ST7LUS5, ST7LU05, ST7LU09

8-bit low cost, low pin-count MCU for automotive, with single voltage Flash memory, ADC, timers

Features

Memories

- 1 to 2 Kbytes single voltage Flash program memory with readout protection, in-circuit and in-application programming (ICP and IAP), 10k write/erase cycles guaranteed, data retention 20 years
- 128 bytes RAM
- Up to 128 bytes data EEPROM, 300k write/erase cycles guaranteed, data retention 20 years

Clock, reset and supply management

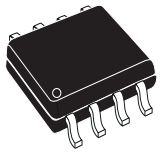
- Internal low voltage supervisor (LVD) for reliable power-on/-off start-up procedures
- Clock sources: Internal trimmable 8 MHz RC oscillator, internal low power, low frequency RC oscillator or external clock
- 5 power saving modes: Halt, auto wake-up from halt, active halt, wait and slow

Interrupt management

- 11 interrupt vectors plus TRAP and reset
- 5 external interrupt lines (on 5 vectors)

I/O ports

- 5 multifunctional bidirectional I/O lines
- 1 additional output line
- 6 alternate function lines
- 5 high sink outputs



SO8 150-mil (1)

2 timers

- One 8-bit lite timer with prescaler including: watchdog, 1 real-time base and 1 input capture
- One 12-bit auto-reload timer with output compare function and PWM

A/D converter

- 10-bit resolution for 0 to V_{DD}
- 5 input channels

Instruction set

- 8-bit data manipulation
- 63 basic instructions with illegal opcode detection
- 17 main addressing modes
- 8 x 8 unsigned multiply instruction

Development tools

- Full hardware/software development package
- Debug module

Table 1. Device summary

Device	Flash program memory	RAM/EEPROM	Peripherals	Operating voltage	CPU frequency	Operating temp.
ST7LUS5	1 Kbyte	128 bytes RAM/no EEPROM	Lite timer with watchdog, auto-reload timer with 1 PWM, 10-bit ADC	3.0V to 5.5V @ f_{CPU} = 8 MHz	Up to 8 MHz RC	-40°C to +85°C/ -40°C to 125°C
ST7LU05	2 Kbytes					
ST7LU09		128 bytes RAM/128 bytes EEPROM				

1. DIP16 package available for development or tool prototyping purposes only. Not orderable in production quantities.

Contents

List of tables

List of figures

1 Description

The ST7LUxx devices are members of the ST7 microcontroller family designed for mid-range automotive applications running from 3 to 5.5V.

All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with Flash or preprogramed memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code.

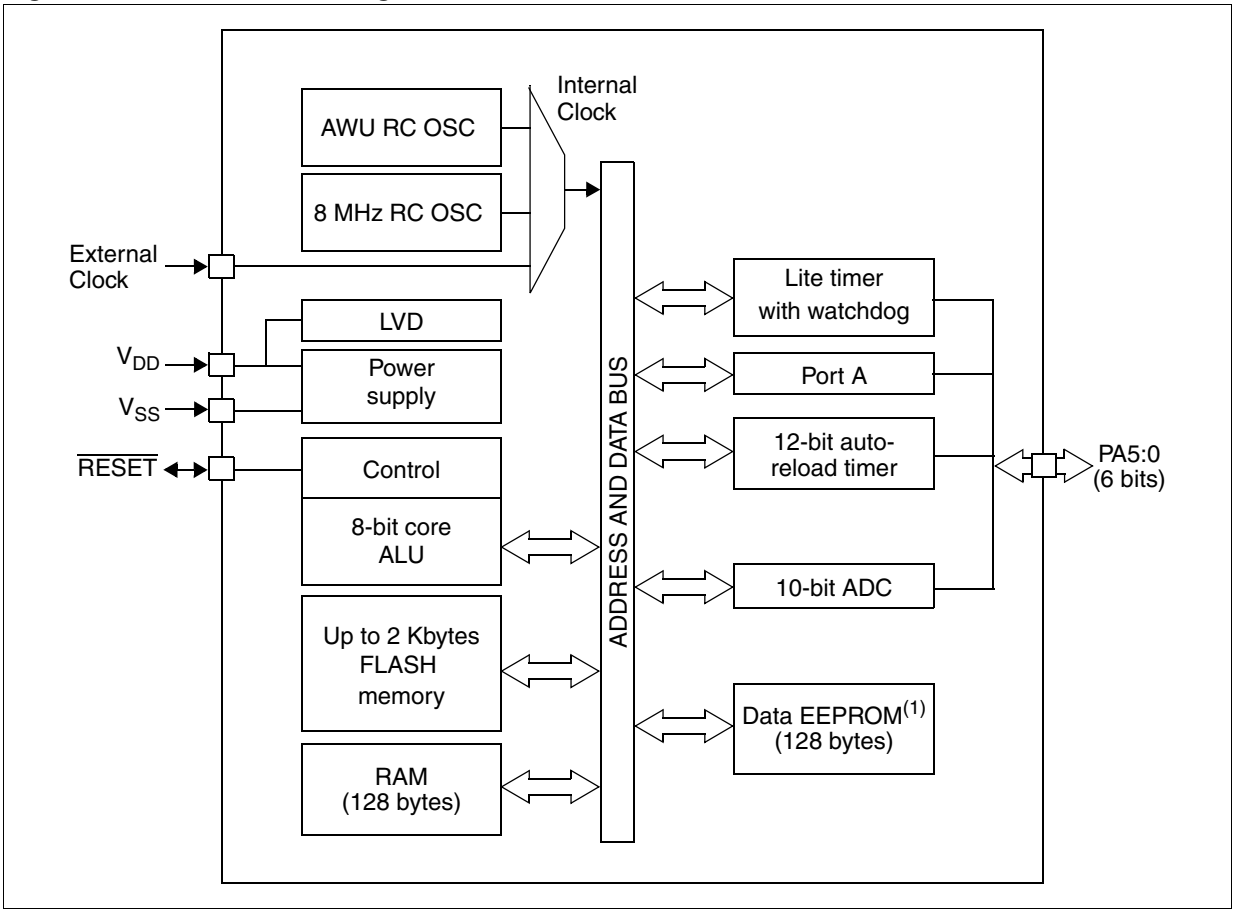
The on-chip peripherals include an A/D converter, a lite timer and and an auto-reload timer. For power economy, the microcontroller can switch dynamically into halt, auto wake-up from halt, active halt, wait, or slow mode when the application is in idle or standby state.

The devices feature 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.*

Typical applications include:

- All types of car body applications such as window lift, DC motor control, sunroof
- Remote keyless entry (RKE)
- Safety microcontroller (sub-MCU)

Figure 1. General block diagram



1. Optional on certain devices

2 Package pinout and device pin description

Figure 2. 8-pin SO package pinout

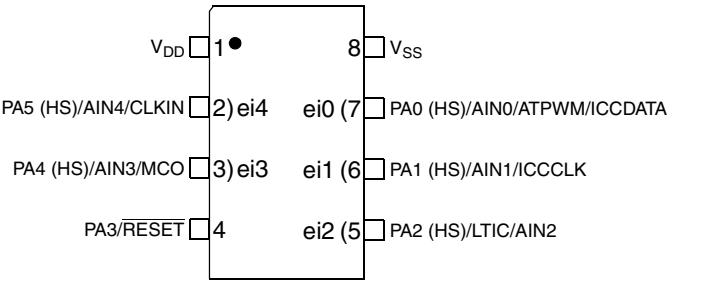
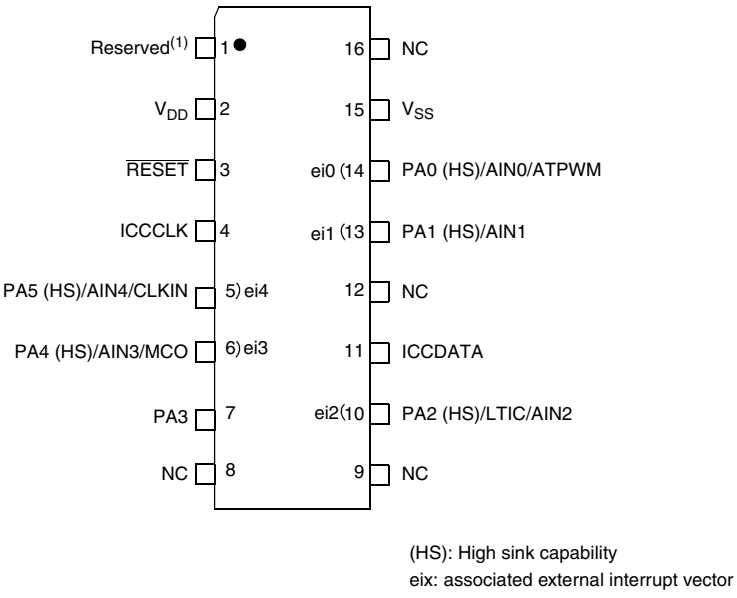


Figure 3. 16-pin DIP package pinout(a)



1. Must be tied to ground

Note: The differences between 16-pin and 8-pin packages are listed below:

1. The ICC signals (ICCCLK and ICCDATA) are mapped on dedicated pins.

2. The reset signal is mapped on a dedicated pin. It is not multiplexed with PA3.

3. PA3 pin is always configured as output. Any change on multiplexed IO reset control registers (MUXCR1 and MUXCR2) will have no effect on PA3 functionality. Refer to Section 4.6: Register description.

a. For development or tool prototyping purposes only. Package not orderable in production quantities.

Table 2. Device pin description(1)

Pin		Type(2)	Level		Port / Control						Function	
					Input(5)				Output(6)		Main (after reset)	Alternate
No.	Name		Input(3)	Output(4)	float	wpu	int	ana	OD	PP		
1	V_{DD}(7)	S									Main power supply	
2	PA5/AIN4/CLKIN	I/O	C_T	HS	**X**	ei4		X	X	X	**Port A5**	Analog input 4 or external clock input
3	PA4/AIN3/MCO	I/O	C_T	HS	**X**	ei3		X	X	X	**Port A4**	Analog input 3 or main clock output
4	PA3/$\overline{\text{RESET}}$(8)	O				**X**			X	X	**Port A3**	Reset(8)
5	PA2/AIN2/LTIC	I/O	C_T	HS	**X**	ei2		X	X	X	**Port A2**	Analog input 2 or lite timer input capture
6	PA1/AIN1/ICCCLK	I/O	C_T	HS	**X**	ei1		X	X	X	**Port A1**	Analog input 1 or in-circuit communication clock **Caution:** During normal operation this pin must be pulled-up, internally or externally (external pull-up of 10k mandatory in noisy environment).
7	PA0/AIN0/ATPWM/ICCDATA	I/O	C_T	HS	**X**	ei0		X	X	X	**Port A0**	Analog input 0 or auto-reload timer PWM or in-circuit communication data
8	V_{SS}(7)	S									Ground	

1. The reset configuration of each pin is shown in bold which is valid as long as the device is in reset state.
2. Type: I = input, O = output, S = supply
3. In/Output level: C_T = CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger
4. Output level: HS = High sink (on N-buffer only)
5. Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog
6. Output: OD = open drain, PP = push-pull
7. It is mandatory to connect all available V_{DD} and V_{DDA} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.
8. After a reset, the multiplexed PA3/$\overline{\text{RESET}}$ pin will act as $\overline{\text{RESET}}$. To configure this pin as output (Port A3), write 55h to MUXCR0 and AAh to MUXCR1. For further details, please refer to *Section 6.4: Register description on page 42*.

3 Register and memory map

As shown in *Figure 6* and *Figure 8*, the MCU is capable of addressing 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, 128 bytes of RAM and 1 Kbyte or 2 Kbytes of user program memory. The RAM space includes up to 64 bytes for the stack from 00C0h to 00FFh.

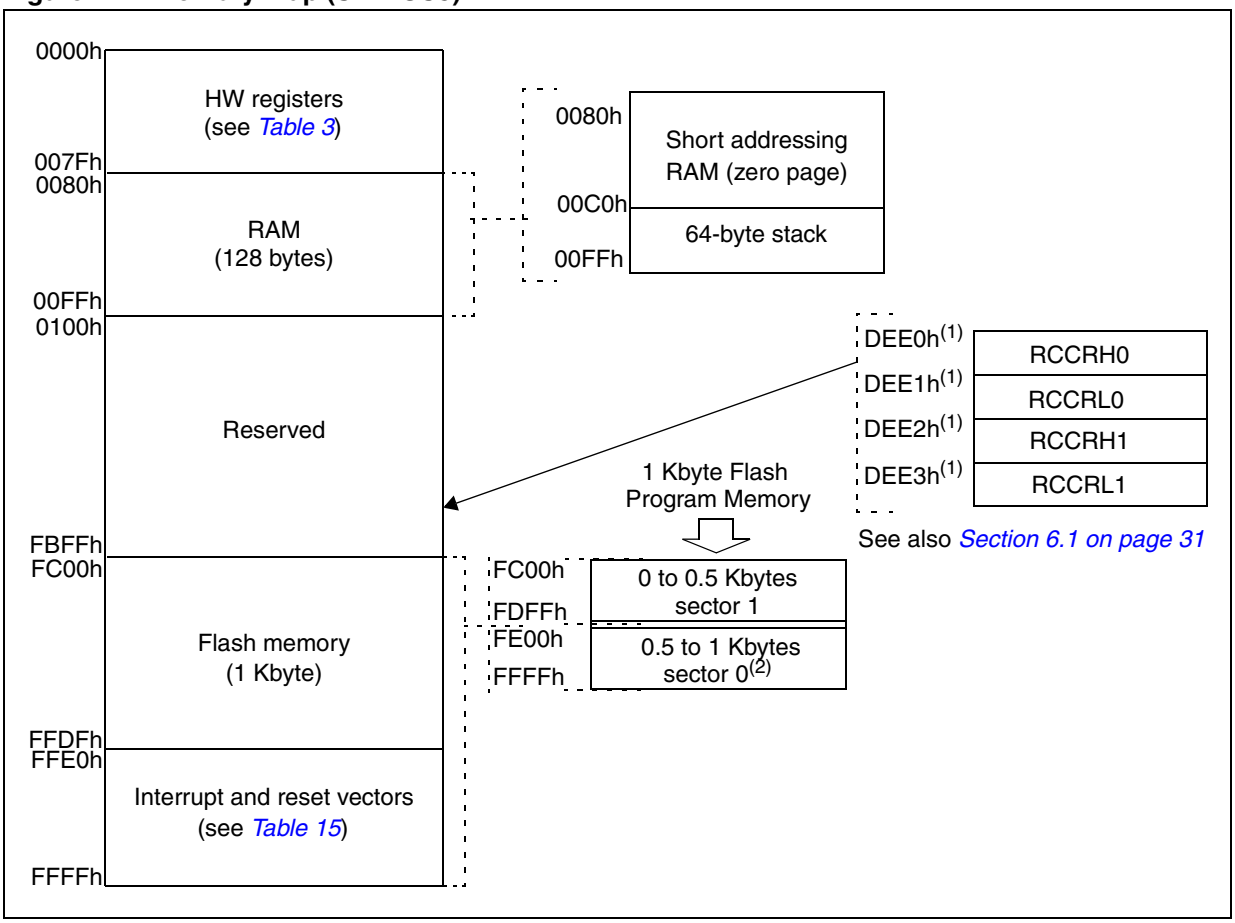
The highest address bytes contain the user reset and interrupt vectors.

The Flash memory contains two sectors (see *Figure 6* and *Figure 8*) mapped in the upper part of the ST7 addressing space whereas the reset and interrupt vectors are located in Sector 0 (FE00h-FFFFh for the ST7LUS5 and FA01h-FFFFh for the ST7LU05/ST7LU09).

The size of Flash sector 0 and other device options are configurable by the option bytes.

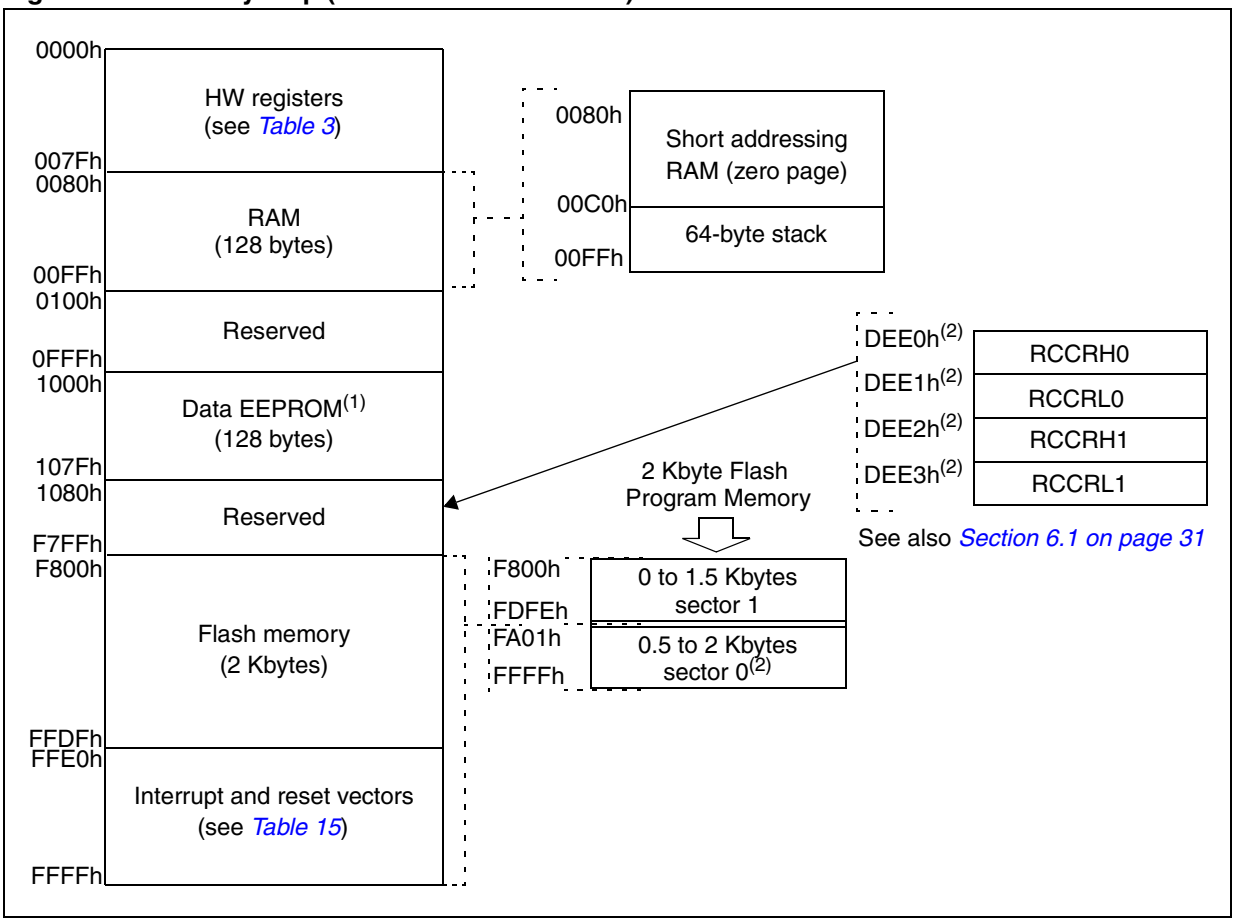
Caution: Memory locations marked as “Reserved” must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 4. Memory map (ST7LUS5)



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 addresses.
2. Size configurable by option byte

Figure 5. Memory map (ST7LU05 and ST7LU09)



1. ST7LU09 only; area reserved in ST7LU05
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 values locations) has been erased (after the readout protection removal), then the RC calibration values can still be obtained through these addresses.
3. Configurable by option byte

Table 3. Hardware register map(1)

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h 0002h	Port A	PADR PADDR PAOR	Port A data register Port A data direction register Port A option register	00h(2) 08h 02h(3)	R/W R/W R/W
0003h-000Ah	Reserved area (8 bytes)				
000Bh 000Ch	Lite timer	LTCSR LTICR	Lite timer control/status register Lite timer input capture register	0xh 00h	R/W Read Only
000Dh 000Eh 000Fh 0010h 0011h 0012h 0013h	Auto-reload timer	ATCSR CNTRH CNTRL ATRH ATRL PWMCR PWM0CSR	Timer control/status register Counter register high Counter register low Auto-reload register high Auto-reload register low PWM output control register PWM 0 control/status register	00h 00h 00h 00h 00h 00h 00h	R/W Read Only Read Only R/W R/W R/W R/W
0014h to 0016h	Reserved area (3 bytes)				
0017h 0018h	Auto-reload timer	DCR0H DCR0L	PWM 0 duty cycle register high PWM 0 duty cycle register low	00h 00h	R/W R/W
0019h to 002Eh	Reserved area (22 bytes)				
0002Fh	Flash	FCSR	Flash control/status register	00h	R/W
0030h to 0033h	Reserved area (4 bytes)				
0034h 0035h 0036h	ADC	ADCCSR ADCDRH ADCDRL	A/D control status register A/D data register high A/D data register low	00h xxh 00h	R/W Read Only R/W
0037h	ITC	EICR1	External interrupt control register 1	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 0000 0x00b	R/W R/W
003Bh to 003Ch	Reserved area (2 bytes)				
003Dh	ITC	EICR2	External interrupt control register 2	00h	R/W
003Eh	RC prescaler	INTRCPRR	Internal RC prescaler selection register	03h	R/W
003Fh	Clock controller	CKCNTCSR	Clock controller control/status register	09h	R/W
0040h to 0046h	Reserved area (7 bytes)				
0047h 0048h	MuxIO-reset	MUXCR0 MUXCR1	Mux IO-reset control register 0 Mux IO-reset control register 1	00h 00h	R/W R/W
0049h 004Ah	AWU	AWUPR AWUCSR	AWU prescaler register awu Control/Status Register	FFh 00h	R/W R/W

Table 3. Hardware register map(1) (continued)

Address	Block	Register label	Register name	Reset status	Remarks
004Bh	DM(4)	DMCR	DM control register	00h	R/W
004Ch		DMSR	DM status register	00h	R/W
004Dh		DMBK1H	DM breakpoint register 1 high	00h	R/W
004Eh		DMBK1L	DM breakpoint register 1 low	00h	R/W
004Fh		DMBK2H	DM breakpoint register 2 high	00h	R/W
0050h		DMBK2L	DM breakpoint register 2 low	00h	R/W
0051h to 007Fh	Reserved area (47 bytes)				

1. 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 DM registers, see the *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 or in-application programming.

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

4.2 Main features

- ICP (in-circuit programming)
- IAP (in-application programming)
- ICT (in-circuit testing) 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 and option byte row can be programmed or erased.
- In-circuit programming - In this mode, Flash sectors 0 and 1 and option byte row can be programmed or erased without removing the device from the application board.
- In-application programming - In this mode, sector 1 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 cable. ICP is performed in three steps:

1. 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.
2. Download ICP driver code in RAM from the ICCDATA pin
3. 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).

This mode is fully controlled by user software. This allows it to be adapted to the user application (user-defined strategy for entering programming mode or choice of communications protocol used to fetch the data to be stored, etc.).

IAP 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

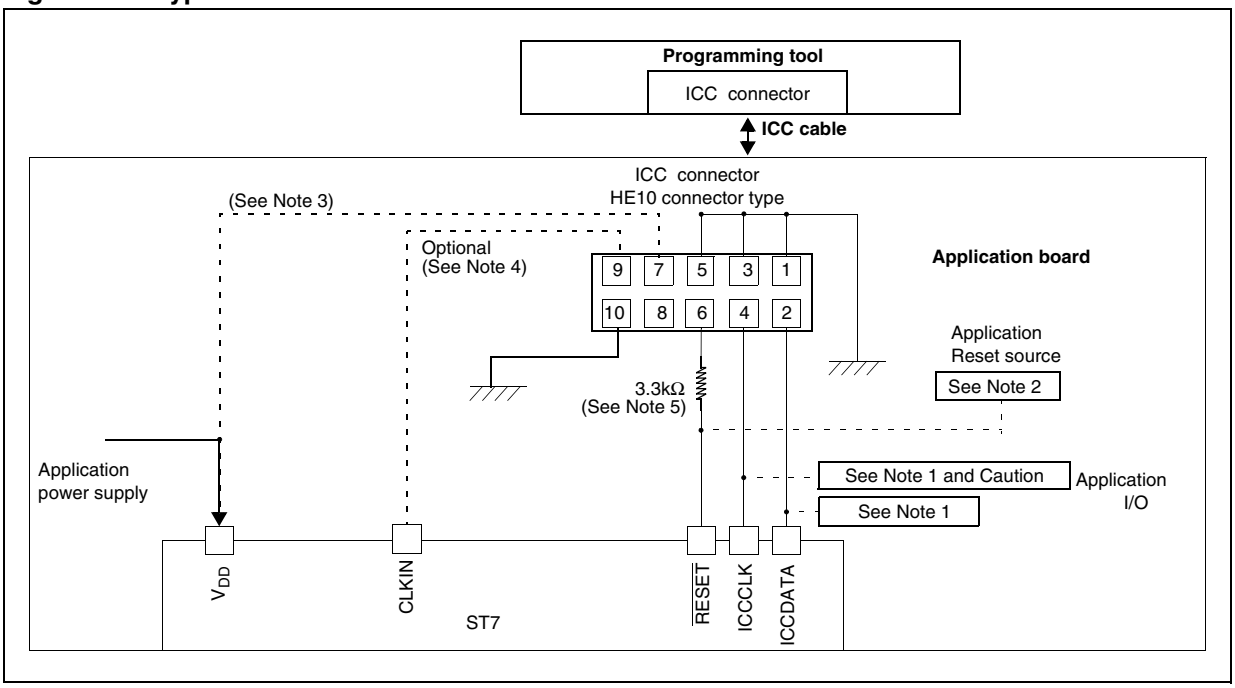
ICP needs a minimum of 4 and up to 6 pins to be connected to the programming tool. These pins are:

- $\overline{\text{RESET}}$: Device reset
- V_{SS}: Device power supply ground
- ICCCLK: ICC output serial clock pin[b]
- ICCDATA: ICC input serial data pin
- CLKIN: Main clock input for external source
- V_{DD}: Application board power supply[c]

b. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the programming tool documentation for recommended resistor values.

c. 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.

Figure 6. 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 has to 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 5mA at high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor>1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the CLKIN pin of the ST7 when ICC mode is selected with option bytes disabled (35-pulse ICC entry mode). When option bytes are enabled (38-pulse ICC entry mode), the internal RC clock (internal RC or AWU RC) is forced. If internal RC is selected in the option byte, the internal RC is provided. If AWU RC or external clock is selected, the AWU RC oscillator is provided.
5. A serial resistor must be connected to ICC connector pin 6 in order to prevent contention on PA3/$\overline{\text{RESET}}$ pin. Contention may occur if a tool forces a state on $\overline{\text{RESET}}$ pin while PA3 pin forces the opposite state in output mode. The resistor value is defined to limit the current below 2mA at 5V. If PA3 is used as output push-pull, then the application must be switched off to allow the tool to take control of the $\overline{\text{RESET}}$ pin (PA3). To allow the programming tool to drive the $\overline{\text{RESET}}$ pin below V_{IL}, special care must also be taken when a pull-up is placed on PA3 for application reasons.

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 will put it back in input pull-up.

4.5 Memory protection

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

4.5.1 Readout protection

Readout protection, when selected provides a protection 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. Program memory is protected.

In Flash devices, this protection is removed by reprogramming the option. In this case, program memory is 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 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. 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.

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*.

4.6 Register description

4.6.1 Flash control/status register (FCSR)

1st RASS Key: 0101 0110 (56h)
2nd RASS Key: 1010 1110 (AEh)

FCSR Reset value: 0000 0000 (00h)

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

Table 4. Flash register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Fh	FCSR Reset value	0	0	0	0	0	OPT 0	LAT 0	PGM 0

5 Central processing unit

5.1 Introduction

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

5.2 Main features

- 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

5.3 CPU registers

The six CPU registers shown in *Figure 10* are not present in the memory mapping and are accessed by specific instructions.

5.3.1 Accumulator (A)

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

5.3.2 Index registers (X and Y)

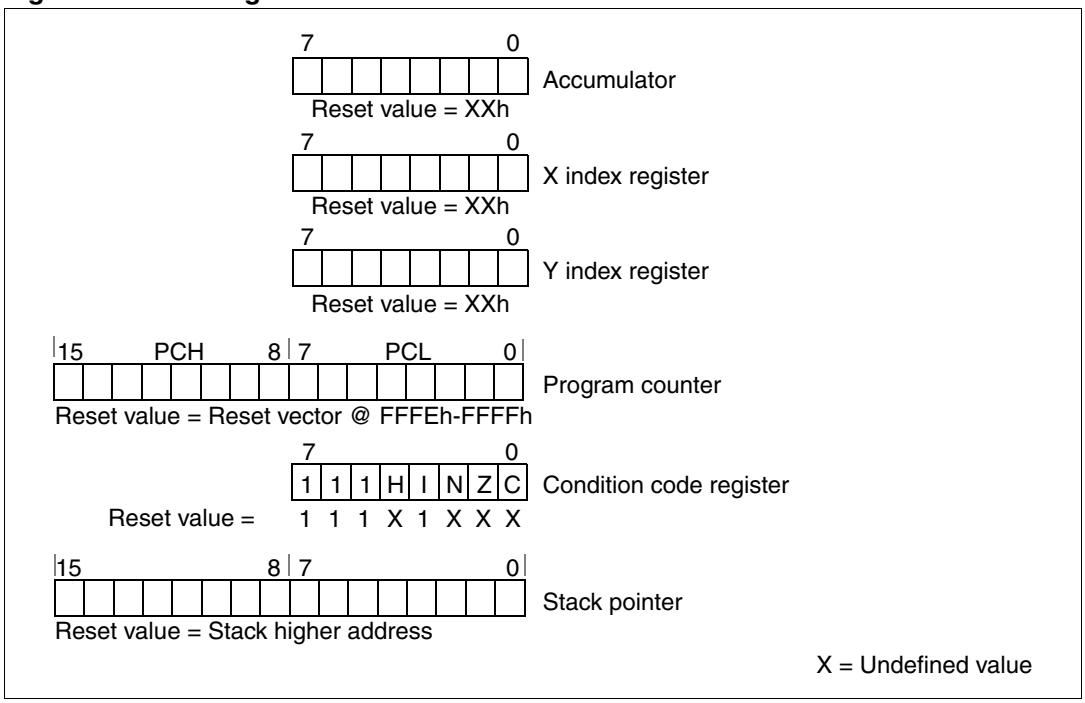
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation (the cross-assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register).

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

5.3.3 Program counter (PC)

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

Figure 7. CPU registers



5.3.4 Condition code register (CC)

CC Reset value: 111x1xxx

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

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

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

Table 5. CC register description

Bit	Name	Function
7:5	-	Reserved, must remain set.
4	H	Half carry This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions. 0: No half carry has occurred. 1: A half carry has occurred. This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Table 5. CC register description (continued)

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 is a copy of the 7th bit of the result. 0: The result of the last operation is positive or null. 1: The result of the last operation is negative (that is, the most significant bit is a logic 1). This bit is accessed by the JRMI and JRPL 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.

5.3.5 Stack pointer (SP)

SP — Reset value: 00 FFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	1	SP[5:0]					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W					

The stack pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see *Figure 11*).

Since the stack is 64 bytes deep, the 10 most significant bits are forced by hardware. Following an MCU reset, or after a reset stack pointer (RSP) instruction, the stack pointer contains its reset value (the SP5 to SP0 bits are set) which is the stack higher address.

The stack pointer least significant byte, called “S”, can be directly accessed by a load (LD) instruction.

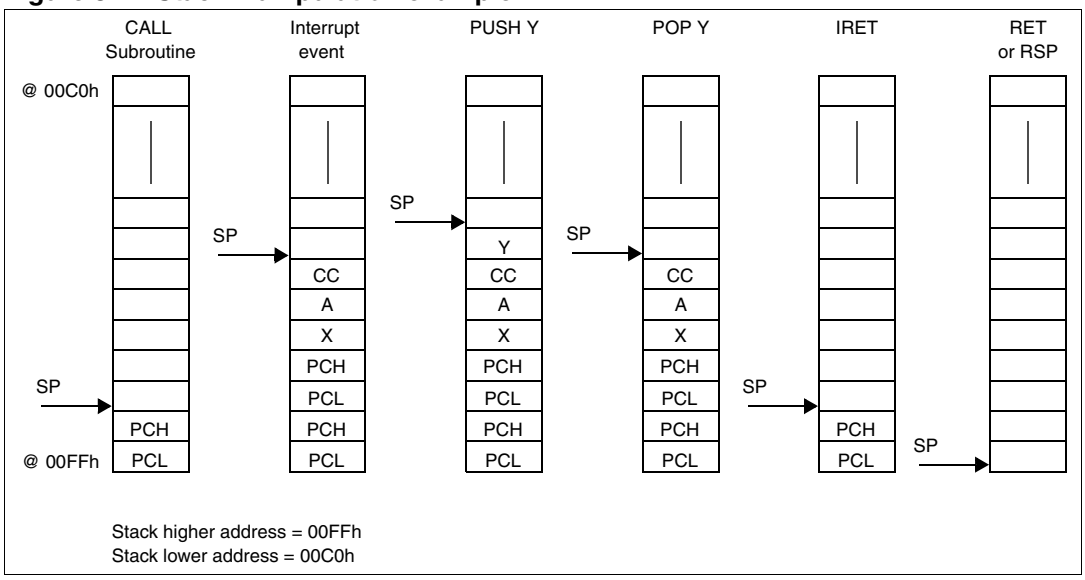
Note: When the lower limit is exceeded, the stack pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

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 11*.

- 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 8. Stack manipulation example



6 Supply, reset and clock management

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

Main features

- Clock management
  - 8 MHz internal RC oscillator (enabled by option byte)
  - External clock input (enabled by option byte)
- Reset Sequence Manager (RSM)
- System integrity management (SI) - Main supply low voltage detection (LVD) with reset generation (enabled by option byte)

6.1 Internal RC oscillator adjustment

The ST7 contains an internal RC oscillator with a specific accuracy for a given device, temperature and voltage. It can be selected as the start up clock through the CKSEL[1:0] option bits (see *Section 14.1 on page 131*). It must be calibrated to obtain the frequency required in the application. This is done by 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 Flash memory for 3.3 and 5V V_{DD} supply voltages at T_{Amax}, as shown in the following *Table 6*.

Table 6. RCCR calibration values

RCCR	Conditions	ST7LUxx address
RCCRH0	V_{DD} = 5V, T_{Amax}, f_{RC} = 8 MHz(1)	DEE0h(2) (CR[9:2] bits)
RCCRL0		DEE1h(2) (CR[1:0] bits)
RCCRH1	V_{DD} = 3.3V, T_{Amax}, f_{RC} = 8 MHz(1)	DEE2h(2) (CR[9:2] bits)
RCCRL1		DEE3h(2) (CR[1:0] bits)

1. xxRCCR0 and RCCR1 calibrated within these conditions in order to reach RC accuracy as mentionned in *Table 65: General operating conditions on page 104*
2. DEE0h, DEE1h, DEE2h and DEE3h 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 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 two addresses.

Note: 1 In ICC mode, the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte. Refer to Note 5 in Section 4.4 for further details.

2 See Chapter 12: Electrical characteristics on page 102 for more information on the frequency and accuracy of the RC oscillator.

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.

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

Refer to application note AN2326 for information on how to calibrate the RC frequency using an external reference signal.

The ST7LUxx also contains an auto wake-up RC oscillator. This RC oscillator should be enabled to enter auto wake-up from halt mode.

The auto wake-up RC oscillator can also be configured as the startup clock through the CKSEL[1:0] option bits (see *Section 14.1 on page 131*).

This is recommended for applications where very low power consumption is required.

Switching from one startup clock to another can be done in run mode as follows (see *Figure 12*):

Case 1: Switching from internal RC to AWU

1. Set the RC/AWU bit in the CKCNTCSR register to enable the AWU RC oscillator.
2. The RC_FLAG is cleared and the clock output is at 1.
3. Wait 3 AWU RC cycles until the AWU_FLAG is set.
4. The switch to the AWU clock is made at the positive edge of the AWU clock signal.
5. Once the switch is made, the internal RC is stopped.

Case 2: Switching from AWU RC to internal RC

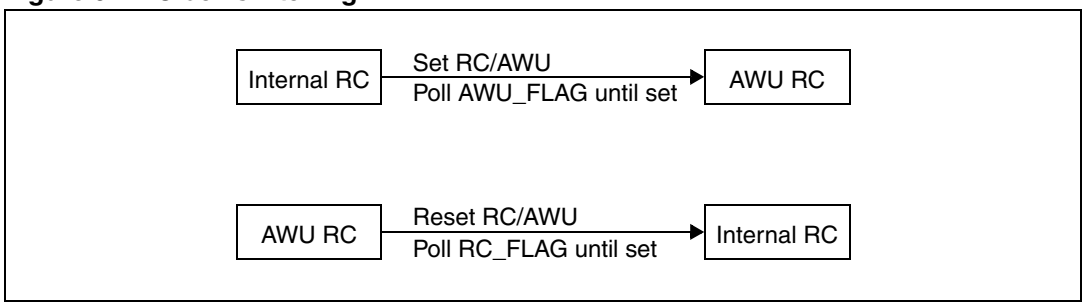
1. Reset the RC/AWU bit to enable the internal RC oscillator.
2. Using a 4-bit counter, wait until 8 internal RC cycles have elapsed. The counter is running on internal RC clock.
3. Wait until the AWU_FLAG is cleared (1 AWU RC cycle) and the RC_FLAG is set (2 RC cycles).
4. The switch to the internal RC clock is made at the positive edge of the internal RC clock signal.
5. Once the switch is made, the AWU RC is stopped.

Note: 1 When the internal RC is not selected, it is stopped so as to save power consumption.

2 When the internal RC is selected, the AWU RC is turned on by hardware when entering Auto Wake-up from Halt mode.

3 When the external clock is selected, the AWU RC oscillator is always on.

Figure 9. Clock switching



6.2 Register description

6.2.1 Main clock control/status register (MCCSR)

MCCSR Reset value: 0000 0000 (00h)

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

Table 7. MCCSR register description

Bit	Name	Function
7:2	-	Reserved, must be kept cleared.
1	MCO	Main clock out enable This bit is read/write by software and cleared by hardware after a reset. This bit allows to enable the MCO output clock. 0: MCO clock disabled; I/O port free for general purpose I/O 1: MCO clock enabled
0	SMS	Slow mode select This bit is read/write by software and cleared by hardware after a reset. This bit selects the input clock f_{OSC} or $f_{OSC}/32$. 0: Normal mode ($f_{CPU} = f_{OSC}C$) 1: Slow mode ($f_{CPU} = f_{OSC}C/32$)

6.2.2 RC control register (RCCR)

RCCR Reset value: 1111 1111 (FFh)

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

Table 8. RCCR register description

Bit	Name	Function
7:0	CR[9:2]	RC oscillator frequency adjustment bits These bits, as well as CR[1:0] bits in the SICSR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain the required accuracy. The application can store the correct value for each voltage range in Flash memory and write it to this register at start-up. 00h = maximum available frequency FFh = lowest available frequency *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.*

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

SICSR Reset value: 0000 0x00 (0xh)

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

Table 9. SICSR register description

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 the required accuracy. Refer to *Section 6.1 on page 31*.
4:3	-	Reserved, must be kept cleared.
2:0	LVDRF	System integrity bits Refer to *Section 7.4: System integrity management (SI)*.
1:0	-	Reserved, must be kept cleared.

6.2.4 Internal RC prescaler selection register (INTRCPRR)

INTRCPRR Reset value: 0000 0011 (03h)

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

Table 10. INTRCPRR register description

Bit	Name	Function
7:5	CK[2:0]	Internal RC prescaler selection These bits are set by software and cleared by hardware after a reset. These bits select the prescaler of the internal RC oscillator as follows (see also *Figure 13 on page 37*): 000: f_{OSC} = 8 MHz 001: f_{OSC} = 4 MHz 010: f_{OSC} = 2 MHz 011: f_{OSC} = 1 MHz 100: f_{OSC} = 500 kHz *Notes:* *1. If the internal RC is used with a supply operating range below 3.3V, a division ratio of at least 2 must be enabled in the RC prescaler.* *2. Configuration '000' is not allowed at 3.3V operation.*
4:0	-	Reserved, must be kept cleared.

6.2.5 Clock controller control/status register (CKCNTCSR)

CKCNTCSR Reset value: 0000 1001 (09h)

7	6	5	4	3	2	1	0
Reserved				AWU_FLAG	RC_FLAG	Reserved	RC/AWU
-				R/W	R/W	-	R/W

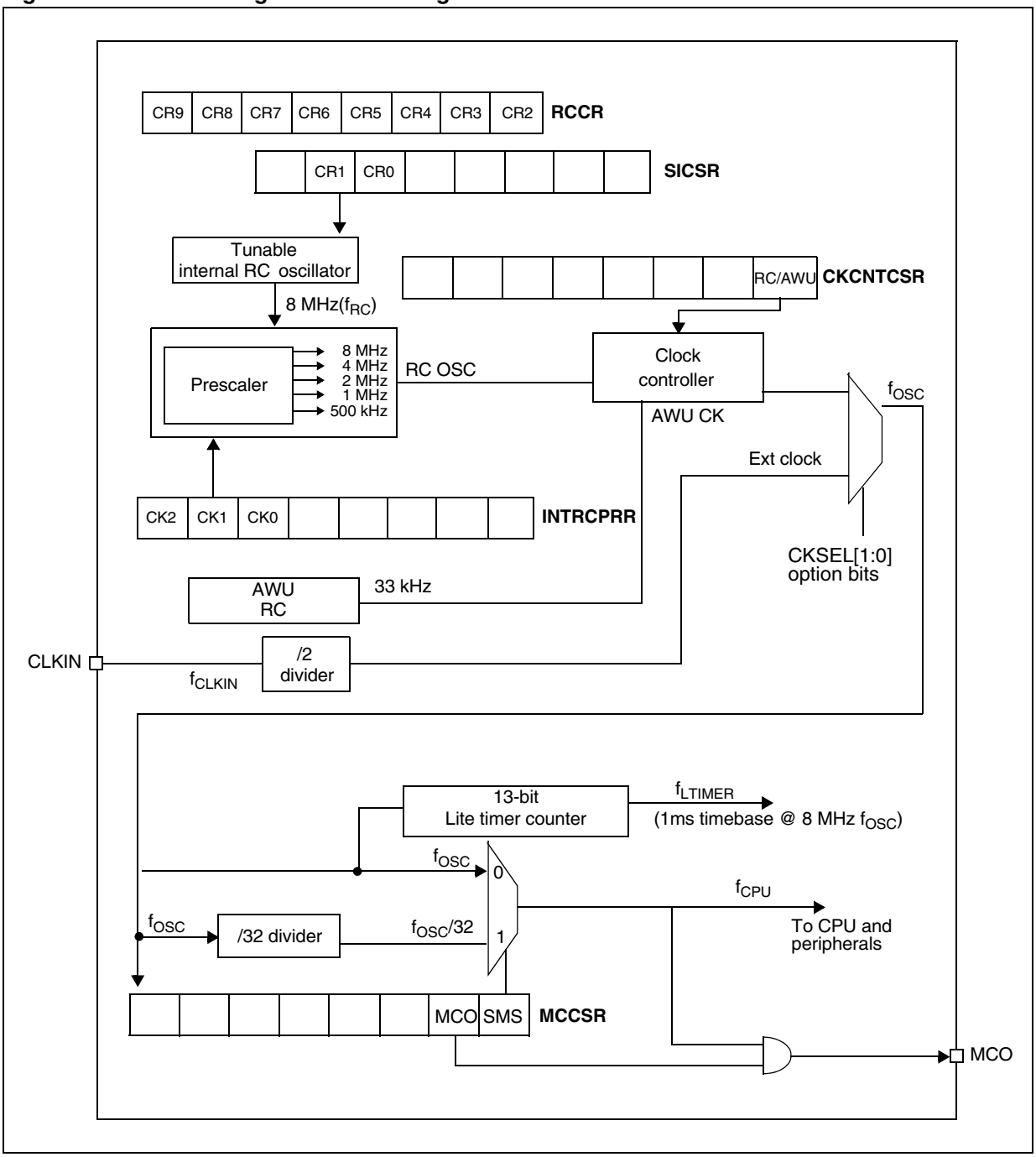
Table 11. CKCNTCSR register description

Bit	Name	Function
7:4	-	Reserved, must be kept cleared.
3	AWU_FLAG	AWU selection This bit is set and cleared by hardware. 0: No switch from AWU to RC requested 1: AWU clock activated and temporization completed
2	RC_FLAG	RC Selection This bit is set and cleared by hardware. 0: No switch from RC to AWU requested 1: RC clock activated and temporization completed
1	-	Reserved, must be kept cleared.
0	RC/AWU	RC/AWU selection 0: RC enabled 1: AWU enabled (default value)

Table 12. Clock register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0038h	MCCSR Reset value							MCO 0	SMS 0
0039h	RCCR Reset value	CR9 1	CR8 1	CR7 1	CR6 1	CR5 1	CR4 1	CR3 1	CR2 1
003Ah	SICSR Reset value		CR1 0	CR0 0			LVDRF x		
003Eh	INTRCPRR Reset value	CK2 0	CK1 0	CK0 0				1	1
003Fh	CKCNTCSR Reset value					AWU_FLAG 1	RC_FLAG 0		RC/AWU 1

Figure 10. Clock management block diagram



6.3 Reset sequence manager (RSM)

6.3.1 Introduction

The reset sequence manager includes three reset sources as shown in *Figure 15*:

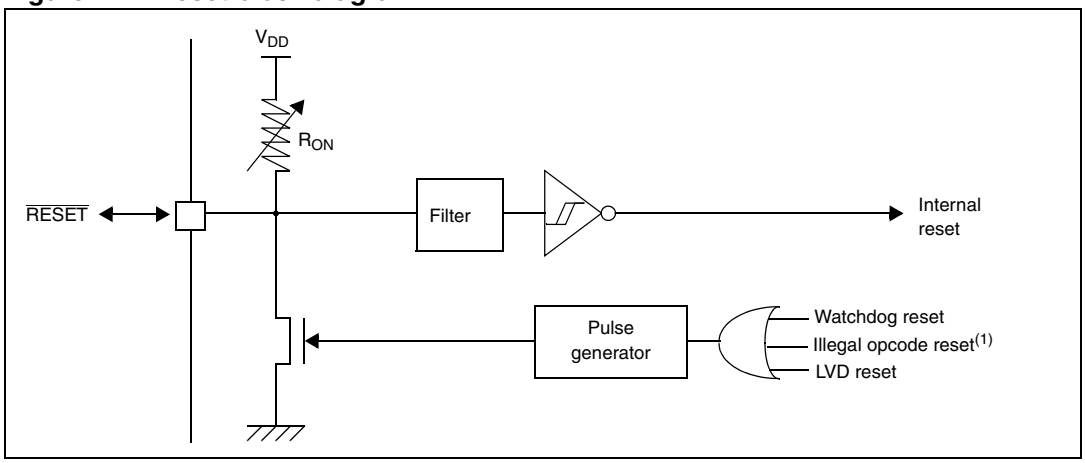
- External $\overline{\text{RESET}}$ source pulse
- Internal LVD reset (low voltage detection)
- Internal watchdog reset

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Figure 15.

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.

Figure 11. Reset block diagram



1. See *Illegal opcode reset on page 99* for more details on illegal opcode reset conditions.

The basic reset sequence consists of three phases as shown in *Figure 16*:

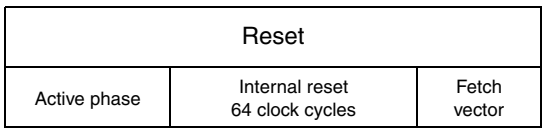
- Active phase depending on the reset source
- 64 CPU clock cycle delay
- Reset vector fetch

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 $\overline{\text{RESET}}$ pin in low state until programming mode is entered, in order to avoid unwanted behavior.

The 64 CPU clock cycle delay allows the oscillator to stabilize and ensures that recovery has taken place from the reset state.

The reset vector fetch phase duration is 2 clock cycles.

Figure 12. Reset sequence phases



6.3.2 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 *Chapter 12: 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 17*). This detection is asynchronous and therefore the MCU can enter reset state even in halt mode.

The $\overline{\text{RESET}}$ pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in *Chapter 12: Electrical characteristics*.

6.3.3 External power-on reset

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

6.3.4 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 17*.

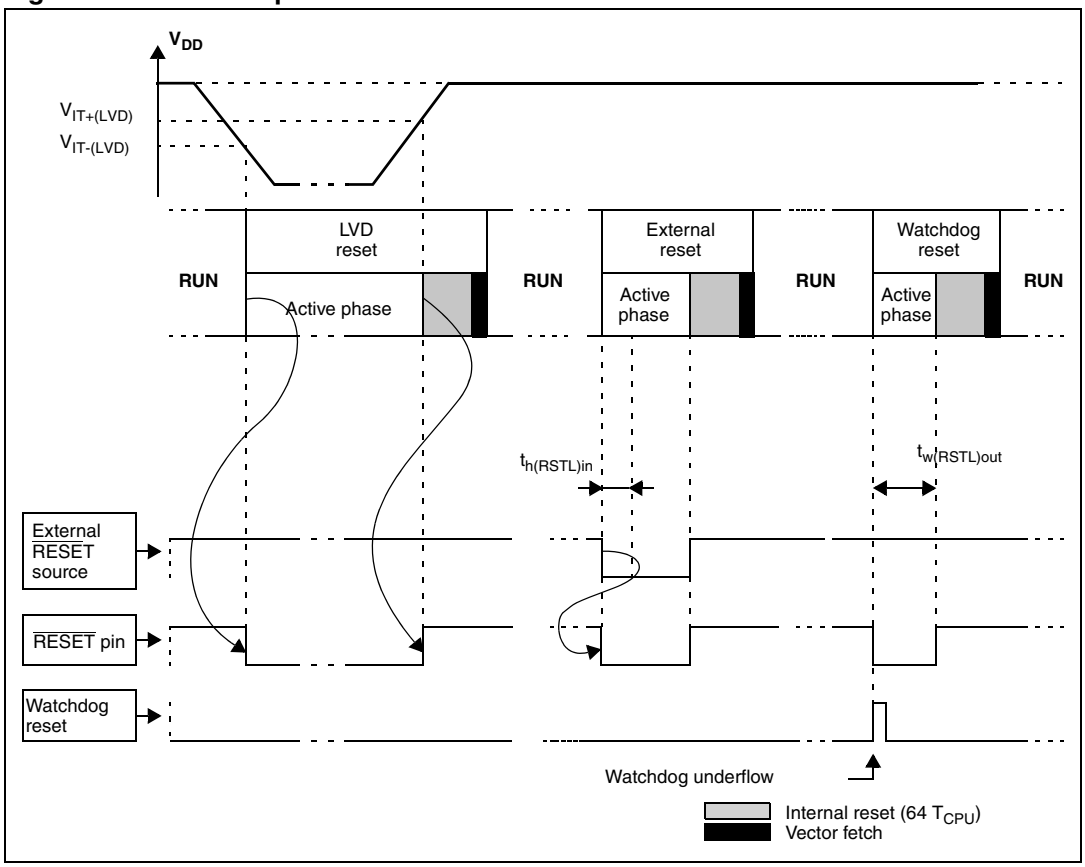
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

6.3.5 Internal watchdog reset

The reset sequence generated by an internal watchdog counter overflow is shown in *Figure 17*.

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 13. Reset sequences



6.4 Register description

6.4.1 Multiplexed I/O reset control register 1 (MUXCR1)

MUXCR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MIR[15:8]							

Read/write once only

6.4.2 Multiplexed I/O reset control register 0 (MUXCR0)

MUXCR0 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MIR[7:0]							

Read/write once only

Table 13. MUXCRx register description

Bit	Name	Function
15:0	MIR[15:0]	This 16-bit register is read/write by software but can be written only once between two reset events. It is cleared by hardware after a reset. When both MUXCR0 and MUXCR1 registers are at 00h, the multiplexed PA3/$\overline{\text{RESET}}$ pin acts as $\overline{\text{RESET}}$. To configure this pin as output (Port A3), write 55h to MUXCR0 and AAh to MUXCR1. These registers are one-time writable only. – To configure PA3 as general purpose output: After power-on/reset, the application program has to configure the I/O port by writing to these registers as described above. Once the pin is configured as an I/O output, it cannot be changed back to a $\overline{\text{RESET}}$ pin by the application code. – To configure PA3 as $\overline{\text{RESET}}$: An internally generated reset (such as POR, LVD, WDG, illegal opcode) will clear the two registers and the pin will act again as a reset function. Otherwise, a power-down is required to put the pin back in reset configuration.

Table 14. Multiplexed I/O register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0047h	MUXCR0 Reset value	MIR7 0	MIR6 0	MIR5 0	MIR4 0	MIR3 0	MIR2 0	MIR1 0	MIR0 0
0048h	MUXCR1 Reset value	MIR15 0	MIR14 0	MIR13 0	MIR12 0	MIR11 0	MIR10 0	MIR9 0	MIR8 0

7 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 46* and a non-maskable software interrupt (TRAP). The interrupt processing flowchart is shown in *Figure 18*.

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 *Section 7.2: External interrupts*).

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, a 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*).

7.1 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*.

7.2 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.

7.3 Peripheral interrupts

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

- The I bit of the CC register is cleared.
- The corresponding enable bit is set in the control register.

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

An interrupt request is cleared by doing one of the following:

- Writing ‘0’ to the corresponding bit in the status register
- Accessing 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 14. Interrupt processing flowchart

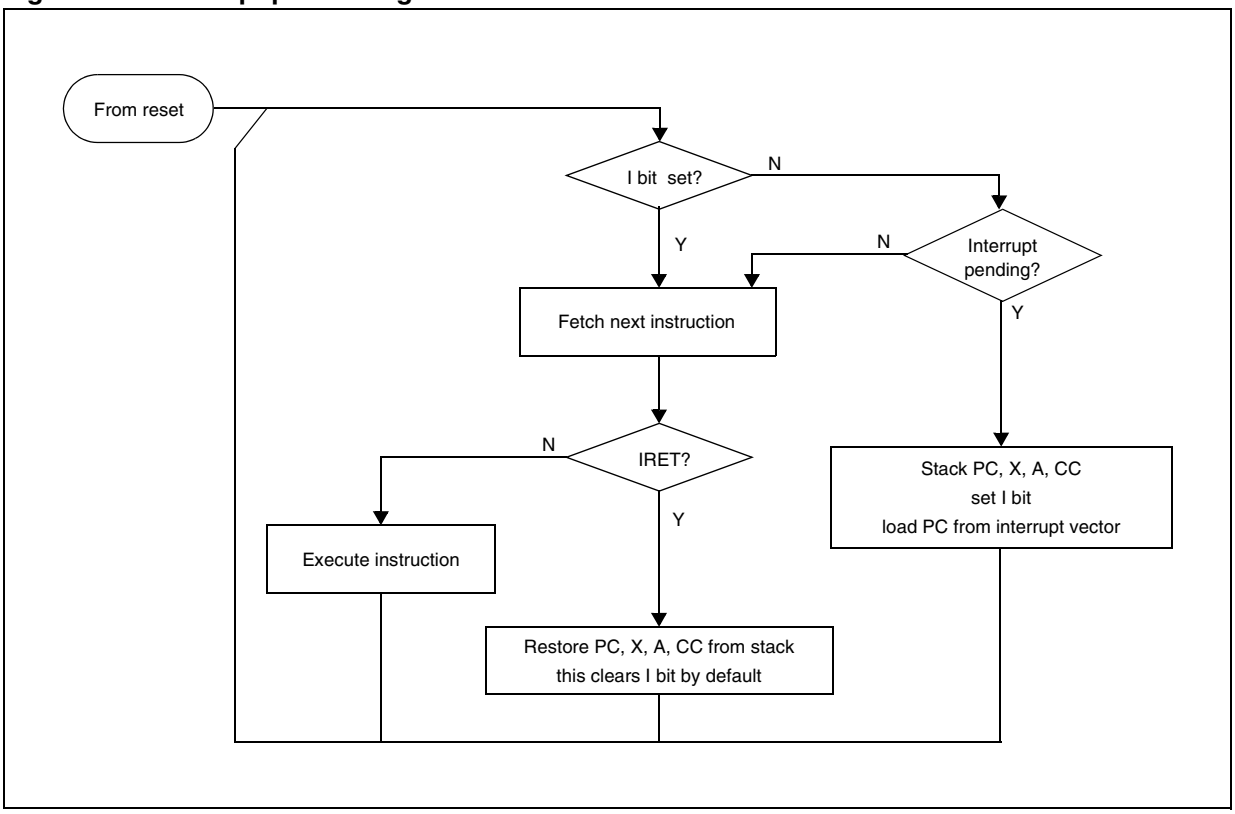


Table 15. Interrupt mapping

No.	Source block	Description	Register label	Priority order	Exit from HALT	Address vector
	RESET	Reset	N/A	Highest priority ↓ Lowest priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
0	AWU	Auto wake-up interrupt	AWUCSR		yes(1)	FFFAh-FFFBh
1	ei0	External interrupt 0	N/A		yes	FFF8h-FFF9h
2	ei1	External interrupt 1				FFF6h-FFF7h
3	ei2	External interrupt 2				FFF4h-FFF5h
4		Not used			no	FFF2h-FFF3h
5	ei3	External interrupt 3			yes	FFF0h-FFF1h
6(2)	ei4(2)	External interrupt 4(2)			no(2)	FFEEh-FFEFh
7		Not used			no	FFECh-FFEDh
8	Auto-reload timer	Auto-reload timer output compare interrupt	PWMxCSR or ATCSR		no	FFEAh-FFEBh
9		Auto-reload timer overflow interrupt	ATCSR		yes(3)	FFE8h-FFE9h
10	Lite timer	Lite timer input capture interrupt	LTCSR		no	FFE6h-FFE7h
11		Lite timer RTC1 interrupt	LTCSR		yes(3)	FFE4h-FFE5h
12		Not used			no	FFE2h-FFE3h
13		Not used			no	FFE0h-FFE1h

1. This interrupt exits the MCU from “auto wake-up from halt” mode only.
2. This interrupt exits the MCU from “wait” and “active halt” modes only. Moreover, IS4[1:0] = 01 is the only safe configuration to avoid a spurious interrupt in Halt and AWUFH mode.
3. These interrupts exit the MCU from “active halt” mode only.

7.3.1 External interrupt control register 1 (EICR1)

EICR1 Reset value: 0000 0000 (00h)

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

Table 16. EICR1 register description

Bit	Name	Function
7:6	-	Reserved
5:4	IS2[1:0]	ei2 sensitivity These bits define the interrupt sensitivity for ei2 according to *Table 18*.
3:2	IS1[1:0]	ei1 sensitivity These bits define the interrupt sensitivity for ei1 according to *Table 18*.
1:0	IS0[1:0]	ei0 sensitivity These bits define the interrupt sensitivity for ei0 according to *Table 18*.

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 section External interrupt function on page 66.

7.3.2 External interrupt control register 2 (EICR2)

EICR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved				IS4[1:0]		IS3[1:0]	
-				R/W		R/W	

Table 17. EICR2 register description

Bit	Name	Function
7:4	-	Reserved
3:2	IS4[1:0]	ei4 sensitivity These bits define the interrupt sensitivity for ei1 according to *Table 18*.
1:0	IS3[1:0]	ei3 sensitivity These bits define the interrupt sensitivity for ei0 according to *Table 18*.

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 section External interrupt function on page 66.

3 IS4[1:0] = 01 is the only safe configuration to avoid a spurious interrupt in Halt and AWUFH modes.

Table 18. 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

7.4 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 Illegal opcode reset on page 99 for further details.

7.4.1 Low voltage detector (LVD)

The low voltage detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a $V_{IT-(LVD)}$ 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 in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

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

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

The LVD function is illustrated in *Figure 19*.

The high voltage threshold can be activated by option byte. See *Section 14.1 on page 131*.

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: Use of LVD with capacitive power supply: If power cuts occur in the application with this type of power supply, it is recommended to pull V_{DD} down to 0V to ensure optimum restart conditions. Refer to circuit example in Figure 67 and Note 4 on page 124.

The LVD is an optional function which can be selected by option byte (see Section 14.1 on page 131). It allows the device to be used without any external reset circuitry. If the LVD is disabled, an external circuitry must be used to ensure a proper power-on reset.

It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from reset, to ensure the application functions properly.

Caution: If an LVD reset occurs after a watchdog reset has occurred, the LVD takes priority and clears the watchdog flag.

Figure 15. Low voltage detector vs. reset

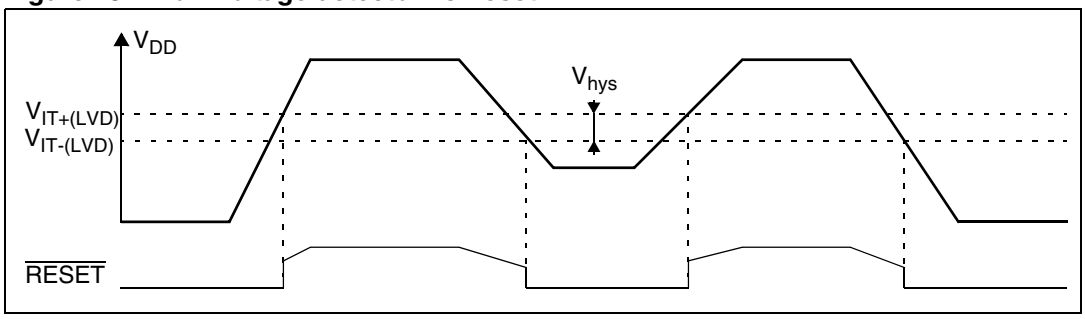
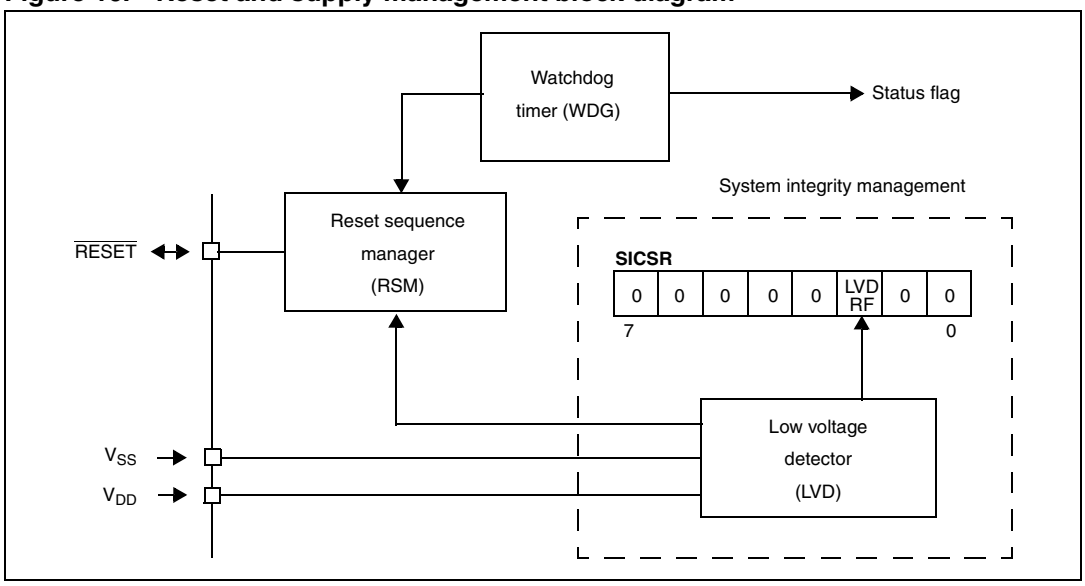


Figure 16. Reset and supply management block diagram



7.4.2 SI register description

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

SICSR Reset value: 0000 0x00 (0xh)

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

Table 19. SICSR register description

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 the required accuracy. Refer to *Section 6.1 on page 31*.
4:3	-	Reserved, must be kept cleared
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 when read. See WDGRF flag description in *Section 10.1* for more details. When the LVD is disabled by option byte, the LVDRF bit value is undefined. *Note: If the selected clock source is one of the two internal ones, and if V_{DD} remains below the selected LVD threshold during less than T_{AWU} (33µs typ.), the LVDRF flag cannot be set even if the device is reset by the LVD.* *If the selected clock source is the external clock (CLKIN), the flag is never set if the reset occurs during halt mode. In Run mode the flag is set only if f_{CLKIN} is greater than 10 MHz.*
1:0	-	Reserved, must be kept cleared

Application notes

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

Table 20. System integrity register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
003Ah	SICSR Reset value	0	CR1 0	CR0 0	0	0	LVDRF x	0	0

8 Power saving modes

8.1 Introduction

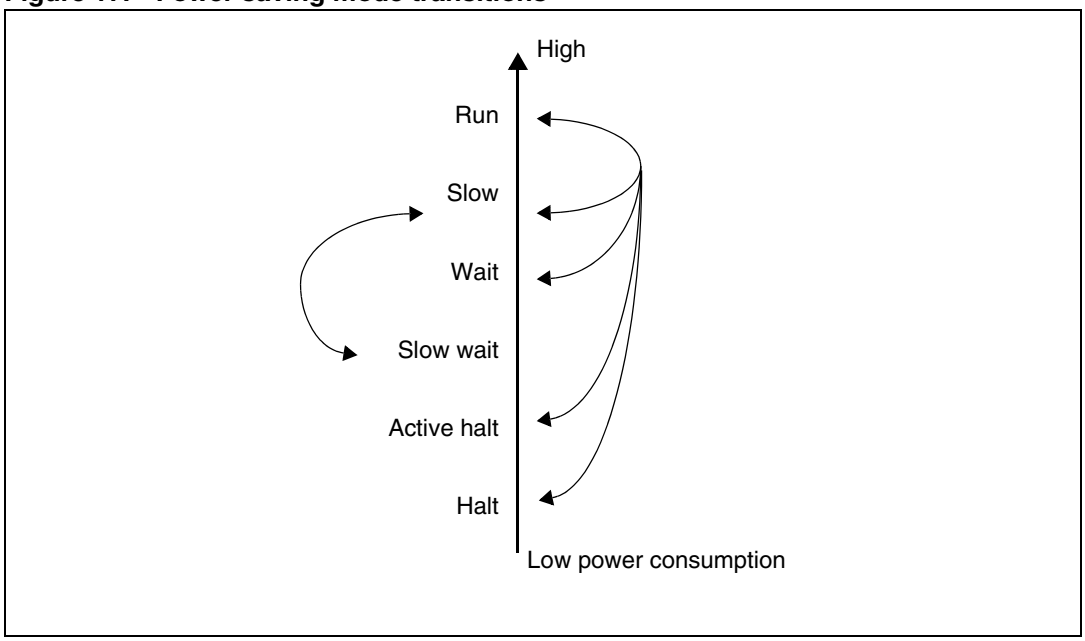
To give a large measure of flexibility to the application in terms of power consumption, five main power saving modes are implemented in the ST7 (see *Figure 21*):

- Slow
- Wait (and slow wait)
- Active halt
- Auto wake-up 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 (f_{OSC}).

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

Figure 17. Power saving mode transitions



8.2 Slow mode

This mode has two targets:

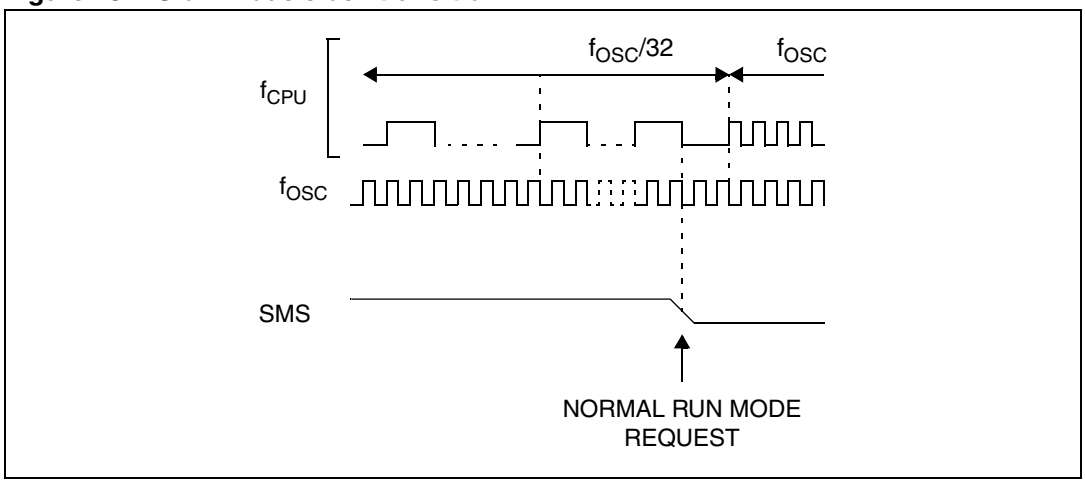
- To reduce power consumption by decreasing the internal clock in the device
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

Slow mode is controlled by 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 18. Slow mode clock transition



8.3 Wait mode

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

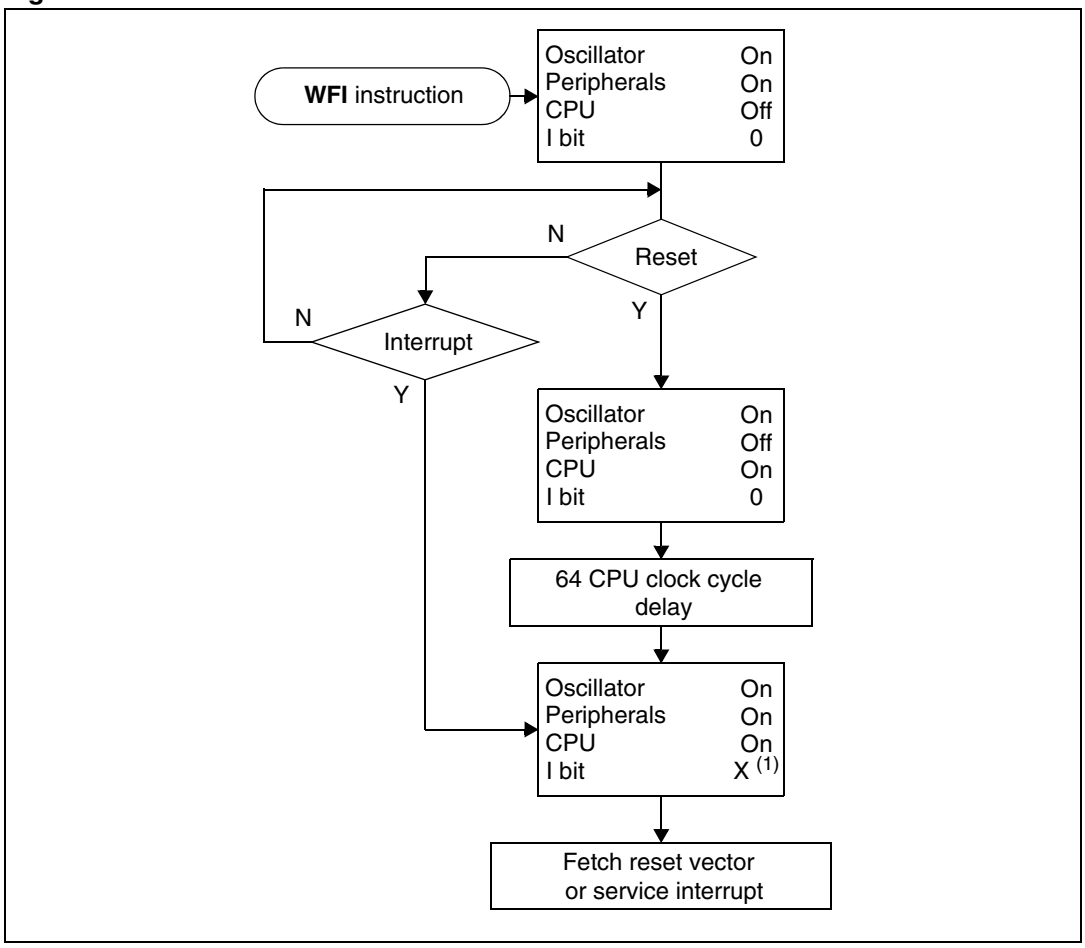
This power saving mode is selected by calling the WFI instruction.

All peripherals remain active. During wait mode, the I 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 the program counter branches to the starting address of the interrupt or reset service routine.

The MCU will remain in wait mode until a reset or an Interrupt occurs, causing it to wake up.

Refer to *Figure 24 on page 54*.

Figure 19. 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.

8.4 Active halt and halt modes

Active halt and halt modes are the two lowest power consumption modes of the MCU. They are both entered by executing the HALT instruction. The decision to enter either in active halt or halt mode is given by the LTCSR/ATCSR register status as shown in the following table:

Table 21. LTC/ATC low power mode selection

LTCSR TBIE bit	ATCSR OVFIE bit	ATCSRCK1 bit	ATCSRCK0 bit	Meaning
0	x	x	0	Active halt mode disabled
0	0	x	x	
0	1	1	1	
1	x	x	x	Active halt mode enabled
x	1	0	1	

8.4.1 Active halt mode

Active halt mode is the lowest power consumption mode of the MCU with a real-time clock available. It is entered by executing the HALT instruction when active halt mode is enabled.

The MCU can exit active halt mode on reception of a lite timer/auto-reload timer interrupt or a reset.

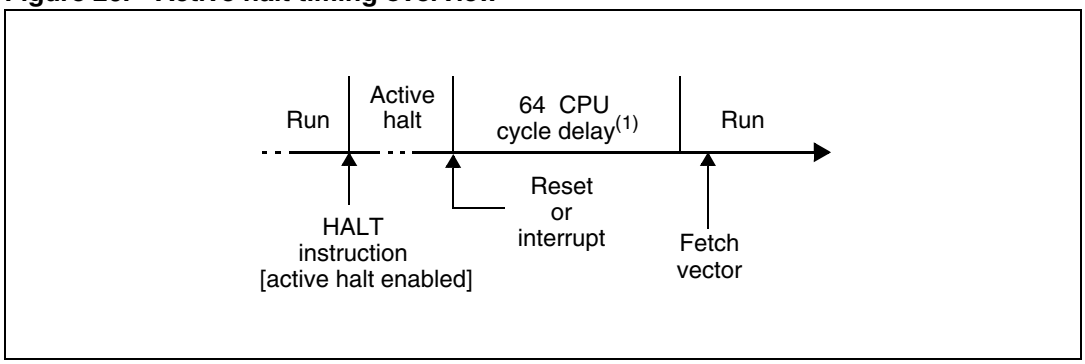
- When exiting active halt mode by means of a reset, a 64 CPU cycle delay occurs. After the start up delay, the CPU resumes operation by fetching the reset vector which woke it up (see *Figure 27*).
- When exiting active halt mode by means of an interrupt, the CPU immediately resumes operation by servicing the interrupt vector which woke it up (see *Figure 27*).

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.

In active halt mode, only the main oscillator and the selected timer counter (lite timer/auto-reload timer) are running to keep a wake-up time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

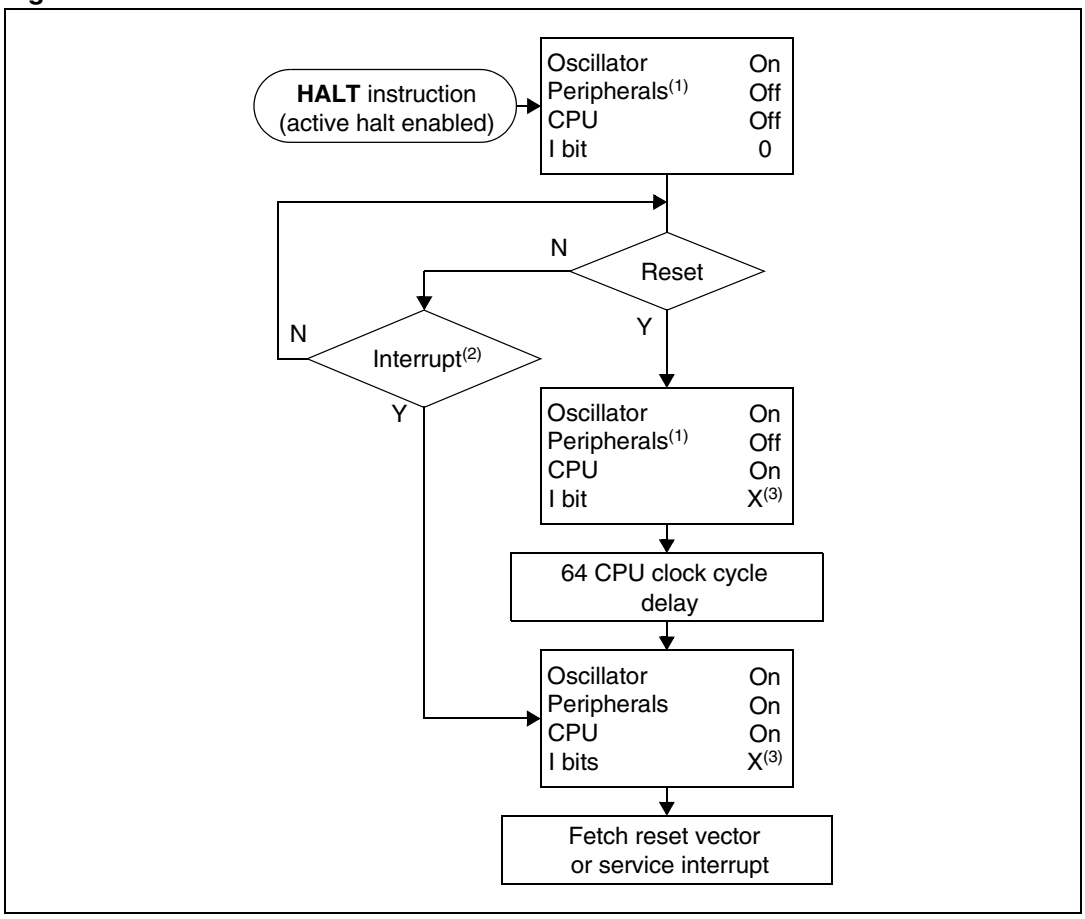
Caution: As soon as active halt is enabled, executing a HALT instruction while the watchdog is active does not generate a reset if the WDGHALT bit is reset. This means that the device cannot spend more than a defined delay in this power saving mode.

Figure 20. Active halt timing overview



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

Figure 21. Active halt mode flowchart



1. Peripherals clocked with an external clock source can still be active.
2. Only the lite timer RTC and auto-reload timer interrupts can exit the MCU from active halt mode.
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.

8.4.2 Halt mode

The halt mode is the lowest power consumption mode of the MCU. It is entered by executing the HALT instruction when active halt mode is disabled.

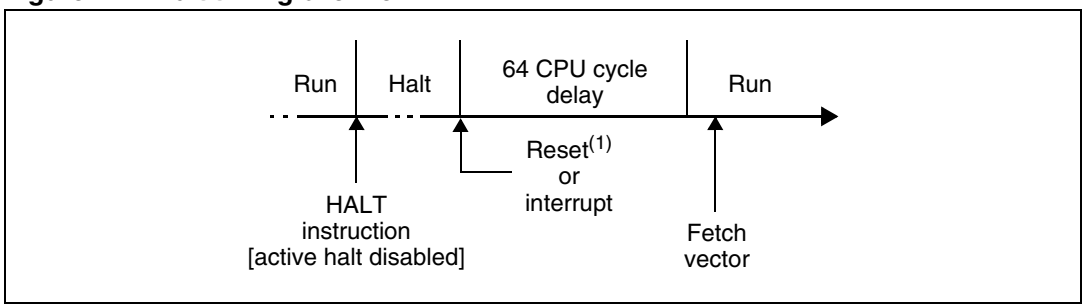
The MCU can exit halt mode on reception of either a specific interrupt (see *Table 15: Interrupt mapping on page 46*) or a reset. When exiting halt mode by means of a reset or an interrupt, the main oscillator is immediately turned on and the 64 CPU cycle delay is used to stabilize it. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see *Figure 30*).

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

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

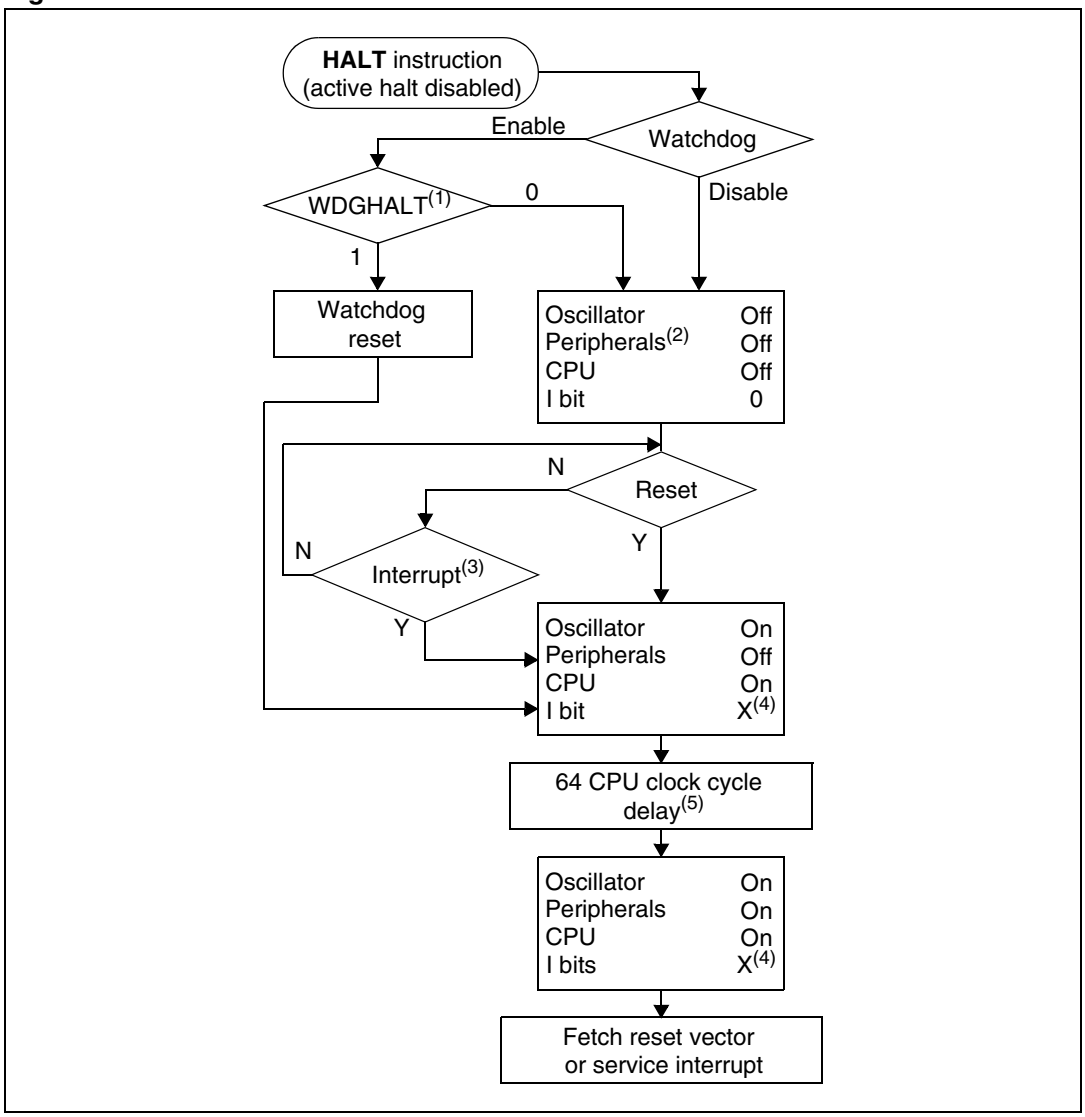
The compatibility of watchdog operation with halt mode is configured by the “WDGHALT” option bit of the option byte. The HALT instruction when executed while the watchdog system is enabled, can generate a watchdog reset (see *Section 14.1 on page 131* for more details).

Figure 22. Halt timing overview



1. A reset pulse of at least 42µs must be applied when exiting from halt mode.

Figure 23. Halt mode flowchart



1. WDGHALT is an option bit. See option bytes in *Section 14.1.1: Flash configuration* 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 46* 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. The CPU clock must be switched to 1MHz (RC/8) or AWU RC before entering halt mode.

Halt mode recommendations

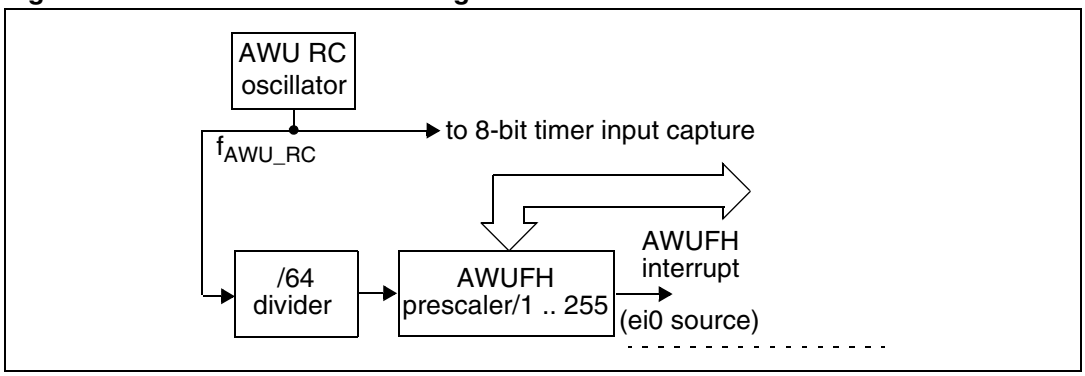
- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- When using an external interrupt to wake up the microcontroller, re-initialize the corresponding I/O as “input pull-up with interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly 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 ROM with the value 0x8E.
- As the HALT instruction clears the I bit in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

8.5 Auto wake-up from halt mode

Auto Wake-up from Halt (AWUFH) mode is similar to halt mode with the addition of a specific internal RC oscillator for wake-up (auto wake-up from halt oscillator) which replaces the main clock which was active before entering halt mode. 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 24. 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 following actions are performed:

- The AWUF flag is set by hardware,
- an interrupt wakes up the MCU from halt mode,
- The main oscillator is immediately turned on and the 64 CPU cycle delay is used to stabilize it.

After this start-up 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 8-bit lite 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 8.4: Active halt and halt modes*).
- 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 causing all internal processing to be stopped, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which get 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 25. AWUF halt timing diagram

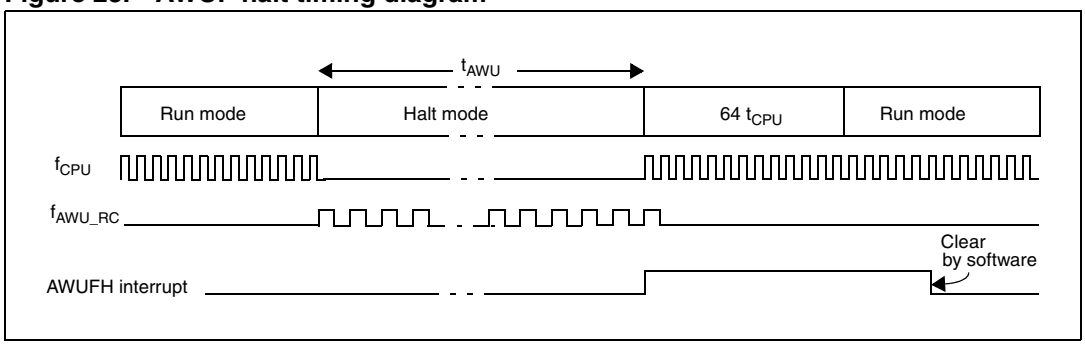
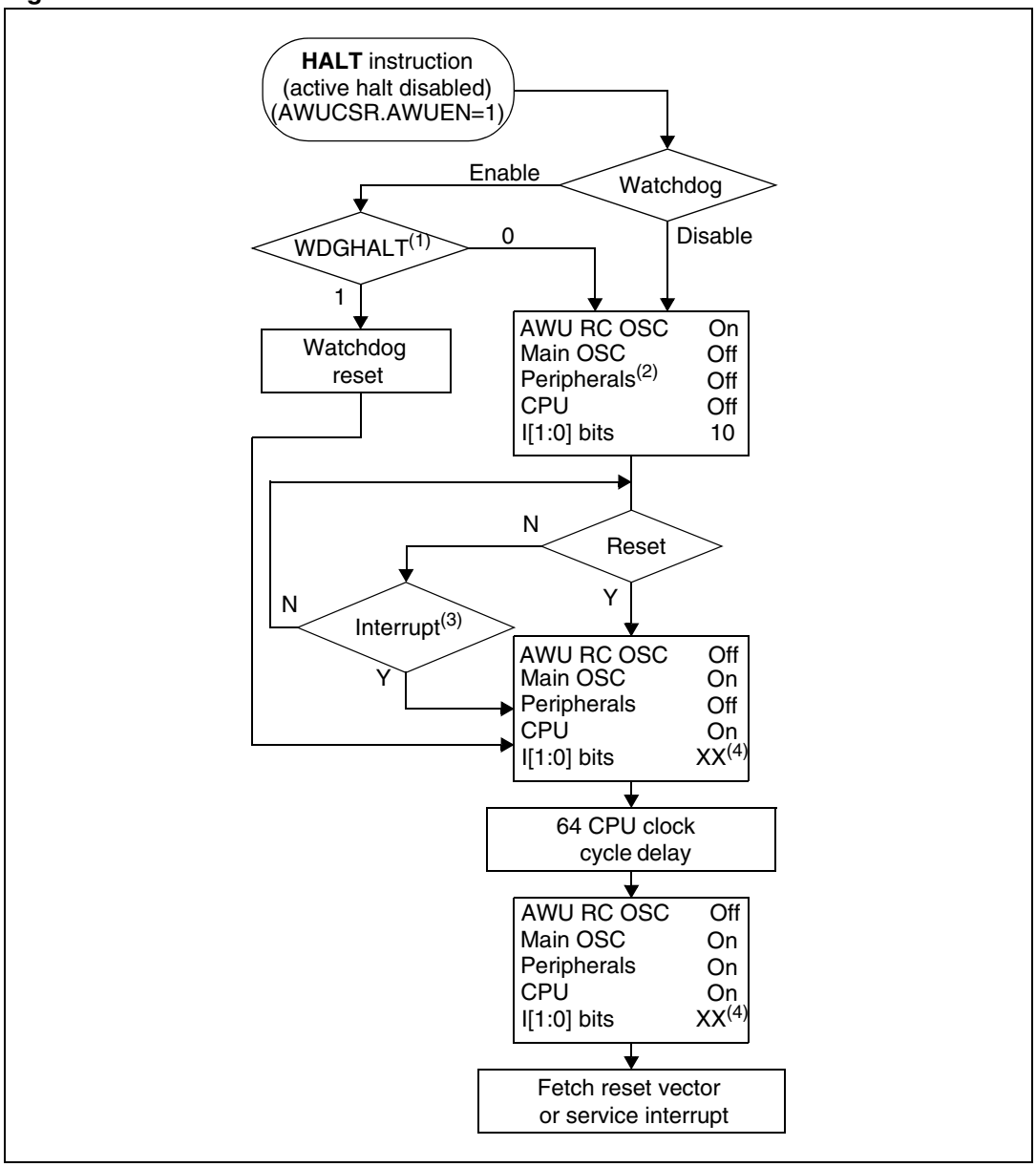


Figure 26. AWUFH mode flowchart



1. WDGHALT is an option bit. See option bytes in *Section 14.1.1: Flash configuration* 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 46* 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.

8.5.1 Register description

AWUFH control/status register (AWUCSR)

AWUCSR Reset value: 0000 0000 (00h)

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

Table 22. AWUCSR register description

Bit	Name	Function
7:3	-	Reserved
2	AWUF	Auto wake-up 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 wake-up measurement This bit enables the AWU RC oscillator and connects its output to the input capture of the 8-bit lite timer. This allows the timer to be used 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 wake-up from halt enabled This bit enables the auto wake-up 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 wake-up from halt) mode disabled 1: AWUFH (auto wake-up from halt) mode enabled *Note: Whatever the clock source, this bit should be set to enable the AWUFH mode once the HALT instruction has been executed.*

AWUFH prescaler register (AWUPR)

AWUPR — Reset value: 1111 1111 (FFh)

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

Table 23. AWUPR register description

Bit	Name	Function
7:0	AWUPR[7:0]	Auto wake-up prescaler These 8 bits define the AWUPR division factor (as explained below). 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 24. AWUPR division factors

AWUPR[7:0]	Division 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 32 on page 61*) 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 25. AWU register map and reset values

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

9 I/O ports

9.1 Introduction

The I/O port offers different functional modes:

- transfer of data through digital inputs and outputs

and for specific pins:

- external interrupt generation
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to six pins. Each pin (except PA3/$\overline{\text{RESET}}$) can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 Functional description

Each port has two main registers:

- Data register (DR)
- Data direction register (DDR)

and one optional register:

- Option register (OR)

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

The following description takes into account the OR register (for specific ports which do not provide this register refer to *Section 9.6: I/O port implementation*). The generic I/O block diagram is shown in *Figure 35*.

9.2.1 Input modes

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

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

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

Note: 1 Writing the DR register modifies the latch value but does not affect the pin status.

2 PA3 cannot be configured as input.

External interrupt function

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

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

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) has to be selected before changing the OR register bit and configuring the appropriate sensitivity again.

Caution: In case 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:
   a) set the interrupt mask with the SIM instruction (in cases where a pin level change could occur)
   b) select rising edge
   c) enable the external interrupt through the OR register
   d) select the desired sensitivity if different from rising edge
   e) reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur)
2. To disable an external interrupt:
   a) set the interrupt mask with the SIM instruction SIM (in cases where a pin level change could occur)
   b) select falling edge
   c) disable the external interrupt through the OR register
   d) select rising edge

9.2.2 Output modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

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

Table 26. I/O output mode selection

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

Note: When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

9.2.3 Alternate functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the standard I/O programming under the following conditions:

- When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).
- When the signal is going to an on-chip peripheral, the I/O pin must be configured in floating input mode. In this case, the pin state is also digitally readable by addressing the DR register.

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

When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.

Figure 27. I/O port general block diagram

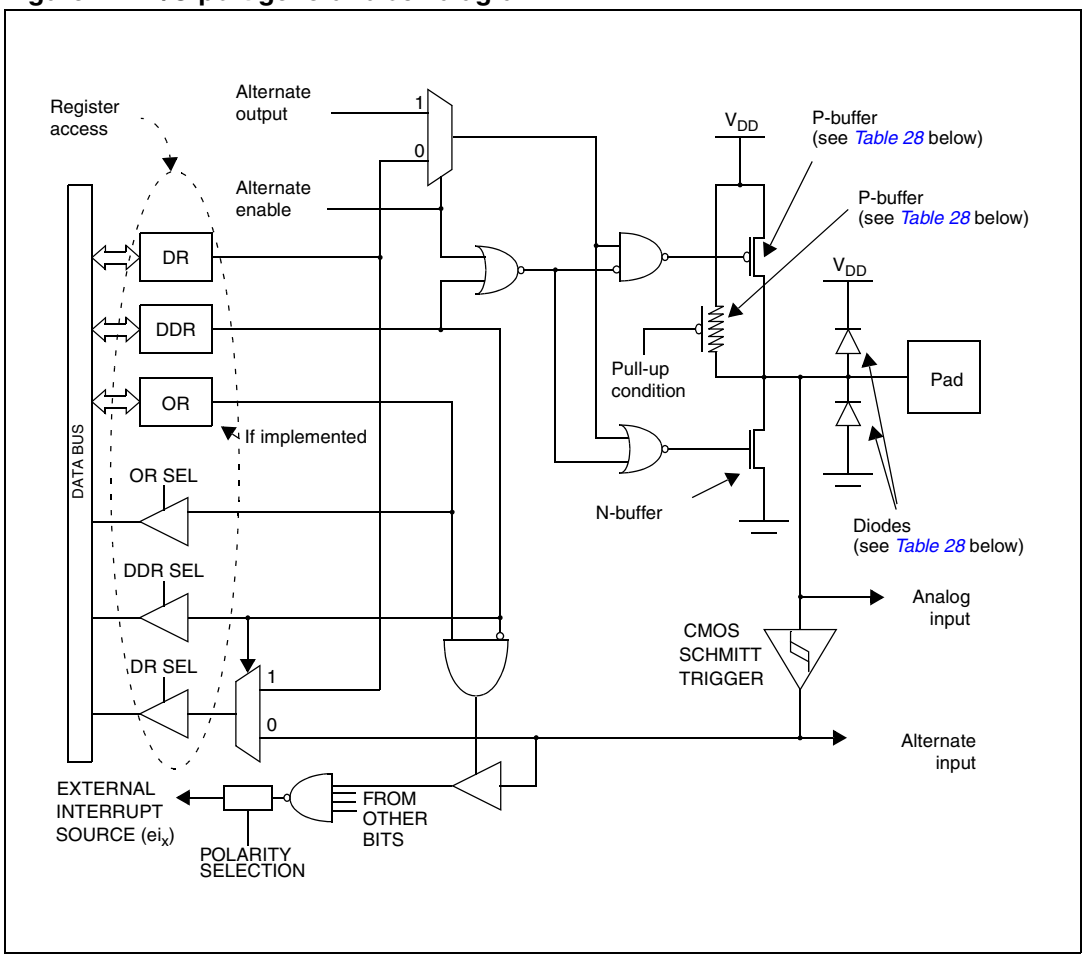
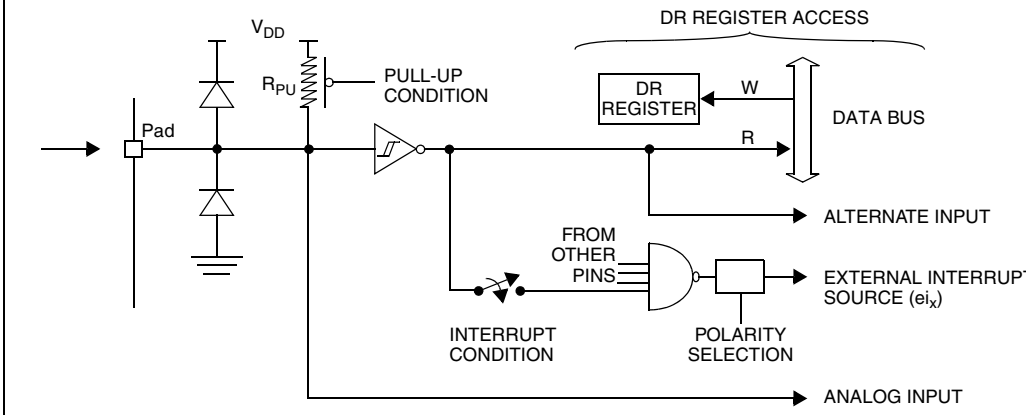
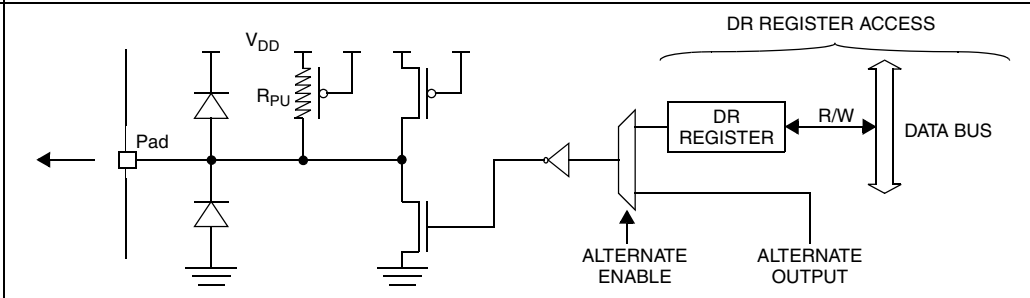
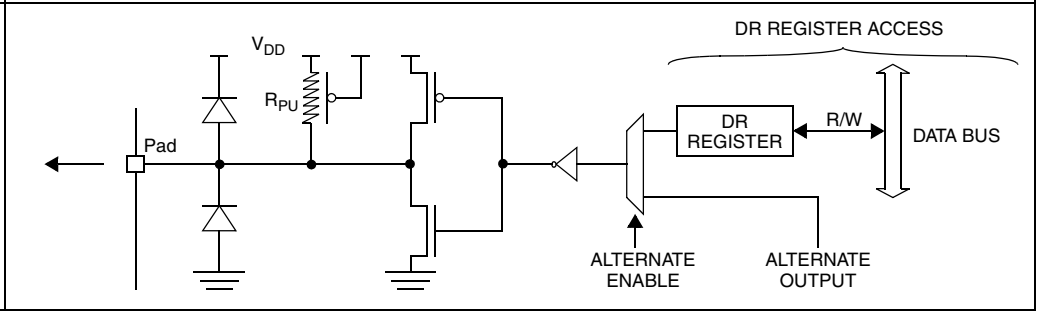


Table 27. I/O port mode options(1)

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. Off = implemented not activated
   On = implemented and activated

Table 28. I/O port 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 will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

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

Analog alternate function

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

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

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

9.3 Unused I/O pins

Unused I/O pins must be connected to fixed voltage levels. Refer to *Section 12.8*.

9.4 Low power modes

Table 29. Effect of ow 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.

9.5 Interrupts

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

Table 30. I/O port interrupt control/wake-up capability

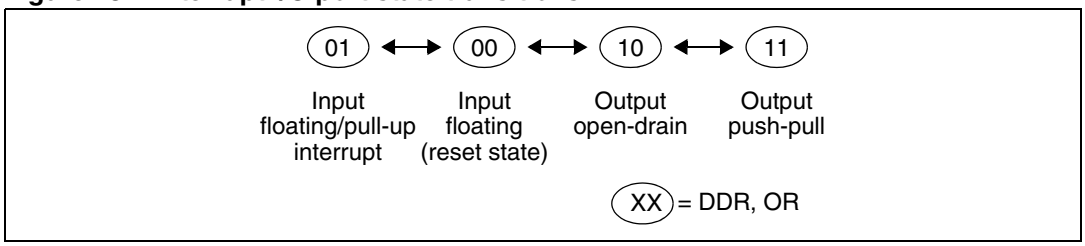
Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from halt
External interrupt on selected external event	-	DDRx, ORx	Yes	Yes

9.6 I/O port implementation

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port such as ADC Input or true open drain.

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in *Figure 36*. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 28. Interrupt I/O port state transitions



The I/O port register configurations are summarized in the following table:

Table 31. Port configuration

Port	Pin name	Input (DDR=0)		Output (DDR=1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA0:2, PA4:5	Floating	Pull-up interrupt	Open drain	Push-pull
	PA3	-	-	Open drain	Push-pull

Note: After reset, to configure PA3 as a general purpose output, the application has to program the MUXCR0 and MUXCR1 registers. See *Section 6.4 on page 42*.

Table 32. 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 0	0	0	0	0	0	0	LSB 0
0001h	PADDR Reset value	MSB 0	0	0	0	1	0	0	LSB 0
0002h	PAOR Reset value	MSB 0	0	0	0	0	0	1	LSB 0

10 On-chip peripherals

10.1 Lite timer

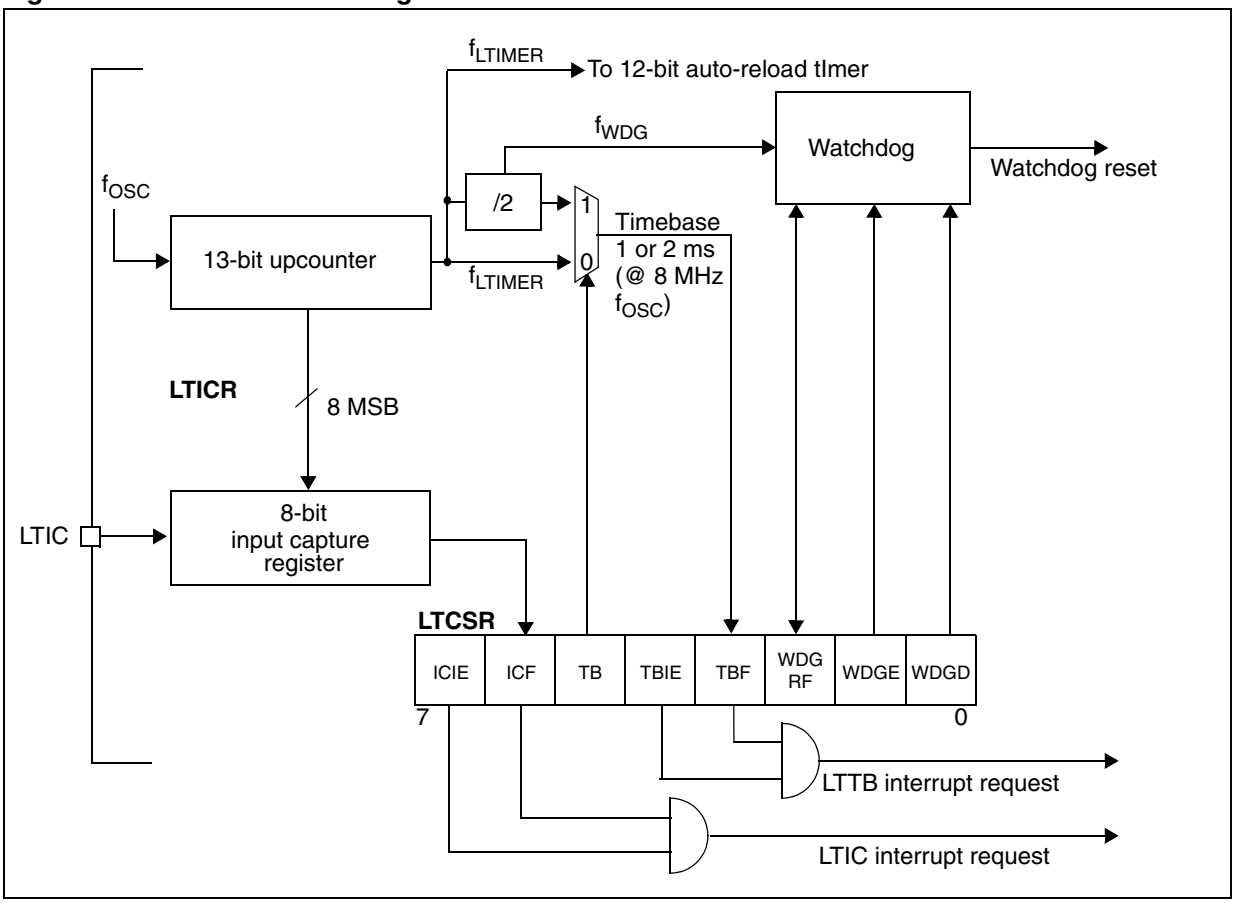
10.1.1 Introduction

The lite timer can be used for general-purpose timing functions. It is based on a free-running 13-bit upcounter with two software-selectable timebase periods, an 8-bit input capture register and watchdog function.

10.1.2 Main features

- Real-time clock
  - 13-bit upcounter
  - 1 ms or 2 ms timebase period (@ 8 MHz f_{OSC})
  - Maskable timebase interrupt
- Input capture
  - 8-bit input capture register (LTICR)
  - Maskable interrupt with wake-up from halt mode capability
- Watchdog
  - Enabled by hardware or software (configurable by option byte)
  - Optional reset on HALT instruction (configurable by option byte)
  - Automatically resets the device unless disable bit is refreshed
  - Software reset (forced watchdog reset)
  - Watchdog reset status flag

Figure 29. Lite timer block diagram



10.1.3 Functional description

The value of the 13-bit counter cannot be read or written by software. After an MCU reset, it starts incrementing from 0 at a frequency of f_{OSC}. A counter overflow event occurs when the counter rolls over from 1F39h to 00h. If f_{OSC} = 8 MHz, then the time period between two counter overflow events is 1ms. This period can be doubled by setting the TB bit in the LTCSR register.

When the timer overflows, the TBF bit is set by hardware and an interrupt request is generated if the TBIE is set. The TBF bit is cleared by software reading the LTCSR register.

Watchdog

The watchdog is enabled using the WDGE bit. The normal watchdog timeout is 2ms (@ f_{OSC} = 8 MHz), after which it then generates a reset.

To prevent this watchdog reset from occurring, software must set the WDGD bit. The WDGD bit is cleared by hardware after t_{WDG}. This means that software must write to the WDGD bit at regular intervals to prevent a watchdog reset from occurring. Refer to *Figure 38*.

If the watchdog is not enabled immediately after reset, the first watchdog timeout will be shorter than 2ms, because this period is counted starting from reset. Moreover, if a 2ms period has already elapsed after the last MCU reset, the watchdog reset will take place as soon as the WDGE bit is set. For these reasons, it is recommended to enable the watchdog immediately after reset or else to set the WDGD bit before the WGDE bit so a watchdog reset will not occur for at least 2ms.

Note: Software can use the timebase feature to set the WDGD bit at 1 or 2 ms intervals.

A watchdog reset can be forced at any time by setting the WDGRF bit. To generate a forced watchdog reset, first watchdog has to be activated by setting the WDGE bit and then the WDGRF bit has to be set.

The WDGRF bit also acts as a flag, indicating that the watchdog was the source of the reset. It is automatically cleared after it has been read.

Caution: When the WDGRF bit is set, software must clear it, otherwise the next time the watchdog is enabled (by hardware or software), the microcontroller will be immediately reset.

Hardware watchdog option

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

Refer to the option byte description in *Chapter 14: Device configuration and ordering information*.

Using halt mode with the watchdog (option)

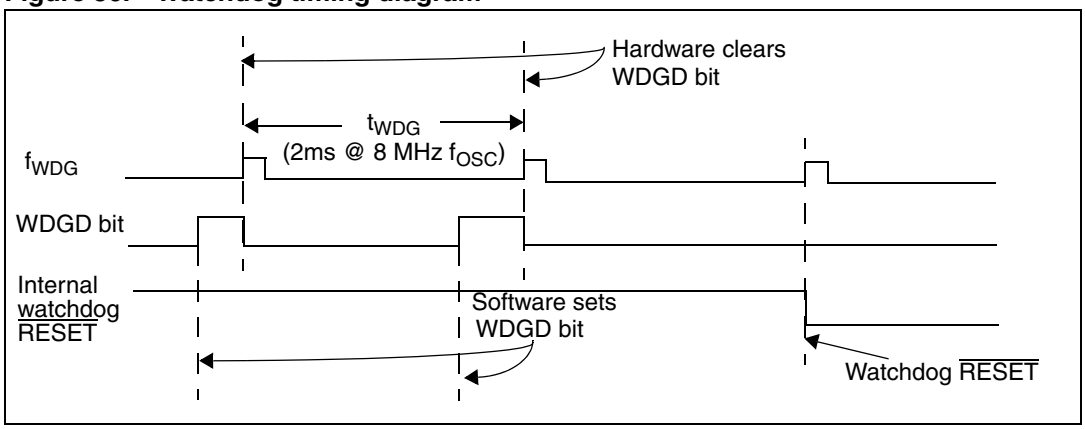
If the watchdog reset on halt option is not selected by option byte, the halt mode can be used when the watchdog is enabled.

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

If an external interrupt is received, the WDG restarts counting after 64 CPU clocks. If a reset is generated, the watchdog is disabled (reset state).

If halt mode with watchdog is enabled by 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.

Figure 30. Watchdog timing diagram



Input capture

The 8-bit input capture register is used to latch the free-running upcounter after a rising or falling edge is detected on the LTIC pin. When an input capture occurs, the ICF bit is set and the LTICR register contains the MSB of the free-running upcounter. 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.

10.1.4 Low power modes

Table 33. Effect of low power modes on lite timer

Mode	Description
Wait	No effect on lite timer
Active halt	
Halt	Lite timer stops counting

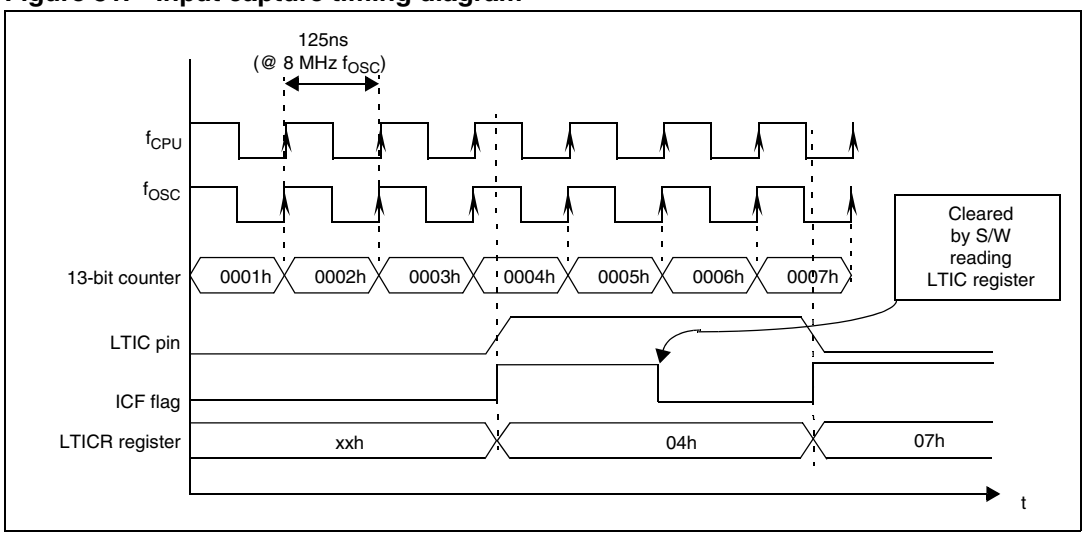
10.1.5 Interrupts

Table 34. Lite timer interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from wait	Exit from halt	Exit from active halt
Timebase event	TBF	TBIE	Yes	No	Yes
IC event	ICF	ICIE			No

Note: The TBF and ICF interrupt events are connected to separate interrupt vectors (see Chapter 7: Interrupts). They generate an interrupt if the enable bit is set in the LTCSR register and the interrupt mask in the CC register is reset (RIM instruction).

Figure 31. Input capture timing diagram



10.1.6 Register description

Lite timer control/status register (LTCSR)

LTCSR Reset value: 0000 0x00 (0xh)

7	6	5	4	3	2	1	0
ICIE	ICF	TB	TBIE	TBF	WDGRF	WDGE	WDGD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35. LTCSR register description

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	TBIE	Timebase interrupt enable This bit is set and cleared by software. 0: Timebase (TB) interrupt disabled 1: Timebase (TB) interrupt enabled
3	TBF	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	WDGRF	Force reset/reset status flag This bit is used in two ways: It is set by software to force a watchdog reset. It is set by hardware when a watchdog reset occurs and cleared by hardware or by software. It is cleared by hardware only when an LVD reset occurs. It can be cleared by software after a read access to the LTCSR register. 0: No watchdog reset occurred. 1: Force a watchdog reset (write), or, a watchdog reset occurred (read).

Table 35. LTCSR register description (continued)

Bit	Name	Function
1	WDGE	Watchdog enable This bit is set and cleared by software. 0: Watchdog disabled 1: Watchdog enabled
0	WDGD	Watchdog reset delay This bit is set by software. It is cleared by hardware at the end of each t_{WDG} period. 0: Watchdog reset not delayed 1: Watchdog reset delayed

Lite timer input capture register (LTICR)

LTICR Reset value: 0000 0000 (00h)

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

Table 36. LTICR register description

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 upcounter will be captured when a rising or falling edge occurs on the LTIC pin.

Table 37. Lite timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0B	LTCSR Reset value	ICIE 0	ICF 0	TB 0	TBIE 0	TBF 0	WDGRF x	WDGE 0	WDGD 0
0C	LTICR Reset value	ICR7 0	ICR6 0	ICR5 0	ICR4 0	ICR3 0	ICR2 0	ICR1 0	ICR0 0

10.2 12-bit autoreload timer

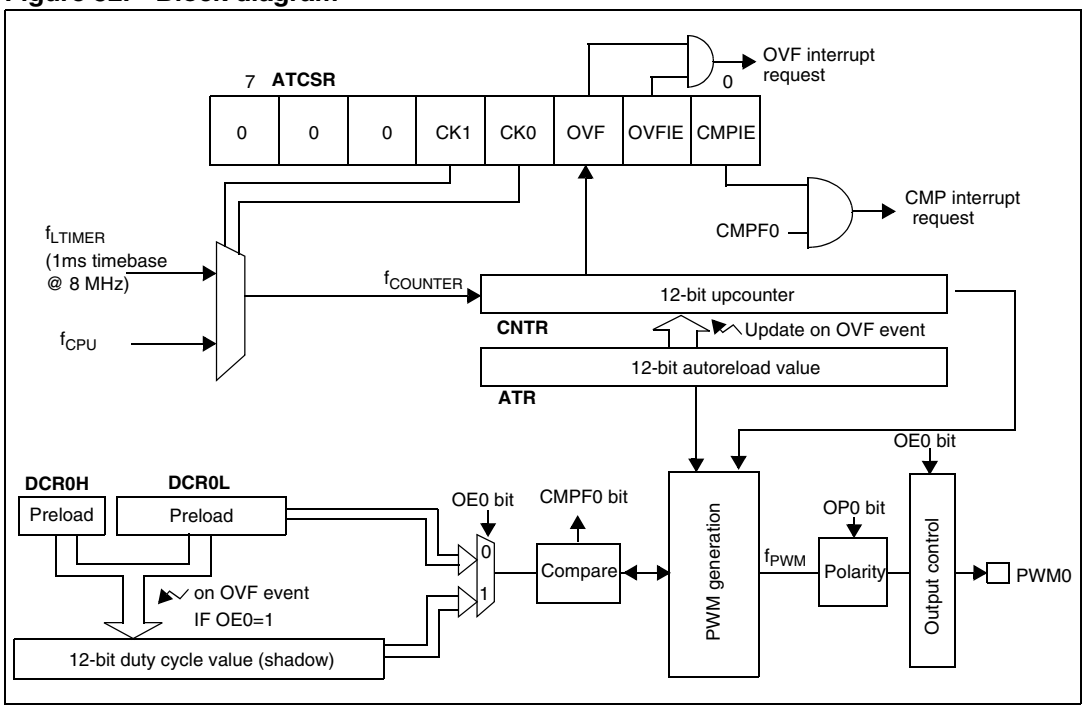
10.2.1 Introduction

The 12-bit autoreload timer can be used for general-purpose timing functions. It is based on a free-running 12-bit upcounter with a PWM output channel.

10.2.2 Main features

- 12-bit upcounter with 12-bit autoreload register (ATR)
- Maskable overflow interrupt
- PWM signal generator
- Frequency range 2 kHz to 4 MHz (@ 8 MHz f_{CPU})
  - Programmable duty-cycle
  - Polarity control
  - Maskable compare interrupt
- Output compare function

Figure 32. Block diagram



10.2.3 Functional description

PWM mode

This mode allows a pulse width modulated signals to be generated on the PWM0 output pin with minimum core processing overhead. The PWM0 output signal can be enabled or disabled using the OE0 bit in the PWMCR register. When this bit is set the PWM I/O pin is configured as output push-pull alternate function.

Note: CMPF0 is available in PWM mode (see PWM0CSR register description on page 85).

PWM frequency and duty cycle

The PWM signal frequency (f_{PWM}) is controlled by the counter period and the ATR register value.

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

Following the above formula, if f_{CPU} is 8 MHz, the maximum value of f_{PWM} is 4 MHz (ATR register value = 4094), and the minimum value is 2 kHz (ATR register value = 0).

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

At reset, the counter starts counting from 0.

Software must write the duty cycle value in the DCR0H and DCR0L preload registers. The DCR0H register must be written first. See caution below.

When an upcounter overflow occurs (OVF event), the ATR value is loaded in the upcounter, the preloaded Duty cycle value is transferred to the Duty Cycle register and the PWM0 signal is set to a high level. When the upcounter matches the DCRx value the PWM0 signals is set to a low level. To obtain a signal on the PWM0 pin, the contents of the DCR0 register must be greater than the contents of the ATR register.

The polarity bit can be used to invert the output signal.

The maximum available resolution for the PWM0 duty cycle is:

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

Note: To get the maximum resolution (1/4096), the ATR register must be 0. With this maximum resolution and assuming that DCR = ATR, a 0% or 100% duty cycle can be obtained by changing the polarity.

Caution: As soon as the DCR0H is written, the compare function is disabled and will start only when the DCR0L value is written. If the DCR0H write occurs just before the compare event, the signal on the PWM output may not be set to a low level. In this case, the DCRx register should be updated just after an OVF event. If the DCR and ATR values are close, then the DCRx register should be updated just before an OVF event, in order not to miss a compare event and to have the right signal applied on the PWM output.

Figure 33. PWM function

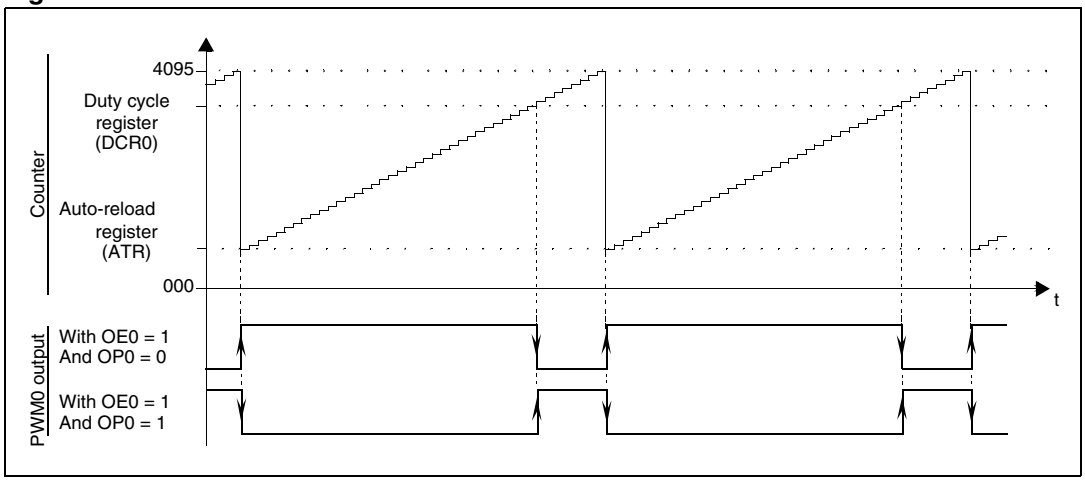
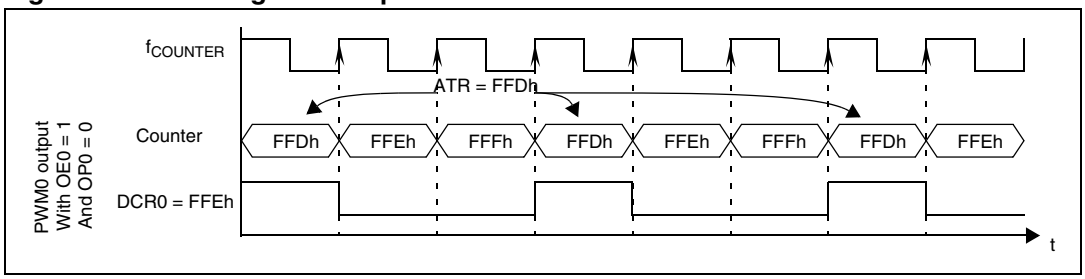


Figure 34. PWM signal example



Output compare mode

To use this function, the OE bit must be 0, otherwise the compare is done with the shadow register instead of the DCRx register. Software must then write a 12-bit value in the DCR0H and DCR0L registers. This value will be loaded immediately (without waiting for an OVF event).

The DCR0H must be written first, the output compare function starts only when the DCR0L value is written.

When the 12-bit upcounter (CNTR) reaches the value stored in the DCR0H and DCR0L registers, the CMPF0 bit in the PWM0CSR register is set and an interrupt request is generated if the CMPIE bit is set.

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

Caution: At each OVF event, the DCRx value is written in a shadow register, even if the DCR0L value has not yet been written (in this case, the shadow register will contain the new DCR0H value and the old DCR0L value), then:
- If OE = 1 (PWM mode): the compare is done between the timer counter and the shadow register (and not DCRx)
- if OE = 0 (OCMP mode): the compare is done between the timer counter and DCRx. There is no PWM signal.

The compare between DCRx or the shadow register and the timer counter is locked until DCR0L is written.

10.2.4 Low power modes

Table 38. Effect of low power modes onthe auto-reload timer

Mode	Description
Slow	The input frequency is divided by 32
Wait	No effect on the auto-reload timer
Active halt	Auto-reload timer halted except if CK0 = 1, CK1 = 0 and OVFIE = 1
Halt	Auto-reload timer halted

10.2.5 Interrupts

Table 39. Auto-reload timer interrupt control/wake-up capability

Interrupt event[1]	Event flag	Enable control bit	Exit from wait	Exit from halt	Exit from active halt
Overflow event	OVF	OVFIE	Yes	No	Yes[2]
CMP event	CMPFx	CMPIE			No

1. The interrupt events are connected to separate interrupt vectors (see *Chapter 7: 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).
2. Only if CK0 = 1 and CK1 = 0.

10.2.6 Register description

Timer control status register (ATCSR)

ATCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved			CK[1:0]		OVF	OVFIE	CMPIE
-			R/W		R/W	R/W	R/W

Table 40. ATCSR register description

Bit	Name	Function
7:5	-	Reserved, must be kept cleared.
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. 00: Off 01: f_{LTIMER} (1ms timebase @ 8 MHz) 10: f_{CPU} 11: Reserved
2	OVF	Overflow flag This bit is set by hardware and cleared by software by reading the ATCSR register. It indicates the transition of the counter from FFFh to ATR value. 0: No counter overflow occurred 1: Counter overflow occurred **Caution:** When set, the OVF bit stays high for 1 $f_{COUNTER}$ cycle (up to 1ms depending on the clock selection) after it has been cleared by software.

Table 40. ATCSR register description

Bit	Name	Function
1	OVFIE	Overflow interrupt enable This bit is read/write by software and cleared by hardware after a reset. 0: OVF interrupt disabled 1: OVF interrupt enabled
0	CMPIE	Compare interrupt enable This bit is read/write by software and cleared by hardware after a reset. It is used to mask the interrupt generation when CMPF bit is set. 0: CMPF interrupt disabled 1: CMPF interrupt enabled

Counter register high (CNTRH)

CNTRH Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved				CNTR11	CNTR10	CNTR9	CNTR8
-				RO	RO	RO	RO

Counter register low (CNTRL)

CNTRL Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
CNTR7	CNTR6	CNTR5	CNTR4	CNTR3	CNTR2	CNTR1	CNTR0
RO	RO	RO	RO	RO	RO	RO	RO

Table 41. CNTR high and low counter descriptions

Bits	Name	Function
15:12	-	Reserved, must be kept cleared.
11:0	CNTR[11:0]	Counter value This 12-bit register is read by software and cleared by hardware after a reset. The counter is incremented 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 fTIMER = fCPU, 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 ATR register.

Auto reload register high (ATRH)

ATRH Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved				ATR11	ATR10	ATR9	ATR8
-				R/W	R/W	R/W	R/W

Auto reload register low (ATRL)

ATRL Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ATR7	ATR6	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42. ATR high and low register descriptions

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

PWM0 duty cycle register high (DCR0H)

DCR0H Reset value: 0000 0000 (00h)

15	14	13	12	11	10	9	8
Reserved				DCR11	DCR10	DCR9	DCR8
-				R/W	R/W	R/W	R/W

PWM0 duty cycle register low (DCR0L)

DCR0L Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
DCR7	DCR6	DCR5	DCR4	DCR3	DCR2	DCR1	DCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43. DCR0 high and low register descriptions

Bits	Name	Function
15:12	-	Reserved, must be kept cleared
11:0	DCR[11:0]	PWMx duty cycle value This 12-bit value is written by software. The high register must be written first. In PWM mode (OE0 = 1 in the PWMCR register) the DCR[11:0] bits define the duty cycle of the PWM0 output signal (see *Figure 41*). In output compare mode, (OE0 = 0 in the PWMCR register) they define the value to be compared with the 12-bit upcounter value.

PWM0 control/status register (PWM0CSR)

PWM0CSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved						OP0	CMPF0
-						R/W	R/W

Table 44. PWM0CSR register description

Bit	Name	Function
7:2	-	Reserved, must be kept cleared.
1	OP0	PWM0 output polarity This bit is read/write by software and cleared by hardware after a reset. This bit selects the polarity of the PWM0 signal. 0: The PWM0 signal is not inverted. 1: The PWM0 signal is inverted.
0	CMPF0	PWM0 compare flag This bit is set by hardware and cleared by software by reading the PWM0CSR register. It indicates that the upcounter value matches the DCR0 register value. 0: Upcounter value does not match DCR value 1: Upcounter value matches DCR value

PWM output control register (PWMCR)

PWMCR Reset value: 0000 0000 (00h)

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

Table 45. PWMCR register description

Bit	Name	Function
7:1	-	Reserved, must be kept cleared.
0	OE0	PWM0 output enable This bit is set and cleared by software. 0: PWM0 output alternate function disabled (I/O pin free for general purpose I/O) 1: PWM0 output enabled

Table 46. Register map and reset values

Address (hex.)	Register label	7	6	5	4	3	2	1	0
0D	ATCSR Reset value	0	0	0	CK1 0	CK0 0	OVF 0	OVFIE 0	CMPIE 0
0E	CNTRH Reset value	0	0	0	0	CN11 0	CN10 0	CN9 0	CN8 0
0F	CNTRL Reset value	CN7 0	CN6 0	CN5 0	CN4 0	CN3 0	CN2 0	CN1 0	CN0 0
10	ATRH Reset value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
11	ATRL Reset value	ATR7 0	ATR6 0	ATR5 0	ATR4 0	ATR3 0	ATR2 0	ATR1 0	ATR0 0
12	PWMCR Reset value	0	0	0	0	0	0	0	OE0 0
13	PWM0CSR Reset value	0	0	0	0	0	0	OP 0	CMPF0 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

10.3 10-bit A/D converter (ADC)

10.3.1 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 5 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 5 different sources.

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

10.3.2 Main features

- 10-bit conversion
- Up to 5 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 43*.

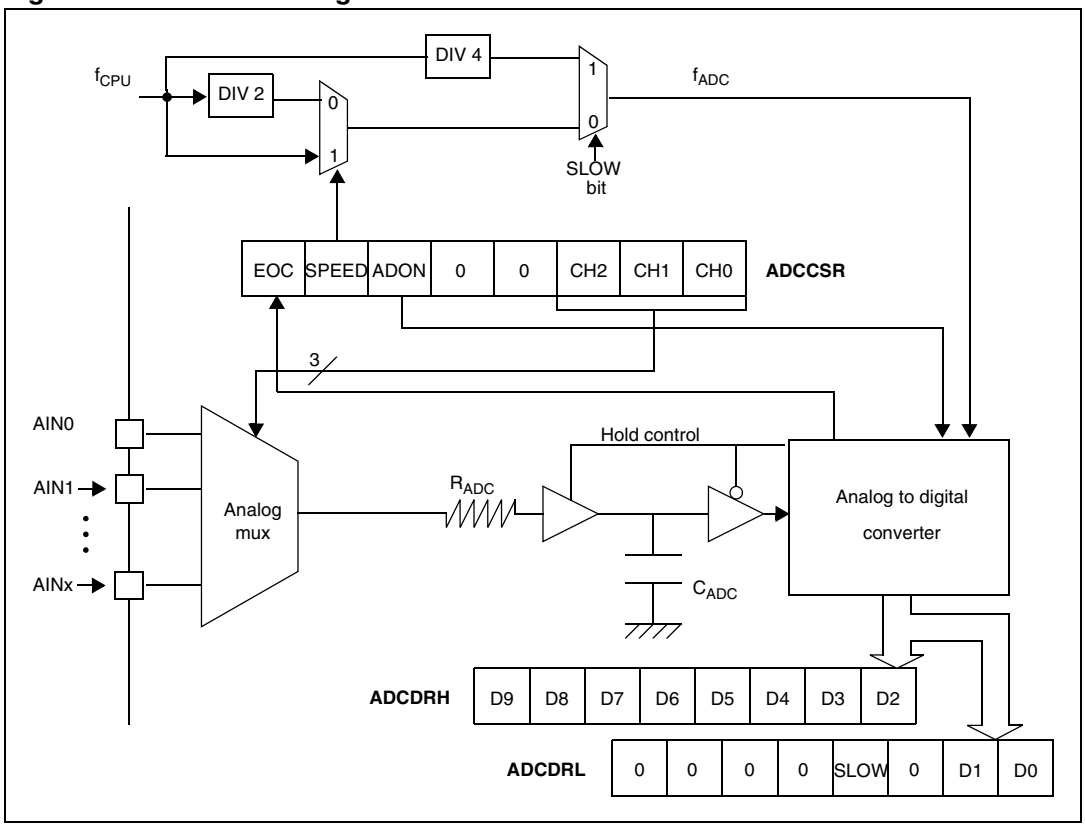
10.3.3 Functional description

Analog power supply

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

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

Figure 35. ADC block diagram



Digital A/D conversion result

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

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

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

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the *Chapter 12: Electrical characteristics*.

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

A/D conversion phases

The A/D conversion is based on two conversion phases:

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

The total conversion time: $t_{CONV} = t_{SAMPLE} + t_{HOLD}$

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

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

A/D conversion

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

In the ADCCSR register:

Select the 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 or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

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

To read only 8 bits, perform the following steps:

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

Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[2:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

10.3.4 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 47. Effect of low power modes on ADC

Mode	Description
Wait	No effect on A/D converter
Halt	A/D converter disabled After wake-up from halt mode, the A/D converter requires a stabilization time t_{STAB} (see *Electrical characteristics*) before accurate conversions can be performed.

10.3.5 Interrupts

None.

10.3.6 Register description

Control/status register (ADCCSR)

ADCCSR Reset value: 0000 0000 (00h)

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

Table 48. ADCCSR register description

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 the table in the SLOW bit description.
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

Table 48. ADCCSR register description (continued)

Bit	Name	Function
2:0	CH[2:0]	Channel selection These bits are set and cleared by software. They select the analog input to convert as follows: 000: Channel pin = AIN0 001: Channel pin = AIN1 010: Channel pin = AIN2 011: Channel pin = AIN3 100: Channel pin = AIN4 *Note: A write to the ADCCSR register (with ADON set) aborts the current conversion, resets the EOC bit and starts a new conversion.*

Data register high (ADCDRH)

ADCDRH Reset value: 0000 0000 (00h)

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

Table 49. ADCDRH register description

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

Data register low (ADCDRL)

ADCDRL Reset value: 0000 0000 (00h)

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

Table 50. ADCDRL register description

Bit	Name	Function
7:4	-	Reserved, forced by hardware to 0
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 52*.
2	-	Reserved, forced by hardware to 0
1:0	D[1:0]	LSB of analog converted value

Table 51. ADC clock speed configuration

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

Table 52. 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	CH2 0	CH1 0	CH0 0
0035h	ADCDRH Reset value	D9 0	D8 0	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0
0036h	ADCDRL Reset value	0	0	0	0	SLOW 0	0	D1 0	D0 0

11 Instruction set

11.1 ST7 addressing modes

The ST7 core features 17 different addressing modes which can be classified in seven main groups.

Table 53. Addressing mode groups

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

The ST7 instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be divided in two submodes 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 54. ST7 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.

11.1.1 Inherent

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

Table 55. Inherent instructions

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

11.1.2 Immediate

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

Table 56. Immediate instructions

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

11.1.3 Direct

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

The direct addressing mode consists of two submodes:

Direct (short)

The address is a byte, thus requires only 1 byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

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

11.1.4 Direct 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 direct indexed addressing mode consists of three submodes:

Direct indexed (no offset)

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

Direct indexed (short)

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

Direct indexed (long)

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

11.1.5 Indirect (short, long)

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

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

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.

11.1.6 Indirect indexed (short, long)

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

The indirect indexed addressing mode consists of two submodes:

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 57. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Type	Instruction	Function
Long and short instructions	LD	Load
	CP	Compare
	AND, OR, XOR	Logical operations
	ADC, ADD, SUB, SBC	Arithmetic addition/subtraction operations
	BCP	Bit compare

Table 57. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes (continued)

Type	Instruction	Function
Short instructions only	CLR	Clear
	INC, DEC	Increment/decrement
	TNZ	Test negative or zero
	CPL, NEG	1 or 2 complement
	BSET, BRES	Bit operations
	BTJT, BTJF	Bit test and jump operations
	SLL, SRL, SRA, RLC, RRC	Shift and rotate operations
	SWAP	Swap nibbles
	CALL, JP	Call or jump subroutine

11.1.7 Relative mode (direct, indirect)

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

Table 58. Available relative direct/indirect instructions

Instruction	Function
JRxx	Conditional jump
CALLR	Call relative

The relative addressing mode consists of two submodes:

Relative (direct)

The offset follows the opcode.

Relative (indirect)

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

11.2 Instruction groups

The ST7 family devices use an instruction set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following *Table 60*.

Table 59. Instruction groups

Group	Instructions							
Load and transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/decrement	INC	DEC						

Table 59. Instruction groups (continued)

Group	Instructions							
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				

Using a prebyte

The instructions are described with 1 to 4 bytes.

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

The whole instruction becomes:

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

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

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

11.2.1 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 60. 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 >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

Table 60. Instruction set overview (continued)

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

12 Electrical characteristics

12.1 Parameter conditions

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

12.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at T_A = 25 °C and T_A = $T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean ± 3Σ).

12.1.2 Typical values

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

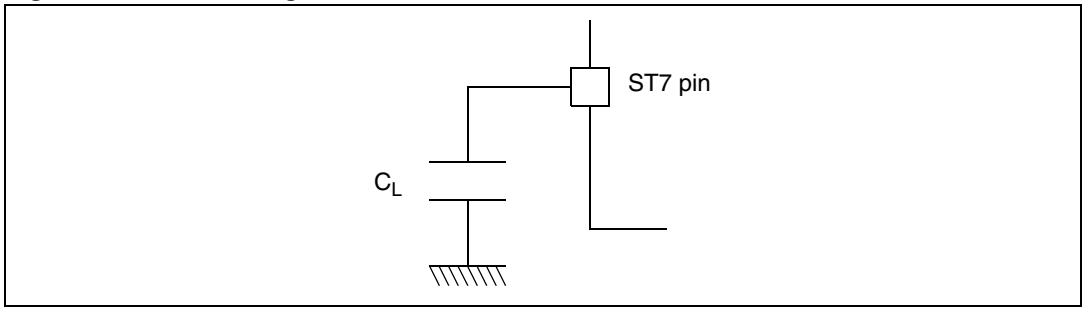
12.1.3 Typical curves

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

12.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 44*.

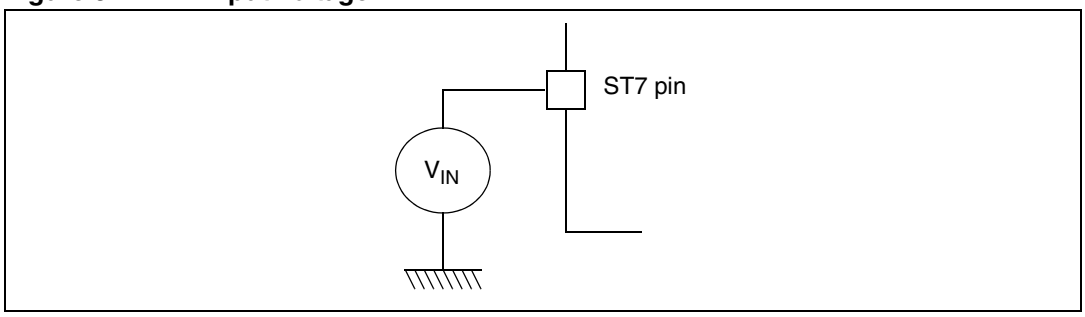
Figure 36. Pin loading conditions



12.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 45*.

Figure 37. Pin input voltage



12.2 Absolute maximum ratings

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

Table 61. 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 12.7.2 on page 114*	
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine model)	see *Section 12.7.2 on page 114*	

1. Directly connecting the I/O pins to V_{DD} or V_{SS} could damage the device if an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 10kΩ for I/Os). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.
2. $I_{INJ(PIN)}$ 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 $V_{IN} < V_{SS}$.

Table 62. 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 $\overline{RESET}$ pin	±5	
	Injected current on any other pin(4)	±5	
$\Sigma I_{INJ(PIN)}$(2)	Total injected current (sum of all I/O and control pins)(4)	±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 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
4. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterisation with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

Table 63. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see *Chapter 13: Package characteristics*)		

12.3 Operating conditions

Table 64. 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	$V_{DD} \geq 3V$	0	16	MHz
T_A	Ambient temperature range	A Suffix version	-40	+85	°C
		C Suffix version		+125	

Figure 38. f_{CLKIN} maximum operating frequency versus V_{DD} supply voltage

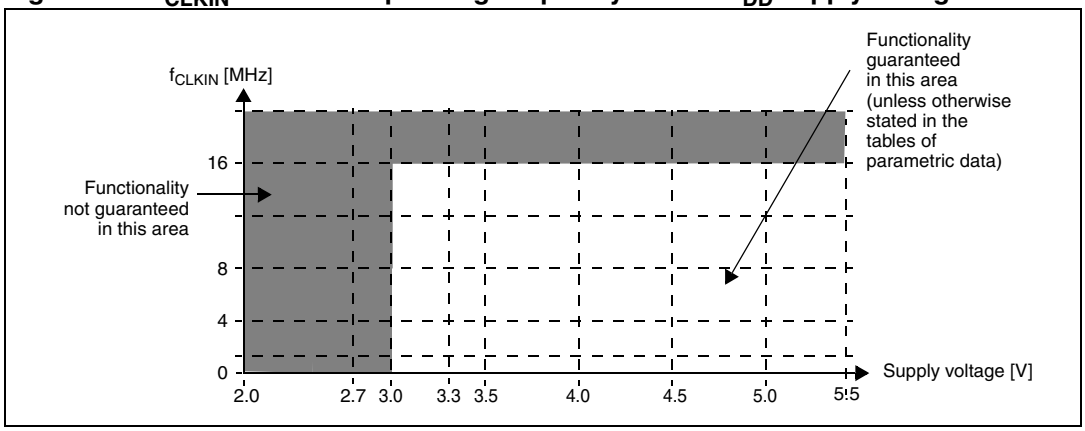


Table 65. Operating conditions with low voltage detector (LVD)(1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)	High threshold	3.50(2)	4.00	4.50	V
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)		3.30	3.80	4.40(2)	
V_{hys}	LVD voltage threshold hysteresis	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$		150		mV
Vt_{POR}	V_{DD} rise time rate(3)(4)		20	-		µs/V
$I_{DD(LVD)}$(2)	LVD current consumption	V_{DD} = 5V		220		µA

1. T_A = -40 to 125°C unless otherwise specified.
2. Not tested in production
3. Not tested in production. The V_{DD} rise time rate condition is needed to ensure a correct device power-on and LVD reset release. When the V_{DD} slope is outside these values, the LVD may not release properly the 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 0V to ensure optimum restart conditions. Refer to circuit example in *Figure 67 on page 124*.

12.3.1 Internal RC oscillator

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

Note: 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.

Table 66. Internal RC oscillator calibrated at 5.0V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC}	Internal RC oscillator frequency	RCCR = FF (reset value), T_A = 25°C, V_{DD} = 5V		4.5		MHz
		RCCR = RCCR0(1), T_A = 25°C, V_{DD} = 5V		8		
ACC_{RC}	RC resolution	V_{DD} = 5V(2)	-2		+2	%
	Accuracy of internal RC oscillator with RCCR = RCCR0(1)	T_A = -40°C to +125°C, V_{DD} = 5V	-7		+7	
		T_A = -40°C to +125°C, V_{DD} = 4.5 to 5.5V(2)	-8		+9	
$I_{DD(RC)}$	RC oscillator current consumption	T_A = 25°C, V_{DD} = 5V			900(3)	µA
$t_{su(RC)}$	RC oscillator setup time	T_A = 25°C, V_{DD} = 5V		4(3)		µs

1. See *Internal RC oscillator adjustment on page 31*
2. Data based on characterization results; not tested in production
3. Guaranteed by Design

Table 67. Internal RC oscillator calibrated at 3.3V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC}	Internal RC oscillator frequency	RCCR = FF (reset value), T_A = 25°C, V_{DD} = 3.3V		4.4		MHz
		RCCR = RCCR1(1), T_A = 25°C, V_{DD} = 3.3V		8		
ACC_{RC}	RC resolution	V_{DD} = 3.3V(2)	-2		+2	%
	Accuracy of internal RC oscillator with RCCR = RCCR1(1)	T_A = -40°C to +125°C, V_{DD} = 3.3V	-7		+7	
		T_A = -40°C to +125°C, V_{DD} = 3.0 to 3.6V(2)	-8		+9	
$I_{DD(RC)}$	RC oscillator current consumption	T_A = 25°C, V_{DD} = 3.3V			900(3)	µA
$t_{su(RC)}$	RC oscillator setup time	T_A = 25°C, V_{DD} = 3.3V		4(3)		µs

1. See *Internal RC oscillator adjustment on page 31*
2. Data based on characterization results; not tested in production
3. Guaranteed by design

Figure 39. Internal RC oscillator frequency vs. temperature (RCCR = RCCR0) at V_{DD} = 5.0V

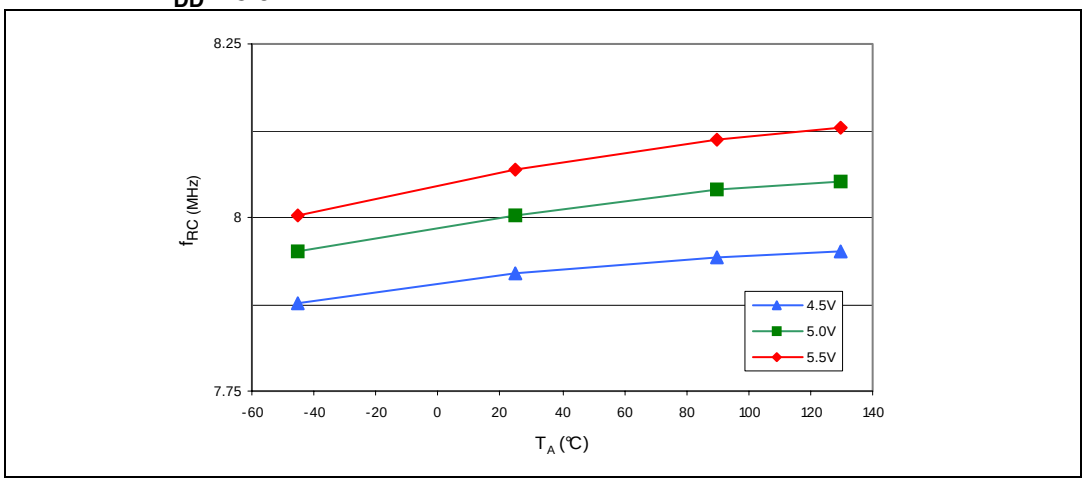
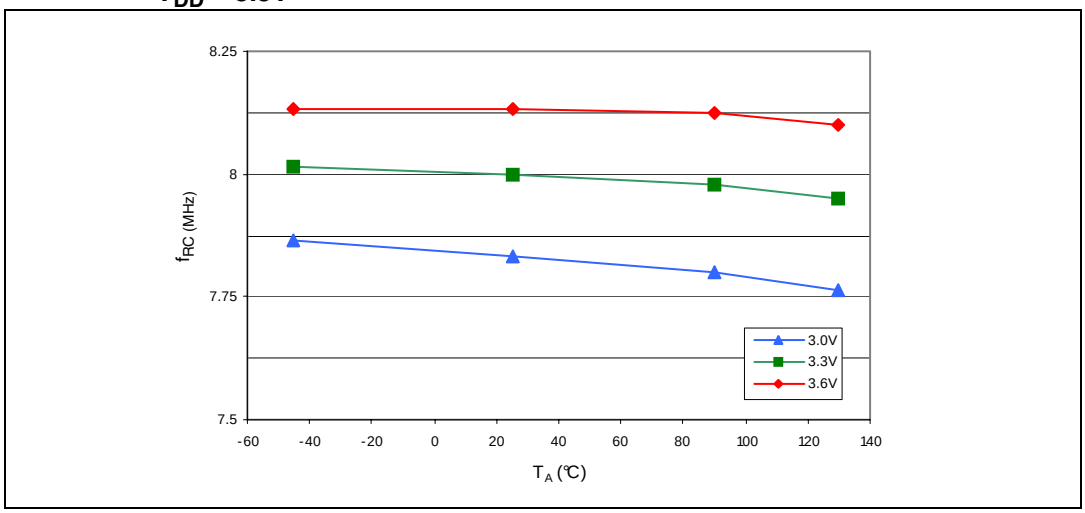


Figure 40. Internal RC oscillator frequency vs. temperature (RCCR = RCCR1) at V_{DD} = 3.3V



12.4 Supply current characteristics

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for halt mode for which the clock is stopped).

Table 68. Supply current consumption

Symbol	Parameter	Conditions(1)	Typ	Max	Unit
I_{DD}	Supply current in run mode(2)	f_{CPU} = 8 MHz	5.0	9.0	mA
	Supply current in wait mode(3)	f_{CPU} = 8 MHz	1.5	4.0	
	Supply current in slow mode(4)	f_{CPU}/32 = 250 kHz	650	1100	µA
	Supply current in slow wait mode(5)	f_{CPU}/32 = 250 kHz	500	900	
	Supply current in AWUFH mode(6)(7)		40	120(8)	
	Supply current in active halt mode		100	250	
	Supply current in halt mode(9)		0.5	3	

1. T_A = -40 to +125 °C unless otherwise specified
2. 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.
3. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
4. Slow mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
5. 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.
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.
8. Data based on characterization, not tested in production
9. 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.

Figure 41. Typical I_{DD} in run mode vs. f_{CPU}

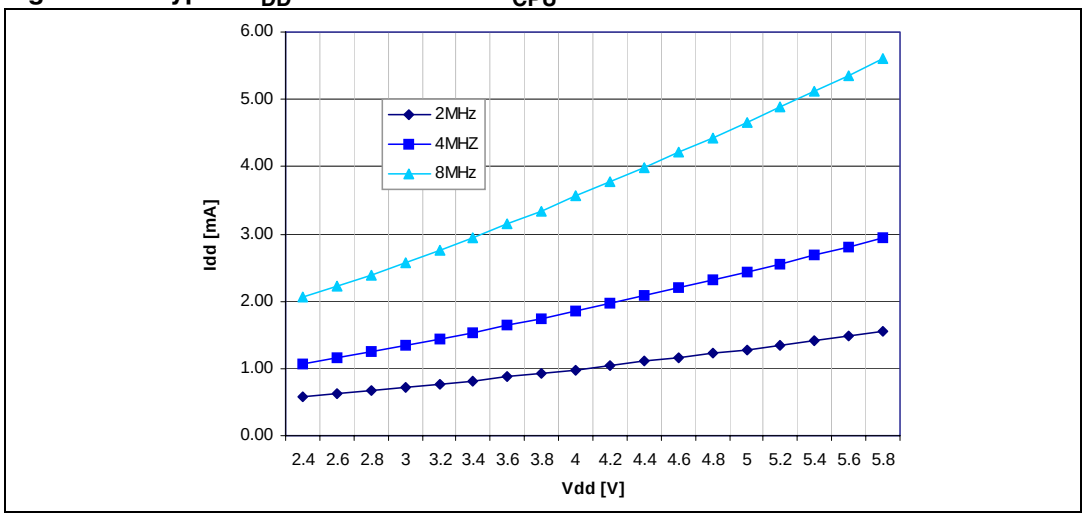


Figure 42. Typical I_{DD} in slow mode vs. f_{CPU}

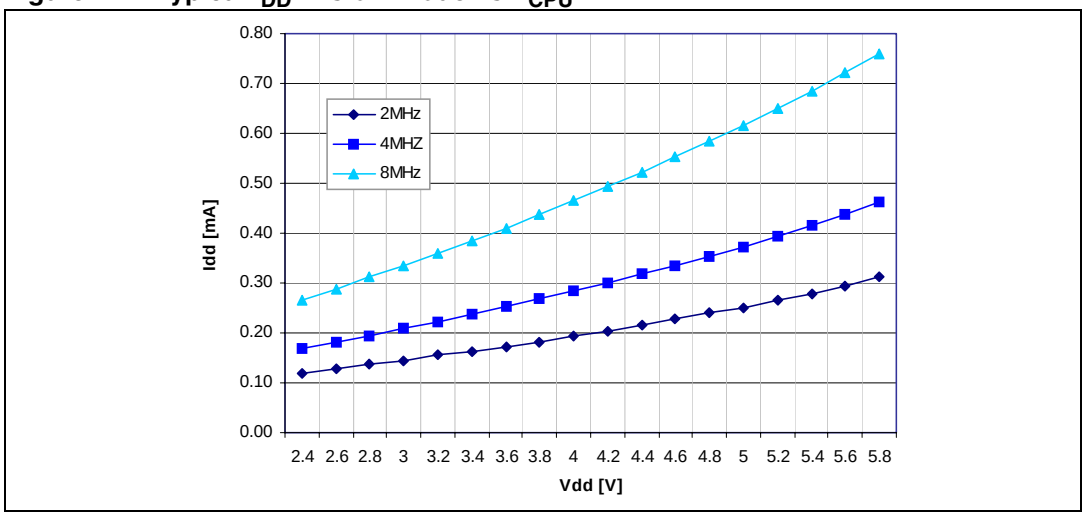


Figure 43. Typical I_{DD} in wait mode vs. f_{CPU}

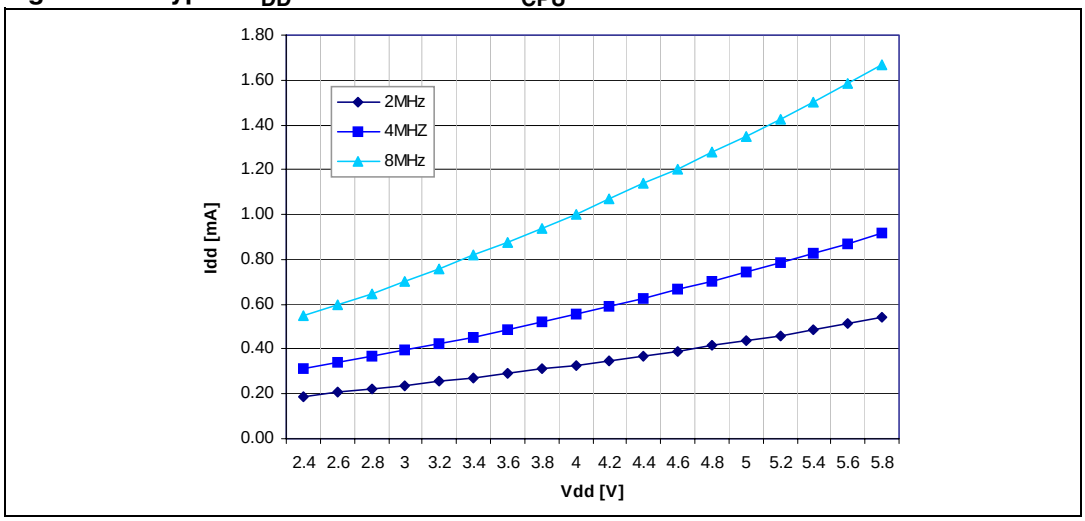


Figure 44. Typical I_{DD} in slow wait mode vs. f_{CPU}

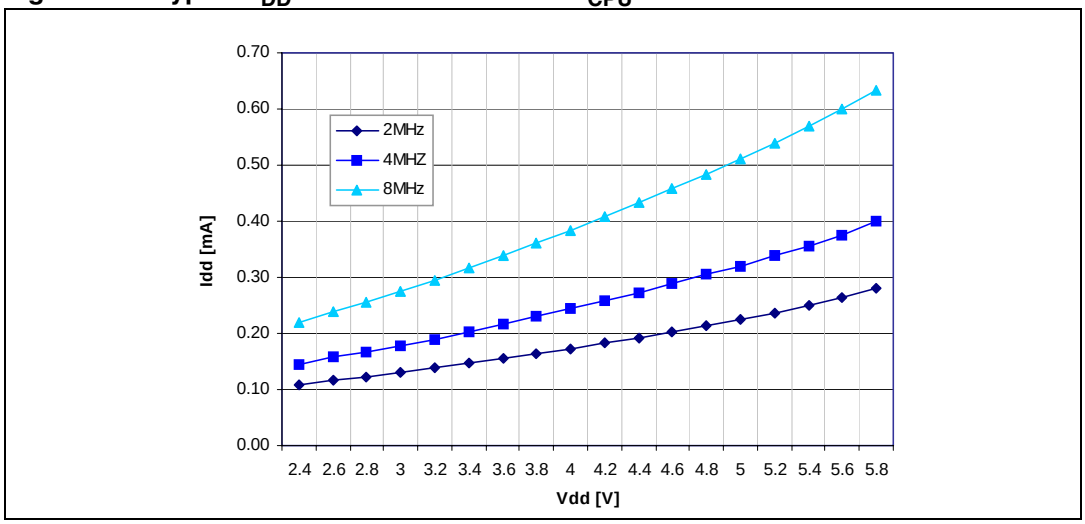


Figure 45. Typical I_{DD} vs. temperature at V_{DD} = 5 V and f_{CPU} = 8 MHz

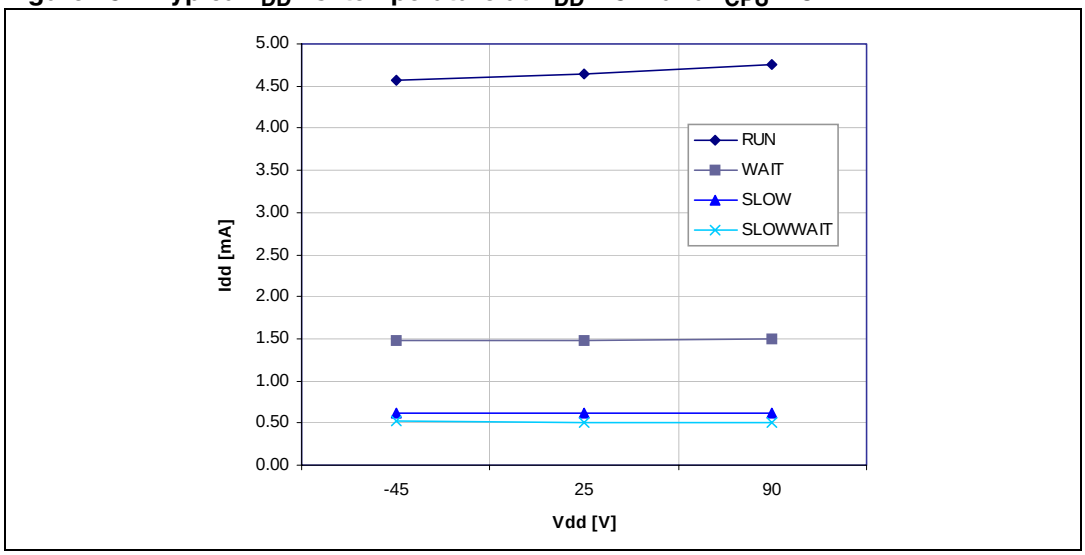


Table 69. On-chip peripherals

Symbol	Parameter	Conditions		Typ(1)	Unit
$I_{DD(AT)}$	12-bit auto-reload timer supply current(2)	f_{CPU} = 8 MHz	V_{DD} = 5.0V	30	µA
$I_{DD(ADC)}$	ADC supply current when converting(3)	f_{ADC} = 4 MHz	V_{DD} = 5.0V	750	

1. Not tested in production, guaranteed by characterization
2. Data based on a differential I_{DD} measurement between reset configuration (timer stopped) and the timer running in PWM mode at f_{CPU} = 8 MHz
3. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions with amplifier off

12.5 Clock and timing characteristics

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

Table 70. 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. Data based on characterization, not tested in production
2. Data based on typical application software
3. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

Table 71. Auto wake-up RC oscillator

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{DD(RCAWU)}$	Supply voltage range		3.0	5.0	5.5	V
$T_{A(RCAWU)}$	Operating temperature range		-40	25	125	°C
$I_{DD(RCAWU)}$	Current consumption[1]	Without prescaler	2.0	8.0	14.0	µA
		AWU RC switched off		0		
$f_{OSC(RCAWU)}$	Output frequency[1]		20	33	60	kHz

1. Data guaranteed by Design

12.6 Memory characteristics

Table 72. RAM and hardware registers[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode[2]	Halt mode (or reset)	1.6			V

1. T_A = -40°C to 125°C, unless otherwise specified.
2. 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). Guaranteed by construction, not tested in production.

Table 73. Flash program memory

Symbol	Parameter	Conditions[1]	Min	Typ	Max	Unit
V_{DD}	Operating voltage for Flash write/erase		3.0		5.5	V
t_{prog}	Programming time for 1~32 bytes[2]	T_A = -40 to +125°C		5	10	ms
	Programming time for 1 Kbyte	T_A = +25°C		0.16	0.32	s
t_{RET}	Data retention[3]	T_A = +55°C[4]	20			years
N_{RW}	Write/erase cycles	T_A = +25°C	10k[5]			cycles

Table 73. Flash program memory (continued)

Symbol	Parameter	Conditions(1)	Min	Typ	Max	Unit
I_{DD}	Supply current(6)	Read/write/erase modes f_{CPU} = 8 MHz, V_{DD} = 5.5V			2.6	mA
		No read/no write mode			100	µA
		Power down mode/halt		0	0.1	

1. T_A = -40°C to 85°C, unless otherwise specified.
2. Up to 32 bytes can be programmed at a time.
3. Data based on reliability test results and monitored in production.
4. The data retention time increases when the T_A decreases.
5. Design target value pending full product characterization.
6. Guaranteed by design. Not tested in production.

Table 74. EEPROM memory (ST7FLU09 only)

Symbol	Parameter	Conditions(1)	Min	Typ	Max	Unit
V_{DD}	Operating voltage for EEPROM write/erase	Refer to operating range of V_{DD} with T_A, *Section 12.3 on page 104*	3.0		5.5	V
t_{prog}	Programming time for 1~32 bytes			5	10	ms
t_{RET}(2)	Data retention(3)	T_A = +55°C	20			years
N_{RW}	Write/erase cycles	T_A = +25°C	300k			cycles

1. T_A = -40°C to 125°C, unless otherwise specified.
2. Data based on reliability test results and monitored in production.
3. The data retention time increases when the T_A decreases.

12.7 EMC (electromagnetic compatibilty) characteristics

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

12.7.1 Functional EMS (electromagnetic susceptibility)

Based on a simple running application on the product (toggling two LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- ESD: Electrostatic 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 given in *Table 77* are 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 $\overline{\text{RESET}}$ 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 behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Table 75. EMS 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} = 5V, T_A = +25°C, f_{OSC} = 8 MHz, SO8 package, conforms to IEC 1000-4-2	3B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD} = 5V, T_A = +25°C, f_{OSC} = 8 MHz, SO8 package, conforms to IEC 1000-4-4	4A

12.7.2 Electromagnetic interference (EMI)

Based on a simple application running on the product (toggling two 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 76. EMI emissions(1)

Symbol	Parameter	Conditions	Monitored frequency band	Max vs. [f_{OSC}/f_{CPU}]	Unit
				-/8 MHz	
S_{EMI}	Peak level	V_{DD} = 5V, T_A = +25°C, SO8 package, conforming to SAE J 1752/3	0.1 MHz to 30 MHz	21	dBµV
			30 MHz to 130 MHz	23	
			130 MHz to 1 GHz	10	
			SAE EMI Level	3	-

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

12.7.3 Absolute maximum ratings (electrical sensitivity)

Based on three different tests (ESD and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity.

Electrostatic discharge (ESD)

Electrostatic 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). This test conforms to the AEC-Q100-002/-003/-011 standard. For more details, refer to the application note AN1181.

Table 77. Absolute maximum ratings

Symbol	Ratings	Conditions	Class	Max value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human body model)	T_A = +25 °C conforming to AEC-Q100-002	H2	4000	V
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine model)	T_A = +25 °C conforming to AEC-Q100-003	M3	400	
$V_{ESD(CDM)}$	Electrostatic discharge voltage (Charged device model)	T_A = +25 °C conforming to AEC-Q100-011	C3A	750	

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

Static latch-up (LU)

Two complementary static tests are required on six parts to assess the latch-up performance:

- A supply overvoltage is applied to each power supply pin.
- A current injection is applied to each input, output and configurable I/O pin.

These tests are compliant with the EIA/JESD78 IC latch-up standard.

Electrical sensitivities

Table 78. Latch-up results

Symbol	Parameter	Conditions	Class
LU	Static latch-up class	T_A = +125°C conforming to JESD78	II level A

12.8 I/O port pin characteristics

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

Table 79. General characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage	-40°C to 125°C				$0.3V_{DD}$	V
V_{IH}	Input high level voltage			$0.7V_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis(1)				400		mV
I_{lkg}	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)(4)	$V_{IN} = V_{SS}$	V_{DD} = 5V	70	120	200	kΩ
			V_{DD} = 3.3V		200(1)		
C_{IO}	I/O pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time(1)	C_L = 50 pF between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time(1)				25		
$t_{w(IT)in}$	External interrupt pulse time(5)			1			t_{CPU}

1. Data based on characterization results, not tested in production
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 58*). 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 56*).
4. R_{PU} not applicable on PA3 because it is multiplexed on $\overline{\text{RESET}}$ pin
5. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

Figure 46. Two typical applications with unused I/O pin

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.

Note: 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.

Figure 47. Typical I_{PU} vs. V_{DD} with $V_{IN} = V_{SS}$

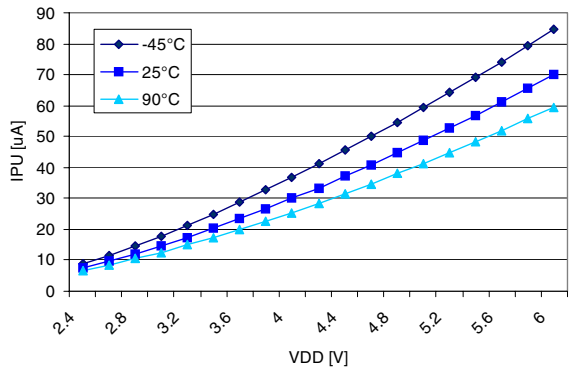


Table 80. Output driving current(1)

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OL}(2)	Output low level voltage for PA3/$\overline{RESET}$ standard I/O pin (see *Figure 58*)	I_{IO} = +5mA		1200	mV
		I_{IO} = +2mA		400	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 60*)	I_{IO} = +20mA		1300	
		I_{IO} = +8mA		750	
V_{OH}(3)(4)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 63*)	I_{IO} = -5mA	V_{DD} - 1500		
		I_{IO} = -2mA	V_{DD} - 800		

1. Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified.
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Table 63: Current characteristics* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The I_{IO} current sourced must always respect the absolute maximum rating specified in *Table 63: Current characteristics* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins do not have V_{OH}.
4. Not tested in production, based on characterization results.

Figure 48. Typical V_{OL} at V_{DD} = 3.3V (standard pins)

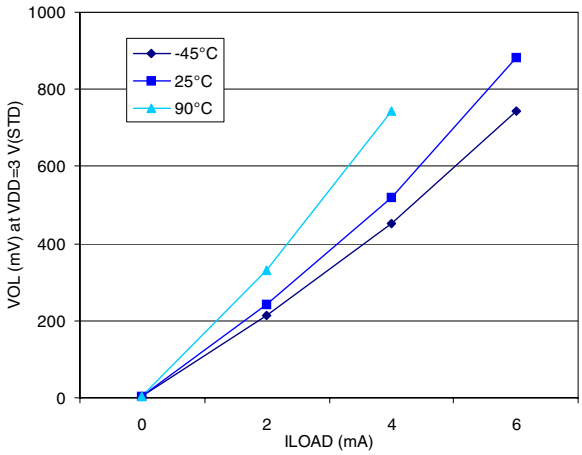


Figure 49. Typical V_{OL} at V_{DD} = 5V (standard pins)

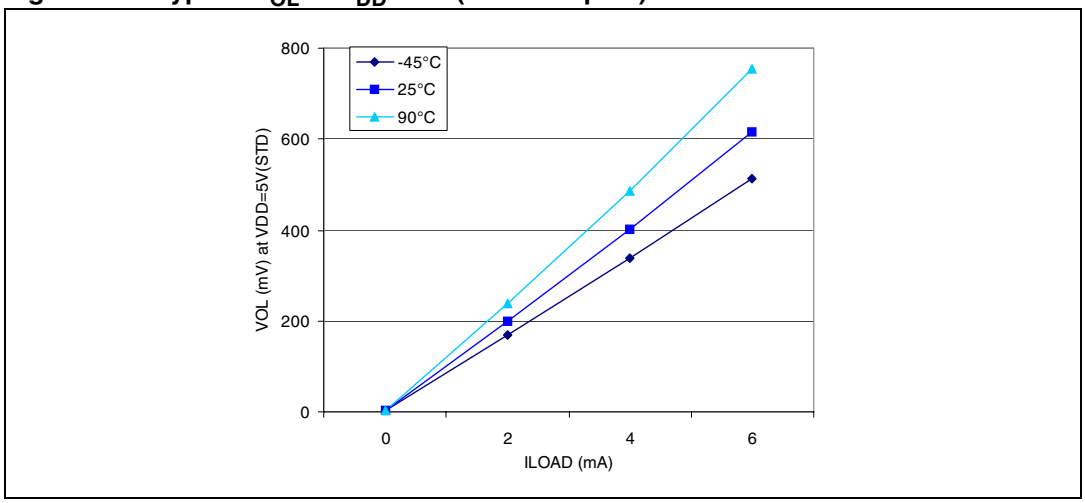


Figure 50. Typical V_{OL} at V_{DD} = 3.3V (HS pins)

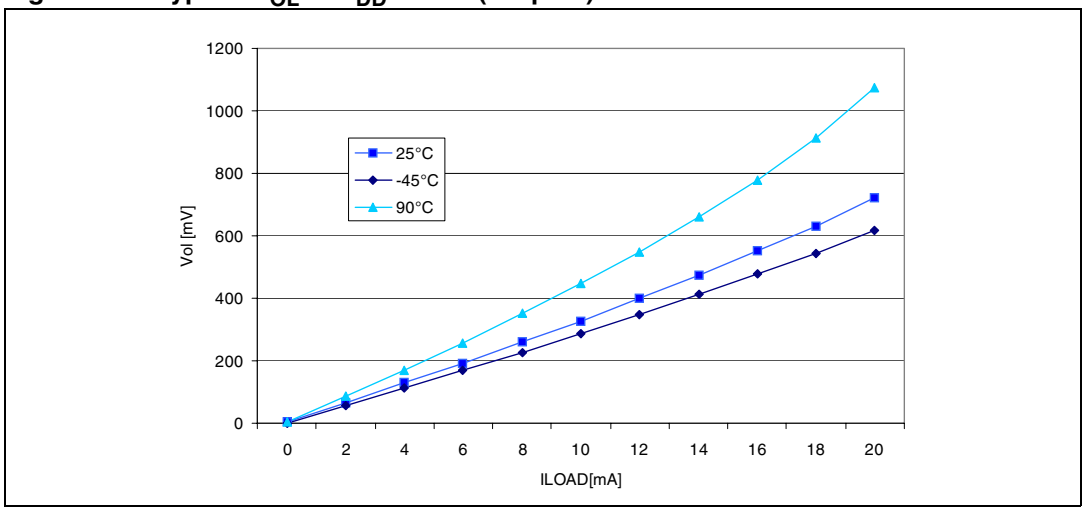


Figure 51. Typical V_{OL} at V_{DD} = 5V (HS pins)

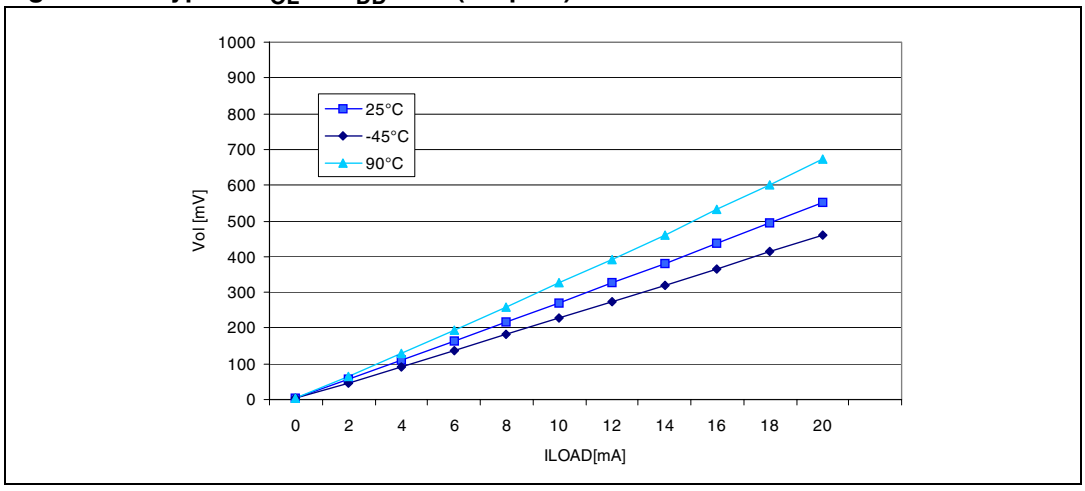


Figure 52. Typical V_{DD} - V_{OH} at V_{DD} = 3.3V (HS pins)

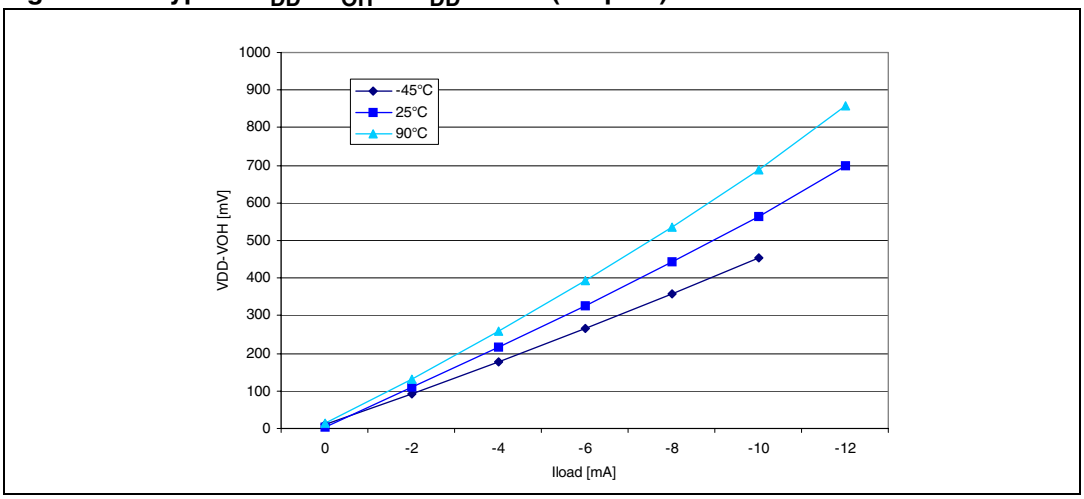


Figure 53. Typical V_{DD} - V_{OH} at V_{DD} = 5V (HS pins)

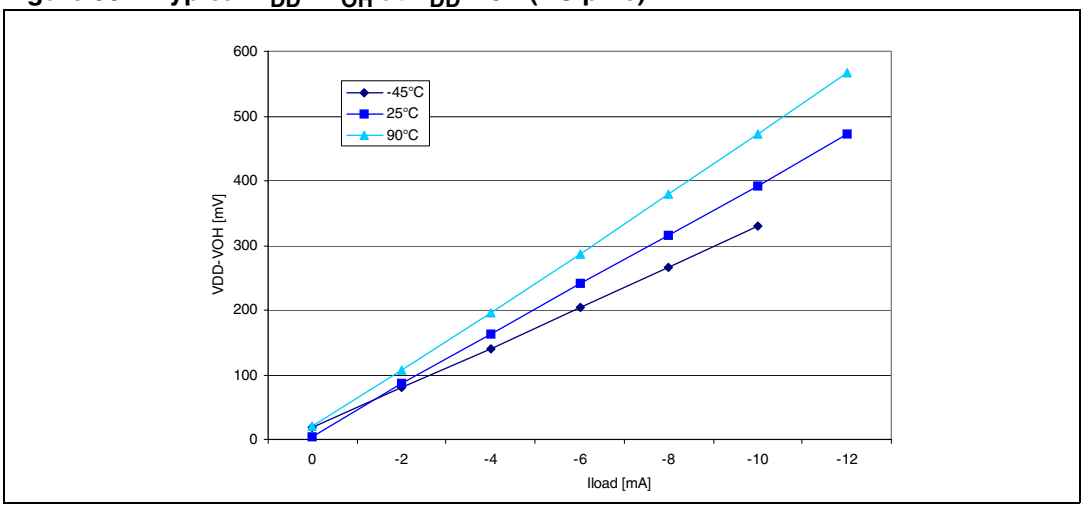


Figure 54. Typical V_{OL} vs. V_{DD} (standard I/Os)

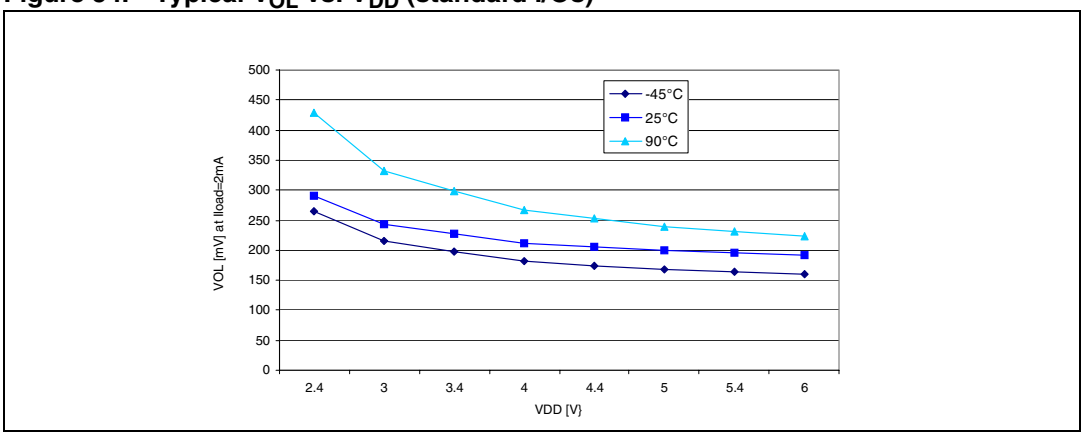


Figure 55. Typical V_{OL} vs. V_{DD} (HS pins)

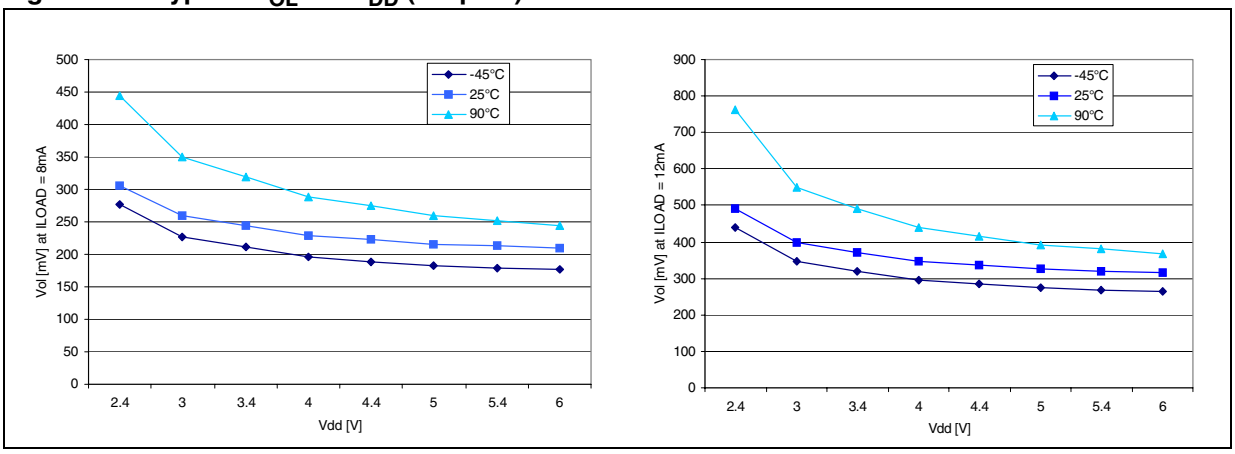
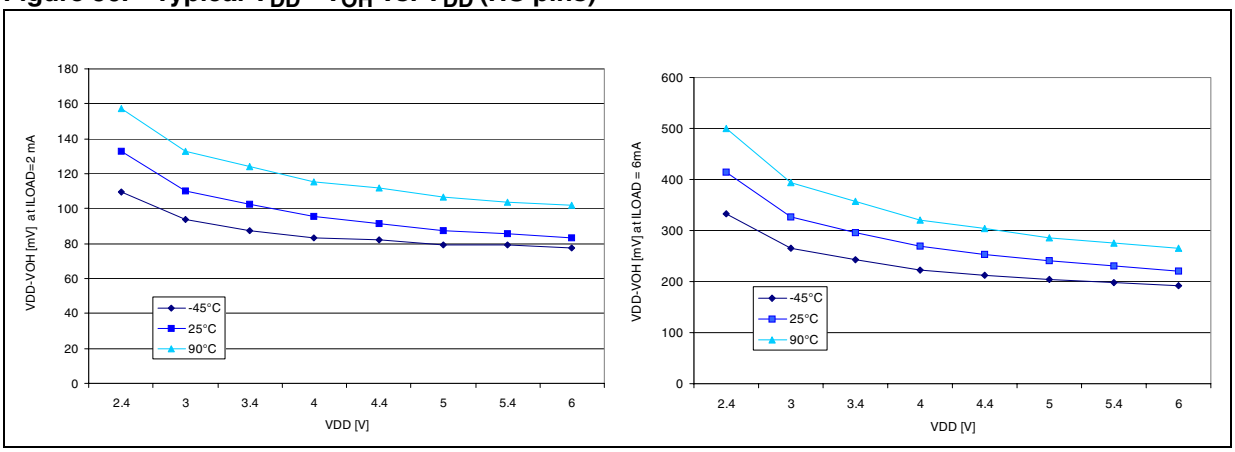


Figure 56. Typical V_{DD} - V_{OH} vs. V_{DD} (HS pins)



12.9 Control pin characteristics

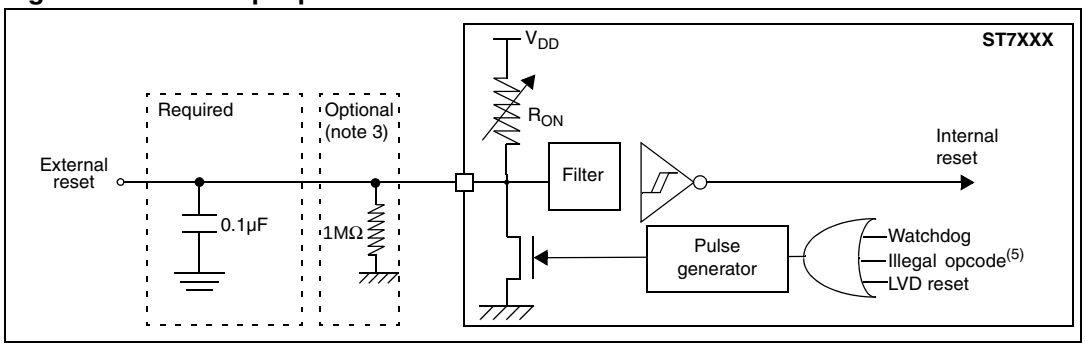
T_A = -40°C to 125°C, unless otherwise specified.

Table 81. Asynchronous $\overline{\text{RESET}}$ pin

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage			V_{SS} - 0.3		$0.3V_{DD}$	V
V_{IH}	Input high level voltage			$0.7V_{DD}$		V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis(1)				2		
V_{OL}	Output low level voltage(2)	V_{DD} = 5V	I_{IO} = +2mA			400	mV
R_{ON}	Pull-up equivalent resistor(3)	$V_{IN} = V_{SS}$	V_{DD} = 5V	10	50	70	kΩ
			V_{DD} = 3.3V		90(1)		
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset sources			90(1)		µs
$t_{h(RSTL)in}$	External reset pulse hold time(4)			20			
$t_{g(RSTL)in}$	Filtered glitch duration				200		ns

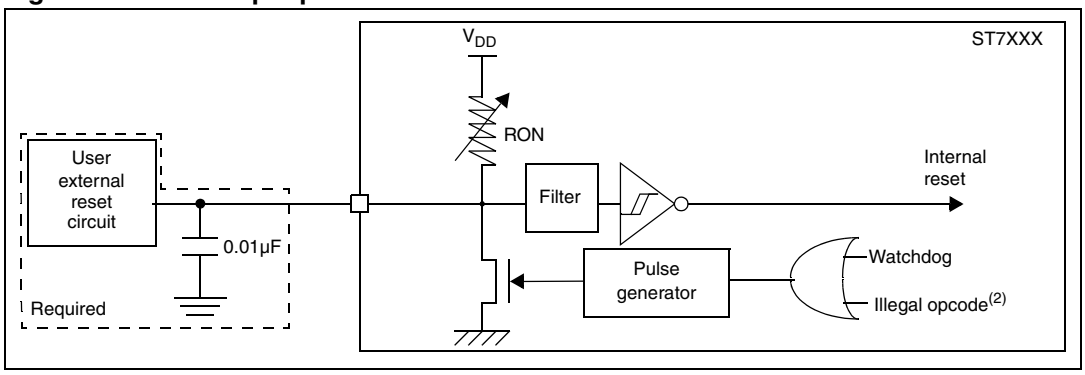
1. Data based on characterization results, not tested in production
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Table 63: Current characteristics* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The R_{ON} pull-up equivalent 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 has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

Figure 57. $\overline{\text{RESET}}$ pin protection when LVD is enabled(1)(2)(3)(4)



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} max. level specified in *Table 85 on page 123*. Otherwise the reset will not be 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 *Table 63 on page 104*.
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. In case 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 will add 5µA to the power consumption of the MCU).
4. Tips when using the LVD:
   A. Check that all recommendations related to ICCCLK and reset circuit have been applied (see caution in *Table 2 on page 15* and notes above)
   B. 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.
   C. 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 A and B above are sufficient for a robust solution. Otherwise: replace 10nF pull-down on the $\overline{\text{RESET}}$ pin with a 5µF to 20µF capacitor.

Figure 58. $\overline{\text{RESET}}$ pin protection when LVD is disabled(1)



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} max. level specified in *Table 85 on page 123*. Otherwise the reset will not be 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 *Table 63 on page 104*.
2. Please refer to *Illegal opcode reset on page 99* for more details on illegal opcode reset conditions.

12.10 10-bit ADC characteristics

Table 82. 10-bit ADC characteristics(1)

Symbol	Parameter	Conditions	Min	Typ(2)	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion voltage range(3)		V_{SSA}		V_{DDA}	V
R_{AIN}	External input resistor	V_{DD} = 5V, f_{ADC} = 4 MHz			8k(4)	Ω
		V_{DD} = 3.3V, f_{ADC} = 4 MHz			7k(4)	
C_{ADC}	Internal sample and hold capacitor			3		pF
t_{STAB}	Stabilization time after ADC enable	f_{CPU} = 8 MHz, f_{ADC} = 4 MHz	0(5)			µs
t_{ADC}	Conversion time (sample+hold)		3.5			
	Sample capacitor loading time		4			$1/f_{ADC}$
	Hold conversion time		10			

1. Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.
2. Unless otherwise specified, typical data are based on T_A = 25°C and V_{DD} - V_{SS} = 5V. They are given only as design guidelines and are not tested.
3. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refers to V_{DD} and V_{SS}.
4. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.
5. The stabilization time of the A/D converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

Figure 59. Typical application with ADC

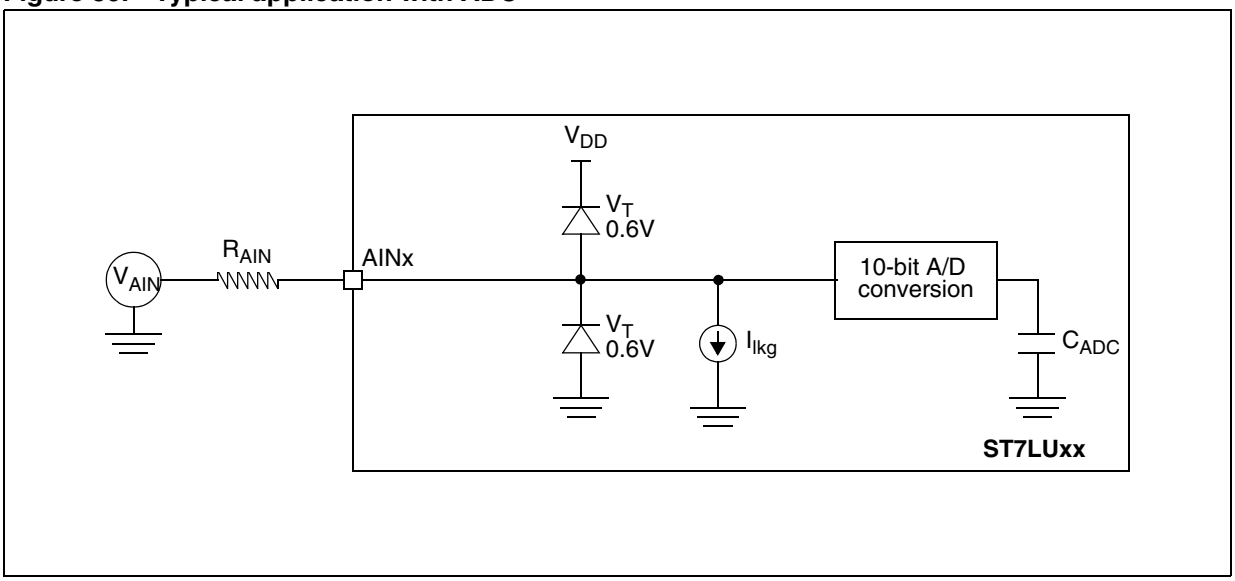


Table 83. ADC accuracy with V_{DD} = 4.5V to 5.5V

Symbol[1]	Parameter	Conditions	Typ	Max	Unit
$\|E_T\|$	Total unadjusted error	f_{CPU} = 8 MHz, f_{ADC} = 4 MHz[1]	2.1	5.0	LSB
$\|E_O\|$	Offset error		0.2	2.5	
$\|E_G\|$	Gain error		0.3	1.5	
$\|E_D\|$	Differential linearity error		1.9	3.5	
$\|E_L\|$	Integral linearity error		1.9	4.5	

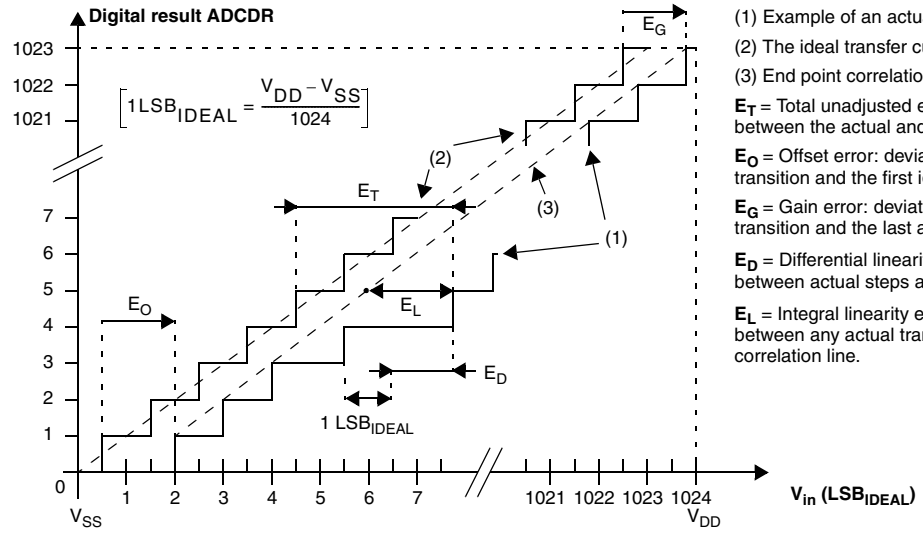
1. Data based on characterization results over the whole temperature range

Table 84. ADC accuracy with V_{DD} = 3.0V to 3.6V

Symbol[1]	Parameter	Conditions	Typ	Max	Unit
$\|E_T\|$	Total unadjusted error	f_{CPU} = 4 MHz, f_{ADC} = 2 MHz[1]	2.0	3.0	LSB
$\|E_O\|$	Offset error		0.1	1.5	
$\|E_G\|$	Gain error		0.4	1.4	
$\|E_D\|$	Differential linearity error		1.8	2.5	
$\|E_L\|$	Integral linearity error		1.7	2.5	

1. Data based on characterization results over the whole temperature range

Figure 60. ADC accuracy characteristics



(1) Example of an actual transfer curve

(2) The ideal transfer curve

(3) End point correlation line

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.

13 Package characteristics

13.1 ECOPACK®

In order to meet environmental requirements, ST offers these devices 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*.

13.2 Package mechanical data

Figure 61. 8-pin plastic small outline package, 150-mil width

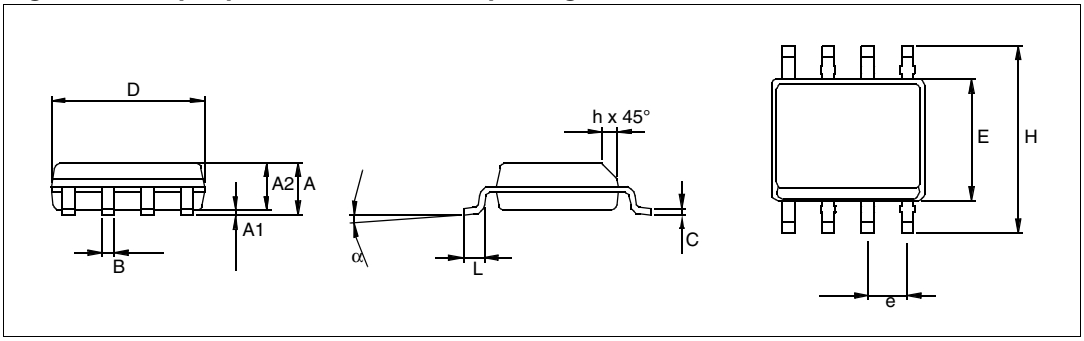


Table 85. 8-pin plastic small outline package, 150-mil width, mechanical data

Dim.	mm			inches(1)		
	Min	Typ	Max	Min	Typ	Max
A	1.350		1.750	0.0531		0.0689
A1	0.100		0.250	0.0039		0.0098
A2	1.100		1.650	0.0433		0.0650
B	0.330		0.510	0.0130		0.0201
C	0.190		0.250	0.0075		0.0098
D	4.800		5.000	0.1890		0.1969
E	3.800		4.000	0.1496		0.1575
e		1.270			0.0500	
H	5.800		6.200	0.2283		0.2441
h	0.250		0.500	0.0098		0.0197
α	0°		8°			
L	0.400		1.270	0.0157		0.0500

1. Values in inches are converted from mm and rounded to 4 decimal digits

Figure 62. 16-pin plastic dual in-line package, 300-mil width

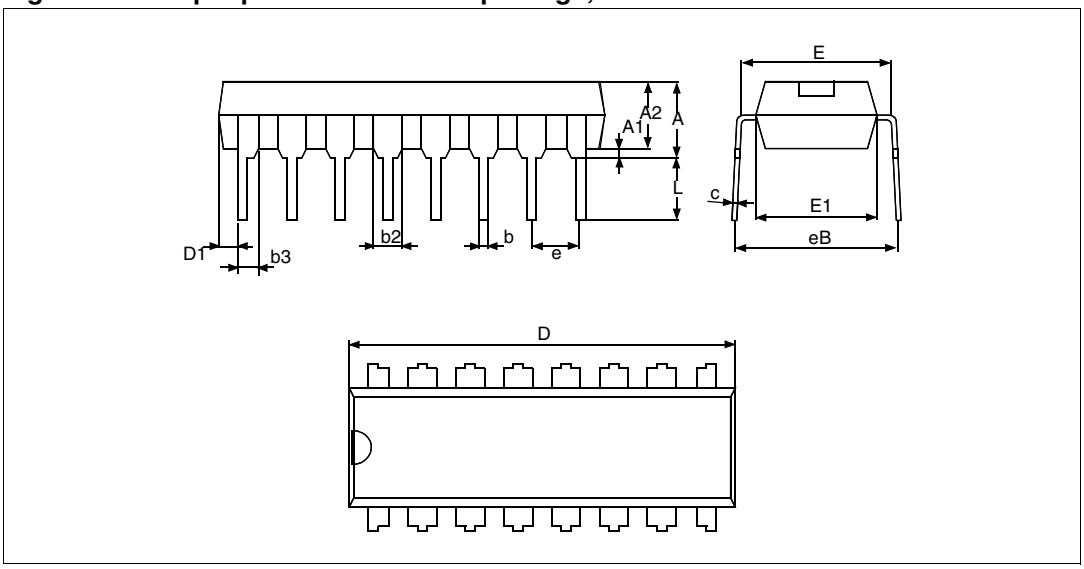


Table 86. 8-pin plastic small outline package, 150-mil width, mechanical data

Dim.	mm			inches[1][2]		
	Min	Typ	Max	Min	Typ	Max
A			5.330			0.2098
A1	0.380			0.0150		
A2	2.920	3.300	4.950	0.1150	0.1299	0.1949
b	0.360	0.460	0.560	0.0142	0.0181	0.0220
b2	1.140	1.520	1.780	0.0449	0.0598	0.0701
b3	0.760	0.990	1.140	0.0299	0.0390	0.0449
c	0.200	0.250	0.360	0.0079	0.0098	0.0142
D	18.670	19.180	19.690	0.7350	0.7551	0.7752
D1	0.130			0.0051		
e		2.540			0.1000	
E	7.620	7.870	8.260	0.3000	0.3098	0.3252
E1	6.100	6.350	7.110	0.2402	0.2500	0.2799
L	2.920	3.300	3.810	0.1150	0.1299	0.1500
eB			10.920			0.4299

1. Values in inches are converted from mm and rounded to 4 decimal digits.
2. This package is for development purposes only and is not available in production

13.3 Thermal characteristics

Table 87. Thermal characteristics

Symbol	Ratings	Package	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient)	SO8	130	°C/W
T_{Jmax}	Maximum junction temperature(1)		150	°C
P_{Dmax}	Power dissipation(2)		180	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 ($I_{DD} \times V_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.

13.4 Soldering compatibility

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

Table 88. Soldering compatibility (wave and reflow soldering process)

Package	Plating material	Pb solder paste	Pb-free solder paste
SO	NiPdAu (nickel-palladium-gold)	Yes	Yes

14 Device configuration and ordering information

This device is available for production in user programmable versions (Flash). ST7FLUxx XFlash devices are shipped to customers with a default program memory content (FFh).

14.1 Flash devices

14.1.1 Flash configuration

Table 89. Flash option bytes

	Option byte 0								Option byte 1							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Reserved				SEC0[1:0]		FMP_R	FMP_W	CKSEL[1:0]		Reserved		LVD[1:0]		WDG SW	WDG HALT
Default value	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1	1

The two option bytes are used to select the hardware configuration of the microcontroller.

The option bytes can be accessed only in programming mode (for example, using a standard ST7 programming tool).

Option byte 0

Table 90. Option byte 0 bit description

Bit	Name	Function
OPT7:4	-	Reserved, must always be 1
OPT3:2	SEC0[1:0]	Sector 0 size definition These option bits indicate the size of sector 0 as follows: 00: 0.5 Kbyte 01: 1 Kbyte 1- : 2 Kbytes
OPT1	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 *Section 4.5* and the *ST7 Flash Programming Reference Manual* for more details. 0: Readout protection off 1: Readout protection on
OPT0	FMP_W	Flash write protection This option indicates if the Flash program memory is write protected. 0: Write protection off 1: Write protection on **Warning: When this option is selected, the program memory (and the option bit itself) can never be erased or programmed again**

Option byte 1

Table 91. Option byte 1 bit description

Bit	Name	Function
OPT7:6	CKSEL[1:0]	Start-up clock selection These option bits are used to select the startup frequency. By default, the internal RC is selected. 00: Internal RC as startup clock 01: AWU RC as a startup clock 10: Reserved 11: External clock on pin PA5
OPT5	-	Reserved, must always be 1
OPT4	-	Reserved, must always be 0
OPT3:2	LVD[1:0]	Low voltage detection selection These option bits enable the LVD block with a selected threshold as follows: 10: Highest voltage threshold 11: LVD off
OPT1	WDG SW	Hardware or software watchdog This option bit selects the watchdog type. 0: Hardware (watchdog always enabled) 1: Software (watchdog to be enabled by software)
OPT0	WDG HALT	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

14.1.2 Flash ordering information

The following *Table 98* serves as a guide for ordering.

Table 92. Flash user programmable device types

Order code	Program memory	RAM/EEPROM	ADC	Temp. range	Package	Packaging
ST7FLUS5MAE	1 Kbyte Flash	128 bytes RAM/ no EEPROM	10-bit	-40°C to +85°C	SO8	Tube
ST7FLUS5MATRE						Tape and reel
ST7FLUS5MCE				-40°C to +125°C		Tube
ST7FLUS5MCTRE						Tape and reel
ST7FLU05MAE	2 Kbytes Flash			-40°C to +85°C		Tube
ST7FLU05MATRE						Tape and reel
ST7FLU05MCE				-40°C to +125°C		Tube
ST7FLU05MCTRE						Tape and reel
ST7FLITEU0ICD			-		DIP16(1)	Tube
ST7FLU09MAE		128 bytes RAM/ 128 bytes EEPROM	10-bit	-40°C to +85°C	SO8	Tube
ST7FLU09MATRE						Tape and reel
ST7FLU09MCE				-40°C to +125°C		Tube
ST7FLU09MCTRE						Tape and reel

1. For development or tool prototyping purposes only. Not orderable in production quantities. Please contact Marketing

14.2 Development tools

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.

14.2.1 Starter kits

ST offers complete, affordable **starter kits**. Starter kits are complete, affordable hardware/software tool packages that include features and samples to help you quickly start developing your application.

14.2.2 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.

14.2.3 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 sub-family 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.

14.2.4 Order codes for development and programming tools

Table 100 on page 137 lists the ordering codes for the ST7LUxx development and programming tools. For additional ordering codes for spare parts and accessories, refer to the online product selector at *www.st.com*.

Table 93. Development and programming tool order codes for the ST7LUxx family

Supported products	In-circuit debugger, RLink series[1]		Emulator	Programming tool	
	Starter kit without demo board	Starter kit with demo board	EMU series	In-circuit programmer	ST socket boards and EPBs
ST7FLUS5 ST7FLU05 ST7FLU09	STX-RLINK[2]	STFLITE-SK/RAIS[2]	ST7MDT10-EMU3	STX-RLINK ST7-STICK[3][4]	ST7SB10-SU0[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. Parallel port connection to PC

14.3 ST7 application notes

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

15 Revision history

Table 94. Document revision history

Date	Revision	Changes
25-Jul-2007	1	Initial release
15-Jan-2008	2	Removed temperature from data retention duration in *Memories on page 1* Updated *Table 1: Device summary on page 1* for memory and packages Added note to *Section 2: Package pinout and device pin description on page 14* Added MCO to pin 3 in *Figure 3* pinout and *Table 2* pin descriptions Added *Figure 4: 16-pin DIP package pinout on page 14* Replaced 25°C with $T_{Amax\ in}$ *Section 6.1 on page 31* Modified conditions and added *Note 1* to *Table 6 on page 31* Added MCO bit to *Section 6.2.1* and to the MCCSR of *Table 12 on page 36* Modified *Figure 13: Clock management block diagram on page 37* Corrected SICSR reset values for bits 6:5 in *Figure 20* and in *Table 20 on page 50* Added AWUCSR reset values for bits 2:0 in *Table 26 on page 65* Corrected name of bit 2 in *Lite timer control/status register (LTCSR) on page 77* Modified description of CNTR bits in *Table 42 on page 83* Corrected CS[2:0] to read CH[2:0] in *A/D conversion on page 89* Modified content of *ADC conversion mode on page 89* Added *Changing the conversion channel on page 89* Modified description of EOC bit in *Table 49: ADCCSR register description on page 90* Added 'direct' to instruction headings in *Section 11.1.4 on page 96* Added *Note 2* to *Table 67 on page 105* Added *Note 2*, changed T_A from 0°C to -40°C (for ACC_{RC} at V_{DD} = 5V and 3.3V), and modified min and max values (for ACC_{RC}) in *Table 68* and *Table 69 on page 106* Replaced "JESD22-A114A/A115A standard" with "AEC-Q100-002/-003/-011 standard" in *Electrostatic discharge (ESD) on page 114* Added the AEC-Q100 standards and 'class' column in *Table 80 on page 115* Updated LU and deleted DLU content in *Static and dynamic latch-up (LU) on page 116* Updated LU and removed DLU content in *Table 82: Latch-up results on page 116* Modified 'I_{lkg}' symbol and modified min and max values of R_{PU} (V_{DD} = 5V) in *Table 83 on page 117* Modified min value for R_{ON} (V_{DD} = 5V) in *Table 85 on page 123* Replaced 'I_L ±1µA' with 'I_{lkg}' in *Figure 69 on page 125* Inches rounded to four decimal places in *Figure 72 on page 128* Added *Figure 73: 16-pin plastic dual in-line package, 300-mil width on page 129* Removed ECOPACK content from *Section 13.4* and added updated ECOPACK content to *Section 13.1 on page 128* Removed FASTROM versions from *Section 14 on page 131* Added DIP16 package to *Table 98 on page 133* and modified *Note 1* Replaced *www.st.com/mcu* with *www.st.com* in *Section 14.3.4 on page 136* Removed 'DVP series' in *Table 100 on page 137*: Removed *Section 15: Known limitations on page 123*
26-Feb-2008	3	Document status promoted from 'preliminary data' to 'datasheet'