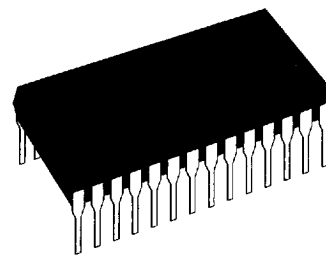
SGS-THOMSON MICROELECTRONICS

ST6260B
ST6265B

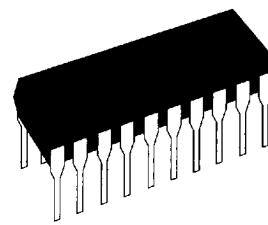
8-BIT HCMOS MCUs WITH A/D CONVERTER, EEPROM & AUTO-RELOAD TIMER

PRELIMINARY DATA

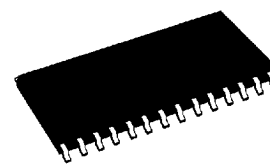
- 3.0 to 6.0V Supply Operating Range
- 8 MHz Maximum Clock Frequency
- -40 to +85°C Operating Temperature Range
- Run, Wait & Stop Modes
- 5 different interrupt vectors
- Look-up table capability in ROM
- User ROM: 3884 bytes
- Data ROM: User selectable size (in program ROM)
- Data RAM: 128 bytes
- EEPROM: 128 bytes
- PDIP20, PSO20 (ST6260B) packages
- PDIP28, PSO28 (ST6265B) packages
- 13/21 fully software programmable I/O as:
  - Input with pull-up resistor
  - Input without pull-up resistor
  - Input with interrupt generation
  - Open-drain or push-pull outputs
  - Analog Inputs
- 6/8 I/O lines can sink up to 20mA for direct LED or TRIAC driving
- 8 bit counter with a 7-bit programmable prescaler (TIMER 1)
- 8 bit Auto-reload Timer with 7-bit programmable prescaler (AR TIMER)
- Digital Watchdog
- 8 bit A/D Converter with up to 7 (ST6260B) and up to 13 (ST6265B) analog inputs
- 8 bit Synchronous Peripheral Interface (SPI)
- On-chip clock oscillator driven by Quartz Crystal, Ceramic resonator or RC network
- User configurable Power-on Reset
- One external not maskable interrupt
- 9 powerful addressing modes
- The development tool of the ST626xB microcontrollers consists of the ST626xB-EMU emulation and development system connected via a standard RS232 serial line to an MS-DOS Personal Computer



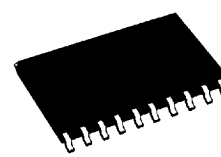
PDIP28



PDIP20



PSO28



PSO20

(Ordering Information at the end of the datasheet)

Figure 1. ST6260B Pin Configuration

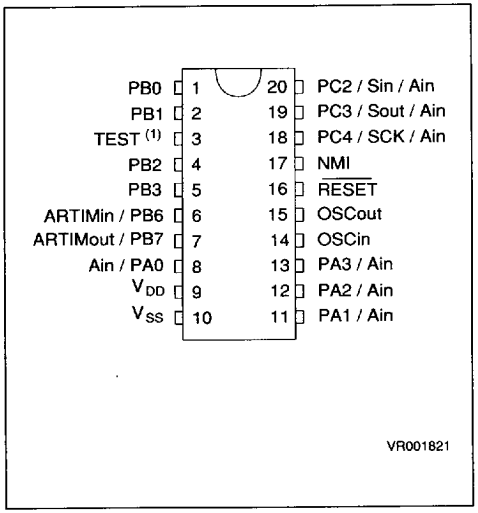
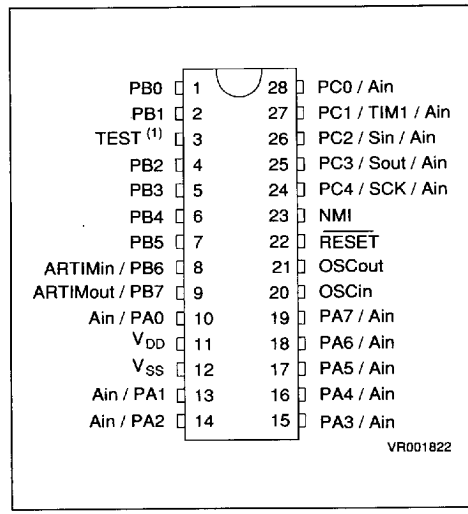
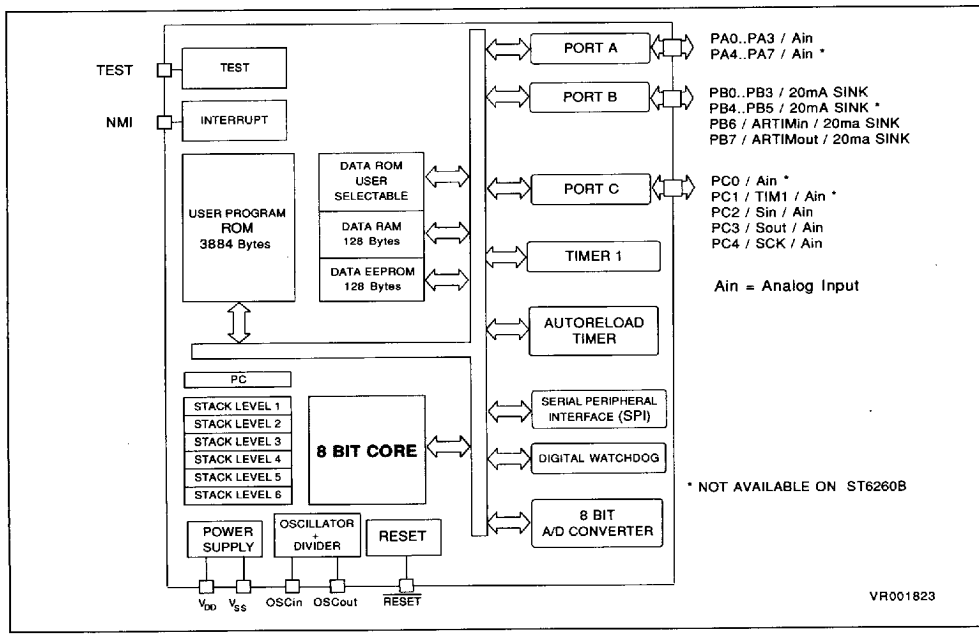


Figure 2. ST6265B Pin Configuration



Note 1: This pin is also the V_{PP} input for EPROM based devices

Figure 3. ST6260B/65B Block Diagram



GENERAL DESCRIPTION

The ST6260B and ST6265B microcontrollers are members of the 8-bit HCMOS ST62xx family, a series of devices oriented to low-medium complexity applications. All ST62xx members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells). The macrocells of the ST6260B and ST6265B are: the Timer peripheral that includes an 8-bit counter with a 7-bit software programmable prescaler (Timer 1), the 8-bit with 7 bit programmable prescaler (AR Timer), the 8-bit A/D Converter with up to 7 (ST6260B) and up to 13 (ST6265B) analog inputs (A/D inputs are alternate functions of I/O pins), the Digital Watchdog (DWD) and an 8-bit Serial synchronous Peripheral Interface (SPI). In addition, these devices offer 128 bytes of EEPROM for non volatile data storage.

The ST626xB are upward compatible with their ST626x counterparts. They feature in addition to RC network, user configurable Power on Reset delay and an External STOP Mode Control option to enlarge the range of power consumption/safety trade-offs.

ST6260B and ST6265B are well suited for automotive, appliance and industrial applications. The ST62E60 and ST62E65B EPROM versions are available for prototypes and low-volume production; also OTP versions are available.

PIN DESCRIPTION

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

OSCin and OSCout. These pins are internally connected with 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 pin OSCout and the ground.The oscillator frequency is internally divided by 1, 2 or 4 by a software controlled divider. 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 to the beginning of its program.

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

NMI. The NMI pin provides the capability for asynchronous interrupt applying an external not maskable interrupt to the MCU. The NMI is falling edge sensitive. It is provided with an on-chip pull-up resistor and Schmitt trigger characteristics.

When the option EXTERNAL STOP MODE CONTROL is enabled the NMI pin, in addition, enables the control of how the STOP instruction is processed.

PC1/TIM1/Ain. This pin can be used as a Port C I/O bit, as 1 I/O pin or as analog input for the on-chip A/D converter. This pin is available only on the ST6265B (28 pin version). If programmed to be the TIMER1 pin, in input mode it is connected to the prescaler and acts as external timer clock or as control gate for the internal timer clock. In the output mode the timer pin outputs the data bit when a time out occurs.

To use this pin as TIMER 1 output a dedicated bit in the TIMER1 Status/Control Register must be set. To use this pin as input pin the I/O pin has to be programmed as input. The analog mode should be programmed to use the line as an analog input.

PB6/ARTIMin, PB7/ARTIMout. These pins are either Port B I/O bits or the Input and Output pins of the . 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.

PA0-PA7. These 8 lines are organized as one I/O port (A). PA4-PA7 are not available on ST6260B (20 pin version). 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, open-drain or push-pull output.

PB0-PB3, PB4, PB5. These 6 lines are organized as one I/O port (B). PB4, PB5 are available only on the ST6265B (28 pin version). When the External STOP Mode Control is disabled, each line may be configured under software control as input with or without internal pull-up resistor, interrupt generating input with pull-up resistor, open-drain or push-pull output. In output mode these lines can also sink 20mA for direct LED and TRIAC driving. When the External STOP Mode Control is enabled, PB0 is forced as open drain output. The other lines are unchanged.

PC0-PC4. These 5 lines are organized as one I/O port (C). PC0 and PC1 are not available on ST6260B (20 pin version). 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. 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.

ST62xx CORE

The core of the ST62xx Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in Figure 5; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST62xx Family core has six registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained 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 is addressed in the data space as RAM location at address FFh. Accordingly the ST62xx instruction set can use the accumulator as any other register of the data space.

Indirect Registers (X, Y). These two indirect registers are used as pointers to the memory locations in the data space. They are used in the register-indirect addressing mode. These registers can be addressed in the data space as RAM locations at

Figure 4. ST62xx Core Programming Model

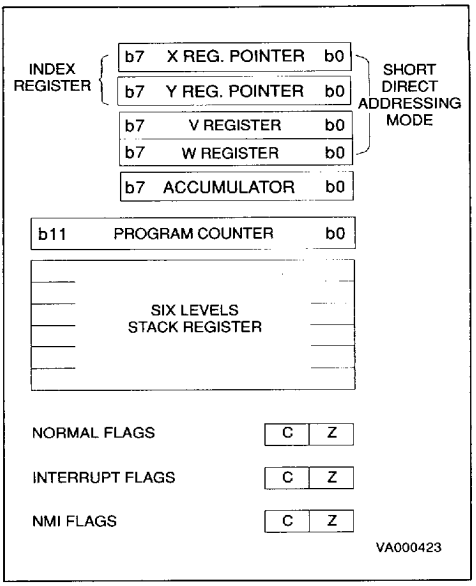
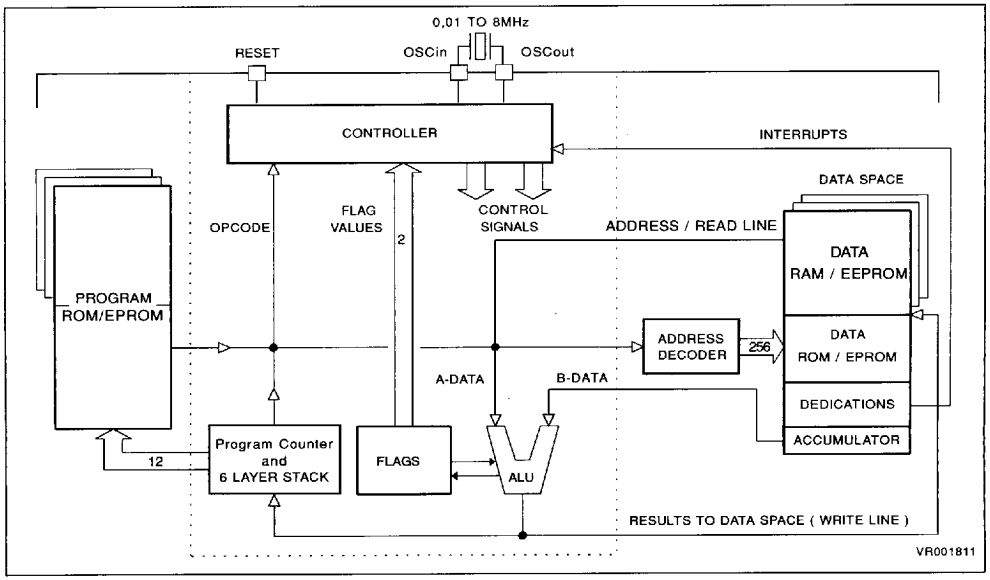


Figure 5. ST62xx Core Block Diagram



ST62xx CORE (Continued)

addresses 80h (X) and 81h (Y). They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST62xx 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 one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at addresses 82h (V) and 83h (W). They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST62xx 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 that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, the further program space can be addressed by using the Program Bank Switch register.

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

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

Flags (C, Z)

The ST62xx core includes three pairs of flags that correspond to 3 different modes: 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, one pair is used during the interrupt mode (CI, ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST62xx core uses the pair of flags that correspond to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST62xx core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. It should be observed that each flag set can only be addressed in its own routine (Not-maskable interrupt, normal interrupt or main routine). The flags are not cleared during the context switching and so remain in the state they were at the exit of the last routine switching.

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, and participates in the rotate left instruction.

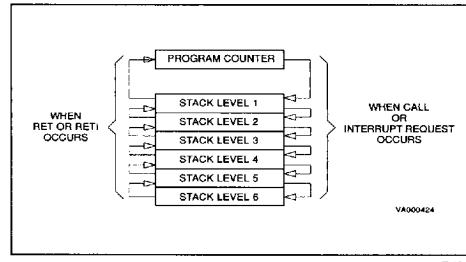
The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

The switching between the three sets of flags is automatically performed when an NMI, an interrupt or a RETI instructions occurs. As the NMI mode is automatically selected after the reset of the MCU, the ST62xx core uses at first the NMI flags.

Stack

The ST62xx core includes true LIFO hardware stack that 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 is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be 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. These two operating modes are described in Figure 6. Since the accumulator, as all other data space registers, is not stored in this stack the handling of these registers should be performed inside the subroutine. The stack pointer will remain in its deepest position if more than 6 calls or interrupts are executed, so that 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 6. Stack Operation



MEMORY SPACES

The MCU operates in three different memory spaces: program space, data space, and stack space. A description of these spaces is shown in the following figures.

Program Space

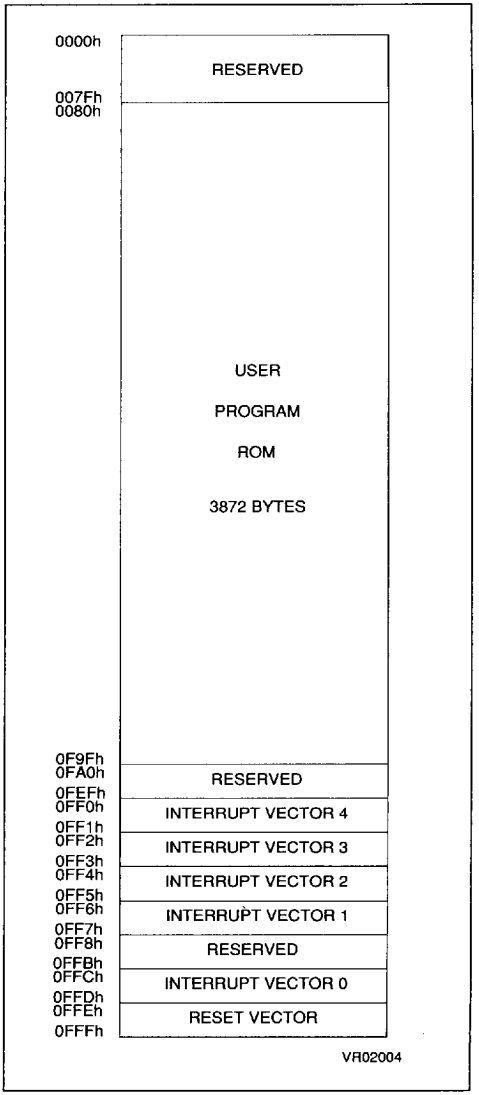
The program space is physically implemented in the ROM memory and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and the user vectors. It is addressed by the 12-bit Program Counter register (PC register) and so the ST62xx core can directly address up to 4K bytes of Program Space. ST62 devices with more than 4K ROM use ROM banked program memory (not available on ST6260B, ST6265B).

The ST6260B,65B program space can be protected against external reading of the ROM contents when the READOUT PROTECTION mask option is selected. If this option is selected, the user can blow a dedicated fuse on the silicon by applying a high voltage at V_{PP}, (see detailed information in the "Electrical Specification").

Note:

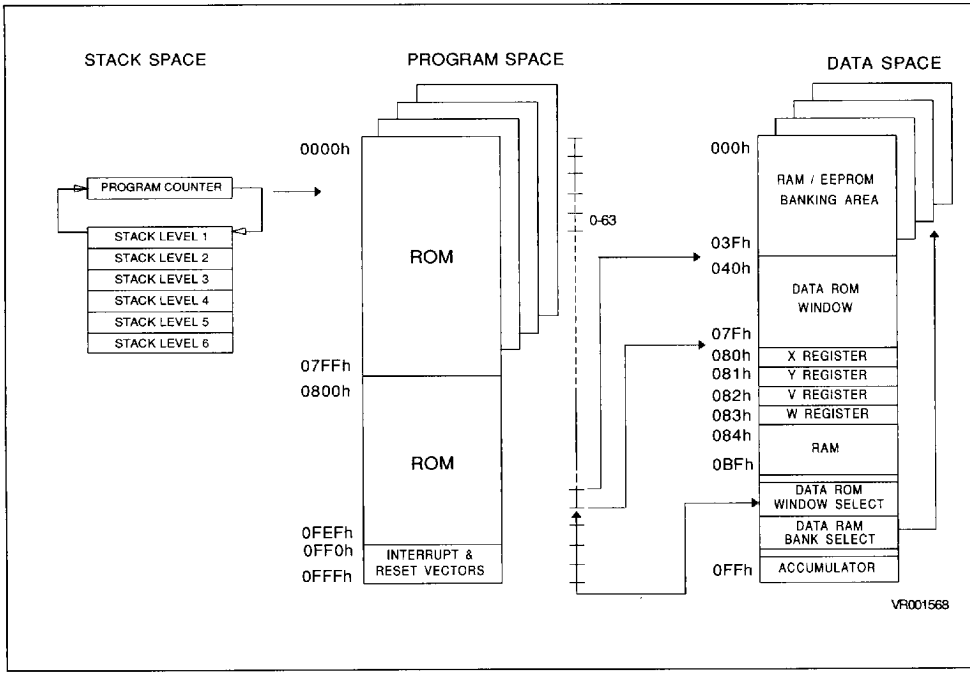
Once the fuse is blown, it is no longer possible, even for SGS-THOMSON, to gain access to the ROM contents. Returned parts with blown fusecan therefore not be accepted.

Figure 7. ST6260B/65B Program ROM Memory Map



MEMORY SPACES (Continued)

Figure 8. Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Data Space

The instruction set of the ST62xx core operates on a specific space, named Data Space, that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM and EEPROM memory, ST62xx core/peripheral registers, and read-only data such as constants and look-up tables.

Data ROM. All the read-only data is physically implemented in the ROM memory in which the Program Space is also implemented. The ROM memory contains consequently the program to be executed, the constants and the look-up tables needed for the program.

The locations of Data Space in which the different constants and look-up tables are addressed by the ST62xx core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM memory .

Data RAM/EEPROM. The ST6260B/65B offer 128 bytes of data RAM memory and 128 bytes of EEPROM. 64 bytes of RAM are directly addressed in data space in the range 080h-0BFh (static space). The additional RAM and EEPROM are addressed using the banks of 64 bytes located between addresses 00h and 3Fh.

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Figure 9. ST6260B/65B Data Memory Space

DATA and EEPROM	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM	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 1 PRESCALER REGISTER	0D2h
TIMER 1 COUNTER REGISTER	0D3h
TIMER 1 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

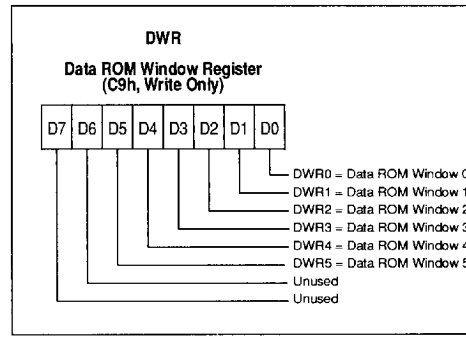
MEMORY SPACES (Continued)

Data Window register (DWR)

The Data ROM window is located from address 040h to address 7Fh in the Data space. It allows the direct reading of 64 consecutive bytes located anywhere in the ROM memory between the addresses 0000h and 0FFFh. All the bytes of the ROM memory can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM memory in writing the appropriate code in the Write-only Data Window register (DWR register, location C9h).

The DWR register can be addressed like a RAM location in the Data Space at the address C9h, nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to move the 64-byte read-only data window (from the 40h address to 7Fh address 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 a data in the ROM memory is obtained by the concatenation of 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 40h of dataspace, and 0 is loaded in the DWR register, the phisycal addressed location in ROM is 00h. The DWR register is not cleared at reset, therefore it must be written to before the first access to the Data ROM window area.

Figure 11. Data ROM Window Register



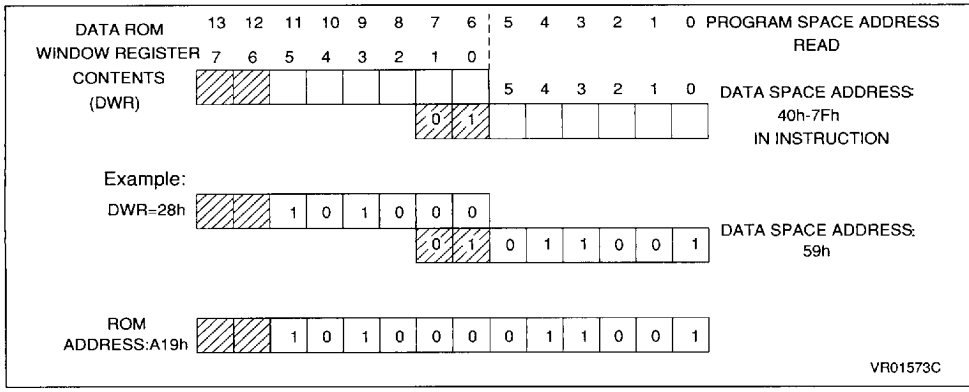
D7-D6. Not used.

DWR5-DWR0. These are the Data ROM Window bits that correspond to the upper bits of the data ROM space.

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 10. Data ROM Window Memory Addressing



MEMORY SPACES (Continued)

Data RAM/EEPROM Bank Register (DRBR)

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.

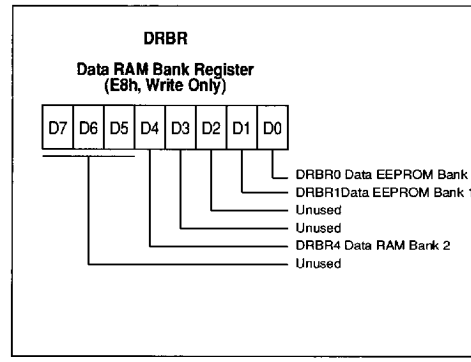
The DRBR register can be addressed like a RAM location in the Data Space at the address E8h; nevertheless it is a write only register that cannot be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of bank has to be 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.

Table 1. Data RAM Bank Register Set-up

DRBR Value	Selection
00h	None
01h	EEPROM Page 0
02h	EEPROM Page 1
10h	RAM Page 2
Other	Reserved

Figure 12. Data RAM/EEPROM Bank Register



D7-D5. These bits are not used.

DRBR4. This bit, when set, selects RAM page 2.

D3-D2. These bits are not used.

DRBR1. This bit, when set, selects EEPROM page 1.

DRBR0. This bit, when set, selects EEPROM page 0.

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.

MEMORY SPACES (Continued)

EEPROM Description

The data space of ST62xx family from 00h to 3Fh is paged as described in Table 2. The ST6260B/65B has 128 bytes of EEPROM located in two pages of 64 bytes (page 0 and 1).

The EEPROM is physically organized in 32 byte modules (2 modules per page) and does not require dedicated instructions to be accessed in reading or writing. Once selected through the Data RAM Bank Register, the active EEPROM page is controlled by the EEPROM Control Register (EECTL) located at address EAh. E20FF bit of the EECTL register must be cleared to "0" prior to any write or read access to the EEPROM. If no bank is selected or if E2OFF is set, any access is meaningless.

Programming must be enabled by setting bit E2ENA of register EECTL.

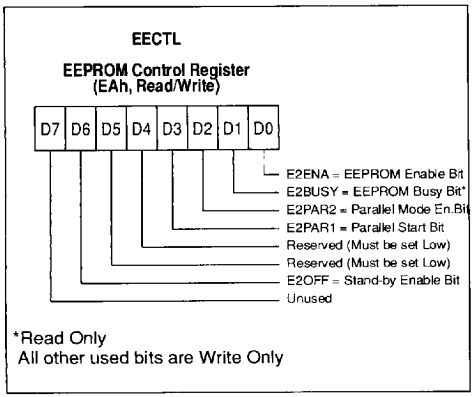
Bit E2BUSY of EECTL register is set to 1 when the EEPROM is performing a programming cycle. Any access to the EEPROM when E2BUSY is set to 1 is meaningless.

Provided E2OFF and E2BUSY are cleared to 0, an EEPROM location is read like any other data location, also in term of access time.

Writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. PMODE consists in simultaneously programing 8 bytes of the same row.

D7. Not Used

Figure 13. EEPROM Control Register



E2OFF. *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 the lowest values.

D5, D4. Reserved, must be set to zero.

E2PAR1. *WRITE ONLY.* Once in Parallel Mode, as soon as the user software sets the E2PAR1 bit the parallel writing of the 8 adjacent registers will start. It is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; the undefined bytes are unaffected by the parallel programming.

E2PAR2. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more than one byte at a time). If E2PAR2 is set and the parallel start bit (E2PAR1) is low, up to 8 adjacent bytes can be written at maximum speed, the contents being stored in volatile registers. 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. 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, leaving the EEPROM registers unchanged.

E2BUSY. *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 read or write EEPROM operation; any attempt to access the EEPROM while the busy bit is set will be aborted and the writing procedure in progress completed.

E2ENA. *WRITE ONLY.* This bit enables programming of the EEPROM cells. It must be set to one before any write into the EEPROM register. Any attempt to write to the EEPROM when E2ENA is low is meaningless and will not trigger a write cycle.

This register is cleared at reset.

Notes:

The data to write has to be written directly at the address that it will have inside the EEPROM space. There is no buffer memory between the data RAM and the EEPROM spaces.

When the EEPROM is busy (E2BUSY = "1") EECTL can not 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 to "1".

Care is required when handling the EECTL register as some bits are write only. For this reason, it is not allowed to change the EECTL 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

MEMORY SPACES (Continued)

Table 2. EEPROM Parallel Write Row Structure

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 at the same time in Parallel Write mode

register must be saved in a RAM location, and each time the program writes to EECTL 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 EECTL is not affected.

Additional Notes on Parallel Mode. If the user wishes to perform parallel programming, the first action should be to set the E2PAR2 bit to one. From this time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by resetting E2PAR2 without programming the EEPROM. After the ROW address latching the ST62xx can "see" only one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while E2PAR2 is set.

As soon as E2PAR2 bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or in a subset. 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, 1Bh and then sets E2PAR1, these three registers will be modified at the same time; the remaining bytes will be unaffected. Note that E2PAR2 is internally reset at the end of the programming procedure. This implies that the user must set E2PAR2 bit between two parallel programming procedures. Note that if the user tries to set E2PAR1 while E2PAR2 is not set there will not be any programming procedure and the E2PAR1 bit will be unaffected. Consequently E2PAR1 bit cannot be set if E2ENA is low. E2PAR1 can be affected by the user to set it, only if E2ENA and E2PAR2 bits are also set to one.

TEST MODE

For normal operation the TEST pin must be held low when reset is active. An on-chip 100kΩ pull-down resistor is internally connected to the TEST pin.

INTERRUPTS

The ST62xx core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address (see Table 1). When a source provides an interrupt request, and the request processing is also enabled by the ST62xx core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction).
Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The ST6260B and ST6265B microcontrollers have eight different interrupt sources associated to five interrupt vectors as it is described in table below.

Table 3. Interrupt Vector/Source Relationship

Interrupt Source	Vector	Vector Address
NMI	Interrupt vector #0	(FFCh, FFDh)
Port A & B	Interrupt vector #1	(FF6h, FF7h)
Port C & SPI	Interrupt vector #2	(FF4h, FF5h)
AR TIMER	Interrupt vector #3	(FF2h, FF3h)
TIMER1 & ADC	Interrupt vector #4	(FF0h, FF1h)

Interrupt Vectors Description

The ST62xx core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines in the static page of the Program Space.

- The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at addresses FFCh,FFDh in the Program Space. On ST6260B and ST6265B this vector is associated with the external falling edge sensitive interrupt pin (NMI).
- The interrupt vector located at addresses FF6h, FF7h is named interrupt vector #1. It is associated with Port A and Port B pins. It can be programmed by software either in the falling edge detection mode or in the low level sensitive detection mode according to the code loaded in the Interrupt Option Register (IOR).
- The interrupt vector located at addresses FF4h, FF5h is named interrupt vector #2. It is associate with Port C pins and the SPI peripheral can be programmed by software either in the falling edge detection mode or in the positive edge detection mode according to the code loaded in the Interrupt Option Register (IOR).
- The interrupt vector located at addresses FF2h, FF3h is named interrupt vector #3. It is associated with the AR TIMER peripheral.
- The interrrupt vector loaded at address FF0h, FF1h is named interrupt vector #4. It is associated with the TIMER 1 and the A/D converter peripherals.

All the on-chip peripherals have an interrupt request flag bit (TMZ for timer, EOC for A/D), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI for timer, EAI for A/D) that must be set to one to allow the transfer of the flag bit to the core.

Interrupt Priority

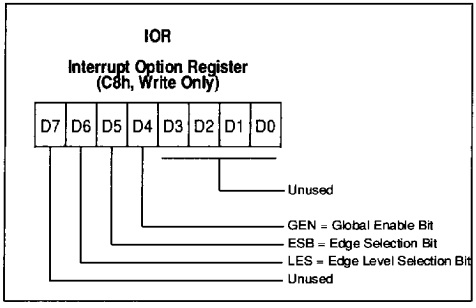
The non-maskable interrupt request NMI has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the four other interrupts can not interrupt each other. If more than one interrupt request are pending, they are processed by the ST62xx core according to their priority level: vector #1 has the higher priority while vector #4 the lower.
The priority of each interrupt source is fixed.

INTERRUPT (Continued)

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the address C8h, nevertheless it is a write-only register that cannot be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 14. Interrupt Option Register



D7. D3-D0 These bits are not used.

LES. Level/Edge Selection Bit. When this bit is set to one, the interrupt #1 (Port A, B lines) is low level sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

ESB. Edge Selection Bit. When this bit is set to one, the interrupt #2 (Port C lines) is positive edge sensitive, when cleared to zero the negative edge sensitive interrupt is selected.

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

This register is cleared on reset.

Table 4. Interrupt Option Register Description

GEN	SET	Enable all interrupts
	CLEARED	Disable all interrupts
ESB	SET	Rising edge mode on interrupt input #2
	CLEARED	Falling edge mode on interrupt input #2
LES	SET	Level-sensitive mode on interrupt input #1
	CLEARED	Falling edge mode on interrupt input #1
OTHERS	NOT USED	

External Interrupts Operating Modes

The NMI interrupt is associated to the NMI pin of the ST6260B/65B. The interrupt request is generated by a falling edge applied to the NMI pin. The NMI interrupt pin signal is latched and is automatically reset by the core at the beginning of the non-maskable interrupt service routine. An on-chip pull-up resistor and a Schmitt trigger are available at pin NMI.

The two interrupt sources associated with the falling/rising edge mode of the external interrupt pins (Ports A and B vector #1, Ports C vector #2) are connected to two internal latches. Each latch is set when a falling/rising edge occurs and is cleared when the associated interrupt routine is started. So, the occurrence of an external interrupt request is stored: a second interrupt, that occurs during the processing of the first one, will be processed as soon as the first one has been finished (if there is not an higher priority interrupt request). If more than one interrupt occurs during the processing of the first one, these other interrupt requests will be lost.

The storage of the interrupt requests is not available in the level sensitive detection mode. To be taken into account, the low level must be present on the interrupt pin when the core samples the line after the execution of the instructions. At the end of each instruction the core tests the interrupt lines and if there is a pending interrupt request the next instruction is not executed and the related interrupt routine is executed.

Note:

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

INTERRUPT (Continued)

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 does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure:

ST62xx actions

- Interrupt detection
- The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (or the NMI flags)
- The value of the PC is stored in the first level of the stack
- The normal interrupt lines are inhibited (NMI still active)
- First internal latch is cleared
- The related interrupt vector is loaded in the PC.

User actions

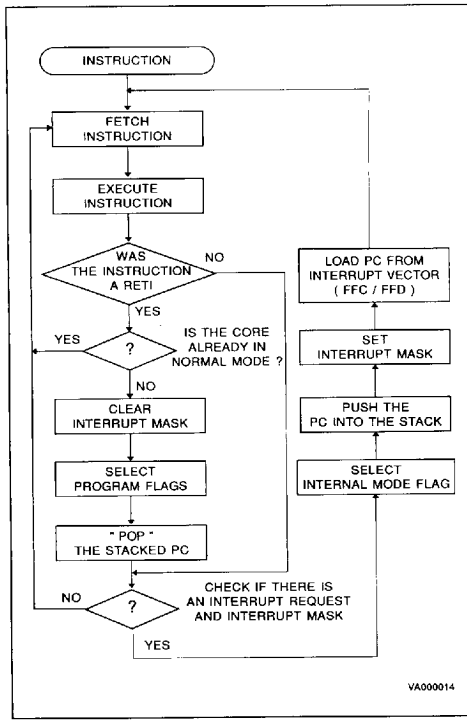
- User selected registers are saved inside the interrupt service routine (normally on a software stack)
- The source of the interrupt is found by polling (if more than one source is associated to the same vector) the interrupt flag of the source.
- Interrupt servicing
- Return from interrupt (RETI)

ST62xx actions

- Automatically the ST62xx core switches back to the normal flags (or the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request (by polling). The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack. After the RETI instruction execution, the core carries out the previous actions and the main routine can continue.

Figure 15. Interrupt Processing Flow-Chart



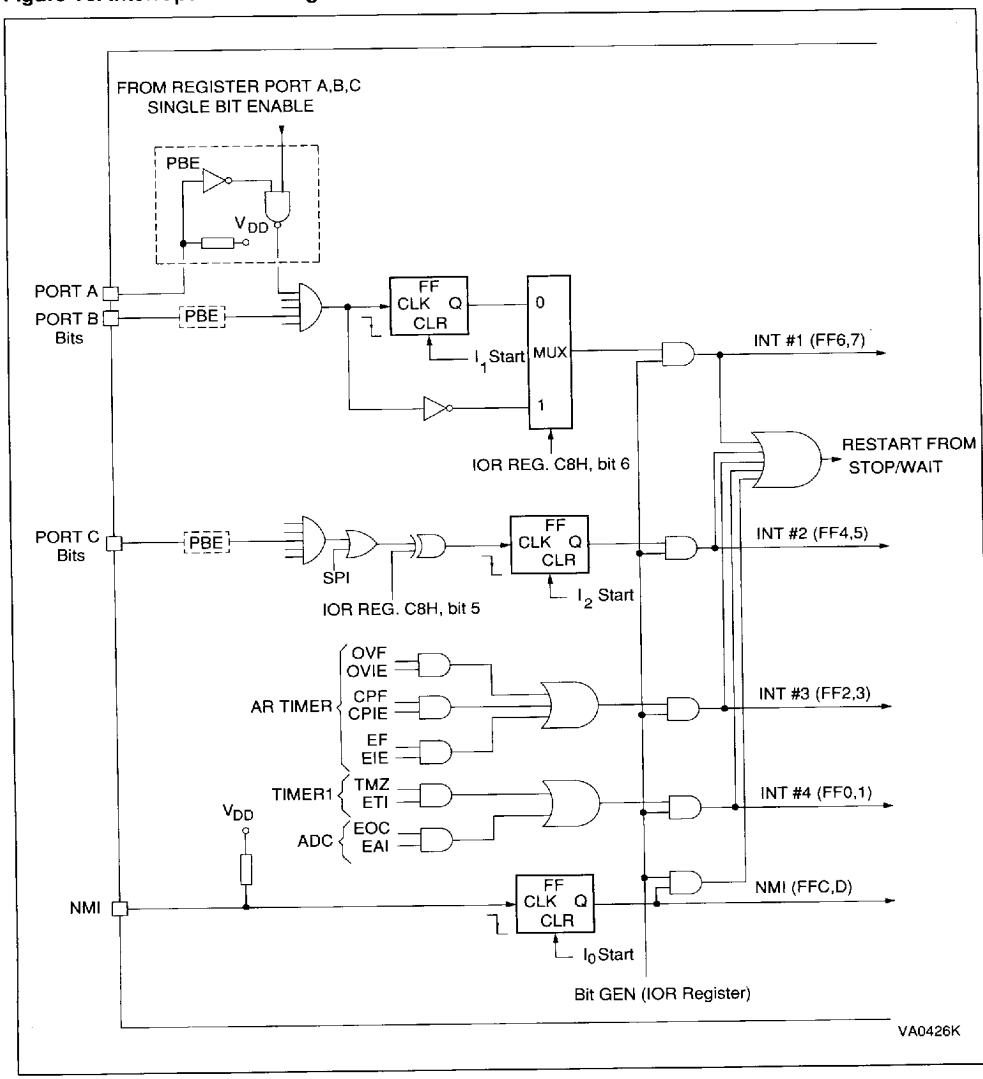
INTERRUPT (Continued)

Table 5. Interrupt Requests and Mask Bits

Peripheral	Register	Register Address	Mask bit	Masked Interrupt Source	Interrupt Vector
GENERAL	IOR	C8h	GEN	All sources, excluding NMI	
TIMER 1	TSCR1	D4h	ETI	TMZ: TIMER 1 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	SPIF: 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

INTERRUPT (Continued)

Figure 16. Interrupt Circuit Diagram



RESET

The ST6260B/65B can be reset in three ways: by the external reset input ($\overline{\text{RESET}}$) tied low, by power-on reset and by the digital Watchdog peripheral

RESET Input

The $\overline{\text{RESET}}$ pin can be connected to a device of the application board in order to restart the MCU during its operation. The activation of the $\overline{\text{RESET}}$ pin may occur in the RUN, WAIT or STOP mode. This input has to be used to reset the MCU internal state and provide a correct start-up procedure. The pin is active low and has a schmitt trigger input. The internal reset signal is generated by adding a delay to the external signal. Therefore even short pulses at the $\overline{\text{RESET}}$ are accepted, provided V_{DD} has finished its rising phase and the oscillator is running correctly (normal RUN or WAIT modes). The MCU is kept in the Reset state as long as the $\overline{\text{RESET}}$ pin is held low.

If the $\overline{\text{RESET}}$ activation occurs in the RUN or WAIT mode, the processing of the program is stopped (in RUN mode only) and the Input/Outputs are placed in input with pull-up resistors. When the level on the $\overline{\text{RESET}}$ pin becomes high, the initialization sequence is executed just after the internal delay.

If a $\overline{\text{RESET}}$ pin activation occurs in the STOP mode, the oscillator starts and all the inputs/outputs are configured in input with pull-up resistors. When the level of the $\overline{\text{RESET}}$ pin becomes high, the initialization sequence is started just after the internal delay.

Power-on Reset

The function of the POR consists in waking up the MCU during the power-on sequence. At the beginning of this sequence, the MCU is configured in the Reset state: every Input/Output port is configured in input with pull-up resistor and no instruction is executed. When the power supply voltage becomes sufficient, the oscillator starts to operate, nevertheless an internal delay is generated to allow the oscillator to be completely stabilized before the execution of the first instruction. The initialization sequence is executed just after the internal delay.

The internal delay is generated by a mask option configurable on-chip counter. When the 2048 CYCLES DELAY option is selected the internal reset is released 2048 cycles of the oscillator after the external reset is released. When the 32768 CYCLES DELAY option is selected, the delay is 32768 cycles of the oscillator.

Note:

To have a correct start-up, the user should take care that the internal reset is not released before the VDD level is sufficient to allow MCU operation at the chosen frequency (see Recommended Operating Conditions).

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

RESET (Continued)

Watchdog Reset

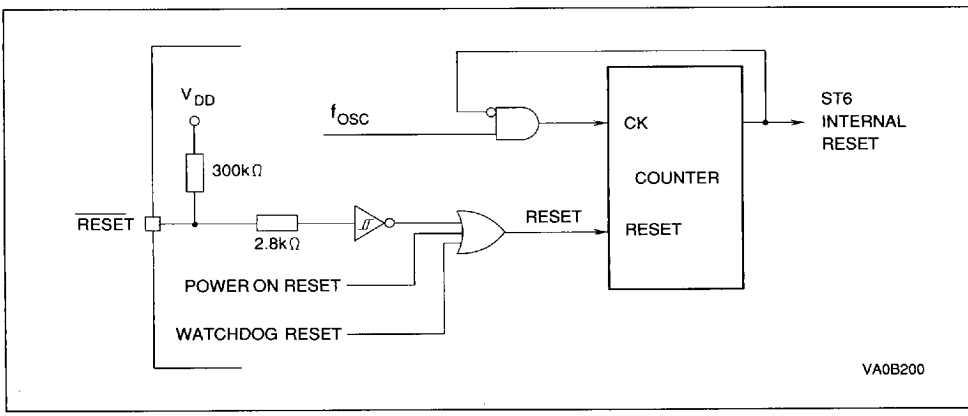
The ST6260B and ST6265B provide an on-chip watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed, preventing the end-of-count being reached, the internal reset is activated. This, in particular, resets the watchdog. The MCU restarts as with normal reset from $\overline{\text{RESET}}$ pin including the internal delay.

Application Notes

An external resistor between V_{DD} and reset pin is not required because an internal pull-up device is provided.

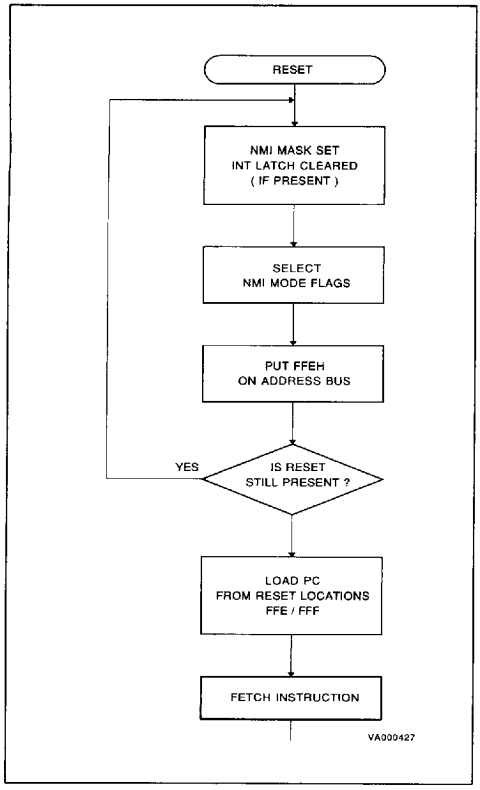
The POR device operates in a dynamic manner in the way that it brings about the initialization of the MCU when it detects a dynamic rising edge of the V_{DD} voltage. The typical detected threshold is about 2 volts, but the actual value of the detected threshold depends on the way in which the V_{DD} voltage rises up. The POR device *DOES NOT* allow the supervision of a static or slowly rising or falling edge of the V_{DD} voltage.

Figure 17. Reset Circuit



RESET (Continued)

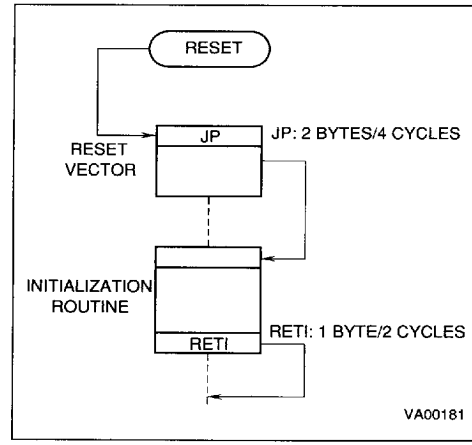
Figure 18. Reset and Interrupt Processing Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to the program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset a NMI is automatically activated so that the core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine the ST62xx will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced

Figure 19. Restart Initialisation Program Flow-Chart



RESET (Continued)

Table 6. Reset Value

Register	Address	Value	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		
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 1 Status/Control	0D4h	00h	TIMER 1 disabled
AR TIMER Mode Control Register	0D5h		AR TIMER stopped
AR TIMER Status/Control 1 Register	0D6h		
AR TIMER Status/Control 2 Register	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 1 Counter Register	0D3h	FFh	
TIMER 1 Prescaler Register	0D2h	7Fh	Max count loaded
Watchdog Counter Register	0D8h	FEh	
A/D Control Register	0D1h	40h	A/D in Standby

WAIT & STOP MODES

The WAIT and STOP modes have been implemented in the ST62xx core in order to reduce the consumption of the product when the latter has no instruction to execute. These two modes are described in the following paragraphs

WAIT Mode

The MCU goes into the WAIT mode as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a "software frozen" state where the core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage, but where the peripherals are still working.

The WAIT mode can be used when the user wants to reduce the consumption of the MCU when it is idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide a clock signal to the peripherals. The TIMER 1 and counting may be enabled as well as both Timer interrupts before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. The above explanation related to the timers applies also to the A/D converter and the SPI.

If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU enters a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behavior depends on the state of the ST62xx core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case is described in the following paragraphs. In any case, the ST62xx core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

If the Watchdog is disabled the 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 contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an interrupt request or Reset activation to output from the STOP state. The interrupt request can be issued by a pin or by an externally clocked timer.

If the exit from the STOP mode is performed with a general RESET (by the activation of the external pin) the MCU will enter a normal reset procedure as described in the RESET chapter. The case of an interrupt depends on the state of the ST62xx core before the initialization of the STOP sequence and also of the kind of the interrupt request that is generated.

This case will be described in the following paragraphs. In any case, the ST62xx core generates a delay after the occurrence of the interrupt request in order to wait the complete stabilization of the oscillator before the execution of the first instruction.

Exit from WAIT and STOP Modes

The following paragraphs describe the output procedure of the ST62xx core from WAIT and STOP modes when an interrupt occurs (not a RESET). It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT or STOP sequence, but also of the type of the interrupt request that is generated.

Normal Mode. If the ST62xx core was in the main routine when the WAIT or STOP instruction has been executed, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the STOP or the WAIT instruction is executed if no other interrupts are pending.

Not Maskable Interrupt Mode. If the STOP or WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST62xx core outputs from the stop or wait mode as soon as any interrupt occurs: the instruction that follows the STOP or the WAIT instruction is executed and the ST62xx core is still in the non-maskable interrupt mode even if another interrupt has been generated.

WAIT & STOP MODES (Continued)

Normal Interrupt Mode. If the ST62xx core was in the interrupt mode before the initialization of the STOP or WAIT sequence, it outputs from the stop or wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

- If the interrupt is a normal interrupt, the interrupt routine in which the wait or stop was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST6xx core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.
- If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then the routine in which the wait or stop was entered will be completed with the execution of the instruction that follows the STOP or the WAIT and the ST6xx core remains in the normal interrupt mode.

Notes:

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

- selecting 4 as ratio of the Oscillator divider.
- configuring unused I/O as input without pull-up with well defined logic levels.
- placing the A/D converter in its power down mode by clearing the PDS bit in the A/D control register before entering the STOP instruction.
- putting the EEPROM on-chip memory in stand-by mode by setting the E2OFF bit in EEPROM Control Register to one.
- setting bit D3 of the Oscillator Control Register to one.
- stopping all external clocks (TIMER 1, AR TIMER and SPI).

When the watchdog is active (independent of its mode), the STOP instruction is deactivated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

If all the interrupt sources are disabled (including NMI if GEN is low), the restart of the MCU can only be done by a Reset activation. The WAIT and STOP instructions are not executed if an enabled interrupt request is pending.

ON-CHIP CLOCK OSCILLATOR

The ST6260B/65B on-chip oscillator has been designed to require a minimum of external components and to reduce the oscillator power consumption.

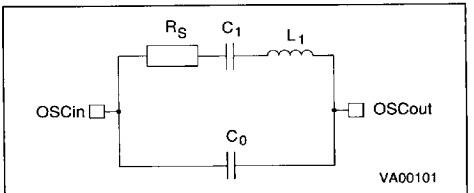
A quartz crystal, a ceramic resonator, an RC network or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The clock generator options connection methods are shown in Figure 22.

The oscillator is configured by mask option. When the CRYSTAL/RESONATOR option is selected, it must be used with a quartz crystal, a ceramic resonator or an external signal. To use the oscillator with an RC network, the RC NETWORK option must be selected.

A programmable divider is provided in order to adjust the internal clock of the micro (core and peripherals) to the best power consumption/performance trade-off. The Division Ratio is selected through the Oscillator Control Register located at address 0DCh.

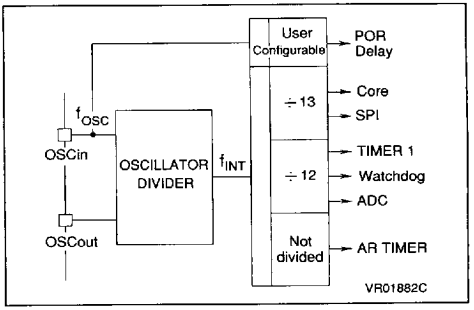
The internal frequency is directly used to clock the AR TIMER. It is further divided by 12 to produce the TIMER 1, the A/D converter and the Watchdog clock and by 13 for the core and SPI clock.

Figure 20. Crystal Parameters



Crystal Parameters: AT-cut Parallel Resonance Crystal
C0 = Parallel Resonance Capacitance

Figure 21. Internal Clock Circuits



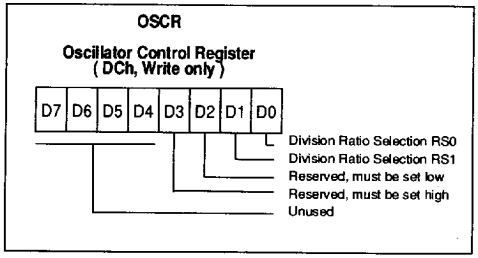
ON-CHIP CLOCK OSCILLATOR (Continued)

With a 8MHz external frequency, the fastest machine cycle is therefore 1.625µs.

The machine cycle is the smallest unit needed to execute any operation (i.e.increment the program counter). An instruction may need two, four, or five machine cycles to be executed.

Oscillator Control Register

Figure 22. Oscillator Control Registers



D7-D4. These bits are not used.

D3. Reserved - Cleared at Reset. THIS BIT MUST BE SET TO 1 BY USER PROGRAM to achieve lowest power consumption.

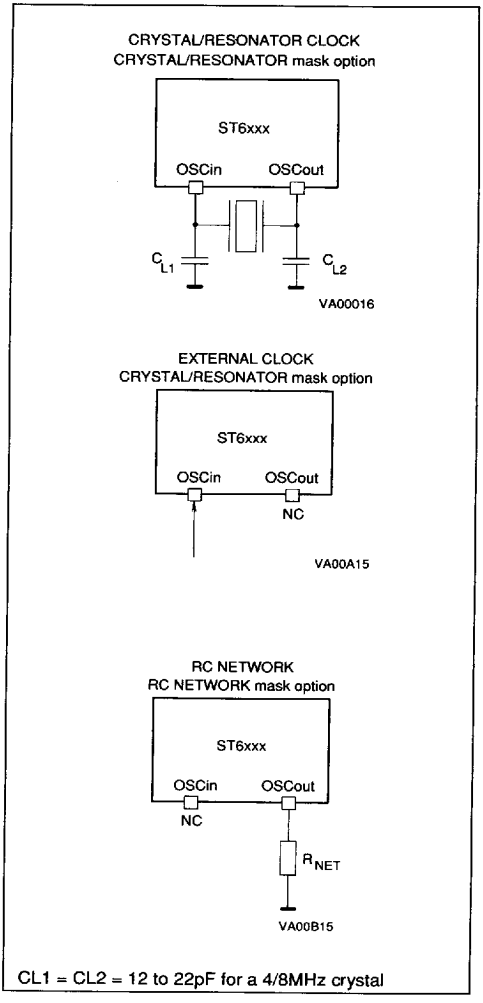
D2. Reserved. Must be kept low.

RS1-RS0. These bits select the division ratio of the Oscillator Divider in order to generate the internal frequency. The following selections 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 23. Oscillator Connection



CL1 = CL2 = 12 to 22pF for a 4/8MHz crystal

INPUT/OUTPUT PORTS

The ST6260B and ST6265B microcontroller have respectively 13 and 21 Input/Output lines that can be individually programmed either in the input mode or the output mode with the following software selectable options:

- Input without pull-up and without interrupt
- Input with pull-up and with interrupt
- Input with pull-up without interrupt
- Analog inputs (PA0-PA7, PC0-PC4)
- Timer 1 I/O line (PC1, not available on ST6260B)
- AR Timer I/O lines (PB6, PB7)
- SPI control signals (PC2-PC4)
- Push-pull output
- Standard Open drain output
- 20mA Open drain output PB lines.

The lines are organized in three Ports (Port A, B and C).

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

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

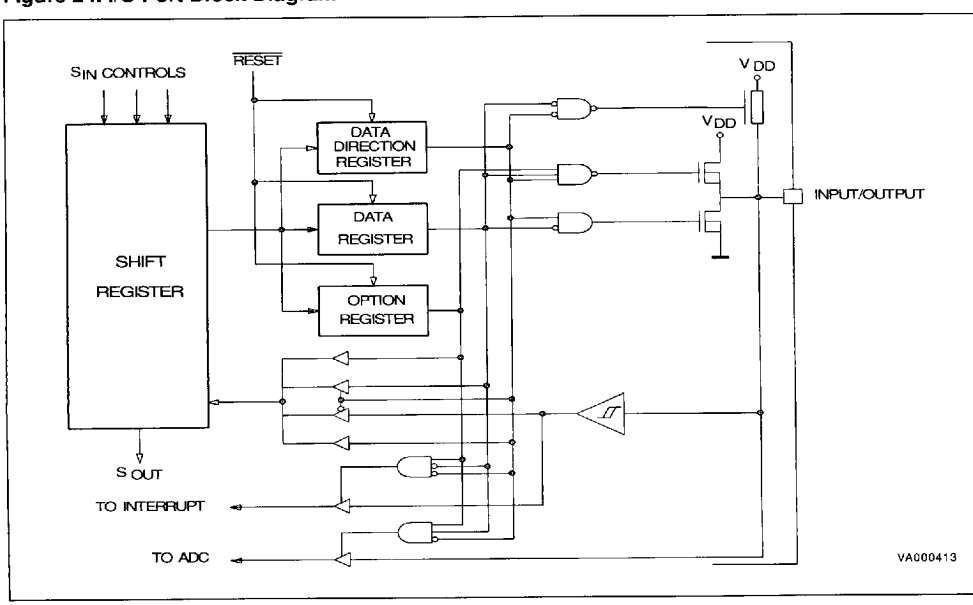
Single-bit operations on I/O registers are possible but care is necessary because reading in input mode is done from I/O pins while writing will directly affect the Port data register causing an undesired change of the input configuration.

The three Data Direction registers (DDRA, DDRB and DDRC) allow the selection of the data direction of each pin (input or output).

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 the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up/no-interrupt is selected on all the pins, thus avoiding pin conflicts.

Figure 24. I/O Port Block Diagram



INPUT/OUTPUT PORTS (Continued)

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations.

This is achieved by writing the relevant bit in the data (DR), data direction register (DDR) and option registers (OR). Table 7 shows all the port configurations that can be selected by user software.

Input Option Description

Interrupt Option. All the input lines can be individually connected by software to the interrupt lines of the ST62xx core according to the codes programmed in the OR and DR registers. The pins of Port A and B are "ORed" and are connected to the interrupt associated to the vector #1. Pins of ports C are "ORed" with the SPI interrupt line and connected to interrupt vector #2. The interrupt modes (falling edge sensitive, rising edge sensitive) can be selected by software for each port by programming the IOR register.

Analog Input Option. The PA0-PA7 and PC0-PC4 pins can be configured to be analog inputs according to the codes programmed in the OR and DR registers. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter. *ONLY ONE* pin should be programmed as analog input at a time, otherwise the selected inputs will be shorted.

Figure 25. I/O Port Data Registers

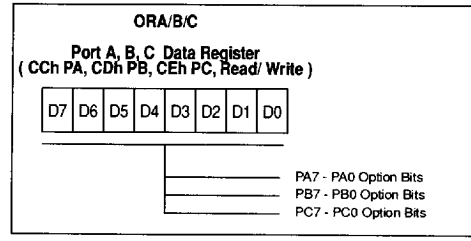


Figure 26. I/O Port Data Direction Registers

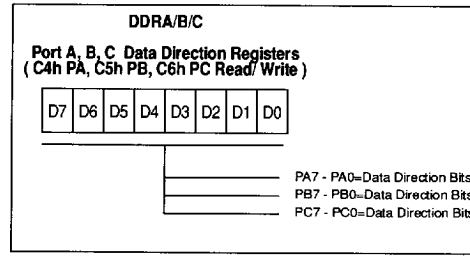
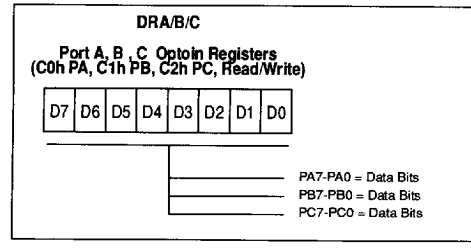


Figure 27. I/O Port Option Registers



Note: For complete coding explanation refer to Table 7

Table 7. I/O Port Options 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, with interrupt
0	1	1	Input	No pull-up, no interrupt (Port B pins)
			Input	Analog input (Ports A and C pins)
1	0	X	Output	Open-drain output (20mA sink current for Port B pins)
1	1	X	Output	Push-pull output (20mA sink current for Port B pins)

Notes: X. Means don't care.

INPUT/OUTPUT PORTS (Continued)

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.

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.

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.

Note. Switching the I/O ports from one state to another should be done in a way that no unwanted side effects can happen. The recommended safe transitions are shown below. All other transitions are risky and should be avoided during change of operation mode as it is most likely that there will be an unwanted side-effect such as interrupt generation or two pins shorted together by the analog input lines.

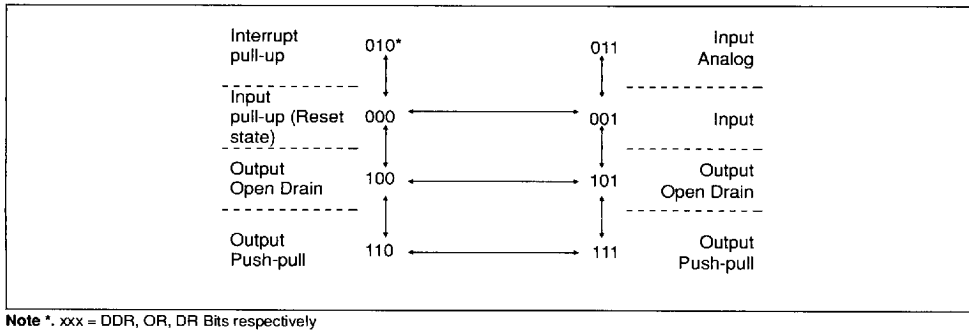
Single bit instructions (SET, RES, JRR and JRS) should be used very carefully with Port A and B data registers because these instructions make an implicit read and write back of the whole addressed register byte. In port input mode however data register address reads from input pins, not from data register latches and data register information in input mode is used to set characteristics of the input pin (interrupt, pull-up, analog input), therefore these characteristics may be unintentionally reprogrammed depending on the state of input pins. As general rule is better to use single bit instructions on data register only when the whole port is in output mode. If input or mixed configuration is needed it is recommended to keep a copy of the data register in RAM. On this copy it is possible to use single bit instructions, then the copy register could be written into the port data register.

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

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

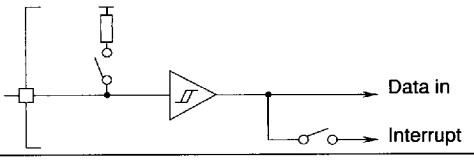
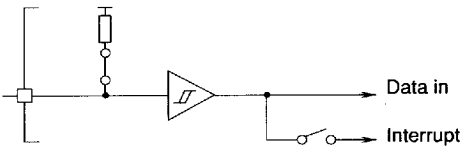
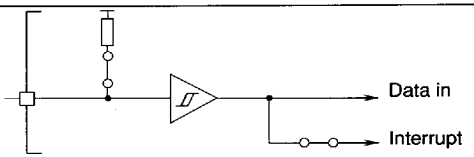
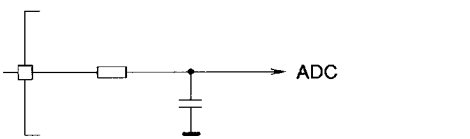
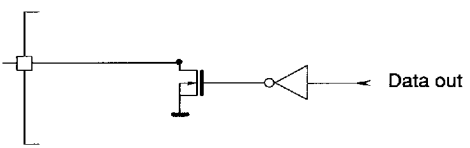
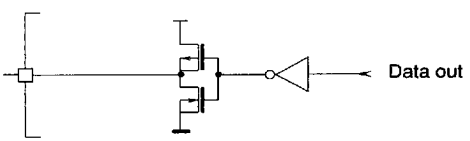
Figure 28. State Transition Diagram for Safe Transitions



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

INPUT/OUTPUT PORTS (Continued)

Table 8. I/O Port Options Selection

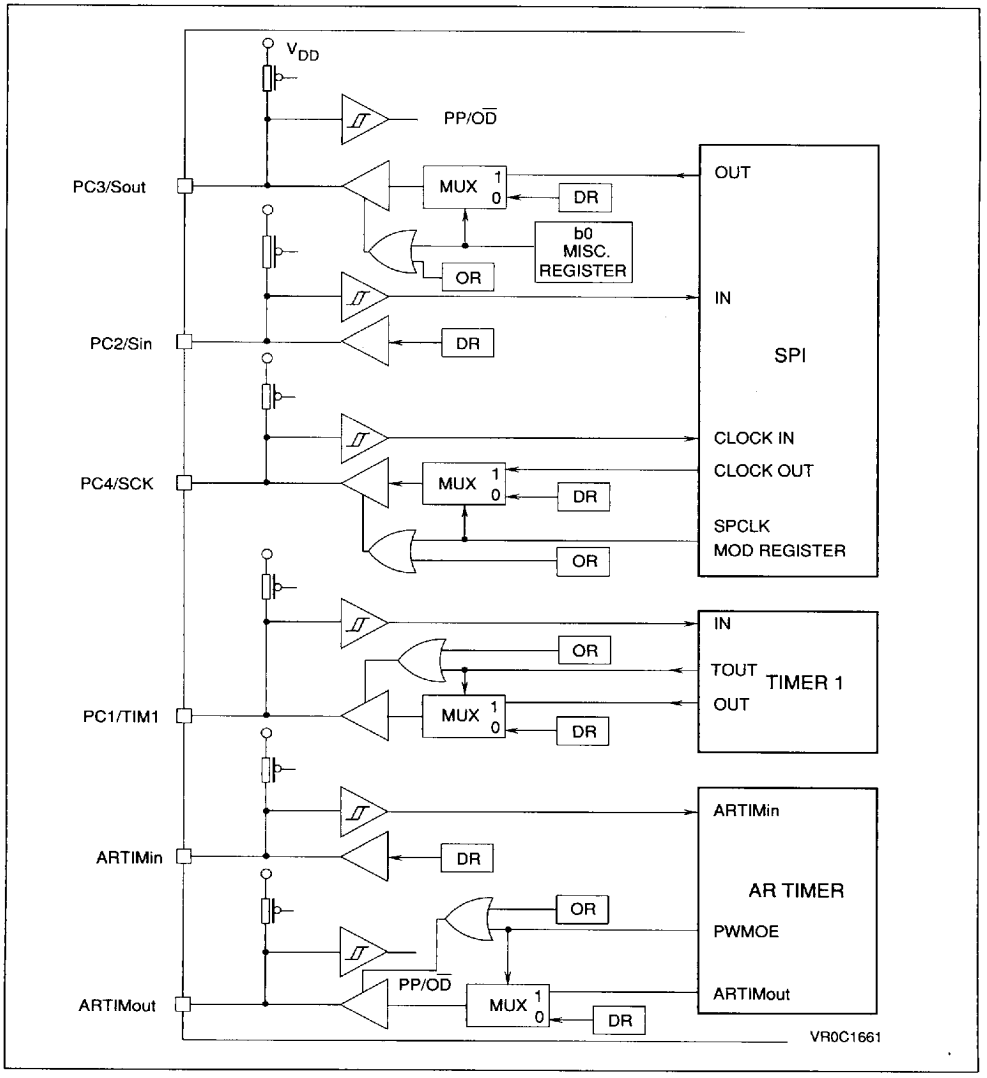
Mode	Available At[1]	Schematic
Input	PA0-PA7 PB0-PB7 PC0-PC4 Sin, SCK ARTIMin, TIM1	
Input with pull up	PA0-PA7 PB0-PB7 PC0-PC4 Sin, SCK ARTIMin, TIM1	
Input with pull up with interrupt	PA0-PA7 PB0-PB7 PC0-PC4 Sin [2], SCK[2] ARTIMin[2], TIM1[2]	
Analog Input	PA0-PA7 PC0-PC4	
Open drain output 5mA Open drain output 20mA	PA0-PA7 PC0-PC4 PB0-PB7	
Push-pull output 5mA Push-pull output 20mA	PA0-PA7 PC0-PC4 TIM1, Sout, SCK PB0-PB7 ARTIMout	

Notes:

1. Provided proper configuration.
2. This configuration is available but should be used with care.

INPUT/OUTPUT PORTS (Continued)

Figure 29. Peripheral Interface Configuration of SPI, Timer 1 and AR Timer



TIMERS

The ST6260B/65B offer two on-chip Timer peripherals named Timer 1 and Auto-reload Timer. Timer 1 consists of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and control logic that allows configuring the peripheral in three operating modes. The Auto-reload Timer is an 8-bit Timer with Auto-reload, Input Capture and Output Compare capabilities. 4 modes are available for PWM, PLL, time measurement and period measurement.

Timer 1

Figure 30 shows the Timer 1 block diagram. An external Timer pin is available for the user (ST6265B only). The content of the 8-bit counter can be read/written in the Timer/Counter register TCR which is addressed in the data space as a RAM location at addresses D3h. The state of the 7-bit prescaler is read in the PSC register at addresses D2h. The control logic device is managed in the TSCR1 register (addresses D4h) 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 TSCR1 is set to one. If the ETI (Enable Timer Interrupt) bit in the TSCR1 is also set to one an interrupt request, associated to interrupt vector #4, is generated. The interrupt service routine then should poll bit TMZ in TSCR1 to determine if the interrupt has been generated by Timer 1 or by the A/D Converter. The Timer 1 interrupt can be used to exit the MCU from the WAIT mode.

The Timer 1 Prescaler input can be the internal clock (after Oscillator Divider) divided by 12 or an external clock at the Timer I/O pin. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in TSCR1, the clock input of the timer/counter register is multiplexed to different sources. On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR1. This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC1 is connected to clock input of TCR1, and so on. The prescaler initialize bit PSI in the TSCR1 register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting. The prescaler can be given any value between 0 and 7Fh by writing to addresses D2h, if bit PSI in the TSCR1 register is set to one. The tap of the prescaler is selected using the PS2,PS1,PS0 bits in the control register. Figure 31 shows the Timer 1 working principle.

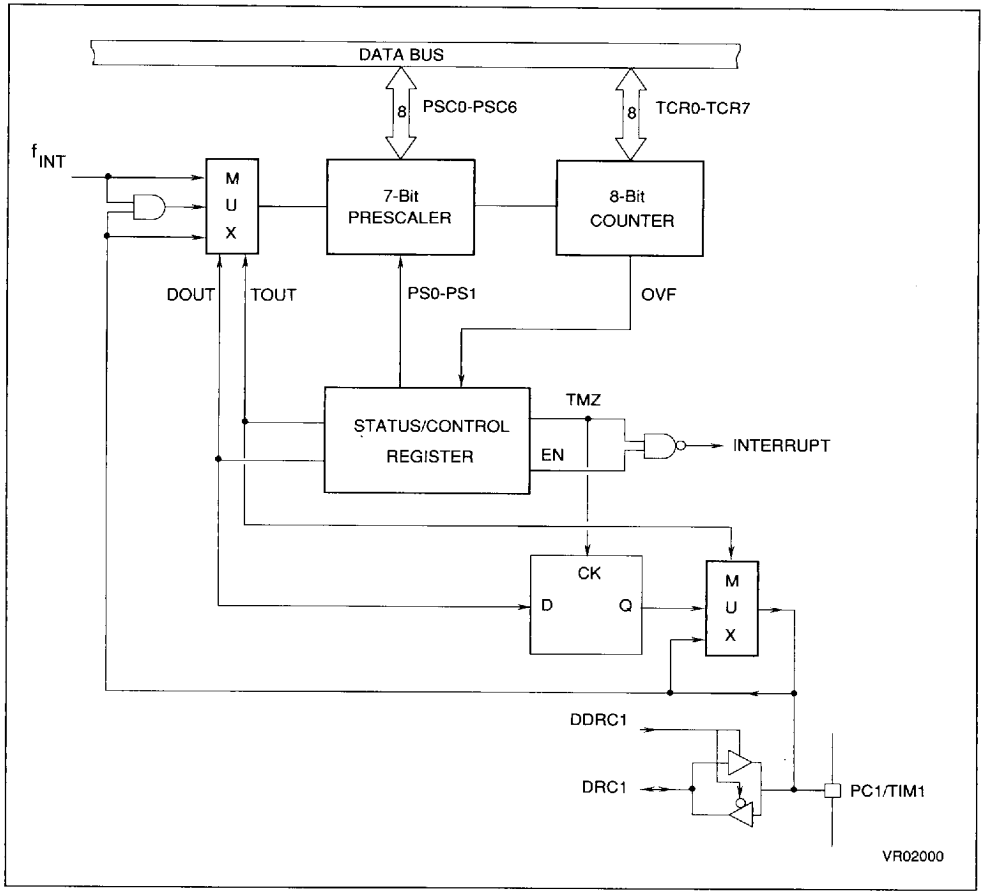
Timer 1 can be configured in 3 modes using the TOUT and DOUT bits of the TSCR1 register. These modes are Event counter, Gated or Output signal.

The internal Timer I/O can in addition be connected to either the PC1/TIM1 pin or bit DRC1 of the Port C Data Register depending on the configuration of bit DDRC1 of the Port C Data Direction Register. Table 9 summarize the modes of Timer 1.

- Event counter: The Prescaler is decremented at each rising edge of the Timer I/O. The Timer I/O is either the PC1/TIM1 pin or the DRC1 bit of the DRC register depending on DDRC1.
- Gated: The Timer 1 is decremented by the Timer clock (f_{INT} divided by 12) when the internal Timer I/O is held high. The Timer I/O is either pin PC1/TIM1 or the DDRC1 bit of register DDRC.
- Output signal: The PC1/TIM1 pin is connected to the DOUT latch and is configured as output regardless of DOUT and DDRC1 bits. The low to high transition of bit TMZ (when counter reaches 00h) is used to latch the data previously stored in DOUT and pass it to the PC1/TIM1 through the Timer I/O. This operating mode allows signal generation.

TIMERS (Continued)

Figure 30. Timer 1 Peripheral Block Diagram



TIMERS (Continued)

Timer 1 Interrupt

If the software controlled ETI (Enable Timer Interrupt) bit is set, when the counter decrements to zero, the TMZ bit in the TSCR register is set to one and an interrupt request associated to interrupt vector #4 is generated.

Since only one interrupt vector is available for both Timer 1 and the A/D Converter, the interrupt service routine should determine from which source the interrupt came by polling the TMZ bit and the EOC bit of the A/D Converter Control Register.

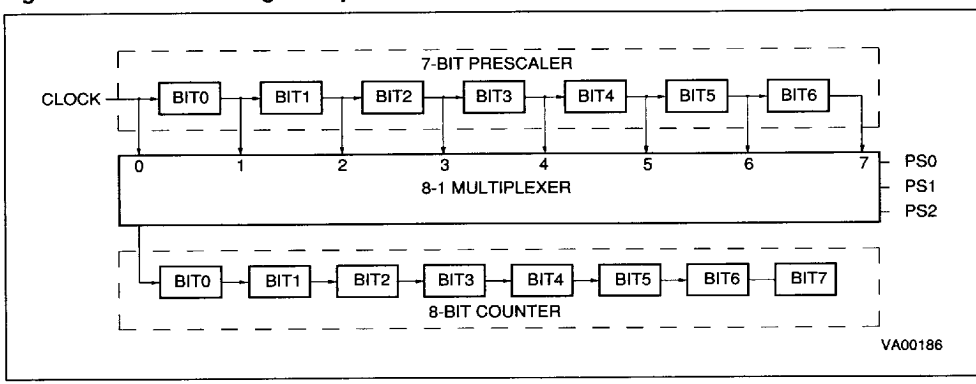
Notes:

TMZ is set when the counter reaches 00h; however, it may be set by writing 00h in the TCR1 register or setting bit 7 of the TSCR1 register. TMZ bit must be cleared by user software when servicing the Timer 1 interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR1 register is cleared which means that Timer 1 is stopped and the Timer 1 interrupt is disabled.

If the Timer 1 is programmed in output mode, DOUT bit is transferred to the TIM1 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.

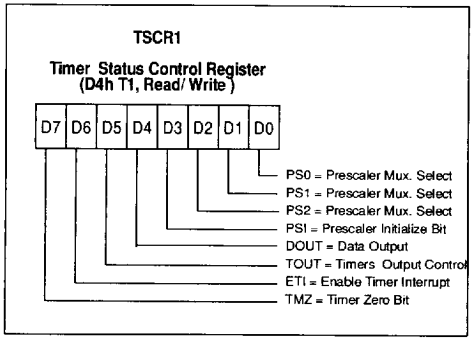
A write to the TCR1 register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR1 register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR1 and the PSC1 registers can be read accurately at any time.

Figure 31. Timer 1 Working Principle



TIMERS (Continued)

Figure 32. Timer Status Control Register



TMZ. Low-to-high transition indicates that the timer count register has decremented to zero. This bit must be cleared by user software before starting with a new count.

ETI. This bit, when set, enables the timer interrupt request (vector #4). If ETI="0" the timer interrupt is disabled. If ETI="1" and TMZ="1" an interrupt request is generated.

TOUT. When low, this bit selects an input mode for the Timer I/O pin. When high the output mode is selected.

DOUT. If Timer 1 is in Output mode, DOUT is the data sent to the PC1/TIM1 pin when TMZ goes high. DOUT enables discrimination between Event Counter and Gated modes if TOUT is low.

PSI. 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.

PS2, PS1, PS0. These bits select the division ratio of the prescaler register.

Table 9. Prescaler Division Factors

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

Figure 33. Timer Counter Register

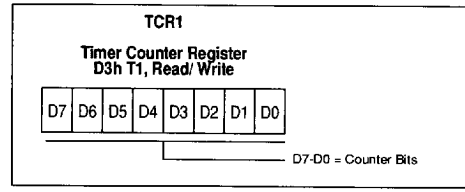


Figure 34. Prescaler Register

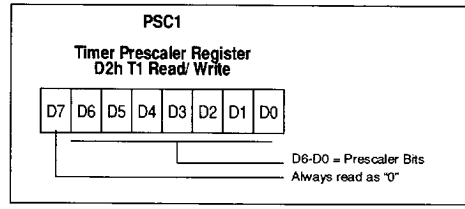


Table 10. Modes of Timer 1

TOUT	DOUT	DDRC1	Mode	Timer I/O
0	0	0	Event Counter	PC1/TIM1
0	0	1	Event Counter	DRC1
0	1	0	Gated	PC1/TIM1
0	1	1	Gated	DRC1
1	X	X	Output	PC1/TIM1

TIMERS (Continued)

Auto-reload Timer

The Auto-reload Timer (AR Timer) on-chip peripheral consists of an 8-bit timer/counter (AR COUNTER) with compare and capture/reload capabilities and a 7-bit prescaler with a clock multiplexer enabling the clock input to be selected as f_{INT}, $f_{INT}/3$ or external clock. One Mode Control Register (AR MODE), two Status Control Registers (ARSC0, ARSC1), an output pin (ARTIMout/PB7) and an input pin (ARTIMin/PB6) 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 with either an internal or an external clock. It also can be used to wake the MCU from STOP mode if used with an external clock provided at pin ARTIMin. A Load register allows the program to read and write the counter on the fly.

AR Timer Description

The AR COUNTER is an 8-bit up-counter incremented on the clock input rising edge. It is loaded from the ReLoad/Capture Register REL/CAP (address D9h) for auto-reload or capture operations as well as for initialization. Direct access to the AR COUNTER is not possible, however by reading/writing the Load Register AR LOAD (address DBh) it is possible to read/write the TC counter content.

The AR Timer input clock is either the internal clock (from Oscillator Divider), the internal clock divided by 3 or the ARTIMin pin. Selection between these clock sources is made through the AR Multiplexer by bits CC0-CC1 of Register ARSC1. The output of the AR Multiplexer feeds the AR Prescaler, ARPSC. ARPSC is a software programmable 7-bit prescaler. Programming of ARPSC is performed by the AR Prescaler Multiplexer AR MUX which selects one of the 8 available taps of the prescaler outputs under the control of PSC0-PSC2 in the AR Mode Control Register (address D5h). So the division factor of PSC prescaler can be set to 2^n (where n = 0, 1,..7).

The clock input to the ARTC counter is enabled by bit TEN (Timer Enable) in the AR Mode Control Register. When TEN is cleared to "0" the TC counter is stopped and the prescaler and counter contents are frozen. When the TEN bit is set to "1" the TC counter runs at the rate of the selected clock source. TC is cleared after system reset.

The ARTC counter can also be initialized by writing into the load register ARLR, which causes also the immediate copy of the value into the ARTC counter regardless of whether ARTC is running or not. Initialization of ARTC, in both ways, will also clear the ARPSC in order to start counting from a known state.

Each interrupt generated by the AR Timer operating modes is associated to interrupt vector #3.

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 output pin with minimum Core processing time used.

ARTC is a free running 8-bit counter fed by the ARPSC prescaler output and is incremented on every rising edge of the clock signal.

When a counter overflow occurs the ARTC counter is automatically reloaded with the contents of the Reload/Capture Register (REL/CAP, address D9h) while ARTIMout is set. When the counter reaches the value contained in the compare register ARCP, ARTIMout is reset.

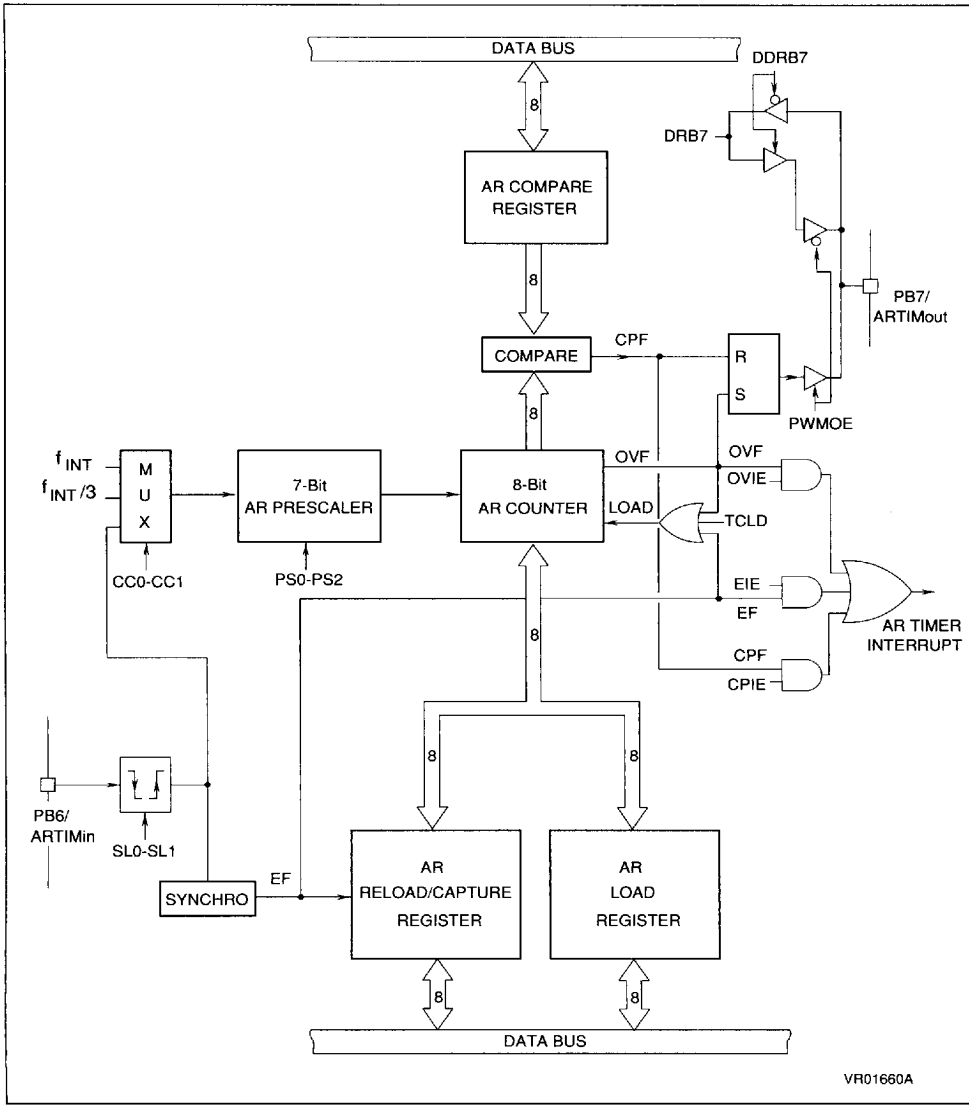
At overflow, the OVF flag of register ARSC0 is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OVIE in the Mode Control Register (ARMC, address D5h), is set to "1". OVF must be reset to zero by the user software.

When the counter reaches the compare value the CPF flag of register ARSC0 is set and a compare interrupt request is generated if the Compare Interrupt enable bit, CPIE, in the Mode Control Register (ARMC, address D5h), is set to one. The interrupt service routine may then adjust the PWM period by loading a new value into ARCP. CPF must be reset to zero by software.

The PWM signal is generated at ARTIMout (refer to block diagram) connected to the ARTIMout output pin. The frequency of this signal is controlled by the prescaler and by the auto-reload value present in the Reload/Capture register ARRC (address D9h). The duty cycle of the PWM signal is controlled by the Compare Register (ARCP, address DAh).

TIMERS (Continued)

Figure 35. AR Timer Block Diagram



TIMERS (Continued)

Note that the reload values will also affect the value and the resolution of the duty cycle of PWM output signal. To achieve a ARTIMout signal 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)]

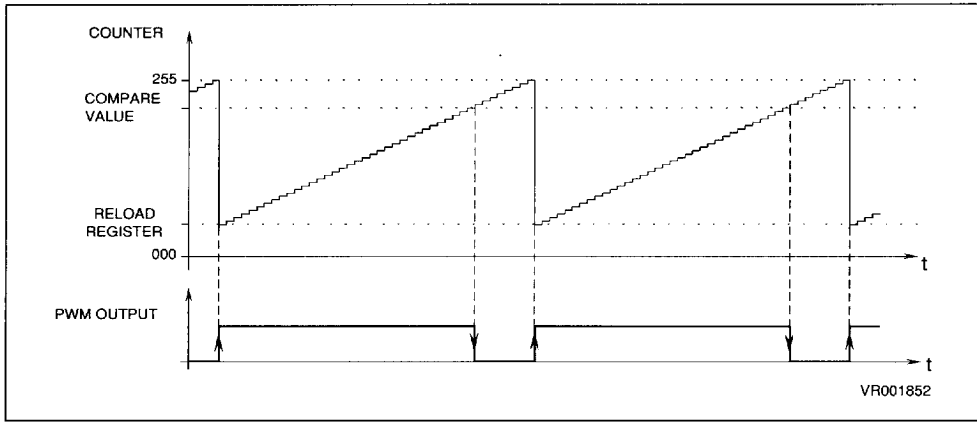
Where ARRC is the content of Reload/Capture register and the compare value loaded in the Compare Register, ARCP, must be in the range from (ARRC) to 255.

The initialization of the ARTC counter is made by writing into the ARRC register, then by setting the TCLD (Timer Load) and the TEN (Timer Clock Enable) bits in the Mode Control register ARMC.

The enable and the selection of clock sources are controlled by CC0, CC1, SL0 and SL1 bits in the Status Control Register ARSC1. The prescaler division ratio is selected by PS0, PS1 and PS2 bits in the ARSC1 Register.

In Auto-reload Mode any clock source can be selected: Internal Clock, Internal Clock divided by 3 or the signal at the ARTIMin input pin.

Figure 36. Auto-reload Timer PWM Function



TIMERS (Continued)

Capture Mode with PWM Generation. In this case, ARTC is a free running 8-bit counter fed by the PSC prescaler output. ARTC is incremented on every clock rising edge.

An 8-bit capture operation from ARTC counter to ARRC register is performed on every active edge at ARTIMin/PC Input 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 to one. The EF flag must be reset by software.

Each ARTC overflow sets ARTIMout, while a match between ARTC and ARCP (Compare Register) contents resets ARTIMout and sets the compare flag CPF and the compare interrupt request is generated if the related compare interrupt enable bit CPIE is set. A PWM signal is generated at ARTIMout. The CPF flag must be reset by software.

The frequency of this signal is controlled by the prescaler. The duty cycle is controlled by register ARCP from 0-255/256.

Initialization and reading of ARTC counter are made in the same way as in the auto-reload mode (see previous paragraph).

The enable and selection of clock sources is controlled by CC0, CC1 bits in the AR Status Control Register ARSC1.

The prescaler division ratio is selected by PS0, PS1 and PS2 bits in the ARSC1 Register.

In Capture mode the possible clock sources are the internal clock and the internal clock divided by 3; the external ARTIMin input pin should not be used.

Capture Mode with Reset of ARTC, ARPSC and PWM Generation. This mode is identical to the previous one, with the difference that a capture condition also resets the ARTC counter and ARPSC prescaler allowing easy measurement of the time between two captures (for input period measurement on ARTIMin pin).

Load on External Input. ARTC is a free running 8-bit counter fed by the ARPSC prescaler. TC is incremented on every clock rising edge.

Each ARTC overflow sets the ARTIMout. A match between ARTC and ARCP (Compare Register) contents resets the ARTIMout and sets the compare flag CPF and the compare interrupt request is generated if the related compare interrupt enable bit CPIE is set. A PWM signal is generated at ARTIMout. The CPF flag must be reset by software

The initialization of ARTC can be done in the same way 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 ARTC counter, whether ARTC 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 timer clock frequency should not be modified while ARTC is counting as the ARTC counter may take an unpredictable value. For example the multiplexer setting should not be modified while ARTC is counting.

Any loading of ARTC (by auto-reload, through ARLR, ARRC or by the Core) resets ARPSC at the same time.

Care should be taken when both the Capture interrupt and the Overflow interrupt are used. The capture and the 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) could be cleared simultaneusly without the interrupt being taken into account.

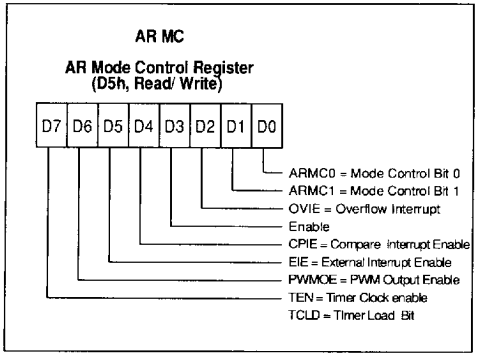
The solution consists in resetting the OVF flag by writing 03h in the ARSC0 register. The value of EF is not affected by this operation. If an interrupt has occured, it will be processed when the micros exits from the interrupt routine (the second interrupt is latched).

TIMERS (Continued)

AR Timer Registers

Mode Control Register ARMC. The AR Mode Control Register ARMC is used to program the different operation modes of AR Timer, to enable the clock of the Timer/Counter and to initialize it. It can be read and written by the Core and it is cleared to zero on system reset (AR Timer is disabled).

Figure 37. AR Mode Control Register



TCLD. This bit, when set to one, will cause the contents of ARRC register to be loaded into the ARTC counter and the contents of ARPSC register are cleared in order to initialize the timer before starting to count. This bit is write only and any attempt to read it will show a logical zero.

TEN. This bit, when set to one, will allow the timer to count. When cleared to zero it will stop the timer and freeze the ARPSC and ARTSC values.

PWMOE. This bit, when set, enables the PWM output to be carried on ARTIMout output pin. When cleared to zero the PWM output is disabled.

EIE. This bit, when set, enables the external interrupt request. If EIE = "0" the external interrupt request is masked. If EIE = "1" and the related flag EF in the ARSC0 register is also set an interrupt request is generated.

CPIE. This bit, when set, enables the compare interrupt request. If CPIE = "0" the compare interrupt request is masked. If CPIE = "1" and the related flag CPF into the ARSC0 register is also set an interrupt request is generated.

OVIE. This bit, when set, enables the overflow interrupt request. If OVIE = "0" the compare interrupt request is masked. If OVIE = "1" and the related flag OVF into the ARSC0 register is also set, an interrupt request is generated.

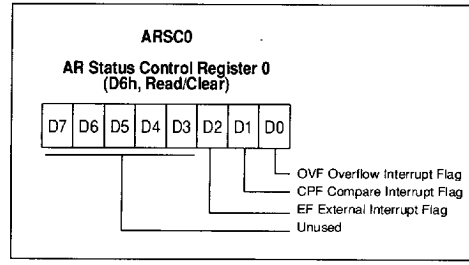
ARMC1, ARMC0. These are the operation mode control bits. The following bit combination will select the different operating modes:

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 provide the AR Timer status information bits and also allows the programming of clock sources, active edge and prescaler multiplexer programming.

ARSC0 register bits 0,1 and 2 contains the interrupt flags of the AR Timer. These bits are read normally. Each one can be reset by writing a zero. Writing a one does not affect the bit value.

Figure 38. AR Status Control Register 0



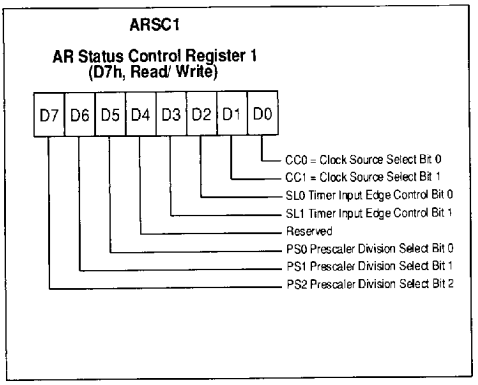
EF. This bit is set to one by any active edge at the external ARTIMin input pin. The flag is cleared by writing a zero in the EF bit.

CPF. This bit is set to one if the contents of ARTC counter and ARCP register are equal. The flag is cleared by writing a zero into CPF bit.

OVF. This bit is set to one by a transition of TC counter from FFh to 00h. The flag is cleared by writing a zero into OVF bit.

TIMERS (Continued)

Figure 39. AR Status Control Register 1



PS2-PS0. These bits control the AR Prescaler division ratio. The prescaler itself is not affected by these bits. The AR PSC division is listed in the following Table 11:

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

D4. Reserved. Must be kept to 0.

SL1-SL0. These bits control the edge function on AR Timer input pin for external synchronization. If bit SL0 is cleared to zero the edge detection is disabled, if set to one the edge detection is enabled. If bit SL1 is cleared to zero the AR Timer input pin is rising edge sensitive, if set to one it is falling edge sensitive.

SL1	SL0	Edge Detection
X	0	Disabled
0	1	Rising Edge
1	1	Falling Edge

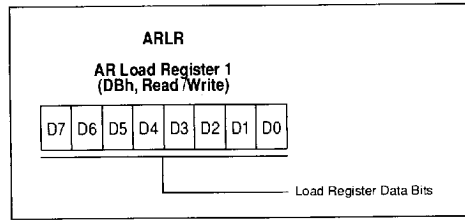
CC1-CC0. These bits select the clock source for the AR Timer through the AR Multiplexer. The programming of the clock sources is explained in the following Table 12:

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 "on the fly" the ARTC counter register, while it is counting. ARLR register is not affected by system reset.

Figure 40. AR Load Register

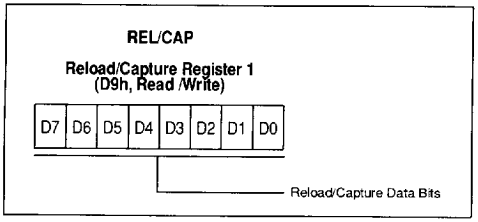


D7-D0. These are the load register data bits.

TIMERS (Continued)

AR Reload/Capture Register. The ARRC reload/capture register is used to hold the auto-reload value that is automatically loaded into ARTC counter from ARRC when overflow occurs.

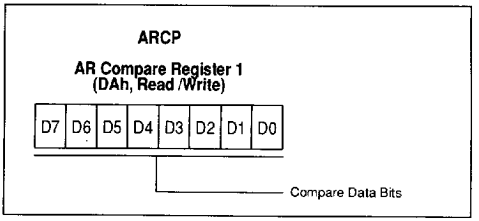
Figure 41. AR Reload/Capture



D7-D0. These are the Reload/Capture register data bits.

AR Compare Register. The CP compare register is used to hold the compare value to perform the compare function with TC counter.

Figure 42. AR Compare Register



D7-D0. These are the Compare register data bits.

DIGITAL WATCHDOG

The digital Watchdog of the ST6260B/65B device consists of a down counter that can be used to provide a controlled recovery from a software upset.

The Watchdog generates a system reset when the counter passes 00h. User software can prevent the reset by reloading the counter. User software should therefore be written in such a way that the counter is regularly reloaded as long as the software runs correctly. In the case of software upset (e.g. infinite loop or power supply fail), user software should not reload the counter so it will pass 00h and reset the MCU.

The Watchdog activation (hardware or software) is user selectable by mask option. If the hardware option is selected the Watchdog is automatically initialized after reset so that this function does not need to be activated by the user program.

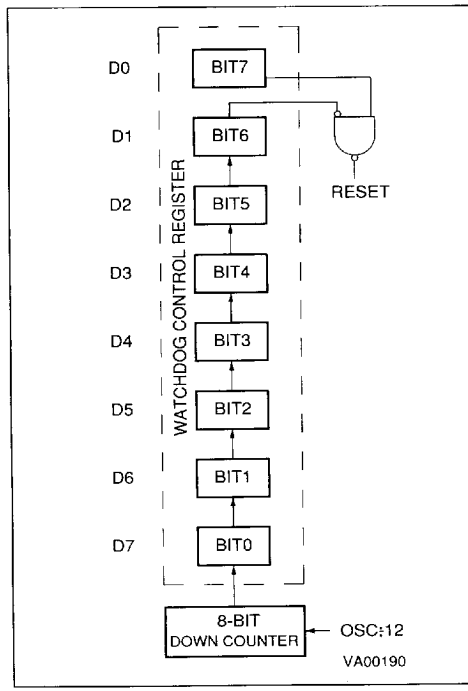
The Watchdog uses one data space register (DWDR location D8h). The Watchdog register is set to FEh on reset and counts down when activated. The Watchdog time can be adjusted through the value reloaded into the DWDR register. Only the 6 MSbits are significant. This gives the possibility to generate a reset in a time between 3072 to 196608 clock cycles in 64 possible steps (with a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded before bits 2-7 decrement from all zeros to all ones.

The actual watchdog behaviour is controlled by two mask options: the WATCHDOG ACTIVATION and the EXTERNAL STOP MODE CONTROL (see Table 13).

The WATCHDOG ACTIVATION can be software (it is launched by software) or hardware (it automatically starts counting down after reset). When the software activation is selected, the watchdog can be launched by setting to 1 bit 0 of the Digital Watchdog Register after bit SR of this register has been set to 1. Once activated, the watchdog cannot be stopped by software: a full external reset is mandatory.

When the EXTERNAL STOP MODE CONTROL is disabled, the STOP instruction is inhibited as soon as the watchdog is active. A WAIT instruction is processed instead and the watchdog continues to countdown. When the EXTERNAL STOP MODE CONTROL is enabled, the NMI pin allows, in addition to the interrupt generation, to control the execution of the STOP instruction. It is inhibited when

Figure 43. Watchdog Working Principle



NMI is low (a WAIT instruction is processed instead). When NMI is high, a STOP instruction freezes the watchdog counter before entering the STOP mode. When the micro exits from the STOP mode (for example, when an NMI interrupt is generated), the watchdog resumes activity. When the EXTERNAL STOP MODE CONTROL is enabled, port PB0 is in addition forced as open drain output.

Note:

When the software activation is selected and the watchdog is not activated, the 7 MSbits of the counter can be used to perform timer functions. Care must be taken as the Watchdog bits are in reverse order.

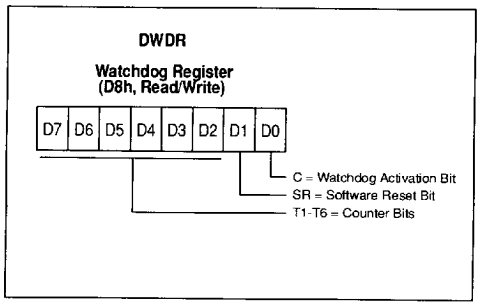
Bit 1 of the Watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero.

DIGITAL WATCHDOG (Continued)

Table 13. Watchdog Activation and External Stop Mode Control

Activation	External Stop Mode Control	NMI	STOP Mode Status
Hardware	Disabled	X	No STOP mode
	Enabled	0	No STOP mode
		1	STOP mode available
Software	Disabled	X	STOP mode available if watchdog not active
	Enabled	0	STOP mode available if watchdog not active
		1	STOP mode available

Figure 44. Digital Watchdog Register



C. This is the Watchdog activation bit. If hardware option is selected, it 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 C to 1. It can then be cleared only by a system reset and is not affected by the STOP Mode Control option. When C is kept low the counter can be used as a 7-bit timer.

When cleared to zero it allows the use of the counter as a 7-bit timer. This bit is cleared on reset.

SR. This bit is set to one during the reset and will generate a software reset if cleared to zero. When C = "0" (watchdog disabled) it is the MSB of the 7-bit timer.

T1-T6. These are the watchdog counter bits. It must be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter.

Application note:

The watchdog has an important role in the high noise immunity of the ST62 devices. It should therefore be used wherever possible. The watchdog related options must be selected as a tradeoff between application security and STOP mode availability. When the STOP mode is not required, the hardware activation with External STOP Mode Control disabled should be preferred as it provides maximum security, especially during power on. When the STOP mode is required, the hardware activation with External STOP Mode Control enabled should be chosen. NMI should be high by default to allow the STOP mode to be entered when no specific action is required. When NMI goes low, processing is required, so security becomes important: it is provided by the automatically started watchdog.

Connecting pin NMI to PB0 (see Figure 46) in addition allows a software control of the level at the NMI pin. With PB0 being forced as open drain output, it can then be used to maintain NMI low for as long as maximum security must be guaranteed or to avoid key bounces or noise. When no more processing is required, PB0 must be released and a STOP instruction can be executed for the device returns to the low power consumption mode (provided the NMI signal is high).

DIGITAL WATCHDOG (Continued)

Software activation should be used only when the watchdog counter must be used as timer. To ensure the watchdog has not been unexpectantly turned on, the following instructions should be executed within the first 27 instructions.

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

These instructions test the C bit and reset it (i.e. deactivate the watchdog) if relevant (i.e. if the watchdog is active) by performing a software reset.

With all modes, a minimum of 28 instructions are executed after activation before the watchdog can generate a reset. Consequently, user software must reload the watchdog counter within the first 27 instructions following watchdog activation (software mode) or the first 27 instructions executed after a reset (hardware activation).

Figure 46. Typical circuit to be used when the EXTERNAL STOP MODE CONTROL is enabled

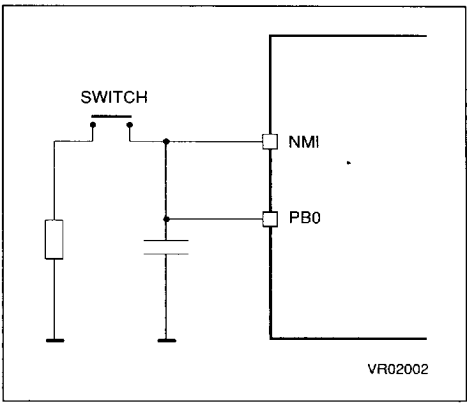
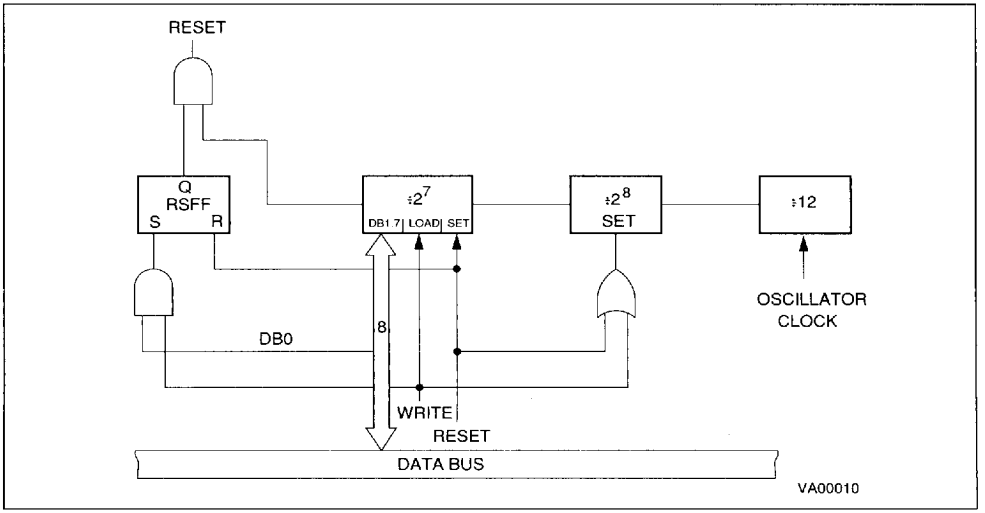


Figure 45. Digital Watchdog Block Diagram



8-BIT A/D CONVERTER

The A/D converter of ST6260B/65B device is an 8-bit analog to digital converter with up to 7 (ST6260B) and up to 13 (ST6265B) analog inputs (as alternate functions of I/O lines PA0-PA7, PC0-PC4) offering 8-bit resolution with total accuracy ±2 LSB and a typical conversion time of 70µs (clock frequency of 8MHz).

The A/D peripheral 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, the A/D converter accuracy is decreased.

The selection of the pin signal that has to be converted is done by configuring the related I/O line as analog input through the I/O ports option and data registers (refer to I/O ports description for additional information). Only one I/O line must be configured as analog input at a time. The user must avoid the situation in which more than one I/O pin is selected to be analog input to avoid malfunction of the ST62xx.

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

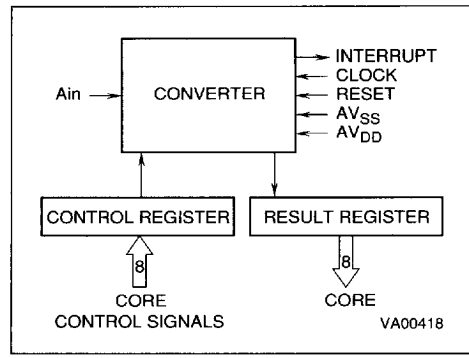
A conversion is started by writing a "1" to the Start bit (STA) in the ADC control register. This automatically clears (resets to "0") the End Of Conversion Bit (EOC). When a conversion has been finished this EOC bit is automatically set to "1" in order to flag that conversion is complete and that the 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 continually being scanned so that if the user sets it to "1" while a previous conversion is in progress then a new conversion is started before the previous one has been completed.The start bit (STA) is a write only bit, any attempt to read it will show a logical "0".

The A/D converter has a maskable interrupt associated to the end of conversion. This interrupt is associated to the 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. That is achieved when the PDS bit in the ADC control register is cleared to "0". If PDS="1", the A/D is supplied and enabled for conversion. This bit must be set at least one instruction before the beginning of the conversion to allow the stabilization of the

Figure 47. A/D Converter Block Diagram



A/D converter. *This action is needed also before entering the WAIT instruction as the A/D comparator is not automatically disabled by the WAIT mode*

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

Notes:

The ST62xx A/D converter does not feature a sample and hold. The analog voltage to be measured should therefore be stable during the conversion time. Variation should not exceed ±1/2 LSB for the best accuracy in measurement. A low pass filter can be used at the analog input pins to reduce input voltage variation during the conversion.

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

$$6.5\ \mu s = 9 \times Cad \times ASI$$

(capacitor loaded over 99.9%), ie 30 kΩ including 50% guardland. ASI can be higher if Cad has been loaded for a longer time by adding instructions before conversion start (adding more than 26 CPU cycles is meaningless).

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 needed. This is because such switching will affect the supply voltages which are used for comparisons.

8-BIT A/D CONVERTER (Continued)

The accuracy of the conversion depends on the quality of the power supply voltages (V_{DD} and V_{SS}). The user must specially take care of applying regulated reference voltage on the V_{DD} and V_{SS} pins (the variation of the power supply voltage must be inferior to 5V/ms). This implies in particular that a suitable decoupling capacitor is used at V_{DD}.

The converter can resolve the input voltage with a resolution of:

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

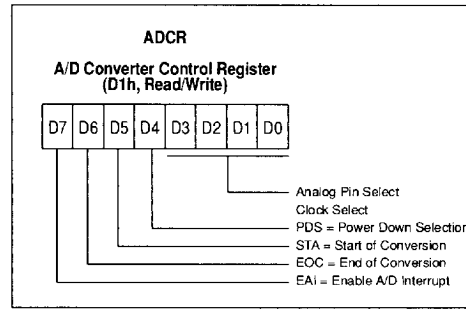
The Input voltage (Ain) which has to be converted must be constant for 1µs before conversion and remain constant during the conversion.

The resolution of the conversion can be improved if the power supply voltage (V_{DD}) of the microcontroller becomes lower.

In order to optimize the resolution of the conversion, the user can configure the microcontroller in the WAIT mode because this mode allows the minimization of the noise disturbances and the variations of the power supply voltages due to the switching of the outputs. Nevertheless, it must be take care of executing the WAIT instruction as soon as possible after the beginning of the conversion because the execution of the WAIT instruction may provide a small variation of the V_{DD} voltage (the negative effect of this variation is minimized at the beginning of the conversion because the latter is less sensitive than the end of the conversion when the less significant bits are determined).

The best configuration from a accuracy point of view is the WAIT mode with the Timer and AR Timer stopped. Indeed, only the ADC peripheral and the oscillator are still working. The MCU has to be wake-up from the WAIT mode by the interrupt of the ADC peripheral at the end of the conversion. It must be noticed that the wake-up of the microcontroller could be done also with the interrupt of the TIMER, but in this case, the Timer is working and some noise could disturb the converter in terms of accuracy.

Figure 48. A/D Converter Control Register



EAI. If this bit is set to one the A/D interrupt (vector #4) is enabled, when EAI=0 the interrupt is disabled.

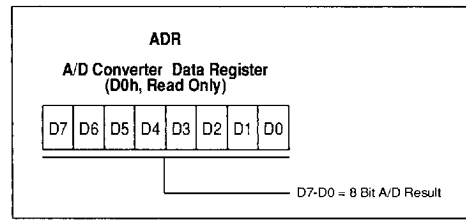
EOC. *Read Only*;This read only bit indicates when a conversion has been completed. This bit is automatically reset to zero when the STA bit is written. If the user is using the interrupt option then this bit can be used as an interrupt pending bit. Data in the data conversion register are valid only when this bit is set to one.

STA. *Write Only*; Writing a "1" in this bit will start a conversion on the selected channel and automatically reset to zero 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.

PDS. This bit activates the A/D converter if set to "1". Writing a zero into this bit will put the ADC in power down mode (idle mode).

D3-D0. Not used

Figure 49. A/D Converter Data Register



SERIAL PERIPHERAL INTERFACE SPI

The ST6260B/65B SPI is an optimized synchronous serial interface with programmable transmission modes and master/slave capabilities supporting a wide range of industry standard SPI specifications. The ST6260B/65B SPI is controlled by simple user software to perform serial data exchange with low-cost external memory or serially controlled peripherals for display or driving motors or relays. The peripheral is composed of an 8-bit data/shift register DSR (address E0h), by a Divide register DIV (address E1h) and by a mode control register MOD (address E2h).

The SPI may be used as either a Master or a Slave Unit. The Master is defined by the synchronous serial clock line SCK being supplied by the MCU, while the Slave mode accepts external data with the SCK clock externally supplied. For the Master mode of the SPI, SCK is internally generated with a frequency derived from a programmable division ratio of the Oscillator Clock divided by 13. Input may also be disabled in Master mode for data security.

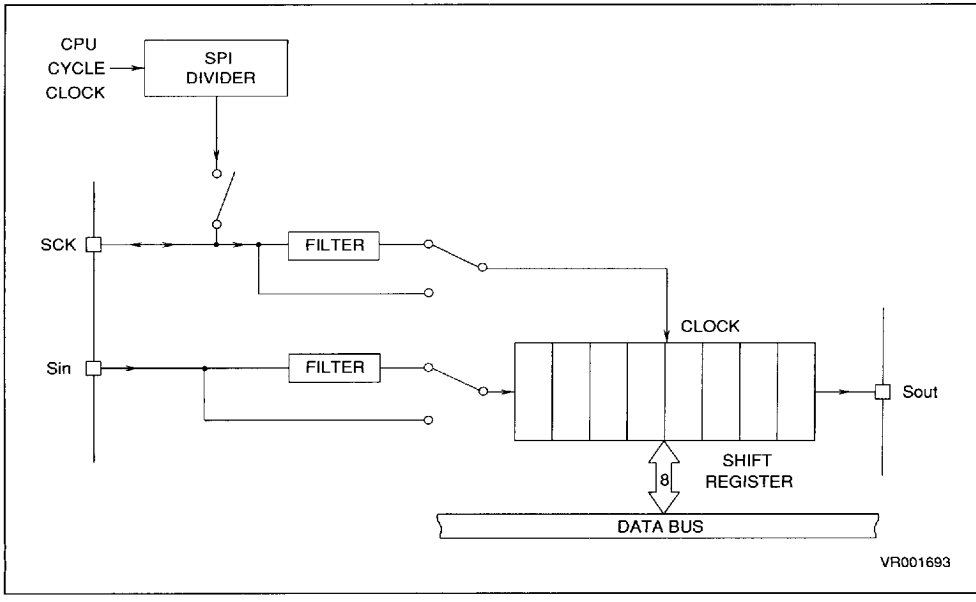
For maximum versatility the SPI can be programmed to sample the data on either the rising or the falling edge of SCK, and with or without a phase shift.

The Sin, Sout and SCK (SPI Data in, Data out and Clock signals respectively) signals are connected, as alternate functions, to I/O pins PC2-PC4. PC2 is connected with the SPI Serial Data Input Sin, PC3 is connected with the SPI Serial Data Output Sout and PC4 is connected with the SPI Clock Input/Output SCK.

For serial input operation PC2/Sin must be configured as input. For serial output operation, PC3/Sout alternate function is selected by programming Bit 0 of Miscellaneous Register (address DDh); writing a zero will set the pin as PC3 I/O line while writing a one will select the SPI Sout functionality. The serial clock Input mode is selected if the PC4 port pin is programmed in input mode and bit SPCLK is cleared. The output mode is selected if PC4 is programmed in output mode and SPCLK is set to 1.

An interrupt request can be associated to the end of transmission. This request is associated to interrupt vector #2 and can be masked by programming bit SPIE of the SPI MOD register. As the SPI interrupt is "ORed" with Port C interrupt source, an interrupt flag bit is available in the DIV register allowing the discrimination of the interrupt request.

Figure 50. SPI Block Diagram



SERIAL PERIPHERAL INTERFACE SPI (Continued)

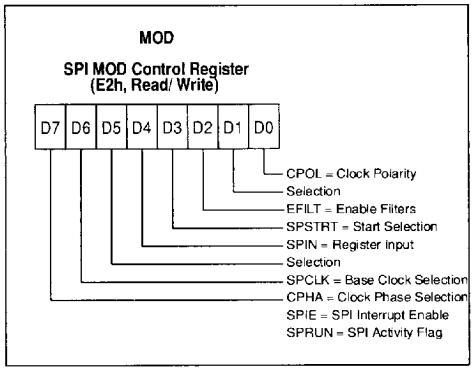
SPI Registers

SPI MODE (SPIMOD) Control Register

The MOD control register defines and controls the transmission modes and characteristics.

This register is read/write and all bits are cleared at reset. The configuration of SPSTRT = 1 and SPIN = 1 is not allowed and must be avoided.

Figure 51. SPI MOD Control Register



SPRUN. This bit is the SPI activity flag. This can be used in either read or write modes; it is automatically cleared by the SPI at the end of a transmission and generates an interrupt request (providing that the SPIE Interrupt Enable bit is set). The Core can stop the running transmission 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 the start bit, in conjunction with the SPSTRT bit, when an external signal is present on the Sin pin.

SPIE. This bit is the SPI Interrupt Enable bit. If this bit is set to one the SPI interrupt (vector #2) is enabled, when SPIE = "0" the interrupt is disabled.

CPHA. 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.

SPCLK. 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.

SPIN. This bit enables the transfer of the data input to the Shift Register in received mode. If this bit is cleared to zero the Shift Register input is 0. If this bit is set to one the Shift Register input corresponds to the input signal present on the Sin pin.

SPSTRT. This bit selects the transmission start mode. If this bit is cleared to zero the internal start condition occurs as soon as the SPRUN bit is enabled (set to one). If this bit is set to one, 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. After the transmission has been started, it will continue even if the Sin signal is reset.

EFILT. 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.

CPOL. 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.

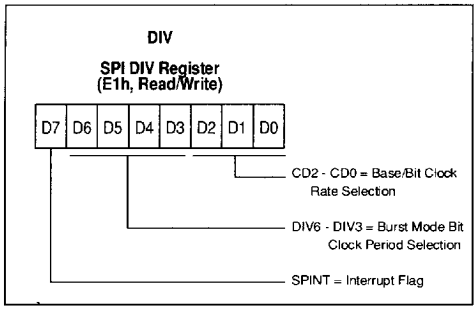
SERIAL PERIPHERAL INTERFACE SPI (Continued)

SPI Divide (SPIDIV) Register

The SPIDIV register defines the SPI transmission rate and frame format. It also contains the interrupt flag bit.

Bits CD0-CD2, DIV3-DIV6 are read/write while SPINT can be read and cleared only. Write access is not allowed if the SPRUN bit of Mode Control register is set to one. All bits are cleared at reset.

Figure 52. SPI DIV Register



SPINT/DIV7. This is the SPI interrupt flag bit. It is automatically set to one by the SPI at the end of a transmission and an interrupt request can be generated in accordance with the state of the interrupt mask bit into the MOD control register. This bit is read only and has to be cleared by the user software at the end of the interrupt service routine.

DIV6-DIV3. These bits define the number of shift register bits that are transmitted in a transmission frame. The available selections are listed in Table 14. The normal setting is 8 bits.

CD2-CD0. These bits define the division ratio between the core clock (f_{INT} divided by 13) and clock supplied to the Shift Register in Master mode.

SPI Data Shift (DSR) Register

SPIDSR is the SPI data shift register.

The Shift Register transmits and receives the Most Significant bit as the first bit.

Table 14. 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

Table 15. 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

SPIDSR is read/write, however write access is not allowed if the SPRUN bit of Mode Control register is set to one.

DSR7-DSR0. These are the SPI shift register data bits.

Data is sampled into DSR on the SCK edge determined by the CPOL and CPHA bits. The affect of these setting is shown in the following diagrams.

Figure 53. SPI Data/Shift Register

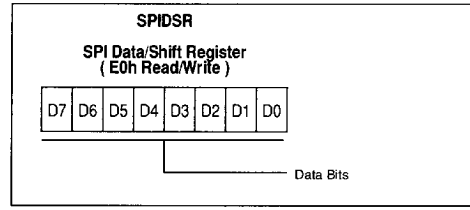
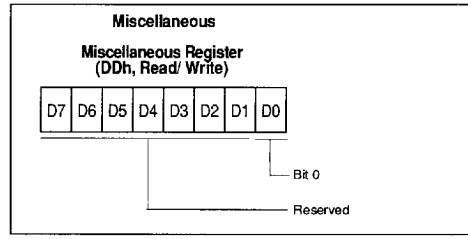


Figure 54. Miscellaneous Register



Bit 0. This bit, when set, selects pin PC3/Sout as the SPI output line. When this bit is cleared to zero, PC3/Sout acts as a standard I/O line.

SERIAL PERIPHERAL INTERFACE (Continued)

SPI Timing Diagrams

Figure 55. CPOL = 0 Clock Polarity Normal, CPHA = 0 Phase Selection Normal

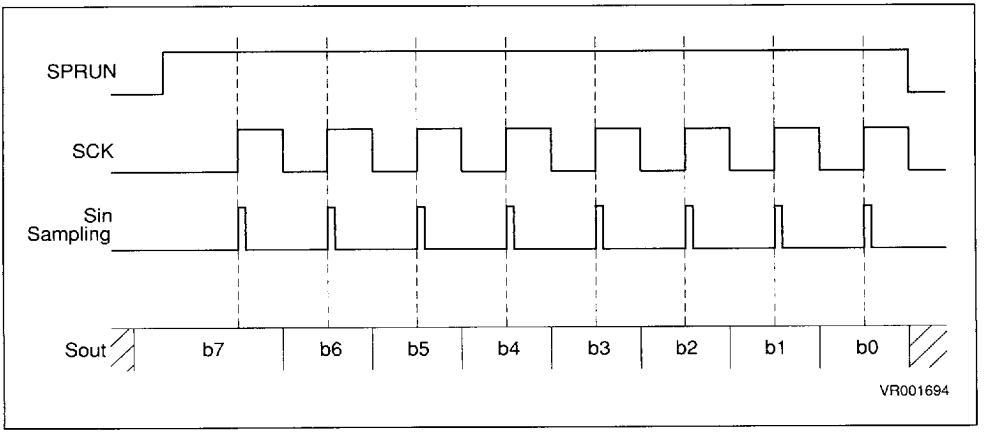


Figure 56. CPOL = 1 Clock Polarity Inverted, CPHA = 0 Phase Selection Normal

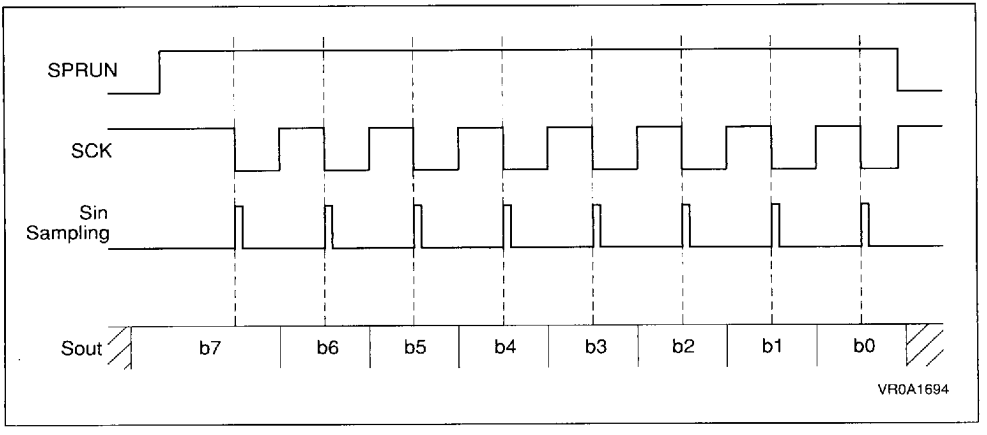


Figure 57. CPOL = 0 Clock Polarity Normal, CPHA = 1 Phase Selection Shifted

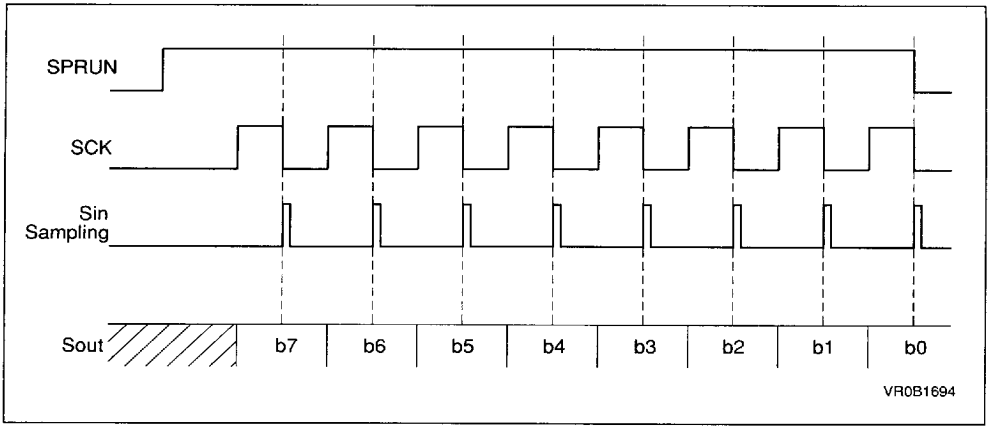
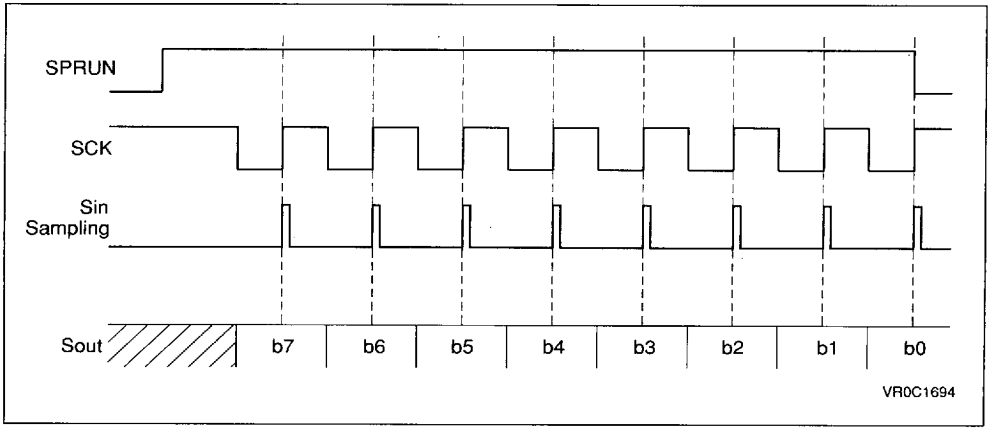


Figure 58. CPOL =1 Clock Polarity Inverted CPHA = 1 Phase Selection Shifted



SOFTWARE DESCRIPTION

The ST62xx 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 ST62xx 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.

Addressing Modes

The ST62xx core has nine addressing modes which are described in the following paragraphs. The ST62xx core uses three different address spaces : 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 that is processed by the instruction is stored in the location that 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) that 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 that 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 that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that 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.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST62xx core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types:load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within 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 16. 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

SOFTWARE DESCRIPTION (Continued)

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 17. 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	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	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 Registers
#. Immediate data (stored in ROM memory)
rr. Data space register

Δ. Affected
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

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 18. 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
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 19. 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 Affected

Table 20. 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 and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ. Affected
*. Not Affected

Table 21. 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

SOFTWARE DESCRIPTION (Continued)

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

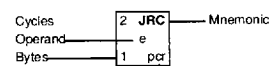
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 pcr	4 LD a,(y) 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	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 pcr	4 LD a,rr 2 dir	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 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 1 inh	2 JRC e 1 pcr	4 CP a,(y) 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	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 pcr	4 CP a,rr 2 dir	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	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 pcr	4 ADD a,(y) 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	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 pcr	4 ADD a,rr 2 dir	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	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 pcr	4 INC (y) 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	#	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 pcr	4 INC rr 2 dir	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	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 pcr	4 LD (y),a 1 ind	8 1000
9 1001	2 JRNZ 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	#	2 JRNZ 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 pcr	4 LD rr,a 2 dir	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	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 RLC a 1 inh	2 JRC e 1 pcr	4 AND a,(y) 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	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 pcr	4 AND a,rr 2 dir	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	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 pcr	4 SUB a,(y) 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	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 pcr	4 SUB a,rr 2 dir	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	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 pcr	4 DEC (y) 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	#	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 pcr	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



ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

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

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

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

$Tj = T_A + PD \times RthJA$

Where : T_A = Ambient Temperature.

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

PD = Pint + Pport.

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

Pport = Port power dissipation (determinated 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
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}	10	mA
I_{INJ+}	Pin Injection current (positive), All I/O, V_{DD} = 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	mA
IV_{SS}	Total Current out of V_{SS} (sink)	50 (2)	mA
Tj	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 guaranteed 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 50 mA. The total current through port C may not exceed 50 mA. If the application is designed with care and observing the limits stated above, total current may reach 100 mA.

THERMAL CHARACTERISTIC

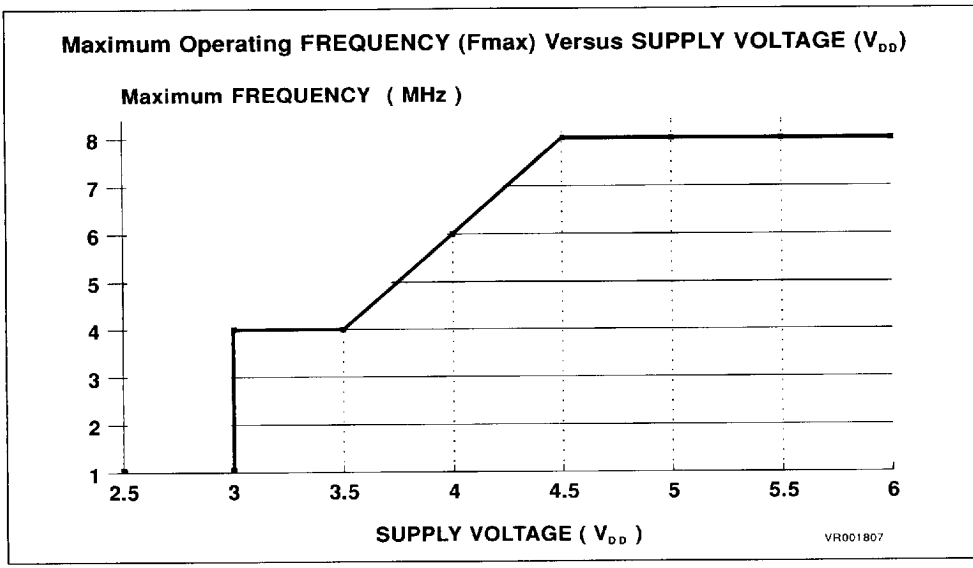
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP28			55	°C/W
		PDIP20			60	
		PSO28			75	
		PSO20			80	

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	f_{OSC}= 4MHz f_{INT} = 4MHz	3.0		6.0	V
		f_{OSC} = 8MHz f_{INT} = 8MHz	4.5		6.0	V
f_{INT}	Internal Frequency (3)	V_{DD} = 3V V_{DD} = 4.5V	0 0		4.0 8.0	MHz MHz
I_{INJ+}	Pin Injection Current (positive) Digital Input (1) Analog Inputs(2)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative) Digital Input (1) Analog Inputs	V_{DD} = 4.5 to 5.5V			-5	mA

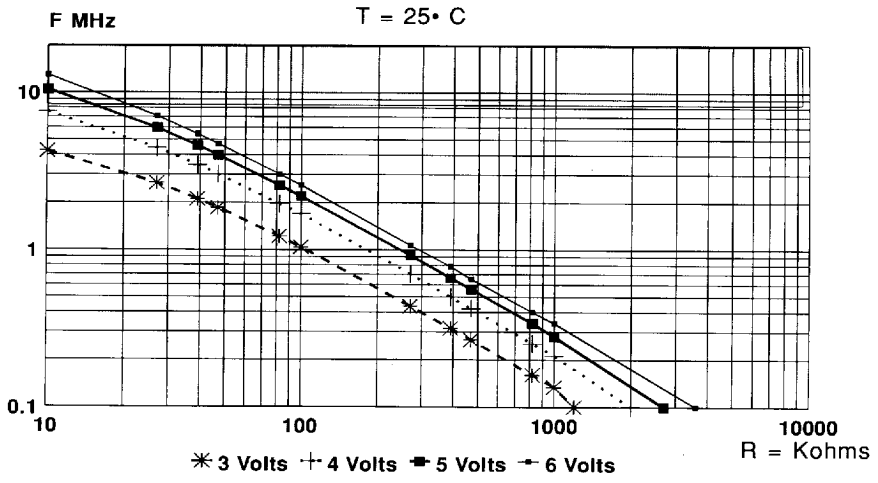
Notes :

1. A current of ± 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device. For a positive current injected into one pin, a part of this current (~ 10%) can be expected to flow from the neighbouring pins.
2. If a total current of +1 mA is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 1mA, all the resulting conversions are shifted by +1 LSB. If a total positive current is flowing into the single analog channel or if the total current flowing into all the analog inputs is of 5mA, all the resulting conversions are shifted by +2 LSB.
3. An internal frequency above 1MHz is recommended for reliable A/D results.



The shaded area is outside the ST6260B/65B operating range, device functionality is not guaranteed.
The striped area is guaranteed only with the LOW VOLTAGE option.

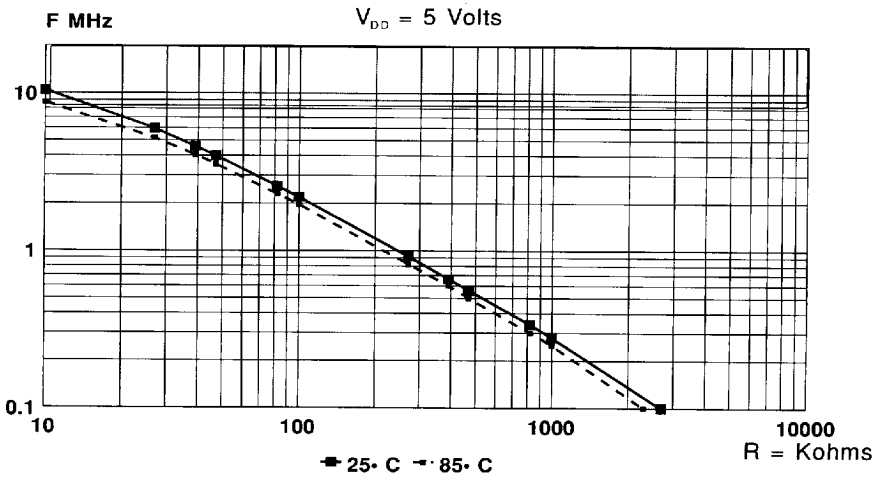
RC Oscillator. fosc Frequency versus RNET (Typical Values)



The shaded area is outside ST62xxB operating range, device functionality is not guaranteed

VR01935

RC Oscillator. fosc Frequency versus RNET (Typical Values)



The shaded area is outside ST62xxB operating range, device functionality is not guaranteed

VR01935A

SGS-THOMSON MICROELECTRONICS

DC ELECTRICAL 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}=3.0V 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}=3.0V 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}=3.0V 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	µA
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

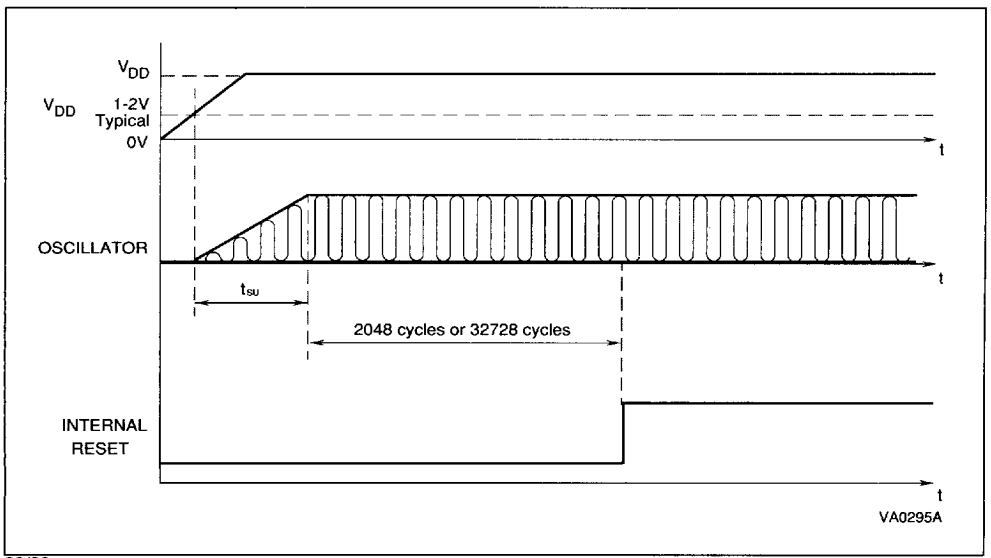
AC ELECTRICAL CHARACTERISTICS

(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

Note:
1. Period for which V_{DD} has to be connected at 0V to allow internal Reset function at next power-up.
2. See Figure 59. This value is highly dependent on the Ceramic Resonator or Quartz Crystal used in the application.

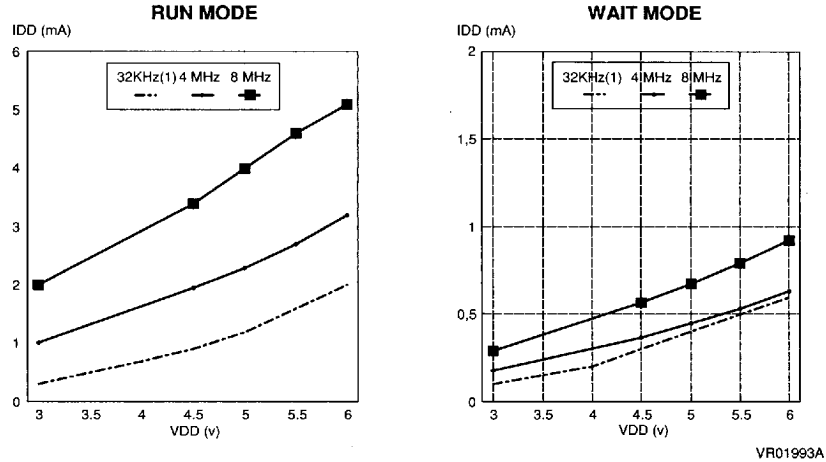
Figure 59. Power On Reset



ELECTRICAL CHARACTERISTICS (Continued)

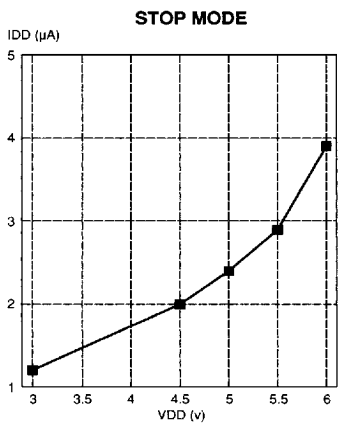
CURRENT CONSUMPTION

IDD Current Versus Supply Voltage
Typical Values (T_A : +85°C)



Note (1) : Using the network described in the Application Note AN673

IDD Current Versus Supply Voltage
Typical Values(T_A : +85°C)



I/O PORT 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	I/O Pins			0.3x V_{DD}	V
V_{IH}	Input High Level Voltage	I/O Pins	0.7x V_{DD}			V
V_{OL}	Low Level Output Voltage	V_{DD}= 5.0V I_{OL}= 10µA , All I/O Pins I_{OL}= 5mA , Standard I/O I_{OL}= 10mA , Port B I_{OL}= 20mA , Port B			0.1 0.8 0.8 1.3	V
V_{OH}	High Level Output Voltage	I_{OH}= – 10µA I_{OH}= – 5mA, V_{DD}= 5.0V I_{OH}= – 1.5mA, V_{DD}= 3.0V	V_{DD}-0.1 3.5 2.0			V
I_{IL} I_{IH}	Input Leakage Current I/O Pins (pull-up resistor off)	Vin= V_{DD} or V_{SS} V_{DD}= 3.0V V_{DD}= 5.5V		0.1 0.1	1.0 1.0	µA
R_{PU}	Pull-up Resistor	Vin= 0V; All I/O Pins	50	100	200	KΩ

SPI CHARACTERISTICS

(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{CL}	Clock Frequency at SCK				500	kHz
t_{SV}	Data Set up time on Sin			TBD		
t_H	Data hold time on Sin			TBD		
t_{TS}	Delay Transmission started on Sin	8MHz	0	Note 1		µs

Note 1. Minimum time 0µs
Maximum time 1 instruction cycle

TIMER1 CHARACTERISTICS
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{12}{f_{INT}}$			s
f_{IN}	Input Frequency on TIM1 Pin(1)				$\frac{f_{INT}}{4}$	MHz
t_W	Pulse Width at TIM1 Pin(1)	V_{DD} = 3.0V V_{DD} = 4.5V V_{DD} = 5.5V	1 125 125			µs ns ns

Note :
1. Not available for ST6260B

AR TIMER CHARACTERISTICS
(T_A = -40 to +85°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{RES}	Resolution		$\frac{1}{f_{INT}}$			s
f_{ARin}	Input Frequency on ARTIMin pin	STOP Mode RUN and WAIT Modes			2 $\frac{f_{INT}}{4}$	MHz MHz
t_W	Pulse Width at ARTIMin Pin	V_{DD} = 3.0V V_{DD} = 4.5V V_{DD} = 5.5V	125 125 125			ns ns ns

A/D CONVERTER CHARACTERISTICS

(T_A= -40 to +85°C unless otherwise specified)

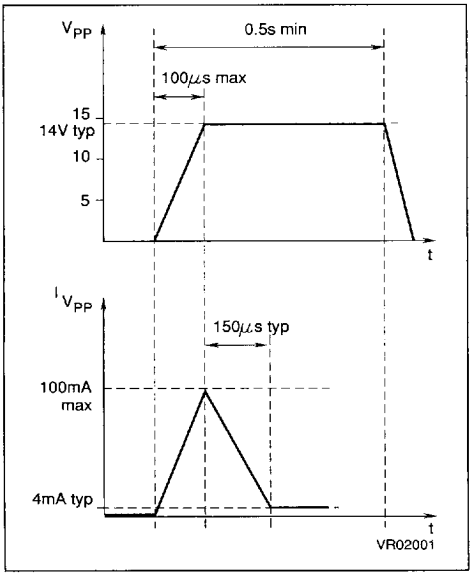
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution			8		Bit
A_{TOT}	Total Accuracy (1) (2)	f_{OSC} > 1.2MHz f_{OSC} > 32kHz			±2 ±4	LSB
t_C	Conversion Time	f_{OSC} = 8MHz		70		µs
V_{AN}	Conversion Range		V_{SS}		V_{DD}	V
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN} (3)	Analog Input Capacitance			2	5	pF
ASI	Analog Source Impedance	Analog Channel switched just before conversion start (4)			30	kΩ

Notes:

1. Noise at V_{DD}, V_{SS} < 10mV
2. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.
3. Excluding Pad Capacitance.
4. ASI can be increased as long as the load of the A/D Converter input capacitor is ensured before conversion start.

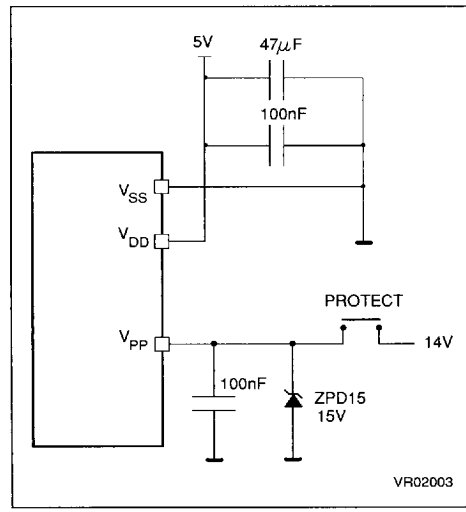
READ PROTECTION FUSE

If the ROM READOUT PROTECTION option is selected as enabled, the following waveform must be applied at the V_{PP} pin for the fuse to be blown:



The following circuit can be used for this purpose:

Figure 60. Example of READOUT PROTECTION Fuse programming circuit



Note: ZPD15 is used for overvoltage protection

ORDERING INFORMATION

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

Communication of the 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 one diskette 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.

Listing Generation & Verification. When SGS-THOMSON receives the diskette, a computer listing is generated from it. This listing refers exactly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed listing constitutes a part of the contractual agreement for the creation of the customer mask.

SGS-THOMSON sales organization will provide detailed information on contractual points.

Table 22. ROM Memory Map

Device Address	Description
0000h-007Fh	Reserved (1)
0080h-0F9Fh	User ROM
0FA0h-0FEFh	Reserved (1)
0FF0h-0FF7h	Interrupt Vectors
0FF8h-0FFBh	Reserved (1)
0FFCh-0FFDh	NMI Interrupt Vector
0FFEh-0FFFh	Reset Vector

Notes :
1. Reserved Areas should be filled with FFh

ORDERING INFORMATION TABLE

Sales Type	ROM x8	I/O	Temperature Range	Package
ST6260BB1/XXX ST6260BB6/XXX	4K Bytes	13 13	0 to +70°C -40 to +85°C	PDIP20
ST6260BM1/XXX ST6260BM6/XXX	4K Bytes	13 13	0 to +70°C -40 to +85°C	PSO20
ST6265BB1/XXX ST6265BB6/XXX	4K Bytes	21 21	0 to +70°C -40 to +85°C	PDIP28
ST6265BM1/XXX ST6265BM6/XXX	4K Bytes	21 21	0 to +70°C -40 to +85°C	PSO28

Note: /XXX is a 2-3 alphanumeric character code added to the generic sales type on receipt of a ROM code and valid options.

ST6260B, ST6265B MICROCONTROLLER OPTION LIST

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference . . . . . .

SGS-THOMSON Microelectronics references

Device:

[ ] ST6260B, [ ] ST6265B

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 Resonnator
[ ] RC Network

Power on Reset Delay:

[ ] 32768 cycles delay
[ ] 2048 cycles delay

Watchdog Selection:

[ ] Software Activation (STOP mode available)
[ ] Hardware Activation (no STOP mode)

External STOP Mode Control:

[ ] Enabled
[ ] Disabled

ROM Readout Protection:

[ ] Disabled

For Enabled option, contact your local SGS-THOMSON office.

STOP Mode Consumption:

[ ] Standard (10µA max)

For Low STOP Mode Consumption option contact your local SGS-THOMSON office.

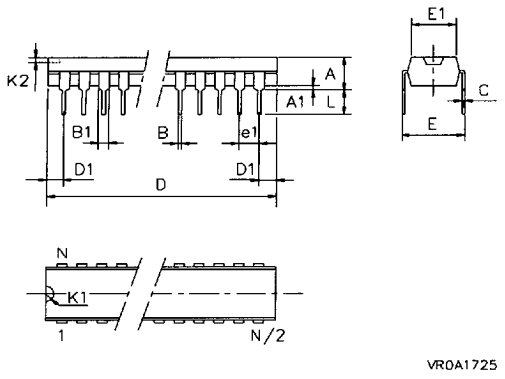
Supply Operating Range: [ ] Standard Range: 3.0V to 6.0V

Notes . . . . . .
. . . . . .
Signature . . . . . .

Date . . . . . .

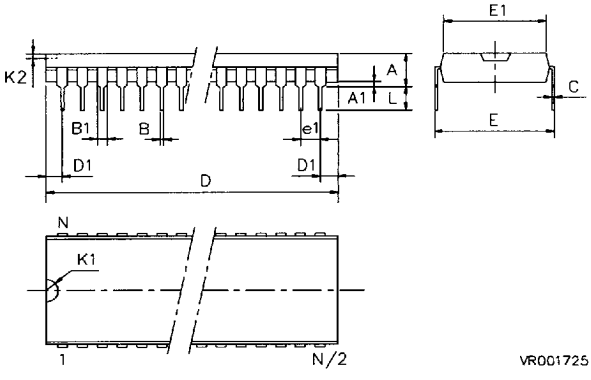
PACKAGE MECHANICAL DATA

Figure 61. 20-Pin Dual in Line Plastic (B), 300-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.93			0.155
A1	0.254			0.010		
B			0.45			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					

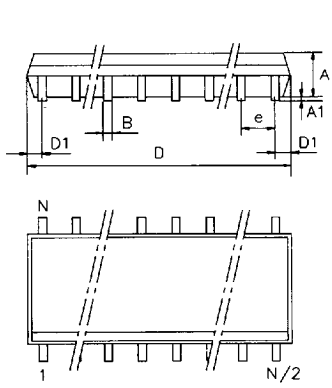
Figure 62. 28-Pin Dual in Line Plastic (B), 600-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.2		4.8	0.086		0.189
A1	0.51		1.77	0.010		0.069
B	0.38		0.58	0.015		0.023
B1	0.97		1.52	0.055		0.065
C	0.2		0.3	0.008		0.009
D	35.06		36.22	1.400		1.425
D1	–	–	–	–	–	–
E			16.3			0.641
E1		12.9				0.508
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	3.18		4.44	1.25		0.174
e1		2.54			0.10	
Number of Pins						
N	28					

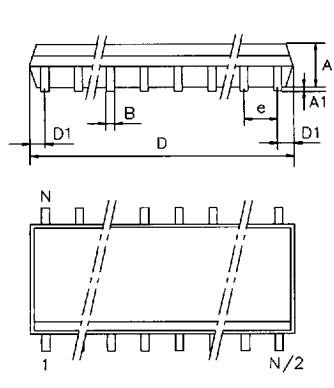
PACKAGES MECHANICAL DATA (Continued)

Figure 63. 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					

Figure 64. 28-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	17.7		18.1	.6969		.7125
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	28					