ST72321J, ST72324J/K

8-BIT MCU WITH NESTED INTERRUPTS, FLASH, 10-BIT ADC, FIVE TIMERS, SPI, SCI, I^{2}C INTERFACE

PRODUCT PREVIEW

■ **Memories**
 – 8 to 60K dual voltage High Density Flash (HD-Flash) or ROM with read-out protection capability. In-Application Programming and In-Circuit Programming for HDFlash devices
 – 384 to 2K bytes RAM

■ **Clock, Reset And Supply Management**
 – Enhanced reset system
 – Enhanced low voltage supervisor (LVD) for main supply with 3 programmable reset thresholds and auxiliary voltage detector (AVD) with interrupt capability
 – Clock sources: crystal/ceramic resonator oscillators or RC oscillator with PLL and by-pass for external clock, clock security system.
 – Four Power Saving Modes: Halt, Active-Halt, Wait and Slow

■ **Interrupt Management**
 – Nested interrupt controller
 – 10 interrupt vectors plus TRAP and RESET
 – 9/6 external interrupt lines (on 4 vectors)

■ **Up to 32 I/O Ports**
 – 32/24 multifunctional bidirectional I/O lines
 – 22/17 alternate function lines
 – 12/10 high sink outputs

■ **5 Timers**
 – Main Clock Controller with: Real time base, Beep and Clock-out capabilities
 – Configurable watchdog timer
 – Two 16-bit timers with: 2 input captures, 2 output compares, external clock input on one timer, PWM and pulse generator modes
 – 8-bit PWM Auto-reload timer with: 4 PWM outputs, output compare and time base interrupt, external clock with event detector

TQFP44
10 x 10

TQFP32
7 x 7

SDIP42
600 mil

SDIP32
200 mil
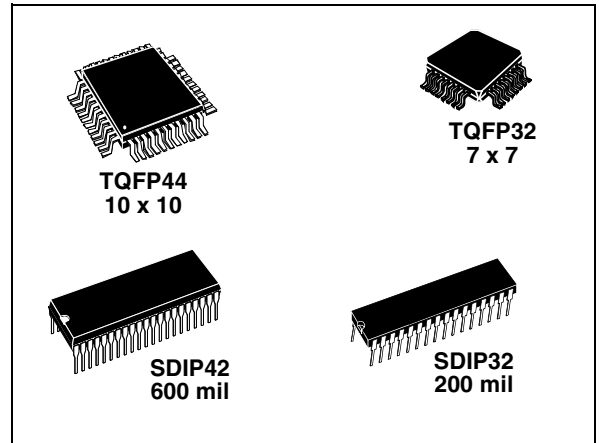
■ **3 Communication Interfaces**
 – SPI synchronous serial interface
 – SCI asynchronous serial interface
 – I^{2}C multimaster interface

■ **1 Analog Peripheral**
 – 10-bit ADC with 12 input pins

■ **Instruction Set**
 – 8-bit Data Manipulation
 – 63 Basic Instructions
 – 17 main Addressing Modes
 – 8 x 8 Unsigned Multiply Instruction
 – True Bit Manipulation

■ **Development Tools**
 – Full hardware/software development package
 – In-Circuit Testing capability

Device Summary

Features	ST72(F)321J9	ST72(F)321J7	ST72(F)324(J/K)6	ST72(F)324(J/K)4	ST72324(J/K)2
Program memory - bytes	60K	48K	32K	16K	8K
RAM (stack) - bytes	2K (256)	1536 (256)	1024 (256)	512 (256)	384 (256)
Peripherals	Watchdog, 16-bit Timers, SPI, SCI, ADC				
	PWM ART, I^{2}C		-	-	-
Operating Supply vs. Frequency	FLASH: 2.7 to 5.5V with f$_{CPU}$≤4MHz or 3.85V to 5.5V with f$_{CPU}$≤8MHz ROM: 2.7 to 5.5V with f$_{CPU}$≤8MHz				ROM: 2.7 to 5.5 V with f$_{CPU}$≤8MHz
Temperature Range	0°C to 70°C / -10°C to +85 °C / -40°C to +85 °C / -40°C to +105°C / -40°C to +125°C				
Packages	TQFP44 10x10 (JxT)		SDIP42 (JxB), TQFP44 10x10 (JxT), SDIP32 (KxB), TQFP32 7x7 (KxT)		

Rev. 1.3

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1 INTRODUCTION

The ST72321J, ST72324K and ST72324J devices are members of the ST7 microcontroller family. They can be grouped as follows:

– The 32-pin ST72324K devices are designed for mid-range applications

– The 42/44-pin ST72321/4J devices target the same range of applications requiring more than 24 I/O ports.

All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruc-

tion set and are available with FLASH or ROM program memory.

Under software control, all devices can be placed in WAIT, SLOW, ACTIVE-HALT or HALT mode, reducing power consumption when the application is in idle or stand-by state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

Figure 1. Device Block Diagram

[1]On some devices only, see Device Summary on page 1
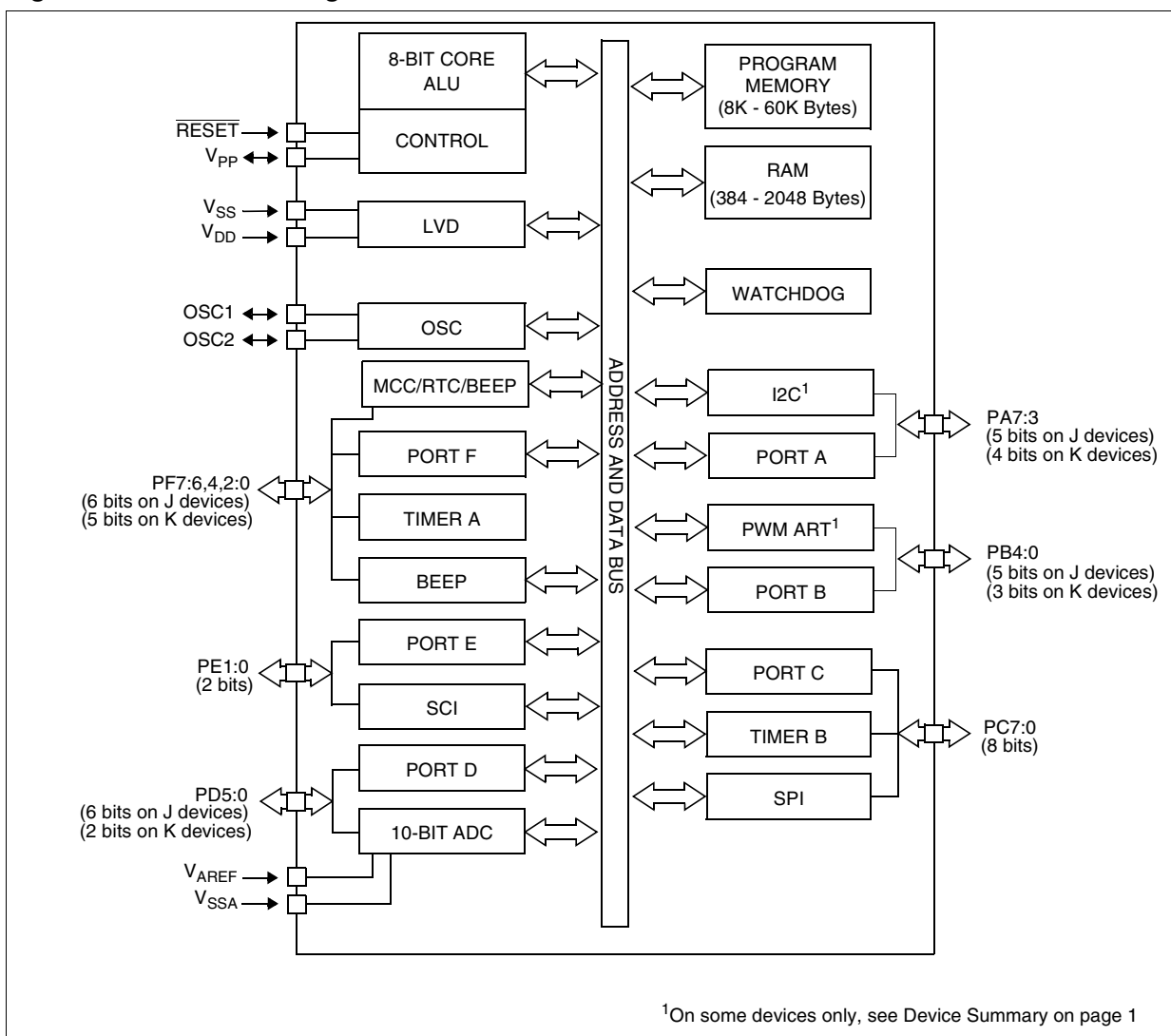
2 PIN DESCRIPTION

Figure 2. 42-Pin SDIP and 44-Pin TQFP Package Pinouts

(HS) 20mA high sink capability
eix associated external interrupt vector
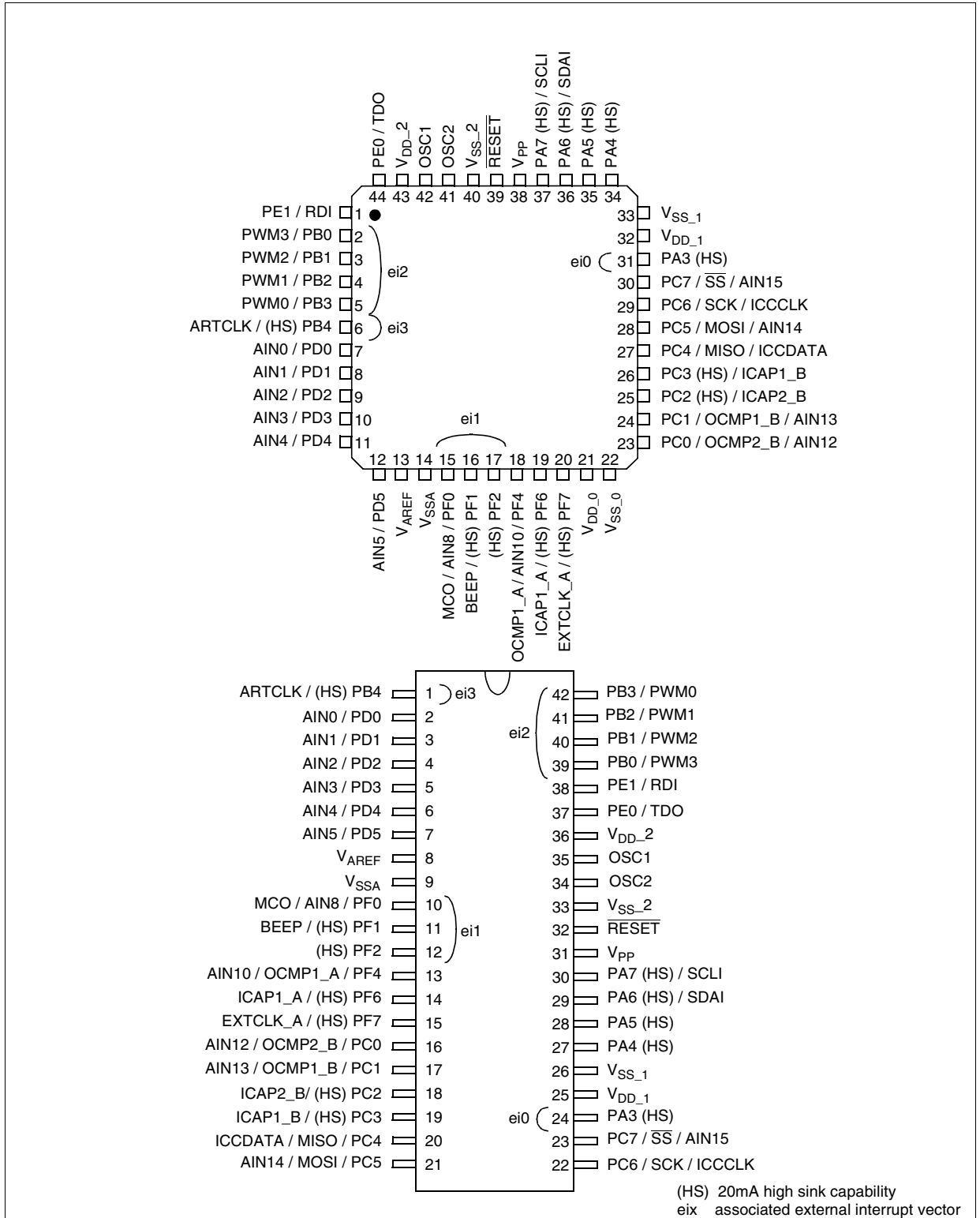
PIN DESCRIPTION (Cont'd)

Figure 3. 32-Pin SDIP Package Pinout
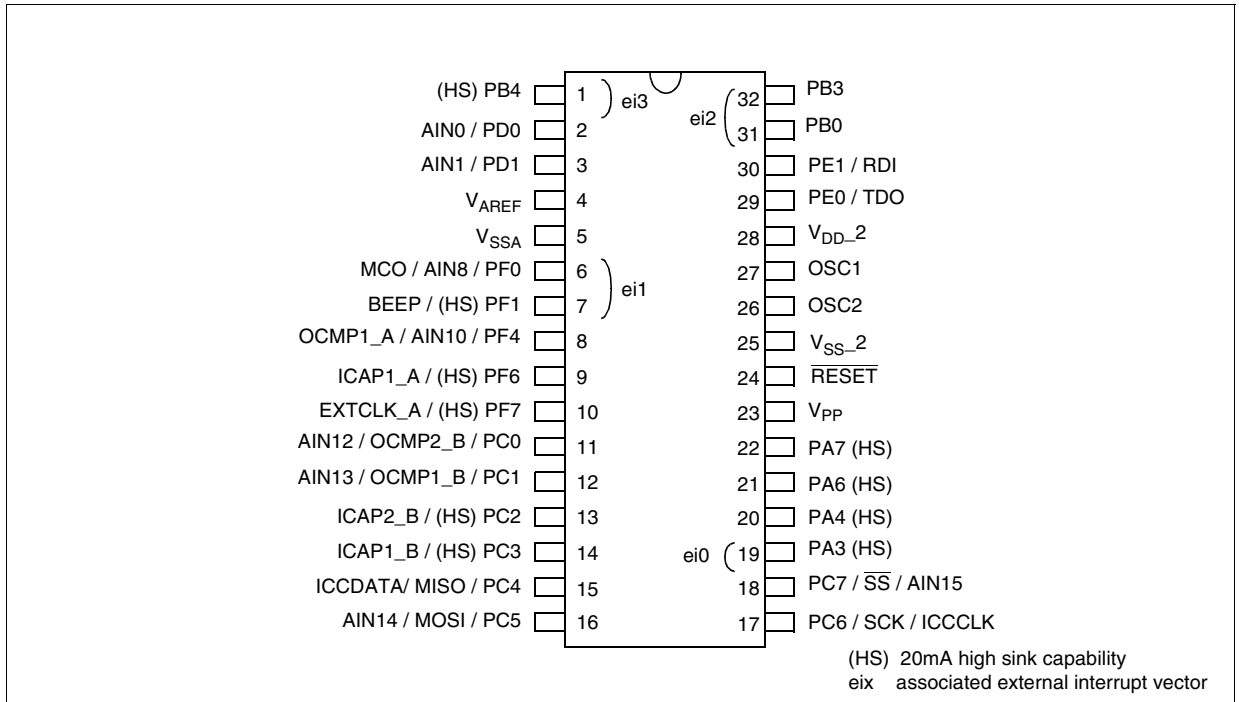
Figure 4. 32-Pin TQFP 7x7 Package Pinout
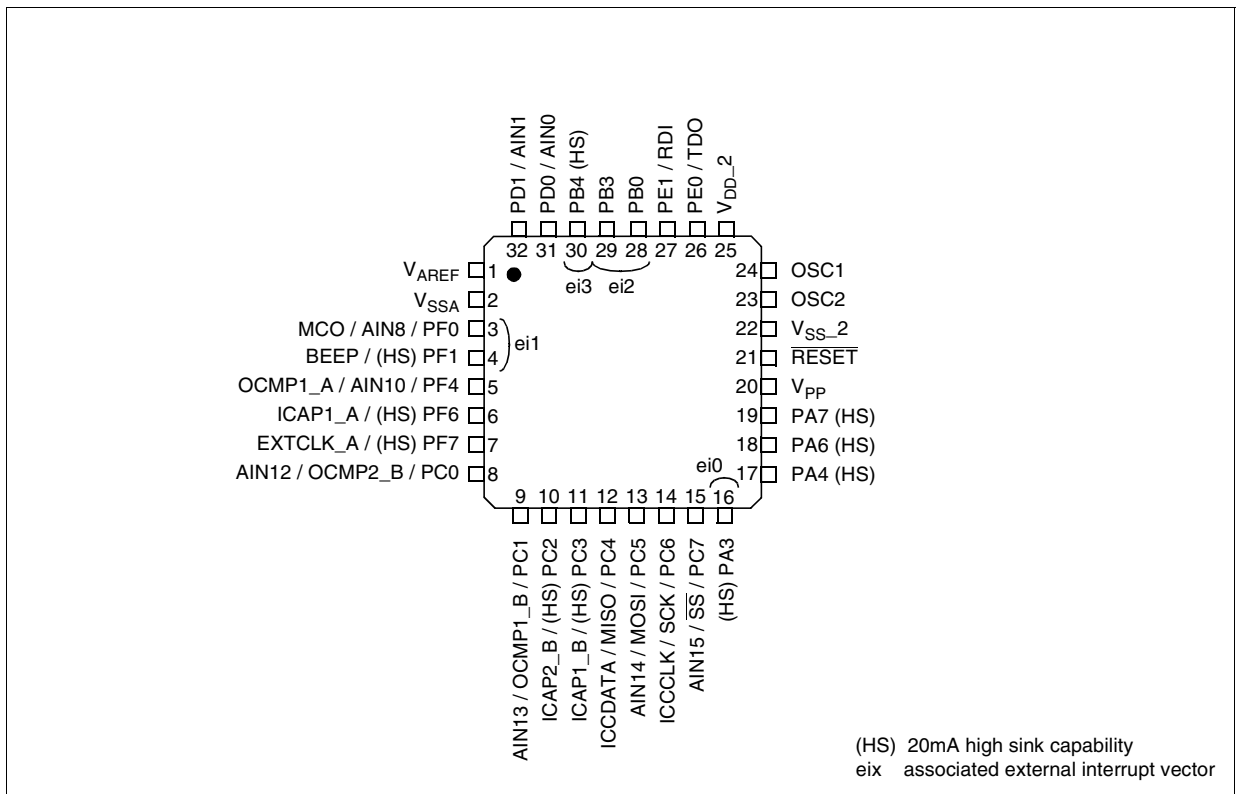
PIN DESCRIPTION (Cont'd)

For external pin connection guidelines, refer to See "ELECTRICAL CHARACTERISTICS" on page 133.

Legend / Abbreviations for Table 1**:**

Type: I = input, O = output, S = supply

Input level: A = Dedicated analog input

In/Output level: C = CMOS $0.3V_{DD}/0.7V_{DD}$
                  C_T= CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

– Input: float = floating, wpu = weak pull-up, int = interrupt [1], ana = analog
– Output: OD = open drain [2], PP = push-pull

Refer to "I/O PORTS" on page 43 for more details on the software configuration of the I/O ports.

The RESET configuration of each pin is shown in bold. This configuration is valid as long as the device is in reset state.

Table 1. Device Pin Description

TQFP44	SDIP42	TQFP32	SDIP32	Pin Name	Type	Input (Level)	Output (Level)	float	wpu	int	ana	OD	PP	Main function (after reset)	Alternate Function	
6	1	30	1	PB4 (HS)/ARTCLK	I/O	C_T	HS	**X**	ei3			X	X	Port B4	PWM-ART External Clock	
7	2	31	2	PD0/AIN0	I/O	C_T		**X**	X		X	X	X	Port D0	ADC Analog Input 0	
8	3	32	3	PD1/AIN1	I/O	C_T		**X**	X		X	X	X	Port D1	ADC Analog Input 1	
9	4			PD2 / AIN2	I/O	C_T		**X**	X		X	X	X	Port D2	ADC Analog Input 2	
10	5			PD3/AIN3	I/O	C_T		**X**	X		X	X	X	Port D3	ADC Analog Input 3	
11	6			PD4/AIN4	I/O	C_T		**X**	X		X	X	X	Port D4	ADC Analog Input 4	
12	7			PD5/AIN5	I/O	C_T		**X**	X		X	X	X	Port D5	ADC Analog Input 5	
13	8	1	4	V_{AREF}	S									Analog Reference Voltage for ADC		
14	9	2	5	V_{SSA}	S									Analog Ground Voltage		
15	10	3	6	PF0/MCO/AIN8	I/O	C_T		**X**	ei1			X	X	Port F0	Main clock out ($f_{OSC}/2$)	ADC Analog Input 8
16	11	4	7	PF1 (HS)/BEEP	I/O	C_T	HS	**X**	ei1			X	X	Port F1	Beep signal output	
17	12			PF2 (HS)	I/O	C_T	HS	**X**		ei1		X	X	Port F2		
18	13	5	8	PF4/OCMP1_A/AIN10	I/O	C_T		**X**	X		X	X	X	Port F4	Timer A Output Compare 1	ADC Analog Input 10
19	14	6	9	PF6 (HS)/ICAP1_A	I/O	C_T	HS	**X**	X			X	X	Port F6	Timer A Input Capture 1	
20	15	7	10	PF7 (HS)/ EXTCLK_A	I/O	C_T	HS	**X**	X			X	X	Port F7	Timer A External Clock Source	
21				V_{DD_0}	S									Digital Main Supply Voltage		
22				V_{SS_0}	S									Digital Ground Voltage		
23	16	8	11	PC0/OCMP2_B/AIN12	I/O	C_T		**X**	X		X	X	X	Port C0	Timer B Output Compare 2	ADC Analog Input 12

TQFP44	SDIP42	TQFP32	SDIP32	Pin Name	Type	Input	Output	float	wpu	int	ana	OD	PP	Main function (after reset)	Alternate Function	
	Pin n°					Level		Port						Main function (after reset)	Alternate Function	
						Input	Output	Input				Output				
24	17	9	12	PC1/OCMP1_B/ AIN13	I/O	C_T		X	X		X	X	X	Port C1	Timer B Output Compare 1	ADC Analog Input 13
25	18	10	13	PC2 (HS)/ICAP2_B	I/O	C_T	HS	X	X			X	X	Port C2	Timer B Input Capture 2	
26	19	11	14	PC3 (HS)/ICAP1_B	I/O	C_T	HS	X	X			X	X	Port C3	Timer B Input Capture 1	
27	20	12	15	PC4/MISO/ICCDA-TA	I/O	C_T		X	X			X	X	Port C4	SPI Master In / Slave Out Data	ICC Data Input
28	21	13	16	PC5/MOSI/AIN14	I/O	C_T		X	X		X	X	X	Port C5	SPI Master Out / Slave In Data	ADC Analog Input 14
29	22	14	17	PC6/SCK/ICCCLK	I/O	C_T		X	X			X	X	Port C6	SPI Serial Clock	ICC Clock Output
30	23	15	18	PC7/$\overline{SS}$/AIN15	I/O	C_T		X	X		X	X	X	Port C7	SPI Slave Select (active low)	ADC Analog Input 15
31	24	16	19	PA3 (HS)	I/O	C_T	HS	X		ei0		X	X	Port A3		
32	25			V_{DD_1}	S									Digital Main Supply Voltage		
33	26			V_{SS_1}	S									Digital Ground Voltage		
34	27	17	20	PA4 (HS)	I/O	C_T	HS	X	X			X	X	Port A4		
35	28			PA5 (HS)	I/O	C_T	HS	X	X			X	X	Port A5		
36	29	18	21	PA6 (HS)/SDAI	I/O	C_T	HS	X				T		Port A6	I^2C Data [2]	
37	30	19	22	PA7 (HS)/SCLI	I/O	C_T	HS	X				T		Port A7	I^2C Clock [2]	
38	31	20	23	V_{PP}	I									Must be tied low. In the programming mode when available, this pin acts as the programming voltage input V_{PP}. See Section 12.9.2 for more details.		
39	32	21	24	$\overline{RESET}$	I/O	C_T								Top priority non maskable interrupt.		
40	33	22	25	V_{SS_2}	S									Digital Ground Voltage		
41	34	23	26	OSC2	O									Resonator oscillator inverter output or capacitor input for RC oscillator		
42	35	24	27	OSC1	I									External clock input or Resonator oscillator inverter input or resistor input for RC oscillator		
43	36	25	28	V_{DD_2}	S									Digital Main Supply Voltage		
44	37	26	29	PE0/TDO	I/O	C_T		X	X			X	X	Port E0	SCI Transmit Data Out	
1	38	27	30	PE1/RDI	I/O	C_T		X	X			X	X	Port E1	SCI Receive Data In	
2	39	28	31	PB0/PWM3	I/O	C_T		X		ei2		X	X	Port B0	PWM Output 3	
3	40			PB1/PWM2	I/O	C_T		X		ei2		X	X	Port B1	PWM Output 2	
4	41			PB2/PWM1	I/O	C_T		X		ei2		X	X	Port B2	PWM Output 1	
5	42	29	32	PB3/PWM0	I/O	C_T		X			ei2	X	X	Port B3	PWM Output 0	

Notes:

1. In the interrupt input column, "eiX" defines the associated external interrupt vector. If the weak pull-up

column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, else the configuration is floating interrupt input.

2. In the open drain output column, "T" defines a true open drain I/O (P-Buffer and protection diode to V_{DD} are not implemented). See See "I/O PORTS" on page 43. and Section 12.8 I/O PORT PIN CHARACTER-ISTICS for more details.

3. OSC1 and OSC2 pins connect a crystal/ceramic resonator, an RC oscillator, or an external source to the on-chip oscillator; see Section 1 INTRODUCTION and Section 12.5 CLOCK AND TIMING CHARAC-TERISTICS for more details.

3 REGISTER & MEMORY MAP

As shown in Figure 5, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 1024 bytes of RAM and up to 32 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

Figure 5. Memory Map
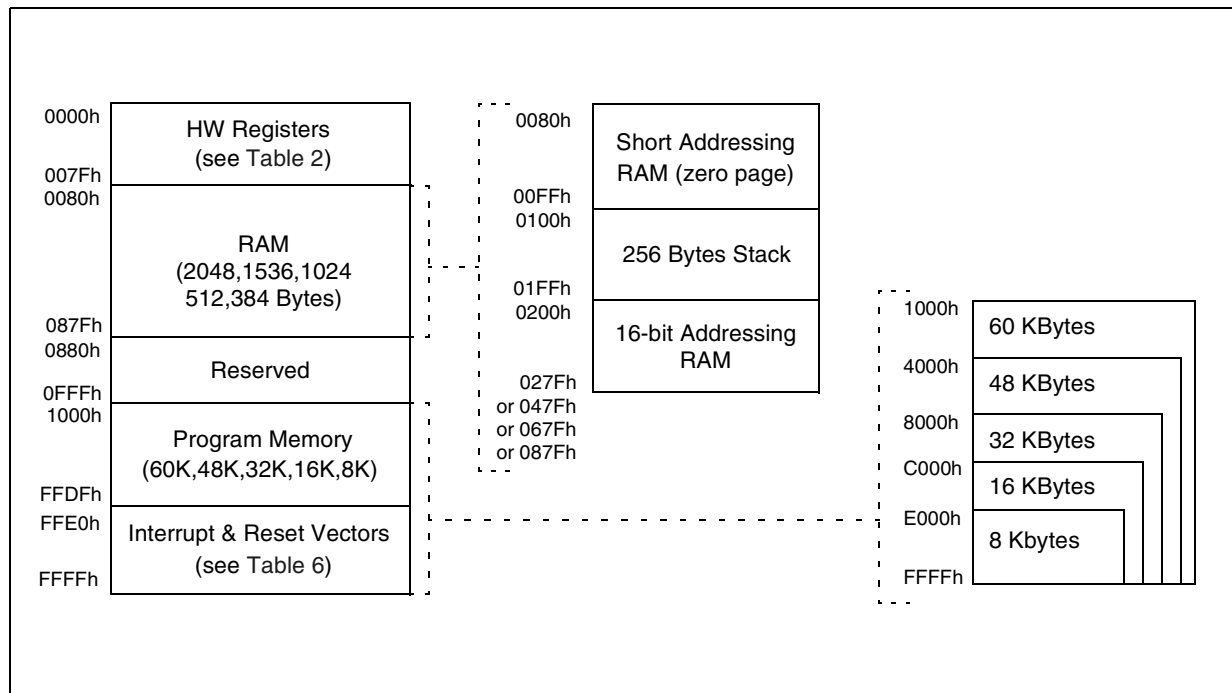
Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h	Port A [2]	PADR	Port A Data Register	00h[1]	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h		PAOR	Port A Option Register	00h	R/W
0003h	Port B [2]	PBDR	Port B Data Register	00h[1]	R/W
0004h		PBDDR	Port B Data Direction Register	00h	R/W
0005h		PBOR	Port B Option Register	00h	R/W
0006h	Port C	PCDR	Port C Data Register	00h[1]	R/W
0007h		PCDDR	Port C Data Direction Register	00h	R/W
0008h		PCOR	Port C Option Register	00h	R/W
0009h	Port D [2]	PDADR	Port D Data Register	00h[1]	R/W
000Ah		PDDDR	Port D Data Direction Register	00h	R/W
000Bh		PDOR	Port D Option Register	00h	R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
000Ch 000Dh 000Eh	Port E [2]	PEDR PEDDR PEOR	Port E Data Register Port E Data Direction Register Port E Option Register	00h[1] 00h 00h	R/W R/W[2] R/W[2]
000Fh 0010h 0011h	Port F [2]	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h[1] 00h 00h	R/W R/W R/W
0012h to 0017h			Reserved Area (6 Bytes)		
0018h 0019h 001Ah 001Bh 001Ch 001Dh 001Eh	I^{2}C	I2CCR I2CSR1 I2CSR2 I2CCCR I2COAR1 I2COAR2 I2CDR	I^{2}C Control Register I^{2}C Status Register 1 I^{2}C Status Register 2 I^{2}C Clock Control Register I^{2}C Own Address Register 1 I^{2}C Own Address Register2 I^{2}C Data Register	00h 00h 00h 00h 00h 00h 00h	R/W Read Only Read Only R/W R/W R/W R/W
001Fh 0020h			Reserved Area (2 Bytes)		
0021h 0022h 0023h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control/Status Register	xxh 0xh 00h	R/W R/W R/W
0024h 0025h 0026h 0027h	ITC	ITSPR0 ITSPR1 ITSPR2 ITSPR3	Interrupt Software Priority Register 0 Interrupt Software Priority Register 1 Interrupt Software Priority Register 2 Interrupt Software Priority Register 3	FFh FFh FFh FFh	R/W R/W R/W R/W
0028h		EICR	External Interrupt Control Register	00h	R/W
0029h	FLASH	FCSR	Flash Control/Status Register	00h	R/W
002Ah	WATCHDOG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh		SICSR	System Integrity Control/Status Register	000x 000x b	R/W
002Ch 002Dh	MCC	MCCSR MCCBCR	Main Clock Control / Status Register Main Clock Controller: Beep Control Register	00h 00h	R/W R/W
002Eh to 0030h			Reserved Area (3 Bytes)		

Address	Block	Register Label	Register Name	Reset Status	Remarks
0031h		TACR2	Timer A Control Register 2	00h	R/W
0032h		TACR1	Timer A Control Register 1	00h	R/W
0033h		TACSR	Timer A Control/Status Register	xxh	R/W
0034h		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
0035h		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
0036h		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
0037h		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
0038h	TIMER A	TACHR	Timer A Counter High Register	FFh	Read Only
0039h		TACLR	Timer A Counter Low Register	FCh	Read Only
003Ah		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Timer A Alternate Counter Low Register	FCh	Read Only
003Ch		TAIC2HR	Timer A Input Capture 2 High Register	xxh	Read Only
003Dh		TAIC2LR	Timer A Input Capture 2 Low Register	xxh	Read Only
003Eh		TAOC2HR	Timer A Output Compare 2 High Register	80h	R/W
003Fh		TAOC2LR	Timer A Output Compare 2 Low Register	00h	R/W
0040h			Reserved Area (1 Byte)		
0041h		TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBCSR	Timer B Control/Status Register	xxh	R/W
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h	TIMER B	TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 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	00xx xxxx b	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 area	---	
0057h		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W
0058h to 006Fh			Reserved Area (24 Bytes)		
0070h		ADCCSR	Control/Status Register	00h	R/W
0071h	ADC	ADCDRH	Data High Register	xxh	Read Only
0072h		ADCDRL	Data Low Register	0000 00xx b	Read Only

Address	Block	Register Label	Register Name	Reset Status	Remarks
0073h		PWMDCR3	PWM AR Timer Duty Cycle Register 3	00h	R/W
0074h		PWMDCR2	PWM AR Timer Duty Cycle Register 2	00h	R/W
0075h		PWMDCR1	PWM AR Timer Duty Cycle Register 1	00h	R/W
0076h		PWMDCR0	PWM AR Timer Duty Cycle Register 0	00h	R/W
0077h		PWMCR	PWM AR Timer Control Register	00h	R/W
0078h	PWM ART	ARTCSR	Auto-Reload Timer Control/Status Register	00h	R/W
0079h		ARTCAR	Auto-Reload Timer Counter Access Register	00h	R/W
007Ah		ARTARR	Auto-Reload Timer Auto-Reload Register	00h	R/W
007Bh		ARTICCSR	AR Timer Input Capture Control/Status Reg.	00h	R/W
007Ch		ARTICR1	AR Timer Input Capture Register 1	00h	Read Only
007Dh		ARTICR2	AR Timer Input Capture Register 1	00h	Read Only
007Eh 007Fh	Reserved Area (2 Bytes)				

Legend: x=undefined, R/W=read/write

Notes:

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.

2. The bits associated with unavailable pins must always keep their reset value.

4 FLASH PROGRAM MEMORY

4.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a Byte-by-Byte basis using an external V_{PP} supply.

The HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (In-Circuit Programming) or IAP (In-Application Programming).

The array matrix organisation allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main Features

- Three Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (In-Circuit Programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (In-Application Programming) In this mode, all sectors except Sector 0, can be programmed or erased without removing the device from the application board and while the application is running.
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Read-out protection against piracy
- Register Access Security System (RASS) to prevent accidental programming or erasing

4.3 Structure

The Flash memory is organised in sectors and can be used for both code and data storage.

Depending on the overall Flash memory size in the microcontroller device, there are up to three user sectors (see Table 3). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

The first two sectors have a fixed size of 4 Kbytes (see Figure 6). They are mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

Table 3. Sectors available in Flash devices

Flash Memory Size (bytes)	Available Sectors
4K	Sector 0
8K	Sectors 0,1
> 8K	Sectors 0,1, 2

4.4 Program Memory Read-out Protection

The read-out protection is enabled through an option bit.

When this option is selected, the programs and data stored in the program memory (Flash or ROM) are protected against read-out piracy (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire program memory is first automatically erased.

Refer to the Option Byte description for more details.

Figure 6. Memory Map and Sector Address
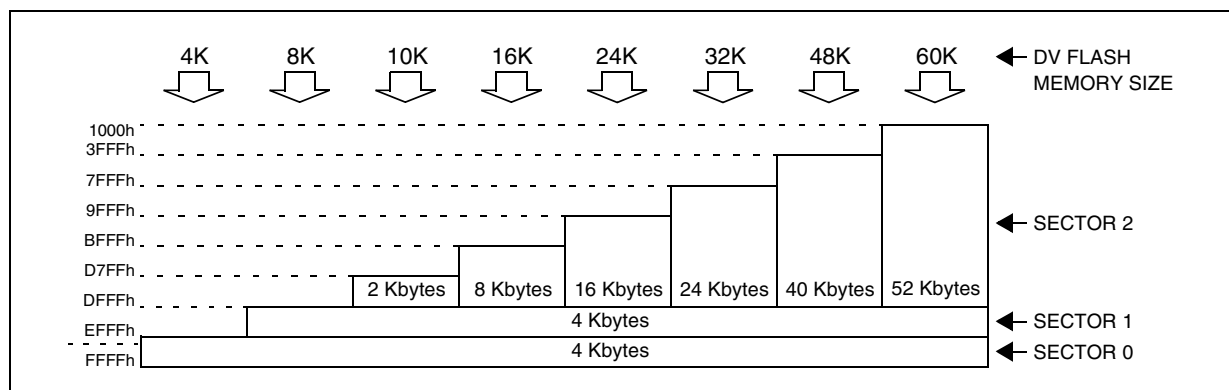
FLASH PROGRAM MEMORY (Cont'd)

4.5 ICP (In-Circuit Programming)

To perform ICP the microcontroller must be switched to ICC (In-Circuit Communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see Figure 7). For more details on the pin locations, refer to the device pinout description.

ICP needs five signals to be connected to the programming tool. These signals are:

– $\overline{\text{RESET}}$: device reset
– V_{SS}: device power supply ground
– ICCCLK: ICC output serial clock pin
– ICCDATA: ICC input serial data pin
– V_{PP}: programming voltage

When the device is not yet configured to support the application clock source (option byte not yet programmed) or if the option bytes have to be programmed using ICP, one more pin has to be connected:

– OSCIN: main clock input for external source

When the device is not supplied (V_{DD}) by the application, one more pin has to be connected:

– V_{DD}: main power supply

CAUTIONS:

1. If $\overline{\text{RESET}}$, ICCCLK or ICCDATA pins are used for other purposes in the application, a serial resistor has to be implemented to avoid a conflict in case one of the other devices forces the signal level.

2. As soon as the external controller is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application.

Note: To develop a custom programming tool, refer to the ST7 Flash Programming and ICC Reference Manual which gives full details on the ICC protocol hardware and software.

4.6 IAP (In-Application Programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from the SPI, SCI, USB or CAN interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

Figure 7. Typical ICP Interface
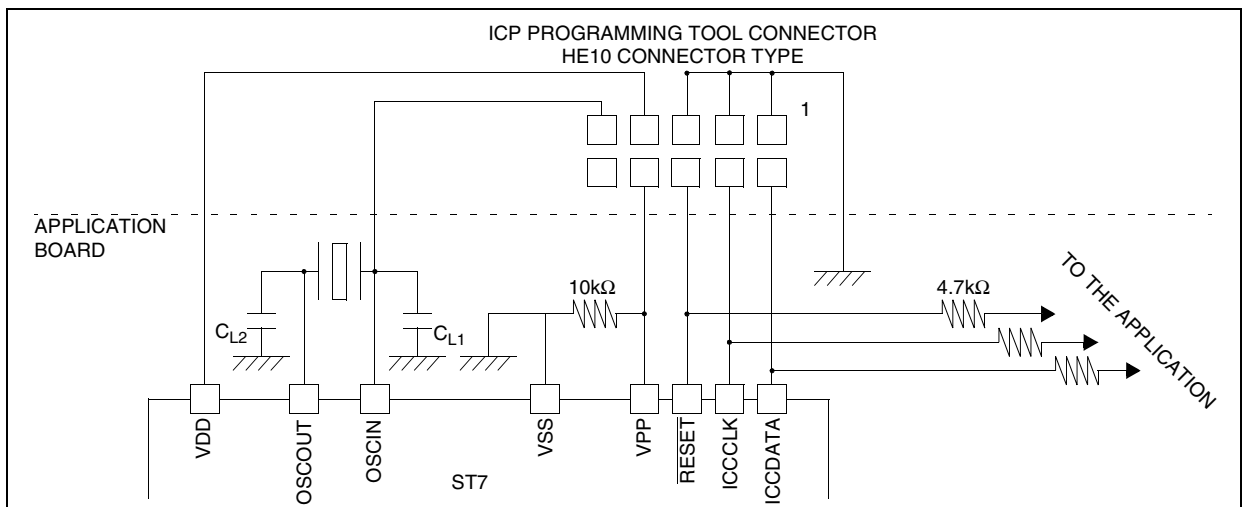
FLASH PROGRAM MEMORY (Cont'd)

4.6.1 Register Description

FLASH CONTROL/STATUS REGISTER (FCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	0	0

This register is reserved for use by Programming Tool software. It controls the Flash programming and erasing operations. For details on customizing Flash programming methods and In-Circuit Testing, refer to the ST7 Flash Programming and ICC Reference Manual.

5 CENTRAL PROCESSING UNIT

5.1 INTRODUCTION

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

5.2 MAIN FEATURES

- Enable executing 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
- Low power HALT and WAIT modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

5.3 CPU REGISTERS

The 6 CPU registers shown in Figure 8 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)

These 8-bit registers are used to create effective addresses or as 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.

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 8. CPU Registers
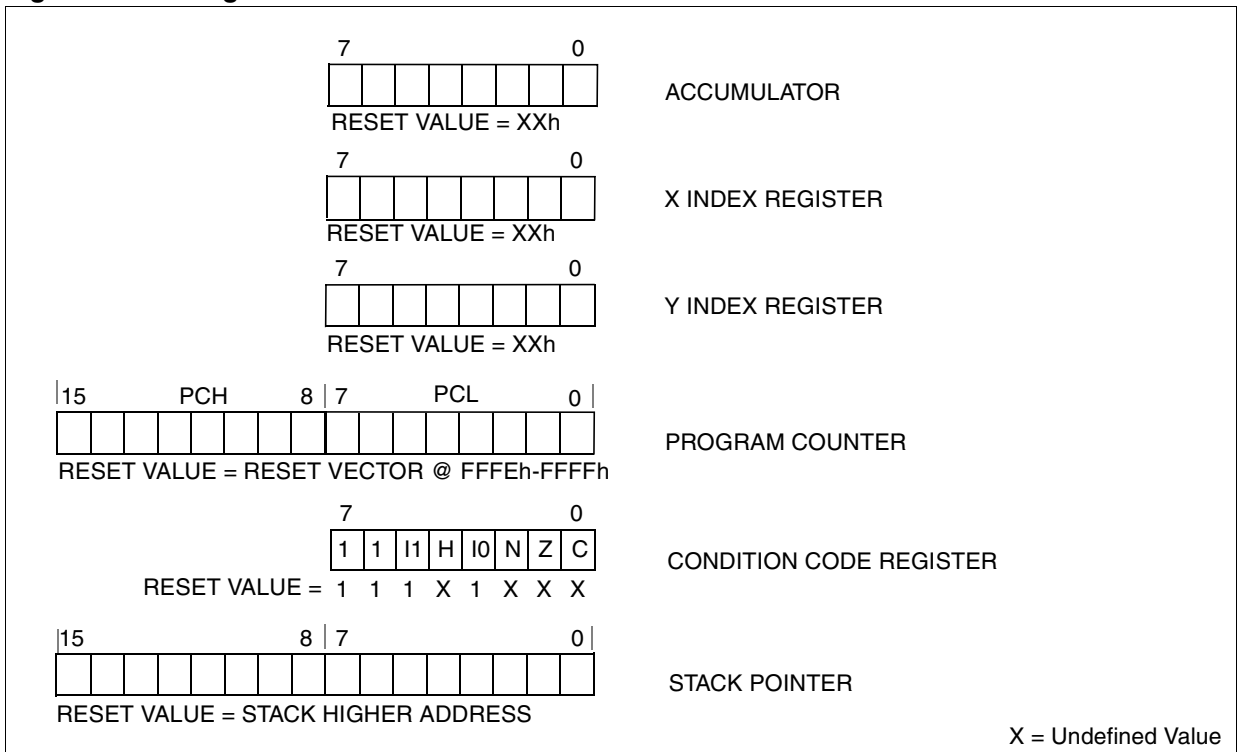
CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	I1	H	I0	N	Z	C

The 8-bit Condition Code register contains the interrupt masks 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.

Arithmetic Management Bits

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 instructions. 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 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's a copy of the result 7th bit.

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.

Interrupt Management Bits

Bit 5,3 = **I1, I0** *Interrupt*

The combination of the I1 and I0 bits gives the current interrupt software priority.

Interrupt Software Priority	I1	I0
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh

15							8
0	0	0	0	0	0	0	1

7							0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
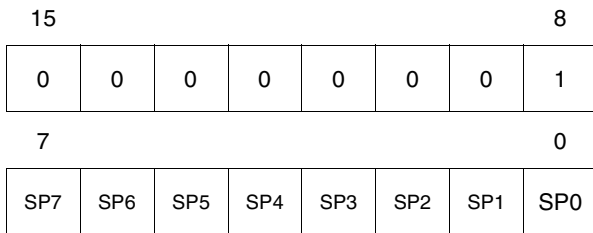
The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 9).

Since the stack is 256 bytes deep, the 8 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 9

– 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 9. Stack Manipulation Example

Stack Higher Address = 01FFh
Stack Lower Address = 0100h
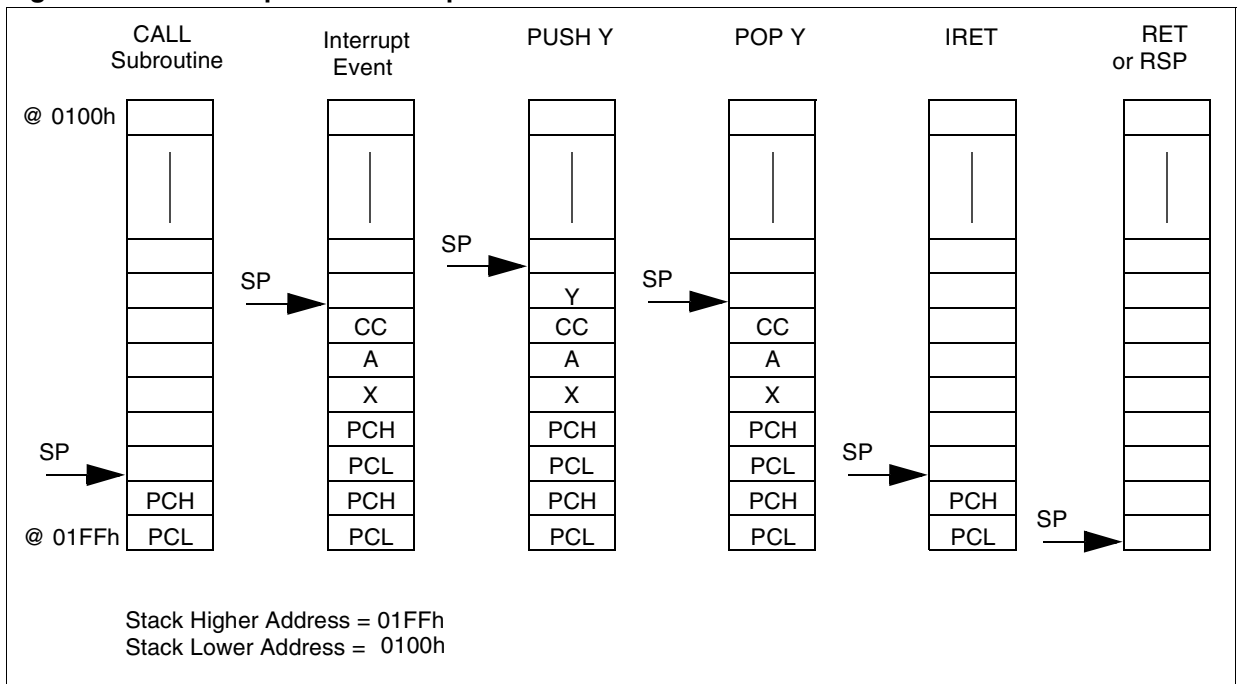
6 SUPPLY, RESET AND CLOCK MANAGEMENT

The device includes a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 11.

For more details, refer to dedicated parametric section.

Main features

- Optional PLL for multiplying the frequency by 2 (enabled by option byte)
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
  - 4 Crystal/Ceramic resonator oscillators
  - 1 External RC oscillator
  - 1 Internal RC oscillator
- System Integrity Management (SI)
  - Main supply Low voltage detection (LVD)
  - Auxiliary Voltage detector (AVD) with interrupt capability for monitoring the main supplyClock

Security System (CSS) with Clock Filter and Backup Safe Oscillator (enabled by option byte)

6.1 PHASE LOCKED LOOP

If the clock frequency input to the PLL is in the range 2 to 4 MHz, the PLL can be used to multiply the frequency by two to obtain an f_{OSC2} of 4 to 8 MHz. The PLL is enabled by option byte. If the PLL is disabled, then $f_{OSC2} = f_{OSC}/2$.

Figure 10. PLL Block Diagram
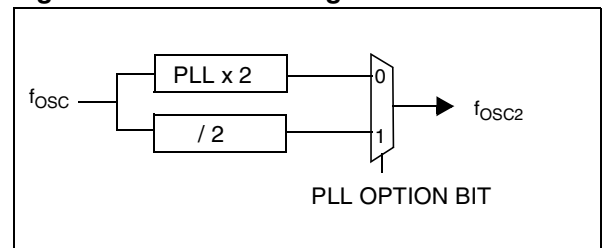
Figure 11. Clock, Reset and Supply Block Diagram
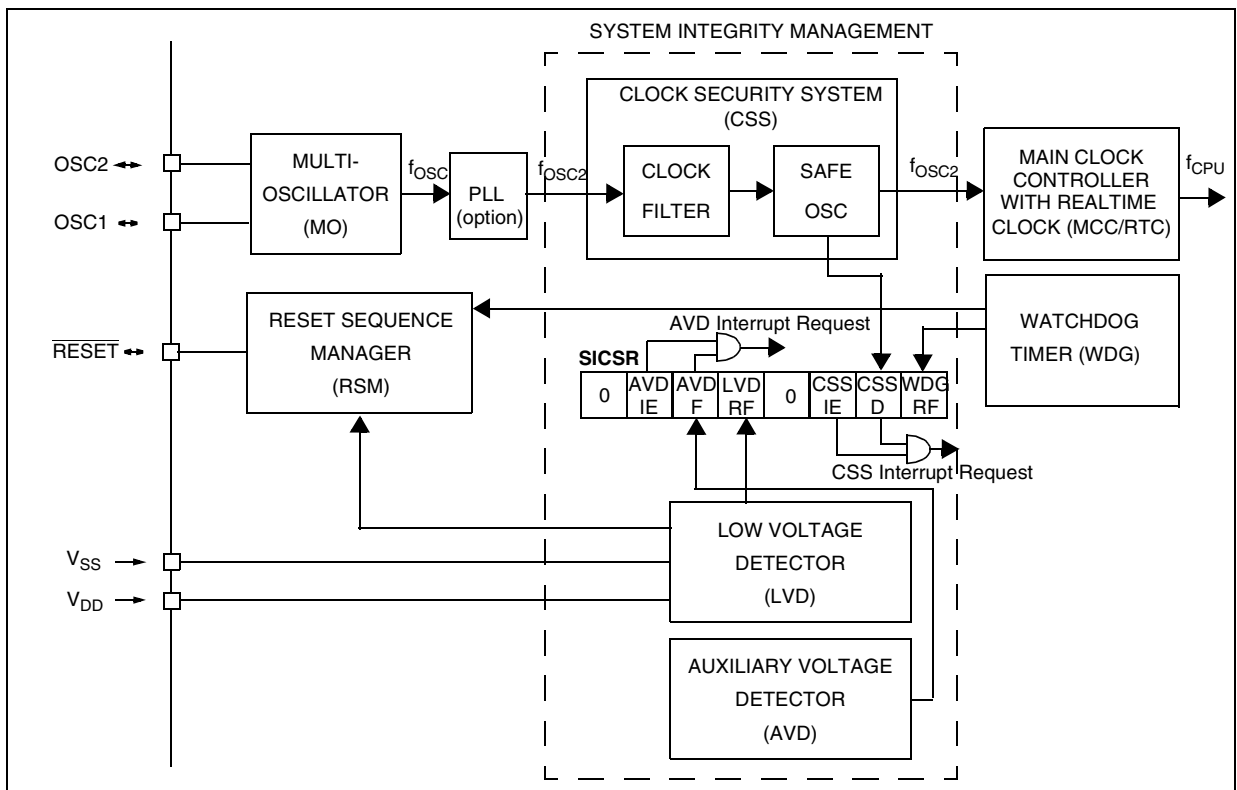
6.2 MULTI-OSCILLATOR (MO)

The main clock of the ST7 can be generated by four different source types coming from the multi-oscillator block:

■ an external source

■ 5 crystal or ceramic resonator oscillators

■ an external RC oscillator

■ an internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configuration are shown in Table 4. Refer to the electrical characteristics section for more details.

External Clock Source

In this external clock mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to ground.

Crystal/Ceramic Oscillators

This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. The selection within a list of 5 oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to Section 14.1 on page 169 for more details on the frequency ranges). In this mode of the multi-oscillator, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators are not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

External RC Oscillator

This oscillator allows a low cost solution for the main clock of the ST7 using only an external resistor and an external capacitor. The frequency of the external RC oscillator (in the range of some MHz.) is fixed by the resistor and the capacitor values. Consequently in this MO mode, the accuracy of the clock is directly linked to the accuracy of the discrete components.

Internal RC Oscillator

The internal RC oscillator mode is based on the same principle as the external RC oscillator including the resistance and the capacitance of the de-

vice. This mode is the most cost effective one with the drawback of a lower frequency accuracy. Its frequency is in the range of several MHz.

In this mode, the two oscillator pins have to be tied to ground.

Table 4. ST7 Clock Sources

6.3 RESET SEQUENCE MANAGER (RSM)

6.3.1 Introduction

The reset sequence manager includes three RE-SET sources as shown in Figure 13:

■ External $\overline{RESET}$ source pulse

■ Internal LVD RESET (Low Voltage Detection)

■ Internal WATCHDOG RESET

These sources act on the $\overline{RESET}$ pin and it is always kept low during the delay phase.

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

The basic RESET sequence consists of 3 phases as shown in Figure 12:

■ Delay depending on the RESET source

■ 256 or 4096 CPU clock cycle delay (selected by option byte)

■ RESET vector fetch

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay should be selected by option byte to correspond to the stabilization time of the external oscillator used in the application (see Section 14.1 on page 169).

The RESET vector fetch phase duration is 2 clock cycles.

Figure 12. RESET Sequence Phases

RESET		
DELAY	INTERNAL RESET 256 or 4096 CLOCK CYCLES	FETCH VECTOR
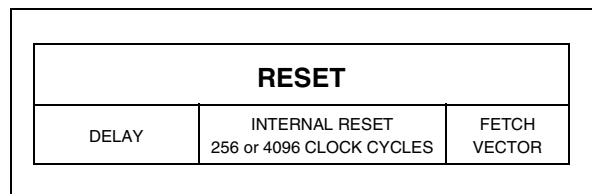
Figure 13. Reset Block Diagram
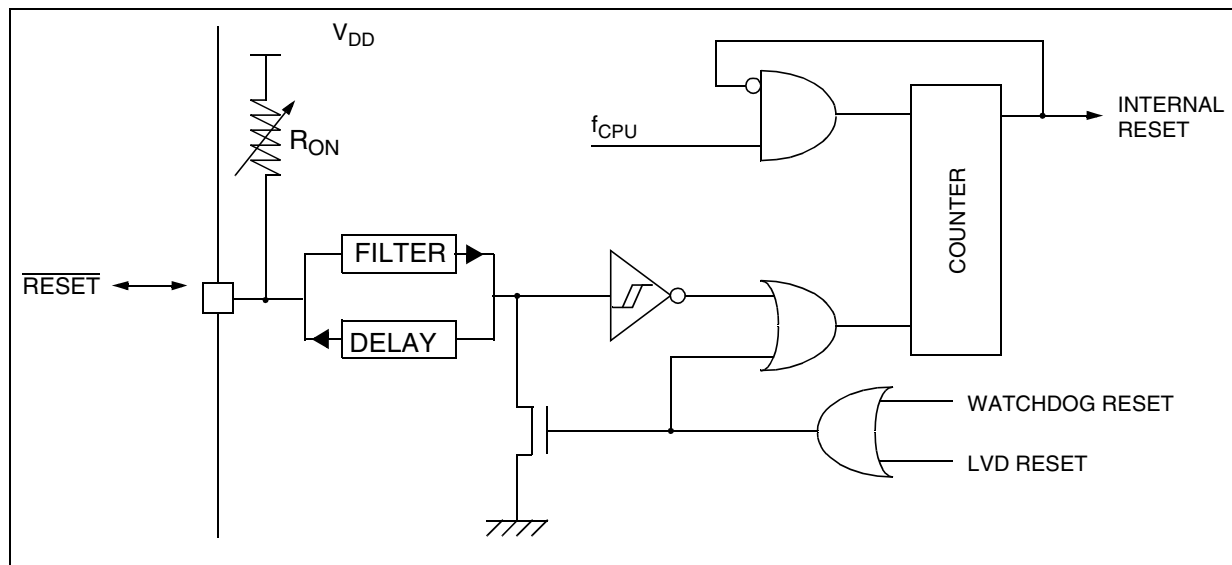
RESET SEQUENCE MANAGER (Cont'd)

6.3.2 Asynchronous External RESET pin

The RESET pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device. See electrical characteristics section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized. This detection is asynchronous and therefore the MCU can enter reset state even in HALT mode.

The RESET pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

Two RESET sequences can be associated with this RESET source: short or long external reset pulse (see Figure 14).

Starting from the external RESET pulse recognition, the device RESET pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

6.3.3 Internal Low Voltage Detection RESET

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:

■ Power-On RESET

■ Voltage Drop RESET

The device RESET pin acts as an output that is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge) as shown in Figure 14.

The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

6.3.4 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 14.

Starting from the Watchdog counter underflow, the device RESET pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 14. RESET Sequences
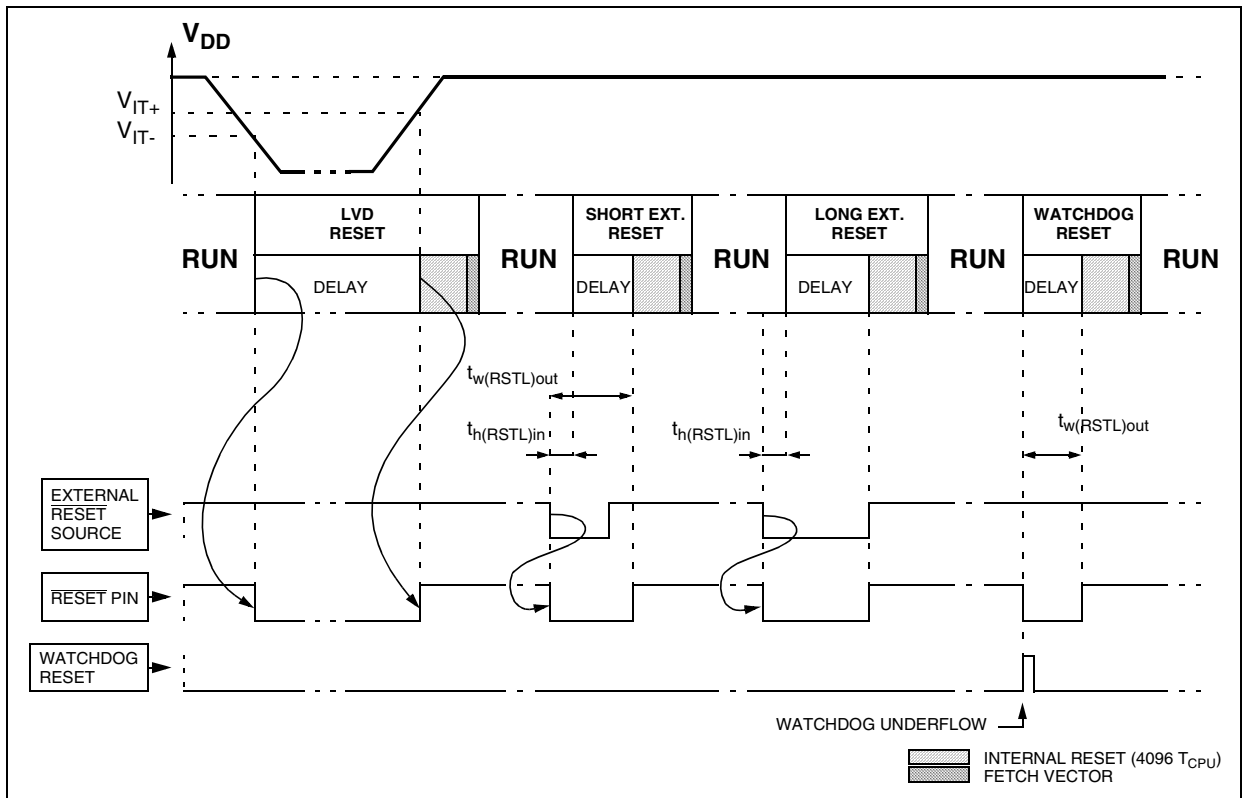
6.4 SYSTEM INTEGRITY MANAGEMENT (SI)

The System Integrity Management block contains the Low Voltage Detector (LVD), Auxiliary Voltage Detector (AVD) and Clock Security System (CSS) functions. It is managed by the SICSR register.

6.4.1 Low Voltage Detector (LVD)

The Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} 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_{IT+} when V_{DD} is rising
  – V_{IT-} when V_{DD} is falling

The LVD function is illustrated in Figure 15.

The voltage threshold can be configured by option byte to be low, medium or high.

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

  – under full software control
  – in static safe reset

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

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

Notes:

The LVD allows the device to be used without any external RESET circuitry.

The LVD is an optional function which can be selected by option byte.

Figure 15. Low Voltage Detector vs Reset
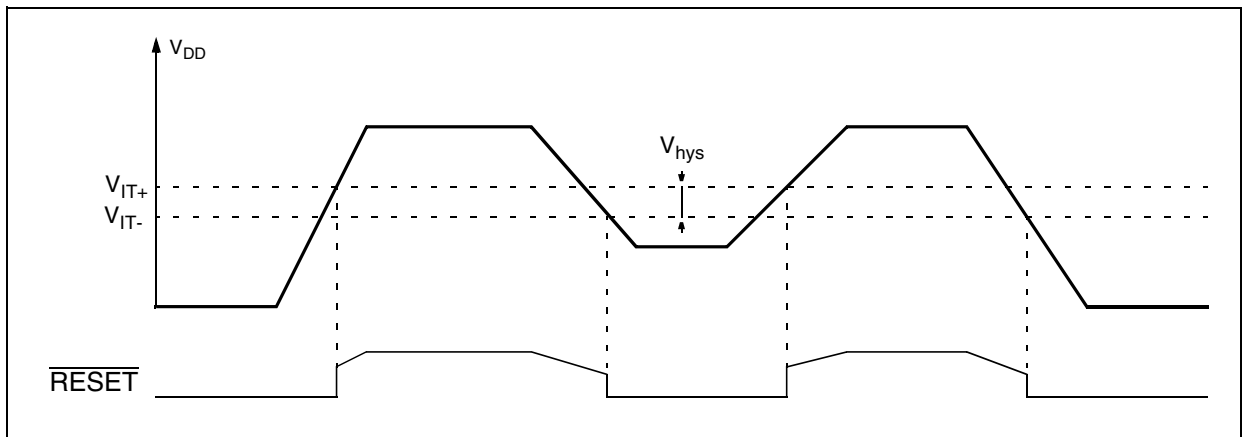
SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.2 Auxiliary Voltage Detector (AVD)

The Voltage Detector function (AVD) is based on an analog comparison between a V_{IT-} and V_{IT+} reference value and the V_{DD} main supply. The V_{IT-} reference value for falling voltage is lower than the V_{IT+} reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (VDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the V_{DD} Main Supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see Section 14.1 on page 169).

If the VD interrupt is enabled, an interrupt is generated when the voltage crosses the V_{IT+} or V_{IT-} threshold (VDF bit toggles).

In the case of a drop in voltage, the VD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 16.

The interrupt on the rising edge is used to inform the application that the V_{DD} warning state is over

Figure 16. Using the AVD to Monitor V_{DD} (VDS bit=0)
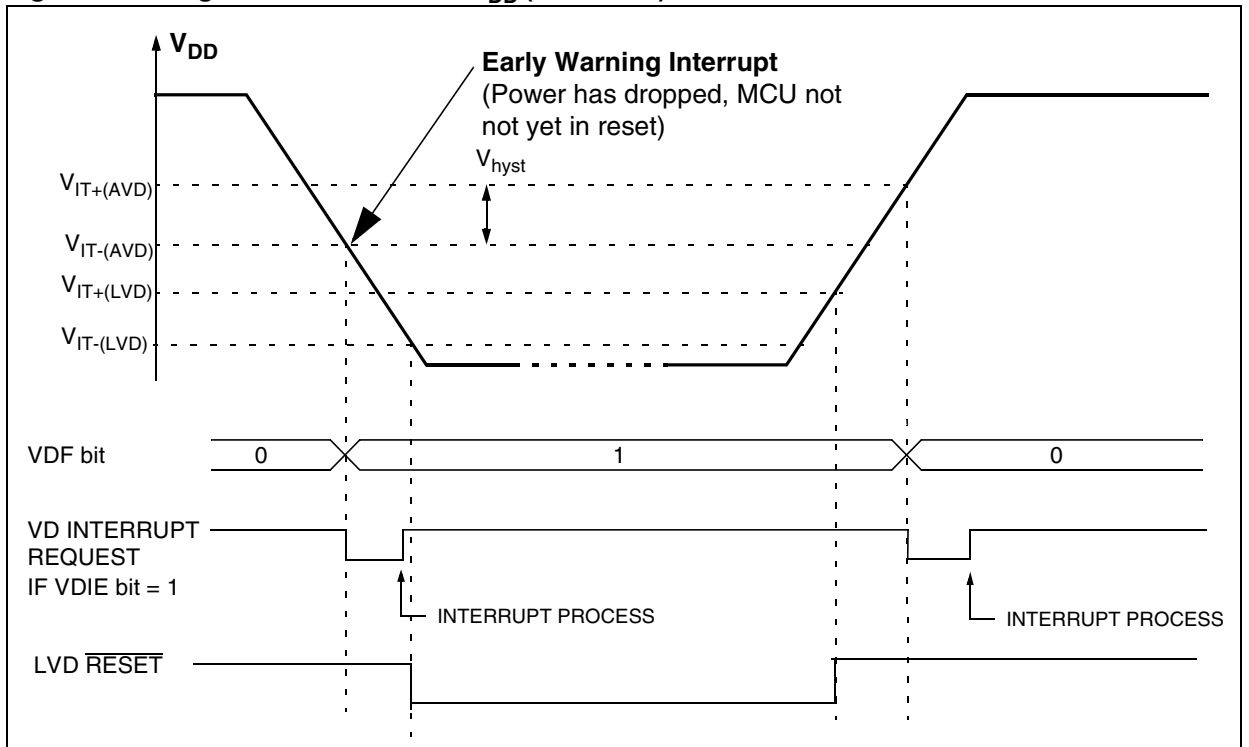
SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.3 Clock Security System (CSS)

The Clock Security System (CSS) protects the ST7 against main clock problems. To allow the integration of the security features in the applications, it is based on a clock filter control and an Internal safe oscillator. The CSS can be enabled or disabled by option byte.

6.4.3.1 Clock Filter Control

The clock filter is based on a clock frequency limitation function.

This filter function is able to detect and filter high frequency spikes on the ST7 main clock.

If the oscillator is not working properly (e.g. working at a harmonic frequency of the resonator), the current active oscillator clock can be totally filtered, and then no clock signal is available for the ST7 from this oscillator anymore. If the original clock source recovers, the filtering is stopped automatically and the oscillator supplies the ST7 clock.

6.4.3.2 Safe Oscillator Control

The safe oscillator of the CSS block is a low frequency back-up clock source (see Figure 17).

If the clock signal disappears (due to a broken or disconnected resonator...) during a safe oscillator period, the safe oscillator delivers a low frequency clock signal which allows the ST7 to perform some rescue operations.

Automatically, the ST7 clock source switches back from the safe oscillator if the original clock source recovers.

Limitation detection

The automatic safe oscillator selection is notified by hardware setting the CSSD bit of the SICSR register. An interrupt can be generated if the CS-SIE bit has been previously set.

These two bits are described in the SICSR register description.

6.4.4 Low Power Modes

Mode	Description
WAIT	No effect on SI. CSS and AVD interrupts cause the device to exit from Wait mode.
HALT	The CRSR register is frozen. The CSS (including the safe oscillator) is disabled until HALT mode is exited. The previous CSS configuration resumes 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. The AVD remains active, and an AVD interrupt can be used to exit from Halt mode.

6.4.4.1 Interrupts

The CSS or AVD interrupt events generate an interrupt if the corresponding Enable Control Bit (CSSIE or AVDIE) is set and the interrupt mask in the CC register is reset (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
CSS event detection (safe oscillator activated as main clock)	CSSD	CSSIE	Yes	No [1]
AVD event	AVDF	AVDIE	Yes	Yes

Note 1: This interrupt allows to exit from active-halt mode.

Figure 17. Clock Filter Function and Safe Oscillator Function
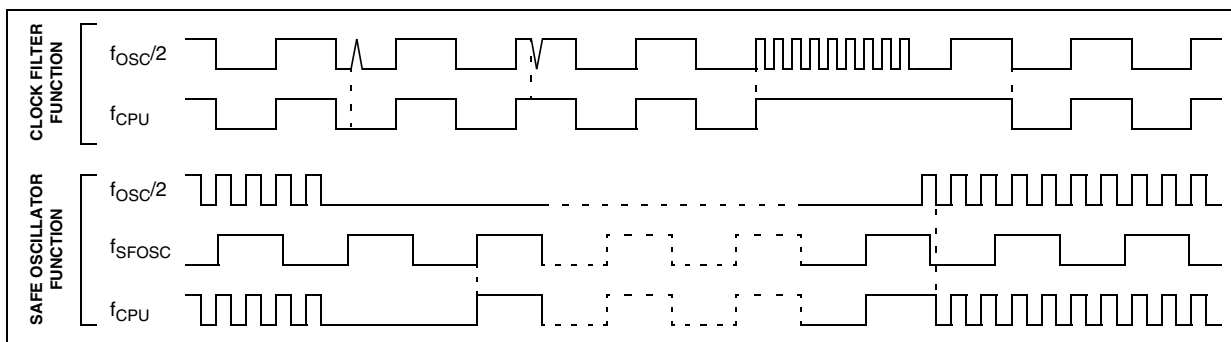
SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.5 Register Description

SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)

Read/Write

Reset Value: 000x 000x (00h)

7							0
0	AVD IE	AVD F	LVD RF	0	CSS IE	CSS D	WDG RF

Bit 7 = Reserved, must be kept cleared.

Bit 6 = **AVDIE** *Voltage Detector interrupt enable*
This bit is set and cleared by software. It enables an interrupt to be generated when the VDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the VD interrupt routine.
0: VD interrupt disabled
1: VD interrupt enabled

Bit 5 = **AVDF** *Voltage Detector flag*
This read-only bit is set and cleared by hardware. If the VDIE bit is set, an interrupt request is generated when the VDF bit changes value.
0: V_{DD} or V_{EVD} over V_{IT+} threshold
1: V_{DD} or V_{EVD} under V_{IT-} threshold

Bit 4 = **LVDRF** *LVD reset flag*
This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero). See WDGRF flag description for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **CSSIE** *Clock security syst interrupt enable*
This bit enables the interrupt when a disturbance is detected by the Clock Security System (CSSD bit set). It is set and cleared by software.
0: Clock security system interrupt disabled

1: Clock security system interrupt enabled
When the CSS is disabled by OPTION BYTE, the CSSIE bit has no effect.

Bit 1 = **CSSD** *Clock security system detection*
This bit indicates that the safe oscillator of the Clock Security System block has been selected by hardware due to a disturbance on the main clock signal (f_{OSC}). It is set by hardware and cleared by reading the SICSR register when the original oscillator recovers.
0: Safe oscillator is not active
1: Safe oscillator has been activated
When the CSS is disabled by OPTION BYTE, the CSSD bit value is forced to 0.

Bit 0 = **WDGRF** *Watchdog reset flag*
This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts).
Combined with the LVDRF flag information, the flag description is given by the following table.

RESET Sources	LVDRF	WDGRF
External $\overline{\text{RESET}}$ pin	0	0
Watchdog	0	1
LVD	1	X

Application notes

The LVDRF flag is not cleared when another RESET type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.
In this case, a watchdog reset can be detected by software while an external reset can not.

CAUTION: When the LVD is not activated with the associated option byte, the WDGRF flag can not be used in the application.

7 INTERRUPTS

7.1 INTRODUCTION

The ST7 enhanced interrupt management provides the following features:

■ Hardware interrupts

■ Software interrupt (TRAP)

■ Nested or concurrent interrupt management with flexible interrupt priority and level management:

– Up to 4 software programmable nesting levels

– Up to 16 interrupt vectors fixed by hardware

– 3 non maskable events: TLI, RESET, TRAP

This interrupt management is based on:

– Bit 5 and bit 3 of the CPU CC register (I1:0),

– Interrupt software priority registers (ISPRx),

– Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order.

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

7.2 MASKING AND PROCESSING FLOW

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see Table 5). The processing flow is shown in Figure 18

When an interrupt request 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.

– I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.

– 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 "Interrupt Mapping" table for vector addresses).

The interrupt service routine should end 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 I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

Table 5. Interrupt Software Priority Levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1		0	1
Level 2		0	0
Level 3 (= interrupt disable)	High	1	1

Figure 18. Interrupt Processing Flowchart
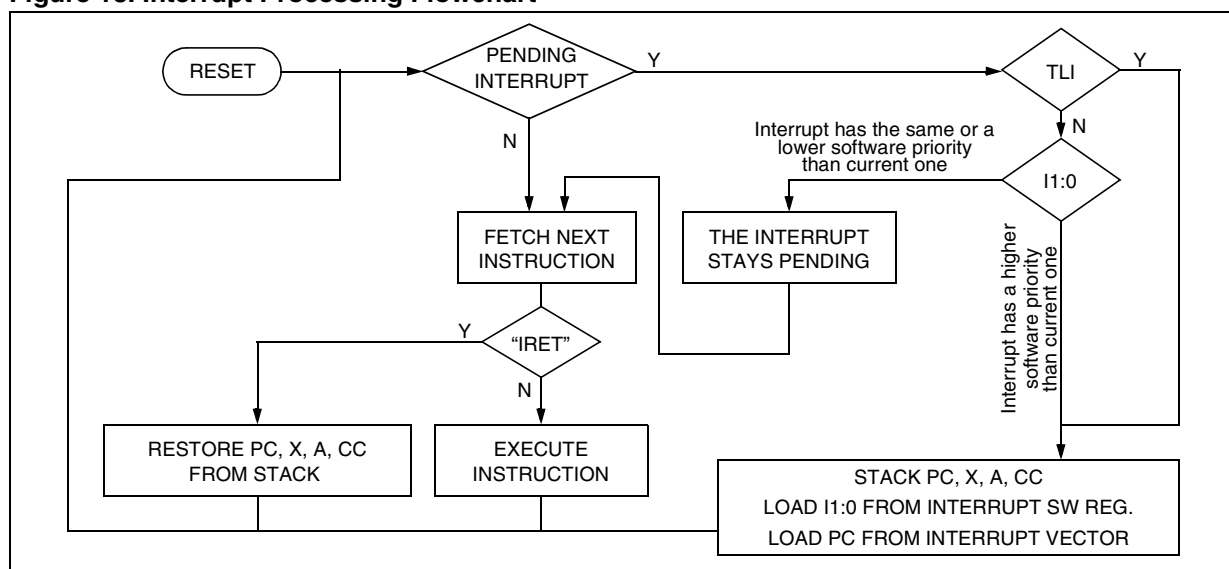
INTERRUPTS (Cont'd)

Servicing Pending Interrupts

As several interrupts can be pending at the same time, the interrupt to be taken into account is determined by the following two-step process:

– the highest software priority interrupt is serviced,

– if several interrupts have the same software priority then the interrupt with the highest hardware priority is serviced first.

Figure 19 describes this decision process.

Figure 19. Priority Decision Process
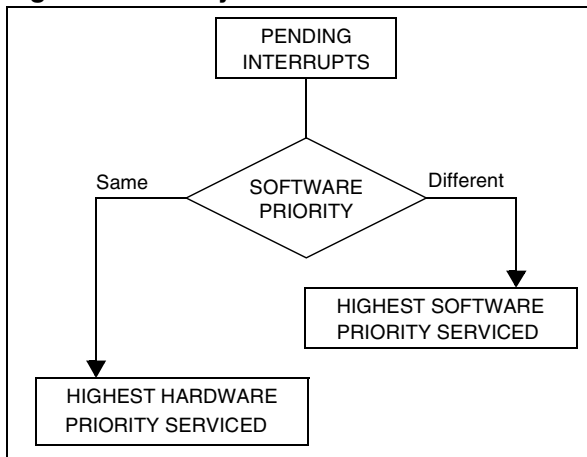
When an interrupt request is not serviced immediately, it is latched and then processed when its software priority combined with the hardware priority becomes the highest one.

Note 1: The hardware priority is exclusive while the software one is not. This allows the previous process to succeed with only one interrupt.
Note 2: RESET, TRAP and TLI are non maskable and they can be considered as having the highest software priority in the decision process.

Different Interrupt Vector Sources

Two interrupt source types are managed by the ST7 interrupt controller: the non-maskable type (RESET, TLI, TRAP) and the maskable type (external or from internal peripherals).

Non-Maskable Sources

These sources are processed regardless of the state of the I1 and I0 bits of the CC register (see Figure 18). After stacking the PC, X, A and CC registers (except for RESET), the corresponding vector is loaded in the PC register and the I1 and I0 bits of the CC are set to disable interrupts (level 3). These sources allow the processor to exit HALT mode.

■ TLI (Top Level Hardware Interrupt)

This hardware interrupt occurs when a specific edge is detected on the dedicated TLI pin. Its detailed specification is given in the External Interrupts chapter.

■ TRAP (Non Maskable Software Interrupt)

This software interrupt is serviced when the TRAP instruction is executed. It will be serviced according to the flowchart on Figure 18 as a TLI.

■ RESET

The RESET source has the highest priority in the ST7. This means that the first current routine has the highest software priority (level 3) and the highest hardware priority.
See the RESET chapter for more details.

Maskable Sources

Maskable interrupt vector sources can be serviced if the corresponding interrupt is enabled and if its own interrupt software priority (in ISPRx registers) is higher than the one currently being serviced (I1 and I0 in CC register). If any of these two conditions is false, the interrupt is latched and thus remains pending.

■ External Interrupts

External interrupts allow the processor to exit from HALT low power mode.
External interrupt sensitivity is software selectable through the External Interrupt Control register (EICR).
External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.
If several input pins of a group connected to the same interrupt line are selected simultaneously, these will be logically ORed.

■ Peripheral Interrupts

Usually the peripheral interrupts cause the MCU to exit from HALT mode except those mentioned in the "Interrupt Mapping" table.
A peripheral interrupt occurs when a specific flag is set in the peripheral status registers and if the corresponding enable bit is set in the peripheral control register.
The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.
Note: The clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being serviced) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

7.3 INTERRUPTS AND LOW POWER MODES

All interrupts allow the processor to exit the WAIT low power mode. On the contrary, only external and other specified interrupts allow the processor to exit the HALT modes (see column "Exit from HALT" in "Interrupt Mapping" table). When several pending interrupts are present while exiting HALT mode, the first one serviced can only be an interrupt with exit from HALT mode capability and it is selected through the same decision process shown in Figure 19.

Note: If an interrupt, that is not able to Exit from HALT mode, is pending with the highest priority when exiting HALT mode, this interrupt is serviced after the first one serviced.

7.4 CONCURRENT & NESTED MANAGEMENT

The following Figure 20 and Figure 21 show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in Figure 21. The interrupt hardware priority is given in this order from the lowest to the highest: MAIN, IT4, IT3, IT2, IT1, IT0, TLI. The software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.

Figure 20. Concurrent Interrupt Management
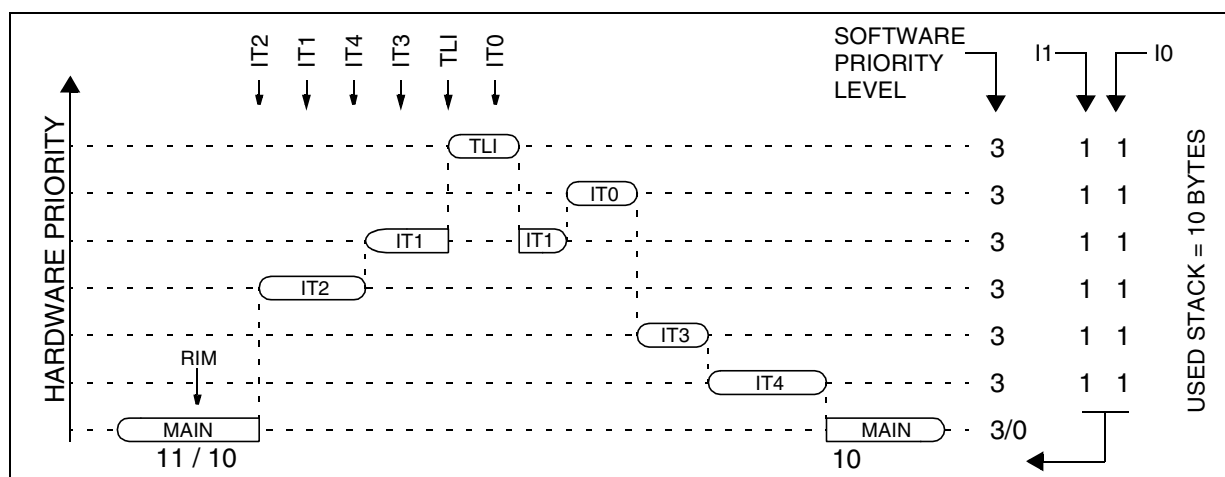
Figure 21. Nested Interrupt Management
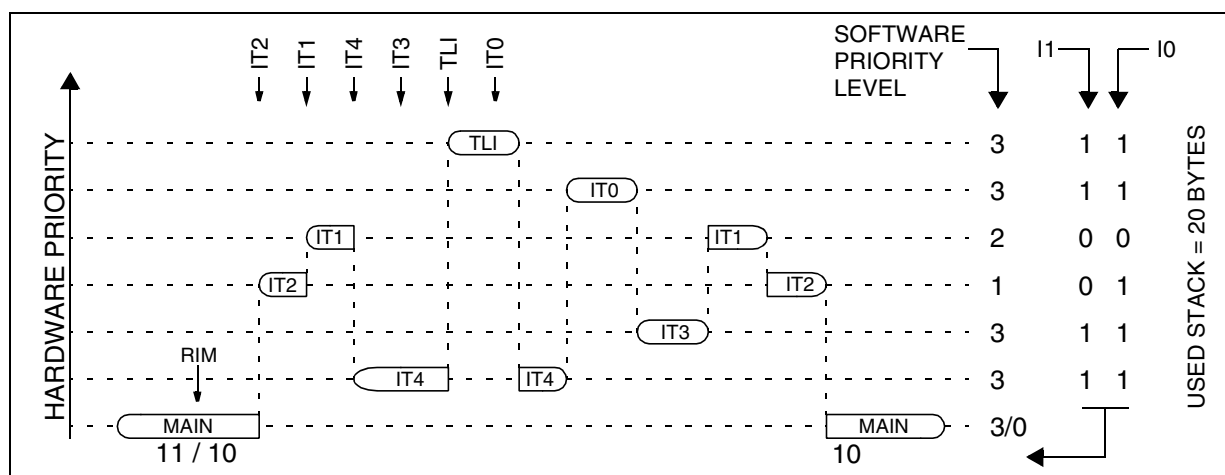
INTERRUPTS (Cont'd)

7.5 INTERRUPT REGISTER DESCRIPTION

CPU CC REGISTER INTERRUPT BITS

Read/Write

Reset Value: 111x 1010 (xAh)

7							0
1	1	**I1**	H	**I0**	N	Z	C
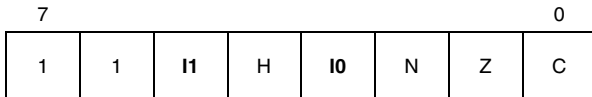
Bit 5, 3 = **I1, I0** *Software Interrupt Priority*

These two bits indicate the current interrupt software priority.

Interrupt Software Priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2		0	0
Level 3 (= interrupt disable*)	High	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see "Interrupt Dedicated Instruction Set" table).

*Note: TLI, TRAP and RESET events are non maskable sources and can interrupt a level 3 program.

INTERRUPT SOFTWARE PRIORITY REGISTERS (ISPRX)

Read/Write (bit 7:4 of **ISPR3** are read only)

Reset Value: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12

These four registers contain the interrupt software priority of each interrupt vector.

– Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondance is shown in the following table.

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_0 bits*
FFF9h-FFF8h	I1_1 and I0_1 bits
...	...
FFE1h-FFE0h	I1_13 and I0_13 bits

– Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

– Level 0 can not be written (I1_x=1, I0_x=0). In this case, the previously stored value is kept. (example: previous=CFh, write=64h, result=44h)

The RESET, TRAP and TLI vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

*Note: Bits in the ISPRx registers which correspond to the TLI can be read and written but they are not significant in the interrupt process management.

Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behaviour has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

INTERRUPTS (Cont'd)

Table 6. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT[1]	Address Vector
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software interrupt			no	FFFCh-FFFDh
0		Not used				FFFAh-FFFBh
1	MCC/RTC CSS	Main clock controller time base interrupt Safe oscillator activation interrupt	MCCSR SICSR		yes	FFF8h-FFF9h
2	ei0	External interrupt port A3..0	N/A		yes	FFF6h-FFF7h
3	ei1	External interrupt port F2..0			yes	FFF4h-FFF5h
4	ei2	External interrupt port B3..0			yes	FFF2h-FFF3h
5	ei3	External interrupt port B7..4			yes	FFF0h-FFF1h
6		Not used				FFEEh-FFEFh
7	SPI	SPI peripheral interrupts	SPICSR		yes	FFECh-FFEDh
8	TIMER A	TIMER A peripheral interrupts	TASR		no	FFEAh-FFEBh
9	TIMER B	TIMER B peripheral interrupts	TBSR		no	FFE8h-FFE9h
10	SCI	SCI Peripheral interrupts	SCISR		no	FFE6h-FFE7h
11	AVD	Auxiliary Voltage detector interrupt	SICSR		yes	FFE4h-FFE5h
12	I2C	I2C Peripheral interrupts	(see periph)	Lowest Priority	no	FFE2h-FFE3h
13	PWM ART	PWM ART overflow interrupt	ARTCSR		yes	FFE0h-FFE1h

Notes:

1. Valid for HALT and ACTIVE-HALT modes except for the MCC/RTC or CSS interrupt source which exits from ACTIVE-HALT mode only.

INTERRUPTS (Cont'd)

7.6 EXTERNAL INTERRUPTS

The pending interrupts are cleared writing a different value in the ISx[1:0], IPA or IPB bits of the EICR.

Note: External interrupts are masked when an I/O (configured as input interrupt) of the same interrupt vector is forced to V_{SS}.

7.6.1 I/O PORT INTERRUPT SENSITIVITY

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the EICR register (Figure 22). This control allows to have up to 4 fully independent external interrupt source sensitivities.

Each external interrupt source can be generated on four (or five) different events on the pin:

■ Falling edge
■ Rising edge
■ Falling and rising edge
■ Falling edge and low level
■ Rising edge and high level (only for ei0 and ei2)

To guarantee correct functionality, the sensitivity bits in the EICR register can be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3).

Figure 22. External Interrupt Control bits
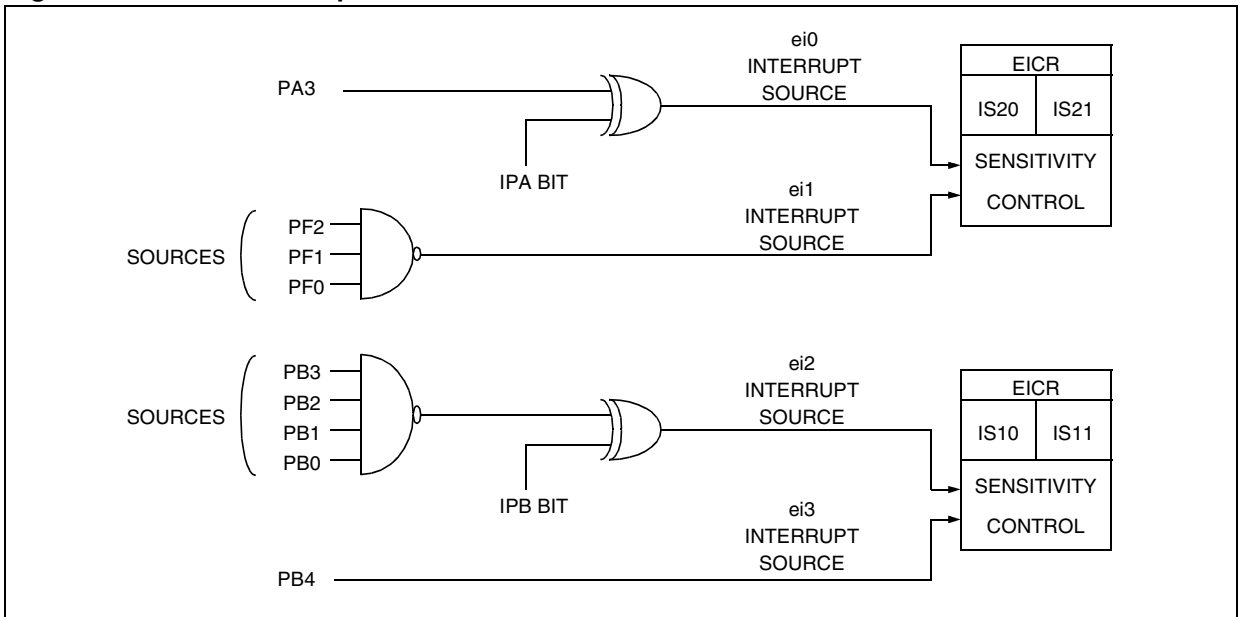
INTERRUPTS (Cont'd)

7.7 EXTERNAL INTERRUPT CONTROL REGISTER (EICR)

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

7							0
IS11	IS10	IPB	IS21	IS20	IPA	0	0

Bit 7:6 = **IS1[1:0]** *ei2 and ei3 sensitivity*
The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts:
- ei2 (port B3..0)

IS11	IS10	External Interrupt Sensitivity	
		IPB bit =0	**IPB bit =1**
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- ei3 (port B4)

IS11	IS10	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 5 = **IPB** *Interrupt polarity for port B*
This bit is used to invert the sensitivity of the port B [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).
0: No sensitivity inversion
1: Sensitivity inversion

Bit 4:3 = **IS2[1:0]** *ei0 and ei1 sensitivity*
The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts:

- ei0 (port A3..0)

IS21	IS20	External Interrupt Sensitivity	
		IPA bit =0	**IPA bit =1**
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- ei1 (port F2..0)

IS21	IS20	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 2 = **IPA** *Interrupt polarity for port A*
This bit is used to invert the sensitivity of the port A [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).
0: No sensitivity inversion
1: Sensitivity inversion

Bits 1:0 = Reserved, must always be kept cleared.

INTERRUPTS (Cont'd)

Table 7. Nested Interrupts Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0024h	**ISPR0**	ei1		ei0		MCC + SI			
		I1_3	I0_3	I1_2	I0_2	I1_1	I0_1		
	Reset Value	1	1	1	1	1	1	1	1
0025h	**ISPR1**	SPI				ei3		ei2	
		I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
	Reset Value	1	1	1	1	1	1	1	1
0026h	**ISPR2**	AVD		SCI		TIMER B		TIMER A	
		I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
	Reset Value	1	1	1	1	1	1	1	1
0027h	**ISPR3**					PWMART		I2C	
						I1_13	I0_13	I1_12	I0_12
	Reset Value	1	1	1	1	1	1	1	1
0028h	**EICR**	IS11	IS10	IPB	IS21	IS20	IPA		
	Reset Value	0	0	0	0	0	0	0	0

8 POWER SAVING MODES

8.1 INTRODUCTION

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see Figure 23): SLOW, WAIT (SLOW WAIT), ACTIVE HALT and HALT.

After a RESET the normal operating mode is selected by default (RUN mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From RUN mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 23. Power Saving Mode Transitions
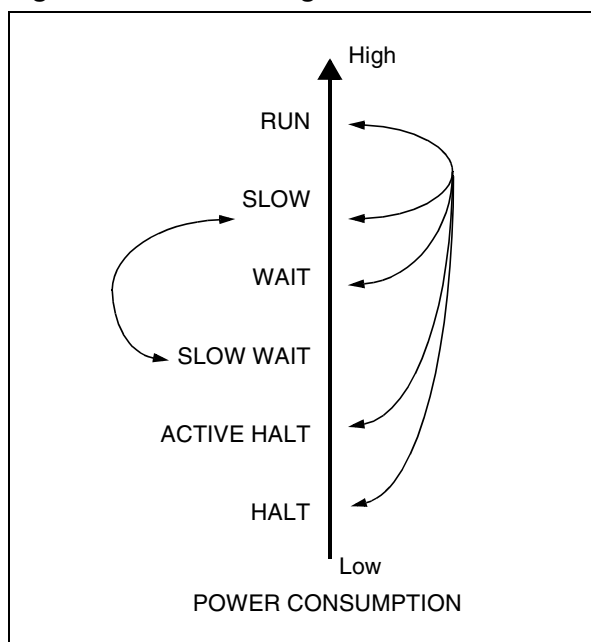
8.2 SLOW MODE

This mode has two targets:

– To reduce power consumption by decreasing the internal clock in the device,

– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the master clock frequency (f_{OSC2}) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency (f_{CPU}).

Note: SLOW-WAIT mode is activated when entering the WAIT mode while the device is already in SLOW mode.

Figure 24. SLOW Mode Clock Transitions
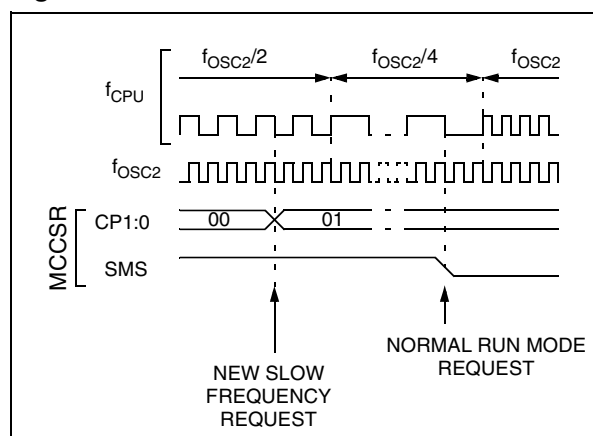
POWER SAVING MODES (Cont'd)

8.3 WAIT MODE

WAIT mode places the MCU in a low power consumption mode by stopping the CPU.
This power saving mode is selected by calling the 'WFI' instruction.
All peripherals remain active. During WAIT mode, the I[1:0] bits of the CC register are forced to '10', to enable all interrupts. All other registers and memory remain unchanged. The MCU remains 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 25.

Figure 25. WAIT Mode Flow-chart
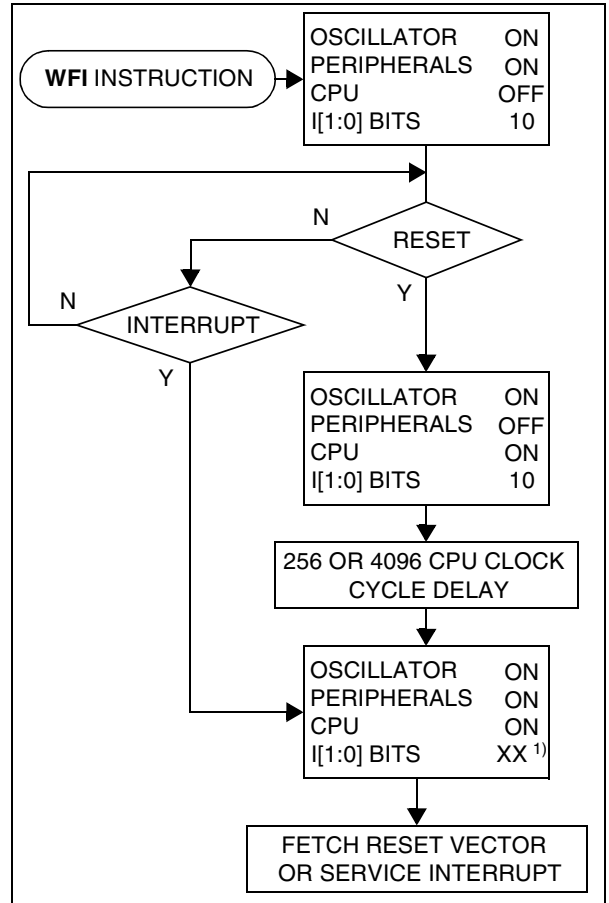
Note:

1. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

POWER SAVING MODES (Cont'd)

8.4 ACTIVE-HALT AND HALT MODES

ACTIVE-HALT and HALT modes are the two lowest power consumption modes of the MCU. They are both entered by executing the 'HALT' instruction. The decision to enter either in ACTIVE-HALT or HALT mode is given by the MCC/RTC interrupt enable flag (OIE bit in MCCSR register).

MCCSR OIE bit	Power Saving Mode entered when HALT instruction is executed
0	HALT mode
1	ACTIVE-HALT mode

8.4.1 ACTIVE-HALT MODE

ACTIVE-HALT mode is the lowest power consumption mode of the MCU with a real time clock available. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCSR) is set (see Section 10.2 on page 52 for more details on the MCCSR register).

The MCU can exit ACTIVE-HALT mode on reception of either an MCC/RTC interrupt, a specific interrupt (see Table 6, "Interrupt Mapping," on page 35) or a RESET. When exiting ACTIVE-HALT mode by means of an interrupt, no 256 or 4096 CPU cycle delay occurs. The CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 27).

When entering ACTIVE-HALT mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In ACTIVE-HALT mode, only the main oscillator and its associated counter (MCC/RTC) are running to keep a wake-up time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

The safeguard against staying locked in ACTIVE-HALT mode is provided by the oscillator interrupt.

Note: As soon as the interrupt capability of one of the oscillators is selected (MCCSR.OIE bit set), entering ACTIVE-HALT mode while the Watchdog is active does not generate a RESET.
This means that the device cannot spend more than a defined delay in this power saving mode.

CAUTION: When exiting ACTIVE-HALT mode following an interrupt, OIE bit of MCCSR register must not be cleared before t_{DELAY} after the interrupt occurs (t_{DELAY} = 256 or 4096 t_{CPU} delay de-

pending on option byte). Otherwise, the ST7 enters HALT mode for the remaining t_{DELAY} period.

Figure 26. ACTIVE-HALT Timing Overview
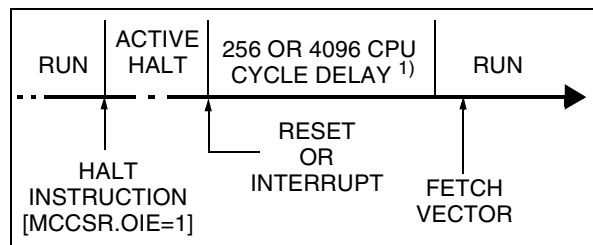
Figure 27. ACTIVE-HALT Mode Flow-chart
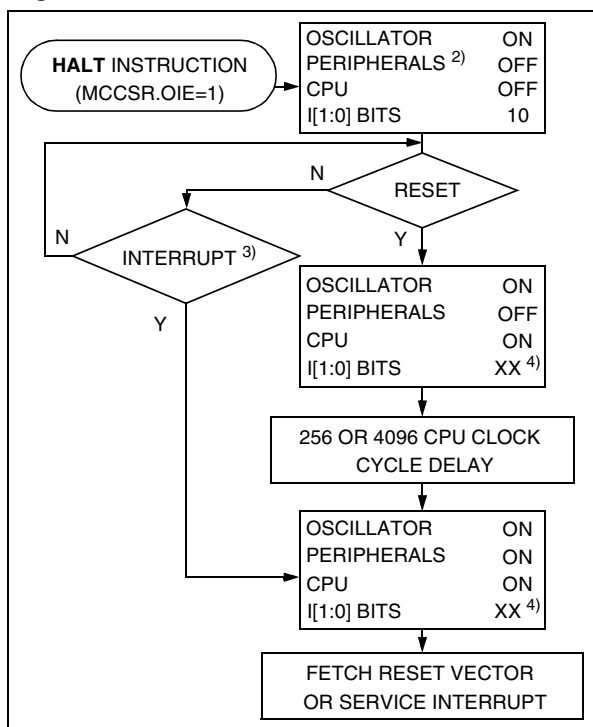
Notes:

1. This delay occurs only if the MCU exits ACTIVE-HALT mode by means of a RESET.
2. Peripheral clocked with an external clock source can still be active.
3. Only the MCC/RTC interrupt and some specific interrupts can exit the MCU from ACTIVE-HALT mode (such as external interrupt). Refer to Table 6, "Interrupt Mapping," on page 35 for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and restored when the CC register is popped.

POWER SAVING MODES (Cont'd)

8.4.2 HALT MODE

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCSR) is cleared (see Section 10.2 on page 52 for more details on the MCCSR register).

The MCU can exit HALT mode on reception of either a specific interrupt (see Table 6, "Interrupt Mapping," on page 35) or a RESET. When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 256 or 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 29).

When entering HALT mode, the I[1:0] bits in the CC register are forced to '10b'to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In HALT mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with HALT mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET (see Section 14.1 on page 169 for more details).

Figure 28. HALT Timing Overview
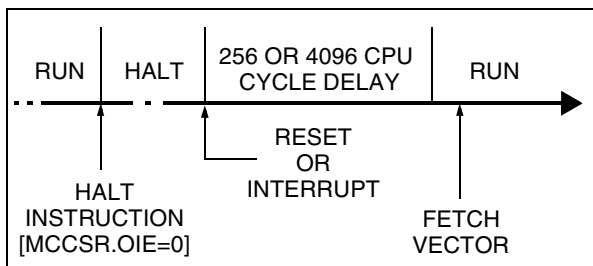
Figure 29. HALT Mode Flow-chart
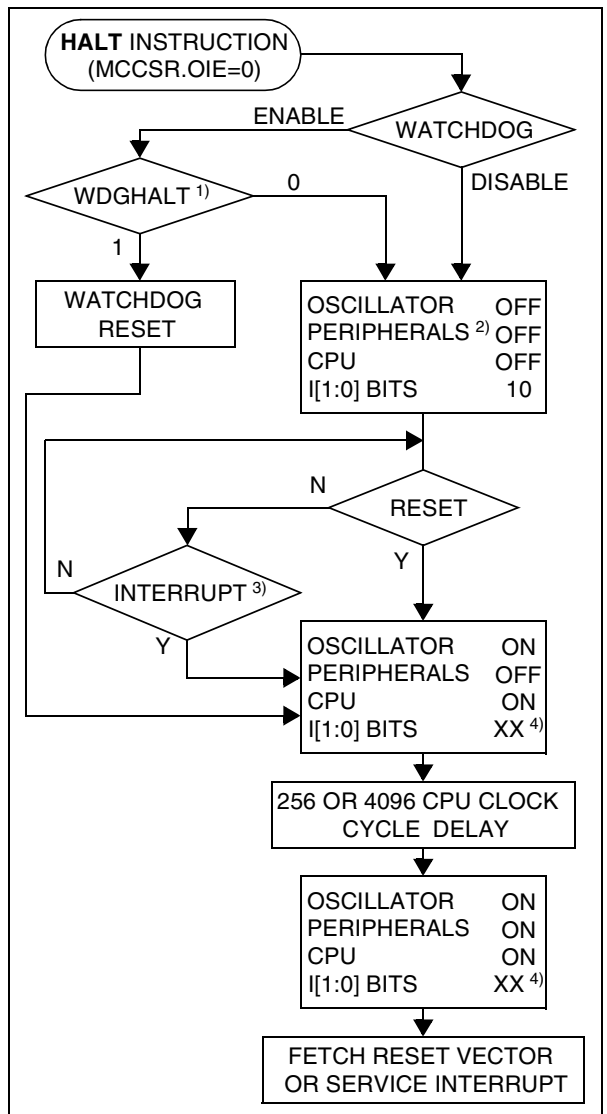
Notes:

1. WDGHALT is an option bit. See option byte section for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to Table 6, "Interrupt Mapping," on page 35 for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

9 I/O PORTS

9.1 INTRODUCTION

The I/O ports offer different functional modes:
– transfer of data through digital inputs and outputs

and for specific pins:
– external interrupt generation
– alternate signal input/output for the on-chip peripherals.

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

9.2 FUNCTIONAL DESCRIPTION

Each port has 2 main registers:

– Data Register (DR)

– Data Direction Register (DDR)

and one optional register:

– Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the 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). The generic I/O block diagram is shown in Figure 30

9.2.1 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. Writing the DR register modifies the latch value but does not affect the pin status.
2. When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

External interrupt function

When an I/O is configured as Input with Interrupt, an event on this I/O can generate an external interrupt request to the CPU.

Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the EICR register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt source, these are logically ANDed. For this reason if one of the interrupt pins is tied low, it masks the other ones.

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the EICR register must be modified.

9.2.2 Output Modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing 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.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	V_{SS}	Vss
1	V_{DD}	Floating

9.2.3 Alternate Functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the 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 must be configured in input mode. In this case, the pin state is also digitally readable by addressing the DR register.

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.

I/O PORTS (Cont'd)

Figure 30. I/O Port General Block Diagram
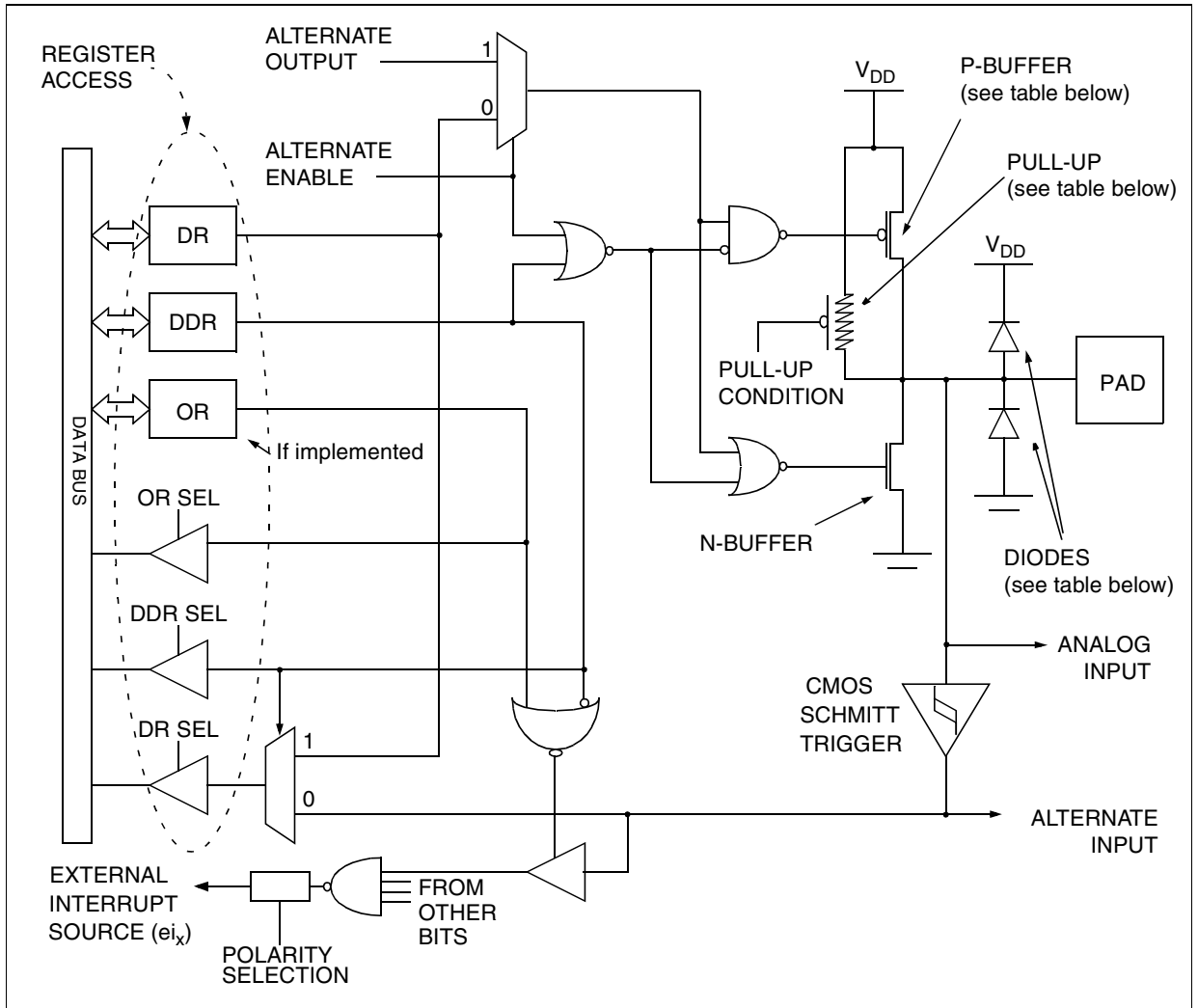
Table 8. I/O Port Mode Options

Configuration Mode		Pull-Up	P-Buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open Drain (logic level)		Off		
	True Open Drain	NI	NI	NI (see note)	

Legend: NI - not implemented
Off - implemented not activated
On - implemented and activated

Note: The diode to V_{DD} is not implemented in the true open drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

I/O PORTS (Cont'd)

Table 9. I/O Port Configurations
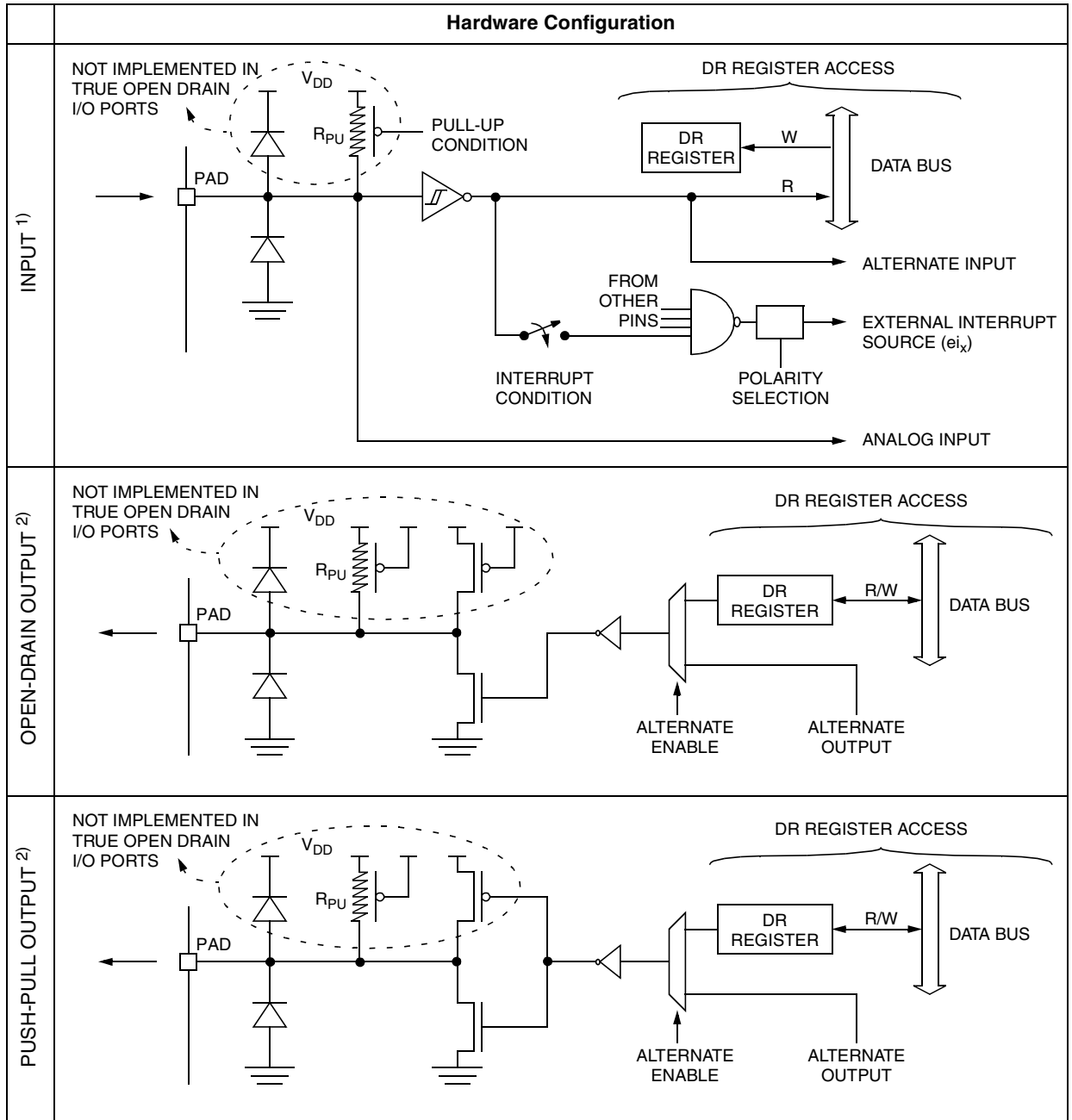
Notes:

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.

2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

I/O PORTS (Cont'd)

CAUTION: 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.

Analog alternate function

When the pin is used as an ADC input, the I/O must be configured as floating input. 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.

9.3 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 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 31 Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 31. Interrupt I/O Port State Transitions
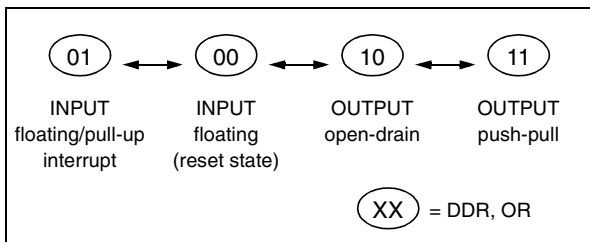
9.4 LOW POWER MODES

Mode	Description
WAIT	No effect on I/O ports. External interrupts cause the device to exit from WAIT mode.
HALT	No effect on I/O ports. External interrupts cause the device to exit from HALT mode.

9.5 INTERRUPTS

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and the interrupt mask in the CC register is not active (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
External interrupt on selected external event	-	DDRx ORx	Yes	Yes

I/O PORTS (Cont'd)

9.5.1 I/O Port Implementation

The I/O port register configurations are summarised as follows.

Standard Ports

**PA5:4, PC7:0, PD5:0,
PE1:0, PF7:6, 4**

MODE	DDR	OR
floating input	0	0
pull-up input	0	1
open drain output	1	0
push-pull output	1	1

Interrupt Ports

PB4, PB2:0, PF1:0 (with pull-up)

MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

PA3, PB3, PF2 (without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
open drain output	1	0
push-pull output	1	1

True Open Drain Ports

PA7:6

MODE	DDR
floating input	0
open drain (high sink ports)	1

Table 10. Port Configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:6	floating		true open-drain	
	PA5:4	floating	pull-up	open drain	push-pull
	PA3	floating	floating interrupt	open drain	push-pull
Port B	PB3	floating	floating interrupt	open drain	push-pull
	PB4, PB2:0	floating	pull-up interrupt	open drain	push-pull
Port C	PC7:0	floating	pull-up	open drain	push-pull
Port D	PD5:0	floating	pull-up	open drain	push-pull
Port E	PE1:0	floating	pull-up	open drain	push-pull
Port F	PF7:6, 4	floating	pull-up	open drain	push-pull
	PF2	floating	floating interrupt	open drain	push-pull
	PF1:0	floating	pull-up interrupt	open drain	push-pull

I/O PORTS (Cont'd)

Table 11. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Reset Value of all I/O port registers		0	0	0	0	0	0	0	0
0000h	**PADR**	MSB							LSB
0001h	**PADDR**								
0002h	**PAOR**								
0003h	**PBDR**	MSB							LSB
0004h	**PBDDR**								
0005h	**PBOR**								
0006h	**PCDR**	MSB							LSB
0007h	**PCDDR**								
0008h	**PCOR**								
0009h	**PDDR**	MSB							LSB
000Ah	**PDDDR**								
000Bh	**PDOR**								
000Ch	**PEDR**	MSB							LSB
000Dh	**PEDDR**								
000Eh	**PEOR**								
000Fh	**PFDR**	MSB							LSB
0010h	**PFDDR**								
0011h	**PFOR**								

10 ON-CHIP PERIPHERALS

10.1 WATCHDOG TIMER (WDG)

10.1.1 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.

10.1.2 Main Features

- Programmable timer (64 increments of 12288 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero

- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

10.1.3 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 32. Watchdog Block Diagram
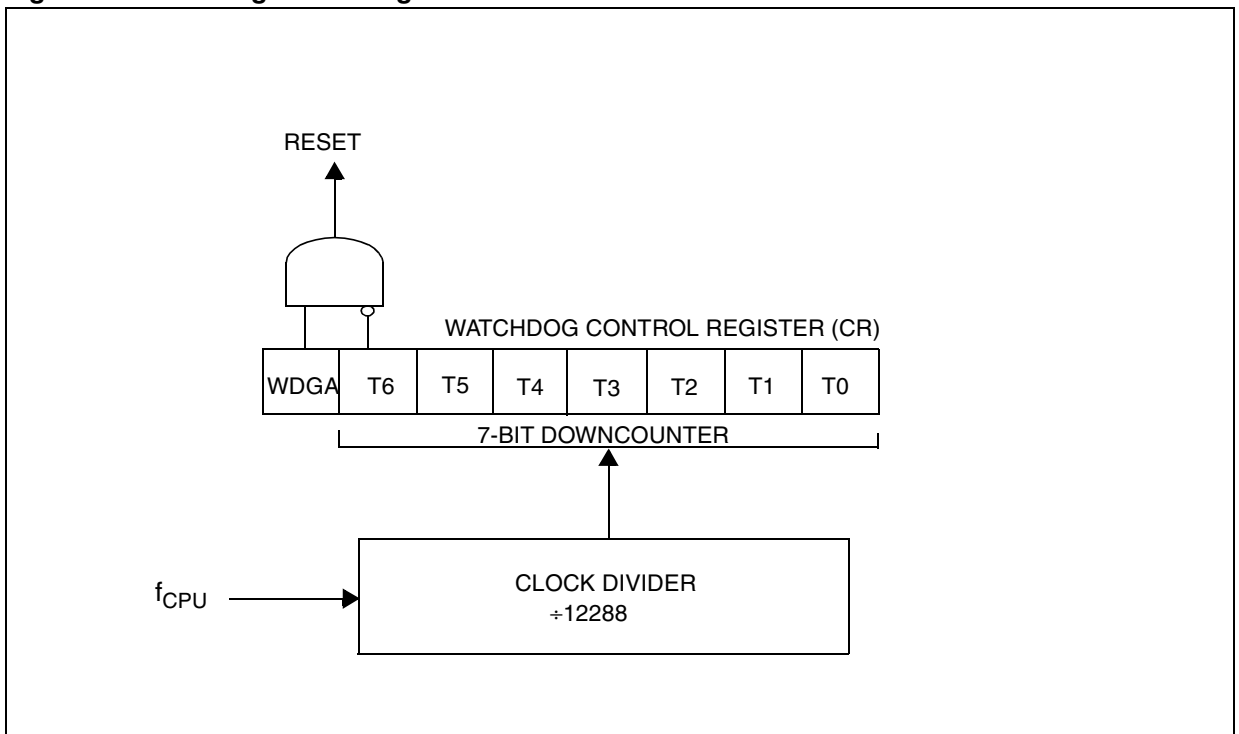
WATCHDOG TIMER (Cont'd)

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 .Watchdog Timing):

– 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.

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.

Table 12.Watchdog Timing

f_{CPU} = 8MHz		
WDG Counter Code	**min [ms]**	**max [ms]**
C0h	1.536	2.048
FFh	130.560	131.072

Notes: The timing variation shown in Table 12 is due to the unknown status of the prescaler when writing to the CR register.

10.1.4 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 Option Byte description.

10.1.5 Using Halt Mode with the WDG (WDGHALT option)

The following recommendations apply if Halt mode is used when the watchdog is enabled.

Recommendations

– Make sure that an external event is available to wake up the microcontroller from Halt mode.

– Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

– When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as "Input Pull-up with Interrupt" before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.

– For the same reason, reinitialize the level sensitiveness of each external interrupt as a precautionary measure.

– The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in ROM with the value 0x8E.

– As the HALT instruction clears the interrupt mask in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

WATCHDOG TIMER (Cont'd)

10.1.6 Low Power Modes

Mode	Description		
WAIT	No effect on Watchdog.		
HALT	**OIE bit in MCCSR register**	**WDGHALT bit in Option Byte**	
	0	0	No Watchdog reset is generated. The MCU enters Halt mode. The Watchdog counter is decremented once and then stops counting and is no longer able to generate a watchdog reset until the MCU receives an external interrupt or a reset.
			If an external interrupt is received, the Watchdog restarts counting after 4096 CPU clocks. If a reset is generated, the Watchdog is disabled (reset state) unless Hardware Watchdog is selected by option byte. For application recommendations see Section 10.1.5 below.
	0	1	A reset is generated.
	1	x	No reset is generated. The MCU enters Active Halt mode. The Watchdog counter is not decremented. It stop counting. When the MCU receives an oscillator interrupt, external interrupt or a reset the Watchdog restarts counting after 4096 CPU clocks.

10.1.7 Interrupts

None.

10.1.8 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).

Table 13. Watchdog Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Ah	**WDGCR** Reset Value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real time clock timer with interrupt capability

Each function can be used independently and simultaneously.

10.2.1 Programmable CPU Clock Prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages SLOW power saving mode (See Section 8.2 SLOW MODE for more details).

The prescaler selects the f_{CPU} main clock frequency and is controlled by three bits in the MCCSR register: CP[1:0] and SMS.

CAUTION: The prescaler does not act on the CAN peripheral clock source. This peripheral is always supplied by the f_{OSC2} clock source.

10.2.2 Clock-out Capability

The clock-out capability is an alternate function of an I/O port pin that outputs a f_{OSC2} clock to drive external devices. It is controlled by the MCO bit in the MCCSR register.
CAUTION: When selected, the clock out pin suspends the clock during ACTIVE-HALT mode.

10.2.3 Real Time Clock Timer (RTC)

The counter of the real time clock timer allows an interrupt to be generated based on an accurate real time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters ACTIVE-HALT mode when the HALT instruction is executed. See Section 8.4 ACTIVE-HALT AND HALT MODES for more details.

10.2.4 Beeper

The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 33. Main Clock Controller (MCC/RTC) Block Diagram
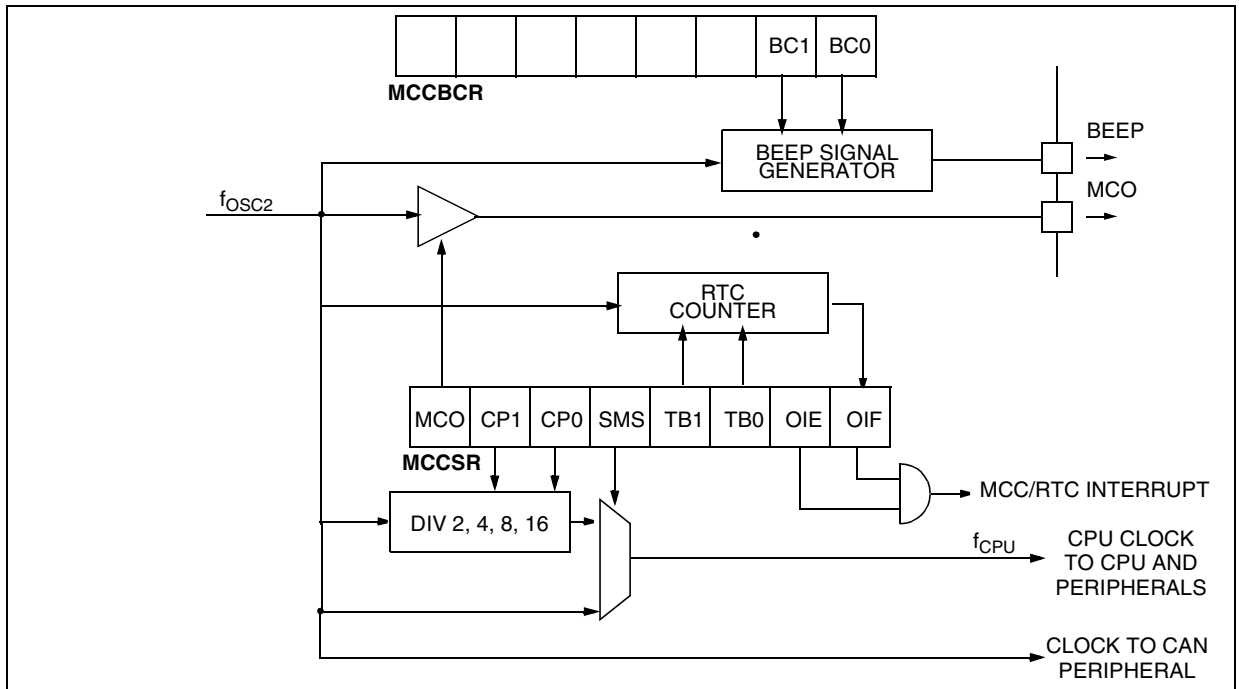
MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (Cont'd)

10.2.5 Low Power Modes

Mode	Description
WAIT	No effect on MCC/RTC peripheral. MCC/RTC interrupt cause the device to exit from WAIT mode.
ACTIVE-HALT	No effect on MCC/RTC counter (OIE bit is set), the registers are frozen. MCC/RTC interrupt cause the device to exit from ACTIVE-HALT mode.
HALT	MCC/RTC counter and registers are frozen. MCC/RTC operation resumes when the MCU is woken up by an interrupt with "exit from HALT" capability.

10.2.6 Interrupts

The MCC/RTC interrupt event generates an interrupt if the OIE bit of the MCCSR register is set and the interrupt mask in the CC register is not active (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Time base overflow event	OIF	OIE	Yes	No [1]

Note:

The MCC/RTC interrupt wakes up the MCU from ACTIVE-HALT mode, not from HALT mode.

10.2.7 Register Description
MCC CONTROL/STATUS REGISTER (MCCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7　　　　　　　　　　　　　　　0

MCO	CP1	CP0	SMS	TB1	TB0	OIE	OIF

Bit 7 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software.
0: MCO alternate function disabled (I/O pin free for general-purpose I/O)
1: MCO alternate function enabled (f_{OSC2} on I/O port)

Note: To reduce power consumption, the MCO function is not active in ACTIVE-HALT mode.

Bit 6:5 = **CP[1:0]** *CPU clock prescaler*
These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f_{CPU} in SLOW mode	CP1	CP0
f_{OSC2} / 2	0	0
f_{OSC2} / 4	0	1
f_{OSC2} / 8	1	0
f_{OSC2} / 16	1	1

Bit 4 = **SMS** *Slow mode select*
This bit is set and cleared by software.
0: Normal mode. $f_{CPU} = f_{OSC2}$
1: Slow mode. f_{CPU} is given by CP1, CP0
See Section 8.2 SLOW MODE and Section 10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC) for more details.

Bit 3:2 = **TB[1:0]** *Time base control*

These bits select the programmable divider time base. They are set and cleared by software.

Counter Prescaler	Time Base		TB1	TB0
	f_{OSC2} =4MHz	f_{OSC2}=8MHz		
32000	4ms	2ms	0	0
64000	8ms	4ms	0	1
160000	20ms	10ms	1	0
400000	50ms	25ms	1	1

A modification of the time base is taken into account at the end of the current period (previously set) to avoid an unwanted time shift. This allows to use this time base as a real time clock.

Bit 1 = **OIE** *Oscillator interrupt enable*
This bit set and cleared by software.
0: Oscillator interrupt disabled
1: Oscillator interrupt enabled
This interrupt can be used to exit from ACTIVE-HALT mode.
When this bit is set, calling the ST7 software HALT instruction enters the ACTIVE-HALT power saving mode.

MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (Cont'd)

Bit 0 = **OIF** *Oscillator interrupt flag*
This bit is set by hardware and cleared by software reading the MCCSR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0).
0: Timeout not reached
1: Timeout reached

CAUTION: The BRES and BSET instructions must not be used on the MCCSR register to avoid unintentionally clearing the OIF bit.

MCC BEEP CONTROL REGISTER (MCCBCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	BC1	BC0

Bit 7:2 = Reserved, must be kept cleared.

Bit 1:0 = **BC[1:0]** *Beep control*
These 2 bits select the PF1 pin beep capability.

BC1	BC0	Beep mode with f_{OSC2}=8MHz	
0	0	Off	
0	1	~2-KHz	Output Beep signal ~50% duty cycle
1	0	~1-KHz	
1	1	~500-Hz	

The beep output signal is available in ACTIVE-HALT mode but has to be disabled to reduce the consumption.

Table 14. Main Clock Controller Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Bh	**SICSR** Reset Value	VDS 0	VDIE 0	VDF 0	LVDRF x	0	CFIE 0	CSSD 0	WDGRF x
002Ch	**MCCSR** Reset Value	MCO 0	CP1 0	CP0 0	SMS 0	TB1 0	TB0 0	OIE 0	OIF 0
002Dh	**MCCBCR** Reset Value	0	0	0	0	0	0	BC1 0	BC0 0

10.3 PWM AUTO-RELOAD TIMER (ART)

10.3.1 Introduction

The Pulse Width Modulated Auto-Reload Timer on-chip peripheral consists of an 8-bit auto reload counter with compare/capture capabilities and of a 7-bit prescaler clock source.

These resources allow five possible operating modes:

– Generation of up to 4 independent PWM signals

– Output compare and Time base interrupt

– Up to two input capture functions

– External event detector

– Up to two external interrupt sources

The three first modes can be used together with a single counter frequency.

The timer can be used to wake up the MCU from WAIT and HALT modes.

Figure 34. PWM Auto-Reload Timer Block Diagram
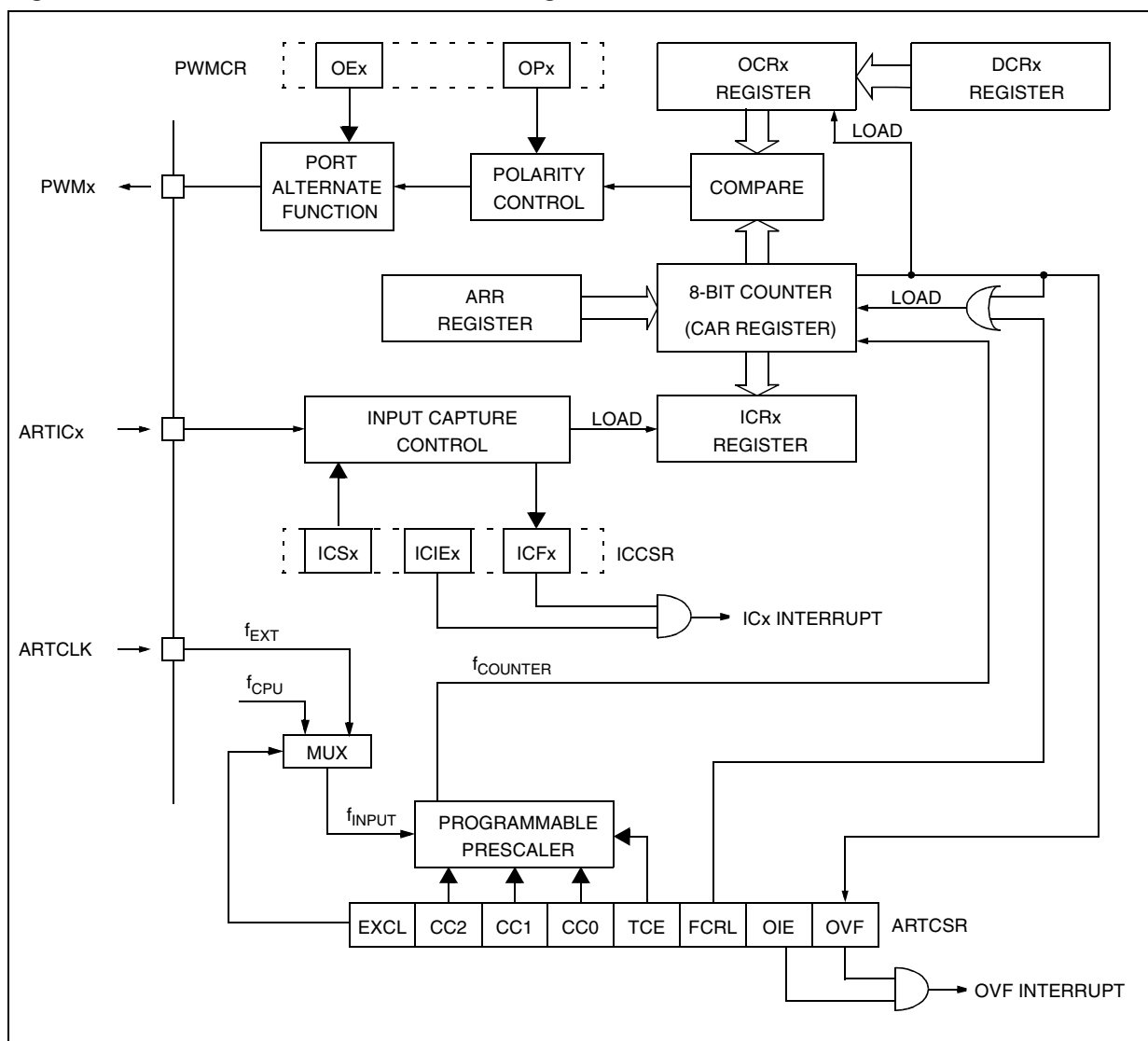
PWM AUTO-RELOAD TIMER (Cont'd)

10.3.2 Functional Description

Counter

The free running 8-bit counter is fed by the output of the prescaler, and is incremented on every rising edge of the clock signal.

It is possible to read or write the contents of the counter on the fly by reading or writing the Counter Access register (CAR).

When a counter overflow occurs, the counter is automatically reloaded with the contents of the ARR register (the prescaler is not affected).

Counter clock and prescaler

The counter clock frequency is given by:

$$f_{COUNTER} = f_{INPUT} / 2^{CC[2:0]}$$

The timer counter's input clock (f_{INPUT}) feeds the 7-bit programmable prescaler, which selects one of the 8 available taps of the prescaler, as defined by CC[2:0] bits in the Control/Status Register (CSR). Thus the division factor of the prescaler can be set to 2^n (where n = 0, 1,..7).

This f_{INPUT} frequency source is selected through the EXCL bit of the CSR register and can be either the f_{CPU} or an external input frequency f_{EXT}.

The clock input to the counter is enabled by the TCE (Timer Counter Enable) bit in the CSR register. When TCE is reset, the counter is stopped and the prescaler and counter contents are frozen.

When TCE is set, the counter runs at the rate of the selected clock source.

Counter and Prescaler Initialization

After RESET, the counter and the prescaler are cleared and $f_{INPUT} = f_{CPU}$.

The counter can be initialized by:

– Writing to the ARR register and then setting the FCRL (Force Counter Re-Load) and the TCE (Timer Counter Enable) bits in the CSR register.

– Writing to the CAR counter access register,

In both cases the 7-bit prescaler is also cleared, whereupon counting will start from a known value.

Direct access to the prescaler is not possible.

Output compare control

The timer compare function is based on four different comparisons with the counter (one for each PWMx output). Each comparison is made between the counter value and an output compare register (OCRx) value. This OCRx register can not be accessed directly, it is loaded from the duty cycle register (DCRx) at each overflow of the counter.

This double buffering method avoids glitch generation when changing the duty cycle on the fly.

Figure 35. Output compare control
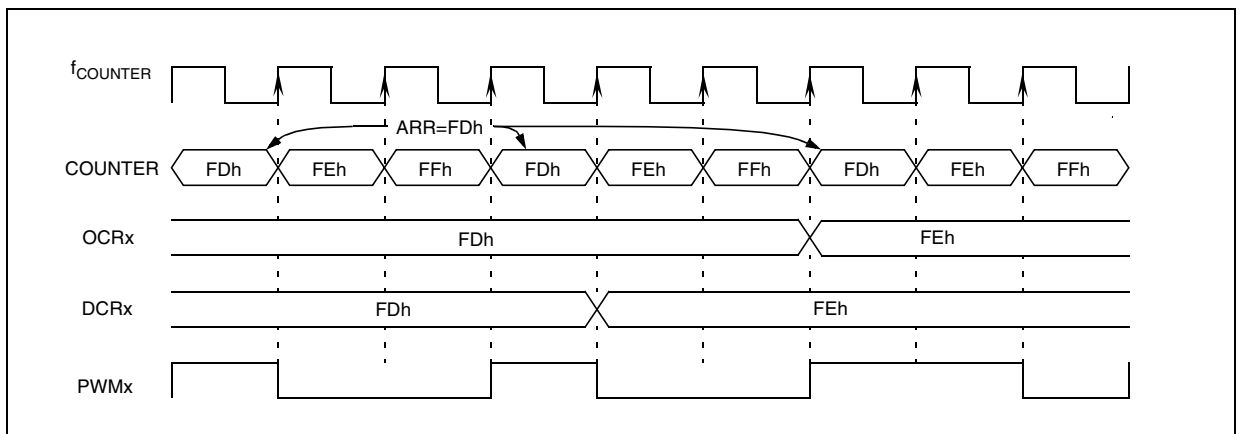
PWM AUTO-RELOAD TIMER (Cont'd)

Independent PWM signal generation

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins with minimum core processing overhead. This function is stopped during HALT mode.

Each PWMx output signal can be selected independently using the corresponding OEx bit in the PWM Control register (PWMCR). When this bit is set, the corresponding I/O pin is configured as output push-pull alternate function.

The PWM signals all have the same frequency which is controlled by the counter period and the ARR register value.

$$f_{PWM} = f_{COUNTER} / (256 - ARR)$$

When a counter overflow occurs, the PWMx pin level is changed depending on the corresponding OPx (output polarity) bit in the PWMCR register. When the counter reaches the value contained in one of the output compare register (OCRx) the corresponding PWMx pin level is restored.

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of the PWM output signal. To obtain a signal on a PWMx pin, the contents of the OCRx register must be greater than the contents of the ARR register.

The maximum available resolution for the PWMx duty cycle is:

$$Resolution = 1 / (256 - ARR)$$

Note: To get the maximum resolution (1/256), the ARR register must be 0. With this maximum resolution, 0% and 100% can be obtained by changing the polarity.

Figure 36. PWM Auto-reload Timer Function
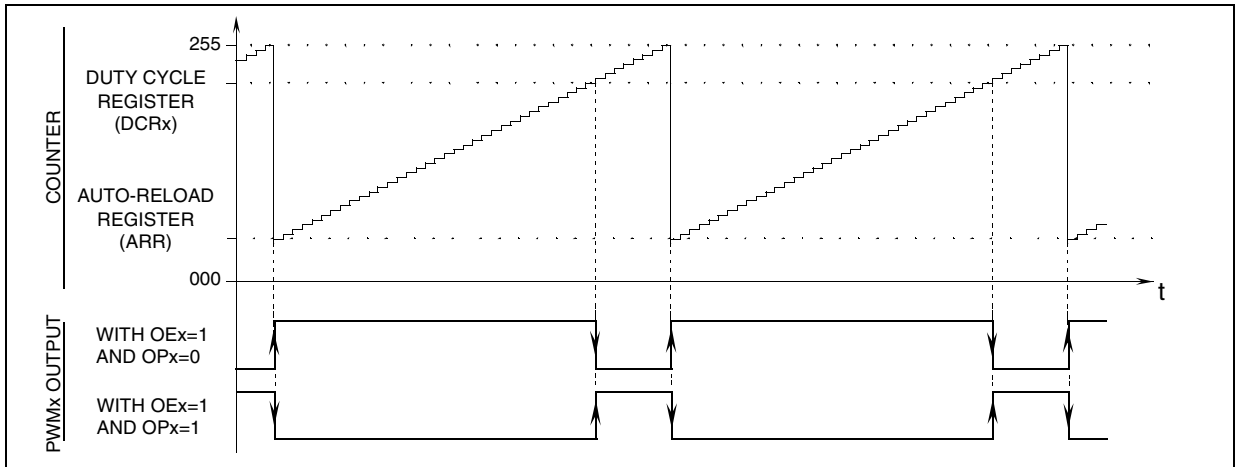
Figure 37. PWM Signal from 0% to 100% Duty Cycle
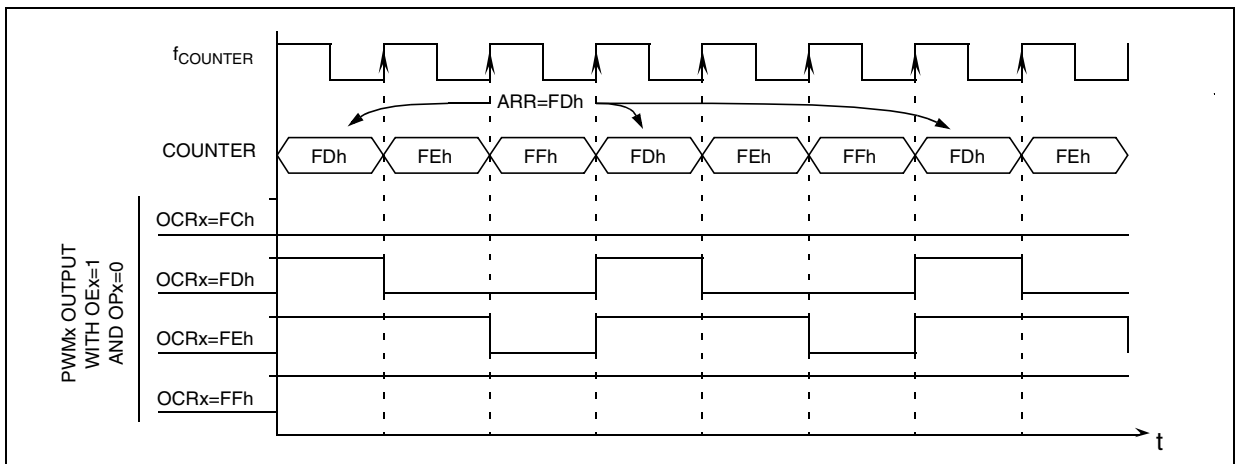
PWM AUTO-RELOAD TIMER (Cont'd)

Output compare and Time base interrupt

On overflow, the OVF flag of the CSR register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OIE, in the CSR register, is set. The OVF flag must be reset by the user software. This interrupt can be used as a time base in the application.

External clock and event detector mode

Using the f_{EXT} external prescaler input clock, the auto-reload timer can be used as an external clock event detector. In this mode, the ARR register is used to select the n_{EVENT} number of events to be counted before setting the OVF flag.

$$n_{EVENT} = 256 - ARR$$

When entering HALT mode while f_{EXT} is selected, all the timer control registers are frozen but the counter continues to increment. If the OIE bit is set, the next overflow of the counter will generate an interrupt which wakes up the MCU.

Figure 38. External Event Detector Example (3 counts)
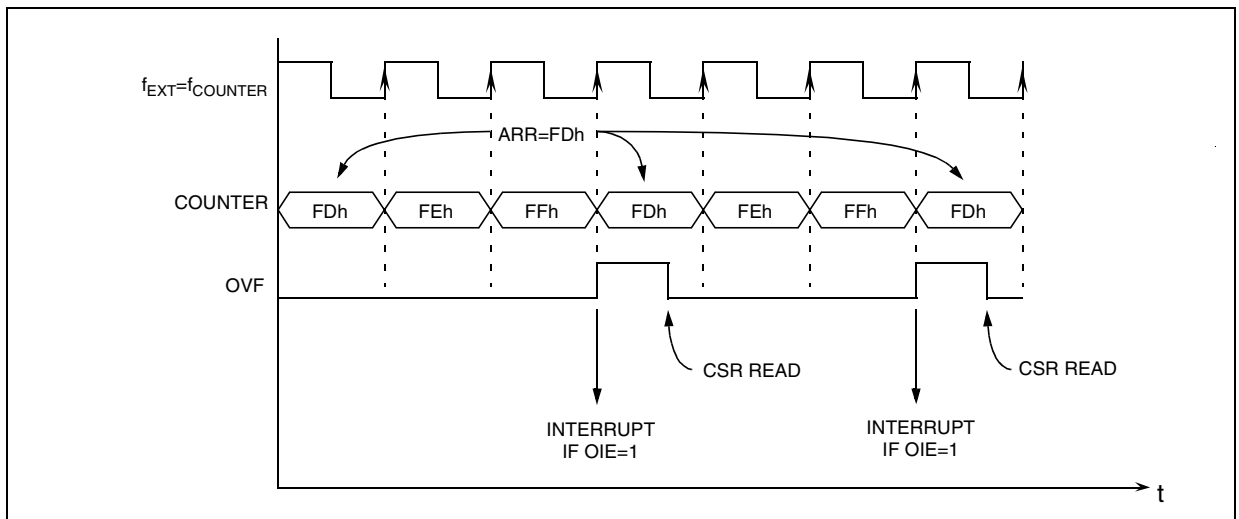
PWM AUTO-RELOAD TIMER (Cont'd)

Input capture function

This mode allows the measurement of external signal pulse widths through ICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ICCSR).

These input capture interrupts are enabled through the CIEx bits of the ICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ICCSR register.

The read only input capture registers (ICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ICRx register is inhibited until it is read (clearing the CFx bit).
The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means, the ICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time (1/$f_{COUNTER}$).

Note: During HALT mode, if both input capture and external clock are enabled, the ICRx register value is not guaranteed if the input capture pin and the external clock change simultaneously.

External interrupt capability

This mode allows the Input capture capabilities to be used as external interrupt sources. The interrupts are generated on the edge of the ARTICx signal.

The edge sensitivity of the external interrupts is programmable (CSx bit of ICCSR register) and they are independently enabled through CIEx bits of the ICCSR register. After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

During HALT mode, the external interrupts can be used to wake up the micro (if the CIEx bit is set).

Figure 39. Input Capture Timing Diagram

PWM AUTO-RELOAD TIMER (Cont'd)

10.3.3 Register Description

CONTROL / STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
EXCL	CC2	CC1	CC0	TCE	FCRL	OIE	OVF

Bit 7 = **EXCL** *External Clock*
This bit is set and cleared by software. It selects the input clock for the 7-bit prescaler.
0: CPU clock.
1: External clock.

Bit 6:4 = **CC[2:0]** *Counter Clock Control*
These bits are set and cleared by software. They determine the prescaler division ratio from f_{INPUT}.

$f_{COUNTER}$	With f_{INPUT}=8 MHz	CC2	CC1	CC0
f_{INPUT}	8 MHz	0	0	0
f_{INPUT} / 2	4 MHz	0	0	1
f_{INPUT} / 4	2 MHz	0	1	0
f_{INPUT} / 8	1 MHz	0	1	1
f_{INPUT} / 16	500 KHz	1	0	0
f_{INPUT} / 32	250 KHz	1	0	1
f_{INPUT} / 64	125 KHz	1	1	0
f_{INPUT} / 128	62.5 KHz	1	1	1

Bit 3 = **TCE** *Timer Counter Enable*
This bit is set and cleared by software. It puts the timer in the lowest power consumption mode.
0: Counter stopped (prescaler and counter frozen).
1: Counter running.

Bit 2 = **FCRL** *Force Counter Re-Load*
This bit is write-only and any attempt to read it will yield a logical zero. When set, it causes the contents of ARR register to be loaded into the counter, and the content of the prescaler register to be cleared in order to initialize the timer before starting to count.
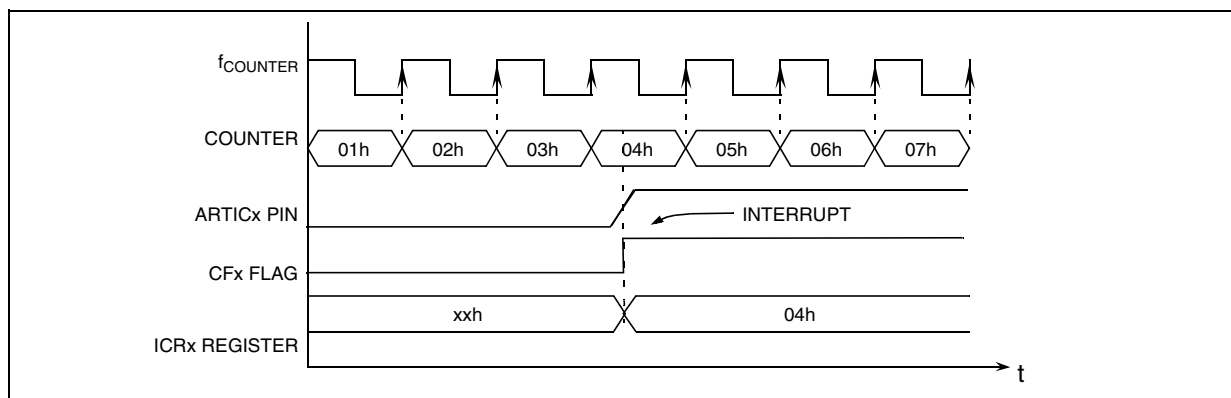
Bit 1 = **OIE** *Overflow Interrupt Enable*
This bit is set and cleared by software. It allows to enable/disable the interrupt which is generated when the OVF bit is set.
0: Overflow Interrupt disable.
1: Overflow Interrupt enable.

Bit 0 = **OVF** *Overflow Flag*
This bit is set by hardware and cleared by software reading the CSR register. It indicates the transition of the counter from FFh to the ARR value.
0: New transition not yet reached
1: Transition reached

COUNTER ACCESS REGISTER (CAR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

Bit 7:0 = **CA[7:0]** *Counter Access Data*

These bits can be set and cleared either by hardware or by software. The CAR register is used to read or write the auto-reload counter "on the fly" (while it is counting).

AUTO-RELOAD REGISTER (ARR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bit 7:0 = **AR[7:0]** *Counter Auto-Reload Data*

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register has two PWM management functions:

 – Adjusting the PWM frequency
 – Setting the PWM duty cycle resolution

PWM Frequency vs. Resolution:

ARR value	Resolution	f_{PWM}	
		Min	**Max**
0	8-bit	~0.244-KHz	31.25-KHz
[0..127]	> 7-bit	~0.244-KHz	62.5-KHz
[128..191]	> 6-bit	~0.488-KHz	125-KHz
[192..223]	> 5-bit	~0.977-KHz	250-KHz
[224..239]	> 4-bit	~1.953-KHz	500-KHz

PWM AUTO-RELOAD TIMER (Cont'd)

PWM CONTROL REGISTER (PWMCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0

Bit 7:4 = **OE[3:0]** *PWM Output Enable*
These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin.
0: PWM output disabled.
1: PWM output enabled.

Bit 3:0 = **OP[3:0]** *PWM Output Polarity*
These bits are set and cleared by software. They independently select the polarity of the four PWM output signals.

PWMx output level		OPx
Counter <= OCRx	**Counter > OCRx**	
1	0	0
0	1	1

Note: When an OPx bit is modified, the PWMx output signal polarity is immediately reversed.

DUTY CYCLE REGISTERS (DCRx)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

Bit 7:0 = **DC[7:0]** *Duty Cycle Data*

These bits are set and cleared by software.

A DCRx register is associated with the OCRx register of each PWM channel to determine the second edge location of the PWM signal (the first edge location is common to all channels and given by the ARR register). These DCR registers allow the duty cycle to be set independently for each PWM channel.

PWM AUTO-RELOAD TIMER (Cont'd)

**INPUT CAPTURE
CONTROL / STATUS REGISTER (ICCSR)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	CS2	CS1	CIE2	CIE1	CF2	CF1

Bit 7:6 = Reserved, always read as 0.

Bit 5:4 = **CS[2:1]** *Capture Sensitivity*
These bits are set and cleared by software. They determine the trigger event polarity on the corresponding input capture channel.
0: Falling edge triggers capture on channel x.
1: Rising edge triggers capture on channel x.

Bit 3:2 = **CIE[2:1]** *Capture Interrupt Enable*
These bits are set and cleared by software. They enable or disable the Input capture channel interrupts independently.
0: Input capture channel x interrupt disabled.
1: Input capture channel x interrupt enabled.

Bit 1:0 = **CF[2:1]** *Capture Flag*
These bits are set by hardware and cleared by software reading the corresponding ICRx register. Each CFx bit indicates that an input capture x has occurred.
0: No input capture on channel x.
1: An input capture has occured on channel x.

INPUT CAPTURE REGISTERS (ICRx)

Read only

Reset Value: 0000 0000 (00h)

7							0
IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

Bit 7:0 = **IC[7:0]** *Input Capture Data*

These read only bits are set and cleared by hardware. An ICRx register contains the 8-bit auto-reload counter value transferred by the input capture channel x event.

PWM AUTO-RELOAD TIMER (Cont'd)

Table 15. PWM Auto-Reload Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0073h	**PWMDCR3** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0074h	**PWMDCR2** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0075h	**PWMDCR1** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0076h	**PWMDCR0** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0077h	**PWMCR** Reset Value	OE3 0	OE2 0	OE1 0	OE0 0	OP3 0	OP2 0	OP1 0	OP0 0
0078h	**ARTCSR** Reset Value	EXCL 0	CC2 0	CC1 0	CC0 0	TCE 0	FCRL 0	RIE 0	OVF 0
0079h	**ARTCAR** Reset Value	CA7 0	CA6 0	CA5 0	CA4 0	CA3 0	CA2 0	CA1 0	CA0 0
007Ah	**ARTARR** Reset Value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0
007Bh	**ARTICCSR** Reset Value	0	0	CE2 0	CE1 0	CS2 0	CS1 0	CF2 0	CF1 0
007Ch	**ARTICR1** Reset Value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0
007Dh	**ARTICR2** Reset Value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0

10.4 16-BIT TIMER

10.4.1 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.

Some ST7 devices have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after a MCU reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In ST7 devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

10.4.2 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
- Reduced Power Mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 40.

*Note: Some timer pins may not available (not bonded) in some ST7 devices. Refer to the device pin out description.

When reading an input signal on a non-bonded pin, the value will always be '1'.

10.4.3 Functional Description

10.4.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high & low.

Counter Register (CR):
- Counter High Register (CHR) is the most significant byte (MS Byte).
- Counter Low Register (CLR) is the least significant byte (LS Byte).

Alternate Counter Register (ACR)
- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte).

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 (Timer overflow flag), located in the Status register, (SR), (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.
Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 16 Clock Control Bits. The value in the counter register repeats every 131.072, 262.144 or 524.288 CPU clock cycles depending on the CC[1:0] bits.
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

16-BIT TIMER (Cont'd)

Figure 40. Timer Block Diagram
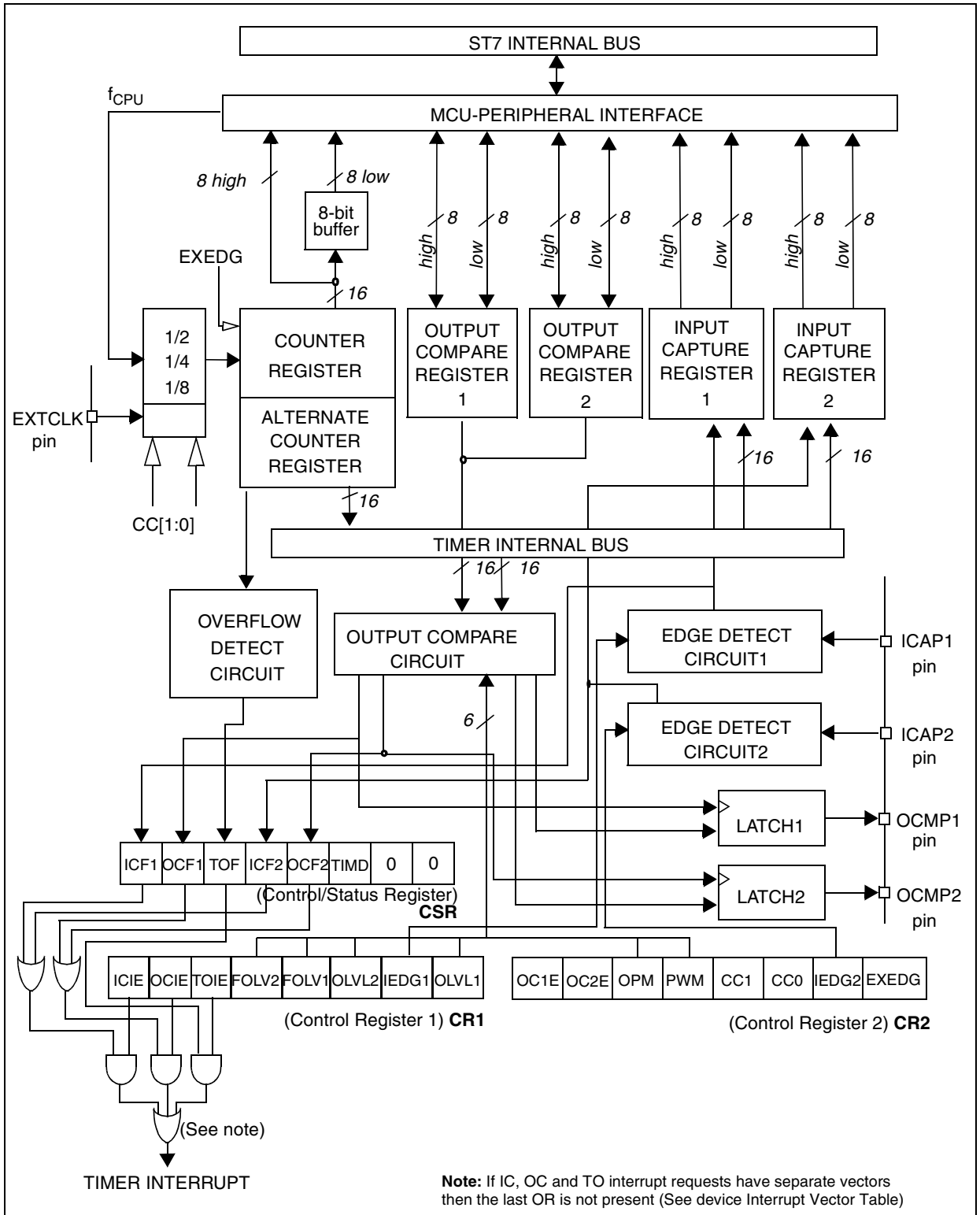
16-BIT TIMER (Cont'd)

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

Beginning of the sequence

At t0 | Read MS Byte | → | LS Byte is buffered |

Other instructions

At t0 +Δt | Read LS Byte | → | Returns the buffered LS Byte value at t0 |
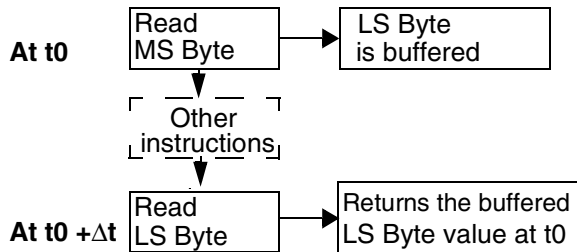
Sequence completed

The user must read the MS Byte first, then the LS Byte value is buffered automatically.

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

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS Byte 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. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading 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).

10.4.3.2 External Clock

The external clock (wthe CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

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

A minimum of 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 41. Counter Timing Diagram, internal clock divided by 2

CPU CLOCK

INTERNAL RESET

TIMER CLOCK

COUNTER REGISTER FFFD FFFE FFFF 0000 0001 0002 0003

TIMER OVERFLOW FLAG (TOF)
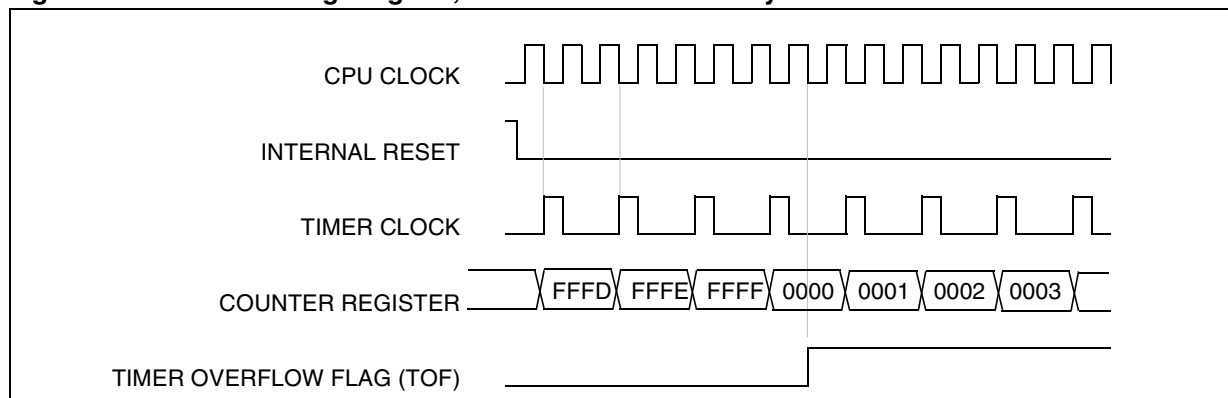
Figure 42. Counter Timing Diagram, internal clock divided by 4

CPU CLOCK

INTERNAL RESET

TIMER CLOCK

COUNTER REGISTER FFFC FFFD 0000 0001

TIMER OVERFLOW FLAG (TOF)
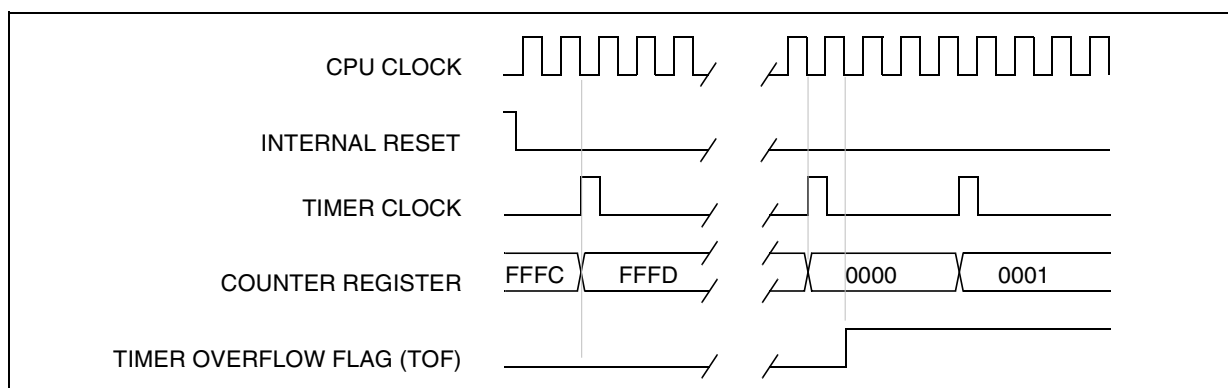
Figure 43. Counter Timing Diagram, internal clock divided by 8

CPU CLOCK

INTERNAL RESET

TIMER CLOCK

COUNTER REGISTER FFFC FFFD 0000

TIMER OVERFLOW FLAG (TOF)
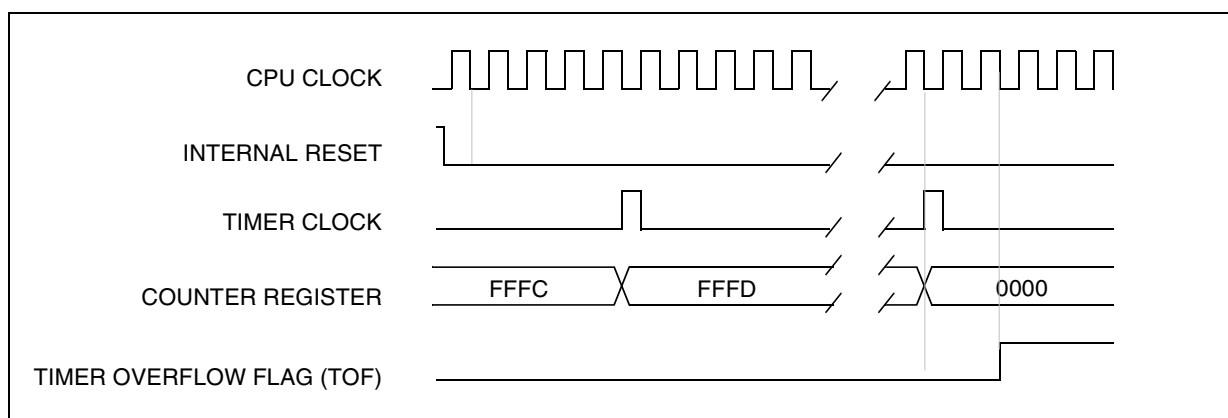
Note: The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.

16-BIT TIMER (Cont'd)

10.4.3.3 Input Capture

In this section, the index, *i*, may be 1 or 2 because there are 2 input capture functions in the 16-bit timer.

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

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR
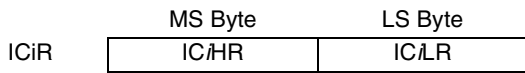
IC*i*R register is a read-only register.

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

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure:

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

– Select the timer clock (CC[1:0]) (see Table 16 Clock Control Bits).

– 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 either the ICAP1 pin or the ICAP2 pin

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin 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 45).

– 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 (i.e. clearing the ICF*i* bit) 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 and ICF*i* will never be set until the IC*i*LR register is also read.

2. The IC*i*R register 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 2 output compare functions.

4. In One pulse Mode and PWM mode only 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 activates the input capture function.
   Moreover if one of the ICAP*i* pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.
   This can be avoided if the input capture function *i* is disabled by reading the IC*i*HR (see note 1).

6. The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFFFh).

16-BIT TIMER (Cont'd)

Figure 44. Input Capture Block Diagram
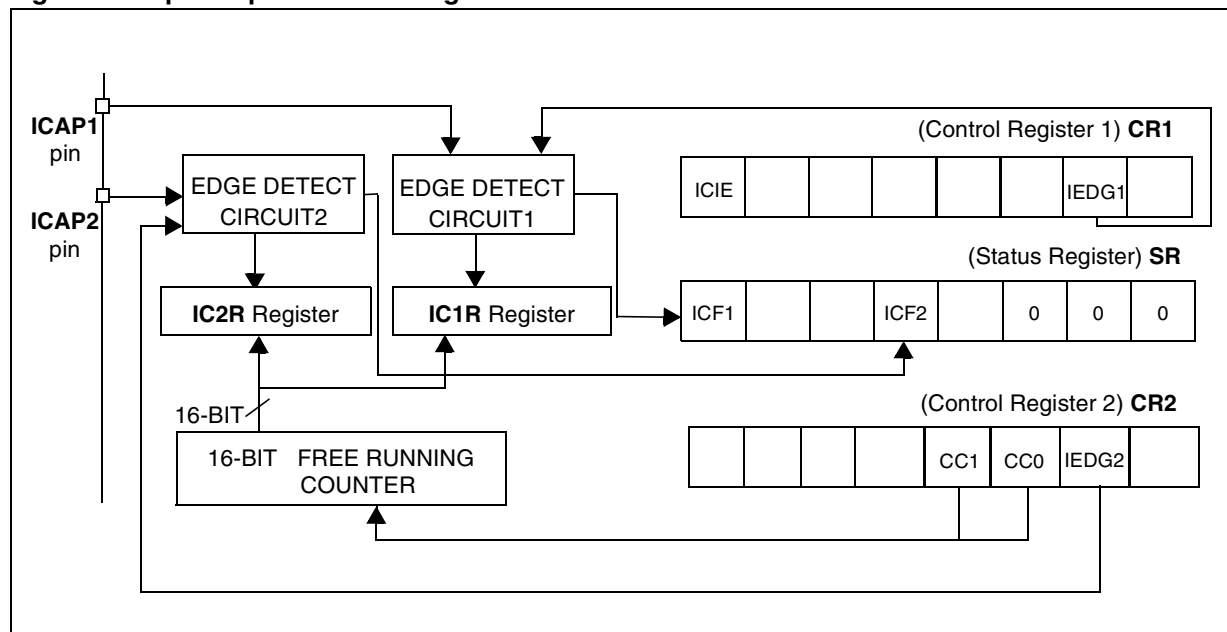
Figure 45. Input Capture Timing Diagram

Note: The rising edge is the active edge.
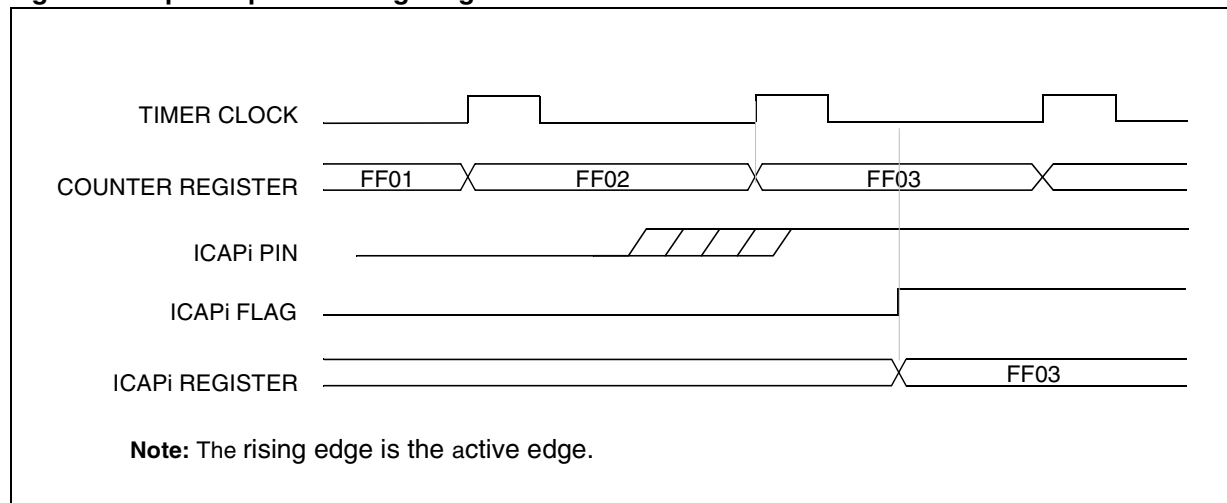
16-BIT TIMER (Cont'd)

10.4.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2 because there are 2 output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate 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 counter register 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: $(f_{CPU/CC[1:0]})$.

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* signal.

– Select the timer clock (CC[1:0]) (see Table 16 Clock Control Bits).

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 between OCRi register and CR register:

– OCF*i* bit is set.

– The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset).

– 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:

$$\Delta\ OC_iR = \frac{\Delta t * f_{CPU}}{PRESC}$$

Where:

Δt = Output compare period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 16 Clock Control Bits)

If the timer clock is an external clock, the formula is:

$$\Delta\ OC_iR = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (i.e. clearing the OCF*i* bit) 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).

16-BIT TIMER (Cont'd)

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 timer clock is $f_{CPU}/2$, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value (see Figure 47 on page 72). This behaviour is the same in OPM or PWM mode.
   When the timer clock is $f_{CPU}/4$, $f_{CPU}/8$ 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 48 on page 72).

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.

Forced Compare Output capability

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.

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

Figure 46. Output Compare Block Diagram
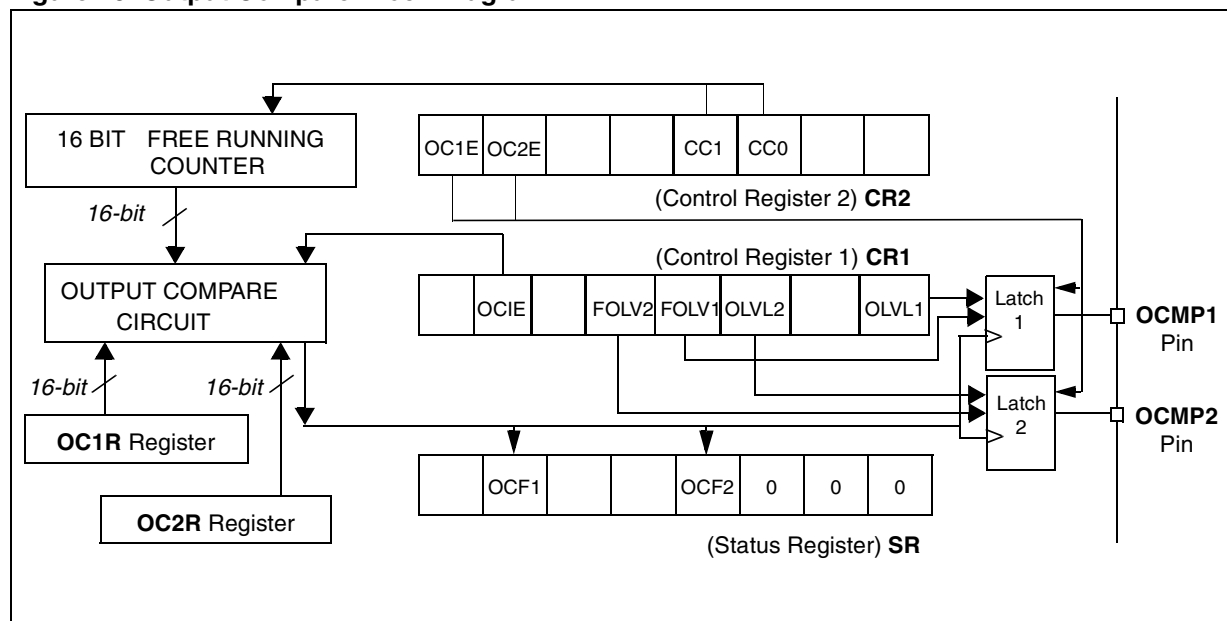
16-BIT TIMER (Cont'd)

Figure 47. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/2$

INTERNAL CPU CLOCK

TIMER CLOCK

COUNTER REGISTER   2ECF 2ED0   2ED1 2ED2   2ED3   2ED4

OUTPUT COMPARE REGISTER *i* (OCR*i*)   2ED3

OUTPUT COMPARE FLAG *i* (OCF*i*)

OCMP*i* PIN (OLVL*i*=1)
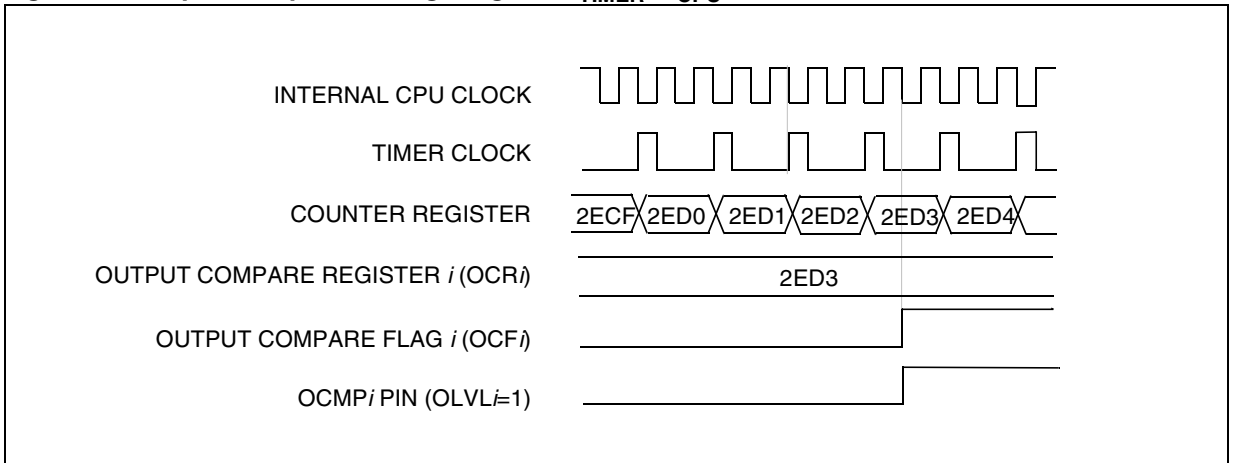
Figure 48. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/4$

INTERNAL CPU CLOCK

TIMER CLOCK

COUNTER REGISTER   2ECF 2ED0   2ED1 2ED2   2ED3   2ED4

OUTPUT COMPARE REGISTER *i* (OCR*i*)   2ED3

COMPARE REGISTER *i* LATCH

OUTPUT COMPARE FLAG *i* (OCF*i*)

OCMP*i* PIN (OLVL*i*=1)
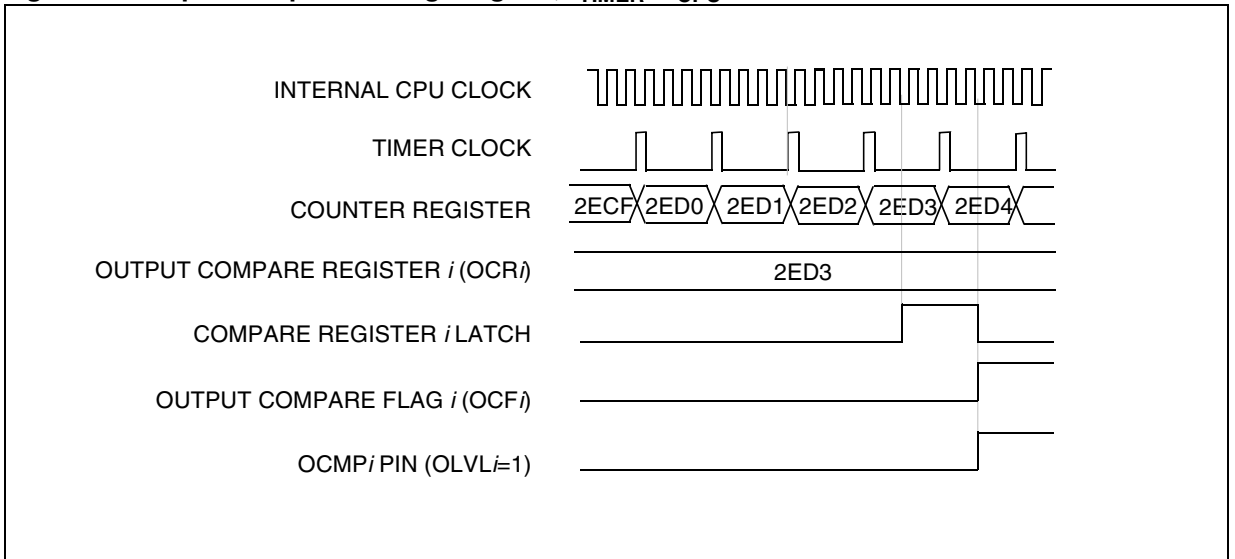
16-BIT TIMER (Cont'd)

10.4.3.5 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 the opposite column).

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 CC[1:0] (see Table 16 Clock Control Bits).

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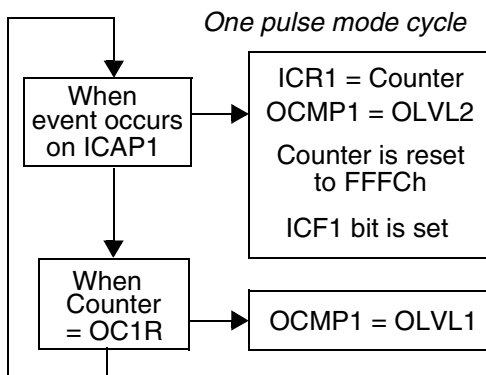
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (i.e. clearing the ICF*i* bit) 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.

The OC1R register value required for a specific timing application can be calculated using the following formula:

$$OCiR \text{ Value} = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits, see Table 16 Clock Control Bits)

If the timer clock is an external clock the formula is:

$$OCiR = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

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 49).

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. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

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

4. 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.

5. When 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.

16-BIT TIMER (Cont'd)

Figure 49. One Pulse Mode Timing Example

Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1
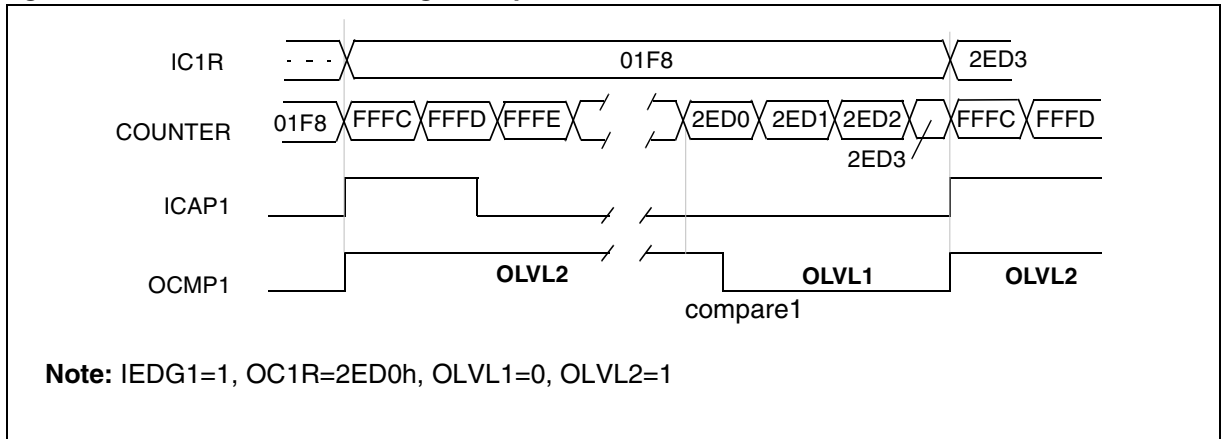
Figure 50. Pulse Width Modulation Mode Timing Example

Note: OC1R=2ED0h, OC2R=34E2, OLVL1=0, OLVL2= 1
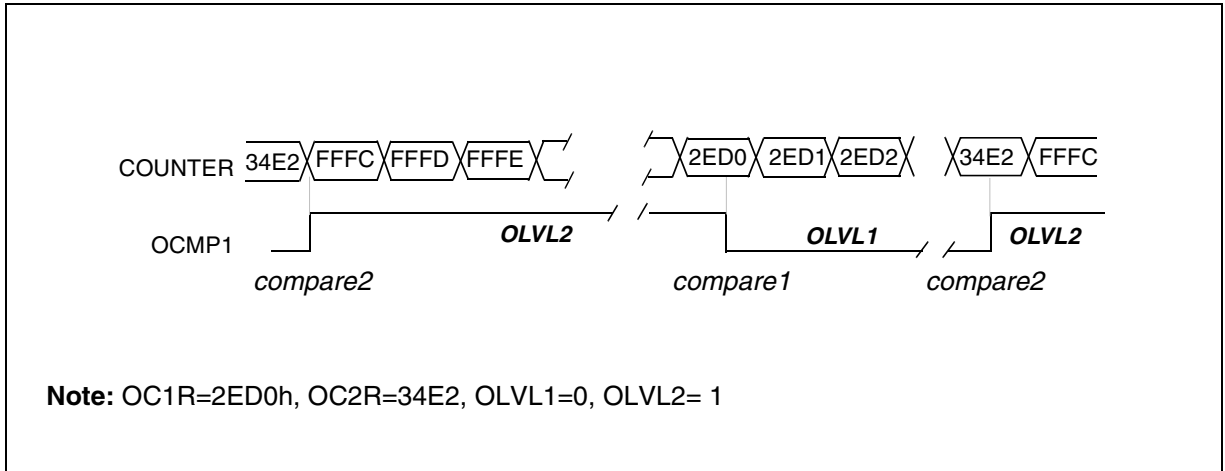
16-BIT TIMER (Cont'd)

10.4.3.6 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.

Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are taken into account only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1).

Procedure

To use pulse width modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula in the opposite column.

2. Load the OC1R register with the value corresponding to the period of the pulse if (OLVL1=0 and OLVL2=1) using the formula in the opposite column.

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 the OC1R register.
   – Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the 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 (CC[1:0]) (see Table 16 Clock Control Bits).

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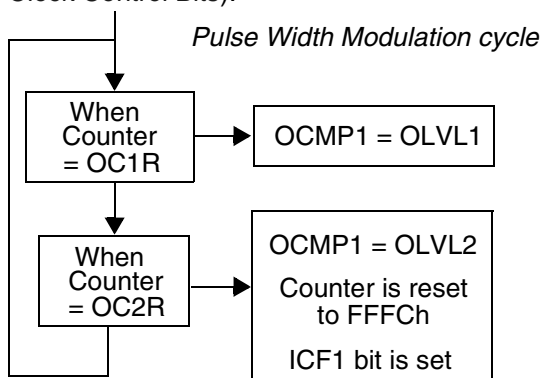
If OLVL1=1 and OLVL2=0 the length of the positive pulse is the difference between the OC2R and OC1R registers.

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

The OCiR 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 = Signal or pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 16 Clock Control Bits)

If the timer clock is an external clock the formula is:

$$\text{OC}i\text{R} = t * f_{EXT} - 5$$

Where:

t = Signal or pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 50)

Notes:

1. After a write instruction to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

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)

10.4.4 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.

10.4.5 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 interrupt mask in the CC register is reset (RIM instruction).

10.4.6 Summary of Timer modes

MODES	AVAILABLE RESOURCES			
	Input Capture 1	**Input Capture 2**	**Output Compare 1**	**Output Compare 2**
Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)	Yes	Yes	Yes	Yes
One Pulse Mode	No	Not Recommended[1]	No	Partially [2]
PWM Mode	No	Not Recommended[3]	No	No

1) See note 4 in Section 10.4.3.5 One Pulse Mode

1) See note 5 in Section 10.4.3.5 One Pulse Mode

1) See note 4 in Section 10.4.3.6 Pulse Width Modulation Mode

16-BIT TIMER (Cont'd)

10.4.7 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 Pin 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 = **CC[1:0]** *Clock Control.*

The timer clock mode depends on these bits:

Table 16. 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

Note: If the external clock pin is not available, programming the external clock configuration stops the counter.

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 counter register.
0: A falling edge triggers the counter register.
1: A rising edge triggers the counter register.

16-BIT TIMER (Cont'd)

CONTROL/STATUS REGISTER (CSR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	0	0

Bit 7 = **ICF1** *Input Capture Flag 1.*
0: No input capture (reset value).
1: An input capture has occurred on the ICAP1 pin 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 Flag.*
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 on the ICAP2 pin. 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 = **TIMD** *Timer disable.*
This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed while it is disabled.
0: Timer enabled
1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared.

16-BIT TIMER (Cont'd)

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
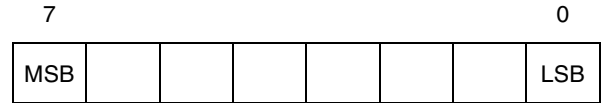
OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

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
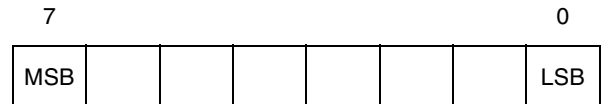
16-BIT TIMER (Cont'd)

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
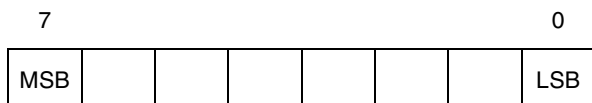
OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

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
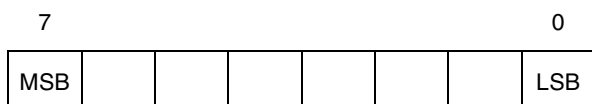
COUNTER LOW REGISTER (CLR)

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 CSR 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
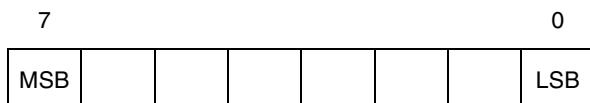
ALTERNATE COUNTER LOW REGISTER (ACLR)

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 CSR register does not clear the TOF bit in the CSR register.

7							0
MSB							LSB
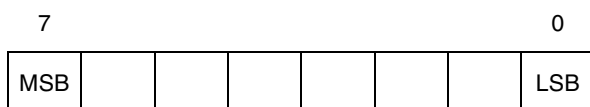
INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

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).
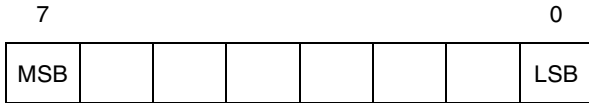
7							0
MSB							LSB
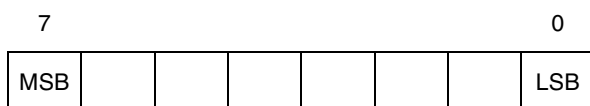
INPUT CAPTURE 2 LOW REGISTER (IC2LR)

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 17. 16-Bit Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Timer A: 32 Timer B: 42	**CR1** Reset Value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer A: 31 Timer B: 41	**CR2** Reset Value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer A: 33 Timer B: 43	**CSR** Reset Value	ICF1 0	OCF1 0	TOF 0	ICF2 0	OCF2 0	TIMD 0	- 0	- 0
Timer A: 34 Timer B: 44	**ICHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 35 Timer B: 45	**ICLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 36 Timer B: 46	**OCHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 37 Timer B: 47	**OCLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3E Timer B: 4E	**OCHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3F Timer B: 4F	**OCLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 38 Timer B: 48	**CHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 39 Timer B: 49	**CLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3A Timer B: 4A	**ACHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 3B Timer B: 4B	**ACLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3C Timer B: 4C	**ICHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3D Timer B: 4D	**ICLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -

10.5 SERIAL PERIPHERAL INTERFACE (SPI)

10.5.1 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.

10.5.2 Main Features

- Full duplex, three-wire synchronous transfers
- Simplex, two wire synchronous transfers
- Master or slave operation selectable either through software or hardware
- Six master mode frequencies
- Maximum slave mode frequency = $f_{CPU}/2$.
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability.
- Overrun error detection flag.

10.5.3 General Description

The SPI is connected to external devices through 4(3) alternate pins:

- MISO: Master In / Slave Out pin
- MOSI: Master Out / Slave In pin
- SCK: Serial Clock pin
- $\overline{SS}$: Slave select pin (if not done through software)

A basic example of interconnections between a single master and a single slave is illustrated on Figure 51.

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 54) but master and slave must be programmed with the same timing mode.

Figure 51. Serial Peripheral Interface Master/Slave
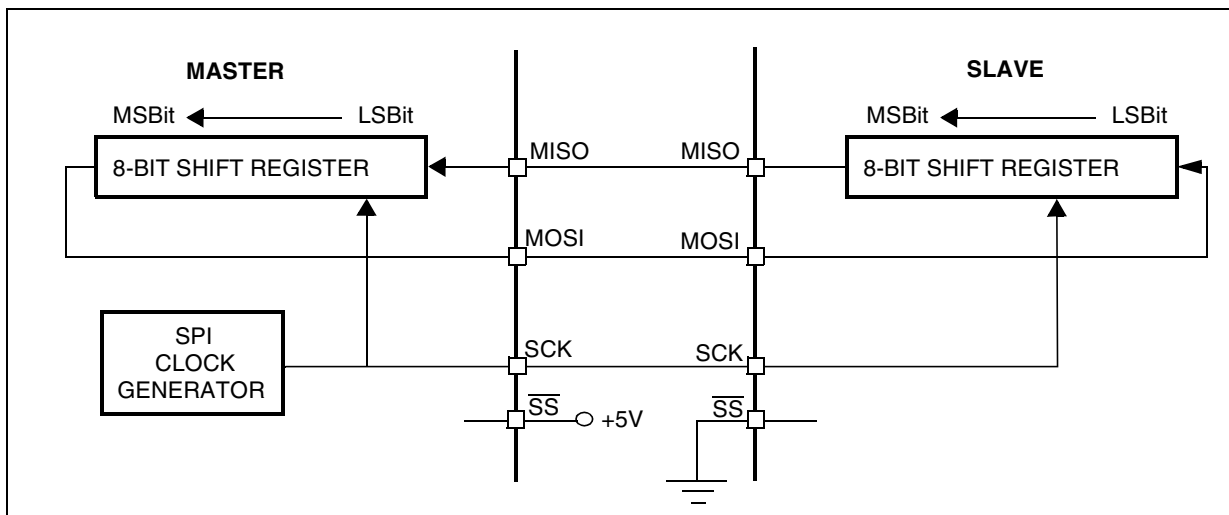
SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 52. Serial Peripheral Interface Block Diagram
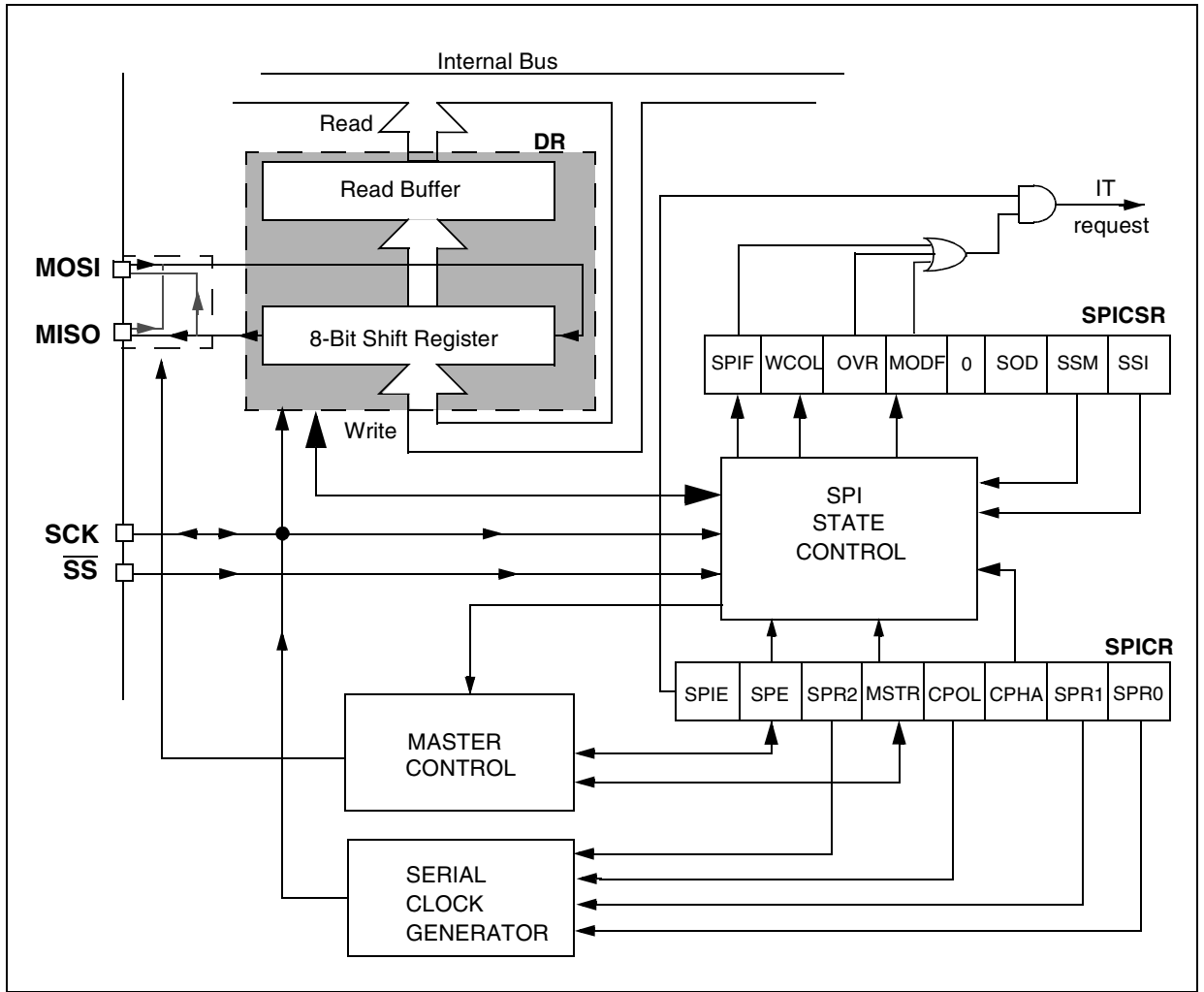
SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4 Functional Description

Figure 51 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

– A Control Register (SPICR)

– A Control Status Register (SPICSR)

– A Data Register (SPIDR)

Refer to the SPICR, SPICSR and SPIDR registers in Section 10.5.5for the bit definitions.

10.5.4.1 $\overline{SS}$ Signal in Hardware/Software Mode

The $\overline{SS}$ signal can obtained in two modes:

– Hardware mode (through the $\overline{SS}$ pin)

– Software mode (through the SSI bit in the SPIC-SR register)

The mode (hardware or software) is selected by the Slave Selection Mode (SSM) bit in the SPICSR register.

Note: In this document, wherever $\overline{SS}$ signal selection is done using the $\overline{SS}$ pin (hardware mode), this can also be done in software mode, using the SSM and SSI bits.

10.5.4.2 Master Configuration

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

Procedure

1. Select the SPR[2:0] bits to define the serial clock baud rate (see SPICR register).

2. Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 54).

3. Connect the $\overline{SS}$ pin to a high level signal during the complete byte transmit sequence or, in software mode, set the SSI bit in the SPICSR register.

4. 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 in 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 SPIDR 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 SPICSR register while the SPIF bit is set

2. A read to the SPIDR register.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4.3 Slave Configuration

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

The value of the SPR[2:0] bits is not used for the data transfer.

Procedure

1. For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 54.

2. The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence or, in software mode, clear the SSI bit in the SPICSR register.

3. 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 SPIDR 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 SPICSR 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 SPIDR register are inhibited until the SPICSR 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 10.5.4.8).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the SPIDR register between each data byte transfer to avoid a write collision (see Section 10.5.4.6).

Slave in Halt mode

In slave configuration, the SPI is able to exit the ST7 device from HALT mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from Halt mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from Halt mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the ST7 from Halt mode only if the Slave Select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the ST7 enters Halt mode. So if Slave selection is configured as external (see Section 10.5.4.1), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters Halt mode.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4.4 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 54, 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 53).

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 53).

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 SPIDR 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 SPIDR without producing a write collision.

Note: The SPI must be disabled by resetting the SPE bit if the CPOL bit is changed at the communication byte boundaries.

10.5.4.5 Output Disable

In order to free the I/O pin so it can be used for other purposes, it is possible to disable the SPI output function by setting the SOD bit in the SPICSR register.

Figure 53. CPHA / $\overline{SS}$ Timing Diagram
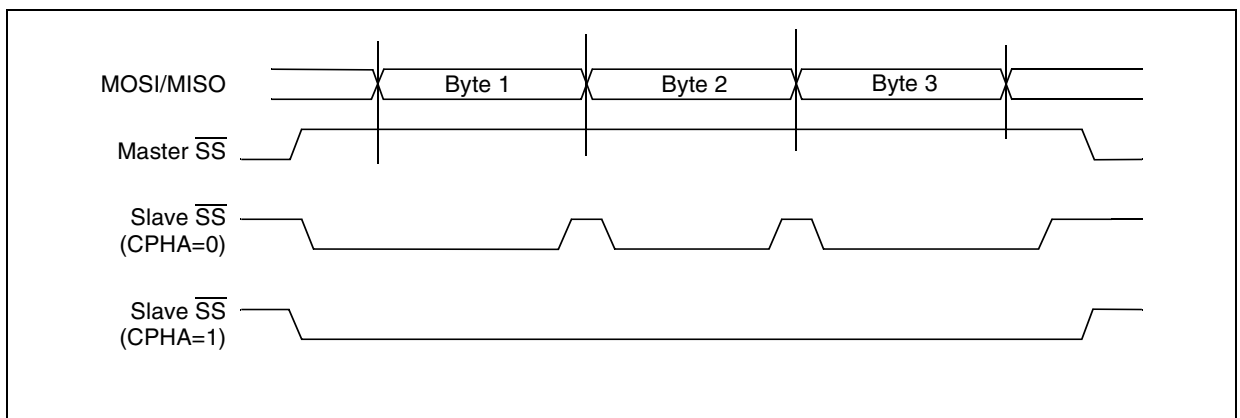
SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 54. Data Clock Timing Diagram

Note: This figure should not be used as a replacement for parametric information.
Refer to the Electrical Characteristics chapter.
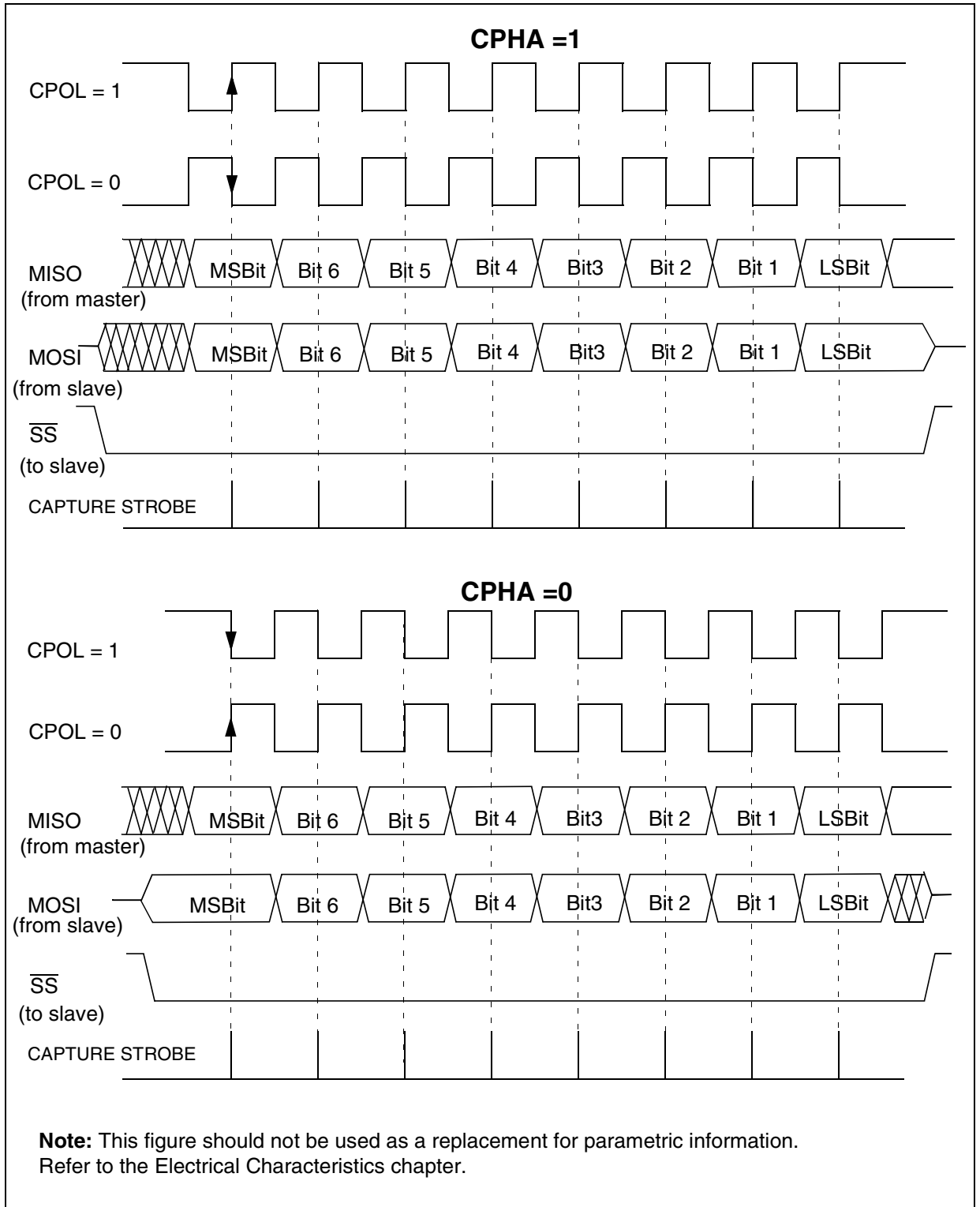
SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4.6 Write Collision Error

A write collision occurs when the software tries to write to the SPIDR 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 SPIDR 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 SPIDR 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 SPIDR 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 SPICSR 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 55).

Figure 55. Clearing the WCOL bit (Write Collision Flag) Software Sequence
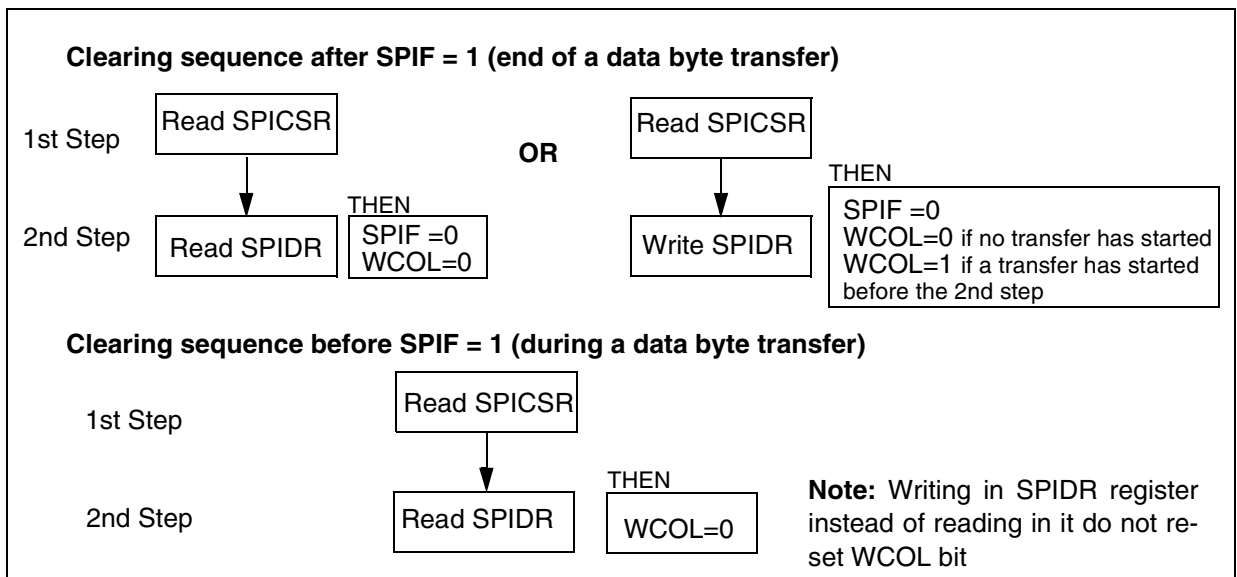
SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4.7 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.

10.5.4.8 Overrun Condition

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previous data byte transmitted, then the OVR bit is set and an interrupt is generated if the SPIE bit is set.

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

The OVR bit is cleared just after it has been read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.4.9 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 56).

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 active 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 SPICR register and the MODF bit in the SPICSR register.

Figure 56. Single Master Configuration
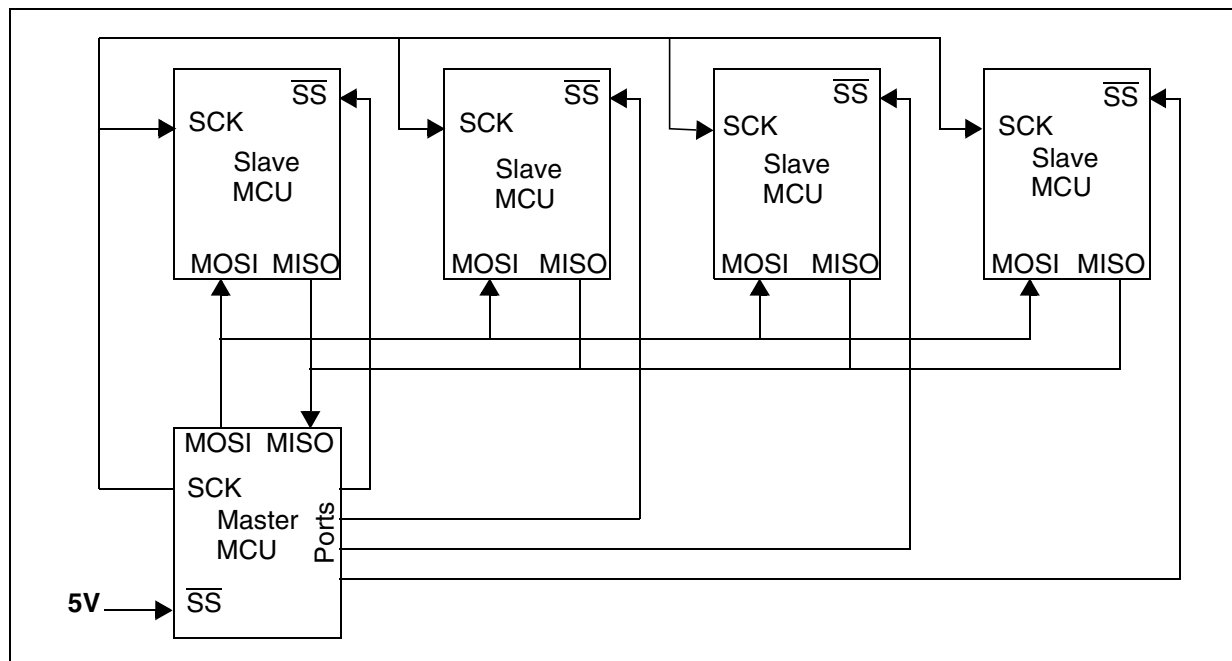
SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.5 Register Description

CONTROL REGISTER (SPICR)

Read/Write

Reset Value: 0000 xxxx (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 10.5.4.7 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 18 Serial Peripheral Baud Rate.
0: Divider by 2 enabled
1: Divider by 2 disabled

Note: This bit has no effect in slave mode.

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 10.5.4.7 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.

Note: The SPI must be disabled by resetting SPE bit if CPOL is changed at the communication byte boundaries.

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.

Bits 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 18. Serial Peripheral Baud Rate

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	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)

CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only)

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only).*
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 (Read only).*
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 55).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** S*PI Overrun error (Read only).*
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 10.5.4.8). An interrupt is generated if SPIE = 1 in SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only).*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 10.5.4.7 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF=1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable.*
This bit is set and cleared by software. When set it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output not disabled
1: SPI output disabled

Bit 1 = **SSM** $\overline{SS}$ *Mode Selection.*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI Slave Select pin and use the SSI bit value instead.
0: Hardware mode ($\overline{SS}$ pin is used for Master/Slave Selection).
1: Software mode (SSI bit is used for Master/Slave Selection). $\overline{SS}$ is free to be used as a standard I/O pin

Note: Master mode selection also depends on the value of the MSTR bit in the SPICR register

Bit 0 = **SSI** $\overline{SS}$ *Internal Mode.*
This bit is set and cleared by software. It replaces pin $\overline{SS}$ of the SPI when bit SSM is set to 1. SSI bit is active low slave select signal when SSM is set to 1.
0 : Slave selected
1 : Slave not selected

DATA I/O REGISTER (SPIDR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR 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 SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the contents of the shift register (see Figure 52).

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.5.6 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. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wake-up event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

10.5.7 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF		Yes	Yes
Master Mode Fault Event	MODF	SPIE	Yes	No
Overrun Error	OVR		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 interrupt mask in the CC register is reset (RIM instruction).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 19. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0021h	**SPIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0022h	**SPICR** Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	**SPICSR** Reset Value	SPIF 0	WCOL 0	OR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

10.6 SERIAL COMMUNICATIONS INTERFACE (SCI)

10.6.1 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.

10.6.2 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
■ Four error detection flags:
  – Overrun error
  – Noise error
  – Frame error
  – Parity error
■ Five interrupt sources with flags:
  – Transmit data register empty
  – Transmission complete
  – Receive data register full
  – Idle line received
  – Overrun error detected
■ Parity control:
  – Transmits parity bit
  – Checks parity of received data byte
■ Reduced power consumption mode

10.6.3 General Description

The interface is externally connected to another device by two pins (see Figure 58):

– 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 these 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 57. SCI Block Diagram
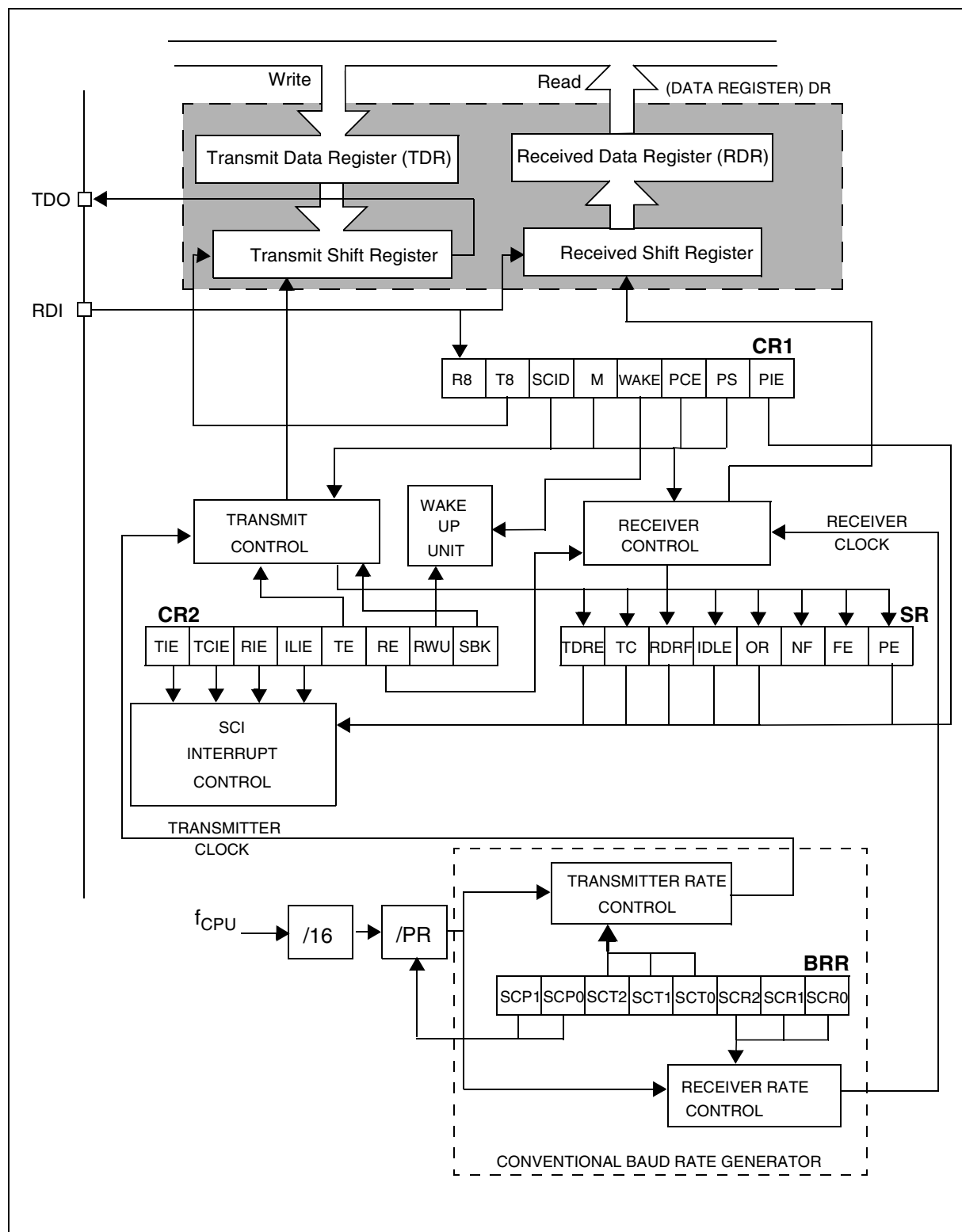
SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 57. It contains 6 dedicated registers:

– Two control registers (SCICR1 & SCICR2)

– A status register (SCISR)

– A baud rate register (SCIBRR)

– An extended prescaler receiver register (SCIERPR)

– An extended prescaler transmitter register (SCI-ETPR)

Refer to the register descriptions in Section 10.6.7for the definitions of each bit.

10.6.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 57).

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 58. Word Length Programming
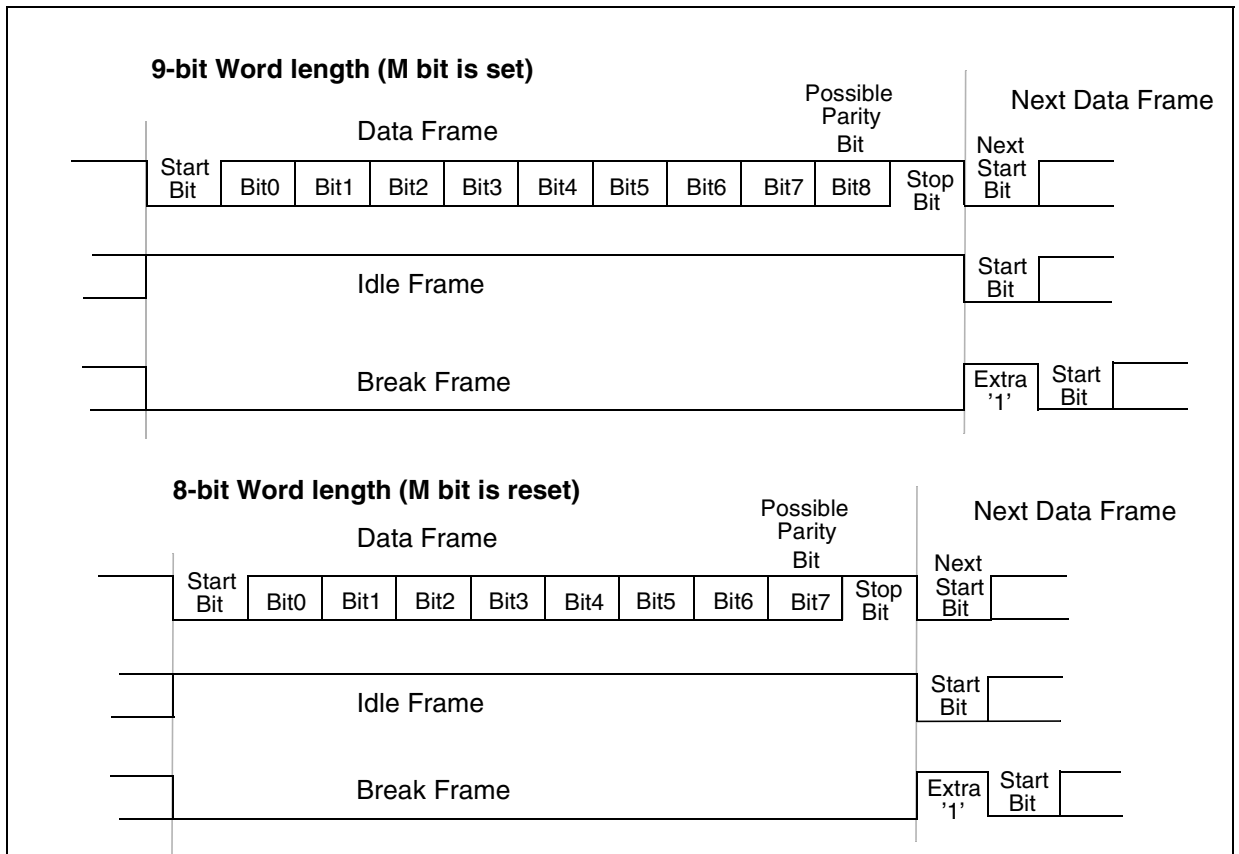
SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4.2 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 SCICR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 57).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the SCIBRR and the SCIETPR 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 SCISR register and write the data to send in the SCIDR 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 SCISR register
2. A write to the SCIDR 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 SCIDR 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 SCIDR 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 SCIDR 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 SCISR register
2. A write to the SCIDR 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 58).

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 SCIDR.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4.3 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 SCICR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the SCIDR register consists or a buffer (RDR) between the internal bus and the received shift register (see Figure 57).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the SCIBRR and the SCIERPR 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 SCISR register

2. A read to the SCIDR 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 RDR 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 SCISR register followed by a SCIDR 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 SCIDR 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 SCISR register read operation followed by a SCIDR 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 SCIDR 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 SCISR register read operation followed by a SCIDR register read operation.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 59. SCI Baud Rate and Extended Prescaler Block Diagram
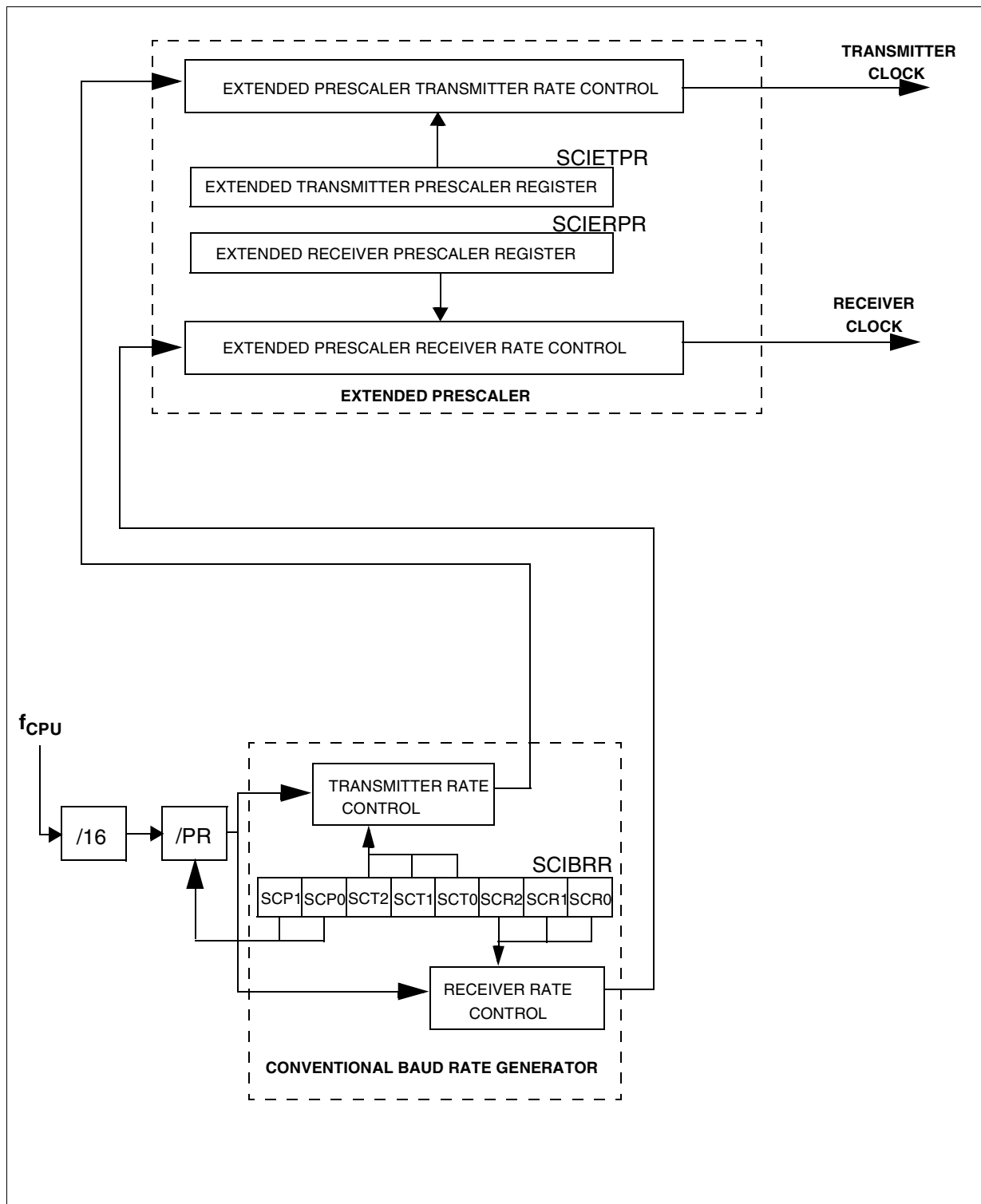
SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4.4 Conventional Baud Rate Generation

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

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

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

(see SCT[2:0] bits)

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

(see SCR[2:0] bits)

All these bits are in the SCIBRR 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 38400 baud.

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

10.6.4.5 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 59.

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 SCIERPR or the SCIETPR register.

Note: the extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value

other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16*ERPR*(PR*RR)}$$

with:

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

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

10.6.4.6 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 interrupts 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)

10.6.4.7 Parity Control

Parity control (generation of parity bit in trasmission and and parity chencking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 20.

Table 20. Frame Formats

M bit	PCE bit	SCI frame
0	0	I SB I 8 bit data I STB I
0	1	I SB I 7-bit data I PB I STB I
1	0	I SB I 9-bit data I STB I
1	1	I SB I 8-bit data PB I STB I

Legend: SB = Start Bit, STB = Stop Bit,
PB = Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: the parity bit is calculated to obtain an even number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 0 if even parity is selected (PS bit = 0).

Odd parity: the parity bit is calculated to obtain an odd number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of "1s" if even parity is selected

(PS=0) or an odd number of "1s" if odd parity is selected (PS=1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.6.5 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.

10.6.6 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
Parity Error	PE	PIE	Yes	No

The SCI interrupt events are connected to the same interrupt vector.

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.7 Register Description

STATUS REGISTER (SCISR)
Read Only
Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE

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 bit=1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR 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 unless the TDRE bit is cleared.

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 SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR 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 to the SCIDR register. An interrupt is generated if RIE=1 in the SCICR2 register. It is cleared by hardware when RE=0 or by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 SCICR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 oc-

curs). 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 SCICR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 SCISR register followed by a read to the SCIDR 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 SCISR register followed by a read to the SCIDR 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 = **PE** *Parity error.*
This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE=1 in the SCICR1 register.
0: No parity error
1: Parity error

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 1 (SCICR1)

Read/Write

Reset Value: x000 0000 (x0h)

7							0
R8	T8	SCID	M	WAKE	PCE	PS	PIE

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 5 = **SCID** *Disabled for low power consumption*
When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption.This bit is set and cleared by software.
0: SCI enabled
1: SCI prescaler and outputs disabled

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

Note: The M bit must not be modified during a data transfer (both transmission and reception).

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

Bit 2 = **PCE** *Parity control enable.*
This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M=1; 8th bit if M=0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission).
0: Parity control disabled
1: Parity control enabled

Bit 1 = **PS** *Parity selection.*
This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte.
0: Even parity
1: Odd parity

Bit 0 = **PIE** *Parity interrupt enable.*
This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software.
0: Parity error interrupt disabled
1: Parity error interrupt enabled.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 2 (SCICR2)

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 SCISR 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 SCISR 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 SCISR 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 SCISR 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 SCISR 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

Note: Before selecting Mute mode (setting the RWU bit), the SCI must receive some data first, otherwise it cannot function in Mute mode with wakeup by idle line detection.

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 (SCIDR)

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 57).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 57).

BAUD RATE REGISTER (SCIBRR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bits 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

Bits 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

Bits 2:0 = **SCR[2:0]** *SCI Receiver rate divisor.*
These 3 bits, in conjunction with the SCP[1:0] 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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

EXTENDED RECEIVE PRESCALER DIVISION REGISTER (SCIERPR)

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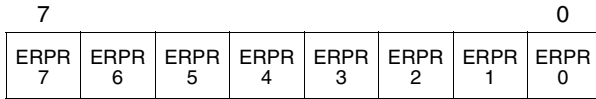
Bits 7:0 = **ERPR[7:0]** *8-bit Extended Receive Prescaler Register.*

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 59) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255).

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

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (SCIETPR)

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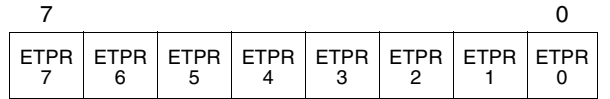
Bits 7:0 = **ETPR[7:0]** *8-bit Extended Transmit Prescaler Register.*

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 59) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255).

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

SERIAL COMMUNICATION INTERFACE (Cont'd)

Table 21. SCI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0050h	**SCISR** Reset Value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0051h	**SCIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0052h	**SCIBRR** Reset Value	SCP1 0	SCP0 0	SCT2 x	SCT1 x	SCT0 x	SCR2 x	SCR1 x	SCR0 x
0053h	**SCICR1** Reset Value	R8 x	T8 x	SCID 0	M x	WAKE x	PCE 0	PS 0	PIE 0
0054h	**SCICR2** Reset Value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	**SCIERPR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	**SCIPETPR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

10.7 I²C BUS INTERFACE (I2C)

10.7.1 Introduction

The I²C Bus Interface serves as an interface between the microcontroller and the serial I²C bus. It provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It supports fast I²C mode (400kHz).

10.7.2 Main Features

■ Parallel-bus/I²C protocol converter
■ Multi-master capability
■ 7-bit/10-bit Addressing
■ Transmitter/Receiver flag
■ End-of-byte transmission flag
■ Transfer problem detection

I²C Master Features:

■ Clock generation
■ I²C bus busy flag
■ Arbitration Lost Flag
■ End of byte transmission flag
■ Transmitter/Receiver Flag
■ Start bit detection flag
■ Start and Stop generation

I²C Slave Features:

■ Stop bit detection
■ I²C bus busy flag
■ Detection of misplaced start or stop condition
■ Programmable I²C Address detection
■ Transfer problem detection
■ End-of-byte transmission flag
■ Transmitter/Receiver flag

10.7.3 General Description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

Mode Selection

The interface can operate in the four following modes:

– Slave transmitter/receiver

– Master transmitter/receiver

By default, it operates in slave mode.

The interface automatically switches from slave to master after it generates a START condition and from master to slave in case of arbitration loss or a STOP generation, allowing then Multi-Master capability.

Communication Flow

In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognising its own address (7 or 10-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to Figure 60.

Figure 60. I²C BUS Protocol
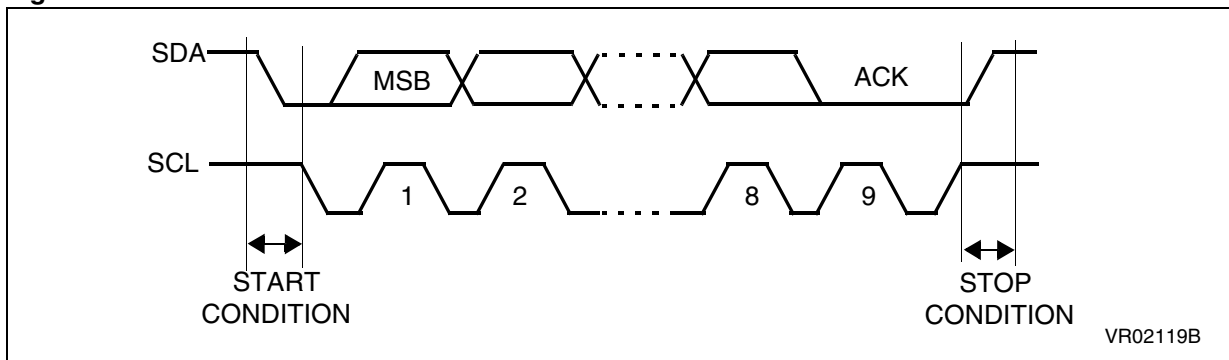
I²C BUS INTERFACE (Cont'd)

Acknowledge may be enabled and disabled by software.

The I²C interface address and/or general call address can be selected by software.

The speed of the I²C interface may be selected between Standard (0-100KHz) and Fast I²C (100-400KHz).

SDA/SCL Line Control

Transmitter mode: the interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the Data Register.

Receiver mode: the interface holds the clock line low after reception to wait for the microcontroller to read the byte in the Data Register.

The SCL frequency (F_{scl}) is controlled by a programmable clock divider which depends on the I²C bus mode.

When the I²C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

When the I²C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 61. I²C Interface Block Diagram
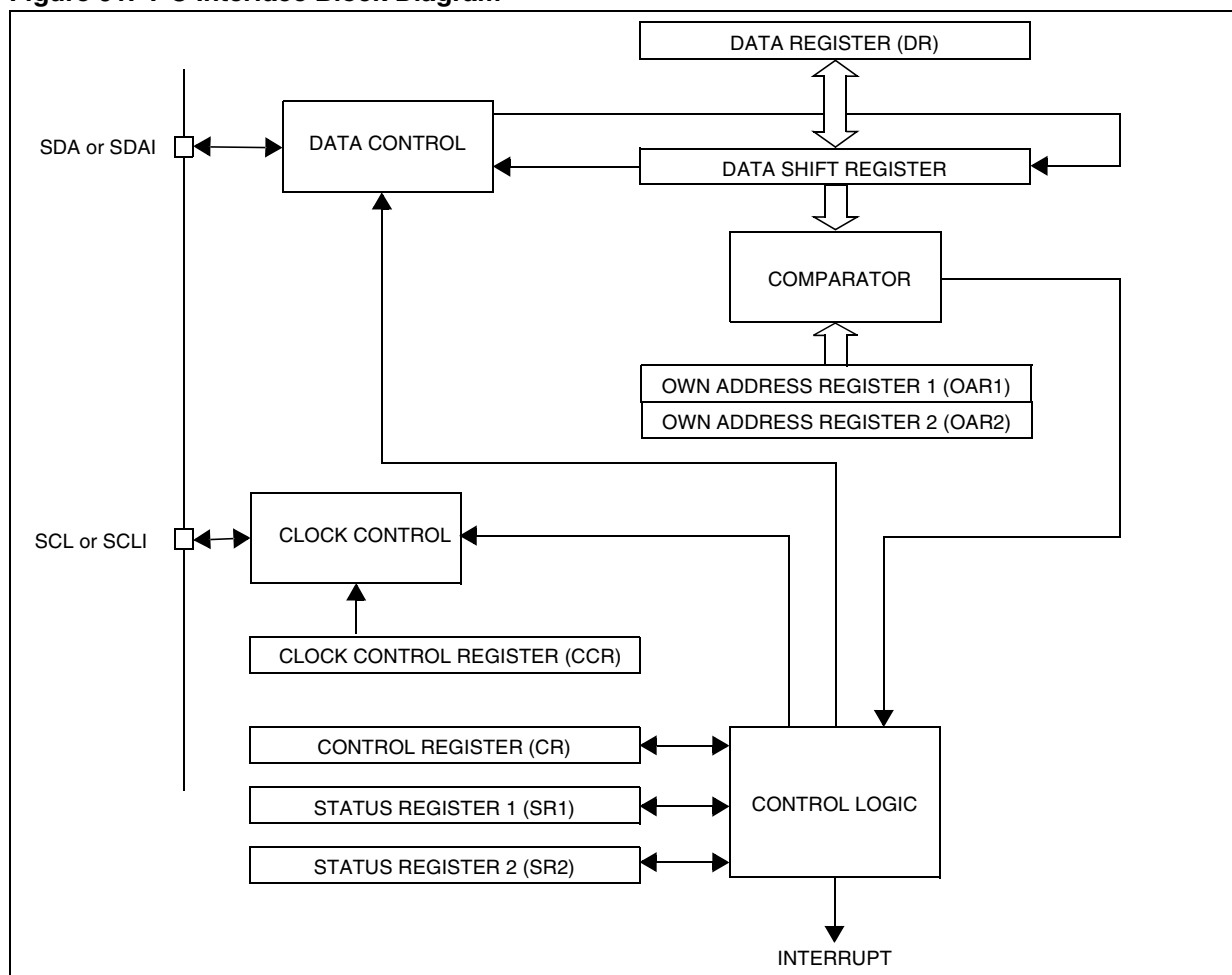
I²C BUS INTERFACE (Cont'd)

10.7.4 Functional Description

Refer to the CR, SR1 and SR2 registers in Section 10.7.7. for the bit definitions.

By default the I²C interface operates in Slave mode (M/SL bit is cleared) except when it initiates a transmit or receive sequence.

First the interface frequency must be configured using the FRi bits in the OAR2 register.

10.7.4.1 Slave Mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the address of the interface or the General Call address (if selected by software).

Note: In 10-bit addressing mode, the comparision includes the header sequence (11110xx0) and the two most significant bits of the address.

Header matched (10-bit mode only): the interface generates an acknowledge pulse if the ACK bit is set.

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:

– Acknowledge pulse if the ACK bit is set.

– EVF and ADSL bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV1).
Next, in 7-bit mode read the DR register to determine from the least significant bit (Data Direction Bit) if the slave must enter Receiver or Transmitter mode.

In 10-bit mode, after receiving the address sequence the slave is always in receive mode. It will enter transmit mode on receiving a repeated Start condition followed by the header sequence with matching address bits and the least significant bit set (11110xx1) .

Slave Receiver

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse if the ACK bit is set

– EVF and BTF bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV2).

Slave Transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV3).

When the acknowledge pulse is received:

– The EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Closing slave communication

After the last data byte is transferred a Stop Condition is generated by the master. The interface detects this condition and sets:

– EVF and STOPF bits with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see Figure 62 Transfer sequencing EV4).

Error Cases

– **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set with an interrupt if the ITE bit is set.
  If it is a Stop then the interface discards the data, released the lines and waits for another Start condition.
  If it is a Start then the interface discards the data and waits for the next slave address on the bus.

– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set with an interrupt if the ITE bit is set.

Note: In both cases, SCL line is not held low; however, SDA line can remain low due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

I^{2}C BUS INTERFACE (Cont'd)

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

10.7.4.2 Master Mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent:

– The EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address, **holding the SCL line low** (see Figure 62 Transfer sequencing EV5).

Slave address transmission

Then the slave address is sent to the SDA line via the internal shift register.

In 7-bit addressing mode, one address byte is sent.

In 10-bit addressing mode, sending the first byte including the header sequence causes the following event:
– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.
Then the master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV9).
Then the second address byte is sent by the interface.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see Figure 62 Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

Note: In 10-bit addressing mode, to switch the master to Receiver mode, software must generate a repeated Start condition and resend the header sequence with the least significant bit set (11110xx1).

Master Receiver

Following the address transmission and after SR1 and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse if if the ACK bit is set

– EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

I²C BUS INTERFACE (Cont'd)

Master Transmitter

Following the address transmission and after SR1 register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 62 Transfer sequencing EV8).

When the acknowledge bit is received, the interface sets:

– EVF and BTF bits with an interrupt if the ITE bit is set.

To close the communication: after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Error Cases

– **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.

– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the START or STOP bit.

– **ARLO:** Detection of an arbitration lost condition. In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: In all these cases, the SCL line is not held low; however, the SDA line can remain low due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

I²C BUS INTERFACE (Cont'd)

Figure 62. Transfer Sequencing
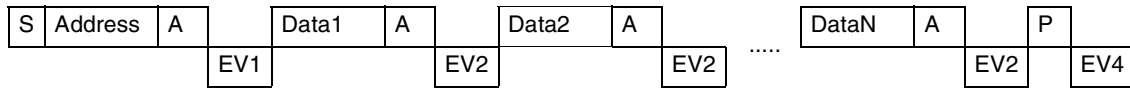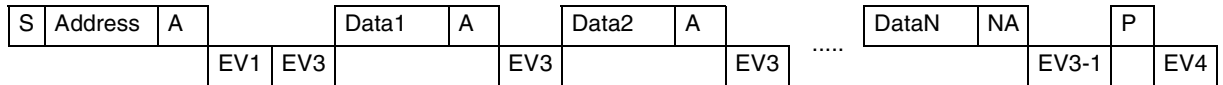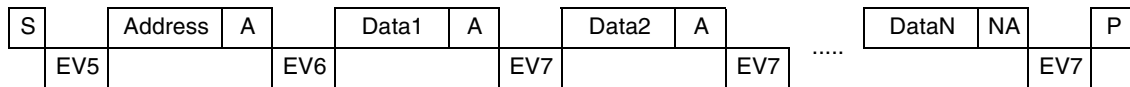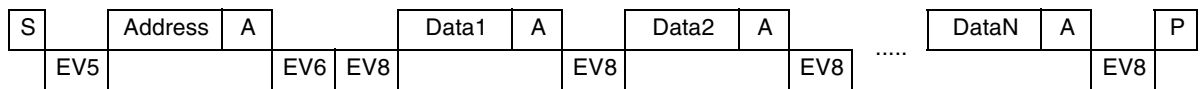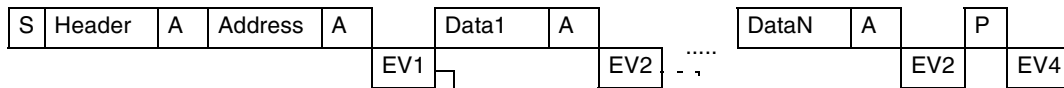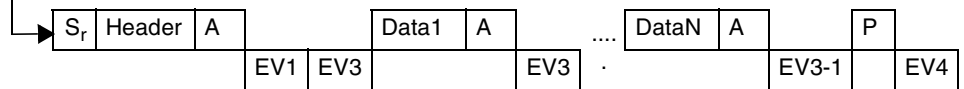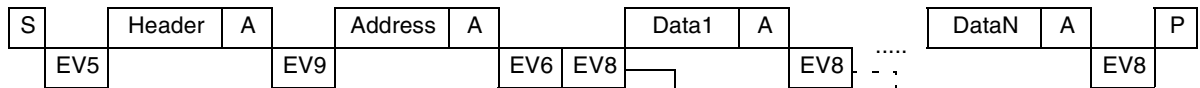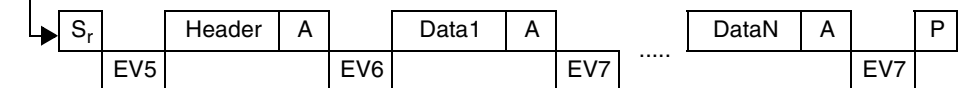
Legend: S=Start, S$_r$ = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1)

EV1: EVF=1, ADSL=1, cleared by reading SR1 register.

EV2: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV3-1: EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). **Note:** If lines are released by STOP=1, STOP=0, the subsequent EV4 is not seen.

EV4: EVF=1, STOPF=1, cleared by reading SR2 register.

EV5: EVF=1, SB=1, cleared by reading SR1 register followed by writing DR register.

EV6: EVF=1, cleared by reading SR1 register followed by writing CR register (for example PE=1).

EV7: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV8: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV9: EVF=1, ADD10=1, cleared by reading SR1 register followed by writing DR register.

I²C BUS INTERFACE (Cont'd)

10.7.5 Low Power Modes

Mode	Description
WAIT	No effect on I²C interface. I²C interrupts cause the device to exit from WAIT mode.
HALT	I²C registers are frozen. In HALT mode, the I²C interface is inactive and does not acknowledge data on the bus. The I²C interface resumes operation when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

10.7.6 Interrupts

Figure 63. Event Flags and Interrupt Generation

* EVF can also be set by EV6 or an error from the SR2 register.
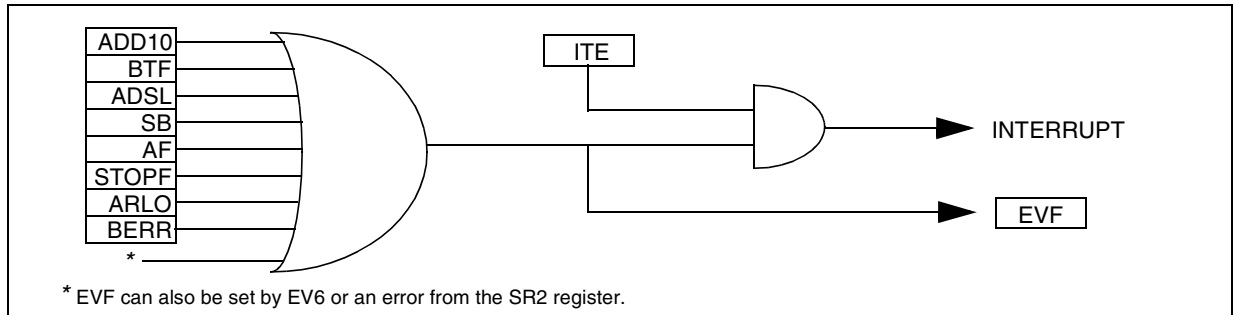
Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10		Yes	No
End of Byte Transfer Event	BTF		Yes	No
Address Matched Event (Slave mode)	ADSEL		Yes	No
Start Bit Generation Event (Master mode)	SB	ITE	Yes	No
Acknowledge Failure Event	AF		Yes	No
Stop Detection Event (Slave mode)	STOPF		Yes	No
Arbitration Lost Event (Multimaster configuration)	ARLO		Yes	No
Bus Error Event	BERR		Yes	No

Note: The I²C 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).

I^{2}C BUS INTERFACE (Cont'd)

10.7.7 Register Description

I^{2}C CONTROL REGISTER (CR)
Read / Write
Reset Value: 0000 0000 (00h)

7							0
0	0	PE	ENGC	START	ACK	STOP	ITE

Bit 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **PE** *Peripheral enable.*
This bit is set and cleared by software.
0: Peripheral disabled
1: Master/Slave capability
Notes:
– When PE=0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE=0
– When PE=1, the corresponding I/O pins are selected by hardware as alternate functions.
– To enable the I^{2}C interface, write the CR register **TWICE** with PE=1 as the first write only activates the interface (only PE is set).

Bit 4 = **ENGC** *Enable General Call.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0). The 00h General Call address is acknowledged (01h ignored).
0: General Call disabled
1: General Call enabled

Bit 3 = **START** *Generation of a Start condition.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0) or when the Start condition is sent (with interrupt generation if ITE=1).

– In master mode:
  0: No start generation
  1: Repeated start generation

– In slave mode:
  0: No start generation
  1: Start generation when the bus is free

Bit 2 = **ACK** *Acknowledge enable.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0).
0: No acknowledge returned
1: Acknowledge returned after an address byte or a data byte is received

Bit 1 = **STOP** *Generation of a Stop condition.*
This bit is set and cleared by software. It is also cleared by hardware in master mode. Note: This bit is not cleared when the interface is disabled (PE=0).

– In master mode:
  0: No stop generation
  1: Stop generation after the current byte transfer or after the current Start condition is sent. The STOP bit is cleared by hardware when the Stop condition is sent.

– In slave mode:
  0: No stop generation
  1: Release the SCL and SDA lines after the current byte transfer (BTF=1). In this mode the STOP bit has to be cleared by software.

Bit 0 = **ITE** *Interrupt enable.*
This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).
0: Interrupts disabled
1: Interrupts enabled
Refer to Figure 63 for the relationship between the events and the interrupt.
SCL is held low when the ADD10, SB, BTF or ADSL flags or an EV6 event (See Figure 62) is detected.

I²C BUS INTERFACE (Cont'd)

I²C STATUS REGISTER 1 (SR1)

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	ADD10	TRA	BUSY	BTF	ADSL	M/SL	SB

Bit 7 = **EVF** *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 62. It is also cleared by hardware when the interface is disabled (PE=0).
0: No event
1: One of the following events has occurred:

  – BTF=1 (Byte received or transmitted)
  – ADSL=1 (Address matched in Slave mode while ACK=1)
  – SB=1 (Start condition generated in Master mode)
  – AF=1 (No acknowledge received after byte transmission)
  – STOPF=1 (Stop condition detected in Slave mode)
  – ARLO=1 (Arbitration lost in Master mode)
  – BERR=1 (Bus error, misplaced Start or Stop condition detected)
  – ADD10=1 (Master has sent header byte)
  – Address byte successfully transmitted in Master mode.

Bit 6 = **ADD10** *10-bit addressing in Master mode*.
This bit is set by hardware when the master has sent the first byte in 10-bit address mode. It is cleared by software reading SR2 register followed by a write in the DR register of the second address byte. It is also cleared by hardware when the peripheral is disabled (PE=0).

0: No ADD10 event occurred.
1: Master has sent first address byte (header)

Bit 5 = **TRA** *Transmitter/Receiver.*
When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after detection of Stop condition (STOPF=1), loss of bus

arbitration (ARLO=1) or when the interface is disabled (PE=0).
0: Data byte received (if BTF=1)
1: Data byte transmitted

Bit 4 = **BUSY** *Bus busy.*
This bit is set by hardware on detection of a Start condition and cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. This information is still updated when the interface is disabled (PE=0).
0: No communication on the bus
1: Communication ongoing on the bus

Bit 3 = **BTF** *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

– Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See Figure 62). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.

– Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done
1: Byte transfer succeeded

Bit 2 = **ADSL** *Address matched (Slave mode).*
This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

0: Address mismatched or not received
1: Received address matched

I²C BUS INTERFACE (Cont'd)

Bit 1 = **M/SL** *Master/Slave.*
This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1). It is also cleared when the interface is disabled (PE=0).
0: Slave mode
1: Master mode

Bit 0 = **SB** *Start bit (Master mode).*
This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).
0: No Start condition
1: Start condition generated

I²C STATUS REGISTER 2 (SR2)
Read Only
Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL

Bit 7:5 = Reserved. Forced to 0 by hardware.

Bit 4 = **AF** *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while AF=1.

0: No acknowledge failure
1: Acknowledge failure

Bit 3 = **STOPF** *Stop detection (Slave mode).*
This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while STOPF=1.

0: No Stop condition detected
1: Stop condition detected

Bit 2 = **ARLO** *Arbitration lost*.
This bit is set by hardware when the interface loses the arbitration of the bus to another master. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

After an ARLO event the interface switches back automatically to Slave mode (M/SL=0).

The SCL line is not held low while ARLO=1.

0: No arbitration lost detected
1: Arbitration lost detected

Bit 1 = **BERR** *Bus error.*
This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while BERR=1.

0: No misplaced Start or Stop condition
1: Misplaced Start or Stop condition

Bit 0 = **GCAL** *General Call (Slave mode).*
This bit is set by hardware when a general call address is detected on the bus while ENGC=1. It is cleared by hardware detecting a Stop condition (STOPF=1) or when the interface is disabled (PE=0).

0: No general call address detected on bus
1: general call address detected on bus

I²C BUS INTERFACE (Cont'd)

I²C CLOCK CONTROL REGISTER (CCR)

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

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).
0: Standard I²C mode
1: Fast I²C mode

Bit 6:0 = **CC6-CC0** *7-bit clock divider.*
These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).

– Standard mode (FM/SM=0): $F_{SCL} <= 100kHz$

$$F_{SCL} = F_{CPU}/(2 \times ([CC6..CC0]+2))$$

– Fast mode (FM/SM=1): $F_{SCL} > 100kHz$

$$F_{SCL} = F_{CPU}/(3 \times ([CC6..CC0]+2))$$

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.

I²C DATA REGISTER (**DR**)

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

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D7-D0** *8-bit Data Register.*
These bits contain the byte to be received or transmitted on the bus.

– Transmitter mode: Byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the following data bytes are received one by one after reading the DR register.

I²C BUS INTERFACE (Cont'd)

I²C OWN ADDRESS REGISTER (OAR1)

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

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

7-bit Addressing Mode

Bit 7:1 = **ADD7-ADD1** *Interface address*.
These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

Bit 0 = **ADD0** *Address direction bit.*
This bit is don't care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

10-bit Addressing Mode

Bit 7:0 = **ADD7-ADD0** *Interface address*.
These are the least significant bits of the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

I²C OWN ADDRESS REGISTER (OAR2)

Read / Write
Reset Value: 0100 0000 (40h)

7							0
FR1	FR0	0	0	0	ADD9	ADD8	0

Bit 7:6 = **FR1-FR0** *Frequency bits.*

These bits are set by software only when the interface is disabled (PE=0). To configure the interface to I²C specifed delays select the value corresponding to the microcontroller frequency F_{CPU}.

F_{CPU} Range (MHz)	FR1	FR0
2.5 - 6	0	0
6 -10	0	1
10 - 14	1	0
14 - 24	1	1

Bit 5:3 = Reserved

Bit 2:1 = **ADD9-ADD8** *Interface address*.
These are the most significant bits of the I²C bus address of the interface (10-bit mode only). They are not cleared when the interface is disabled (PE=0).

Bit 0 = Reserved.

I²C BUS INTERFACE (Cont'd)

Table 22. I²C Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0018h	**I2CCR** Reset Value	0	0	PE 0	ENGC 0	START 0	ACK 0	STOP 0	ITE 0
0019h	**I2CSR1** Reset Value	EVF 0	ADD10 0	TRA 0	BUSY 0	BTF 0	ADSL 0	M/SL 0	SB 0
001Ah	**I2CSR2** Reset Value	0	0	0	AF 0	STOPF 0	ARLO 0	BERR 0	GCAL 0
001Bh	**I2CCCR** Reset Value	FM/SM 0	CC6 0	CC5 0	CC4 0	CC3 0	CC2 0	CC1 0	CC0 0
001Ch	**I2COAR1** Reset Value	ADD7 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
001Dh	**I2COAR2** Reset Value	FR1 0	FR0 1	0	0	0	ADD9 0	ADD8 0	0
001Eh	**I2CDR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

10.8 10-BIT A/D CONVERTER (ADC)

10.8.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.8.2 Main Features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation

- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 64.

10.8.3 Functional Description

10.8.3.1 Analog Power Supply

V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to device pin out description) they are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

Figure 64. ADC Block Diagram
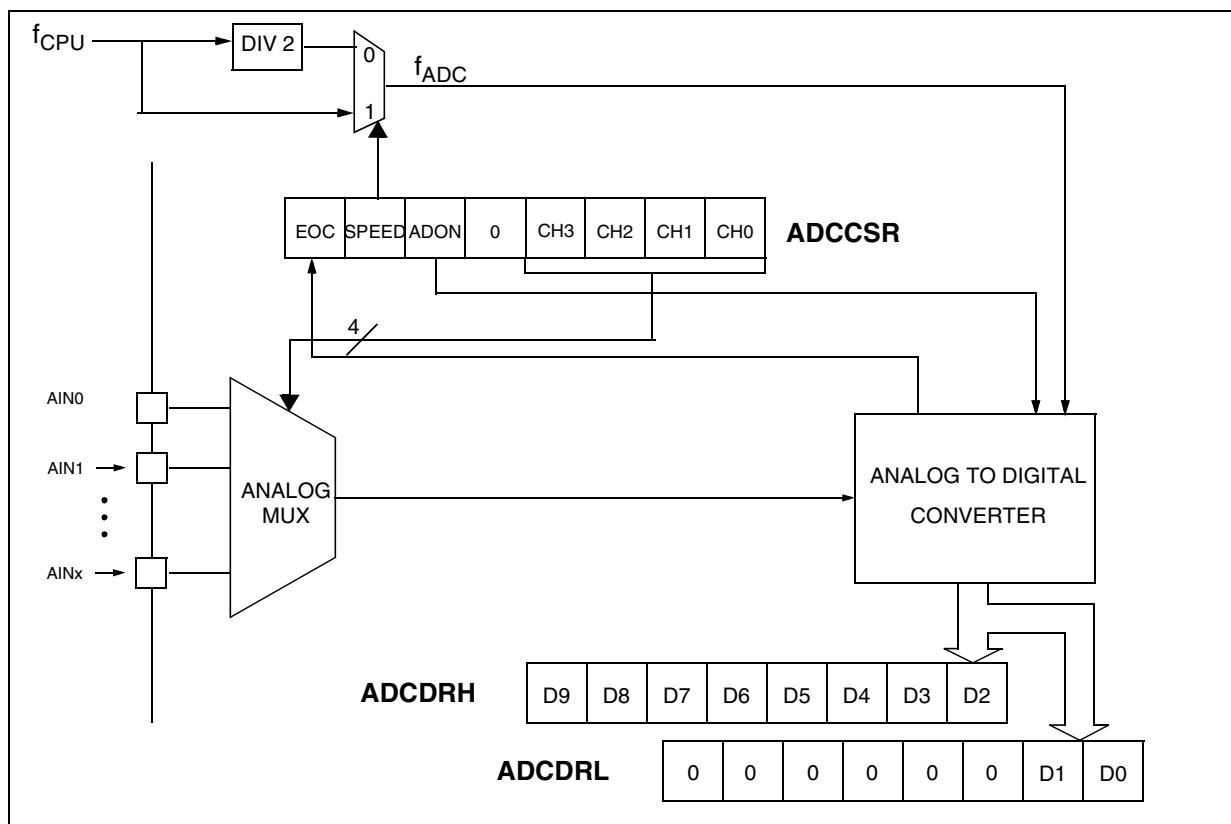
10-BIT A/D CONVERTER (ADC) (Cont'd)

10.8.3.2 Digital A/D Conversion Result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{DDA} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics Section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

10.8.3.3 A/D Conversion

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

– Select the CS[2:0] bits to assign the analog channel to convert.

ADC Conversion mode

In the ADCCSR register:

Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

– The EOC bit is set by hardware.
– The result is in the ADCDR registers.

A read to the ADCDRH resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll EOC bit
2. Read ADCDRL
3. Read ADCDRH. This clears EOC automatically.

To read only 8 bits, perform the following steps:

1. Poll EOC bit
2. Read ADCDRH. This clears EOC automatically.

10.8.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilisation time t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

10.8.5 Interrupts

None.

10-BIT A/D CONVERTER (ADC) (Cont'd)

10.8.6 Register Description

CONTROL/STATUS REGISTER (ADCCSR)

Read/Write (Except bit 7 read only)

Reset Value: 0000 0000 (00h)

7							0
EOC	SPEED	ADON	0	CH3	CH2	CH1	CH0

Bit 7 = **EOC** *End of Conversion*
This bit is set by hardware. It is cleared by software reading the ADCDRH register.
0: Conversion is not complete
1: Conversion complete

Bit 6 = **SPEED** *ADC clock selection*
This bit is set and cleared by software.
0: $f_{ADC} = f_{CPU}/2$
1: $f_{ADC} = f_{CPU}$

Bit 5 = **ADON** *A/D Converter on*
This bit is set and cleared by software.
0: Enable ADC and start conversion
1: Disable ADC and stop conversion

Bit 4 = **Reserved.** Must be kept cleared.

Bit 3:0 = **CH[3:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert.

Channel Pin*	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

*The number of channels is device dependent. Refer to the device pinout description.

DATA REGISTER (ADCDRH)

Read Only

Reset Value: 0000 0000 (00h)

7							0
D9	D8	D7	D6	D5	D4	D3	D2

Bit 7:0 = **D[9:2]** *MSB of Analog Converted Value*

DATA REGISTER (ADCDRL)

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	D1	D0

Bit 7:2 = Reserved. Forced by hardware to 0.

Bit 1:0 = **D[1:0]** *LSB of Analog Converted Value*

10-BIT A/D CONVERTER (Cont'd)

Table 23. ADC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	**ADCCSR** Reset Value	EOC 0	SPEED 0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0
0071h	**ADCDRH** Reset Value	D9 0	D8 0	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0
0072h	**ADCDRL** Reset Value							D1 0	D0 0

11 INSTRUCTION SET

11.1 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 24. ST7 Addressing Mode Overview

Mode			Syntax	Destination	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
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+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	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

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.1 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 (level 3)
RIM	Reset Interrupt Mask (level 0)
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

11.1.2 Immediate

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

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

11.1.3 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.

11.1.4 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.

11.1.5 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.

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.6 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 25. 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 Additions/Substractions 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

11.1.7 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 sub-modes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

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

INSTRUCTION SET OVERVIEW (Cont'd)

11.2 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 opcodes.

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 SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			I1	H	I0	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 Port B INT pin = 1	(no Port B Interrupts)								
JRIL	Jump if Port B INT pin = 0	(Port B interrupt)								
JRH	Jump if H = 1	H = 1 ?								
JRNH	Jump if H = 0	H = 0 ?								
JRM	Jump if I1:0 = 11	I1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?								
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 SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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	Pop from the Stack	pop CC	CC	M	I1	H	I0	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	I1:0 = 10 (level 0)			1		0			
RLC	Rotate left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate right true C	C => A => C	reg, M					N	Z	C
RSP	Reset Stack Pointer	S = Max allowed								
SBC	Substract with Carry	A = A - M - C	A	M				N	Z	C
SCF	Set carry flag	C = 1								1
SIM	Disable Interrupts	I1:0 = 11 (level 3)			1		1			
SLA	Shift left Arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift left Logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift right Logic	0 => A => C	reg, M					0	Z	C
SRA	Shift right Arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Substraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for Neg & Zero	tnz lbl1						N	Z	
TRAP	S/W trap	S/W interrupt			1		1			
WFI	Wait for Interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

12 ELECTRICAL CHARACTERISTICS

12.1 PARAMETER CONDITIONS

Unless otherwise specified, all voltages are referred to V_{SS}.

12.1.1 Minimum and Maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at T_A=25°C and T_A=$T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean±3Σ).

12.1.2 Typical values

Unless otherwise specified, typical data are based on T_A=25°C, V_{DD}=5V (for the 4.5V≤V_{DD}≤5.5V voltage range) and V_{DD}=3.3V (for the 3V≤V_{DD}≤4V voltage range). They are given only as design guidelines and are not tested.
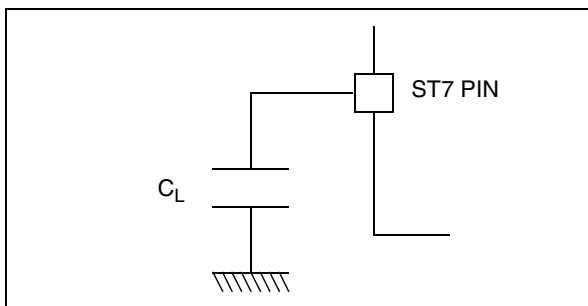
12.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

12.1.4 Loading capacitor

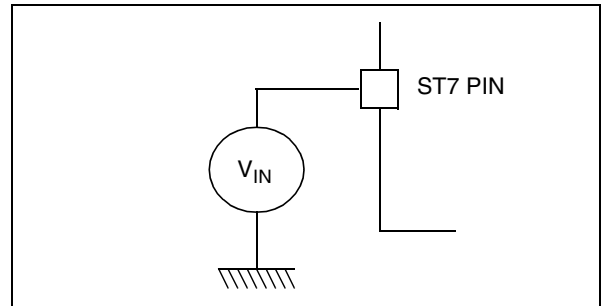The loading conditions used for pin parameter measurement are shown in Figure 65.

Figure 65. Pin loading conditions

12.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 66.

Figure 66. Pin input voltage

12.2 ABSOLUTE MAXIMUM RATINGS

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 under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

12.2.1 Voltage Characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{PP} - V_{SS}	Programming Voltage	14	
V_{IN} [1) & 2)]	Input Voltage on true open drain pin	V_{SS}-0.3 to 6.5	
	Input voltage on any other pin	V_{SS}-0.3 to V_{DD}+0.3	
$|\Delta V_{DDx}|$ and $|\Delta V_{SSx}|$	Variations between different digital power pins	50	mV
$|V_{SSA} - V_{SSx}|$	Variations between digital and analog ground pins	50	
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	see Section 12.7.3 on page 148	
$V_{ESD(MM)}$	Electro-static discharge voltage (Machine Model)		

12.2.2 Current Characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) [3)]	150	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) [3)]	150	
I_{IO}	Output current sunk by any standard I/O and control pin	25	
	Output current sunk by any high sink I/O pin	50	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$ [2) & 4)]	Injected current on V_{PP} pin	± 5	
	Injected current on $\overline{RESET}$ pin	± 5	
	Injected current on OSC1 and OSC2 pins	± 5	
	Injected current on any other pin [5) & 6)]	± 5	
$\Sigma I_{INJ(PIN)}$ [2)]	Total injected current (sum of all I/O and control pins) [5)]	± 25	

12.2.3 Thermal Characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see Section 13.2 THERMAL CHARACTERISTICS)		

Notes:
1. Directly connecting the $\overline{RESET}$ and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7kΩ for $\overline{RESET}$, 10kΩ for I/Os). For the same reason, unused I/O pins must not be directly tied to V_{DD} or V_{SS}.
2. When the current limitation is not possible, the V_{IN} absolute maximum rating must be respected, otherwise refer to $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$.
3. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.
4. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
- Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
- Pure digital pins must have a negative injection less than 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
5. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterisation with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.
6. True open drain I/O port pins do not accept positive injection.

12.3 OPERATING CONDITIONS

12.3.1 General Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CPU}	Internal clock frequency versus V_{DD}	FLASH: $2.7V \leq V_{DD} \leq 5.5V$	0 [1]	4	MHz
		FLASH: $3.85V$ [2] $\leq V_{DD} \leq 5.5V$ ROM: $2.7V \leq V_{DD} \leq 5.5V$		8	
T_A	Ambient temperature range	1 Suffix Version	0	70	°C
		5 Suffix Version	-10	85	
		6 Suffix Version	-40	85	
		7 Suffix Version	-40	105	
		3 Suffix Version	-40	125	

Notes:

1. Guaranteed by construction. A/D operation and resonator oscillator start-up are not guaranteed below 1 MHz.

2. This value is the minimum LVD V_{IT-} value when the highest LVD threshold is selected.

OPERATING CONDITIONS (Cont'd)

12.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IT+}	Reset release threshold (V_{DD} rise)	High Threshold Med. Threshold Low Threshold	4.10 [2] 3.75 [2] 3.25 [2]	4.30 3.90 3.35	4.50 4.05 3.45	V
V_{IT-}	Reset generation threshold (V_{DD} fall)	High Threshold Med. Threshold Low Threshold	3.85 3.50 3.00	4.05 3.65 3.10	4.25 3.80 3.20	
V_{hys}	LVD voltage threshold hysteresis	V_{IT+}-V_{IT-}	200	250	300	mV
Vt_{POR}	V_{DD} rise time rate [3]		5		5000	µs/V
$t_{g(VDD)}$	Filtered glitch delay on V_{DD} [2]	Not detected by the LVD			40	ns

Notes:

1. LVD typical data are based on T_A=25°C. They are given only as design guidelines and are not tested.

2. Data based on characterization results, not tested in production.

3. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. Not tested in production.

12.3.3 Internal Voltage Detector (IVD) Thresholds

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IT+}	1⇒0 VDF flag toggle threshold (V_{DD} rise)	VD level = Low in option byte VD level = Med. in option byte VD level = High in option byte		3.60 4.00 4.95		V
V_{IT-}	0⇒1 VDF flag toggle threshold (V_{DD} fall)	VD level = Low in option byte VD level = Med. in option byte VD level = High in option byte		3.35 3.75 4.70		
V_{hyst}	IVD voltage threshold hysteresis	V_{IT+}-V_{IT-}		250		mV

12.3.4 External Voltage Detector (EVD) Thresholds

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IT+}	1⇒0 VDF flag toggle threshold (V_{DD} rise)			1.35		V
V_{IT-}	0⇒1 VDF flag toggle threshold (V_{DD} fall)			1.10		
V_{hyst}	EVD voltage threshold hysteresis	V_{IT+}-V_{IT-}		250		mV

12.4 SUPPLY CURRENT CHARACTERISTICS

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode for which the clock is stopped).

Symbol	Parameter	Conditions	Max	Unit
$\Delta I_{DD(\Delta Ta)}$	Supply current variation vs. temperature	Constant V_{DD} and f_{CPU}	10	%

12.4.1 RUN and SLOW Modes

Symbol	Parameter	Conditions		Typ [1]	Max [2]	Unit
I_{DD}	Supply current in RUN mode [3] (see Figure 67)	$4.5V \leq V_{DD} \leq 5.5V$	f_{OSC}=2MHz, f_{CPU}=1MHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=8MHz, f_{CPU}=4MHz f_{OSC}=16MHz, f_{CPU}=8MHz	1.5 2.3 3.7 6.5	2.5 3.5 5.0 8.0	mA
	Supply current in SLOW mode [4] (see Figure 68)		f_{OSC}=2MHz, f_{CPU}=62.5kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=8MHz, f_{CPU}=250kHz f_{OSC}=16MHz, f_{CPU}=500kHz	0.9 1.0 1.2 1.4	2.0 2.2 2.4 2.6	
	Supply current in RUN mode [3] (see Figure 67)	$3V \leq V_{DD} \leq 3.6V$	f_{OSC}=2MHz, f_{CPU}=1MHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=8MHz, f_{CPU}=4MHz f_{OSC}=16MHz, f_{CPU}=8MHz	0.8 1.2 2.0 3.7	1.0 1.5 3.0 5.0	
	Supply current in SLOW mode [4] (see Figure 68)		f_{OSC}=2MHz, f_{CPU}=62.5kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=8MHz, f_{CPU}=250kHz f_{OSC}=16MHz, f_{CPU}=500kHz	0.35 0.4 0.5 0.6	0.6 0.7 0.8 1.0	

Figure 67. Typical I_{DD} in RUN vs. f_{CPU}
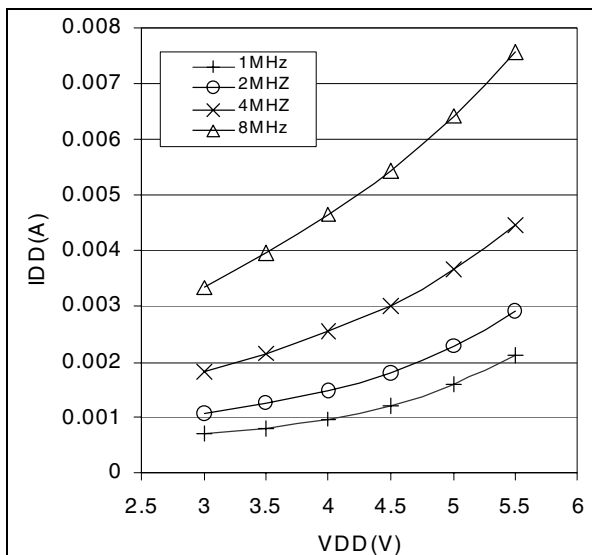
Figure 68. Typical I_{DD} in SLOW vs. f_{CPU}
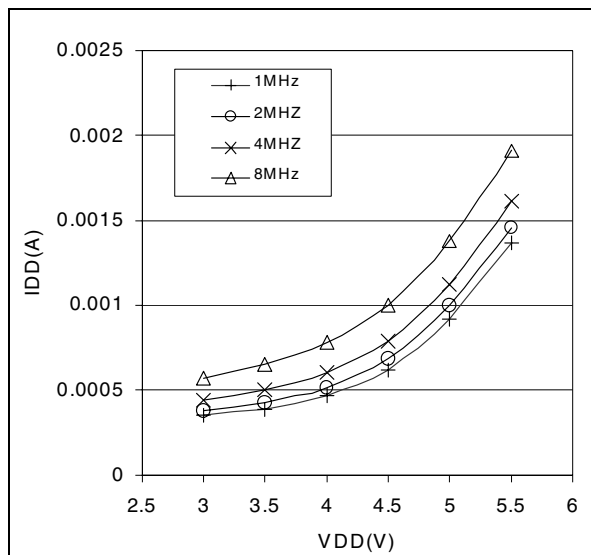
Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \leq V_{DD} \leq 5.5V$ range) and V_{DD}=3.3V ($3V \leq V_{DD} \leq 3.6V$ range).

2. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.

3. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (OSC1) driven by external square wave, CSS and LVD disabled.

4. SLOW mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (OSC1) driven by external square wave, CSS and LVD disabled.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.2 WAIT and SLOW WAIT Modes

Symbol	Parameter	Conditions		Typ [1]	Max [2]	Unit
I_{DD}	Supply current in WAIT mode [3] (see Figure 69)	$4.5V \leq V_{DD} \leq 5.5V$	f_{OSC}=2MHz, f_{CPU}=1MHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=8MHz, f_{CPU}=4MHz f_{OSC}=16MHz, f_{CPU}=8MHz	1.9 2.3 3.0 4.7	2.7 3.2 4.0 5.8	mA
	Supply current in SLOW WAIT mode [4] (see Figure 70)		f_{OSC}=2MHz, f_{CPU}=62.5kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=8MHz, f_{CPU}=250kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD TBD TBD	TBD TBD TBD TBD	
	Supply current in WAIT mode [3] (see Figure 69)	$3V \leq V_{DD} \leq 3.6V$	f_{OSC}=2MHz, f_{CPU}=1MHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=8MHz, f_{CPU}=4MHz f_{OSC}=16MHz, f_{CPU}=8MHz	0.7 0.9 1.4 2.4	1.4 1.7 2.2 3.3	
	Supply current in SLOW WAIT mode [4] (see Figure 70)		f_{OSC}=2MHz, f_{CPU}=62.5kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=8MHz, f_{CPU}=250kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD TBD TBD	TBD TBD TBD TBD	

Figure 69. Typical I_{DD} in WAIT vs. f_{CPU}
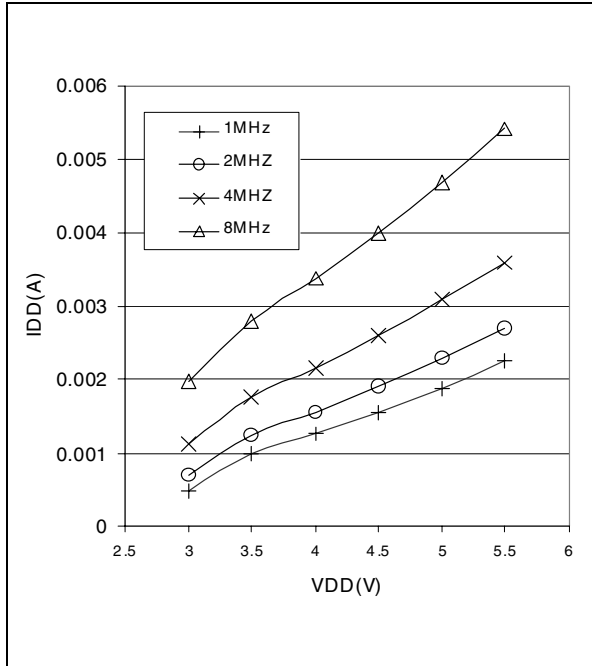
Figure 70. Typical I_{DD} in SLOW-WAIT vs. f_{CPU}
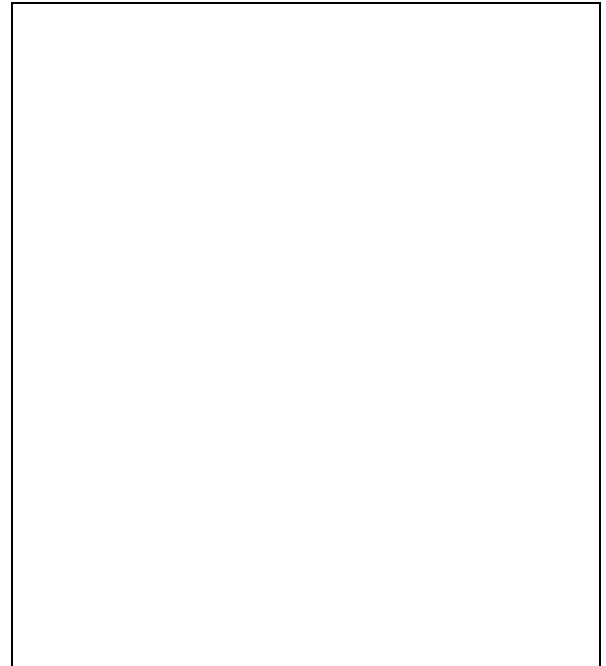
Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V (4.5V≤V_{DD}≤5.5V range) and V_{DD}=3.3V (3V≤V_{DD}≤3.6V range).

2. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.

3. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (OSC1) driven by external square wave, CSS and LVD disabled.

4. SLOW-WAIT mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (OSC1) driven by external square wave, CSS and LVD disabled.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.3 HALT and ACTIVE-HALT Modes

Symbol	Parameter	Conditions		Typ [1]	Max	Unit
I_{DD}	Supply current in HALT mode [2]	V_{DD}=5.5V	-40°C≤T_A≤+85°C	0	10	μA
			-40°C≤T_A≤+125°C		50	
		V_{DD}=3.6V	-40°C≤T_A≤+85°C		6	
			-40°C≤T_A≤+125°C		50	
	Supply current in ACTIVE-HALT mode [3]			50	150	

12.4.4 Supply and Clock Managers

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode).

Symbol	Parameter	Conditions	Typ [1]	Max [4]	Unit
$I_{DD(CK)}$	Supply current of internal RC oscillator		see Section 12.5.4 on page 144		μA
	Supply current of external RC oscillator [5]				
	Supply current of resonator oscillator [5] & [6]		see Section 12.5.3 on page 142		
	Clock security system supply current		150	350	
$I_{DD(LVD)}$	LVD supply current	HALT mode	100	150	

Notes:

1. Typical data are based on T_A=25°C.

2. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), CSS and LVD disabled. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.

3. Data based on design simulation and/or technology characteristics, not tested in production. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled.

4. Data based on characterization results, not tested in production.

5. Data based on characterization results done with the external components specified in Section 12.5.3 and Section 12.5.4, not tested in production.

6. As the oscillator is based on a current source, the consumption does not depend on the voltage.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.5 On-Chip Peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(TIM)}$	16-bit Timer supply current [1]	f_{CPU}=8MHz	V_{DD}=3.3V	50	μA
			V_{DD}=5.0V	150	
$I_{DD(I2C)}$	I2C supply current	f_{CPU}=8MHz	V_{DD}=5.0V	TBD	
$I_{DD(CAN)}$	CAN supply current	f_{CPU}=8MHz	V_{DD}=5.0V	TBD	
$I_{DD(ART)}$	ART PWM supply current	f_{CPU}=8MHz	V_{DD}=3.3V	TBD	
			V_{DD}=5.0V	TBD	
$I_{DD(SPI)}$	SPI supply current [2]	f_{CPU}=8MHz	V_{DD}=3.3V	250	
			V_{DD}=5.0V	350	
$I_{DD(ADC)}$	ADC supply current when converting [3]	f_{ADC}=4MHz	V_{DD}=3.3V	800	
			V_{DD}=5.0V	1100	

Notes:

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at f_{CPU}/4) and timer counter stopped (selecting external clock capability). Data valid for one timer.

2. Data based on a differential I_{DD} measurement between reset configuration and a permanent SPI master communication (data sent equal to 55h).

3. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

12.5 CLOCK AND TIMING CHARACTERISTICS

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A.

12.5.1 General Timings

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
$t_{c(INST)}$	Instruction cycle time		2	3	12	t_{CPU}
		f_{CPU}=8MHz	250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time [2] $t_{v(IT)} = \Delta t_{c(INST)} + 10$		10		22	t_{CPU}
		f_{CPU}=8MHz	1.25		2.75	µs

12.5.2 External Clock Source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage		$0.7 \times V_{DD}$		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		$0.3 \times V_{DD}$	
$t_{w(OSC1H)}$ $t_{w(OSC1L)}$	OSC1 high or low time [3]	see Figure 71	15			ns
$t_{r(OSC1)}$ $t_{f(OSC1)}$	OSC1 rise or fall time [3]				15	
I_L	OSCx Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			±1	µA

Figure 71. Typical Application with an External Clock Source
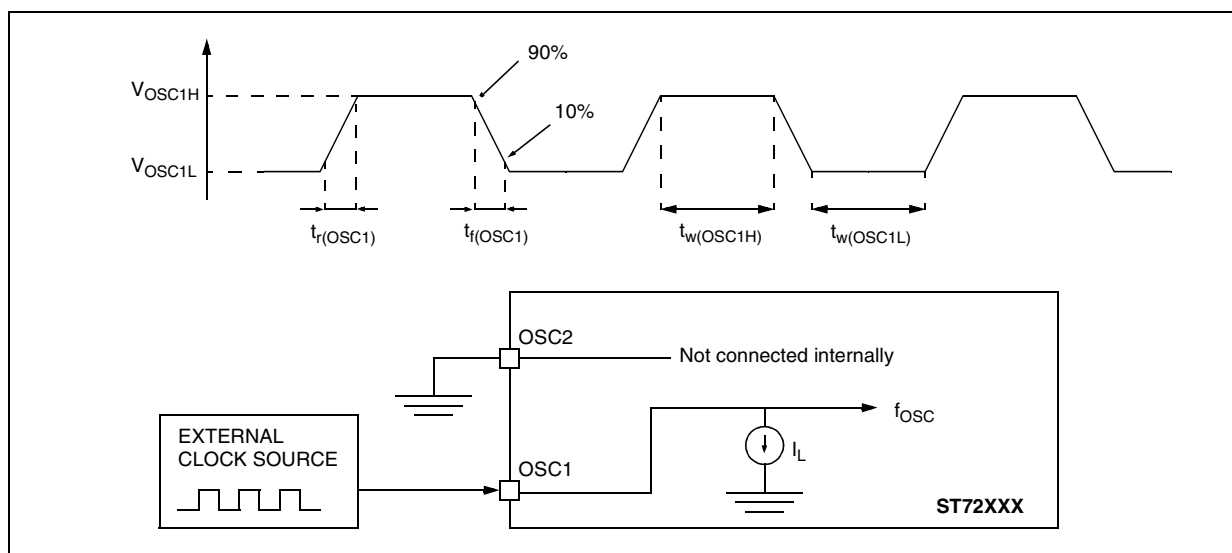
Notes:

1. Data based on typical application software.

2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

3. Data based on design simulation and/or technology characteristics, not tested in production.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

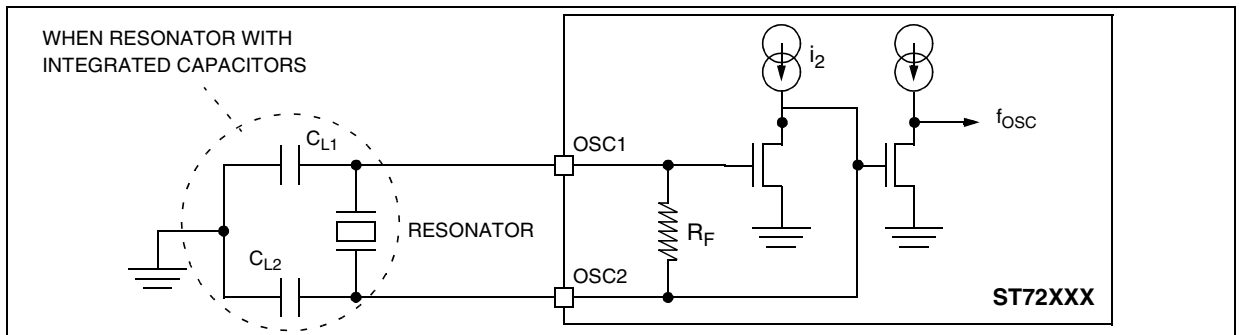### 12.5.3 Crystal and Ceramic Resonator Oscillators

The ST7 internal clock can be supplied with four different Crystal/Ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

Symbol	Parameter	Conditions		Min	Max	Unit
f_{OSC}	Oscillator Frequency [1]	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator		1 >2 >4 >8	2 4 8 16	MHz
R_F	Feedback resistor			20	40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	R_S=200Ω R_S=200Ω R_S=200Ω R_S=100Ω	VLP oscillator LP oscillator MP oscillator MS oscillator HS oscillator	TBD 38 32 18 15	TBD 56 46 26 21	pF

Symbol	Parameter	Conditions		Typ	Max	Unit
i_2	OSC2 driving current	V_{DD}=5V V_{IN}=V_{SS}	VLP oscillator LP oscillator MP oscillator MS oscillator HS oscillator	10 90 180 350 690	20 150 250 450 850	μA

Figure 72. Typical Application with a Crystal or Ceramic Resonator

Notes:

1. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

Oscil.			Typical Crystal or Ceramic Resonators			C_{L1}	C_{L2}	$t_{SU(osc)}$
			Reference	Freq.	Characteristic [1]	[pF]	[pF]	[ms] [2]
Crystal	LP	JAUCH	S-200-30-30/50	2MHz	$\Delta f_{OSC}=[\pm30ppm_{25°C},\pm30ppm_{\Delta Ta}]$, Typ. $R_S=200\Omega$	22	22	8
	MP		SS3-400-30-30/30	4MHz	$\Delta f_{OSC}=[\pm30ppm_{25°C},\pm30ppm_{\Delta Ta}]$, Typ. $R_S=60\Omega$	22	22	6.5
	MS		SS3-800-30-30/30	8MHz	$\Delta f_{OSC}=[\pm30ppm_{25°C},\pm30ppm_{\Delta Ta}]$, Typ. $R_S=25\Omega$	33	33	3.25
	HS		SS3-1600-30-30/30	16MHz	$\Delta f_{OSC}=[\pm30ppm_{25°C},\pm30ppm_{\Delta Ta}]$, Typ. $R_S=15\Omega$	33	33	1.3
Ceramic	LP	MURATA	CSA2.00MG	2MHz	$\Delta f_{OSC}=[\pm0.5\%_{tolerance},\pm0.3\%_{\Delta Ta},\pm0.3\%_{aging},\pm x.x\%_{correl}]$	22	22	4
	MP		CSA4.00MG	4MHz	$\Delta f_{OSC}=[\pm0.5\%_{tolerance},\pm0.3\%_{\Delta Ta},\pm0.3\%_{aging},\pm x.x\%_{correl}]$	22	22	2
	MS		CSA8.00MTZ	8MHz	$\Delta f_{OSC}=[\pm0.5\%_{tolerance},\pm0.5\%_{\Delta Ta},\pm0.3\%_{aging},\pm x.x\%_{correl}]$	33	33	1
	HS		CSA16.00MXZ040	16MHz	$\Delta f_{OSC}=[\pm0.5\%_{tolerance},\pm0.3\%_{\Delta Ta},\pm0.3\%_{aging},\pm x.x\%_{correl}]$	33	33	0.7

Notes:

1. Resonator characteristics given by the crystal/ceramic resonator manufacturer.

2. $t_{SU(OSC)}$ is the typical oscillator start-up time measured between $V_{DD}=2.8V$ and the fetch of the first instruction (with a quick V_{DD} ramp-up from 0 to 5V (<50µs).

CLOCK CHARACTERISTICS (Cont'd)

12.5.4 RC Oscillators

The ST7 internal clock can be supplied with an RC oscillator. This oscillator can be used with internal or external components (selectable by option byte).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	Internal RC oscillator frequency [1]	R_{IN}=50KΩ, C_{IN}=20pF ±20%		4		MHz
	External RC oscillator frequency [2]	R_{EX}=33KΩ, C_{EX}=22pF ±5% R_{EX}=47KΩ, C_{EX}=22pF ±5% R_{EX}=47KΩ, C_{EX}=100pF ±5%		4.3 3.1 1.2		
$t_{SU(OSC)}$	Internal RC Oscillator Start-up Time [3]	R_{IN}=50KΩ, C_{IN}=20pF ±20%		60		µs
	External RC Oscillator Start-up Time [3]	R_{EX}=33KΩ, C_{EX}=22pF ±5% R_{EX}=47KΩ, C_{EX}=22pF ±5% R_{EX}=47KΩ, C_{EX}=100pF ±5%		60 80 200		
R_{EX}	Oscillator external resistor [4]		10		100	KΩ
C_{EX}	Oscillator external capacitor		15 [5]		470	pF
$|i_{CEX}|$	Capacitor load current [6]	OSC1 = V_{SS} or 1.5V		290	320	µA

Figure 73. Typical Application with RC oscillator
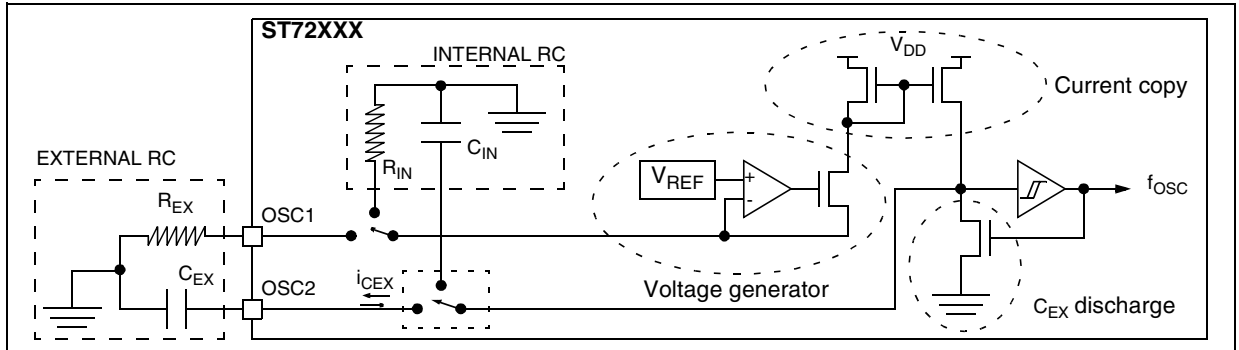
Notes:

1. Data based on characterization results.

2. Guaranteed frequency range with the specified C_{EX} and R_{EX} ranges taking into account the device process variation. Data based on design simulation.

3. Data based on characterization results done with V_{DD} nominal at 5V, not tested in production.

4. R_{EX} must have a positive temperature coefficient (ppm/°C), carbon resistors should therefore not be used.

5. **Important:** when no external C_{EX} is applied, the capacitance to be considered is the global parasitic capacitance which is subject to high variation (package, application...). In this case, the RC oscillator frequency tuning has to be done by trying out several resistor values.

6. i_{CEX} is the current needed to load the C_{EX} capacitor while OSC1 is forced to V_{SS} or 1.5V (RC oscillation voltage range)

CLOCK CHARACTERISTICS (Cont'd)

12.5.5 Clock Security System (CSS)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SFOSC}	Safe Oscillator Frequency [1]		2	3	4	MHz
Δf_{CPU}	Typical CSS operating frequency range limits (versus oscillator range option) [2]	32kHz~100kHz	13.5		115	kHz
		1MHz~2MHz	0.44		3.3	MHz
		2MHz~4MHz with PLL off	845		7.4	
		4MHz~8MHz	1.65		30	
		8MHz~16MHz or 2MHz~4MHz with PLL on	3.2			

Note:

1. Data based on simulation results.
2. Data given only as information, to be characterized later on.

12.5.6 PLL Characteristics

Symbol	Parameter	Conditions	Typ	Max	Unit
I_{PLL}	PLL Consumption [1]		100	150	µA
f_{OSC}	PLL input frequency range [1]		3	5	MHz
$t_{w(JIT)}$	PLL jitter period [1]		60		kHz
$\Delta f_{CPU}/f_{CPU}$	PLL jitter [1]		± 0.4	± 1	%

Figure 74. PLL $\Delta f_{CPU}/f_{CPU}$ versus the time
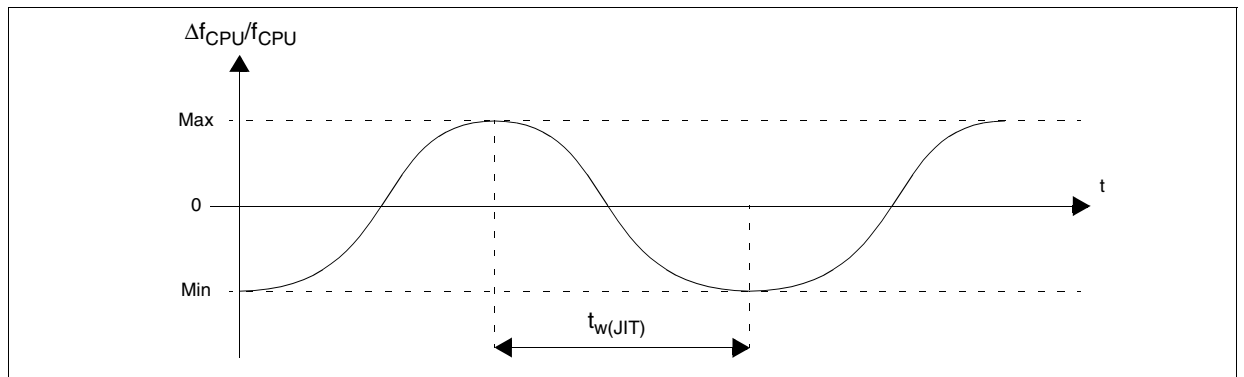
Note:

1. Data based on characterization results, not tested in production.

12.6 MEMORY CHARACTERISTICS

12.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	HALT mode (or RESET)	1.6			V

12.6.2 FLASH Memory

DUAL VOLTAGE HDFLASH MEMORY						
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CPU}	Operating frequency	read mode	0		8	MHz
		write / erase mode	1		8	
V_{PP}	Programming voltage [2]	4.5V <= V_{DD} <= 5.5V	11.4		12.6	V
I_{DD}	Supply current	RUN mode (f_{CPU} = 4MHz)			3	mA
		Write / Erase		0		
		Power down mode / HALT		1	10	
I_{PP}	VPP current	Read $\quad$ $V_{PP} > V_{DD}$ $\quad\quad$ $V_{PP} <= V_{DD}$			200 ±15	µA
		Write / Erase			30	mA
t_{PROG}	Byte programming time			100	TBD	µs
t_{ERASE}	Sector erasing time	f_{CPU} = 8MHz		0.5	TBD	sec
	Device erasing time			2	10	
		f_{CPU} = 8MHz, T_A=25°C			3	
t_{VPP}	Internal V_{PP} stabilization time			10		µs
t_{RET}	Data retention	T_A=55°C	20			years
N_{RW}	Write erase cycles	T_A=25°C	100			cycles
T_{PROG} T_{ERASE}	Programming or erasing temperature range		TBD	25	TBD	°C

Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in HALT mode or under RESET) or in hardware registers (only in HALT mode). Guaranted by construction, not tested in production.

2. V_{PP} must be applied only during the programming or erasing operation and not permanently for reliability reasons.

12.7 EMC CHARACTERISTICS

Susceptibility tests are performed on a sample basis during product characterization.

12.7.1 Functional EMS

(Electro Magnetic Susceptibility)

Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electro magnetic events until a failure occurs (indicated by the LEDs).

■ **ESD**: Electro-Static Discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.

■ **FTB**: A Burst of Fast Transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed.

Symbol	Parameter	Conditions	Neg [1]	Pos [1]	Unit
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD}=5V, T_A=+25°C, f_{OSC}=8MHz conforms to IEC 1000-4-2	-1	1	kV
V_{FFTB}	Fast transient voltage burst limits to be applied through 100pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD}=5V, T_A=+25°C, f_{OSC}=8MHz conforms to IEC 1000-4-4	-1.5	1.5	kV

Figure 75. EMC Recommended star network power supply connection [2]
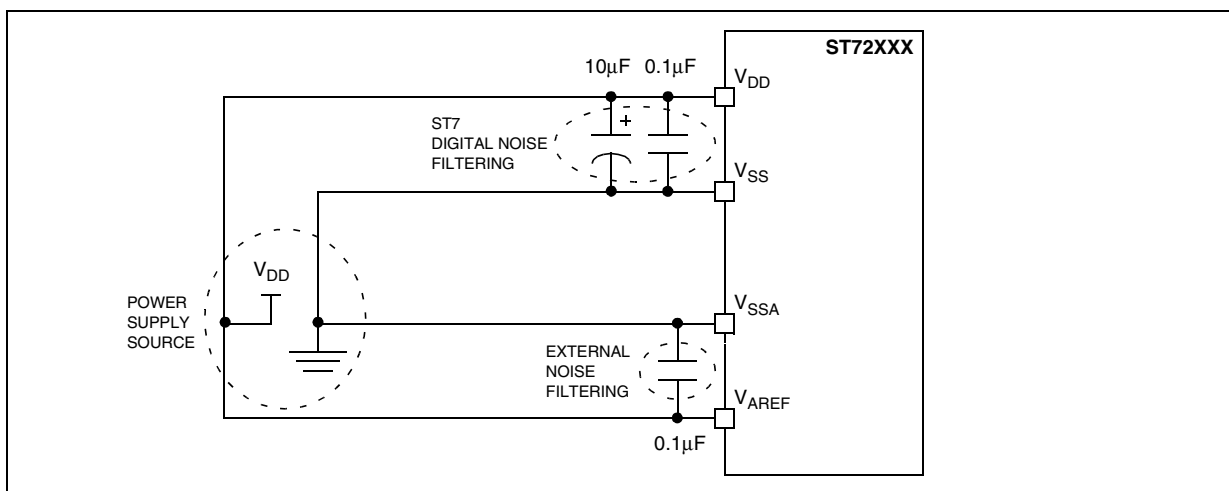
12.7.2 Electro Magnetic Interference (EMI)

Based on a simple application running on the product (toggling 2 LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Symbol	Parameter	Conditions	Monitored Frequency Band	Max vs. [f_{OSC}/f_{CPU}]		Unit
				8/4MHz	**16/8MHz**	
S_{EMI}	Peak level	V_{DD}=5V, T_A=+25°C, TQFP64 14x14 package conform with SAE J 1752/3	0.1MHz to 30MHz	9	9	dBµV
			30MHz to 130MHz	15	21	
			130MHz to 1GHz	-1	4	
			SAE EMI Level	1.5	2.0	-

Notes:

1. Data based on characterization results, not tested in production.

2. The suggested 10µF and 0.1µF decoupling capacitors on the power supply lines are proposed as a good price vs. EMC performance trade-off. They have to be put as close as possible to the device power supply pins. Other EMC recommendations are given in other sections (I/Os, RESET, OSCx pin characteristics).

EMC CHARACTERISTICS (Cont'd)

12.7.3 Absolute Electrical Sensitivity

Based on three different tests (ESD, LU and DLU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the AN1181 ST7 application note.

12.7.3.1 Electro-Static Discharge (ESD)

Electro-Static Discharges (3 positive then 3 negative pulses separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends of the number of supply pins of the device (3 parts*(n+1) supply pin). Two models are usually simulated: Human Body Model and Machine Model. This test conforms to the JESD22-A114A/A115A standard. See Figure 76 and the following test sequences.

Human Body Model Test Sequence

– C_L is loaded through S1 by the HV pulse generator.

– S1 switches position from generator to R.

– A discharge from C_L through R (body resistance) to the ST7 occurs.

– S2 must be closed 10 to 100ms after the pulse delivery period to ensure the ST7 is not left in charge state. S2 must be opened at least 10ms prior to the delivery of the next pulse.

Machine Model Test Sequence

– C_L is loaded through S1 by the HV pulse generator.

– S1 switches position from generator to ST7.

– A discharge from C_L to the ST7 occurs.

– S2 must be closed 10 to 100ms after the pulse delivery period to ensure the ST7 is not left in charge state. S2 must be opened at least 10ms prior to the delivery of the next pulse.
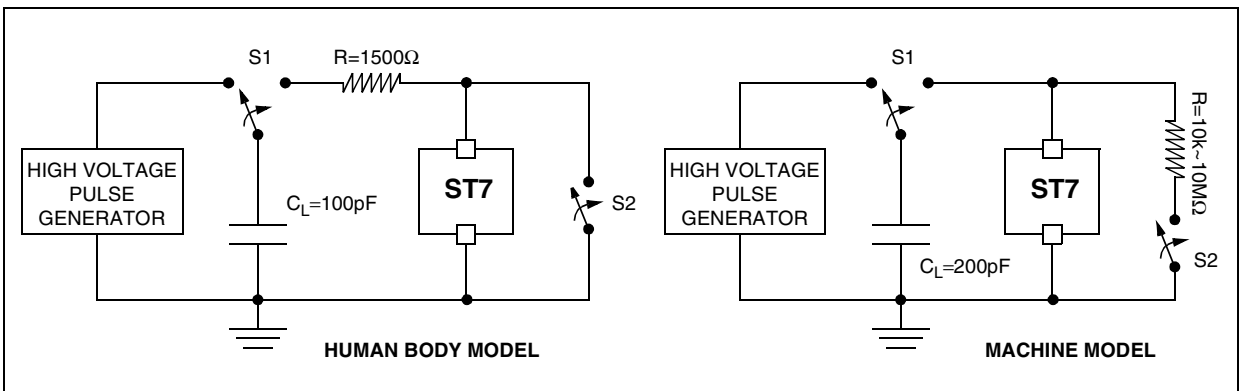
– R (machine resistance), in series with S2, ensures a slow discharge of the ST7.

Absolute Maximum Ratings

Symbol	Ratings	Conditions	Maximum value [1]	Unit
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	T_A=+25°C	2000	V
$V_{ESD(MM)}$	Electro-static discharge voltage (Machine Model)	T_A=+25°C	TBD	

Figure 76. Typical Equivalent ESD Circuits

Notes:

1. Data based on characterization results, not tested in production.

EMC CHARACTERISTICS (Cont'd)

12.7.3.2 Static and Dynamic Latch-Up

■ **LU**: 3 complementary static tests are required on 10 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin), a current injection (applied to each input, output and configurable I/O pin) and a power supply switch sequence are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard. For more details, refer to the AN1181 ST7 application note.

■ **DLU**: Electro-Static Discharges (one positive then one negative test) are applied to each pin of 3 samples when the micro is running to assess the latch-up performance in dynamic mode. Power supplies are set to the typical values, the oscillator is connected as near as possible to the pins of the micro and the component is put in reset mode. This test conforms to the IEC1000-4-2 and SAEJ1752/3 standards and is described in Figure 77. For more details, refer to the AN1181 ST7 application note.

Electrical Sensitivities

Symbol	Parameter	Conditions	Class [1]
LU	Static latch-up class	T_A=+25°C T_A=+85°C T_A=+125°C	A A A
DLU	Dynamic latch-up class	V_{DD}=5.5V, f_{OSC}=4MHz, T_A=+25°C	A

Figure 77. Simplified Diagram of the ESD Generator for DLU
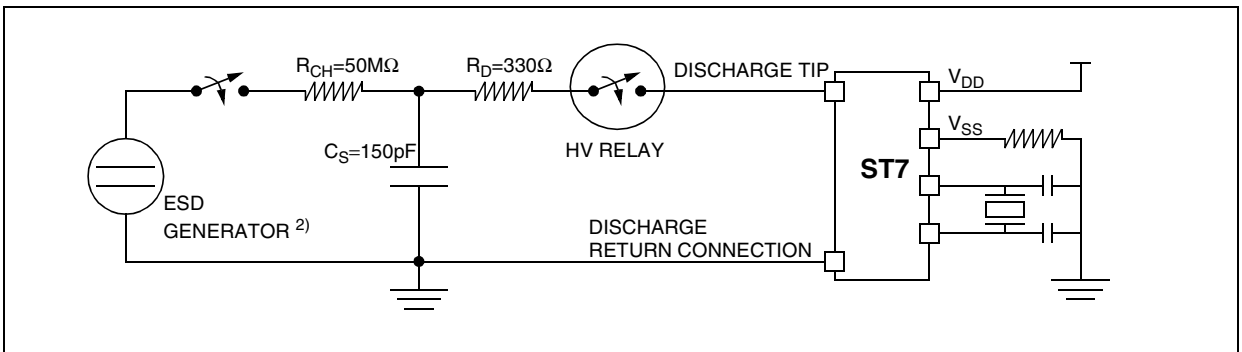
Notes:

1. Class description: A Class is an STMicroelectronics internal specification. All its limits are higher than the JEDEC specifications, that means when a device belongs to Class A it exceeds the JEDEC standard. B Class strictly covers all the JEDEC criteria (international standard).

2. Schaffner NSG435 with a pointed test finger.

EMC CHARACTERISTICS (Cont'd)

12.7.4 ESD Pin Protection Strategy

To protect an integrated circuit against Electro-Static Discharge the stress must be controlled to prevent degradation or destruction of the circuit elements. The stress generally affects the circuit elements which are connected to the pads but can also affect the internal devices when the supply pads receive the stress. The elements to be protected must not receive excessive current, voltage or heating within their structure.

An ESD network combines the different input and output ESD protections. This network works, by allowing safe discharge paths for the pins subjected to ESD stress. Two critical ESD stress cases are presented in Figure 78 and Figure 79 for standard pins and in Figure 80 and Figure 81 for true open drain pins.
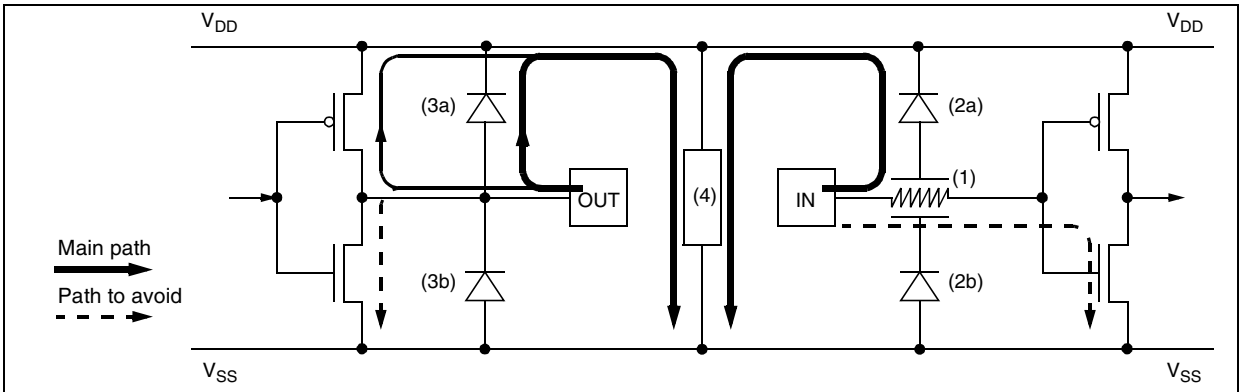
Standard Pin Protection

To protect the output structure the following elements are added:

  – A diode to V_{DD} (3a) and a diode from V_{SS} (3b)
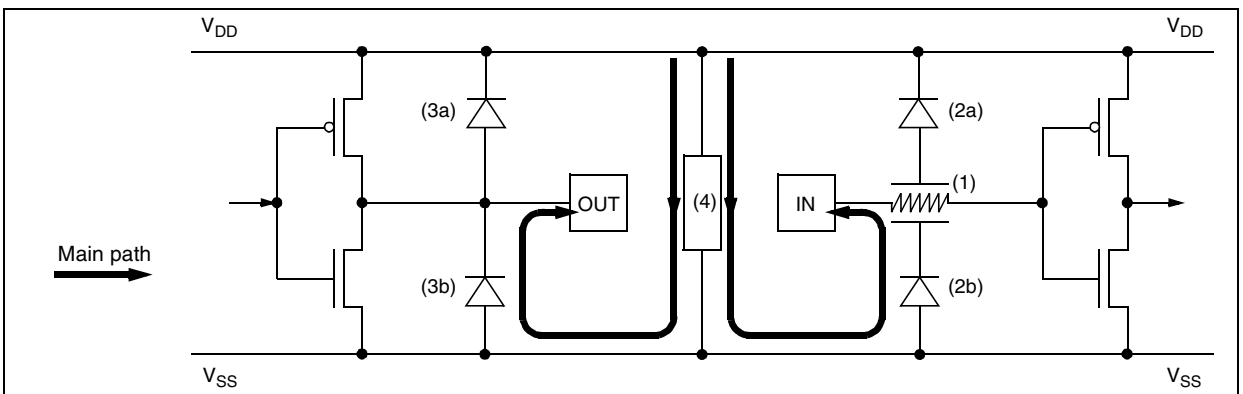 – A protection device between V_{DD} and V_{SS} (4)

To protect the input structure the following elements are added:

  – A resistor in series with the pad (1)
  – A diode to V_{DD} (2a) and a diode from V_{SS} (2b)
  – A protection device between V_{DD} and V_{SS} (4)

Figure 78. Positive Stress on a Standard Pad vs. V_{SS}

Figure 79. Negative Stress on a Standard Pad vs. V_{DD}

EMC CHARACTERISTICS (Cont'd)

True Open Drain Pin Protection

The centralized protection (4) is not involved in the discharge of the ESD stresses applied to true open drain pads due to the fact that a P-Buffer and diode to V_{DD} are not implemented. An additional local protection between the pad and V_{SS} (5a & 5b) is implemented to completely absorb the positive ESD discharge.

Multisupply Configuration

When several types of ground (V_{SS}, V_{SSA}, ...) and power supply (V_{DD}, V_{AREF}, ...) are available for any reason (better noise immunity...), the structure shown in Figure 82 is implemented to protect the device against ESD.
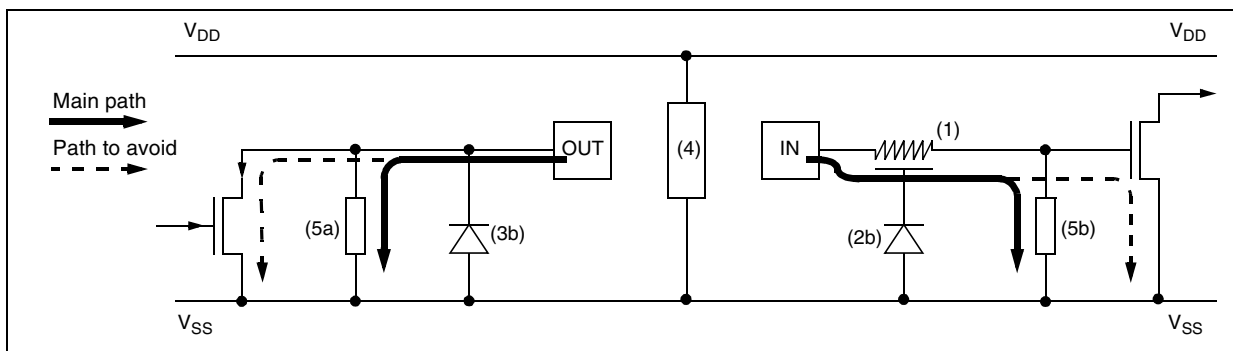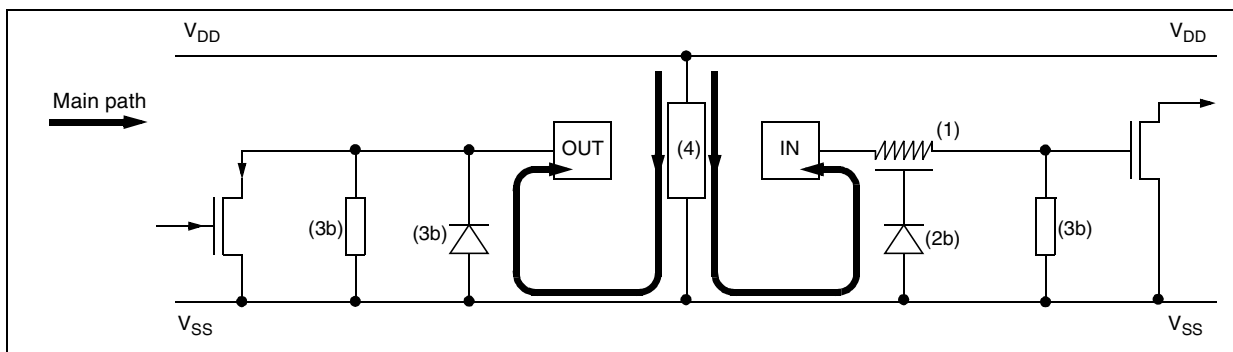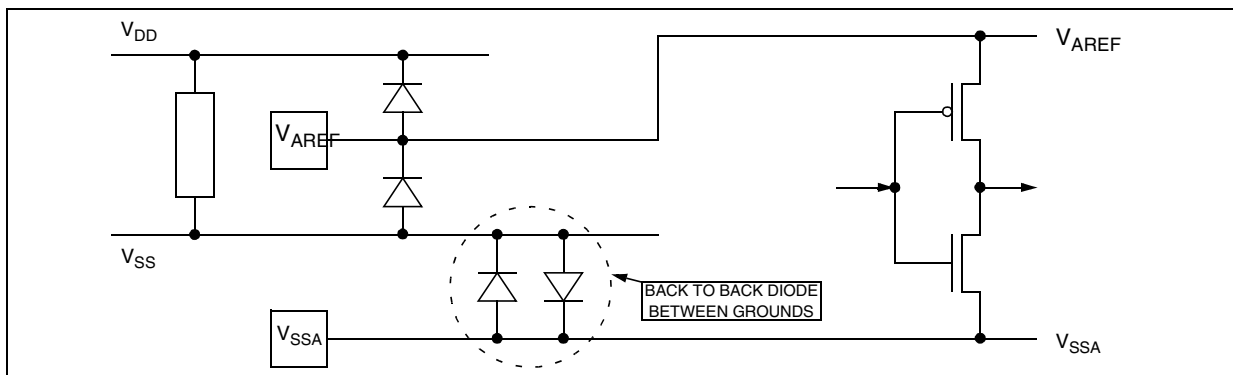
Figure 80. Positive Stress on a True Open Drain Pad vs. V_{SS}

Figure 81. Negative Stress on a True Open Drain Pad vs. V_{DD}


Figure 82. Multisupply Configuration

12.8 I/O PORT PIN CHARACTERISTICS

12.8.1 General Characteristics

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

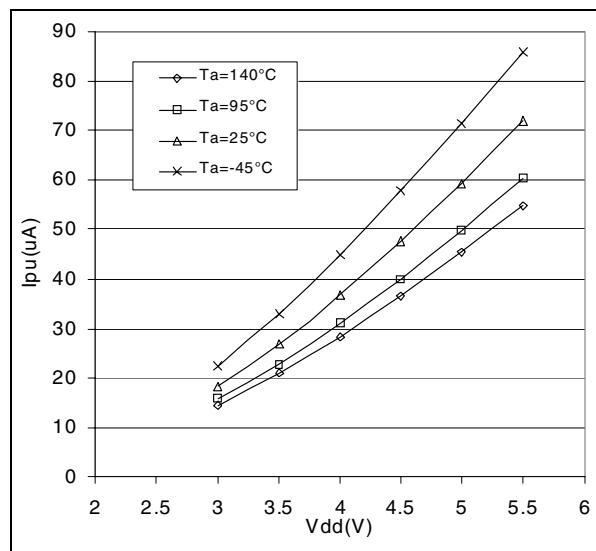| Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage [2]	CMOS ports				$0.3 \times V_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7 \times V_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
V_{IL}	Input low level voltage [2]	TTL ports				0.8	V
V_{IH}	Input high level voltage [2]			2			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
$I_{INJ(PIN)}$ [4]	Injected Curent on an IO pin	V_{DD}=5V				± 4	mA
$\Sigma I_{INJ(PIN)}$ [4]	Total injected current (sum of all I/O and control pins) [6]					± 25	
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$				±1	μA
I_S	Static current consumption [5]	Floating input mode				200	
R_{PU}	Weak pull-up equivalent resistor [6]	V_{IN}=V_{SS}	V_{DD}=5V	80	120	250	kΩ
			V_{DD}=3.3V	170	200	230	
C_{IO}	I/O pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time [7]	C_L=50pF Between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time [7]				25		
$t_{w(IT)in}$	External interrupt pulse time [8]			1			t_{CPU}

Figure 83. Unused I/O Pin Applications
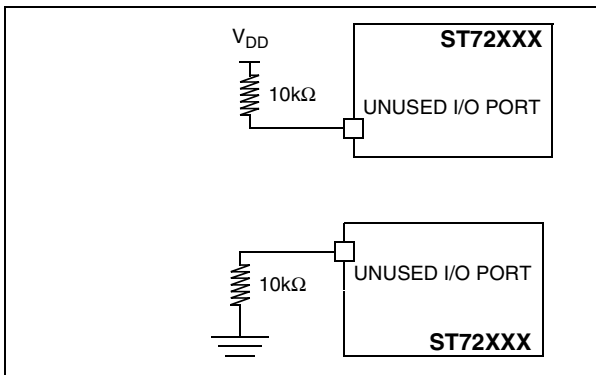
Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.

2. Data based on characterization results, not tested in production.

3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.

4. When the current limitation is not possible, the V_{IN} maximum must be respected, otherwise refer to $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. Refer to Section 12.2.2 on page 134 for more details.

5. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see Figure 83). Data based on design simulation and/or technology characteristics, not tested in production.

6. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in Figure 84). This data is based on characterization results, tested in production at V_{DD} max.

7. Data based on characterization results, not tested in production.

8. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 84. Typical I$_{PU}$ vs. V$_{DD}$ with V$_{IN}$=V$_{SS}$

I/O PORT PIN CHARACTERISTICS (Cont'd)

12.8.2 Output Driving Current

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} [1]	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 85)	I_{IO}=+5mA	V_{DD}=5V		1.2	V
		I_{IO}=+2mA			0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 86 and Figure 88)	I_{IO}=+20mA, T_A≤85°C $\quad T_A$≥85°C			1.3 1.5	
		I_{IO}=+8mA			0.6	
V_{OH} [2]	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 87 and Figure 90)	I_{IO}=-5mA, $\quad T_A$≤85°C $\quad\quad\quad T_A$≥85°C	V_{DD}-1.4 V_{DD}-1.6			
		I_{IO}=-2mA	V_{DD}-0.7			

Figure 85. Typical V_{OL} at V_{DD}=5V (standard)
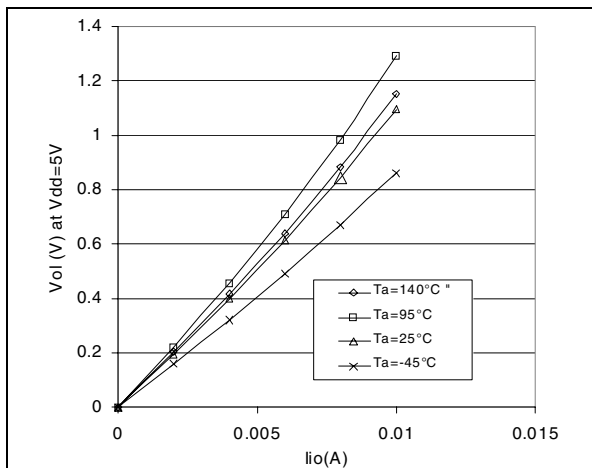
Figure 87. Typical V_{DD}-V_{OH} at V_{DD}=5V
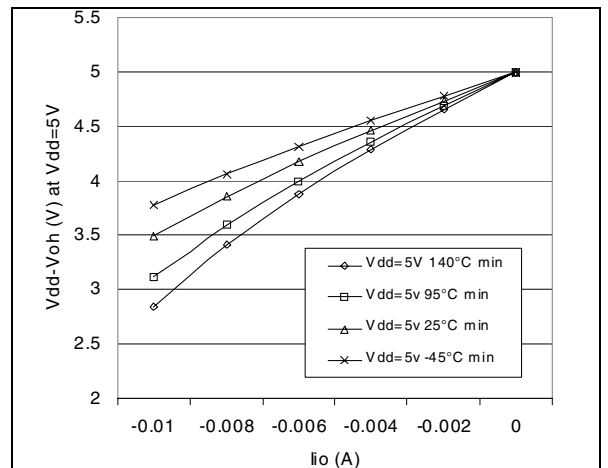
Figure 86. Typical V_{OL} at V_{DD}=5V (high-sink)
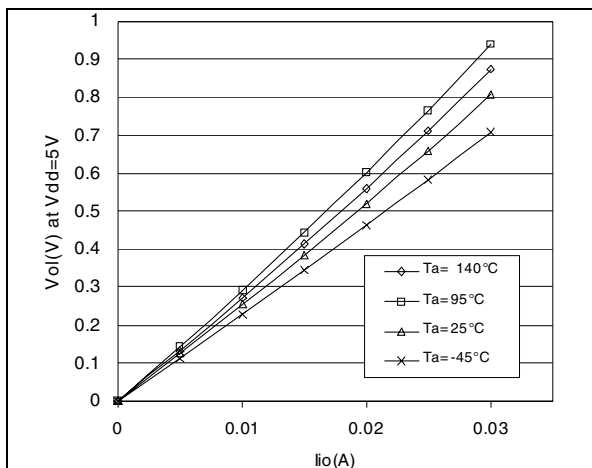
Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

2. The I_{IO} current sourced must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 88. Typical V$_{OL}$ vs. V$_{DD}$ (standard)
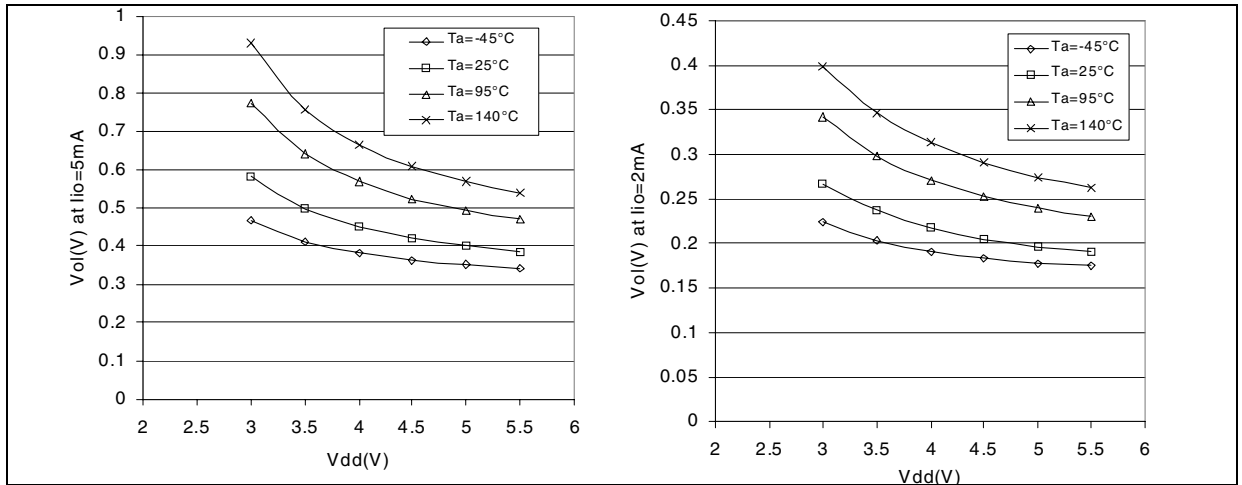
Figure 89. Typical V$_{OL}$ vs. V$_{DD}$ (high-sink)
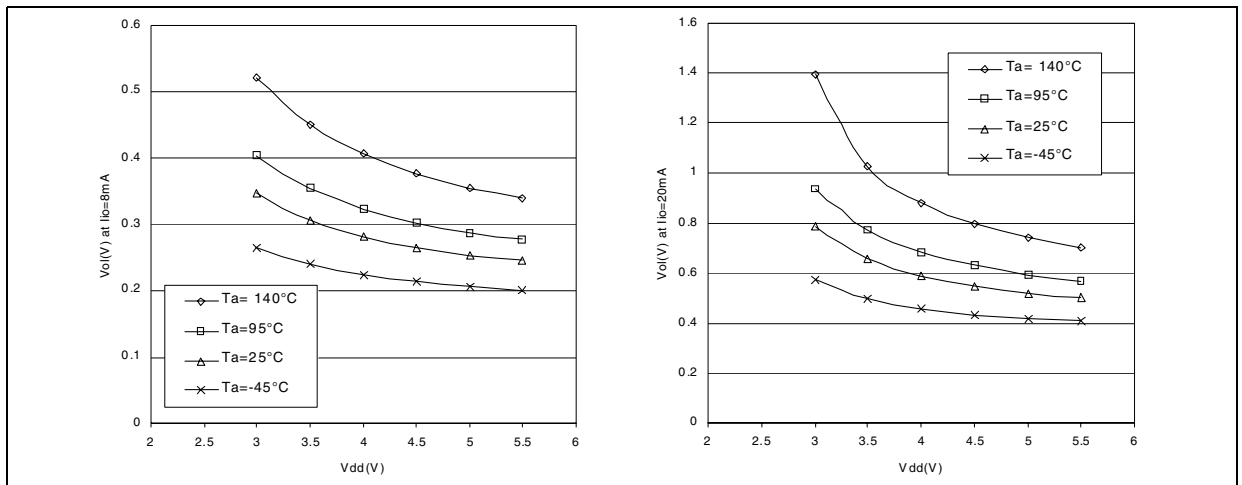
Figure 90. Typical V$_{DD}$-V$_{OH}$ vs. V$_{DD}$
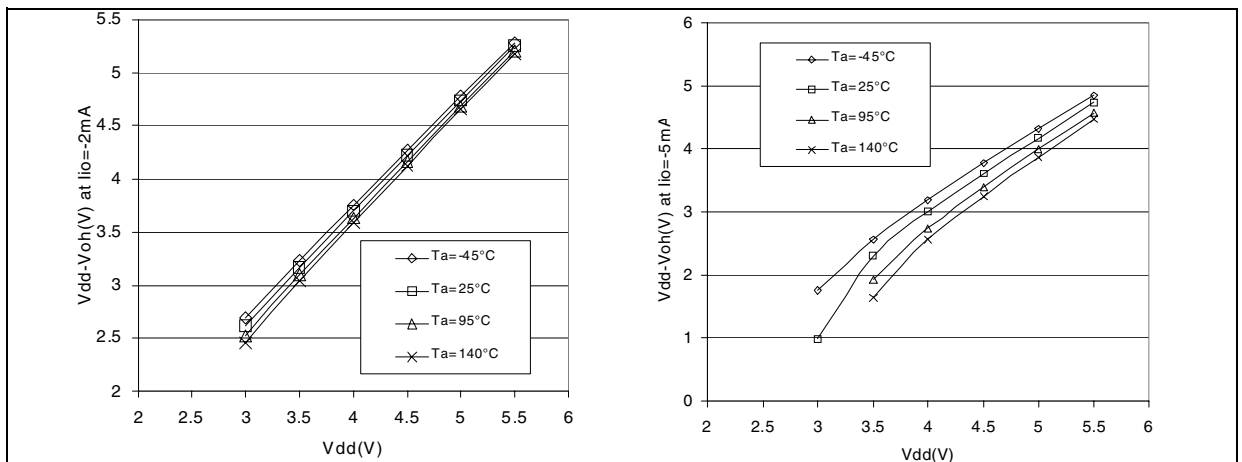
12.9 CONTROL PIN CHARACTERISTICS

12.9.1 Asynchronous $\overline{\text{RESET}}$ Pin

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage [2]					$0.3 \times V_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7 \times V_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
V_{OL}	Output low level voltage [4]	V_{DD}=5V	I_{IO}=+5mA		0.68	0.95	V
			I_{IO}=+2mA		0.28	0.45	
R_{ON}	Weak pull-up equivalent resistor [5]	V_{IN}=V_{SS}	V_{DD}=5V	20	40	60	kΩ
			V_{DD}=3.3V	80	100	120	
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or internal reset sources			30		µs
$t_{h(RSTL)in}$	External reset pulse hold time [6]			20			µs
$t_{g(RSTL)in}$	Filtered glitch duration [7]					100	ns

Figure 91. Typical Application with $\overline{\text{RESET}}$ pin [8]

Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.

2. Data based on characterization results, not tested in production.

3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
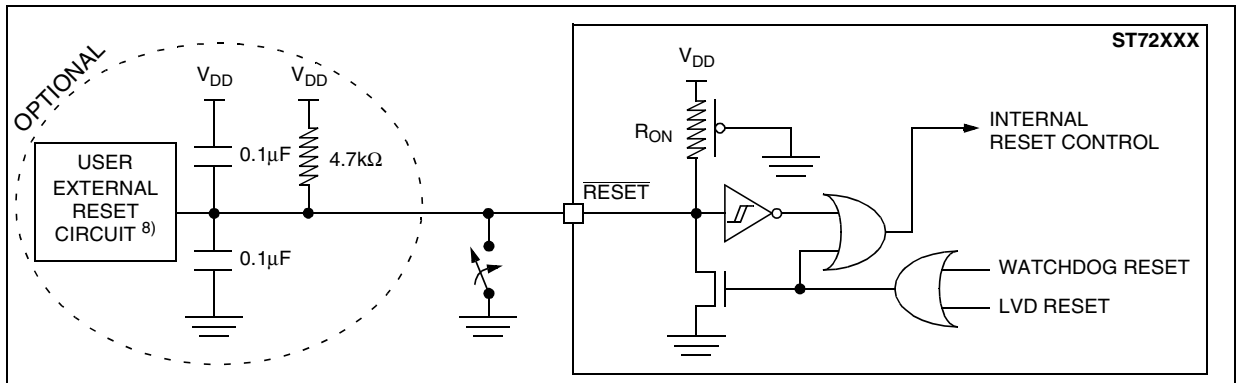
4. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

5. The R_{ON} pull-up equivalent resistor is based on a resistive transistor.

6. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

7. The reset network (the resistor and two capacitors) protects the device against parasitic resets, especially in a noisy environments.

8. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).

CONTROL PIN CHARACTERISTICS (Cont'd)

12.9.2 V_{PP} Pin

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage [1]		V_{SS}	0.2	V
V_{IH}	Input high level voltage [1]		V_{DD}-0.1	12.6	
I_L	Input leakage current	V_{IN}=V_{SS}		±1	µA

Figure 92. Two typical Applications with V_{PP} Pin [2]
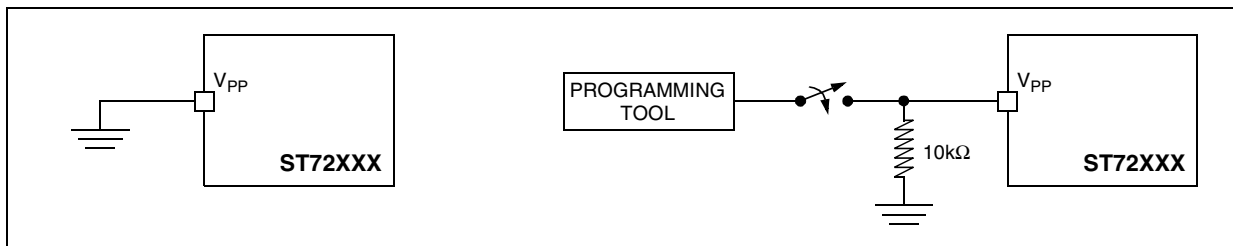
Notes:

1. Data based on design simulation and/or technology characteristics, not tested in production.
2. When the ICC mode is not required by the application V_{PP} pin must be tied to V_{SS}.

12.10 TIMER PERIPHERAL CHARACTERISTICS

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (output compare, input capture, external clock, PWM output...).

12.10.1 Watchdog Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(WDG)}$	Watchdog time-out duration		12,288		786,432	t_{CPU}
		f_{CPU}=8MHz	1.54		98.3	ms

12.10.2 8-Bit PWM-ART Auto-Reload Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{res(PWM)}$	PWM resolution time		1			t_{CPU}
		f_{CPU}=8MHz	125			ns
f_{EXT}	ART external clock frequency		0		f_{CPU}/2	MHz
f_{PWM}	PWM repetition rate		0		f_{CPU}/2	MHz
Res_{PWM}	PWM resolution				8	bit
V_{OS}	PWM/DAC output step voltage	V_{DD}=5V, Res=8-bits		20		mV

12.10.3 16-Bit Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(ICAP)in}$	Input capture pulse time		1			t_{CPU}
$t_{res(PWM)}$	PWM resolution time		2			t_{CPU}
		f_{CPU}=8MHz	250			ns
f_{EXT}	Timer external clock frequency		0		f_{CPU}/4	MHz
f_{PWM}	PWM repetition rate		0		f_{CPU}/4	MHz
Res_{PWM}	PWM resolution				16	bit

12.11 COMMUNICATION INTERFACE CHARACTERISTICS

12.11.1 SPI - Serial Peripheral Interface

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Symbol	Parameter	Conditions		Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master	f_{CPU}=8MHz	f_{CPU}/128 0.0625	f_{CPU}/4 2	MHz
		Slave	f_{CPU}=8MHz	0	f_{CPU}/2 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time			see I/O port pin description		
$t_{su(\overline{SS})}$	$\overline{SS}$ setup time	Slave		120		
$t_{h(\overline{SS})}$	$\overline{SS}$ hold time	Slave		120		
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master Slave		100 90		
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave		100 100		ns
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master Slave		100 100		
$t_{a(SO)}$	Data output access time	Slave		0	120	
$t_{dis(SO)}$	Data output disable time	Slave			240	
$t_{v(SO)}$	Data output valid time	Slave (after enable edge)			120	
$t_{h(SO)}$	Data output hold time			0		
$t_{v(MO)}$	Data output valid time	Master (before capture edge)		0.25		t_{CPU}
$t_{h(MO)}$	Data output hold time			0.25		

Figure 93. SPI Slave Timing Diagram with CPHA=0 [3]
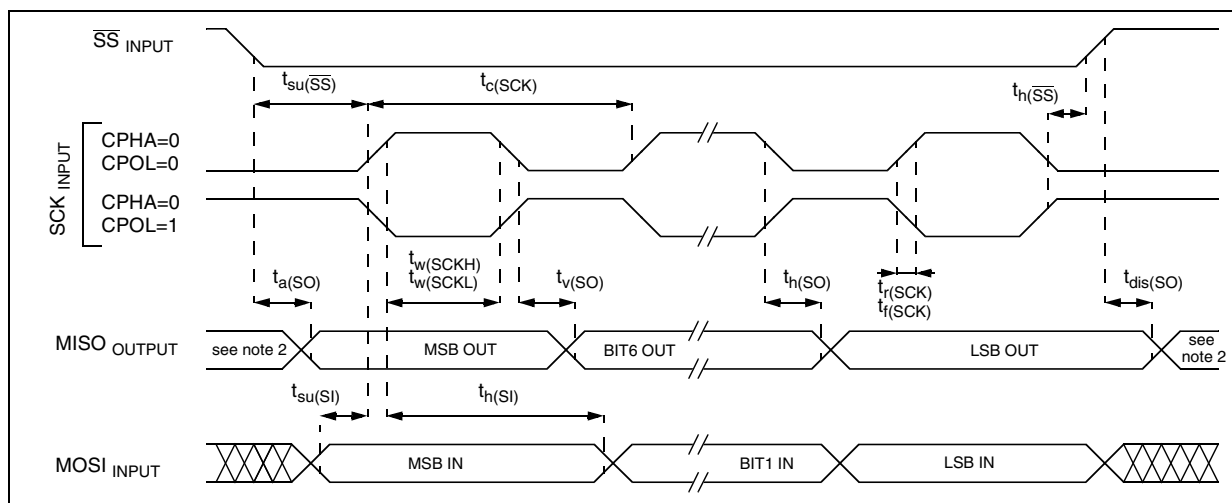
Notes:

1. Data based on design simulation and/or characterisation results, not tested in production.

2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

3. Measurement points are done at CMOS levels: 0.3xV_{DD} and 0.7xV_{DD}.

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

Figure 94. SPI Slave Timing Diagram with CPHA=1[1]
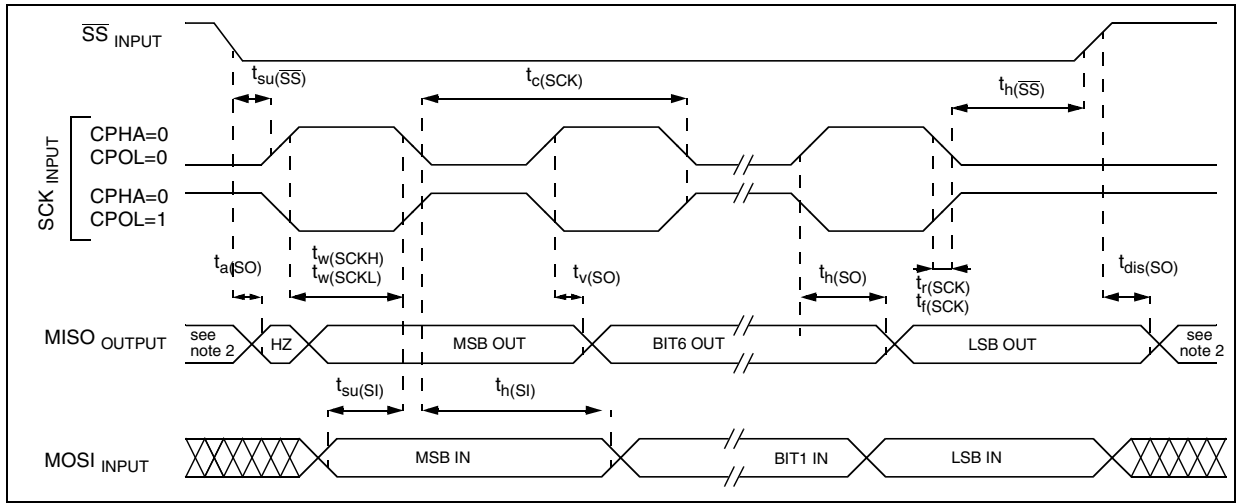
Figure 95. SPI Master Timing Diagram [1]
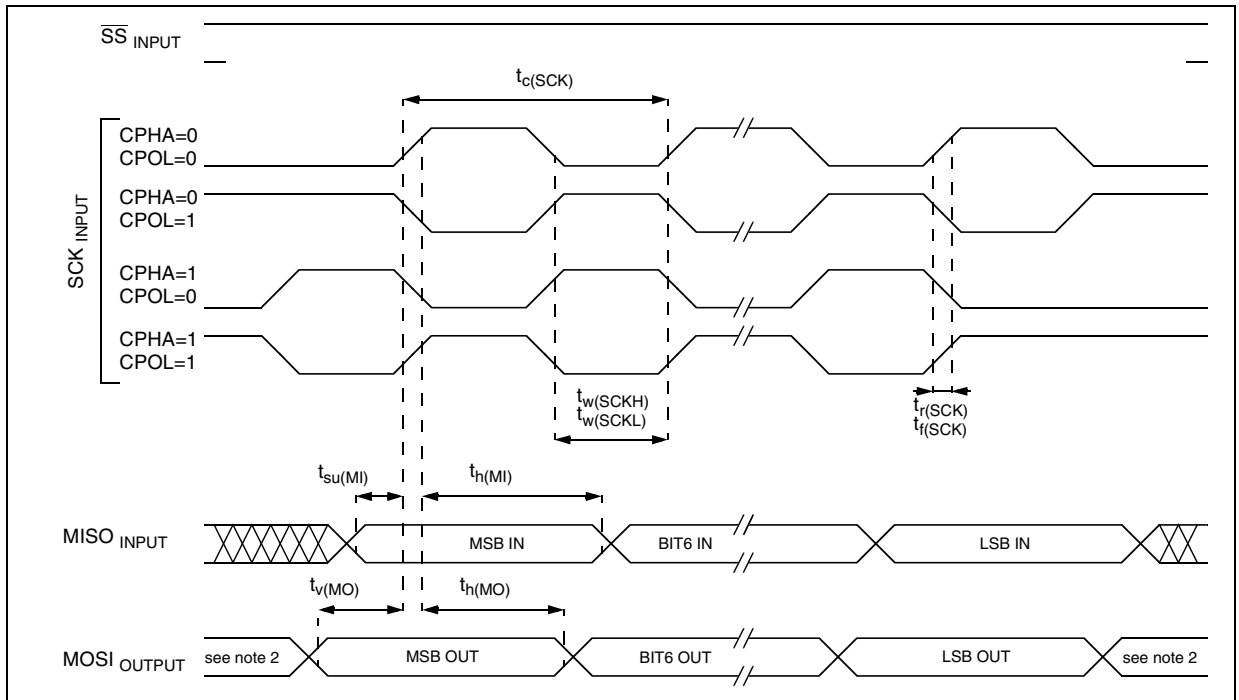
Notes:

1. Measurement points are done at CMOS levels: $0.3 \times V_{DD}$ and $0.7 \times V_{DD}$.

2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

**COMMUNICATIONS INTERFACE CHARACTERISTICS** (Cont'd)

12.11.2 SCI - Serial Communications Interface

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (RDI and TDO).

Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8MHz	~0.16%	Conventional Mode TR (or RR)=128, PR=13 TR (or RR)= 32, PR=13 TR (or RR)= 16, PR=13 TR (or RR)= 8, PR=13 TR (or RR)= 4, PR=13 TR (or RR)= 16, PR= 3 TR (or RR)= 2, PR=13 TR (or RR)= 1, PR=13	300 1200 2400 4800 9600 10400 19200 38400	~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77 ~38461.54	Hz
			~0.79%	Extended Mode ETPR (or ERPR) = 35, TR (or RR)= 1, PR=1	14400	~14285.71	

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

12.11.3 I²C - Inter IC Control Interface

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (SDAI and SCLI). The ST7 I²C interface meets the requirements of the Standard I²C communication protocol described in the following table.

Symbol	Parameter	Standard mode I²C		Fast mode I²C		Unit
		Min [1]	Max [1]	Min [1]	Max [1]	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		μs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		
$t_{h(SDA)}$	SDA data hold time	0 [3]		0 [2]	900 [3]	
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	$20+0.1C_b$	300	ns
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300	$20+0.1C_b$	300	
$t_{h(STA)}$	START condition hold time	4.0		0.6		μs
$t_{su(STA)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(STO)}$	STOP condition setup time	4.0		0.6		ns
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		ms
C_b	Capacitive load for each bus line		400		400	pF

Figure 96. Typical Application with I²C Bus and Timing Diagram [4]
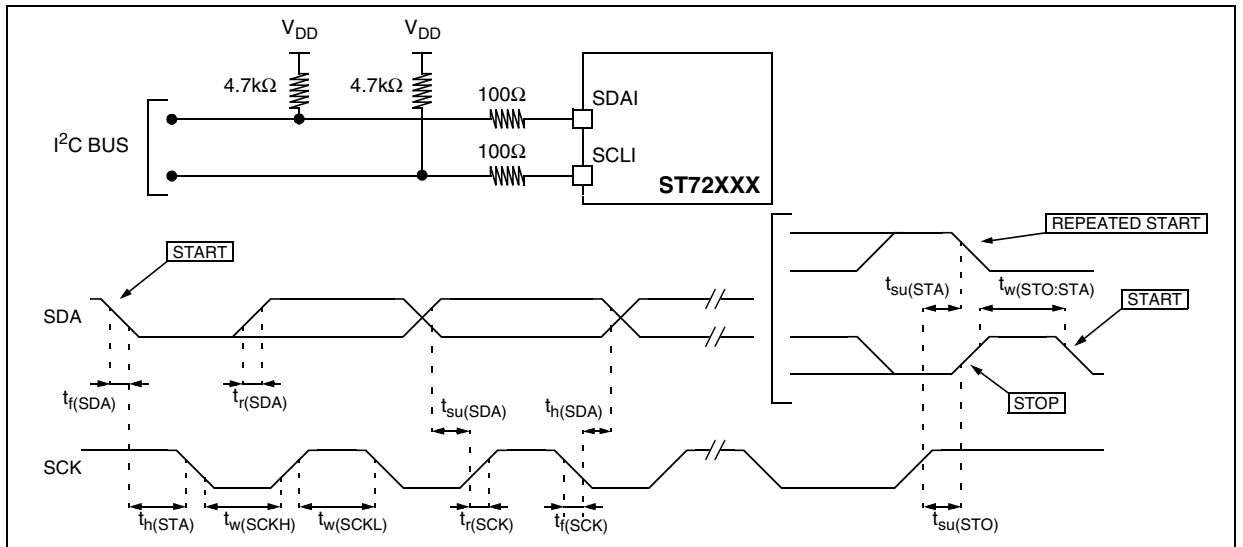
Notes:

1. Data based on standard I²C protocol requirement, not tested in production.

2. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.

3. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.

4. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

12.12 10-BIT ADC CHARACTERISTICS

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion voltage range [2]		V_{SSA}		V_{AREF}	V
R_{AIN}	External input resistor				10 [3]	kΩ
C_{ADC}	Internal sample and hold capacitor			6		pF
t_{STAB}	Stabilization time after ADC enable	f_{CPU}=8MHz, f_{ADC}=4MHz		0 [4]		μs
t_{ADC}	Conversion time (Sample+Hold)			7.25		
	- Sample capacitor loading time - Hold conversion time			8 21		$1/f_{ADC}$

Figure 97. Typical Application with ADC
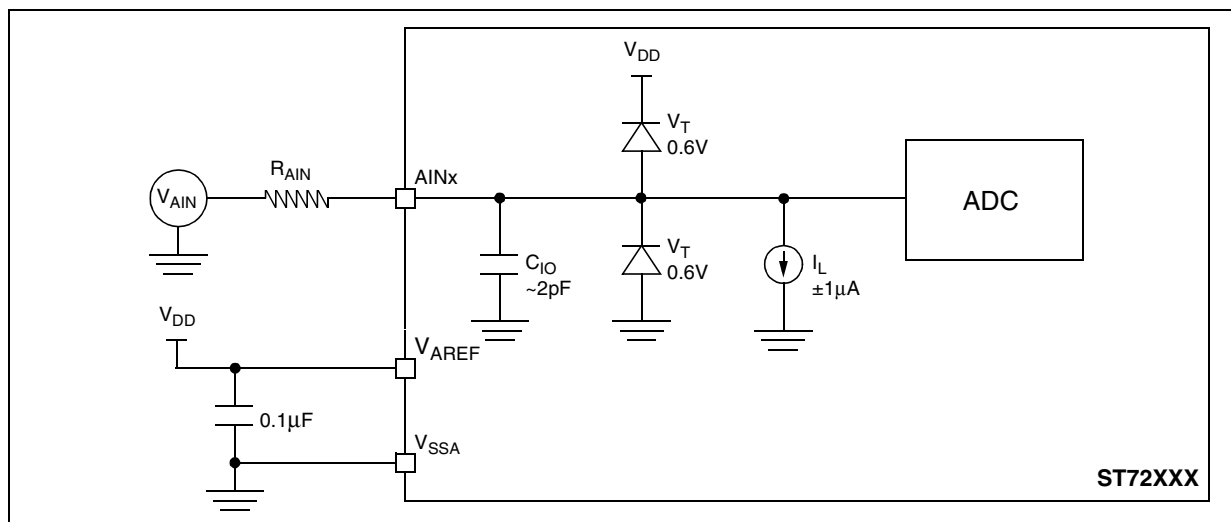
Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2. When V_{AREF} and V_{SSA} pins are not available on the pinout, the ADC refer to V_{DD} and V_{SS}.

3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.

4. The stabilization time of the AD converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

10-BIT ADC CHARACTERISTICS (Cont'd)

ADC Accuracy

Symbol	Parameter	Conditions ⇒ ⇓	V_{DD}=5.5V, [2) f_{CPU}=1MHz		V_{DD}=5.0V, [3) f_{CPU}=8MHz		V_{DD}=3.3V, [3) f_{CPU}=8MHz	
			Min	**Max**	**Min**	**Max**	**Min**	**Max**
$|E_T|$	Total unadjusted error [1)	V_{SS}+0.5V≤V_{IN}≤V_{DD}-0.5V		TBD		4		TBD
		E_O software compensated		TBD		TBD		TBD
				TBD		TBD		TBD
E_O	Offset error [1)		TBD	TBD	2.5	3.5	TBD	TBD
E_G	Gain Error [1)		TBD	TBD	0	2	TBD	TBD
$|E_D|$	Differential linearity error [1)			TBD		4.5		TBD
$|E_L|$	Integral linearity error [1)			TBD		4.5		TBD

Figure 98. ADC Accuracy Characteristics
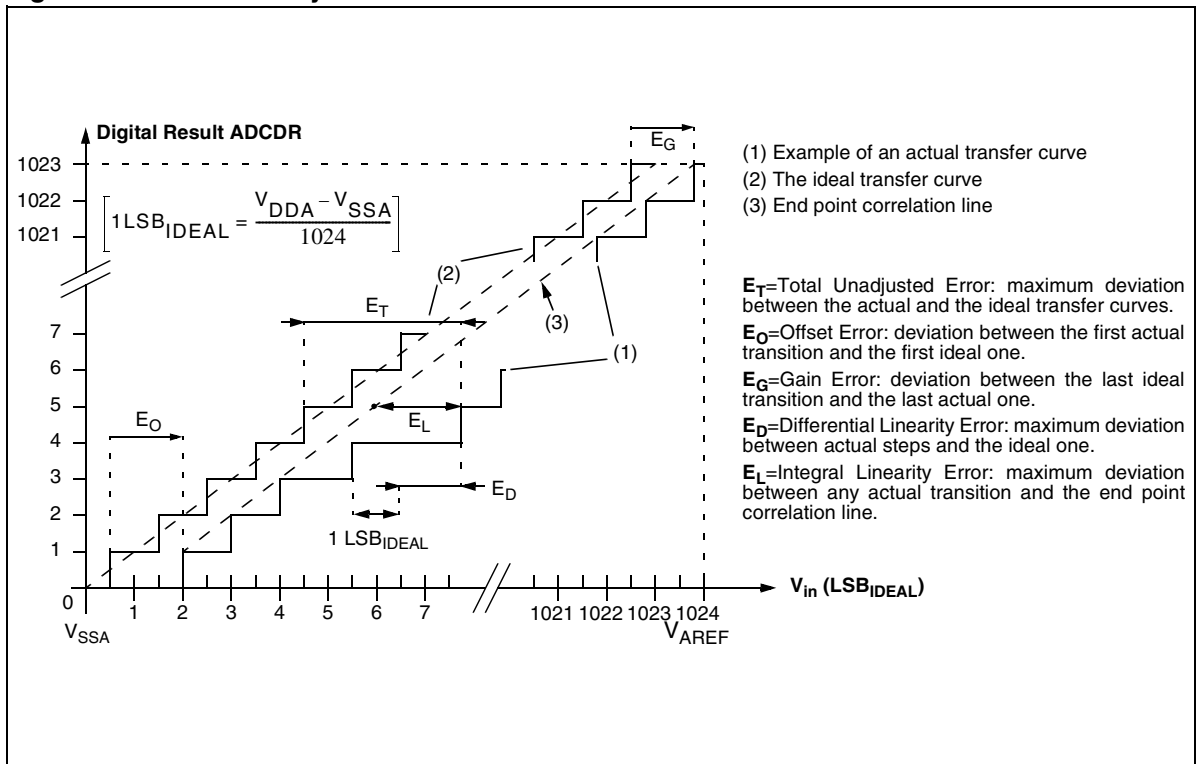
Notes:

1. ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 4 LSB for each 10KΩ increase of the external analog source impedance. This effect on the ADC accuracy has been observed under worst-case conditions for injection:
- negative injection
- injection to an Input with analog capability, adjacent to the enabled Analog Input
- at 5V V_{DD} supply, and worst case temperature.

2. Data based on characterization results with T_A=25°C.

3. Data based on characterization results over the whole temperature range, monitored in production.

13 PACKAGE CHARACTERISTICS

13.1 PACKAGE MECHANICAL DATA

Figure 99. 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					
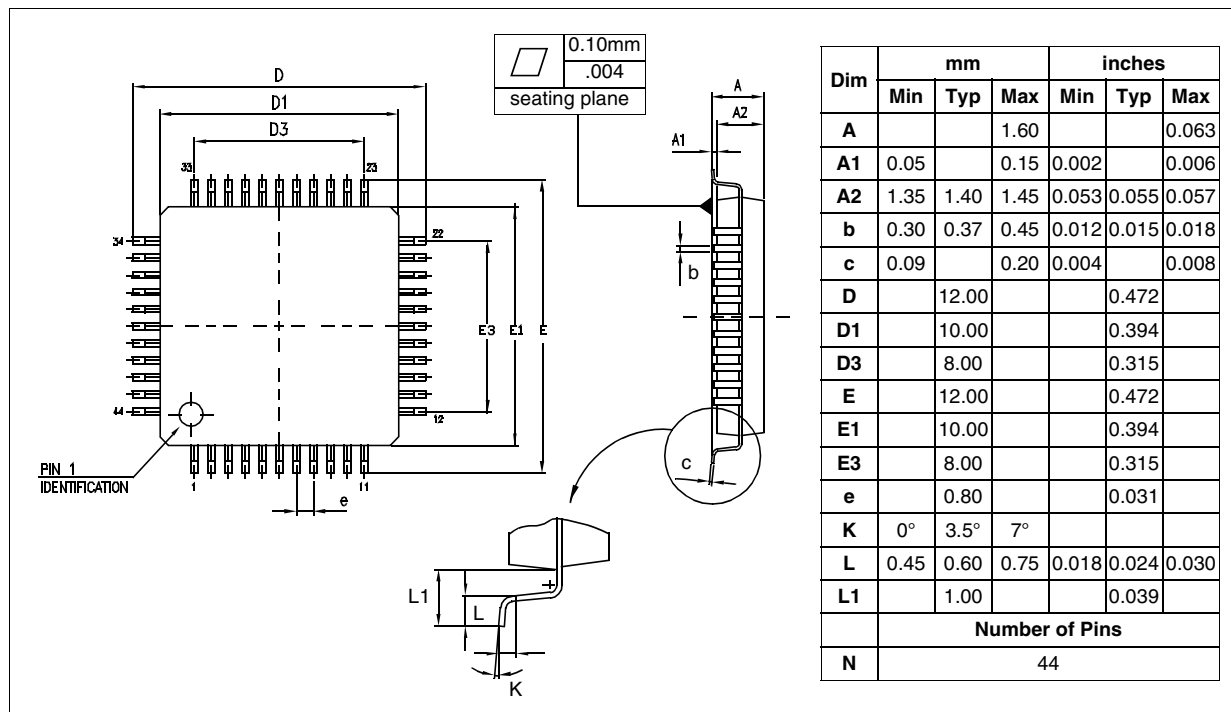
PACKAGE CHARACTERISTICS (Cont'd)

Figure 100. 42-Pin Shrink Plastic Dual In-Line Package, 600-mil Width

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	0.00		1.52	0.000		0.060
L	2.54	3.30	3.56	0.100	0.130	0.140
	Number of Pins					
N	42					

PDIP42S
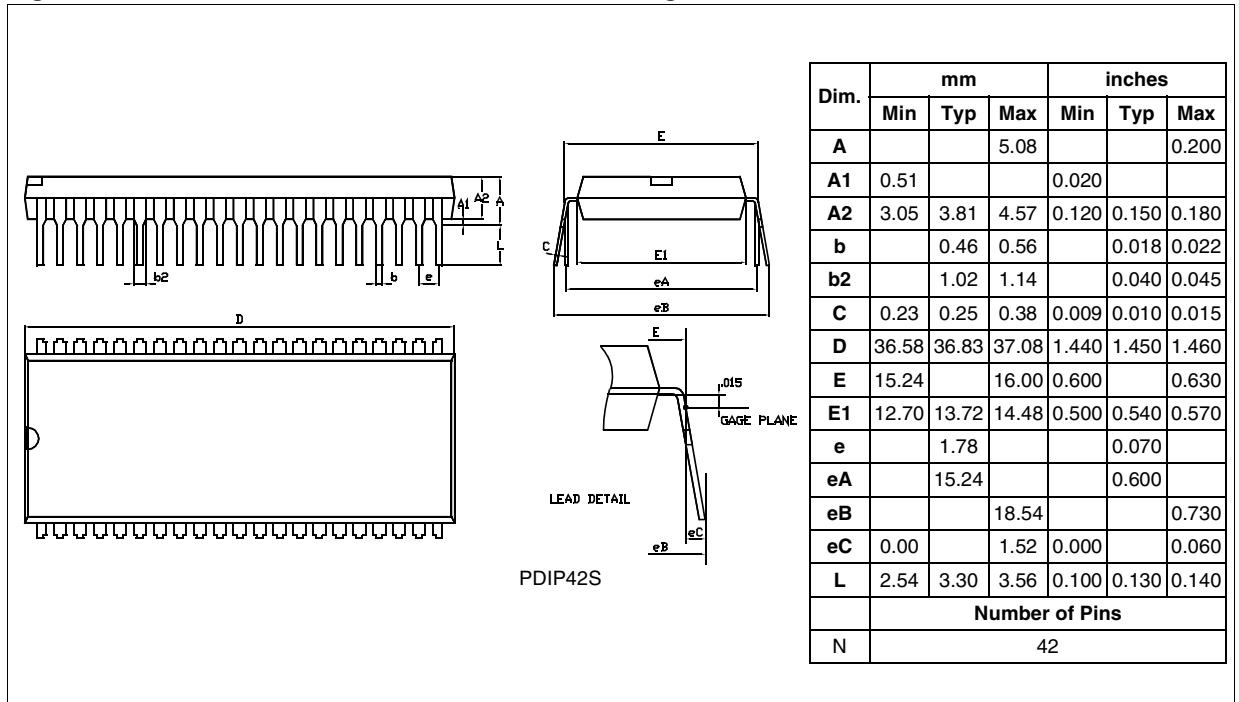
Figure 101. 32-Pin Thin Quad Flat Package

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		9.00			0.354	
D1		7.00			0.276	
D3		5.60			0.220	
e		0.80			0.031	
E		9.00			0.354	
E1		7.00			0.276	
E3		5.60			0.220	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
K	0°		7°			
	Number of Pins					
N	32					
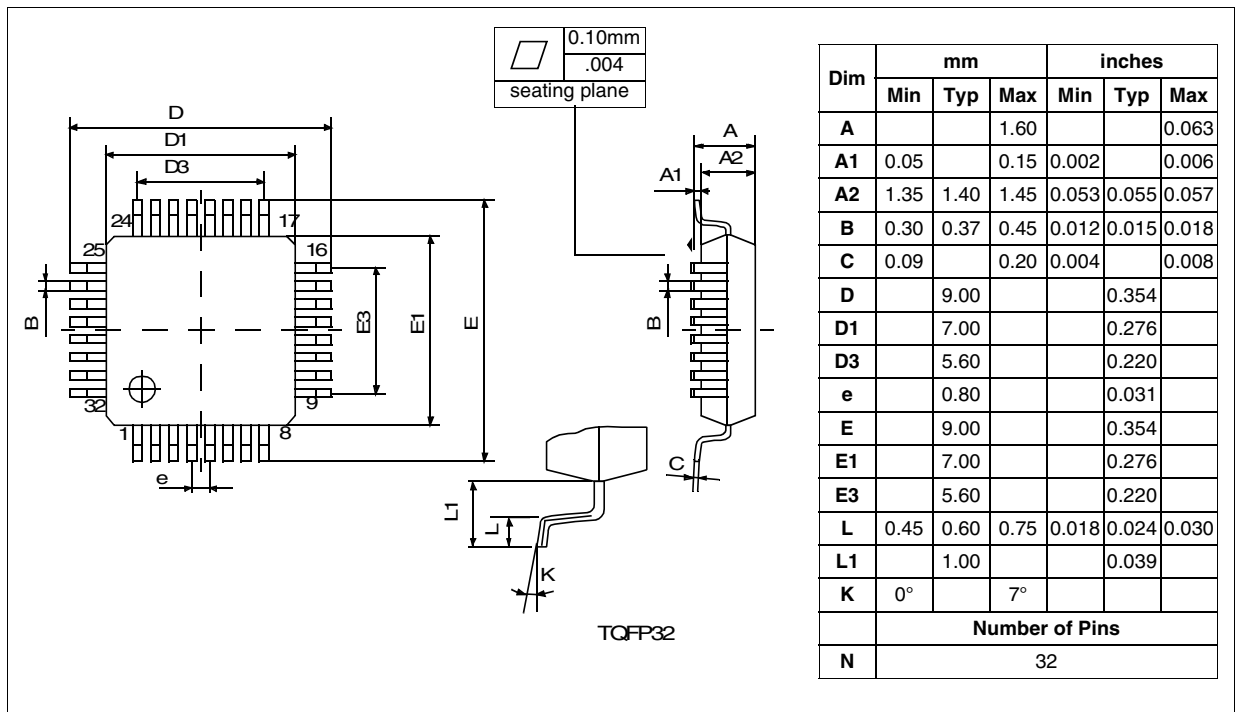
TQFP32

PACKAGE CHARACTERISTICS (Cont'd)

Figure 102. 32-Pin Shrink Plastic Dual In Line Package

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.140	0.148	0.200
A1	0.51			0.020		
A2	3.05	3.56	4.57	0.120	0.140	0.180
b	0.36	0.46	0.58	0.014	0.018	0.023
b1	0.76	1.02	1.40	0.030	0.040	0.055
C	0.20	0.25	0.36	0.008	0.010	0.014
D	27.43	27.94	28.45	1.080	1.100	1.120
E	9.91	10.41	11.05	0.390	0.410	0.435
E1	7.62	8.89	9.40	0.300	0.350	0.370
e		1.78			0.070	
eA		10.16			0.400	
eB			12.70			0.500
L	2.54	3.05	3.81	0.100	0.120	0.150
	Number of Pins					
N	32					
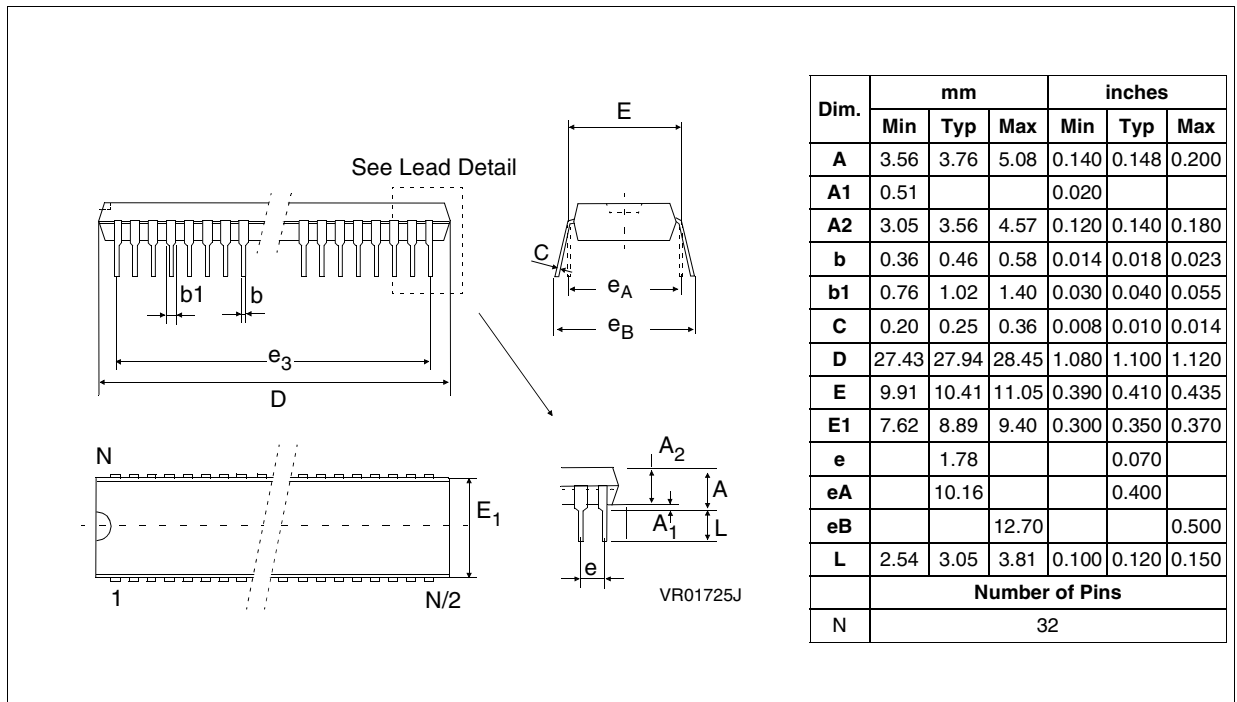
13.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) TQFP44 10x10 TQFP32 7x7 SDIP42 600mil SDIP32 200mil	TBD TBD TBD TBD	°C/W
P_D	Power dissipation [1]	500	mW
T_{Jmax}	Maximum junction temperature [2]	150	°C

Notes:

1. The power dissipation is obtained from the formula $P_D=P_{INT}+P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD}xV_{DD}$) and P_{PORT} is the port power dissipation determined by the user.

2. The average chip-junction temperature can be obtained from the formula $T_J = T_A + P_D$ x RthJA.

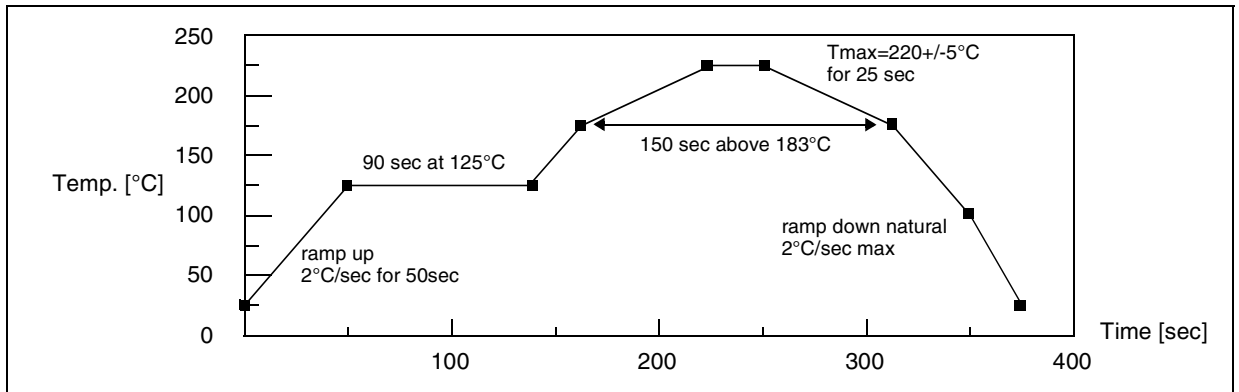## 13.3 SOLDERING AND GLUEABILITY INFORMATION

Recommended soldering information given only as design guidelines.

Figure 103. Recommended Wave Soldering Profile (with 37% Sn and 63% Pb)
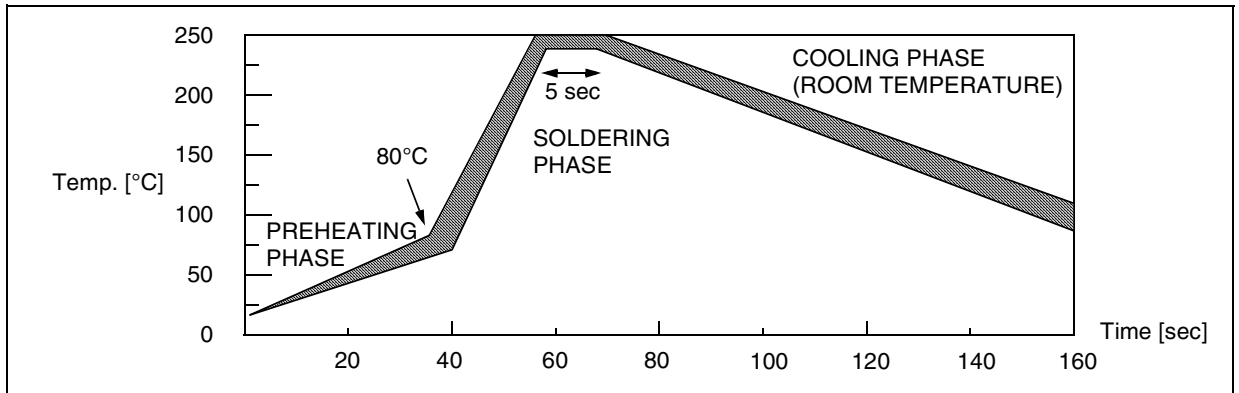
Figure 104. Recommended Reflow Soldering Oven Profile (MID JEDEC)

Recommended glue for SMD plastic packages dedicated to molding compound with silicone:

■ Heraeus: PD945, PD955
■ Loctite: 3615, 3298

14 ST72321J/K DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user pro-grammable versions (FLASH) as well as in factory coded versions (ROM). FLASH devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer us-ing the Option Bytes while the ROM devices are factory-configured.

14.1 FLASH OPTION BYTES

STATIC OPTION BYTE 2								STATIC OPTION BYTE 1							
15							8	7							0
PKG32P	RSTC	OSCTYPE		OSCRANGE			PLLOFF	WDG		CSS	VD				FMP_R
		1	0	2	1	0		HALT	SW		1	0			
Default	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

The option bytes allows the hardware configura-tion of the microcontroller to be selected. They have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The de-fault content of the FLASH is fixed to FFh. To pro-gram directly the FLASH devices using ICP, FLASH devices are shipped to customers with the internal RC clock source. In masked ROM devic-es, the option bytes are fixed in hardware by the ROM code (see option list).

OPT15= **PKG32P** *32 Pin package selection*
This option bit selects the 32 pin package.

Version	Selected Package	PKG
J	TQFP44 / SDIP42	1
K	TQFP32 / SDIP32	0

OPT14 = **RSTC** *RESET clock cycle selection*
This option bit selects the number of CPU cycles applied during the RESET phase and when exiting HALT mode. For resonator oscillators, it is advised to select 4096 due to the long crystal stabilization time.
0: Reset phase with 4096 CPU cycles
1: Reset phase with 256 CPU cycles
Note: when the CSS is enabled, the device starts to count immediately thanks to the backup oscilla-tor.

OPT13:12 = **OSCTYPE[1:0]** *Oscillator Type*
These option bits select the ST7 main clock source type.

Clock Source	OSCTYPE	
	1	0
Resonator Oscillator	0	0
External RC Oscillator	0	1
Internal RC Oscillator	1	0
External Source	1	1

OPT11:9 = **OSCRANGE[2:0]** *Oscillator range*
When the resonator oscillator type is selected, these option bits select the resonator oscillator current source. This selection corresponds to used resonator frequency range. Otherwise, these bits are used to select the normal operating frequency range (see CSS option description).

Typ. Freq. Range		OSCRANGE		
		2	1	0
VLP	32~100kHz	1	x	x
LP	1~2MHz	0	0	0
MP	2~4MHz	0	0	1
MS	4~8MHz	0	1	0
HS	8~16MHz	0	1	1

ST72321J/K DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

OPT13 = **PLL OFF** *PLL activation*
This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL is guaranteed only with an input frequency between 2 and 4MHz.
0: PLL x2 enabled
1: PLL x2 disabled
CAUTION: the PLL can be enabled only if the "OSC RANGE" (OPT11:9) bits are configured to "MP - 2~4MHz". Otherwise, the device functionality is not guaranteed.

OPT7= **WDG HALT** *Watchdog and HALT mode*
This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

OPT6= **WDG SW** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

OPT5 = **CSS** *Clock security system on/off*
This option bit enables or disables the clock security system function (CSS) which include the clock filter and the backup safe oscillator.
0: CSS enabled
1: CSS disabled
When enabled, the selected "OSC RANGE" is used to determine which is the selected operating frequency range to switch to the backup oscillator when the frequency is out of range.

OPT4:3= **VD[1:0]** *Voltage detection*
These option bits enable the voltage detection block (LVD, and AVD) with a selected threshold for

Typ. Freq. Range	OSCRANGE			Compared frequencies to decide to switch to the backup oscillator	
	2	**1**	**0**		
32~100kHz	1	x	x	f_{CPU}	$f_{SFOSC}/64$
1~2MHz	0	0	0	2 x f_{CPU}	f_{SFOSC}
2~4MHz	0	0	1	f_{CPU}	
4~8MHz	0	1	0	$f_{CPU}/2$	
8~16MHz	0	1	1	$f_{CPU}/4$	

Selected Low Voltage Detector	VD1	VD0
LVD and AVD Off	1	1
Lowest Voltage Threshold (V_{DD}~3.5V)	1	0
Medium Voltage Threshold (V_{DD}~4V)	0	1
Highest Voltage Threshold (V_{DD}~5V)	0	0

OPT0= **FMP_R** *Flash memory read-out protection*
This option indicates if the user flash memory is protected against read-out piracy. This protection is based on read and a write protection of the memory in test modes and ICP mode. Erasing the option bytes when the FMP_R option is selected induce the whole user memory erasing first.
0: read-out protection disabled
1: read-out protection enabled

ST72321J/K DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE

Customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.
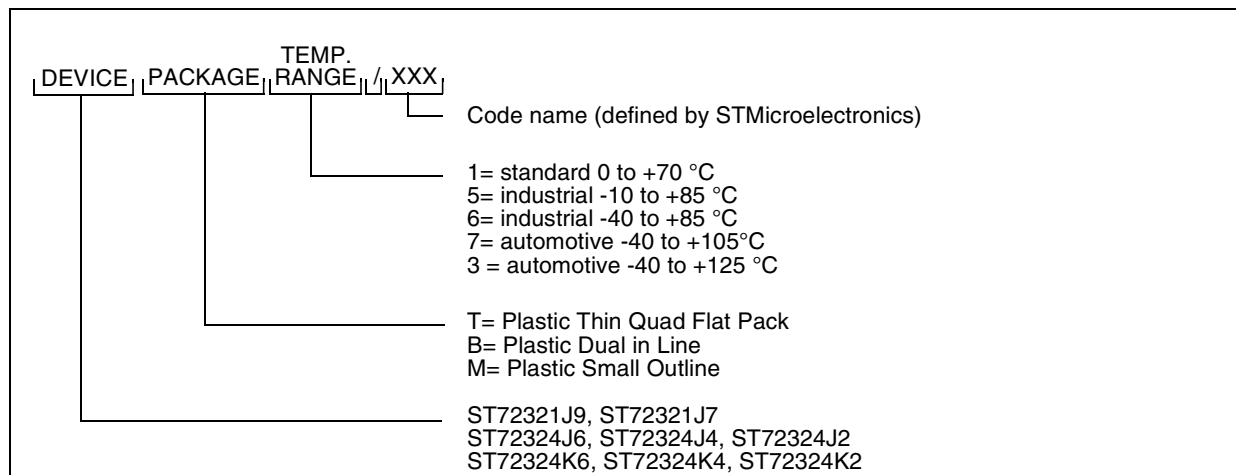
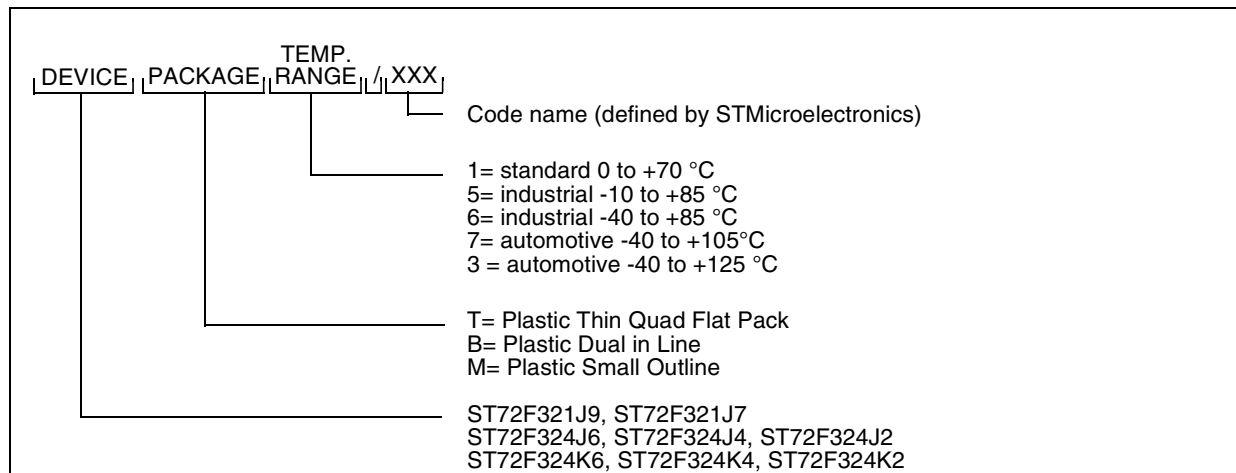**Figure 105. ROM Factory Coded Device Types**

Figure 106. FLASH User Programmable Device Types

ST72321J/K DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

<div style="border:1px solid">

<p align="center">**MICROCONTROLLER OPTION LIST**</p>

Customer . . . .
Address . . . .
. . . .
Contact . . . .
Phone No . . . .
Reference . . . .

STMicroelectronics references

Device: [ ] ST72321J9 [ ] ST72321J7
[ ] ST72324J6 [ ] ST72324J4 [ ] ST72324J2
[ ] ST72324K6 [ ] ST72324K4 [ ] ST72324K2

Package: [ ] TQFP44 10x10 [ ] SDIP42 600mil
[ ] TQFP32 7x7 [ ] SDIP32 200mil

Conditioning: [ ] Tube [ ] Tape & Reel (not available for SDIP packages)

Temperature Range: [ ] 0°C to + 70°C [ ] - 40°C to + 105°C
[ ] - 10°C to + 85°C [ ] - 40°C to + 125°C
[ ] - 40°C to + 85°C

Clock Source Selection: [ ] Resonator: [ ] VLP: V. Low power resonator (32 to 100KHz)
[ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] RC Network: [ ] Internal
[ ] External
[ ] External Clock
PLL [ ] Disabled [ ] Enabled

Clock Security System: [ ] Disabled [ ] Enabled

Watchdog Selection: [ ] Software Activation [ ] Hardware Activation
Halt when Watchdog on: [ ] Reset [ ] No reset

Readout Protection: [ ] Disabled [ ] Enabled

LVD Reset [ ] Disabled [ ] Enabled: [ ] Highest threshold
[ ] Medium threshold
[ ] Lowest threshold

Reset Delay [ ] 256 Cycles [ ] 4096 Cycles

Comments : . . . .
. . . .
. . . .
Supply Operating Range in the application: . . . .
Notes . . . .
. . . .
. . . .
Signature . . . .
Date . . . .

</div>

DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.3 DEVELOPMENT TOOLS

STmicroelectronics offers a range of hardware and software development tools for the ST7 microcontroller family. Full details of tools available for the ST7 from third party manufacturers can be obtain from the STMicroelectronics Internet site:
➡ http//mcu.st.com.

Tools from these manufacturers include C compliers, emulators and gang programmers.

STMicroelectronics Tools

Three types of development tool are offered by ST, all of them connect to a PC via a parallel (LPT) port: see Table 26 and Table 27 for more details.

Table 26. STMicroelectronics Tool Features

	In-Circuit Emulation	**Programming Capability**[1]	**Software Included**
ST7 Development Kit	Yes. (Same features as HDS2 emulator but without logic analyzer)	Yes (DIP packages only)	ST7 CD ROM with: – ST7 Assembly toolchain – STVD7 powerful Source Level Debugger for Win 3.1, Win 9x and NT – C compiler demo versions – ST Realizer for Win 3.1 and Win 95. – Windows Programming Tools for Win 3.1, Win 9x and NT
ST7 HDS2 Emulator	Yes, powerful emulation features including trace/ logic analyzer	No	
ST7 Programming Board	No	Yes (All packages)	

Table 27. Dedicated STMicroelectronics Development Tools

Supported Products	**ST7 Development Kit**	**ST7 HDS2 Emulator**	**ST7 Programming Board**
ST72321J, ST72F321J ST72324J, ST72F324J ST72324K, ST72F324K	N/A	ST7MDT20-EMU2B	ST7MDT20-EPB2/EU ST7MDT20-EPB2/US ST7MDT20-EPB2/UK

Note:
1. Flash Programming interface for FLASH devices.

14.4 ST7 APPLICATION NOTES

IDENTIFICATION	DESCRIPTION
PROGRAMMING AND TOOLS	
AN985	EXECUTING CODE IN ST7 RAM
AN986	USING THE ST7 INDIRECT ADDRESSING MODE
AN987	ST7 IN-CIRCUIT PROGRAMMING
AN988	STARTING WITH ST7 ASSEMBLY TOOL CHAIN
AN989	STARTING WITH ST7 HIWARE C
AN1039	ST7 MATH UTILITY ROUTINES
AN1064	WRITING OPTIMIZED HIWARE C LANGUAGE FOR ST7
AN1106	TRANSLATING ASSEMBLY CODE FROM HC05 TO ST7
EXAMPLE DRIVERS	
AN969	ST7 SCI COMMUNICATION BETWEEN THE ST7 AND A PC
AN970	ST7 SPI COMMUNICATION BETWEEN THE ST7 AND E²PROM
AN971	ST7 I²C COMMUNICATION BETWEEN THE ST7 AND E²PROM
AN972	ST7 SOFTWARE SPI MASTER COMMUNICATION
AN973	SCI SOFTWARE COMMUNICATION WITH A PC USING ST72251 16-BIT TIMER
AN974	REAL TIME CLOCK WITH THE ST7 TIMER OUTPUT COMPARE
AN976	DRIVING A BUZZER USING THE ST7 PWM FUNCTION
AN979	DRIVING AN ANALOG KEYBOARD WITH THE ST7 ADC
AN980	ST7 KEYPAD DECODING TECHNIQUES, IMPLEMENTING WAKE-UP ON KEYSTROKE
AN1017	USING THE ST7 USB MICROCONTROLLER
AN1041	USING ST7 PWM SIGNAL TO GENERATE ANALOG OUTPUT (SINUSOID)
AN1042	ST7 ROUTINE FOR I²C SLAVE MODE MANAGEMENT
AN1044	MULTIPLE INTERRUPT SOURCES MANAGEMENT FOR ST7 MCUS
AN1045	ST7 SOFTWARE IMPLEMENTATION OF I²C BUS MASTER
AN1046	ST7 UART EMULATION SOFTWARE
AN1047	MANAGING RECEPTION ERRORS WITH THE ST7 SCI PERIPHERAL
AN1048	ST7 SOFTWARE LCD DRIVER
AN1078	ST7 TIMER PWM DUTY CYCLE SWITCH FOR TRUE 0% or 100% DUTY CYCLE
AN1082	DESCRIPTION OF THE ST72141 MOTOR CONTROL
AN1083	ST72141 BLDC MOTOR CONTROL SOFTWARE AND FLOWCHART EXAMPLE
AN1129	PWM MANAGEMENT FOR BLDC MOTOR DRIVES USING THE ST72141
AN1130	BRUSHLESS DC MOTOR DRIVE WITH ST72141
AN1148	USING THE ST7263 FOR DESIGNING A USB MOUSE
AN1149	HANDLING SUSPEND MODE ON A USB MOUSE
AN1180	USING THE ST7263 KIT TO IMPLEMENT A USB GAME PAD
AN1182	USING THE ST7 USB LOW-SPEED FIRMWARE
PRODUCT OPTIMIZATION	
AN982	USING CERAMIC RESONATORS WITH THE ST7
AN1014	HOW TO MINIMIZE THE ST7 POWER CONSUMPTION
AN1070	ST7 CHECKSUM SELFCHECKING CAPABILITY
AN1179	PROGRAMMING ST7 FLASH MICROCONTROLLERS IN REMOTE ISP
PRODUCT EVALUATION	
AN910	ST7 AND ST9 PERFORMANCE BENCHMARKING
AN990	ST7 BENEFITS VERSUS INDUSTRY STANDARD
AN1086	ST7 / ST10U435 CAN-do SOLUTIONS FOR CAR MULTIPLEXING
AN1150	BENCHMARK ST72 VS PC16
AN1151	PERFORMANCE COMPARISON BETWEEN ST72254 & PC16F8

14.5 TO GET MORE INFORMATION

To get the latest information on this product please use the ST web server: http://mcu.st.com/

15 SUMMARY OF CHANGES

Revision	Main Changes	Date
1.3	ST72F521/321/324 FLASH memory is now named HDFlash (High Density Flash). Device summary updates: sale types and temperature ranges. - 24K devices removed. - SO34 Package removed.TQFP32 pinout modified (Figure 4 on page 9 and Table 1 on page 10). CAUTION added for WDGRF flag use when LVD is not activated (Section 6.4.5 on page 30). CAUTION added when exiting Active-Halt mode and clearing OIE of MCCSR (Section 8.4.1 on page 41) Electrical characteristics: - V_{IN} for open-drain IOs and V_{PP} added in Absolute Maximum Rating (Section 12.2.1 on page 134). - Operating range updated for ROM and FLASH in Section 12.3.1 on page 135. - $Vt_{(POR)}$ updated in Section 12.3.2 on page 136. - RAM and FLASH notes added in Section 12.6 on page 146. - FLASH programming and erasing temperature range in Section 12.6.2 on page 146. - EMC characteristics updated in Section 12.7 on page 147 - I_{INJ} functionnal data added for I/O port pins in Section 12.8 on page 152. - IOs typical curves have been updated with preliminary caracterisation report in Section 12.8 on page 152. - ADC convertion time and accuracy have been updated in Section 12.12 on page 163. Default option byte value modified to Internal RC clock source (Section 14.1 on page 169). New temperature range in device configuration (Section 14.2 on page 171).	09-Feb-01

Notes: