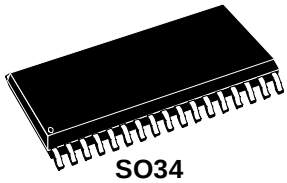
ST72C171

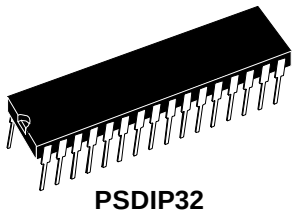
8-BIT MCU with 8K FLASH, ADC, WDG, SPI, SCI, TIMERS SPGAs (Software Programmable Gain Amplifiers), OP-AMP

PRODUCT PREVIEW

- Memories
  - 8K of single voltage Flash Program memory with read-out protection
  - In-Situ Programming (Remote ISP)
- Clock, Reset and Supply Management
  - Enhanced Reset System
  - Low voltage supervisor (LVD) with 3 programmable levels
  - Low consumption resonator or RC oscillators (internal or external) and by-pass for external clock source, with safe control capabilities
  - 3 Power Saving modes
- 22 I/O Ports
  - 22 multifunctional bidirectional I/O lines:
  - 16 interrupt inputs on 2 independent lines
  - 8 lines configurable as analog inputs
  - 20 alternate functions
  - EMI filtering
- 2 Timers and Watchdog
  - One 16-bit Timer with: 2 Input Captures, 2 Output Compares, external Clock input, PWM and Pulse Generator modes
  - One 8-bit Autoreload Timer (ART) with: 2 PWM output channels (internally connectable to the SPGA inputs), 1 Input Capture, external clock input
  - Configurable watchdog (WDG)
- 2 Communications Interfaces
  - Synchronous Serial Peripheral Interface (SPI)
  - Serial Communications Interface (SCI)



SO34



PSDIP32

- 3 Analog peripherals
  - 2 Software Programmable Gain Operational Amplifiers (SPGAs) with rail-to-rail input and output, V_{DD} independent (band gap) and programmable reference voltage (1/8 V_{DD} resolution), Offset compensation, DAC & on/off switching capability
  - 1 rail-to-rail input and output Op-Amp
  - 8-bit A/D Converter with up to 11 channels (including 3 internal channels connected to the Op-Amp & SPGA outputs)
- 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

Device Summary

Features	**ST72C171K2M**	**ST72C171K2B**
Flash - bytes	8K Single Voltage	
RAM (stack) - bytes	256 (128)	
Peripherals	2 SPGAs, 1 Op-Amp, Watchdog, 3 Timers, SPI, SCI, ADC (11 chan.)	2 SPGAs, Watchdog, 3 Timers, SPI, SCI, ADC (11 chan.)
Operating Supply	3.2 V to 5.5 V	
CPU Frequency	Up to 8 MHz (with up to 16 MHz oscillator)	
Temperature Range	- 40°C to + 85°C	
Package	SO34	PSDIP32

Rev. 1.4

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST72C171 is a member of the ST7 family of Microcontrollers. All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

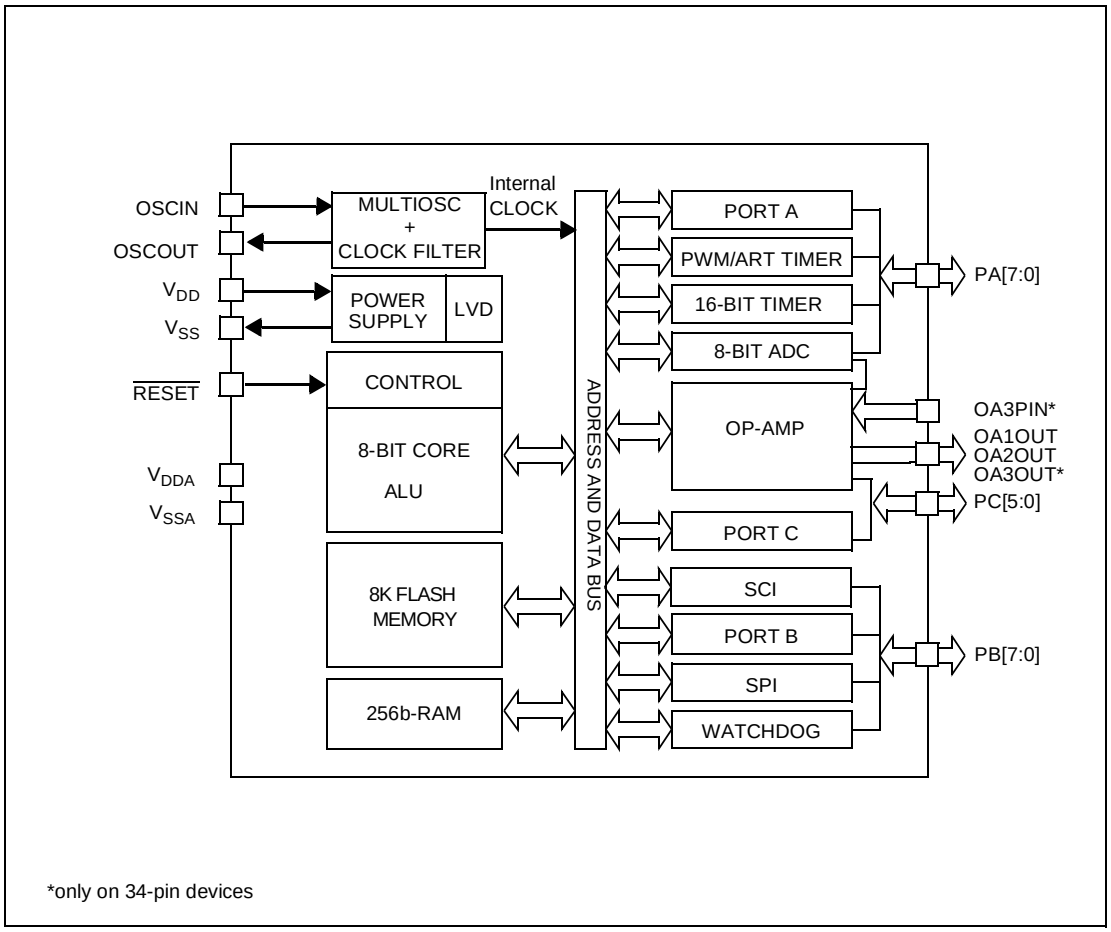
The ST72C171 features single-voltage FLASH memory with byte-by-byte In-Situ Programming (ISP) capability.

Under software control, the device can be placed in WAIT, SLOW, or HALT mode, reducing power consumption when the application is in idle or standby 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 The device includes a low consumption and fast start on-chip oscillator, CPU, Flash program memory, RAM, 22 I/O lines and the following on-chip peripherals: Analog-to-Digital converter (ADC) with 8 multiplexed analog inputs, Op-Amp module, synchronous SPI serial interface, asyncronous serial interface (SCI), Watchdog timer, a 16-bit Timer featuring external Clock Input, Pulse Generator capabilities, 2 Input Captures and 2 Output Compares, an 8-bit Timer featuring external Clock Input, Pulse Generator Capabilities (2 channels), Autoreload and Input Capture.

The Op-Amp module adds on-chip analog features to the MCU, that usually require using external components.

Figure 1. ST72C171 Block Diagram



1.2 PIN DESCRIPTION

Figure 2. 34-Pin SO Package Pinout

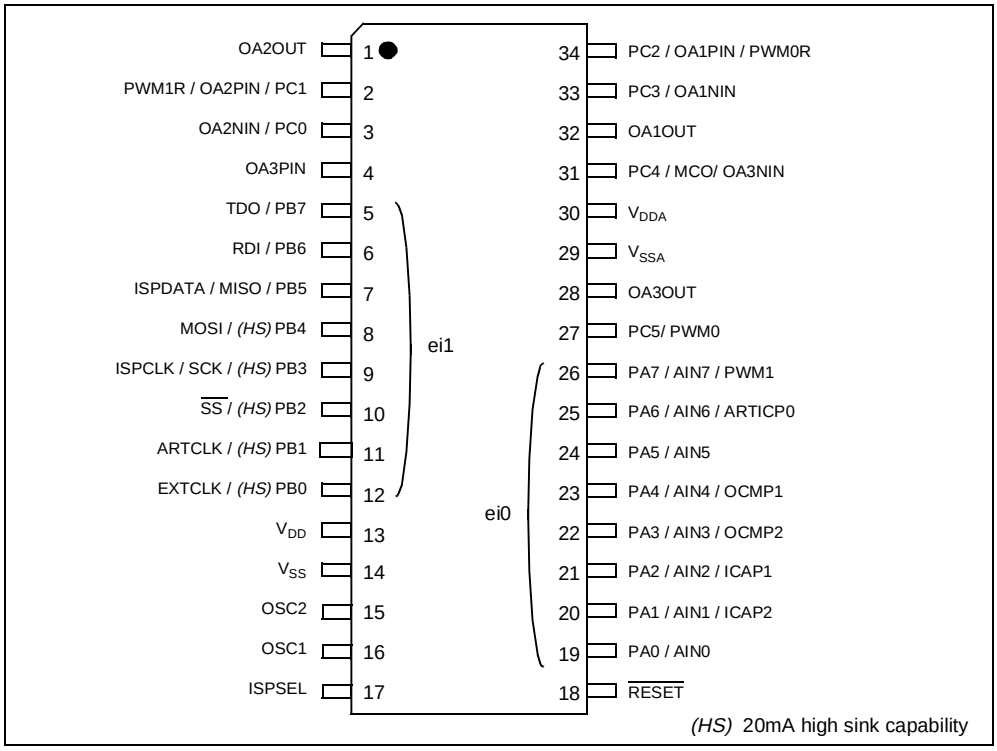
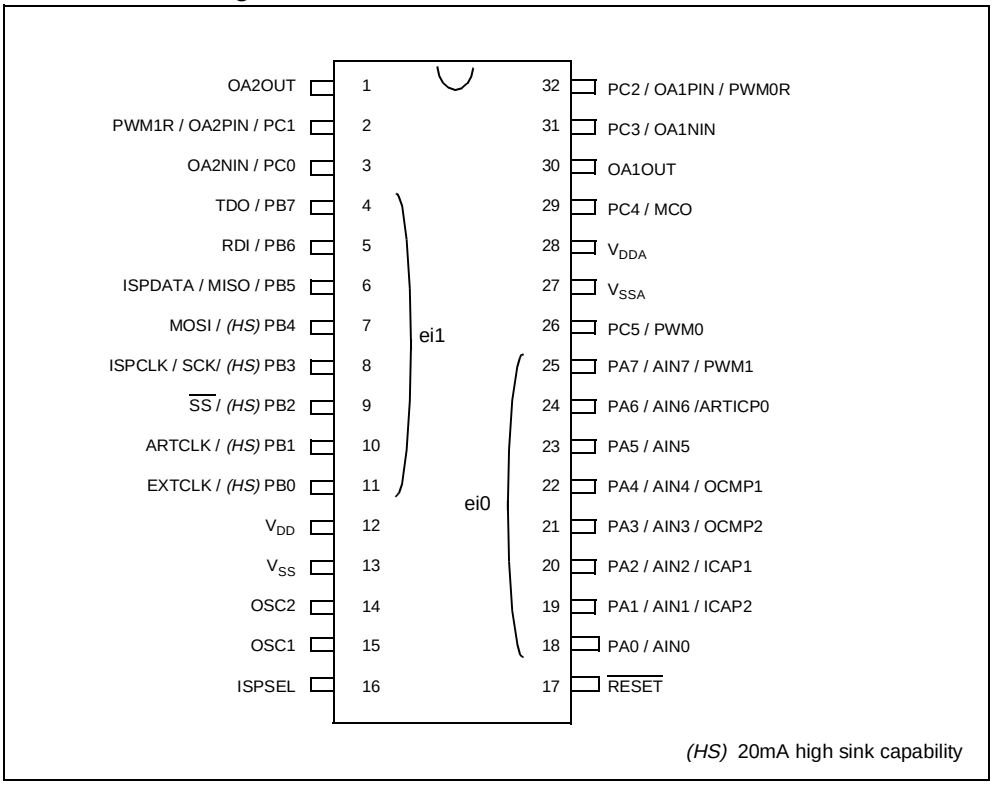


Figure 3. 32-Pin SDIP Package Pinout



PIN DESCRIPTION (Cont'd)

Legend / Abbreviations:

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

In/Output level: C = CMOS $0.3V_{DD}/0.7V_{DD}$,
C_R = CMOS Levels with resistive output (1K)
A = Analog levels

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

Port configuration capabilities:

– Input:float = floating, wpu = weak pull-up, int = interrupt, ana = analog

– Output: OD = open drain, T = true open drain, PP = push-pull

Note: the Reset configuration of each pin is shown in bold.

Table 1. Device Pin Description

Pin n°		Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
						Input				Output			
SDIP32	SO34			Input	Output	float	wpu	int	ana	OD	PP		
1	1	OA2OUT	O		A							OA2 output	
2	2	PC1/OA2PIN/ PWM1R	I/O	C	C/C_R	**X**	X		X	X	X	Port C1	OA2 noninverting input and/or ART PWM1 resistive output
3	3	PC0/OA2NIN	I/O	C/A	C	**X**	X		X	X	X	Port C0	OA2 inverting input
-	4	OA3PIN	I	A									OA3 noninverting input
4	5	PB7/TDO	I/O	C		**X**	ei1			X	X	Port B7	SCI transmit
5	6	PB6/RDI	I/O	C		**X**	ei1			X	X	Port B6	SCI receive
6	7	PB5/MISO/ISPDATA	I/O	C		**X**	ei1			X	X	Port B5	SPI data master in/slave out or In Situ Programming Data Input
7	8	PB4/MOSI	I/O	C	HS	**X**	ei1			X	X	Port B4	SPI data master out/slave in
8	9	PB3/SCK/ISPCLK	I/O	C	HS	**X**	ei1			X	X	Port B3	SPI Clock or In Situ Programming Clock Output
9	10	PB2/$\overline{SS}$	I/O	C	HS	**X**	ei1			X	X	Port B2	SPI Slave Select (active low)
10	11	PB1/ARTCLK	I/O	C	HS	**X**	ei1			X	X	Port B1	ART External Clock
11	12	PB0/EXTCLK	I/O	C	HS	**X**	ei1			X	X	Port B0	Timer16 External Clock
12	13	V_{DD}	S									Digital Main Supply Voltage	
13	14	V_{SS}	S									Digital ground voltage	
14	15	OSC2										Resonator oscillator inverter output or capacitor input for RC oscillator	
15	16	OSC1										External clock input or Resonator oscillator inverter input or resistor input for RC oscillator	
16	17	ISPSEL	I	C								In Situ Programming Mode Select Must be tied to V_{SS} in user mode	
17	18	$\overline{RESET}$	I/O	C			X				X	External $\overline{Reset}$	
18	19	PA0/AIN0	I/O	C		**X**	ei0		X	X	X	Port A0	ADC input 0
19	20	PA1/AIN1/ICAP2	I/O	C		**X**	ei0		X	X	X	Port A1	ADC input 1 orTimer16 input capture 2

Pin n°		Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
						Input				Output			
SDIP32	SO34			Input	Output	float	wpu	int	ana	OD	PP		
20	21	PA2/AIN2/ICAP1	I/O	C		**X**	ei0		X	X	X	Port A2	ADC input 2 or Timer16 input capture 1
21	22	PA3/AIN3/OCMP2	I/O	C		**X**	ei0		X	X	X	Port A3	ADC input 3 or Timer16 output compare 2
22	23	PA4 /AIN4/OCMP1	I/O	C		**X**	ei0		X	X	X	Port A4	ADC input 4 or Timer16 output compare 1
23	24	PA5/AIN5	I/O	C		**X**	ei0		X	X	X	Port A5	ADC input 5
24	25	PA6/AIN6/ARTICP0	I/O	C		**X**	ei0		X	X	X	Port A6	ADC input 6 or ART input capture
25	26	PA7/AIN7/PWM1	I/O	C		**X**	ei0		X	X	X	Port A7	ADC input 7 or ART PWM1 output
26	27	PC5 / PWM0	I/O	C		**X**	X			X	X	Port C5	ART PWM0 output
-	28	OA3OUT	O		A							OA3 output	
27	29	V_{SSA}										Analog ground	
28	30	V_{DDA}										Analog supply	
29	31	PC4/MCO/OA3NIN	I/O	C		**X**	X			X	X	Port C4	Main Clock Out or OA3 inverting input
30	32	OA1OUT	O		A							OA1 output	
31	33	PC3/OA1NIN	I/O	C/A	C	**X**	X			X	X	Port C3	OA1 inverting input
32	34	PC2/OA1PIN/ PWM0R	I/O	C/A	C/C_R	**X**	X			X	X	Port C2	OA1 non-inverting input and/ or ART PWM0 resistive output

Notes:

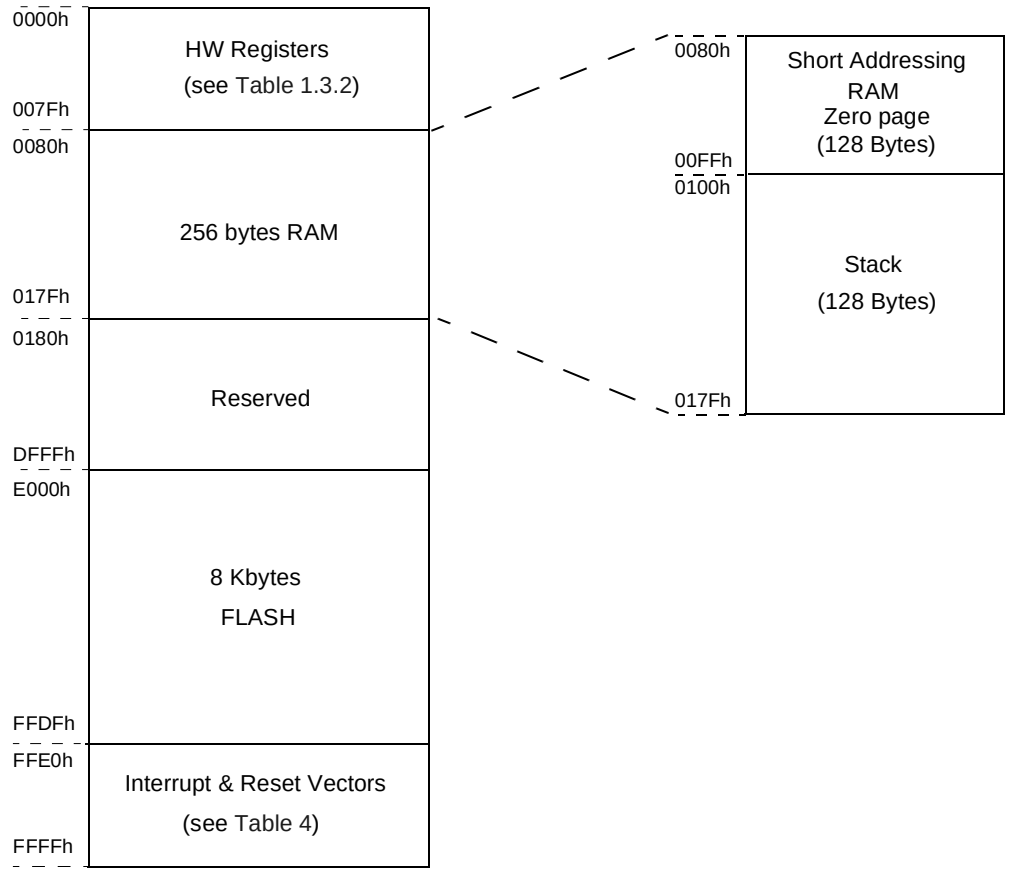
1. In the interrupt input column, “eix” defines the associated external interrupt vector. If the weak pull-up column (wpu) is associated with the interrupt column (int), then the I/O configuration is pull-up interrupt input, else the configuration is floating interrupt input.

2. OSC1 and OSC2 pins connect a crystal or ceramic resonator, an external RC, or an external source to the on-chip oscillator see dedicated See “PIN DESCRIPTION” on page 5. for more details.

1.3 MEMORY MAP

1.3.1 Introduction

Figure 4. Program Memory Map



1.3.2 Data Register

Table 2. Hardware Register Memory Map

Address	Block Name	Register Label	Register name	Reset Status	Remarks
0000h	Port A	PADR	Data Register	00h	R/W
0001h		PADDR	Data Direction Register	00h	R/W
0002h		PAOR	Option Register	00h	R/W
0003h			Not Used		Absent
0004h	Port B	PBDR	Data Register	00h	R/W
0005h		PBDDR	Data Direction Register	00h	R/W
0006h		PBOR	Option Register	00h	R/W
0007h			Not Used		Absent
0008h	Port C	PCDR	Data Register	00h	R/W
0009h		PCDDR	Data Direction Register	00h	R/W
000Ah		PCOR	Option Register	00h	R/W
000Bh to 001Ah	Reserved Area (16 Bytes)				
001Bh	OPAMP	OA1CR	OA1 Control Register	00h	R/W
001Ch		OA2CR	OA2 Control Register	00h	R/W
001Dh		OA3CR	OA3 Control Register	00h	R/W
001Eh		OAIRR	OA Interrupt & Readout Register	00h	Section 7.3
001Fh		OAVRCR	OA Voltage Reference Control Register	00h	R/W
0020h	MISC1	MISCR1	Miscellaneous Register 1	00h	see Section 4.3.5
0021h	SPI	SPIDR	Data I/O Register	xxh	R/W
0022h		SPICR	Control Register	0xh	R/W
0023h		SPISR	Status Register	00h	Read Only
0024h	WDG	WDGCR	Watchdog Control register	7Fh	R/W
0025h	CRS	CRSR	Clock, Reset and Supply Control / Status Register	00h	R/W
0026h to 0030h	Reserved Area (11 Bytes)				
0031h	TIMER16	TACR2	Control Register2	00h	R/W
0032h		TACR1	Control Register1	00h	R/W
0033h		TASR	Status Register	xxh	Read Only
0034h-		TAIC1HR	Input Capture1 High Register	xxh	Read Only
0035h		TAIC1LR	Input Capture1 Low Register	xxh	Read Only
0036h-		TAOC1HR	Output Compare1 High Register	80h	R/W
0037h		TAOC1LR	Output Compare1 Low Register	00h	R/W
0038h-		TACHR	Counter High Register	FFh	Read Only
0039h		TACLR	Counter Low Register	FCh	Read Only
003Ah-		TAACHR	Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Alternate Counter Low Register	FCh	Read Only
003Ch-		TAIC2HR	Input Capture2 High Register	xxh	Read Only
003Dh		TAIC2LR	Input Capture2 Low Register	xxh	Read Only
003Eh-		TAOC2HR	Output Compare2 High Register	80h	R/W
003Fh		TAOC2LR	Output Compare2 Low Register	00h	R/W
0040h	MISC2	MISCR2	Miscellaneous Register2	00h	see Section 7.2.2

Address	Block Name	Register Label	Register name	Reset Status	Remarks
0041h to 004Fh	Reserved Area (15 Bytes)				
0050h	SCI	SCISR	Status Register	0C0h	Read Only
0051h		SCIDR	Data Register	0xxh	R/W
0052h		SCIBRR	Baud Rate Register	0Xxh	R/W
0053h		SCICR1	Control Register 1	0xxh	R/W
0054h		SCICR2	Control Register 2	00h	R/W
0055h to 006Fh	Reserved Area (27 Bytes)				
0070h	ADC	ADCDR	Data Register	00h	Read Only
0071h		ADCCSR	Control/Status Register	00h	R/W
0072h 0073h	Reserved Area (2 Bytes)				
0074h	ART/PWM	PWMDCR1	PWM Duty Cycle Register 1	00h	R/W
0075h		PWMDCR0	PWM Duty Cycle Register 0	00h	R/W
0076h		PWMCR	PWM Control Register	00h	R/W
0077h		ARTCSR	Control/Status Register	00h	R/W
0078h		ARTCAR	Counter Access Register	00h	R/W
0079h		ARTARR	Auto Reload Register	00h	R/W
007Ah		ARTICCSR	Input Capture Control Status Register	00h	R/W
007Bh		ARTICR1	Input Capture Register 1	00h	Read Only
007Ch to 007Fh	Reserved Area (4 Bytes)				

2 FLASH PROGRAM MEMORY

2.1 INTRODUCTION

FLASH devices have a single voltage non-volatile FLASH memory that may be programmed in-situ (or plugged in a programming tool) on a byte-by-byte basis.

2.2 MAIN FEATURES

- Remote In-Situ Programming (ISP) mode
- Up to 16 bytes programmed in the same cycle
- MTP memory (Multiple Time Programmable)
- Read-out memory protection against piracy

2.3 STRUCTURAL ORGANISATION

The FLASH program memory is organised in a single 8-bit wide memory block which can be used for storing both code and data constants.

The FLASH program memory is mapped in the upper part of the ST7 addressing space and includes the reset and interrupt user vector area .

2.4 IN-SITU PROGRAMMING (ISP) MODE

The FLASH program memory can be programmed using Remote ISP mode. This ISP mode allows the contents of the ST7 program memory to be updated using a standard ST7 programming tools after the device is mounted on the application board. This feature can be implemented with a minimum number of added components and board area impact.

An example Remote ISP hardware interface to the standard ST7 programming tool is described below. For more details on ISP programming, refer to the ST7 Programming Specification.

Remote ISP Overview

The Remote ISP mode is initiated by a specific sequence on the dedicated ISPSEL pin.

The Remote ISP is performed in three steps:

- Selection of the RAM execution mode
- Download of Remote ISP code in RAM
- Execution of Remote ISP code in RAM to program the user program into the FLASH

Remote ISP hardware configuration

In Remote ISP mode, the ST7 has to be supplied with power (V_{DD} and V_{SS}) and a clock signal (oscillator and application crystal circuit for example).

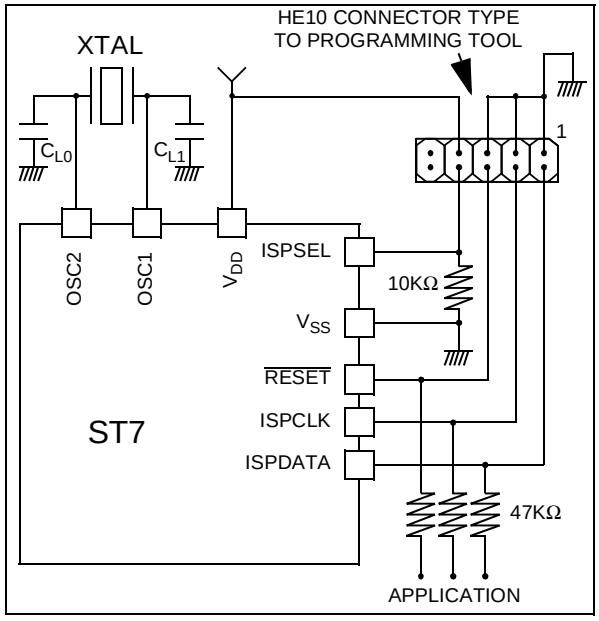
This mode needs five signals (plus the V_{DD} signal if necessary) to be connected to the programming tool. This signals are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device ground power supply
- ISPCLK: ISP output serial clock pin
- ISPDATA: ISP input serial data pin
- ISPSEL: Remote ISP mode selection. This pin must be connected to V_{SS} on the application board through a pull-down resistor.

If any of these pins are used for other purposes on the application, a serial resistor has to be implemented to avoid a conflict if the other device forces the signal level.

Figure 1 shows a typical hardware interface to a standard ST7 programming tool. For more details on the pin locations, refer to the device pinout description.

Figure 5. Typical Remote ISP Interface



2.5 MEMORY READ-OUT PROTECTION

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

For FLASH devices, when this option is selected, the program and data stored in the FLASH memory are protected against read-out piracy (including a re-write protection). When this protection option is removed the entire FLASH program memory is first automatically erased. However, the E^2PROM data memory (when available) can be protected only with ROM devices.

3 CENTRAL PROCESSING UNIT

3.1 INTRODUCTION

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

3.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

3.3 CPU REGISTERS

The 6 CPU registers shown in Figure 1 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

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

Index Registers (X and Y)

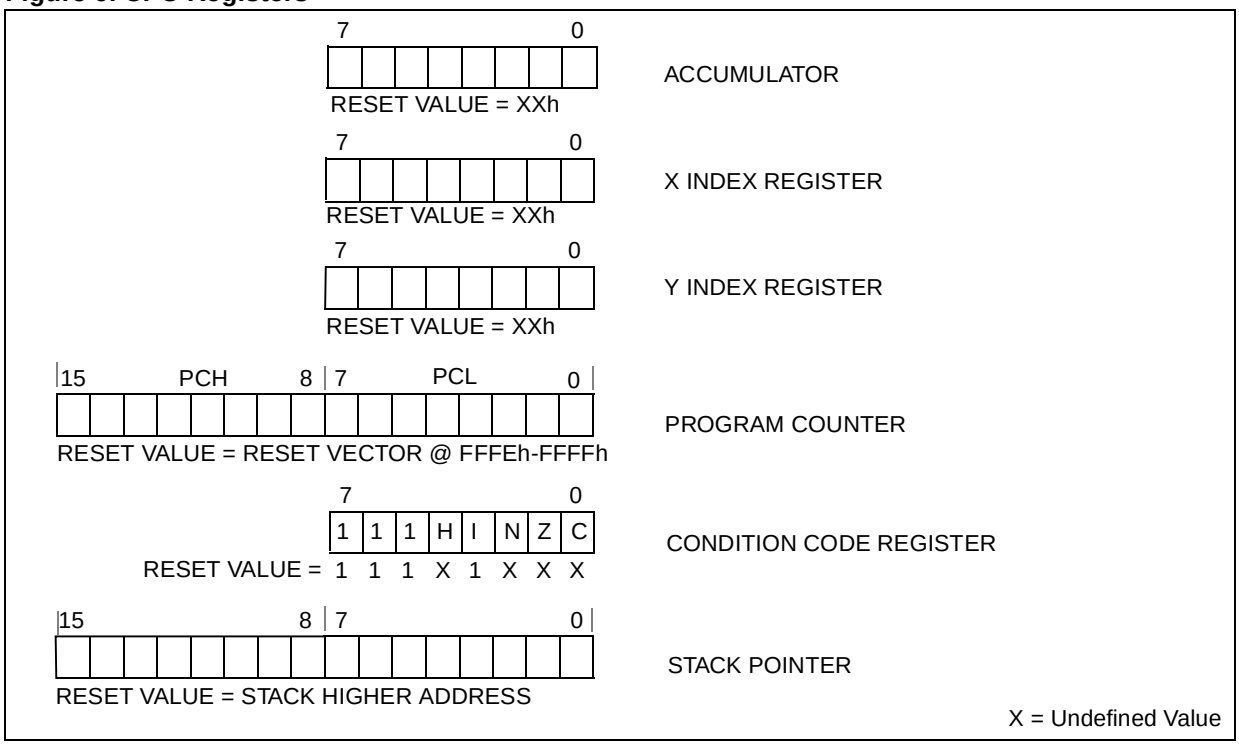
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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

Program Counter (PC)

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

Figure 6. CPU Registers



CPU REGISTERS (Cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

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

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

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

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

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

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

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

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

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

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

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

This bit is accessed by the JRMI and JRPL instructions.

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

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

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

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

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

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

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 7Fh

15							8
0	0	0	0	0	0	0	1

7							0
0	1	SP5	SP4	SP3	SP2	SP1	SP0

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

Since the stack is 128 bytes deep, the 10 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 SP5 to SP0 bits are set) which is the stack higher address.

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

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

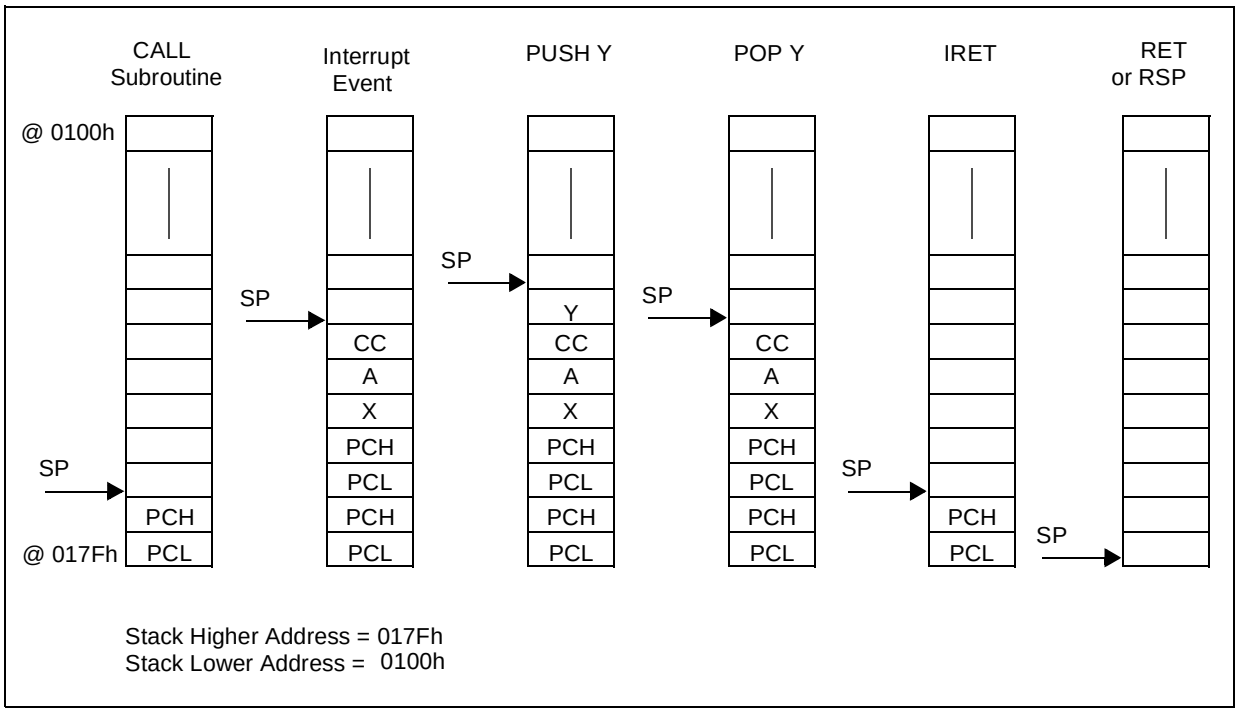
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the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

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

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

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

Figure 7. Stack Manipulation Example



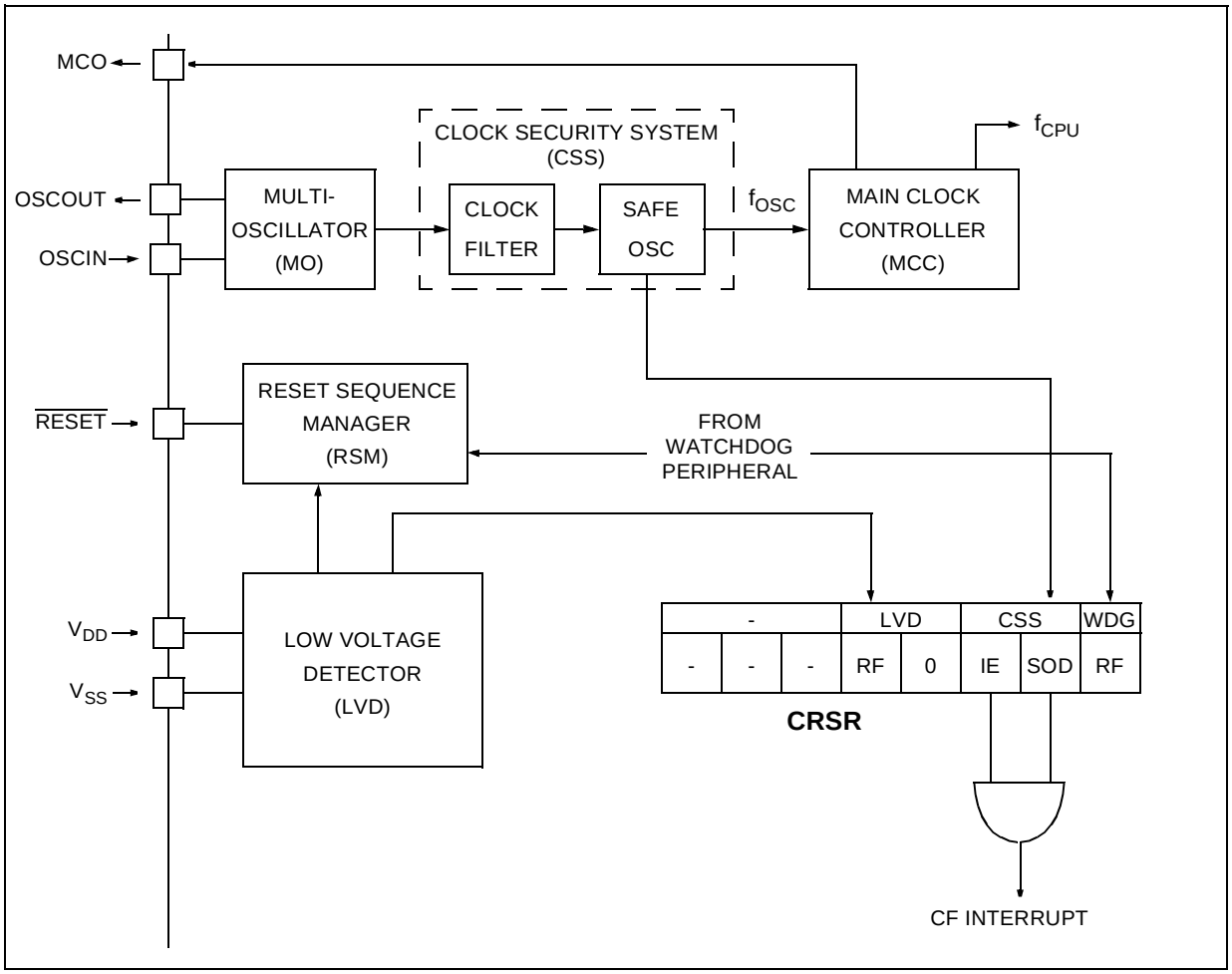
4 SUPPLY, RESET AND CLOCK MANAGEMENT

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 8.

4.1 Main Features

- Supply Manager
  - Main supply Low voltage detection (LVD)
  - Global power down
- Reset Sequence Manager (RSM)
- Multi-Oscillator (MO)
  - 4 Crystal/Ceramic resonator oscillators
  - 2 External RC oscillators
  - 1 Internal RC oscillator
- Clock Security System (CSS)
  - Clock Filter
  - Backup Safe Oscillator
- Main Clock controller (MCC)

Figure 8. Clock, Reset and Supply Block Diagram



4.2 LOW VOLTAGE DETECTOR (LVD)

To allow the integration of power management features in the application, 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 the Figure .

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 $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

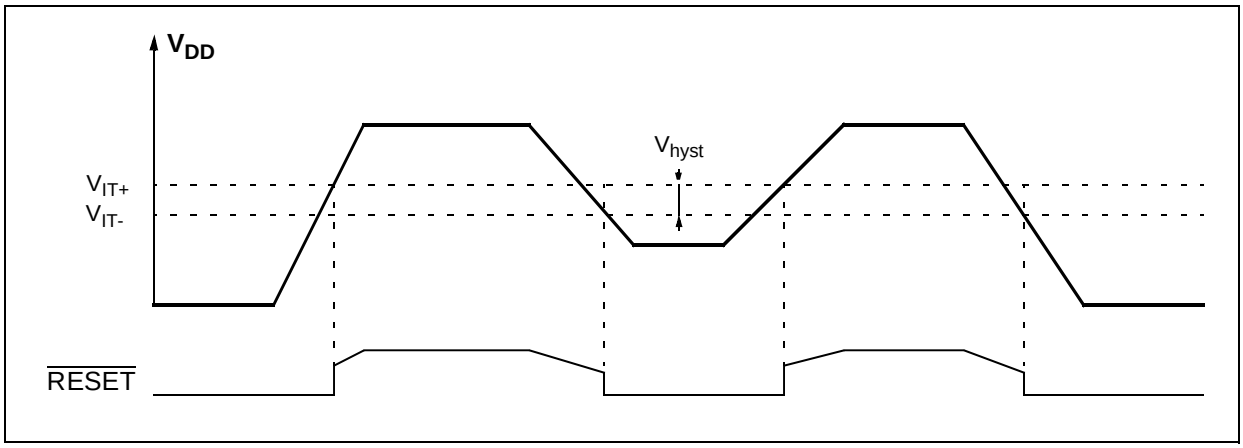
Notes:
1. The LVD allows the device to be used without any external RESET circuitry.
2. Three different reference levels are selectable through the option byte according to the application requirement.

LVD application note

Application software can detect a reset caused by the LVD by reading the LVDRF bit in the CRSR register.

This bit is set by hardware when a LVD reset is generated and cleared by software (writing zero).

Figure 9. Low Voltage Detector vs Reset



4.2.1 Reset Sequence Manager (RSM)

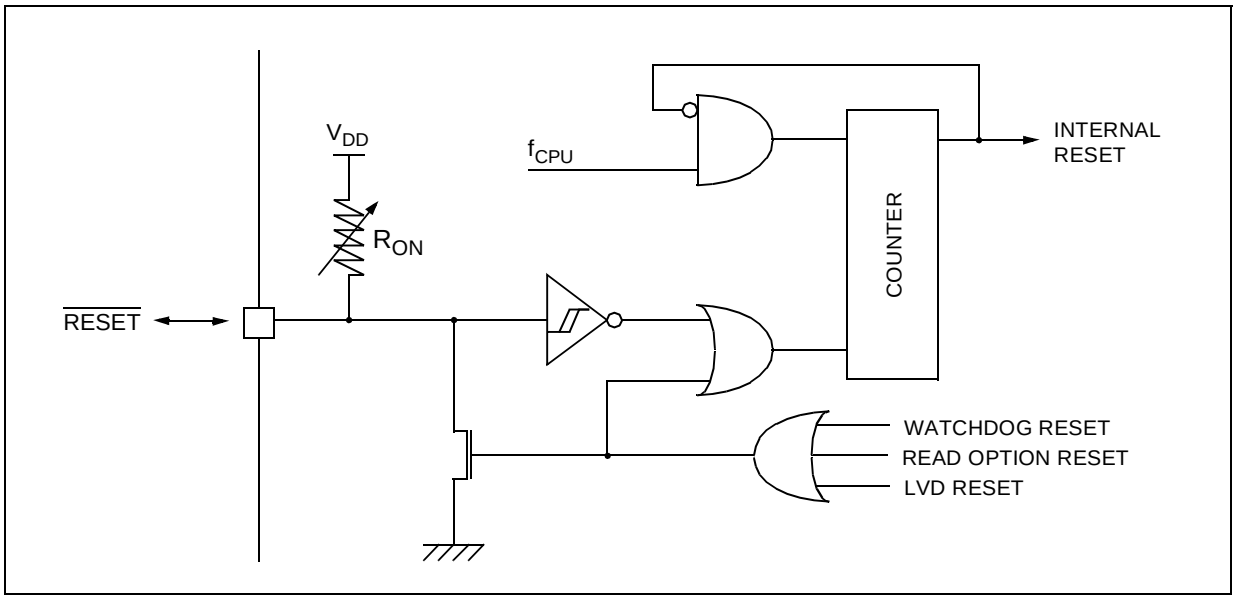
The RSM block of the CROSS Module includes three RESET sources as shown in Figure 10:

- EXTERNAL $\overline{\text{RESET}}$ SOURCE pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

These sources act on the $\overline{\text{RESET}}$ PIN and it is always kept low during the READ OPTION RESET phase.

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

Figure 10. Reset Block Diagram



The basic RESET sequence consists of 4 phases as shown in Figure 11:

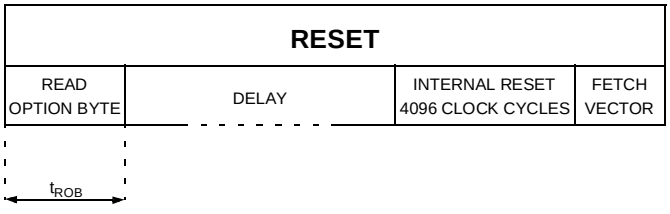
- OPTION BYTE reading to configure the device
- Delay depending on the RESET source
- 4096 cpu clock cycle delay
- RESET vector fetch

The duration of the OPTION BYTE reading phase (t_{ROB}) is defined in the Electrical Characteristics section. This first phase is initiated by an external $\overline{\text{RESET}}$ pin pulse detection, a Watchdog RESET detection, or when V_{DD} rises up to V_{LVDopt}.

The 4096 cpu clock cycle delay allows the oscillator to stabilise and to ensure that recovery has taken place from the Reset state.

The RESET vector fetch phase duration is 2 clock cycles.

Figure 11. RESET Sequence Phases



RESET SEQUENCE MANAGER (Cont'd)

4.2.2 Asynchronous External $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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 12).

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

4.2.3 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 $\overline{\text{RESET}}$ 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 12.

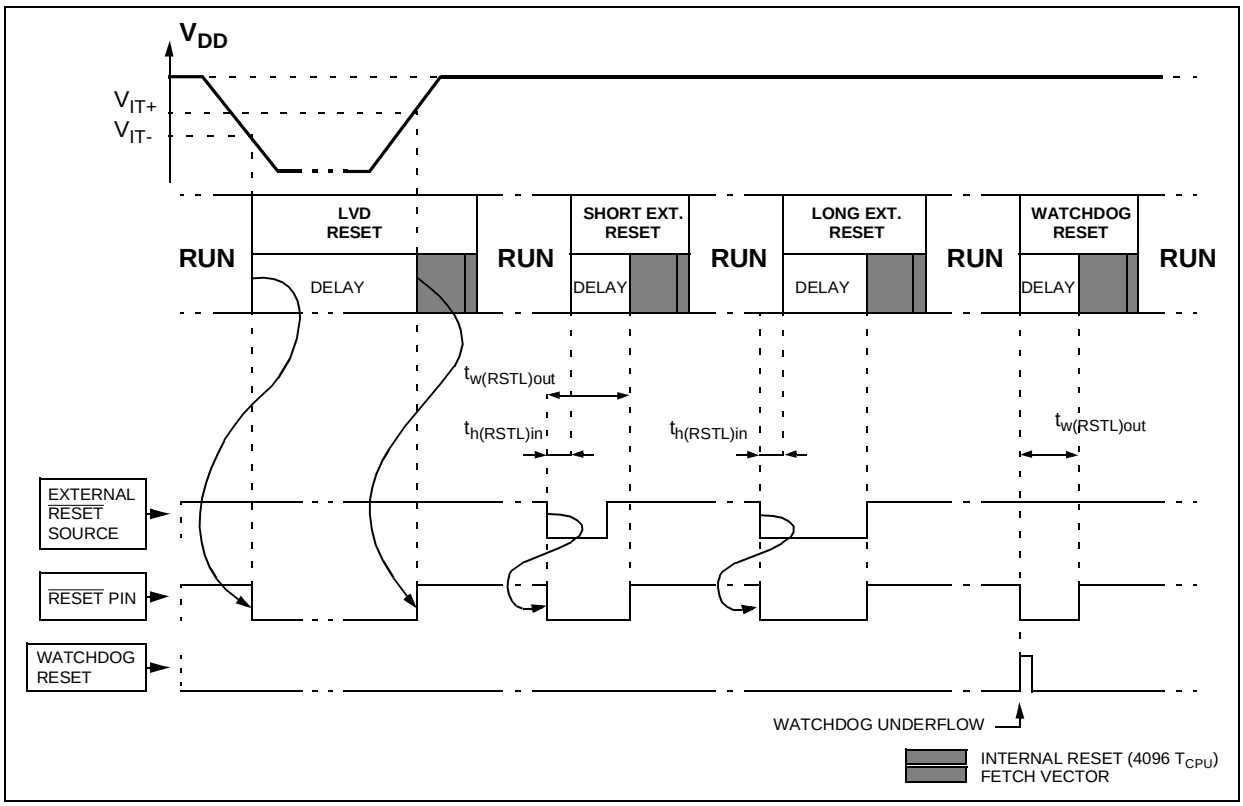
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

4.2.4 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 12.

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

Figure 12. RESET Sequences



4.2.4.1 Multi-Oscillator (MO)

The Multi-Oscillator (MO) block is the main clock supplier of the ST7. To insure an optimum integration in the application, it is based on an external clock source and six different selectable oscillators.

The main clock of the ST7 can be generated by 8 different sources comming from the MO block:

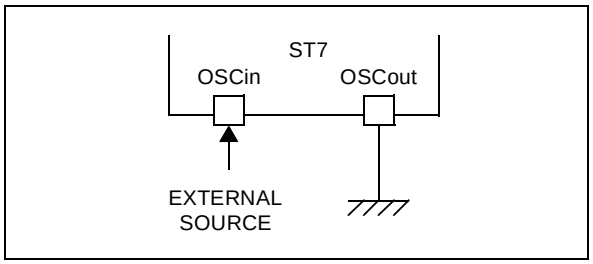
- an External source
- 4 Crystal or Ceramic resonator oscillators
- 1 External RC oscillators
- 1 Internal High Frequency RC oscillator

Each oscillator is optimized for a given frequency range in term of consumption and is selectable through the Option Byte.

External Clock Source

The default Option Byte value selects the External Clock in the MO block. In this mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSCin pin while the OSCout pin is tied to ground (see Figure 13).

Figure 13. MO External Clock

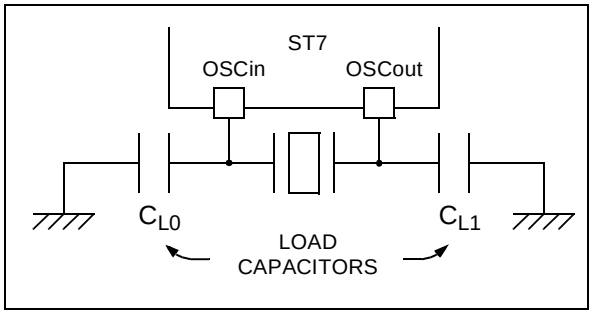


Crystal/Ceramic Oscillators

This family of oscillators allows a high accuracy on the main clock of the ST7. The selection within the list of 4 oscillators has to be done by Option Byte according to the resonator frequency in order to reduce the consumption. In this mode of the MO block, the resonator and the load capacitors have to be connected as shown in Figure 14 and have to be mounted as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time.

These oscillators, when selected via the Option Byte, are not stopped during the RESET phase to avoid losing time in the oscillator starting phase.

Figure 14. MO Crystal/Ceramic Resonator



MULTIOSCILLATOR (MO) (Cont'd)

External RC Oscillator

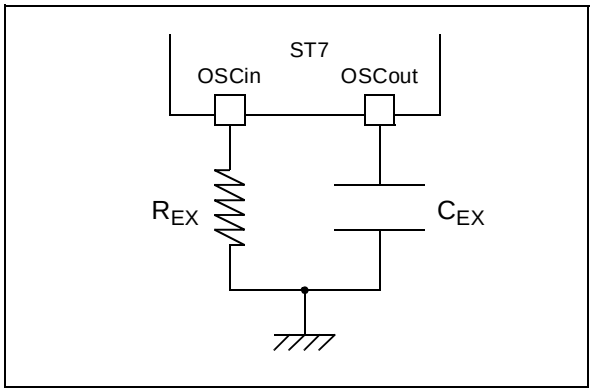
This oscillator allows a low cost solution on the main clock of the ST7 using only an external resistor and an external capacitor (see Figure 15). The selection of the external RC oscillator has to be done by Option Byte.

The frequency of the external RC oscillator is fixed by the resistor and the capacitor values:

$$f_{OSC} \sim \frac{N}{R_{EX} \cdot C_{EX}}$$

The previous formula shows that in this MO mode, the accuracy of the clock is directly linked to the accuracy of the discrete components.

Figure 15. MO External RC



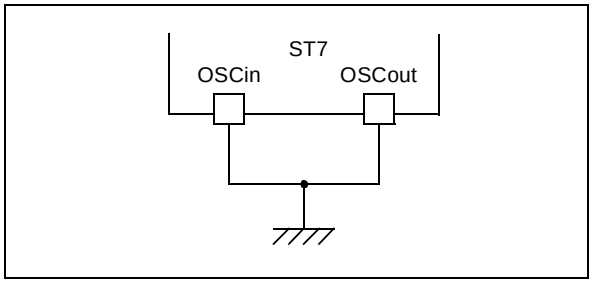
Internal RC Oscillator

The Internal RC oscillator mode is based on the same principle as the External RC one including the an on-chip resistor and capacitor. 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 as shown in Figure 16.

The selection of the internal RC oscillator has to be done by Option Byte.

Figure 16. MO Internal RC



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.

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.

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 CRSR register. An interrupt can be generated if the CSSIE bit has been previously set.
These two bits are described in the CRSR register description.

4.3.3 Low Power Modes

Mode	Description
WAIT	No effect on CSS. CSS interrupt 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.

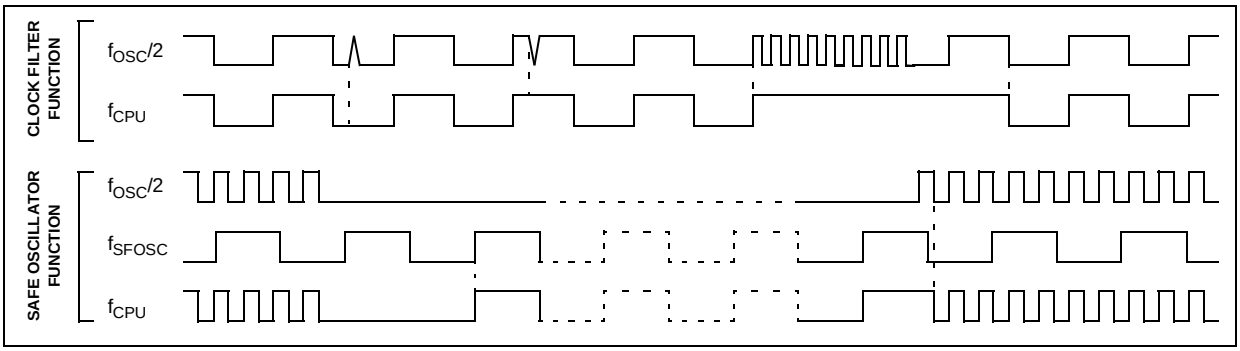
4.3.4 Interrupts

The CSS interrupt event generates an interrupt if the corresponding Enable Control Bit (CSSIE) 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[1)]
CSS event detection (safe oscillator activated as main clock)	CSSD	CSSIE	Yes	No

Note 1: This interrupt allows to exit from active-halt mode if this mode is available in the MCU.

Figure 17. Clock Filter Function and Safe Oscillator Function



4.3.5 Main Clock Controller (MCC)

The MCC block supplies the clock for the ST7 CPU and its internal peripherals. It allows the power saving modes such as SLOW mode to be managed by the application.

All functions are managed by the Miscellaneous Register 1 (MISCR1).

The MCC block consists of:

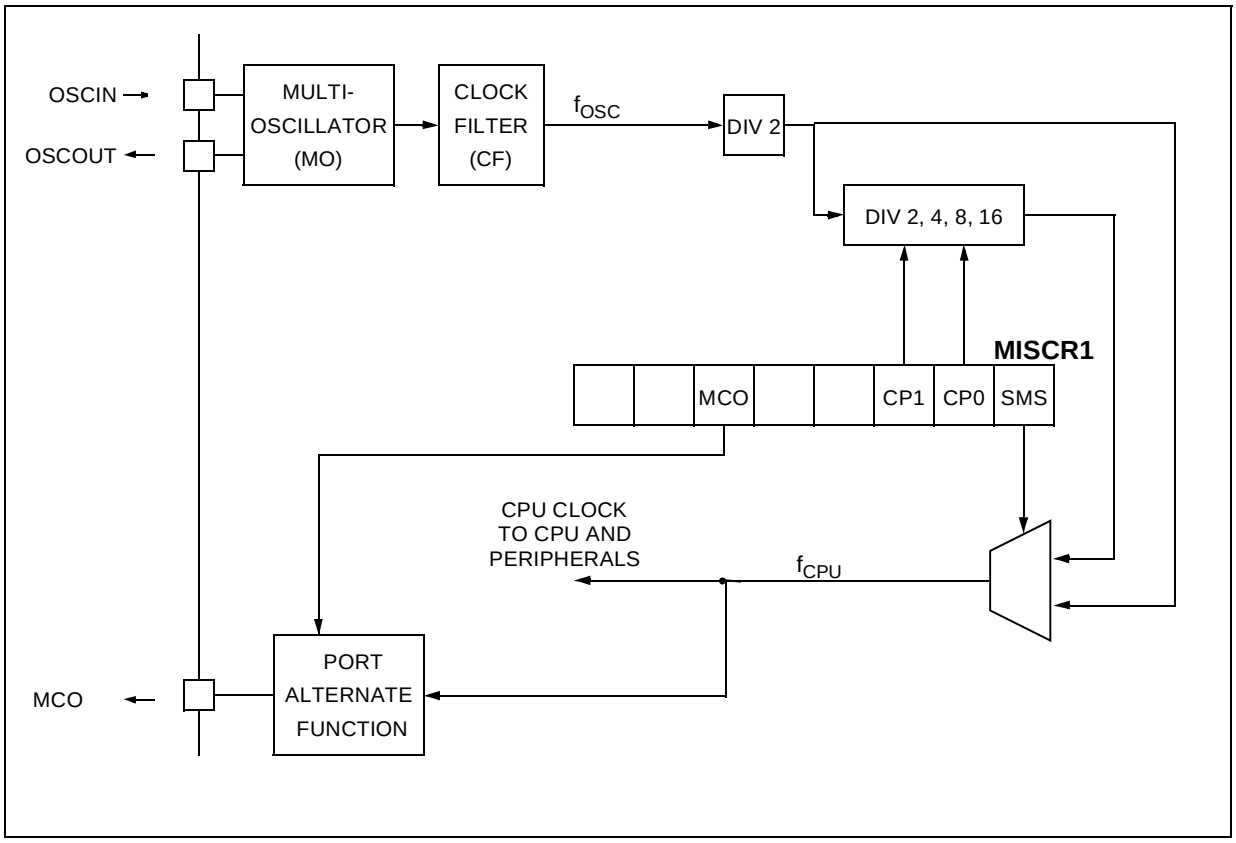
– a programmable CPU clock prescaler

– a clock-out signal to supply external devices

The prescaler allows the selection of the main clock frequency and is controlled with three bits of the MISCR1: CP1, CP0 and SMS.

The clock-out capability is an Alternate Function of an I/O port pin, providing the f_{CPU} clock as an output for driving external devices. It is controlled by the MCO bit in the MISCR1 register.

Figure 18. Main Clock Controller (MCC) Block Diagram



4.4 CLOCK, RESET AND SUPPLY REGISTER DESCRIPTION

CLOCK RESET AND SUPPLY REGISTER (CRSR)

Read/Write

Reset Value: 000x 000x (00h)

7							0
-	-	-	LVD RF	-	CSS IE	CSS D	WDG RF

Bit 7:5 = Reserved.

Bit 4 = **LVDRF** *LVD Reset Flag*
This bit indicates when set that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero) or a Watchdog Reset. See WDGRF flag description for more details.

Bit 3 = Reserved.

Bit 2 = **CSSIE** *CSS Interrupt Enable*
This bit allows to enable the interrupt when a distrurbance is detected by the Clock Security System (CSSD bit set). It is set and cleared by software.
0: Clock Filter interrupt disable
1: Clock Filter interrupt enable

Bit 1 = **CSSD** *CSS Safe Osc. Detection*
This bit indicates that the safe oscillator of the CSS block has been selected. It is set by hardware and cleared by reading the CRSR register when the original oscillator recovers.
0: Safe oscillator is not active
1: Safe oscillator has been activated

Bit 0 = **WDGRF** *WatchDog Reset Flag*
This bit indicates when set 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.
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

Table 3. Supply, Reset and Clock Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0020h	MISCR Reset Value	PEI3 0	PEI2 0	MCO 0	PEI1 0	PEI0 0	CP1 0	CP0 0	SMS 0
0025h	CRSR Reset Value	- 0	- 0	- 0	LVDRF x	- 0	CSSIE 0	CSSD 0	WDGRF x

5 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in the Interrupt Mapping Table and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 19.
The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

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

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

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

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

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

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

Priority Management

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

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

Interrupts and Low Power Mode

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

5.1 NON MASKABLE SOFTWARE INTERRUPT

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on Figure 19.

5.2 EXTERNAL INTERRUPTS

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

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

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

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

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

5.3 PERIPHERAL INTERRUPTS

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

– The I bit of the CC register is cleared.

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

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

Clearing an interrupt request is done by:

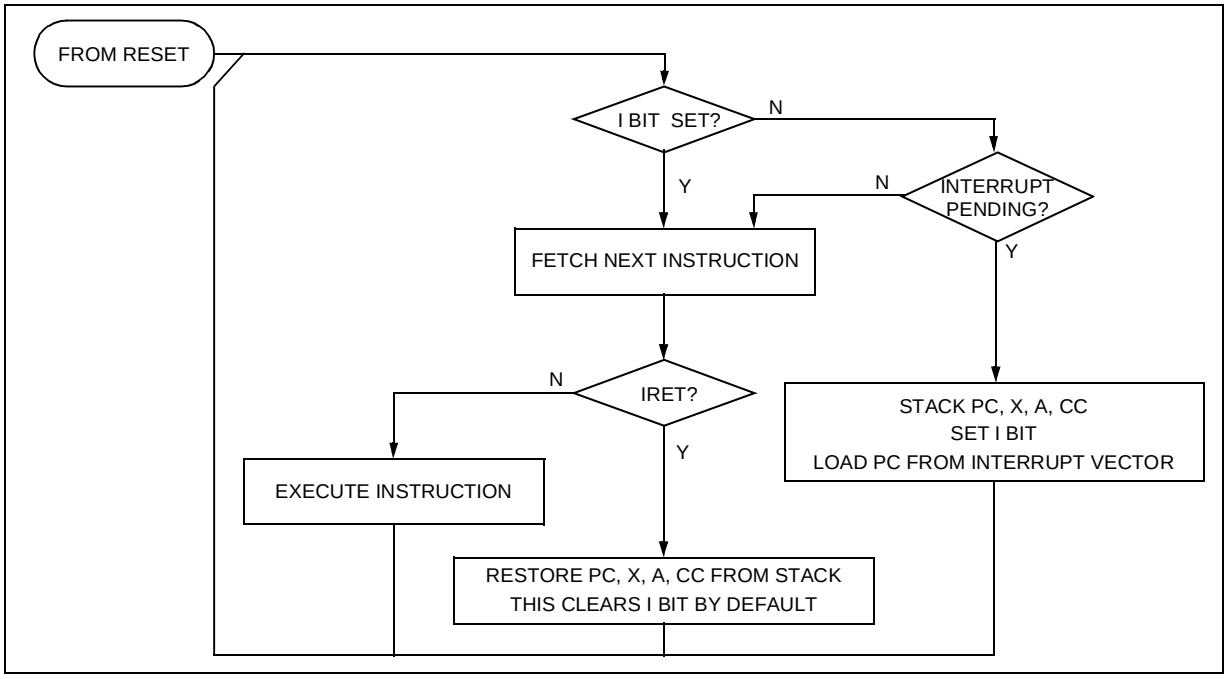
– Writing “0” to the corresponding bit in the status register or

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

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

INTERRUPTS (Cont'd)

Figure 19. Interrupt Processing Flowchart



INTERRUPTS (Cont'd)

Table 4. Interrupt Mapping

Source Block	Description	Register Label	Flag	Exit from HALT	Vector Address	Priority Order
RESET	Reset	N/A	N/A	yes	FFFEh-FFFFh	Highest Priority
TRAP	Software	N/A	N/A	no	FFFCh-FFFDh	
ei0	Ext. Interrupt ei0	N/A	N/A	yes	FFFAh-FFFBh	
ei1	Ext. Interrupt ei1	N/A	N/A	yes	FFF8h-FFF9h	
CSS	Clock Filter Interrupt	CRSR	CSSD	no	FFF6h-FFF7h	
SPI	Transfer Complete	SPISR	SPIF	no	FFF4h-FFF5h	
	Mode Fault		MODF			
TIMER 16	Input Capture 1	TASR	ICF1_1	no	FFF2h-FFF3h	
	Output Compare 1		OCF1_1			
	Input Capture 2		ICF2_1			
	Output Compare 2		OCF2_1			
	Timer Overflow		TOF_1			
ART/PWM	Input Capture 1	ARTICCSR	ICF0	yes	FFF0h-FFF1h	
	Timer Overflow	ARTCSR	OVF		FFEEh-FFEFh	
OP-AMP	OA1 Interrupt	OIRR	OA1V	yes	FFECh-FFEDh	
	OA2 Interrupt		OA2V		FFEAh-FFEBh	
NOT USED					FFE6-FFE9	
SCI	SCI Peripheral Interrupts			no	FFE4-FFE5	
NOT USED					FFE0h-FFE3h	Lowest Priority

6 POWER SAVING MODES

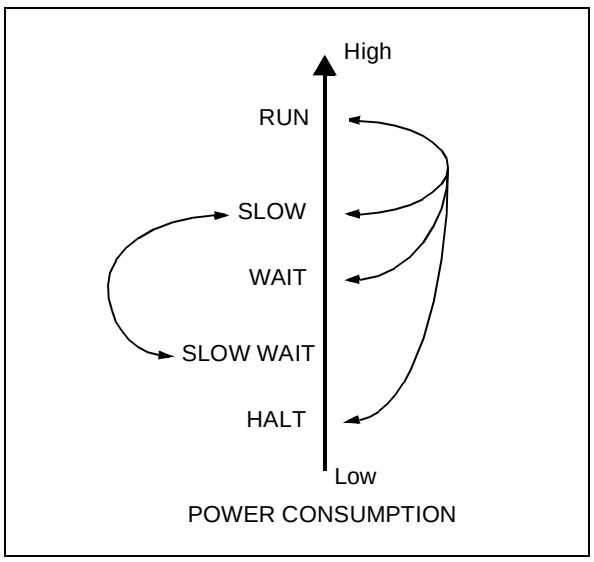
6.1 INTRODUCTION

To give a large measure of flexibility to the application in terms of power consumption, three main power saving modes are implemented in the ST7 (see Figure 20).

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 by 2 (f_{CPU}).

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 the oscillator status.

Figure 20. Power Saving Mode Transitions



6.2 SLOW MODE

This mode has two targets:

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

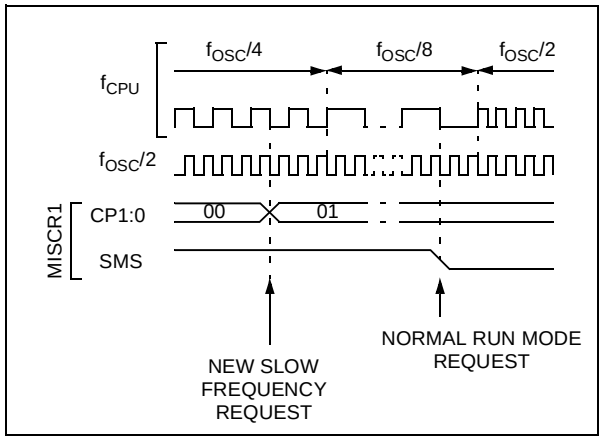
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by three bits in the MISCR1 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 oscillator frequency can be divided by 4, 8, 16 or 32 instead of 2 in normal operating mode. The CPU and peripherals are clocked at this lower frequency.

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

Figure 21. SLOW Mode Clock Transitions



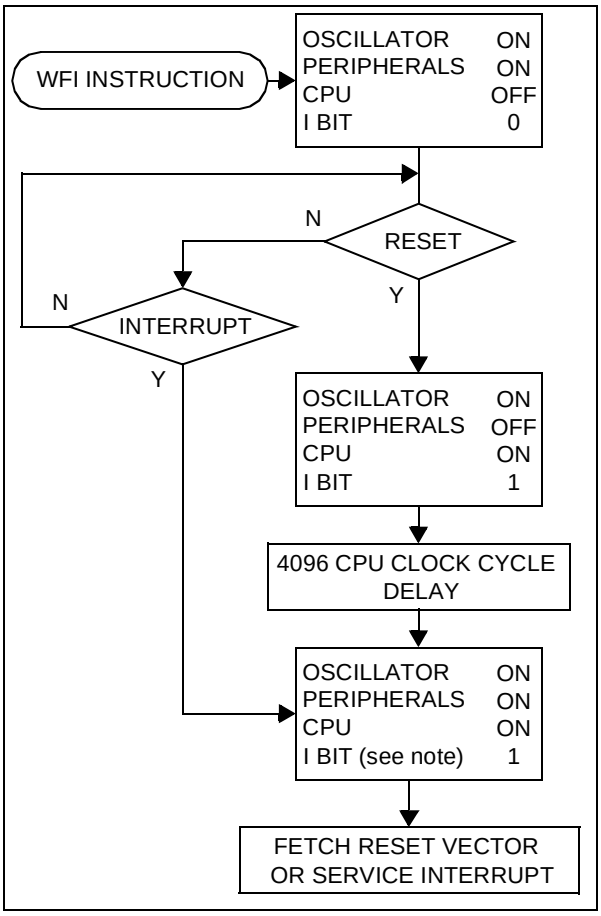
POWER SAVING MODES (Cont'd)

6.3 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" ST7 software instruction.
All peripherals remain active. During WAIT mode, the I bit of the CC register are forced to 0, 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 22.

Figure 22. WAIT Mode Flow-chart



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

POWER SAVING MODES (Cont'd)

6.4 HALT MODE

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the ST7 HALT instruction (see Figure 24).

The MCU can exit HALT mode on reception of either an specific interrupt (see Table 4, "Interrupt Mapping," on page 26) or a RESET. When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 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 23).

When entering HALT mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes immediately.

In the 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 11.1 OPTION BYTES for more details).

Figure 23. HALT Mode Timing Overview

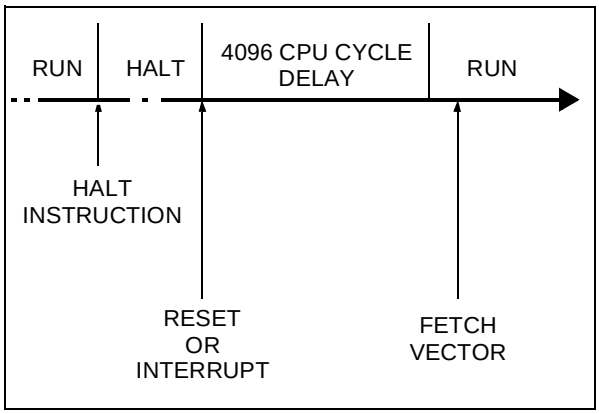
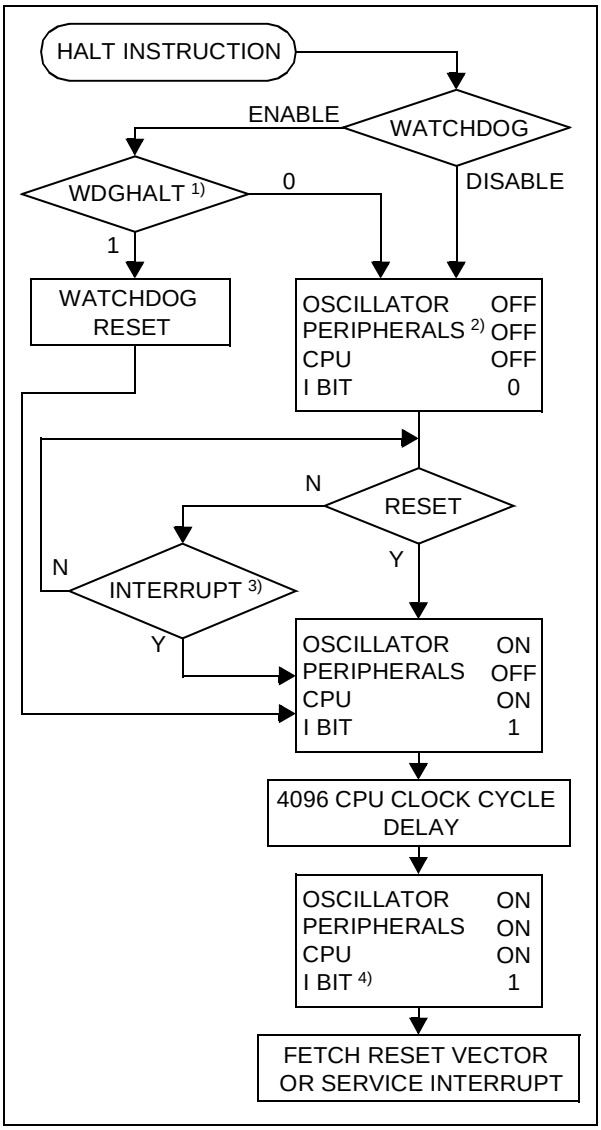


Figure 24. HALT Mode Flow-chart



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 4, "Interrupt Mapping," on page 26 for more details.

4. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

7 ON-CHIP PERIPHERALS

7.1 I/O PORTS

7.1.1 Introduction

The I/O ports offer different functional modes:

– transfer of data through digital inputs and outputs

and for specific pins:

– analog signal input (ADC)

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

– external interrupt generation

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

7.1.2 Functional Description

Each port is associated to 2 main registers:

– Data Register (DR)

– Data Direction Register (DDR)

and some of them to an optional register (see register description):

– Option Register (OR)

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

The following description takes into account the OR register, for specific ports which do not provide this register refer to the I/O Port Implementation Section 7.1.2.5. The generic I/O block diagram is shown on Figure 26.

7.1.2.1 Input Modes

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

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

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

Notes:

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

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

Interrupt function

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

Each pin can independently generate an Interrupt request.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see Interrupts section). If more than one input pin is selected simultaneously as interrupt source, this is logically ORed. For this reason if one of the interrupt pins is tied low, it masks the other ones.

7.1.2.2 Output Mode

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

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

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

7.1.2.3 Digital Alternate Function

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

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

Notes:

1. Input pull-up configuration can cause an unexpected value at the input of the alternate peripheral input.

2. When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

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

I/O PORTS (Cont'd)

7.1.2.4 Analog Alternate Function

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

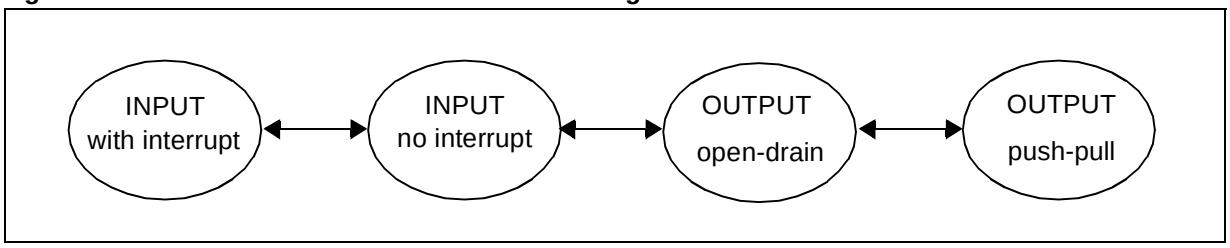
It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

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

7.1.2.5 I/O Port Implementation

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port such as ADC Input (see Figure 26) 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 25. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 25. Recommended I/O State Transition Diagram



I/O PORTS (Cont'd)

Figure 26. I/O Block Diagram

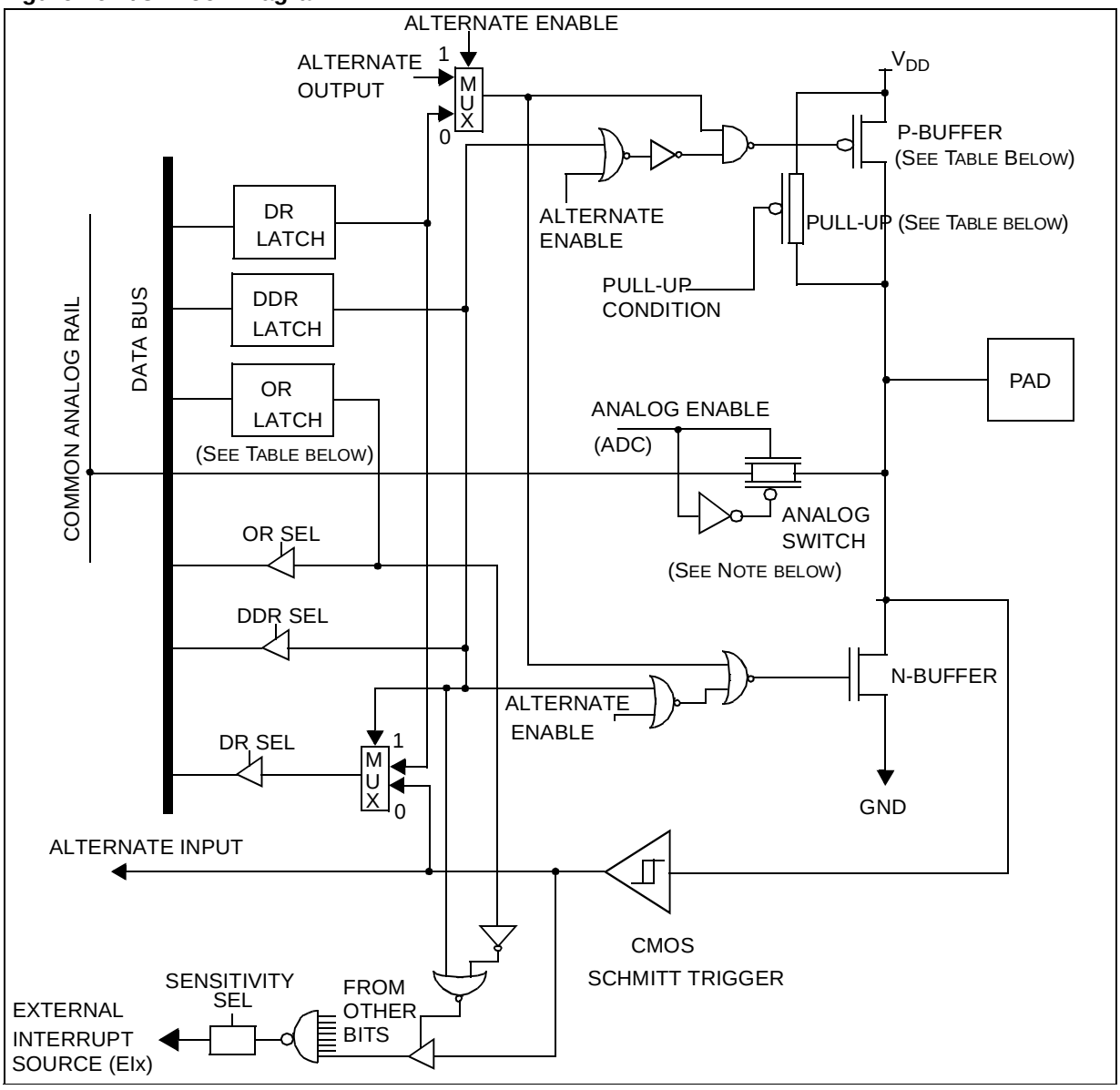


Table 5. Port Mode Configuration

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

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

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

I/O PORTS (Cont'd)

7.1.2.6 Device Specific Configurations

Table 6. Port Configuration

Port	Pin name	Input (DDR =0)		Output (DDR=1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7: PA0	floating*	pull-up with interrupt	open drain	push-pull
Port B	PB0:PB4	floating*	pull-up with interrupt	open drain high sink capability	push-pull
	PB5:PB7	floating*	pull-up with interrupt	open drain	push-pull
Port C	PC0:PC5	floating*	pull-up	open drain	push-pull

*Reset state.

I/O PORTS (Cont'd)

7.1.3 Register Description

DATA REGISTERS

Port A Data Register (PADR)

Port B Data Register (PBDR)

Port C Data Register (PCDR)

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

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = D[7:0] *Data Register 8 bits.*

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

DATA DIRECTION REGISTERS

Port A Data Direction Register (PADDR)

Port B Data Direction Register (PBDDR)

Port C Data Direction Register (PCDDR)

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

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = DD[7:0] *Data Direction Register 8 bits.*

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

0: Input mode
1: Output mode

OPTION REGISTERS

PORT A Option Register (PAOR)

PORT B Option Register (PBOR)

PORT C Option Register (PCOR)

Read/Write
Reset Value: 0000 0000 (00h) (no interrupt)

7							0
O7	O6	O5	O4	O3	O2	O1	O0

Bit 7:0 = **O[7:0]** *Option Register 8 bits.*

The PAOR, PBOR and PCOR registers are used to select pull-up or floating configuration in input mode.

Each bit is set and cleared by software.

Input mode:

0: Floating input
1: Input pull-up (with or without interrupt see Table 6)

I/O PORTS (Cont'd)

Table 7. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0000h	PADR	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0001h	PADDR	D7	D6	D5	DD4	DD3	DD2	DD1	DD0
	Reset Value	0	0	0	0	0	0	0	0
0002h	PAOR	D7	D6	D5	O4	O3	O2	O1	O0
	Reset Value	0	0	0	0	0	0	0	0
0004h	PBDR	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0005h	PBDDR	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
	Reset Value	0	0	0	0	0	0	0	0
0006h	PBOR	O7	O6	O5	O4	O3	O2	O1	O0
	Reset Value	0	0	0	0	0	0	0	0
0008h	PCDR	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0009h	PCDDR	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
	Reset Value	0	0	0	0	0	0	0	0
000Ah	PCOR	O7	O6	O5	O4	O3	O2	O1	O0
	Reset Value	0	0	0	0	0	0	0	0

7.2 MISCELLANEOUS REGISTERS

7.2.1 Miscellaneous Register 1 (MISCR1)

Miscellaneous register 1 is used select SLOW operating mode. Bits 3, 4, 6, and 7 determine the polarity of external interrupt requests.

Register Address: 0020h — Read/Write
Reset Value: 0000 0000 (00h)

7							0
PEI3	PEI2	MCO	PEI1	PEI0	CP1	CP0	SMS

Bit 7:6 = **PEI[3:2]** *Polarity Options of External Interrupt ei1. (Port B)*.
These bits are set and cleared by software. These bits determine which event causes the external interrupt (ei1) on port B according to Table 8.

Table 8. ei1 Ext. Int. Polarity Options

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

Bit 5 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the 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_{OSC}/2 on I/O port)

This bit is set and cleared by software. When set it can be used to output the internal clock to the dedicated I/O port.

Bit 4:3 = **PEI[1:0]** *Polarity Options of External Interrupt ei0. (Port A)*

These bits determine which event causes the external interrupt (ei0) on port A according to Table 9.

Table 9. ei0 Ext. Int. Polarity Options

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

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

Table 10. f_{CPU} Value in Slow Mode

f_{CPU} Value	CP1	CP0
f_{OSC} / 4	0	0
f_{OSC} / 8	1	0
f_{OSC} / 16	0	1
f_{OSC} / 32	1	1

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

7.2.2 Miscellaneous Register 2 (MISCR2)

Miscellaneous register 2 is used to configure of SPI and the output selection of the PWMs.

Register Address: 0040h — Read/Write

Reset Value: 0000 0000 (00h)

7							0
-	-	-	SPIOD	P1OS	P0OS	SSM	SSI

Bit 7:5 = not used

Bit 4 = **SPIOD** *SPI output disable*
This bit is used to disable the SPI output on the I/O port (in both master or slave mode).
0: SPI output enabled
1: SPI output disabled
(I/O pin free for general-purpose I/O)

Bit 3 = **P1OS** *PWM1 output select*
This bit is used to select the output for the PWM1 channel of the ART/PWM Timer.
0: PWM1 output on PWM1 pin
1: PWM1 output on PWM1R pin and connected to the OA2PIN pin

Note: In order to use the PC1 port pin as a PWM output pin, bit 1 of port C must be programmed as floating input. This should be done prior to setting the P1OS bit.

Bit 2 = **P0OS** *PWM0 output select*
This bit is used to select the output for the PWM0 channel of ART/PWM Timer.
0: PWM0 output on PWM0 pin
1: PWM0 output on PWM0R pin and connected to the OA1PIN pin

Note: In order to use the PC2 port pin as a PWM output pin, bit 2 of port C must be programmed as floating input. This should be done prior to setting the P0OS bit.

Bit 1 = **SSM** $\overline{SS}$ *mode selection*
It is set and cleared by software.
0: Normal mode - $\overline{SS}$ uses information coming from the $\overline{SS}$ pin of the SPI.
1: I/O mode, the SPI uses the information stored into bit SSI.

Bit 0 = **SSI** $\overline{SS}$ *internal mode*
This bit replaces pin $\overline{SS}$ of the SPI when bit SSM is set to 1. (see SPI description). It is set and cleared by software.

7.3 OP-AMP MODULE

7.3.1 Introduction

The ST7 Op-Amp module is designed to cover most types of microcontroller applications where analog signal amplifiers are used.

It may be used to perform a a variety of functions such as: differential voltage amplifier, comparator/ threshold detector, ADC zooming, impedance adaptor, general purpose operational amplifier.

7.3.2 Main features

This module includes:

- 2 rail-to-rail SPGAs (Software Programmable Gain Amplifier), and 1 stand alone rail-to-rail Op-Amp that may be externally connected using I/O pins
- A band gap voltage reference
- A programmable eight-step reference voltage
- ART Timer PWM outputs internally connected to SPGAs input 1 and 2.
- SPGAs and Op-Amp outputs are internally connected to the ADC inputs (Channel 8, 9 & 10).
- Input offset compensation

7.3.3 General description

The module contains two SPGAs (OA1 & OA2) and 1 stand alone operational amplifier (OA3) depending on the device package. OA1 and OA2 each have associated circuitry for input and gain selection. The third operational amplifier, OA3, without input and gain selection circuitry, is available in some devices (see device pin out description).

7.3.3.1 Inputs

The non-inverting input of OA1 or OA2 may be connected to an I/O pin, to the band-gap reference voltage, to an 8-step voltage reference or to the analog ground.

The eight-step voltage reference uses a resistive network in order to generate two voltages between 1/8 V_{DD} and V_{DD} (in 1/8 V_{DD} steps) that can be connected to the non-inverting input of the two SPGAs. These voltages may be used as programmable thresholds with the corresponding SPGA used as a comparator or, with the SPGA programmed to have a gain of 2, 4 or 8, they may be used for extending the ADC precision (analog zooming).

The 2 inverting inputs of OA1 or OA2 may be used to achieve this function. The input impedance of these inputs is around 2K.

The ART Timer PWM resistive outputs are internally connected to OA1PIN and OA2PIN pins. The PWM outputs are enabled by the PWMCR register and the resistive outputs are selected by Miscellaneous register 2. Refer to Figure 28.

The inverting input of OA1 or OA2 may be connected to an I/O pin, to the analog ground or may be left unconnected (in this case the SPGA can be used as a repeater, with the output of the SPGA connected to this input via the resistive loopback).

7.3.3.2 Outputs

The SPGA outputs are connected either to external pins or, internally, to the ADC input (Channel 8 & 9). The output value, digitized by a Schmitt trigger, may be read by the application software or may generate an interrupt.

The OA3 output is connected to an ADC input (Channel 10).

7.3.3.3 Advanced features

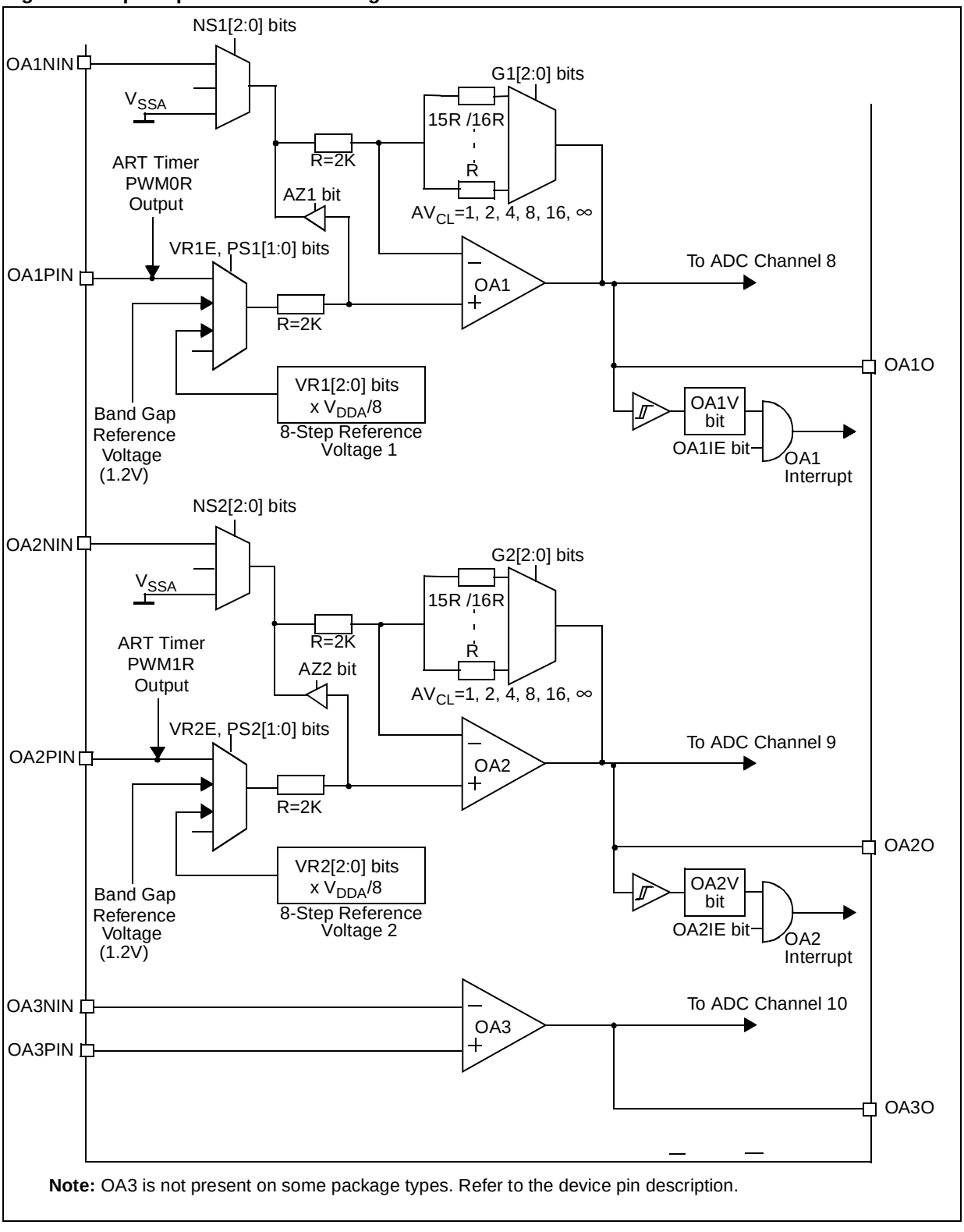
The gain of OA1 or OA2 is programmed using an internal resistive network. The possible values are: 1, 2, 4, 8 and 16. The internal resistive loopback may also be de-activated in order to obtain the open-loop gain (comparator) or to use the op-amp with an external loopback network.

Input offset compensation

In a special calibration mode (autozero mode), the negative input pin of OA1 or OA2 can be connected internally to the positive input pin. This mode allows the measurement of the input offset voltage of the SPGA using the ADC. This value may be stored in RAM and subsequently used for offset correction (for ADC conversions). Refer to Section 9.3.4.

OP-AMP MODULE (Cont'd**)**

Figure 27. Op-Amp Module Block Diagram



Note: OA3 is not present on some package types. Refer to the device pin description.

OP-AMP MODULE (Cont'd**)**

7.3.4 Autozero Mode

When the following description refers to both OA1 or OA2, x stands for 1 or 2.

In order to eliminate the ADC errors due to the SPGA offset voltage, this voltage may be determined, prior to the A/D conversion (at power on or periodically) and stored in RAM. The stored value may be used afterwards to eliminate the errors of any A/D conversion that uses the SPGA (ADC zooming). The measurement may be done independently for OA1 and OA2.

The measurement algorithm has 3 steps:

1. The SPGA is in repeater mode (NSx[1:0] = 01), with the lowest gain (Gx[2:0]=000), the autozeroing switch is left open (AZx = 0). The positive input of the op-amp is connected to a DC value, using the VRx reference voltage generator (PSx[1:0] = 00), and the output is sent to the ADC. Under these conditions, the ADC measures the value:

   Vo = VRx -Voff

   of the SPGA output.

2. Set the gain (G) according the application requirement. The AZx bit is set to 1. The output voltage of the SPGA becomes:

   V'o = VRx - Voff - G * Voff

3. Voff calculated with 1) - 2)

   Voff =(Vo- V'o) /G

As the offset voltage of the SPGAs may vary with the common mode voltage value, the measurement must be done choosing VRx to match the application conditions. Alternatively, nine measurements may be done with the noninverting input voltage varying between 0 and V_{DDA} in 1/8 V_{DD} steps, in order to fully characterise the offset voltage of the op-amp.

7.3.5 Comparator mode with Interrupts

The 2 SPGAs can be configured in comparator mode (GX[2:0]=111). In this case the positive input can be connected to the internal reference voltage. The negative input can be used to receive the analog voltage to be compared with the voltage connected to the positive input.

By means of a Schmitt trigger, the SPGA output is readable as a logical level in the OAxVR bit in the OAIRR register. These bits are read only.

An interrupt request remains pending as long as the output value (OAxVR) is equal to the corresponding polarity bit (OAxPR) and when the interrupt enable bit (OAxIE) is set. There is one interrupt vector for each SPGA.

OP-AMP MODULE (Cont'd)

7.3.6 DAC Function using ART Timer PWMR Outputs

The PWMR outputs are connected to a serial resistor and internally connected to the OA1PIN/ OA2PIN inputs. An external capacitor must be connected to the PWM0R/OA1PIN and/or PWM1R/OA2PIN pins (see Figure 28) if the PWMR outputs are used.

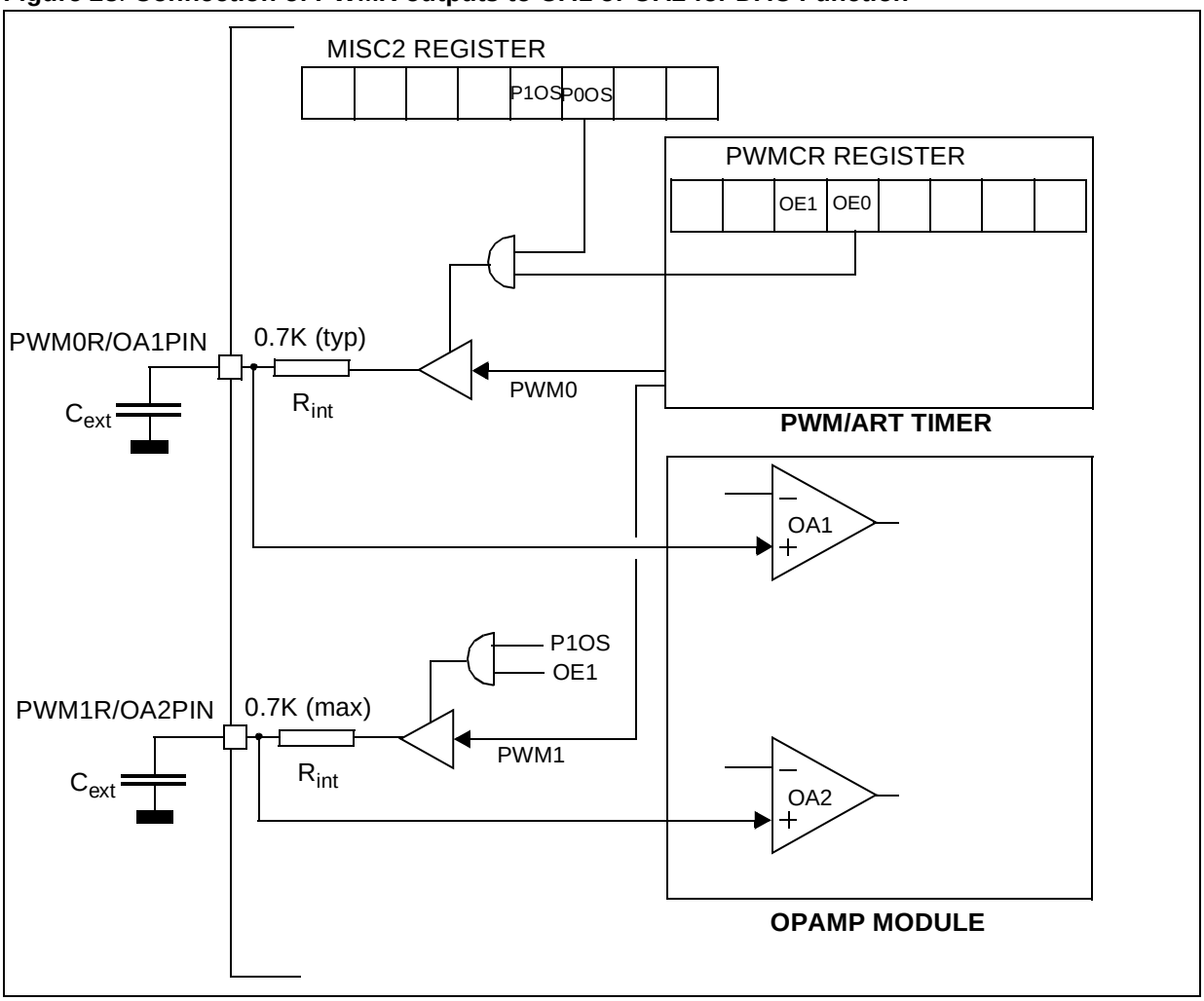
This feature allows the microcontroller to be used as a Digital to Analog converter and generating a DC voltage on the positive input pin, so the SPGAs may be used for the following functions:

– A comparator

– An amplifier of an external voltage connected to the negative input pin (OA1NIN or OA2NIN).

– A repeater, to obtain the same voltage on the OA output pin as on the input pin, with increased current capability.

Figure 28. **Connection of PWMR outputs to OA1 or OA2 for DAC Function**



OP-AMP MODULE (Cont'd**)**

7.3.7 Low Power Modes

Mode	Description
WAIT	No effect on op-amp. SPGA interrupts cause the device to exit from WAIT mode.
HALT	No effect on op-amp. SPGA interrupts cause the device to exit from HALT mode.

Note: Low Power modes have no effect on the SPGAs & the Op-Amp. They can be switched off to reduce the power consumption of the ST7 (OAxON bits).

7.3.8 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Op-Amp 1 output in comparator mode equals to OA1P bit value	NA*	OA1IE	Yes	Yes
Op-Amp 2 output in comparator mode equals to OA2P bit value	NA*	OA2IE	Yes	Yes

* The interrupt event occurs when the OAxP bit equals the OAxV bit value.

Note: The SPGA interrupt events are connected to 2 interrupt vectors (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).

7.3.9 Register Description

OA1 CONTROL REGISTER (OA1CR)
Read/Write
Reset value: 0000 00000 (00h)

7							0
AZ1	G12	G11	G10	PS11	PS10	NS11	NS10

Bit 7 = **AZ1** *OA1 Autozero Mode.*

This bit is set and reset by hardware. It enables Autozero mode (used to measure the OA1 input offset).
0: Autozero mode disabled
1: Autozero mode enabled

Bit 6:4 = **G1[2:0]** *Gain Control.*
These bits are set and reset by software and control the OA1 gain by modifying the resistive loopback network. The value of the gain is adjusted to the desired value (for inverting / non-inverting amplification) corresponding to the selected positive input source - see PS1[1:0] table, Gain Adjust column.

Gain inv / Ninv	G12	G11	G10
-1 / 2	0	0	0
-2 / 3	0	0	1
-3 / 4	0	1	0
-4 / 5	0	1	1
-8 / 8	1	0	0
-16 / 16	1	0	1
Comparator External Loopback	1	1	1

Bit 3:2 = **PS1[1:0]** *Positive Input Select / Gain adjust***.**

These bits are set and reset by software and control the OA1 positive input selection.

OA1 Positive Input	Gain Adj.	PS11	PS10
8-step Ref.Voltage 1	inv	0	0
OA1PIN	ninv	0	1
Band Gap Ref. Voltage (1.2V)	inv	1	0

Bit 1:0 = **NS1[1:0]** *Negative Input Select.*

These bits are set and reset by software and control the OA1 positive input selection.

OA1 Negative Input	NS11	NS10
AGND	0	0
Floating - Repeater mode	0	1
OA1NIN	1	X

OA2 CONTROL REGISTER (OA2CR)
Read/Write
Reset value: 0000 0000 (00h)

7							0
AZ2	G22	G21	G20	PS21	PS20	NS21	NS20

Bit 7 = **AZ2** *OA2 Autozero Mode.*

This bit is set and reset by hardware. It enables Autozero mode (used to measure the OA2 input offset).
0: Autozero mode disabled
1: Autozero mode enabled

Bit 6:4 = **G2[2:0]** *Gain Control.*
These bits are set and reset by software and control the OA2 gain by modifying the resistive loopback network. The value of the gain is adjusted to the desired value (for inverting/noninverting amplification) corresponding to the selected positive input source - see PS2[1:0] table, Gain Adjust column.

Gain inv / Ninv	G22	G21	G20
-1 / 2	0	0	0
-2 / 3	0	0	1
-3 / 4	0	1	0
-4 / 5	0	1	1
-8 / 8	1	0	0
-16 / 16	1	0	1
Comparator External Loopback	1	1	1

OP-AMP MODULE (Cont'd**)**

Bit 3:2 = **PS2[1:0]** *Positive Input Select / Gain adjust.*

These bits are set and reset by software and control the OA2 positive input selection.*t*

OA2 Positive Input	Gain Adj.	PS21	PS20
8-step Ref.Voltage 1	inv	0	0
OA2PIN	ninv	0	1
Band Gap Ref. Voltage (1.2V)	inv	1	0
Floating	ninv	1	1

Bit 1:0 = **NS2[1:0]** *Negative Input Select.*

These bits are set and reset by software and control the OA2 negative input selection.

OA2 Negative Input	NS21	NS20
AGND	0	0
Floating -Repeater mode	0	1
OA2NIN	1	X

OA3 CONTROL REGISTER (OA3CR)

Read/Write

Reset value: 0000 0000 (00h)

7							0
OA3ON	-	-	-	-	-	-	-

Bit 7 = **OA3ON** *OA3 on/off (low power)*

Stand Alone Op-Amp on/off control bit, it is set and reset by software. It reduces power consumption when reset.

0: Op-amp 3 off

1: Op-amp 3 on

Note: This bit must be kept cleared in devices without OA3 (refer to device block diagram and pin description)

Bit 6:0 = Reserved.

OP-AMP MODULE (Cont'd**)**

OP-AMP INTERRUPT AND READOUT REGISTER (OAIRR)
Read/Write*
Reset value: 0000 0000 (00h)

7							0
OA1IE	OA1P	OA1V	OA1ON	OA2IE	OA2P	OA2V	OA2ON

Bit 7 = **OA1IE** *OA1 interrupt enable*
This bit is set and reset by software. When it is set, it enables an interrupt to be generated if the OA1P bit and the OA1V bit have the same value.
0: OA1 interrupt disabled
1: OA1 interrupt enabled

Bit 6 = **OA1P** *OA1 interrupt polarity select*
This bit is set and reset by software. It specifies the OA1 SPGA output level which will generate an interrupt if the bit OA1IE is set.
0: Active low
1: Active high

Bit 5 = **OA1V** *OA1 output value (read only)*
This bit is set and reset by hardware. It contains the OA1 SPGA output voltage value filtered by a Schmitt trigger.
0: OA1+ voltage < OA1- voltage
1: OA1+ voltage > OA1- voltage

Bit 4 = **OA1ON** *OA1 on/off (low power)*

This bit is set and reset by software. It reduces power consumption when reset.
0: Op-amp 1 off
1: Op-amp 1 on

Bit 3 = **OA2IE** *OA2 interrupt enable*

This bit is set and reset by software. When it is set, it enables an interrupt to be generated if the OA2P bit and the OA2V bit have the same value.
0: OA2 interrupt disabled
1: OA2 interrupt enabled

Bit 2 = **OA2P** *OA2 interrupt polarity select*

This bit is set and reset by software. It specifies the OA2 SPGA output level which will generate an interrupt if the bit OA2IE is set.
0: Active low
1: Active high

Bit 1- **OA2V** *OA2 output value (read only)*

This bit is set and reset by hardware. It contains the OA2 SPGA output voltage value filtered by a Schmitt trigger.
0: OA2+ voltage < OA2- voltage
1: OA2+ voltage > OA2- voltage

Bit 0 - **OA2ON** *OA2 on/off (low power)*
0: Op-amp 2 off (reducing power consumption)
1: Op-amp 2 on

Note: If OA1ON, OA2ON and OA3ON are 0, The entire module is disabled, giving the lowest power consumption.

* OA1V and OA2V are read only.

OP-AMP MODULE (Cont'd**)**

VOLTAGE REFERENCE CONTROL REGISTER (OAVRCR)
Read/Write
Reset value: 0000 0000 (00h)

7							0
VR2E	VR22	VR21	VR20	VR1E	VR12	VR11	VR10

Bit 7 = **VR2E**: *VR2 Enable*
This bit is set and reset by software. When the refererence voltage is selected (PS2[1:0] = 00 in the OA2CR register) it connects V_{SSA} (analog ground) or Reference Voltage 2 (VR2) to the OA2 positive input.
0: OA2 positive input is connected to V_{SSA}
1: OA2 positive input is connected to VR2 voltage value

Bit 6:4 = **VR2[2:0]** *Voltage selection for channel 2 of the 8-step reference voltage*
These bits are set and reset by software, they specify the Reference Voltage 2 (VR2) connected to the OA2 positive input when PS2[1:0] = 00 in the OA2CR register..

Reference Voltage 2	VR2E	VR22	VR21	VR20
0 (V_{SSA})	0	x	x	x
$V_{DDA}/8$	1	0	0	0
2 x $V_{DDA}/8$	1	0	0	1
3 x $V_{DDA}/8$	1	0	1	0
4 x $V_{DDA}/8$	1	0	1	1
5 x $V_{DDA}/8$	1	1	0	0
6 x $V_{DDA}/8$	1	1	0	1
7 x $V_{DDA}/8$	1	1	1	0
V_{DDA}	1	1	1	1

Bit 3= **VR1E** *VR1 Enable*
This bit is set and reset by software. When the refererence voltage is selected (PS1[1:0] = 00 in the OA1CR register) it connects V_{SSA} (analog ground) or Reference Voltage 1 (VR1) to the OA1 positive input.
0: OA1 positive input is connected to V_{SSA}
1: OA1 positive input is connected to VR1 voltage value

Bit 2:0 - **VR1[2:0]** *Voltage selection for channel 1 of the 8-step reference voltage*

These bits are set and reset by software, they specify the Reference Voltage 1 (VR1) connected to the OA1 positive input when PS1[1:0] = 00 in the OA1CR register.

Reference Voltage 1	VR1E	VR12	VR11	VR10
0 (V_{SSA})	0	x	x	x
$V_{DDA}/8$	1	0	0	0
2 x $V_{DDA}/8$	1	0	0	1
3 x $V_{DDA}/8$	1	0	1	0
4 x $V_{DDA}/8$	1	0	1	1
5 x $V_{DDA}/8$	1	1	0	0
6 x $V_{DDA}/8$	1	1	0	1
7 x $V_{DDA}/8$	1	1	1	
V_{DDA}	1	1	1	

Note: When both VR2E and VR1E are reset, the 8-step voltage reference cell is disabled and enters low power mode.

Table 11. OP-AMP Module Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
001Bh	OA1CR Reset Value	AZ1 0	G12 0	G11 0	G10 0	PS11 0	PS10 0	NS11 0	NS10 0
001Ch	OA2CR Reset Value	AZ2 0	G22 0	G21 0	G20 0	PS21 0	PS20 0	NS21 0	NS20 0
001Dh	OA3CR Reset Value	OA3ON 0	- 0	- 0	- 0	- 0	- 0	- 0	- 0
001Eh	OIRR Reset Value	OA1IE 0	OA1P 0	OA1V 0	OA2ON 0	OA2IE 0	OA2P 0	OA2V 0	OA1ON 0
001Fh	VRCR Reset Value	VR2E 0	VR22 0	VR21 0	VR20 0	VR1E 0	VR12 0	VR11 0	VR10 0

7.4 WATCHDOG TIMER (WDG)

7.4.1 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.

7.4.2 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.

7.4.3 Functional Description

The counter value stored in the CR register (bits T6:T0), 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 T6:T0) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

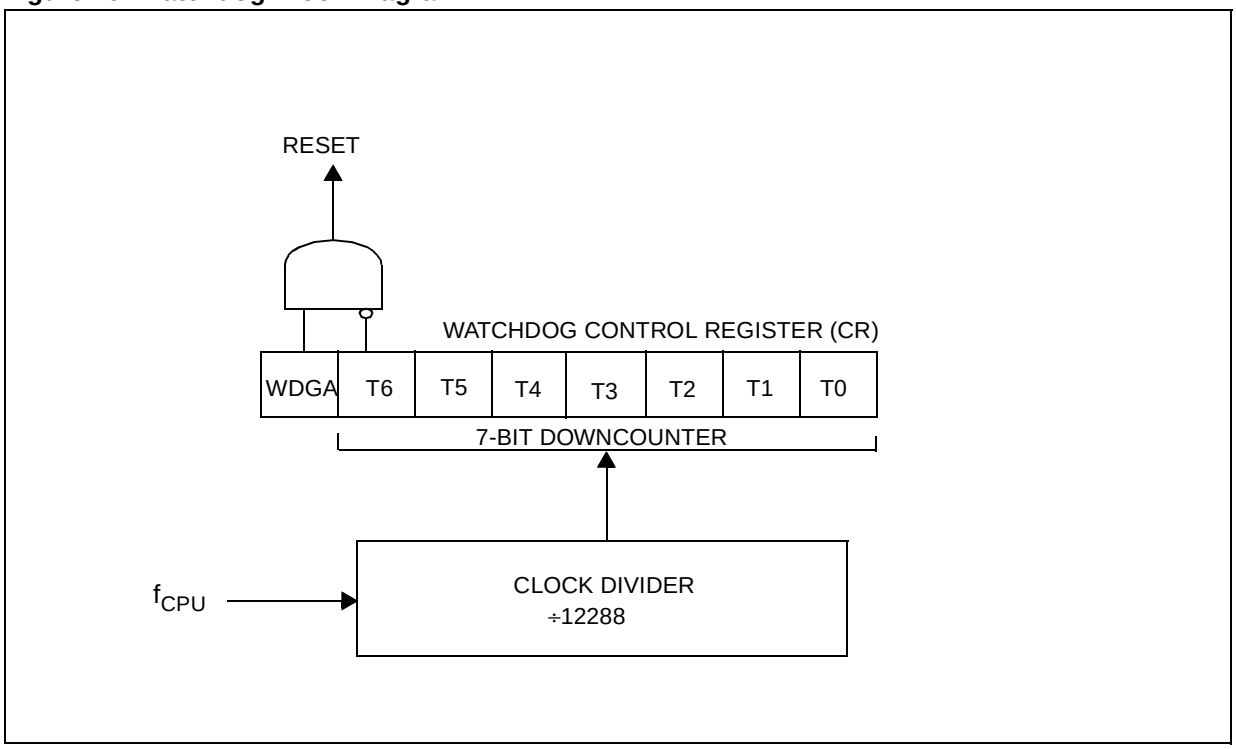
The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 13 . Watchdog Timing (fCPU = 8 MHz)):

– The WDGA bit is set (watchdog enabled)

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

– The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Figure 29. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

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

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

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

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

7.4.4 Hardware Watchdog Option

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

Refer to the device-specific Option Byte description.

7.4.5 Low Power Modes

WAIT Instruction

No effect on Watchdog.

HALT Instruction

If the Watchdog reset on HALT option is selected by option byte, a HALT instruction causes an immediate reset generation if the Watchdog is activated (WDGA bit is set).

7.4.5.1 Using Halt Mode with the WDG (option)

If the Watchdog reset on HALT option is not selected by option byte, the Halt mode can be used when the watchdog is enabled.

In this case, the HALT instruction stops the oscillator. When the oscillator is stopped, the WDG stops counting and is no longer able to generate a reset until the microcontroller receives an external interrupt or a reset.

If an external interrupt is received, the WDG restarts counting after 4096 CPU clocks. If a reset is generated, the WDG is disabled (reset state).

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 I bit 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).

7.4.6 Interrupts

None.

7.4.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

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

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).

WATCHDOG TIMER (Cont'd)

Table 13. WDG Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
24	CR	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1

7.5 16-BIT TIMER

7.5.1 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 measuring the pulse lengths of up to two input signals (*input capture*) or generating 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).

7.5.2 Main Features

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

The Block Diagram is shown in Figure 1.

***Note:** Some timer pins may not be 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'.

7.5.3 Functional Description

7.5.3.1 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).

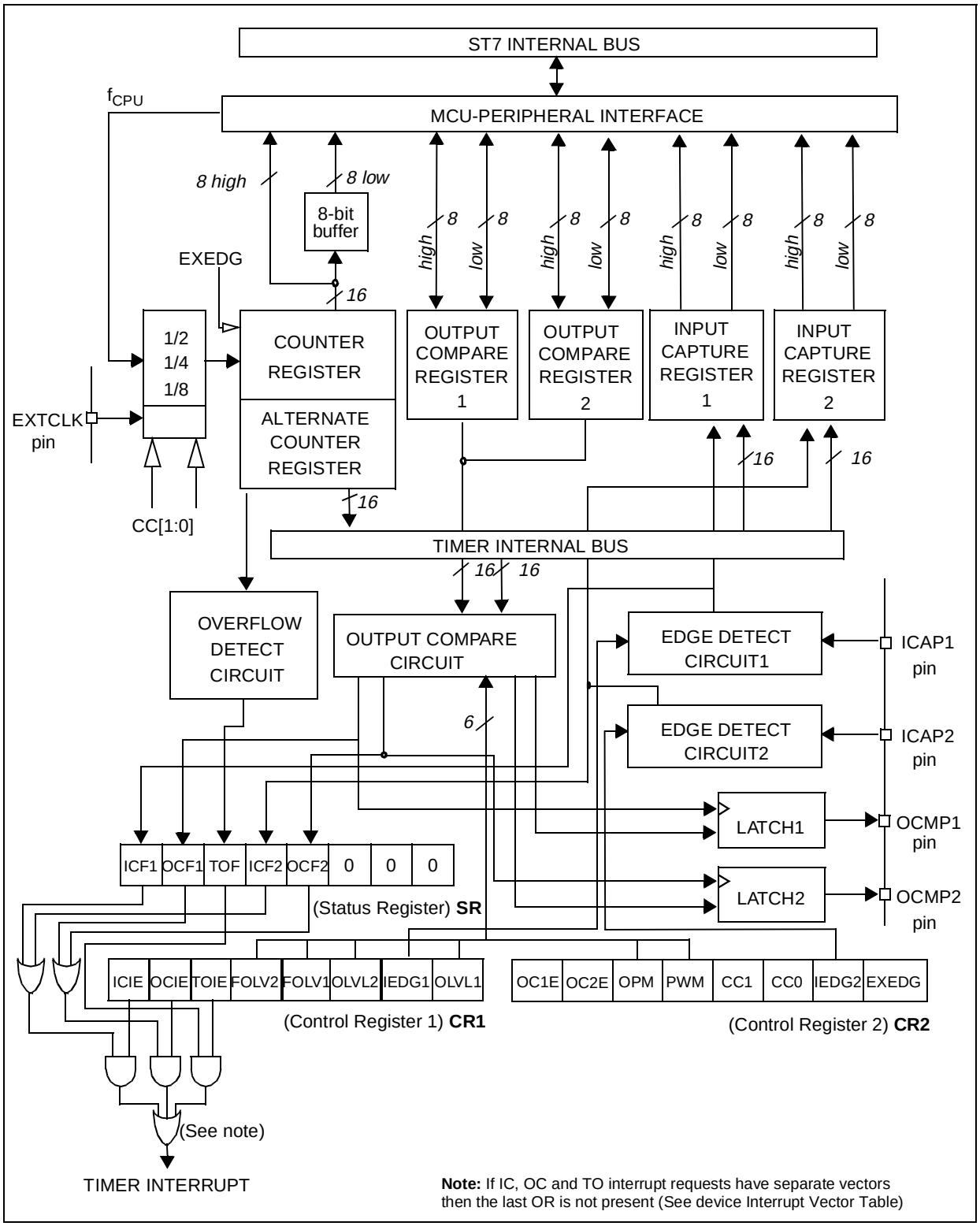
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register (SR). (See note at the end of paragraph titled 16-bit read sequence).

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 1. 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.

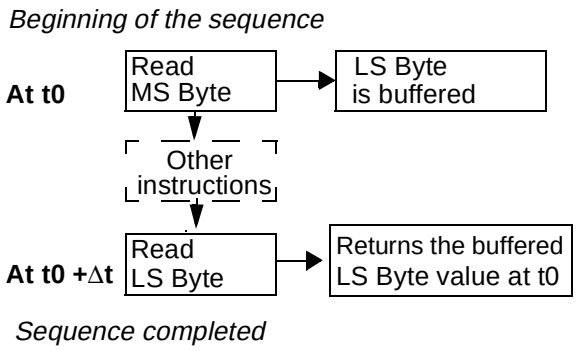
16-BIT TIMER (Cont'd)

Figure 30. Timer Block Diagram



16-BIT TIMER (Cont'd)

16-bit Read Sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the 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.

Note: The TOF bit is not cleared by accessing the 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).

7.5.3.2 External Clock

The external clock (where available) is selected if CC0=1 and CC1=1 in the 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 synchronised 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 31. Counter Timing Diagram, internal clock divided by 2

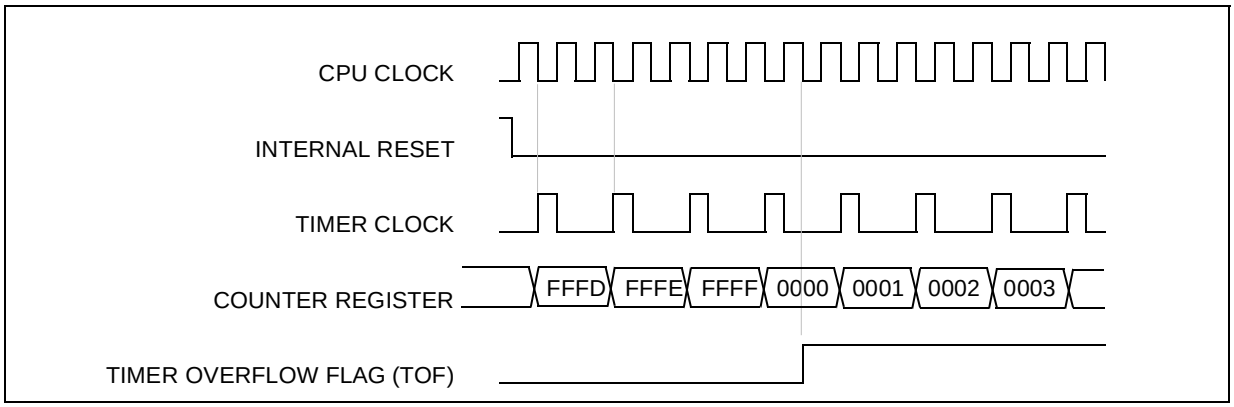


Figure 32. Counter Timing Diagram, internal clock divided by 4

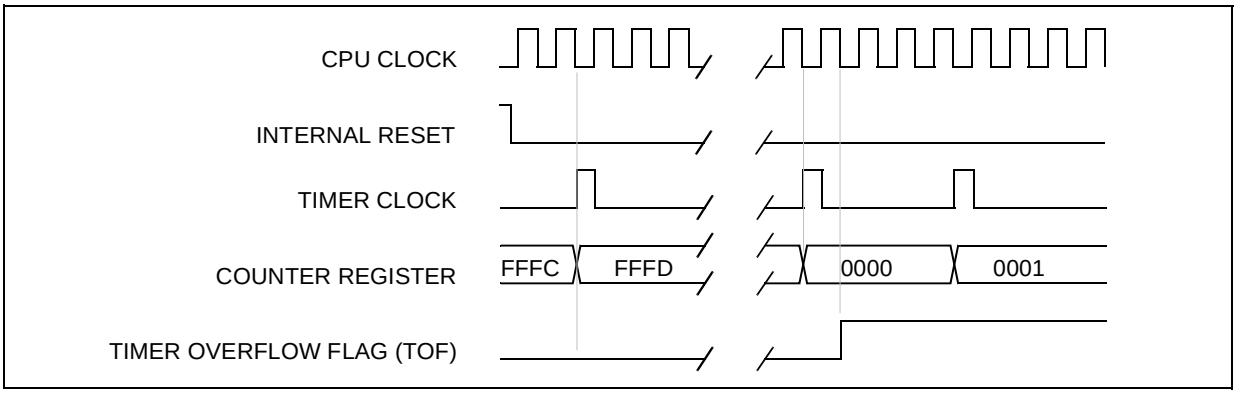
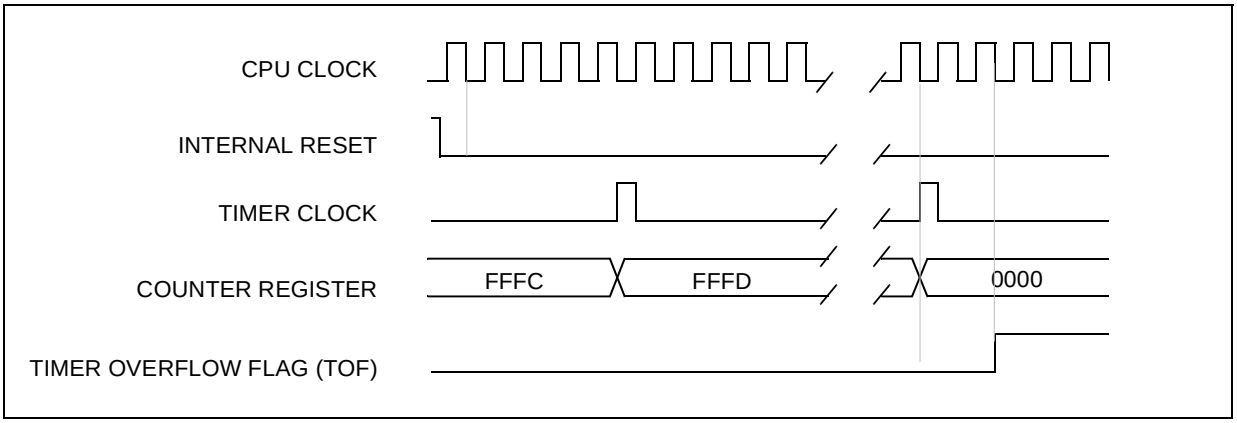


Figure 33. Counter Timing Diagram, internal clock divided by 8



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)

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

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

The 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 1).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as a 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 ICAP1pin must be configured as a floating input).

When an input capture occurs:

– The 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 6).

– 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, the 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 the input capture 2 function can be used.
5. The alternate inputs (ICAP1 & ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture function.
   Moreover if one of the ICAP*i* pin is configured as an input and the second one as an output, an interrupt can be generated if the user 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 an interrupt in order to measure events that exceed the timer range (FFFFh).

16-BIT TIMER (Cont'd)

Figure 34. Input Capture Block Diagram

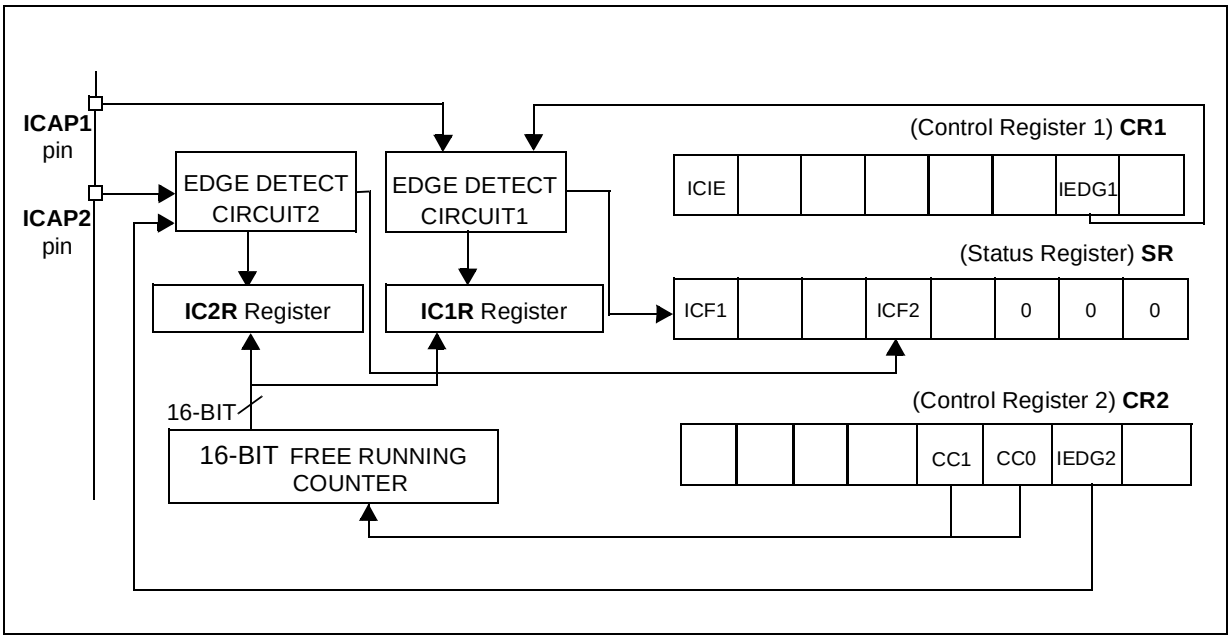
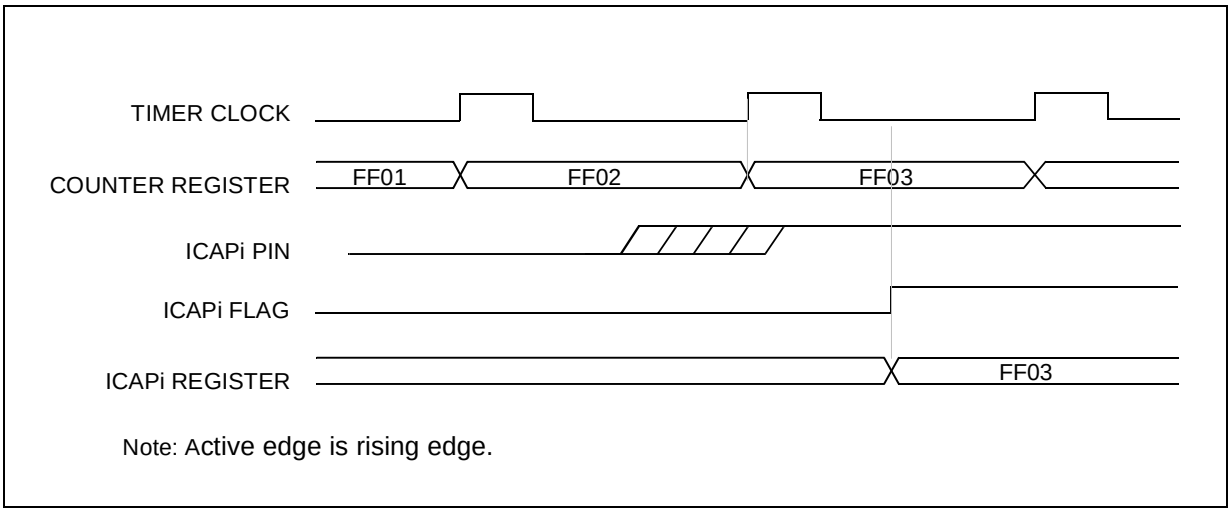


Figure 35. Input Capture Timing Diagram



Note: Active edge is rising edge.

16-BIT TIMER (Cont'd)

7.5.3.4 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 1).

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:

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

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 1)

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

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

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:

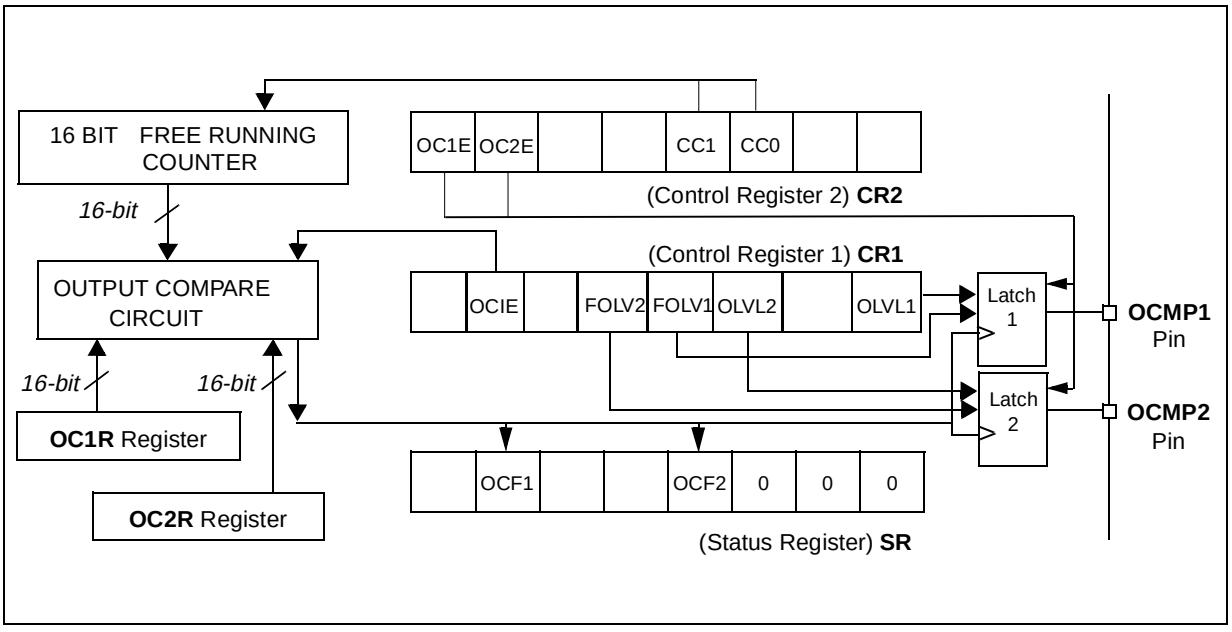
1. After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when a match is found but an interrupt could be generated 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 8). 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 9).
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.

FOLVL*i* bits have no effect in either One-Pulse mode or PWM mode.

Figure 36. Output Compare Block Diagram



16-BIT TIMER (Cont'd)

Figure 37. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/2$

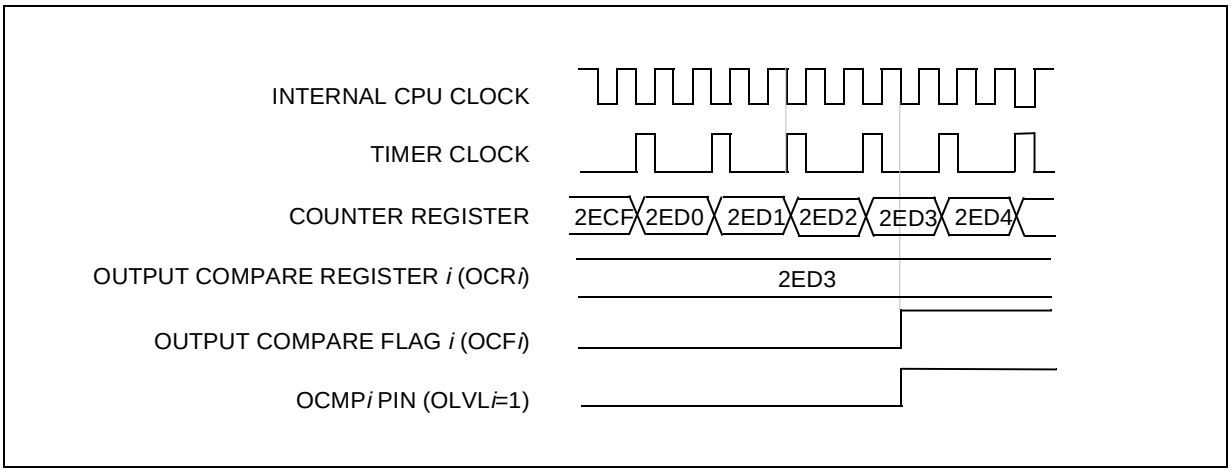
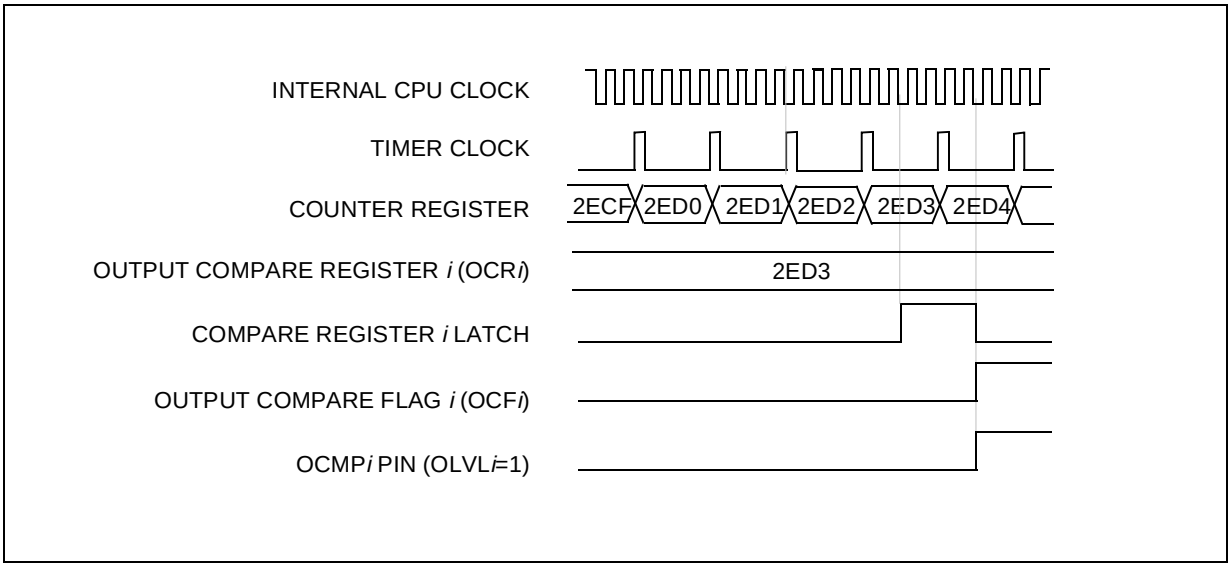


Figure 38. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/4$



16-BIT TIMER (Cont'd)

7.5.3.5 One Pulse Mode

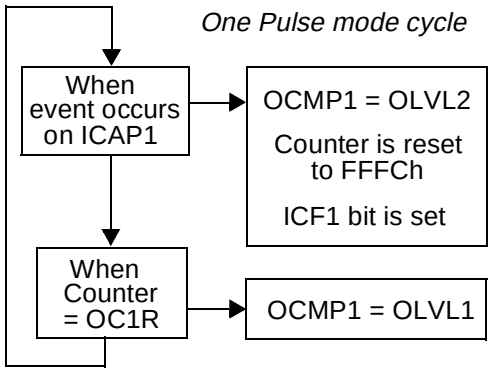
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

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 1).



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and the 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:

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

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 1)

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)

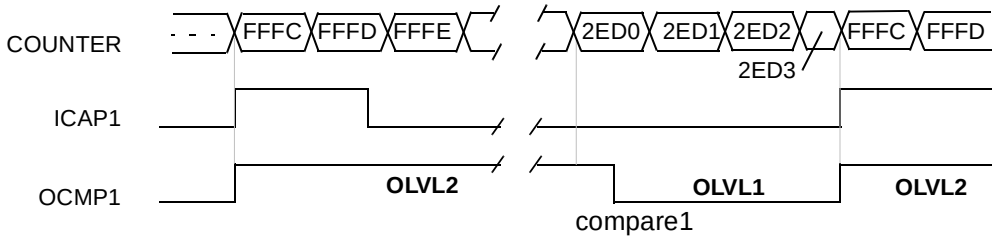
When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see Figure 10).

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 that a period of time has elapsed but cannot generate an output waveform because the OLVL2 level is dedicated to One Pulse mode.

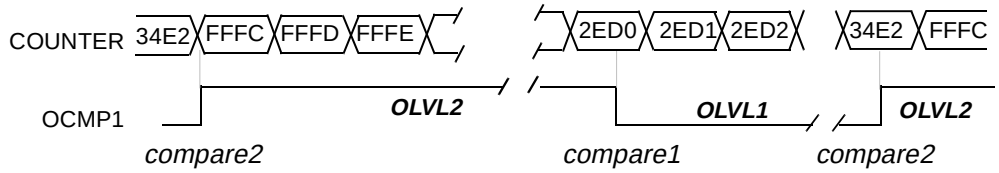
16-BIT TIMER (Cont'd)

Figure 39. One Pulse Mode Timing Example



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

Figure 40. Pulse Width Modulation Mode Timing Example



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

16-BIT TIMER (Cont'd)

7.5.3.6 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.

The Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so these functions cannot be used when the PWM mode is activated.

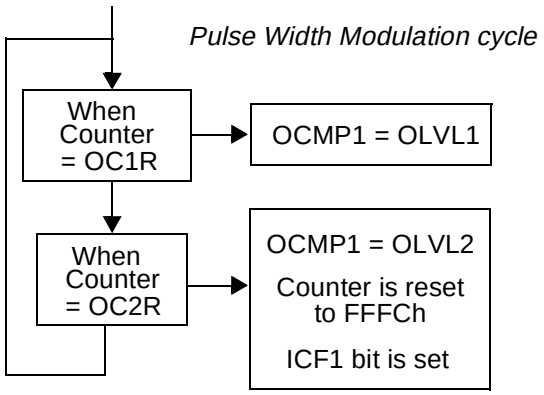
Procedure

To use Pulse Width Modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal 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 OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see Table 1).

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

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



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

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

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 1)

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

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

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 11)

Notes:

1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
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 from 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 after each period and ICF1 can also generate an 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)

7.5.4 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.

7.5.5 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).

7.5.6 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 0.1.3.5 One Pulse Mode

2) See note 5 in Section 0.1.3.5 One Pulse Mode

3) See note 4 in Section 0.1.3.6 Pulse Width Modulation Mode

16-BIT TIMER (Cont'd)

7.5.7 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 internal 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 internal 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.

Bits 3:2 = **CC[1:0]** *Clock Control.*

The timer clock mode depends on these bits:

Table 14. Clock Control Bits

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

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)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0

Bit 7 = **ICF1** *Input Capture Flag 1.*
0: No input capture (reset value).
1: An input capture has occurred 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 matches 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 has 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 matches the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

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

INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

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

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

7							0
MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

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

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

7							0
MSB							LSB

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

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

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

7							0
MSB							LSB

OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

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

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

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

Read Only
Reset Value: 1111 1111 (FFh)

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

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

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

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

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

7							0
MSB							LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

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

7							0
MSB							LSB

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

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

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0032h	CR1	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
	Reset Value	0	0	0	0	0	0	0	0
0031h	CR2	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
	Reset Value	0	0	0	0	0	0	0	0
0033h	SR	ICF1	OCF1	TOF	ICF2	OCF2	-	-	-
	Reset Value	0	0	0	0	0	0	0	0
0034h-0035h	IC1HR	MSB	-	-	-	-	-	-	LSB
	Reset Value	-							-
	IC1LR	MSB	-	-	-	-	-	-	LSB
	Reset Value	-							-
0036h-0037h	OC1HR	MSB	-	-	-	-	-	-	LSB
	Reset Value	1	0	0	0	0	0	0	0
	OC1LR	MSB	-	-	-	-	-	-	LSB
	Reset Value	0	0	0	0	0	0	0	0
003Eh-003Fh	OC2HR	MSB	-	-	-	-	-	-	LSB
	Reset Value	1	0	0	0	0	0	0	0
	OC2LR	MSB	-	-	-	-	-	-	LSB
	Reset Value	0	0	0	0	0	0	0	0
0038h-0039h	CHR	MSB	1	1	1	1	1	1	LSB
	Reset Value	1							1
	CLR	MSB	1	1	1	1	1	0	LSB
	Reset Value	1							0
003Ah-003Bh	ACHR	MSB	1	1	1	1	1	1	LSB
	Reset Value	1							1
	ACLR	MSB	1	1	1	1	1	0	LSB
	Reset Value	1							0
003Ch-003Dh	IC2HR	MSB	-	-	-	-	-	-	LSB
	Reset Value	-							-
	IC2LR	MSB	-	-	-	-	-	-	LSB
	Reset Value	-							-

7.6 PWM AUTO-RELOAD TIMER (ART)

7.6.1 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

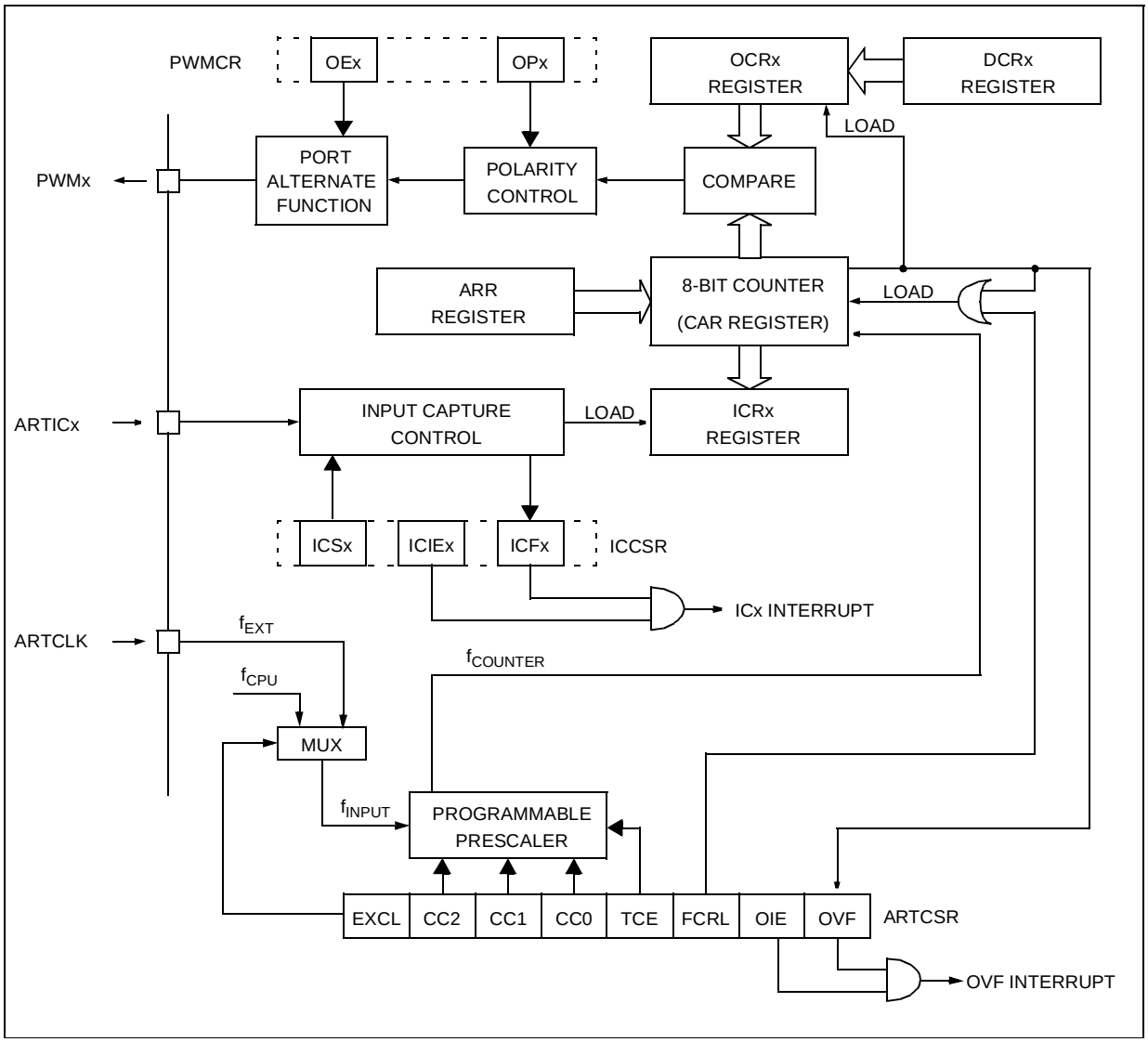
– External event detector

– 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 41. PWM Auto-Reload Timer Block Diagram



PWM AUTO-RELOAD TIMER (Cont'd)

7.6.2 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.

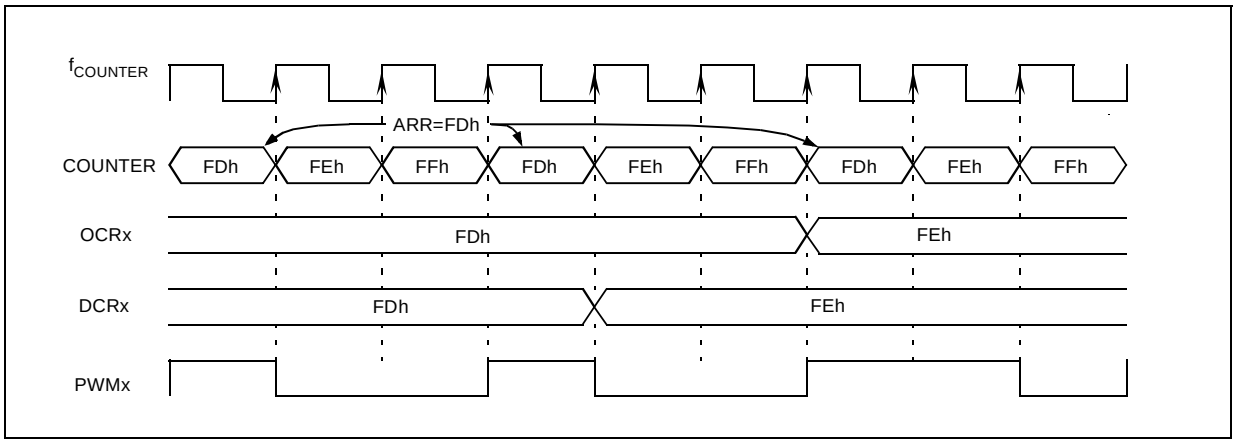
Direct access to the prescaler is not possible.

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 42. 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:

$$\text{Resolution} = 1 / (256 - ARR)$$

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 43. PWM Auto-reload Timer Function

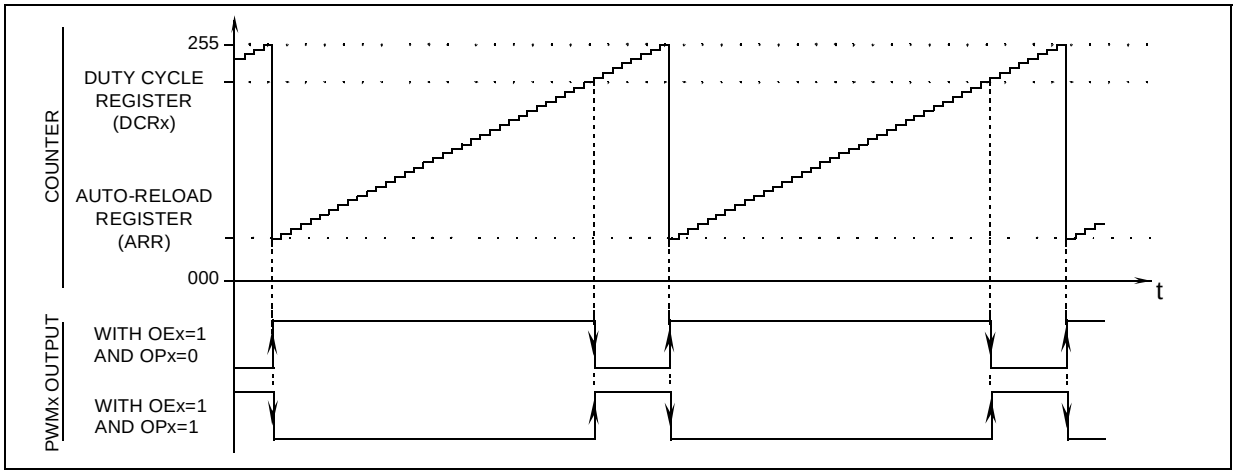
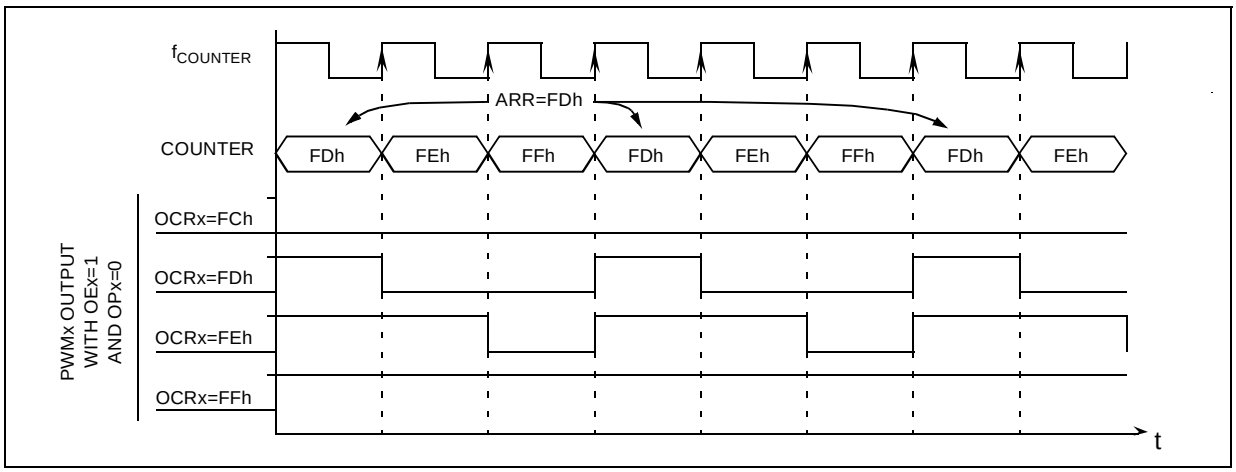


Figure 44. PWM Signal from 0% to 100% Duty Cycle



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.

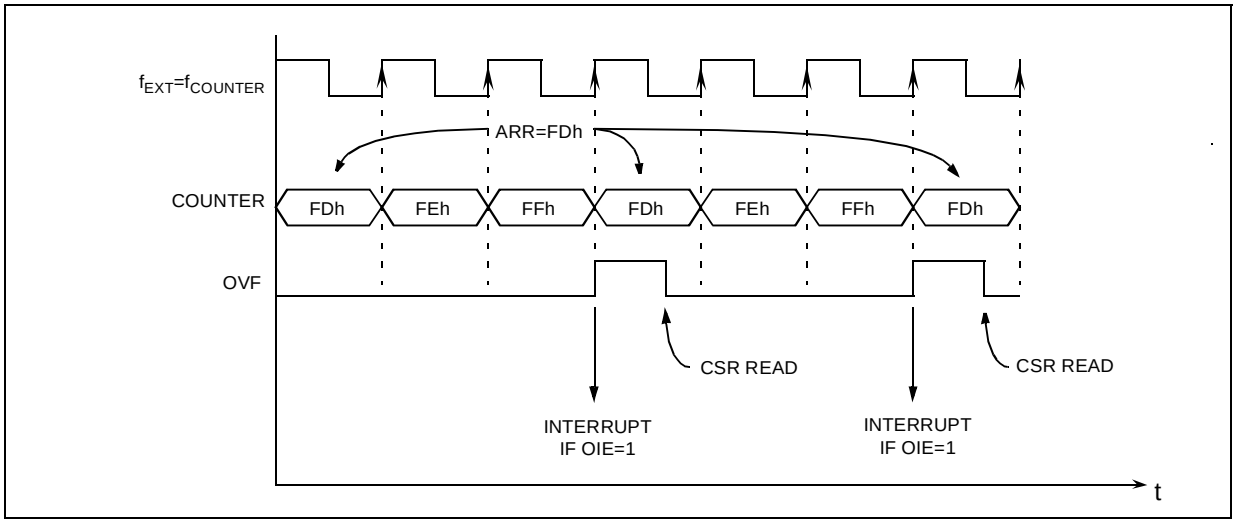
External clock and event detector mode

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 45. 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.

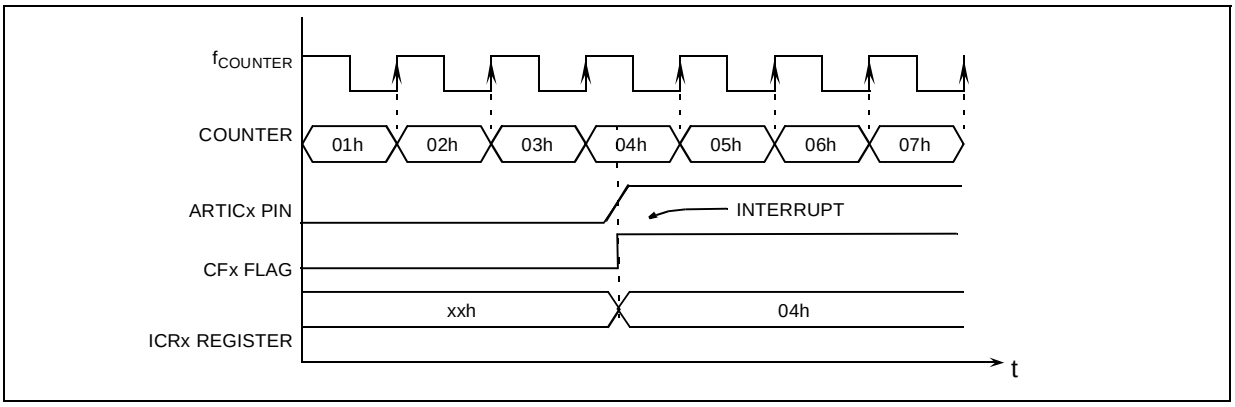
External interrupt capability

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 46. Input Capture Timing Diagram



PWM AUTO-RELOAD TIMER (Cont'd)

7.6.3 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
0	0	OE1	OE0	0	0	OP1	OP0

Bit 7:6 = Reserved.

Bit 5:4 = **OE[1:0]** *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 3:2 = Reserved.

Bit 1:0 = **OP[1:0]** *PWM Output Polarity*
These bits are set and cleared by software. They independently select the polarity of the two 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 allow to enable or not 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 16. PWM Auto-Reload Timer Register Map and Reset Values

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

7.7 SERIAL COMMUNICATIONS INTERFACE (SCI)

7.7.1 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.

7.7.2 Main Features

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

7.7.3 General Description

The interface is externally connected to another device by two pins (see Figure 47):

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

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

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

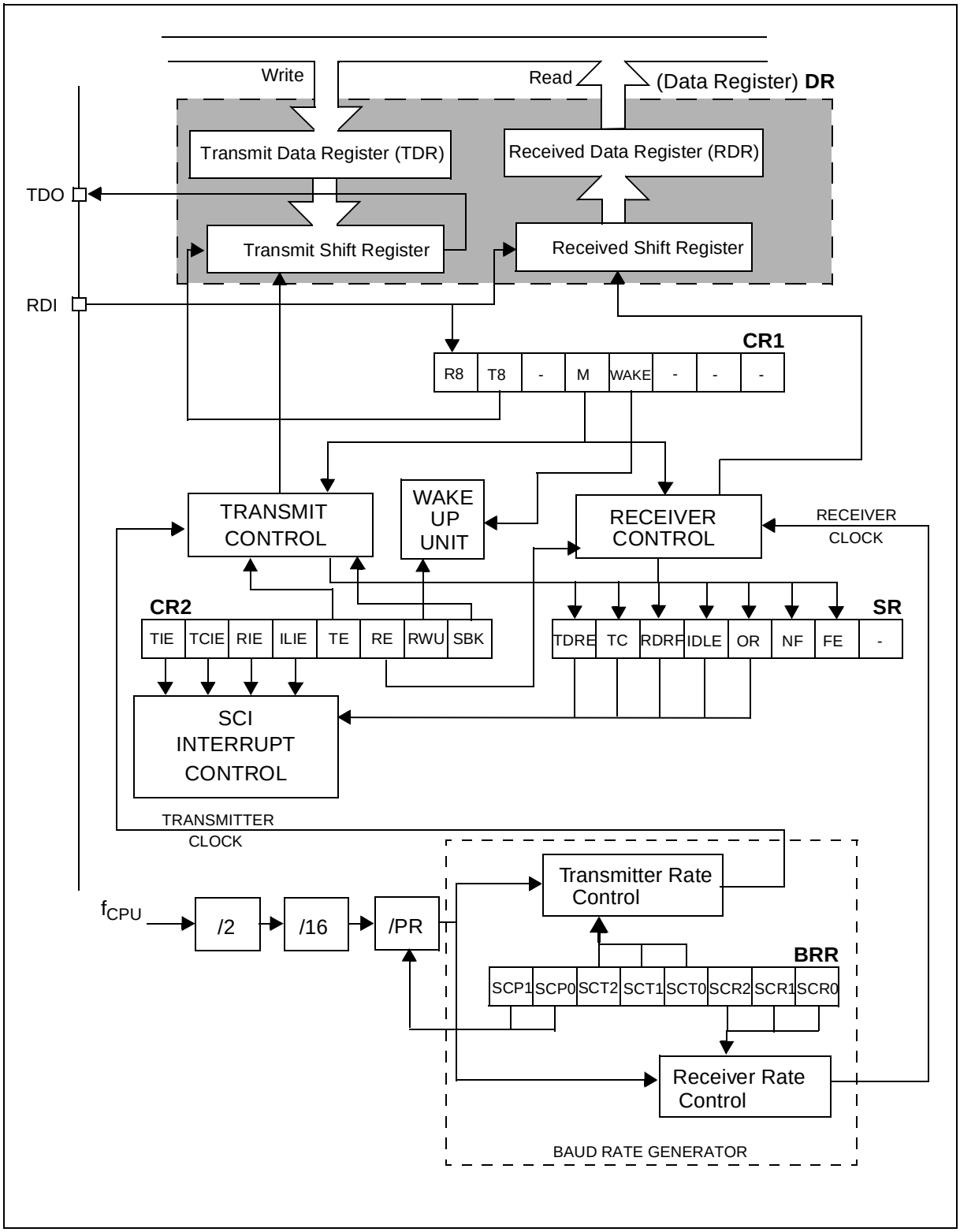
– An Idle Line prior to transmission or reception

– A start bit

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

– A Stop bit indicating that the frame is complete.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)
Figure 47. SCI Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

7.7.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 47. It contains 4 dedicated registers:

– Two control registers (CR1 & CR2)

– A status register (SR)

– A baud rate register (BRR)

Refer to the register descriptions in Section 9.7.7 for the definitions of each bit.

7.7.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the CR1 register (see Figure 47).

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

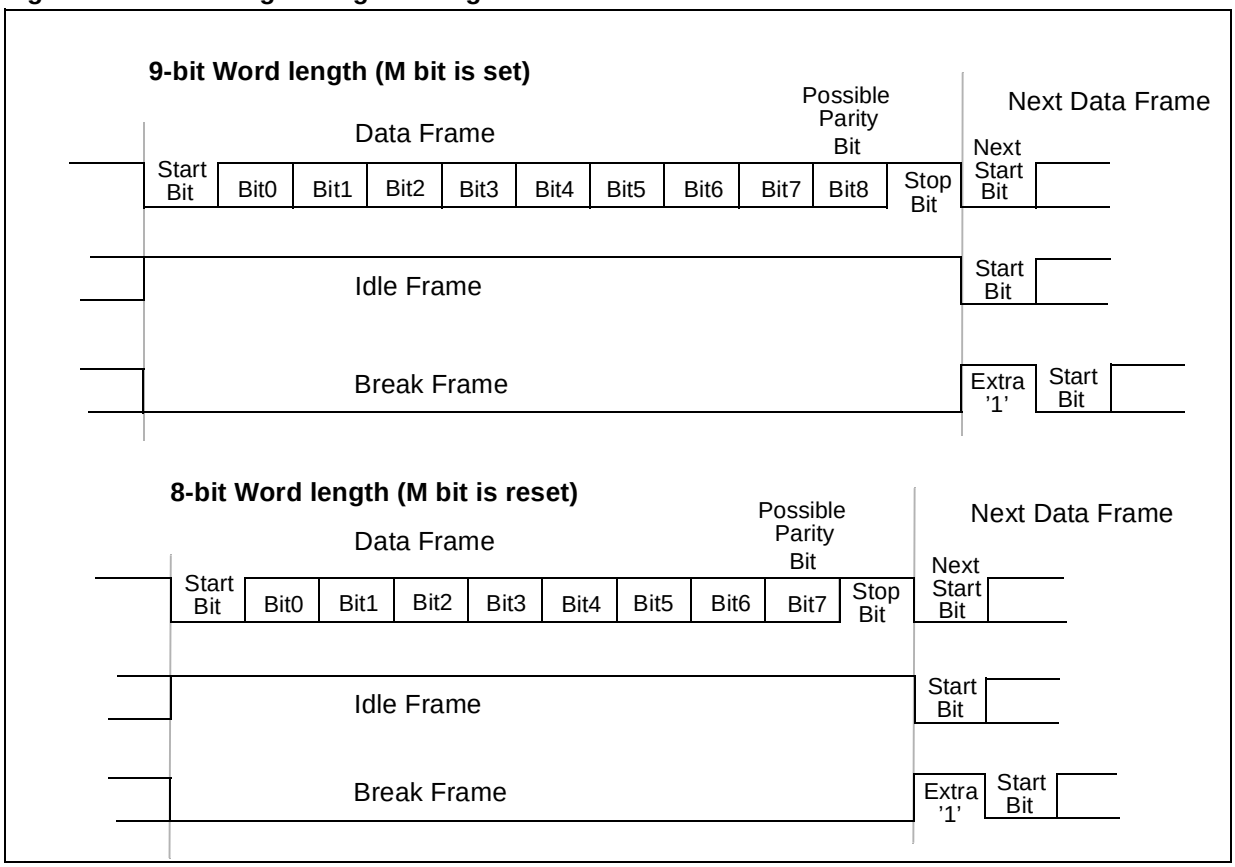
The TDO pin is in high state during the stop bit.

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 48. Word Length Programming



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

7.7.4.2 Transmitter

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

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the DR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 47).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR register.

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

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

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

The TDRE bit is set by hardware and it indicates that:

– The TDR register is empty.

– The data transfer is beginning.

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

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

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

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

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

The following software sequence is always to clear the TC bit:
1. An access to the SR register
2. A write to the DR register

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

Break Characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see Figure 48).

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

Idle Characters

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

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

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

7.7.4.3 Receiver

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

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the DR register consists of a buffer (RDR) between the internal bus and the received shift register (see Figure 47).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR register.

– Set the RE bit to enable the receiverto begin 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 CC register.

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

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

1. An access to the SR register

2. A read to the DR register.

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

Break Character

When a break character is received, the SCI 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 CC register.

Overrun Error

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

When a overrun error occurs:

– The OR bit is set.

– The RDR content will not be lost.

– The shift register will be overwritten.

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

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

Noise Error

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

When noise is detected in a frame:

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

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

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

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

Framing Error

A framing error is detected when:

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

– A break is received.

When the framing error is detected:

– The FE bit is set by hardware

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

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

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

7.7.4.4 Baud Rate Generation

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

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

with:

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

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

(see SCT0, SCT1 & SCT2 bits)

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

(see SCR0,SCR1 & SCR2 bits)

All these bits are in the BRR register.

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

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

7.7.4.5 Receiver Muting and Wake-up Feature

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

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

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

All the reception status bits can not be set.

All the receive interrupt are inhibited.

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

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

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

The 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.

The 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)

7.7.5 Low Power Modes

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

7.7.6 Interrupts

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

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

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

7.7.7 Register Description

STATUS REGISTER (SR)

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0

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 TIE =1 in the CR2 register. It is cleared by a software sequence (an access to the SR register followed by a write to the DR register).
0: Data is not transferred to the shift register
1: Data is transferred to the shift register

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

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

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

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

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

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

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

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

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

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

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

Bit 0 = Reserved, forced by hardware to 0.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: Undefined

7 0

R8	T8	0	M	WAKE	0	0	0

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 = Reserved, forced by hardware to 0.

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

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

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

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7 0

TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

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

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

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

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

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

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

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

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

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

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

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

DATA REGISTER (DR)

Read/Write

Reset Value: Undefined

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

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).
The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 47).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 47).

BAUD RATE REGISTER (BRR)

Read/Write

Reset Value: 00xx xxxx (XXh)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

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

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

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

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

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

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

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

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

Table 17. SCI Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0050h	SR	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	-	-	-	-	-	-	-	-
0051h	DR	SPIE	SPE	-	MSTR	CPOL	CPHA	SPR1	SPR0
	Reset Value	0	0	0	0	x	x	x	x
0052h	BRR	SPIF	WCOL	-	MODF	-	-	-	-
	Reset Value	0	0	0	0	0	0	0	0
0053h	CR1	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	-	-	-	-	-	-	-	-
0054h	CR2	SPIE	SPE	-	MSTR	CPOL	CPHA	SPR1	SPR0
	Reset Value	0	0	0	0	x	x	x	x

7.8 SERIAL PERIPHERAL INTERFACE (SPI)

7.8.1 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.

7.8.2 Main Features

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

7.8.3 General description

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

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

A basic example of interconnections between a single master and a single slave is illustrated on Figure 49.

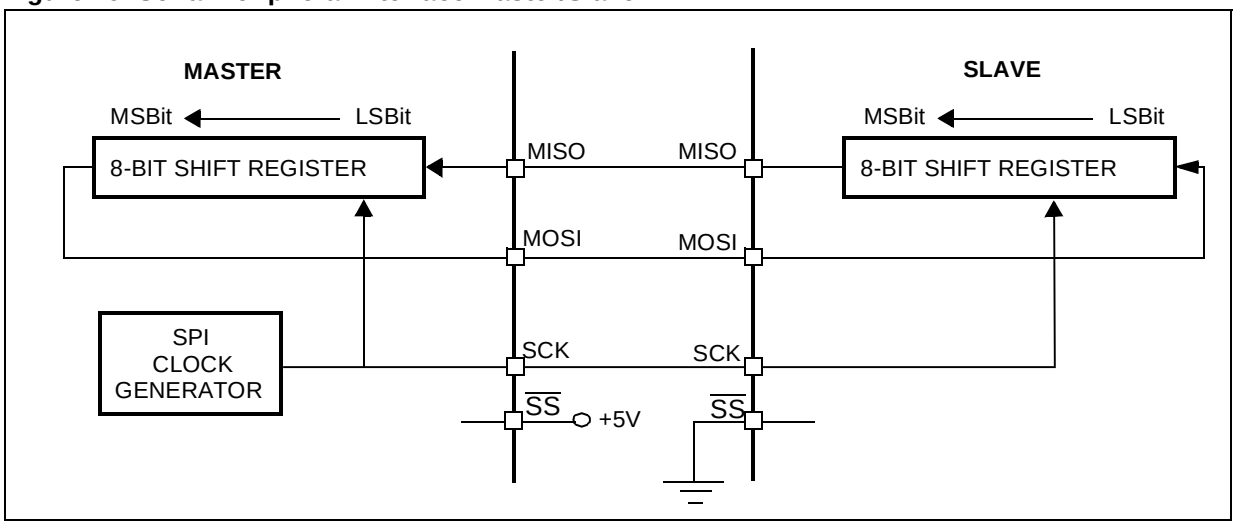
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

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.

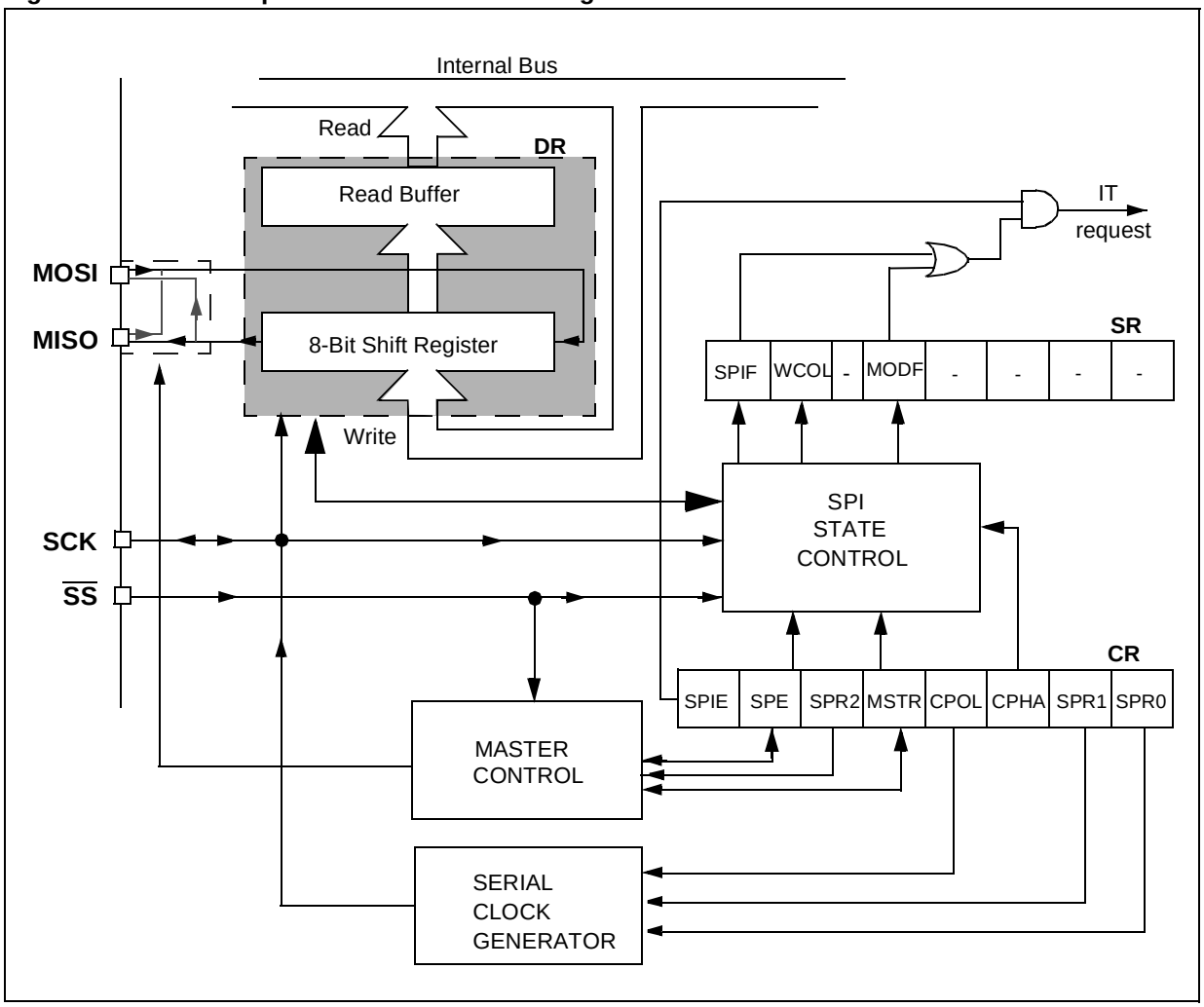
Four possible data/clock timing relationships may be chosen (see Figure 52) but master and slave must be programmed with the same timing mode.

Figure 49. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 50. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4 Functional Description

Figure 49 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

- A Control Register (CR)
- A Status Register (SR)
- A Data Register (DR)

Refer to the CR, SR and DR registers in Section 9.8.7for the bit definitions.

7.8.4.1 Master Configuration

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

Procedure

- Select the SPR0 & SPR1 bits to define the serial clock baud rate (see CR register).
- Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 52).
- The $\overline{SS}$ pin must be connected to a high level signal during the complete byte transmit sequence.
- The MSTR and SPE bits must be set (they remain set only if the $\overline{SS}$ pin is connected to a high level signal).

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

Transmit sequence

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

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

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt is generated if the SPIE bit is set and the I bit in the CCR register is cleared.

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

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

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

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

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4.2 Slave Configuration

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

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

Procedure

- For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 52.
- The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence.
- Clear the MSTR bit and set the SPE bit to assign the pins to alternate function.

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

Transmit Sequence

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

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

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt is generated if SPIE bit is set and I bit in CCR register is cleared.

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

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

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

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

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an overrun condition (see Section 9.8.4.6).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the DR register between each data byte transfer to avoid a write collision (see Section 9.8.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4.3 Data Transfer Format

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

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

Clock Phase and Clock Polarity

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

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

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

Figure 52, 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 second clock transition.

No write collision should occur even if the $\overline{SS}$ pin stays low during a transfer of several bytes (see Figure 51).

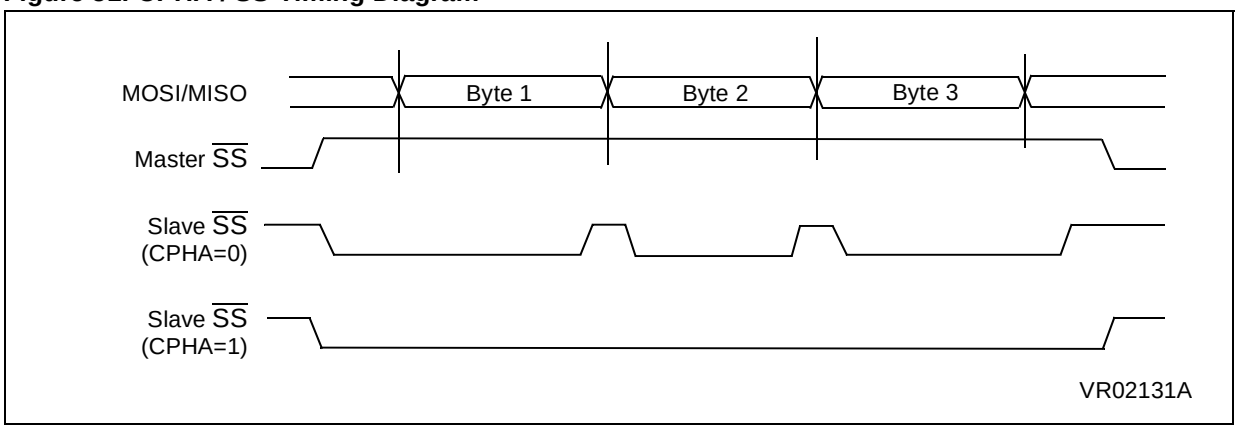
CPHA bit is reset

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 first clock transition.

The $\overline{SS}$ pin must be toggled high and low between each byte transmitted (see Figure 51).

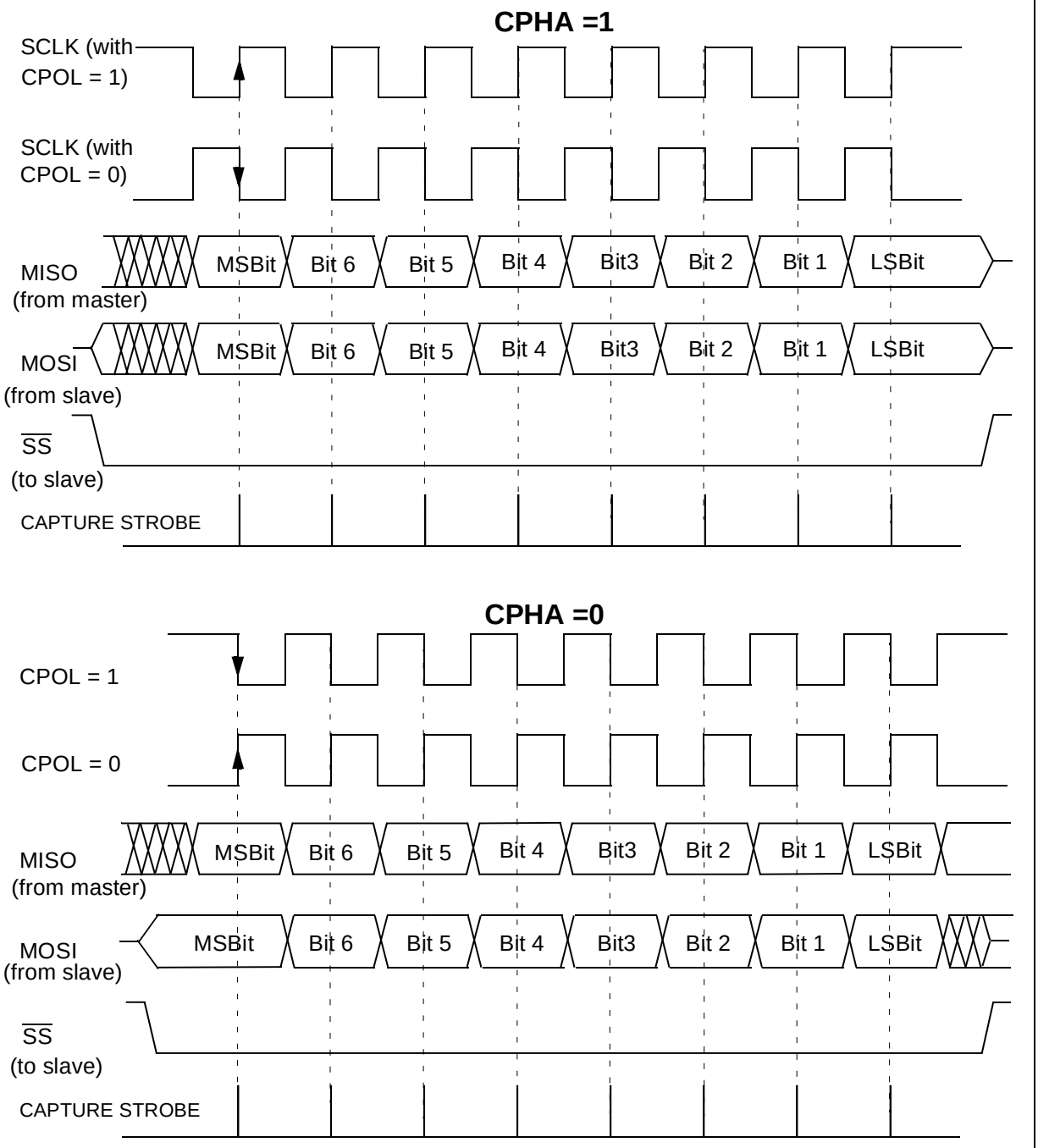
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its DR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the DR without producing a write collision.

Figure 51. CPHA / $\overline{SS}$ Timing Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 52. Data Clock Timing Diagram



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

VR02131B

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4.4 Write Collision Error

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

Write collisions can occur both in master and slave mode.

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

In Slave mode

When the CPHA bit is set:

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

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

When the CPHA bit is reset:

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

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

In Master mode

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

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

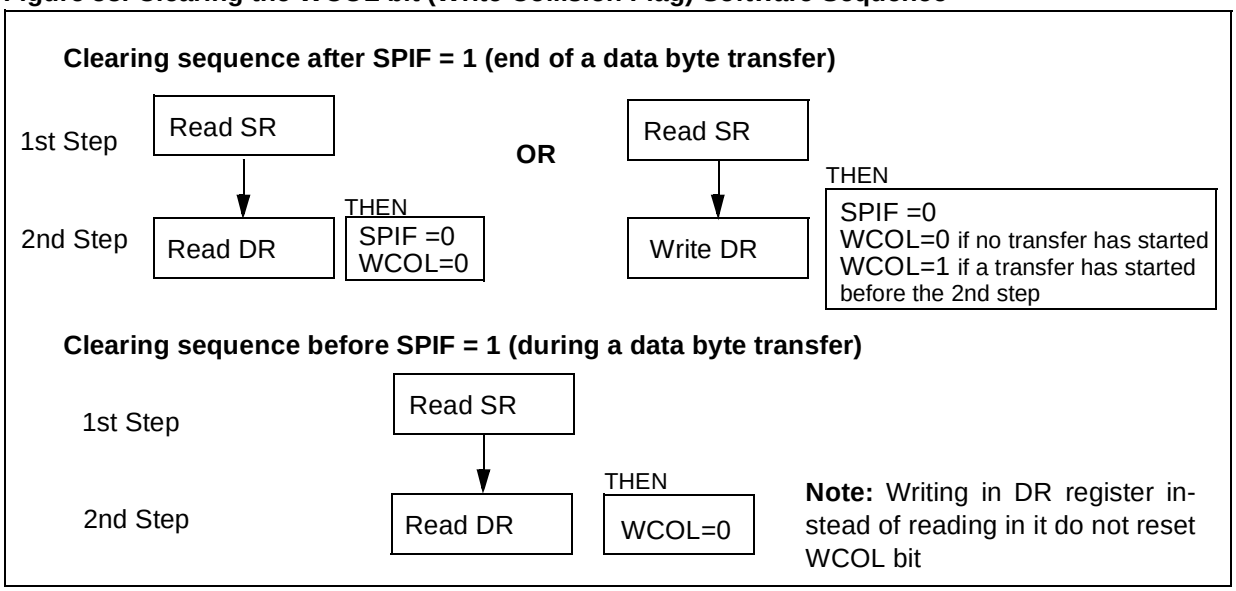
WCOL bit

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

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

Clearing the WCOL bit is done through a software sequence (see Figure 53).

Figure 53. Clearing the WCOL bit (Write Collision Flag) Software Sequence



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4.5 Master Mode Fault

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

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

– The MODF bit is set and an SPI interrupt is generated if the SPIE bit is set.

– The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.

– The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

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

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

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

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

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

7.8.4.6 Overrun Condition

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

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

This condition is not detected by the SPI peripheral.

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.4.7 Single Master and Multimaster Configurations

There are two types of SPI systems:

– Single Master System

– Multimaster System

Single Master System

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see Figure 54).

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.

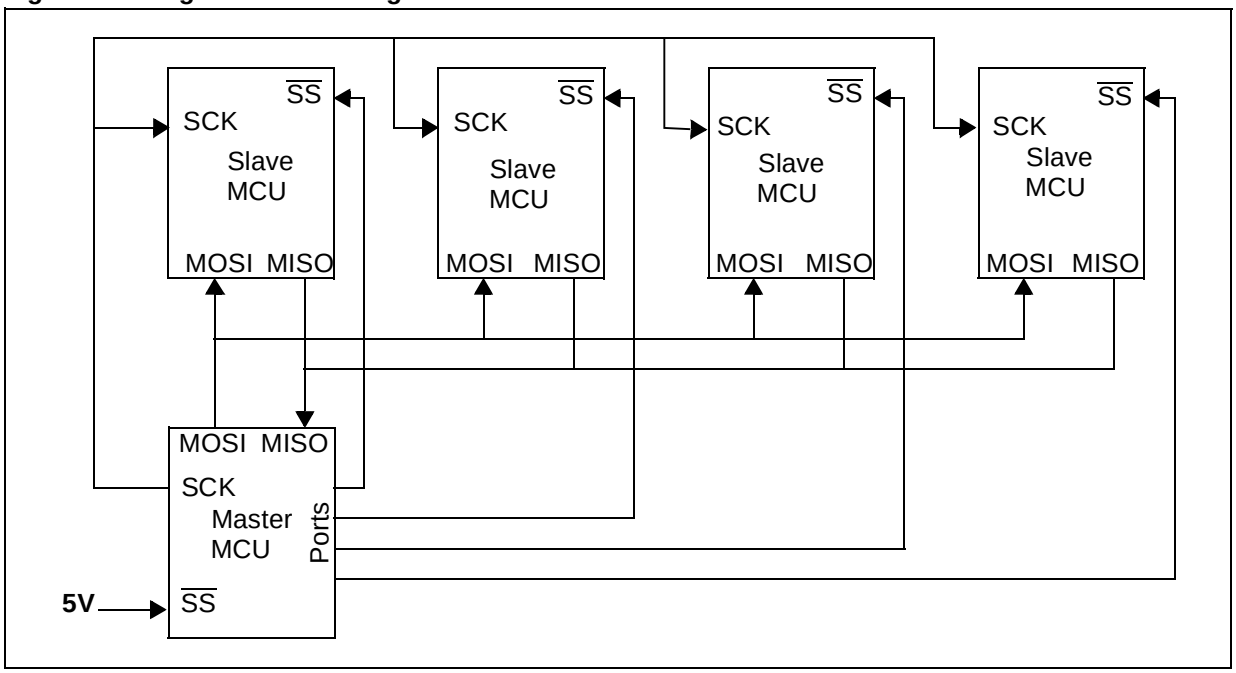
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

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 CR register and the MODF bit in the SR register.

Figure 54. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.5 Low Power Modes

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

7.8.6 Interrupts

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

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

SERIAL PERIPHERAL INTERFACE (Cont'd)

7.8.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0000xxxx (0xh)

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

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

Bit 6 = **SPE** *Serial peripheral output enable.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 9.8.4.5 Master Mode Fault).
0: I/O port connected to pins
1: SPI alternate functions connected to pins

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

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

this bit is set and cleared by software and it is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 19.
0: Divider by 2 enabled
1: Divider by 2 disabled

Bit 4 = **MSTR** *Master.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 9.8.4.5 Master Mode Fault).
0: Slave mode is selected
1: Master mode is selected, the function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

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

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

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

These 2 bits have no effect in slave mode.

Table 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)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	-	MODF	-	-	-	-

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

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

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

This bit is set by hardware when a write to the DR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 53).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = Unused.

Bit 4 = **MODF** *Mode Fault flag.*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 9.8.4.5 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the CR register. This bit is cleared by a software sequence (An access to the SR register while MODF=1 followed by a write to the CR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3-0 = Unused.

DATA I/O REGISTER (DR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

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

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

Warning:

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

A write to the the DR register returns the value located in the buffer and not the contents of the shift register (See Figure 50).

Table 19. SPI Register Map and Reset Values

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

7.9 8-BIT A/D CONVERTER (ADC)

7.9.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-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 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

7.9.2 Main Features

- 8-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 55.

7.9.3 Functional Description

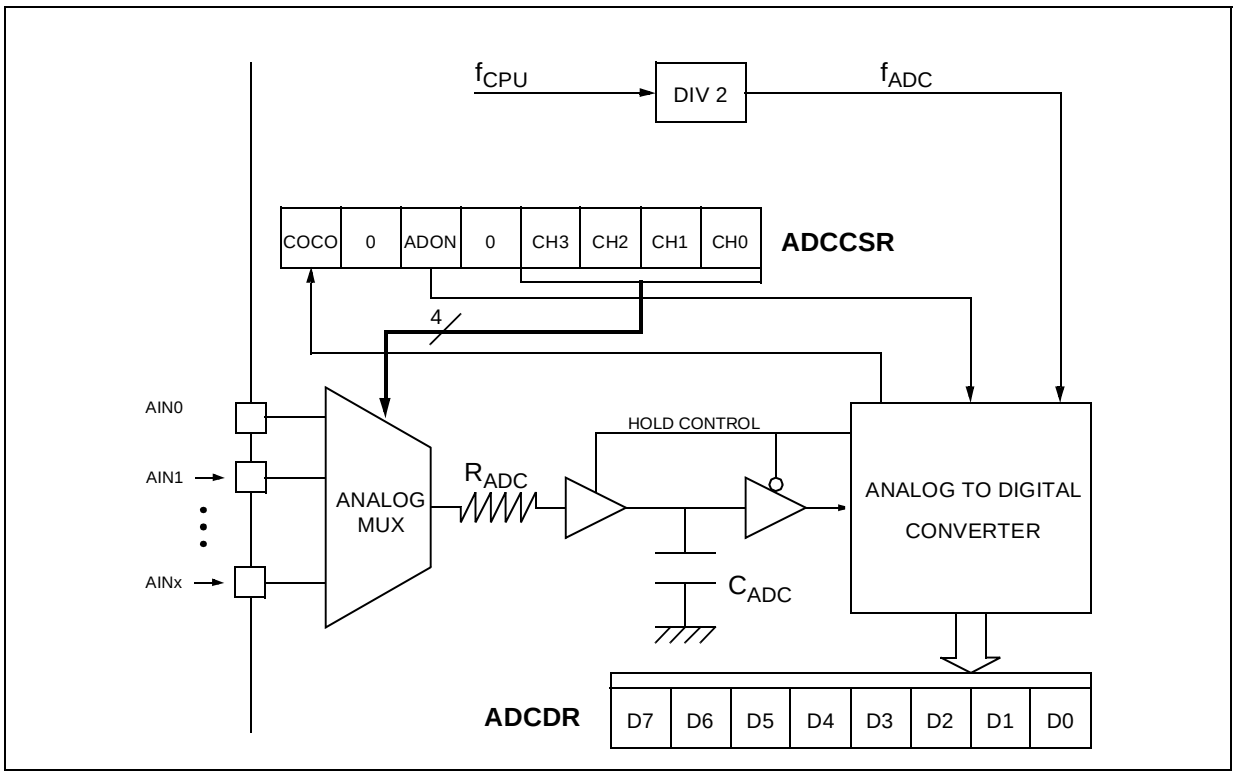
7.9.3.1 Analog Power Supply

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.

See electrical characteristics section for more details.

Figure 55. ADC Block Diagram



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

7.9.3.2 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 or equal to V_{DDA} (high-level voltage reference) then the conversion result in the DR register is FFh (full scale) without overflow indication.

If input voltage (V_{AIN}) is lower than or equal to V_{SSA} (low-level voltage reference) then the conversion result in the DR register is 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDR register. The accuracy of the conversion is described in the parametric 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.

7.9.3.3 A/D Conversion Phases

The A/D conversion is based on two conversion phases as shown in Figure 56:

- Sample capacitor loading [duration: t_{LOAD}]
  During this phase, the V_{AIN} input voltage to be measured is loaded into the C_{ADC} sample capacitor.
- A/D conversion [duration: t_{CONV}]
  During this phase, the A/D conversion is computed (8 successive approximations cycles) and the C_{ADC} sample capacitor is disconnected from the analog input pin to get the optimum analog to digital conversion accuracy.

While the ADC is on, these two phases are continuously repeated.

At the end of each conversion, the sample capacitor is kept loaded with the previous measurement load. The advantage of this behaviour is that it minimizes the current consumption on the analog pin in case of single input channel measurement.

7.9.3.4 Software Procedure

Refer to the control/status register (CSR) and data register (DR) in Section 9.9.6 for the bit definitions and to Figure 56 for the timings.

ADC Configuration

The total duration of the A/D conversion is 12 ADC clock periods ($1/f_{ADC}=2/f_{CPU}$).

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 CSR register:

– Select the CH[3:0] bits to assign the analog channel to be converted.

ADC Conversion

In the CSR register:

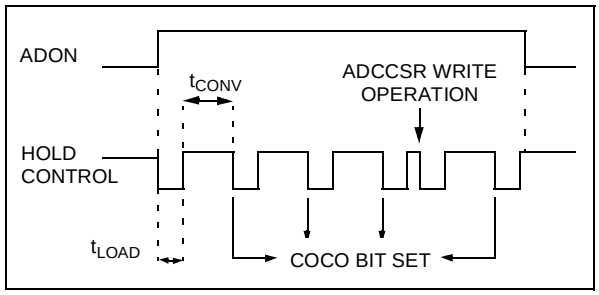
– Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete

– The COCO bit is set by hardware.
– No interrupt is generated.
– The result is in the DR register and remains valid until the next conversion has ended.

A write to the CSR register (with ADON set) aborts the current conversion, resets the COCO bit and starts a new conversion.

Figure 56. ADC Conversion Timings



7.9.4 Low Power Modes

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 before accurate conversions can be performed.

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.

7.9.5 Interrupts

None

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

7.9.6 Register Description

CONTROL/STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
COCO	0	ADON	0	CH3	CH2	CH1	CH0

Bit 7 = **COCO** *Conversion Complete*
This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.
0: Conversion is not complete
1: Conversion can be read from the DR register

Bit 6 = **Reserved.** *must always be cleared.*

Bit 5 = **ADON** *A/D Converter On*
This bit is set and cleared by software.
0: A/D converter is switched off
1: A/D converter is switched on

Bit 4 = **Reserved.** *must always be 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

***Note**: The number of pins AND the channel selection varies according to the device. Refer to the device pinout.

DATA REGISTER (DR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *Analog Converted Value*
This register contains the converted analog value in the range 00h to FFh.

Note: Reading this register reset the COCO flag.

Table 20. ADC Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0070h	DR	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
	Reset Value	0	0	0	0	0	0	0	0
0071h	CSR	COCO	EXTCK	ADON	0	CH3	CH2	CH1	CH0
	Reset Value	0	0	0	0	0	0	0	0

8 INSTRUCTION SET

8.1 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 21. ST7 Addressing Mode Overview

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

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

ST7 ADDRESSING MODES (Cont'd)

8.1.1 Inherent

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

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

8.1.2 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

8.1.3 Direct

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

The direct addressing mode consists of two submodes:

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.

8.1.4 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.

8.1.5 Indirect (Short, Long)

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

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

Indirect (short)

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

Indirect (long)

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

ST7 ADDRESSING MODES (Cont'd)

8.1.6 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 22. Instructions Supporting Direct, Indexed, Indirect and Indirect Indexed Addressing Modes

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

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

8.1.7 Relative Mode (Direct, Indirect)

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

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

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

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

8.2 INSTRUCTION GROUPS

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

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

Using a pre-byte

The instructions are described with one to four bytes.

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

The whole instruction becomes:

- PC-2 End of previous instruction
- PC-1 Prebyte
- PC Opcode
- PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

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

- PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.
- PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
  It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.
- PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

INSTRUCTION GROUPS (Cont'd)

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

INSTRUCTION GROUPS (Cont'd)

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

9 ELECTRICAL CHARACTERISTICS

9.1 PARAMETER CONDITIONS

Unless otherwise specified, all voltages are referred to V_{SS}.

9.1.1 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).

9.1.2 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.

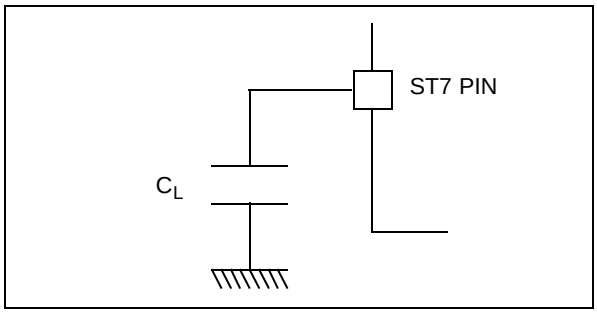
9.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

9.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in Figure 57.

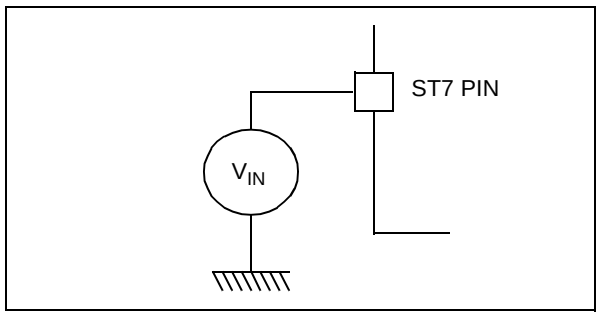
Figure 57. Pin loading conditions



9.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 58.

Figure 58. Pin input voltage



9.2 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.

9.2.1 Voltage Characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{IN}	Input voltage on any pin [1) & 2)]	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	see Section 9.7.2 Absolute Electrical Sensitivity	
$V_{ESD(MM)}$	Electro-static discharge voltage (Machine Model)		

9.2.2 Current Characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) [3)]	80	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) [3)]	80	
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 ISPSEL 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)]	± 20	

9.2.3 Thermal Characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see Section 10.2 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). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.

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.

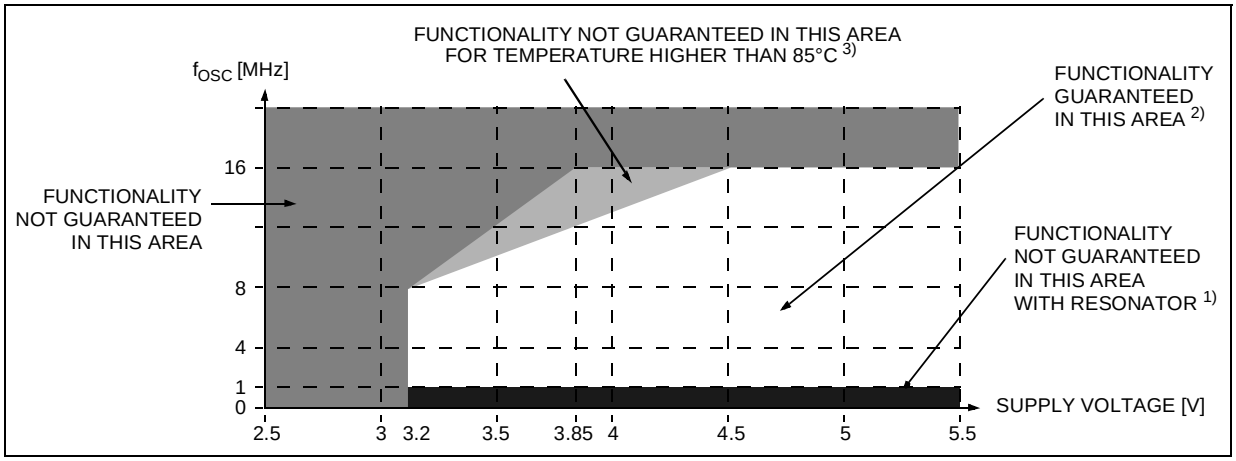
6. True open drain I/O port pins do not accept positive injection.

9.3 OPERATING CONDITIONS

9.3.1 General Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	Supply voltage	see Figure 59 and Figure 60	3.2	5.5	V
f_{OSC}	External clock frequency	V_{DD}≥4.5V	0 [1]	16	MHz
		V_{DD}≥3.0V	0 [1]	8	
T_A	Ambient temperature range		-40	85	°C

Figure 59. f_{OSC} Maximum Operating Frequency Versus V_{DD} Supply Voltage for FLASH devices



Notes:

1. Guaranteed by construction. A/D operation and resonator oscillator start-up are not guaranteed below 1MHz.

3. FLASH programming tested in production at maximum T_A with two different conditions: V_{DD}=5.5V, f_{CPU}=8MHz and V_{DD}=3V, f_{CPU}=4MHz.

OPERATING CONDITIONS (Cont'd)

9.3.2 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[4]	3.85[2] 3.50[2] 3.00	4.05 3.65 3.10	4.25 3.80 3.20	
V_{hyst}	LVD voltage threshold hysteresis	V_{IT+}-V_{IT-}	200	250	300	mV
Vt_{POR}	V_{DD} rise time rate [3]		0.2		50	V/ms
$t_{g(VDD)}$	Filtered glitch delay on V_{DD} [2]	Not detected by the LVD			40	ns

Figure 60. High LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [3]

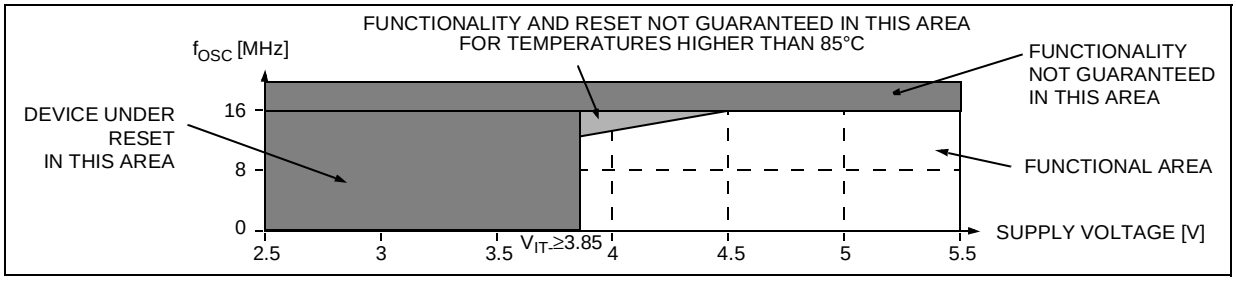


Figure 61. Medium LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [3]

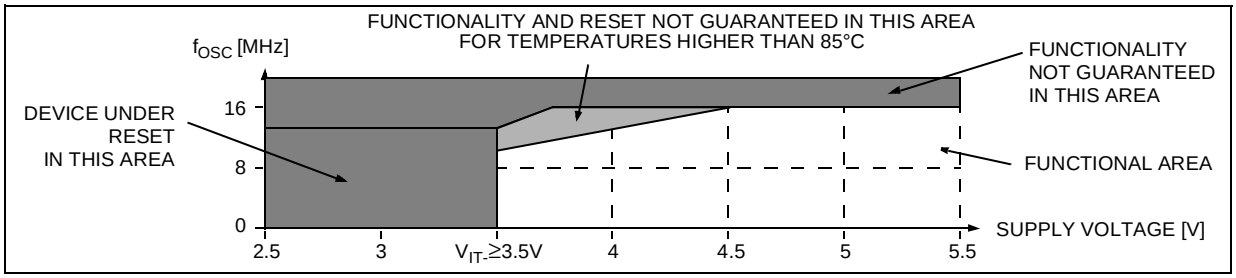
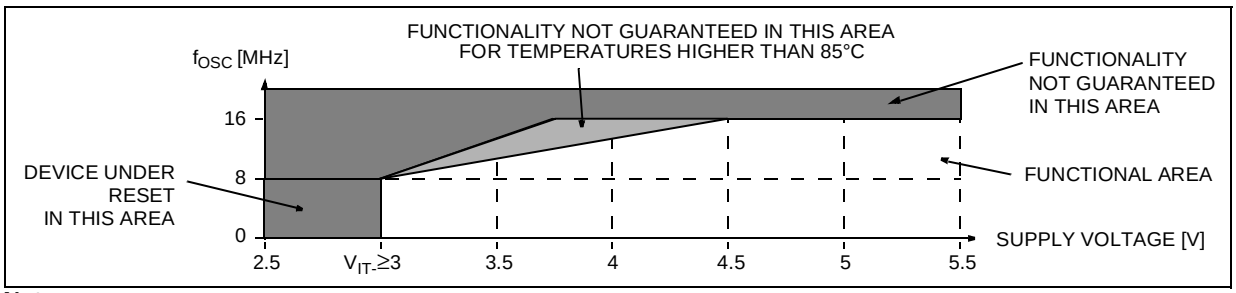


Figure 62. Low LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [2][4]



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.
4. If the low LVD threshold is selected, when V_{DD} falls below 3.2V, (V_{DD} minimum operating voltage), the device is guaranteed to continue functioning until it goes into reset state. The specified V_{DD} min. value is necessary in the device power on phase, but during a power down phase or voltage drop the device will function below this min. level.

9.4 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	%

9.4.1 RUN and SLOW Modes

Symbol	Parameter	Conditions		Typ [1]	Max [2]	Unit
I_{DD}	Supply current in RUN mode [3] (see Figure 63)	$4.5V \le V_{DD} \le 5.5V$	f_{OSC}=1MHz, f_{CPU}=500kHz	500	900	µA
			f_{OSC}=4MHz, f_{CPU}=2MHz	1500	2500	
			f_{OSC}=16MHz, f_{CPU}=8MHz	5600	9000	
	Supply current in SLOW mode [4] (see Figure 64)		f_{OSC}=1MHz, f_{CPU}=31.25kHz	150	450	
			f_{OSC}=4MHz, f_{CPU}=125kHz	250	550	
			f_{OSC}=16MHz, f_{CPU}=500kHz	670	1250	
	Supply current in RUN mode [3] (see Figure 63)	$3.2V \le V_{DD} \le 3.6V$	f_{OSC}=1MHz, f_{CPU}=500kHz	300	550	
			f_{OSC}=4MHz, f_{CPU}=2MHz	970	1350	
			f_{OSC}=16MHz, f_{CPU}=8MHz	3600	4500	
	Supply current in SLOW mode [4] (see Figure 64)		f_{OSC}=1MHz, f_{CPU}=31.25kHz	100	250	
			f_{OSC}=4MHz, f_{CPU}=125kHz	170	300	
			f_{OSC}=16MHz, f_{CPU}=500kHz	420	700	

Figure 63. Typical IDD in RUN vs. fCPU

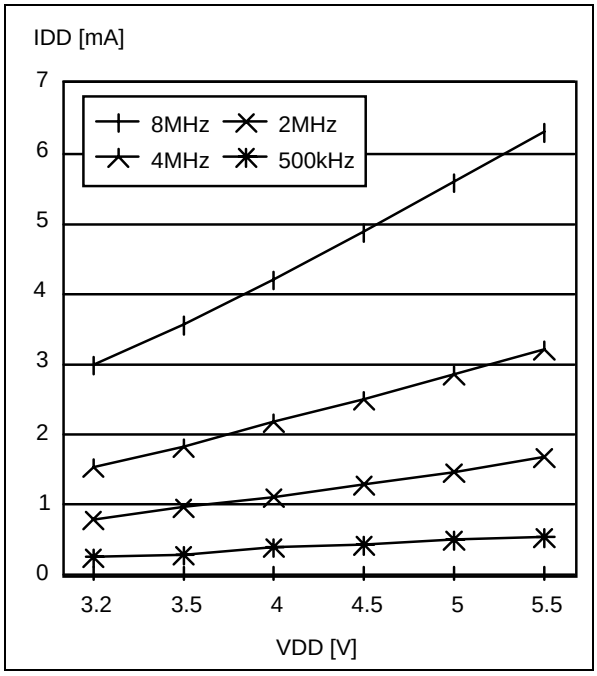
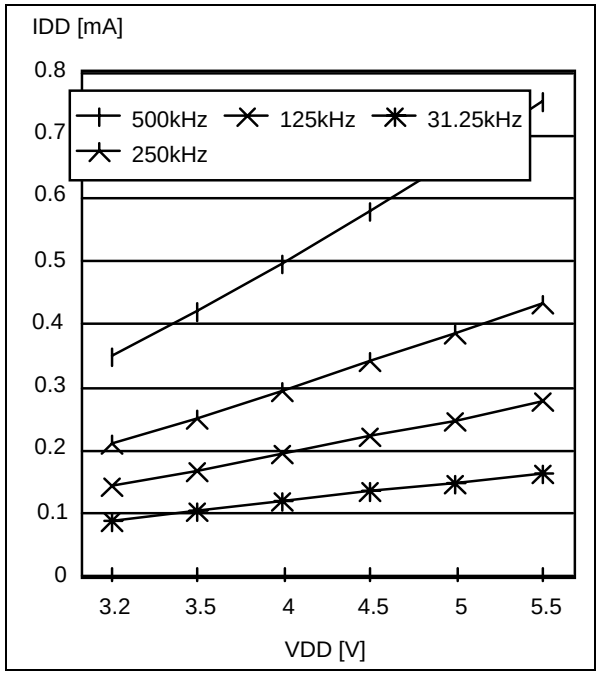


Figure 64. Typical I_{DD} in SLOW vs. f_{CPU}



Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \le V_{DD} \le 5.5V$ range) and V_{DD}=3.4V ($3.2V \le V_{DD} \le 3.6V$ 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 output 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)

9.4.2 WAIT and SLOW WAIT Modes

Symbol	Parameter	Conditions		Typ [1]	Max [2]	Unit
I_{DD}	Supply current in WAIT mode [3] (see Figure 65)	$4.5V \le V_{DD} \le 5.5V$	f_{OSC}=1MHz, f_{CPU}=500kHz	150	280	µA
			f_{OSC}=4MHz, f_{CPU}=2MHz	560	900	
			f_{OSC}=16MHz, f_{CPU}=8MHz	2200	3000	
	Supply current in SLOW WAIT mode [4] (see Figure 66)		f_{OSC}=1MHz, f_{CPU}=31.25kHz	20	70	
			f_{OSC}=4MHz, f_{CPU}=125kHz	90	190	
			f_{OSC}=16MHz, f_{CPU}=500kHz	340	850	
	Supply current in WAIT mode [3] (see Figure 65)	$3.2V \le V_{DD} \le 3.6V$	f_{OSC}=1MHz, f_{CPU}=500kHz	90	200	
			f_{OSC}=4MHz, f_{CPU}=2MHz	350	550	
			f_{OSC}=16MHz, f_{CPU}=8MHz	1370	1900	
	Supply current in SLOW WAIT mode [4] (see Figure 66)		f_{OSC}=1MHz, f_{CPU}=31.25kHz	10	20	
			f_{OSC}=4MHz, f_{CPU}=125kHz	50	80	
			f_{OSC}=16MHz, f_{CPU}=500kHz	200	350	

Figure 65. Typical I_{DD} in WAIT vs. f_{CPU}

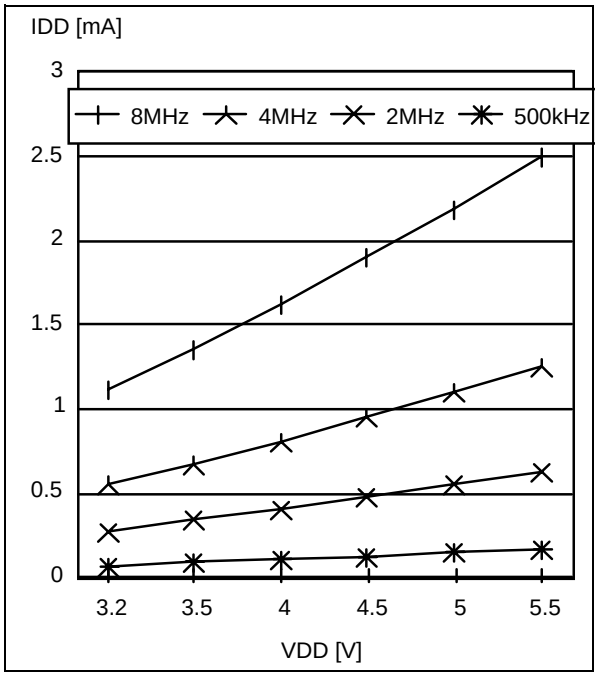
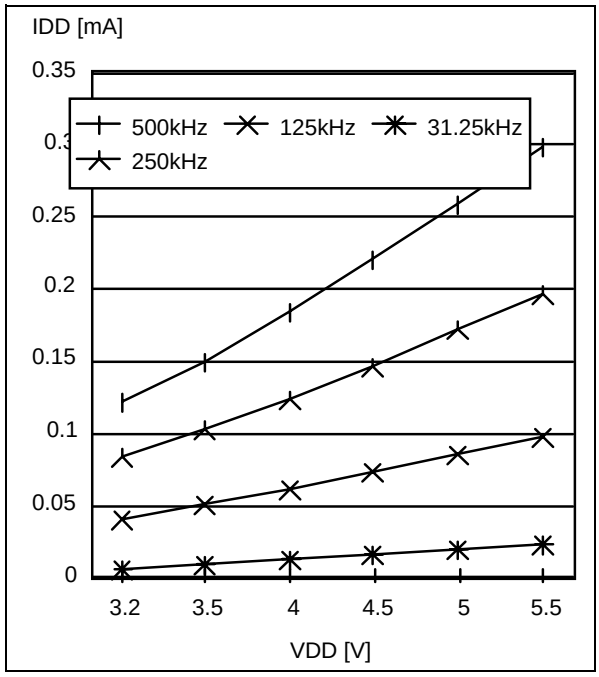


Figure 66. Typical I_{DD} in SLOW-WAIT vs. f_{CPU}



Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \le V_{DD} \le 5.5V$ range) and V_{DD}=3.4V ($3.2V \le V_{DD} \le 3.6V$ range).
2. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
3. All I/O pins in output 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)

9.4.3 HALT Mode

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
		V_{DD}=3.6V	-40°C≤T_A≤+85°C		6	

9.4.4 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 [3]	Unit
$I_{DD(CK)}$	Supply current of internal RC oscillator		500	750	µA
	Supply current of external RC oscillator [4]		525	750	
	Supply current of resonator oscillator [4] & [5]	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator	200 300 450 700	400 550 750 1000	
	Clock security system supply current		150	350	
$I_{DD(LVD)}$	LVD supply current	HALT mode	100	150	

9.4.5 On-Chip Peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(TIM)}$	16-bit Timer supply current [6]	f_{CPU}=8MHz	V_{DD}=3.4V	50	µA
			V_{DD}=5.0V	150	
$I_{DD(SPI)}$	SPI supply current [7]	f_{CPU}=8MHz	V_{DD}=3.4V	250	
			V_{DD}=5.0V	350	
$I_{DD(I2C)}$	I^2C supply current [8]	f_{CPU}=8MHz	V_{DD}=3.4V	250	
			V_{DD}=5.0V	350	
$I_{DD(ADC)}$	ADC supply current when converting [9]	f_{ADC}=4MHz	V_{DD}=3.4V	800	
			V_{DD}=5.0V	1100	

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 characterization results, not tested in production.
4. Data based on characterization results done with the external components specified in Section 9.5.3 and Section 9.5.4, not tested in production.
5. As the oscillator is based on a current source, the consumption does not depend on the voltage.
6. 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.
7. Data based on a differential I_{DD} measurement between reset configuration and a permanent SPI master communication (data sent equal to 55h).
8. Data based on a differential I_{DD} measurement between reset configuration and I2C peripheral enabled (PE bit set).
9. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

9.5 CLOCK AND TIMING CHARACTERISTICS

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

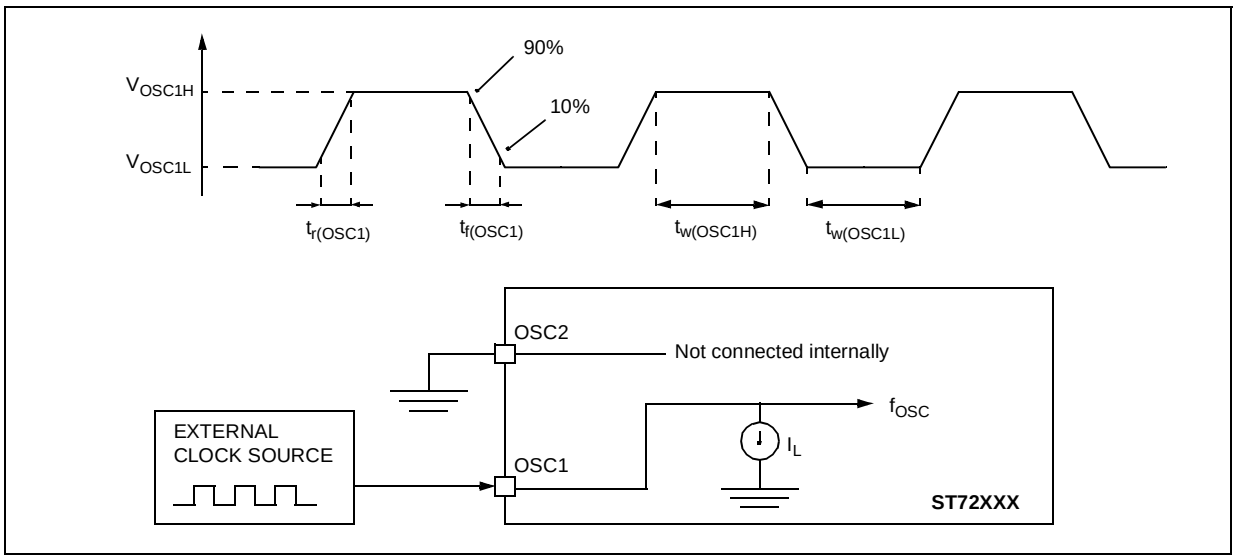
9.5.1 General Timings

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

9.5.2 External Clock Source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage	see Figure 67	$0.7xV_{DD}$		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		$0.3xV_{DD}$	
$t_{w(OSC1H)}$ $t_{w(OSC1L)}$	OSC1 high or low time [3]		15			ns
$t_{r(OSC1)}$ $t_{f(OSC1)}$	OSC1 rise or fall time [3]				15	
I_L	OSCx Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA

Figure 67. 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)

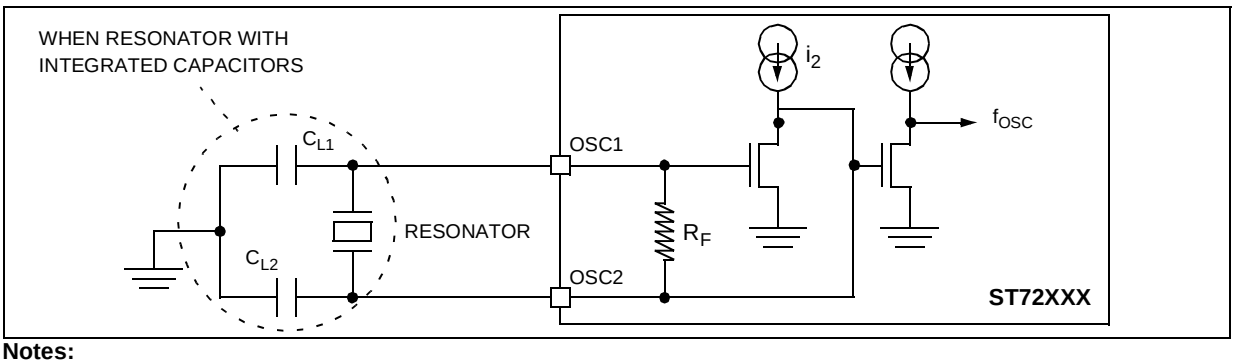
9.5.3 Crystal and Ceramic Resonator Oscillators

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 componants. 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 [3]	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}	Recommanded load capacitances versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	R_S=200Ω R_S=200Ω R_S=200Ω R_S=100Ω	LP oscillator MP oscillator MS oscillator HS oscillator	38 32 18 15	56 46 26 21	pF
i_2	OSC2 driving current	V_{DD}=5V V_{IN}=V_{SS}	LP oscillator MP oscillator MS oscillator HS oscillator	40 50 100 250	130 300 550 820	µA

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	Δf_{OSC}=[±30ppm$_{25°C}$,±30ppm$_{\Delta Ta}$], Typ. R_S=200Ω	33	34	10~15
	MP		SS3-400-30-30/30	4MHz	Δf_{OSC}=[±30ppm$_{25°C}$,±30ppm$_{\Delta Ta}$], Typ. R_S=60Ω	33	34	7~10
	MS		SS3-800-30-30/30	8MHz	Δf_{OSC}=[±30ppm$_{25°C}$,±30ppm$_{\Delta Ta}$], Typ. R_S=25Ω	33	34	2.5~3
	HS		SS3-1600-30-30/30	16MHz	Δf_{OSC}=[±30ppm$_{25°C}$,±30ppm$_{\Delta Ta}$], Typ. R_S=15Ω	33	34	1~1.5
Ceramic	LP	MURATA	CSA2.00MG	2MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.3%$_{\Delta Ta}$,±0.3%$_{aging}$,±x.x%$_{correl}$]	33	30	4.2
	MP		CSA4.00MG	4MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.3%$_{\Delta Ta}$,±0.3%$_{aging}$,±x.x%$_{correl}$]	33	30	2.1
	MS		CSA8.00MTZ	8MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.5%$_{\Delta Ta}$,±0.3%$_{aging}$,±x.x%$_{correl}$]	33	30	1.1
	HS		CSA16.00MXZ040	16MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.3%$_{\Delta Ta}$,±0.3%$_{aging}$,±x.x%$_{correl}$]	33	30	0.7

Figure 68. Typical Application with a Crystal or Ceramic Resonator



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).

3. 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 CHARACTERISTICS (Cont'd)

9.5.4 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]	see Figure 69	3.60		5.10	MHz
	External RC oscillator frequency [2]		1		14	
$t_{SU(OSC)}$	Internal RC Oscillator Start-up Time [3]			2.0		ms
	External RC Oscillator Start-up Time [3]	R_{EX}=47KΩ, C_{EX}="0"pF R_{EX}=47KΩ, C_{EX}=100pF R_{EX}=10KΩ, C_{EX}=6.8pF R_{EX}=10KΩ, C_{EX}=470pF		1.0 6.5 0.7 3.0		
R_{EX}	Oscillator external resistor [4]	see Figure 70	10		47	KΩ
C_{EX}	Oscillator external capacitor		0 [5]		470	pF

Figure 69. Typical Application with RC oscillator

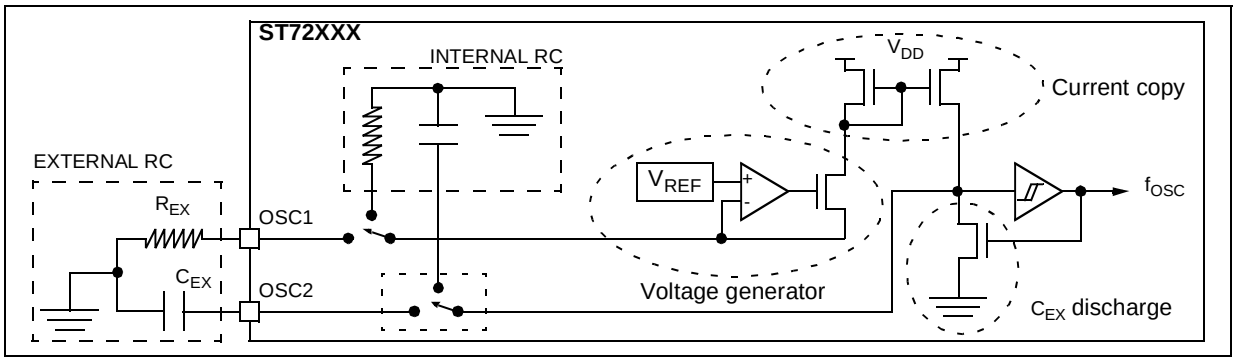


Figure 70. Typical Internal RC Oscillator

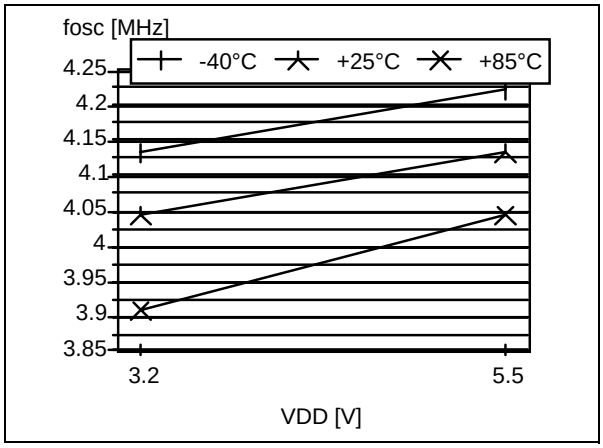
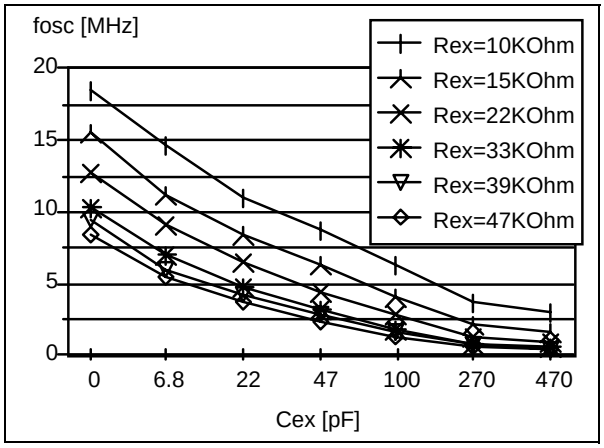


Figure 71. Typical External RC Oscillator



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.

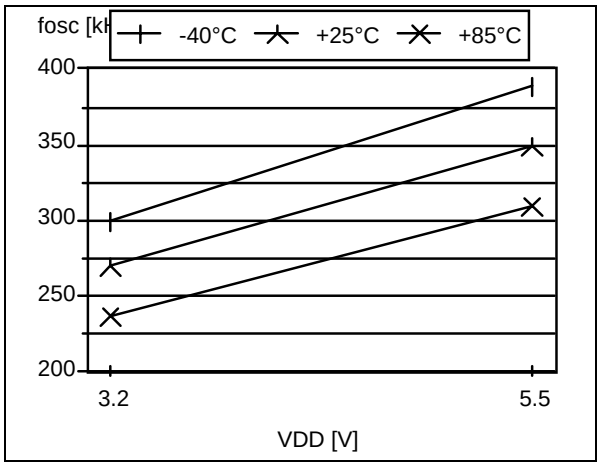
5. **Important:** when no external C_{EX} is applied, the capacitance to be considered is the global parasitic capacitance which is subject to high variation (package, application...). In this case, the RC oscillator frequency tuning has to be done by trying out several resistor values.

CLOCK CHARACTERISTICS (Cont'd)

9.5.5 Clock Security System (CSS)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SFOSC}	Safe Oscillator Frequency [1]	T_A=25°C, V_{DD}=5.0V	250	340	430	kHz
		T_A=25°C, V_{DD}=3.4V	190	260	330	
f_{GFOSC}	Glitch Filtered Frequency [2]			30		MHz

Figure 72. Typical Safe Oscillator Frequencies



Note:

1. Data based on characterization results, tested in production between 90KHz and 500KHz.
2. Filtered glitch on the f_{OSC} signal. See functional description in section 4.3 on page 21 for more details.

9.6 MEMORY CHARACTERISTICS

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

9.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	HALT mode (or RESET)	1.6			V

9.6.2 FLASH Program Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{A(prog)}$	Programming temperature range [2]		0	25	70	°C
t_{prog}	Programming time for 1~16 bytes [3]	T_A=+25°C		8	25	ms
	Programming time for 4 or 8kBytes	T_A=+25°C		2.1	6.4	sec
t_{ret}	Data retention [5]	T_A=+55°C [4]	20			years
N_{RW}	Write erase cycles [5]	T_A=+25°C	100			cycles

Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in in HALT mode or under RESET) or in hardware registers (only in HALT mode). Guaranteed by construction, not tested in production.
2. Data based on characterization results, tested in production at T_A=25°C.
3. Up to 16 bytes can be programmed at a time for a 4kBytes FLASH block (then up to 32 bytes at a time for an 8k device)
4. The data retention time increases when the T_A decreases.
5. Data based on reliability test results and monitored in production.

9.7 EMC CHARACTERISTICS

Susceptibility tests are performed on a sample basis during product characterization.

9.7.1 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).

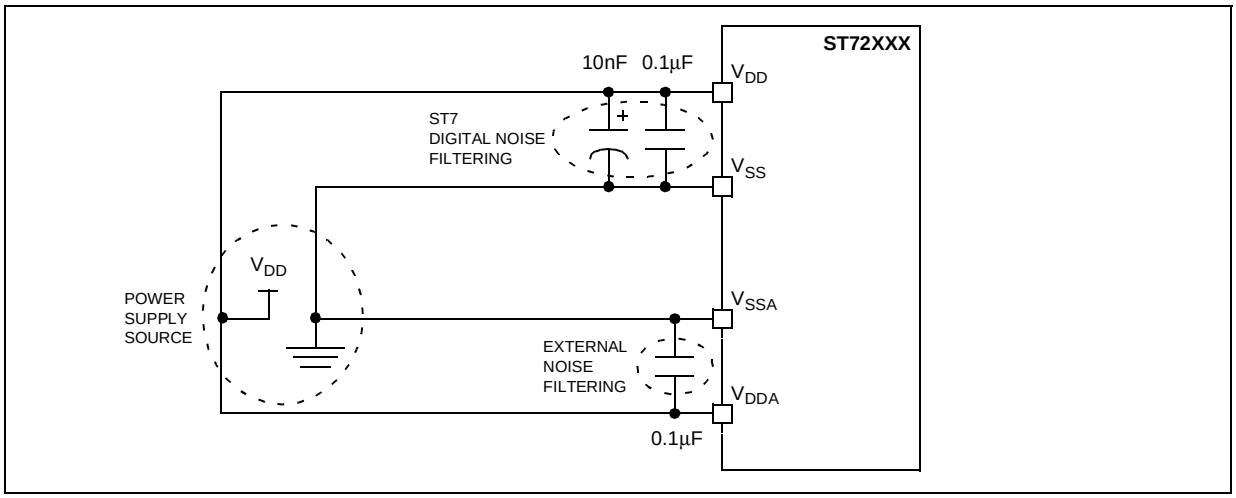
– ESD: Electro-Static Discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.

– 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	-4	4	

Figure 73. EMC Recommended star network power supply connection [2]



Notes:

1. Data based on characterization results, not tested in production.

2. The suggested 10nF and 0.1µF decoupling capacitors on the power supply lines are proposed as a good price vs. EMC performance tradeoff. 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)

9.7.2 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.

9.7.2.1 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 74 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.

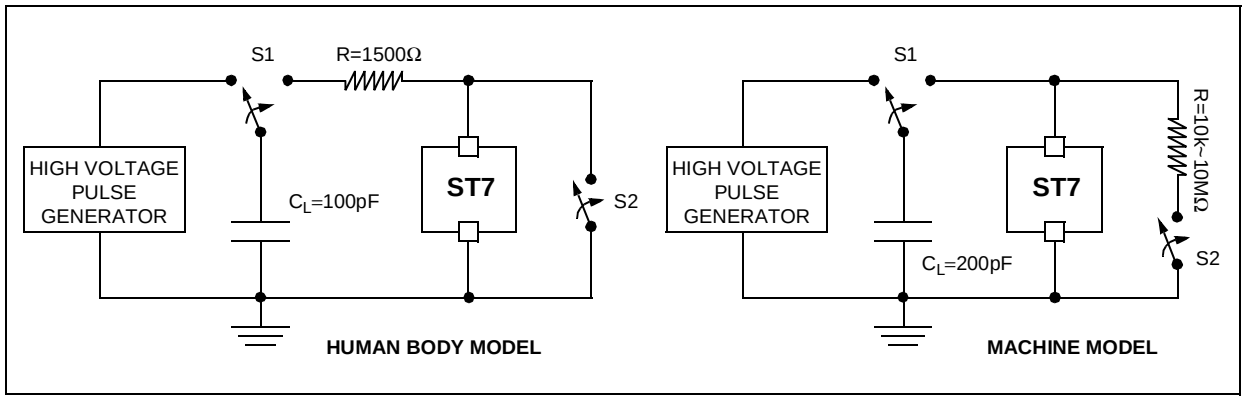
– S2 must be closed 10 to 100ms after the pulse delivery period to ensure the ST7 is not left in charge state. S2 must be opened at least 10ms prior to the delivery of the next pulse.

– 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	200	

Figure 74. Typical Equivalent ESD Circuits



Notes:

1. Data based on characterization results, not tested in production.

EMC CHARACTERISTICS (Cont'd)

9.7.2.2 Static and Dynamic Latch-Up

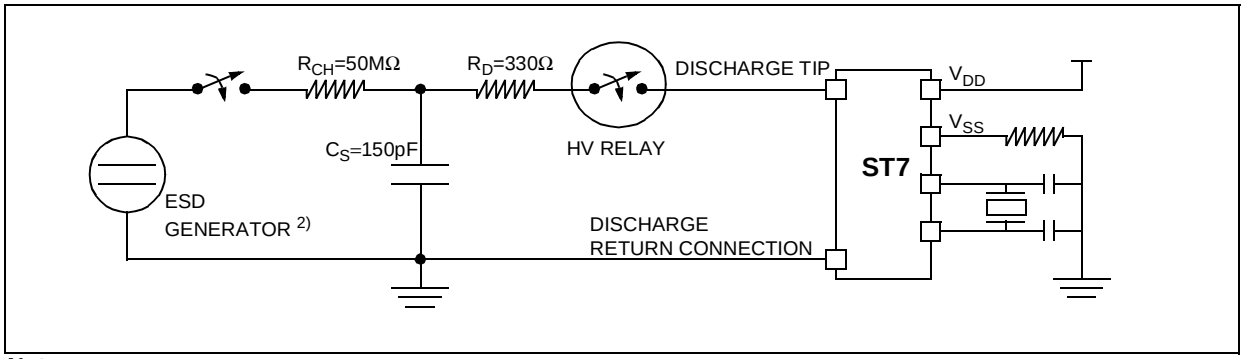
– LU: 3 complementary static tests are required on 10 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin), a current injection (applied to each input, output and configurable I/O pin) and a power supply switch sequence are performed on each sample. This test 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 75. 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	A A
DLU	Dynamic latch-up class	V_{DD}=5.5V, f_{OSC}=4MHz, T_A=+25°C	A

Figure 75. 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)

9.7.3 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 76 and Figure 77 for standard pins and in Figure 78 and Figure 79 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 76. Positive Stress on a Standard Pad vs. V_{SS}

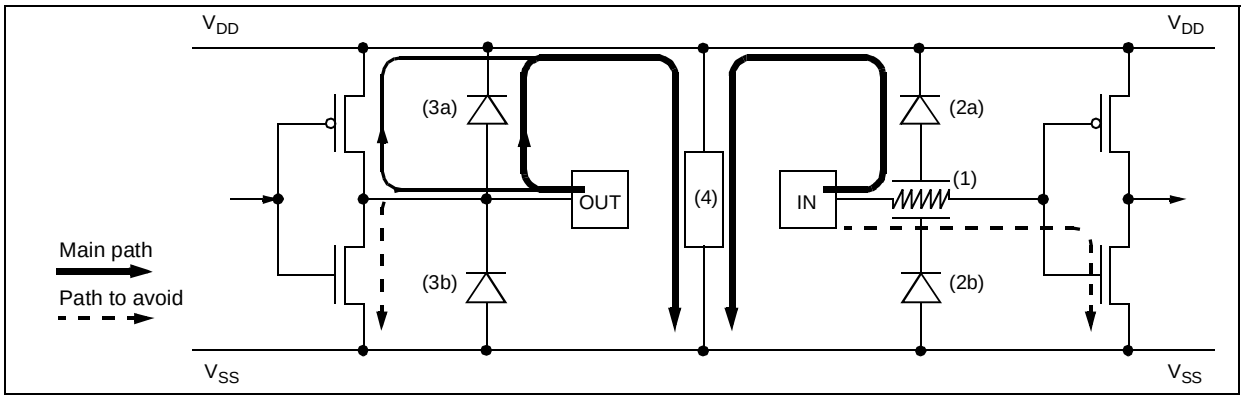
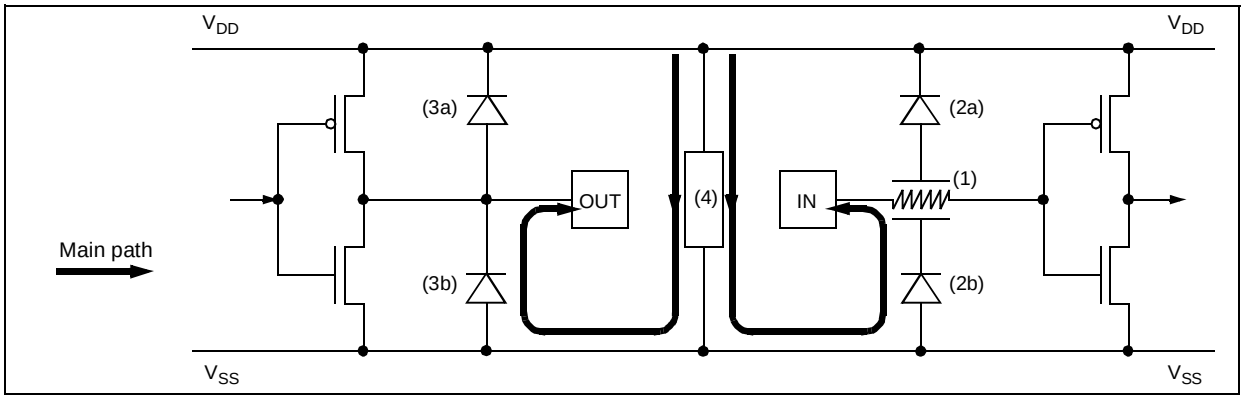


Figure 77. Negative Stress on a Standard Pad vs. V_{DD}



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 completly absorb the positive ESD discharge.

Multisupply Configuration

When several types of ground (V_{SS}, V_{SSA}, ...) and power supply (V_{DD}, V_{DDA}, ...) are available for any reason (better noise immunity...), the structure shown in Figure 80 is implemented to protect the device against ESD.

Figure 78. Positive Stress on a True Open Drain Pad vs. V_{SS}

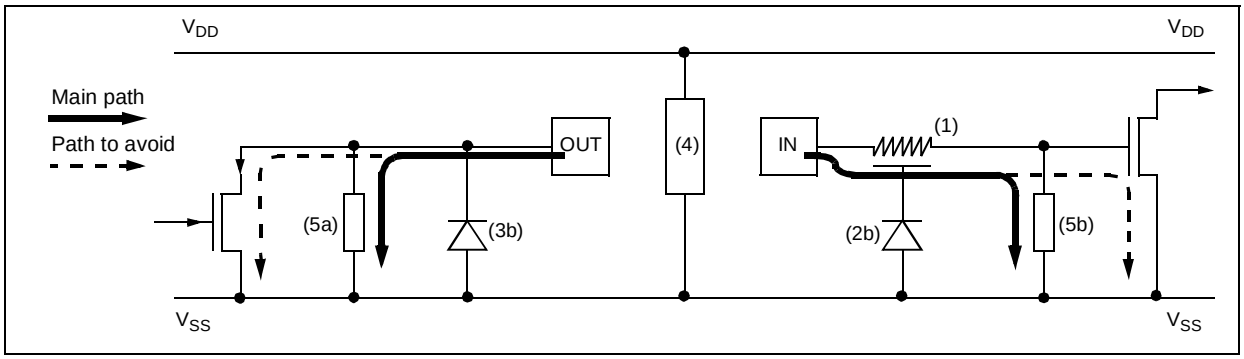


Figure 79. Negative Stress on a True Open Drain Pad vs. V_{DD}

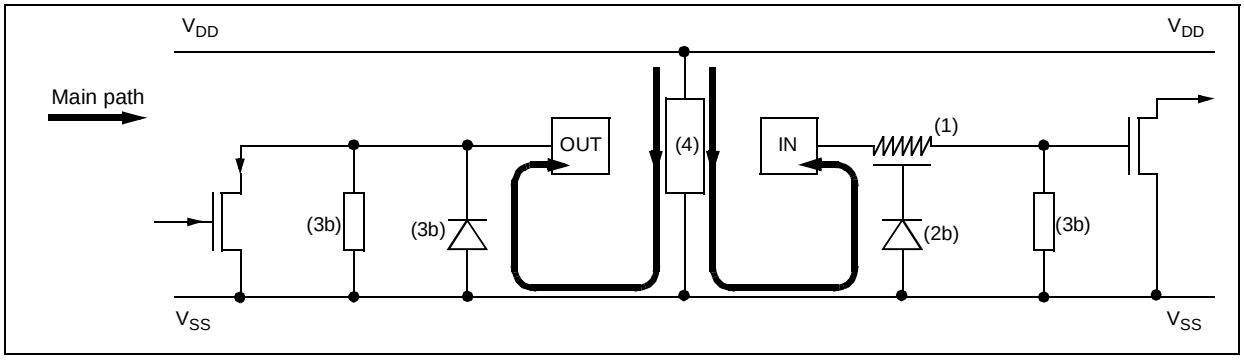
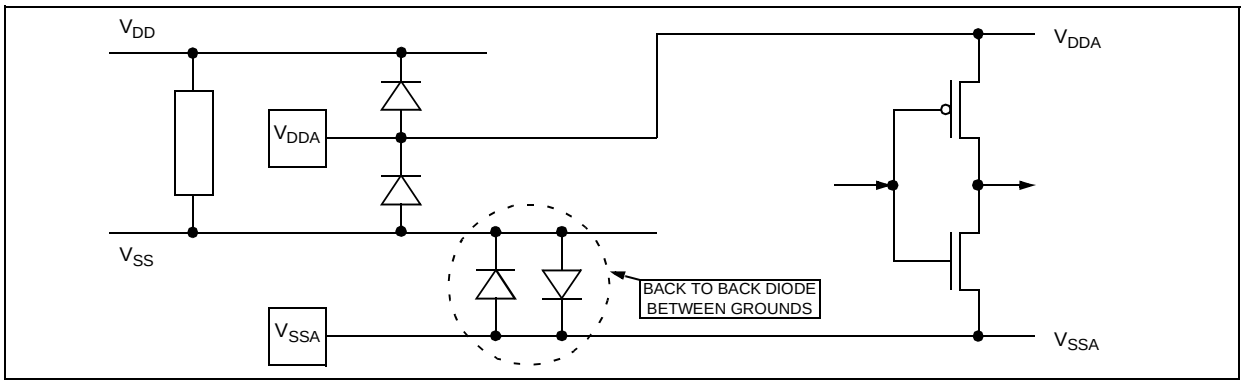


Figure 80. Multisupply Configuration



9.8 I/O PORT PIN CHARACTERISTICS

9.8.1 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]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
I_L	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$				±1	µA
I_S	Static current consumption [4]	Floating input mode				200	
R_{PU}	Weak pull-up equivalent resistor [5]	$V_{IN}=V_{SS}$	V_{DD}=5V	70	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 [6]	C_L=50pF Between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time [6]				25		
$t_{w(IT)in}$	External interrupt pulse time [7]			1			t_{CPU}

Figure 81. Two typical Applications with unused I/O Pin

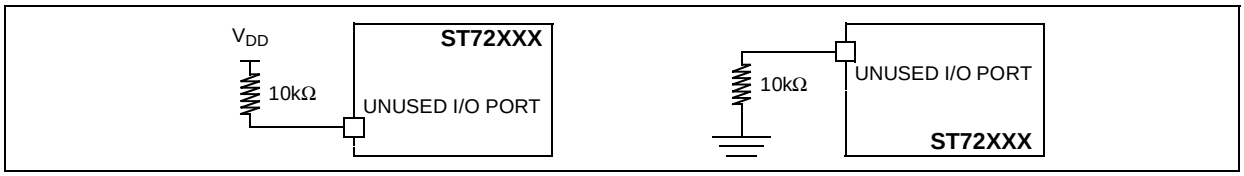
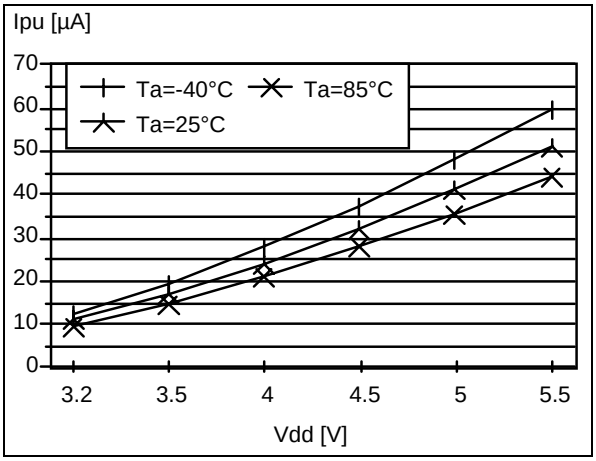


Figure 82. Typical I_{PU} vs. V_{DD} with $V_{IN}=V_{SS}$



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. 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 81). Data based on design simulation and/or technology characteristics, not tested in production.
5. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in Figure 82). This data is based on characterization results, tested in production at V_{DD} max.
6. Data based on characterization results, not tested in production.
7. 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)

9.8.2 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 83 and Figure 86)	V_{DD}=5V	I_{IO}=+5mA		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 85 and Figure 87)		I_{IO}=+20mA		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 86 and Figure 88)		I_{IO}=-5mA	V_{DD}-1.8		
			I_{IO}=-2mA	V_{DD}-0.7		

Figure 83. Typical V_{OL} at V_{DD}=5V (standard)

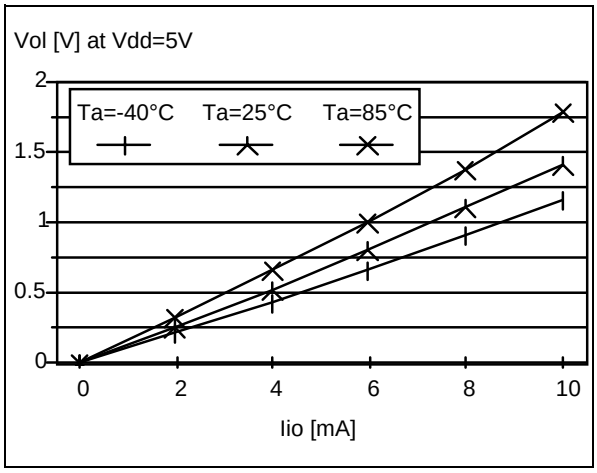


Figure 84. Typical V_{OL} at V_{DD}=5V (high-sink)

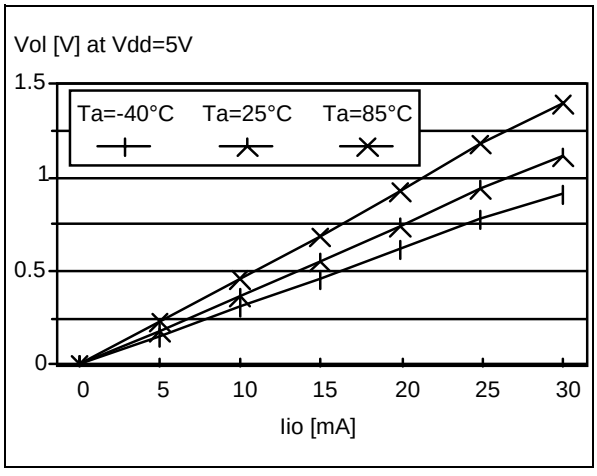
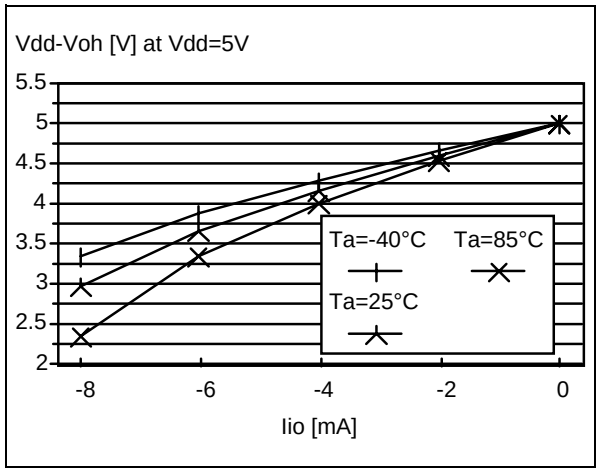


Figure 85. Typical V_{DD}-V_{OH} at V_{DD}=5V



Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 9.2.2 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 9.2.2 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 86. Typical V_{OL} vs. V_{DD} (standard I/Os)

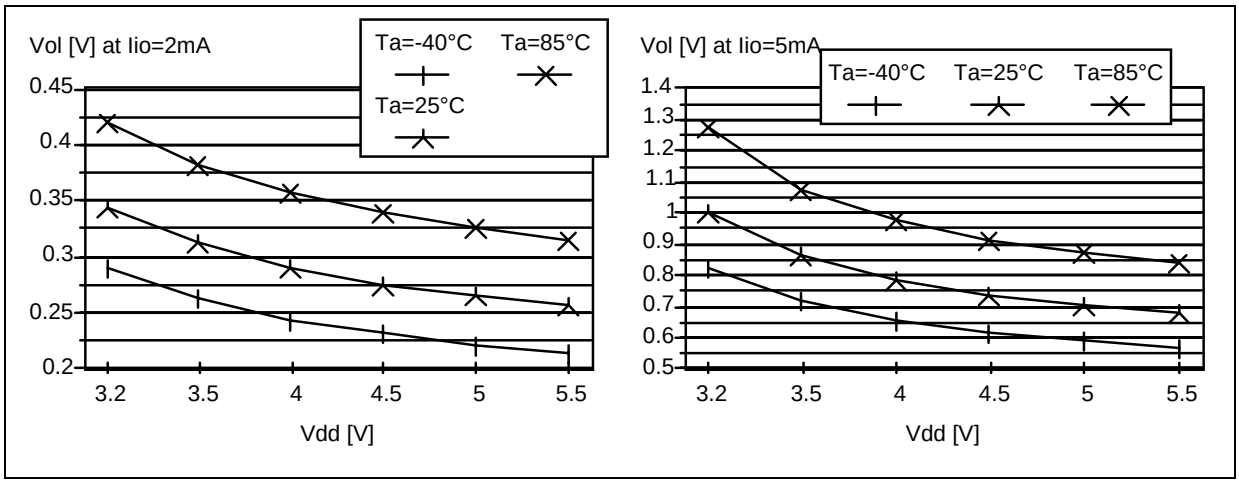


Figure 87. Typical V_{OL} vs. V_{DD} (high-sink I/Os)

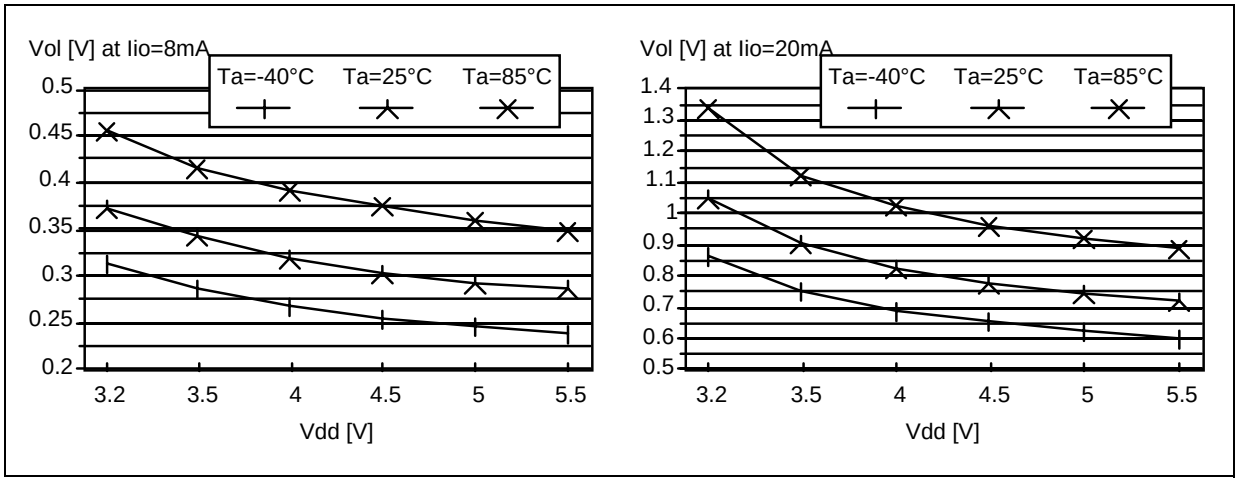
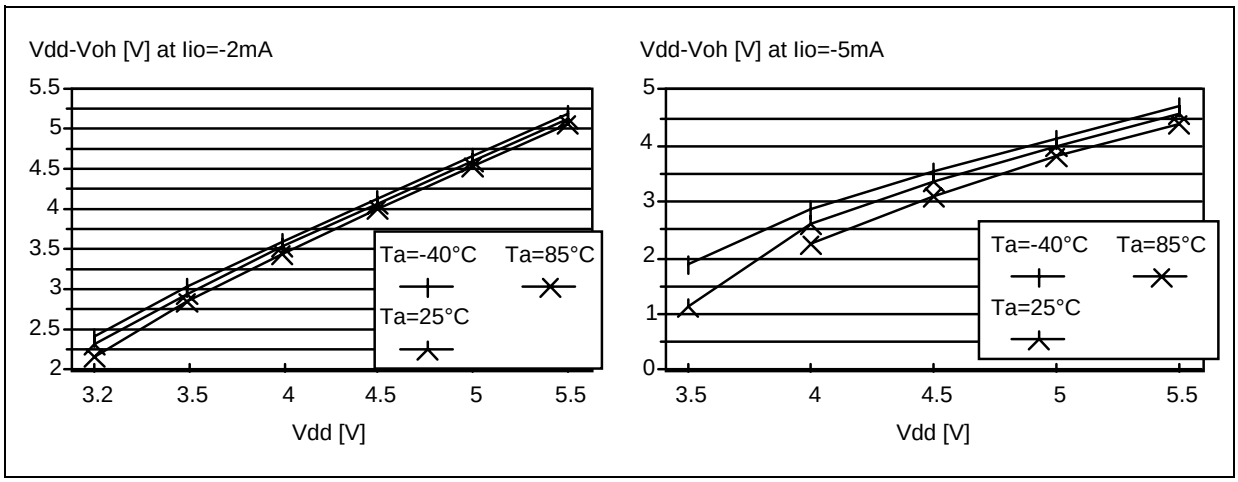


Figure 88. Typical V_{DD}-V_{OH} vs. V_{DD}



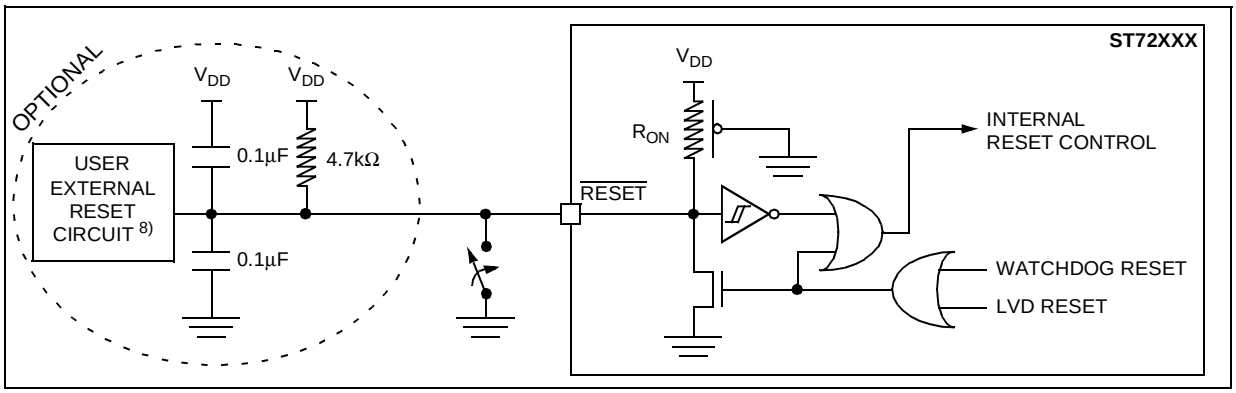
9.9 CONTROL PIN CHARACTERISTICS

9.9.1 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]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
V_{OL}	Output low level voltage [4] (see Figure 91, Figure 92)	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}=3.4V	80	100	120	
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or internal reset sources			6 30		$1/f_{SFOSC}$ µs
$t_{h(RSTL)in}$	External reset pulse hold time [6]			20			µs
$t_{g(RSTL)in}$	Filtered glitch duration [7]					100	ns

Figure 89. 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 9.2.2 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 (corresponding I_{ON} current characteristics described in Figure 90). This data is based on characterization results, not tested in production.
5. To guarantee the reset of the device, a minimum pulse has to be applied to $\overline{\text{RESET}}$ pin.
6. All short pulse 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 environment.
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)

Figure 90. Typical I_{ON} vs. V_{DD} with $V_{IN}=V_{SS}$

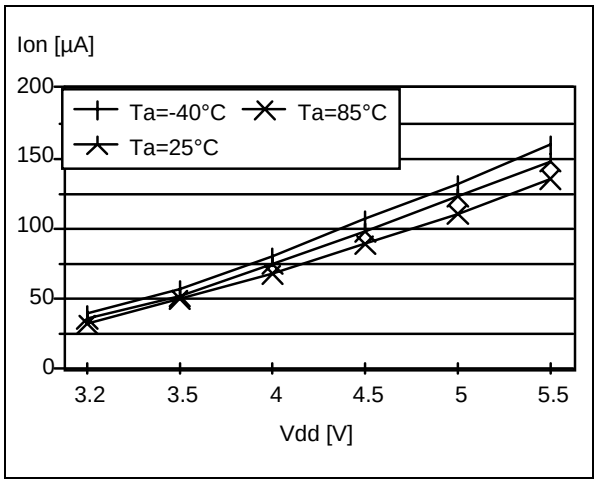


Figure 91. Typical V_{OL} at V_{DD}=5V ($\overline{RESET}$)

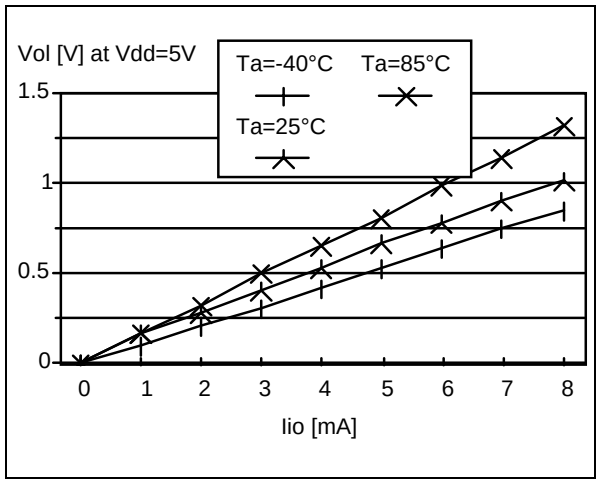
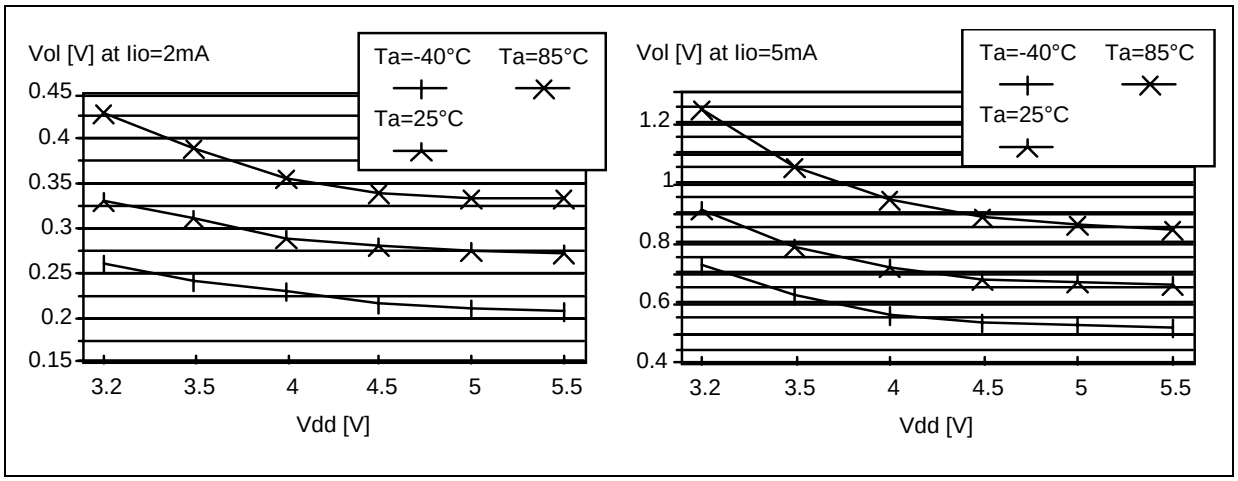


Figure 92. Typical V_{OL} vs. V_{DD} ($\overline{RESET}$)



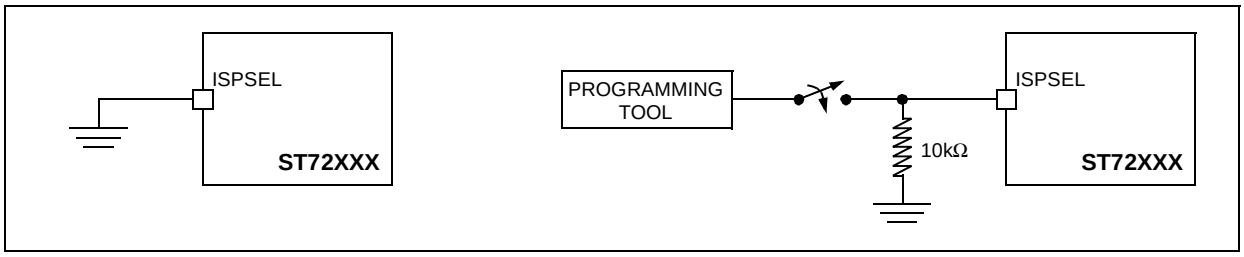
CONTROL PIN CHARACTERISTICS (Cont'd)

9.9.2 ISPSEL 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 93. Two typical Applications with ISPSEL Pin [2]



Notes:

1. Data based on design simulation and/or technology characteristics, not tested in production.
2. When the ISP Remote mode is not required by the application ISPSEL pin must be tied to V_{SS}.

9.10 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...).

9.10.1 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

9.10.2 8-Bit PWM Auto-reload 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		1			t_{CPU}
		f_{CPU}=8MHz	125			ns
f_{EXT}	Timer external clock frequency		0		$f_{CPU}/2$	MHz
f_{PWM}	PWM repetition rate		0		$f_{CPU}/2$	MHz
Res_{PWM}	PWM resolution				8	bit

9.10.3 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

9.11 COMMUNICATION INTERFACE CHARACTERISTICS

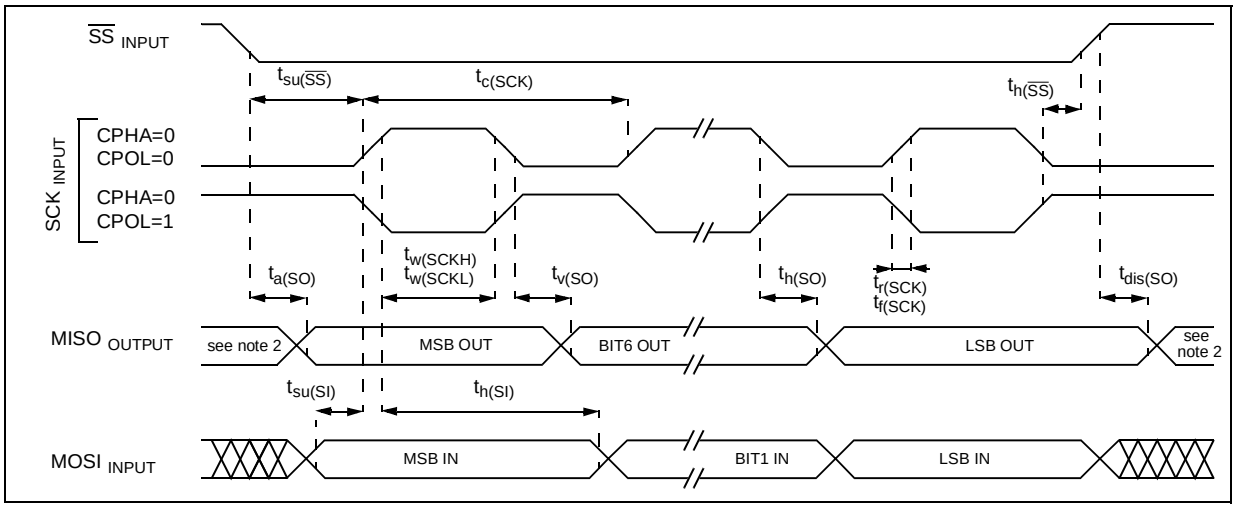
9.11.1 SPI - Serial Peripheral Interface

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		ns
$t_{h(\overline{SS})}$	$\overline{SS}$ 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		
$t_{h(MI)}$ $t_{h(SI)}$	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 94. 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 95. SPI Slave Timing Diagram with CPHA=1[1)]

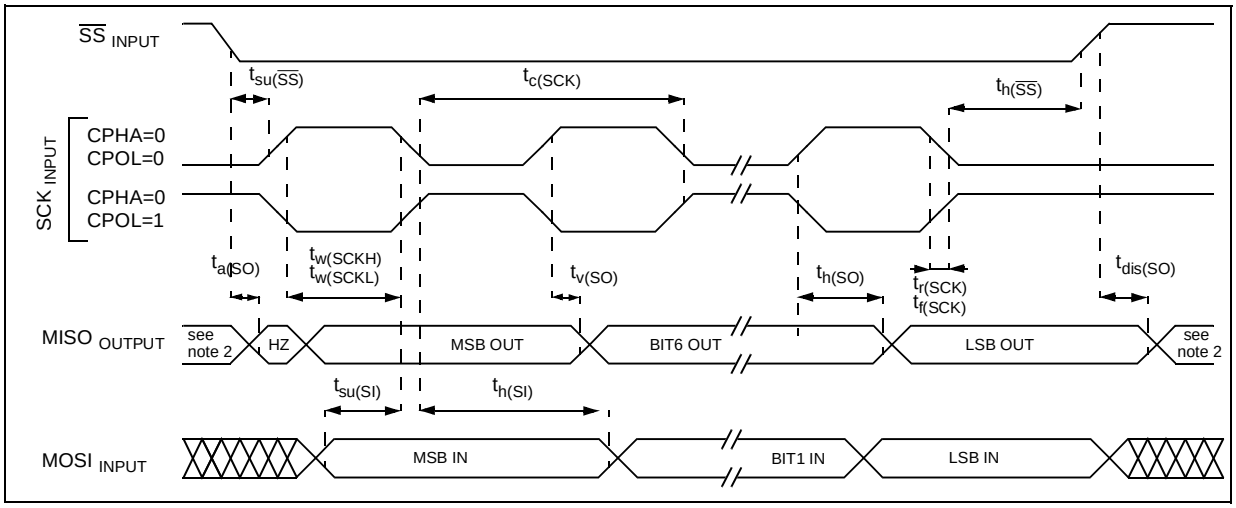
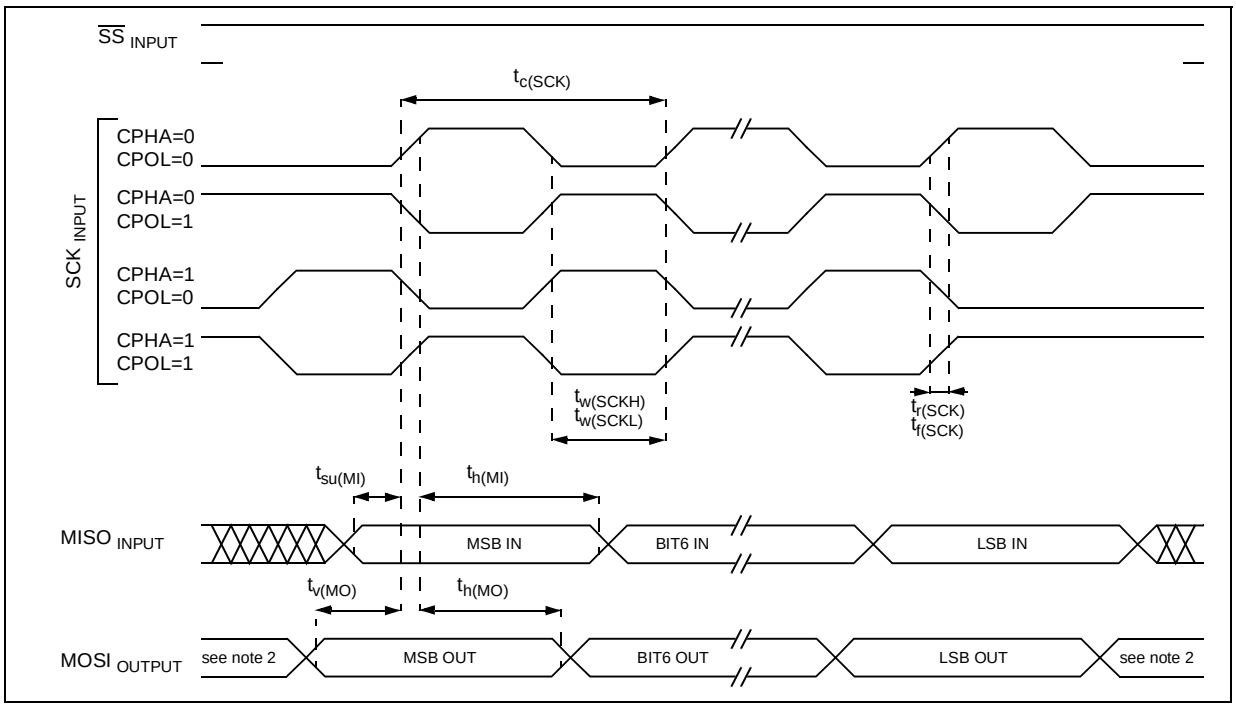


Figure 96. SPI Master Timing Diagram [1)]



Notes:

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

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)

9.11.2 SCI - Serial Communications 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 (RDI and TDO).

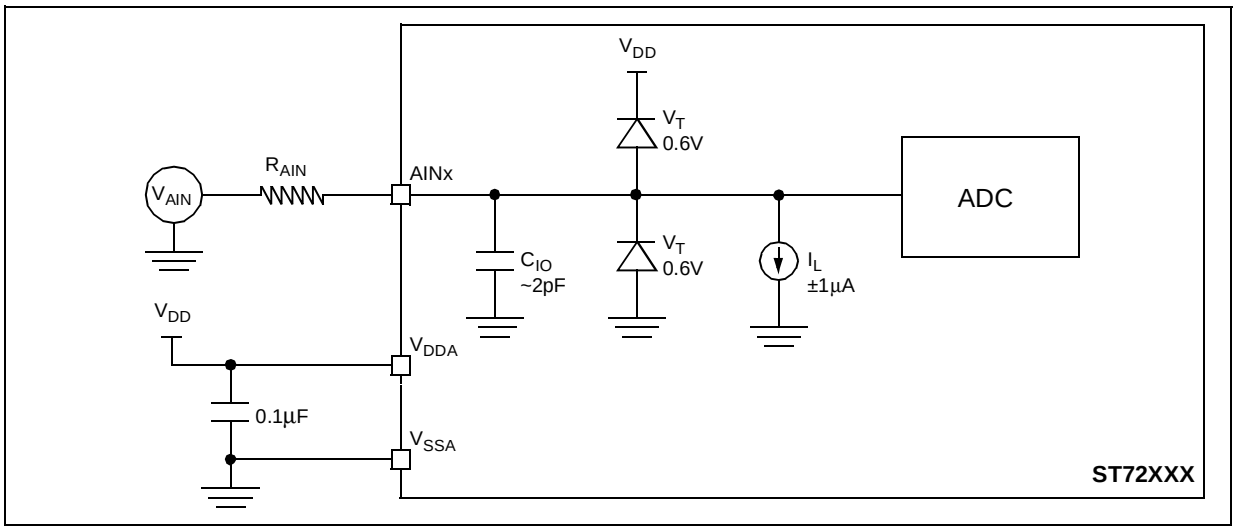
Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8MHz	~0.16%	Conventional Mode TR (or RR)=64, PR=13 TR (or RR)=16, PR=13 TR (or RR)= 8, PR=13 TR (or RR)= 4, PR=13 TR (or RR)= 2, PR=13 TR (or RR)= 8, PR= 3 TR (or RR)= 1, PR=13	 300 1200 2400 4800 9600 10400 19200	 ~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77	Hz
				Extended Mode ETPR (or ERPR) = 13	38400	~38461.54	
			~0.79%	Extended Mode ETPR (or ERPR) = 35	14400	~14285.71	

9.12 8-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 range voltage [2]		V_{SSA}		V_{DDA}	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)		3			µs
	- Sample capacitor loading time		4			$1/f_{ADC}$
	- Hold conversion time		8			$1/f_{ADC}$

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_{DDA} 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.

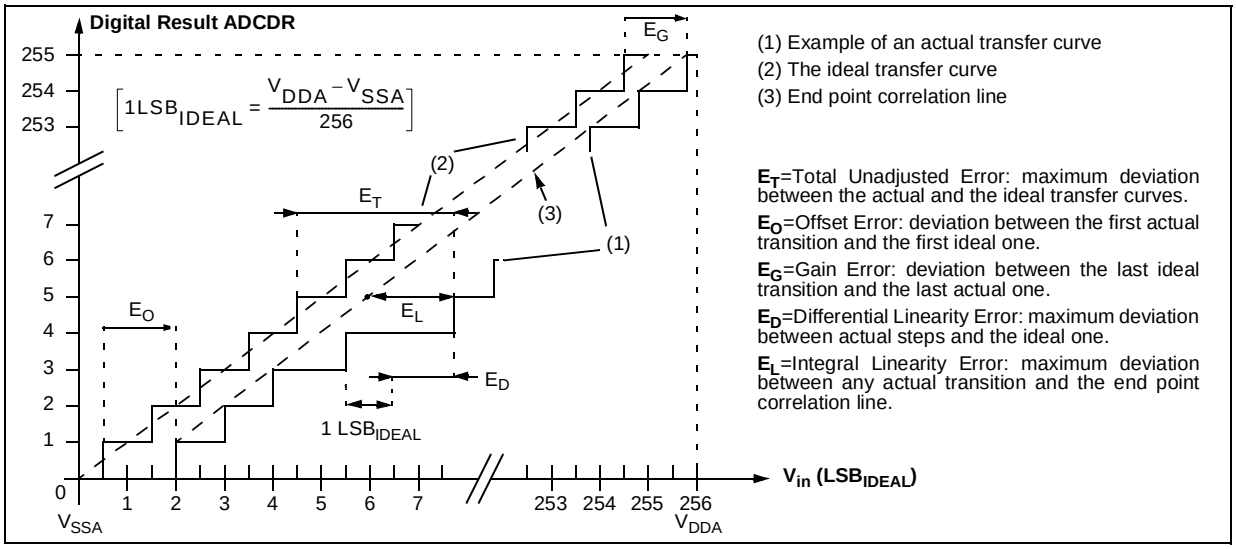
4. The stabilization time of the AD converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

8-BIT ADC CHARACTERISTICS (Cont'd)

ADC Accuracy

Symbol	Parameter	Conditions	Min	Max	Unit
$\|E_T\|$	Total unadjusted error [1]	V_{DD}=5.0V, [3] f_{CPU}=8MHz		1	LSB
E_O	Offset error [1]		-0.5	0.5	
E_G	Gain Error [1]		-0.5	0.5	
$\|E_D\|$	Differential linearity error [1]			0.5	
$\|E_L\|$	Integral linearity error [1]			0.5	

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 1 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.

3. Data based on characterization results over the whole temperature range, monitored in production.

9.13 OP-AMP Module Characteristics

These op-amp specific values take precedence over any generic values given elsewhere in the document.

($T = 25^oC$, $V_{DD} - V_{SS} = 5$ V).

SPGA1 / SPGA2 - Software Programmable Gain Operational Amplifiers						
Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
|Vio|	Input Offset Voltage			3	10	mV
I_{CC}	Supply Current per amplifier [2]	V_{DD}=5.0V, A_{VCL}=1, no load[1]		0.8	2	mA
CMR [3]	Common Mode Rejection Ratio		70			dB
SVR [3]	Supply Voltage Rejection Ratio		70			dB
Avd [3]	Voltage Gain	(R_L=1KΩ)	100			V/mV
V_{OH}	High Level Ouput Voltage	(R_L=10KΩ) V_{DD}=5V	4.9			V
V_{OL}	Low Level Ouput Voltage	(R_L=10KΩ) V_{DD}=5V			0.10	V
I_{SC} [3]	Short circuit Current Sourced Short circuit Current Sunk	Vo= 5V connected to V_{SS} Vo= 0V connected to V_{DD}	45 70			mA mA
GPB	Gain Bandwidth Product			4		MHz
SR^+	Slew Rate [5]	A_{VCL}=1[1]		1		V/µs
SR^-	Slew Rate [5]	A_{VCL}=1[1]		1		V/µs
en [3]	Thermal Noise				50	$nV\sqrt{Hz}^{-1}$
Φm [3]	Phase margin		40	55		Degrees
Cin [4]	Input Capacitance			10		pF
V_{icm} [4]	Common Mode Input Voltage Range		V_{SS}-0.2		V_{DD}+0.2	V
ΔVRef	Reference Voltage (V_{DDA}/8 step) Precision				±10	%
ΔV_{BG}	Band Gap Precision				±10	%
ΔGain	Programmable Gain Precision				±10	%

1) A_{VCL} = Closed loop gain (repeater configuration)

2) Tested with positve input connected to internal band gap (reference voltage enabled) and negative input floating.

3) Data based on characterization, not tested in production

4) Data guaranteed by design, not tested in production

5) Slew rate is the rate of change from 10% to 90% of the output voltage step.

OP-AMP MODULE CHARACTERISTICS (Cont'd)

OA3 Operational Amplifier						
Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
|Vio|	Input Offset Voltage			3	10	mV
I_{CC}	Supply Current per amplifier	V_{DD}=5.0V, A_{VCL}=1, no load[1)]		300	500	µA
CMR [2)]	Common Mode Rejection Ratio		70			dB
SVR [2)]	Supply Voltage Rejection Ratio		70			dB
Avd [2)]	Voltage Gain	(R_L=1KΩ)	100			V/mV
V_{OH}	High Level Ouput Voltage	(R_L=10KΩ) V_{DDA}=5V	4.9			V
V_{OL}	Low Level Ouput Voltage	(R_L=10KΩ) V_{DDA}=5V			0.10	V
I_{SC} [2)]	Short circuit Current Sourced Short circuit Current Sunk	Vo= 1 connected to V_{SS} Vo= 0 connected to V_{DD}	45 70			mA mA
GPB	Gain Bandwidth Product			6		MHz
SR^+	Slew Rate[4)]	A_{VCL}=1[1)]		1		V/µs
SR^-	Slew Rate[4)]	A_{VCL}=1[1)]		1		V/µs
en [2)]	Thermal Noise				50	$nV\sqrt{Hz}^{-1}$
Φm [2)]	Phase margin		40	55		Degrees
Cin [3)]	Input Capacitance			10		pF
V_{icm} [3)]	Common Mode Input Voltage Range		V_{SS}-0.2		V_{DD}+0.2	V

1) A_{VCL} = Closed loop gain (repeater configuration)

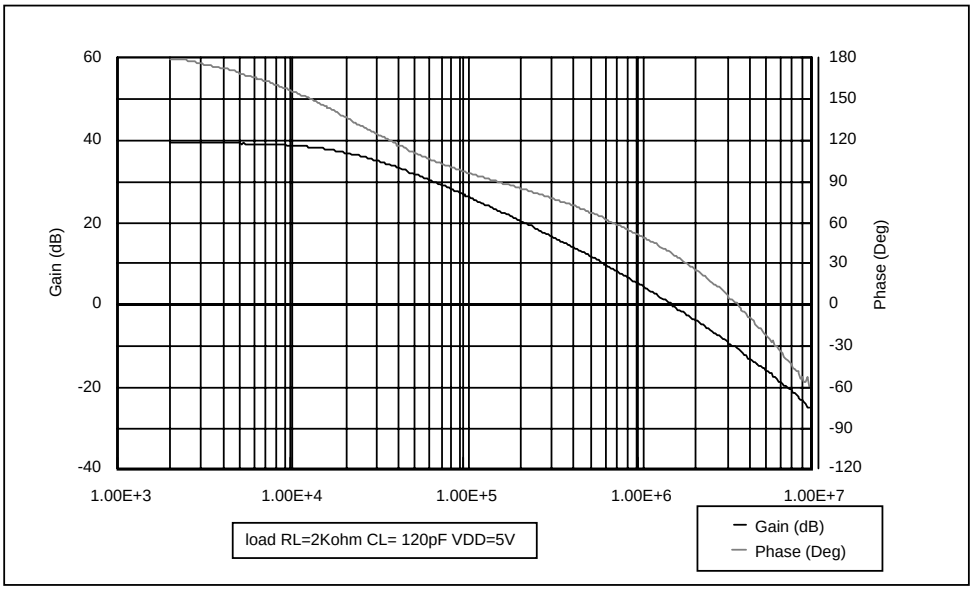
2) Data based on characterization, not tested in production

3) Data guaranteed by design, not tested in production

4) Slew rate is the rate of change from 10% to 90% of the output voltage step.

9.13.1 Typical Phase Gain vs. Frequency

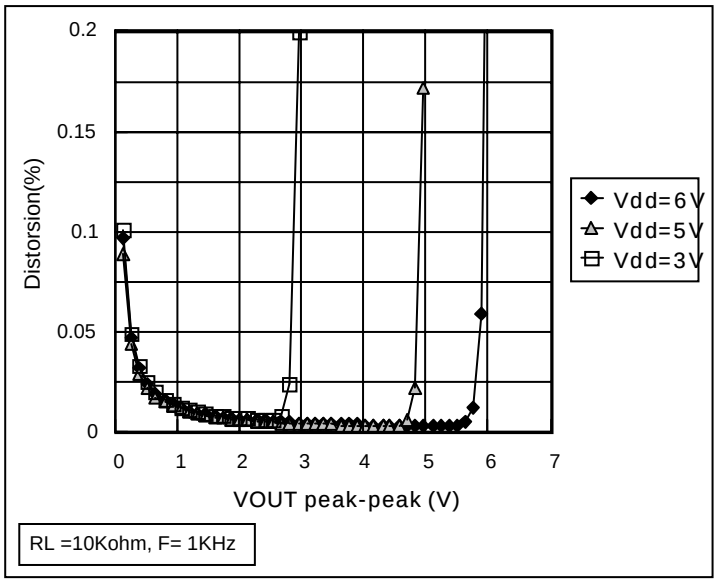
Figure 99. Gain vs Frequency



9.13.2 Typical Total Harmonic Distorsion

Figure 100 shows three typical curves for different V_{DD} values. This characterisation has been done at T_A 25°C using a 1 kHz sine wave signal with an average value of $V_{DD}/2$. This signal is input to the SPGA configured in non-inverter mode with a gain of 1. The SPGA output is loaded with a 1K resistor.

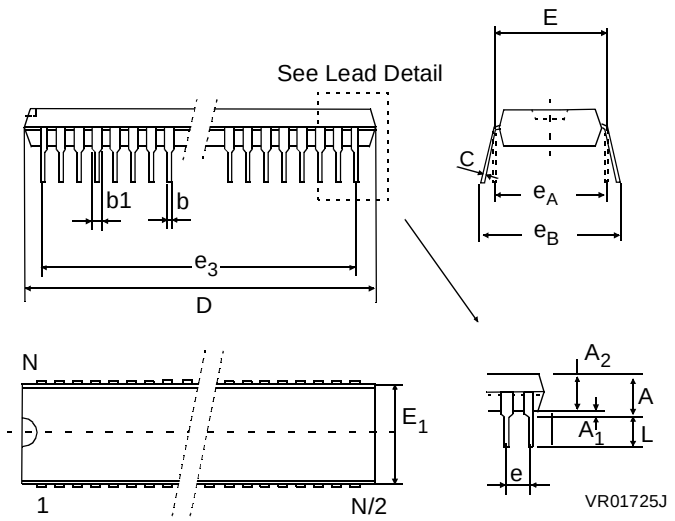
Figure 100. Total Harmonic Distorsion vs Vout



10 GENERAL INFORMATION

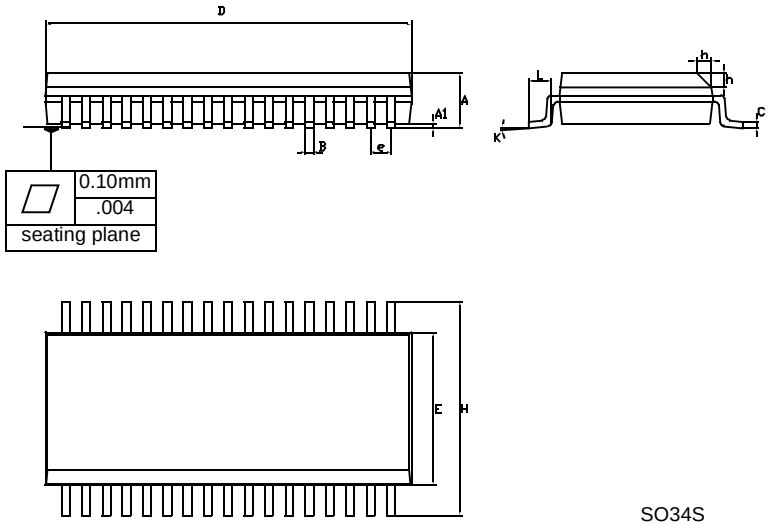
10.1 PACKAGE MECHANICAL DATA

Figure 101. 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					

Figure 102. 34-Pin Small Outline



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.46		2.64	0.097		0.104
A1	0.13		0.29	0.005		0.0115
B	0.36		0.48	0.014		0.019
C	0.23		0.32	0.0091		0.0125
D	17.73		18.06	0.698		0.711
E	7.42		7.59	0.292		0.299
e		1.02			0.040	
H	10.16		10.41	0.400		0.410
h	0.64		0.74	0.025		0.029
K				0°		8°
L	0.61		1.02	0.024		0.040
	Number of Pins					
N	34					

10.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) SDIP32 SO34	 60 70	°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.

10.3 SOLDERING AND GLUEABILITY INFORMATION

Recommended soldering information given only as design guidelines in Figure 103 and Figure 104.

Recommended glue for SMD plastic packages dedicated to molding compound with silicone:

- Heraeus: PD945, PD955
- Loctite: 3615, 3298

Figure 103. Recommended Wave Soldering Profile (with 37% Sn and 63% Pb)

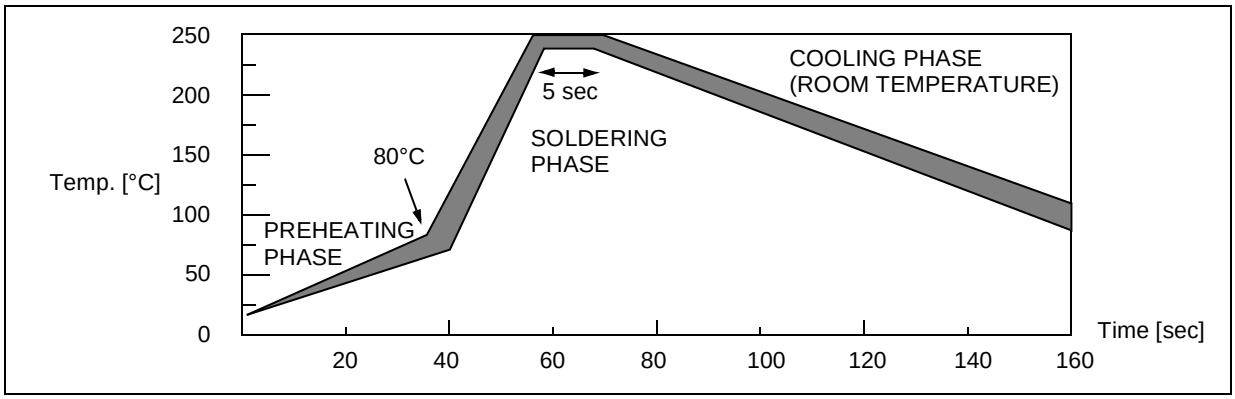
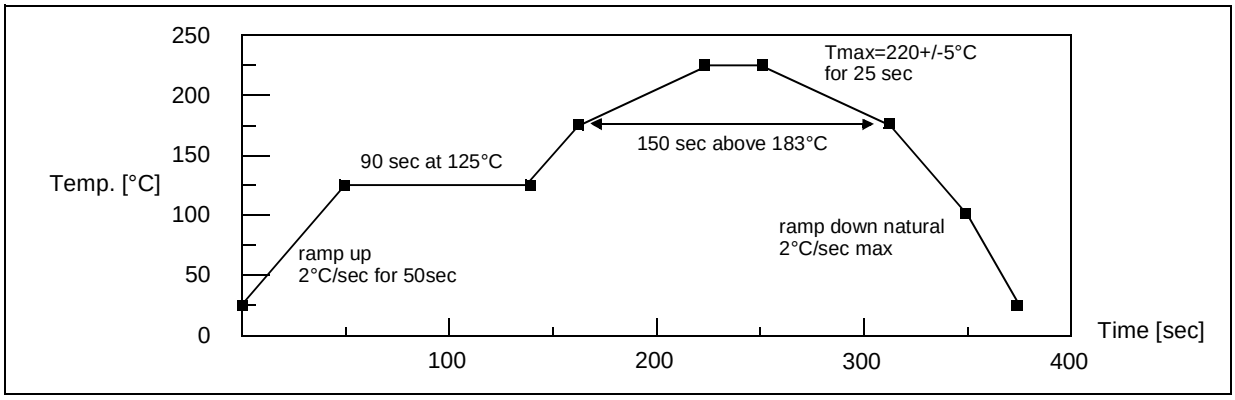


Figure 104. Recommended Reflow Soldering Oven Profile (MID JEDEC)



10.4 PACKAGE/SOCKET FOOTPRINT PROPOSAL

Table 23. Suggested List of SDIP32 Socket Types

Package / Probe	Adaptor / Socket Reference	Same Footprint	Socket Type
SDIP32 EMU PROBE	TEXTOOL 232-1291-00	X	Textool

Table 24. Suggested List of SO34 Socket Types

Package / Probe	Adaptor / Socket Reference	Same Footprint	Socket Type
SO34 EMU PROBE	Emulator Probe includes an adapter with S034 footprint to be soldered on user PCB	X	N/A

11 DEVICE CONFIGURATION AND ORDERING INFORMATION

The device is available for production a user programmable version (FLASH). FLASH devices are shipped to customers with a default content (FFh). FLASH devices have to be configured by the customer using the Option Bytes.

11.1 OPTION BYTES

The two option bytes allow the hardware configuration of the microcontroller to be selected.
The option bytes have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the FLASH is fixed to FFh.
In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

USER OPTION BYTE 0

Bit 7:1 = **Reserved**, must always be 1.

Bit 1= **OA3E** *Op-Amp 3 Enable*
This option bit enables or disables the third Op-Amp of the on-chip Op-Amp Module.
0: OE3 disabled
1: OE3 enabled

Bit 0 = **FMP** *Full memory protection.*
This option bit enables or disables external access to the internal program memory (read-out protection). Clearing this bit causes the erasing (to 00h) of the whole memory (including the option byte).
0: Program memory not read-out protected
1: Program memory read-out protected

USER OPTION BYTE 1

Bit 7 = **CFC** *Clock filter control on/off*
This option bit enables or disables the clock filter (CF) features.
0: Clock filter enabled
1: Clock filter disabled

Bit 6:4 = **OSC[2:0]** *Oscillator selection*
These three option bits can be used to select the main oscillator as shown in Table 25.

Bit 3:2 = **LVD[1:0]** *Low voltage detection selection*
These option bits enable the LVD block with a selected threshold as shown in Table 26.

Bit 1 = **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

Bit 0 = **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)

Table 25. Main Oscillator Configuration

Selected Oscillator	OSC2	OSC1	OSC0
External Clock (Stand-by)	1	1	1
~4 MHz Internal RC	1	1	0
1~14 MHz External RC	1	0	X
Low Power Resonator (LP)	0	1	1
Medium Power Resonator (MP)	0	1	0
Medium Speed Resonator (MS)	0	0	1
High Speed Resonator (HS)	0	0	0

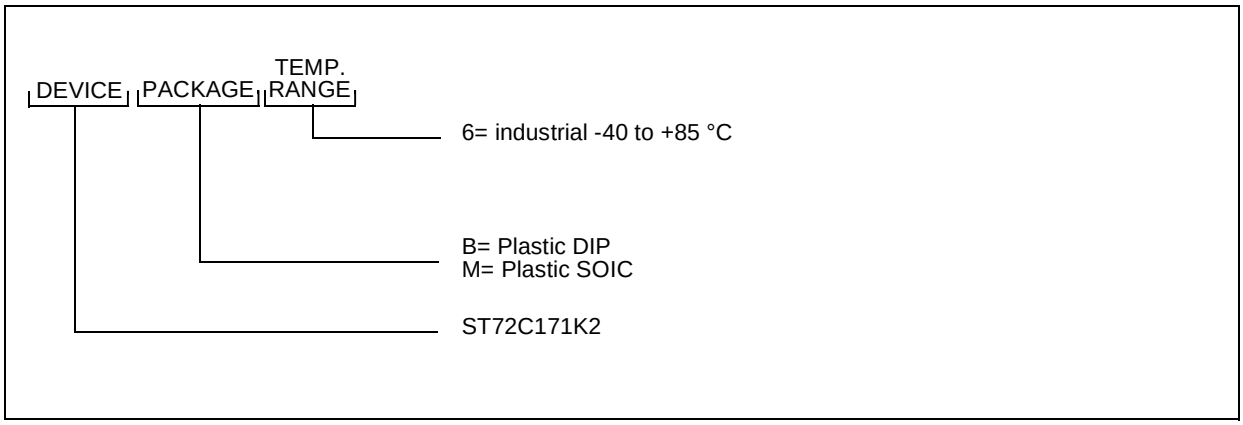
Table 26. LVD Threshold Configuration

Configuration	LVD1	LVD0
LVD Off	1	1
Highest Voltage Threshold (~4.50V)	1	0
Medium Voltage Threshold (~4.05V)	0	1
Lowest Voltage Threshold (~3.45V)	0	0

	USER OPTION BYTE 0								USER OPTION BYTE 1							
	7							0	7							0
	Reserved						OA3E	FMP	CFC	OSC 2	OSC 1	OSC 0	LVD1	LVD0	WDG HALT	WDG SW
Default Value	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1

11.2 DEVICE ORDERING INFORMATION

Figure 105. FLASH User Programmable Device Type



11.3 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.

Third Party Tools

- ACTUM
- BP
- COSMIC
- CMX
- DATA I/O
- HITEX
- HIWARE
- ISYSTEM
- KANDA
- LEAP

Tools from these manufacturers include C compliers, emulators and gang programmers.

STMicroelectronics Tools

Two types of development tool are offered by ST, all of them connect to a PC via a parallel (LPT) port: see Table 27 and Table 28 for more details.

Table 27. STMicroelectronic Tool Features

	In-Circuit Emulation	Programming Capability[1)]	Software Included
ST7 HDS2 Emulator	Yes, powerful emulation features including trace/ logic analyzer	No	ST7 CD ROM with: – ST7 Assembly toolchain – STVD7 and WGDB7 powerful Source Level Debugger for Win 3.1, Win 95 and NT – C compiler demo versions – ST Realizer for Win 3.1 and Win 95. – Windows Programming Tools for Win 3.1, Win 95 and NT
ST7 Programming Board	No	Yes (All packages),support also ISP [1)]	

Table 28. Dedicated STMicroelectronics Development Tools

Supported Product	ST7 HDS2 Emulator	ST7 Programming Board
ST72C171K2,	ST7MDT6-EMU2B	ST7MDT6-EPB2/EU ST7MDT6-EPB2/US ST7MDT6-EPB2/UK

Note:
1. In-Situ Programming (ISP) interface for FLASH devices.

11.4 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	PERMANENT MAGNET DC MOTOR DRIVE.
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

11.5 TO GET MORE INFORMATION

To get the latest information on this product please use the ST web server: http://mcu.st.com/

12 SUMMARY OF CHANGES

Description of the changes between the current release of the specification and the previous one.

Revision	Main changes	Date
1.4	Added Figure 99 and Figure 100.	Oct-00

Notes: