

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	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324k6t6-tr

Table of Contents

1 INTRODUCTION	7
2 PIN DESCRIPTION	8
3 REGISTER & MEMORY MAP	13
4 FLASH PROGRAM MEMORY	17
4.1 INTRODUCTION	17
4.2 MAIN FEATURES	17
4.3 STRUCTURE	17
4.3.1 Read-out Protection	17
4.4 ICC INTERFACE	18
4.5 ICP (IN-CIRCUIT PROGRAMMING)	19
4.6 IAP (IN-APPLICATION PROGRAMMING)	19
4.7 RELATED DOCUMENTATION	19
4.7.1 Register Description	19
5 CENTRAL PROCESSING UNIT	20
5.1 INTRODUCTION	20
5.2 MAIN FEATURES	20
5.3 CPU REGISTERS	20
6 SUPPLY, RESET AND CLOCK MANAGEMENT	23
6.1 PHASE LOCKED LOOP	23
6.2 MULTI-OSCILLATOR (MO)	24
6.3 RESET SEQUENCE MANAGER (RSM)	25
6.3.1 Introduction	25
6.3.2 Asynchronous External RESET pin	25
6.3.3 External Power-On RESET	26
6.3.4 Internal Low Voltage Detector (LVD) RESET	26
6.3.5 Internal Watchdog RESET	26
6.4 SYSTEM INTEGRITY MANAGEMENT (SI)	27
6.4.1 Low Voltage Detector (LVD)	27
6.4.2 Auxiliary Voltage Detector (AVD)	28
6.4.3 Low Power Modes	29
6.4.4 Register Description	30
7 INTERRUPTS	31
7.1 INTRODUCTION	31
7.2 MASKING AND PROCESSING FLOW	31
7.3 INTERRUPTS AND LOW POWER MODES	33
7.4 CONCURRENT & NESTED MANAGEMENT	33
7.5 INTERRUPT REGISTER DESCRIPTION	34
7.6 EXTERNAL INTERRUPTS	36
7.6.1 I/O Port Interrupt Sensitivity	36
7.7 EXTERNAL INTERRUPT CONTROL REGISTER (EICR)	38
8 POWER SAVING MODES	40
8.1 INTRODUCTION	40
8.2 SLOW MODE	40
8.3 WAIT MODE	41

Table of Contents

8.4	ACTIVE-HALT AND HALT MODES	42
8.4.1	ACTIVE-HALT MODE	42
8.4.2	HALT MODE	43
9	I/O PORTS	45
9.1	INTRODUCTION	45
9.2	FUNCTIONAL DESCRIPTION	45
9.2.1	Input Modes	45
9.2.2	Output Modes	45
9.2.3	Alternate Functions	45
9.3	I/O PORT IMPLEMENTATION	48
9.4	LOW POWER MODES	48
9.5	INTERRUPTS	48
9.5.1	I/O Port Implementation	49
10	ON-CHIP PERIPHERALS	51
10.1	WATCHDOG TIMER (WDG)	51
10.1.1	Introduction	51
10.1.2	Main Features	51
10.1.3	Functional Description	51
10.1.4	How to Program the Watchdog Timeout	52
10.1.5	Low Power Modes	54
10.1.6	Hardware Watchdog Option	54
10.1.7	Using Halt Mode with the WDG (WDGHALT option)	54
10.1.8	Interrupts	54
10.1.9	Register Description	54
10.2	MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)	56
10.2.1	Programmable CPU Clock Prescaler	56
10.2.2	Clock-out Capability	56
10.2.3	Real Time Clock Timer (RTC)	56
10.2.4	Beeper	56
10.2.5	Low Power Modes	57
10.2.6	Interrupts	57
10.2.7	Register Description	57
10.3	16-BIT TIMER	59
10.3.1	Introduction	59
10.3.2	Main Features	59
10.3.3	Functional Description	59
10.3.4	Low Power Modes	71
10.3.5	Interrupts	71
10.3.6	Summary of Timer modes	71
10.3.7	Register Description	72
10.4	SERIAL PERIPHERAL INTERFACE (SPI)	79
10.4.1	Introduction	79
10.4.2	Main Features	79
10.4.3	General Description	79
10.4.4	Clock Phase and Clock Polarity	83
10.4.5	Error Flags	84
10.4.6	Low Power Modes	86

1 INTRODUCTION

The ST72324 devices are members of the ST7 microcontroller family designed for the 5V operating range.

- The 32-pin devices are designed for mid-range applications
- The 42/44-pin devices target the same range of applications requiring more than 24 I/O ports.

For a description of the differences between ST72324 and ST72324B devices refer to Section 14.2 on page 152

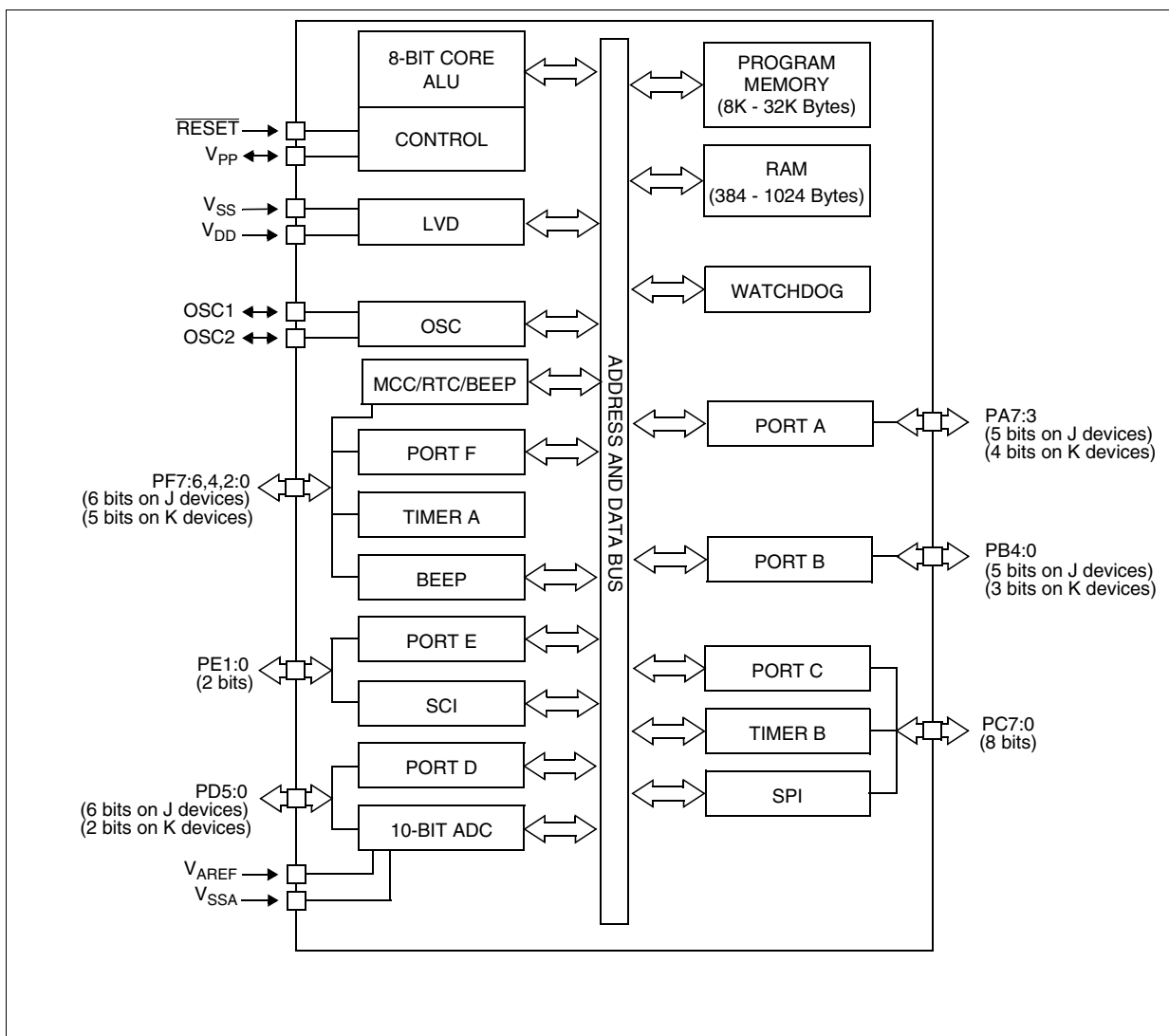
All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with FLASH program

memory.

Under software control, all devices can be placed in WAIT, SLOW, ACTIVE-HALT or HALT mode, reducing power consumption when the application is in idle or stand-by state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

Figure 1. Device Block Diagram



PIN DESCRIPTION (Cont'd)

Figure 3. 32-Pin SDIP Package Pinout

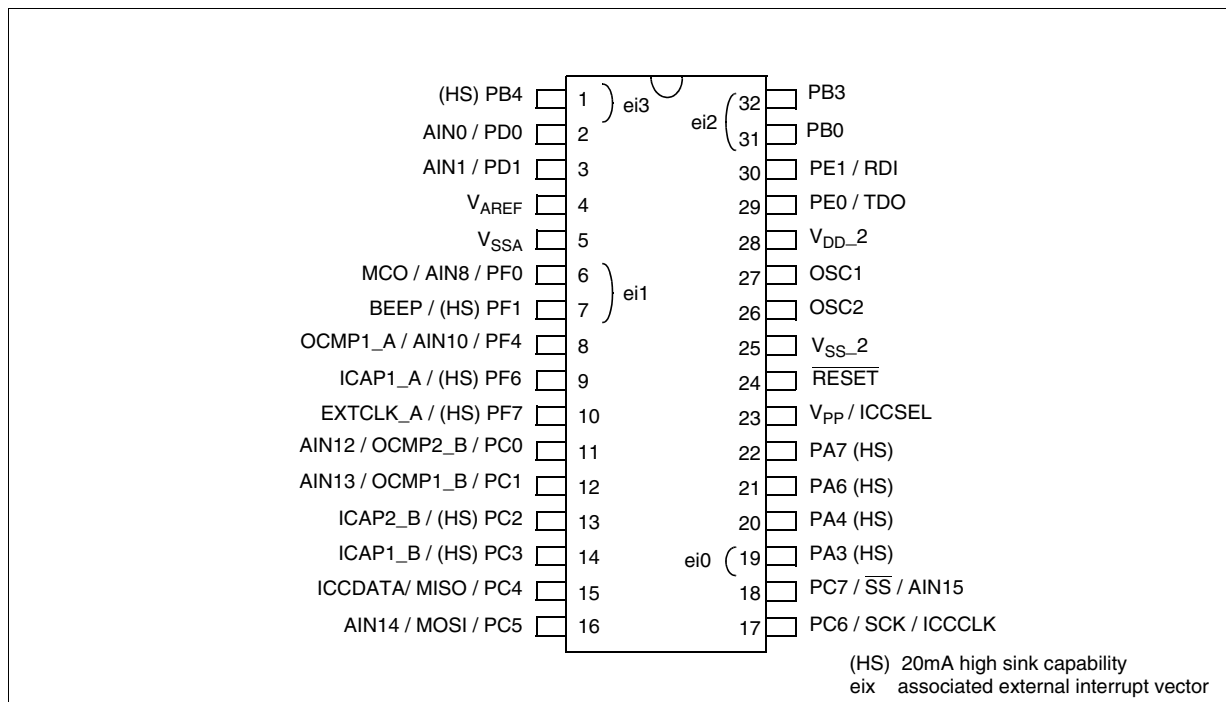
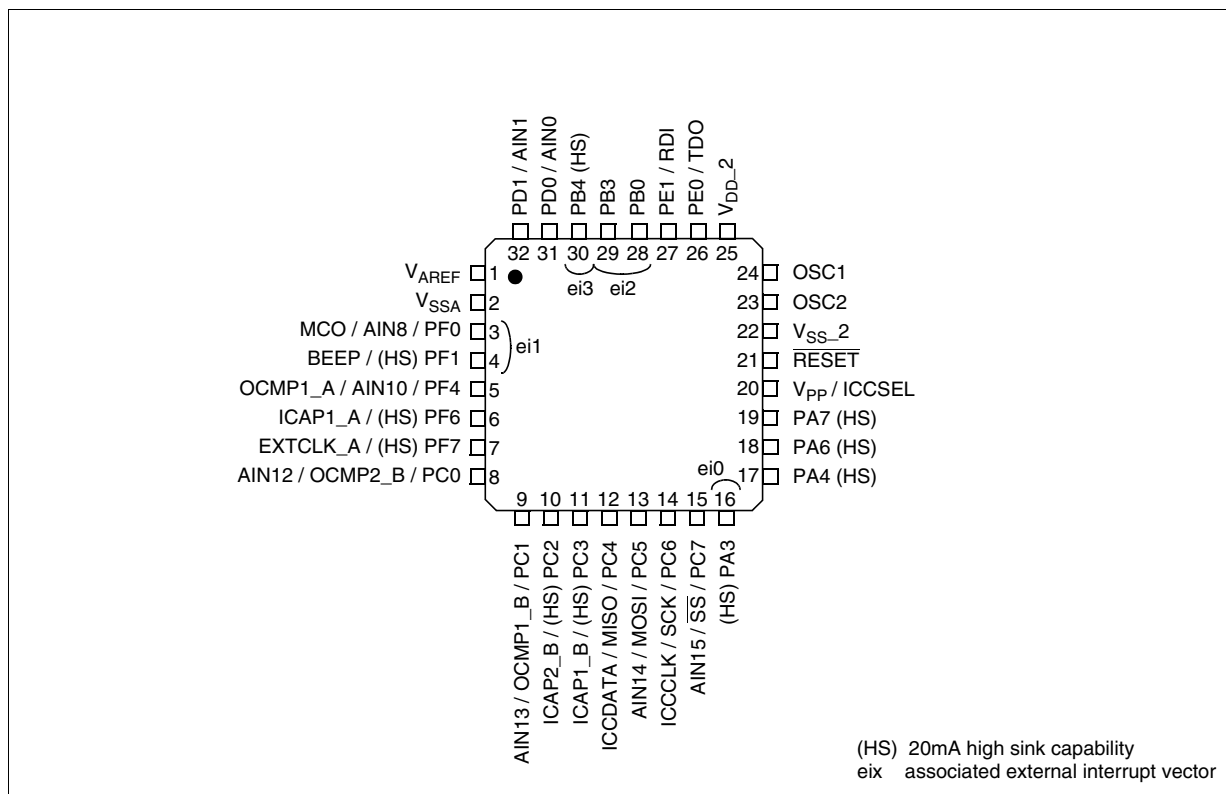


Figure 4. 32-Pin TQFP 7x7 Package Pinout



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-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

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

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in Figure 15.

The voltage threshold can be configured by option byte to be low, medium or high.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above V_{IT-} , the MCU can only be in two modes:

- under full software control
- in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Notes:

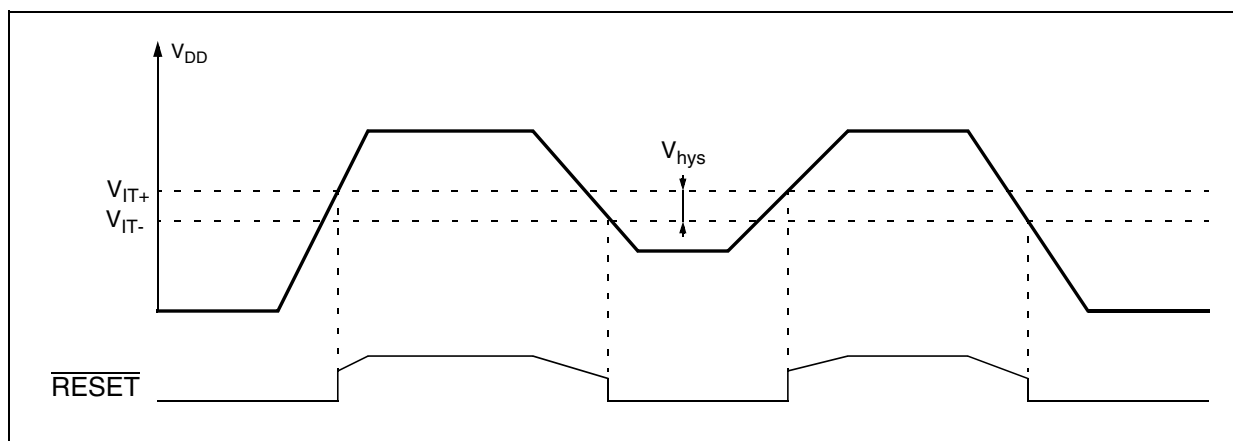
The LVD allows the device to be used without any external RESET circuitry.

If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed.

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



SYSTEM INTEGRITY MANAGEMENT (Cont'd)**6.4.3 Low Power Modes**

Mode	Description
WAIT	No effect on SI. AVD interrupt causes the device to exit from Wait mode.
HALT	The CRSR register is frozen.

6.4.3.1 Interrupts

The AVD interrupt event generates an interrupt if the AVDIE bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
AVD event	AVDF	AVDIE	Yes	No

INTERRUPTS (Cont'd)

7.5 INTERRUPT REGISTER DESCRIPTION

CPU CC REGISTER INTERRUPT BITS

Read/Write

Reset Value: 111x 1010 (xAh)

7							0
1	1	I1	H	I0	N	Z	C

Bit 5, 3 = **I1, I0** *Software Interrupt Priority*

These two bits indicate the current interrupt software priority.

Interrupt Software Priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2	↓	0	0
Level 3 (= interrupt disable*)	High	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see "Interrupt Dedicated Instruction Set" table).

***Note:** TRAP and RESET events can interrupt a level 3 program.

INTERRUPT SOFTWARE PRIORITY REGISTERS (ISPRx)

Read/Write (bit 7:4 of **ISPR3** are read only)

Reset Value: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12

These four registers contain the interrupt software priority of each interrupt vector.

– Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondance is shown in the following table.

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_0 bits*
FFF9h-FFF8h	I1_1 and I0_1 bits
...	...
FFE1h-FFE0h	I1_13 and I0_13 bits

– Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

– Level 0 can not be written (I1_x=1, I0_x=0). In this case, the previously stored value is kept. (example: previous=CFh, write=64h, result=44h)

The RESET, and TRAP vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behaviour has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

8 POWER SAVING MODES

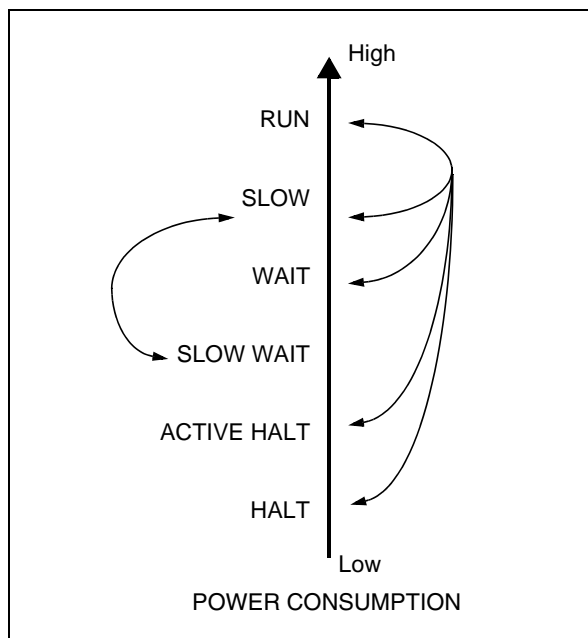
8.1 INTRODUCTION

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see Figure 22): SLOW, WAIT (SLOW WAIT), ACTIVE HALT and HALT.

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

From RUN mode, the different power saving modes 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 22. Power Saving Mode Transitions



8.2 SLOW MODE

This mode has two targets:

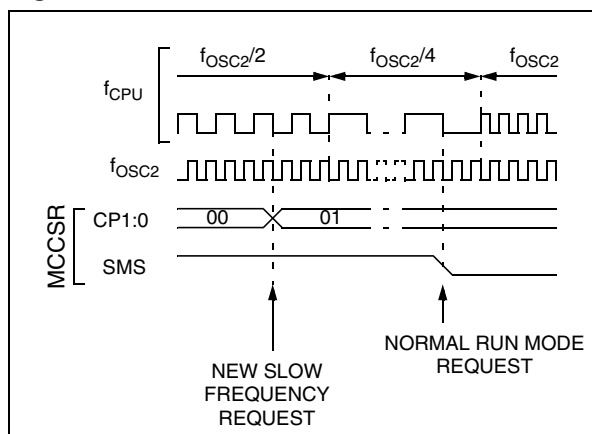
- To reduce power consumption by decreasing the internal clock in the device,
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the master clock frequency (f_{OSC2}) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency (f_{CPU}).

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

Figure 23. SLOW Mode Clock Transitions



POWER SAVING MODES (Cont'd)

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 MCC/RTC interrupt enable flag (OIE bit in MCCR register).

MCCR OIE bit	Power Saving Mode entered when HALT instruction is executed
0	HALT mode
1	ACTIVE-HALT mode

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 the OIE bit of the Main Clock Controller Status register (MCCR) is set (see Section 10.2 on page 56 for more details on the MCCR register).

The MCU can exit ACTIVE-HALT mode on reception of either an MCC/RTC interrupt, a specific interrupt (see Table 8, "Interrupt Mapping," on page 36) or a RESET. When exiting ACTIVE-HALT mode by means of an interrupt, no 256 or 4096 CPU cycle delay occurs. The CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 26).

When entering ACTIVE-HALT mode, the CC register is forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In ACTIVE-HALT mode, only the main oscillator and its associated counter (MCC/RTC) 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).

The safeguard against staying locked in ACTIVE-HALT mode is provided by the oscillator interrupt.

Note: As soon as the interrupt capability of one of the oscillators is selected (MCCR.OIE bit set), entering ACTIVE-HALT mode while the Watchdog is active does not generate a RESET.

This means that the device cannot spend more than a defined delay in this power saving mode.

CAUTION: When exiting ACTIVE-HALT mode following an interrupt, OIE bit of MCCR register must not be cleared before t_{DELAY} after the interrupt occurs ($t_{\text{DELAY}} = 256$ or $4096 t_{\text{CPU}}$ delay de-

pending on option byte). Otherwise, the ST7 enters HALT mode for the remaining t_{DELAY} period.

Figure 25. ACTIVE-HALT Timing Overview

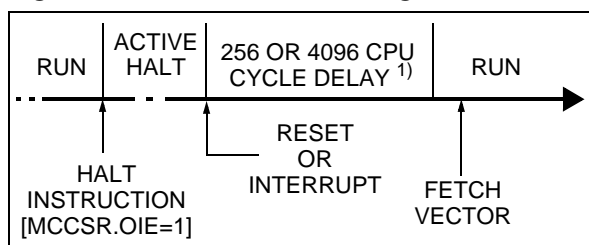
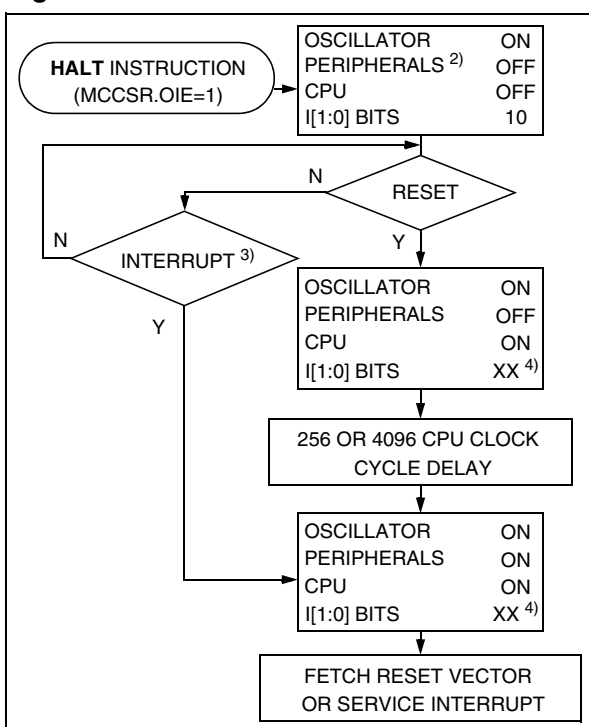


Figure 26. ACTIVE-HALT Mode Flow-chart

**Notes:**

1. This delay occurs only if the MCU exits ACTIVE-HALT mode by means of a RESET.

2. Peripheral clocked with an external clock source can still be active.

3. Only the MCC/RTC interrupt and some specific interrupts can exit the MCU from ACTIVE-HALT mode (such as external interrupt). Refer to Table 8, "Interrupt Mapping," on page 36 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 restored when the CC register is popped.

WATCHDOG TIMER (Cont'd)

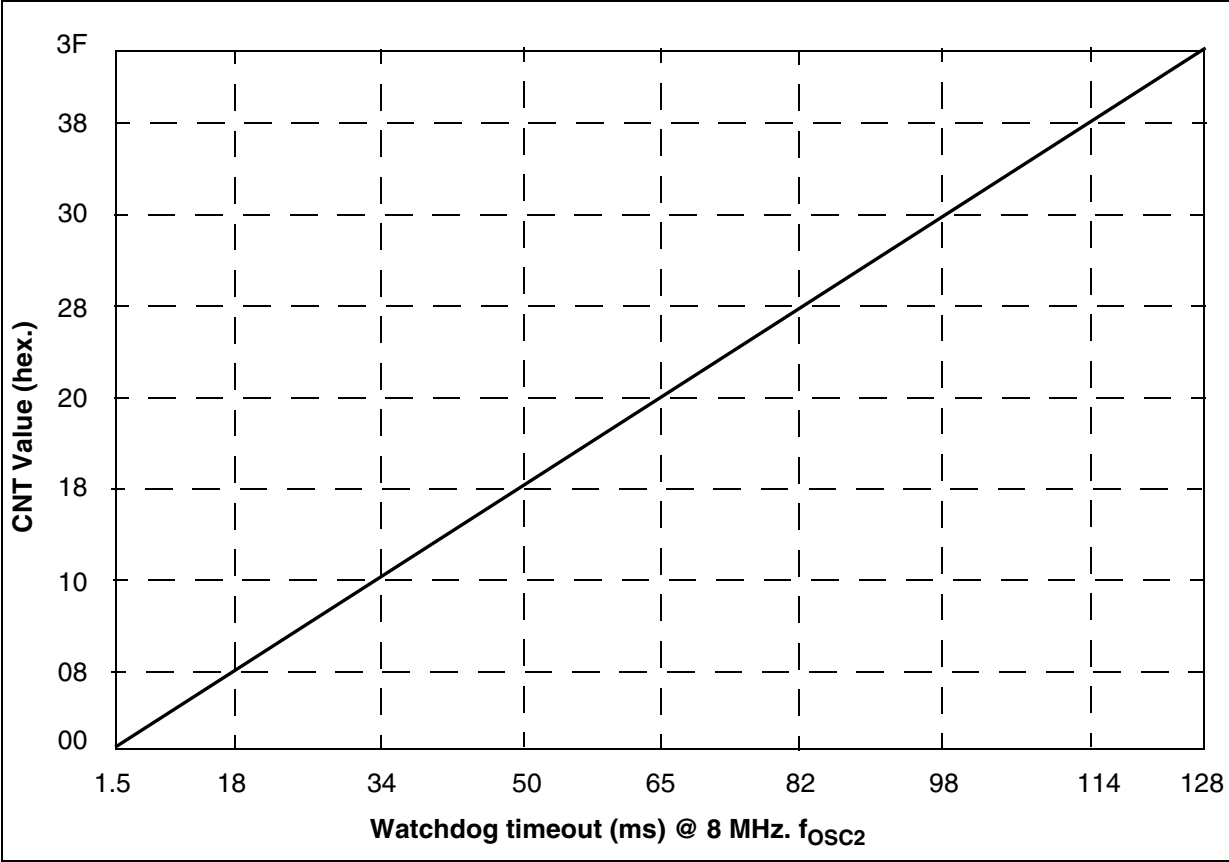
10.1.4 How to Program the Watchdog Timeout

Figure 32 shows the linear relationship between the 6-bit value to be loaded in the Watchdog Counter (CNT) and the resulting timeout duration in milliseconds. This can be used for a quick calculation without taking the timing variations into account. If

more precision is needed, use the formulae in Figure 33.

Caution: When writing to the WDGCR register, always write 1 in the T6 bit to avoid generating an immediate reset.

Figure 32. Approximate Timeout Duration



MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (Cont'd)

10.2.5 Low Power Modes

Mode	Description
WAIT	No effect on MCC/RTC peripheral. MCC/RTC interrupt cause the device to exit from WAIT mode.
ACTIVE-HALT	No effect on MCC/RTC counter (OIE bit is set), the registers are frozen. MCC/RTC interrupt cause the device to exit from ACTIVE-HALT mode.
HALT	MCC/RTC counter and registers are frozen. MCC/RTC operation resumes when the MCU is woken up by an interrupt with "exit from HALT" capability.

10.2.6 Interrupts

The MCC/RTC interrupt event generates an interrupt if the OIE bit of the MCCR register is set and the interrupt mask in the CC register is not active (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Time base overflow event	OIF	OIE	Yes	No ¹⁾

Note:

The MCC/RTC interrupt wakes up the MCU from ACTIVE-HALT mode, not from HALT mode.

10.2.7 Register Description

MCC CONTROL/STATUS REGISTER (MCCR)

Read/Write

Reset Value: 0000 0000 (00h)

7

0

MCO	CP1	CP0	SMS	TB1	TB0	OIE	OIF
-----	-----	-----	-----	-----	-----	-----	-----

Bit 7 = **MCO** Main clock out selection

This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software.

0: MCO alternate function disabled (I/O pin free for general-purpose I/O)

1: MCO alternate function enabled (f_{CPU} on I/O port)

Note: To reduce power consumption, the MCO function is not active in ACTIVE-HALT mode.

Bit 6:5 = **CP[1:0]** CPU clock prescaler

These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f_{CPU} in SLOW mode	CP1	CP0
$f_{OSC2} / 2$	0	0
$f_{OSC2} / 4$	0	1
$f_{OSC2} / 8$	1	0
$f_{OSC2} / 16$	1	1

Bit 4 = **SMS** Slow mode select

This bit is set and cleared by software.

0: Normal mode. $f_{CPU} = f_{OSC2}$

1: Slow mode. f_{CPU} is given by CP1, CP0

See Section 8.2 SLOW MODE and Section 10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC) for more details.

Bit 3:2 = **TB[1:0]** Time base control

These bits select the programmable divider time base. They are set and cleared by software.

Counter Prescaler	Time Base		TB1	TB0
	$f_{OSC2} = 4\text{MHz}$	$f_{OSC2} = 8\text{MHz}$		
16000	4ms	2ms	0	0
32000	8ms	4ms	0	1
80000	20ms	10ms	1	0
200000	50ms	25ms	1	1

A modification of the time base is taken into account at the end of the current period (previously set) to avoid an unwanted time shift. This allows to use this time base as a real time clock.

Bit 1 = **OIE** Oscillator interrupt enable

This bit set and cleared by software.

0: Oscillator interrupt disabled

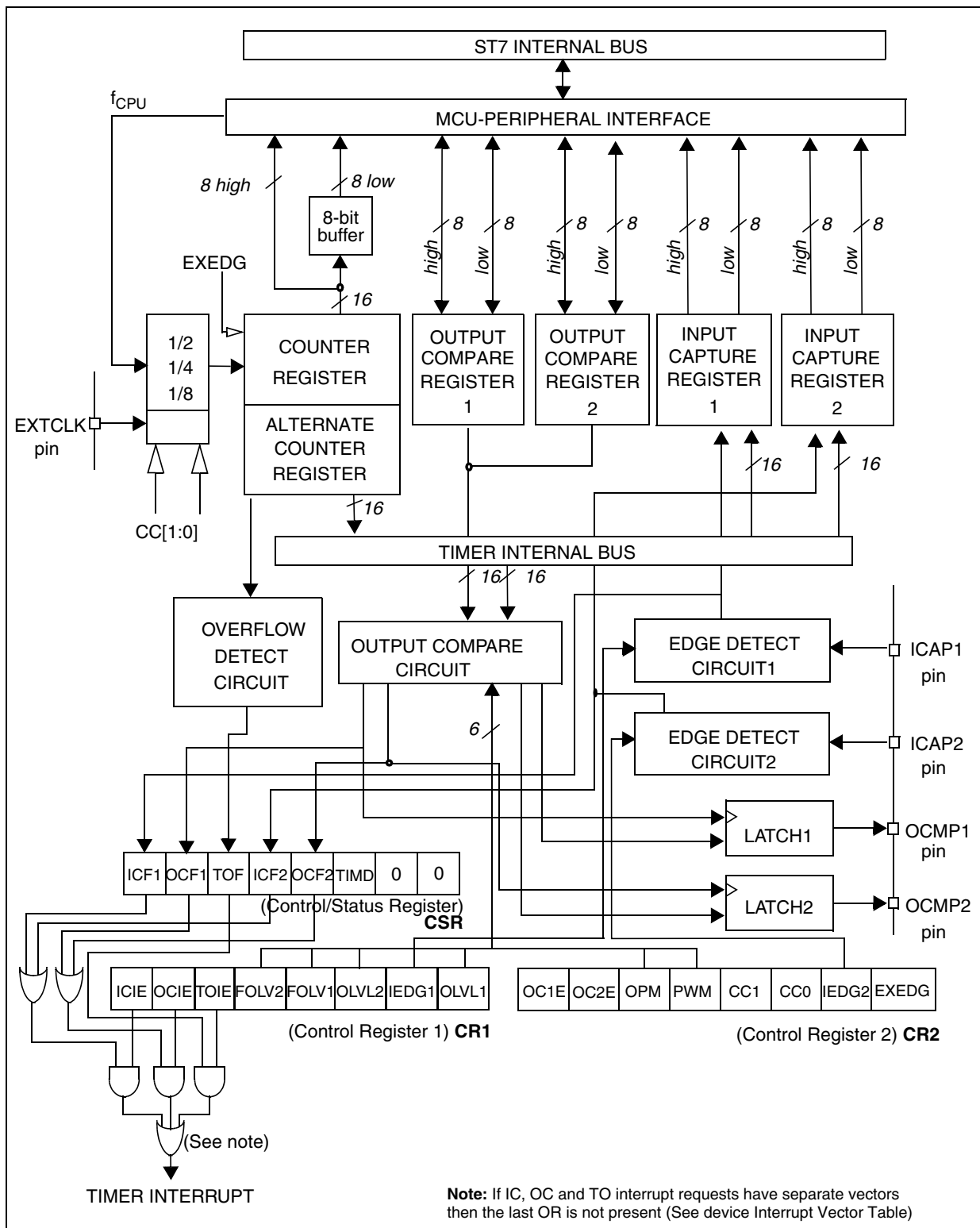
1: Oscillator interrupt enabled

This interrupt can be used to exit from ACTIVE-HALT mode.

When this bit is set, calling the ST7 software HALT instruction enters the ACTIVE-HALT power saving mode.

16-BIT TIMER (Cont'd)

Figure 35. Timer Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)**10.4.3.3 Master Mode Operation**

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

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

To operate the SPI in master mode, perform the following steps in order (if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account):

1. Write to the SPICR register:
 - Select the clock frequency by configuring the SPR[2:0] bits.
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 50 shows the four possible configurations.

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

2. Write to the SPICSR register:
 - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.

3. Write to the SPICR register:
 - Set the MSTR and SPE bits

Note: MSTR and SPE bits remain set only if SS is high).

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

10.4.3.4 Master Mode Transmit Sequence

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

When data transfer is complete:

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

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

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

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

10.4.3.5 Slave Mode Operation

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

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:

- Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 50).

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

- Manage the \overline{SS} pin as described in Section 10.4.3.2 and Figure 48. If CPHA=1 \overline{SS} must be held low continuously. If CPHA=0 \overline{SS} must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.

2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

10.4.3.6 Slave Mode Transmit Sequence

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

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

When data transfer is complete:

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

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

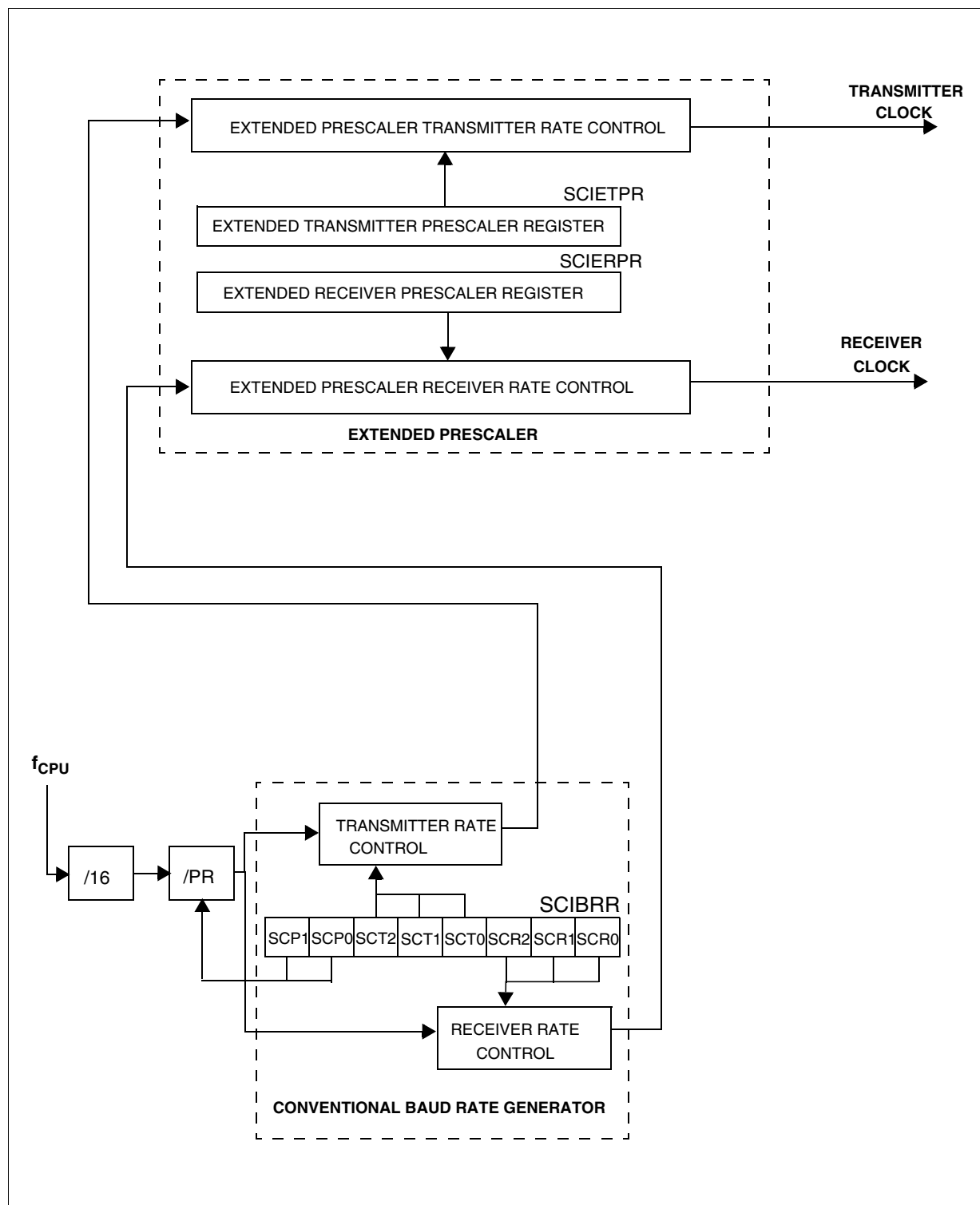
1. An access to the SPICSR register while the SPIF bit is set.
2. A write or a read to the SPIDR register.

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

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 10.4.5.2).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 55. SCI Baud Rate and Extended Prescaler Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)**10.5.4.7 Parity Control**

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 1.

Table 20. Frame Formats

M bit	PCE bit	SCI frame
0	0	SB 8 bit data STB
0	1	SB 7-bit data PB STB
1	0	SB 9-bit data STB
1	1	SB 8-bit data PB STB

Legend: SB = Start Bit, STB = Stop Bit,
PB = Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: the parity bit is calculated to obtain an even number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 0 if even parity is selected (PS bit = 0).

Odd parity: the parity bit is calculated to obtain an odd number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an

even number of “1s” if even parity is selected (PS = 0) or an odd number of “1s” if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.5.4.8 SCI Clock Tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: If the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value is “1”, but the Noise Flag bit is set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 Kbaud (bit length is 64µs), then the 8th, 9th and 10th samples are at 28µs, 32µs and 36µs respectively (the first sample starting ideally at 0µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4µs. This means the entire bit length must be at least 40µs (36µs for the 10th sample + 4µs for synchronization with the internal sampling clock).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)**10.5.7 Register Description****STATUS REGISTER (SCISR)**

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE

Bit 7 = TDRE *Transmit data register empty.*

This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register).

0: Data is not transferred to the shift register

1: Data is transferred to the shift register

Note: Data is not transferred to the shift register unless the TDRE bit is cleared.

Bit 6 = TC *Transmission complete.*

This bit is set by hardware when transmission of a frame containing Data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register).

0: Transmission is not complete

1: Transmission is complete

Note: TC is not set after the transmission of a Preamble or a Break.

Bit 5 = RDRF *Received data ready flag.*

This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: Data is not received

1: Received data is ready to be read

Bit 4 = IDLE *Idle line detect.*

This bit is set by hardware when an Idle Line is detected. An interrupt is generated if the ILIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Idle Line is detected

1: Idle Line is detected

Note: The IDLE bit is not set again until the RDRF bit has been set itself (that is, a new idle line occurs).

Bit 3 = OR *Overrun error.*

This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF = 1. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Overrun error

1: Overrun error is detected

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

Bit 2 = NF *Noise flag.*

This bit is set by hardware when noise is detected on a received frame. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No noise is detected

1: Noise is detected

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

Bit 1 = FE *Framing error.*

This bit is set by hardware when a de-synchronization, excessive noise or a break character is detected. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Framing error is detected

1: Framing error or break character is detected

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

Bit 0 = PE *Parity error.*

This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE = 1 in the SCICR1 register.

0: No parity error

1: Parity error

10-BIT A/D CONVERTER (Cont'd)**Table 23. ADC Register Map and Reset Values**

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	ADCCSR Reset Value	EOC 0	SPEED 0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0
0071h	ADCDRH Reset Value	D9 0	D8 0	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0
0072h	ADCDRL Reset Value	0	0	0	0	0	0	D1 0	D0 0

EMC CHARACTERISTICS (Cont'd)**12.8.2 Electro Magnetic Interference (EMI)**

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

Symbol	Parameter	Conditions	Device/ Package	Monitored Frequency Band	Max vs. [f_{osc}/f_{CPU}]		Unit
					8/4MHz	16/8MHz	
S_{EMI}	Peak level	$V_{DD}=5V$, $T_A=+25^{\circ}C$ conforming to SAE J 1752/3	8/16K Flash/ TQFP44	0.1MHz to 30MHz	12	18	dB μ V
				30MHz to 130MHz	19	25	
				130MHz to 1GHz	15	22	
				SAE EMI Level	3	3.5	
			32K Flash/TQFP44	0.1MHz to 30MHz	20	21	dB μ V
				30MHz to 130MHz	26	31	
				130MHz to 1GHz	22	28	
				SAE EMI Level	3.5	4.0	
			Flash/TQFP32	0.1MHz to 30MHz	25	27	dB μ V
				30MHz to 130MHz	30	36	
				130MHz to 1GHz	18	23	
				SAE EMI Level	3.0	3.5	

Notes:

1. Data based on characterization results, not tested in production.
2. Refer to Application Note AN1709 for data on other package types.

12.12 COMMUNICATION INTERFACE CHARACTERISTICS

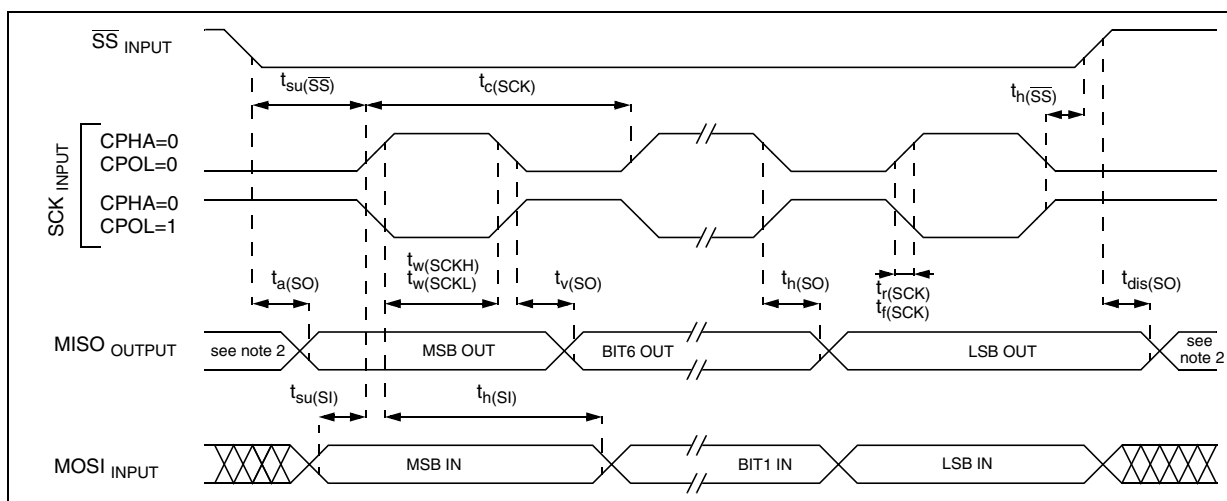
12.12.1 SPI - Serial Peripheral Interface

Subject to general operating conditions for V_{DD} , f_{CPU} , and T_A unless otherwise specified. Data based on design simulation and/or characterisation results, not tested in production.

When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration. Refer to I/O port characteristics for more details on the input/output alternate function characteristics (SS, SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master $f_{CPU}=8MHz$	$f_{CPU}/128$ 0.0625	$f_{CPU}/4$ 2	MHz
		Slave $f_{CPU}=8MHz$	0	$f_{CPU}/2$ 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su}(\overline{SS})$	\overline{SS} setup time	Slave	120		ns
$t_{h(\overline{SS})}$	\overline{SS} hold time	Slave	120		
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master Slave	100 100		
$t_a(SO)$	Data output access time	Slave	0	120	
$t_{dis(SO)}$	Data output disable time	Slave		240	
$t_{v(SO)}$	Data output valid time	Slave (after enable edge)		90	
$t_{h(SO)}$	Data output hold time		0		
$t_{v(MO)}$	Data output valid time	Master (before capture edge)	0.25		t_{CPU}
$t_{h(MO)}$	Data output hold time		0.25		

Figure 80. SPI Slave Timing Diagram with CPHA=0¹⁾



Notes:

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

ST72324 DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)**OPTION BYTE 1**OPT7= **PKG1** *Pin package selection bit*

This option bit selects the package.

Version	Selected Package	PKG1
J	TQFP44 / SDIP42	1
K	TQFP32 / SDIP32	0

Note: On the chip, each I/O port has 8 pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.

OPT6 = **RSTC** *RESET clock cycle selection*

This option bit selects the number of CPU cycles applied during the RESET phase and when exiting HALT mode. For resonator oscillators, it is advised to select 4096 due to the long crystal stabilization time.

0: Reset phase with 4096 CPU cycles

1: Reset phase with 256 CPU cycles

OPT5:4 = **OSCTYPE**[1:0] *Oscillator Type*

These option bits select the ST7 main clock source type.

Clock Source	OSCTYPE	
	1	0
Resonator Oscillator	0	0
Reserved	0	1
Internal RC Oscillator	1	0
External Source	1	1

OPT3:1 = **OSCRANGE**[2:0] *Oscillator range*

When the resonator oscillator type is selected,

these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. Otherwise, these bits are used to select the normal operating frequency range.

Typ. Freq. Range		OSCRANGE		
		2	1	0
LP	1~2MHz	0	0	0
MP	2~4MHz	0	0	1
MS	4~8MHz	0	1	0
HS	8~16MHz	0	1	1

OPT0 = **PLL OFF** *PLL activation*

This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator. The PLL is guaranteed only with an input frequency between 2 and 4MHz.

0: PLL x2 enabled

1: PLL x2 disabled

CAUTION: the PLL can be enabled only if the "OSC RANGE" (OPT3:1) bits are configured to "MP - 2~4MHz". Otherwise, the device functionality is not guaranteed.