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

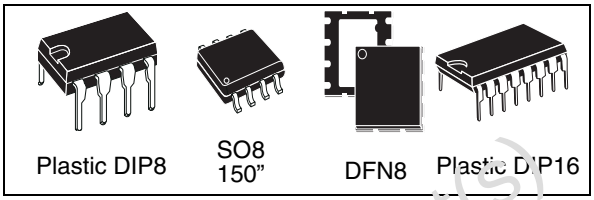
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ST7LITEUS2
ST7LITEUS5

8-bit MCU with single voltage Flash memory, ADC, timers

Features

- Memories
  - 1 Kbytes single-voltage Flash Program memory with readout protection, ICP and IAP)
    10 K write/erase cycles guaranteed
    data retention: 20 years at 55 °C
  - 128 bytes RAM
- Clock, Reset and Supply management
  - 3-level low-voltage supervisor (LVD) and auxiliary voltage detector (AVD) for safe power-on/off
  - Clock sources: internal trimmable 8 MHz RC oscillator, internal low power, low frequency RC oscillator or external clock
  - Five power saving modes: Halt, Auto-wakeup from Halt, Active-halt, Wait, Slow
- Interrupt management
  - 11 interrupt vectors plus TRAP and RESET
  - 5 external interrupt lines (on 5 vectors)
- I/O ports
  - 5 multifunctional bidirectional I/O lines
  - 1 additional Output line
  - 6 alternate function lines
  - 5 high sink outputs



- 2 Timers
  - One 8-bit Lite timer (LT) with prescaler including: watchdog, one realtime base and one 8-bit input capture.
  - One 12-bit auto-reload timer (AT) with output compare function and PWM
- A/D Converter
  - 10-bit resolution for 0 to V_{DD}
  - 5 input channels
- Instruction Set
  - 8-bit data manipulation
  - 63 basic instructions with illegal opcode detection
  - 17 main addressing modes
  - 8x8 unsigned multiply instruction
- Development Tools
  - Full hardware/software development package
  - Debug module

Table 1. Device summary

Features	ST7LITEUS2	ST7LITEUS5
Program memory	1 Kbytes	
RAM (stack)	128 (64) bytes	
Peripherals	LT Timer w/ Wdg, AT Timer w/ 1 PWM	
ADC	-	10-bit
Operating Supply	2.4 to 3.3 V @f_{CPU}=4 MHz, 3.3 to 5.5 V @f_{CPU}=8 MHz	
CPU Frequency	up to 8 MHz RC	
Operating Temperature	-40 to +85 °C / -40 to 125 °C	
Packages	SO8 150”, Pastic DIP8, DFN8, Pastic DIP16(1)	

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

Contents

List of tables

List of figures

1 Introduction

The ST7LITEUS2 and ST7LITEUS5 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 ST7LITEUS2 and ST7LITEUS5 feature FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

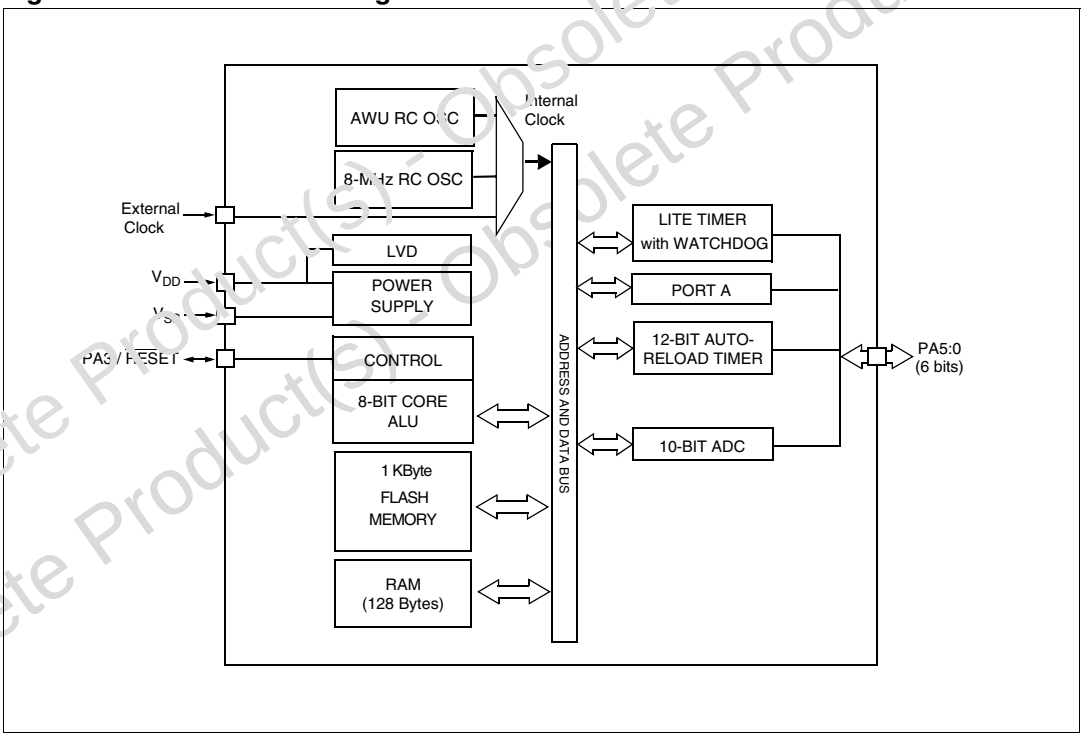
Under software control, the ST7LITEUS2 and ST7LITEUS5 can be placed in Wait, Slow, or Halt mode, reducing power consumption when the application is in idle or standby state.

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 12 on page 92*.

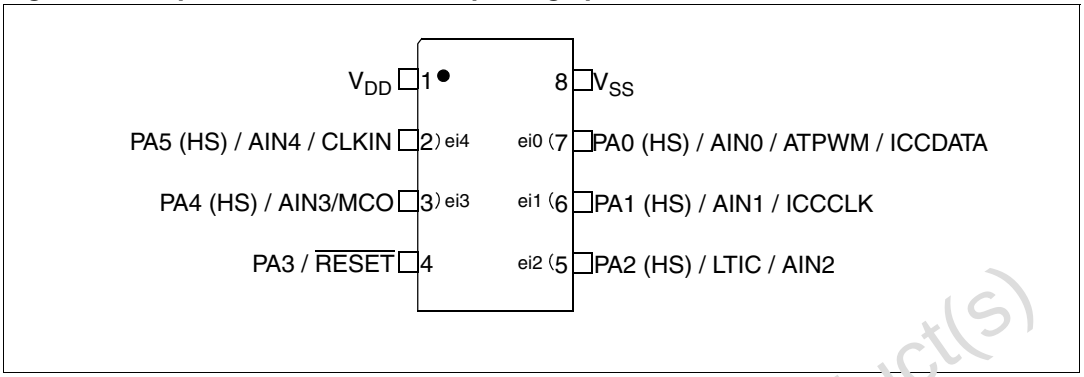
The devices feature an on-chip debug module (DM) to support in circuit debugging (ICD). For a description of the DM registers, refer to the ST7 I^{2}C protocol reference manual.

Figure 1. General block diagram



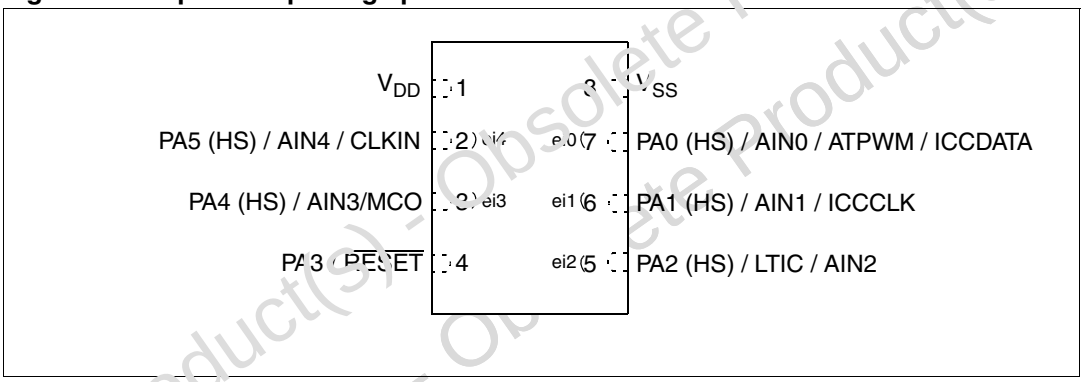
2 Pin description

Figure 2. 8-pin SO and Plastic DIP package pinout



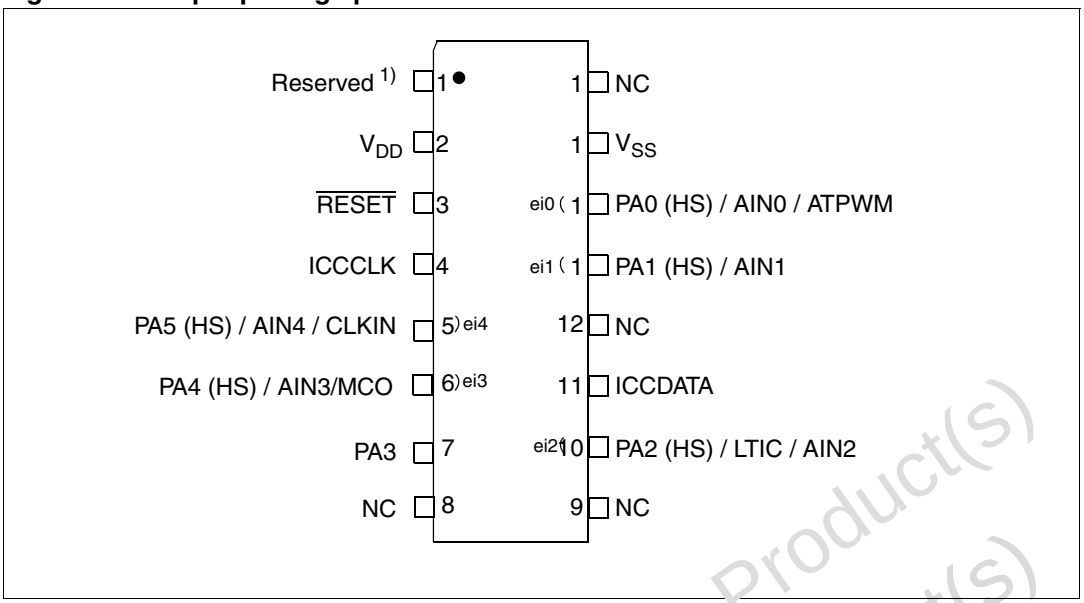
1. HS: High sink capability.
2. eix : associated external interrupt vector

Figure 3. 8-pin DFN package pinout



1. HS: High sink capability.
2. eix : associated external interrupt vector

Figure 4. 16-pin package pinout



1. Reserved pins must be tied to ground.
2. The differences versus the 8-pin packages are listed below:
   The I^2C signals (ICCCLK and ICCDATA) are mapped on dedicated pins.
   The RESET signal is mapped on a dedicated pin. It is not multiplexed with PA3.
   PA3 pin is always configured as output. Any change on multiplexed IO reset control registers (MUXCR1 and MUXCR2) will have no effect on PA3 functionality. Refer to *Section 6.5: Register description on page 37*.

Legend/abbreviations for *Table 2*

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

In/Output level: C_T= CMOS 0.3 V_{DD} /0.7 V_{DD} with input trigger

Output level: HS = High sink (on N-buffer only)

Port and control configuration

- Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog
- Output: OD = open drain, PP = push-pull

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

Table 2. Device pin description

Pin no.	Pin name	Type	Level		Port/control						Main function (after reset)	Alternate function
					Input				Output			
			Input	Output	float	wpu	int	ana	OD	PP		
1	V_{DD}	S									Main power supply	
2	PA5/AIN4/CLKIN	I/O	C_T	HS	X	ei4		X	X	X	Port A5	Analog input 4 or External Clock Input
3	PA4/AIN3/MCO	I/O	C_T	HS	X	ei3		X	X	X	Port A4	Analog input 3 or main clock output
4	PA3/$\overline{\text{RESET}}$ (1)	O				X			X	X	Port A3	RESET(1)
5	PA2/AIN2/LTIC	I/O	C_T	HS	X	ei2		X	X	X	Port A2	Analog input 2 or Lite Timer Input Capture
6	PA1/AIN1/ ICCCLK	I/O	C_T	HS	X	ei1		X	X	X	Port A1	Analog input 1 or In Circuit Communication Clock Caution: During normal operation this pin must be pulled-up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering I^2C mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset will put it back in pull-up
7	PA0/AIN0/ATPWM/ICCDATA	I/O	C_T	HS	X	ei0		X	X	X	Port A0	Analog input 0 or Auto-Reload Timer PWM or In Circuit Communication Data
8	V_{SS}	S									Ground	

1. After a reset, the multiplexed PA3/$\overline{\text{RESET}}$ pin will act as $\overline{\text{RESET}}$. To configure this pin as output (Port A3), write 55h to MUXCR0 and AAh to MUXCR1. For further details, please refer to *Section 6.5 on page 37*.

3 Register and memory map

As shown in *Figure 5*, the MCU is capable of addressing 64K bytes of memories and I/O registers.

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

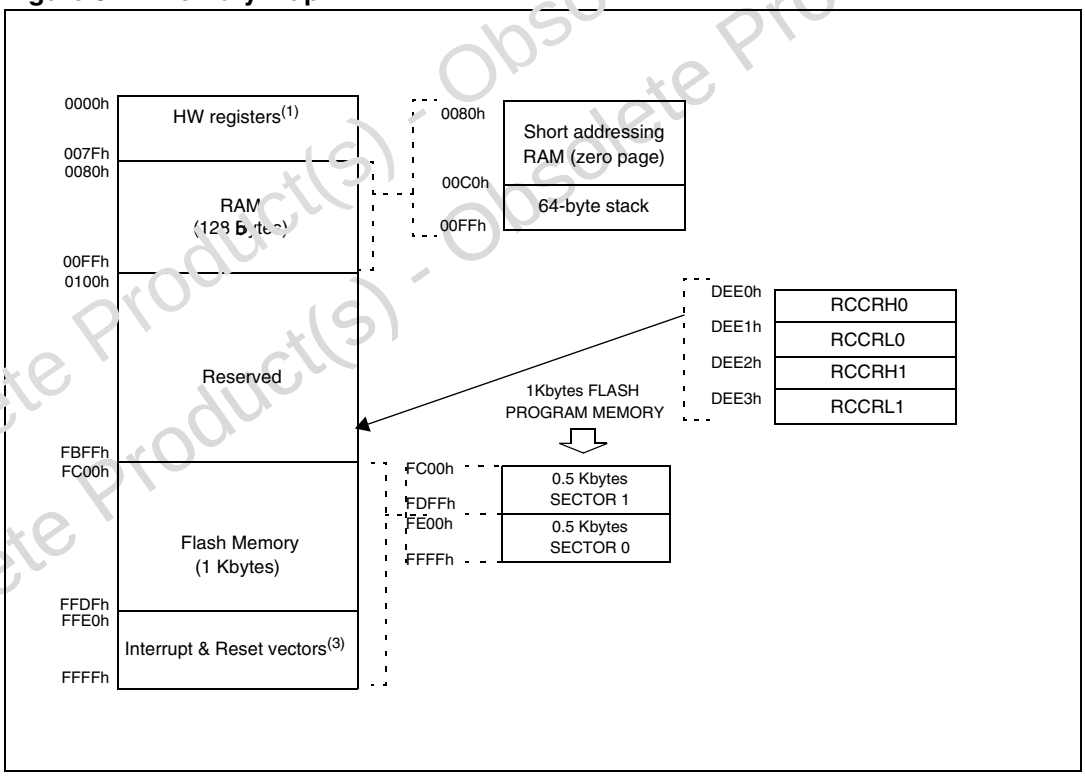
The highest address bytes contain the user reset and interrupt vectors.

The Flash memory contains two sectors (see *Figure 5*) mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (FE00h-FFFFh).

The size of Flash Sector 0 and other device options are configurable by option byte

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

Figure 5. Memory map



1. See *Table 3*.
2. See *Section 6.2 on page 28* for the description of RCCRHx registers.
3. See *Table 9*.

Table 3. Hardware register map (1)

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h 0002h	Port A	PADR PADDR PAOR	Port A Data register Port A Data Direction register Port A Option register	00h(2) 08h 02h(3)	R/W R/W R/W
0003h- 000Ah	Reserved area (8 bytes)				
000Bh 000Ch	LITE TIMER	LTCSR LTICR	Lite Timer Control/Status register Lite Timer Input Capture register	0xh 00h	R/W Read only
000Dh 000Eh 000Fh 0010h 0011h 0012h 0013h	AUTO-RELOAD TIMER	ATCSR CNTRH CNTRL ATRH ATRL PWMCR PWM0CSR	Timer Control/Status register Counter register High Counter register Low Auto-Reload register High Auto-Reload register Low PWM Output Control register PWM 0 Control/Status register	00h 00h 00h 00h 00h 00h 00h	R/W Read only Read only R/W R/W R/W R/W
0014h to 0016h	Reserved area (3 bytes)				
0017h 0018h	AUTO-RELOAD TIMER	DCR0H DCR0L	PWM 0 Duty Cycle register High PWM 0 Duty Cycle register Low	00h 00h	R/W R/W
0019h to 002Eh	Reserved area (22 bytes)				
0002Fh	FLASH	FCSR	Flash Control/Status register	00h	R/W
0030h to 0033h	Reserved area (4 bytes)				
0034h 0035h 0036h	ADC	ADCCSR ADCDRH ADCDRL	A/D Control Status register A/D Data register High A/D Data register Low	00h xxh 00h	R/W Read only R/W
0037h	ITC	EICR1	External Interrupt Control register 1	00h	R/W
0038h	MCC	MCCSR	Main Clock Control/Status register	00h	R/W
0039h 003Ah	Clock and Reset	RCCR SICSR	RC oscillator Control register System Integrity Control/Status register	FFh 0000 0x00b	R/W R/W
003Bh to 003Ch	Reserved area (2 bytes)				
003Dh	ITC	EICR2	External Interrupt Control register 2	00h	R/W
003Eh	AVD	AVDTHCR	AVD Threshold Selection register	03h	R/W
003Fh	Clock controller	CKCNTCSR	Clock Controller Control/Status register	09h	R/W
0040h to 0046h	Reserved area (7 bytes)				
0047h 0048h	MuxIO-reset	MUXCR0 MUXCR1	Mux IO-Reset Control register 0 Mux IO-Reset Control register 1	00h 00h	R/W R/W

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

Address	Block	Register label	Register name	Reset status	Remarks
0049h 004Ah	AWU	AWUPR AWUCSR	AWU Prescaler register AWU Control/Status register	FFh 00h	R/W R/W
004Bh 004Ch 004Dh 004Eh 004Fh 0050h	DM(4)	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L	DM Control register DM Status register DM Breakpoint register 1 High DM Breakpoint register 1 Low DM Breakpoint register 2 High DM Breakpoint register 2 Low	00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W
0051h to 007Fh	Reserved area (47 bytes)				

1. Legend: x=undefined, R/W=read/write
2. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
3. The bits associated with unavailable pins must always keep their reset value.
4. For a description of the DM registers, see the ST7 I^2C Protocol Reference Manual.

4 Flash program memory

4.1 Introduction

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

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using in-circuit programming or in-application programming.

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

4.2 Main features

- ICP (in-circuit programming)
- IAP (in-application programming)
- ICT (in-circuit testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Readout and write protection

4.3 Programming modes

The ST7 can be programmed in three different ways:

- Insertion in a programming tool

  In this mode, FLASH sectors 0 and 1 and option byte row can be programmed or erased.
- In-circuit programming

  In this mode, FLASH sectors 0 and 1 and option byte row can be programmed or erased without removing the device from the application board.
- In-application programming

  In this mode, sector 1 can be programmed or erased without removing the device from the application board and while the application is running.

4.3.1 In-circuit programming (ICP)

ICP uses a protocol called I^2C (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 I^2C mode. This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the $\overline{\text{RESET}}$ pin is pulled low. When the ST7 enters I^2C mode, it fetches a specific RESET vector which points to the ST7 system memory containing

the I^2C protocol routine. This routine enables the ST7 to receive bytes from the I^2C 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.

4.4 I^2C interface

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

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: I^2C output serial clock pin
- ICCDATA: I^2C input serial data pin
- CLKIN: main clock input for external source
- V_{DD}: application board power supply

Refer to *Figure 6* for a description of the I^2C interface.

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 I^2C 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.

During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push pull output or pull-up resistor<1 kΩ). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1 kΩ or a reset management IC with open drain output and pull-up resistor>1 kΩ, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the I^2C session.

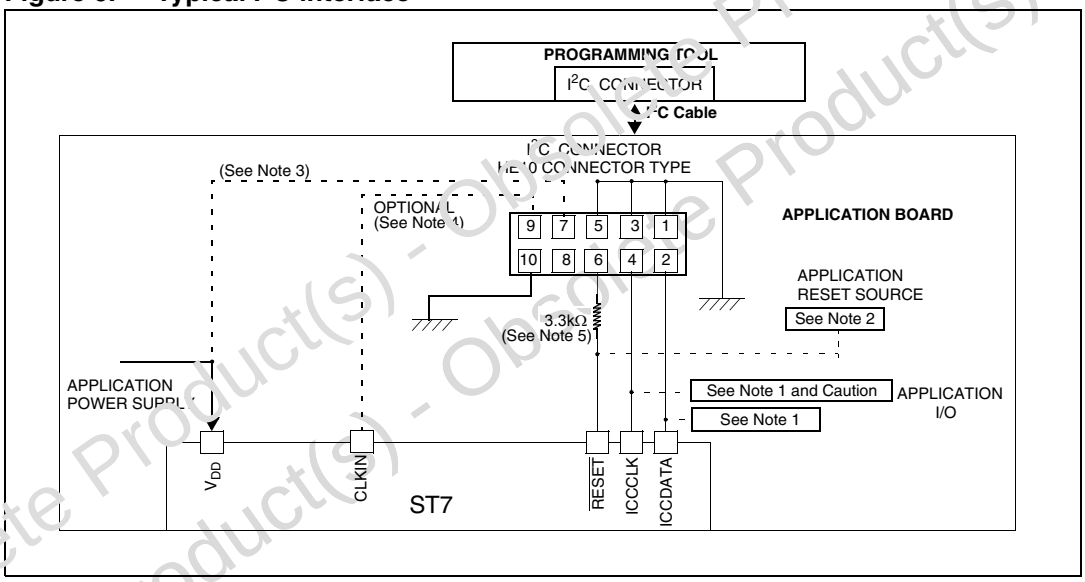
The use of Pin 7 of the I^2C connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.

Pin 9 has to be connected to the CLKIN pin of the ST7 when I^2C mode is selected with option bytes disabled (35-pulse I^2C entry mode). When option bytes are enabled (38-pulse I^2C entry mode), the internal RC clock (internal RC or AWU RC) is forced. If internal RC is selected in the option byte, the internal RC is provided. If AWU RC or external clock is selected, the AWU RC oscillator is provided.

A serial resistor must be connected to I^2C connector pin 6 in order to prevent contention on PA3/$\overline{\text{RESET}}$ pin. Contention may occur if a tool forces a state on $\overline{\text{RESET}}$ pin while PA3 pin forces the opposite state in output mode. The resistor value is defined to limit the current below 2 mA at 5 V. If PA3 is used as output push-pull, then the application must be switched off to allow the tool to take control of the $\overline{\text{RESET}}$ pin (PA3). To allow the programming tool to drive the $\overline{\text{RESET}}$ pin below V_{IL}, special care must also be taken when a pull-up is placed on PA3 for application reasons.

Caution: During normal operation, ICCCLK pin must be pulled- up, internally or externally (external pull-up of 10 kΩ mandatory in noisy environment). This is to avoid entering I^2C 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 I^2C interface



4.5 Memory protection

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

4.5.1 Readout protection

Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Program memory is protected.

In flash devices, this protection is removed by reprogramming the option. In this case, program memory is automatically erased, and the device can be reprogrammed.

Readout protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the option list.

4.5.2 Flash Write/Erase protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

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

Write/erase protection is enabled through the FMP_W bit in the option byte.

4.6 Related documentation

For details on Flash programming and I^2C protocol, refer to the ST7 Flash programming reference manual and to the ST7 I^2C protocol reference manual.

4.7 Register description

4.7.1 Flash Control/Status register (FCSR)

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

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

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

Reset value: 000 0000 (00h)

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

Table 4. FLASH register map and reset values

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

5 Central processing unit

5.1 Introduction

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

5.2 Main features

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

5.3 CPU registers

The six CPU registers shown in *Figure 7* are not present in the memory mapping and are accessed by specific instructions.

5.3.1 Accumulator (A)

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

5.3.2 Index registers (X and Y)

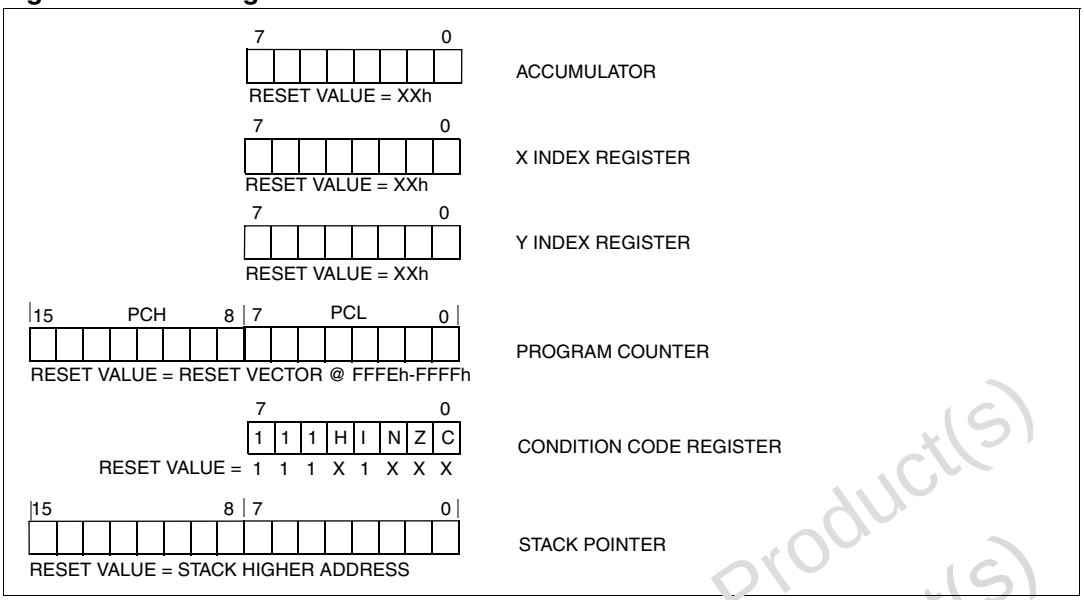
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The cross-assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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

5.3.3 Program counter (PC)

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

Figure 7. CPU registers



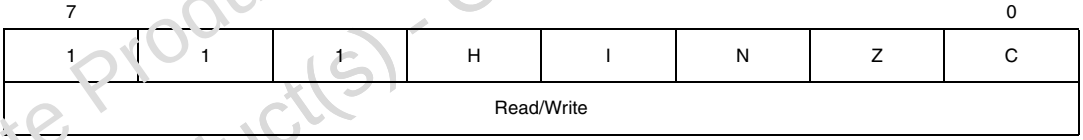
1. X = Undefined value

5.3.4 Condition Code register (CC)

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.

Reset value: 111x1xxx



Bit 7:5 Set to ‘1’

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^{th} 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.