

ST6369

DATA SHEET

USE IN LIFE SUPPORT DEVICES OR SYSTEMS MUST BE EXPRESSLY AUTHORIZED.

SGS-THOMSON PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF SGS-THOMSON Microelectronics.

As used herein :

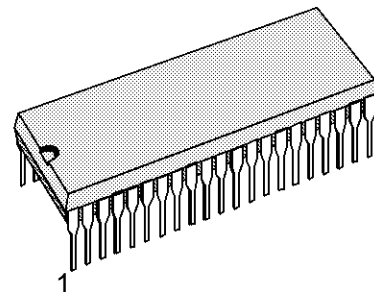
1. Life support devices or systems are those which (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided with the product, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can reasonably be expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

	Pages
ST6369	1
GENERAL DESCRIPTION	2
PIN DESCRIPTION	4
ST6369 CORE	6
MEMORY SPACES	9
INTERRUPT	15
RESET	19
WAIT & STOP MODES	21
ON-CHIP CLOCK OSCILLATOR	22
INPUT/OUTPUT PORTS	23
TIMERS	26
HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION	29
SERIAL PERIPHERAL INTERFACE	30
14-BIT PWM D/A CONVERTER	39
6-BIT PWM D/A CONVERTERS	40
A/D COMPARATOR	40
DEDICATED LATCHES	42
SOFTWARE DESCRIPTION	43
ABSOLUTE MAXIMUM RATINGS	48
PACKAGE MECHANICAL DATA	52
ORDERING INFORMATION TABLE	53
 ST63E69 ST63T69	 55
GENERAL DESCRIPTION	56
PIN DESCRIPTION	58
ST63E69,T69 EPROM/OTP DESCRIPTION	60
ABSOLUTE MAXIMUM RATINGS	61
ORDERING INFORMATION	67

8-BIT HCMOS MCU FOR DIGITAL CONTROLLED MULTI FREQUENCY MONITOR

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program ROM: 7948 bytes
- Reserved Test ROM: 244 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384 bytes
- 40-Pin Dual in Line Plastic Package
- Up to 23 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/I²C BUS and standard serial protocols
- One 14-Bit PWM D/A Converter
- Six 6-Bit PWM D/A Converters
- One A/D converter with 0.5V resolution
- Five interrupt vectors (HSYNC/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- ST6369 is supported by pin-to-pin EPROM and OTP versions.
- The development tool of the ST6369 microcontroller consists of the ST6369-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.



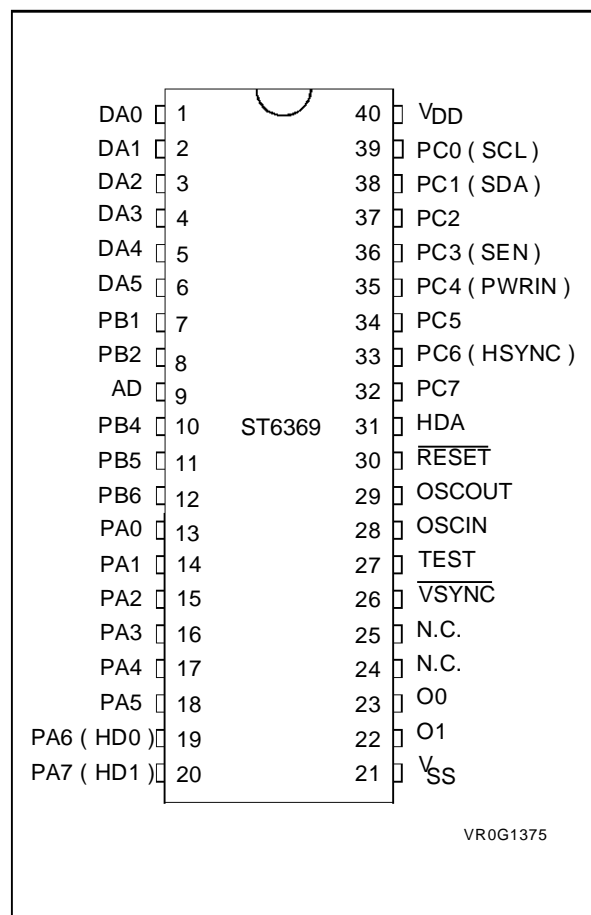
PDIP40

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	D/A Conv.
ST6369	8K	256	384	7

Figure 1. ST6369 Pin Configuration



GENERAL DESCRIPTION

The ST6369 microcontroller is member of the 8-bit HCMOS ST638x family, a series of devices specially oriented to Digital Controlled Multi Frequency Monitor applications. ST6369 members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to DCMF Monitor applications. The macrocells of the ST6369 are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), six 6-bit PWM D/A converters, an A/D converter with 0.5V resolution, a 14-bit PWM D/A converter. In addition the following memory resources are available: program ROM (8K bytes), data RAM (256 bytes), EEPROM (384 bytes).

Figure 2. ST6369 Block Diagram

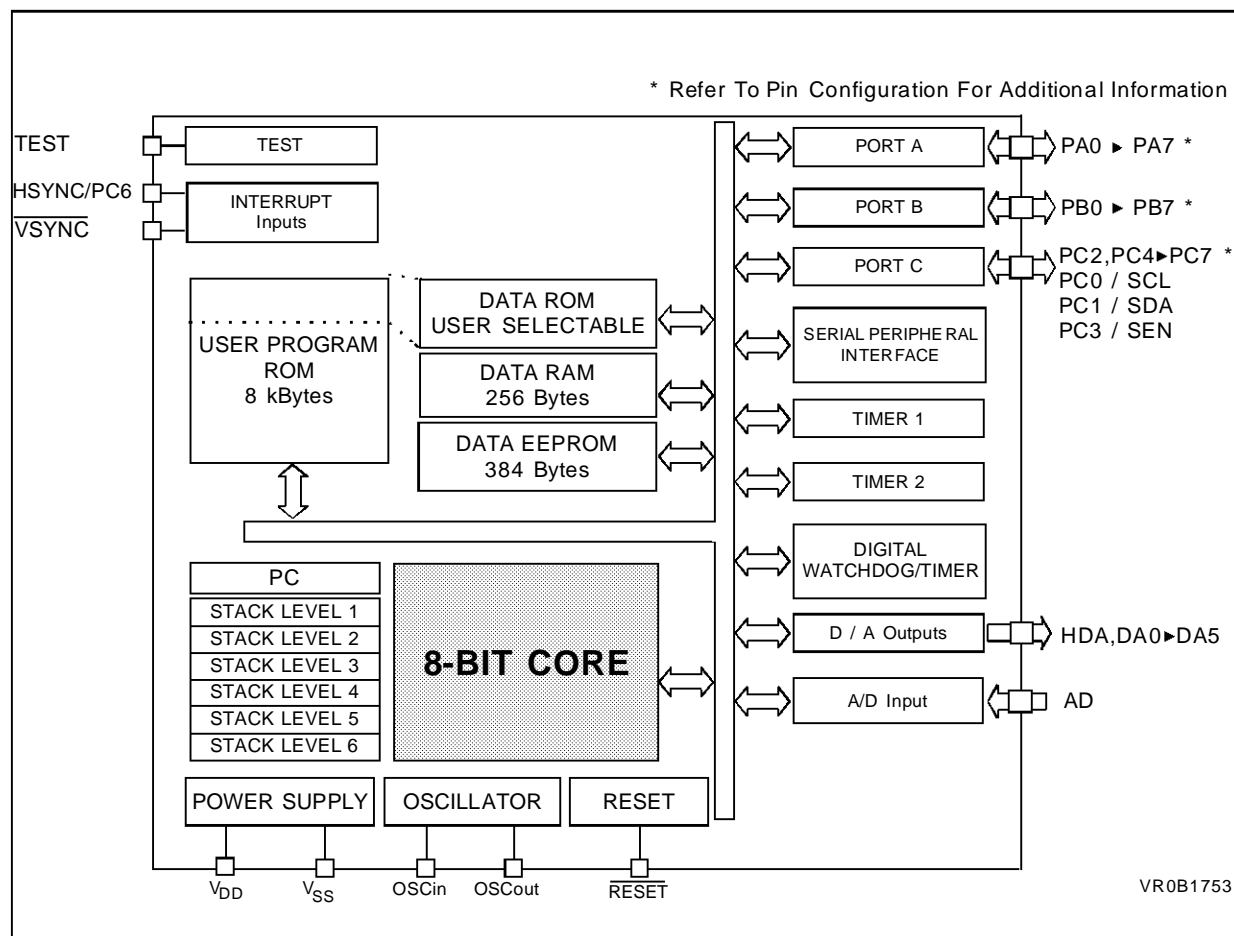


Table 1. Device Summary

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	EMULATING DEVICES
ST6369	8K	256	384	1	1	6	ST63E69, ST63T69

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, OSCOUT. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCIN pin is the input pin, the OSCOUT pin is the output pin.

RESET. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the RESET pin is low to reduce power consumption during reset phase.

TEST. The TEST pin must be held at V_{SS} for normal operation.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB1-PB2, PB4-PB6. These 5 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the HSYNC/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AD. This is the input of the on-chip 10 levels comparator that can be used to implement the Analog Keyboard function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

VSYNC. This is the Vertical Synchronization pin. This pin is connected to an internal timer interrupt.

O0, O1. These two lines are output open-drain pins with 12V drive.

HDA. This is the output pin of the on-chip 14-bit PWM D/A Converter. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AD	Input, High Impedance, 12V
HDA	Output, Push-Pull
VSYNC	Input, Pull-up, Schmitt Trigger
TEST	Input, Pull-Down
OSCIN	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCOU	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB1-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
O0, O1	Output, Open-Drain, 12V
V _{DD} , V _{SS}	Power Supply Pins

ST6369 CORE

The Core of the ST6369 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 the following block diagram figure; 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 ST6369 Core has five registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs together with the program and data memory page registers.

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 the FFH address. Accordingly, the ST6369 instruction set can use the accumulator as any other register of the data space.

Figure 4. Core Programming Model

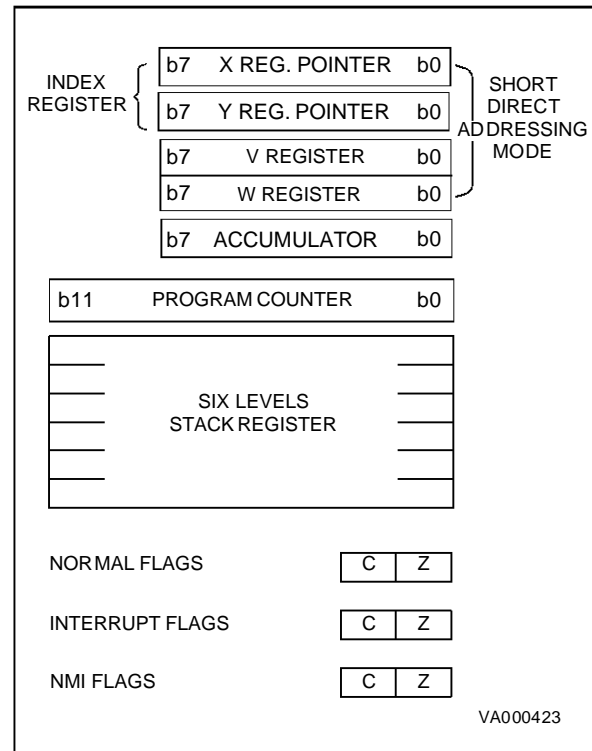
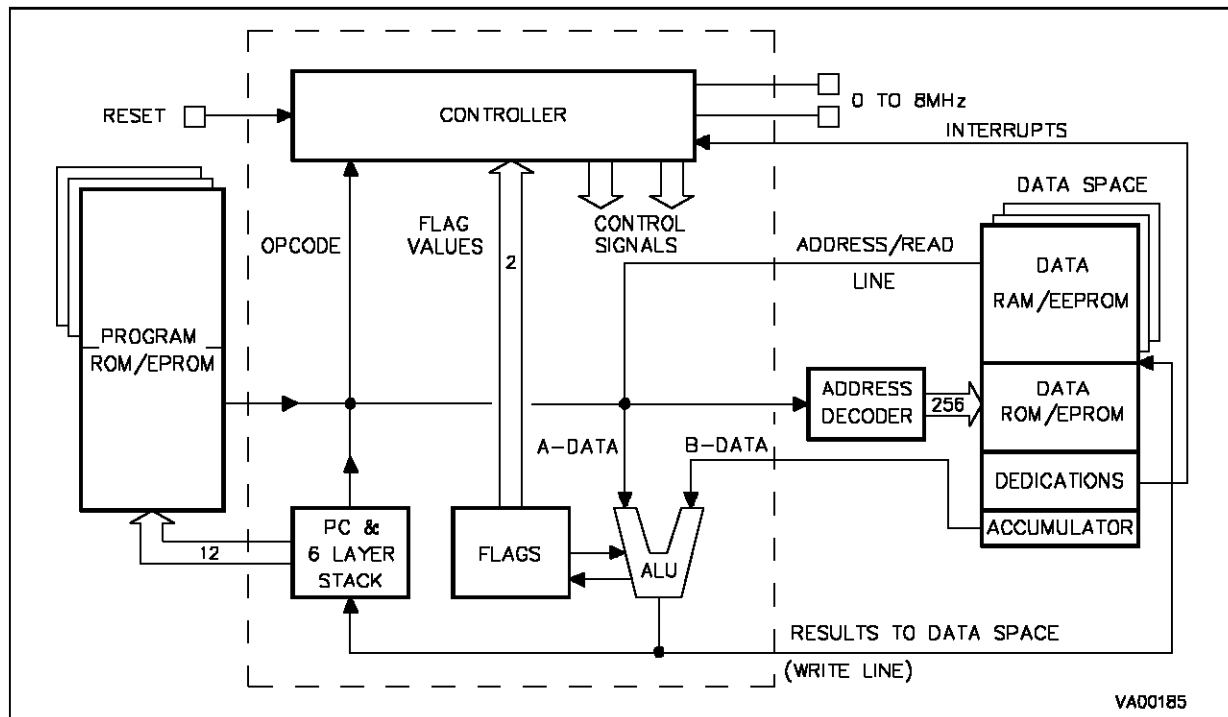


Figure 3. Core Block Diagram



ST6369 CORE (Continued)

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 the 80H (X) and 81H (Y) addresses. They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST638x 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 the 82H (V) and 83H (W) addresses. They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST638x 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 ROM Page Register. The PC value is incremented, after it is read for the address of the current instruction, by sending it through the ALU, so giving the address of the next byte in the program. To execute relative jumps the PC and the offset values 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)

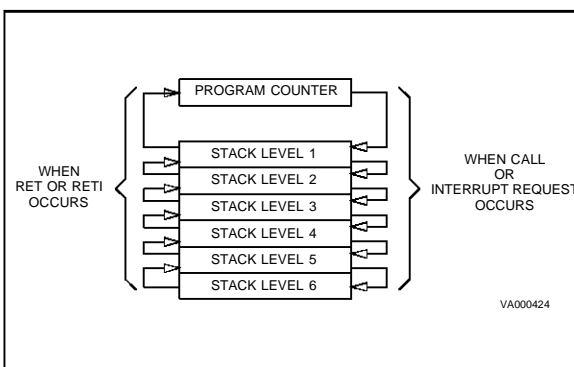
The ST6369 Core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable-Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI, ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST6369 Core uses the pair of flags that corresponds to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST6369 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. Should be observed that each flag set can only be addressed in its own routine (Not-maskable interrupt, normal interrupt or main routine). The interrupt flags are not cleared during the context switching and so, they 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 these three sets is automatically performed when an NMI, an interrupt and a RETI instructions occur. As the NMI mode is automatically selected after the reset of the MCU, the ST6369 Core uses at first the NMI flags.

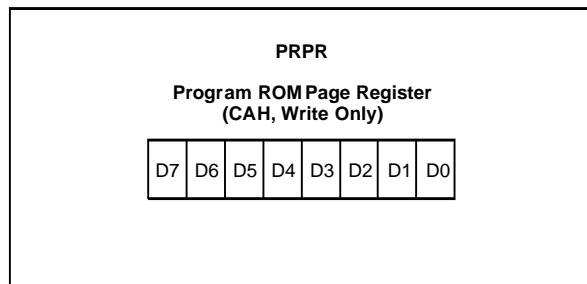
Figure 5. Stack Operation

ST6369 CORE (Continued)**Stack**

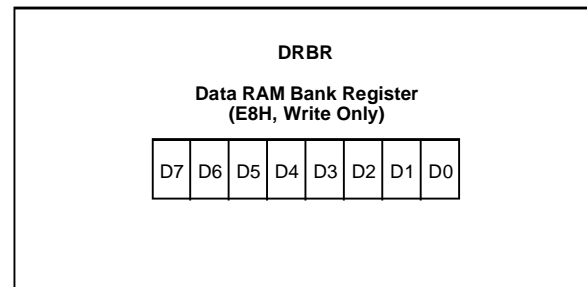
The ST6369 Core includes true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When 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 shifted back into the previous level. These two operating modes are described in Figure 5. Since the accumulator, as all other data space registers, is not stored in this stack the handling of this registers shall 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.

Memory Registers

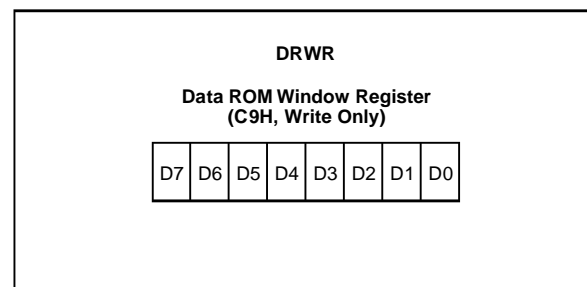
The **PRPR** can be addressed like a RAM location in the Data Space at the CAH address; nevertheless it is a write-only register that can not be accessed with single-bit operations. This register is used to select the 2-Kbyte ROM bank of the Program Space that will be addressed. The number of the page has to be loaded in the PRPR. The PRPR is not cleared during the MCU initialization and should therefore be defined before jumping out of the static page. Refer to the Program Space description for additional information concerning the use of this register. The PRPR is not modified when an interrupt or a subroutine occurs.

Figure 6. Program ROM Page Register

The **DRBR** can be addressed like a RAM location in the Data Space at the E8H address, nevertheless it is write-only register that can not 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 the bank has to be loaded in the DRBR and the instruction has to point to the selected location as it was in the 0 bank (from 00H address to 3FH address). This register is undefined after Reset. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

Figure 7. Data RAM Bank Register

The **DRWR** register can be addressed like a RAM location in the Data Space at the C9H address, nevertheless it is write-only register that can not be accessed with single-bit operations. This register is used to move up and down the 64-byte read-only data window (from the 40H address to 7FH address of the Data Space) along the ROM of the MCU by step of 64 bytes. The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address given in the instruction (as less significant bits) and the content of the DRWR (as most significant bits). Refer to the Data Space description for additional information.

Figure 8. Data ROM Window Register

MEMORY SPACES

The MCUs operate in three different memory spaces: Stack Space, Program Space and Data Space. A description of these spaces is shown in Figure 9.

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.

Program Space

The program space is physically implemented in the ROM 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 user vectors. It is addressed thanks to the 12-bit Program Counter register (PC register) and so, the ST6369 Core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2-Kbyte ROM banks as it is shown in Figure 11 in which a 8K bytes memory is described. These banks are addressed by pointing to the 000H-7FFH locations of the Program Space thanks to the Program Counter, and by writing the appropriate code in the Program ROM Page Register (PRPR) located at the CAH address of the Data Space. Because interrupts and common sub-

outines should be available all the time only the lower 2K byte of the 4K program space are bank switched while the upper 2K byte can be seen as static space. Table 3 gives the different codes that allows the selection of the corresponding banks. Note that, from the memory point of view, the Page 1 and the Static Page represent the same physical memory: it is only a different way of addressing the same location.

Figure 10. 8K Bytes Program Space Addressing Description

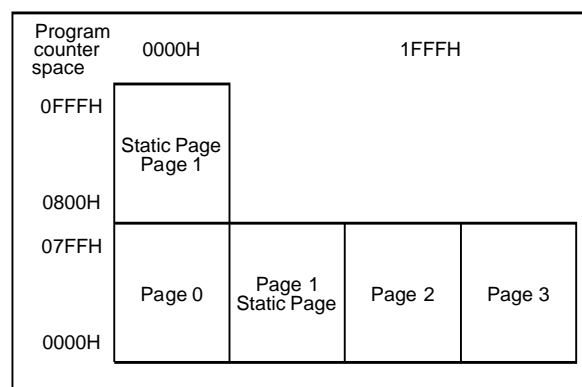
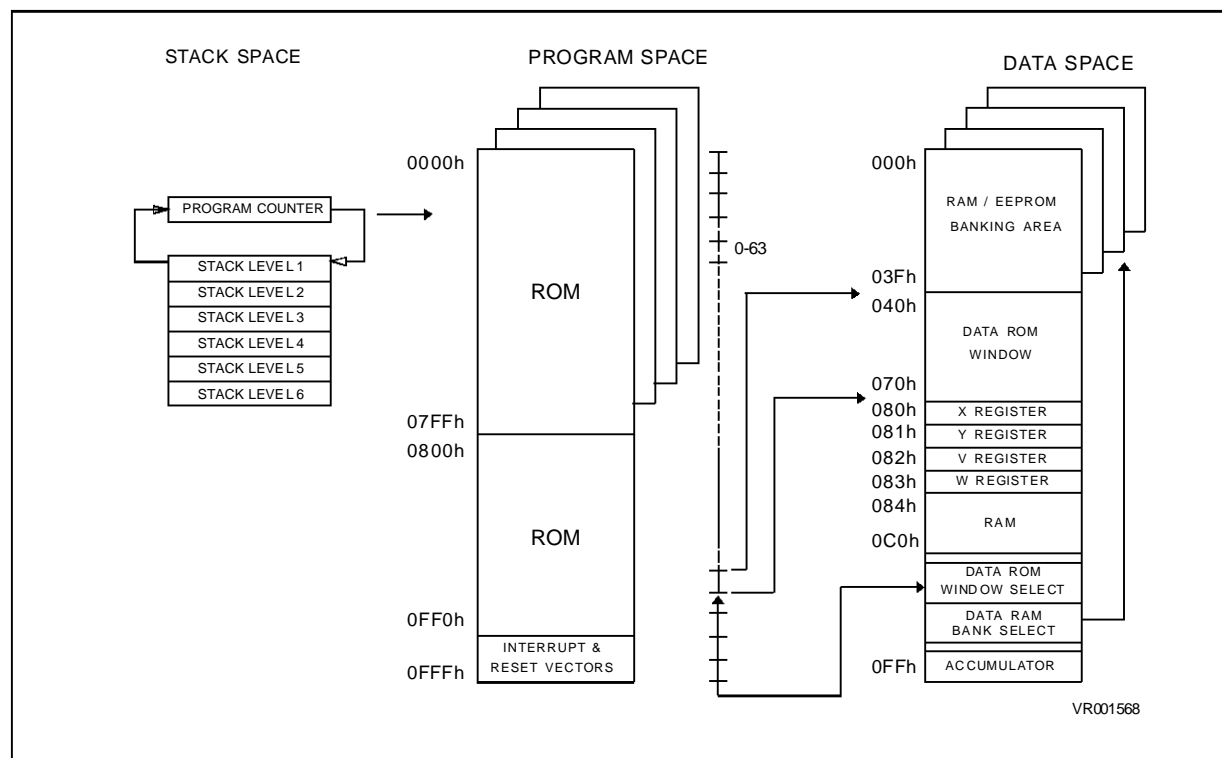
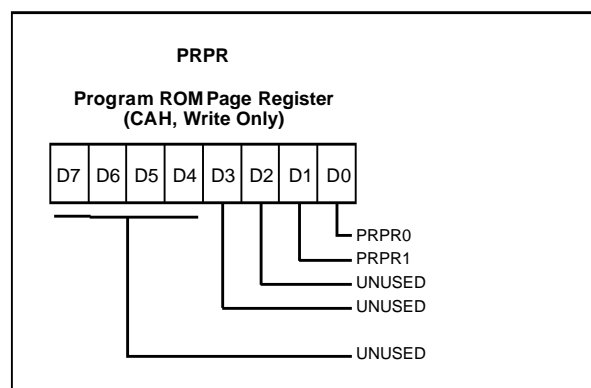


Figure 9. Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Figure 11. Program ROM Page Register



D7-D2. These bits are not used but have to be written to "0".

PRPR1-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of 4K program address space as specified in Table 3. This register is undefined on reset.

Table 3. Program ROM Page Register Coding

PRPR1	PRPR0	PC11	Memory Page
X	X	1	Static Page (Page 1)
0	0	0	Page 0
0	1	0	Page 1 (Static Page)
1	0	0	Page 2
1	1	0	Page 3

Note. Only the lower part of address space has been bankswitched because interrupt vectors and common subroutines should be available all the time. The reason of this structure is due to the fact that it is not possible to jump from a dynamic page to another, unless jumping back to the static page, changing contents of PRPR, and, then, jumping to a different dynamic page.

Care is required when handling the PRPR as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupts drivers, as the driver cannot save and then restore its previous content. Anyway, this operation may be necessary if the sum of common routines and interrupt drivers will take more than 2K bytes; in this case could be necessary to divide the interrupt driver in a (minor) part in the static page (start and end), and in the second (major) part in one dynamic page. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the PRPR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the PRPR is not affected.

Table 4. ST6369 Program ROM Map

ROM Page	Device Address	Description
PAGE 0	0000H-007FH 0080H-07FFH	Reserved User ROM
PAGE 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000H-000FH 0010H-07FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	Reserved User ROM

MEMORY SPACES (Continued)**Data Space**

The instruction set of the ST6369 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 ad-

ressing of RAM (256 bytes), EEPROM (384 bytes), ST6369 Core/peripheral registers, and read-only data such as constants and the look-up tables.

Figure 12. Data Space

b7	b0	
DATA RAM/EEPROM BANK AREA		000H
		03FH
DATA ROM WINDOW AREA		040H
		07FH
X REGISTER		080H
Y REGISTER		081H
V REGISTER		082H
W REGISTER		083H
		084H
DATA RAM		
		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
PROGRAM ROM PAGE REGISTER		0CAH
RESERVED		0CBH
SPI DATA REGISTER		0CCH
		0CDH
RESERVED		
		0D1H
TIMER 1 PRESCALER REGISTER		0D2H
TIMER 1 COUNTER REGISTER		0D3H
TIMER 1 STATUS/CONTROL REG.		0D4H
		0D5H
RESERVED		
		0D7H
WATCHDOG REGISTER		0D8H

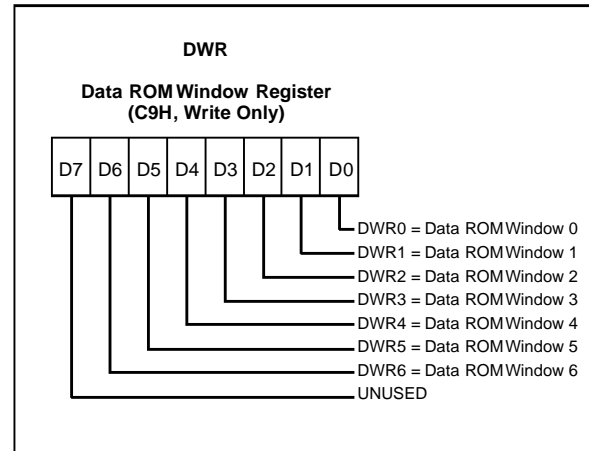
Figure 13. Data Space (Continued)

b7	b0	
RESERVED		0D9H
TIMER 2 PRESCALER REGISTER		0DAH
TIMER 2 COUNTER REGISTER		0DBH
TIMER 2 STATUS CONTROL REG.		0DCH
		0DDH
RESERVED		
		0DFH
DA0 DATA/CONTROL REGISTER		0E0H
DA1 DATA/CONTROL REGISTER		0E1H
DA2 DATA/CONTROL REGISTER		0E2H
DA3 DATA/CONTROL REGISTER		0E3H
AD, HSYNC RESULT REGISTER		0E4H
OUTPUTS CONTROL REGISTER		0E5H
DA4 DATA/CONTROL REGISTER		0E6H
DA5 DATA/CONTROL REGISTER		0E7H
DATA RAM BANK REGISTER		0E8H
DEDIC. LATCHES CONTROL REG.		0E9H
EEPROM CONTROL REGISTER		0EAH
SPI CONTROL REGISTER 1		0EBH
SPI CONTROL REGISTER 2		0ECH
RESERVED		0EDH
HDA DATA REGISTER 1		0EEH
HDA DATA REGISTER 2		0EFH
		0F0H
RESERVED		
		0FEH
ACCUMULATOR		0FFH

MEMORY SPACES (Continued)

Data ROM Addressing. All the read-only data are physically implemented in the ROM in which the Program Space is also implemented. The ROM therefore contains the program to be executed and also 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 ST6369 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. This window is located from the 40H address to the 7FH address in the Data space and allows the direct reading of the bytes from the 000H address to the 03FH address in the ROM. All the bytes of the ROM 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 in writing the appropriate code in the Write-only Data ROM Window register (DRWR, location C9H). The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address in the Data Space (as less significant bits) and the content of the DRWR (as most significant bits). So when addressing location 40H of data space, and 0 is loaded in the DRWR, the physical addressed location in ROM is 00H.

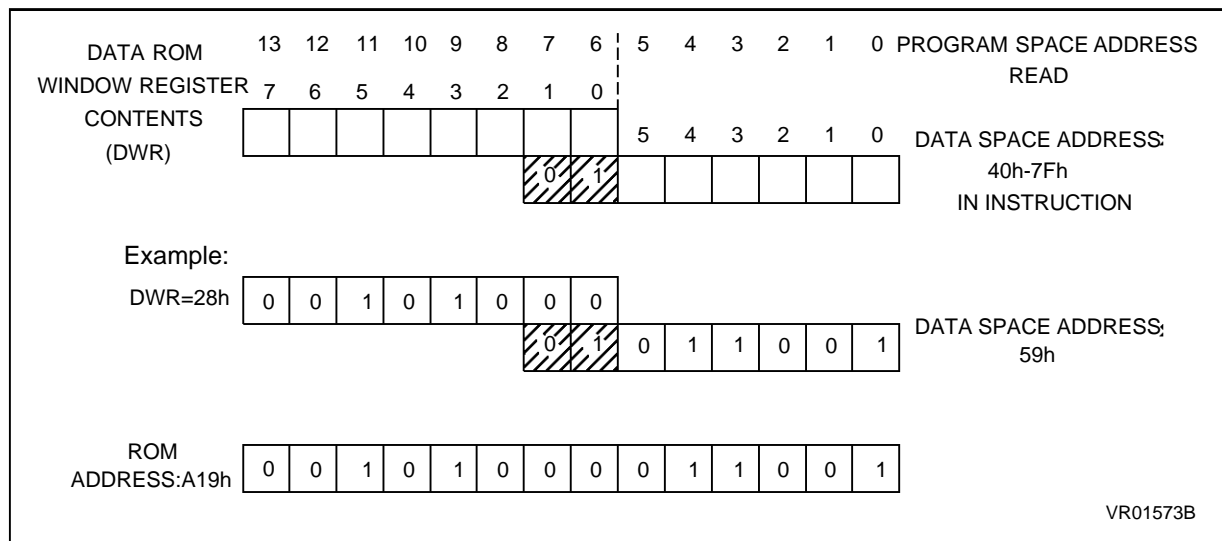
Figure 14. Data ROM Window Register



DWR6-DWR0. These are the Data Rom Window bits that correspond to the upper bits of data ROM program space. This register is undefined after reset.

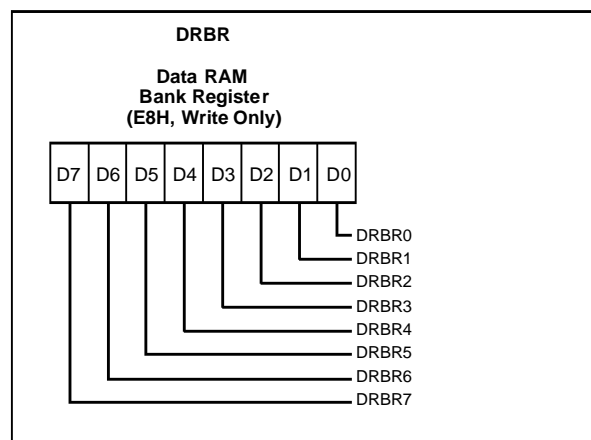
Note. Care is required when handling the DRWR as it is write only. For this reason, it is not allowed to change the DRWR contents while executing interrupts drivers, as the driver cannot save and then restore its previous content. If it is impossible to avoid the writing of this register in interrupts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

Figure 15. Data ROM Window Memory Addressing



MEMORY SPACES (Continued)**Data RAM/EEPROM**

In the ST6369 64 bytes of data RAM are directly addressable in the data space from 80H to BFH addresses. The additional 192 bytes of RAM, the 384 bytes of EEPROM can be addressed using the banks of 64 bytes located between addresses 00H and 3FH. The selection of the bank is done by programming the Data RAM Bank Register (DRBR) located at the E8H address of the Data Space. In this way each bank of RAM, EEPROM can select 64 bytes at a time. No more than one bank should be set at a time.

Figure 16. Data RAM Bank Register

DRBR7,DRBR1,DRBR0. These bits select the EEPROM pages.

DRBR4,DRBR3,DRBR2. Each of these bits, when set, will select one RAM page.

This register is undefined after reset.

Table 5 summarizes how to set the Data RAM Bank Register in order to select the various banks or pages.

Note :

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

EEPROM Description

The data space of ST6369 family from 00H to 3FH is paged as described in Table 5. 384 bytes of EEPROM located in six pages of 64 bytes (pages 0,1,2,3,4 and 5, see Table 5).

Table 5. Data RAM Bank Register Set-up

DRBR Value		Selection
Hex.	Binary	
01H	0000 0001	EEPROM Page 0
02H	0000 0010	EEPROM Page 1
03H	0000 0011	EEPROM Page 2
81H	1000 0001	EEPROM Page 3
82H	1000 0010	EEPROM Page 4
83H	1000 0011	EEPROM Page 5
04H	0000 0100	RAM Page 2
08H	0000 1000	RAM Page 3
10H	0001 0000	RAM Page 4

MEMORY SPACES (Continued)

Through the programming of the Data RAM Bank Register (DRBR=E8H) the user can select the bank or page leaving unaffected the way to address the static registers. The way to address the "dynamic" page is to set the DRBR as described in Table 5 (e.g. to select EEPROM page 0, the DRBR has to be loaded with content 01H, see Data RAM/EEPROM addressing for additional information). Bits 0, 1 and 7 of the DRBR are dedicated to the EEPROM.

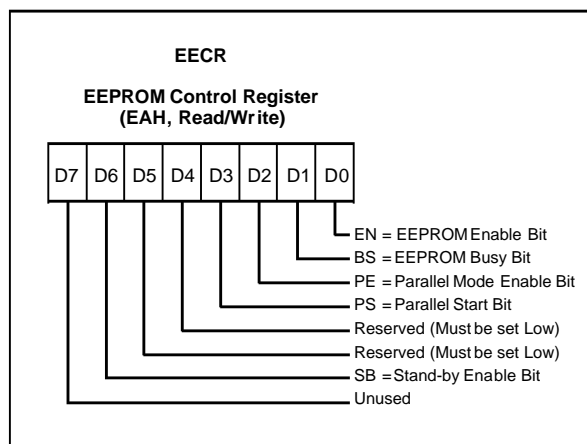
The EEPROM pages do not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECR=EAH). Any EEPROM location can be read just like any other data location, also in terms of access time.

To write an EEPROM location takes an average time of 5 ms (10ms max) and during this time the EEPROM is not accessible by the Core. A busy flag can be read by the Core to know the EEPROM status before trying any access. In writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). The BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. The PMODE consists in accessing 8 bytes per time.

D7. Not used

SB. 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 leakage values.

Figure 17. EEPROM Control Register



D5, D4. Reserved for testing purposes, they must be set to zero.

PS. SET ONLY. Once in Parallel Mode, as soon as the user software sets the PS bit the parallel writing of the 8 adjacent registers will start. PS is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the remaining undefined bytes will have no particular content.

PE. WRITE ONLY. This bit must be set by the user program in order to perform parallel programming (more bytes per time). If PE is set and the "parallel start bit" (PS) is low, up to 8 adjacent bytes can be written at the maximum speed, the content being stored in volatile registers. These 8 adjacent bytes can be considered as row, whose A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bytes. PE is automatically reset at the end of any parallel programming procedure. PE can be reset by the user software before starting the programming procedure, leaving unchanged the EEPROM registers.

BS. READ ONLY. This bit will be automatically set by the CORE when the user program modifies an EEPROM register. The user program has to test it before any read or write EEPROM operation; any attempt to access the EEPROM while "busy bit" is set will be aborted and the writing procedure in progress completed.

EN. WRITE ONLY. This bit MUST be set to one in order to write any EEPROM register. If the user program will attempt to write the EEPROM when EN= "0" the involved registers will be unaffected and the "busy bit" will not be set.

After RESET the content of EECR register will be 00H.

Notes :

When the EEPROM is busy (BS="1") the EECR can not be accessed in write mode, it is only possible to read BS status. This implies that as long as the EEPROM is busy it is not possible to change the status of the EEPROM control register. EECR bits 4 and 5 are reserved for test purposes, and must never be set to "1".

Additional Notes on Parallel Mode. If the user wants to perform a parallel programming the first action should be the set to one the PE bit; from this moment the first 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 reset.

MEMORY SPACES (Continued)

ting PE without programming the EEPROM. After the ROW address latching the Core can “see” just one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while PE is set.

As soon as PE bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or just in a subset. PS setting will modify the EEPROM registers corresponding to the ROW latches accessed after PE. For example, if the software sets PE and accesses EEPROM in writing at addresses 18H, 1AH, 1BH and then sets PS, these three registers will be modified at the same time; the remaining bytes will have no particular content. Note that PE is internally reset at the end of the programming procedure. This implies that the user must set PE bit between two parallel programming procedures. Anyway the user can set and then reset PE without performing any EEPROM programming. PS is a set only bit and is internally reset at the end of the programming procedure. Note that if the user tries to set PS while PE is not set there will not be any programming procedure and the PS bit will be unaffected. Consequently PS bit can not be set if EN is low. PS can be affected by the user set if, and only if, EN and PE bits are also set to one.

INTERRUPT

The ST6369 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 6). When a source provides an interrupt request, and the request processing is also enabled by the ST6369 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 relationship between vector and source and the associated priority is hardware fixed for the different ST638x devices. For some interrupt sources it is also possible to select by software the kind of event that will generate the interrupt.

All interrupts can be disabled by writing to the GEN bit (global interrupt enable) of the interrupt option register (address C8H). After a reset, ST6369 is in non maskable interrupt mode, so no interrupts will be accepted and NMI flags will be used, until a RETI instruction is executed. If an interrupt is executed, one special cycle is made by the core, during that the PC is set to the related interrupt vector address. A jump instruction at this address has to redirect program execution to the beginning of the related interrupt routine. The interrupt detecting cycle, also resets the related interrupt flag (not available to the user), so that another interrupt can be stored for this current vector, while its driver is under execution.

If additional interrupts arrive from the same source, they will be lost. NMI can interrupt other interrupt routines at any time, while other interrupts cannot interrupt each other. If more than one interrupt is waiting for service, they are executed according to their priority. The lower the number, the higher the priority. Priority is, therefore, fixed. Interrupts are checked during the last cycle of an instruction (RETI included). Level sensitive interrupts have to be valid during this period.

Table 6 details the different interrupt vectors/sources relationships.

Interrupt Vectors/Sources

The ST6369 Core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines. The interrupt vectors are located in the fixed (or static) page of the Program Space.

INTERRUPT (Continued)**Table 6. Interrupt Vectors/Sources Relationships**

Interrupt Source	Associated Vector	Vector Address
PC6/IRIN Pin (1)	Interrupt Vector # 0 (NMI)	0FFCH-0FFDH
Timer 2	Interrupt Vector # 1	0FF6H-0FF7H
Vsync	Interrupt Vector # 2	0FF4H-0FF5H
Timer 1	Interrupt Vector # 3	0FF2H-0FF3H
PC4/PWRIN	Interrupt Vector # 4	0FF0H-0FF1H

Note: 1. This pin is associated with the NMI Interrupt Vector

The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at the (FFCH,FFDH) addresses in the Program Space. This vector is associated with the PC6/IRIN pin.

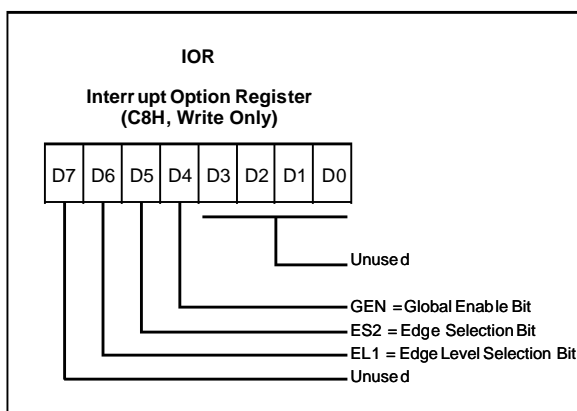
The interrupt vectors located at addresses (FF6H,FF7H), (FF4H,FF5H), (FF2H,FF3H), (FF0H,FF1H) are named interrupt vectors #1, #2, #3 and #4 respectively. These vectors are associated with TIMER 2 (#1), VSYNC (#2), TIMER 1 (#3) and PC4(PWRIN) (#4).

Interrupt Priority

The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST6369 Core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is hardware fixed.

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 C8H address, nevertheless it is write-only register that can not 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 4 and 5 of the IOR register.

Figure 18. Interrupt Option Register

D7. Not used.

EL1. This is the Edge/Level selection bit of interrupt #1. When set to one, the interrupt is generated on low level of the related signal; when cleared to zero, the interrupt is generated on falling edge. The bit is cleared to zero after reset.

ES2. This is the edge selection bit on interrupt #2. This bit is used on the ST6369 devices with on-chip OSD generator for VSYNC detection.

GEN. This is the global enable bit. When set to one all interrupts are globally enabled; when this bit is cleared to zero all interrupts are disabled (excluding NMI).

D3 - D0. These bits are not used.

INTERRUPT (Continued)

Interrupt Procedure

The interrupt procedure is very similar to a call procedure; 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 (refer also to Figure 19. Interrupt Processing Flow Chart):

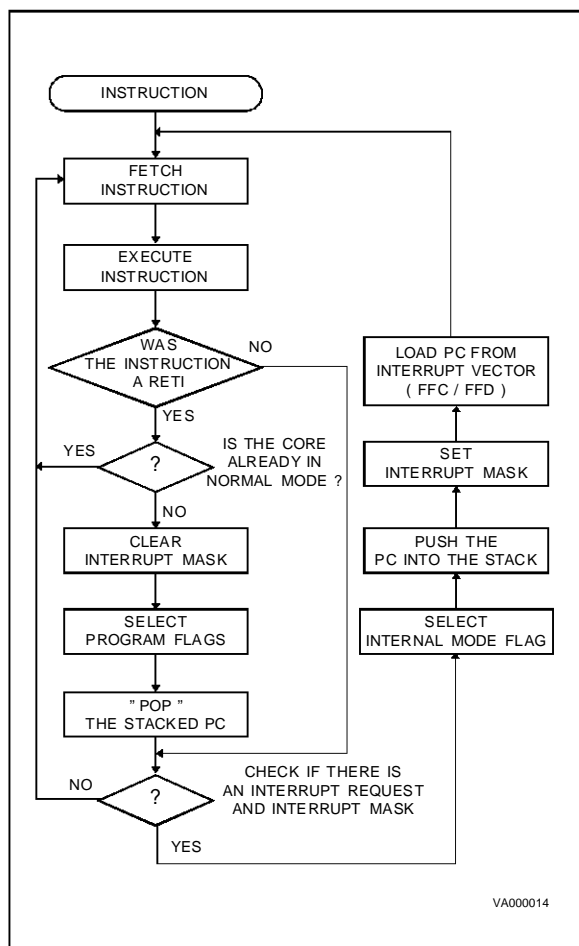
- Interrupt detection
- The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (resp. the NMI flags)
- The value of the PC is stored in the first level of the stack - The normal interrupt lines are inhibited (NMI still active)
- The edge flip-flop is reset
- The related interrupt vector is loaded in the PC.
- 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)
- Interrupt servicing
- Return from interrupt (RETI)
- Automatically the ST63xx core switches back to the normal flags (resp 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. 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.

ST6369 Interrupt Details

IR Interrupt (#0). The IRIN/PC6 Interrupt is connected to the first interrupt #0 (NMI, 0FFCH). If the $\overline{\text{IRINT}}$ interrupt is disabled at the Latch circuitry, then it will be high. The #0 interrupt input detects a

Figure 19. Interrupt Processing Flow-Chart



high to low level. Note that once #0 has been latched, then the only way to remove the latched #0 signal is to service the interrupt. #0 can interrupt the other interrupts. A simple latch is provided from the PC6(IRIN) pin in order to generate the $\overline{\text{IRINT}}$ signal. This latch can be triggered by either the positive or negative edge of IRIN signal. $\overline{\text{IRINT}}$ is inverted with respect to the latch. The latch can be read by software and reset by software.

INTERRUPT (Continued)

TIMER 2 Interrupt (#1). The TIMER 2 Interrupt is connected to the interrupt #1 (0FF6H). The TIMER 2 interrupt generates a low level (which is latched in the timer). Only the low level selection for #1 can be used. Bit 6 of the interrupt option register C8H has to be set.

VSYNC Interrupt (#2). The VSYNC Interrupt is connected to the interrupt #2. When disabled the VSYNC INT signal is low. The VSYNC INT signal is inverted with respect to the signal applied to the VSYNC pin. Bit 5 of the interrupt option register C8H is used to select the negative edge (ES2=0) or the positive edge (ES2=1); the edge will depend on the application. Note that once an edge has been latched, then the only way to remove the latched signal is to service the interrupt. Care must be taken not to generate spurious interrupts. This interrupt may be used for synchronize to the VSYNC signal in order to change characters in the OSD only when the screen is on vertical blanking (if desired). This method may also be used to blink characters.

TIMER 1 Interrupt (#3). The TIMER 1 Interrupt is connected to the fourth interrupt #3 (0FF2H) which detects a low level (latched in the timer).

PWR Interrupt (#4). The PWR Interrupt is connected to the fifth interrupt #4 (0FF0H). If the PWRINT is disabled at the PWR circuitry, then it will be high. The #4 interrupt input detects a low level. A simple latch is provided from the PC4 (PWRIN) pin in order to generate the PWRINT signal. This latch can be triggered by either the positive or negative edge of the PWRIN signal. PWRINT is inverted with respect to the latch. The latch can be reset by software.

Notes Global disable does not reset edge sensitive interrupt flags. These edge sensitive interrupts become pending again when global disabling is released. Moreover, edge sensitive interrupts are stored in the related flags also when interrupts are globally disabled, unless each edge sensitive interrupt is also individually disabled before the interrupting event happens. Global disable is done by clearing the GEN bit of Interrupt option register, while any individual disable is done in the control register of the peripheral. The on-chip Timer peripherals have an interrupt request flag bit (TMZ), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI) that must be set to one to allow the transfer of the flag bit to the Core.

RESET

The ST6369 devices can be reset in two ways: by the external reset input (RESET) tied low and by the hardware activated digital watchdog peripheral.

RESET Input

The external active low reset pin is used to reset the ST6369 devices and provide an orderly software startup procedure. The activation of the Reset pin may occur at any time in the RUN or WAIT mode. Even short pulses at the reset pin will be accepted since the reset signal is latched internally and is only cleared after 2048 clocks at the oscillator pin. The clocks from the oscillator pin to the reset circuitry are buffered by a schmitt trigger so that an oscillator in start-up conditions will not give spurious clocks. When the reset pin is held low, the external crystal oscillator is also disabled in order to reduce current consumption. The MCU is configured in the Reset mode as long as the signal of the RESET pin is low. The processing of the program is stopped and the standard Input/Output ports (port A, port B and port C) are in the input state. As soon as the level on the reset pin becomes high, the initialization sequence is executed. Refer to the MCU initialization sequence for additional information.

Watchdog Reset

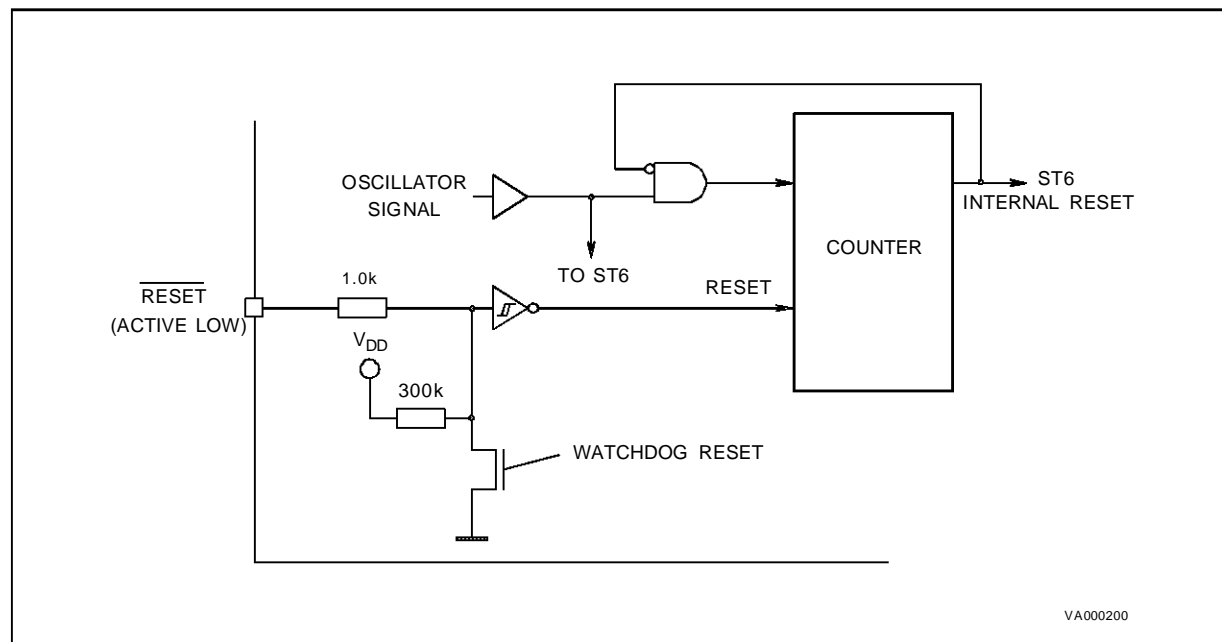
The ST6369 devices are provided with an on-chip hardware activated digital watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed and the end-of-count is reached, then the reset state will be latched into the MCU and an internal circuit pulls down the reset pin. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the reset pin. This causes the positive transition at the reset pin. The MCU will then exit the reset state after 2048 clocks on the oscillator pin.

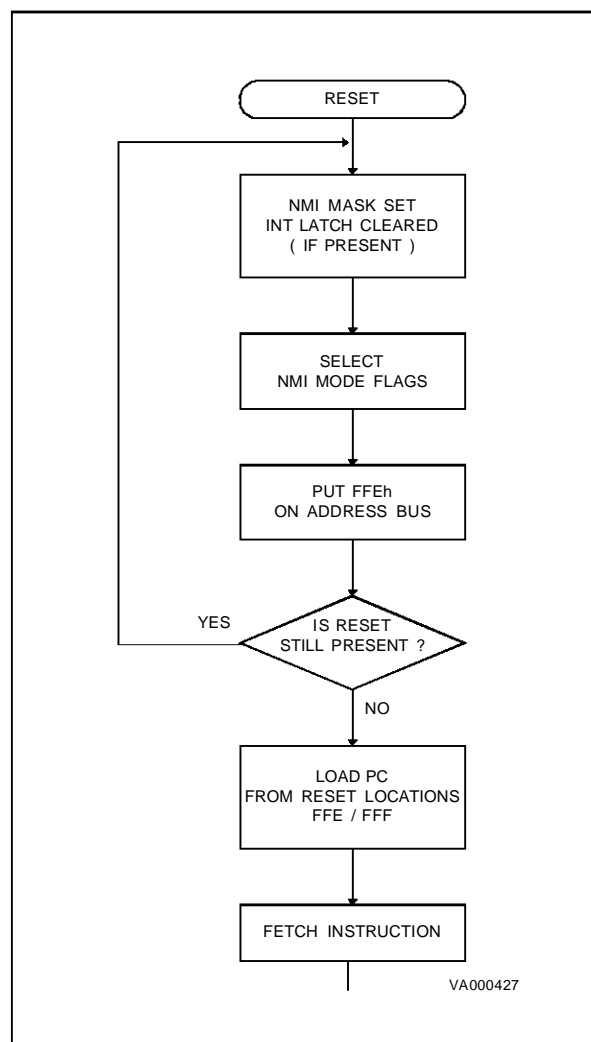
Application Notes

An external resistor between V_{DD} and the reset pin is not required because an internal pull-up device is provided. The user may prefer to add an external pull-up resistor.

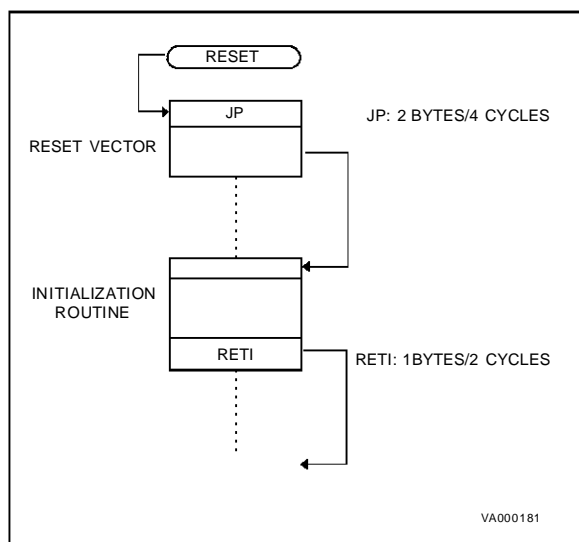
An internal Power-on device does not guarantee that the MCU will exit the reset state when V_{DD} is above 4.5V and therefore the RESET pin should be externally controlled.

Figure 20. Internal Reset Circuit



RESET (Continued)**Figure 21. Reset & Interrupt Processing Flow-Chart****MCU Initialization Sequence**

When a reset occurs the stack is reset to 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

Figure 22. Restart Initialization Program Flow-Chart

beginning of the program has to be written into these locations. After a reset the interrupt mask 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 ST6369 will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

RESET Low Power Mode

When the reset pin is low, the quartz oscillator is Disabled allowing reduced current consumption. When the reset pin is raised the quartz oscillator is enabled and oscillations will start to build up. The internal reset circuitry will count 2048 clocks on the oscillator pin before allowing the MCU to go out of the reset state; the clocks are after a schmitt trigger so that false or multiple counts are not possible.

WAIT & STOP MODES

The STOP and WAIT modes have been implemented in the ST6369 Core in order to reduce the consumption of the device when the latter has no instruction to execute. These two modes are described in the following paragraphs. On ST6369 as the hardware activated digital watchdog function is present the STOP instruction is de-activated and any attempt to execute it will cause the automatic execution of a WAIT instruction.

WAIT Mode

The configuration of the MCU in the WAIT mode occurs 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 is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide clock signal to the peripherals. The timers counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. 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 will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behaviour depends on the state of the ST6369 Core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST6369 Core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

On ST6369 the hardware watchdog is present and the STOP instruction has been de-activated. Any attempt to execute a STOP will cause the automatic execution of a WAIT instruction.

Exit from WAIT Mode

The following paragraphs describe the output procedure of the ST6369 Core from WAIT mode when

an interrupt occurs. 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 sequence, but also of the type of the interrupt request that is generated. In all cases the GEN bit of IOR has to be set to 1 in order to restart from WAIT mode. Contrary to the operation of NMI in the run mode, the NMI is masked in WAIT mode if GEN=0.

Normal Mode. If the ST6369 Core was in the main routine when the WAIT instruction has been executed, the ST6369 Core outputs from the 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 WAIT instruction is executed if no other interrupts are pending.

Non-maskable Interrupt Mode. If the WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST6369 Core outputs from the wait mode as soon as any interrupt occurs: the instruction that follows the WAIT instruction is executed and the ST6369 Core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST6369 Core was in the interrupt mode before the initialization of the WAIT sequence, it outputs from the 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 was entered will be completed with the execution of the instruction that follows the WAIT and the ST6369 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 was entered will be completed with the execution of the instruction that follows the WAIT and the ST6369 Core is still in the normal interrupt mode.

Notes :

If all the interrupt sources are disabled, the restart of the MCU can only be done by a Reset activation. The Wait instruction is not executed if an enabled interrupt request is pending. In the ST6369 the hardware activated digital watchdog function is present. As the watchdog is always activated the STOP instruction is de-activated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal quartz, a ceramic resonator, or an external signal (provided to the OSCIN pin) may be used to generate a system clock with various stability/cost trade-offs. The typical clock frequency is 8MHz. Please note that different frequencies will affect the operation of those peripherals (D/As, SPI) whose reference frequencies are derived from the system clock.

The different clock generator options connection methods are shown in Figures 23 and 24. One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625 μ Sec.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially RS), oscillator load capacitance (CL), IC parameters, ambient temperature, and supply voltage. It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1 and CL2 are in the range of 15pF to 22pF but these should be chosen based on the crystal manufacturers specification. Typical input capacitance for OSCIN and OSCOUT pins is 5pF.

The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer and the Watchdog clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to be executed (See Table 7).

Table 7. Instructions Timing with 8MHz Clock

Instruction Type	Cycles	Execution Time
Branch if set/reset	5 Cycles	8.125 μ s
Branch & Subroutine Branch	4 Cycles	6.50 μ s
Bit Manipulation	4 Cycles	6.50 μ s
Load Instruction	4 Cycles	6.50 μ s
Arithmetic & Logic	4 Cycles	6.50 μ s
Conditional Branch	2 Cycles	3.25 μ s
Program Control	2 Cycles	3.25 μ s

Figure 23. Clock Generator Option (1)

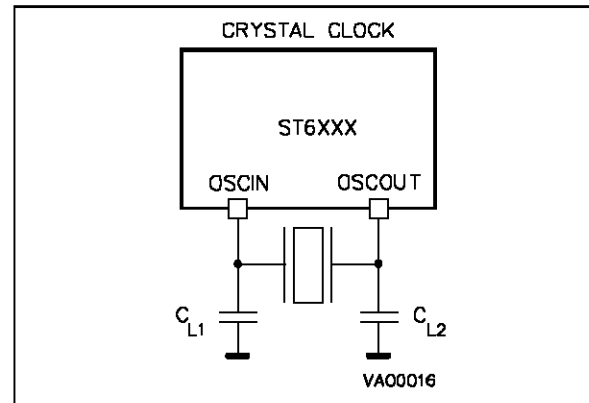


Figure 24. Clock Generator Option (2)

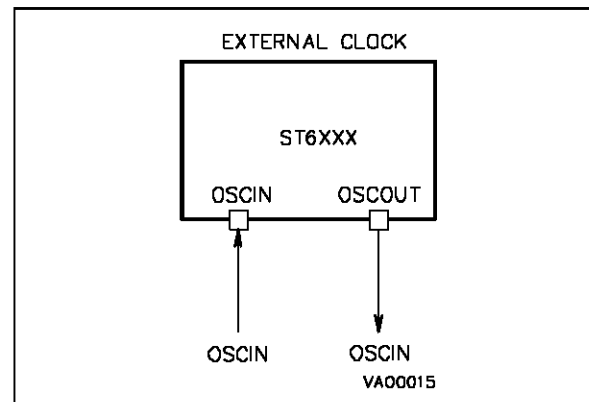
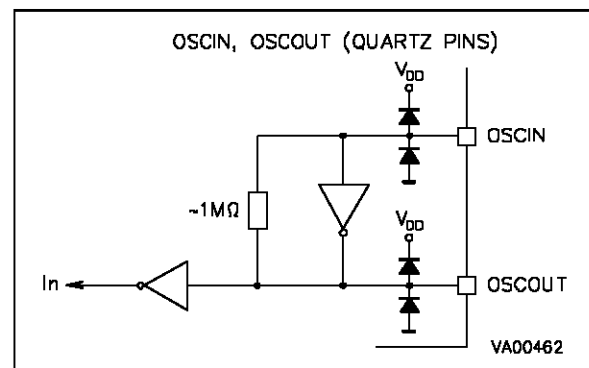


Figure 25. OSCIN, OSCOUT Diagram



INPUT/OUTPUT PORTS

The ST6369 microcontrollers use three standard I/O ports (A,B,C) with up to eight pins on each port; refer to the device pin configurations to see which pins are available.

Each line can be individually programmed either in the input mode or the output mode as follows by software.

- Output
- Input with on-chip pull-up resistor (selected by software)
- Input without on-chip pull-up resistor (selected by software)

Note: pins with 12V open-drain capability do not have pull-up resistors.

In output mode the following hardware configurations are available:

- Open-drain output 12V (PA4-PA7, PC4-PC7)
- Open-drain output 5V (PC0-PC3)
- Push-pull output (PA0-PA3, PB0-PB6)

The lines are organized in three ports (port A,B,C). The ports occupy 6 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 and Direction registers are associated with the PA0 line of Port A).

There are three Data registers (DRA, DRB, DRC), that 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 Data Direction Register, to select the different input mode options. Single-bit operations on I/O registers (bit set/reset instructions) are possible but care is necessary because reading in input mode is made from I/O pins and therefore might be influenced by the external load, while writing will directly affect the Port data register causing an undesired changes of the input configuration. The three Data Direction registers (DDRA, DDRB, DDRC) allow the selection of the direction of each pin (input or output).

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 is selected on all the pins thus avoiding pin conflicts (with the exception of PC2 that is set in output mode and is set high ie. high impedance).

Details of I/O Ports

When programmed as an input a pull-up resistor (if available) can be switched active under program control. When programmed as an output the I/O port will operate either in the push-pull mode or the open-drain mode according to the hardware fixed configuration as specified below.

Port A. PA0-PA3 are available as push-pull when outputs. PA4-PA7 are available as open-drain (no push-pull programmability) capable of withstanding 12V (no resistive pull-up in input mode). PA6-PA7 has been specially designed for higher driving capability and are able to sink 25mA with a maximum V_{OL} of 1V.

Port B. All lines are configured as push-pull when outputs.

Port C. PC0-PC3 are available as open-drain capable of withstanding a maximum $V_{DD}+0.3V$. PC4-PC7 are available as open-drain capable of withstanding 12V (no resistive pull-up in input mode). Some lines are also used as I/O buffers for signals coming from the on-chip SPI.

In this case the final signal on the output pin is equivalent to a wired AND with the programmed data output.

If the user needs to use the serial peripheral, the I/O line should be set in output mode while the open-drain configuration is hardware fixed; the corresponding data bit must set to one. If the latched interrupt functions are used (HSYNC, PWRIN) then the corresponding pins should be set to input mode.

On ST6369 the I/O pins with double or special functions are:

- PC0/SCL (connected to the SPI clock signal)
- PC1/SDA (connected to the SPI data signal)
- PC3/SEN (connected to the SPI enable signal)
- PC4/PWRIN (connected to the PWRIN interrupt latch)
- PC6/HSYNC (connected to the HSYNC interrupt latch)

All the Port A,B and C I/O lines have Schmitt-trigger input configuration with a typical hysteresis of 1V.

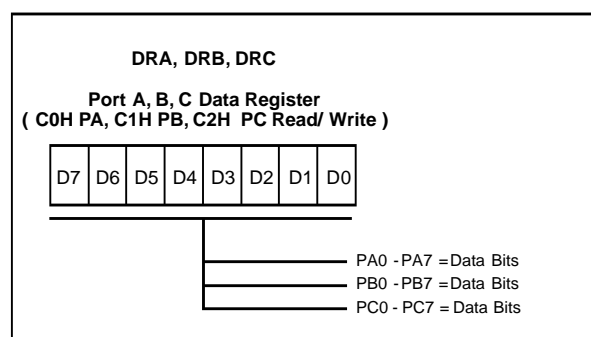
INPUT/OUTPUT PORTS (Continued)

Table 8. I/O Port Options Selection

DDR	DR	Mode	Option
0	0	Input	With on-chip pull-up resistor
0	1	Input	Without on-chip pull-up resistor
1	X	Output	Open-drain or Push-Pull

Note: X: Means don't care.

Figure 26. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data bits. Reset at power-on.

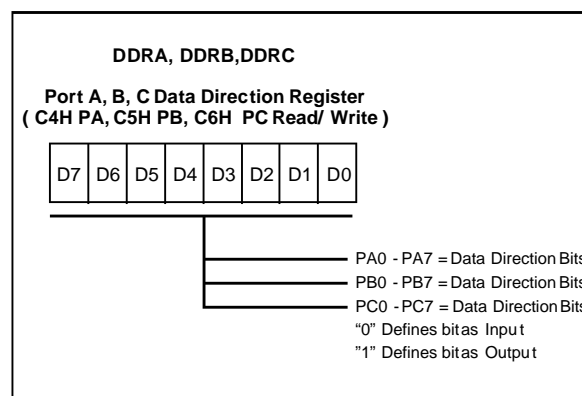
PB7-PB0. These are the I/O port B data bits. Reset at power-on.

PC7-PC0. Set to 04H at power-on. Bit 2 (PC2 pin) is set to one (open drain therefore high impedance).

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 to the relevant bit in the data (DR) and data direction register (DDR). Table 8 shows all the port configurations that can be selected by the user software.

Figure 27. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

PB7-PB0. These are the I/O port B data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

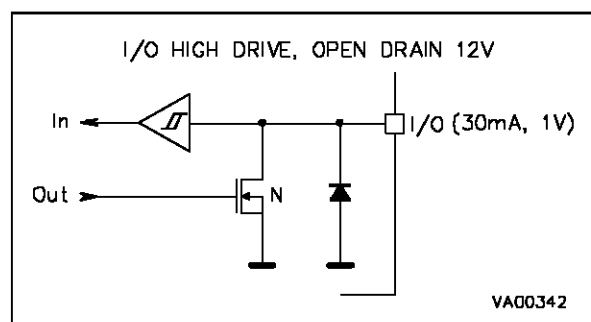
PC7-PC0. These are the I/O port C data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Set to 04H at power-on. Bit 2 (PC2 pin) is set to one (output mode selected).

INPUT/OUTPUT PORTS (Continued)

Input/Output Configurations

The following schematics show the I/O lines hardware configuration for the different options. Figure 28 shows the I/O configuration for an I/O pin with open-drain 12V capability (standard drive and high drive). Figure 29 shows the I/O configuration for an I/O pin with push-pull and with open drain 5V capability.

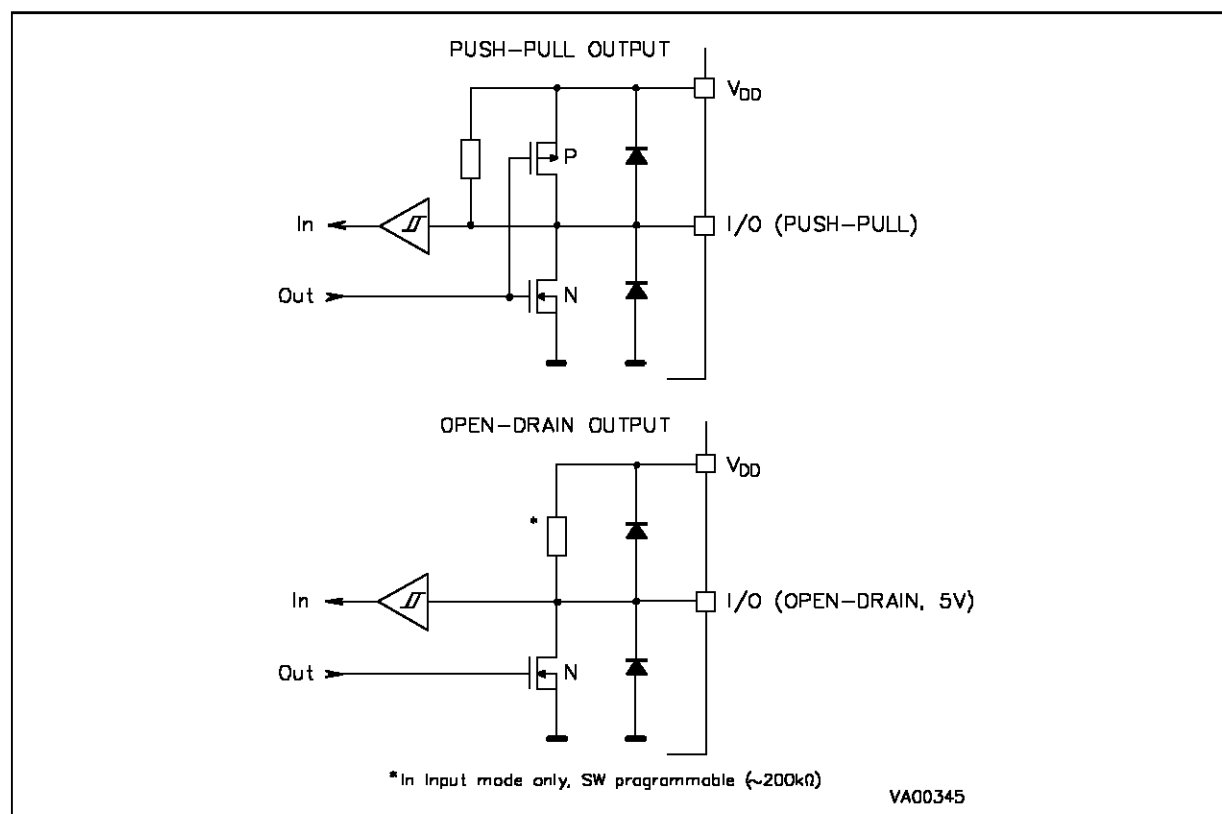
Figure 28. I/O Configuration Diagram (Open Drain 12V)

**Notes :**

The WAIT instruction allows the ST6369 to be used in situations where low power consumption is needed. This can only be achieved however if the I/O pins either are programmed as inputs with well defined logic levels or have no power consuming resistive loads in output mode. The unavailable I/O lines PB0, PB3 and PB7 should be programmed in output mode.

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

Figure 29. I/O Configuration Diagram (Open Drain 5V, Push-pull)



TIMERS

The ST6369 devices offer two on-chip Timer peripherals consisting of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15} , and a control logic that allows configuring the peripheral operating mode. Figure 30 shows the timer block diagram. The content of the 8-bit counters can be read/written in the Timer/Counter registers TCR that can be addressed in the data space as RAM location at addresses D3H (Timer 1) and DBH (Timer 2). The state of the 7-bit prescaler can be read in the PSC register at addresses D2H (Timer 1) and DAH (Timer 2). The control logic is managed by TSCR registers at D4H (Timer 1) and DCH (Timer 2) addresses as described in the following paragraphs.

The following description applies to both Timer 1 and Timer 2. The 8-bit counter is decremented by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (timer zero) bit in the TSCR is set to one. If the ETI (enable timer interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3 (for Timer 1) and #1 for Timer 2, is generated. The interrupt of the timer can be used to exit the MCU from the WAIT mode.

The prescaler decrements on rising edge. The prescaler input is the oscillator frequency divided by 12.

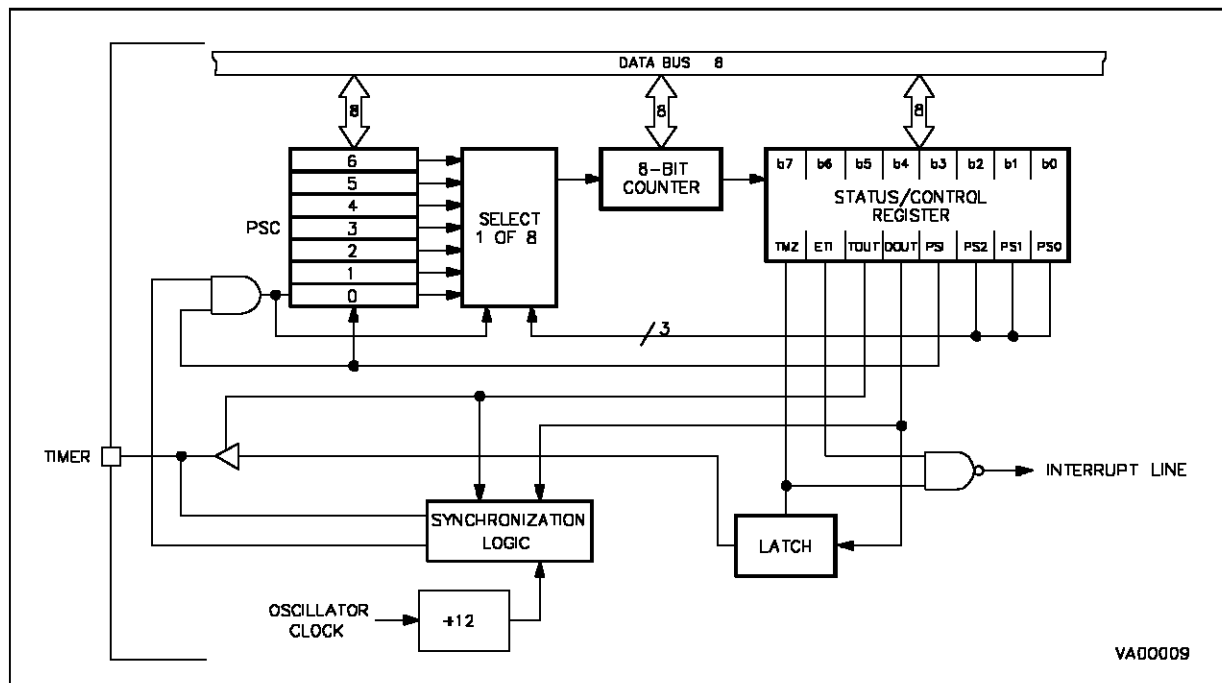
Depending on the division factor programmed by PS2/PS1/PS0 (see table 9) bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources.

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 TCR.

This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. On division factor 128, the MSB bit 6 of PSC is connected to clock input of TCR. The prescaler initialize bit (PSI) in the TSCR 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.

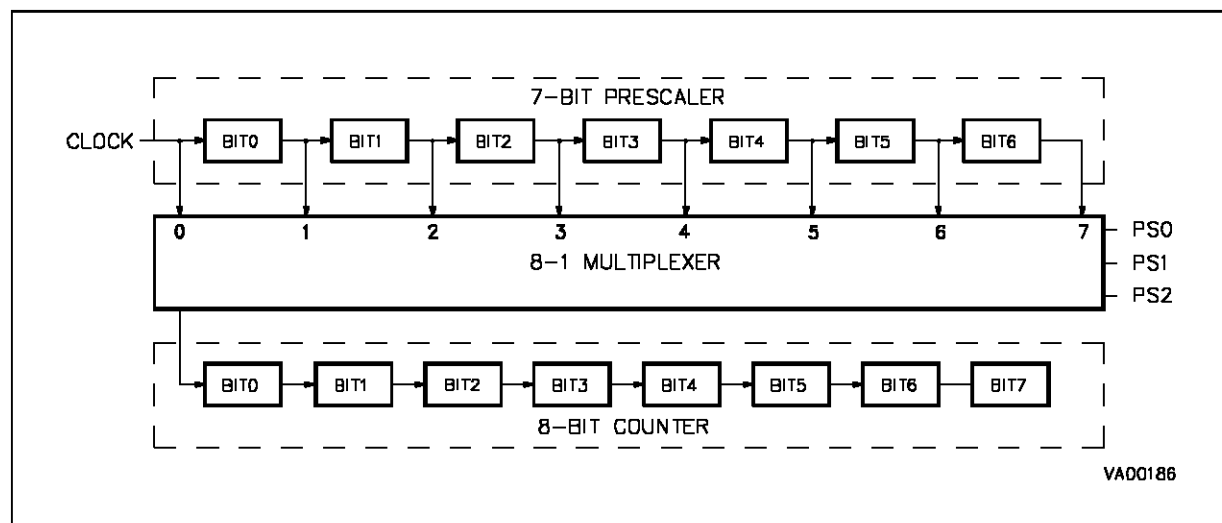
The prescaler can be given any value between 0 and 7FH by writing to the related register address, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2/PS1/PS0 bits in the control register. Figure 31 shows the timer working principle.

Figure 30. Timer Peripheral Block Diagram



TIMERS (Continued)

Figure 31. Timer Working Principle



Timer Operating Modes

As on ST6369 devices the external TIMER pin is not available the only allowed operating mode is the output mode that have to be selected by setting to 1 bit 4 and by clearing to 0 bit 5 in the TSCR1 register. This procedure will enable both Timer 1 and Timer 2.

Output Mode (TSCR1 D4 = 1, TSCR1 D5 = 0). On this mode the timer prescaler is clocked by the prescaler clock input (OSC/12). The user can select the desired prescaler division ratio through the PS2/PS1/PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR.

The TMZ bit can be tested under program control to perform timer functions whenever it goes high. Bit D4 and D5 on TSCR2 (Timer 2) register are not implemented.

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (enable timer interrupt) bit is set to one then an interrupt request associ-

ated to interrupt vector #3 (for Timer 1) and to interrupt vector #1 (for Timer 2) is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

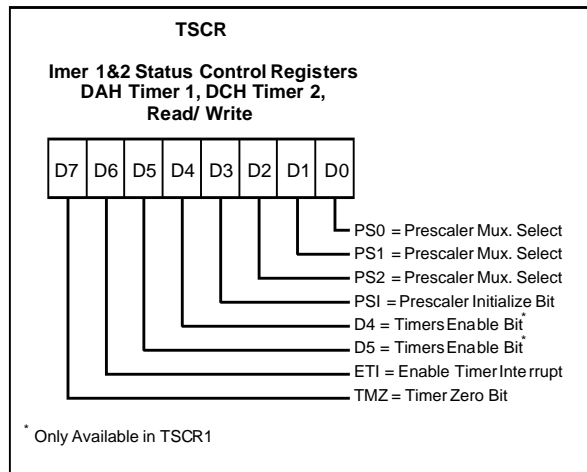
Notes :

TMZ is set when the counter reaches 00H ; however, it may be set by writing 00H in the TCR register or setting the bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFH while the 7-bit prescaler is loaded to 7FH, and the TSCR register is cleared which means that timer is stopped (PSI=0) and timer interrupt disabled.

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

TIMERS (Continued)

Figure 32. Timer Status Control Registers



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

ETI. This bit, when set, enables the timer interrupt (vector #3 for Timer 1, vector #1 for Timer 2) request. If ETI=0 the timer interrupt is disabled. If ETI= 1 and TMZ= 1 an interrupt request is generated.

D5. This is the timers enable bit D5. It must be cleared to 0 together with a set to 1 of bit D4 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D4. This is the timers enable bit D4. This bit must be set to 1 together with a clear to 0 of bit D5 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D5	D4	Timers
0	0	Disabled
0	1	Enabled
1	X	Reserved

PS1. Used to initialize the prescaler and inhibit its counting while 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-PS0. These bits select the division ratio of the prescaler register. (see table 9)

The TSCR1 and TSCR2 registers are cleared on reset. The correct D4-D5 combination must be written in TSCR1 by user's software to enable the operation of Timer 1 and Timer 2.

Table 9. Prescaler Division Factors

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

Figure 33. Timer Counter Registers

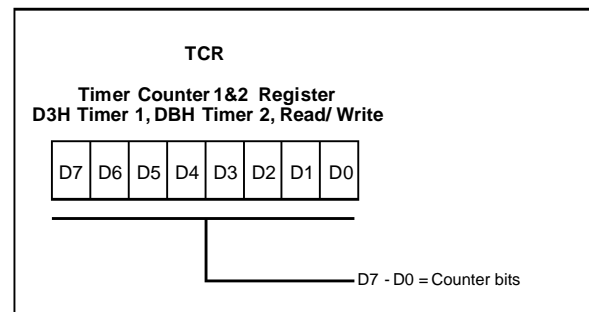
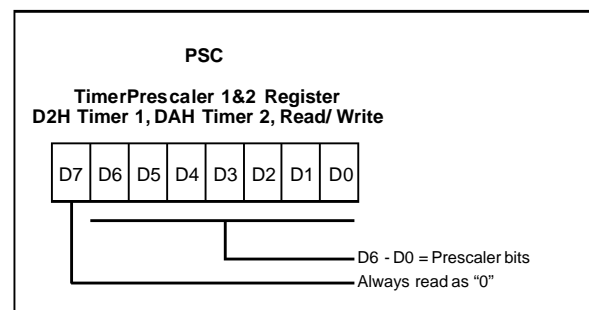


Figure 34. Timer Counter Registers



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION

The hardware activated digital watchdog function consists of a down counter that is automatically initialized after reset so that this function does not need to be activated by the user program. As the watchdog function is always activated this down counter can not be used as a timer. The watchdog is using one data space register (HWDR location D8H). The watchdog register is set to FEH on reset and immediately starts to count down, requiring no software start. Similarly the hardware activated watchdog can not be stopped or delayed by software.

The watchdog time can be programmed using the 6 MSbits in the watchdog register, this gives the possibility to generate a reset in a time between 3072 to 196608 oscillator cycles in 64 possible steps. (With a clock frequency of 8MHz this means from 384 μ s to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones.

The presence of the hardware watchdog deactivates the STOP instruction and a WAIT instruction is automatically executed instead of a STOP. Bit 1 of the watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero). Figure 35 shows the watchdog block diagram while Figure 36 shows its working principle.

Figure 36. Hardware Activated Watchdog Working Principle

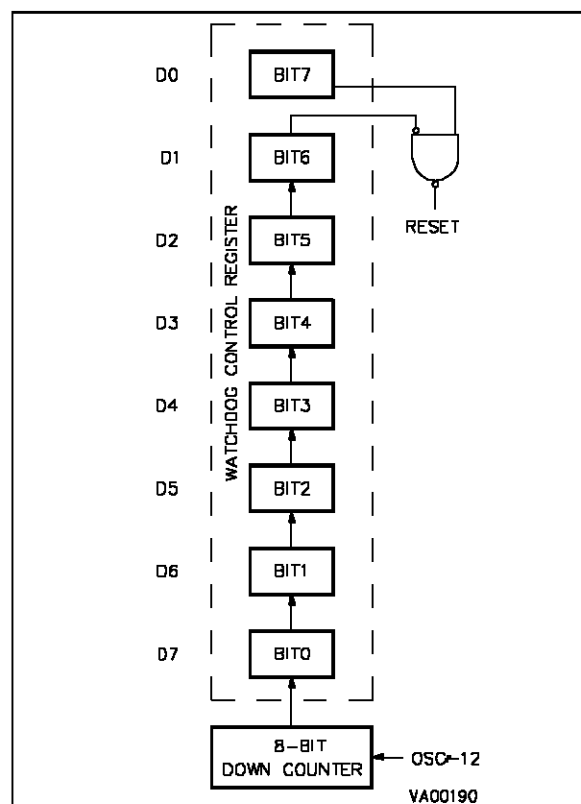
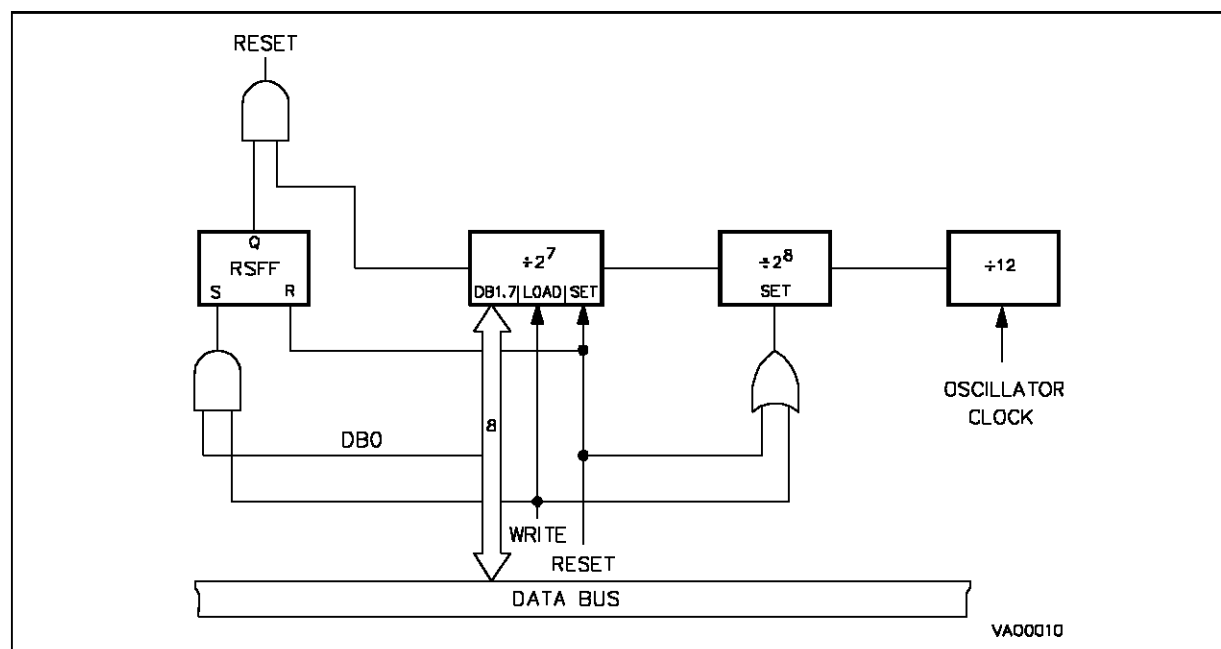
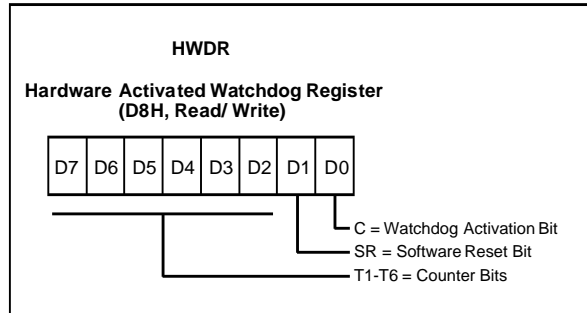


Figure 35. Hardware Activated Watchdog Block Diagram



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION (Continued)

Figure 37. Watchdog Register



T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter, these bits are in the opposite order to normal.

SR. This bit is set to one during the reset phase and will generate a software reset if cleared to zero.

C. This is the watchdog activation bit that is hardware set. The watchdog function is always activated independently of changes of value of this bit.

The register reset value is FEH (Bit 1-7 set to one, Bit 0 cleared).

SERIAL PERIPHERAL INTERFACE

The ST6369 Serial Peripheral Interface (SPI) has been designed to be cost effective and flexible in interfacing the various peripherals in TV applications.

It maintains the software flexibility but adds hardware configurations suitable to drive devices which require a fast exchange of data. The three pins dedicated for serial data transfer (single master only) can operate in the following ways:

- as standard I/O lines (software configuration)
- as S-BUS or as I²C BUS (two pins)
- as standard (shift register) SPI

When using the hardware SPI, a fixed clock rate of 62.5kHz is provided.

It has to be noted that the first bit that is output on the data line by the 8-bit shift register is the MSB.

SPI Data/Control Registers

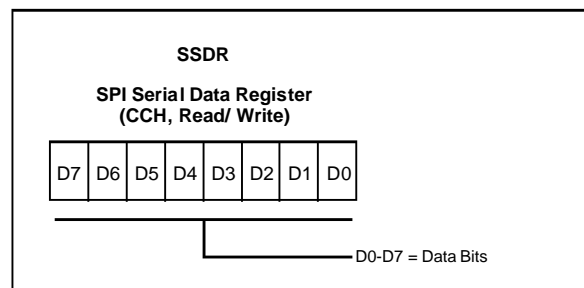
For I/O details on SCL (Serial Clock), SDA (Serial Data) and SEN (Serial Enable) please refer to I/O Ports description with reference to the following registers:

Port C data register, Address C2H (Read/Write).

- BIT D0 "SCL"
- BIT D1 "SDA"
- BIT D3 "SEN"

Port C data direction register, Address C6H (Read/Write).

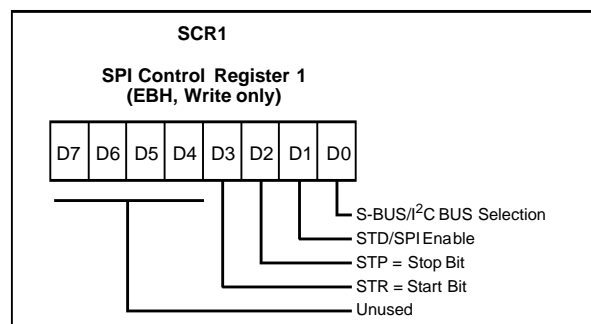
Figure 38. SPI Serial Data Register



D7-D0. These are the SPI data bits. They can be neither read nor written when SPI is operating (BUSY bit set). They are undefined after reset.

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 39. SPI Control Register 1



D7-D4. These bits are not used.

STR. This is Start bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Start generation, before beginning of transmission, is enabled. Set to zero after reset.

STP. This is Stop bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Stop condition generation is enabled. STP bit must be reset when standard protocol is used (this is also the default reset conditions). Set to zero after reset.

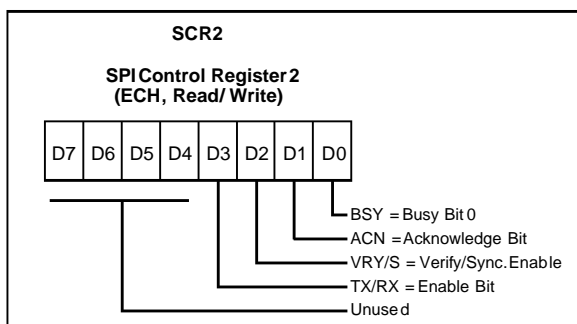
STD, SPI Enable. This bit, in conjunction with S-BUS/I²CBUS bit, allows the SPI disable and will select between I²CBUS/S-BUS and Standard shift register protocols. If this bit is set to one, it selects both I²CBUS and S-BUS protocols; final selection between them is made by S-BUS/I²CBUS bit. If this bit is cleared to zero when S-BUS/I²CBUS is set to "1" the Standard shift register protocol is selected. If this bit is cleared to "0" when S-BUS/I²CBUS is cleared to 0 the SPI is disabled. Set to zero after reset.

S-BUS/I²CBUS Selection. This bit, in conjunction with STD/SPI bit, allows the SPI disable and will select between I²CBUS and S-BUS protocols. If this bit is cleared to "0" when STD bit is also "0", the SPI interface is disabled. If this bit is cleared to zero when STD bit is set to "1", the I²CBUS protocol will be selected. If this bit is set to "1" when STD bit is set to "1", the S-BUS protocol will be selected. Cleared to zero after reset.

Table 10. SPI Modes Selection

D1 STD/SP	D0 S-BUS/I ² C BUS	SPI Function
0	0	Disabled
0	1	STD Shift Reg.
1	0	I ² C BUS
1	1	S-BUS

Figure 40. SPI Control Register2



D7-D4. These bits are not used.

TX/RX. Write Only. When this bit is set, current byte operation is a transmission. When it is reset, current operation is a reception. Set to zero after reset.

VRY/S. Read Only/Write Only. This bit has two different functions in relation to read or write operation. Reading Operation: when STD and/or TRX bits is cleared to 0, this bit is meaningless. When bits STD and TX are set to 1, this bit is set each time BSY bit is set. This bit is reset during byte operation if real data on SDA line are different from the output from the shift register. Set to zero after reset. Writing Operation: it enables (if set to one) or disables (if cleared to zero) the interrupt coming from VSYNC pin. Undefined after reset. Refer to OSD description for additional information.

ACN. Read Only. If STD bit (D1 of SCR1 register) is cleared to zero this bit is meaningless. When STD is set to one, this bit is set to one if no Acknowledge has been received. In this case it is automatically reset when BSY is set again. Set to zero after reset.

BSY. Read/Set Only. This is the busy bit. When a one is loaded into this bit the SPI interface start the transmission of the data byte loaded into SSDR data register or receiving and building the receive data into the SSDR data register. This is done in accordance with the protocol, direction and start/stop condition(s). This bit is automatically cleared at the end of the current byte operation. Cleared to zero after reset.

Note :

The SPI shift register is also the data transmission register and the data received register; this feature is made possible by using the serial structure of the ST6369 and thus reducing size and complexity.

SERIAL PERIPHERAL INTERFACE (Continued)

During transmission or reception of data, all access to serial data register is therefore disabled. The reception or transmission of data is started by setting the BUSY bit to "1"; this will be automatically reset at the end of the operation. After reset, the busy bit is cleared to "0", and the hardware SPI disabled by clearing bit 0 and bit 1 of SPI control register 1 to "0". The outputs from the hardware SPI are "ANDed" to the standard I/O software controlled outputs. If the hardware SPI is in operation the Port C pins related to the SPI should be configured as outputs using the Data Direction Register and should be set high. When the SPI is configured as the S-BUS, the three pins PC0, PC1 and PC3 become the pins SCL, SDA and SEN respectively. When configured as the I²C BUS the pins PC0 and PC1 are configured as the pins SCL and SDA; PC3 is not driven and can be used as a general purpose I/O pin. In the case of the STD SPI the pins PC0 and PC1 become the signals CLOCK and DATA, PC3 is not driven and can be used as general purpose I/O pin. The VERIFY bit is available when the SPI is configured as either S-BUS or I²C BUS. At the start of a byte transmission, the verify bit is set to one. If at any time during the transmission of the following eight bits, the data on the SDA line does not match the data forced by the SPI (while SCL is high), then the VERIFY bit is reset. The verify is available only during transmission for the S-BUS and I²C BUS; for other protocol it is not defined. The SDA and SCL signal entering the SPI are buffered in order to remove any minor glitches. When STD bit is set to one (S-BUS or I²C BUS selected), and TRX bit is reset (receiving data), and STOP bit is set (last byte of current communication), the SPI interface does not generate the Acknowledge, according to S-BUS/I²C BUS specifications. PC0-SCL, PC1-SDA and PC3-SEN lines are standard drive I/O port pins with open-drain output configuration (maximum voltage that can be applied to these pins is $V_{DD} + 0.3V$).

S-BUS/I²C BUS Protocol Information

The S-BUS is a three-wire bidirectional data-bus with functional features similar to the I²C BUS. In fact the S-BUS includes decoding of Start/Stop conditions and the arbitration procedure in case of multimaster system configuration (the ST6369 SPI allows a single-master only operation). The SDA line, in the I²C BUS represents the AND combination of SDA and SEN lines in the S-BUS. If the SDA and the SEN lines are short-circuit connected, they appear as the SDA line of the I²C BUS. The Start/Stop conditions are detected (by the external peripherals suited to work with S-BUS/I²C BUS) in the following way:

- On S-BUS by a transition of the SEN line (1 to 0 Start, 0 to 1 Stop) while the SCL line is at high level.
- On I²C BUS by a transition of the SDA line (10 Start, 01 Stop) while the SCL line is at high level.

Start and Stop condition are always generated by the master (ST6369 SPI can only work as single master). The bus is busy after the start condition and can be considered again free only when a certain time delay is left after the stop condition. In the S-BUS configuration the SDA line is only allowed to change during the time SCL line is low. After the start information the SEN line returns to high level and remains unchanged for all the data transmission time. When the transmission is completed the SDA line is set to high level and, at the same time, the SEN line returns to the low level in order to supply the stop information with a low to high transition, while the SCL line is at high level. On the S-BUS, as on the I²C BUS, each eight bit information (byte) is followed by one acknowledged bit which is a high level put on the SDA line by the transmitter. A peripheral that acknowledges has to pull down the SDA line during the acknowledge clock pulse. An addressed receiver has to generate an acknowledge after the reception of each byte; otherwise the SDA line remains at the high level during the ninth clock pulse time. In this case the master transmitter can generate the Stop condition, via the SEN (or SDA in I²C BUS) line, in order to abort the transfer.

SERIAL PERIPHERAL INTERFACE (Continued)

Start/Stop Acknowledge. The timing specs of the S-BUS protocol require that data on the SDA (only on this line for I²CBUS) and SEN lines be stable during the “high” time of SCL. Two exceptions to this rule are foreseen and they are used to signal the start and stop condition of data transfer.

- On S-BUS by a transition of the SEN line (10 Start, 01 Stop) while the SCL line is at high level.
- On I²CBUS by a transition of the SDA line (10 Start, 01 Stop) while the SCL line is at high level.

Data are transmitted in 8-bit groups; after each group, a ninth bit is interposed, with the purpose of acknowledging the transmitting sequence (the transmit device place a “1” on the bus, the acknowledging receiver a “0”).

Interface Protocol. This paragraph deals with the description of data protocol structure. The interface protocol includes:

- A start condition
- A “slave chip address” byte, transmitted by the master, containing two different information:
 - a. the code identifying the device the master wants to address (this information is present in the first seven bits)
 - b. the direction of transmission on the bus (this information is given in the 8th bit of the byte); “0” means “Write”, that is from the master to the slave, while “1” means “Read”. The addressed slave must always acknowledge.

The sequence from, now on, is different according to the value of R/W bit.

1. $R/\overline{W} = “0”$ (\overline{Write})

In all the following bytes the master acts as transmitter; the sequence follows with:

- a. an optional data byte to address (if needed) the slave location to be written (it can be a word address in a memory or a register address, etc.).
- b. a “data” byte which will be written at the address given in the previous byte.
- c. further data bytes.
- d. a STOP condition

A data transfer is always terminated by a stop condition generated from the master. The ST6369 peripheral must finish with a stop condition before another start is given. Figure 44 shows an example of write operation.

2. $R/\overline{W} = “1”$ (Read)

In this case the slave acts as transmitter and, therefore, the transmission direction is changed. In read mode two different conditions can be considered:

- a. The master reads slave immediately after first byte. In this case after the slave address sent from the master with read condition enabled the master transmitter becomes master receiver and the slave receiver becomes slave transmitter.
- b. The master reads a specified register or location of the slave. In this case the first sent byte will contain the slave address with write condition enabled, then the second byte will specify the address of the register to be read. At this moment a new start is given together with the slave address in read mode and the procedure will proceed as described in previous point “a”.

The diagram illustrates the bit sequence for a 2-wire serial interface. The sequence of bits is: S (Start), SLAVE ADDRESS, 0 (R/W bit), A (Acknowledge), WORD ADDRESS, A (Acknowledge), DATA, A (Acknowledge), and P (Stop). The R/W bit is 0 for read and 1 for write. The A bit is used for acknowledgment from the slave. The P bit is the stop bit. The diagram also shows the timing of the interface, with START, R/W, and STOP signals. The R/W signal is active-low, indicated by a bar over the W. The START and STOP signals are active-low, indicated by a bar over the S and P respectively. The diagram also shows the timing of the interface, with START, R/W, and STOP signals. The R/W signal is active-low, indicated by a bar over the W. The START and STOP signals are active-low, indicated by a bar over the S and P respectively.

The diagram illustrates the structure of a two-byte I2C data frame. The frame is represented as a horizontal sequence of fields: **S** (Start), **SLAVE ADDRESS**, **1** (R/W bit), **A** (Acknowledge), **DATA**, **A** (Acknowledge), **DATA**, **1** (R/W bit), and **P** (Stop). Above the frame, three acknowledgment signals are shown: **ACKNOWLEDGE FROM SLAVE** pointing to the first **A** field, **ACKNOWLEDGE FROM MASTER** pointing to the second **A** field, and **NO ACKNOWLEDGE FROM MASTER** pointing to the **1** field before the final **P** field. Below the frame, a double-headed arrow labeled **n BYTES** spans the length of the two data fields. The **START** signal is indicated by an upward arrow at the beginning of the **S** field, and the **STOP** signal is indicated by an upward arrow at the end of the **P** field. The **R/W** label is positioned below the **1** field, indicating its function as the Read/Write bit.

Timing diagram for a 16-bit data transfer. The diagram shows two horizontal timelines. The top timeline represents the address phase: it starts with a START signal, followed by a Slave Address, then a bit '0' (indicated by an $\overline{R/W}$ signal), then a bit 'A' (indicated by an ACKNOWLEDGE FROM SLAVE signal), followed by a Word Address, then a bit 'A' (indicated by an ACKNOWLEDGE FROM SLAVE signal), and ends with a STOP signal. The bottom timeline represents the data phase: it starts with a START signal, followed by a Slave Address, then a bit '1' (indicated by an $\overline{R/W}$ signal), then a bit 'A' (indicated by an ACKNOWLEDGE FROM SLAVE signal), followed by a Data field (indicated by an ACKNOWLEDGE FROM MASTER signal), then another Data field (indicated by an ACKNOWLEDGE FROM MASTER signal), then a bit '1' (indicated by a NO ACKNOWLEDGE FROM MASTER signal), and ends with a STOP signal. Labels 'MSB' are placed above the first and second data fields.

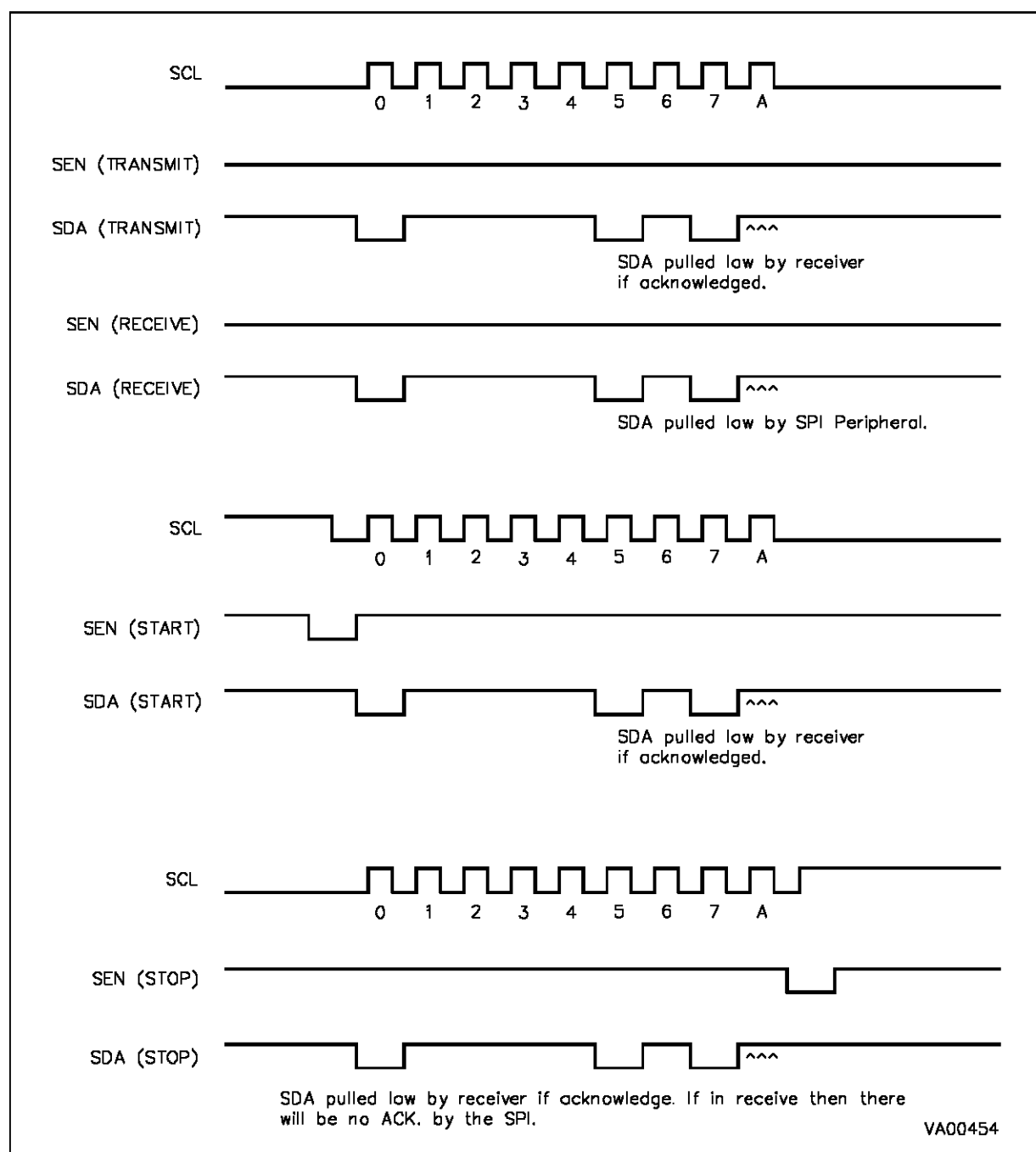
SERIAL PERIPHERAL INTERFACE (Continued)

S-BUS/I²C BUS Timing Diagrams

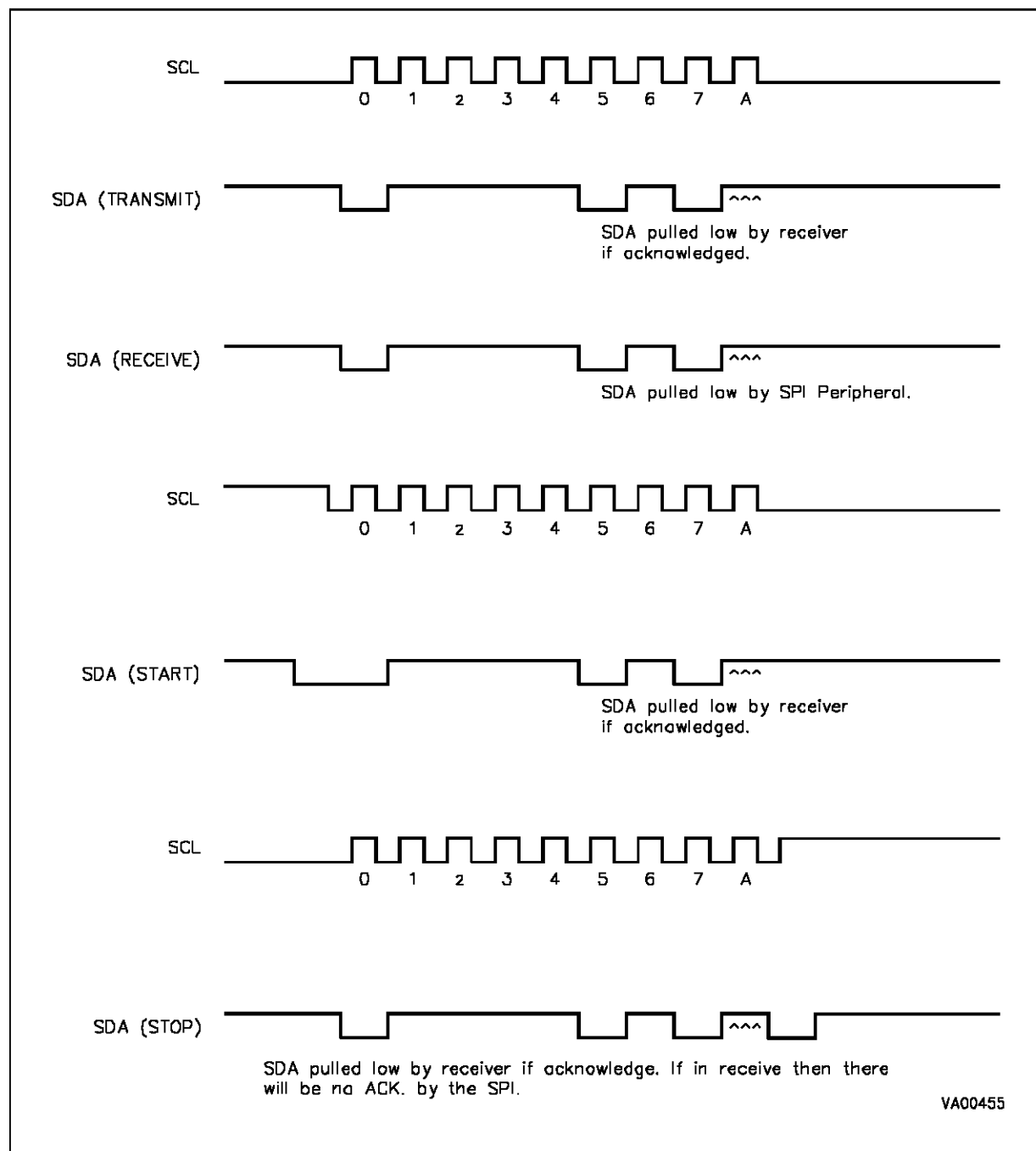
The clock of the S-BUS/I²C BUS of the ST6369 SPI (single master only) has a fixed bus clock frequency of 62.5KHz. All the devices connected to the bus must be able to follow transfers with fre-

quencies up to 62.5KHz, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure which will force the master into a wait state and stretch low periods.

Figure 44. S-BUS Timing Diagram



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 45. I²C BUS Timing Diagram

Note: The third pin, SEN, should be high; it is not used in the I²C BUS. Logically SDA is the AND of the S-BUS SDA and SEN.

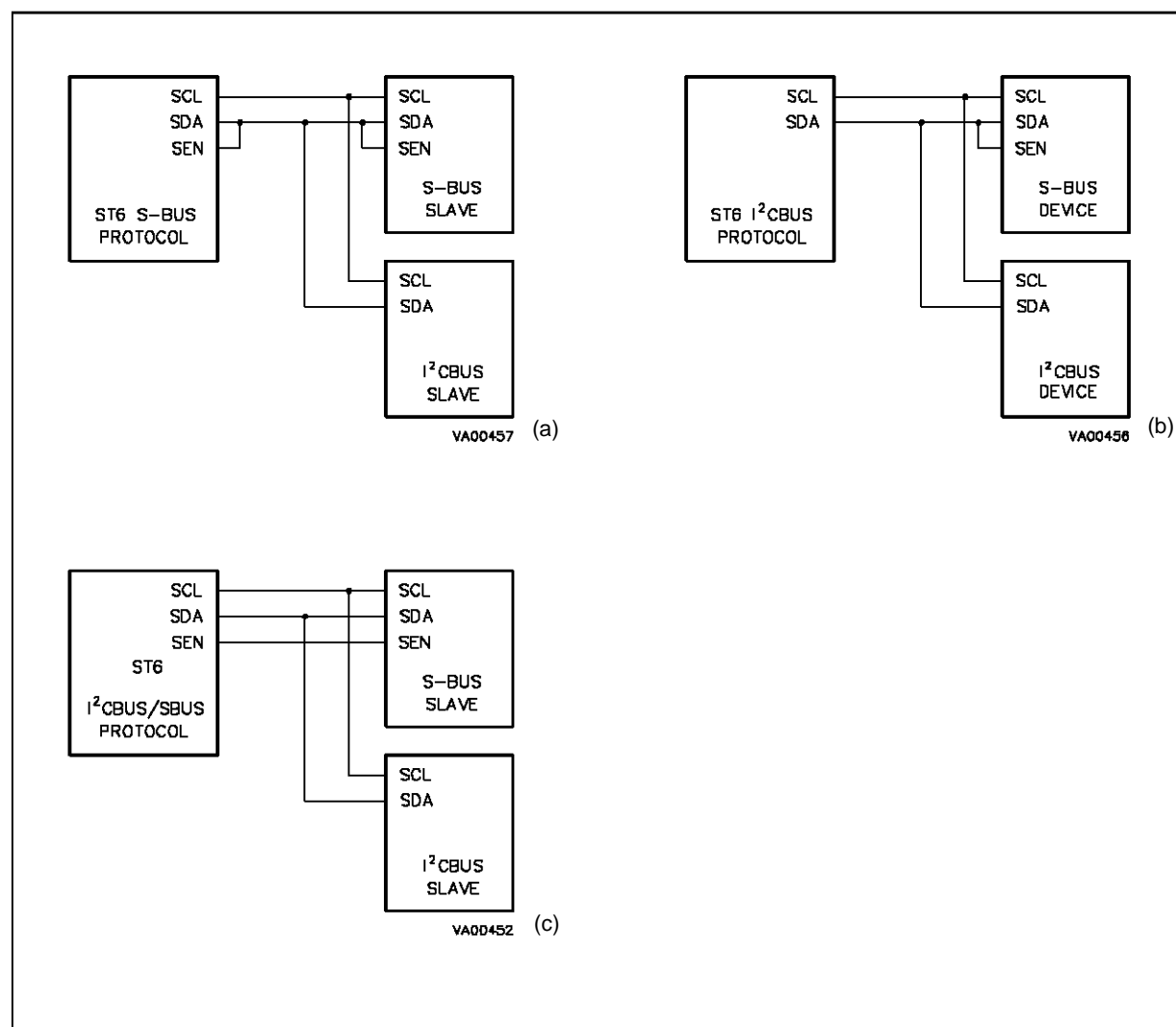
SERIAL PERIPHERAL INTERFACE (Continued)

Compatibility S-BUS/I²CBUS

Using the S-BUS protocol it is possible to implement mixed system including S-BUS/I²CBUS bus peripherals. In order to have the compatibility with the I²CBUS peripherals, the devices including the S-BUS interface must have their SDA and SEN pins connected together as shown in the following Figure 46 (a and b).

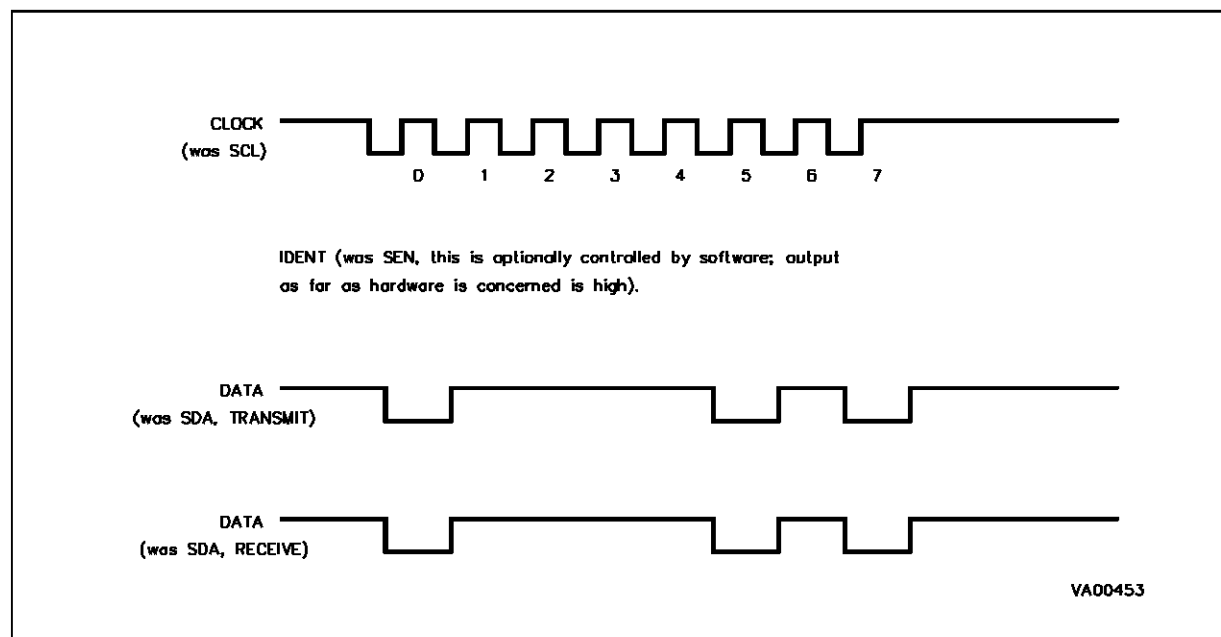
It is also possible to use mixed S-BUS/I²CBUS protocols as showed in Figure 46 (c). S-BUS peripherals will only react to S-BUS protocol signals, while I²CBUS peripherals will only react to I²CBUS signals. Multimaster configuration is not possible with the ST6369 SPI (single master only).

Figure 46. S-BUS/I²C BUS Mixed Configurations



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 47. Software Bus (Hardware Bus Disabled) Timing Diagram

**STD SPI Protocol (Shift Register)**

This protocol is similar to the I^2C BUS with the exception that there is no acknowledge pulse and there are no stop or start bits. The clock cannot be slowed down by the external peripherals.

In this case all three outputs should be high in order not to lock the software I/Os from functioning.

SPI Standard Bus Protocol: The standard bus protocol is selected by loading the SPI Control

Register 1 (SCR1 Add. EBH). Bit 0 named I^2C must be set at one and bit 1 named STD must be reset. When the standard bus protocol is selected bit 2 of the SCR1 is meaningless.

This bit named STOP bit is used only in I^2C BUS or SBUS. However take care that THE STOP BIT MUST BE RESET WHEN THE STANDARD PROTOCOL IS USED. This bit is set to ZERO after RESET.

14-BIT PWM D/A CONVERTER

The ST6369 PWM D/A CONVERTER (HDA) is composed of a 14-bit counter that allows the conversion of the digital content in an analog voltage, available at the HDA output pin, by using Pulse Width Modification (PWM), and Bit Rate Multiplier (BRM) techniques.

The tuning word consists of a 14-bit word contained in the registers HDADATA1 (location 0EEH) and HDADATA2 (location 0EFH). Coarse tuning (PWM) is performed using the seven MSBs, while fine tuning (BRM) is performed using the data in the seven LSBs. With all zeros loaded the output is zero; as the tuning voltage increases from all zeros, the number of pulses in one period increases to 128 with all pulses being the same width. For values larger than 128, the PWM takes over and the number of pulses in one period remains constant at 128, but the width changes. At the other end of the scale, when almost all ones are loaded, the pulses will start to link together and the number of pulses will decrease. When all ones are loaded, the output will be almost 100% high but will have a low pulse (1/16384 of the high pulse).

Output Details

Inside the on-chip D/A CONVERTER are included the register latches, a reference counter, PWM and BRM control circuitry. In the ST6369 the clock for the 14-bit reference counter is 2MHz derived from the 8MHz system clock. From the circuit point of view, the seven most significant bits control the coarse tuning, while the seven least significant bits control the fine tuning. From the application and software point of view, the 14 bits can be considered as one binary number.

As already mentioned the coarse tuning consists of a PWM signal with 128 steps; we can consider the fine tuning to cover 128 coarse tuning cycles. The addition of pulses is described in the following Table.

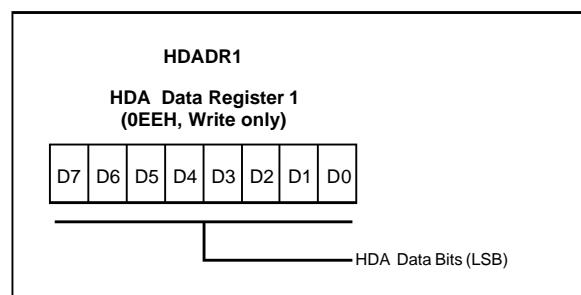
Table 11. Fine Tuning Pulse Addition

Fine Tuning (7 LSB)	N° of Pulses added at the following cycles (0...127)
0000001	64
0000010	32, 96
0000100	16, 48, 80, 112
0001000	8, 24, ..., 104, 120
0010000	4, 12, ..., 116, 124
0100000	2, 6, ..., 122, 126
1000000	1, 3, ..., 125, 127

The HDA output pin has a standard drive push-pull output configuration.

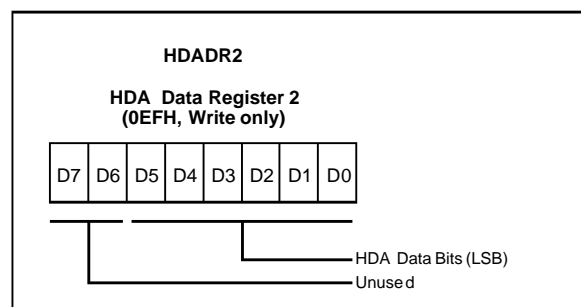
HDA Tuning Cell Registers

Figure 48. HDA Data Register 1



D7-D0. These are the 8 least significant HDA data bits. Bit 0 is the LSB. This register is undefined on reset.

Figure 49. HDA Data Register 2



D7-D6. These bits are not used.

D5-D0. These are the 6 most significant HDA data bits. Bit 5 is the MSB. This register is undefined on reset.

6-BIT PWM D/A CONVERTERS

The D/A macrocell contains up to six PWM D/A outputs (31.25kHz repetition, DA0-DA5) with six bit resolution.

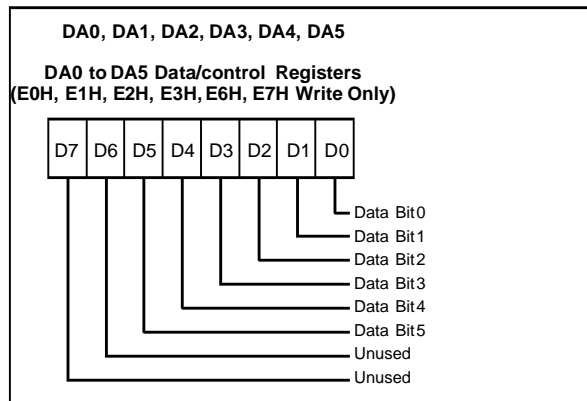
Each D/A converter of ST6369 is composed by the following main blocks:

- pre-divider
- 6-bit counter
- data latches and compare circuits

The pre-divider uses the clock input frequency (8MHz typical) and its output clocks the 6-bit free-running counter. The data latched in the six registers (E0H, E1H, E2H, E3H, E6H and E7H) control the six D/A outputs (DA0, 1, 2, 3, 4 and 5). When all zeros are loaded the relevant output is an high logic level; all 1's correspond to a pulse with a 1/64 duty cycle and almost 100% zero level.

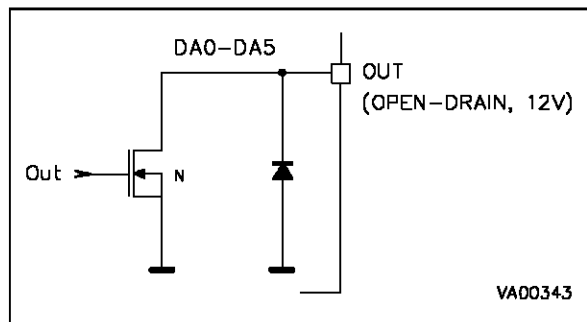
The repetition frequency is 31.25KHz and is related to the 8MHz clock frequency. Use of a different oscillator frequency will result in a different repetition frequency. All D/A outputs are open-drain with standard current drive capability and able to withstand up to 12V.

Figure 50. DA0-DA5 Data/Control Registers



DA0-DA5. These are the 6 bits of the PWM digital to analog converter. Undefined after reset.

Figure 51. 6-bit PWM D/A Output Configuration



A/D COMPARATOR

A/D INPUT, HSYNC/PC6 RESULT, VSYNC RESULT AND O0, O1 OUTPUTS

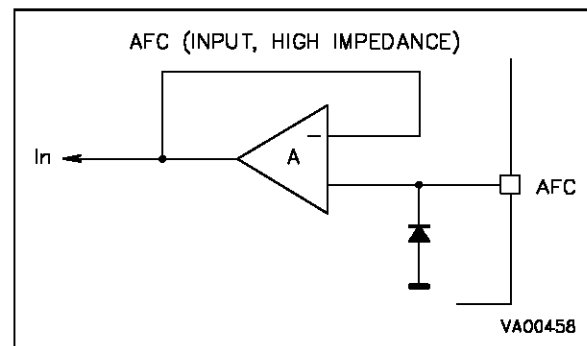
The A/D macrocell contains an A/D comparator with five levels at intervals of 1V from 1V to 5V. The levels can all be lowered by 0.5V to effectively double the resolution.

The A/D used to perform the AFC function (when high threshold is selected) has the following voltage levels: 1, 2, 3, 4 and 5V. Bits 0-2 of AFC result register (E4H address) will provide the result in binary form (less than 1V is 000, greater than 5V is 101).

If the application requires a greater resolution, the sensitivity can be doubled by clearing to zero bit 2 of the OUTPUTS control register, address E5H. In this case all levels are shifted lower by 0.5V. If the two results are now added within a software routine then the A/D S-curve can be located within a resolution of 0.5V.

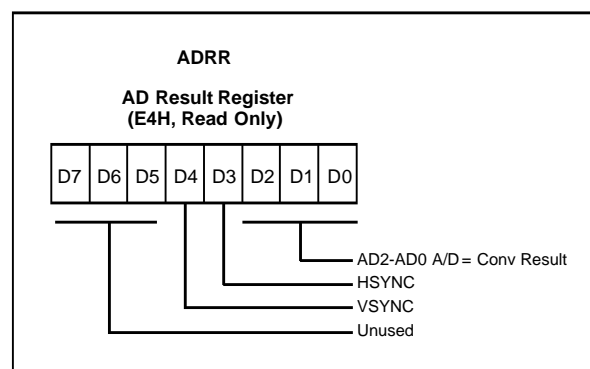
The A/D input has high impedance able to withstand up to 13V signals (input level tolerances $\pm 200\text{mV}$ absolute and $\pm 100\text{mV}$ relative to 5V).

Figure 52. A/D Inputs Configuration Diagram



A/D COMPARATOR (Continued)

Figure 53. A/D, HSYNC and VSYNC Result Register



D7-D5. These bits are not used.

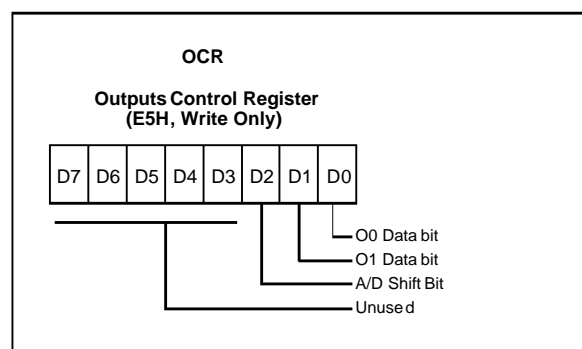
VSYNC. This bit reads the status of the VSYNC pin. It is inverted with respect to the pin.

HSYNC. This bit reads the status of the HSYNC latch. If a signal has been latched this bit will be high.

AD2-AD0. These bits store the real time conversion of the value present on the AD input pin. Undefined reset value.

D7, D6, D5, D4, D3. These bits are not used.

Figure 54. Outputs Control Register



A/D Shift. This bit determines the voltage range of the AFC input. Writing a zero will select the 0.5V to 4.5V range. Writing a one will select the 1.0V to 5.0V range. Undefined after reset.

O1,O0. These bits control the output pins O1,O0. They are undefined after reset.

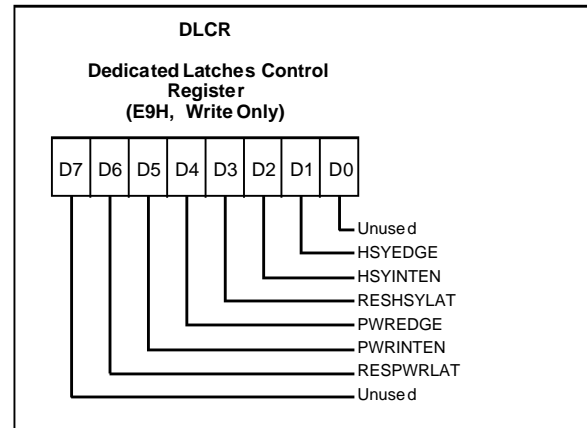
DEDICATED LATCHES

Two latches are available which may generate interrupts to the ST6369 core.

The HSYNC latch is set either by the falling or rising edge of the signal on pin PC6(HSYNC). If bit 1 (HSYEDGE) of the latches register (E9H) is high, then the latch will be triggered on the rising edge of the signal at PC6(HSYNC). If bit 1 (HSYEDGE) is low, then the latch will be triggered on the falling edge of the signal at PC6(HSYNC). The HSYNC latch can be reset by setting bit 3 (RESHSYLAT) of the latches register; the bit is set only and a high should be written every time the HSYNC latch needs to be reset. If bit 2 (HSYINTEN) of the latches register (E9H) is high, then the output of the HSYNC latch, HSYNCN, may generate an interrupt (#0). HSYNCN is inverted with respect to the state of the HSYNC latch. If bit 2 (HSYINTEN) is low, then the output of the HSYNC latch, HSYNCN, is forced high. The state of the HSYNC latch may be read from bit 3 (HSYNC) of register E4H; if the HSYNC latch is set, then bit 3 will be high.

The PWR latch is set either by the falling or rising edge of the signal on pin PC4(PWRIN). If bit 4 (PWREDGE) of the latches register (E9H) is high, then the latch will be triggered on the rising edge of the signal at PC4(PWRIN). If bit 4 (PWREDGE) is low, then the latch will be triggered on the falling edge of the signal at PC4(PWRIN). The PWR latch can be reset by setting bit 6 (RESPWRLAT) of the latches register; the bit is set only and a high should be written every time the PWR latch needs to be reset. If bit 5 (PWRINTEN) of the latches register (E9H) is high, then the output of the PWR latch, PWRINTN, may generate an interrupt (#4). PWRINTN is inverted with respect to the state of the PWR latch. If bit 5 (PWRINTEN) is low, then the output of the PWR latch, PWRINTN, is forced high.

Figure 55. Dedicated Latches Control Register



D0. This bit is not used

D7. This bit is not used

RESPWRLAT. Resets the PWR latch; this bit is set only.

PWRINTEN. This bit enables the PWRINTN signal (#4) from the latch to the ST6369 core. Undefined after reset.

PWREDGE. The bit determines the edge which will cause the PWRIN latch to be set. If this bit is high, then the PWRIN latch will be set on the rising edge of the PWRIN signal. Undefined after reset.

RESHSYLAT. Resets the HSYNC latch; this bit is set only.

HSYINTEN. This bit enables the HSYNCN signal (#0) from the latch to the ST6369 core. Undefined after reset.

HSYEDGE. The bit determines the edge which will cause the HSYNC latch to be set. If this bit is high, then the HSYNC latch will be set on the rising edge of the HSYNC signal. Undefined after reset.

SOFTWARE DESCRIPTION

The ST6369 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 ST6369 Core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST6369 Core has 9 addressing modes which are described in the following paragraphs. The ST6369 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 sub-routines 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 ST6369 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. See Table 12.

Table 12. 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. See Table 13.

Table 13. 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 rr	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. See Table 14.

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. See Table 15.

Control Instructions. The control instructions control the MCU operations during program execution. See Table 16.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space. Refer to Table 17.

Table 14. 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 rr. Data space register

e. 5 bit signed displacement in the range -15 to +16

ee. 8 bit signed displacement in the range -126 to +129

Δ . Affected
* . Not Affected

Table 15. 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;

* . Not Affected

rr. Data space register;

Table 16. 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 hardware activated watchdog function is selected.

Δ . Affected

* . Not Affected

Table 17. 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)

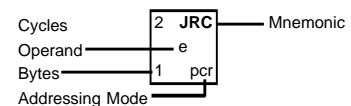
Low Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Low Hi
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 pcr	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 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 pcr	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 pcr	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 pcr	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 pcr	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 pcr	4 ADD 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 pcr	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 pcr	#	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 pcr	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 pcr	#	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 pcr	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 pcr	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 pcr	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 pcr	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 pcr	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 pcr	#	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 1 byte dataspace address
nn 1 byte immediate data
abc 12 bit address
ee 8 bit Displacement



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 than 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, T_j , in Celsius can be obtained from :

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

Where : T_A = Ambient Temperature.

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

PD = $P_{int} + P_{port}$.

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

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

Symbol	Parameter	Value	Unit	
V_{DD}	Supply Voltage	-0.3 to 7.0	V	
V_I	Input Voltage (AD IN)	V	$V_{SS} - 0.3$ to +13	V
V_I	Input Voltage (Other Inputs)	V	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)		$V_{SS} - 0.3$ to +13	V
V_O	Output Voltage (Other Outputs)	V	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
I_O	Current Drain per Pin Excluding V_{DD} , V_{SS} , PA6, PA7		± 10	mA
I_O	Current Drain per Pin (PA6, PA7)		± 50	mA
I_{VDD}	Total Current into V_{DD} (source)	50	mA	
I_{VSS}	Total Current out of V_{SS} (sink)	150	mA	
T_j	Junction Temperature	150	°C	
T_{STG}	Storage Temperature	-60 to 150	°C	

Note : 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.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
R_{thJA}	Thermal Resistance	PSDIP42	67	°C/W		

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version	0		70	°C
V_{DD}	Operating Supply Voltage	4.5 5.0 6.0		V		
f_{osc}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

($T_A = 0$ to $+70^\circ\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins	$0.2 \times V_{DD}$		V_{DD}	V
V_{IH}	Input High Level Voltage	All I/O Pins	$0.8 \times V_{DD}$			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins $V_{DD} = 5V$		1.0		V
V_{OL}	Low Level Output Voltage	DA0-DA5, PB1-PB2, PB4-PB6, PC0-PC7, O0, O1, PA0-PA5 $V_{DD} = 4.5V$ $I_{OL} = 1.6mA$ $I_{OL} = 5.0mA$			0.4	V
					1.0	V
V_{OL}	Low Level Output Voltage	PA6-PA7 $V_{DD} = 4.5V$ $I_{OL} = 1.6mA$ $I_{OL} = 25mA$			0.4	V
					1.0	V
V_{OL}	Low Level Output Voltage	OSCOUT $V_{DD} = 4.5V$ $I_{OL} = 0.4mA$			0.4	V
V_{OL}	Low Level Output Voltage	HDA Output $V_{DD} = 4.5V$ $I_{OL} = 0.5mA$ $I_{OL} = 1.6mA$			0.4	V
					1.0	V
V_{OH}	High Level Output Voltage	PB1-PB2, PB4-PB6, PA0-PA3, $V_{DD} = 4.5V$ $I_{OH} = -1.6mA$	4.1			V
V_{OH}	High Level Output Voltage	OSCOUT, $V_{DD} = 4.5V$ $I_{OH} = -0.4mA$	4.1			V
V_{OH}	High Level Output Voltage	HDA Output $V_{DD} = 4.5V$ $I_{OH} = -0.5mA$	4.1			V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB1-PB2, PB4-PB6, PA0-PA3, PC0-PC3 $V_{IN} = V_{SS}$	-100	-50	-25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCIN $V_{IN} = V_{SS}$ $V_{IN} = V_{DD}$	-10 0.1	-1 1	-0.1 10	μA
I_{IL}	Input Pull-down current in Reset	OSCIN	100			μA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up $V_{IN} = V_{DD}$ or V_{SS}	-10		10	μA

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{DDRAM}	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up $V_{IN} = V_{SS}$	- 50	- 30	- 10	μA
I_{IL} I_{IH}	Input Leakage Current	AD Pin $V_{IH} = V_{DD}$ $V_{IL} = V_{SS}$ $V_{IH} = 12.0V$	- 1		1 40	μA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 $V_{OH} = V_{DD}$			10	μA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 $V_{OH} = 12V$			40	μA
I_{DD}	Supply Current RUN Mode	$f_{OSC} = 8MHz$, $I_{Load} = 0mA$ $V_{DD} = 6.0V$		6	16	mA
I_{DD}	Supply Current WAIT Mode	$f_{OSC} = 8MHz$, $I_{Load} = 0mA$ $V_{DD} = 6V$		3	10	mA
I_{DD}	Supply Current at transition to RESET	$f_{OSC} = \text{Not App.}$, $I_{Load} = 0mA$ $V_{DD} = 6V$		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin	$0.3xV_{DD}$			V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	$0.8xV_{DD}$			V
V_{TA}	Input Level Absolute Tolerance	AD Pin $V_{DD} = 5V$			± 200	mV
V_{TR}	Input Level Relative Tolerance (1)	AD Pin Relative to other levels $V_{DD} = 5V$			± 100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

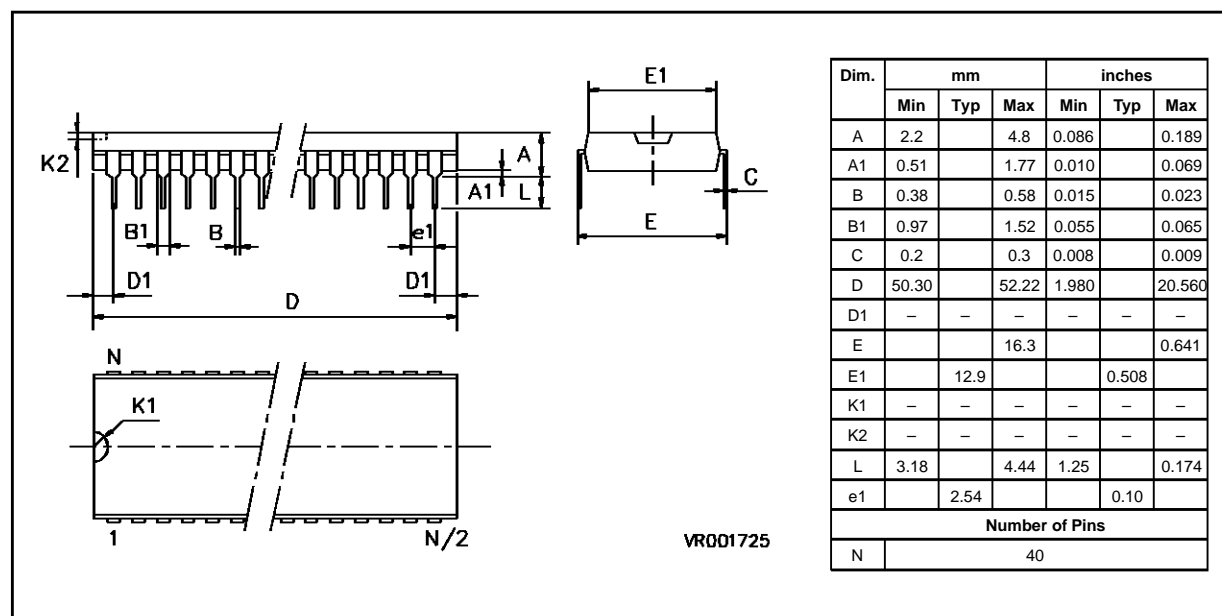
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t _{WRES}	Minimum Pulse Width	RESET Pin	125			ns
t _{OHL}	High to Low Transition Time	PA6, PA7 V _{DD} = 5V, CL = 1000pF (2)		100		ns
t _{OHL}	High to Low Transition Time	DA0-DA5, PB1-PB2, PB4-PB6, PC0-PC7, V _{DD} = 5V, CL = 100pF		20		ns
t _{OLH}	Low to High Transition Time	PA0-PA3, PB1-PB2, PB4-PB6, PC0-PC3 V _{DD} = 5V, CL = 100pF		20		ns
t _{OH}	Data HOLD Time SPI after clock goes low I ² CBUS/S-BUS Only		175			ns
f _{DA}	D/A Converter Repetition Frequency ⁽¹⁾		31.25			kHz
f _{SIO}	SIO Baud Rate ⁽¹⁾		62.50			kHz
t _{WEE}	EEPROM Write Time	T _A = 25°C One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	QA LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T _A = 25°C	10			years
C _{IN}	Input Capacitance (3)	All Inputs Pins	10	pF		
C _{OUT}	Output Capacitance (3)	All outputs Pins	10	pF		
COSCIN, COSCOUT	Oscillator Pins Internal Capacitance(3)			5		pF

Notes:

1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

PACKAGE MECHANICAL DATA

Figure 56. ST6369 40 Pin Plastic Dual-In-Line Package



ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program/Data ROM memories to SGS-THOMSON, the customer has to send:

- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the PROGRAM Memory
- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

The program ROM should respect the ROM Memory Map as in Table 18.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFH.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

- The content should be written into an INTEL INTELLEC format file.
- In the case of 384 bytes of EEPROM, the starting address is 000H and the end address is 7FH. The order of the pages (64 bytes each) is as in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110, 111).
- Undefined or don't care bytes should have the content FFH.

pared and a computer listing is generated from them. 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 list 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.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are com-

ST6369 MICROCONTROLLER OPTION LIST

Table 18. ROM Memory Map

ROM Page	Device Address	EPROM Address (1)	Description
Page 0	0000H-007FH 0080H-07FFH	0000H-007FH 0080H-07FFH	Reserved User ROM
Page 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000H-000FH 0010H-07FFH	1000H-100FH 1010H-17FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	1800H-180FH 1810H-1FFFH	Reserved user ROM

Notes:

- EPROM addresses are related to the use of ST63E69 emulation devices.

ORDERING INFORMATION TABLE

Sales Type	ROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST6369B1/XX	8K/384 Bytes	7	0 to + 70 °C	PDIP40

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

Customer:
Address:
Contact:
Phone No:
Reference:

Package [] (p) Temperature Range [] (t)

For marking one line with 16 characters maximum is possible

Special Marking [] (y/n) Line1 " _____ " (N)

Notes:

(p) B= Dual in Line Plastic

(t) 1= 0 to 70°C

(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
ROM CODE	[]	[]
EEPROM Code (if Desired)	[]	[]

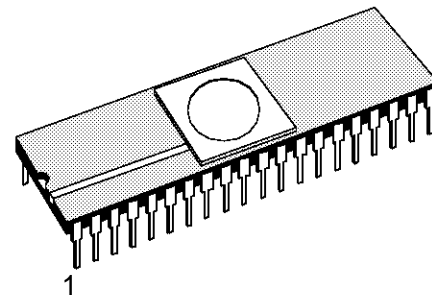
Signature

Date

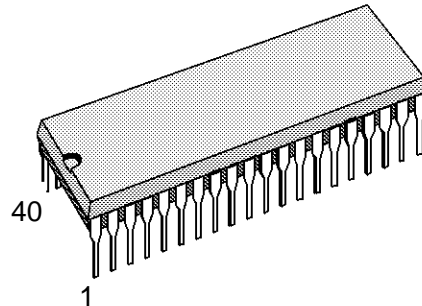
8-BIT EPROM HCMOS MCUs FOR DIGITAL CONTROLLED MULTI FREQUENCY MONITOR

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program EPROM: 7948 bytes
- Reserved Test EPROM: 244 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384 bytes
- 40-Pin Ceramic Dual in Line Package for EPROM version
- 40-Pin Plastic Dual in Line Package for OTP version
- Up to 23 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/I²C BUS and standard serial protocols
- One 14-Bit PWM D/A Converter
- Six 6-Bit PWM D/A Converters
- One A/D converter with 0.5V resolution
- Five interrupt vectors (HSYNC/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- These EPROM and OTP versions are fully pin to pin compatible with ST6369 ROM version.
- The development tool of the ST6369 microcontrollers consists of the ST6369-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.
- EPROM programming board ST6369-EPB



CDIP40W



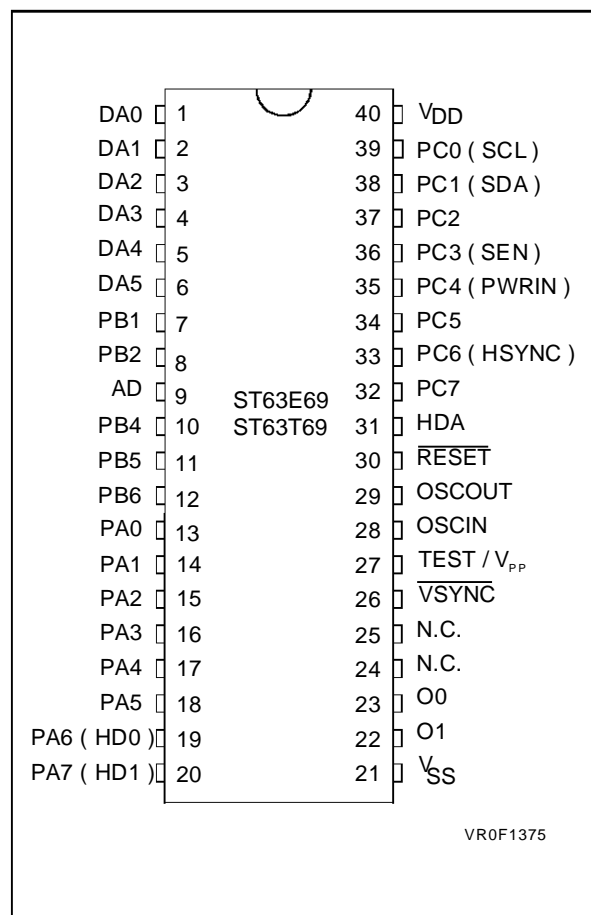
PDIP40

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

EPROM DEVICE	OTP DEVICE	EPROM (Bytes)	EEPROM (Bytes)	D/A Conv.
ST63E69	ST63T69	8K	384	7

Figure 1. ST63E69, T69 Pin Configuration



GENERAL DESCRIPTION

The ST63E69 microcontroller is member of the 8-bit HCMOS ST638x family, a series of devices specially oriented to Digital Controlled Multi Frequency Monitor applications. They are the EPROM/OTP versions of the ST6369 ROM device and are suitable for product prototyping and low volume production. ST6369 is based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to DCMF monitor applications. The macrocells of the ST6369 are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), six 6-bit PWM D/A converters, an A/D converter with 0.5V resolution, a 14-bit PWM D/A converter. In addition the following memory resources are available: program EPROM (8K), data RAM (256 bytes), EEPROM (384 bytes).

Figure 2. ST63E69, T69 Block Diagram

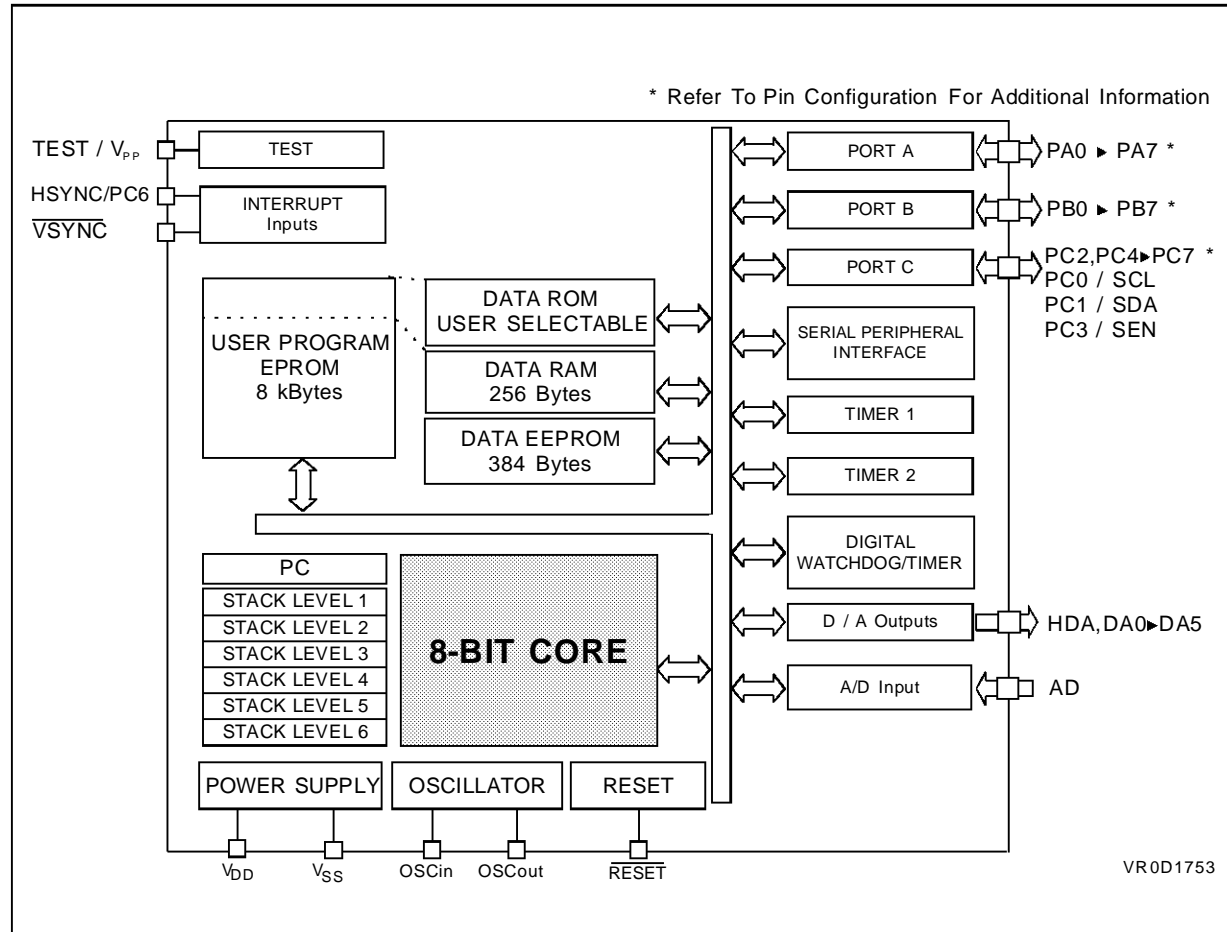


Table 1. Device Summary

DEVICE	EPROM (Bytes)	OTPROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	TARGET ROM DEVICE
ST63E69	8K		256	384	1	1	6	ST6369
ST63T69		8K	256	384	1	1	6	ST6369

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, OSCOUT. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCIN pin is the input pin, the OSCOUT pin is the output pin.

RESET. The active low **RESET** pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the **RESET** pin is low to reduce power consumption during reset phase.

TEST/V_{PP}. The TEST pin must be held at V_{SS} for normal operation.

If this pin is connected to a +12.5V level during the reset phase, The EPROM programming mode is entered.

CAUTION: Exceeding 13V on TEST/V_{PP} pin will permanently damaged the device

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB1-PB2, PB4-PB6. These 5 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the HSYNC/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AD. This is the input of the on-chip 10 levels comparator that can be used to implement the Analog Keyboard function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

VS_{YNC}. This is the Vertical Synchronization pin. This pin is connected to an internal timer interrupt.

O0, O1. These two lines are output open-drain pins with 12V drive.

HDA. This is the output pin of the on-chip 14-bit PWM D/A Converter. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AD	Input, High Impedance, 12V
HDA	Output, Push-Pull
TEST/V _{PP}	Input, Pull-Down, V _{PP} EPROM Programming Voltage Input
OSCIN	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCOU	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB1-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
O0, O1	Output, Open-Drain, 12V
V _{DD} , V _{SS}	Power Supply Pins

ST63E69,T69 EPROM/OTP DESCRIPTION.

The ST63E69 is the EPROM version of the ST6369 ROM product. It is intended for use during the development of an application, and for pre-production and small volume production. The ST63T69 OTP has the same characteristics. They both include EPROM memory instead of the ROM memory of the ST6369, and so the program and constants of the program can be easily modified by the user with the ST63E69 EPROM programming board from SGS-THOMSON.

The Table 3 is a summary of the EPROM/ROM map and its reserved area.

From a user point of view (with the following exceptions) the ST63E69,T69 products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST63E69,T69 is described in the User Manual of the EPROM Programming board.

On the ST63E69, all the 7948 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST63T69 (OTP device) a reserved area for test purposes exists, as for the ST6369 ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended not to use these reserved areas, even when using the ST63E69.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST6369 ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST63E69 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST63E69 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST63E69 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure of the ST63E69 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm² power rating. The ST63E69 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

Table 3. EPROM/ROM Map

ROM Page	Device Address	Description
PAGE 0	0000H-007FH 0080H-07FFH	Reserved User ROM
PAGE 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000H-000FH 0010H-07FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	Reserved User ROM

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 than 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, T_J , in Celsius can be obtained from :

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

Where : T_A = Ambient Temperature.

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

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

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

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

Symbol	Parameter	Value	Unit	
V_{DD}	Supply Voltage	-0.3 to 7.0	V	
V_I	Input Voltage (AD IN)	V	$V_{SS} - 0.3$ to +13	V
V_I	Input Voltage (Other Inputs)	V	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)		$V_{SS} - 0.3$ to +13	V
V_O	Output Voltage (Other Outputs)	V	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
V_{PP}	EPROM programming Voltage	-0.3 to 13.0	V	
I_O	Current Drain per Pin Excluding V_{DD} , V_{SS} , PA6, PA7		± 10	mA
I_O	Current Drain per Pin (PA6, PA7)		± 50	mA
I_{VDD}	Total Current into V_{DD} (source)	50	mA	
I_{VSS}	Total Current out of V_{SS} (sink)	150	mA	
T_J	Junction Temperature	150	°C	
T_{STG}	Storage Temperature	-60 to 150	°C	

Note : 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.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	0	70	°C		
V_{DD}	Operating Supply Voltage	4.5 5.0	6.0	V		
V_{PP}	EPROM programming Voltage	12.0	12.5	13.0	V	
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP40	67	°C/W		

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V _{IL}	Input Low Level Voltage	All I/O Pins	0.2xV		DD	V
V _{IH}	Input High Level Voltage	All I/O Pins	0.8xV _{DD}			V
V _{HYS}	Hysteresis Voltage(1)	All I/O Pins V _{DD} = 5V		1.0		V
V _{OL}	Low Level Output Voltage	DA0-DA5, PB1-PB2, PB3-PB6 PC0-PC7, O0, O1, PA0-PA5 V _{DD} = 4.5V I _{OL} = 1.6mA I _{OL} = 5.0mA			0.4	V
					1.0	V
V _{OL}	Low Level Output Voltage	PA6-PA7 V _{DD} = 4.5V I _{OL} = 1.6mA I _{OL} = 25mA			0.4	V
					1.0	V
V _{OL}	Low Level Output Voltage	OSCOU V _{DD} = 4.5V I _{OL} = 0.4mA			0.4	V
V _{OL}	Low Level Output Voltage	HDA Output V _{DD} = 4.5V I _{OL} = 0.5mA I _{OL} = 1.6mA			0.4	V
					1.0	V
V _{OH}	High Level Output Voltage	PB1-PB2, PB3-PB6, PA0-PA3 V _{DD} = 4.5V I _{OH} = -1.6mA	4.1			V
V _{OH}	High Level Output Voltage	OSCOU, V _{DD} = 4.5V I _{OH} = -0.4mA	4.1			V
V _{OH}	High Level Output Voltage	HDA Output V _{DD} = 4.5V I _{OH} = -0.5mA	4.1			V

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB1-PB2, PB3-PB6, PA0-PA3, PC0-PC3 $V_{IN} = V_{SS}$	- 100	- 50	- 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCIN $V_{IN} = V_{SS}$ $V_{IN} = V_{DD}$	- 10 0.1	- 1 1	- 0.1 10	μA
I_{IL}	Input Pull-down current in Reset	OSCIN	100			μA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up $V_{IN} = V_{DD}$ or V_{SS}	- 10		10	μA
V_{DDRAM}	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up $V_{IN} = V_{SS}$	- 50	- 30	- 10	μA
I_{IL} I_{IH}	Input Leakage Current	AD Pin $V_{IH} = V_{DD}$ $V_{IL} = V_{SS}$ $V_{IH} = 12.0V$	- 1		1 40	μA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 $V_{OH} = V_{DD}$			10	μA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 $V_{OH} = 12V$			40	μA
I_{DD}	Supply Current RUN Mode	$f_{OSC} = 8MHz$, $I_{Load} = 0mA$ $V_{DD} = 6.0V$		6	16	mA
I_{DD}	Supply Current WAIT Mode	$f_{OSC} = 8MHz$, $I_{Load} = 0mA$ $V_{DD} = 6V$		3	10	mA
I_{DD}	Supply Current at transition to RESET	$f_{OSC} =$ Not App, $I_{Load} = 0mA$ $V_{DD} = 6V$		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin	0.3xV		V_{DD}	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	0.8xV _{DD}			V
V_{TA}	Input Level Absolute Tolerance	AD Pin $V_{DD} = 5V$			± 200	mV
V_{TR}	Input Level Relative Tolerance (1)	AD Pin Relative to other levels $V_{DD} = 5V$			± 100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t _{WRES}	Minimum Pulse Width	RESET Pin	125			ns
t _{OHL}	High to Low Transition Time	PA6, PA7 V _{DD} = 5V, CL = 1000pF (2)		100		ns
t _{OHL}	High to Low Transition Time	DA0-DA5, PB1-PB2, PB4-PB6 PC0-PC7 V _{DD} = 5V, CL = 100pF		20		ns
t _{OLH}	Low to High Transition Time	PB1-PB2, PB4-PB6, PA0-PA3, PC0-PC3 V _{DD} = 5V, CL = 100pF		20		ns
t _{OH}	Data HOLD Time SPI after clock goes low I ² CBUS/S-BUS Only		175			ns
f _{DA}	D/A Converter Repetition Frequency ⁽¹⁾		31.25			kHz
f _{SIO}	SIO Baud Rate ⁽¹⁾		62.50			kHz
t _{WEE}	EEPROM Write Time	T _A = 25°C, One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q _A LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T _A = 25°C	10			years
C _{IN}	Input Capacitance (3)	All Inputs Pins	10	pF		
C _{OUT}	Output Capacitance (3)	All outputs Pins	10	pF		
COSCIN, COSCOUT	Oscillator Pins Internal Capacitance(3)			5		pF

Notes:

1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

ORDERING INFORMATION

To ensure compatibility between the EPROM/OTP parts and the corresponding ROM families, the following information is provided. The user should take this information into account when programming the memory of the EPROM parts.

Communication of the ROM Codes. To communicate the contents of memories to SGS-THOMSON, the customer has to send:

- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)
- a filled Option List form as described in the OPTION LIST paragraph.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

- a. The content should be written into an INTEL INTELLEC format file.
- b. In the case of 384 bytes of EEPROM, the starting address is 000h and the end address is 7Fh. The order of the pages (64 bytes each) is as in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110, 111).
- c. Undefined or don't care bytes should have the content FFh.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. 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 list 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.

ST63E69, T69 MICROCONTROLLER OPTION LIST

Customer:
Address:
Contact:
Phone No:
Reference:

Device ☐ (d) Package ☐ (p) Temperature Range ☐ (t)

For marking one line with 16 characters maximum is possible

Special Marking ☐ (y/n) Line1 " _____ " (N)
(For Plastic Package only)

Notes:

(d) 1= ST63E69, 2= ST63T69

(p) B= Plastic Dual in Line, D= Ceramic Dual in line with Window

(t) 1= 0 to 70°C

(N) Letters, digits, ' . , ' - ' , ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
EEPROM Code (if Desired)	<input type="checkbox"/>	<input type="checkbox"/>

Signature

Date

ORDERING INFORMATION TABLE

Sales Type	EPROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST63E69D1/XX	8K/384 Bytes	7	0 to + 70 °C	CDIP40W
ST63T69B1/XX	8K/384 Bytes	7	0 to + 70 °C	PDIP40

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of SGS-THOMSON Microelectronics.

© 1994 SGS-THOMSON Microelectronics - All rights reserved.

Purchase of I²C Components by SGS-THOMSON Microelectronics conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system is granted provided that the system conforms to the I²C Standard Specification as defined by Philips.

SGS-THOMSON Microelectronics Group of Companies

Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.