Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST6
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI
Peripherals	LED, LVD, POR, WDT
Number of I/O	21
Program Memory Size	3.8KB (3.8K x 8)
Program Memory Type	OTP
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	A/D 13x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t65cm6

Table of Contents

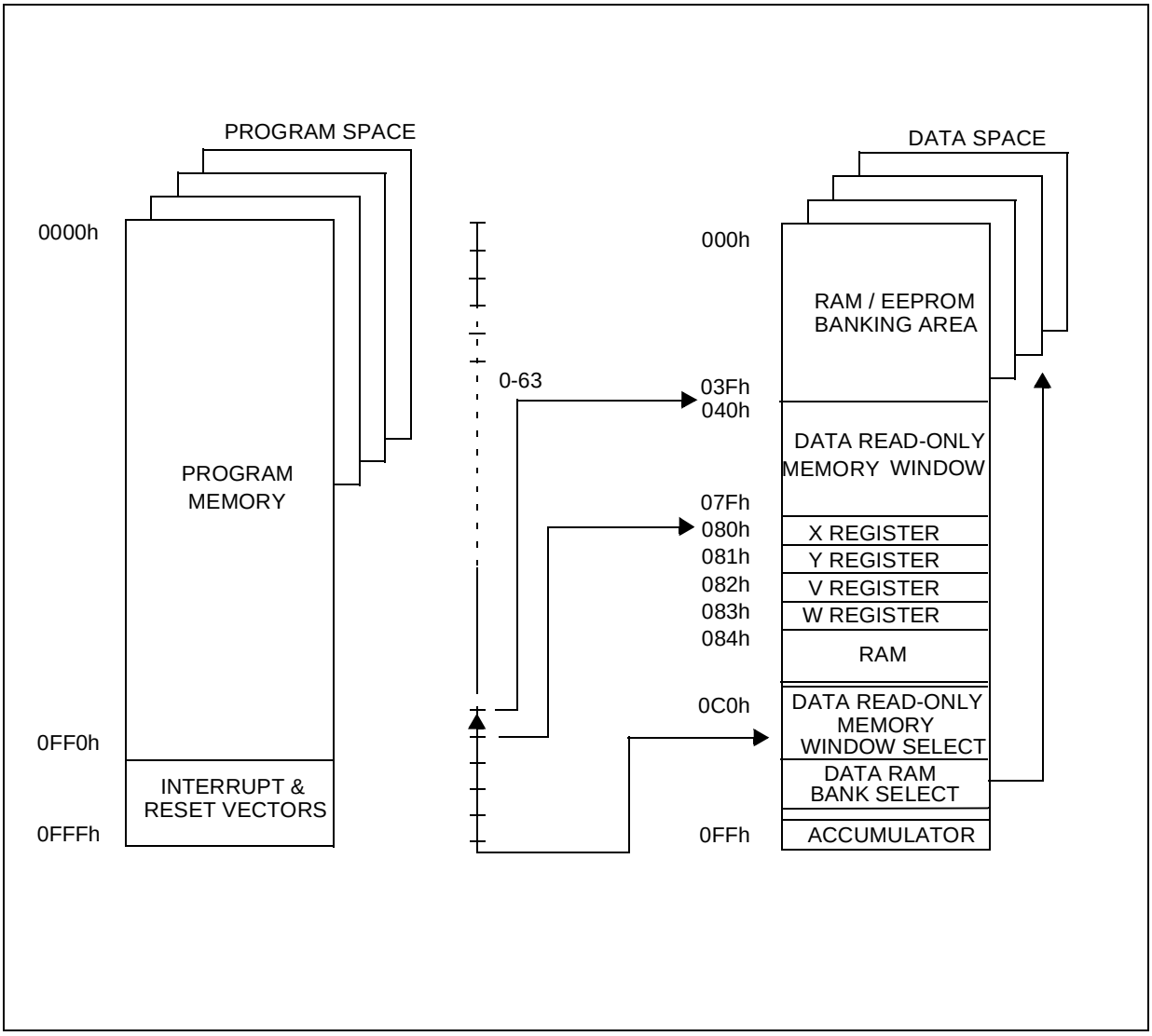
1.3 MEMORY MAP

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 3Memory 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).

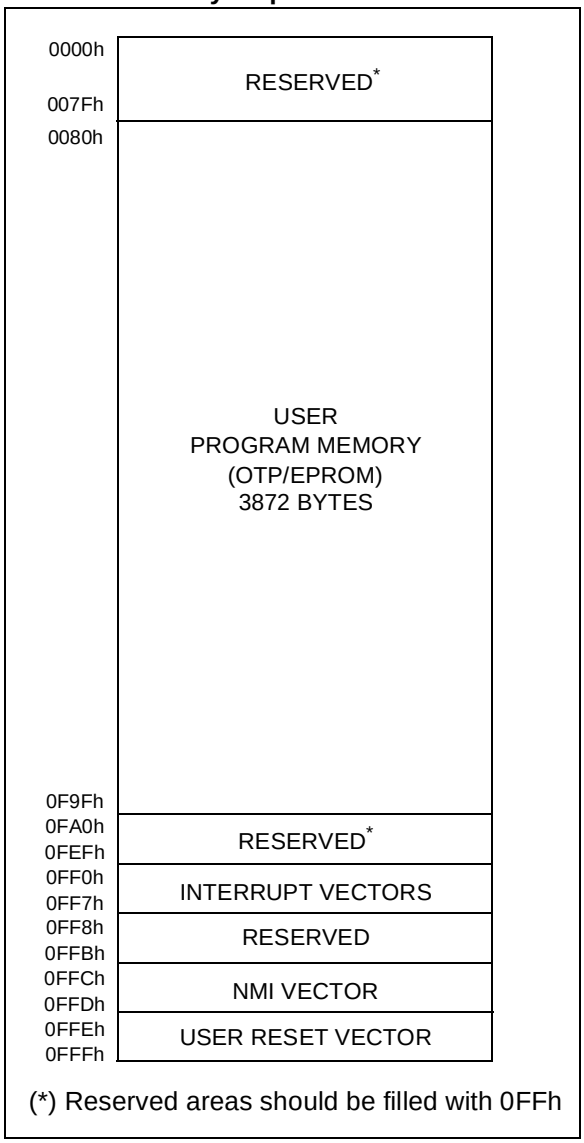
1.3.2.1 Program Memory Protection

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 STMicroelectronics, to gain access to the OTP contents. Returned parts with a protection set can therefore not be accepted.

Figure 4.ST62T55C/T65C/E65C 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 ST62T55C, ST62T65C and ST62E65C 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
ST62T55C	1 x 64 bytes	-
ST62T65C/E65C	1 x 64 bytes	2 x 64 bytes

Table 2. ST62T55C, ST62T65C and ST62E65C 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	0E0h
SPI DIVIDER REGISTER	0E1h
SPI MODE REGISTER	0E2h
RESERVED	0E3h 0E7h
DATA RAM/EEPROM REGISTER	0E8h*
RESERVED	0E9h
EEPROM CONTROL REGISTER	0EAh
RESERVED	0EBh 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

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 9, Figure 10, Figure 11 and Figure 12.

A programmable divider on F_{INT} is also provided in order to adjust the internal clock of the MCU to the best power consumption and performance trade-off.

Figure 8 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 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 11.

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

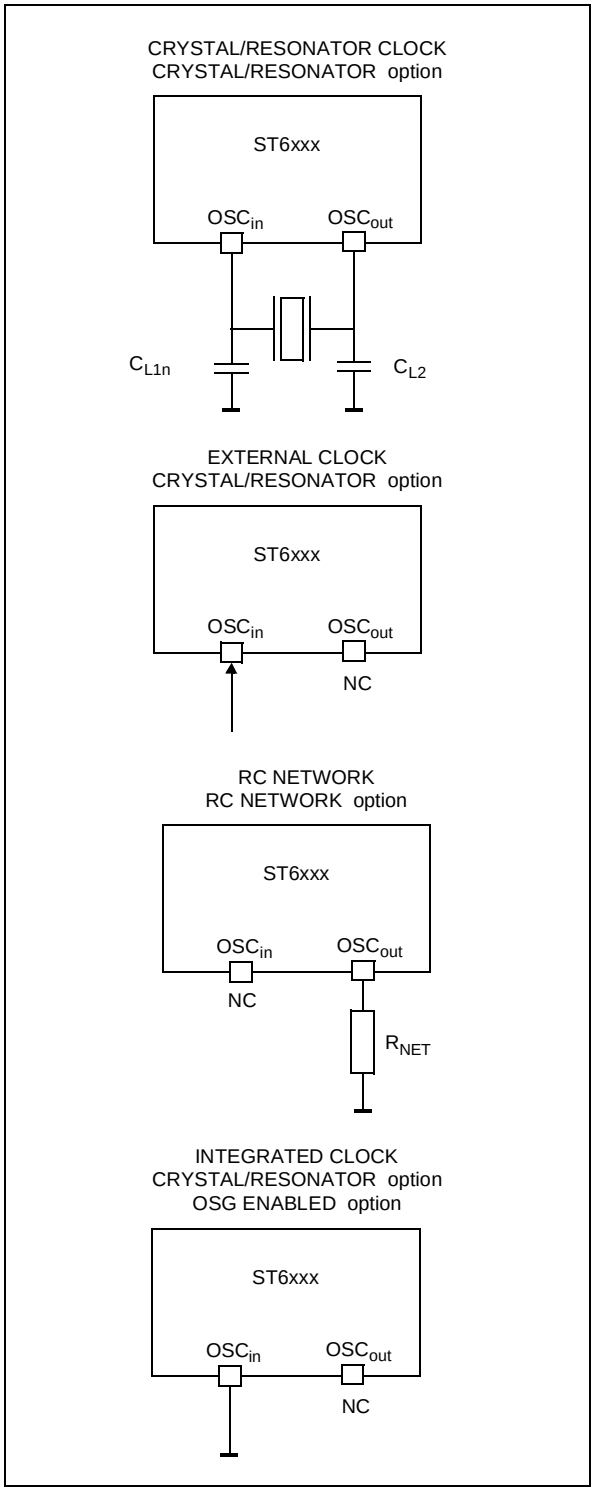
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

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.

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

Figure 8. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Figure 9. OSG Filtering Principle

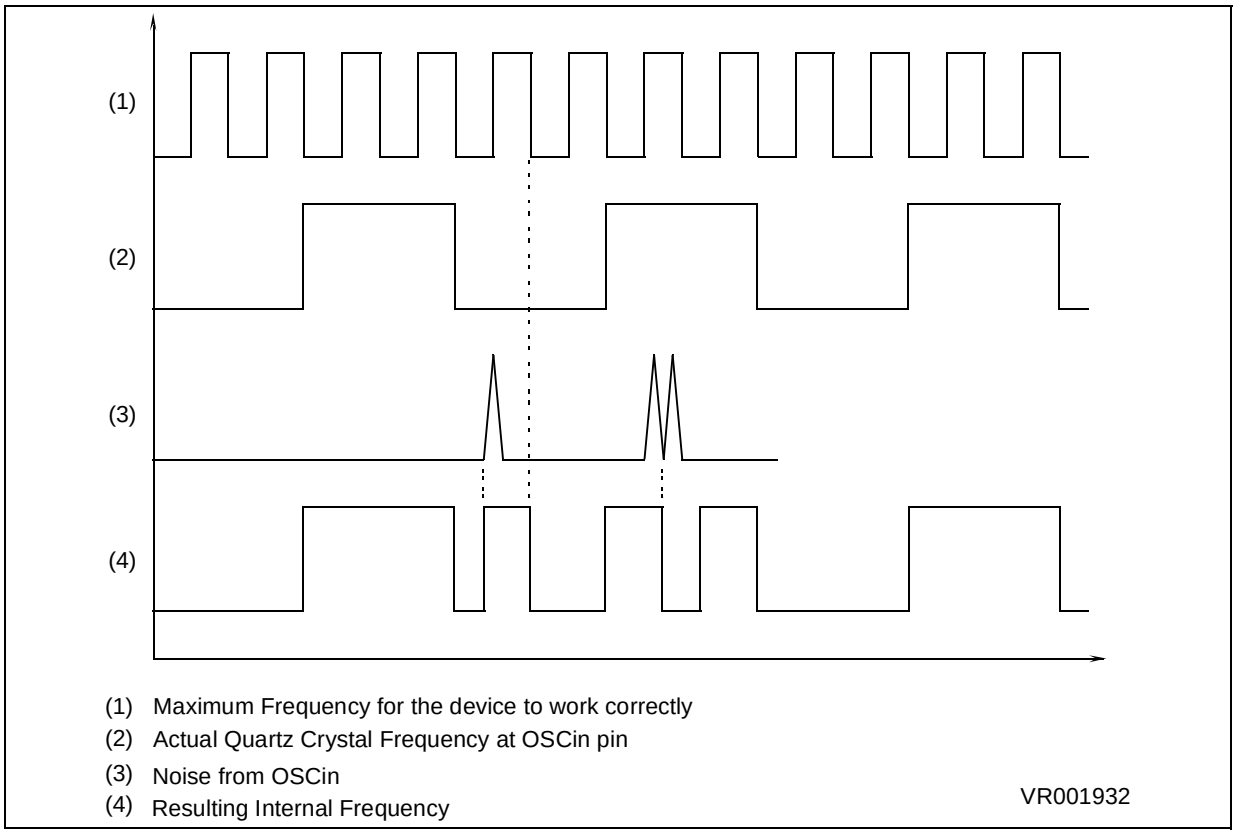
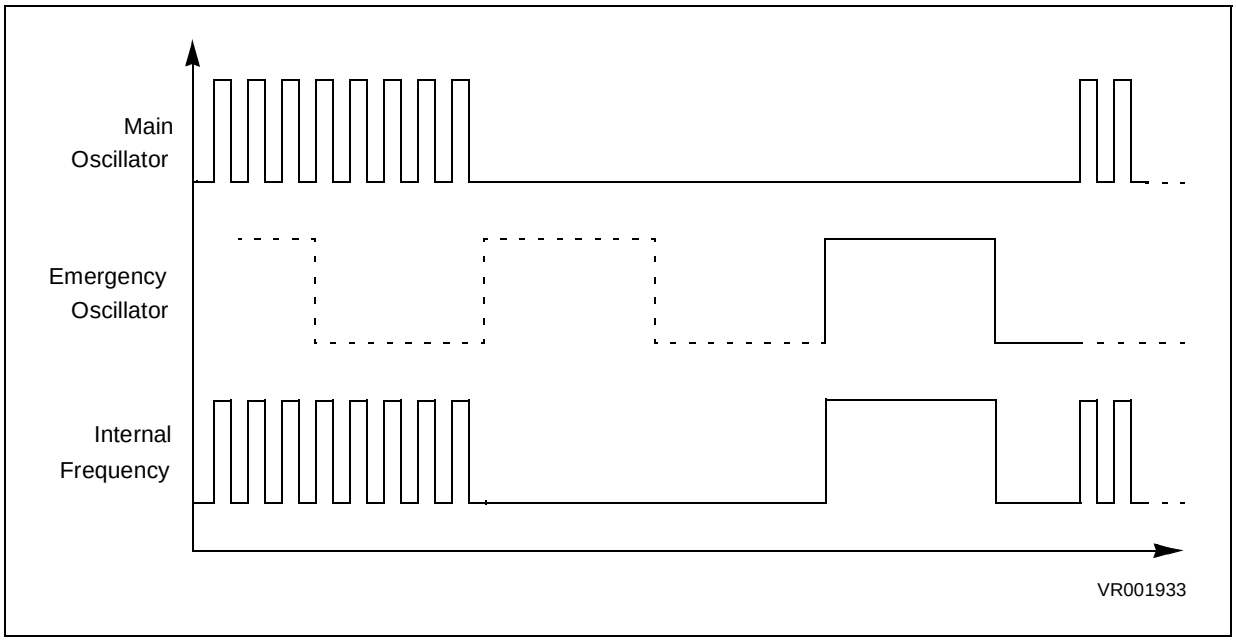


Figure 10. OSG Emergency Oscillator Principle



RESETS (Cont'd)

Table 5Register Reset Status

Register	Address(es)	Status	Comment
Oscillator Control Register	0DCh	00h	
EEPROM Control Register	0EAh	00h	EEPROM disabled (if available)
Port Data Registers	0C0h to 0C2h	00h	I/O are Input with pull-up
Port Direction Register	0C4h to 0C6h	00h	I/O are Input with pull-up
Port Option Register	0CCh to 0CEh	00h	I/O are Input with pull-up
Interrupt Option Register	0C8h	00h	Interrupt disabled
TIMER Status/Control	0D4h	00h	TIMER disabled
AR TIMER Mode Control Register	0D5h	00h	AR TIMER stopped
AR TIMER Status/Control 0 Register	0D6h	02h	
AR TIMER Status/Control 1 Register	0D7h	00h	
AR TIMER Compare Register	0DAh	00h	
AR TIMER Load Register	0DBh	00h	
Miscellaneous Register	0DDh	00h	SPI Output not connected to PC3
SPI Registers	0E0h to 0E2h	00h	SPI disabled
SPI DIV Register	0E1h	00h	SPI disabled
SPI MOD Register	0E2h	00h	SPI disabled
SPI DSR Register	0E0h	Undefined	SPI disabled
X, Y, V, W, Register	080H TO 083H		
Accumulator	0FFh		
Data RAM	084h to 0BFh		
Data RAM EEPROM 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

INTERRUPTS (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 source #1 is level sensitive. When cleared to zero the edge sensitive mode for interrupt request is selected.

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

The bit ESB selects the polarity of the interrupt source #2.

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.

3.4.4 Interrupt sources

Interrupt sources available on these MCUs are summarized in the Table 9 with associated mask bit to enable/disable the interrupt request.

Table 9Interrupt 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

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.

I/O PORTS (Cont'd)

4.1.2 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 2. 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 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:

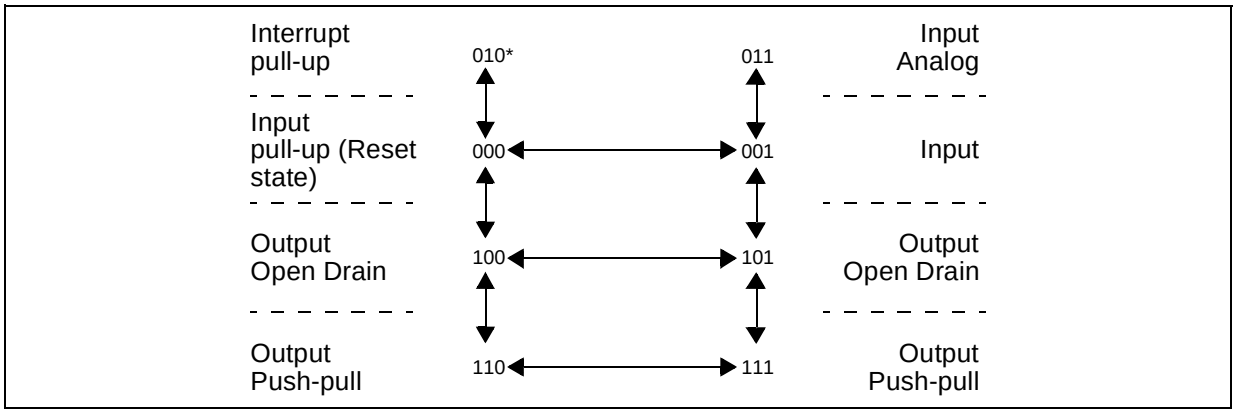
```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

Warning: Care must also be taken to not use instructions that act on a whole port register (INC, DEC, or read operations) when all 8 bits are not available on the device. Unavailable bits must be masked by software (AND instruction).

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 23. Diagram showing Safe I/O State Transitions



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

I/O PORTS (Cont'd)

4.1.3 Timer 1 Alternate function Option

When bit TOUT of register TSCR1 is low, pin PC1/Timer 1 is configured through the port registers as any standard pin of Port B. It is in addition connected to the Timer 1 input for Gated and Event counter modes. When bit TOUT of register TSCR1 is high, pin PC1/Timer 1 is forced as Timer 1 output, independently of the port registers configuration.

4.1.4 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.

4.1.5 SPI Alternate function Option

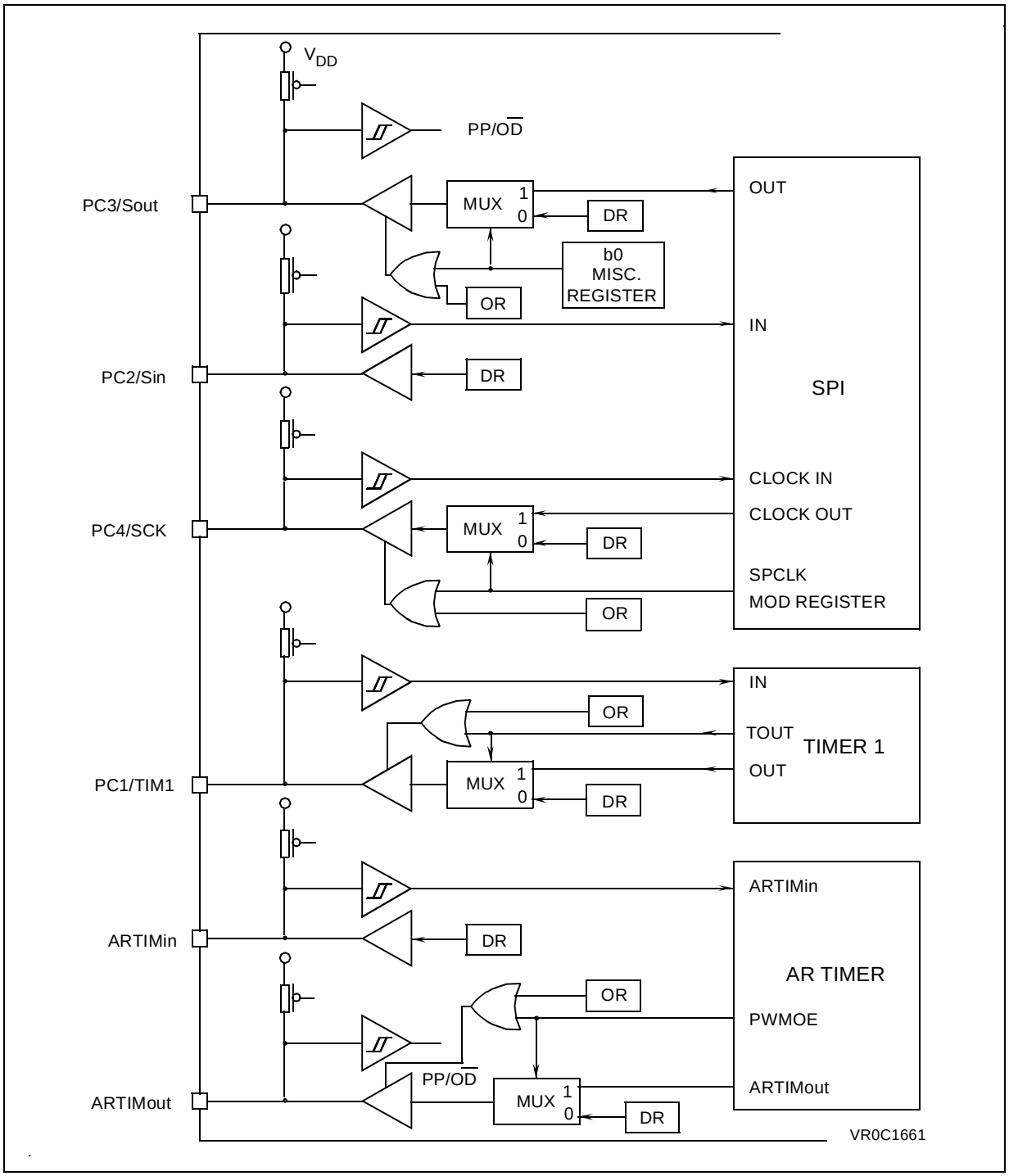
PC2/PC4 are used as standard I/O as long as bit SPCLK of the SPI Mode Register is kept low. When PC2/Sin is configured as input, it is automatically connected to the SPI shift register input, independent of the state at SPCLK.

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.

I/O PORTS (Cont'd)

Figure 24Peripheral 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}. The peripheral may be configured in three different operating modes.

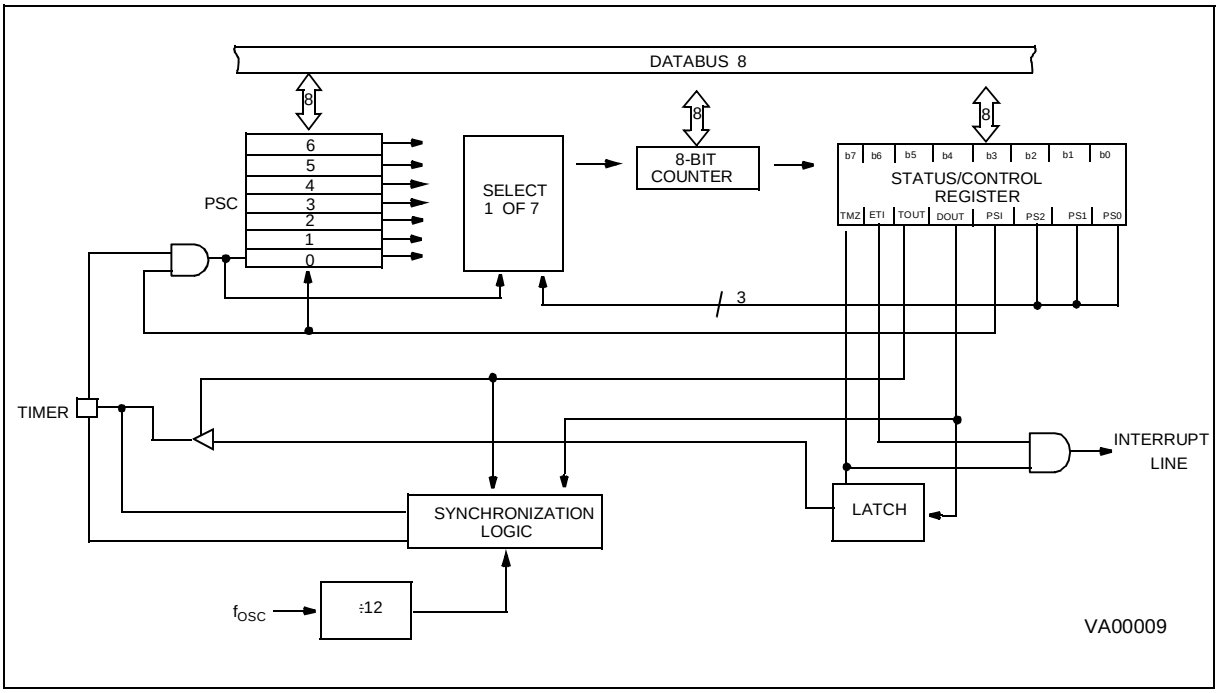
Figure 25 shows the Timer Block Diagram. The external TIMER pin is available to the user. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, while the state of the 7-bit prescaler can be read in the PSC register. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decremented 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 is generated as described in the Interrupt Chapter. 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. 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 26 illustrates the Timer's working principle.

Figure 25. Timer Block Diagram



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.

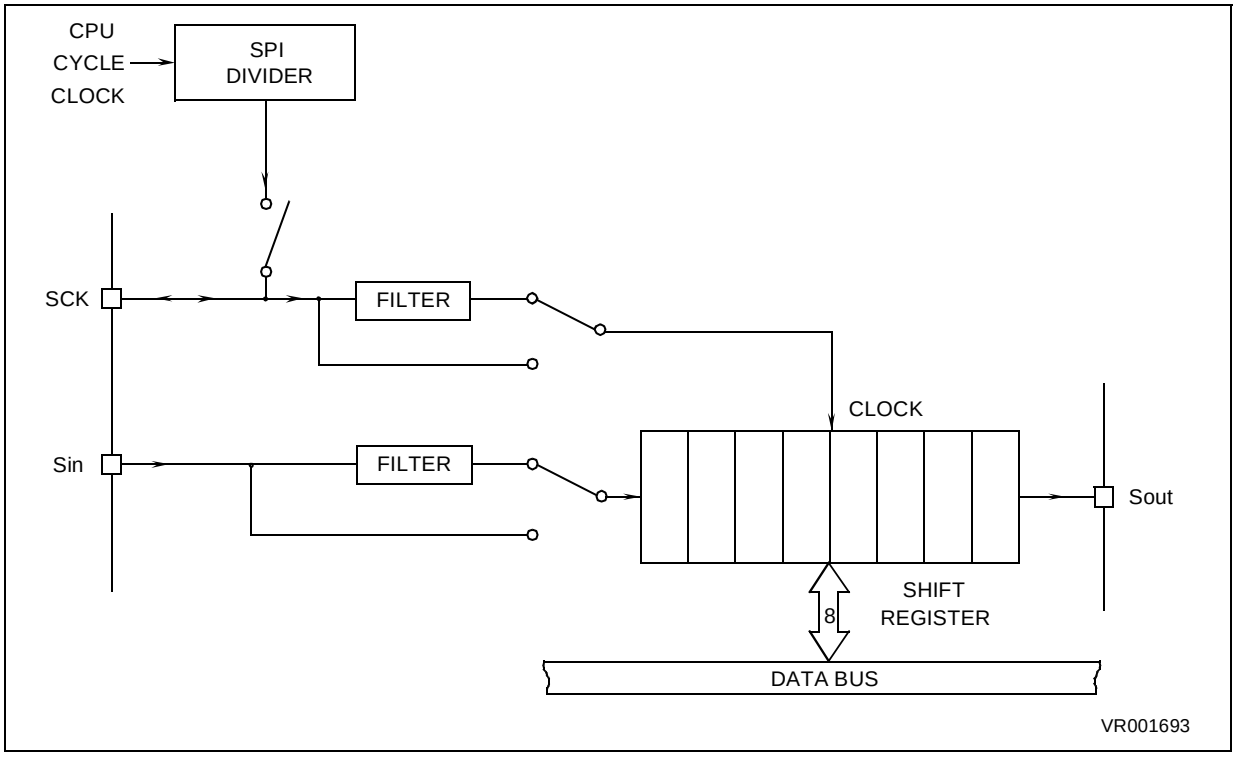
For maximum versatility the SPI may be programmed to sample data either on the rising or on the falling edge of SCK, with or without phase shift (clock Polarity and Phase selection).

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 30. SPI Block Diagram



SERIAL PERIPHERAL INTERFACE SPI (Cont'd)

Figure 33. CPOL = 0 Clock Polarity Normal, CPHA = 1 Phase Selection Shifted

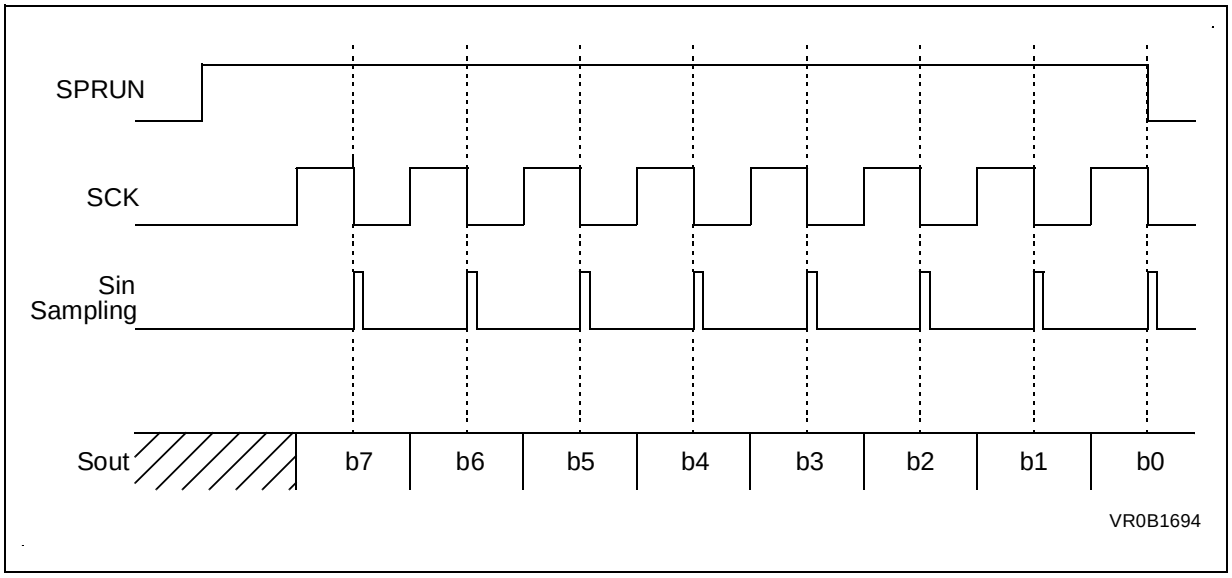
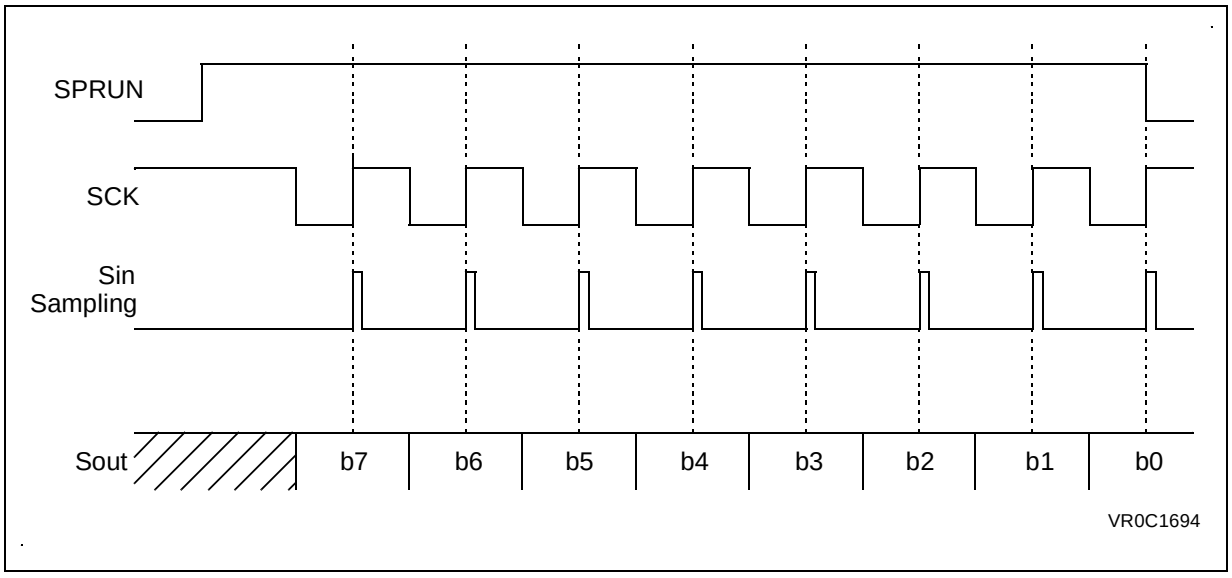


Figure 34. CPOL = 1 Clock Polarity Inverted, CPHA = 1 Phase Selection Shifted



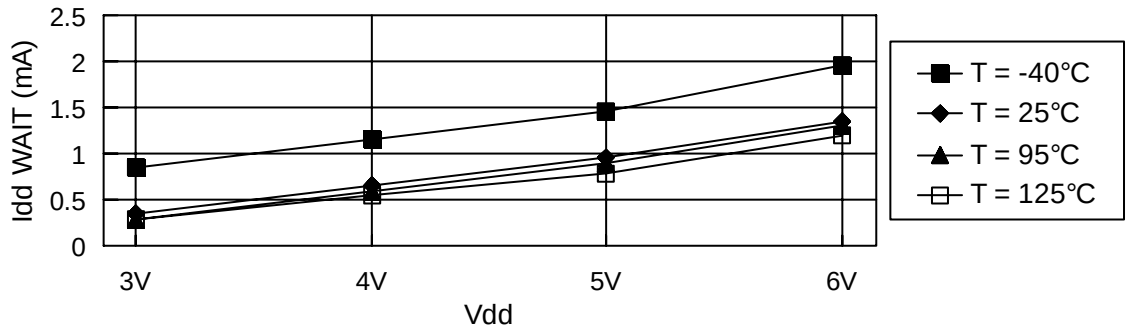
6.3 DC ELECTRICAL CHARACTERISTICS

(T_A = -40 to +125°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage All Input pins				V_{DD} x 0.3	V
V_{IH}	Input High Level Voltage All Input pins		V_{DD} x 0.7			V
V_{Hys}	Hysteresis Voltage (1) All Input pins	V_{DD}= 5V V_{DD}= 3V	0.2 0.2			V
V_{up}	LVD Threshold in power-on			4.1	4.3	
V_{dn}	LVD threshold in powerdown		3.5	3.8		
V_{OL}	Low Level Output Voltage All Output pins	V_{DD}= 5.0V; I_{OL} = +10µA V_{DD}= 5.0V; I_{OL} = + 3mA			0.1 0.8	V
	Low Level Output Voltage 30 mA Sink I/O pins	V_{DD}= 5.0V; I_{OL} = +10µA V_{DD}= 5.0V; I_{OL} = +7mA V_{DD}= 5.0V; I_{OL} = +15mA			0.1 0.8 1.3	
V_{OH}	High Level Output Voltage All Output pins	V_{DD}= 5.0V; I_{OH} = -10µA V_{DD}= 5.0V; I_{OH} = -3.0mA	4.9 3.5			V
R_{PU}	Pull-up Resistance	All Input pins	40	100	350	KΩ
		RESET pin	150	350	900	
I_{IL} I_{IH}	Input Leakage Current All Input pins but RESET	V_{IN} = V_{SS} (No Pull-Up configured) V_{IN} = V_{DD}		0.1	1.0	µA
	Input Leakage Current RESET pin	V_{IN} = V_{SS} V_{IN} = V_{DD}	-8	-16	-30 10	
I_{DD}	Supply Current in RESET Mode	V_{RESET}=V_{SS} f_{OSC}=8MHz			7	mA
	Supply Current in RUN Mode (2)	V_{DD}=5.0V f_{INT}=8MHz			7	mA
	Supply Current in WAIT Mode (3)	V_{DD}=5.0V f_{INT}=8MHz			2.5	mA
	Supply Current in STOP Mode, with LVD disabled(3)	I_{LOAD}=0mA V_{DD}=5.0V			20	µA
	Supply Current in STOP Mode, with LVD enabled(3)	I_{LOAD}=0mA V_{DD}=5.0V			500	
Retention	EPROM Data Retention	T_A = 55°C	10			years

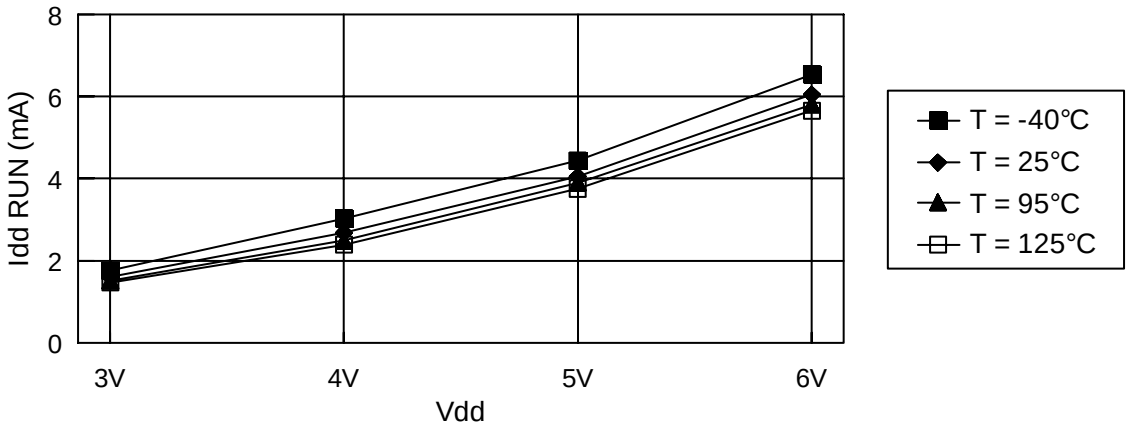
Notes:
(1) Hysteresis voltage between switching levels
(2) All peripherals running
(3) All peripherals in stand-by

Figure 45. Idd WAIT versus V_{DD} at 8Mhz for ROM devices



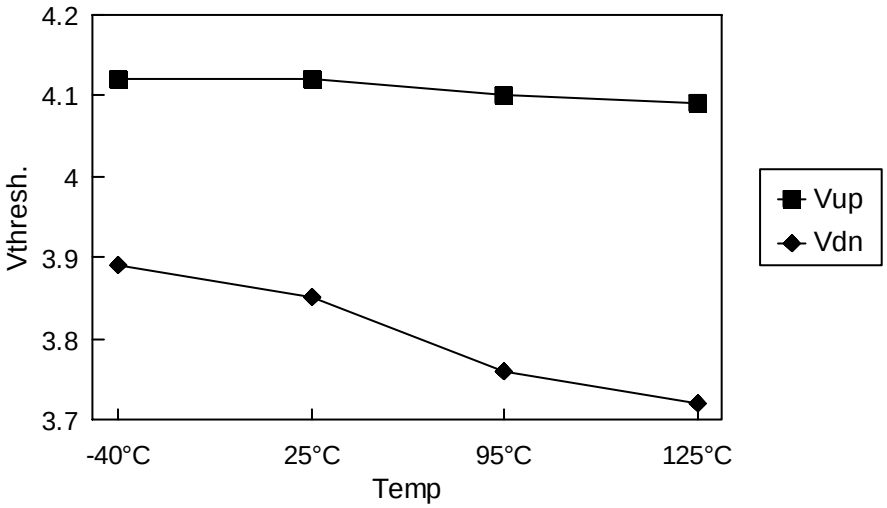
This curves represents typical variations and is given for guidance only

Figure 46. Idd RUN versus V_{DD} at 8 Mhz for ROM and OTP devices



This curves represents typical variations and is given for guidance only

Figure 47. LVD thresholds versus temperature

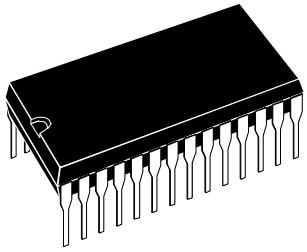


This curves represents typical variations and is given for guidance only

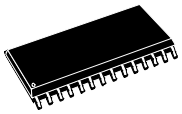
ST62P55C
ST62P65C

8-BIT FASTROM MCUs WITH A/D CONVERTER, SAFE RESET, AUTO-RELOAD TIMER, EEPROM AND SPI

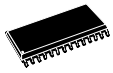
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +125°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- Data RAM: 128 bytes
- Data EEPROM: 128 bytes (none on ST62T55C)
- User Programmable Options
- 21 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
- 8 I/O lines can sink up to 30mA 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
- Oscillator Safe Guard
- Low Voltage Detector for Safe Reset
- 8-bit A/D Converter with 13 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 a parallel port)



PDIP28



PS028



SS0P28

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

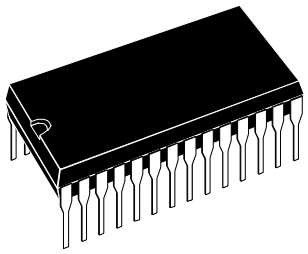
DEVICE	ROM (Bytes)	EEPROM
ST62P55C	3884	-
ST62P65C	3884	128

Rev. 2.9

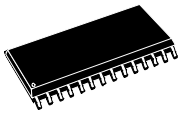
ST6255C
ST6265B

8-BIT ROM MCUs WITH A/D CONVERTER, SAFE RESET, AUTO-RELOAD TIMER, EEPROM AND SPI

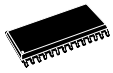
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +125°C Operating Temperature Range
- Run, Wait and Stop Modes
- 5 Interrupt Vectors
- Look-up Table capability in Program Memory
- Data Storage in Program Memory: User selectable size
- Data RAM: 128 bytes
- Data EEPROM: 128 bytes (none on ST62T55C)
- User Programmable Options
- 21 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
- 8 I/O lines can sink up to 30mA 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 13 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 a parallel port)



PDIP28



PS028



SS0P28

(See end of Datasheet for Ordering Information)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	EEPROM	LVD & OSG
ST6255C	3884	-	Yes
ST6265B	3884	128	No

Rev. 2.9

2 SUMMARY OF CHANGES

Rev.	Main Changes	Date
2.9	Modification of “Additional Notes for EEPROM Parallel Mode” (p.13) In section 4.2.4 on page 45: vector #4 instead of vector #3 in description of bit 6 (TSCR register). Changed f_{RC} values in section 6.4 on page 68 Changed Figure 48 on page 74. Changed option list on page 84.	July 2001