



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





ST6215C ST6225C

8-bit MCUs with A/D converter,
two timers, oscillator safeguard & safe reset

■ Memories

- 2K or 4K bytes Program memory (OTP, EPROM, FASTROM or ROM) with read-out protection
- 64 bytes RAM

■ Clock, Reset and Supply Management

- Enhanced reset system
- Low Voltage Detector (LVD) for Safe Reset
- Clock sources: crystal/ceramic resonator or RC network, external clock, backup oscillator (LFAO)
- Oscillator Safeguard (OSG)
- 2 Power Saving Modes: Wait and Stop

■ Interrupt Management

- 4 interrupt vectors plus NMI and RESET
- 20 external interrupt lines (on 2 vectors)
- 1 external non-interrupt line

■ 20 I/O Ports

- 20 multifunctional bidirectional I/O lines
- 16 alternate function lines
- 4 high sink outputs (20mA)

■ 2 Timers

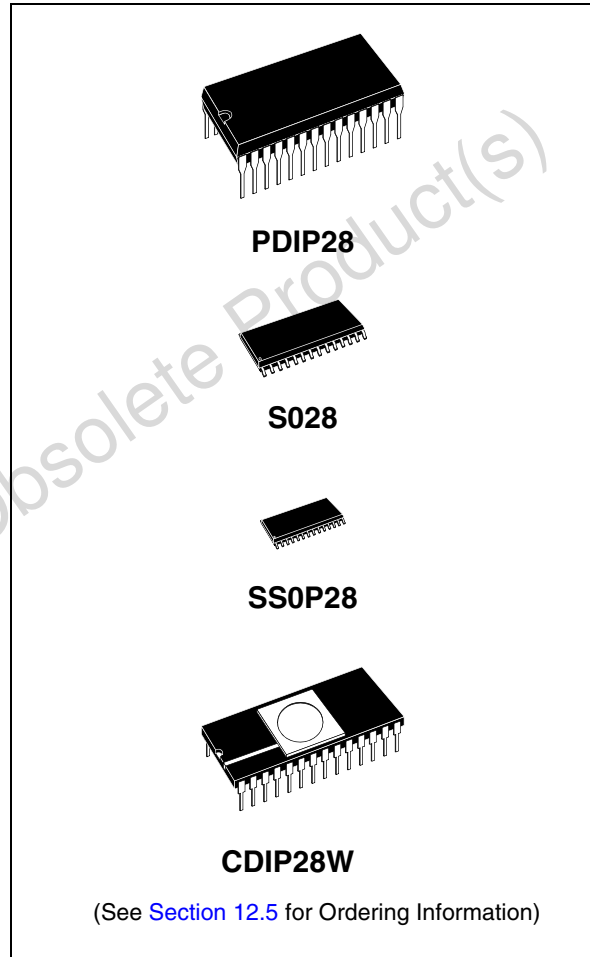
- Configurable watchdog timer
- 8-bit timer/counter with a 7-bit prescaler

■ Analog Peripheral

- 8-bit ADC with 16 input channels

■ Instruction Set

- 8-bit data manipulation
- 40 basic instructions
- 9 addressing modes
- Bit manipulation



■ Development Tools

- Full hardware/software development package

Device Summary

Features	ST6215C	ST6225C
Program memory - bytes	2K	4K
RAM - bytes	64	
Operating Supply	3.0V to 6V	
Clock Frequency	8MHz Max	
Operating Temperature	-40°C to +125°C	
Packages	PDIP28 / SO28 / SSOP28	

Table of Contents

1 INTRODUCTION	6
2 PIN DESCRIPTION	7
3 MEMORY MAPS, PROGRAMMING MODES AND OPTION BYTES	9
3.1 MEMORY AND REGISTER MAPS	9
3.1.1 Introduction	9
3.1.2 Program Space	11
3.1.3 Readout Protection	11
3.1.4 Data Space	11
3.1.5 Stack Space	11
3.1.6 Data ROM Window Mechanism	13
3.2 PROGRAMMING MODES	15
3.2.1 Program Memory	15
3.2.2 EPROM Erasing	15
3.3 OPTION BYTES	16
4 CENTRAL PROCESSING UNIT	17
4.1 INTRODUCTION	17
4.2 MAIN FEATURES	17
4.3 CPU REGISTERS	17
5 CLOCKS, SUPPLY AND RESET	19
5.1 CLOCK SYSTEM	19
5.1.1 Main Oscillator	20
5.1.2 Oscillator Safeguard (OSG)	21
5.1.3 Low Frequency Auxiliary Oscillator (LFAO)	22
5.1.4 Register Description	22
5.2 LOW VOLTAGE DETECTOR (LVD)	23
5.3 RESET	24
5.3.1 Introduction	24
5.3.2 RESET Sequence	24
5.3.3 RESET Pin	25
5.3.4 Watchdog Reset	26
5.3.5 LVD Reset	26
6 INTERRUPTS	27
6.1 INTERRUPT RULES AND PRIORITY MANAGEMENT	29
6.2 INTERRUPTS AND LOW POWER MODES	29
6.3 NON MASKABLE INTERRUPT	29
6.4 PERIPHERAL INTERRUPTS	29
6.5 EXTERNAL INTERRUPTS (I/O PORTS)	30
6.5.1 Notes on using External Interrupts	30
6.6 INTERRUPT HANDLING PROCEDURE	31
6.6.1 Interrupt Response Time	31
6.7 REGISTER DESCRIPTION	32
7 POWER SAVING MODES	33
7.1 INTRODUCTION	33

Table of Contents

7.2	WAIT MODE	34
7.3	STOP MODE	35
7.4	NOTES RELATED TO WAIT AND STOP MODES	37
7.4.1	Exit from Wait and Stop Modes	37
7.4.2	Recommended MCU Configuration	37
8	I/O PORTS	38
8.1	INTRODUCTION	38
8.2	FUNCTIONAL DESCRIPTION	38
8.2.1	Digital Input Modes	38
8.2.2	Analog Inputs	38
8.2.3	Output Modes	38
8.2.4	Alternate Functions	38
8.2.5	Instructions NOT to be used to access Port Data registers (SET, RES, INC and DEC)	40
8.2.6	Recommendations	40
8.3	LOW POWER MODES	40
8.4	INTERRUPTS	40
8.5	REGISTER DESCRIPTION	42
9	ON-CHIP PERIPHERALS	43
9.1	WATCHDOG TIMER (WDG)	43
9.1.1	Introduction	43
9.1.2	Main Features	43
9.1.3	Functional Description	44
9.1.4	Recommendations	44
9.1.5	Low Power Modes	45
9.1.6	Interrupts	45
9.1.7	Register Description	46
9.2	8-BIT TIMER	47
9.2.1	Introduction	47
9.2.2	Main Features	47
9.2.3	Counter/Prescaler Description	48
9.2.4	Functional Description	49
9.2.5	Low Power Modes	51
9.2.6	Interrupts	51
9.2.7	Register Description	52
9.3	A/D CONVERTER (ADC)	53
9.3.1	Introduction	53
9.3.2	Main Features	53
9.3.3	Functional Description	54
9.3.4	Recommendations	55
9.3.5	Low Power Modes	56
9.3.6	Interrupts	56
9.3.7	Register Description	56
10	INSTRUCTION SET	57
10.1	ST6 ARCHITECTURE	57
10.2	ADDRESSING MODES	57

Table of Contents

10.3	INSTRUCTION SET	58
11	ELECTRICAL CHARACTERISTICS	63
11.1	PARAMETER CONDITIONS	63
11.1.1	Minimum and Maximum Values	63
11.1.2	Typical Values	63
11.1.3	Typical Curves	63
11.1.4	Loading Capacitor	63
11.1.5	Pin Input Voltage	63
11.2	ABSOLUTE MAXIMUM RATINGS	64
11.2.1	Voltage Characteristics	64
11.2.2	Current Characteristics	64
11.2.3	Thermal Characteristics	64
11.3	OPERATING CONDITIONS	65
11.3.1	General Operating Conditions	65
11.3.2	Operating Conditions with Low Voltage Detector (LVD)	66
11.4	SUPPLY CURRENT CHARACTERISTICS	67
11.4.1	RUN Modes	67
11.4.2	WAIT Modes	68
11.4.3	STOP Mode	71
11.4.4	Supply and Clock System	72
11.4.5	On-Chip Peripherals	72
11.5	CLOCK AND TIMING CHARACTERISTICS	73
11.5.1	General Timings	73
11.5.2	External Clock Source	73
11.5.3	Crystal and Ceramic Resonator Oscillators	74
11.5.4	RC Oscillator	75
11.5.5	Oscillator Safeguard (OSG) and Low Frequency Auxiliary Oscillator (LFAO)	76
11.6	MEMORY CHARACTERISTICS	77
11.6.1	RAM and Hardware Registers	77
11.6.2	EPROM Program Memory	77
11.7	EMC CHARACTERISTICS	78
11.7.1	Functional EMS	78
11.7.2	Absolute Electrical Sensitivity	79
11.7.3	ESD Pin Protection Strategy	81
11.8	I/O PORT PIN CHARACTERISTICS	82
11.8.1	General Characteristics	82
11.8.2	Output Driving Current	83
11.9	CONTROL PIN CHARACTERISTICS	86
11.9.1	Asynchronous RESET Pin	86
11.9.2	NMI Pin	87
11.10	TIMER PERIPHERAL CHARACTERISTICS	88
11.10.1	Watchdog Timer	88
11.10.2	28-Bit Timer	88
11.11	8-BIT ADC CHARACTERISTICS	89
12	GENERAL INFORMATION	91
12.1	PACKAGE MECHANICAL DATA	91

Table of Contents

12.2	THERMAL CHARACTERISTICS	93
12.3	ECOPACK INFORMATION	94
12.4	PACKAGE/SOCKET FOOTPRINT PROPOSAL	95
12.5	ORDERING INFORMATION	96
12.6	TRANSFER OF CUSTOMER CODE	97
	12.6.1 FASTROM Version	97
	12.6.2 ROM Version	98
13	DEVELOPMENT TOOLS	100
14	ST6 APPLICATION NOTES	102
15	SUMMARY OF CHANGES	104
16	TO GET MORE INFORMATION	104

Obsolete Product(s) - Obsolete Product(s)

1 INTRODUCTION

The ST6215C, 25C devices are low cost members of the ST62xx 8-bit HCMOS family of microcontrollers, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST62E25C is the erasable EPROM version of the ST62T15C, T25C devices, which may be used during the development phase for the ST62T15C, T25C target devices, as well as the respective ST6215C, 25C ROM devices.

OTP and EPROM devices are functionally identical. OTP devices offer all the advantages of user programmability at low cost, which make them the ideal choice in a wide range of applications where frequent code changes, multiple code versions or last minute programmability are required.

The ROM based versions offer the same functionality, selecting the options defined in the program-

mable option bytes of the OTP/EPROM versions in the ROM option list (See Section 12.6 on page 97).

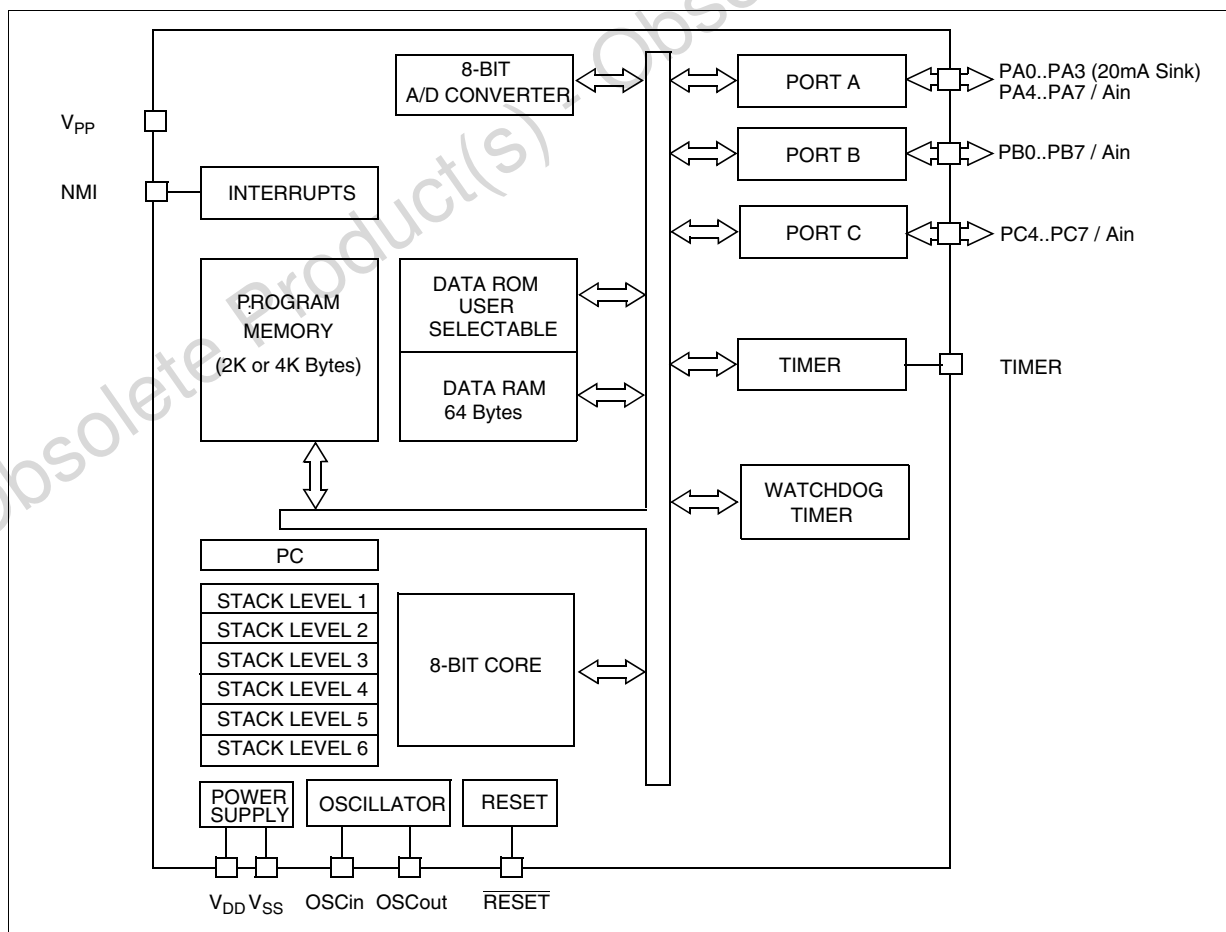
The ST62P15C/P25C are the **Factory Advanced Service Technique ROM (FASTROM)** versions of ST62T15C, T25C OTP devices.

They offer the same functionality as OTP devices, but they do not have to be programmed by the customer (See Section 12 on page 91).

These compact low-cost devices feature a Timer comprising an 8-bit counter with a 7-bit programmable prescaler, an 8-bit A/D Converter with 16 analog inputs and a Digital Watchdog timer, making them well suited for a wide range of automotive, appliance and industrial applications.

For easy reference, all parametric data is located in Section 11 on page 63.

Figure 1. Block Diagram



2 PIN DESCRIPTION

Figure 2. 28-Pin Package Pinout

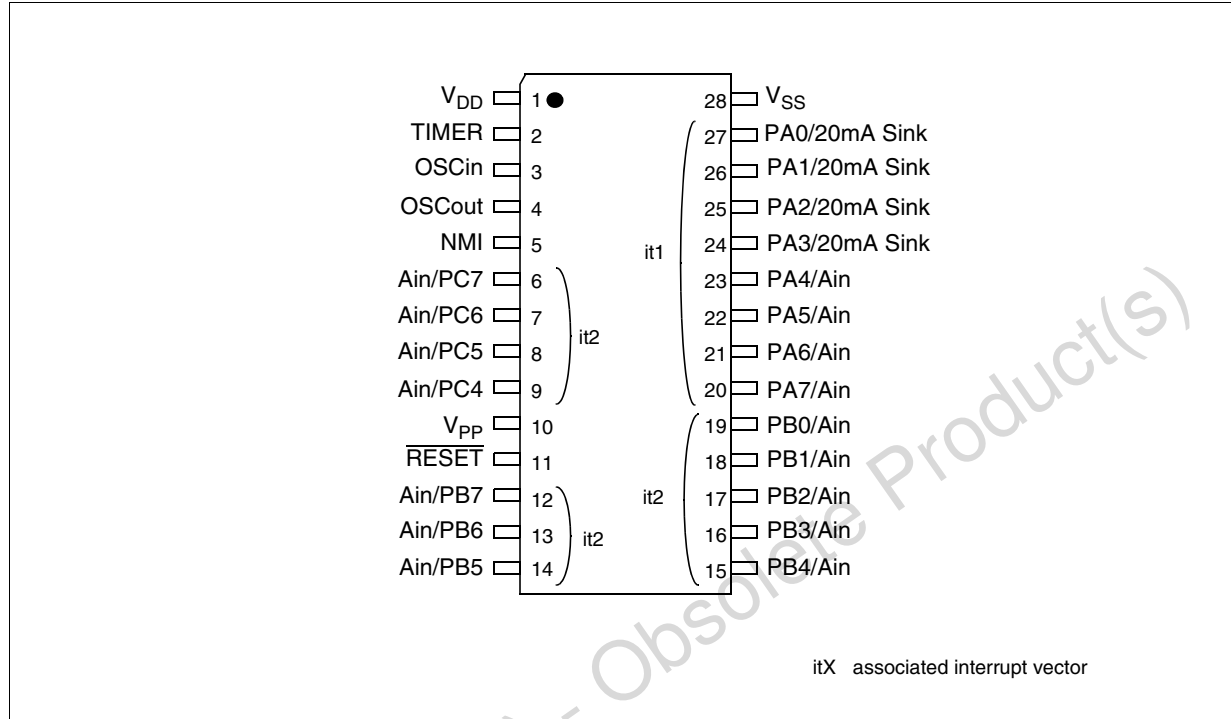


Table 1. Device Pin Description

Pin n°	Pin Name	Type	Main Function (after Reset)	Alternate Function
1	V _{DD}	S	Main power supply	
2	TIMER	I/O	Timer input or output	
3	OSCin	I	External clock input or resonator oscillator inverter input	
4	OSCout	O	Resonator oscillator inverter output or resistor input for RC oscillator	
5	NMI	I	Non maskable interrupt (falling edge sensitive)	
6	PC7/Ain	I/O	Pin C7 (IPU)	Analog input
7	PC6/Ain	I/O	Pin C6 (IPU)	Analog input
8	PC5/Ain	I/O	Pin C5 (IPU)	Analog input
9	PC4/Ain	I/O	Pin C4 (IPU)	Analog input
10	V _{PP}		Must be held at V _{SS} for normal operation, if a 12.5V level is applied to the pin during the reset phase, the device enters EPROM programming mode.	
11	RESET	I/O	Top priority non maskable interrupt (active low)	
12	PB7/Ain	I/O	Pin B7 (IPU)	Analog input
13	PB6/Ain	I/O	Pin B6 (IPU)	Analog input

Pin n°	Pin Name	Type	Main Function (after Reset)	Alternate Function
14	PB5/Ain	I/O	Pin B5 (IPU)	Analog input
15	PB4/Ain	I/O	Pin B4 (IPU)	Analog input
16	PB3/Ain	I/O	Pin B3 (IPU)	Analog input
17	PB2/Ain	I/O	Pin B2 (IPU)	Analog input
18	PB1/Ain	I/O	Pin B1 (IPU)	Analog input
19	PB0/Ain	I/O	Pin B0 (IPU)	Analog input
20	PA7/Ain	I/O	Pin A7 (IPU)	Analog input
21	PA6/Ain	I/O	Pin A6 (IPU)	Analog input
22	PA5/Ain	I/O	Pin A5 (IPU)	Analog input
23	PA4/Ain	I/O	Pin A4 (IPU)	Analog input
24	PA3/ 20mA Sink	I/O	Pin A3 (IPU)	
25	PA2/ 20mA Sink	I/O	Pin A2 (IPU)	
26	PA1/ 20mA Sink	I/O	Pin A1 (IPU)	
27	PA0/ 20mA Sink	I/O	Pin A0 (IPU)	
28	V _{SS}	S	Ground	

Legend / Abbreviations for Table 1:

I = input, O = output, S = supply, IPU = input pull-up

The input with pull-up configuration (reset state) is valid as long as the user software does not change it.

Refer to [Section 8 "I/O PORTS" on page 38](#) for more details on the software configuration of the I/O ports.

3 MEMORY MAPS, PROGRAMMING MODES AND OPTION BYTES

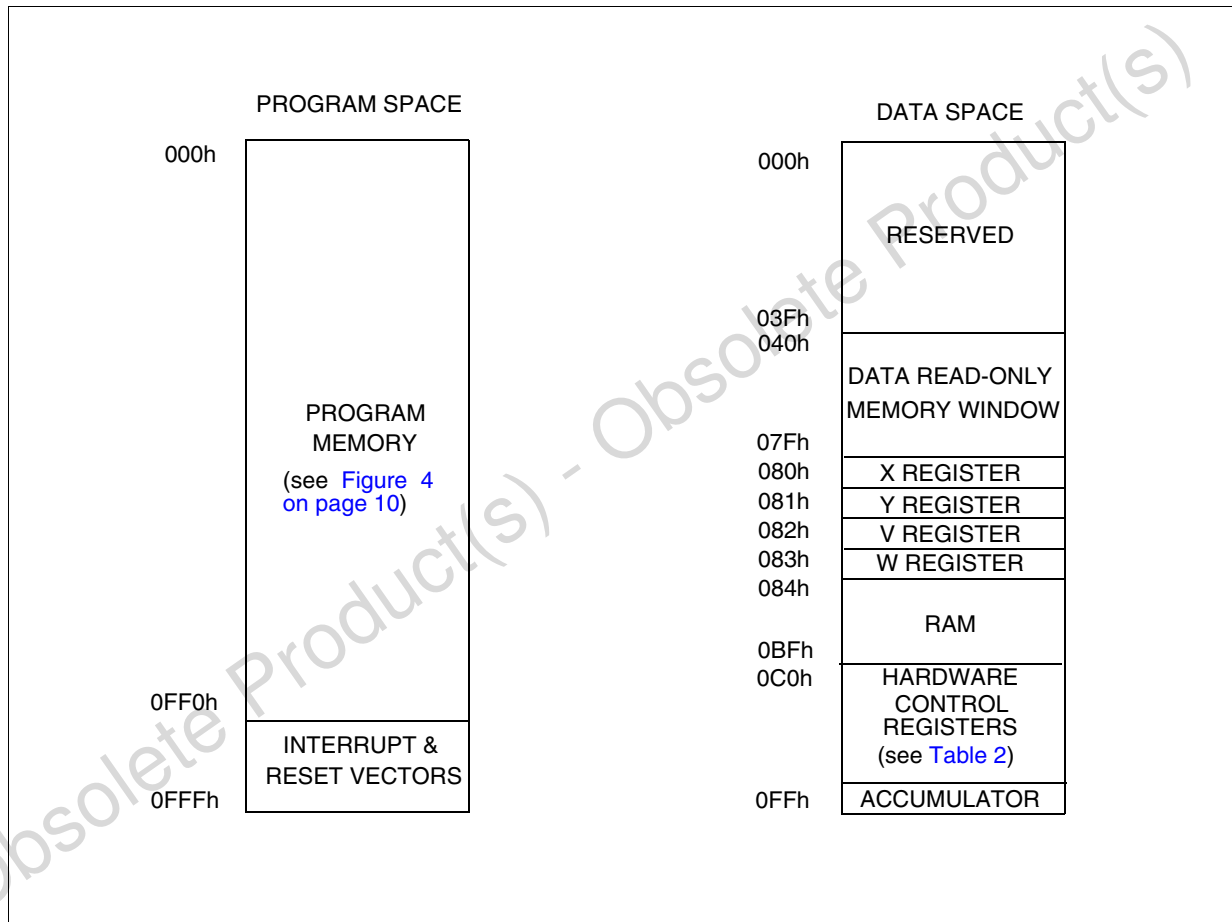
3.1 MEMORY AND REGISTER MAPS

3.1.1 Introduction

The MCU operates in three separate memory spaces: Program space, Data space, and Stack space. Operation in these three memory spaces is described in the following paragraphs.

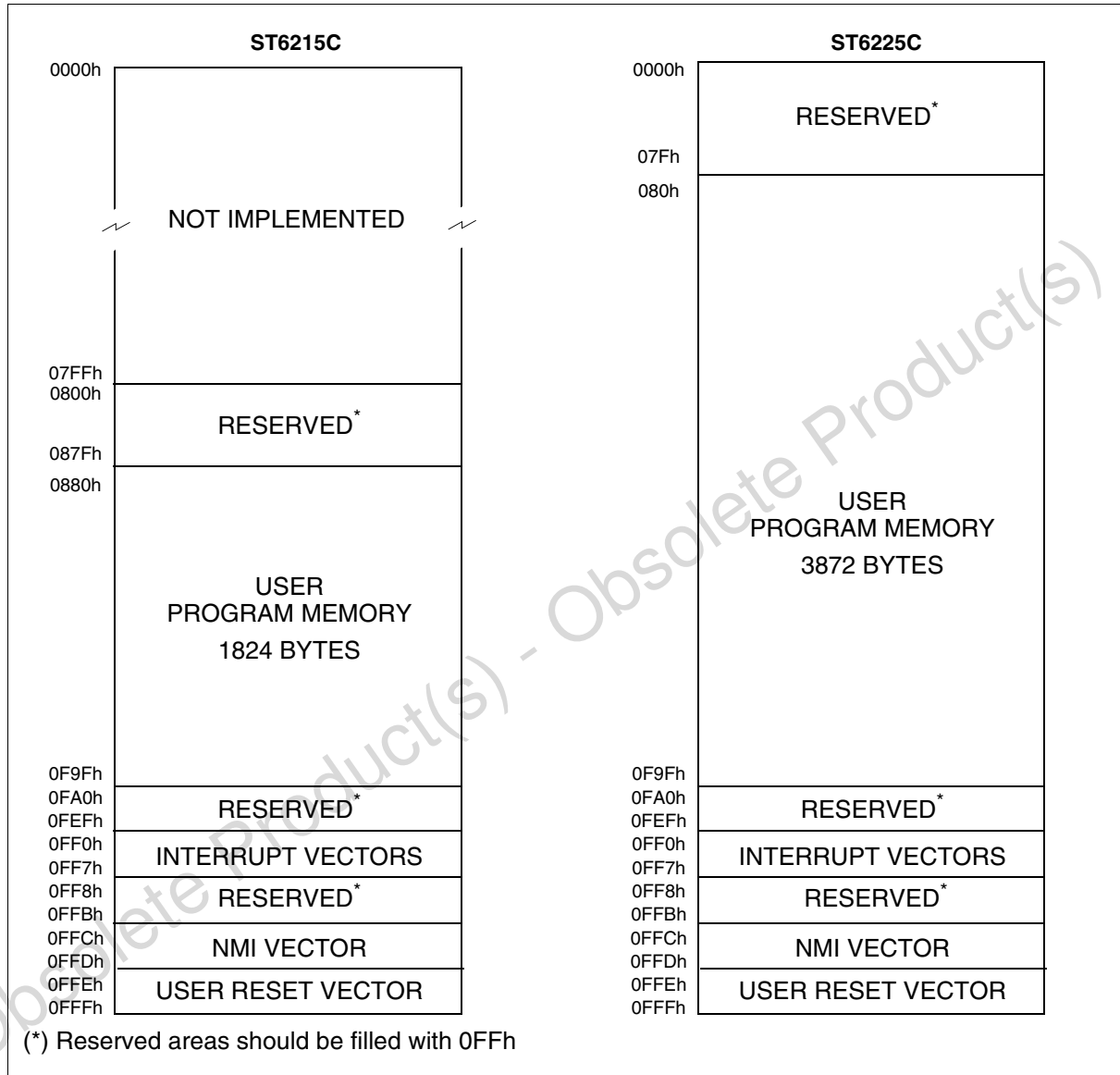
Briefly, Program space contains user program code in OTP and user vectors; Data space contains user data in RAM and in OTP, and Stack space accommodates six levels of stack for sub-routine and interrupt service routine nesting.

Figure 3. Memory Addressing Diagram



MEMORY MAP (Cont'd)

Figure 4. Program Memory Map



MEMORY MAP (Cont'd)

3.1.2 Program Space

Program Space comprises the instructions to be executed, the data required for immediate addressing mode instructions, the reserved factory test area and the user vectors. Program Space is addressed via the 12-bit Program Counter register (PC register). Thus, the MCU is capable of addressing 4K bytes of memory directly.

3.1.3 Readout Protection

The Program Memory in OTP or EPROM devices can be protected against external readout of memory by setting the Readout Protection bit in the option byte (Section 3.3 on page 16).

In the EPROM parts, Readout Protection option can be deactivated only by U.V. erasure that also results in the whole EPROM context being erased.

Note: Once the Readout Protection is activated, it is no longer possible, even for STMicroelectronics, to gain access to the OTP contents. Returned parts can therefore not be accepted if the Readout Protection bit is set.

3.1.4 Data Space

Data Space accommodates all the data necessary for processing the user program. This space comprises the RAM resource, the processor core and peripheral registers, as well as read-only data

such as constants and look-up tables in OTP/EPROM.

3.1.4.1 Data ROM

All read-only data is physically stored in program memory, which also accommodates the Program Space. The program memory consequently contains the program code to be executed, as well as the constants and look-up tables required by the application.

The Data Space locations in which the different constants and look-up tables are addressed by the processor core may be thought of as a 64-byte window through which it is possible to access the read-only data stored in OTP/EPROM.

3.1.4.2 Data RAM

The data space includes the user RAM area, the accumulator (A), the indirect registers (X), (Y), the short direct registers (V), (W), the I/O port registers, the peripheral data and control registers, the interrupt option register and the Data ROM Window register (DRWR register).

3.1.5 Stack Space

Stack space consists of six 12-bit registers which are used to stack subroutine and interrupt return addresses, as well as the current program counter contents.

MEMORY MAP (Cont'd)

Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
080h to 083h	CPU	X,Y,V,W	X,Y index registers V,W short direct registers	xxh	R/W
0C0h 0C1h 0C2h	I/O Ports	DRA ^{1) 2) 3)} DRB ^{1) 2) 3)} DRC ^{1) 2) 3)}	Port A Data Register Port B Data Register Port C Data Register	00h 00h 00h	R/W R/W R/W
0C3h	Reserved (1 Byte)				
0C4h 0C5h 0C6h	I/O Ports	DDRA ²⁾ DDRB ²⁾ DDRC ²⁾	Port A Direction Register Port B Direction Register Port C Direction Register	00h 00h 00h	R/W R/W R/W
0C7h	Reserved (1 Byte)				
0C8h	CPU	IOR	Interrupt Option Register	xxh	Write-only
0C9h	ROM	DRWR	Data ROM Window register	xxh	Write-only
0CAh 0CBh	Reserved (2 Bytes)				
0CCh 0CDh 0CEh	I/O Ports	ORA ²⁾ ORB ²⁾ ORC ²⁾	Port A Option Register Port B Option Register Port C Option Register	00h 00h 00h	R/W R/W R/W
0CFh	Reserved (1 byte)				
0D0h 0D1h	ADC	ADR ADCR	A/D Converter Data Register A/D Converter Control Register	xxh 40h	Read-only Ro/Wo
0D2h 0D3h 0D4h	Timer1	PSCR TCR TSCR	Timer 1 Prescaler Register Timer 1 Counter Register Timer 1 Status Control Register	7Fh 0FFh 00h	R/W R/W R/W
0D5h to 0D7h	Reserved (3 Bytes)				
0D8h	Watchdog Timer	WDGR	Watchdog Register	0FEh	R/W
0D9h to 0FEh	Reserved (38 Bytes)				
0FFh	CPU	A	Accumulator	xxh	R/W

Legend:

x = undefined, R/W = Read/Write, Ro = Read-only Bit(s) in the register, Wo = Write-only Bit(s) in the register.

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 be kept at their reset value.
3. Do not use single-bit instructions (SET, RES...) on Port Data Registers if any pin of the port is configured in input mode (refer to [Section 8 "I/O PORTS" on page 38](#) for more details).

MEMORY MAP (Cont'd)

3.1.6 Data ROM Window Mechanism

The Data read-only memory window is located from address 0040h to address 007Fh in Data space. It allows direct reading of 64 consecutive bytes located anywhere in program memory, between address 0000h and 0FFFh.

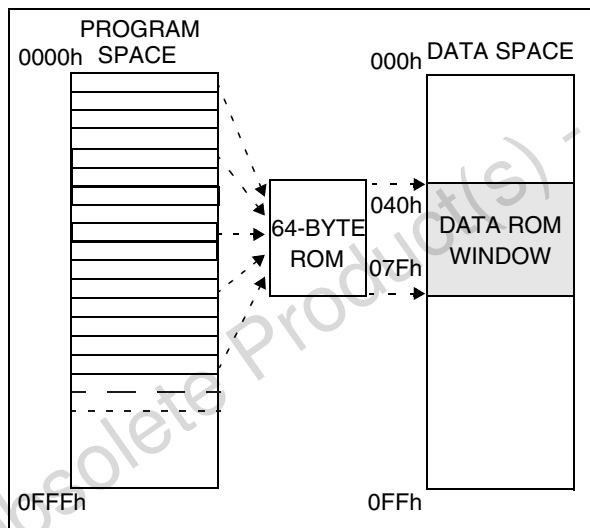
There are 64 blocks of 64 bytes in a 4K device:

- Block 0 is related to the address range 0000h to 003Fh.
- Block 1 is related to the address range 0040h to 007Fh.

and so on...

All the program memory can therefore be used to store either instructions or read-only data. The Data ROM window can be moved in steps of 64 bytes along the program memory by writing the appropriate code in the Data ROM Window Register (DRWR).

Figure 5. Data ROM Window



3.1.6.1 Data ROM Window Register (DRWR)

The DRWR can be addressed like any RAM location in the Data Space.

This register is used to select the 64-byte block of program memory to be read in the Data ROM window (from address 40h to address 7Fh in Data space). The DRWR register is not cleared on reset, therefore it must be written to before accessing the Data read-only memory window area for the first time.

Address: 0C9h — Write Only

Reset Value = xxh (undefined)

7								0
-	-	DRWR5	DRWR4	DRWR3	DRWR2	DRWR1	DRWR0	

Bits 7:6 = **Reserved**, must be cleared.

Bits 5:0 = **DRWR[5:0]** *Data read-only memory Window Register Bits*. These are the Data read-only memory Window bits that correspond to the upper bits of the data read-only memory space.

Caution: This register is undefined on reset, it is write-only, therefore do not read it nor access it using Read-Modify-Write instructions (SET, RES, INC and DEC).

MEMORY MAP (Cont'd)

3.1.6.2 Data ROM Window memory addressing

In cases where some data (look-up tables for example) are stored in program memory, reading these data requires the use of the Data ROM window mechanism. To do this:

1. The DRWR register has to be loaded with the 64-byte block number where the data are located (in program memory). This number also gives the start address of the block.
2. Then, the offset address of the byte in the Data ROM Window (corresponding to the offset in the 64-byte block in program memory) has to be loaded in a register (A, X,...).

When the above two steps are completed, the data can be read.

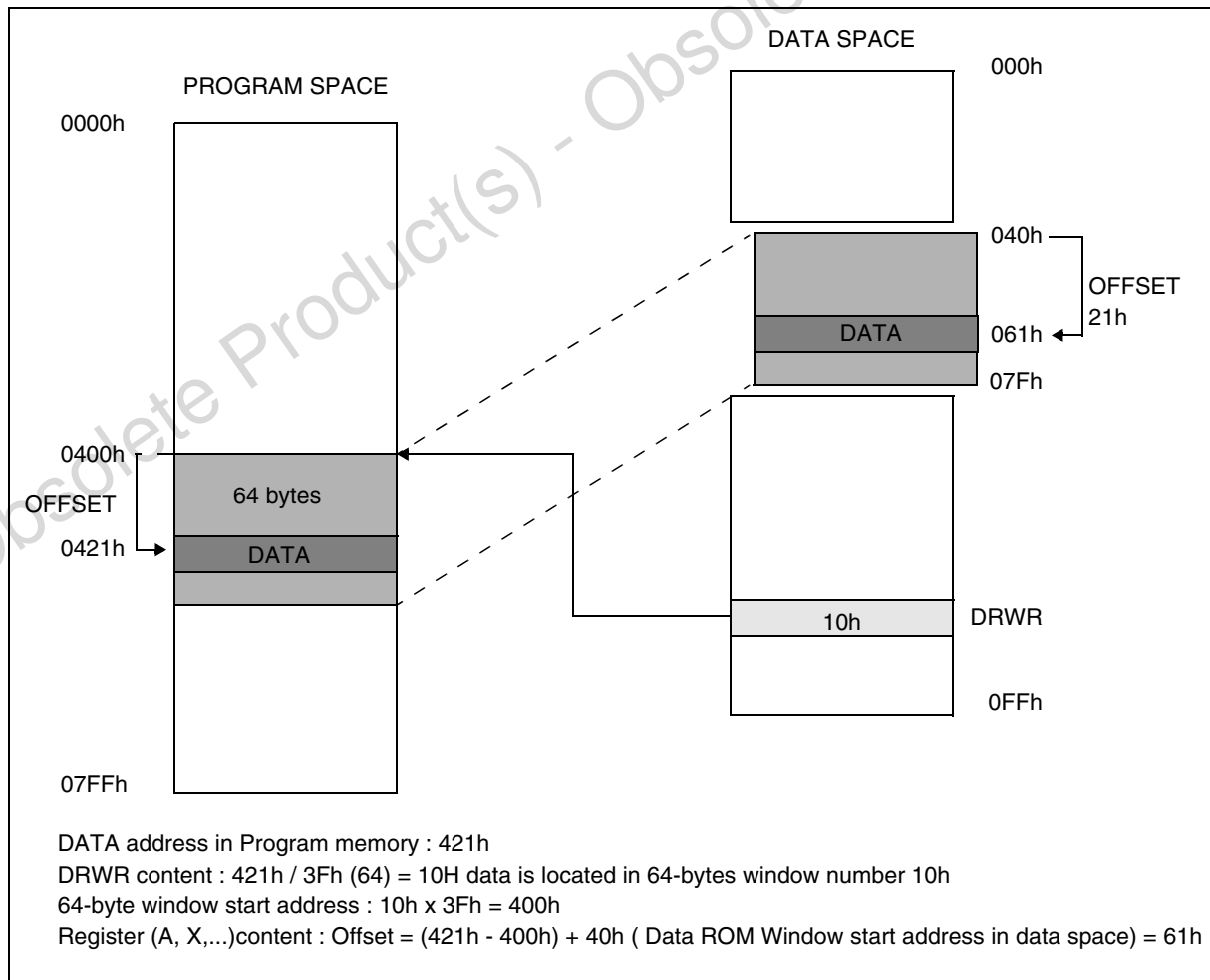
To understand how to determine the DRWR and the content of the register, please refer to the example shown in Figure 6. In any case the calculation is automatically handled by the ST6 development tools.

Please refer to the user manual of the corresponding tool.

3.1.6.3 Recommendations

Care is required when handling the DRWR register as it is write only. For this reason, the DRWR contents should not be changed while executing an interrupt service routine, as the service routine cannot save and then restore the register's previous contents. If it is impossible to avoid writing to the DRWR during the interrupt service routine, an image of the register must be saved in a RAM location, and each time the program writes to the DRWR, it must also write to the image register. The image register must be written first so that, if an interrupt occurs between the two instructions, the DRWR is not affected.

Figure 6. Data read-only memory Window Memory Addressing



3.2 PROGRAMMING MODES

3.2.1 Program Memory

EPROM/OTP programming mode is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming flow of the ST62T15C,T25C/E25C is described in the User Manual of the EPROM Programming Board.

Table 3. ST6215C Program Memory Map

Device Address	Description
0000h-087Fh	Reserved
0880h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Table 4. ST6225C Program Memory Map

Device Address	Description
0000h-007Fh	Reserved
0080h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Note: OTP/EPROM devices can be programmed with the development tools available from STMicroelectronics (please refer to [Section 13 on page 100](#)).

3.2.2 EPROM Erasing

The EPROM devices can be erased by exposure to Ultra Violet light. The characteristics of the MCU are such that erasure begins when the memory is exposed to light with a wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å.

It is thus recommended that the window of the MCU packages be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure is exposure to short wave ultraviolet light which have a wavelength 2537Å. The integrated dose (i.e. U.V. intensity x exposure time) for erasure should be a minimum of 30W-sec/cm². The erasure time with this dosage is approximately 30 to 40 minutes using an ultraviolet lamp with 12000µW/cm² power rating. The EPROM device should be placed within 2.5cm (1inch) of the lamp tubes during erasure.

3.3 OPTION BYTES

Each device is available for production in user programmable versions (OTP) as well as in factory coded versions (ROM). OTP devices are shipped to customers with a default content (00h), while ROM factory coded parts contain the code supplied by the customer. This implies that OTP devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

The two option bytes allow the hardware configuration of the microcontroller to be selected. The option bytes have no address in the memory map and can be accessed only in programming mode (for example using a standard ST6 programming tool).

In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see [Section 12.6.2 "ROM Version" on page 98](#)). It is therefore impossible to read the option bytes.

The option bytes can be only programmed once. It is not possible to change the selected options after they have been programmed.

In order to reach the power consumption value indicated in [Section 11.4](#), the option byte must be programmed to its default value. Otherwise, an over-consumption will occur.

MSB OPTION BYTE

Bits 15:10 = **Reserved**, must be always cleared.

Bit 9 = **EXTCNTL** *External STOP MODE control*.
 0: EXTCNTL mode not available. STOP mode is not available with the watchdog active.
 1: EXTCNTL mode available. STOP mode is available with the watchdog active by setting NMI pin to one.

Bit 8 = **LVD** *Low Voltage Detector on/off*.
 This option bit enable or disable the Low Voltage Detector (LVD) feature.

0: Low Voltage Detector disabled
 1: Low Voltage Detector enabled.

LSB OPTION BYTE

Bit 7 = **PROTECT** *Readout Protection*.
 This option bit enables or disables external access to the internal program memory.
 0: Program memory not read-out protected
 1: Program memory read-out protected

Bit 6 = **OSC** *Oscillator selection*.
 This option bit selects the main oscillator type.
 0: Quartz crystal, ceramic resonator or external clock
 1: RC network

Bits 5:4 = **Reserved**, must be always cleared.

Bit 3 = **NMI PULL** *NMI Pull-Up on/off*.
 This option bit enables or disables the internal pull-up on the NMI pin.
 0: Pull-up disabled
 1: Pull-up enabled

Bit 2 = **TIM PULL** *TIMER Pull-Up on/off*.
 This option bit enables or disables the internal pull-up on the TIMER pin.
 0: Pull-up disabled
 1: Pull-up enabled

Bit 1 = **WDACT** *Hardware or software watchdog*.
 This option bit selects the watchdog type.
 0: Software (watchdog to be enabled by software)
 1: Hardware (watchdog always enabled)

Bit 0 = **OSGEN** *Oscillator Safeguard on/off*.
 This option bit enables or disables the oscillator Safeguard (OSG) feature.
 0: Oscillator Safeguard disabled
 1: Oscillator Safeguard enabled

	MSB OPTION BYTE								LSB OPTION BYTE								
	15							8	7	0							
	Reserved							EXT CTL	LVD	PROTECT	OSC	Res.	Res.	NMI PULL	TIM PULL	WD ACT	OSG EN
Default Value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

4 CENTRAL PROCESSING UNIT

4.1 INTRODUCTION

The CPU Core of ST6 devices is independent of the I/O or Memory configuration. As such, it may be thought of as an independent central processor communicating with on-chip I/O, Memory and Peripherals via internal address, data, and control buses.

4.2 MAIN FEATURES

- 40 basic instructions
- 9 main addressing modes
- Two 8-bit index registers
- Two 8-bit short direct registers
- Low power modes
- Maskable hardware interrupts
- 6-level hardware stack

4.3 CPU REGISTERS

The ST6 Family CPU core features six registers and three pairs of flags available to the programmer. These are described in the following paragraphs.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipula-

tions. The accumulator can be addressed in Data Space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data Space.

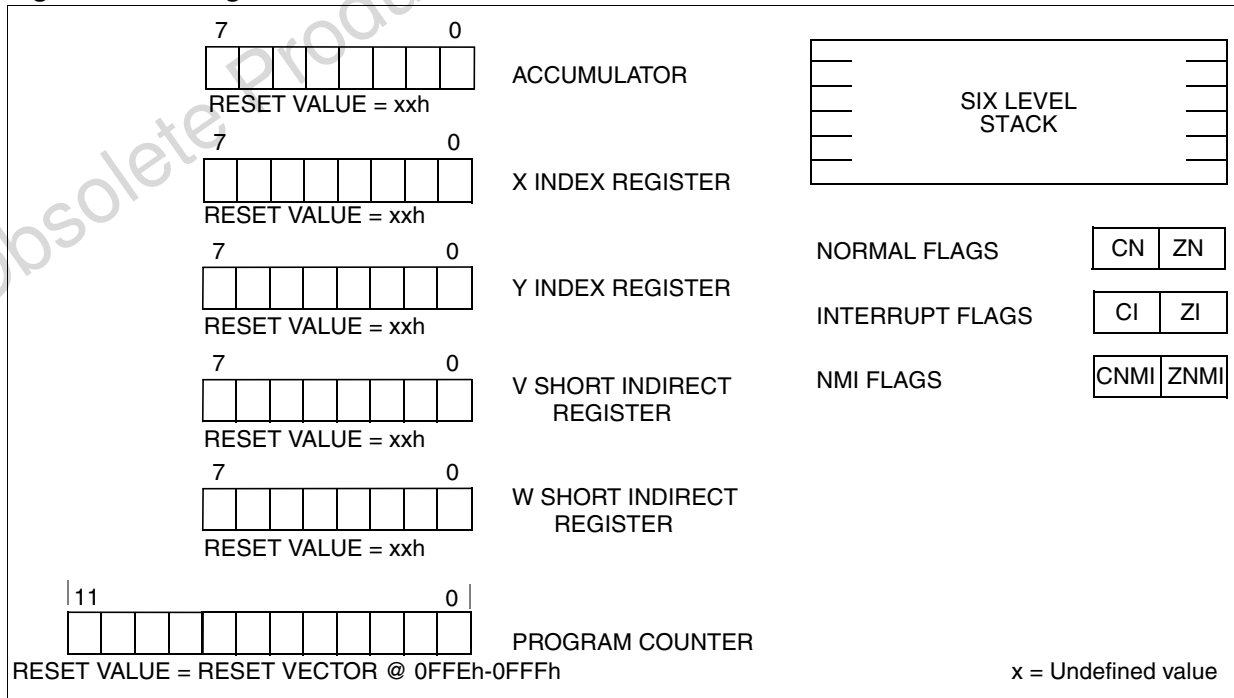
Index Registers (X, Y). These two registers are used in Indirect addressing mode as pointers to memory locations in Data Space. They can also be accessed in Direct, Short Direct, or Bit Direct addressing modes. They are mapped in Data Space at addresses 80h (X) and 81h (Y) and can be accessed like any other memory location.

Short Direct Registers (V, W). These two registers are used in Short Direct addressing mode. This means that the data stored in V or W can be accessed with a one-byte instruction (four CPU cycles). V and W can also be accessed using Direct and Bit Direct addressing modes. They are mapped in Data Space at addresses 82h (V) and 83h (W) and can be accessed like any other memory location.

Note: The X and Y registers can also be used as Short Direct registers in the same way as V and W.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next instruction to be executed by the core. This ROM location may be an opcode, an operand, or the address of an operand.

Figure 7. CPU Registers



CPU REGISTERS (Cont'd)

The 12-bit length allows the direct addressing of 4096 bytes in Program Space.

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program ROM Page register.

The PC value is incremented after reading the address of the current instruction. To execute relative jumps, the PC and the offset are shifted through the ALU, where they are added; the result is then shifted back into the PC. The program counter can be changed in the following ways:

- JP (Jump) instruction PC = Jump address
- CALL instruction PC = Call address
- Relative Branch Instruction PC = PC +/- offset
- Interrupt PC = Interrupt vector
- Reset PC = Reset vector
- RET & RETI instructions PC = Pop (stack)
- Normal instruction PC = PC + 1

Flags (C, Z). The ST6 CPU includes three pairs of flags (Carry and Zero), each pair being associated with one of the three normal modes of operation: Normal mode, Interrupt mode and Non Maskable Interrupt mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during Normal operation, another pair is used during Interrupt mode (CI, ZI), and a third pair is used in the Non Maskable Interrupt mode (CNMI, ZNMI).

The ST6 CPU uses the pair of flags associated with the current mode: as soon as an interrupt (or a Non Maskable Interrupt) is generated, the ST6 CPU uses the Interrupt flags (or the NMI flags) instead of the Normal flags. When the RETI instruction is executed, the previously used set of flags is restored. It should be noted that each flag set can only be addressed in its own context (Non Maskable Interrupt, Normal Interrupt or Main routine). The flags are not cleared during context switching and thus retain their status.

C : Carry flag.

This bit is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

- 0: No carry has occurred
- 1: A carry has occurred

Z : Zero flag

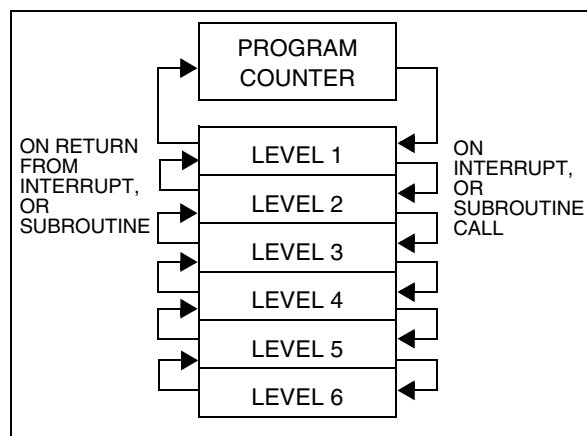
This flag is set if the result of the last arithmetic or logical operation was equal to zero; otherwise it is cleared.

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

Switching between the three sets of flags is performed automatically when an NMI, an interrupt or a RETI instruction occurs. As NMI mode is automatically selected after the reset of the MCU, the ST6 core uses the NMI flags first.

Stack. The ST6 CPU includes a true LIFO (Last In First Out) hardware stack which eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level are shifted into the next level down, while the content of the PC is shifted into the first level (the original contents of the sixth stack level are lost). When a subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is popped back into the previous level.

Figure 8. Stack manipulation



Since the accumulator, in common with all other data space registers, is not stored in this stack, management of these registers should be performed within the subroutine.

Caution: The stack will remain in its “deepest” position if more than 6 nested calls or interrupts are executed, and consequently the last return address will be lost.

It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

5 CLOCKS, SUPPLY AND RESET

5.1 CLOCK SYSTEM

The main oscillator of the MCU can be driven by any of these clock sources:

- external clock signal
- external AT-cut parallel-resonant crystal
- external ceramic resonator
- external RC network (R_{NET}).

In addition, an on-chip Low Frequency Auxiliary Oscillator (LFAO) is available as a back-up clock system or to reduce power consumption.

An optional Oscillator Safeguard (OSG) filters spikes from the oscillator lines, and switches to the LFAO backup oscillator in the event of main oscillator failure. It also automatically limits the internal clock frequency (f_{INT}) as a function of V_{DD} , in order to guarantee correct operation. These functions are illustrated in [Figure 10](#), and [Figure 11](#).

[Table 5](#) illustrates various possible oscillator configurations using an external crystal or ceramic resonator, an external clock input, an external resistor (R_{NET}), or the lowest cost solution using only the LFAO.

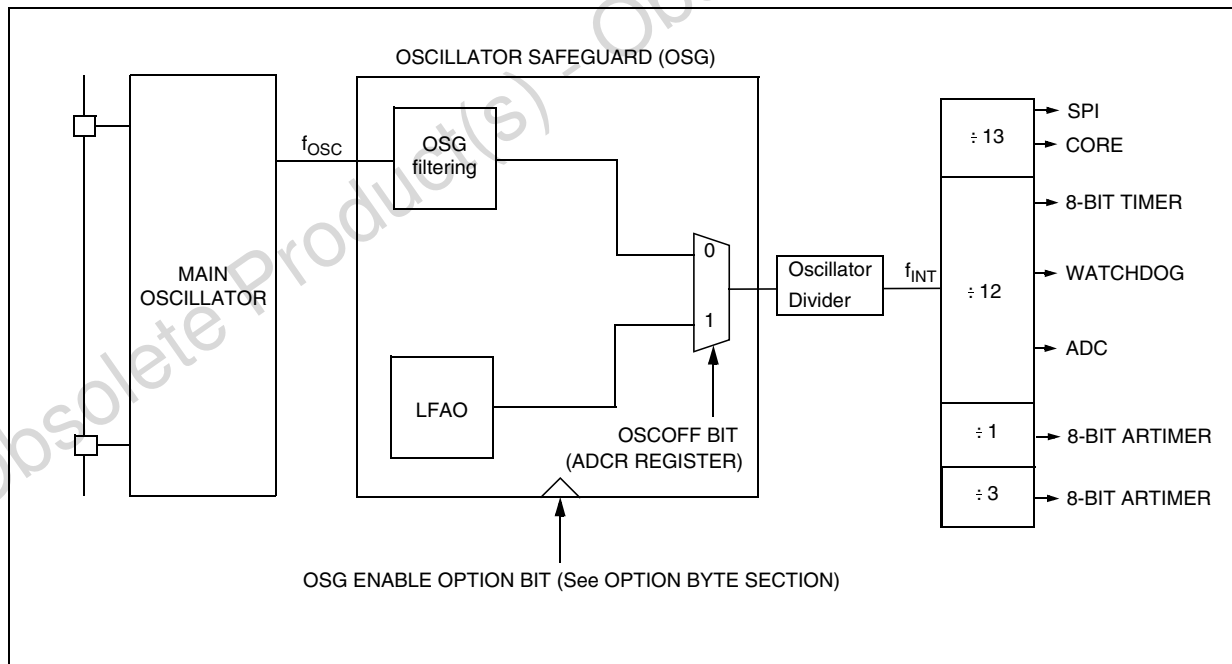
For more details on configuring the clock options, refer to the Option Bytes section of this document.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the Watchdog timer and the A/D converter, by 13 to drive the CPU core and the SPI and by 1 or 3 to drive the ARTIMER, as shown in [Figure 9](#).

With an 8 MHz oscillator, the fastest CPU cycle is therefore 1.625 μ s.

A CPU cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five CPU cycles for execution.

Figure 9. Clock Circuit Block Diagram



CLOCK SYSTEM (Cont'd)

5.1.1 Main Oscillator

The oscillator configuration is specified by selecting the appropriate option in the option bytes (refer to the Option Bytes section of this document). When the CRYSTAL/RESONATOR option is selected, it must be used with a quartz crystal, a ceramic resonator or an external signal provided on the OSCin pin. When the RC NETWORK option is selected, the system clock is generated by an external resistor (the capacitor is implemented internally).

The main oscillator can be turned off (when the OSG ENABLED option is selected) by setting the OSCOFF bit of the ADC Control Register (not available on some devices). This will automatically start the Low Frequency Auxiliary Oscillator (LFAO).

The main oscillator can be turned off by resetting the OSCOFF bit of the A/D Converter Control Register or by resetting the MCU. When the main oscillator starts there is a delay made up of the oscillator start-up delay period plus the duration of the software instruction at a clock frequency f_{LFAO} .

Caution: It should be noted that when the RC network option is selected, the accuracy of the frequency is about 20% so it may not be suitable for some applications (For more details, please refer to the Electrical Characteristics Section).

Table 5. Oscillator Configurations

	Hardware Configuration
Crystal/Resonator Option ¹⁾	<p>External Clock</p>
Crystal/Resonator Option ¹⁾	<p>Crystal/Resonator Clock²⁾</p>
RC Network Option ¹⁾	<p>RC Network</p>
OSG Enabled Option ¹⁾	<p>LFAO</p>

Notes:

1. To select the options shown in column 1 of the above table, refer to the Option Byte section.
2. This schematic are given for guidance only and are subject to the schematics given by the crystal or ceramic resonator manufacturer.
3. For more details, please refer to the Electrical Characteristics Section.

CLOCK SYSTEM (Cont'd)**5.1.2 Oscillator Safeguard (OSG)**

The Oscillator Safeguard (OSG) feature is a means of dramatically improving the operational integrity of the MCU. It is available when the OSG ENABLED option is selected in the option byte (refer to the Option Bytes section of this document).

The OSG acts as a filter whose cross-over frequency is device dependent and provides three basic functions:

- Filtering spikes on the oscillator lines which would result in driving the CPU at excessive frequencies
- Management of the Low Frequency Auxiliary Oscillator (LFAO), (useable as low cost internal clock source, backup clock in case of main oscillator failure or for low power consumption)
- Automatically limiting the f_{INT} clock frequency as a function of supply voltage, to ensure correct operation even if the power supply drops.

5.1.2.1 Spike Filtering

Spikes on the oscillator lines result in an effectively increased internal clock frequency. In the absence of an OSG circuit, this may lead to an over frequency for a given power supply voltage. The OSG filters out such spikes (as illustrated in [Figure 10](#)). In all cases, when the OSG is active, the maximum

internal clock frequency, f_{INT} , is limited to f_{OSG} , which is supply voltage dependent.

5.1.2.2 Management of Supply Voltage Variations

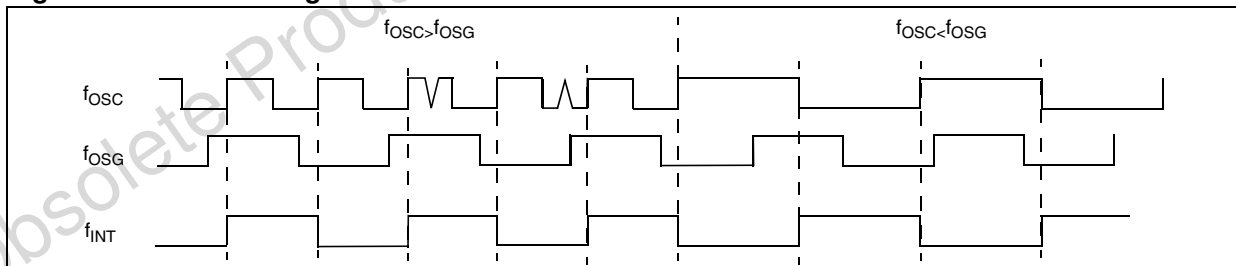
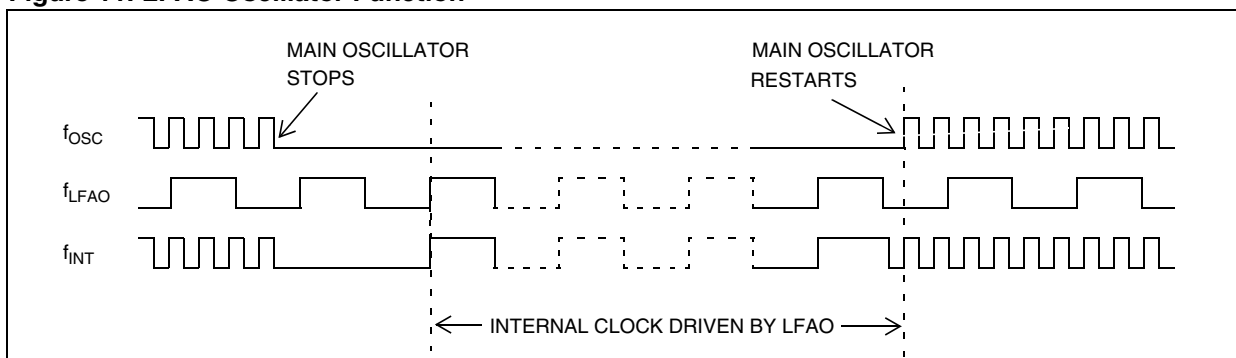
Over-frequency, at a given power supply level, is seen by the OSG as spikes; it therefore filters out some cycles in order that the internal clock frequency of the device is kept within the range the particular device can stand (depending on V_{DD}), and below f_{OSG} : the maximum authorised frequency with OSG enabled.

5.1.2.3 LFAO Management

When the OSG is enabled, the Low Frequency Auxiliary Oscillator can be used (see [Section 5.1.3](#)).

Note: The OSG should be used wherever possible as it provides maximum security for the application. It should be noted however, that it can increase power consumption and reduce the maximum operating frequency to f_{OSG} (see Electrical Characteristics section).

Caution: Care has to be taken when using the OSG, as the internal frequency is defined between a minimum and a maximum value and may vary depending on both V_{DD} and temperature. For precise timing measurements, it is not recommended to use the OSG.

Figure 10. OSG Filtering Function**Figure 11. LFAO Oscillator Function**

CLOCK SYSTEM (Cont'd)

5.1.3 Low Frequency Auxiliary Oscillator (LFAO)

The Low Frequency Auxiliary Oscillator has three main purposes. Firstly, it can be used to reduce power consumption in non timing critical routines. Secondly, it offers a fully integrated system clock, without any external components. Lastly, it acts as a backup oscillator in case of main oscillator failure.

This oscillator is available when the OSG ENABLED option is selected in the option byte (refer to the Option Bytes section of this document). In this case, it automatically starts one of its periods after the first missing edge of the main oscillator, whatever the reason for the failure (main oscillator defective, no clock circuitry provided, main oscillator switched off...). See [Figure 11](#).

User code, normal interrupts, WAIT and STOP instructions, are processed as normal, at the reduced f_{LFAO} frequency. The A/D converter accuracy is decreased, since the internal frequency is below 1.2 MHz.

At power on, until the main oscillator starts, the reset delay counter is driven by the LFAO. If the main oscillator starts before the 2048 cycle delay has elapsed, it takes over.

The Low Frequency Auxiliary Oscillator is automatically switched off as soon as the main oscillator starts.

5.1.4 Register Description

ADC CONTROL REGISTER (ADCR)

Address: 0D1h — Read/Write

Reset value: 0100 0000 (40h)

7						0	
ADCR 7	ADCR 6	ADCR 5	ADCR 4	ADCR 3	OSC OFF	ADCR 1	ADCR 0

Bit 7:3, 1:0 = **ADCR[7:3]**, **ADCR[1:0]** *ADC Control Register*.

These bits are used to control the A/D converter (if available on the device) otherwise they are not used.

Bit 2 = **OSCOFF** *Main Oscillator Off*.

0: Main oscillator enabled

1: Main oscillator disabled

Note: The OSG must be enabled using the OSGEN option in the Option Byte, otherwise the OSCOFF setting has no effect.

Obsolete Product(s)

5.2 LOW VOLTAGE DETECTOR (LVD)

The on-chip Low Voltage Detector is enabled by setting a bit in the option bytes (refer to the Option Bytes section of this document).

The LVD allows the device to be used without any external RESET circuitry. In this case, the RESET pin should be left unconnected.

If the LVD is not used, an external circuit is mandatory to ensure correct Power On Reset operation, see figure in the Reset section. For more details, please refer to the application note AN669.

The LVD generates a static Reset when the supply voltage is below a reference value. This means that it secures the power-up as well as the power-down keeping the ST6 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 12](#).

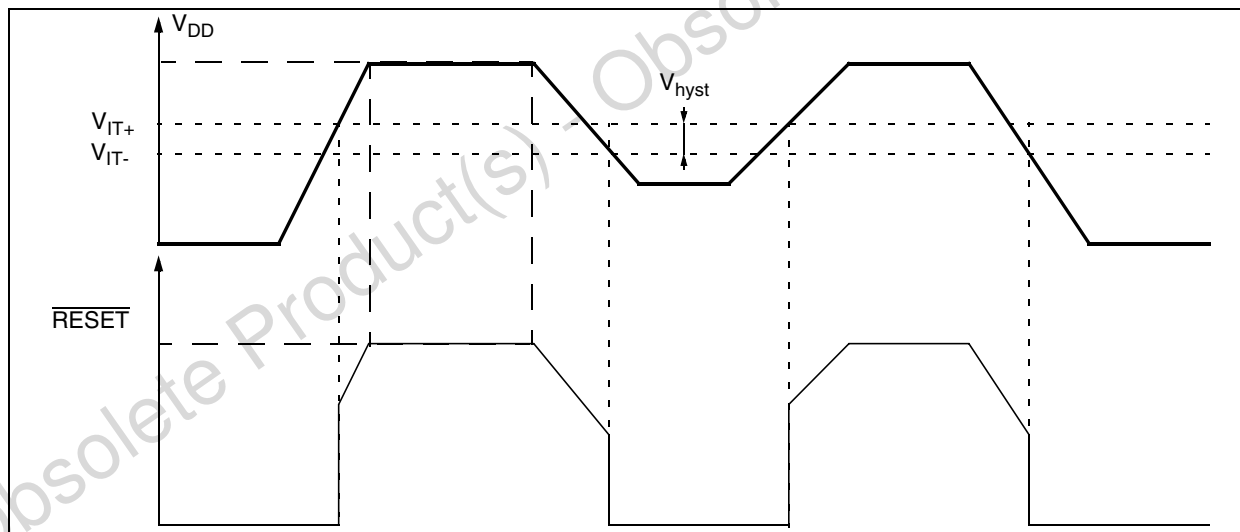
If the LVD is enabled, the MCU can be in only one of two states:

- Over the input threshold voltage, it is running under full software control
- Below the input threshold voltage, it is in static safe reset

In these conditions, secure operation is guaranteed 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.

Figure 12. Low Voltage Detector Reset



5.3 RESET

5.3.1 Introduction

The MCU can be reset in three ways:

- A low pulse input on the $\overline{\text{RESET}}$ pin
- Internal Watchdog reset
- Internal Low Voltage Detector (LVD) reset

5.3.2 RESET Sequence

The basic RESET sequence consists of 3 main phases:

- Internal (watchdog or LVD) or external Reset event
- A delay of 2048 clock (f_{INT}) cycles
- RESET vector fetch

The reset delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state.

The RESET vector fetch phase duration is 2 clock cycles.

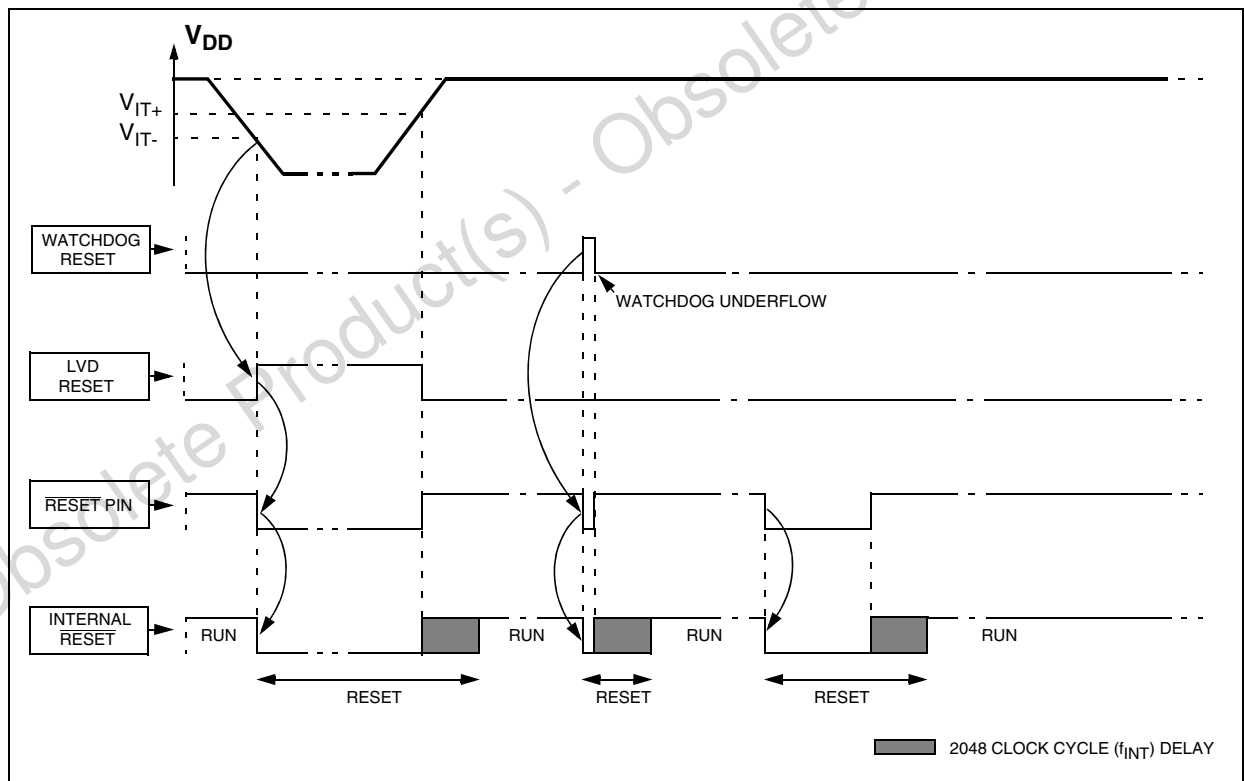
When a reset occurs:

- The stack is cleared
- The PC is loaded with the address of the Reset vector. It is located in program ROM starting at address 0FFEh.

A jump to the beginning of the user program must be coded at this address.

- The interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode. This prevents the initialization routine from being interrupted. The initialization routine should therefore be terminated by a RETI instruction, in order to go back to normal mode.

Figure 13. RESET Sequence



RESET (Cont'd)**5.3.3 $\overline{\text{RESET}}$ Pin**

The $\overline{\text{RESET}}$ pin may be connected to a device on the application board in order to reset the MCU if required. The $\overline{\text{RESET}}$ pin may be pulled low in RUN, WAIT or STOP mode. This input can be used to reset the internal state of the MCU and ensure it starts-up correctly. The pin, which is connected to an internal pull-up, is active low and features a Schmitt trigger input. A delay (2048 clock cycles) added to the external signal ensures that even short pulses on the $\overline{\text{RESET}}$ pin are accepted as valid, provided V_{DD} has completed its rising phase and that the oscillator is running correctly (normal RUN or WAIT modes). The MCU is kept in the Reset state as long as the $\overline{\text{RESET}}$ pin is held low.

If the $\overline{\text{RESET}}$ pin is grounded while the MCU is in RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the I/O ports are configured as inputs with pull-up resistors and the main oscillator is restarted. When the level on the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed at the end of the internal delay period.

If the $\overline{\text{RESET}}$ pin is grounded while the MCU is in STOP mode, the oscillator starts up and all the I/O ports are configured as inputs with pull-up resistors. When the $\overline{\text{RESET}}$ pin level then goes high, the initialization sequence is executed at the end of the internal delay period.

A simple external $\overline{\text{RESET}}$ circuitry is shown in [Figure 15](#). For more details, please refer to the application note AN669.

Figure 14. Reset Block Diagram