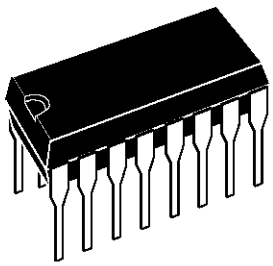
SGS-THOMSON MICROELECTRONICS

ST62T00
ST62T01, E01

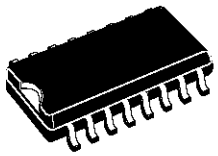
8-BIT OTP/EPROM MCUs WITH A/D CONVERTER

PRODUCT PREVIEW

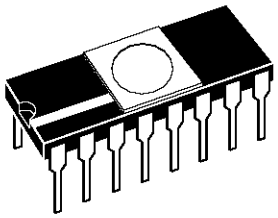
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in OTP/EPROM
- Data OTP/EPROM: User selectable size (in program OTP/EPROM)
- Data RAM: 64 bytes
- 9 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 3 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer with 7-bit programmable prescaler
- Digital Watchdog
- 8-bit A/D Converter with 4 analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal or Ceramic resonator
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP16



PSO16



CDIP16W

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	EPROM (Bytes)	OTP (Bytes)	I/O Pins
ST62T00		1036	9
ST62T01		1836	9
ST62E01	1836		9

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T00, T01 and E01 devices are low cost members of the 8-bit HCMOS ST62xx 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 ST62E01 device is an erasable EPROM version of the ST62T01 device, which may be used to emulate the T00 and T01, as well as the respective ST6200B and 01B ROM based devices.

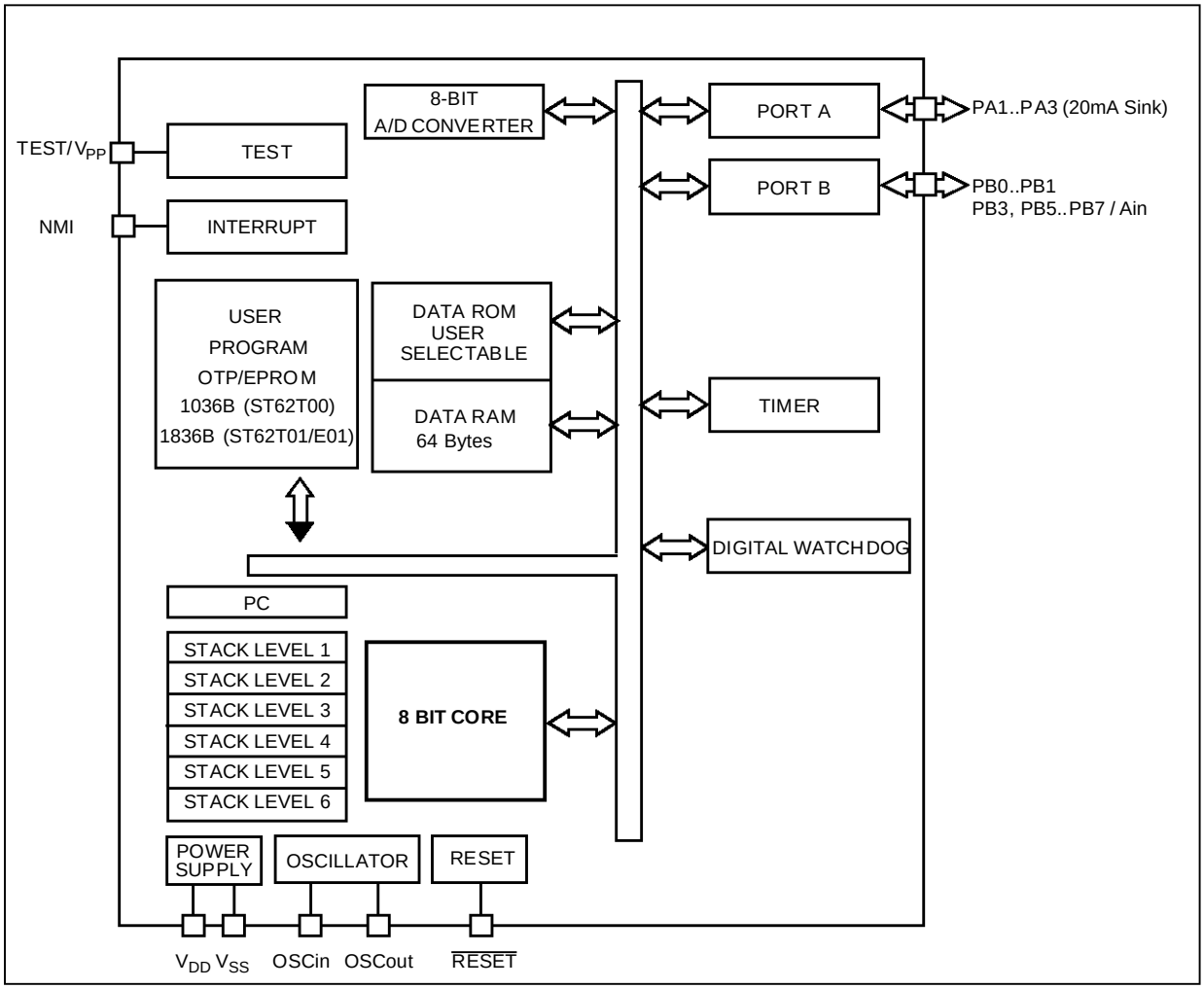
OTP and EPROM devices are functionally identical. The ROM based versions offer the following additional features: RC Oscillator, Oscillator Safeguard, external Stop mode control, program code readout protection and the possibility of having an internal pullup on the NMI pin.

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.

EPROM devices, thanks to their ease of erasure and reprogrammability, are best suited for program development and evaluation.

These compact low-cost devices feature a Timer comprising an 8-bit counter and a 7-bit programmable prescaler, an 8-bit A/D Converter with 4 analog inputs and a Digital Watchdog timer, making them well suited for a wide range of automotive, appliance and industrial applications.

Figure 1. Block Diagram



1.2 PIN DESCRIPTIONS

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

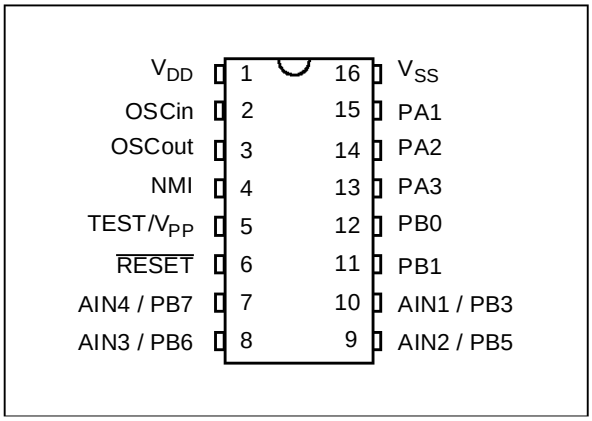
TEST/V_{PP}. This pin must be held at V_{SS} for normal operation (an internal 100kΩ pull-down resistor selects normal operating mode if the TEST pin is not connected externally). The OTP/EPROM programming mode is entered by connecting this pin to a 12.5V level during the Reset phase.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. A pull-up device must be provided externally on OTP and EPROM devices.

PA1-PA3. These 3 lines are organized as an I/O port (A). Each line may be configured under software control as an input with or without internal pull-up resistors, as an interrupt generating input with pull-up resistors, or as an open-drain or push-pull output. PA1-PA3 can sink up to 20mA for direct LED drive capability.

PB0, PB1, PB3, PB5-PB7. These 6 lines are organized as one I/O port (B). Each line may be configured under software control as an input with or without internal pull-up, as an interrupt generating input with pull-up, or as an open-drain or push-pull output. PB5-PB7 and PB3 may also be used as analog inputs to the A/D converter.

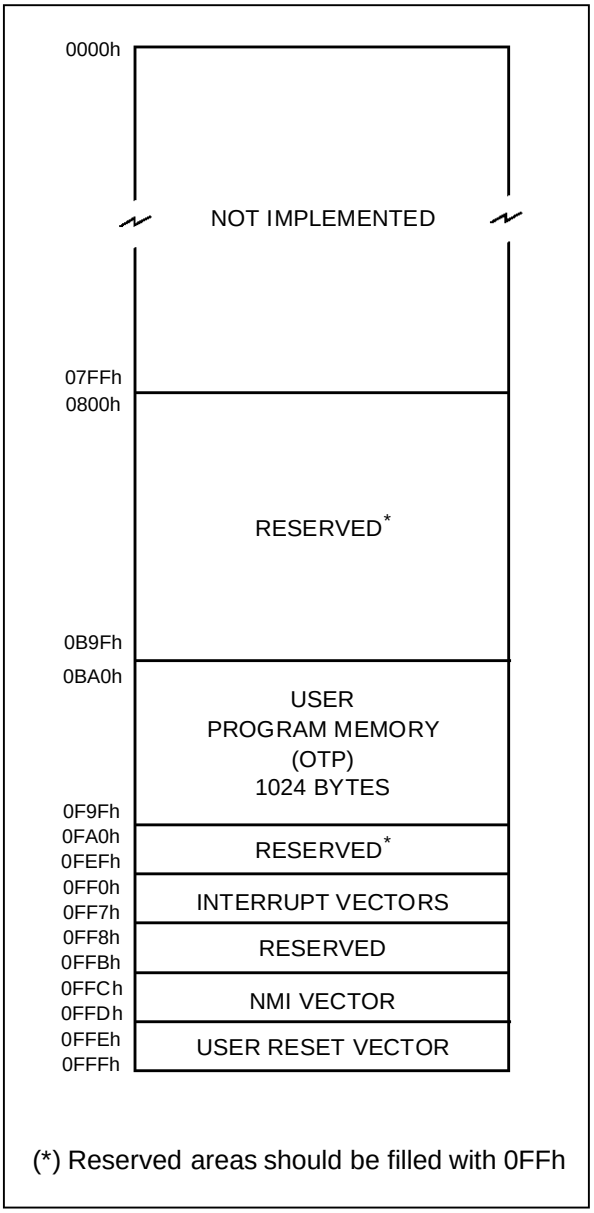
Figure 2. ST62T00, T01, E01 Pin Configuration



1.3 MEMORY MAPS

1.3.1 Program Memory Maps

Figure 3. ST62T00 Program Memory Map



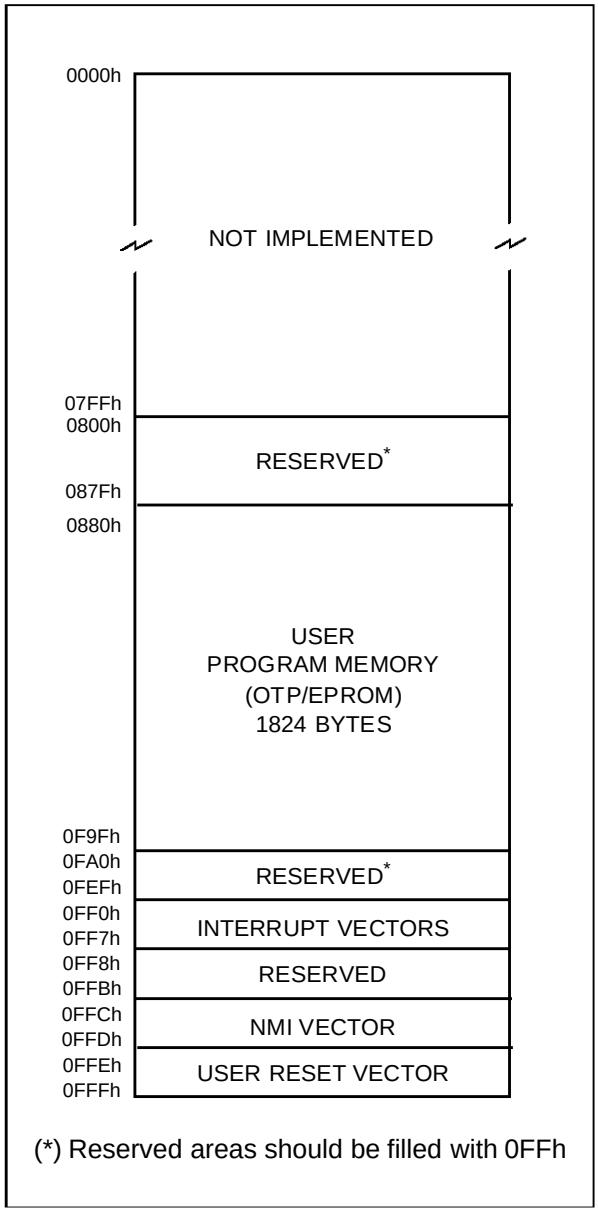
(*) Reserved areas should be filled with 0FFh

1.3.2 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.

The Data Space is fully described and illustrated on page 20.

Figure 4. ST62T01,E01 Program Memory Map



(*) Reserved areas should be filled with 0FFh

1.4 PARTICULARITIES OF OTP AND EPROM DEVICES

OTP and EPROM devices are identical save for the package which, in the EPROM device, is fitted with a transparent window to allow erasure of memory contents by exposure to UV light.

Both OTP and EPROM parts may be programmed using programming equipment approved by SGS-THOMSON.

1.4.1 OTP/EPROM Programming

Programming mode is selected by applying a 12.5V voltage to the V_{PP}/TEST pin during reset. Programming of OTP and EPROM parts is fully described in the EPROM Programming Board User Manual.

1.4.2 Eprom Erasure

Thanks to the transparent window present in the EPROM package, its memory contents may be erased by exposure to UV light.

Erasure begins when the device is exposed to light with a wavelength shorter than 4000Å. It should be noted that sunlight, as well as some types of artificial light, includes wavelengths in the 3000-4000Å range which, on prolonged exposure, can cause erasure of memory contents. It is thus recommended that EPROM devices be fitted with an opaque label over the window area in order to prevent unintentional erasure.

The recommended erasure procedure for EPROM devices consists of exposure to short wave UV light having a wavelength of 2537Å. The minimum recommended integrated dose (intensity x exposure time) for complete erasure is 15Wsec/cm^2. This is equivalent to an erasure time of 15-20 minutes using a UV source having an intensity of 12mW/cm^2 at a distance of 25mm (1 inch) from the device window.

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

The CPU Core may be thought of as an independent central processor communicating with on-chip I/O, memory and peripherals. For further details refer to page 16.

2.2 CPU REGISTERS

The CPU Core features six registers and three pairs of flags available to the programmer. For a detailed description refer to page 22.

3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The Oscillator may be driven by an external clock, or by a crystal or ceramic resonator. ROM devices also offer RC oscillator and Oscillator Safeguard features. For a complete description refer to page 24.

3.2 RESETS

The MCU can be reset in three ways: by the external Reset input being pulled low, by the Power-on Reset circuit, or by the Digital Watchdog timing out. For further details refer to page 28.

3.3 DIGITAL WATCHDOG

The Digital Watchdog can be used to provide controlled recovery from software upsets. Software and Hardware enabled Watchdog options are available in order to achieve optimum trade-off between power consumption and noise immunity. For a complete description and a selection guide refer to page 31.

3.4 INTERRUPTS

The CPU can manage four Maskable and one Non-Maskable Interrupt source. Each source is associated with a specific Interrupt Vector. An internal pullup option on the NMI pin is available on ROM devices. For a complete description refer to page 35.

3.5 POWER SAVING MODES

WAIT mode reduces electrical consumption during idle periods, while STOP mode achieves the lowest power consumption by stopping all CPU activity. For a complete description refer to page 38.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

Input/Output lines may be individually programmed as one of a number of different configurations. For further details refer to page 40.

4.2 TIMER

The on-chip Timer peripheral consists of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. For a complete description refer to page 44.

4.3 A/D CONVERTER (ADC)

The 8-bit on-chip ADC features multiplexed analog inputs, as alternate I/O functions. Conversion is by successive approximations, with a typical conversion time of 70us, at 8MHz oscillator frequency. For a complete description refer to page 47

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 architecture has been designed to exploit the hardware in the most efficient way possible, while keeping byte usage to a minimum. For further details refer to page 49.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes: Immediate, Direct, Short Direct, Extended, Program Counter Relative, Bit Direct, Bit Test & Branch, Indirect, and Inherent. For a complete description of the available addressing modes, refer to page 49.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes; these may be subdivided into six types: load/store, arithmetic/logic, conditional branch, control, jump/call, and bit manipulation. For further details refer to page 50.

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices designed to protect the inputs against damage due to high static voltages; however, it is advisable to take normal precautions to avoid applying voltages higher than the specified maximum ratings.

For proper operation, it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, T_j, in degrees Celsius can be obtained from:

$$T_j = T_A + P_D \times R_{thJA}$$

Where:

T_A = Ambient Temperature.

R_{thJA} =Package thermal resistance (junction-to ambient).

$P_D = P_{int} + P_{port}$

$P_{int} = I_{DD} \times V_{DD}$ (chip internal power).

P_{port} =Port power dissipation (to be determined by the user)

.

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_{PP}	OTP/EPROM Programming Voltage	13	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into V_{DD} (source)	50(2)	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

Stresses above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

(1) Within these limits, clamping diodes are non-conducting. Voltages outside these limits are authorised provided injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

6.2 THERMAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{thJA}	Thermal Resistance (junction to ambient)	PDIP16			60	
		PSO16			80	
		CDIP16W			n/a	

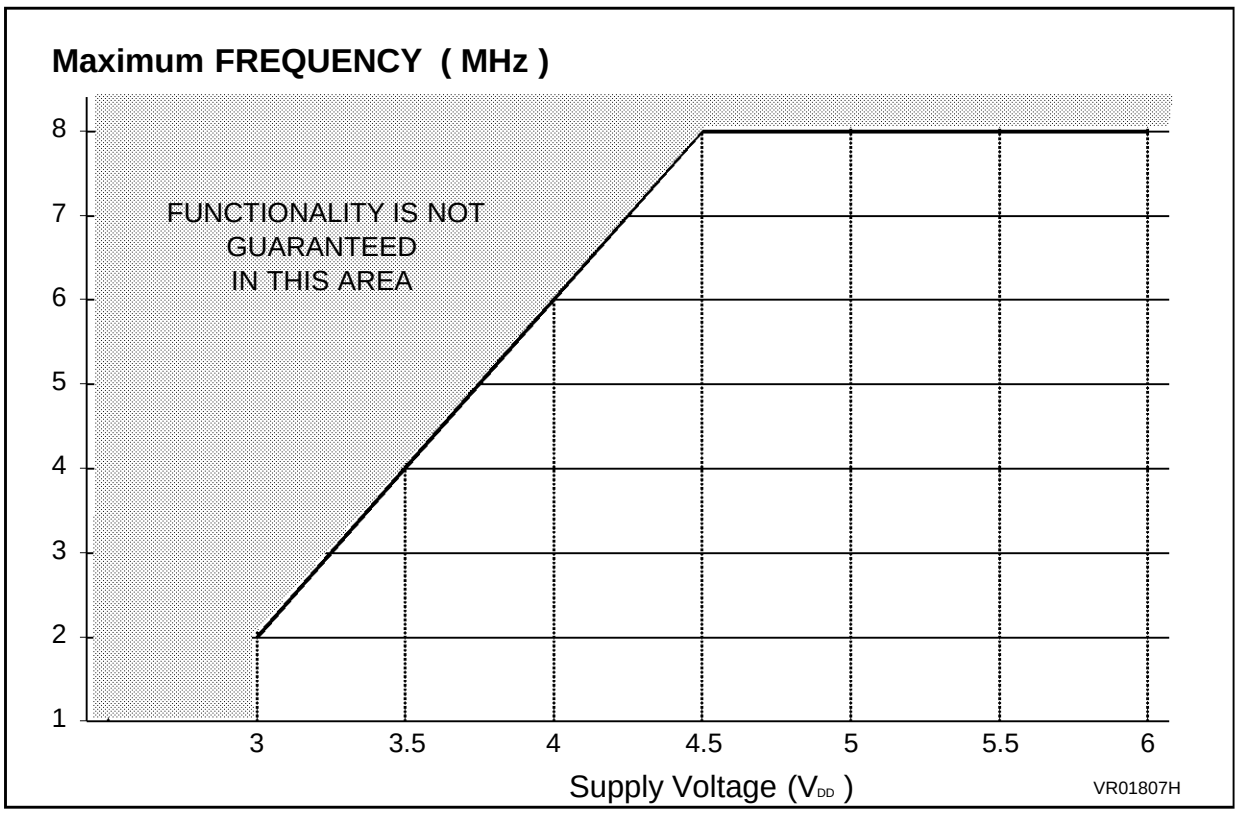
6.3 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

If a total current of +1mA is flowing into a single analog channel, or if the total current flowing into all the analog inputs is 1mA, all resulting A/D conversions will be shifted by + 1 LSB. If a total positive current is flowing into a single analog channel, or if the total current flowing into all analog inputs is 5mA, all the resulting conversions are shifted by + 2 LSB.

Figure 5. Maximum Operating FREQUENCY (F_{MAX}) Versus SUPPLY VOLTAGE (V_{DD})

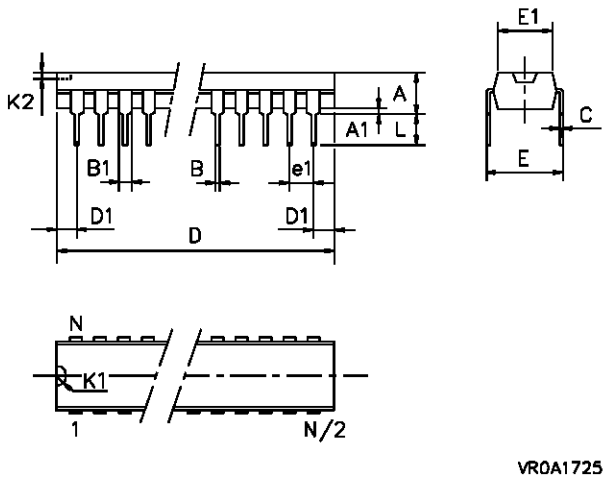


The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

7 GENERAL INFORMATION

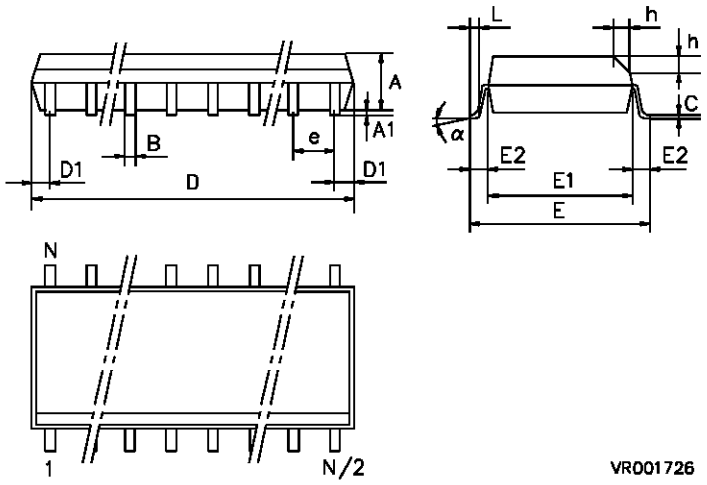
7.1 PACKAGE MECHANICAL DATA

Figure 6. 16-Pin Plastic Dual In Line Package (B), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			.200
A1			.508	.020		
B	.381	.508	.533	.015	.020	.021
B1	.762		1.651	.030		.065
C	.203	.254	.304	.008	.010	.012
D	18.92	19.18	19.56	.745	.755	.770
D1			1.27			.050
E	7.37	7.62	7.874	.290	.300	.310
E1		5.334			.210	
K1						
K2						
L	2.997	3.302	3.708	.118	.130	.146
e	2.286	2.54	2.794	.090	.100	.110
	Number of Pins					
N	16					

Figure 7. 16-Pin Plastic Small Outline Package (M), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		2.286			.090	
A1	.102		.305	0.004		.012
B	.381		.483	.015		.019
C	.229		.254	.009		.010
D	10.26		10.34	.404		.407
D1	-	-	-	-	-	-
E	10.24		10.34	.403		.407
E1	7.44		7.54	.293		.297
E2	-	-	-	-	-	-
L		.832			.033	
e		1.27			.050	
h		.508			.020	
alpha		5^o			5^o	
	Number of Pins					
N	16					

Table 1. OTP Device Sales Types

Sales Type	I/O Pins	Option	Temperature range	Package
ST62T00B6/HWD	9	Hardware Watchdog	-40°C TO +85°C	PDIP16
ST62T00B6/SWD	9	Software Watchdog		
ST62T00M6/HWD	9	Hardware Watchdog		PSO16
ST62T00M6/SWD	9	Software Watchdog		
ST62T01B6/HWD	9	Hardware Watchdog		PDIP16
ST62T01B6/SWD	9	Software Watchdog		
ST62T01M6/HWD	9	Hardware Watchdog		PSO16
ST62T01M6/SWD	9	Software Watchdog		

Table 2. EPROM Device Sales Types

Sales Type	I/O Pins	Option	Temperature range	Package
ST62E01F1/HWD	9	Hardware Watchdog	0 TO +70°C	CDIP16W
ST62E01F1/SWD	9	Software Watchdog		

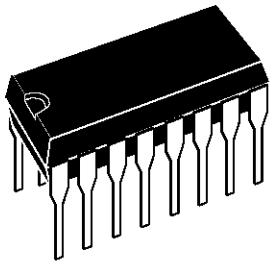
Notes :

SGS-THOMSON MICROELECTRONICS

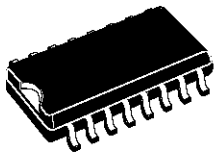
ST6200B
ST6201B

8-BIT HCMOS MCUs WITH A/D CONVERTER

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in ROM
- Data ROM: User selectable size (in program ROM)
- Data RAM: 64 bytes
- ROM readout Protection
- PDIP16, PSO16 packages
- 9 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 3 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer with 7-bit programmable prescaler
- Digital Watchdog
- Oscillator Safe Guard
- 8-bit A/D Converter with 4 analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal, Ceramic resonator or RC network
- Power-on Reset
- One external Non-Maskable Interrupt
- 9 powerful Addressing Modes
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP16



PSO16

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	I/O Pins
ST6200B	1036	9
ST6201B	1836	9

3 GENERAL DESCRIPTION

3.1 INTRODUCTION

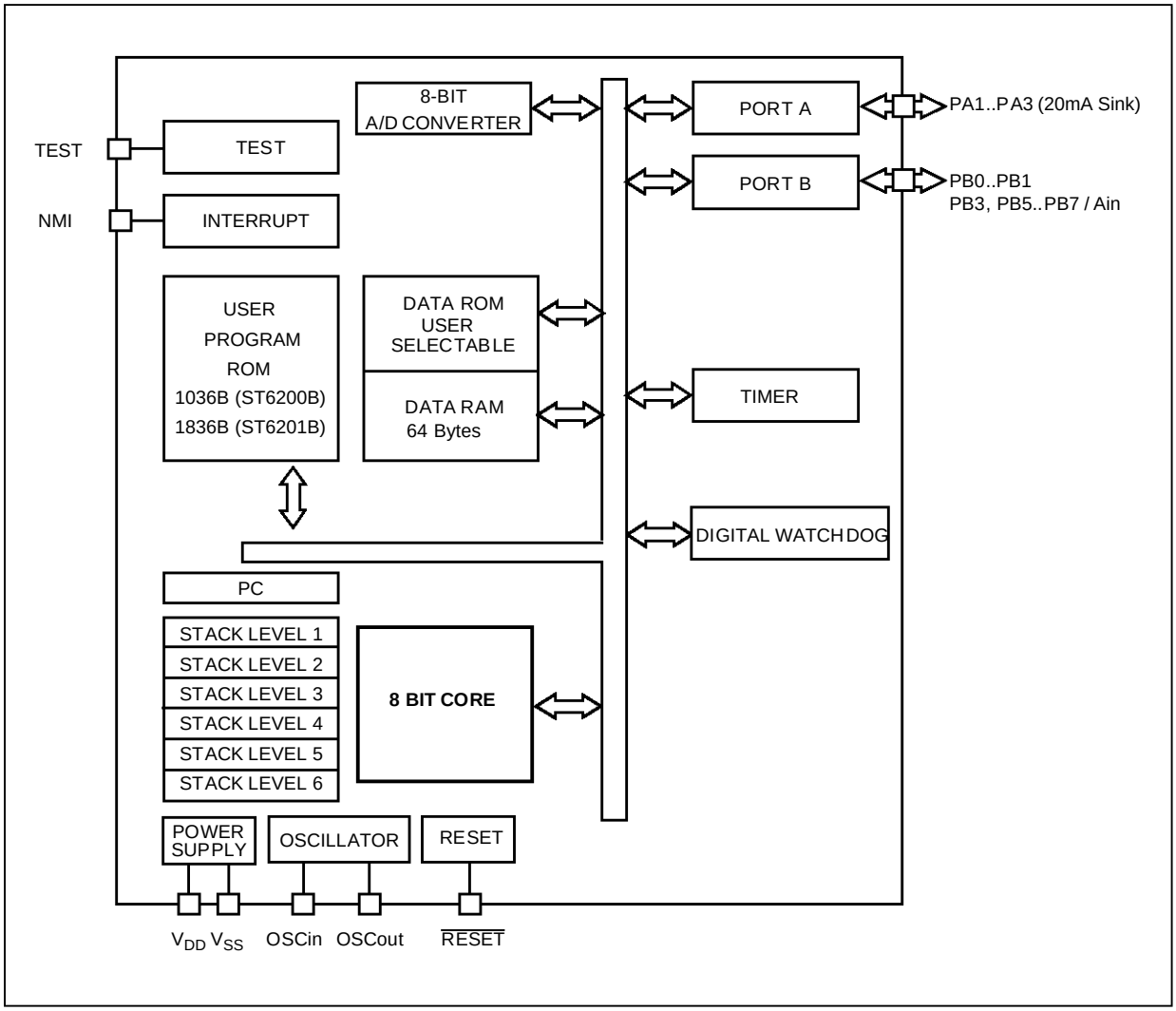
The ST6200B and ST6201B microcontrollers are members of the ST62xx 8-bit HCMOS family of devices, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST6200B and ST6201B devices are functionally identical, save for their program memory size. The devices feature the following peripherals: a Timer comprising an 8-bit counter and a 7-bit programmable prescaler, an 8-bit A/D Converter with 4 analog inputs (A/D inputs are I/O pin alternate functions), and a Digital Watchdog timer.

The ST6200B and ST6201B devices feature a choice of Quartz, Ceramic or RC oscillators, an Oscillator Safe Guard circuit, Readout Protection against unauthorised copying of program code, and an External STOP Mode Control option to enlarge the range of power consumption versus reliability trade-offs.

These devices are well suited for automotive, appliance and industrial applications. The user programmable part for program development is the ST62E01.

Figure 1. Block Diagram



3.2 PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. When the QUARTZ/CERAMIC RESONATOR Mask Option is selected, a quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. When the RC OSCILLATOR Mask Option is selected, a resistor must be connected between the OSCout pin and ground. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

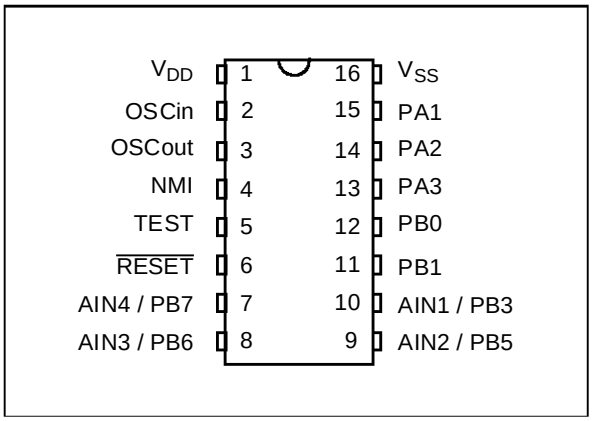
TEST. The TEST pin must be held at V_{SS} for normal operation (an internal 100kΩ pull-down resistor selects normal operating mode if the TEST pin is not connected externally).

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI is falling edge sensitive. A ROM mask option makes available an on-chip pull-up on the NMI pin.

PA1-PA3. These 3 lines are organized as an I/O port (A). Each line may be configured under software control as an input with or without internal pull-up resistors, as an interrupt generating input with pull-up resistors, or as an open-drain or push-pull output. PA1-PA3 can sink up to 20mA for direct LED drive capability. When the External STOP Mode Control option is enabled, PA2 must be defined as an input.

PB0, PB1, PB3, PB5-PB7. These 6 lines are organized as one I/O port (B). When the External STOP Mode Control option is disabled, each line may be configured under software control as an input with or without internal pull-up resistor, as an interrupt generating input with pull-up resistor, or as an open-drain or push-pull output. PB7-PB5 and PB3 can also be used as analog inputs to the A/D converter. When the External STOP Mode Control option is enabled, PB0 can only be configured as open-drain in output mode (push-pull output is not available). The other lines are unchanged.

Figure 2. ST6200B and 01B Pin Configuration



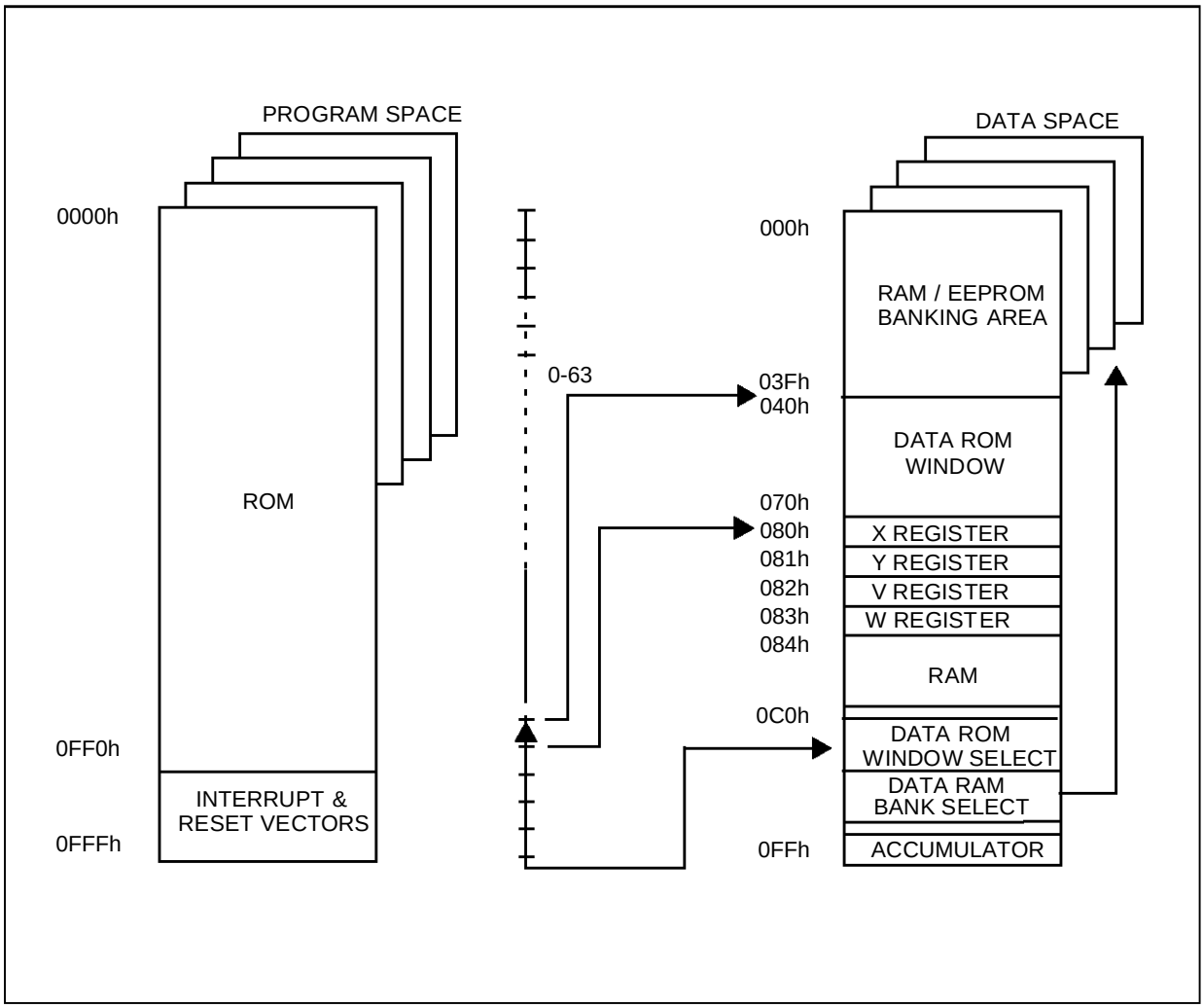
3.3 MEMORY MAP

3.3.1 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 ROM and user vectors; Data space contains user data in RAM and in ROM, and Stack space accommodates six levels of stack for subroutine and interrupt service routine nesting.

Figure 3. Memory Addressing Diagram



MEMORY MAP (Cont'd)

3.3.2 Program Space

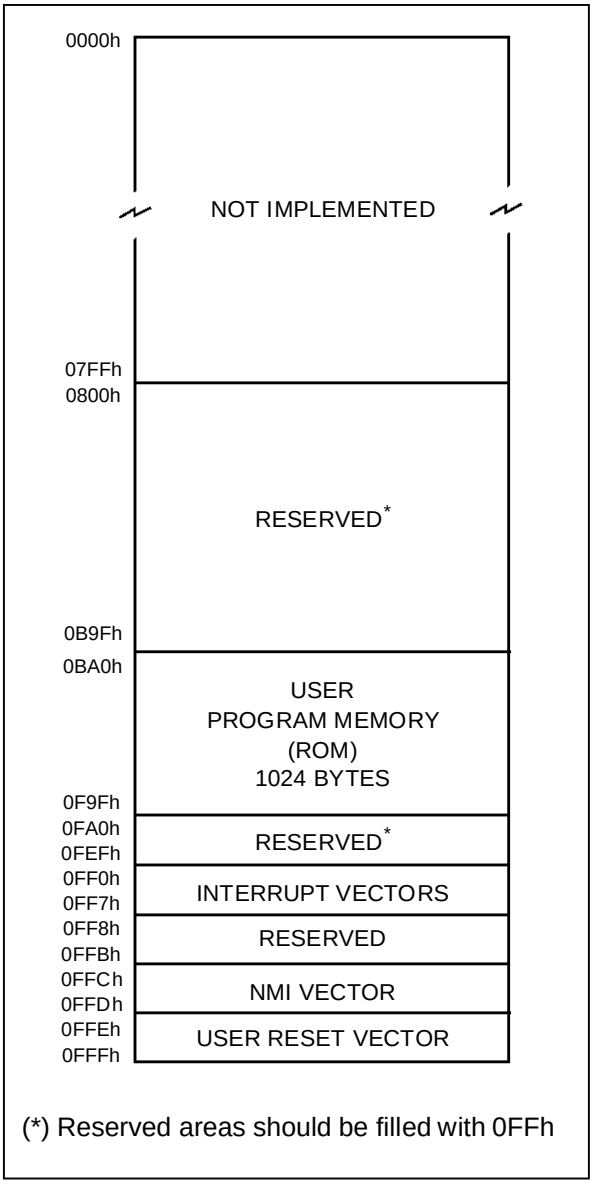
Program Space is physically implemented in ROM memory. It comprises the instructions to be executed, the data required for immediate addressing mode instructions, the reserved factory test area and the user vectors. Program Space is addressed via the 12-bit Program Counter register (PC register)

3.3.2.1 ROM Protection

The Program Space can be protected against external readout of ROM contents when the READOUT PROTECTION maskoption is chosen. This option allows the user to blow a dedicated fuse on the silicon, by applying a high voltage at V_{PP} (see detailed information in the "Electrical Specification").

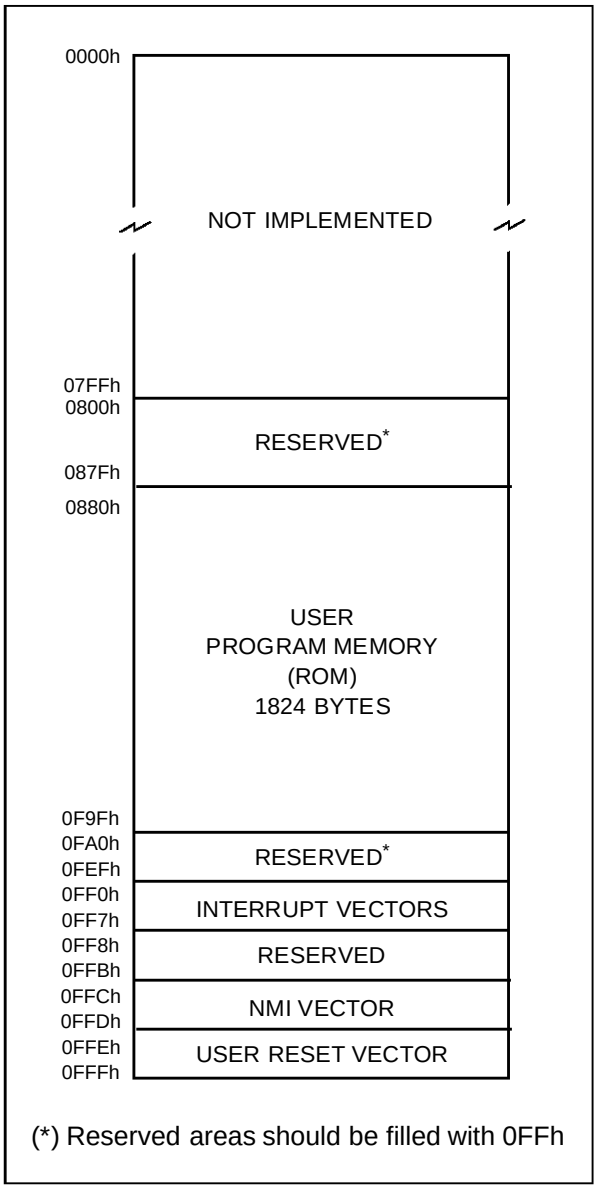
Note: Once the Readout Protection fuse is blown, it is no longer possible, even for SGS-THOMSON, to gain access to the ROM contents. Returned parts with a blown fuse can therefore not be accepted.

Figure 4. ST6200B Program Memory Map



(*) Reserved areas should be filled with 0FFh

Figure 5. ST6201B Program Memory Map



(*) Reserved areas should be filled with 0FFh

MEMORY MAP (Cont'd)

3.3.3 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 ROM.

3.3.3.1 Data ROM

All read-only data is physically stored in ROM memory, which also accommodates the Program Space. The ROM 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 ROM.

3.3.3.2 Data RAM

The data space includes 60 bytes of RAM, 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 (DRW register).

3.3.4 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.

Table 1. Data Memory Space

NOT IMPLEMENTED	000h 03Fh
DATA ROM WINDOW 64 BYTES	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
RESERVED	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
RESERVED	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PSC REGISTER	0D2h
TIMER DATA REGISTER	0D3h
TIMER TSCR REGISTER	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

3.3.5 Data Window Register (DWR)

The Data ROM window is located from address 0040h to address 007Fh in Data space. It allows direct reading of 64 consecutive bytes located anywhere in ROM memory, between address 0000h and 1FFFh (top memory address depends on the specific device). All the ROM memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the ROM memory by writing the appropriate code in the Write-only Data Window register (DWR register, location 00C9h).

The DWR register can be addressed like any RAM location in the Data Space at address 00C9h, it is however a write-only register and cannot be accessed using single-bit operations. This register is used to move the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as data in ROM memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 6). So when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in ROM is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data ROM window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

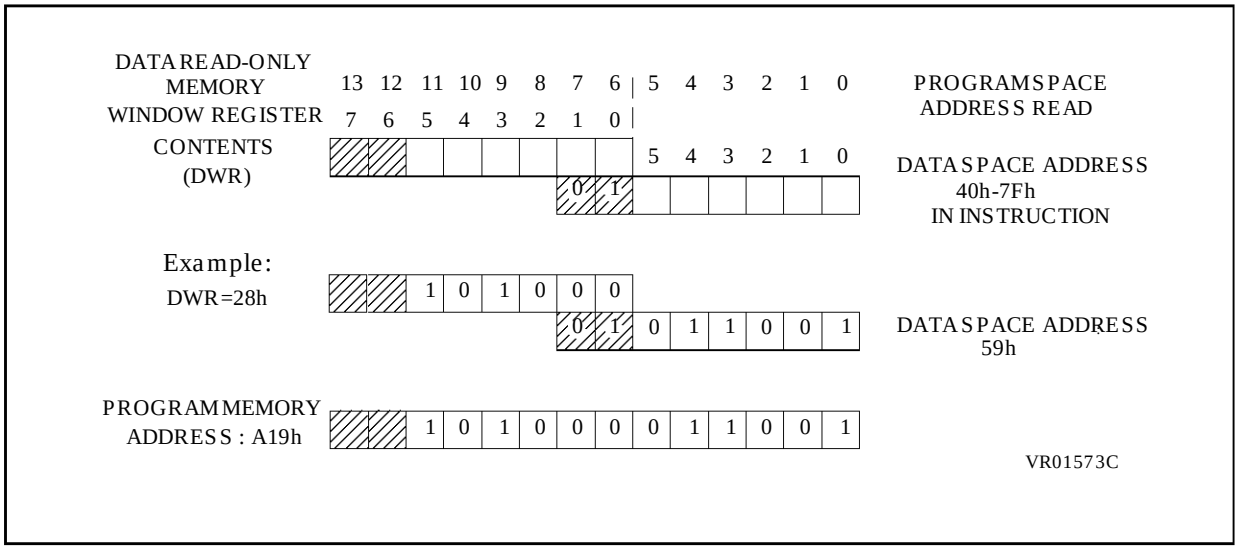
Bit 7 = This bit is not used.

Bit 6-0 = **DWR6-DWR0:** *Data ROM Window Register Bits.* These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

Caution: *This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.*

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

Figure 6. Data ROM Window Memory Addressing



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. In-core communication is arranged as shown in Figure 7; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

4.2 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.

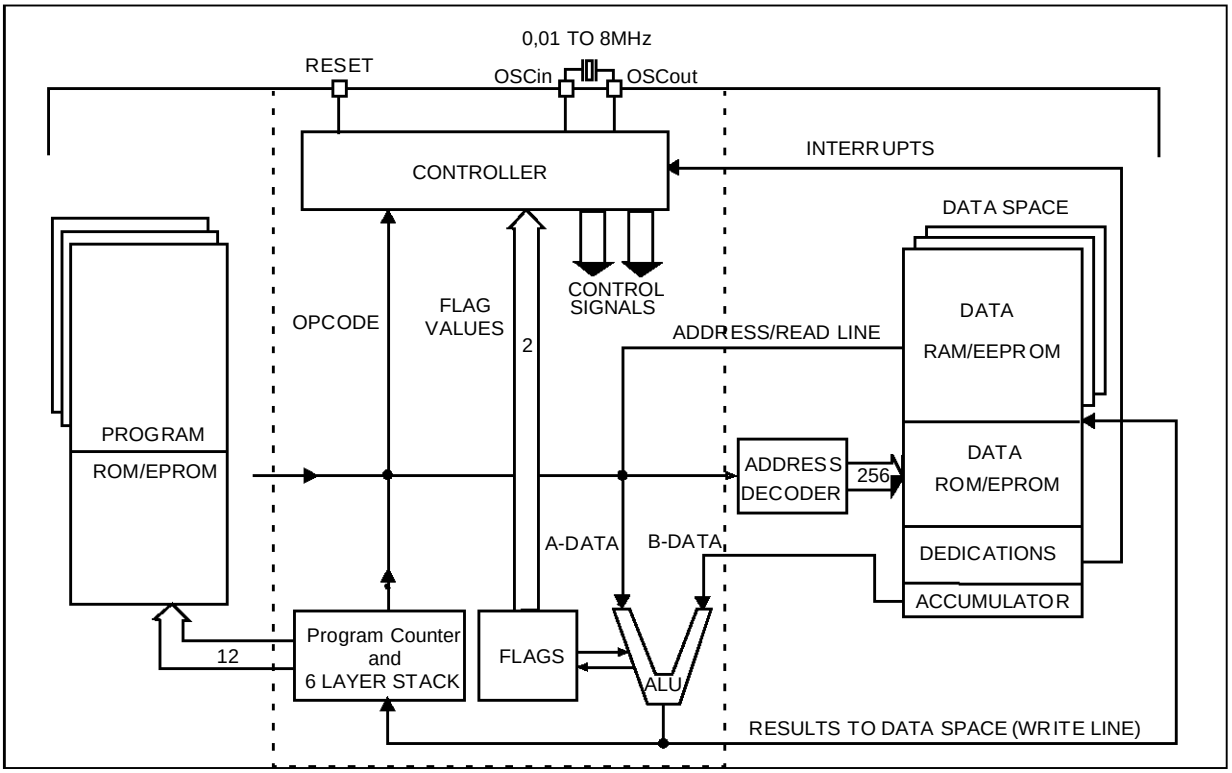
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 7. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

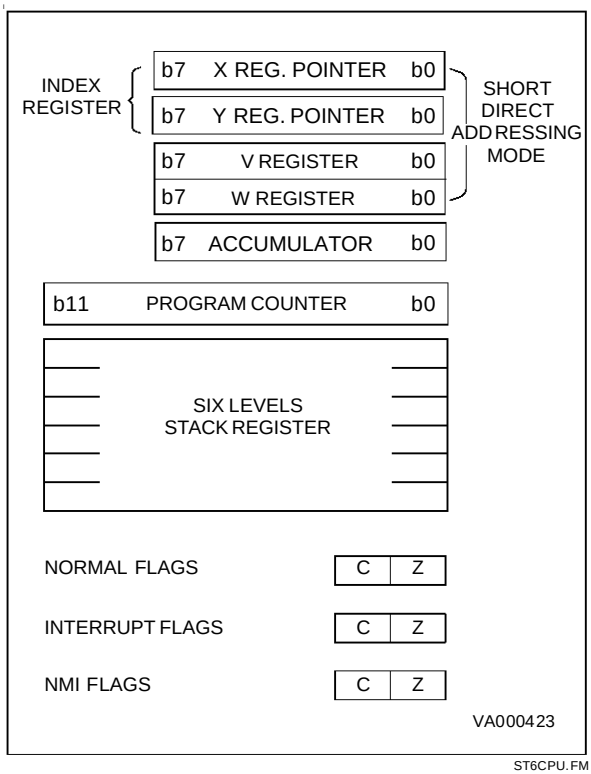
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 8. ST6 CPU Programming Mode



ST6CPU.FM

5 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

5.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}). In addition, a Low Frequency Auxiliary Oscillator (LFAO) can be switched in for security reasons, to reduce power consumption, or to offer the benefits of a back-up clock system.

The Oscillator Safeguard (OSG) option filters spikes from the oscillator lines, provides access to the LFAO to provide a backup oscillator in the event of main oscillator failure and 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, Figure 11, Figure 12 and Figure 13.

Figure 9 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. C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 27 and Figure 28.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 12.

With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

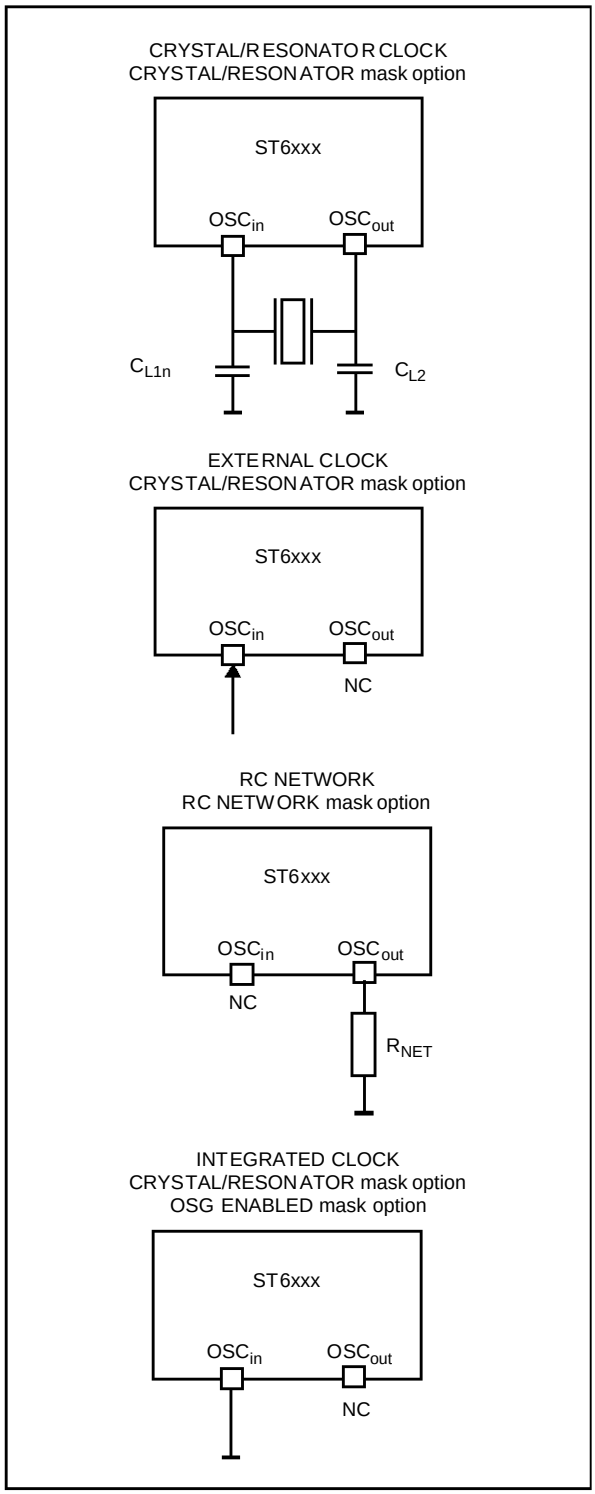
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

5.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate mask option. 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 main oscillator can be turned off (when the OSG ENABLED mask option is selected) by setting the OSCOFF bit of the ADC Control Register. The Low Frequency Auxiliary Oscillator is automatically started.

Figure 9. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Turning on the main oscillator is achieved by resetting the OSCOFF bit of the A/D Converter Control Register or by resetting the MCU. Restarting the main oscillator implies a delay comprising the oscillator start up delay period plus the duration of the software instruction at f_{LFAO} clock frequency.

5.1.2 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 safety oscillator in case of main oscillator failure.

This oscillator is available when the OSG ENABLED mask option is selected. In this case, it automatically starts one of its periods after the first missing edge from the main oscillator, whatever the reason (main oscillator defective, no clock circuitry provided, main oscillator switched off...).

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 1MHz.

At power on, the Low Frequency Auxiliary Oscillator starts faster than the Main Oscillator. It therefore feeds the on-chip counter generating the POR delay until the Main Oscillator runs.

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

ADCR

Address: 0D1h — Read/Write

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

Bit 7-3, 1-0= **ADCR7-ADCR3, ADCR1-ADCR0**: *ADC Control Register.* These bits are not used.

Bit 2 = **OSCOFF**. When low, this bit enables main oscillator to run. The main oscillator is switched off when OSCOFF is high.

5.1.3 Oscillator Safe Guard

The Oscillator Safe Guard (OSG) affords drastically increased operational integrity in ST62xx devices. The OSG circuit provides three basic functions: it filters spikes from the oscillator lines which would result in over frequency to the ST62 CPU; it gives access to the Low Frequency Auxiliary Oscillator (LFAO), used to ensure minimum processing in case of main oscillator failure, to offer reduced power consumption or to provide a fixed frequency low cost oscillator; finally, it automatically limits the internal clock frequency as a function of supply voltage, in order to ensure correct operation even if the power supply should drop.

The OSG is enabled or disabled by choosing the relevant OSG mask option. It may be viewed as a filter whose cross-over frequency is device dependent.

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. This relationship is illustrated in Figure 13.

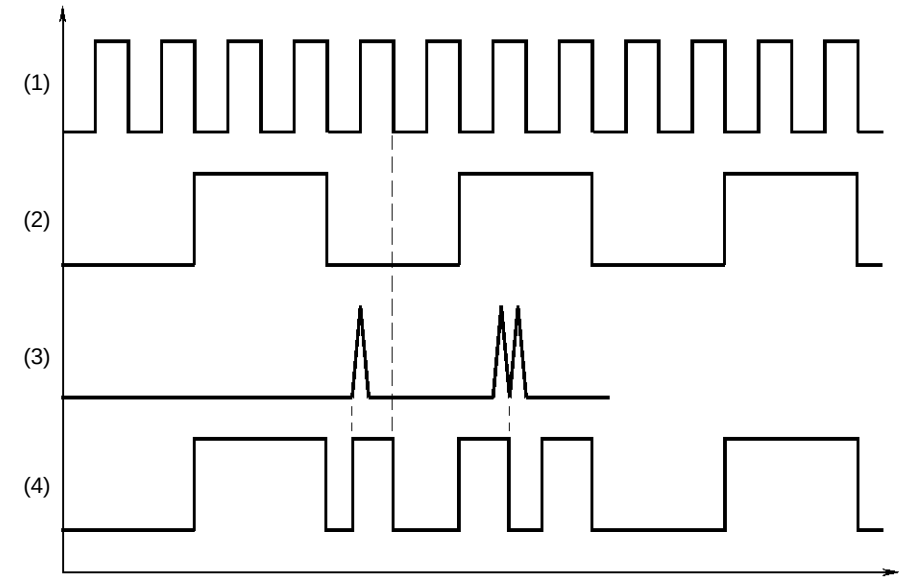
When the OSG is enabled, the Low Frequency Auxiliary Oscillator may be accessed. This oscillator starts operating after the first missing edge of the main oscillator (see Figure 11).

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.

Note. The OSG should be used wherever possible as it provides maximum safety. Care must be taken, however, as it can increase power consumption and reduce the maximum operating frequency to f_{OSG}.

CLOCK SYSTEM (Cont'd)

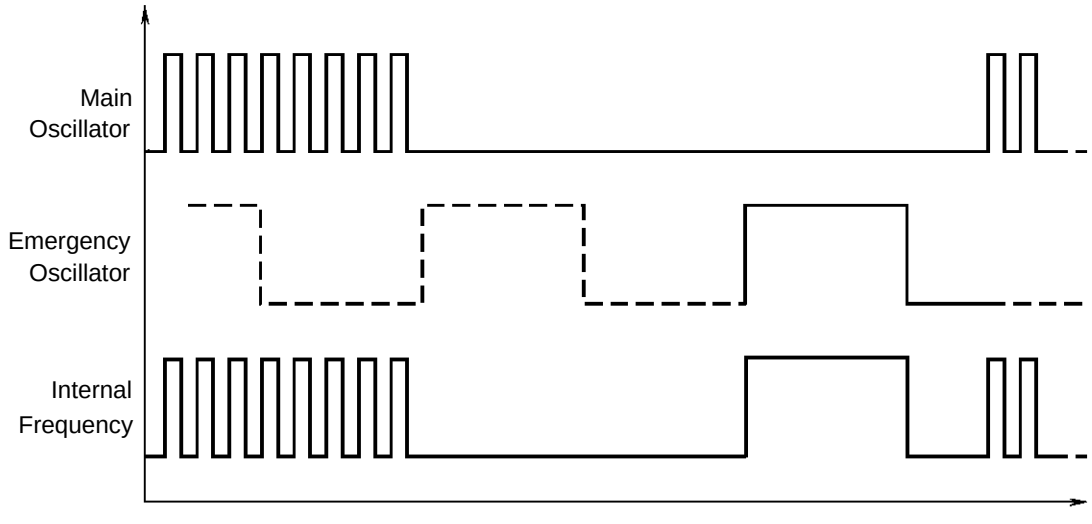
Figure 10. OSG Filtering Principle



(1) Maximum Frequency for the device to work correctly
(2) Actual Quartz Crystal Frequency at OSCin pin
(3) Noise from OSCin
(4) Resulting Internal Frequency

VR001932

Figure 11. OSG Emergency Oscillator Principle



VR001933

CLOCK SYSTEM (Cont'd)

Figure 12. Clock Circuit Block Diagram

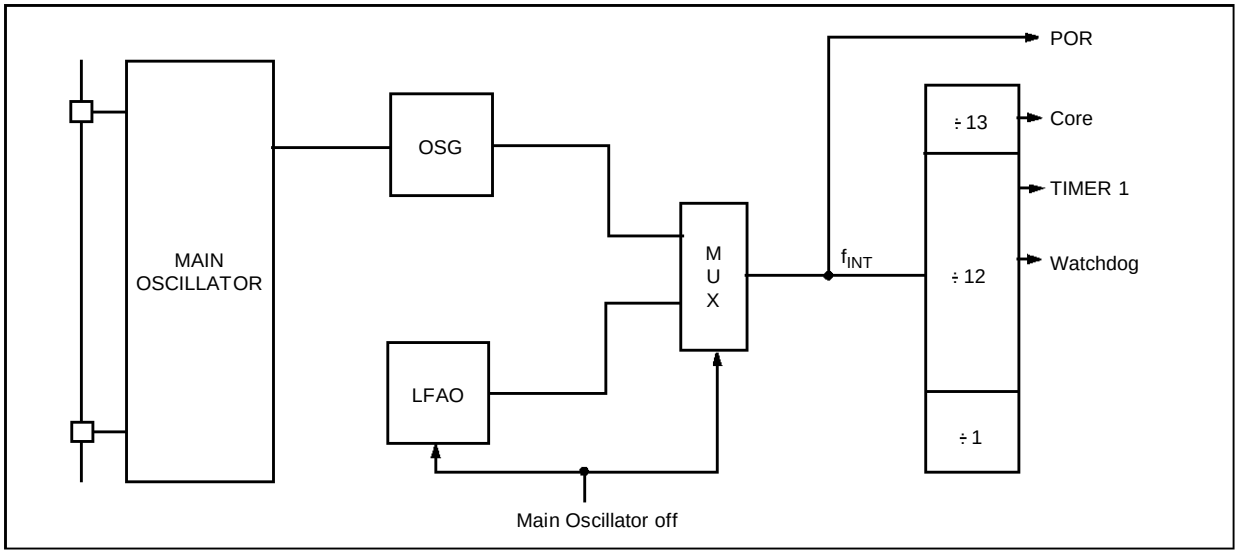
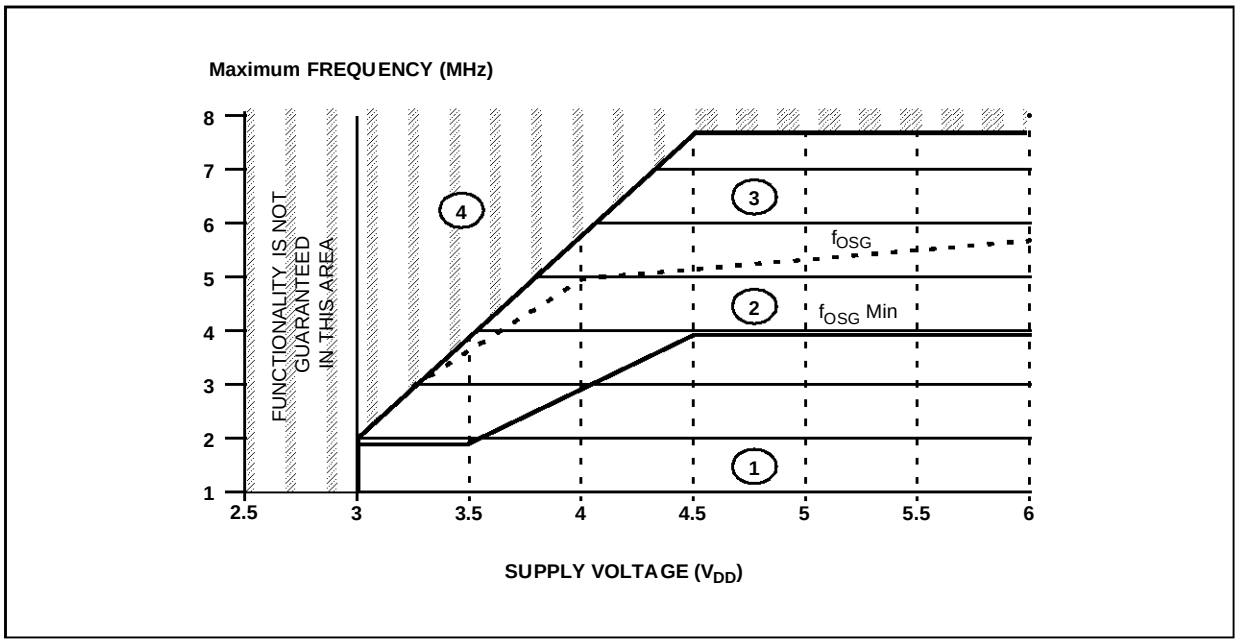


Figure 13. Maximum Operating Frequency (f_{MAX}) versus Supply Voltage (V_{DD})



Notes:

1. In this area, operation is guaranteed at the quartz crystal frequency.
2. When the OSG is disabled, operation in this area is guaranteed at the crystal frequency. When the OSG is enabled, operation in this area is guaranteed at a frequency of at least $f_{OSG\ Min.}$
3. When the OSG is disabled, operation in this area is guaranteed at the quartz crystal frequency. When the OSG is enabled, access to this area is prevented. The internal frequency is kept a f_{OSG}.
4. When the OSG is disabled, operation in this area is not guaranteed
When the OSG is enabled, access to this area is prevented. The internal frequency is kept at f_{OSG}.

5.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

5.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

5.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

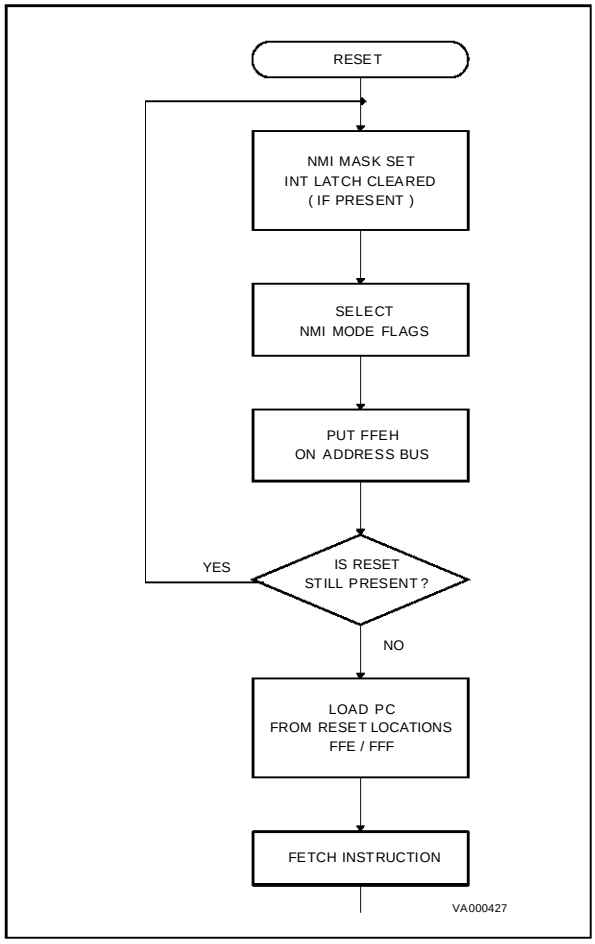
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 14. Reset and Interrupt Processing



RESETS (Cont'd)

5.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

5.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

5.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 15. Reset and Interrupt Processing

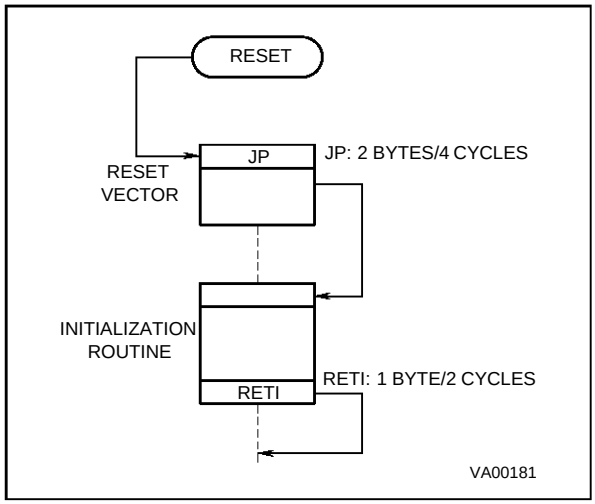
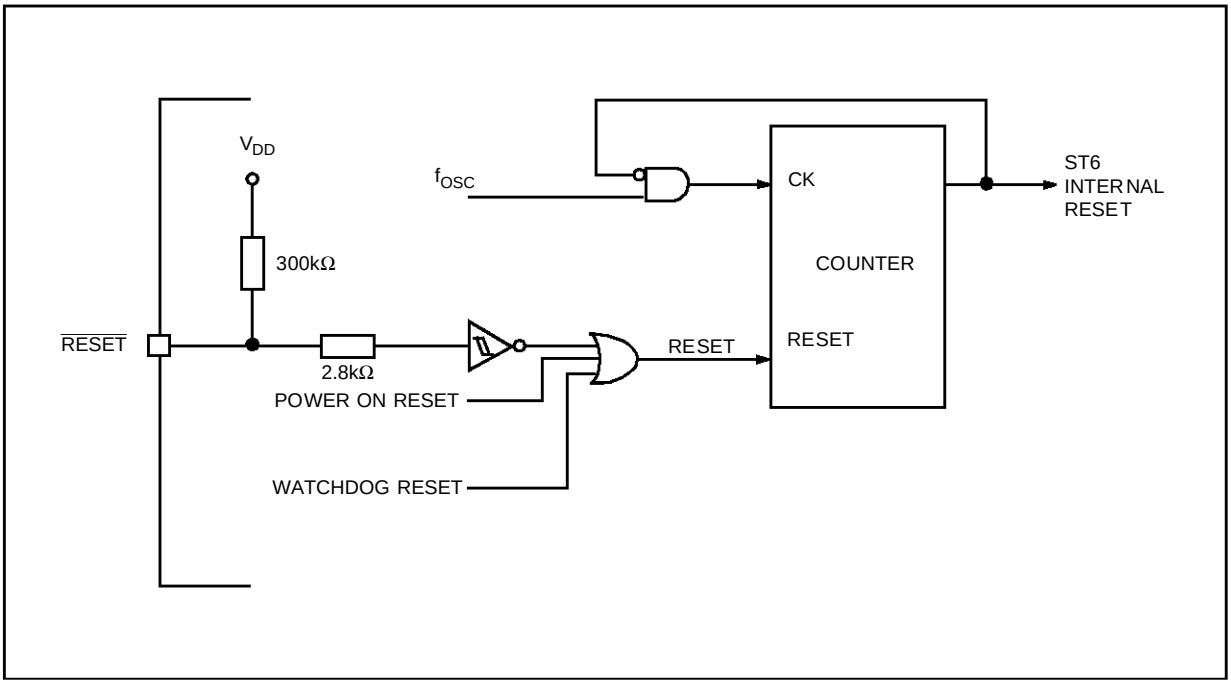


Figure 16. Reset Block Diagram



RESETS (Cont'd)

Table 2. Register Reset Status

Register	Address(es)	Status	Comment
Port Data Registers (PA, PB)	0C0h to 0C1h		
Port Direction Register (PA, PB)	0C4h to 0C5h		I/Os are Inputs with pull-up
Port Option Register (PA, PB)	0CCh to 0CDh	00h	I/Os are Inputs with pull-up
Interrupt Option Register	0C8h		Interrupts disabled
Timer Status/Control	0D4h		Timer disabled
X, Y, V, W Register	080h to 083h		
Accumulator	0FFh		
Data RAM	084h to 0BFh	Undefined	
Data ROM Window Register	0C9h		
A/D Result Register	0D0h		
Timer Counter Register	0D3h	FFh	
Timer Prescaler Register	0D2h	7Fh	Maximum count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Stand-by, main oscillator on

5.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two mask options, known as “WATCHDOG ACTIVATION” (i.e. HARDWARE or SOFTWARE) and “EXTERNAL STOP MODE CONTROL” (see Table 3).

In the SOFTWARE mask option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, save by resetting the MCU.

In the HARDWARE mask option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL mask option (available in ROM versions only) has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL mask option has been selected, port PB0 must be defined as an open-drain output, and PA2 as an input.

Table 3. Recommended Mask Option Choices

Functions Required	Recommended Mask Options
Stop Mode & Watchdog	“EXTERNAL STOP MODE” & “HARDWARE WATCHDOG”
Stop Mode	“SOFTWARE WATCHDOG”
Watchdog	“HARDWARE WATCHDOG”

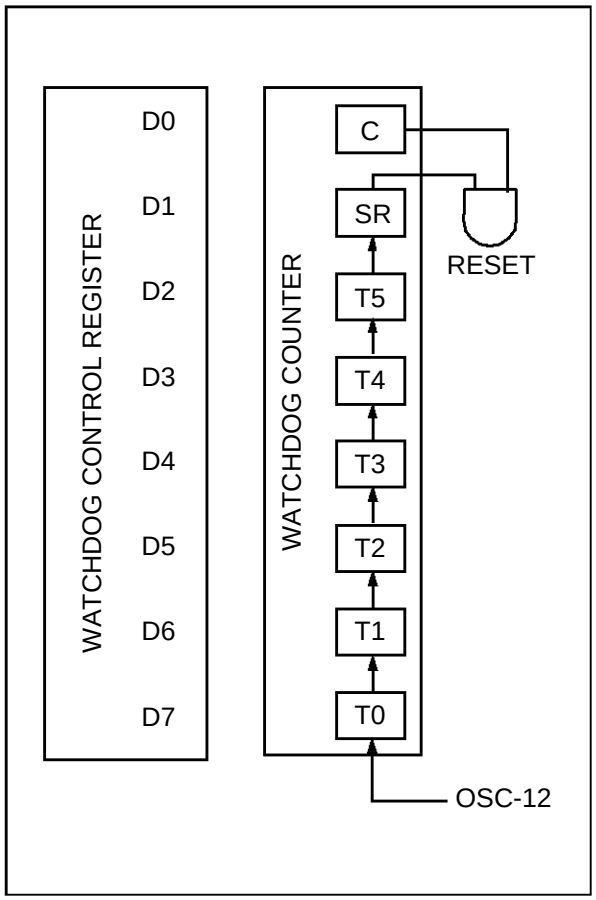
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 5.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 17.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 17. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

5.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

5.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 18) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 18. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

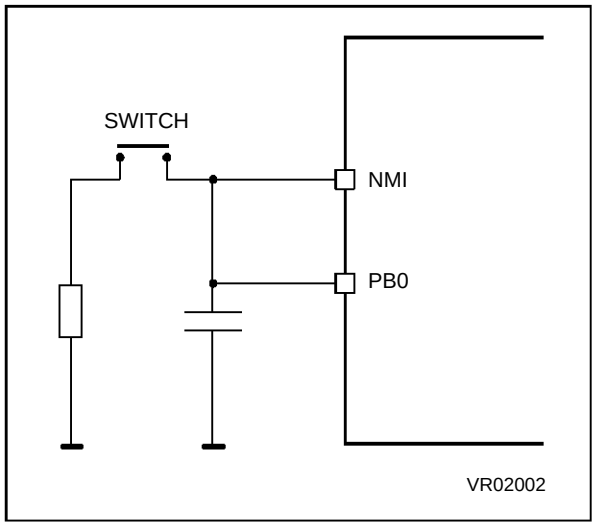
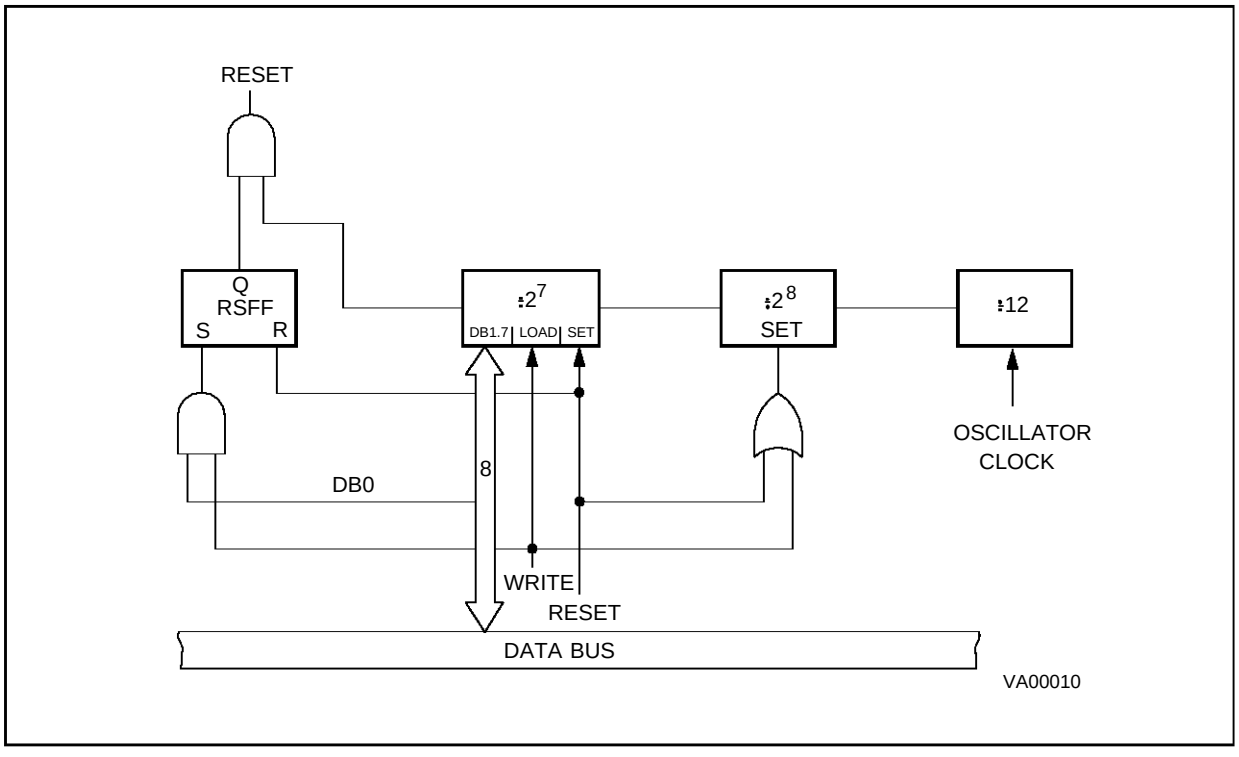


Figure 19. Digital Watchdog Block Diagram



5.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 4).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 4. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A pins	Interrupt vector #1	(FF6h-FF7h)
Port B pins	Interrupt vector #2	(FF4h-FF5h)
TIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
ADC peripheral	Interrupt vector #4	(FF0h-FF1h)

5.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

– The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named . It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port B pins is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named . It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the TIMER peripheral and vector #4 with the A/D Converter peripheral.

Each on-chip peripheral has an associated interrupt request flag (TMZ for the Timer, EOC for the A/D Converter), which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (ETI for the Timer, EAI for the A/D Converter), which must be set to “1” to enable the associated interrupt request.

5.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

5.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port B) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 5. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A
	CLEARED	Falling edge mode on Port A
ESB	SET	Level sensitive mode on Port B
	CLEARED	Falling edge mode on Port B

5.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A-vector #1, Port B-vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

IINTERRUPTS (Cont'd)

5.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

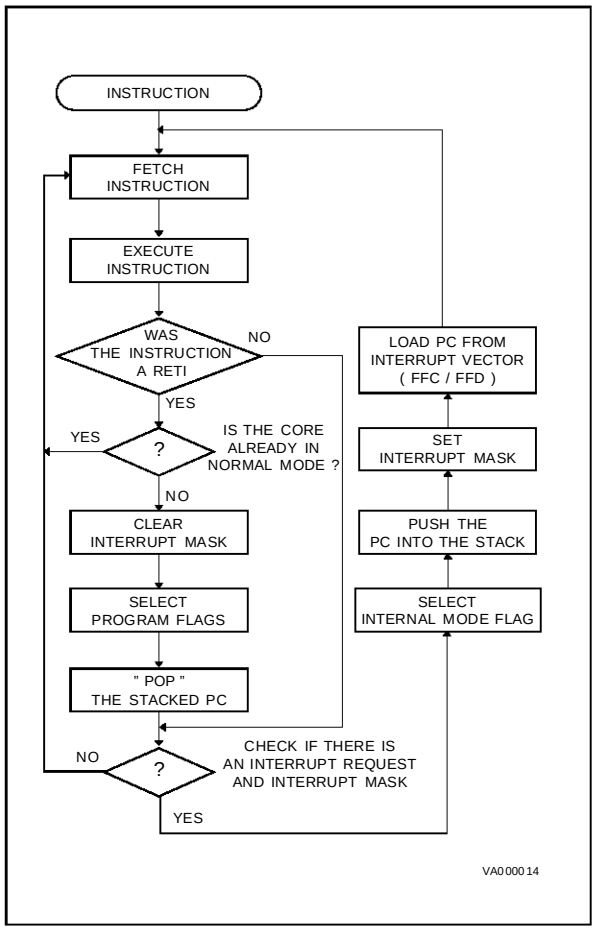
– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 20. Interrupt Processing Flow Chart



5.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

In addition, the Low Frequency Auxiliary Oscillator (LFAO) can be used instead of the main oscillator to reduce power consumption in RUN and WAIT modes.

5.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator (main oscillator or LFAO) is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the power consumption has to be further reduced, the Low Frequency Auxiliary Oscillator (LFAO) can be used in place of the main oscillator, if its operating frequency is lower. If required, the LFAO must be switched on before entering the WAIT mode.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

5.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

5.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection, consequently, when the LFAO is used, the user program must manage oscillator selection as soon as normal RUN mode is resumed.

5.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

5.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

5.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

– selecting the Low Frequency Auxiliary Oscillator (provided this runs at a lower frequency than the main oscillator).

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

6 ON-CHIP PERIPHERALS

6.1 I/O PORTS

The MCU features 9 Input/Output lines which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Analog input (PB5-PB7, PB3)

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PA1-PA3 only)

The lines are organized as two Ports (A and B).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA and DRB), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

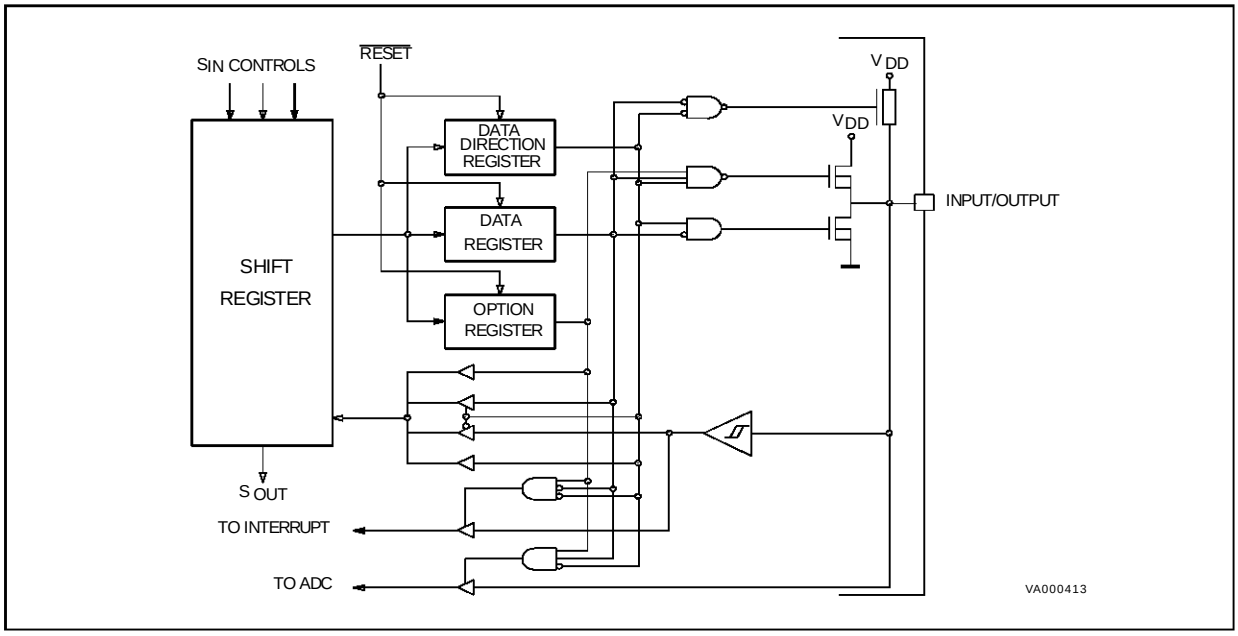
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA and DDRB) allow the data direction (input or output) of each pin to be set.

The two Option registers (ORA and ORB) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 21. I/O Port Block Diagram



I/O PORTS (Cont'd)

6.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 6 illustrates the various port configurations which can be selected by user software.

6.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

6.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A are AND-connected to the interrupt associated with Vector #1. The pins of Port B are AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

6.1.1.3 Analog Input Options

The four pins, PB5-PB7 and PB3, can be configured as analog inputs by programming the OR and DR registers accordingly. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

6.1.2 I/O Port Option Registers

ORA/B (CCh PA, CDh PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Option Register bits.*

6.1.3 I/O Port Data Direction Registers

DDRA/B (C4h PA, C5h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Direction Registers bits.*

6.1.4 I/O Port Data Registers

DRA/B (C0h PA, C1h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Registers bits.*

Note: Bits corresponding to non-existant external pin connections must be kept in their Reset state (i.e. set to "0").

Table 6. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Input	No pull-up, no interrupt (PA1-PA3 pins)
			Input	Analog input (PB3, PB5-PB7 pins)
1	0	X	Output	20mA sink open-drain output (PA1-PA3 pins)
1	0	X	Output	Standard open-drain output (PB0, PB1, PB3, PB5-PB7 pins)
1	1	X	Output	20mA sink push-pull output (PA1-PA3 pins)

Note: X = Don't care

I/O PORTS (Cont'd)

6.1.5 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 22. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A and B Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

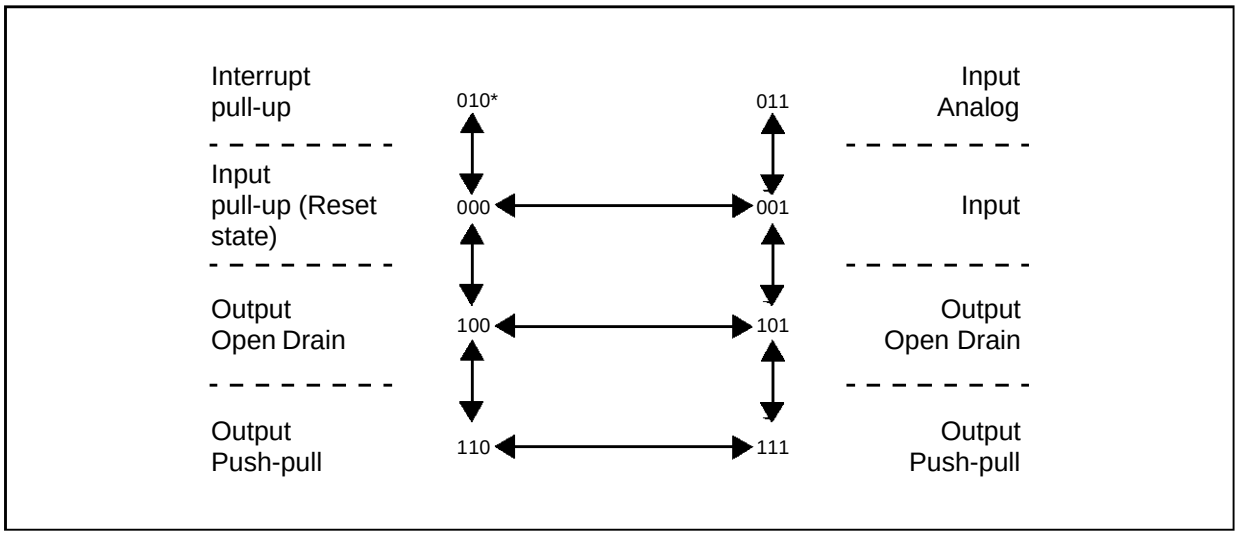
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

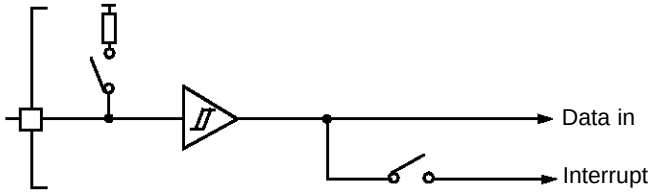
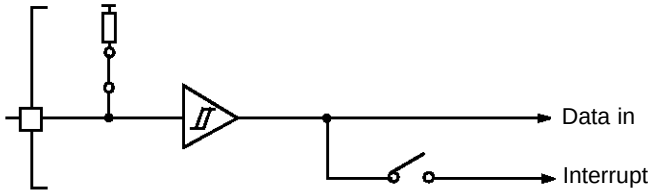
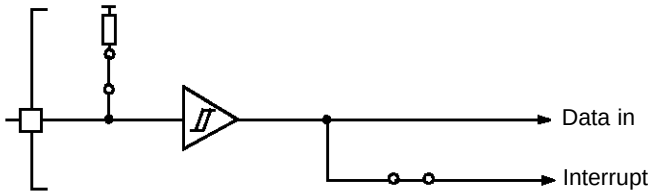
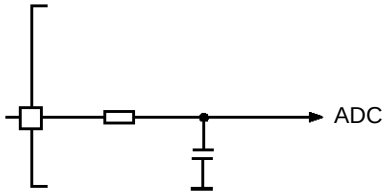
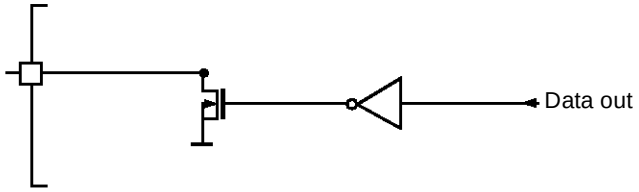
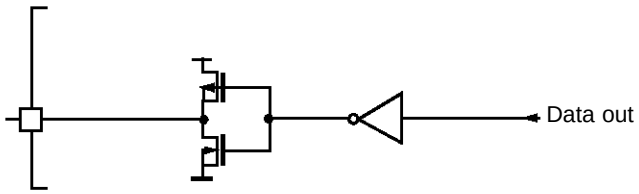
Figure 22. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 7. I/O Port Option Selections

MODE	AVAILABLE ON(1)	SCHEMATIC
Input	PA1-PA3 PB0-PB7	
Input with pull up	PA1-PA3 PB0-PB7	
Input with pull up with interrupt	PA1-PA3 PB0-PB7	
Analog Input	PB3, PB5-PB7	
Open drain output 5mA Open drain output 20mA	PB0-PB7 PA1-PA3	
Push-pull output 5mA Push-pull output 20mA	PB0-PB7 PA1-PA3	

Note 1. Provided the correct configuration has been selected.

6.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}.

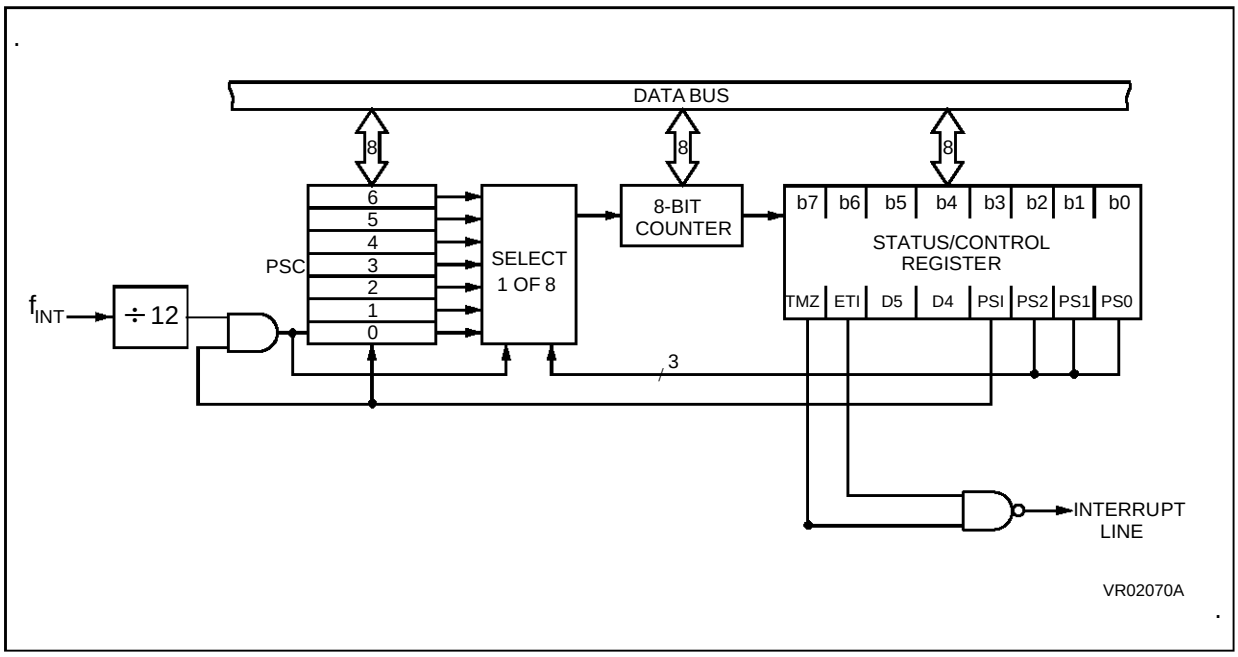
Figure 23 shows the Timer Block Diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero)bit in the TSCR is set. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set, an interrupt request is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input is the internal frequency (f_{INT}) divided by 12. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 8), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to allow the prescaler (and hence the counter) to start. If it is cleared, all the prescaler bits are set and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set. The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 24 illustrates the Timer's working principle.

Figure 23. Timer Block Diagram



TIMER (Cont'd)

6.2.1 Timer Operation

The Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high.

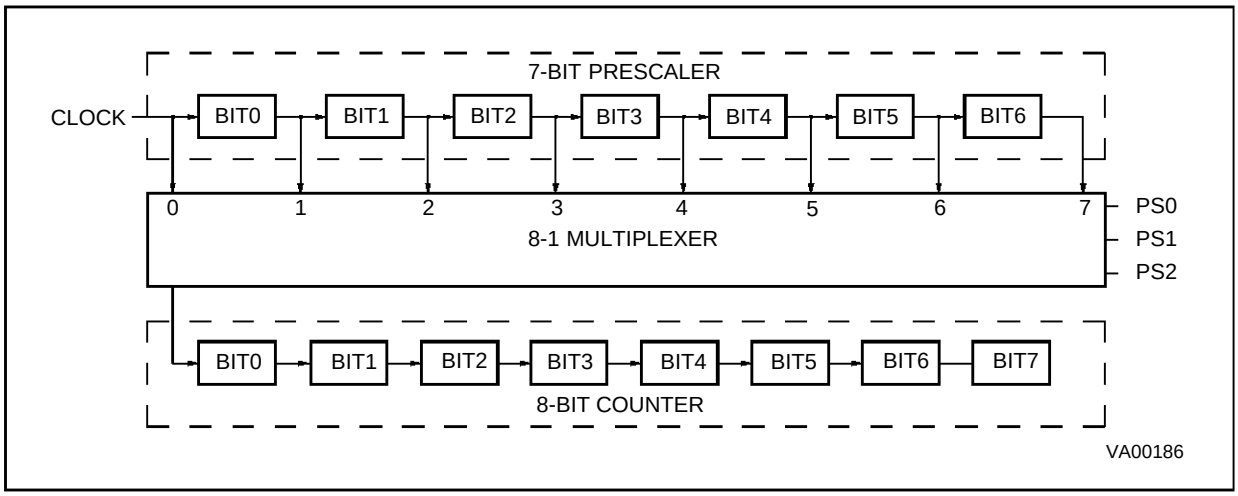
6.2.2 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

6.2.3 Application Notes

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

Figure 24. Timer Working Principle



TIMER (Cont'd)

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

6.2.4 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	D5	D4	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrup*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **D5**: *Reserved*

Must be reset.

Bit 4 = **D4**

When set, the timer is enabled; when reset the timer is disabled.

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 8. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register (TCR)

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

6.3 A/D CONVERTER (ADC)

The A/D converter peripheral is an 8-bit analog to digital converter with analog inputs as alternate I/O functions (the number of which is device dependent), offering 8-bit resolution with a typical conversion time of 70us (at an oscillator clock frequency of 8MHz).

The ADC converts the input voltage by a process of successive approximations, using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, conversion accuracy is decreased.

Selection of the input pin is done by configuring the related I/O line as an analog input via the Option and Data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously, to avoid device malfunction.

The ADC uses two registers in the data space: the ADC data conversion register, ADR, which stores the conversion result, and the ADC control register, ADCR, used to program the ADC functions.

A conversion is started by writing a “1” to the Start bit (STA) in the ADC control register. This automatically clears (resets to “0”) the End Of Conversion Bit (EOC). When a conversion is complete, the EOC bit is automatically set to “1”, in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

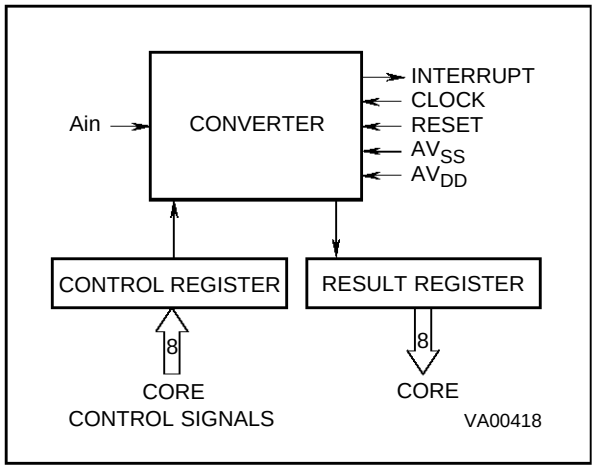
The STA bit is continuously scanned so that, if the user sets it to “1” while a previous conversion is in progress, a new conversion is started before completing the previous one. The start bit (STA) is a write only bit, any attempt to read it will show a logical “0”.

The A/D converter features a maskable interrupt associated with the end of conversion. This interrupt is associated with interrupt vector #4 and occurs when the EOC bit is set (i.e. when a conversion is completed). The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. This is done by setting the PDS bit in the ADC control register to “0”. If PDS=“1”, the A/D is powered and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow stabilisation of the A/D converter. This action is also needed before entering WAIT mode, since the A/D comparator is not automatically disabled in WAIT mode.

During Reset, any conversion in progress is stopped, the control register is reset to 40h and the ADC interrupt is masked (EAI=0).

Figure 25. ADC Block Diagram



6.3.1 Application Notes

The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for the optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 12pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. In worst case conditions, the impedance, ASI, of the analog voltage source is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband. ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

A/D CONVERTER (Cont'd)

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

The accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

The converter resolution is given by::

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

In order to optimise conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the less significant bits are determined.

The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. Indeed, only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. It should be noted that waking up the microcontroller could also be done using the Timer interrupt, but in this case the Timer will be working and the resulting noise could affect conversion accuracy.

A/D Converter Control Register (ADCR)

Address: 0D1h — Read/Write

7							0
EAI	EOC	STA	PDS	D3	D2	D1	D0

Bit 7 = **EAI**: *Enable A/D Interrupt.* If this bit is set to "1" the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

Bit 6 = **EOC**: *End of conversion. Read Only*. This read only bit indicates when a conversion has been completed. This bit is automatically reset to "0" when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to "1".

Bit 5 = **STA***: Start of Conversion. Write Only*. Writing a "1" to this bit will start a conversion on the selected channel and automatically reset to "0" the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

Bit 4 = **PDS***: Power Down Selection.* This bit activates the A/D converter if set to "1". Writing a "0" to this bit will put the ADC in power down mode (idle mode).

Bit 3-0 = **D3-D0.** Not used

A/D Converter Data Register (ADR)

Address: 0D0h — Read only

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0***: 8 Bit A/D Conversion Result.*

7 SOFTWARE

7.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

7.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

7.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 9. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:

X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 10. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 11. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected. The tested bit is shifted into carry.
* . Not Affected

Table 12. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not<M> Affected

Table 13. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ . Affected
*. Not Affected

Table 14. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
* . Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

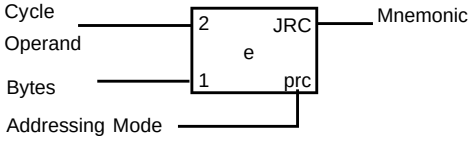
LOW / HI	**0** **0000**	**1** **0001**	**2** **0010**	**3** **0011**	**4** **0100**	**5** **0101**	**6** **0110**	**7** **0111**	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1** **0001**
2 **0010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2** **0010**
3 **0011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3** **0011**
4 **0100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4** **0100**
5 **0101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5** **0101**
6 **0110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	**6** **0110**
7 **0111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	**7** **0111**
8 **1000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	**8** **1000**
9 **1001**	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	**9** **1001**
A **1010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A** **1010**
B **1011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B** **1011**
C **1100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C** **1100**
D **1101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D** **1101**
E **1110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	**E** **1110**
F **1111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	**F** **1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

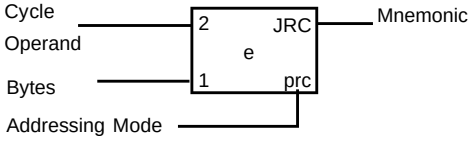
LOW / HI	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	1 0001
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	3 0011
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	5 0101
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	7 0111
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	8 1000
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	9 1001
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	B 1011
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	D 1101
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



8 ELECTRICAL CHARACTERISTICS

8.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determine by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.
- (2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

8.2 THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{thJA}	Thermal Resistance (junction to ambient)	PDIP16			60	
		PSO16			80	

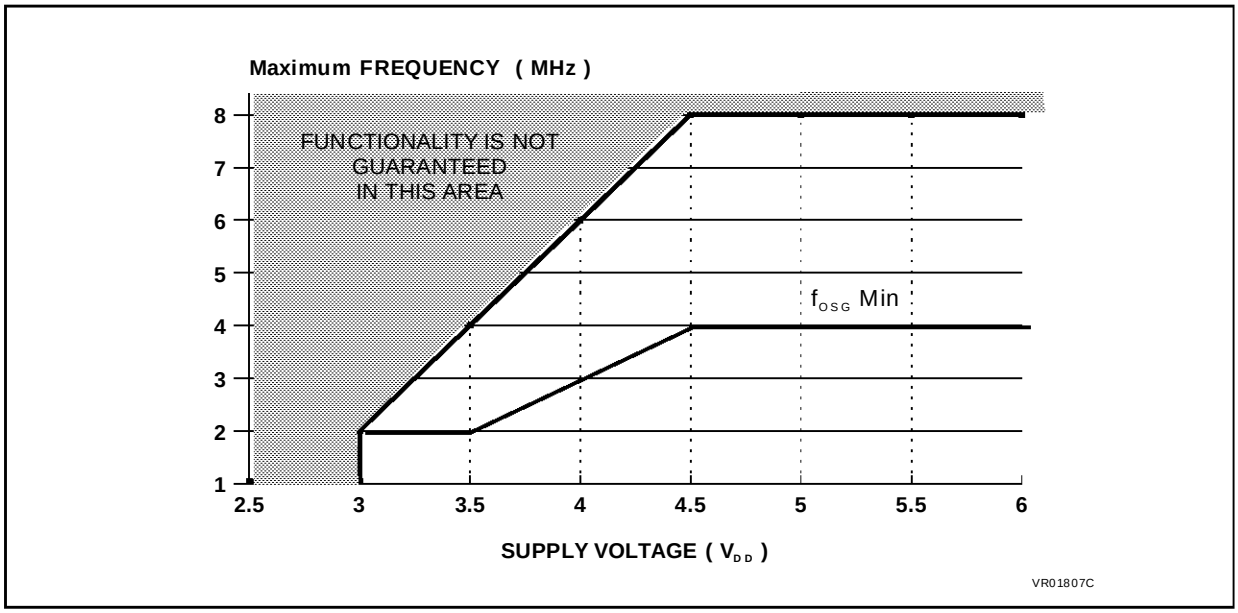
8.3 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

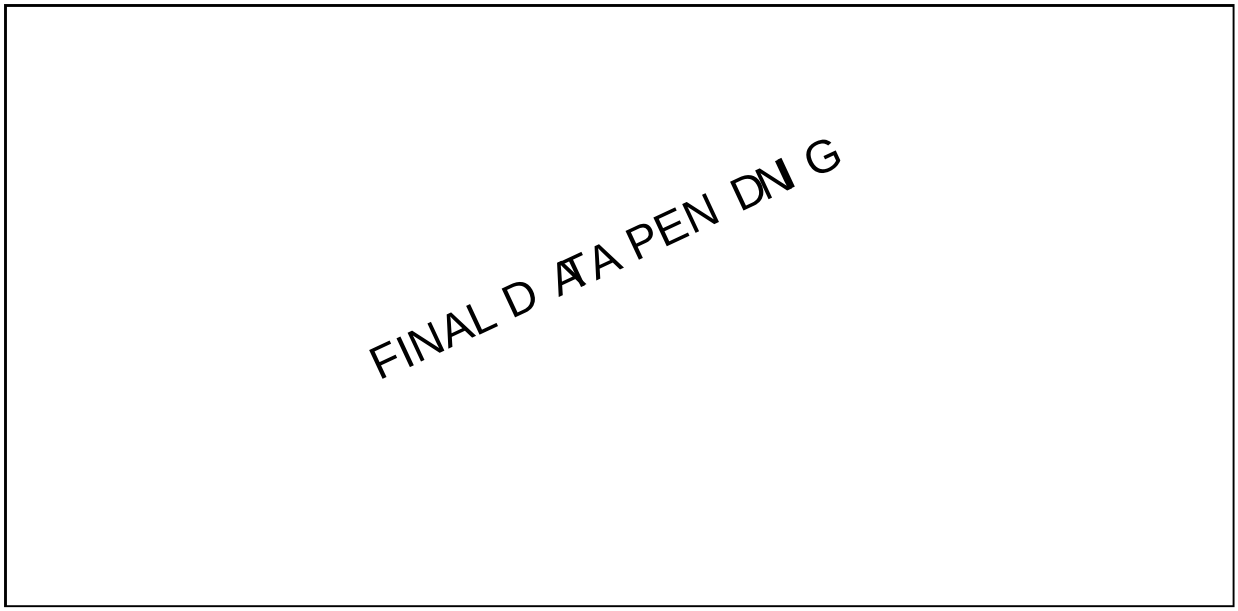
If a total current of +1mA is flowing into a single analog channel, or if the total current flowing into all the analog inputs is 1mA, all resulting A/D conversions will be shifted by + 1 LSB. If a total positive current is flowing into a single analog channel, or if the total current flowing into all analog inputs is 5mA, all the resulting conversions are shifted by + 2 LSB.

Figure 26. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



Note: The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 27. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 28. RC Oscillator. F_{INT} versus RNET (Indicative Values)

The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

8.4 READOUT PROTECTION FUSE

If the ROM READOUT PROTECTION option is selected, the waveform illustrated below must be applied to the TEST pin in order to blow the fuse.

The following circuit can be used for this purpose:

Figure 29. Programming wave form

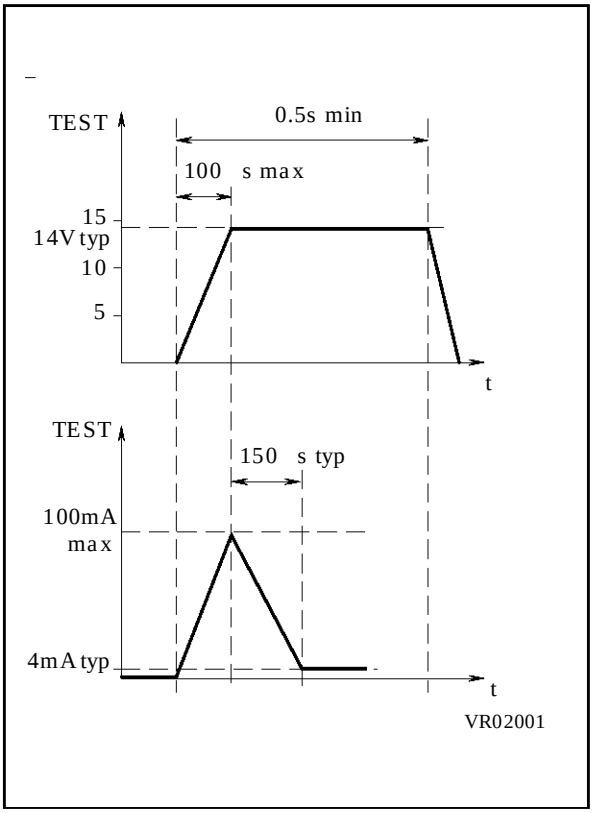
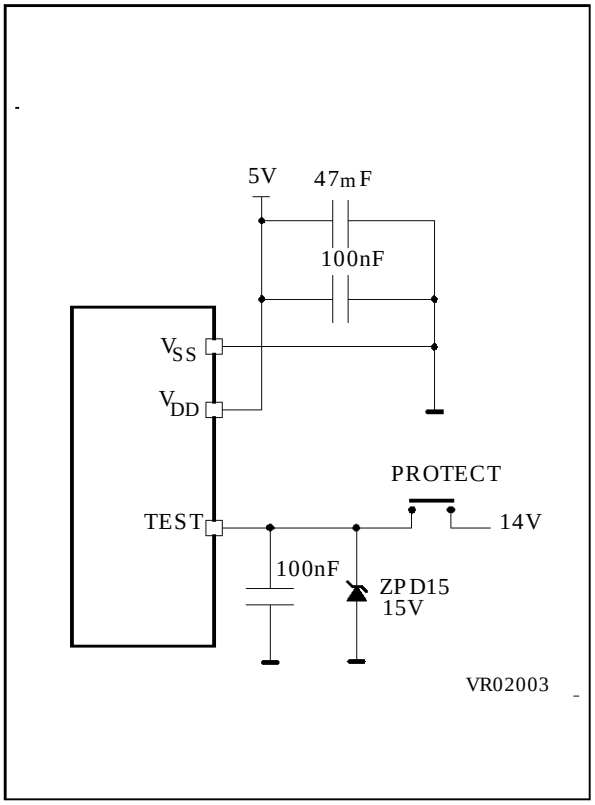


Figure 30. Programming Circuit

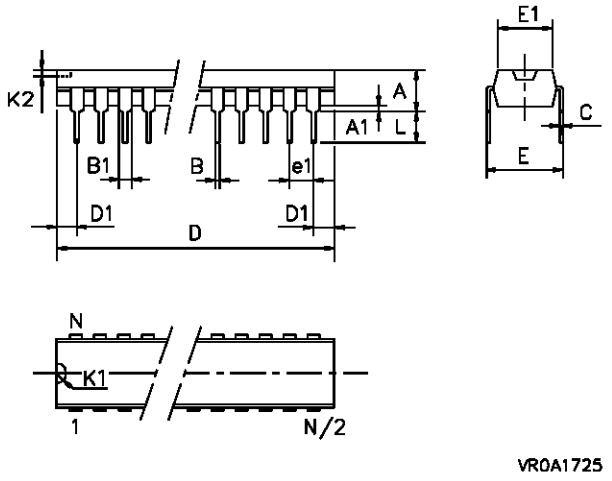


Note: ZPD15 is used for overvoltage protection

9 GENERAL INFORMATION

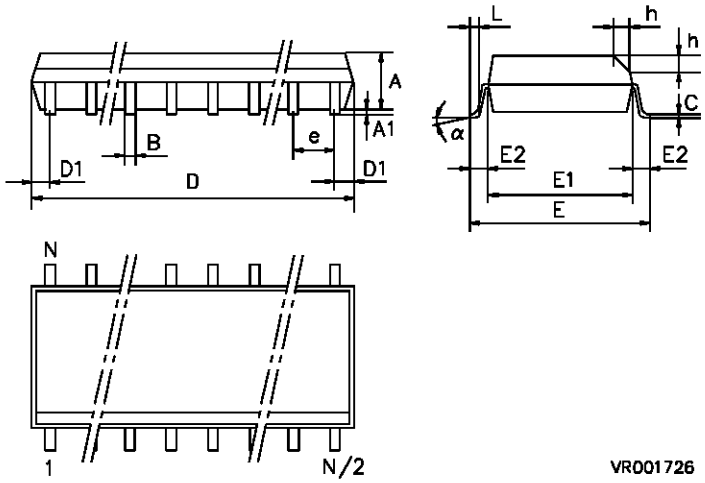
9.1 PACKAGE MECHANICAL DATA

Figure 31. 16-Pin Plastic Dual In Line Package (B), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			.200
A1			.508	.020		
B	.381	.508	.533	.015	.020	.021
B1	.762		1.651	.030		.065
C	.203	.254	.304	.008	.010	.012
D	18.92	19.18	19.56	.745	.755	.770
D1			1.27			.050
E	7.37	7.62	7.874	.290	.300	.310
E1		5.334			.210	
K1						
K2						
L	2.997	3.302	3.708	.118	.130	.146
e	2.286	2.54	2.794	.090	.100	.110
	Number of Pins					
N	16					

Figure 32. 16-Pin Plastic Small Outline Package (M), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		2.286			.090	
A1	.102		.305	0.004		.012
B	.381		.483	.015		.019
C	.229		.254	.009		.010
D	10.26		10.34	.404		.407
D1	-	-	-	-	-	-
E	10.24		10.34	.403		.407
E1	7.44		7.54	.293		.297
E2	-	-	-	-	-	-
L		.832			.033	
e		1.27			.050	
h		.508			.020	
alpha		5^o			5^o	
	Number of Pins					
N	16					

ST6200, 01B MICROCONTROLLER OPTION LIST

Customer
Address

Contact
Phone No
Reference

SGS-THOMSON Microelectronics references
Device: [ ] ST6200B [ ] ST6201B
Package:[ ] Dual in Line Plastic[ ] Small Outline Plastic
In this case, select conditioning
[ ] Standard (Stick)
[ ] Tape & Reel
Temperature Range: [ ] 0°C to + 70°C [ ] - 40°C to + 85°C
Special Marking: [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _ "
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count DIP16: 9 SO16: 6
Oscillator Source Selection: [ ] Crystal Quartz/Ceramic resonator (Default)
[ ] RC Network
Watchdog Selection:[ ] Software Activation (STOP mode available)
[ ] Hardware Activation (no STOP mode)
OSG: [ ] Enabled
[ ] Disabled (Default)
Input pull-up selection on NMI pin:[ ] Yes [ ] No
ROM Readout Protection: [ ] Standard (Fuse cannot be blown)
[ ] Enabled (Fuse can be blown by the customer)
Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.
External STOP Mode Control [ ] Enabled[ ] Disabled (Default)
Comments :
Supply Operating Range in the application:
Oscillator Fequency in the application:
Notes

Signature
Date

9.2 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

9.2.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electrnic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

9.2.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 15. ROM Memory Map for ST6201B

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 16. ROM Memory Map for ST6200B

Device Address	Description
0000h-0B9Fh	Reserved
0BA0h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Table 17. Ordering Information

Sales Type	ROM	I/O	Additional Features	Temperature Range	Package
ST6200BB1/XXX ST6200BB6/XXX	1036K Bytes	9	A/D CONVERTER	0 to +70°C -40 to + 85°C	PDIP16
ST6201BM1/XXX ST6201BM6/XXX	1836K Bytes			0 to +70°C -40 to + 85°C	PSO16

Notes:

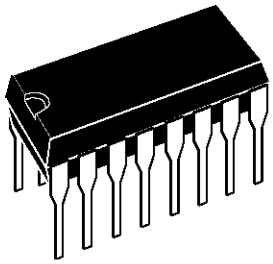
SGS-THOMSON MICROELECTRONICS

ST62T03

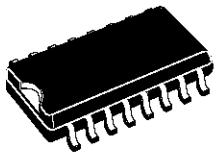
8-BIT OTP MCUs

PRODUCT PREVIEW

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 4 Interrupt Vectors
- Look-up Table capability in OTP
- Data OTP: User selectable size (in program OTP)
- Data RAM: 64 bytes
- 9 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
- 3 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer with 7-bit programmable prescaler
- Digital Watchdog
- On-chip Clock oscillator can be driven by Quartz crystal or Ceramic resonator
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP16



PSO16

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	OTP (Bytes)	I/O Pins
ST62T03	1036	9

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T03 device is a low cost member 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.

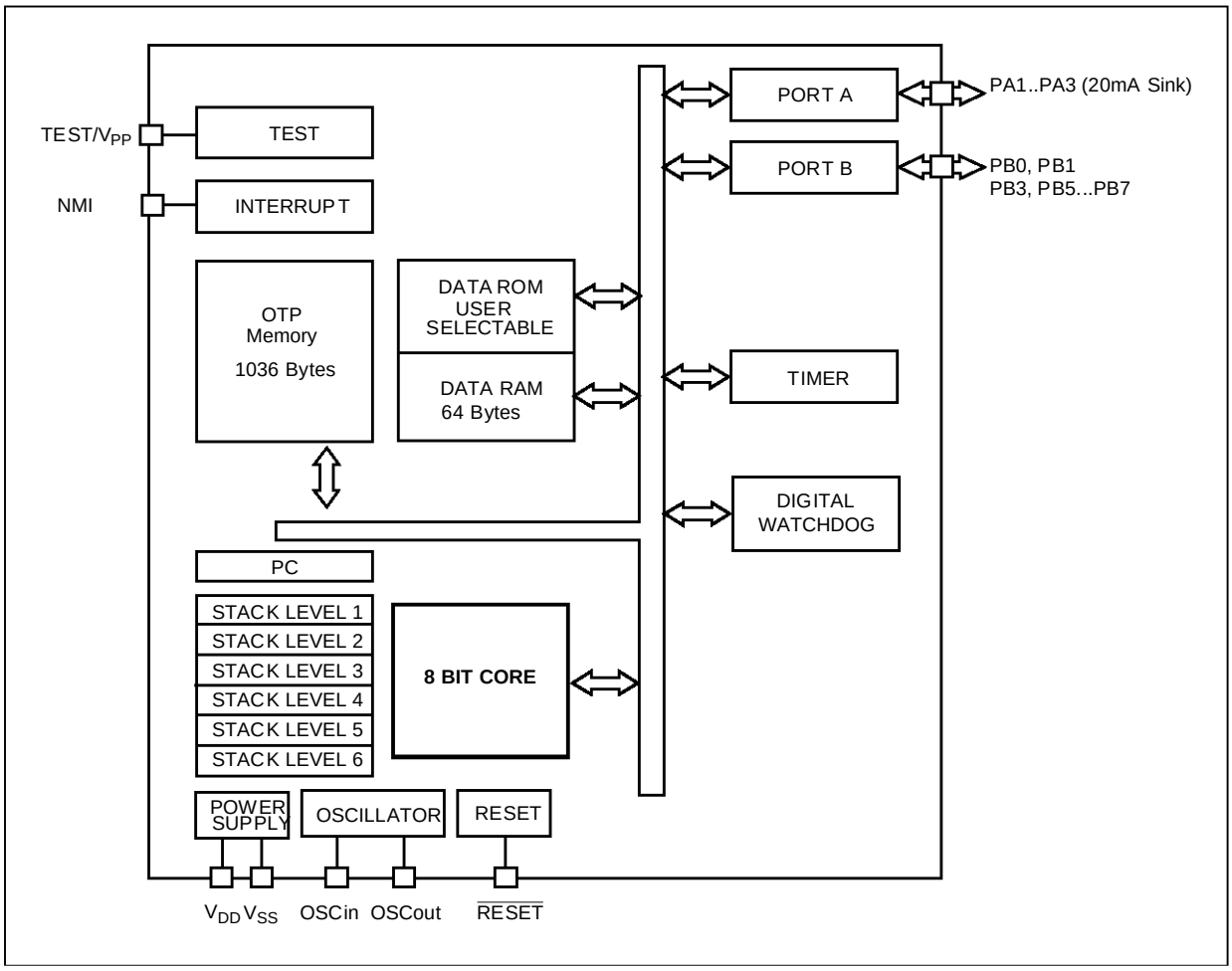
OTP devices are functionally identical to their ROM counterparts; in addition, the ROM based versions offer the following additional features: RC Oscillator, Oscillator Safeguard, external Stop mode control, program code readout protection and an optional internal pullup on the NMI pin.

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 ST62E01 (2K EPROM) device may be used to emulate the ST62T03 OTP device, but care should be taken not to exceed the memory size of the T03 device.

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

Figure 1. Block Diagram



1.2 PIN DESCRIPTIONS

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

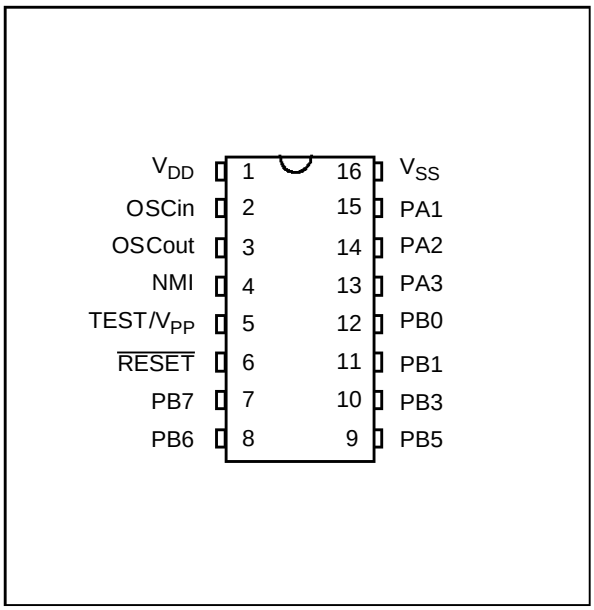
TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation. If TEST pin is connected to a +12.5V level during the reset phase, the EPROM/OTP programming Mode is entered.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. A pull-up device must be provided externally on OTP and EPROM devices.

PA1-PA3. These 3 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs. **PA1-PA3** can also sink 20mA for direct LED driving.

PB0, PB1, PB3, PB5-PB7. These 6 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs.

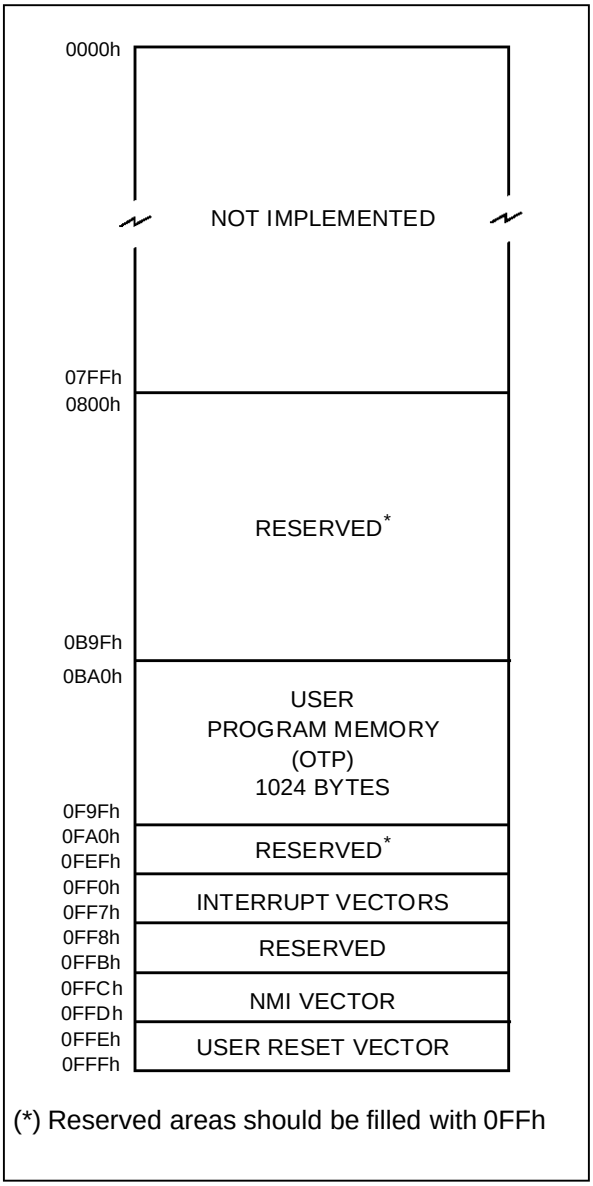
Figure 1. ST62T03 Pin Configuration



1.3 MEMORY MAP

1.3.1 Program Memory Map

Figure 2. ST62T03 Program Memory Map



(*) Reserved areas should be filled with 0FFh

1.3.2 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.

The Data Space is fully described and illustrated on page 18.

1.4 PARTICULARITIES OF OTP DEVICES

OTP and EPROM devices are identical save for the package which, in the EPROM device, is fitted with a transparent window to allow erasure of memory contents by exposure to UV light.

Both OTP and EPROM parts may be programmed using programming equipment approved by SGS-THOMSON.

1.4.1 OTP Programming

Programming mode is selected by applying a 12.5V voltage to the V_{PP}/TEST pin during reset. Programming of OTP and EPROM parts is fully described in the EPROM Programming Board User Manual.

1.4.2 Eprom Erasure

Thanks to the transparent window present in the EPROM package, its memory contents may be erased by exposure to UV light.

Erasure begins when the device is exposed to light with a wavelength shorter than 4000Å. It should be noted that sunlight, as well as some types of artificial light, includes wavelengths in the 3000-4000Å range which, on prolonged exposure, can cause erasure of memory contents. It is thus recommended that EPROM devices be fitted with an opaque label over the window area in order to prevent unintentional erasure.

The recommended erasure procedure for EPROM devices consists of exposure to short wave UV light having a wavelength of 2537Å. The minimum recommended integrated dose (intensity x exposure time) for complete erasure is 15Wsec/cm^2. This is equivalent to an erasure time of 15-20 minutes using a UV source having an intensity of 12mW/cm^2 at a distance of 25mm (1 inch) from the device window.

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

The CPU Core may be thought of as an independent central processor communicating with on-chip I/O, memory and peripherals. For further details refer to page 14.

2.2 CPU REGISTERS

The CPU Core features six registers and three pairs of flags available to the programmer. For a detailed description refer to page 20.

3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The Oscillator may be driven by an external clock, or by a crystal or ceramic resonator. ROM devices also offer RC oscillator and Oscillator Safeguard features. For a complete description refer to page 22.

3.2 RESETS

The MCU can be reset in three ways: by the external Reset input being pulled low, by the Power-on Reset circuit, or by the Digital Watchdog timing out. For further details refer to page 26.

3.3 DIGITAL WATCHDOG

The Digital Watchdog can be used to provide controlled recovery from software upsets. Software and Hardware enabled Watchdog options are available in order to achieve optimum trade-off between power consumption and noise immunity. For a complete description and a selection guide refer to page 29.

3.4 INTERRUPTS

The CPU can manage four Maskable and one Non-Maskable Interrupt source. Each source is associated with a specific Interrupt Vector. An internal pullup option on the NMI pin is available on ROM devices. For a complete description refer to page 33.

3.5 POWER SAVING MODES

WAIT mode reduces electrical consumption during idle periods, while STOP mode achieves the lowest power consumption by stopping all CPU activity. For a complete description refer to page 37.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

Input/Output lines may be individually programmed as one of a number of different configurations. For further details refer to page 39.

4.2 TIMER

The on-chip Timer peripheral consists of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. For a complete description refer to page 43.

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 architecture has been designed to exploit the hardware in the most efficient way possible, while keeping byte usage to a minimum. For further details refer to page 46.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes: Immediate, Direct, Short Direct, Extended, Program Counter Relative, Bit Direct, Bit Test & Branch, Indirect, and Inherent. For a complete description of the available addressing modes, refer to page 46.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes; these may be subdivided into six types: load/store, arithmetic/logic, conditional branch, control, jump/call, and bit manipulation. For further details refer to page 47.

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices designed to protect the inputs against damage due to high static voltages; however, it is advisable to take normal precautions to avoid applying voltages higher than the specified maximum ratings.

For proper operation, it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, T_j, in degrees Celsius can be obtained from:

$$T_j = T_A + P_D \times R_{thJA}$$

Where:

T_A = Ambient Temperature.

R_{thJA} =Package thermal resistance (junction-to ambient).

$P_D = P_{int} + P_{port}$

$P_{int} = I_{DD} \times V_{DD}$ (chip internal power).

P_{port} =Port power dissipation (to be determined by the user)

.

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_{PP}	OTP/EPROM Programming Voltage	13	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into V_{DD} (source)	50(2)	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

Stresses above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

(1) Within these limits, clamping diodes are non-conducting. Voltages outside these limits are authorised provided injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

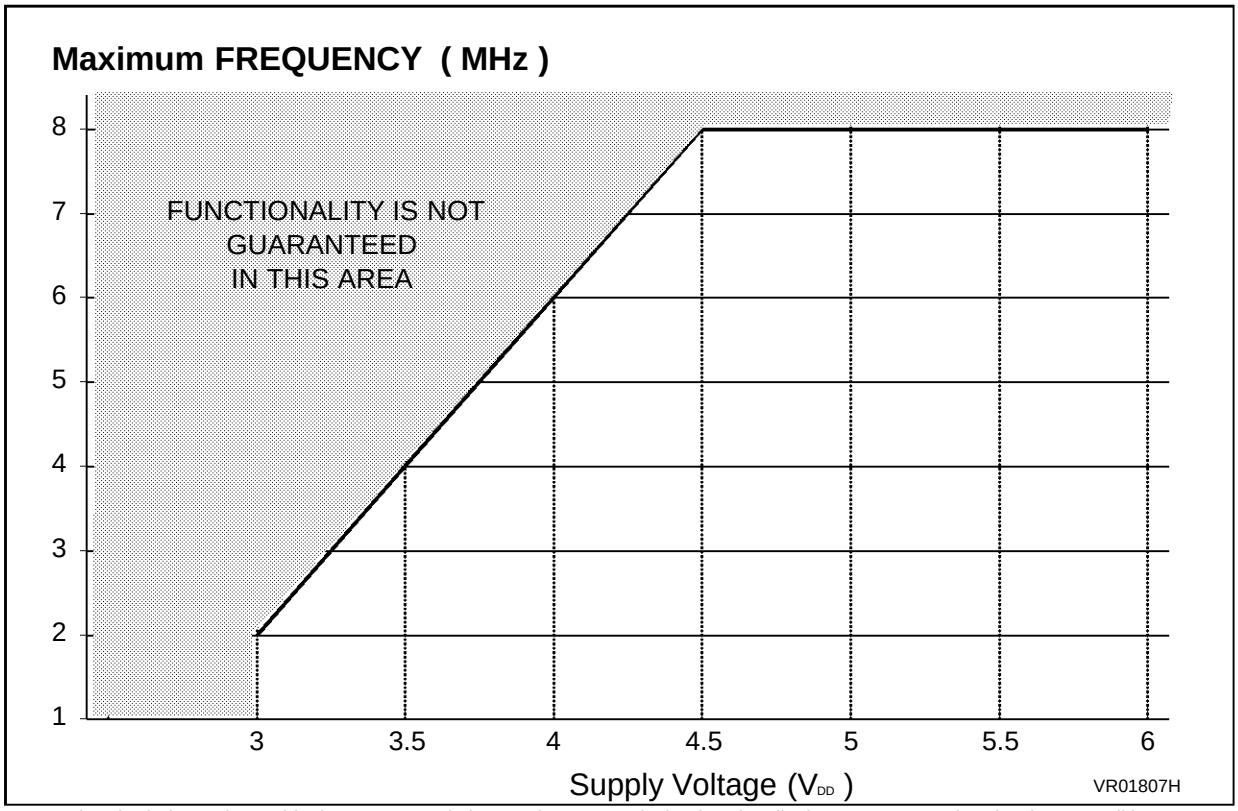
6.2 THERMAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{thJA}	Thermal Resistance (junction to ambient)	PDIP16			60	°C/W
		PSO16			80	

6.3 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input	V_{DD} = 4.5 to 5.5V			-5	mA

Figure 2. Maximum Operating FREQUENCY (F_{MAX}) Versus SUPPLY VOLTAGE (V_{DD})

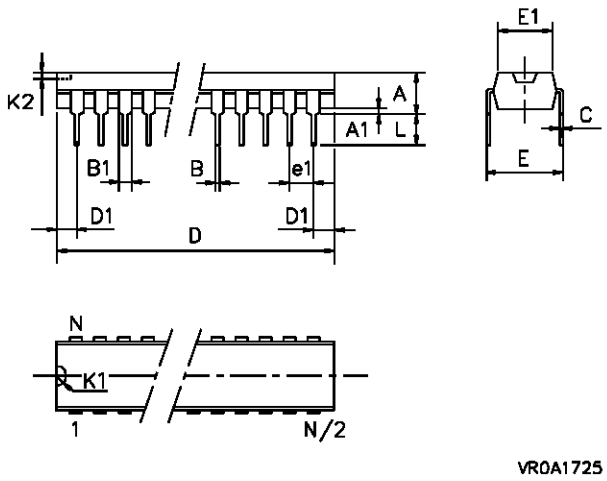


The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

7 GENERAL INFORMATION

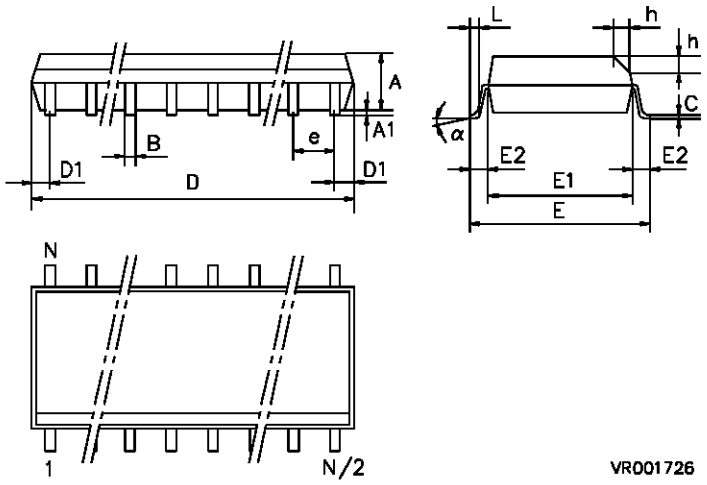
7.1 PACKAGE MECHANICAL DATA

Figure 3. 16-Pin Plastic Dual In Line Package (B), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			.200
A1			.508	.020		
B	.381	.508	.533	.015	.020	.021
B1	.762		1.651	.030		.065
C	.203	.254	.304	.008	.010	.012
D	18.92	19.18	19.56	.745	.755	.770
D1			1.27			.050
E	7.37	7.62	7.874	.290	.300	.310
E1		5.334			.210	
K1						
K2						
L	2.997	3.302	3.708	.118	.130	.146
e	2.286	2.54	2.794	.090	.100	.110
	Number of Pins					
N	16					

Figure 4. 16-Pin Plastic Small Outline Package (M), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		2.286			.090	
A1	.102		.305	0.004		.012
B	.381		.483	.015		.019
C	.229		.254	.009		.010
D	10.26		10.34	.404		.407
D1	-	-	-	-	-	-
E	10.24		10.34	.403		.407
E1	7.44		7.54	.293		.297
E2	-	-	-	-	-	-
L		.832			.033	
e		1.27			.050	
h		.508			.020	
alpha		5^o			5^o	
	Number of Pins					
N	16					

Table 1. OTP Device Sales Types

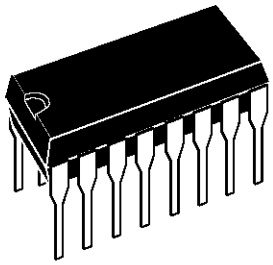
Sales Type	I/O Pins	Option	Temperature range	Package
ST62T03B6/HWD	9	Hardware Watchdog	-40°C TO +85°C	PDIP16
ST62T03B6/SWD		Software Watchdog		
ST62T03M6/HWD		Hardware Watchdog		PSO16
ST62T03M6/SWD		Software Watchdog		

SGS-THOMSON MICROELECTRONICS

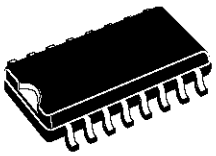
ST6203B

8-BIT HCMOS MCU

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 4 Interrupt Vectors
- Look-up Table capability in ROM
- Data ROM: User selectable size (in program ROM)
- Data RAM: 64 bytes
- ROM read-out Protection
- 9 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
- 3 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- Digital Watchdog
- Oscillator Safe Guard
- On-chip Clock oscillator can be driven by Quartz crystal, Ceramic resonator or RC network
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP16



PSO16

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	I/O Pins
ST6203B	1036	9

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

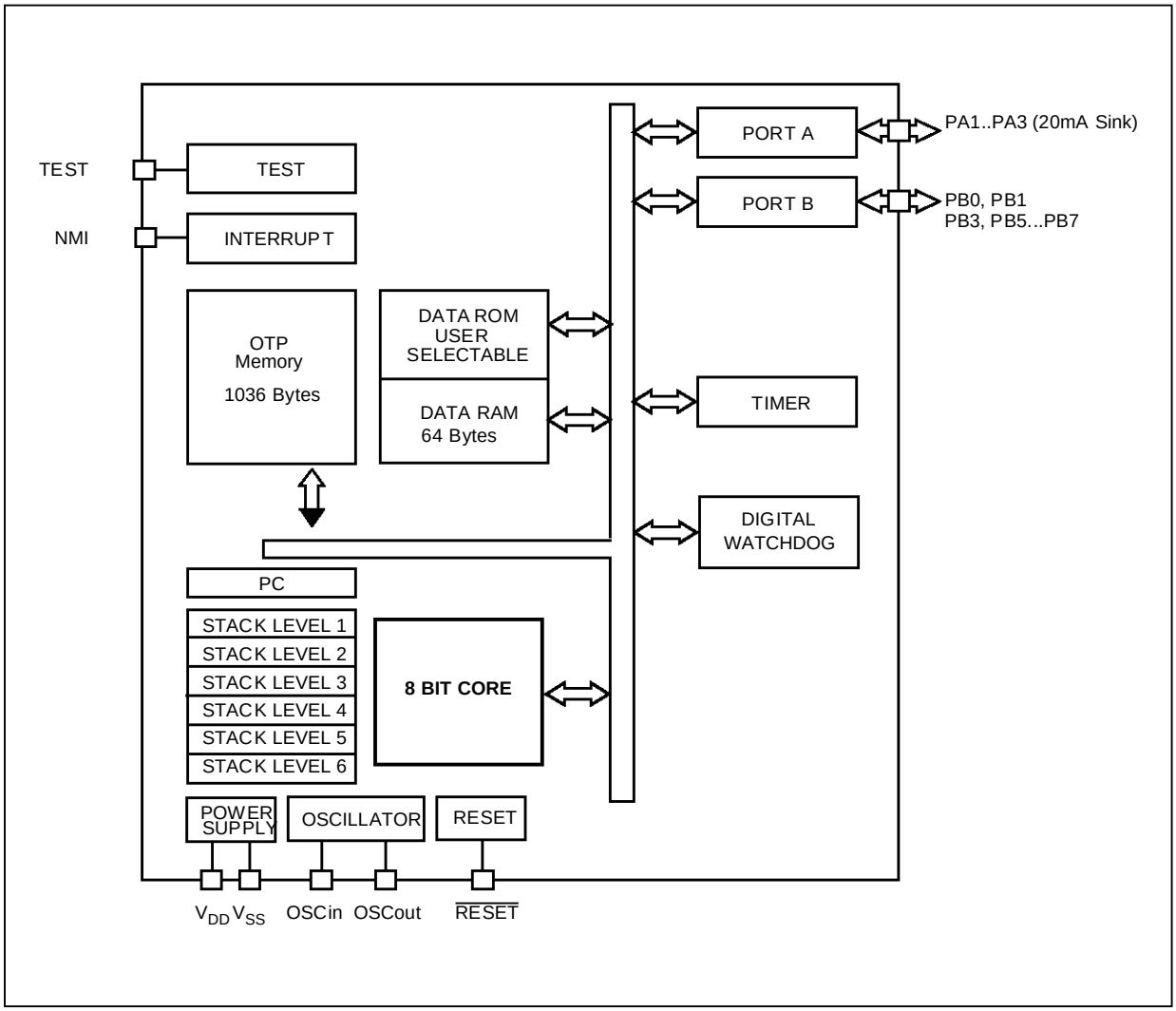
The ST6203B microcontroller is a member of the 8-bit HCMOS ST62xx family of devices, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST6203B features the following peripherals: a Timer comprising an 8-bit counter equipped with a 7-bit software programmable prescaler and a Digital Watchdog timer.

The ST6203B features a choice of Quartz, Ceramic or RC oscillators, an Oscillator Safe Guard circuit, Read-out Protection against unauthorised copying of program code, and an External STOP Mode Control option to enlarge the range of power consumption versus reliability trade-offs.

These devices are well suited for automotive, appliance and industrial applications. The user programmable part for program development is the ST62E01, which is a pin compatible device with 2Kbytes of EPROM. Care must be taken to only use the memory available on the ST6203B when using the ST62E20 for evaluation or development.

Figure 3. Block Diagram



1.2 PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. When the QUARTZ/CERAMIC RESONATOR Mask Option is selected, a quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. When the RC OSCILLATOR Mask Option is selected, a resistor must be connected between the OSCout pin and ground. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

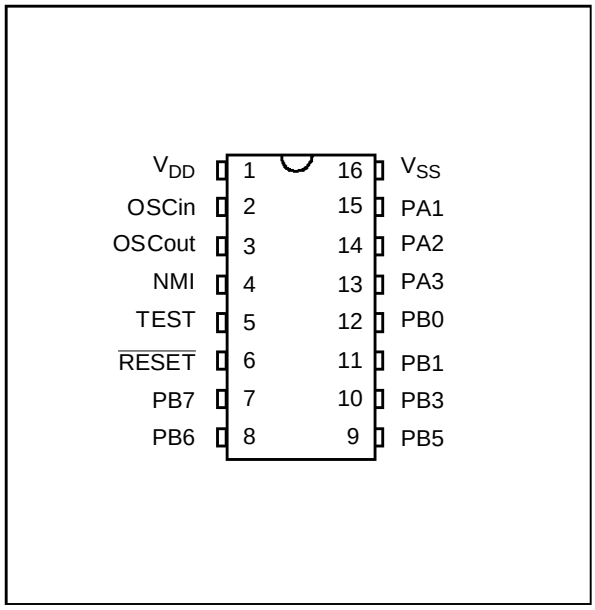
TEST. The TEST pin must be held at V_{SS} for normal operation (an internal 100kΩ pull-down resistor selects normal operating mode if the TEST pin is not connected externally).

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI is falling edge sensitive. A ROM mask option makes available an on-chip pull-up on the NMI pin.

PA1-PA3. These 3 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs. **PA1-PA3** can also sink 20mA for direct LED driving.

PB0, PB1, PB3, PB5-PB7. These 6 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs.

Figure 4. ST6203B Pin Configuration



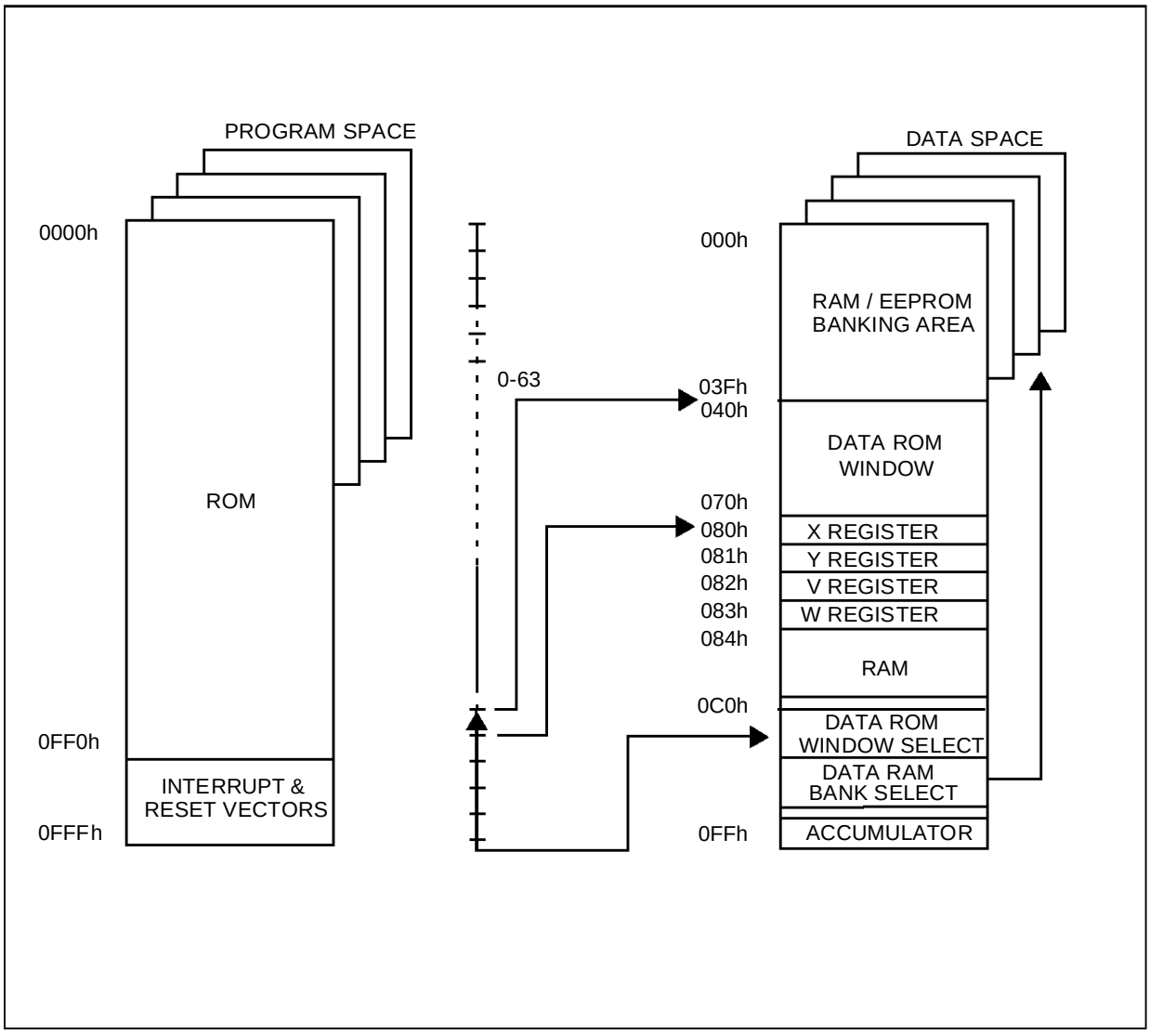
1.3 MEMORY MAP

1.3.1 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 ROM and user vectors; Data space contains user data in RAM and in ROM, and Stack space accommodates six levels of stack for subroutine and interrupt service routine nesting.

Figure 5. Memory Addressing Diagram



MEMORY MAP (Cont'd)

1.3.2 Program Space

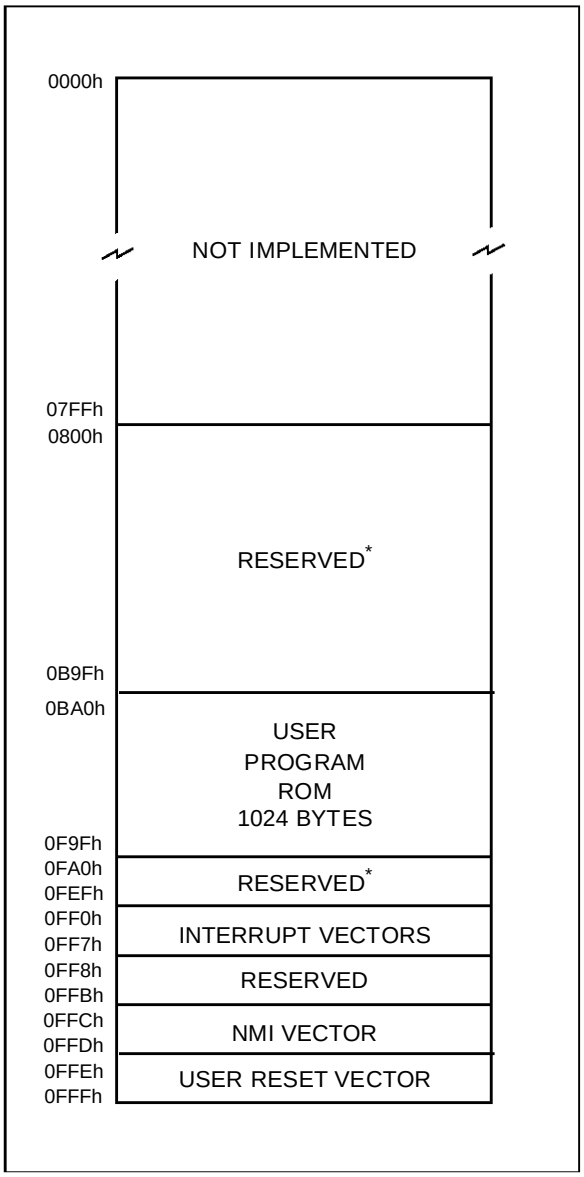
Program Space is physically implemented in ROM memory. It comprises the instructions to be executed, the data required for immediate addressing mode instructions, the reserved factory test area and the user vectors. Program Space is addressed via the 12-bit Program Counter register (PC register)

1.3.2.1 ROM Protection

The ST6203B Program Space can be protected against external readout of ROM contents when the READOUT PROTECTION mask option is chosen. This option allows the user to blow a dedicated fuse on the silicon, by applying a high voltage at V_{PP} (see detailed information in the "Electrical Specification").

Note: Once the Readout Protection fuse is blown, it is no longer possible, even for SGS-THOMSON, to gain access to the ROM contents. Returned parts with a blown fuse can therefore not be accepted.

Figure 6. ST6203B ROM Memory Map



(*) Reserved areas should be filled with 0FFh

MEMORY MAP (Cont'd)

1.3.3 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 ROM.

1.3.3.1 Data ROM

All read-only data is physically stored in ROM memory, which also accommodates the Program Space. The ROM 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 ROM.

1.3.3.2 Data RAM

In ST6203B devices, the data space includes 60 bytes of RAM, 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 (DRW register).

1.3.4 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.

Table 1. ST6203B Data Memory Space

NOT IMPLEMENTED	000h 03Fh
DATA ROM WINDOW 64 BYTES	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
RESERVED	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
RESERVED	0CEh
RESERVED	0CFh
RESERVED	0D0h
OSCILLATOR CONTROL REGISTER	0D1h
TIMER PSC REGISTER	0D2h
TIMER DATA REGISTER	0D3h
TIMER TSCR REGISTER	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

1.3.5 Data Window Register (DWR)

The Data ROM window is located from address 0040h to address 007Fh in Data space. It allows direct reading of 64 consecutive bytes located anywhere in ROM memory, between address 0000h and 1FFFh (top memory address depends on the specific device). All the ROM memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the ROM memory by writing the appropriate code in the Write-only Data Window register (DWR register, location 00C9h).

The DWR register can be addressed like any RAM location in the Data Space at address 00C9h, it is however a write-only register and cannot be accessed using single-bit operations. This register is used to move the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as data in ROM memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 7). So when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in ROM is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data ROM window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

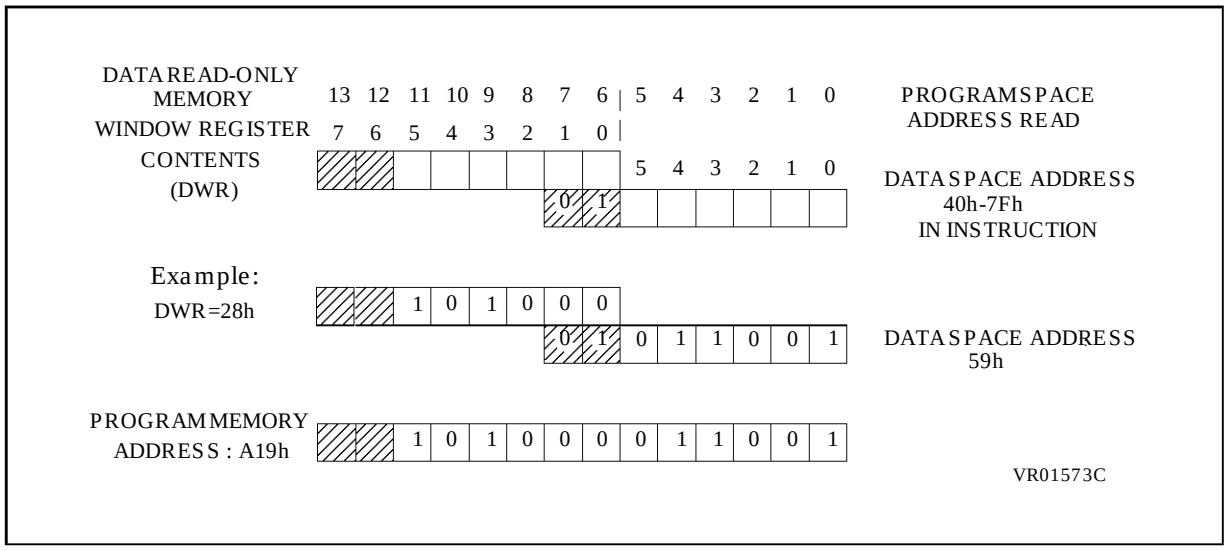
Bit 7 = This bit is not used.

Bit 6-0 = **DWR6-DWR0:** *Data ROM Window Register Bits.* These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

Caution: *This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.*

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

Figure 7. Data ROM Window Memory Addressing



2 CENTRAL PROCESSING UNIT

2.1 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. In-core communication is arranged as shown in Figure 8; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

2.2 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.

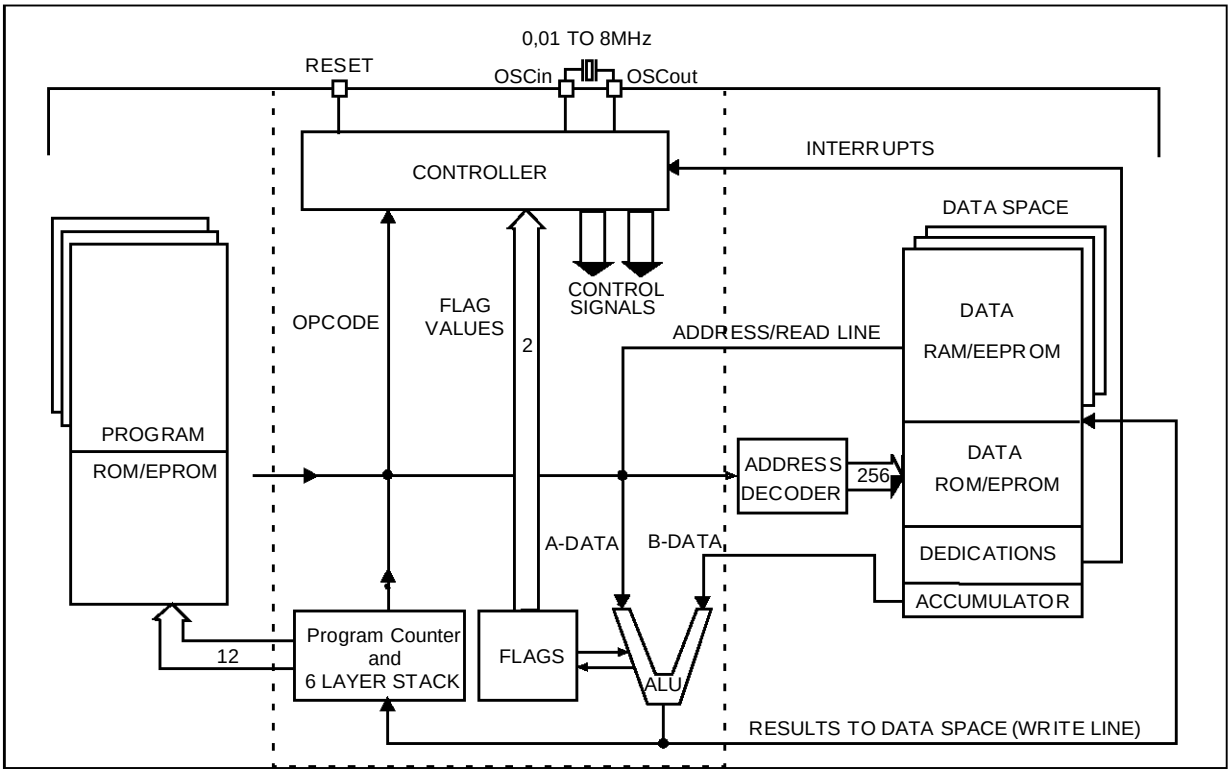
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 8. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

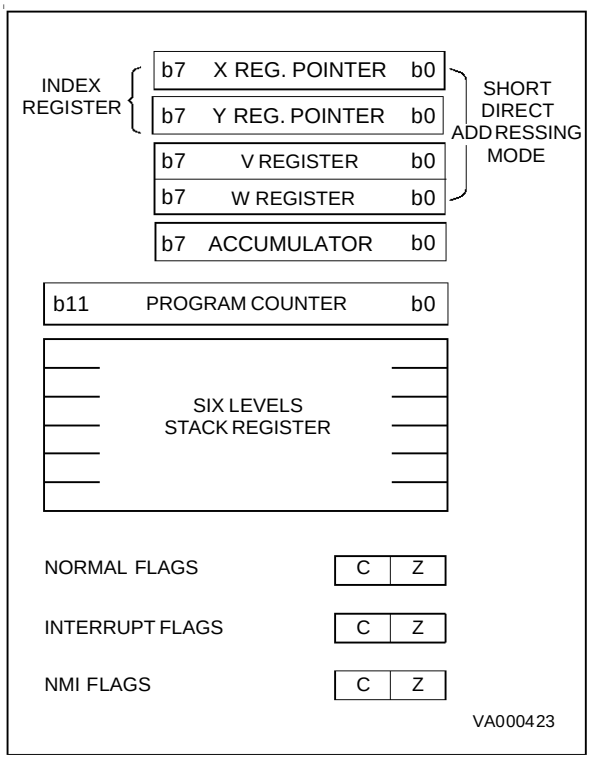
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 9. ST6 CPU Programming Mode



3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}). In addition, a Low Frequency Auxiliary Oscillator (LFAO) can be switched in for security reasons, to reduce power consumption, or to offer the benefits of a back-up clock system.

The Oscillator Safeguard (OSG) option filters spikes from the oscillator lines, provides access to the LFAO to provide a backup oscillator in the event of main oscillator failure and 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 11, Figure 12, Figure 13 and Figure 14.

Figure 10 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. C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 28 and Figure 29.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 13.

With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

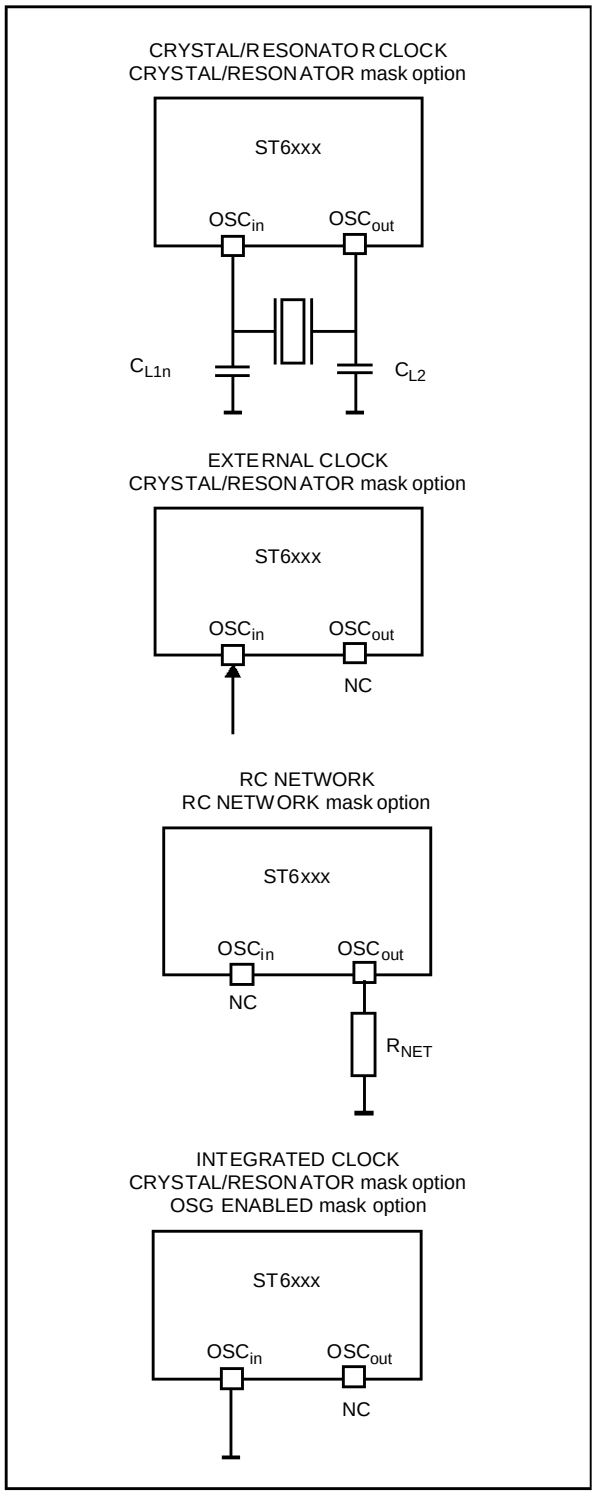
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate mask option. 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 main oscillator can be turned off (when the OSG ENABLED mask option is selected) by setting the OSCOFF bit of the Oscillator Control Register (0D1h). The Low Frequency Auxiliary Oscillator is automatically started.

Figure 10. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Turning on the main oscillator is achieved by resetting the OSCOFF bit of the Oscillator Control Register or by resetting the MCU. Restarting the main oscillator implies a delay comprising the oscillator start up delay period plus the duration of the software instruction at f_{LFAO} clock frequency.

3.1.2 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 safety oscillator in case of main oscillator failure.

This oscillator is available when the OSG ENABLED mask option is selected. In this case, it automatically starts one of its periods after the first missing edge from the main oscillator, whatever the reason (main oscillator defective, no clock circuitry provided, main oscillator switched off...).

User code, normal interrupts, WAIT and STOP instructions, are processed as normal, at the reduced f_{LFAO} frequency.

At power on, the Low Frequency Auxiliary Oscillator starts faster than the Main Oscillator. It therefore feeds the on-chip counter generating the POR delay until the Main Oscillator runs.

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

OSCR

Address: 0D1h — Read/Write

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

Bit 7-3, 1-0= **OSCR7-OSCR3, OSCR1-OSCR0**: *ADC Control Register.* These bits are not used.

Bit 2 = **OSCOFF**. When low, this bit enables main oscillator to run. The main oscillator is switched off when OSCOFF is high.

3.1.3 Oscillator Safe Guard

The Oscillator Safe Guard (OSG) affords drastically increased operational integrity in ST62xx devices. The OSG circuit provides three basic functions: it filters spikes from the oscillator lines which would result in over frequency to the ST62 CPU; it gives access to the Low Frequency Auxiliary Oscillator (LFAO), used to ensure minimum processing in case of main oscillator failure, to offer reduced power consumption or to provide a fixed frequency low cost oscillator; finally, it automatically limits the internal clock frequency as a function of supply voltage, in order to ensure correct operation even if the power supply should drop.

The OSG is enabled or disabled by choosing the relevant OSG mask option. It may be viewed as a filter whose cross-over frequency is device dependent.

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 11). 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. This relationship is illustrated in Figure 14.

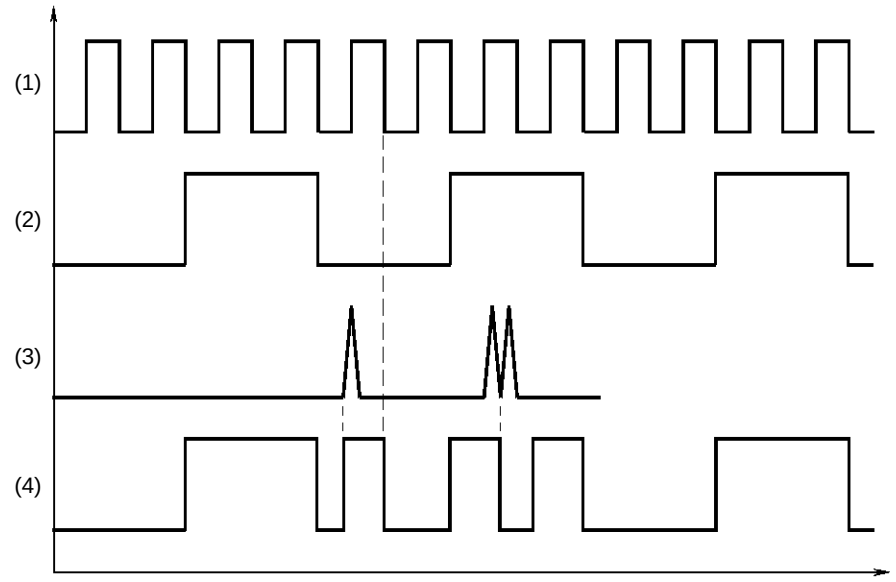
When the OSG is enabled, the Low Frequency Auxiliary Oscillator may be accessed. This oscillator starts operating after the first missing edge of the main oscillator (see Figure 12).

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.

Note. The OSG should be used wherever possible as it provides maximum safety. Care must be taken, however, as it can increase power consumption and reduce the maximum operating frequency to f_{OSG}.

CLOCK SYSTEM (Cont'd)

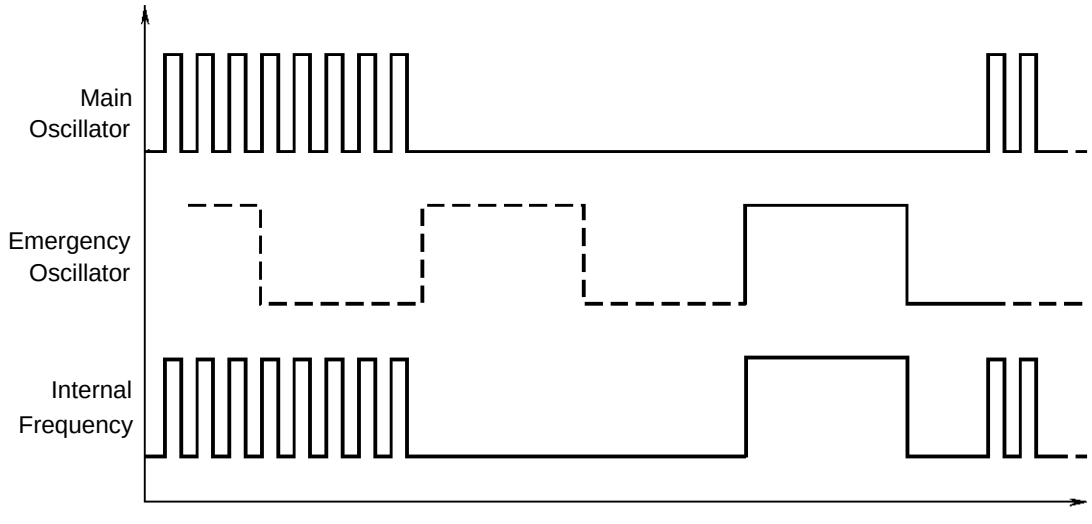
Figure 11. OSG Filtering Principle



(1) Maximum Frequency for the device to work correctly
(2) Actual Quartz Crystal Frequency at OSCin pin
(3) Noise from OSCin
(4) Resulting Internal Frequency

VR001932

Figure 12. OSG Emergency Oscillator Principle



VR001933

CLOCK SYSTEM (Cont'd)

Figure 13. Clock Circuit Block Diagram

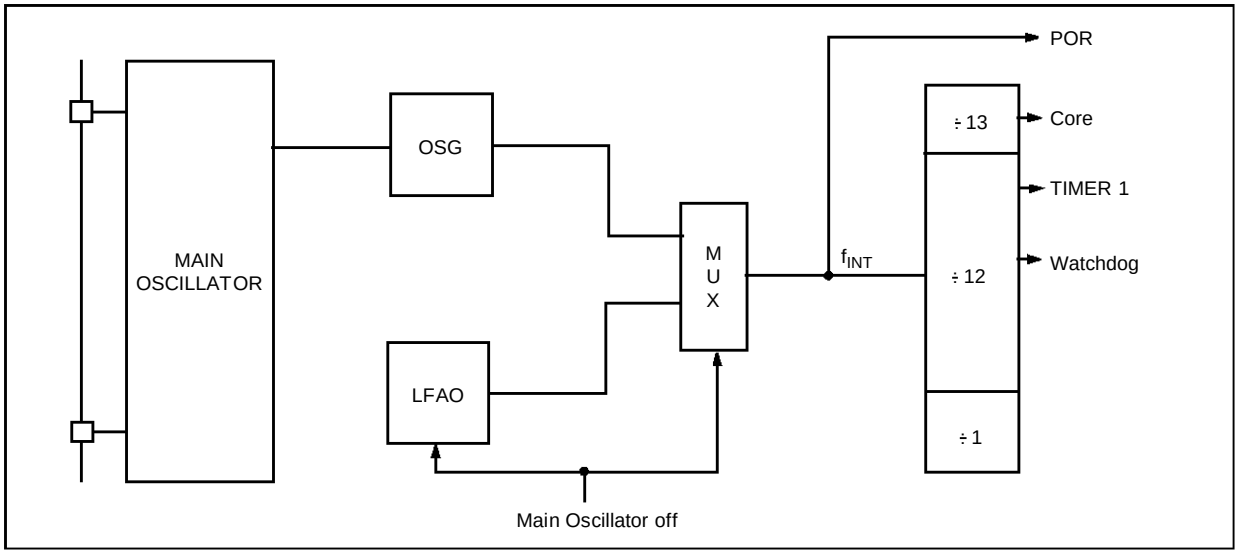
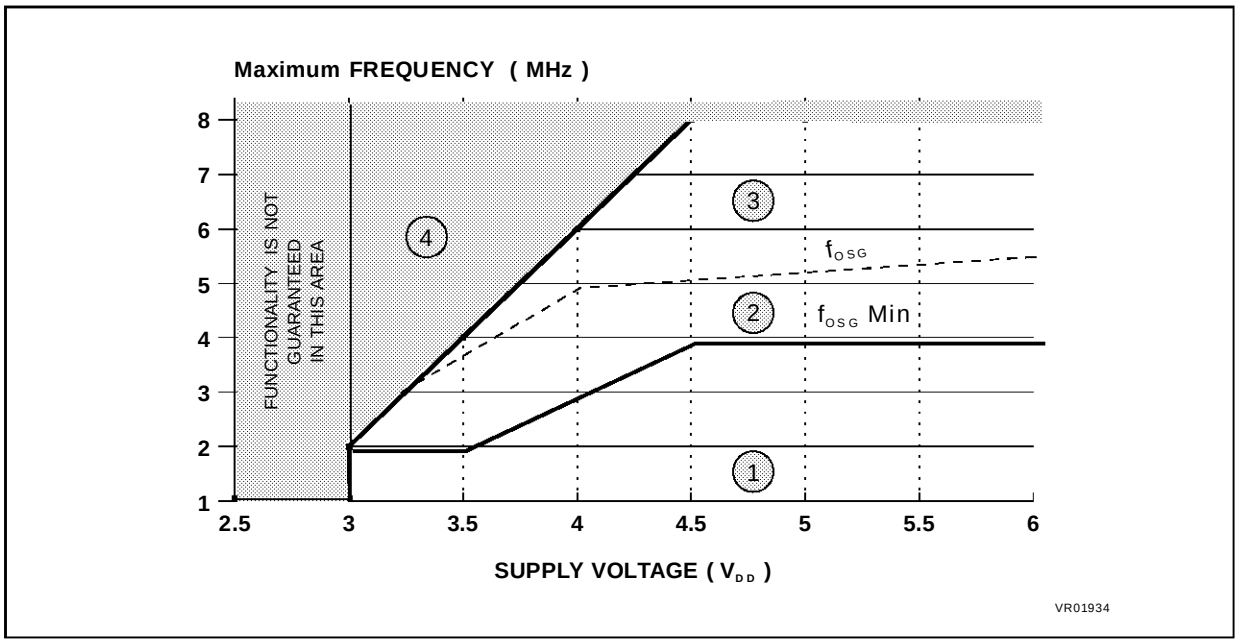


Figure 14. Maximum Operating Frequency (f_{MAX}) versus Supply Voltage (V_{DD})



Notes:

1. In this area, operation is guaranteed at the quartz crystal frequency.
2. When the OSG is disabled, operation in this area is guaranteed at the crystal frequency. When the OSG is enabled, operation in this area is guaranteed at a frequency of at least $f_{OSG\ Min.}$
3. When the OSG is disabled, operation in this area is guaranteed at the quartz crystal frequency. When the OSG is enabled, access to this area is prevented. The internal frequency is kept a f_{OSG}.
4. When the OSG is disabled, operation in this area is not guaranteed
When the OSG is enabled, access to this area is prevented. The internal frequency is kept at f_{OSG}.

3.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

3.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

3.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

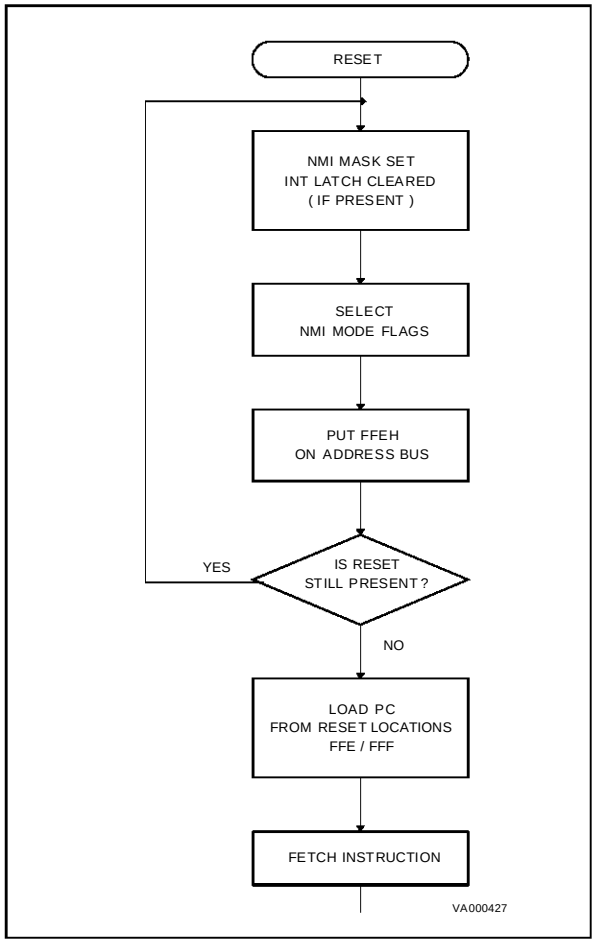
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 15. Reset and Interrupt Processing



RESETS (Cont'd)

3.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

3.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

3.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 16. Reset and Interrupt Processing

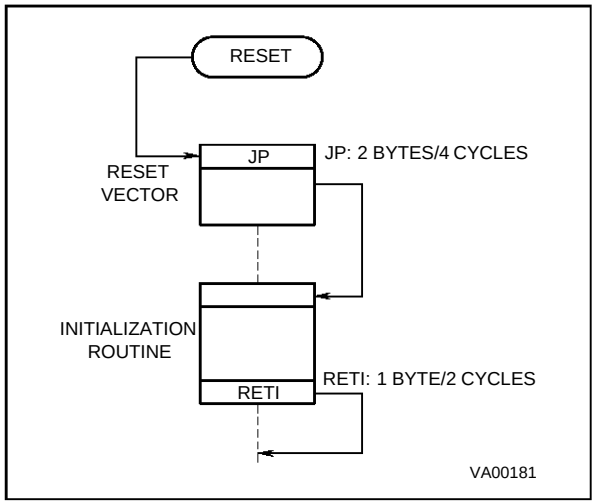
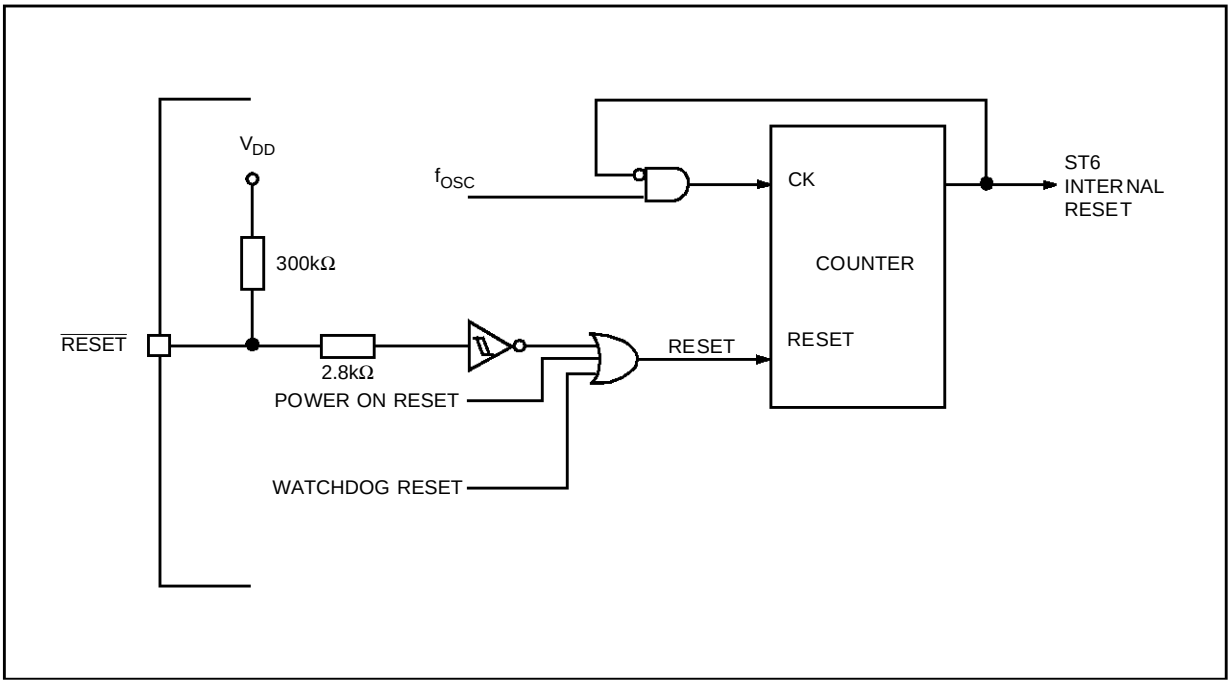


Figure 17. Reset Block Diagram



RESETS (Cont'd)

Table 2. Register Reset Status

Register	Address(es)	Status	Comment
Port Data Registers (PA, PB)	0C0h to 0C1h		
Port Direction Register (PA, PB)	0C4h to 0C5h		I/Os are Inputs with pull-up
Port Option Register (PA, PB)	0CCh to 0CDh	00h	I/Os are Inputs with pull-up
Interrupt Option Register	0C8h		Interrupts disabled
Timer Status/Control	0D4h		Timer disabled
X, Y, V, W Register	080h to 083h		
Accumulator	0FFh	Undefined	
Data RAM	084h to 0BFh		
Data ROM Window Register	0C9h		
Timer Counter Register	0D3h	FFh	
Timer Prescaler Register	0D2h	7Fh	Maximum count loaded
Watchdog Counter Register	0D8h	FEh	
Oscillator Control Register	0D1h	40h	main oscillator on

3.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two mask options, known as “WATCHDOG ACTIVATION” (i.e. HARDWARE or SOFTWARE) and “EXTERNAL STOP MODE CONTROL” (see Table 3).

In the SOFTWARE mask option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, save by resetting the MCU.

In the HARDWARE mask option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL mask option (available in ROM versions only) has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL mask option has been selected, port PB0 must be defined as an open-drain output, and PA2 as an input.

Table 3. Recommended Mask Option Choices

Functions Required	Recommended Mask Options
Stop Mode & Watchdog	“EXTERNAL STOP MODE” & “HARDWARE WATCHDOG”
Stop Mode	“SOFTWARE WATCHDOG”
Watchdog	“HARDWARE WATCHDOG”

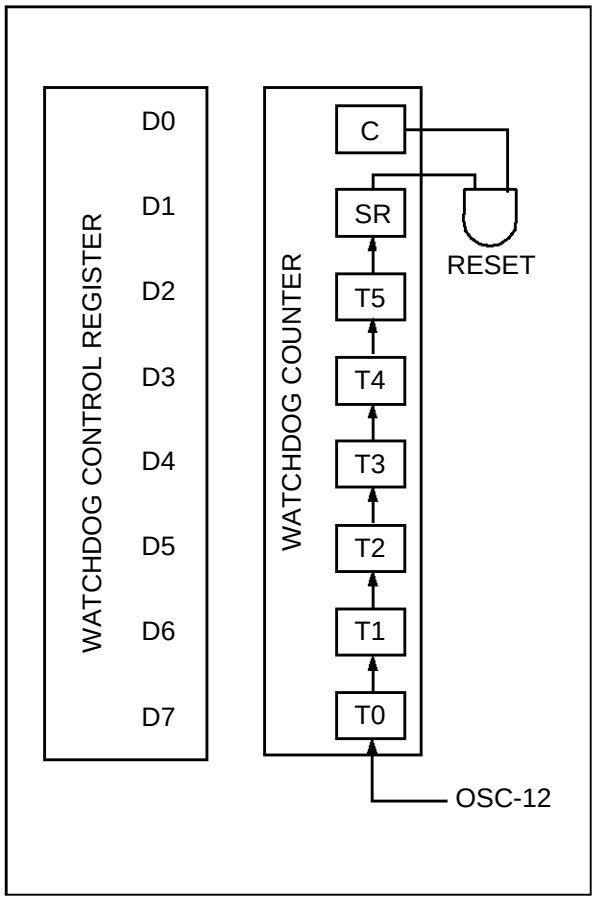
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 3.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 18.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 18. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 19) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 19. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

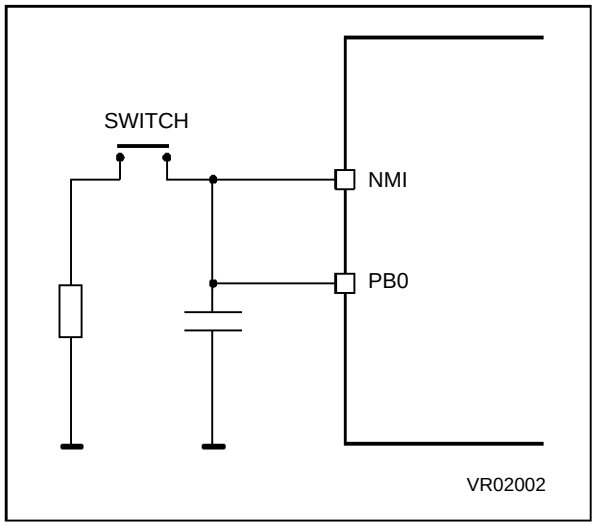
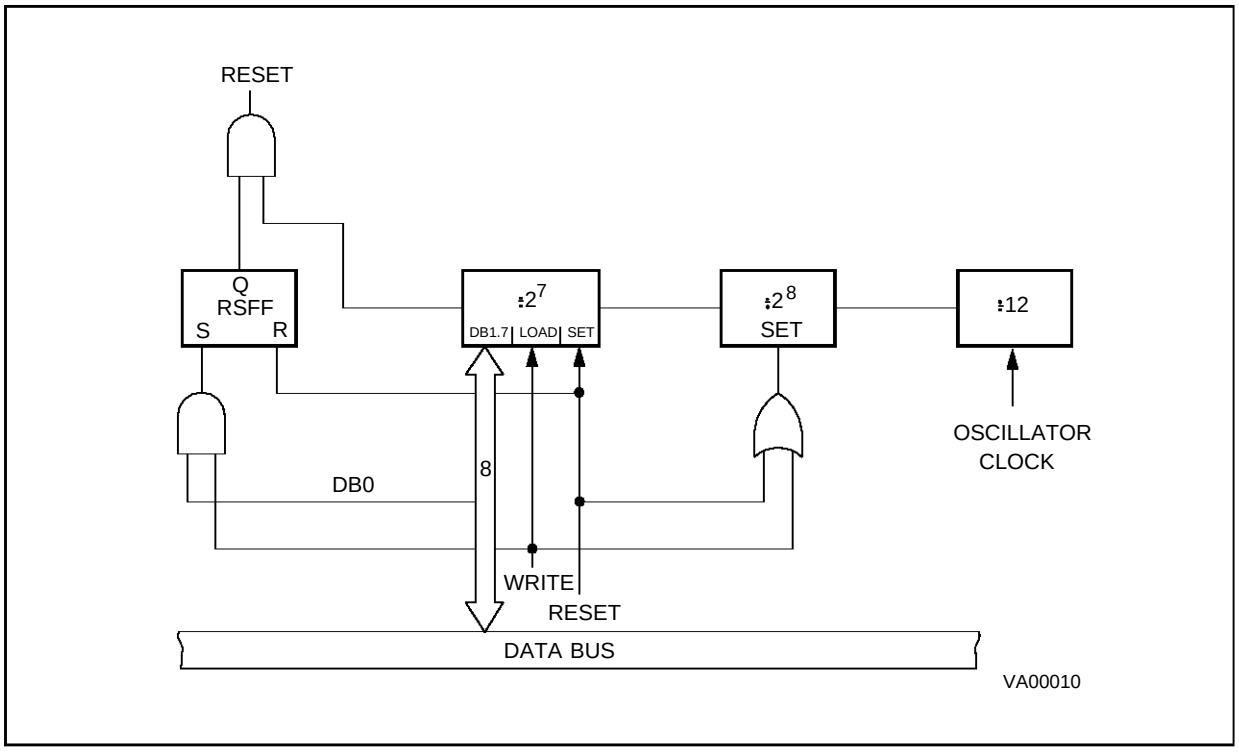


Figure 20. Digital Watchdog Block Diagram



3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 4).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 4. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A pins	Interrupt vector #1	(FF6h-FF7h)
Port B pins	Interrupt vector #2	(FF4h-FF5h)
TIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
Reserved	Interrupt vector #4	(FF0h-FF1h)

3.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named. It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port B pins is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named . It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the TIMER peripheral and Vector #4 is reserved.

On-chip peripheral has an associated interrupt request flag (TMZ for the Timer, which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (ETI for the Timer), which must be set to “1” to enable the associated interrupt request.

3.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #3 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

3.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port B) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 5. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A
	CLEARED	Falling edge mode on Port A
ESB	SET	Level sensitive mode on Port B
	CLEARED	Falling edge mode on Port B

3.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A-vector #1, Port B-vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

IINTERRUPTS (Cont'd)

3.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

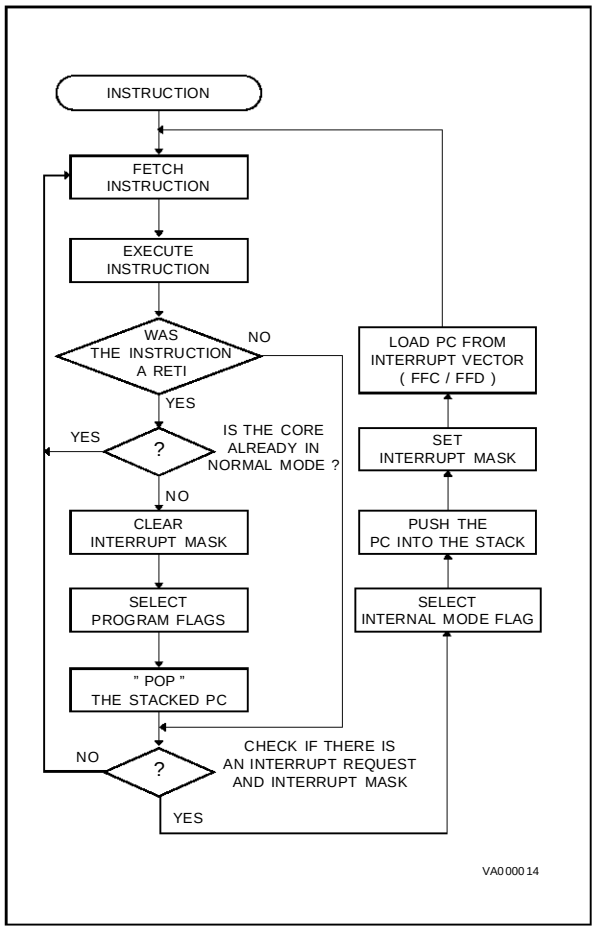
– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 21. Interrupt Processing Flow Chart

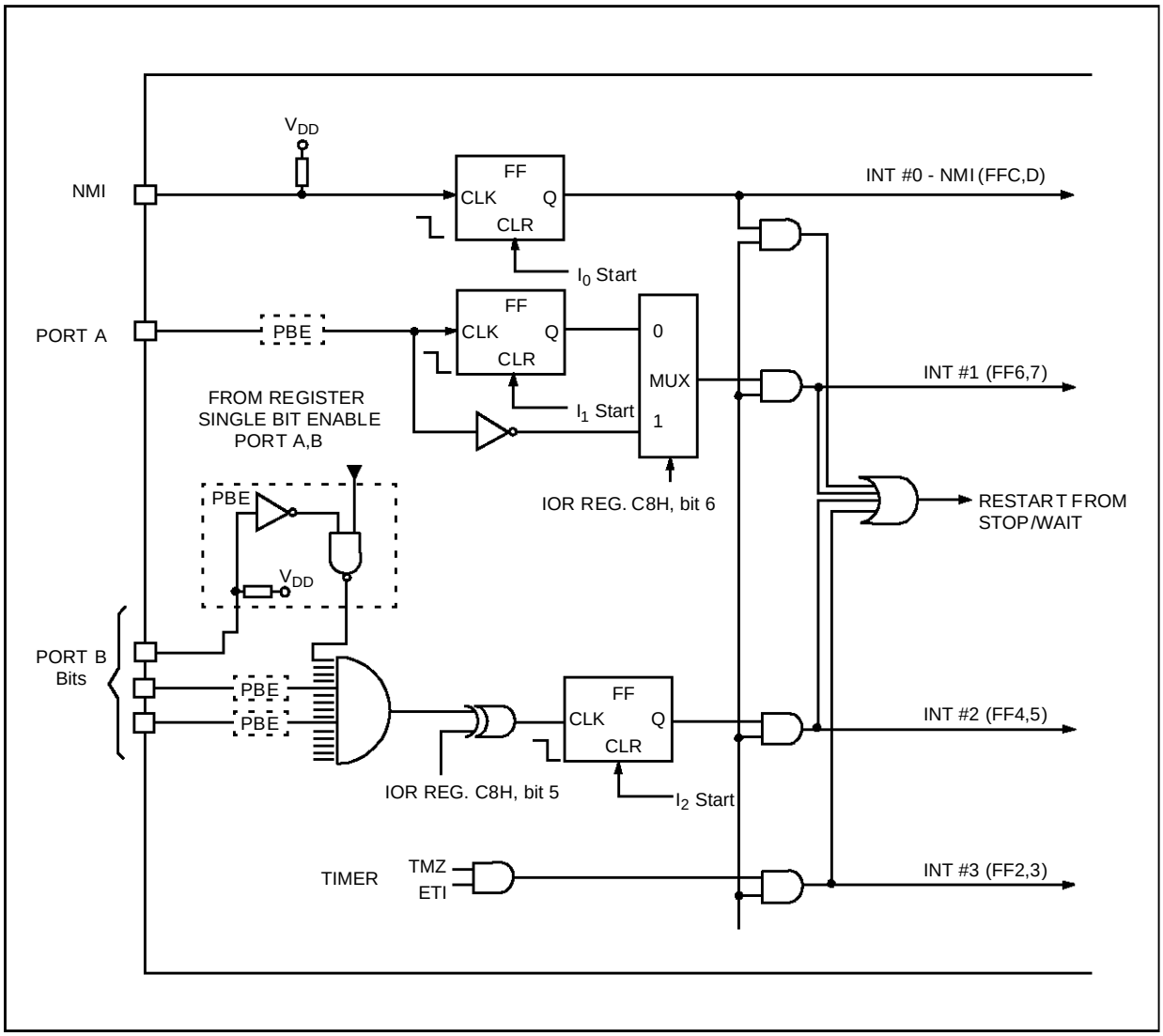


INTERRUPTS (Cont'd)

Table 6. Interrupt Requests and Mask Bits

Peripheral	Register	Address Register	Mask bit	Masked Interrupt Source	Interrupt vector
GENERAL	IOR	C8h	GEN	All Interrupts, excluding NMI	
TIMER	TSCR	D4h	ETI	TMZ: TIMER Overflow	Vector 3
Port PAn	ORPA-DRPA	C4h-CCh	ORPAn-DRPAn	PAn pin	Vector 1
Port PBn	ORPB-DRPB	C5h-CDh	ORPBn-DRPBn	PBn pin	Vector 2

Figure 22. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

In addition, the Low Frequency Auxiliary Oscillator (LFAO) can be used instead of the main oscillator to reduce power consumption in RUN and WAIT modes.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator (main oscillator or LFAO) is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the power consumption has to be further reduced, the Low Frequency Auxiliary Oscillator (LFAO) can be used in place of the main oscillator, if its operating frequency is lower. If required, the LFAO must be switched on before entering the WAIT mode.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

3.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection, consequently, when the LFAO is used, the user program must manage oscillator selection as soon as normal RUN mode is resumed.

3.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

3.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

3.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

– selecting the Low Frequency Auxiliary Oscillator (provided this runs at a lower frequency than the main oscillator).

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

The MCU features 9 Input/Output lines which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PA0-PA3 only)

The lines are organized as two Ports (A and B).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA and DRB), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

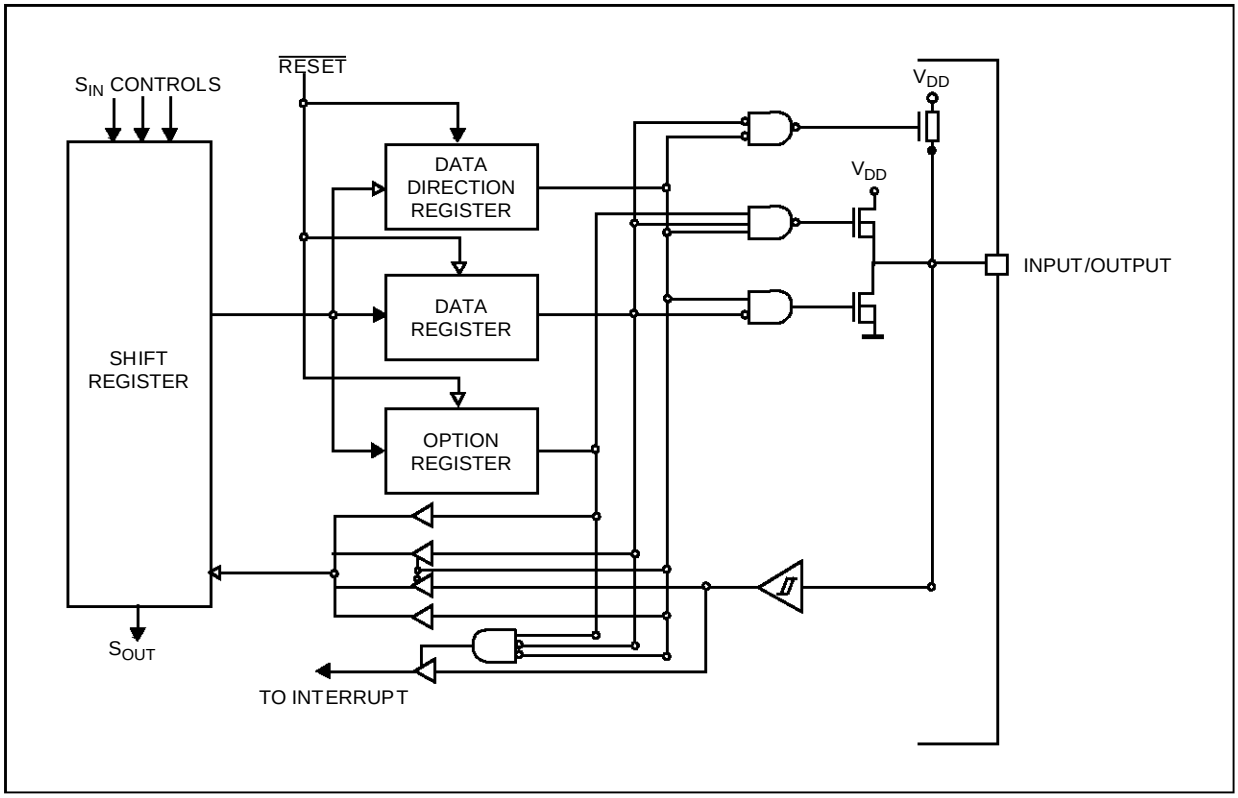
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA and DDRB) allow the data direction (input or output) of each pin to be set.

The two Option registers (ORA and ORB) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 23. I/O Port Block Diagram



I/O PORTS (Cont'd)

4.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 7 illustrates the various port configurations which can be selected by user software.

4.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

4.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A are AND-connected to the interrupt associated with Vector #1. The pins of Port B are AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

4.1.2 I/O Port Option Registers

ORA/B (CCh PA, CDh PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Option Register bits.*

4.1.3 I/O Port Data Direction Registers

DDRA/B (C4h PA, C5h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Direction Registers bits.*

4.1.4 I/O Port Data Registers

DRA/B (C0h PA, C1h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Registers bits.*

Table 7. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Reserved	
1	0	X	Output	20mA sink open-drain output (PA1-PA3 pins)
1	0	X	Output	Standard open-drain output (PB0, PB1, PB3, PB5-PB7 pins)
1	1	X	Output	20mA sink push-pull output (PA1-PA3 pins)

Note: X = Don't care

I/O PORTS (Cont'd)

4.1.5 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 24. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A and B Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

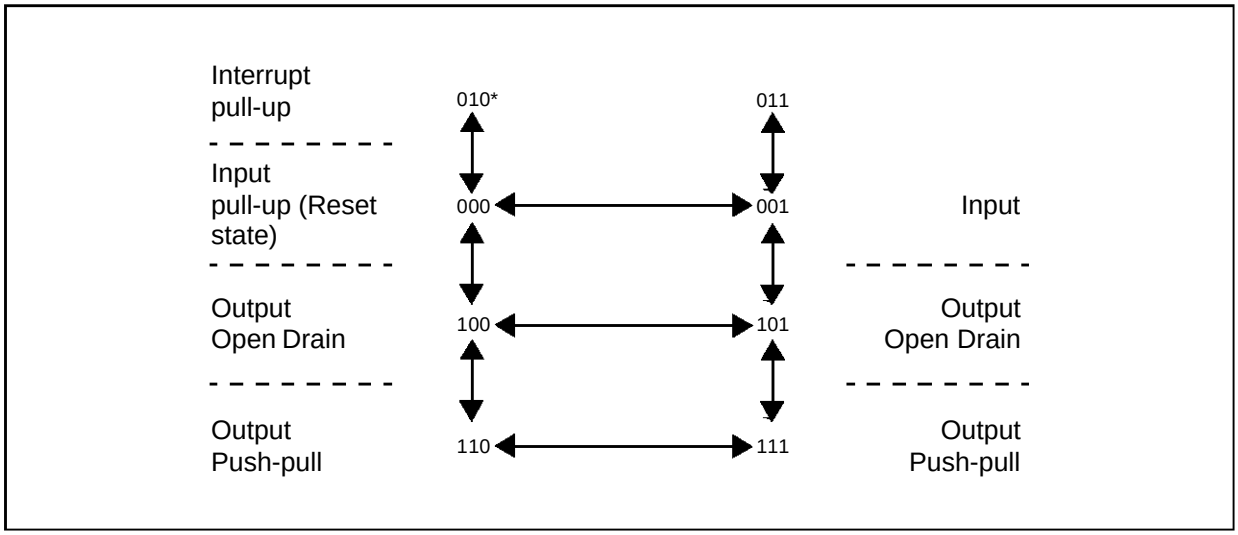
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

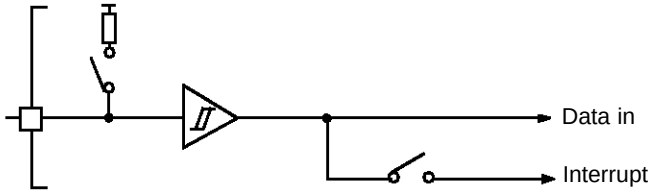
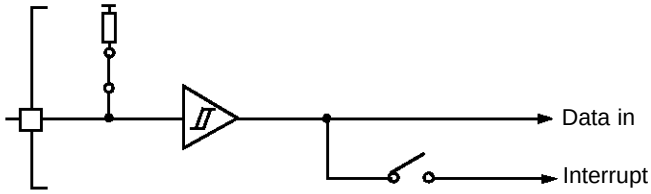
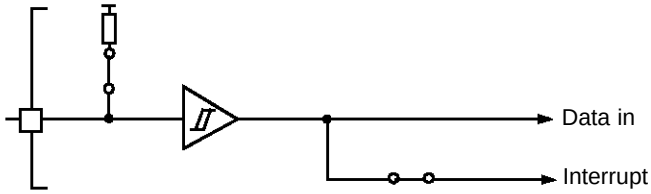
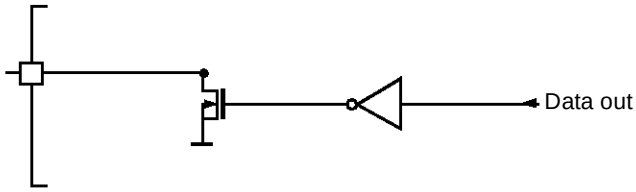
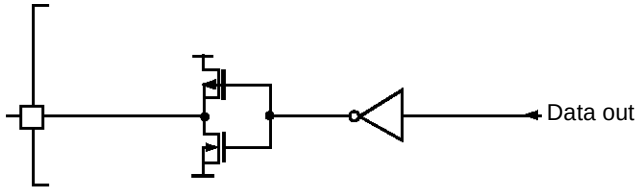
Figure 24. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 8. I/O Port Option Selections

MODE	AVAILABLE ON(1)	SCHEMATIC
Input	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
Input with pull up	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
Input with pull up with interrupt	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
Open drain output 5mA Open drain output 20mA	PB0, PB1, PB3, PB5-PB7 PA1-PA3	
Push-pull output 5mA Push-pull output 20mA	PB0, PB1, PB3, PB5-PB7 PA1-PA3	

Note 1. Provided the correct configuration has been selected.

4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}.

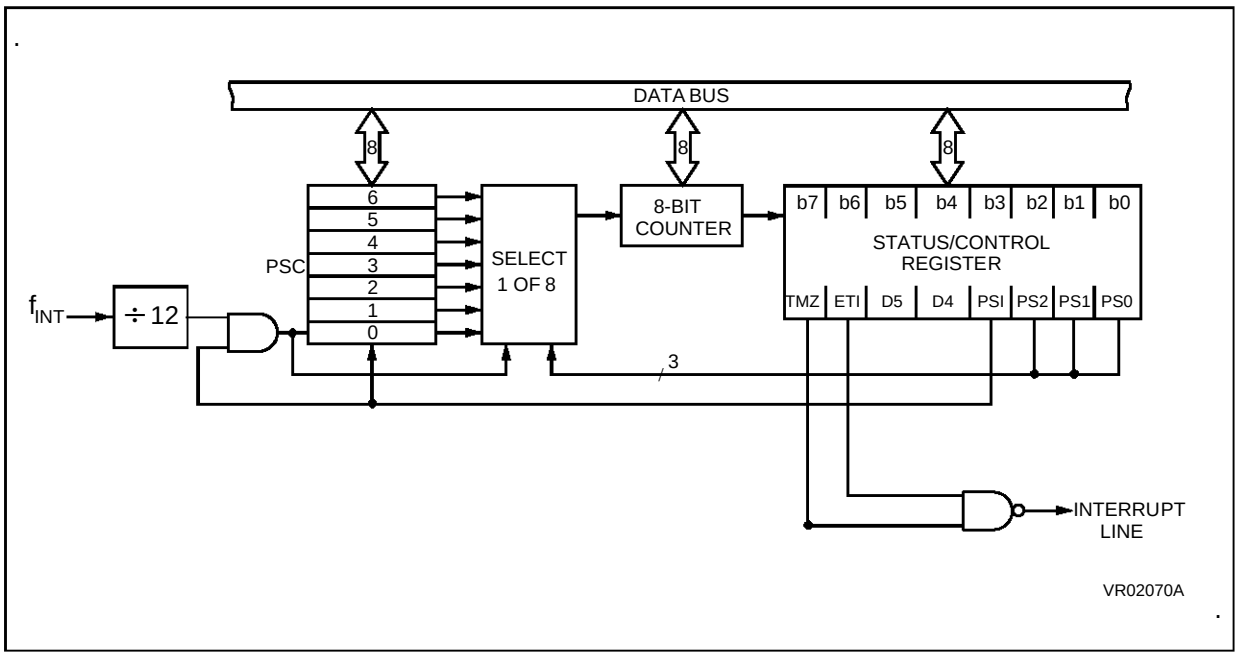
Figure 25 shows the Timer Block Diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero)bit in the TSCR is set. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set, an interrupt request is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input is the internal frequency (f_{INT}) divided by 12. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 9), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to allow the prescaler (and hence the counter) to start. If it is cleared, all the prescaler bits are set and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set. The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 26 illustrates the Timer's working principle.

Figure 25. Timer Block Diagram



TIMER (Cont'd)

4.2.1 Timer Operation

The Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high.

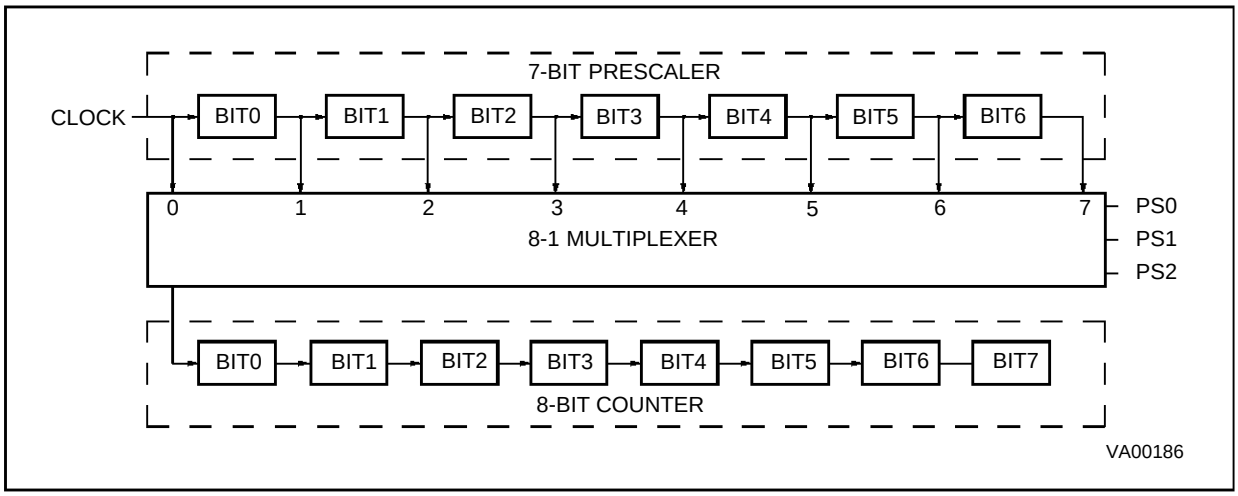
4.2.2 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

4.2.3 Application Notes

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

Figure 26. Timer Working Principle



TIMER (Cont'd)

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

4.2.4 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	D5	D4	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrup*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **D5**: *Reserved*

Must be reset.

Bit 4 = **D4**

When set, the timer is enabled; when reset the timer is disabled.

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 9. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register (TCR)

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 10. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:

X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 11. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 12. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected. The tested bit is shifted into carry.
*. Not Affected

Table 13. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not<M> Affected

Table 14. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ. Affected
*. Not Affected

Table 15. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
*. Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

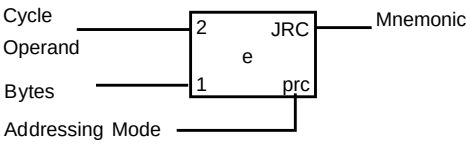
LOW / HI	**0 0000**	**1 0001**	**2 0010**	**3 0011**	**4 0100**	**5 0101**	**6 0110**	**7 0111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

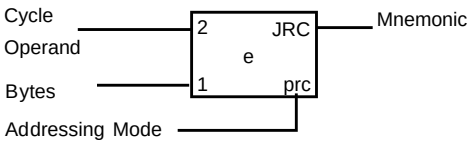
LOW / HI	**8 1000**	**9 1001**	**A 1010**	**B 1011**	**C 1100**	**D 1101**	**E 1110**	**F 1111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

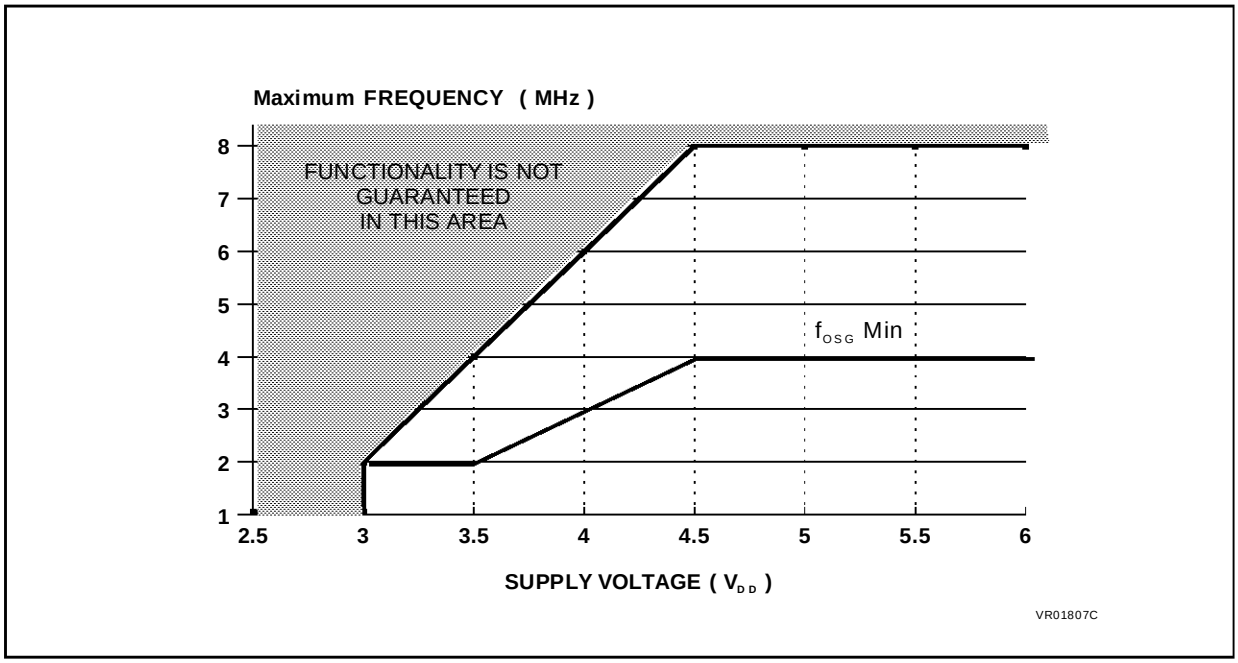
THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	°C/W
		PSO20			80	

6.2 RECOMMENDED OPERATING CONDITIONS

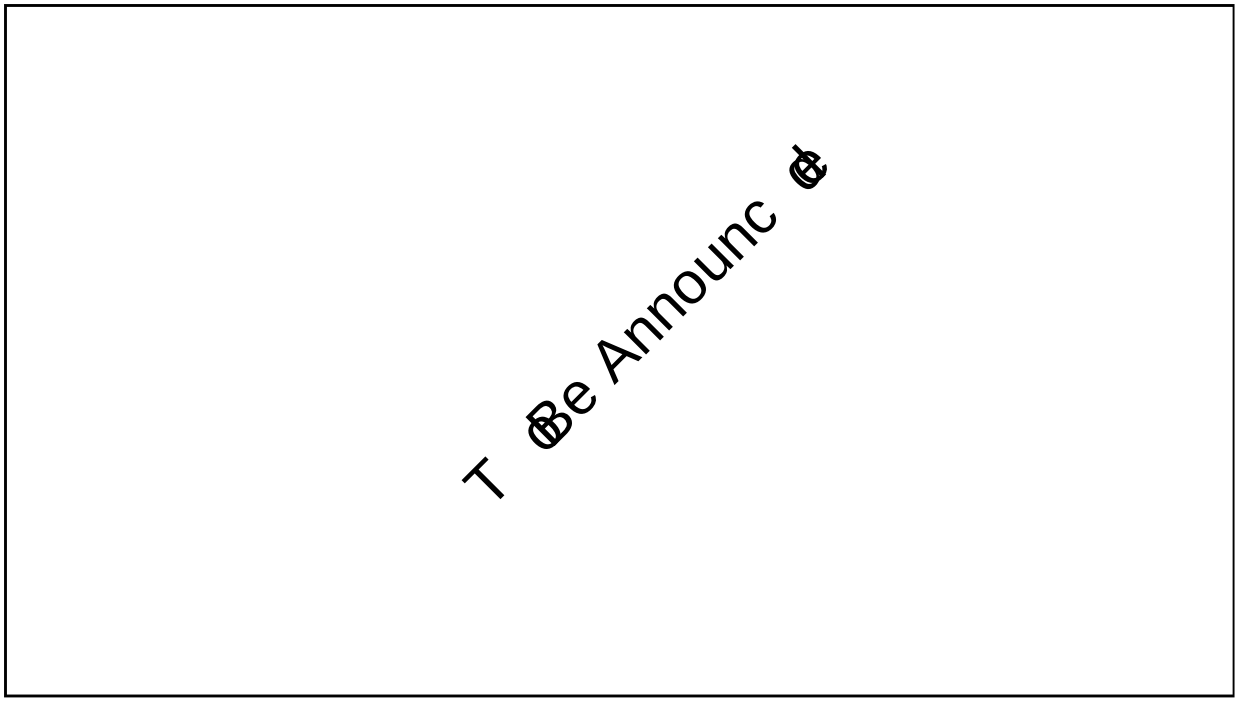
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input	V_{DD} = 4.5 to 5.5V			-5	mA

Figure 27. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



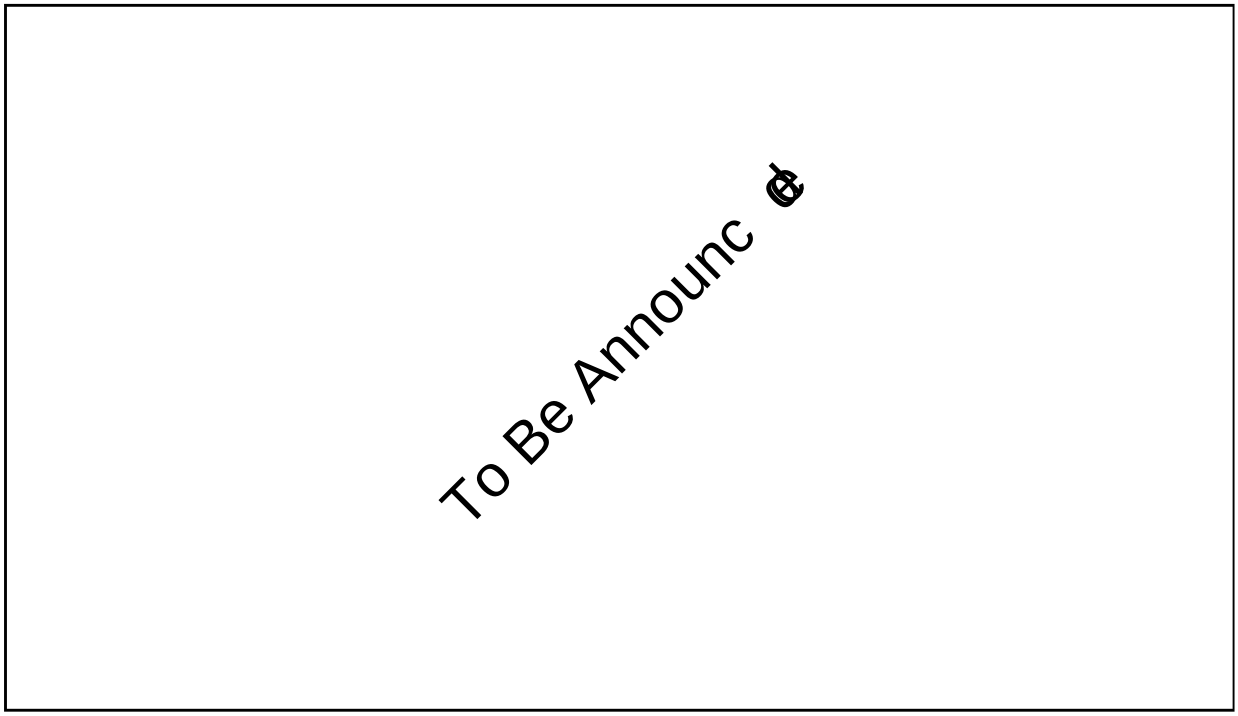
The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 28. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 29. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

6.3 READOUT PROTECTION FUSE

If the ROM READOUT PROTECTION option is selected, the waveform illustrated below must be applied to the TEST pin in order to blow the fuse.

The following circuit can be used for this purpose:

Figure 30. Programming wave form

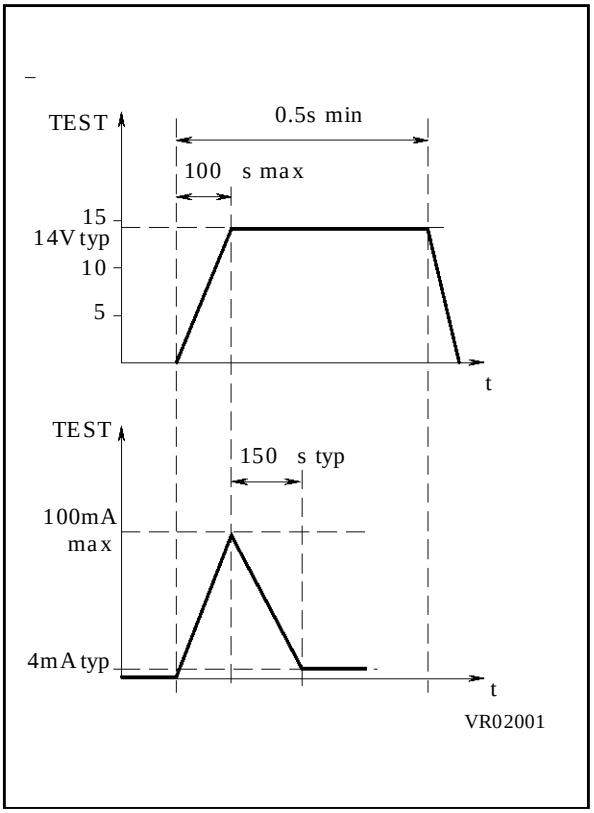
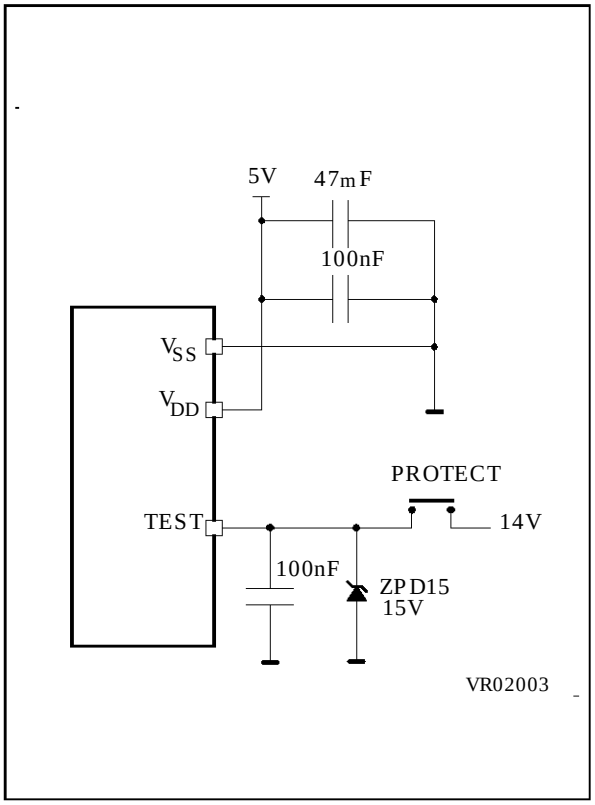


Figure 31. Programming Circuit

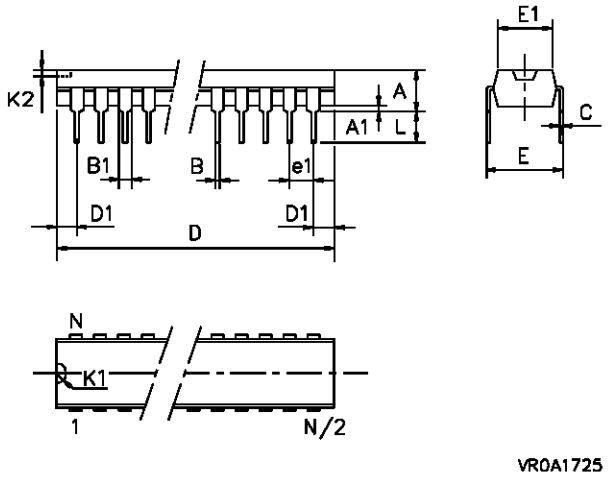


Note: ZPD15 is used for overvoltage protection

7 GENERAL INFORMATION

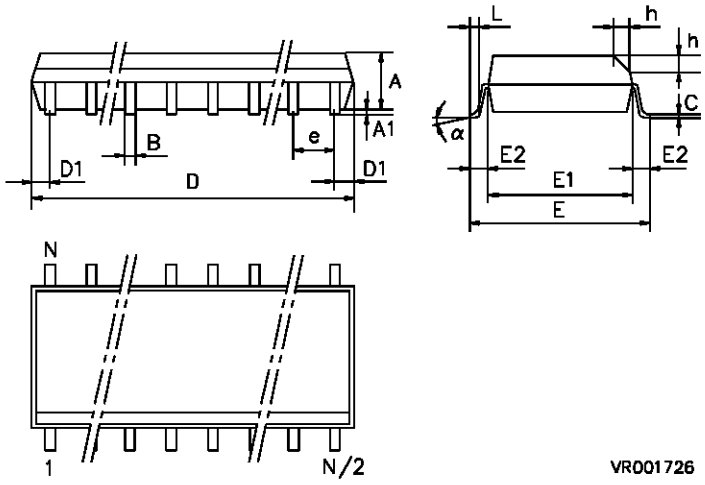
7.1 PACKAGE MECHANICAL DATA

Figure 32. 16-Pin Plastic Dual In Line Package (B), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			.200
A1			.508	.020		
B	.381	.508	.533	.015	.020	.021
B1	.762		1.651	.030		.065
C	.203	.254	.304	.008	.010	.012
D	18.92	19.18	19.56	.745	.755	.770
D1			1.27			.050
E	7.37	7.62	7.874	.290	.300	.310
E1		5.334			.210	
K1						
K2						
L	2.997	3.302	3.708	.118	.130	.146
e	2.286	2.54	2.794	.090	.100	.110
	Number of Pins					
N	16					

Figure 33. 16-Pin Plastic Small Outline Package (M), 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		2.286			.090	
A1	.102		.305	0.004		.012
B	.381		.483	.015		.019
C	.229		.254	.009		.010
D	10.26		10.34	.404		.407
D1	-	-	-	-	-	-
E	10.24		10.34	.403		.407
E1	7.44		7.54	.293		.297
E2	-	-	-	-	-	-
L		.832			.033	
e		1.27			.050	
h		.508			.020	
alpha		5^o			5^o	
	Number of Pins					
N	16					

ST6203B MICROCONTROLLER OPTION LIST

Customer
Address

Contact
Phone No
Reference

SGS-THOMSON Microelectronics references

Device:	[ ] ST6203B
Package:	[ ] Dual in Line Plastic[ ] Small Outline Plastic
	In this case, select conditioning
	[ ] Standard (Stick)
	[ ] Tape & Reel
Temperature Range:	[ ] 0°C to + 70°C[ ] - 40°C to + 85°C
Special Marking:	[ ] No
	[ ] Yes "_ _ _ _ _ _ _ _ _ _ "

Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count: DIP20: 9
SO20: 6
Oscillator Source Selection:[ ] Crystal Quartz/Ceramic resonator (Default)
[ ] RC Network
Watchdog Selection: [ ] Software Activation (STOP mode available)
[ ] Hardware Activation (no STOP mode)
OSG: [ ] Enabled
[ ] Disabled (Default)
Input pull-up selection on NMI pin:[ ] Yes[ ] No
ROM Readout Protection: [ ] Standard (Fuse cannot be blown)
[ ] Enabled (Fuse can be blown by the customer)
Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.
External STOP Mode Control[ ] Enabled[ ] Disabled (Default)
Comments :
Supply Operating Range in the application:
Oscillator Fequency in the application:
Notes

Signature
Date

7.2 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

7.2.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

7.2.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 16. ROM Memory Map for ST6203B

Device Address	Description
0000h-0B9Fh	Reserved
0BA0h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Table 17. Ordering Information

Sales Type	ROM	I/O	Temperature Range	Package
ST6203BB1/XXX ST6203BB6/XXX	1036 Bytes	9	0 to +70°C -40 to + 85°C	PDIP16
ST6203BM1/XXX ST6203BM6/XXX			0 to +70°C -40 to + 85°C	PSO16

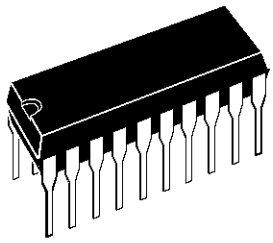
SGS-THOMSON MICROELECTRONICS

ST62T09

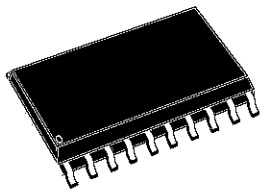
8-BIT OTP MCUs WITH A/D CONVERTER

PRELIMINARY DATA

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 4 Interrupt Vectors
- Look-up Table capability in OTP
- Data OTP: User selectable size (in program OTP)
- Data RAM: 64 bytes
- 12 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 4 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- Digital Watchdog
- 8-bit A/D Converter with 4 analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal or Ceramic resonator
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	OTP (Bytes)	I/O Pins
ST62T09	1036	12

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T09 device is a low cost member 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.

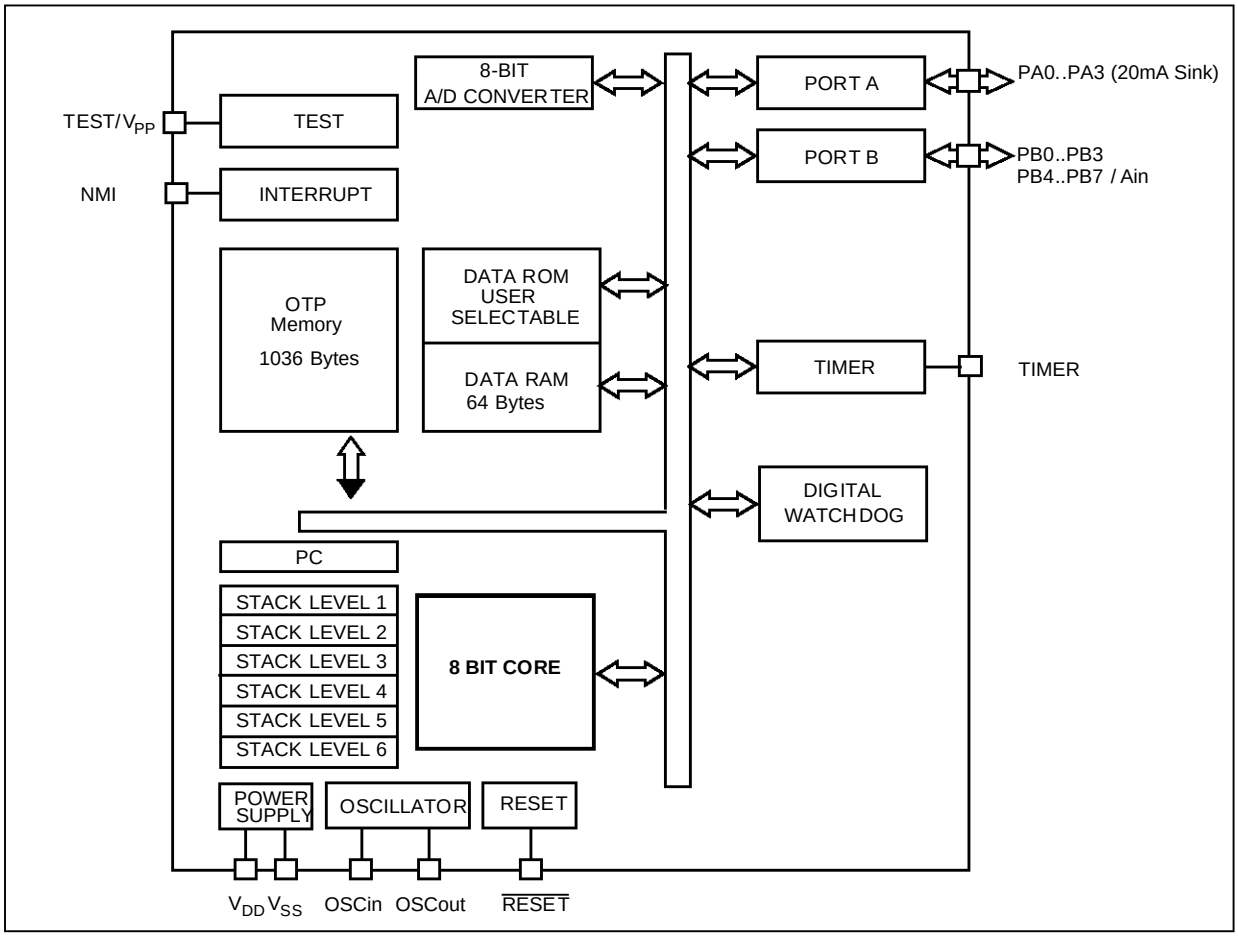
OTP devices are functionally identical to their ROM counterparts; in addition, the ROM based versions offer the following additional features: RC Oscillator, Oscillator Safeguard, external Stop mode control, program code readout protection and an optional internal pullup on the NMI and TIMER pins.

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 ST62E20 (4K EPROM) device may be used to emulate the ST62T09 OTP device, but care should be taken not to exceed the memory size of the T09 device.

These compact low-cost devices feature a Timer comprising an 8-bit counter and a 7-bit programmable prescaler, an 8-bit A/D Converter with 4 analog inputs and a Digital Watchdog timer, making them well suited for a wide range of automotive, appliance and industrial applications.

Figure 1. Block Diagram



1.2 PIN DESCRIPTIONS

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation. If TEST pin is connected to a +12.5V level during the reset phase, the EPROM/OTP programming Mode is entered.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. A pull-up device must be provided externally on OTP and EPROM devices.

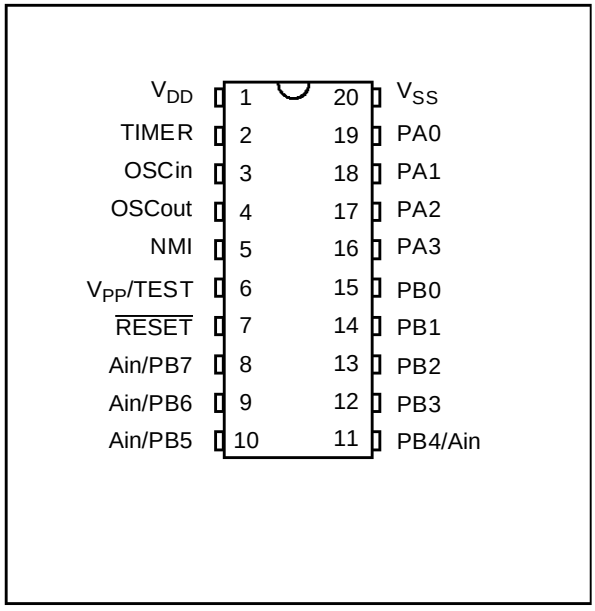
TIMER. This is the timer I/O pin. In input mode it is connected to the prescaler and acts as external timer clock input or as control gate input for the internal timer clock. In output mode the timer pin outputs the data bit when a time-out occurs. A pull-up device must be provided externally on OTP and EPROM devices.

PA0-PA3. These 4 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs.

PA0-PA3 can also sink 20mA for direct LED driving.

PB0-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs and as analog inputs for the A/D converter.

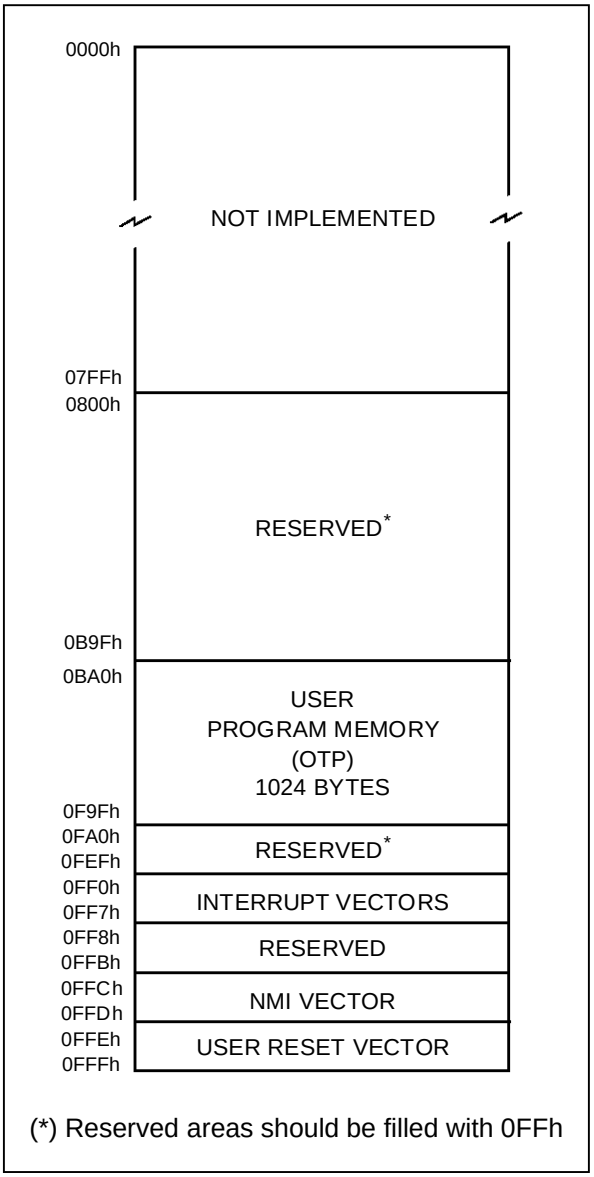
Figure 1. ST62T09 Pin Configuration



1.3 MEMORY MAP

1.3.1 Program Memory Map

Figure 2. ST62T09 Program Memory Map



(*) Reserved areas should be filled with 0FFh

1.3.2 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.

The Data Space is fully described and illustrated on page 18.

1.4 PARTICULARITIES OF OTP DEVICES

OTP and EPROM devices are identical save for the package which, in the EPROM device, is fitted with a transparent window to allow erasure of memory contents by exposure to UV light.

Both OTP and EPROM parts may be programmed using programming equipment approved by SGS-THOMSON.

1.4.1 OTP Programming

Programming mode is selected by applying a 12.5V voltage to the V_{PP}/TEST pin during reset. Programming of OTP and EPROM parts is fully described in the EPROM Programming Board User Manual.

1.4.2 Eprom Erasure

Thanks to the transparent window present in the EPROM package, its memory contents may be erased by exposure to UV light.

Erasure begins when the device is exposed to light with a wavelength shorter than 4000Å. It should be noted that sunlight, as well as some types of artificial light, includes wavelengths in the 3000-4000Å range which, on prolonged exposure, can cause erasure of memory contents. It is thus recommended that EPROM devices be fitted with an opaque label over the window area in order to prevent unintentional erasure.

The recommended erasure procedure for EPROM devices consists of exposure to short wave UV light having a wavelength of 2537Å. The minimum recommended integrated dose (intensity x exposure time) for complete erasure is 15Wsec/cm^2. This is equivalent to an erasure time of 15-20 minutes using a UV source having an intensity of 12mW/cm^2 at a distance of 25mm (1 inch) from the device window.

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

The CPU Core may be thought of as an independent central processor communicating with on-chip I/O, memory and peripherals. For further details refer to page 14.

2.2 CPU REGISTERS

The CPU Core features six registers and three pairs of flags available to the programmer. For a detailed description refer to page 20.

3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The Oscillator may be driven by an external clock, or by a crystal or ceramic resonator. ROM devices also offer RC oscillator and Oscillator Safeguard features. For a complete description refer to page 22.

3.2 RESETS

The MCU can be reset in three ways: by the external Reset input being pulled low, by the Power-on Reset circuit, or by the Digital Watchdog timing out. For further details refer to page 26.

3.3 DIGITAL WATCHDOG

The Digital Watchdog can be used to provide controlled recovery from software upsets. Software and Hardware enabled Watchdog options are available in order to achieve optimum trade-off between power consumption and noise immunity. For a complete description and a selection guide refer to page 29.

3.4 INTERRUPTS

The CPU can manage four Maskable and one Non-Maskable Interrupt source. Each source is associated with a specific Interrupt Vector. An internal pullup option on the NMI pin is available on ROM devices. For a complete description refer to page 33.

3.5 POWER SAVING MODES

WAIT mode reduces electrical consumption during idle periods, while STOP mode achieves the lowest power consumption by stopping all CPU activity. For a complete description refer to page 37.

3.6 I/O PORTS

Input/Output lines may be individually programmed as one of a number of different configurations. For further details refer to page 39.

3.7 TIMER

The on-chip Timer peripheral consists of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. For a complete description refer to page 43.

3.8 A/D CONVERTER (ADC)

The 8-bit on-chip ADC features multiplexed analog inputs, as alternate I/O functions. Conversion is by successive approximations, with a typical conversion time of 70us, at 8MHz oscillator frequency. For a complete description refer to page 46.

4 SOFTWARE

4.1 ST6 ARCHITECTURE

The ST6 architecture has been designed to exploit the hardware in the most efficient way possible, while keeping byte usage to a minimum. For further details refer to page 48.

4.2 ADDRESSING MODES

The ST6 core offers nine addressing modes: Immediate, Direct, Short Direct, Extended, Program Counter Relative, Bit Direct, Bit Test & Branch, Indirect, and Inherent. For a complete description of the available addressing modes, refer to page 48.

4.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes; these may be subdivided into six types: load/store, arithmetic/logic, conditional branch, control, jump/call, and bit manipulation. For further details refer to page 49.

5 ELECTRICAL CHARACTERISTICS

5.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices designed to protect the inputs against damage due to high static voltages; however, it is advisable to take normal precautions to avoid applying voltages higher than the specified maximum ratings.

For proper operation, it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, T_j, in degrees Celsius can be obtained from:

$$T_j = T_A + P_D \times R_{thJA}$$

Where:

T_A = Ambient Temperature.

R_{thJA} =Package thermal resistance (junction-to ambient).

$P_D = P_{int} + P_{port}$

$P_{int} = I_{DD} \times V_{DD}$ (chip internal power).

P_{port} =Port power dissipation (to be determined by the user)

.

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_{PP}	OTP/EPROM Programming Voltage	13	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into V_{DD} (source)	50(2)	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

Stresses above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

(1) Within these limits, clamping diodes are non-conducting. Voltages outside these limits are authorised provided injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

5.2 THERMAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{thJA}	Thermal Resistance (junction to ambient)	PDIP20			60	°C/W
		PSO20			80	

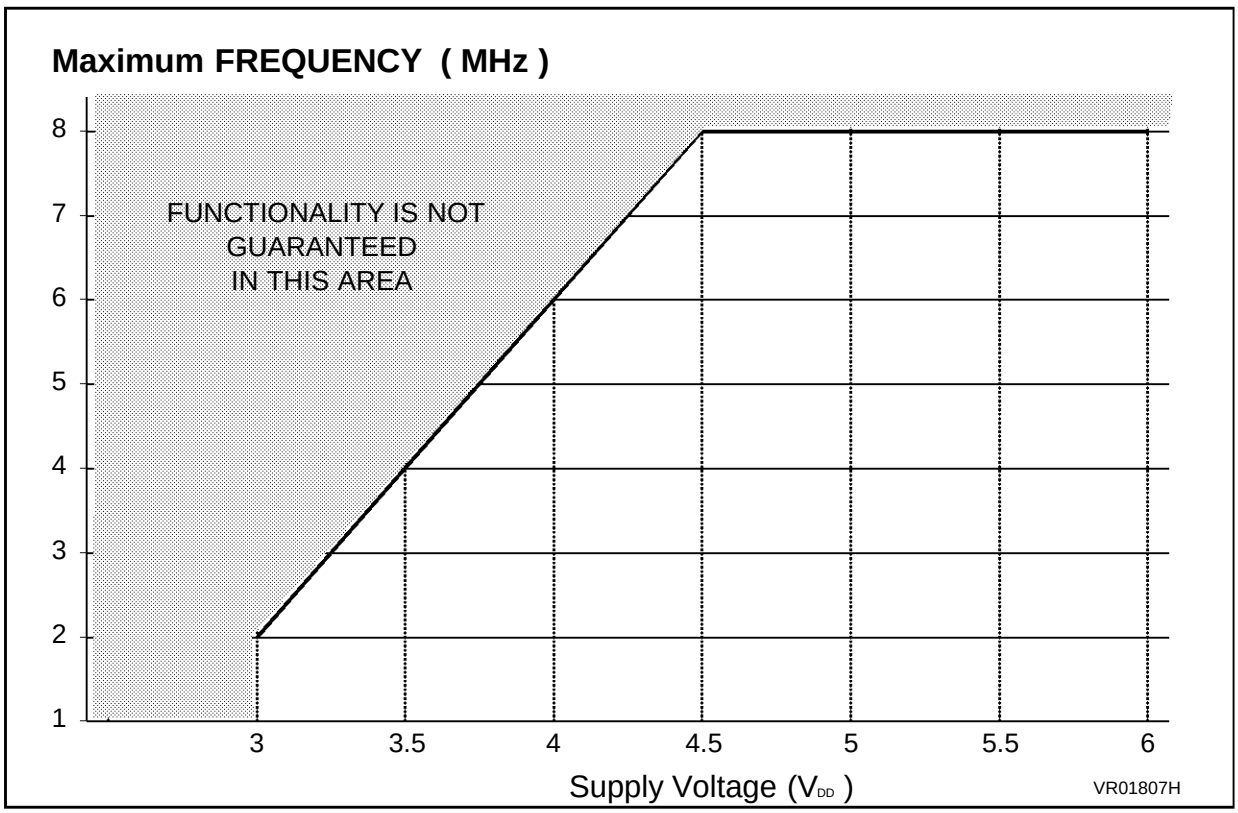
5.3 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

If a total current of +1mA is flowing into a single analog channel, or if the total current flowing into all the analog inputs is 1mA, all resulting A/D conversions will be shifted by + 1 LSB. If a total positive current is flowing into a single analog channel, or if the total current flowing into all analog inputs is 5mA, all the resulting conversions are shifted by + 2 LSB.

Figure 2. Maximum Operating FREQUENCY (F_{MAX}) Versus SUPPLY VOLTAGE (V_{DD})

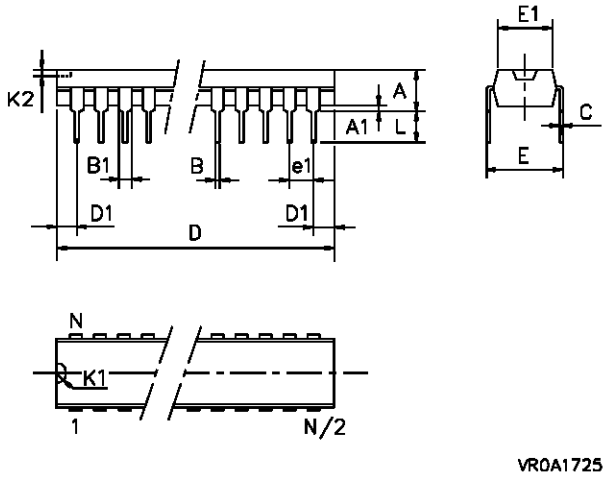


The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

6 GENERAL INFORMATION

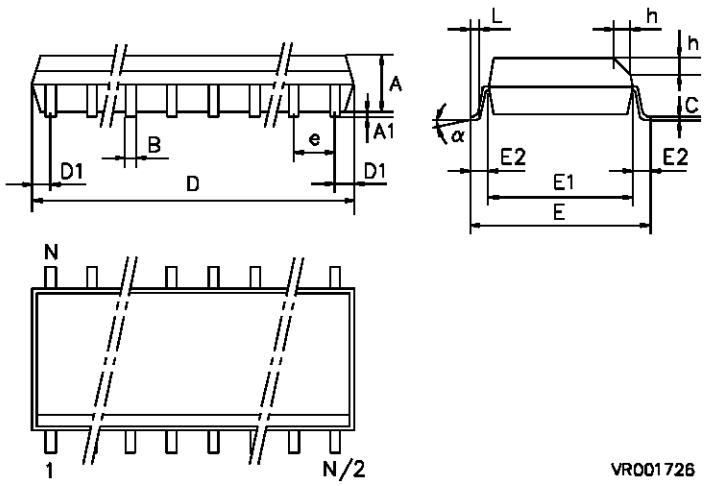
6.1 PACKAGE MECHANICAL DATA

Figure 3. 20-Pin Plastic Dual In-Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
A2	-	-	-	-	-	-
B		0.45			0.018	
B1	1.39		1.65	0.055		0.065
C		0.25			0.010	
D			25.4			1.000
e3		22.86			0.900	
E1			7.1			0.280
e		2.54			0.100	
eB		8.5			0.335	
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.3			0.130	
	Number of Pins					
N	20					

Figure 4. 20-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

Table 1. OTP Device Sales Types

Sales Type	I/O Pins	Option	Temperature range	Package
ST62T09B6/HWD	12	Hardware Watchdog	-40°C TO +85°C	PDIP20
ST62T09B6/SWD	12	Software Watchdog		
ST62T09M6/HWD	12	Hardware Watchdog		PSO20
ST62T09M6/SWD	12	Software Watchdog		

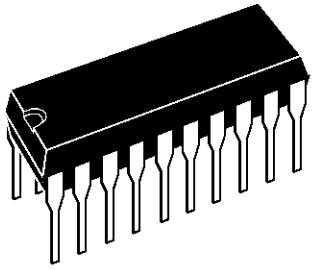
SGS-THOMSON MICROELECTRONICS

ST6209B

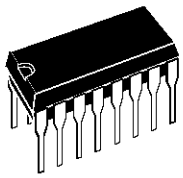
8-BIT HCMOS MCU WITH A/D CONVERTER

PRELIMINARY DATA

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in ROM
- Data ROM: User selectable size (in program ROM)
- Data RAM: 64 bytes
- ROM read-out Protection
- 12 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 4 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- Digital Watchdog
- Oscillator Safe Guard
- 8-bit A/D Converter with 4 analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal, Ceramic resonator or RC network
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	I/O Pins
ST6209B	1036	12

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

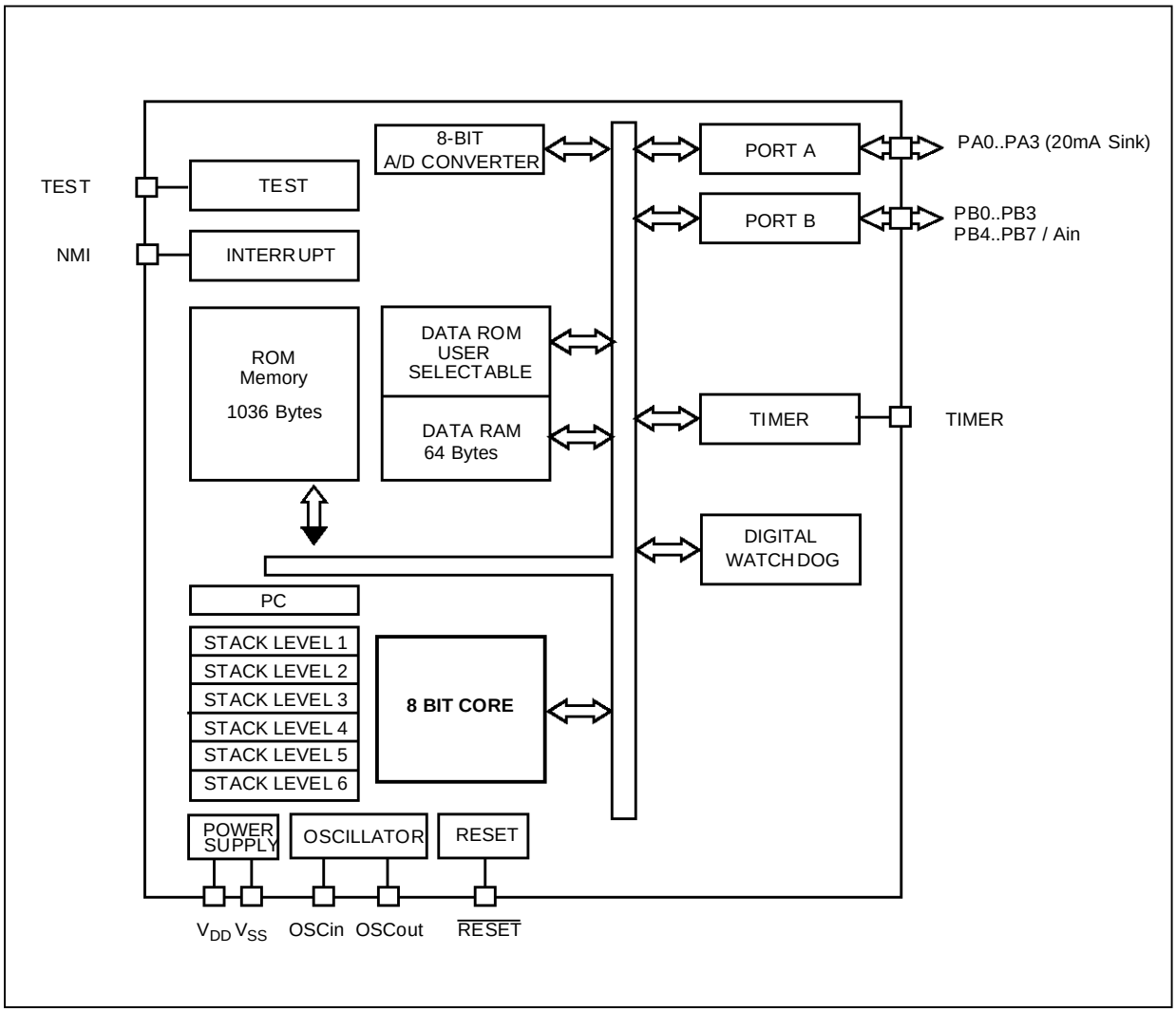
The ST6209B microcontroller is a member of the 8-bit HCMOS ST62xx family of devices, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST6209B features the following peripherals: a Timer comprising an 8-bit counter equipped with a 7-bit software programmable prescaler, an 8-bit A/D Converter with 4 analog inputs (A/D inputs are I/O pin alternate functions), and a Digital Watchdog timer.

The ST6209B features a choice of Quartz, Ceramic or RC oscillators, an Oscillator Safe Guard circuit, Read-out Protection against unauthorised copying of program code, and an External STOP Mode Control option to enlarge the range of power consumption versus reliability trade-offs.

These devices are well suited for automotive, appliance and industrial applications. The user programmable part for program development is the ST62E20, which is a pin compatible device with 4Kbytes of EPROM. Care must be taken to only use the memory available on the ST6209B when using the ST62E20 for evaluation or development.

Figure 5. Block Diagram



1.2 PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. When the QUARTZ/CERAMIC RESONATOR Mask Option is selected, a quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. When the RC OSCILLATOR Mask Option is selected, a resistor must be connected between the OSCout pin and ground. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST. The TEST pin must be held at V_{SS} for normal operation (an internal 100kΩ pull-down resistor selects normal operating mode if the TEST pin is not connected externally).

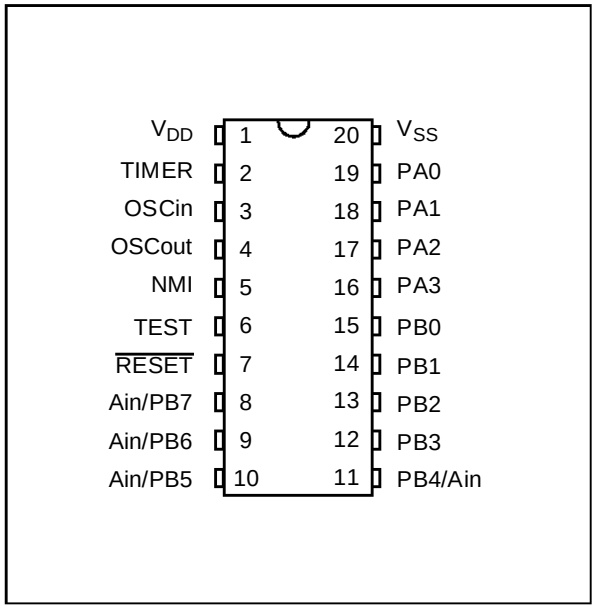
NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI is falling edge sensitive. A ROM mask option makes available an on-chip pull-up on the NMI pin.

TIMER. This is the timer I/O pin. In input mode it is connected to the prescaler and acts as external timer clock input or as control gate input for the internal timer clock. In output mode the timer pin outputs the data bit when a time-out occurs. On the ST6209B the user can select as a ROM mask option the presence of an on-chip pull-up on the TIMER pin.

PA0-PA3. These 4 lines are organized as an I/O port (A). Each line may be configured under software control as an input with or without internal pull-up resistors, as an interrupt generating input with pull-up resistors, or as an open-drain or push-pull output. PA0-PA3 can sink up to 20mA for direct LED drive capability. When the External STOP Mode Control option is enabled, PA2 must be defined as an input.

PB0-PB7. These 8 lines are organized as one I/O port (B). When the External STOP Mode Control option is disabled, each line may be configured under software control as an input with or without internal pull-up resistor, as an interrupt generating input with pull-up resistor, or as an open-drain or push-pull output. PB4-PB7 can also be used as analog inputs to the A/D converter. When the External STOP Mode Control option is enabled, PB0 can only be configured as open-drain in output mode (push-pull output is not available). The other lines are unchanged.

Figure 6. ST6209B Pin Configuration



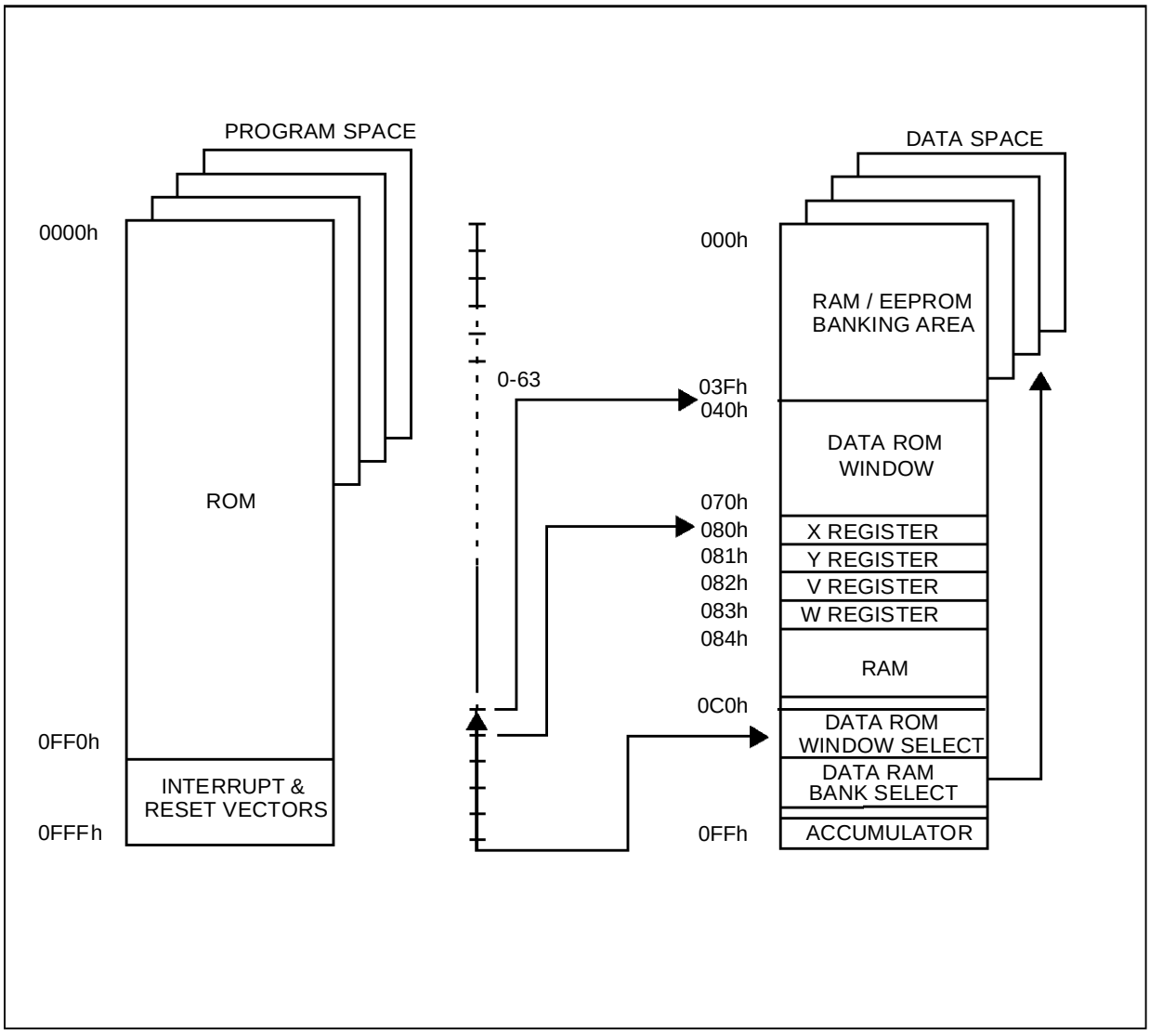
1.3 MEMORY MAP

1.3.1 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 ROM and user vectors; Data space contains user data in RAM and in ROM, and Stack space accommodates six levels of stack for subroutine and interrupt service routine nesting.

Figure 7. Memory Addressing Diagram



MEMORY MAP (Cont'd)

1.3.2 Program Space

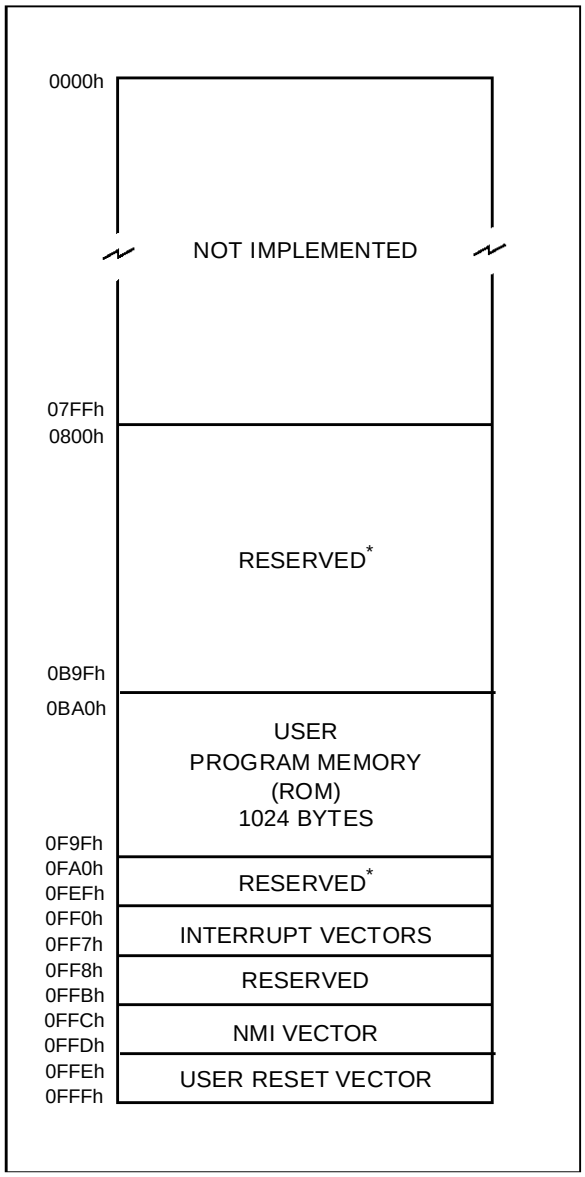
Program Space is physically implemented in ROM memory. It comprises the instructions to be executed, the data required for immediate addressing mode instructions, the reserved factory test area and the user vectors. Program Space is addressed via the 12-bit Program Counter register (PC register)

1.3.2.1 ROM Protection

The ST6209B Program Space can be protected against external readout of ROM contents when the READOUT PROTECTION mask option is chosen. This option allows the user to blow a dedicated fuse on the silicon, by applying a high voltage at V_{PP} (see detailed information in the "Electrical Specification").

Note: Once the Readout Protection fuse is blown, it is no longer possible, even for SGS-THOMSON, to gain access to the ROM contents. Returned parts with a blown fuse can therefore not be accepted.

Figure 8. ST6209B Program Memory Map



(*) Reserved areas should be filled with 0FFh

MEMORY MAP (Cont'd)

1.3.3 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 ROM.

1.3.3.1 Data ROM

All read-only data is physically stored in ROM memory, which also accommodates the Program Space. The ROM 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 ROM.

1.3.3.2 Data RAM

In ST6209B devices, the data space includes 60 bytes of RAM, 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 (DRW register).

1.3.4 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.

Table 1. ST6209B Data Memory Space

NOT IMPLEMENTED	000h 03Fh
DATA ROM WINDOW 64 BYTES	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
RESERVED	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
RESERVED	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
RESERVED	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PSC REGISTER	0D2h
TIMER DATA REGISTER	0D3h
TIMER TSCR REGISTER	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

1.3.5 Data Window Register (DWR)

The Data ROM window is located from address 0040h to address 007Fh in Data space. It allows direct reading of 64 consecutive bytes located anywhere in ROM memory, between address 0000h and 1FFFh (top memory address depends on the specific device). All the ROM memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the ROM memory by writing the appropriate code in the Write-only Data Window register (DWR register, location 00C9h).

The DWR register can be addressed like any RAM location in the Data Space at address 00C9h, it is however a write-only register and cannot be accessed using single-bit operations. This register is used to move the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as data in ROM memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 9). So when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in ROM is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data ROM window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

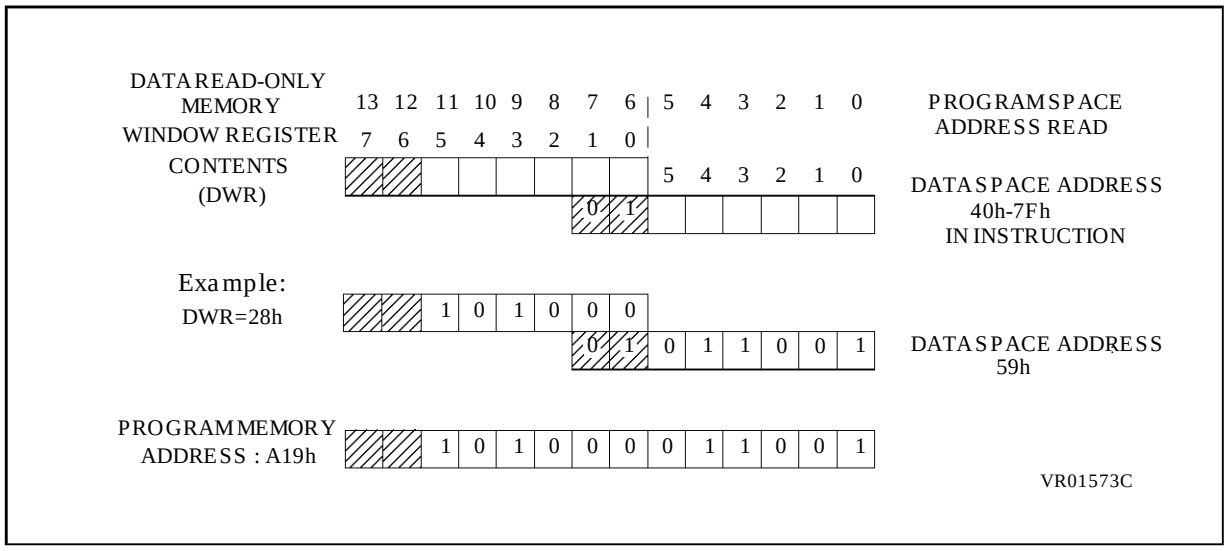
Bit 7 = This bit is not used.

Bit 6-0 = **DWR6-DWR0:** *Data ROM Window Register Bits.* These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

Caution: *This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.*

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

Figure 9. Data ROM Window Memory Addressing



2 CENTRAL PROCESSING UNIT

2.1 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. In-core communication is arranged as shown in Figure 10; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

2.2 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.

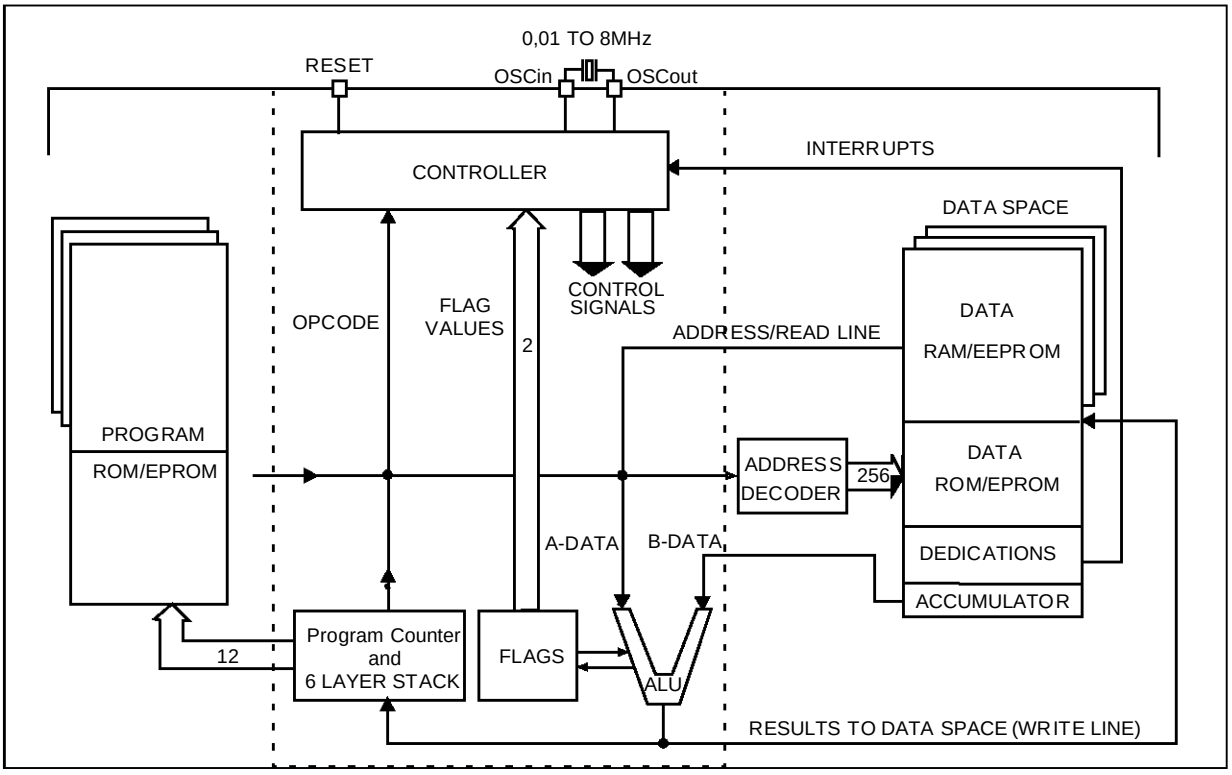
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 10. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

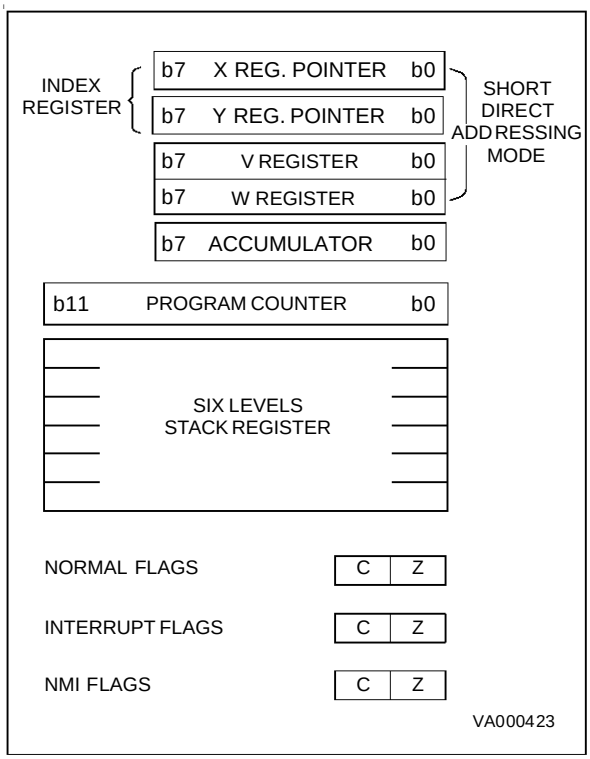
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 11. ST6 CPU Programming Mode



3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}). In addition, a Low Frequency Auxiliary Oscillator (LFAO) can be switched in for security reasons, to reduce power consumption, or to offer the benefits of a back-up clock system.

The Oscillator Safeguard (OSG) option filters spikes from the oscillator lines, provides access to the LFAO to provide a backup oscillator in the event of main oscillator failure and 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 13, Figure 14, Figure 15 and Figure 16.

Figure 12 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. C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 31 and Figure 32.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 15.

With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

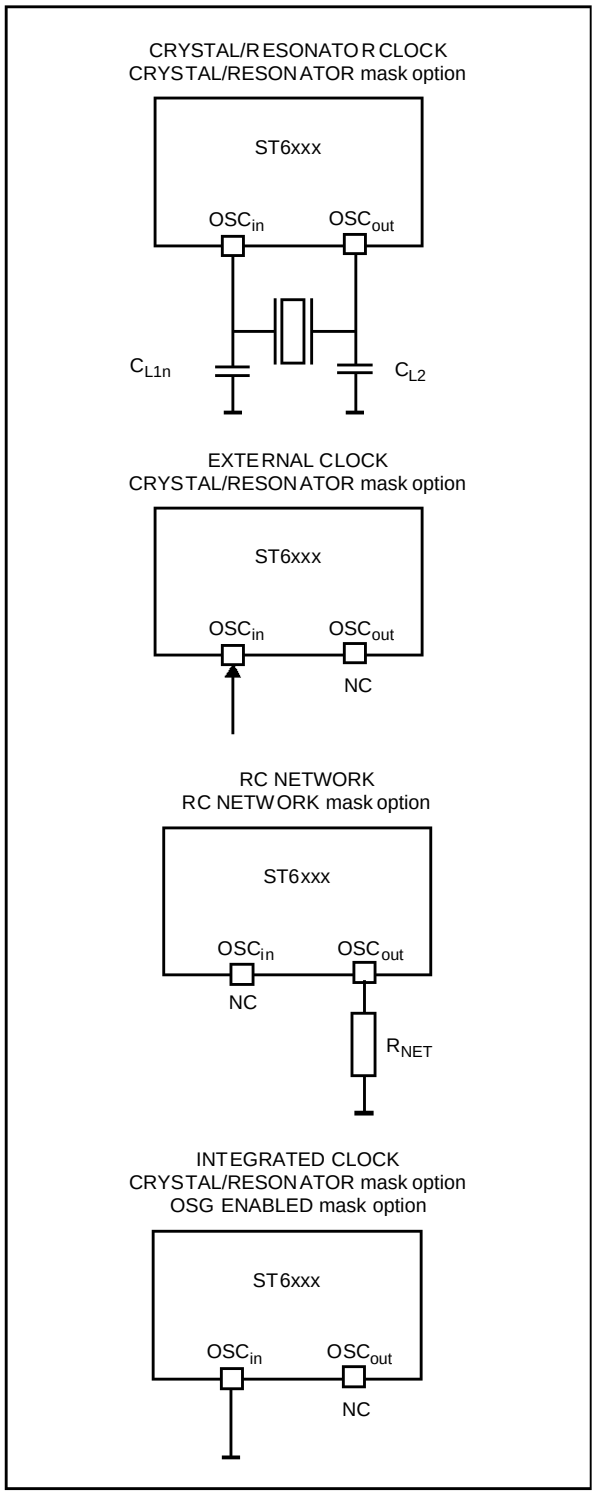
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate mask option. 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 main oscillator can be turned off (when the OSG ENABLED mask option is selected) by setting the OSCOFF bit of the ADC Control Register. The Low Frequency Auxiliary Oscillator is automatically started.

Figure 12. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Turning on the main oscillator is achieved by resetting the OSCOFF bit of the A/D Converter Control Register or by resetting the MCU. Restarting the main oscillator implies a delay comprising the oscillator start up delay period plus the duration of the software instruction at f_{LFAO} clock frequency.

3.1.2 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 safety oscillator in case of main oscillator failure.

This oscillator is available when the OSG ENABLED mask option is selected. In this case, it automatically starts one of its periods after the first missing edge from the main oscillator, whatever the reason (main oscillator defective, no clock circuitry provided, main oscillator switched off...).

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 1MHz.

At power on, the Low Frequency Auxiliary Oscillator starts faster than the Main Oscillator. It therefore feeds the on-chip counter generating the POR delay until the Main Oscillator runs.

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

ADCR

Address: 0D1h — Read/Write

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

Bit 7-3, 1-0= **ADCR7-ADCR3, ADCR1-ADCR0**: *ADC Control Register.* These bits are not used.

Bit 2 = **OSCOFF**. When low, this bit enables main oscillator to run. The main oscillator is switched off when OSCOFF is high.

3.1.3 Oscillator Safe Guard

The Oscillator Safe Guard (OSG) affords drastically increased operational integrity in ST62xx devices. The OSG circuit provides three basic functions: it filters spikes from the oscillator lines which would result in over frequency to the ST62 CPU; it gives access to the Low Frequency Auxiliary Oscillator (LFAO), used to ensure minimum processing in case of main oscillator failure, to offer reduced power consumption or to provide a fixed frequency low cost oscillator; finally, it automatically limits the internal clock frequency as a function of supply voltage, in order to ensure correct operation even if the power supply should drop.

The OSG is enabled or disabled by choosing the relevant OSG mask option. It may be viewed as a filter whose cross-over frequency is device dependent.

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 13). 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. This relationship is illustrated in Figure 16.

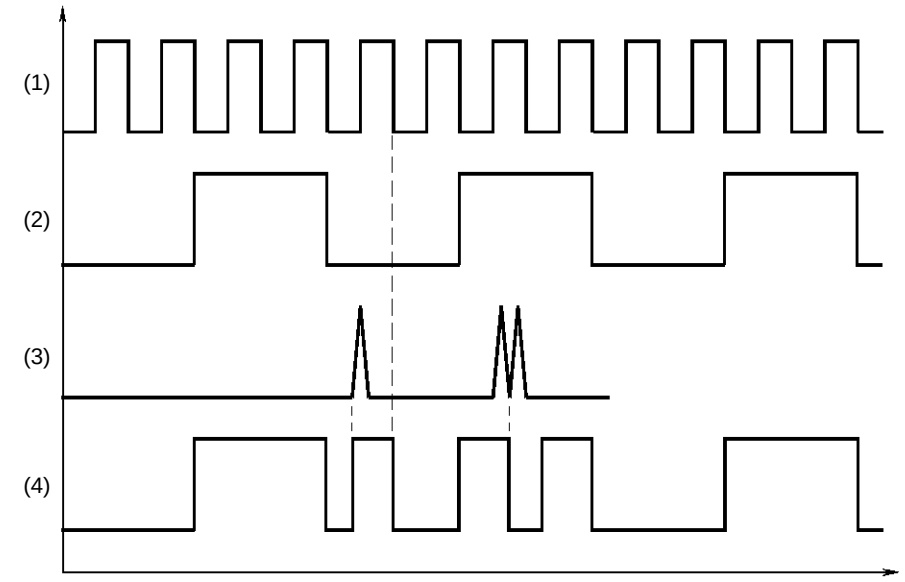
When the OSG is enabled, the Low Frequency Auxiliary Oscillator may be accessed. This oscillator starts operating after the first missing edge of the main oscillator (see Figure 14).

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.

Note. The OSG should be used wherever possible as it provides maximum safety. Care must be taken, however, as it can increase power consumption and reduce the maximum operating frequency to f_{OSG}.

CLOCK SYSTEM (Cont'd)

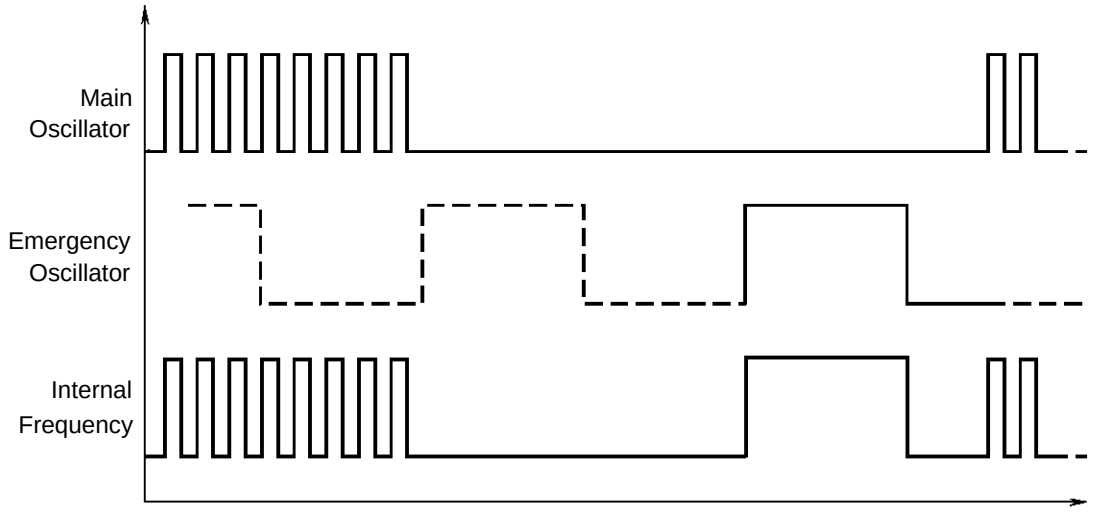
Figure 13. OSG Filtering Principle



(1) Maximum Frequency for the device to work correctly
(2) Actual Quartz Crystal Frequency at OSCin pin
(3) Noise from OSCin
(4) Resulting Internal Frequency

VR001932

Figure 14. OSG Emergency Oscillator Principle



VR001933

CLOCK SYSTEM (Cont'd)

Figure 15. Clock Circuit Block Diagram

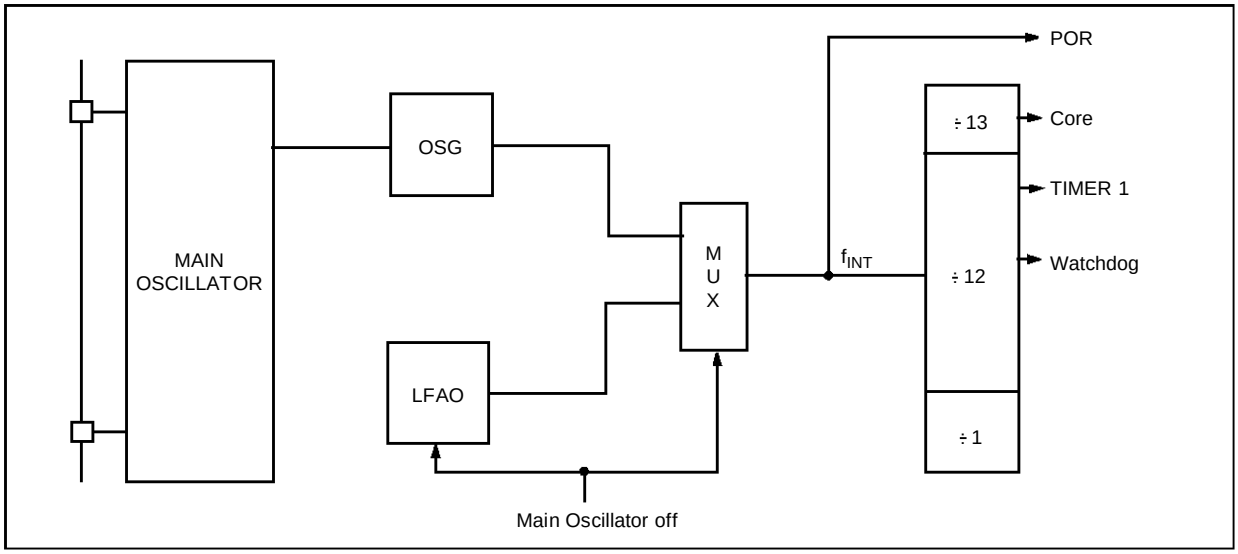
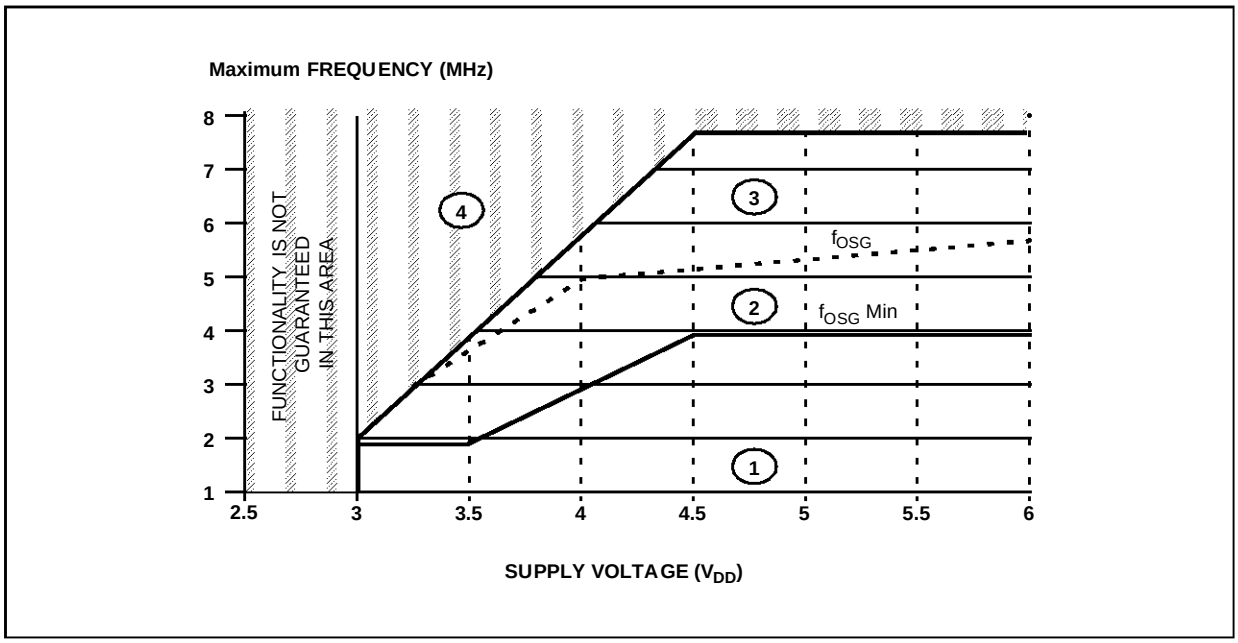


Figure 16. Maximum Operating Frequency (f_{MAX}) versus Supply Voltage (V_{DD})



Notes:

1. In this area, operation is guaranteed at the quartz crystal frequency.
2. When the OSG is disabled, operation in this area is guaranteed at the crystal frequency. When the OSG is enabled, operation in this area is guaranteed at a frequency of at least $f_{OSG\ Min.}$
3. When the OSG is disabled, operation in this area is guaranteed at the quartz crystal frequency. When the OSG is enabled, access to this area is prevented. The internal frequency is kept a f_{OSG}.
4. When the OSG is disabled, operation in this area is not guaranteed
When the OSG is enabled, access to this area is prevented. The internal frequency is kept at f_{OSG}.

3.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

3.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

3.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

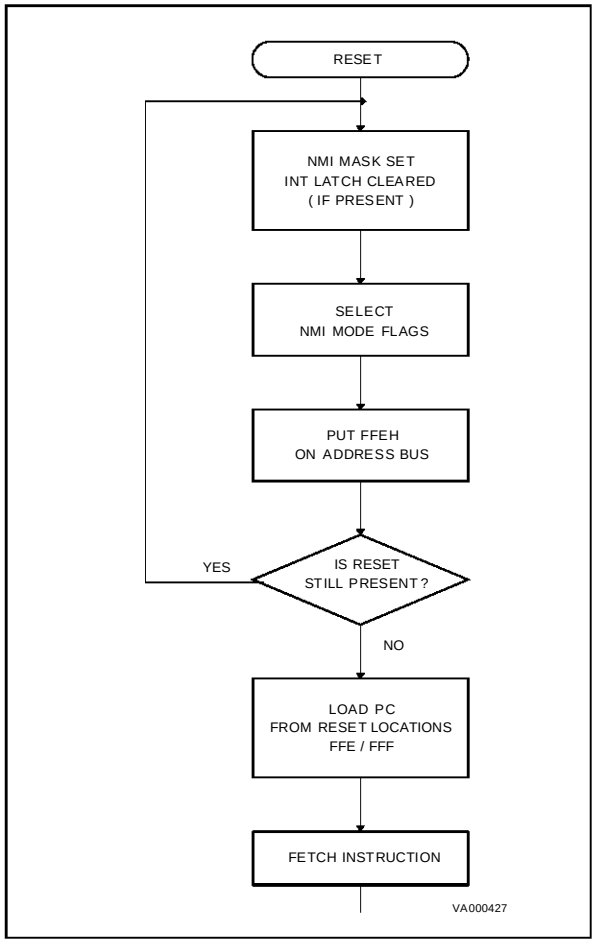
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 17. Reset and Interrupt Processing



RESETS (Cont'd)

3.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

3.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

3.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 18. Reset and Interrupt Processing

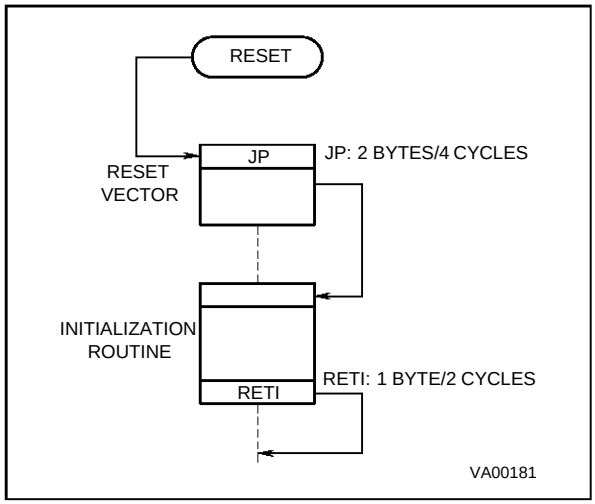
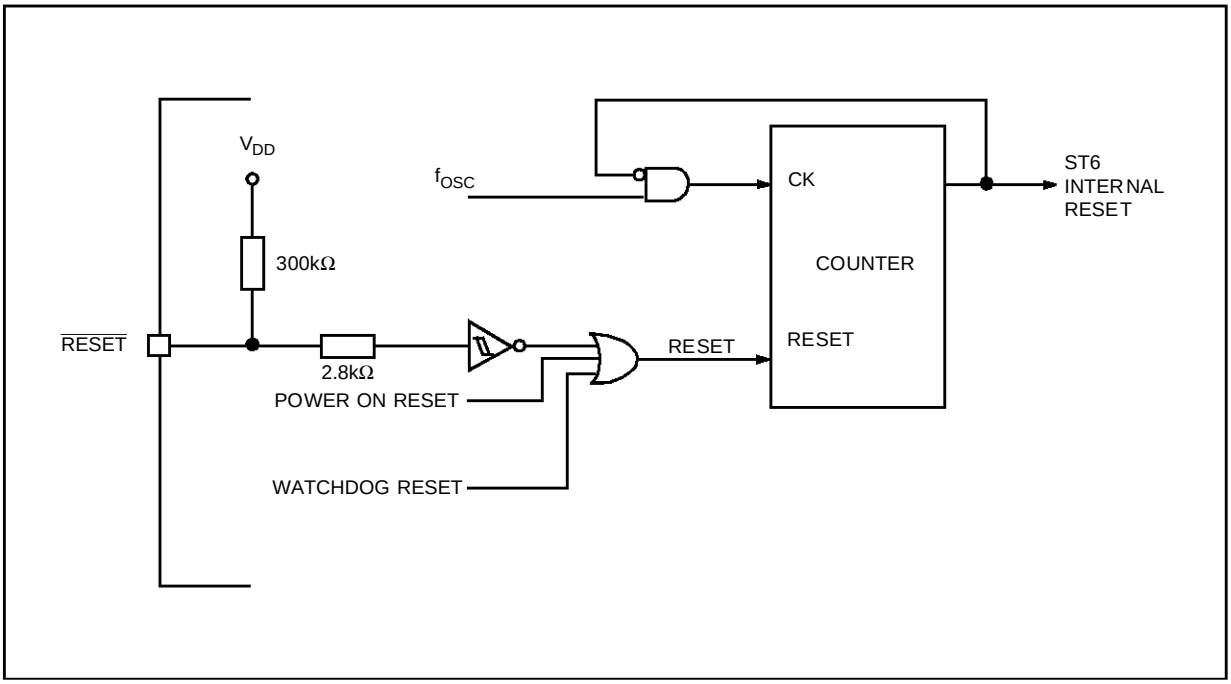


Figure 19. Reset Block Diagram



RESETS (Cont'd)

Table 2. Register Reset Status

Register	Address(es)	Status	Comment
Port Data Registers (PA, PB)	0C0h to 0C1h		
Port Direction Register (PA, PB)	0C4h to 0C5h		I/Os are Inputs with pull-up
Port Option Register (PA, PB)	0CCh to 0CDh	00h	I/Os are Inputs with pull-up
Interrupt Option Register	0C8h		Interrupts disabled
Timer Status/Control	0D4h		Timer disabled
X, Y, V, W Register	080h to 083h		
Accumulator	0FFh		
Data RAM	084h to 0BFh	Undefined	
Data ROM Window Register	0C9h		
A/D Result Register	0D0h		
Timer Counter Register	0D3h	FFh	
Timer Prescaler Register	0D2h	7Fh	Maximum count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Stand-by, main oscillator on

3.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two mask options, known as “WATCHDOG ACTIVATION” (i.e. HARDWARE or SOFTWARE) and “EXTERNAL STOP MODE CONTROL” (see Table 3).

In the SOFTWARE mask option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, save by resetting the MCU.

In the HARDWARE mask option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL mask option (available in ROM versions only) has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL mask option has been selected, port PB0 must be defined as an open-drain output, and PA2 as an input.

Table 3. Recommended Mask Option Choices

Functions Required	Recommended Mask Options
Stop Mode & Watchdog	“EXTERNAL STOP MODE” & “HARDWARE WATCHDOG”
Stop Mode	“SOFTWARE WATCHDOG”
Watchdog	“HARDWARE WATCHDOG”

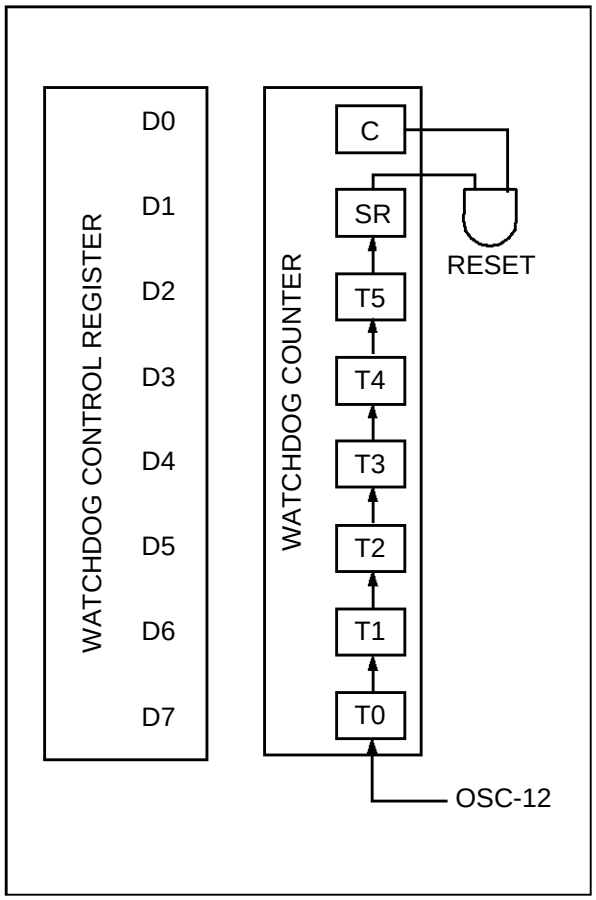
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 3.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 20.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 20. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 21) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 21. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

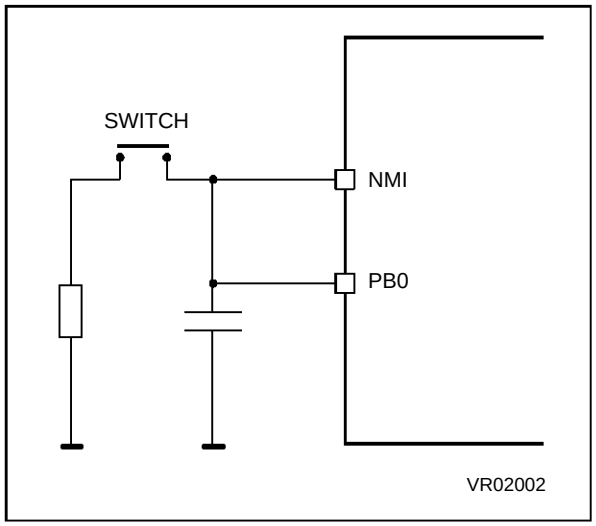
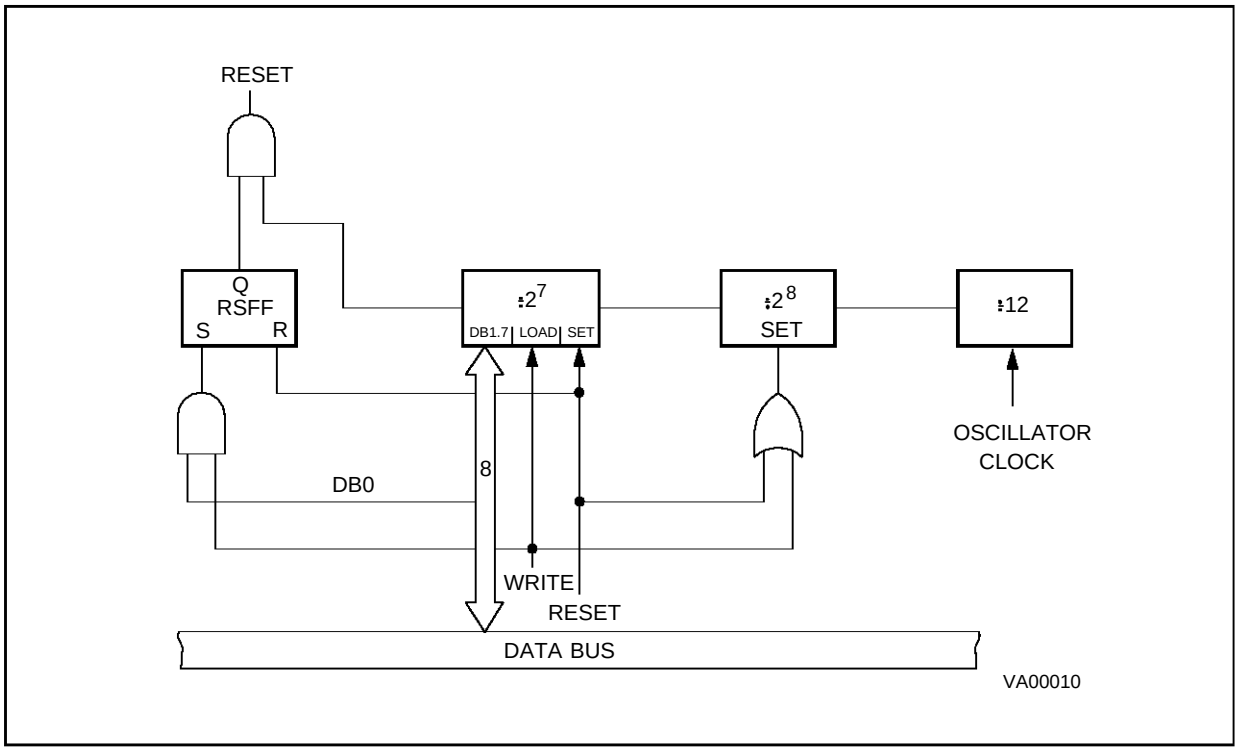


Figure 22. Digital Watchdog Block Diagram



3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 4).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 4. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A pins	Interrupt vector #1	(FF6h-FF7h)
Port B pins	Interrupt vector #2	(FF4h-FF5h)
TIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
ADC peripheral	Interrupt vector #4	(FF0h-FF1h)

3.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named. It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port B pins is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named . It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the TIMER peripheral and vector #4 with the A/D Converter peripheral.

Each on-chip peripheral has an associated interrupt request flag (TMZ for the Timer, EOC for the A/D Converter), which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (ETI for the Timer, EAI for the A/D Converter), which must be set to “1” to enable the associated interrupt request.

3.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

3.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port B) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 5. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A
	CLEARED	Falling edge mode on Port A
ESB	SET	Level sensitive mode on Port B
	CLEARED	Falling edge mode on Port B

3.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A-vector #1, Port B-vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

IINTERRUPTS (Cont'd)

3.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

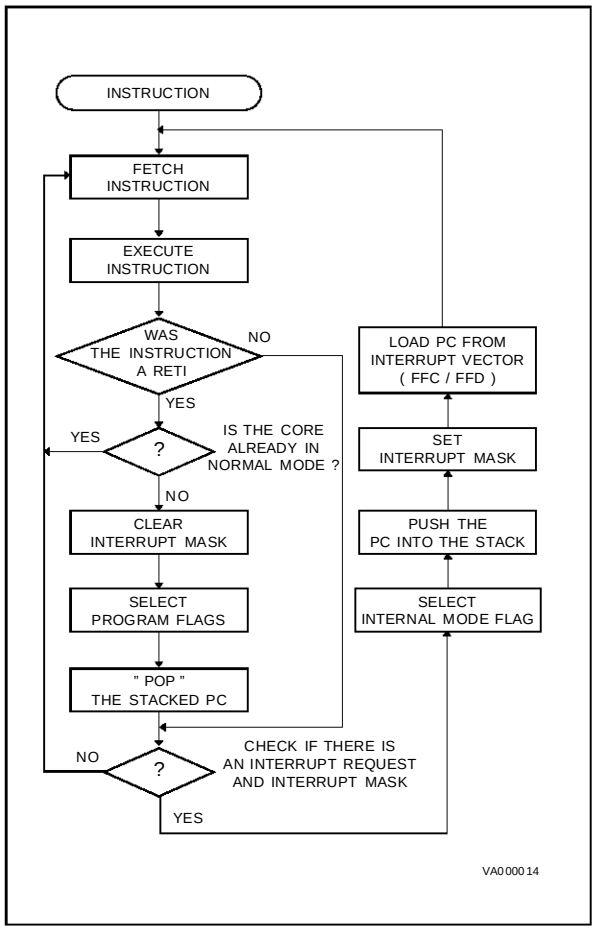
– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 23. Interrupt Processing Flow Chart

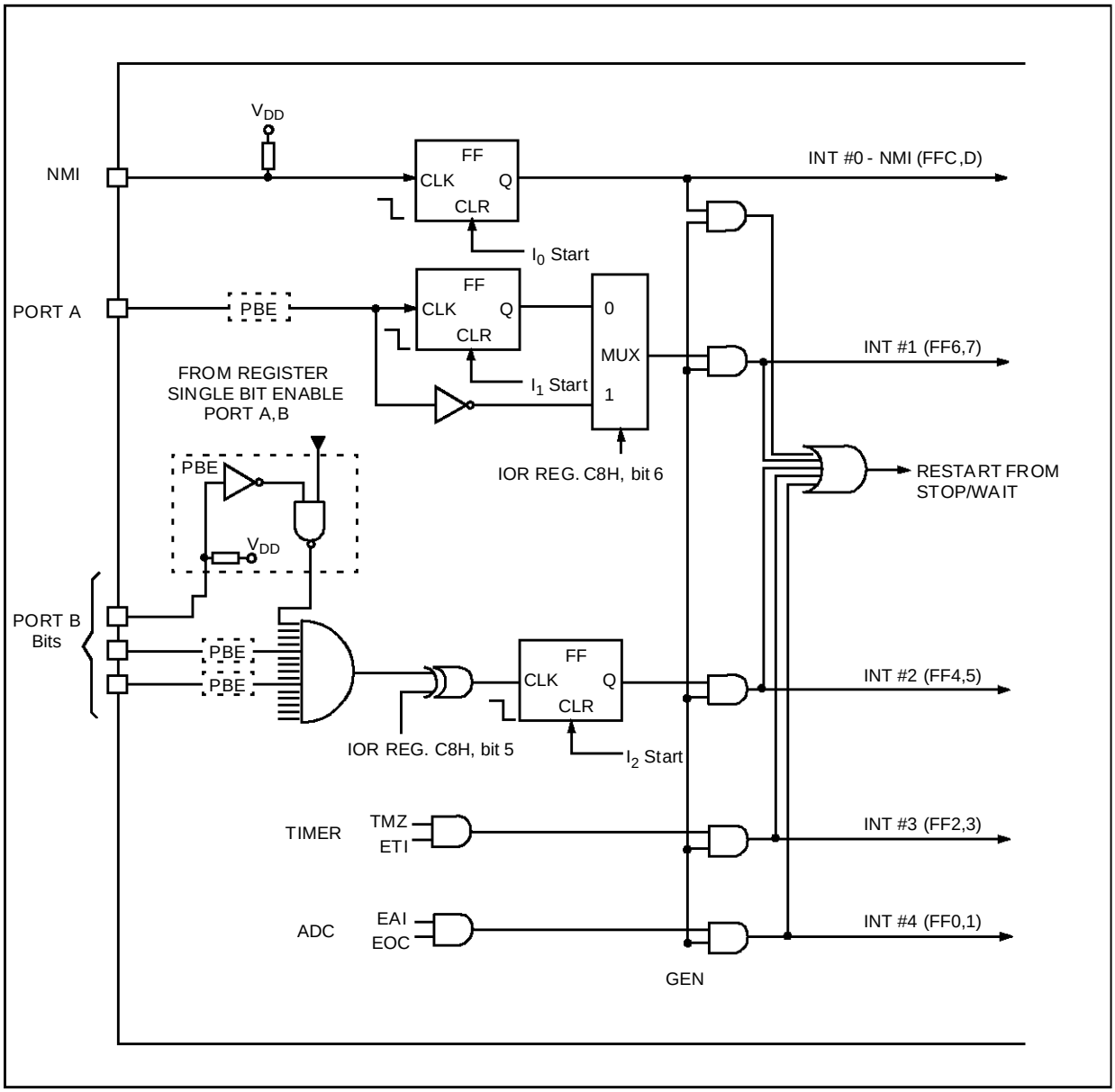


INTERRUPTS (Cont'd)

Table 6. Interrupt Requests and Mask Bits

Peripheral	Register	Address Register	Mask bit	Masked Interrupt Source	Interrupt vector
GENERAL	IOR	C8h	GEN	All Interrupts, excluding NMI	
TIMER	TSCR	D4h	ETI	TMZ: TIMER Overflow	Vector 3
A/D CONVERTER	ADCR	D1h	EAI	EOC: End of Conversion	Vector 4
Port PAn	ORPA-DRPA	C4h-CCh	ORPAn-DRPAn	PAn pin	Vector 1
Port PBn	ORPB-DRPB	C5h-CDh	ORPBn-DRPBn	PBn pin	Vector 2

Figure 24. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

In addition, the Low Frequency Auxiliary Oscillator (LFAO) can be used instead of the main oscillator to reduce power consumption in RUN and WAIT modes.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator (main oscillator or LFAO) is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the power consumption has to be further reduced, the Low Frequency Auxiliary Oscillator (LFAO) can be used in place of the main oscillator, if its operating frequency is lower. If required, the LFAO must be switched on before entering the WAIT mode.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

3.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection, consequently, when the LFAO is used, the user program must manage oscillator selection as soon as normal RUN mode is resumed.

3.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

3.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

3.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

– selecting the Low Frequency Auxiliary Oscillator (provided this runs at a lower frequency than the main oscillator).

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

The MCU features 12 Input/Output lines which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Analog input (PB4-PB7)

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PA0-PA3 only)

The lines are organized as two Ports (A and B).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The two DATA registers (DRA and DRB), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

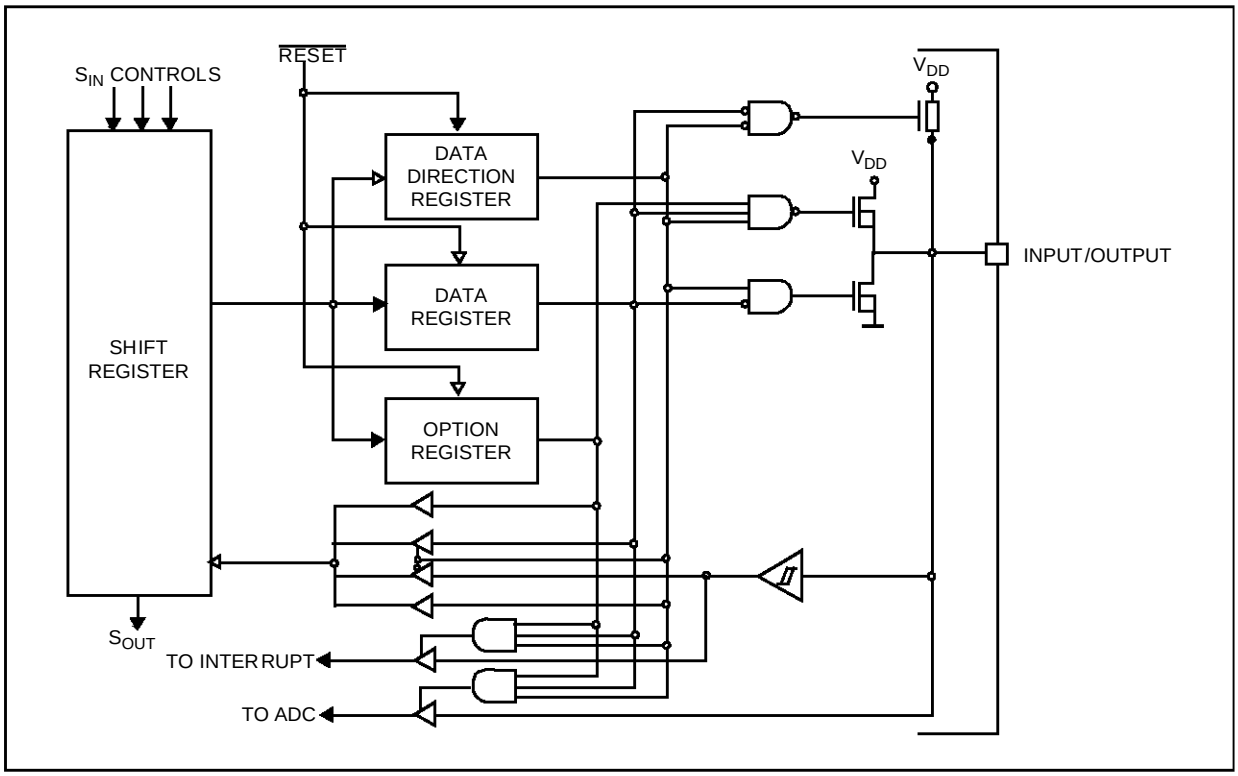
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The two Data Direction registers (DDRA and DDRB) allow the data direction (input or output) of each pin to be set.

The two Option registers (ORA and ORB) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 25. I/O Port Block Diagram



I/O PORTS (Cont'd)

4.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 7 illustrates the various port configurations which can be selected by user software.

4.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

4.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A are AND-connected to the interrupt associated with Vector #1. The pins of Port B are AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

4.1.1.3 Analog Input Options

The three pins, PB5-PB7, can be configured as analog inputs by programming the OR and DR registers accordingly. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

4.1.2 I/O Port Option Registers

ORA/B (CCh PA, CDh PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Option Register bits.*

4.1.3 I/O Port Data Direction Registers

DDRA/B (C4h PA, C5h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Direction Registers bits.*

4.1.4 I/O Port Data Registers

DRA/B (C0h PA, C1h PB)
Read/Write

7							0
PA7/PB7	PA6/PB6	PA5/PB5	PA4/PB4	PA3/PB3	PA2/PB2	PA1/PB1	PA0/PB0

Bit 7-0 = **PA/PB7. PA/PB0**: *Port A, B Data Registers bits.*

Table 7. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Input	No pull-up, no interrupt (PA0-PA3 pins)
			Input	Analog input (PB4-PB7 pins)
1	0	X	Output	20mA sink open-drain output (PA0-PA3 pins)
1	0	X	Output	Standard open-drain output (PB0-PB7 pins)
1	1	X	Output	20mA sink push-pull output (PA0-PA3 pins)

Note: X = Don't care

I/O PORTS (Cont'd)

4.1.5 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 26. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A and B Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

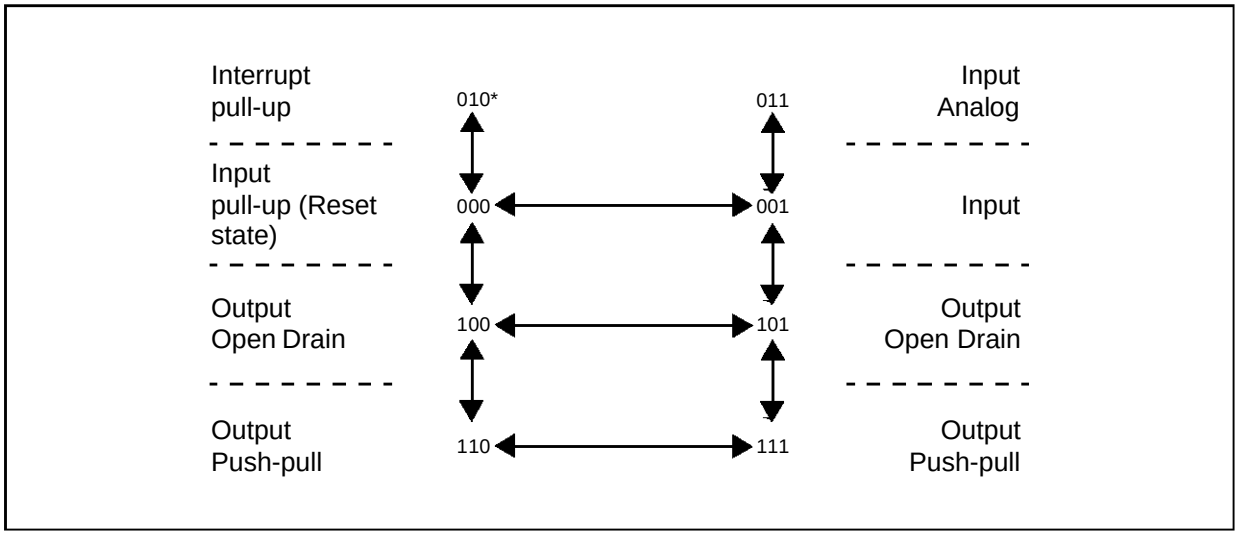
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

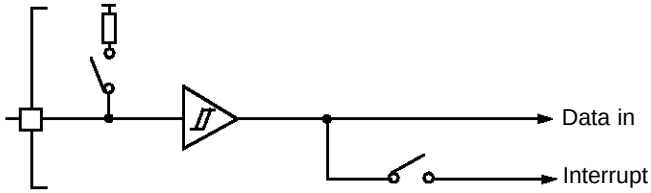
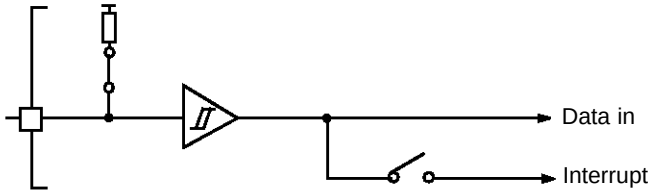
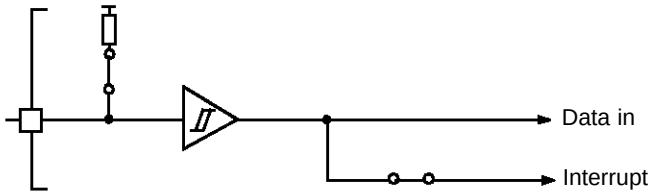
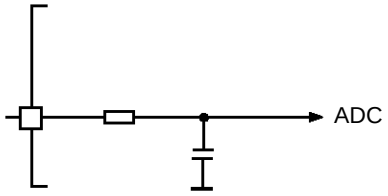
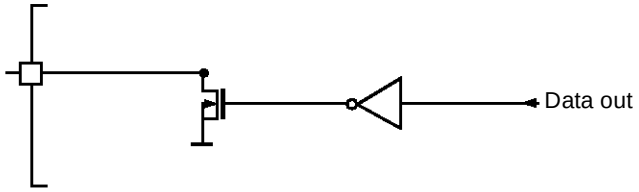
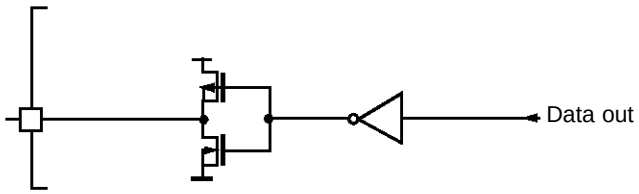
Figure 26. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 8. I/O Port Option Selections

MODE	AVAILABLE ON(1)	SCHEMATIC
Input	PA0-PA3 PB0-PB7	
Input with pull up	PA0-PA3 PB0-PB7	
Input with pull up with interrupt	PA0-PA3 PB0-PB7	
Analog Input	PB4-PB7	
Open drain output 5mA Open drain output 20mA	PB0-PB7 PA0-PA3	
Push-pull output 5mA Push-pull output 20mA	PB0-PB7 PA0-PA3	

Note 1. Provided the correct configuration has been selected.

4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. The peripheral may be configured in three different operating modes.

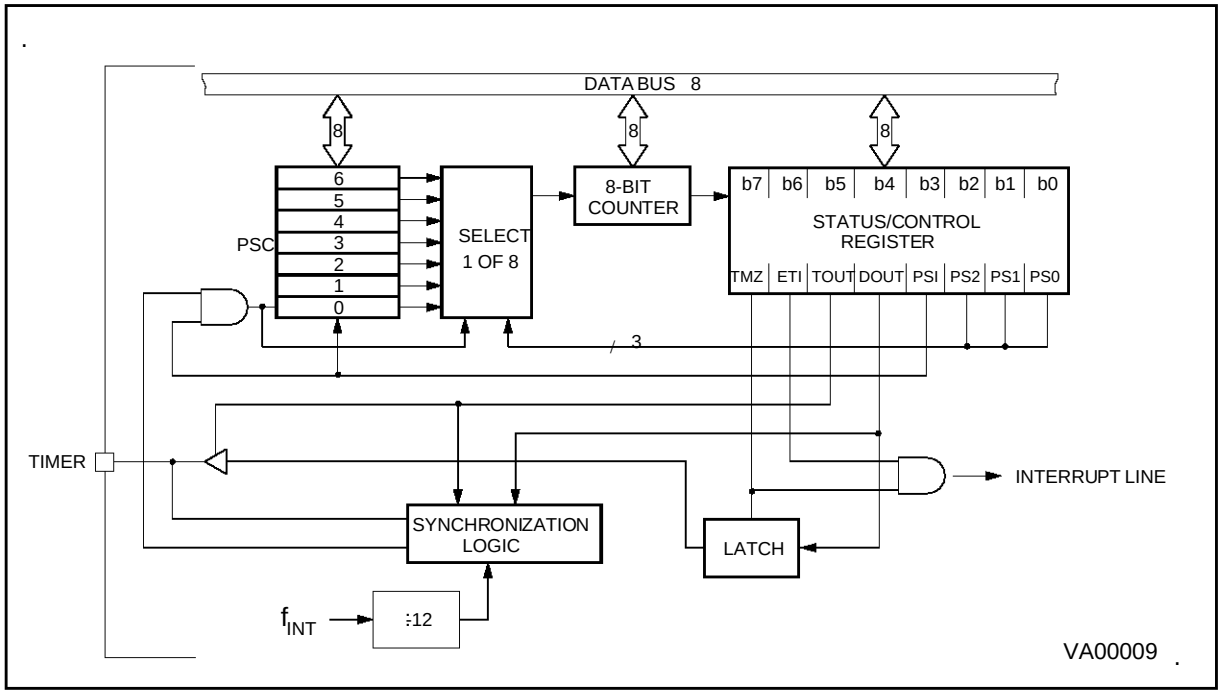
Figure 27 shows the Timer Block Diagram. The external TIMER pin is available to the user. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero) bit in the TSCR is set to "1". If the ETI (Enable Timer Interrupt) bit in the TSCR is also set to "1", an interrupt request, associated with interrupt vector #3, is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input can be the internal frequency f_{INT} divided by 12 or an external clock applied to the TIMER pin. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 10), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to "1" to allow the prescaler (and hence the counter) to start. If it is cleared to "0", all the prescaler bits are set to "1" and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set to "1". The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 28 illustrates the Timer's working principle.

Figure 27. Timer Block Diagram



TIMER (Cont'd)

4.2.1 Timer Operating Modes

There are three operating modes, which are selected by the TOUT and DOUT bits (see TSCR register). These three modes correspond to the two clocks which can be connected to the 7-bit prescaler ($f_{INT} \div 12$ or TIMER pin signal), and to the output mode.

4.2.2 Gated Mode

(TOUT = "0", DOUT = "1")

In this mode the prescaler is decremented by the Timer clock input ($f_{INT} \div 12$), but ONLY when the signal on the TIMER pin is held high (allowing pulse width measurement). This mode is selected by clearing the TOUT bit in the TSCR register to "0" (i.e. as input) and setting the DOUT bit to "1".

4.2.3 Clock Input Mode

(TOUT = "0", DOUT = "0")

In this mode, the TIMER pin is an input and the prescaler is decremented on the rising edge.

4.2.4 Output Mode

(TOUT = "1", DOUT = data out)

The TIMER pin is connected to the DOUT latch, hence the Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and transfer it to the TIMER pin. This operating mode allows external signal generation on the TIMER pin.

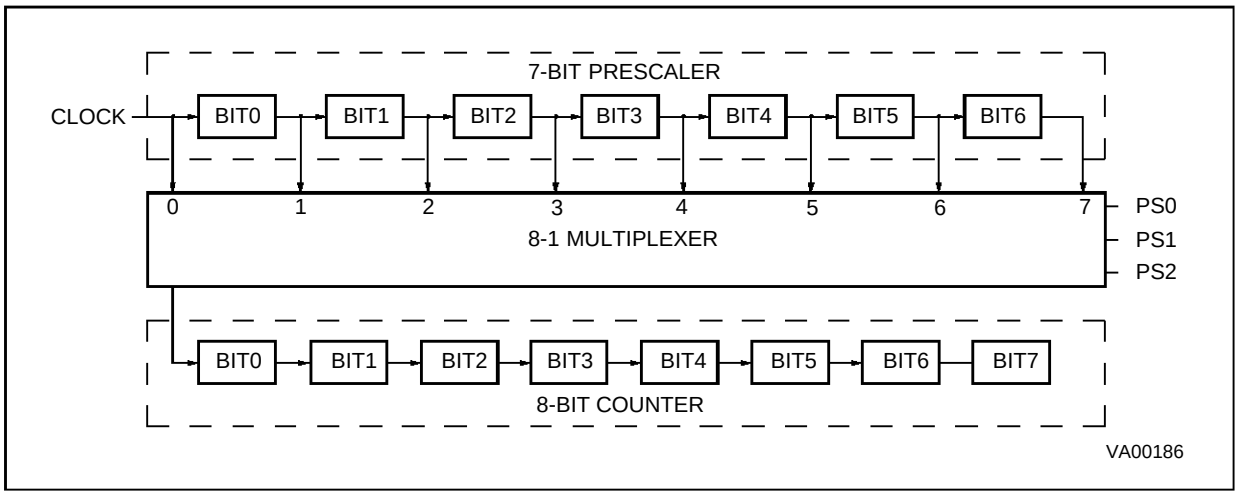
Table 9. Timer Operating Modes

TOUT	DOUT	Timer Pin	Timer Function
0	0	Input	Event Counter
0	1	Input	Gated Input
1	0	Output	Output "0"
1	1	Output	Output "1"

4.2.5 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

Figure 28. Timer Working Principle



TIMER (Cont'd)

4.2.6 Application Notes

The user can select the presence of an on-chip pull-up on the TIMER pin as a ROM mask option (see Option List at the end of the Datasheet).

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

If the Timer is programmed in output mode, the DOUT bit is transferred to the TIMER pin when TMZ is set to one (by software or due to counter decrement). When TMZ is high, the latch is transparent and DOUT is copied to the timer pin. When TMZ goes low, DOUT is latched.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time

.

4.2.7 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	TOUT	DOUT	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrupt*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **TOUT**: *Timers Output Control*

When low, this bit selects the input mode for the TIMER pin. When high the output mode is selected.

Bit 4 = **DOUT**: *Data Output*

Data sent to the timer output when TMZ is set high (output mode only). Input mode selection (input mode only).

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 10. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register TCR

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

4.3 A/D CONVERTER (ADC)

The A/D converter peripheral is an 8-bit analog to digital converter with analog inputs as alternate I/O functions (the number of which is device dependent), offering 8-bit resolution with a typical conversion time of 70us (at an oscillator clock frequency of 8MHz).

The ADC converts the input voltage by a process of successive approximations, using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, conversion accuracy is decreased.

Selection of the input pin is done by configuring the related I/O line as an analog input via the Option and Data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously, to avoid device malfunction.

The ADC uses two registers in the data space: the ADC data conversion register, ADR, which stores the conversion result, and the ADC control register, ADCR, used to program the ADC functions.

A conversion is started by writing a “1” to the Start bit (STA) in the ADC control register. This automatically clears (resets to “0”) the End Of Conversion Bit (EOC). When a conversion is complete, the EOC bit is automatically set to “1”, in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

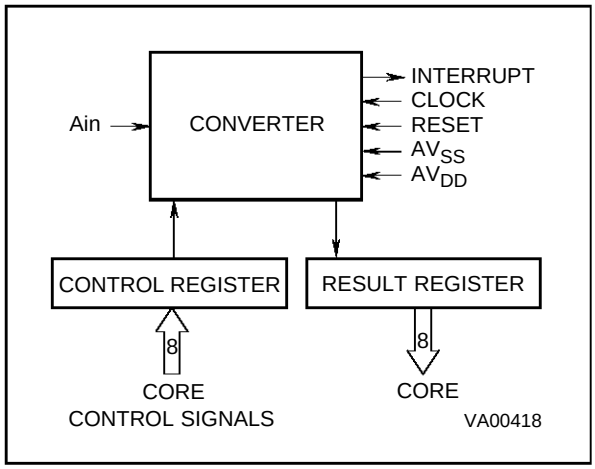
The STA bit is continuously scanned so that, if the user sets it to “1” while a previous conversion is in progress, a new conversion is started before completing the previous one. The start bit (STA) is a write only bit, any attempt to read it will show a logical “0”.

The A/D converter features a maskable interrupt associated with the end of conversion. This interrupt is associated with interrupt vector #4 and occurs when the EOC bit is set (i.e. when a conversion is completed). The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. This is done by setting the PDS bit in the ADC control register to “0”. If PDS=“1”, the A/D is powered and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow stabilisation of the A/D converter. This action is also needed before entering WAIT mode, since the A/D comparator is not automatically disabled in WAIT mode.

During Reset, any conversion in progress is stopped, the control register is reset to 40h and the ADC interrupt is masked (EAI=0).

Figure 29. ADC Block Diagram



4.3.1 Application Notes

The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for the optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 12pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. In worst case conditions, the impedance, ASI, of the analog voltage source is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband. ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

A/D CONVERTER (Cont'd)

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

The accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

The converter resolution is given by::

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

In order to optimise conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the less significant bits are determined.

The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. Indeed, only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. It should be noted that waking up the microcontroller could also be done using the Timer interrupt, but in this case the Timer will be working and the resulting noise could affect conversion accuracy.

A/D Converter Control Register (ADCR)

Address: 0D1h — Read/Write

7							0
EAI	EOC	STA	PDS	D3	D2	D1	D0

Bit 7 = **EAI**: *Enable A/D Interrupt.* If this bit is set to "1" the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

Bit 6 = **EOC**: *End of conversion. Read Only*. This read only bit indicates when a conversion has been completed. This bit is automatically reset to "0" when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to "1".

Bit 5 = **STA***: Start of Conversion. Write Only*. Writing a "1" to this bit will start a conversion on the selected channel and automatically reset to "0" the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

Bit 4 = **PDS***: Power Down Selection.* This bit activates the A/D converter if set to "1". Writing a "0" to this bit will put the ADC in power down mode (idle mode).

Bit 3-0 = **D3-D0.** Not used

A/D Converter Data Register (ADR)

Address: 0D0h — Read only

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0***: 8 Bit A/D Conversion Result.*

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 11. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:

X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 12. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 13. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected. The tested bit is shifted into carry.
* . Not Affected

Table 14. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not<M> Affected

Table 15. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ . Affected
*. Not Affected

Table 16. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
* . Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

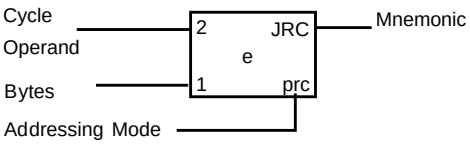
LOW / HI	**0 0000**	**1 0001**	**2 0010**	**3 0011**	**4 0100**	**5 0101**	**6 0110**	**7 0111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

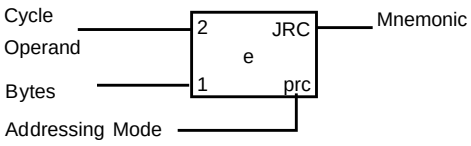
LOW / HI	**8 1000**	**9 1001**	**A 1010**	**B 1011**	**C 1100**	**D 1101**	**E 1110**	**F 1111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

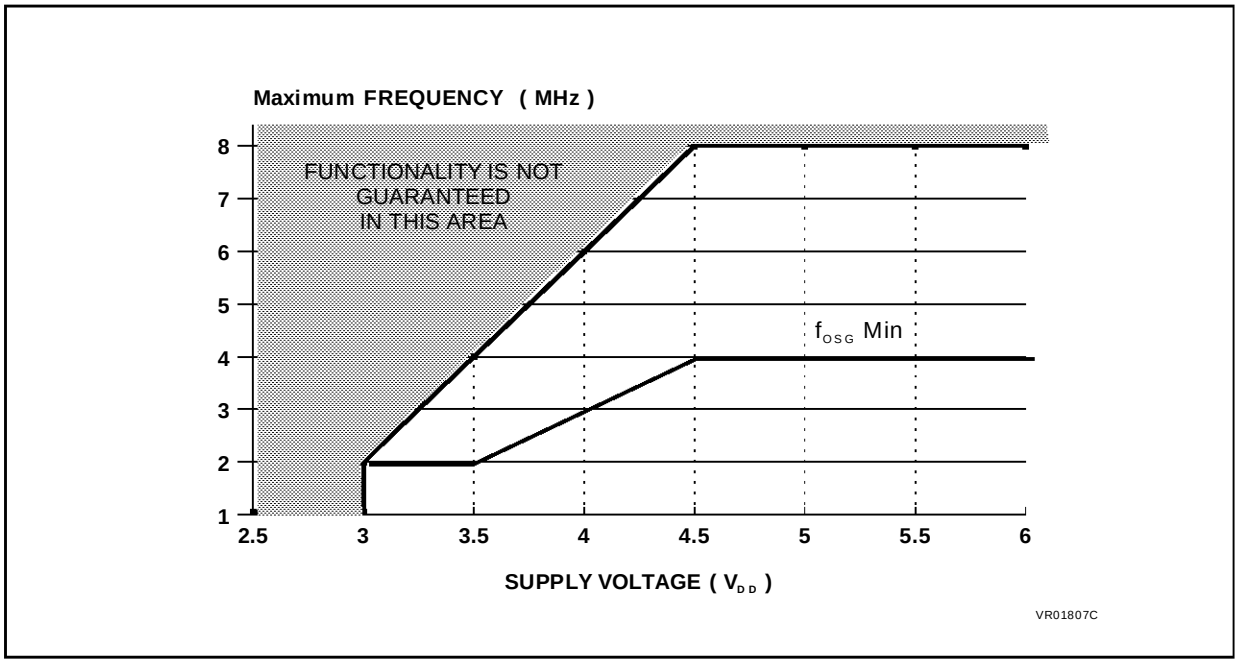
THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	°C/W
		PSO20			80	

6.2 RECOMMENDED OPERATING CONDITIONS

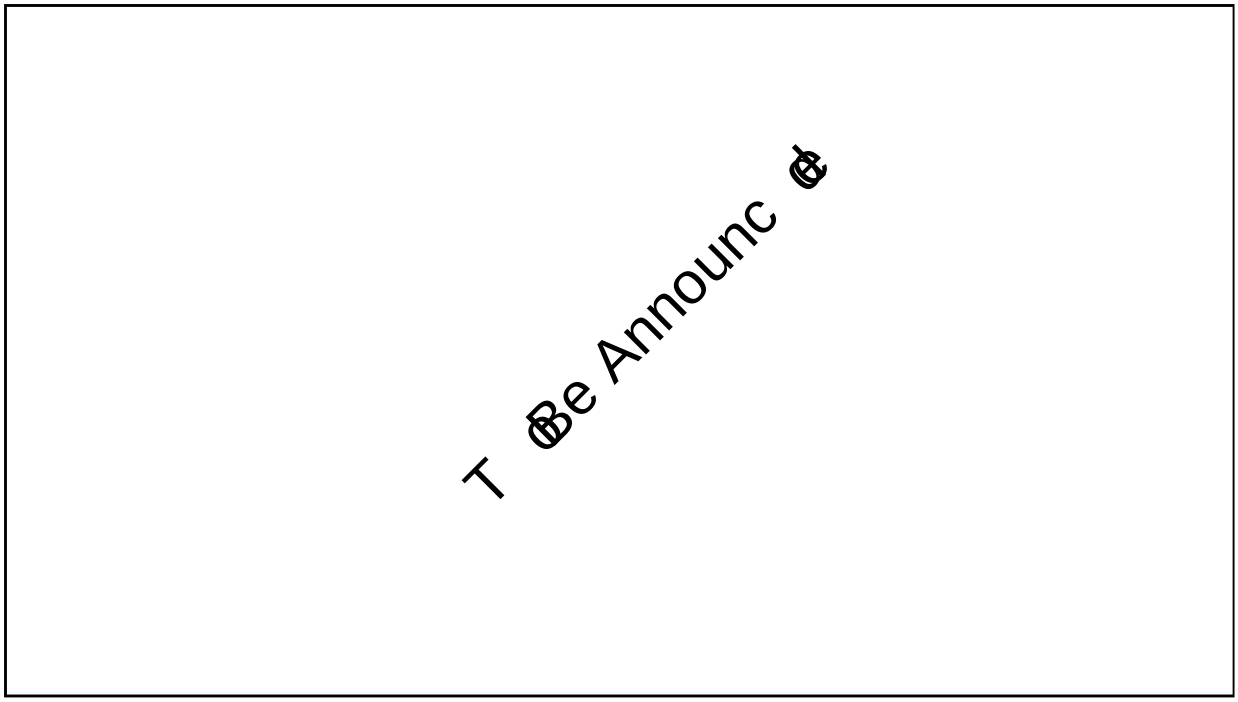
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input	V_{DD} = 4.5 to 5.5V			-5	mA

Figure 30. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



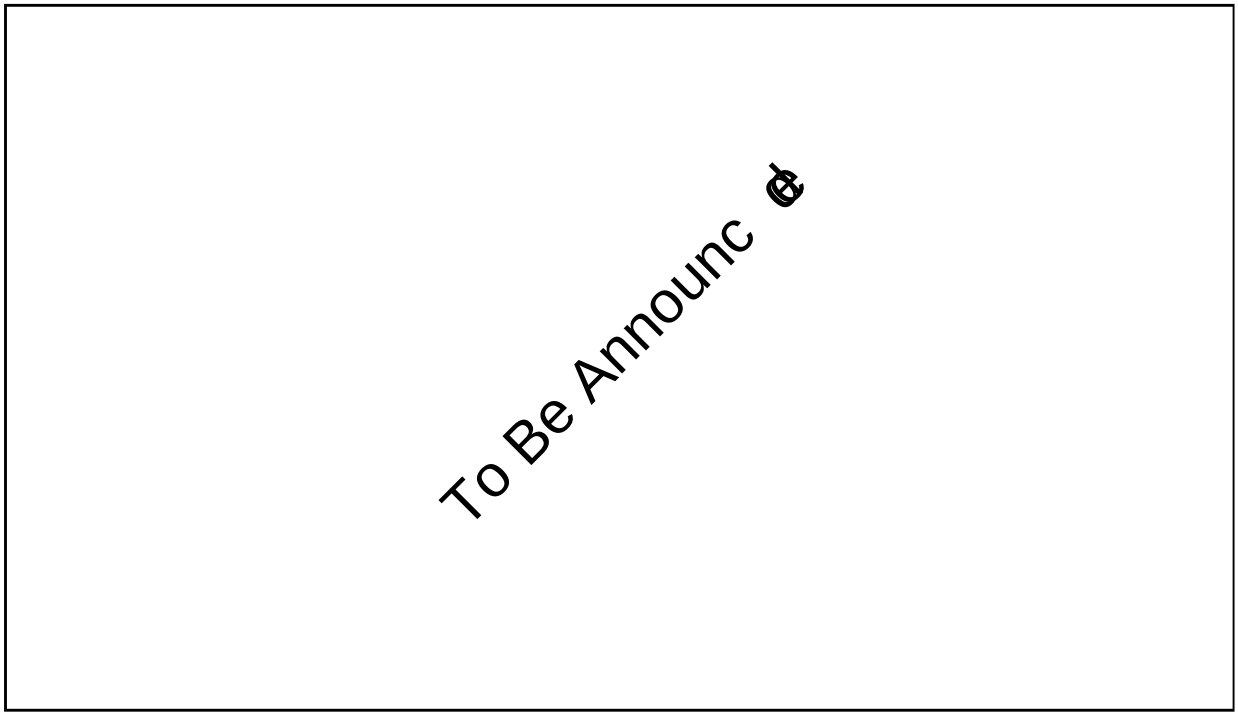
The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 31. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 32. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

6.3 READOUT PROTECTION FUSE

If the ROM READOUT PROTECTION option is selected, the waveform illustrated below must be applied to the TEST pin in order to blow the fuse.

The following circuit can be used for this purpose:

Figure 33. Programming wave form

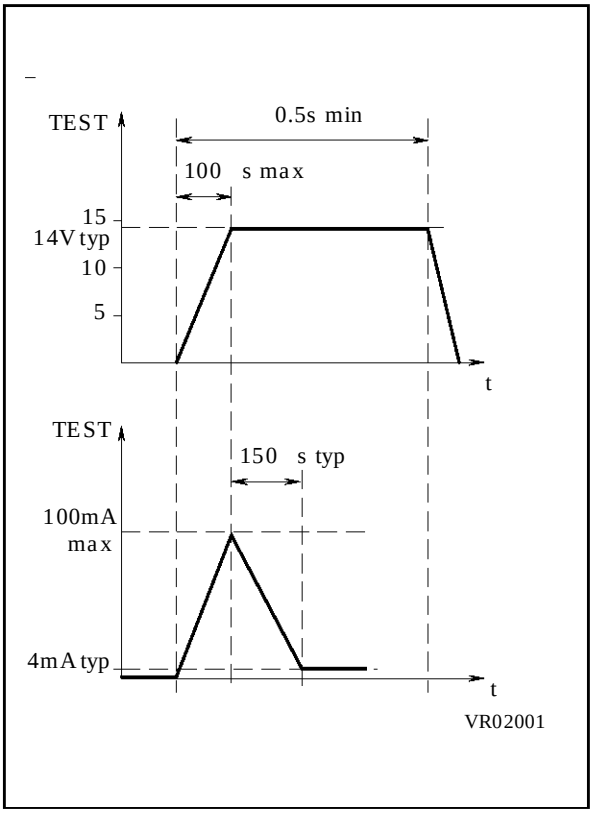
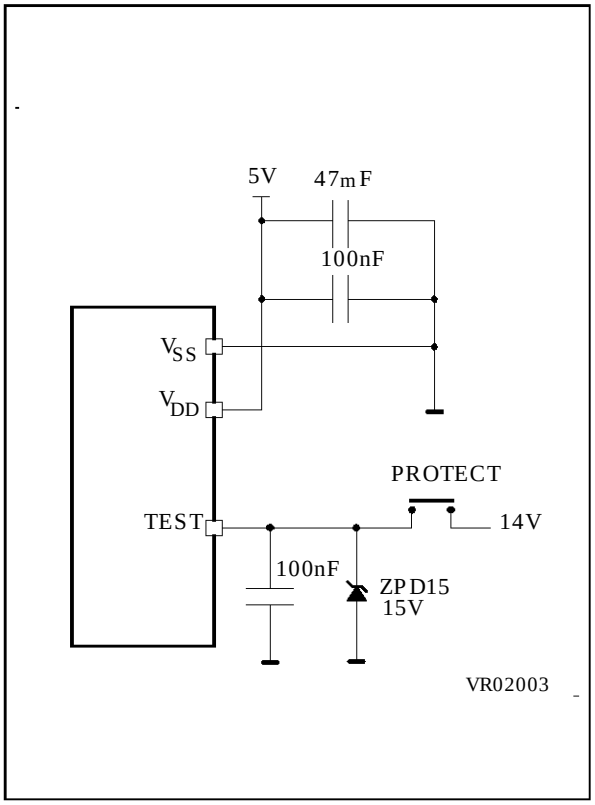


Figure 34. Programming Circuit

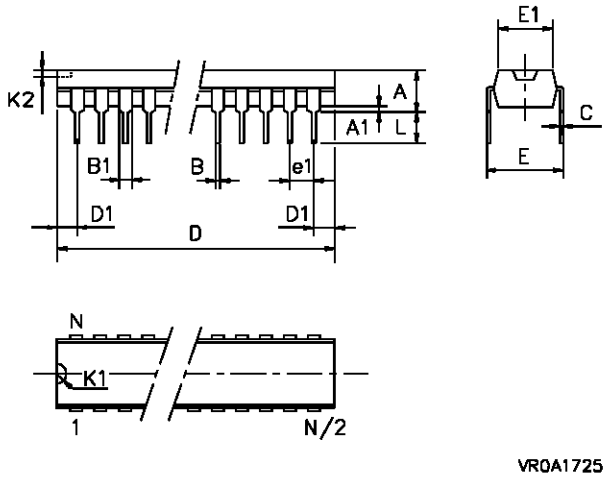


Note: ZPD15 is used for overvoltage protection

7 GENERAL INFORMATION

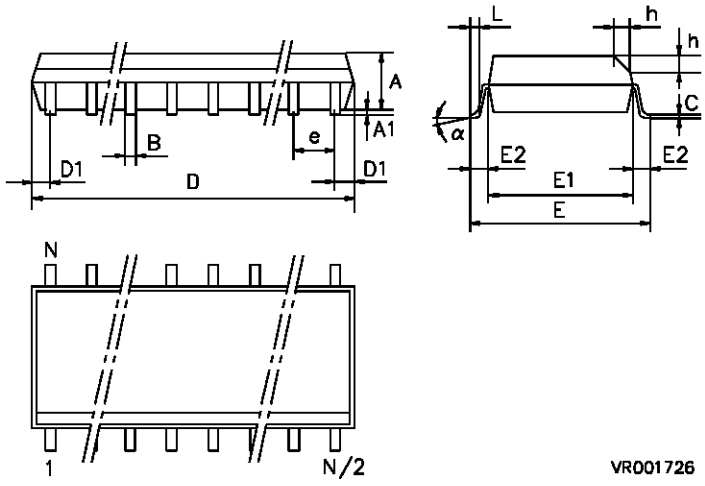
7.1 PACKAGE MECHANICAL DATA

Figure 35. 20-Pin Dual in Line Plastic (B), 300-Mil width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.185
A1	0.254			0.010		
B			0.46			0.017
B1	1.39		1.65	0.055		0.065
C			0.25			0.009
D			25.4			1.000
D1			1.34			0.053
E		8.5			0.334	
E1			7.1			0.279
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.30			0.129	
e1		2.54			0.10	
Number of Pins						
N	20					

Figure 36. 20-Lead Small Outline Plastic (M), 300-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.05		2.55	0.081		0.10
A1	0.1		0.3	0.004		.0118
B	0.35		0.49	.0138		.0192
C	0.23		0.32	.0091		.0125
D	12.6		13.0	.4961		.5118
D1	-	-	-	-	-	-
E	10.0		10.65	0.394		0.419
E1	7.40		7.60	.2914		.2992
E2	-	-	-	-	-	-
L	0.40		1.27	0.016		0.050
e		1.27			0.05	
α	-	-	-	-	-	-
Number of Pins						
N	20					

ST6209B MICROCONTROLLER OPTION LIST

Customer
Address

Contact
Phone No
Reference

SGS-THOMSON Microelectronics references

Device: [ ] ST6209B

Package: [ ] Dual in Line Plastic[ ] Small Outline Plastic
In this case, select conditioning
[ ] Standard (Stick)
[ ] Tape & Reel

Temperature Range: [ ] 0°C to + 70°C[ ] - 40°C to + 85°C

Special Marking: [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _ "

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Maximum character count: DIP20: 10
SO20: 8

Oscillator Source Selection:[ ] Crystal Quartz/Ceramic resonator (Default)
[ ] RC Network

Watchdog Selection: [ ] Software Activation (STOP mode available)
[ ] Hardware Activation (no STOP mode)

OSG: [ ] Enabled
[ ] Disabled (Default)

Input pull-up selection on NMI pin:[ ] Yes[ ] No

Input pull-up selection on TIMER pin: [ ] Yes[ ] No

ROM Readout Protection: [ ] Standard (Fuse cannot be blown)
[ ] Enabled (Fuse can be blown by the customer)

Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.

External STOP Mode Control[ ] Enabled[ ] Disabled (Default)

Comments :

Supply Operating Range in the application:

Oscillator Fequency in the application:

Notes

Signature
Date

7.2 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

7.2.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

7.2.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 17. ROM Memory Map for ST6209B

Device Address	Description
0000h-0B9Fh	Reserved
0BA0h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Table 18. Ordering Information

Sales Type	ROM	I/O	Additional Features	Temperature Range	Package
ST6209BB1/XXX ST6209BB6/XXX	1036 Bytes	12	A/D CONVERTER	0 to +70°C -40 to + 85°C	PDIP20
ST6209BM1/XXX ST6209BM6/XXX				0 to +70°C -40 to + 85°C	PSO20

SGS-THOMSON MICROELECTRONICS

ST62T10, T15, T20, T25
ST62E20, E25

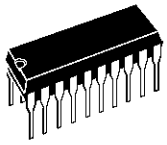
8-BIT OTP/EPROM MCUs WITH A/D CONVERTER

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in OTP/EPROM
- Data OTP/EPROM: User selectable size (in program EPROM)
- Data RAM: 64 bytes
- 12/20 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 4 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer with 7-bit programmable prescaler and external input
- Digital Watchdog
- 8-bit A/D Converter with 8 (ST62T10, T20, E20) or 16 (ST62T15, T25, E25) analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal or Ceramic resonator
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).

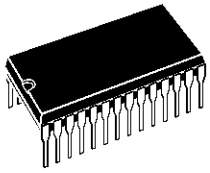
DEVICE SUMMARY

DEVICE	EPROM (Bytes)	OTP (Bytes)	I/O Pins
ST62T10		1836	12
ST62T20		3884	
ST62E20	3884		
ST62T15		1836	20
ST62T25		3884	
ST62E25	3884		

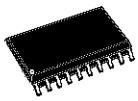
OTP PACKAGES



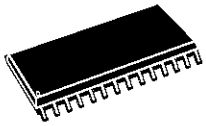
PDIP20



PDIP28

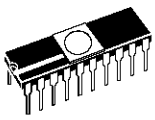


PSO20

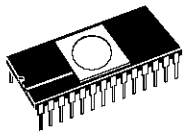


PSO28

EPROM PACKAGES



CDIP20W



CDIP28W

(See end of Datasheet for Ordering Information)

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T10, T15, T20 and T25 devices are low cost members of the 8-bit HCMOS ST62xx 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 ST62E20 device is an erasable EPROM version of the ST62T20 device, which may be used to emulate the T10 and T20 devices, as well as the respective ST6210B and 20B ROM devices.

The ST62E25 device is an erasable EPROM version of the ST62T25 device, which may be used to emulate the T15 and T25 devices, as well as the respective ST6215B and 25B ROM devices.

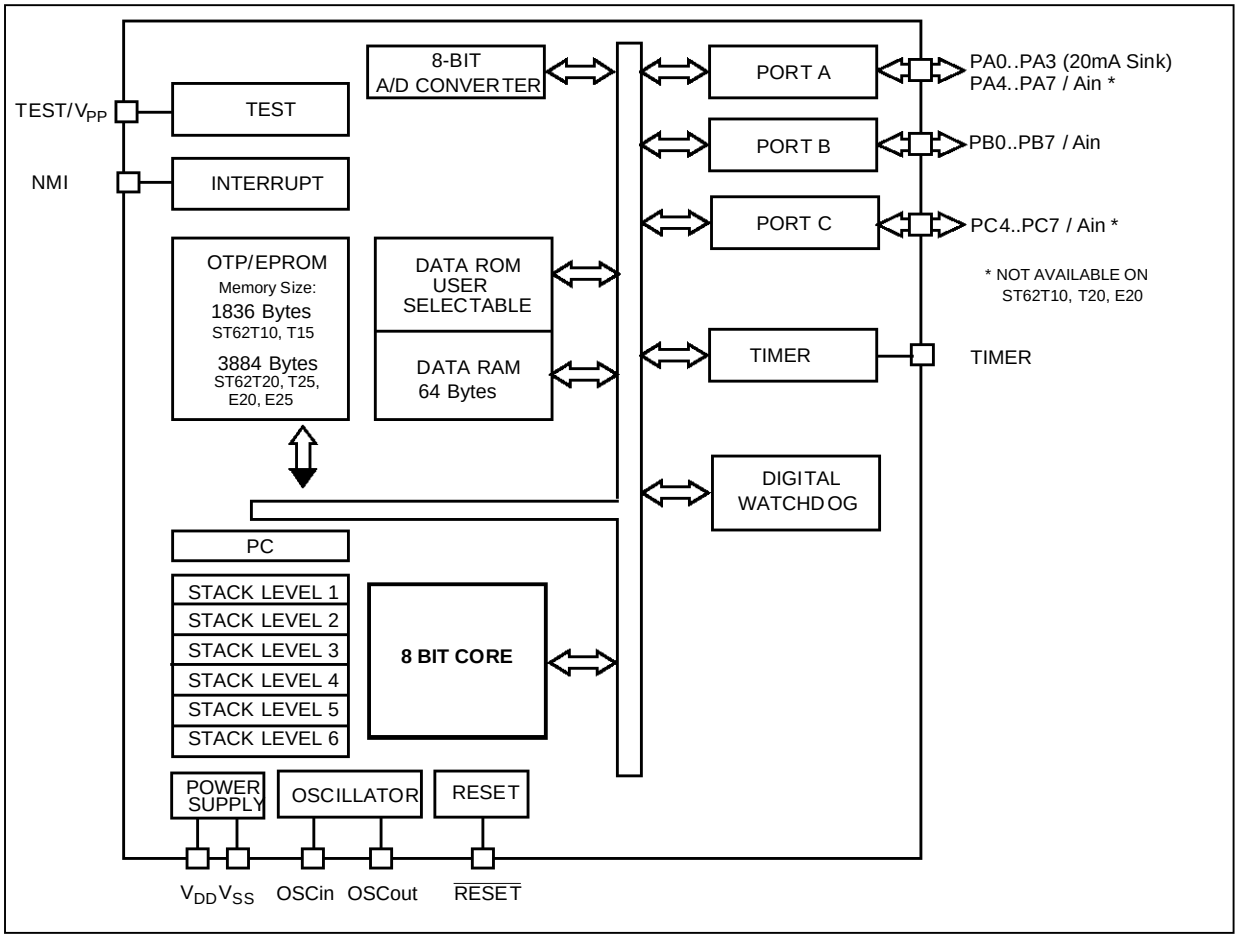
OTP and EPROM devices are functionally identical. The ROM based versions offer the following additional features: RC Oscillator, Oscillator Safeguard, external Stop mode control, program code readout protection and the possibility of having an internal pullup on the NMI and TIMER pins.

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.

EPROM devices, thanks to their ease of erasure and reprogrammability, are best suited for program development and evaluation.

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

Figure 1. Block Diagram



1.2 PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation. If TEST pin is connected to a +12.5V level during the reset phase, the EPROM programming Mode is entered.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. A pull-up device must be provided externally on OTP and EPROM devices.

TIMER. This is the timer I/O pin. In input mode it is connected to the prescaler and acts as external timer clock input or as control gate input for the internal timer clock. In output mode the timer pin outputs the data bit when a time-out occurs. A pull-up device must be provided externally on OTP and EPROM devices.

PA0-PA3,PA4-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs. PA0-PA3 can also sink 20mA for direct LED driving while PA4-PA7 can be programmed as analog inputs for the A/D converter.
Note: PA4-PA7 are not available on ST62T10, T20 or E20.

PB0-PB7. These 8 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs, or as analog inputs for the A/D converter.

PC4-PC7. These 4 lines are organized as one I/O port (C). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs, or as analog inputs for the A/D converter.
Note: PC4-PC7 are not available on ST62T10, T20 or E20.

Figure 2. ST62T10, T20, E20 Pin Configuration

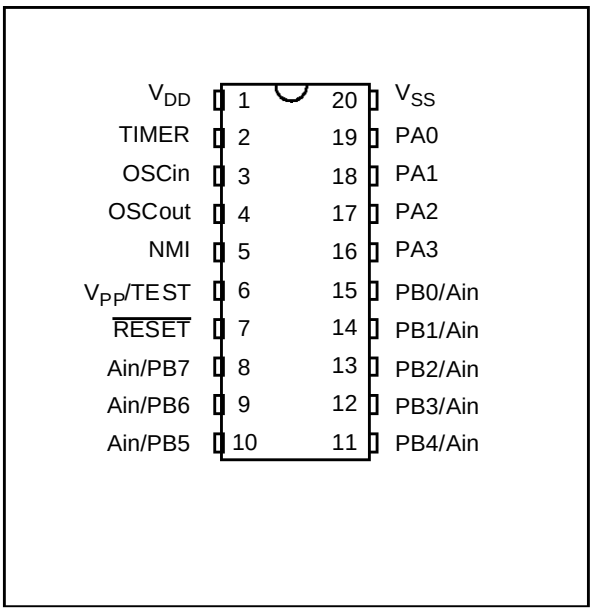
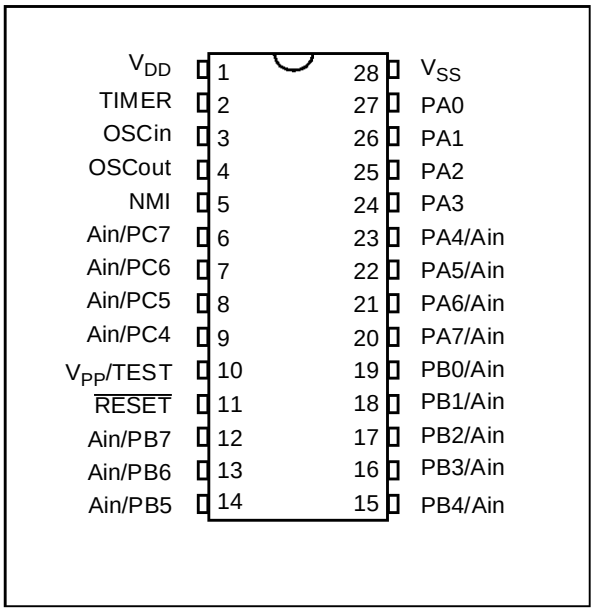


Figure 3. ST62T15, T25, E25 Pin Configuration



1.3 MEMORY MAPS

1.3.1 Program Memory Maps

Figure 4. ST62T10, T15 Program Memory Map

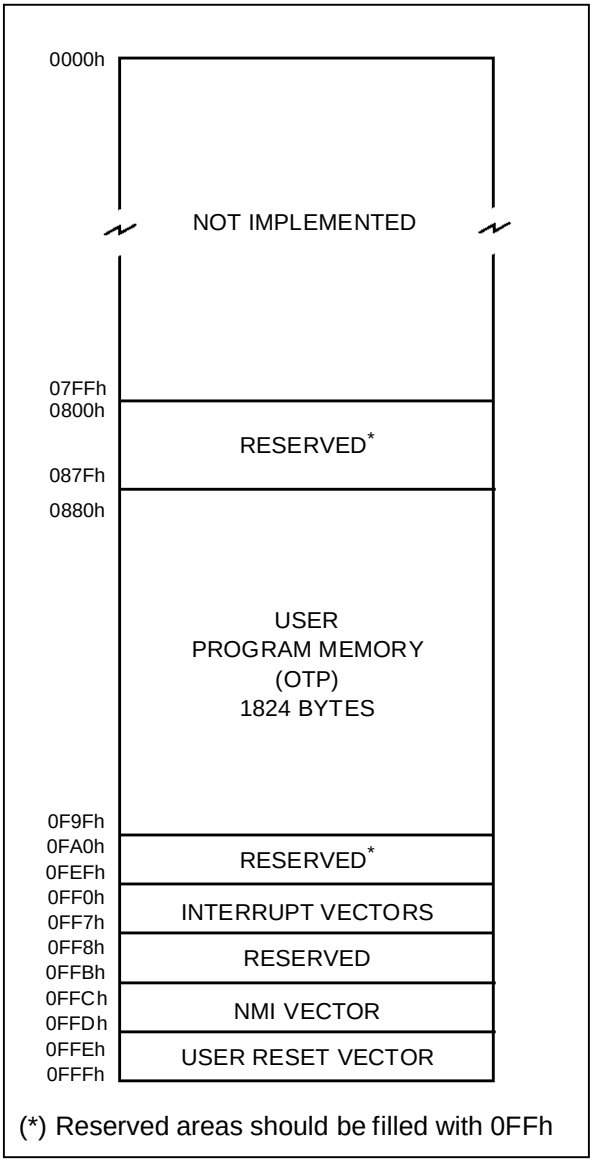
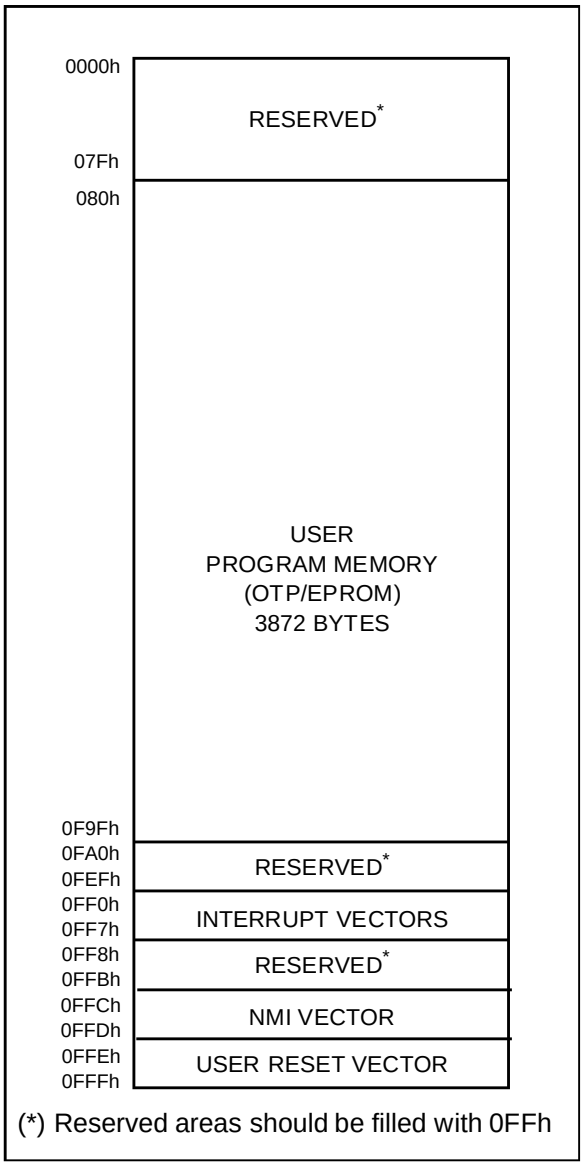


Figure 5. ST62T20, T25, E20, E25 Program Memory Map



1.3.2 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.

The Data Space is fully described and illustrated on page 21.

1.4 PARTICULARITIES OF OTP AND EPROM DEVICES

OTP and EPROM devices are identical save for the package which, in the EPROM device, is fitted with a transparent window to allow erasure of memory contents by exposure to UV light.

Both OTP and EPROM parts may be programmed using programming equipment approved by SGS-THOMSON.

1.4.1 OTP/EPROM Programming

Programming mode is selected by applying a 12.5V voltage to the V_{PP}/TEST pin during reset. Programming of OTP and EPROM parts is fully described in the EPROM Programming Board User Manual.

1.4.2 Eprom Erasure

Thanks to the transparent window present in the EPROM package, its memory contents may be erased by exposure to UV light.

Erasure begins when the device is exposed to light with a wavelength shorter than 4000Å. It should be noted that sunlight, as well as some types of artificial light, includes wavelengths in the 3000-4000Å range which, on prolonged exposure, can cause erasure of memory contents. It is thus recommended that EPROM devices be fitted with an opaque label over the window area in order to prevent unintentional erasure.

The recommended erasure procedure for EPROM devices consists of exposure to short wave UV light having a wavelength of 2537Å. The minimum recommended integrated dose (intensity x exposure time) for complete erasure is 15Wsec/cm^2. This is equivalent to an erasure time of 15-20 minutes using a UV source having an intensity of 12mW/cm^2 at a distance of 25mm (1 inch) from the device window.

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

The CPU Core may be thought of as an independent central processor communicating with on-chip I/O, memory and peripherals. For further details refer to page 18.

2.2 CPU REGISTERS

The CPU Core features six registers and three pairs of flags available to the programmer. For a detailed description refer to page 24.

3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The Oscillator may be driven by an external clock, or by a crystal or ceramic resonator. ROM devices also offer RC oscillator and Oscillator Safeguard features. For a complete description refer to page 26.

3.2 RESETS

The MCU can be reset in three ways: by the external Reset input being pulled low, by the Power-on Reset circuit, or by the Digital Watchdog timing out. For further details refer to page 30.

3.3 DIGITAL WATCHDOG

The Digital Watchdog can be used to provide controlled recovery from software upsets. Software and Hardware enabled Watchdog options are available in order to achieve optimum trade-off between power consumption and noise immunity. For a complete description and a selection guide refer to page 33.

3.4 INTERRUPTS

The CPU can manage four Maskable and one Non-Maskable Interrupt source. Each source is associated with a specific Interrupt Vector. An internal pullup option on the NMI pin is available on ROM devices. For a complete description refer to page 37.

3.5 POWER SAVING MODES

WAIT mode reduces electrical consumption during idle periods, while STOP mode achieves the lowest power consumption by stopping all CPU activity. For a complete description refer to page 41.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

Input/Output lines may be individually programmed as one of a number of different configurations. For further details refer to page 43.

4.2 TIMER

The on-chip Timer peripheral consists of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. For a complete description refer to page 47.

4.3 A/D CONVERTER (ADC)

The 8-bit on-chip ADC features multiplexed analog inputs, as alternate I/O functions. Conversion is by successive approximations, with a typical conversion time of 70us, at 8MHz oscillator frequency. For a complete description refer to page 50.

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 architecture has been designed to exploit the hardware in the most efficient way possible, while keeping byte usage to a minimum. For further details refer to page 52.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes: Immediate, Direct, Short Direct, Extended, Program Counter Relative, Bit Direct, Bit Test & Branch, Indirect, and Inherent. For a complete description of the available addressing modes, refer to page 52.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes; these may be subdivided into six types: load/store, arithmetic/logic, conditional branch, control, jump/call, and bit manipulation. For further details refer to page 53.

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices designed to protect the inputs against damage due to high static voltages; however, it is advisable to take normal precautions to avoid applying voltages higher than the specified maximum ratings.

For proper operation, it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, T_j, in degrees Celsius can be obtained from:

$$T_j = T_A + P_D \times R_{thJA}$$

Where:

T_A = Ambient Temperature.

R_{thJA} =Package thermal resistance (junction-to ambient).

$P_D = P_{int} + P_{port}$

$P_{int} = I_{DD} \times V_{DD}$ (chip internal power).

P_{port} =Port power dissipation (to be determined by the user)

.

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_O	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3(1)	V
V_{PP}	OTP/EPROM Programming Voltage	13	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into V_{DD} (source)	50(2)	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

Stresses above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

(1) Within these limits, clamping diodes are non-conducting. Voltages outside these limits are authorised provided injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. The total current through port C combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 100mA.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	
		PSO20			80	
		PDIP28			55	
		PSO28			75	

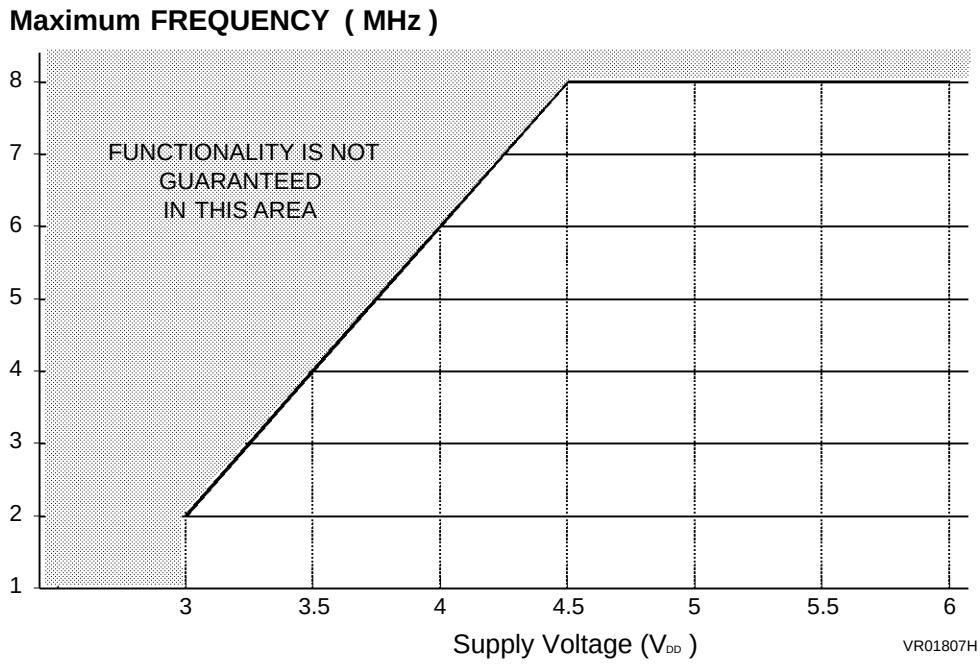
6.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

If a total current of +1mA is flowing into a single analog channel, or if the total current flowing into all the analog inputs is 1mA, all resulting A/D conversions will be shifted by + 1 LSB. If a total positive current is flowing into a single analog channel, or if the total current flowing into all analog inputs is 5mA, all the resulting conversions are shifted by + 2 LSB.

Figure 6. MAXIMUM OPERATING FREQUENCY (F_{MAX}) VERSUS SUPPLY VOLTAGE (V_{DD})

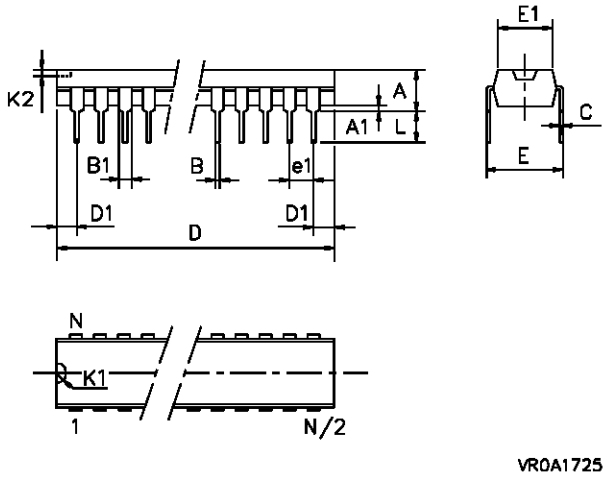


Note: The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

7 GENERAL INFORMATION

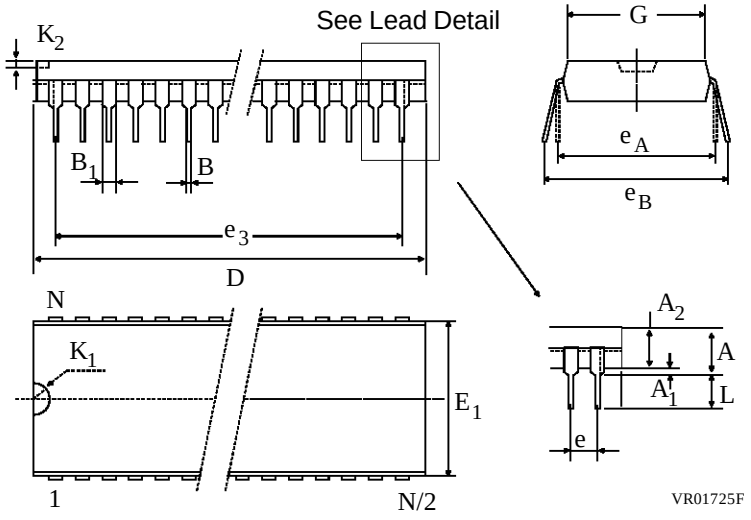
7.1 PACKAGE MECHANICAL DATA

Figure 1. 20-Pin Plastic Dual In-Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
A2	-	-	-	-	-	-
B		0.45			0.018	
B1	1.39		1.65	0.055		0.065
C		0.25			0.010	
D			25.4			1.000
e3		22.86			0.900	
E1			7.1			0.280
e		2.54			0.100	
eB		8.5			0.335	
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.3			0.130	
	Number of Pins					
N	20					

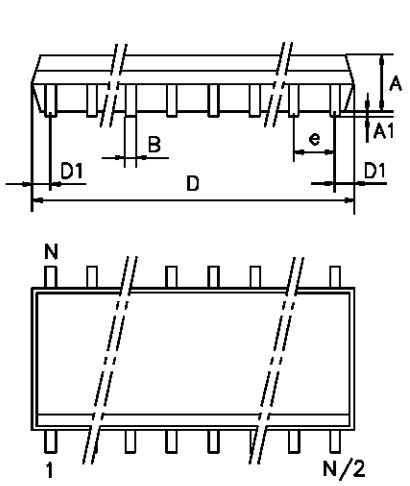
Figure 7. 28-Pin Plastic Dual In-Line Package, 600-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		4.445			0.175	
A1		0.63			0.025	
A2	-	-	-	-	-	-
B		0.45			0.018	
B1		1.27			0.050	
C	0.23		0.310	0.009		0.012
D			37.34			1.470
e3		33.02			1.300	
E1			14.10			0.555
e		2.54			0.100	
eA						
eB		15.20	16.68		0.598	0.657
G						
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.30			0.130	
	Number of Pins					
N	28					

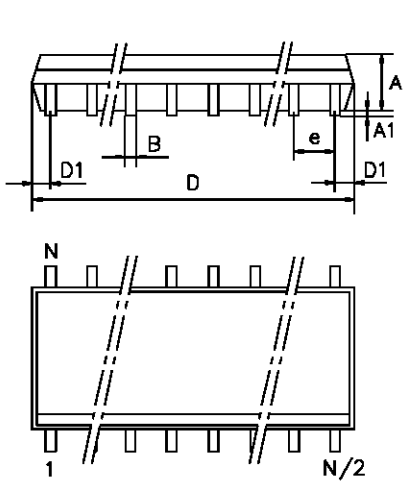
PACKAGE MECHANICAL DATA (Cont'd)

Figure 8. 20-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

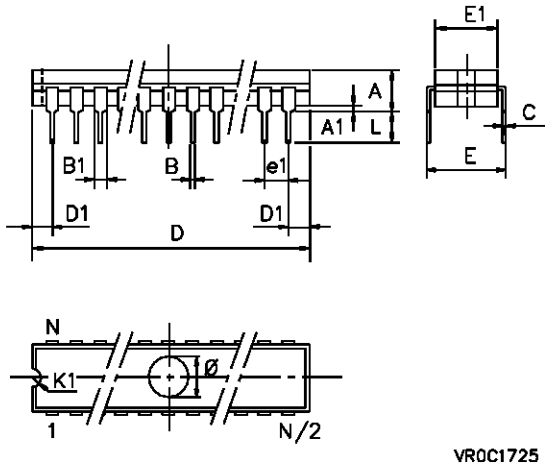
Figure 9. 28-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	17.70		18.10	0.697		0.713
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	28					

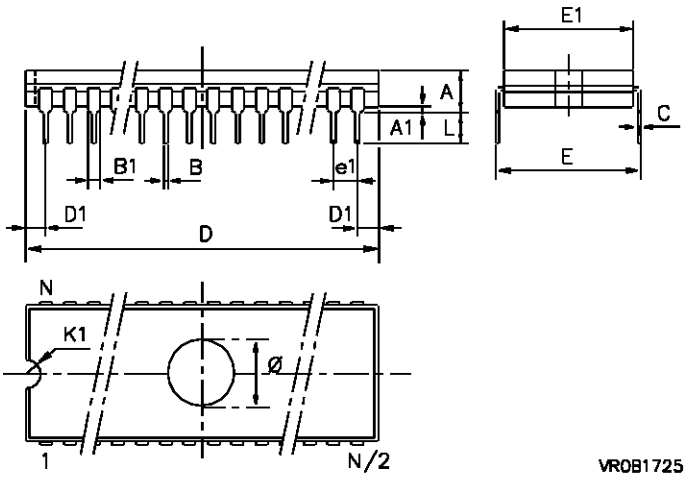
PACKAGE MECHANICAL DATA (Cont'd)

Figure 10. 20-Pin Ceramic Dual In Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.71			0.225
A1	0.50		1.78	0.020		0.070
B	0.40		0.55	0.016		0.022
C	0.22		0.31	0.009		0.012
D			26.92			1.059
D1	0.51		1.27	0.020		0.050
E	-	-	-	-	-	-
E1			7.80			0.307
L	3.40			0.134		
e1	2.29		2.79	0.090		0.110
∅	4.24		4.39	0.167		0.173
	Number of Pins					
N	20					

Figure 11. 28-Ceramic Dual In Line Package, 600-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.71			0.225
A1	0.50		1.78	0.020		0.070
B	0.40		0.55	0.016		0.022
B1	1.17		1.42	0.046		0.056
C	0.22		0.31	0.009		0.012
D			38.10			1.50
D1	1.52		2.49	0.060		0.098
E	-	-	-	-	-	-
E1	13.05		13.36	0.514		0.526
L	3.00			0.118		
e1	2.29		2.79	0.090		0.110
∅	6.86		7.36	0.270		0.290
	Number of Pins					
N	28					

7.2 ORDERING INFORMATION

Table 1. OTP Device Sales Types

Sales Type	OTP (Bytes)	I/O Pins	Option	Temperature range	Package
ST62T10B6/HWD	1836	12	Hardware Watchdog	-40°C TO +85°C	PDIP20
ST62T10B6/SWD			Software Watchdog		
ST62T20B6/HWD	3884		Hardware Watchdog		
ST62T20B6/SWD			Software Watchdog		
ST62T10M6/HWD	1836		Hardware Watchdog		PSO20
ST62T10M6/SWD			Software Watchdog		
ST62T20M6/HWD	3884		Hardware Watchdog		
ST62T20M6/SWD			Software Watchdog		
ST62T15B6/HWD	1836	20	Hardware Watchdog		PDIP28
ST62T15B6/SWD			Software Watchdog		
ST62T25B6/HWD	3884		Hardware Watchdog		
ST62T25B6/SWD			Software Watchdog		
ST62T15M6/HWD	1836		Hardware Watchdog		PSO28
ST62T15M6/SWD			Software Watchdog		
ST62T25M6/HWD	3884		Hardware Watchdog		
ST62T25M6/SWD			Software Watchdog		

Table 2. EPROM Device Sales Types

Sales Type	EPROM (Bytes)	I/O Pins	Option	Temperature range	Package
ST62E20F1/HWD	3884	12	Hardware Watchdog	0 TO +70°C	CDIP20W
ST62E20F1/SWD			Software Watchdog		
ST62E25F1/HWD		20	Hardware Watchdog		CDIP28W
ST62E25F1/SWD			Software Watchdog		

Notes

SGS-THOMSON MICROELECTRONICS

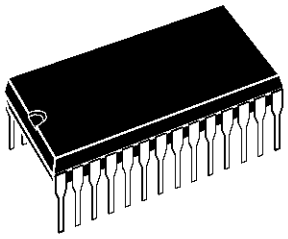
ST6210B, 15B, 20B, 25B

8-BIT HCMOS MCU WITH A/D CONVERTER

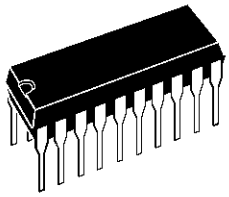
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in ROM
- Data ROM: User selectable size (in program ROM)
- Data RAM: 64 bytes
- ROM read-out Protection
- 12/20 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 4 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- Digital Watchdog
- Oscillator Safe Guard
- 8-bit A/D Converter with 8 (ST6210B, ST6220B) and 16 (ST6215B, ST6225B) analog inputs
- On-chip Clock oscillator can be driven by Quartz crystal, Ceramic resonator or RC network
- Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).

DEVICE SUMMARY

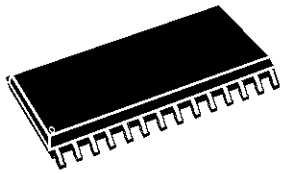
DEVICE	ROM (Bytes)	I/O Pins
ST6210B	1836	12
ST6220B	3884	
ST6215B	1836	20
ST6225B	3884	



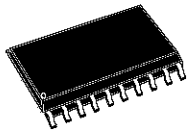
PDIP28



PDIP20



PSO28



PSO20

(See end of Datasheet for Ordering Information)

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

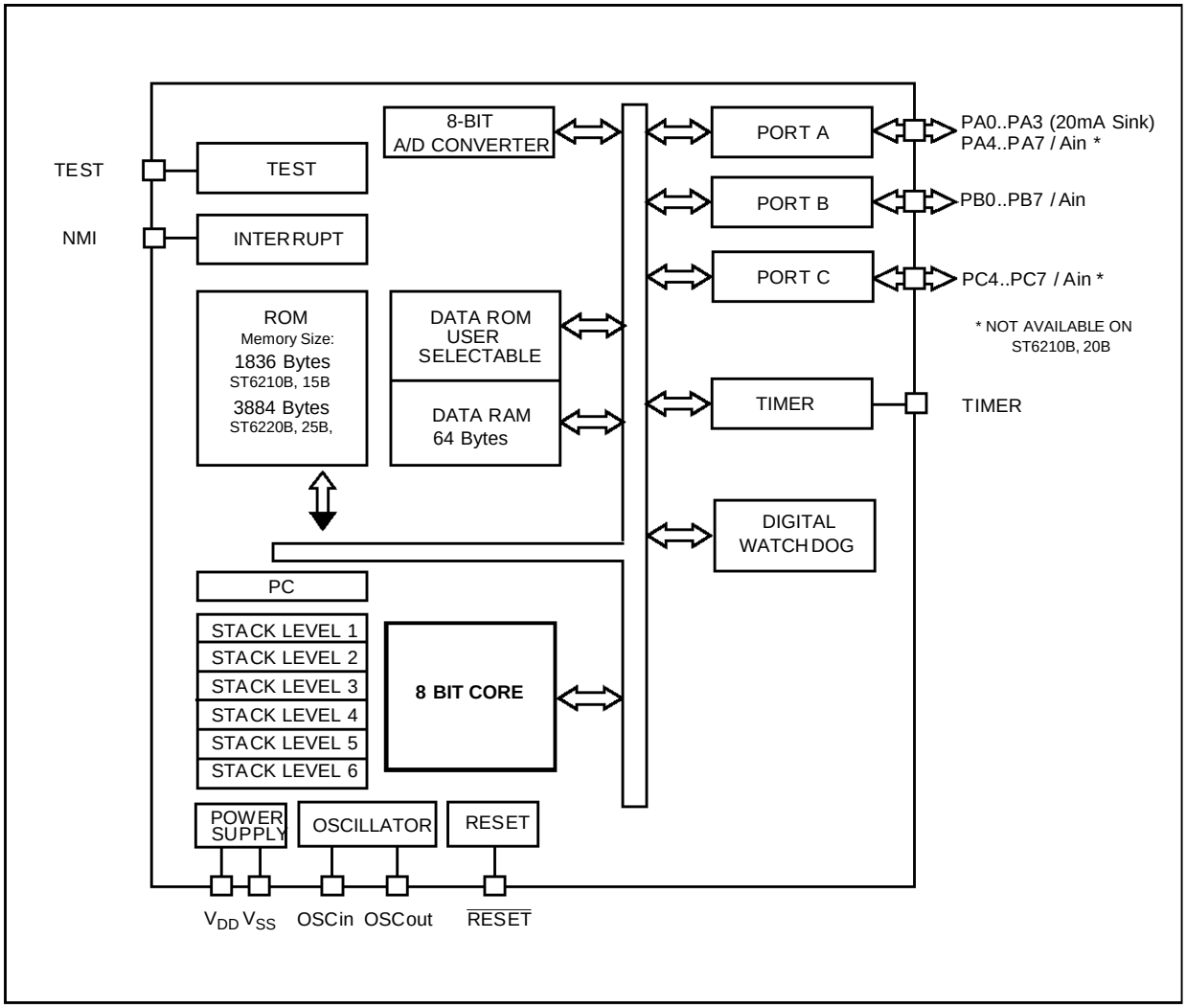
The ST6210B, ST6215B, ST6220B and ST6225B microcontrollers are members of the 8-bit HCMOS ST62xx family of devices, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST6210B, ST6215B, ST6220B and ST6225B devices feature the following peripherals: a Timer comprising an 8-bit counter equipped with a 7-bit software programmable prescaler, an 8-bit A/D Converter with up to 16 analog inputs (as I/O pin alternate functions), and a Digital Watchdog timer.

The ST6210B, ST6215B, ST6220B and ST6225B devices feature various options such as a choice of Quartz, Ceramic or RC oscillators, an Oscillator Safe Guard circuit, Read-out Protection against unauthorised copying of program code, and an External STOP Mode Control to offer optimum tradeoff between power consumption and noise immunity, depending on the application.

These devices are well suited for automotive, appliance and industrial applications. The user programmable parts for program development are the ST62E20 and E25 which are pin compatible devices with 4Kbytes of EPROM.

Figure 2. Block Diagram



1.2 PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. When the QUARTZ/CERAMIC RESONATOR Mask Option is selected, a quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. When the RC OSCILLATOR Mask Option is selected, a resistor must be connected between the OSCout pin and ground. The OSCin pin is the input pin, the OSCout pin is the output pin.

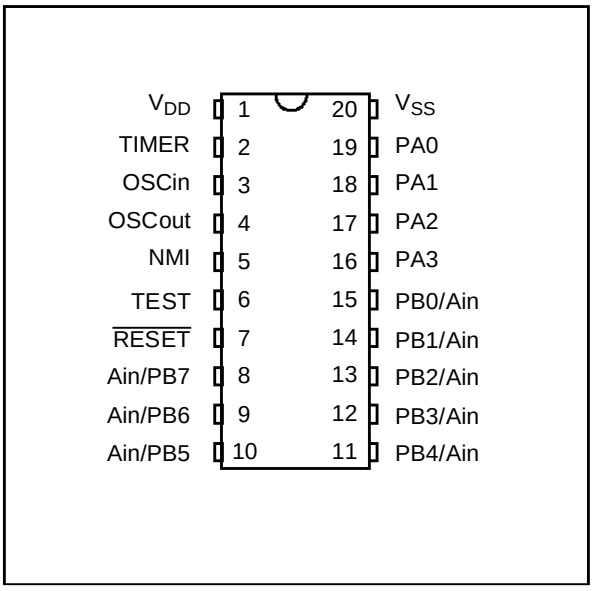
$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST. The TEST pin must be held at V_{SS} for normal operation (an internal 100kΩ pull-down resistor selects normal operating mode if the TEST pin is not connected externally).

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI is falling edge sensitive. A ROM mask option makes available an on-chip pull-up on the NMI pin.

TIMER. This is the timer I/O pin. In input mode it is connected to the prescaler and acts as external timer clock input or as control gate input for the internal timer clock. In output mode the timer pin outputs the data bit when a time-out occurs. A ROM mask option makes available an on-chip pull-up on the TIMER pin.

Figure 3. ST6210B, ST6220B Pin Configuration

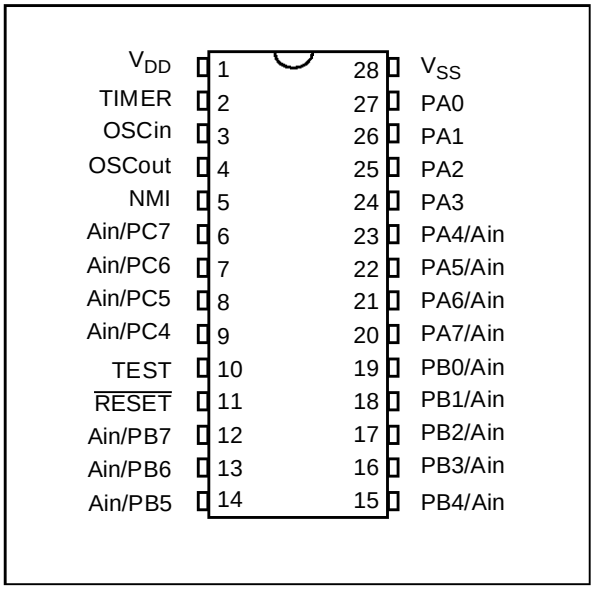


PA0-PA3,PA4-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs. PA0-PA3 can also sink 20mA for direct LED driving while PA4-PA7 can be programmed as analog inputs for the A/D converter.
Note: PA4-PA7 are not available on ST6210B, ST6220B.

PB0-PB7. These 8 lines are organized as one I/O port (B). When the External STOP Mode Control option is disabled, each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs and as analog inputs for the A/D converter. When the External STOP Mode Control option is enabled, PB0 output Mode is forced as open-drain (push-pull output is not possible). The other lines are unchanged.

PC4-PC7. These 4 lines are organized as one I/O port (C). When the External STOP Mode Control option is disabled, each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs and as analog inputs for the A/D converter. When the External STOP Mode Control is enabled PC7 output Mode is forced as open-drain (push-pull output is not possible). The other lines are unchanged.
Note: PC4-PC7 are not available on ST6210B, ST6220B.

Figure 4. ST6215B, ST6225B Pin Configuration



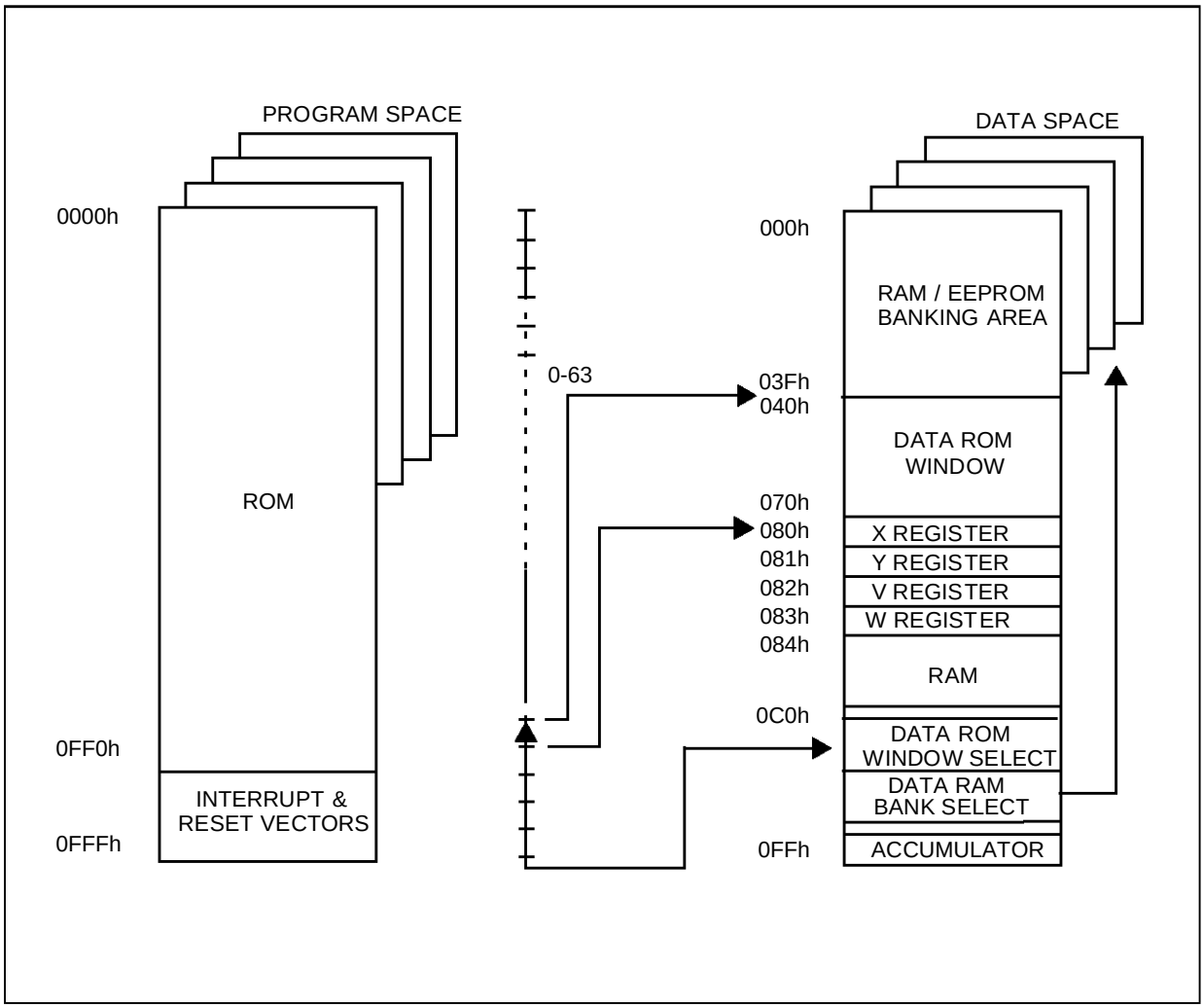
1.3 MEMORY MAP

1.3.1 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 ROM and user vectors; Data space contains user data in RAM and in ROM, and Stack space accommodates six levels of stack for subroutine and interrupt service routine nesting.

Figure 5. Memory Addressing Diagram



MEMORY MAP (Cont'd)

1.3.2 Program Space

Program Space is physically implemented in ROM memory. It 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)

1.3.2.1 ROM Protection

The ST6210B, ST6215B, ST6220B and ST6225B Program Space can be protected against external read-out of ROM contents when the READOUT PROTECTION mask option is chosen. This option allows the user to blow a dedicated fuse on the silicon, by applying a high voltage at V_{PP} (see detailed information in the "Electrical Specification").

Note: Once the Read-out Protection fuse is blown, it is no longer possible, even for SGS-THOMSON, to gain access to the ROM contents. Returned parts with a blown fuse can therefore not be accepted.

1.3.3 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 ROM.

Figure 6. ST6210B/15B Program Memory Map

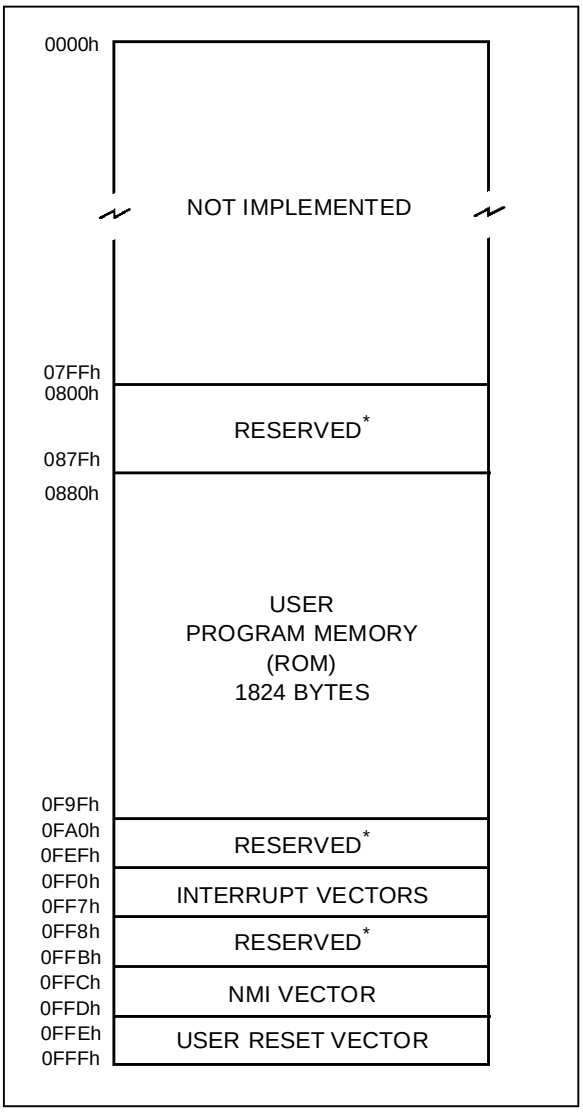
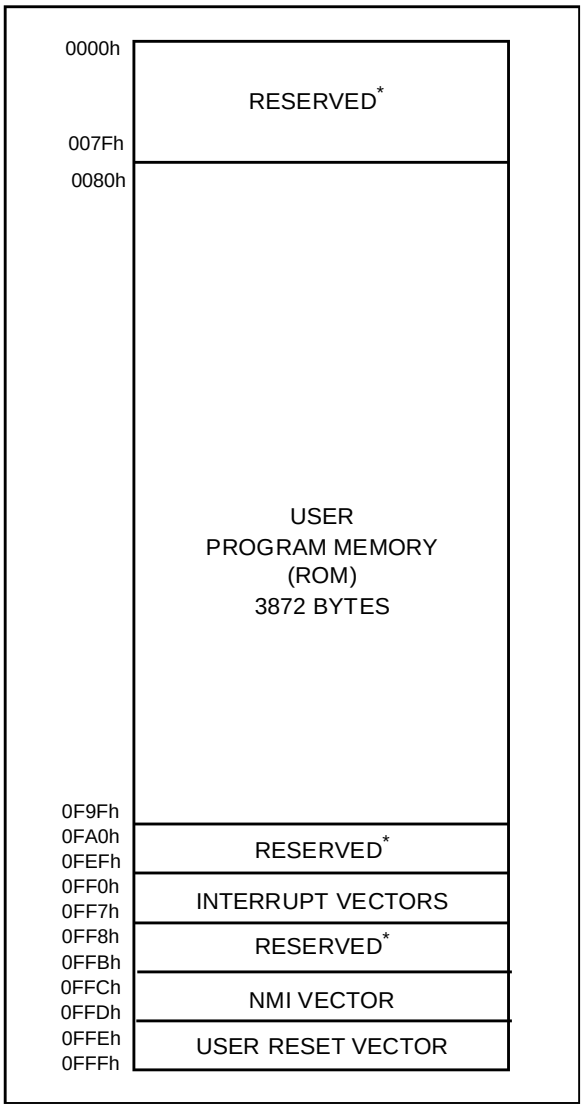


Figure 7. ST6220B/25B Program Memory Map



MEMORY MAP (Cont'd)

1.3.3.1 Data ROM

All read-only data is physically stored in ROM memory, which also accommodates the Program Space. The ROM 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 ROM.

1.3.3.2 Data RAM

In ST6210B, ST6215B, ST6220B and ST6225B devices, the data space includes 60 bytes of RAM, 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 (DRW register).

1.3.4 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.

Table 1. ST6210B, ST6215B, ST6220B and ST6225B Data Memory Space

NOT IMPLEMENTED	000h 03Fh
DATA ROM WINDOW 64 BYTES	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
PORT C OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PSC REGISTER	0D2h
TIMER DATA REGISTER	0D3h
TIMER TSCR REGISTER	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h
RESERVED	0D9h 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

1.3.5 Data Window Register (DWR)

The Data ROM window is located from address 0040h to address 007Fh in Data space. It allows direct reading of 64 consecutive bytes located anywhere in ROM memory, between address 0000h and 1FFFh (top memory address depends on the specific device). All the ROM memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the ROM memory by writing the appropriate code in the Write-only Data Window register (DWR register, location 00C9h).

The DWR register can be addressed like any RAM location in the Data Space at address 00C9h, it is however a write-only register and cannot be accessed using single-bit operations. This register is used to move the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) up and down the ROM memory of the MCU in steps of 64 bytes. The effective address of the byte to be read as data in ROM memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits, see Figure 8). So when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in ROM is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data ROM window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

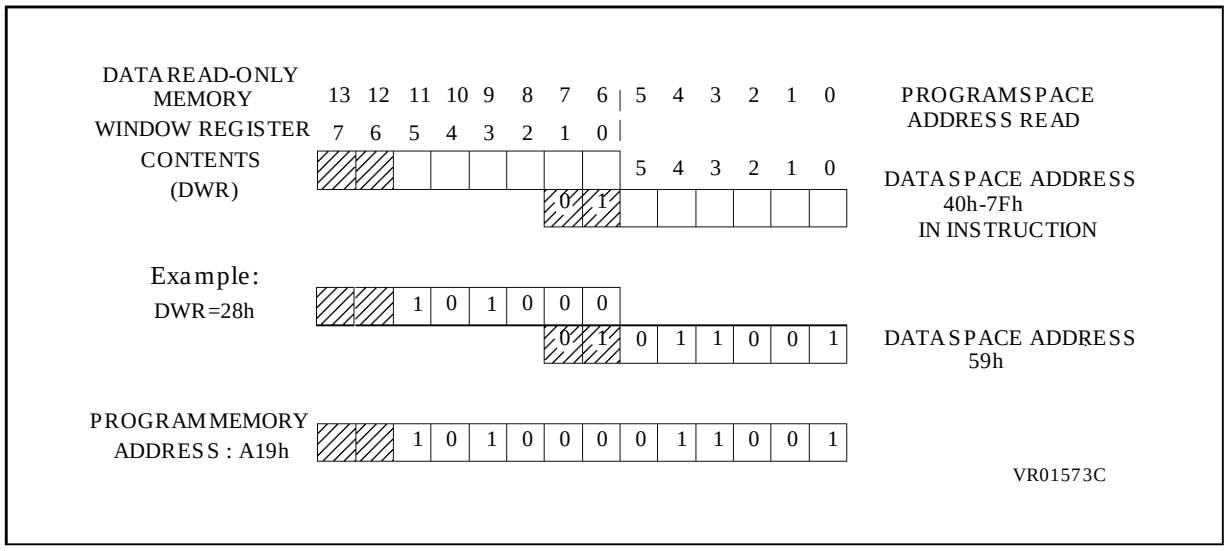
Bit 7 = This bit is not used.

Bit 6-0 = **DWR6-DWR0:** *Data ROM Window Register Bits.* These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

Caution: *This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.*

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

Figure 8. Data ROM Window Memory Addressing



2 CENTRAL PROCESSING UNIT

2.1 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. In-core communication is arranged as shown in Figure 9; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

2.2 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.

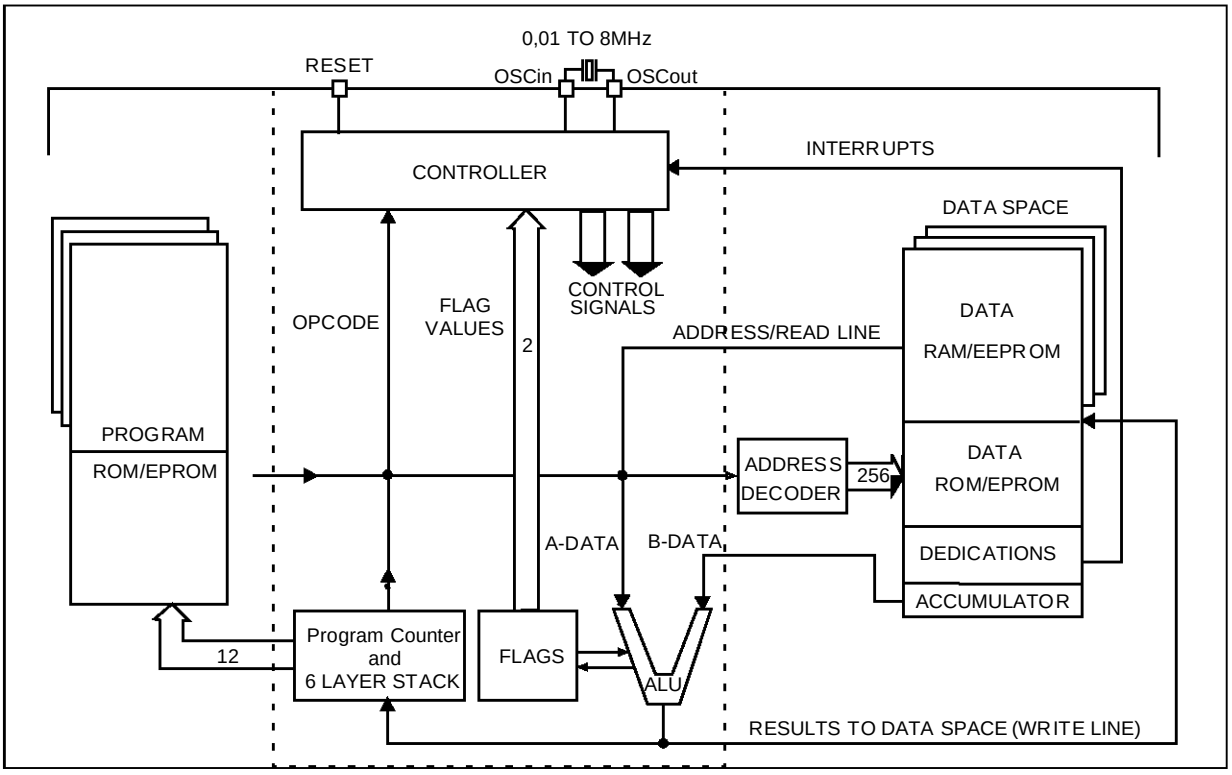
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 9. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

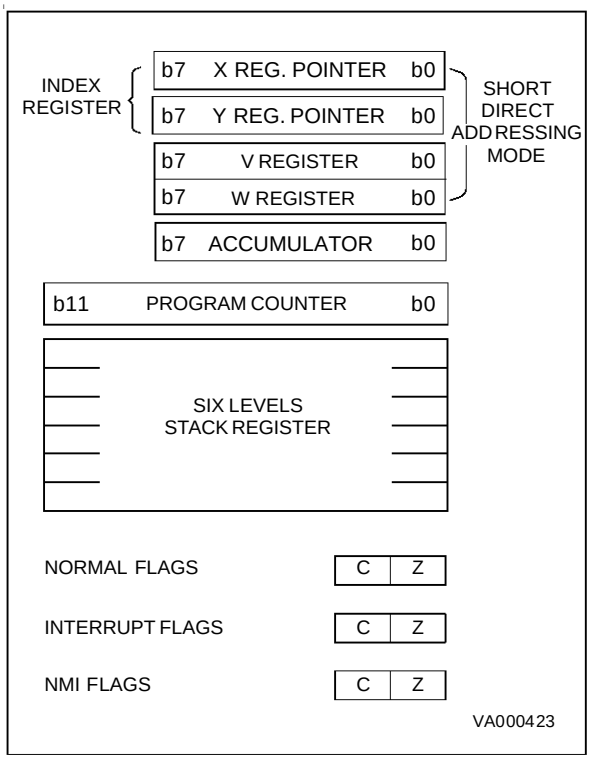
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 10. ST6 CPU Programming Mode



3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}). In addition, a Low Frequency Auxiliary Oscillator (LFAO) can be switched in for security reasons, to reduce power consumption, or to offer the benefits of a back-up clock system.

The Oscillator Safeguard (OSG) option filters spikes from the oscillator lines, provides access to the LFAO to provide a backup oscillator in the event of main oscillator failure and 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 12, Figure 13, Figure 14 and Figure 15.

Figure 11 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. C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 31 and Figure 32.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 14.

With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

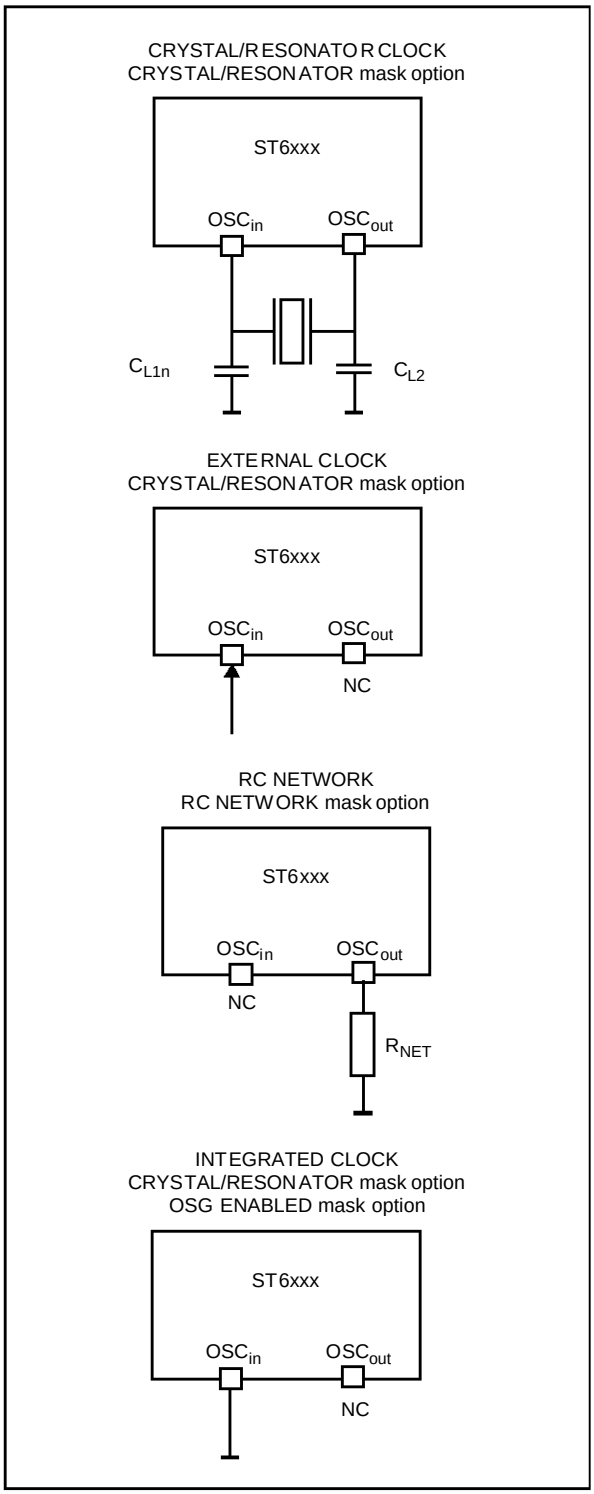
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate mask option. 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 main oscillator can be turned off (when the OSG ENABLED mask option is selected) by setting the OSCOFF bit of the ADC Control Register. The Low Frequency Auxiliary Oscillator is automatically started.

Figure 11. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Turning on the main oscillator is achieved by resetting the OSCOFF bit of the A/D Converter Control Register or by resetting the MCU. Restarting the main oscillator implies a delay comprising the oscillator start up delay period plus the duration of the software instruction at f_{LFAO} clock frequency.

3.1.2 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 safety oscillator in case of main oscillator failure.

This oscillator is available when the OSG ENABLED mask option is selected. In this case, it automatically starts one of its periods after the first missing edge from the main oscillator, whatever the reason (main oscillator defective, no clock circuitry provided, main oscillator switched off...).

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 1MHz.

At power on, the Low Frequency Auxiliary Oscillator starts faster than the Main Oscillator. It therefore feeds the on-chip counter generating the POR delay until the Main Oscillator runs.

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

ADCR

Address: 0D1h — Read/Write

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

Bit 7-3, 1-0= **ADCR7-ADCR3, ADCR1-ADCR0**: *ADC Control Register.* These bits are not used.

Bit 2 = **OSCOFF**. When low, this bit enables main oscillator to run. The main oscillator is switched off when OSCOFF is high.

3.1.3 Oscillator Safe Guard

The Oscillator Safe Guard (OSG) affords drastically increased operational integrity in ST62xx devices. The OSG circuit provides three basic functions: it filters spikes from the oscillator lines which would result in over frequency to the ST62 CPU; it gives access to the Low Frequency Auxiliary Oscillator (LFAO), used to ensure minimum processing in case of main oscillator failure, to offer reduced power consumption or to provide a fixed frequency low cost oscillator; finally, it automatically limits the internal clock frequency as a function of supply voltage, in order to ensure correct operation even if the power supply should drop.

The OSG is enabled or disabled by choosing the relevant OSG mask option. It may be viewed as a filter whose cross-over frequency is device dependent.

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 12). 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. This relationship is illustrated in Figure 15.

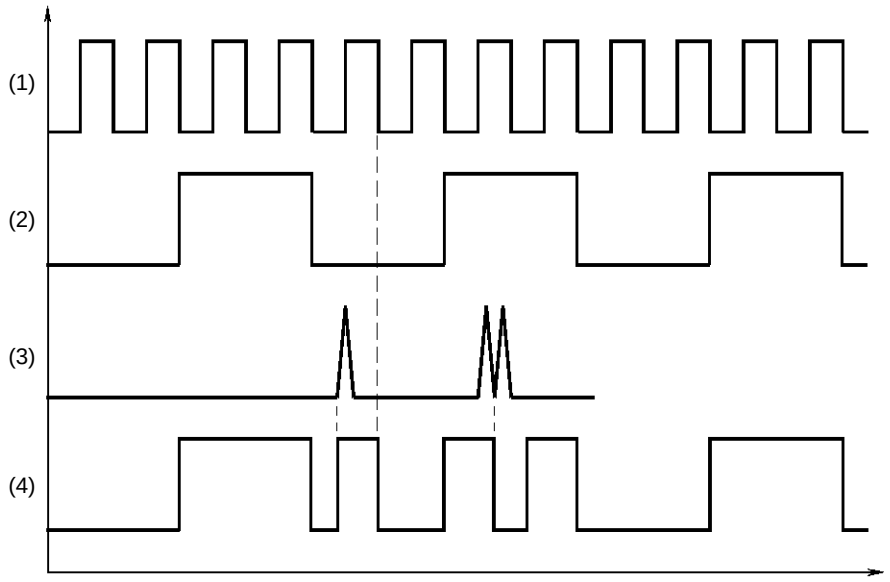
When the OSG is enabled, the Low Frequency Auxiliary Oscillator may be accessed. This oscillator starts operating after the first missing edge of the main oscillator (see Figure 13).

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.

Note. The OSG should be used wherever possible as it provides maximum safety. Care must be taken, however, as it can increase power consumption and reduce the maximum operating frequency to f_{OSG}.

CLOCK SYSTEM (Cont'd)

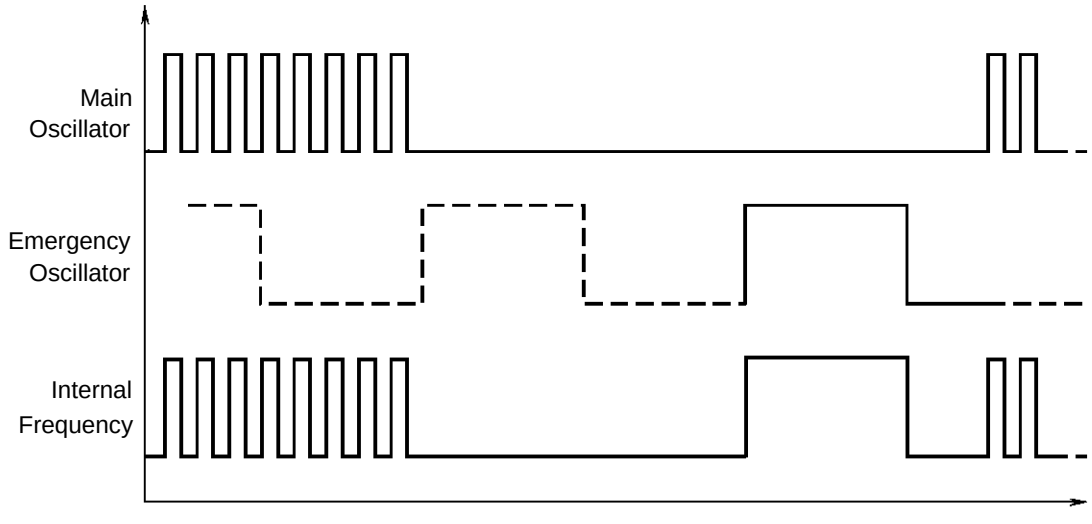
Figure 12. OSG Filtering Principle



(1) Maximum Frequency for the device to work correctly
(2) Actual Quartz Crystal Frequency at OSCin pin
(3) Noise from OSCin
(4) Resulting Internal Frequency

VR001932

Figure 13. OSG Emergency Oscillator Principle



VR001933

CLOCK SYSTEM (Cont'd)

Figure 14. Clock Circuit Block Diagram

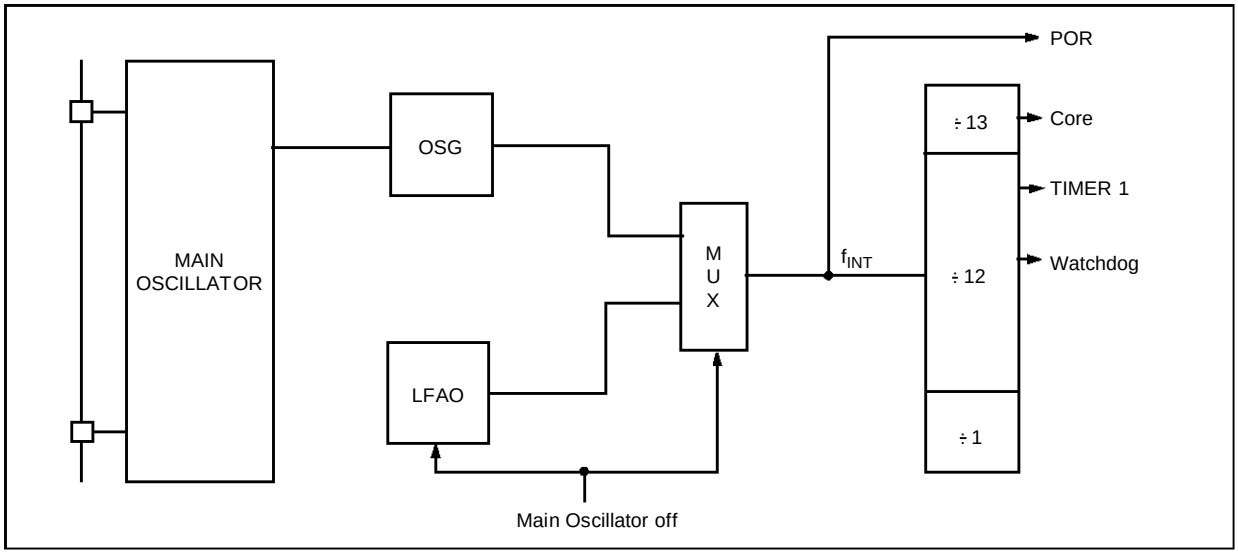
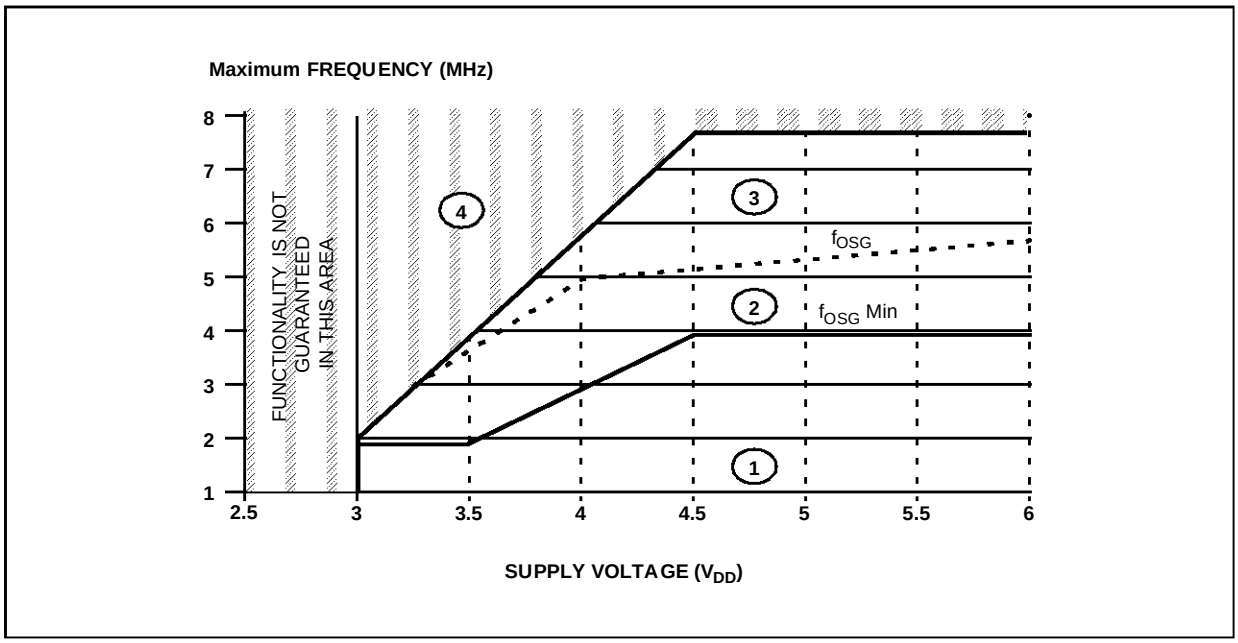


Figure 15. Maximum Operating Frequency (f_{MAX}) versus Supply Voltage (V_{DD})



Notes:

1. In this area, operation is guaranteed at the quartz crystal frequency.
2. When the OSG is disabled, operation in this area is guaranteed at the crystal frequency. When the OSG is enabled, operation in this area is guaranteed at a frequency of at least $f_{OSG\ Min.}$
3. When the OSG is disabled, operation in this area is guaranteed at the quartz crystal frequency. When the OSG is enabled, access to this area is prevented. The internal frequency is kept a f_{OSG}.
4. When the OSG is disabled, operation in this area is not guaranteed
When the OSG is enabled, access to this area is prevented. The internal frequency is kept at f_{OSG}.

3.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

3.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

3.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

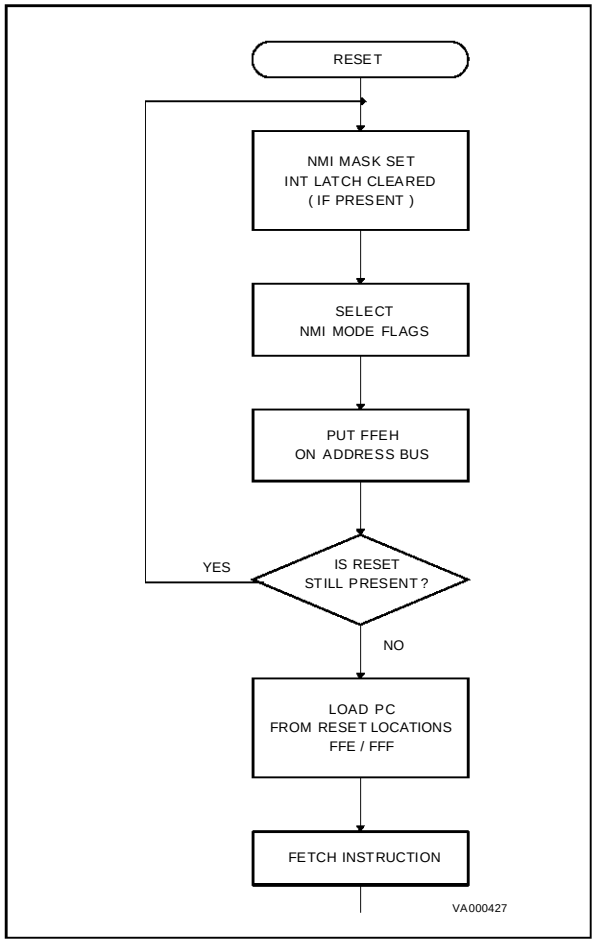
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 16. Reset and Interrupt Processing



RESETS (Cont'd)

3.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

3.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

3.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 17. Reset and Interrupt Processing

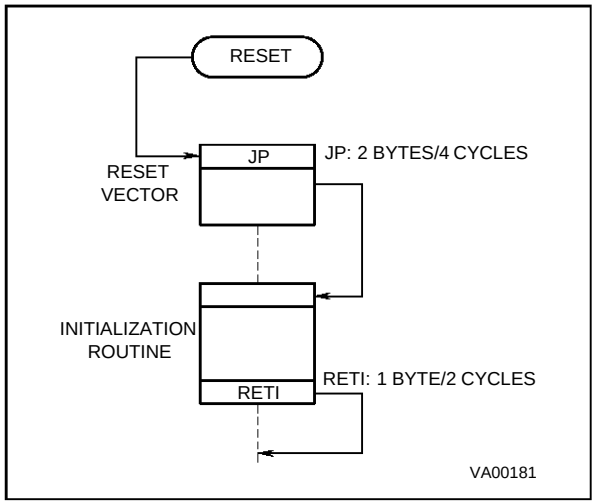
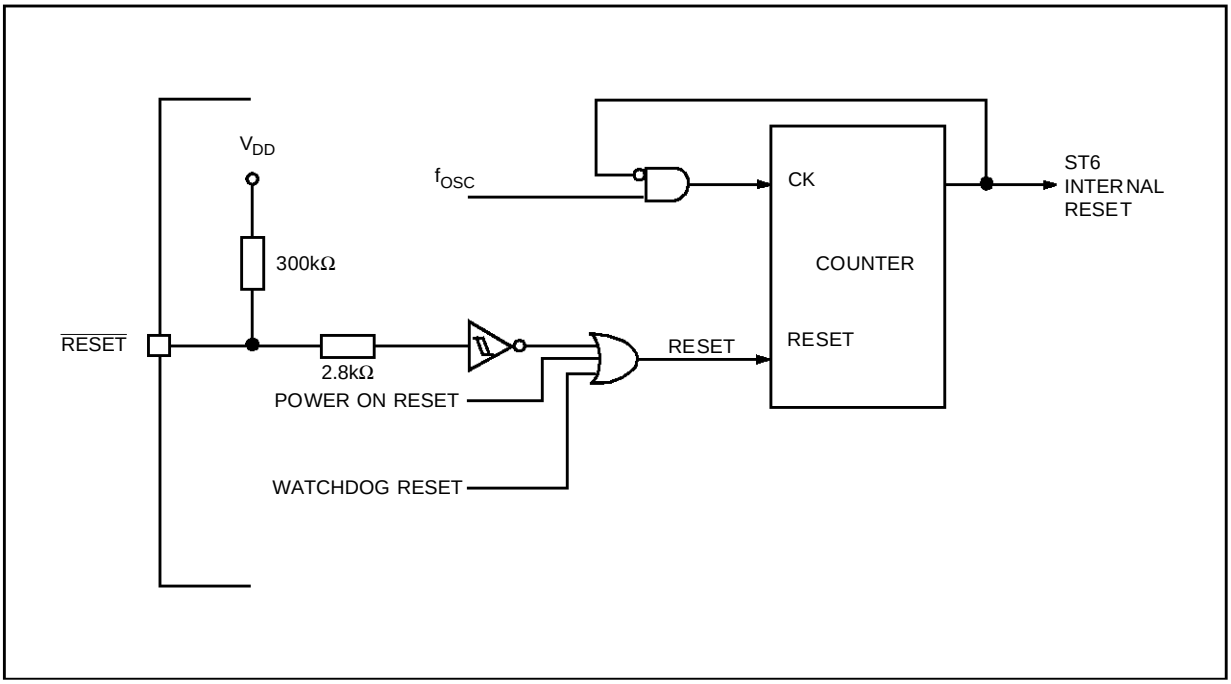


Figure 18. Reset Block Diagram



RESETS (Cont'd)

Table 2. Register Reset Status

Register	Address(es)	Status	Comment
Port Data Registers (PA, PB, PC)	0C0h to 0C2h		
Port Direction Register (PA, PB, PC)	0C4h to 0C6h		I/Os are Inputs with pull-up
Port Option Register (PA, PB, PC)	0CCh to 0CEh	00h	I/Os are Inputs with pull-up
Interrupt Option Register	0C8h		Interrupts disabled
Timer Status/Control	0D4h		Timer disabled
X, Y, V, W Register	080h to 083h		
Accumulator	0FFh		
Data RAM	084h to 0BFh	Undefined	
Data ROM Window Register	0C9h		
A/D Result Register	0D0h		
Timer Counter Register	0D3h	FFh	
Timer Prescaler Register	0D2h	7Fh	Maximum count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Stand-by, main oscillator on

3.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two mask options, known as “WATCHDOG ACTIVATION” (i.e. HARDWARE or SOFTWARE) and “EXTERNAL STOP MODE CONTROL” (see Table 3).

In the SOFTWARE mask option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, save by resetting the MCU.

In the HARDWARE mask option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL mask option (available in ROM versions only) has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL mask option has been selected, port PB0 must be defined as an open-drain output, and PA2 as an input.

Table 3. Recommended Mask Option Choices

Functions Required	Recommended Mask Options
Stop Mode & Watchdog	“EXTERNAL STOP MODE” & “HARDWARE WATCHDOG”
Stop Mode	“SOFTWARE WATCHDOG”
Watchdog	“HARDWARE WATCHDOG”

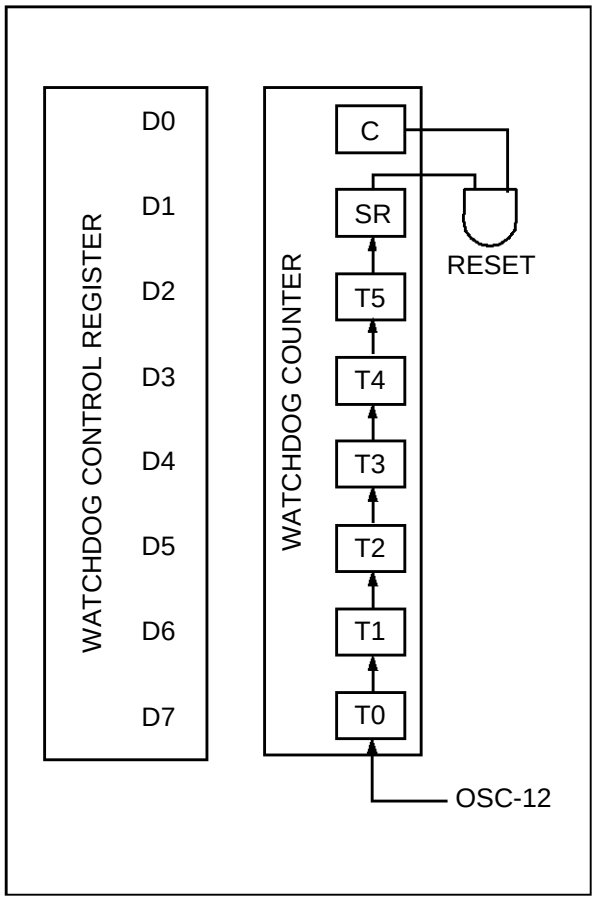
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 3.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 19.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 19. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 20) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 20. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

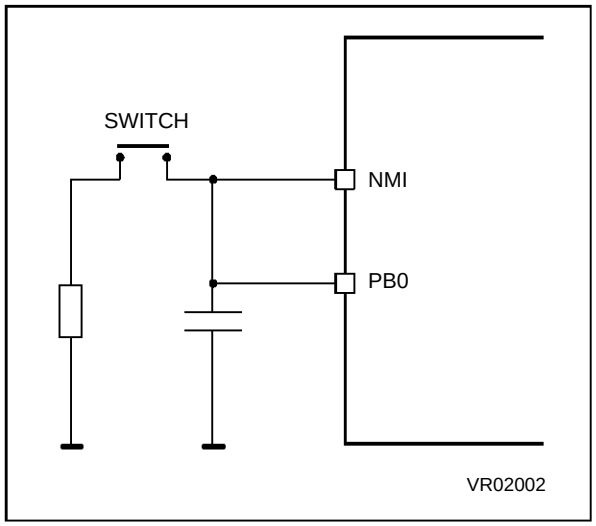
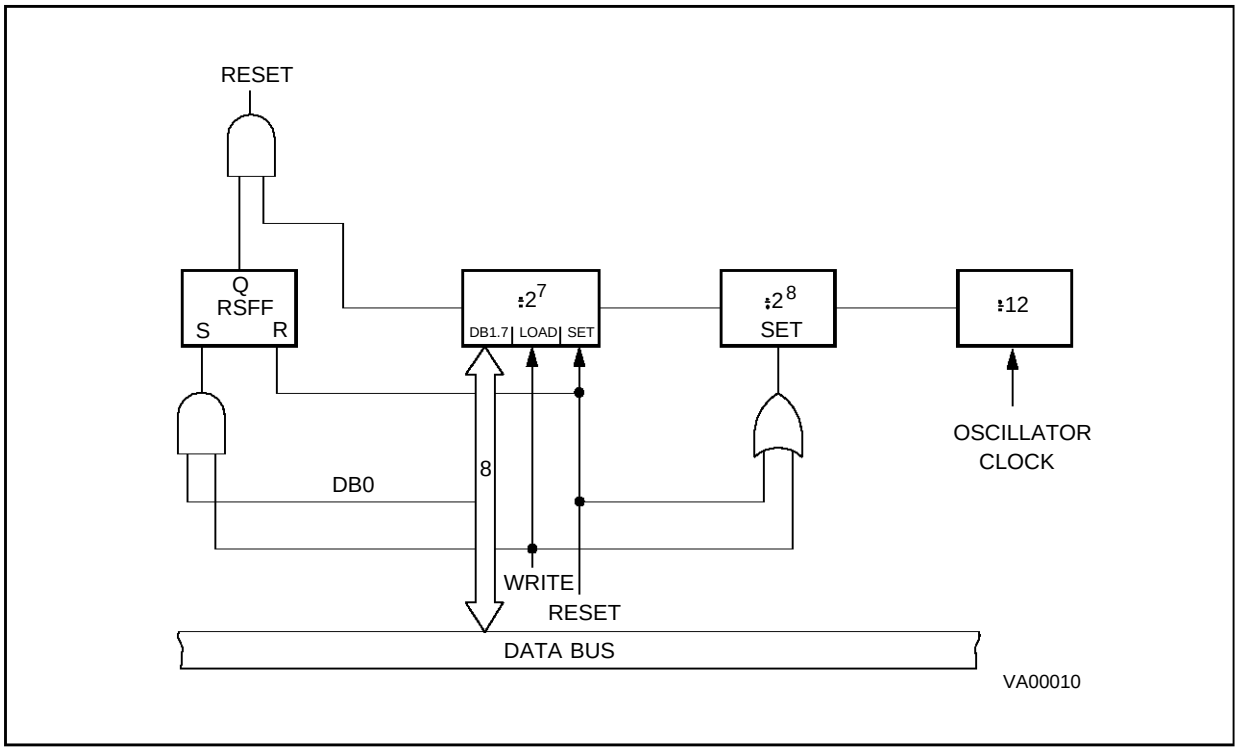


Figure 21. Digital Watchdog Block Diagram



3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 4).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 4. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A pins	Interrupt vector #1	(FF6h-FF7h)
Port B & C pins	Interrupt vector #2	(FF4h-FF5h)
TIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
ADC peripheral	Interrupt vector #4	(FF0h-FF1h)

3.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

– The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named. It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port B and C pins is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named. It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the TIMER peripheral and vector #4 with the A/D Converter peripheral.

Each on-chip peripheral has an associated interrupt request flag (TMZ for the Timer, EOC for the A/D Converter), which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (ETI for the Timer, EAI for the A/D Converter), which must be set to “1” to enable the associated interrupt request.

3.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

3.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port B and C) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 5. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A
	CLEARED	Falling edge mode on Port A
ESB	SET	Level sensitive mode on Port B & C
	CLEARED	Falling edge mode on Port B & C

3.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A-vector #1, Port B & C - vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

3.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

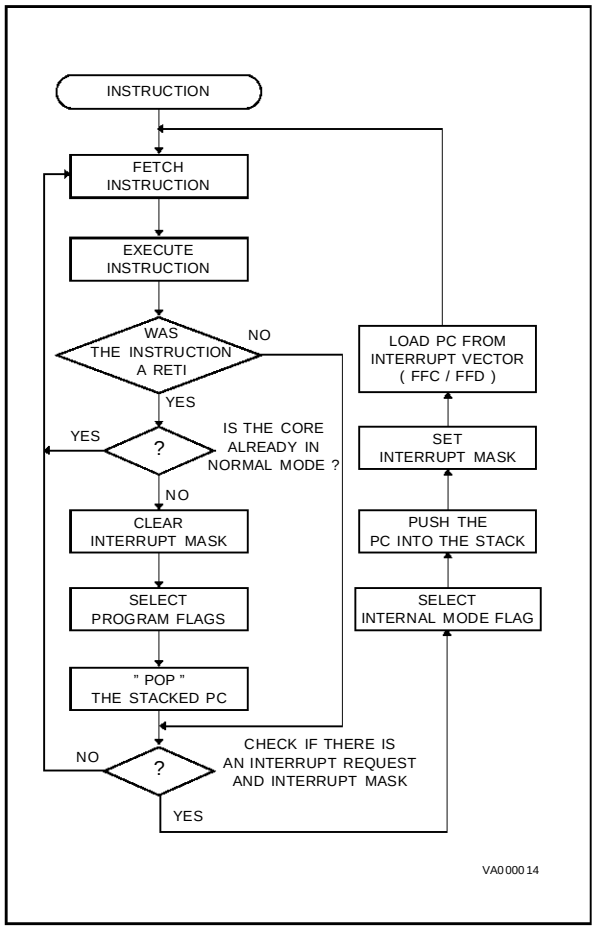
– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 22. Interrupt Processing Flow Chart

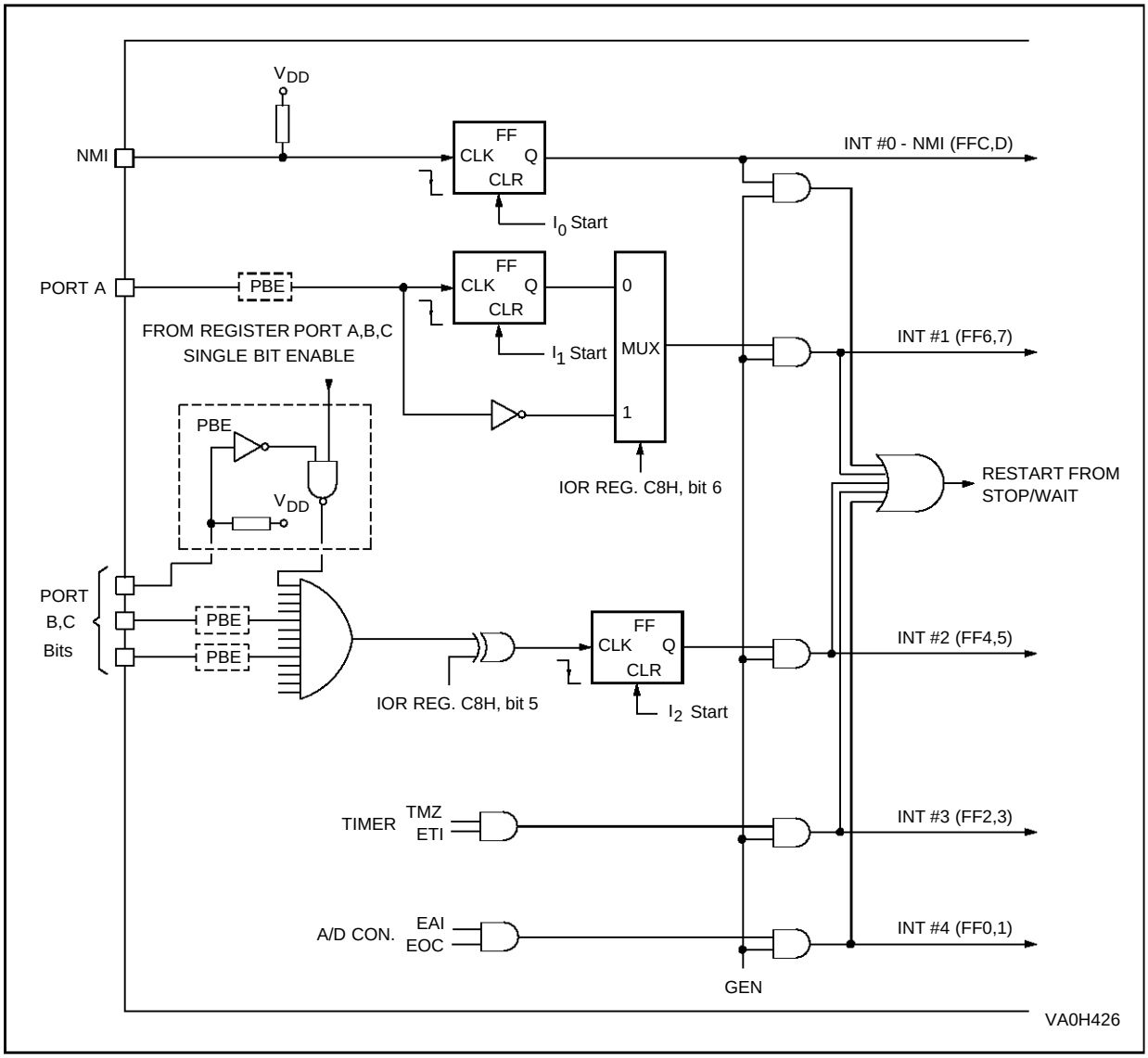


INTERRUPTS (Cont'd)

Table 6. Interrupt Requests and Mask Bits

Peripheral	Register	Address Register	Mask bit	Masked Interrupt Source	Interrupt vector
GENERAL	IOR	C8h	GEN	All Interrupts, excluding NMI	
TIMER	TSCR	D4h	ETI	TMZ: TIMER Overflow	Vector 3
A/D CONVERTER	ADCR	D1h	EAI	EOC: End of Conversion	Vector 4
Port PAn	ORA-DDRA	C4h-CCh	ORAn-DDRAn	PAn pin	Vector 1
Port PBn	ORB-DDRB	C5h-CDh	ORBn-DDRBn	PBn pin	Vector 2
Port PCn	ORC-DDRC	C6h-CEh	ORCn-DDRCn	PCn pin	Vector 2

Figure 23. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

In addition, the Low Frequency Auxiliary Oscillator (LFAO) can be used instead of the main oscillator to reduce power consumption in RUN and WAIT modes.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator (main oscillator or LFAO) is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the power consumption has to be further reduced, the Low Frequency Auxiliary Oscillator (LFAO) can be used in place of the main oscillator, if its operating frequency is lower. If required, the LFAO must be switched on before entering the WAIT mode.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

3.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection, consequently, when the LFAO is used, the user program must manage oscillator selection as soon as normal RUN mode is resumed.

3.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

3.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

3.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

– selecting the Low Frequency Auxiliary Oscillator (provided this runs at a lower frequency than the main oscillator).

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

The 20-pin devices feature 12 Input/Output lines and the 28 pin devices feature 20 Input/Output lines, (refer to the Block Diagram in Figure 2), which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Analog input (only on certain pins, see Figure 2)

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PA0-PA3 only)

The lines are organized as three Ports (A, B and C).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The three DATA registers (DRA, DRB and DRC), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

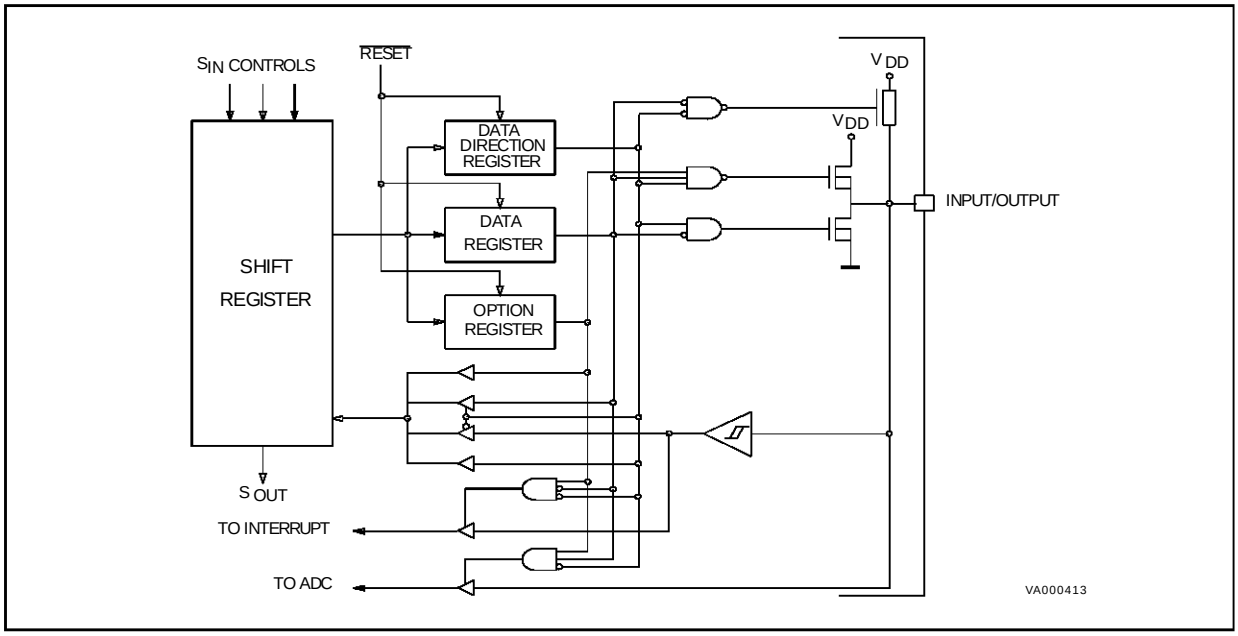
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The three Data Direction registers (DDRA, DDRB and DDRC) allow the data direction (input or output) of each pin to be set.

The three Option registers (ORA, ORB and ORC) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 24. I/O Port Block Diagram



I/O PORTS (Cont'd)

4.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 and PC7 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 7 illustrates the various port configurations which can be selected by user software.

4.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

4.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A are AND-connected to the interrupt associated with Vector #1. The pins of Port B and C are AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

4.1.1.3 Analog Input Options

Some pins (refer to the Block Diagram, Figure 2), can be configured as analog inputs by programming the OR and DR registers accordingly. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

4.1.2 I/O Port Option Registers

ORA/B/C (CCh PA, CDh PB, CEh PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Option Register bits.*

4.1.3 I/O Port Data Direction Registers

DDRA/B/C (C4h PA, C5h PB, C6h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Direction Registers bits.*

4.1.4 I/O Port Data Registers

DRA/B/C (C0h PA, C1h PB, C2h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Registers bits.*

Table 7. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Input	No pull-up, no interrupt (PA0-PA3 pins)
			Input	Analog input (PA4-PA7, PB0-PB7, PC4-PC7 pins)*
1	0	X	Output	20mA sink open-drain output (PA0-PA3 pins)
1	0	X	Output	Standard open-drain output (PA4-PA7, PB0-PB7, PC4-PC7 pins)*
1	1	X	Output	20mA sink push-pull output (PA0-PA3 pins)

Note: X = Don't care
* Device dependent

I/O PORTS (Cont'd)

4.1.5 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 25. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A and B Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

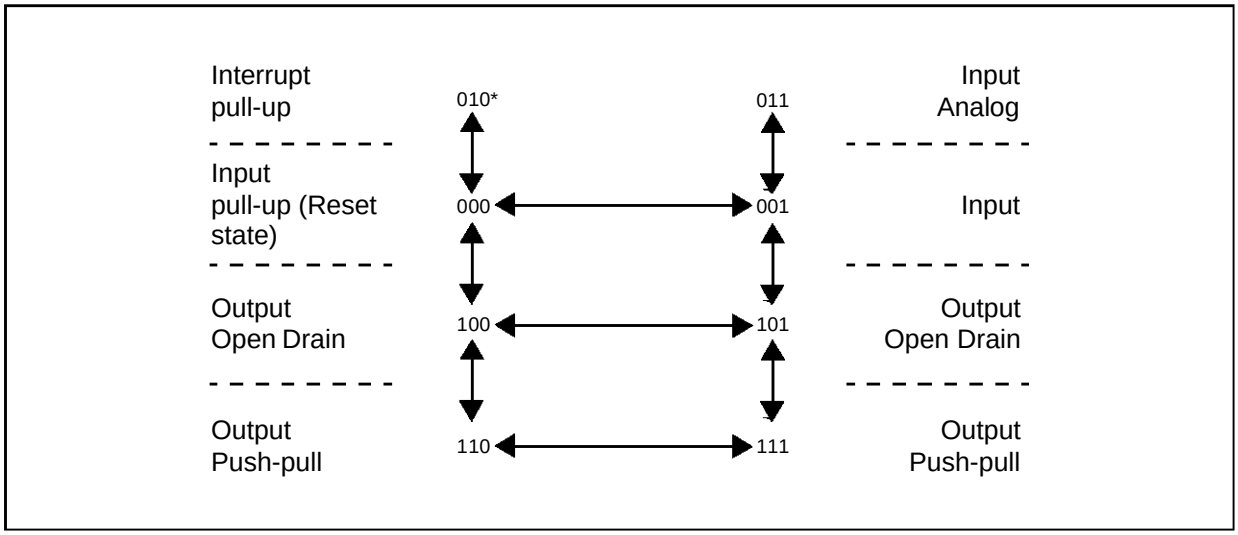
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

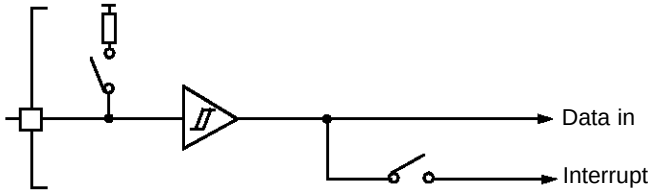
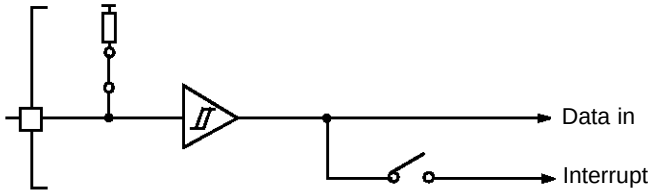
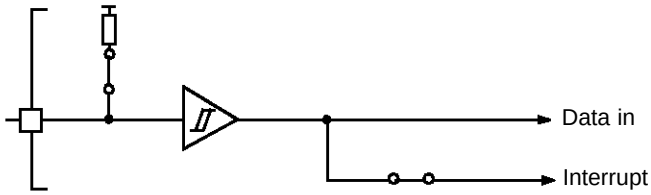
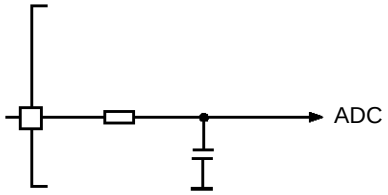
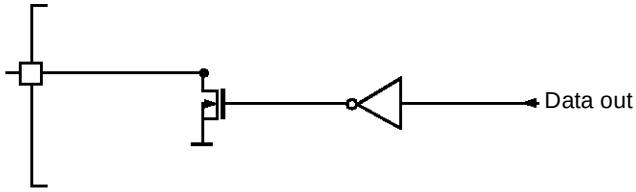
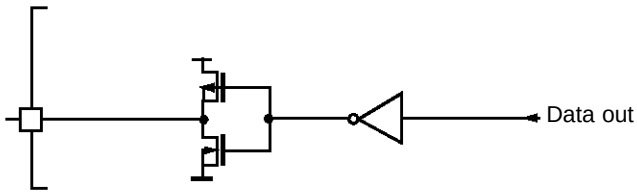
Figure 25. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 8. I/O Port Option Selections

MODE	AVAILABLE ON(1) (Device Dependent)	SCHEMATIC
Input	PA0-PA7 PB0-PB7 PC4-PC7	
Input with pull up	PA0-PA7 PB0-PB7 PC4-PC7	
Input with pull up with interrupt	PA0-PA7 PB0-PB7 PC4-PC7	
Analog Input (Device Dependent)	PA4-PA7 PB0-PB7 PC4-PC7	
Open drain output 5mA Open drain output 20mA	PA4-PA7 PB0-PB7 PC4-PC7 PA0-PA3	
Push-pull output 5mA Push-pull output 20mA	PA4-PA7 PB0-PB7 PC4-PC7 PA0-PA3	

Note 1. Provided the correct configuration has been selected.

4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}. The peripheral may be configured in three different operating modes.

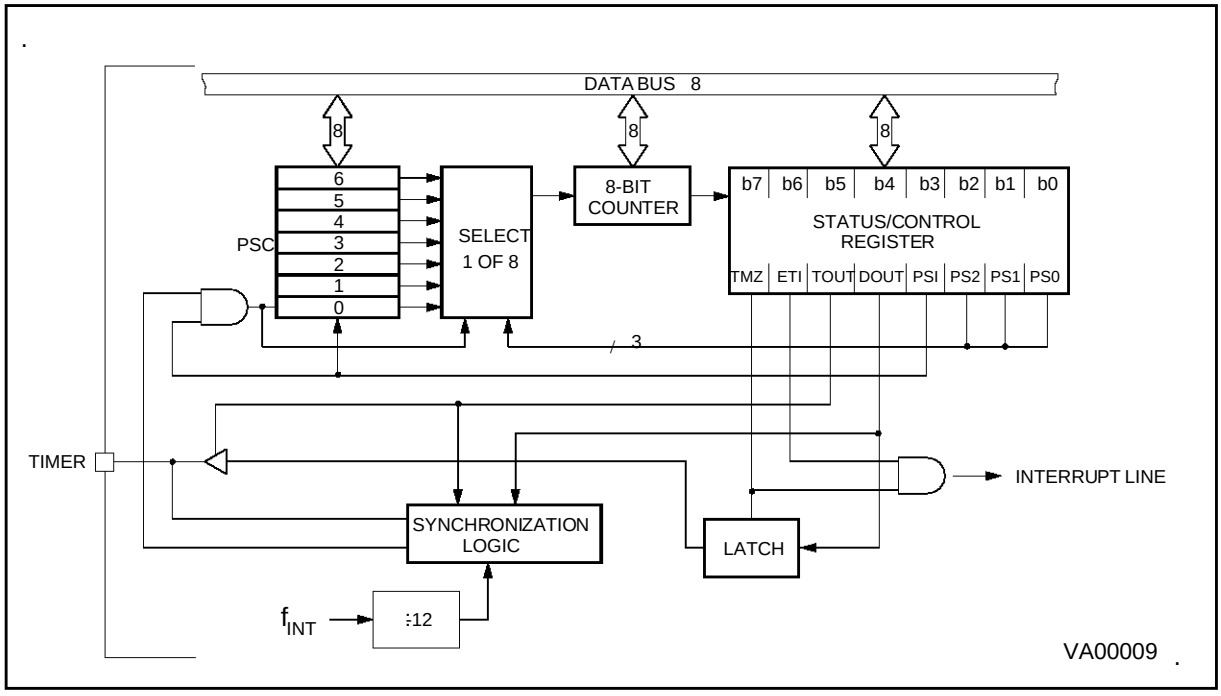
Figure 26 shows the Timer Block Diagram. The external TIMER pin is available to the user. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero) bit in the TSCR is set to "1". If the ETI (Enable Timer Interrupt) bit in the TSCR is also set to "1", an interrupt request, associated with interrupt vector #3, is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input can be the internal frequency f_{INT} divided by 12 or an external clock applied to the TIMER pin. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 10), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to "1" to allow the prescaler (and hence the counter) to start. If it is cleared to "0", all the prescaler bits are set to "1" and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set to "1". The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 27 illustrates the Timer's working principle.

Figure 26. Timer Block Diagram



TIMER (Cont'd)

4.2.1 Timer Operating Modes

There are three operating modes, which are selected by the TOUT and DOUT bits (see TSCR register). These three modes correspond to the two clocks which can be connected to the 7-bit prescaler ($f_{INT} \div 12$ or TIMER pin signal), and to the output mode.

4.2.2 Gated Mode

(TOUT = "0", DOUT = "1")

In this mode the prescaler is decremented by the Timer clock input ($f_{INT} \div 12$), but ONLY when the signal on the TIMER pin is held high (allowing pulse width measurement). This mode is selected by clearing the TOUT bit in the TSCR register to "0" (i.e. as input) and setting the DOUT bit to "1".

4.2.3 Clock Input Mode

(TOUT = "0", DOUT = "0")

In this mode, the TIMER pin is an input and the prescaler is decremented on the rising edge.

4.2.4 Output Mode

(TOUT = "1", DOUT = data out)

The TIMER pin is connected to the DOUT latch, hence the Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and transfer it to the TIMER pin. This operating mode allows external signal generation on the TIMER pin.

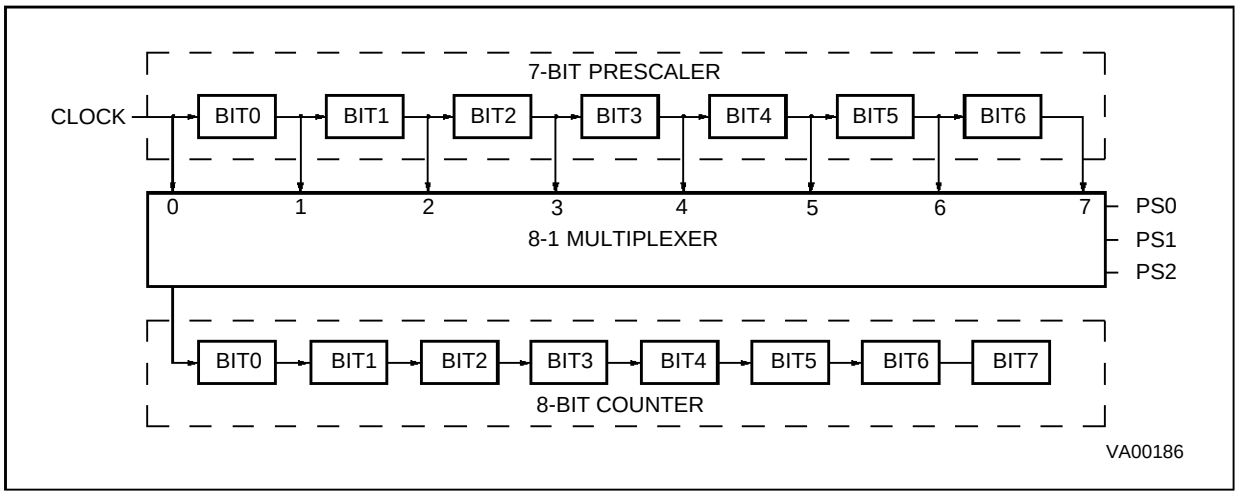
Table 9. Timer Operating Modes

TOUT	DOUT	Timer Pin	Timer Function
0	0	Input	Event Counter
0	1	Input	Gated Input
1	0	Output	Output "0"
1	1	Output	Output "1"

4.2.5 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

Figure 27. Timer Working Principle



TIMER (Cont'd)

4.2.6 Application Notes

The user can select the presence of an on-chip pull-up on the TIMER pin as a ROM mask option (see Option List at the end of the Datasheet).

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

If the Timer is programmed in output mode, the DOUT bit is transferred to the TIMER pin when TMZ is set to one (by software or due to counter decrement). When TMZ is high, the latch is transparent and DOUT is copied to the timer pin. When TMZ goes low, DOUT is latched.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time

.

4.2.7 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	TOUT	DOUT	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrupt*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **TOUT**: *Timers Output Control*

When low, this bit selects the input mode for the TIMER pin. When high the output mode is selected.

Bit 4 = **DOUT**: *Data Output*

Data sent to the timer output when TMZ is set high (output mode only). Input mode selection (input mode only).

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 10. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register TCR

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

4.3 A/D CONVERTER (ADC)

The A/D converter peripheral is an 8-bit analog to digital converter with analog inputs as alternate I/O functions (the number of which is device dependent), offering 8-bit resolution with a typical conversion time of 70us (at an oscillator clock frequency of 8MHz).

The ADC converts the input voltage by a process of successive approximations, using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, conversion accuracy is decreased.

Selection of the input pin is done by configuring the related I/O line as an analog input via the Option and Data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously, to avoid device malfunction.

The ADC uses two registers in the data space: the ADC data conversion register, ADR, which stores the conversion result, and the ADC control register, ADCR, used to program the ADC functions.

A conversion is started by writing a “1” to the Start bit (STA) in the ADC control register. This automatically clears (resets to “0”) the End Of Conversion Bit (EOC). When a conversion is complete, the EOC bit is automatically set to “1”, in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

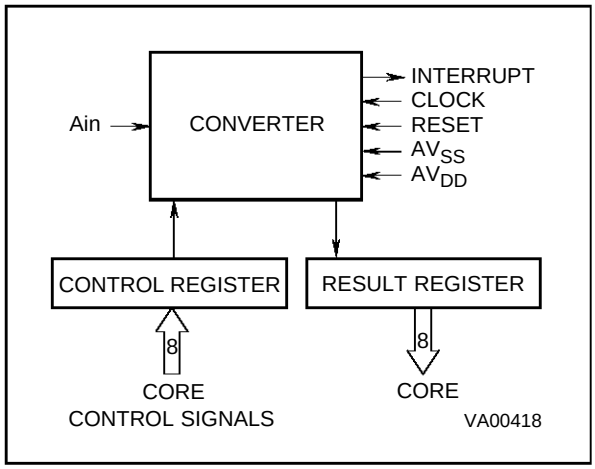
The STA bit is continuously scanned so that, if the user sets it to “1” while a previous conversion is in progress, a new conversion is started before completing the previous one. The start bit (STA) is a write only bit, any attempt to read it will show a logical “0”.

The A/D converter features a maskable interrupt associated with the end of conversion. This interrupt is associated with interrupt vector #4 and occurs when the EOC bit is set (i.e. when a conversion is completed). The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. This is done by setting the PDS bit in the ADC control register to “0”. If PDS=“1”, the A/D is powered and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow stabilisation of the A/D converter. This action is also needed before entering WAIT mode, since the A/D comparator is not automatically disabled in WAIT mode.

During Reset, any conversion in progress is stopped, the control register is reset to 40h and the ADC interrupt is masked (EAI=0).

Figure 28. ADC Block Diagram



4.3.1 Application Notes

The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for the optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 12pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. In worst case conditions, the impedance, ASI, of the analog voltage source is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband. ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

A/D CONVERTER (Cont'd)

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

The accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

The converter resolution is given by::

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

In order to optimise conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the less significant bits are determined.

The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. Indeed, only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. It should be noted that waking up the microcontroller could also be done using the Timer interrupt, but in this case the Timer will be working and the resulting noise could affect conversion accuracy.

A/D Converter Control Register (ADCR)

Address: 0D1h — Read/Write

7							0
EAI	EOC	STA	PDS	D3	D2	D1	D0

Bit 7 = **EAI**: *Enable A/D Interrupt.* If this bit is set to "1" the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

Bit 6 = **EOC**: *End of conversion. Read Only*. This read only bit indicates when a conversion has been completed. This bit is automatically reset to "0" when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to "1".

Bit 5 = **STA***: Start of Conversion. Write Only*. Writing a "1" to this bit will start a conversion on the selected channel and automatically reset to "0" the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

Bit 4 = **PDS***: Power Down Selection.* This bit activates the A/D converter if set to "1". Writing a "0" to this bit will put the ADC in power down mode (idle mode).

Bit 3-0 = **D3-D0.** Not used

A/D Converter Data Register (ADR)

Address: 0D0h — Read only

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0***: 8 Bit A/D Conversion Result.*

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 11. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:

X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 12. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 13. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected. The tested bit is shifted into carry.
* . Not Affected

Table 14. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not<M> Affected

Table 15. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ . Affected
*. Not Affected

Table 16. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
* . Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

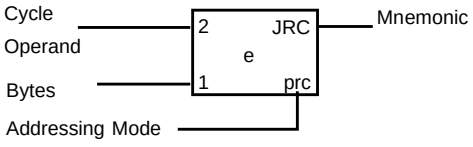
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	1 0001
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	3 0011
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	5 0101
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	7 0111
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	8 1000
9 1001	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	9 1001
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	B 1011
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	D 1101
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

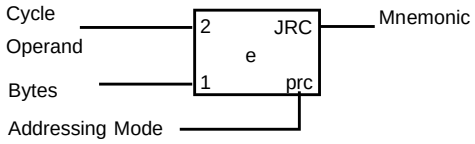
LOW / HI	**8 1000**	**9 1001**	**A 1010**	**B 1011**	**C 1100**	**D 1101**	**E 1110**	**F 1111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.
- (2) The total current through ports A and B combined may not exceed 50mA. The total current through port C may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 100mA.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	
		PSO20			80	
		PDIP28			55	
		PSO28			75	

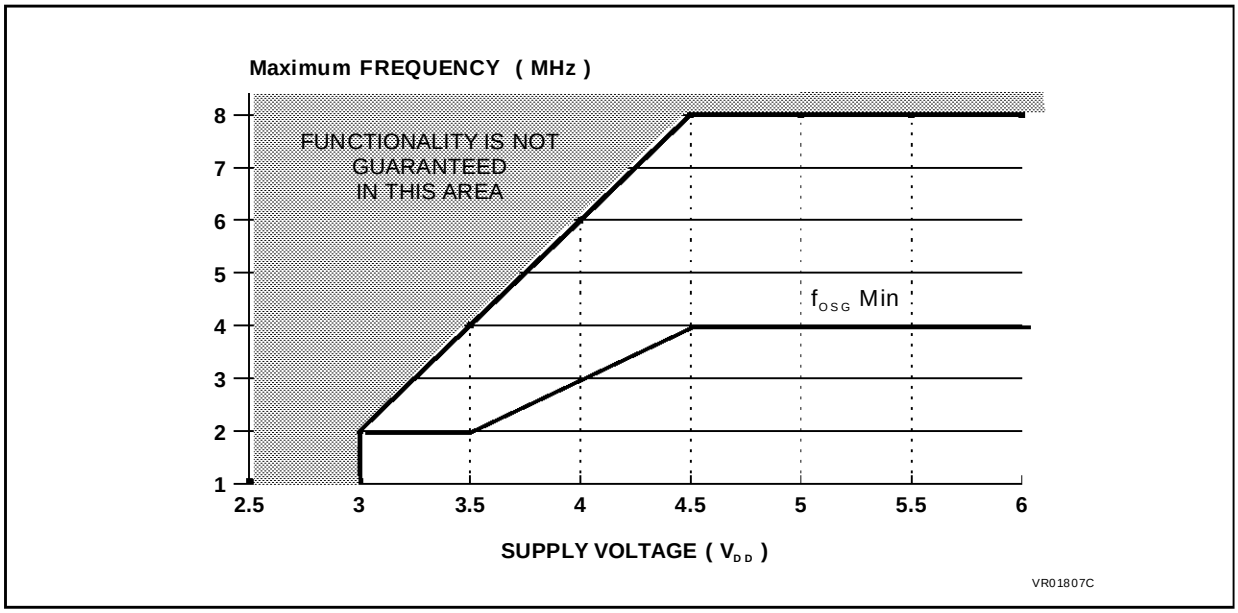
6.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

If a total current of +1mA is flowing into a single analog channel, or if the total current flowing into all the analog inputs is 1mA, all resulting A/D conversions will be shifted by + 1 LSB. If a total positive current is flowing into a single analog channel, or if the total current flowing into all analog inputs is 5mA, all the resulting conversions are shifted by + 2 LSB.

Figure 29. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



Note: The shaded area is outside the ST6210B, ST6215B, ST6220B and ST6225B recommended operating range; device functionality is not guaranteed under these conditions.

6.3 DC TELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage TIMER,NMI,RESET pins				V_{DD} x 0.3	V
V_{IH}	Input High Level Voltage TIMER,NMI,RESET pins		V_{DD} x 0.7			V
V_{Hys}	Hysteresis Voltage (4) All Inputs	V_{DD}= 5V V_{DD}= 3V		1 0.5		V
V_{OL}	Low Level Output Voltage TIMER pin	I_{OL} = + 5.0mA			0.2 x VDD	V
V_{OH}	High Level Output Voltage TIMER pin	I_{OH} = – 5.0mA	V_{DD} x 0.65			V
R_{PU}	Pull-up TIMER, NMI pins		50	100	200	kΩ
I_{IL} I_{IH}	Input Leakage Current(1) TIMER, NMI pins	V_{IN}= V_{SS} V_{IN}= V_{DD}		0.1	1.0	µA
I_{IL} I_{IH}	Input Leakage Current RESET pin	V_{IN}=V_{DD}; Watchdog Res. V_{IN}=V_{DD}; No Watch. Res. V_{IN}=V_{SS}; External Res.	 -8	 -16	1 10 -30	mA µA µA
I_{DD}	Supply Current in RESET Mode	V_{RESET}= V_{SS} f_{OSC}= 8MHz			3.5	mA
	Supply Current in RUN Mode (2)	V_{DD} = 5.0V f_{INT}=8MHz V_{DD} = 5.0V f_{INT}=f_{LFAO} V_{DD} = 3.0V f_{INT}=2MHz			3.5 TBD TBD	mA
	Supply Current in WAIT Mode (3)	V_{DD}= 5.0V f_{INT}=8MHz V_{DD} = 5.0V f_{INT}=f_{LFAO} V_{DD}= 3.0V f_{INT}=2MHz			1.50 TBD TBD	mA
	Supply Current in STOP Mode (3)	I_{LOAD}= 0mA V_{DD}= 5.0V			10	µA

Notes:
1. No watchdog reset activated.
2. All peripherals running
3. A/D Converter in Stand-by
4.Hysteresis voltage between switching levels

6.4 AC TELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency	V_{DD} = 3.0V; OSG disabled V_{DD} = 4.5V; OSG disabled			2 8	MHz
f_{OSG}	Maximum internal frequency with OSG enabled	VDD = 3.0V V_{DD} = 4.5V	2 4			MHz
f_{LFAO}	Low Frequency Auxiliary Oscillator		200	400	800	kHz
t_{SU}	Oscillator Start-up Time at Power On (2)	Ceramic Resonator C_{L1} = C_{L2} = 22pF		5	100	ms
t_{SUS}	Oscillator STOP mode Recovery Time (2)	8MHz Ceramic Resonator CL1=C_{L2}=22pF		0.2	100	^
		8MHz Quartz CL1=C_{L2}=22pF		10	100	^
t_{REC}	Supply Recovery Time (1)		100			^
T_{WR}	Minimum Pulse Width (V_{DD} = 5V) RESET pin NMI pin		 100 100			ns
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:
1. Period for which V_{DD} has to be connected at 0V to allow internal Reset function at next power-up.
2. This value is highly dependent on the Ceramic Resonator or Quartz Crystal used in the application.

Figure 30. Power-on-Reset

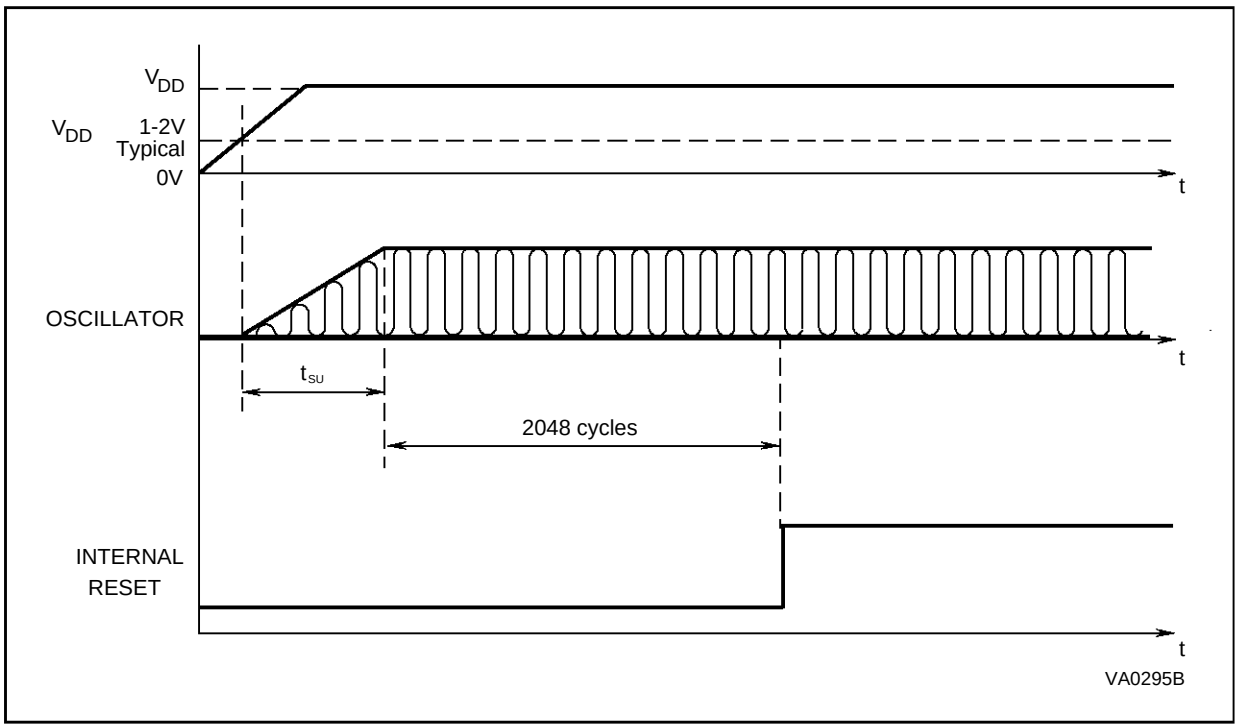
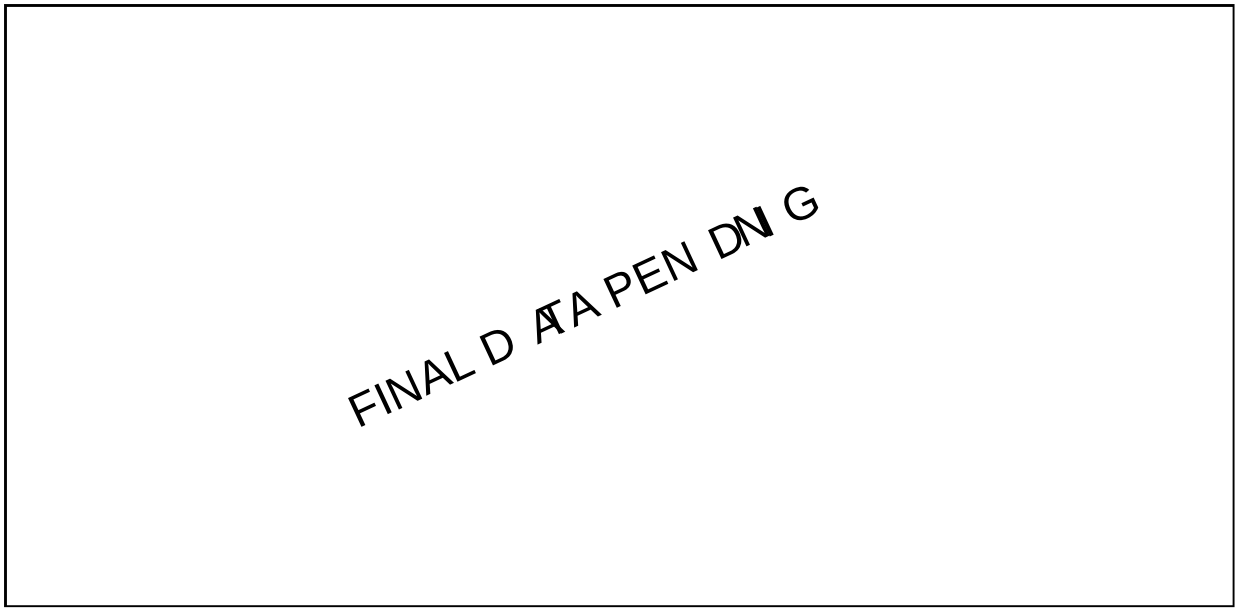
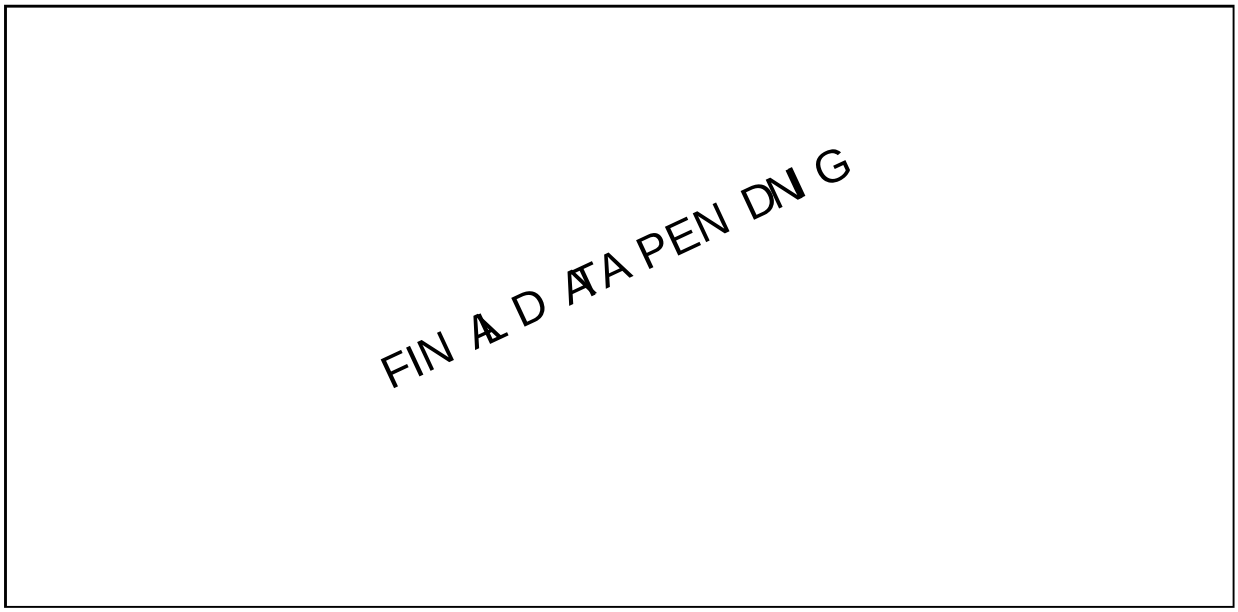


Figure 31. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the ST6210B, ST6215B, ST6220B and ST6225B recommended operating range; device functionality is not guaranteed under these conditions.

Figure 32. RC Oscillator. F_{INT} versus RNET (Indicative Values)



Note: The shaded area is outside the ST6210B, ST6215B, ST6220B and ST6225B recommended operating range; device functionality is not guaranteed under these conditions.

6.5 READOUT PROTECTION FUSE

If the ROM READOUT PROTECTION option is selected, the waveform illustrated below must be applied to the TEST pin in order to blow the fuse.

The following circuit can be used for this purpose:

Figure 33. Programming wave form

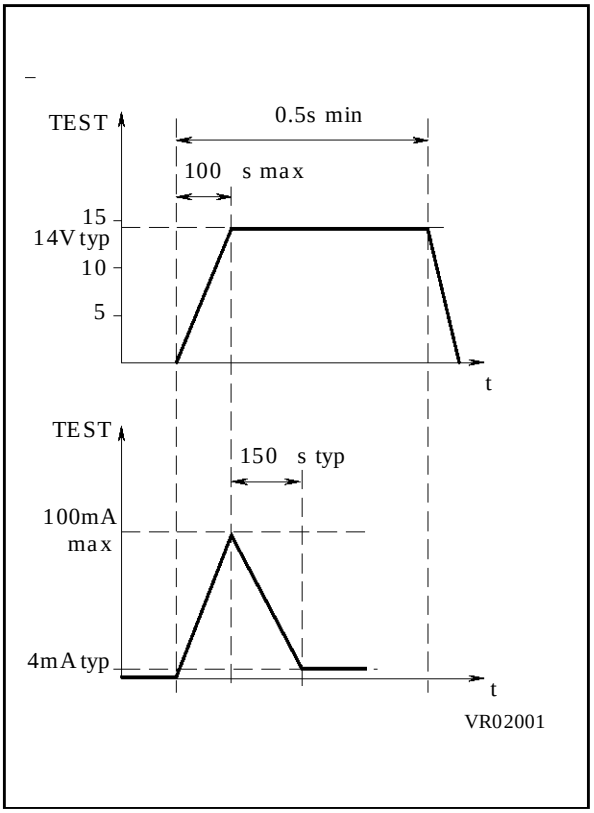
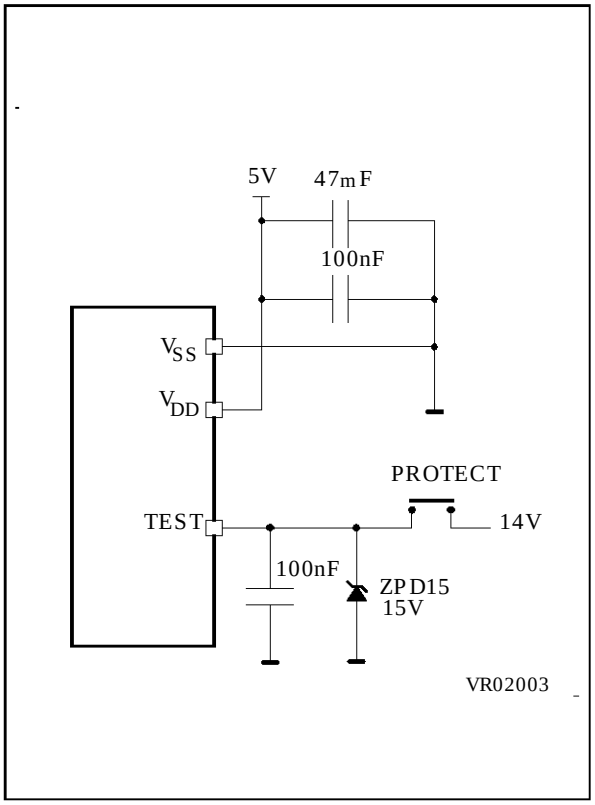


Figure 34. Programming Circuit

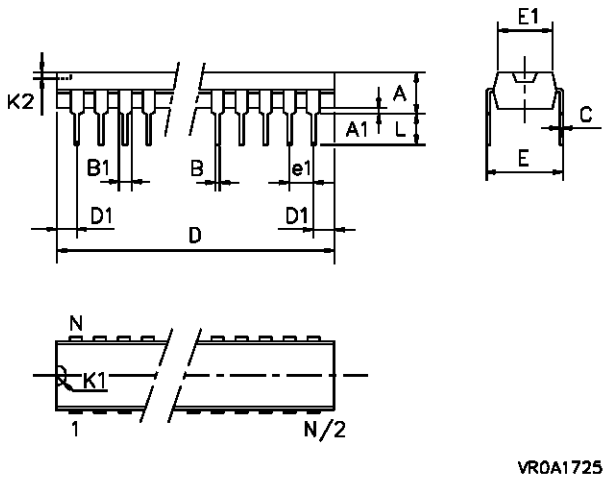


Note: ZPD15 is used for overvoltage protection

7 GENERAL INFORMATION

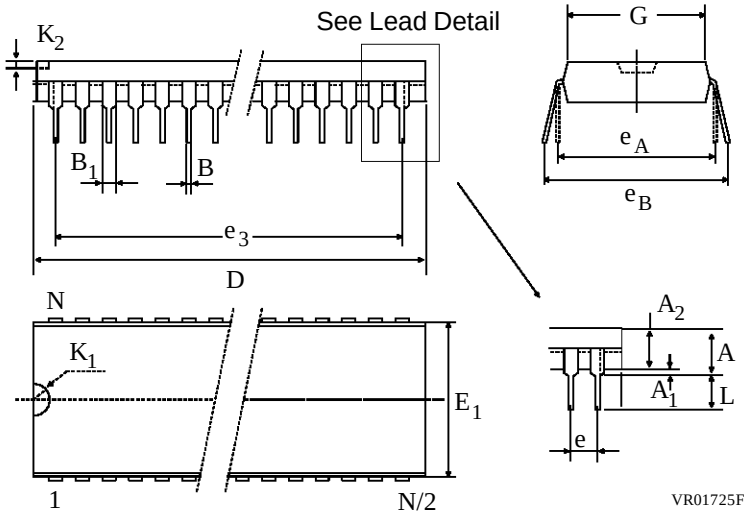
7.1 PACKAGE MECHANICAL DATA

Figure 35. 20-Pin Plastic Dual In-Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
A2	-	-	-	-	-	-
B		0.45			0.018	
B1	1.39		1.65	0.055		0.065
C		0.25			0.010	
D			25.4			1.000
e3		22.86			0.900	
E1			7.1			0.280
e		2.54			0.100	
eB		8.5			0.335	
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.3			0.130	
	Number of Pins					
N	20					

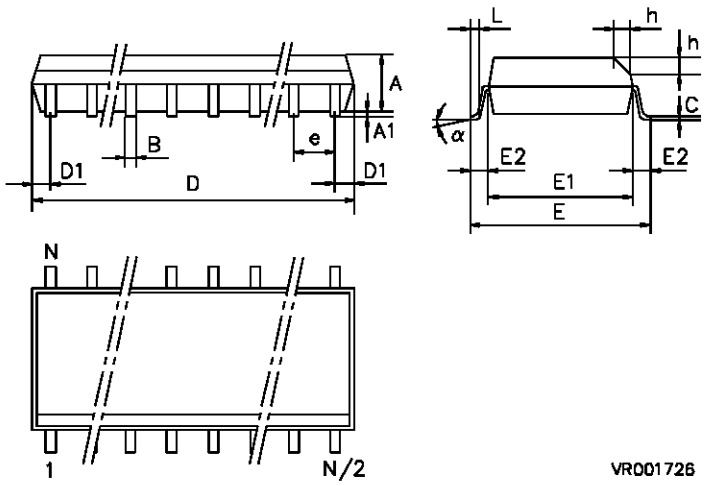
Figure 36. 28-Pin Plastic Dual In-Line Package, 600-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		4.445			0.175	
A1		0.63			0.025	
A2	-	-	-	-	-	-
B		0.45			0.018	
B1		1.27			0.050	
C	0.23		0.310	0.009		0.012
D			37.34			1.470
e3		33.02			1.300	
E1			14.10			0.555
e		2.54			0.100	
eA						
eB		15.20	16.68		0.598	0.657
G						
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.30			0.130	
	Number of Pins					
N	28					

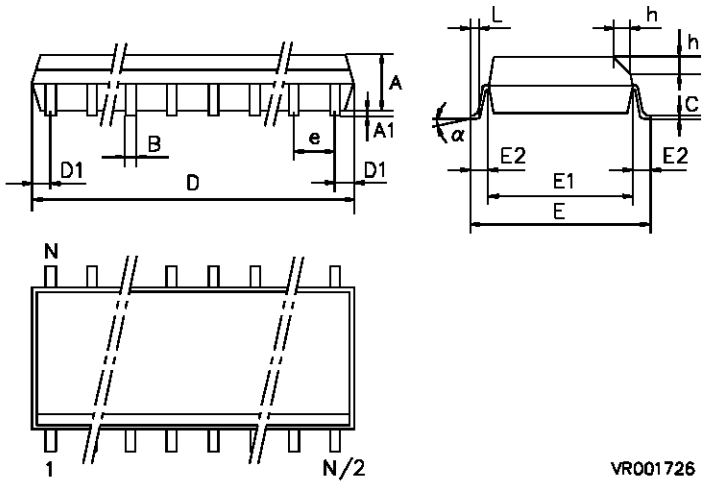
PACKAGE MECHANICAL DATA (Cont'd)

Figure 37. 20-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

Figure 38. 28-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	17.70		18.10	0.697		0.713
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	28					

ST6210B, ST6215B, ST6220B and ST6225B MICROCONTROLLER OPTION LIST

Customer
Address

Contact
Phone No
Reference

SGS-THOMSON Microelectronics references
Device: [ ] ST6210B[ ] ST6215B[ ] ST6220B[ ] ST6225B
Package: [ ] Dual in Line Plastic [ ] Small Outline Plastic
In this case, select conditioning
[ ] Standard (Stick)
[ ] Tape & Reel
Temperature Range: [ ] 0°C to + 70°C[ ] - 40°C to + 85°C
Special Marking: [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _ "
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count: DIP20 - DIP28: 10
SO20 - SO28: 8
Oscillator Source Selection:[ ] Crystal Quartz/Ceramic resonator (Default)
[ ] RC Network
Watchdog Selection: [ ] Software Activation (STOP mode available)
[ ] Hardware Activation (no STOP mode)
OSG: [ ] Enabled
[ ] Disabled (Default)
Input pull-up selection on NMI pin:[ ] Yes [ ] No
Input pull-up selection on TIMER pin: [ ] Yes[ ] No
ROM Readout Protection: [ ] Standard (Fuse cannot be blown)
[ ] Enabled (Fuse can be blown by the customer)
Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.
External STOP Mode Control[ ] Enabled [ ] Disabled (Default)
Comments :
Supply Operating Range in the application:
Oscillator Fequency in the application:
Notes

Signature
Date

7.2 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

7.2.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

7.2.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 17. Program Memory Map for ST6210B & ST6215B (1836 Bytes ROM)

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 18. Program Memory Map for ST6220B & ST6225B (3884 Bytes ROM)

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 1. Reserved Areas should be filled with FFh

Table 19. Ordering Information

Sales Type	ROM	I/O	Additional Features	Temperature Range	Package
ST6210BB1/XXX ST6210BB6/XXX	1836 Bytes	12	A/D CONVERTER	0 to +70°C -40 to + 85°C	PDIP20
ST6220BB1/XXX ST6220BB6/XXX	3884 Bytes			0 to +70°C -40 to + 85°C	
ST6210BM1/XXX ST6210BM6/XXX	1836 Bytes			0 to +70°C -40 to + 85°C	PSO20
ST6220BM1/XXX ST6220BM6/XXX	3884 Bytes			0 to +70°C -40 to + 85°C	
ST6215BB1/XXX ST6215BB6/XXX	1836 Bytes	20		0 to +70°C -40 to + 85°C	PDIP28
ST6225BB1/XXX ST6225BB6/XXX	3884 Bytes			0 to +70°C -40 to + 85°C	
ST6215BM1/XXX ST6215BM6/XXX	1836 Bytes			0 to +70°C -40 to + 85°C	PSO28
ST6225BM1/XXX ST6225BM6/XXX	3884 Bytes			0 to +70°C -40 to + 85°C	

Note: /XXX is a 2-3 alphanumeric character code added to the generic sales type on receipt of a ROM code and valid options

Notes:

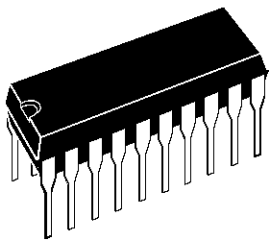
SGS-THOMSON MICROELECTRONICS

ST62T53B

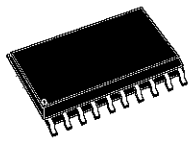
8-BIT OTP MCUs WITH A/D CONVERTER & AUTO-RELOAD TIMER

PRODUCT PREVIEW

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- User OTP: 1836 bytes
- Data RAM: 64 bytes
- User Programmable Options
- 13 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 6 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- 8-bit Auto-reload Timer with 7-bit programmable prescaler (AR Timer)
- Digital Watchdog
- 8-bit A/D Converter with 7 analog inputs
- On-chip Clock oscillator can be driven by Quartz Crystal Ceramic resonator or RC network
- User configurable Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU2 Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	OTP (Bytes)	RAM	I/O Pins
ST62T53B	1836	64	13

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T53B is a low cost member 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.

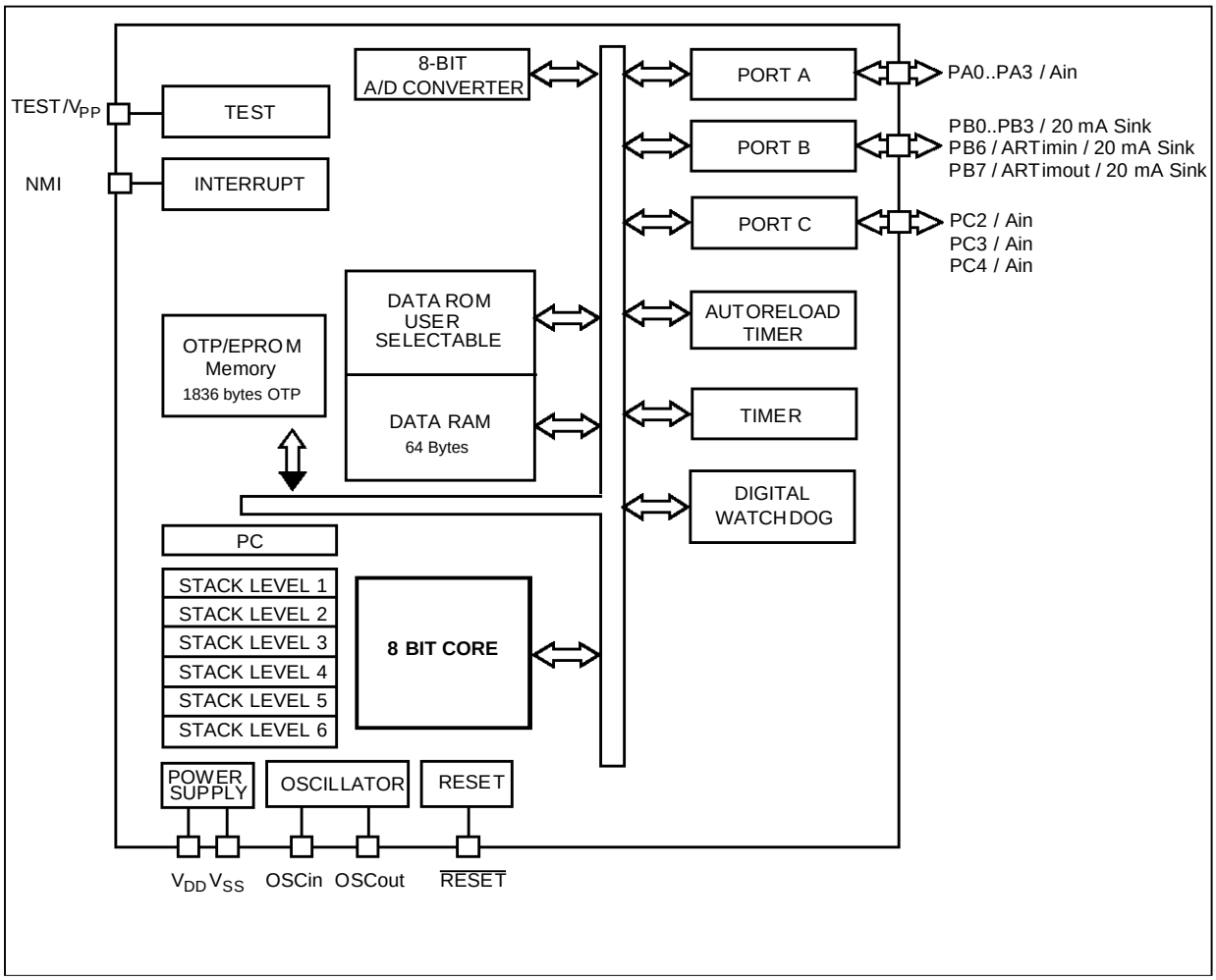
The ST62E60B device may be used to emulate and debug the ST62T53B OTP device, but care should be taken to use only the resources available on the target device.

The ROM based version offers the same functionality selecting as ROM options the options defined in the programmable option byte of the OTP version.

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.

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

Figure 1. Block Diagram



1.2 PIN DESCRIPTIONS

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation. If TEST pin is connected to a +12.5V level during the reset phase, the EPROM/OTP programming Mode is entered.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. It is provided with an on-chip pullup resistor and Schmitt trigger characteristics.

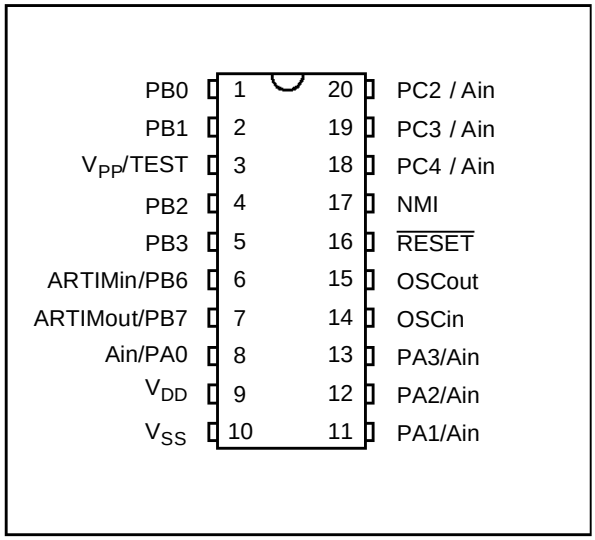
PA0-PA3. These 4 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs, analog inputs for the A/D converter.

PB0-PB3. These 4 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs.
PB0-PB3 can also sink 20mA for direct LED driving.

PB6/ARTIMin, PB7/ARTIMout. These pins are either Port B I/O bits or the Input and Output pins of the AR TIMER. To be used as timer input function PB6 has to be programmed as input with or without pull-up. A dedicated bit in the AR TIMER Mode Control Register sets PB7 as timer output function.
PB6-PB7 can also sink 20mA for direct LED driving.

PC2-PC4. These 3 lines are organized as one I/O port (C). Each line may be configured under software control as input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, analog input for the A/D converter, open-drain or push-pull output.

Figure 2. ST62T53B Pin Configuration



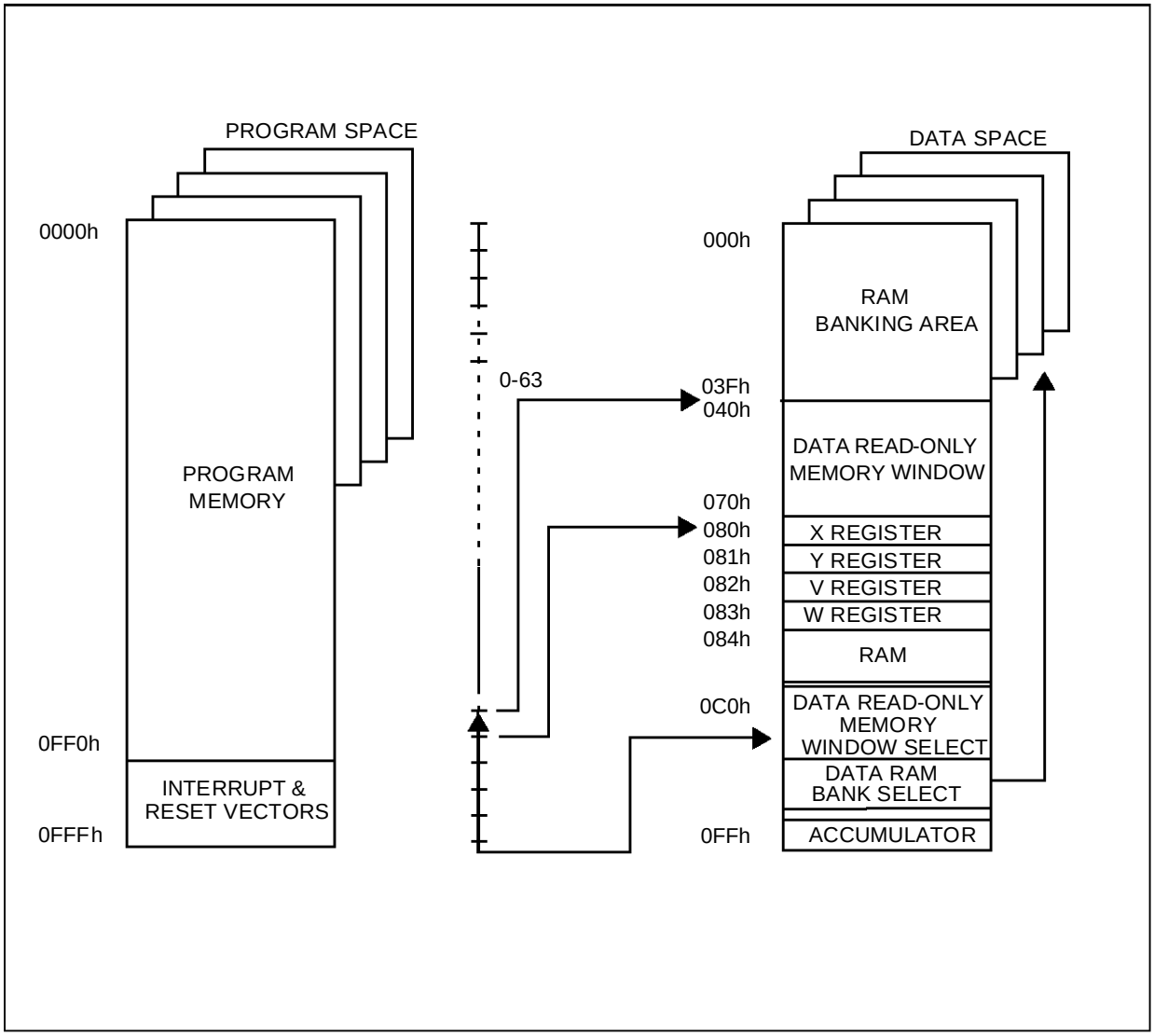
1.3 MEMORY MAP

1.3.1 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 subroutine and interrupt service routine nesting.

Figure 3. Memory Addressing Diagram



MEMORY MAP (Cont'd)

1.3.2 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).

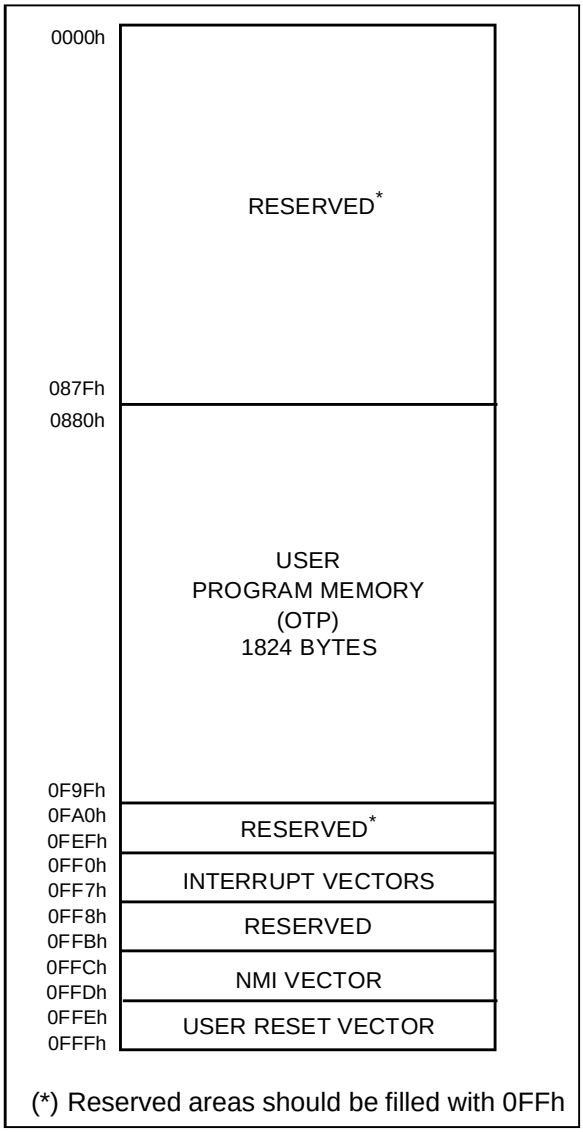
1.3.2.1 Program Memory Protection

The Program Memory in OTP or EPROM devices can be protected against external readout of memory by selecting the READOUT PROTECTION option in the option byte.

In the EPROM parts, READOUT PROTECTION option can be disactivated only by U.V. erasure that also results into the whole EPROM context erasure.

Note: Once the Readout Protection is activated, it is no longer possible, even for SGS-THOMSON, to gain access to the OTP contents. Returned parts with a protection set can therefore not be accepted.

Figure 4. ST62T53B Program Memory Map



MEMORY MAP (Cont'd)

1.3.3 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.

1.3.3.1 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.

1.3.3.2 Data RAM

In the MCU, the data space includes 60 bytes of RAM, 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 (DRW register).

1.3.4 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.

Table 1. ST62T53B Data Memory Space

NOT IMPLEMENTED	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
PORT C OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PRESCALER REGISTER	0D2h
TIMER COUNTER REGISTER	0D3h
TIMER STATUS CONTROL REGISTER	0D4h
AR TIMER MODE CONTROL REGISTER	0D5h
AR TIMER STATUS/CONTROL REGISTER1	0D6h
AR TIMER STATUS/CONTROL REGISTER2	0D7h
WATCHDOG REGISTER	0D8h
AR TIMER RELOAD/CAPTURE REGISTER	0D9h
AR TIMER COMPARE REGISTER	0DAh
AR TIMER LOAD REGISTER	0DBh
OSCILLATOR CONTROL REGISTER	0DCh*
MISCELLANEOUS	0DDh
RESERVED	0DEh 0DFh
RESERVED	0E0h
RESERVED	0E1h
RESERVED	0E2h
RESERVED	0E3h 0E7h
RESERVED	0E8h*
RESERVED	0E9h
RESERVED	0EAh
RESERVED	0EBh 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

1.3.5 Data Window Register

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 1FFFh (top memory address depends on the specific device). All the program memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the program memory by writing the appropriate code in the Data Window Register (DWR).

The DWR can be addressed like any RAM location in the Data Space, it is however a write-only register and therefore cannot be accessed using single-bit operations. This register is used to position the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) in program memory in 64-byte steps. The effective address of the byte to be read as data in program memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits), as illustrated in Figure 5 below. For instance, when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in program memory is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data read-only memory window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

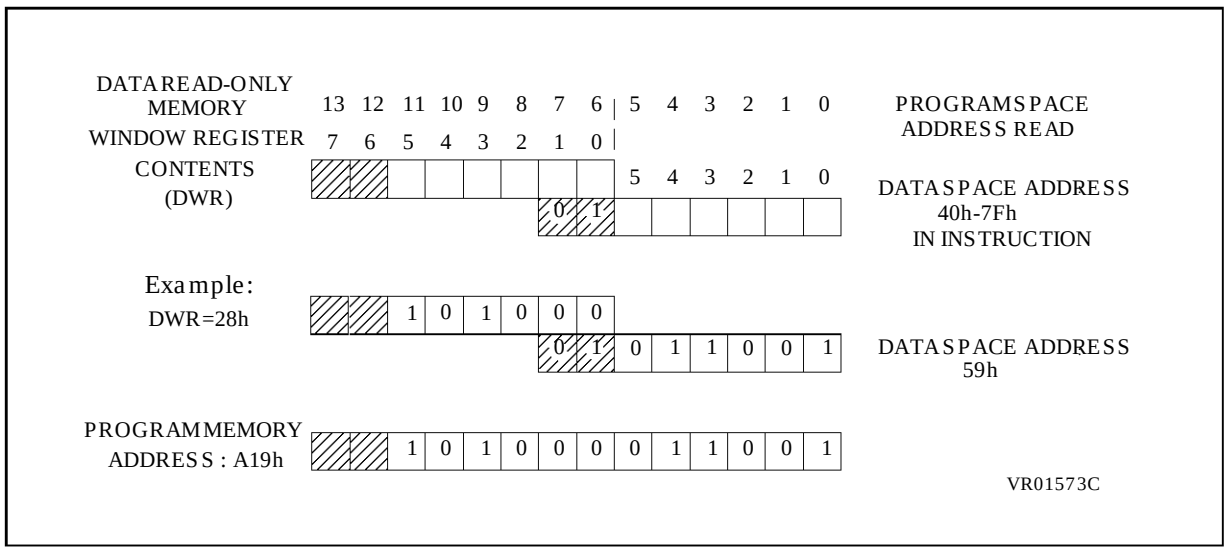
Bits 6, 7 = Not used.

Bit 5-0 = **DWR5-DWR0:** *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. Neither read nor single bit instructions may be used to address this register.*

Note: Care is required when handling the DWR register as it is write only. For this reason, the DWR 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 DWR 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 DWR, 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 DWR is not affected.

Figure 5. Data read-only memory Window Memory Addressing



1.4 PROGRAMMING MODES

1.4.1 Option Byte

The Option Byte allows configuration capability to the MCUs. Option byte's content is automatically read, and the selected options enabled, when the chip reset is activated.

It can only be accessed during the programming mode. This access is made either automatically (copy from a master device) or by selecting the OPTION BYTE PROGRAMMING mode of the programmer.

The option byte is located in a non-user map. No address has to be specified.

EPROM Code Option Byte

7							0
PRO-TECT	EXTC-NTL	-	-	WDACT	DELAY	OSCIL	-

PROTECT. This bit allows the protection of the software contents against piracy. When the bit PROTECT is set high, readout of the OTP contents is prevented by hardware. No programming equipment is able to gain access to the user program. When this bit is low, the user program can be read.

EXTCNTL. This bit selects the External STOP Mode capability. When EXTCNTL is high, pin NMI controls if the STOP mode can be accessed when the watchdog is active. In addition, PB0 is forced as open drain output. When EXTCNTL is low, the STOP instruction is processed as a WAIT as soon as the watchdog is active.

D5-D4. Reserved. Must be cleared to zero.

WDACT. This bit controls the watchdog activation. When it is high, hardware activation is selected. The software activation is selected when WDACT is low.

DELAY. This bit enables the selection of the delay internally generated after pin RESET is released. When DELAY is low, the delay is 2048 cycles of the oscillator, it is of 32768 cycles when DELAY is high.

OSCIL. When this bit is low, the oscillator must be controlled by a quartz crystal, a ceramic resonator or an external frequency. When it is high, the oscillator must be controlled by an RC network, with only the resistor having to be externally provided.

D0. Reserved. Must be cleared to zero.

The Option byte is written during programming either by using the PC menu (PC driven Mode) or automatically (stand-alone mode)

1.4.2 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 ST62E60B/T60B and ST62T63B is described in the User Manual of the EPROM Programming Board.

The MCUs can be programmed with the ST62E6xB EPROM programming tools available from SGS-THOMSON.

Table 2. ST62T53B 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

Note: OTP devices can be programmed with the development tools available from SGS-THOMSON (ST62E1X-EPB or ST622X-KIT).

2 CENTRAL PROCESSING UNIT

2.1 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. In-core communication is arranged as shown in Figure 6; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

2.2 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.

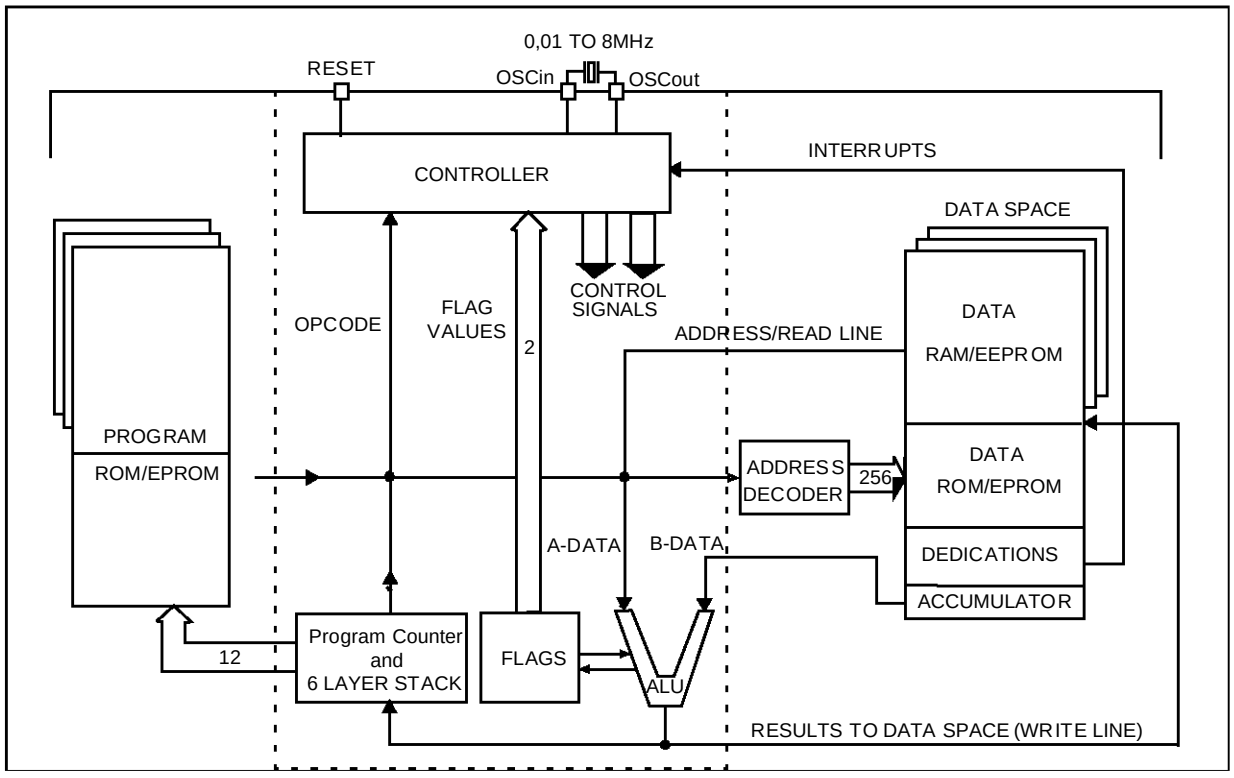
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 6. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

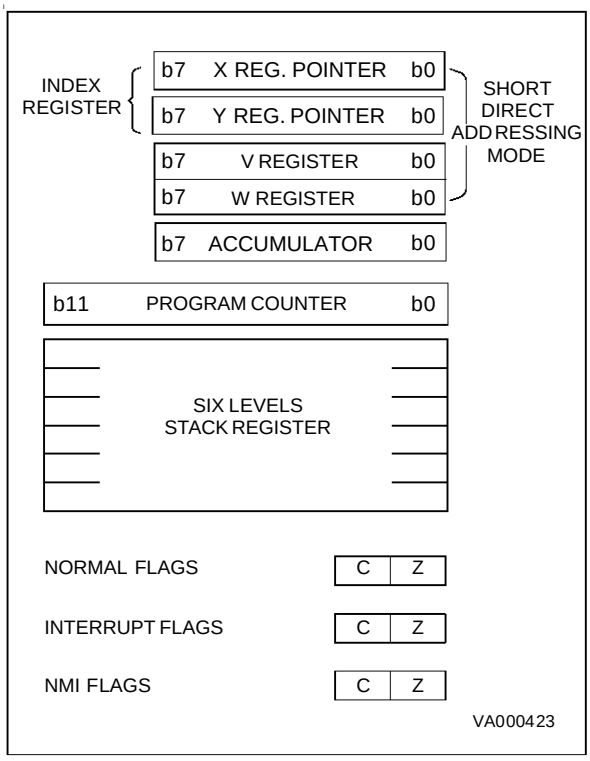
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 7. ST6 CPU Programming Mode



3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}).

Figure 8 illustrates various possible oscillator configurations using an external crystal or ceramic resonator, an external clock input, an external resistor (R_{NET}). C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 26 and Figure 27.

A programmable divider is provided in order to adjust the internal clock of the MCU to the best power consumption and performance trade-off.

The internal MCU clock frequency (f_{INT}) drives directly the AR TIMER while it is divided by 12 to drive the TIMER, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 9.

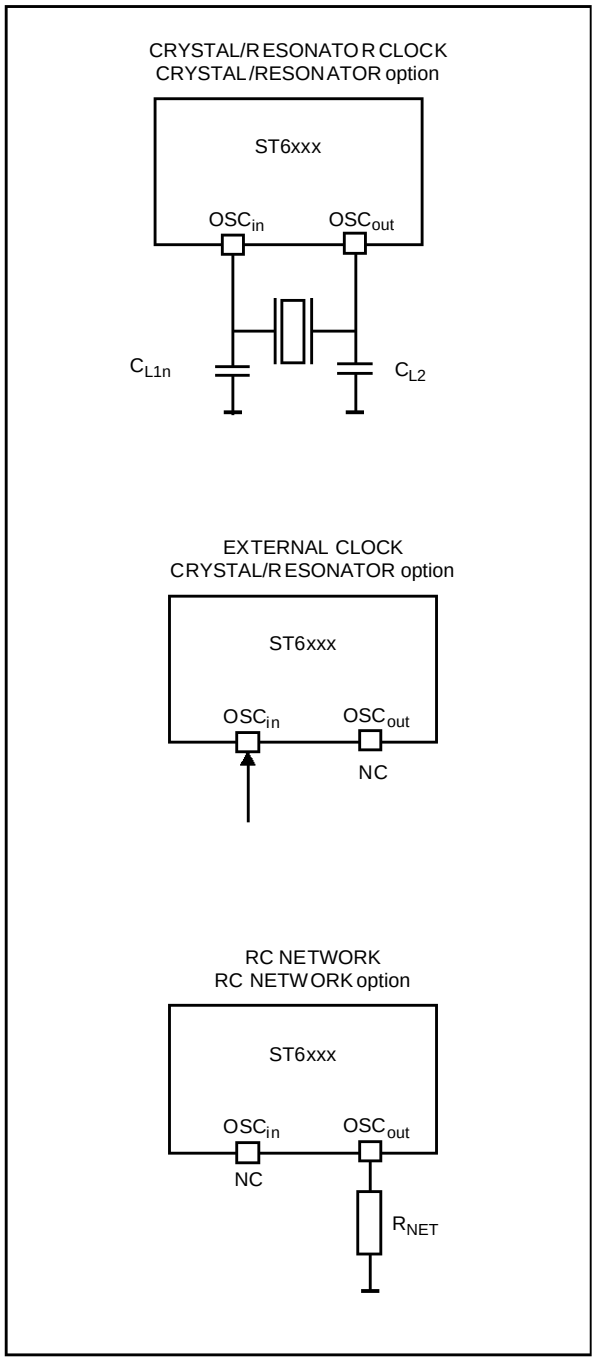
With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

A machine 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 machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate option. 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.

Figure 8. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Oscillator Control Registers

Address: DCh — Write only

7							0
-	-	-	-	OSCR 3	OSCR 2	RS1	RS0

Bit 7-4. These bits are not used.

Bit 3. Reserved. Cleared at Reset. THIS BIT MUST BE SET TO 1 BY USER PROGRAM to achieve lowest power consumption.

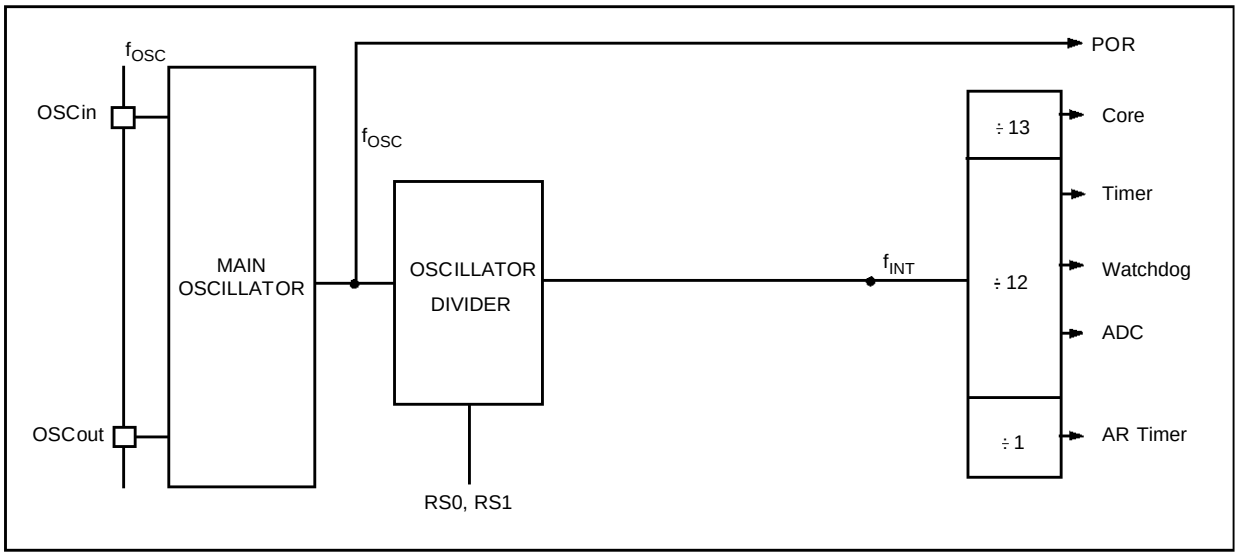
Bit 2. Reserved. Must be kept low.

RS1-RS0. These bits select the division ratio of the Oscillator Divider in order to generate the internal frequency. The following selctions are available:

RS1	RS0	Division Ratio
0	0	1
0	1	2
1	0	4
1	1	4

Note: Care is required when handling the OSCR register as some bits are write only. For this reason, it is not allowed to change the OSCR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to OSCR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the OSCR is not affected.

Figure 9. Clock Circuit Block Diagram



3.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

3.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

3.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

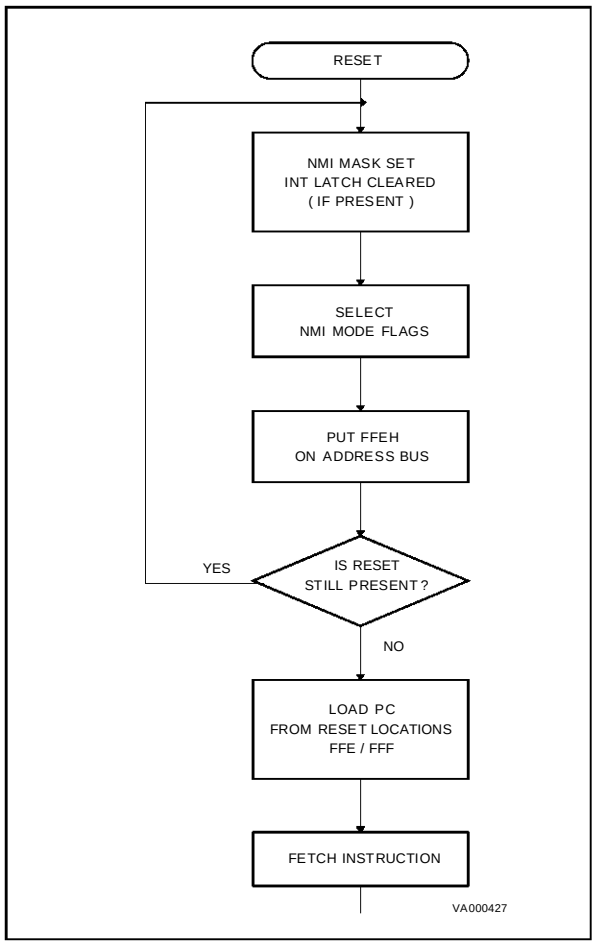
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 10. Reset and Interrupt Processing



RESETS (Cont'd)

3.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

3.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

3.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 11. Reset and Interrupt Processing

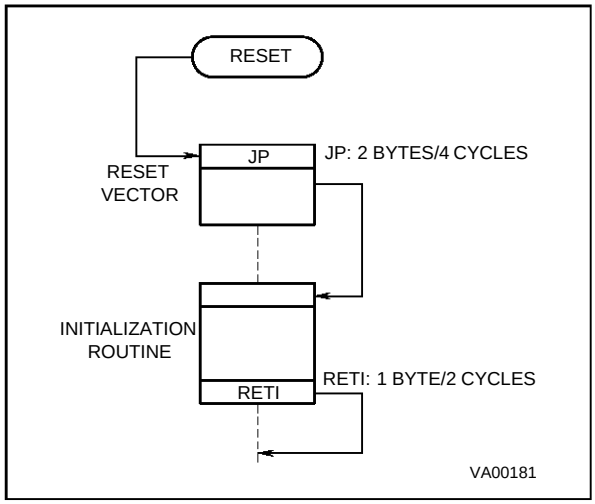
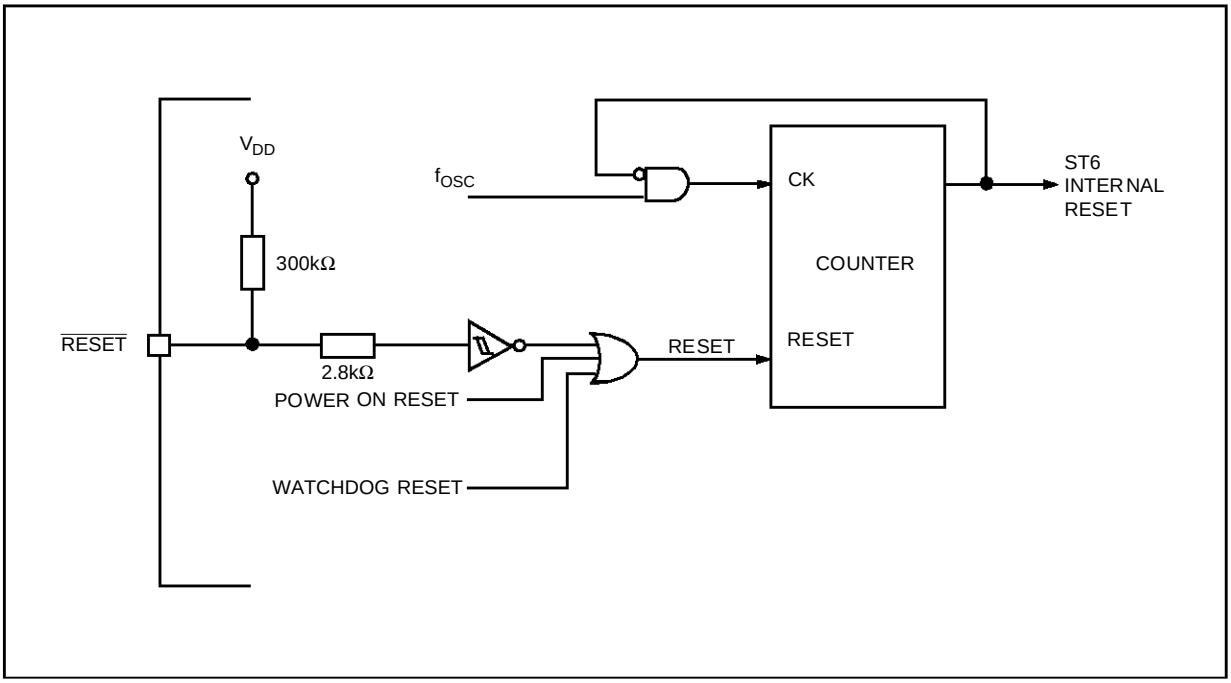


Figure 12. Reset Block Diagram



RESETS (Cont'd)

Table 3. Register Reset Status

Register	Address(es)	Status	Comment
Oscillator Control Register	0DCh		$f_{INT} = f_{OSC}$; user must set bit 3 to 1
Port Data Registers	0C0h to 0C2h		I/O are Input with pull-up
Port Direction Register	0C4h to 0C6h		I/O are Input with pull-up
Port Option Register	0CCh to 0CEh		I/O are Input with pull-up
Interrupt Option Register	0C8h		Interrupt disabled
TIMER Status/Control	0D4h	00h	TIMER disabled
AR TIMER Mode Control Register	0D5h		AR TIMER stopped
AR TIMER Status/Control 1 Register	0D6h		
AR TIMER Status/Control 2Register	0D7h		
AR TIMER Compare Register	0DAh		
Miscellaneous Register	0DDh		
X, Y, V, W, Register	080H TO 083H		
Accumulator	0FFh		
Data RAM	084h to 0BFh		
Data RAM Page REgister	0E8h	Undefined	
Data ROM Window Register	0C9h		
A/D Result Register	0D0h		
AR TIMER Load Register	0DBh		
AR TIMER Reload/Capture Register	0D9h		
TIMER Counter Register	0D3h	FFh	
TIMER Prescaler Register	0D2h	7Fh	Max count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Standby

3.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two options, known as "WATCHDOG ACTIVATION" (i.e. HARDWARE or SOFTWARE) and "EXTERNAL STOP MODE CONTROL" (see Table 4).

In the SOFTWARE option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, except by resetting the MCU.

In the HARDWARE option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL option has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL option has been selected, port PB0 must be defined as an open-drain output.

Table 4. Recommended Option Choices

Functions Required	Recommended Options
Stop Mode & Watchdog	"EXTERNAL STOP MODE" & "HARDWARE WATCHDOG"
Stop Mode	"SOFTWARE WATCHDOG"
Watchdog	"HARDWARE WATCHDOG"

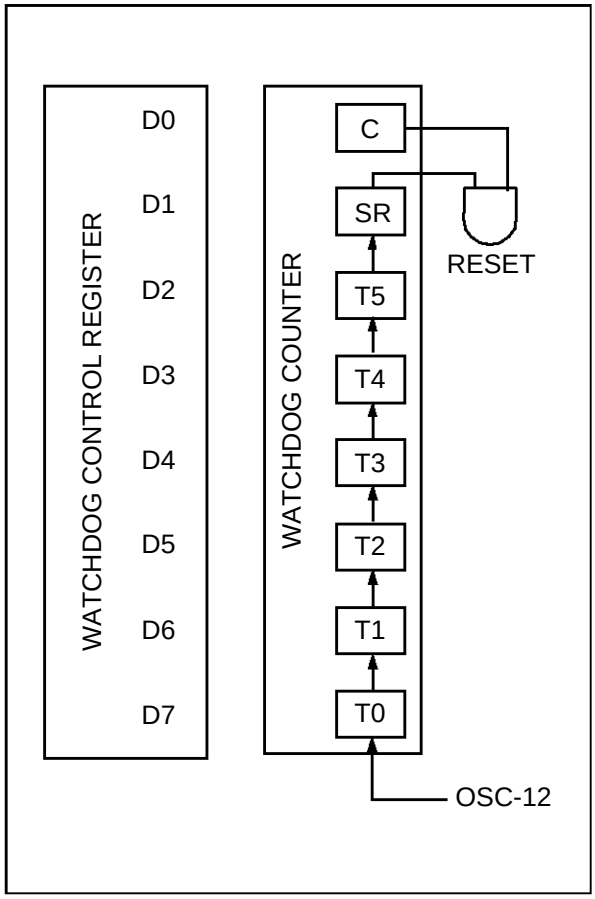
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 3.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 13.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 13. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 14) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 14. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

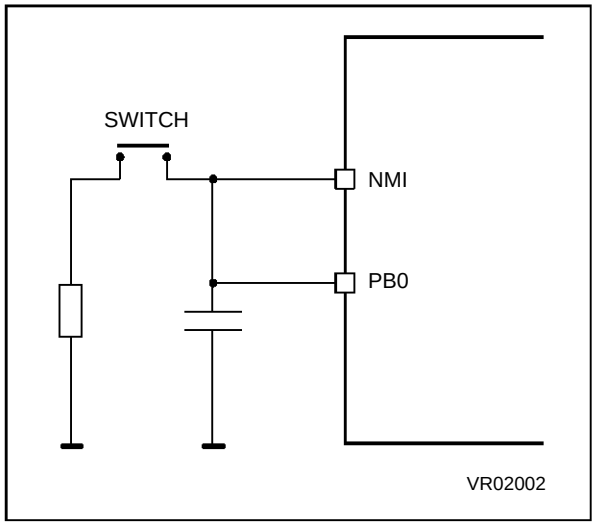
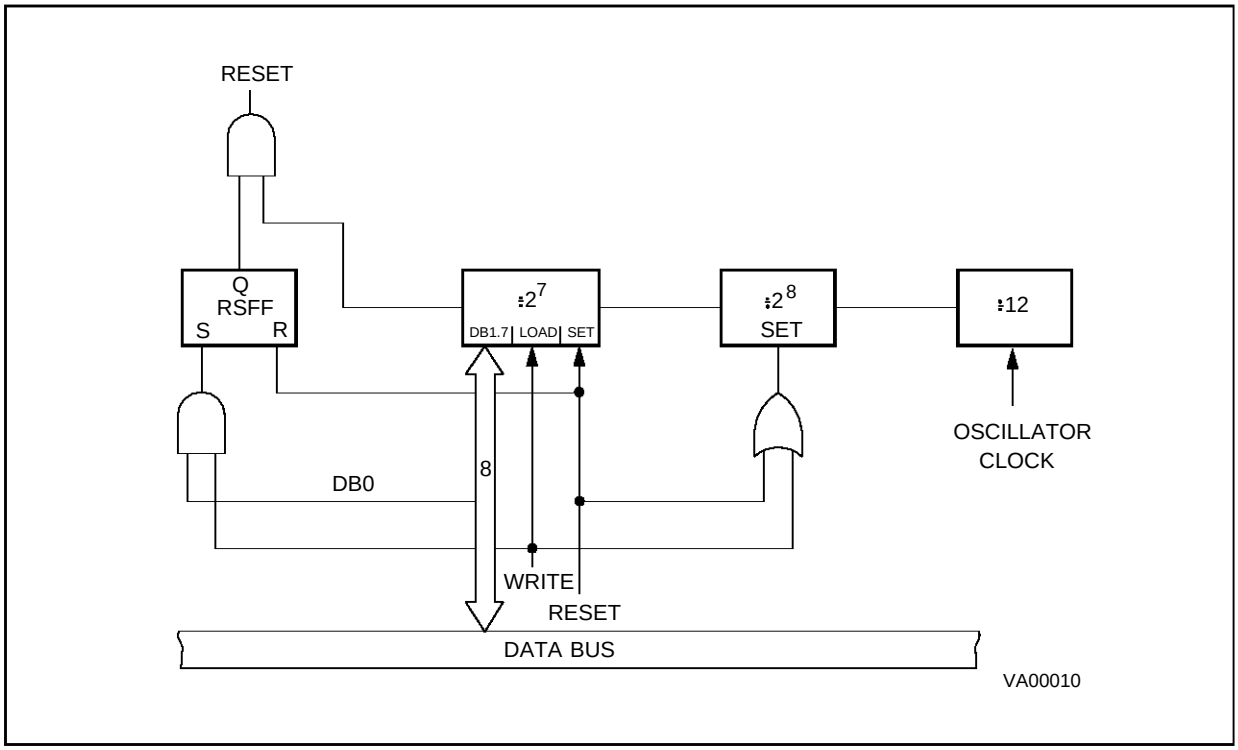


Figure 15. Digital Watchdog Block Diagram



3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 5).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 5. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A, B	Interrupt vector #1	(FF6h-FF7h)
Port C	Interrupt vector #2	(FF4h-FF5h)
ARTIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
TIMER and ADC peripherals	Interrupt vector #4	(FF0h-FF1h)

3.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A and B pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named. It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port C pins is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named . It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the ARTIMER peripheral and vector #4 with the A/D Converter or Timer peripherals.

Each on-chip peripheral has an associated interrupt request flag (A/D Converter, OVF, CPF and EF for the ARTIMER), which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (A/D Converter, OVIE and EIE for the ARTIMER), which must be set to “1” to enable the associated interrupt request.

3.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

3.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A, B) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port C) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 6. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A, B
	CLEARED	Falling edge mode on Port A, B
ESB	SET	Level sensitive mode on Port C
	CLEARED	Falling edge mode on Port C

3.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A, B-vector #1, Port C-vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

3.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

IINTERRUPTS (Cont'd)

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 16. Interrupt Processing Flow Chart

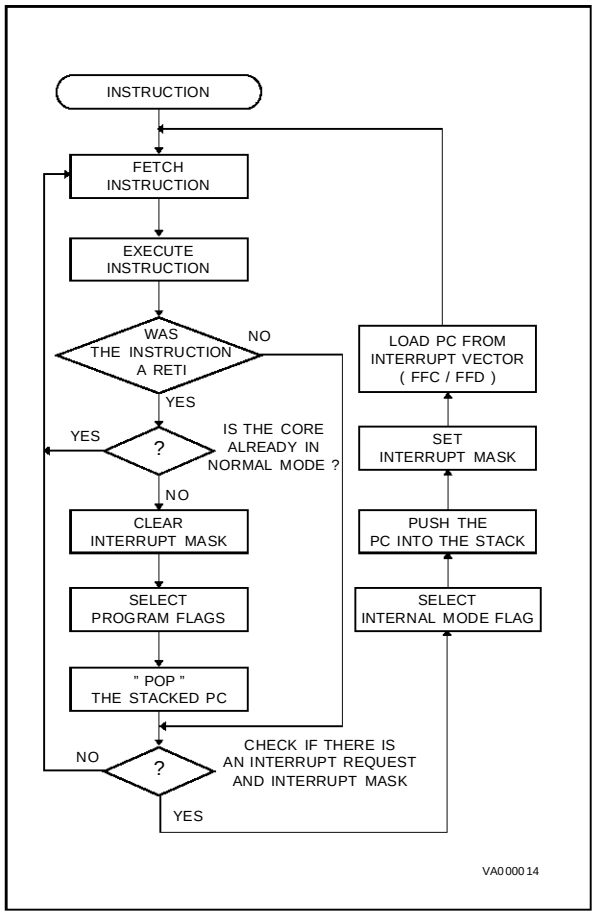
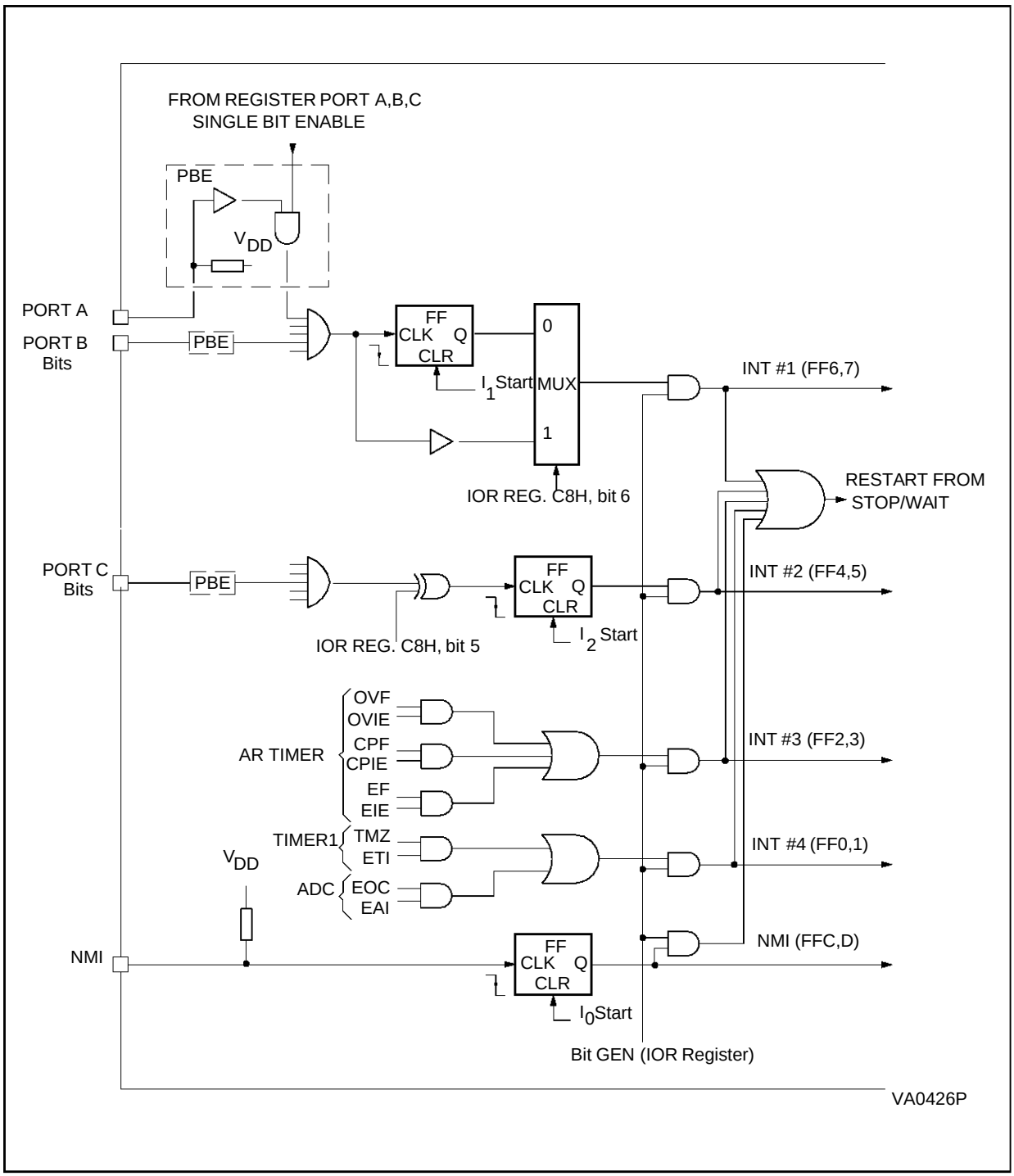


Table 7. Interrupt Requests and Mask Bits

Peripheral	Register	Address Register	Mask bit	Masked Interrupt Source	Interrupt vector
GENERAL	IOR	C8h	GEN	All Interrupts, excluding NMI	
TIMER	TSCR1	D4h	ETI	TMZ: TIMER Overflow	Vector 4
A/D CONVERTER	ADCR	D1h	EAI	EOC: End of Conversion	Vector 4
AR TIMER	ARMC	D5h	OVIE CPIE EIE	OVF: AR TIMER Overflow CPF: Successful compare EF: Active edge on ARTIMin	Vector 3
Port PAn	ORPA-DRPA	C0h-C4h	ORPAn-DRPAn	PAn pin	Vector 1
Port PBn	ORPB-DRPB	C1h-C5h	ORPBn-DRPBn	PBn pin	Vector 1
Port PCn	ORPC-DRPC	C2h-C6h	ORPCn-DRPCn	PCn pin	Vector 2

INTERRUPTS (Cont'd)

Figure 17. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

3.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection.

3.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

3.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

3.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

The MCU features 13 Input/Output lines which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Analog input (PA0-PA3, PC2-PC4)

– Artimer I/O lines : PB6-PB7

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PB0-PB3, PB6-PB7)

The lines are organized as three Ports (A, B and C).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The three DATA registers (DRA, DRB and DRC), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

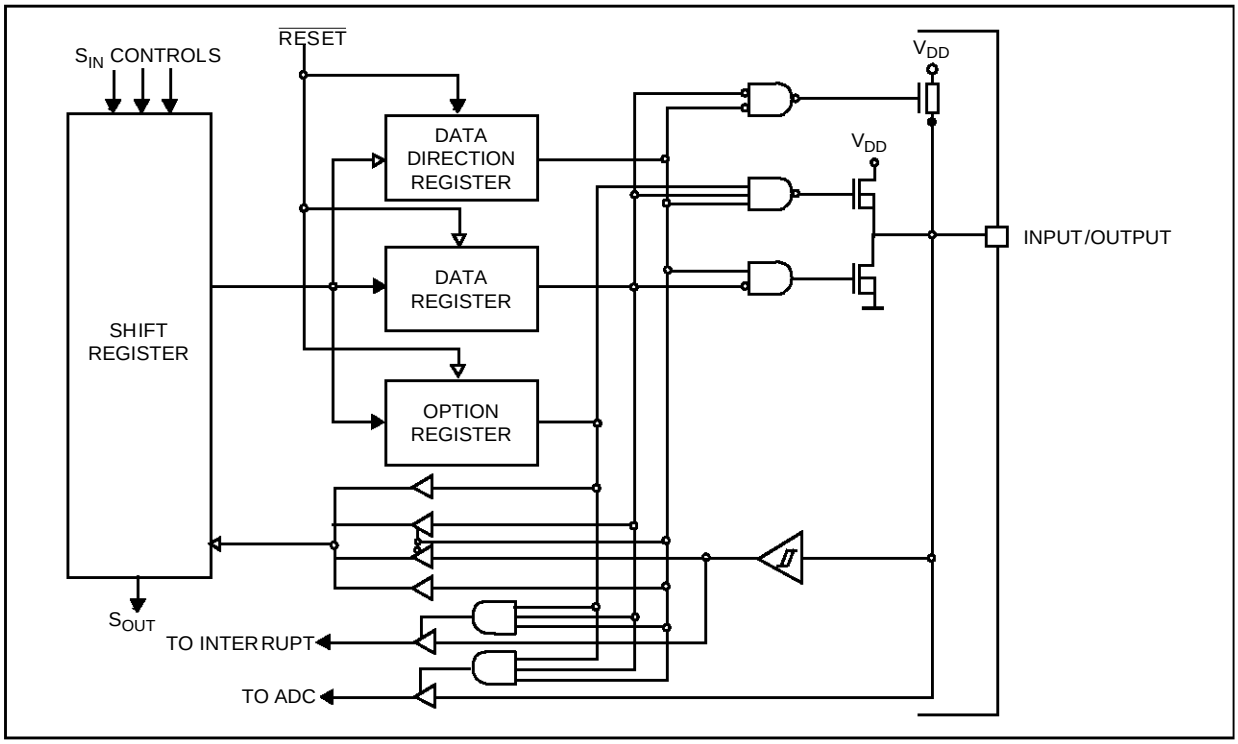
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The three Data Direction registers (DDRA, DDRB AND DRC) allow the data direction (input or output) of each pin to be set.

The three Option registers (ORA, ORB and ORC) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 18. I/O Port Block Diagram



I/O PORTS (Cont'd)

4.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 8 illustrates the various port configurations which can be selected by user software.

4.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

4.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A and B are AND-connected to the interrupt associated with Vector #1. The pins of Port care AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

4.1.1.3 Analog Input Options

The seven pins, PA0-PA3, PC2-PC4, can be configured as analog inputs by programming the OR and DR registers accordingly. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

4.1.2 I/O Port Option Registers

ORA/B/C (CCh PA, CDh PB, CEh PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Option Register bits.*

4.1.3 I/O Port Data Direction Registers

DDRA/B/C (C4h PA, C5h PB, C6h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Direction Registers bits.*

4.1.4 I/O Port Data Registers

DRA/B/C (C0h PA, C1h PB, C2h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Registers bits.*

Table 8. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Input	No pull-up, no interrupt (PB0-PB3,PB6-PB7)
			Input	Analog input (PA0-PA3, PC2-PC4)
1	0	X	Output	20mA sink open-drain output (PB0-PB3,PB6-PB7)
1	0	X	Output	Standard open-drain output (PA0-PA3, PC2-PC4)
1	1	X	Output	20mA sink push-pull output (PB0-PB3,PB6-PB7)

Note: X = Don't care

I/O PORTS (Cont'd)

4.1.5 AR Timer Alternate function Option

When bit PWMOE of register ARMC is low, pin ARTIMout/PB7 is configured as any standard pin of port B through the port registers. When PWMOE is high, ARTIMout/PB7 is the PWM output, independently of the port registers configuration.

ARTIMin/PB6 is connected to the AR Timer input. It is configured through the port registers as any standard pin of port B. To use ARTIMin/PB6 as AR Timer input, it must be configured as input through DDRB.

4.1.6 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 19. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A, B and C Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole (8-bit) port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

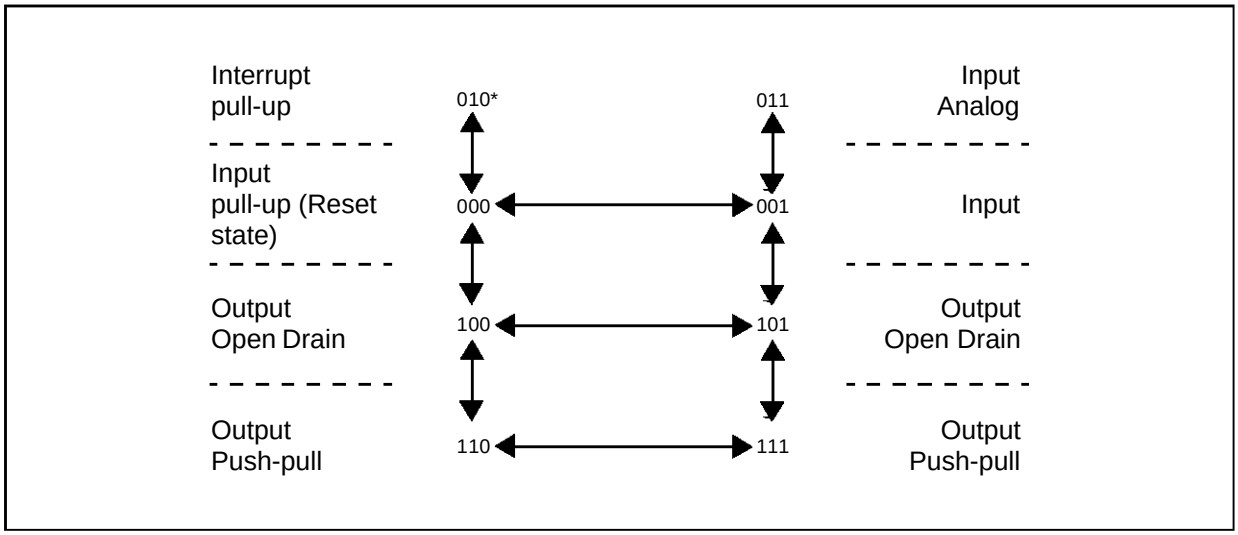
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

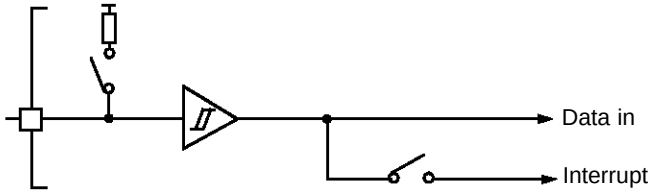
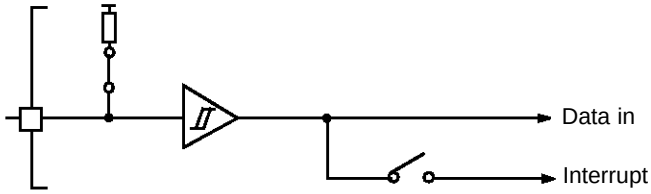
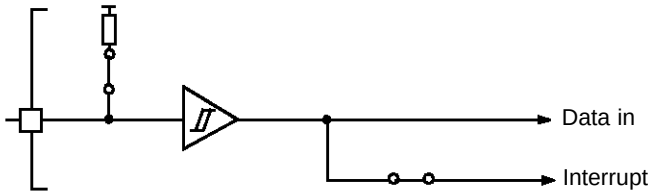
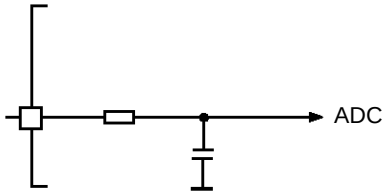
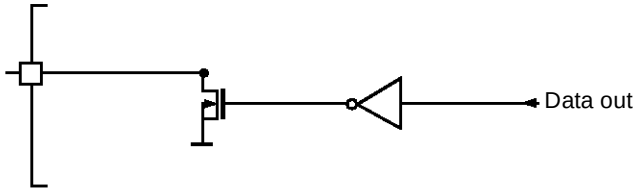
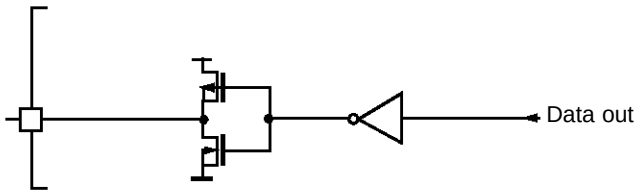
Figure 19. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 9. I/O Port Option Selections

MODE	AVAILABLE ON(1)	SCHEMATIC
Input	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Input with pull up	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Input with pull up with interrupt	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Analog Input	PA0-PA3 PC2-PC4	
Open drain output 5mA Open drain output 20mA	PA0-PA3 PC2-PC4 PB0-PB3, PB6-PB7	
Push-pull output 5mA Push-pull output 20mA	PA0-PA3 PC2-PC4 PB0-PB3, PB6-PB7	

Note 1. Provided the correct configuration has been selected.

4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}.

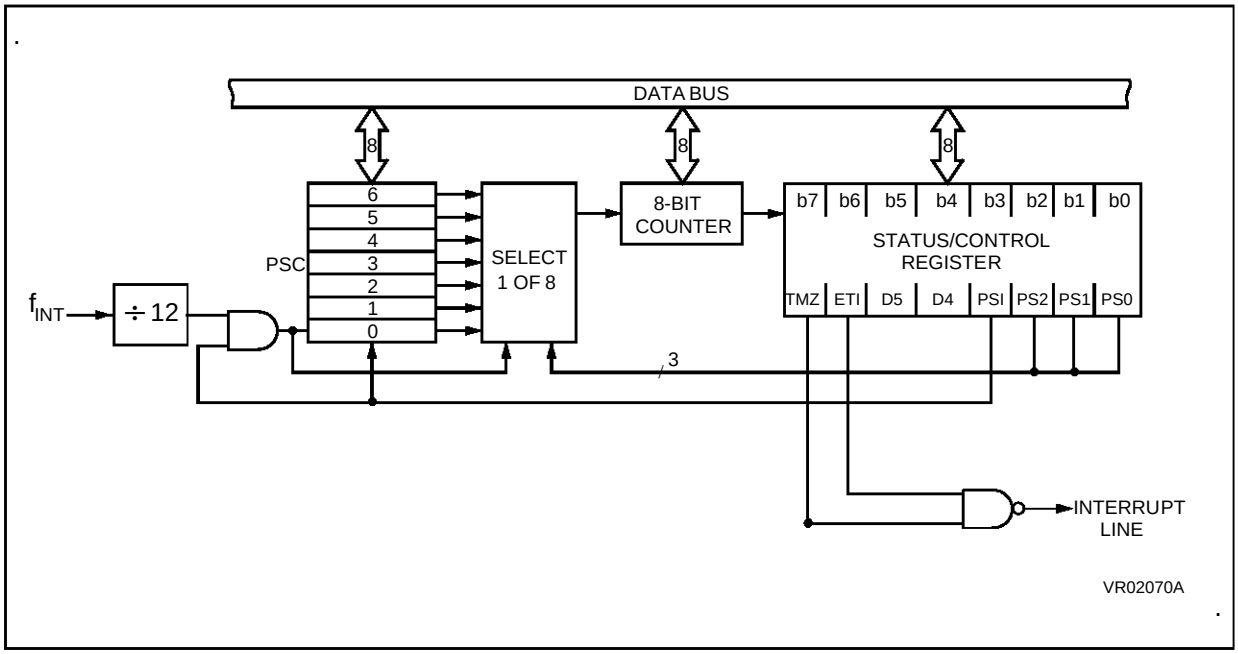
Figure 20 shows the Timer Block Diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero)bit in the TSCR is set. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set, an interrupt request is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input is the internal frequency (f_{INT}) divided by 12. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 10), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to allow the prescaler (and hence the counter) to start. If it is cleared, all the prescaler bits are set and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set. The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 21 illustrates the Timer's working principle.

Figure 20. Timer Block Diagram



TIMER (Cont'd)

4.2.1 Timer Operation

The Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high.

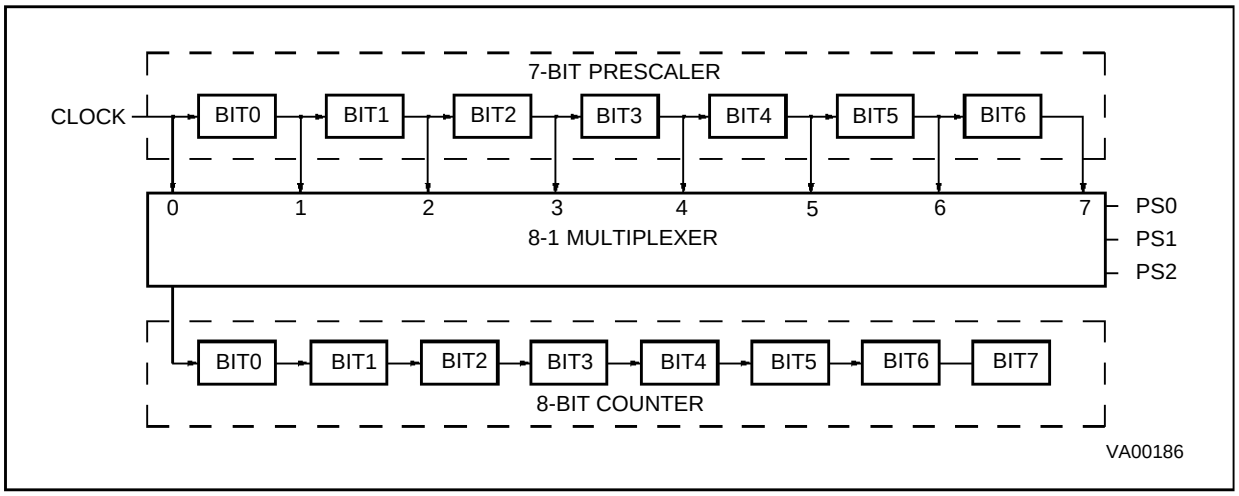
4.2.2 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

4.2.3 Application Notes

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

Figure 21. Timer Working Principle



TIMER (Cont'd)

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

4.2.4 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	D5	D4	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrup*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **D5**: *Reserved*

Must be reset.

Bit 4 = **D4**

When set, the timer is enabled; when reset the timer is disabled.

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 10. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register (TCR)

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

4.3 AUTO-RELOAD TIMER

The Auto-Reload Timer (AR Timer) on-chip peripheral consists of an 8-bit timer/counter with compare and capture/reload capabilities and of a 7-bit prescaler with a clock multiplexer, enabling the clock input to be selected as f_{INT}, $f_{INT/3}$ or an external clock source. A Mode Control Register, ARMC, two Status Control Registers, ARSC0 and ARSC1, an output pin, ARTIMout, and an input pin, ARTIMin, allow the Auto-Reload Timer to be used in 4 modes:

– Auto-reload (PWM generation),

– Output compare and reload on external event (PLL),

– Input capture and output compare for time measurement.

– Input capture and output compare for period measurement.

The AR Timer can be used to wake the MCU from WAIT mode either with an internal or with an external clock. It also can be used to wake the MCU from STOP mode, if used with an external clock signal connected to the ARTIMin pin. A Load register allows the program to read and write the counter on the fly.

4.3.1 AR Timer Description

The AR COUNTER is an 8-bit up-counter incremented on the input clock's rising edge. The counter is loaded from the ReLoad/Capture Register, ARRC, for auto-reload or capture operations, as well as for initialization. Direct access to the AR counter is not possible; however, by reading or writing the ARLR load register, it is possible to read or write the counter's contents on the fly.

The AR Timer's input clock can be either the internal clock (from the Oscillator Divider), the internal clock divided by 3, or the clock signal connected to the ARTIMin pin. Selection between these clock sources is effected by suitably programming bits CC0-CC1 of the ARSC1 register. The output of the AR Multiplexer feeds the 7-bit programmable AR Prescaler, ARPSC, which selects one of the 8 available taps of the prescaler, as defined by PSC0-PSC2 in the AR Mode Control Register. Thus the division factor of the prescaler can be set to 2n (where n = 0, 1,..7).

The clock input to the AR counter is enabled by the TEN (Timer Enable) bit in the ARMC register. When TEN is reset, the AR counter is stopped and the prescaler and counter contents are frozen. When TEN is set, the AR counter runs at the rate of the selected clock source. The counter is cleared on system reset.

The AR counter may also be initialized by writing to the ARLR load register, which also causes an immediate copy of the value to be placed in the AR counter, regardless of whether the counter is running or not. Initialization of the counter, by either method, will also clear the ARPSC register, whereupon counting will start from a known value.

4.3.2 Timer Operating Modes

Four different operating modes are available for the AR Timer:

Auto-reload Mode with PWM Generation. This mode allows a Pulse Width Modulated signal to be generated on the ARTIMout pin with minimum Core processing overhead.

The free running 8-bit counter is fed by the prescaler's output, and is incremented on every rising edge of the clock signal.

When a counter overflow occurs, the counter is automatically reloaded with the contents of the Reload/Capture Register, ARCC, and ARTIMout is set. When the counter reaches the value contained in the compare register (ARCP), ARTIMout is reset.

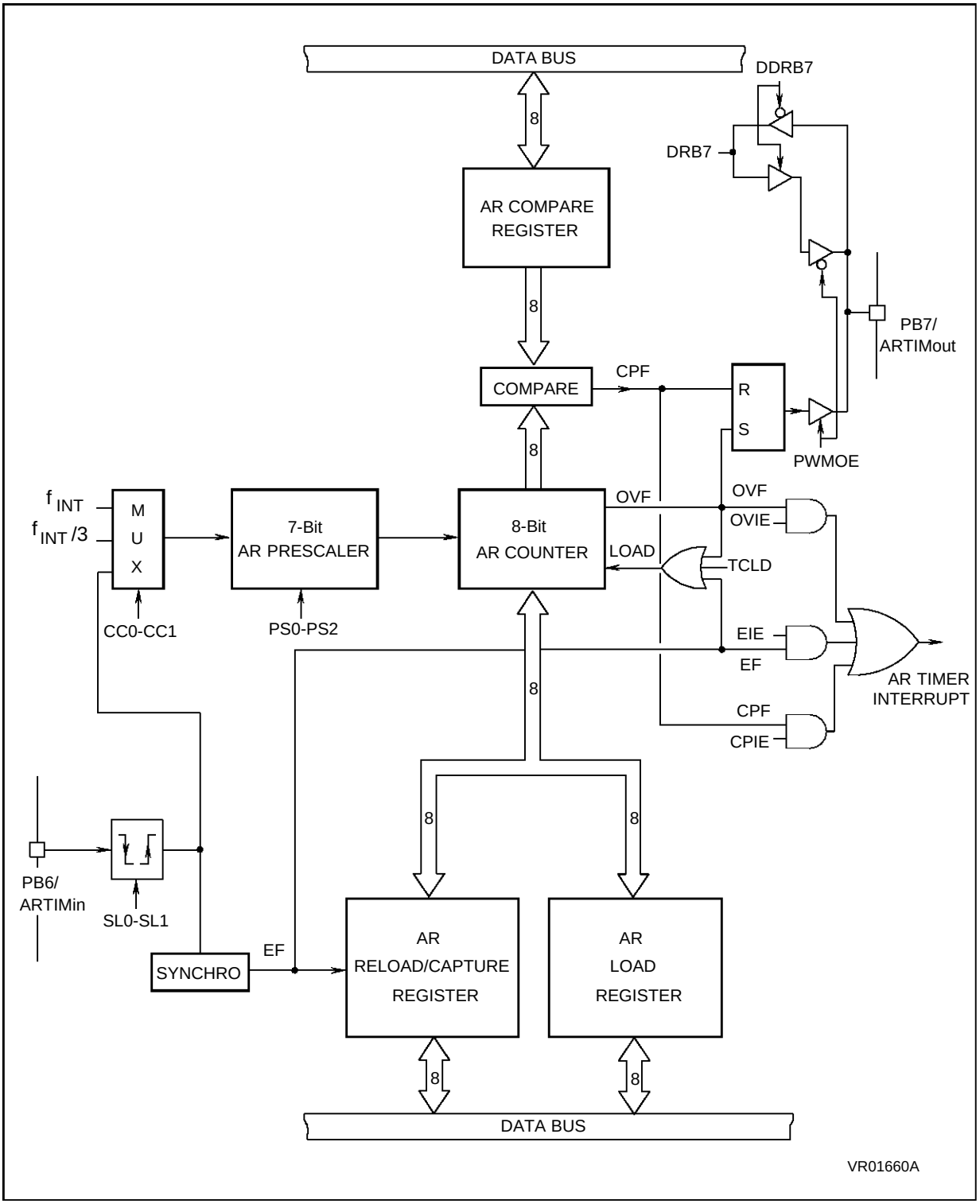
On overflow, the OVF flag of the ARSC0 register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OVIE, in the Mode Control Register (ARMC), is set. The OVF flag must be reset by the user software.

When the counter reaches the compare value, the CPF flag of the ARSC0 register is set and a compare interrupt request is generated, if the Compare Interrupt enable bit, CPIE, in the Mode Control Register (ARMC), is set. The interrupt service routine may then adjust the PWM period by loading a new value into ARCP. The CPF flag must be reset by user software.

The PWM signal is generated on the ARTIMout pin (refer to the Block Diagram). The frequency of this signal is controlled by the prescaler setting and by the auto-reload value present in the Reload/Capture register, ARRC. The duty cycle of the PWM signal is controlled by the Compare Register, ARCP.

AUTO-RELOAD TIMER (Cont'd)

Figure 22. AR Timer Block Diagram



AUTO-RELOAD TIMER (Cont'd)

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of PWM output signal. To obtain a signal on ARTIMout, the contents of the ARCP register must be greater than the contents of the ARRC register.

The maximum available resolution for the ARTIMout duty cycle is:

Resolution = 1/[255-(ARRC)]

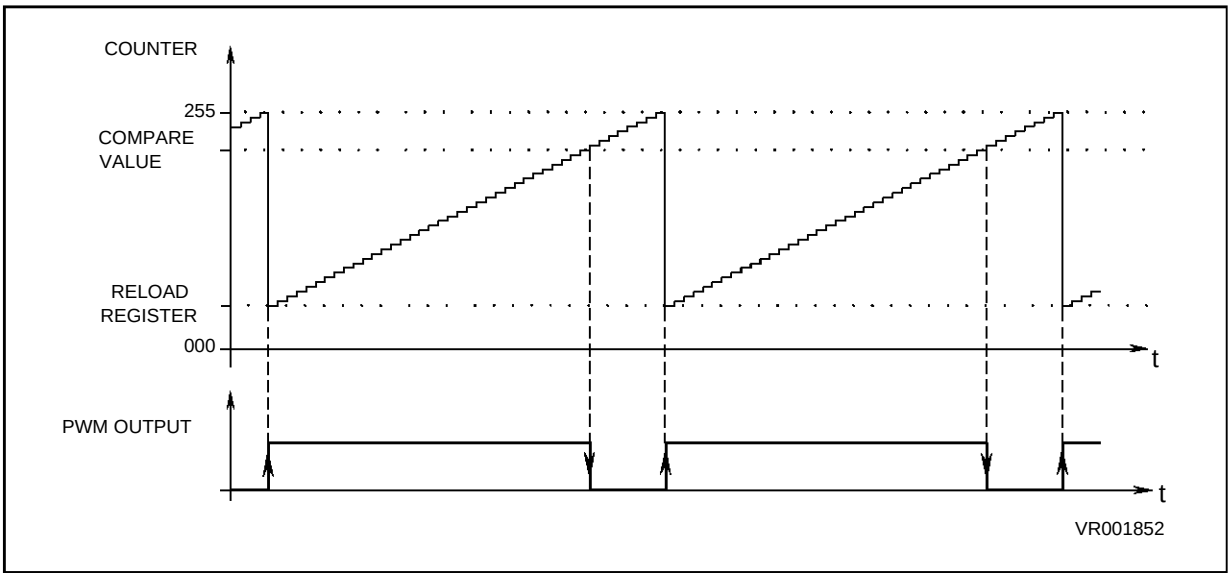
Where ARRC is the content of the Reload/Capture register. The compare value loaded in the Compare Register, ARCP, must be in the range from (ARRC) to 255.

The ARTC counter is initialized by writing to the ARRC register and by then setting the TCLD (Timer Load) and the TEN (Timer Clock Enable) bits in the Mode Control register, ARMC.

Enabling and selection of the clock source is controlled by the CC0, CC1, SL0 and SL1 bits in the Status Control Register, ARSC1. The prescaler division ratio is selected by the PS0, PS1 and PS2 bits in the ARSC1 register.

In Auto-reload Mode, any of the three available clock sources can be selected: Internal Clock, Internal Clock divided by 3 or the clock signal present on the ARTIMin pin.

Figure 23. Auto-reload Timer PWM Function



AUTO-RELOAD TIMER (Cont'd)

Capture Mode with PWM Generation. In this mode, the AR counter operates as a free running 8-bit counter fed by the prescaler output. The counter is incremented on every clock rising edge.

An 8-bit capture operation from the counter to the ARRC register is performed on every active edge on the ARTIMin pin, when enabled by Edge Control bits SL0, SL1 in the ARSC1 register. At the same time, the External Flag, EF, in the ARSC0 register is set and an external interrupt request is generated if the External Interrupt Enable bit, EIE, in the ARMC register, is set. The EF flag must be reset by user software.

Each ARTC overflow sets ARTIMout, while a match between the counter and ARCP (Compare Register) resets ARTIMout and sets the compare flag, CPF. A compare interrupt request is generated if the related compare interrupt enable bit, CPIE, is set. A PWM signal is generated on ARTIMout. The CPF flag must be reset by user software.

The frequency of the generated signal is determined by the prescaler setting. The duty cycle is determined by the ARCP register.

Initialization and reading of the counter are identical to the auto-reload mode (see previous description).

Enabling and selection of clock sources is controlled by the CC0 and CC1 bits in the AR Status Control Register, ARSC1.

The prescaler division ratio is selected by programming the PS0, PS1 and PS2 bits in the ARSC1 Register.

In Capture mode, the allowed clock sources are the internal clock and the internal clock divided by 3; the external ARTIMin input pin should not be used as a clock source.

Capture Mode with Reset of counter and prescaler, and PWM Generation. This mode is identical to the previous one, with the difference that a capture condition also resets the counter and the prescaler, thus allowing easy measurement of the time between two captures (for input period measurement on the ARTIMin pin).

Load on External Input. The counter operates as a free running 8-bit counter fed by the prescaler. the count is incremented on every clock rising edge.

Each counter overflow sets the ARTIMout pin. A match between the counter and ARCP (Compare Register) resets the ARTIMout pin and sets the compare flag, CPF. A compare interrupt request is generated if the related compare interrupt enable bit, CPIE, is set. A PWM signal is generated on ARTIMout. The CPF flag must be reset by user software.

Initialization of the counter is as described in the previous paragraph. In addition, if the external ARTIMin input is enabled, an active edge on the input pin will copy the contents of the ARRC register into the counter, whether the counter is running or not.

Notes:

The allowed AR Timer clock sources are the following:

AR Timer Mode	Clock Sources
Auto-reload mode	f_{INT}, $f_{INT/3}$, ARTIMin
Capture mode	f_{INT}, $f_{INT/3}$
Capture/Reset mode	f_{INT}, $f_{INT/3}$
External Load mode	f_{INT}, $f_{INT/3}$

The clock frequency should not be modified while the counter is counting, since the counter may be set to an unpredictable value. For instance, the multiplexer setting should not be modified while the counter is counting.

Loading of the counter by any means (by auto-reload, through ARLR, ARRC or by the Core) resets the prescaler at the same time.

Care should be taken when both the Capture interrupt and the Overflow interrupt are used. Capture and overflow are asynchronous. If the capture occurs when the Overflow Interrupt Flag, OVF, is high (between counter overflow and the flag being reset by software, in the interrupt routine), the External Interrupt Flag, EF, may be cleared simultaneusly without the interrupt being taken into account.

The solution consists in resetting the OVF flag by writing 03h in the ARSC0 register. The value of EF is not affected by this operation. If an interrupt has occured, it will be processed when the MCU exits from the interrupt routine (the second interrupt is latched).

AUTO-RELOAD TIMER (Cont'd)

4.3.3 AR Timer Registers

AR Mode Control Register (ARMC)

Address: D5h — Read/Write

Reset status: 00h

7							0
TCLD	TEN	PWMOE	EIE	CPIE	OVIE	ARMC1	ARMC0

The AR Mode Control Register ARMC is used to program the different operating modes of the AR Timer, to enable the clock and to initialize the counter. It can be read and written to by the Core and it is cleared on system reset (the AR Timer is disabled).

Bit 7 = **TLCD**: *Timer Load Bit.* This bit, when set, will cause the contents of ARRC register to be loaded into the counter and the contents of the prescaler register, ARPSC, are cleared in order to initialize the timer before starting to count. This bit is write-only and any attempt to read it will yield a logical zero.

Bit 6 = **TEN***: Timer Clock Enable.* This bit, when set, allows the timer to count. When cleared, it will stop the timer and freeze ARPSC and ARTSC.

Bit 5 = **PWMOE**: *PWM Output Enable.* This bit, when set, enables the PWM output on the ARTIMout pin. When reset, the PWM output is disabled.

Bit 4 = **EIE**: *External Interrupt Enable.* This bit, when set, enables the external interrupt request. When reset, the external interrupt request is masked. If EIE is set and the related flag, EF, in the ARSC0 register is also set, an interrupt request is generated.

Bit 3 = **CPIE**: *Compare Interrupt Enable.* This bit, when set, enables the compare interrupt request. If CPIE is reset, the compare interrupt request is masked. If CPIE is set and the related flag, CPF, in the ARSC0 register is also set, an interrupt request is generated.

Bit 2 = **OVIE**: *Overflow Interrupt.* This bit, when set, enables the overflow interrupt request. If OVIE is reset, the compare interrupt request is masked. If OVIE is set and the related flag, OVF in the ARSC0 register is also set, an interrupt request is generated.

Bit 1-0 = **ARMC1-ARMC0**: *Mode Control Bits 1-0.* These are the operating mode control bits. The following bit combinations will select the various operating modes:

ARMC1	ARMC0	Operating Mode
0	0	Auto-reload Mode
0	1	Capture Mode
1	0	Capture Mode with Reset of ARTC and ARPSC
1	1	Load on External Edge Mode

AR Timer Status/Control Registers ARSC0 & ARSC1. These registers contain the AR Timer status information bits and also allow the programming of clock sources, active edge and prescaler multiplexer setting.

ARSC0 register bits 0,1 and 2 contain the interrupt flags of the AR Timer. These bits are read normally. Each one may be reset by software. Writing a one does not affect the bit value.

AR Status Control Register 0 (ARSC0)

Address: D6h — Read/Clear

7							0
D7	D6	D5	D4	D3	EF	CPF	OVF

Bits 7-3 = **D7-D3**: *Unused*

Bit 2 = **EF**: *External Interrupt Flag.* This bit is set by any active edge on the external ARTIMin input pin. The flag is cleared by writing a zero to the EF bit.

Bit 1 = **CPF**: *Compare Interrupt Flag.* This bit is set if the contents of the counter and the ARCP register are equal. The flag is cleared by writing a zero to the CPF bit.

Bit 0 = **OVF**: *Overflow Interrupt Flag.* This bit is set by a transition of the counter from FFh to 00h (overflow). The flag is cleared by writing a zero to the OVF bit.

AUTO-RELOAD TIMER (Cont'd)

AR Status Control Register 1(ARSC1)

Address: D7h — Read/Write

7							0
PS2	PS1	PS0	D4	SL1	SL0	CC1	CC0

Bist 7-5 = **PS2-PS0**: *Prescaler Division Selection Bits 2-0.* These bits determine the Prescaler division ratio. The prescaler itself is not affected by these bits. The prescaler division ratio is listed in the following table:

Table 11. Prescaler Division Ratio Selection

PS2	PS1	PS0	ARPSC Division Ratio
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Bit 4 = **D4**: *Reserved*. Must be kept reset.

Bit 3-2 = **SL1-SL0**: *Timer Input Edge Control Bits 1-0.* These bits control the edge function of the Timer input pin for external synchronization. If bit SL0 is reset, edge detection is disabled; if set edge detection is enabled. If bit SL1 is reset, the AR Timer input pin is rising edge sensitive; if set, it is falling edge sensitive.

SL1	SL0	Edge Detection
X	0	Disabled
0	1	Rising Edge
1	1	Falling Edge

Bit 1-0 = **CC1-CC0**: *Clock Source Select Bit 1-0.* These bits select the clock source for the AR Timer through the AR Multiplexer. The programming of the clock sources is explained in the following Table 12:

Table 12. Clock Source Selection.

CC1	CC0	Clock Source
0	0	F_{int}
0	1	F_{int} Divided by 3
1	0	ARTIMin Input Clock
1	1	Reserved

AR Load Register ARLR. The ARLR load register is used to read or write the ARTC counter register "on the fly" (while it is counting). The ARLR register is not affected by system reset.

AR Load Register (ARLR)

Address: DBh — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Load Register Data Bits.* These are the load register data bits.

AR Reload/Capture Register. The ARRC reload/capture register is used to hold the auto-reload value which is automatically loaded into the counter when overflow occurs.

AR Reload/Capture (ARRC)

Address: D9h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Reload/Capture Data Bits*. These are the Reload/Capture register data bits.

AR Compare Register. The CP compare register is used to hold the compare value for the compare function.

AR Compare Register (ARCP)

Address: DAh — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Compare Data Bits*. These are the Compare register data bits.

4.4 A/D CONVERTER (ADC)

The A/D converter peripheral is an 8-bit analog to digital converter with analog inputs as alternate I/O functions (the number of which is device dependent), offering 8-bit resolution with a typical conversion time of 70us (at an oscillator clock frequency of 8MHz).

The ADC converts the input voltage by a process of successive approximations, using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, conversion accuracy is decreased.

Selection of the input pin is done by configuring the related I/O line as an analog input via the Option and Data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously, to avoid device malfunction.

The ADC uses two registers in the data space: the ADC data conversion register, ADR, which stores the conversion result, and the ADC control register, ADCR, used to program the ADC functions.

A conversion is started by writing a "1" to the Start bit (STA) in the ADC control register. This automatically clears (resets to "0") the End Of Conversion Bit (EOC). When a conversion is complete, the EOC bit is automatically set to "1", in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

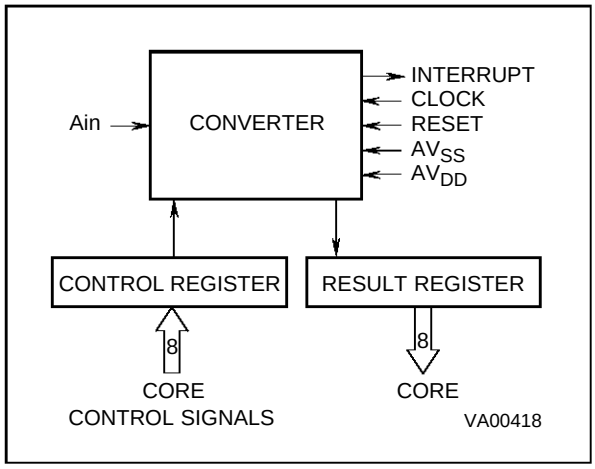
The STA bit is continuously scanned so that, if the user sets it to "1" while a previous conversion is in progress, a new conversion is started before completing the previous one. The start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter features a maskable interrupt associated with the end of conversion. This interrupt is associated with interrupt vector #4 and occurs when the EOC bit is set (i.e. when a conversion is completed). The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. This is done by setting the PDS bit in the ADC control register to "0". If PDS="1", the A/D is powered and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow stabilisation of the A/D converter. This action is also needed before entering WAIT mode, since the A/D comparator is not automatically disabled in WAIT mode.

During Reset, any conversion in progress is stopped, the control register is reset to 40h and the ADC interrupt is masked (EAI=0).

Figure 24. ADC Block Diagram



4.4.1 Application Notes

The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for the optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 12pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. In worst case conditions, the impedance, ASI, of the analog voltage source is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband. ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

A/D CONVERTER (Cont'd)

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

The accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

The converter resolution is given by::

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

In order to optimise conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the less significant bits are determined.

The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. Indeed, only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. It should be noted that waking up the microcontroller could also be done using the Timer interrupt, but in this case the Timer will be working and the resulting noise could affect conversion accuracy.

A/D Converter Control Register (ADCR)

Address: 0D1h — Read/Write

7							0
EAI	EOC	STA	PDS	D3	D2	D1	D0

Bit 7 = **EAI**: *Enable A/D Interrupt.* If this bit is set to "1" the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

Bit 6 = **EOC**: *End of conversion. Read Only.* This read only bit indicates when a conversion has been completed. This bit is automatically reset to "0" when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to "1".

Bit 5 = **STA***: Start of Conversion. Write Only.* Writing a "1" to this bit will start a conversion on the selected channel and automatically reset to "0" the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

Bit 4 = **PDS***: Power Down Selection.* This bit activates the A/D converter if set to "1". Writing a "0" to this bit will put the ADC in power down mode (idle mode).

Bit 3-0 = **D3-D0.** Not used

A/D Converter Data Register (ADR)

Address: 0D0h — Read only

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0***: 8 Bit A/D Conversion Result.*

5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 13. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 14. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 15. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected. The tested bit is shifted into carry.
* . Not Affected

Table 16. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not<M> Affected

Table 17. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ . Affected
*. Not Affected

Table 18. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
* . Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

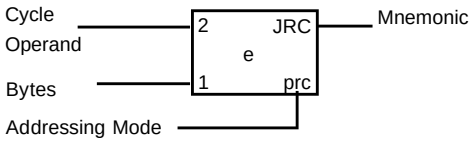
LOW / HI	**0** **0000**	**1** **0001**	**2** **0010**	**3** **0011**	**4** **0100**	**5** **0101**	**6** **0110**	**7** **0111**	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1** **0001**
2 **0010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2** **0010**
3 **0011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3** **0011**
4 **0100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4** **0100**
5 **0101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5** **0101**
6 **0110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 INC (x) 1 ind	**6** **0110**
7 **0111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	# 	**7** **0111**
8 **1000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD (x),a 1 ind	**8** **1000**
9 **1001**	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	# 	**9** **1001**
A **1010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A** **1010**
B **1011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B** **1011**
C **1100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C** **1100**
D **1101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D** **1101**
E **1110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 DEC (x) 1 ind	**E** **1110**
F **1111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	# 	**F** **1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

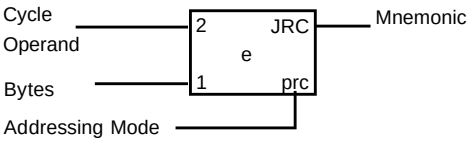
LOW / HI	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

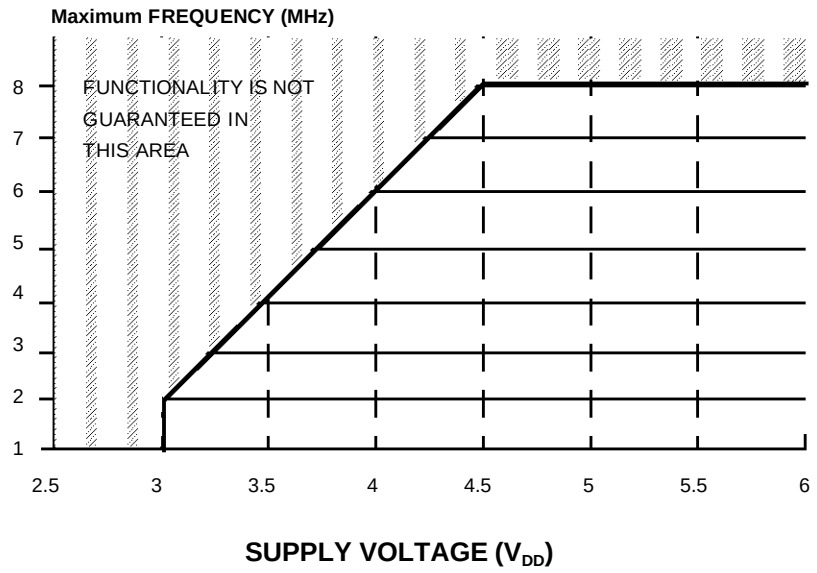
THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	°C/W
		PSO20			80	

6.2 RECOMMENDED OPERATING CONDITIONS

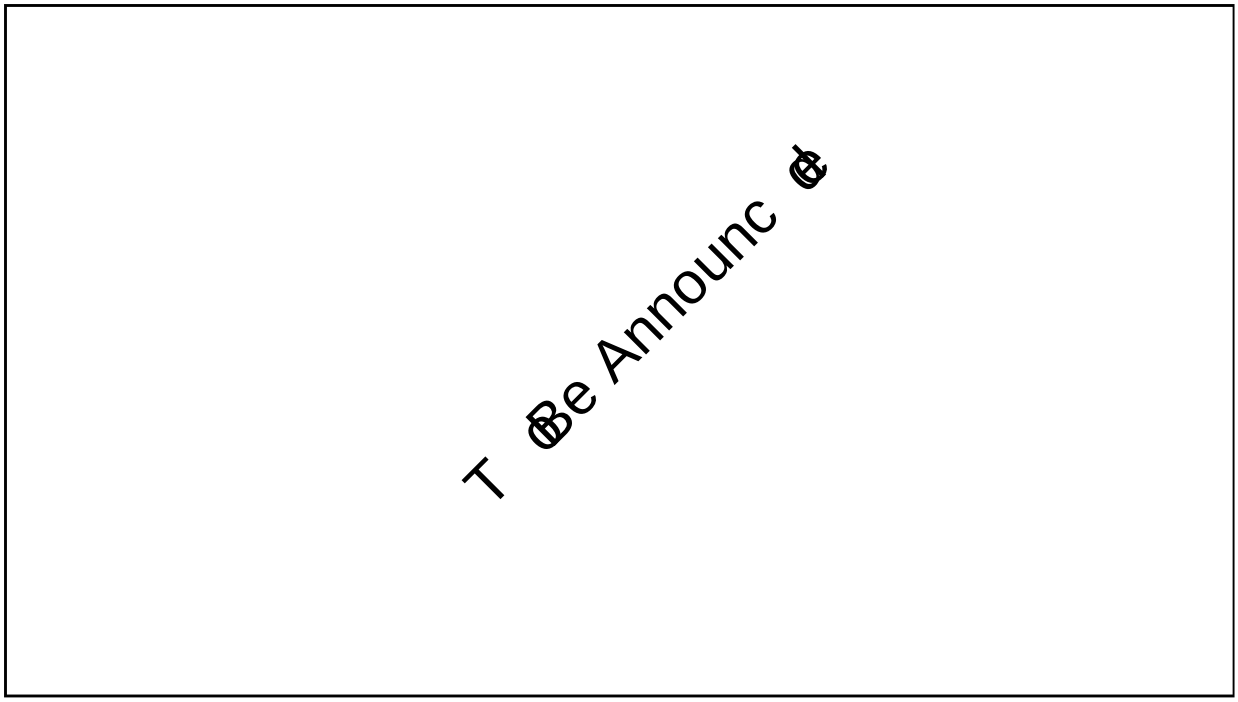
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input	V_{DD} = 4.5 to 5.5V			-5	mA

Figure 25. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



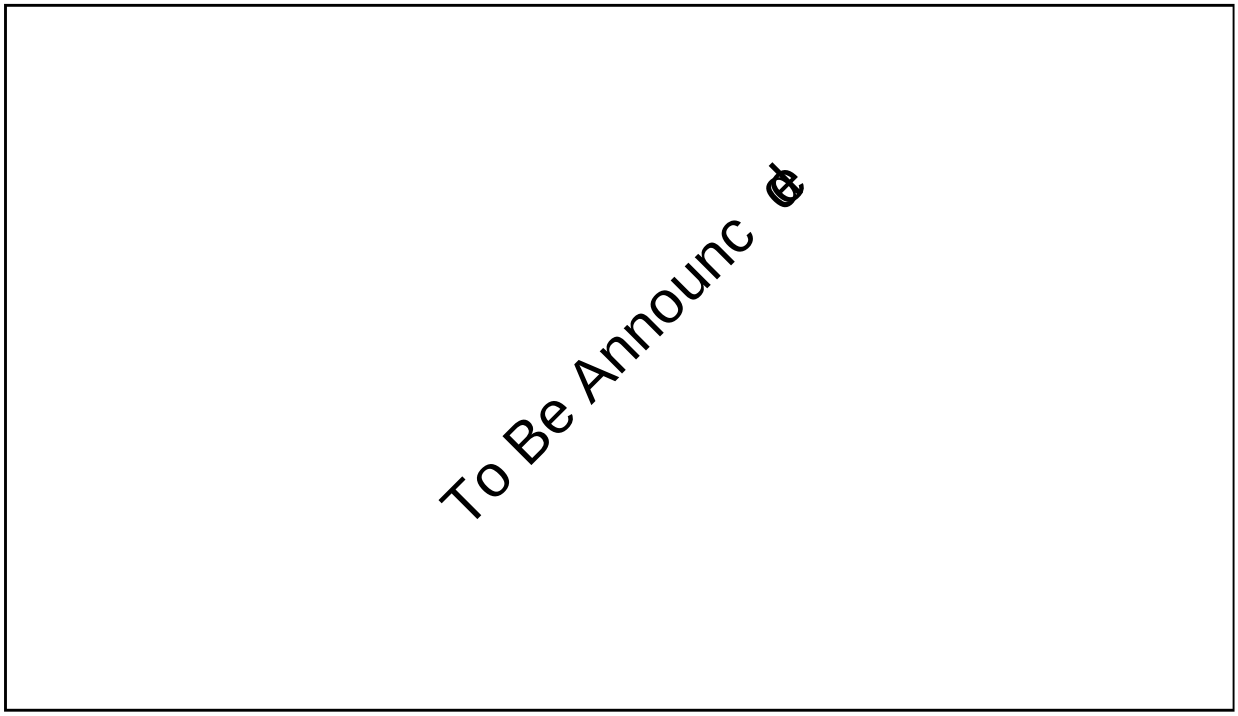
The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 26. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

Figure 27. RC Oscillator. F_{INT} versus RNET (Indicative Values)

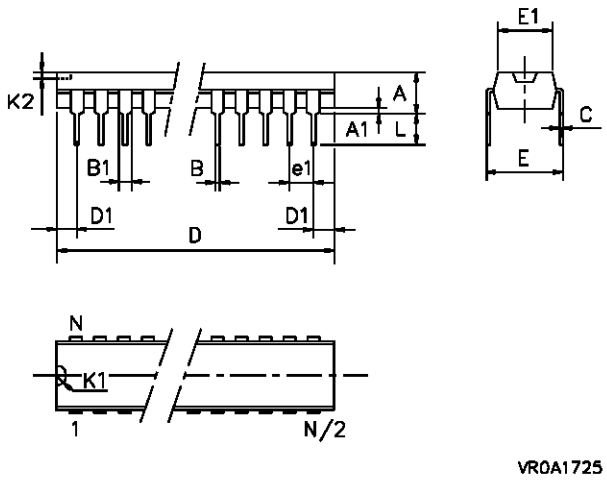


The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

7 GENERAL INFORMATION

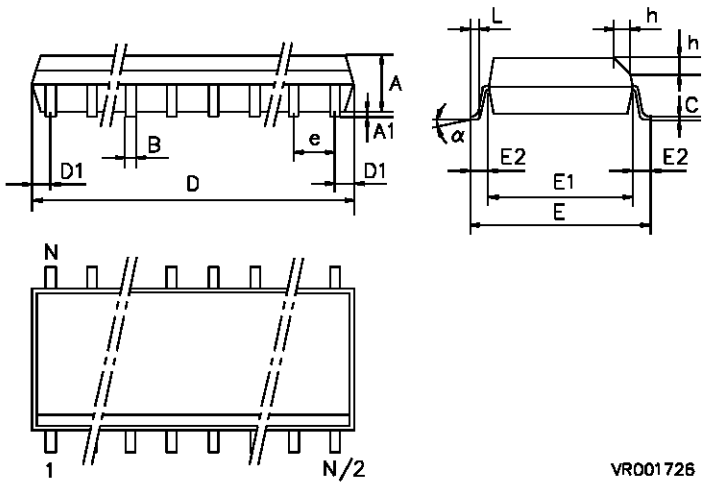
7.1 PACKAGE MECHANICAL DATA

Figure 28. 20-Pin Plastic Dual In-Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
A2	-	-	-	-	-	-
B		0.45			0.018	
B1	1.39		1.65	0.055		0.065
C		0.25			0.010	
D			25.4			1.000
e3		22.86			0.900	
E1			7.1			0.280
e		2.54			0.100	
eB		8.5			0.335	
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.3			0.130	
	Number of Pins					
N	20					

Figure 29. 20-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

PACKAGE MECHANICAL DATA (Cont'd)

7.2 .ORDERING INFORMATION

Table 19. OTP/EPROM VERSION ORDERING INFORMATION

Sales Type	Program Memory (Bytes)	I/O	Additional Features	Temperature Range	Package
ST62T53BB6/XXX	1836 (OTP)	13	A/D CONVERTER	-40 to + 85°C	PDIP20
ST62T53BM6/XXX					PSO20

Notes:

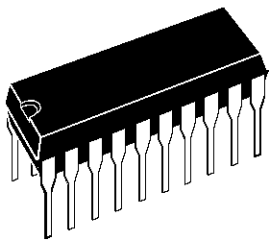
SGS-THOMSON MICROELECTRONICS

ST6253B

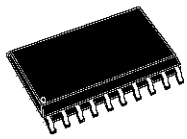
8-BIT MCUs WITH A/D CONVERTER, AUTO-RELOAD TIMER

PRODUCT PREVIEW

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- User ROM: 1836 bytes
- Data RAM: 64 bytes
- 13 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 6 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- 8-bit Auto-reload Timer with 7-bit programmable prescaler (AR Timer)
- Digital Watchdog
- 8-bit A/D Converter with 7 analog inputs
- On-chip Clock oscillator can be driven by Quartz Crystal Ceramic resonator or RC network
- User configurable Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU2 Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	RAM	I/O Pins
ST6253B	1836	64	13

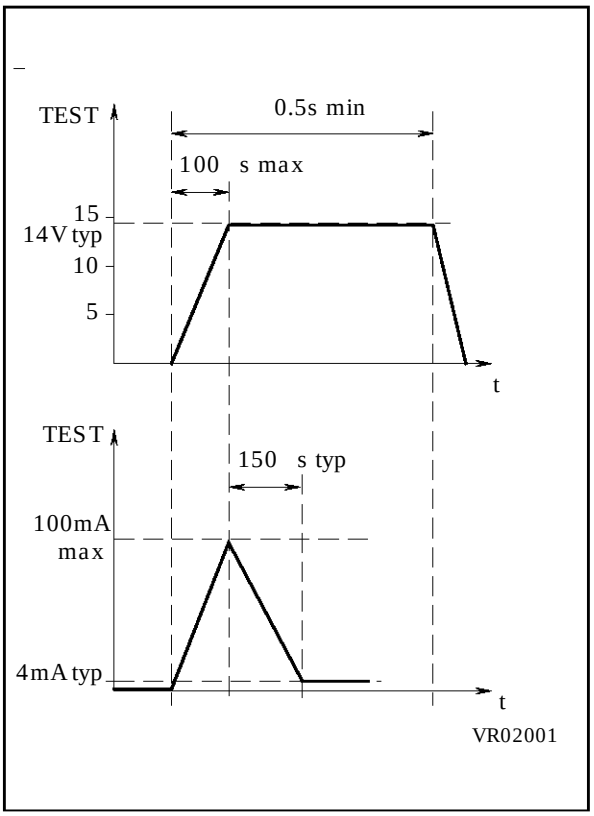
1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST6253B is a mask programmed ROM version of ST62T53B OTP device.

They offer the same functionality as OTP devices, selecting as ROM options the options defined in the programmable option byte of the OTP version.

Figure 1. Protection programming wave form

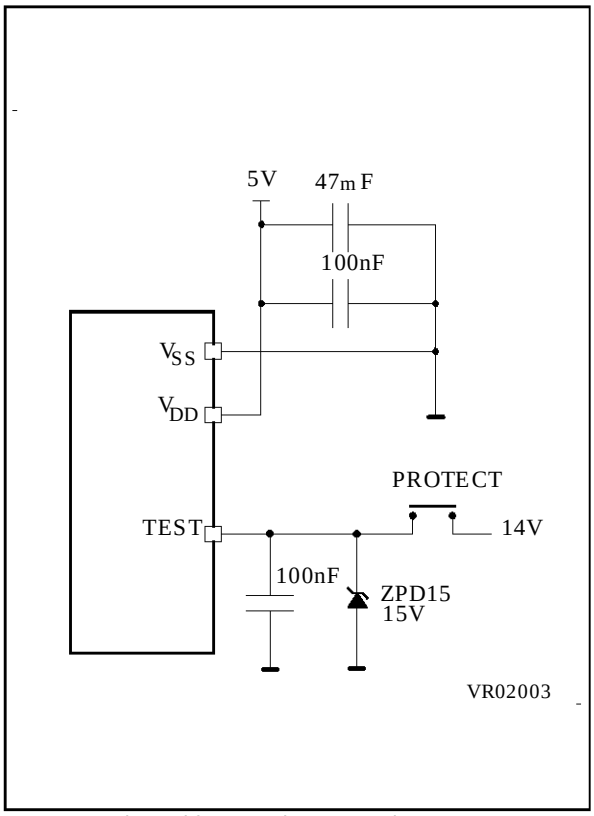


1.2 ROM READOUT PROTECTION

If the ROM READOUT PROTECTION option is selected, a protection fuse can be blown to prevent any access to the program memory content.

In case the user wants to blow this fuse, high voltage must be applied on the TEST pin.

Figure 2. Example of READOUT PROTECTION Fuse Programming Circuit



Note: ZPD15 is used for overvoltage protection

ST6253B MICROCONTROLLER OPTION LIST

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference . . . . . .

SGS-THOMSON Microelectronics references

Device: [ ] ST6253B

Package: [ ] Dual in Line Plastic [ ] Small Outline Plastic

In this case, select conditioning

[ ] Standard (Stick)

[ ] Tape & Reel

Temperature Range: [ ] 0°C to + 70°C [ ] - 40°C to + 85°C

Special Marking: [ ] No

[ ] Yes "_ _ _ _ _ _ _ _ _ _ "

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Maximum character count: DIP20: 10

SO20: 8

Oscillator Source Selection: [ ] Crystal Quartz/Ceramic resonator (Default)

[ ] RC Network

Watchdog Selection: [ ] Software Activation (STOP mode available)

[ ] Hardware Activation (no STOP mode)

Power on Reset Delay

[ ] 32768 cycle delay

[ ] 2048 cycle delay

ROM Readout Protection: [ ] Standard (Fuse cannot be blown)

[ ] Enabled (Fuse can be blown by the customer)

Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.

External STOP Mode Control

[ ] Enabled

[ ] Disabled (Default)

Comments :

Supply Operating Range in the application:

Oscillator Fequency in the application:

Notes . . . . . .

Signature . . . . . .

Date . . . . . .

1.3 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

1.3.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

1.3.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 1. ROM Memory Map for ST6253B

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 2. ROM version Ordering Information

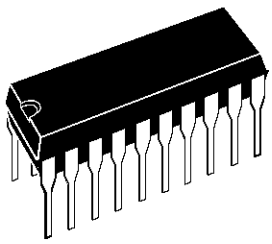
Sales Type	ROM	I/O	Additional Features	Temperature Range	Package
ST6253BB1/XXX ST6253BB6/XXX	1836 Bytes	13	A/D CONVERTER	0 to +70°C -40 to + 85°C	PDIP20
ST6253BM1/XXX ST6253BM6/XXX				0 to +70°C -40 to + 85°C	PSO20

SGS-THOMSON MICROELECTRONICS

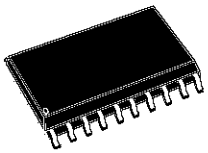
ST62T60B/E60B
ST62T63B

8-BIT MCUs WITH A/D CONVERTER, AUTO-RELOAD TIMER, EEPROM AND SPI

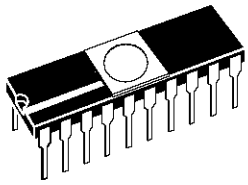
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- Data RAM: 64/128 bytes
- Data EEPROM: 64/128 bytes
- User Programmable Options
- 13 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 6 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- 8-bit Auto-reload Timer with 7-bit programmable prescaler (AR Timer)
- Digital Watchdog
- 8-bit A/D Converter with 7 analog inputs
- 8-bit Synchronous Peripheral Interface (SPI)
- On-chip Clock oscillator can be driven by Quartz Crystal Ceramic resonator or RC network
- User configurable Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU2 Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20



CDIP20W

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	OTP (Bytes)	EPROM (Bytes)	EEPROM	RAM	SPI	I/O Pins
ST62T60B	3884	-	128	128	yes	13
ST62T63B	1836	-	64	64	no	13
ST62E60B		3884	128	128	yes	13

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST62T60B, ST62T63B and ST62E60B 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.

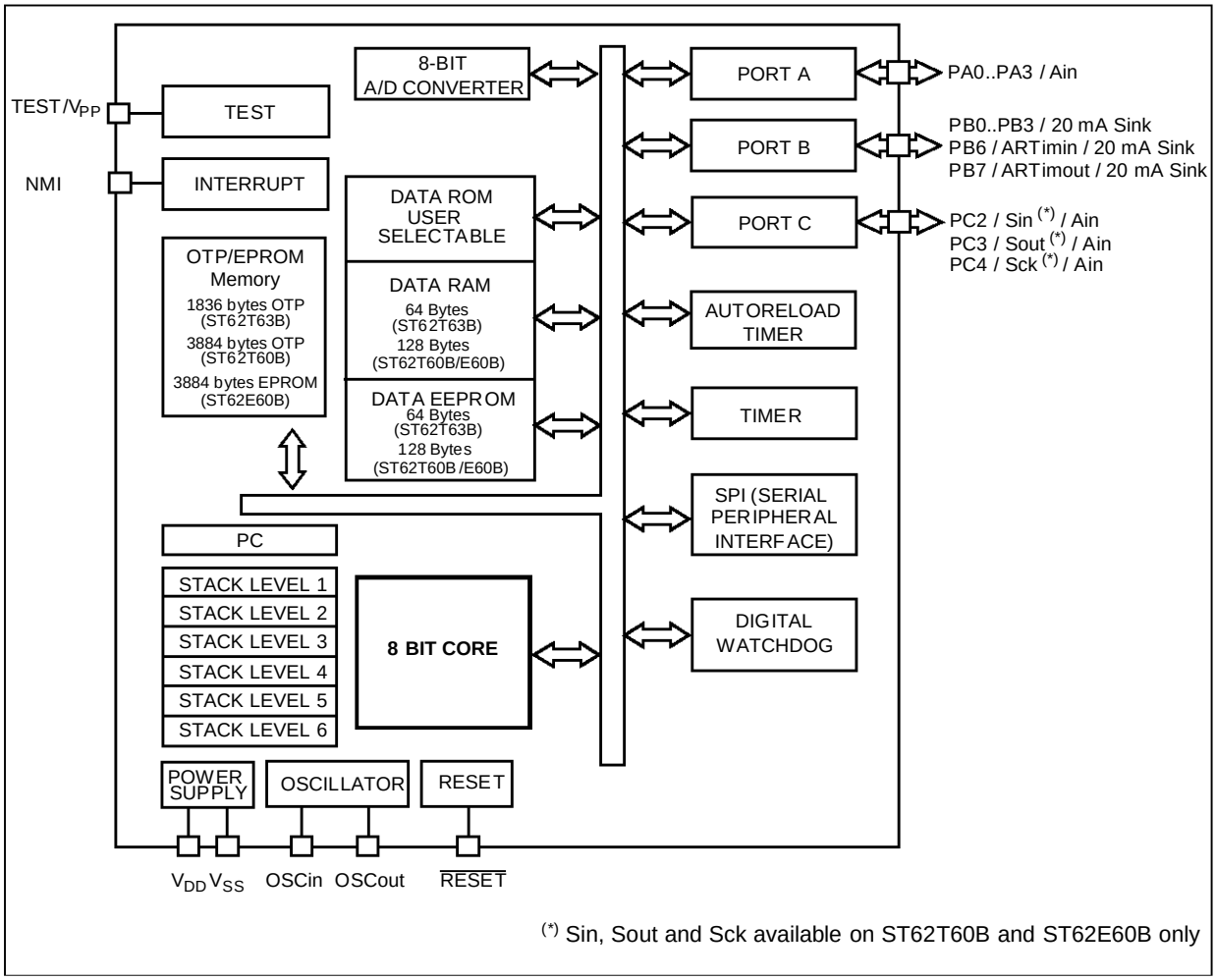
The ST62E60B is the erasable EPROM version of the ST62T60B device, which may be used to emulate the ST62T60B and T63B devices, as well as the respective ST6260B and ST6263B ROM devices.

OTP and EPROM devices are functionally identical. The ROM based versions offer the same functionality selecting as ROM options the options defined in the programmable option byte of the OTP/EPROM versions.

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.

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

Figure 1. Block Diagram



1.2 PIN DESCRIPTIONS

V_{DD} and V_{SS}. Power is supplied to the MCU via these two pins. V_{DD} is the power connection and V_{SS} is the ground connection.

OSCin and OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal, a ceramic resonator or an external clock signal can be connected between these two pins. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active-low $\overline{\text{RESET}}$ pin is used to restart the microcontroller.

TEST/V_{PP}. The TEST must be held at V_{SS} for normal operation. If TEST pin is connected to a +12.5V level during the reset phase, the EPROM/OTP programming Mode is entered.

NMI. The NMI pin provides the capability for asynchronous interruption, by applying an external non maskable interrupt to the MCU. The NMI input is falling edge sensitive. It is provided with an on-chip pullup resistor and Schmitt trigger characteristics.

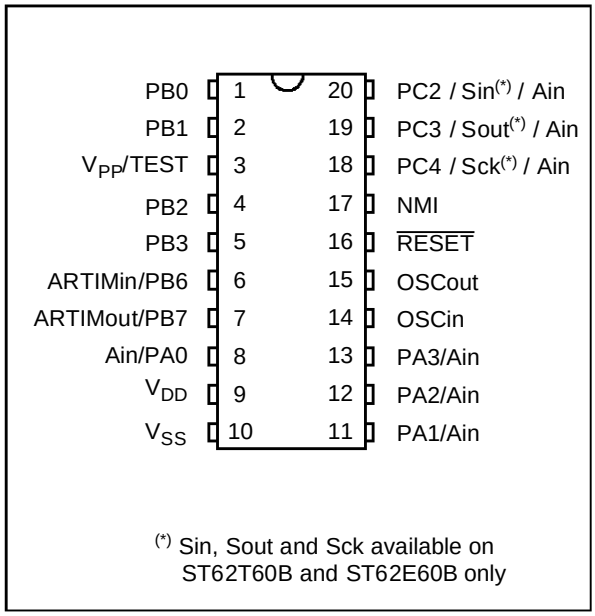
PA0-PA3. These 4 lines are organized as one I/O port (A). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs, analog inputs for the A/D converter.

PB0-PB3. These 4 lines are organized as one I/O port (B). Each line may be configured under software control as inputs with or without internal pull-up resistors, interrupt generating inputs with pull-up resistors, open-drain or push-pull outputs.
PB0-PB3 can also sink 20mA for direct LED driving.

PB6/ARTIMin, PB7/ARTIMout. These pins are either Port B I/O bits or the Input and Output pins of the AR TIMER. To be used as timer input function PB6 has to be programmed as input with or without pull-up. A dedicated bit in the AR TIMER Mode Control Register sets PB7 as timer output function.
PB6-PB7 can also sink 20mA for direct LED driving.

PC2-PC4. These 3 lines are organized as one I/O port (C). Each line may be configured under software control as input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, analog input for the A/D converter, open-drain or push-pull output.
On the ST62T60B and ST62E60B, PC2-PC4 can also be used as respectively Data in, Data out and Clock I/O pins for the on-chip SPI to carry the synchronous serial I/O signals.

Figure 2. ST62T60Band ST62E60B Pin Configuration



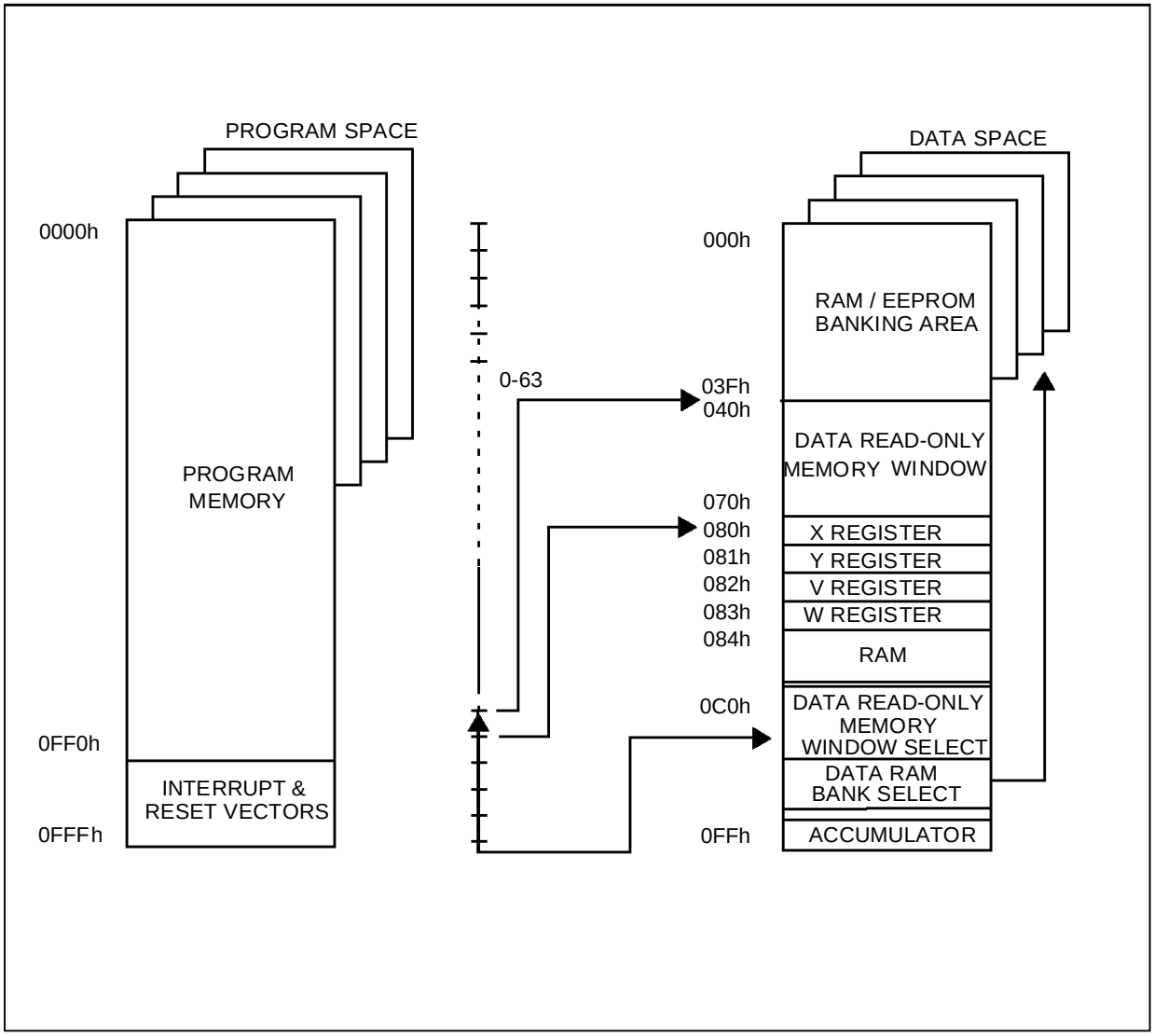
1.3 MEMORY MAP

1.3.1 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 subroutine and interrupt service routine nesting.

Figure 3. Memory Addressing Diagram



MEMORY MAP (Cont'd)

1.3.2 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).

1.3.2.1 Program Memory Protection

The Program Memory in OTP or EPROM devices can be protected against external readout of memory by selecting the READOUT PROTECTION option in the option byte.

In the EPROM parts, READOUT PROTECTION option can be disactivated only by U.V. erasure that also results into the whole EPROM context erasure.

Note: Once the Readout Protection is activated, it is no longer possible, even for SGS-THOMSON, to gain access to the OTP contents. Returned parts with a protection set can therefore not be accepted.

Figure 4. ST62E60B/T60B Program Memory Map

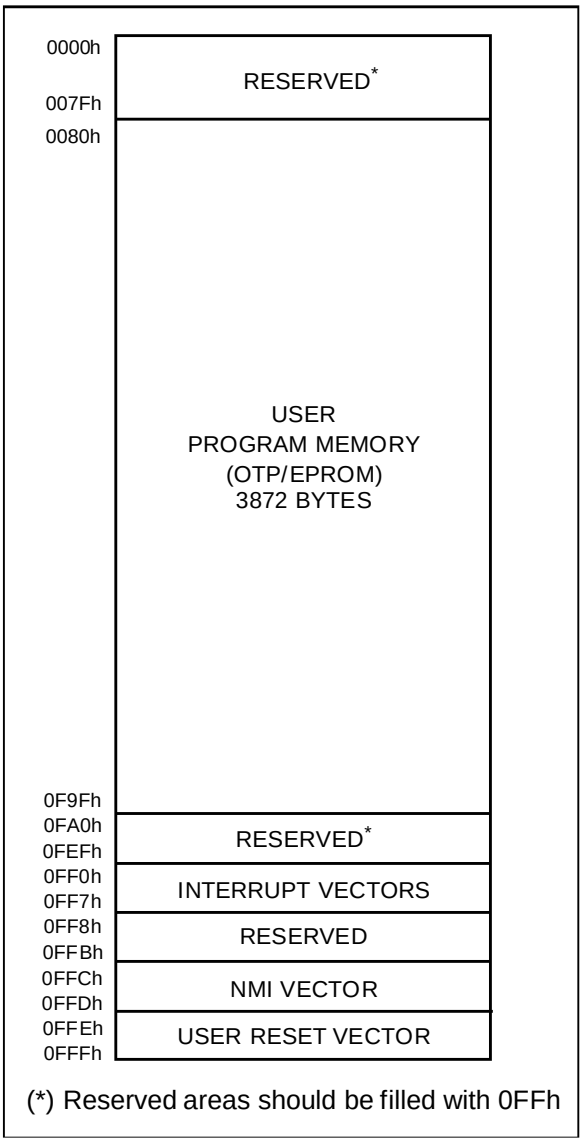
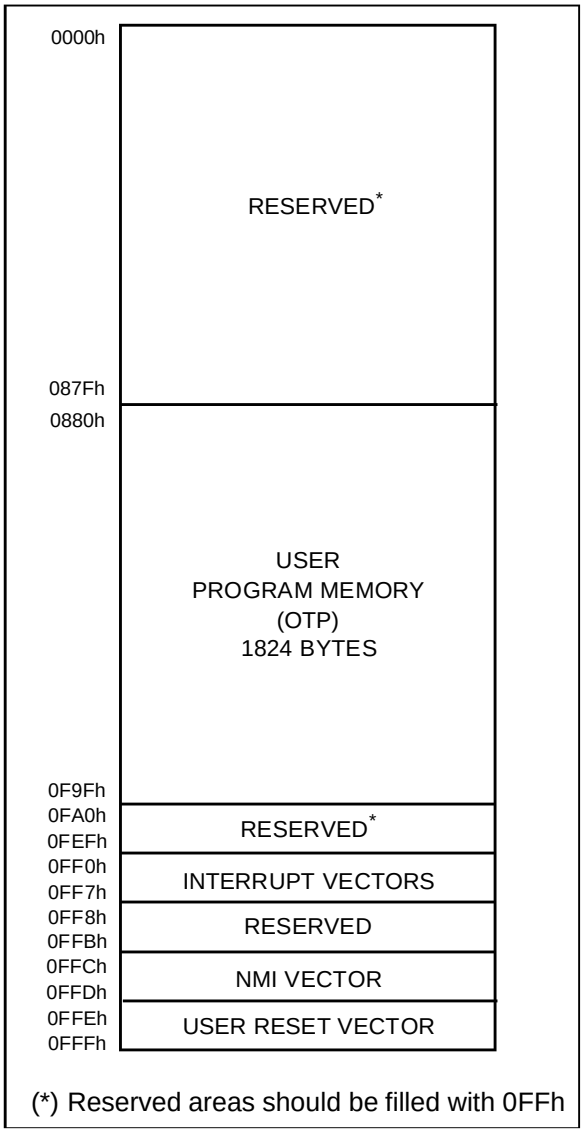


Figure 5. ST62T63B Program Memory Map



MEMORY MAP (Cont'd)

1.3.3 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.

1.3.3.1 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.

1.3.3.2 Data RAM/EEPROM

In ST62T60B, T63B and ST62E60B devices, the data space includes 60 bytes of RAM, 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 (DRW register).

Additional RAM and EEPROM pages can also be addressed using banks of 64 bytes located between addresses 00h and 3Fh.

1.3.4 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.

Table 1. Additional RAM/EEPROM Banks

Device	RAM	EEPROM
ST62T60B/E60B	1 x 64 bytes	2 x 64 bytes
ST62T63B	None	1 x 64 bytes

Table 2. ST62T60B, T63B and ST62E60B Data Memory Space

RAM and EEPROM	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
PORT C OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PRESCALER REGISTER	0D2h
TIMER COUNTER REGISTER	0D3h
TIMER STATUS CONTROL REGISTER	0D4h
AR TIMER MODE CONTROL REGISTER	0D5h
AR TIMER STATUS/CONTROL REGISTER1	0D6h
AR TIMER STATUS/CONTROL REGISTER2	0D7h
WATCHDOG REGISTER	0D8h
AR TIMER RELOAD/CAPTURE REGISTER	0D9h
AR TIMER COMPARE REGISTER	0DAh
AR TIMER LOAD REGISTER	0DBh
OSCILLATOR CONTROL REGISTER	0DCh*
MISCELLANEOUS	0DDh
RESERVED	0DEh 0DFh
SPI DATA REGISTER (ST62T60B/E60B only)	0E0h
SPI DIVIDER REGISTER (ST62T60B/E60B only)	0E1h
SPI MODE REGISTER (ST62T60B/E60B only)	0E2h
RESERVED	0E3h 0E7h
DATA RAM/EEPROM REGISTER	0E8h*
RESERVED	0E9h
EEPROM CONTROL REGISTER	0EAh
RESERVED	0EBh 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

1.3.5 Data Window Register (DWR)

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 1FFFh (top memory address depends on the specific device). All the program memory can therefore be used to store either instructions or read-only data. Indeed, the window can be moved in steps of 64 bytes along the program memory by writing the appropriate code in the Data Window Register (DWR).

The DWR can be addressed like any RAM location in the Data Space, it is however a write-only register and therefore cannot be accessed using single-bit operations. This register is used to position the 64-byte read-only data window (from address 40h to address 7Fh of the Data space) in program memory in 64-byte steps. The effective address of the byte to be read as data in program memory is obtained by concatenating the 6 least significant bits of the register address given in the instruction (as least significant bits) and the content of the DWR register (as most significant bits), as illustrated in Figure 6 below. For instance, when addressing location 0040h of the Data Space, with 0 loaded in the DWR register, the physical location addressed in program memory is 00h. The DWR register is not cleared on reset, therefore it must be written to prior to the first access to the Data read-only memory window area.

Data Window Register (DWR)

Address: 0C9h — Write Only

7							0
-	-	DWR5	DWR4	DWR3	DWR2	DWR1	DWR0

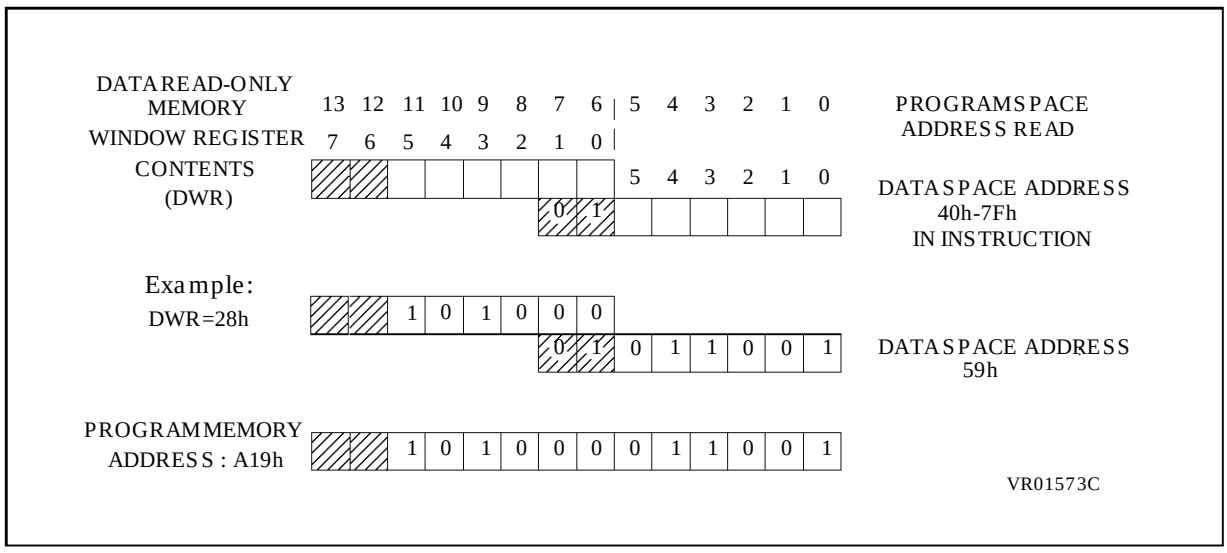
Bits 6, 7 = Not used.

Bit 5-0 = **DWR6-DWR0:** *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. Neither read nor single bit instructions may be used to address this register.*

Note: Care is required when handling the DWR register as it is write only. For this reason, the DWR 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 DWR 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 DWR, 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 DWR is not affected.

Figure 6. Data read-only memory Window Memory Addressing



MEMORY MAP (Cont'd)

1.3.6 Data RAM/EEPROM Bank Register (DRBR)

Address: E8h — Write only

7							0
-	-	-	DRBR 4	DRBR 3	-	DRBR 1	DRBR 0

Bit 7-5 = These bits are not used

Bit 4 - **DRBR4**. This bit, when set, selects RAM Page 2.

Bit 3 - **DRBR3**. This bit, when set, selects RAM Page 1.

Bit2. These bits are not used.

Bit 1 - **DRBR1**. This bit, when set, selects EEPROM Page 1.

Bit 0 - **DRBR0**. This bit, when set, selects EEPROM Page 0.

The selection of the bank is made by programming the Data RAM Bank Switch register (DRBR register) located at address E8h of the Data Space according to Table 1. No more than one bank should be set at a time.

The DRBR register can be addressed like a RAM Data Space at the address E8h; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of banks has to be loaded in the DRBR register and the instruction has to point to the selected location as if it was in bank 0 (from 00h address to 3Fh address).

This register is not cleared during the MCU initialization, therefore it must be written before the first access to the Data Space bank region. Refer to the Data Space description for additional information. The DRBR register is not modified when an interrupt or a subroutine occurs.

Notes :

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

In DRBR Register, only 1 bit must be set. Otherwise two or more pages are enabled in parallel, producing errors.

Table 3. Data RAM Bank Register Set-up

DRBR	ST62T60B/E60B	ST62T63B
00	None	None
01	EEPROM Page 0	EEPROM Page 0
02	EEPROM Page 1	Not Available
08	Not Available	Not Available
10h	RAM Page 2	Not Available
other	Reserved	Reserved

MEMORY MAP (Cont'd)

1.4 EEPROM DESCRIPTION

Depending on the specific device, 64 or 128 bytes of EEPROM memory are located in one or two 64-byte pages in data space. This memory may be used by the user program for non-volatile data storage.

Data space from 00h to 3Fh is paged as described in Table 4. EEPROM locations are accessed directly by addressing these paged sections of data space.

The EEPROM does not require dedicated instructions for read or write access. Once selected via the Data RAM Bank Register, the active EEPROM page is controlled by the EEPROM Control Register (EECTL), which is described below.

Bit E20FF of the EECTL register must be reset prior to any write or read access to the EEPROM. If no bank has been selected, or if E2OFF is set, any access is meaningless.

Programming must be enabled by setting the E2ENA bit of the EECTL register.

The E2BUSY bit of the EECTL register is set when the EEPROM is performing a programming cycle. Any access to the EEPROM when E2BUSY is set is meaningless.

Provided E2OFF and E2BUSY are reset, an EEPROM location is read just like any other data location, also in terms of access time.

Writing to the EEPROM may be carried out in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). In BMODE, one byte is accessed at a time, while in PMODE up to 8 bytes in the same row are programmed simultaneously (with consequent speed and power consumption advantages, the latter being particularly important in battery powered circuits).

EEPROM Control Register (EECTL)

Address: EAh — Read/Write

Reset status: 00h

7							0
D7	E2OFF	D5	D4	E2PAR1	E2PAR2	E2BUSY	E2ENA

Bit 7 = **D7**: *Unused.*

Bit 6 = **E2OFF**: *Stand-by Enable Bit.* WRITE ONLY. If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to its lowest value.

Bit 5-4 = **D5-D4**: *Reserved.* MUST be kept reset.

Bit 3 = **E2PAR1**: *Parallel Start Bit.* WRITE ONLY. Once in Parallel Mode, as soon as the user software sets the E2PAR1 bit, parallel writing of the 8 adjacent registers will start. This bit is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written if required, the undefined bytes being unaffected by the parallel programming cycle; this is explained in greater detail in the Additional Notes on Parallel Mode overleaf.

Bit 2 = **E2PAR2**: *Parallel Mode En. Bit.* WRITE ONLY. This bit must be set by the user program in order to perform parallel programming. If E2PAR2 is set and the parallel start bit (E2PAR1) is reset, up to 8 adjacent bytes can be written simultaneously. These 8 adjacent bytes are considered as a row, whose address lines A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bits, as illustrated in Table 4. E2PAR2 is automatically reset at the end of any parallel programming procedure. It can be reset by the user software before starting the programming procedure, thus leaving the EEPROM registers unchanged.

Bit 1 = **E2BUSY**: *EEPROM Busy Bit.* READ ONLY. This bit is automatically set by the EEPROM control logic when the EEPROM is in programming mode. The user program should test it before any EEPROM read or write operation; any attempt to access the EEPROM while the busy bit is set will be aborted and the writing procedure in progress will be completed.

Bit 0 = **E2ENA**: *EEPROM Enable Bit.* WRITE ONLY. This bit enables programming of the EEPROM cells. It must be set before any write to the EEPROM register. Any attempt to write to the EEPROM when E2ENA is low is meaningless and will not trigger a write cycle.

MEMORY MAP (Cont'd)

General Notes:

Data should be written directly to the intended address in EEPROM space. There is no buffer memory between data RAM and the EEPROM space.

When the EEPROM is busy (E2BUSY = "1") EECTL cannot be accessed in write mode, it is only possible to read the status of E2BUSY. This implies that as long as the EEPROM is busy, it is not possible to change the status of the EEPROM Control Register. EECTL bits 4 and 5 are reserved and must never be set.

Care is required when dealing with the EECTL register, as some bits are write only. For this reason, the EECTL contents must not be altered while executing an interrupt service routine.

If it is impossible to avoid writing to this register within an interrupt service routine, an image of the register must be saved in a RAM location, and each time the program writes to EECTL it must also write to the image register. The image register must be written to first so that, if an interrupt occurs between the two instructions, the EECTL will not be affected.

Additional Notes on Parallel Mode:

If the user wishes to perform parallel programming, the first step should be to set the E2PAR2 bit. From this time on, the EEPROM will be addressed in write mode, the ROW address will be latched and it will be possible to change it only at the end of the programming cycle, or by resetting E2PAR2 without programming the EEPROM. After the ROW address is latched, the MCU can only "see" the selected EEPROM row and any attempt to write or read other rows will produce errors.

The EEPROM should not be read while E2PAR2 is set.

As soon as the E2PAR2 bit is set, the 8 volatile ROW latches are cleared. From this moment on, the user can load data in all or in part of the ROW. Setting E2PAR1 will modify the EEPROM registers corresponding to the ROW latches accessed after E2PAR2. For example, if the software sets E2PAR2 and accesses the EEPROM by writing to addresses 18h, 1Ah and 1Bh, and then sets E2PAR1, these three registers will be modified simultaneously; the remaining bytes in the row will be unaffected.

Note that E2PAR2 is internally reset at the end of the programming cycle. This implies that the user must set the E2PAR2 bit between two parallel programming cycles. Note that if the user tries to set E2PAR1 while E2PAR2 is not set, there will be no programming cycle and the E2PAR1 bit will be unaffected. Consequently, the E2PAR1 bit cannot be set if E2ENA is low. The E2PAR1 bit can be set by the user, only if the E2ENA and E2PAR2 bits are also set.

Table 4. Row Arrangement for Parallel Writing of EEPROM Locations

Byte	0	1	2	3	4	5	6	7	Dataspace addresses. Banks 0 and 1.
ROW7									38h-3Fh
ROW6									30h-37h
ROW5									28h-2Fh
ROW4									20h-27h
ROW3									18h-1Fh
ROW2									10h-17h
ROW1									08h-0Fh
ROW0									00h-07h

Up to 8 bytes in each row may be programmed simultaneously in Parallel Write mode.
The number of available 64-byte banks (1 or 2) is device dependent.

1.5 PROGRAMMING MODES

1.5.1 Option Byte

The Option Byte allows configuration capability to the MCUs. Option byte's content is automatically read, and the selected options enabled, when the chip reset is activated.

It can only be accessed during the programming mode. This access is made either automatically (copy from a master device) or by selecting the OPTION BYTE PROGRAMMING mode of the programmer.

The option byte is located in a non-user map. No address has to be specified.

EPROM Code Option Byte

7							0
PRO-TECT	EXTC-NTL	-	-	WDACT	DELAY	OSCIL	-

PROTECT. This bit allows the protection of the software contents against piracy. When the bit PROTECT is set high, readout of the OTP contents is prevented by hardware. No programming equipment is able to gain access to the user program. When this bit is low, the user program can be read.

EXTCNTL. This bit selects the External STOP Mode capability. When EXTCNTL is high, pin NMI controls if the STOP mode can be accessed when the watchdog is active. In addition, PB0 is forced as open drain output. When EXTCNTL is low, the STOP instruction is processed as a WAIT as soon as the watchdog is active.

D5-D4. Reserved. Must be cleared to zero.

WDACT. This bit controls the watchdog activation. When it is high, hardware activation is selected. The software activation is selected when WDACT is low.

DELAY. This bit enables the selection of the delay internally generated after pin RESET is released. When DELAY is low, the delay is 2048 cycles of the oscillator, it is of 32768 cycles when DELAY is high.

OSCIL. When this bit is low, the oscillator must be controlled by a quartz crystal, a ceramic resonator or an external frequency. When it is high, the oscillator must be controlled by an RC network, with only the resistor having to be externally provided.

D0. Reserved. Must be cleared to zero.

The Option byte is written during programming either by using the PC menu (PC driven Mode) or automatically (stand-alone mode)

1.5.2 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 ST62E60B/T60B and ST62T63B is described in the User Manual of the EPROM Programming Board.

The MCUs can be programmed with the ST62E6xB EPROM programming tools available from SGS-THOMSON.

Table 5. ST62E60B/T60B Program Memory Map

Device Address	Description
0000h-007Fh	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 6. ST62T63B 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

Note: OTP/EPROM devices can be programmed with the development tools available from SGS-THOMSON (ST62E6X-EPB or ST626X-KIT).

1.5.3 EEPROM Data Memory

EEPROM data pages are supplied in the virgin state 00h. Partial or total programming of EEPROM data memory can be performed either through the application software, or through an external programmer. Any SGS-THOMSON tool used for the program memory (OTP/EPROM) can also be used to program the EEPROM data memory.

PROGRAMMING MODES (Cont'd)

1.5.4 EPROM Erasing

The EPROM of the windowed package of the MCUs may be erased by exposure to Ultra Violet light. The erasure characteristic of the MCUs is such that erasure begins when the memory is exposed to light with a wave lengths shorter than approximately 4000Å. It should be noted that sunlights and some types of fluorescent lamps have wavelengths in the range 3000-4000Å.

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

The recommended erasure procedure of the MCUs EPROM is the exposure to short wave ultraviolet light which have a wave-length 2537A. The integrated dose (i.e. U.V. intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST62E60B should be placed within 2.5cm (1Inch) of the lamp tubes during erasure.

2 CENTRAL PROCESSING UNIT

2.1 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. In-core communication is arranged as shown in Figure 7; the controller being externally linked to both the Reset and Oscillator circuits, while the core is linked to the dedicated on-chip peripherals via the serial data bus and indirectly, for interrupt purposes, through the control registers.

2.2 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.

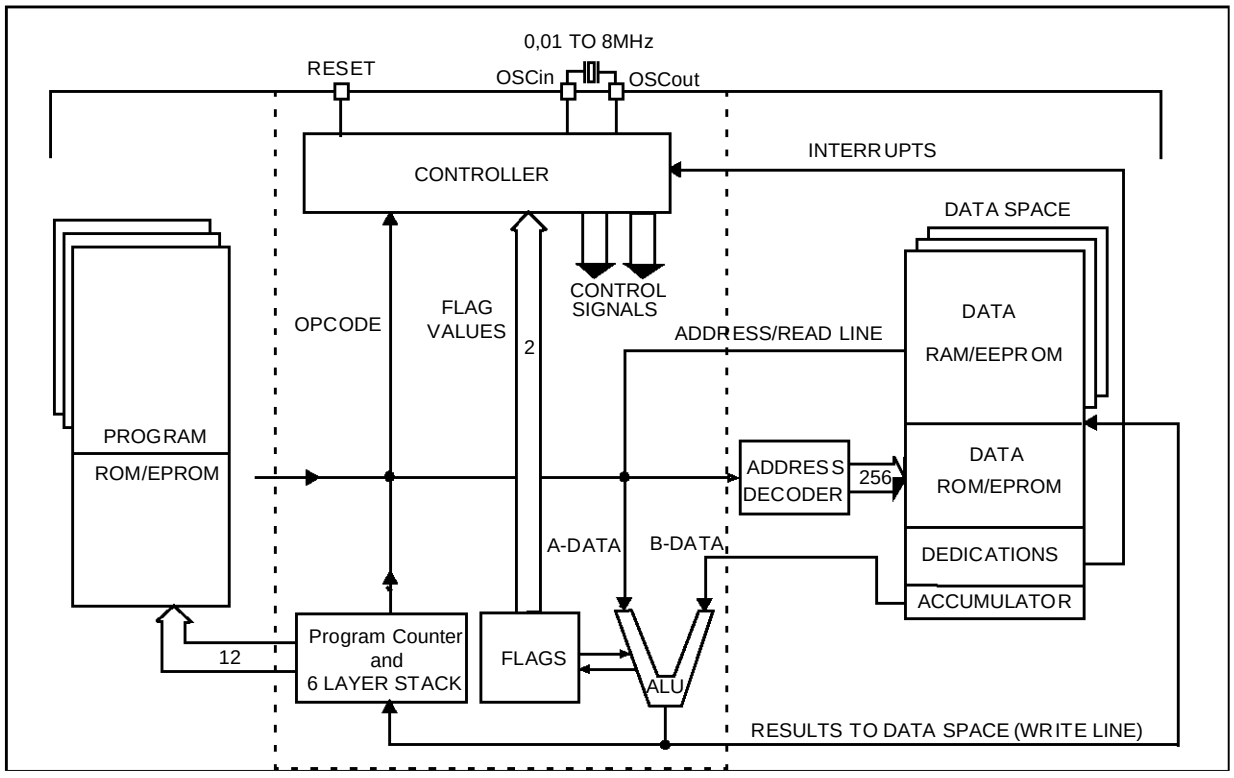
Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator can be addressed in Data space as a RAM location at address FFh. Thus the ST6 can manipulate the accumulator just like any other register in Data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to memory locations in Data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST6 instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save a byte in short direct addressing mode. They can be addressed in Data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed using the direct and bit direct addressing modes. Thus, the ST6 instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC). The program counter is a 12-bit register which contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or the address of an operand. The 12-bit length allows the direct addressing of 4096 bytes in Program space.

Figure 7. ST6 Core Block Diagram



CPU REGISTERS (Cont'd)

However, if the program space contains more than 4096 bytes, the additional memory in program space can be addressed by using the Program Bank Switch 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) instructionPC=Jump address
- CALL instructionPC= Call address
- Relative Branch Instruction.PC= PC +/- offset
- Interrupt PC=Interrupt vector
- ResetPC= Reset vector
- RET & RETI instructionsPC= Pop (stack)
- Normal instructionPC= 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 (resp. 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.

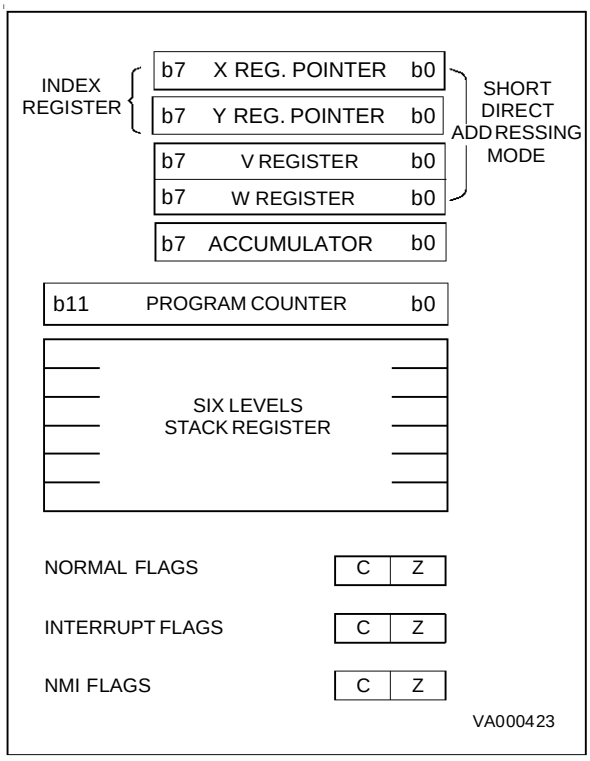
The Carry flag is set when a carry or a borrow occurs during arithmetic operations; otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction; it also participates in the rotate left instruction.

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

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

Stack. The ST6 CPU includes a true LIFO 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 higher level, 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. 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. 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.

Figure 8. ST6 CPU Programming Mode



3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}).

Figure 9 illustrates various possible oscillator configurations using an external crystal or ceramic resonator, an external clock input, an external resistor (R_{NET}). C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range. The value of RNET can be obtained by referring to Figure 34 and Figure 35.

A programmable divider is provided in order to adjust the internal clock of the MCU to the best power consumption and performance trade-off.

The internal MCU clock frequency (f_{INT}) drives directly the AR TIMER while it is divided by 12 to drive the TIMER, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 10.

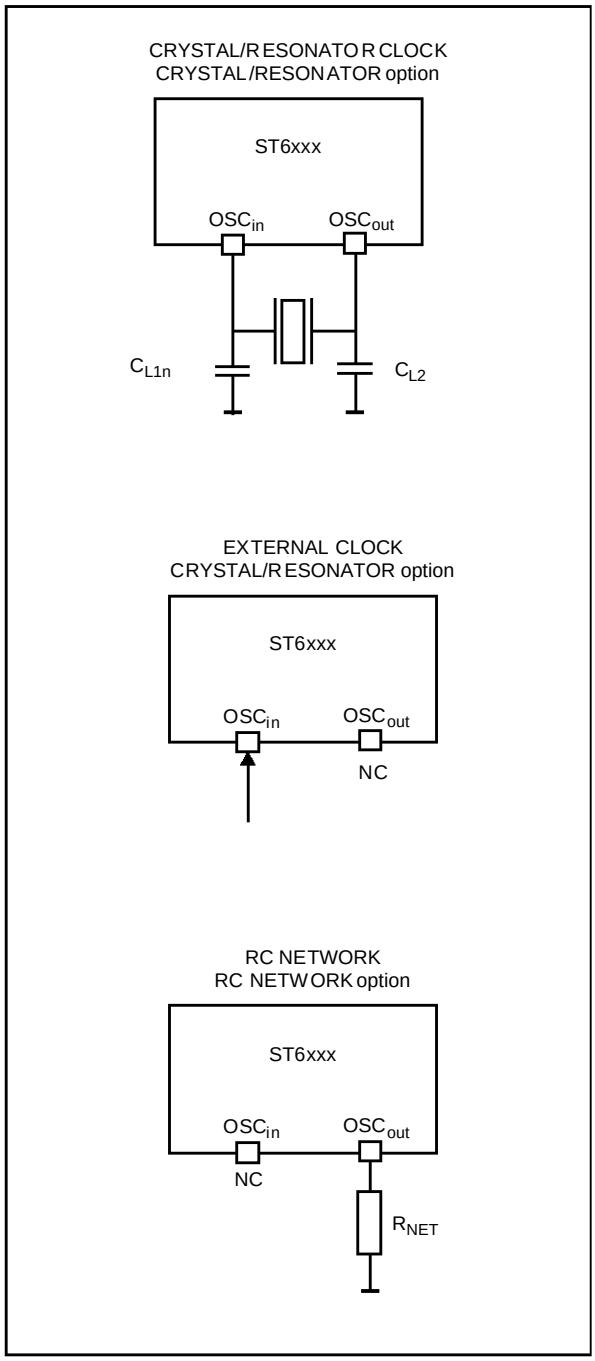
With an 8MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

A machine 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 machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate option. 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.

Figure 9. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Oscillator Control Registers

Address: DCh — Write only

7							0
-	-	-	-	OSCR 3	OSCR 2	RS1	RS0

Bit 7-4. These bits are not used.

Bit 3. Reserved. Cleared at Reset. THIS BIT MUST BE SET TO 1 BY USER PROGRAM to achieve lowest power consumption.

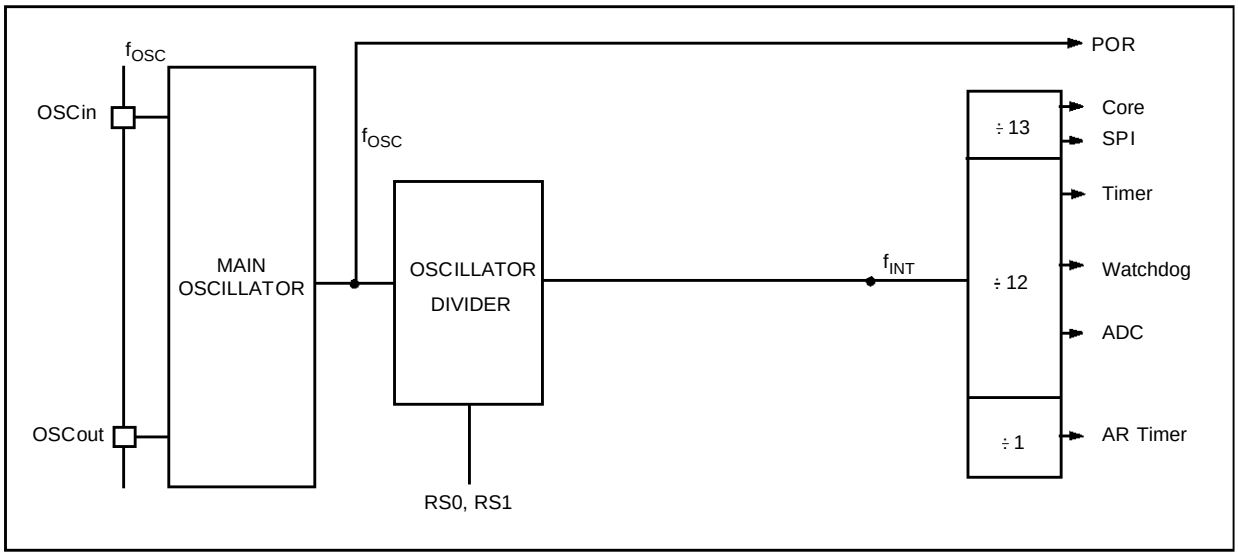
Bit 2. Reserved. Must be kept low.

RS1-RS0. These bits select the division ratio of the Oscillator Divider in order to generate the internal frequency. The following selctions are available:

RS1	RS0	Division Ratio
0	0	1
0	1	2
1	0	4
1	1	4

Note: Care is required when handling the OSCR register as some bits are write only. For this reason, it is not allowed to change the OSCR contents while executing interrupt service routine, as the service routine cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupt service routine, an image of this register must be saved in a RAM location, and each time the program writes to OSCR it must write also to the image register. The image register must be written first, so if an interrupt occurs between the two instructions the OSCR is not affected.

Figure 10. Clock Circuit Block Diagram



3.2 RESETS

The MCU can be reset in three ways:

– by the external Reset input being pulled low;

– by Power-on Reset;

– by the digital Watchdog peripheral timing out.

3.2.1 RESET Input

The $\overline{\text{RESET}}$ pin may be connected to a device of 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 MCU internal state and ensure a correct start-up procedure. The pin is active low and features a Schmitt trigger input. The internal Reset signal is generated by adding a delay to the external signal. Therefore even short pulses on the $\overline{\text{RESET}}$ pin are acceptable, 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 $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT modes, processing of the user program is stopped (RUN mode only), the Inputs and Outputs are configured as inputs with pull-up resistors and the main Oscillator is restarted. When the level on the RESET pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

If $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts up and all Inputs and Outputs are configured as inputs with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin then goes high, the initialization sequence is executed following expiry of the internal delay period.

3.2.2 Power-on Reset

The function of the POR circuit consists in waking up the MCU at an appropriate stage during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: all I/O ports are configured as inputs with pull-up resistors and no instruction is executed. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

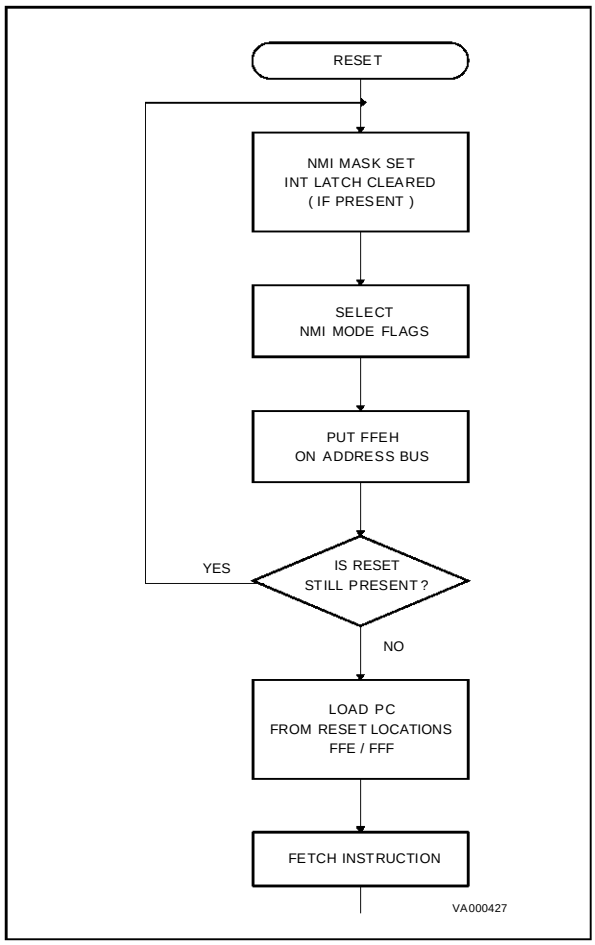
The internal delay is generated by an on-chip counter. The internal reset line is released 2048 internal clock cycles after release of the external reset.

Notes:

To ensure correct start-up, the user should take care that the reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

Figure 11. Reset and Interrupt Processing



RESETS (Cont'd)

3.2.3 Watchdog Reset

The MCU provides a Watchdog timer function in order to ensure graceful recovery from software upsets. If the Watchdog register is not refreshed before an end-of-count condition is reached, the internal reset will be activated. This, amongst other things, resets the watchdog counter.

The MCU restarts just as though the Reset had been generated by the $\overline{\text{RESET}}$ pin, including the built-in stabilisation delay period.

3.2.4 Application Notes

No external resistor is required between V_{DD} and the Reset pin, thanks to the built-in pull-up device.

The POR circuit operates dynamically, in that it triggers MCU initialization on detecting the rising edge of V_{DD}. The typical threshold is in the region of 2 volts, but the actual value of the detected threshold depends on the way in which V_{DD} rises.

The POR circuit is *NOT* designed to supervise static, or slowly rising or falling V_{DD}.

3.2.5 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 12. Reset and Interrupt Processing

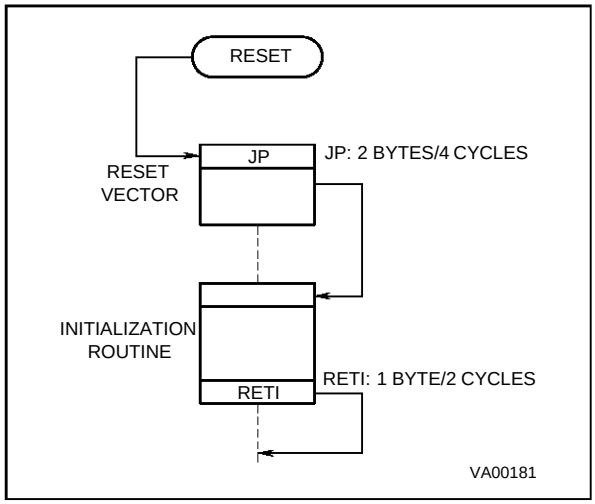
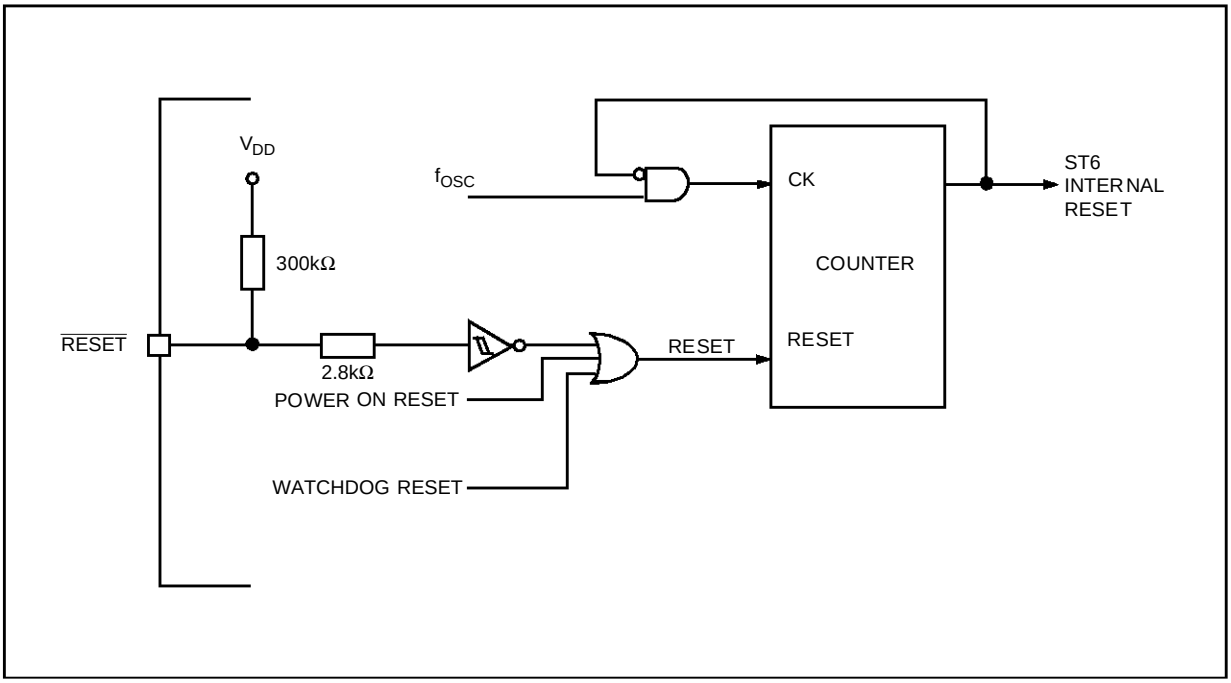


Figure 13. Reset Block Diagram



RESETS (Cont'd)

Table 7. Register Reset Status

Register	Address(es)	Status	Comment
Oscillator Control Register	0DCh		$f_{INT} = f_{OSC}$; user must set bit3 to 1
EEPROM Control Register	0EAh		EEPROM disabled
Port Data Registers	0C0h to 0C2h		I/O are Input with pull-up
Port Direction Register	0C4h to 0C6h		I/O are Input with pull-up
Port Option Register	0CCh to 0CEh		I/O are Input with pull-up
Interrupt Option Register	0C8h		Interrupt disabled
TIMER Status/Control	0D4h	00h	TIMER disabled
AR TIMER Mode Control Register	0D5h		AR TIMER stopped
AR TIMER Status/Control 1 Register	0D6h		
AR TIMER Status/Control 2Register	0D7h		
AR TIMER Compare Register	0DAh		
Miscellaneous Register	0DDh		SPI Output not connected to PC3
SPI Registers	0E0h to 0E2h		SPI disabled *)
X, Y, V, W, Register	080H TO 083H		
Accumulator	0FFh		
Data RAM	084h to 0BFh		
Data RAM Page REgister	0E8h		
Data ROM Window Register	0C9h	Undefined	
EEPROM	00h to 03Fh		As written if programmed
A/D Result Register	0D0h		
AR TIMER Load Register	0DBh		
AR TIMER Reload/Capture Register	0D9h		
TIMER Counter Register	0D3h	FFh	
TIMER Prescaler Register	0D2h	7Fh	Max count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Standby

*) On ST62T60B and ST62E60B only

3.3 DIGITAL WATCHDOG

The digital Watchdog consists of a reloadable downcounter timer which can be used to provide controlled recovery from software upsets.

The Watchdog circuit generates a Reset when the downcounter reaches zero. User software can prevent this reset by reloading the counter, and should therefore be written so that the counter is regularly reloaded while the user program runs correctly. In the event of a software mishap (usually caused by externally generated interference), the user program will no longer behave in its usual fashion and the timer register will thus not be reloaded periodically. Consequently the timer will decrement down to 00h and reset the MCU. In order to maximise the effectiveness of the Watchdog function, user software must be written with this concept in mind.

Watchdog behaviour is governed by two options, known as "WATCHDOG ACTIVATION" (i.e. HARDWARE or SOFTWARE) and "EXTERNAL STOP MODE CONTROL" (see Table 8).

In the SOFTWARE option, the Watchdog is disabled until bit C of the DWDR register has been set. When the Watchdog is disabled, low power Stop mode is available. Once activated, the Watchdog cannot be disabled, except by resetting the MCU.

In the HARDWARE option, the Watchdog is permanently enabled. Since the oscillator will run continuously, low power mode is not available. The STOP instruction is interpreted as a WAIT instruction, and the Watchdog continues to countdown.

However, when the EXTERNAL STOP MODE CONTROL option has been selected low power consumption may be achieved in Stop Mode.

Execution of the STOP instruction is then governed by a secondary function associated with the NMI pin. If a STOP instruction is encountered when the NMI pin is low, it is interpreted as WAIT, as described above. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode.

When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

Note: when the EXTERNAL STOP MODE CONTROL option has been selected, port PB0 must be defined as an open-drain output.

Table 8. Recommended Option Choices

Functions Required	Recommended Options
Stop Mode & Watchdog	"EXTERNAL STOP MODE" & "HARDWARE WATCHDOG"
Stop Mode	"SOFTWARE WATCHDOG"
Watchdog	"HARDWARE WATCHDOG"

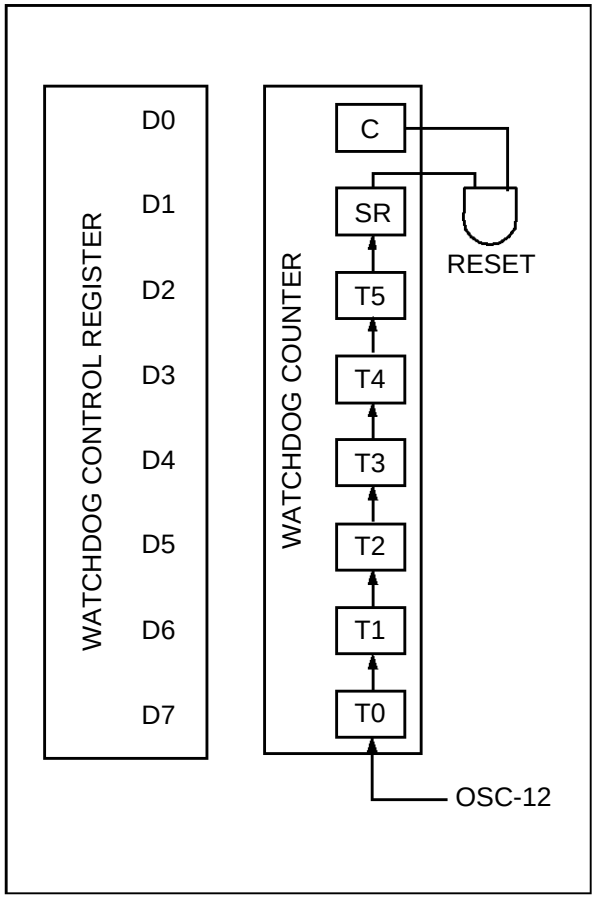
DIGITAL WATCHDOG (Cont'd)

The Watchdog is associated with a Data space register (Digital WatchDog Register, DWDR, location 0D8h) which is described in greater detail in Section 3.3.1. This register is set to 0FEh on Reset: bit C is cleared to "0", which disables the Watchdog; the timer downcounter bits, T0 to T5, and the SR bit are all set to "1", thus selecting the longest Watchdog timer period. This time period can be set to the user's requirements by setting the appropriate value for bits T0 to T5 in the DWDR register. The SR bit must be set to "1", since it is this bit which generates the Reset signal when it changes to "0"; clearing this bit would generate an immediate Reset.

It should be noted that the order of the bits in the DWDR register is inverted with respect to the associated bits in the down counter: bit 7 of the DWDR register corresponds, in fact, to T0 and bit 2 to T5. The user should bear in mind the fact that these bits are inverted and shifted with respect to the physical counter bits when writing to this register. The relationship between the DWDR register bits and the physical implementation of the Watchdog timer downcounter is illustrated in Figure 14.

Only the 6 most significant bits may be used to define the time period, since it is bit 6 which triggers the Reset when it changes to "0". This offers the user a choice of 64 timed periods ranging from 3,072 to 196,608 clock cycles (with an oscillator frequency of 8MHz, this is equivalent to timer periods ranging from 384µs to 24.576ms).

Figure 14. Watchdog Counter Control



DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to PB0 (see Figure 15) to allow its state to be controlled by software. PB0 can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, PB0 is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

DIGITAL WATCHDOG (Cont'd)

These instructions test the C bit and Reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

In all modes, a minimum of 28 instructions are executed after activation, before the Watchdog can generate a Reset. Consequently, user software should load the watchdog counter within the first 27 instructions following Watchdog activation (software mode), or within the first 27 instructions executed following a Reset (hardware activation).

It should be noted that when the GEN bit is low (interrupts disabled), the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

Figure 15. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature

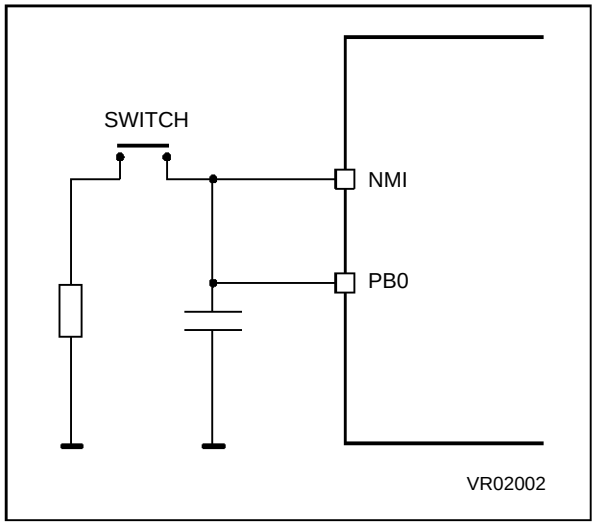
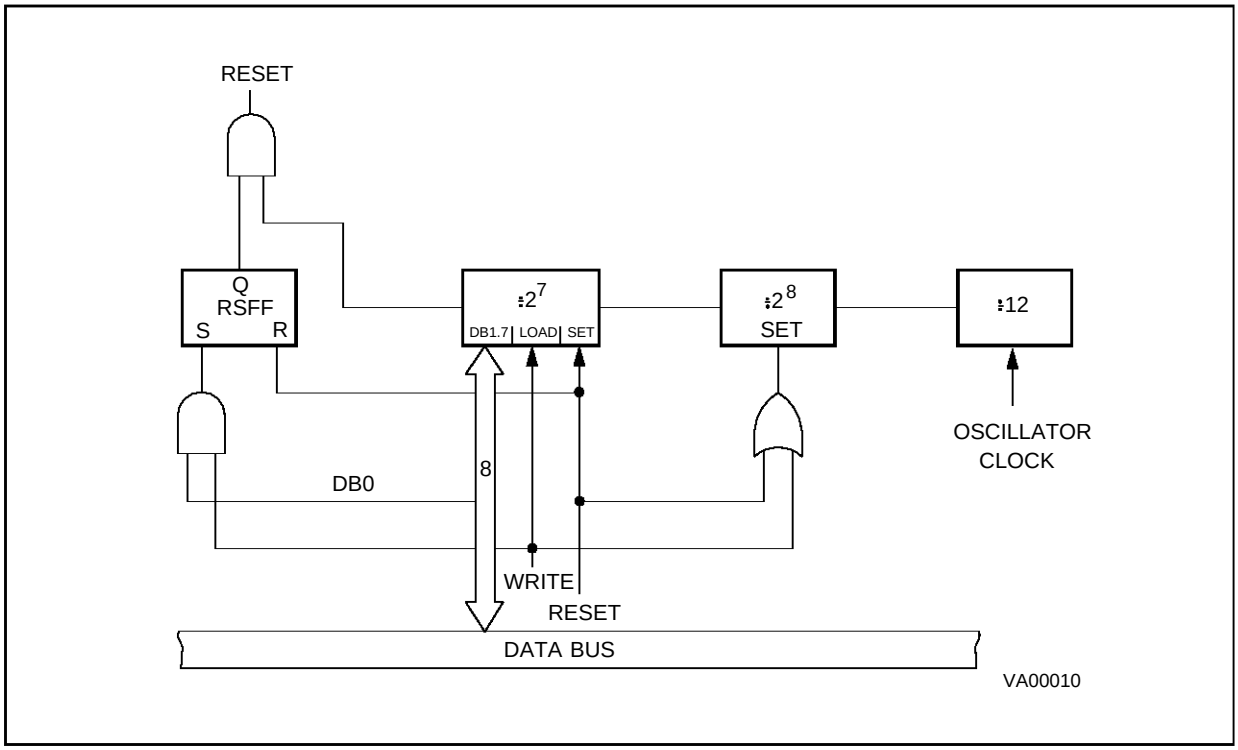


Figure 16. Digital Watchdog Block Diagram



3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 9).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Table 9. Interrupt Vector Map

Interrupt Source	Associated Vector	Vector Address
NMI pin	Interrupt vector #0 (NMI)	(FFCh-FFDh)
Port A, B	Interrupt vector #1	(FF6h-FF7h)
Port C, SPI	Interrupt vector #2	(FF4h-FF5h)
ARTIMER peripheral	Interrupt vector #3	(FF2h-FF3h)
TIMER and ADC peripherals	Interrupt vector #4	(FF0h-FF1h)

3.4.1 Interrupt Vectors

Interrupt vectors are Jump addresses to the associated service routine, which reside in specific areas of Program space. The following vectors are present:

The interrupt vector associated with the non-maskable interrupt source is referred to as Interrupt Vector #0. It is located at addresses 0FFCh and 0FFDh in Program space. This vector is associated with the falling edge sensitive Non Maskable Interrupt pin (NMI).

– The interrupt vector associated with Port A and B pins is referred to as interrupt vector #1. It is located at addresses 0FF6h, 0FF7h is named. It can be programmed either as falling edge sensitive or as low level sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The interrupt vector associated with Port C pins and SPI is referred to as interrupt vector #2. It is located at addresses 0FF4h, 0FF5h is named . It can be programmed either as falling edge sensitive or as rising edge sensitive, by setting the Interrupt Option Register (IOR) accordingly.

– The two interrupt vectors located respectively at addresses 0FF2h, 0FF3h and addresses 0FF0h, 0FF1h are respectively known as Interrupt Vectors #3 and #4. Vector #3 is associated with the ARTIMER peripheral and vector #4 with the A/D Converter or Timer peripherals.

Each on-chip peripheral has an associated interrupt request flag (A/D Converter, OVF, CPF and EF for the ARTIMER, SPRUN for the SPI), which is set to “1” when the peripheral generates an interrupt request. Each on-chip peripheral also has an associated mask bit (A/D Converter, OVIE and EIE for the ARTIMER, SPIE for the SPI), which must be set to “1” to enable the associated interrupt request.

Note: SPI interrupts are not available on ST62T63B.

3.4.2 Interrupt Priorities

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is fixed.

IINTERRUPTS (Cont'd)

3.4.3 Interrupt Option Register (IOR)

The Interrupt Option Register (IOR) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register is write-only and cannot be accessed by single-bit operations.

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Bit 7, Bits 3-0 = *Unused*.

Bit 6 = **LES**: *Level/Edge Selection bit*.

When this bit is set to one, the interrupt #1 (Port A, B) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 5 = **ESB**: *Edge Selection bit*.

When this bit is set to one, the interrupt #2 (Port C, SPI) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

Bit 4 = **GEN**: *Global Enable Interrupt*. When this bit is set to one, all interrupts are enabled. When this bit is cleared to zero all the interrupts (excluding NMI) are disabled.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

This register is cleared on reset.

Table 10. Interrupt Options

GEN	SET	Enables all interrupts
	CLEARED	Disables all interrupts (Except NMI)
LES	SET	Rising edge mode on Port A, B
	CLEARED	Falling edge mode on Port A, B
ESB	SET	Level sensitive mode on Port C, SPI
	CLEARED	Falling edge mode on Port C, SPI

3.4.4 External Interrupt Operating Modes

The NMI interrupt is associated with the external interrupt pin. This pin is falling edge sensitive and the interrupt pin signal is latched by a flip-flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine. A Schmitt trigger is present on the NMI pin. The user can choose to have an on-chip pull-up on the NMI pin by specifying the appropriate ROM mask option (see Option List at the end of the Datasheet).

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Port A, B-vector #1, Port C, SPI-vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs during the processing of the previous one, will be processed as soon as the first one has been serviced (unless a higher priority interrupt request is present). If more than one interrupt occurs while processing the first one, the subsequent ones will be lost.

Storage of interrupt requests is not available in level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

When the GEN bit is low, the NMI interrupt is active but cannot cause a wake up from STOP/WAIT modes.

3.4.5 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

IINTERRUPTS (Cont'd)

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.

– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).

– The PC contents are stored in the first level of the stack.

– The normal interrupt lines are inhibited (NMI still active).

– The first internal latch is cleared.

– The associated interrupt vector is loaded in the PC.

User

– User selected registers are saved within the interrupt service routine (normally on a software stack).

– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).

– The interrupt is serviced.

– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 17. Interrupt Processing Flow Chart

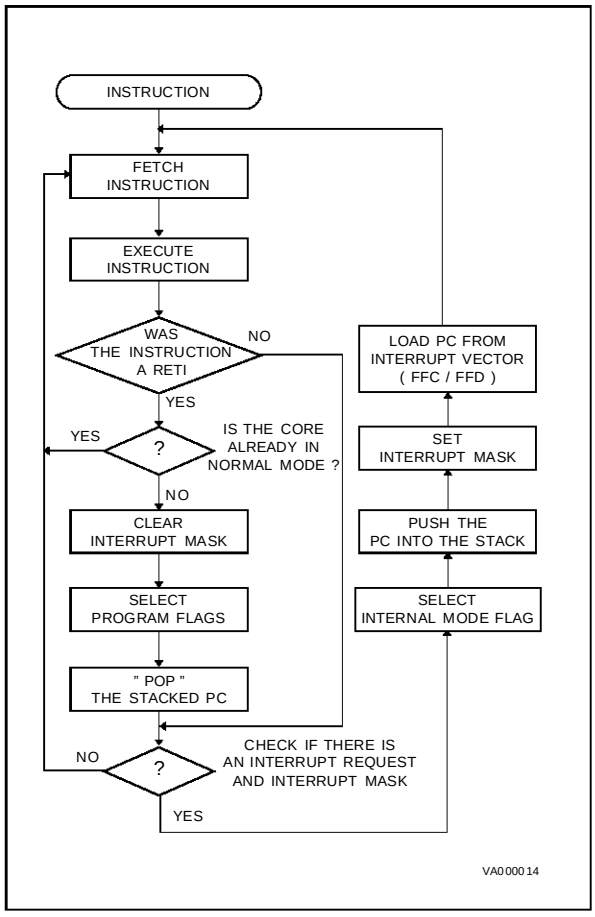
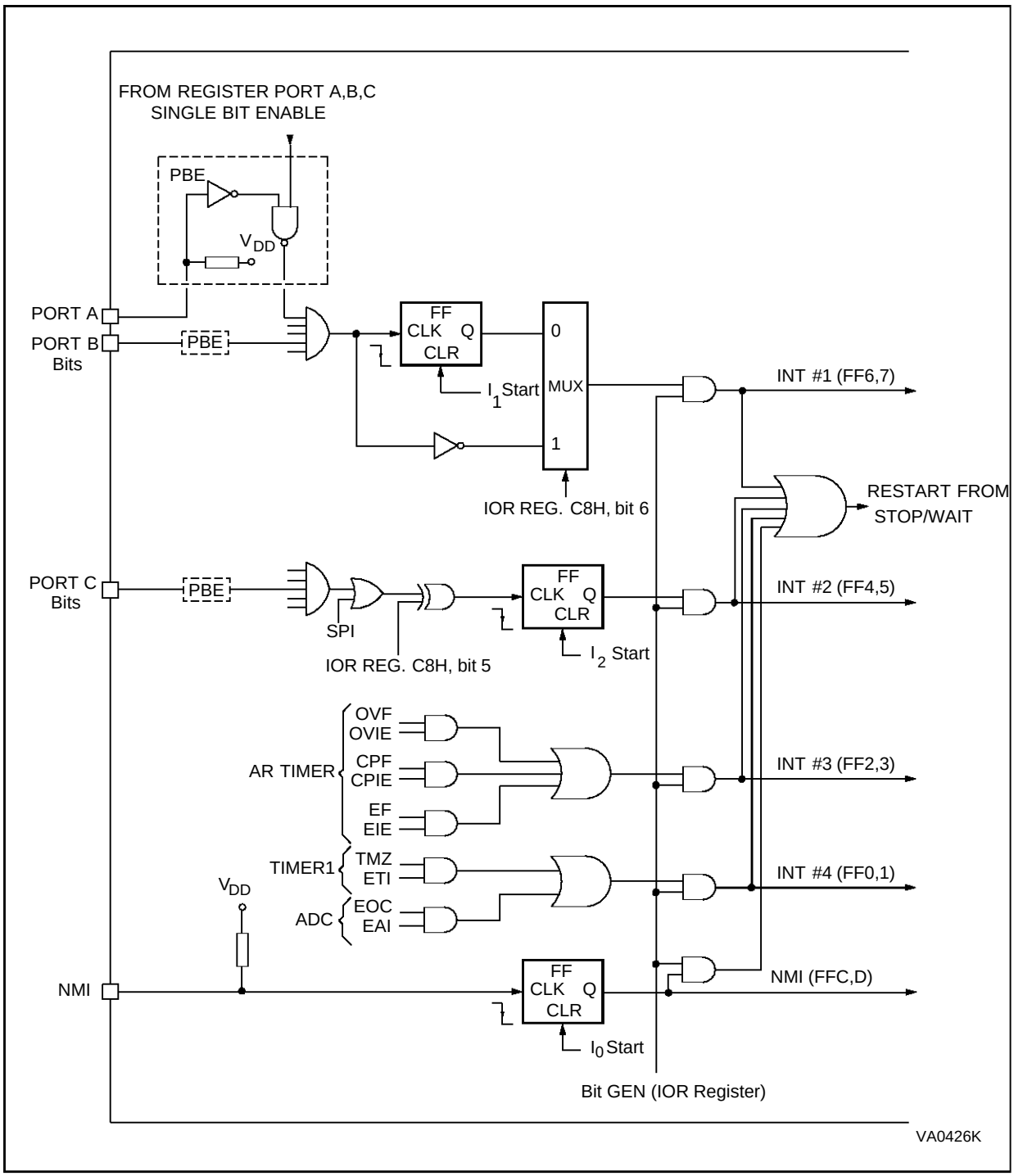


Table 11. Interrupt Requests and Mask Bits

Peripheral	Register	Address Register	Mask bit	Masked Interrupt Source	Interrupt vector
GENERAL	IOR	C8h	GEN	All Interrupts, excluding NMI	
TIMER	TSCR1	D4h	ETI	TMZ: TIMER Overflow	Vector 4
A/D CONVERTER	ADCR	D1h	EAI	EOC: End of Conversion	Vector 4
AR TIMER	ARMC	D5h	OVIE CPIE EIE	OVF: AR TIMER Overflow CPF: Successful compare EF: Active edge on ARTIMin	Vector 3
SPI	SPIMOD	E2h	SPIE	SPRUN: End of Transmission	Vector 2
Port PAn	ORPA-DRPA	C0h-C4h	ORPAn-DRPAn	PAn pin	Vector 1
Port PBn	ORPB-DRPB	C1h-C5h	ORPBn-DRPBn	PBn pin	Vector 1
Port PCn	ORPC-DRPC	C2h-C6h	ORPCn-DRPCn	PCn pin	Vector 2

INTERRUPTS (Cont'd)

Figure 18. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the capability of monitoring external events. The active oscillator is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

POWER SAVING MODE (Cont'd)

3.5.3 Exit from WAIT and STOP Modes

The following paragraphs describe how the MCU exits from WAIT and STOP modes, when an interrupt occurs (not a Reset). It should be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) prior to entering WAIT or STOP mode, as well as on the interrupt type.

Interrupts do not affect the oscillator selection.

3.5.3.1 Normal Mode

If the MCU was in the main routine when the WAIT or STOP instruction was executed, exit from Stop or Wait mode will occur as soon as an interrupt occurs; the related interrupt routine is executed and, on completion, the instruction which follows the STOP or WAIT instruction is then executed, providing no other interrupts are pending.

3.5.3.2 Non Maskable Interrupt Mode

If the STOP or WAIT instruction has been executed during execution of the non-maskable interrupt routine, the MCU exits from the Stop or Wait mode as soon as an interrupt occurs: the instruction which follows the STOP or WAIT instruction is executed, and the MCU remains in non-maskable interrupt mode, even if another interrupt has been generated.

3.5.3.3 Normal Interrupt Mode

If the MCU was in interrupt mode before the STOP or WAIT instruction was executed, it exits from STOP or WAIT mode as soon as an interrupt occurs. Nevertheless, two cases must be considered:

– If the interrupt is a normal one, the interrupt routine in which the WAIT or STOP mode was entered will be completed, starting with the execution of the instruction which follows the STOP or the WAIT instruction, and the MCU is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance with their priority.

– In the event of a non-maskable interrupt, the non-maskable interrupt service routine is processed first, then the routine in which the WAIT or STOP mode was entered will be completed by executing the instruction following the STOP or WAIT instruction. The MCU remains in normal interrupt mode.

Notes:

To achieve the lowest power consumption during RUN or WAIT modes, the user program must take care of:

– configuring unused I/Os as inputs without pull-up (these should be externally tied to well defined logic levels);

– placing all peripherals in their power down modes before entering STOP mode;

When the hardware activated Watchdog is selected, or when the software Watchdog is enabled, the STOP instruction is disabled and a WAIT instruction will be executed in its place.

If all interrupt sources are disabled (GEN low), the MCU can only be restarted by a Reset. Although setting GEN low does not mask the NMI as an interrupt, it will stop it generating a wake-up signal.

The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

The MCU features 13 Input/Output lines which may be individually programmed as any of the following input or output configurations:

– Input without pull-up or interrupt

– Input with pull-up and interrupt

– Input with pull-up, but without interrupt

– Analog input (PA0-PA3, PC2-PC4)

– Artimer I/O lines : PB6-PB7

– Push-pull output

– Standard Open drain output

– 20mA Open drain output (PB0-PB3, PB6-PB7)

The lines are organized as three Ports (A, B and C).

Each port is associated with 3 registers in Data space. Each bit of these registers is associated with a particular line (for instance, bits 0 of Port A Data, Direction and Option registers are associated with the PA0 line of Port A).

The three DATA registers (DRA, DRB and DRC), are used to read the voltage level values of the lines which have been configured as inputs, or to write the logic value of the signal to be output on the lines configured as outputs. The port data registers can be read to get the effective logic levels of the pins, but they can be also written by user software, in conjunction with the related option registers, to select the different input mode options.

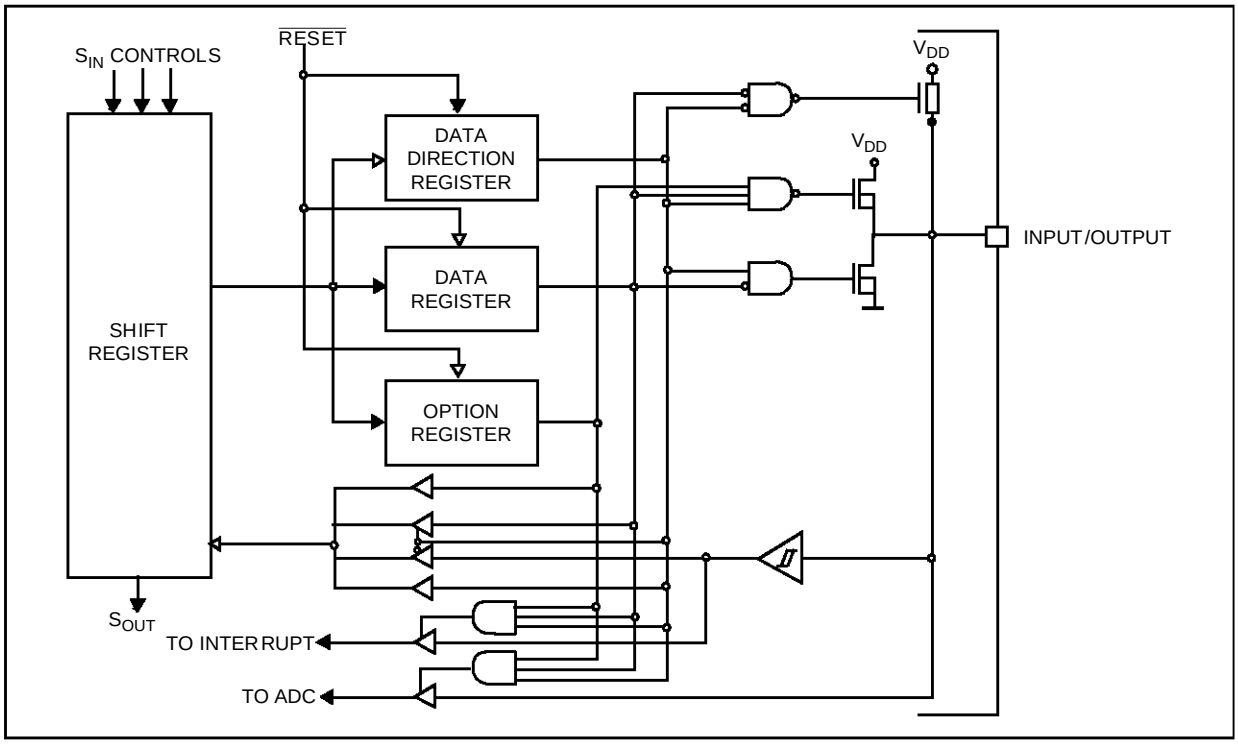
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The three Data Direction registers (DDRA, DDRB AND DRC) allow the data direction (input or output) of each pin to be set.

The three Option registers (ORA, ORB and ORC) are used to select the different port options available both in input and in output mode.

All I/O registers can be read or written to just as any other RAM location in Data space, so no extra RAM cells are needed for port data storage and manipulation. During MCU initialization, all I/O registers are cleared and the input mode with pull-ups and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

Figure 19. I/O Port Block Diagram



I/O PORTS (Cont'd)

4.1.1 Operating Modes

Each pin may be individually programmed as input or output with various configurations (except for PB0 on devices with the EXTERNAL STOP MODE CONTROL option).

This is achieved by writing the relevant bit in the Data (DR), Data Direction (DDR) and Option registers (OR). Table 12 illustrates the various port configurations which can be selected by user software.

4.1.1.1 Input Options

Pull-up, High Impedance Option. All input lines can be individually programmed with or without an internal pull-up by programming the OR and DR registers accordingly. If the pull-up option is not selected, the input pin will be in the high-impedance state.

4.1.1.2 Interrupt Options

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The pins of Port A and B are AND-connected to the interrupt associated with Vector #1. The pins of Port care AND-connected to the interrupt associated with Vector #2. The interrupt trigger modes (falling edge, rising edge and low level) can be selected by software for each port by programming the IOR register accordingly.

4.1.1.3 Analog Input Options

The seven pins, PA0-PA3, PC2-PC4, can be configured as analog inputs by programming the OR and DR registers accordingly. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

4.1.2 I/O Port Option Registers

ORA/B/C (CCh PA, CDh PB, CEh PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Option Register bits.*

4.1.3 I/O Port Data Direction Registers

DDRA/B/C (C4h PA, C5h PB, C6h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Direction Registers bits.*

4.1.4 I/O Port Data Registers

DRA/B/C (C0h PA, C1h PB, C2h PC)
Read/Write

7							0
Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0

Bit 7-0 = **Px7 - Px0**: *Port A, B and C Data Registers bits.*

Table 12. I/O Port Option Selection

DDR	OR	DR	Mode	Option
0	0	0	Input	With pull-up, no interrupt (Reset state)
0	0	1	Input	No pull-up, no interrupt
0	1	0	Input	With pull-up and with interrupt
0	1	1	Input	No pull-up, no interrupt (PB0-PB3,PB6-PB7)
			Input	Analog input (PA0-PA3, PC2-PC4)
1	0	X	Output	20mA sink open-drain output (PB0-PB3,PB6-PB7)
1	0	X	Output	Standard open-drain output (PA0-PA3, PC2-PC4)
1	1	X	Output	20mA sink push-pull output (PB0-PB3,PB6-PB7)

Note: X = Don't care

I/O PORTS (Cont'd)

4.1.5 AR Timer Alternate function Option

When bit PWMOE of register ARMC is low, pin ARTIMout/PB7 is configured as any standard pin of port B through the port registers. When PWMOE is high, ARTIMout/PB7 is the PWM output, independently of the port registers configuration.

ARTIMin/PB6 is connected to the AR Timer input. It is configured through the port registers as any standard pin of port B. To use ARTIMin/PB6 as AR Timer input, it must be configured as input through DDRB.

4.1.6 SPI Alternate function Option

PC2/PC4 are used as standard I/O as long as bit SPCLK of the SPI Mode Register is kept low.When PC2/Sin is configured as input, it is automatically connected to the SPI shift register input, independent of the state at SPCLK.

PC3/SOUT is configured as SPI push-pull output by setting bit 0 of the Miscellaneous Register (address DDh), regardless of the state of Port C registers. PC4/SCK is configured as push-pull output clock (master mode) by programming it as push-pull output through DDRC register and by setting bit SPCLK of the SPI Mode Register.

PC4/SCK is configured as input clock (slave mode) by programming it as input through DDRC register and by clearing bit SPCLK of the SPI Mode Register. With this configuration, PC4 can simultaneously be used as an input.

4.1.7 Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 20. All other transitions are potentially risky and should be avoided when changing the I/O operating mode, as it is most likely that undesirable side-effects will be experienced, such as spurious interrupt generation or two pins shorted together by the analog multiplexer.

Single bit instructions (SET, RES, INC and DEC) should be used with great caution on Ports A, B and C Data registers, since these instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins. As a general rule, it is better to limit the use of single bit instructions on data registers to when the whole (8-bit) port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

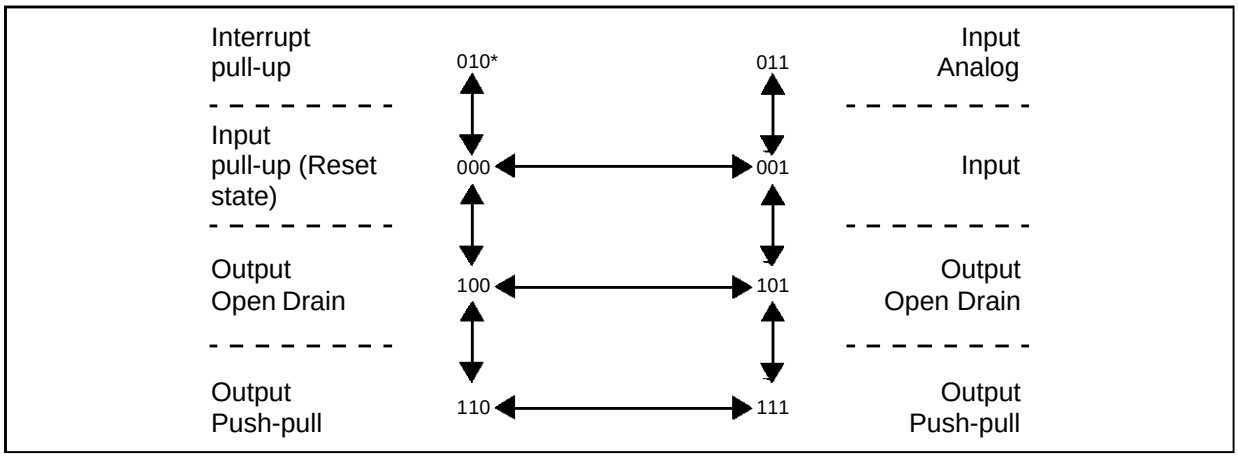
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Care must also be taken to not use INC or DEC instructions on a port register when the 8 bits are not available on the devices.

The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in input mode with well-defined logic levels.

The user must take care not to switch outputs with heavy loads during the conversion of one of the analog inputs in order to avoid any disturbance to the conversion.

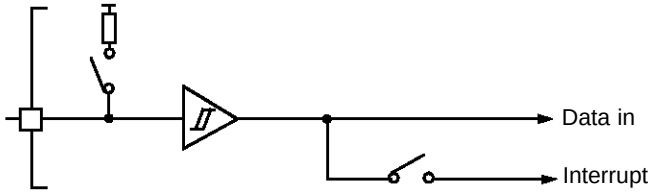
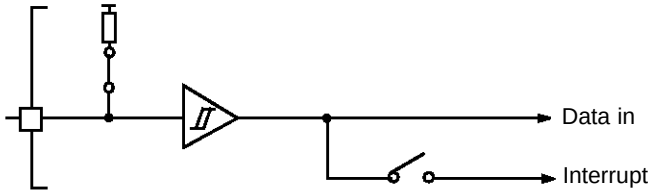
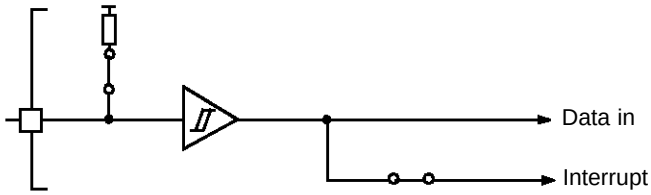
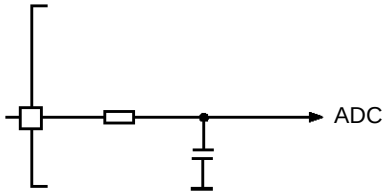
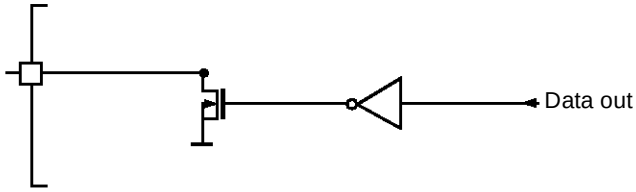
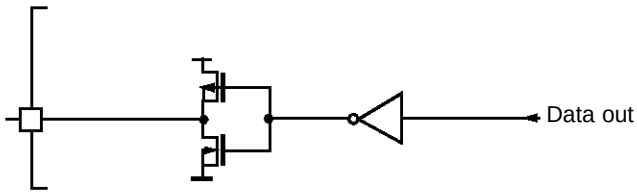
Figure 20. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

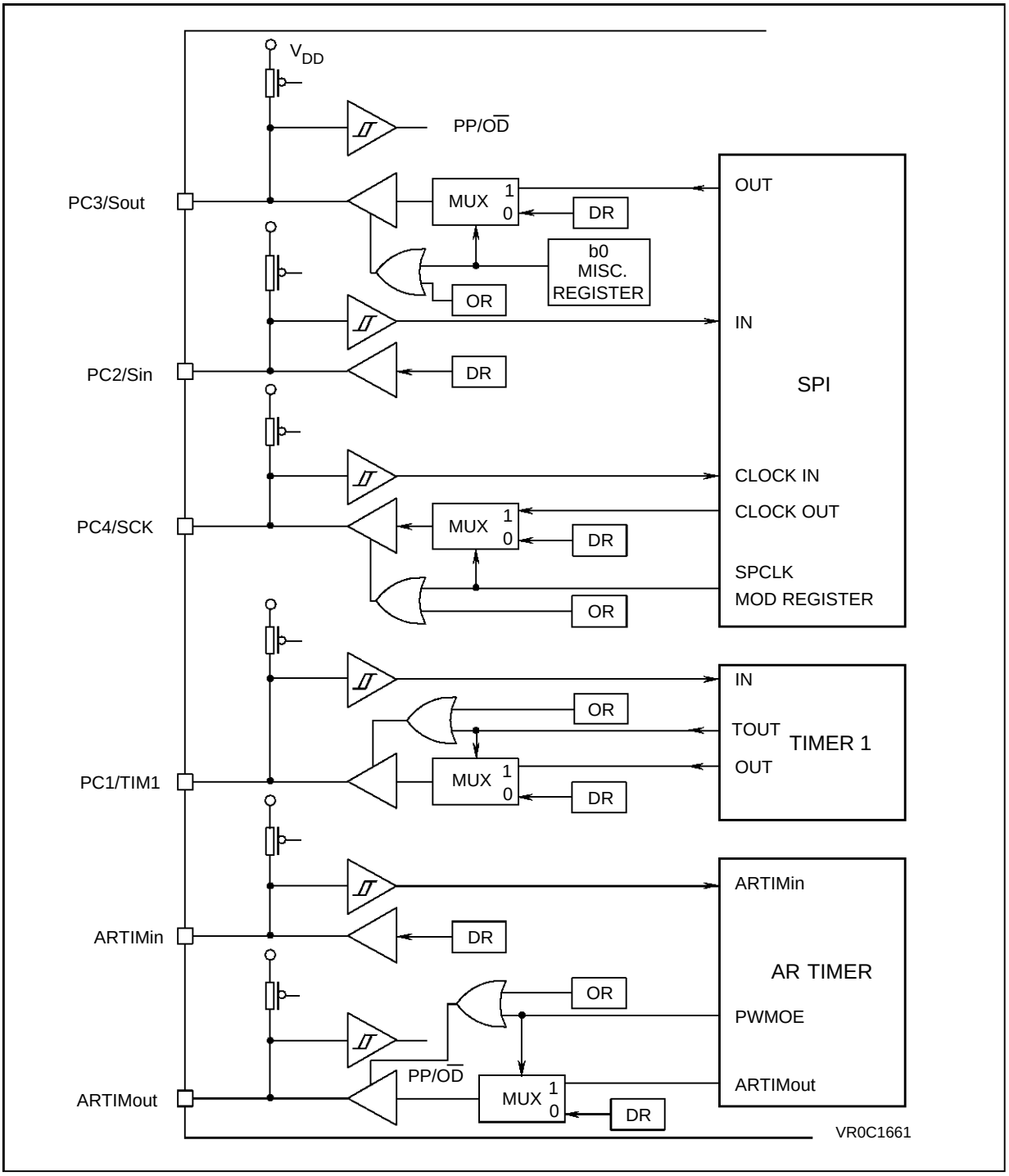
Table 13. I/O Port Option Selections

MODE	AVAILABLE ON(1)	SCHEMATIC
Input	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Input with pull up	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Input with pull up with interrupt	PA0-PA3 PB0-PB3, PB6-PB7 PC2-PC4	
Analog Input	PA0-PA3 PC2-PC4	
Open drain output 5mA Open drain output 20mA	PA0-PA3 PC2-PC4 PB0-PB3, PB6-PB7	
Push-pull output 5mA Push-pull output 20mA	PA0-PA3 PC2-PC4 PB0-PB3, PB6-PB7	

Note 1. Provided the correct configuration has been selected.

I/O PORTS (Cont'd)

Figure 21. Peripheral Interface Configuration of SPI, Timer 1 and AR Timer



4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}.

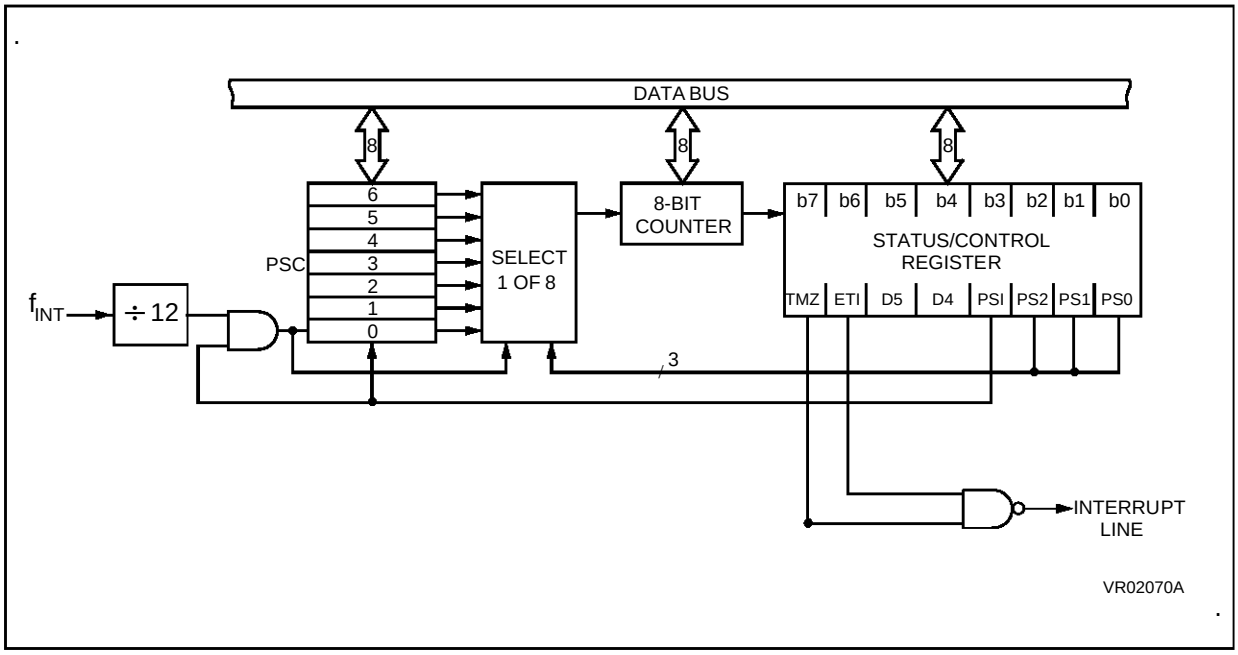
Figure 22 shows the Timer Block Diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero)bit in the TSCR is set. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set, an interrupt request is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input is the internal frequency (f_{INT}) divided by 12. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Table 14), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to allow the prescaler (and hence the counter) to start. If it is cleared, all the prescaler bits are set and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set. The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

Figure 23 illustrates the Timer's working principle.

Figure 22. Timer Block Diagram



TIMER (Cont'd)

4.2.1 Timer Operation

The Timer prescaler is clocked by the prescaler clock input ($f_{INT} \div 12$).

The user can select the desired prescaler division ratio through the PS2, PS1, PS0 bits. When the TCR count reaches 0, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high.

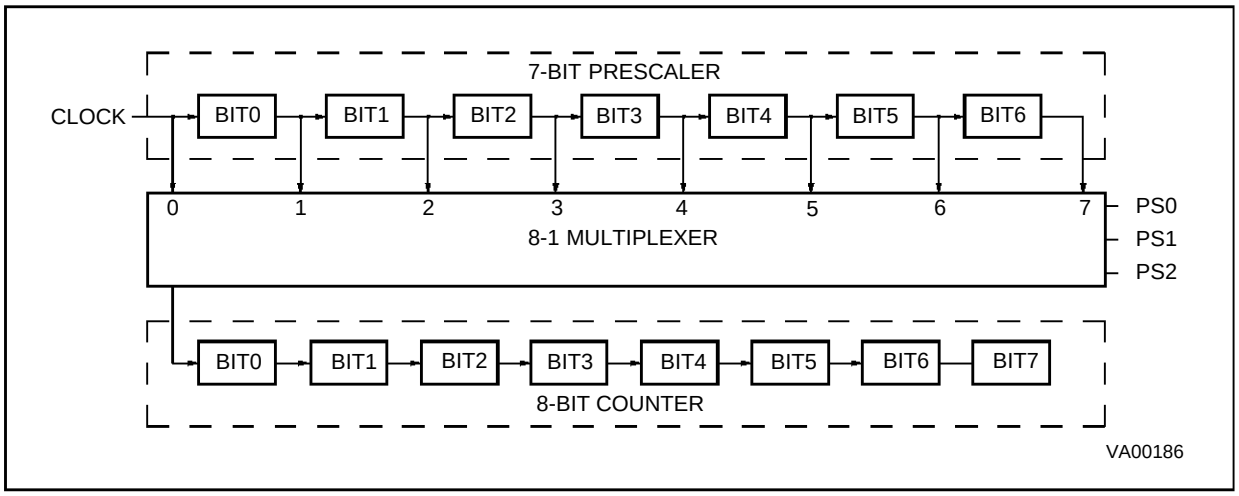
4.2.2 Timer Interrupt

When the counter register decrements to zero with the ETI (Enable Timer Interrupt) bit set to one, an interrupt request associated with Interrupt Vector #3 is generated. When the counter decrements to zero, the TMZ bit in the TSCR register is set to one.

4.2.3 Application Notes

TMZ is set when the counter reaches zero; however, it may also be set by writing 00h in the TCR register or by setting bit 7 of the TSCR register. The TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded with 0FFh, while the 7-bit prescaler is loaded with 07Fh, and the TSCR register is cleared. This means that the Timer is stopped (PSI="0") and the timer interrupt is disabled.

Figure 23. Timer Working Principle



TIMER (Cont'd)

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

4.2.4 Timer Registers

Timer Status Control Register (TSCR)

Address: 0D4h — Read/Write

7							0
TMZ	ETI	D5	D4	PSI	PS2	PS1	PS0

Bit 7 = **TMZ**: *Timer Zero bit*

A low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before starting a new count.

Bit 6 = **ETI**: *Enable Timer Interrup*

When set, enables the timer interrupt request (vector #3). If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.

Bit 5 = **D5**: *Reserved*

Must be reset.

Bit 4 = **D4**

When set, the timer is enabled; when reset the timer is disabled.

Bit 3 = **PSI**: *Prescaler Initialize Bit*

Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSI="0" both counter and prescaler are not running.

Bit 2, 1, 0 = **PS2, PS1, PS0**: *Prescaler Mux. Select.* These bits select the division ratio of the prescaler register.

Table 14. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Timer Counter Register (TCR)

Address: 0D3h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Counter Bits.*

Prescaler Register PSC

Address: 0D2h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 = **D7**: Always read as "0".

Bit 6-0 = **D6-D0**: Prescaler Bits.

4.3 AUTO-RELOAD TIMER

The Auto-Reload Timer (AR Timer) on-chip peripheral consists of an 8-bit timer/counter with compare and capture/reload capabilities and of a 7-bit prescaler with a clock multiplexer, enabling the clock input to be selected as f_{INT}, $f_{INT/3}$ or an external clock source. A Mode Control Register, ARMC, two Status Control Registers, ARSC0 and ARSC1, an output pin, ARTIMout, and an input pin, ARTIMin, allow the Auto-Reload Timer to be used in 4 modes:

– Auto-reload (PWM generation),

– Output compare and reload on external event (PLL),

– Input capture and output compare for time measurement.

– Input capture and output compare for period measurement.

The AR Timer can be used to wake the MCU from WAIT mode either with an internal or with an external clock. It also can be used to wake the MCU from STOP mode, if used with an external clock signal connected to the ARTIMin pin. A Load register allows the program to read and write the counter on the fly.

4.3.1 AR Timer Description

The AR COUNTER is an 8-bit up-counter incremented on the input clock's rising edge. The counter is loaded from the ReLoad/Capture Register, ARRC, for auto-reload or capture operations, as well as for initialization. Direct access to the AR counter is not possible; however, by reading or writing the ARLR load register, it is possible to read or write the counter's contents on the fly.

The AR Timer's input clock can be either the internal clock (from the Oscillator Divider), the internal clock divided by 3, or the clock signal connected to the ARTIMin pin. Selection between these clock sources is effected by suitably programming bits CC0-CC1 of the ARSC1 register. The output of the AR Multiplexer feeds the 7-bit programmable AR Prescaler, ARPSC, which selects one of the 8 available taps of the prescaler, as defined by PSC0-PSC2 in the AR Mode Control Register. Thus the division factor of the prescaler can be set to 2n (where n = 0, 1,..7).

The clock input to the AR counter is enabled by the TEN (Timer Enable) bit in the ARMC register. When TEN is reset, the AR counter is stopped and the prescaler and counter contents are frozen. When TEN is set, the AR counter runs at the rate of the selected clock source. The counter is cleared on system reset.

The AR counter may also be initialized by writing to the ARLR load register, which also causes an immediate copy of the value to be placed in the AR counter, regardless of whether the counter is running or not. Initialization of the counter, by either method, will also clear the ARPSC register, whereupon counting will start from a known value.

4.3.2 Timer Operating Modes

Four different operating modes are available for the AR Timer:

Auto-reload Mode with PWM Generation. This mode allows a Pulse Width Modulated signal to be generated on the ARTIMout pin with minimum Core processing overhead.

The free running 8-bit counter is fed by the prescaler's output, and is incremented on every rising edge of the clock signal.

When a counter overflow occurs, the counter is automatically reloaded with the contents of the Reload/Capture Register, ARCC, and ARTIMout is set. When the counter reaches the value contained in the compare register (ARCP), ARTIMout is reset.

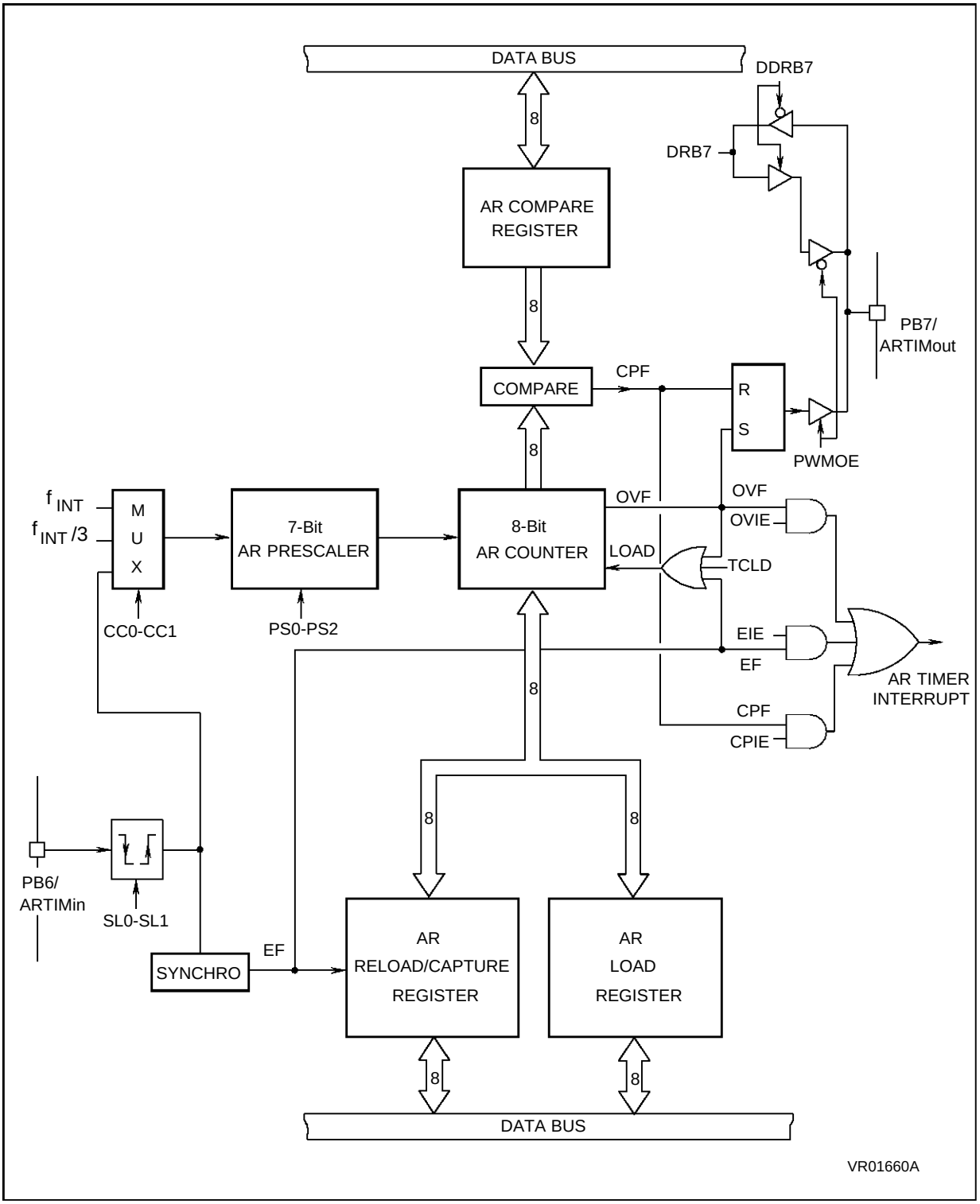
On overflow, the OVF flag of the ARSC0 register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OVIE, in the Mode Control Register (ARMC), is set. The OVF flag must be reset by the user software.

When the counter reaches the compare value, the CPF flag of the ARSC0 register is set and a compare interrupt request is generated, if the Compare Interrupt enable bit, CPIE, in the Mode Control Register (ARMC), is set. The interrupt service routine may then adjust the PWM period by loading a new value into ARCP. The CPF flag must be reset by user software.

The PWM signal is generated on the ARTIMout pin (refer to the Block Diagram). The frequency of this signal is controlled by the prescaler setting and by the auto-reload value present in the Reload/Capture register, ARRC. The duty cycle of the PWM signal is controlled by the Compare Register, ARCP.

AUTO-RELOAD TIMER (Cont'd)

Figure 24. AR Timer Block Diagram



AUTO-RELOAD TIMER (Cont'd)

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of PWM output signal. To obtain a signal on ARTIMout, the contents of the ARCP register must be greater than the contents of the ARRC register.

The maximum available resolution for the ARTIMout duty cycle is:

Resolution = 1/[255-(ARRC)]

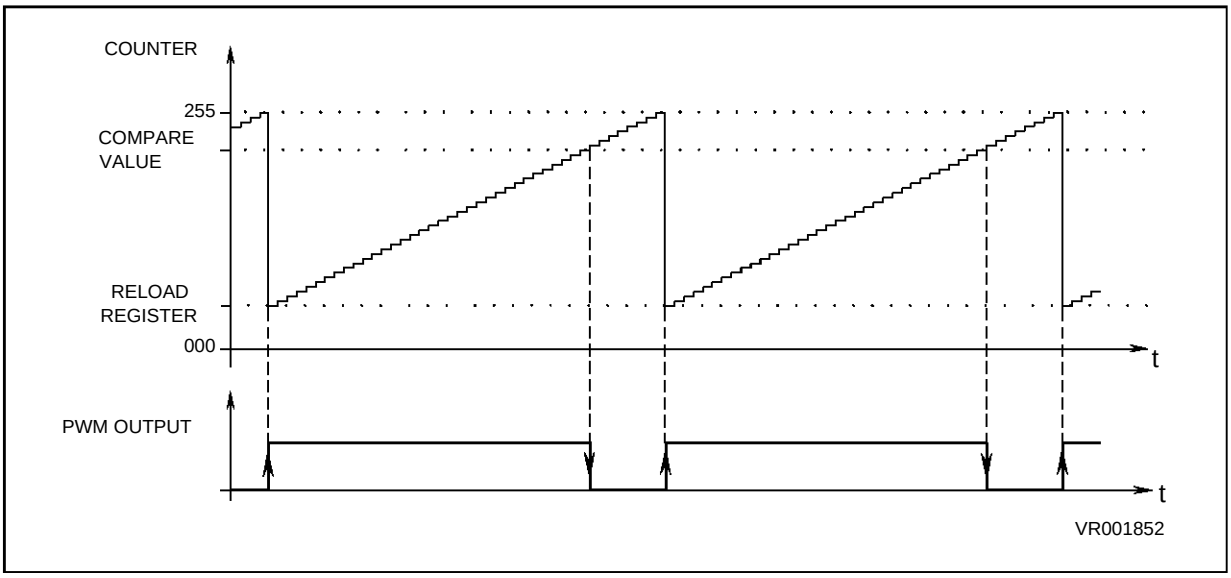
Where ARRC is the content of the Reload/Capture register. The compare value loaded in the Compare Register, ARCP, must be in the range from (ARRC) to 255.

The ARTC counter is initialized by writing to the ARRC register and by then setting the TCLD (Timer Load) and the TEN (Timer Clock Enable) bits in the Mode Control register, ARMC.

Enabling and selection of the clock source is controlled by the CC0, CC1, SL0 and SL1 bits in the Status Control Register, ARSC1. The prescaler division ratio is selected by the PS0, PS1 and PS2 bits in the ARSC1 register.

In Auto-reload Mode, any of the three available clock sources can be selected: Internal Clock, Internal Clock divided by 3 or the clock signal present on the ARTIMin pin.

Figure 25. Auto-reload Timer PWM Function



AUTO-RELOAD TIMER (Cont'd)

Capture Mode with PWM Generation. In this mode, the AR counter operates as a free running 8-bit counter fed by the prescaler output. The counter is incremented on every clock rising edge.

An 8-bit capture operation from the counter to the ARRC register is performed on every active edge on the ARTIMin pin, when enabled by Edge Control bits SL0, SL1 in the ARSC1 register. At the same time, the External Flag, EF, in the ARSC0 register is set and an external interrupt request is generated if the External Interrupt Enable bit, EIE, in the ARMC register, is set. The EF flag must be reset by user software.

Each ARTC overflow sets ARTIMout, while a match between the counter and ARCP (Compare Register) resets ARTIMout and sets the compare flag, CPF. A compare interrupt request is generated if the related compare interrupt enable bit, CPIE, is set. A PWM signal is generated on ARTIMout. The CPF flag must be reset by user software.

The frequency of the generated signal is determined by the prescaler setting. The duty cycle is determined by the ARCP register.

Initialization and reading of the counter are identical to the auto-reload mode (see previous description).

Enabling and selection of clock sources is controlled by the CC0 and CC1 bits in the AR Status Control Register, ARSC1.

The prescaler division ratio is selected by programming the PS0, PS1 and PS2 bits in the ARSC1 Register.

In Capture mode, the allowed clock sources are the internal clock and the internal clock divided by 3; the external ARTIMin input pin should not be used as a clock source.

Capture Mode with Reset of counter and prescaler, and PWM Generation. This mode is identical to the previous one, with the difference that a capture condition also resets the counter and the prescaler, thus allowing easy measurement of the time between two captures (for input period measurement on the ARTIMin pin).

Load on External Input. The counter operates as a free running 8-bit counter fed by the prescaler. the count is incremented on every clock rising edge.

Each counter overflow sets the ARTIMout pin. A match between the counter and ARCP (Compare Register) resets the ARTIMout pin and sets the compare flag, CPF. A compare interrupt request is generated if the related compare interrupt enable bit, CPIE, is set. A PWM signal is generated on ARTIMout. The CPF flag must be reset by user software.

Initialization of the counter is as described in the previous paragraph. In addition, if the external ARTIMin input is enabled, an active edge on the input pin will copy the contents of the ARRC register into the counter, whether the counter is running or not.

Notes:

The allowed AR Timer clock sources are the following:

AR Timer Mode	Clock Sources
Auto-reload mode	f_{INT}, $f_{INT/3}$, ARTIMin
Capture mode	f_{INT}, $f_{INT/3}$
Capture/Reset mode	f_{INT}, $f_{INT/3}$
External Load mode	f_{INT}, $f_{INT/3}$

The clock frequency should not be modified while the counter is counting, since the counter may be set to an unpredictable value. For instance, the multiplexer setting should not be modified while the counter is counting.

Loading of the counter by any means (by auto-reload, through ARLR, ARRC or by the Core) resets the prescaler at the same time.

Care should be taken when both the Capture interrupt and the Overflow interrupt are used. Capture and overflow are asynchronous. If the capture occurs when the Overflow Interrupt Flag, OVF, is high (between counter overflow and the flag being reset by software, in the interrupt routine), the External Interrupt Flag, EF, may be cleared simultaneously without the interrupt being taken into account.

The solution consists in resetting the OVF flag by writing 03h in the ARSC0 register. The value of EF is not affected by this operation. If an interrupt has occured, it will be processed when the MCU exits from the interrupt routine (the second interrupt is latched).

AUTO-RELOAD TIMER (Cont'd)

4.3.3 AR Timer Registers

AR Mode Control Register (ARMC)

Address: D5h — Read/Write

Reset status: 00h

7							0
TCLD	TEN	PWMOE	EIE	CPIE	OVIE	ARMC1	ARMC0

The AR Mode Control Register ARMC is used to program the different operating modes of the AR Timer, to enable the clock and to initialize the counter. It can be read and written to by the Core and it is cleared on system reset (the AR Timer is disabled).

Bit 7 = **TLCD**: *Timer Load Bit.* This bit, when set, will cause the contents of ARRC register to be loaded into the counter and the contents of the prescaler register, ARPSC, are cleared in order to initialize the timer before starting to count. This bit is write-only and any attempt to read it will yield a logical zero.

Bit 6 = **TEN***: Timer Clock Enable.* This bit, when set, allows the timer to count. When cleared, it will stop the timer and freeze ARPSC and ARTSC.

Bit 5 = **PWMOE**: *PWM Output Enable.* This bit, when set, enables the PWM output on the ARTIMout pin. When reset, the PWM output is disabled.

Bit 4 = **EIE**: *External Interrupt Enable.* This bit, when set, enables the external interrupt request. When reset, the external interrupt request is masked. If EIE is set and the related flag, EF, in the ARSC0 register is also set, an interrupt request is generated.

Bit 3 = **CPIE**: *Compare Interrupt Enable.* This bit, when set, enables the compare interrupt request. If CPIE is reset, the compare interrupt request is masked. If CPIE is set and the related flag, CPF, in the ARSC0 register is also set, an interrupt request is generated.

Bit 2 = **OVIE**: *Overflow Interrupt.* This bit, when set, enables the overflow interrupt request. If OVIE is reset, the compare interrupt request is masked. If OVIE is set and the related flag, OVF in the ARSC0 register is also set, an interrupt request is generated.

Bit 1-0 = **ARMC1-ARMC0**: *Mode Control Bits 1-0.* These are the operating mode control bits. The following bit combinations will select the various operating modes:

ARMC1	ARMC0	Operating Mode
0	0	Auto-reload Mode
0	1	Capture Mode
1	0	Capture Mode with Reset of ARTC and ARPSC
1	1	Load on External Edge Mode

AR Timer Status/Control Registers ARSC0 & ARSC1. These registers contain the AR Timer status information bits and also allow the programming of clock sources, active edge and prescaler multiplexer setting.

ARSC0 register bits 0,1 and 2 contain the interrupt flags of the AR Timer. These bits are read normally. Each one may be reset by software. Writing a one does not affect the bit value.

AR Status Control Register 0 (ARSC0)

Address: D6h — Read/Clear

7							0
D7	D6	D5	D4	D3	EF	CPF	OVF

Bits 7-3 = **D7-D3**: *Unused*

Bit 2 = **EF**: *External Interrupt Flag.* This bit is set by any active edge on the external ARTIMin input pin. The flag is cleared by writing a zero to the EF bit.

Bit 1 = **CPF**: *Compare Interrupt Flag.* This bit is set if the contents of the counter and the ARCP register are equal. The flag is cleared by writing a zero to the CPF bit.

Bit 0 = **OVF**: *Overflow Interrupt Flag.* This bit is set by a transition of the counter from FFh to 00h (overflow). The flag is cleared by writing a zero to the OVF bit.

AUTO-RELOAD TIMER (Cont'd)

AR Status Control Register 1(ARSC1)

Address: D7h — Read/Write

7							0
PS2	PS1	PS0	D4	SL1	SL0	CC1	CC0

Bist 7-5 = **PS2-PS0**: *Prescaler Division Selection Bits 2-0.* These bits determine the Prescaler division ratio. The prescaler itself is not affected by these bits. The prescaler division ratio is listed in the following table:

Table 15. Prescaler Division Ratio Selection

PS2	PS1	PS0	ARPSC Division Ratio
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Bit 4 = **D4**: *Reserved*. Must be kept reset.

Bit 3-2 = **SL1-SL0**: *Timer Input Edge Control Bits 1-0.* These bits control the edge function of the Timer input pin for external synchronization. If bit SL0 is reset, edge detection is disabled; if set edge detection is enabled. If bit SL1 is reset, the AR Timer input pin is rising edge sensitive; if set, it is falling edge sensitive.

SL1	SL0	Edge Detection
X	0	Disabled
0	1	Rising Edge
1	1	Falling Edge

Bit 1-0 = **CC1-CC0**: *Clock Source Select Bit 1-0.* These bits select the clock source for the AR Timer through the AR Multiplexer. The programming of the clock sources is explained in the following Table 16:

Table 16. Clock Source Selection.

CC1	CC0	Clock Source
0	0	F_{int}
0	1	F_{int} Divided by 3
1	0	ARTIMin Input Clock
1	1	Reserved

AR Load Register ARLR. The ARLR load register is used to read or write the ARTC counter register "on the fly" (while it is counting). The ARLR register is not affected by system reset.

AR Load Register (ARLR)

Address: DBh — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Load Register Data Bits.* These are the load register data bits.

AR Reload/Capture Register. The ARRC reload/capture register is used to hold the auto-reload value which is automatically loaded into the counter when overflow occurs.

AR Reload/Capture (ARRC)

Address: D9h — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Reload/Capture Data Bits*. These are the Reload/Capture register data bits.

AR Compare Register. The CP compare register is used to hold the compare value for the compare function.

AR Compare Register (ARCP)

Address: DAh — Read/Write

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0**: *Compare Data Bits*. These are the Compare register data bits.

4.4 A/D CONVERTER (ADC)

The A/D converter peripheral is an 8-bit analog to digital converter with analog inputs as alternate I/O functions (the number of which is device dependent), offering 8-bit resolution with a typical conversion time of 70us (at an oscillator clock frequency of 8MHz).

The ADC converts the input voltage by a process of successive approximations, using a clock frequency derived from the oscillator with a division factor of twelve. With an oscillator clock frequency less than 1.2MHz, conversion accuracy is decreased.

Selection of the input pin is done by configuring the related I/O line as an analog input via the Option and Data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously, to avoid device malfunction.

The ADC uses two registers in the data space: the ADC data conversion register, ADR, which stores the conversion result, and the ADC control register, ADCR, used to program the ADC functions.

A conversion is started by writing a "1" to the Start bit (STA) in the ADC control register. This automatically clears (resets to "0") the End Of Conversion Bit (EOC). When a conversion is complete, the EOC bit is automatically set to "1", in order to flag that conversion is complete and that the data in the ADC data conversion register is valid. Each conversion has to be separately initiated by writing to the STA bit.

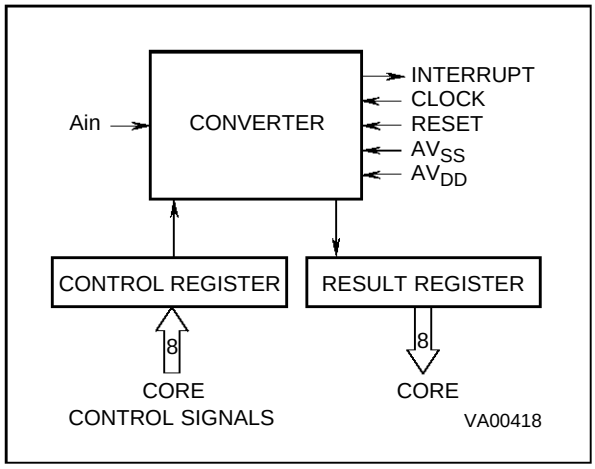
The STA bit is continuously scanned so that, if the user sets it to "1" while a previous conversion is in progress, a new conversion is started before completing the previous one. The start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter features a maskable interrupt associated with the end of conversion. This interrupt is associated with interrupt vector #4 and occurs when the EOC bit is set (i.e. when a conversion is completed). The interrupt is masked using the EAI (interrupt mask) bit in the control register.

The power consumption of the device can be reduced by turning off the ADC peripheral. This is done by setting the PDS bit in the ADC control register to "0". If PDS="1", the A/D is powered and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow stabilisation of the A/D converter. This action is also needed before entering WAIT mode, since the A/D comparator is not automatically disabled in WAIT mode.

During Reset, any conversion in progress is stopped, the control register is reset to 40h and the ADC interrupt is masked (EAI=0).

Figure 26. ADC Block Diagram



4.4.1 Application Notes

The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for the optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 12pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. In worst case conditions, the impedance, ASI, of the analog voltage source is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband. ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

A/D CONVERTER (Cont'd)

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

The accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

The converter resolution is given by::

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

In order to optimise conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the less significant bits are determined.

The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. Indeed, only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. It should be noted that waking up the microcontroller could also be done using the Timer interrupt, but in this case the Timer will be working and the resulting noise could affect conversion accuracy.

A/D Converter Control Register (ADCR)

Address: 0D1h — Read/Write

7							0
EAI	EOC	STA	PDS	D3	D2	D1	D0

Bit 7 = **EAI**: *Enable A/D Interrupt.* If this bit is set to "1" the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

Bit 6 = **EOC**: *End of conversion. Read Only.* This read only bit indicates when a conversion has been completed. This bit is automatically reset to "0" when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to "1".

Bit 5 = **STA***: Start of Conversion. Write Only.* Writing a "1" to this bit will start a conversion on the selected channel and automatically reset to "0" the EOC bit. If the bit is set again when a conversion is in progress, the present conversion is stopped and a new one will take place. This bit is write only, any attempt to read it will show a logical zero.

Bit 4 = **PDS***: Power Down Selection.* This bit activates the A/D converter if set to "1". Writing a "0" to this bit will put the ADC in power down mode (idle mode).

Bit 3-0 = **D3-D0.** Not used

A/D Converter Data Register (ADR)

Address: 0D0h — Read only

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7-0 = **D7-D0***: 8 Bit A/D Conversion Result.*

4.5 SERIAL PERIPHERAL INTERFACE (SPI)

The SPI peripheral is an optimized synchronous serial interface with programmable transmission modes and master/slave capabilities supporting a wide range of industry standard SPI specifications. The SPI interface may also implement asynchronous data transfer, in which case processor overhead is limited to data transfer from or to the shift register on an interrupt driven basis.

The SPI may be controlled by simple user software to perform serial data exchange with low-cost external memory, or with serially controlled peripherals to drive displays, motors or relays.

The SPI's shift register is simultaneously fed by the Sin pin and feeds the Sout pin, thus transmission and reception are essentially the same process. Suitable setting of the number of bits in the data frame can allow filtering of unwanted leading data bits in the incoming data stream.

The SPI comprises an 8-bit Data/Shift Register, DSR, a Divide register, DIV, a Mode Control Register MOD, and a Miscellaneous register, MISCR.

The SPI may be operated either in Master mode or in Slave mode.

Master mode is defined by the synchronous serial clock being supplied by the MCU, by suitably programming the clock divider (DIV register). Slave mode is defined by the serial clock being supplied externally on the SCK pin by the external Master device.

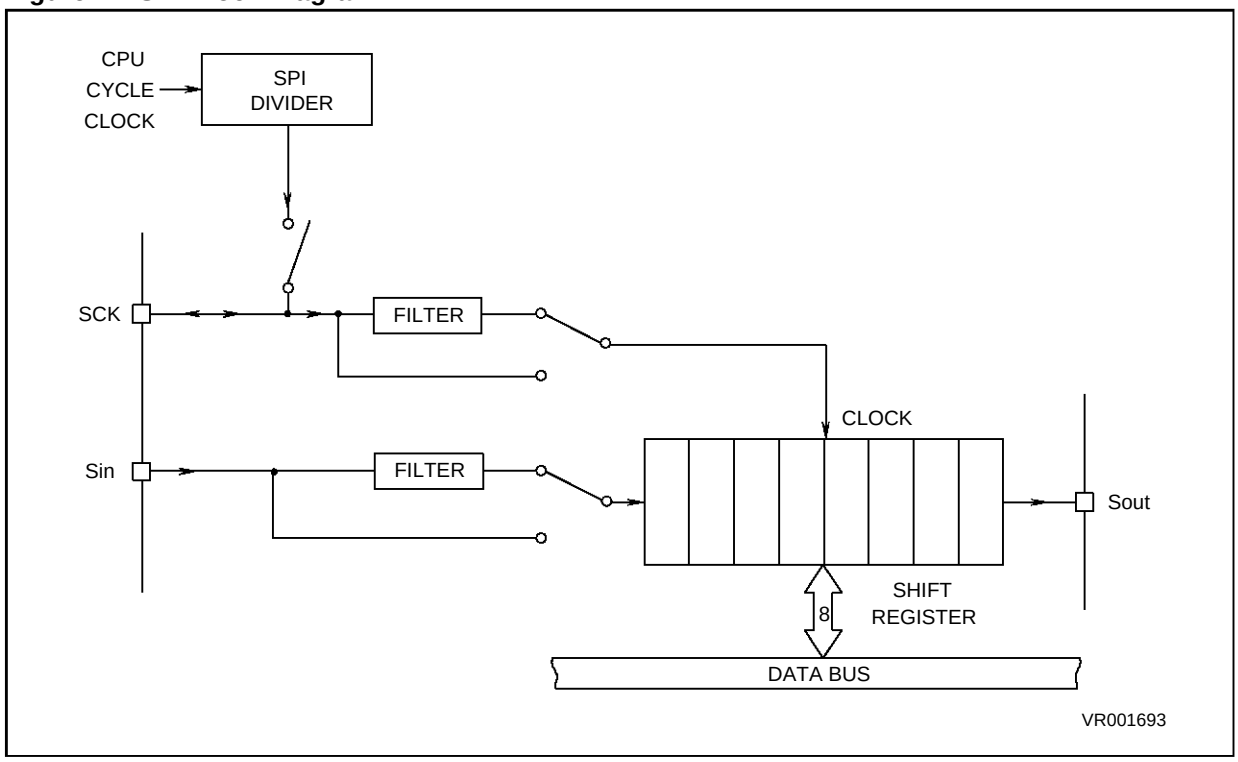
For maximum versatility the SPI may be programmed to sample data either on the rising or on the falling edge of SCK, with or without phase shift (clock Polarity and Phase selection).

The Sin, Sout and SCK signals are connected as alternate I/O pin functions.

For serial input operation, Sin must be configured as an input. For serial output operation, Sout is selected as an output by programming Bit 0 of the Miscellaneous Register: clearing this bit will set the pin as a standard I/O line, while setting the bit will select the Sout function.

An interrupt request may be associated with the end of a transmission or reception cycle; this is defined by programming the number of bits in the data frame and by enabling the interrupt. This request is associated with interrupt vector #2, and can be masked by programming the SPIE bit of the MOD register. Since the SPI interrupt is "ORed" with the port interrupt source, an interrupt flag bit is available in the DIV register allowing discrimination of the interrupt request.

Figure 27. SPI Block Diagram



SERIAL PERIPHERAL INTERFACE SPI (Cont'd)

4.5.1 SPI Registers

SPI Mode Control Register (MOD)

Address: E2h — Read/Write

Reset status: 00h

7							0
SPRUN	SPIE	CPHA	SPCLK	SPIN	SPSTRT	EFILT	CPOL

The MOD register defines and controls the transmission modes and characteristics.

This register is read/write and all bits are cleared at reset. Setting SPSTRT = 1 and SPIN = 1 is not allowed and must be avoided.

Bit 7 = **SPRUN**: *SPI Run*. This bit is the SPI activity flag. This can be used in either transmit or receive modes; it is automatically cleared by the SPI at the end of a transmission or reception and generates an interrupt request (providing that the SPIE Interrupt Enable bit is set). The Core can stop transmission or reception at any time by resetting the SPRUN bit; this will also generate an interrupt request (providing that the SPIE Interrupt enable bit is set). The SPRUN bit can be used as a start condition parameter, in conjunction with the SPSTRT bit, when an external signal is present on the Sin pin. Note that a rising edge is then necessary to initiate reception; this may require external data inversion.

Bit 6 = **SPIE**: *SPI Interrupt Enable*. This bit is the SPI Interrupt Enable bit. If this bit is set the SPI interrupt (vector #2) is enabled, when SPIE is reset, the interrupt is disabled.

Bit 5 = **CPHA**: *Clock Phase Selection*. This bit selects the clock phase of the clock signal. If this bit is cleared to zero the normal state is selected; in this case Bit 7 of the data frame is present on Sout pin as soon as the SPI Shift Register is loaded. If this bit is set to one the shifted state' is selected; in this case Bit 7 of data frame is present on Sout pin on the first falling edge of Shift Register clock. The polarity relation and the division ratio between Shift Register and SPI base clock are also programmable; refer to DIV register and Timing Diagrams for more information.

Bit 4= **SPCLK**: *Base Clock Selection*

This bit selects the SPI base clock source. It is either the core cycle clock (f_{INT}/13) (Master mode) or the signal provided at SCK pin by an external device (slave mode). If SPCLK is low and the SCK pin is configured as input, the slave mode is selected. If SPCLK is high and the SCK pin is configured as output, the master mode is selected. In this case, the phase and polarity of the clock are controlled by CPOL and CPHA.

Bit 3 = **SPIN**: *Input Selection*

This bit enables the transfer of the data input to the Shift Register in receive mode. If this bit is cleared the Shift Register input is 0. If this bit is set, the Shift Register input corresponds to the input signal present on the Sin pin.

Bit 2 = **SPSTRT**: *Start Selection*

This bit selects the transmission or reception start mode. If SPSTRT is cleared, the internal start condition occurs as soon as the SPRUN bit is set. If SPSTRT is set, the internal start signal is the logic "AND" between the SPRUN bit and the external signal present on the Sin pin; in this case transmission will start after the latest of both signals providing that the first signal is still present (note that this implies a rising edge). After the transmission or recetion has been started, it will continue even if the Sin signal is reset.

Bit 1 = **EFILT**: *Enable Filters*

This bit enables/disables the input noise filters on the Sin and SCK inputs. If it is cleared to zero the filters are disabled, if set to one the filters are enabled. These noise filters will eliminate any pulse on Sin and SCK with a pulse width smaller than one to two Core clock periods (depending on the occurrence of the signal edge with respect to the Core clock edge). For example, if the ST6260B/65B runs with an 8MHz crystal, Sin and SCK will be delayed by 125 to 250ns.

Bit 0 = **CPOL**: *Clock Polarity*

This bit controls the relationship between the data on the Sin and Sout pins and SCK. The CPOL bit selects the clock edge which captures data and allows it to change state. It has the greatest impact on the first bit transmitted (the MSB) as it does (or does not) allow a clock transition before the first data capture edge.

Refer to the timing diagrams at the end of this section for additional details. These show the relationship between CPOL, CPHA and SCK, and indicate the active clock edges and strobe times.

SERIAL PERIPHERAL INTERFACE SPI (Cont'd)

SPI DIV Register (DIV)

Address: E1h — Read/Write

Reset status: 00h

7							0
SPINT	DOV6	DIV5	DIV4	DIV3	CD2	CD1	CD0

The SPIDIV register defines the transmission rate and frame format and contains the interrupt flag.

Bits CD0-CD2, DIV3-DIV6 are read/write while SPINT can be read and cleared only. Write access is not allowed if SPRUN in the MOD register is set.

Bit 7 = **SPINT**: *Interrupt Flag.* It is automatically set to one by the SPI at the end of a transmission or reception and an interrupt request can be generated depending on the state of the interrupt mask bit in the MOD control register. This bit is read-only and must be cleared by user software at the end of the interrupt service routine.

Bit 6-3 = **DIV6-DIV3**: *Burst Mode Bit Clock Period Selection.* Define the number of shift register bits that are transmitted or received in a frame. The available selections are listed in Table 18. The normal maximum setting is 8 bits, since the shift register is 8 bits wide. Note that by setting a greater number of bits, in conjunction with the SPIN bit in the MOD register, unwanted data bits may be filtered from the data stream.

Bit 2-0 = **CD2-CD0**: *Base/Bit Clock Rate Selection.* Define the division ratio between the core clock (f_{INT} divided by 13) and the clock supplied to the Shift Register in Master mode.

Table 17. Base/Bit Clock Ratio Selection

CD2-CD0			Divide Ratio (decimal)
0	0	0	Divide by 1
0	0	1	Divide by 2
0	1	0	Divide by 4
0	1	1	Divide by 8
1	0	0	Divide by 16
1	0	1	Divide by 32
1	1	0	Divide by 64
1	1	1	Divide by 256

Note: For example, when an 8MHz CPU clock is used, asynchronous operation at 9600 Baud is possible (8MHz/13/64). Other Baud rates are available by proportionally selecting division factors depending on CPU clock frequency.

Data setup time on Sin is typically 250ns min, while data hold time is typically 50ns min.

Table 18. Burst Mode Bit Clock Periods

DIV6-DIV3				Number of bits sent
0	0	0	0	Reserved (not to be used)
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Refer to the description of the DIV6-DIV3 bits in the DIV Register

SPI Data/Shift Register (SPIDSR)

Address: E0h — Read/Write

Reset status: XXh

7							0
D7	D6	D5	D4	D3	D2	D1	D0

SPIDSR is read/write, however write access is not allowed if the SPRUN bit of Mode Control register is set to one.

Data is sampled into SPDSR on the SCK edge determined by the CPOL and CPHA bits. The affect of these setting is shown in the following diagrams.

The Shift Register transmits and receives the Most Significant Bit first.

Bit 7-0 = **DSR7-DSR0**: *Data Bits.* These are the SPI shift register data bits.

Miscellaneous Register (MISCR)

Address: DDh — Read/Write

Reset status: xxxxxxxb

7							0
-	-	-	-	-	-	-	D0

Bit 7-1 = **D7-D1**: *Reserved.*

Bit 0 = **D0**: *Bit 0.* This bit, when set, selects the Sout pin as the SPI output line. When this bit is cleared, Sout acts as a standard I/O line.

SERIAL PERIPHERAL INTERFACE SPI (Cont'd)

4.6 SPI Timing Diagrams

Figure 28. CPOL = 0 Clock Polarity Normal, CPHA = 0 Phase Selection Normal

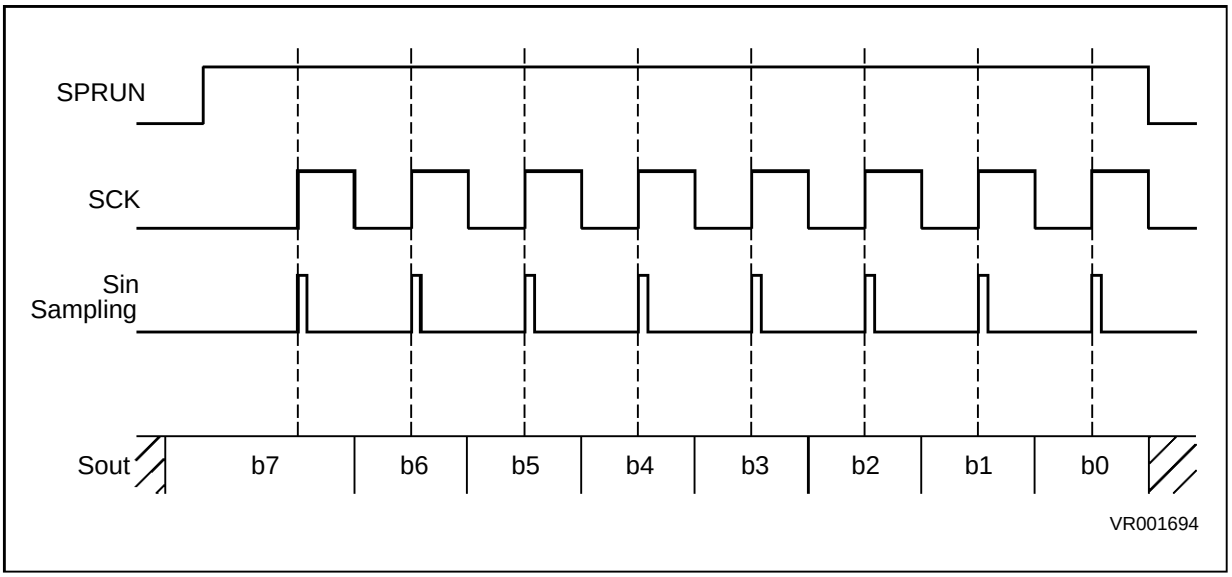
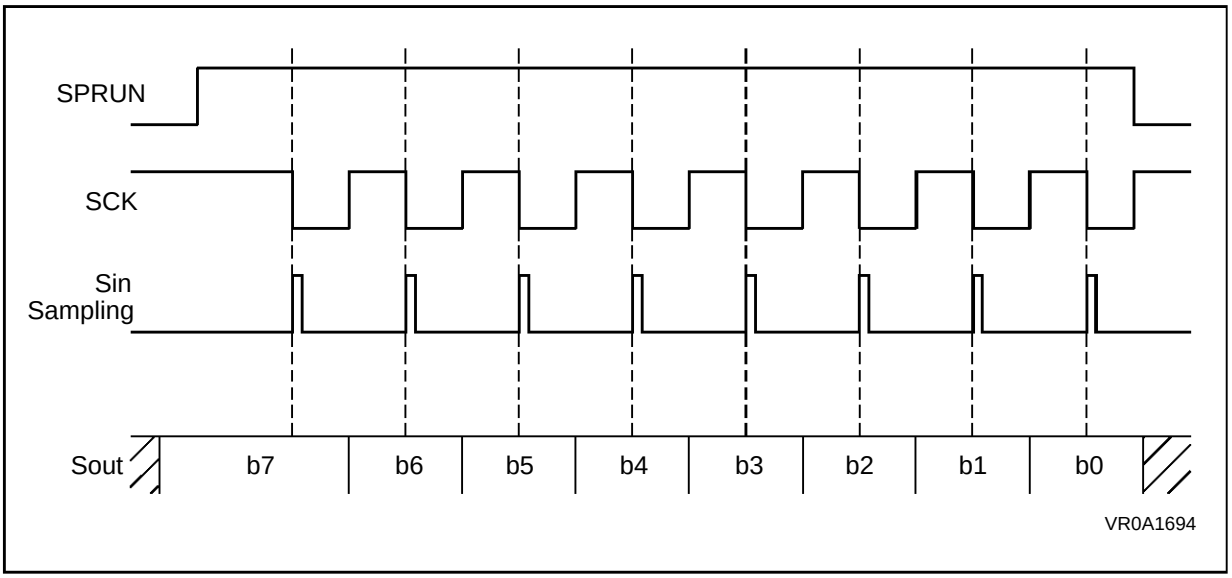


Figure 29. CPOL = 1 Clock Polarity Inverted, CPHA = 0 Phase Selection Normal



SERIAL PERIPHERAL INTERFACE SPI (Cont'd)

Figure 30. CPOL = 0 Clock Polarity Normal, CPHA = 1 Phase Selection Shifted

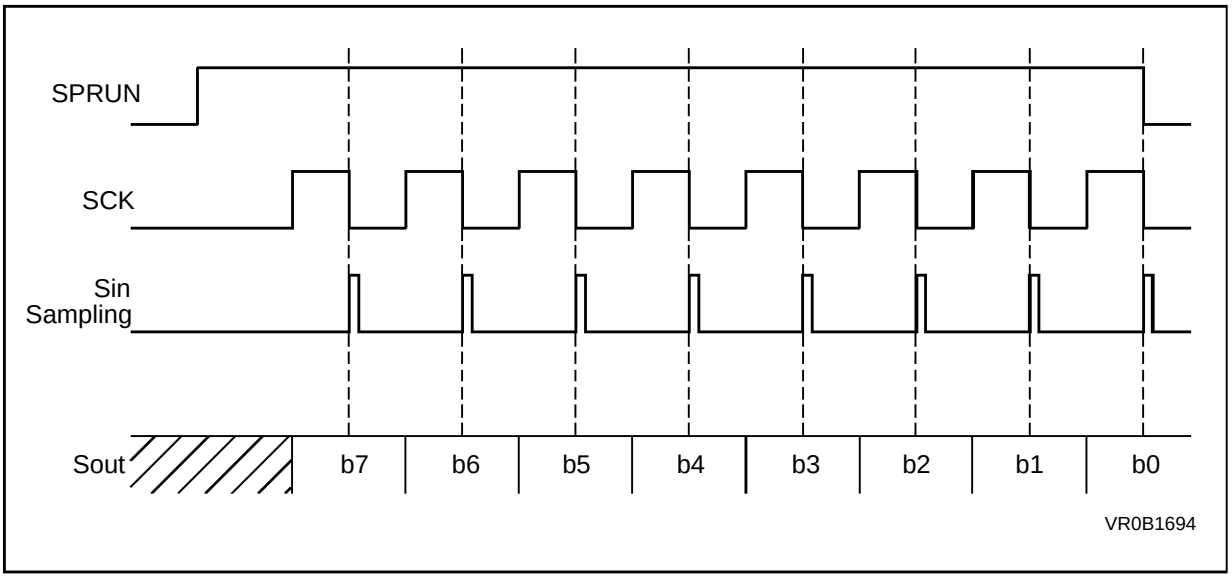
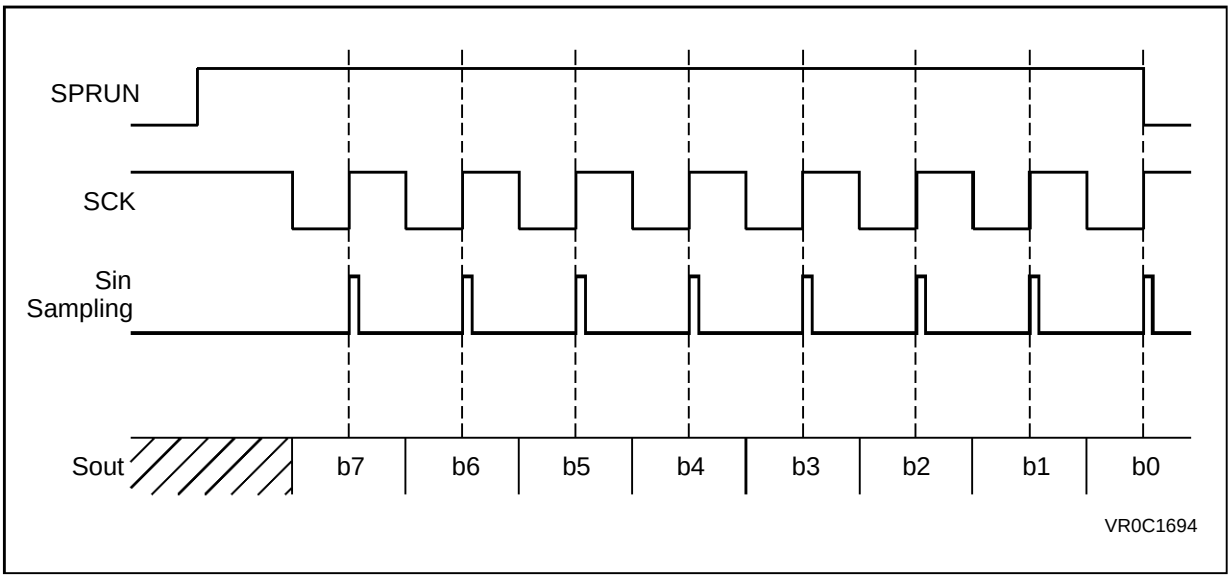


Figure 31. CPOL = 1 Clock Polarity Inverted, CPHA = 1 Phase Selection Shifted



5 SOFTWARE

5.1 ST6 ARCHITECTURE

The ST6 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short, to provide byte efficient programming capability. The ST6 core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

5.2 ADDRESSING MODES

The ST6 core offers nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits which characterize the kind of the test, one bit which determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits which give the span of the branch (0h to Fh) which must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

5.3 INSTRUCTION SET

The ST6 core offers a set of 40 basic instructions which, when combined with nine addressing modes, yield 244 usable opcodes. They can be divided into six different types: load/store, arithmetic/logic, conditional branch, control instructions, jump/call, and bit manipulation. The following paragraphs describe the different types.

All the instructions belonging to a given type are presented in individual tables.

Load & Store. These instructions use one, two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data.

Table 19. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
* . Not Affected

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 20. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y.Indirect Register Pointers, V & W Short Direct RegistersD. Affected
. Immediate data (stored in ROM memory)* . Not Affected
rr. Data space register

INSTRUCTION SET (Cont'd)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. The control instructions control the MCU operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space.

Table 21. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16<F128M>
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ . Affected. The tested bit is shifted into carry.
* . Not Affected

Table 22. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
* . Not<M> Affected

Table 23. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated<N>and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ . Affected
*. Not Affected

Table 24. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
* . Not Affected

Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

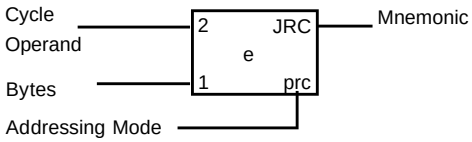
LOW / HI	**0** **0000**	**1** **0001**	**2** **0010**	**3** **0011**	**4** **0100**	**5** **0101**	**6** **0110**	**7** **0111**	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1** **0001**
2 **0010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2** **0010**
3 **0011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3** **0011**
4 **0100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4** **0100**
5 **0101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5** **0101**
6 **0110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	**6** **0110**
7 **0111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	**7** **0111**
8 **1000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	**8** **1000**
9 **1001**	2 RNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	**9** **1001**
A **1010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A** **1010**
B **1011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B** **1011**
C **1100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C** **1100**
D **1101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D** **1101**
E **1110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	**E** **1110**
F **1111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	**F** **1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary. (Continued)

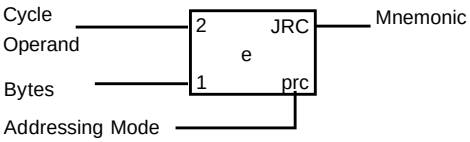
LOW / HI	**8 1000**	**9 1001**	**A 1010**	**B 1011**	**C 1100**	**D 1101**	**E 1110**	**F 1111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**F 1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5 Bit Displacement
b	3 Bit Address
rr	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advisable to take normal precaution to avoid application of any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations.The average chip-junction temperature, Tj, in Celsius can be obtained from:

Tj=TA + PD x RthJA

Where:TA = Ambient Temperature.

RthJA =Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint =IDD x VDD (chip internal power).

Pport =Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	VSS - 0.3 to VDD + 0.3(1)	V
V_O	Output Voltage	VSS - 0.3 to VDD + 0.3(1)	V
I_O	Current Drain per Pin Excluding VDD, VSS	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, VDD = 4.5V	+5	mA
I_{INJ-}	Pin Injection current (negative), All I/O, VDD = 4.5V	-5	mA
IV_{DD}	Total Current into VDD (source)	50(2)	mA
IV_{SS}	Total Current out of VSS (sink)	50(2)	mA
T_j	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Notes:

- Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- (1) Within these limits, clamping diodes are guarantee to be not conductive. Voltages outside these limits are authorised as long as injection current is kept within the specification.

(2) The total current through ports A and B combined may not exceed 50mA. If the application is designed with care and observing the limits stated above, total current may reach 50mA.

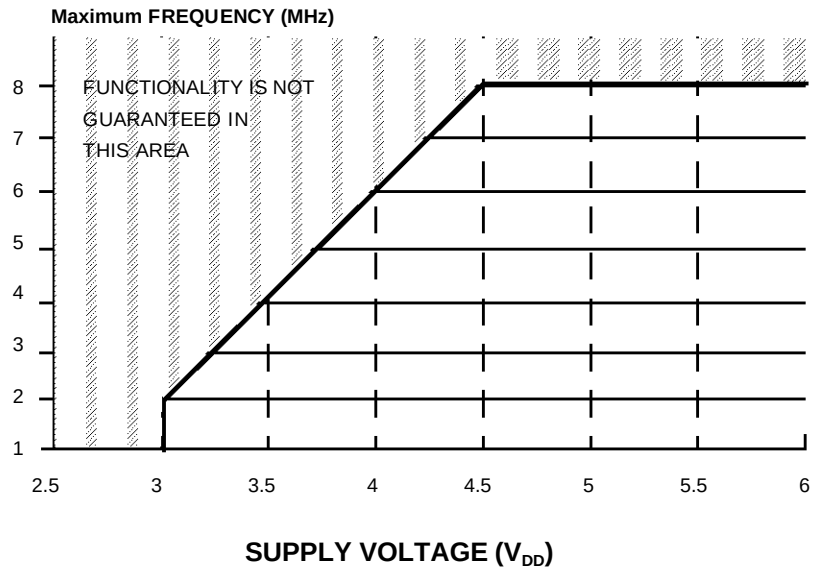
THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP20			60	°C/W
		PSO20			80	

6.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version	-40 0		85 70	°C
V_{DD}	Operating Supply Voltage		3.0V		6.0V	V
V_{PP}	Programming Voltage		12	12.5	13	V
I_{INJ+}	Pin Injection Current (positive) Digital Input	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input	V_{DD} = 4.5 to 5.5V			-5	mA

Figure 32. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

6.3 DC TELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage All inputs				V_{DD} x 0.3	V
V_{IH}	Input High Level Voltage All inputs		V_{DD} x 0.7			V
V_{Hys}	Hysteresis Voltage (4) All Inputs	V_{DD}= 5V V_{DD}= 3V	0.2 0.2			V
V_{OL}	Low Level Output Voltage Port A, C	V_{DD} = 4.5V I_{OL} = +1.6mA V_{DD} = 4.5V I_{OL} = +5.0mA V_{DD} = 3.0 V I_{OL} = +0.7mA			0.4 1.3 0.4	V
V_{OL}	Low Level Output Voltage Port B	V_{DD} = 4.5V I_{OL} = +1.6mA V_{DD} = 4.5V I_{OL} = +20.0mA V_{DD} = 3.0 V I_{OL} = +0.7mA			0.4 1.3 0.4	V
V_{OH}	High Level Output Voltage Port A, B, C	V_{DD} = 4.5V I_{OL} = -1.6mA V_{DD} = 4.5V I_{OL} = -5.0mA V_{DD} = 3.0 V I_{OL} = -0.7mA	4.1 3.5 2.6			V
I_{PU}	Input Pull-up Current Input Mode with Pull-up Port A, B, C, NMI	V_{IN} = V_{SS}, V_{DD} = 2.5 - 6V			100	kΩ
I_{IL} I_{IH}	Input Leakage Current(1)	V_{IN} = V_{SS} V_{IN} = V_{DD}			1.0	µA
I_{DD}	Supply Current in RESET Mode	V_{RESET}= V_{SS} f_{OSC}= 8MHz			3.5	mA
	Supply Current in RUN Mode (2)	V_{DD} = 5.0V f_{INT}=8MHz V_{DD} = 3.0V f_{INT}=4MHz			6.6 TBD	mA
	Supply Current in WAIT Mode (3)	V_{DD}= 5.0V f_{INT}=8MHz V_{DD}= 3.0V f_{INT}=4MHz			1.50 TBD	mA
	Standard STOP Mode Consumption Option (3)	I_{LOAD}= 0mA V_{DD}= 6.0V; 70°C			10	µA
	Low STOP Mode Consumption Option (3)	I_{LOAD}= 0mA V_{DD}= 3.0V; 70°C			2	µA

Notes:
1. Only when pull-ups are not inserted.
2. All peripherals running
3. EEPROM and A/D Converter in Stand-by
4.Hysteresis voltage between switching levels

6.4 AC TELECTRICAL CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Oscillator Frequency	V_{DD} = 3.0V V_{DD} = 4.5V			4 8	MHz
t_{SU}	Oscillator Start-up Time at Power On (2)	Ceramic Resonator $C_{L1} = C_{L2}$ = 22pF		5	100	ms
t_{SUS}	Oscillator STOP mode Recovery Time (2)	8MHz Ceramic Resonator CL1=C_{L2}=22pF		0.2	100	ms
		8MHz Quartz CL1=C_{L2}=22pF		10	100	ms
t_{REC}	Supply Recovery Time (1)		100			ms
T_{WR}	Minimum Pulse Width (V_{DD} = 5V) RESET pin NMI pin		 100 100			ns
T_{WEE}	EEPROM Write Time	T_A = 25°C T_A = 85°C T_A = 125°C		5 10 20	10 20 30	ms
Endurance	EEPROM WRITE/ERASE Cycle	Q_A L_{OT} Acceptance	300,000			cycles
Retention	EEPROM Data Retention	T_A = 25°C	10			years
C_{IN}	Input Capacitance	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance	All Outputs Pins			10	pF

Notes:
1. Period for which V_{DD} has to be connected at 0V to allow internal Reset function at next power-up.
2. This value is highly dependent on the Ceramic Resonator or Quartz Crystal used in the application. See Figure 33.

Figure 33. Power-on-Reset

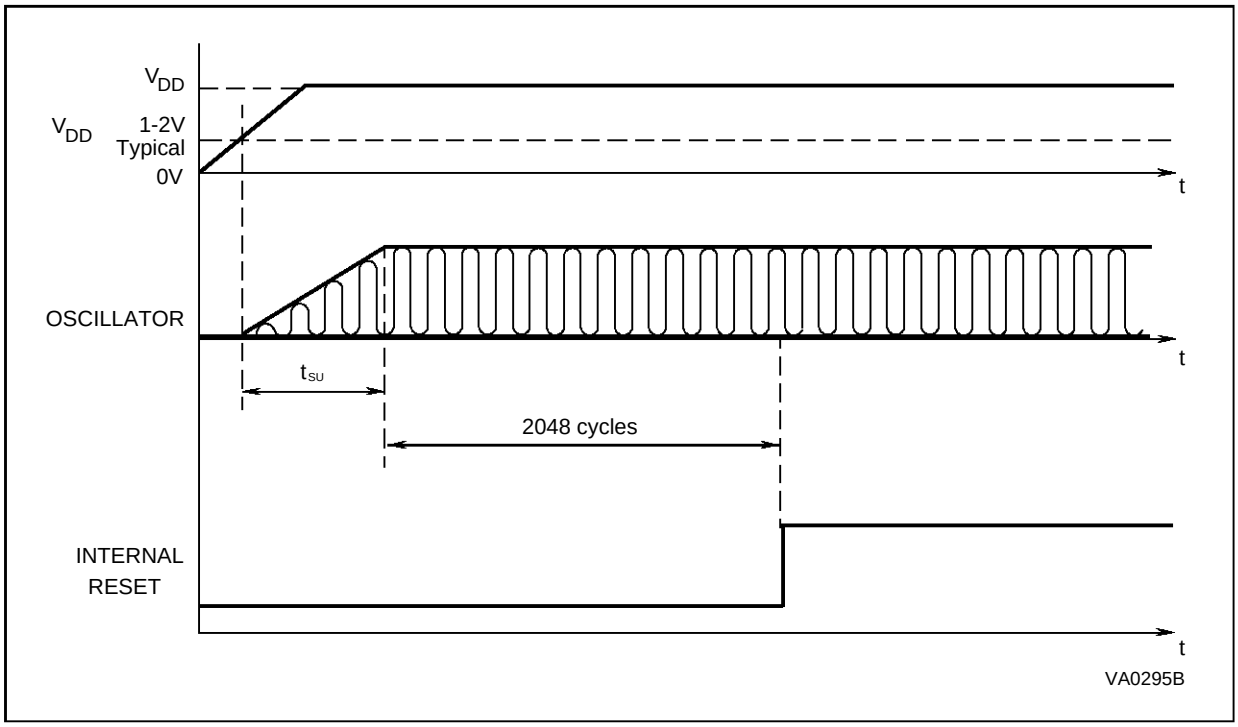
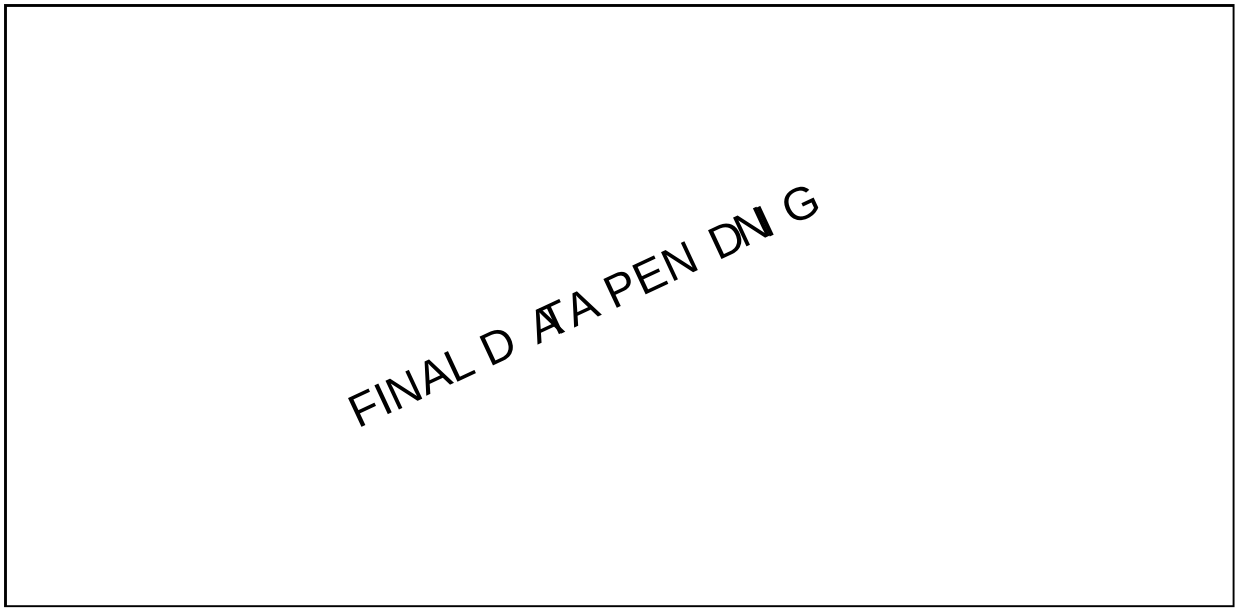
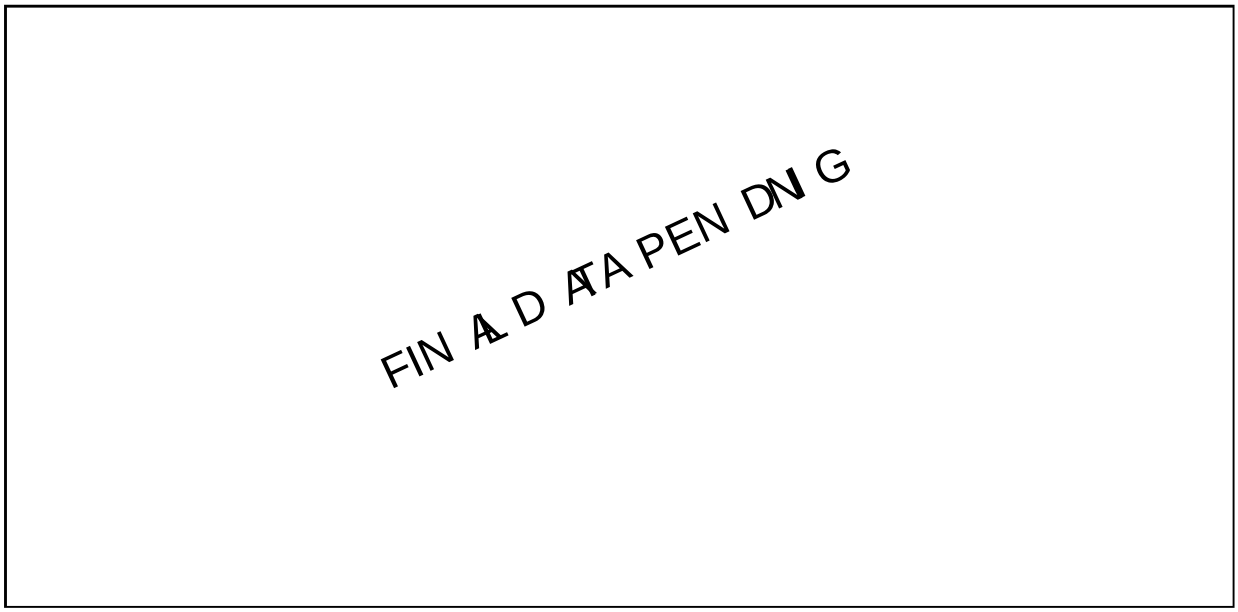


Figure 34. RC Oscillator. F_{INT} versus RNET (Indicative Values)



The shaded area is outside the ST6210B, ST6215B, ST6220B and ST6225B recommended operating range; device functionality is not guaranteed under these conditions.

Figure 35. RC Oscillator. F_{INT} versus RNET (Indicative Values)

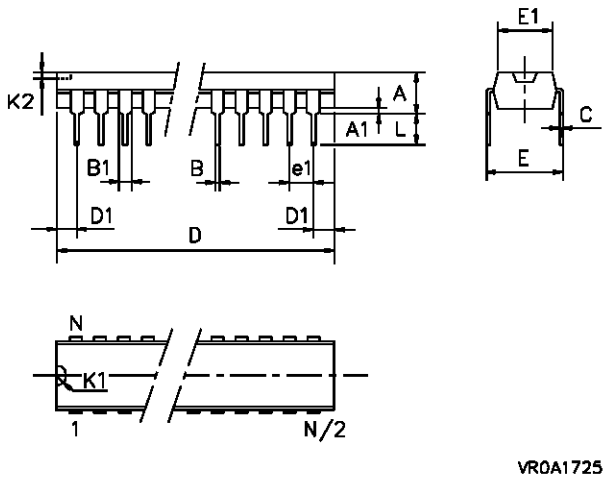


Note: The shaded area is outside the ST6210B, ST6215B, ST6220B and ST6225B recommended operating range; device functionality is not guaranteed under these conditions.

7 GENERAL INFORMATION

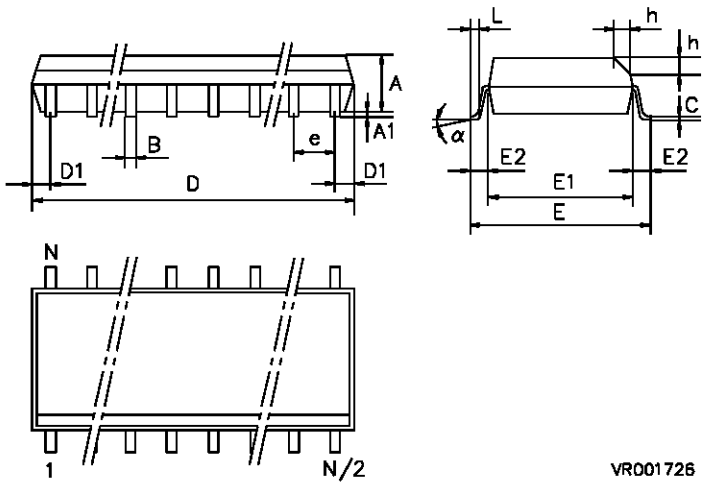
7.1 PACKAGE MECHANICAL DATA

Figure 36. 20-Pin Plastic Dual In-Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
A2	-	-	-	-	-	-
B		0.45			0.018	
B1	1.39		1.65	0.055		0.065
C		0.25			0.010	
D			25.4			1.000
e3		22.86			0.900	
E1			7.1			0.280
e		2.54			0.100	
eB		8.5			0.335	
K1	-	-	-	-	-	-
K2	-	-	-	-	-	-
L		3.3			0.130	
	Number of Pins					
N	20					

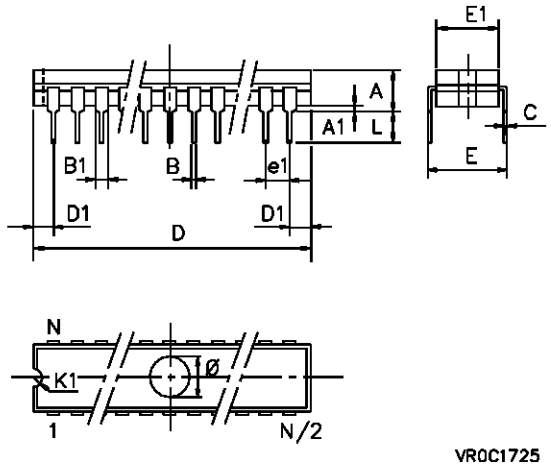
Figure 37. 20-Pin Plastic Small Outline Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

PACKAGE MECHANICAL DATA (Cont'd)

Figure 38. 20-Pin Ceramic Dual In Line Package, 300-mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.093		0.104
A1	0.10		0.30	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.60		13.00	0.496		0.512
D1	-	-	-	-	-	-
E	10.00		10.65	0.394		0.419
E1	7.40		7.60	0.291		0.299
E2	-	-	-	-	-	-
e		1.27			0.050	
h	0.25		0.75	0.010		0.030
L	0.40		1.27	0.016		0.050
∝	0°		8°			
	Number of Pins					
N	20					

7.2 .ORDERING INFORMATION

Table 25. OTP/EPROM VERSION ORDERING INFORMATION

Sales Type	Program Memory (Bytes)	I/O	Additional Features	Temperature Range	Package
ST62E60BF1/XXX	3884 (EPROM)	13	A/D CONVERTER SPI	0 to +70°C	CDIP20
ST62T60BB6/XXX	3884 (OTP)			-40 to + 85°C	PDIP20
ST62T60BM6/XXX					PSO20
ST62T63BB6/XXX	1836 (OTP)		A/D CONVERTER		PDIP20
ST62T63BM6/XXX					PSO20

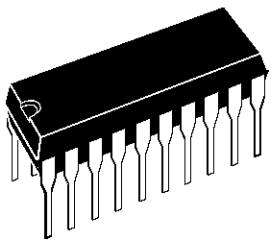
Notes:

SGS-THOMSON MICROELECTRONICS

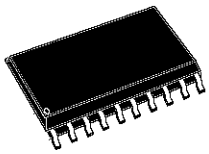
ST6260B
ST6263B

8-BIT MCUs WITH A/D CONVERTER, AUTO-RELOAD TIMER, EEPROM AND SPI

- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- Data RAM: 64/128 bytes
- Data EEPROM: 64/128 bytes
- 13 I/O pins, fully programmable as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull output
  - Analog Input
- 6 I/O lines can sink up to 20mA to drive LEDs or TRIACs directly
- 8-bit Timer/Counter with 7-bit programmable prescaler
- 8-bit Auto-reload Timer with 7-bit programmable prescaler (AR Timer)
- Digital Watchdog
- 8-bit A/D Converter with 7 analog inputs
- 8-bit Synchronous Peripheral Interface (SPI)
- On-chip Clock oscillator can be driven by Quartz Crystal Ceramic resonator or RC network
- User configurable Power-on Reset
- One external Non-Maskable Interrupt
- ST626x-EMU2 Emulation and Development System (connects to an MS-DOS PC via an RS232 serial line).



PDIP20



PSO20

(See end of Datasheet for Ordering Information)

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST6263B and ST6260B are mask programmed ROM versions of ST62T63B and ST62T60B OTP devices.

They offer the same functionality as OTP devices, selecting as ROM options the options defined in the programmable option byte of the OTP version.

1.2 ROM READOUT PROTECTION

If the ROM READOUT PROTECTION option is selected, a protection fuse can be blown to prevent any access to the program memory content.

In case the user wants to blow this fuse, high voltage must be applied on the TEST pin.

Figure 1. Programming wave form

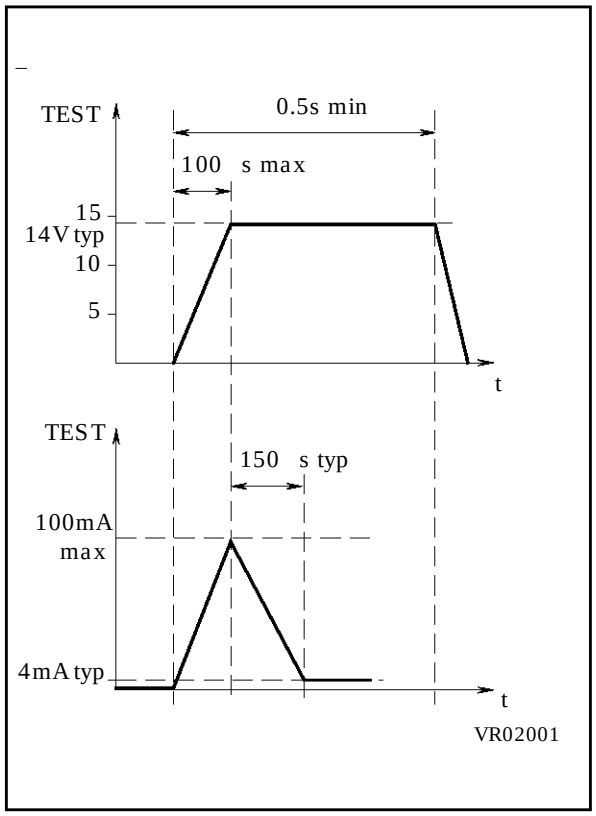
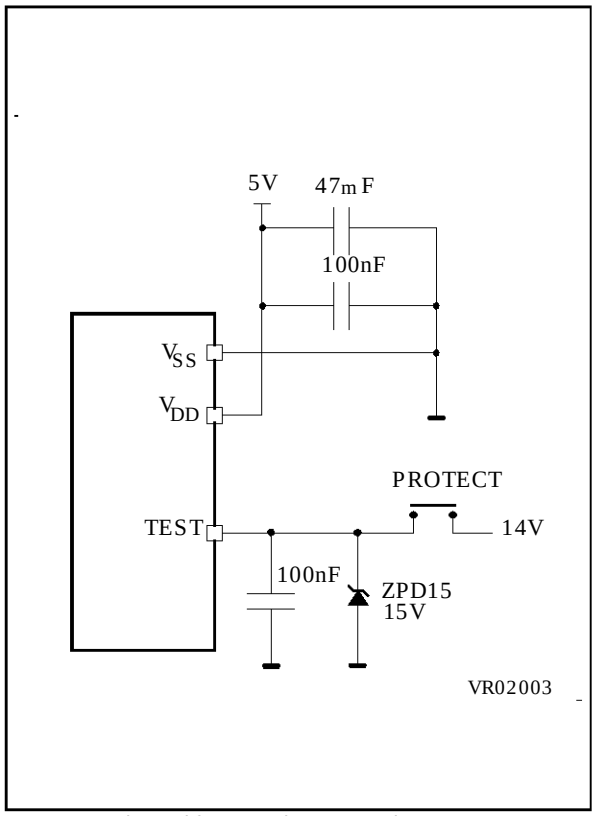


Figure 2. Programming Circuit



Note: ZPD15 is used for overvoltage protection

ST6260B and ST6263B MICROCONTROLLER OPTION LIST

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference . . . . . .

SGS-THOMSON Microelectronics references

Device: [ ] ST6260B and ST6263B

Package: [ ] Dual in Line Plastic [ ] Small Outline Plastic

In this case, select conditioning

[ ] Standard (Stick)

[ ] Tape & Reel

Temperature Range: [ ] 0°C to + 70°C [ ] - 40°C to + 85°C

Special Marking: [ ] No

[ ] Yes "_ _ _ _ _ _ _ _ _ _ "

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Maximum character count: DIP20: 10

SO20: 8

Oscillator Source Selection: [ ] Crystal Quartz/Ceramic resonator (Default)

[ ] RC Network

Watchdog Selection: [ ] Software Activation (STOP mode available)

[ ] Hardware Activation (no STOP mode)

Power on Reset Delay

[ ] 32768 cycle delay

[ ] 2048 cycle delay

ROM Readout Protection: [ ] Standard (Fuse cannot be blown)

[ ] Enabled (Fuse can be blown by the customer)

Note: No part is delivered with protected ROM.
The fuse must be blown for protection to be effective.

External STOP Mode Control

[ ] Enabled

[ ] Disabled (Default)

Comments :

Supply Operating Range in the application:

Oscillator Fequency in the application:

Notes . . . . . .

Signature . . . . . .

Date . . . . . .

1.3 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to SGS-THOMSON.

1.3.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected mask options are communicated to SGS-THOMSON using the correctly filled OPTION LIST appended.

1.3.2 Listing Generation and Verification

When SGS-THOMSON receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the mask which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing forms a part of the contractual agreement for the creation of the specific customer mask.

The SGS-THOMSON Sales Organization will be pleased to provide detailed information on contractual points.

Table 1. ROM Memory Map for ST6260B

Device Address	Description
0000h-007Fh	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 2. ROM Memory Map for ST6263B

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 3. ROM version Ordering Information

Sales Type	ROM	I/O	Additional Features	Temperature Range	Package
ST6260BB1/XXX ST6260BB6/XXX	3884 Bytes	13	A/D CONVERTER SPI	0 to +70°C -40 to + 85°C	PDIP20
ST6260BM1/XXX ST6260BM6/XXX				0 to +70°C -40 to + 85°C	PSO20
ST6263BB1/XXX ST6263BB6/XXX	1836 Bytes		A/D CONVERTER	0 to +70°C -40 to + 85°C	PDIP20
ST6263BM1/XXX ST6263BM6/XXX				0 to +70°C -40 to + 85°C	PSO20

SGS-THOMSON MICROELECTRONICS

ST622x-KIT

STARTER KIT
FOR ST620x, ST621x and ST622x MCUs

HARDWARE FEATURES

- Immediate evaluation of all ST620x, ST621x and ST622x devices, with stand-alone demonstration routines.
- Simulation and debugging within the user's real application environment.
- In-socket programming of all DIL OTP and EPROM ST620x, ST621x and ST622x devices.
- In-circuit programming of all DIL and SO OTP and EPROM ST620x, ST621x and ST622x devices directly on the user's application board.

SOFTWARE FEATURES

- Software simulation, including I/O read/write.
- Assembler, Linker and Debugger.
- In-socket OTP and EPROM programming utilities.
- In-circuit OTP and EPROM programming utilities
- Application examples and demonstrations

1 DESCRIPTION

The ST622x Starter Kit can be used for evaluation, simulation and emulation purposes. First, it can be used to demonstrate the capabilities of the ST6225. It is only necessary to connect the supply to the board and to load the demonstration software provided with the Kit into the ST62E25 sample.

The same board can be used as a hardware interface to the software simulator when connected to the PC. Analog or digital values from the ST622x I/O pins can also be loaded directly to the simulator.

Once the program is successfully simulated, it can be loaded in a ST62E25 or ST62E20 by using the on-board programmer (DIL packaged devices only). The application environment can be connected to the Starter Kit via the I/O connector to perform a full evaluation of the user application.

In addition, an in-circuit programming facility is provided with the Kit to enable programming, via the Starter Kit board, of any ST62E25 (EPROM) or ST62T2x (OTP) already mounted in the user application board.

1.1 HARDWARE ITEMS

The Kit includes 2 samples of ST62E25, 2 samples of ST62E20, an RS232 interface, a temperature control circuit, a trimmer, a set of LED and buttons, all cables plus a power supply.

Pins are available for direct connection to an external user application.

The board is connected to the PC via the parallel port.

1.2 SOFTWARE ITEMS

The diskette provided with this kit includes an enhanced simulator including I/O read/write, assembler, linker, EPROM/OTP ST6 programming facilities and demonstration examples.

1.3 DOCUMENTATION

A full set of documents is provided with the Kit including the ST622x data book, a Kit guide and the ST62/63 Software Development Tools user manual.

1.4 SYSTEM REQUIREMENT

The ST622x Starter Kit communicates with a PC-AT compatible Personal Computer equipped with a hard disk and a 3 1/2" diskette drive, 640k of conventional memory, one parallel Centronic compatible port and MS-DOS version 3.10 or higher.

1.5 BULLETIN BOARD SYSTEMS

Software Tools upgrades, sample code and documentation are available to registered users on the SGS-THOMSON Bulletin Board Systems (BBS). These are accessible by Modem at the following numbers:

– *In the USA*:
(1) 847 517-1898
2400-14.4k baud (V22bis),
8-bits, No Parity, 1 Stop bit (8,N,1)

– *In Europe*:
Micros Technical Support Hotline (France):
(+33) 76 04 93 99
9600 baud (V32) and lower, 8,N,1

Note: This product conforms with the 89/336/EEC directive; it also complies with the EN55022 emissions standard for ITE, as well as with generic 50082-1 immunity standards.

The product is a **Class A apparatus**. In a residential environment this device may cause radioelectrical disturbances which may require that the user adopt appropriate precautions.

The product is not contained in an outer casing, and cannot therefore be immune against electrostatic discharge (ESD): **it should therefore only be handled at static safe work stations.**

Figure 1. Block Diagram of ST622x Starter Kit

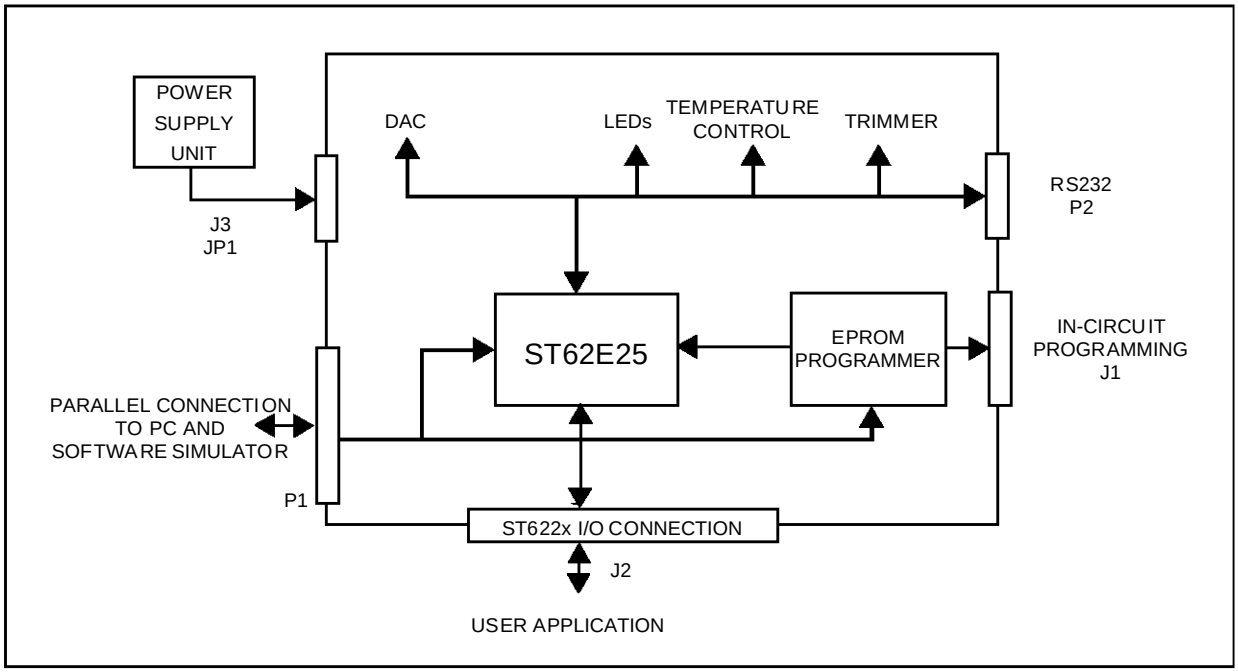


Table 1. kit Contents

ST622x STARTER KIT Hardware:	– ST6225 MCU Core and Peripherals evaluation – ST620x, ST621x and ST622x EPROM/OTP programmer – DIL support for ST620x, ST621x and ST622x programming, in 16, 20 and 28 lead packages – Power supply and PC-AT connection cable
SOFTWARE TOOLS:	– AST6/LST6 ST6 family assembler/linker – SIMST6 simulator software – ST622xPG EPROM/OTP programming software
APPLICATION ROUTINES:	– Demonstration programs – Basic subroutines library
ASSOCIATED DOCUMENTS:	– STARTER KIT USER MANUAL – ST6 SOFTWARE TOOLS MANUAL (DBST6SOFTOST/x)* – ST62 GENERAL PURPOSE APPLICATION MANUAL (AMST62APPLST/x)* – ST62 GENERAL PURPOSE DATA BOOK (DBST6ST/5)

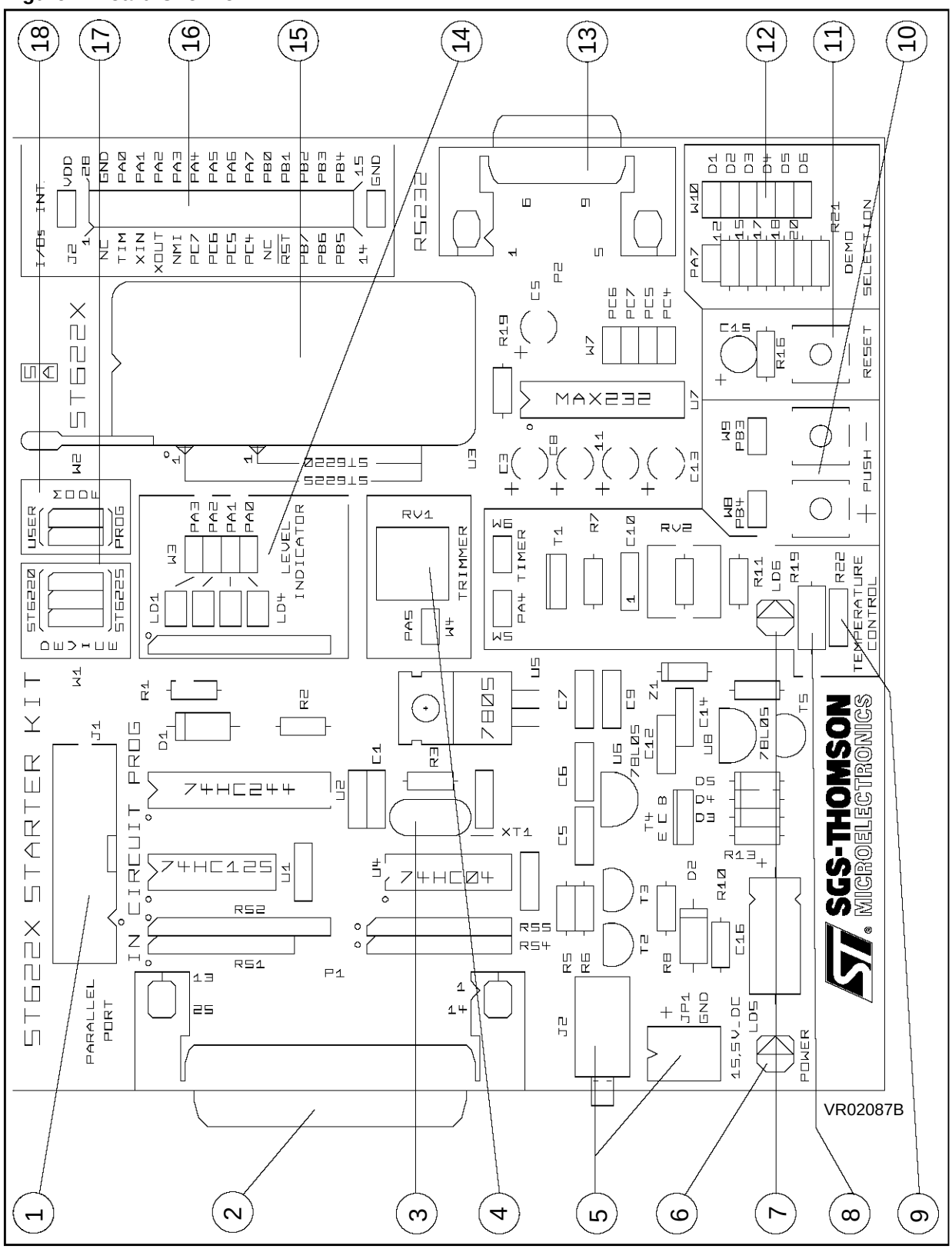
*Contact Sales for the latest version.

2 HARDWARE DESCRIPTION

2.1 BOARD OVERVIEW

1- IN CIRCUIT programming connector J1
2- PC station connector P1 (for links to simulator and programming softwares)
3- 8 Mhz crystal oscillator
4- 10 Kohms trimer including jumper W4-PA5
5- Power supply JACK connector J3 and JP1 pads
6- Power supply LED indicator LD1
7- Heater resistor power LED indicator LD2
8- Heater resistor circuit including jumper W6-TIMER
9- Thermistor circuit including jumper W5-PA4
10- "+" and "-" pushbuttons including jumpers W8-PB4 and W9-PB3
11- RESET pushbutton
12- Demonstration routine selector including jumpers W10
13- RS232 interface circuit and connector including jumpers W7
14- 4 LEDs Level indicator including jumpers W3
15- DIL 20-28 MCU socket
16- User's I/O interface connector J2
17- "ST6220" or "ST6225" device selection jumpers W1
18- "Programming" or "User" operating mode selection jumpers W2
19-

Figure 2. Board Overview



ORDERING INFORMATION

Starter Kit	Device	Description
ST622x-KIT/220	ST620x ST621x ST622x	Complete Kit for operation from 220 Vac mains
ST622x-KIT/110	ST620x ST621x ST622x	Complete Kit for operation from 110 Vac mains
ST622x-KIT/UK	ST620x ST621x ST622x	Complete Kit for operation in United Kingdom

SGS-THOMSON MICROELECTRONICS

ACTUM REALIZER® FOR THE ST62

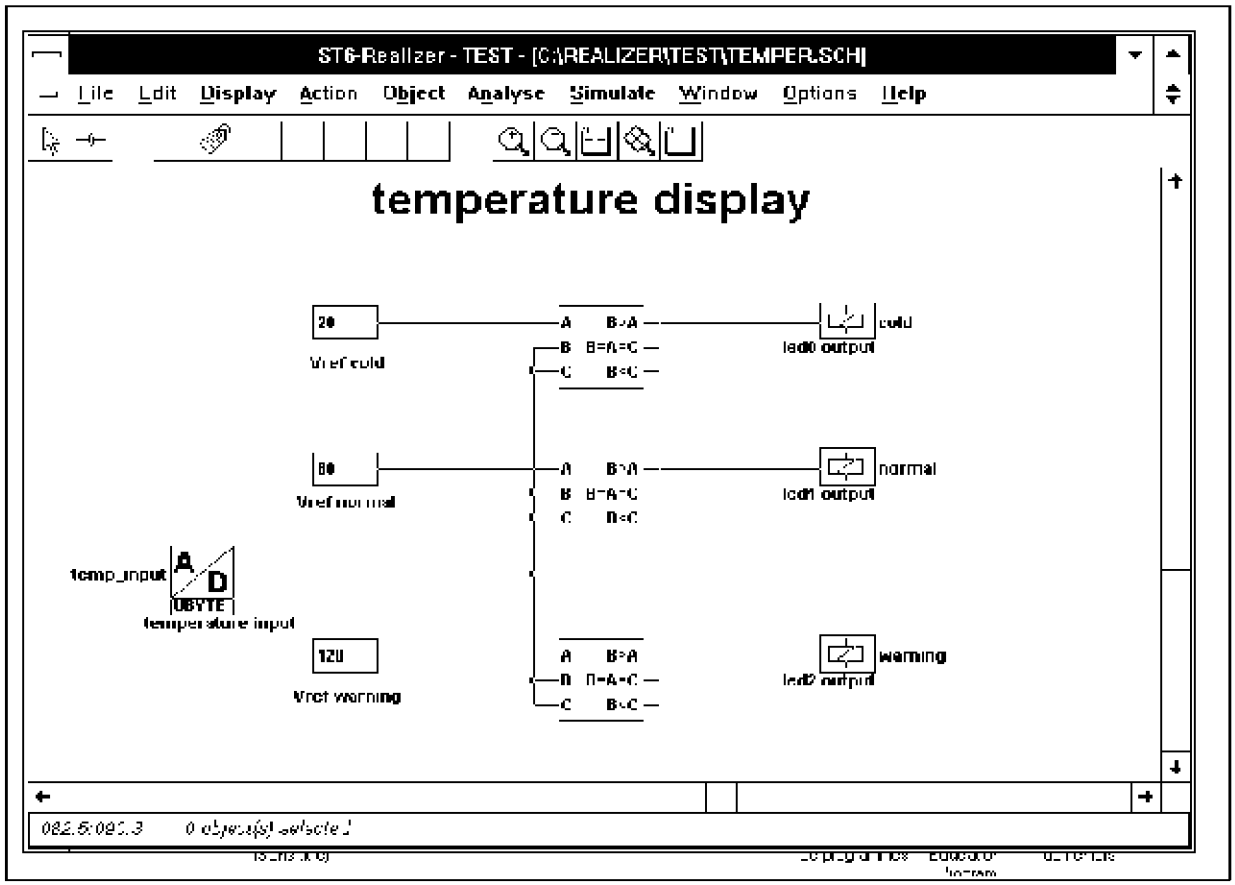
SOFTWARE AIDED COMPUTER ENGINEERING FOR ST62 8-BIT MICROCONTROLLER

PRELIMINARY DATA

GRAPHIC DESIGN AND DEBUG

- SCHEMATIC-BASED SOFTWARE DESIGN
  - Industry Standard Graphic symbols
  - Extensive Symbol Library
  - Select and Wire on-screen to generate Application
  - Built-In Self-Documentation
- SCHEMATIC-BASED ANALYSIS
  - Single click Analyze Operation
  - Path and Functional Verification
  - Efficient code generation for ST62
- SCHEMATIC-BASED SIMULATOR FOR DEBUG
  - Runs on design schematic
  - Stimulate and Observe On-line
  - Add Virtual tools (Oscilloscope, Pulse, Time, Data Generators, logic probes)
  - Uses ST62 Code from Analyzer
  - Dedicated to ST62 Microcontroller

Figure 1. Schematic Entry of a simple Application



DESCRIPTION

The ST62-Realizer is a version of The Realizer® by Actum Solutions [1], The Netherlands, dedicated to design and development of applications for the ST62 family of microcontrollers (MCUs).

The schematic-based architecture for both design and Simulation allows all engineers, even without Microcontroller experience, to create a Microcontroller application, with all the benefits and flexibility that microcontrollers offer.

DESIGN

Use your knowledge of the application to decide Input/Output functions, then draw and build up the application functionality graphically using the industry standard symbols from the library (or create your own). A state machine can be added if required. Draw wires between the symbols on-screen to create the application. The schematic can be printed for documentation.

ANALYSE

Select an ST62 device and allocate I/O functions to actual “physical” I/O pins. Run the Analyzer to verify and run process analysis on the application as drawn. If errors are found, edit the schematic and re-analyze. When all is satisfactory, allow the Analyzer to generate the efficient ST62 software code.

DEBUG

Use the integrated ST62 Symbolic Schematic Simulator to stimulate and monitor the circuit functionality directly on the schematic. Add Virtual development tools to the circuit as graphical symbols and run or step through the functionality to verify that the application works as expected. If not, return to the Schematic entry environment and revise the drawing, then Analyse and Debug again until the function is as desired.

Now you can easily program EPROM or OTP ST62 microcontrollers with the final generated software to build prototypes. Field trials can then be performed with confidence that the application program functionality has been assured, giving less time needed for expensive field modifications.

EXPANSION

The Realizer can produce a complete standalone application program, or just the main part to startup and control the program functions designed on-screen. In this latter case, the code can be linked with other library routines (for example Fuzzy Logic routines) to expand on the functionality. The expanded code is then able to be debugged with the ST62 Hardware Emulation Tools.

HARDWARE/SOFTWARE REQUIREMENTS

- A 80386 (or higher) PC with at least 2Mbytes of memory
- MS-Windows 3.0 or higher and MS-DOS 3.3 or higher
- Hard disk with 5MB of free disk space and a 3.5” floppy disk drive
- VGA monitor supported by Windows and a pointing device (e.g. a mouse)

Note [1]: Actum Solutions:
PO Box 373
1700 HJ HEERHUGOWAARD
The Netherlands

Figure 2. Adding functional Blocks as Symbols

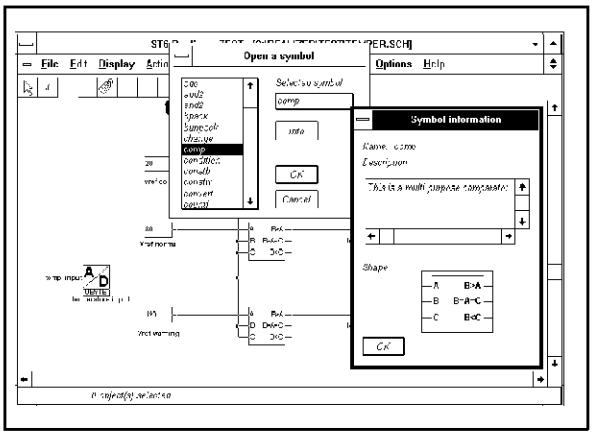
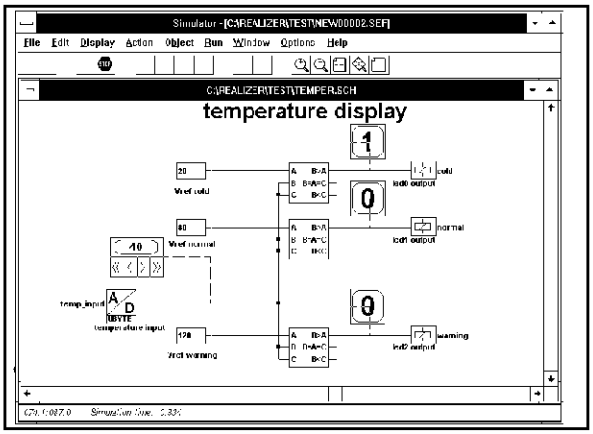


Figure 3. Testing with virtual Tools



Extract of available Symbols and Functions

adc	Analog to digital converter
add2	Two input adder with type inheritance
and2	Two input bitwise AND function, with type inheritance
bpack	Eight bits to one byte packer
bunpack	One byte to eight bits unpacker
change	Change detector.
comp	Multi purpose comparator
condition	Condition function for the state machines
constb	Constant bit symbol
constw	Constant word symbol
convert	Type conversion symbol
countf	Counter with a fixed preset value
countv	Counter with a variable preset value
dac	Digital to analog convertor
delf	Delay with a fixed on and off delay time
delfoff	Delay with a fixed off delay time
delfon	Delay with a fixed on delay time
delv	Delay with a variable on and off delay time
delvoff	Delay with a variable off delay time
delvon	Delay with a variable on delay time
dff	D-flipflop with a multiple type data input
digin	Digital input
digout	Digital output
div	2 input divider with quotient and remainder output, with multi type inheritance
edge	Rising edge detector
indextable	Index table, the input is used as the index in a table
init	Output only TRUE during the first loop after a reset
inv	Multi type bitwise inverter
limf	Fixed limiter, the output will not be larger than the top value and not smaller than the bottom value
limv	Variable limiter, the output will not be larger than the top value and not smaller than the bottom value
lookuptable	Look up table, input value is used to search through a table to find the output value
loopdel	Output always the previous value of the current input (delays one loop)
mul	2 input multipier, multi type inheritance
mux1	Two input multiplexer. If the selection input is FALSE input 0 is copied to the output, otherwise input 1 is copied
mux2	Four input multiplexer
or2	Two input OR function, multi type inheritance
oscf	Fixed time oscillator, frequency equals: f(Hz) = 1/(2*time)
oscv	Variable time oscillator, frequency equals: f(hz) = 1/(2*time)
portin	Connect sub scheme symbol pins from the parent scheme with the sub scheme nets
portout	Connect sub scheme symbol pins from the parent scheme to the sub scheme nets
shift	Shift register symbol with parallel in, serial in, shift up, shift down
srff	Set/reset flip flop
sss	Example of a sub scheme symbol
state	State symbol, used within a state machine
statein	State input symbol to connect to a state machine
stateinit	Initial state of a state machine
stateout	State output symbol for extracting a state from a state machine
sub2	Two input subtractor, with type inheritance
timf	Fixed timer, which will generate a pulse on a rising edge on the input
timv	Variable timer, which will generate a pulse on a rising edge on the input
title	Title block is used for archiving purposes
wmerge	Build a word out of a high byte and a low byte
wpack	Pack sixteen bits into one word
wsplit	Split a word into a high byte and a low byte
wunpack	Unpack a word into sixteen bits
xor	Two input bitwise exor function, multi type inheritance

ORDERING INFORMATION

Sales Type	Description
ST6-REALIZER/PC	Software Aided Computer Engineering for ST62 8-bit Microcontroller, Microsoft Windows 3™ Edition