ST7L15
ST7L19

8-bit MCU for automotive with single voltage Flash/ROM memory, data EEPROM, ADC, 5 timers, SPI

Features

- Memories
  - 4 Kbytes program memory: Single voltage extended Flash (XFlash) or ROM with readout protection capability. In-application programming and in-circuit programming (IAP and ICP) for XFlash devices
  - 256 bytes RAM
  - 128 bytes data EEPROM (XFlash and ROM devices) with readout protection, 300 K write/erase cycles guaranteed
  - XFlash and EEPROM data retention 20 years at 55°C
- Clock, reset and supply management
  - Enhanced reset system
  - Enhanced low voltage supervisor (LVD) for main supply
  - Clock sources: Internal 1% RC oscillator, crystal/ceramic resonator or external clock
  - Optional x4 or x8 PLL for 4 or 8 MHz internal clock (x4 PLL only available for Flash devices)
  - 5 power saving modes: Halt, active halt, auto wakeup from halt, wait and slow
- I/O ports
  - Up to 17 multifunctional bidirectional I/O lines
  - 7 high sink outputs

SO20
300mil

- 5 timers
  - Configurable watchdog timer
  - Two 8-bit lite timers with prescaler, 1 real-time base and 1 input capture
  - Two 12-bit autoreload timers with 4 PWM outputs, 1 input capture, 1 pulse and 4 output compare functions
- Communication interface
  - SPI synchronous serial interface
- Interrupt management
  - 12 interrupt vectors plus TRAP and reset
  - 15 external interrupt lines (on 4 vectors)
- A/D converter
  - 7 input channels
  - 10-bit precision
- Instruction set
  - 8-bit data manipulation
  - 63 basic instructions with illegal opcode detection
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instructions
- Development tools
  - Full hardware/software development package
  - DM (debug module)

Table 1. Device summary

Features	ST7L15	ST7L19
Program memory - bytes	4 K	
RAM (stack) - bytes	256 (128)	
Data EEPROM - bytes	-	128
Peripherals	Lite timer with watchdog, autoreload timer, SPI, 10-bit ADC	
Operating supply	3 V to 5.5 V	
CPU frequency	Up to 8 MHz (w/ext OSC up to 16 MHz and int 1 MHz RC 1%, PLLx8/4 MHz)	
Operating temperature	Up to -40 to +85°C/-40 to +125°C	
Packages	SO20 300mil	

Contents

List of tables

List of figures

1 Description

The ST7L1x is a member of the ST7 microcontroller family suitable for automotive applications. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

The ST7L1 features Flash memory with byte-by-byte in-circuit programming (ICP) and in-application programming (IAP) capability.

Under software control, the ST7L1 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.

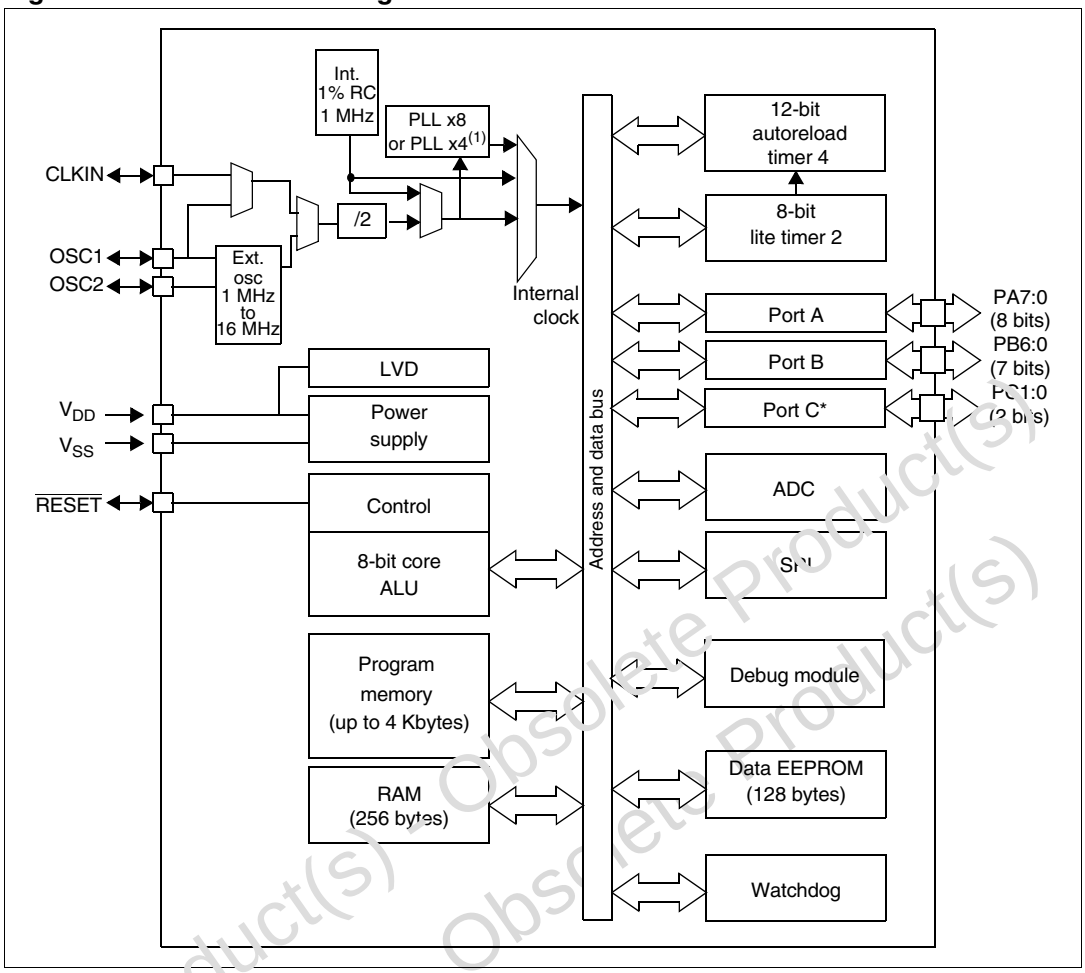
1.1 Parametric data

For easy reference, all parametric data is located in *Section 13. Electrical characteristics*.

1.2 Debug module

The ST7L1 features an on-chip debug module (DM) to support in-circuit debugging (ICD). For a description of the DM registers, refer to the *ST7 ICC protocol reference manual*.

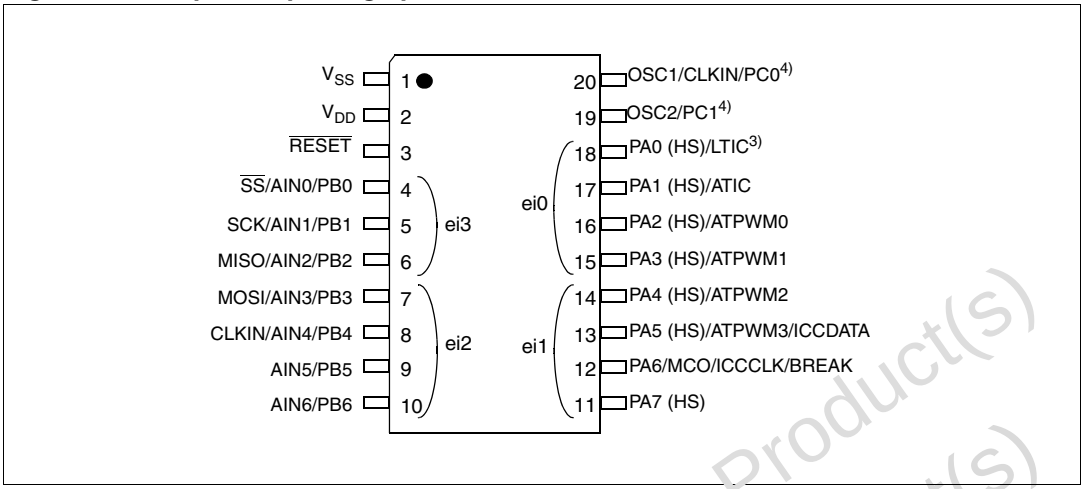
Figure 1. General block diagram



1. Not available on ROM devices

2 Pin description

Figure 2. 20-pin SO package pinout



1. eix: Associated external interrupt vector
2. (HS): 20mA high sink capability
3. This pin cannot be configured as external interrupt in ROM devices
4. OSC1 and OSC2 are not multiplexed in ROM devices and port C is not present

Table 2. Device pin description(1)(2)

Pin no. SO20	Pin name	Type	Level		Port/control						Main function (after reset)	Alternate function
					Input				Output			
			Input	Output	float	wpu	int	ana	OD	PP		
1	V_{SS}	S									Ground	
2	V_{DD}	S									Main power supply	
3	$\overline{RESET}$	I/O	C_T			X			X		Top priority non maskable interrupt (active low)	
4	PB0/AIN0/$\overline{SS}$	I/O	C_T		X	ei3		X	X	X	Port B0	ADC analog input 0 or SPI slave select (active low) **Caution:** No negative current injection allowed on this pin
5	PB1/AIN1/SCK	I/O	C_T		X			X	X	X	Port B1	ADC analog input 1 or SPI serial clock
6	PB2/AIN2/MISO	I/O	C_T		X			X	X	X	Port B2	ADC analog input 2 or SPI master in/slave out data
7	PB3/AIN3/MOS	I/O	C_T		X	ei2		X	X	X	Port B3	ADC analog input 3 or SPI master out/slave in data
8(3)	PB4/AIN4/CLKIN/ COMPIN-	I/O	C_T		X			X	X	X	Port B4	ADC analog input 4 or external clock input
9(3)	PB5/AIN5	I/O	C_T		X			X	X	X	Port B5	ADC analog input 5
10(3)	PB6/AIN6	I/O	C_T		X			X	X	X	Port B6	ADC analog input 6
11(3)	PA7	I/O	C_T	HS	X	ei1			X	X	Port A7	
12	PA6 /MCO/ICCCLK/ BREAK	I/O	C_T		X	ei1			X	X	Port A6	Main clock output or in-circuit communication clockor external BREAK **Caution:** During normal operation this pin must be pulled- up, internally or externally (external pull-up of 10 K mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset puts it back in input pull-up
13	PA5/ICCDATA/ ATPWM3	I/O	C_T	HS	X	ei1			X	X	Port A5	In-circuit communication data or autoreload timer PWM3
14	PA4/ATPWM2	I/O	C_T	HS	X				X	X	Port A4	Autoreload timer PWM2

Table 2. Device pin description[1][2]

Pin no. SO20	Pin name	Type	Level		Port/control						Main function (after reset)	Alternate function
					Input				Output			
			Input	Output	float	wpu	int	ana	OD	PP		
15	PA3/ATPWM1	I/O	C_T	HS	X	ei0			X	X	Port A3	Autoreload timer PWM1
16	PA2/ATPWM0	I/O	C_T	HS	X				X	X	Port A2	Autoreload timer PWM0
17	PA1/ATIC	I/O	C_T	HS	X				X	X	Port A1	Autoreload timer input capture
18[3]	PA0/LTIC	I/O	C_T	HS	X				X	X	Port A0	Lite timer input capture
19[4]	OSC2/PC1	I/O			X					X	Port C1[5]	Resonator oscillator inverter output
20[4]	OSC1/CLKIN/PC0	I/O			X					X	Port C0[5]	Resonator oscillator inverter input or external clock input

1. Legend/abbreviations for *Table 2*:
   Type: I = input, O = output, S = supply
   Input level: C_T = CMOS 0.3 V_{DD}/0.7 V_{DD} with input trigger
   Output level: HS = 20mA high sink (on N-buffer only)
   Port and control configuration inputs: float = floating, wpu = weak pull-up, int = interrupt, ana = analog ports
   Port and control configuration outputs: OD = open drain, PP = push-pull
2. The reset configuration of each pin (shown in bold) is valid as long as the device is in reset state
3. This pin cannot be configured as external interrupt in ROM devices
4. OSC1 and OSC2 are not multiplexed in ROM devices and port C is not present
5. PCOR not implemented but p-transistor always active in output mode (refer to *Figure 29: I/O port general block diagram on page 70*)

3 Register and memory map

As shown in *Figure 3*, the MCU can address 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, 256 bytes of RAM, 128 bytes of data EEPROM and up to 4 Kbytes of Flash program memory. The RAM space includes up to 128 bytes for the stack from 180h to 1FFh.

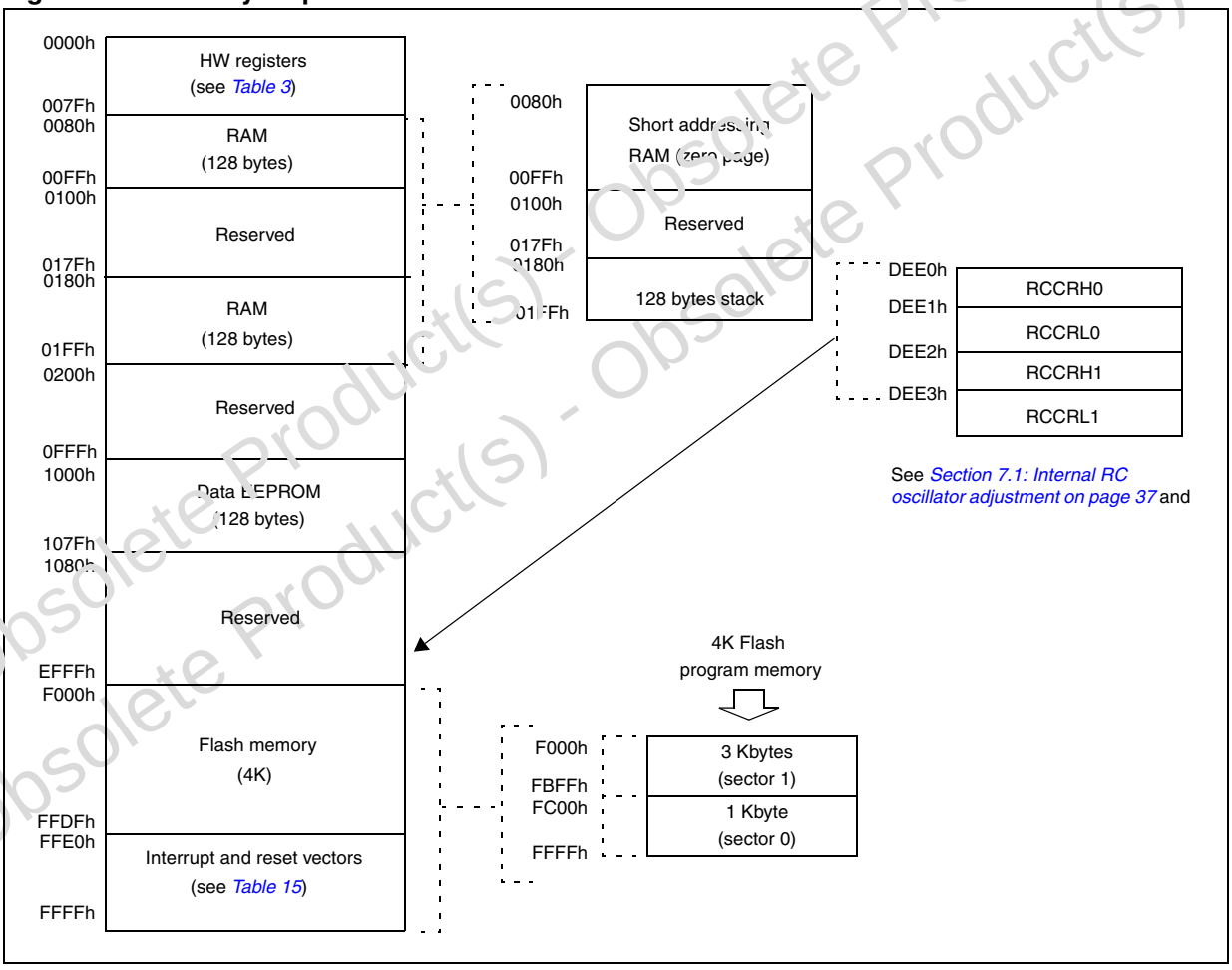
The highest address bytes contain the user reset and interrupt vectors.

The Flash memory contains two sectors (see *Figure 3*) mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

The size of Flash sector 0 and other device options are configurable by option byte (refer to *Section 15.2: Option bytes on page 179*).

Note: Memory locations marked as 'Reserved' must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 3. Memory map



1. *DEE0h, DEE1h, DEE2h and DEE3h addresses are located in a reserved area but are special bytes containing also the RC calibration values which are read-accessible only in user mode. If all the EEPROM data or Flash space (including the RC calibration values locations) has been erased (after the readout protection removal), then the RC calibration values can still be obtained through these four addresses.*

Table 3. Hardware register map(1)

Address	Block	Register label	Register name	Reset status	Remarks
0000h	Port A	PADR	Port A data register	FFh(2)	R/W
0001h		PADDR	Port A data direction register	00h	R/W
0002h		PAOR	Port A option register	40h	R/W
0003h	Port B	PBDR	Port B data register	FFh(2)	R/W
0004h		PBDDR	Port B data direction register	00h	R/W
0005h		PBOR	Port B option register	00h	R/W(3)
0006h	Port C	PCDR	Port C data register	0xh	R/W
0007h		PCDDR	Port C data direction register	00h	R/W
0008h	Lite timer 2	LTCSR2	Lite timer control/status register 2	00h	R/W
0009h		LTARR	Lite timer autoreload register	00h	R/W
000Ah		LTCNTR	Lite timer counter register	00h	Read only
000Bh		LTCSR1	Lite timer control/status register 1	0x00 0000b	R/W
000Ch		LTICR	Lite timer input capture register	xxh	Read only
000Dh	Autoreload timer 4	ATCSR	Timer control/status register	0x00 0000b	R/W
000Eh		CNTR1H	Counter register 1 high	00h	Read only
000Fh		CNTR1L	Counter register 1 low	00h	Read only
0010h		ATRH	Autoreload register high	00h	R/W
0011h		ATRL	Autoreload register low	00h	R/W
0012h		PWMCR	PWM output control register	00h	R/W
0013h		PWM0CSR	PWM 0 control/status register	00h	R/W
0014h		PWM1CSR	PWM 1 control/status register	00h	R/W
0015h		PWM2CSR	PWM 2 control/status register	00h	R/W
0016h		PWM3CSR	PWM 3 control/status register	00h	R/W
0017h		DCR0H	PWM 0 duty cycle register high	00h	R/W
0018h		DCR0L	PWM 0 duty cycle register low	00h	R/W
0019h		DCR1H	PWM 1 duty cycle register high	00h	R/W
001Ah		DCR1L	PWM 1 duty cycle register low	00h	R/W
001Bh		DCR2H	PWM 2 duty cycle register high	00h	R/W
001Ch		DCR2L	PWM 2 duty cycle register low	00h	R/W
001Dh		DCR3H	PWM 3 duty cycle register high	00h	R/W
001Eh		DCR3L	PWM 3 duty cycle register low	00h	R/W
001Fh		ATICRH	Input capture register high	00h	Read only
0020h		ATICRL	Input capture register low	00h	Read only
0021h		ATCSR2	Timer control/status register 2	03h	R/W
0022h		BREAKCR	Break control register	00h	R/W
0023h		ATR2H	Autoreload register 2 high	00h	R/W
0024h		ATR2L	Autoreload register 2 low	00h	R/W
0025h		DTGR	Dead time generation register	00h	R/W
0026h		BREAKEN	Break enable register	03h	R/W
0027h to 002Dh	Reserved area (7 bytes)				
002Eh	WDG	WDGCR	Watchdog control register	7Fh	R/W
0002Fh	FLASH	FCSR	Flash control/status register	00h	R/W
00030h	EEPROM	EECSR	Data EEPROM control/status register	00h	R/W
0031h	SPI	SPIDR	SPI data I/O register	xxh	R/W
0032h		SPICR	SPI control register	0xh	R/W
0033h		SPICSR	SPI control status register	00h	R/W

Table 3. Hardware register map(1)

Address	Block	Register label	Register name	Reset status	Remarks
0034h 0035h 0036h	ADC	ADCCSR ADCDRH ADCDRL	A/D control status register A/D data register high Data low register	00h xxh 0xh	R/W Read only R/W
0037h	ITC	EICR	External interrupt control register	00h	R/W
0038h	MCC	MCCSR	Main clock control/status register	00h	R/W
0039h 003Ah	Clock and reset	RCCR SICSR	RC oscillator control register System integrity control/status register	FFh 0110 0xx0b	R/W R/W
003Bh	PLL clock select	PLLTST	PLL test register	00h	R/W
003Ch	ITC	EISR	External interrupt selection register	00h	R/W
003Dh to 0048h	Reserved area (12 bytes)				
0049h 004Ah	AWU	AWUPR AWUCSR	AWU prescaler register AWU control/status register	FFh 00h	R/W R/W
004Bh 004Ch 004Dh 004Eh 004Fh 0050h 0051h	DM(4)	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L DMCR2	DM control register DM status register DM breakpoint register 1 high DM breakpoint register 1 low DM breakpoint register 2 high DM breakpoint register 2 low DM control register 2	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W
0052h to 007Fh	Reserved area (46 bytes)				

1. Legend: x = undefined, R/W = read/write
2. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents
3. The bits associated with unavailable pins must always keep their reset value
4. For a description of the debug module registers, see *ST7 ICC protocol reference manual*

4 Flash program memory

4.1 Introduction

The ST7 single voltage extended Flash (XFlash) is a non-volatile memory that can be electrically erased and programmed either on a byte-by-byte basis or up to 32 bytes in parallel.

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using in-circuit programming (ICP) or in-application programming (IAP).

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

4.2 Main features

- In-circuit programming (ICP)
- In-application programming (IAP)
- In-circuit testing (ICT) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Readout and write protection

4.3 Programming modes

The ST7 can be programmed in three different ways:

– Insertion in a programming tool. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased.

– In-circuit programming. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased without removing the device from the application board.

– In-application programming. In this mode, sector 1 and data EEPROM (if present) can be programmed or erased without removing the device from the application board and while the application is running.

4.3.1 In-circuit programming (ICP)

ICP uses a protocol called ICC (in-circuit communication) which allows an ST7 plugged on a printed circuit board (PCB) to communicate with an external programming device connected via a cable. ICP is performed in three steps:

– Switch the ST7 to ICC mode (in-circuit communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the $\overline{\text{RESET}}$ pin is pulled low. When the ST7 enters ICC mode, it fetches a specific reset vector which points to the ST7 system memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

– Download ICP driver code in RAM from the ICCDATA pin

– Execute ICP driver code in RAM to program the Flash memory

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

4.3.2 In-application programming (IAP)

This mode uses an IAP driver program previously programmed in sector 0 by the user (in ICP mode).

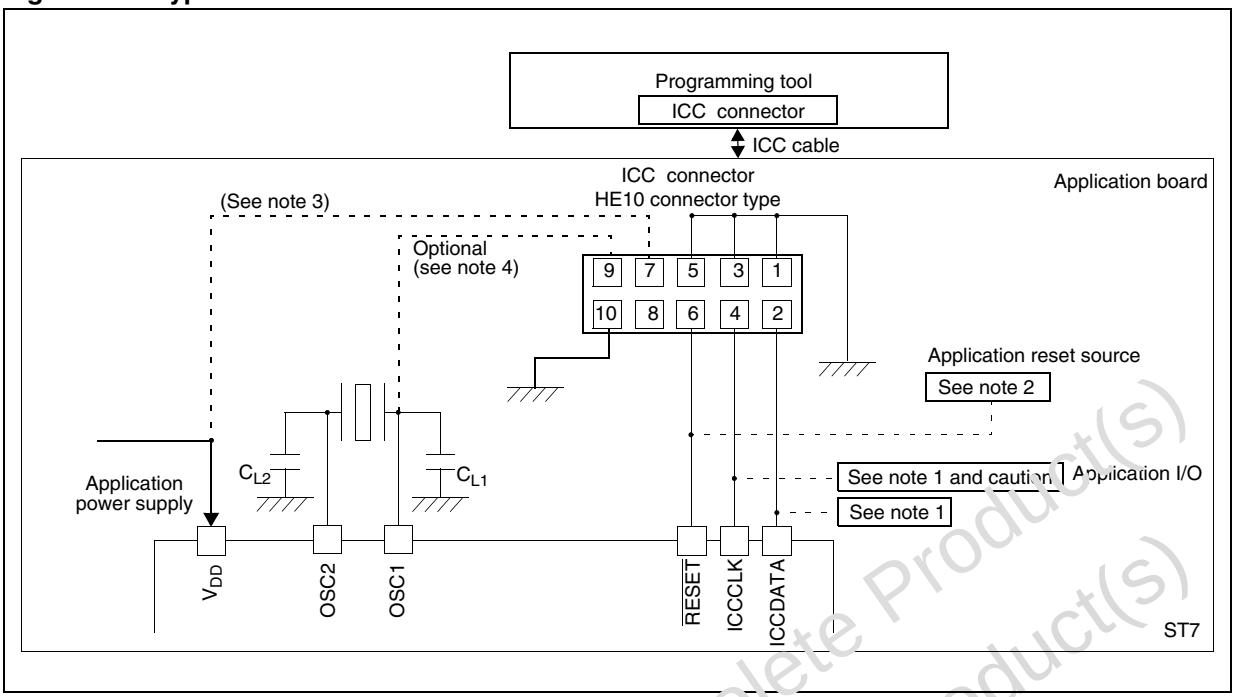
IAP mode is fully controlled by user software, allowing it to be adapted to the user application (such as a user-defined strategy for entering programming mode or a choice of communications protocol used to fetch the data to be stored). This mode can be used to program any memory areas except sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

4.4 ICC interface

ICP needs a minimum of four and up to six pins to be connected to the programming tool. These pins are:

- $\overline{\text{RESET}}$: Device reset
- V_{SS}: Device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- OSC1: Main clock input for external source (not required on devices without OSC1/OSC2 pins)
- V_{DD}: Application board power supply (optional, see note 3, *Figure 4: Typical ICC interface on page 24*)

Figure 4. Typical ICC interface



1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor must be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.
2. During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push-pull output or pull-up resistor < 1K). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R > 1K or a reset management IC with open drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 must be connected to the OSC1 pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. On ST7 devices with multi-oscillator capability, OSC2 must be grounded in this case.
5. In 38-pulse ICC mode, the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte. For ST7L1 devices which do not support the internal RC oscillator, the 'option byte disabled' mode must be used (35-pulse ICC mode entry, clock provided by the tool).

Caution: During normal operation the ICCCLK pin must be pulled up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset puts it back in input pull-up.

4.5 Memory protection

There are two different types of memory protection: Readout protection and write/erase protection, which can be applied individually.

4.5.1 Readout protection

Readout protection, when selected, protects against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data EE memory are protected.

In Flash devices, this protection is removed by reprogramming the option. In this case, both program and data EE memory are automatically erased and the device can be reprogrammed.

Readout protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by the mask option specified in the option list.

4.5.2 Flash write/erase protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to EE data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

Warning: Once set, write/erase protection can never be removed. A write-protected Flash device is no longer reprogrammable.

Write/erase protection is enabled through the FMP_W bit in the option byte.

4.6 Related documentation

For details on Flash programming and ICC protocol, refer to the *ST7 Flash programming reference manual* and to the *ST7 ICC protocol reference manual*.

4.7 Register description

Flash control/status register (FCSR)

FCSR

Reset value: 0000 0000 (00h)
1st RASS key: 0101 0110 (56)
2nd RASS key: 10101110 (AEh)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	OPT	LAT	PGM
-	-	-	-	-	R/W	R/W	R/W

Note: This register is reserved for programming using ICP, IAP or other programming methods. It controls the XFlash programming and erasing operations.

When an EPB or another programming tool is used (in socket or ICP mode), the RASS keys are sent automatically.

5 Data EEPROM

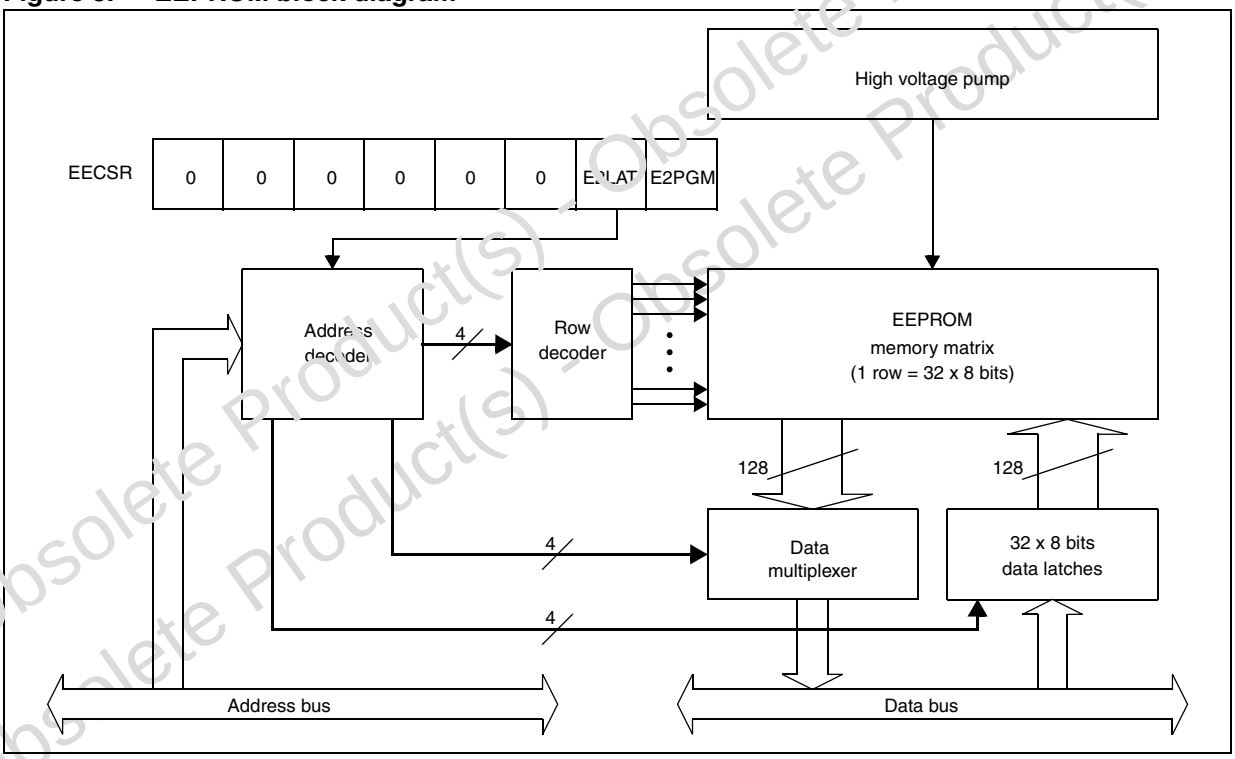
5.1 Introduction

The electrically erasable programmable read only memory can be used as a non volatile back-up for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

5.2 Main features

- Up to 32 bytes programmed in the same cycle
- EEPROM mono-voltage (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- WAIT mode management
- Readout protection

Figure 5. EEPROM block diagram



5.3 Memory access

The data EEPROM memory read/write access modes are controlled by the E2LAT bit of the EEPROM control/status register (EECSR). The flowchart in *Figure 6: Data EEPROM programming flowchart on page 29* describes these different memory access modes.

Read operation (E2LAT = 0)

The EEPROM can be read as a normal ROM location when the E2LAT bit of the EECSR register is cleared.

On this device, data EEPROM can also be used to execute machine code. Do not write to the data EEPROM while executing from it. This would result in an unexpected code being executed.

Write operation (E2LAT = 1)

To access the write mode, the E2LAT bit must be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

When PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must ensure that all the bytes written between two programming sequences have the same high address: Only the five least significant bits of the address can change.

At the end of the programming cycle, the PGM and LAT bits are cleared simultaneously.

Note: Care should be taken during the programming cycle. Writing to the same memory location over-programs the memory (logical AND between the two write access data results) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit. It is not possible to read the latched data. This note is illustrated by the Figure 8: Data EEPROM programming cycle on page 31.

Figure 6. Data EEPROM programming flowchart

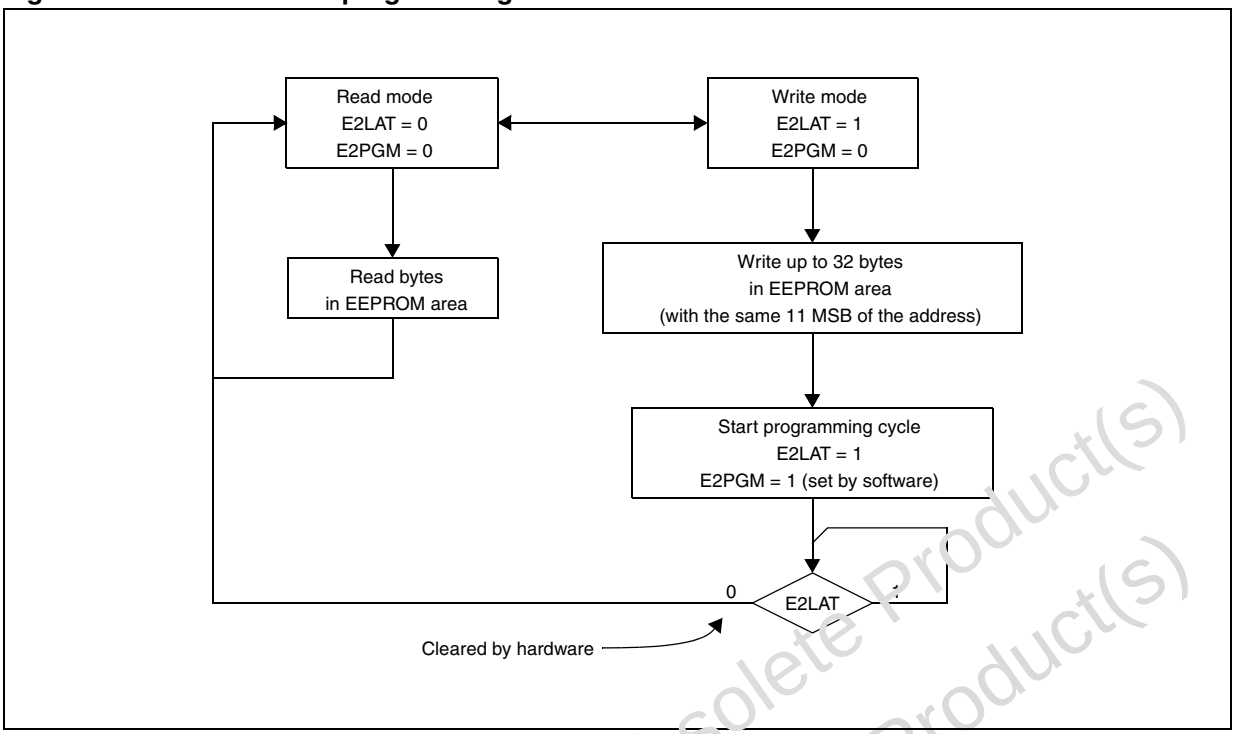
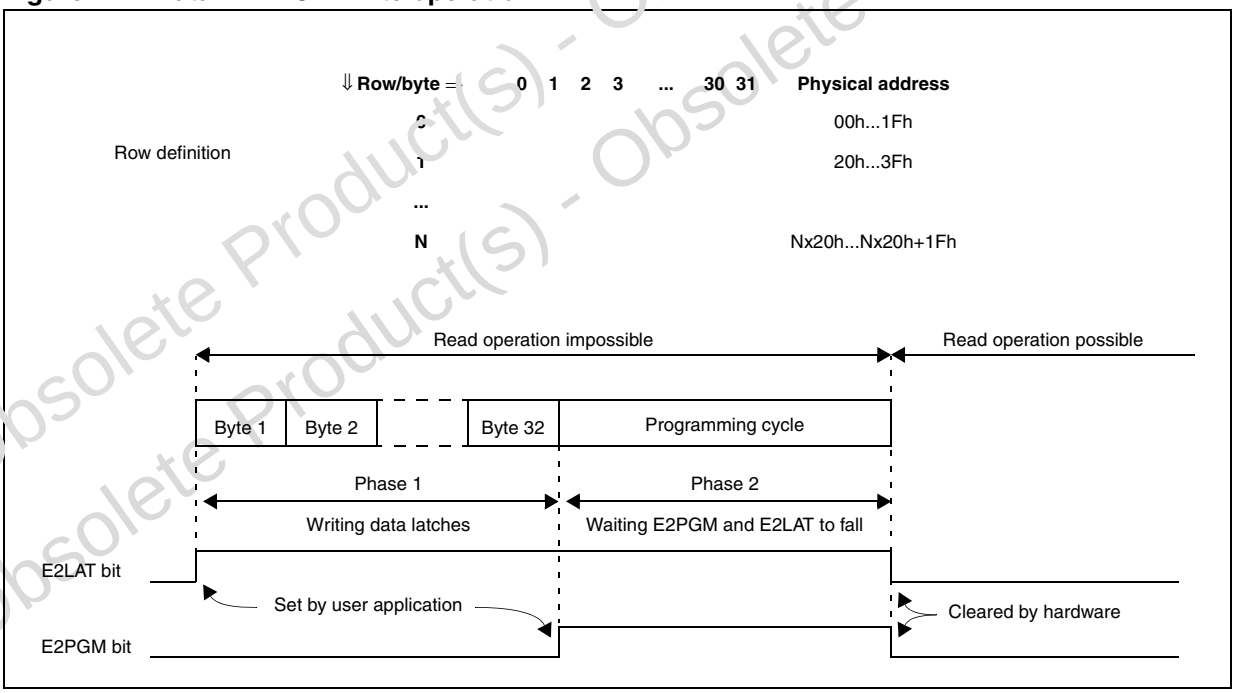


Figure 7. Data EEPROM write operation



1. If a programming cycle is interrupted (by a reset action), the integrity of the data in memory is not guaranteed.

5.4 Power saving modes

Wait mode

The data EEPROM can enter wait mode on execution of the WFI instruction of the microcontroller or when the microcontroller enters active halt mode.The data EEPROM immediately enters this mode if there is no programming in progress, otherwise the data EEPROM finishes the cycle and then enters wait mode.

Active halt mode

Refer to wait mode.

Halt mode

The data EEPROM immediately enters halt mode if the microcontroller executes the HALT instruction. Therefore, the EEPROM stops the function in progress, and data may be corrupted.

5.5 Access error handling

If a read access occurs while E2LAT = 1, then the data bus is not driven.

If a write access occurs while E2LAT = 0, then the data on the bus is not latched.

If a programming cycle is interrupted (by reset action), the integrity of the data in memory is not guaranteed.

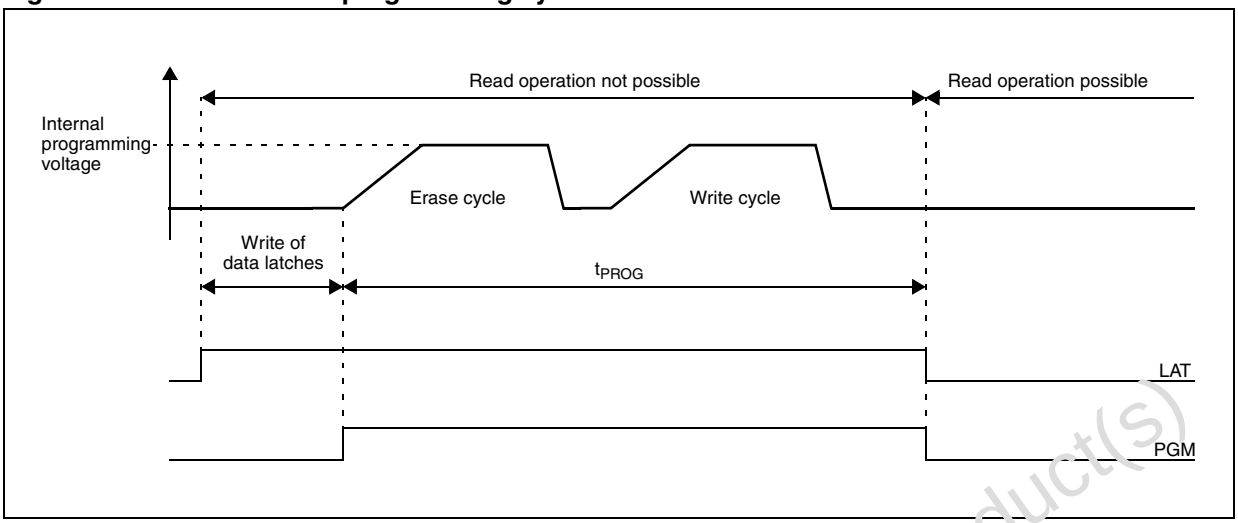
5.6 Data EEPROM readout protection

The readout protection is enabled through an option bit (see *Section 15.2: Option bytes on page 179*).

When this option is selected, the programs and data stored in the EEPROM memory are protected against readout (including a rewrite protection). In Flash devices, when this protection is removed by reprogramming the option byte, the entire program memory and EEPROM is first automatically erased.

Note: Both program memory and data EEPROM are protected using the same option bit.

Figure 8. Data EEPROM programming cycle



5.7 Register description

EEPROM control/status regiSTER (EECSR)

EECSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	E2LAT	E2PGM
-	-	-	-	-	-	R/W	R/W

Table 4. CC register description

Bit	Bit name	Function
7:2	-	Reserved, forced by hardware to 0
1	E2LAT	Latch access transfer This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if the E2PGM bit is cleared. 0: Read mode 1: Write mode
0	E2PGMC	Programming control and status This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware. 0: Programming finished or not yet started 1: Programming cycle is in progress *Note: If the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed*

Table 5. Data EEPROM register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0030h	EECSR Reset value	0	0	0	0	0	0	E2LAT 0	E2PGM 0

6 Central processing unit

6.1 Introduction

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

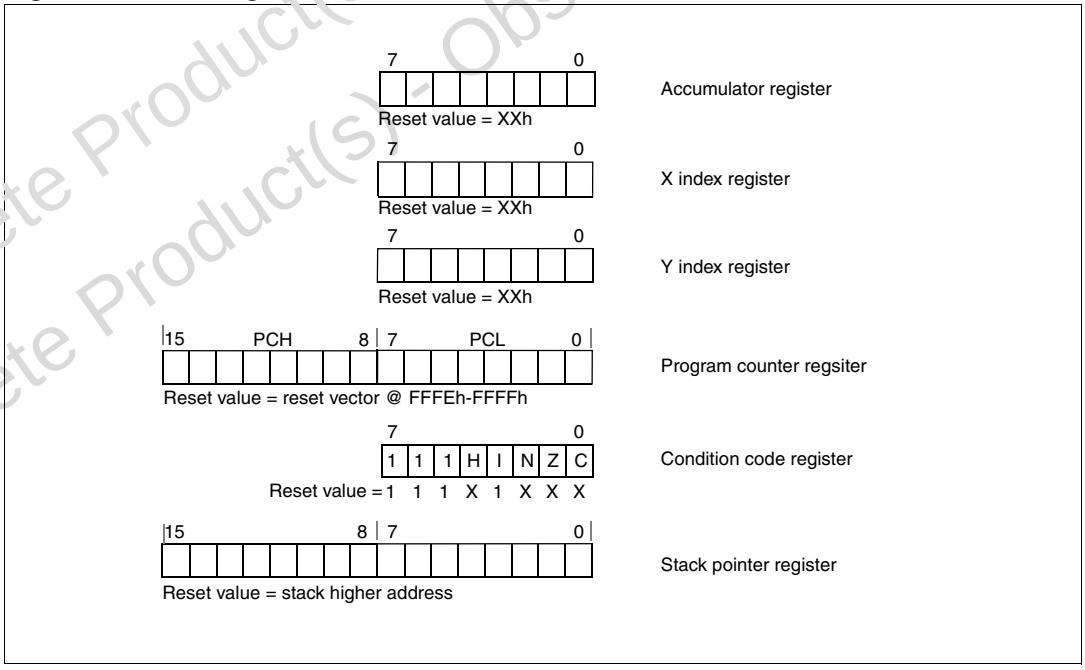
6.2 Main features

- 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

6.3 CPU registers

The six CPU registers shown in *Figure 9* are not present in the memory mapping and are accessed by specific instructions.

Figure 9. CPU registers



1. X = undefined value

Accumulator register (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 register (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).

Condition code register (CC)

CC Reset value: 111x 1xxx

7	6	5	4	3	2	1	0
1			H	I	N	Z	C
	R/W		R/W	R/W	R/W	R/W	R/W

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

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

Table 6. CC register description

Bit	Bit name	Function
4	H	Half carry This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions. 0: No half carry has occurred 1: A half carry has occurred This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Table 6. CC register description (continued)

Bit	Bit name	Function
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 interruptible 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.*
2	N	Negative This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result 7^{th} bit. 0: The result of the last operation is positive or null 1: The result of the last operation is negative (in other words, the most significant bit is a logic 1) This bit is accessed by the JRMI and JRPL test instructions.
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.
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.

Stack pointer register (SP)

SP — Reset value: 01 FFh

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	1
-	-	-	-	-	-	-	R/W

7	6	5	4	3	2	1	0
1	SP[6:0]						
R/W	R/W						

The stack pointer is a 16-bit register which always points 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 10: Stack manipulation example on page 36*).

Since the stack is 128 bytes deep, the 9 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 SP6 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.

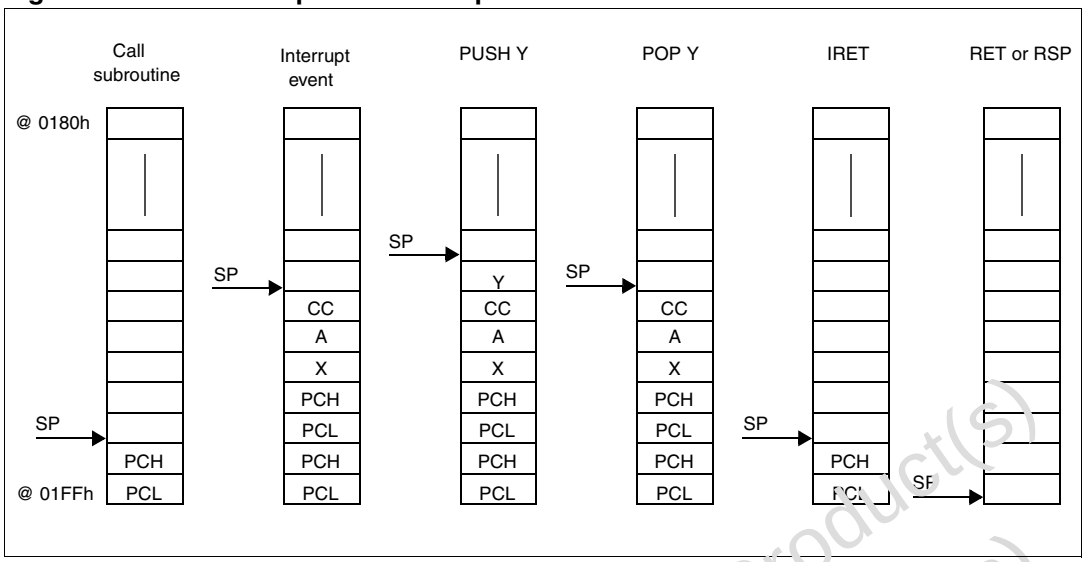
Note: When the lower limit is exceeded, the stack pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

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 10: Stack manipulation example on page 36*.

- 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 10. Stack manipulation example



1. Legend: stack higher address = 01FFh; stack lower address = 0100h

7 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.

Main features

- Clock management
  - 1 MHz internal RC oscillator (enabled by option byte)
  - 1 to 16 MHz external crystal/ceramic resonator
  - External clock input (enabled by option byte)
  - PLL for multiplying the frequency by 8 or 4 (enabled by option byte). Only multiplying by 8 is available for ROM devices.
- Reset sequence manager (RSM)
- System integrity management (SI)
  - Main supply low voltage detection (LVD) with reset generation (enabled by option byte)

7.1 Internal RC oscillator adjustment

The device contains an internal RC oscillator with high accuracy for a given device, temperature and voltage. It must be calibrated to obtain the frequency required in the application. This is done by the software writing a 10-bit calibration value in the RCCR (RC control register) and in the bits [6:5] in the SICSR (SI control status register).

Whenever the microcontroller is reset, the RCCR returns to its default value (FFh), that is, each time the device is reset, the calibration value must be loaded in the RCCR. Predefined calibration values are stored in EEPROM for 3.3 V and 5 V V_{DD} supply voltages at $T_A$max, as shown in *Table 7: RCCR calibration registers on page 38*.

Table 7. RCCR calibration registers

RCCR	Conditions	ST7L1 address
RCCRH0	V_{DD} = 5 V $T_A$max f_{RC} = 1 MHz(1)	DEE0h(2) (CR[9:2])
RCCRL0		DEE1h(2) (CR[1:0])
RCCRH1	V_{DD} = 3.3 V $T_A$max f_{RC} = 1 MHz(1)	DEE2h(2) (CR[9:2])
RCCRL1		DEE3h(2) (CR[1:0])

1. RCCR0 and RCCR1 calibrated within these conditions in order to reach RC accuracy as mentionned in *Table 81: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 4.5 to 5.5 V on page 145* and *Table 83: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 3.0 to 3.6 V on page 147*

2. DEE0h, DEE1h, DEE2h, and DEE3h addresses are located in a reserved area but are special bytes containing also the RC calibration values which are read-accessible only in user mode. If all the EEPROM data or Flash space (including the RC calibration value locations) has been erased (after the readout protection removal), then the RC calibration values can still be obtained through these four addresses. For compatibility reasons with the SICSR register, CR[1:0] bits are stored in the fifth and sixth position of the DEE1 and DEE3 addresses.

Note: 1 In 38-pulse ICC mode, the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte. For ST7L1 devices which do not support the internal RC oscillator, the 'option byte disabled' mode must be used (35-pulse ICC mode entry, clock provided by the tool).

2 For more information on the frequency and accuracy of the RC oscillator see Section 13: Electrical characteristics.

3 To improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.

4 These bytes are systematically programmed by ST, including on FASTROM devices.

Caution: If the voltage or temperature conditions change in the application, the frequency may need to be recalibrated.

Refer to application note AN1324 for information on how to calibrate the RC frequency using an external reference signal.

7.2 Phase locked loop

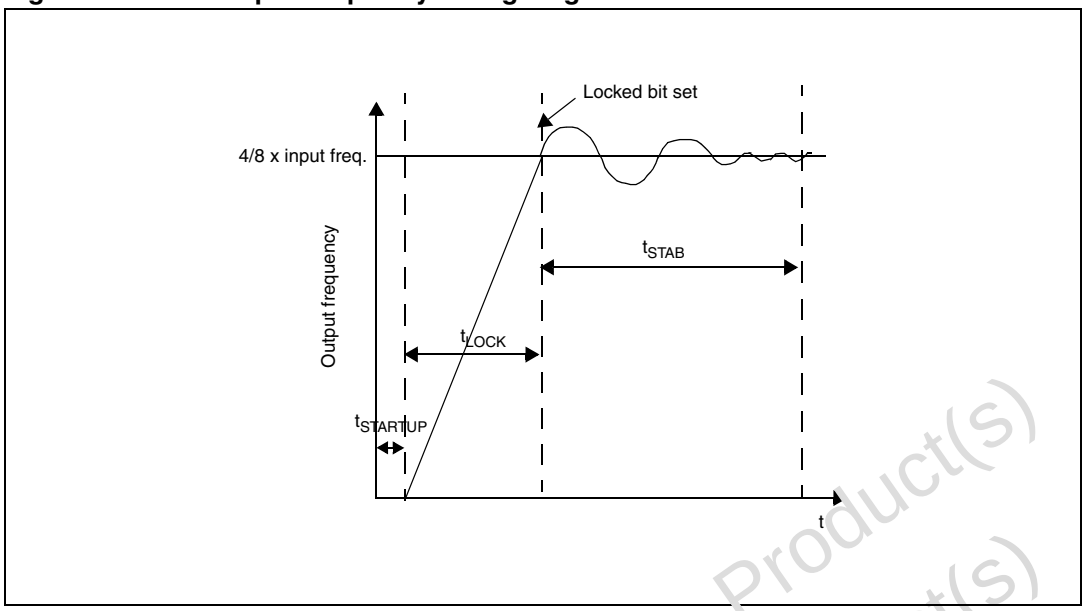
The PLL can be used to multiply a 1 MHz frequency from the RC oscillator or the external clock by 4 or 8 to obtain f_{OSC} of 4 or 8 MHz. The PLL is enabled and the multiplication factor of 4 or 8 is selected by 2 option bits:

- The x4 PLL is intended for operation with V_{DD} in the 3 V to 3.6 V range (available only on Flash devices)
- The x8 PLL is intended for operation with V_{DD} in the 3.6 V to 5.5 V range

Note: It is possible to obtain f_{OSC} = 4 MHz in the 3.3 V to 5.5 V range with internal RC and PLL enabled by selecting 1 MHz RC and x8 PLL and setting the PLLdiv2 bit in the PLLTST register (see Section 7.6.3: Register description on page 48).

Refer to *Section 15.2: Option bytes on page 179* for the option byte description.
If the PLL is disabled and the RC oscillator is enabled, then f_{OSC} = 1 MHz.
If both the RC oscillator and the PLL are disabled, f_{OSC} is driven by the external clock.

Figure 11. PLL output frequency timing diagram



When the PLL is started, after reset or wakeup from halt mode or AWUFH mode, it outputs the clock after a delay of $t_{STARTUP}$.

When the PLL output signal reaches the operating frequency, the locked bit in the SICSCR register is set. Full PLL accuracy (ACC_{PLL}) is reached after a stabilization time of t_{STAB} (see *Figure 11* and *Section 13.3.3: Internal RC oscillator and PLL on page 150*).

Refer to *Section 7.6.3: Register description on page 48* for a description of the locked bit in the SICSR register.

7.3 Register description

Main clock control/status register (MCCSR)

MCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MCO	SMS
-	-	-	-	-	-	R/W	R/W

Table 8. MCCSR register description

Bit	Bit name	Function
7:2	-	Reserved, must be kept cleared
1	MCO	Main clock out enable This bit is read/writen by software and cleared by hardware after a reset. This bit enables the MCO output clock. 0: MCO clock disabled, I/O port free for general purpose I/O 1: MCO clock enabled

Table 8. MCCSR register description (continued)

Bit	Bit name	Function
0	SMS	Slow mode select This bit is read/writen by software and cleared by hardware after a reset. This bit selects the input clock f_{OSC} or $f_{OSC}/32$. 0: Normal mode ($f_{CPU} = f_{OSC}$) 1: Slow mode ($f_{CPU} = f_{OSC}/32$)

RC control register (RCCR)

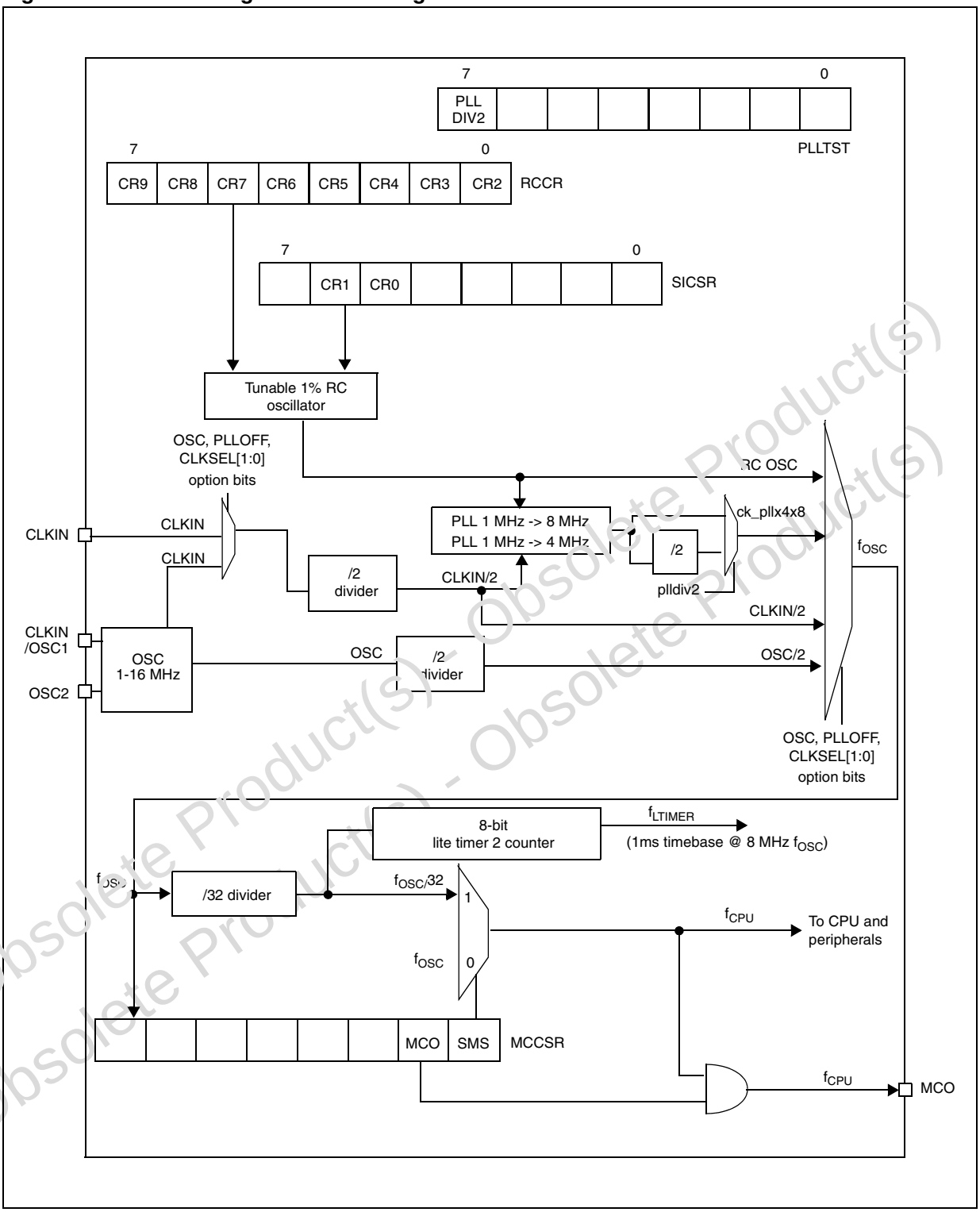
RCCR Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
CR[9:2]							
R/W							

Table 9. RCCR register description

Bit	Bit name	Function
7:0	CR[9:2]	RC oscillator frequency adjustment bits These bits must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. The application can store the correct value for each voltage range in EEPROM and write it to this register at startup. 00h = Maximum available frequency FFh = Lowest available frequency These bits are used with the CR[1:0] bits in the SICSR register. Refer to *Section 7.6.3: Register description on page 48*. *Note: To tune the oscillator, write a series of different values in the register until the correct frequency is reached. The fastest method is to use a dichotomy starting with 80h.*

Figure 12. Clock management block diagram



7.4 Multi-oscillator (MO)

The main clock of the ST7 can be generated by four different source types coming from the multi-oscillator block (1 to 16 MHz):

- An external source
- Crystal or ceramic resonator oscillators
- An internal high frequency RC oscillator

The associated hardware configurations are shown in *Table 10: ST7 clock sources on page 43*. Refer to *Section 13: Electrical characteristics* for more details.

7.4.1 External clock source

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

Note: When the multi-oscillator is not used, PB4 is selected by default as the external clock.

7.4.2 Crystal/ceramic oscillators

In this mode, with a self-controlled gain feature, an oscillator of any frequency from 1 to 16 MHz can be placed on OSC1 and OSC2 pins. This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. In this mode of the multi-oscillator, the resonator and the load capacitors must be placed as close as possible to the oscillator pins to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

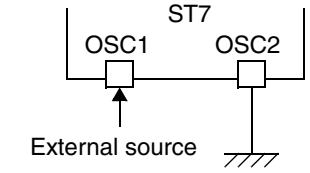
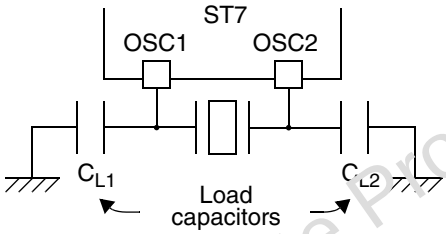
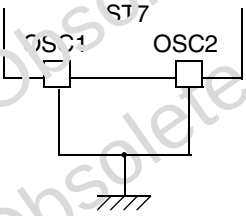
These oscillators are not stopped during the reset phase to avoid losing time in the oscillator startup phase.

7.4.3 Internal RC oscillator

In this mode, the tunable 1% RC oscillator is the main clock source. The two oscillator pins must be tied to ground if dedicated to oscillator use, otherwise they are general purpose I/O.

The calibration is done through the RCCR[7:0] and SICSR[6:5] registers.

Table 10. ST7 clock sources

	Hardware configuration
External clock	
Crystal/ceramic resonators	
Internal RC oscillator	

7.5 Reset sequence manager (RSM)

7.5.1 Introduction

The reset sequence manager includes three reset sources as shown in *Figure 14: Reset block diagram on page 45*:

- External $\overline{\text{RESET}}$ source pulse
- Internal LVD reset (low voltage detection)
- Internal watchdog reset

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 12.2.2: Illegal opcode reset on page 138 for further details.

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

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

The basic reset sequence consists of three phases as shown in *Figure 13*:

- Active phase depending on the reset source
- 256 or 4096 CPU clock cycle delay (see *Table 11*)
- Reset vector fetch

Caution: When the ST7 is unprogrammed or fully erased, the Flash is blank and the reset vector is not programmed. For this reason, it is recommended to keep the reset pin in low state until programming mode is entered, in order to avoid unwanted behavior.

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilize and ensures that recovery has taken place from the reset state. The shorter or longer clock cycle delay is automatically selected depending on the clock source chosen by option byte:

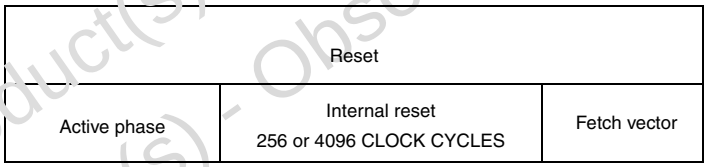
The reset vector fetch phase duration is two clock cycles.

Table 11. Clock cycle delays

Clock source	CPU clock cycle delay
Internal RC oscillator	256
External clock (connected to CLKIN pin)	256
External crystal/ceramic oscillator (connected to OSC1/OSC2 pins)	4096

If the PLL is enabled by option byte, it outputs the clock after an additional delay of $t_{STARTUP}$ (see *Figure 11: PLL output frequency timing diagram on page 39*).

Figure 13. Reset sequence phases

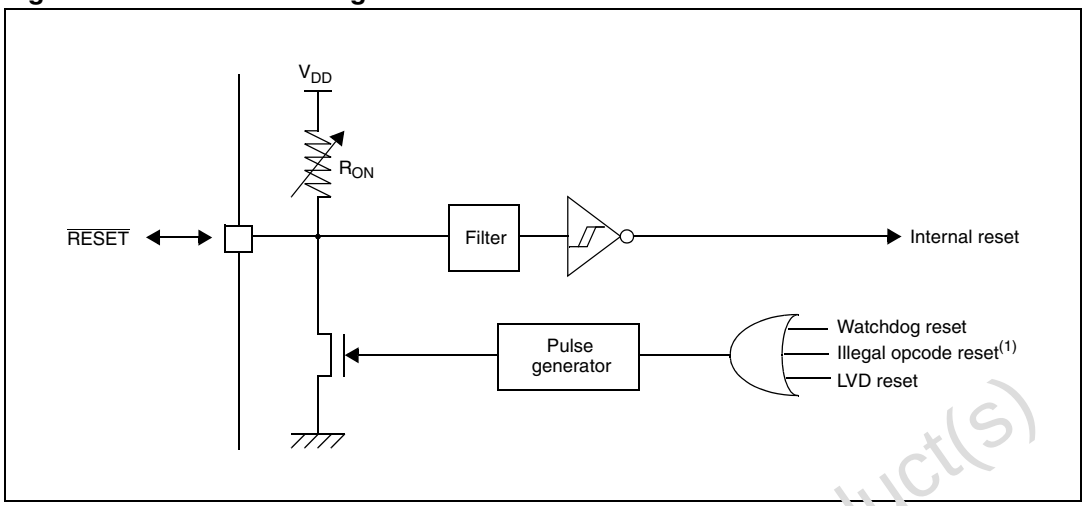


7.5.2 Asynchronous external $\overline{\text{RESET}}$ pin

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 *Section 13: Electrical characteristics* 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 (see *Figure 15: Reset sequences on page 46*). This detection is asynchronous and therefore the MCU can enter the reset state even in halt mode.

Figure 14. Reset block diagram



1. See *Section 12.2.2: Illegal opcode reset on page 138* for more details on illegal opcode reset conditions

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 *Section 13: Electrical characteristics*.

7.5.3 External power-on reset

If the LVD is disabled by the option byte, to start up the microcontroller correctly, the user must use an external reset circuit to ensure that the reset signal is held low until V_{DD} is over the minimum level specified for the selected f_{OSC} frequency.

A proper reset signal for a slow rising V_{DD} supply can generally be provided by an external RC network connected to the $\overline{\text{RESET}}$ pin.

7.5.4 Internal low voltage detector (LVD) 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 15: Reset sequences on page 46*.

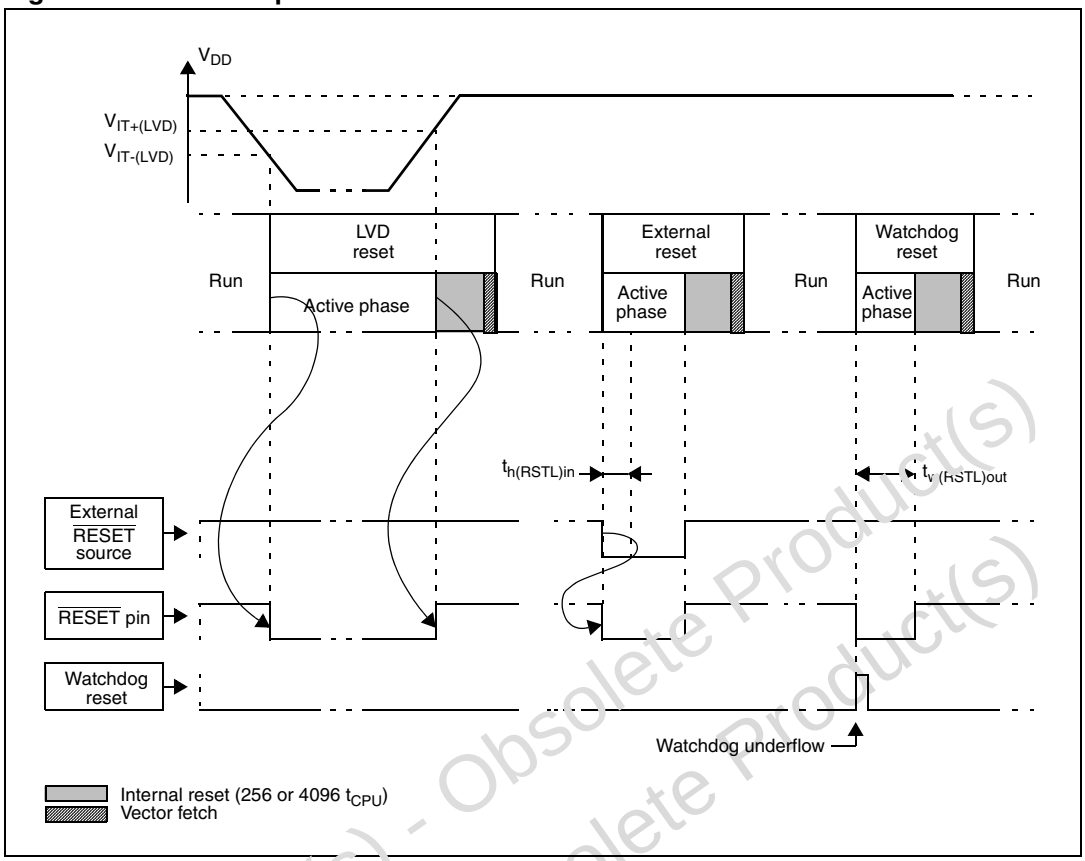
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

7.5.5 Internal watchdog reset

The reset sequence generated by an internal Watchdog counter overflow is shown in *Figure 15: Reset sequences on page 46*.

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 15. Reset sequences



7.6 System integrity management (SI)

The system integrity management block contains the low voltage detector (LVD) function. It is managed by the SICSR register.

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 12.2.2: Illegal opcode reset on page 138 for further details.

7.6.1 Low voltage detector (LVD)

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

The $V_{IT-(LVD)}$ reference value for a voltage drop is lower than the $V_{IT+(LVD)}$ reference value for power-on 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:

- $V_{IT+(LVD)}$ when V_{DD} is rising
- $V_{IT-(LVD)}$ when V_{DD} is falling

The LVD function is illustrated in *Figure 16: Low voltage detector vs reset on page 47*.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above $V_{IT-(LVD)}$, 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.

Note: 1 The LVD allows the device to be used without any external reset circuitry.

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

3 Use of LVD with capacitive power supply: With this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0 V to ensure optimum restart conditions. Refer to the circuit example in Figure 95: RESET pin protection when LVD is disabled on page 170 and Note 4 on the same page.

4 For the application to function correctly, it is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from reset.

Figure 16. Low voltage detector vs reset

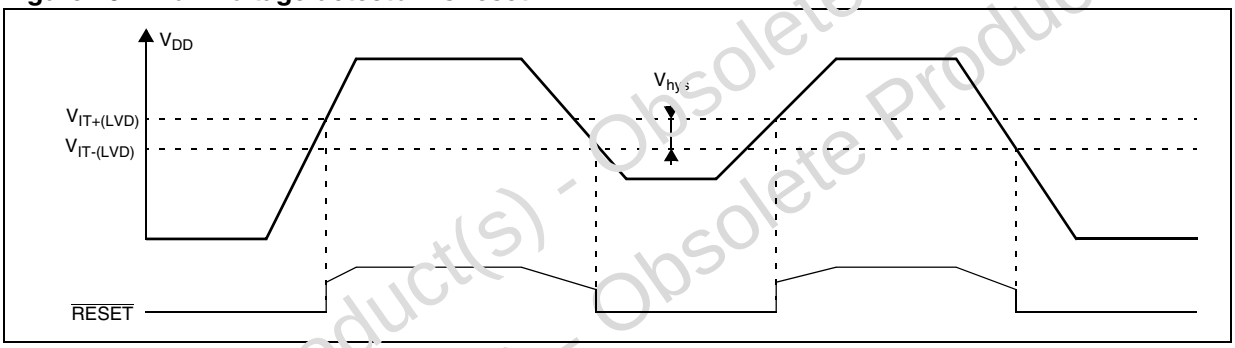
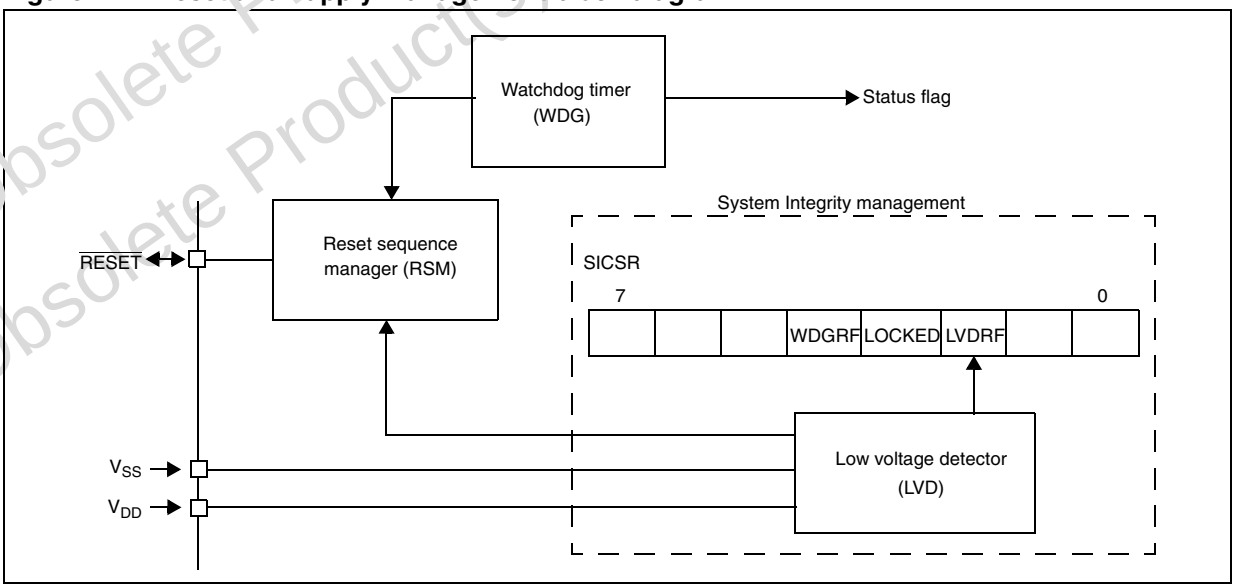


Figure 17. Reset and supply management block diagram



7.6.2 Low power modes

Table 12. Effect of low power modes on system integrity

Mode	Description
Wait	No effect on SI
Halt	The SICSR register is frozen

7.6.3 Register description

System integrity (SI) control/status register (SICSR)

SICSR Reset value: 0110 0x00 (6xh)

7	6	5	4	3	2	1	0
Reserved	CR[1:0]		WDGRF	LOCKED	LVDRF	Reserved	
-	R/W		R/W	R/W	R/W	-	

Table 13. SICSR register description

Bit	Bit name	Function
7	-	Reserved, must be kept cleared
6:5	CR[1:0]	RC oscillator frequency adjustment bits These bits, as well as CR[9:2] bits in the RCCR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. Refer to *Section 7.3: Register description on page 39*
4	WDGRF	Watchdog reset flag This bit indicates that the last reset was generated by the watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD reset (to ensure a stable cleared state of the WDGRF flag when CPU starts). Combined with the LVDRF flag information, the flag description is given as follows: 00 (LVDRF, WDGRF): Reset sources = External $\overline{\text{RESET}}$ pin 01 (LVDRF, WDGRF): Reset sources = Watchdog 1X (LVDRF, WDGRF): Reset sources = LVD
3	LOCKED	PLL locked flag This bit is set and cleared by hardware. It is set automatically when the PLL reaches its operating frequency. 0: PLL not locked 1: PLL locked

Table 13. SICSR register description (continued)

Bit	Bit name	Function
2	LVDRF	LVD reset flag This bit indicates that the last reset was generated by the LVD block. It is set by hardware(LVD reset) and cleared by software (by reading). When the LVD is disabled by option byte, the LVDRF bit value is undefined. *Note: The LVDRF flag is not cleared when another reset type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure. In this case, a watchdog reset can be detected by software while an external reset can not.*
1:0	-	Reserved, must be kept cleared

PLL test register (PLLTST)

PLLTST Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
PLLdiv2	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	-	-	-	-	-	-	-

Table 14. PLLTST register description

Bit	Bit name	Function
7	PLLdiv2	PLL clock divide by 2 This bit is read or written by software and cleared by hardware after reset. This bit divides the PLL output clock by 2. 0: PLL output clock 1: Divide by 2 of PLL output clock Refer to *Figure 12: Clock management block diagram on page 41*. *Note: Writing this bit is effective after two t_{CPU} cycles (if system clock is 8 MHz). Otherwise, writing is effective after one cycle (if system clock is 4 MHz), that is, if the effective time is 250 ns.*
6:0	-	Reserved, must be kept cleared

8 Interrupts

The ST7 core may be interrupted by one of two different methods: Maskable hardware interrupts as listed in *Table 15: Interrupt mapping on page 52* and a non-maskable software interrupt (TRAP). The interrupt processing flowchart is shown in *Figure 18: Interrupt processing flowchart on page 52*.

The maskable interrupts must be enabled by 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).

Note: After reset, all interrupts are disabled.

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 *Table 15: Interrupt mapping* 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 is cleared and the main program resumes.

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 *Table 15: Interrupt mapping*).

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 *Table 15: Interrupt mapping*).

8.1 Non maskable software interrupt

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It is serviced according to the flowchart in *Figure 18: Interrupt processing flowchart on page 52*.

8.2 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.

Caution: The type of sensitivity defined in the miscellaneous or interrupt register (if available) applies to the ei source. In case of a NANDed source (as described in *Section 10: I/O ports*), a low level on an I/O pin, configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

8.3 Peripheral interrupts

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

- I bit of the CC register is cleared
- Corresponding enable bit is set in the control register

If either 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 (that is, waiting for being enabled) will therefore be lost if the clear sequence is executed.

Figure 18. Interrupt processing flowchart

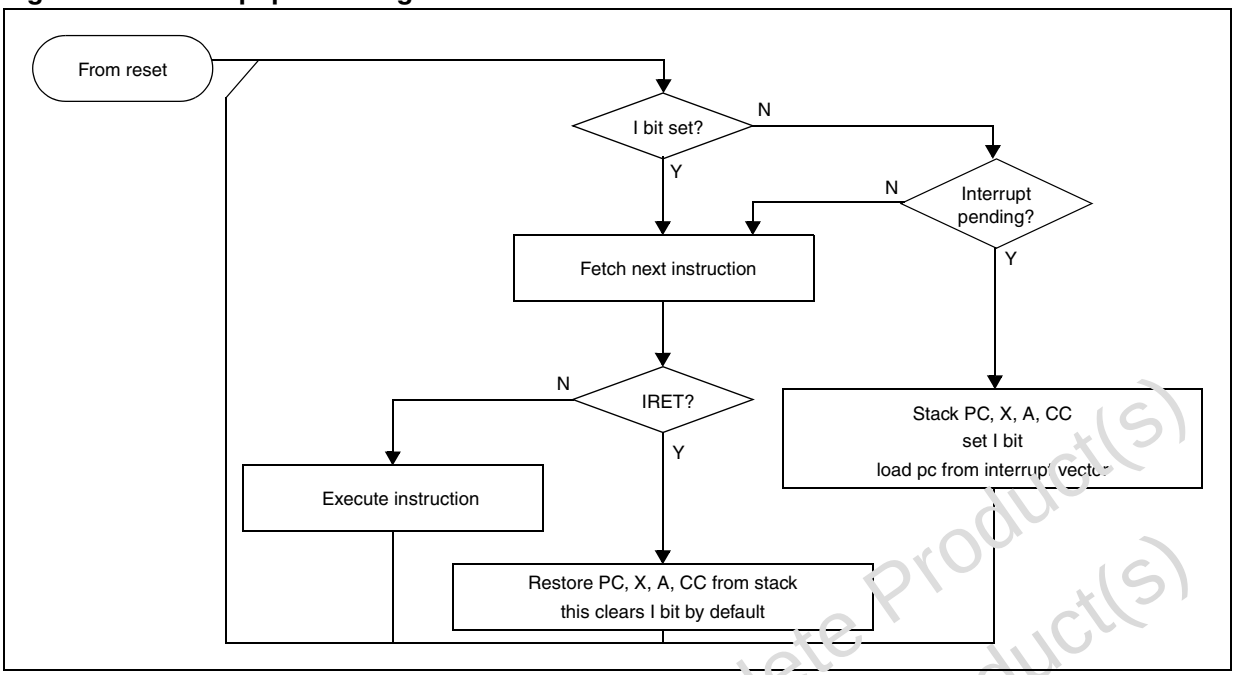


Table 15. Interrupt mapping

No.	Source block	Description	Register label	Priority order	Exit from halt or AWUFH	Address vector
	Reset	Reset	-	Highest priority	Yes	FFFEh-FFFFh
	TRAP	Software interrupt			No	FFFCh-FFFDh
0	AWU	Auto wakeup interrupt	AWUCSR		Yes(1)	FFFAh-FFFBh
1	ei0	External interrupt 0	-		Yes	FFF8h-FFF9h
2	ei1	External interrupt 1				FFF6h-FFF7h
3	ei2	External interrupt 2				FFF4h-FFF5h
4	ei3	External interrupt 3				FFF2h-FFF3h
5	Lite timer	Lite timer RTC2 interrupt	LTCSR2		No	FFF0h-FFF1h
6	Not used					FFEEh-FFEFh
7	Not used					FFECh-FFEDh
8	At timer	At timer output compare Interrupt or input capture interrupt	PWMxCSR or ATCSR		No	FFEAh-FFEBh
9		At timer overflow interrupt	ATCSR		Yes(2)	FFE8h-FFE9h
10	Lite timer	Lite timer input capture interrupt	LTCSR		No	FFE6h-FFE7h
11		Lite timer RTC1 interrupt	LTCSR		Yes(2)	FFE4h-FFE5h
12	SPI	SPI peripheral interrupts	SPICSR		Yes	FFE2h-FFE3h
13	At timer	At timer overflow interrupt	ATCSR2	lowest priority	No	FFE0h-FFE1h

1. This interrupt exits the MCU from ‘auto wakeup from halt’ mode only
2. These interrupts exit the MCU from ‘active halt’ mode only

External interrupt control register (EICR)

EICR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
IS3[1:0]		IS2[1:0]		IS1[1:0]		IS0[1:0]	
R/W		R/W		R/W		R/W	

Table 16. EICR register description

Bit	Bit name	Function
7:6	IS3[1:0]	ei3 sensitivity These bits define the interrupt sensitivity for ei3 (port B0) according to *Table 17*
5:4	IS2[1:0]	ei2 sensitivity These bits define the interrupt sensitivity for ei2 (port B3) according to *Table 17*
3:2	IS1[1:0]	ei1 sensitivity These bits define the interrupt sensitivity for ei1 (port A7) according to *Table 17*
1:0	IS0[1:0]	ei0 sensitivity These bits define the interrupt sensitivity for ei0 (port A0) according to *Table 17*

Note: 1 These 8 bits can be written only when the I bit in the CC register is set.

2 Changing the sensitivity of a particular external interrupt clears this pending interrupt. This can be used to clear unwanted pending interrupts. Refer to the External interrupt function on page 67

Table 17. Interrupt sensitivity bits

ISx1	ISx0	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

External interrupt selection register (EISR)

EISR — Reset value: 0000 1100 (0Ch)

7	6	5	4	3	2	1	0
ei3[1:0]		ei2[1:0]		ei1[1:0]		ei0[1:0]	
R/W		R/W		R/W		R/W	

Table 18. EISR register description

Bit	Bit name	Function
7:6	ei3[1:0]	ei3 pin selection These bits are written by software. They select the port B I/O pin used for the ei3 external interrupt as follows: 00: I/O pin = PB0 (reset state) 01: I/O pin = PB1 10: I/O pin = PB2
5:4	ei2[1:0]	ei2 pin selection These bits are written by software. They select the port B I/O pin used for the ei2 external interrupt as follows: 00: I/O pin = PB3 (reset state) 01: I/O pin = PB4(1) 10: I/O pin = PB5 11: I/O pin = PB6
3:2	ei1[1:0]	ei1 pin selection These bits are written by software. They select the port A I/O pin used for the ei1 external interrupt as follows: 00: I/O pin = PA4 01: I/O pin = PA5 10: I/O pin = PA6 11: I/O pin = PA7 (reset state)
1:0	ei0[1:0]	ei0 pin selection These bits are written by software. They select the port A I/O pin used for the ei0 external interrupt as follows: 00: I/O pin = PA0 (reset state) 01: I/O pin = PA1 10: I/O pin = PA2 11: I/O pin = PA3

1. PB4 cannot be used as an external interrupt in HALT mode

9 Power saving modes

9.1 Introduction

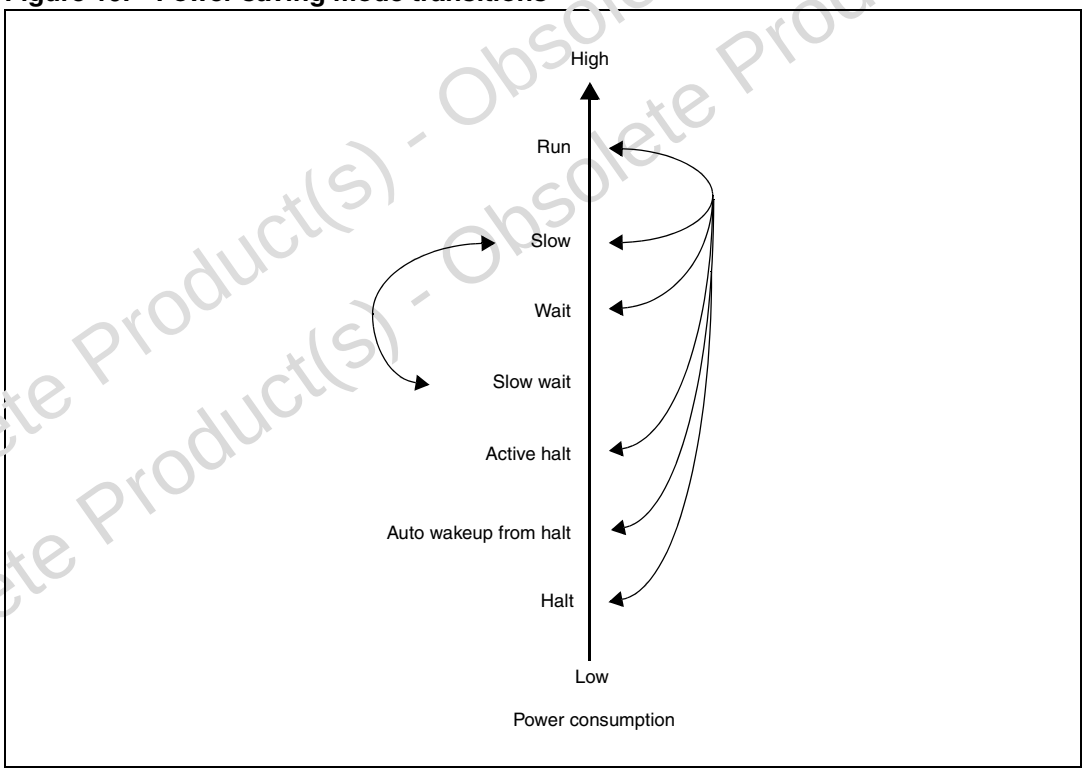
To give a large measure of flexibility to the application in terms of power consumption, five main power saving modes are implemented in the ST7 (see *Figure 19*):

- Slow
- Wait (and slow-wait)
- Active halt
- Auto wakeup from halt (AWUFH)
- Halt

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

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

Figure 19. Power saving mode transitions



9.2 Slow mode

This mode has two targets:

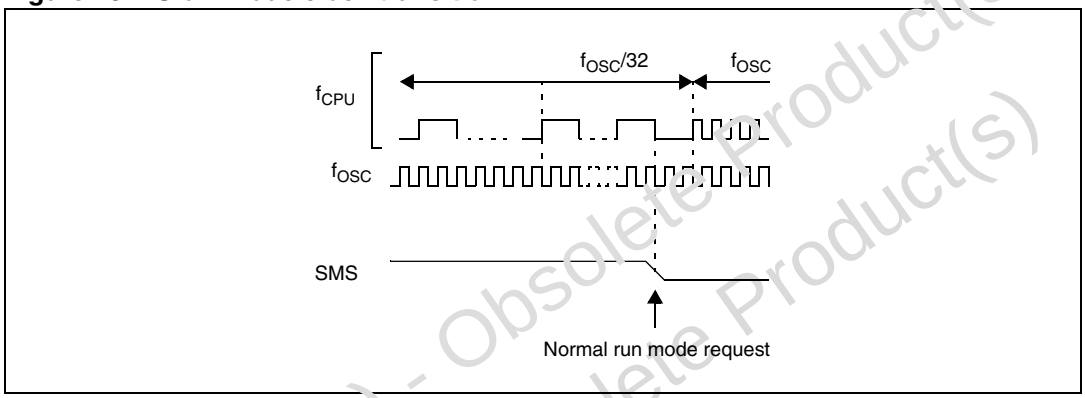
- To reduce power consumption by decreasing the internal clock in the device
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

Slow mode is controlled by the SMS bit in the MCCSR register which enables or disables slow mode.

In this mode, the oscillator frequency is divided by 32. The CPU and peripherals are clocked at this lower frequency.

Note: Slow-wait mode is activated when entering wait mode while the device is already in slow mode.

Figure 20. Slow mode clock transition



9.3 Wait mode

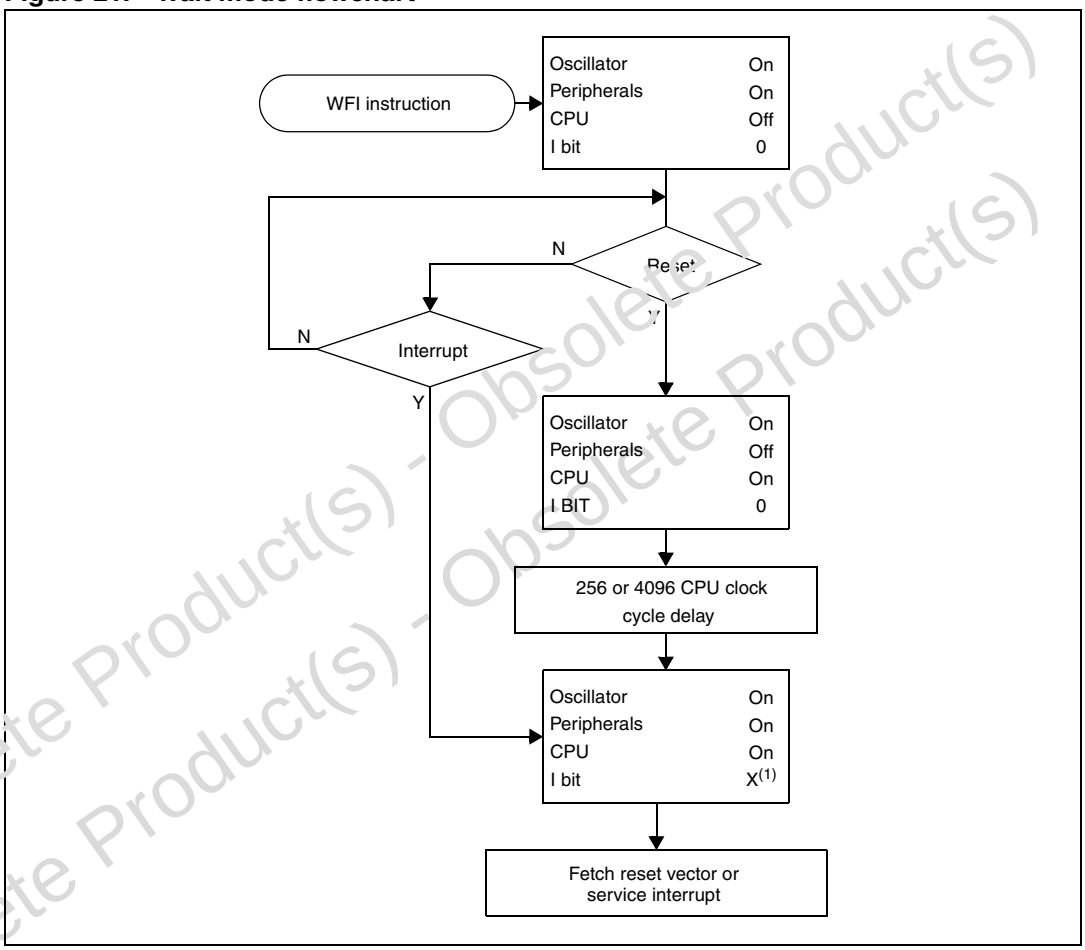
Wait mode places the MCU into a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the 'WFI' instruction.

All peripherals remain active. During wait mode, the I bit of the CC register is cleared to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in wait mode until an interrupt or reset occurs, whereupon it wakes up and the program counter branches to the starting address of the interrupt or reset service routine.

Refer to *Figure 21: Wait mode flowchart on page 57*.

Figure 21. Wait mode flowchart



1. 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.

9.4 Halt mode

The halt mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when active halt is disabled (see *Section 9.5: Active halt mode on page 60* for more details) and when the AWUEN bit in the AWUCSR register is cleared.

The MCU can exit halt mode on reception of either a specific interrupt (see *Table 15: Interrupt mapping on page 52*) or a reset. When exiting halt mode by means of a reset or an interrupt, the oscillator is immediately turned on and the 256 or 4096 CPU cycle delay is used to stabilize the oscillator. After the startup delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see *Figure 23: Halt mode flowchart on page 59*).

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 up immediately.

In halt mode, the main oscillator is turned off, stopping all internal processing, including the operation of the on-chip peripherals. All peripherals are not clocked except those which receive 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 15.2: Option bytes on page 179* for more details).

Figure 22. Halt timing overview

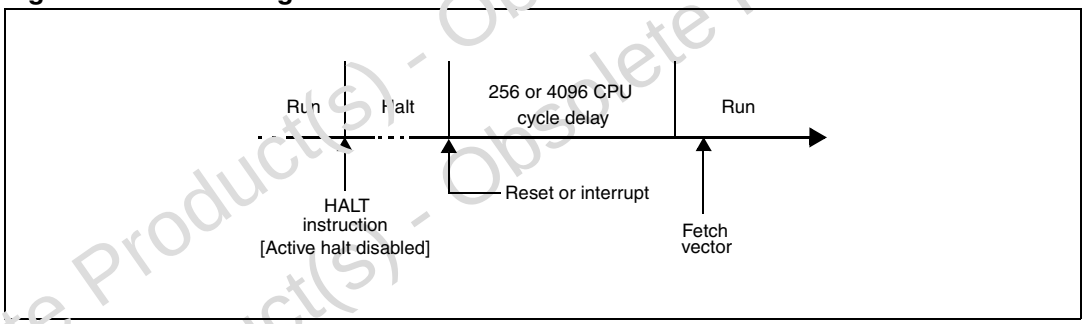
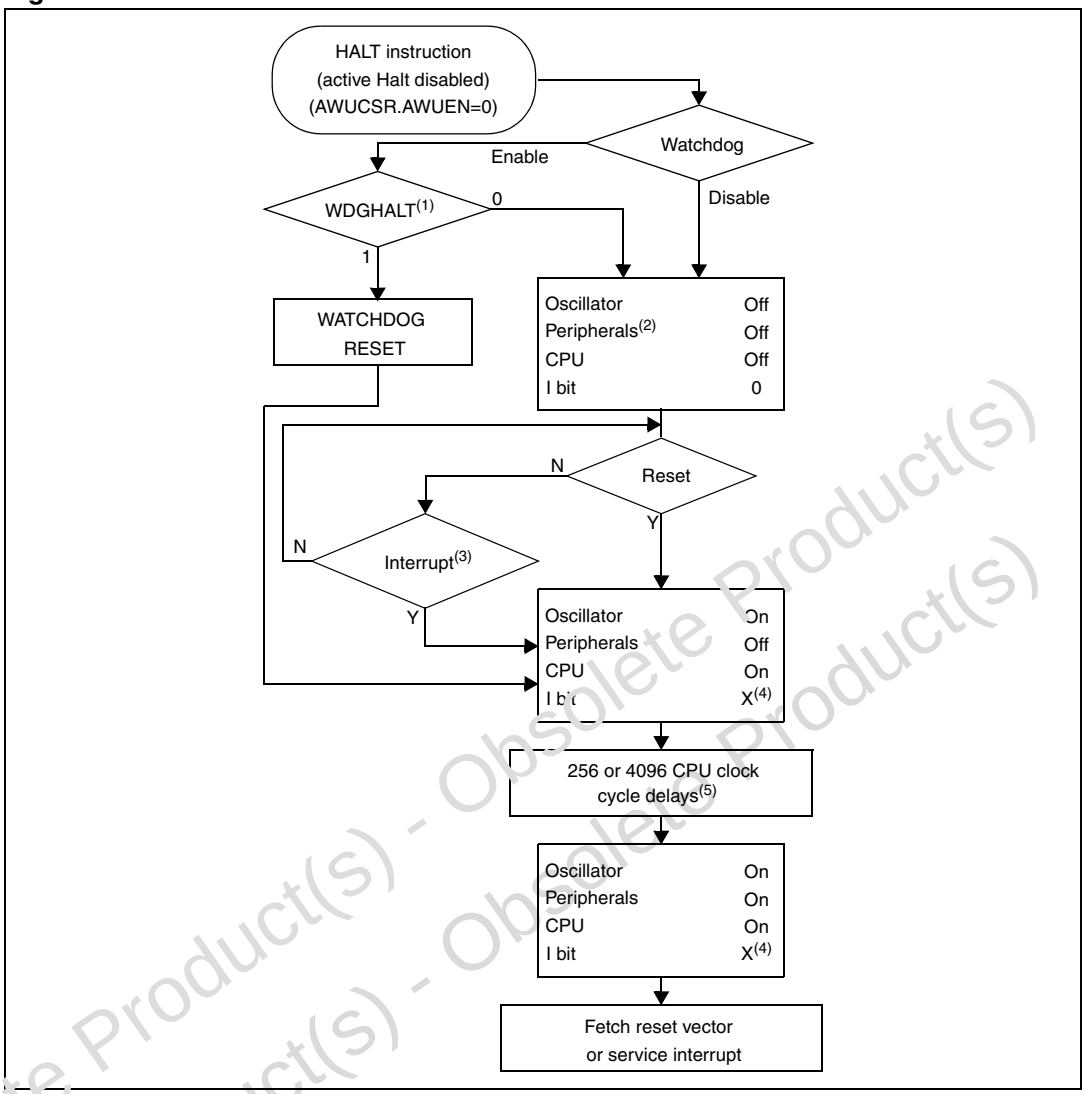


Figure 23. Halt mode flowchart



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 15: Interrupt mapping on page 52* 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.
5. If the PLL is enabled by option byte, it outputs the clock after a delay of $t_{STARTUP}$ (see *Figure 11 on page 39*).

9.4.1 Halt mode recommendations

- Make sure that an external event is available to wake up the microcontroller from halt mode.
- When using an external interrupt to wake up the microcontroller, re-initialize 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 incorrectly configured due to external interference or by an unforeseen logical condition.
- For the same reason, re-initialize 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 program memory with the value 0x8E.
- As the HALT instruction clears the interrupt mask in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wakeup event (reset or external interrupt).

9.5 Active halt mode

Active halt mode is the lowest power consumption mode of the MCU with a real time clock available. It is entered by executing the 'HALT' instruction. The decision to enter either in active halt or halt mode is given by the LTCSR/ATCSR register status as shown in the following table:

Table 19. LTCSR/ATCSR register status

LTCSR1 TB1IE bit	ATCSR OVFIE bit	ATCSRCK1 bit	ATCSRCK0 bit	Meaning
0	x	x	0	Active halt mode disabled
0	0	x	x	
1	x	x	x	Active halt mode enabled
x	1	0	1	

The MCU exits active halt mode on reception of a specific interrupt (see *Table 15: Interrupt mapping on page 52*) or a reset.

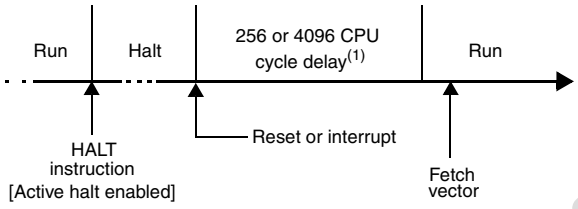
- When exiting active halt mode by means of a reset, a 256 or 4096 CPU cycle delay occurs. After the startup delay, the CPU resumes operation by fetching the reset vector which woke it up (see *Figure 25: Active halt mode flowchart on page 61*).
- When exiting active halt mode by means of an interrupt, the CPU immediately resumes operation by servicing the interrupt vector which woke it up (see *Figure 25: Active halt mode flowchart on page 61*).

When entering active halt mode, the I bit in the CC register is cleared to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately (see *Figure 25*, *Note 2*).

In active halt mode, only the main oscillator and the selected timer counter (LT/AT) are running to keep a wakeup time base. All other peripherals are not clocked except those which receive their clock supply from another clock generator (such as external or auxiliary oscillator).

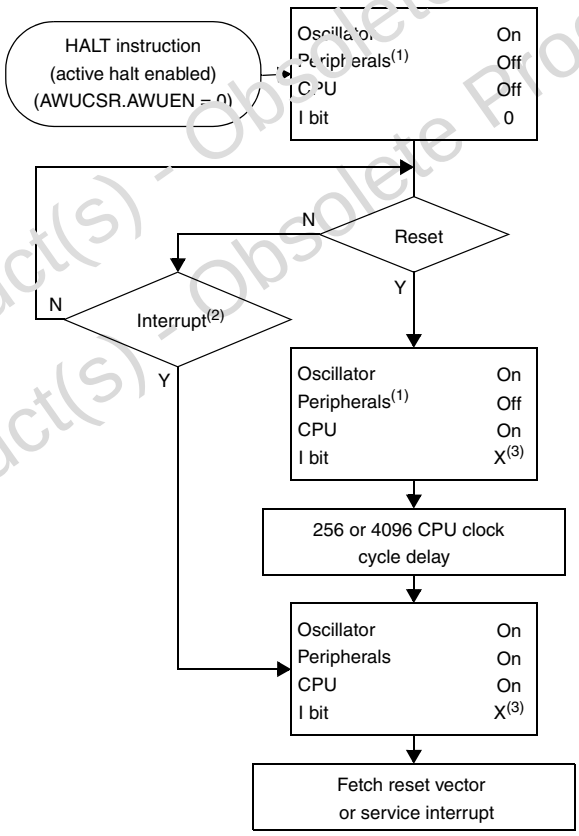
Note: As soon as active halt is enabled, executing a HALT instruction while the watchdog is active does not generate a reset. This means that the device cannot exceed a defined delay in this power saving mode.

Figure 24. Active halt timing overview



1. This delay occurs only if the MCU exits active halt mode by means of a reset.

Figure 25. Active halt mode flowchart



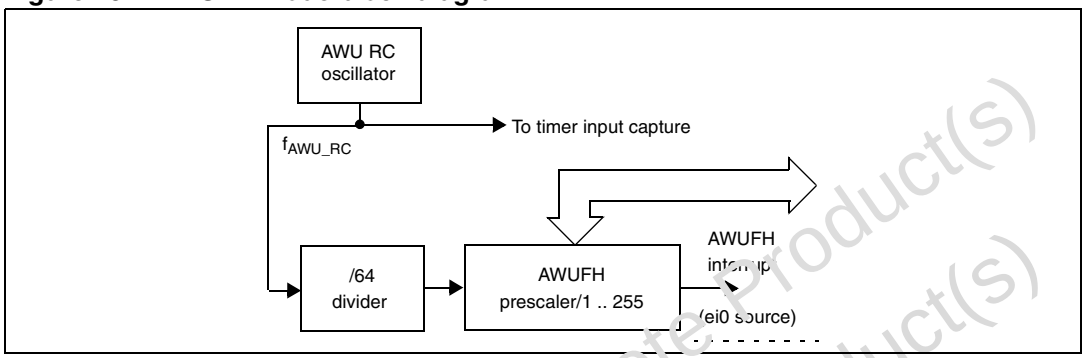
1. Peripherals clocked with an external clock source can still be active.
2. Only the RTC1 interrupt and some specific interrupts can exit the MCU from active halt mode. Refer to *Table 15: Interrupt mapping* for more details.
3. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

9.6 Auto wakeup from halt mode

Auto wakeup from halt (AWUFH) mode is similar to halt mode with the addition of a specific internal RC oscillator for wakeup (auto wakeup from halt oscillator). Compared to active halt mode, AWUFH has lower power consumption (the main clock is not kept running but there is no accurate real-time clock available).

It is entered by executing the HALT instruction when the AWUEN bit in the AWUCSR register has been set.

Figure 26. AWUFH mode block diagram



As soon as halt mode is entered and if the AWUEN bit has been set in the AWUCSR register, the AWU RC oscillator provides a clock signal (f_{AWU_RC}). Its frequency is divided by a fixed divider and a programmable prescaler controlled by the AWUPR register. The output of this prescaler provides the delay time. When the delay has elapsed, the AWUF flag is set by hardware and an interrupt wakes up the MCU from halt mode. At the same time, the main oscillator is immediately turned on and a 256 or 4096 cycle delay is used to stabilize it. After this startup delay, the CPU resumes operation by servicing the AWUFH interrupt. The AWU flag and its associated interrupt are cleared by software reading the AWUCSR register.

To compensate for any frequency dispersion of the AWU RC oscillator, it can be calibrated by measuring the clock frequency f_{AWU_RC} and then calculating the right prescaler value. Measurement mode is enabled by setting the AWUM bit in the AWUCSR register in run mode. This connects f_{AWU_RC} to the input capture of the 12-bit autoreload timer, allowing the f_{AWU_RC} to be measured using the main oscillator clock as a reference timebase.

Similarities with halt mode

The following AWUFH mode behavior is the same as normal halt mode:

- The MCU can exit AWUFH mode by means of any interrupt with exit from halt capability or a reset (see *Section 9.4: Halt mode on page 58*).
- When entering AWUFH mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.
- In AWUFH mode, the main oscillator is turned off, stopping all internal processing, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which receive their clock supply from another clock generator (such as an external or auxiliary oscillator like the AWU oscillator).
- The compatibility of watchdog operation with AWUFH mode is configured by the WDGHALT option bit in the option byte. Depending on this setting, the HALT instruction, when executed while the Watchdog system is enabled, can generate a watchdog reset.

Figure 27. AWUF halt timing diagram

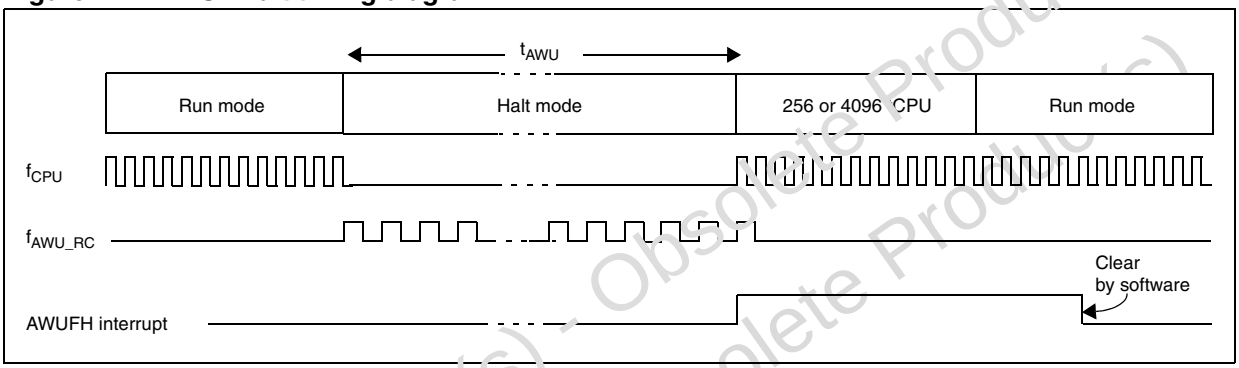
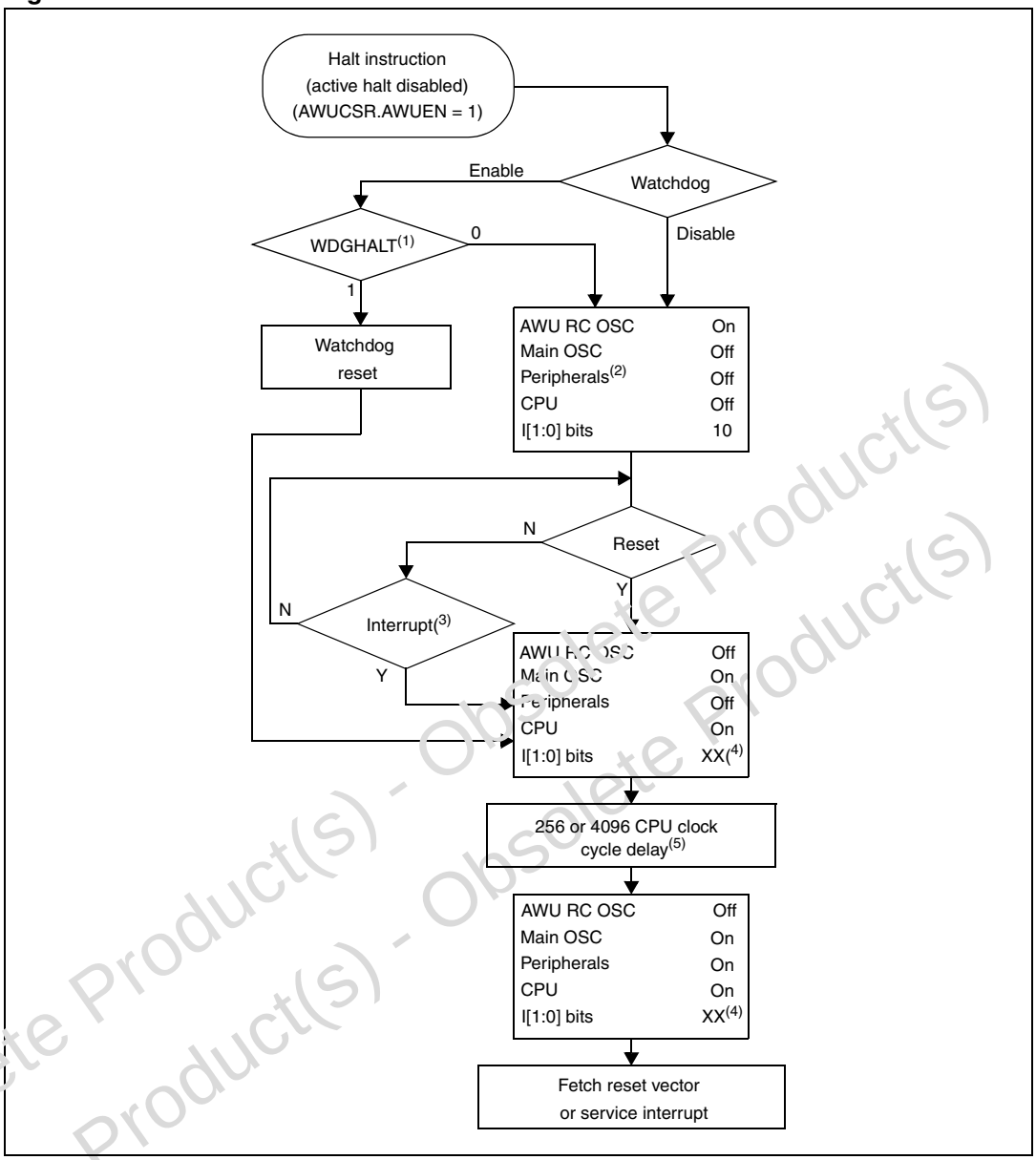


Figure 28. AWUFH mode flowchart



1. WDGHALT is an option bit. See *Section 15.2: Option bytes on page 179* for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only an AWUFH interrupt and some specific interrupts can exit the MCU from halt mode (such as external interrupt). Refer to *Table 15: Interrupt mapping on page 52* for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.
5. If the PLL is enabled by the option byte, it outputs the clock after an additional delay of $t_{STARTUP}$ (see *Figure 11: PLL output frequency timing diagram on page 39*).

Register description

AWUFH control/status register (AWUCSR)

AWUCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	AWUF	AWUM	AWUEN
-	-	-	-	-	R/W	R/W	R/W

Table 20. AWUCSR register description

Bit	Bit name	Function
7:3	-	Reserved, must be kept cleared
2	AWUF	Auto wakeup flag This bit is set by hardware when the AWU module generates an interrupt and cleared by software on reading AWUCSR. Writing to this bit does not change its value. 0: No AWU interrupt occurred 1: AWU interrupt occurred
1	AWUM	Auto wakeup measurement This bit enables the AWU RC oscillator and connects its output to the input capture of the 12-bit autoreload timer. This allows the timer to measure the AWU RC oscillator dispersion and then compensate this dispersion by providing the right value in the AWUPRE register. 0: Measurement disabled 1: Measurement enabled
0	AWUEN	Auto wakeup from halt enabled This bit enables the auto wakeup from halt feature: Once halt mode is entered, the AWUFH wakes up the microcontroller after a time delay dependent on the AWU prescaler value. It is set and cleared by software. 0: AWUFH (auto wakeup from halt) mode disabled 1: AWUFH (auto wakeup from halt) mode enabled

AWUPR prescaler register (AWUPR)

AWUPR — Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
AWUPR[7:0]							
R/W							

Table 21. AWUPR register description

Bit	Bit name	Function
7:0	AWUPR[7:0]	Auto wakeup prescaler These 8 bits define the AWUPR dividing factor as explained

Table 22. AWUPR dividing factor

AWUPR[7:0]	Dividing factor
00h	Forbidden
01h	1
...	...
FEh	254
FFh	255

In AWU mode, the period that the MCU stays in halt mode (t_{AWU} in *Figure 27: AWUF halt timing diagram on page 63*) is defined by

$$t_{AWU} = 64 \times AWUPR \times \frac{1}{f_{AWURC}} + t_{RCSTRT}$$

This prescaler register can be programmed to modify the time that the MCU stays in halt mode before waking up automatically.

Note: If 00h is written to AWUPR, depending on the product, an interrupt is generated immediately after a HALT instruction or the AWUPR remains unchanged.

Table 23. AWU register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0049h	**AWUPR** Reset value	AWUPR7 1	AWUPR6 1	AWUPR5 1	AWUPR4 1	AWUPR3 1	AWUPR2 1	AWUPR1 1	AWUPR0 1
004Ah	**AWUCSR** Reset value	0	0	0	0	0	AWUF	AWUM	AWUEN

10 I/O ports

10.1 Introduction

The I/O ports allow data transfer. An I/O port contains up to eight pins. Each pin can be programmed independently either as a digital input or digital output. In addition, specific pins may have several other functions. These functions can include external interrupt, alternate signal input/output for on-chip peripherals or analog input.

10.2 Functional description

A data register (DR) and a data direction register (DDR) are always associated with each port. The option register (OR), which allows input/output options, may or may not be implemented. The following description takes into account the OR register. Refer to *Section 10.7: Device-specific I/O port configuration on page 73* for device specific information.

An I/O pin is programmed using the corresponding bits in the DDR, DR and OR registers: Bit x corresponding to pin x of the port.

Figure 29: I/O port general block diagram on page 70 shows the generic I/O block diagram.

10.2.1 Input modes

Clearing the DDRx bit selects input mode. In this mode, reading its DR bit returns the digital value from that I/O pin.

If an OR bit is available, different input modes can be configured by software: Floating or pull-up. Refer to *Section 10.3: I/O port implementation on page 72* for configuration.

Note: 1 Writing to the DR modifies the latch value but does not change the state of the input pin.

2 Do not use read/modify/write instructions (BSET/BRES) to modify the DR register.

External interrupt function

External interrupt capability is selected using the EISR register. If EISR bits are <> 0, the corresponding pin is used as external interrupt. In this case, the ORx bit can select the pin as either interrupt floating or interrupt pull-up. In this configuration, a signal edge or level input on the I/O generates an interrupt request via the corresponding interrupt vector (eix).

Falling or rising edge sensitivity is programmed independently for each interrupt vector. The external interrupt control register (EICR) or the miscellaneous register controls this sensitivity, depending on the device.

A device may have up to seven external interrupts. Several pins may be tied to one external interrupt vector. Refer to *Section 2: Pin description* to see which ports have external interrupts.

If several I/O interrupt pins on the same interrupt vector are selected simultaneously, they are logically combined. For this reason, if one of the interrupt pins is tied low, it may mask the others.

External interrupts are hardware interrupts. Fetching the corresponding interrupt vector automatically clears the request latch. Changing the sensitivity of a particular external

interrupt clears this pending interrupt. This can be used to clear unwanted pending interrupts.

Spurious interrupts

When enabling/disabling an external interrupt by setting/resetting the related OR register bit, a spurious interrupt is generated if the pin level is low and its edge sensitivity includes falling/rising edge. This is due to the edge detector input, which is switched to '1' when the external interrupt is disabled by the OR register.

To avoid this unwanted interrupt, a ‘safe’ edge sensitivity (rising edge for enabling and falling edge for disabling) must be selected before changing the OR register bit and configuring the appropriate sensitivity again.

Caution: If a pin level change occurs during these operations (asynchronous signal input), as interrupts are generated according to the current sensitivity, it is advised to disable all interrupts before and to re-enable them after the complete previous sequence in order to avoid an external interrupt occurring on the unwanted edge.

This corresponds to the following steps:

1. To enable an external interrupt:
   - Set the interrupt mask with the SIM instruction (in cases where a pin level change could occur)
   - Select rising edge
   - Enable the external interrupt through the OR register
   - Select the desired sensitivity if different from rising edge
   - Reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur)
2. To disable an external interrupt:
   - Set the interrupt mask with the SIM instruction (in cases where a pin level change could occur)
   - Select falling edge
   - Disable the external interrupt through the OR register
   - Select rising edge
   - Reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur)

10.2.2 Output modes

Setting the DDRx bit selects output mode. Writing to the DR bits applies a digital value to the I/O through the latch. Reading the DR bits returns the previously stored value.

If an OR bit is available, different output modes can be selected by software: Push-pull or open-drain. Refer to *Section 10.3: I/O port implementation on page 72* for configuration.

Table 24. DR value and output pin status

DR	Push-pull	Open-drain
0	V_{OL}	V_{OL}
1	V_{OH}	Floating

10.2.3 Alternate functions

Many ST7 I/Os have one or more alternate functions. These may include output signals from, or input signals to, on-chip peripherals. *Table 2: Device pin description on page 17* describes which peripheral signals can be input/output to which ports.

A signal coming from an on-chip peripheral can be output on an I/O. To do this, enable the on-chip peripheral as an output (enable bit in the peripheral's control register). The peripheral configures the I/O as an output and takes priority over standard I/O programming. The I/O's state is readable by addressing the corresponding I/O data register.

Configuring an I/O as floating enables alternate function input. It is not recommended to configure an I/O as pull-up as this increases current consumption. Before using an I/O as an alternate input, configure it without interrupt. Otherwise spurious interrupts can occur.

Configure an I/O as input floating for an on-chip peripheral signal which can be input and output.

Caution: I/Os which can be configured as both an analog and digital alternate function need special attention. The user must control the peripherals so that the signals do not arrive at the same time on the same pin. If an external clock is used, only the clock alternate function should be employed on that I/O pin and not the other alternate function.

Figure 29. I/O port general block diagram

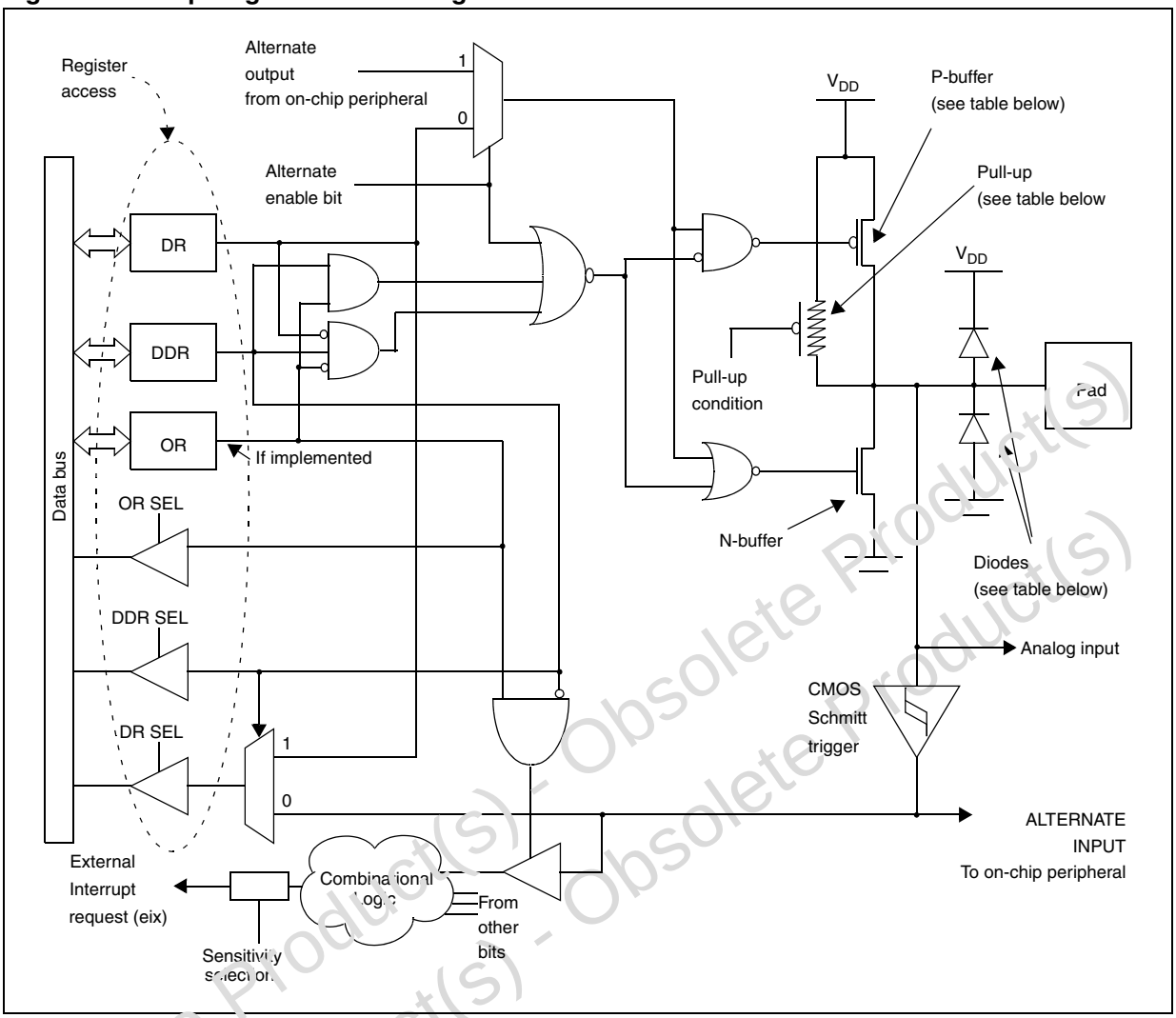
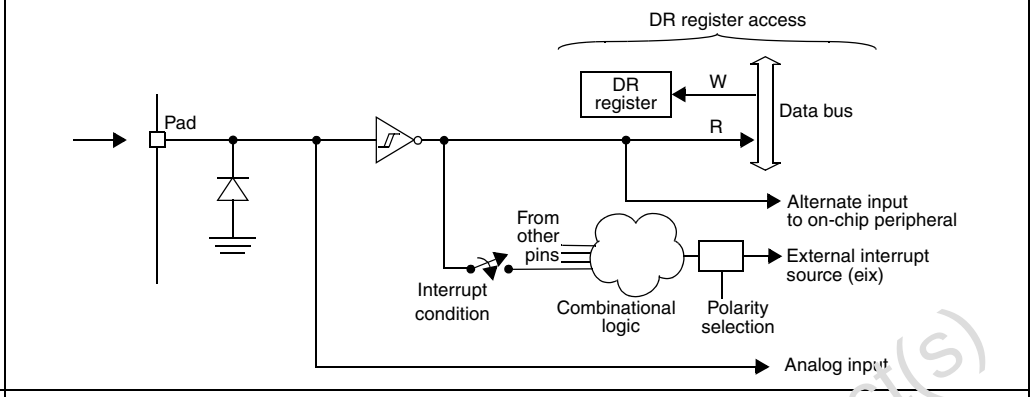
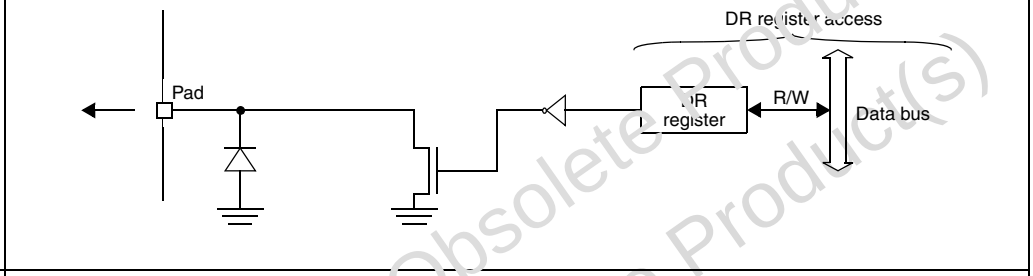
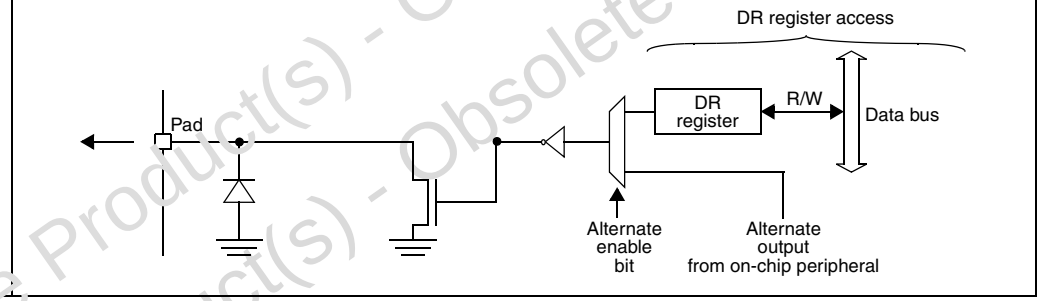


Table 25. Port mode options(1)

Configuration mode		Pull-up	P-buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without interrupt	Off	Off	On	On
	Pull-up with/without interrupt	On			
Output	Push-pull	Off	On		
	Open drain (logic level)		Off		

1. Legend: Off = implemented not activated; On = implemented and activated

Table 26. I/O configurations

	Hardware configuration
input(1)	
Open-drain output(2)	
Push-pull output(2)	

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

Analog alternate function

Configure the I/O as floating input to use an ADC input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail, connected to the ADC input.

Analog recommendations

Do not change the voltage level or loading on any I/O while conversion is in progress. Do not have clocking pins located close to a selected analog pin.

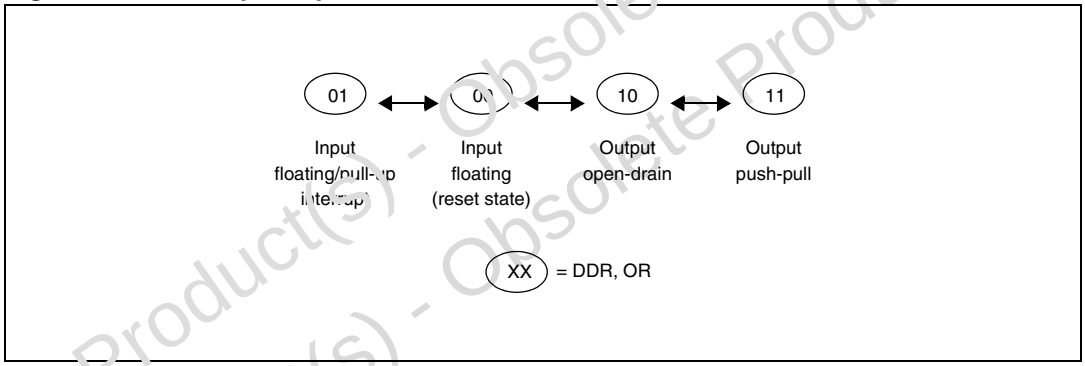
Warning: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

10.3 I/O port implementation

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific I/O port features such as ADC input or 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 30*. Other transitions are potentially risky and should be avoided, since they may present unwanted side-effects such as spurious interrupt generation.

Figure 30. Interrupt I/O port state transitions



10.4 Unused I/O pins

Unused I/O pins must be connected to fixed voltage levels. Refer to *Section 13.8: I/O port pin characteristics on page 161* .

10.5 Low-power modes

Table 27. Effect of low power modes on I/O ports

Mode	Description
Wait	No effect on I/O ports. External interrupts cause the device to exit from wait mode.
Halt	No effect on I/O ports. External interrupts cause the device to exit from halt mode.

10.6 Interrupts

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and if the I bit in the CC register is cleared (RIM instruction).

Table 28. I/O interrupt control/wake-up capability

Interrupt event	Event flag	Enable controlbit	Exit from wait	Exit from halt
External interrupt on selected external event	-	DDRx ORx	Yes	Yes

Related Documentation

- SPI communication between ST7 and EEPROM (AN970)
- S/W implementation of I^2C bus master (AN1045)
- Software LCD driver (AN1048)

10.7 Device-specific I/O port configuration

The I/O port register configurations are summarized as follows:

Table 29. Port configuration (standard ports)

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:0	Floating	Pull-up	Open drain	Push-pull
Port B	PB6:0	Floating	Pull-up	Open drain	Push-pull
Port C	PC1:0	Floating		Push-pull	

The selection between OSC1 or PC0 and OSC2 or PC1 is done by the option byte (refer to *Section 15.2: Option bytes on page 179*). Interrupt capability is not available on PC1:0. Port C is not present on ROM devices.

Note: PCOR not implemented but p-transistor always active in output mode (refer to Figure 29: I/O port general block diagram on page 70).

Table 30. Interrupt port configuration(1)

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:0	Floating	Pull-up interrupt	Open drain	Push-pull
Port B	PB6:0	floating	Pull-up interrupt	Open drain	Push-pull

1. On ports where the external interrupt capability is selected using the EISR register

Table 31. I/O port register map and reset values

Address(Hex.)	Register label	7	6	5	4	3	2	1	0
0000h	PADR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
0001h	PADDR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0002h	PAOR Reset value	MSB 0	1	0	0	0	0	0	LSB 0
0003h	PBDR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
0004h	PBDDR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0005h	PBOR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0006h	PCDR Reset value	MSB 0	0	0	0	0	0	1	LSB 1
0007h	PCDDR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

10.8 Multiplexed input/output ports

OSC1/PC0 are multiplexed on one pin (pin20) and OSC2/PC1 are multiplexed on pin19.

11 On-chip peripherals

11.1 Watchdog timer (WDG)

11.1.1 Introduction

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

11.1.2 Main features

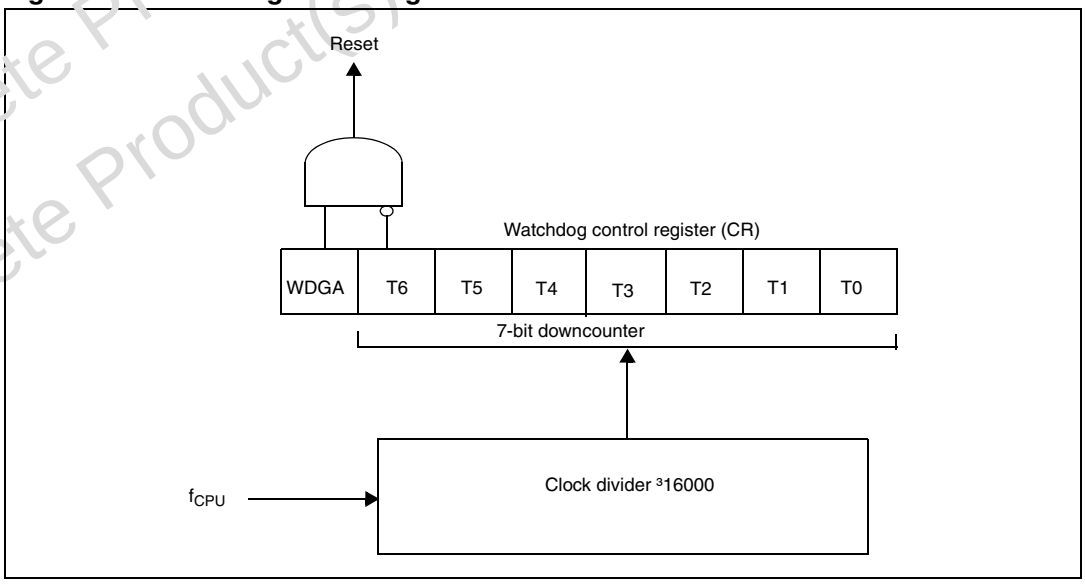
- Programmable free-running downcounter (64 increments of 16000 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware watchdog selectable by option byte

11.1.3 Functional description

The counter value stored in the CR register (bits T[6:0]) is decremented every 16000 machine cycles and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 30µs.

Figure 31. Watchdog block diagram



The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: It counts down, even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see *Table 32*):

- The WDGA bit is set (watchdog enabled).
- The T6 bit is set to prevent generating an immediate reset.
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated, it can be disabled only by a reset.

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

If the watchdog is activated, the HALT instruction generates a reset.

Table 32. Watchdog timing(1)(2)

f_{CPU} = 8 MHz		
WDG counter code	**min (ms)**	**max (ms)**
C0h	1	2
FFh	127	128

1. The timing variation shown in *Table 32* is due to the unknown status of the prescaler when writing to the CR register.
2. The number of CPU clock cycles applied during the reset phase (256 or 4096) must be taken into account in addition to these timings.

11.1.4 Hardware watchdog option

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

Refer to the option byte description in *Section 15.2: Option bytes on page 179*.

Using halt mode with the WDG (WDGHALT option)

If halt mode with watchdog is enabled by the option byte (no watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller (same behavior in active halt mode).

11.1.5 Interrupts

None.

11.1.6 Register description

Control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T[6:0]						
R/W	R/W						

Table 33. WDGCR register description

Bit	Bit name	Function
7	WDGA	Activation bit(1) 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
6:0	T[6:0]	7-bit counter (MSB to LSB) These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

1. The WDGA bit is not used if the hardware watchdog option is enabled by option byte.

Table 34. Watchdog timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Eh	WDGCR Reset value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

11.2 Dual 12-bit autoreload timer 4 (AT4)

11.2.1 Introduction

The 12-bit autoreload timer can be used for general-purpose timing functions. It is based on one or two free-running 12-bit upcounters with an input capture register and four PWM output channels. There are seven external pins:

- 4 PWM outputs
- ATIC/LTIC pins for the input capture function
- BREAK pin for forcing a break condition on the PWM outputs

11.2.2 Main features

- Single timer or dual timer mode with two 12-bit upcounters (CNTR1/CNTR2) and two 12-bit autoreload registers (ATR1/ATR2)
- Maskable overflow interrupts
- PWM mode
  - Generation of four independent PWMx signals
  - Dead time generation for half-bridge driving mode with programmable dead time
  - Frequency 2 kHz to 4 MHz (@ 8 MHz f_{CPU})
  - Programmable duty-cycles
  - Polarity control
  - Programmable output modes
- Output compare mode
- Input capture mode
  - 12-bit input capture register (ATICR)
  - Triggered by rising and falling edges
  - Maskable IC interrupt
  - Long range input capture
- Break control
- Flexible clock control
- One pulse mode on PWM2/3 (available only on Flash devices)
- Force update (available only on Flash devices)

Figure 32. Single timer mode (ENCNTR2 = 0)

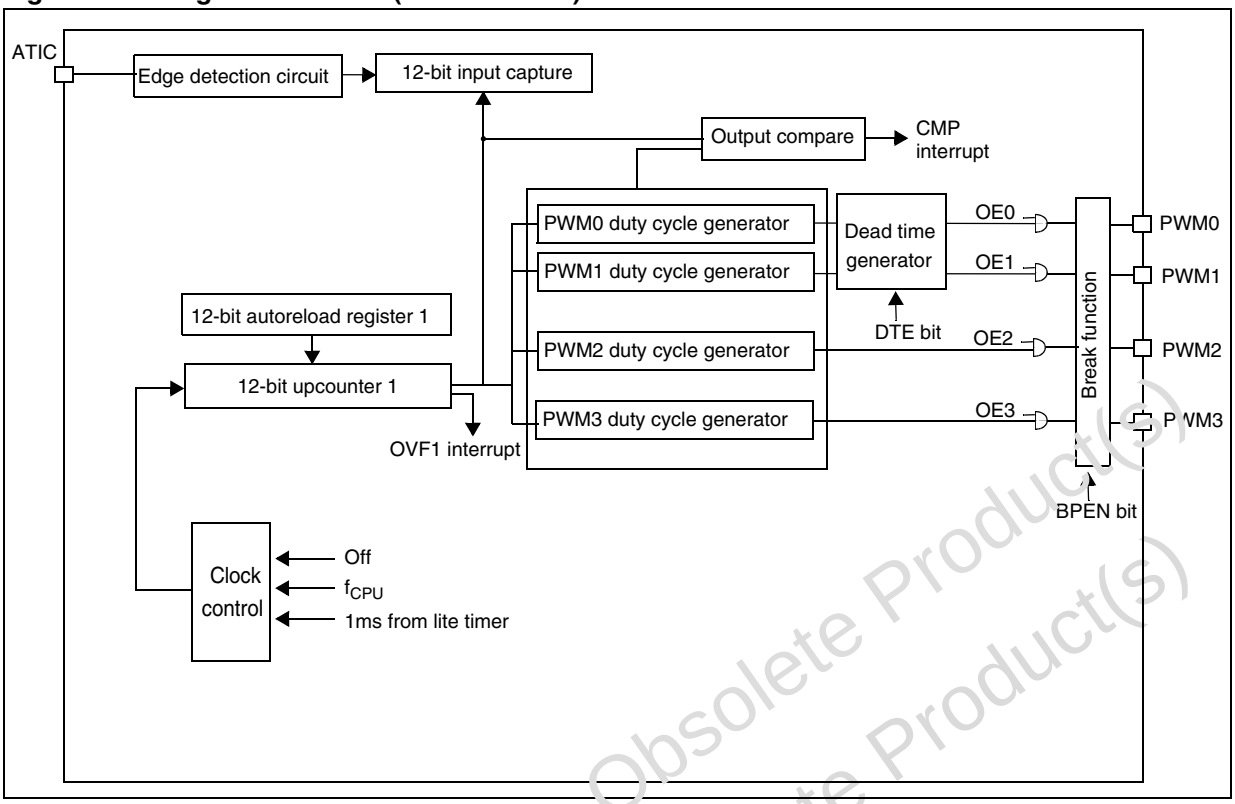
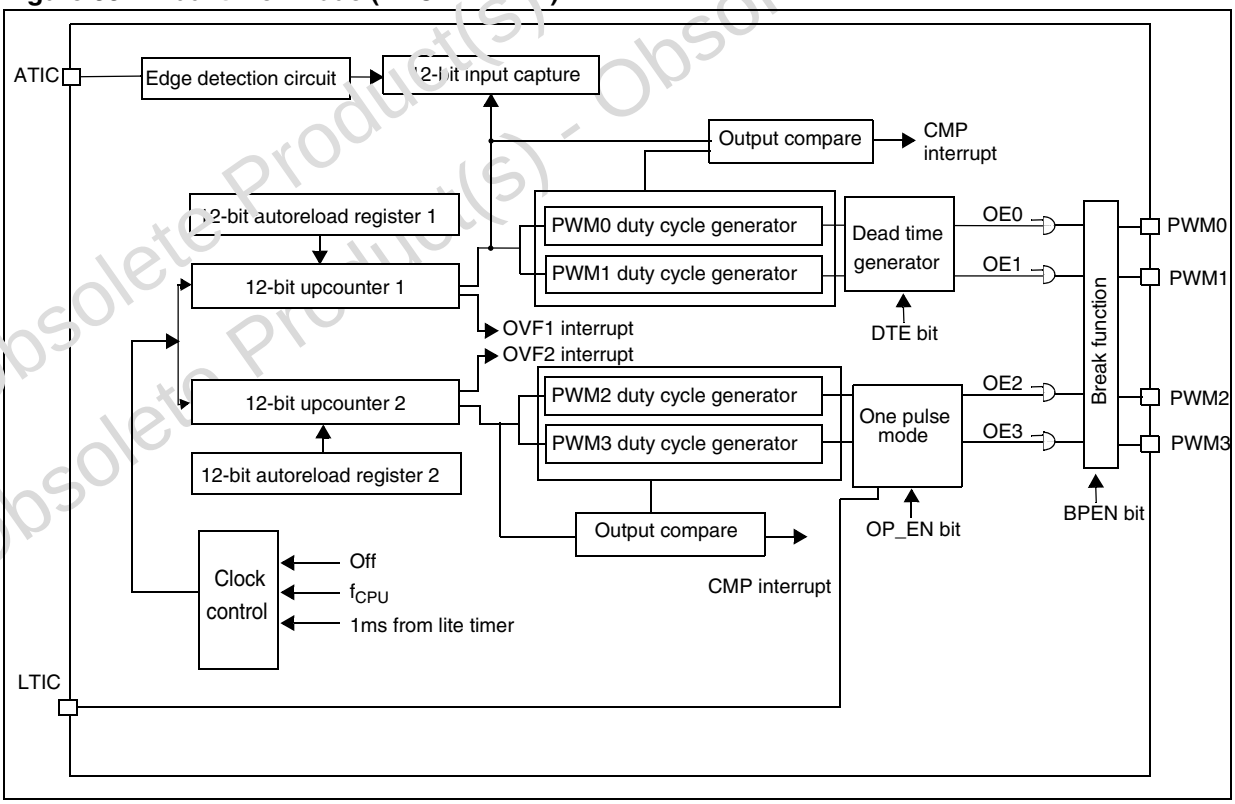


Figure 33. Dual timer mode (ENCNTR2 = 1)



11.2.3 Functional description

PWM mode

This mode allows up to four pulse width modulated signals to be generated on the PWMx output pins.

PWM frequency

The four PWM signals can have the same frequency (f_{PWM}) or can have two different frequencies. This is selected by the ENCNTR2 bit which enables single timer or dual timer mode (see *Figure 32: Single timer mode (ENCNTR2 = 0) on page 79* and *Figure 33: Dual timer mode (ENCNTR2 = 1) on page 79*).

The frequency is controlled by the counter period and the ATR register value. In dual timer mode, PWM2 and PWM3 can be generated with a different frequency controlled by CNTR2 and ATR2.

$$f_{PWM} = f_{COUNTER}/(4096 - ATR)$$

Following the above formula, if $f_{COUNTER}$ is 4 MHz, the maximum value of f_{PWM} is 2 MHz (ATR register value = 4094),the minimum value is 1 kHz (ATR register value = 0).

Note: 1 The maximum value of ATR is 4094 because it must be lower than the DC4R value, which in this case must be 4095.

Duty cycle

The duty cycle is selected by programming the DCRx registers. These are preload registers. The DCRx values are transferred in active duty cycle registers after an overflow event if the corresponding transfer bit (TRANx bit) is set.

The TRAN1 bit controls the PWMx outputs driven by counter 1 and the TRAN2 bit controls the PWMx outputs driven by counter 2.

PWM generation and output compare are done by comparing these active DCRx values with the counter.

The maximum available resolution for the PWMx duty cycle is:

$$\text{Resolution} = 1/(4096 - ATR)$$

Where ATR is equal to 0. With this maximum resolution, 0% and 100% duty cycle can be obtained by changing the polarity.

At reset, the counter starts counting from 0.

When an upcounter overflow occurs (OVF event), the preloaded duty cycle values are transferred to the active duty cycle registers and the PWMx signals are set to a high level. When the upcounter matches the active DCRx value, the PWMx signals are set to a low level. To obtain a signal on a PWMx pin, the contents of the corresponding active DCRx register must be greater than the contents of the ATR register.

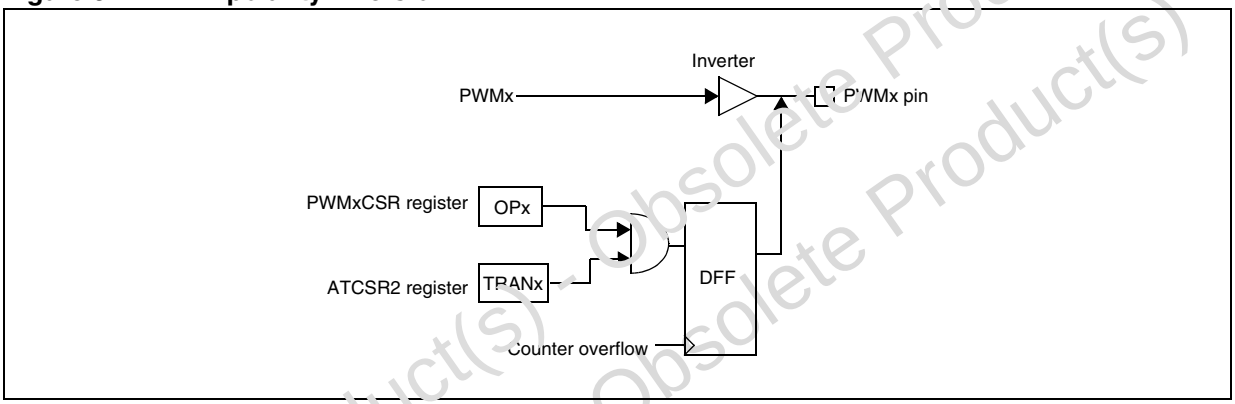
Note: ***For ROM devices only:*** *The PWM can be enabled/disabled only in overflow ISR, otherwise the first pulse of PWM can be different from expected one because no force overflow function is present.*

The maximum value of ATR is 4094 because it must be lower than the DCR value, which in this case must be 4095.

Polarity inversion

The polarity bits can be used to invert any of the four output signals. The inversion is synchronized with the counter overflow if the corresponding transfer bit in the ATCSR2 register is set (reset value). See *Figure 34*.

Figure 34. PWM polarity inversion



The data flip flop (DFF) applies the polarity inversion when triggered by the counter overflow input.

Output control

The PWMx output signals can be enabled or disabled using the OEx bits in the PWMCR register.

Figure 35. PWM function

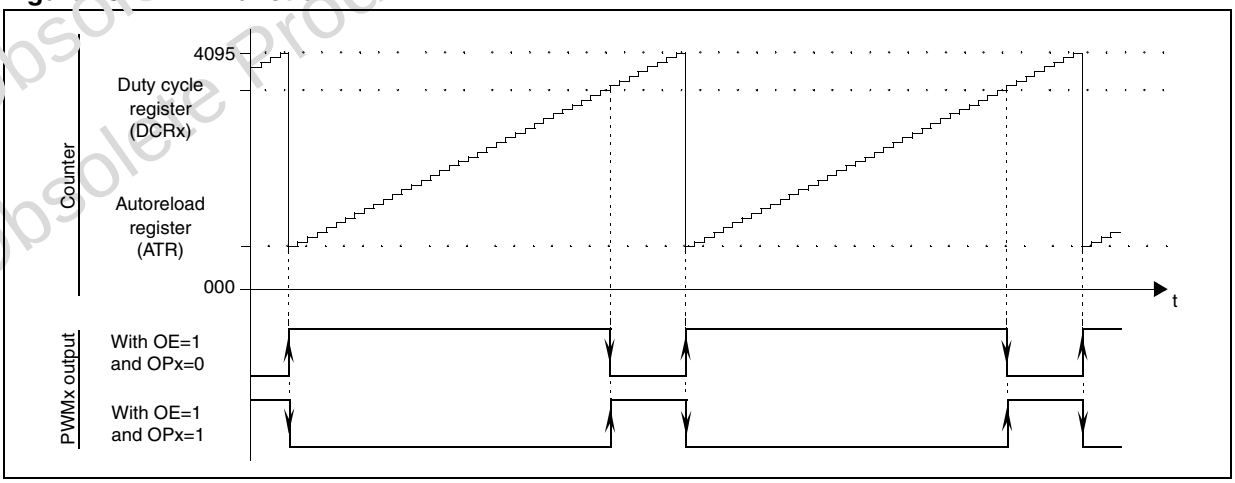
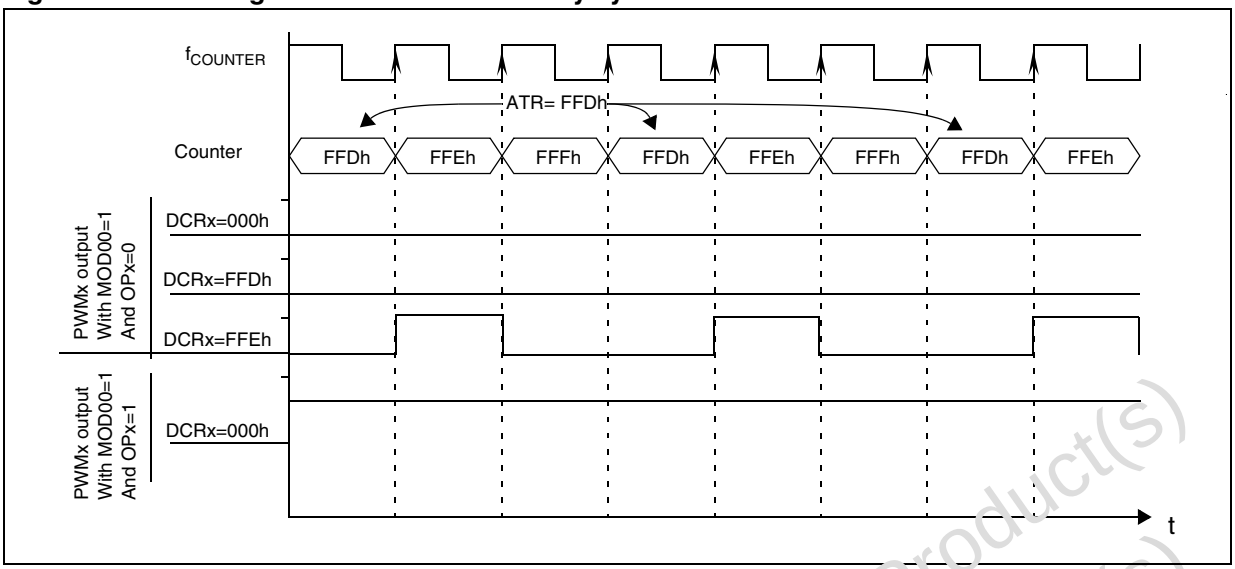


Figure 36. PWM signal from 0% to 100% duty cycle



Dead time generation

A dead time can be inserted between PWM0 and PWM1 using the DTGR register. This is required for half-bridge driving where PWM signals must not be overlapped. The non-overlapping PWM0/PWM1 signals are generated through a programmable dead time by setting the DTE bit.

Dead time value = DT[6:0] x Tcounter1

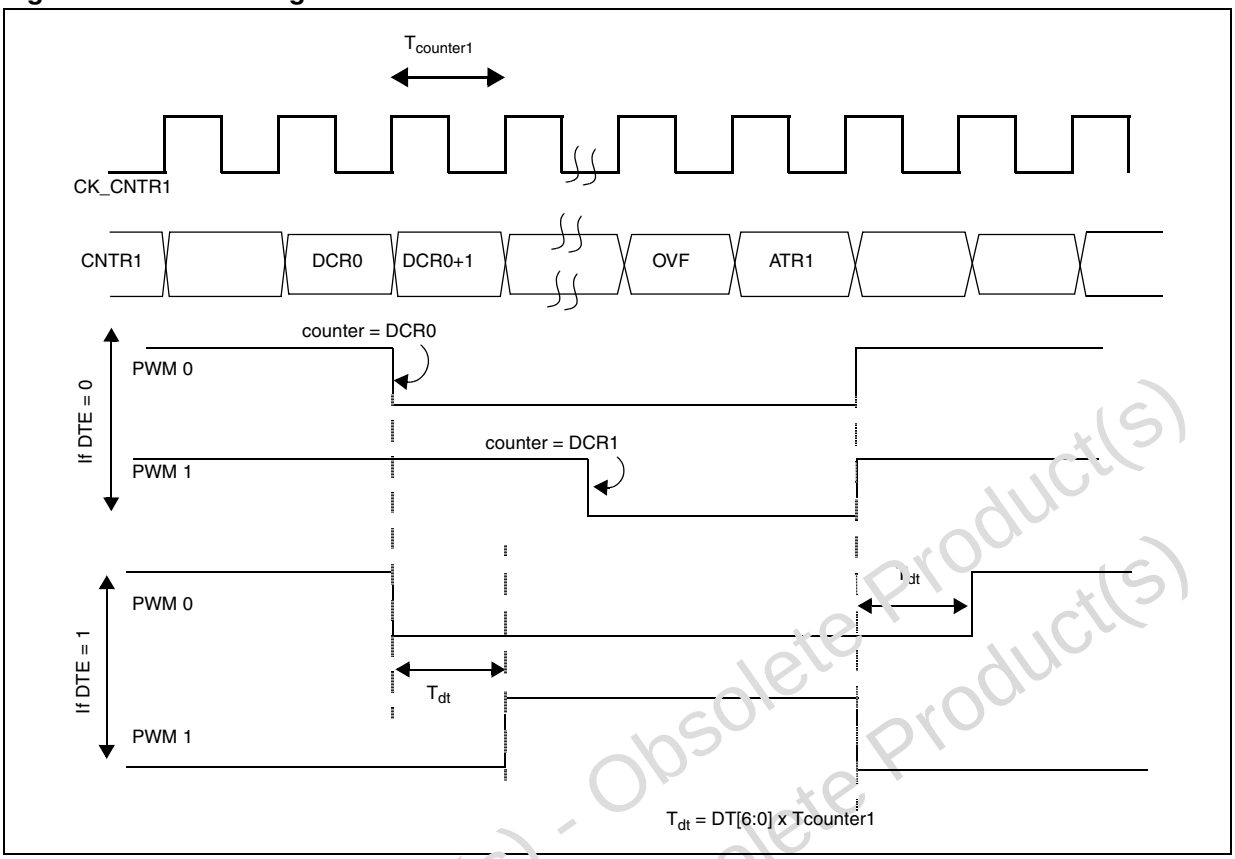
DTGR[7:0] is buffered inside so as to avoid deforming the current PWM cycle. The DTGR effect will take place only after an overflow.

Note: 1 Dead time is generated only when DTE = 1 and DT[6:0] ≠ 0. If DTE is set and DT[6:0] = 0, PWM output signals will be at their reset state.

2 Half-bridge driving is possible only if polarities of PWM0 and PWM1 are not inverted, that is, if OP0 and OP1 are not set. If polarity is inverted, overlapping PWM0/PWM1 signals will be generated.

3 Dead time generation does not work at 1ms timebase.

Figure 37. Dead time generation



In the above example, when the DTE bit is set:

- PWM goes low at DCR0 match and goes high at ATR1 + T_{dt}
- PWM1 goes high at DCR0 + T_{dt} and goes low at ATR match.

With this programmable delay (T_{dt}), the PWM0 and PWM1 signals which are generated are not overlapped.

Break function

The break function can be used to perform an emergency shutdown of the application being driven by the PWM signals.

The break function is activated by the external BREAK pin. In order to use the break function it must be previously enabled by software setting the BPEN bit in the BREAKCR register.

The break active level can be programmed by the BREDGE bit in the BREAKCR register (in ROM devices the active level is not programmable; the break active level is low). When an active level is detected on the BREAK pin, the BA bit is set and the break function is activated. In this case, the PWM signals are forced to BREAK value if the respective OEx bit is set in the PWMCR register.

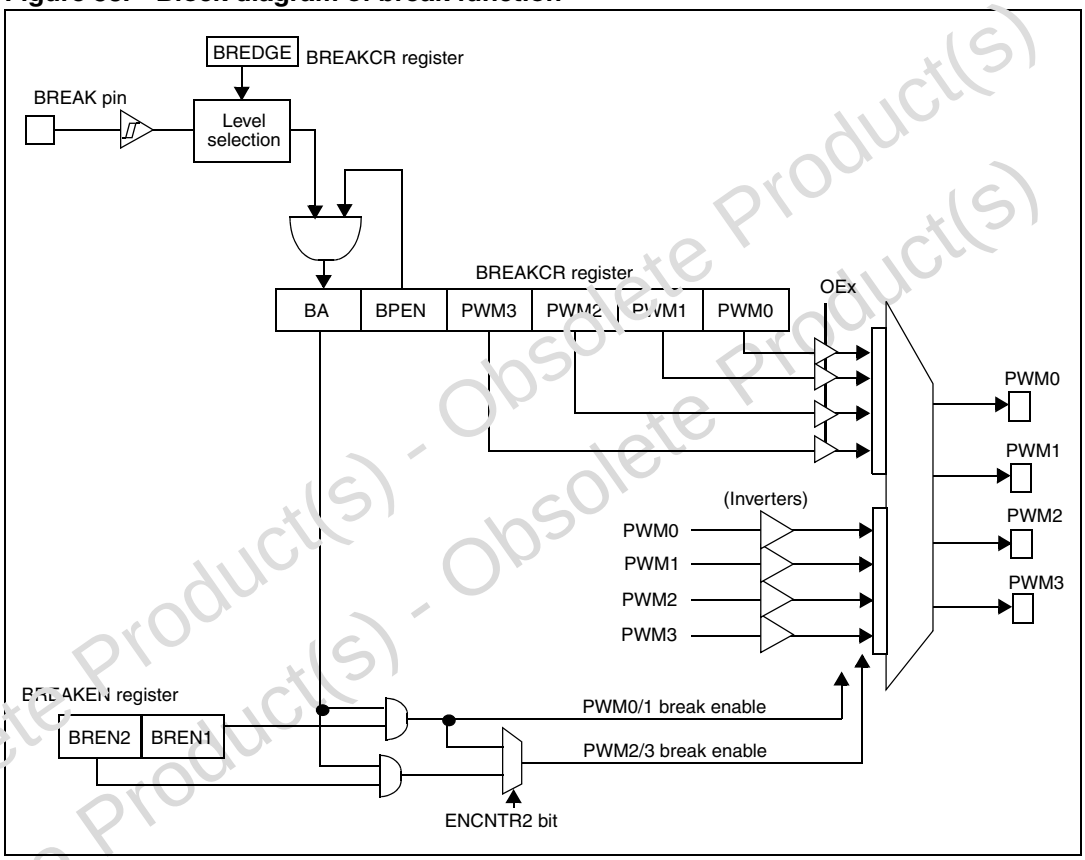
Software can set the BA bit to activate the break function without using the BREAK pin. The BREN1 and BREN2 bits in the BREAKEN register are used to enable the break activation on the two counters respectively. In dual timer mode, the break for PWM2 and PWM3 is enabled by the BREN2 bit. In single timer mode, the BREN1 bit enables the break for all PWM channels. In ROM devices, BREN1 and BREN2 are both forced by hardware at high level and all PWMs are enabled.

When a break function is activated (BA bit = 1 and BREN1/BREN2 = 1):

- The break pattern (PWM[3:0] bits in the BREAKCR is forced directly on the PWMx output pins if respective OEx is set (after the inverter)
- The 12-bit PWM counter CNTR1 is put to its reset value, that is 00h (if BREN1 = 1)
- The 12-bit PWM counter CNTR2 is put to its reset value, that is 00h (if BREN2 = 1)
- ATR1, ATR2, preload and active DCRx are put to their reset values
- Counters stop counting

When the break function is deactivated after applying the break (BA bit goes from 1 to 0 by software), the timer takes the control of the PWM ports.

Figure 38. Block diagram of break function



Output compare mode

To use this function, load a 12-bit value in the preload DCRxH and DCRxL registers.

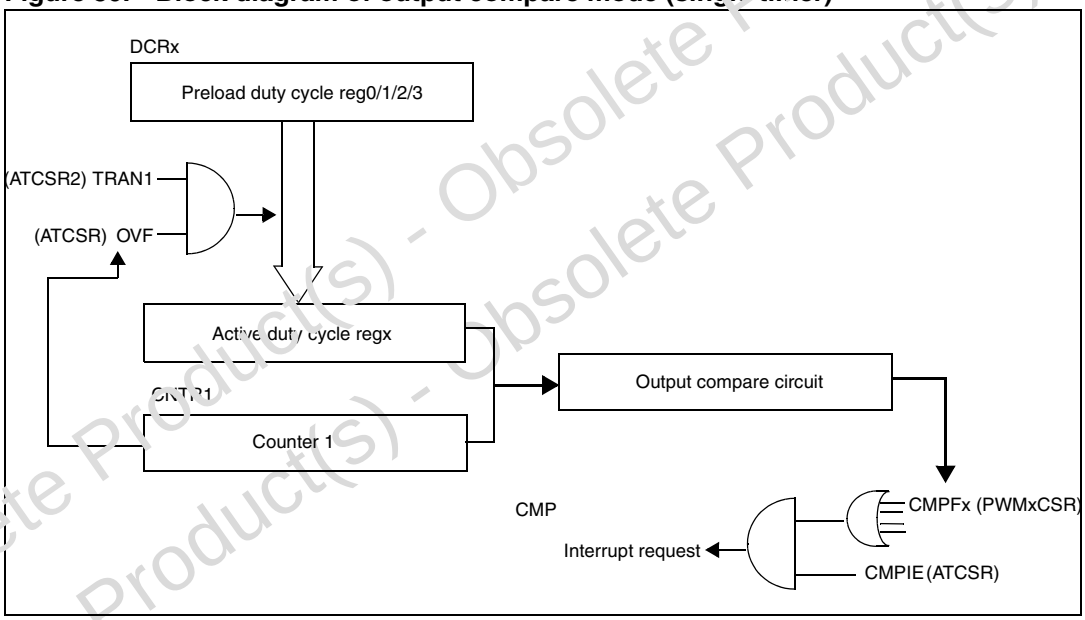
When the 12-bit upcounter CNTR1 reaches the value stored in the active DCRxH and DCRxL registers, the CMPFx bit in the PWMxCSR register is set and an interrupt request is generated if the CMPIE bit is set.

In single timer mode the output compare function is performed only on CNTR1. The difference between both the modes is that in single timer mode, CNTR1 can be compared with any of the four DCR registers, and in dual timer mode, CNTR1 is compared with DCR0 or DCR1 and CNTR2 is compared with DCR2 or DCR3. In ROM devices, the CNTR2 counter is not used for this comparison.

Note: 1 The output compare function is only available for DCRx values other than 0 (reset value).

2 Duty cycle registers are buffered internally. The CPU writes in Preload Duty Cycle Registers and these values are transferred to Active Duty Cycle Registers after an overflow event if the corresponding transfer bit (TRANx bit) is set. Output compare is done by comparing these active DCRx values with the counters.

Figure 39. Block diagram of output compare mode (single timer)



Input capture mode

The 12-bit ATICR register is used to latch the value of the 12-bit free running upcounter CNTR1 after a rising or falling edge is detected on the ATIC pin. When an input capture occurs, the ICF bit is set and the ATICR register contains the value of the free running upcounter. An IC interrupt is generated if the ICIE bit is set. The ICF bit is reset by reading the ATICRH/ATICRL register when the ICF bit is set. The ATICR is a read only register and always contains the free running upcounter value which corresponds to the most recent input capture. Any further input capture is inhibited while the ICF bit is set.

Figure 40. Block diagram of input capture mode

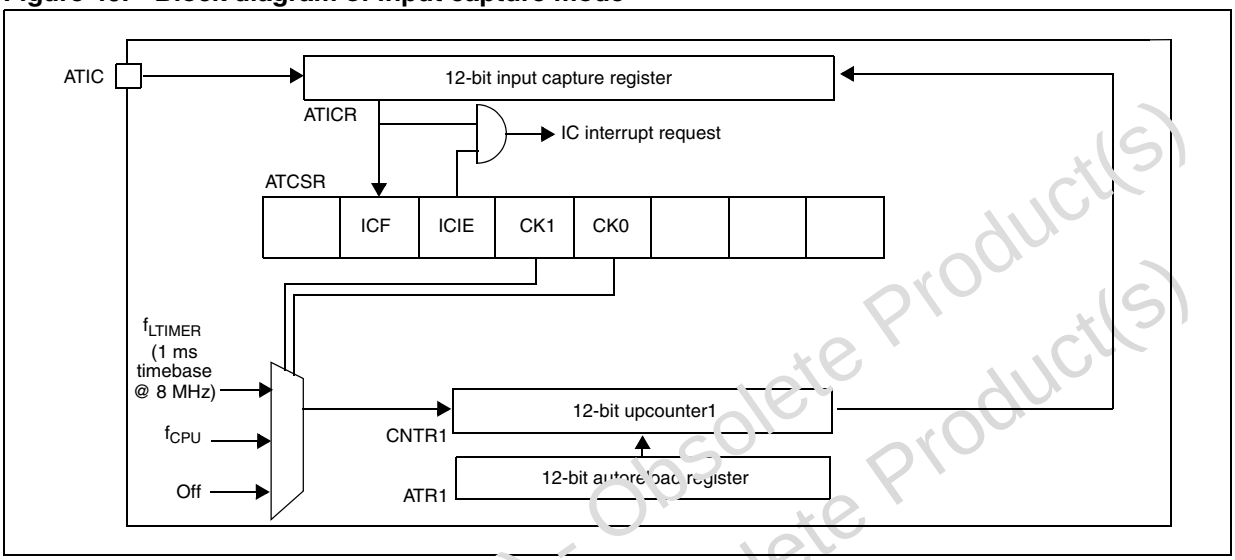
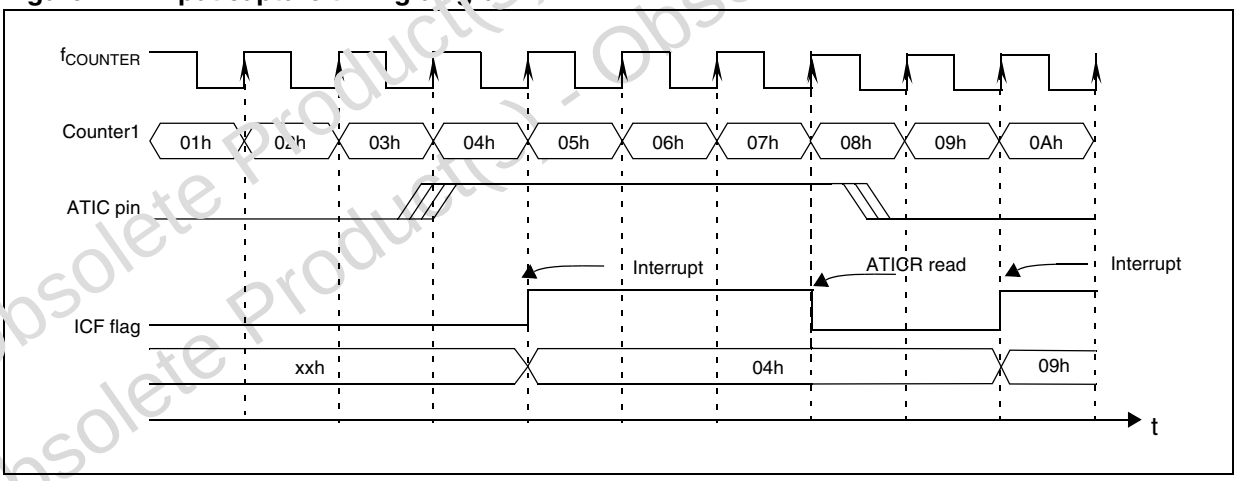


Figure 41. Input capture timing diagram



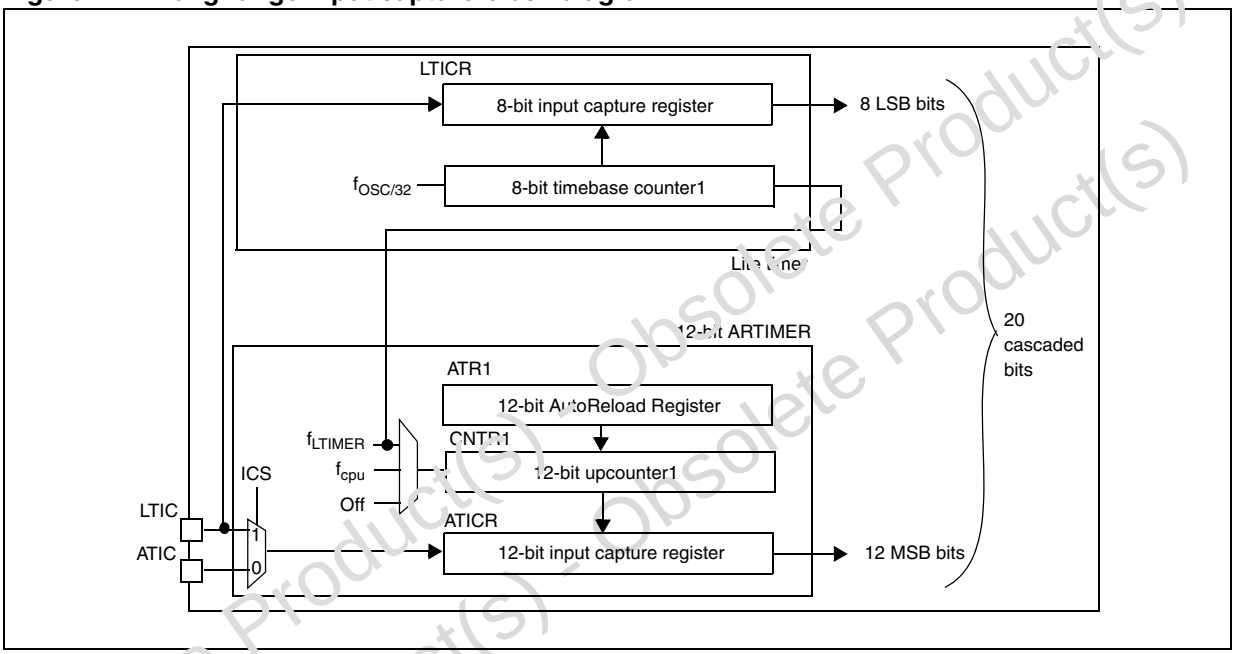
Long input capture

Pulses that last more than 8 µs can be measured with an accuracy of 4µs if f_{OSC} = 8 MHz under the following conditions:

- The 12-bit AT4 timer is clocked by the lite timer (RTC pulse: CK[1:0] = 01 in the ATCSR register)
- The ICS bit in the ATCSR2 register is set so that the LTIC pin is used to trigger the AT4 timer capture.
- The signal to be captured is connected to LTIC pin
- Input capture registers LTICR, ATICRH and ATICRL are read

This configuration allows to cascade the lite timer and the 12-bit AT4 timer to get a 20-bit input capture value. Refer to *Figure 42*.

Figure 42. Long range input capture block diagram



Since the input capture flags (ICF) for both timers (AT4 timer and LT timer) are set when signal transition occurs, software must mask one interrupt by clearing the corresponding ICIE bit before setting the ICS bit.

If the ICS bit changes (from 0 to 1 or from 1 to 0), a spurious transition might occur on the input capture signal because of different values on LTIC and ATIC. To avoid this situation, it is recommended to do the following:

- First, reset both ICIE bits
- Then set the ICS bit
- Reset both ICF bits
- Finally, set the ICIE bit of desired interrupt

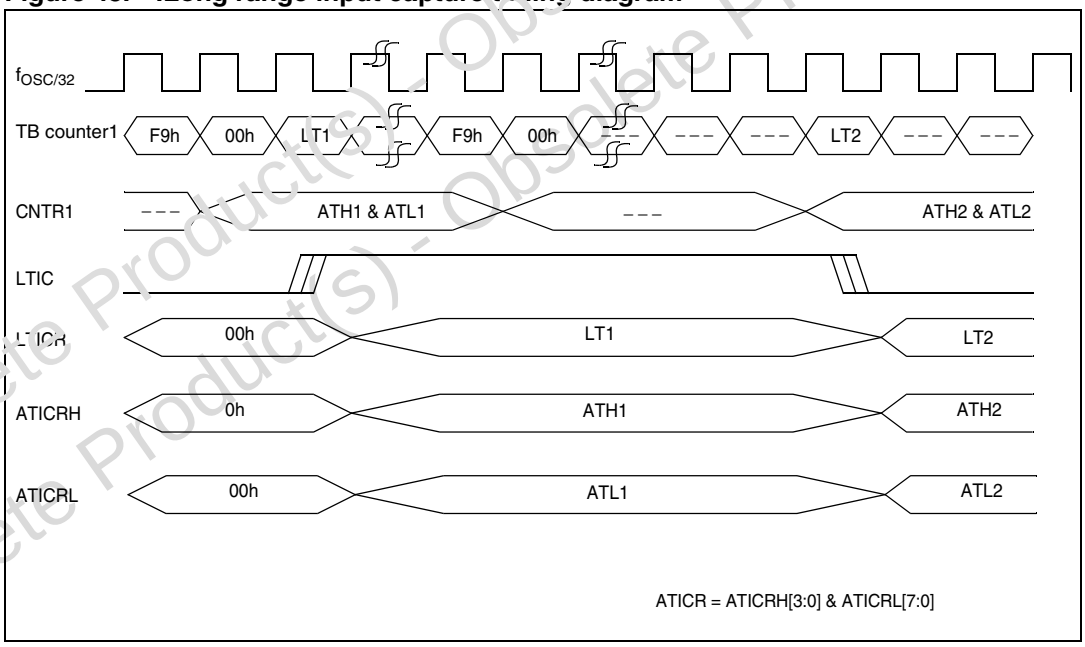
Both timers are used to compute a pulse length with long input capture feature. The procedure is not straight-forward and is as follows:

- At the first input capture on the rising edge of the pulse, we assume that values in the registers are as follows:
  - LTICR = LT1
  - ATICRH = ATH1
  - ATICRL = ATL1
  - Hence ATICR1 [11:0] = ATH1 & ATL1
  - Refer to *Figure 43*.
- At the second input capture on the falling edge of the pulse, we assume that the values in the registers are as follows:
  - LTICR = LT2
  - ATICRH = ATH2
  - ATICRL = ATL2
  - Hence ATICR2 [11:0] = ATH2 & ATL2

Now pulse width P between first capture and second capture will be:

P = decimal (F9 – LT1 + LT2 + 1) * 0.004ms + decimal ((FFF * N) + N + ATICR2 - ATICR1 – 1) * 1ms where N = No of overflows of 12-bit CNTR1

Figure 43. .Long range input capture timing diagram



One pulse mode (available only on Flash devices)

One pulse mode can be used to control PWM2/3 signal with an external LTIC pin. This mode is available only in dual timer mode that is only for CNTR2, when the OP_EN bit in PWM3CSR register is set.

One pulse mode is activated by the external LTIC input. The active edge of the LTIC pin is selected by the OPEDGE bit in the PWM3CSR register.

After obtaining the active edge of the LTIC pin, CNTR2 is reset (000h) and PWM3 is set to high. CNTR2 starts counting from 000h and when it reaches the active DCR3 value, PWM3 goes low. Until this time, any further transitions on the LTIC signal will have no effect. If there are LTIC transitions after CNTR2 reaches the DCR3 value, CNTR2 is reset again and PWM3 goes high.

If there are no more LTIC active edges after the first active edge, CNTR2 counts until it reaches the ARR2 value, it is then reset and PWM3 is set to high. The counter again starts counting from 000h. When it reaches the active DCR3 value, PWM3 goes low, after which the counter counts until it reaches ARR2, it is reset and PWM3 is set to high again, and the cycle continues in this manner.

The same operation applies for PWM2, but in this case the comparison is done on DCR2

OP_EN and OPEDGE bits take effect on the fly and are not synchronized with the counter 2 overflow.

The output bit OP2/3 can be used to invert the polarity of PWM2/3 in one pulse mode. The update of these bits (OP2/3) is synchronized with the Counter 2 overflow, provided the TRAN2 bit is set.

Note:

1. *The time taken from activation of LTIC input and CNTR2 reset is between 1 and 2 t_{CPU} cycles, that is 125 ns to 250 ns (with 8 MHz f_{CPU}).*
2. *To avoid spurious interrupts, the litetimer input capture interrupt should be disabled while 12-bit ARTimer is in one pulse mode.*
3. *Priority of various conditions is as follows for PWM3:*

   Break > one pulse mode with active LTIC edge > forced overflow by s/w > one pulse mode without active LTIC edge > normal PWM operation
4. *It is possible to update DCR2/3 and OP2/3 at the counter 2 reset because the update is synchronized with the counter reset. This is managed by the overflow interrupt which is generated if the counter is reset either due to ARR match or active pulse at LTIC pin.*
5. *DCR2/3 and OP2/3 update in one pulse mode is done dynamically using force update in software.*
6. *DCR3 update in this mode is not synchronized with any event. That may lead to a longer next PWM3 cycle duration than expected just after the change (refer to Figure 46: Dynamic DCR2/3 update in one pulse mode on page 91).*
7. *In one pulse mode, the ATR2 value must be greater than the DCR2/3 value for PWM2/3 (opposite to normal PWM mode).*
8. *If there is an active edge on the LTIC pin after the counter has reset due to an ARR2 match, then the timer again is reset and appears as modified duty cycle, depending on whether the new DCR value is less than or more than the previous value.*
9. *The TRAN2 bit should be set along with the FORCE2 bit with the same instruction after a write to the DCR register.*

10 ARR2 value should be changed after an overflow in one pulse mode to avoid any irregular PWM cycle.

11 When exiting from one pulse mode, the OP_EN bit in the PWM3CSR register should be reset first and then the ENCNTR2 bit (if counter 2 must be stopped).

How to enter one pulse mode

1. Load ATR2H/ATR2L with required value
2. Load DCR3H/DCR3L for PWM3. ATR2 value must be greater than DCR3
3. Set OP3 in PWM3CSR if polarity change is required
4. Select CNTR2 by setting ENCNTR2 bit in ATCSR2
5. Set TRAN2 bit in ATCSR2 to enable transfer
6. 'Wait for overflow' by checking the OVF2 flag in ATCSR2
7. Select counter clock using CK<1:0> bits in ATCSR
8. Set OP_EN bit in PWM3CSR to enable one pulse mode
9. Enable PWM3 by OE3 bit of PWMCR

The 'wait for overflow' in step 6 can be replaced by forced update.

Follow the same procedure for PWM2 with the bits corresponding to PWM2.

Note: When break is applied in one pulse mode, dual 12-bit autoreload timer 4, CNTR2, DCR2/3 and ATR2 registers are reset. Consequently, these registers must be initialized again when break is removed.

Figure 44. Block diagram of one pulse mode

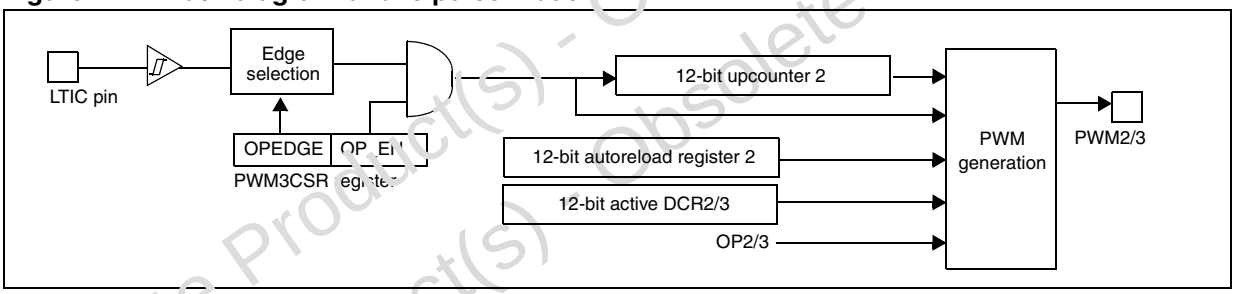


Figure 45. One pulse mode timing diagram

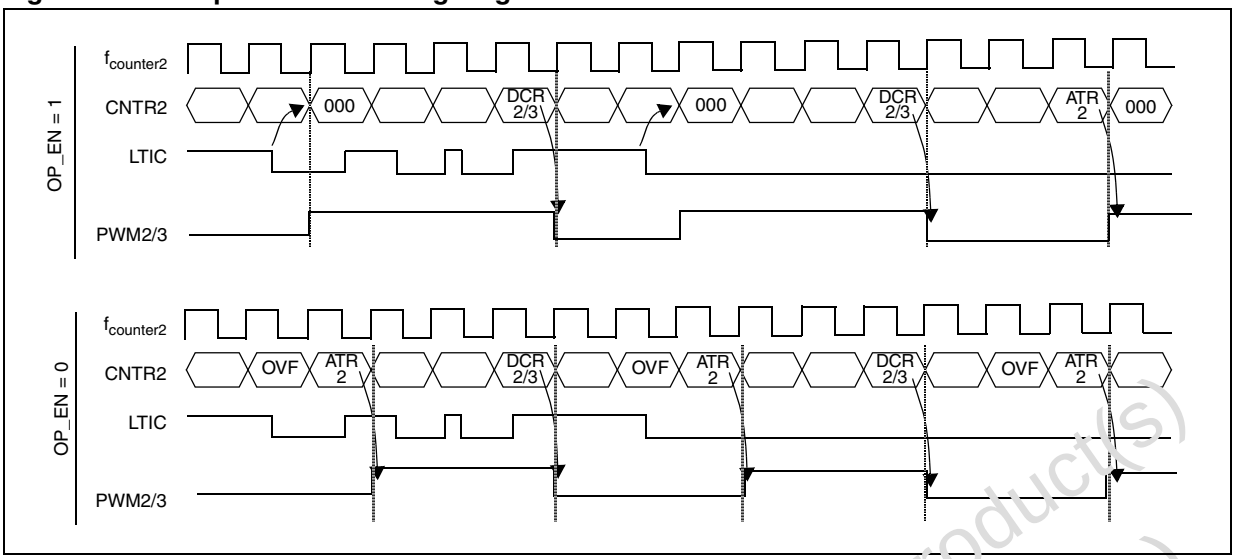
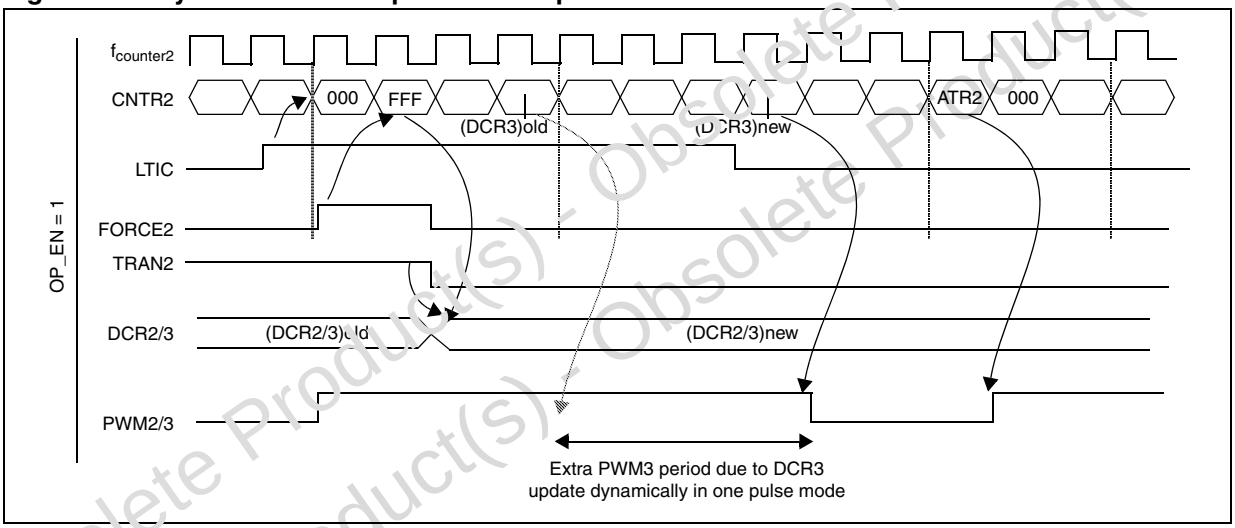


Figure 46. Dynamic DCR2/3 update in one pulse mode



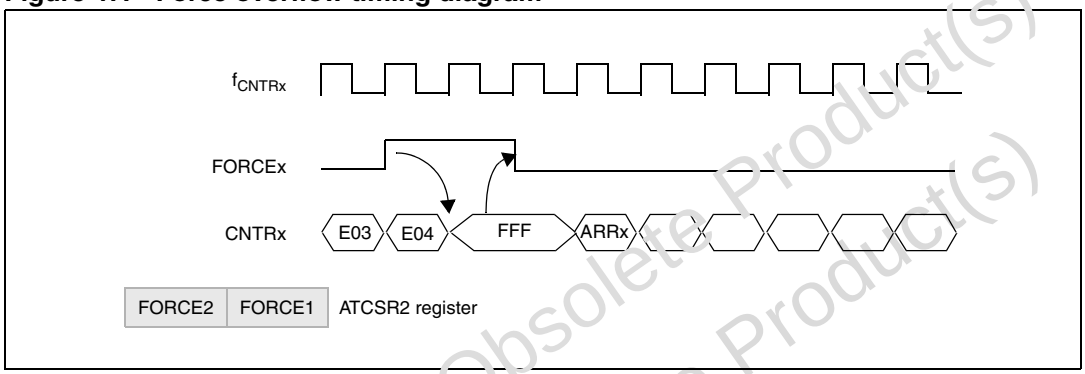
Force update (available only on Flash devices)

In order not to wait for the counter$_x$ overflow to load the value into active DCRx registers, a programmable counter$_x$ overflow is provided. For both counters, a separate bit is provided which when set, starts the counters with the overflow value, that is FFFh. After overflow, the counters start counting from their respective autoreload register values.

These bits are FORCE1 and FORCE2 in the ATCSR2 register. FORCE1 is used to force an overflow on counter 1 and FORCE2 is used for counter 2. These bits are set by software and reset by hardware after the respective counter overflow event has occurred.

This feature can be used at any time. All related features such as PWM generation, output compare, input capture and one pulse can be used this way.

Figure 47. Force overflow timing diagram



11.2.4 Low power modes

Table 35. Effect of low power modes on AT4

Mode	Description
Wait	No effect on AT timer
Halt	AT timer halted

11.2.5 Interrupts

Table 36. AT4 interrupt control/wake-up capability(1)

Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from Halt	Exit from active halt
Overflow event	OVF1	OVFIE1	Yes	No	Yes
AT4 IC event	ICF	ICIE			No
CMP event	CMPFx	CMPIE			
Overflow event2	OVF2	OVFIE2			

1. The CMP and AT4 IC events are connected to the same interrupt vector. The OVF event is mapped on a separate vector (see *Section 8: Interrupts*). They generate an interrupt if the enable bit is set in the ATCSR register and the interrupt mask in the CC register is reset (RIM instruction).

11.2.6 Register description

Timer control status register (ATCSR)

ATCSR Reset value: 0x00 0000 (x0h)

7	6	5	4	3	2	1	0
Reserved	ICF	ICIE	CK[1:0]		OVF1	OVFIE1	CMPIE
-	R/W	R/W	R/W		R/W	R/W	R/W

Table 37. ATCSR register description

Bit	Bit name	Function
7	-	Reserved, must be kept cleared
6	ICF	Input capture flag This bit is set by hardware and cleared by software by reading the ATICR register (a read access to ATICRH or ATICRL will clear this flag). Writing to this bit does not change the bit value. 0: No input capture 1: An input capture has occurred
5	ICIE	IC interrupt enable This bit is set and cleared by software. 0: Input capture interrupt disabled 1: Input capture interrupt enabled
4:3	CK[1:0]	Counter clock selection These bits are set and cleared by software and cleared by hardware after a reset. They select the clock frequency of the counter as follows/ 00: Counter clock selection = off 01: Counter clock selection = f_{LTIMER} (1ms timebase @ 8 MHz) 10: Counter clock selection = f_{CPU}
2	OVF1	Overflow flag This bit is set by hardware and cleared by software by reading the ATCSR register. It indicates the transition of the counter CNTR1 from FFFh to ATR1 value. 0: No counter overflow occurred 1: Counter overflow occurred
1	OVFIE1	Overflow interrupt enable This bit is read/write by software and cleared by hardware after a reset. 0: Overflow interrupt disabled 1: Overflow interrupt enabled
0	CMPIE	Compare interrupt enable This bit is read/writen by software and cleared by hardware after a reset. It can be used to mask the interrupt generated when any of the CMPFx bit is set. 0: Output compare interrupt disabled 1: Output compare interrupt enabled

Counter register 1 high (CNTR1H)

CNTR1H — Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	CNTR1[11:8]			
-	-	-	-	R			

Counter register 1 low (CNTR1L)

CNTR1L — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
CNTR1[7:0]							
R							

Table 38. CNTR1H and CNTR1L register descriptions

Bit	Bit name	Function
15:12	-	Reserved, must be kept cleared
11:0	CNTR1[11:0]	Counter value This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. The CNTRH register can be incremented between the two reads, and in order to be accurate when $f_{TIMER} = f_{CPU}$, the software should take this into account when CNTRL and CNTRH are read. If CNTRL is close to its highest value, CNTRH could be incremented before it is read. When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

Autoreload register high (ATR1H)

ATR1H Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	ATR1[11:8]			
-	-	-	-	R/W			

Autoreload register low (ATR1L)

ATR1L Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ATR1[7:0]							
R/W							

Table 39. ATR1H and ATR1L register descriptions

Bit	Bit name	Function
15:12	-	Reserved, must be kept cleared
11:0	ATR1[11:0]	Autoreload register 1 This is a 12-bit register which is written by software. The ATR1 register value is automatically loaded into the upcounter CNTR1 when an overflow occurs. The register value is used to set the PWM frequency.

PWM output control register (PWMCR)

PWMCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	OE3	Reserved	OE2	Reserved	OE1	Reserved	OE0
-	R/W	-	R/W	-	R/W	-	R/W

Table 40. PWMCR register description

Bit	Bit name	Function
7, 5, 3, 1	-	Reserved, must be kept cleared
6, 4, 2, 0	OE[3:0]	PWMx output enable These bits are set and cleared by software and cleared by hardware after a reset. 0: PWM mode disabled. PWMx output alternate function disabled (I/O pin free for general purpose I/O) 1: PWM mode enabled

PWMx control status register (PWMxCSR)

PWMxCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	OP_EN	OPEDGE	OPx	CMPFx
-	-	-	-	R/W	R/W	R/W	R/W

Table 41. PWMxCSR register description

Bit	Bit name	Function
7:4	-	Reserved, must be kept cleared
3	OP_EN	One pulse mode enable (not applicable to ROM devices) This bit is read/writen by software and cleared by hardware after a reset. This bit enables the one pulse feature for PWM2 and PWM3. (It is only available for PWM3CSR). 0: One pulse mode disable for PWM2/3 1: One pulse mode enable for PWM2/3
2	OPEDGE	One pulse edge selection (not applicable to ROM devices) This bit is read/writen by software and cleared by hardware after a reset. This bit selects the polarity of the LTIC signal for one pulse feature. This bit will be effective only if OP_EN bit is set. (It is Only available for PWM3CSR) 0: Falling edge of LTIC is selected 1: Rising edge of LTIC is selected
1	OPx	PWMx output polarity This bit is read/writen by software and cleared by hardware after a reset. This bit selects the polarity of the PWM signal. 0: The PWM signal is not inverted 1: The PWM signal is inverted
0	CMPFx	PWMx *compare flag* This bit is set by hardware and cleared by software by reading the PWMxCSR register. It indicates that the upcounter value matches the active DCRx register value. 0: Upcounter value does not match DCRx value 1: Upcounter value matches DCRx value

Break control register (BREAKCR)

BREAKCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	BREDGE	BA	BPEN	PWM[3:0]			
-	R/W	R/W	R/W	R/W			

Table 42. BREAKCR register description

Bit	Bit name	Function
7	-	Reserved, must be kept cleared
6	BREDGE	Break input edge selection (not applicable to ROM devices) This bit is read/writen by software and cleared by hardware after reset. It selects the active level of break signal 0: Low level of break selected as active level 1: High level of break selected as active level
5	BA	Break active This bit is read/writne by software, cleared by hardware after reset and set by hardware when the BREAK pin is low. It activates/deactivates the break function. 0: Break not active 1: Break active
4	BPEN	Break pin enable This bit is read/write by software and cleared by hardware after reset. 0: Break pin disabled 1: Break pin enabled
3:0	PWM[3:0]	Break pattern These bits are read/write by software and cleared by hardware after a reset. They are used to force the four PWMx output signals into a stable state when the break function is active and corresponding OEx bit is set.

PWMx duty cycle register high (DCRxH)

DCRxH — Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	DCRx[11:8]			
-	-	-	-	R/W			

PWMx duty cycle register low (DCRxL)

DCRxL — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
DCRx[7:0]							
R/W							

Table 43. DCRxH and DCRxL register descriptions

Bit	Bit name	Function
15:12	-	Reserved, must be kept cleared
11:0	DCRx[11:0]	PWMx duty cycle value This 12-bit value is written by software. It defines the duty cycle of the corresponding PWM output signal (see *Figure 35: PWM function on page 81*). In PWM mode (OEx = 1 in the PWMCR register) the DCR[11:0] bits define the duty cycle of the PWMx output signal (see *Figure 35*). In output compare mode, they define the value to be compared with the 12-bit upcounter value.

Input capture register high (ATICRH)

ATICRH Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	ICR[11:8]			
-	-	-	-	R			

Input capture register low (ATICRL)

ATICRL Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICR[7:0]							
R							

Table 44. ATICRH and ATICRL register descriptions

Bit	Bit name	Function
15:12	-	Reserved, must be kept cleared
11:0	ICR[11:0]	Input capture data This is a 12-bit register which is readable by software and cleared by hardware after a reset. The ATICR register contains the captured value of the 12-bit CNTR1 register when a rising or falling edge occurs on the ATIC or LTIC pin (depending on ICS). Capture will only be performed when the ICF flag is cleared.

Break enable register (BREAKEN)

BREAKEN Reset value: 0000 0011 (03h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BREN2	BREN1
-	-	-	-	-	-	R/W	R/W

Table 45. BREAKEN register description

Bit	Bit name	Function
7:2	-	Reserved, must be kept cleared
1	BREN2	Break enable for counter 2 (forced at high level in ROM devices) This bit is read/writen by software. It enables the break functionality for counter 2 if BA bit is set in BREAKCR. It controls PWM2/3 if ENCNTR2 bit is set. 0: No break applied for CNTR2 1: Break applied for CNTR2
0	BREN1	Break enable for counter 1 (forced at high level in ROM devices) This bit is read/writen by software. It enables the break functionality for counter 1. If BA bit is set, it controls PWM0/1 by default, and controls PWM2/3 also if ENCNTR2 bit is reset. 0: No break applied for CNTR1 1: Break applied for CNTR1

Timer control register2 (ATCSR2)

ATCSR2 Reset value: 0000 0011 (03h)

7	6	5	4	3	2	1	0
FORCE2	FORCE1	ICS	OVFIE2	OVF2	ENCNTR2	TRAN2	TRAN1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46. ATCSR2 register description

Bit	Bit name	Function
7	FORCE2	Force counter 2 overflow (not applicable to ROM devices) This bit is read/set by software. When set, it loads FFFh in the CNTR2 register. It is reset by hardware one CPU clock cycle after counter 2 overflow has occurred. 0: No effect on CNTR2 1: Loads FFFh in CNTR2 *Note: This bit must not be reset by software*
6	FORCE1	Force counter 1 overflow (forced at high level in ROM devices) This bit is read/set by software. When set, it loads FFFh in CNTR1 register. It is reset by hardware one CPU clock cycle after counter 1 overflow has occurred. 0: No effect on CNTR1 1: Loads FFFh in CNTR1 *Note: This bit must not be reset by software*
5	ICS	Input capture shorted This bit is read/writen by software. It allows the ATtimer CNTR1 to use the LTIC pin for long input capture. 0: ATIC for CNTR1 input capture 1: LTIC for CNTR1 input capture
4	OVFIE2	Overflow interrupt 2 enable This bit is read/writen by software and controls the overflow interrupt of counter 2. 0: Overflow interrupt disabled 1: Overflow interrupt enabled
3	OVF2	Overflow flag This bit is set by hardware and cleared by software by reading the ATCSR2 register. It indicates the transition of the counter 2 from FFFh to ATR2 value. 0: No counter overflow occurred 1: Counter overflow occurred
2	ENCNTR2	Enable counter 2 for PWM2/3 This bit is read/writen by software and switches the PWM2/3 operation to the CNTR2 counter. If this bit is set, PWM2/3 will be generated using CNTR2. 0: PWM2/3 is generated using CNTR1 1: PWM2/3 is generated using CNTR2 *Note: Counter 2 becomes frozen when the ENCNTR2 bit is reset. When ENCNTR2 is set again, the counter will restart from the last value.*

Table 46. ATCSR2 register description (continued)

Bit	Bit name	Function
1	TRAN2	Transfer enable 2 This bit is read/writen by software, cleared by hardware after each completed transfer and set by hardware after reset. It controls the transfers on CNTR2. It allows the value of the preload DCRx registers to be transferred to the active DCRx registers after the next overflow event. The OPx bits are transferred to the shadow OPx bits in the same way. *Note 1: DCR2/3 transfer is controlled using this bit if ENCNTR2 bit is set* *Note 2: This bit must not be reset by software*
0	TRAN1	Transfer enable 1 This bit is read/writen by software, cleared by hardware after each completed transfer and set by hardware after reset. It controls the transfers on CNTR1. It allows the value of the preload DCRx registers to be transferred to the active DCRx registers after the next overflow event. The OPx bits are transferred to the shadow OPx bits in the same way. *Note 1: DCR0,1 transfers are always controlled using this bit* *Note 2: DCR2/3 transfer is controlled using this bit if ENCNTR2 is reset* *Note 3: This bit must not be reset by software*

Autoreload register2 high (ATR2H)

ATR2H Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	ATR2[11:8]			
-		-	-	R/W			

Autoreload register2 low (ATR2L)

ATR2L Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ATR2[7:0]							
R/W							

Table 47. ATR2H and ATR2L register descriptions

Bit	Bit name	Function
15:12	-	Reserved, must be kept cleared
11:0	ATR2[11:0]	Autoreload register 2 This is a 12-bit register which is written by software. The ATR2 register value is automatically loaded into the upcounter CNTR2 when an overflow of CNTR2 occurs. The register value is used to set the PWM2/PWM3 frequency when ENCNTR2 is set.

Dead time generator register (DTGR)

DTGR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
DTE	DT[6:0]						
R/W	R/W						

Table 48. DTGR register description

Bit	Bit name	Function
7	DTE	Dead time enable This bit is read/writen by software. It enables a dead time generation on PWM0/PWM1. 0: No dead time insertion 1: Dead time insertion enabled
6:0	DT[6:0]	Dead time value These bits are read/writen by software. They define the dead time inserted between PWM0/PWM1. Dead time is calculated as follows: Dead time = DT[6:0] x Tcounter *Note: If DTE is set and DT[6:0] = 0, PWM output signals are at their reset state*

Table 49. Register map and reset values

Add (Hex.)	Register label	7	6	5	4	3	2	1	0
0D	ATCSR Reset value	0	ICF 0	ICIE 0	CK1 0	CK0 0	OVF1 0	OVFIE1 0	CMPIE 0
0E	CNTR1H Reset value	0	0	0	0	CNTR1_11 0	CNTR1_10 0	CNTR1_9 0	CNTR1_8 0
0F	CNTR1L Reset value	CNTR1_7 0	CNTR1_8 0	CNTR1_7 0	CNTR1_6 0	CNTR1_3 0	CNTR1_2 0	CNTR1_1 0	CNTR1_0 0
10	ATR1H Reset value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
11	ATR1L Reset value	ATR7 0	ATR6 0	ATR5 0	ATR4 0	ATR3 0	ATR2 0	ATR1 0	ATR0 0
12	PWMCR Reset value	0	OE3 0	0	OE2 0	0	OE1 0	0	OE0 0
13	PWM0CSR Reset value	0	0	0	0	0	0	OP0 0	CMPF0 0
14	PWM1CSR Reset value	0	0	0	0	0	0	OP1 0	CMPF1 0
15	PWM2CSR Reset value	0	0	0	0	0	0	OP2 0	CMPF2 0
16	**PWM3CSR** Reset value	0	0	0	0	OP_EN 0	OPEDGE 0	OP3 0	CMPF3 0
17	**DCR0H** Reset value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
18	**DCR0L** Reset value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
19	**DCR1H** Reset value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1A	**DCR1L** Reset value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1B	**DCR2H** Reset value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1C	**DCR2L** Reset value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1D	**DCR3H** Reset value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1E	**DCR3L** Reset value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1F	**ATICRH** Reset value	0	0	0	0	ICR11 0	ICR10 0	ICR9 0	ICR8 0
20	**ATICRL** Reset value	ICR7 0	ICR6 0	ICR5 0	ICR4 0	ICR3 0	ICR2 0	ICR1 0	ICR0 0
21	**ATCSR2** Reset value	FORCE2 0	FORCE1 0	ICS 0	OVFIE2 0	OVF2 0	ENCNTR2 0	TRAN2 1	TRAN1 1

Table 49. Register map and reset values (continued)

Add (Hex.)	Register label	7	6	5	4	3	2	1	0
22	**BREAKCR** Reset value	BRSEL 0	BREDGE 0	BA 0	BPEN 0	PWM3 0	PWM2 0	PWM1 0	PWM0 0
23	**ATR2H** Reset value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
24	**ATR2L** Reset value	ATR7 0	ATR6 0	ATR5 0	ATR4 0	ATR3 0	ATR2 0	ATR1 0	ATR0 0
25	**DTGR** Reset value	DTE 0	DT6 0	DT5 0	DT4 0	DT3 0	DT2 0	DT1 0	DT0 0
26	**BREAKEN** Reset value	0	0	0	0	0	0	BREN2 1	BREN1 1

11.3 Lite timer 2 (LT2)

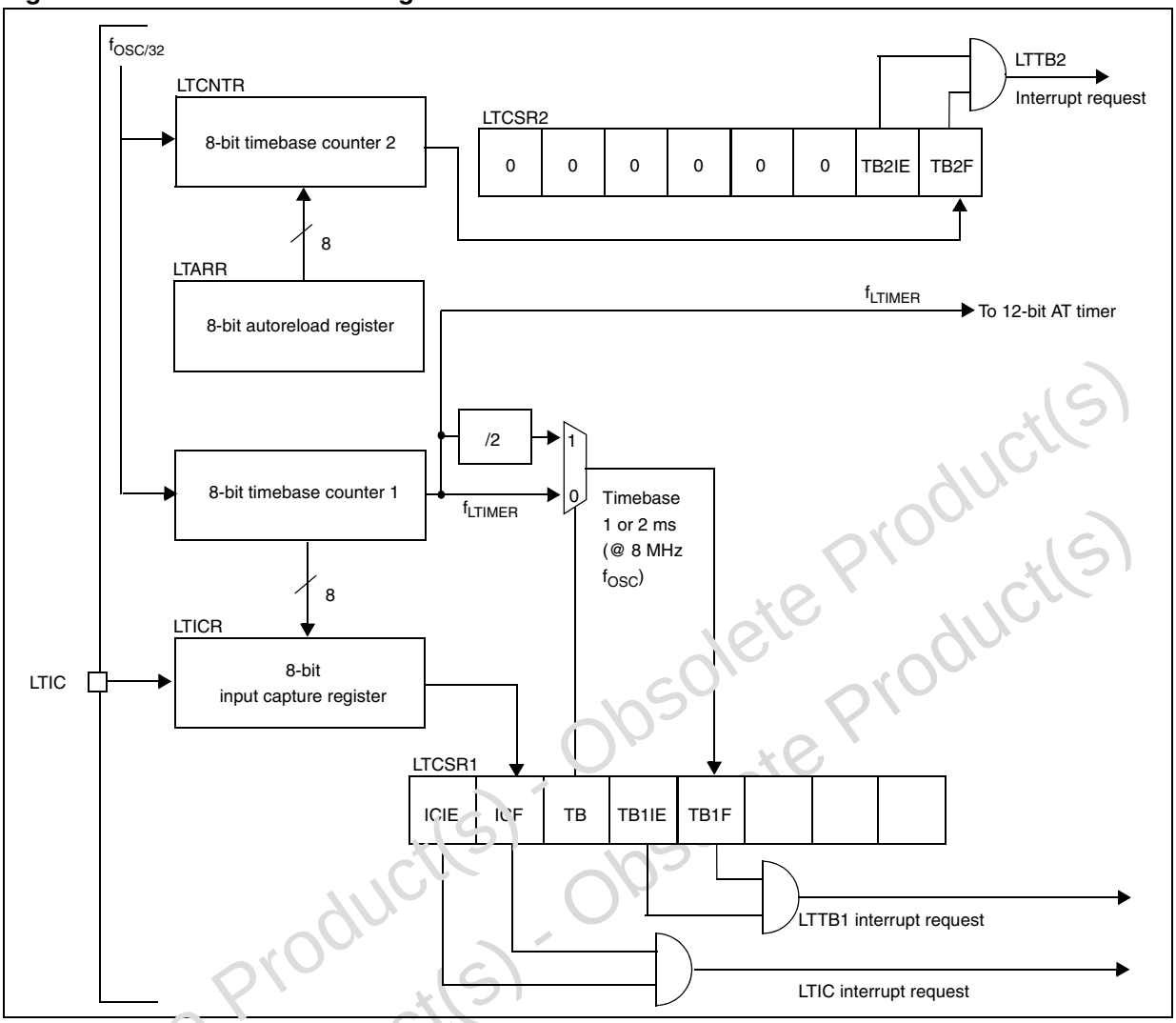
11.3.1 Introduction

The lite timer can be used for general-purpose timing functions. It is based on two free-running 8-bit upcounters and an 8-bit input capture register.

11.3.2 Main features

- Real-time clock
  - One 8-bit upcounter 1ms or 2ms timebase period (@ 8 MHz f_{OSC})
  - One 8-bit upcounter with autoreload and programmable timebase period from 4µs to 1.024ms in 4µs increments (@ 8 MHz f_{OSC})
  - 2 maskable timebase interrupts
- Input capture
  - 8-bit input capture register (LTICR)
  - Maskable interrupt with wakeup from halt mode capability

Figure 48. Lite timer 2 block diagram



11.3.3 Functional description

Timebase counter 1

The 8-bit value of counter 1 cannot be read or written by software. After an MCU reset, it starts incrementing from 0 at a frequency of $f_{OSC}/32$. An overflow event occurs when the counter rolls over from F9h to 00h. If f_{OSC} = 8 MHz, then the time period between two counter overflow events is 1 ms. This period can be doubled by setting the TB bit in the LTCSR1 register.

When counter 1 overflows, the TB1F bit is set by hardware and an interrupt request is generated if the TB1IE bit is set. The TB1F bit is cleared by software reading the LTCSR1 register.

Input capture

The 8-bit input capture register is used to latch the free-running upcounter (counter 1) 1 after a rising or falling edge is detected on the LTIC pin. When an input capture occurs, the ICF bit is set and the LTICR1 register contains the MSB of counter 1. An interrupt is generated if the ICIE bit is set. The ICF bit is cleared by reading the LTICR register.

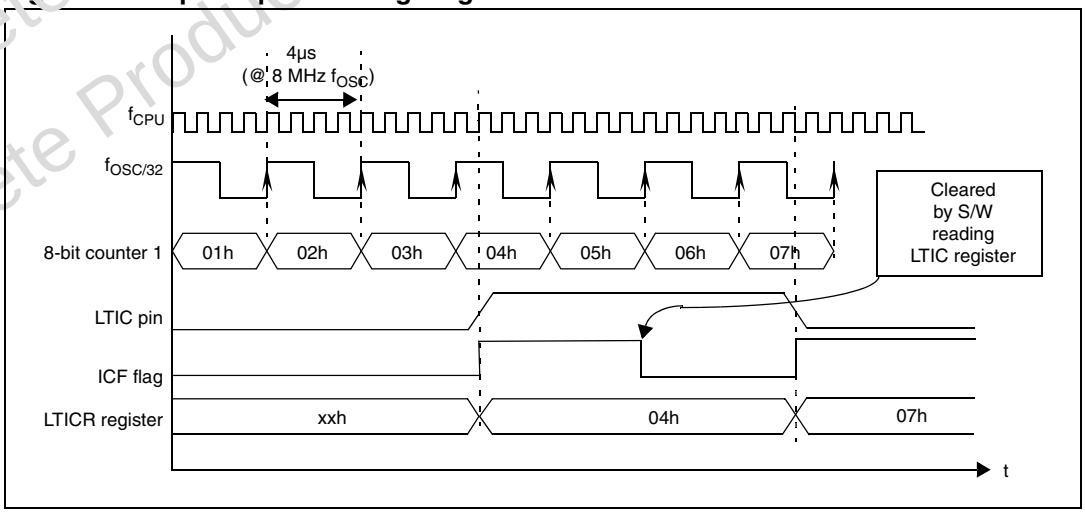
The LTICR is a read-only register and always contains the data from the last input capture. Input capture is inhibited if the ICF bit is set.

Timebase counter 2

Counter 2 is an 8-bit autoreload upcounter. It can be read by accessing the LTCNTR register. After an MCU reset, it increments at a frequency of $f_{OSC}/32$ starting from the value stored in the LTARR register. A counter overflow event occurs when the counter rolls over from FFh to the LTARR reload value. Software can write a new value at any time in the LTARR register, this value will be automatically loaded in the counter when the next overflow occurs.

When counter 2 overflows, the TB2F bit in the LTCSR2 register is set by hardware and an interrupt request is generated if the TB2IE bit is set. The TB2F bit is cleared by software reading the LTCSR2 register.

Figure 49. Input capture timing diagram.



11.3.4 Low power modes

Table 50. Effect of low power modes on lite timer 2

Mode	Description
Slow	No effect on lite timer (this peripheral is driven directly by $f_{OSC}/32$)
Wait	No effect on lite timer
Active halt	No effect on lite timer
Halt	Lite timer stops counting

11.3.5 Interrupts

Table 51. Lite timer 2 interrupt control/wake-up capability(1)

Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from active halt	Exit from halt
Timebase 1 event	TB1F	TB1IE	Yes	Yes	No
Timebase 2 event	TB2F	TB2IE		No	
IC event	ICF	ICIE		No	

1. The TBxF and ICF interrupt events are connected to separate interrupt vectors (see *Section 8: Interrupts*). They generate an interrupt if the enable bit is set in the LTCSR1 or LTCSR2 register and the interrupt mask in the CC register is reset (RIM instruction).

11.3.6 Register description

Lite timer control/status register 2 (LTCSR2)

DTGR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TB2IE	TB2F
-	-	-	-	-	-	R/W	R/W

Table 52. LTCSR2 register description

Bit	Bit name	Function
7:2	-	Reserved, must be kept cleared
1	TB2IE	Timebase 2 interrupt enable This bit is set and cleared by software. 0: Timebase (TB2) interrupt disabled 1: Timebase (TB2) interrupt enabled
0	TB2F	Timebase 2 interrupt flag This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect. 0: No counter 2 overflow 1: A counter 2 overflow has occurred

Lite timer autoreload register (LTARR)

LTARR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
AR[7:0]							
R/W							

Table 53. LTARR register description

Bit	Bit name	Function
7:0	AR[7:0]	Counter 2 reload value These bits are read/writen by software. The LTARR value is automatically loaded into counter 2 (LTCNTR) when an overflow occurs.

Lite timer counter 2 (LTCNTR)

LTCNTR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
CNT[7:0]							
R							

Table 54. LTCNTR register description

Bit	Bit name	Function
7:0	CNT[7:0]	Counter 2 reload value This register is read by software. The LTARR value is automatically loaded into counter 2 (LTCNTR) when an overflow occurs

Lite timer control/status register (LTCSR1)

LTCSR1 Reset value: 0000 0000 (x0h)

7	6	5	4	3	2	1	0
ICIE	ICF	TB	TB1IE	TB1F	Reserved	Reserved	Reserved
R/W	R/W	R/W	R/W	R/W	-	-	-

Table 55. LTCSR1 register description

Bit	Bit name	Function
7	ICIE	Interrupt enable This bit is set and cleared by software. 0: Input capture (IC) interrupt disabled 1: Input capture (IC) interrupt enabled
6	ICF	Input capture flag This bit is set by hardware and cleared by software by reading the LTICR register. Writing to this bit does not change the bit value. 0: No input capture 1: An input capture has occurred *Note: After an MCU reset, software must initialize the ICF bit by reading the LTICR register*
5	TB	Timebase period selection This bit is set and cleared by software. 0: Timebase period = t_{OSC} * 8000 (1ms @ 8 MHz) 1: Timebase period = t_{OSC} * 16000 (2ms @ 8 MHz)
4	TB1IE	Timebase interrupt enable This bit is set and cleared by software. 0: Timebase (TB1) interrupt disabled 1: Timebase (TB1) interrupt enabled

Table 55. LTCSR1 register description (continued)

Bit	Bit name	Function
3	TB1F	Timebase interrupt flag This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect. 0: No counter overflow 1: A counter overflow has occurred
2:0	-	Reserved, must be kept cleared

Lite timer input capture register (LTICR)

LTICR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICR[7:0]							
R							

Table 56. LTICR register description

Bit	Bit name	Function
7:0	ICR[7:0]	Input capture value These bits are read by software and cleared by hardware after a reset. If the ICF bit in the LTCSR is cleared, the value of the 8-bit up-counter will be captured when a rising or falling edge occurs on the LTIC pin.

Table 57. Lite timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
08	LTCSR2 Reset value	0	0	0	0	0	0	TB2IE 0	TB2F 0
09	LTARR Reset value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0
0A	LTCNTR Reset value	CNT7 0	CNT6 0	CNT5 0	CNT4 0	CNT3 0	CNT2 0	CNT1 0	CNT0 0
0B	LTCSR1 Reset value	ICIE 0	ICF x	TB 0	TB1IE 0	TB1F 0	0	0	0
0C	LTICR Reset value	ICR7 0	ICR6 0	ICR5 0	ICR4 0	ICR3 0	ICR2 0	ICR1 0	ICR0 0

11.4 Serial peripheral interface (SPI)

11.4.1 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.

11.4.2 Main features

- Full duplex synchronous transfers (on three lines)
- Simplex synchronous transfers (on two lines)
- Master or slave operation
- 6 master mode frequencies (f_{CPU}/4 max.)
- f_{CPU}/2 max. slave mode frequency (see note)
- $\overline{SS}$ management by software or hardware
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision, master mode fault and overrun flags

Note: In slave mode, continuous transmission is not possible at maximum frequency due to the software overhead for clearing status flags and to initiate the next transmission sequence.

11.4.3 General description

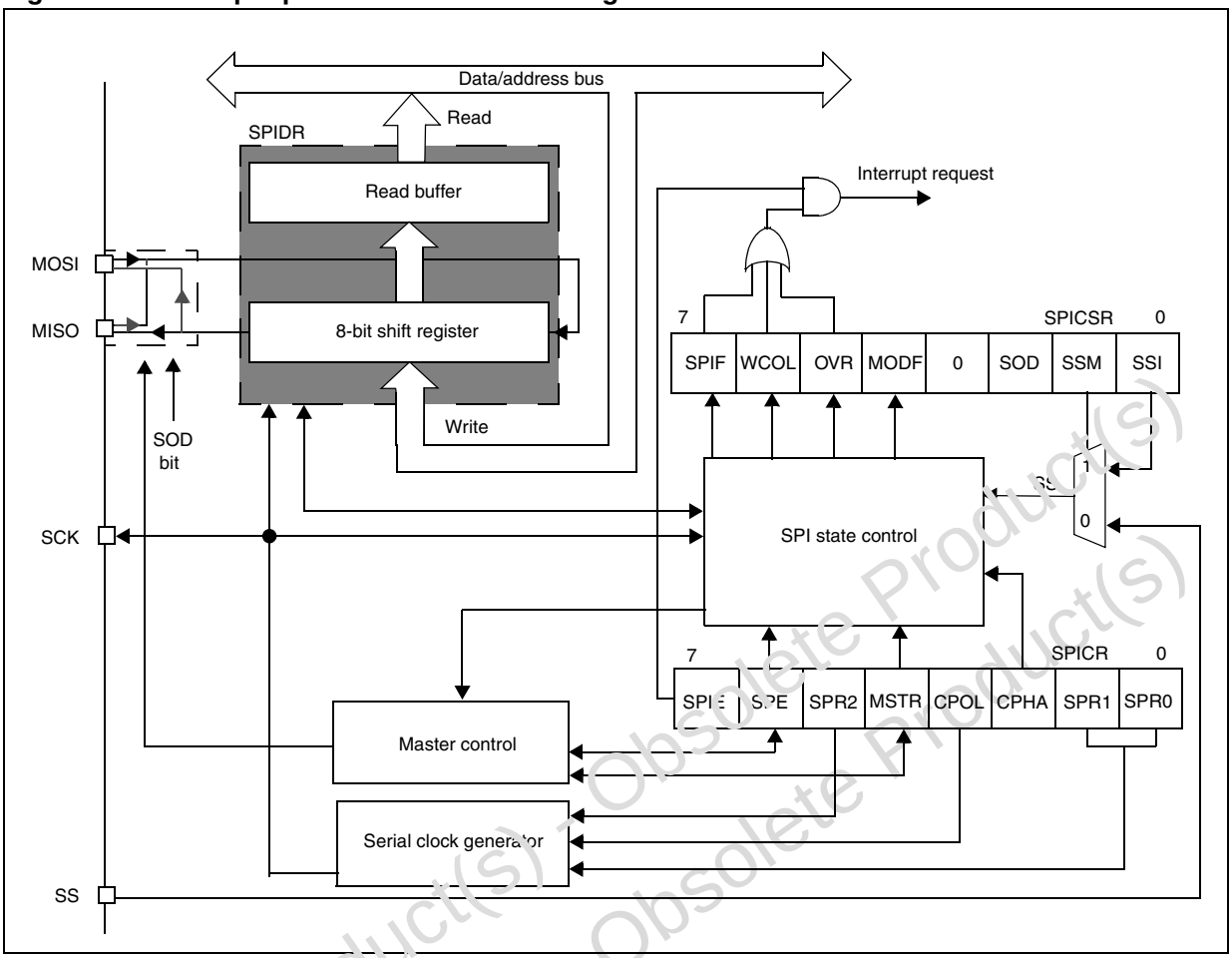
Figure 50: Serial peripheral interface block diagram on page 113 shows the serial peripheral interface (SPI) block diagram. There are three registers:

- SPI control register (SPICR)
- SPI control/status register (SPICSR)
- SPI data register (SPIDR)

The SPI is connected to external devices through four pins:

- MISO: master in/slave out data
- MOSI: master out/slave In data
- SCK: Serial clock out by SPI masters and input by SPI slaves
- $\overline{SS}$: Slave select: This input signal acts as a 'chip select' to let the SPI master communicate with slaves individually and to avoid contention on the data lines. Slave $\overline{SS}$ inputs can be driven by standard I/O ports on the master device.

Figure 50. Serial peripheral interface block diagram



Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 51: Single master/single slave application on page 114*.

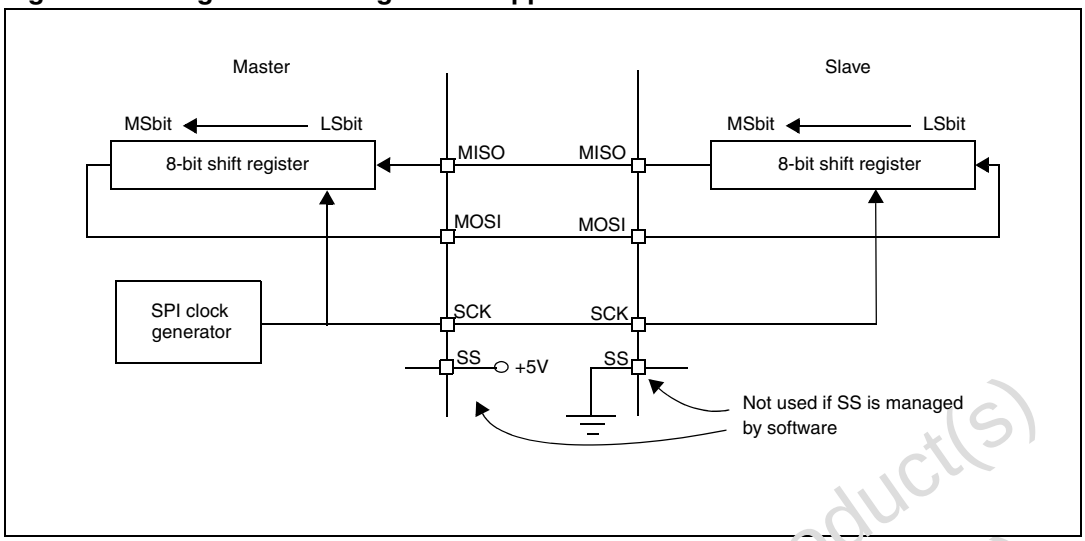
The MOSI pins are connected together and the MISO pins are connected together. In this way data are transferred serially between master and slave (most significant bit first).

The communication is always initiated by the master. 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 communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see *Figure 54: Data clock timing diagram on page 118*) but master and slave must be programmed with the same timing mode.

Figure 51. Single master/single slave application



Slave select management

As an alternative to using the $\overline{SS}$ pin to control the slave select signal, the application can choose to manage the slave select signal by software. This is configured by the SSM bit in the SPICSR register (see *Figure 53: Hardware/software slave select management on page 115*).

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

In master mode:

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

In slave mode:

There are two cases depending on the data/clock timing relationship (see *Figure 52: Generic SS timing diagram on page 115*):

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

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

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

– $\overline{SS}$ internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If $\overline{SS}$ is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see *Write collision error (WCOL) on page 119*).

Figure 52. Generic $\overline{SS}$ timing diagram

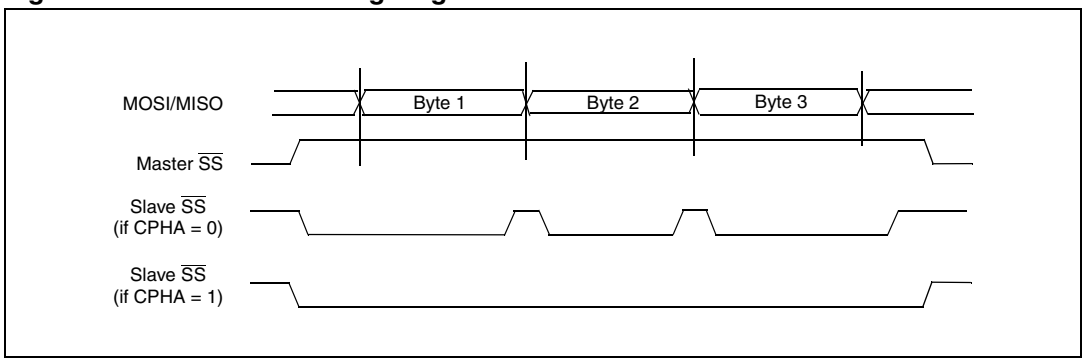
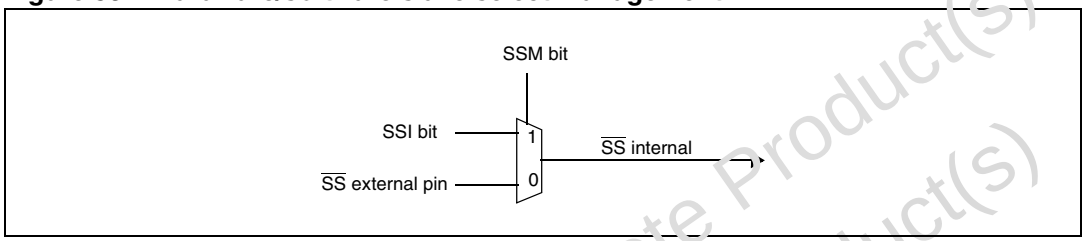


Figure 53. Hardware/software slave select management



Master mode operation

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

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

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

1. Write to the SPICR register:
   - Select the clock frequency by configuring the SPR[2:0] bits
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. *Figure 54: Data clock timing diagram on page 118* shows the four possible configurations.
     Note: The slave must have the same CPOL and CPHA settings as the master
2. Write to the SPICSR register:
   - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the $\overline{SS}$ pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
   - Set the MSTR and SPE bits
     Note: MSTR and SPE bits remain set only if $\overline{SS}$ is high)

Note: If the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account.

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

Master mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

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

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

1. An access to the SPICSR register while the SPIF bit is set
2. A read to the SPIDR register

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

Slave mode operation

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

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see *Figure 54: Data clock timing diagram on page 118*).
     Note: The slave must have the same CPOL and CPHA settings as the master
   - Manage the $\overline{SS}$ pin as described in *Slave select management on page 114* and *Figure 52: Generic SS timing diagram on page 115*. If CPHA = 1 $\overline{SS}$ must be held low continuously. If CPHA = 0 $\overline{SS}$ must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

Slave mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

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

When data transfer is complete:

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

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

1. An access to the SPICSR register while the SPIF bit is set
2. A write or a read to the SPIDR register

Note: 1 While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

2 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 Overrun condition (OVR) on page 119).

11.4.4 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (see *Figure 54: Data clock timing diagram on page 118*).

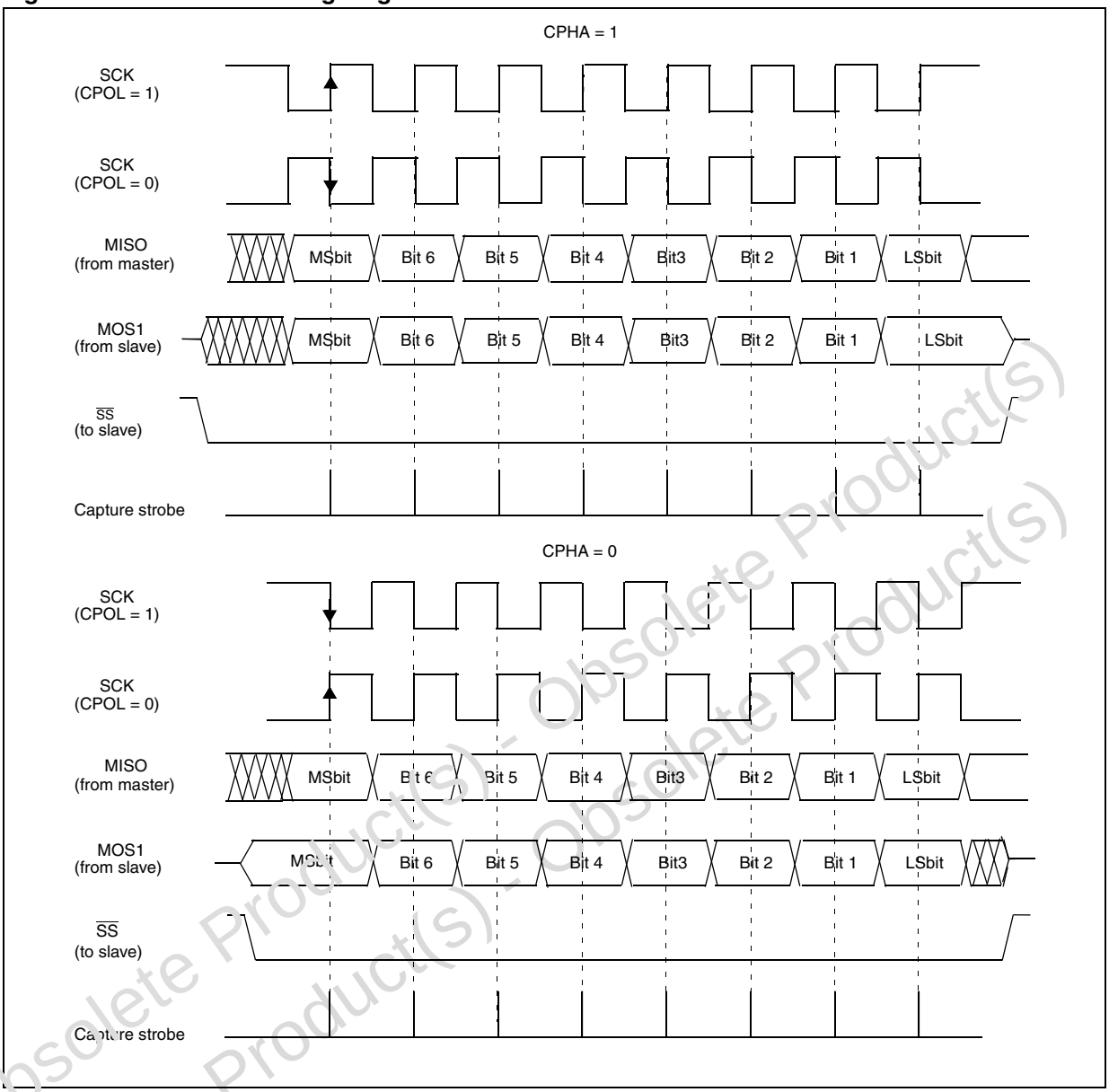
Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

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

Figure 54: Data clock timing diagram on page 118 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 and the MOSI pin are directly connected between the master and the slave device.

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

Figure 54. Data clock timing diagram



1. This figure should not be used as a replacement for parametric information. Refer to *Section 13: Electrical characteristics*.

11.4.5 Error flags

Master mode fault (MODF)

Master mode fault occurs when the master device's $\overline{SS}$ pin is pulled low.

When a master mode fault occurs:

- The MODF bit is set and an SPI interrupt request 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 access to the SPICSR register while the MODF bit is set.
2. A write to the SPICR register.

Note: 1 To avoid any conflicts in an application with multiple slaves, the $\overline{SS}$ pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

2 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.

3 In a slave device, the MODF bit can not be set, but in a multimaster configuration the device can be in slave mode with the MODF bit set.

4 The MODF bit indicates that there might have been a multimaster conflict and allows software to handle this using an interrupt routine and either perform a reset or return to an application default state.

Overrun condition (OVR)

An overrun condition occurs when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an overrun occurs:

- The OVR bit is set and an interrupt request is generated if the SPIE bit is set.

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

The OVR bit is cleared by reading the SPICSR register.

Write collision error (WCOL)

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

Write collisions can occur both in master and slave mode. See also *Slave select management on page 114*.

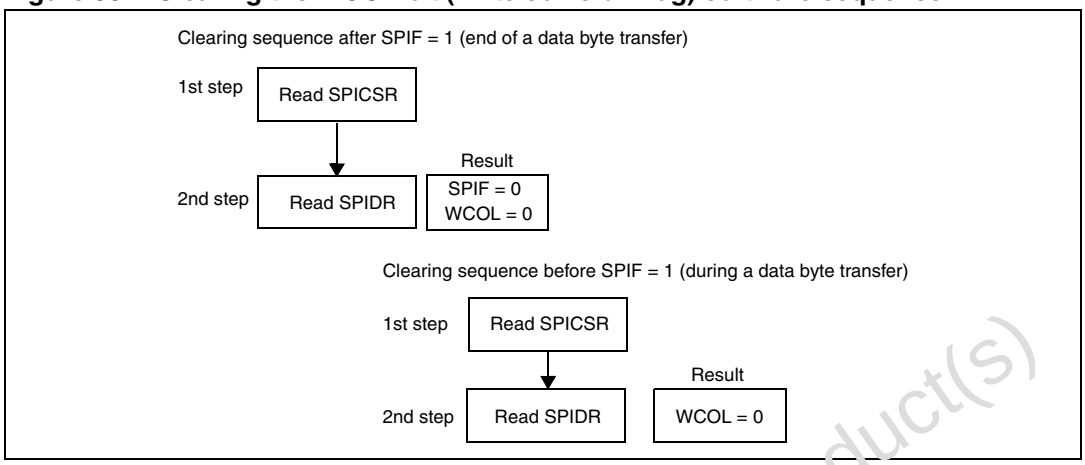
Note: A 'read collision' will never occur since the received data byte is placed in a buffer in which access is always synchronous with the CPU operation.

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

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

Clearing the WCOL bit is done through a software sequence (see *Figure 55*).

Figure 55. Clearing the WCOL bit (write collision flag) software sequence



1. Writing to the SPIDR register instead of reading it does not reset the WCOL bit.

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 a device as the master and four devices as slaves (see *Figure 56: Single master/multiple slave configuration on page 121*).

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 to its SPIDR register.

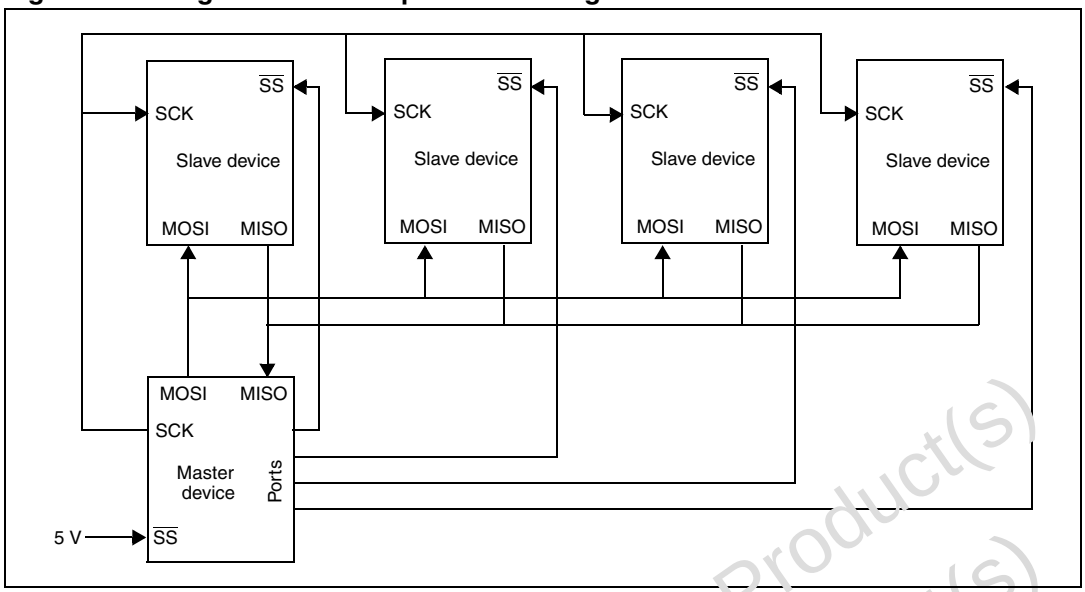
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multimaster system

A multimaster 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 multimaster system is principally handled by the MSTR bit in the SPICR register and the MODF bit in the SPICSR register.

Figure 56. Single master/multiple slave configuration



11.4.6 Low power modes

Table 58. Effect of low power modes on SPI

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 device is woken up by an interrupt with 'exit from Halt mode' capability. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wakeup event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

Using the SPI to wake up the device from halt mode

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

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

Caution: The SPI can wake up the device from halt mode only if the slave select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the device enters halt mode. So, if slave selection is configured as external (see *Slave select management on page 114*), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters halt mode.

11.4.7 Interrupts

Table 59. SPI interrupt control/wake-up capability(1)

Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from halt
SPI end of transfer event	SPIF	SPIE	Yes	Yes
Master mode fault event	MODF			No
Overrun error	OVR			

1. The SPI interrupt events are connected to the same interrupt vector (see *Section 8: Interrupts*). They generate an interrupt if the corresponding enable control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

11.4.8 Register description

SPI control register (SPICR)

SPICR Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 60. SPICR register description

Bit	Bit name	Function
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, MODF = 1 or OVR = 1 in the SPICSR register
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 *Master mode fault (MODF) on page 119*). The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins. 0: I/O pins free for general purpose I/O 1: SPI I/O pin alternate functions enabled
5	SPR2	Divider enable This bit is set and cleared by software and is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate (see bits [1:0] below). 0: Divider by 2 enabled 1: Divider by 2 disabled *Note: The SPR2 bit has no effect in slave mode*

Table 60. SPICR register description (continued)

Bit	Bit name	Function
4	MSTR	Master mode This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 119*). 0: Slave mode 1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.
3	CPOL	Clock polarity This bit is set and cleared by software. This bit determines the idle state of the serial clock. The CPOL bit affects both the master and slave modes. 0: SCK pin has a low level idle state 1: SCK pin has a high level idle state *Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE 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 *Note: The slave must have the same CPOL and CPHA settings as the master.*
1:0	SPR[1:0]	Serial clock frequency These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode: 100: serial clock = $f_{CPU}/4$ 000: serial clock = $f_{CPU}/8$ 001: serial clock = $f_{CPU}/16$ 110: serial clock = $f_{CPU}/32$ 010: serial clock = $f_{CPU}/64$ 011: serial clock = $f_{CPU}/128$ *Note: These 2 bits have no effect in slave mode.*

SPI control/status register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SPIF	WCOL	OVR	MODF	Reserved	SOD	SSM	SSI
R	R	R	R	-	R/W	R/W	R/W

Table 61. SPICSR register description

Bit	Bit name	Function
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 SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register). 0: Data transfer is in progress or the flag has been cleared 1: Data transfer between the device and an external device has been completed *Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.*
6	WCOL	Write collision status This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see *Figure 55: Clearing the WCOL bit (write collision flag) software sequence on page 120*). 0: No write collision occurred 1: A write collision has been detected
5	OVR	SPI overrun error This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (see *Overrun condition (OVR) on page 119*). An interrupt is generated if SPIE = 1 in SPICR register. The OVR bit is cleared by software reading the SPICSR register. 0: No overrun error 1: Overrun error detected
4	MODF	Mode fault flag This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see *Master mode fault (MODF) on page 119*). An SPI interrupt can be generated if SPIE = 1 in the SPICSR register. This bit is cleared by a software sequence (an access to the SPICR register while MODF = 1 followed by a write to the SPICR register). 0: No master mode fault detected 1: A fault in master mode has been detected
3	-	Reserved, must be kept cleared.
2	SOD	SPI output disable This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode/MISO in slave mode). 0: SPI output enabled (if SPE = 1) 1: SPI output disabled

Table 61. SPICSR register description (continued)

Bit	Bit name	Function
1	SSM	$\overline{SS}$ management This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See *Slave select management on page 114*. 0: Hardware management ($\overline{SS}$ managed by external pin) 1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)
0	SSI	$\overline{SS}$ internal mode This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set. 0 : Slave selected 1 : Slave deselected

SPI data I/O register (SPIDR)

SPIDR Reset value: undefined

7	6	5	4	3	2	1	0
D[7:0]							
R/W							

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register initiates transmission/reception of another byte.

Note: 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.

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

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see *Figure 50: Serial peripheral interface block diagram on page 113*).

Table 62. SPI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0031h	SPIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0032h	SPICR Reset value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0033h	SPICSR Reset value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

11.5 10-bit A/D converter (ADC)

11.5.1 Introduction

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

The result of the conversion is stored in a 10-bit data register. The A/D converter is controlled through a control/status register.

11.5.2 Main features

- 10-bit conversion
- Up to 7 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 57: ADC block diagram on page 127*.

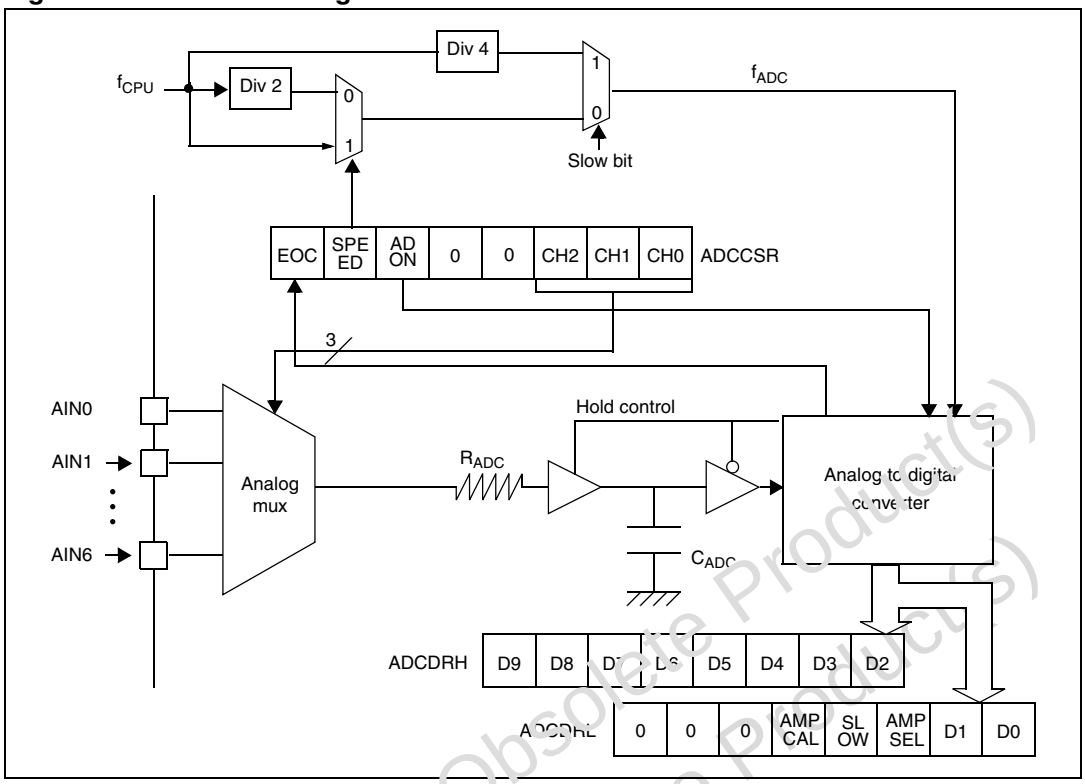
11.5.3 Functional description

Analog power supply

V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to *Section 2: Pin description*) they are internally connected to the V_{DD} and V_{SS} pins.

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

Figure 57. ADC block diagram



Digital A/D conversion result

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

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

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

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the *Section 13: Electrical characteristics*.

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

A/D conversion

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to *Section 10: I/O ports*. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register, select the CH[2:0] bits to assign the analog channel to convert.

ADC conversion mode

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

When a conversion is complete:

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

A read to the ADCDRH resets the EOC bit.

To read the 10 bits, perform the following steps:

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

To read only 8 bits, perform the following steps:

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

11.5.4 Low-power modes

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

Table 63. Effect of low power modes on the A/D converter

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 stabilization time t_{STAB} (*Section 13: Electrical characteristics*) before accurate conversions can be performed.

11.5.5 Interrupts

None.

11.5.6 Register description

Control/status register (ADCCSR)

ADCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EOC	SPEED	ADON	Reserved	Reserved	CH[2:0]		
R	R/W	R/W	-	-	R/W		

Table 64. ADCCSR register description

Bit	Bit name	Function
7	EOC	End of conversion This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete 1: Conversion complete
6	SPEED	ADC clock selection This bit is set and cleared by software. It is used together with the SLOW bit to configure the ADC clock speed. Refer to *Table 67: ADC clock configuration on page 130* in the SLOW bit description (ADCDRL register).
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
4:3	-	Reserved, must be kept cleared.
2:0	CH[2:0]	Channel selection These bits are set and cleared by software. They select the analog input to convert: 000: channel pin = AIN0 001: channel pin = AIN1 010: channel pin = AIN2 011: channel pin = AIN3 100: channel pin = AIN4 101: channel pin = AIN5 110: channel pin = AIN6 *Note: The number of channels is device dependent. Refer to the Section 2: Pin description.*

Data register high (ADCDRH)

ADCDRH Reset value: xxxx xxxx (xxh)

7	6	5	4	3	2	1	0
D[9:2]							
R							

Table 65. ADCDRH register description

Bit	Bit name	Function
7:0	D[9:2]	MSB of analog converted value

AMP control/data register low (ADCDRL)

ADCDRL Reset value: 0000 00xx (0xh)

7	6	5	4	3	2	1	0
0	0	0	Reserved	SLOW	Reserved	D[1:0]	
-	-	-	-	R/W	-	R/W	

Table 66. ADCDRL register description

Bit	Bit name	Function
7:5	-	Reserved, must be kept cleared
4	-	Reserved, must be kept cleared
3	SLOW	Slow mode This bit is set and cleared by software. It is used together with the SPEED bit in the ADCCSR register to configure the ADC clock speed as shown in *Table 67: ADC clock configuration on page 130*
2	-	Reserved, must be kept cleared
1:0	D[1:0]	LSB of converted analog value

Table 67. ADC clock configuration(1)

f_{ADC}	SLOW	SPEED
$f_{CPU}/2$	0	0
f_{CPU}	0	1
$f_{CPU}/4$	1	x

1. Max f_{ADC} allowed = 4 MHz (see *Section 13.11: 10-bit ADC characteristics on page 174*)

Table 68. ADC register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0034h	ADCCSR Reset value	EOC 0	SPEED 0	ADON 0	0 0	0 0	CH2 0	CH1 0	CH0 0
0035h	ADCDRH Reset value	D9 x	D8 x	D7 x	D6 x	D5 x	D4 x	D3 x	D2 x
0036h	ADCDRL Reset value	0 0	0 0	0 0	AMPCAL 0	SLOW 0	AMPSEL 0	D1 x	D0 x

12 Instruction set

12.1 ST7 addressing modes

The ST7 core features 17 different addressing modes which can be classified in 7 main groups (see *Table 69*).

Table 69. CPU addressing mode 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,#3

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 70. CPU addressing mode overview

Mode			Syntax	Destination/source	Pointer address	Pointer size	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

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

12.1.1 Inherent

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

Table 71. Inherent instructions

Inherent instruction	Function
NOP	No operation
TRAP	S/W interrupt
WFI	Wait for interrupt (low power mode)
HALT	Halt oscillator (lowest power mode)
RET	Subroutine return
IRET	Interrupt subroutine 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

12.1.2 Immediate

Immediate instructions have 2 bytes, the first byte contains the opcode, the second byte contains the operand value.

Table 72. Immediate instructions

Immediate instruction	Function
LD	Load
CP	Compare
BCP	Bit compare
AND, OR, XOR	Logical operations
ADC, ADD, SUB, SBC	Arithmetic operations

12.1.3 Direct

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

The direct addressing mode consists of two sub-modes:

- Direct instructions (short)

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

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

12.1.4 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 two bytes after the opcode.

12.1.5 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.

12.1.6 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 73. 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/substraction operations
BCP	Bit compare

Table 74. Short instructions and functions

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

12.1.7 Relative mode (direct, indirect)

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

Table 75. Relative mode instructions (direct and indirect)

Available relative direct/indirect instructions	Function
JRxx	Conditional jump
CALLR	Call relative

The relative addressing mode consists of two sub-modes:

- Relative (direct)
  The offset follows the opcode.
- Relative (indirect)
  The offset is defined in memory, which address follows the opcode.

12.2 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:

Table 76. Instruction groups

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			
Condition code flag modification	SIM	RIM	SCF	RCF			

12.2.1 Using a prebyte

The instructions are described with one to four opcodes.

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

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC Opcode

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

These prebytes enable instructions in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instructions in X or the instructions 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

12.2.2 Illegal opcode reset

In order to provide enhanced robustness to the device against unexpected behavior, a system of illegal opcode detection is implemented. If a code to be executed does not correspond to any opcode or prebyte value, a reset is generated. This, combined with the watchdog, allows the detection and recovery from an unexpected fault or interference.

Note: A valid prebyte associated with a valid opcode forming an unauthorized combination does not generate a reset.

Table 77. Instruction set overview

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 >=							

Table 77. Instruction set overview (continued)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRUGT	Jump if (C + Z = 0)	Unsigned >							
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			

13 Electrical characteristics

13.1 Parameter conditions

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

13.1.1 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).

13.1.2 Typical values

Unless otherwise specified, typical data are based on T_A = 25 °C, V_{DD} = 5 V (for the 4.5 V ≤V_{DD} ≤5.5 V voltage range) and V_{DD} = 3.3 V (for the 3 V ≤V_{DD} ≤3.6 V voltage range). They are given only as design guidelines and are not tested.

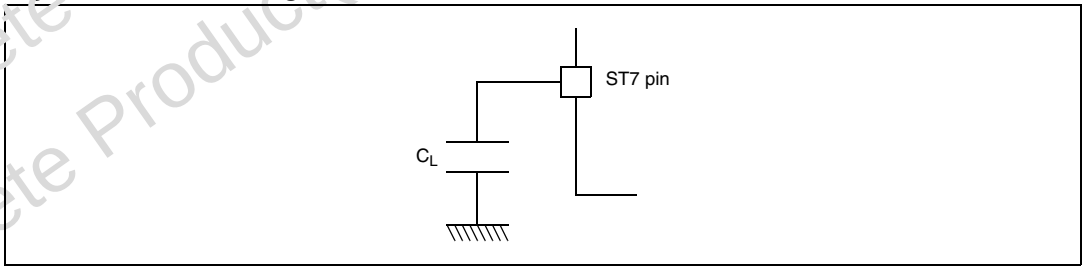
13.1.3 Typical curves

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

13.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 58*.

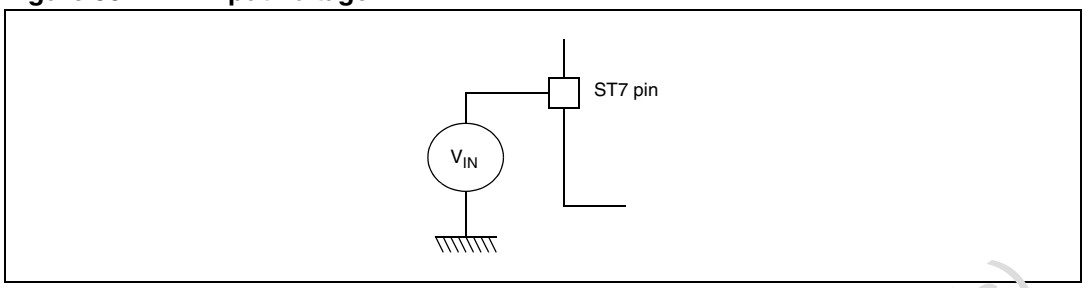
Figure 58. Pin loading conditions



13.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 59*.

Figure 59. Pin input voltage



13.2 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.

13.2.1 Voltage characteristics

Table 78. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	7.0	V
V_{IN}	Input voltage on any pin(1)(2)	V_{SS} - 0.3 to V_{DD} + 0.3	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	See *Section 13.7.3: Absolute maximum ratings (electrical sensitivity) on page 160*	
$V_{ESD(MM)}$	Electrostatic discharge voltage (machine model)		

1. Directly connecting the $\overline{\text{RESET}}$ 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 must be made through a pull-up or pull-down resistor (typical: 4.7 kΩ for $\overline{\text{RESET}}$, 10 kΩ 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. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$.

13.2.2 Current characteristics

Table 79. Current characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source)(1)	75	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink)(1)	150	
I_{IO}	Output current sunk by any standard I/O and control pin	20	
	Output current sunk by any high sink I/O pin	40	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$(2)(3)	Injected current on ISPSEL pin	± 5	
	Injected current on $\overline{RESET}$ pin	± 5	
	Injected current on OSC1 and OSC2 pins	± 5	
	Injected current on PB0 pin(4)	± 5	
	Injected current on any other pin(5)	± 5	
$\Sigma I_{INJ(PIN)}$(2)	Total injected current (sum of all I/O and control pins)(5)	± 20	

1. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$.
3. 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 1.6 mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
4. No negative current injection allowed on PB0 pin.
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 characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

13.3 Operating conditions

13.3.1 General operating conditions

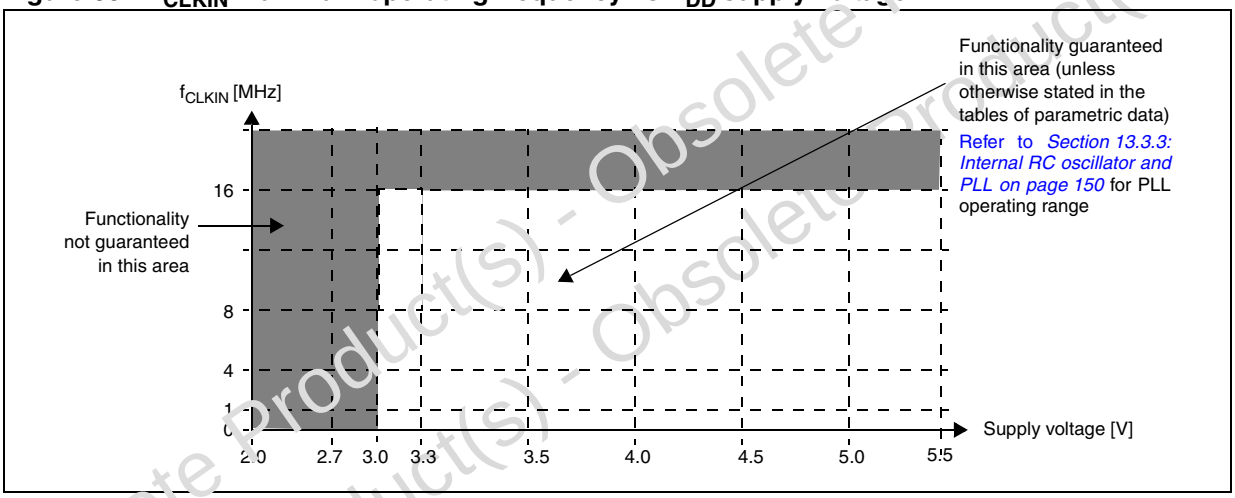
T_A = -40 to +125 °C unless otherwise specified.

Table 80. General operating conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	Supply voltage	f_{OSC} = 16 MHz max T_A = -40°C to T_A max	3.0	5.5	V
f_{CLKIN}	External clock frequency on CLKIN pin	$V_{DD} \geq 3$ V	0	16	MHz
T_A	Ambient temperature range	A suffix version	-40	+85	°C
		B suffix version(1)	-40	+105	
		C suffix version	-40	+125	

1. For ROM devices only

Figure 60. f_{CLKIN} maximum operating frequency vs V_{DD} supply voltage



1. For further information on clock management block diagram for f_{CLKIN} description, refer to *Figure 12: Clock management block diagram on page 41*.

The RC oscillator and PLL characteristics are temperature-dependent.

Table 81. Operating conditions (tested for T_A = -40 to +125 °C) @ V_{DD} = 4.5 to 5.5 V

Symbol	Parameter	Conditions	Flash			ROM			Unit
			Min	Typ	Max	Min	Typ	Max	
f_{RC}(1)	Internal RC oscillator frequency	RCCR = FF (reset value), T_A = 25 °C, V_{DD} = 5 V		700			630		kHz
		RCCR = RCCR0(2), T_A = -40 to 125 °C, V_{DD} = 5 V	950	1000	1070	TBD	1000	TBD	
ACC_{RC}	RC resolution	V_{DD} = 5 V	-1		+1	TBD		TBD	%
	Accuracy of internal RC oscillator with RCCR = RCCR0(2)(3)	T_A = -40 to +125 °C, V_{DD} = 5 V	-5		+7	TBD		TBD	
		T_A = -40 to +125 °C, V_{DD} = 4.5 V to 5.5 V(4)	-6.5		+8.5	TBD		TBD	

1. If the RC oscillator clock is selected, to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.
2. See *Section 7.1: Internal RC oscillator adjustment on page 37*
3. Minimum value is obtained for hot temperature and maximum value is obtained for cold temperature
4. Data based on characterization results, not tested in production

Table 82. Operating conditions (tested for T_A = -40 to +125 °C) @ V_{DD} = 4.5 to 5.5 V

Symbol	Parameter	Conditions	Flash and ROM			Unit
			Min	Typ	Max	
$I_{DD(RC)}$	RC oscillator current consumption	T_A = 25 °C, V_{DD} = 5 V		600(1)(2)		µA
$t_{su(RC)}$	RC oscillator setup time				10(3)	µs
f_{PLL}	x4 PLL input clock			1		MHz
t_{LOCK}	PLL lock time(4)			2		ms
t_{STAB}	PLL stabilization time(4)			4		
ACC_{PLL}	x4 PLL accuracy	f_{CLKIN}/2 or f_{RC} = 1 MHz @ T_A = -40 to +125 °C		0.2(5)		%
J_{PLL}	PLL jitter ($\Delta f_{CPU}/f_{CPU}$)			1(6)		
$I_{DD(PLL)}$	PLL current consumption	T_A = 25 °C		600(2)		µA

1. Measurement made with RC calibrated at 1 MHz
2. Data based on characterization results, not tested in production
3. See *Section 7.1: Internal RC oscillator adjustment on page 37*
4. After the LOCKED bit is set ACC_{PLL} is maximum 10% until t_{STAB} has elapsed. See *Figure 11: PLL output frequency timing diagram on page 39*.
5. Averaged over a 4ms period. After the LOCKED bit is set, a period of t_{STAB} is required to reach ACC_{PLL} accuracy
6. Guaranteed by design

Figure 61. Typical accuracy with RCCR = RCCR0 vs V_{DD} = 4.5 to 5.5 V and temperature

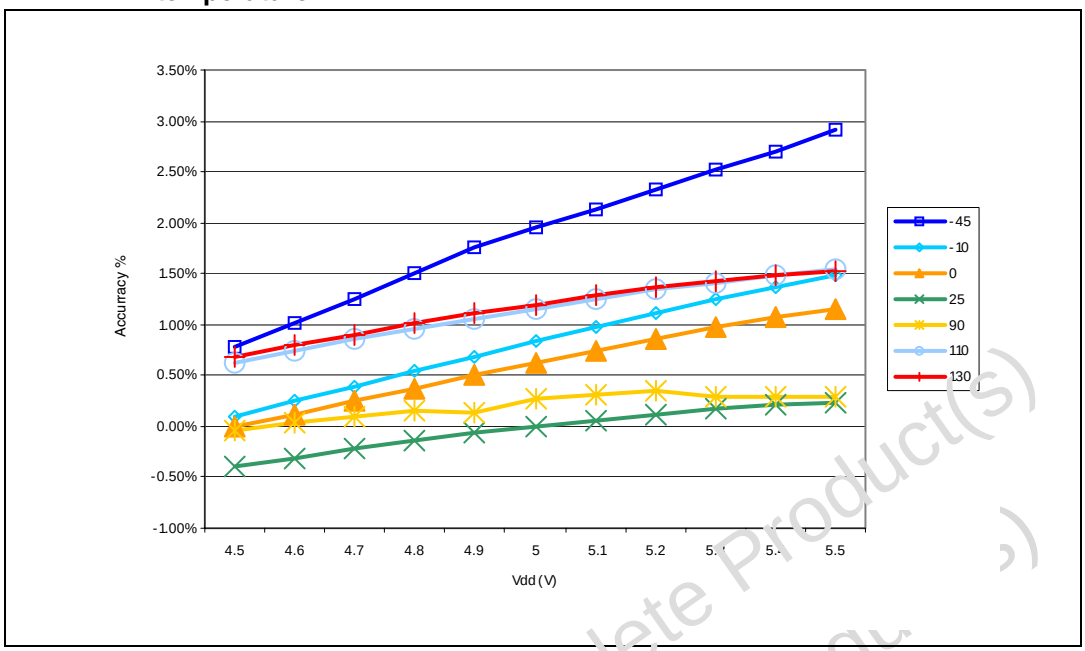


Figure 62. f_{RC} vs V_{DD} and temperature for calibrated RCCR0

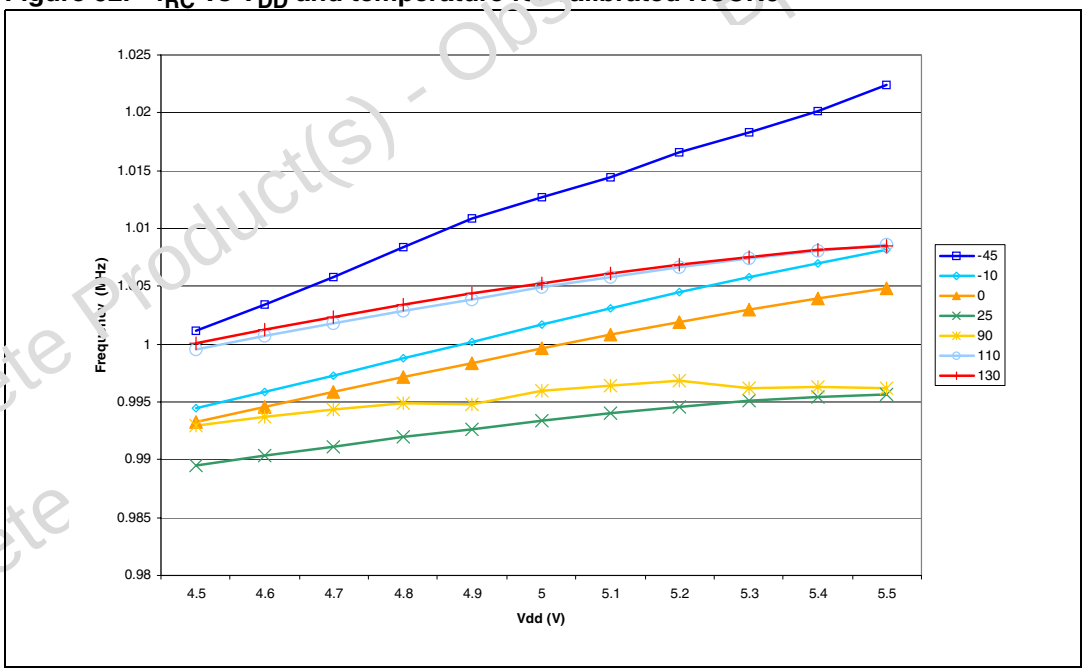


Table 83. Operating conditions (tested for T_A = -40 to +125 °C) @ V_{DD} = 3.0 to 3.6 V

Symbol	Parameter	Conditions	Flash			ROM			Unit
			Min	Typ	Max	Min	Typ	Max	
f_{RC}(1)	Internal RC oscillator frequency	RCCR = FF (reset value), T_A = 25 °C, V_{DD} = 3.3 V		630			630		kHz
		RCCR = RCCR(2), T_A = -40 to +125 °C, V_{DD} = 3.3 V	950	1000	1070	TBD	1000	TBD	
ACC_{RC}	RC resolution	V_{DD} = 3.3 V	-1		+1	TBD		TBD	%
	Accuracy of internal RC oscillator with RCCR = RCCR0(2)(3)	T_A = -40 to +125 °C, V_{DD} = 3.3 V	-5		+7	TBD		TBD	
		T_A = -40 to +125 °C, V_{DD} = 3.0 V to 3.6 V(4)	-6		+8	TBD		TBD	

1. If the RC oscillator clock is selected, to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.
2. See *Section 7.1: Internal RC oscillator adjustment on page 37*.
3. Minimum value is obtained for hot temperature and max value is obtained for cold temperature.
4. Data based on characterization results, not tested in production.

Table 84. Operating conditions (tested for T_A = -40 to +125 °C) @ V_{DD} = 3.0 to 3.6 V(1)

	Parameter(1)	Conditions	Flash and ROM			
			Min	Typ	Max	
$I_{DD(RC)}$	RC oscillator current consumption	T_A = 25 °C, V_{DD} = 3.3 V		400(2)		µA
$t_{su(RC)}$	RC oscillator setup time				10(2)	µs
f_{PLL}	x4 PLL input clock			0.7		MHz
t_{LOCK}	PLL lock time(3)			2		ms
t_{STAB}	PLL stabilization time(3)			4		
ACC_{PLL}	x4 PLL accuracy	f_{CLKIN}/2 or f_{RC} = 1 MHz @ T_A = -40 to +125 °C		0.2(4)		%
JIT_{PLL}	PLL jitter ($\Delta f_{CPU}/f_{CPU}$)			1(5)		
$I_{DD(PLL)}$	PLL current consumption	T_A = 25 °C		190(1)		µA

1. Data based on characterization results, not tested in production.
2. Measurement made with RC calibrated at 1 MHz.
3. After the LOCKED bit is set ACC_{PLL} is maximum 10% until t_{STAB} has elapsed. See *Figure 11: PLL output frequency timing diagram on page 39*.
4. Averaged over a 4ms period. After the LOCKED bit is set, a period of t_{STAB} is required to reach ACC_{PLL} accuracy.
5. Guaranteed by design.

Figure 63. Typical accuracy with RCCR = RCCR1 vs V_{DD} = 3 to 3.6 V and temperature

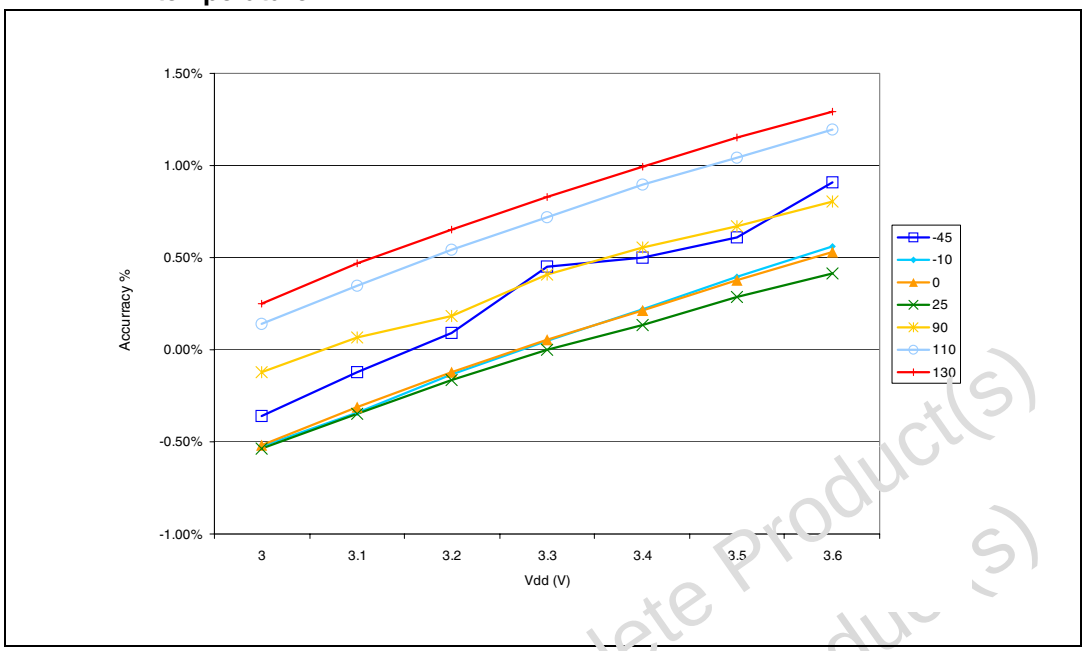


Figure 64. f_{RC} vs V_{DD} and temperature for calibrated RCCR1

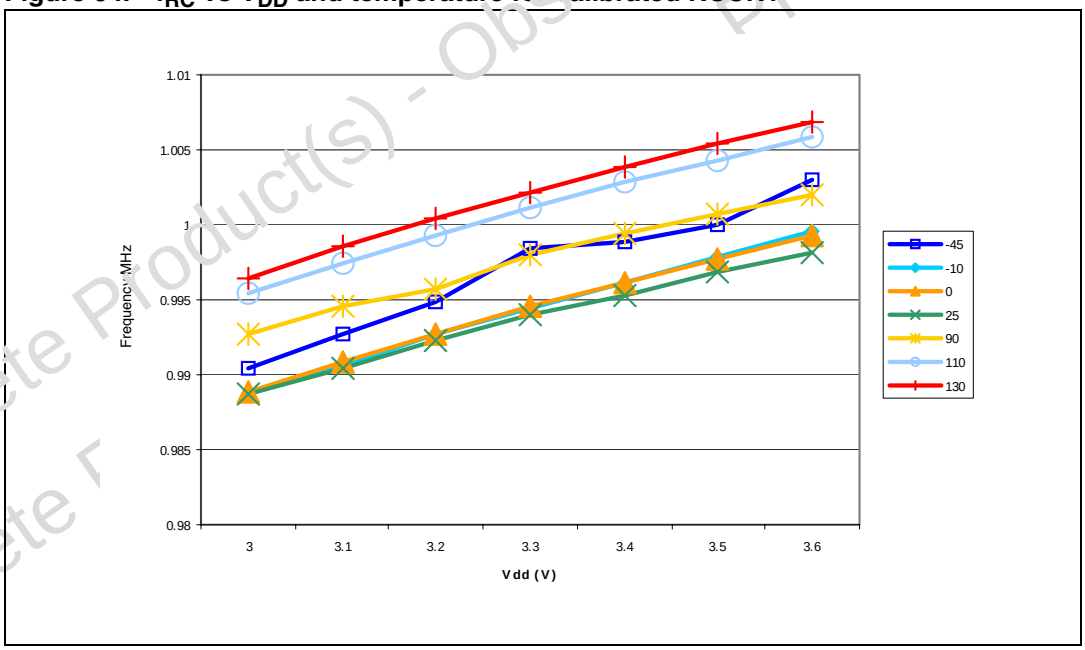
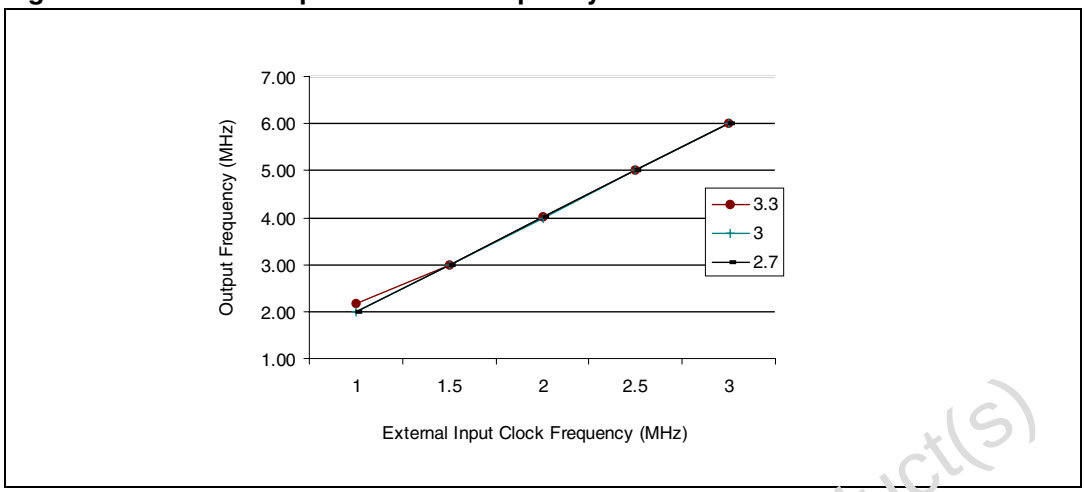
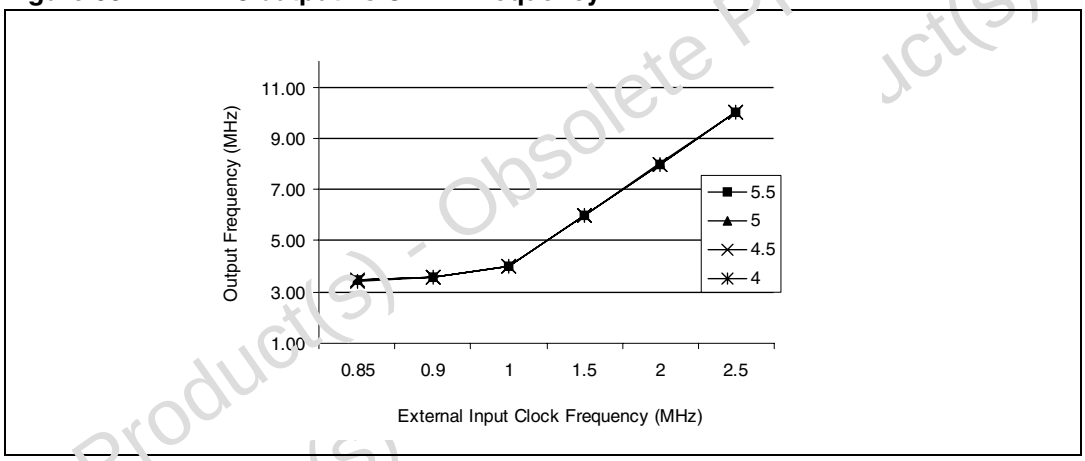


Figure 65. PLL x 4 output vs CLKIN frequency



1. $f_{OSC} = f_{CLKIN}/2*PLL4$

Figure 66. PLL x 8 output vs CLKIN frequency



1. $f_{OSC} = f_{CLKIN}/2*PLL8$

13.3.2 Operating conditions with low voltage detector (LVD)

T_A = -40 to +125 °C, unless otherwise specified.

Table 85. Operating conditions with low voltage detector

Symbol	Parameter	Conditions(1)	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)		3.80(2)	4.20	4.60	V
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)		3.70	4.00	4.35(2)	
V_{hys}	LVD voltage threshold hysteresis	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$		200		mV
Vt_{POR}	V_{DD} rise time rate(3)(4)		0.02(2)		100(2)	ns/V
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}	Not detected by the LVD			50(5)	ns
$I_{DD(LVD)}$	LVD current consumption			200		µA

1. LVD functionality guaranteed only within the V_{DD} operating range specified in *Section 13.3.1: General operating conditions on page 144*.
2. Not tested in production.
3. Not tested in production. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. When the V_{DD} slope is outside these values, the LVD may not ensure a proper reset of the MCU.
4. Use of LVD with capacitive power supply: With this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0 V to ensure optimum restart conditions. Refer to circuit example in *Figure 95: RESET pin protection when LVD is disabled on page 170*.
5. Based on design simulation.

13.3.3 Internal RC oscillator and PLL

The ST7 internal clock can be supplied by an internal RC oscillator and PLL (selectable by option byte).

Table 86. Internal RC oscillator and PLL

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{DD(RC)}$	Internal RC oscillator operating voltage	Refer to operating range of V_{DD} with T_A, *Section 13.3.1: General operating conditions on page 144*	3.0		5.5	V
$V_{DD(x4PLL)}$	x4 PLL operating voltage(1)		3.0		3.6	
$V_{DD(x8PLL)}$	x8 PLL operating voltage		3.6		5.5	
$t_{STARTUP}$	PLL startup time			60		PLL input clock (f_{PLL}) cycles

1. x4 PLL option only applicable on Flash devices.

13.4 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 obtain the total device consumption, the two current values must be added (except for halt mode for which the clock is stopped).

13.4.1 Supply current

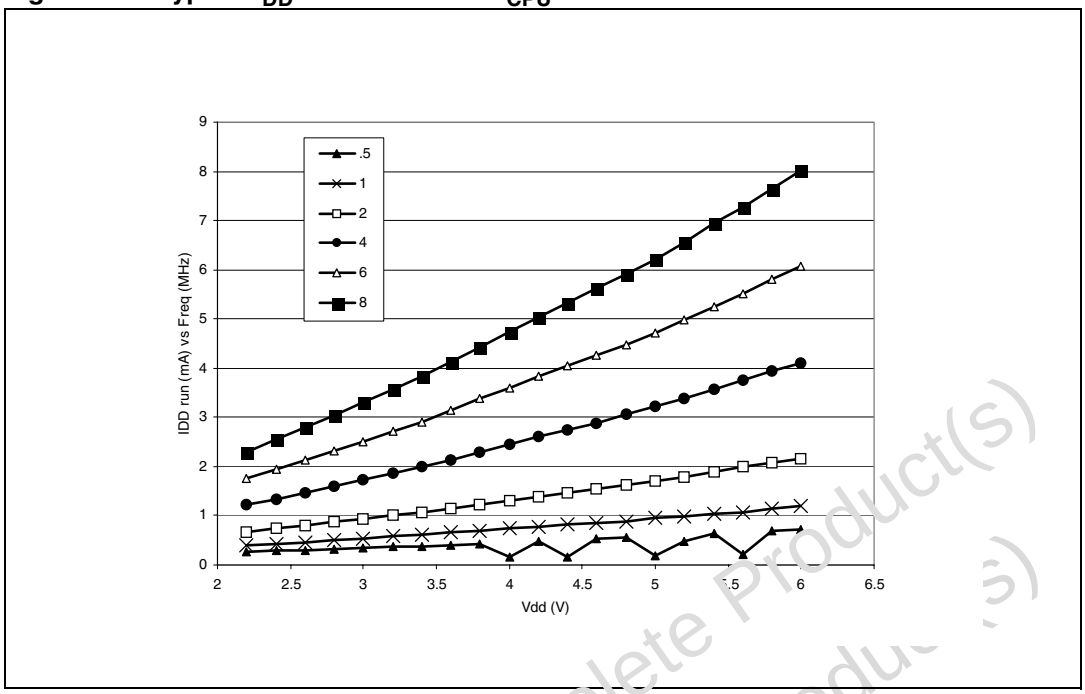
T_A = -40 to +125°C, unless otherwise specified.

Table 87. Supply current

Symbol	Parameter	Conditions		Typ	Max	Unit
I_{DD}	Supply current in run mode	V_{DD} = 5.5 V	f_{CPU} = 8 MHz(1)	7	9	mA
	Supply current in wait mode		f_{CPU} = 8 MHz(2)	3	3.6	
	Supply current in slow mode		f_{CPU} = 250 kHz(3)	0.7	0.9	
	Supply current in slow wait mode		f_{CPU} = 250 kHz(4)	0.5	0.8	
	Supply current in halt mode(5)		-40°C ≤T_A ≤+125°C	1	6	µA
	Supply current in AWUFH mode(6)(7)		-40°C ≤T_A ≤+125°C	60	90	
	Supply current in active halt mode		-40°C ≤T_A ≤+125°C	0.7		mA

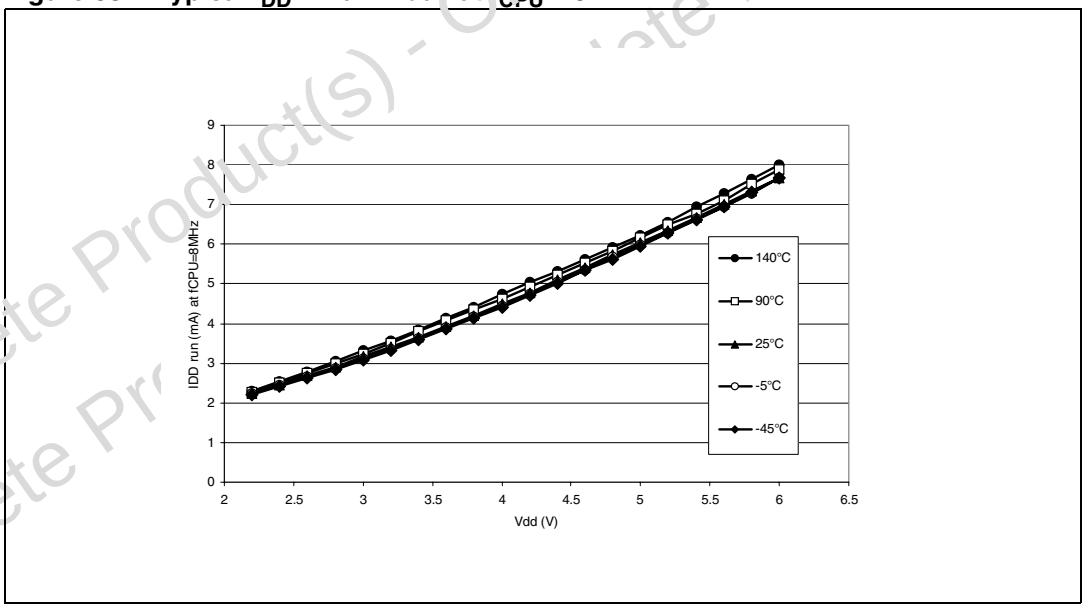
1. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
2. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
3. 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 (CLKIN) driven by external square wave, 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 (CLKIN) driven by external square wave, LVD disabled.
5. All I/O pins in output mode with a static value at V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max and f_{CPU} max.
6. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load). Data tested in production at V_{DD} max. and f_{CPU} max.
7. This consumption refers to the Halt period only and not the associated run period which is software dependent.

Figure 67. Typical I_{DD} in run mode vs f_{CPU}



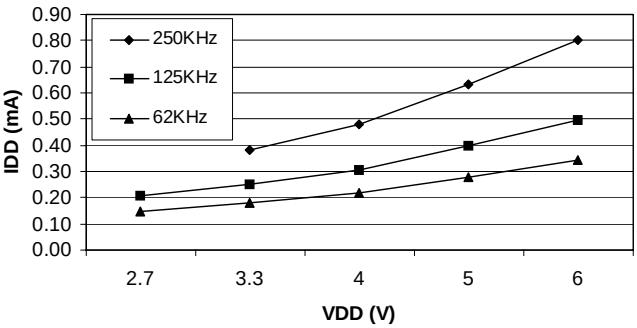
1. Graph displays data beyond the normal operating range of 3 V to 5.5 V

Figure 68. Typical I_{DD} in run mode at f_{CPU} = 8 MHz



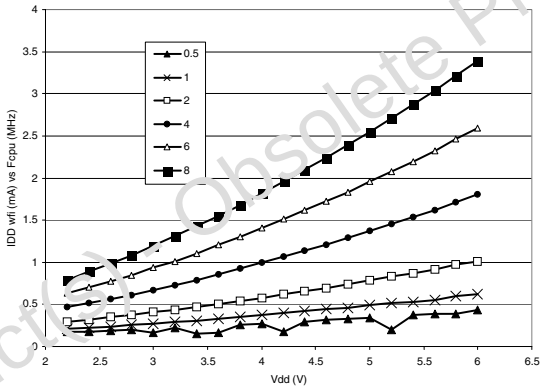
1. Graph displays data beyond the normal operating range of 3 V - 5.5 V

Figure 69. Typical I_{DD} in slow mode vs f_{CPU}



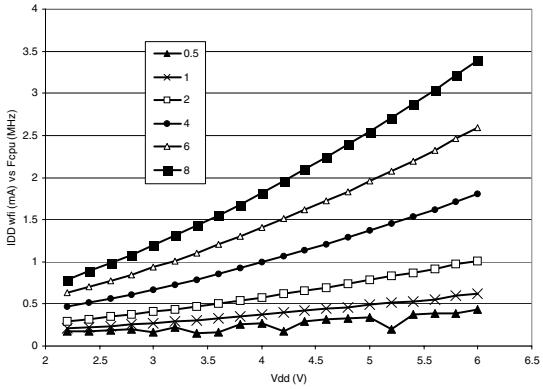
1. Graph displays data beyond the normal operating range of 3 V - 5.5 V

Figure 70. Typical I_{DD} in wait mode vs f_{CPU}



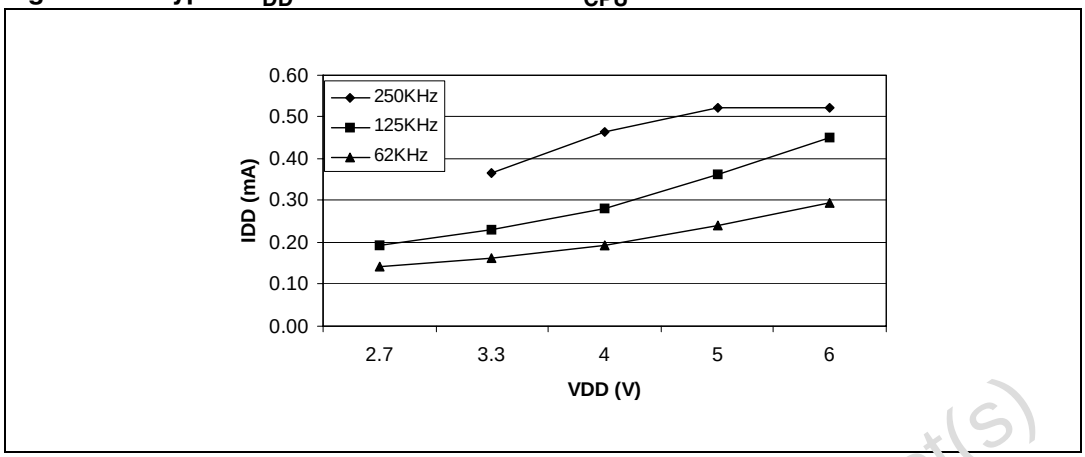
1. Graph displays data beyond the normal operating range of 3 V - 5.5 V

Figure 71. Typical I_{DD} in wait mode at f_{CPU} = 8 MHz



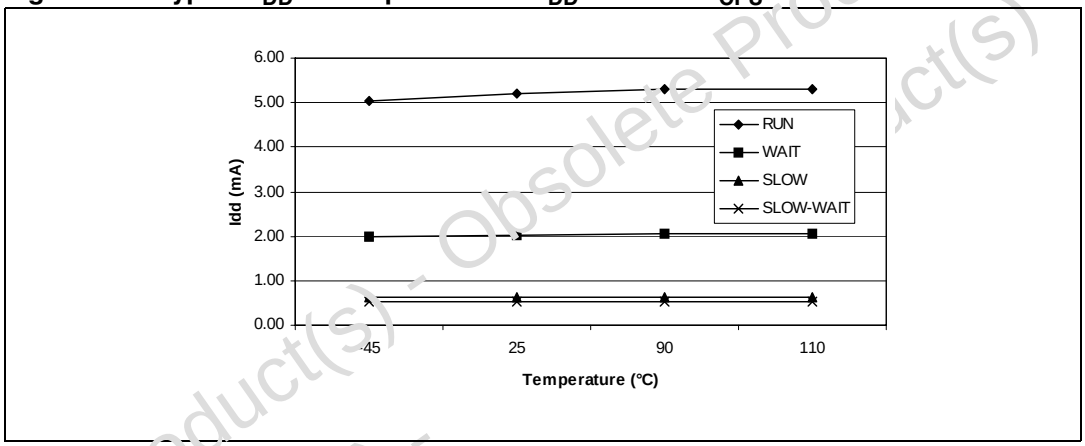
1. Graph displays data beyond the normal operating range of 3 V - 5.5 V

Figure 72. Typical I_{DD} in slow-wait mode vs f_{CPU}



1. Graph displays data beyond the normal operating range of 3 V - 5.5 V

Figure 73. Typical I_{DD} vs temperature at V_{DD} = 5 V and f_{CPU} = 8 MHz



13.4.2 On-chip peripherals

Table 88. On-chip peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(AT)}$	12-bit autoreload timer supply current(1)	f_{CPU} = 4 MHz	V_{DD} = 3.3 V	150	µA
		f_{CPU} = 8 MHz	V_{DD} = 5 V	1000	
$I_{DD(SPI)}$	SPI supply current(2)	f_{CPU} = 4 MHz	V_{DD} = 3.3 V	50	
		f_{CPU} = 8 MHz	V_{DD} = 5 V	200	
$I_{DD(ADC)}$	ADC supply current when converting(3)	f_{ADC} = 4 MHz	V_{DD} = 3.3 V	250	
			V_{DD} = 5 V	1100	

1. Data based on a differential I_{DD} measurement between reset configuration (timer stopped) and a timer running in PWM mode at f_{CPU} = 8 MHz.
2. Data based on a differential I_{DD} measurement between reset configuration and a permanent SPI master communication (data sent equal to 55h).
3. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

13.5 Clock and timing characteristics

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

13.5.1 General timings

Table 89. General timings

Symbol	Parameter[1]	Conditions	Min	Typ[2]	Max	Unit
$t_{c(INST)}$	Instruction cycle time	f_{CPU} = 8 MHz	2	3	12	t_{CPU}
			250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time[3] $t_{v(IT)} = \Delta t_{c(INST)} + 10$		10		22	t_{CPU}
			1.25		2.75	µs

1. Guaranteed by design. Not tested in production.
2. Data based on typical application software.
3. Time measured between interrupt event and interrupt vector fetch. $Dt_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

13.5.2 Auto wake-up from halt oscillator (AWU)

Table 90. Auto wake-up from halt oscillator (AWU)[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{AWU}	AWU oscillator frequency		50	125	250	kHz
t_{RCSRT}	AWU oscillator startup time				50	µs

1. Guaranteed by design. Not tested in production.

13.5.3 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with ten different crystal/ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

Table 91. Oscillator parameters (ROM devices)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CrOSC}	Crystal oscillator frequency[1]		2		16	MHz
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)		See *Table 92: Oscillator parameters (Flash devices)*			pF

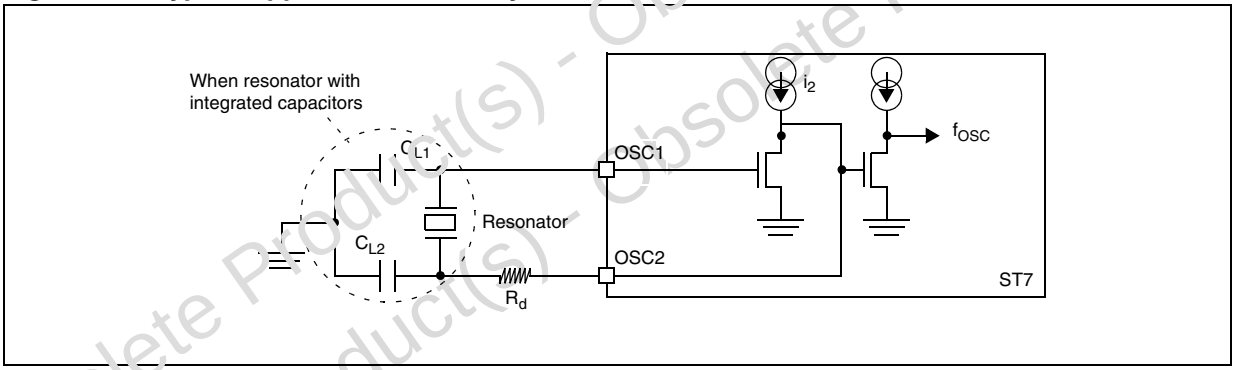
1. When PLL is used, please refer to *Table 86: Internal RC oscillator and PLL on page 150* and to *Section 7: Supply, reset and clock management* (f_{CrOSC} min. is 8 MHz with PLL).

Table 92. Oscillator parameters (Flash devices)

Supplier	f_{CrOSC} (MHz)	Typical ceramic resonators(1)		CL1(2) (pF)	CL2(2) (pF)	Rd (Ω)	Supply voltage range (V)	Temperature range (°C)
		Type(3)	Reference					
Murata	1	SMD	CSBFB1M00J58-R0	220	220	2.2 k	3.6 V to 5.5 V	-40 to +125
		LEAD	CSBLA1M00J58-B0					
	2	SMD	CSTCC2M00G56Z-R0	(47)	(47)	0	3.0 V to 5.5 V	
	4	SMD	CSTCR4M00G53Z-R0	(15)	(15)	0		
		LEAD	CSTLS4M00G53Z-B0					
	8	SMD	CSTCE8M00G52Z-R0	(10)	(10)	0		
		LEAD	CSTLS8M00G53Z-B0	(15)	(15)	0		
	12	SMD	CSTCE12M0G52Z-R0	(10)	(10)	0	3.6 V to 5.5 V	
	16	SMD	CSTCE16M0V51Z-R0	(5)	(5)	0		
		LEAD	CSTLS16M0X51Z-B0					

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult www.murata.com
2. () means load capacitor built in resonator
3. SMD = [-R0: plastic tape package (∅ =180 mm)] LEAD = [-B0: bulk]

Figure 74. Typical application with a crystal or ceramic resonator



13.6 Memory characteristics

13.6.1 RAM and hardware registers

T_A = -40 to +125 °C, unless otherwise specified.

Table 93. RAM and hardware registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode(1)	Halt mode (or reset)	1.6			V

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in halt mode or under reset) or in hardware registers (only in halt mode). Not tested in production.

13.6.2 Flash program memory

T_A = -40 to +125°C, unless otherwise specified.

Table 94. Characteristics of dual voltage HDFlash memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating voltage for Flash write/erase	Refer to operating range of V_{DD} with T_A, *Section 13.3.1: General operating conditions on page 144*	3.0		5.5	V
t_{prog}	Programming time for 1~32 bytes(1)	T_A = -40 to +85 °C		5	10	ms
	Programming time for 1.5 Kbytes	T_A = 25 °C		0.24	0.48	s
t_{RET}(2)	Data retention	T_A = 55 °C(3)	20			years
N_{RW}	Write/erase cycles	T_{PROG} = 25 °C	1K			cycles
		T_{PROG} = 85 °C	300			
I_{DD}	Supply current	Read/write/erase modes f_{CPU} = 8 MHz, V_{DD} = 5.5 V			2.6(4)	mA
		No read/no write mode			100	µA
		Power down mode/halt		0	0.1	

1. Up to 32 bytes can be programmed at a time
2. Data based on reliability test results and monitored in production
3. The data retention time increases when the T_A decreases
4. Guaranteed by design. Not tested in production

13.6.3 EEPROM data memory

T_A = -40 to +125°C, unless otherwise specified.

Table 95. Characteristics of EEPROM data memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating voltage for EEPROM write/erase	Refer to operating range of V_{DD} with T_A, *Section 13.3.1: General operating conditions on page 144*	3.0		5.5	V
t_{prog}	Programming time for 1~32 bytes	T_A = -40 to +125 °C		5	10	ms
t_{RET}(1)	Data retention with 1 k cycling (T_{PROG} = -40 to +125 °C	T_A = 55 °C(2)	20			Years
	Data retention with 10 k cycling (T_{PROG} = -40 to +125 °C)		10			
	Data retention with 100 k cycling (T_{PROG} = -40 to +125 °C)		1			
N_{RW}	Write erase cycles	T_A = 125 °C	300 K			Cycles

1. Data based on reliability test results and monitored in production
2. The data retention time increases when the T_A decreases

13.7 Electromagnetic compatability (EMC) characteristics

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

13.7.1 Functional electromagnetic susceptibility (EMS)

Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electromagnetic 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. The test results are given in the table below based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations

The software flowchart must include the management of runaway conditions such as:

- Corrupted program counter
- Unexpected reset
- Critical data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the reset pin or the oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behaviour is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Table 96. Electromagnetic test results

Symbol	Parameter	Conditions	Level/ class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD} = 5 V, T_A = 25 °C, f_{OSC} = 8 MHz, conforms to IEC 1000-4-2	2B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100 pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD} = 5 V, T_A = 25 °C, f_{OSC} = 8 MHz, conforms to IEC 1000-4-4	3B

13.7.2 Electromagnetic interference (EMI)

Based on a simple application running on the product (toggling 2 LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Table 97. EMI emissions

Sym.	Parameter	Conditions	Monitored frequency band	Max vs. [f_{OSC}/f_{CPU}]		Unit
				8/4 MHz	16/8 MHz	
S_{EMI}	Peak level(1)	V_{DD} = 5 V, T_A = 25 °C, SO20 package, conforming to SAE J 1752/3	0.1MHz to 30 MHz	15	20	dBµV
			30 MHz to 130 MHz	17	21	
			130 MHz to 1 GHz	12	15	
			SAE EMI level	3		-

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

13.7.3 Absolute maximum ratings (electrical sensitivity)

Based on two different tests (ESD and LU) 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 application note AN1181.

Electro-static discharge (ESD)

Electro-static discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). Two models can be simulated: the human body model and the machine model. This test conforms to the JESD22-A114A/A115A standard.

Table 98. ESD absolute maximum ratings

Symbol	Ratings	Conditions	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human body model)	T_A = +25 °C	8000	V
$V_{ESD(MM)}$	Electro-static discharge voltage (Machine model)		400	
$V_{ESD(CDM)}$	Electro-static discharge voltage (Charged device model)		1000	

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

Static and dynamic latch-up (LU)

Three complementary static tests are required on six parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin) and a current injection (applied to each input, output and configurable I/O pin) are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard. For more details, refer to application note AN1181.

Electrical sensitivities

Table 99. Latch up results

Symbol	Parameter	Conditions	Class(1)
LU	Static latch-up class	T_A = 25 °C T_A = 125 °C	A
DLU	Dynamic latch-up class	V_{DD} = 5.5 V, f_{OSC} = 4 MHz, T_A = 25 °C	A

1. Class description: A class is an STMicroelectronics internal specification. All its limits are higher than the JEDEC specifications, which means when a device belongs to class A it exceeds the JEDEC standard. Class B strictly covers all the JEDEC criteria (international standard).

13.8 I/O port pin characteristics

13.8.1 General characteristics

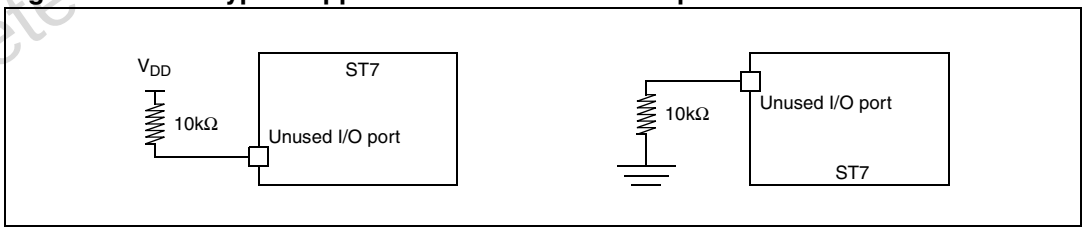
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 100. I/O general port pin characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage			V_{SS} - 0.3		0.3 x V_{DD}	V
V_{IH}	Input high level voltage			0.7 x V_{DD}		V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis(1)				400		mV
I_L	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$				±1	µA
I_S	Static current consumption induced by each floating input pin(2)	Floating input mode			400		
R_{PU}	Weak pull-up equivalent resistor(3)	$V_{IN} = V_{SS}$	V_{DD} = 5 V	50	120	250	kΩ
			V_{DD} = 3.3 V		160		
C_{IO}	I/O pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time1)	C_L = 50 pF between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time1)				25		
$t_{w(IT)in}$	External interrupt pulse time(4)			1			t_{CPU}

1. Data based on validation/design results.
2. 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 75*). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values
3. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 76: Typical VOL at VDD = 3.3 V (standard) on page 163*)
4. To generate an external interrupt, a minimum pulse width must be applied on an I/O port pin configured as an external interrupt source.

Figure 75. Two typical applications with unused I/O pin



1. I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

Caution: During normal operation the ICCCLK pin must be pulled up, internally or externally (external pull-up of 10 k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset.

13.8.2 Output driving current

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A (-40 to +125 °C) unless otherwise specified.

Table 101. Output driving current

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{OL} (1)	Output low level voltage for a standard I/O pin when eight pins are sunk at same time (see *Figure 77*)	V_{DD} = 5 V	I_{IO} = +5 mA			1.0	V
			I_{IO} = +2 mA			0.4	
	Output low level voltage for a high sink I/O pin when four pins are sunk at same time (see *Figure 80*)		I_{IO} = +20 mA			1.4	
			I_{IO} = +8 mA			0.75	
V_{OH} (2)	Output high level voltage for an I/O pin when four pins are sourced at same time (see *Figure 86*)		I_{IO} = -5 mA	V_{DD} - 1.5			
			I_{IO} = -2 mA	V_{DD} - 1.0			
V_{OL} (1)(3)	Output low level voltage for a standard I/O pin when eight pins are sunk at same time (see *Figure 76*)	V_{DD} = 3.3 V	I_{IO} = +2 mA,			0.5	
	Output low level voltage for a high sink I/O pin when four pins are sunk at same time		I_{IO} = +8 mA,				
V_{OH} (2)(3)	Output high level voltage for an I/O pin when four pins are sourced at same time (*Figure 85*)		I_{IO} = -2 mA,	V_{DD} - 0.8			

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 13.2.2: Current characteristics on page 143* 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 13.2.2: Current characteristics on page 143* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}.
3. Not tested in production, based on characterization results.

Figure 76. Typical V_{OL} at V_{DD} = 3.3 V (standard)

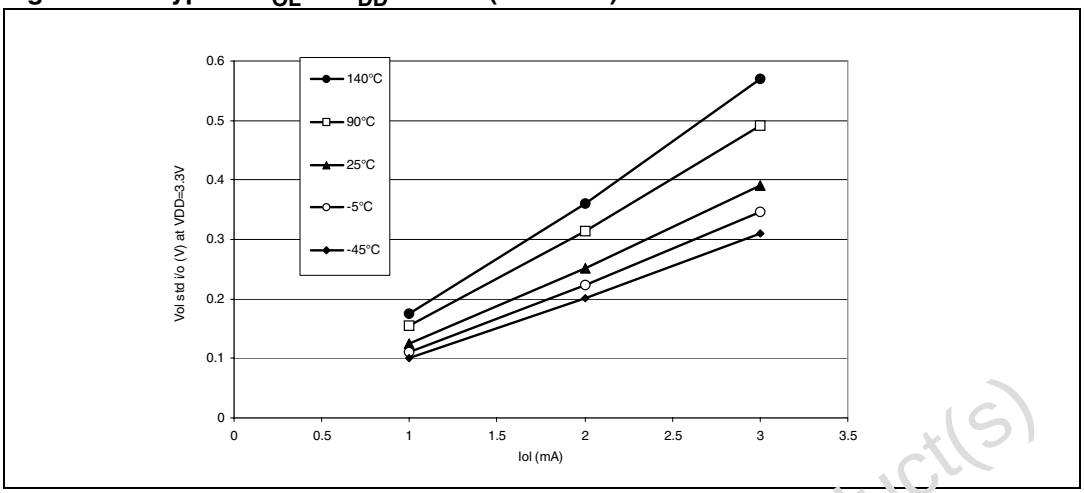


Figure 77. Typical V_{OL} at V_{DD} = 5 V (standard)

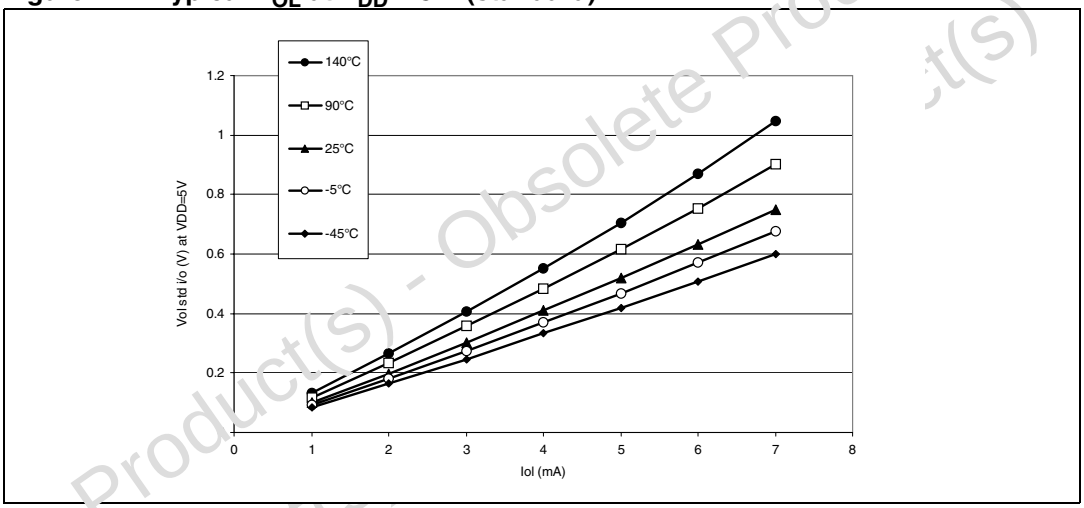


Figure 78. Typical V_{OL} at V_{DD} = 3.3 V (port C)

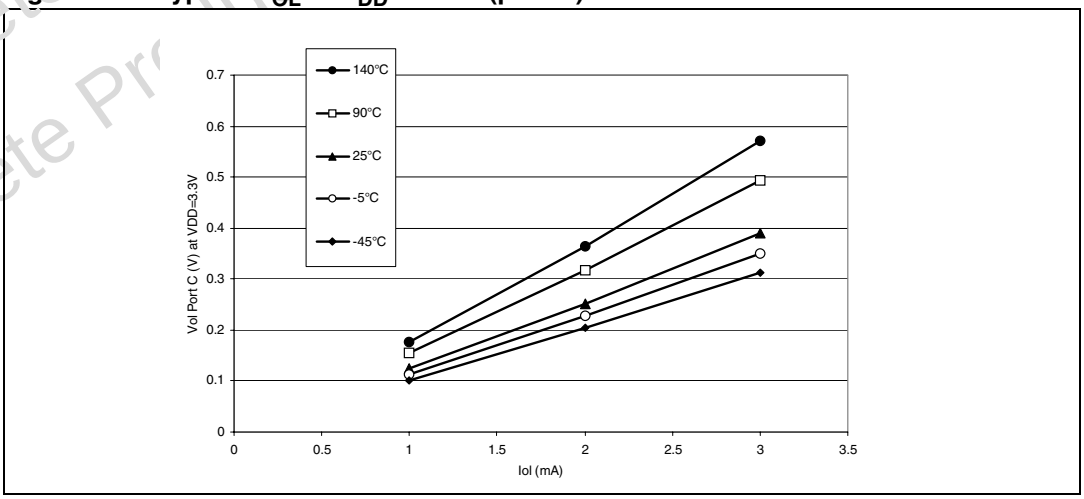


Figure 79. Typical V_{OL} at V_{DD} = 5 V (port C)

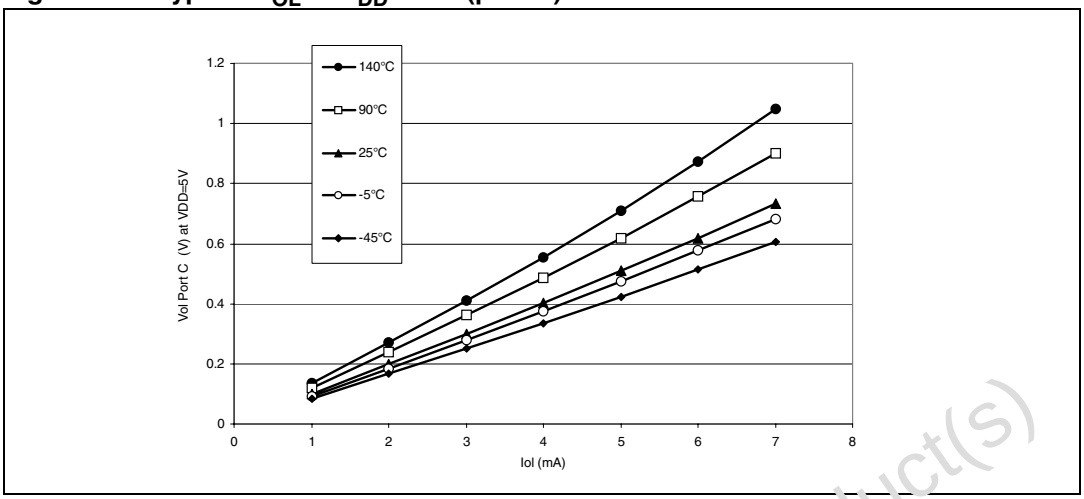


Figure 80. Typical V_{OL} at V_{DD} = 3.3 V (high-sink)

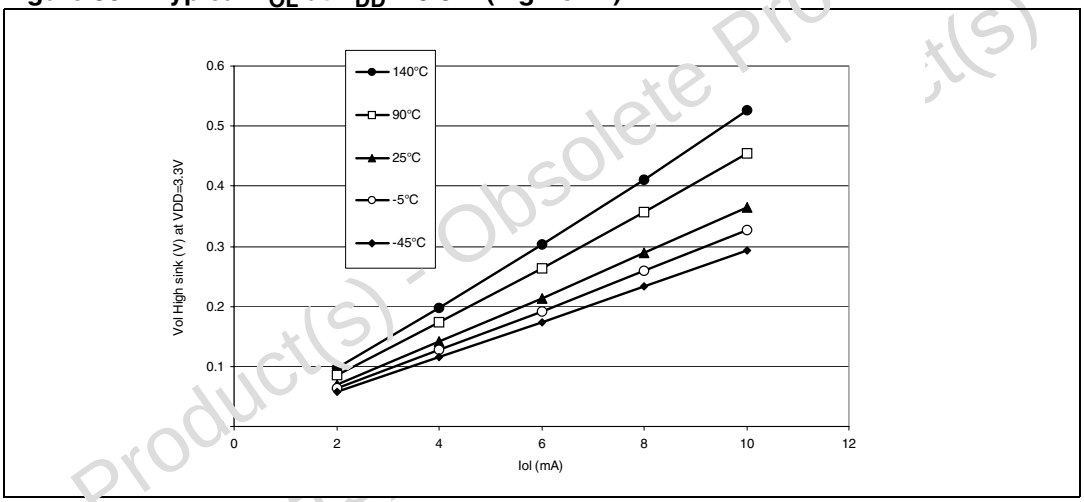


Figure 81. Typical V_{OL} at V_{DD} = 5 V (high-sink)

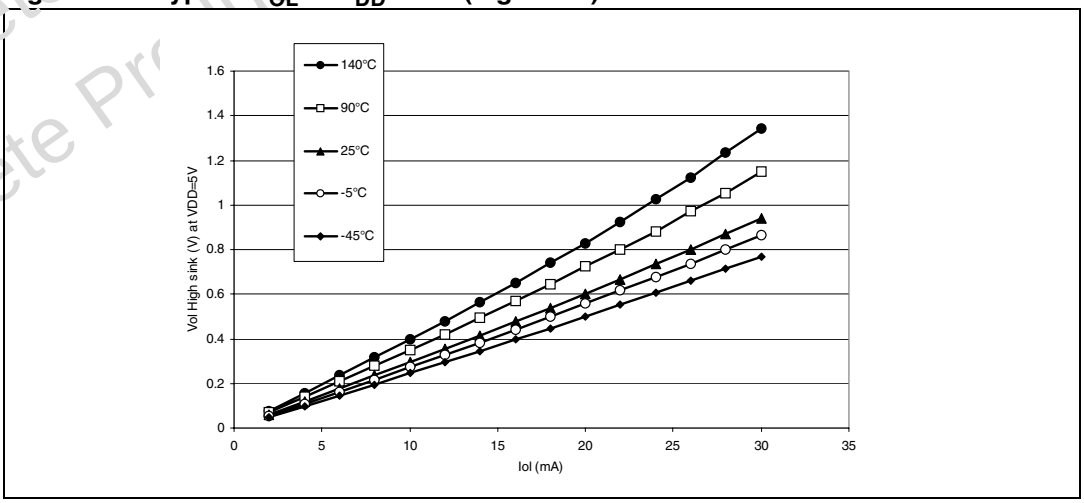


Figure 82. Typical V_{OL} vs V_{DD} (standard I/Os)

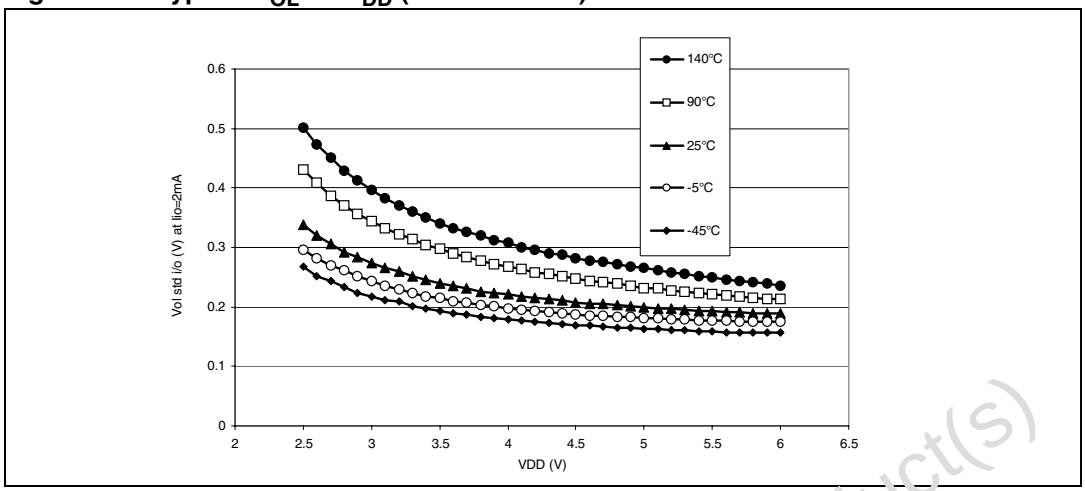


Figure 83. Typical V_{OL} vs V_{DD} (high-sink)

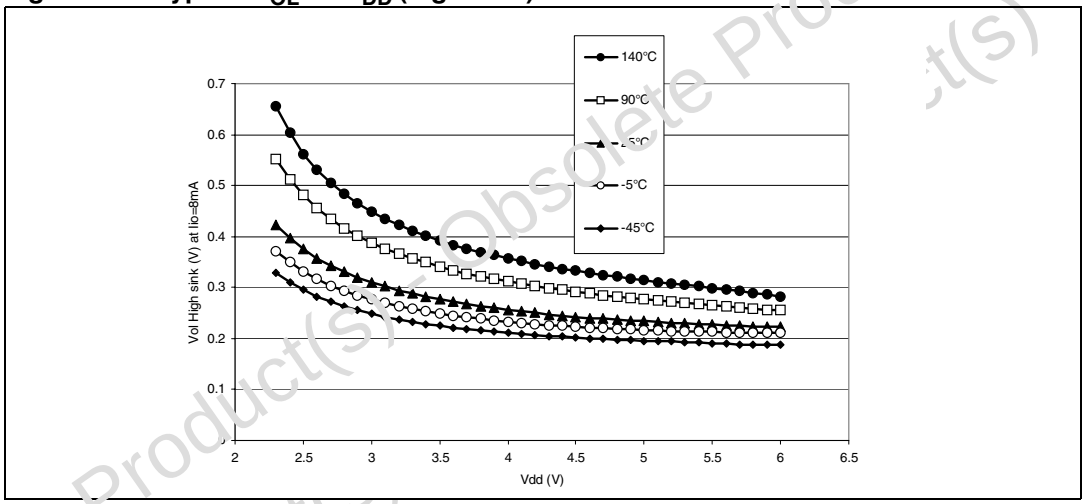


Figure 84. Typical V_{OL} vs V_{DD} (port C)

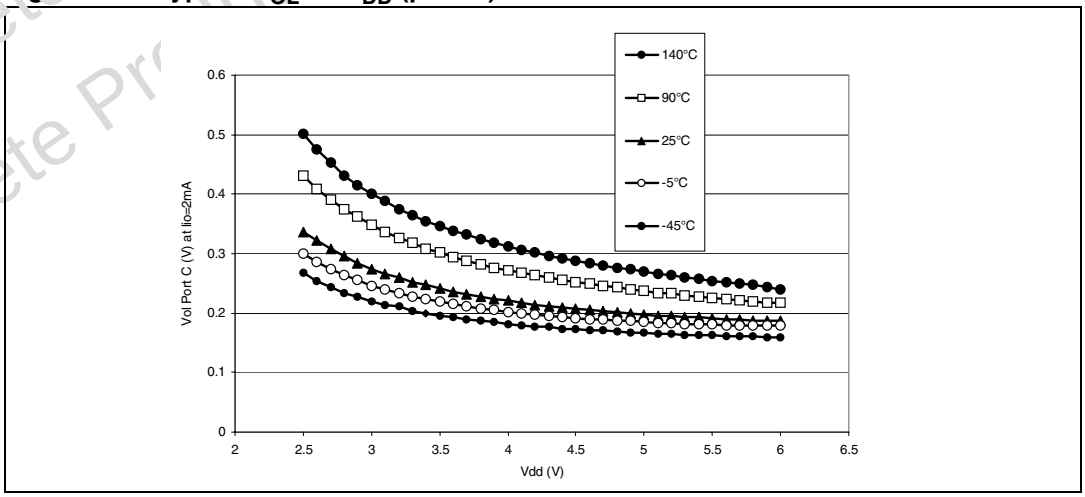


Figure 85. Typical V_{DD} - V_{OH} at V_{DD} = 3.3 V

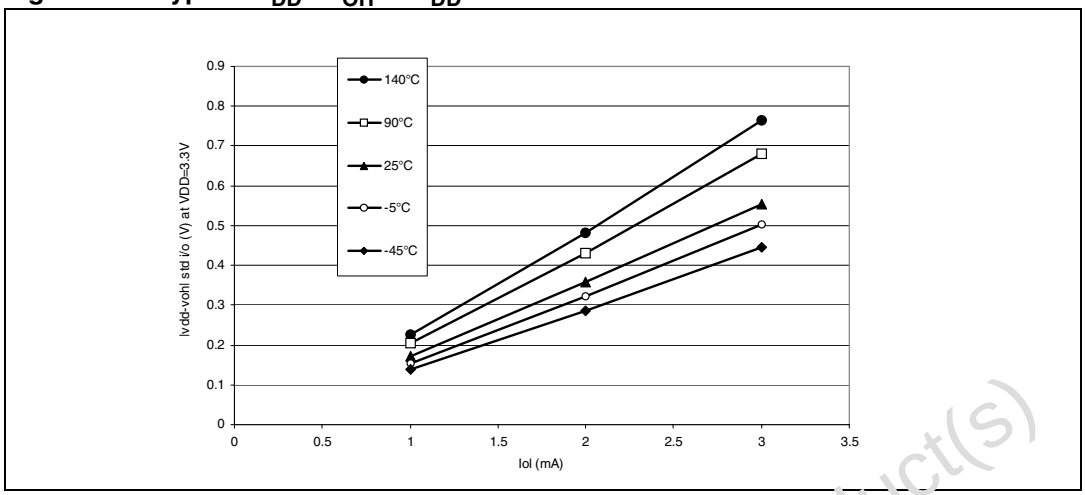


Figure 86. Typical V_{DD} - V_{OH} at V_{DD} = 5 V

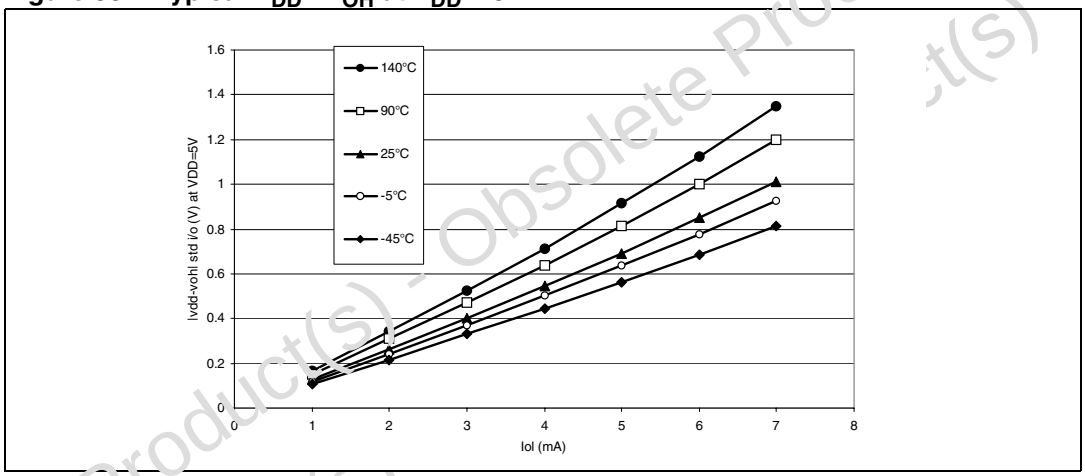


Figure 87. Typical V_{DD} - V_{OH} at V_{DD} = 3.3 V (HS)

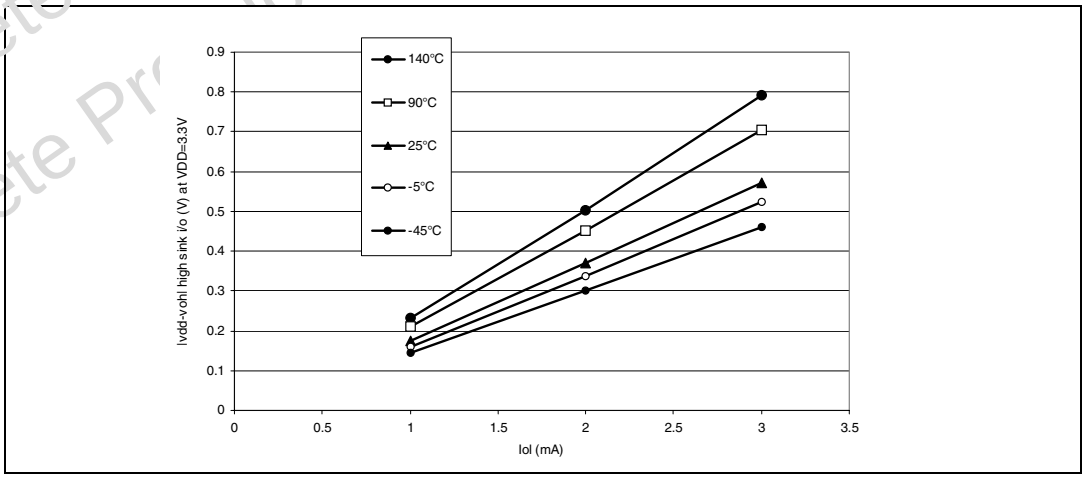


Figure 88. Typical V_{DD} - V_{OH} at V_{DD} = 5 V (HS)

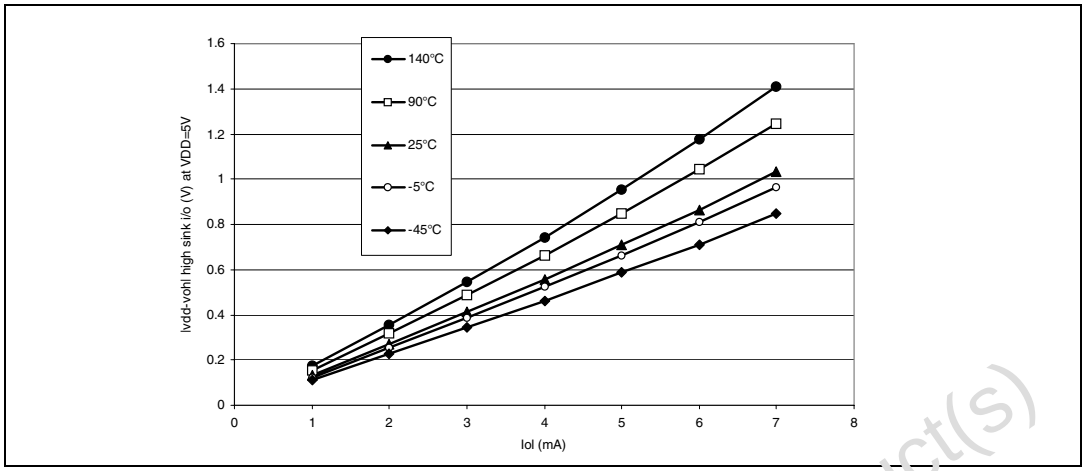


Figure 89. Typical V_{DD} - V_{OH} at V_{DD} = 3.3 V (port C)

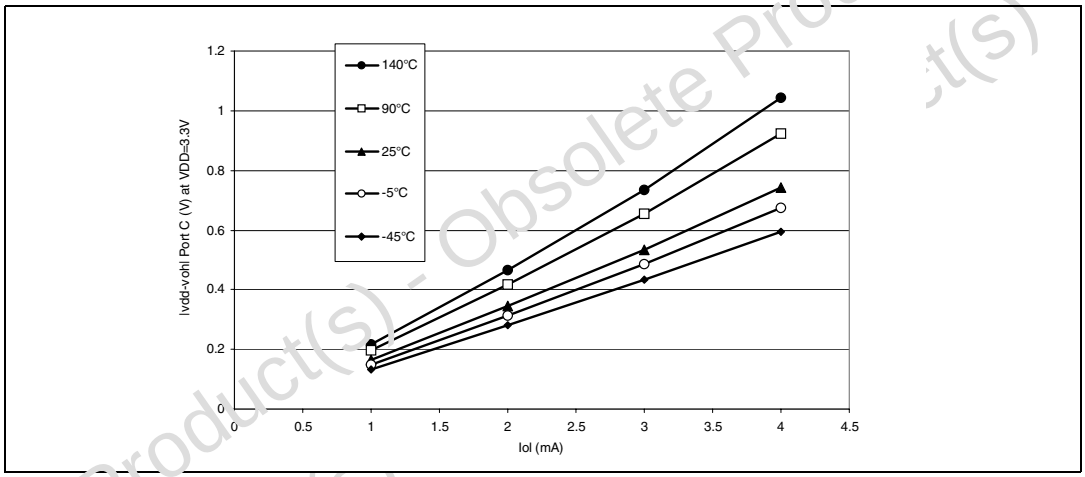


Figure 90. Typical V_{DD} - V_{OH} at V_{DD} = 5 V (port C)

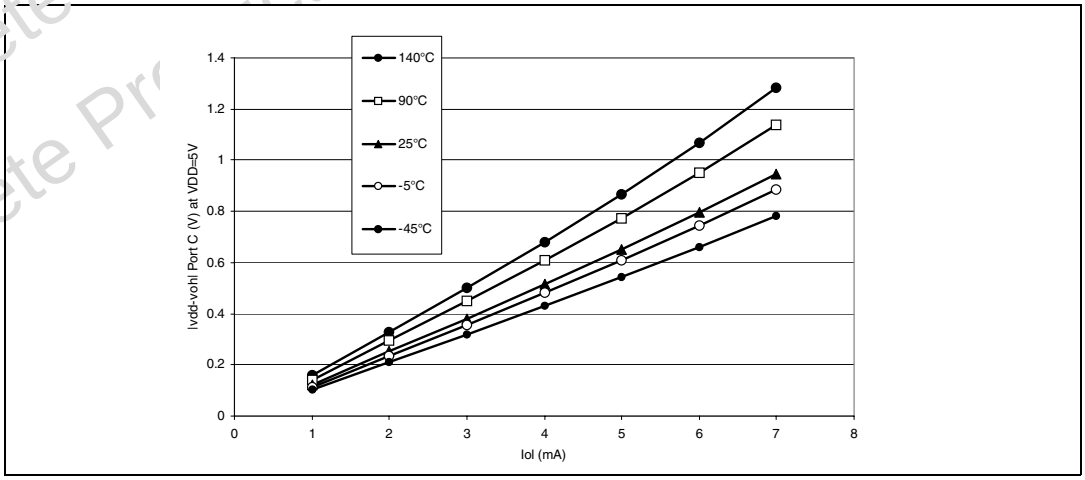


Figure 91. Typical V_{DD} - V_{OH} vs V_{DD} (standard)

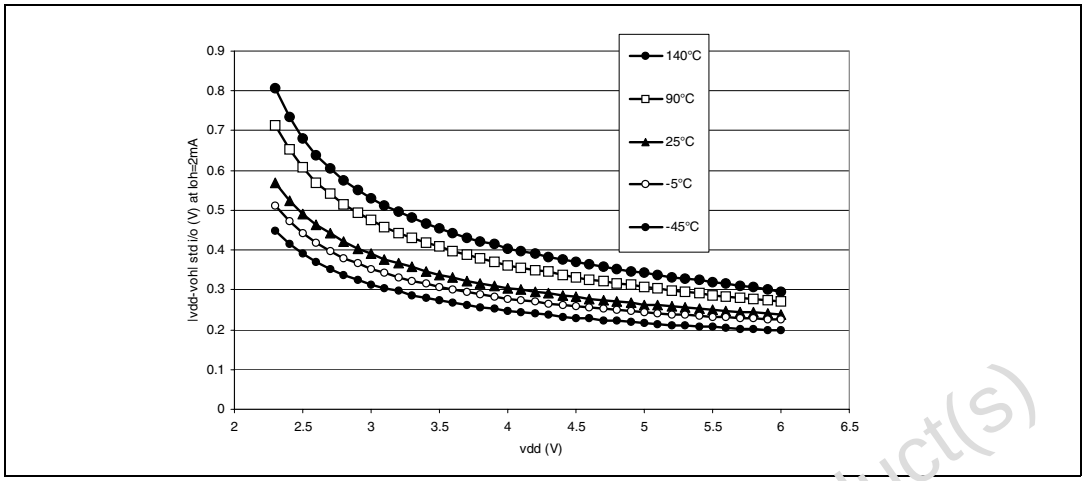


Figure 92. Typical V_{DD}-V_{OH} vs V_{DD} (high sink)

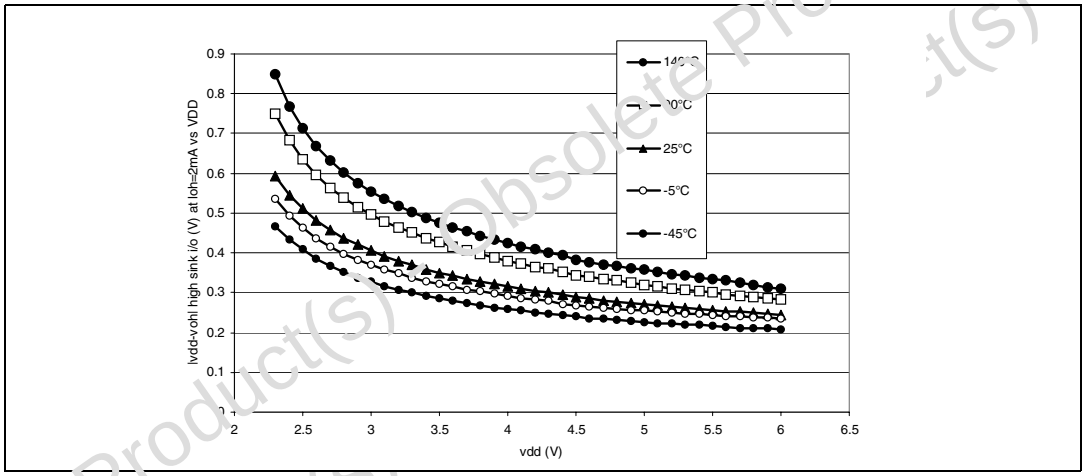
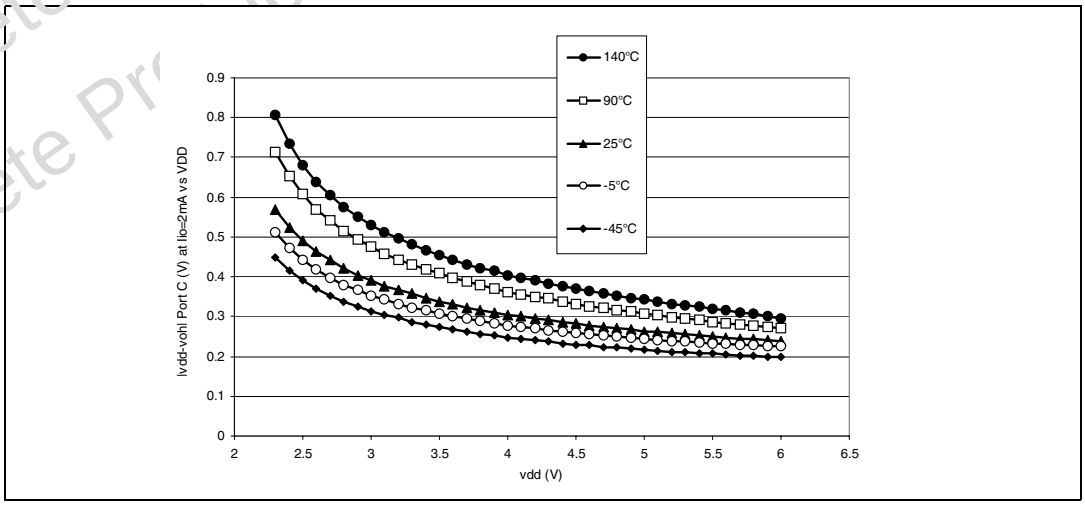


Figure 93. Typical V_{DD} - V_{OH} vs V_{DD} (port C)



13.9 Control pin characteristics

13.9.1 Asynchronous $\overline{\text{RESET}}$ pin

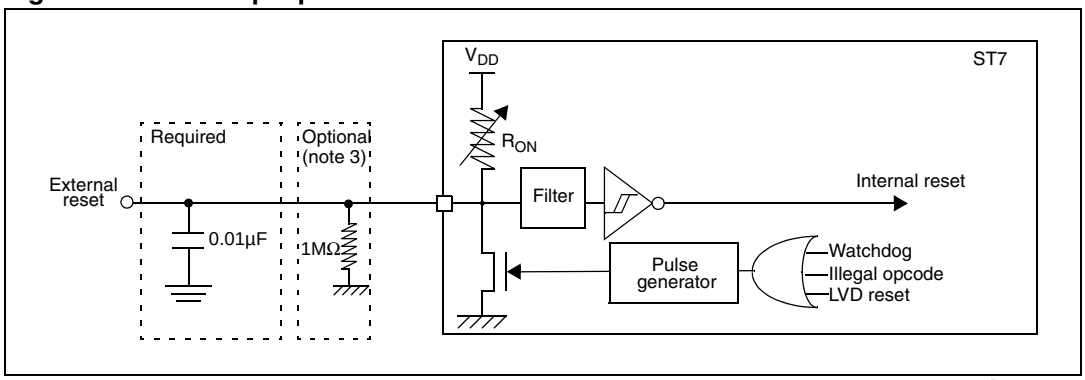
T_A = -40 to +125 °C, unless otherwise specified.

Table 102. Asynchronous $\overline{\text{RESET}}$ pin

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL} (1)	Input low-level voltage			V_{ss} - 0.3		0.3 x V_{DD}	V
V_{IH} (1)	Input high-level voltage			0.7 x V_{DD}		V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis(1)				2		
V_{OL} (1)	Output low-level voltage(2)	V_{DD} = 5 V	I_{IO} = +5 mA, $T_A \le$ +125 °C		0.5	1.0	
			I_{IO} = +2 mA, $T_A \le$ +125 °C		0.2	0.4	
R_{ON}	Pull-up equivalent resistor(1)(3)	V_{DD} = 5 V		20	40	80	kΩ
		V_{DD} = 3.3 V(1)		40	70	120	
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset source			30		µs
$t_{h(RSTL)in}$	External reset pulse hold time(4)			20			
$t_{g(RSTL)in}$	Filtered glitch duration				200		ns

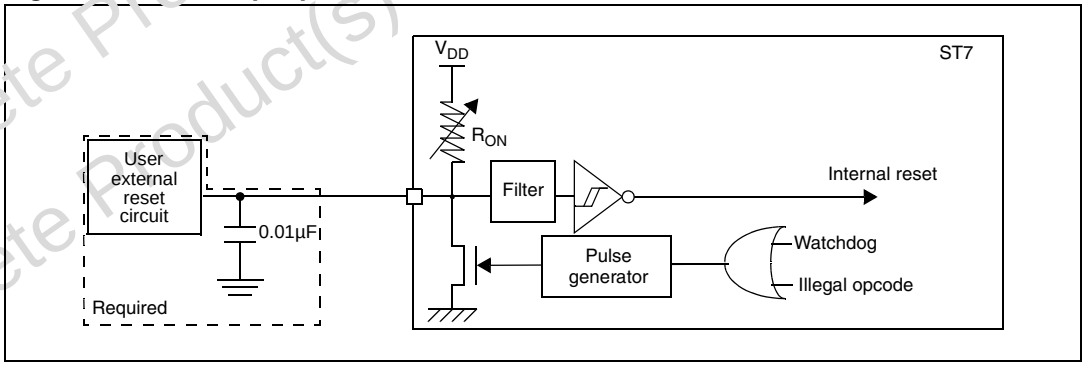
1. Data based on characterization results, not tested in production.
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 13.2.2: Current characteristics on page 143* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. Specified for voltages on $\overline{\text{RESET}}$ pin between V_{ILmax} and V_{DD}.
4. To guarantee the reset of the device, a minimum pulse must be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

Figure 94. $\overline{\text{RESET}}$ pin protection when LVD Is enabled



1. The reset network protects the device against parasitic resets.
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). Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} maximum level specified in *Section 13.9.1: Asynchronous RESET pin on page 169*. Otherwise the reset is not taken into account internally.
Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 13.2.2: Current characteristics on page 143*.
2. When the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.
3. If a capacitive power supply is used, it is recommended to connect a 1MΩ pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by the capacitive effect of the power supply (this adds 5µA to the power consumption of the MCU).
4. Tips when using the LVD:
Check that all recommendations related to ICC, K and reset circuit have been applied (see caution in *Table 1: Device summary on page 1* and notes above)
Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709 and AN2017. If this cannot be done, it is recommended to put a 100nF + 1MΩ pull-down on the $\overline{\text{RESET}}$ pin. The capacitors connected on the $\overline{\text{RESET}}$ pin and also the power supply are key to avoid any start-up marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: Replace 10nF pull-down on the $\overline{\text{RESET}}$ pin with a 5µF to 20µF capacitor."
5. Please refer to *Section 12.2.2: Illegal opcode reset on page 138* for more details on illegal opcode reset conditions.

Figure 95. $\overline{\text{RESET}}$ pin protection when LVD is disabled



1. The reset network protects the device against parasitic resets.
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). Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} maximum level specified in *Section 13.9.1: Asynchronous RESET pin on page 169*. Otherwise the reset is not taken into account internally.
Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 13.2.2: Current characteristics on page 143*.
2. Please refer to *Section 12.2.2: Illegal opcode reset on page 138* for more details on illegal opcode reset conditions.

13.10 Communication interface characteristics

13.10.1 Serial peripheral interface (SPI)

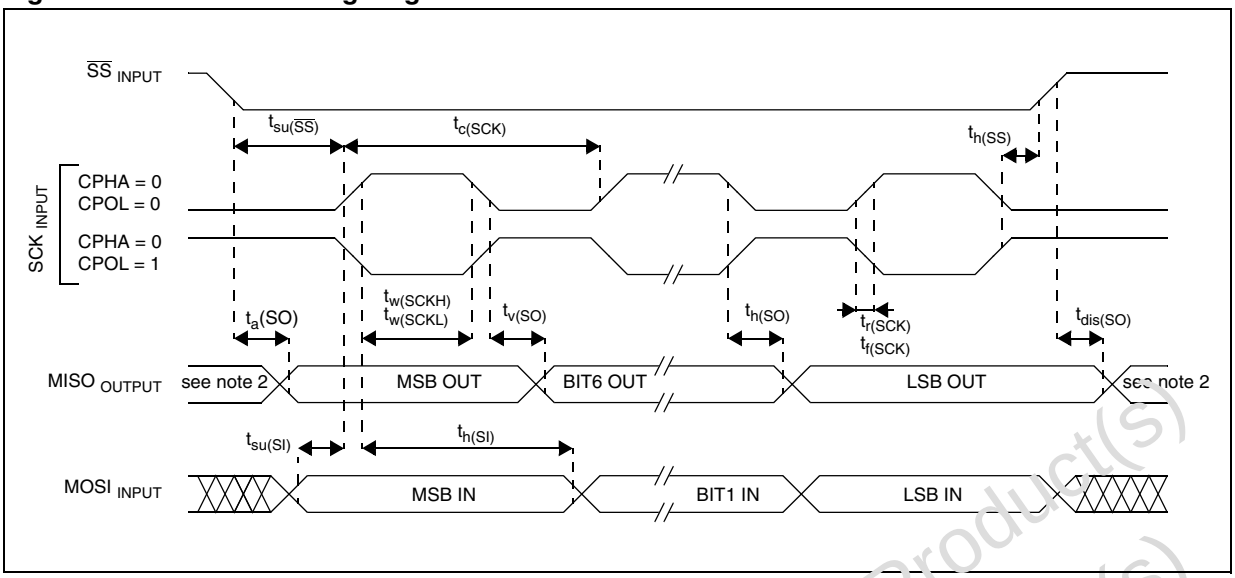
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified. Refer to *Section 10: I/O ports* for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Table 103. SPI characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
$f_{SCK} = 1/t_{c(SCK)}$	SPI clock frequency	Master, f_{CPU} = 8 MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave, f_{CPU} = 8 MHz	0	$f_{CPU}/2 = 4$	
$t_{r(SCK)}$	SPI clock rise and fall time		See *Table 2: Device pin description on page 17*		
$t_{f(SCK)}$					
$t_{su(\overline{SS})}$ (1)	$\overline{SS}$ setup time(2)	Slave	$(4 \times T_{CPU}) + 50$		ns
$t_{h(\overline{SS})}$ (1)	$\overline{SS}$ hold time		120		
$t_{w(SCKH)}$ (1)	SCK high and low time	Master	100		
$t_{w(SCKL)}$ (1)		Slave	90		
$t_{su(MI)}$ (1)	Data input setup time	Master	100		
$t_{su(SI)}$ (1)		Slave			
$t_{h(MI)}$ (1)	Data input hold time	Master			
$t_{h(SI)}$ (1)		Slave			
$t_{a(SO)}$ (1)	Data output access time	Slave	0	120	
$t_{dis(SO)}$ (1)	Data output disable time			240	
$t_{v(SO)}$ (1)	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ (1)	Data output hold time		0		
$t_{v(MO)}$ (1)	Data output valid time	Master (after enable edge)		120	
$t_{h(MO)}$ (1)	Data output hold time		0		

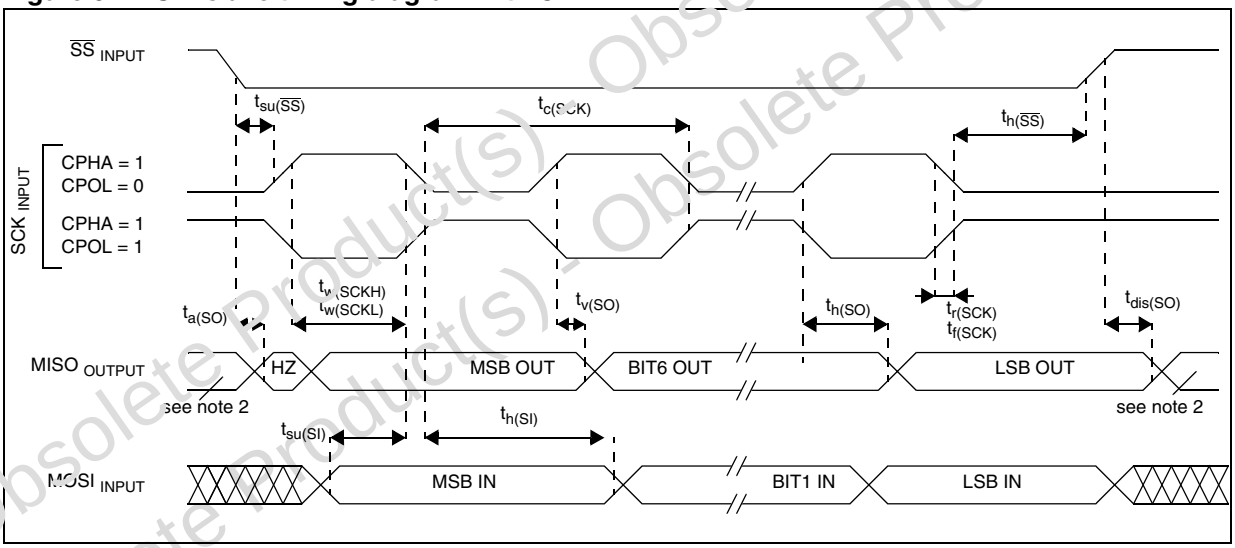
1. Data based on design simulation, not tested in production.
2. Depends on f_{CPU}. For example, if f_{CPU} = 8 MHz, then $t_{CPU} = 1/f_{CPU}$ = 125 ns and $t_{su}(\overline{SS})$ = 550 ns.

Figure 96. SPI slave timing diagram with CPHA = 0



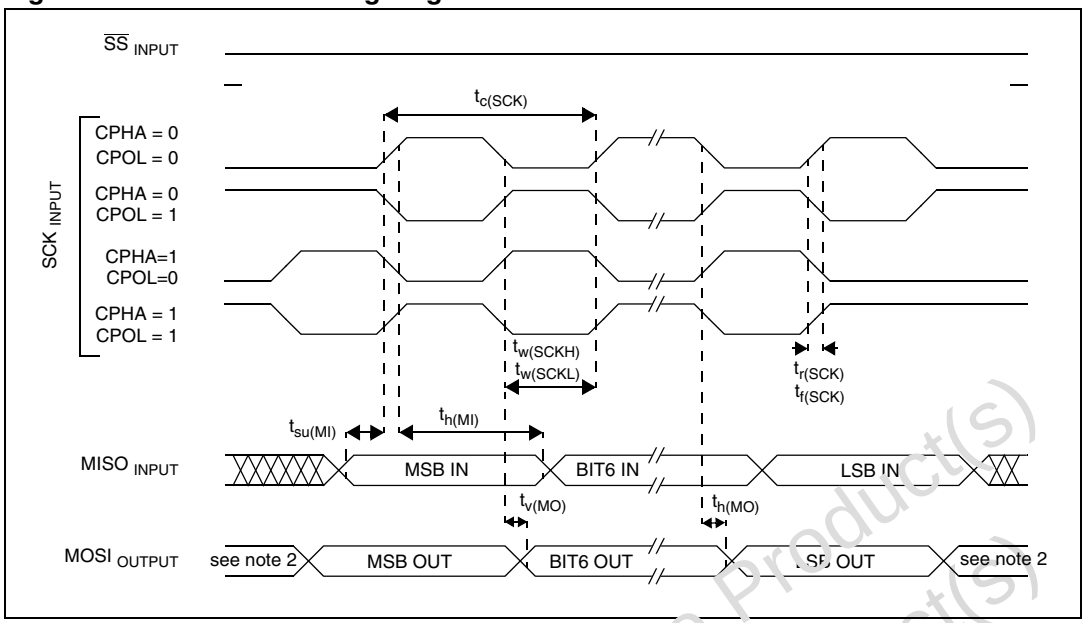
1. Measurement points are done at CMOS levels: 0.3 x V_{DD} and 0.7 x V_{DD}.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

Figure 97. SPI slave timing diagram with CPHA = 1



1. Measurement points are done at CMOS levels: 0.3 x V_{DD} and 0.7 x V_{DD}.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

Figure 98. SPI master timing diagram



1. Measurement points are done at CMOS levels: 0.3 x V_{DD} and 0.7 x V_{DD}.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

13.11 10-bit ADC characteristics

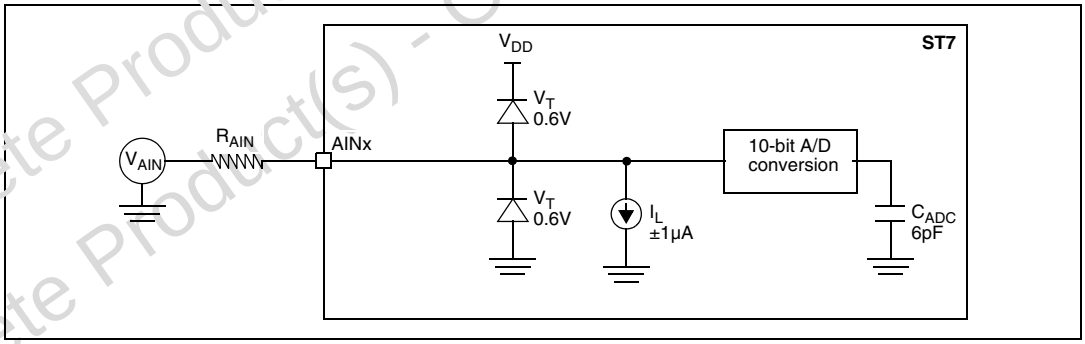
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 104. 10-bit ADC characteristics

Symbol	Parameter	Conditions	Min	Typ(1)	Max	Unit
f_{ADC}	ADC clock frequency	4.5 V ≤V_{DD} ≤5.5 V			4	MHz
		3.0 V ≤V_{DD} ≤3.6 V			2	
V_{AIN}	Conversion voltage range(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} = 8 MHz, f_{ADC} = 4 MHz	0(4)			µs
t_{ADC}	Conversion time (sample+hold)		3.5			$1/f_{ADC}$
	- Sample capacitor loading time - Hold conversion time		4 10			
I_{ADC}	Analog part			1		mA
	Digital part			0.2		

1. Unless otherwise specified, typical data is based on T_A = 25 °C and V_{DD} - V_{SS} = 5 V. 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 refers to V_{DD} and V_{SS}
3. Any added external serial resistor downgrades 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.

Figure 99. Typical application with ADC



Related application notes

- Understanding and minimizing ADC conversion errors (AN1636)
- Software techniques for compensating ST7 ADC errors (AN1711)

Table 105. ADC accuracy with 4.5 V ≤ V_{DD} ≤ 5.5 V

Symbol	Parameter	Conditions	Typ	Max(1)	Unit
$\|E_T\|$	Total unadjusted error	f_{CPU} = 8 MHz, f_{ADC} = 4 MHz(2)(3)	4	6	LSB
$\|E_O\|$	Offset error		3	5	
$\|E_G\|$	Gain error		1	4	
$\|E_D\|$	Differential linearity error		1.5	3	
$\|E_L\|$	Integral linearity error				

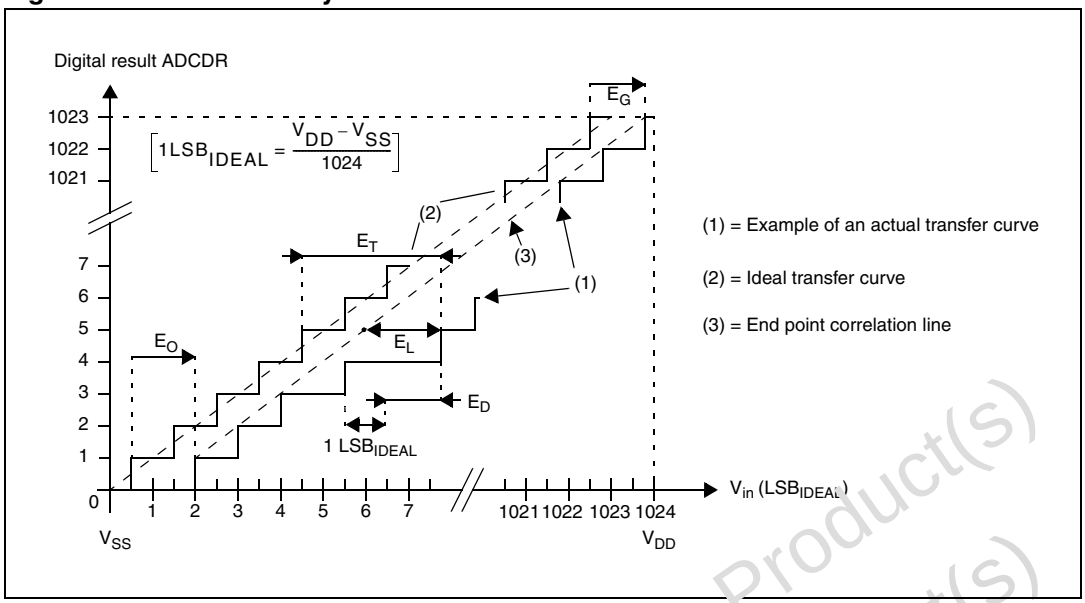
1. Data based on characterization results, monitored in production to guarantee 99.73 % within ± max value from -40 °C to +125 °C (± 3σ distribution limits).
2. Data based on characterization results over the whole temperature range, monitored in production.
3. ADC accuracy vs negative injection current: Injecting negative current on any of the analog input pins may reduce the accuracy of the conversion being performed on another analog input.
The effect of negative injection current on robust pins is specified in *Section 13.11: 10-bit ADC characteristics on page 174*
Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 13.8: I/O port pin characteristics on page 161* does not affect the ADC accuracy.

Table 106. ADC accuracy with 3 V ≤ V_{DD} ≤ 3.6 V

Symbol	Parameter	Conditions	Typ	Max(1)	Unit
$\|E_T\|$	Total unadjusted error	f_{CPU} = 4 MHz, f_{ADC} = 2 MHz(2)(3)	2.8	5.7	LSB
$\|E_O\|$	Offset error		0.25	1.2	
$\|E_G\|$	Gain error		0.6	2.3	
$\|E_D\|$	Differential linearity error		2.9	5.6	
$\|E_L\|$	Integral linearity error		2.6	5.3	

1. Data based on characterization results, monitored in production to guarantee 99.73 % within ± max value from -40 °C to +125 °C (± 3σ distribution limits).
2. Data based on characterization results over the whole temperature range, monitored in production.
3. ADC accuracy vs negative injection current: Injecting negative current on any of the analog input pins may reduce the accuracy of the conversion being performed on another analog input.
The effect of negative injection current on robust pins is specified in *Section 13.11: 10-bit ADC characteristics on page 174*
Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 13.8: I/O port pin characteristics on page 161* does not affect the ADC accuracy.

Figure 100. ADC accuracy characteristics



1. Legend:
E_T = Total unadjusted error: maximum deviation between the actual and the ideal transfer curves
E_O = Offset error: deviation between the first actual transition and the first ideal one
E_G = Gain error: deviation between the last ideal transition and the last actual one
E_D = Differential linearity error: maximum deviation between actual steps and the ideal one
E_L = Integral linearity error: maximum deviation between any actual transition and the end point correlation line

14 Package characteristics

In order to meet environmental requirements, ST offers this device in ECOPACK® packages. These packages have a lead-free second level interconnect. The category of second level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

ECOPACK is an ST trademark. ECOPACK specifications are available at www.st.com.

14.1 Package mechanical data

Figure 101. 20-pin plastic small outline package, 300-mil width

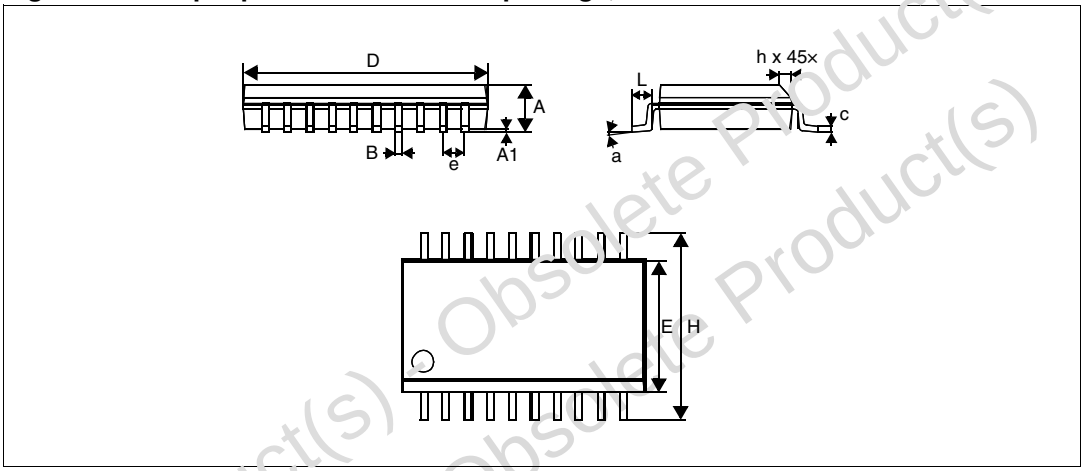


Table 107. 20-pin plastic small outline package, 300-mil width, mechanical data

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
E	7.40		7.60	0.291		0.299
e		1.27			0.050	
H	10.00		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
α	0°		8°	0°		8°
L	0.40		1.27	0.016		0.050

14.2 Thermal characteristics

Table 108. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
R_{thJA}	Package thermal resistance (junction to ambient)	70	°C/W
T_{Jmax}	Maximum junction temperature(1)	150	°C
P_{Dmax}	Power dissipation(2)	< 350	mW

1. The maximum chip-junction temperature is based on technology characteristics
2. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A)/R_{thJA}$.
The power dissipation of an application can be defined by the user with the formula: $P_D = P_{INT} + P_{PORT}$, where P_{INT} is the chip internal power (I_{DD} x V_{DD}) and P_{PORT} is the port power dissipation depending on the ports used in the application.

14.3 Soldering information

In accordance with the RoHS European directive, all STMicroelectronics packages have been converted to lead-free technology, named ECOPACK®.

- ECOPACK® packages are qualified according to the JEDEC STD-020C compliant soldering profile
- Detailed information on the STMicroelectronic ECOPACK® transition program is available on www.st.com/stonline/Leadfree/, with specific technical application notes covering the main technical aspects related to Lead-free conversion (AN2033, AN2034, AN2035, AN2036).

Forward compatibility

ECOPACK® SO packages are fully compatible with a Lead (Pb) containing soldering process (see application note AN2034).

Table 109. Soldering compatibility (wave and reflow soldering process)

Package	Plating material	Pb solder paste	Pb-free solder paste
SO	NiPdAu (Nickel-Palladium-Gold)	Yes	Yes

15 Device configuration and ordering information

15.1 Introduction

Each device is available for production in user programmable versions (Flash) as well as in factory coded versions (ROM). ST7L1x devices are ROM versions

ST7PL1x devices are factory advanced service technique ROM (FASTROM) versions: They are factory programmed Flash devices.

ST7FL1x Flash devices are shipped to customers with a default program memory content (FFh), while ROM/FASTROM factory coded parts contain the code supplied by the customer. This implies that Flash devices have to be configured by the customer using the option bytes while the ROM/FASTROM devices are factory-configured.

15.2 Option bytes

The option bytes have no address in the memory map and are accessed only in programming mode (for example using a standard ST7 programming tool). Difference in option byte configuration between Flash and ROM devices are presented in the following table and are described in *Section 15.2.1: Flash option bytes on page 180* and *Section 15.2.2: ROM option bytes on page 182*.

Table 110. Flash and ROM option bytes

		Option byte 0								Option byte 1							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Flash	Name	Reserved		CLKSEL		SEC 1	SEC 0	FM PR	FM PW	PLL x4x8	PLL OFF	Res	OSC	LVD 1:0		WDG SW	WDG HALT
	Default value	0	1	1	1	0	1	0	0	1	1	1	0	1	1	1	1

ROM	Name	AW UCK	OSCRANGE 2:0			Reserved		ROP _R	ROP _D	Res	PLL OFF	Res	OSC	LVD 1:0		WDG SW	WDG HALT
	Default value	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1

15.2.1 Flash option bytes

The 2 option bytes allow the hardware configuration of the microcontroller to be selected.

Table 111. Option byte 0 description

Bit	Bit name	Function
7:6	-	Reserved, must be kept cleared
5:4	CLKSEL	Clock source selection When the internal RC oscillator is not selected (option OSC = 1), these option bits select the clock source: Resonator oscillator or external clock (see *Table 112*) *Note: When the internal RC oscillator is selected, the CLKSEL option bits must be kept at their default value in order to select the 256 clock cycle delay (see Section 7.5: Reset sequence manager (RSM) on page 43)*
3:2	SEC[1:0]	Sector 0 size definition These option bits indicate the size of sector 0 as follows: 00: Sector 0 size = 0.5 Kbytes 01: Sector 0 size = 1 Kbytes 10: Sector 0 size = 2 Kbytes 11: Sector 0 size = 4 Kbytes
1	FMP_R	Readout protection Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected will cause the whole memory to be erased first and the device can be reprogrammed. Refer to the *ST7 Flash Programming Reference Manual* and *Section 4.5: Memory protection on page 25* for more details. 0: Readout protection off 1: Readout protection on
0	FMP_W	Flash write protection This option indicates if the Flash program memory is write protected. **Warning: When this option is selected, the program memory (and the option bit itself) can never be erased or programmed again.** 0: Write protection off 1: Write protection on

Table 112. Clock source selection

Clock source		Port C	CLKSEL	
			2	1
Resonator		Ext. osc enabled/port C disabled	0	0
External clock source: CLKIN	on PB4	Ext. osc disabled/ port C enabled	0	1
	on PC0		1	1
Reserved			1	0

Table 113. Option byte 1 description

Bit	Bit name	Function
7	PLLx4x8	PLL factor selection 0: PLLx4 1: PLLx8
6	PLLOFF	PLL disable This option bit enables or disables the PLL. 0: PLL enabled 1: PLL disabled (bypassed)
5	-	Reserved, must be kept cleared
4	OSC RC	Oscillator selection This option bit enables to select the internal RC oscillator. 0: RC oscillator on 1: RC oscillator off *Note: If the RC oscillator is selected, then to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100 nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.*
3:2	LVD[1:0]	Low voltage selection These option bits enable the voltage detection block (LVD) with a selected threshold to the LVD: 11: LVD off 10: LVD high threshold
1	WDGSW	Hardware or software watchdog 0: Hardware (watchdog always enabled) 1: Software (watchdog to be enabled by software)
0	WDGHALT	Watchdog reset on halt 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

Table 114. List of valid option combinations

Operating conditions		PLL	Typ f_{CPU}	Option bits		
V_{DD} range	Clock source			OSC	PLLOFF	PLLx4x8
3.0 to 3.6 V	Internal RC 1%	off	1 MHz @ 3.3 V	0	1	1
		x4	4 MHz @ 3.3 V		0	0
		x8	-	-	-	-
	External clock or resonator (depending on OPT5:4 selection)	off	0 to 4 MHz	1	1	1
		x4	4 MHz		0	0
		x8	-	-	-	-

Table 114. List of valid option combinations (continued)

Operating conditions		PLL	Typ f_{CPU}	Option bits		
V_{DD} range	Clock source			OSC	PLLOFF	PLLx4x8
4.5 to 5.5V	Internal RC 1%	off	1 MHz @ 5V	0	1	1
		x4	-	-	-	-
		x8	8 MHz @ 5V	0	0	1
	External clock or resonator (depending on OPT5:4 selection)	off	0 to 8 MHz	1	1	1
		x4	-	-	-	-
		x8	8 MHz	1	0	1

15.2.2 ROM option bytes

The 2 ROM option bytes allow the hardware configuration of the microcontroller to be selected.

Table 115. Option byte 0 description

Bit	Bit name	Function
7	AWUCK	Auto wake up clock selection 0: 32 kHz oscillator (VLP) selected as AWU clock 1: AWU RC oscillator selected as AWU clock. *Note: If this bit is reset, internal RC oscillator must be selected (option OSC = 0).*
6:4	OSCRANGE[2:0]	Oscillator range When the internal RC oscillator is not selected (option OSC = 1), these option bits select the range of the resonator oscillator current source or the external clock source. 000: Typ. frequency range with resonator (LP) = 1~2 MHz 001: Typ. frequency range with resonator (MP) = 2~4 MHz) 010: Typ. frequency range with resonator (MS) = 4~8 MHz) 011: Typ. frequency range with resonator (HS) = 8~16 MHz) 100: Typ. frequency range with resonator (VLP) = 32.768~ kHz) 101: External clock on OSC1 110: Reserved *Note: OSCRANGE[2:0] has no effect when AWUCK option is set to 0. In this case, the VLP oscillator range is automatically selected as AWU clock*
3:2	-	Reserved, must be kept cleared
1	ROP_R	Readout protection for ROM This option is for read protection of ROM 0: Readout protection off 1: Readout protection on
0	ROP_D	Readout protection for data EEPROM This option is for read protection of EEPROM memory. 0: Readout protection off 1: Readout protection on

Table 116. Option byte 1 description

Bit	Bit name	Function
7	-	Reserved, must be kept cleared
6	PLLOFF	PLL disable This option bit enables or disables the PLL. 0: PLL enabled 1: PLL disabled (bypassed)
5	-	Reserved, must be kept cleared
4	OSC RC	Oscillator selection This option bit enables to select the internal RC oscillator. 0: RC oscillator on 1: RC oscillator off *Note: If the RC oscillator is selected, then to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100 nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.*
3:2	LVD[1:0]	Low voltage selection These option bits enable the voltage detection block (LVD) with a selected threshold to the LVD: 11: LVD off 10: LVD high threshold
1	WDGSW	Hardware or software watchdog 0: Hardware (watchdog always enabled) 1: Software (watchdog to be enabled by software)
0	WDGHALT	Watchdog reset on halt 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

15.3 Device ordering information

Figure 102. Flash commercial product code structure

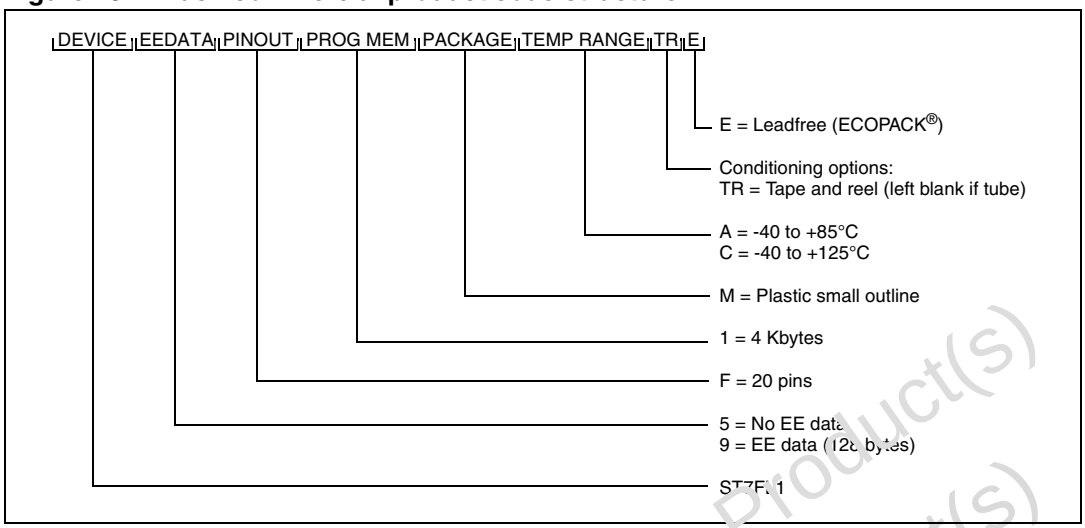


Table 117. Flash user programmable device types

Order code	Program memory (bytes)	RAM (bytes)	Data EEPROM (bytes)	Temperature range	Package
ST7FL15F1MAE	4 K Flash	256	-	-40 to +85°C	SO20
ST7FL19F1MAE			128		
ST7FL15F1MCE			-	-40 to +125°C	
ST7FL19F1MCE			128		

Figure 103. FASTROM commercial product code structure

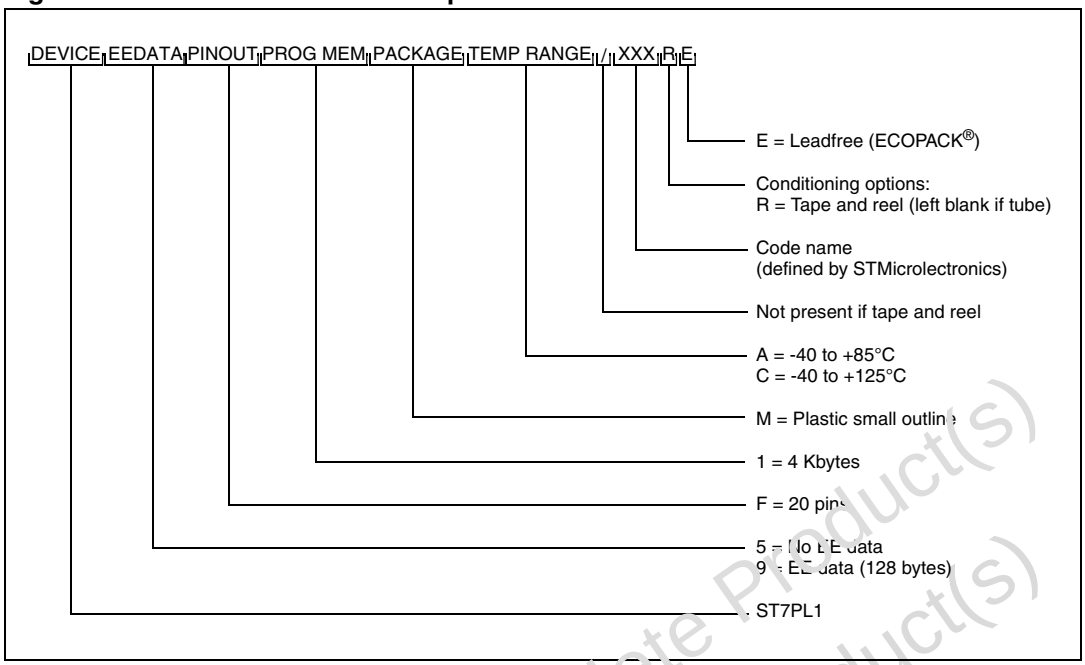


Table 118. FASTROM factory coded device types

Order code[1]	Program memory (bytes)	RAM (bytes)	Data EEPROM (bytes)	Temperature range	Package
ST7PL15F1MAxxxRE	4 K FASTROM	256	-	-40 to +85°C	SO20
ST7PL19F1MAxxxRE			128		
ST7PL15F1MCxxxRE			-	-40 to +125°C	
ST7PL19F1MCxxxRE			128		

1. 'xxx' represents the code name defined by STMicroelectronics
'R' = Tape and reel (left blank if tube)

Figure 104. ROM commercial product code structure

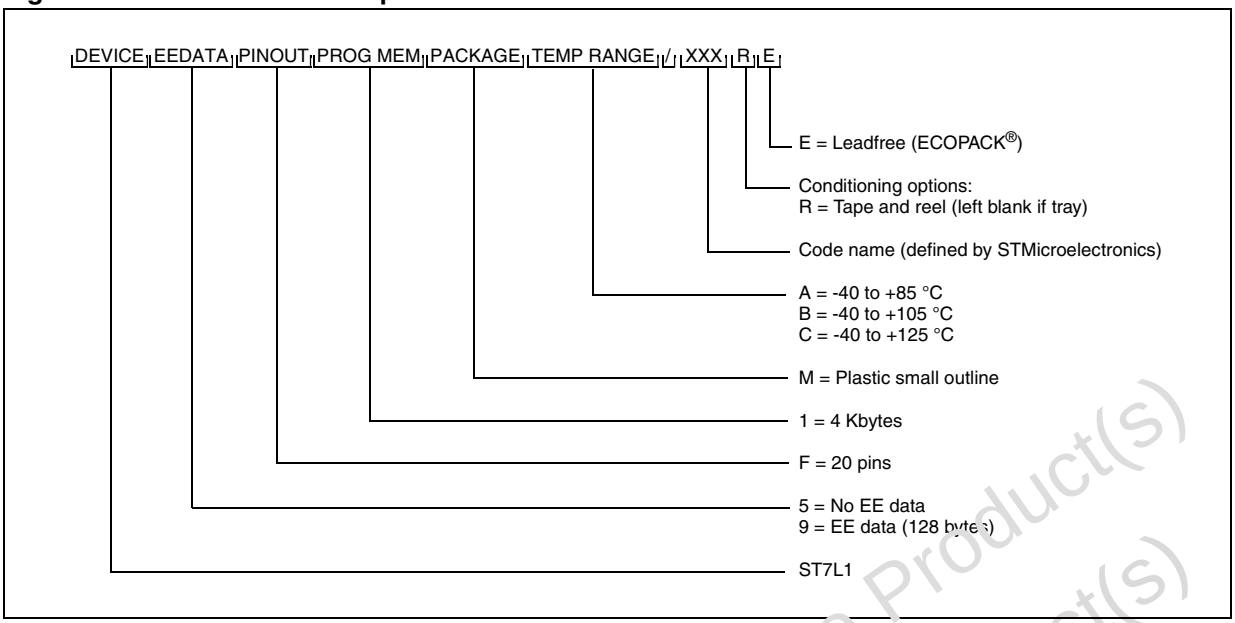


Table 119. ROM factory coded device types

Order code[1]	Program memory (bytes)	RAM (bytes)	Data EEPROM (bytes)	Temperature range	Package
ST7L15F1MA/xxxRE	4 K ROM	256	-	-40 to +85 °C	SO20
ST7L19F1MA/xxxRE			128		
ST7L15F1MB/xxxRE			-	-40 to +105 °C	
ST7L19F1MB/xxxRE			128		
ST7L15F1MC/xxxRE			-	-40 to +125 °C	
ST7L19F1MC/xxxRE			128		

1. 'xxx' represents the code name defined by STMicroelectronics
'R' = Tape and reel (left blank if tube)

ST7L1 FASTROM and ROM microcontroller option list

(Last update: September 2007)

Customer: .....
Address: .....
.....
Contact: .....
Phone No: .....
Reference/FASTROM or ROM code: .....

The FASTROM/ROM code name is assigned by STMicroelectronics.
FASTROM/ROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device type/memory size/package (check only one option):

FASTROM device	SO20 4 K
	[ ] ST7PL15F1M
	[ ] ST7PL19F1M

ROM device	SO20 4 K
	[ ] ST7L15F1M
	[ ] ST7L19F1M

Conditioning: [ ] Tape and reel [ ] Tube
(check only one option)

Special marking: [ ] No [ ] Yes "..... ... ..."
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count: SO20 (8 char. max): .....

Temperature range: [ ] A (-40 to +85 °C) [ ] B (-40 to +105 °C)(1) [ ] C (-40 to +125 °C)
LVD reset threshold: [ ] Disabled [ ] Enabled (highest voltage threshold)
Watchdog selection: [ ] Software activation [ ] Hardware activation
Watchdog reset on halt: [ ] Disabled [ ] Enabled

Flash devices only:
PLL: [ ] Disabled [ ] PLLx4 [ ] PLLx8
Clock source selection: [ ] Resonator
[ ] External clock [ ] on PB4
[ ] on OSCI
[ ] Internal RC oscillator
Sector 0 size: [ ] 0.5 K [ ] 1 K [ ] 2 K [ ] 4 K
Readout protection: [ ] Disabled [ ] Enabled
Flash write protection: [ ] Disabled [ ] Enabled

ROM devices only
PLL: [ ] Disabled [ ] Enabled
Clock source selection: [ ] Resonator
[ ] VLP: Very low power resonator (32 to 100 kHz)
[ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] External clock
[ ] on PB4
[ ] on OSCI
[ ] Internal RC oscillator

AWUCK selection: [ ] 32 kHz oscillator [ ] AWU RC oscillator
ROM readout protection: [ ] Disabled [ ] Enabled
EEDATA readout protection: [ ] Disabled [ ] Enabled

Comments: .....
Supply operating range in the application: .....
Notes: .....
Date: .....
Signature: .....

1. For ROM device only

15.4 Development tools

15.4.1 Introduction

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

15.4.2 Evaluation tools and starter kits

ST offers complete, affordable starter kits and full-featured evaluation boards that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

15.4.3 Development and debugging tools

Application development for ST7 is supported by fully optimizing C compilers and the ST7 assembler-linker toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The cosmic C compiler is available in a free version that outputs up to 16 Kbytes of code.

The range of hardware tools includes full featured ST7-EMU3 series emulators, cost effective ST7-DVP3 series emulators and the low-cost RLink in-circuit debugger/programmer. These tools are supported by the ST7 Toolset from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

15.4.4 Programming tools

During the development cycle, the ST7-DVP3 and ST7-EMU3 series emulators and the RLink provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer, the ST7-STICK, as well as ST7 socket boards which provide all the sockets required for programming any of the devices in a specific ST7 subfamily on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

15.4.5 Order codes for development and programming tools

Table 120 below lists the ordering codes for the ST7L1 development and programming tools. For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.

Table 120. ST7L1 development and programming tools

Supported products	In-circuit debugger, RLink Series(1)	Emulator		Programming tool	
	Starter kit without demo board	DVP series	EMU series	In-circuit programmer	ST socket boards and EPBs
ST7FL15 ST7FL19	STX-RLINK(2)(3)	ST7MDT10-DVP3(4)	ST7MDT10-EMU3	ST7-STICK(3)(5) STX-RLINK(6)	ST7SB10-123(3)

1. Available from ST or from Raisonance, www.raisonance.com
2. USB connection to PC
3. Add suffix /EU, /UK or /US for the power supply for your region
4. Includes connection kit for DIP16/SO16 only. See “How to order an EMU or DVP” in ST product and tool selection guide for connection kit ordering information
5. Parallel port connection to PC
6. RLink with ST7 tool set

15.5 ST7 application notes

All relevant ST7 application notes can be found on www.st.com.

16 Revision history

Table 121. Revision history

Date	Revision	Main changes
16-Aug-2006	1	Initial release
20-Dec-2006	2	Replaced 'ST7L1' with *ST7L19 on page 1* Added 'Features' heading above list of *Features on page 1* Removed AVD feature from 'Clock reset and supply management' in *Features on page 1* Changed *Section 1 on page 14* *Figure 1 on page 15*: - removed AVD - replaced autoreload timer 2 with autoreload timer 4 - referenced port C to figure footnote *Table 3 on page 20*: - replaced autoreload timer 2 with autoreload timer 4 - added '1' to register name and register label of AT4 counter register *Section 7: Supply, reset and clock management on page 37*: - Changed clock management feature - Removed AVD from SI management feature Changed *Section 7.4: Multi-oscillator (MO) on page 42* *Section 7.6: System integrity management (SI) on page 46*: Removed mention of AVD function from first paragraph Added caution about avoiding unwanted behavior during reset sequence in *Section 7.5.1: Introduction on page 43* *Figure 12 on page 41*: Removed lock32 from bit 7 in SICSR *Figure 17 on page 47*: Removed AVD from bits 1:0 in SICSR Removed *Section 7.6.2 Auxiliary voltage detector (AVD)* Removed *Monitoring the VDD main supply* Removed *Figure 18. Using the AVD to Monitor VDD from Section 7* *Section 7.6.2: Low power modes on page 48*: Removed AVD references Removed *subsection 'Interrupts'* *Section 7.6.3: Register description on page 48*: Changed bits 1:0 to reserved in SICSR Changed *Section 11.2: Dual 12-bit autoreload timer 4 (AT4) on page 77* by replacing AT3 with AT4 Changed description of bits 11:0 of CNTR register in *Section 11.2.6: Register description on page 93* *Table 15: Interrupt mapping on page 52*: Changed description of interrupt No. 7 *Section 13.3.2: Operating conditions with low voltage detector (LVD) on page 150*: Removed AVD from current consumption Removed *Section 13.3.3 Auxiliary voltage detector (AVD) thresholds* *Section 13.4.1: Supply current on page 151*: Changed typical and max values for AWUFH and for active halt *Table 106: ADC accuracy with 3 V < VDD < 3.6 V on page 175*: Changed typical and maximum values *Table 105: ADC accuracy with 4.5 V < VDD < 5.5 V on page 175*: Redistributed max value footnote links *Table 108: Thermal characteristics on page 178*: Changed package thermal resistance and power dissapation values

Table 121. Revision history

Date	Revision	Main changes
20-Dec-2006	2 cont'd	Removed text concerning Pb-containing packages from *Section 14.2 on page 178* *Table 109: Soldering compatibility (wave and reflow soldering process) on page 178*: - changed title of 'Plating material' column - removed note concerning Pb-package temperature for leadfree soldering compatibility Changed *Section 15: Device configuration and ordering information on page 179* *Section 15.3: Device ordering information on page 184:* - removed *Table 26: Supported part numbers* - added *Figure 102: Flash commercial product code structure* - added *Table 117: Flash user programmable device types* - added *Figure 103: FASTROM commercial product code structure* - added *Table 118: FASTROM factory coded device types* - added *Figure 104: ROM commercial product code structure* - added *Table 119: ROM factory coded device types* Updated *ST7L1 FASTROM and ROM microcontroller option list on page 187* Changed *Section 15.5 on page 189*. Removed *Table 28: ST7 Application Notes* and added a statement to indicate that application notes can be found on the ST website
15-Jan-2007	3	Corrected revision number on page 1 (previous revision 2 inadvertently stated Rev. 1 at bottom of cover page)
11-Sep-2007	4	Status of document updated from 'preliminary data' to 'datasheet' Updated *Memories on page 1* *Section 7.1: Internal RC oscillator adjustment on page 37*: Removed reference to '4.5 V to 5.5 V' from text and changed 'T_A = 25 °C' to '$T_A$max' *Table 7: RCCR calibration registers on page 38*: Changed 'T_A = 25 °C' to '$T_A$max' and added footnote 1 concerning RCCRO and RCCR1 calibrations Simplified *Section 10.7: Device-specific I/O port configuration on page 73* *PWM mode on page 80*: Removed all references to the PLL 32 MHz signal *Table 80: General operating conditions on page 144*: Added B suffix temperature range (-40 to + 105 °C)for ROM devices only *Table 81: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 4.5 to 5.5 V on page 145* and *Table 83: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 3.0 to 3.6 V on page 147*: Updated f_{RC}, ACC_{RC}, and ACC_{PLL} parameters; added ROM data *Table 82: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 4.5 to 5.5 V on page 145* and *Table 84: Operating conditions (tested for TA = -40 to +125 °C) @ VDD = 3.0 to 3.6 V on page 147*: Modified typical value of PLL x8 accuracy from 0.1 % to 0.2 %. Added $f_{CLKIN}/2$ to the conditions *Table 87: Supply current on page 151*: Replaced 'TBD' with 90 µA for supply current in AWUFH mode and with 0.7 for supply current in active halt mode *Table 95: Characteristics of EEPROM data memory on page 158*: Added minimum data and conditions for write erase cycles (N_{RW})

Table 121. Revision history

Date	Revision	Main changes
11-Sep-2007	4 cont'd	*Table 100: I/O general port pin characteristics on page 161*: Modified V_{OL}, V_{OH} data for V_{DD} = 5 V Modified *Figure 102 on page 184, Figure 103 on page 185* and *Figure 104 on page 186* *Table 104: 10-bit ADC characteristics on page 174*: Modified f_{ADC} parameter to include a maximum value of 2 MHz at 3 V ≤ V_{DD} ≤ 3.6 V Updated *ST7L1 FASTROM and ROM microcontroller option list on page 187*

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com