**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 | LINbusSCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 32 |
| Program Memory Size | 48KB (48K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3.8V ~ 5.5V |
| Data Converters | A/D 11x10b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st72f361j7t6 |

## 6.3 RESET SEQUENCE MANAGER (RSM)

### 6.3.1 Introduction

The reset sequence manager includes three RESET sources as shown in Figure 2:

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

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

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

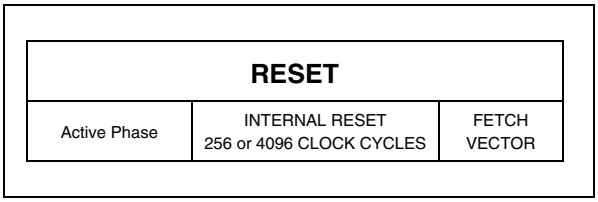
The basic RESET sequence consists of three phases as shown in Figure 1:

- Active Phase depending on the RESET source
- 256 or 4096 CPU clock cycle delay (selected by option byte)
- RESET vector fetch

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilize and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay should be selected by option byte to correspond to the stabilization time of the external oscillator used in the application.

The RESET vector fetch phase duration is two clock cycles.

**Figure 12. RESET Sequence Phases**



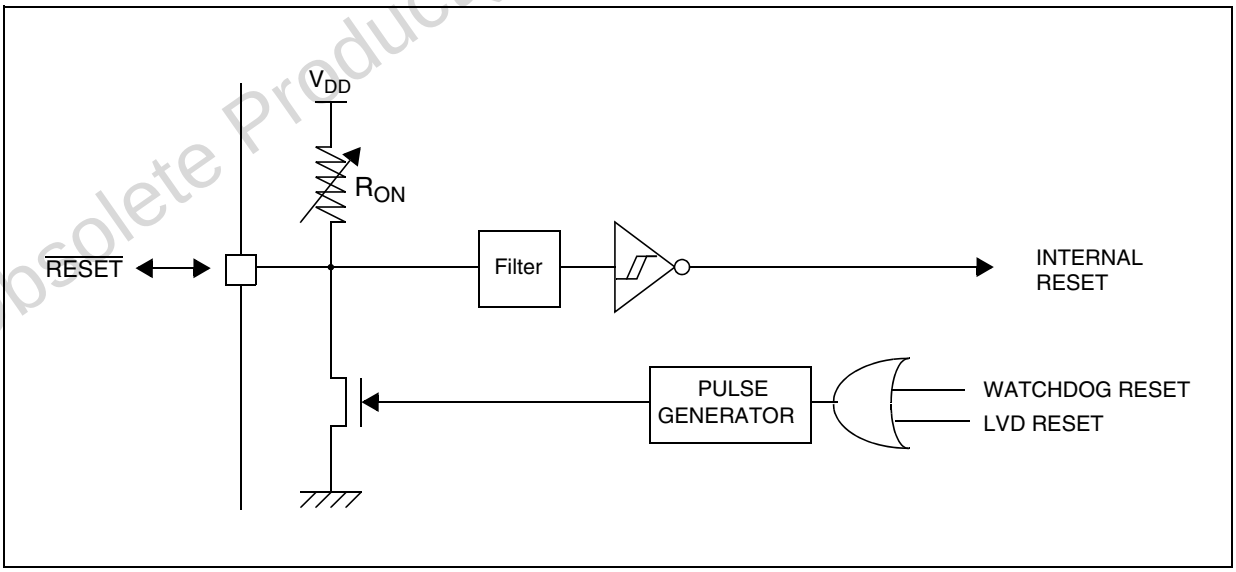

**Caution:** When the ST7 is unprogrammed or fully erased, the Flash is blank and the RESET vector is not programmed. For this reason, it is recommended to keep the RESET pin in low state until programming mode is entered, in order to avoid unwanted behavior.

### 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 Electrical Characteristic section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see Figure 3). This detection is asynchronous and therefore the MCU can enter reset state even in HALT mode.

**Figure 13. Reset Block Diagram**



**RESET SEQUENCE MANAGER** (Cont'd)

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.

### 6.3.3 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 $f_{OSC}$ frequency.

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

### 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 3.

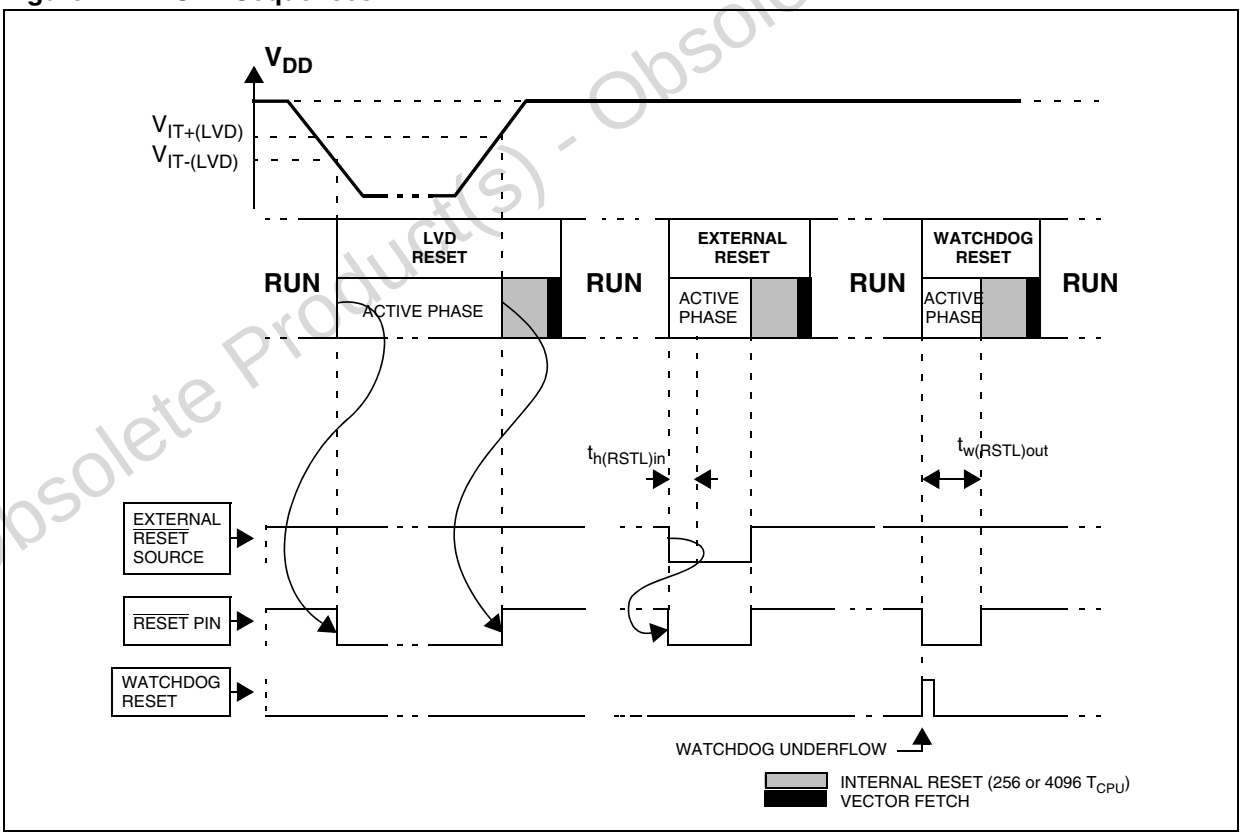
The LVD filters spikes on $V_{DD}$ larger than $t_{g(VDD)}$ to avoid parasitic resets.

### 6.3.5 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 3.

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 14. RESET Sequences**



### 6.4 SYSTEM INTEGRITY MANAGEMENT (SI)

The System Integrity Management block contains the Low Voltage Detector (LVD) and Auxiliary Voltage Detector (AVD) functions. It is managed by the SICSR register.

#### 6.4.1 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 15.

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.

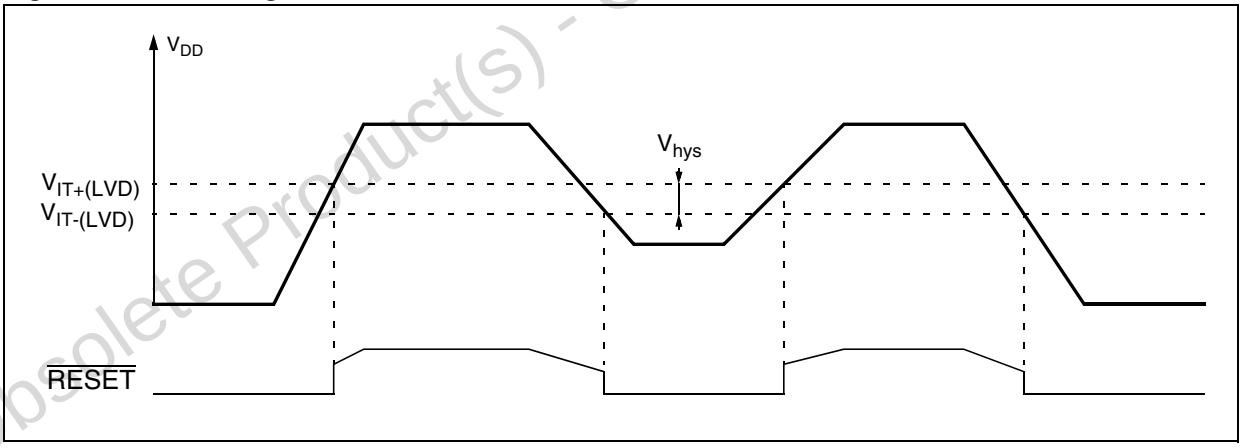
**Notes**:

The LVD allows the device to be used without any external RESET circuitry.

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

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.

**Figure 15. Low Voltage Detector vs Reset**



**I/O PORTS** (Cont'd)

### 9.6 I/O PORT REGISTER CONFIGURATIONS

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

#### 9.6.1 Standard Ports

**PB7:6, PC0, PC3, PC7:5, PD3:2, PD5, PE7:0, PF7:0**

| MODE | DDR | OR |
|---|---|---|
| floating input | 0 | 0 |
| pull-up input | 0 | 1 |
| open drain output | 1 | 0 |
| push-pull output | 1 | 1 |

#### 9.6.2 Interrupt Ports

**PA0,2,4,6; PB0,2,4; PC1; PD0,6**

(with pull-up)

| MODE | DDR | OR |
|---|---|---|
| floating input | 0 | 0 |
| pull-up interrupt input | 0 | 1 |
| open drain output | 1 | 0 |
| push-pull output | 1 | 1 |

**PA1,3,5,7; PB1,3,5; PC2; PD1,4,7**

(without pull-up)

| MODE | DDR | OR |
|---|---|---|
| floating input | 0 | 0 |
| floating interrupt input | 0 | 1 |
| open drain output | 1 | 0 |
| push-pull output | 1 | 1 |

#### 9.6.3 Pull-up Input Port

**PC4**

| MODE |
|---|
| pull-up input |

The PC4 port cannot operate as a general purpose output. If DDR = 1 it is still possible to read the port through the DR register.

**I/O PORTS** (Cont'd)

**Table 14. Port Configuration**

| Port | Pin name | Input | | Output | |
|---|---|---|---|---|---|
| | | OR = 0 | OR = 1 | OR = 0 | OR = 1 |
| Port A | PA0 | floating | pull-up interrupt (ei0) | open drain | push-pull |
| | PA1 | | floating interrupt (ei0) | | |
| | PA2 | | pull-up interrupt (ei0) | | |
| | PA3 | | floating interrupt (ei0) | | |
| | PA4 | | pull-up interrupt (ei0) | | |
| | PA5 | | floating interrupt (ei0) | | |
| | PA6 | | pull-up interrupt (ei0) | | |
| | PA7 | | floating interrupt (ei0) | | |
| Port B | PB0 | floating | pull-up interrupt (ei1) | open drain | push-pull |
| | PB1 | | floating interrupt (ei1) | | |
| | PB2 | | pull-up interrupt (ei1) | | |
| | PB3 | | floating interrupt (ei1) | | |
| | PB4 | | pull-up interrupt (ei1) | | |
| | PB5 | | floating interrupt (ei1) | | |
| Port C | PC0 | floating | pull-up | open drain | push-pull |
| | PC1 | | pull-up interrupt (ei2) | | |
| | PC2 | | floating interrupt (ei2) | | |
| | PC3 | | pull-up | | |
| | PC4 | pull-up | | N/A | |
| | PC7:5 | floating | pull-up | open drain | push-pull |
| Port D | PD0 | floating | pull-up interrupt (ei3) | open drain | push-pull |
| | PD1 | | floating interrupt (ei3) | | |
| | PD3:2 | | pull-up | | |
| | PD4 | | floating interrupt (ei3) | | |
| | PD5 | | pull-up | | |
| | PD6 | | pull-up interrupt (ei3) | | |
| | PD7 | | floating interrupt (ei3) | | |
| Port E | PE7:0 | floating (TTL) | pull-up (TTL) | open drain | push-pull |
| Port F | PF7:0 | floating (TTL) | pull-up (TTL) | open drain | push-pull |

**WINDOW WATCHDOG** (Cont'd)

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This operation must occur only when the counter value is lower than the window register value. The value to be stored in the WDGCR register must be between FFh and C0h (see Figure 2):

– Enabling the watchdog:
When Software Watchdog is selected (by option byte), the watchdog is disabled after a reset. It is enabled by setting the WDGA bit in the WDGCR register, then it cannot be disabled again except by a reset.

When Hardware Watchdog is selected (by option byte), the watchdog is always active and the WDGA bit is not used.

– Controlling the downcounter:
This downcounter is free-running: It counts down even if the watchdog is disabled. When the watchdog is enabled, the T6 bit must be set to prevent generating an immediate reset.
The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset (see Figure 2. Approximate Timeout Duration). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see Figure 3).

The window register (WDGWR) contains the high limit of the window: To prevent a reset, the downcounter must be reloaded when its value is lower than the window register value and greater than 3Fh. Figure 4 describes the window watchdog process.

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

– Watchdog Reset on Halt option
If the watchdog is activated and the watchdog reset on halt option is selected, then the HALT instruction will generate a Reset.

### 10.1.4 Using Halt Mode with the WDG

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.

**16-BIT TIMER** (Cont'd)

**CONTROL/STATUS REGISTER (CSR)**

Read/Write (bits 7:3 read only)

Reset Value: xxxx x0xx (xxh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| ICF1 | OCF1 | TOF | ICF2 | OCF2 | TIMD | 0 | 0 |

Bit 7 = **ICF1** *Input Capture Flag 1.*
0: No input capture (reset value).
1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow Flag.*
0: No timer overflow (reset value).
1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

**Note:** Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*
0: No input capture (reset value).
1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2 = **TIMD** *Timer disable.*
This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed, or the counter reset, while it is disabled.
0: Timer enabled
1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared.

**8-BIT TIMER** (Cont'd)

**Figure 63. Input Capture Block Diagram**

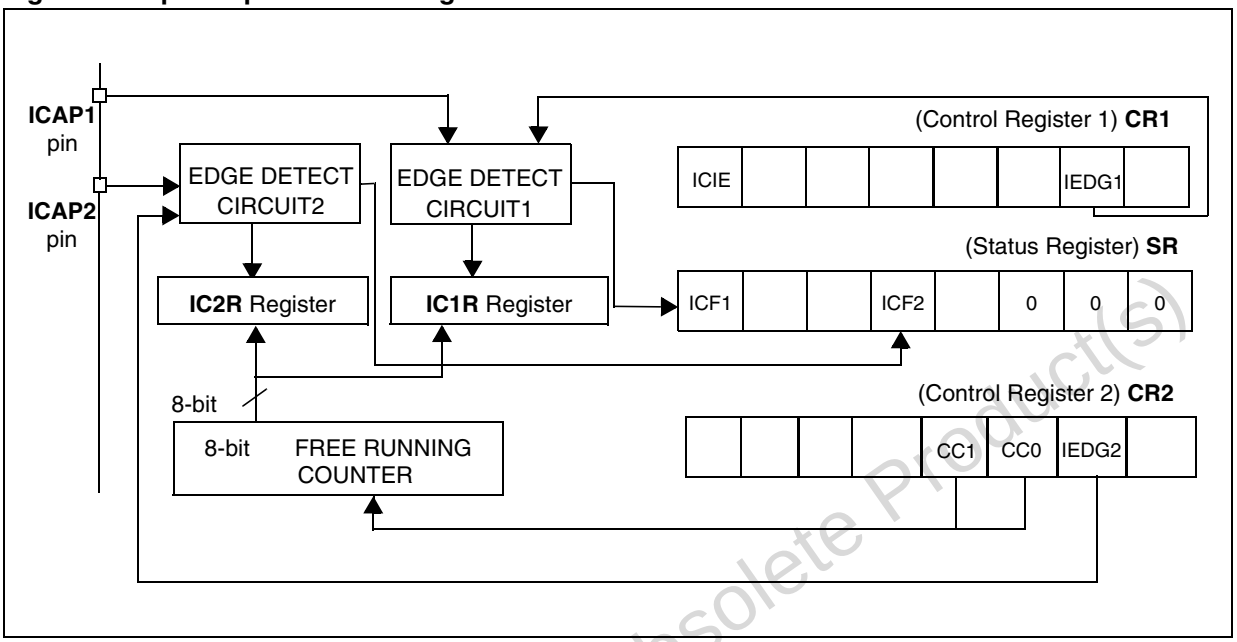
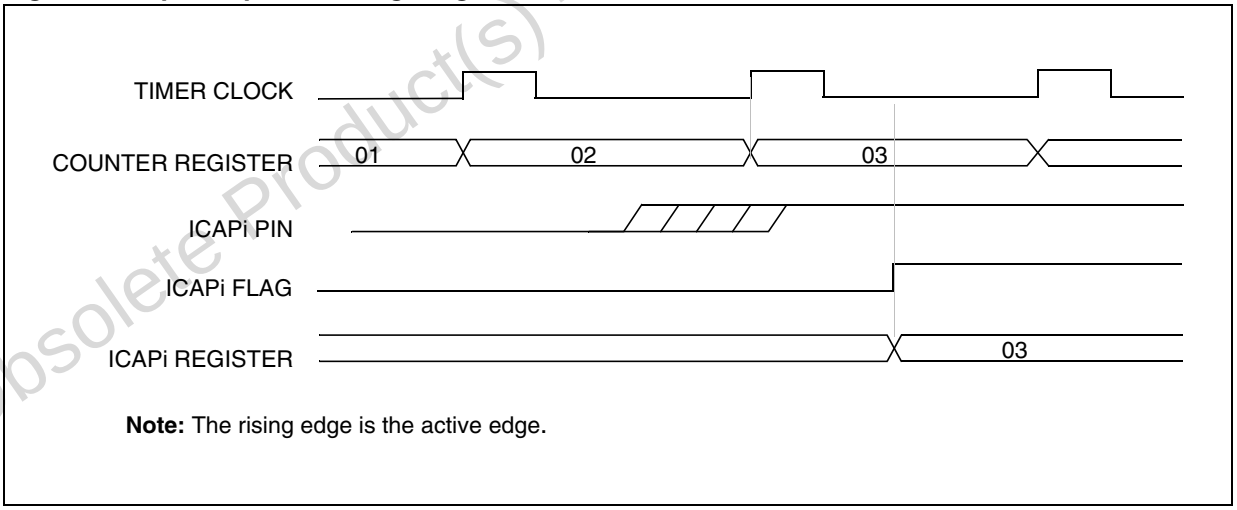



**Figure 64. Input Capture Timing Diagram**




**Note:** The rising edge is the active edge.

### 10.6.8 Register Description

**SPI CONTROL REGISTER (SPICR)**

Read/Write

Reset Value: 0000 xxxx (0xh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| SPIE | SPE | SPR2 | MSTR | CPOL | CPHA | SPR1 | SPR0 |

Bit 7 = **SPIE** *Serial Peripheral Interrupt Enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SPI interrupt is generated whenever an End of Transfer event, Master Mode Fault or Overrun error occurs (SPIF = 1, MODF = 1 or OVR = 1 in the SPICSR register)

Bit 6 = **SPE** *Serial Peripheral Output Enable*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see Section 10.6.5.1 "Master Mode Fault (MODF)"). The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins.
0: I/O pins free for general purpose I/O
1: SPI I/O pin alternate functions enabled

Bit 5 = **SPR2** *Divider Enable*
This bit is set and cleared by software and is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 20 SPI Master Mode SCK Frequency.
0: Divider by 2 enabled
1: Divider by 2 disabled

**Note:** This bit has no effect in slave mode.

Bit 4 = **MSTR** *Master Mode*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see Section 10.6.5.1 "Master Mode Fault (MODF)").
0: Slave mode
1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

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

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

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

**Note:** The slave must have the same CPOL and CPHA settings as the master.

Bits 1:0 = **SPR[1:0]** *Serial Clock Frequency*
These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode.

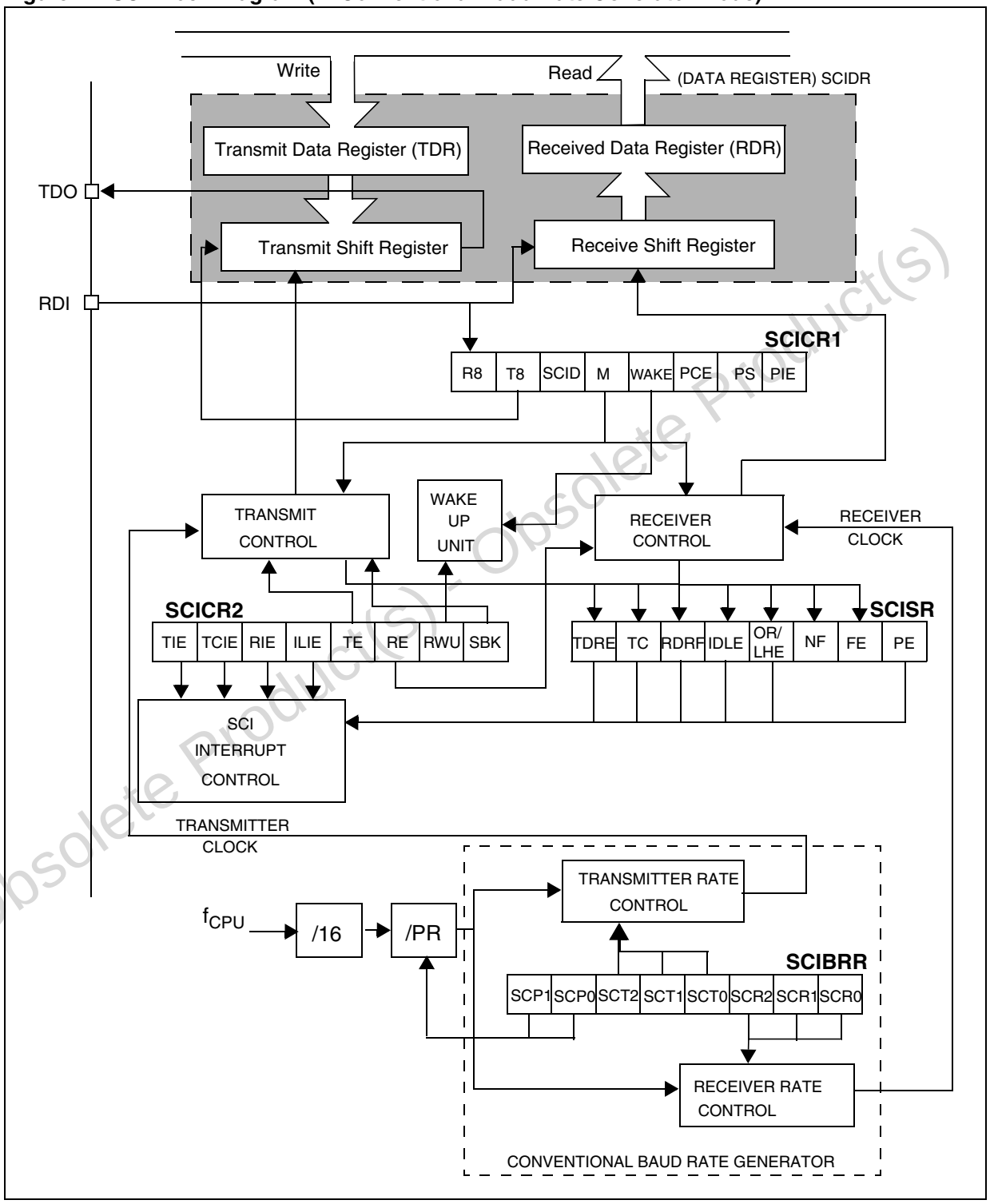
**Note:** These 2 bits have no effect in slave mode.

**Table 21. SPI Master Mode SCK Frequency**

| Serial Clock | SPR2 | SPR1 | SPR0 |
|---|---|---|---|
| $f_{CPU}/4$ | 1 | 0 | 0 |
| $f_{CPU}/8$ | 0 | | |
| $f_{CPU}/16$ | | | 1 |
| $f_{CPU}/32$ | 1 | 1 | 0 |
| $f_{CPU}/64$ | 0 | | |
| $f_{CPU}/128$ | | | 1 |

**LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode)** (cont'd)

**Figure 77. SCI Block Diagram (in Conventional Baud Rate Generator Mode)**



**LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode)** (cont'd)

#### 10.7.9.9 Error due to LIN Synch measurement

The LIN Synch Field is measured over eight bit times.

This measurement is performed using a counter clocked by the CPU clock. The edge detections are performed using the CPU clock cycle.

This leads to a precision of 2 CPU clock cycles for the measurement which lasts 16*8*LDIV clock cycles.

Consequently, this error ($D_{MEAS}$) is equal to:

$2 / (128*LDIV_{MIN})$.

$LDIV_{MIN}$ corresponds to the minimum LIN prescaler content, leading to the maximum baud rate, taking into account the maximum deviation of +/-15%.

#### 10.7.9.10 Error due to Baud Rate Quantization

The baud rate can be adjusted in steps of 1 / (16 * LDIV). The worst case occurs when the "real" baud rate is in the middle of the step.

This leads to a quantization error ($D_{QUANT}$) equal to $1 / (2*16*LDIV_{MIN})$.

#### 10.7.9.11 Impact of Clock Deviation on Maximum Baud Rate

The choice of the nominal baud rate ($LDIV_{NOM}$) will influence both the quantization error ($D_{QUANT}$) and the measurement error ($D_{MEAS}$). The worst case occurs for $LDIV_{MIN}$.

Consequently, at a given CPU frequency, the maximum possible nominal baud rate ($LPR_{MIN}$) should be chosen with respect to the maximum tolerated deviation given by the equation:

$D_{TRA} + 2 / (128*LDIV_{MIN}) + 1 / (2*16*LDIV_{MIN}) + D_{REC} + D_{TCL} < 3.75\%$

Example:

A nominal baud rate of 20Kbits/s at $T_{CPU}$ = 125ns (8 MHz) leads to $LDIV_{NOM}$ = 25d.

$LDIV_{MIN} = 25 - 0.15*25 = 21.25$

$D_{MEAS} = 2 / (128*LDIV_{MIN}) * 100 = 0.00073\%$

$D_{QUANT} = 1 / (2*16*LDIV_{MIN}) * 100 = 0.0015\%$

#### LIN Slave systems

For LIN Slave systems (the LINE and LSLV bits are set), receivers wake up by LIN Synch Break or LIN Identifier detection (depending on the LHDM bit).

#### Hot Plugging Feature for LIN Slave Nodes

In LIN Slave Mute Mode (the LINE, LSLV and RWU bits are set) it is possible to hot plug to a network during an ongoing communication flow. In this case the SCI monitors the bus on the RDI line until 11 consecutive dominant bits have been detected and discards all the other bits received.

**LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Master Only)** (Cont'd)

### 10.8.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 88 on page 153. It contains seven dedicated registers:

– Three control registers (SCICR1, SCICR2 and SCICR3)

– A status register (SCISR)

– A baud rate register (SCIBRR)

– An extended prescaler receiver register (SCIERPR)

– An extended prescaler transmitter register (SCIETPR)

Refer to the register descriptions in Section 10.7.8for the definitions of each bit.

#### 10.8.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 89).

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

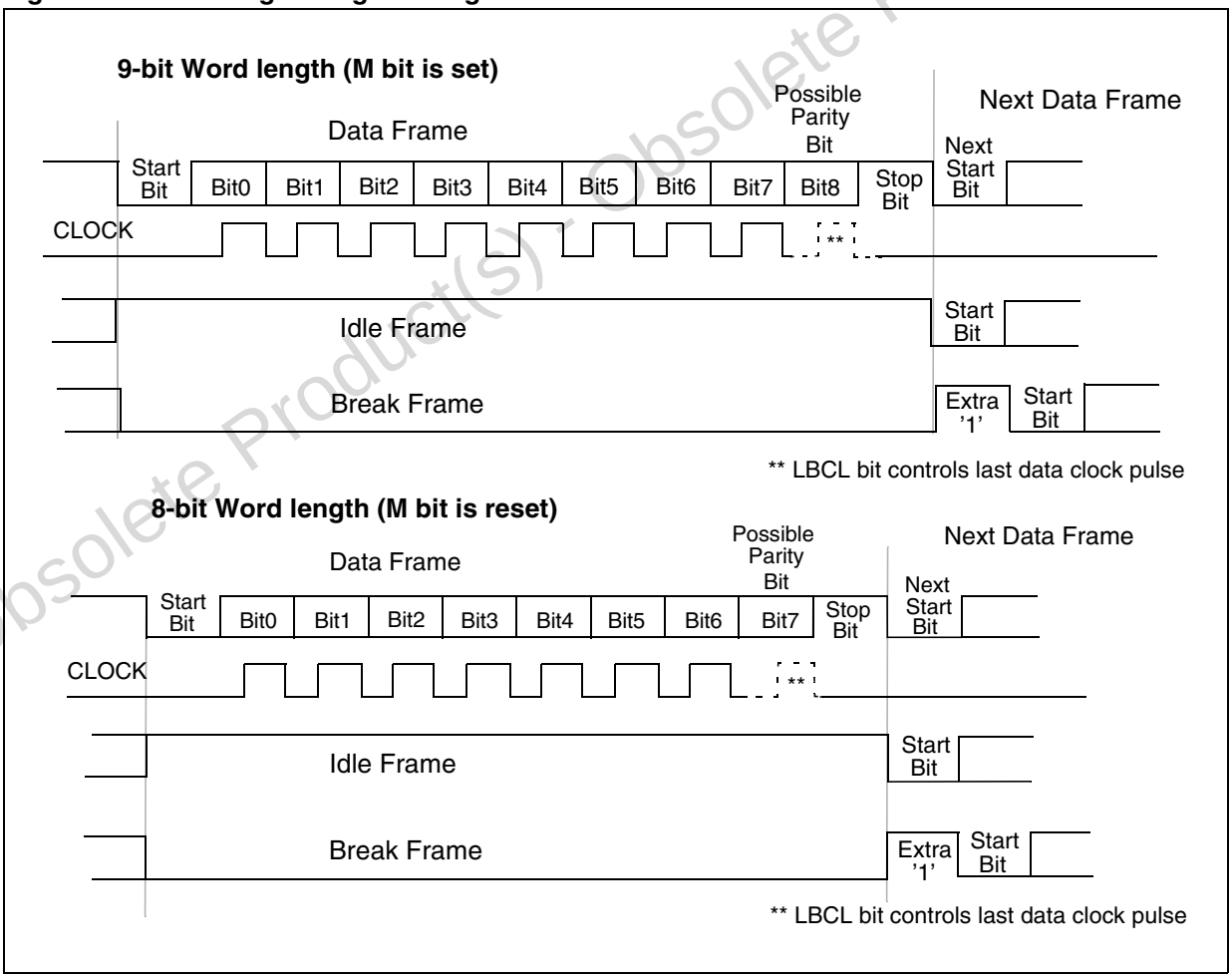
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of "1"s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving "0"s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra "1" bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

**Figure 89. Word Length Programming**



**INSTRUCTION SET OVERVIEW** (Cont'd)

### 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.

| Inherent Instruction | Function |
|---|---|
| NOP | No operation |
| TRAP | S/W Interrupt |
| WFI | Wait For Interrupt (Low Power Mode) |
| HALT | Halt Oscillator (Lowest Power Mode) |
| RET | Sub-routine Return |
| IRET | Interrupt Sub-routine Return |
| SIM | Set Interrupt Mask (level 3) |
| RIM | Reset Interrupt Mask (level 0) |
| 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.

| Immediate 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 one byte after the opcode, but only allows 00 - FF addressing space.

**Direct (long)**

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

### 11.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three submodes:

**Indexed (No Offset)**

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

**Indexed (Short)**

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

**Indexed (long)**

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

### 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.

**INSTRUCTION SET OVERVIEW** (Cont'd)

| Mnemo | Description | Function/Example | Dst | Src | I1 | H | I0 | 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 | | | | 1 | | 0 | | | |
| IRET | Interrupt routine return | Pop CC, A, X, PC | | | I1 | H | I0 | 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. INT pin = 1 | (ext. INT pin high) | | | | | | | | |
| JRIL | Jump if ext. INT pin = 0 | (ext. INT pin low) | | | | | | | | |
| JRH | Jump if H = 1 | H = 1 ? | | | | | | | | |
| JRNH | Jump if H = 0 | H = 0 ? | | | | | | | | |
| JRM | Jump if I1:0 = 11 | I1:0 = 11 ? | | | | | | | | |
| JRNM | Jump if I1:0 <> 11 | I1:0 <> 11 ? | | | | | | | | |
| 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 > | | | | | | | | |

### 12.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for $T_A$.

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IT+(LVD)}$ | Reset release threshold ($V_{DD}$ rise) | | 4.0[1)] | 4.2 | 4.5 | V |
| $V_{IT-(LVD)}$ | Reset generation threshold ($V_{DD}$ fall) | | 3.8 | 4.0 | 4.25[1)] | |
| $V_{hys(LVD)}$ | LVD voltage threshold hysteresis[1)] | $V_{IT+(LVD)}$-$V_{IT-(LVD)}$ | 150 | 200 | 250 | mV |
| $Vt_{POR}$ | $V_{DD}$ rise time rate[1)] | | 6 | | | µs/V |
| | | | | | 100 | ms/V |
| $t_{g(VDD)}$ | $V_{DD}$ glitches filtered (not detected) by LVD[1)] | Measured at $V_{IT-(LVD)}$ | | | 40 | ns |

**Notes:**

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

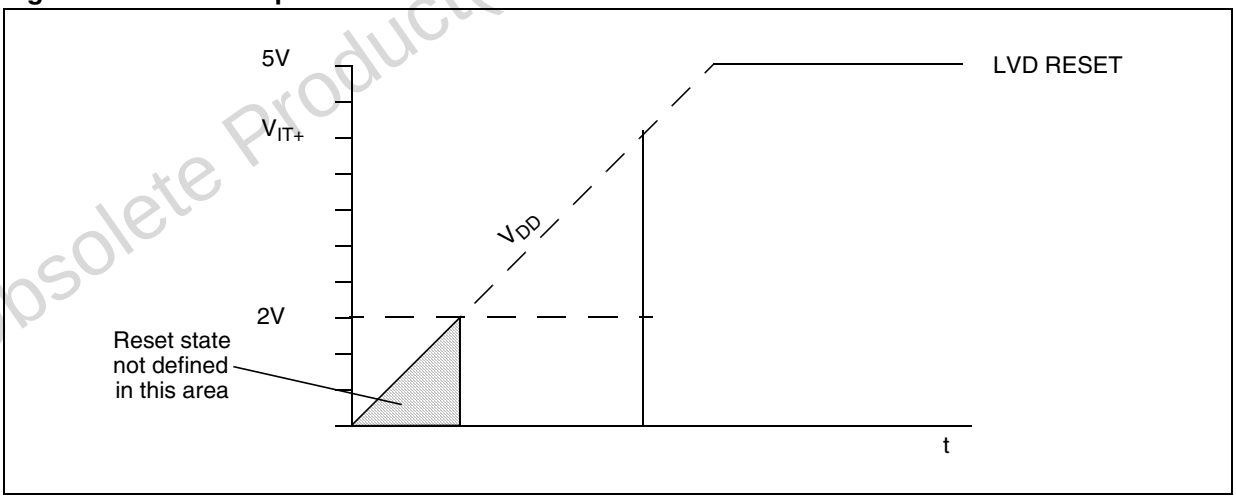
### 12.3.3 Auxiliary Voltage Detector (AVD) Thresholds

Subject to general operating conditions for $T_A$.

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IT+(AVD)}$ | 1⇒0 AVDF flag toggle threshold ($V_{DD}$ rise) | | 4.4[1)] | 4.6 | 4.9 | V |
| $V_{IT-(AVD)}$ | 0⇒1 AVDF flag toggle threshold ($V_{DD}$ fall) | | 4.2 | 4.4 | 4.65[1)] | |
| $V_{hys(AVD)}$ | AVD voltage threshold hysteresis | $V_{IT+(AVD)}$-$V_{IT-(AVD)}$ | | 250 | | mV |
| $\Delta V_{IT-}$ | Voltage drop between AVD flag set and LVD reset activated | $V_{IT-(AVD)}$-$V_{IT-(LVD)}$ | | 450 | | |

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

**Figure 98. LVD Startup Behavior**




**Note:** When the LVD is enabled, the MCU reaches its authorized operating voltage from a reset state. However, in some devices, the reset signal may be undefined until $V_{DD}$ is approximately 2V. As a consequence, the I/Os may toggle when $V_{DD}$ is below this voltage.

Because Flash write access is impossible below this voltage, the Flash memory contents will not be corrupted.

**SUPPLY CURRENT CHARACTERISTICS** (Cont'd)

### 12.4.1 Supply and Clock Managers

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To obtain the total device consumption, the two current values must be added (except for HALT mode).

| Symbol | Parameter | Conditions | Typ | Max[1)] | Unit |
|---|---|---|---|---|---|
| $I_{DD(RES)}$ | Supply current of resonator oscillator[2)3)] | | See Section 12.5.3 on page 186 | | µA |
| $I_{DD(PLL)}$ | PLL supply current | $V_{DD}$ = 5V | 360 | | |
| $I_{DD(LVD)}$ | LVD supply current | HALT mode, $V_{DD}$ = 5V | 150 | 300 | |

**Notes:**

1. Data based on characterization results, not tested in production.
2. Data based on characterization results done with the external components specified in Section 12.5.3, not tested in production.
3. As the oscillator is based on a current source, the consumption does not depend on the voltage.

## 12.8 EMC CHARACTERISTICS

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

### 12.8.1 Functional EMS (Electro Magnetic Susceptibility)

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

- **ESD**: Electro-Static Discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A Burst of Fast Transient voltage (positive and negative) is applied to $V_{DD}$ and $V_{SS}$ through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

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

#### 12.8.1.1 Designing hardened software to avoid noise problems

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

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

**Software recommendations:**

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

– Corrupted program counter

– Unexpected reset

– Critical Data corruption (control registers...)

**Prequalification trials:**

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

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

| Symbol | Parameter | Conditions | Level/ Class |
|---|---|---|---|
| $V_{FESD}$ | Voltage limits to be applied on any I/O pin to induce a functional disturbance | LQFP64, $V_{DD}$ = 5V, $T_A$ = +25°C, $f_{OSC}$ = 8 MHz, conforms to IEC 1000-4-2 | 3B |
| | | LQFP44, $V_{DD}$ = 5V, $T_A$ = +25°C, $f_{OSC}$ = 8 MHz, conforms to IEC 1000-4-2 | 2B |
| $V_{FFTB}$ | Fast transient voltage burst limits to be applied through 100pF on $V_{DD}$ and $V_{DD}$ pins to induce a functional disturbance | LQFP64, $V_{DD}$ = 5V, $T_A$ = +25°C, $f_{OSC}$ = 8 MHz, conforms to IEC 1000-4-4 | 3B |
| | | LQFP44, $V_{DD}$ = 5V, $T_A$ = +25°C, $f_{OSC}$ = 8 MHz, conforms to IEC 1000-4-4 | 2B |

## TRANSFER OF CUSTOMER CODE (Cont'd)

**ST72361 MICROCONTROLLER OPTION LIST**
(Last update: September 2006)

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

Device Type/Memory Size/Package (check only one option)

| ROM: | Package | 60K | 48K | 32K | |
|---|---|---|---|---|---|
| | LQFP64 10x10: | [ ] ST72361AR9 | [ ] ST72361AR7 | [ ] ST72361AR6 | |
| | LQFP44: | [ ] ST72361J9 | [ ] ST72361J7 | [ ] ST72361J6 | |
| | LQFP32: | [ ] ST72361K9 | [ ] ST72361K7 | [ ] ST72361K6 | |

| FASTROM: | Package | 60K | 48K | 32K | |
|---|---|---|---|---|---|
| | LQFP64 10x10: | [ ] ST72P361AR9 | [ ] ST72P361AR7 | [ ] ST72P361AR6 | |
| | LQFP44: | [ ] ST72P361J9 | [ ] ST72P361J7 | [ ] ST72P361J6 | |
| | LQFP32: | [ ] ST72P361K9 | [ ] ST72P361K7 | [ ] ST72P361K6 | |

Conditioning: [ ] Tray [ ] Tape & Reel
Special Marking: [ ] No [ ] Yes " _ _ _ _ _ _ _ _ _ " (10 char. max)
Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Temp. Range. Please refer to datasheet for specific sales conditions:

| - | Temp. Range |
|---|---|
| [ ] | -40°C to +85°C |
| [ ] | -40°C to +125°C |

Clock Source Selection: [ ] Resonator:
[ ] External Source
Oscillator/External source range:
[ ] LP: Low power (1 to 2 MHz)
[ ] MP: Medium power (2 to 4 MHz)
[ ] MS: Medium speed (4 to 8 MHz)
[ ] HS: High speed (8 to 16 MHz)
LVD [ ] Disabled [ ] Enabled
PLL[1] [ ] Disabled [ ] Enabled
Watchdog Selection [ ] Software Activation [ ] Hardware Activation
Watchdog Reset on Halt [ ] Reset [ ] No Reset

Read-out Protection [ ] Disabled [ ] Enabled

Reset Delay [ ] 256 Cycles [ ] 4096 Cycles

LINSCI2 Mapping [ ] Not available (AFIMAP[1] = 0) [ ] Mapped (AFIMAP[1] = 1)
T16_ICAP2 Mapping [ ] On PD1 (AFIMAP[0] = 0) [ ] On PC1 (AFIMAP[0] = 1)

Comments:
Supply Operating Range in the application: . . . . . .

Notes . . . . . .
Signature . . . . . .
Date . . . . . .

[1] If PLL is enabled, medium power (2 to 4 MHz range) has to be selected (MP)
Please download the latest version of this option list from:
**http://www.st.com**

# 15 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.

### 15.0.1 Evaluation Tools and Starter Kits

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

### 15.0.2 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 16K of code.

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

### 15.0.3 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 dedicated 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.

For additional ordering codes for spare parts, accessories and tools available for the ST7 (including from third party manufacturers), refer to the online product selector at www.st.com.

### 15.0.4 Order Codes for ST72361 Development Tools

**Table 37. Development Tool Order Codes for the ST72361 Family**

| MCU | In-circuit Debugger | Emulator | Programming Tool | |
|---|---|---|---|---|
| | Starter Kit with Demo Board | | In-circuit Programmer | ST7 Socket Board[6] |
| ST72361 | ST72F36X-SK/RAIS[1] | ST7MDT25-DVP3[2] | ST7-STICK[3][4]<br>STX-RLINK[5] | ST7SB25[3] |

**Notes:**

1. In-circuit programming only. In-circuit debugging is not yet supported for HDFlash devices without debug module such as the ST72F36x.
2. Requires optional connection kit. See "How to order and EMU or DVP" for connection kit ordering information in ST product and tool selection guide
3. Add suffix /EU, /UK or /US for the power supply for your region
4. Parellel port connection to PC
5. USB connection to PC
6. Socket boards complement any tool with ICC capabilities (ST7-STICK, InDART, RLINK, DVP3, EMU3, etc.)

duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

The exact sequence is:

- Disable interrupts
- Reset and Set TE (IDLE request)
- Set and Reset SBK (Break Request)
- Re-enable interrupts

**LIN mode**

If the LINE bit in the SCICR3 is set and the M bit in the SCICR1 register is reset, the LINSCI is in LIN master mode. A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 24 bits instead of 13 bits