



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





# ST7LITE0xY0, ST7LITESxY0

8-bit microcontroller with single voltage  
Flash memory, data EEPROM, ADC, timers, SPI

## ■ Memories

- 1K or 1.5 Kbytes single voltage Flash Program memory with read-out protection, In-Circuit and In-Application Programming (ICP and IAP). 10 K write/erase cycles guaranteed, data retention: 20 years at 55 °C.
- 128 bytes RAM.
- 128 bytes data EEPROM with read-out protection. 300K write/erase cycles guaranteed, data retention: 20 years at 55 °C.

## ■ Clock, Reset and Supply Management

- 3-level low voltage supervisor (LVD) and auxiliary voltage detector (AVD) for safe power-on/off procedures
- Clock sources: internal 1MHz RC 1% oscillator or external clock
- PLL x4 or x8 for 4 or 8 MHz internal clock
- Four Power Saving Modes: Halt, Active-Halt, Wait and Slow

## ■ Interrupt Management

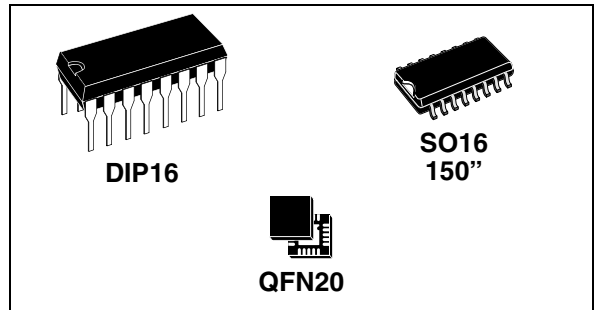
- 10 interrupt vectors plus TRAP and RESET
- 4 external interrupt lines (on 4 vectors)

## ■ I/O Ports

- 13 multifunctional bidirectional I/O lines
- 9 alternate function lines
- 6 high sink outputs

## ■ 2 Timers

- One 8-bit Lite Timer (LT) with prescaler including: watchdog, 1 realtime base and 1 input capture.



- One 12-bit Auto-reload Timer (AT) with output compare function and PWM

## ■ 1 Communication Interface

- SPI synchronous serial interface

## ■ A/D Converter

- 8-bit resolution for 0 to  $V_{DD}$
- Fixed gain Op-amp for 11-bit resolution in 0 to 250 mV range (@ 5V  $V_{DD}$ )
- 5 input channels

## ■ Instruction Set

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

## ■ Development Tools

- Full hardware/software development package

## Device Summary

Features	ST7LITESxY0 (ST7SUPERLITE)		ST7LITE0xY0		
	ST7LITES2Y0	ST7LITES5Y0	ST7LITE02Y0	ST7LITE05Y0	ST7LITE09Y0
Program memory - bytes	1K	1K	1.5K	1.5K	1.5K
RAM (stack) - bytes	128 (64)	128 (64)	128 (64)	128 (64)	128 (64)
Data EEPROM - bytes	-	-	-	-	128
Peripherals	LT Timer w/ Wdg, AT Timer w/ 1 PWM, SPI	LT Timer w/ Wdg, AT Timer w/ 1 PWM, SPI, 8-bit ADC	LT Timer w/ Wdg, AT Timer w/ 1 PWM, SPI	LT Timer w/ Wdg, AT Timer w/ 1 PWM, SPI, 8-bit ADC w/ Op-Amp	
Operating Supply	2.4V to 5.5V				
CPU Frequency	1MHz RC 1% + PLLx4/8MHz				
Operating Temperature	-40°C to +85°C				
Packages	SO16 150", DIP16, QFN20				

---

# Table of Contents

---

<b>ST7LITE0xY0, ST7LITESxY0</b> .....	<b>1</b>
<b>1 DESCRIPTION</b> .....	<b>5</b>
<b>2 PIN DESCRIPTION</b> .....	<b>6</b>
<b>3 REGISTER &amp; MEMORY MAP</b> .....	<b>9</b>
<b>4 FLASH PROGRAM MEMORY</b> .....	<b>13</b>
4.1 INTRODUCTION .....	13
4.2 MAIN FEATURES .....	13
4.3 PROGRAMMING MODES .....	13
4.4 ICC INTERFACE .....	14
4.5 MEMORY PROTECTION .....	15
4.6 RELATED DOCUMENTATION .....	15
4.7 REGISTER DESCRIPTION .....	15
<b>5 DATA EEPROM</b> .....	<b>16</b>
5.1 INTRODUCTION .....	16
5.2 MAIN FEATURES .....	16
5.3 MEMORY ACCESS .....	17
5.4 POWER SAVING MODES .....	19
5.5 ACCESS ERROR HANDLING .....	19
5.6 DATA EEPROM READ-OUT PROTECTION .....	19
5.7 REGISTER DESCRIPTION .....	20
<b>6 CENTRAL PROCESSING UNIT</b> .....	<b>21</b>
6.1 INTRODUCTION .....	21
6.2 MAIN FEATURES .....	21
6.3 CPU REGISTERS .....	21
<b>7 SUPPLY, RESET AND CLOCK MANAGEMENT</b> .....	<b>24</b>
7.1 INTERNAL RC OSCILLATOR ADJUSTMENT .....	24
7.2 PHASE LOCKED LOOP .....	24
7.3 REGISTER DESCRIPTION .....	25
7.4 RESET SEQUENCE MANAGER (RSM) .....	27
<b>8 INTERRUPTS</b> .....	<b>29</b>
8.1 NON MASKABLE SOFTWARE INTERRUPT .....	29
8.2 EXTERNAL INTERRUPTS .....	29
8.3 PERIPHERAL INTERRUPTS .....	29
8.4 SYSTEM INTEGRITY MANAGEMENT (SI) .....	32
<b>9 POWER SAVING MODES</b> .....	<b>37</b>
9.1 INTRODUCTION .....	37
9.2 SLOW MODE .....	37
9.3 WAIT MODE .....	38
9.4 ACTIVE-HALT AND HALT MODES .....	39

---

# Table of Contents

---

<b>10 I/O PORTS</b> .....	<b>42</b>
10.1 INTRODUCTION .....	42
10.2 FUNCTIONAL DESCRIPTION .....	42
10.3 UNUSED I/O PINS .....	46
10.4 LOW POWER MODES .....	46
10.5 INTERRUPTS .....	46
10.6 I/O PORT IMPLEMENTATION .....	46
<b>11 ON-CHIP PERIPHERALS</b> .....	<b>48</b>
11.1 LITE TIMER (LT) .....	48
11.2 12-BIT AUTORELOAD TIMER (AT) .....	53
11.3 SERIAL PERIPHERAL INTERFACE (SPI) .....	59
11.4 8-BIT A/D CONVERTER (ADC) .....	70
<b>12 INSTRUCTION SET</b> .....	<b>75</b>
12.1 ST7 ADDRESSING MODES .....	75
12.2 INSTRUCTION GROUPS .....	78
<b>13 ELECTRICAL CHARACTERISTICS</b> .....	<b>81</b>
13.1 PARAMETER CONDITIONS .....	81
13.2 ABSOLUTE MAXIMUM RATINGS .....	82
13.3 OPERATING CONDITIONS .....	83
13.4 SUPPLY CURRENT CHARACTERISTICS .....	89
13.5 CLOCK AND TIMING CHARACTERISTICS .....	91
13.6 MEMORY CHARACTERISTICS .....	92
13.7 EMC (ELECTROMAGNETIC COMPATIBILITY) CHARACTERISTICS .....	93
13.8 I/O PORT PIN CHARACTERISTICS .....	95
13.9 CONTROL PIN CHARACTERISTICS .....	100
13.10 COMMUNICATION INTERFACE CHARACTERISTICS .....	102
13.11 8-BIT ADC CHARACTERISTICS .....	104
<b>14 PACKAGE CHARACTERISTICS</b> .....	<b>109</b>
14.1 PACKAGE MECHANICAL DATA .....	109
14.2 THERMAL CHARACTERISTICS .....	111
<b>15 DEVICE CONFIGURATION AND ORDERING INFORMATION</b> .....	<b>112</b>
15.1 OPTION BYTES .....	112
15.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE . . .	114
15.3 DEVELOPMENT TOOLS .....	117
15.4 ST7 APPLICATION NOTES .....	118
<b>16 KNOWN LIMITATIONS</b> .....	<b>121</b>
16.1 EXECUTION OF BTJX INSTRUCTION .....	121
16.2 IN-CIRCUIT PROGRAMMING OF DEVICES PREVIOUSLY PROGRAMMED WITH HARD- WARE WATCHDOG OPTION 121	
16.3 IN-CIRCUIT DEBUGGING WITH HARDWARE WATCHDOG .....	121
16.4 RECOMMENDATIONS WHEN LVD IS ENABLED .....	121
16.5 CLEARING ACTIVE INTERRUPTS OUTSIDE INTERRUPT ROUTINE .....	121
<b>17 REVISION HISTORY</b> .....	<b>122</b>

---

# Table of Contents

---

To obtain the most recent version of this datasheet,  
please check at [www.st.com](http://www.st.com)

Please also pay special attention to the Section “KNOWN LIMITATIONS” on page 121.

## 1 DESCRIPTION

The ST7LITE0x and ST7SUPERLITE (ST7LITESx) are members of the ST7 microcontroller family. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

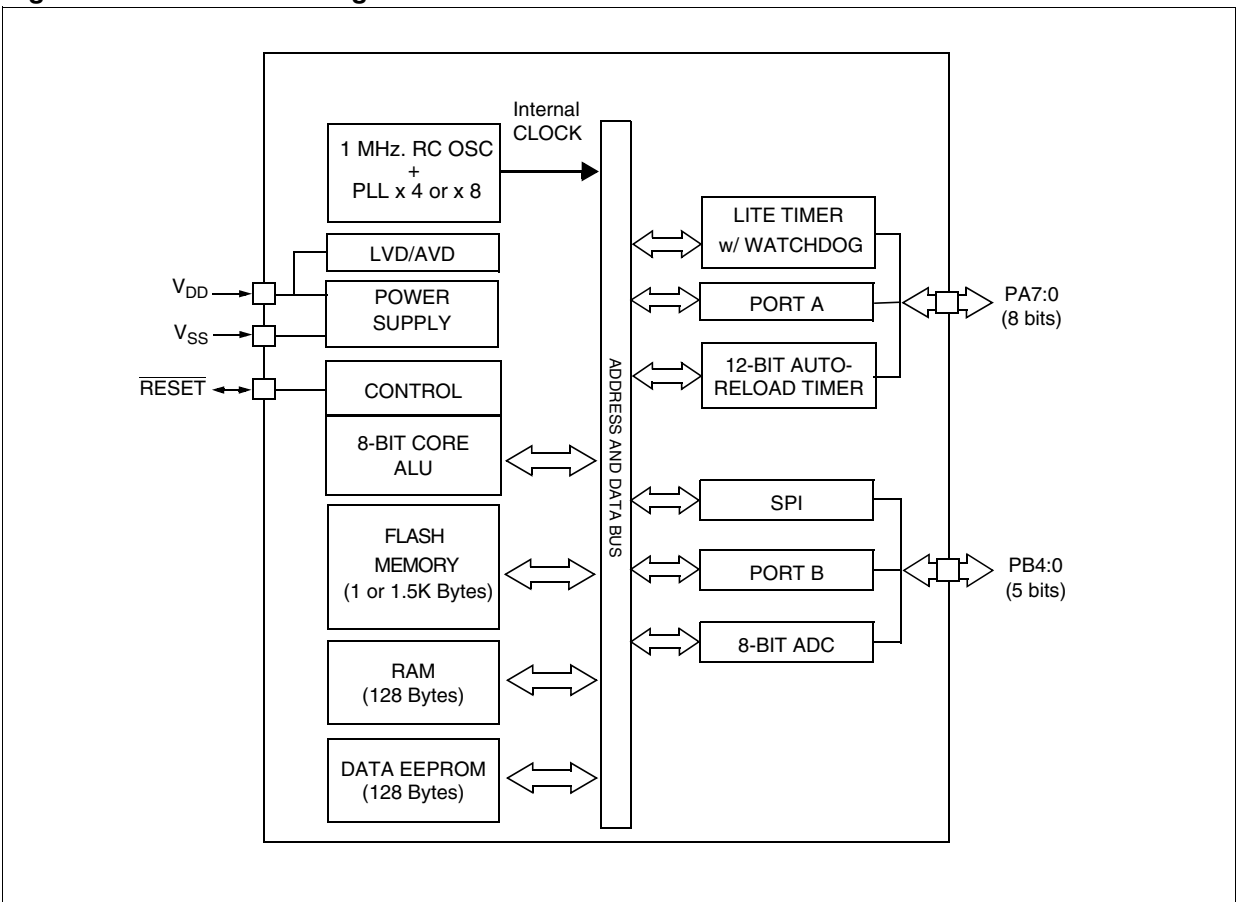
The ST7LITE0x and ST7SUPERLITE feature FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

Under software control, the ST7LITE0x and ST7SUPERLITE devices can be placed in WAIT, SLOW, or HALT mode, reducing power consumption when the application is in idle or standby state.

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

For easy reference, all parametric data are located in [section 13 on page 81](#).

**Figure 1. General Block Diagram**



## 2 PIN DESCRIPTION

Figure 2. 20-Pin QFN Package Pinout

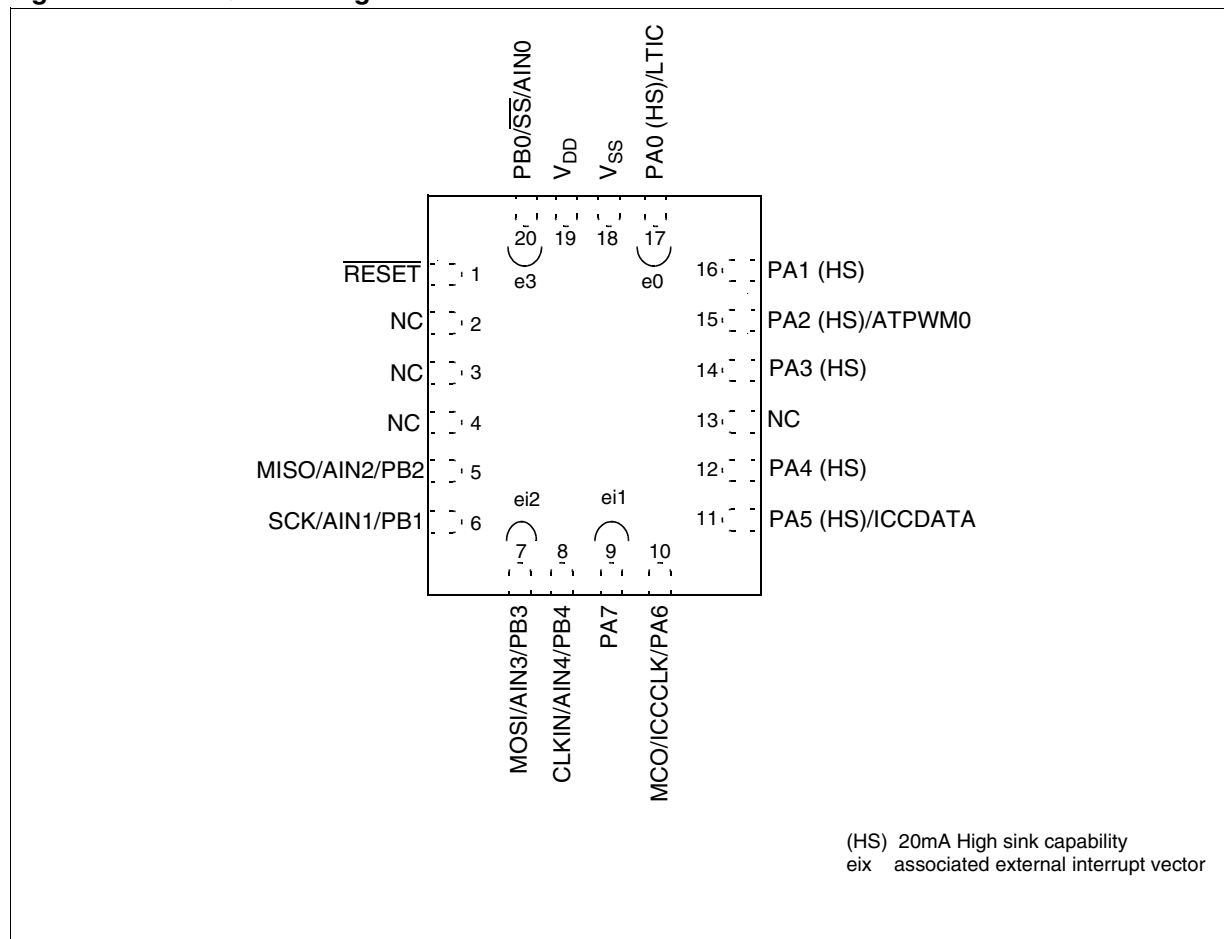
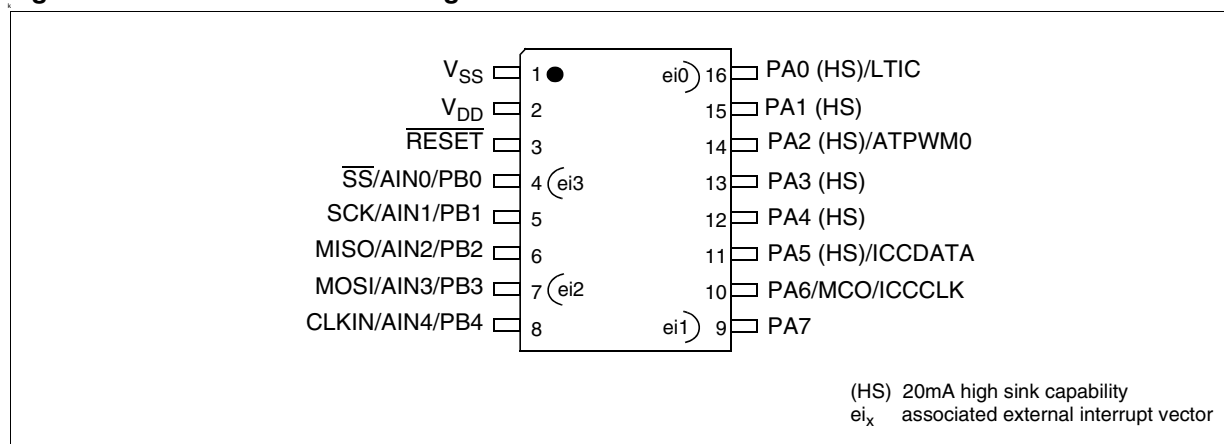


Figure 3. 16-Pin SO and DIP Package Pinout





**PIN DESCRIPTION** (Cont'd)**Legend / Abbreviations for Table 1:**

Type: I = input, O = output, S = supply

In/Output level: C = CMOS 0.15V<sub>DD</sub>/0.85V<sub>DD</sub> with input triggerC<sub>T</sub> = CMOS 0.3V<sub>DD</sub>/0.7V<sub>DD</sub> with input trigger

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

– Input: float = floating, wpu = weak pull-up, int = interrupt <sup>1)</sup>, ana = analog

– Output: OD = open drain, PP = push-pull

**Table 1. Device Pin Description**

Pin n°		Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
QFN20	SO16/DIP16			Input	Output	Input				Output			
						float	wpu	int	ana	OD	PP		
18	1	V <sub>SS</sub>	S										Ground
19	2	V <sub>DD</sub>	S										Main power supply
1	3	$\overline{\text{RESET}}$	I/O	C <sub>T</sub>		X				X			Top priority non maskable interrupt (active low)
20	4	PB0/AIN0/ $\overline{\text{SS}}$	I/O	C <sub>T</sub>	X	ei3		X	X	X		<b>Port B0</b>	ADC Analog Input 0 or SPI Slave Select (active low)
6	5	PB1/AIN1/SCK	I/O	C <sub>T</sub>	X	X		X	X	X		<b>Port B1</b>	ADC Analog Input 1 or SPI Clock <b>Caution:</b> No negative current injection allowed on this pin. For details, refer to <a href="#">section 13.2.2 on page 82</a>
5	6	PB2/AIN2/MISO	I/O	C <sub>T</sub>	X	X		X	X	X		<b>Port B2</b>	ADC Analog Input 2 or SPI Master In/ Slave Out Data
7	7	PB3/AIN3/MOSI	I/O	C <sub>T</sub>	X	ei2		X	X	X		<b>Port B3</b>	ADC Analog Input 3 or SPI Master Out / Slave In Data
8	8	PB4/AIN4/CLKIN	I/O	C <sub>T</sub>	X	X		X	X	X		<b>Port B4</b>	ADC Analog Input 4 or External clock input
9	9	PA7	I/O	C <sub>T</sub>	X	ei1				X	X	<b>Port A7</b>	
10	10	PA6 /MCO/ ICCCLK	I/O	C <sub>T</sub>	X	X				X	X	<b>Port A6</b>	Main Clock Output/In Circuit Communication Clock. <b>Caution:</b> During normal operation this pin must be pulled-up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset will put it back in input pull-up
11	11	PA5/ ICCDATA	I/O	C <sub>T</sub>	HS	X	X			X	X	<b>Port A5</b>	In Circuit Communication Data
12	12	PA4	I/O	C <sub>T</sub>	HS	X	X			X	X	<b>Port A4</b>	
14	13	PA3	I/O	C <sub>T</sub>	HS	X	X			X	X	<b>Port A3</b>	



## ST7LITE0xY0, ST7LITESxY0

Pin n°		Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
QFN20	SO16/DIP16			Input	Output	Input				Output			
						float	wpu	int	ana	OD	PP		
15	14	PA2/ATPWM0	I/O	C <sub>T</sub>	HS	X	X			X	X	<b>Port A2</b>	Auto-Reload Timer PWM0
16	15	PA1	I/O	C <sub>T</sub>	HS	X	X			X	X	<b>Port A1</b>	
17	16	PA0/LTIC	I/O	C <sub>T</sub>	HS	X	ei0			X	X	<b>Port A0</b>	Lite Timer Input Capture

### Note:

In the interrupt input column, “ei<sub>x</sub>” defines the associated external interrupt vector. If the weak pull-up column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, else the configuration is floating interrupt input.

### 3 REGISTER & MEMORY MAP

As shown in [Figure 4](#) and [Figure 5](#), the MCU is capable of addressing 64K bytes of memories and I/O registers.

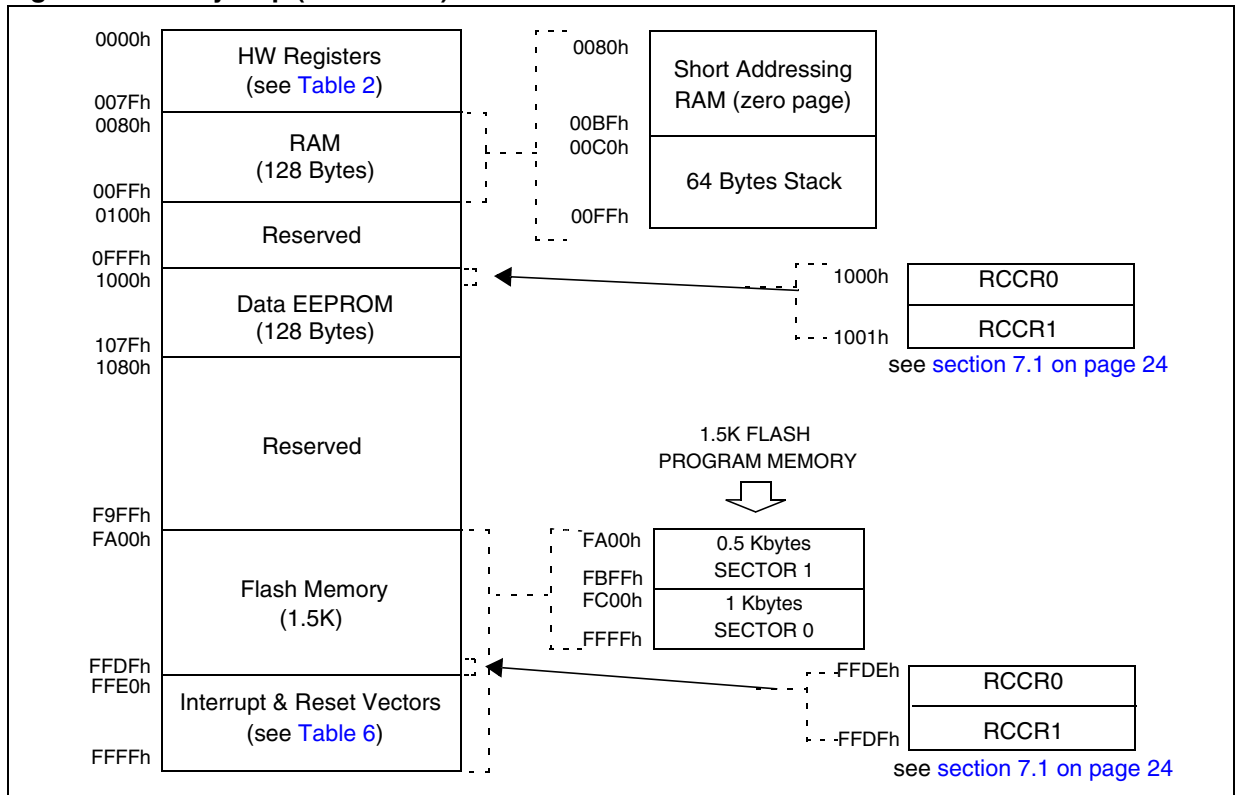
The available memory locations consist of up to 128 bytes of register locations, 128 bytes of RAM, 128 bytes of data EEPROM and up to 1.5 Kbytes of user program memory. The RAM space includes up to 64 bytes for the stack from 0C0h to 0FFh.

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

The size of Flash Sector 0 is configurable by Option byte.

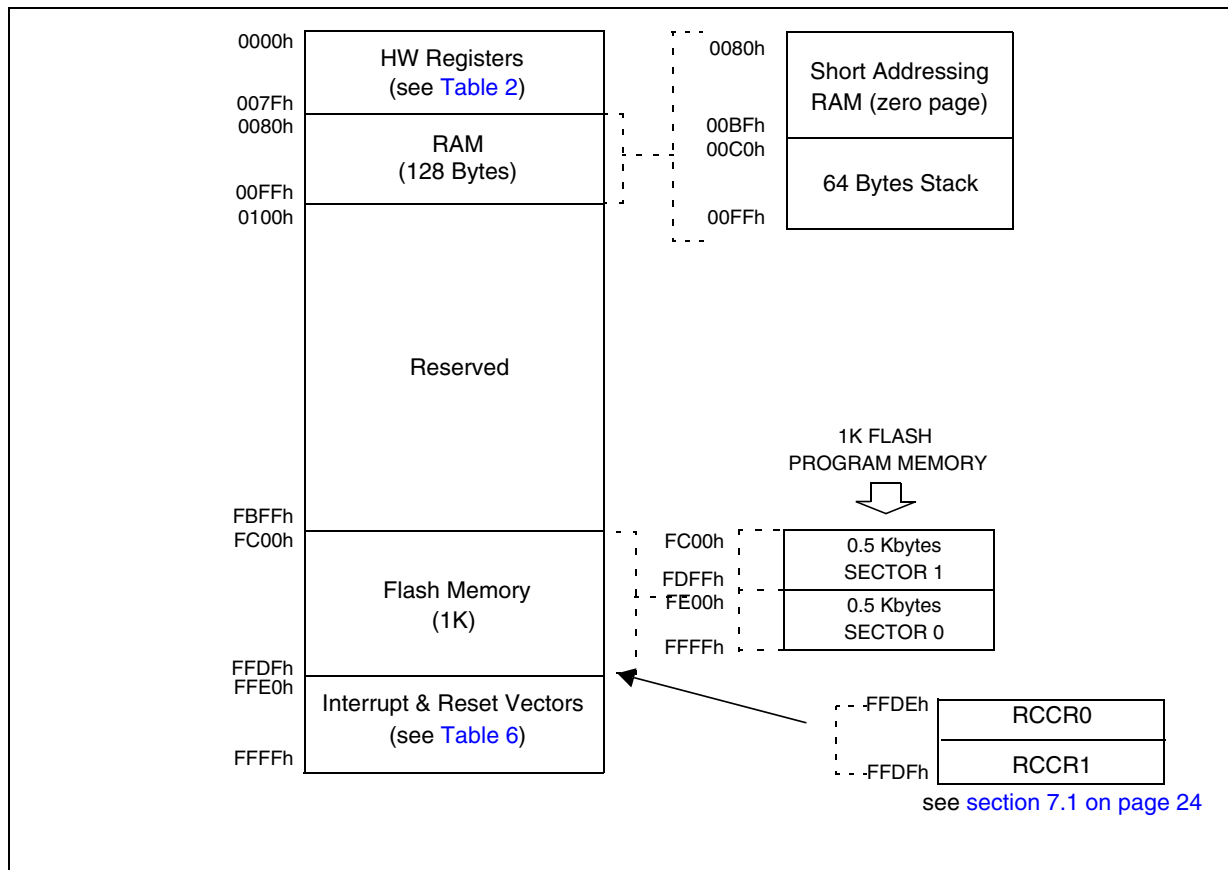
**IMPORTANT:** Memory locations marked as “Reserved” must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

**Figure 4. Memory Map (ST7LITE0x)**



REGISTER AND MEMORY MAP (Cont'd)

Figure 5. Memory Map (ST7SUPERLITE)



## REGISTER AND MEMORY MAP (Cont'd)

Legend: x=undefined, R/W=read/write

Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h 0001h 0002h	Port A	PADR PADDR PAOR	Port A Data Register Port A Data Direction Register Port A Option Register	00h <sup>1)</sup> 00h 40h	R/W R/W R/W
0003h 0004h 0005h	Port B	PBDR PBDDR PBOR	Port B Data Register Port B Data Direction Register Port B Option Register	E0h <sup>1)</sup> 00h 00h	R/W R/W R/W <sup>2)</sup>
0006h to 000Ah	Reserved area (5 bytes)				
000Bh 000Ch	LITE TIMER	LTCSR LTICR	Lite Timer Control/Status Register Lite Timer Input Capture Register	xxh xxh	R/W Read Only
000Dh 000Eh 000Fh 0010h 0011h 0012h 0013h	AUTO-RELOAD TIMER	ATCSR CNTRH CNTRL ATRH ATRL PWMCR PWM0CSR	Timer Control/Status Register Counter Register High Counter Register Low Auto-Reload Register High Auto-Reload Register Low PWM Output Control Register PWM 0 Control/Status Register	00h 00h 00h 00h 00h 00h 00h	R/W Read Only Read Only R/W R/W R/W R/W
0014h to 0016h	Reserved area (3 bytes)				
0017h 0018h	AUTO-RELOAD TIMER	DCR0H DCR0L	PWM 0 Duty Cycle Register High PWM 0 Duty Cycle Register Low	00h 00h	R/W R/W
0019h to 002Eh	Reserved area (22 bytes)				
002Fh	FLASH	FCSR	Flash Control/Status Register	00h	R/W
00030h	EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
0031h 0032h 0033h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control/Status Register	xxh 0xh 00h	R/W R/W R/W
0034h 0035h 0036h	ADC	ADCCSR ADCDR ADCAMP	A/D Control Status Register A/D Data Register A/D Amplifier Control Register	00h 00h 00h	R/W Read Only R/W
0037h	ITC	EICR	External Interrupt Control Register	00h	R/W
0038h 0039h	CLOCKS	MCCSR RCCR	Main Clock Control/Status Register RC oscillator Control Register	00h FFh	R/W R/W

## ST7LITE0xY0, ST7LITESxY0

---

Address	Block	Register Label	Register Name	Reset Status	Remarks
003Ah	SI	SICSR	System Integrity Control/Status Register	0xh	R/W
003Bh to 007Fh	Reserved area (69 bytes)				

### Notes:

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
2. The bits associated with unavailable pins must always keep their reset value.

## 4 FLASH PROGRAM MEMORY

### 4.1 Introduction

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

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using In-Circuit Programming or In-Application Programming.

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

### 4.2 Main Features

- ICP (In-Circuit Programming)
- IAP (In-Application Programming)
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Read-out and write protection

### 4.3 PROGRAMMING MODES

The ST7 can be programmed in three different ways:

- Insertion in a programming tool. In this mode, FLASH sectors 0 and 1, option byte row and data EEPROM can be programmed or erased.
- In-Circuit Programming. In this mode, FLASH sectors 0 and 1, option byte row and data EEPROM can be programmed or erased without removing the device from the application board.
- In-Application Programming. In this mode, sector 1 and data EEPROM can be programmed or erased without removing the device from the application board and while the application is running.

#### 4.3.1 In-Circuit Programming (ICP)

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

Switch the ST7 to ICC mode (In-Circuit Communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the RESET pin is pulled low. When the ST7 enters ICC mode, it fetches a specific RESET vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

- Download ICP Driver code in RAM from the ICCDATA pin
- Execute ICP Driver code in RAM to program the FLASH memory

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

#### 4.3.2 In Application Programming (IAP)

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

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

IAP mode can be used to program any memory areas except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

FLASH PROGRAM MEMORY (Cont'd)

4.4 ICC interface

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

- RESET: device reset
- V<sub>SS</sub>: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- CLKIN: main clock input for external source
- V<sub>DD</sub>: application board power supply (optional, see Note 3)

Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.
2. During the ICP session, the programming tool must control the RESET pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at

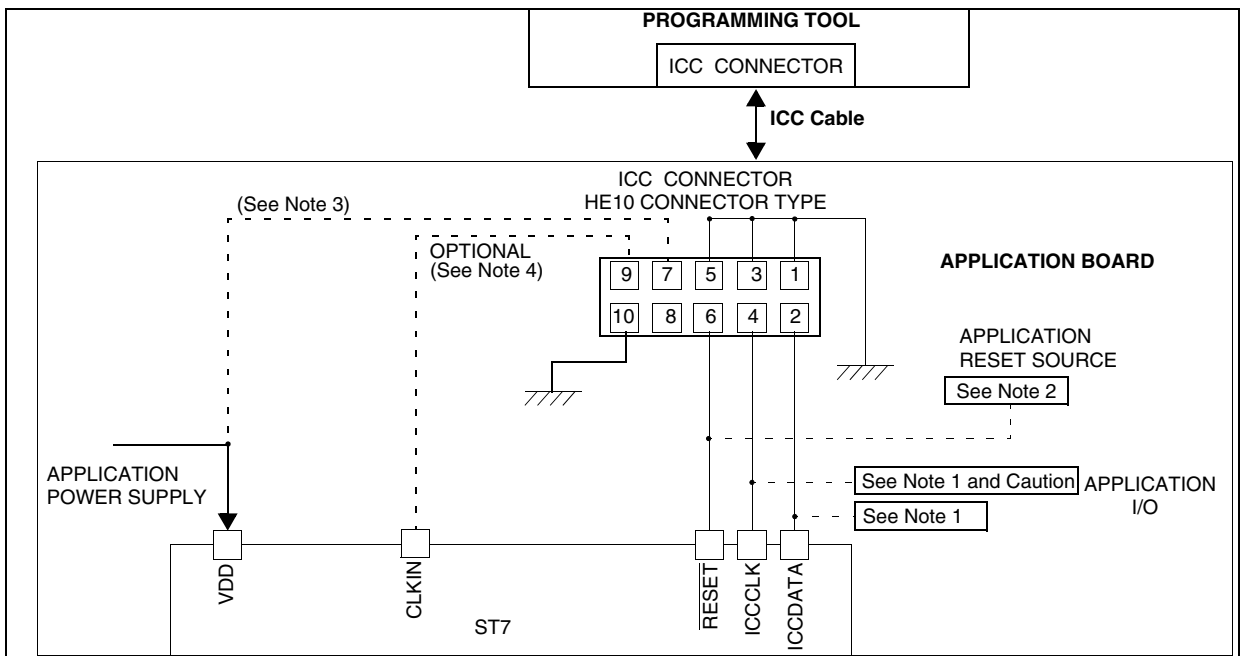
high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor>1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4. Pin 9 has to be connected to the CLKIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte.

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

Figure 6. Typical ICC Interface





## FLASH PROGRAM MEMORY (Cont'd)

### 4.5 Memory Protection

There are two different types of memory protection: Read Out Protection and Write/Erase Protection which can be applied individually.

#### 4.5.1 Read out Protection

Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data E<sup>2</sup> memory are protected.

In flash devices, this protection is removed by re-programming the option. In this case, both program and data E<sup>2</sup> memory are automatically erased, and the device can be reprogrammed.

Read-out protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP\_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the Option List.

#### 4.5.2 Flash Write/Erase Protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to E<sup>2</sup> data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

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

**Table 3. 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

Write/erase protection is enabled through the FMP\_W bit in the option byte.

### 4.6 Related Documentation

For details on Flash programming and ICC protocol, refer to the ST7 Flash Programming Reference Manual and to the ST7 ICC Protocol Reference Manual.

### 4.7 Register Description

#### FLASH CONTROL/STATUS REGISTER (FCSR)

Read/Write

Reset Value: 000 0000 (00h)

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

7							0
0	0	0	0	0	OPT	LAT	PGM

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

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

## 5 DATA EEPROM

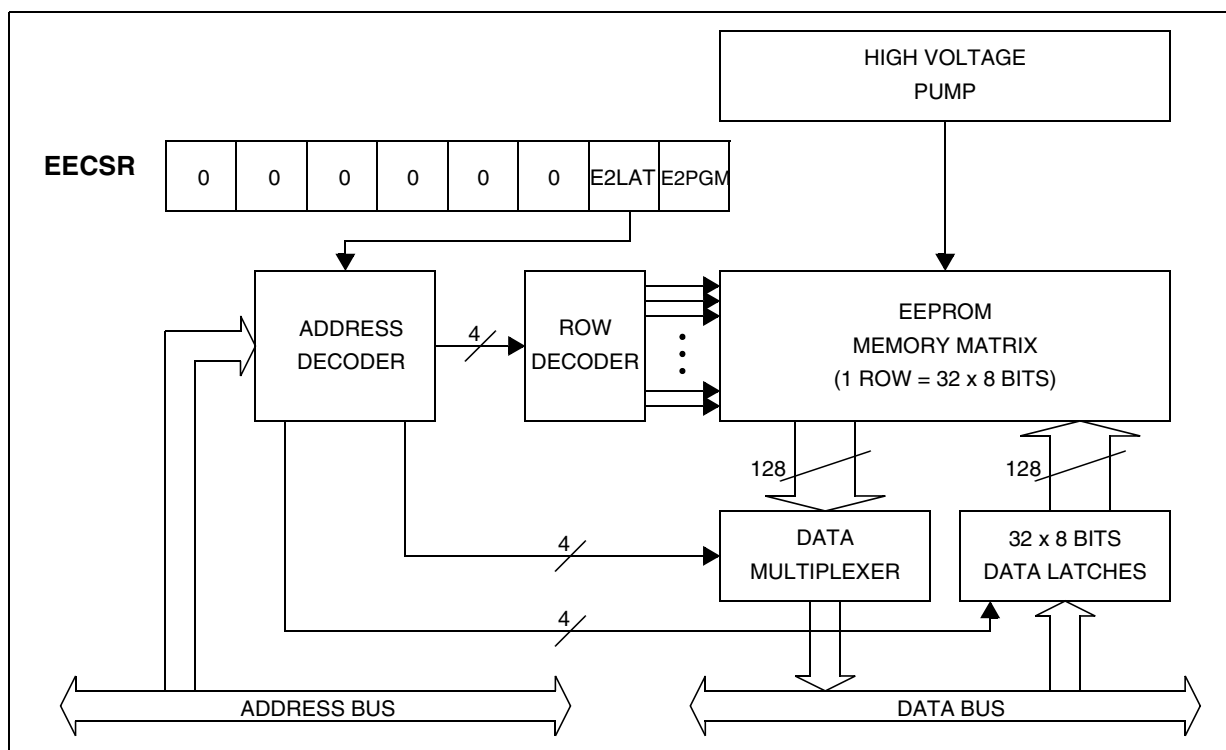
### 5.1 INTRODUCTION

The Electrically Erasable Programmable Read Only Memory can be used as a non-volatile back-up for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

### 5.2 MAIN FEATURES

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

Figure 7. EEPROM Block Diagram



## DATA EEPROM (Cont'd)

## 5.3 MEMORY ACCESS

The Data EEPROM memory read/write access modes are controlled by the E2LAT bit of the EEPROM Control/Status register (EECSR). The flowchart in [Figure 8](#) describes these different memory access modes.

**Read Operation (E2LAT = 0)**

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

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

**Write Operation (E2LAT = 1)**

To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs,

the value is latched inside the 32 data latches according to its address.

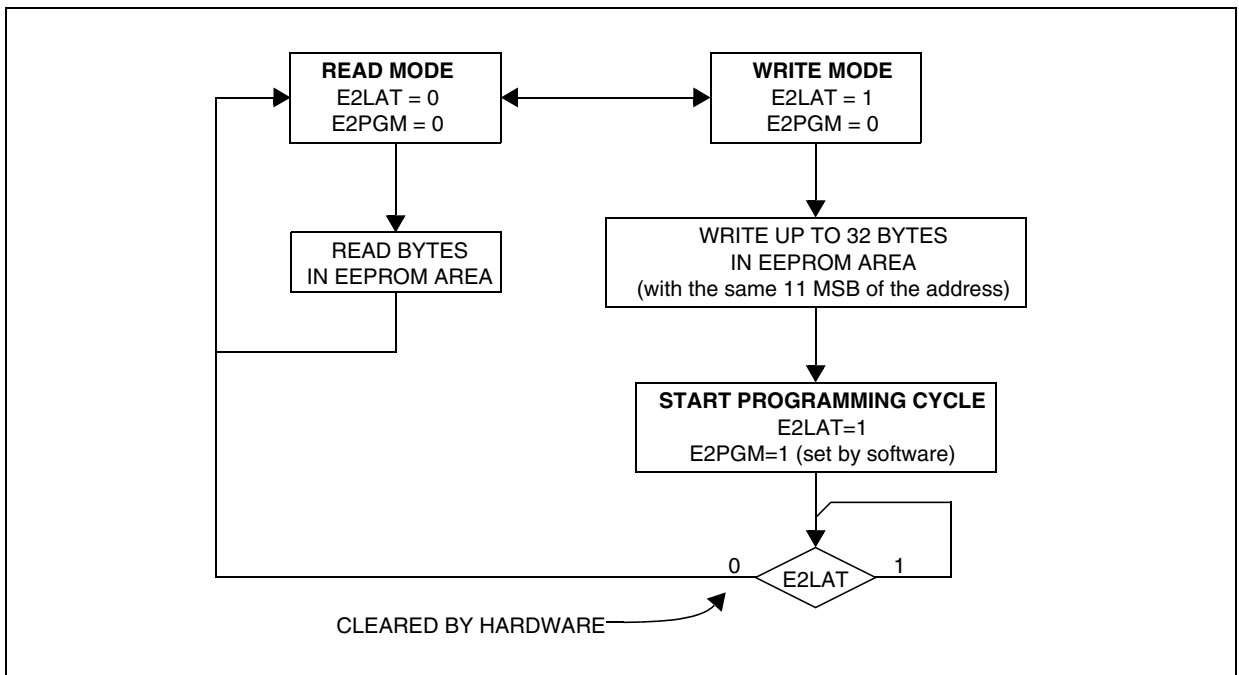
When PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between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

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

**Note:** Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit.

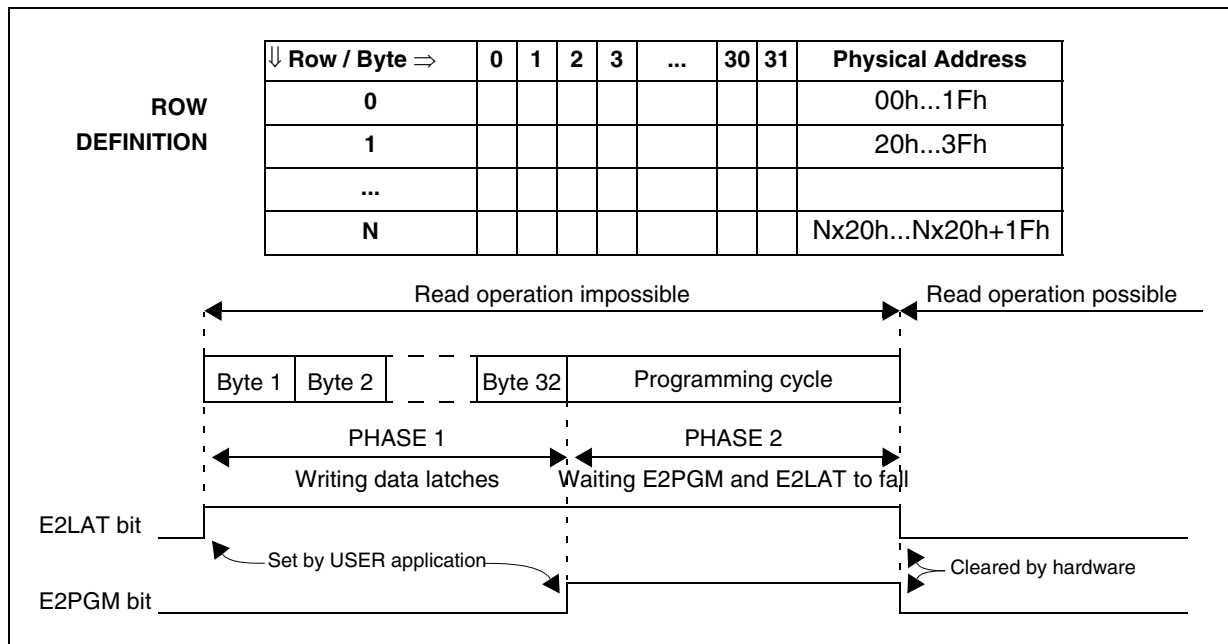
It is not possible to read the latched data. This note is illustrated by the [Figure 10](#).

**Figure 8. Data EEPROM Programming Flowchart**



DATA EEPROM (Cont'd)

Figure 9. Data E<sup>2</sup>PROM Write Operation



**Note:** If a programming cycle is interrupted (by RESET action), the integrity of the data in memory will not be guaranteed.

## DATA EEPROM (Cont'd)

### 5.4 POWER SAVING MODES

#### Wait mode

The DATA EEPROM can enter WAIT mode on execution of the WFI instruction of the microcontroller or when the microcontroller enters Active Halt mode. The DATA EEPROM will immediately enter this mode if there is no programming in progress, otherwise the DATA EEPROM will finish the cycle and then enter WAIT mode.

#### Active Halt mode

Refer to Wait mode.

#### Halt mode

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

### 5.5 ACCESS ERROR HANDLING

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

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

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

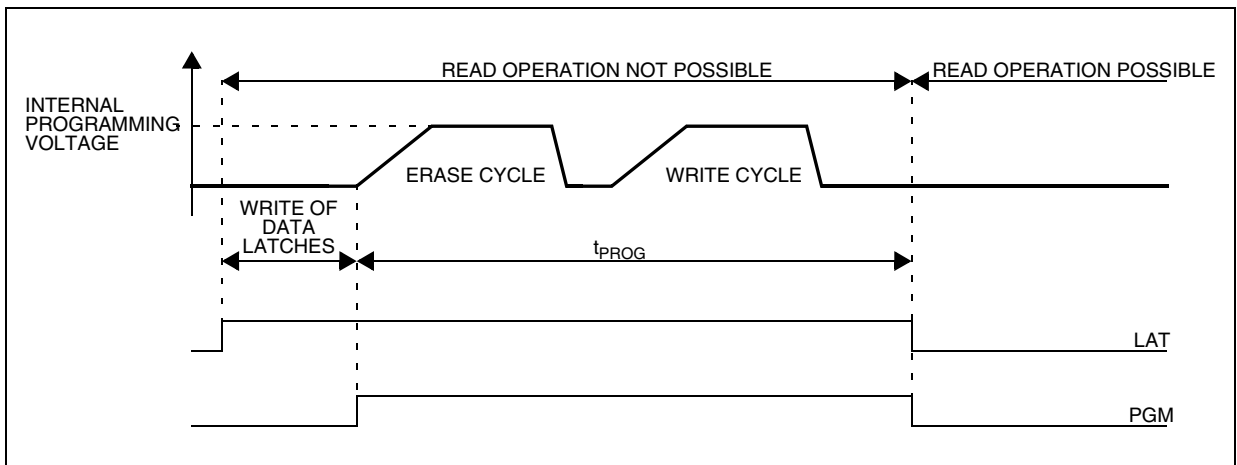
### 5.6 DATA EEPROM READ-OUT PROTECTION

The read-out protection is enabled through an option bit (see option byte section).

When this option is selected, the programs and data stored in the EEPROM memory are protected against read-out (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire Program memory and EEPROM is first automatically erased.

**Note:** Both Program Memory and data EEPROM are protected using the same option bit.

Figure 10. Data EEPROM Programming Cycle



**DATA EEPROM (Cont'd)**

**5.7 REGISTER DESCRIPTION**

**EEPROM CONTROL/STATUS REGISTER (EECSR)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	E2LAT	E2PGM

Bits 7:2 = Reserved, forced by hardware to 0.

**Bit 1 = E2LAT *Latch Access Transfer***

This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if the E2PGM bit is cleared.

- 0: Read mode
- 1: Write mode

**Bit 0 = E2PGM *Programming control and status***

This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware.

- 0: Programming finished or not yet started
- 1: Programming cycle is in progress

**Note:** If the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed.

**Table 4. DATA EEPROM Register Map and Reset Values**

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

## 6 CENTRAL PROCESSING UNIT

### 6.1 INTRODUCTION

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

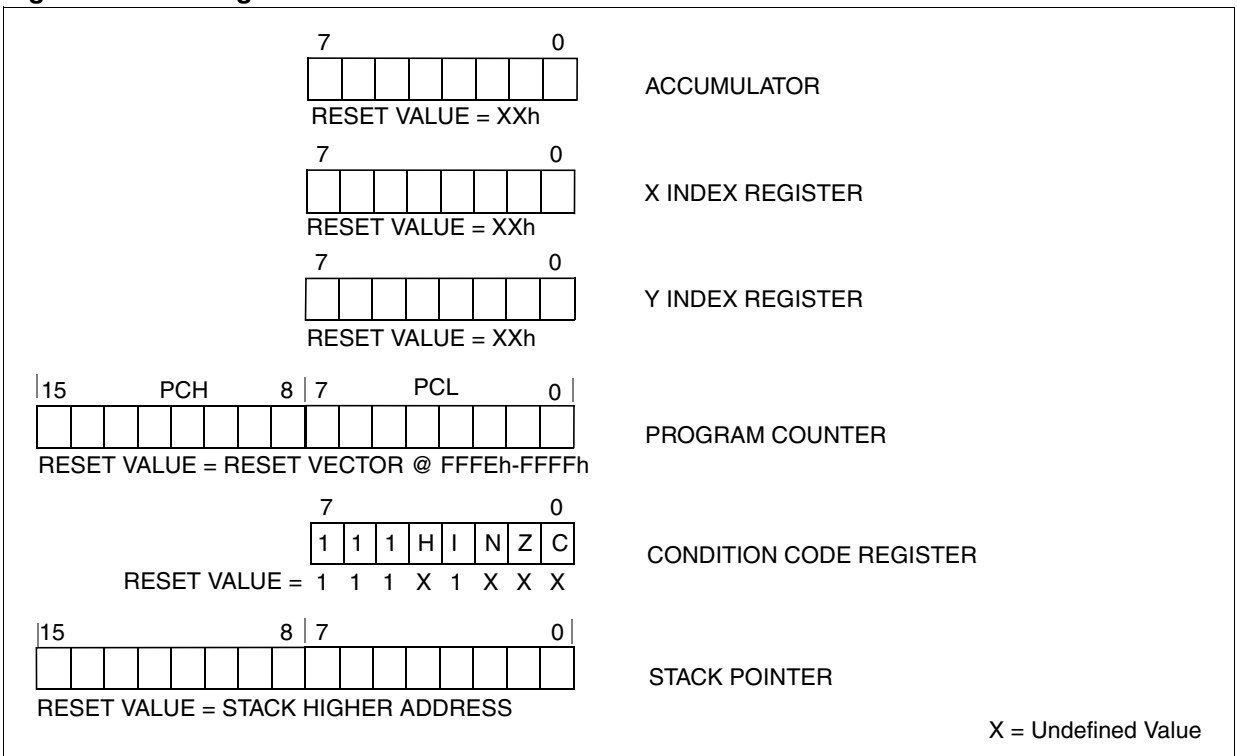
### 6.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

### 6.3 CPU REGISTERS

The six CPU registers shown in [Figure 11](#) are not present in the memory mapping and are accessed by specific instructions.

**Figure 11. CPU Registers**



### 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.

### Index Registers (X and Y)

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

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

### Program Counter (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).



**CPU REGISTERS** (Cont'd)**CONDITION CODE REGISTER (CC)**

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

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

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

**Bit 4 = H Half carry.**

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

**Bit 3 = I Interrupt mask.**

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.

0: Interrupts are enabled.

1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

**Note:** Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible

because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

**Bit 2 = N Negative.**

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7<sup>th</sup> bit of the result.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

**Bit 1 = Z Zero.**

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.

1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

**Bit 0 = C Carry/borrow.**

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

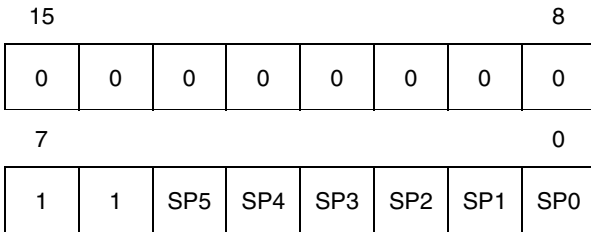
This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

**CPU REGISTERS** (Cont'd)

**Stack Pointer (SP)**

Read/Write

Reset Value: 00 FFh



The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 12).

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

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

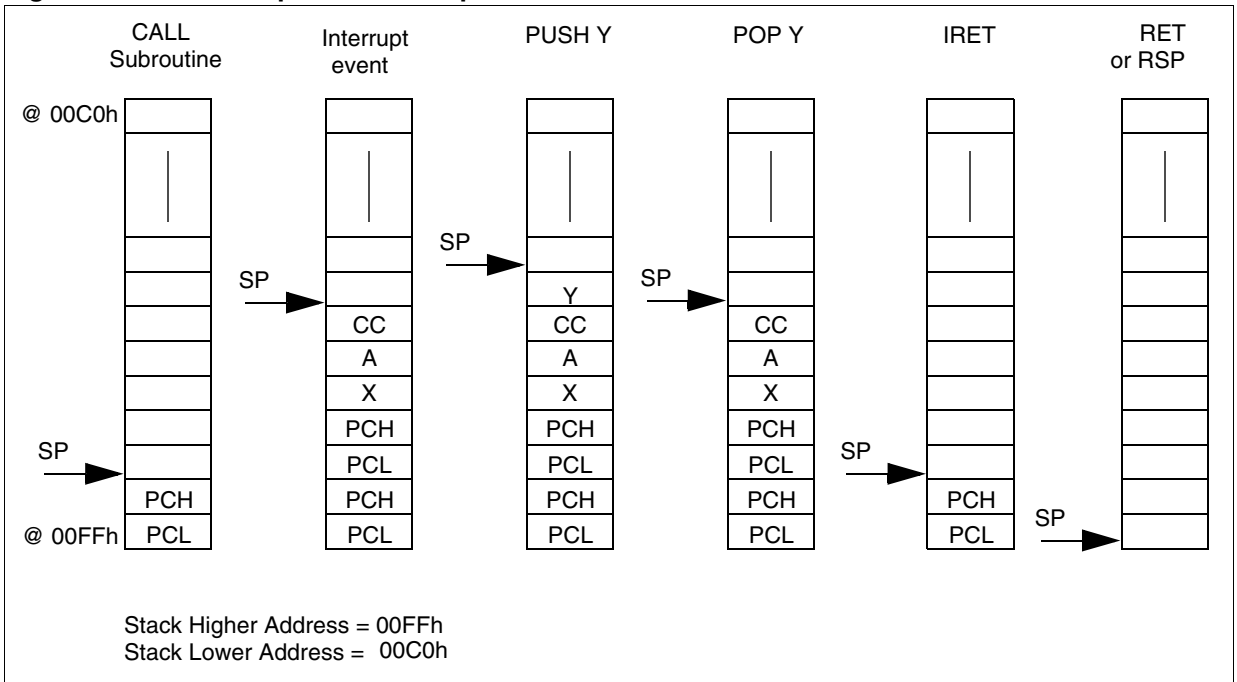
**Note:** When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 12.

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

**Figure 12. Stack Manipulation Example**



## 7 SUPPLY, RESET AND CLOCK MANAGEMENT

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

### Main features

- Clock Management
  - 1 MHz internal RC oscillator (enabled by option byte)
  - External Clock Input (enabled by option byte)
  - PLL for multiplying the frequency by 4 or 8 (enabled by option byte)
- Reset Sequence Manager (RSM)
- System Integrity Management (SI)
  - Main supply Low voltage detection (LVD) with reset generation (enabled by option byte)
  - Auxiliary Voltage detector (AVD) with interrupt capability for monitoring the main supply (enabled by option byte)

### 7.1 INTERNAL RC OSCILLATOR ADJUSTMENT

The ST7 contains an internal RC oscillator with an accuracy of 1% for a given device, temperature and voltage. It must be calibrated to obtain the frequency required in the application. This is done by software writing a calibration value in the RCCR (RC Control Register).

Whenever the microcontroller is reset, the RCCR returns to its default value (FFh), i.e. each time the device is reset, the calibration value must be loaded in the RCCR. Predefined calibration values are stored in EEPROM for 3.0 and 5V  $V_{DD}$  supply voltages at 25°C, as shown in the following table.

#### Notes:

- See “ELECTRICAL CHARACTERISTICS” on page 81. for more information on the frequency and accuracy of the RC oscillator.
- To improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the  $V_{DD}$  and  $V_{SS}$  pins as close as possible to the ST7 device.

RCCR	Conditions	ST7FLITE09 Address	ST7FLITE05/ST7FLITES5 Address
RCCR0	$V_{DD}=5V$ $T_A=25^{\circ}C$ $f_{RC}=1MHz$	1000h and FFDEh	FFDEh
RCCR1	$V_{DD}=3.0V$ $T_A=25^{\circ}C$ $f_{RC}=700KHz$	1001h and FFDFh	FFDFh

– These two bytes are systematically programmed by ST, including on FASTROM devices. Consequently, customers intending to use FASTROM service must not use these two bytes.

– RCCR0 and RCCR1 calibration values will be erased if the read-out protection bit is reset after it has been set. See “Read out Protection” on page 15.

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

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

### 7.2 PHASE LOCKED LOOP

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

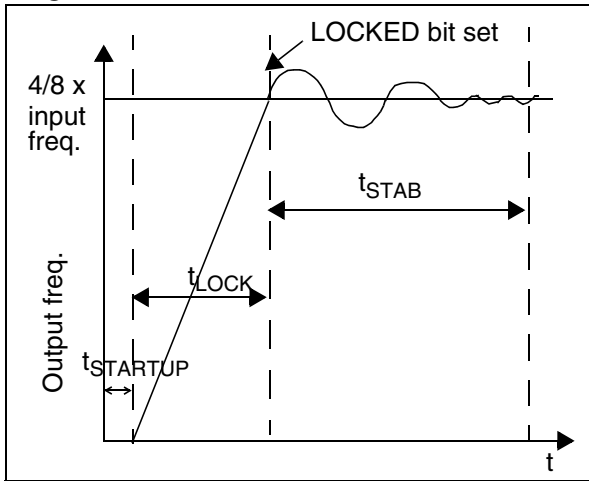
- The x4 PLL is intended for operation with  $V_{DD}$  in the 2.4V to 3.3V range
- The x8 PLL is intended for operation with  $V_{DD}$  in the 3.3V to 5.5V range

Refer to [Section 15.1](#) for the option byte description.

If the PLL is disabled and the RC oscillator is enabled, then  $f_{OSC} = 1MHz$ .

If both the RC oscillator and the PLL are disabled,  $f_{OSC}$  is driven by the external clock.

**Figure 13. PLL Output Frequency Timing Diagram**



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

When the PLL output signal reaches the operating frequency, the LOCKED bit in the SICSCR register is set. Full PLL accuracy ( $ACC_{PLL}$ ) is reached after a stabilization time of  $t_{STAB}$  (see Figure 13 and 13.3.4 Internal RC Oscillator and PLL)

Refer to section 8.4.4 on page 36 for a description of the LOCKED bit in the SICSCR register.

**7.3 REGISTER DESCRIPTION**

**MAIN CLOCK CONTROL/STATUS REGISTER (MCCSR)**

Read / Write  
Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	MCO	SMS

Bits 7:2 = Reserved, must be kept cleared.

**Table 5. Clock Register Map and Reset Values**

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0038h	<b>MCCSR</b> Reset Value	0	0	0	0	0	0	MCO 0	SMS 0
0039h	<b>RCCR</b> Reset Value	CR7 1	CR6 1	CR5 1	CR4 1	CR3 1	CR2 1	CR1 1	CR0 1

**Bit 1 = MCO Main Clock Out enable**  
This bit is read/write by software and cleared by hardware after a reset. This bit allows to enable the MCO output clock.  
0: MCO clock disabled, I/O port free for general purpose I/O.  
1: MCO clock enabled.

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

**RC CONTROL REGISTER (RCCR)**

Read / Write  
Reset Value: 1111 1111 (FFh)

7							0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

Bits 7:0 = **CR[7:0] RC Oscillator Frequency Adjustment Bits**

These bits must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. The application can store the correct value for each voltage range in EEPROM and write it to this register at start-up.  
00h = maximum available frequency  
FFh = lowest available frequency

**Note:** To tune the oscillator, write a series of different values in the register until the correct frequency is reached. The fastest method is to use a dichotomy starting with 80h.