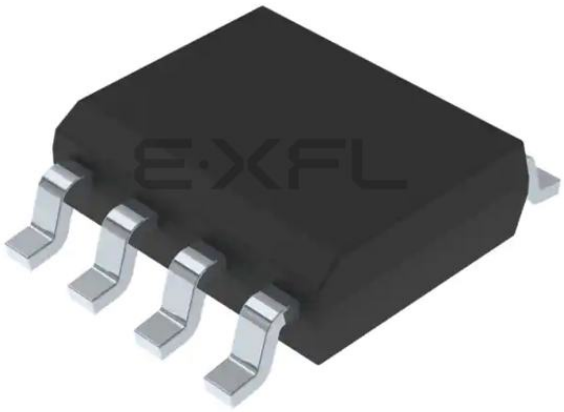


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	5
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fliteus5m6tr

4.7 Register description

4.7.1 Flash Control/Status register (FCSR)

This register controls the XFlash erasing and programming using ICP, IAP or other programming methods.

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

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

Reset value: 000 0000 (00h)

7							0
0	0	0	0	0	OPT	LAT	PGM
Read/write							

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 7* 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).

6.3.4 AVD Threshold Selection register (AVDTHCR)

Reset value: 0000 0011 (03h)

7							0
CK2	CK1	CK0	0	0	0	AVD1	AVD0
Read / Write							

Bits 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. See *Figure 10 on page 34* and *Table 6*.

Bits 4:2 Reserved, must be kept cleared.

Bits 1:0 AVD Threshold Selection bits. Refer to *Section 7.4: System integrity management (SI)*.

Table 6. Internal RC prescaler selection bits(1)

CK2	CK1	CK0	f_{OSC}
0	0	0	f_{RC}
0	0	1	$f_{RC/2}$
0	1	0	$f_{RC/4}$
0	1	1	$f_{RC/8}$
1	0	0	$f_{RC/16}$

1. If the internal RC is used with a supply operating range below 3.3 V, a division ratio of at least 2 must be enabled in the RC prescaler.

6.3.5 Clock Controller Control/Status register (CKCNTCSR)

Read/Write

Reset value: 0000 1001 (09h)

7							0
0	0	0	0	AWU_FLAG	RC_ FLAG	0	RC/AWU
Read / Write							

Bits 7:4 Reserved, must be kept cleared.

Bit 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

Bit 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

Bit 1 = Reserved, must be kept cleared.

Bit 0 = **RC/AWU** *RC/AWU Selection*
0: RC enabled
1: AWU enabled (default value)

Table 7. Clock register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0038h	MCCSR Reset value	0	0	0	0	0	0	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	0	CR1	CR0	0	0	LVDRF x	AVDF 0	AVDIE 0
003Eh	AVDTHCR reset value	CK2 0	CK1 0	CK0 0	0	0	0	AVD1 1	AVD2 1
003Fh	CKCNTCSR reset value	0	0	0	0	AWU_FLAG 1	RC_FLAG 0	0	RC/AWU 1

Figure 10. Clock management block diagram

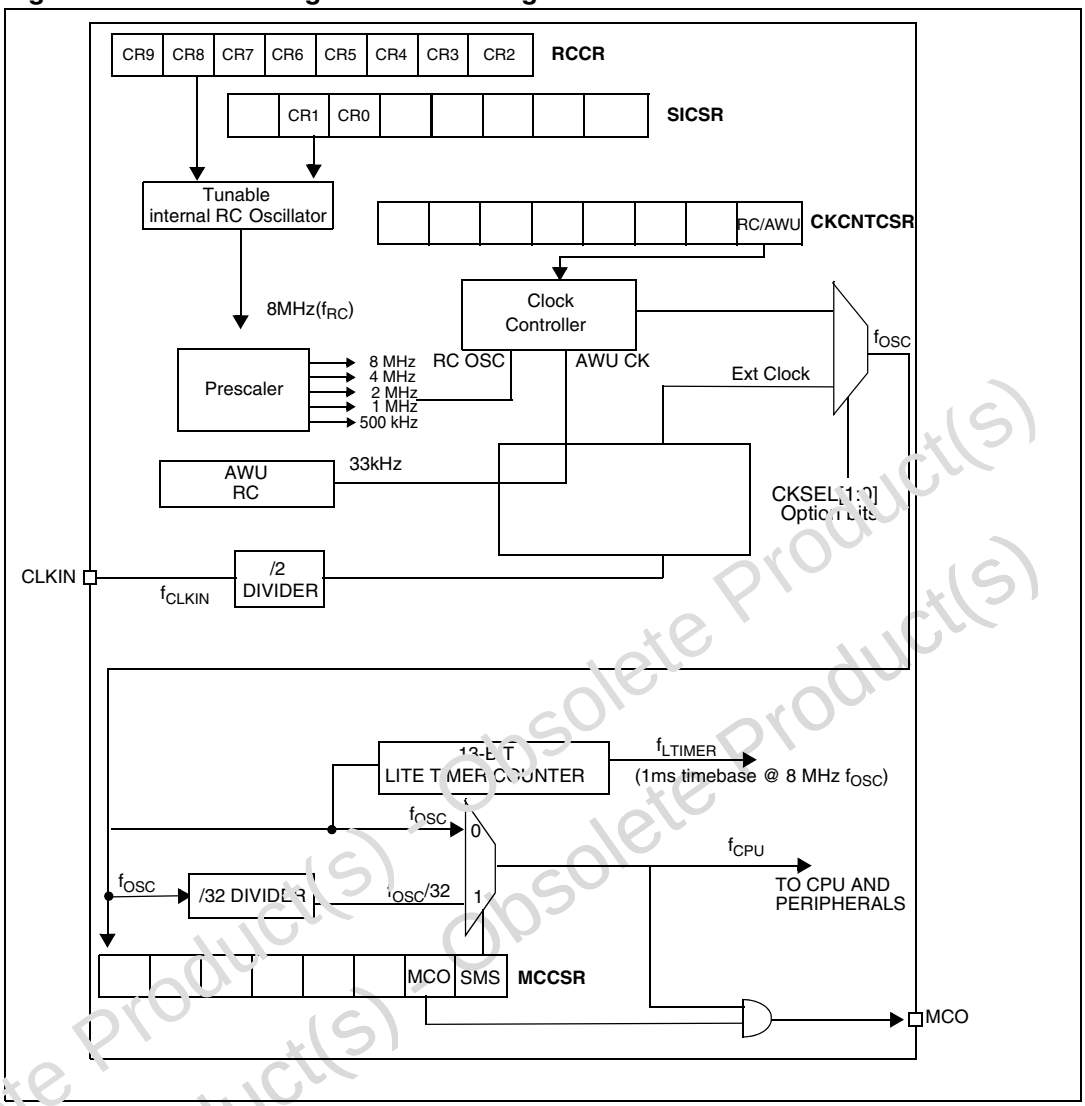
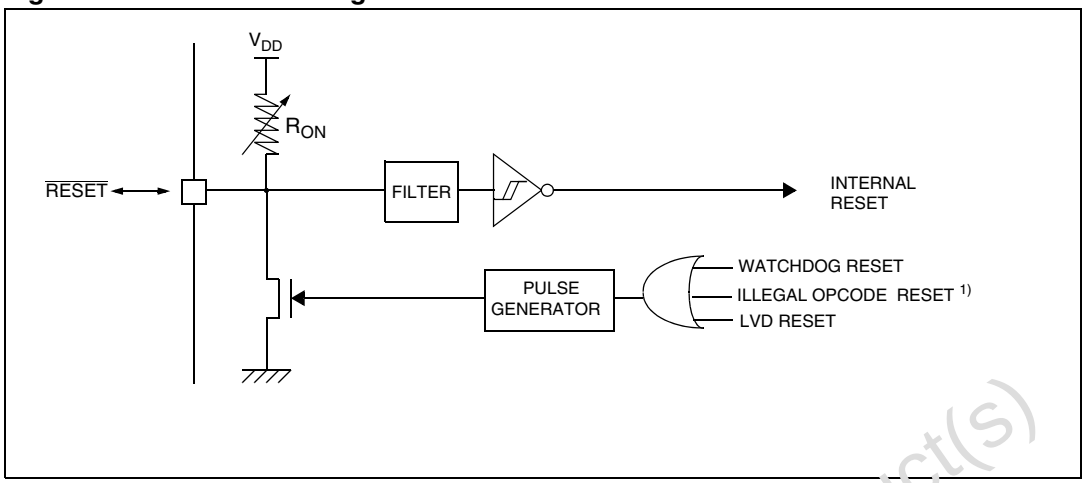


Figure 12. Reset block diagram



1. *Section 11.2.1: Illegal opcode reset* for more details on illegal opcode reset conditions

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

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

A $\overline{\text{RESET}}$ signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see *Figure 13*). This detection is asynchronous and therefore the MCU can enter reset state even in Halt mode.

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

6.4.3 External Power-on reset

If the LVD is disabled by 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.

6.4.4 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 13*.

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

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. In case of a NANDed source (as described in the I/O ports section), a low level on an I/O pin, configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

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.

Clearing an interrupt request is done by:

- Writing “0” to the corresponding bit in the status register or
- Access to the status register while the flag is set followed by a read or write of an associated register.

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

Figure 14. Interrupt processing flowchart

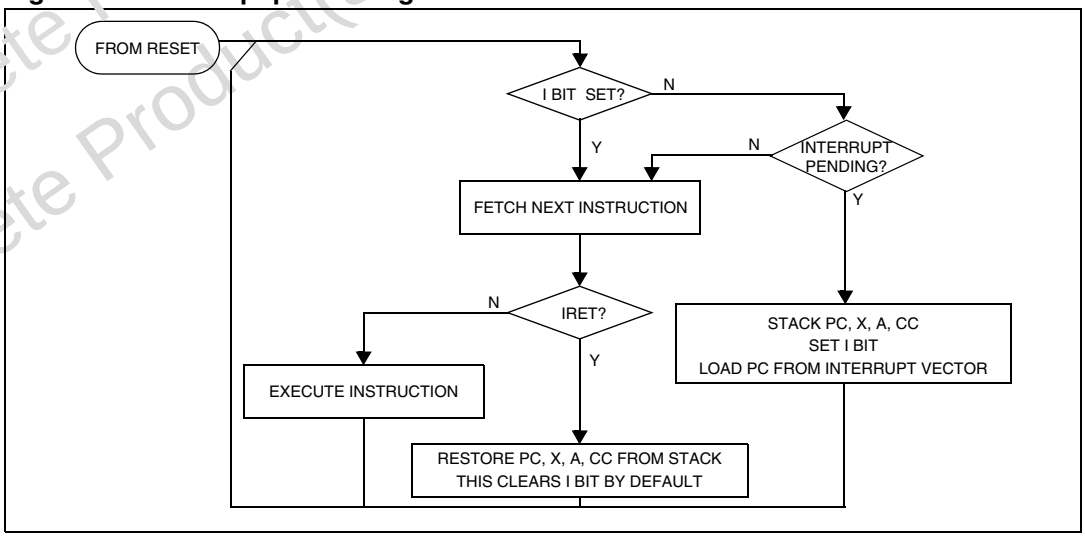
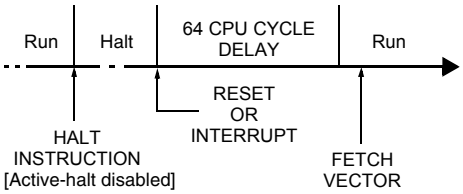
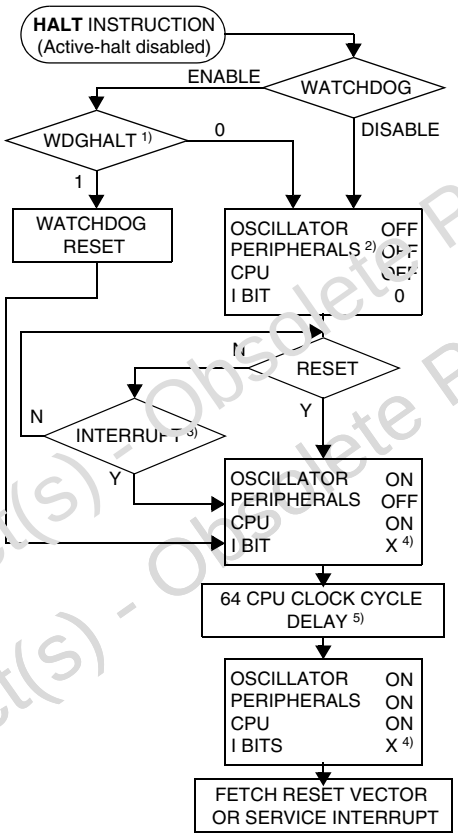


Figure 23. Halt timing overview



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

Figure 24. Halt mode flowchart



1. WDGHALT is an option bit. See option byte section for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from Halt mode (such as external interrupt). Refer to *Table 9: Interrupt mapping* 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 1 MHz (RC/8) or AWU RC before entering Halt mode.

Halt mode recommendations

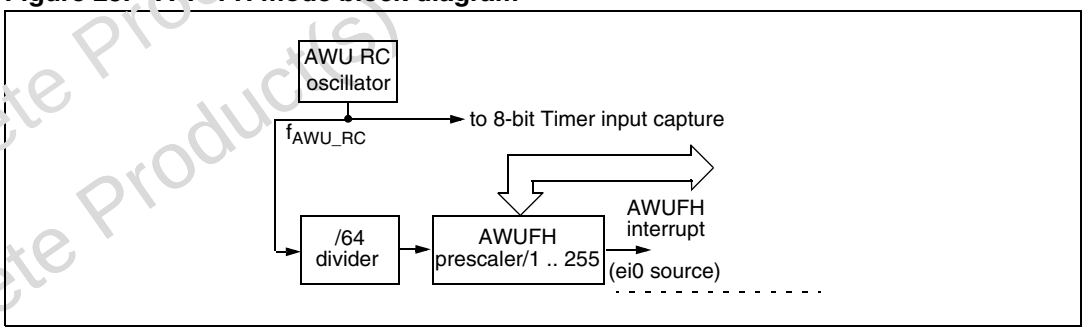
- Make sure that an external event is available to wakeup the microcontroller from Halt mode.
- When using an external interrupt to wakeup the microcontroller, reinitialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- For the same reason, reinitialize the level sensitiveness of each external interrupt as a precautionary measure.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in ROM with the value 0x8E.
- As the HALT instruction clears the I bit in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wakeup event (reset or external interrupt).

8.5 Auto-wakeup from Halt mode

Auto-wakeup from Halt (AWUFH) mode is similar to Halt mode with the addition of a specific internal RC oscillator for wakeup (Auto-wakeup from Halt oscillator) 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 realtime clock available.

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

Figure 25. 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.

11.1.5 Indirect modes (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 mode (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 mode (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 modes (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 mode (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 mode (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 30. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Instructions	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
Short instructions only	
CLR	Clear
INC, DEC	Increment/decrement
TNZ	Test negative or zero

Table 39. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes (continued)

Instructions	Function
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 modes (direct, indirect)

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

Table 40. Instructions supporting relative modes

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

The relative addressing mode consists of two submodes:

Relative mode (Direct)

The offset follows the opcode.

Relative mode (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:

Table 41. ST7 instruction set

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit operation	BSET	BRES						
Conditional bit test and branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			

Table 42. Illegal opcode detection (continued)

Mnemo	Description	Function/example	Dst	Src	H	I	N	Z	C
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 >	-	-	-	-	-	-	-
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

Table 42. Illegal opcode detection (continued)

Mnemo	Description	Function/example	Dst	Src	H	I	N	Z	C
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	-

12.7 EMC 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 2 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 100 pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results are given in the table below based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

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

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

Software recommendations

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

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

Pre-qualification 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 59. EMC characteristics

Symbol	Parameter	Conditions	Level/ class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD}=5 V, T_A=+25 °C, f_{OSC}=8 MHz, SO8 package, conforms to IEC 1000-4-2	3B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100 pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD}=5 V, T_A=+25 °C, f_{OSC}=8 MHz, SO8 package, conforms to IEC 1000-4-4	4B

12.8 I/O port pin characteristics

12.8.1 General characteristics

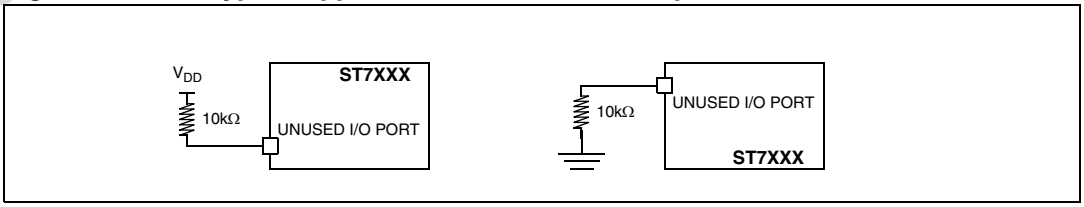
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 63. 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_L	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$				±1	µA
I_S	Static current consumption induced by each floating input pin(2)	Floating input mode			100		
R_{PU}	Weak pull-up equivalent resistor(3) (4)	$V_{IN}=V_{SS}$	V_{DD}=5 V	80	120	170	kΩ
			V_{DD}=3 V		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 48*). 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 49*).
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 48. Two typical applications with unused I/O pin



1. **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 I^2C mode unexpectedly during a reset.
2. 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.

14 Device configuration and ordering information

Each device is available for production in user programmable versions (FLASH) as well as in factory coded versions (FASTROM). Refer to *Table 79* for the full list of supported part numbers:

- ST7FLITEUSA2xx and ST7FLITEUSA5xx XFlash devices are shipped to customers with a default program memory content (FFh).
- Factory Advanced Service Technique ROM (FASTROM) versions are also available: they are factory-programmed XFlash devices.

The FASTROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer using the Option Bytes while the FASTROM devices are factory-configured.

14.1 Option bytes

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

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

14.1.1 OPTION BYTE 1

Bit 7:6 **CKSEL[1:0]** *Startup clock selection.*
This bit is used to select the startup frequency. By default, the internal RC is selected (see *Table 75: Startup clock selection*).

Bit 5 Reserved, must always be 1.

Bit 4 Reserved, must always be 0.

Bits 3:2 **LVD[1:0]** *Low Voltage Detection selection*
These option bits enable the LVD block with a selected threshold as shown in *Table 76: LVD threshold configuration*.

Bit 1 **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)

Bit 0 **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

Figure 69. Option list

```
                ST7LITEUS FASTROM MICROCONTROLLER OPTION LIST
                        (Last update: February 2009)

Customer: . . . . . .
Address:  . . . . . .
          . . . . . .
Contact:  . . . . . .
Phone No: . . . . . .
Reference/FASTROM Code*:. . . . . .
*FASTROM code name is assigned by STMicroelectronics.
FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device Type/Memory Size/Package (check only one option):
------------------|-----------------------|
FASTROM DEVICE:   |     1K FASTROM        |
------------------|-----------------------|
PDIP8             |      [ ]              |
SO8               |      [ ]              |
DFN8              |      [ ]              |


Conditioning (check only one option):
-----------------------------------------------------------------------
DIP package: [ ] Tube
SO package:  [ ] Tape & Reel  [ ] Tube
DFN package: [ ] Tape & Reel  [ ] Tray (no. ST7PLUSA5U6xxx and ST7PLUSA5U3xxx
only)

Special Marking:  [ ] No        [ ] Yes "_ _ _ _ _ _ _ _"
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count:
PDIP8/SO8/DFN8 (8 char. max)  : _ _ _ _ _ _ _ _

Temperature range     [ ] -40°C to +85°C      [ ] -40°C to +125°C

Clock Source Selection: [ ] External Clock
                        [ ] AWU RC oscillator
                        [ ] Internal RC   oscillator

Sector 0 size:          [ ] 0.5K                [ ] 1K
Readout protection:     [ ] Disabled            [ ] Enabled
FLASH Write Protection: [ ] Disabled            [ ] Enabled
LVD Reset               [ ] Disabled            [ ] Highest threshold
                                                [ ] Medium threshold
                                                [ ] Lowest threshold
Watchdog Selection:     [ ] Software Activation [ ] Hardware Activation
Watchdog Reset on Halt: [ ] Disabled            [ ] Enabled


Comments: . . . . . .
Supply Operating Range in the application:. . . . . .
Notes:. . . . . .
Date: . . . . . .
Signature:. . . . . .

Important note: Not all configurations are available. Refer to datasheet for
authorized option byte combinations.

Please download the latest version of this option list from:
http://www.st.com/mcu > downloads > ST7 microcontrollers > Option list
```