

Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/stmicroelectronics/st72f32ak1b6">https://www.e-xfl.com/product-detail/stmicroelectronics/st72f32ak1b6</a>

---

# Table of Contents

---

<b>1 INTRODUCTION</b>	<b>7</b>
<b>2 PIN DESCRIPTION</b>	<b>8</b>
<b>3 REGISTER &amp; MEMORY MAP</b>	<b>13</b>
<b>4 FLASH PROGRAM MEMORY</b>	<b>17</b>
4.1 INTRODUCTION	17
4.2 MAIN FEATURES	17
4.3 STRUCTURE	17
4.3.1 Read-out Protection	17
4.4 ICC INTERFACE	18
4.5 ICP (IN-CIRCUIT PROGRAMMING)	19
4.6 IAP (IN-APPLICATION PROGRAMMING)	19
4.7 RELATED DOCUMENTATION	19
4.7.1 Register Description	19
<b>5 CENTRAL PROCESSING UNIT</b>	<b>20</b>
5.1 INTRODUCTION	20
5.2 MAIN FEATURES	20
5.3 CPU REGISTERS	20
<b>6 SUPPLY, RESET AND CLOCK MANAGEMENT</b>	<b>23</b>
6.1 PHASE LOCKED LOOP	23
6.2 MULTI-OSCILLATOR (MO)	24
6.3 RESET SEQUENCE MANAGER (RSM)	25
6.3.1 Introduction	25
6.3.2 Asynchronous External RESET pin	25
6.3.3 External Power-On RESET	26
6.3.4 Internal Watchdog RESET	26
6.4 SYSTEM INTEGRITY MANAGEMENT	27
6.4.1 Register Description	27
<b>7 INTERRUPTS</b>	<b>28</b>
7.1 INTRODUCTION	28
7.2 MASKING AND PROCESSING FLOW	28
7.3 INTERRUPTS AND LOW POWER MODES	30
7.4 CONCURRENT & NESTED MANAGEMENT	30
7.5 INTERRUPT REGISTER DESCRIPTION	31
7.6 EXTERNAL INTERRUPTS	33
7.6.1 I/O Port Interrupt Sensitivity	33
7.7 EXTERNAL INTERRUPT CONTROL REGISTER (EICR)	35
<b>8 POWER SAVING MODES</b>	<b>37</b>
8.1 INTRODUCTION	37
8.2 SLOW MODE	37
8.3 WAIT MODE	38
8.4 ACTIVE-HALT AND HALT MODES	39
8.4.1 ACTIVE-HALT MODE	39
8.4.2 HALT MODE	40

---

# Table of Contents

---

<b>9 I/O PORTS</b>	<b>42</b>
9.1 INTRODUCTION	42
9.2 FUNCTIONAL DESCRIPTION	42
9.2.1 Input Modes	42
9.2.2 Output Modes	42
9.2.3 Alternate Functions	42
9.3 I/O PORT IMPLEMENTATION	45
9.4 LOW POWER MODES	45
9.5 INTERRUPTS	45
9.5.1 I/O Port Implementation	46
<b>10 ON-CHIP PERIPHERALS</b>	<b>48</b>
10.1 WATCHDOG TIMER (WDG)	48
10.1.1 Introduction	48
10.1.2 Main Features	48
10.1.3 Functional Description	48
10.1.4 How to Program the Watchdog Timeout	49
10.1.5 Low Power Modes	51
10.1.6 Hardware Watchdog Option	51
10.1.7 Using Halt Mode with the WDG (WDGHALT option)	51
10.1.8 Interrupts	51
10.1.9 Register Description	51
10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)	53
10.2.1 Programmable CPU Clock Prescaler	53
10.2.2 Clock-out Capability	53
10.2.3 Real Time Clock Timer (RTC)	53
10.2.4 Beeper	53
10.2.5 Low Power Modes	54
10.2.6 Interrupts	54
10.2.7 Register Description	54
10.3 16-BIT TIMER	56
10.3.1 Introduction	56
10.3.2 Main Features	56
10.3.3 Functional Description	56
10.3.4 Low Power Modes	68
10.3.5 Interrupts	68
10.3.6 Summary of Timer modes	68
10.3.7 Register Description	69
10.4 SERIAL PERIPHERAL INTERFACE (SPI)	76
10.4.1 Introduction	76
10.4.2 Main Features	76
10.4.3 General Description	76
10.4.4 Clock Phase and Clock Polarity	80
10.4.5 Error Flags	81
10.4.6 Low Power Modes	83
10.4.7 Interrupts	83
10.4.8 Register Description	84
10.5 SERIAL COMMUNICATIONS INTERFACE (SCI)	87

---

# Table of Contents

---

10.5.1	Introduction	87
10.5.2	Main Features	87
10.5.3	General Description	87
10.5.4	Functional Description	89
10.5.5	Low Power Modes	96
10.5.6	Interrupts	96
10.5.7	Register Description	97
10.6	10-BIT A/D CONVERTER (ADC)	103
10.6.1	Introduction	103
10.6.2	Main Features	103
10.6.3	Functional Description	104
10.6.4	Low Power Modes	104
10.6.5	Interrupts	104
10.6.6	Register Description	105
<b>11</b>	<b>INSTRUCTION SET</b>	<b>107</b>
11.1	CPU ADDRESSING MODES	107
11.1.1	Inherent	108
11.1.2	Immediate	108
11.1.3	Direct	108
11.1.4	Indexed (No Offset, Short, Long)	108
11.1.5	Indirect (Short, Long)	108
11.1.6	Indirect Indexed (Short, Long)	109
11.1.7	Relative mode (Direct, Indirect)	109
11.2	INSTRUCTION GROUPS	110
<b>12</b>	<b>ELECTRICAL CHARACTERISTICS</b>	<b>113</b>
12.1	PARAMETER CONDITIONS	113
12.1.1	Minimum and Maximum values	113
12.1.2	Typical values	113
12.1.3	Typical curves	113
12.1.4	Loading capacitor	113
12.1.5	Pin input voltage	113
12.2	ABSOLUTE MAXIMUM RATINGS	114
12.2.1	Voltage Characteristics	114
12.2.2	Current Characteristics	114
12.2.3	Thermal Characteristics	115
12.3	OPERATING CONDITIONS	115
12.3.1	Operating Conditions	115
12.4	SUPPLY CURRENT CHARACTERISTICS	116
12.4.1	CURRENT CONSUMPTION	116
12.4.2	Supply and Clock Managers	118
12.4.3	On-Chip Peripherals	119
12.5	CLOCK AND TIMING CHARACTERISTICS	120
12.5.1	General Timings	120
12.5.2	External Clock Source	120
12.5.3	Crystal and Ceramic Resonator Oscillators	121
12.5.4	PLL Characteristics	123
12.6	MEMORY CHARACTERISTICS	124

## 1 INTRODUCTION

The ST72F32A and ST7232A devices are members of the ST7 microcontroller family designed for the 5V operating range.

The 32 and 44-pin devices are designed for mid-range applications

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

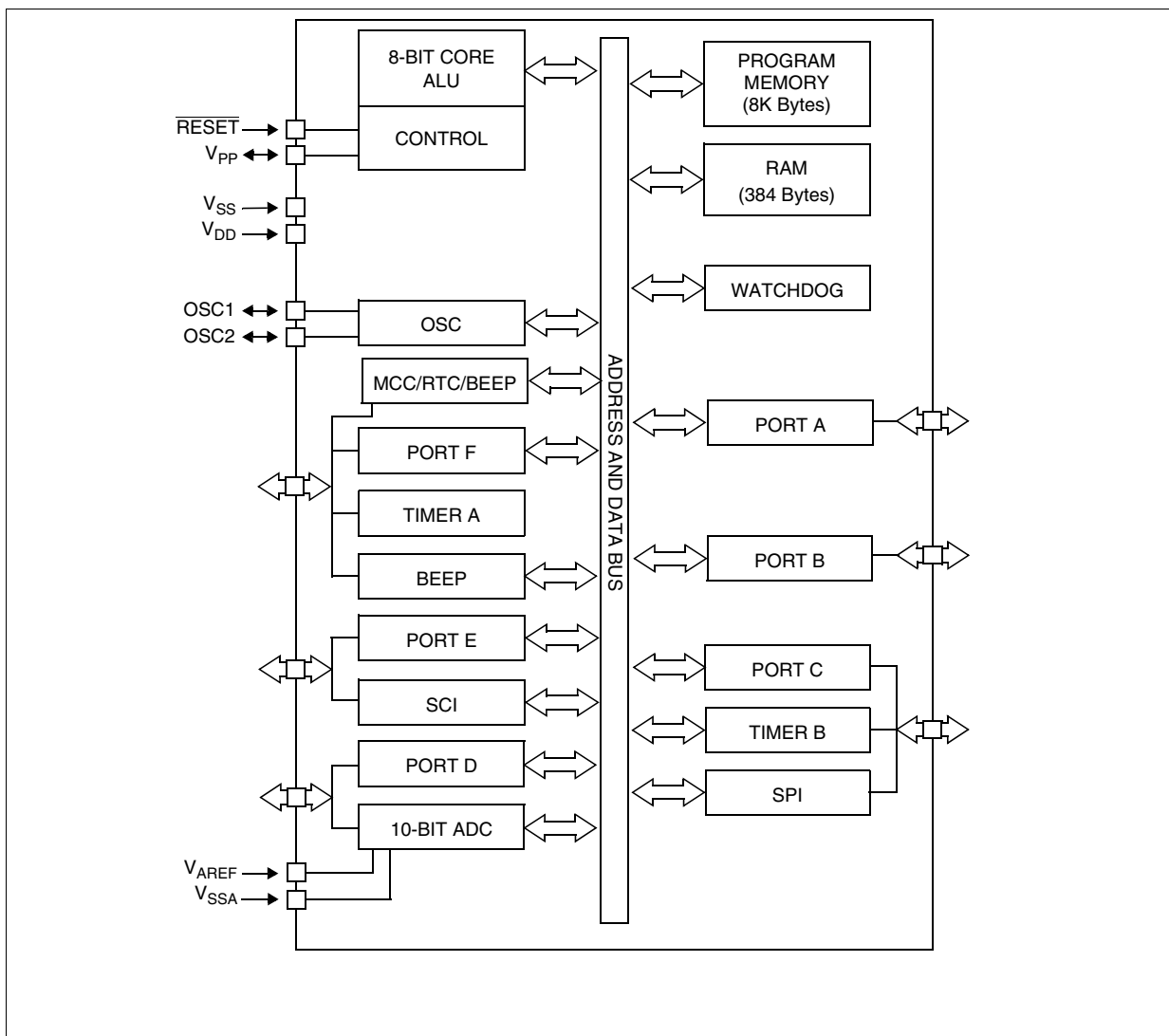


Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h 0001h 0002h	Port A <sup>2)</sup>	PADR PADDR PAOR	Port A Data Register Port A Data Direction Register Port A Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W R/W
0003h 0004h 0005h	Port B <sup>2)</sup>	PBDR PBDDR PBOR	Port B Data Register Port B Data Direction Register Port B Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W R/W
0006h 0007h 0008h	Port C	PCDR PCDDR PCOR	Port C Data Register Port C Data Direction Register Port C Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W R/W
0009h 000Ah 000Bh	Port D <sup>2)</sup>	PDADR PDDDR PDOR	Port D Data Register Port D Data Direction Register Port D Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W R/W
000Ch 000Dh 000Eh	Port E <sup>2)</sup>	PEDR PEDDR PEOR	Port E Data Register Port E Data Direction Register Port E Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W <sup>2)</sup> R/W <sup>2)</sup>
000Fh 0010h 0011h	Port F <sup>2)</sup>	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h <sup>1)</sup> 00h 00h	R/W R/W R/W
0012h to 0020h	Reserved Area (15 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	ISPR0 ISPR1 ISPR2 ISPR3	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	Reserved Area (1 Byte)				
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)				

## FLASH PROGRAM MEMORY (Cont'd)

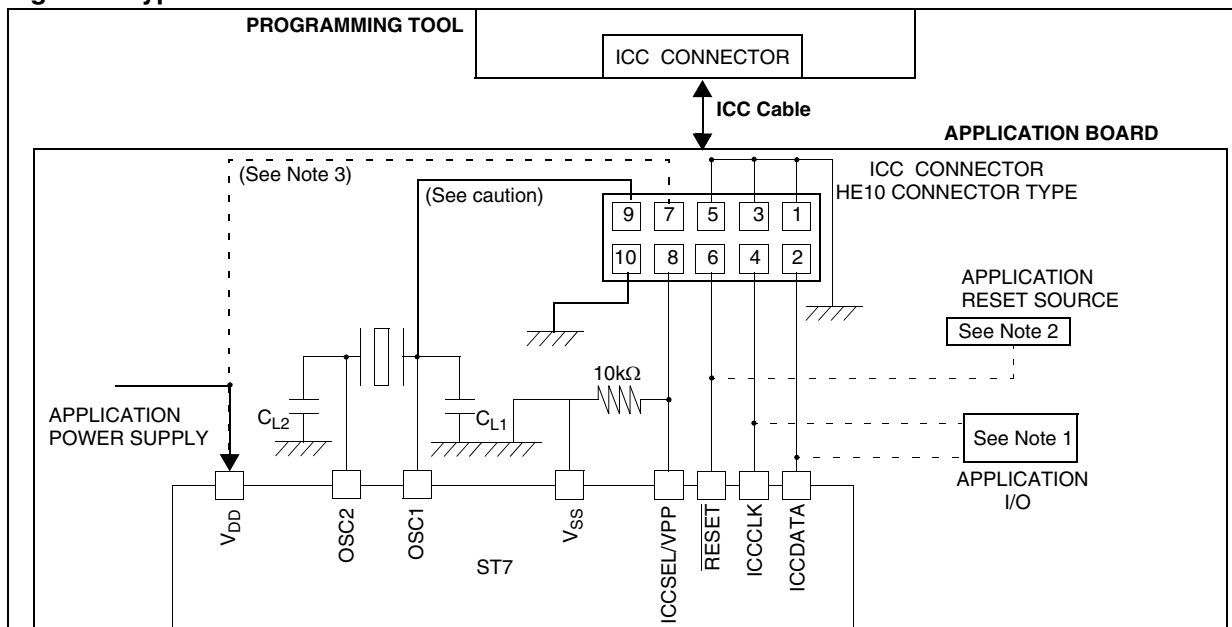
## 4.4 ICC Interface

ICC needs a minimum of 4 and up to 6 pins to be connected to the programming tool (see Figure 7). These pins are:

- $\overline{\text{RESET}}$ : device reset
- $V_{SS}$ : device power supply ground

- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/ $V_{PP}$ : programming voltage
- OSC1(or OSCIN): main clock input for external source (optional)
- $V_{DD}$ : application board power supply (optional, see Figure 7, Note 3)

Figure 7. Typical ICC Interface



## Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool 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. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2. During the ICC session, the programming tool must control the  $\overline{\text{RESET}}$  pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push pull output or pull-up resistor < 1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a

classical RC network with  $R > 1K$  or a reset management IC with open drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

**Caution:** External clock ICC entry mode is mandatory. Pin 9 must be connected to the OSC1 or OSCIN pin of the ST7 and OSC2 must be grounded.

**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 7<sup>th</sup> 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.



## 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
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
  - 5 Crystal/Ceramic resonator oscillators

### 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  $f_{OSC2} = f_{OSC}/2$ .

**Caution:** The PLL is not recommended for applications where timing accuracy is required.

Figure 10. PLL Block Diagram

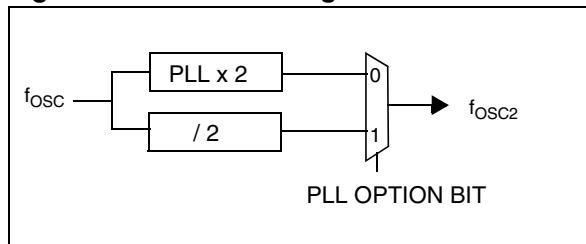
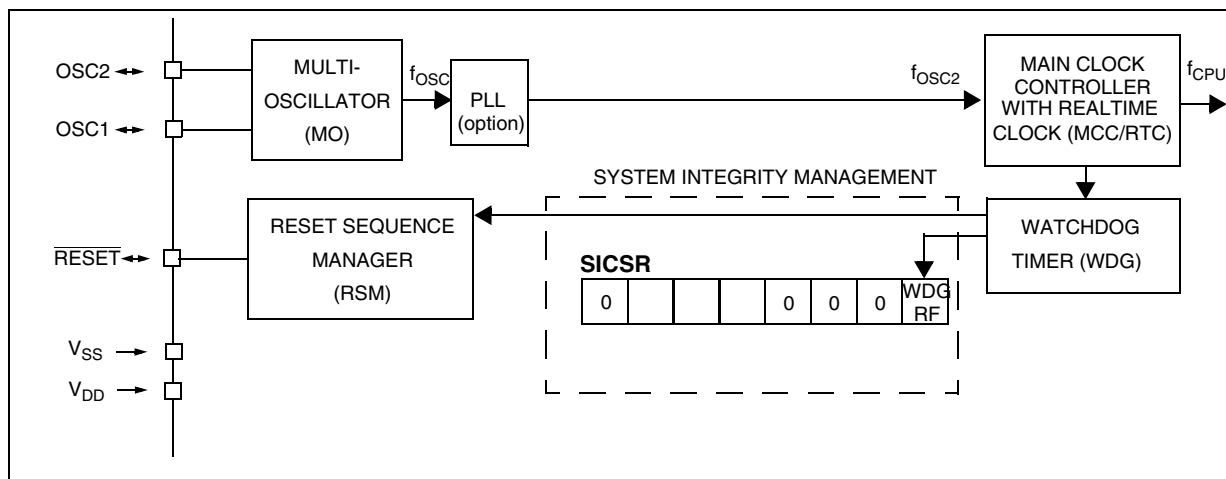


Figure 11. Clock, Reset and Supply Block Diagram



6.2 MULTI-OSCILLATOR (MO)

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

- an external source
- 4 crystal or ceramic resonator oscillators

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in Table 5. Refer to the electrical characteristics section for more details.

**Caution:** The OSC1 and/or OSC2 pins must not be left unconnected. For the purposes of Failure Mode and Effect Analysis, it should be noted that if the OSC1 and/or OSC2 pins are left unconnected, the ST7 main oscillator may start and, in this configuration, could generate an  $f_{OSC}$  clock frequency in excess of the allowed maximum ( $>16\text{MHz.}$ ), putting the ST7 in an unsafe/undefined state. The product behaviour must therefore be considered undefined when the OSC pins are left unconnected.

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.

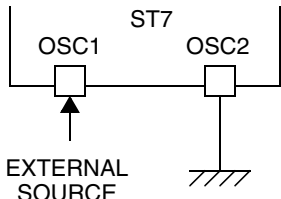
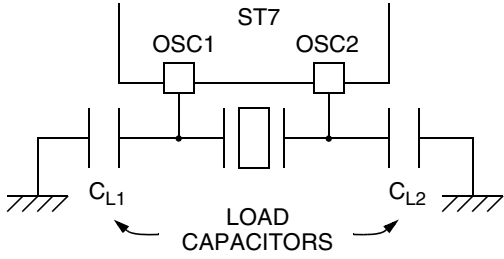
Crystal/Ceramic Oscillators

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 4 oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to Section 14.1 on page 145 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.

Table 5. ST7 Clock Sources

	Hardware Configuration
External Clock	
Crystal/Ceramic Resonators	

## 6.4 SYSTEM INTEGRITY MANAGEMENT

### 6.4.1 Register Description

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

Read/Write

Reset Value: 0000 000x (00h)

7							0
0	0	0	0	0	0	0	WDG RF

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).

Bits 7:1 = Reserved, must be kept cleared.

## 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
  - 2 non maskable events: 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 6). The processing flow is shown in Figure 15

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).

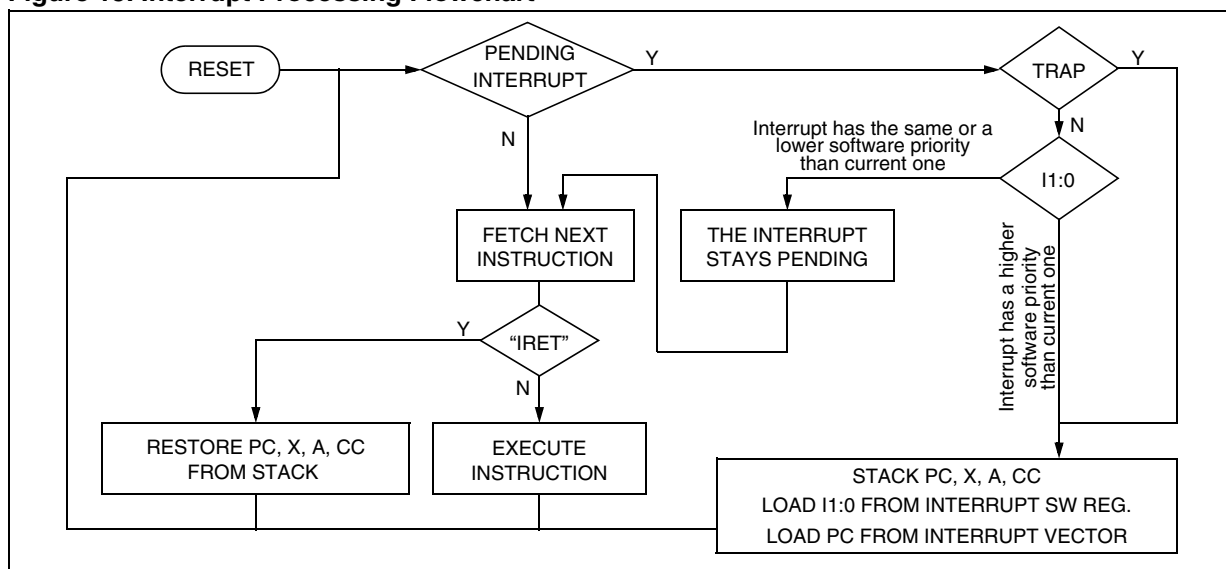
The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

**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 6. 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 15. Interrupt Processing Flowchart**



## INTERRUPTS (Cont'd)

Table 7. Dedicated Interrupt Instruction Set

Instruction	New Description	Function/Example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11 (level 3)	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						
POP CC	Pop CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load I0 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load I1 in I1:0 of CC	1		1			
TRAP	Software trap	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

**Note:** During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

## 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 MCCR register: CP[1:0] and SMS.

### 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 MCCR register.

**CAUTION:** When selected, the clock out pin suspends the clock during ACTIVE-HALT mode.

### 10.2.3 Real Time Clock Timer (RTC)

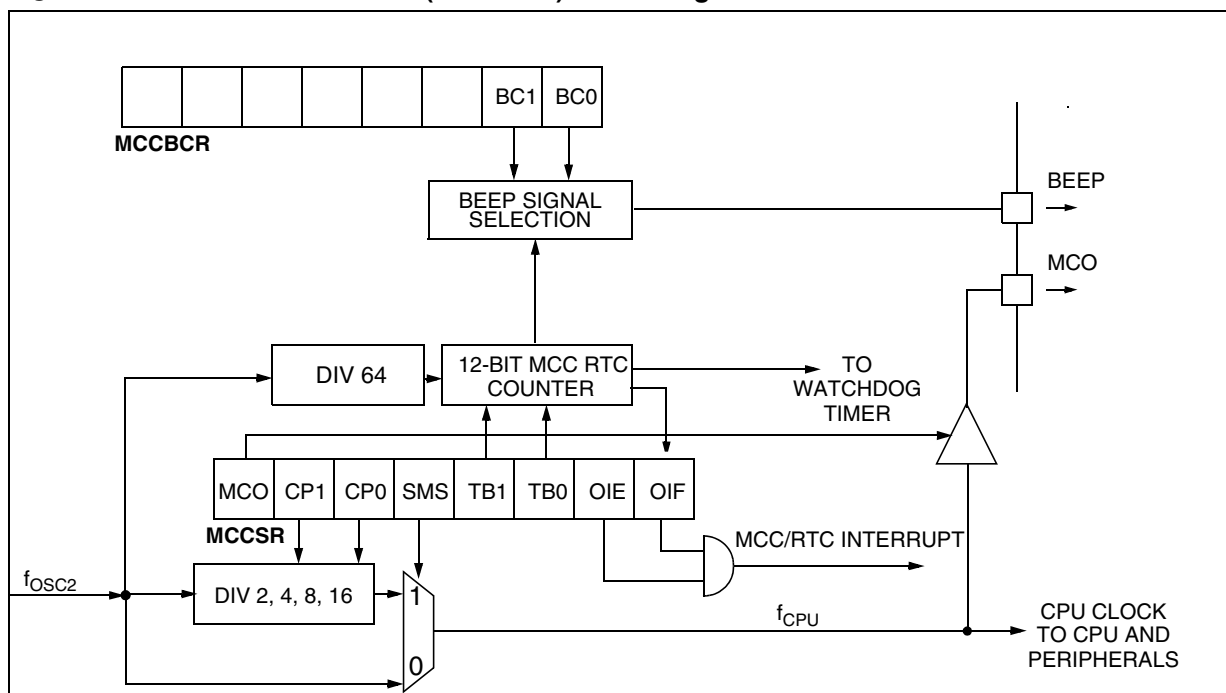
The counter of the real time clock timer allows an interrupt to be generated based on an accurate real time clock. Four different time bases depending directly on  $f_{OSC2}$  are available. The whole functionality is controlled by four bits of the MCCR register: TB[1:0], OIE and OIF.

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 32. Main Clock Controller (MCC/RTC) Block Diagram



**SERIAL PERIPHERAL INTERFACE (Cont'd)****10.4.3.2 Slave Select Management**

As an alternative to using the  $\overline{SS}$  pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see Figure 47)

In software management, the external  $\overline{SS}$  pin is free for other application uses and the internal  $\overline{SS}$  signal level is driven by writing to the SSI bit in the SPICSR register.

**In Master mode:**

- $\overline{SS}$  internal must be held high continuously

**In Slave Mode:**

There are two cases depending on the data/clock timing relationship (see Figure 46):

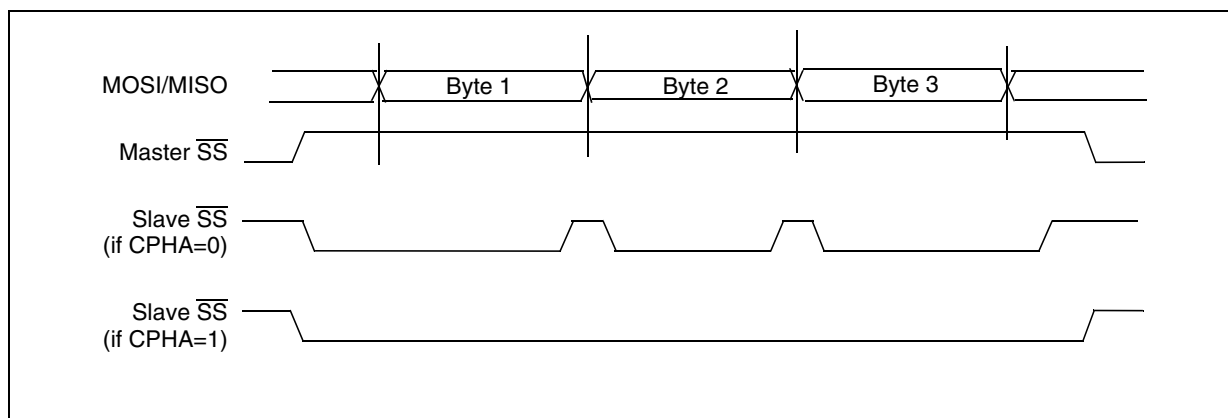
If CPHA=1 (data latched on 2nd clock edge):

- $\overline{SS}$  internal must be held low during the entire transmission. This implies that in single slave applications the  $\overline{SS}$  pin either can be tied to  $V_{SS}$ , or made free for standard I/O by managing the  $\overline{SS}$  function by software (SSM= 1 and SSI=0 in the in the SPICSR register)

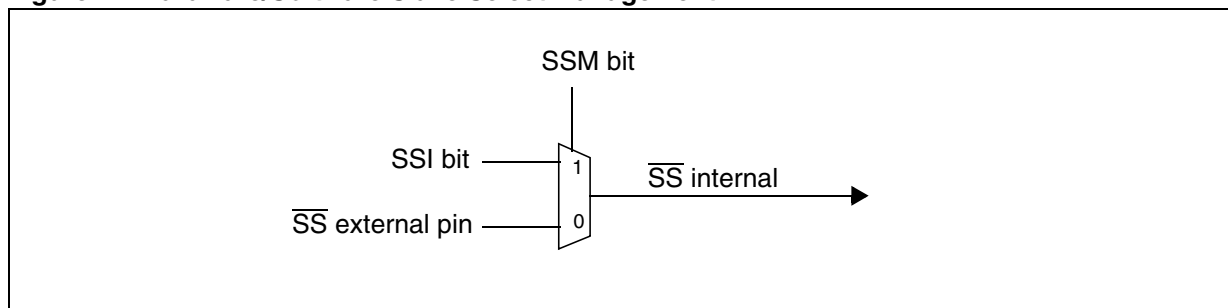
If CPHA=0 (data latched on 1st clock edge):

- $\overline{SS}$  internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If  $\overline{SS}$  is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see Section 10.4.5.3).

**Figure 46. Generic  $\overline{SS}$  Timing Diagram**



**Figure 47. Hardware/Software Slave Select Management**



## SERIAL PERIPHERAL INTERFACE (Cont'd)

### 10.4.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

**Note:** The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

To operate the SPI in master mode, perform the following steps in order (if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account):

1. Write to the SPICR register:
  - Select the clock frequency by configuring the SPR[2:0] bits.
  - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 48 shows the four possible configurations.  
**Note:** The slave must have the same CPOL and CPHA settings as the master.
2. Write to the SPICSR register:
  - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
  - Set the MSTR and SPE bits  
**Note:** MSTR and SPE bits remain set only if SS is high).

The transmit sequence begins when software writes a byte in the SPIDR register.

### 10.4.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register 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 request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

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.

### 10.4.3.5 Slave Mode Operation

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

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
  - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 48).  
**Note:** The slave must have the same CPOL and CPHA settings as the master.
  - Manage the  $\overline{SS}$  pin as described in Section 10.4.3.2 and Figure 46. If CPHA=1  $\overline{SS}$  must be held low continuously. If CPHA=0  $\overline{SS}$  must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

### 10.4.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register 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 request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

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 to the SPIDR 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 10.4.5.2).



# SERIAL PERIPHERAL INTERFACE (Cont'd)

## 10.4.6 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.

### 10.4.6.1 Using the SPI to wakeup the MCU from Halt mode

In slave configuration, the SPI is able to wakeup 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 10.4.3.2), make sure the master drives a low level on the  $\overline{SS}$  pin when the slave enters Halt mode.

### 10.4.7 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	Yes
Master Mode Fault Event	MODF		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

## SERIAL COMMUNICATIONS INTERFACE (Cont'd)

### 10.5.4.3 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 of a buffer (RDR) between the internal bus and the received shift register (see Figure 51).

#### 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. Normal data bits are considered valid if three consecutive samples (8th, 9th, 10th) have the same bit value, otherwise the NF flag is set. In the case of start bit detection, the NF flag is set on the basis of an algorithm combining both valid edge detection and three samples (8th, 9th, 10th). Therefore, to prevent the NF flag getting set during start bit reception, there should be a valid edge detection as well as three valid samples.

When noise is detected in a frame:

- The NF flag 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 flag is reset by a SCISR register read operation followed by a SCIDR register read operation.

During reception, if a false start bit is detected (e.g. 8th, 9th, 10th samples are 011,101,110), the frame is discarded and the receiving sequence is not started for this frame. There is no RDRF bit set for this frame and the NF flag is set internally (not accessible to the user). This NF flag is accessible along with the RDRF bit when a next valid frame is received.

**Note:** If the application Start Bit is not long enough to match the above requirements, then the NF Flag may get set due to the short Start Bit. In this case, the NF flag may be ignored by the application software when the first valid byte is received.

See also Section 10.5.4.10.

**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. It is set and cleared by software.

0: Transmitter is disabled

1: Transmitter is enabled

**Notes:**

– During transmission, a “0” pulse on the TE bit (“0” followed by “1”) sends a preamble (idle line) after the current word.

– When TE is set there is a 1 bit-time delay before the transmission starts.

**Caution:** The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).

**Bit 2 = RE** *Receiver enable.*

This bit enables the receiver. It is set and cleared by software.

0: Receiver is disabled

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.

## 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 condi-

tions 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 <sub>DD</sub> - V <sub>SS</sub>	Supply voltage	6.5	V
V <sub>PP</sub> - V <sub>SS</sub>	Programming Voltage	13	
V <sub>IN</sub> <sup>1) &amp; 2)</sup>	Input Voltage on true open drain pin	V <sub>SS</sub> -0.3 to 6.5	
	Input voltage on any other pin	V <sub>SS</sub> -0.3 to V <sub>DD</sub> +0.3	
ΔV <sub>DDx</sub>   and  ΔV <sub>SSx</sub>	Variations between different digital power pins	50	mV
V <sub>SSA</sub> - V <sub>SSx</sub>	Variations between digital and analog ground pins	50	
V <sub>ESD</sub> (HBM)	Electro-static discharge voltage (Human Body Model)	see Section 12.7.3 on page 127	
V <sub>ESD</sub> (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) <sup>3)</sup>	32-pin devices	75	mA
$I_{VSS}$	Total current out of $V_{SS}$ ground lines (sink) <sup>3)</sup>	32-pin devices	75	mA
$I_{IO}$	Output current sunk by any standard I/O and control pin		25	mA
	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		$\pm 5$	
	Injected current on $\overline{RESET}$ pin		$\pm 5$	
	Injected current on OSC1 and OSC2 pins		$\pm 5$	
	Injected current on Flash device pin PB0		+5	
	Injected current on any other pin <sup>5) \&amp; 6)</sup>		$\pm 5$	
$\Sigma I_{INJ(PIN)}^{2)}$	Total injected current (sum of all I/O and control pins) <sup>5)</sup>		$\pm 25$	

#### 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.  $I_{INJ(PIN)}$  must never be exceeded. This is implicitly insured if  $V_{IN}$  maximum is respected. If  $V_{IN}$  maximum cannot be respected, the injection current must be limited externally to the  $I_{INJ(PIN)}$  value. A positive injection is induced by  $V_{IN} > V_{DD}$  while a negative injection is induced by  $V_{IN} < V_{SS}$ . For true open-drain pads, there is no positive injection current, and the corresponding  $V_{IN}$  maximum must always be respected.

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. See note in “ADC Accuracy” on page 140.

For best reliability, it is recommended to avoid negative injection of more than 1.6mA.

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.

## 12.8 I/O PORT PIN CHARACTERISTICS

### 12.8.1 General Characteristics

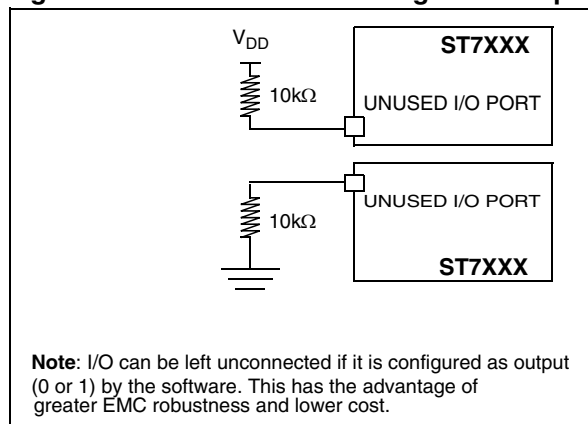
Subject to general operating conditions for  $V_{DD}$ ,  $f_{OSC}$ , and  $T_A$  unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IL}$	Input low level voltage (standard voltage devices) <sup>1)</sup>				$0.3 \times V_{DD}$	V
$V_{IH}$	Input high level voltage <sup>1)</sup>		$0.7 \times V_{DD}$			V
$V_{hys}$	Schmitt trigger voltage hysteresis <sup>2)</sup>			0.7		
$I_{INJ(PIN)}^{3)}$	Injected Current on Flash device pin PB0	$V_{DD}=5V$	0		+4	mA
	Injected Current on other I/O pins				$\pm 4$	
$\Sigma I_{INJ(PIN)}^{3)}$	Total injected current (sum of all I/O and control pins)				$\pm 25$	
$I_{lkg}$	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			$\pm 1$	$\mu A$
$I_S$	Static current consumption induced by each floating input pin	Floating input mode <sup>4)</sup>		200		
$R_{PU}$	Weak pull-up equivalent resistor <sup>5)</sup>	$V_{IN}=V_{SS}$   $V_{DD}=5V$	50	120	250	$k\Omega$
$C_{IO}$	I/O pin capacitance			5		pF
$t_{f(I/O)out}$	Output high to low level fall time <sup>1)</sup>	$C_L=50pF$ Between 10% and 90%		25		ns
$t_{r(I/O)out}$	Output low to high level rise time <sup>1)</sup>			25		
$t_w(IT)in$	External interrupt pulse time <sup>6)</sup>		1			$t_{CPU}$

#### Notes:

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
3. 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 114 for more details.
4. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example and leaving the I/O unconnected on the board or an external pull-up or pull-down resistor (see Figure 66). Static peak current value taken at a fixed  $V_{IN}$  value, based on design simulation and technology characteristics, not tested in production. This value depends on  $V_{DD}$  and temperature values.
5. The  $R_{PU}$  pull-up equivalent resistor is based on a resistive transistor (corresponding IPU current characteristics described in Figure 67).
6. 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.

**Figure 66. Unused I/O Pins configured as input**



**Figure 67. Typical  $I_{PU}$  vs.  $V_{DD}$  with  $V_{IN}=V_{SS}$**

