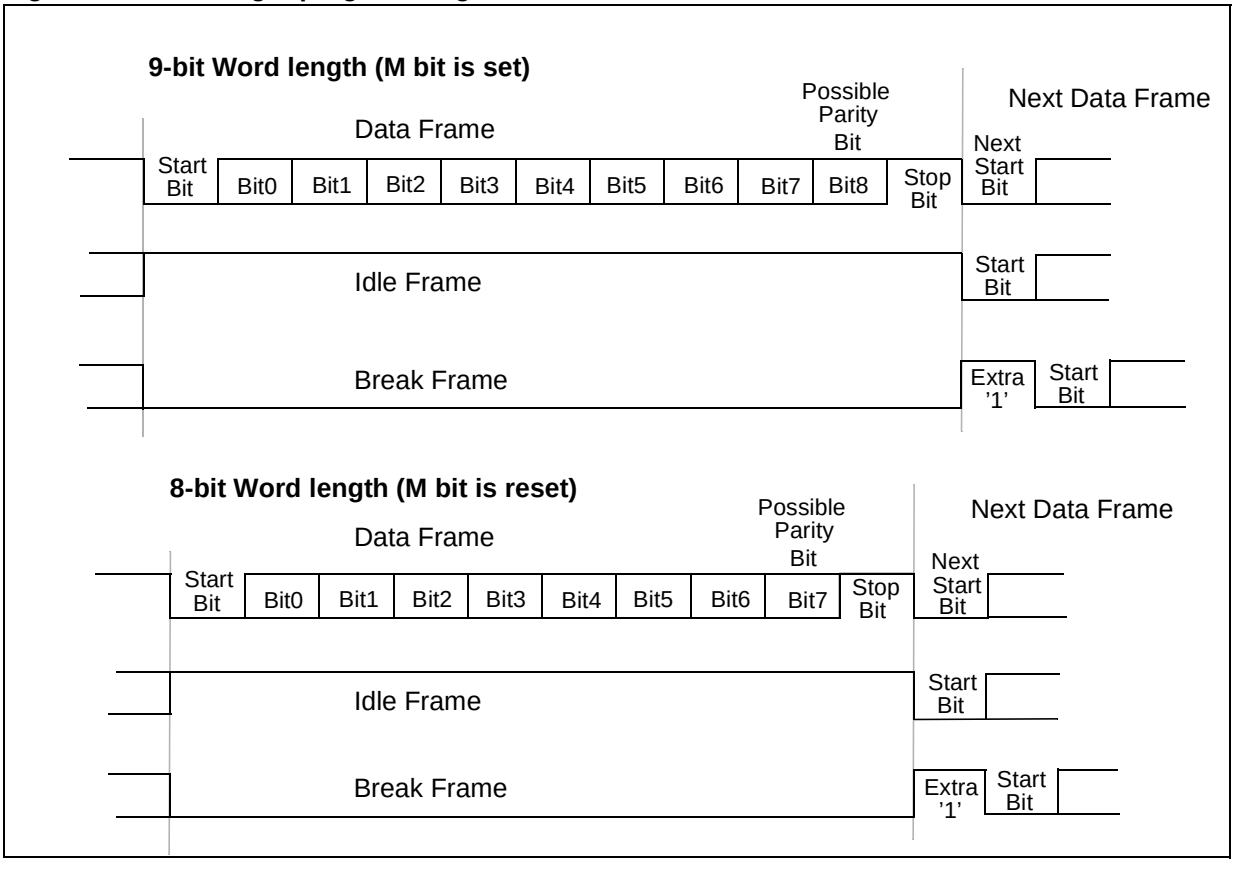
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of "1"s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving "0"s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra "1" bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

Figure 33. Word length programming



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.4.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the CR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the DR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 32).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR and the ETPR registers.

– Set the TE bit to assign the TDO pin to the alternate function and to send a idle frame as first transmission.

– Access the SR register and write the data to send in the DR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:
1. An access to the SR register
2. A write to the DR register

The TDRE bit is set by hardware and it indicates:

– The TDR register is empty.

– The data transfer is beginning.

– The next data can be written in the DR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I bit is cleared in the CCR register.

When a transmission is taking place, a write instruction to the DR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the DR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a frame transmission is complete (after the stop bit or after the break frame) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CCR register.

Clearing the TC bit is performed by the following software sequence:
1. An access to the SR register
2. A write to the DR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break Characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see Figure 33).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this bit by software the SCI insert a logic 1 bit at the end of the last break frame to guarantee the recognition of the start bit of the next frame.

Idle Characters

Setting the TE bit drives the SCI to send an idle frame before the first data frame.

Clearing and then setting the TE bit during a transmission sends an idle frame after the current word.

Note: Resetting and setting the TE bit causes the data in the TDR register to be lost. Therefore the best time to toggle the TE bit is when the TDRE bit is set i.e. before writing the next byte in the DR.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.4.3 Receiver

The SCI can receive data words of either 8 or 9 bits. When the M bit is set, word length is 9 bits and the MSB is stored in the R8 bit in the CR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, DR register consists in a buffer (RDR) between the internal bus and the received shift register (see Figure 32).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR and the ERPR registers.

– Set the RE bit, this enables the receiver which begins searching for a start bit.

When a character is received:

– The RDRF bit is set. It indicates that the content of the shift register is transferred to the RDR.

– An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

– The error flags can be set if a frame error, noise or an overrun error has been detected during reception.

Clearing the RDRF bit is performed by the following software sequence done by:

1. An access to the SR register

2. A read to the DR register.

The RDRF bit must be cleared before the end of the reception of the next character to avoid an overrun error.

Break Character

When a break character is received, the SPI handles it as a framing error.

Idle Character

When a idle frame is detected, there is the same procedure as a data received character plus an interrupt if the ILIE bit is set and the I bit is cleared in the CCR register.

Overrun Error

An overrun error occurs when a character is received when RDRF has not been reset. Data can not be transferred from the shift register to the TDR register as long as the RDRF bit is not cleared.

When a overrun error occurs:

– The OR bit is set.

– The RDR content will not be lost.

– The shift register will be overwritten.

– An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

The OR bit is reset by an access to the SR register followed by a DR register read operation.

Noise Error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

When noise is detected in a frame:

– The NF is set at the rising edge of the RDRF bit.

– Data is transferred from the Shift register to the DR register.

– No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The NF bit is reset by a SR register read operation followed by a DR register read operation.

Framing Error

A framing error is detected when:

– The stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.

– A break is received.

When the framing error is detected:

– the FE bit is set by hardware

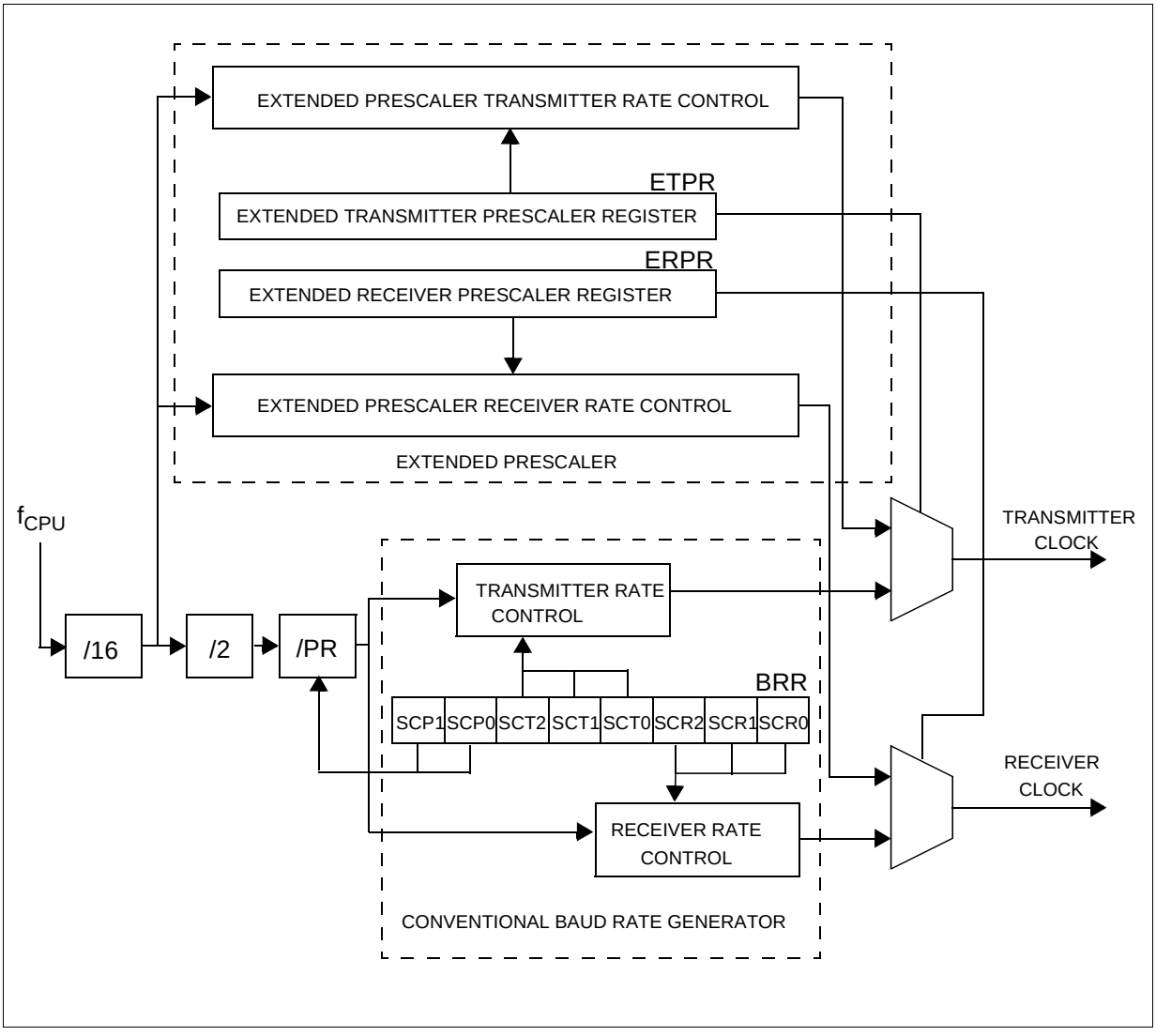
– Data is transferred from the Shift register to the DR register.

– No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The FE bit is reset by a SR register read operation followed by a DR register read operation.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 34. SCI Baud Rate and Extended Prescaler Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.4.4 Conventional Baud Rate Generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(32*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(32*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP0 & SCP1 bits)

TR = 1, 2, 4, 8, 16, 32, 64,128

(see SCT0, SCT1 & SCT2 bits)

RR = 1, 2, 4, 8, 16, 32, 64,128

(see SCR0,SCR1 & SCR2 bits)

All this bits are in the BRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR=13 and TR=RR=1, the transmit and receive baud rates are 19200 baud.

Note: the baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

4.4.4.5 Extended Baud Rate Generation

The extended prescaler option gives a very fine tuning on the baud rate, using a 255 value prescaler, whereas the conventional Baud Rate Generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the Figure 34.

The output clock rate sent to the transmitter or to the receiver will be the output from the 16 divider divided by a factor ranging from 1 to 255 set in the ERPR or the ETPR register.

Note: the extended prescaler is activated by setting the ETPR or ERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR} \qquad Rx = \frac{f_{CPU}}{16*ERPR}$$

with:

ETPR = 1,..,255 (see ETPR register)

ERPR = 1,.. 255 (see ERPR register)

4.4.4.6 Receiver Muting and Wake-up Feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non addressed receivers.

The non addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

All the reception status bits can not be set.

All the receive interrupt are inhibited.

A muted receiver may be awakened by one of the following two ways:

– by Idle Line detection if the WAKE bit is reset,

– by Address Mark detection if the WAKE bit is set.

Receiver wakes-up by Idle Line detection when the Receive line has recognised an Idle Frame. Then the RWU bit is reset by hardware but the IDLE bit is not set.

Receiver wakes-up by Address Mark detection when it received a "1" as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.5 Low Power Modes

Mode	Description
WAIT	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
HALT	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

4.4.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrrun Error Detected	OR		Yes	No
Idle Line Detected	IDLE	ILIE	Yes	No

The SCI interrupt events are connected to the same interrupt vector (see Interrupts chapter).

These events generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

4.4.7 Register Description

STATUS REGISTER (SR)

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	-

Bit 7 = **TDRE** *Transmit data register empty.*
This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE =1 in the CR2 register. It is cleared by a software sequence (an access to the SR register followed by a write to the DR register).
0: Data is not transferred to the shift register
1: Data is transferred to the shift register

Note: data will not be transferred to the shift register as long as the TDRE bit is not reset.

Bit 6 = **TC** *Transmission complete.*
This bit is set by hardware when transmission of a frame containing Data, a Preamble or a Break is complete. An interrupt is generated if TCIE=1 in the CR2 register. It is cleared by a software sequence (an access to the SR register followed by a write to the DR register).
0: Transmission is not complete
1: Transmission is complete

Bit 5 = **RDRF** *Received data ready flag.*
This bit is set by hardware when the content of the RDR register has been transferred into the DR register. An interrupt is generated if RIE=1 in the CR2 register. It is cleared by hardware when RE=0 or by a software sequence (an access to the SR register followed by a read to the DR register).
0: Data is not received
1: Received data is ready to be read

Bit 4 = **IDLE** *Idle line detect.*
This bit is set by hardware when a Idle Line is detected. An interrupt is generated if the ILIE=1 in the CR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Idle Line is detected
1: Idle Line is detected

Note: The IDLE bit will not be set again until the RDRF bit has been set itself (i.e. a new idle line occurs). This bit is not set by an idle line when the receiver wakes up from wake-up mode.

Bit 3 = **OR** *Overrun error.*
This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF=1. An interrupt is generated if RIE=1 in the CR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Overrun error
1: Overrun error is detected

Note: When this bit is set RDR register content will not be lost but the shift register will be overwritten.

Bit 2 = **NF** *Noise flag.*
This bit is set by hardware when noise is detected on a received frame. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No noise is detected
1: Noise is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt.

Bit 1 = **FE** *Framing error.*
This bit is set by hardware when a de-synchronization, excessive noise or a break character is detected. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Framing error is detected
1: Framing error or break character is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt. If the word currently being transferred causes both frame error and overrun error, it will be transferred and only the OR bit will be set.

Bit 0 = Unused.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: Undefined

7							0
R8	T8	-	M	WAKE	-	-	-

Bit 7 = **R8** *Receive data bit 8.*
This bit is used to store the 9th bit of the received word when M=1.

Bit 6 = **T8** *Transmit data bit 8.*
This bit is used to store the 9th bit of the transmitted word when M=1.

Bit 4 = **M** *Word length.*
This bit determines the word length. It is set or cleared by software.
0: 1 Start bit, 8 Data bits, 1 Stop bit
1: 1 Start bit, 9 Data bits, 1 Stop bit

Bit 3 = **WAKE** *Wake-Up method.*
This bit determines the SCI Wake-Up method, it is set or cleared by software.
0: Idle Line
1: Address Mark

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

Bit 7 = **TIE** *Transmitter interrupt enable*.
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever TDRE=1 in the SR register.

Bit 6 = **TCIE** *Transmission complete interrupt enable*

This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever TC=1 in the SR register

Bit 5 = **RIE** *Receiver interrupt enable*.
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever OR=1 or RDRF=1 in the SR register

Bit 4 = **ILIE** *Idle line interrupt enable.*
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever IDLE=1 in the SR register.

Bit 3 = **TE** *Transmitter enable.*
This bit enables the transmitter and assigns the TDO pin to the alternate function. It is set and cleared by software.
0: Transmitter is disabled, the TDO pin is back to the I/O port configuration.
1: Transmitter is enabled

Note: during transmission, a "0" pulse on the TE bit ("0" followed by "1") sends a preamble after the current word.

Bit 2 = **RE** *Receiver enable.*
This bit enables the receiver. It is set and cleared by software.
0: Receiver is disabled, it resets the RDRF, IDLE, OR, NF and FE bits of the SR register.
1: Receiver is enabled and begins searching for a start bit.

Bit 1 = **RWU** *Receiver wake-up.*
This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized.
0: Receiver in active mode
1: Receiver in mute mode

Bit 0 = **SBK** *Send break.*
This bit set is used to send break characters. It is set and cleared by software.
0: No break character is transmitted
1: Break characters are transmitted

Note: If the SBK bit is set to "1" and then to "0", the transmitter will send a BREAK word at the end of the current word.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

DATA REGISTER (DR)

Read/Write

Reset Value: Undefined

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).
The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 32).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 32).

BAUD RATE REGISTER (BRR)

Read/Write

Reset Value: 00xx xxxx (XXh)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bit 7:6= **SCP[1:0]** *First SCI Prescaler*
These 2 prescaling bits allow several standard clock division ranges:

PR Prescaling factor	SCP1	SCP0
1	0	0
3	0	1
4	1	0
13	1	1

Bit 5:3 = **SCT[2:0]** *SCI Transmitter rate divisor*

These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the transmit rate clock in conventional Baud Rate Generator mode.

TR dividing factor	SCT2	SCT1	SCT0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Note: this TR factor is used only when the ETPR fine tuning factor is equal to 00h; otherwise, TR is replaced by the ETPR dividing factor.

Bit 2:0 = **SCR[2:0]** *SCI Receiver rate divisor.*

These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the receive rate clock in conventional Baud Rate Generator mode.

RR dividing factor	SCR2	SCR1	SCR0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Note: this RR factor is used only when the ERPR fine tuning factor is equal to 00h; otherwise, RR is replaced by the ERPR dividing factor.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

EXTENDED RECEIVE PRESCALER DIVISION REGISTER (ERPR)

Read/Write

Reset Value: 0000 0000 (00h)

Allows setting of the Extended Prescaler rate division factor for the receive circuit.

7							0
ERPR 7	ERPR 6	ERPR 5	ERPR 4	ERPR 3	ERPR 2	ERPR 1	ERPR 0

Bit 7:1 = **ERPR[7:0]** *8-bit Extended Receive Prescaler Register.*

The extended Baud Rate Generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see Figure 34) is divided by the binary factor set in the ERPR register (in the range 1 to 255).

The extended baud rate generator is not used after a reset.

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (ETPR)

Read/Write

Reset Value:0000 0000 (00h)

Allows setting of the External Prescaler rate division factor for the transmit circuit.

7							0
ETPR 7	ETPR 6	ETPR 5	ETPR 4	ETPR 3	ETPR 2	ETPR 1	ETPR 0

Bit 7:1 = **ETPR[7:0]** *8-bit Extended Transmit Prescaler Register.*

The extended Baud Rate Generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see Figure 34) is divided by the binary factor set in the ETPR register (in the range 1 to 255).

The extended baud rate generator is not used after a reset.

Table 16. SCI Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
50	**SR**	TDRE	TC	RDRF	IDLE	OR	NF	FE	-
	Reset Value	1	1	0	0	0	0	0	0
51	**DR**	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
	Reset Value	-	-	-	-	-	-	-	-
52	**BRR**	SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0
	Reset Value	0	0	x	x	x	x	x	x
53	**CR1**	R8	T8		M	WAKE			
	Reset Value	-	-	-	-	-	-	-	-
54	**CR2**	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	Reset Value	0	0	0	0	0	0	0	0
55	**ERPR**	ERPR7	ERPR6	ERPR5	ERPR4	ERPR3	ERPR2	ERPR1	ERPR0
	Reset Value	0	0	0	0	0	0	0	0
57	**ETPR**	ETPR7	ETPR6	ETPR5	ETPR4	ETPR3	ETPR2	ETPR1	ETPR0
	Reset Value	0	0	0	0	0	0	0	0

4.5 SERIAL PERIPHERAL INTERFACE (SPI)

4.5.1 Introduction

The Serial Peripheral Interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves or a system in which devices may be either masters or slaves.

The SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller.

Refer to the Pin Description chapter for the device-specific pin-out.

4.5.2 Main Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- Four master mode frequencies
- Maximum slave mode frequency = fCPU/2.
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability.

4.5.3 General description

The SPI is connected to external devices through 4 alternate pins:

- MISO: Master In Slave Out pin
- MOSI: Master Out Slave In pin
- SCK: Serial Clock pin
- $\overline{SS}$: Slave select pin

A basic example of interconnections between a single master and a single slave is illustrated on Figure 35.

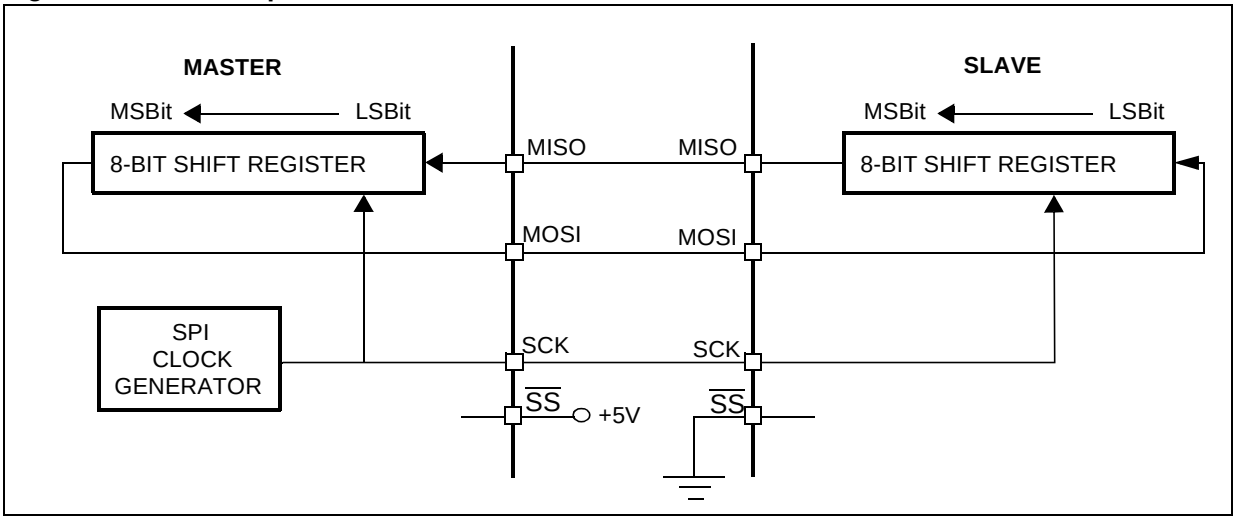
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full bits. A status flag is used to indicate that the I/O operation is complete.

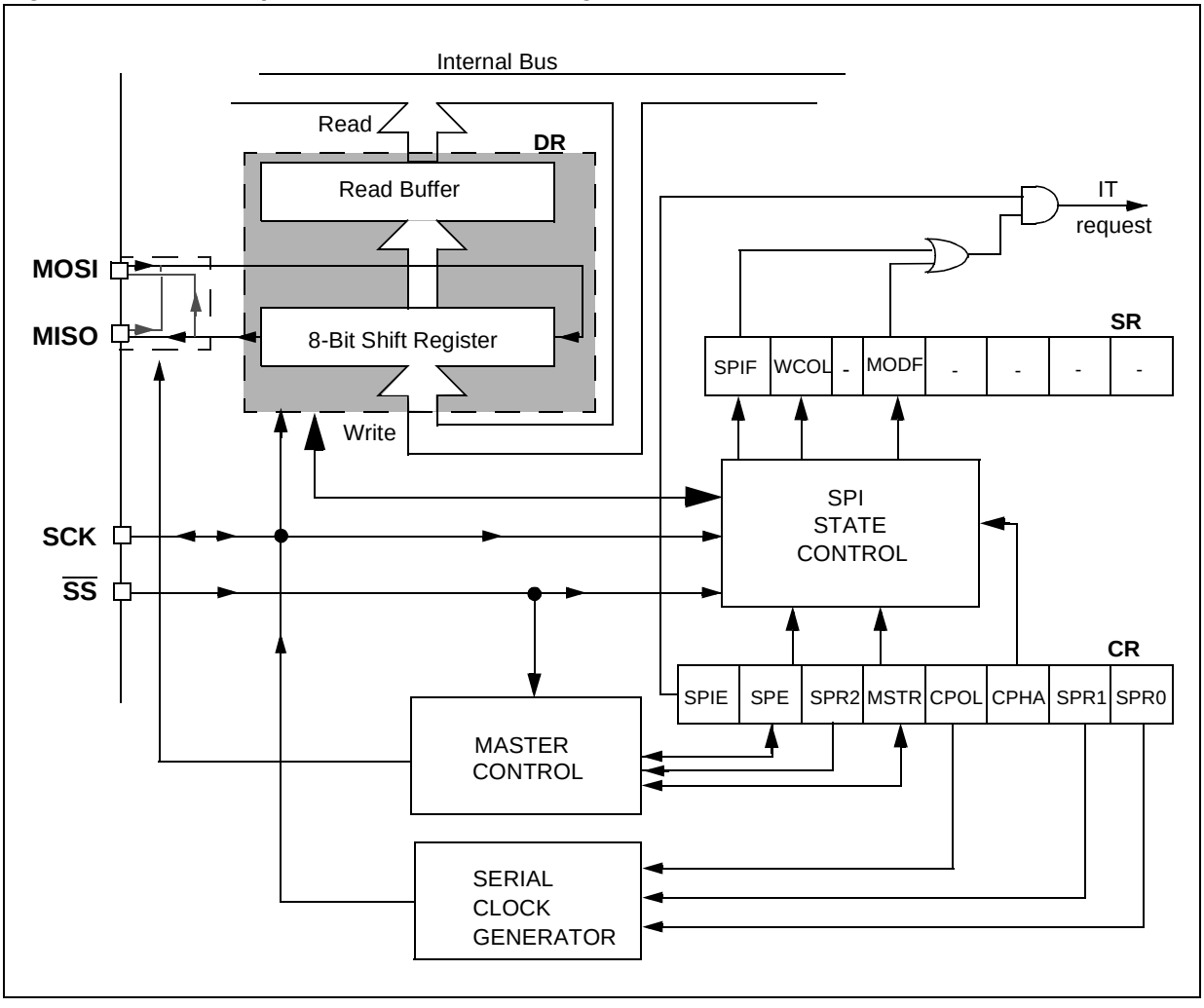
Four possible data/clock timing relationships may be chosen (see Figure 38) but master and slave must be programmed with the same timing mode.

Figure 35. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 36. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4 Functional Description

Figure 35 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

– A Control Register (CR)

– A Status Register (SR)

– A Data Register (DR)

Refer to the CR, SR and DR registers in Section 4.5.7for the bit definitions.

4.5.4.1 Master Configuration

In a master configuration, the serial clock is generated on the SCK pin.

Procedure

– Select the SPR0 & SPR1 bits to define the serial clock baud rate (see CR register).

– Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 38).

– The $\overline{SS}$ pin must be connected to a high level signal during the complete byte transmit sequence.

– The MSTR and SPE bits must be set (they remain set only if the $\overline{SS}$ pin is connected to a high level signal).

In this configuration the MOSI pin is a data output and to the MISO pin is a data input.

Transmit sequence

The transmit sequence begins when a byte is written the DR register.

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

– The SPIF bit is set by hardware

– An interrupt is generated if the SPIE bit is set and the I bit in the CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set
2. A write or a read of the DR register.

Note: While the SPIF bit is set, all writes to the DR

SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4.2 Slave Configuration

In slave configuration, the serial clock is received on the SCK pin from the master device.

The value of the SPR0 & SPR1 bits is not used for the data transfer.

Procedure

– For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 38.

– The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence.

– Clear the MSTR bit and set the SPE bit to assign the pins to alternate function.

In this configuration the MOSI pin is a data input and the MISO pin is a data output.

Transmit Sequence

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

– The SPIF bit is set by hardware

– An interrupt is generated if SPIE bit is set and I bit in CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set.
2. A write or a read of the DR register.

Notes: While the SPIF bit is set, all writes to the DR register are inhibited until the SR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an overrun condition (see Section 4.5.4.6).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the DR register between each data byte transfer to avoid a write collision (see Section 4.5.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4.3 Data Transfer Format

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The serial clock is used to synchronize the data transfer during a sequence of eight clock pulses.

The $\overline{SS}$ pin allows individual selection of a slave device; the other slave devices that are not selected do not interfere with the SPI transfer.

Clock Phase and Clock Polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits.

The CPOL (clock polarity) bit controls the steady state value of the clock when no data is being transferred. This bit affects both master and slave modes.

The combination between the CPOL and CPHA (clock phase) bits selects the data capture clock edge.

Figure 38, shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin, the MOSI pin are directly connected between the master and the slave device.

The $\overline{SS}$ pin is the slave device select input and can be driven by the master device.

The master device applies data to its MOSI pin-clock edge before the capture clock edge.

CPHA bit is set

The second edge on the SCK pin (falling edge if the CPOL bit is reset, rising edge if the CPOL bit is set) is the MSBit capture strobe. Data is latched on the occurrence of the first clock transition.

No write collision should occur even if the $\overline{SS}$ pin stays low during a transfer of several bytes (see Figure 37).

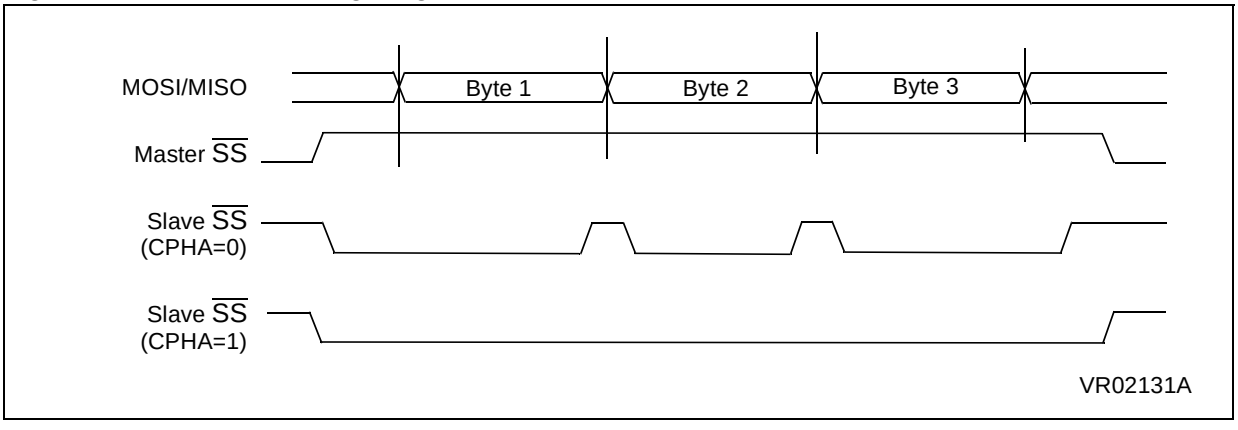
CPHA bit is reset

The first edge on the SCK pin (falling edge if CPOL bit is set, rising edge if CPOL bit is reset) is the MSBit capture strobe. Data is latched on the occurrence of the second clock transition.

This pin must be toggled high and low between each byte transmitted (see Figure 37).

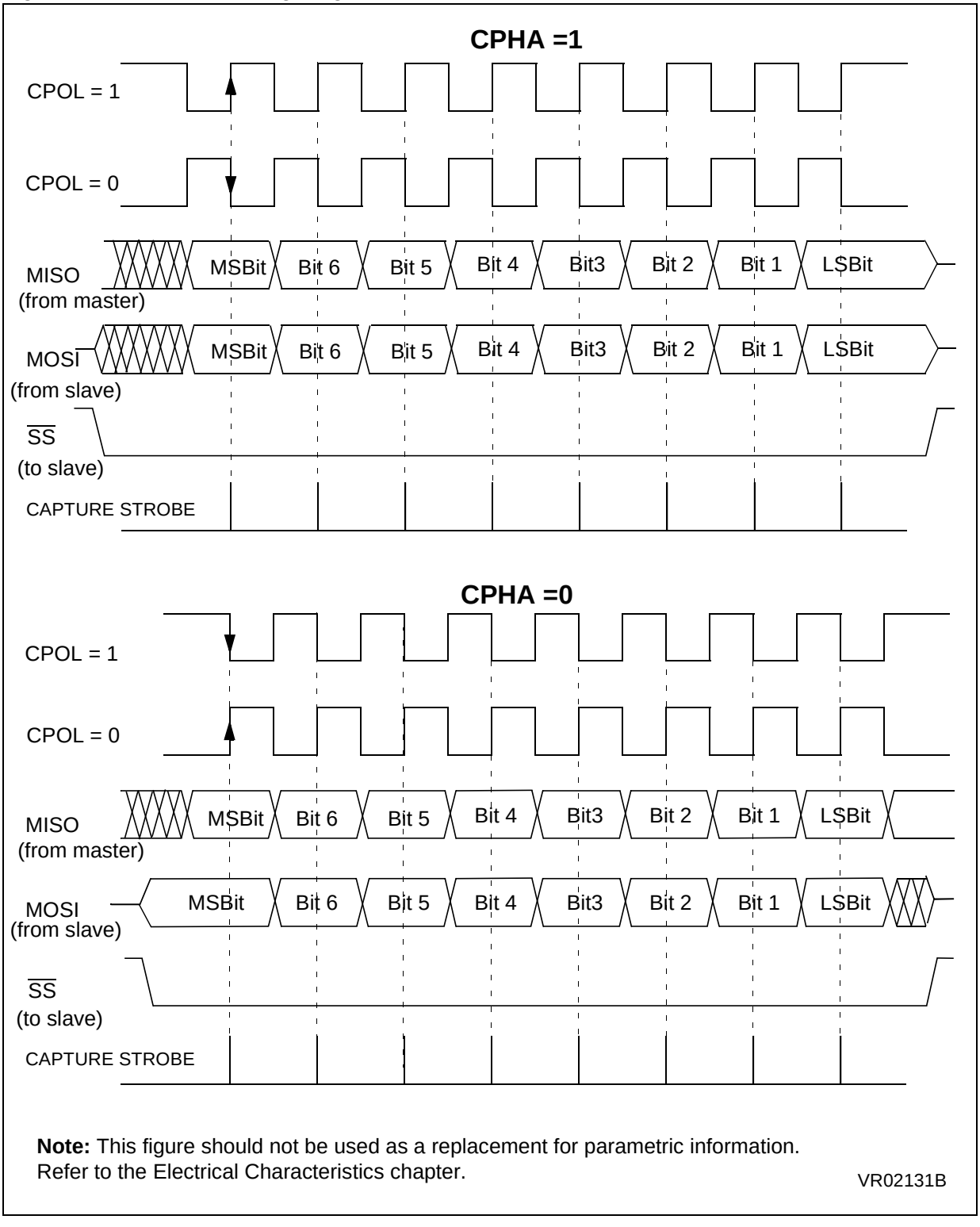
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its DR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the DR without producing a write collision.

Figure 37. CPHA / $\overline{SS}$ Timing Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 38. Data Clock Timing Diagram



Note: This figure should not be used as a replacement for parametric information. Refer to the Electrical Characteristics chapter.

VR02131B

SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4.4 Write Collision Error

A write collision occurs when the software tries to write to the DR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode.

Note: a "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation.

In Slave mode

When the CPHA bit is set:

The slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device DR register and output the MSBit on to the external MISO pin of the slave device.

The $\overline{SS}$ pin low state enables the slave device but the output of the MSBit onto the MISO pin does not take place until the first data transfer clock edge.

When the CPHA bit is reset:

Data is latched on the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when software attempts to write the DR register after its $\overline{SS}$ pin has been pulled low.

For this reason, the $\overline{SS}$ pin must be high, between each data byte transfer, to allow the CPU to write in the DR register without generating a write collision.

In Master mode

Collision in the master device is defined as a write of the DR register while the internal serial clock (SCK) is in the process of transfer.

The $\overline{SS}$ pin signal must be always high on the master device.

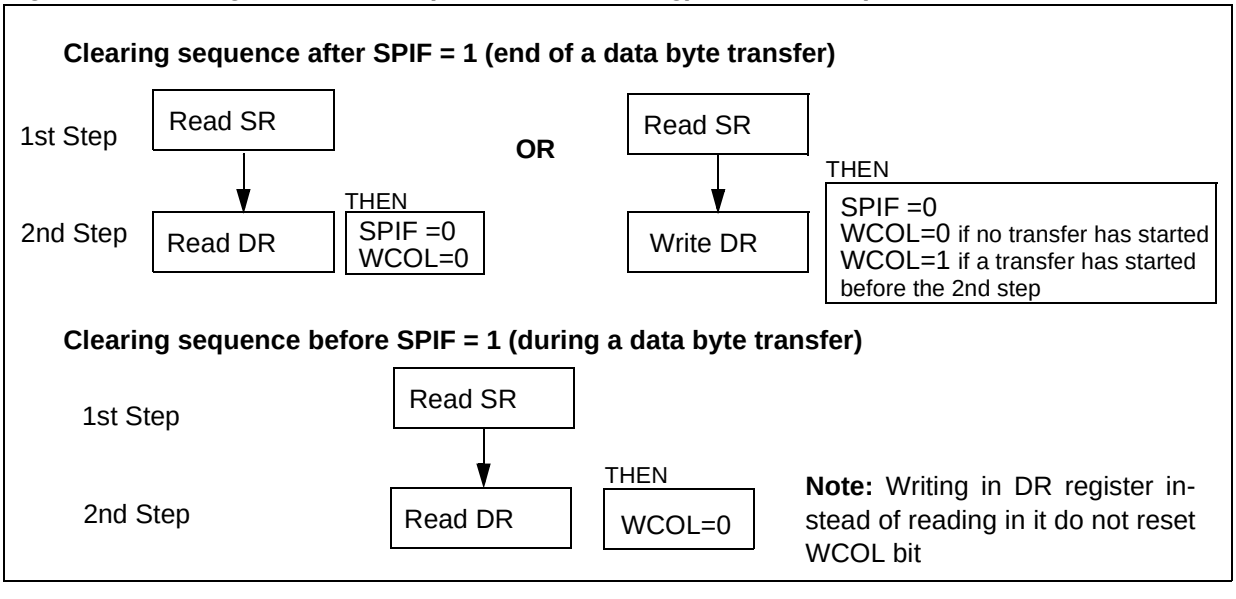
WCOL bit

The WCOL bit in the SR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see Figure 39).

Figure 39. Clearing the WCOL bit (Write Collision Flag) Software Sequence



SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4.5 Master Mode Fault

Master mode fault occurs when the master device has its $\overline{SS}$ pin pulled low, then the MODF bit is set.

Master mode fault affects the SPI peripheral in the following ways:

- The MODF bit is set and an SPI interrupt is generated if the SPIE bit is set.
- The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.
- The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read or write access to the SR register while the MODF bit is set.
2. A write to the CR register.

Notes: To avoid any multiple slave conflicts in the case of a system comprising several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of the MODF bit. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

In a slave device the MODF bit can not be set, but in a multi master configuration the device can be in slave mode with this MODF bit set.

The MODF bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state using an interrupt routine.

4.5.4.6 Overrun Condition

An overrun condition occurs, when the master device has sent several data bytes and the slave device has not cleared the SPIF bit issuing from the previous data byte transmitted.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the DR register returns this byte. All other bytes are lost.

This condition is not detected by the SPI peripheral.

SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.4.7 Single Master and Multimaster Configurations

There are two types of SPI systems:

– Single Master System

– Multimaster System

Single Master System

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see Figure 40).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written its DR register.

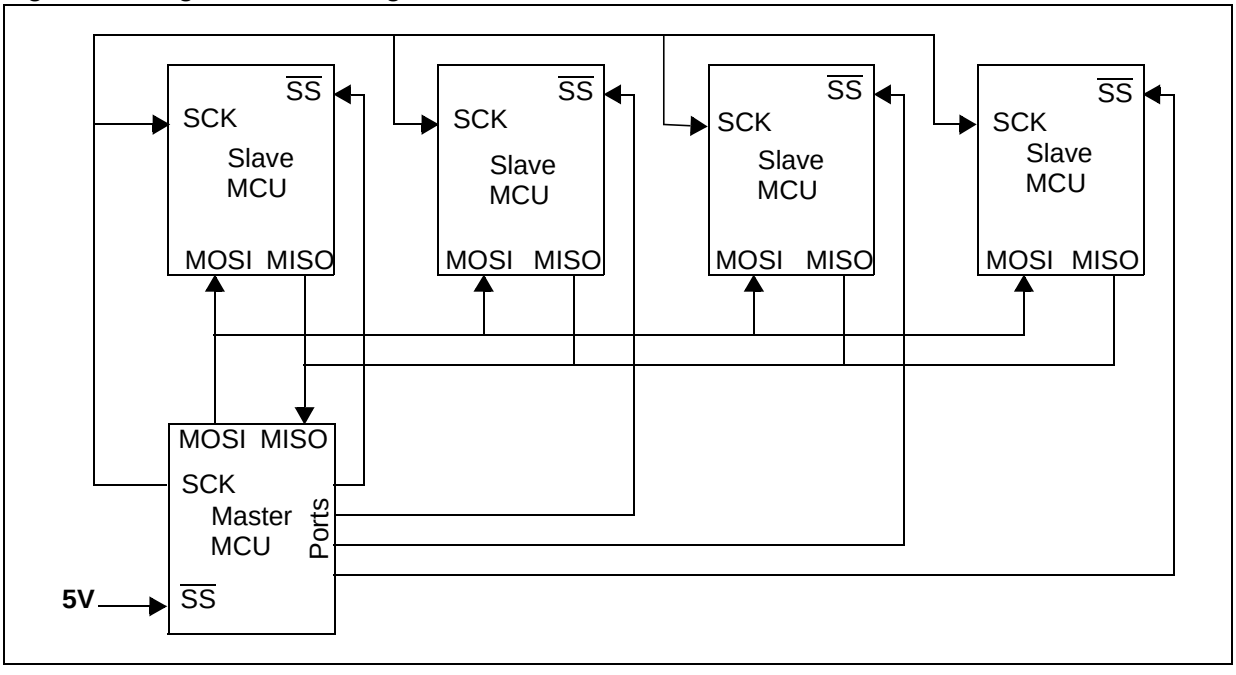
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multi-master System

A multi-master system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multi-master system is principally handled by the MSTR bit in the CR register and the MODF bit in the SR register.

Figure 40. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.5 Low Power Modes

Mode	Description
WAIT	No effect on SPI. SPI interrupt events cause the device to exit from WAIT mode.
HALT	SPI registers are frozen. In HALT mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

4.5.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	No
Master Mode Fault Event	MODF		Yes	No

Note: The SPI interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

SERIAL PERIPHERAL INTERFACE (Cont'd)

4.5.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0000xxxx (0xh)

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

Bit 7 = **SPIE** *Serial peripheral interrupt enable.*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SPI interrupt is generated whenever SPIF=1 or MODF=1 in the SR register

Bit 6 = **SPE** *Serial peripheral output enable.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 4.5.4.5 Master Mode Fault).
0: I/O port connected to pins
1: SPI alternate functions connected to pins

The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins.

Bit 5 = **SPR2** *Divider Enable*.

this bit is set and cleared by software and it is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 17.
0: Divider by 2 enabled
1: Divider by 2 disabled

Bit 4 = **MSTR** *Master.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 4.5.4.5 Master Mode Fault).
0: Slave mode is selected
1: Master mode is selected, the function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

Bit 3 = **CPOL** *Clock polarity.*
This bit is set and cleared by software. This bit determines the steady state of the serial Clock. The CPOL bit affects both the master and slave modes.
0: The steady state is a low value at the SCK pin.
1: The steady state is a high value at the SCK pin.

Bit 2 = **CPHA** *Clock phase.*
This bit is set and cleared by software.
0: The first clock transition is the first data capture edge.
1: The second clock transition is the first capture edge.

Bit 1:0 = **SPR[1:0]** *Serial peripheral rate.*
These bits are set and cleared by software.Used with the SPR2 bit, they select one of six baud rates to be used as the serial clock when the device is a master.

These 2 bits have no effect in slave mode.

Table 17. Serial Peripheral Baud Rate

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/2$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SERIAL PERIPHERAL INTERFACE (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	-	MODF	-	-	-	-

Bit 7 = **SPIF** *Serial Peripheral data transfer flag.*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE=1 in the CR register. It is cleared by a software sequence (an access to the SR register followed by a read or write to the DR register).
0: Data transfer is in progress or has been approved by a clearing sequence.
1: Data transfer between the device and an external device has been completed.

Note: While the SPIF bit is set, all writes to the DR register are inhibited.

Bit 6 = **WCOL** *Write Collision status.*

This bit is set by hardware when a write to the DR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 39).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = Unused.

Bit 4 = **MODF** *Mode Fault flag.*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 4.5.4.5 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the CR register. This bit is cleared by a software sequence (An access to the SR register while MODF=1 followed by a write to the CR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3-0 = Unused.

DATA I/O REGISTER (DR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The DR register is used to transmit and receive data on the serial bus. In the master device only a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

Warning:

A write to the DR register places data directly into the shift register for transmission.

A write to the the DR register returns the value located in the buffer and not the contents of the shift register (See Figure 36).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 18. SPI Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
21	**DR**	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	x	x	x	x	x	x	x	x
22	**CR**	SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
	Reset Value	0	0	0	0	x	x	x	x
23	**SR**	SPIF	WCOL	-	MODF	-	-	-	-
	Reset Value	0	0	0	0	0	0	0	0

5 INSTRUCTION SET

5.1 ST7 ADDRESSING MODES

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two sub-modes called long and short:

– Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.

– Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 19. ST7 Addressing Mode Overview

Mode			Syntax	Destination/ Source	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0 (with X register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC-128/PC+127[1)]			+ 1
Relative	Indirect		jrne [$10]	PC-128/PC+127[1)]	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

Note 1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

ST7 ADDRESSING MODES (Cont'd)

5.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

5.1.2 Immediate

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the the operand value. .

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

5.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two sub-modes:

Direct (short)

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

5.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three sub-modes:

Indexed (No Offset)

There is no offset, (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

5.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

ST7 ADDRESSING MODES (Cont'd)

5.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 20. Instructions Supporting Direct, Indexed, Indirect and Indirect Indexed Addressing Modes

Long and Short Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

5.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/ Indirect Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

The offset is defined in memory, which address follows the opcode.

5.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, T_J, in Celsius can be obtained from:

T_J= TA + PD x RthJA

Where: T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

P_D = P_{INT} + P_{PORT}.

P_{INT} = I_{DD} x V_{DD} (chip internal power).

P_{PORT} =Port power dissipation determined by the user)

Symbol	Parameter	Value	Unit
V_{DD}	Digital Supply Voltage	-0.3 to 6.0	V
V_{DDA}	Analog Supply and Reference Voltage	V_{DD} - 0.3 to V_{DD} + 0.3	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_{AI}	Analog Input Voltage (A/D Converter)	V_{SS} - 0.3 to V_{DD} + 0.3 V_{SSA}-0.3 to V_{DDA}+0.3	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
IV_{DD}	Total Current into V_{DD} (source)	100	mA
IV_{SS}	Total Current out of V_{SS} (sink)	100	mA
T_J	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note: Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

6.2 RECOMMENDED OPERATING CONDITIONS

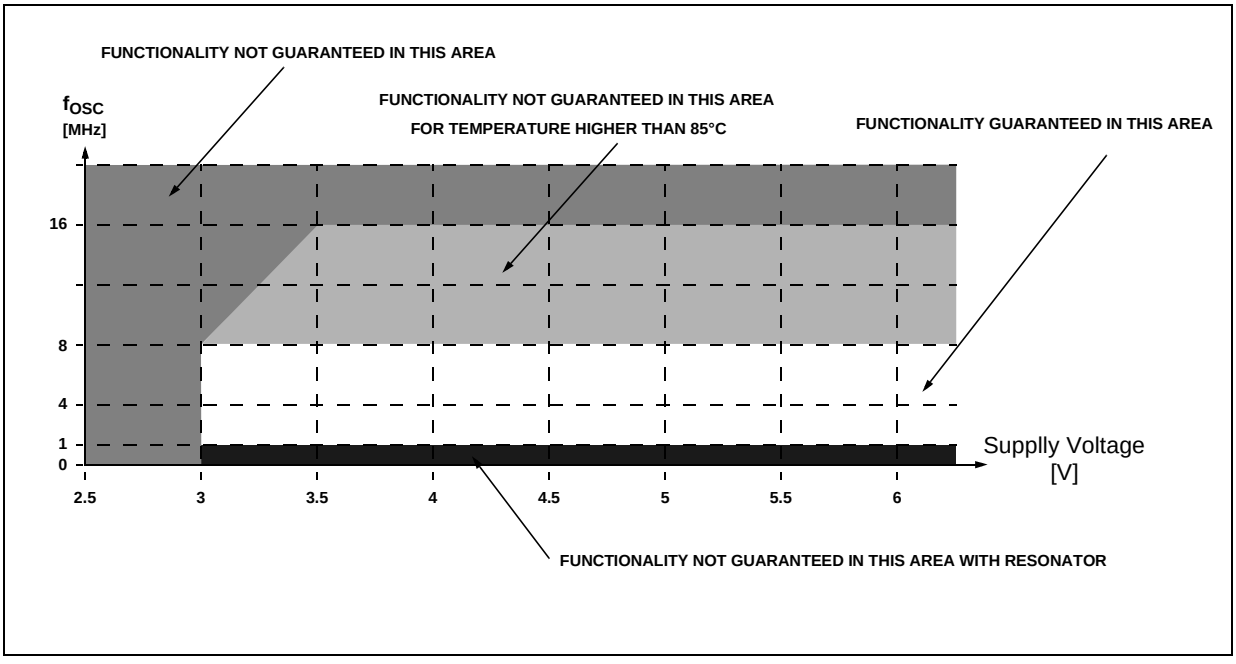
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version	0		70	°C
		6 Suffix Version	-40		85	°C
		3 Suffix Version	-40		125	°C
V_{DD}	Operating Supply Voltage	f_{OSC} = 16 MHz (1 & 6 Suffix) f_{OSC} = 8 MHz	3.5 1) 3.0		5.5 5.5	V
f_{OSC}	Oscillator Frequency	V_{DD} = 3.0V V_{DD} = 3.5V (1 & 6 Suffix)	0 2) 0 2)		8 16	MHz

Note

1) A safe reset (with Low Voltage Detector option) is not guaranteed at 16 MHz.

2) A/D operation and Oscillator start-up are not guaranteed below 1MHz.

Figure 41. Maximum Operating Frequency (f_{OSC}) Versus Supply Voltage (V_{DD})



6.3 DC ELECTRICAL CHARACTERISTICS

(T_A = -40°C to +125°C and V_{DD} = 5V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage All Input pins	3V < V_{DD} < 5.5V			V_{DD} x 0.3	V
V_{IH}	Input High Level Voltage All Input pins	3V < V_{DD} < 5.5V	V_{DD} x 0.7			V
V_{HYS}	Hysteresis Voltage [1] All Input pins			400		mV
V_{OL}	Low Level Output Voltage All Output pins	I_{OL} = +10µA I_{OL} = + 2mA			0.1 0.4	V
	Low Level Output Voltage High Sink I/O pins	I_{OL} = +10µA I_{OL} = +10mA I_{OL} = + 15mA I_{OL} = + 20mA, T_A = 85°C max			0.1 1.5 3.0 3.0	
V_{OH}	High Level Output Voltage All Output pins	I_{OH} = - 10µA I_{OH}= - 2mA	4.9 4.2			V
I_{IL} I_{IH}	Input Leakage Current All Input pins but RESET [4]	V_{IN} = V_{SS} (No Pull-up configured) V_{IN} = V_{DD}		0.1	1.0	µA
I_{IH}	Input Leakage Current RESET pin	V_{IN} = V_{DD}		0.1	1.0	
R_{ON}	Reset Weak Pull-up R_{ON}	V_{IN} > V_{IH} V_{IN} < V_{IL}	20 60	40 120	80 240	kΩ
R_{PU}	I/O Weak Pull-up R_{PU}	V_{IN} < V_{IL}		100		kΩ
I_{DD}	Supply Current in RUN Mode [2]	f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz		3.5 6 11	7 12 20	mA
	Supply Current in SLOW Mode [2]	f_{OSC} = 4 MHz, f_{CPU}= 125 kHz f_{OSC} = 8 MHz, f_{CPU}= 250 kHz f_{OSC} = 16 MHz, f_{CPU}= 500 kHz		1.5 2.5 4.5	3 5 9	mA
	Supply Current in WAIT Mode [3]	f_{OSC} = 4MHz, f_{CPU} = 2MHz f_{OSC} = 8MHz, f_{CPU} = 4 MHz f_{OSC} = 16MHz, f_{CPU} = 8 MHz		2 4 6.5	4 8 12	mA
	Supply Current in WAIT- MINIMUM Mode[5]	f_{OSC} = 4 MHz, f_{CPU}= 125 kHz f_{OSC} = 8 MHz, f_{CPU}= 250 kHz f_{OSC} = 16 MHz, f_{CPU}= 500 kHz		0.8 1 1.6	1.5 2 3.5	mA
	Supply Current in HALT Mode	I_{LOAD} = 0mA without LVD, T_A = 85°C max I_{LOAD} = 0mA without LVD I_{LOAD} = 0mA with LVD		1 5 70	10 20 100	µA

Notes:

1. Hysteresis voltage between switching levels. Based on characterisation results, not tested.
2. CPU running with memory access, no DC load or activity on I/O's; clock input (OSCIN) driven by external square wave.
3. No DC load or activity on I/O's; clock input (OSCIN) driven by external square wave.
4. Except OSCIN and OSCOUT
5. WAIT Mode with SLOW Mode selected. Based on characterisation results, not tested.

6.4 RESET CHARACTERISTICS

(T_A=-40...+125°C and V_{DD}=5V±10% unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
R_{ON}	Reset Weak Pull-up R_{ON}	$V_{IN} > V_{IH}$ $V_{IN} < V_{IL}$	20 60	40 120	80 240	kΩ
t_{RESET}	Pulse duration generated by watch-dog and POR reset			1		µs
t_{PULSE}	Minimum pulse duration to be applied on external $\overline{RESET}$ pin		10 [1]			ns

Note:

1) These values given only as design guidelines and are not tested.

6.5 OSCILLATOR CHARACTERISTICS

(T_A = -40°C to +125°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
g_m	Oscillator transconductance		2		9	mA/V
f_{OSC}	Crystal frequency		1		16	MHz
t_{start}	Osc. start up time	V_{DD} = 5V±10%			50	ms

6.6 PERIPHERAL CHARACTERISTICS

Low Voltage Detection Reset Electrical Specifications (Option)						
Symbol	**Parameter**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
V_{LVDUP}	LVD Reset Trigger, V_{DD} rising edge	f_{OSC} = 8 MHz max[1].	3.6 [2]	3.85	4.1	V
$V_{LVDDOWN}$	LVD Reset Trigger, V_{DD} falling edge		3.35	3.6	3.85	V
V_{LVDHYS}	LVD Reset Trigger, hysteresis[2]			250		mV

Notes:

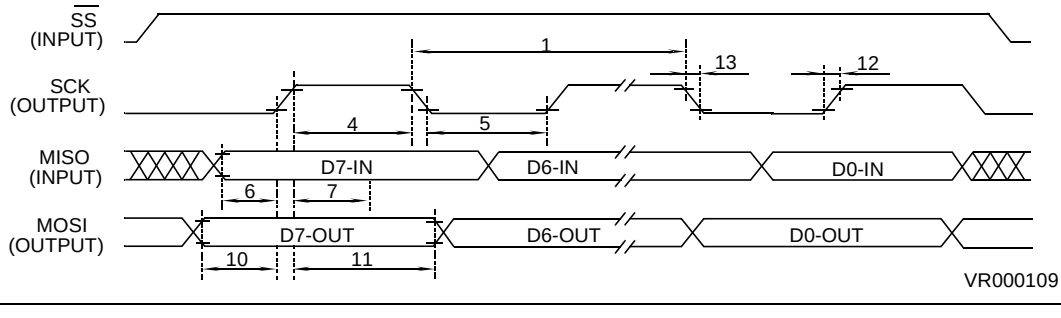
1. The safe reset cannot be guaranted by the LVD when fosc is greater than 8MHz.
2. Based on characterisation results, not tested.

PERIPHERAL CHARACTERISTICS (Cont'd)

Serial Peripheral Interface						
Ref.	**Symbol**	**Parameter**	**Condition**	**Value**		**Unit**
				Min.	**Max.**	
	f_{SPI}	SPI frequency	Master Slave	1/128 dc	1/4 1/2	f_{CPU}
1	t_{SPI}	SPI clock periode	Master Slave	4 2		t_{CPU}
2	t_{Lead}	Enable lead time	Slave	120		ns
3	t_{Lag}	Enable lag time	Slave	120		ns
4	t_{SPI_H}	Clock (SCK) high time	Master Slave	100 90		ns
5	t_{SPI_L}	Clock (SCK) low time	Master Slave	100 90		ns
6	t_{SU}	Data set-up time	Master Slave	100 100		ns
7	t_H	Data hold time (inputs)	Master Slave	100 100		ns
8	t_A	Access time (time to data active from high impedance state)	Slave	0	120	ns
9	t_{Dis}	Disable time (hold time to high impedance state)			240	ns
10	t_V	Data valid	Master (before capture edge) Slave (after enable edge)	0.25	 120	t_{CPU} ns
11	t_{Hold}	Data hold time (outputs)	Master (before capture edge) Slave (after enable edge)	0.25 0		t_{CPU} ns
12	t_{Rise}	Rise time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs
13	t_{Fall}	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs

Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI Timing Diagram

Figure 42. SPI Master Timing Diagram CPHA=0, CPOL=0



PERIPHERAL CHARACTERISTICS (Cont'd)

Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI Timing Diagram

Figure 43. SPI Master Timing Diagram CPHA=0, CPOL=1

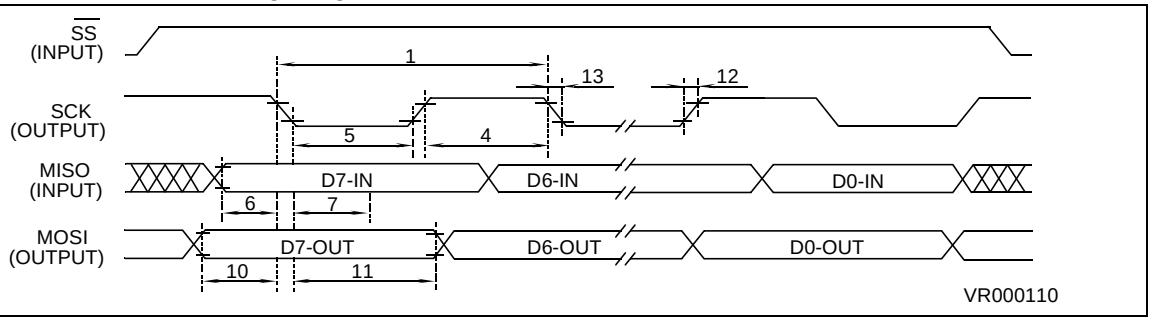


Figure 44. SPI Master Timing Diagram CPHA=1, CPOL=0

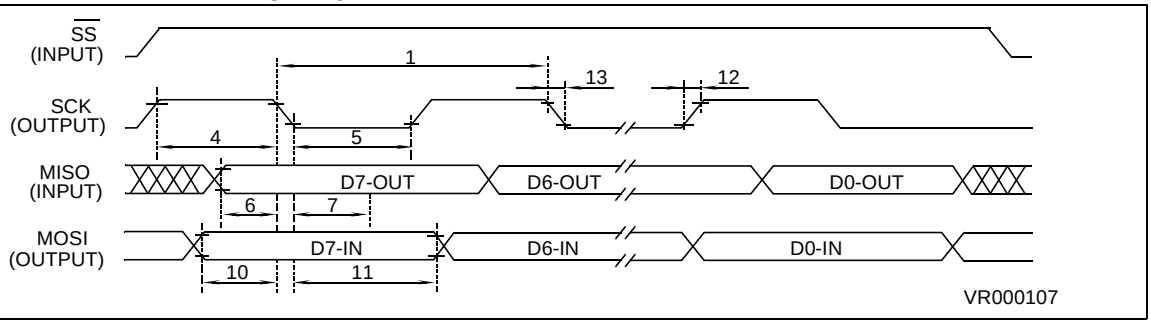
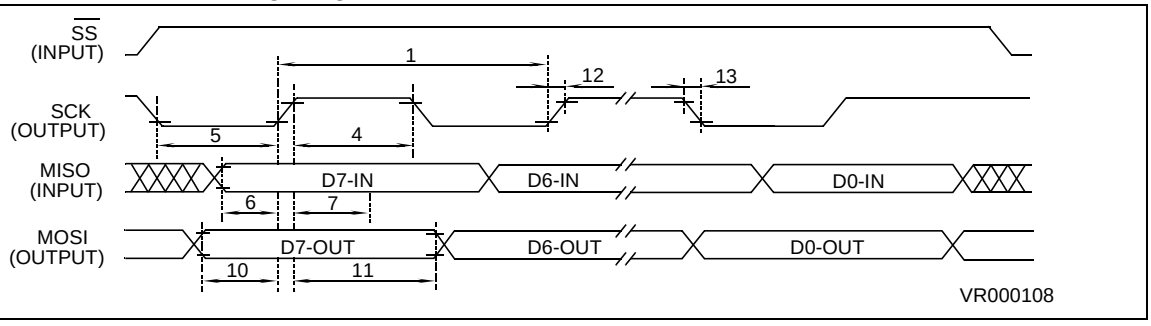


Figure 45. SPI Master Timing Diagram CPHA=1, CPOL=1



PERIPHERAL CHARACTERISTICS (Cont'd)

Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI Timing Diagram

Figure 46. SPI Slave Timing Diagram CPHA=0, CPOL=0

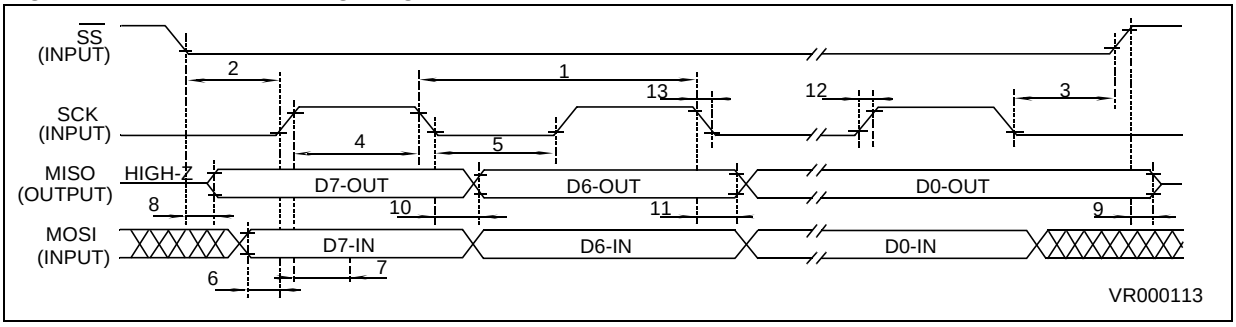


Figure 47. SPI Slave Timing Diagram CPHA=0, CPOL=1

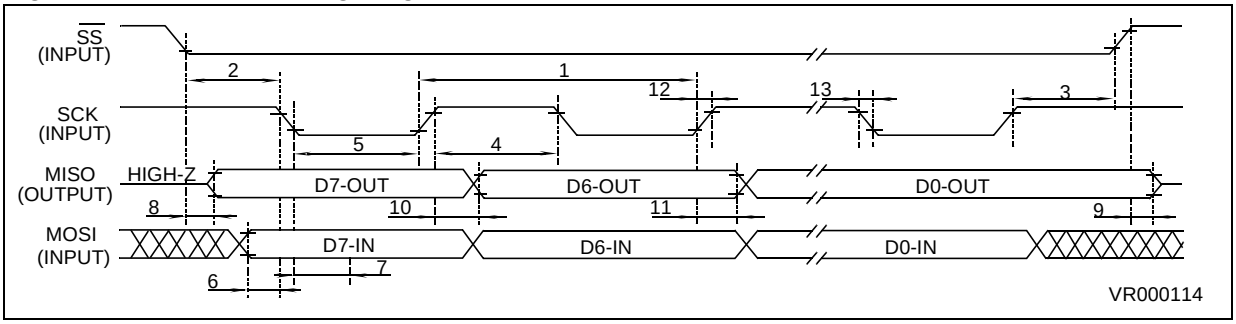


Figure 48. SPI Slave Timing Diagram CPHA=1, CPOL=0

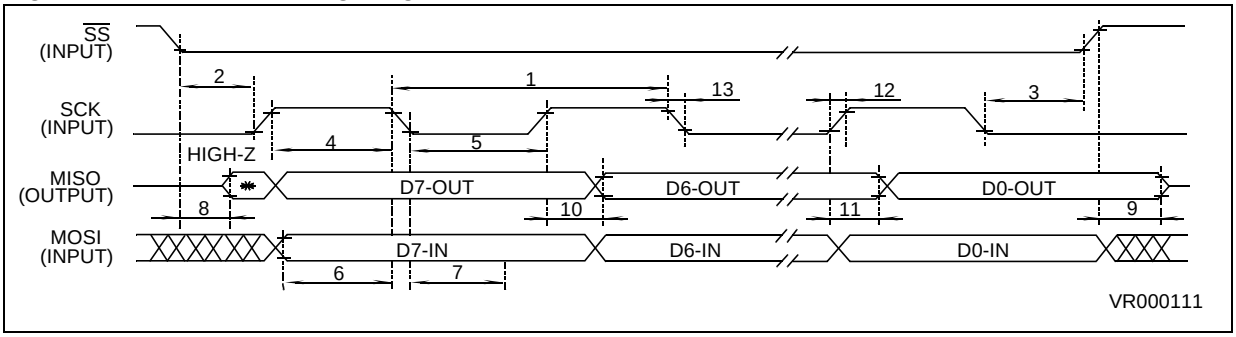
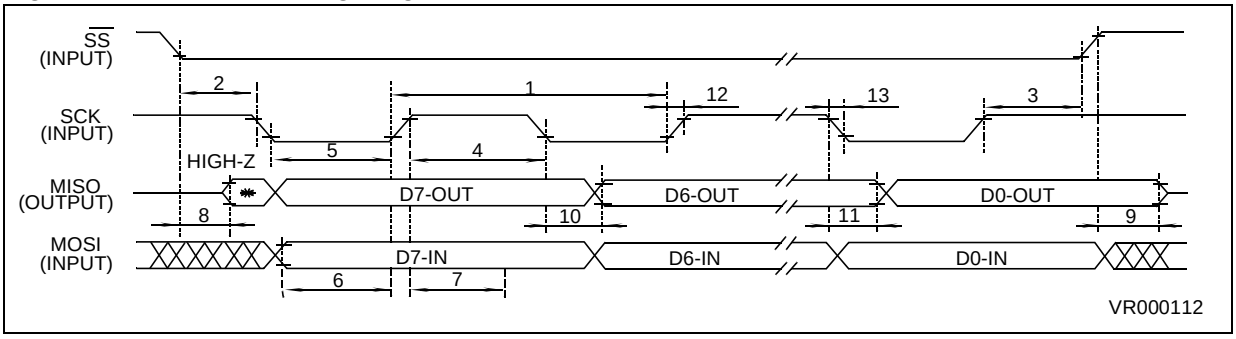


Figure 49. SPI Slave Timing Diagram CPHA=1, CPOL=1



7 GENERAL INFORMATION

7.1 EPROM ERASURE

EPROM version devices are erased by exposure to high intensity UV light admitted through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current.

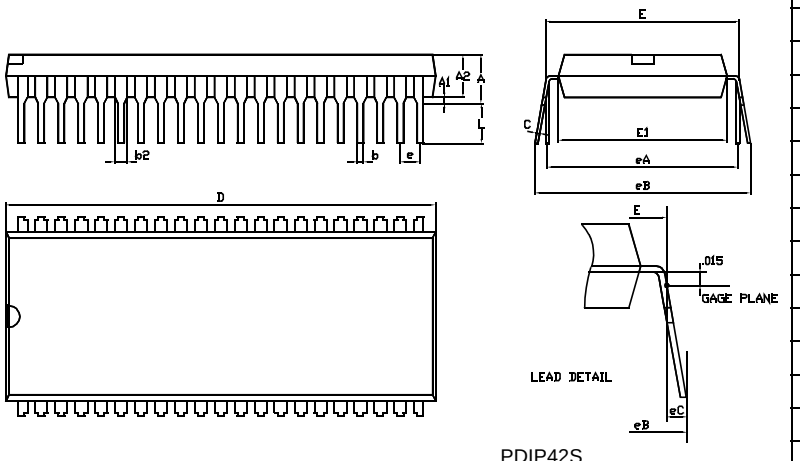
It is recommended that the EPROM devices be kept out of direct sunlight, since the UV content of sunlight can be sufficient to cause functional failure. Extended exposure to room level fluorescent lighting may also cause erasure.

An opaque coating (paint, tape, label, etc...) should be placed over the package window if the product is to be operated under these lighting conditions. Covering the window also reduces I_{DD} in power-saving modes due to photo-diode leakage currents.

An Ultraviolet source of wave length 2537 Å yielding a total integrated dosage of 15 Watt-sec/cm^2 is required to erase the device. It will be erased in 15 to 20 minutes if such a UV lamp with a 12mW/cm^2 power rating is placed 1 inch from the device window without any interposed filters.

7.2 PACKAGE MECHANICAL DATA

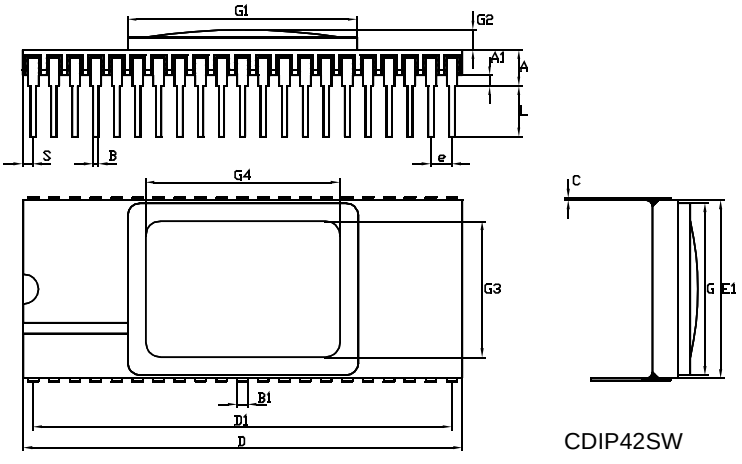
Figure 50. 42-Pin Shrink Plastic Dual In-Line Package, 600-mil Width



PDIP42S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
A2	3.05	3.81	4.57	0.120	0.150	0.180
b		0.46	0.56		0.018	0.022
b2		1.02	1.14		0.040	0.045
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.58	36.83	37.08	1.440	1.450	1.460
E	15.24		16.00	0.600		0.630
E1	12.70	13.72	14.48	0.500	0.540	0.570
e		1.78			0.070	
eA		15.24			0.600	
eB			18.54			0.730
eC			1.52	0.000		0.060
L	2.54	3.30	3.56	0.100	0.130	0.140
	Number of Pins					
N	42					

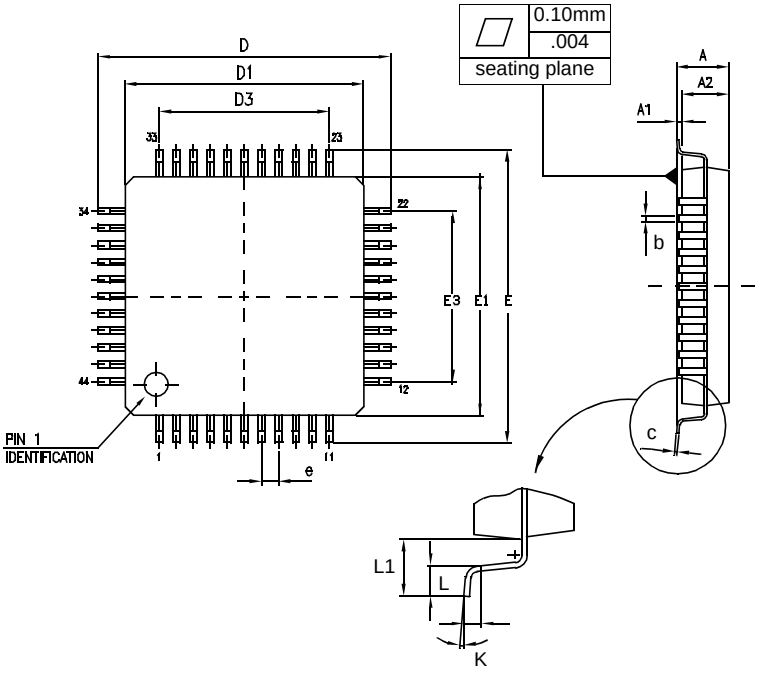
Figure 51. 42-Pin Shrink Ceramic Dual In-Line Package, 600-mil Width



CDIP42SW

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			4.01			0.158
A1	0.76			0.030		
B	0.38	0.46	0.56	0.015	0.018	0.022
B1	0.76	0.89	1.02	0.030	0.035	0.040
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.68	37.34	38.00	1.444	1.470	1.496
D1		35.56			1.400	
E1	14.48	14.99	15.49	0.570	0.590	0.610
e		1.78			0.070	
G	14.12	14.38	14.63	0.556	0.566	0.576
G1	18.69	18.95	19.20	0.736	0.746	0.756
G2		1.14			0.045	
G3	11.05	11.30	11.56	0.435	0.445	0.455
G4	15.11	15.37	15.62	0.595	0.605	0.615
L	2.92		5.08	0.115		0.200
S		0.89			0.035	
	Number of Pins					
N	42					

Figure 52. 44-Pin Thin Quad Flat Package

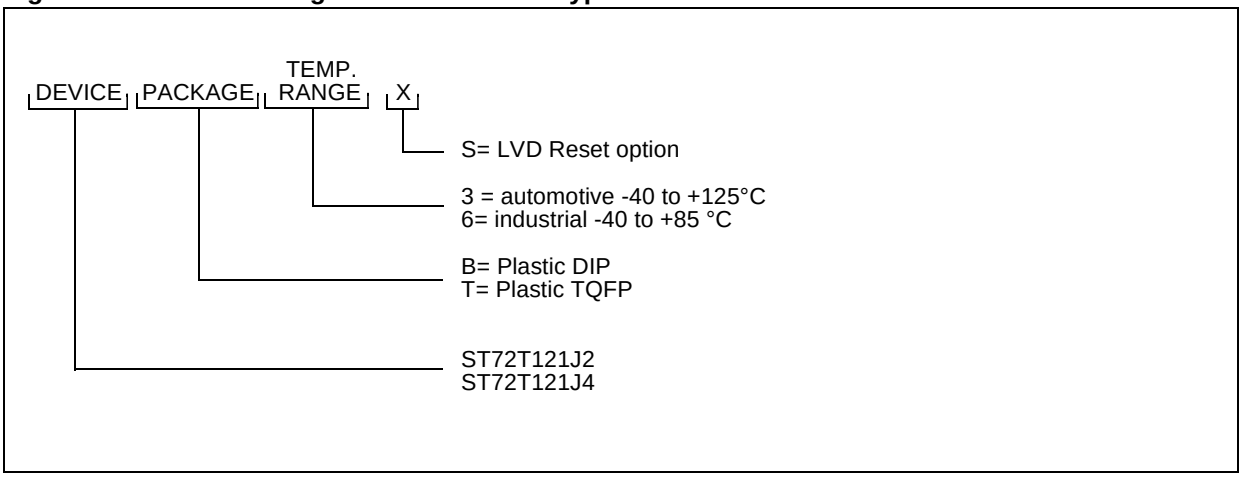


Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.015	0.018
c	0.09		0.20	0.004		0.008
D		12.00			0.472	
D1		10.00			0.394	
D3		8.00			0.315	
E		12.00			0.472	
E1		10.00			0.394	
E3		8.00			0.315	
e		0.80			0.031	
K	0°	3.5°	7°			
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	44					

7.3 ORDERING INFORMATION

Each device is available for production in user programmable version (OTP). OTP devices are shipped to customer with a default blank content FFh. There is one common EPROM version for debugging and prototyping which features the maximum memory size and peripherals of the family. Care must be taken to only use resources available on the target device.

Figure 53. OTP User Programmable Device Types



Notes:

– The ST72E121J4D0 (CERDIP 25 °C) is used as the EPROM versions for the above devices.

– The ROM versions are supported by the ST72124 family.

8 SUMMARY OF CHANGES

Change Description (Rev. 1.5 to 1.6)	Page
Added new External Connections section	7
Removed RP external resistor	15
Changed ORed to ANDed in External interrupts paragraph, to read “If several input pins, connected to the same interrupt vector, are configured as interrupts, their signals are logically ANDed before entering the edge/level detection block”.	18 and 24
Added note ”Any modification of one of these two bits resets the interrupt request related to this interrupt vector.”	23
Added clamping diodes to I/O pin figure and table	26
Added sections on low power modes and interrupts to peripheral descriptions	31, 44, 57, 71
Changed 16-bit Timer chapter	33 to 49
Added details to description of FOLV1 and FOLV2 bits	45
Added Reset characteristics section	84
Added min. value for V_{LVDUP}	84
Removed ST72121 ROM device (supported by ST72124)	
Change Description (Rev. 1.6 to 1.7)	
SPR2 bit reinstated in SPI chapter	62 to 74