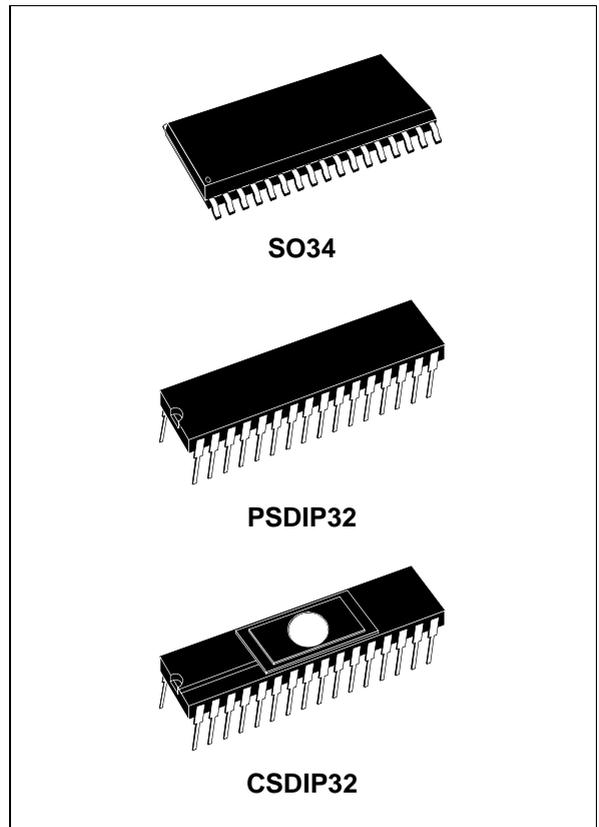




8-BIT MCUs WITH 8 TO 16K ROM/OTP/EPROM, 384 TO 512 BYTES RAM, ADC, DAC (PWM), TIMER AND I²C

PRODUCT PREVIEW

- User Program Memory ROM/OTP/EPROM: 8 to 16K bytes
- Data RAM: 384 to 512 bytes (256 bytes stack)
- Master Reset and Power-On Reset
- Run, Wait, Slow, Halt and RAM Retention modes
- 18 I/O lines:
 - 1 programmable interrupt input
 - 5 high sink outputs
 - 4 analog alternate inputs
 - 8 alternate functions
 - EMI filtering
- Programmable watchdog (WDG)
- 16-bit Timer with 2 Input Capture and 2 Output Compare functions (with 1 output pin)
- 8-bit Analog to Digital Converter with 4 channels
- Four 10-bit Digital to Analog Converter channels with PWM output
- Fast I²C Multi Master Interface
- 63 basic instructions and 17 main address modes
- 8x8 unsigned multiply instruction
- True bit manipulation
- Versatile Development Tools (DOS and Windows) including assembler, linker, C-compiler, archiver, source level debugger, and hardware emulator



Device Summary

Features	ST72272K2	ST72272K4
Program Memory - bytes	8K	16K
RAM (stack) - bytes	384 (256)	512 (256)
10-Bit D/A Converter	4 channels	
A/D Converter	4 channels	
16-Bit Timer	1	
I ² C Bus	1 multimaster	
I/Os	24	
Operating Supply	4.0 to 5.5 V	
CPU Frequency	8 MHz max (24 MHz quartz)	
Temperature Range	0°C to + 70°C	
Package	SO34 - SDIP32	

Table of Contents

ST72272	1
1 GENERAL DESCRIPTION	4
1.1 INTRODUCTION	4
1.2 PIN DESCRIPTION	5
1.3 MEMORY MAP	7
2 CENTRAL PROCESSING UNIT	10
2.1 INTRODUCTION	10
2.2 MAIN FEATURES	10
2.3 CPU REGISTERS	10
3 CLOCKS, RESET, INTERRUPTS & POWER SAVING MODES	13
3.1 CLOCK SYSTEM	13
3.1.1 General Description	13
3.1.2 External Clock	13
3.2 RESET	14
3.2.1 Introduction	14
3.2.2 External Reset	14
3.2.3 Reset Operation	14
3.2.4 Power-on Reset	14
3.3 INTERRUPTS	16
3.4 POWER SAVING MODES	19
3.4.1 Introduction	19
3.4.2 Slow Mode	19
3.4.3 Wait Mode	19
3.4.4 Halt Mode	20
3.5 REGISTER DESCRIPTION	21
4 ON-CHIP PERIPHERALS	22
4.1 I/O PORTS	22
4.1.1 Introduction	22
4.1.2 Functional Description	22
4.1.3 Register Description	26
4.2 WATCHDOG TIMER (WDG)	28
4.2.1 Introduction	28
4.2.2 Main Features	28
4.2.3 Functional Description	29
4.2.4 Register Description	29
4.3 16-BIT TIMER	30
4.3.1 Introduction	30
4.3.2 Main Features	30
4.3.3 Functional Description	30
4.3.4 Register Description	40
4.4 I2C BUS INTERFACE (I2C)	45
4.4.1 Introduction	45
4.4.2 Main Features	45
4.4.3 General Description	45

Table of Contents

4.4.4	Functional Description	47
4.4.5	Register Description	50
4.5	PWM/BRM GENERATOR (DAC)	55
4.5.1	Introduction	55
4.5.2	Main Features	55
4.5.3	Functional Description	55
4.5.4	Register Description	59
4.6	8-BIT A/D CONVERTER (ADC)	61
4.6.1	Introduction	61
4.6.2	Main Features	61
4.6.3	Functional Description	62
4.6.4	Register Description	63
5	INSTRUCTION SET	64
5.1	ST7 ADDRESSING MODES	64
5.1.1	Inherent	65
5.1.2	Immediate	65
5.1.3	Direct	65
5.1.4	Indexed (No Offset, Short, Long)	65
5.1.5	Indirect (Short, Long)	65
5.1.6	Indirect Indexed (Short, Long)	66
5.1.7	Relative mode (Direct, Indirect)	66
5.2	INSTRUCTION GROUPS	67
6	ELECTRICAL CHARACTERISTICS	70
6.1	ABSOLUTE MAXIMUM RATINGS	70
6.2	RECOMMENDED OPERATING CONDITIONS	71
6.3	DC ELECTRICAL CHARACTERISTICS	72
6.4	A/D CONVERTER CHARACTERISTICS	72
6.5	PWM (DAC) CHARACTERISTICS	73
6.6	I2C CHARACTERISTICS	73
7	GENERAL INFORMATION	75
7.1	EPROM ERASURE	75
7.2	PACKAGE MECHANICAL DATA	76
7.3	ORDERING INFORMATION	78
7.3.1	Transfer Of Customer Code	78

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST72272 series is a HCMOS microcontroller unit (MCU) from the ST7 family with a dedicated D/A Converter peripherals offering 4 PWM outputs.

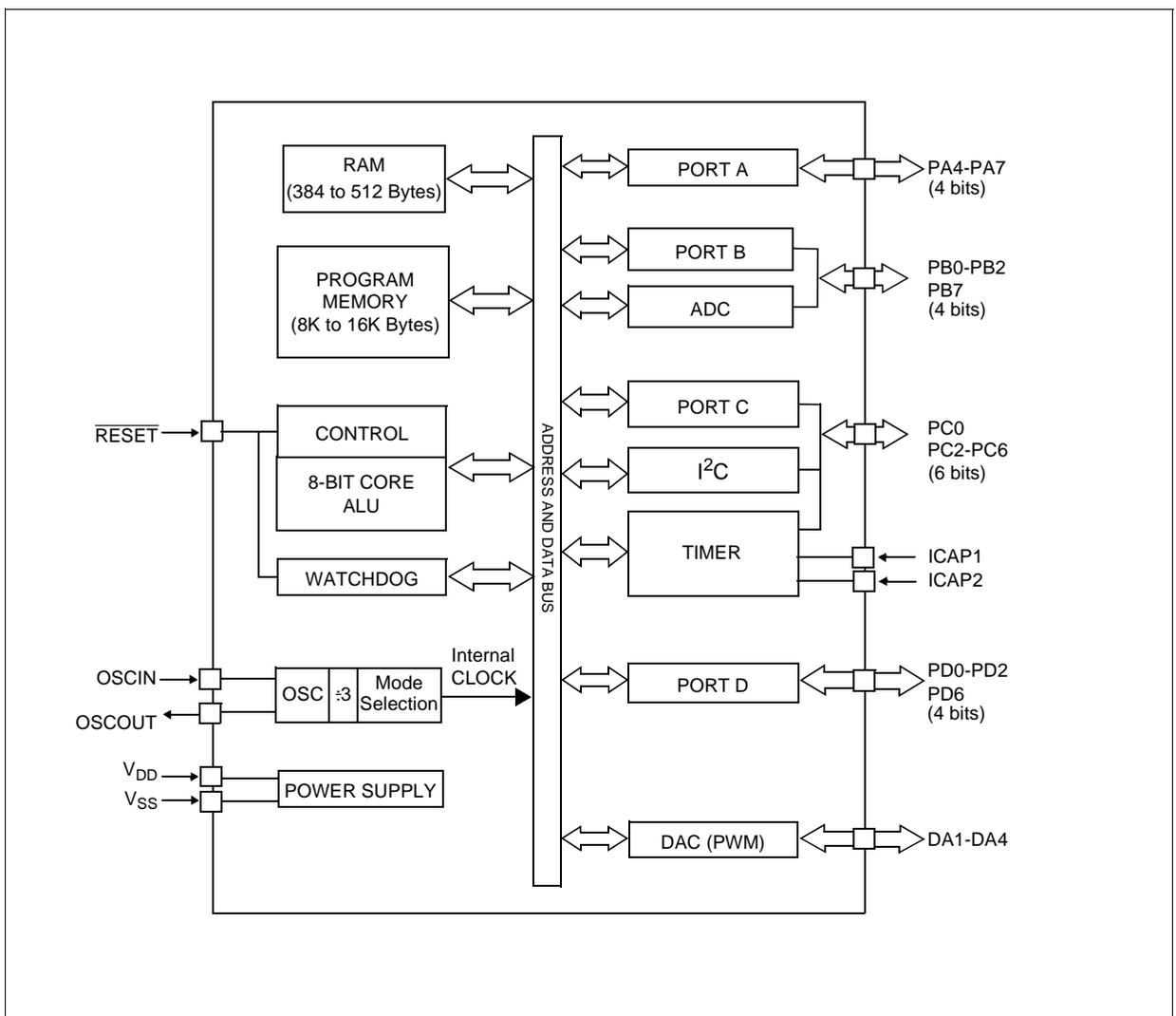
It is based around an industry standard 8-bit core and offers an enhanced instruction set. The processor runs with an external clock up to 24 MHz with a 5.5V supply. Due to the fully static design of this device, operation down to DC is possible. Under software control the ST72272 can be placed in WAIT, SLOW or HALT mode thus reducing power consumption. The enhanced instruction set and addressing modes afford real

programming potential.

In addition to standard 8-bit data management the ST7 features true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes on the whole memory.

The device includes an on-chip oscillator, CPU, 8 to 16 Kbytes ROM/OTP/EPROM, 384 to 512 bytes RAM, 18 I/O lines, a Timer with 2 Input Captures and 2 Output Compares, a 4-channel A/D Converter, an I²C multi Master, a Watchdog Reset and a 4-channel 10-bit D/A Converter with PWM output.

Figure 1. ST72272 Block Diagram



1.2 PIN DESCRIPTION

Figure 2. 34-Pin SO Package Pinout

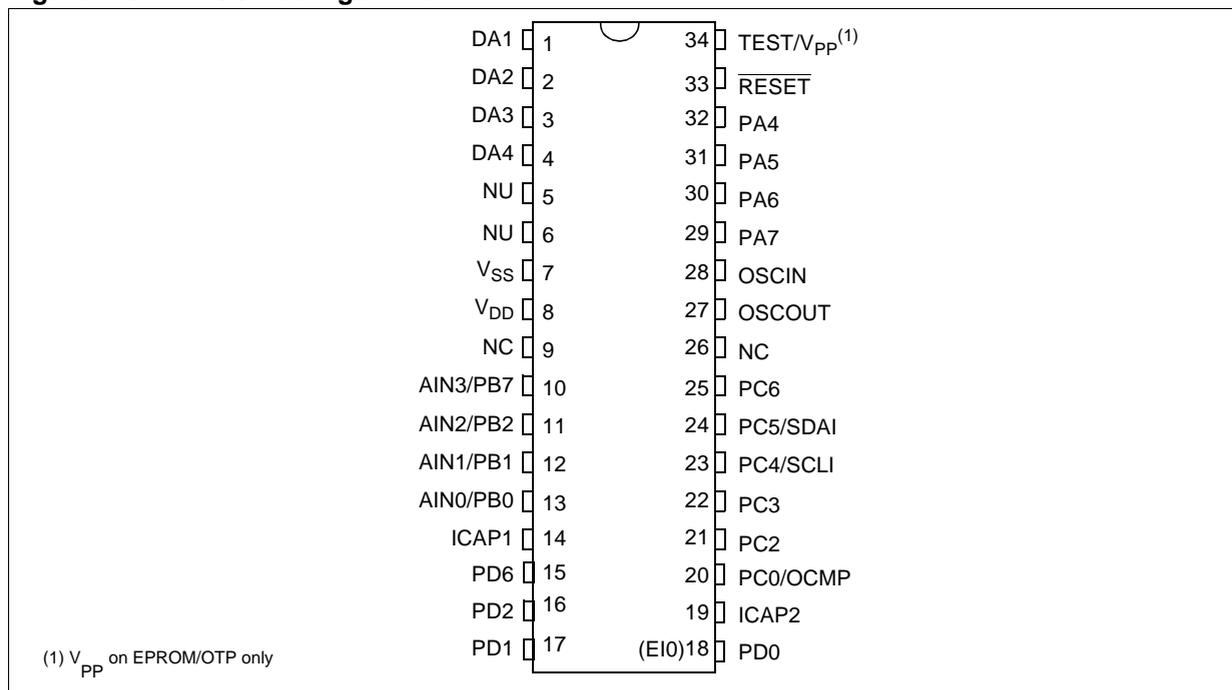
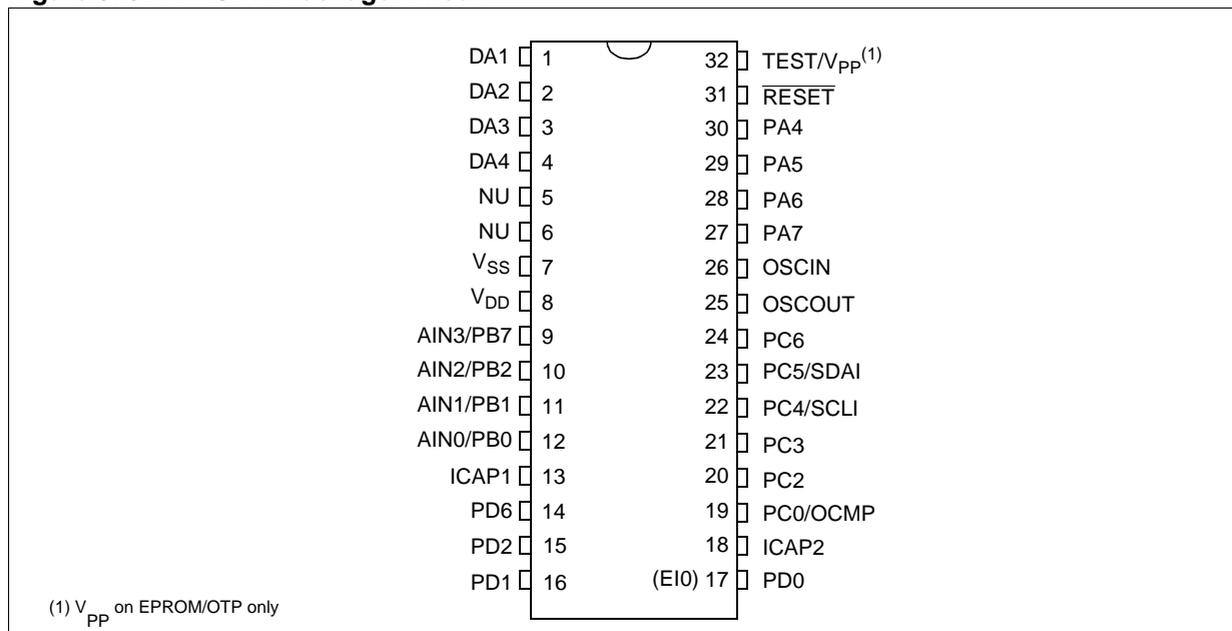


Figure 3. 32-Pin SDIP Package Pinout



Note: Several pins of the I/O ports assume software programmable alternate functions as shown in the pin description.

PIN DESCRIPTION (Cont'd)

Table 1. 34-Pin SO and 32-Pin SDIP Package Pin Description

Pin n° SO34	Pin n° SDIP32	Pin Name	Type	Description	Remarks
1	1	DA1	O	10-bit D/A (PWM output)	For analog controls, after external filtering
2	2	DA2	O	10-bit D/A (PWM output)	
3	3	DA3	O	10-bit D/A (PWM output)	
4	4	DA4	O	10-bit D/A (PWM output)	
5	5	NU		Non User pin. Must be left unconnected	
6	6	NU		Non User pin. Must be left unconnected	
7	7	V _{SS}	S	Ground	
8	8	V _{DD}	S	Main power supply	
9		NC		Not Connected	
10	9	PB7/AIN3	I/O	Port B7 or ADC Analog Input 3	
11	10	PB2/AIN2	I/O	Port B6 or ADC Analog Input 2	
12	11	PB1/AIN1	I/O	Port B5 or ADC Analog Input 1	
13	12	PB0/AIN0	I/O	Port B4 or ADC Analog Input 0	
14	13	ICAP1		Timer Input Capture 1	Not for general purpose I/O
15	14	PD6	I/O	Port D6	
16	15	PD2	I/O	Port D2	
17	16	PD1	I/O	Port D1	
18	17	PD0	I/O	Port D0	External Interrupt: EI0
19	18	ICAP2		Timer Input Capture 2 with 256 prescaler	Not for general purpose I/O
20	19	PC0/OCMP	I/O	Port C0 or Timer Output Compare	
21	20	PC2	I/O	Port C2	
22	21	PC3	I/O	Port C3	
23	22	PC4/SCLI	I/O	Port C4 or I ² C Serial Clock	
24	23	PC5/SDAI	I/O	Port C5 or I ² C Serial Data	
25	24	PC6	I/O	Port C6	High Current
26		NC		Not Connected	
27	25	OSCOUT	O	Input/Output Oscillator pin. These pins connect a parallel-resonant crystal, or an external source to the on-chip oscillator.	
28	26	OSCIN	I		
29	27	PA7	I/O	Port A7	High Current
30	28	PA6	I/O	Port A6	High Current
31	29	PA5	I/O	Port A5	High Current
32	30	PA4	I/O	Port A4	High Current
33	31	RESET	I/O	Bidirectional. Active low. Top priority non maskable interrupt.	It can be used to reset external peripherals.
34	32	V _{PP} /TEST	S	Test mode pin. In EPROM devices acts as programming voltage input V _{PP} .	This pin should be tied low in user mode

Note: S= Supply

1.3 MEMORY MAP

Figure 4. Program Memory Map

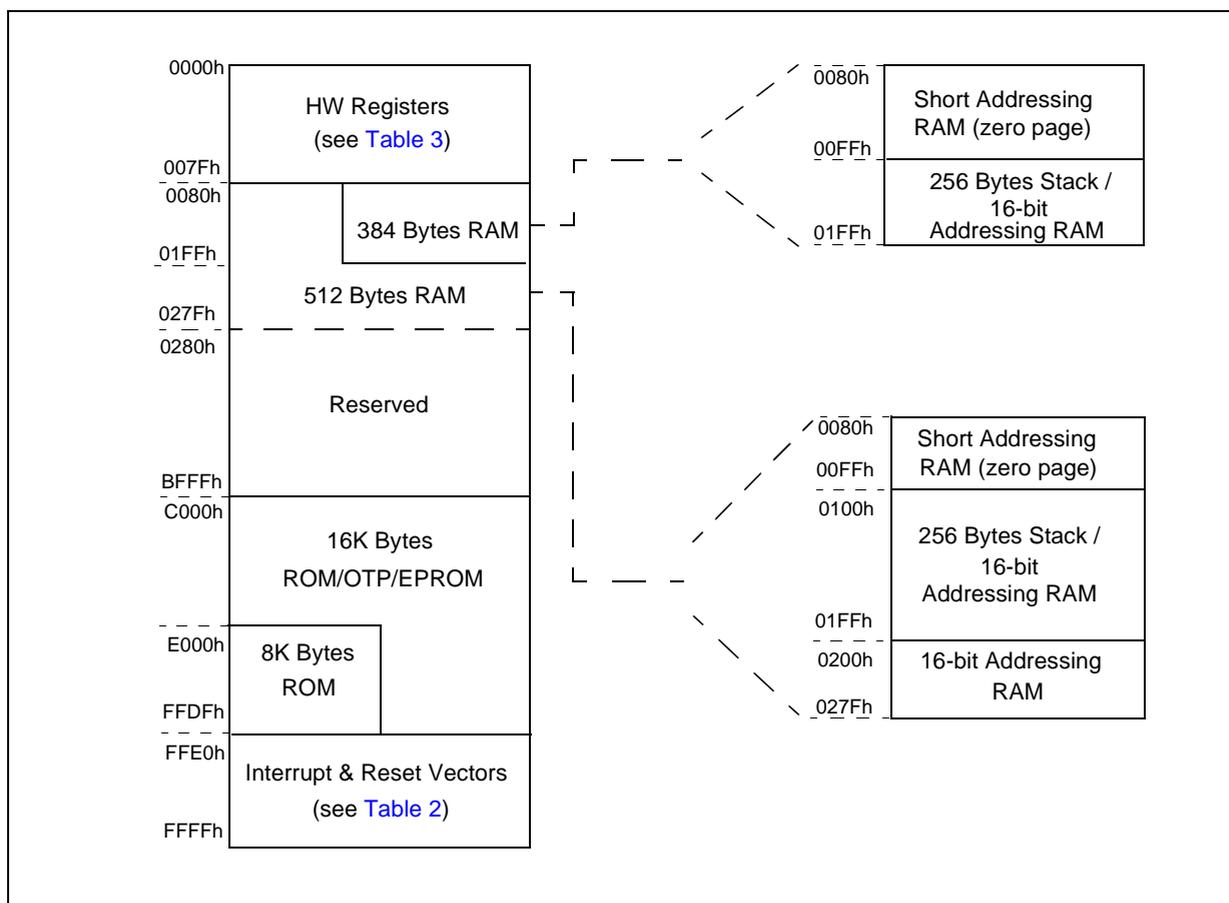


Table 2. Interrupt Vector Map

Vector Address	Description	Remarks
FFE0-FFE1h	Reserved	
FFE2-FFE3h	Reserved	
FFE4-FFE5h	I ² C Interrupt Vector	Internal Interrupts
FFE6-FFE7h	Timer Overflow Interrupt Vector	"
FFE8-FFE9h	Timer Output Compare Interrupt Vector	"
FFEA-FFEBh	Timer Input Capture Interrupt Vector	"
FFEC-FFEDh	Reserved	
FFEE-FFEFh	Reserved	
FFF0-FFF1h	EIO Interrupt Vector	External Interrupt
FFF2-FFF3h	Reserved	
FFF4-FFF5h	Reserved	
FFF6-FFF7h	Reserved	
FFF8-FFF9h	Reserved	
FFFA-FFFBh	Reserved	
FFFC-FFFDh	TRAP Interrupt Vector	Software Interrupt
FFFE-FFFFh	RESET Vector	CPU Interrupt

MEMORY MAP (Cont'd)

Table 3. Hardware Register Memory Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h 0001h	Port A	PADR PADDR	Port A Data Register Port A Data Direction Register	00h 00h	R/W R/W
0002h 0003h	Port C	PCDR PCDDR	Port C Data Register Port C Data Direction Register	00h 00h	R/W R/W
0004h 0005h	Port D	PDDR PDDDR	Port D Data Register Port D Data Direction Register	00h 00h	R/W R/W
0006h 0007h 0008h	Port B	PBDR PBDDR PBICFGR	Port B Data Register Port B Data Direction Register Port B Input Pull-Up Configuration Register	00h 00h 00h	R/W R/W R/W
0009h		MISCR	Miscellaneous Register	00h	R/W
000Ah 000Bh	ADC	ADCDR ADCCSR	ADC Data Register ADC Control Status register	00h 00h	Read only R/W
000Ch	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
000Dh 000Fh	Reserved Area (3 bytes)				
00010h	ITR	ITRFRE	Interrupt Register	00h	R/W
0011h 0012h 0013h 0014h 0015h 0016h 0017h 0018h 0019h 001Ah 001Bh 001Ch 001Dh 001Eh 001Fh	TIM	TIMCR2 TIMCR1 TIMSR TIMIC1HR TIMIC1LR TIMOC1HR TIMOC1LR TIMCHR TIMCLR TIMACHR TIMACLR TIMIC2HR TIMIC2LR TIMOC2HR TIMOC2LR	Timer Control Register 2 Timer Control Register 1 Timer Status Register Timer Input Capture 1 High Register Timer Input Capture 1 Low Register Timer Output Compare 1 High Register Timer Output Compare 1 Low Register Timer Counter High Register Timer Counter Low Register Timer Alternate Counter High Register Timer Alternate Counter Low Register Timer Input Capture 2 High Register Timer Input Capture 2 Low Register Timer Output Compare 2 High Register Timer Output Compare 2 Low Register	00h 00h 00h xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W Read only Read only Read only R/W R/W Read only Read only Read only Read only Read only Read only R/W R/W
0020h 0023h	Reserved Area (4 bytes)				
0024h 0025h 0026h 0027h 0028h 0029h		PWM1 BRM21 PWM2 PWM3 BRM43 PWM4	10 BIT PWM / BRM Register	80h 00h 80h 80h 00h 80h	R/W R/W R/W R/W R/W R/W
002Ah to 0042h	Reserved Area (25 bytes)				
0043h	TIM		ICAP Pin Configuration Warning: Write 0Ch in this register to use the ICAP1 and ICAP2 functions.	08h	R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
0044h 0058h	Reserved Area (21 bytes)				
0059h 005Ah 005Bh 005Ch 005Dh 005Eh 005Fh	I ² C	I2CDR	I ² C Data Register Reserved	00h	R/W
		I2COAR	I ² C (7 Bits) Slave Address Register	00h	R/W
		I2CCCR	I ² C Clock Control Register	00h	R/W
		I2CSR2	I ² C Status Register 2	00h	Read only
		I2CSR1	I ² C Status Register 1	00h	Read only
		I2CCR	I ² C Control Register	00h	R/W
0060h to 007Fh		Reserved Area (32 bytes)			

2 CENTRAL PROCESSING UNIT

2.1 Introduction

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

2.2 Main Features

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- 8 MHz CPU internal frequency
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

2.3 CPU Registers

The 6 CPU registers shown in [Figure 5](#) are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

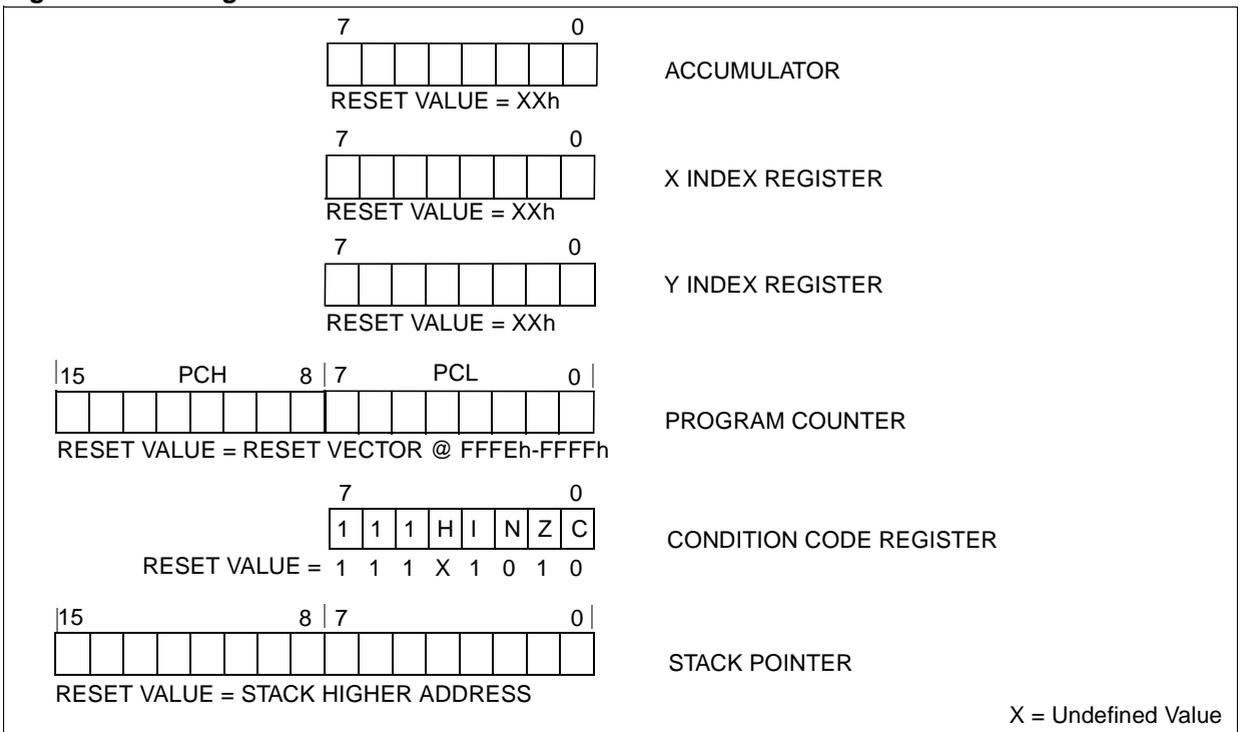
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCH (Program Counter Low which is the LSB) and PCL (Program Counter High which is the MSB).

Figure 5. CPU Registers



CENTRAL PROCESSING UNIT (Cont'd)**Condition Code Register (CC)**

Read/Write

Reset Value: 111x1010

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = H Half carry.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 3 = I Interrupt mask.

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.

0: Interrupts are enabled.

1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptable because the I bit is set by hardware when you

enter it and reset by the IRET instruction at the end of the interrupt routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = N Negative.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7th bit of the result.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative

(i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = Z Zero.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

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

1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = C Carry/borrow.

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

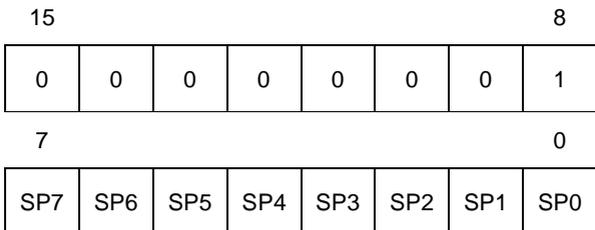
This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh



The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 6).

Since the stack is 256 bytes deep, the most significant byte is forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer can be directly accessed by a LD instruction.

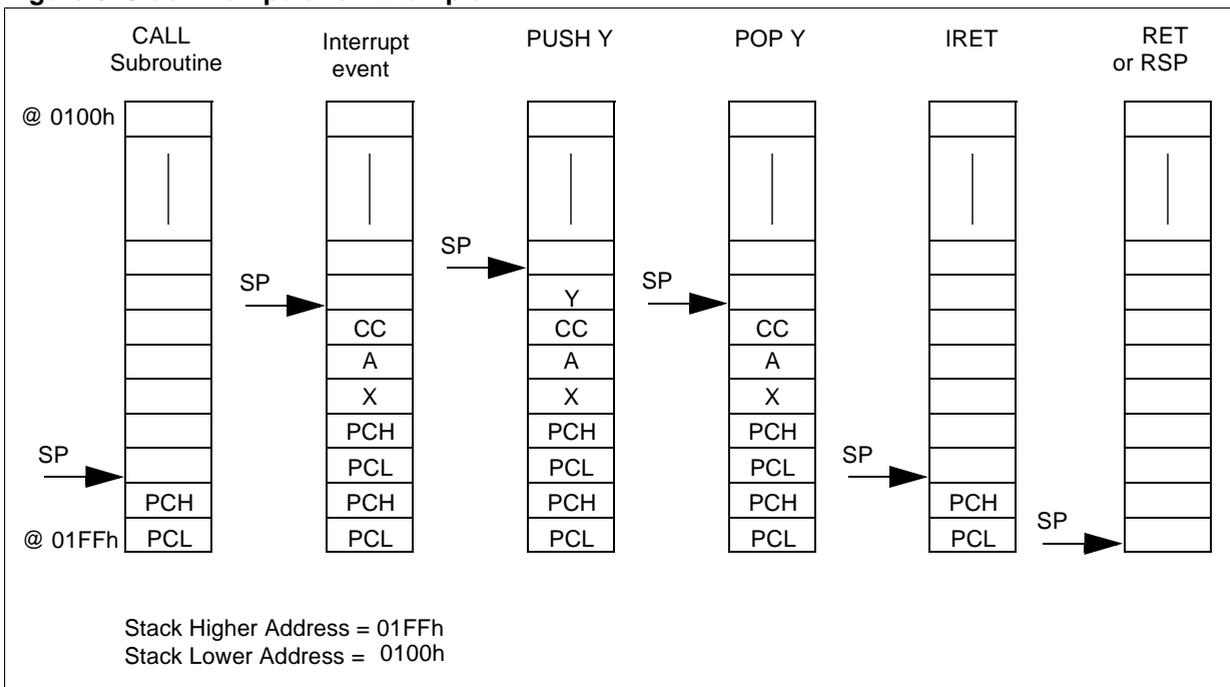
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 6.

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 6. Stack Manipulation Example



3 CLOCKS, RESET, INTERRUPTS & POWER SAVING MODES

3.1 CLOCK SYSTEM

3.1.1 General Description

The MCU accepts either a Crystal or Ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}).

The external Oscillator clock is first divided by 3, and an additional division factor of 2 can be applied if Slow Mode is selected by resetting the SMS bit in the Miscellaneous Register. This reduces the frequency of the f_{CPU} ; the clock signal is also routed to the on-chip peripherals. The CPU clock signal consists of a square wave with a duty cycle of 50%.

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} . The circuit shown in Figure 7 is recommended when using a crystal, and Table 4 lists the recommended capacitance and feedback resistance values. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilisation time.

Use of an external CMOS oscillator is recommended when crystals outside the specified frequency ranges are to be used.

Table 4. Recommended Crystal Values

24 Mhz				Unit
R_{SMAX}	70	25	20	Ohms
C_{L1}	22	47	56	pf
C_{L2}	22	47	56	pf

Legend:

C_{L1} , C_{L2} = Maximum total capacitance on pins OSCIN and OSCOUT (the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device).

R_{SMAX} = Maximum series parasitic resistance of the quartz allowed.

Note: The tables relate to the quartz crystal only (not ceramic resonator).

3.1.2 External Clock

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected. The $t_{\text{OX-OV}}$ specifications does not apply when using an external clock input. The equivalent specification of the external clock source should be used instead of $t_{\text{OX-OV}}$ (see Electrical Characteristics).

Figure 7. Crystal/Ceramic Resonator

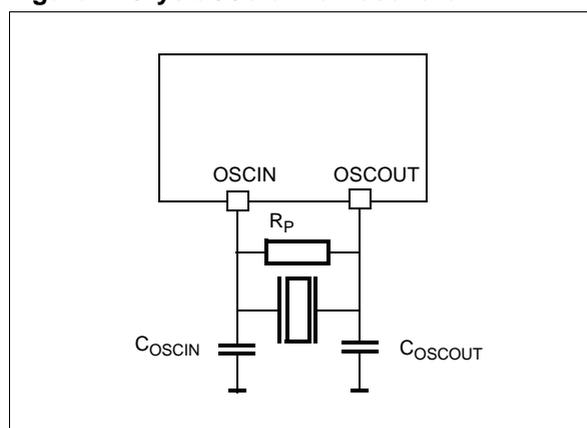
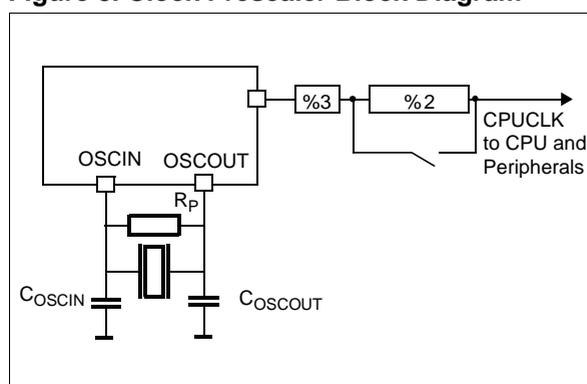


Figure 8. Clock Prescaler Block Diagram



3.2 RESET

3.2.1 Introduction

There are three sources of Reset:

- $\overline{\text{RESET}}$ pin (external source)
- Power-On Reset (Internal source)
- WATCHDOG (Internal Source)

The Reset Service Routine vector is located at address FFFEh-FFFFh.

3.2.2 External Reset

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated pull-up resistor. When one of the internal Reset sources is active, the Reset pin is driven low to reset the whole application.

3.2.3 Reset Operation

The duration of the Reset condition, which is also reflected on the output pin, is fixed at 4096 internal CPU Clock cycles. A Reset signal originating from an external source must have a duration of at least 1.5 internal CPU Clock cycles in order to be recognised. At the end of the Power-On Reset cycle, the MCU may be held in the Reset condition by an External Reset signal. The $\overline{\text{RESET}}$ pin may thus be used to ensure V_{DD} has risen to a point where the MCU can operate correctly before the user program is run. Following a Power-On Reset event, or

after exiting Halt mode, a 4096 CPU Clock cycle delay period is initiated in order to allow the oscillator to stabilise and to ensure that recovery has taken place from the Reset state.

During the Reset cycle, the device Reset pin acts as an output that is pulsed low. In its high state, an internal pull-up resistor of about $300\text{K}\Omega$ is connected to the Reset pin. This resistor can be pulled low by external circuitry to reset the device.

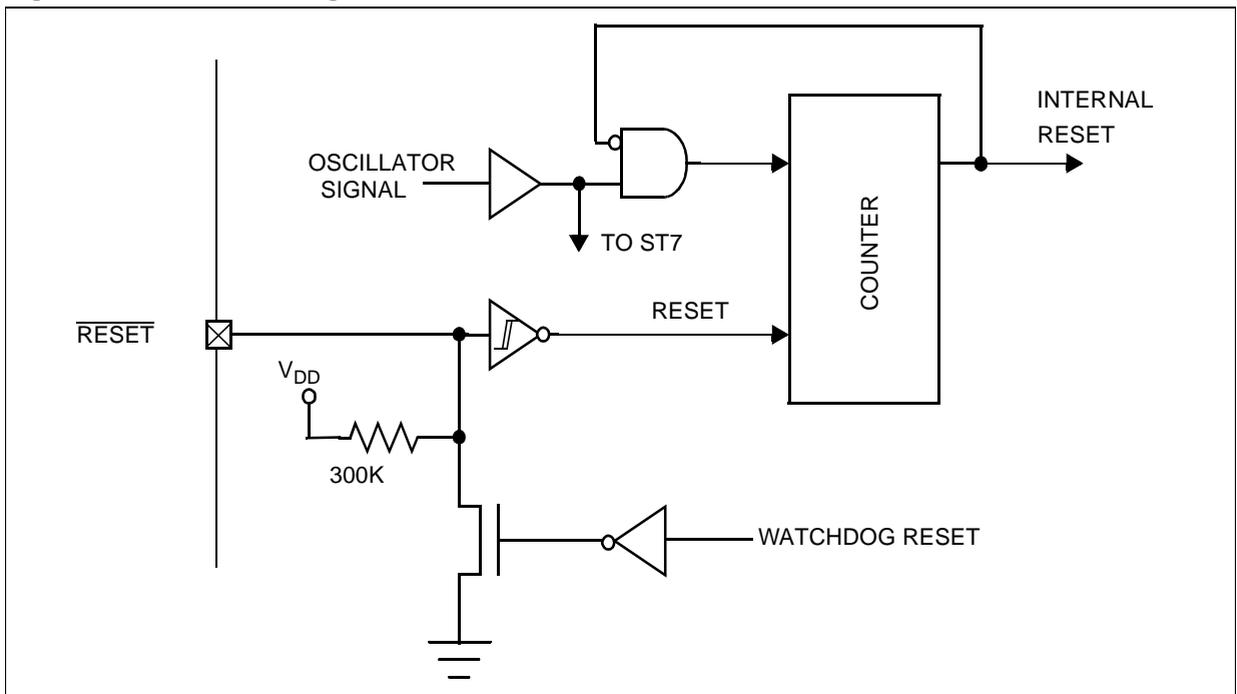
3.2.4 Power-on Reset

This circuit detects the ramping up of V_{DD} , and generates a pulse that is used to reset the application (at approximately $V_{DD} = 2\text{V}$).

Power-On Reset is designed exclusively to cope with power-up conditions, and should not be used in order to attempt to detect a drop in the power supply voltage.

Caution: to re-initialize the Power-On Reset, the power supply must fall below approximately 0.8V (V_{tn}), prior to rising above 2V . If this condition is not respected, on subsequent power-up the Reset pulse may not be generated. An external Reset pulse may be required to correctly reactivate the circuit.

Figure 9. Reset Block Diagram

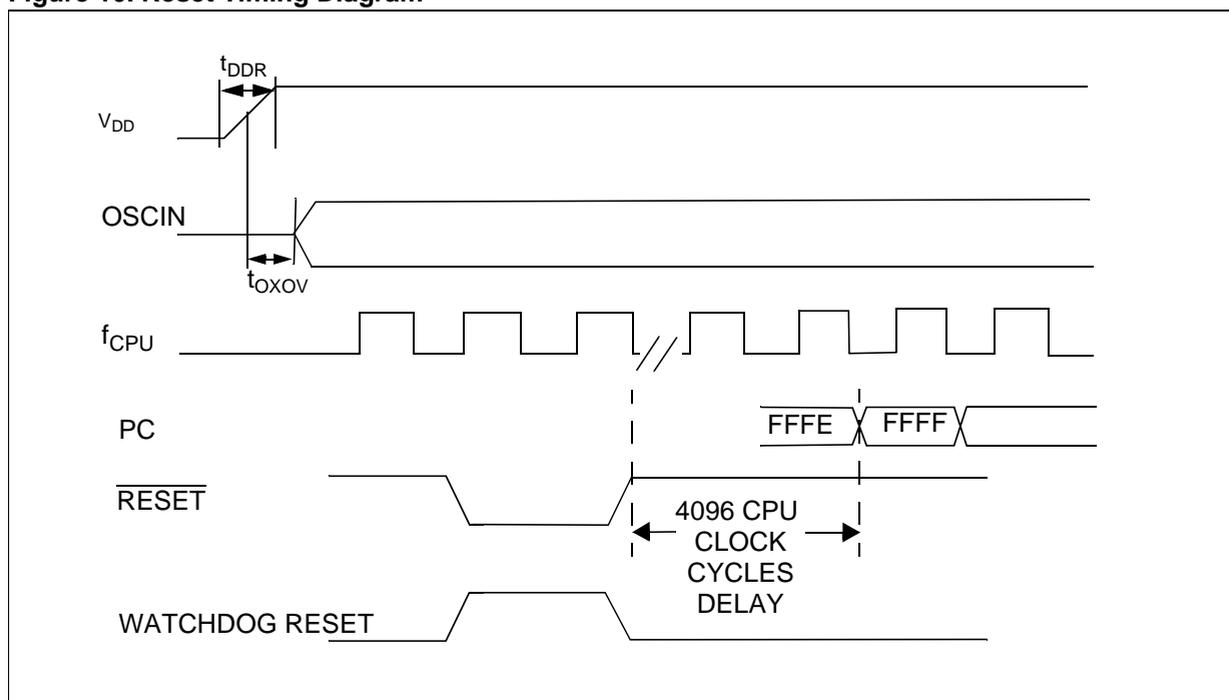


RESET (Cont'd)

Table 5. List of sections affected by RESET, WAIT and HALT (Refer to 3.6 for Wait and Halt Modes)

Section	RESET	WAIT	HALT
CPU clock running at 4 MHz	X		
Timer Prescaler reset to zero	X		
Timer Counter set to FFFCh	X		
All Timer enable bits set to 0 (disabled)	X		
Data Direction Registers set to 0 (as Inputs)	X		
Set Stack Pointer to 01FFh	X		
Force Internal Address Bus to restart vector FFFEh, FFFFh	X		
Set Interrupt Mask Bit (I-Bit, CC) to 1 (Interrupt disable)	X		
Set Interrupt Mask Bit (I-Bit, CC) to 0 (Interrupt enable)		X	X
Reset HALT latch	X		
Reset WAIT latch	X		
Disable Oscillator (for 4096 cycles)	X		X
Set Timer Clock to 0	X		X
Watchdog counter reset	X		
Watchdog register reset	X		
Port data registers reset	X		
Other on-chip peripherals: registers reset	X		

Figure 10. Reset Timing Diagram



Note: Refer to Electrical Characteristics for values of t_{DDR} and t_{OXOV}

3.3 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in [Table 6](#) and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in [Figure 11](#).

The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to [Table 6](#) for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority management

By default, a servicing interrupt can not be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see [Table 6](#)).

Non Maskable Software Interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on [Figure 11](#).

Interrupts and Low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific men-

tioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT” column in [Table 6](#)).

External Interrupts

External interrupt vectors can be loaded in the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the Halt low power mode.

The external interrupt polarity can be selected through the Miscellaneous register or Interrupt register (if available) (see [Section 3.5](#)).

An external interrupt triggered on edge will be latched and the interrupt request automatically cleared on entering the interrupt service routine.

More than one input pin can be connected to the same interrupt request (depending on the device). In this case, all inputs configured as interrupt are logically ORed.

Warning: The type of polarity defined in the Miscellaneous or Interrupt register (if available) applies to the EI source. In case of an ORed source, a low level on an I/O pin configured as input with interrupt, masks the interrupt request even in case of rising-edge polarity.

Peripheral Interrupts

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both:

- The I bit of the CC register is cleared.
- The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

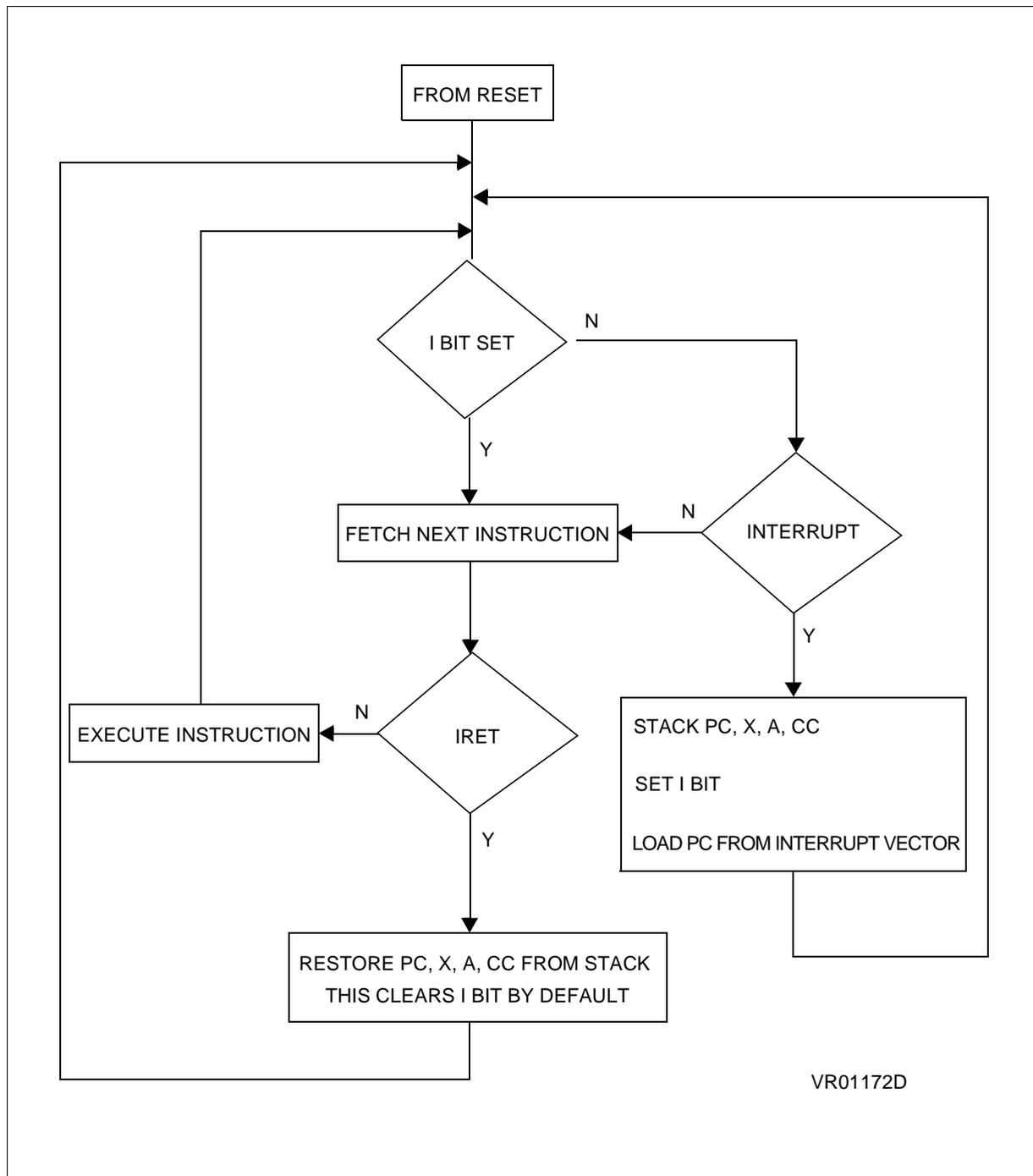
Clearing an interrupt request is done by:

- writing “0” to the corresponding bit in the status register or
- an access to the status register while the flag is set followed by a read or write of an associated register.

Note: the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being enabled) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

Figure 11. Interrupt Processing Flowchart



VR01172D

INTERRUPTS (Cont'd)

Table 6. Interrupt Mapping

Source Block	Description	Register Label	Flag	Exit from HALT	Vector Address	Priority Order
RESET	Reset	N/A	N/A	yes	FFFEh-FFFFh	Highest Priority ↓ Lowest Priority
TRAP	Software Interrupt	N/A	N/A	no	FFFCh-FFFDh	
NOT USED					FFF2h-FFFBh	
EIO	Ext. Interrupt PDO, falling edge	ITRFRE	EIOF	yes	FFF0h-FFF1h	
NOT USED					FFFECh-FFFEFh	
TIMER	Input Capture 1	TIMSR	ICF1	no	FFEAh-FFEBh	
	Input Capture 2		ICF2		FFE8h-FFE9h	
	Output Compare 1		OCF1		FFE6h-FFE7h	
	Output Compare 2		OCF2			
	Timer Overflow		TOF			
I2C	I2C Interface Interrupt	I2CSR1 I2CSR2	*		FFE4h-FFE5h	
NOT USED					FFE0h-FFE3h	

* Many flags can cause an interrupt: see peripheral interrupt status register description.

3.4 POWER SAVING MODES

3.4.1 Introduction

There are three Power Saving modes. Slow Mode is selected by setting the relevant bits in the Miscellaneous register. Wait and Halt modes may be entered using the WFI and HALT instructions.

3.4.2 Slow Mode

In Slow mode, the oscillator frequency can be divided by a value defined in the Miscellaneous Register. The CPU and peripherals are clocked at this lower frequency. Slow mode is used to reduce power consumption, and enables the user to adapt clock frequency to available supply voltage.

Note: On reset, Slow mode is selected by default ($F_{OSC}/6$).

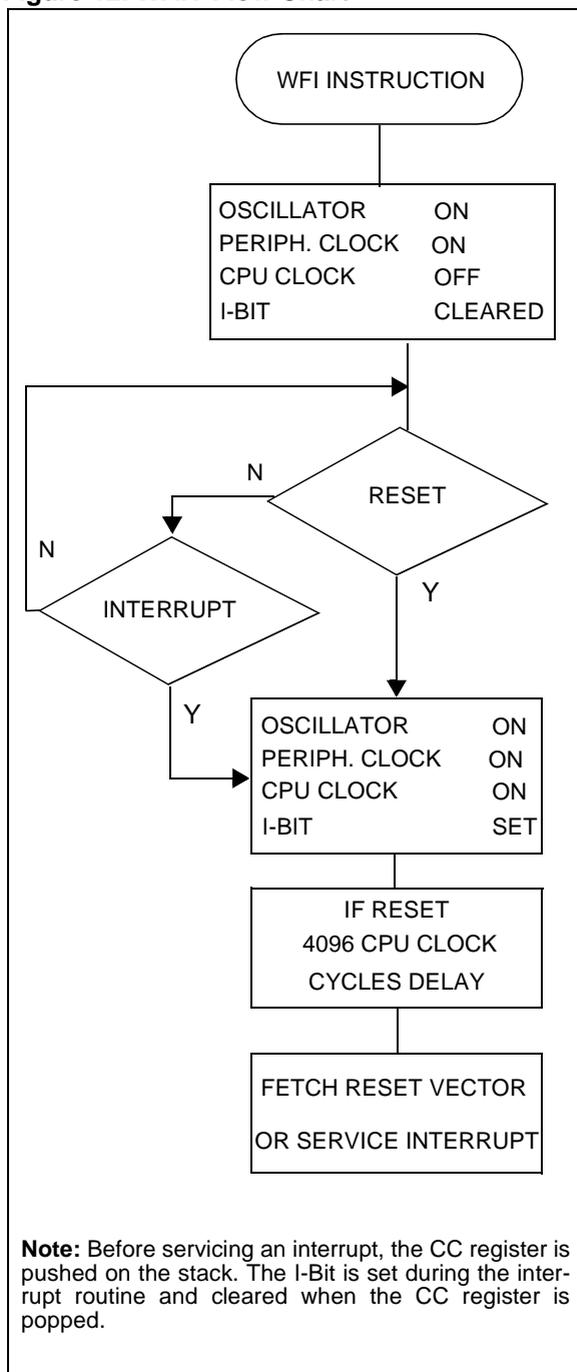
3.4.3 Wait Mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU. All peripherals remain active. During Wait mode, the I bit (CC Register) is cleared, so as to enable all interrupts. All other registers and memory remain unchanged. The MCU will remain in Wait mode until an Interrupt or Reset occurs, whereupon the Program Counter branches to the starting address of the Interrupt or Reset Service Routine.

The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to [Figure 12](#) below.

Figure 12. WAIT Flow Chart



POWER SAVING MODES (Cont'd)

3.4.4 Halt Mode

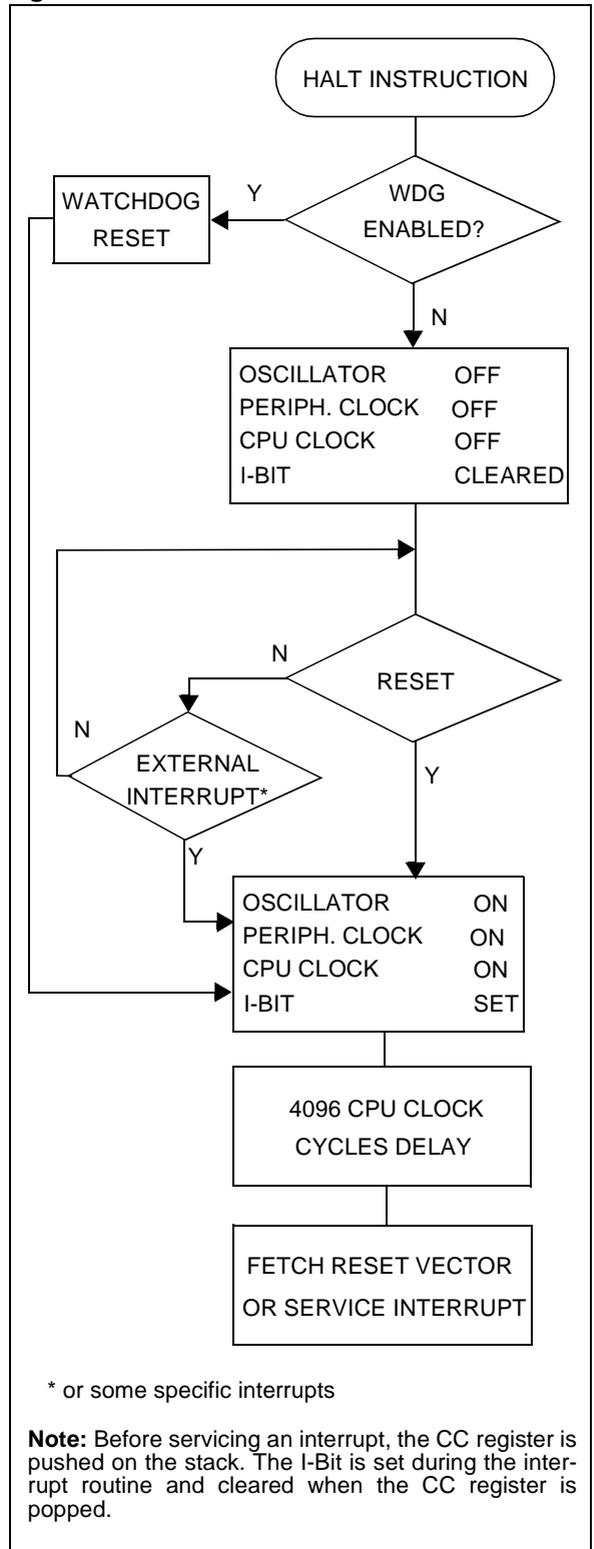
The Halt mode is the MCU lowest power consumption mode. The Halt mode is entered by executing the HALT instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals. The Halt mode cannot be used when the watchdog is enabled, if the HALT instruction is executed while the watchdog system is enabled, a watchdog reset is generated thus resetting the entire MCU.

When entering Halt mode, the I bit in the CC Register is cleared so as to enable External Interrupts. If an interrupt occurs, the CPU becomes active.

The MCU can exit the Halt mode upon reception of an interrupt or a reset. Refer to the Interrupt Mapping Table. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

Figure 13. HALT Flow Chart



3.5 Register Description

MISCELLANEOUS REGISTER (MISCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
-	-	SMS	-	-	-	-	POC0

Bit 7:6 = Reserved

Bit 5 = **SMS** *Slow Mode Select*.

This bit is set and cleared by software. It is used to select the slow or fast mode CPU frequency.

0: $f_{CPU} = \text{Oscillator frequency} / 6$ (slow mode)

1: $f_{CPU} = \text{Oscillator frequency} / 3$ (normal mode)

Bit 4:1 = Reserved

Bit 0 = **POC0** *PWM/BRM Output Configuration Bit*. This bit is set and cleared by software. They select the PWM/BRM output configuration for pins DA1-DA4.

0: Push-pull

1: Open drain

INTERRUPT REGISTER (ITRFRE)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
EIOF	-	-	-	EIOITE	-	-	-

Bit 7 = **EIOF** *Falling Edge Detector Flag*.

This bit is set by hardware when a falling edge occurs on the pin assigned to EI0. It is cleared by software. When this bit is set, an interrupt is generated if the corresponding ITE bit = 1 and the I bit in the CC register = 0.

0: No falling edge detected

1: Falling edge detected

Bit 6:4 = Reserved

Bit 3:0 = **EIOITE** *Interrupt Enable Bit*.

This bit is set and cleared by software.

0: Interrupt disabled

1: Interrupt enabled

4 ON-CHIP PERIPHERALS

4.1 I/O PORTS

4.1.1 Introduction

The I/O ports offer different functional modes:

- transfer of data through digital inputs and outputs and for specific pins:
- analog signal input (ADC)
- alternate signal input/output for the on-chip peripherals.
- external interrupt generation

An I/O port is composed of up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

4.1.2 Functional Description

Each port is associated to 2 main registers:

- Data Register (DR)
 - Data Direction Register (DDR)
- and some of them to an optional register:
- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, however some specific ports do not provide this register. The generic I/O block diagram is shown on [Figure 14](#).

4.1.2.1 Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Notes:

1. All the inputs are triggered by a CMOS Schmitt trigger.
2. When switching from input mode to output mode, the DR register should be written first to

output the correct value as soon as the port is configured as an output.

4.1.2.2 External Interrupt Generation

An I/O can be used to generate an external Interrupt request to the CPU. External Interrupts are enabled and their polarity selected using the OR, MISC and ITRFRE registers (where available).

Each external interrupt vector is linked to a dedicated group of I/O port pins (see Interrupts section). If more than one input pin is selected simultaneously as interrupt source, this is logically ORed. For this reason if one of the interrupt pins is tied low, it masks the other ones.

4.1.2.3 Output Mode

The pin is configured in output mode by setting the corresponding DDR register bit.

In this mode, writing “0” or “1” to the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Note: In this mode, the interrupt function is disabled.

4.1.2.4 Digital Alternate Function

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over standard I/O programming. When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin has to be configured in input mode. In this case, the pin's state is also digitally readable by addressing the DR register.

Note: When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

Warning: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

I/O PORTS (Cont'd)

4.1.2.5 Analog Alternate Function

When the pin is used as an ADC input the I/O must be configured as input, floating. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

Warning: The analog input voltage level must be within the limits stated in the Absolute Maximum Ratings.

I/O PORTS (Cont'd)

Figure 14. I/O Block Diagram

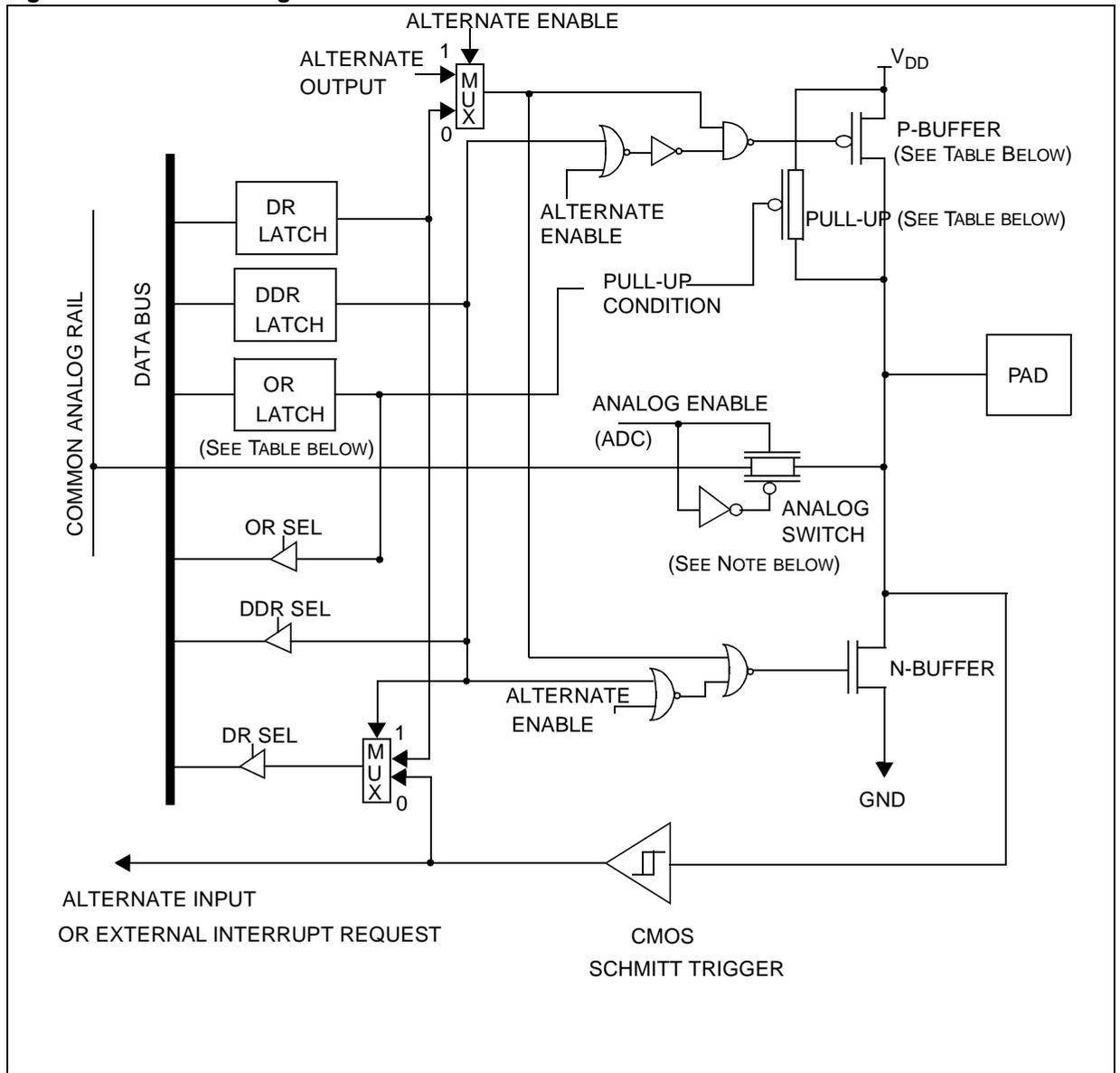


Table 7. Port Mode Configuration

Configuration Mode	Pull-up	P-buffer
Floating	0	0
Pull-up	1	0
Push-pull	0	1
True Open Drain	not present	not present
Open Drain (logic level)	not present	0

Legend:

- 0 - present, not activated
- 1 - present and activated

Notes:

- No OR Register on some ports (see register map).
- ADC Switch on ports with analog alternate functions.

I/O PORTS (Cont'd)

4.1.2.6 Device Specific Configurations

Table 8. ST72272K Port Configuration

Port	Pin name	Input (DDR=0)		Output (DDR=1)	
		OR=0*	OR=1	OR=0	OR=1
Port A	PA4:7	floating		true open drain, high sink capability	
Port B	PB0:2, PB7	pull-up	floating (for analog conversion only)	push-pull	
Port C	PC0	pull-up		push-pull	
	PC2:PC5	floating		open drain	
	PC6	pull-up		push-pull	
Port D	PD1:PD2, PD6	pull-up		push-pull	

*Reset state.

Note: The DA1-DA4 output pins are configurable as push pull or open drain using the POC0 Bit in the Miscellaneous Register.

I/O PORTS (Cont'd)**4.1.3 Register Description****DATA REGISTERS (DR)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D7-D0** *Data Register 8 bits.*

The behaviour of the DR register depends on the selected input/output configuration. Writing the DR register is always taken in account even if the pin is configured as an input. Reading the DR register returns either the DR register latch content (pin configured as output) or the digital value applied to the I/O pin (pin configured as input).

DATA DIRECTION REGISTERS (DDR)

Read/Write

Reset Value: 0000 0000 (00h) (input mode)

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = **DD7-DD0** *Data Direction Register 8 bits.*

The DDR register gives the input/output direction configuration of the pins. Each bit is set and cleared by software.

0: Input mode

1: Output mode

OPTION REGISTER (OR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AD7	-	-	-	-	AD2	AD1	AD0

Bit 7 = **AD7** *Digital/Analog Input Configuration.*

0: The pull-up is connected and pin configured as digital input (reset condition)

1: The pull-up is disconnected and the pin is configured as analog input.

Bit 6:3 = Reserved

Bit 2:0 = **AD[2:0]** *Digital/Analog Input Configuration Bits.*

0: The pull-up is connected and pin configured as digital input (reset condition)

1: The pull-up is disconnected and the pin is configured as analog input.

I/O PORTS (Cont'd)

Table 9. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0000h	PADR Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0001h	PADDR Reset Value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0002h	PCDR Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0003h	PCDDR Reset Value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0004h	PDDR Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0005h	PDDDR Reset Value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0006h	PBDR Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0007h	PBDDR Reset Value	DD7 0	DD6 0	DD5 0	DD4 0	DD3 0	DD2 0	DD1 0	DD0 0
0008h	PBOR Reset Value	AD7 0	- 0	- 0	- 0	- 0	AD2 0	AD1 0	AD0 0

4.2 WATCHDOG TIMER (WDG)

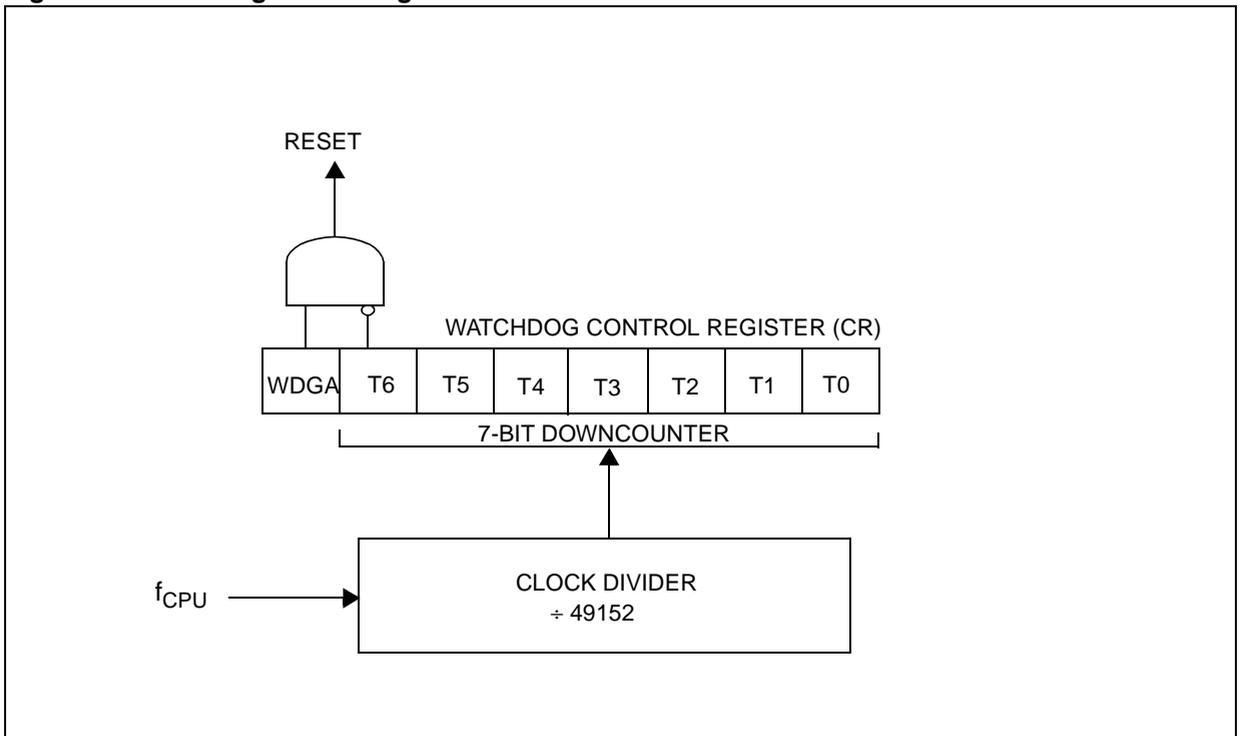
4.2.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

4.2.2 Main Features

- Programmable timer (64 increments of 49,152 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) after a HALT instruction or when the T6 bit reaches zero

Figure 15. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)**4.2.3 Functional Description**

The counter value stored in the CR register (bits T6:T0), is decremented every 49,152 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 become cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to stored in the CR register must be between FFh and C0h (see [Table 10](#)):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 10. Watchdog Timing ($f_{CPU} = 8 \text{ MHz}$)

	CR Register initial value	WDG timeout period (ms)
Max	FFh	393.216
Min	C0h	6.144

Notes: Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

Table 11. WDG Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0C Reset Value	CR	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

4.2.4 Register Description**CONTROL REGISTER (CR)**

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7= **WDGA** Activation bit.

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

Bit 6:0 = **T[6:0]** 7-bit timer (MSB to LSB).

These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 become cleared) if WDGA=1.

4.3 16-BIT TIMER

4.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

4.3.2 Main Features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with
 - 2 dedicated 16-bit registers
 - 2 dedicated programmable signals
 - 2 dedicated status flags
 - 1 dedicated maskable interrupt
- Input capture functions with
 - 2 dedicated 16-bit registers
 - 2 dedicated active edge selection signals
 - 2 dedicated status flags
 - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- 5 alternate functions on I/O ports*

The Block Diagram is shown in [Figure 16](#).

***Note:** Some external pins are not available on all devices. Refer to the device pin out description.

When reading an input signal which is not available on an external pin, the value will always be '1'.

4.3.3 Functional Description

4.3.3.1 Counter

The principal block of the Programmable Timer is a 16-bit free running increasing counter and its associated 16-bit registers:

Counter Registers

- Counter High Register (CHR) is the most significant byte (MSB).
- Counter Low Register (CLR) is the least significant byte (LSB).

Alternate Counter Registers

- Alternate Counter High Register (ACHR) is the most significant byte (MSB).
- Alternate Counter Low Register (ACLR) is the least significant byte (LSB).

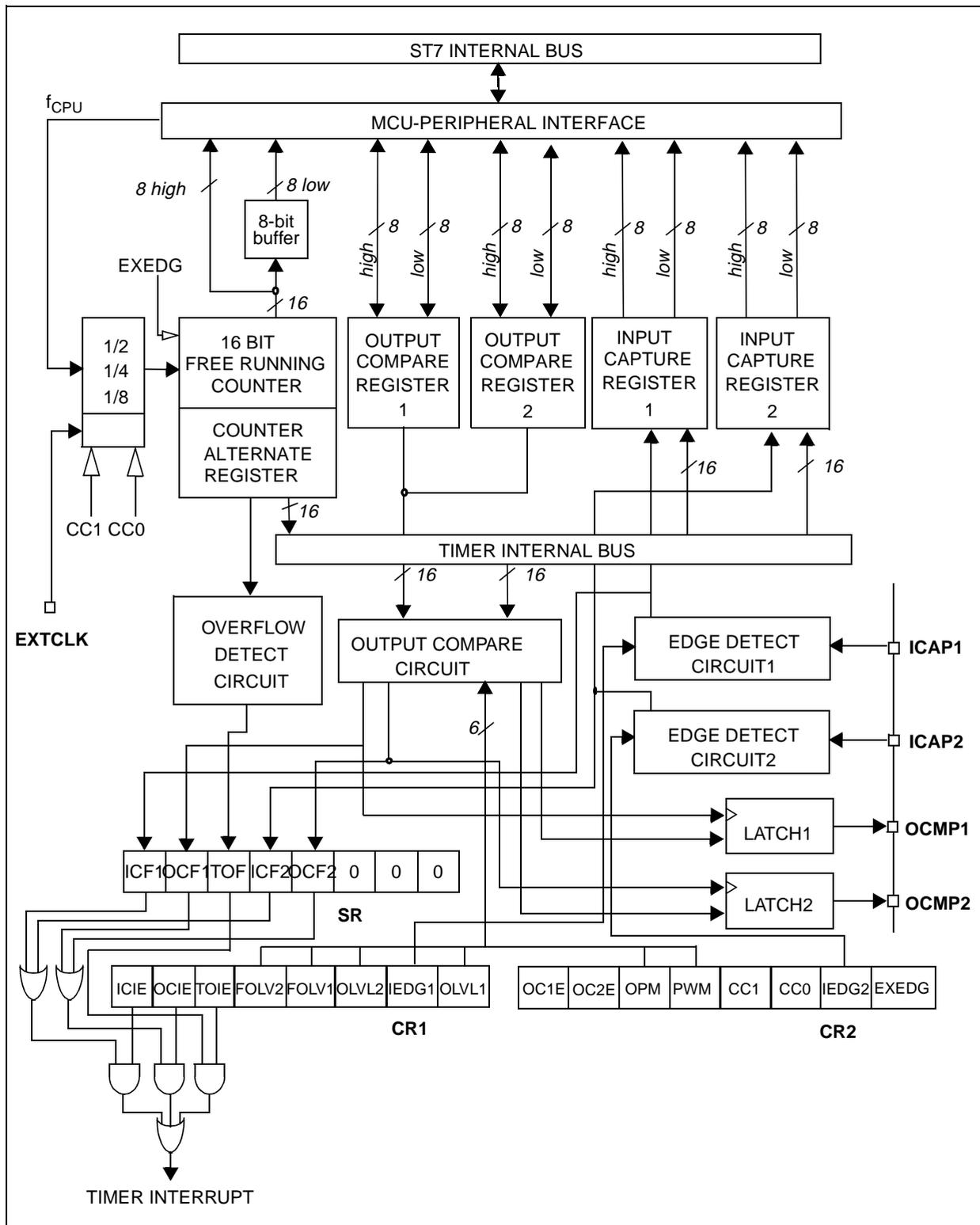
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (overflow flag), (see note at the end of paragraph titled 16-bit read sequence).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in [Table 12](#). The value in the counter register repeats every 131.072, 262.144 or 524.288 internal processor clock cycles depending on the CC1 and CC0 bits.

16-BIT TIMER (Cont'd)

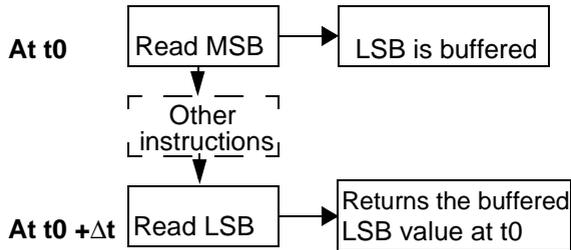
Figure 16. Timer Block Diagram



16-BIT TIMER (Cont'd)

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).

Beginning of the sequence



Sequence completed

The user must read the MSB first, then the LSB value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

An overflow occurs when the counter rolls over from FFFFh to 0000h then:

- The TOF bit of the SR register is set.
- A timer interrupt is generated if:
 - TOIE bit of the CR1 register is set and
 - I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Notes: The TOF bit is not cleared by accesses to ACLR register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

4.3.3.2 External Clock

The external clock (where available) is selected if CC0=1 and CC1=1 in CR2 register.

The status of the EXEDG bit determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronised with the falling edge of the internal CPU clock.

At least four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

16-BIT TIMER (Cont'd)

Figure 17. Counter Timing Diagram, internal clock divided by 2

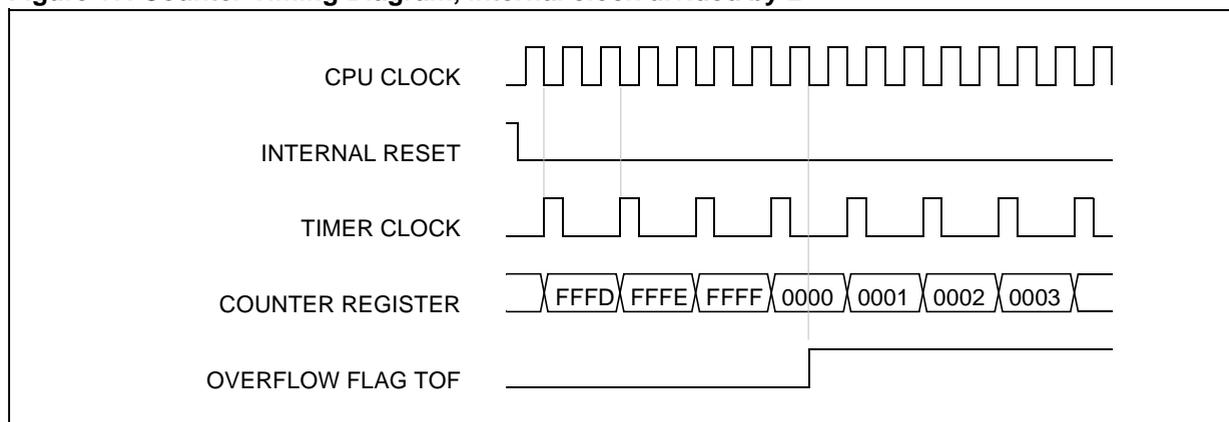


Figure 18. Counter Timing Diagram, internal clock divided by 4

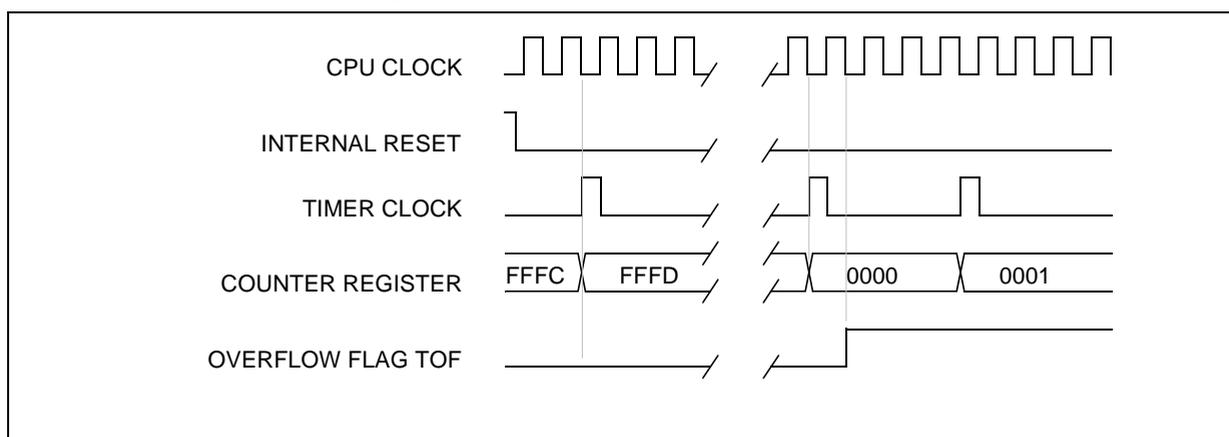
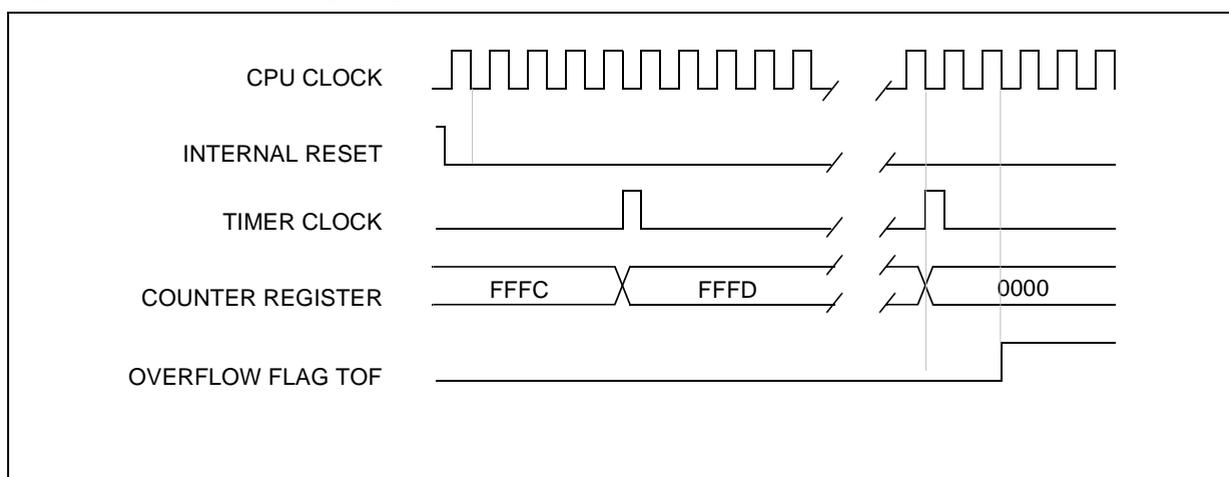


Figure 19. Counter Timing Diagram, internal clock divided by 8

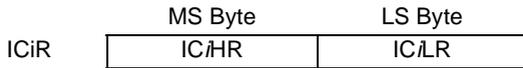


16-BIT TIMER (Cont'd)

4.3.3.3 Input Capture

In this section, the index, i , may be 1 or 2.

The two input capture 16-bit registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition detected by the ICAP i pin (see figure 5).



IC i R register is a read-only register.

The active transition is software programmable through the IEDG i bit of the Control Register (CR i).

Timing resolution is one count of the free running counter: ($f_{CPU}/(CC1.CC0)$).

Procedure

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC1-CC0) (see [Table 12](#)).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit.

And select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture.
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit.

When an input capture occurs:

- ICF i bit is set.
- The IC i R register contains the value of the free running counter on the active transition on the ICAP i pin (see [Figure 21](#)).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request is done in two steps:

1. Reading the SR register while the ICF i bit is set.
2. An access (read or write) to the IC i LR register.

Note: After reading the IC i HR register, transfer of input capture data is inhibited until the IC i LR register is also read.

The IC i R register always contains the free running counter value which corresponds to the most recent input capture.

During HALT mode, if at least one valid input capture edge occurs on the ICAP i pin, the input capture detection circuitry is armed. This does not set any timer flags, and does not “wake-up” the MCU. If the MCU is awoken by an interrupt, the input capture flag will become active, and data corresponding to the first valid edge during HALT mode will be present.

16-BIT TIMER (Cont'd)

Figure 20. Input Capture Block Diagram

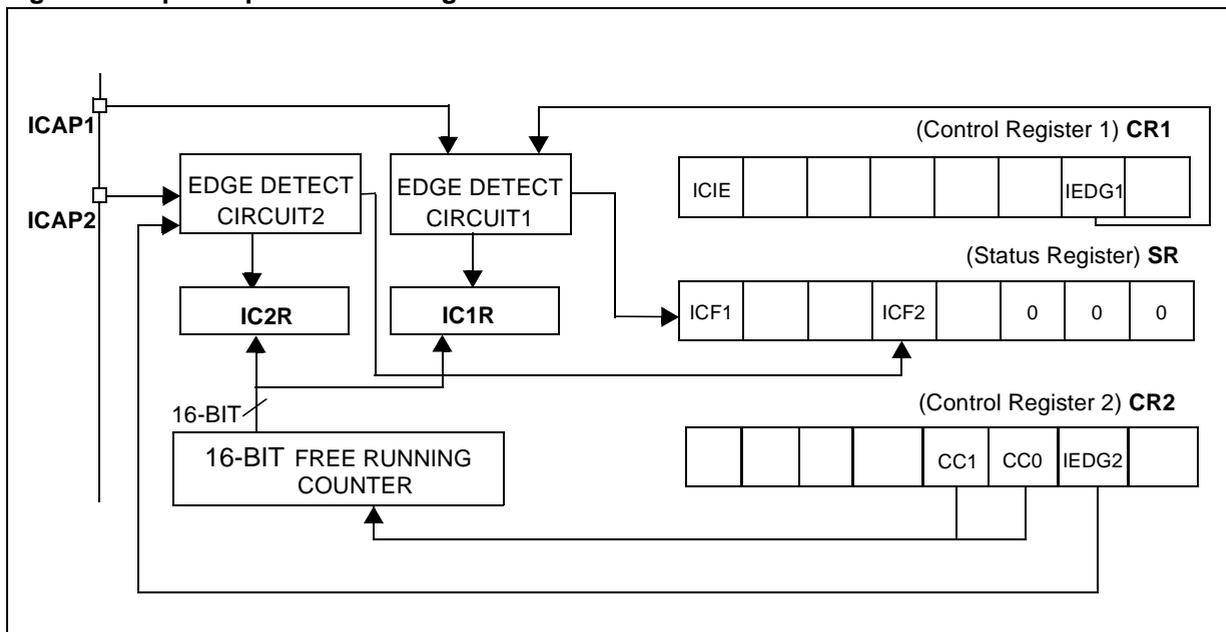
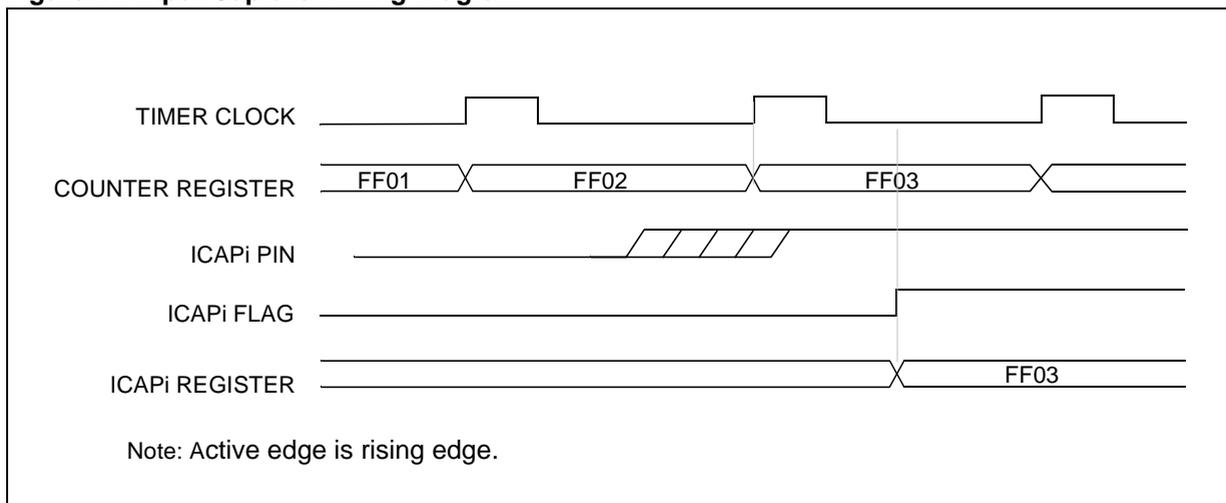


Figure 21. Input Capture Timing Diagram



16-BIT TIMER (Cont'd)

4.3.3.4 Output Compare

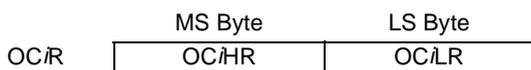
In this section, the index, i , may be 1 or 2.

This function can be used to control an output waveform or indicating when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OCIE bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the free running counter each timer clock cycle.



These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*R* value to 8000h.

Timing resolution is one count of the free running counter: $(f_{\text{CPU}}/(\text{CC1.CC0}))$.

Procedure

To use the output compare function, select the following in the CR2 register:

- Set the OC*E* bit if an output is needed then the OCMP*i* pin is dedicated to the output compare i function.
- Select the timer clock (CC1-CC0) (see [Table 12](#)).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When match is found:

- OCF*i* bit is set.
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset and stays low until valid compares change it to a high level).

- A timer interrupt is generated if the OCIE bit is set in the CR2 register and the I bit is cleared in the CC register (CC).

Clearing the output compare interrupt request is done by:

3. Reading the SR register while the OCF*i* bit is set.
4. An access (read or write) to the OC*LR* register.

Note: After a processor write cycle to the OC*HR* register, the output compare function is inhibited until the OC*LR* register is also written.

If the OC*E* bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when match is found but an interrupt could be generated if the OCIE bit is set.

The value in the 16-bit OC*R* register and the OLV*i* bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

The OC*R* register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{OC}iR = \frac{\Delta t * f_{\text{CPU}}}{t_{\text{PRESC}}}$$

Where:

- Δt = Desired output compare period (in seconds)
- f_{CPU} = Internal clock frequency
- t_{PRESC} = Timer clock prescaler (CC1-CC0 bits, see [Table 12](#))

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*R* register:

- Write to the OC*HR* register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*LR* register (enables the output compare function and clears the OCF*i* bit).

16-BIT TIMER (Cont'd)

Figure 22. Output Compare Block Diagram

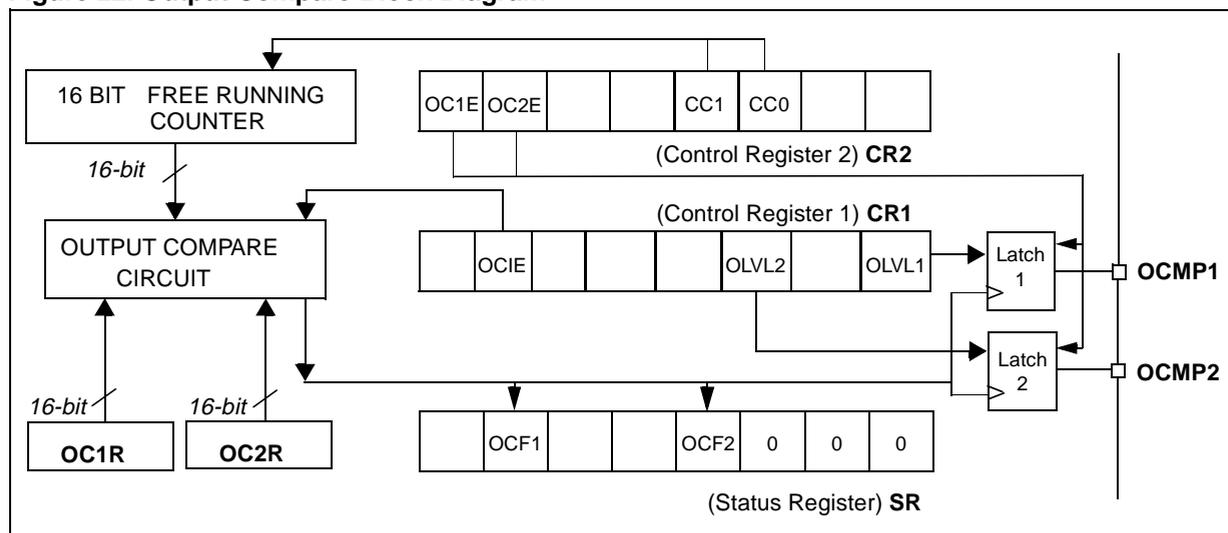
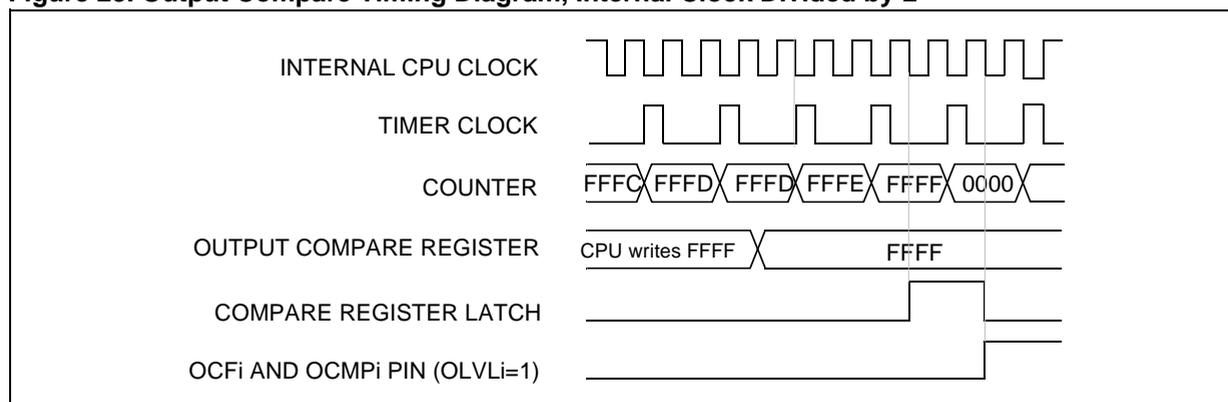


Figure 23. Output Compare Timing Diagram, Internal Clock Divided by 2



16-BIT TIMER (Cont'd)

4.3.3.5 Forced Compare Mode

In this section *i* may represent 1 or 2.

The following bits of the CR1 register are used:



When the FOLV*i* bit is set, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC1E bit=1).

The OCF*i* bit is not set, and thus no interrupt request is generated.

4.3.3.6 One Pulse Mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

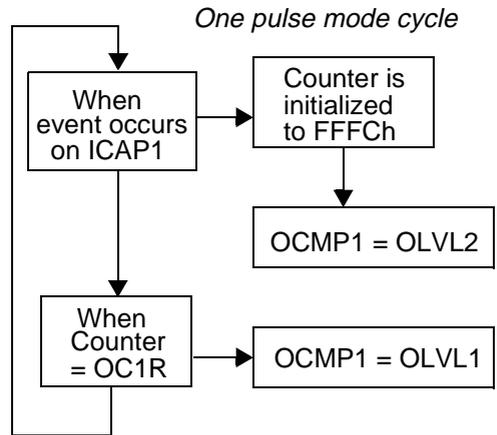
The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure

To use one pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in Section 4.3.3.7).
2. Select the following in the the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
 - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit.
3. Select the following in the CR2 register:
 - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
 - Set the OPM bit.

- Select the timer clock CC1-CC0 (see Table 12).



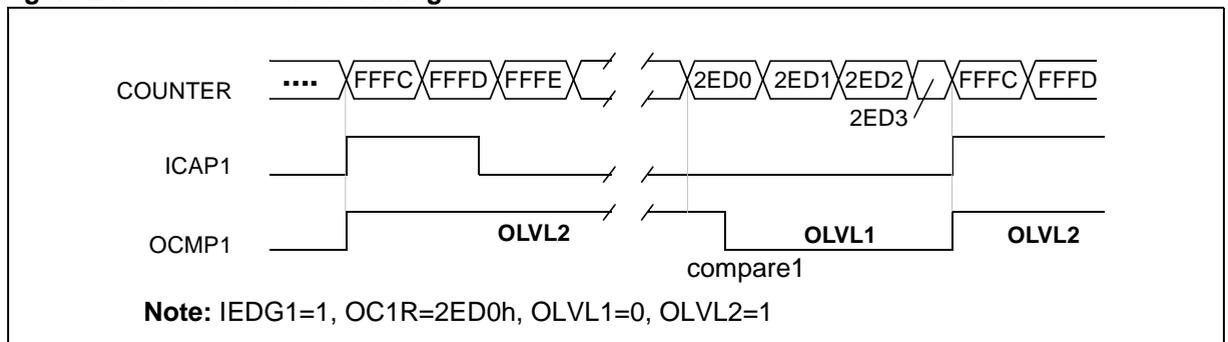
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin. When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 24).

Note: The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.

When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

Figure 24. One Pulse Mode Timing



16-BIT TIMER (Cont'd)

4.3.3.7 Pulse Width Modulation Mode

Pulse Width Modulation mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

The pulse width modulation mode uses the complete Output Compare 1 function plus the OC2R register.

Procedure

To use pulse width modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal.
2. Load the OC1R register with the value corresponding to the length of the pulse if (OLVL1=0 and OLVL2=1).
3. Select the following in the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC1R register.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
 - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
 - Set the PWM bit.
 - Select the timer clock (CC1-CC0) (see [Table 12](#)).

If OLVL1=1 and OLVL2=0 the length of the pulse is the difference between the OC2R and OC1R registers.

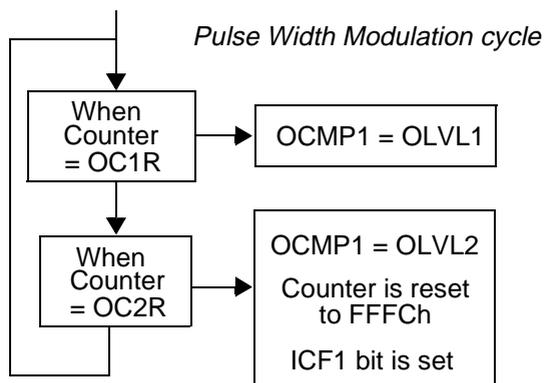
The OC/R register value required for a specific timing application can be calculated using the following formula:

$$\text{OC/R Value} = \frac{t \cdot f_{\text{CPU}}}{t_{\text{PRESC}}} - 5$$

Where:

- t = Desired output compare period (seconds)
- f_{CPU} = Internal clock frequency (see Miscellaneous register)
- t_{PRESC} = Timer clock prescaler (CC1-CC0 bits, see [Table 12](#))

The Output Compare 2 event causes the counter to be initialized to FFFCh (See [Figure 25](#)).



Note: After a write instruction to the OC/R register, the output compare function is inhibited until the OCLR register is also written.

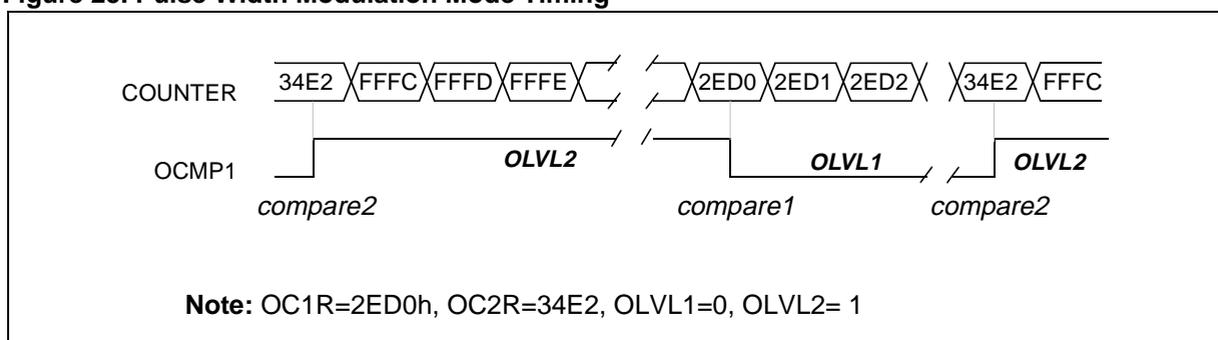
The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.

Therefore the Input Capture 1 function is inhibited but the Input Capture 2 is available.

The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.

When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

Figure 25. Pulse Width Modulation Mode Timing



16-BIT TIMER (Cont'd)**4.3.4 Register Description**

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable*.

0: Interrupt is inhibited.

1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = **OCIE** *Output Compare Interrupt Enable*.

0: Interrupt is inhibited.

1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = **TOIE** *Timer Overflow Interrupt Enable*.

0: Interrupt is inhibited.

1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = **FOLV2** *Forced Output Compare 2*.

This bit is set and cleared by software.

0: No effect on the OCMP2 pin.

1: Forces the OLVL2 bit to be copied to the OCMP2 pin.

Bit 3 = **FOLV1** *Forced Output Compare 1*.

This bit is set and cleared by software.

0: No effect on the OCMP1 pin.

1: Forces OLVL1 to be copied to the OCMP1 pin.

Bit 2 = **OLVL2** *Output Level 2*.

This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse Mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1*.

This bit determines which type of level transition on the ICAP1 pin will trigger the capture.

0: A falling edge triggers the capture.

1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1*.

The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

16-BIT TIMER (Cont'd)**CONTROL REGISTER 2 (CR2)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Enable*.

0: Output Compare 1 function is enabled, but the OCMP1 pin is a general I/O.

1: Output Compare 1 function is enabled, the OCMP1 pin is dedicated to the Output Compare 1 capability of the timer.

Bit 6 = **OC2E** *Output Compare 2 Enable*.

0: Output Compare 2 function is enabled, but the OCMP2 pin is a general I/O.

1: Output Compare 2 function is enabled, the OCMP2 pin is dedicated to the Output Compare 2 capability of the timer.

Bit 5 = **OPM** *One Pulse Mode*.

0: One Pulse Mode is not active.

1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation*.

0: PWM mode is not active.

1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

Bit 3, 2 = **CC1-CC0** *Clock Control*.

The value of the timer clock depends on these bits:

Table 12. Clock Control Bits

Timer Clock	CC1	CC0
$f_{CPU} / 4$	0	0
$f_{CPU} / 2$	0	1
$f_{CPU} / 8$	1	0
External Clock (where available)	1	1

Bit 1 = **IEDG2** *Input Edge 2*.

This bit determines which type of level transition on the ICAP2 pin will trigger the capture.

0: A falling edge triggers the capture.

1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge*.

This bit determines which type of level transition on the external clock pin EXTCLK will trigger the free running counter.

0: A falling edge triggers the free running counter.

1: A rising edge triggers the free running counter.

16-BIT TIMER (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

					0	0	0	
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0	

Bit 7 = **ICF1** *Input Capture Flag 1.*

0: No input capture (reset value).

1: An input capture has occurred or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*

0: No match (reset value).

1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow.*

0: No timer overflow (reset value).

1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*

0: No input capture (reset value).

1: An input capture has occurred. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*

0: No match (reset value).

1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2-0 = Reserved, forced by hardware to 0.

INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

Read/Write

Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

Read/Write

Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

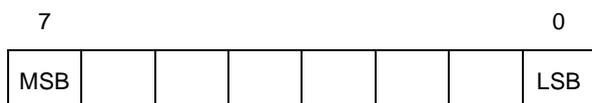
MSB							LSB

16-BIT TIMER (Cont'd)**OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)**

Read/Write

Reset Value: 1000 0000 (80h)

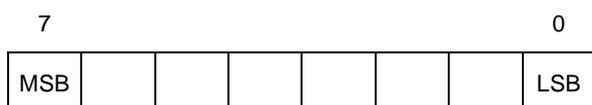
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

**OUTPUT COMPARE 2 LOW REGISTER (OC2LR)**

Read/Write

Reset Value: 0000 0000 (00h)

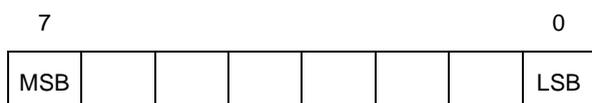
This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

**COUNTER HIGH REGISTER (CHR)**

Read Only

Reset Value: 1111 1111 (FFh)

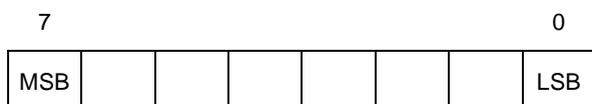
This is an 8-bit register that contains the high part of the counter value.

**COUNTER LOW REGISTER (CLR)**

Read Only

Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the SR register clears the TOF bit.

**ALTERNATE COUNTER HIGH REGISTER (ACHR)**

Read Only

Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

**ALTERNATE COUNTER LOW REGISTER (ACLR)**

Read Only

Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to SR register does not clear the TOF bit in SR register.

**INPUT CAPTURE 2 HIGH REGISTER (IC2HR)**

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

**INPUT CAPTURE 2 LOW REGISTER (IC2LR)**

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).



Table 13. 16-Bit Timer Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
11	CR2	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
12	CR1	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
13	SR	ICF1	OCF1	TOF	ICF2	OCF2	0	0	0
14	IC1HR	MSB							LSB
15	IC1LR	MSB							LSB
16	OC1HR	MSB							LSB
17	OC1LR	MSB							LSB
18	CHR	MSB							LSB
19	CLR	MSB							LSB
1A	ACHR	MSB							LSB
1B	ACLR	MSB							LSB
1C	IC2HR	MSB							LSB
1D	IC2LR	MSB							LSB
1E	OC2HR	MSB							LSB
1F	OC2LR	MSB							LSB
43	CONFIG Reset Value	- 0	- 0	- 0	- 0	- 1	ICAP 0	- 0	- 0

Warning: Write 0Ch in the CONFIG register to use the ICAP1 and ICAP2 pins (set bits 3 and 2).

4.4 I²C BUS INTERFACE (I2C)

4.4.1 Introduction

The I²C Bus Interface serves as an interface between the microcontroller and the serial I²C bus. It provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It supports fast I²C mode (400kHz).

4.4.2 Main Features

- Parallel-bus/I²C protocol converter
- Multi-master capability
- 7-bit Addressing
- Transmitter/Receiver flag
- End-of-byte transmission flag
- Transfer problem detection

I²C Master Features:

- Clock generation
- I²C bus busy flag
- Arbitration Lost Flag
- End of byte transmission flag
- Transmitter/Receiver Flag
- Start bit detection flag
- Start and Stop generation

I²C Slave Features:

- Stop bit detection
- I²C bus busy flag
- Detection of misplaced start or stop condition
- Programmable I²C Address detection
- Transfer problem detection
- End-of-byte transmission flag
- Transmitter/Receiver flag

4.4.3 General Description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled

handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

Mode Selection

The interface can operate in the four following modes:

- Slave transmitter/receiver
- Master transmitter/receiver

By default, it operates in slave mode.

The interface automatically switches from slave to master after it generates a START condition and from master to slave in case of arbitration loss or a STOP generation, this allows Multi-Master capability.

Communication Flow

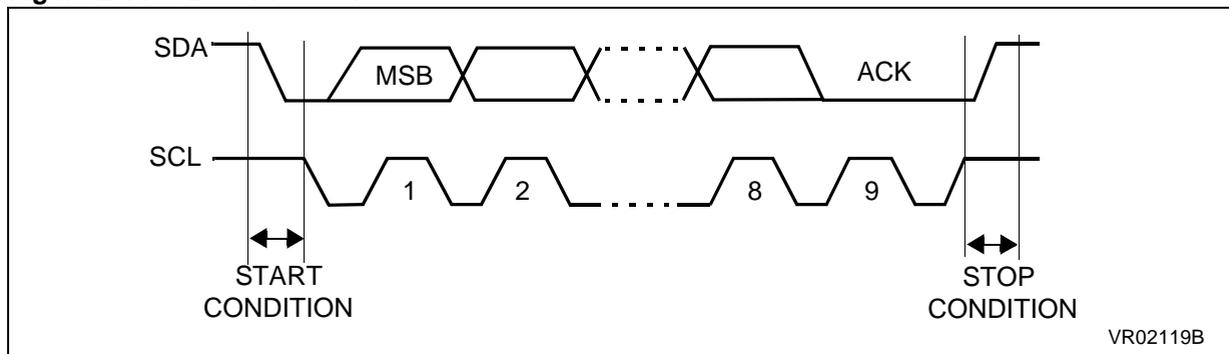
In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognising its own address (7-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition is the address byte; it is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to [Figure 26](#).

Figure 26. I²C BUS Protocol



I²C BUS INTERFACE (Cont'd)

Acknowledge may be enabled and disabled by software.

The I²C interface address and/or general call address can be selected by software.

The speed of the I²C interface may be selected between Standard (0-100KHz) and Fast I²C (100-400KHz).

SDA/SCL Line Control

Transmitter mode: the interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the Data Register.

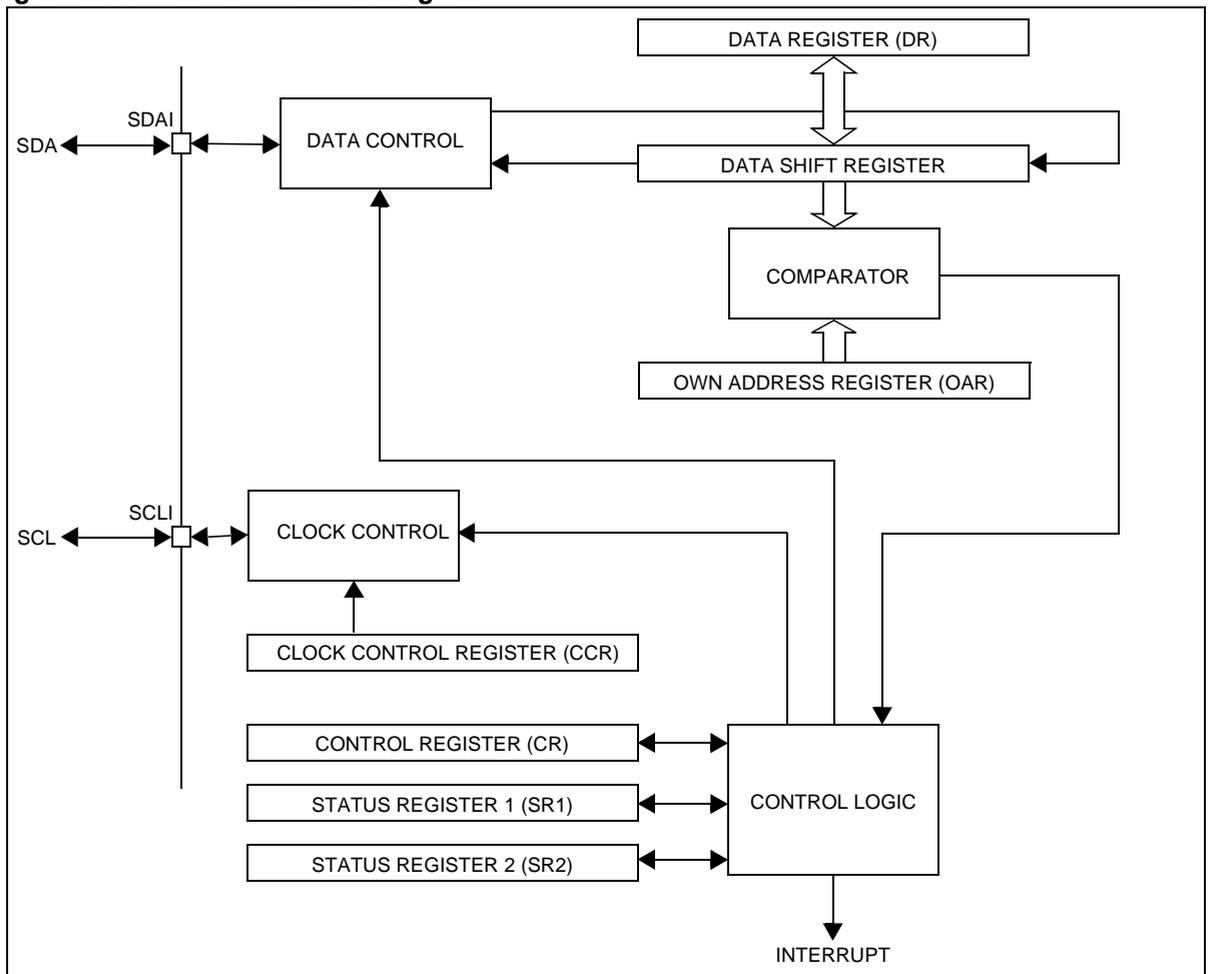
Receiver mode: the interface holds the clock line low after reception to wait for the microcontroller to read the byte in the Data Register.

The SCL frequency (F_{SCL}) is controlled by a programmable clock divider which depends on the I²C bus mode.

When the I²C cell is enabled, the SDA and SCL ports must be configured as floating open-drain output or floating input. In this case, the value of the external pull-up resistance used depends on the application.

When the I²C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 27. I²C Interface Block Diagram



I²C BUS INTERFACE (Cont'd)

4.4.4 Functional Description

Refer to the CR, SR1 and SR2 registers in [Section 4.4.5](#) for the bit definitions.

By default the I²C interface operates in Slave mode (M/SL bit is cleared) except when it initiates a transmit or receive sequence.

4.4.4.1 Slave Mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the address of the interface or the General Call address (if selected by software).

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set.
- EVF and ADSL bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV1).

Next, read the DR register to determine from the least significant bit if the slave must enter Receiver or Transmitter mode.

Slave Receiver

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV2).

Slave Transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV3).

When the acknowledge pulse is received:

- The EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Closing slave communication

After the last data byte is transferred a Stop Condition is generated by the master. The interface detects this condition and sets:

- EVF and STOPF bits with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see [Figure 28](#) Transfer sequencing EV4).

Error Cases

- **BERR:** Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set with an interrupt if the ITE bit is set.

If it is a Stop then the interface discards the data, released the lines and waits for another Start condition.

If it is a Start then the interface discards the data and waits for the next slave address on the bus.

- **AF:** Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set with an interrupt if the ITE bit is set.

Note: In both cases, SCL line is not held low; however, SDA line can remain low due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

I²C BUS INTERFACE (Cont'd)

4.4.4.2 Master Mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition and Transmit Slave address

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent:

- The EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address byte, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV5).

Then the slave address byte is sent to the SDA line via the internal shift register.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

- The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

Master Receiver

Following the address transmission and after SR1 and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Master Transmitter

Following the address transmission and after SR1 register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see [Figure 28](#) Transfer sequencing EV8).

When the acknowledge bit is received, the interface sets:

- EVF and BTF bits with an interrupt if the ITE bit is set.

To close the communication: after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Error Cases

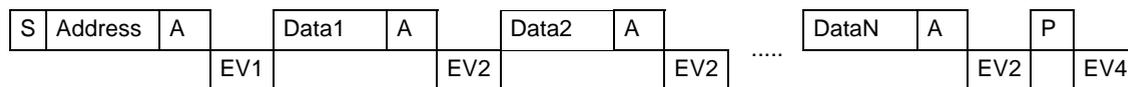
- **BERR:** Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.
- **AF:** Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the START or STOP bit.
- **ARLO:** Detection of an arbitration lost condition. In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: In all these cases, the SCL line is not held low; however, the SDA line can remain low due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

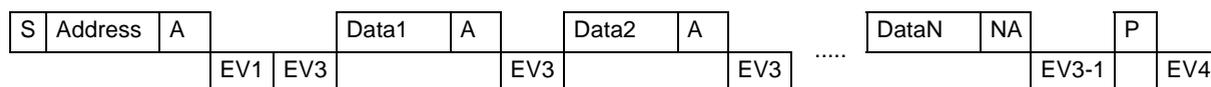
I²C BUS INTERFACE (Cont'd)

Figure 28. Transfer Sequencing

Slave receiver:



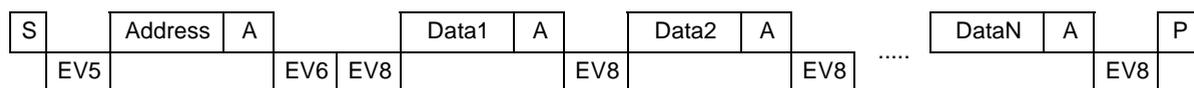
Slave transmitter:



Master receiver:



Master transmitter:



Legend:

S=Start, P=Stop, A=Acknowledge, NA=Non-acknowledge

EVx=Event (with interrupt if ITE=1)

EV1: EVF=1, ADSL=1, cleared by reading SR1 register.

EV2: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV3-1: EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). **Note:** If lines are released by STOP=1, STOP=0, the subsequent EV4 is not seen.

EV4: EVF=1, STOPF=1, cleared by reading SR2 register.

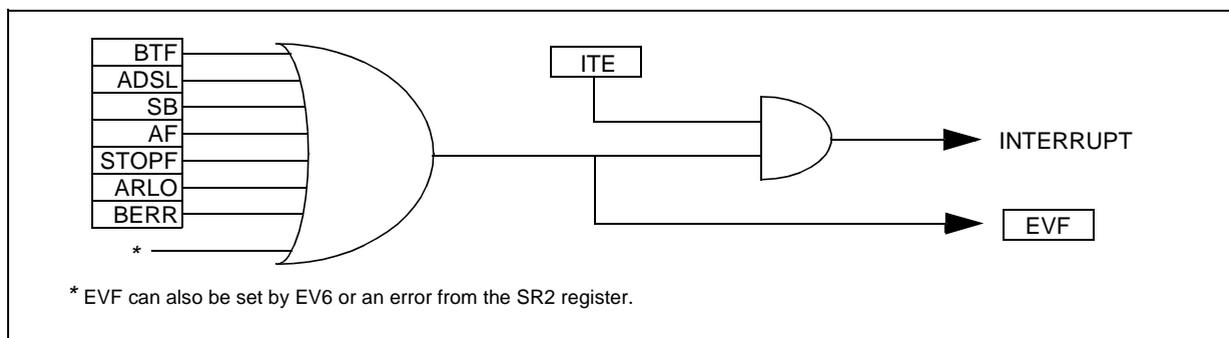
EV5: EVF=1, SB=1, cleared by reading SR1 register followed by writing DR register.

EV6: EVF=1, cleared by reading SR1 register followed by writing CR register (for example PE=1).

EV7: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV8: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

Figure 29. Event Flags and Interrupt Generation



I²C BUS INTERFACE (Cont'd)**4.4.5 Register Description****I²C CONTROL REGISTER (CR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	PE	ENGC	START	ACK	STOP	ITE

Bit 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **PE** *Peripheral enable*.

This bit is set and cleared by software.

0: Peripheral disabled

1: Master/Slave capability

Notes:

- When PE=0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE=0
- When PE=1, the corresponding I/O pins are selected by hardware as alternate functions.
- To enable the I²C interface, write the CR register **TWICE** with PE=1 as the first write only activates the interface (only PE is set).

Bit 4 = **ENGC** *Enable General Call*.

This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0). The 00h General Call address is acknowledged (01h ignored).

0: General Call disabled

1: General Call enabled

Bit 3 = **START** *Generation of a Start condition*.

This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0) or when the Start condition is sent (with interrupt generation if ITE=1).

- In master mode:
 - 0: No start generation
 - 1: Repeated start generation
- In slave mode:
 - 0: No start generation
 - 1: Start generation when the bus is free

Bit 2 = **ACK** *Acknowledge enable*.

This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0).

0: No acknowledge returned

1: Acknowledge returned after an address byte or a data byte is received

Bit 1 = **STOP** *Generation of a Stop condition*.

This bit is set and cleared by software. It is also cleared by hardware in master mode. Note: This bit is not cleared when the interface is disabled (PE=0).

– In master mode:

0: No stop generation

1: Stop generation after the current byte transfer or after the current Start condition is sent. The STOP bit is cleared by hardware when the Stop condition is sent.

– In slave mode:

0: No stop generation

1: Release the SCL and SDA lines after the current byte transfer (BTF=1). In this mode the STOP bit has to be cleared by software.

Bit 0 = **ITE** *Interrupt enable*.

This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).

0: Interrupts disabled

1: Interrupts enabled

Refer to [Figure 29](#) for the relationship between the events and the interrupt.

SCL is held low when the SB, BTF or ADSL flags or an EV6 event (See [Figure 28](#)) is detected.

I²C INTERFACE (Cont'd)**I²C STATUS REGISTER 1 (SR1)**

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	0	TRA	BUSY	BTF	ADSL	M/SL	SB

Bit 7 = EVF Event flag.

This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in [Figure 28](#). It is also cleared by hardware when the interface is disabled (PE=0).

0: No event

1: One of the following events has occurred:

- BTF=1 (Byte received or transmitted)
- ADSL=1 (Address matched in Slave mode while ACK=1)
- SB=1 (Start condition generated in Master mode)
- AF=1 (No acknowledge received after byte transmission)
- STOPF=1 (Stop condition detected in Slave mode)
- ARLO=1 (Arbitration lost in Master mode)
- BERR=1 (Bus error, misplaced Start or Stop condition detected)
- Address byte successfully transmitted in Master mode.

Bit 6 = Reserved. Forced to 0 by hardware.

Bit 5 = TRA Transmitter/Receiver.

When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after detection of Stop condition (STOPF=1), loss of bus arbitration (ARLO=1) or when the interface is disabled (PE=0).

0: Data byte received (if BTF=1)

1: Data byte transmitted

Bit 4 = BUSY Bus busy.

This bit is set by hardware on detection of a Start condition and cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. This information is still updated when the interface is disabled (PE=0).

0: No communication on the bus

1: Communication ongoing on the bus

Bit 3 = BTF Byte transfer finished.

This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

– Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See [Figure 28](#)). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.

– Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done

1: Byte transfer succeeded

Bit 2 = ADSL Address matched (Slave mode).

This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

0: Address mismatched or not received

1: Received address matched

Bit 1 = M/SL Master/Slave.

This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1). It is also cleared when the interface is disabled (PE=0).

0: Slave mode

1: Master mode

Bit 0 = SB Start bit (Master mode).

This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).

0: No Start condition

1: Start condition generated

I²C INTERFACE (Cont'd)**I²C STATUS REGISTER 2 (SR2)**

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL

Bit 7:5 = Reserved. Forced to 0 by hardware.

Bit 4 = **AF** *Acknowledge failure.*

This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while AF=1.

0: No acknowledge failure

1: Acknowledge failure

Bit 3 = **STOPF** *Stop detection (Slave mode).*

This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while STOPF=1.

0: No Stop condition detected

1: Stop condition detected

Bit 2 = **ARLO** *Arbitration lost.*

This bit is set by hardware when the interface los-

es the arbitration of the bus to another master. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

After an ARLO event the interface switches back automatically to Slave mode (M/SL=0).

The SCL line is not held low while ARLO=1.

0: No arbitration lost detected

1: Arbitration lost detected

Bit 1 = **BERR** *Bus error.*

This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while BERR=1.

0: No misplaced Start or Stop condition

1: Misplaced Start or Stop condition

Bit 0 = **GCAL** *General Call (Slave mode).*

This bit is set by hardware when a general call address is detected on the bus while ENGC=1. It is cleared by hardware detecting a Stop condition (STOPF=1) or when the interface is disabled (PE=0).

0: No general call address detected on bus

1: general call address detected on bus

I²C INTERFACE (Cont'd)**I²C CLOCK CONTROL REGISTER (CCR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode.*

This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).

0: Standard I²C mode1: Fast I²C modeBit 6:0 = **CC6-CC0** *7-bit clock divider.*These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).– Standard mode (FM/SM=0): $F_{SCL} \leq 100\text{kHz}$

$$F_{SCL} = f_{CPU} / (2x([CC6..CC0]+2))$$

– Fast mode (FM/SM=1): $F_{SCL} > 100\text{kHz}$

$$F_{SCL} = f_{CPU} / (3x([CC6..CC0]+2))$$

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.**I²C DATA REGISTER (DR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D7-D0** *8-bit Data Register.*

These bits contains the byte to be received or transmitted on the bus.

– Transmitter mode: Byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.

Then, the next data bytes are received one by one after reading the DR register.

I²C OWN ADDRESS REGISTER (OAR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Bit 7:1 = **ADD7-ADD1** *Interface address.*These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).Bit 0 = **ADD0** *Address direction bit.*

This bit is don't care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

I2C INTERFACE (Cont'd)

Table 14. I²C Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
5F	CR			PE	ENGC	START	ACK	STOP	ITE
5E	SR1	EVF		TRA	BUSY	BTF	ADSL	M/SL	SB
5D	SR2				AF	STOPF	ARLO	BERR	GCAL
5C	CCR	FM/SM	CC6 .. CC0						
5B	OAR	ADD7 .. ADD0							
59	DR	DR7 .. DR0							

4.5 PWM/BRM GENERATOR (DAC)

4.5.1 Introduction

This PWM/BRM peripheral includes a 6-bit Pulse Width Modulator (PWM) and a 4-bit Binary Rate Multiplier (BRM) Generator. It allows the digital to analog conversion (DAC) when used with external filtering.

Note: The number of PWM and BRM channels available depends on the device. Refer to the device pin description and register map.

4.5.2 Main Features

- Fixed frequency: $f_{\text{CPU}}/64$
- Resolution: T_{CPU}
- Steps of $V_{\text{DD}}/2^{10}$ (5mV if $V_{\text{DD}}=5\text{V}$)

4.5.3 Functional Description

The 10 bits of the 10-bit PWM/BRM are distributed as 6 PWM bits and 4 BRM bits. The generator consists of a 10-bit counter (common for all channels), a comparator and the PWM/BRM generation logic.

PWM Generation

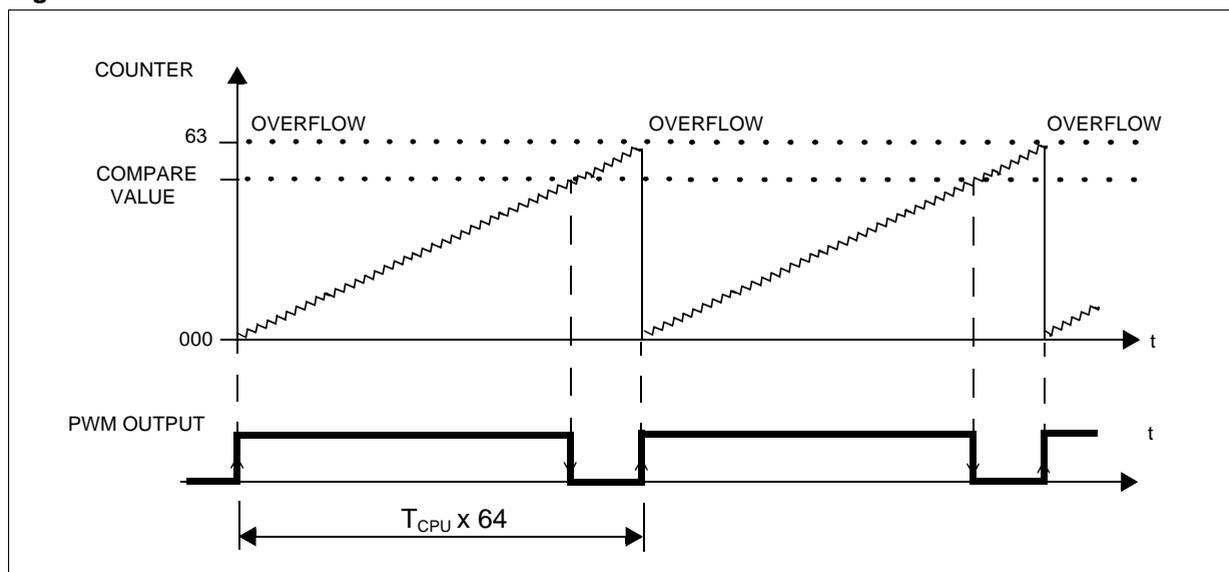
The counter increments continuously, clocked at internal CPU clock. Whenever the 6 least significant bits of the counter (defined as the PWM counter) overflow, the output level for all active channels is set.

The state of the PWM counter is continuously compared to the PWM binary weight for each channel, as defined in the relevant PWM register, and when a match occurs the output level for that channel is reset.

This Pulse Width modulated signal must be filtered, using an external RC network placed as close as possible to the associated pin. This provides an analog voltage proportional to the average charge passed to the external capacitor. Thus for a higher mark/space ratio (high time much greater than low time) the average output voltage is higher. The external components of the RC network should be selected for the filtering level required for control of the system variable.

Each output may individually have its polarity inverted by software, and can also be used as a logical output.

Figure 30. PWM Generation



PWM/BRM GENERATOR (Cont'd)

PWM/BRM Outputs

The PWM/BRM outputs are assigned to dedicated pins.

In these pins, the PWM/BRM outputs are connected to a serial resistor which must be taken into account to calculate the RC filter (see [Figure 31](#)).

In any case, the RC filter time must be higher than $T_{CPU} \times 64$.

Figure 31. Typical PWM Output Filter

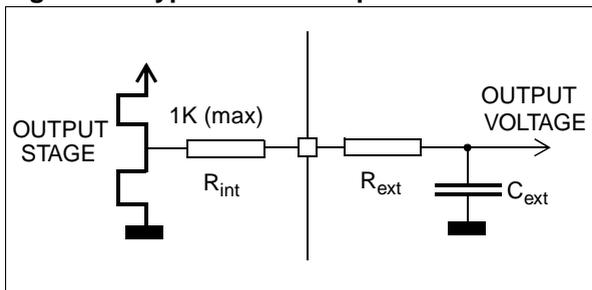


Table 15. 6-Bit PWM Ripple After Filtering

C _{ext} (μF)	V _{ripple} (mV)
0.128	78
1.28	7.8
12.8	0.78

With RC filter (R=1KΩ),

f_{cpu} = 8 MHz

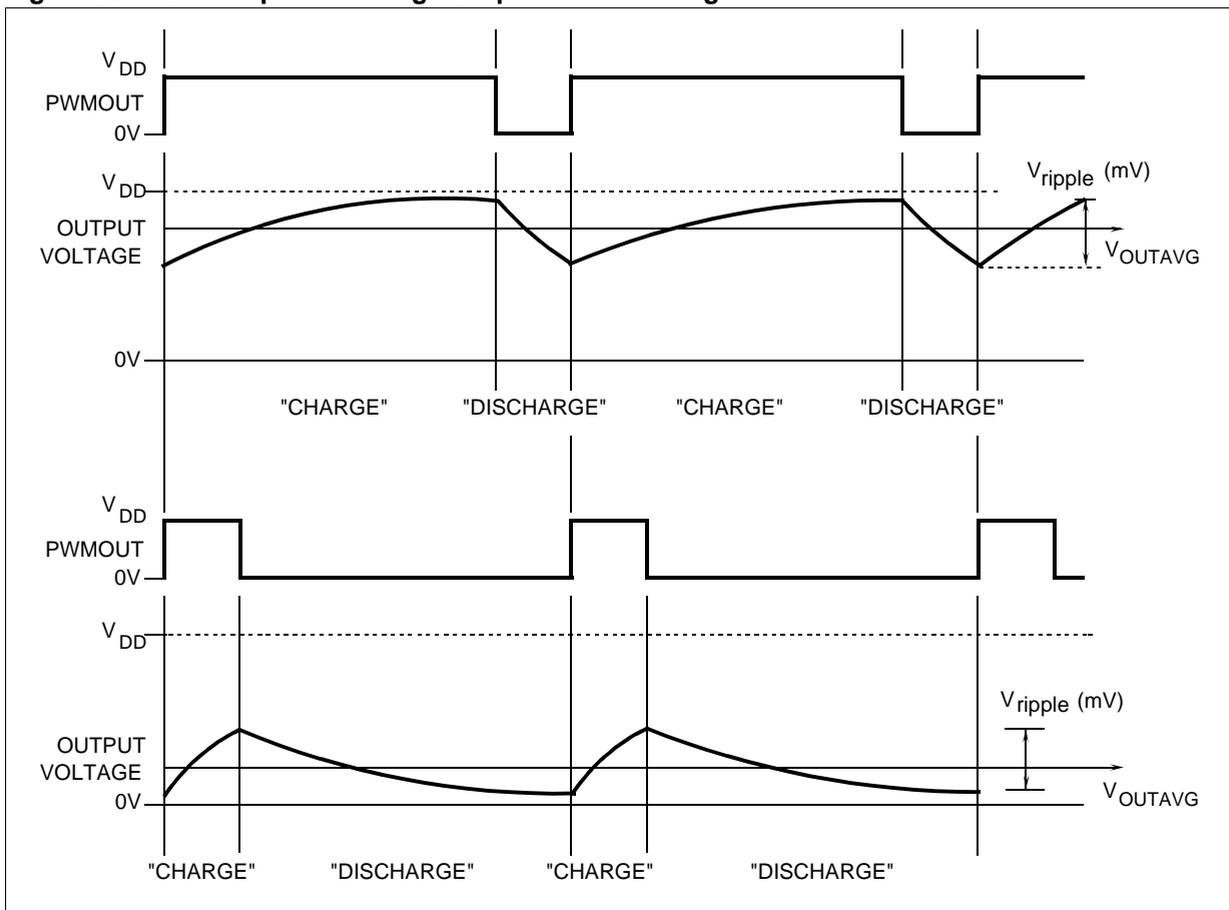
V_{DD} = 5V

PWM Duty Cycle 50%

R=R_{int}+R_{ext} (R_{ext} is optional).

Note: after a reset these pins are tied low by default and are not in a high impedance state.

Figure 32. PWM Simplified Voltage Output After Filtering



PWM/BRM GENERATOR (Cont'd)

BRM Generation

The BRM bits allow the addition of a pulse to widen a standard PWM pulse for specific PWM cycles. This has the effect of “fine-tuning” the PWM Duty cycle (without modifying the base duty cycle), thus, with the external filtering, providing additional fine voltage steps.

The incremental pulses (with duration of T_{CPU}) are added to the beginning of the original PWM pulse. The PWM intervals which are added to are specified in the 4-bit BRM register and are encoded as shown in the following table. The BRM values shown may be combined together to provide a summation of the incremental pulse intervals specified.

The pulse increment corresponds to the PWM resolution.

For example, if

- Data 18h is written to the PWM register
- Data 06h (00000110b) is written to the BRM register
- with a 8MHz internal clock (125ns resolution)

Then 3.0 μ s-long pulse will be output at 8 μ s intervals, except for cycles numbered 2,4,6,10,12,14, where the pulse is broadened to 3.125 μ s.

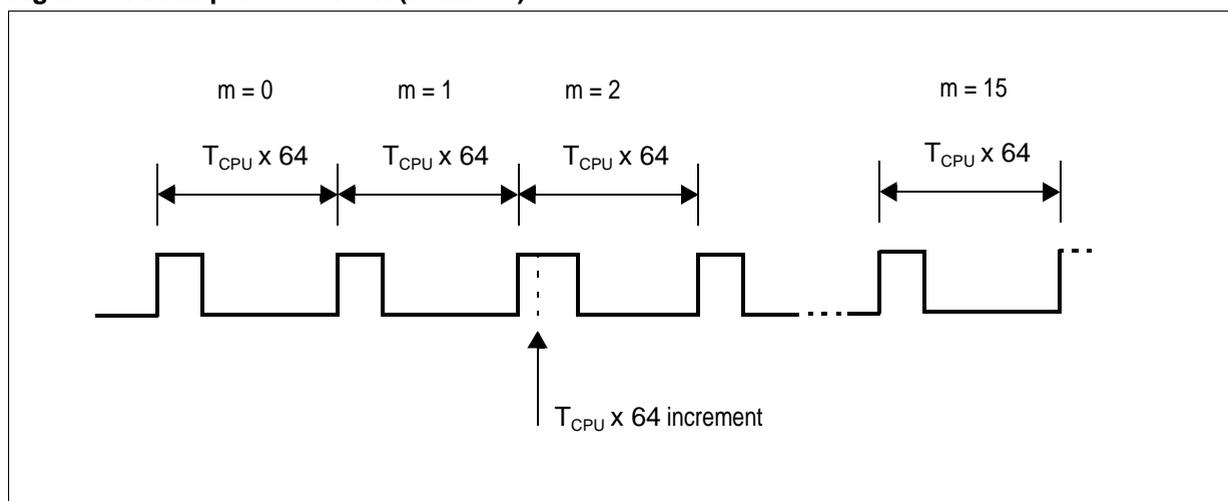
Note. If 00h is written to both PWM and BRM registers, the generator output will remain at “0”. Conversely, if both registers hold data 3Fh and 0Fh, respectively, the output will remain at “1” for all intervals 1 to 15, but it will return to zero at interval 0 for an amount of time corresponding to the PWM resolution (T_{CPU}).

An output can be set to a continuous “1” level by clearing the PWM and BRM values and setting POL = “1” (inverted polarity) in the PWM register. This allows a PWM/BRM channel to be used as an additional I/O pin if the DAC function is not required.

Table 16. Bit BRM Added Pulse Intervals (Interval #0 not selected).

BRM 4 - Bit Data	Incremental Pulse Intervals
0000	none
0001	i = 8
0010	i = 4,12
0100	i = 2,6,10,14
1000	i = 1,3,5,7,9,11,13,15

Figure 33. BRM pulse addition (PWM > 0)



PWM/BRM GENERATOR (Cont'd)

Figure 34. Simplified Filtered Voltage Output Schematic with BRM Added

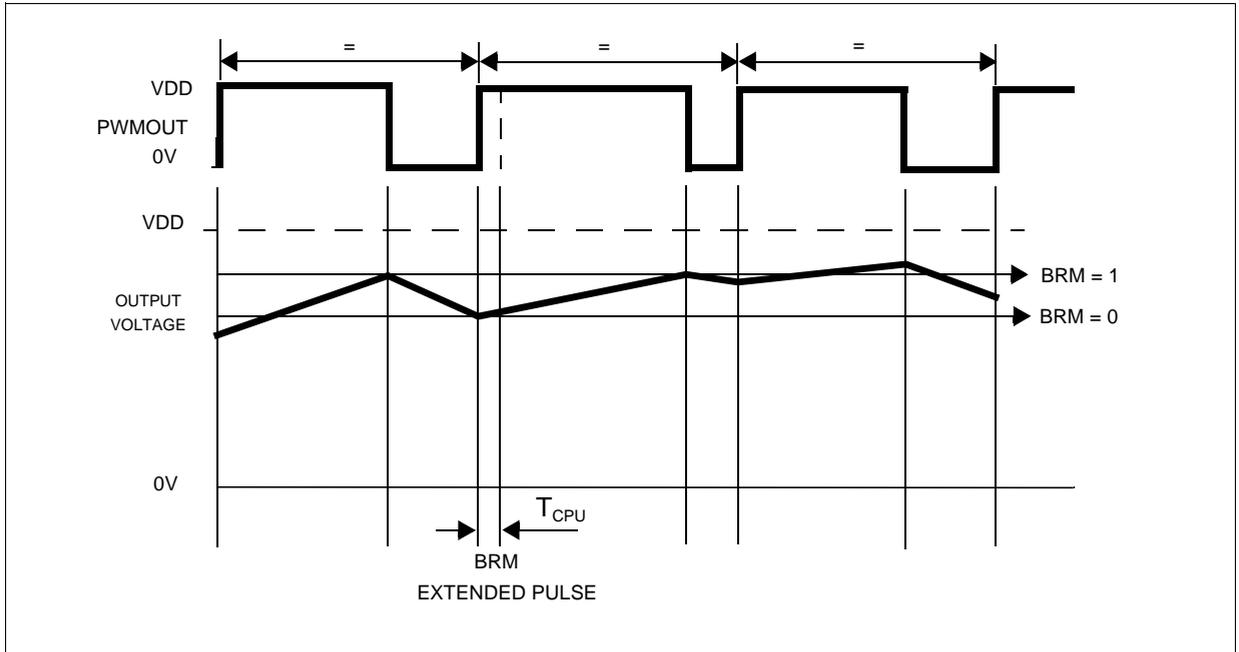
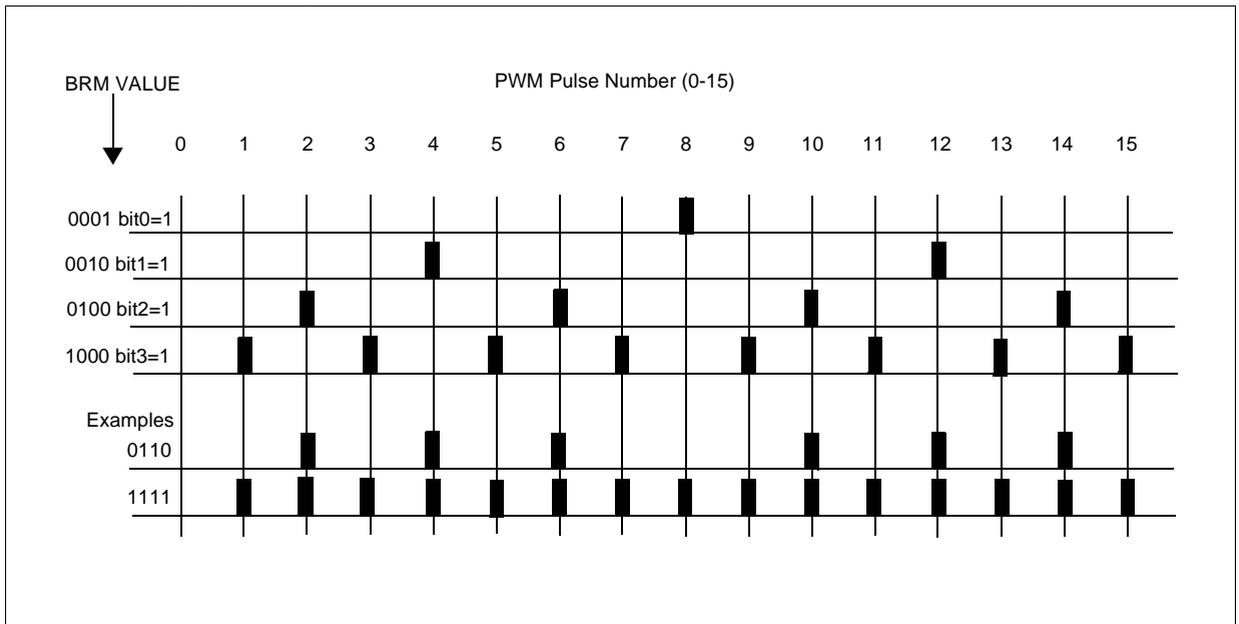
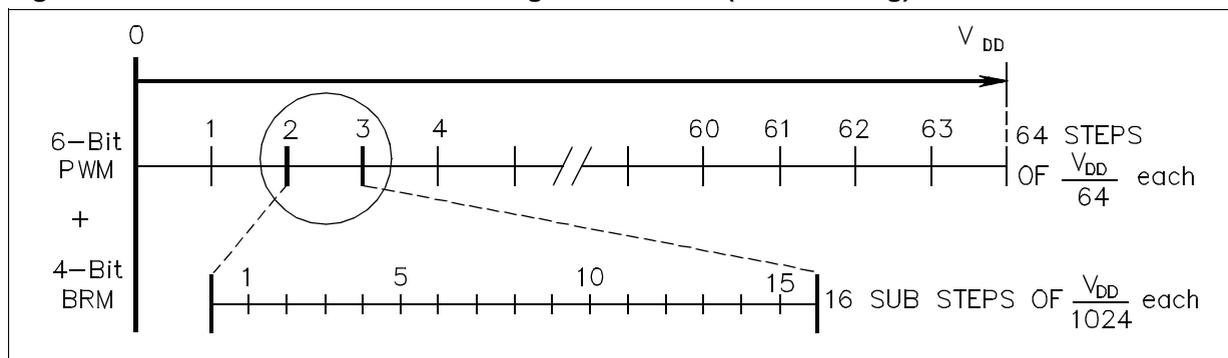


Figure 35. Graphical Representation of 4-Bit BRM Added Pulse Positions



PWM/BRM GENERATOR (Cont'd)

Figure 36. Precision for PWM/BRM Tuning for VOUTEFF (After filtering)



4.5.4 Register Description

On a channel basis, the 10 bits are separated into two data registers:

Note: The number of PWM and BRM channels available depends on the device. Refer to the device pin description and register map.

PULSE BINARY WEIGHT REGISTERS (PWM_i)

Read / Write

Reset Value 1000 0000 (80h)

7							0
1	POL	P5	P4	P3	P2	P1	P0

Bit 7 = Reserved (Forced by hardware to "1")

Bit 6 = **POL** Polarity Bit for channel *i*.

0: The channel *i* outputs a "1" level during the binary pulse and a "0" level after.

1: The channel *i* outputs a "0" level during the binary pulse and a "1" level after.

Bit 5:0 = **P[5:0]** PWM Pulse Binary Weight for channel *i*.

This register contains the binary value of the pulse.

For example :

1	POL	P	P	P	P	P	P	+	B	B	B	B
---	-----	---	---	---	---	---	---	---	---	---	---	---

Effective (with external RC filtering) DAC value

1	POL	P	P	P	P	P	P	B	B	B	B
---	-----	---	---	---	---	---	---	---	---	---	---

BRM REGISTERS

Read / Write

Reset Value: 0000 0000 (00h)

7							0
B7	B6	B5	B4	B3	B2	B1	B0

These registers define the intervals where an incremental pulse is added to the beginning of the original PWM pulse. Two BRM channel values share the same register.

Bit 7:4 = **B[7:4]** BRM Bits (channel *i*+1).

Bit 3:0 = **B[3:0]** BRM Bits (channel *i*)

Note: From the programmer's point of view, the PWM and BRM registers can be regarded as being combined to give one data value.

Table 17. PWM (DAC) Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
24	PWM1		POL	P5 ..P0					
25	BRM21	BRM Channel 2				BRM Channel 1			
26	PWM2		POL	P5 ..P0					
27	PWM3		POL	P5 ..P0					
28	BRM43	BRM Channel 4				BRM Channel 3			
29	PWM4		POL	P5 ..P0					

4.6 8-BIT A/D CONVERTER (ADC)

4.6.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 8 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 8 different sources.

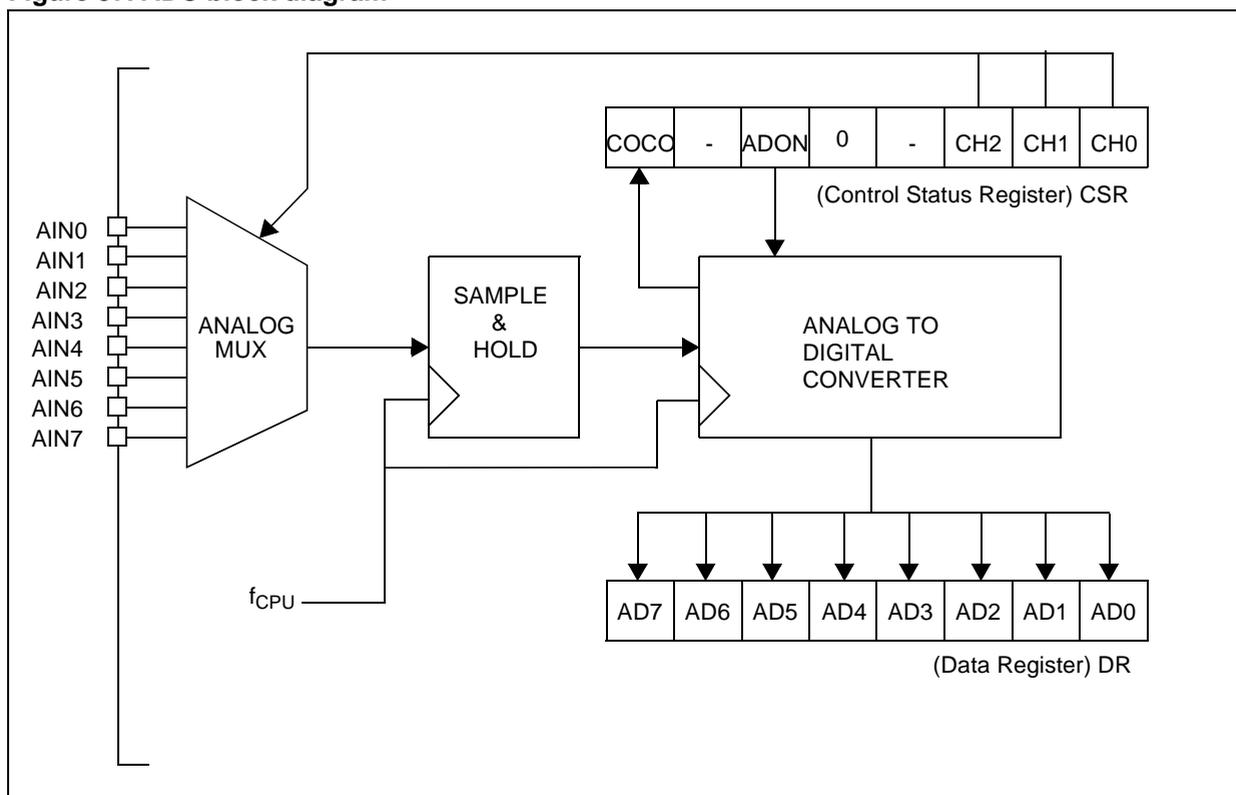
The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

4.6.2 Main Features

- 8-bit conversion
- Up to 8 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in [Figure 37](#).

Figure 37. ADC block diagram



8-BIT A/D CONVERTER (ADC) (Cont'd)

4.6.3 Functional Description

The high level reference voltage V_{DDA} must be connected externally to the V_{DD} pin. The low level reference voltage V_{SSA} must be connected externally to the V_{SS} pin. In some devices (refer to device pin out description) high and low level reference voltages are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be degraded by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

Characteristics:

The conversion is monotonic meaning the result never decreases if the analog input does not and never increases if the analog input does not.

If input voltage is greater than or equal to V_{DD} (voltage reference high) then results = FFh (full scale) without overflow indication.

If input voltage $\leq V_{SS}$ (voltage reference low) then the results = 00h.

The conversion time is 64 CPU clock cycles including a sampling time of 31.5 CPU clock cycles.

The A/D converter is linear and the digital result of the conversion is given by the formula:

$$\text{Digital result} = \frac{255 \times \text{Input Voltage}}{\text{Reference Voltage}}$$

Where Reference Voltage is $V_{DD} - V_{SS}$.

The accuracy of the conversion is described in the Electrical Characteristics Section.

Procedure:

Refer to the CSR and SR registers [Section 4.6.4](#) for the bit definitions.

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the CSR register:

- Select the CH2 to CH0 bits to assign the analog channel to convert. Refer to [Table 18](#).
- Set the ADON bit. Then the A/D converter is enabled after a stabilization time (typically 30 μ s). It then performs a continuous conversion of the selected channel.

When a conversion is complete

- The COCO bit is set by hardware.
- No interrupt is generated.
- The result is in the DR register.

A write to the CSR register aborts the current conversion, resets the COCO bit and starts a new conversion.

Notes: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed.

The A/D converter is not affected by WAIT mode.

When the MCU enters HALT mode with the A/D converter enabled, the converter is disabled until the HALT mode is exited and the start-up delay has elapsed. A stabilisation time is also required before accurate conversions can be performed.

8-BIT A/D CONVERTER (ADC) (Cont'd)**4.6.4 Register Description****CONTROL/STATUS REGISTER (CSR)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
COCO	-	ADON	0	-	CH2	CH1	CH0

Bit 7 = COCO Conversion Complete.

This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.

0: Conversion is not complete.

1: Conversion can be read from the DR register.

Bit 6 = Reserved. Must always be cleared.

Bit 5 = ADON A/D converter On.

This bit is set and cleared by software.

0: A/D converter is switched off.

1: A/D converter is switched on.

Note: a typically 30µs delay time is necessary for the ADC to stabilize when the ADON bit is set.

Bit 4 = Reserved. Forced by hardware to 0.

Bit 3 = Reserved. Must always be cleared.

Bits 2-0: CH2-CH0 Channel Selection.

These bits are set and cleared by software. They select the analog input to convert.

Pin*	CH2	CH1	CH0
AIN0	0	0	0
AIN1	0	0	1
AIN2	0	1	0
AIN3	0	1	1
AIN4	1	0	0
AIN5	1	0	1
AIN6	1	1	0
AIN7	1	1	1

***IMPORTANT NOTE:** The number of pins AND the channel selection varies according to the device. REFER TO THE DEVICE PINOUT).

DATA REGISTER (DR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Bit 7:0 = AD7-AD0 Analog Converted Value.

This register contains the converted analog value in the range 00h to FFh.

Reading this register reset the COCO flag.

Table 19. ADC Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0B	CSR	COCO	-	ADON	0	-	CH2	CH1	CH0
0A	DR	AD7 .. AD0							

5 INSTRUCTION SET

5.1 ST7 ADDRESSING MODES

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#\$55
Direct	ld A,\$55
Indexed	ld A,(\$55,X)
Indirect	ld A,([\$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do

Table 20. ST7 Addressing Mode Overview

Mode		Syntax	Destination/ Source	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)	
Inherent		nop				+ 0	
Immediate		ld A,#\$55				+ 1	
Short	Direct	ld A,\$10	00..FF			+ 1	
Long	Direct	ld A,\$1000	0000..FFFF			+ 2	
No Offset	Direct	Indexed	ld A,(X)	00..FF		+ 0 (with X register) + 1 (with Y register)	
Short	Direct	Indexed	ld A,(\$10,X)	00..1FE		+ 1	
Long	Direct	Indexed	ld A,(\$1000,X)	0000..FFFF		+ 2	
Short	Indirect		ld A,[\$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[\$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([\$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([\$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC-128/PC+127 ¹⁾		+ 1	
Relative	Indirect		jrne [\$10]	PC-128/PC+127 ¹⁾	00..FF	byte	+ 2
Bit	Direct		bset \$10,#7	00..FF		+ 1	
Bit	Indirect		bset [\$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt \$10,#7,skip	00..FF		+ 2	
Bit	Indirect	Relative	btjt [\$10],#7,skip	00..FF	00..FF	byte	+ 3

Note 1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

so, most of the addressing modes may be subdivided in two sub-modes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

ST7 ADDRESSING MODES (Cont'd)

5.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

5.1.2 Immediate

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the the operand value. .

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

5.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two sub-modes:

Direct (short)

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

5.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three sub-modes:

Indexed (No Offset)

There is no offset, (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

5.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

ST7 ADDRESSING MODES (Cont'd)**5.1.6 Indirect Indexed (Short, Long)**

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 21. Instructions Supporting Direct, Indexed, Indirect and Indirect Indexed Addressing Modes

Long and Short Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

5.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/Indirect Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two sub-modes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

The offset is defined in memory, which address follows the opcode.

5.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may

be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte pockets are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction
 PC-1 Prebyte
 PC opcode
 PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.

It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	$A = A + M + C$	A	M	H		N	Z	C
ADD	Addition	$A = A + M$	A	M	H		N	Z	C
AND	Logical And	$A = A \cdot M$	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	$A = FFH - A$	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg \$10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg pop CC	reg CC	M M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz b 1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

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

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

Power Considerations. The average chip-junction temperature, 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).

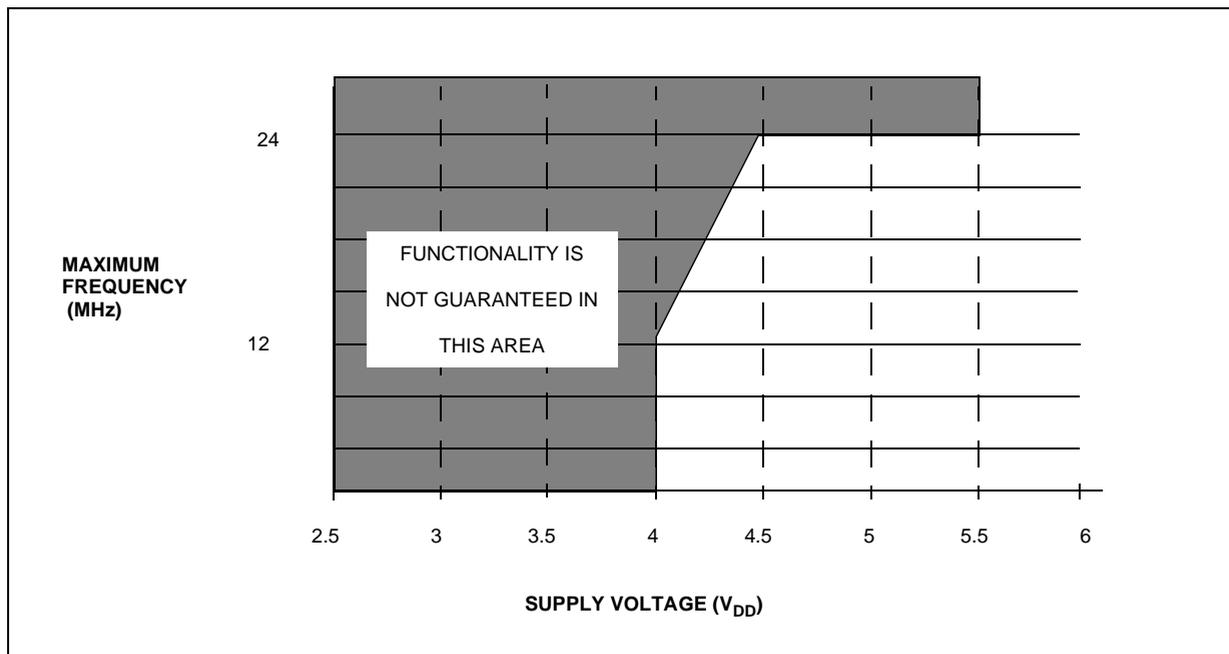
P_{PORT} = Port power dissipation (determined by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 6.0	V
V_I	Input Voltage	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
V_{AI}	Analog Input Voltage (A/D Converter)	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
V_O	Output Voltage	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
$I_{V_{DD}}$	Total Current into V_{DD} (source)	TBD	mA
$I_{V_{SS}}$	Total Current out of V_{SS} (sink)	TBD	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.

6.2 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	$f_{CPU} = 8 \text{ MHz}$ $f_{CPU} = 4 \text{ MHz}$	4.5 4.0		5.5 5.5	V
f_{OSC}	Oscillator Frequency	$V_{DD} = 4.0\text{V}$ $V_{DD} = 4.5\text{V}$	0 0		12 24	MHz

Figure 38. Maximum Operating Frequency (Fmax) Versus Supply Voltage (V_{DD})

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

6.3 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 Input pins				$V_{DD} \times 0.3$	V
V_{IH}	Input High Level Voltage All Input pins		$V_{DD} \times 0.7$			V
V_{HYS}	Hysteresis Voltage ¹⁾ All Input pins	$V_{DD} = 5V$	TBD			V
V_{OL}	Low Level Output Voltage All Output pins	$V_{DD} = 5.0V; I_{OL} = +10\mu A$ $V_{DD} = 5.0V; I_{OL} = +1.6mA$			0.1 0.4	V
	Low Level Output Voltage High Sink I/O pins	$V_{DD} = 5.0V; I_{OL} = +10\mu A$ $V_{DD} = 5.0V; I_{OL} = +1.6mA$ $V_{DD} = 5.0V; I_{OL} = +10mA$			0.1 0.4 1.5	
V_{OH}	High Level Output Voltage All Output pins	$V_{DD} = 5.0V; I_{OL} = -10\mu A$ $V_{DD} = 5.0V; I_{OL} = 1.6mA$	4.9 4			V
I_{IL} I_{IH}	Input Leakage Current All Input pins but RESET	$V_{IN} = V_{SS}$ (No Pull-Up configured) $V_{IN} = V_{DD}$		0.1	10	μA
	Input Leakage Current RESET pin	$V_{IN} = V_{SS}$ $V_{IN} = V_{DD}$		-50	10	
I_{DD}	Supply Current in RUN Mode ²⁾	$V_{DD} = 5.0V$ $f_{OSC} = 12\text{ MHz}, f_{CPU} = 4\text{ MHz}$ $f_{OSC} = 24\text{ MHz}, f_{CPU} = 8\text{ MHz}$		TBD 14	TBD 18	mA
	Supply Current in SLOW Mode ³⁾	$V_{DD} = 5.0V$ $f_{OSC} = 12\text{ MHz}, f_{CPU} = 2\text{ MHz}$ $f_{OSC} = 24\text{ MHz}, f_{CPU} = 4\text{ MHz}$			TBD	mA
	Supply Current in WAIT Mode ³⁾	$V_{DD} = 5.0V$ $f_{OSC} = 12\text{ MHz}, f_{CPU} = 4\text{ MHz}$ $f_{OSC} = 24\text{ MHz}, f_{CPU} = 8\text{ MHz}$		TBD 12	TBD 18	mA
	Supply Current in HALT Mode	$I_{LOAD} = 0mA$ $V_{DD} = 5.0V$		250	500	μA

Notes:

1. Hysteresis voltage between switching levels
2. CPU running with memory access.
3. All peripherals in stand-by

6.4 A/D CONVERTER CHARACTERISTICS

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

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution			8		Bit
DLE ILE	Differential linearity error Integral linearity error	$f_{OSC} = 24\text{ MHz}$		± 0.3	± 0.5 ± 1	LSB
t_C	Conversion Time	$f_{CPU} = 8\text{ MHz}$		8		μs

Note: Noise at $AV_{DD}, AV_{SS} < 10mV$

6.5 PWM (DAC) CHARACTERISTICS

PWM/BRM Electrical and Timings						
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
F	Repetition rate	$T_{CPU} = 125\text{ns}$		125		kHz
Res	Resolution	$T_{CPU} = 125\text{ns}$		125		ns
S	Output step	$V_{DD} = 5\text{V}, 10\text{ bits}$		5		mV
		$V_{DD} = 5\text{V}, 12\text{ bits}$		1.25		mV
R_S	Serial resistor	-		700	1000	Ohms

6.6 I2C CHARACTERISTICS

I2C Electrical specifications						
Parameter	Symbol	Unit	Standard mode I2C		Fast mode I2C	
			Min	Max	Min	Max
Hysteresis of Schmitt trigger inputs	V_{HYS}	V	N/A	N/A	0.2	
Fixed input levels			N/A	N/A	$0.05 V_{DD}$	
V_{DD} -related input levels			N/A	N/A		
Pulse width of spikes which must be suppressed by the input filter	T_{SP}	ns	N/A	N/A	0 ns	50 ns
Output fall time from V_{IH} min to V_{IL} max with a bus capacitance from 10 pF to 400 pF with up to 3 mA sink current at VOL1 with up to 6 mA sink current at VOL2	T_{OF}	ns	N/A	250	$20+0.1C_b$	250
			N/A	N/A	$20+0.1C_b$	250
Input current each I/O pin with an input voltage between 0.4V and 0.9 V_{DD} max	I	μA	- 10	10	-10	10
Capacitance for each I/O pin	C	pF		10		10

N/A = Not Applicable

 C_b = Capacitance of one bus in pF

I2C Bus Timings

Parameter	Standard I2C		Fast I2C		Symbol	Unit
	Min	Max	Min	Max		
Bus free time between a STOP and START condition	4.7		1.3		T_{BUF}	ms
Hold time START condition. After this period, the first clock pulse is generated	4.0		0.6		$T_{HD:STA}$	μs
LOW period of the SCL clock	4.7		1.3		T_{LOW}	μs
HIGH period of the SCL clock	4.0		0.6		T_{HIGH}	μs
Set-up time for a repeated START condition	4.7		0.6		$T_{SU:STA}$	μs
Data hold time	0 ¹⁾		0 ¹⁾	0.9 ²⁾	$T_{HD:DAT}$	ns
Data set-up time	250		100		$T_{SU:DAT}$	ns
Rise time of both SDA and SCL signals		1000	$20+0.1C_b$	300	TR	ns
Fall time of both SDA and SCL signals		300	$20+0.1C_b$	300	TF	ns

I2C Bus Timings						
Parameter	Standard I2C		Fast I2C		Symbol	Unit
	Min	Max	Min	Max		
Set-up time for STOP condition	4.0		0.6		$T_{SU:STO}$	ns
Capacitive load for each bus line		400		400	Cb	pF

Notes:

1. The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.
2. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.

7 GENERAL INFORMATION

7.1 EPROM ERASURE

EPROM version devices are erased by exposure to high intensity UV light admitted through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current.

It is recommended that the EPROM devices be kept out of direct sunlight, since the UV content of sunlight can be sufficient to cause functional failure. Extended exposure to room level fluorescent lighting may also cause erasure.

An opaque coating (paint, tape, label, etc...) should be placed over the package window if the product is to be operated under these lighting conditions. Covering the window also reduces I_{DD} in power-saving modes due to photo-diode leakage currents.

An Ultraviolet source of wave length 2537 Å yielding a total integrated dosage of 15 Watt-sec/cm² is required to erase the device. It will be erased in 15 to 20 minutes if such a UV lamp with a 12mW/cm² power rating is placed 1 inch from the device window without any interposed filters.

7.2 PACKAGE MECHANICAL DATA

Figure 39. 34-Pin Plastic Small Outline Package

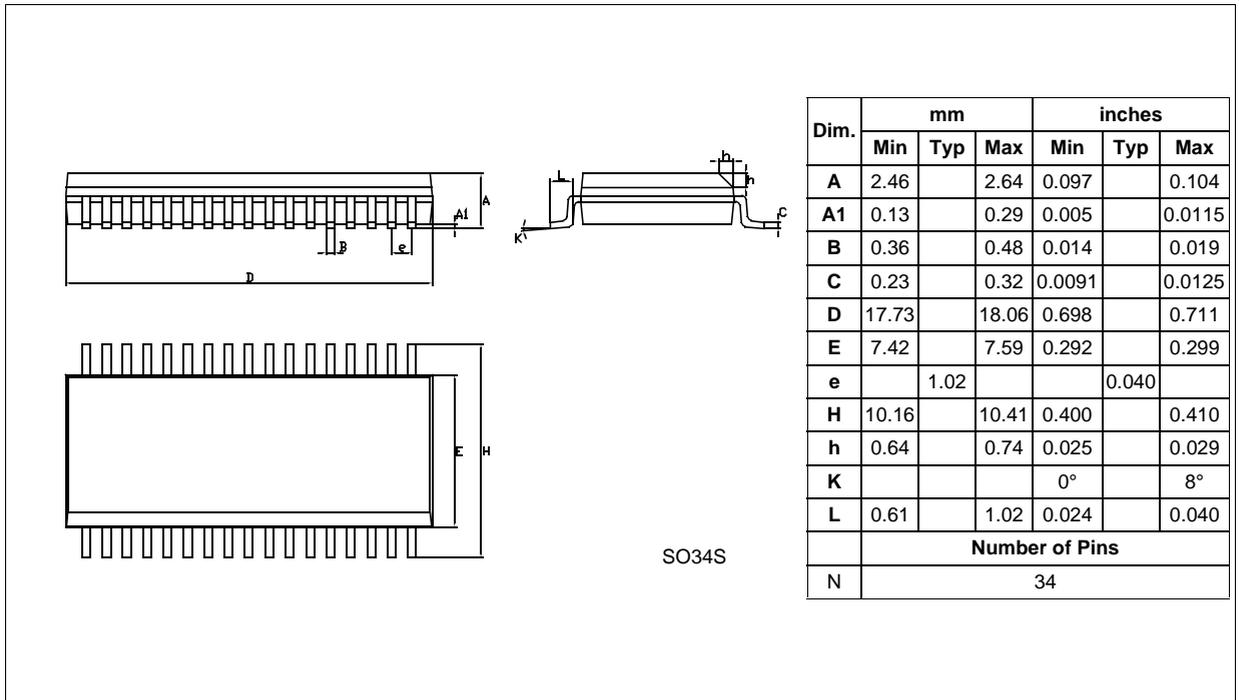


Figure 40. 32-Pin Shrink Plastic Dual In Line Package

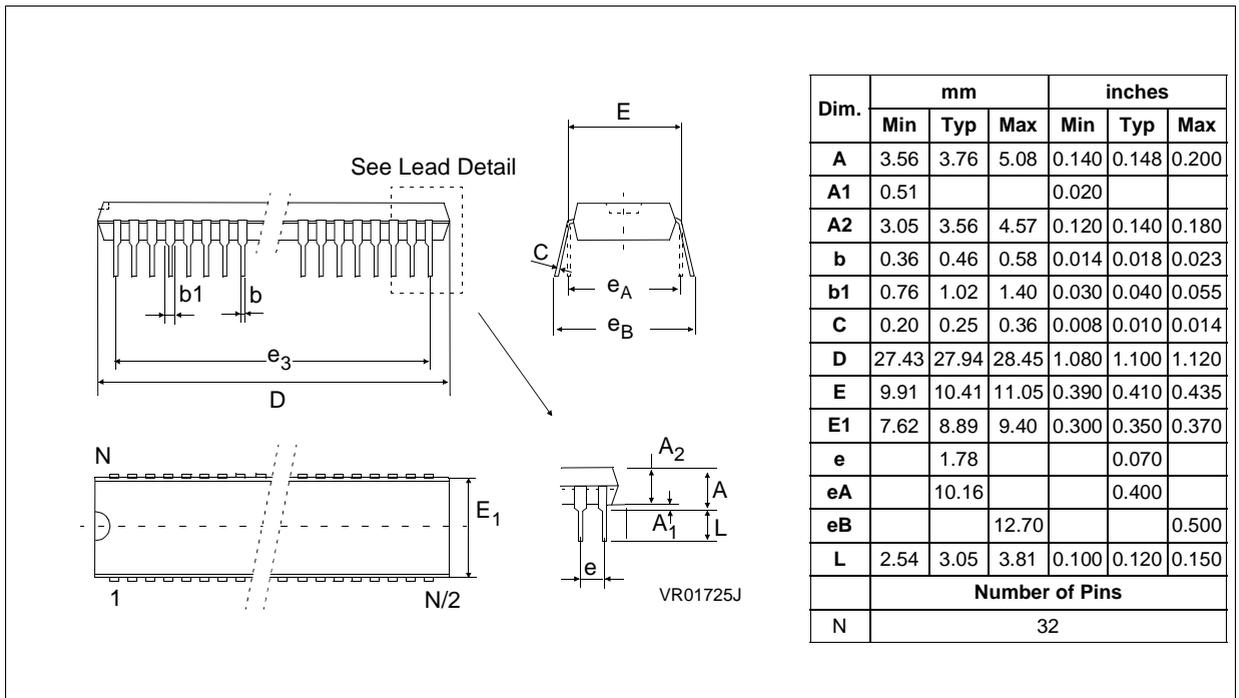
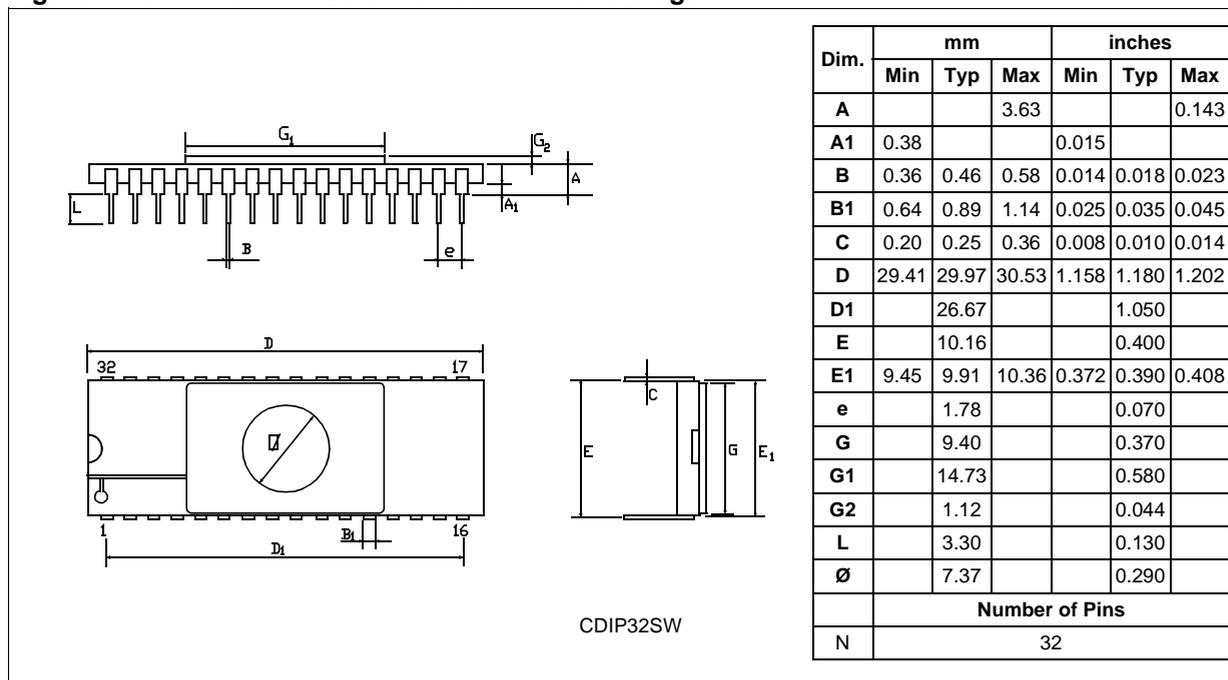


Figure 41. 32-Pin Shrink Ceramic Dual In-Line Package



7.3 ORDERING INFORMATION

Each device is available for production in user programmable version (OTP) as well as in factory coded version (ROM). OTP devices are shipped to customer with a default blank content FFh, while ROM factory coded parts contain the code sent by customer. There is one common EPROM version for debugging and prototyping which features the maximum memory size and peripherals of the sub-family. Care must be taken to only use resources available on the target device.

Contact sales office for further ordering information and availability.

7.3.1 Transfer Of Customer Code

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

The selected options are communicated to ST using the correctly completed OPTION LIST appended.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Figure 42. ROM Factory Coded Device Types

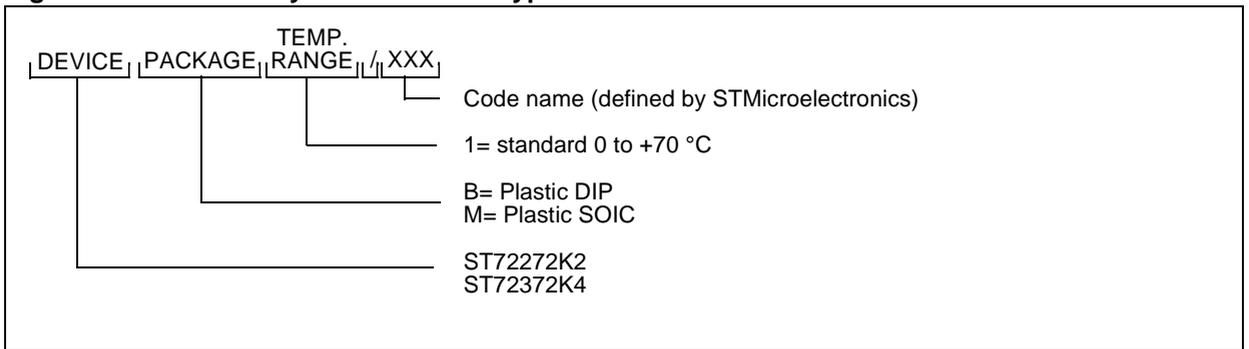
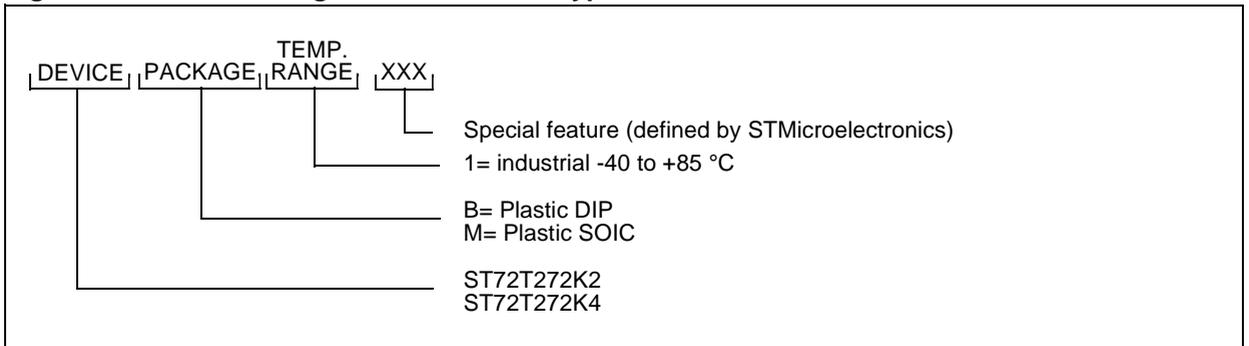


Figure 43. OTP User Programmable Device Types



Note: The ST72E272K4D0 (32-pin ceramic SDIP) is used as the EPROM version for the above devices. The EPROM devices are tested for operation at 25 °C only.

Notes

Information furnished is believed to be accurate and reliable. However, STMicroelectronics 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 STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

©1998 STMicroelectronics - All Rights Reserved.

Purchase of I²C Components by STMicroelectronics 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.

STMicroelectronics Group of Companies

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