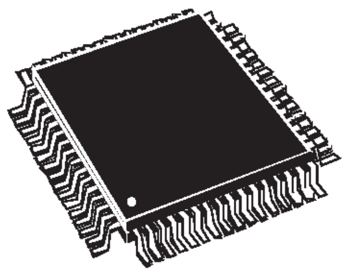
ST72311R, ST72511R, ST72512R, ST72532R

8-BIT MCU WITH NESTED INTERRUPTS, EEPROM, ADC, 16-BIT TIMERS, 8-BIT PWM ART, SPI, SCI, CAN INTERFACES

PRELIMINARY DATA

- 16K to 60K Program memory (ROM/OTP/EPROM) with read-out protection
- EEPROM Data memory (only on ST72532R4)
- Master Reset and Power-on Reset
- Low voltage supply supervisor
- Low consumption resonator oscillator and by-pass for external clock source
- 4 Power saving modes
- Nested interrupt controller
- NMI dedicated non maskable interrupt pin
- 48 multifunctional bidirectional I/O lines with:
  - External interrupt capability (4 vectors)
  - 32 alternate function lines
  - 12 high sink outputs
- Real time base, Beep and Clock-out capabilities
- Configurable watchdog reset
- Two 16-bit timers with:
  - 2 input captures
  - 2 output compares
  - External clock input on one timer
  - PWM and Pulse generator modes
- 8-bit PWM Auto-reload timer (except on ST72512R4, ST72532R4) with:
  - 4 independent output channels
  - Output compare and time base interrupt
  - External clock with event detector
- SPI synchronous serial interface
- SCI asynchronous serial interface
- CAN interface (except on ST72311Rx)
- 8-bit ADC with 8 input pins



TQFP64
14 x 14

- 8-bit data manipulation
- 63 basic instructions
- 17 main addressing modes
- 8 x 8 unsigned multiply instruction
- True bit manipulation

- Full hardware/software development package

Device Summary

Features	ST72511R9	ST72511R7	ST72511R6	ST72311R9	ST72311R7	ST72311R6	ST72512R4	ST72532R4
Program memory - bytes	60K	48K	32K	60K	48K	32K	16K	16K
RAM (stack) - bytes	2048 (256)	1536 (256)	1024 (256)	2048 (256)	1536 (256)	1024 (256)	1024 (256)	1024 (256)
EEPROM - bytes	-	-	-	-	-	-	-	256
Peripherals	Watchdog, 16-bit Timers, 8-bit PWM ART, SPI, SCI, CAN, ADC			Watchdog, 16-bit Timers, 8-bit PWM ART, SPI, SCI, ADC			Watchdog, 16-bit Timers, SPI, SCI, CAN, ADC	
Operating Supply	3.0V to 5.5V							3.0 to 5.5V [1)]
CPU Frequency	2 to 8 MHz (with 4 to 16 MHz oscillator)							2 to 4 MHz [1)]
Operating Temperature	-40°C to +85°C (-40°C to +105/125°C optional)							
Packages	TQFP64							

Note 1. See Section 7.3.1 on page 130 for more information on V_{DD} versus f_{OSC}.

Rev. 1.5

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST72311R, ST72511R, ST72512R and ST72532R devices are members of the ST7 microcontroller family. They can be grouped as follows:

– ST725xxR devices are designed for mid-range applications with a CAN bus interface (Controller Area Network)

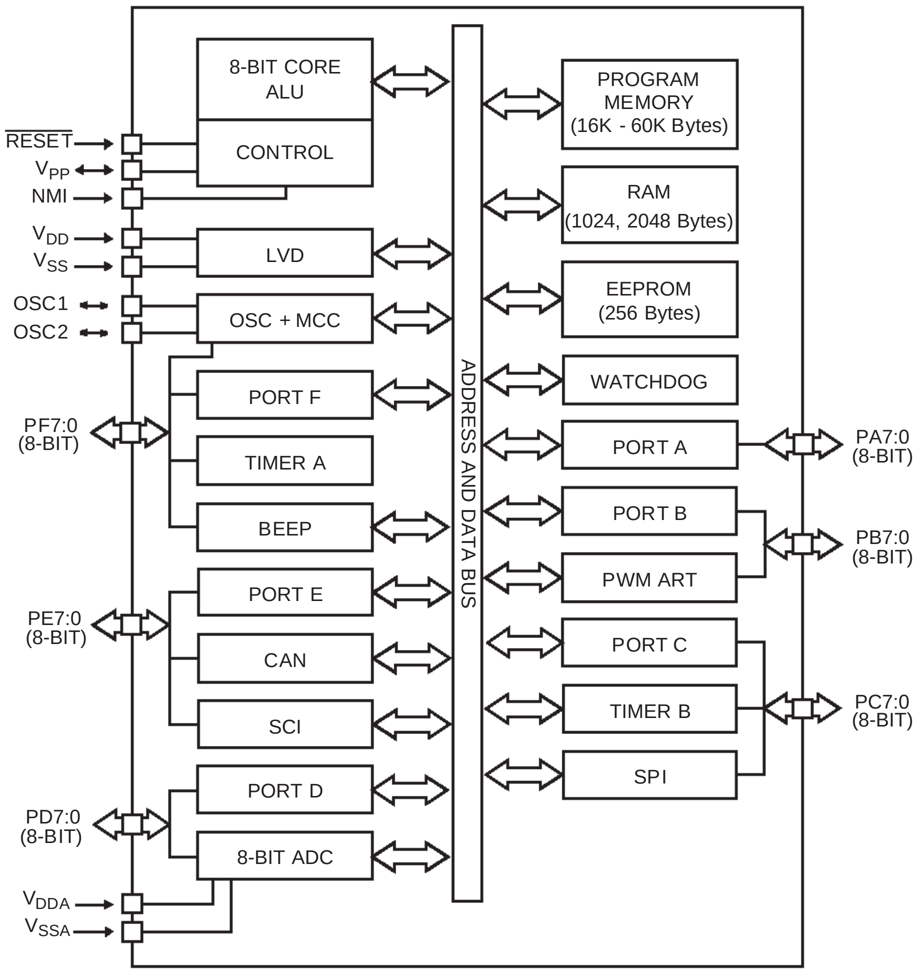
– ST72311R devices target the same range of applications but without CAN interface.

All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

Under software control, all devices can be placed in WAIT, SLOW, ACTIVE-HALT or HALT mode, reducing power consumption when the application is in idle or standby state.

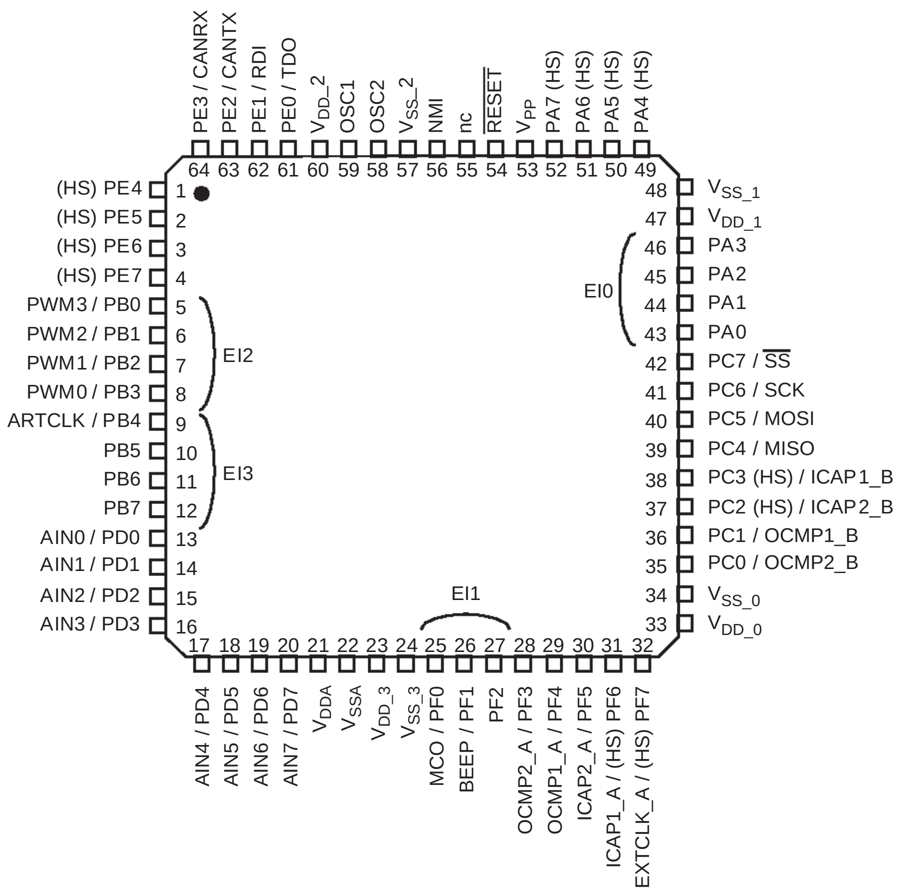
The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

Figure 1. Device Block Diagram



1.2 PIN DESCRIPTION

Figure 2. 64-Pin TQFP Package Pinout



PIN DESCRIPTION (Cont'd)

Legend / Abbreviations:

Type:	I = input, O = output, S = supply
Input level:	A = Dedicated analog input
In/Output level:	C = CMOS 0.3V_{DD}/0.7V_{DD}, C_T= CMOS 0.3V_{DD}/0.7V_{DD} with input trigger
Output level:	HS = high sink (on N-buffer only),

Port configuration capabilities:

– Input:	float = floating, wpu = weak pull-up, int = interrupt, ana = analog
– Output:	OD = open drain, T = true open drain, PP = push-pull

Note: the Reset configuration of each pin is shown in bold.

Table 1. Device Pin Description

Pin n°	Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
					Input				Output			
TQFP64			Input	Output	float	wpu	int	ana	OD	PP		
1	PE4 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E4	
2	PE5 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E5	
3	PE6 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E6	
4	PE7 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E7	
5	PB0/PWM3	I/O	C_T		**X**	EI2			X	X	Port B0	PWM Output 3
6	PB1/PWM2	I/O	C_T		**X**	EI2			X	X	Port B1	PWM Output 2
7	PB2/PWM1	I/O	C_T		**X**	EI2			X	X	Port B2	PWM Output 1
8	PB3/PWM0	I/O	C_T		**X**		EI2		X	X	Port B3	PWM Output 0
9	PB4/ARTCLK	I/O	C_T		**X**		EI3		X	X	Port B4	PWM-ART External Clock
10	PB5	I/O	C_T		**X**	EI3			X	X	Port B5	
11	PB6	I/O	C_T		**X**	EI3			X	X	Port B6	
12	PB7	I/O	C_T		**X**	EI3			X	X	Port B7	
13	PD0/AIN0	I/O	C_T		**X**	X		X	X	X	Port D0	ADC Analog Input 0
14	PD1/AIN1	I/O	C_T		**X**	X		X	X	X	Port D1	ADC Analog Input 1
15	PD2/AIN2	I/O	C_T		**X**	X		X	X	X	Port D2	ADC Analog Input 2
16	PD3/AIN3	I/O	C_T		**X**	X		X	X	X	Port D3	ADC Analog Input 3
17	PD4/AIN4	I/O	C_T		**X**	X		X	X	X	Port D4	ADC Analog Input 4
18	PD5/AIN5	I/O	C_T		**X**	X		X	X	X	Port D5	ADC Analog Input 5
19	PD6/AIN6	I/O	C_T		**X**	X		X	X	X	Port D6	ADC Analog Input 6
20	PD7/AIN7	I/O	C_T		**X**	X		X	X	X	Port D7	ADC Analog Input 7
21	V_{DDA}	S									Analog Power Supply Voltage	
22	V_{SSA}	S									Analog Ground Voltage	
23	V_{DD_3}	S									Digital Main Supply Voltage	
24	V_{SS_3}	S									Digital Ground Voltage	
25	PF0/MCO	I/O	C_T		**X**	EI1			X	X	Port F0	Main clock output (f_{OSC}/2)
26	PF1/BEEP	I/O	C_T		**X**	EI1			X	X	Port F1	Beep signal output
27	PF2	I/O	C_T		**X**		EI1		X	X	Port F2	

Pin n°	Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
TQFP64			Input	Output	Input				Output			
					float	wpu	int	ana	OD	PP		
28	PF3/OCMP2_A	I/O	C_T		**X**	X			X	X	Port F3	Timer A Output Compare 2
29	PF4/OCMP1_A	I/O	C_T		**X**	X			X	X	Port F4	Timer A Output Compare 1
30	PF5/ICAP2_A	I/O	C_T		**X**	X			X	X	Port F5	Timer A Input Capture 2
31	PF6 (HS)/ICAP1_A	I/O	C_T	HS	**X**	X			X	X	Port F6	Timer A Input Capture 1
32	PF7 (HS)/EXTCLK_A	I/O	C_T	HS	**X**	X			X	X	Port F7	Timer A External Clock Source
33	V_{DD_0}	S									Digital Main Supply Voltage	
34	V_{SS_0}	S									Digital Ground Voltage	
35	PC0/OCMP2_B	I/O	C_T		**X**	X			X	X	Port C0	Timer B Output Compare 2
36	PC1/OCMP1_B	I/O	C_T		**X**	X			X	X	Port C1	Timer B Output Compare 1
37	PC2 (HS)/ICAP2_B	I/O	C_T	HS	**X**	X			X	X	Port C2	Timer B Input Capture 2
38	PC3 (HS)/ICAP1_B	I/O	C_T	HS	**X**	X			X	X	Port C3	Timer B Input Capture 1
39	PC4/MISO	I/O	C_T		**X**	X			X	X	Port C4	SPI Master In / Slave Out Data
40	PC5/MOSI	I/O	C_T		**X**	X			X	X	Port C5	SPI Master Out / Slave In Data
41	PC6/SCK	I/O	C_T		**X**	X			X	X	Port C6	SPI Serial Clock
42	PC7/$\overline{SS}$	I/O	C_T		**X**	X			X	X	Port C7	SPI Slave Select (active low)
43	PA0	I/O	C_T		**X**	EI0			X	X	Port A0	
44	PA1	I/O	C_T		**X**	EI0			X	X	Port A1	
45	PA2	I/O	C_T		**X**	EI0			X	X	Port A2	
46	PA3	I/O	C_T		**X**		EI0		X	X	Port A3	
47	V_{DD_1}	S									Digital Main Supply Voltage	
48	V_{SS_1}	S									Digital Ground Voltage	
49	PA4 (HS)	I/O	C_T	HS	**X**	X			X	X	Port A4	
50	PA5 (HS)	I/O	C_T	HS	**X**	X			X	X	Port A5	
51	PA6 (HS)	I/O	C_T	HS	**X**				T		Port A6	
52	PA7 (HS)	I/O	C_T	HS	**X**				T		Port A7	
53	V_{PP}	I									Must be tied low in user mode. In programming mode when available, this pin acts as the programming voltage input V_{PP}.	
54	$\overline{\text{RESET}}$	I/O	C			X				X	Top priority non maskable interrupt (active low)	
55	NC	Not Connected										
56	NMI	I	C_T		X						Non maskable interrupt input pin	
57	V_{SS_3}	S									Digital Ground Voltage	
58	OSC2	I/O									External clock mode input pull-up or crystal/ceramic resonator oscillator inverter output	
59	OSC1	I									External clock input or crystal/ceramic resonator oscillator inverter input	
60	V_{DD_3}	S									Digital Main Supply Voltage	
61	PE0/TDO	I/O	C_T		**X**	X			X	X	Port E0	SCI Transmit Data Out
62	PE1/RDI	I/O	C_T		**X**	X			X	X	Port E1	SCI Receive Data In
63	PE2/CANTX	I/O	C_T			**X**					Port E2	CAN Transmit Data Output
64	PE3/CANRX	I/O	C_T		**X**	X			X	X	Port E3	CAN Receive Data Input

1.3 REGISTER & MEMORY MAP

As shown in the Figure 3, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register location, up to 2Kbytes of RAM, up to 256 bytes of data EEPROM and up to 60Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

Figure 3. Memory Map

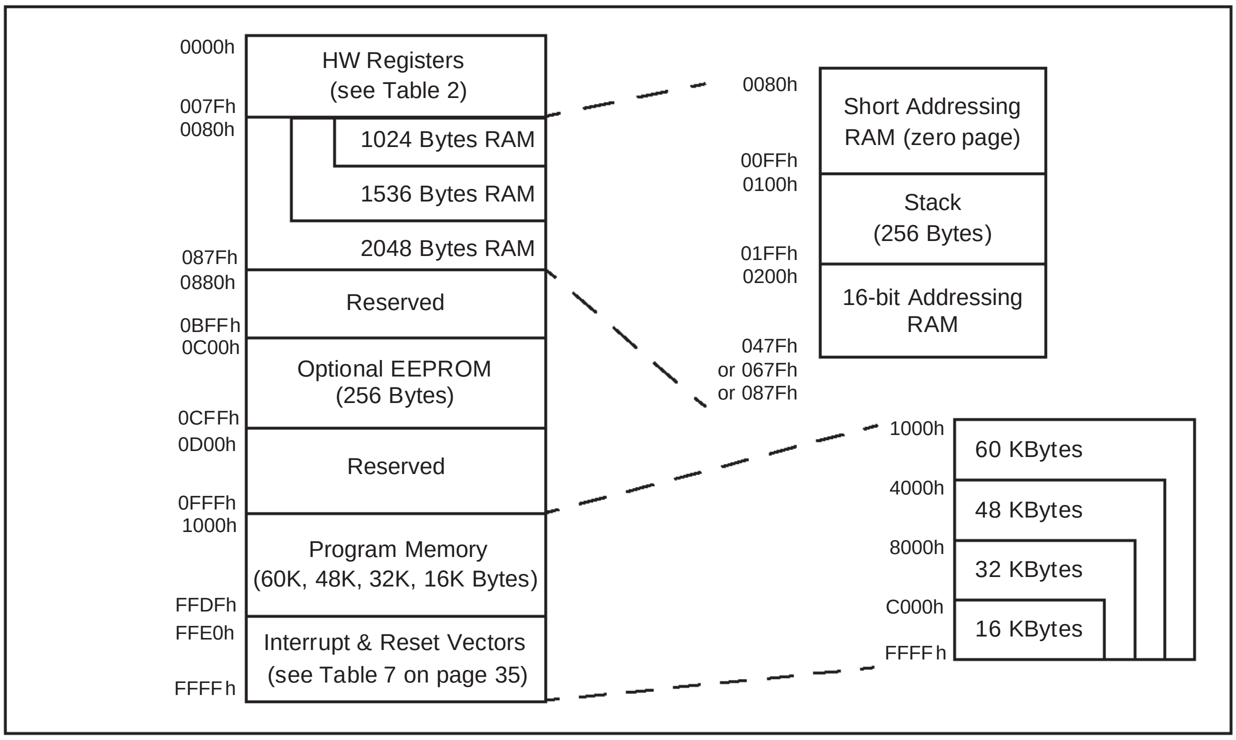


Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h 0001h 0002h	Port A	PADR PADDR PAOR	Port A Data Register Port A Data Direction Register Port A Option Register	00h [1)] 00h 00h	R/W R/W R/W [2)]
0003h	Reserved Area (1 Byte)				
0004h 0005h 0006h	Port C	PCDR PCDDR PCOR	Port C Data Register Port C Data Direction Register Port C Option Register	00h [1)] 00h 00h	R/W R/W R/W
0007h	Reserved Area (1 Byte)				
0008h 0009h 000Ah	Port B	PBDR PBDDR PBOR	Port B Data Register Port B Data Direction Register Port B Option Register	00h [1)] 00h 00h	R/W R/W R/W
000Bh	Reserved Area (1 Byte)				
000Ch 000Dh 000Eh	Port E	PEDR PEDDR PEOR	Port E Data Register Port E Data Direction Register Port E Option Register	00h [1)] 00h 00h	R/W R/W [2)] R/W [2)]
000Fh	Reserved Area (1 Byte)				
0010h 0011h 0012h	Port D	PDDR PDDDR PDOR	Port D Data Register Port D Data Direction Register Port D Option Register	00h [1)] 00h 00h	R/W R/W R/W
0013h	Reserved Area (1 Byte)				
0014h 0015h 0016h	Port F	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h [1)] 00h 00h	R/W R/W R/W
0017h to 001Fh	Reserved Area (9 Bytes)				
0020h		MISCR1	Miscellaneous Register 1	00h	R/W
0021h 0022h 0023h	SPI	SPIDR SPICR SPISR	SPI Data I/O Register SPI Control Register SPI Status Register	xxh 0xh 00h	R/W R/W Read Only
0024h 0025h 0026h 0027h	ITC	ISPR0 ISPR1 ISPR2 ISPR3	Interrupt Software Priority Register 0 Interrupt Software Priority Register 1 Interrupt Software Priority Register 2 Interrupt Software Priority Register 3	FFh FFh FFh FFh	R/W R/W R/W R/W
0028h	Reserved Area (1 Byte)				
0029h	MCC	MCCSR	Main Clock Control / Status Register	01h	R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
002Ah	WATCHDOG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh		WDGSR	Watchdog Status Register	000x 000x	R/W
002Ch	EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
002Dh to 0030h	Reserved Area (4 Bytes)				
0031h	TIMER A	TACR2	Timer A Control Register 2	00h	R/W
0032h		TACR1	Timer A Control Register 1	00h	R/W
0033h		TASR	Timer A Status Register	xxh	Read Only
0034h		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
0035h		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
0036h		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
0037h		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
0038h		TACHR	Timer A Counter High Register	FFh	Read Only
0039h		TACLR	Timer A Counter Low Register	FCh	Read Only
003Ah		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Timer A Alternate Counter Low Register	FCh	Read Only
003Ch		TAIC2HR	Timer A Input Capture 2 High Register	xxh	Read Only
003Dh		TAIC2LR	Timer A Input Capture 2 Low Register	xxh	Read Only
003Eh		TAOC2HR	Timer A Output Compare 2 High Register	80h	R/W
003Fh		TAOC2LR	Timer A Output Compare 2 Low Register	00h	R/W
0040h		MISCR2	Miscellaneous Register 2	00h	R/W
0041h	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBSR	Timer B Status Register	xxh	Read Only
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h		TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W
0050h	SCI	SCISR	SCI Status Register	C0h	Read Only
0051h		SCIDR	SCI Data Register	xxh	R/W
0052h		SCIBRR	SCI Baud Rate Register	00xx xxxx	R/W
0053h		SCICR1	SCI Control Register 1	xxh	R/W
0054h		SCICR2	SCI Control Register 2	00h	R/W
0055h		SCIERPR	SCI Extended Receive Prescaler Register	00h	R/W
0056h			Reserved area		
0057h		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
0058h 0059h	Reserved Area (2 Bytes)				
005Ah 005Bh 005Ch 005Dh 005Eh 005Fh 0060h to 006Fh	CAN	CANISR CANICR CANCSR CANBRPR CANBTR CANPSR	CAN Interrupt Status Register CAN Interrupt Control Register CAN Control / Status Register CAN Baud Rate Prescaler Register CAN Bit Timing Register CAN Page Selection Register First address to Last address of CAN page X	00h 00h 00h 00h 23h 00h	R/W R/W R/W R/W R/W R/W See CAN Description
0070h 0071h	ADC	ADCDR ADCCSR	Data Register Control/Status Register	xxh 00h	Read Only R/W
0072h 0073h 0074h 0075h 0076h 0077h 0078h 0079h	PWM ART	PWMDCR3 PWMDCR2 PWMDCR1 PWMDCR0 PWMCR ARTCSR ARTCAR ARTARR	PWM AR Timer Duty Cycle Register 3 PWM AR Timer Duty Cycle Register 2 PWM AR Timer Duty Cycle Register 1 PWM AR Timer Duty Cycle Register 0 PWM AR Timer Control Register Auto-Reload Timer Control/Status Register Auto-Reload Timer Counter Access Register Auto-Reload Timer Auto-Reload Register	00h 00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W R/W
007Ah to 007Fh	Reserved Area (6 Bytes)				

Legend: x=unknown, R/W=read/write

Notes:

1. The real value of I/O port data register is readable only in output configuration. As the reset configuration of the I/O port is usually input, the associated pin status is read instead of the real register content when the reading at the DR address.

2. The unused bits must always keep the reset value.

1.4 EPROM PROGRAM MEMORY

The program memory of the OTP and EPROM devices can be programmed with EPROM programming tools available from STMicroelectronics

EPROM Erasure

EPROM 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.

1.5 DATA EEPROM

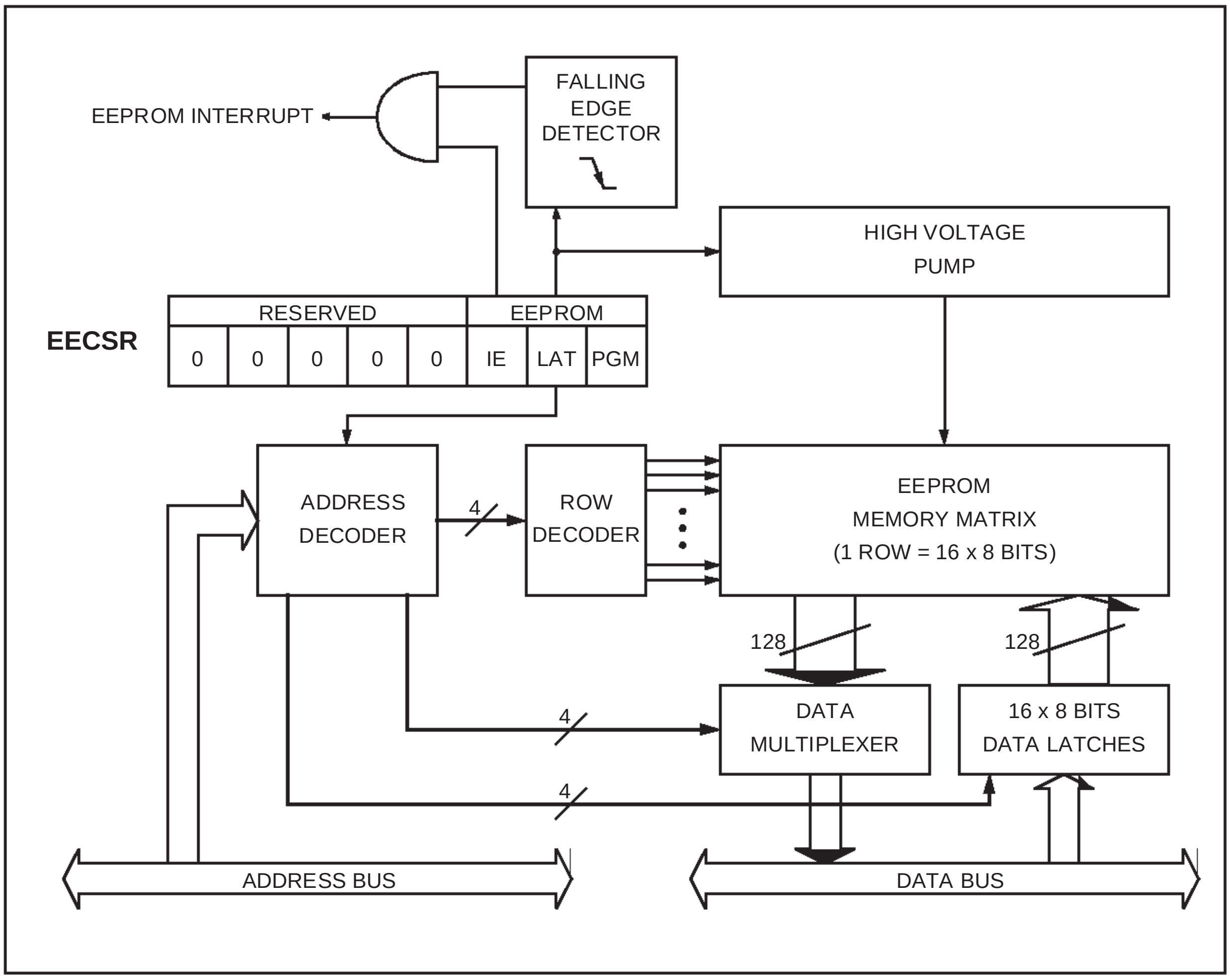
1.5.1 Introduction

The Electrically Erasable Programmable Read Only Memory can be used as a non volatile back-up for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

1.5.2 Main Features

- Up to 16 Bytes programmed in the same cycle
- EEPROM mono-voltage (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- End of programming cycle interrupt flag
- WAIT mode management

Figure 4. EEPROM Block Diagram



DATA EEPROM (Cont'd)

1.5.3 Memory Access

The Data EEPROM memory read/write access modes are controlled by the LAT bit of the EEPROM Control/Status register (EECSR). The flowchart in Figure 5 describes these different memory access modes.

Read Operation (LAT=0)

The EEPROM can be read as a normal ROM location when the LAT bit of the EECSR register is cleared. In a read cycle, the byte to be accessed is put on the data bus in less than 1 CPU clock cycle. This means that reading data from EEPROM takes the same time as reading data from EPROM, but this memory cannot be used to execute machine code.

Write Operation (LAT=1)

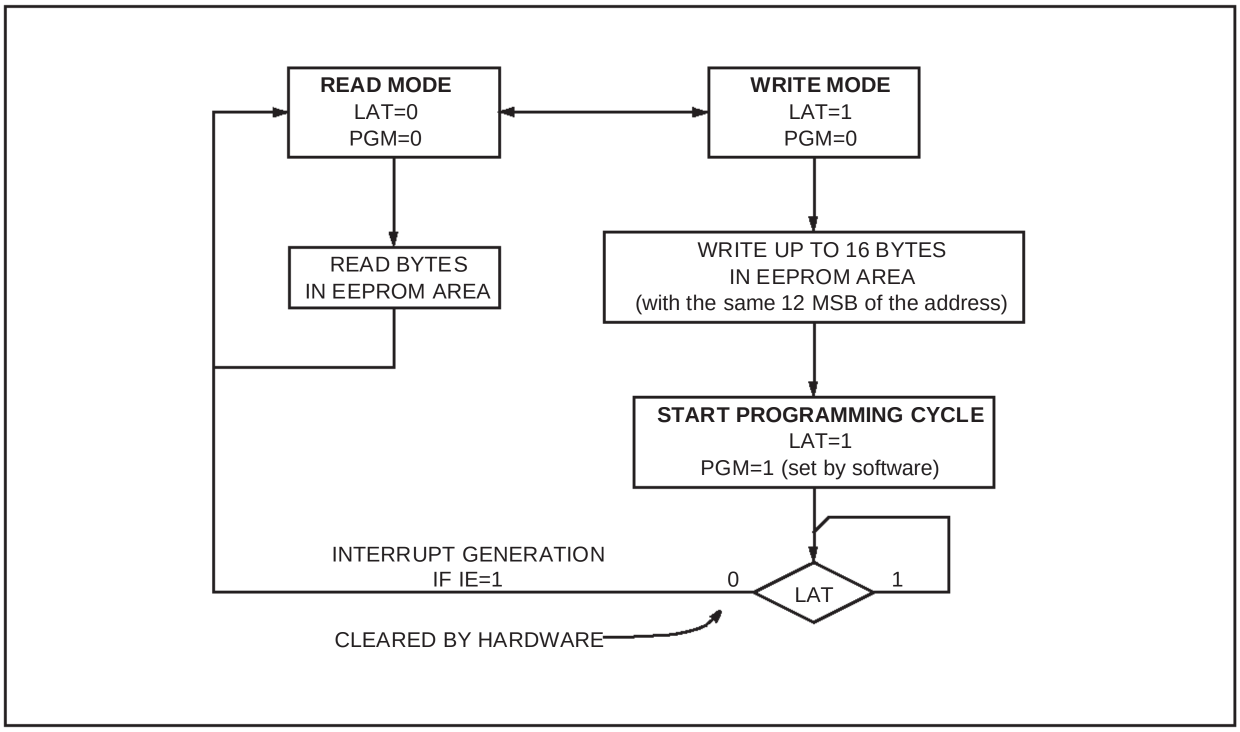
To access the write mode, the LAT bit has to be set by software (the PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 16 data latches according to its address.

When PGM bit is set by the software, all the previous bytes written in the data latches (up to 16) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the four Least Significant Bits of the address can change.

At the end of the programming cycle, the PGM and LAT bits are cleared simultaneously, and an interrupt is generated if the IE bit is set. The Data EEPROM interrupt request is cleared by hardware when the Data EEPROM interrupt vector is fetched.

Note: Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of LAT bit.
It is not possible to read the latched data.
This note is ilustrated by the Figure 13.

Figure 5. Data EEPROM Programming Flowchart


READ MODE
LAT=0
PGM=0
WRITE MODE
LAT=1
PGM=0
READ BYTES IN EEPROM AREA
WRITE UP TO 16 BYTES IN EEPROM AREA (with the same 12 MSB of the address)
START PROGRAMMING CYCLE
LAT=1
PGM=1 (set by software)
INTERRUPT GENERATION IF IE=1
0
LAT
1
CLEARED BY HARDWARE

DATA EEPROM (Cont'd)

1.5.4 Data EEPROM and Power Saving Modes

Wait mode

The DATA EEPROM can enter WAIT mode on execution of the WFI instruction of the microcontroller. The DATA EEPROM will immediately enter this mode if there is no programming in progress, otherwise the DATA EEPROM will finish the cycle and then enter WAIT mode.

Halt mode

The DATA EEPROM immediatly enters HALT mode if the microcontroller executes the HALT instruction. Therefore the EEPROM will stop the function in progress, and data may be corrupted.

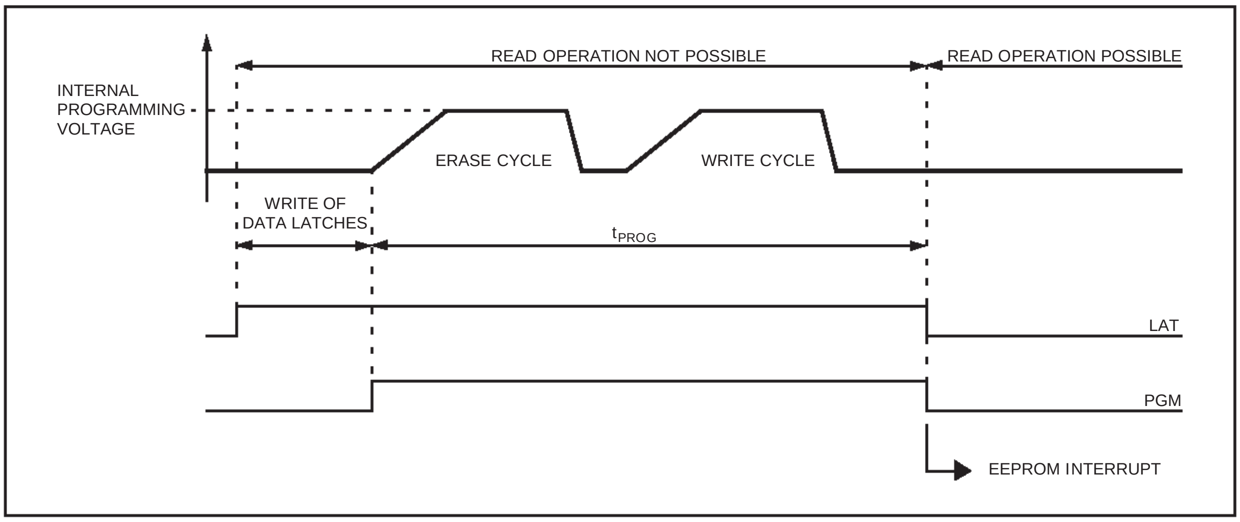
1.5.5 Data EEPROM Access Error Handling

If a read access occurs while LAT=1, then the data bus will not be driven.

If a write access occurs while LAT=0, then the data on the bus will not be latched.

If a programming cycle is interrupted (by software/ RESET action), the memory data will not be guaranteed.

Figure 6. Data EEPROM Programming Cycle



DATA EEPROM (Cont'd)

1.5.6 Register Description

CONTROL/STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	IE	LAT	PGM

Bit 7:3 = Reserved, forced by hardware to 0.

Bit 2 = **IE** *Interrupt enable*
This bit is set and cleared by software. It enables the Data EEPROM interrupt capability when the PGM bit is cleared by hardware. The interrupt request is automatically cleared when the software enters the interrupt routine.
0: Interrupt disabled
1: Interrupt enabled

Bit 1 = **LAT** *Latch Access Transfer*
This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can only be cleared by software if PGM bit is cleared.
0: Read mode
1: Write mode

Bit 0 = **PGM** *Programming control and status*
This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware and an interrupt is generated if the ITE bit is set.
0: Programming finished or not yet started
1: Programming cycle is in progress

Note: if the PGM bit is cleared during the programming cycle, the memory data is not guaranteed.

Table 3. DATA EEPROM Register Map and Reset Values

Address (Hex.)	**Register Label**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
002Ch	**EECSR** Reset Value	0	0	0	0	0	IE 0	RWM 0	PGM 0

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 HALT and WAIT modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

2.3 CPU REGISTERS

The 6 CPU registers shown in Figure 7 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)

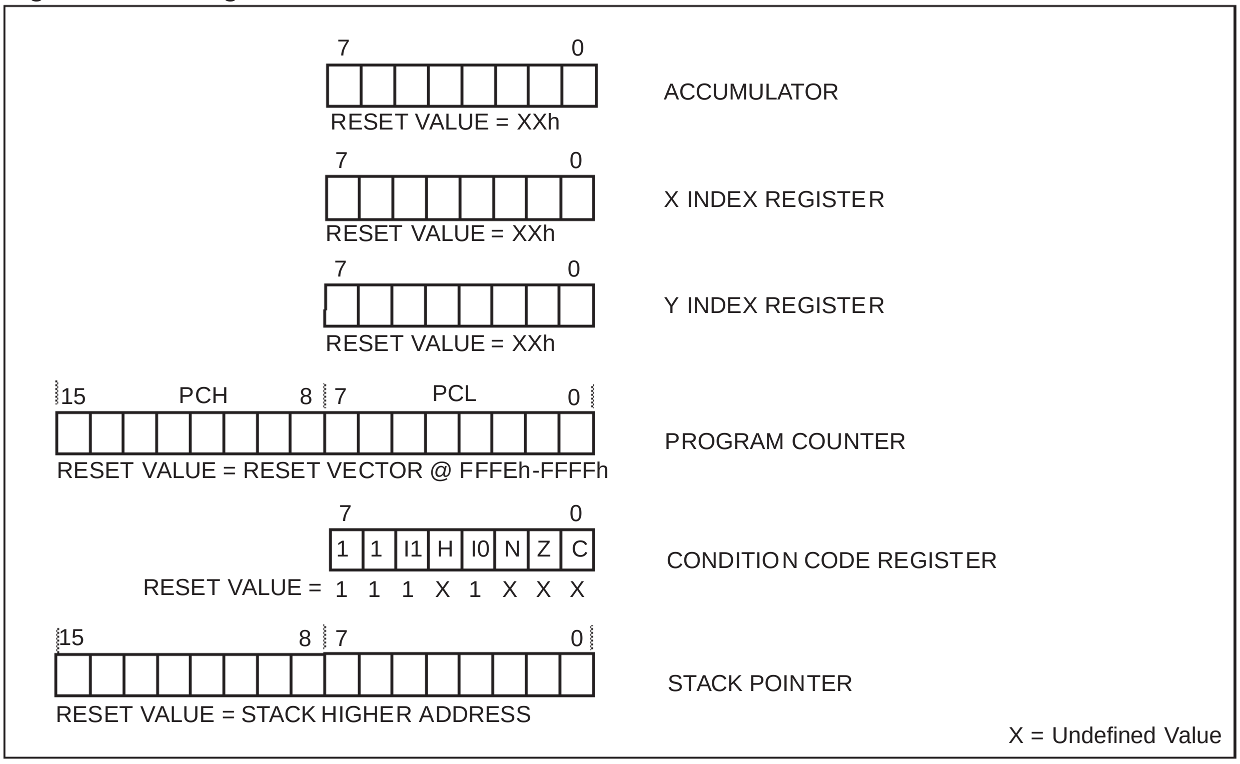
These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

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 PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 7. CPU Registers



CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	I1	H	I0	N	Z	C

The 8-bit Condition Code register contains the interrupt masks 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.

Arithmetic management bits

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 instructions. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: An half carry has occurred.

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

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's a copy of the result 7th bit.
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.

Interrupt management bits

Bit 5,3 = **I1, I0** *Interrupt.*

The combination of the Iand I0 bits gives the current interrupt software priority.

Interrupt Software Priority	I1	I0
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh

15							8
0	0	0	0	0	0	0	1

7							0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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 8).

Since the stack is 256 bytes deep, the 8th most significant bits are 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 (called S) 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.

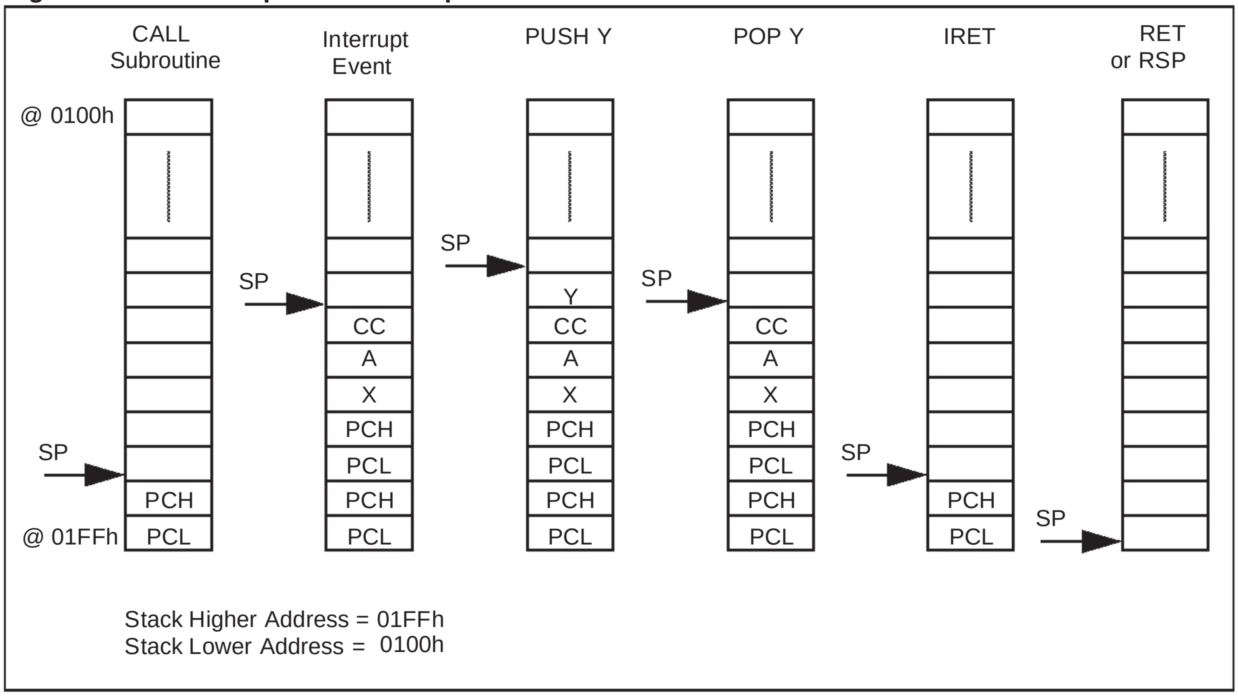
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 8

– 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 8. Stack Manipulation Example



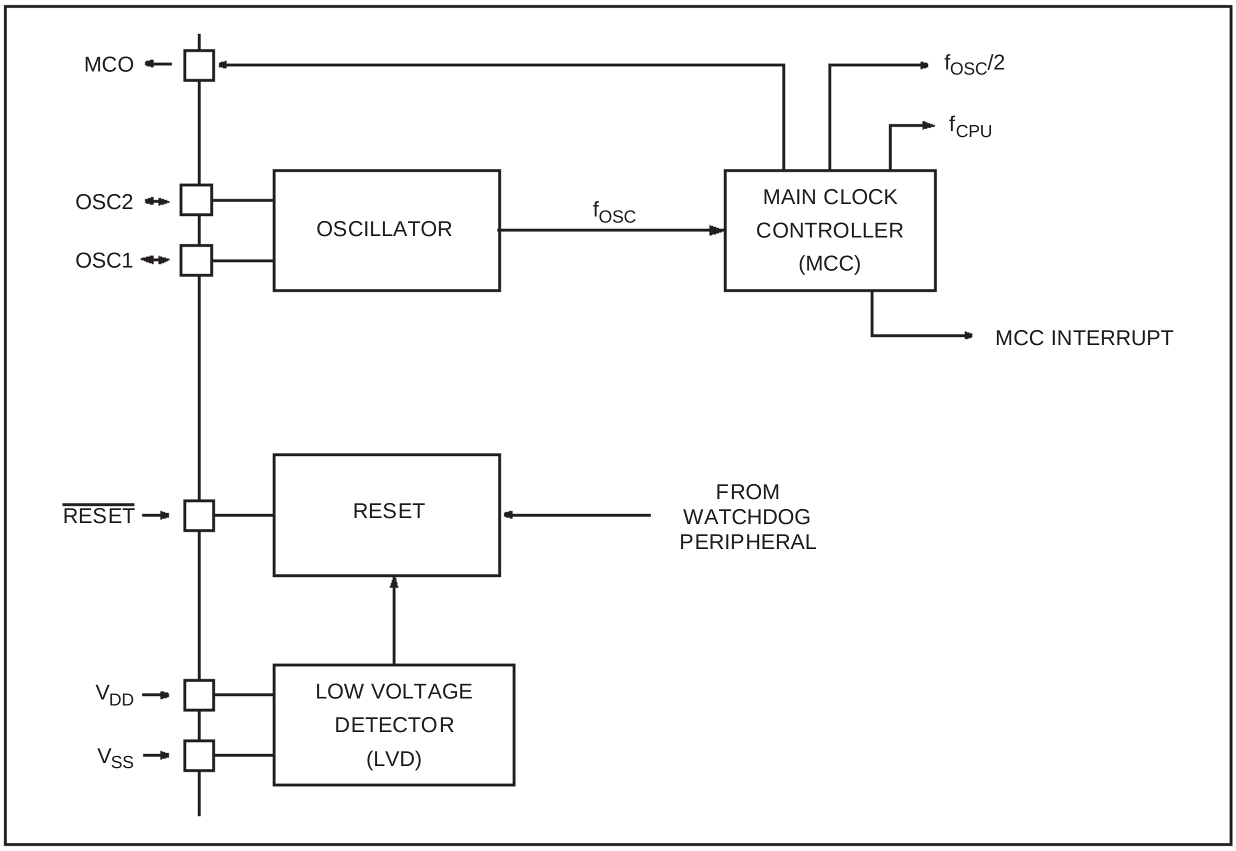
3 SUPPLY, RESET AND CLOCK MANAGEMENT

The ST72311R, ST72511R, ST72512R and ST72532R microcontrollers include a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 9.

Main features

- Main supply low voltage detection (LVD)
- RESET Manager
- Low consumption resonator oscillator
- Main clock controller (MCC)

Figure 9. Clock, RESET, Option and Supply Management Overview



3.1 LOW VOLTAGE DETECTOR (LVD)

To allow the integration of power management features in the application, the Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD Reset circuitry generates a reset when V_{DD} is below:

– V_{IT+} when V_{DD} is rising
– V_{IT-} when V_{DD} is falling

The LVD function is illustrated in Figure 10.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is below V_{IT-}, the MCU can only be in two modes:

– under full software control
– in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

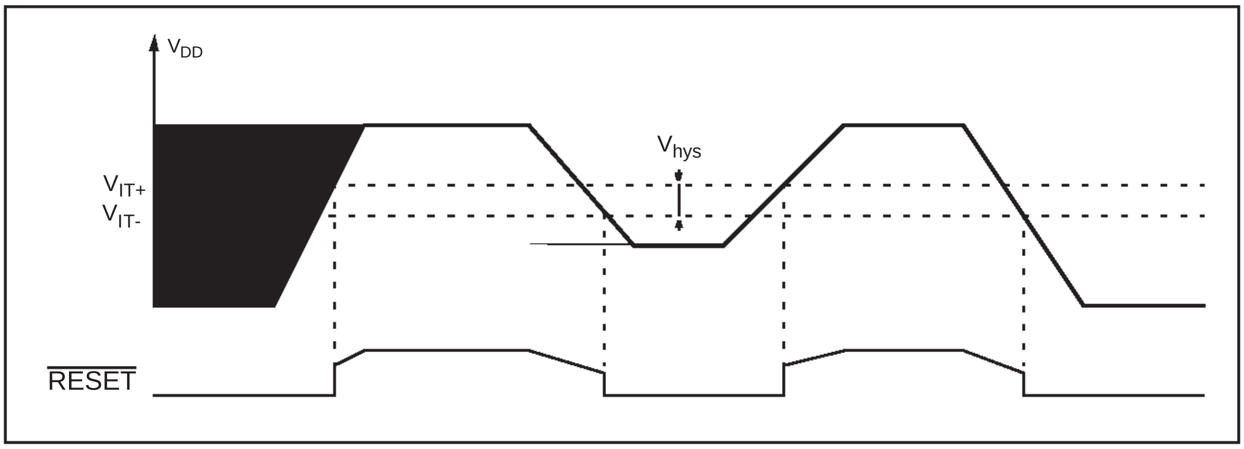
During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Notes:

The LVD allows the device to be used without any external RESET circuitry.

The LVD is an optional function which can be selected when ordering the device (ordering information).

Figure 10. Low Voltage Detector vs Reset



3.2 RESET MANAGER

The RESET block includes three RESET sources as shown in Figure 11:

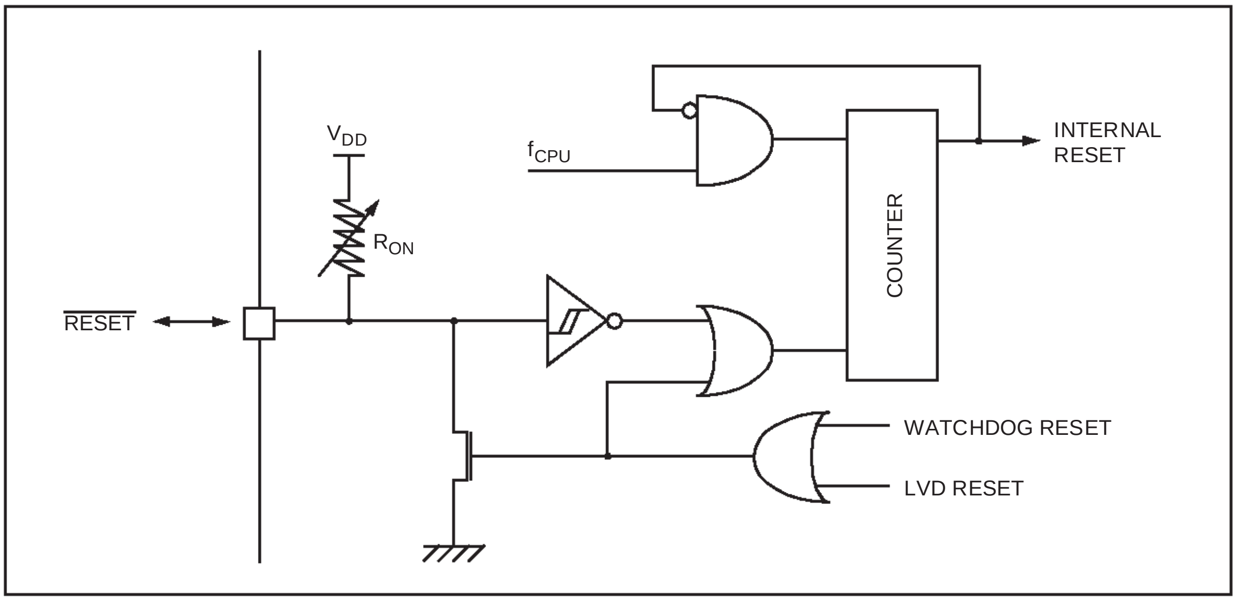
- External $\overline{\text{RESET}}$ source pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

A 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state.

The RESET vector fetch phase duration is 2 clock cycles.

Figure 11. Reset Block Diagram



RESET MANAGER (Cont'd)

External RESET pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor (see Figure 11). This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized. Two RESET sequences can be associated with this RESET source as shown in Figure 12.

When the RESET is generated by a internal source, during the two first phases of the RESET sequence, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low.

Generic Power On RESET

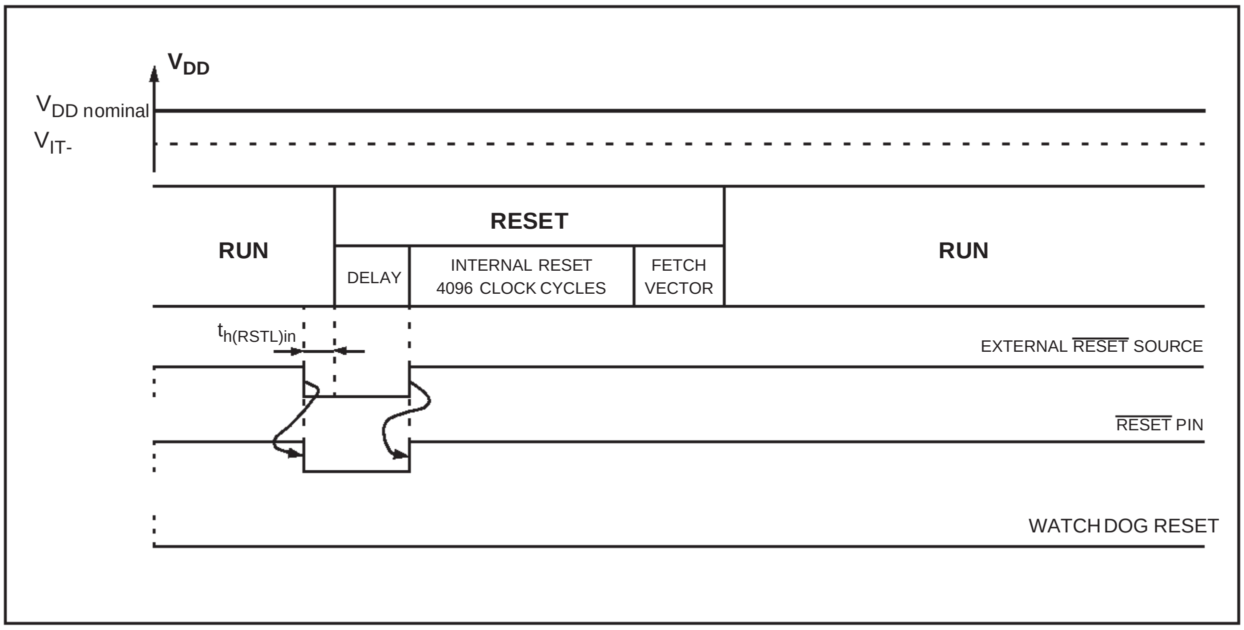
The function of the POR circuit consists of waking up the MCU by detecting (at around 2V) a dynamic (rising edge) variation of the V_{DD} Supply. At the beginning of this sequence, the MCU is configured in the RESET state. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal 4096 CPU cycles delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

To ensure correct start-up, the user should take care that the VDD Supply is stabilized at a sufficient level for the chosen frequency (see Electrical Characteristics) before the reset signal is released. In addition, supply rising must start from 0V.

As a consequence, the POR does not allow to supervise static, slowly rising, or falling, or noisy (oscillating) V_{DD} supplies.

An external RC network connected to the $\overline{\text{RESET}}$ pin, or the LVD reset can be used instead to get the best performance.

Figure 12. External RESET Sequences



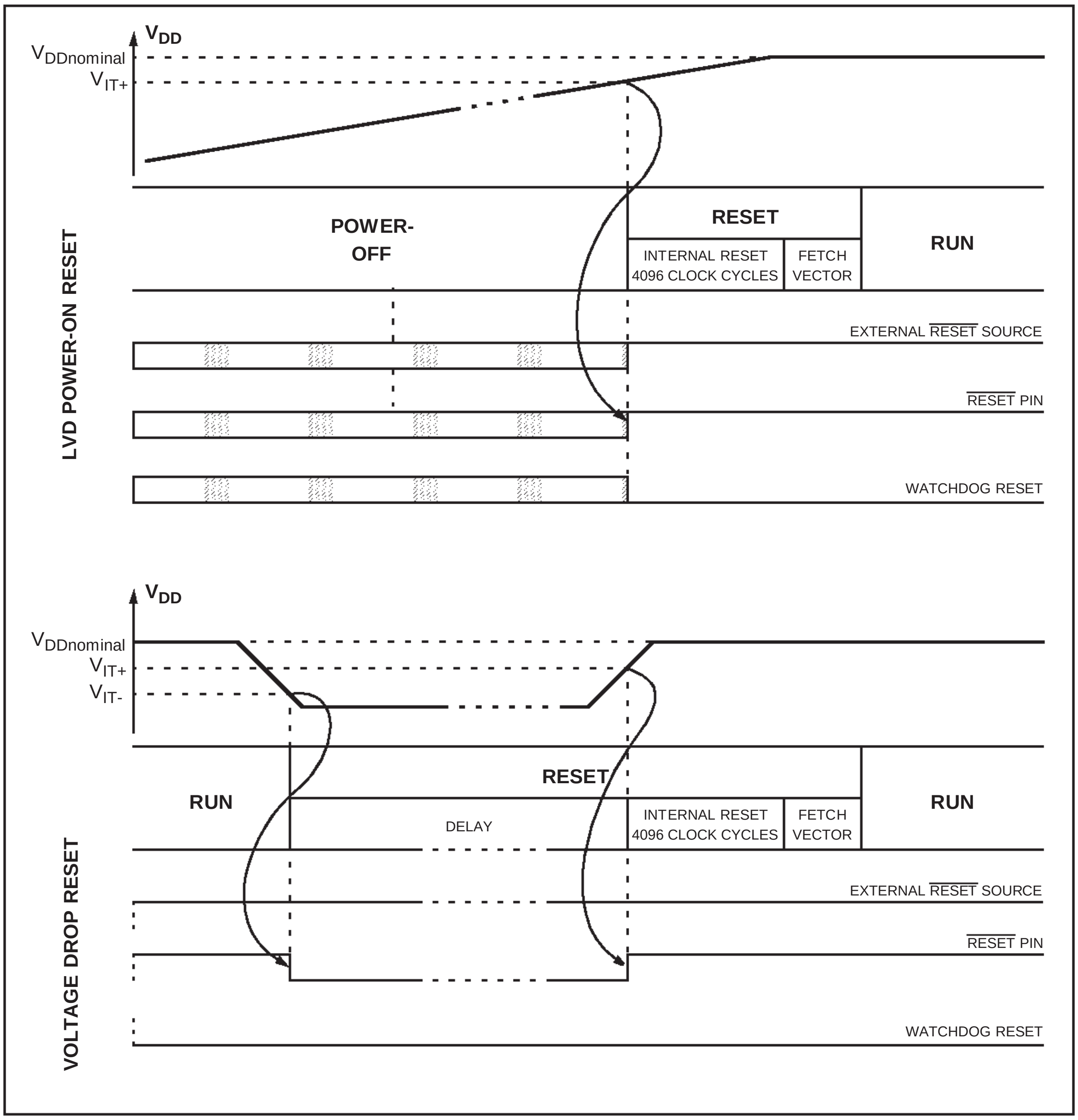
RESET MANAGER (Cont'd)

Internal Low Voltage Detection RESET (option)

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:
- LVD Power-On RESET
- Voltage Drop RESET

In the second sequence, a "delay" phase is used to keep the device in RESET state until V_{DD} rises up to V_{IT+} (see Figure 13).

Figure 13. LVD RESET Sequences

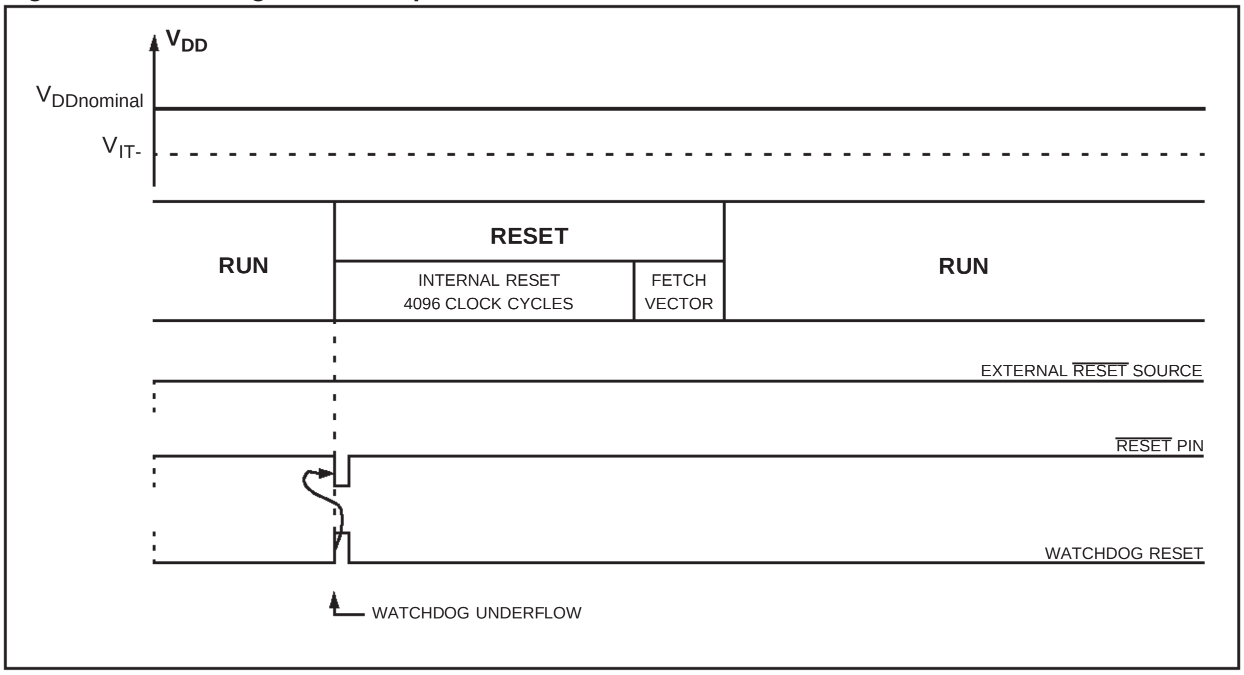


RESET MANAGER (Cont'd)

Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow has the shortest reset phase (see Figure 14).

Figure 14. Watchdog RESET Sequence



3.3 LOW CONSUMPTION OSCILLATOR

The main clock of the ST7 can be generated by two different sources:

- an external source
- a crystal or ceramic resonator oscillators

External Clock Source

In this mode, a square clock signal with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to V_{DD} through a R_{OBP} resistance (see Figure 15).

Crystal/Ceramic Oscillators

This oscillator (based on constant current source) is optimized in terms of consumption and has the advantage of producing a very accurate rate on the main clock of the ST7.
When using this oscillator, the resonator and the load capacitances have to be connected as shown in Figure 16 and have to be mounted as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time.

This oscillator is not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

Figure 15. External Clock

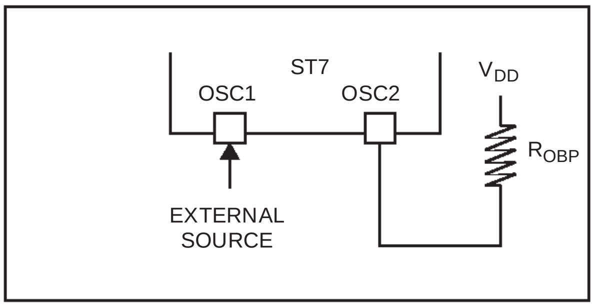
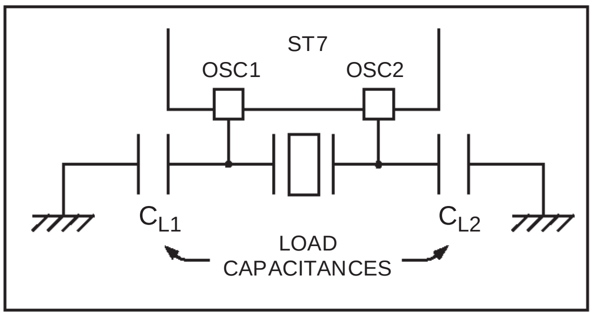


Figure 16. Crystal/Ceramic Resonator



3.4 MAIN CLOCK CONTROLLER (MCC)

The MCC block supplies the clock for the ST7 CPU and its internal peripherals. It allows to manage the power saving modes such as the SLOW and ACTIVE-HALT modes. The whole functionality is managed by the Main Clock Control/Status Register (MCCSR) and the Miscellaneous Register 1 (MISCR1).

The MCC block consists of:

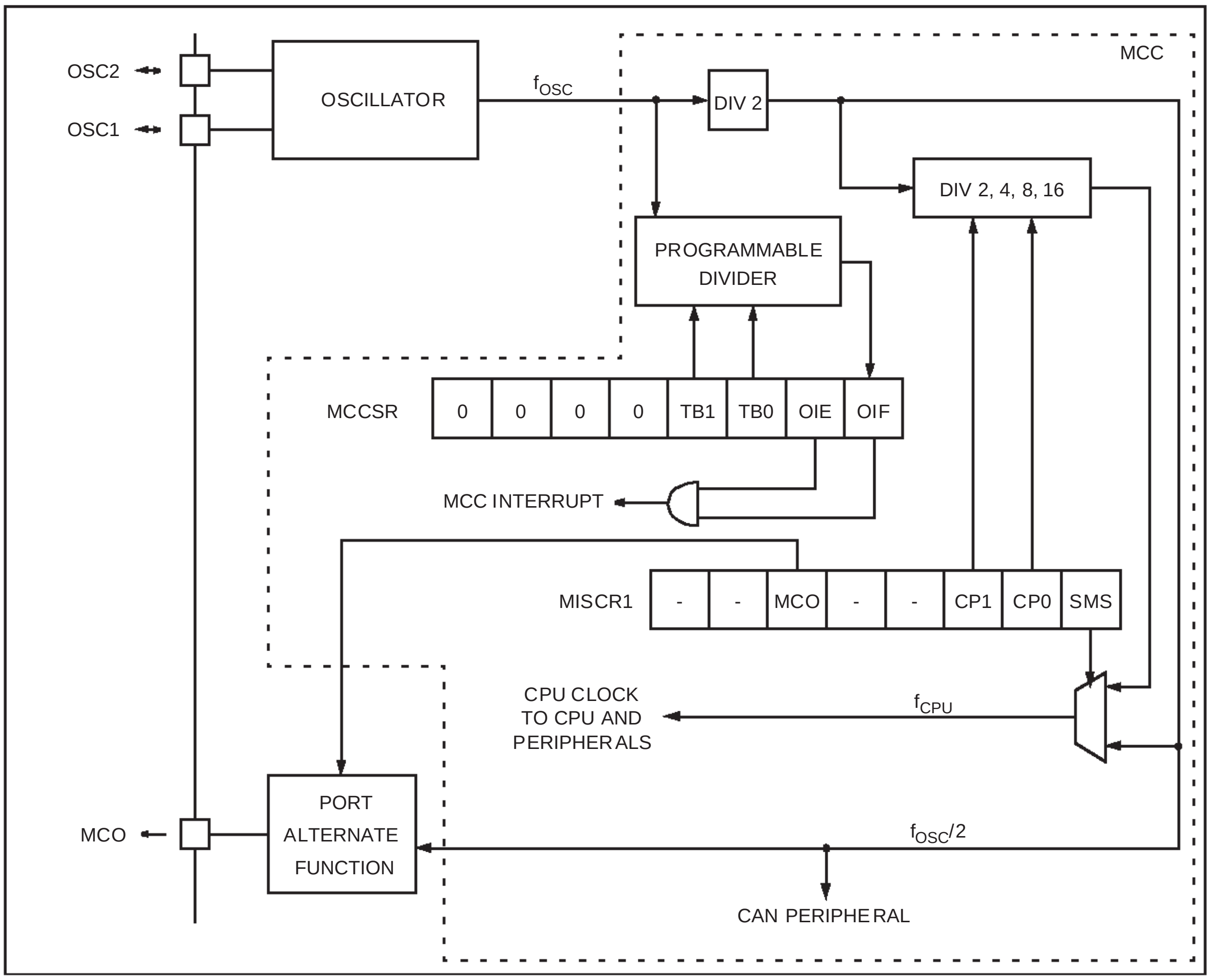
– a programmable CPU clock prescaler
– a time base counter with interrupt capability
– a clock-out signal to supply external devices

The prescaler allows to select the main clock frequency and is controlled with three bits of the MISCR1: CP1, CP0 and SMS.

The counter allows to generate an interrupt based on a accurate real time clock. Four different time bases depending directly on f_{OSC} are available. The whole functionality is controlled by four bits of the MCCSR register: TB1, TB0, OIE and OIF.

The clock-out capability allows to configure a dedicated I/O port pin as an $f_{OSC}/2$ clock out to drive external devices. It is controlled by the MCO bit in the MISCR1 register.
When selected, the clock out pin suspends the clock during ACTIVE-HALT mode.

Figure 17. Main Clock Controller (MCC) Block Diagram



MAIN CLOCK CONTROLLER (Cont'd)

MISCELLANEOUS REGISTER 1 (MISCR1)

See "MISCELLANEOUS REGISTERS" Section.

MAIN CLOCK CONTROL/STATUS REGISTER (MCCSR)

Read/Write

Reset Value: 0000 0001 (01h)

7							0
0	0	0	0	TB1	TB0	OIE	OIF

Bit 7:4 = Reserved, always read as 0.

Bit 3:2 = **TB1-TB0** *Time base control*

These bits select the programmable divider time base. They are set and cleared by software.

Counter Prescaler	Time Base		TB1	TB0
	f_{OSC} =8MHz	f_{OSC}=16MHz		
32000	4ms	2ms	0	0
64000	8ms	4ms	0	1
160000	20ms	10ms	1	0
400000	50ms	25ms	1	1

A modification of the time base is taken into account at the end of the current period (previously set) to avoid unwanted time shift. This allows to use this time base as a real time clock.

Bit 1 = **OIE** *Oscillator interrupt enable*

This bit set and cleared by software.
0: Oscillator interrupt disabled
1: Oscillator interrupt enabled
This interrupt allows to exit from ACTIVE-HALT mode.
When this bit is set, calling the ST7 software HALT instruction enters the ACTIVE-HALT power saving mode.

Bit 0 = **OIF** *Oscillator interrupt flag*

This bit is set by hardware and cleared by software reading the CSR register. It indicates when set that the main oscillator has measured the selected elapsed time (TB1:0).
0: Timeout not reached
1: Timeout reached

Warning: The BRES and BSET instructions must not be used on the MCCSR register to avoid unintentionally clearing the OIF bit.

Table 4. MCC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0029h	**MCCSR** Reset Value	0	0	0	0	TB1 0	TB0 0	OIE 0	OIF 1

4 INTERRUPTS & POWER SAVING MODES

4.1 INTERRUPTS

4.1.1 Introduction

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
  - Up to 4 software programmable nesting levels
  - Up to 16 interrupt vectors fixed by hardware
  - 3 non maskable events: NMI, RESET, TRAP

This interrupt management is based on:

– Bit 5 and bit 3 of the CPU CC register (I1:0),

– Interrupt software priority registers (ISPRx),

– Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order.

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

4.1.2 Interrupt Masking and Processing Flow

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see Table 5). The processing flow is shown in Figure 18

When an interrupt request 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.

– I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.

– 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 “Interrupt Mapping” table for vector addresses).

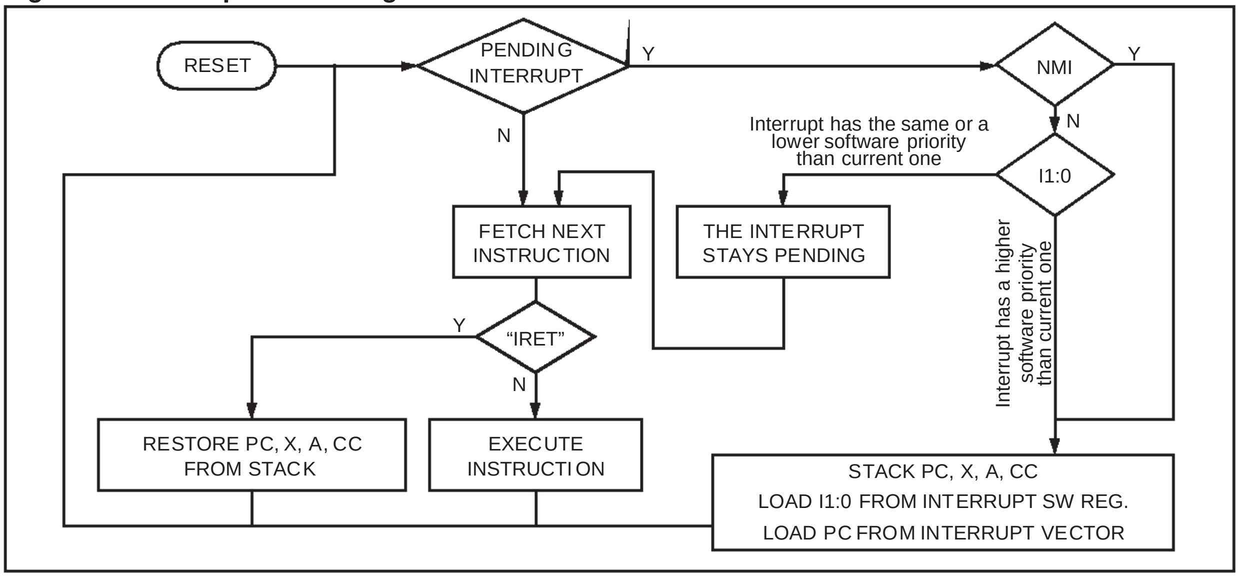
The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

Table 5. Interrupt Software Priority Levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2		0	0
Level 3 (= interrupt disable)	High	1	1

Figure 18. Interrupt Processing Flowchart


RESET
PENDING INTERRUPT
Y
N
NMI
Y
N
Interrupt has the same or a lower software priority than current one
I1:0
FETCH NEXT INSTRUCTION
THE INTERRUPT STAYS PENDING
Interrupt has a higher software priority than current one
Y
“IRET”
N
RESTORE PC, X, A, CC FROM STACK
EXECUTE INSTRUCTION
STACK PC, X, A, CC
LOAD I1:0 FROM INTERRUPT SW REG.
LOAD PC FROM INTERRUPT VECTOR

INTERRUPTS (Cont'd)

Servicing Pending Interrupts

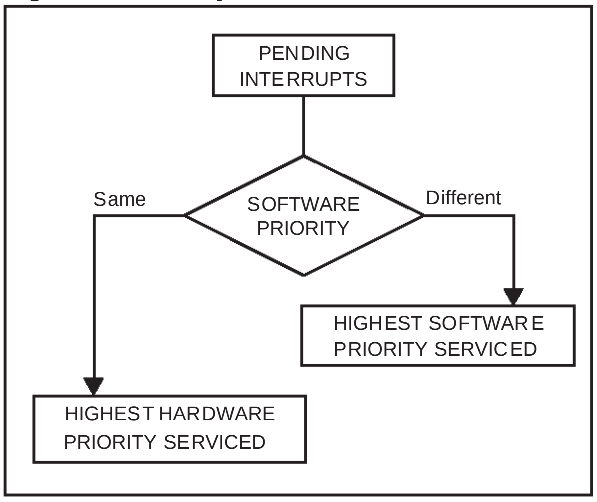
As several interrupts can be pending at the same time, the interrupt to be taken into account is determined by the following two-step process:

– the highest software priority interrupt is serviced,

– if several interrupts have the same software priority then the interrupt with the highest hardware priority is serviced first.

Figure 19 describes this decision process.

Figure 19. Priority Decision Process


PENDING INTERRUPTS
Same
SOFTWARE PRIORITY
Different
HIGHEST SOFTWARE PRIORITY SERVICED
HIGHEST HARDWARE PRIORITY SERVICED

When an interrupt request is not serviced immediately, it is latched and then processed when its software priority combined with the hardware priority becomes the highest one.

Note 1: The hardware priority is exclusive while the software one is not. This allows the previous process to succeed with only one interrupt.
Note 2: RESET, TRAP and NMI are non maskable and they can be considered as having the highest software priority in the decision process.

Different Interrupt Vector Sources

Two interrupt source types are managed by the ST7 interrupt controller: the non-maskable type (RESET, NMI, TRAP) and the maskable type (external or from internal peripherals).

Non-Maskable Sources

These sources are processed regardless of the state of the I1 and I0 bits of the CC register (see Figure 18). After stacking the PC, X, A and CC registers (except for RESET), the corresponding vector is loaded in the PC register and the I1 and I0 bits of the CC are set to disable interrupts (level 3). These sources allow the processor to exit HALT mode.

■ NMI (Non Maskable Hardware Interrupt)

This hardware interrupt occurs when a specific edge is detected on the dedicated NMI pin. Its detailed specification is given in the Miscellaneous register chapter.

■ TRAP (Non Maskable Software Interrupt)

This software interrupt is serviced when the TRAP instruction is executed. It will be serviced according to the flowchart on Figure 18 as an NMI.

■ RESET

The RESET source has the highest priority in the ST7. This means that the first current routine has the highest software priority (level 3) and the highest hardware priority.
See the RESET chapter for more details.

Maskable Sources

Maskable interrupt vector sources can be serviced if the corresponding interrupt is enabled and if its own interrupt software priority (in ISPRx registers) is higher than the one currently being serviced (I1 and I0 in CC register). If any of these two conditions is false, the interrupt is latched and thus remains pending.

■ External Interrupts

External interrupts allow the processor to exit from HALT low power mode.
External interrupt sensitivity is software selectable through the Miscellaneous registers (MISCRx).
External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.
If several input pins of a group connected to the same interrupt line are selected simultaneously, these will be logically ORed.

■ Peripheral Interrupts

Usually the peripheral interrupts cause the MCU to exit from HALT mode except those mentioned in the "Interrupt Mapping" table.
A peripheral interrupt occurs when a specific flag is set in the peripheral status registers and if the corresponding enable bit is set in the peripheral control register.
The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.
Note: The clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being serviced) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

4.1.3 Interrupts and Low Power Modes

All interrupts allow the processor to exit the WAIT low power mode. On the contrary, only external and other specified interrupts allow the processor to exit the HALT modes (see column "Exit from HALT" in "Interrupt Mapping" table). When several pending interrupts are present while exiting HALT mode, the first one serviced can only be an interrupt with exit from HALT mode capability and it is selected through the same decision process shown in Figure 19

Note: If an interrupt, that is not able to Exit from HALT mode, is pending with the highest priority when exiting HALT mode, this interrupt is serviced after the first one serviced.

4.1.4 Concurrent and Nested Interrupt Management

The following Figure 20 and Figure 21 show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in Figure 21 The interrupt hardware priority is given in this order from the lowest to the highest: MAIN, IT4, IT3, IT2, IT1, IT0, NMI. The software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.

Figure 20. Concurrent interrupt management

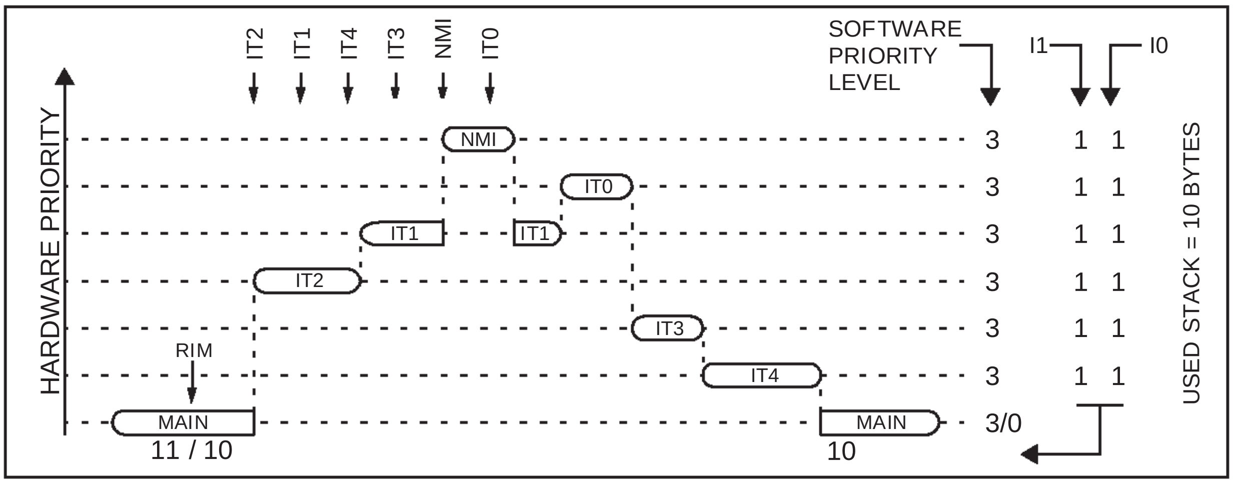
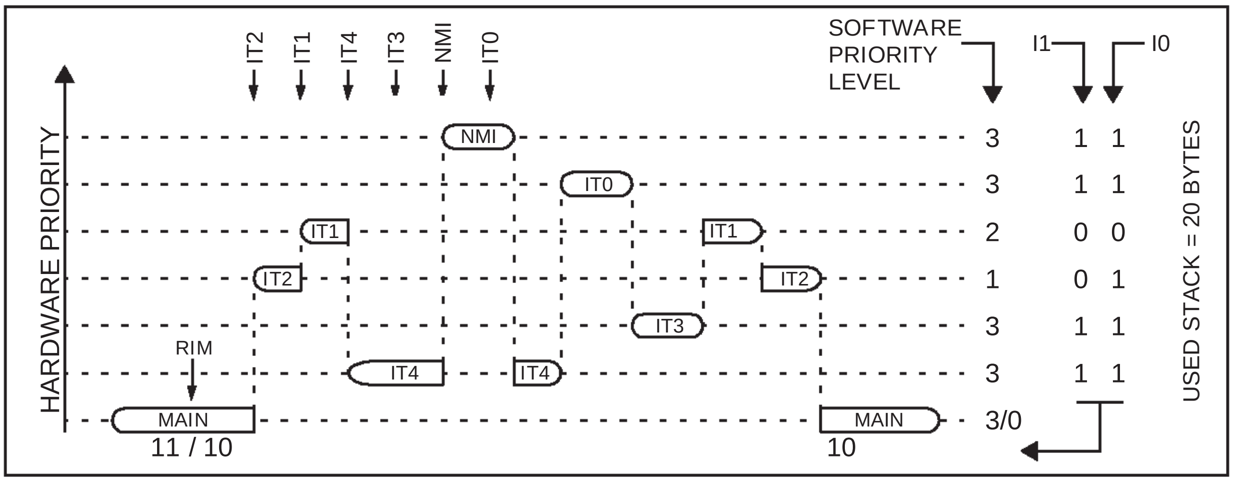


Figure 21. Nested interrupt management



INTERRUPTS (Cont'd)

4.1.5 Interrupt Register Descriptions

CPU CC REGISTER INTERRUPT BITS

Read/Write

Reset Value: 111x 1010 (xAh)

7							0
1	1	**I1**	H	**I0**	N	Z	C

Bit 5, 3 = **I1, I0** *Software Interrupt Priority*

These two bits indicate the current interrupt software priority.

Interrupt Software Priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2		0	0
Level 3 (= interrupt disable*)	High	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see "Interrupt Dedicated Instruction Set" table).

***Note**: NMI, TRAP and RESET events are non maskable sources and can interrupt a level 3 program.

INTERRUPT SOFTWARE PRIORITY REGISTERS (ISPRX)

Read/Write (bit 7:4 of **ISPR3** are read only)

Reset Values: 1111 1111 (FFh)

	7							0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12

These four registers contain the interrupt software priority of each interrupt vector.

– Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondance is shown in the following table.

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_0 bits*
FFF9h-FFF8h	I1_1 and I0_1 bits
...	...
FFE1h-FFE0h	I1_13 and I0_13 bits

– Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

– Level 0 can not be written (I1_x=1, I0_x=0). In this case, the previously stored value is kept. (example: previous=CFh, write=64h, result=44h)

The RESET, TRAP and NMI vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

***Note**: Bits in the ISPRx registers which correspond to the NMI can be read and written but they are not significant in the interrupt process management.

Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behaviour has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

INTERRUPTS (Cont'd)

Table 6. Dedicated Interrupt Instruction Set

Instruction	New Description	Function/Example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						
POP CC	Pop CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load 10 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load 11 in I1:0 of CC	1		1			
TRAP	Software trap	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

Note: During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

In order not to lose the current software priority level, the RIM, SIM, HALT, WFI and POP CC instructions should never be used in an interrupt routine.

Table 7. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT	Address Vector
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
0	NMI	External Non Maskable Interrupt	MISCR2		yes	FFFAh-FFFBh
1	MCC	Main Clock Controller Time Base Interrupt	MCCSR			FFF8h-FFF9h
2	EI0	External Interrupt Port A3..0	N/A			FFF6h-FFF7h
3	EI1	External Interrupt Port F2..0				FFF4h-FFF5h
4	EI2	External Interrupt Port B3..0				FFF2h-FFF3h
5	EI3	External Interrupt Port B7..4				FFF0h-FFF1h
6	CAN	CAN Peripheral Interrupts	CANISR			FFEEh-FFEFh
7	SPI	SPI Peripheral Interrupts	SPISR		no	FFECh-FFEDh
8	TIMER A	TIMER A Peripheral Interrupts	TASR			FFEAh-FFEBh
9	TIMER B	TIMER B Peripheral Interrupts	TBSR			FFE8h-FFE9h
10	SCI	SCI Peripheral Interrupts	SCISR			FFE6h-FFE7h
11	EEPROM	EEPROM Interrupt	EECSR			FFE4h-FFE5h
12		Not Used		Lowest Priority		FFE2h-FFE3h
13	PWM ART	PWM ART Overflow Interrupt	ARTCSR		Yes	FFE0h-FFE1h

INTERRUPTS (Cont'd)

Table 8. Nested Interrupts Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0024h	**ISPR0**	EI1		EI0		MCC		NMI	
		I1_3	I0_3	I1_2	I0_2	I1_1	I0_1		
	Reset Value	1	1	1	1	1	1	1	1
0025h	**ISPR1**	SPI		CAN		EI3		EI2	
		I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
	Reset Value	1	1	1	1	1	1	1	1
0026h	**ISPR2**	EEPROM		SCI		TIMER B		TIMER A	
		I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
	Reset Value	1	1	1	1	1	1	1	1
0027h	**ISPR3**					PWMART		Not Used	
						I1_13	I0_13	I1_12	I0_12
	Reset Value	1	1	1	1	1	1	1	1

4.2 POWER SAVING MODES

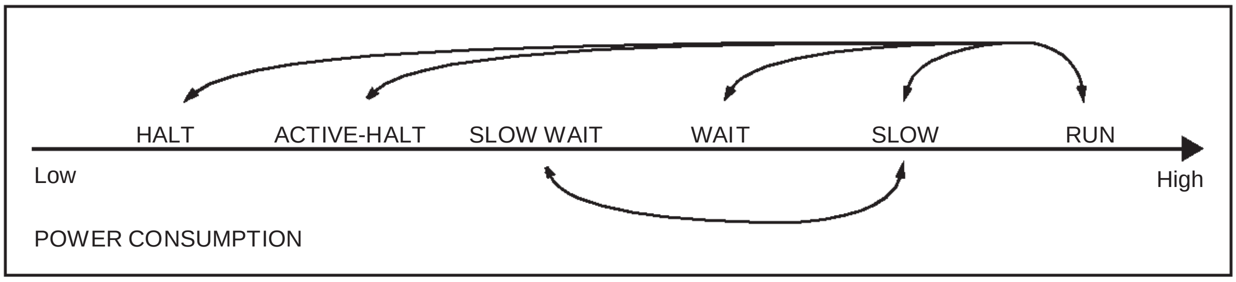
4.2.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7.

After a RESET the normal operating mode is selected by default (RUN mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided by 2 (f_{CPU}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the the oscillator status.

Figure 22. Power saving mode consumption / transitions



4.2.2 HALT Modes

The HALT modes are the lowest power consumption modes of the MCU. They are entered by executing the ST7 HALT instruction (see Figure 24).

Two different HALT modes can be distinguished:

– HALT: main oscillator is turned off,

– ACTIVE-HALT: only main oscillator is running.

The decision to enter either in HALT or ACTIVE-HALT mode is given by the main oscillator enable interrupt flag (OIE bit in CROSS-MCCSR register: see Table 9).

When entering HALT modes, the I1 and I0 bits in the CC Register are forced to level 0 (“10”) to enable interrupts.

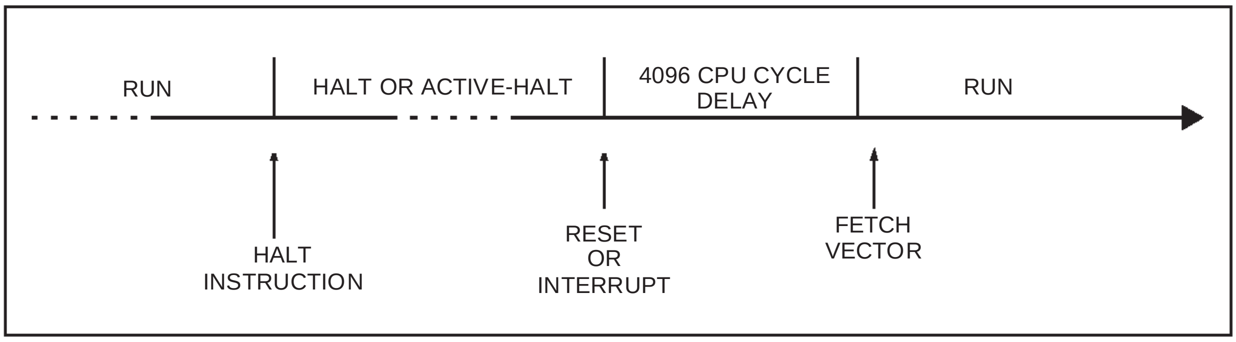
The MCU can exit HALT or ACTIVE-HALT modes on reception of an interrupt with Exit from Halt Mode capability or a reset (see Figure 7 on page 35). A 4096 CPU clock cycles delay is performed before the CPU operation resumes (see Figure 23).

After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up.

Table 9. HALT Modes selection

MCCSR OIE flag	Power Saving Mode entered when HALT instruction is executed
0	HALT (reset if watchdog enabled)
1	ACTIVE-HALT (no reset if watchdog enabled)

Figure 23. HALT /ACTIVE-HALT Modes timing overview



POWER SAVING MODES (Cont'd)

Standard HALT mode

In this mode the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).
The compatibility of Watchdog operation with Halt mode is configured by the "WDGHALT" option bit of the OPTION BYTE. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET (see Section 9.1 OPTION BYTES for more details).
When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 4096 CPU cycle delay is used to stabilize the oscillator.

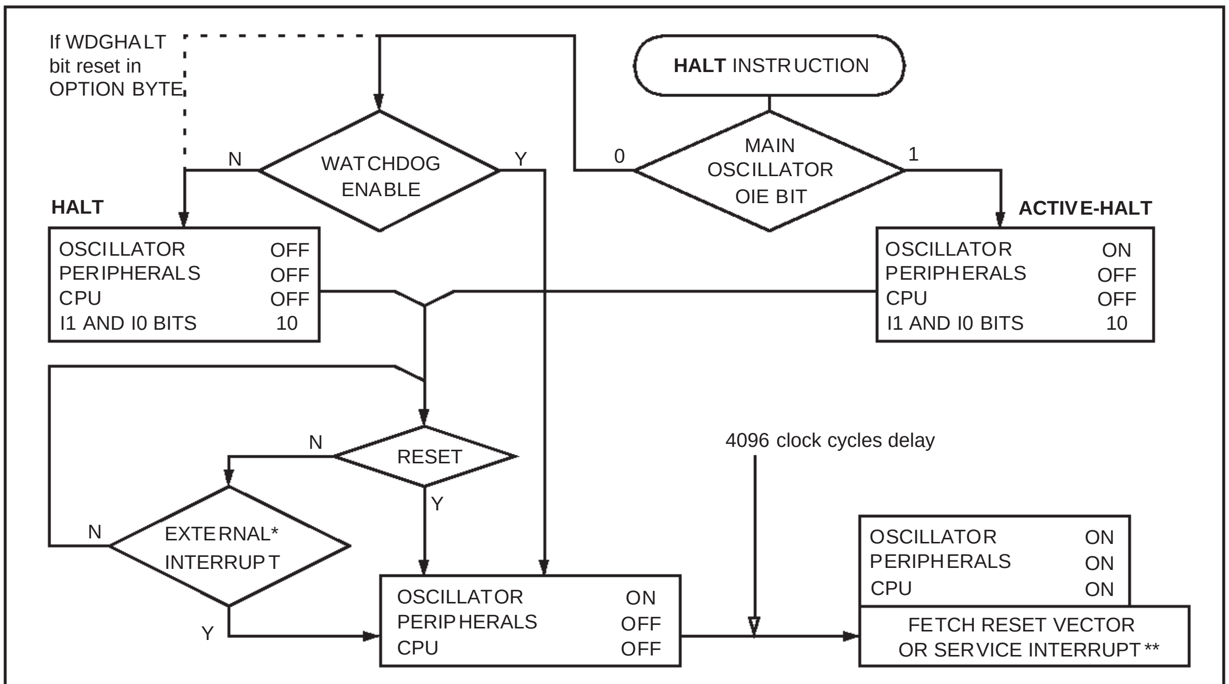
Specific ACTIVE-HALT mode

As soon as the interrupt capability of the main oscillator is selected (OIE bit set), the HALT instruction will make the device enter a specific ACTIVE-HALT power saving mode instead of the standard HALT one.
This mode consists of having only the main oscillator and its associated counter running to keep a wake-up time base. All other peripherals are not clocked except the ones which get their clock supply from another clock generator (such as external or auxiliary oscillator).

The safeguard against staying locked in this ACTIVE-HALT mode is insured by the oscillator interrupt.

Note: As soon as the interrupt capability of one of the oscillators is selected (OIE bit set), entering in ACTIVE-HALT mode while the Watchdog is active does not generate a RESET.
This means that the device cannot to spend more than a defined delay in this power saving mode.

Figure 24. HALT modes flow-chart



Notes: * External interrupt or internal interrupts with Exit from Halt Mode capability
** Before servicing an interrupt, the CC register is pushed on the stack.

POWER SAVING MODES (Cont'd)

4.2.3 WAIT Mode

WAIT mode places the MCU in a low power consumption mode by stopping the CPU.

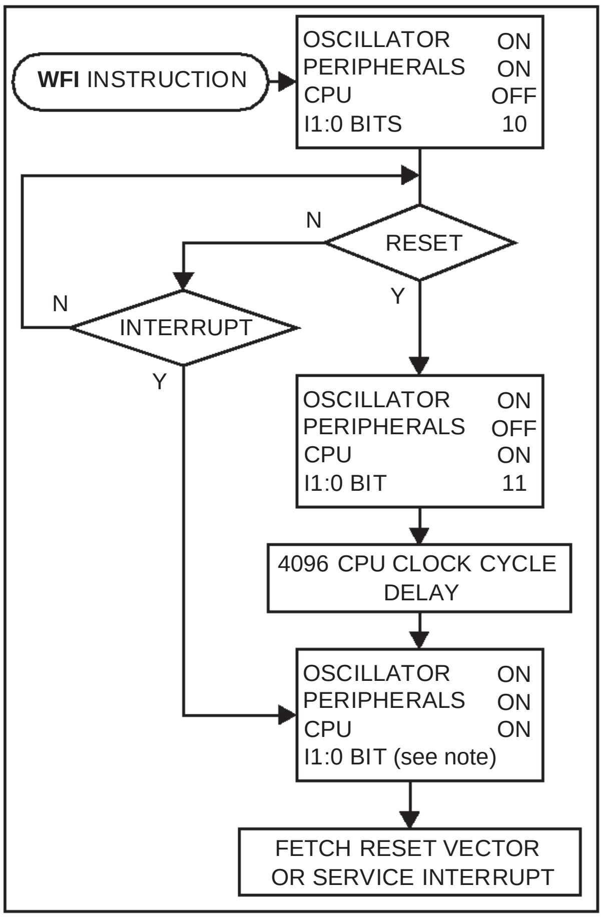
This power saving mode is selected by calling the "WFI" ST7 software instruction.

All peripherals remain active. During WAIT mode, the I1 and I0 bits of the CC register are forced to level 0 ("10"), to enable all interrupts. All other registers and memory remain unchanged. The MCU remains 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 25.

Figure 25. WAIT Mode Flow-chart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I1:0 bits of the CC register are set according to the software priority of the serviced interrupt routine. They are restored when the CC register is popped.

4.2.4 SLOW Mode

This mode has two targets:

– To reduce power consumption by decreasing the internal clock in the device,

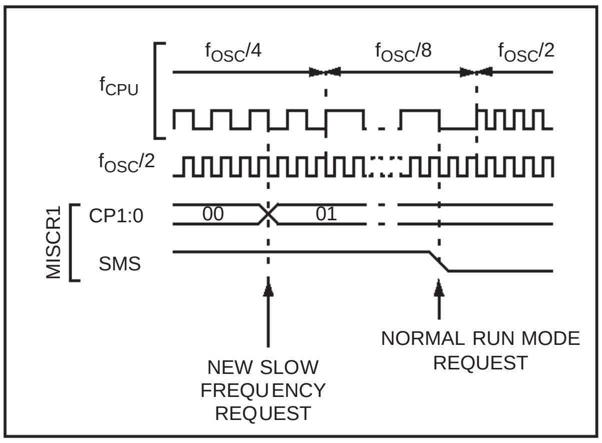
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by three bits in the main oscillator MISCR1 register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the oscillator frequency can be divided by 4, 8, 16 or 32 instead of 2 in normal operating mode. The CPU and peripherals (except CAN, see Note) are clocked at this lower frequency.

Note: Before entering SLOW mode and in order to guarantee low power operation, the CAN peripheral must be placed by software in STANDBY mode.

Figure 26. SLOW Mode Clock Transitions



5 ON-CHIP PERIPHERALS

5.1 I/O PORTS

5.1.1 Introduction

The I/O ports offer different functional modes:
– transfer of data through digital inputs and outputs and for specific pins:
– external interrupt generation
– alternate signal input/output for the on-chip peripherals (SPI, SCI, TIMERs...).

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

5.1.2 Functional Description

Each port is associated to 2 main registers:

– Data Register (DR)

– Data Direction Register (DDR)

and one 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, for specific port which do not provide this register refer to the I/O Port Implementation section. The generic I/O block diagram is shown on Figure 27

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.

Note1: Writing the DR register modifies the latch value but does not affect the pin status.
Note2: When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the ports is configured as an output.

External interrupt function

When an I/O is configured in Input with Interrupt, an event on this I/O can generate an external Interrupt request to the CPU.

Each pin can independently generate an Interrupt request. The interrupt sensitivity is given independently according to the description mentioned in the Miscellaneous register.

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

In case of a floating input with interrupt configuration, special cares mentioned in the I/O port implementation section have to be taken.

Output Mode

The output configuration is selected by setting the corresponding DDR register bit.

In this case, writing 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.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	V_{SS}	Vss
1	V_{DD}	Floating

Note: In this mode, interrupt function is disabled.

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 the 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: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.

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)

Figure 27. I/O Block Diagram

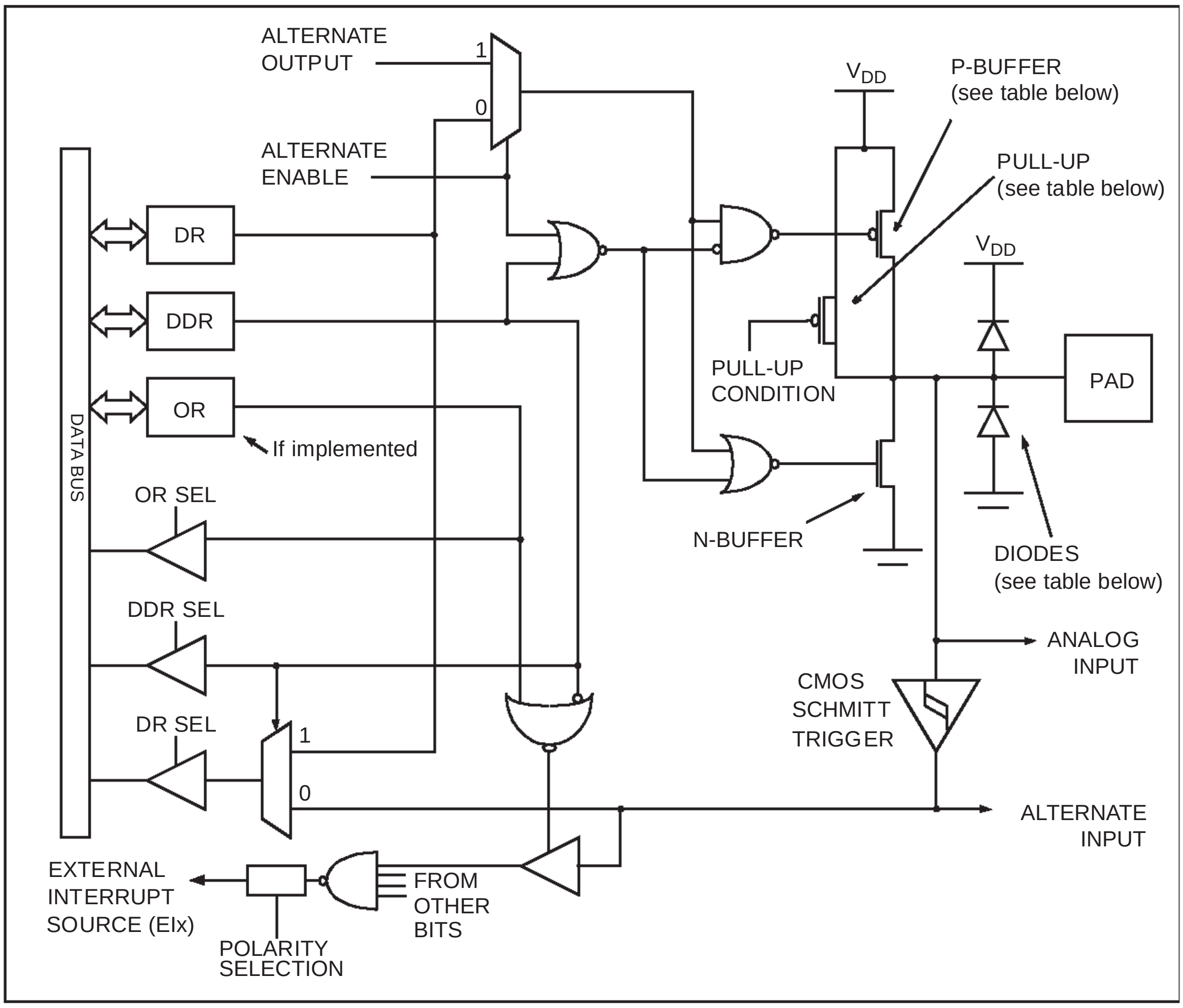


Table 10. Port Mode Options

Configuration Mode		Pull-Up	P-Buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open Drain (logic level)		Off		
	True Open Drain	NI	NI	NI (see note)	

Legend:NI - not implemented
Off - implemented not activated
On - implemented and activated

Note: the diode to V_{DD} is not implemented in the true open drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

I/O PORTS (Cont'd)

5.1.3 I/O Port Implementation

The I/O port register configurations are summarised as following.

Standard Ports

PA5:4, PC7:0, PD7:0, PE7:3, PE1:0, PF7:3

MODE	DDR	OR
floating input	0	0
pull-up input	0	1
open drain output	1	0
push-pull output	1	1

Interrupt Ports

PA2:0, PB7:5, PB2:0, PF1:0 (with pull-up)

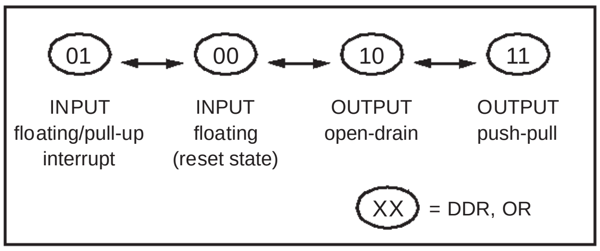
MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

PA3, PB4, PB3, PF2 (without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
open drain output	1	0
push-pull output	1	1

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 28 Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 28. Interrupt I/O Port State Transition


01
00
10
11
INPUT floating/pull-up interrupt
INPUT floating (reset state)
OUTPUT open-drain
OUTPUT push-pull
XX = DDR, OR

True Open Drain Ports

PA7:6

MODE	DDR
floating input	0
open drain (high sink ports)	1

Pull-up Input Port (CANTX requirement)

PE2

MODE
pull-up input

Table 11. Port Configuration

Port	Pin name	Input		Output		
		OR = 0	OR = 1	OR = 0	OR = 1	High-Sink
Port A	PA7:6	floating		true open-drain		Yes
	PA5:4	floating	pull-up	open drain	push-pull	
	PA3	floating	floating interrupt	open drain	push-pull	No
	PA2:0	floating	pull-up interrupt	open drain	push-pull	
Port B	PB4, PB3	floating	floating interrupt	open drain	push-pull	
	PB7:5, PB2:0	floating	pull-up interrupt	open drain	push-pull	
Port C	PC7:0	floating	pull-up	open drain	push-pull	PC3:2 only
Port D	PD7:0	floating	pull-up	open drain	push-pull	No
Port E	PE7:3, PE1:0	floating	pull-up	open drain	push-pull	PE7:4 only
	PE2	pull-up input only *				No
Port F	PF7:3	floating	pull-up	open drain	push-pull	PF7:6 only
	PF2	floating	floating interrupt	open drain	push-pull	No
	PF1:0	floating	pull-up interrupt	open drain	push-pull	

* Note: when the CANTX alternate function is selected the IO port operates in output push-pull mode.

I/O PORTS (Cont'd)

5.1.4 Register Description

DATA REGISTER (DR)

Port x Data Register
PxDR with x = A, B, C, D, E or F.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *Data register 8 bits.*

The DR register has a specific behaviour according to the selected input/output configuration. Writing the DR register is always taken into account even if the pin is configured as an input; this allows to always have the expected level on the pin when toggling to output mode. 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 REGISTER (DDR)

Port x Data Direction Register
PxDDR with x = A, B, C, D, E or F.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = **DD[7:0]** *Data direction register 8 bits.*

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

0: Input mode
1: Output mode

OPTION REGISTER (OR)

Port x Option Register
PxOR with x = A, B, C, D, E or F.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
O7	O6	O5	O4	O3	O2	O1	O0

Bit 7:0 = **O[7:0]** *Option register 8 bits.*

For specific I/O pins, this register is not implemented. In this case the DDR register is enough to select the I/O pin configuration.

The OR register allows to distinguish: in input mode if the pull-up with interrupt capability or the basic pull-up configuration is selected, in output mode if the push-pull or open drain configuration is selected.

Each bit is set and cleared by software.

Input mode:
0: floating input
1: pull-up input with or without interrupt

Output mode:
0: output open drain (with P-Buffer unactivated)
1: output push-pull

I/O PORTS (Cont'd)

Table 12. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Reset Value of all IO port registers		0	0	0	0	0	0	0	0
0000h	**PADR**	MSB							LSB
0001h	**PADDR**								
0002h	**PAOR**								
0004h	**PCDR**	MSB							LSB
0005h	**PCDDR**								
0006h	**PCOR**								
0008h	**PBDR**	MSB							LSB
0009h	**PBDDR**								
000Ah	**PBOR**								
000Ch	**PEDR**	MSB							LSB
000Dh	**PEDDR**								
000Eh	**PEOR**								
0010h	**PDDR**	MSB							LSB
0011h	**PDDDR**								
0012h	**PDOR**								
0014h	**PFDR**	MSB							LSB
0015h	**PFDDR**								
0016h	**PFOR**								

5.2 MISCELLANEOUS REGISTERS

The miscellaneous registers allow control over several features such as the external interrupts or the I/Oalternate functions.

5.2.1 I/O Port Interrupt Sensitivity Description

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the Miscellaneous registers (Figure 29). This control allows to have up to 4 fully independent external interrupt source sensitivities.

Each external interrupt source can be generated on four (or five) different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level
- Rising edge and high level (only for EI0 and EI2)

To guarantee correct functionality, the sensitivity bits in the MISCR registers must be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3). See I/O port register and Miscellaneous register descriptions for more details on the programming.

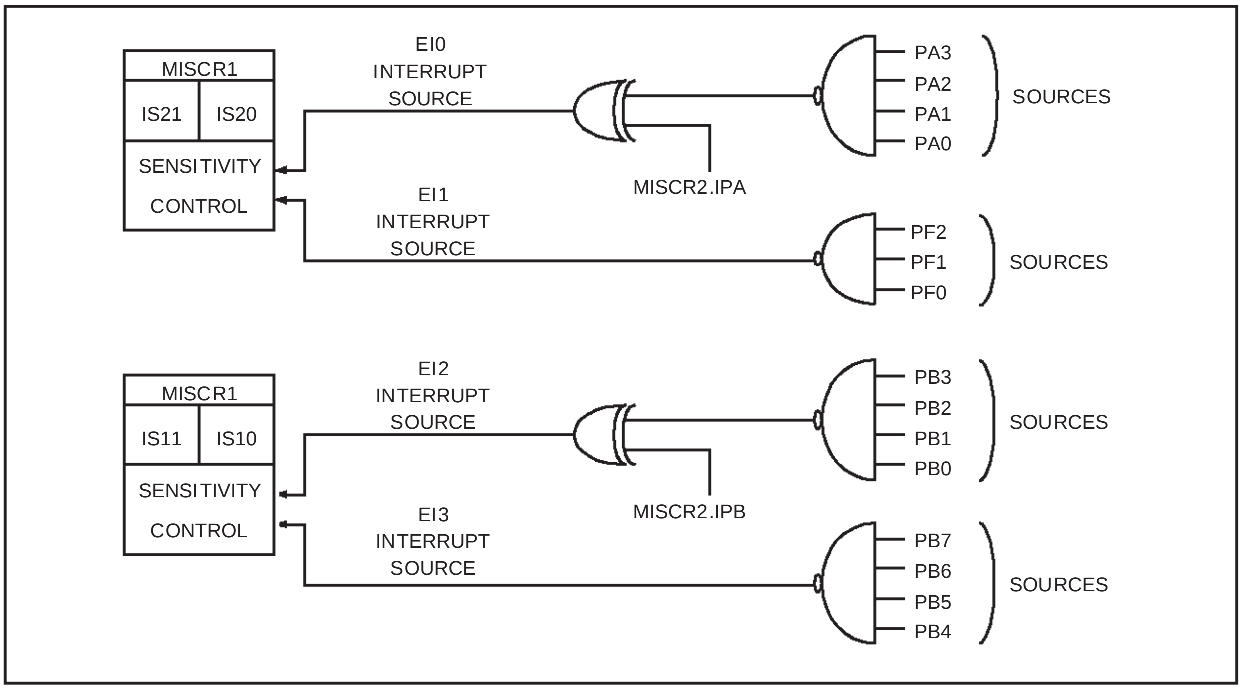
5.2.2 I/O Port Alternate Functions

The MISCR registers allow to manage four I/O port miscellaneous alternate functions:

- Main clock signal ($f_{OSC}/2$) output on PF0
- A Beep signal output on PF1 (with three selectable audio frequencies)
- A NMI management on a dedicated pin
- A SPI $\overline{SS}$ pin internal control to use the PC7 I/O port function while the SPI is active.

These functions are described in details in the Section 5.2.3 Miscellaneous Registers Description.

Figure 29. External Interrupt Sources vs MISCR



MISCELLANEOUS REGISTERS (Cont'd)

5.2.3 Miscellaneous Registers Description

MISCELLANEOUS REGISTER 1 (MISCR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IS11	IS10	MCO	IS21	IS20	CP1	CP0	SMS

Bit 7:6 = **IS1[1:0]** *EI2 and EI3 sensitivity*
The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts:
- EI2 (port B3..0)

IS11	IS10	External Interrupt Sensitivity	
		MISCR2.IPB=0	MISCR2.IPB=1
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- EI3 (port B7..4)

IS11	IS10	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 5 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software.
0: MCO alternate function disabled (I/O pin free for general-purpose I/O)
1: MCO alternate function enabled (f_{OSC}/2on I/O port)

Note: To reduce power consumption, the MCO function is not active in ACTIVE-HALT mode.

Bit 4:3 = **IS2[1:0]** *EI0 and EI1 sensitivity*
The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts:
- EI0 (port A3..0)

IS21	IS20	External Interrupt Sensitivity	
		MISCR2.IPA=0	MISCR2.IPA=1
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

- EI1 (port F2..0)

IS21	IS20	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 2:1 = **CP[1:0]** *CPU clock prescaler*
These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f_{CPU} in SLOW mode	CP1	CP0
f_{OSC} / 4	0	0
f_{OSC} / 8	1	0
f_{OSC} / 16	0	1
f_{OSC} / 32	1	1

Bit 0 = **SMS** *Slow mode select*
This bit is set and cleared by software.
0: Normal mode. f_{CPU} = f_{OSC} / 2
1: Slow mode. f_{CPU} is given by CP1, CP0
See low power consumption mode and MCC chapters for more details.

MISCELLANEOUS REGISTERS (Cont'd)

MISCELLANEOUS REGISTER 2 (MISCR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IPA	IPB	BC1	BC0	NMIS	NMIE	SSM	SSI

Bit 7 = **IPA** *Interrupt polarity for port A*
This bit is used to invert the sensitivity of the port A [3:0] external interrupts. It is set and cleared by software.
0: No sensitivity inversion
1: Sensitivity inversion
See Section 5.2.1 I/O Port Interrupt Sensitivity Description and the description of the IS2x bits of the MISCR1 register for more details.

Bit 6 = **IPB** *Interrupt polarity for port B*
This bit is used to invert the sensitivity of the port B [3:0] external interrupts. It is set and cleared by software.
0: No sensitivity inversion
1: Sensitivity inversion
See Section 5.2.1 I/O Port Interrupt Sensitivity Description and the description of the IS1x bits of the MISCR1 register for more details.

Bit 5:4 = **BC[1:0]** *Beep control*
These 2 bits select the PF1 pin beep capability.

BC1	**BC0**	**Beep mode with f_{OSC}=16MHz**	
0	0	Off	
0	1	~2-KHz	Output Beep signal ~50% duty cycle
1	0	~1-KHz	
1	1	~500-Hz	

The beep output signal is available in ACTIVE-HALT mode but has to be disabled to reduce the consumption.

Bit 3 = **NMIS** *NMI sensitivity*
This bit allows to toggle the NMI edge sensitivity. It can be set and cleared by software only when NMIE bit is cleared.
0: Falling edge
1: Rising edge

Bit 2 = **NMIE** *NMI enable*
This bit allows to enable or disable the NMI capability on the dedicated pin. It is set and cleared by software.
0: NMI disabled
1: NMI enabled
Note: a parasitic interrupt can be generated when clearing the NMIE bit.

Bit 1 = **SSM** $\overline{SS}$ *mode selection*
This bit is set and cleared by software.
0: Normal mode - the level of the SPI $\overline{SS}$ signal is input from the external $\overline{SS}$ pin.
1: I/O mode (PC7), the level of the SPI $\overline{SS}$ signal is read from the SSI bit.

Bit 0 = **SSI** $\overline{SS}$ *internal mode*
This bit replaces pin $\overline{SS}$ of the SPI when bit SSM is set to 1. (see SPI description). It is set and cleared by software.

MISCELLANEOUS REGISTERS (Cont'd)

Table 13. Miscellaneous Register Map and Reset Values

Address (Hex.)	**Register Label**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0020h	**MISCR1** Reset Value	IS11 0	IS10 0	MCO 0	IS21 0	IS20 0	CP1 0	CP0 0	SMS 0
0040h	**MISCR2** Reset Value	IPA 0	IPB 0	BC1 0	BC0 0	NMIS 0	NMIE 0	SSM 0	SSI 0

5.3 WATCHDOG TIMER (WDG)

5.3.1 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.

5.3.2 Main Features

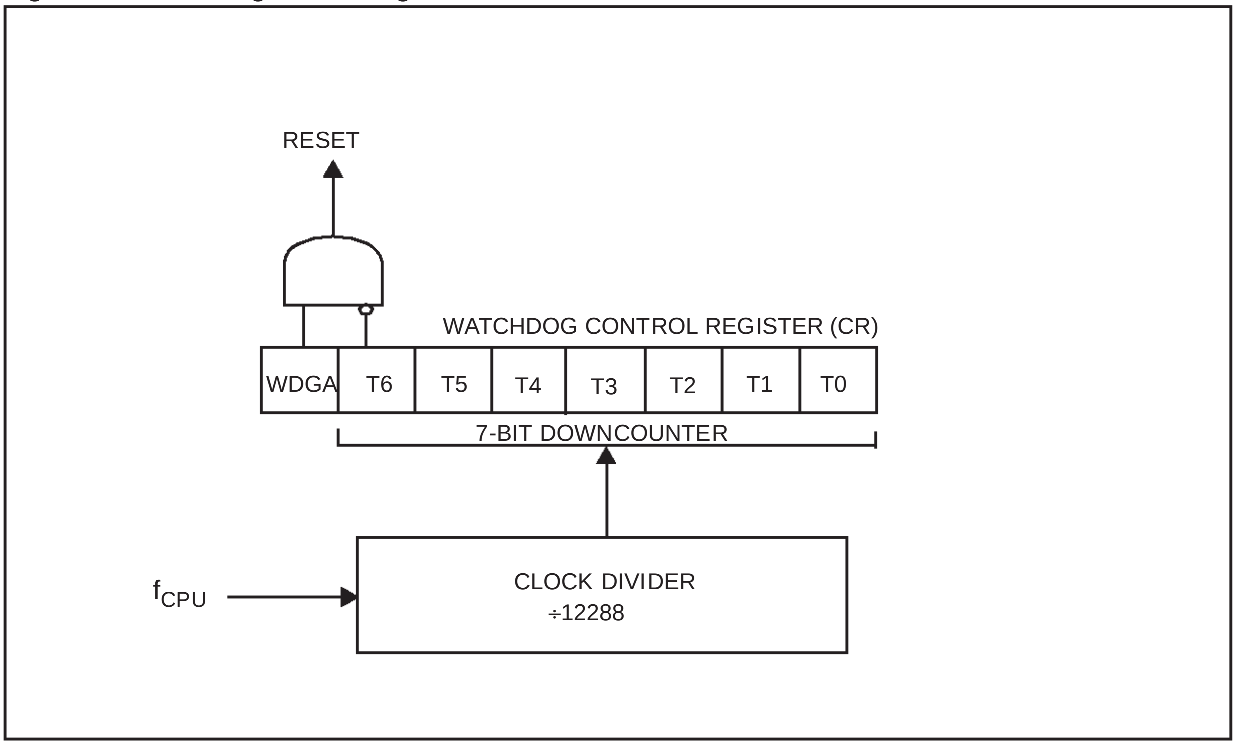
- Programmable timer (64 increments of 12288 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) after a HALT instruction or when the T6 bit reaches zero
- Hardware Watchdog selectable by option byte
- Watchdog Reset indicated by status flag (in versions with Safe Reset option only)

5.3.3 Functional Description

The counter value stored in the CR register (bits T[6:0]), is decremented every 12,288 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 T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

Figure 30. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 14 .Watchdog Timing (fCPU = 8 MHz)):

– The WDGA bit is set (watchdog enabled)

– The T6 bit is set to prevent generating an immediate reset

– The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 14.Watchdog Timing (f_{CPU} = 8 MHz)

	CR Register initial value	**WDG timeout period (ms)**
Max	FFh	98.304
Min	C0h	1.536

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

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.

5.3.4 Hardware Watchdog Option

If Hardware Watchdog Is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the device-specific Option Byte description.

5.3.5 Low Power Modes

Mode	**Description**
WAIT	No effect on Watchdog.
HALT	Immediate reset generation as soon as the HALT instruction is executed if the Watchdog is activated (WDGA bit is set).

5.3.6 Interrupts

None.

5.3.7 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

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

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 becomes cleared).

STATUS REGISTER (SR)

Read/Write

Reset Value*: 0000 0000 (00h)

7							0
-	-	-	-	-	-	-	WDOGF

Bit 0 = **WDOGF** *Watchdog flag*.
This bit is set by a watchdog reset and cleared by software or a power on/off reset. This bit is useful for distinguishing power/on off or external reset and watchdog reset.
0: No Watchdog reset occurred
1: Watchdog reset occurred

* Only by software and power on/off reset

Note: This register is not used in versions without LVD Reset.

WATCHDOG TIMER (Cond't)

Table 15. Watchdog Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Ah	**WDGCR** Reset Value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1
002Bh	**WDGSR** Reset Value	- 0	- 0	- 0	- 0	- 0	- 0	- 0	WDOGF 0

5.4 PWM AUTO-RELOAD TIMER (ART)

5.4.1 Introduction

The Pulse Width Modulated Auto-Reload Timer on-chip peripheral consists of an 8-bit auto reload counter with compare capabilities and of a 7-bit prescaler clock source.

These resources allow three possible operating modes:

– Generation of up to 4 independent PWM signals

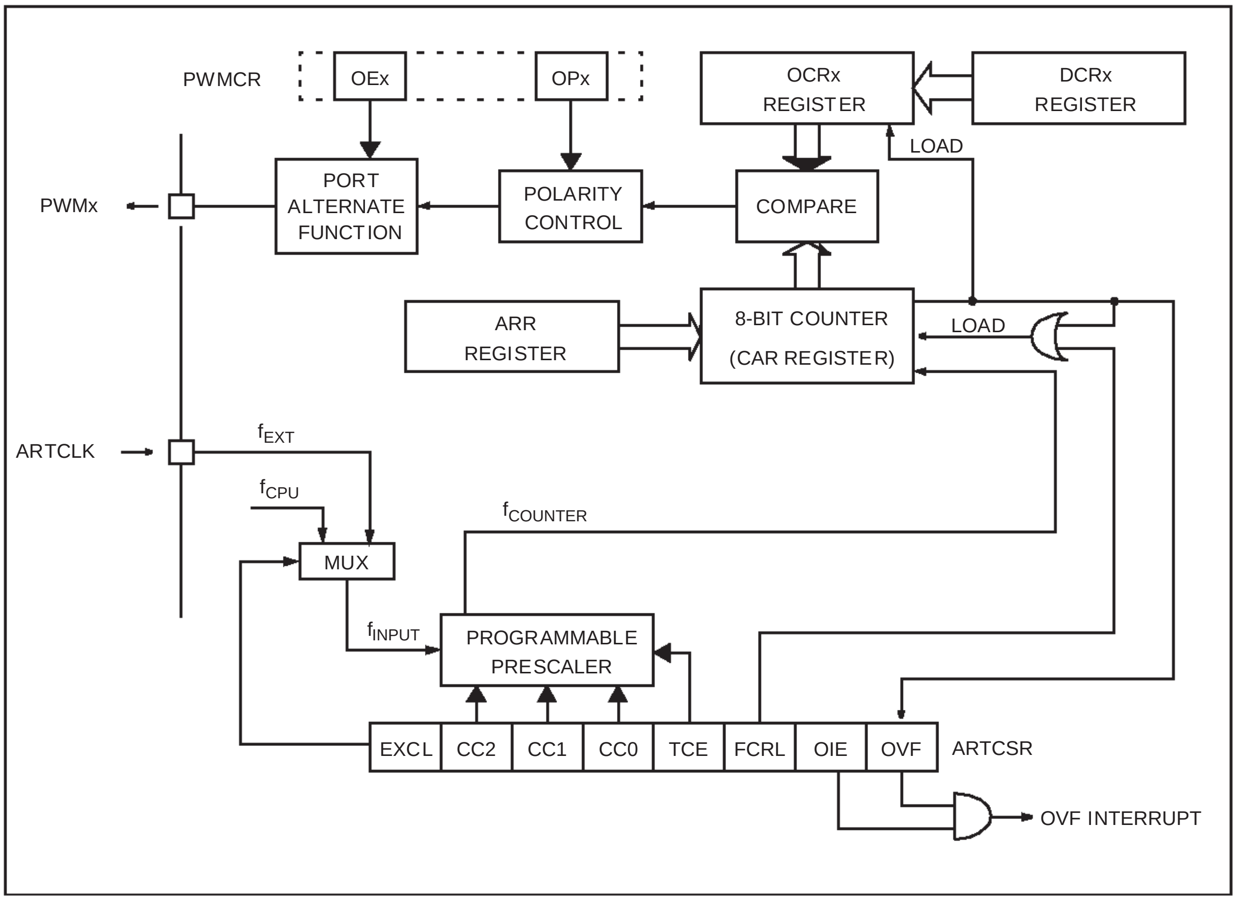
– Output compare and Time base interrupt

– External event detector

The two first modes can be used together with a single counter frequency.

The timer can be used to wake up the MCU from WAIT and HALT modes.

Figure 31. PWM Auto-Reload Timer Block Diagram



PWM AUTO-RELOAD TIMER (Cont'd)

5.4.2 Functional Description

Counter

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

It is possible to read or write the contents of the counter on the fly by reading or writing the Counter Access register (CAR).

When a counter overflow occurs, the counter is automatically reloaded with the contents of the ARR register (the prescaler is not affected).

Counter clock and prescaler

The counter clock frequency is given by:

$$f_{COUNTER} = f_{INPUT} / 2^{CC[2:0]}$$

The timer counter's input clock (f_{INPUT}) feeds the 7-bit programmable prescaler, which selects one of the 8 available taps of the prescaler, as defined by CC[2:0] bits in the Control/Status Register (CSR). Thus the division factor of the prescaler can be set to 2^n (where n = 0, 1,..7).

This f_{INPUT} frequency source is selected through the EXCL bit of the CSR register and can be either the f_{CPU} or an external input frequency f_{EXT}.

The clock input to the counter is enabled by the TCE (Timer Counter Enable) bit in the CSR register. When TCE is reset, the counter is stopped and the prescaler and counter contents are frozen. When TCE is set, the counter runs at the rate of the selected clock source.

Counter and Prescaler Initialization

After RESET, the counter and the prescaler are cleared and $f_{INPUT} = f_{CPU}$.

The counter can be initialized by:

– Writing to the ARR register and then setting the FCRL (Force Counter Re-Load) and the TCE (Timer Counter Enable) bits in the CSR register.

– Writing to the CAR counter access register,

In both cases the 7-bit prescaler is also cleared, whereupon counting will start from a known value.

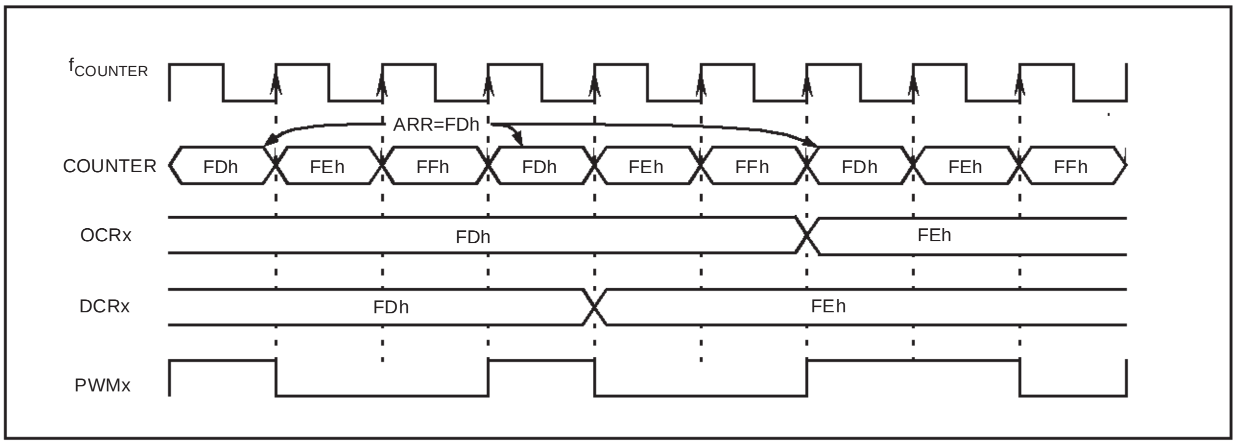
Direct access to the prescaler is not possible.

Output compare control

The timer compare function is based on four different comparisons with the counter (one for each PWMx output). Each comparison is made between the counter value and an output compare register (OCRx) value. This OCRx register can not be accessed directly, it is loaded from the duty cycle register (DCRx) at each overflow of the counter.

This double buffering method avoids glitch generation when changing the duty cycle on the fly.

Figure 32. Output compare control



PWM AUTO-RELOAD TIMER (Cont'd)

Independent PWM signal generation

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins with minimum core processing overhead. This function is stopped during HALT mode.

Each PWMx output signal can be selected independently using the corresponding OEx bit in the PWM Control register (PWMCR). When this bit is set, the corresponding I/O pin is configured as output push-pull alternate function.

The PWM signals all have the same frequency which is controlled by the counter period and the ARR register value.

$$f_{PWM} = f_{COUNTER} / (256 - ARR)$$

When a counter overflow occurs, the PWMx pin level is changed depending on the corresponding OPx (output polarity) bit in the PWMCR register. When the counter reaches the value contained in one of the output compare register (OCRx) the corresponding PWMx pin level is restored.

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of the PWM output signal. To obtain a signal on a PWMx pin, the contents of the OCRx register must be greater than the contents of the ARR register.

The maximum available resolution for the PWMx duty cycle is:

$$Resolution = 1 / (256 - ARR)$$

Note: To get the maximum resolution (1/256), the ARR register must be 0. With this maximum resolution, 0% and 100% can be obtained by changing the polarity.

Figure 33. PWM Auto-reload Timer Function

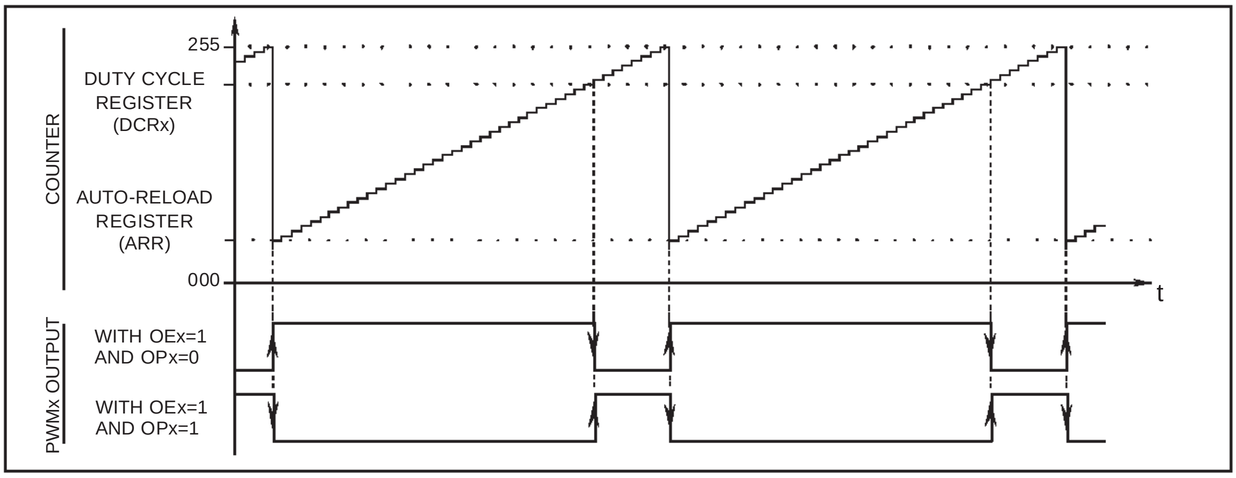
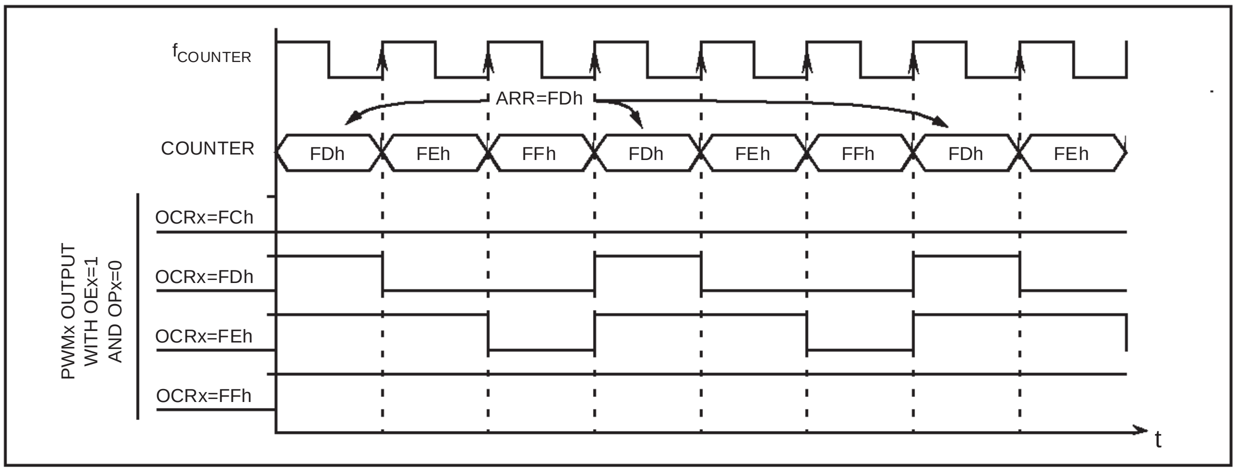


Figure 34. PWM Signal from 0% to 100% Duty Cycle



PWM AUTO-RELOAD TIMER (Cont'd)

Output compare and Time base interrupt

On overflow, the OVF flag of the CSR register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OIE, in the CSR register, is set. The OVF flag must be reset by the user software. This interrupt can be used as a time base in the application.

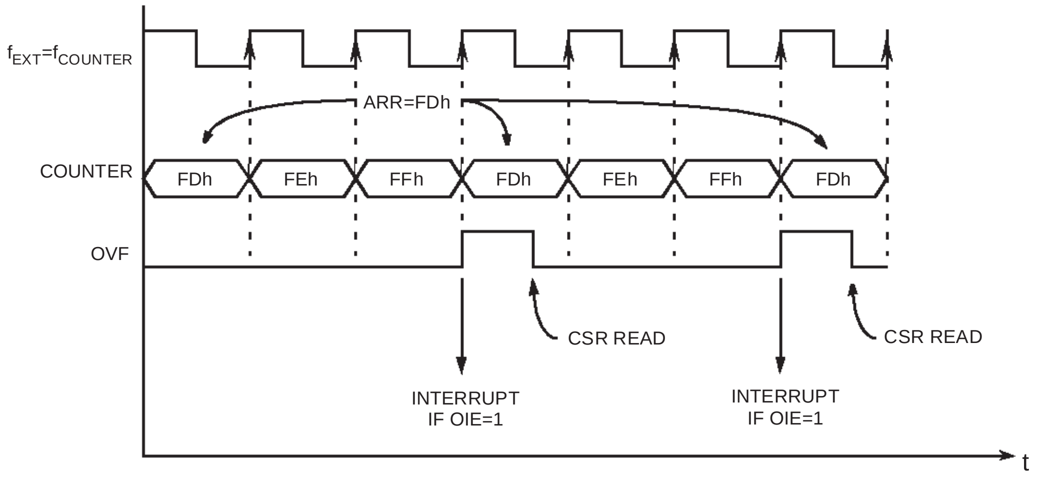
External clock and event detector mode

Using the f_{EXT} external prescaler input clock, the auto-reload timer can be used as an external clock event detector. In this mode, the ARR register is used to select the n_{EVENT} number of events to be counted before setting the OVF flag.

$$n_{EVENT} = 256 - ARR$$

When entering HALT mode while f_{EXT} is selected, all the timer control registers are frozen but the counter continues to increment. If the OIE bit is set, the next overflow of the counter will generate an interrupt which wakes up the MCU.

Figure 35. External Event Detector Example (3 counts)



PWM AUTO-RELOAD TIMER (Cont'd)

5.4.3 Register Description

CONTROL / STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
EXCL	CC2	CC1	CC0	TCE	FCRL	OIE	OVF

Bit 7 = **EXCL** *External Clock*
This bit is set and cleared by software. It selects the input clock for the 7-bit prescaler.
0: CPU clock.
1: External clock.

Bit 6:4 = **CC[2:0]** *Counter Clock Control*
These bits are set and cleared by software. They determine the prescaler division ratio from f_{INPUT}.

$f_{COUNTER}$	With f_{INPUT}=8 MHz	**CC2**	**CC1**	**CC0**
f_{INPUT}	8 MHz	0	0	0
f_{INPUT} / 2	4 MHz	0	0	1
f_{INPUT} / 4	2 MHz	0	1	0
f_{INPUT} / 8	1 MHz	0	1	1
f_{INPUT} / 16	500 KHz	1	0	0
f_{INPUT} / 32	250 KHz	1	0	1
f_{INPUT} / 64	125 KHz	1	1	0
f_{INPUT} / 128	62.5 KHz	1	1	1

Bit 3 = **TCE** *Timer Counter Enable*
This bit is set and cleared by software. It puts the timer in the lowest power consumption mode.
0: Counter stopped (prescaler and counter frozen).
1: Counter running.

Bit 2 = **FCRL** *Force Counter Re-Load*
This bit is write-only and any attempt to read it will yield a logical zero. When set, it causes the contents of ARR register to be loaded into the counter, and the content of the prescaler register to be cleared in order to initialize the timer before starting to count.

Bit 1 = **OIE** *Overflow Interrupt Enable*
This bit is set and cleared by software. It allows to enable/disable the interrupt which is generated when the OVF bit is set.
0: Overflow Interrupt disable.
1: Overflow Interrupt enable.

Bit 0 = **OVF** *Overflow Flag*
This bit is set by hardware and cleared by software reading the CSR register. It indicates the transition of the counter from FFh to the ARR value.
0: New transition not yet reached
1: Transition reached

COUNTER ACCESS REGISTER (CAR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

Bit 7:0 = **CA[7:0]** *Counter Access Data*

These bits can be set and cleared either by hardware or by software. The CAR register is used to read or write the auto-reload counter "on the fly" (while it is counting).

AUTO-RELOAD REGISTER (ARR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bit 7:0 = **AR[7:0]** *Counter Auto-Reload Data*

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register has two PWM management functions:

- Adjusting the PWM frequency
- Setting the PWM duty cycle resolution

PWM Frequency vs. Resolution:

ARR value	**Resolution**	**f_{PWM}**	
		Min	**Max**
0	8-bit	~0.244-KHz	31.25-KHz
[0..127]	> 7-bit	~0.244-KHz	62.5-KHz
[128..191]	> 6-bit	~0.488-KHz	125-KHz
[192..223]	> 5-bit	~0.977-KHz	250-KHz
[224..239]	> 4-bit	~1.953-KHz	500-KHz

PWM AUTO-RELOAD TIMER (Cont'd)

PWM CONTROL REGISTER (PWMCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0

Bit 7:4 = **OE[3:0]** *PWM Output Enable*
These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin.
0: PWM output disabled.
1: PWM output enabled.

Bit 3:0 = **OP[3:0]** *PWM Output Polarity*
These bits are set and cleared by software. They independently select the polarity of the four PWM output signals.

PWMx output level		OPx
Counter <= OCRx	**Counter > OCRx**	
1	0	0
0	1	1

Note: When an OPx bit is modified, the PWMx output signal polarity is immediately reversed.

DUTY CYCLE REGISTERS (DCRx)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

Bit 7:0 = **DC[7:0]** *Duty Cycle Data*

These bits are set and cleared by software.

A DCRx register is associated with the OCRx register of each PWM channel to determine the second edge location of the PWM signal (the first edge location is common to all channels and given by the ARR register). These DCR registers allow the duty cycle to be set independently for each PWM channel.

PWM AUTO-RELOAD TIMER (Cont'd)

Table 16. PWM Auto-Reload Timer Register Map and Reset Values

Address (Hex.)	**Register Label**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0072h	**PWMDCR3** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0073h	**PWMDCR2** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0074h	**PWMDCR1** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0075h	**PWMDCR0** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0076h	**PWMCR** Reset Value	OE3 0	OE2 0	OE1 0	OE0 0	OP3 0	OP2 0	OP1 0	OP0 0
0077h	**ARTCSR** Reset Value	EXCL 0	CC2 0	CC1 0	CC0 0	TCE 0	FCRL 0	OIE 0	OVF 0
0078h	**ARTCAR** Reset Value	CA7 0	CA6 0	CA5 0	CA4 0	CA3 0	CA2 0	CA1 0	CA0 0
0079h	**ARTARR** Reset Value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0

5.5 16-BIT TIMER

5.5.1 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.

5.5.2 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 (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 36.

***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'.

5.5.3 Functional Description

5.5.3.1 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).

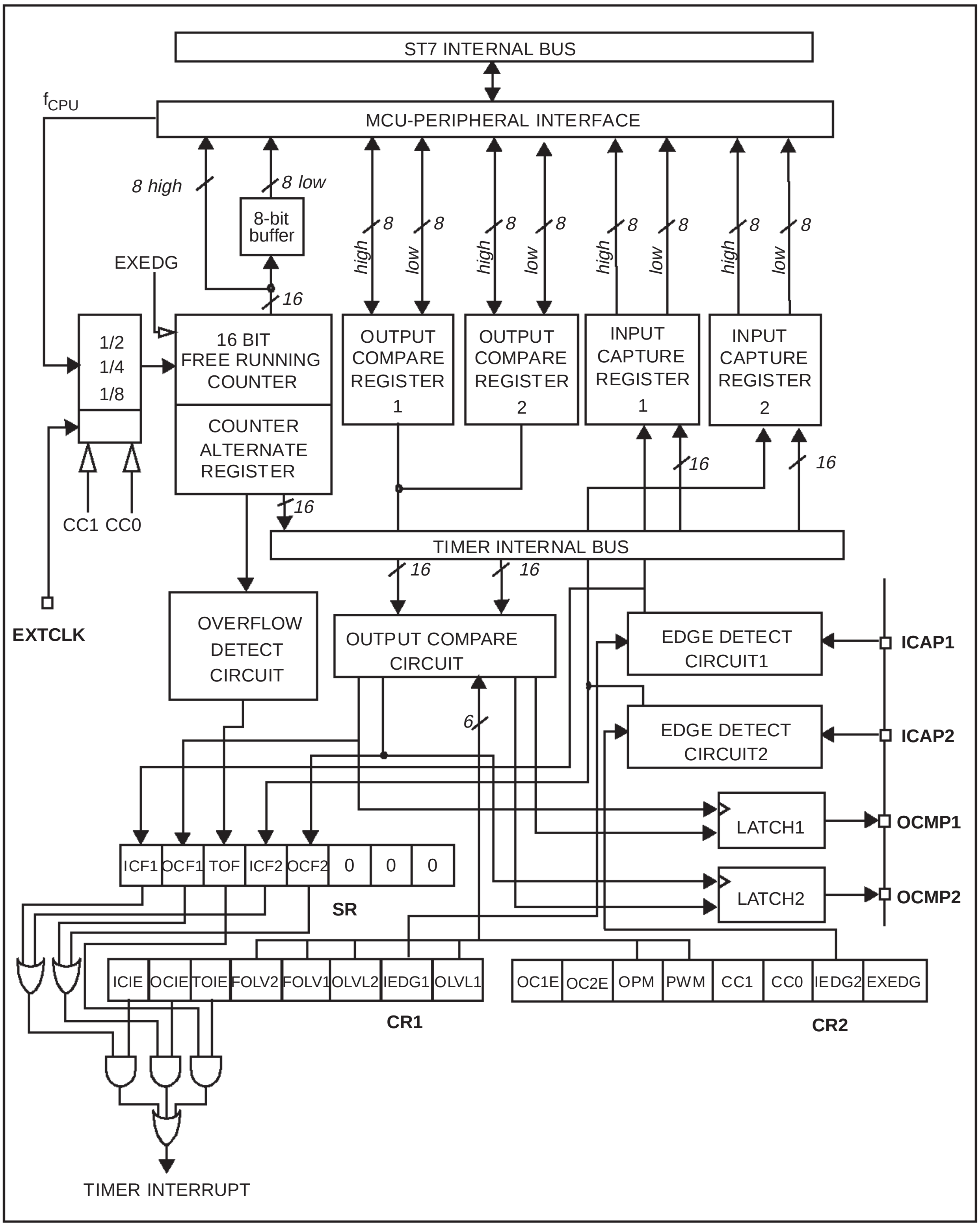
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (overflow flag), (see note at the end of paragraph titled 16-bit read sequence).

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 17 Clock Control Bits. The value in the counter register repeats every 131.072, 262.144 or 524.288 internal processorclock cycles depending on the CC1 and CC0 bits.

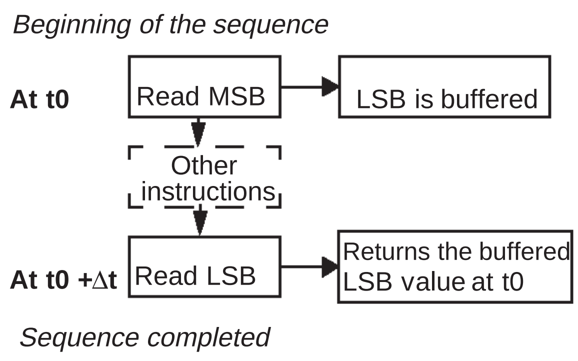
16-BIT TIMER (Cont'd)

Figure 36. Timer Block Diagram



16-BIT TIMER (Cont'd)

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



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.

Whatever the timer mode used (input capture, output compare, one pulse mode or PWM mode) 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).

5.5.3.2 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 37. Counter Timing Diagram, internal clock divided by 2

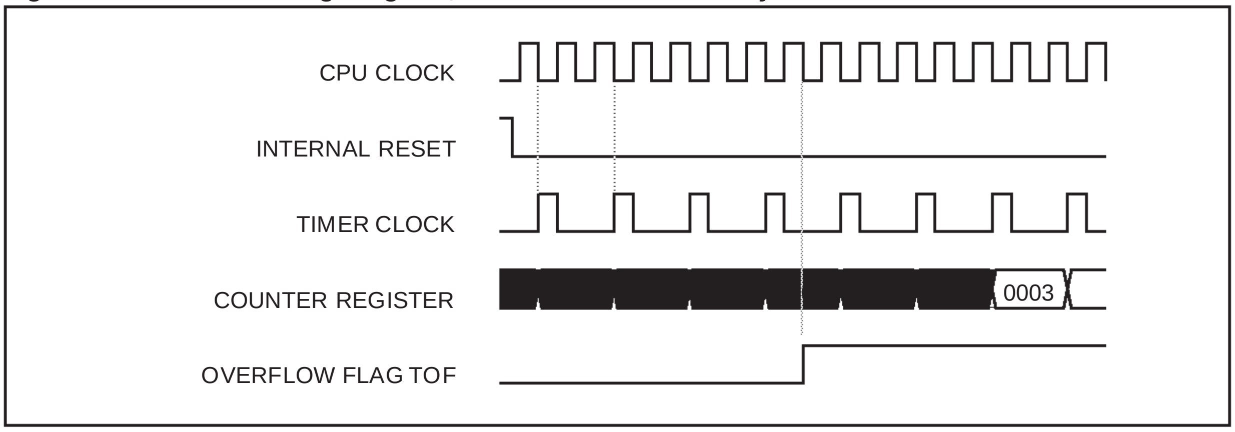


Figure 38. Counter Timing Diagram, internal clock divided by 4

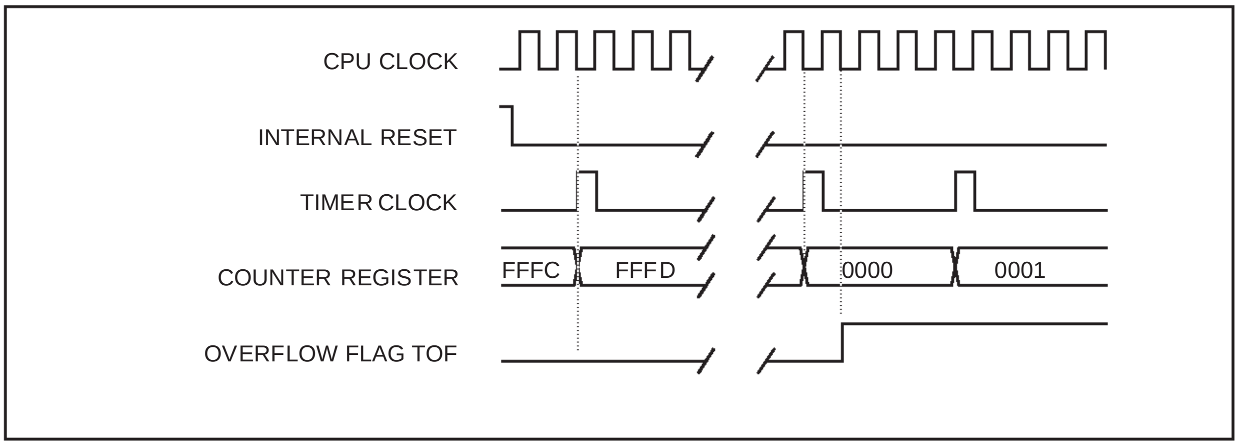
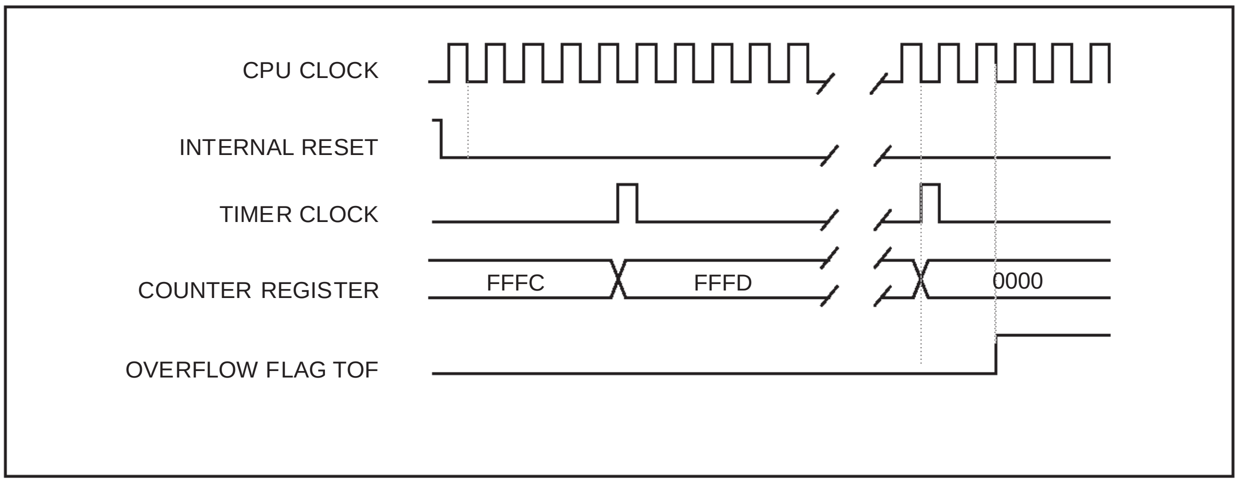


Figure 39. Counter Timing Diagram, internal clock divided by 8



16-BIT TIMER (Cont'd)

5.5.3.3 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).

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

IC*i* 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 17 Clock Control Bits).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input).

And select the following in the CR1 register:

– Set the ICIE bit to generate an interrupt after an input capture coming from both the ICAP1 pin or the ICAP2 pin

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input).

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 41).

– 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.

Notes:

1. After reading the IC*i*HR register, transfer of input capture data is inhibited until the IC*i*LR register is also read.
2. The IC*i*R register always contains the free running counter value which corresponds to the most recent input capture.
3. The 2 input capture functions can be used together even if the timer also uses the output compare mode.
4. In One pulse Mode and PWM mode only the input capture 2 can be used.
5. The alternate inputs (ICAP1 & ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture process.
6. Moreover if one of the ICAP*i* pin is configured as an input and the second one as an output, an interrupt can be generated if the user toggle the output pin and if the ICIE bit is set.
7. The TOF bit can be used with interrupt in order to measure event that go beyond the timer range (FFFFh).

16-BIT TIMER (Cont'd)

Figure 40. Input Capture Block Diagram

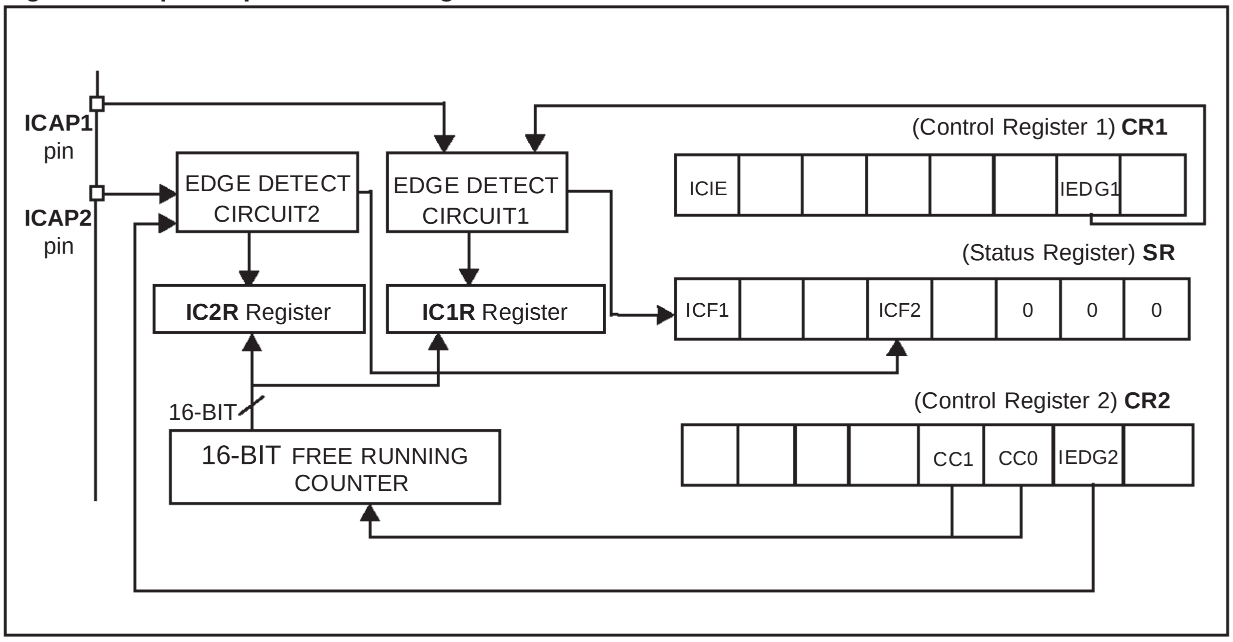
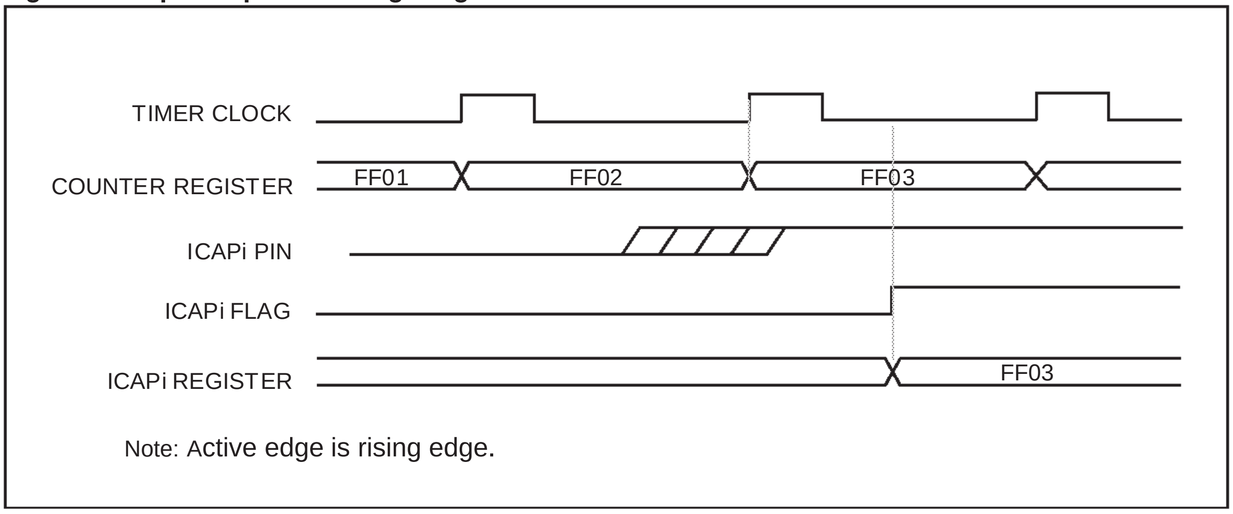


Figure 41. Input Capture Timing Diagram



16-BIT TIMER (Cont'd)

5.5.3.4 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.

	MS Byte	LS Byte
OC*i*R	OC*i*HR	OC*i*LR

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

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

Procedure:

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

– Set the OC*i*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 17 Clock Control Bits).

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 a 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).

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

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

Δt	=	Desired output compare period (in seconds)
f_{CPU}	=	Internal clock frequency
PRESC	=	Timer prescaler factor (2, 4 or 8 depending on CC1-CC0 bits, see Table 17 Clock Control Bits)

Clearing the output compare interrupt request is done by:

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

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*i*R register:

– Write to the OC*i*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*i*LR register (enables the output compare function and clears the OCF*i* bit).

Notes:

1. After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.
3. When the clock is divided by 2, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value (see Figure 43 on page 66). This behaviour is the same in OPM or PWM mode.
   When the clock is divided by 4, 8 or in external clock mode, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value plus 1 (see Figure 44 on page 66).
4. The output compare functions can be used both for generating external events on the OCMP*i* pins even if the input capture mode is also used.
5. The value in the 16-bit OC*i*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.

16-BIT TIMER (Cont'd)

Figure 42. Output Compare Block Diagram

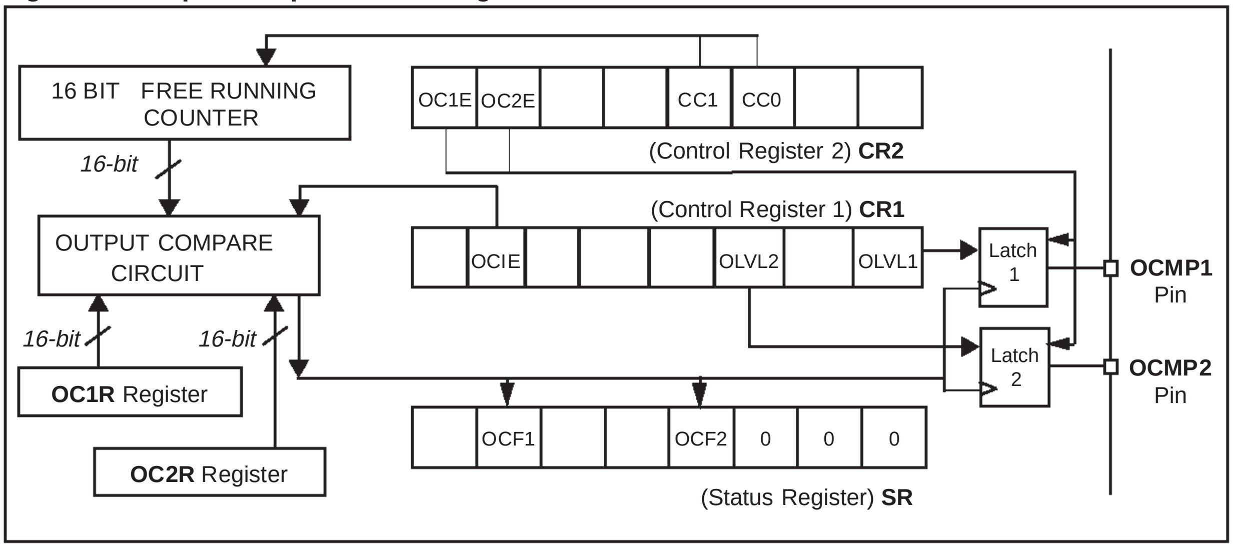


Figure 43. Output Compare Timing Diagram, Internal Clock Divided by 2

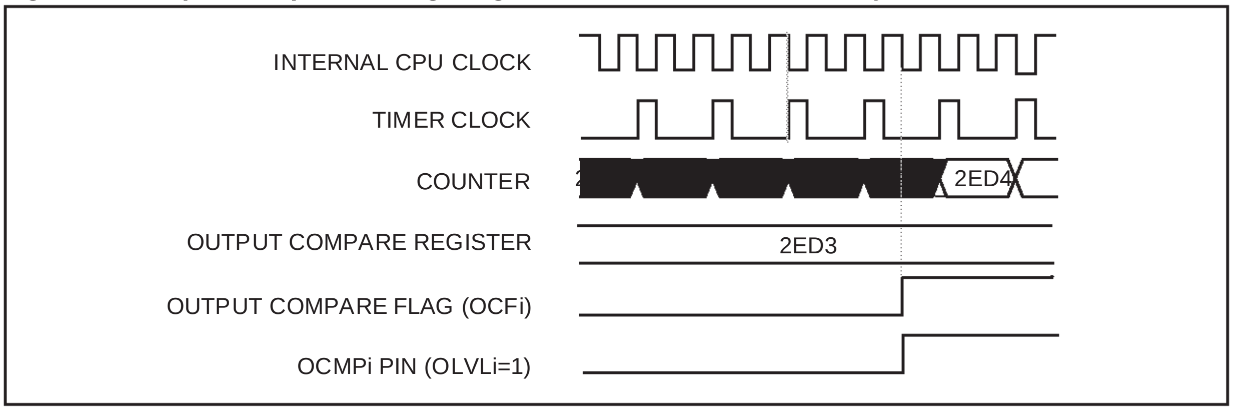
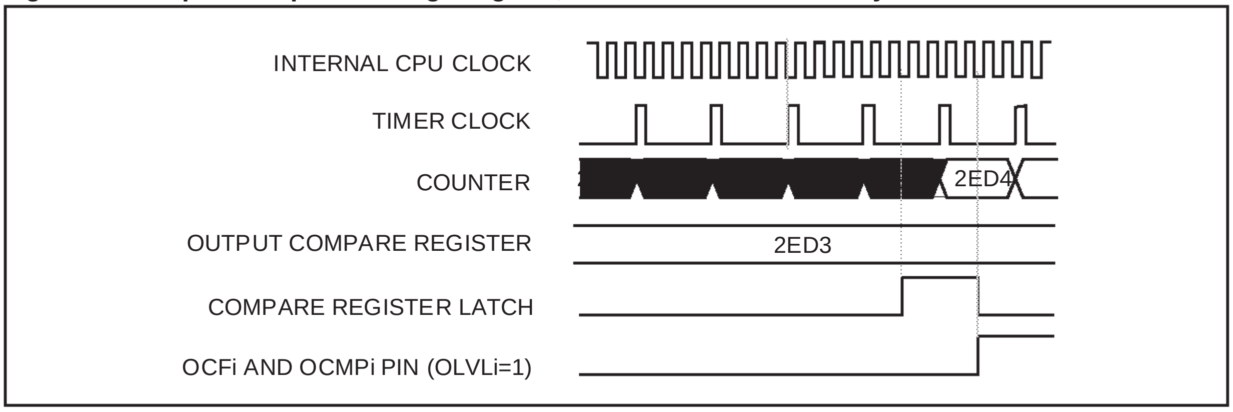


Figure 44. Output Compare Timing Diagram, Internal Clock Divided by 4



16-BIT TIMER (Cont'd)

5.5.3.5 Forced Compare

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

The following bits of the CR1 register are used:

			FOLV2	FOLV1	OLVL2		OLVL1

When the FOLV*i* bit is set by software, 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 (OC*i*E bit=1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

5.5.3.6 One Pulse Mode

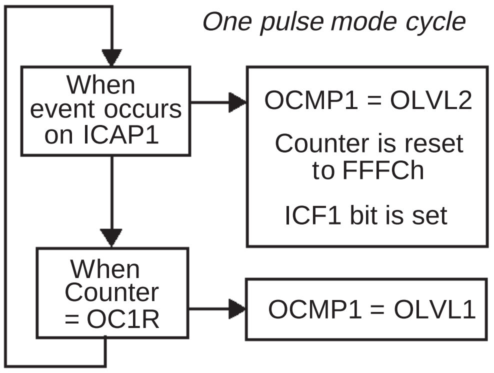
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

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 5.5.3.7).
2. Select the following in 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 (the ICAP1 pin must be configured as floating input).
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 17 Clock Control Bits).



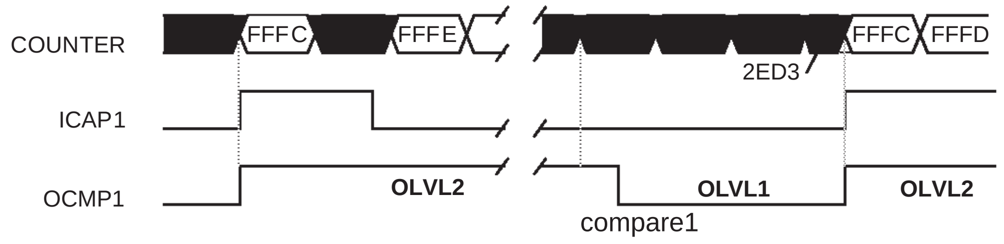
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

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 45).

Notes:

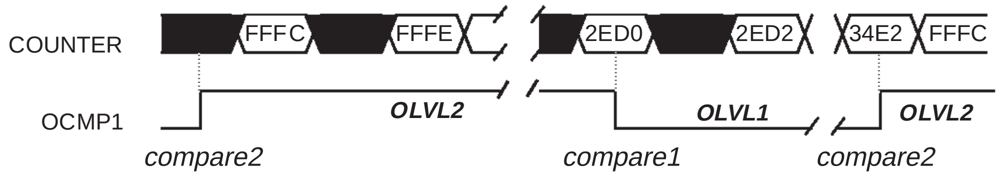
1. The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.
2. The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.
3. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.
4. If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.
5. The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.
6. When the one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 45. One Pulse Mode Timing Example



Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1

Figure 46. Pulse Width Modulation Mode Timing Example



Note: OC1R=2ED0h, OC2R=34E2, OLVL1=0, OLVL2= 1

16-BIT TIMER (Cont'd)

5.5.3.7 Pulse Width Modulation Mode

Pulse Width Modulation (PWM) 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, and so these functionality can not be used when the PWM mode is activated.

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 17 Clock Control Bits).

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

If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.

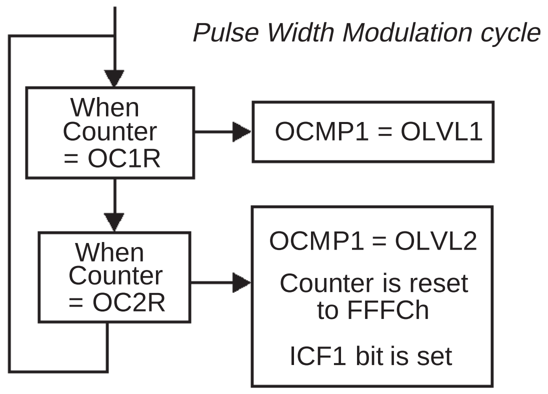
The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$OCiR\ Value = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t	=	Desired output compare period (in seconds)
f_{CPU}	=	Internal clock frequency
PRESC	=	Timer prescaler factor (2, 4 or 8 depending on CC1-CC0 bits, see Table 17 Clock Control Bits)

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 46).



Notes:

1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
   Therefore the Input Capture 1 function is inhibited but the Input Capture 2 is available.
2. The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.
3. 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.
4. In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected to the timer. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each period and ICF1 can also generates interrupt if ICIE is set.
5. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

16-BIT TIMER (Cont'd)

5.5.4 Low Power Modes

Mode	Description
WAIT	No effect on 16-bit Timer. Timer interrupts cause the device to exit from WAIT mode.
HALT	16-bit Timer registers are frozen. In HALT mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the MCU is woken up by an interrupt with "exit from HALT mode" capability or from the counter reset value when the MCU is woken up by a RESET. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the MCU is woken up by an interrupt with "exit from HALT mode" capability, the ICF*i* bit is set, and the counter value present when exiting from HALT mode is captured into the IC*i*R register.

5.5.5 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Input Capture 1 event/Counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2		Yes	No
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE	Yes	No
Output Compare 2 event (not available in PWM mode)	OCF2		Yes	No
Timer Overflow event	TOF	TOIE	Yes	No

Note: The 16-bit Timer interrupt events are connected to the same interrupt vector (see Interrupts chapter). These events generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

16-BIT TIMER (Cont'd)

5.5.6 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, if the OC2E bit is set and even if there is no successful comparison.

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, if the OC1E bit is set and even if there is no successful comparison.

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 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the Output Compare 1 function of the timer remains active.
0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP1 pin alternate function enabled.

Bit 6 = **OC2E** *Output Compare 2 Enable.*
This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the Output Compare 2 function of the timer remains active.
0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP2 pin alternate function enabled.

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 17. 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.

7							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).

7							0
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).

7							0
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.

7							0
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.

7							0
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.

7							0
MSB							LSB

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.

7							0
MSB							LSB

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.

7							0
MSB							LSB

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.

7							0
MSB							LSB

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.

7							0
MSB							LSB

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.

7							0
MSB							LSB

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).

7							0
MSB							LSB

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).

7							0
MSB							LSB

16-BIT TIMER (Cont'd)

Table 18. 16-Bit Timer Register Map and Reset Values

Address (Hex.)	**Register Label**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Timer A: 32 Timer B: 42	**CR1** Reset Value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer A: 31 Timer B: 41	**CR2** Reset Value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer A: 33 Timer B: 43	**SR** Reset Value	ICF1 0	OCF1 0	TOF 0	ICF2 0	OCF2 0	- 0	- 0	- 0
Timer A: 34 Timer B: 44	**ICHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 35 Timer B: 45	**ICLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 36 Timer B: 46	**OCHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 37 Timer B: 47	**OCLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3E Timer B: 4E	**OCHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3F Timer B: 4F	**OCLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 38 Timer B: 48	**CHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 39 Timer B: 49	**CLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3A Timer B: 4A	**ACHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 3B Timer B: 4B	**ACLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3C Timer B: 4C	**ICHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3D Timer B: 4D	**ICLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -

5.6 SERIAL PERIPHERAL INTERFACE (SPI)

5.6.1 Introduction

The Serial Peripheral Interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves or a system in which devices may be either masters or slaves.

The SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller.

Refer to the Pin Description chapter for the device-specific pin-out.

5.6.2 Main Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- Four master mode frequencies
- Maximum slave mode frequency = fCPU/2.
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability.

5.6.3 General description

The SPI is connected to external devices through 4 alternate pins:

– MISO: Master In Slave Out pin

– MOSI: Master Out Slave In pin

– SCK: Serial Clock pin

– $\overline{SS}$: Slave select pin

A basic example of interconnections between a single master and a single slave is illustrated on Figure 47.

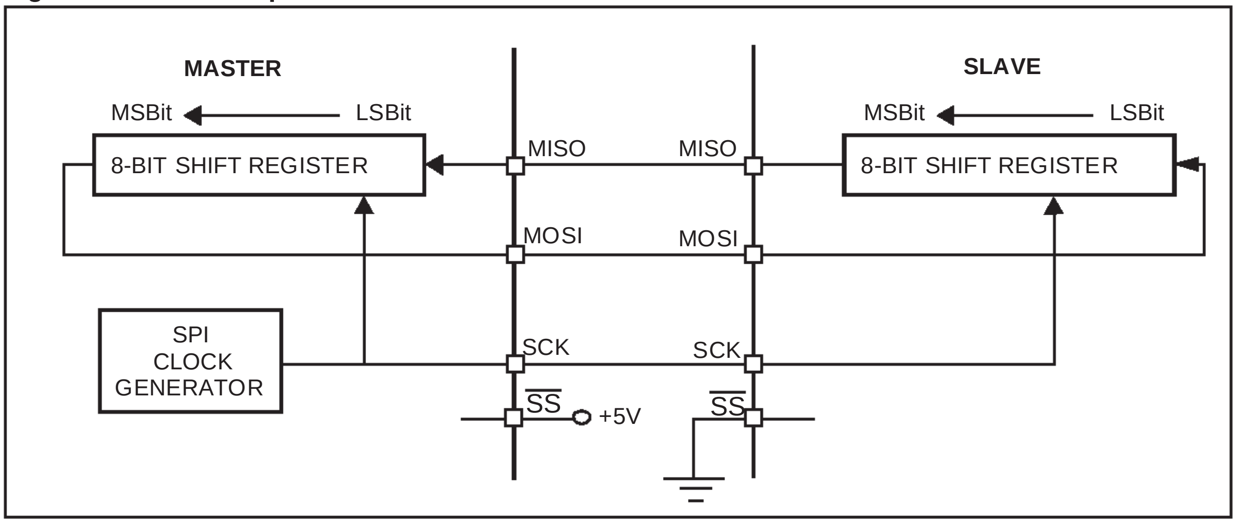
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full bits. A status flag is used to indicate that the I/O operation is complete.

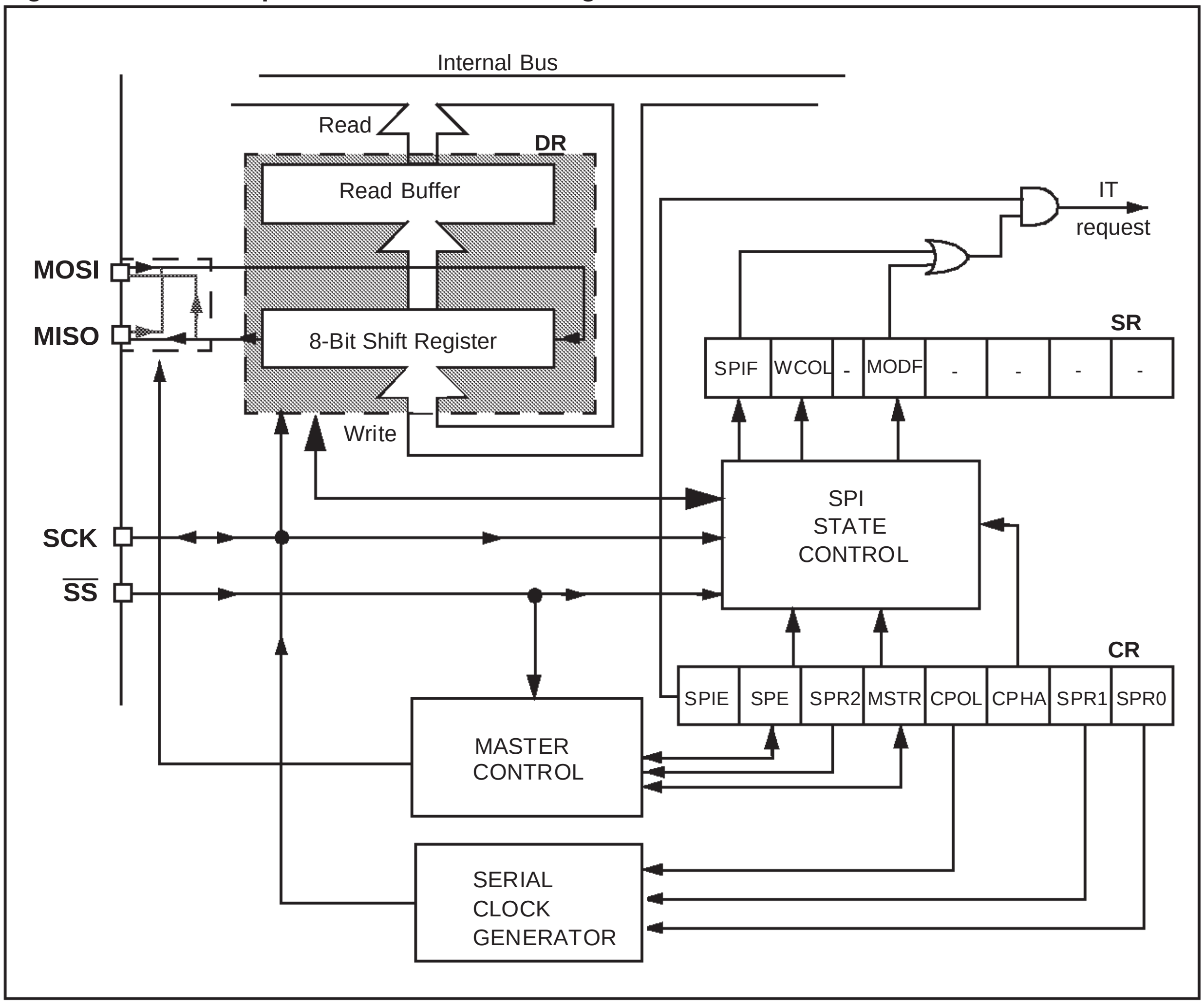
Four possible data/clock timing relationships may be chosen (see Figure 50) but master and slave must be programmed with the same timing mode.

Figure 47. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 48. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4 Functional Description

Figure 47 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

– A Control Register (CR)

– A Status Register (SR)

– A Data Register (DR)

Refer to the CR, SR and DR registers in Section 5.6.7for the bit definitions.

5.6.4.1 Master Configuration

In a master configuration, the serial clock is generated on the SCK pin.

Procedure

– Select the SPR0 & SPR1 bits to define the serial clock baud rate (see CR register).

– Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 50).

– The $\overline{\text{SS}}$ pin must be connected to a high level signal during the complete byte transmit sequence.

– The MSTR and SPE bits must be set (they remain set only if the $\overline{\text{SS}}$ pin is connected to a high level signal).

In this configuration the MOSI pin is a data output and to the MISO pin is a data input.

Transmit sequence

The transmit sequence begins when a byte is written the DR register.

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

– The SPIF bit is set by hardware

– An interrupt is generated if the SPIE bit is set and the I bit in the CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set
2. A write or a read of the DR register.

Note: While the SPIF bit is set, all writes to the DR

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4.2 Slave Configuration

In slave configuration, the serial clock is received on the SCK pin from the master device.

The value of the SPR0 & SPR1 bits is not used for the data transfer.

Procedure

– For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 50.

– The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence.

– Clear the MSTR bit and set the SPE bit to assign the pins to alternate function.

In this configuration the MOSI pin is a data input and the MISO pin is a data output.

Transmit Sequence

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

– The SPIF bit is set by hardware

– An interrupt is generated if SPIE bit is set and I bit in CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set.
2. A write or a read of the DR register.

Notes: While the SPIF bit is set, all writes to the DR register are inhibited until the SR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an overrun condition (see Section 5.6.4.6).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the DR register between each data byte transfer to avoid a write collision (see Section 5.6.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4.3 Data Transfer Format

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The serial clock is used to synchronize the data transfer during a sequence of eight clock pulses.

The $\overline{SS}$ pin allows individual selection of a slave device; the other slave devices that are not selected do not interfere with the SPI transfer.

Clock Phase and Clock Polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits.

The CPOL (clock polarity) bit controls the steady state value of the clock when no data is being transferred. This bit affects both master and slave modes.

The combination between the CPOL and CPHA (clock phase) bits selects the data capture clock edge.

Figure 50, shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin, the MOSI pin are directly connected between the master and the slave device.

The $\overline{SS}$ pin is the slave device select input and can be driven by the master device.

The master device applies data to its MOSI pin-clock edge before the capture clock edge.

CPHA bit is set

The second edge on the SCK pin (falling edge if the CPOL bit is reset, rising edge if the CPOL bit is set) is the MSBit capture strobe. Data is latched on the occurrence of the first clock transition.

No write collision should occur even if the $\overline{SS}$ pin stays low during a transfer of several bytes (see Figure 49).

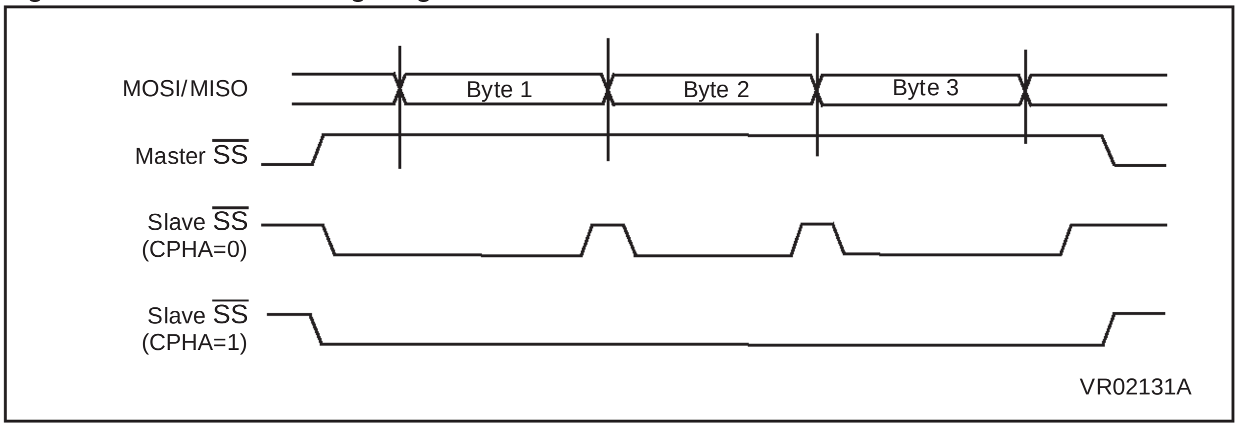
CPHA bit is reset

The first edge on the SCK pin (falling edge if CPOL bit is set, rising edge if CPOL bit is reset) is the MSBit capture strobe. Data is latched on the occurrence of the second clock transition.

This pin must be toggled high and low between each byte transmitted (see Figure 49).

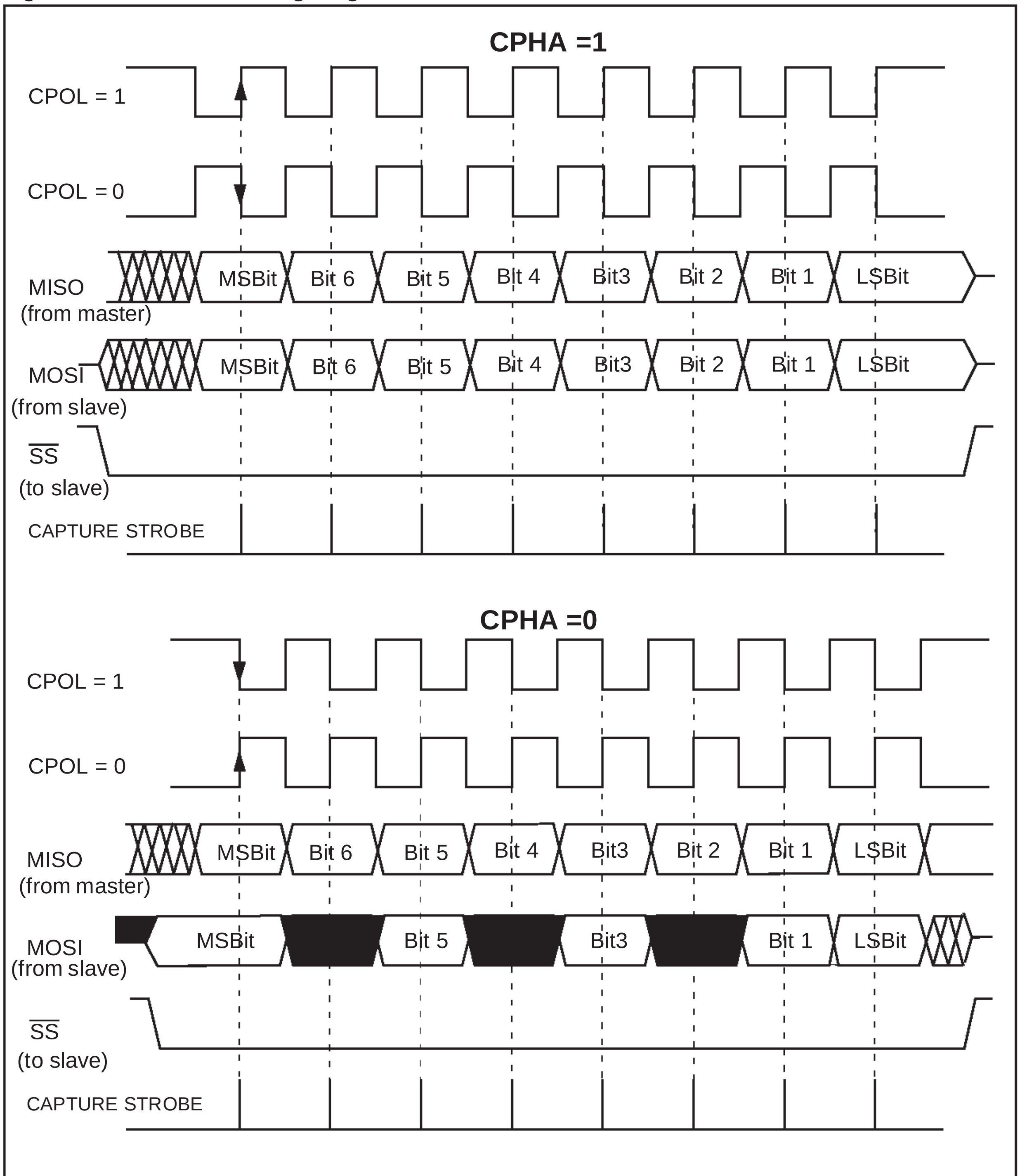
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its DR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the DR without producing a write collision.

Figure 49. CPHA / $\overline{SS}$ Timing Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 50. Data Clock Timing Diagram



Note: This figure should not be used as a replacement for parametric information.
Refer to the Electrical Characteristics chapter.

VR02131B

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4.4 Write Collision Error

A write collision occurs when the software tries to write to the DR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode.

Note: a "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation.

In Slave mode

When the CPHA bit is set:

The slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device DR register and output the MSBit on to the external MISO pin of the slave device.

The $\overline{SS}$ pin low state enables the slave device but the output of the MSBit onto the MISO pin does not take place until the first data transfer clock edge.

When the CPHA bit is reset:

Data is latched on the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when software attempts to write the DR register after its $\overline{SS}$ pin has been pulled low.

For this reason, the $\overline{SS}$ pin must be high, between each data byte transfer, to allow the CPU to write in the DR register without generating a write collision.

In Master mode

Collision in the master device is defined as a write of the DR register while the internal serial clock (SCK) is in the process of transfer.

The $\overline{SS}$ pin signal must be always high on the master device.

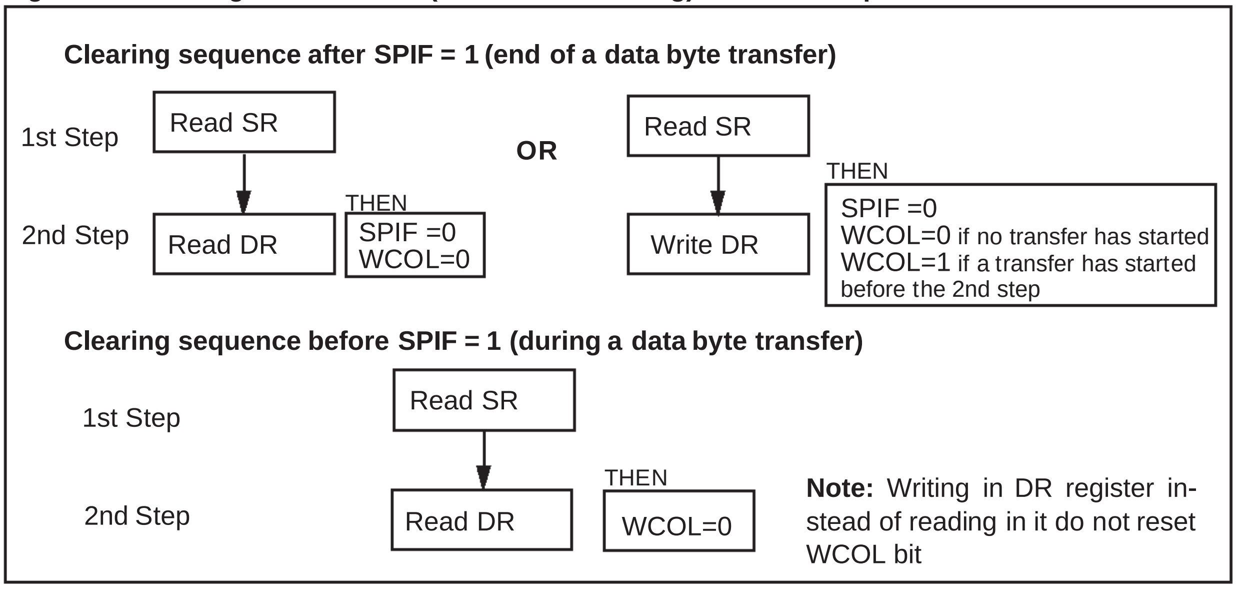
WCOL bit

The WCOL bit in the SR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see Figure 51).

Figure 51. Clearing the WCOL bit (Write Collision Flag) Software Sequence



Clearing sequence after SPIF = 1 (end of a data byte transfer)
1st Step
Read SR
2nd Step
Read DR
THEN
SPIF =0
WCOL=0
OR
Read SR
Write DR
THEN
SPIF =0
WCOL=0 if no transfer has started
WCOL=1 if a transfer has started before the 2nd step
Clearing sequence before SPIF = 1 (during a data byte transfer)
1st Step
Read SR
2nd Step
Read DR
THEN
WCOL=0
Note: Writing in DR register instead of reading in it do not reset WCOL bit

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4.5 Master Mode Fault

Master mode fault occurs when the master device has its $\overline{SS}$ pin pulled low, then the MODF bit is set.

Master mode fault affects the SPI peripheral in the following ways:

– The MODF bit is set and an SPI interrupt is generated if the SPIE bit is set.

– The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.

– The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read or write access to the SR register while the MODF bit is set.
2. A write to the CR register.

Notes: To avoid any multiple slave conflicts in the case of a system comprising several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of the MODF bit. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

In a slave device the MODF bit can not be set, but in a multi master configuration the device can be in slave mode with this MODF bit set.

The MODF bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state using an interrupt routine.

5.6.4.6 Overrun Condition

An overrun condition occurs, when the master device has sent several data bytes and the slave device has not cleared the SPIF bit issuing from the previous data byte transmitted.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the DR register returns this byte. All other bytes are lost.

This condition is not detected by the SPI peripheral.

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.4.7 Single Master and Multimaster Configurations

There are two types of SPI systems:

– Single Master System

– Multimaster System

Single Master System

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see Figure 52).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written its DR register.

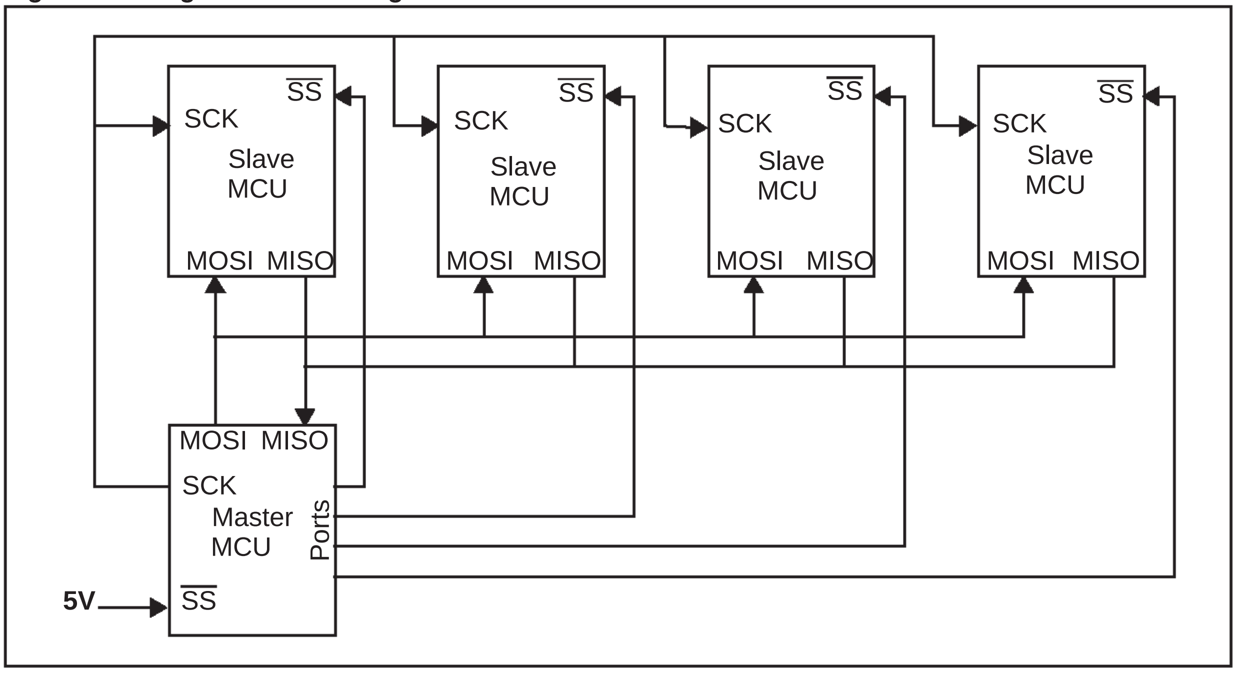
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multi-master System

A multi-master system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multi-master system is principally handled by the MSTR bit in the CR register and the MODF bit in the SR register.

Figure 52. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.5 Low Power Modes

Mode	Description
WAIT	No effect on SPI. SPI interrupt events cause the device to exit from WAIT mode.
HALT	SPI registers are frozen. In HALT mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

5.6.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	No
Master Mode Fault Event	MODF		Yes	No

Note: The SPI interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

SERIAL PERIPHERAL INTERFACE (Cont'd)

5.6.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0000xxxx (0xh)

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

Bit 7 = **SPIE** *Serial peripheral interrupt enable.*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SPI interrupt is generated whenever SPIF=1 or MODF=1 in the SR register

Bit 6 = **SPE** *Serial peripheral output enable.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 5.6.4.5 Master Mode Fault).
0: I/O port connected to pins
1: SPI alternate functions connected to pins

The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins.

Bit 5 = **SPR2** *Divider Enable*.

this bit is set and cleared by software and it is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 19.
0: Divider by 2 enabled
1: Divider by 2 disabled

Bit 4 = **MSTR** *Master.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 5.6.4.5 Master Mode Fault).
0: Slave mode is selected
1: Master mode is selected, the function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

Bit 3 = **CPOL** *Clock polarity.*
This bit is set and cleared by software. This bit determines the steady state of the serial Clock. The CPOL bit affects both the master and slave modes.
0: The steady state is a low value at the SCK pin.
1: The steady state is a high value at the SCK pin.

Bit 2 = **CPHA** *Clock phase.*
This bit is set and cleared by software.
0: The first clock transition is the first data capture edge.
1: The second clock transition is the first capture edge.

Bit 1:0 = **SPR[1:0]** *Serial peripheral rate.*
These bits are set and cleared by software.Used with the SPR2 bit, they select one of six baud rates to be used as the serial clock when the device is a master.

These 2 bits have no effect in slave mode.

Table 19. Serial Peripheral Baud Rate

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/2$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SERIAL PERIPHERAL INTERFACE (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	-	MODF	-	-	-	-

Bit 7 = **SPIF** *Serial Peripheral data transfer flag.*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE=1 in the CR register. It is cleared by a software sequence (an access to the SR register followed by a read or write to the DR register).
0: Data transfer is in progress or has been approved by a clearing sequence.
1: Data transfer between the device and an external device has been completed.

Note: While the SPIF bit is set, all writes to the DR register are inhibited.

Bit 6 = **WCOL** *Write Collision status.*

This bit is set by hardware when a write to the DR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 51).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = Unused.

Bit 4 = **MODF** *Mode Fault flag.*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 5.6.4.5 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the CR register. This bit is cleared by a software sequence (An access to the SR register while MODF=1 followed by a write to the CR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3-0 = Unused.

DATA I/O REGISTER (DR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The DR register is used to transmit and receive data on the serial bus. In the master device only a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

Warning:

A write to the DR register places data directly into the shift register for transmission.

A write to the the DR register returns the value located in the buffer and not the contents of the shift register (See Figure 48).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 20. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0021h	**SPIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0022h	**SPICR** Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	**SPISR** Reset Value	SPIF 0	WCOL 0	0	MODF 0	0	0	0	0

5.7 SERIAL COMMUNICATIONS INTERFACE (SCI)

5.7.1 Introduction

The Serial Communications Interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

5.7.2 Main Features

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Dual baud rate generator systems
- Independently programmable transmit and receive baud rates up to 250K baud.
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- Two receiver wake-up modes:
  – Address bit (MSB)
  – Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- Three error detection flags:
  – Overrun error
  – Noise error
  – Frame error
- Five interrupt sources with flags:
  – Transmit data register empty
  – Transmission complete
  – Receive data register full
  – Idle line received
  – Overrun error detected

5.7.3 General Description

The interface is externally connected to another device by two pins (see Figure 54):

– TDO: Transmit Data Output. When the transmitter is disabled, the output pin returns to its I/O port configuration. When the transmitter is enabled and nothing is to be transmitted, the TDO pin is at high level.

– RDI: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

Through this pins, serial data is transmitted and received as frames comprising:

– An Idle Line prior to transmission or reception

– A start bit

– A data word (8 or 9 bits) least significant bit first

– A Stop bit indicating that the frame is complete.

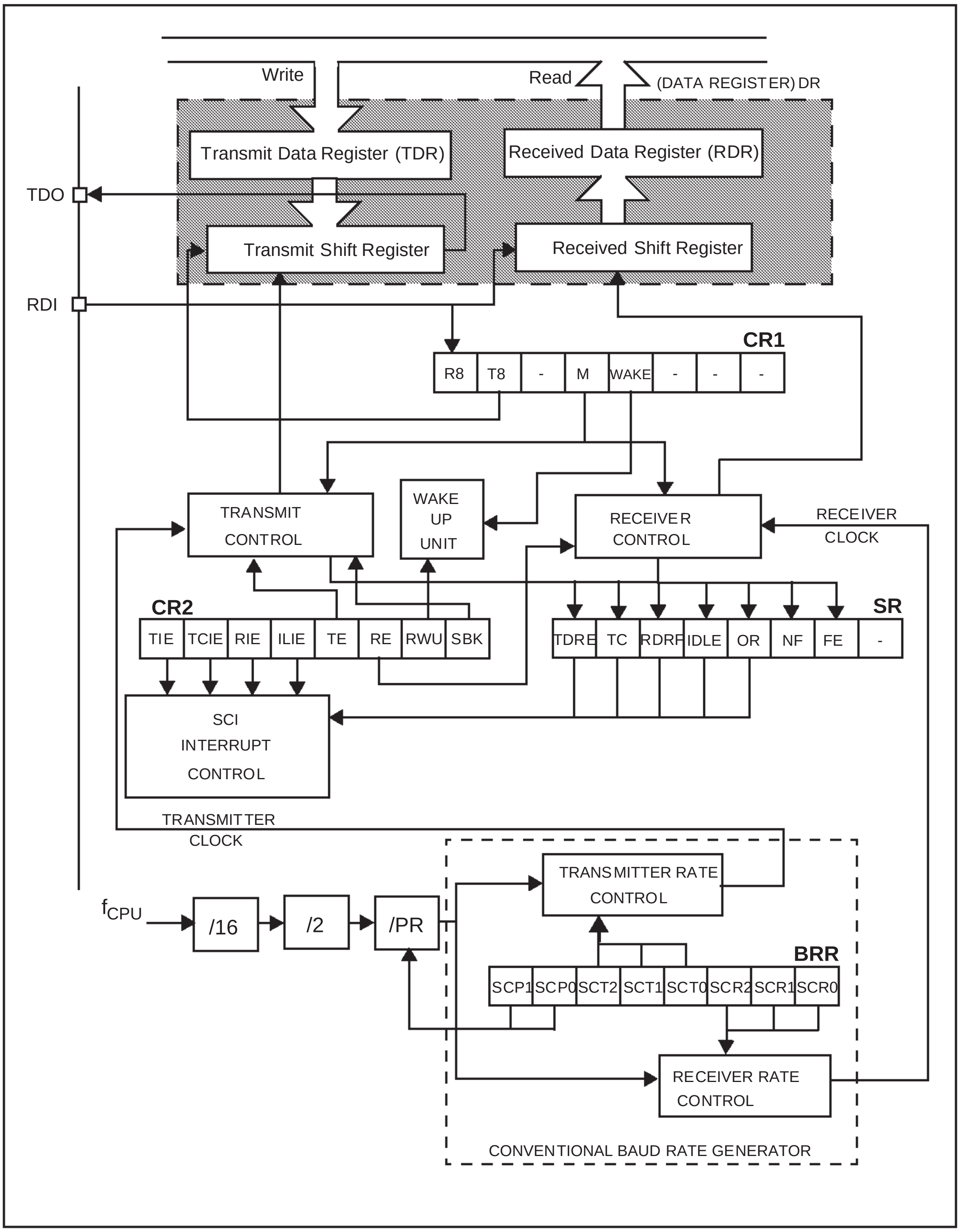
This interface uses two types of baud rate generator:

– A conventional type for commonly-used baud rates,

– An extended type with a prescaler offering a very wide range of baud rates even with non-standard oscillator frequencies.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 53. SCI Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 53. It contains 6 dedicated registers:

– Two control registers (CR1 & CR2)

– A status register (SR)

– A baud rate register (BRR)

– An extended prescaler receiver register (ERPR)

– An extended prescaler transmitter register (ETPR)

Refer to the register descriptions in Section 5.7.7for the definitions of each bit.

5.7.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the CR1 register (see Figure 53).

The TDO pin is in low state during the start bit.

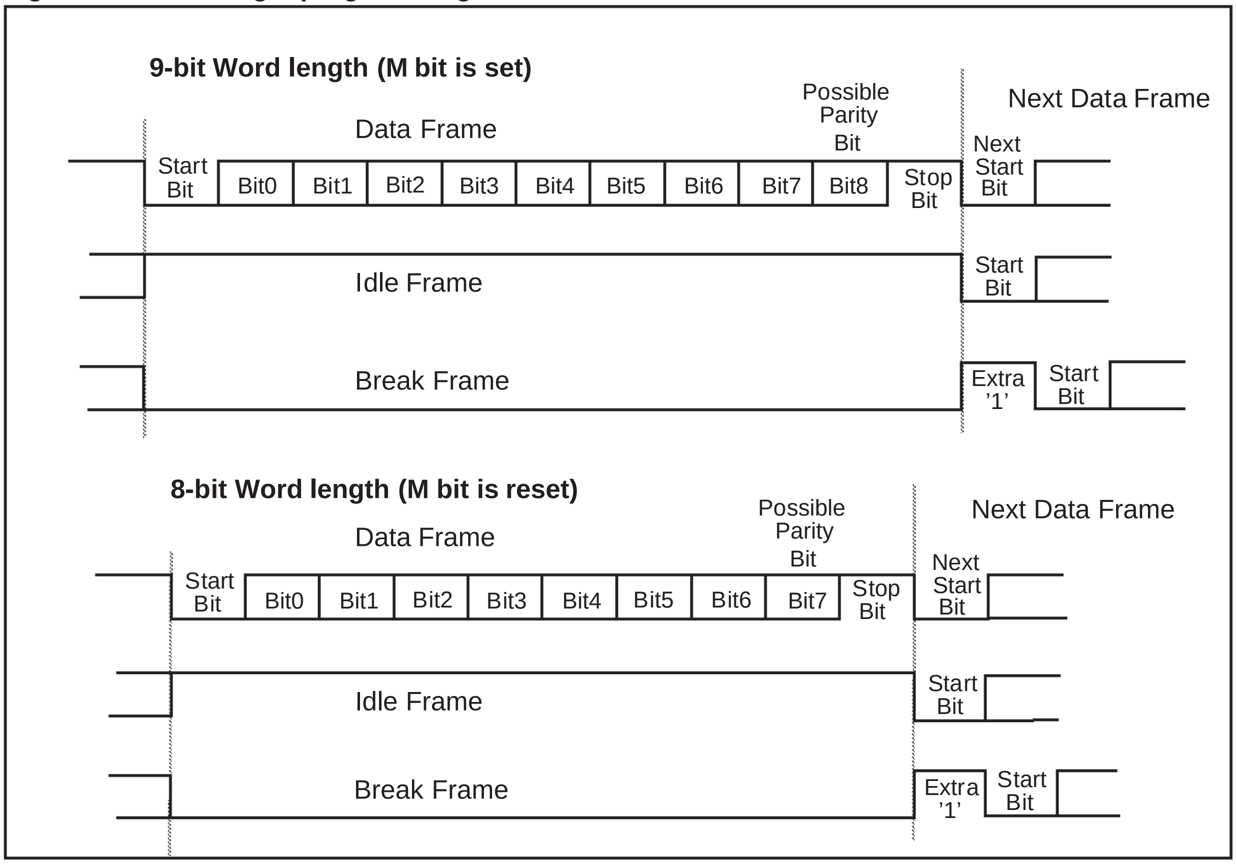
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of "1"s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving "0"s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra "1" bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

Figure 54. Word length programming



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.4.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the CR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the DR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 53).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR and the ETPR registers.

– Set the TE bit to assign the TDO pin to the alternate function and to send a idle frame as first transmission.

– Access the SR register and write the data to send in the DR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:
1. An access to the SR register
2. A write to the DR register

The TDRE bit is set by hardware and it indicates:

– The TDR register is empty.

– The data transfer is beginning.

– The next data can be written in the DR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I bit is cleared in the CCR register.

When a transmission is taking place, a write instruction to the DR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the DR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a frame transmission is complete (after the stop bit or after the break frame) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CCR register.

Clearing the TC bit is performed by the following software sequence:
1. An access to the SR register
2. A write to the DR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break Characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see Figure 54).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this bit by software the SCI insert a logic 1 bit at the end of the last break frame to guarantee the recognition of the start bit of the next frame.

Idle Characters

Setting the TE bit drives the SCI to send an idle frame before the first data frame.

Clearing and then setting the TE bit during a transmission sends an idle frame after the current word.

Note: Resetting and setting the TE bit causes the data in the TDR register to be lost. Therefore the best time to toggle the TE bit is when the TDRE bit is set i.e. before writing the next byte in the DR.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.4.3 Receiver

The SCI can receive data words of either 8 or 9 bits. When the M bit is set, word length is 9 bits and the MSB is stored in the R8 bit in the CR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, DR register consists in a buffer (RDR) between the internal bus and the received shift register (see Figure 53).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the BRR and the ERPR registers.

– Set the RE bit, this enables the receiver which begins searching for a start bit.

When a character is received:

– The RDRF bit is set. It indicates that the content of the shift register is transferred to the RDR.

– An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

– The error flags can be set if a frame error, noise or an overrun error has been detected during reception.

Clearing the RDRF bit is performed by the following software sequence done by:

1. An access to the SR register

2. A read to the DR register.

The RDRF bit must be cleared before the end of the reception of the next character to avoid an overrun error.

Break Character

When a break character is received, the SPI handles it as a framing error.

Idle Character

When a idle frame is detected, there is the same procedure as a data received character plus an interrupt if the ILIE bit is set and the I bit is cleared in the CCR register.

Overrun Error

An overrun error occurs when a character is received when RDRF has not been reset. Data can not be transferred from the shift register to the TDR register as long as the RDRF bit is not cleared.

When a overrun error occurs:

– The OR bit is set.

– The RDR content will not be lost.

– The shift register will be overwritten.

– An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

The OR bit is reset by an access to the SR register followed by a DR register read operation.

Noise Error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

When noise is detected in a frame:

– The NF is set at the rising edge of the RDRF bit.

– Data is transferred from the Shift register to the DR register.

– No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The NF bit is reset by a SR register read operation followed by a DR register read operation.

Framing Error

A framing error is detected when:

– The stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.

– A break is received.

When the framing error is detected:

– the FE bit is set by hardware

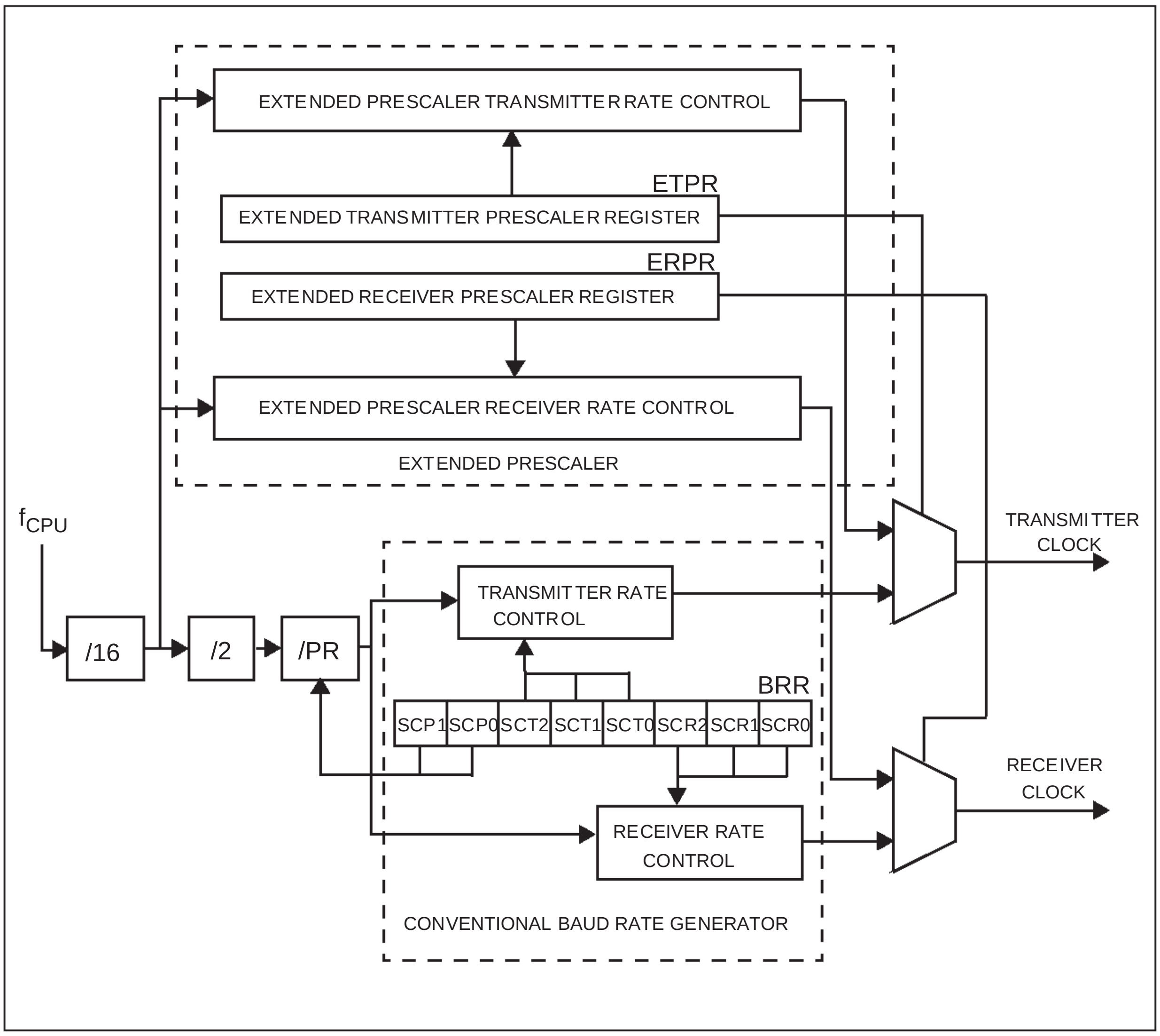
– Data is transferred from the Shift register to the DR register.

– No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The FE bit is reset by a SR register read operation followed by a DR register read operation.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 55. SCI Baud Rate and Extended Prescaler Block Diagram


EXTENDED PRESCALER TRANSMITTER RATE CONTROL
ETPR
EXTENDED TRANSMITTER PRESCALER REGISTER
ERPR
EXTENDED RECEIVER PRESCALER REGISTER
EXTENDED PRESCALER RECEIVER RATE CONTROL
EXTENDED PRESCALER
f_{CPU}
/16
/2
/PR
TRANSMITTER RATE CONTROL
BRR
SCP1 SCP0 SCT2 SCT1 SCT0 SCR2 SCR1 SCR0
RECEIVER RATE CONTROL
CONVENTIONAL BAUD RATE GENERATOR
TRANSMITTER CLOCK
RECEIVER CLOCK

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.4.4 Conventional Baud Rate Generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(32*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(32*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP0 & SCP1 bits)

TR = 1, 2, 4, 8, 16, 32, 64,128

(see SCT0, SCT1 & SCT2 bits)

RR = 1, 2, 4, 8, 16, 32, 64,128

(see SCR0,SCR1 & SCR2 bits)

All this bits are in the BRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR=13 and TR=RR=1, the transmit and receive baud rates are 19200 baud.

Note: the baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

5.7.4.5 Extended Baud Rate Generation

The extended prescaler option gives a very fine tuning on the baud rate, using a 255 value prescaler, whereas the conventional Baud Rate Generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the Figure 55.

The output clock rate sent to the transmitter or to the receiver will be the output from the 16 divider divided by a factor ranging from 1 to 255 set in the ERPR or the ETPR register.

Note: the extended prescaler is activated by setting the ETPR or ERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR} \qquad Rx = \frac{f_{CPU}}{16*ERPR}$$

with:

ETPR = 1,..,255 (see ETPR register)

ERPR = 1,.. 255 (see ERPR register)

5.7.4.6 Receiver Muting and Wake-up Feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non addressed receivers.

The non addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

All the reception status bits can not be set.

All the receive interrupt are inhibited.

A muted receiver may be awakened by one of the following two ways:

– by Idle Line detection if the WAKE bit is reset,

– by Address Mark detection if the WAKE bit is set.

Receiver wakes-up by Idle Line detection when the Receive line has recognised an Idle Frame. Then the RWU bit is reset by hardware but the IDLE bit is not set.

Receiver wakes-up by Address Mark detection when it received a “1” as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.5 Low Power Modes

Mode	Description
WAIT	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
HALT	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

5.7.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrrun Error Detected	OR		Yes	No
Idle Line Detected	IDLE	ILIE	Yes	No

The SCI interrupt events are connected to the same interrupt vector (see Interrupts chapter).

These events generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

5.7.7 Register Description

STATUS REGISTER (SR)

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	-

Bit 7 = **TDRE** *Transmit data register empty.*
This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE =1 in the CR2 register. It is cleared by a software sequence (an access to the SR register followed by a write to the DR register).
0: Data is not transferred to the shift register
1: Data is transferred to the shift register

Note: data will not be transferred to the shift register as long as the TDRE bit is not reset.

Bit 6 = **TC** *Transmission complete.*
This bit is set by hardware when transmission of a frame containing Data, a Preamble or a Break is complete. An interrupt is generated if TCIE=1 in the CR2 register. It is cleared by a software sequence (an access to the SR register followed by a write to the DR register).
0: Transmission is not complete
1: Transmission is complete

Bit 5 = **RDRF** *Received data ready flag.*
This bit is set by hardware when the content of the RDR register has been transferred into the DR register. An interrupt is generated if RIE=1 in the CR2 register. It is cleared by hardware when RE=0 or by a software sequence (an access to the SR register followed by a read to the DR register).
0: Data is not received
1: Received data is ready to be read

Bit 4 = **IDLE** *Idle line detect.*
This bit is set by hardware when a Idle Line is detected. An interrupt is generated if the ILIE=1 in the CR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Idle Line is detected
1: Idle Line is detected

Note: The IDLE bit will not be set again until the RDRF bit has been set itself (i.e. a new idle line occurs). This bit is not set by an idle line when the receiver wakes up from wake-up mode.

Bit 3 = **OR** *Overrun error.*
This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF=1. An interrupt is generated if RIE=1 in the CR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Overrun error
1: Overrun error is detected

Note: When this bit is set RDR register content will not be lost but the shift register will be overwritten.

Bit 2 = **NF** *Noise flag.*
This bit is set by hardware when noise is detected on a received frame. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No noise is detected
1: Noise is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt.

Bit 1 = **FE** *Framing error.*
This bit is set by hardware when a de-synchronization, excessive noise or a break character is detected. It is cleared by hardware when RE=0 by a software sequence (an access to the SR register followed by a read to the DR register).
0: No Framing error is detected
1: Framing error or break character is detected

Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt. If the word currently being transferred causes both frame error and overrun error, it will be transferred and only the OR bit will be set.

Bit 0 = Unused.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: Undefined

7							0
R8	T8	-	M	WAKE	-	-	-

Bit 7 = **R8** *Receive data bit 8.*
This bit is used to store the 9th bit of the received word when M=1.

Bit 6 = **T8** *Transmit data bit 8.*
This bit is used to store the 9th bit of the transmitted word when M=1.

Bit 4 = **M** *Word length.*
This bit determines the word length. It is set or cleared by software.
0: 1 Start bit, 8 Data bits, 1 Stop bit
1: 1 Start bit, 9 Data bits, 1 Stop bit

Bit 3 = **WAKE** *Wake-Up method.*
This bit determines the SCI Wake-Up method, it is set or cleared by software.
0: Idle Line
1: Address Mark

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

Bit 7 = **TIE** *Transmitter interrupt enable.*
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever TDRE=1 in the SR register.

Bit 6 = **TCIE** *Transmission complete interrupt enable*

This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever TC=1 in the SR register

Bit 5 = **RIE** *Receiver interrupt enable.*
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever OR=1 or RDRF=1 in the SR register

Bit 4 = **ILIE** *Idle line interrupt enable.*
This bit is set and cleared by software.
0: interrupt is inhibited
1: An SCI interrupt is generated whenever IDLE=1 in the SR register.

Bit 3 = **TE** *Transmitter enable.*
This bit enables the transmitter and assigns the TDO pin to the alternate function. It is set and cleared by software.
0: Transmitter is disabled, the TDO pin is back to the I/O port configuration.
1: Transmitter is enabled

Note: during transmission, a “0” pulse on the TE bit (“0” followed by “1”) sends a preamble after the current word.

Bit 2 = **RE** *Receiver enable.*
This bit enables the receiver. It is set and cleared by software.
0: Receiver is disabled, it resets the RDRF, IDLE, OR, NF and FE bits of the SR register.
1: Receiver is enabled and begins searching for a start bit.

Bit 1 = **RWU** *Receiver wake-up.*
This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized.
0: Receiver in active mode
1: Receiver in mute mode

Bit 0 = **SBK** *Send break.*
This bit set is used to send break characters. It is set and cleared by software.
0: No break character is transmitted
1: Break characters are transmitted

Note: If the SBK bit is set to “1” and then to “0”, the transmitter will send a BREAK word at the end of the current word.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

DATA REGISTER (DR)

Read/Write

Reset Value: Undefined

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).
The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 53).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 53).

BAUD RATE REGISTER (BRR)

Read/Write

Reset Value: 00xx xxxx (XXh)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bit 7:6= **SCP[1:0]** *First SCI Prescaler*
These 2 prescaling bits allow several standard clock division ranges:

PR Prescaling factor	SCP1	SCP0
1	0	0
3	0	1
4	1	0
13	1	1

Bit 5:3 = **SCT[2:0]** *SCI Transmitter rate divisor*

These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the transmit rate clock in conventional Baud Rate Generator mode.

TR dividing factor	SCT2	SCT1	SCT0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Note: this TR factor is used only when the ETPR fine tuning factor is equal to 00h; otherwise, TR is replaced by the ETPR dividing factor.

Bit 2:0 = **SCR[2:0]** *SCI Receiver rate divisor.*

These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the receive rate clock in conventional Baud Rate Generator mode.

RR dividing factor	SCR2	SCR1	SCR0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Note: this RR factor is used only when the ERPR fine tuning factor is equal to 00h; otherwise, RR is replaced by the ERPR dividing factor.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

EXTENDED RECEIVE PRESCALER DIVISION REGISTER (ERPR)

Read/Write

Reset Value: 0000 0000 (00h)

Allows setting of the Extended Prescaler rate division factor for the receive circuit.

7							0
ERPR 7	ERPR 6	ERPR 5	ERPR 4	ERPR 3	ERPR 2	ERPR 1	ERPR 0

Bit 7:1 = **ERPR[7:0]** *8-bit Extended Receive Prescaler Register.*

The extended Baud Rate Generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see Figure 55) is divided by the binary factor set in the ERPR register (in the range 1 to 255).

The extended baud rate generator is not used after a reset.

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (ETPR)

Read/Write

Reset Value:0000 0000 (00h)

Allows setting of the External Prescaler rate division factor for the transmit circuit.

7							0
ETPR 7	ETPR 6	ETPR 5	ETPR 4	ETPR 3	ETPR 2	ETPR 1	ETPR 0

Bit 7:1 = **ETPR[7:0]** *8-bit Extended Transmit Prescaler Register.*

The extended Baud Rate Generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see Figure 55) is divided by the binary factor set in the ETPR register (in the range 1 to 255).

The extended baud rate generator is not used after a reset.

Table 21. SCI Register Map and Reset Values

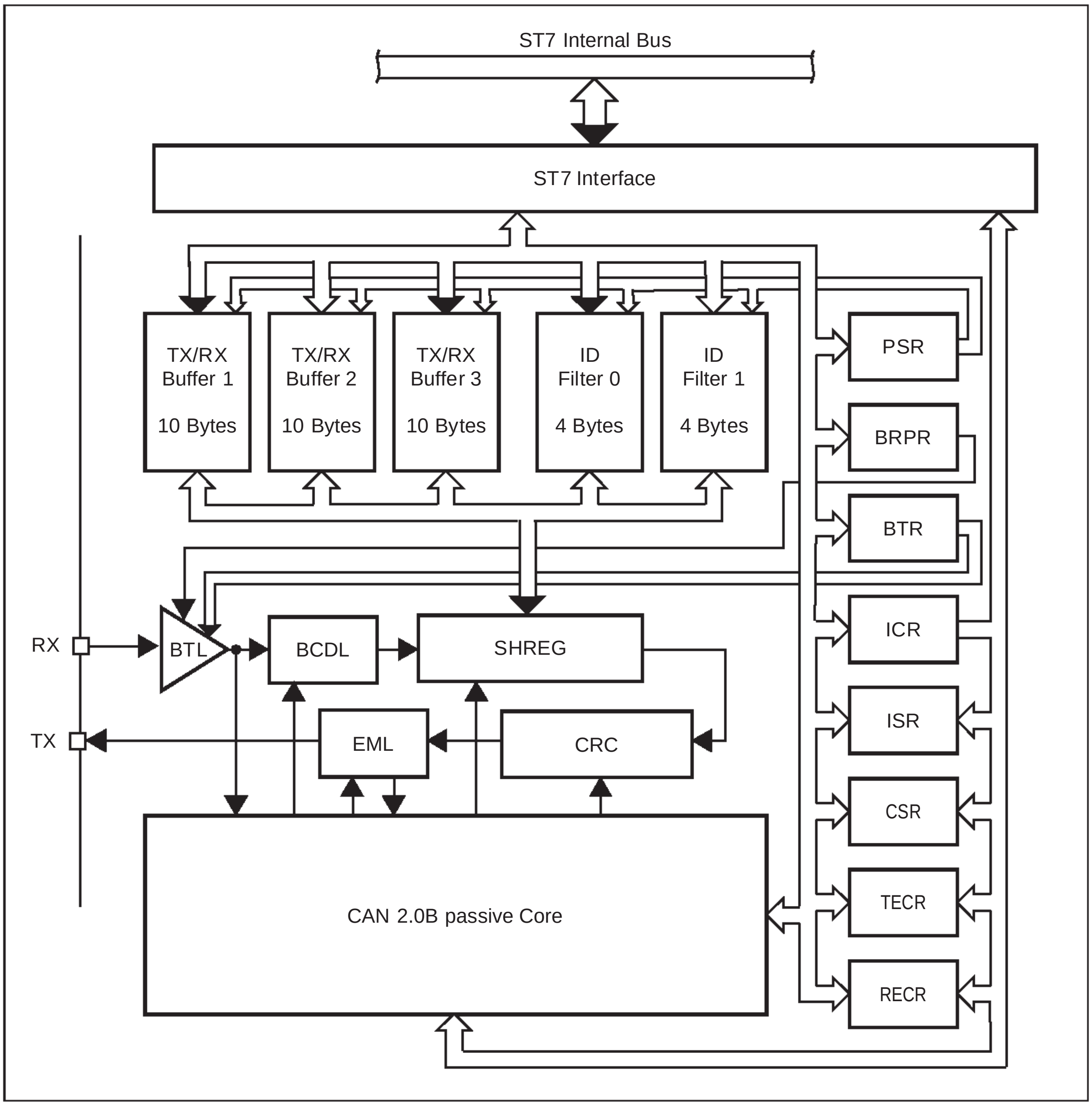
Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0050h	**SCISR** Reset Value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	0
0051h	**SCIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0052h	**SCIBRR** Reset Value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	**SCICR1** Reset Value	R8 x	T8 x	0	M x	WAKE x	0	0	0
0054h	**SCICR2** Reset Value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	**SCIERPR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	**SCIETPR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

5.8 CONTROLLER AREA NETWORK (CAN)

5.8.1 Introduction

This peripheral is designed to support serial data exchanges using a multi-master contention based priority scheme as described in CAN specification Rev. 2.0 part A. It can also be connected to a 2.0 B network without problems, since extended frames are checked for correctness and acknowledged accordingly although such frames cannot be transmitted nor received. The same applies to overload frames which are recognized but never initiated.

Figure 56. CAN Block Diagram



CONTROLLER AREA NETWORK (Cont'd)

5.8.2 Main Features

– Support of CAN specification 2.0A and 2.0B passive
– Three prioritized 10-byte Transmit/Receive message buffers
– Two programmable global 12-bit message acceptance filters
– Programmable baud rates up to 1 MBit/s
– Buffer flip-flopping capability in transmission
– Maskable interrupts for transmit, receive (one per buffer), error and wake-up
– Automatic low-power mode after 20 recessive bits or on demand (standby mode)
– Interrupt-driven wake-up from standby mode upon reception of dominant pulse
– Optional dominant pulse transmission on leaving standby mode
– Automatic message queuing for transmission upon writing of data byte 7
– Programmable loop-back mode for self-test operation
– Advanced error detection and diagnosis functions
– Software-efficient buffer mapping at a unique address space
– Scalable architecture.

5.8.3 Functional Description

5.8.3.1 Frame Formats

A summary of all the CAN frame formats is given in Figure 57 for reference. It covers only the standard frame format since the extended one is only acknowledged.

A message begins with a start bit called Start Of Frame (SOF). This bit is followed by the arbitration field which contains the 11-bit identifier (ID) and the Remote Transmission Request bit (RTR). The RTR bit indicates whether it is a data frame or a remote request frame. A remote request frame does not have any data byte.

The control field contains the Identifier Extension bit (IDE), which indicates standard or extended format, a reserved bit (ro) and, in the last four bits, a count of the data bytes (DLC). The data field ranges from zero to eight bytes and is followed by the Cyclic Redundancy Check (CRC) used as a frame integrity check for detecting bit errors.

The acknowledgement (ACK) field comprises the ACK slot and the ACK delimiter. The bit in the ACK slot is placed on the bus by the transmitter as a recessive bit (logical 1). It is overwritten as a dominant bit (logical 0) by those receivers which have at this time received the data correctly. In this way, the transmitting node can be assured that at least one receiver has correctly received its message. Note that messages are acknowledged by the receivers regardless of the outcome of the acceptance test.

The end of the message is indicated by the End Of Frame (EOF). The intermission field defines the minimum number of bit periods separating consecutive messages. If there is no subsequent bus access by any station, the bus remains idle.

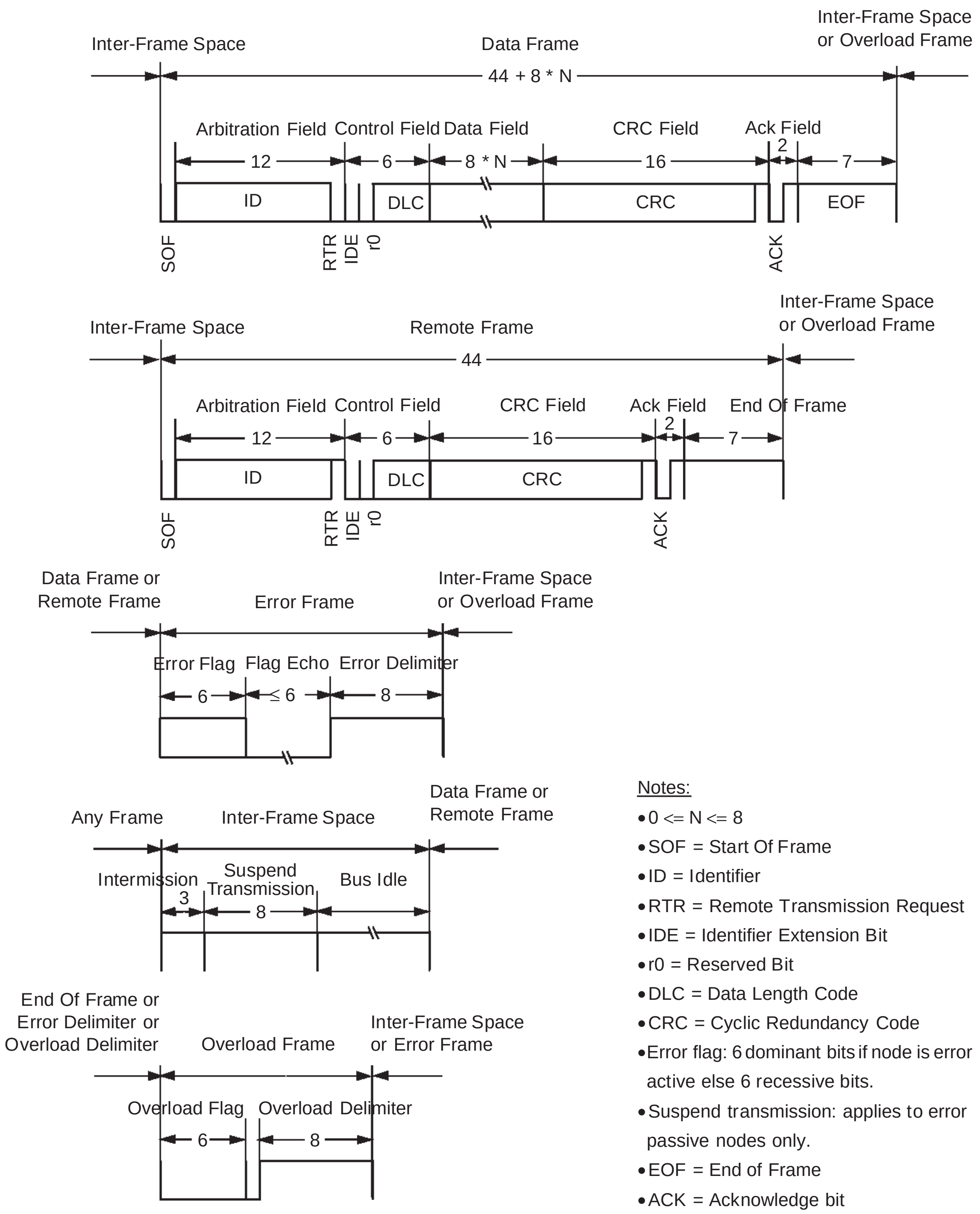
5.8.3.2 Hardware Blocks

The CAN controller contains the following functional blocks (refer to Figure 56):

– ST7 Interface: buffering of the ST7 internal bus and address decoding of the CAN registers.
– TX/RX Buffers: three 10-byte buffers for transmission and reception of maximum length messages.
– ID Filters: two 12-bit compare and don't care masks for message acceptance filtering.
– PSR: page selection register (see memory map).
– BRPR: clock divider for different data rates.
– BTR: bit timing register.
– ICR: interrupt control register.
– ISR: interrupt status register.
– CSR: general purpose control/status register.
– TECR: transmit error counter register.
– RECR: receive error counter register.
– BTL: bit timing logic providing programmable bit sampling and bit clock generation for synchronization of the controller.
– BCDL: bit coding logic generating a NRZ-coded datastream with stuff bits.
– SHREG: 8-bit shift register for serialization of data to be transmitted and parallelisation of received data.
– CRC: 15-bit CRC calculator and checker.
– EML: error detection and management logic.
– CAN Core: CAN 2.0B passive protocol controller.

CONTROLLER AREA NETWORK (Cont'd)

Figure 57. CAN Frames



Notes:
- 0 <= N <= 8
- SOF = Start Of Frame
- ID = Identifier
- RTR = Remote Transmission Request
- IDE = Identifier Extension Bit
- r0 = Reserved Bit
- DLC = Data Length Code
- CRC = Cyclic Redundancy Code
- Error flag: 6 dominant bits if node is error active else 6 recessive bits.
- Suspend transmission: applies to error passive nodes only.
- EOF = End of Frame
- ACK = Acknowledge bit

CONTROLLER AREA NETWORK (Cont'd)

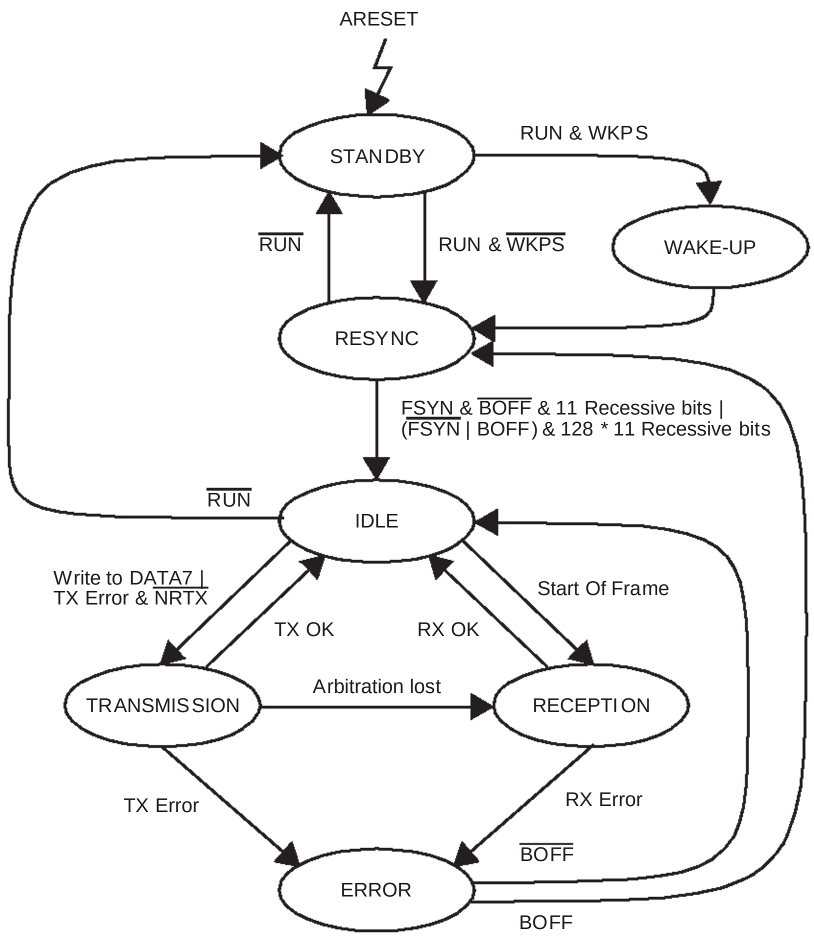
5.8.3.3 Modes of Operation

The CAN Core unit assumes one of the seven states described below:

– **STANDBY**. Standby mode is entered either on a chip reset or on resetting the RUN bit in the Control/Status Register (CSR). Any on-going transmission or reception operation is not interrupted and completes normally before the Bit Time Logic and the clock prescaler are turned off for minimum power consumption. This state is signalled by the RUN bit being read-back as 0.
  Once in standby, the only event monitored is the reception of a dominant bit which causes a wake-up interrupt if the SCIE bit of the Interrupt Control Register (ICR) is set.
  The STANDBY mode is left by setting the RUN bit. If the WKPS bit is set in the CSR register, then the controller passes through WAKE-UP otherwise it enters RESYNC directly.
  It is important to note that the wake-up mechanism is software-driven and therefore carries a significant time overhead. All messages received after the wake-up bit and before the controller is set to run and has completed synchronization are ignored.

– **WAKE-UP**. The CAN bus line is forced to dominant for one bit time signalling the wake-up condition to all other bus members.

Figure 58. CAN Controller State Diagram



CONTROLLER AREA NETWORK (Cont'd)

– **RESYNC**. The resynchronization mode is used to find the correct entry point for starting transmission or reception after the node has gone asynchronous either by going into the STANDBY or bus-off states.
  Resynchronization is achieved when 128 sequences of 11 recessive bits have been monitored unless the node is not bus-off and the FSYN bit in the CSR register is set in which case a single sequence of 11 recessive bits needs to be monitored.

– **IDLE**. The CAN controller looks for one of the following events: the RUN bit is reset, a Start Of Frame appears on the CAN bus or the DATA7 register of the currently active page is written to.

– **TRANSMISSION**. Once the LOCK bit of a Buffer Control/Status Register (BCSRx) has been set and read back as such, a transmit job can be submitted by writing to the DATA7 register. The message with the highest priority will be transmitted as soon as the CAN bus becomes idle. Among those messages with a pending transmission request, the highest priority is given to Buffer 3 then 2 and 1. If the transmission fails due to a lost arbitration or to an error while the NRTX bit of the CSR register is reset, then a new transmission attempt is performed . This goes on until the transmission ends successfully or until the job is cancelled by unlocking the buffer, by setting the NRTX bit or if the node ever enters bus-off or if a higher priority message becomes pending. The RDY bit in the BCSRx register, which was set since the job was submitted, gets reset. When a transmission is in progress, the BUSY bit in the BCSRx register is set. If it ends successfully then the TXIF bit in the Interrupt Status Register (ISR) is set, else the TEIF bit is set. An interrupt is generated in either case provided the TXIE and TEIE bits of the ICR register are set. The ETX bit in the same register is used to get an early transmit interrupt and to automatically unlock the transmitting buffer upon successful completion of its job. This enables the CPU to get a new transmit job pending by the end of the current transmission while always leaving two buffers available for reception. An uninterrupted stream of messages may be transmitted in this way at no overrun risk.

  Note 1: Setting the SRTE bit of the CSR register allows transmitted messages to be simultaneously received when they pass the acceptance filtering. This is particularly useful for checking the integrity of the communication path.
  Note 2: When the ETX bit is reset, the buffer with the highest priority and with a pending transmission request is always transmitted. When the ETX bit is set, once a buffer participates in the arbitration phase, it is sent until it wins the arbitration even if another transmission is requested from a buffer with a higher priority.

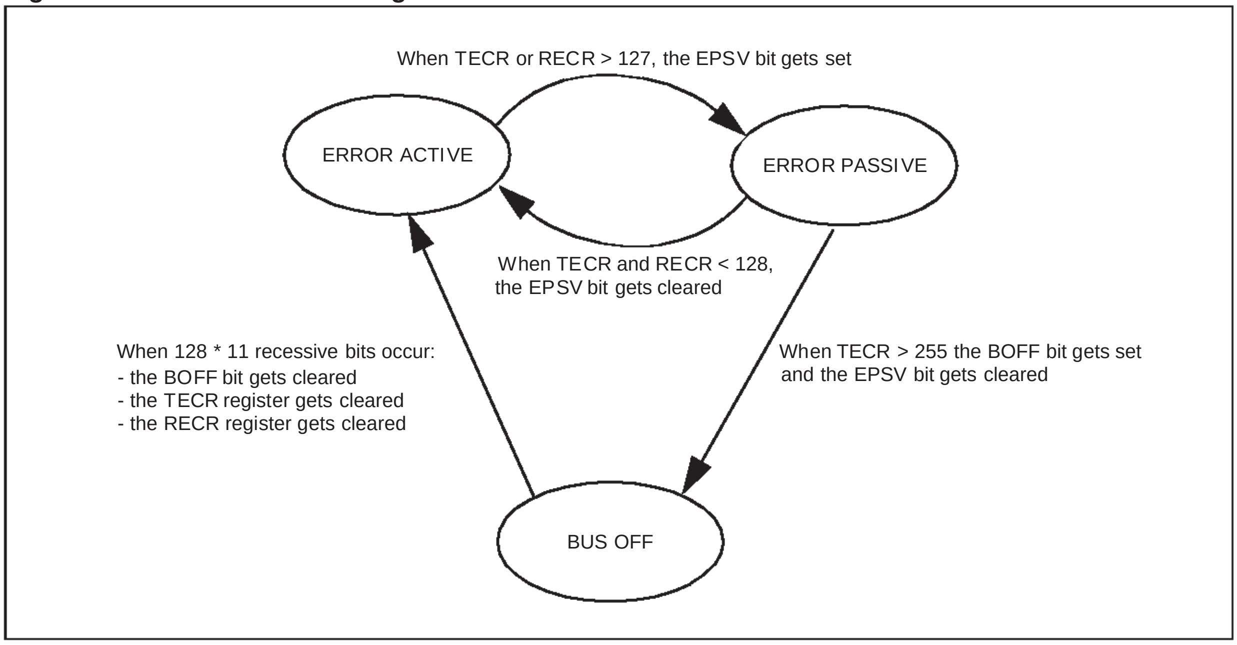
– **RECEPTION**. Once the CAN controller has synchronized itself onto the bus activity, it is ready for reception of new messages. Every incoming message gets its identifier compared to the acceptance filters. If the bitwise comparison of the selected bits ends up with a match for at least one of the filters then that message is elected for reception and a target buffer is searched for. This buffer will be the first one - order is 1 to 3 - that has the LOCK and RDY bits of its BCSRx register reset.
  – When no such buffer exists then an overrun interrupt is generated if the ORIE bit of the ICR register has been set. In this case the identifier of the last message is made available in the Last Identifier Register (LIDHR and LIDLR) at least until it gets overwritten by a new identifier picked-up from the bus.
  – When a buffer does exist, the accepted message gets written into it, the ACC bit in the BCSRx register gets the number of the matching filter, the RDY and RXIF bits get set and an interrupt is generated if the RXIE bit in the ISR register is set.

  Up to three messages can be automatically received without intervention from the CPU because each buffer has its own set of status bits, greatly reducing the reactiveness requirements in the processing of the receive interrupts.

CONTROLLER AREA NETWORK (Cont'd)

– **ERROR**. The error management as described in the CAN protocol is completely handled by hardware using 2 error counters which get incremented or decremented according to the error condition. Both of them may be read by the application to determine the stability of the network. Moreover, as one of the node status bits (EPSV or BOFF of the CSR register) changes, an interrupt is generated if the SCIE bit is set in the ICR Register. Refer to Figure 59.

Figure 59. CAN Error State Diagram


When TECR or RECR > 127, the EPSV bit gets set
ERROR ACTIVE
ERROR PASSIVE
When TECR and RECR < 128, the EPSV bit gets cleared
When 128 * 11 recessive bits occur:
- the BOFF bit gets cleared
- the TECR register gets cleared
- the RECR register gets cleared
When TECR > 255 the BOFF bit gets set and the EPSV bit gets cleared
BUS OFF

CONTROLLER AREA NETWORK (Cont'd)

5.8.3.4 Bit Timing Logic

The bit timing logic monitors the serial bus-line and performs sampling and adjustment of the sample point by synchronizing on the start-bit edge and re-synchronizing on following edges.

Its operation may be explained simply when the nominal bit time is divided into three segments as follows:

– **Synchronisation segment (SYNC_SEG)**: a bit change is expected to lie within this time segment. It has a fixed length of one time quanta (1 x t_{CAN}).

– **Bit segment 1 (BS1)**: defines the location of the sample point. It includes the PROP_SEG and PHASE_SEG1 of the CAN standard. Its duration is programmable between 1 and 16 time quanta but may be automatically lengthened to compensate for positive phase drifts due to differences in the frequency of the various nodes of the network.

– **Bit segment 2 (BS2)**: defines the location of the transmit point. It represents the PHASE_SEG2 of the CAN standard. Its duration is programmable between 1 and 8 time quanta but may also be automatically shortened to compensate for negative phase drifts.

The resynchronization jump width (RJW) defines an upper bound to the amount of lengthening or shortening of the bit segments. It is programmable between 1 and 4 time quanta.

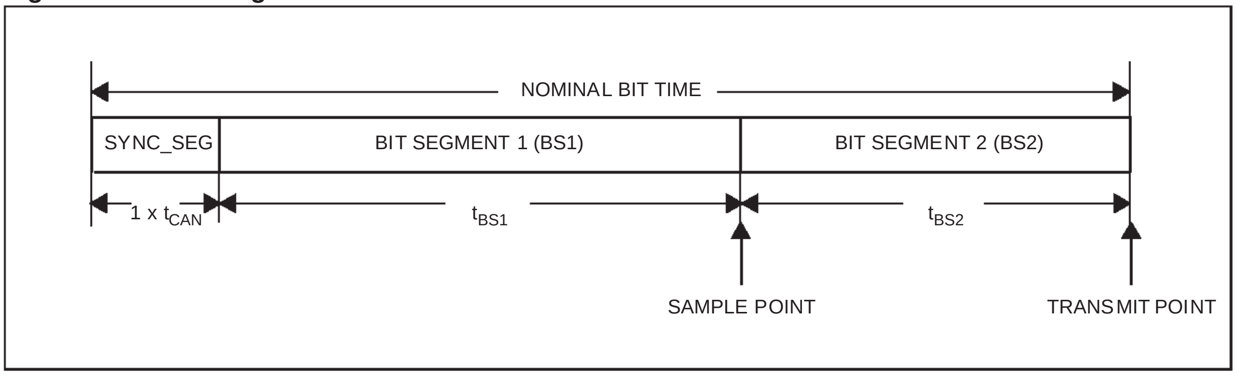
A valid edge is defined as the first transition in a bit time from dominant to recessive bus level provided the controller itself does not send a recessive bit.

If a valid edge is detected in BS1 instead of SYNC_SEG, BS1 is extended by up to RJW so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2 instead of SYNC_SEG, BS2 is shortened by up to RJW so that the transmit point is moved earlier.

As a safeguard against programming errors, the configuration of the Bit Timing Register (BTR) is only possible while the device is in STANDBY mode.

Figure 60. Bit Timing



CONTROLLER AREA NETWORK (Cont'd)

5.8.4 Register Description

The CAN registers are organized as 6 general purpose registers plus 5 pages of 16 registers spanning the same address space and primarily used for message and filter storage. The page actually selected is defined by the content of the Page Selection Register. Refer to Figure 61.

5.8.4.1 General Purpose Registers

INTERRUPT STATUS REGISTER (ISR)

Read/Write

Reset Value: 00h

7							0
RXIF3	RXIF2	RXIF1	TXIF	SCIF	ORIF	TEIF	EPND

Bit 7 = **RXIF3** *Receive Interrupt Flag for Buffer 3*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 3.
Cleared by software to release buffer 3.
Also cleared by resetting bit RDY of BCSR3.

Bit 6 = **RXIF2** *Receive Interrupt Flag for Buffer 2*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 2.
Cleared by software to release buffer 2.
Also cleared by resetting bit RDY of BCSR2.

Bit 5 = **RXIF1** *Receive Interrupt Flag for Buffer 1*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 1.
Cleared by software to release buffer 1.
Also cleared by resetting bit RDY of BCSR1.

Bit 4 = **TXIF** *Transmit Interrupt Flag*
– Read/Clear
Set by hardware to signal that the highest priority message queued for transmission has been successfully transmitted (ETX = 0) or that it has passed successfully the arbitration (ETX = 1).
Cleared by software.

Bit 3 = **SCIF** *Status Change Interrupt Flag*
– Read/Clear
Set by hardware to signal the reception of a dominant bit while in standby or a change from error active to error passive and bus-off while in run. Also signals any receive error when ESCI = 1.
Cleared by software.

Bit 2 = **ORIF** *Overrun Interrupt Flag*
– Read/Clear
Set by hardware to signal that a message could not be stored because no receive buffer was available.
Cleared by software.

Bit 1 = **TEIF** *Transmit Error Interrupt Flag*
– Read/Clear
Set by hardware to signal that an error occurred during the transmission of the highest priority message queued for transmission.
Cleared by software.

Bit 0 = **EPND** *Error Interrupt Pending*
– Read Only
Set by hardware when at least one of the three error interrupt flags SCIF, ORIF or TEIF is set.
Reset by hardware when all error interrupt flags have been cleared.

Caution;

Interrupt flags are reset by writing a "0" to the corresponding bit position. The appropriate way consists in writing an immediate mask or the one's complement of the register content initially read by the interrupt handler. Bit manipulation instruction BRES should never be used due to its read-modify-write nature.

CONTROLLER AREA NETWORK (Cont'd)

INTERRUPT CONTROL REGISTER (ICR)

Read/Write

Reset Value: 00h

7							0
0	ESCI	RXIE	TXIE	SCIE	ORIE	TEIE	ETX

Bit 6 = **ESCI** *Extended Status Change Interrupt*
— Read/Set/Clear
Set by software to specify that SCIF is to be set on receive errors also.
Cleared by software to set SCIF only on status changes and wake-up but not on all receive errors.

Bit 5 = **RXIE** *Receive Interrupt Enable*
— Read/Set/Clear
Set by software to enable an interrupt request whenever a message has been received free of errors.
Cleared by software to disable receive interrupt requests.

Bit 4 = TXIE *Transmit Interrupt Enable*
— Read/Set/Clear
Set by software to enable an interrupt request whenever a message has been successfully transmitted.
Cleared by software to disable transmit interrupt requests.

Bit 3 = **SCIE** *Status Change Interrupt Enable*
— Read/Set/Clear
Set by software to enable an interrupt request whenever the node's status changes in run mode or whenever a dominant pulse is received in standby mode.
Cleared by software to disable status change interrupt requests.

Bit 2 = **ORIE** *Overrun Interrupt Enable*
— Read/Set/Clear
Set by software to enable an interrupt request whenever a message should be stored and no receive buffer is avalaible.
Cleared by software to disable overrun interrupt requests.

Bit 1 = **TEIE** *Transmit Error Interrupt Enable*
— Read/Set/Clear
Set by software to enable an interrupt whenever an error has been detected during transmission of a message.
Cleared by software to disable transmit error interrupts.

Bit 0 = **ETX** *Early Transmit Interrupt*
— Read/Set/Clear
Set by software to request the transmit interrupt to occur as soon as the arbitration phase has been passed successfully.
Cleared by software to request the transmit interrupt to occur at the completion of the transfer.

CONTROLLER AREA NETWORK (Cont'd)

CONTROL/STATUS REGISTER (CSR)

Read/Write

Reset Value: 00h

7							0
0	BOFF	EPSV	SRTE	NRTX	FSYN	WKPS	RUN

Bit 6 = **BOFF** *Bus-Off State*
– Read Only
Set by hardware to indicate that the node is in bus-off state, i.e. the Transmit Error Counter exceeds 255.
Reset by hardware to indicate that the node is involved in bus activities.

Bit 5 = **EPSV** *Error Passive State*
– Read Only
Set by hardware to indicate that the node is error passive.
Reset by hardware to indicate that the node is either error active (BOFF = 0) or bus-off.

Bit 4 = **SRTE** *Simultaneous Receive/Transmit Enable* – Read/Set/Clear
Set by software to enable simultaneous transmission and reception of a message passing the acceptance filtering. Allows to check the integrity of the communication path.
Reset by software to discard all messages transmitted by the node. Allows remote and data frames to share the same identifier.

Bit 3 = **NRTX** *No Retransmission*
– Read/Set/Clear
Set by software to disable the retransmission of unsuccessful messages.
Cleared by software to enable retransmission of messages until success is met.

Bit 2 = **FSYN** *Fast Synchronization*
– Read/Set/Clear
Set by software to enable a fast resynchronization when leaving standby mode, i.e. wait for only 11 recessive bits in a row.
Cleared by software to enable the standard resynchronization when leaving standby mode, i.e. wait for 128 sequences of 11 recessive bits.

Bit 1 = **WKPS** *Wake-up Pulse*
– Read/Set/Clear
Set by software to generate a dominant pulse when leaving standby mode.
Cleared by software for no dominant wake-up pulse.

Bit 0 = **RUN** *CAN Enable*
– Read/Set/Clear

Set by software to leave standby mode after 128 sequences of 11 recessive bits or just 11 recessive bits if FSYN is set.
Cleared by software to request a switch to the standby or low-power mode as soon as any on-going transfer is complete. Read-back as 1 in the meantime to enable proper signalling of the standby state. The CPU clock may therefore be safely switched OFF whenever RUN is read as 0.

CONTROLLER AREA NETWORK (Cont'd)

BAUD RATE PRESCALER REGISTER (BRPR)

Read/Write in Standby mode

Reset Value: 00h

7							0
RJW1	RJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

RJW[1:0] determine the maximum number of time quanta by which a bit period may be shortened or lengthened to achieve resynchronization.
$t_{RJW} = t_{CAN} * (RJW + 1)$

BRP[5:0] determine the CAN system clock cycle time or time quanta which is used to build up the individual bit timing.
$t_{CAN} = t_{CPU} * (BRP + 1)$

Where t_{CPU} = time period of the CPU clock.
The resulting baud rate can be computed by the formula:

$$BR = \frac{1}{t_{CPU} \times (BRP + 1) \times (BS1 + BS2 + 3)}$$

Note: Writing to this register is allowed only in Standby mode to prevent any accidental CAN protocol violation through programming errors.

BIT TIMING REGISTER (BTR)

Read/Write in Standby mode

Reset Value: 23h

7							0
0	BS22	BS21	BS20	BS13	BS12	BS11	BS10

BS2[2:0] determine the length of Bit Segment 2.
$t_{BS2} = t_{CAN} * (BS2 + 1)$

BS1[3:0] determine the length of Bit Segment 1.
$t_{BS1} = t_{CAN} * (BS1 + 1)$

Note: Writing to this register is allowed only in Standby mode to prevent any accidental CAN protocol violation through programming errors.

PAGE SELECTION REGISTER (PSR)

Read/Write

Reset Value: 00h

7							0
0	0	0	0	0	PAGE2	PAGE1	PAGE0

PAGE[2:0] determine which buffer or filter page is mapped at addresses 0010h to 001Fh.

PAGE2	PAGE1	PAGE0	Page Title
0	0	0	Diagnosis
0	0	1	Buffer 1
0	1	0	Buffer 2
0	1	1	Buffer 3
1	0	0	Filters
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

CONTROLLER AREA NETWORK (Cont'd)

5.8.4.2 Paged Registers

LAST IDENTIFIER HIGH REGISTER (LIDHR)

Read/Write

Reset Value: Undefined

7							0
LID10	LID9	LID8	LID7	LID6	LID5	LID4	LID3

LID[10:3] are the most significant 8 bits of the last Identifier read on the CAN bus.

LAST IDENTIFIER LOW REGISTER (LIDLR)

Read/Write

Reset Value: Undefined

7							0
LID2	LID1	LID0	LRTR	LDLC3	LDLC2	LDLC1	LDLC0

LID[2:0] are the least significant 3 bits of the last Identifier read on the CAN bus.

LRTR is the last Remote Transmission Request bit read on the CAN bus.

LDLC[3:0] is the last Data Length Code read on the CAN bus.

TRANSMIT ERROR COUNTER REG. (TECR)

Read Only

Reset Value: 00h

7							0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

TEC[7:0] is the least significant byte of the 9-bit Transmit Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during transmission, this counter is incremented by 8. It is decremented by 1 after every successful transmission. When the counter value exceeds 127, the CAN controller enters the error passive state. When a value of 256 is reached, the CAN controller is disconnected from the bus.

RECEIVE ERROR COUNTER REG. (RECR)

Page: 00h — Read Only

Reset Value: 00h

7							0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0

REC[7:0] is the Receive Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during reception, this counter is incremented by 1 or by 8 depending on the error condition as defined by the CAN standard. After every successful reception the counter is decremented by 1 or reset to 120 if its value was higher than 128. When the counter value exceeds 127, the CAN controller enters the error passive state.

IDENTIFIER HIGH REGISTERS (IDHRx)

Read/Write

Reset Value: Undefined

7							0
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3

ID[10:3] are the most significant 8 bits of the 11-bit message identifier.The identifier acts as the message's name, used for bus access arbitration and acceptance filtering.

CONTROLLER AREA NETWORK (Cont'd)

IDENTIFIER LOW REGISTERS (IDLRx)

Read/Write

Reset Value: Undefined

7							0
ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0

ID[2:0] are the least significant 3 bits of the 11-bit message identifier.

RTR is the Remote Transmission Request bit. It is set to indicate a remote frame and reset to indicate a data frame.

DLC[3:0] is the Data Length Code. It gives the number of bytes in the data field of the message.The valid range is 0 to 8.

DATA REGISTERS (DATA0-7x)

Read/Write

Reset Value: Undefined

7							0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DATA[7:0] is a message data byte. Up to eight such bytes may be part of a message. Writing to byte DATA7 initiates a transmit request and should always be done even when DATA7 is not part of the message.

BUFFER CONTROL/STATUS REGs. (BCSRx)

Read/Write

Reset Value: 00h

7							0
0	0	0	0	ACC	RDY	BUSY	LOCK

Bit 3 = **ACC** *Acceptance Code*
— Read Only
Set by hardware with the id of the highest priority filter which accepted the message stored in the buffer.
ACC = 0: Match for Filter/Mask0. Possible match for Filter/Mask1.
ACC = 1: No match for Filter/Mask0 and match for Filter/Mask1.

Reset by hardware when either RDY or RXIF gets reset.

Bit 2 = **RDY** *Message Ready*
— Read/Clear
Set by hardware to signal that a new error-free message is available (LOCK = 0) or that a transmission request is pending (LOCK = 1).
Cleared by software when LOCK = 0 to release the buffer and to clear the corresponding RXIF bit in the Interrupt Status Register.
Cleared by hardware when LOCK = 1 to indicate that the transmission request has been serviced or cancelled.

Bit 1 = **BUSY** *Busy Buffer*
— Read Only
Set by hardware when the buffer is being filled (LOCK = 0) or emptied (LOCK = 1).
Reset by hardware when the buffer is not accessed by the CAN core for transmission nor reception purposes.

Bit 0 = **LOCK** *Lock Buffer*
— Read/Set/Clear
Set by software to lock a buffer. No more message can be received into the buffer thus preserving its content and making it available for transmission.
Cleared by software to make the buffer available for reception. Cancels any pending transmission request.
Cleared by hardware once a message has been successfully transmitted provided the early transmit interrupt mode is on. Left untouched otherwise.

Note that in order to prevent any message corruption or loss of context, LOCK cannot be set nor reset while BUSY is set. Trying to do so will result in LOCK not changing state.

CONTROLLER AREA NETWORK (Cont'd)

FILTER HIGH REGISTERS (FHRx)

Read/Write

Reset Value: Undefined

7							0
FIL11	FIL10	FIL9	FIL8	FIL7	FIL6	FIL5	FIL4

FIL[11:3] are the most significant 8 bits of a 12-bit message filter. The acceptance filter is compared bit by bit with the identifier and the RTR bit of the incoming message. If there is a match for the set of bits specified by the acceptance mask then the message is stored in a receive buffer.

FILTER LOW REGISTERS (FLRx)

Read/Write

Reset Value: Undefined

7							0
FIL3	FIL2	FIL1	FIL0	0	0	0	0

FIL[3:0] are the least significant 4 bits of a 12-bit message filter.

MASK HIGH REGISTERS (MHRx)

Read/Write

Reset Value: Undefined

7							0
MSK11	MSK10	MSK9	MSK8	MSK7	MSK6	MSK5	MSK4

MSK[11:3] are the most significant 8 bits of a 12-bit message mask. The acceptance mask defines which bits of the acceptance filter should match the identifier and the RTR bit of the incoming message.
MSK_i = 0: don't care.
MSK_i = 1: match required.

MASK LOW REGISTERS (MLRx)

Read/Write

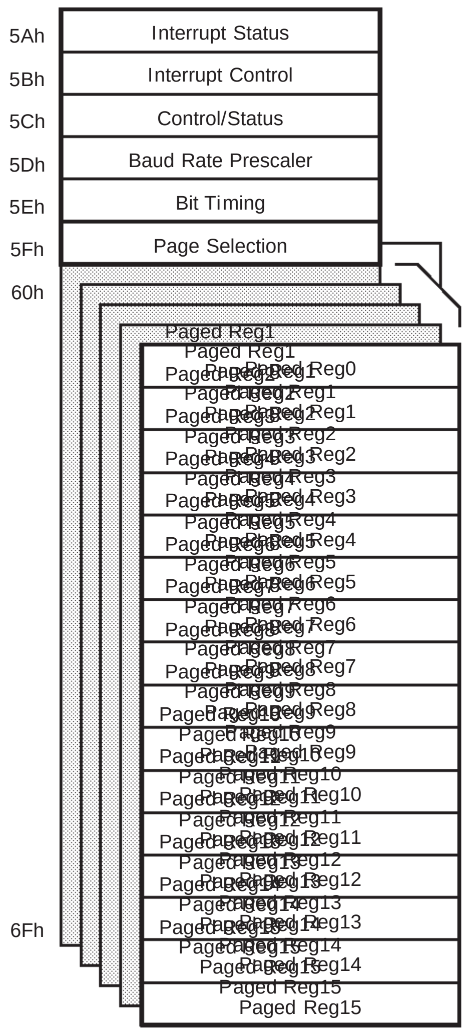
Reset Value: Undefined

7							0
MSK3	MSK2	MSK1	MSK0	0	0	0	0

MSK[3:0] are the least significant 4 bits of a 12-bit message mask.

CONTROLLER AREA NETWORK (Cont'd)

Figure 61. CAN Register Map



CONTROLLER AREA NETWORK (Cont'd)

Figure 62. Page Maps

	PAGE 0	PAGE 1	PAGE 2	PAGE 3	PAGE 4
60h	LIDHR	IDHR1	IDHR2	IDHR3	FHR0
61h	LIDLR	IDLR1	IDLR2	IDLR3	FLR0
62h	Reserved	DATA01	DATA02	DATA03	MHR0
63h		DATA11	DATA12	DATA13	MLR0
64h		DATA21	DATA22	DATA23	FHR1
65h		DATA31	DATA32	DATA33	FLR1
66h		DATA41	DATA42	DATA43	MHR1
67h		DATA51	DATA52	DATA53	MLR1
68h		DATA61	DATA62	DATA63	Reserved
69h		DATA71	DATA72	DATA73	
6Ah		Reserved	Reserved	Reserved	
6Bh					
6Ch					
6Dh	TSTR				
6Eh	TECR				
6Fh	RECR	BCSR1	BCSR2	BCSR3	
	Diagnosis	Buffer 1	Buffer 2	Buffer 3	Acceptance Filters

CONTROLLER AREA NETWORK (Cont'd)

Table 22. CAN Register Map and Reset Values

Address (Hex.)	Page	Register Label	7	6	5	4	3	2	1	0
5A		**CANISR**	RXIF3	RXIF2	RXIF1	TXIF	SCIF	ORIF	TEIF	EPND
		Reset Value	0	0	0	0	0	0	0	0
5B		**CANICR**		ESCI	RXIE	TXIE	SCIE	ORIE	TEIE	ETX
		Reset Value	0	0	0	0	0	0	0	0
5C		**CANCSR**		BOFF	EPSV	SRTE	NRTX	FSYN	WKPS	RUN
		Reset Value	0	0	0	0	0	0	0	0
5D		**CANBRPR**	RJW1	RJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
		Reset Value	0	0	0	0	0	0	0	0
5E		**CANBTR**		BS22	BS21	BS20	BS13	BS12	BS11	BS10
		Reset Value	0	0	1	0	0	0	1	1
5F		**CANPSR**						PAGE2	PAGE1	PAGE0
		Reset Value	0	0	0	0	0	0	0	0
60	0	**CANLIDHR**	LID10	LID9	LID8	LID7	LID6	LID5	LID4	LID3
		Reset Value	x	x	x	x	x	x	x	x
	1 to 3	**CANIDHRx**	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
		Reset Value	x	x	x	x	x	x	x	x
60, 64	4	**CANFHRx**	FIL11	FIL10	FIL9	FIL8	FIL7	FIL6	FIL5	FIL4
		Reset Value	x	x	x	x	x	x	x	x
61	0	**CANLIDLR**	LID2	LID1	LID0	LRTR	LDLC3	LDLC2	LDLC1	LDLC0
		Reset Value	x	x	x	x	x	x	x	x
	1 to 3	**CANIDLRx**	ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
		Reset Value	x	x	x	x	x	x	x	x
61, 65	4	**CANFLRx**	FIL3	FIL2	FIL1	FIL0				
		Reset Value	x	x	x	x	0	0	0	0
62 to 69	1 to 3	**CANDRx**	MSB							LSB
		Reset Value	x	x	x	x	x	x	x	x
62, 66	4	**CANMHRx**	MSK11	MSK10	MSK9	MSK8	MSK7	MSK6	MSK5	MSK4
		Reset Value	x	x	x	x	x	x	x	x
63, 67	4	**CANMLRx**	MSK3	MSK2	MSK1	MSK0				
		Reset Value	x	x	x	x	0	0	0	0
6E	0	**CANTECR**	MSB							LSB
		Reset Value	0	0	0	0	0	0	0	0
6F		**CANRECR**	MSB							LSB
		Reset Value	0	0	0	0	0	0	0	0
	1 to 3	**CANBCSRx**					ACC	RDY	BUSY	LOCK
		Reset Value	0	0	0	0	0	0	0	0

5.9 8-BIT A/D CONVERTER (ADC)

5.9.1 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 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 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.

5.9.2 Main Features

- 8-bit conversion
- Up to 16 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 64.

5.9.3 Functional Description

5.9.3.1 Analog Power Supply

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

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

Figure 63. Recommended Ext. Connections

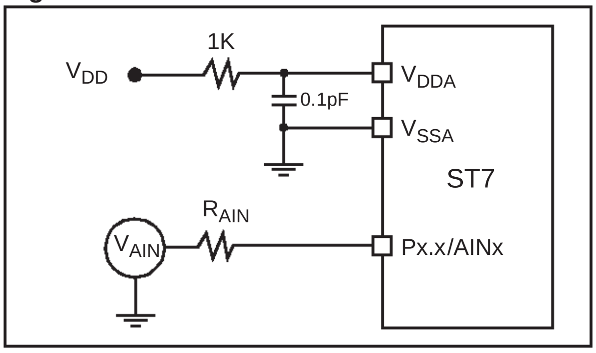
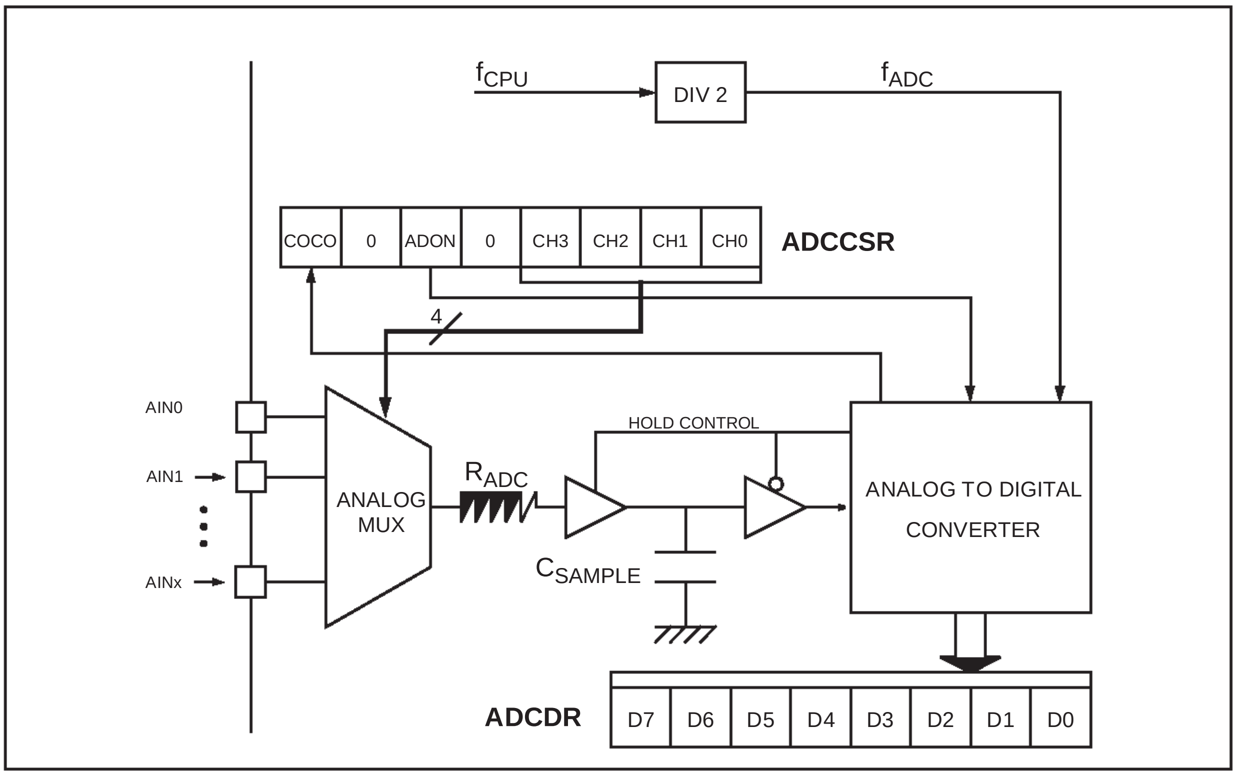


Figure 64. ADC Block Diagram



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

5.9.3.2 Digital A/D Conversion Result

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

If the input voltage (V_{AIN}) is greater than or equal to V_{DDA} (high-level voltage reference) then the conversion result in the DR register is FFh (full scale) without overflow indication.

If input voltage (V_{AIN}) is lower than or equal to V_{SSA} (low-level voltage reference) then the conversion result in the DR register is 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDR register. The accuracy of the conversion is described in the Electrical Characteristics Section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

5.9.3.3 A/D Conversion Phases

The A/D conversion is based on two conversion phases as shown in Figure 65:

- Sample capacitor loading
  [duration: t_{LOAD}]
  During this phase, the V_{AIN} input voltage to be measured is loaded into the C_{SAMPLE} sample capacitor.
- A/D conversion
  [duration: t_{CONV}]
  During this phase, the A/D conversion is computed (8 successive approximations cycles) and the C_{SAMPLE} sample capacitor is disconnected from the analog input pin to get the optimum A/D conversion accuracy.

While the ADC is on, these two phases are continuously repeated.

At the end of each conversion, the sample capacitor is kept loaded with the previous measurement load. The advantage of this behaviour is that it minimizes the current consumption on the analog pin in case of single input channel measurement.

5.9.3.4 Software Procedure

Refer to the control/status register (CSR) and data register (DR) in Section 5.9.6 for the bit definitions and to Figure 65 for the timings.

ADC Configuration

The total duration of the A/D conversion is 12 ADC clock periods ($1/f_{ADC}=2/f_{CPU}$).

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 CH[3:0] bits to assign the analog channel to convert.

ADC Conversion

In the CSR register:

– Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete

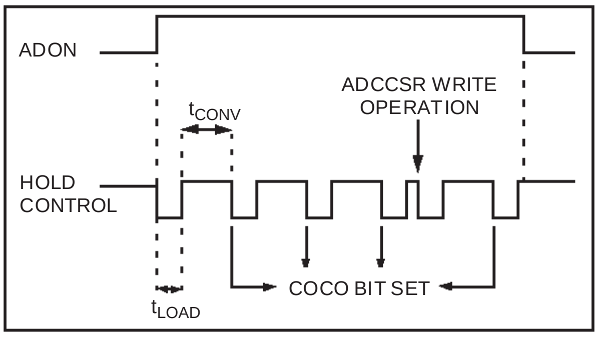
– The COCO bit is set by hardware.

– No interrupt is generated.

– The result is in the DR register and remains valid until the next conversion has ended.

A write to the CSR register (with ADON set) aborts the current conversion, resets the COCO bit and starts a new conversion.

Figure 65. ADC Conversion Timings



5.9.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions..

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilisation time before accurate conversions can be performed.

5.9.5 Interrupts

None

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

5.9.6 Register Description

CONTROL/STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
COCO	0	ADON	0	CH3	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

Bit 4 = **Reserved.** *must always be cleared.*

Bit 3:0 = **CH[3:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert.

Channel Pin*	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

***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
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *Analog Converted Value*
This register contains the converted analog value in the range 00h to FFh.

Note: Reading this register reset the COCO flag.

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

Table 23. ADC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	**ADCDR** Reset Value	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0	D1 0	D0 0
0071h	**ADCCSR Standard** Reset Value	COCO 0	0	ADON 0	0	0	CH2 0	CH1 0	CH0 0

6 INSTRUCTION SET

6.1 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 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.

Table 24. ST7 Addressing Mode Overview

Mode			Syntax	Destination	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
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+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	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

INSTRUCTION SET OVERVIEW (Cont'd)

6.1.1 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 (level 3)
RIM	Reset Interrupt Mask (level 0)
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

6.1.2 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

6.1.3 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.

6.1.4 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.

6.1.5 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.

INSTRUCTION SET OVERVIEW (Cont'd)

6.1.6 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 25. 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 Additions/Substractions 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

6.1.7 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 submodes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

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

INSTRUCTION SET OVERVIEW (Cont'd)

6.2 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 opcodes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes 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 SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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 . 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				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			I1	H	I0	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 Port B INT pin = 1	(no Port B Interrupts)								
JRIL	Jump if Port B INT pin = 0	(Port B interrupt)								
JRH	Jump if H = 1	H = 1 ?								
JRNH	Jump if H = 0	H = 0 ?								
JRM	Jump if I1:0 = 11	I1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?								
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 SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	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	reg	M						
		pop CC	CC	M	I1	H	I0	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	I1:0 = 10 (level 0)			1		0			
RLC	Rotate left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate right true C	C => A => C	reg, M					N	Z	C
RSP	Reset Stack Pointer	S = Max allowed								
SBC	Substract with Carry	A = A - M - C	A	M				N	Z	C
SCF	Set carry flag	C = 1								1
SIM	Disable Interrupts	I1:0 = 11 (level 3)			1		1			
SLA	Shift left Arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift left Logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift right Logic	0 => A => C	reg, M					0	Z	C
SRA	Shift right Arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Substraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for Neg & Zero	tnz lbl1						N	Z	
TRAP	S/W trap	S/W interrupt			1		1			
WFI	Wait for Interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

7 ELECTRICAL CHARACTERISTICS

7.1 PARAMETER CONDITIONS

Unless otherwise specified, all voltages are referred to V_{SS}.

7.1.1 Minimum and Maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambiant temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambiant temperature at T_A=25°C and T_A=$T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean$\pm 3\Sigma$).

7.1.2 Typical values

Unless otherwise specified, typical data are based on T_A=25°C, V_{DD}=5V (for the $4.5V \leq V_{DD} \leq 5.5V$ voltage range) and V_{DD}=3.3V (for the $3V \leq V_{DD} \leq 4V$ voltage range). They are given only as design guidelines and are not tested.

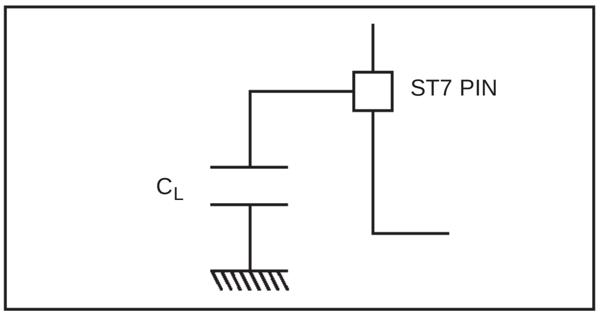
7.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

7.1.4 Loading capacitor

The loading conditions used for pin parameter measurement is shown in Figure 66.

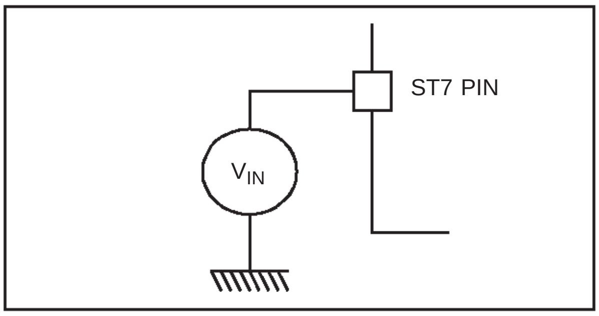
Figure 66. Pin loading conditions



7.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 67.

Figure 67. Pin input voltage



7.2 ABSOLUTE MAXIMUM RATINGS

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

7.2.1 Voltage Characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{DDA} - V_{SSA}	Analog reference voltage ($V_{DD} \geq V_{DDA}$)	6.5	
$\|\Delta V_{DDx}\|$ and $\|\Delta V_{SSx}\|$	Variations between different digital power pins	50	mV
$\|V_{SSA} - V_{SSx}\|$	Variations between digital and analog ground pins	50	
V_{IN}	Input voltage on V_{PP} pin	V_{SS}-0.3 to 13	V
	Input voltage on any other pin 1) & 2)	V_{SS}-0.3 to V_{DD}+0.3	
V_{DESD}	Electro-static discharge voltage	2000	

7.2.2 Current Characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) 3)	150	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) 3)	150	
I_{IO}	Output current sunk by any standard I/O and control pin	25	
	Output current sunk by any high sink I/O pin	50	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$ 2) & 4)	Injected current on V_{PP} pin	± 5	
	Injected current on $\overline{\text{RESET}}$ pin	± 5	
	Injected current on OSC1 and OSC2 pins	± 5	
	Injected current on any other pin 5) & 6)	± 5	
$\Sigma I_{INJ(PIN)}$ 2)	Total injected current (sum of all I/O and control pins) 5)	± 20	

7.2.3 Thermal Characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see Section 8.2 THERMAL CHARACTERISTICS)		

Notes:

1. Directly connecting the $\overline{\text{RESET}}$ and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7kΩ for $\overline{\text{RESET}}$, 10kΩ for I/Os). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.

2. When the current limitation is not possible, the V_{IN} absolute maximum rating must be respected, otherwise refer to $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$.

3. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.

4. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effect on analog part, care must be taken:
- Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
- Pure digital pins must have a negative injection less than 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.

5. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterisation with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

6. True open drain I/O port pins do not accept positive injection.

7.3 OPERATING CONDITIONS

7.3.1 General Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	Supply voltage	see Figure 68 and Figure 69	3.0	5.5	V
f_{OSC}	External clock frequency	V_{DD}≥3.5V (without EEPROM) V_{DD}≥4.5V (with EEPROM)	0 [1]	16	MHz
		V_{DD}≥3.0V	0 [1]	8	
T_A	Ambient temperature range	1 Suffix Version	0	70	°C
		6 Suffix Version	-40	85	
		7 Suffix Version	-40	105	
		3 Suffix Version	-40	125	

Figure 68. f_{OSC} Maximum Operating Frequency Versus V_{DD} Supply for devices without EEPROM [2]

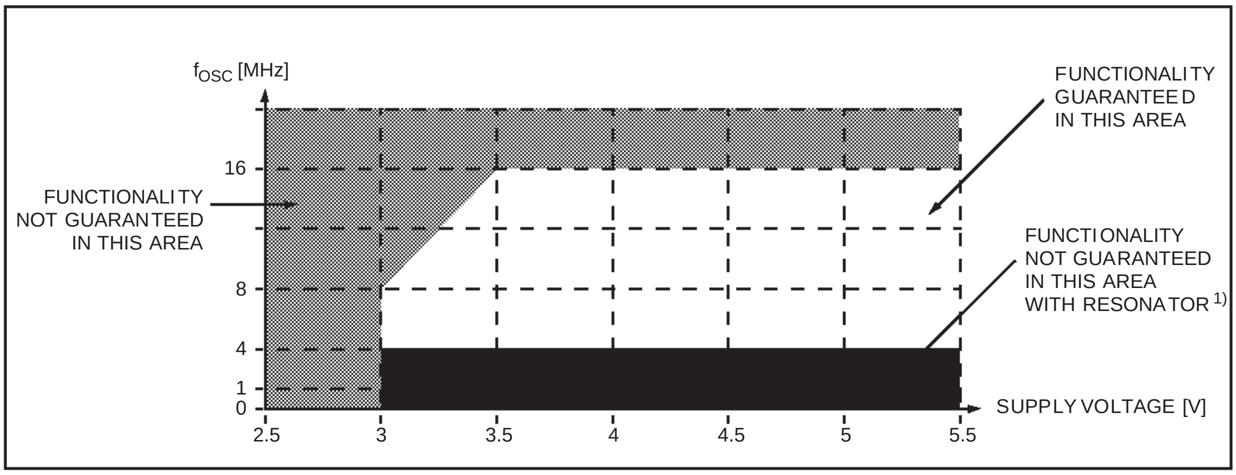
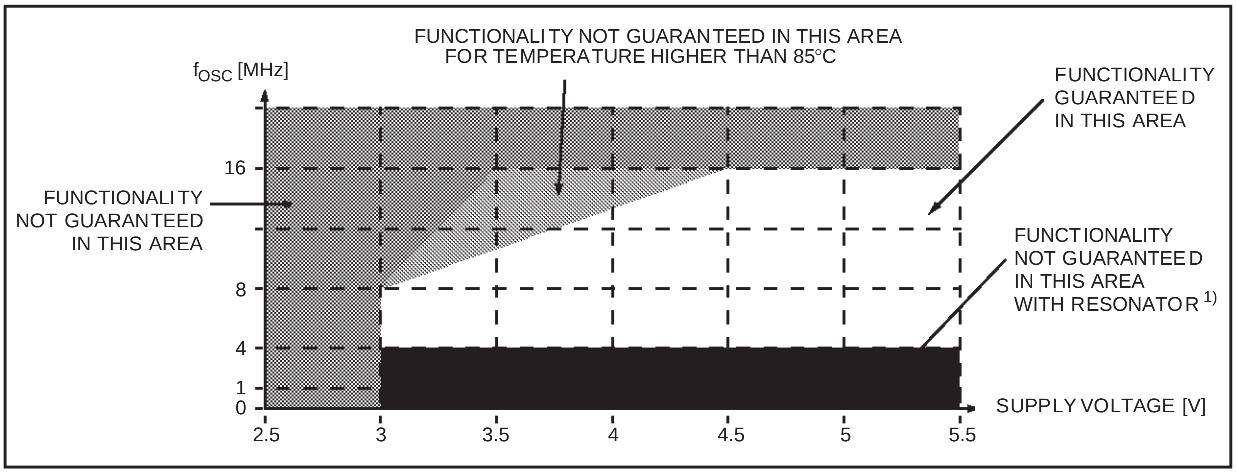


Figure 69. f_{OSC} Maximum Operating Frequency Versus V_{DD} Supply for device with EEPROM [2]



Notes:
1. Guaranteed by construction. A/D operation is not guaranteed below 1MHz.
2. Operating conditions with T_A=-40 to +125°C.

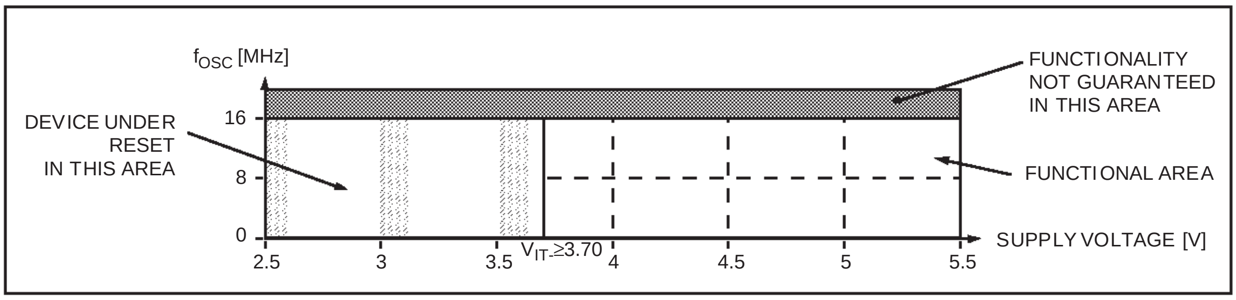
OPERATING CONDITIONS (Cont'd)

7.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A.

Symbol	Parameter	Conditions	Min	Typ [1)]	Max	Unit
V_{IT+}	Reset release threshold (V_{DD} rise)		3.95	4.15	4.35	V
V_{IT-}	Reset generation threshold (V_{DD} fall)		3.70	3.90	4.10	
V_{hys}	LVD voltage threshold hysteresis	V_{IT+}-V_{IT-}		250		mV
Vt_{POR}	V_{DD} rise time rate [2)]		0.02			V/ms
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}	Not detected by the LVD			40	ns

Figure 70. LVD Threshold Versus V_{DD} and f_{OSC} for ROM devices [2)]



Notes:

1. LVD typical data are based on T_A=25°C. They are given only as design guidelines and are not tested.
2. The minimum V_{DD} rise time rate is needed to insure a correct device power-on and LVD reset. Not tested in production.

7.4 SUPPLY CURRENT CHARACTERISTICS

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode for which the clock is stopped).

Symbol	Parameter	Conditions	Max	Unit
$\Delta I_{DD(\Delta Ta)}$	Supply current variation vs. temperature	Constant V_{DD} and f_{CPU}	10	%

7.4.1 RUN and SLOW Modes

Symbol	Parameter	Conditions		Typ [1]	Max [2]	Unit
I_{DD}	Supply current in RUN mode [3] (see Figure 71)	$4.5V \le V_{DD} \le 5.5V$	f_{OSC}=1MHz, f_{CPU}=500kHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=16MHz, f_{CPU}=8MHz	2.5 6.5 14.5	4 9 20	mA
	Supply current in SLOW mode [4] (see Figure 72)		f_{OSC}=1MHz, f_{CPU}=31.25kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD 1.8	TBD TBD 3.0	
	Supply current in RUN mode [3] (see Figure 71)	$3V \le V_{DD} \le 3.6V$	f_{OSC}=1MHz, f_{CPU}=500kHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=16MHz, f_{CPU}=8MHz	1.6 3.6 8	2.4 5.4 12	
	Supply current in SLOW mode [4] (see Figure 72)		f_{OSC}=1MHz, f_{CPU}=31.25kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD 1	TBD TBD 1.5	

Figure 71. Typical I_{DD} in RUN vs. f_{CPU}

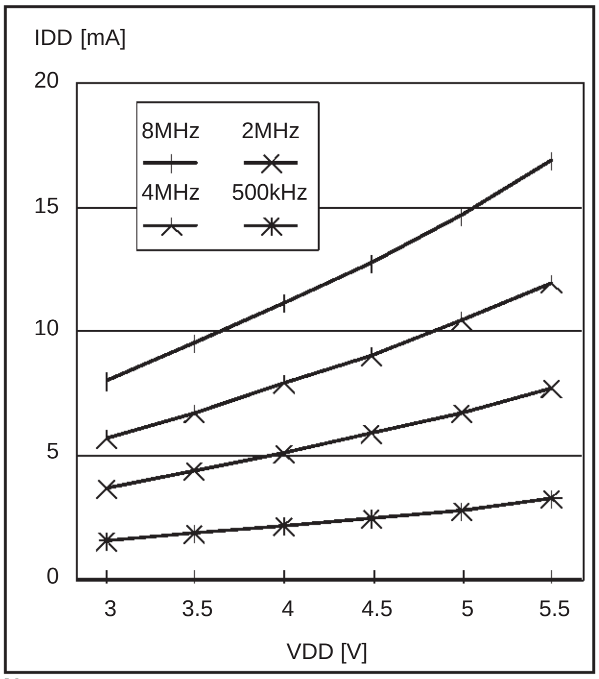
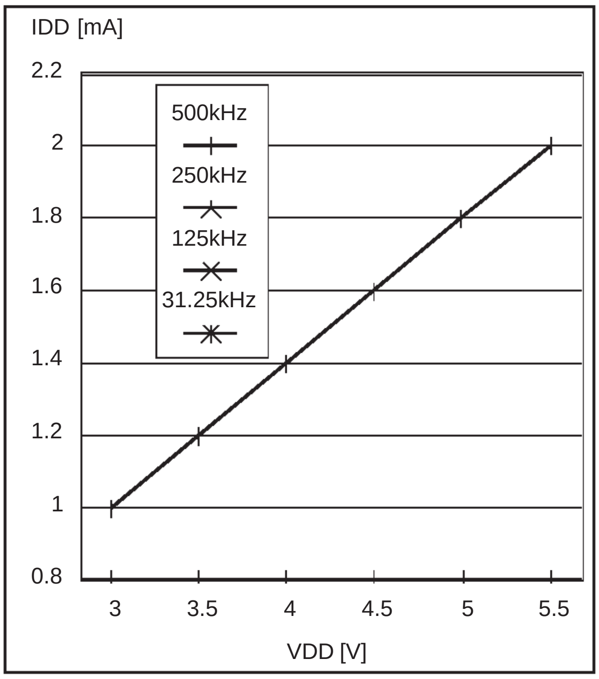


Figure 72. Typical I_{DD} in SLOW vs. f_{CPU}



Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \le V_{DD} \le 5.5V$ range) and V_{DD}=3.3V ($3V \le V_{DD} \le 3.6V$ range).
2. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
3. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSC1) driven by external square wave, LVD disabled.
4. SLOW mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSC1) driven by external square wave, LVD disabled.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

7.4.2 WAIT and SLOW WAIT Modes

Symbol	Parameter	Conditions		Typ [1)]	Max [2)]	Unit
I_{DD}	Supply current in WAIT mode [3)] (see Figure 73)	$4.5V \le V_{DD} \le 5.5V$	f_{OSC}=1MHz, f_{CPU}=500kHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=16MHz, f_{CPU}=8MHz	TBD TBD 5.2	TBD TBD 9	mA
	Supply current in SLOW WAIT mode [4)] (see Figure 74)		f_{OSC}=1MHz, f_{CPU}=31.25kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD 1.2	TBD TBD 2	
	Supply current in WAIT mode [3)] (see Figure 73)	$3V \le V_{DD} \le 3.6V$	f_{OSC}=1MHz, f_{CPU}=500kHz f_{OSC}=4MHz, f_{CPU}=2MHz f_{OSC}=16MHz, f_{CPU}=8MHz	TBD TBD 2.7	TBD TBD 4.5	
	Supply current in SLOW WAIT mode [4)] (see Figure 74)		f_{OSC}=1MHz, f_{CPU}=31.25kHz f_{OSC}=4MHz, f_{CPU}=125kHz f_{OSC}=16MHz, f_{CPU}=500kHz	TBD TBD 0.6	TBD TBD 1	

Figure 73. Typical I_{DD} in WAIT vs. f_{CPU}

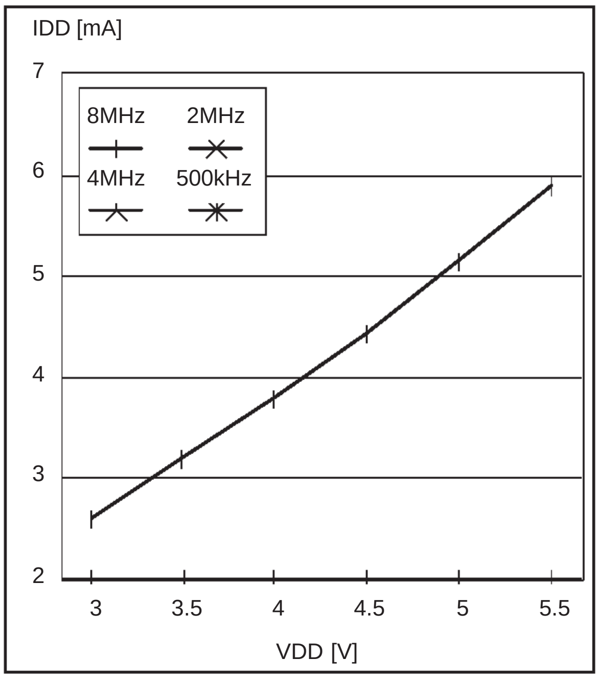
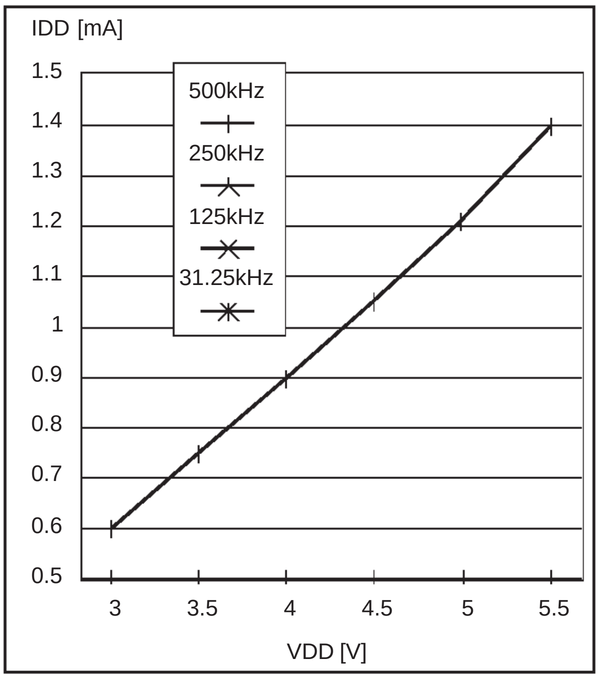


Figure 74. Typical I_{DD} in SLOW-WAIT vs. f_{CPU}



Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \le V_{DD} \le 5.5V$ range) and V_{DD}=3.3V ($3V \le V_{DD} \le 3.6V$ range).
2. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
3. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSC1) driven by external square wave, LVD disabled.
4. SLOW-WAIT mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSC1) driven by external square wave, LVD disabled.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

7.4.3 HALT and ACTIVE-HALT Modes

Symbol	Parameter	Conditions		Typ [1]	Max	Unit
I_{DD}	Supply current in HALT mode [2]	$3.0V \leq V_{DD} \leq 5.5V$ $-40°C \leq T_A \leq +125°C$		0	10	µA
	Supply current in ACTIVE-HALT mode [3]	$V_{DD}=5.5V$	$-40°C \leq T_A \leq +85°C$	100	150	
			$-40°C \leq T_A \leq +125°C$		TBD	
		$V_{DD}=3.6V$	$-40°C \leq T_A \leq +85°C$	TBD	TBD	
			$-40°C \leq T_A \leq +125°C$		TBD	

7.4.4 Supply and Clock Managers

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode).

Symbol	Parameter	Conditions	Typ [4]	Max [5]	Unit
$I_{DD(CK)}$	Supply current of resonator oscillator [6] & [7]		600	850	µA
$I_{DD(LVD)}$	LVD supply current	HALT mode	100	150	

7.4.5 On-Chip Peripheral

Symbol	Parameter	Conditions		Typ [4]	Max [5]	Unit
$I_{DD(ADC)}$	ADC supply current when converting	$f_{ADC}=4MHz$	$4.5V \leq V_{DD} \leq 5.5V$		1	mA

Notes:

1. Typical data are based on $T_A=25°C$.
2. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), LVD disabled.
3. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled.
4. Typical data are based on $T_A=25°C$, $V_{DD}=5V$.
5. Data based on characterization results, not tested in production.
6. Data based on characterization results done with the typical external components, not tested in production.
7. As the oscillator is based on a current source, the consumption does not depend on the voltage.

7.5 CLOCK AND TIMING CHARACTERISTICS

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A.

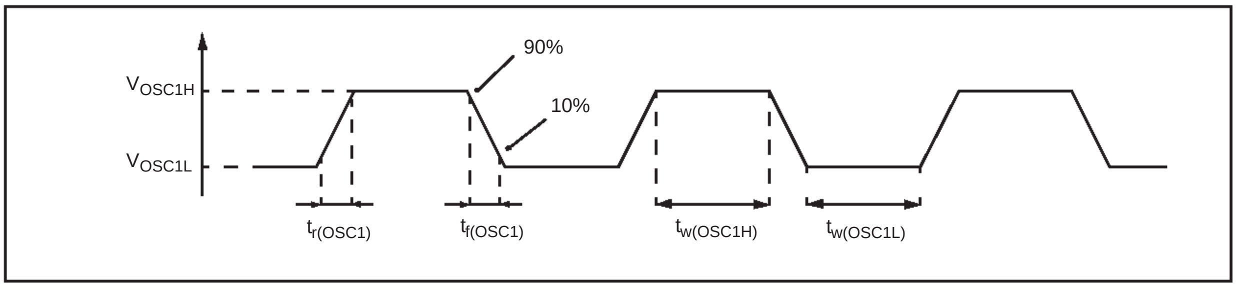
7.5.1 General Timings

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
$t_{c(INST)}$	Instruction cycle time		2	4	12	t_{CPU}
		f_{CPU}=8MHz	250	500	1500	ns
$t_{v(IT)}$	Interrupt reaction time [2] $t_{v(IT)} = \Delta t_{c(INST)} + 10$		10		22	t_{CPU}
		f_{CPU}=8MHz	1.25		2.75	μs

7.5.2 External Clock Source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage	see Figure 75	$0.7xV_{DD}$		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		$0.3xV_{DD}$	
$t_{w(OSC1H)}$ $t_{w(OSC1L)}$	OSC1 high or low time [3]		15			ns
$t_{r(OSC1)}$ $t_{f(OSC1)}$	OSC1 rise or fall time [3]				15	
R_{OBP}	Oscillator bypass exteranl resistor			1		kΩ

Figure 75. Typical Application with an External Clock Source



7.5.3 Crystal and Ceramic Resonator Oscillators

Symbol	Parameter	Conditions	Min	Max	Unit
f_{OSC}	Oscillator Frequency		4	16	MHz
C_{L1}, C_{L2}	Load capacitance [4]	R_S=100Ω [5]	12	21	kΩ
t_{START}	Oscillator start-up time	Depends on resonator quality. A typical value is 10ms			

Notes:

1. Data based on typical application software.
2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.
3. Data based on design simulation and/or technology characteristics, not tested in production.
4. C_{L1} (resp.C_{L2}) is load capacitance on OSC1 (resp. OSC2) pin.
5. R_S is the equivalent serial resistance of the crystal or ceramic resonator.

7.6 MEMORY CHARACTERISTICS

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

7.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1)]	HALT mode (or RESET)	1.6			V

7.6.2 EEPROM Data Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{prog}	Programming time (for 1 up to 16 bytes at a time)	$-40°C \leq T_A \leq +85°C$			10	ms
		$-40°C \leq T_A \leq +125°C$			15	
t_{ret}	Data retention [3)]	T_A=+55°C [2)]	20			Years
N_{RW}	Write erase cycles [3)]	T_A=+25°C	300 000			Cycles

7.6.3 EPROM Program Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
W_{ERASE}	UV lamp	Lamp wavelength 2537Å		15		Watt.sec /cm^2
t_{erase}	Erase Time [4)]	UV lamp is placed 1 inch from the device window without any interposed filters	15		20	min
t_{ret}	Data retention [3)]	T_A=+55°C [2)]	20			years

Notes:

1. Minimum V_{DD} supply voltage without losing data stored into RAM (in in HALT mode or under RESET) or into hardware registers (only in HALT mode). Guaranteed by construction, not tested in production.
2. The data retention time increase when the T_A decreases.
3. Data based on reliability test results and monitored in production.
4. Data given only as guidelines.

7.7 ESD PIN PROTECTION STRATEGY

To protect an integrated circuit against Electro-Static Discharge the stress must be controlled to prevent degradation or destruction of the circuit elements. The stress generally affects the circuit elements which are connected to the pads but can also affect the internal devices when the supply pads receive the stress. The elements to be protected must not receive excessive current, voltage or heating within their structure.

An ESD network combines the different input and output ESD protections. This network works, by allowing safe discharge paths for the pins subjected to ESD stress. Two critical ESD stress cases are presented in Figure 76 and Figure 77 for standard pins and in Figure 78 and Figure 79 for true open drain pins.

Standard Pin Protection

To protect the output structure the following elements are added:

– A diode to V_{DD} (3a) and a diode from V_{SS} (3b)
– A protection device between V_{DD} and V_{SS} (4)

To protect the input structure the following elements are added:

– A resistor in series with the pad (1)
– A diode to V_{DD} (2a) and a diode from V_{SS} (2b)
– A protection device between V_{DD} and V_{SS} (4)

Figure 76. Positive Stress on a Standard Pad vs. V_{SS}

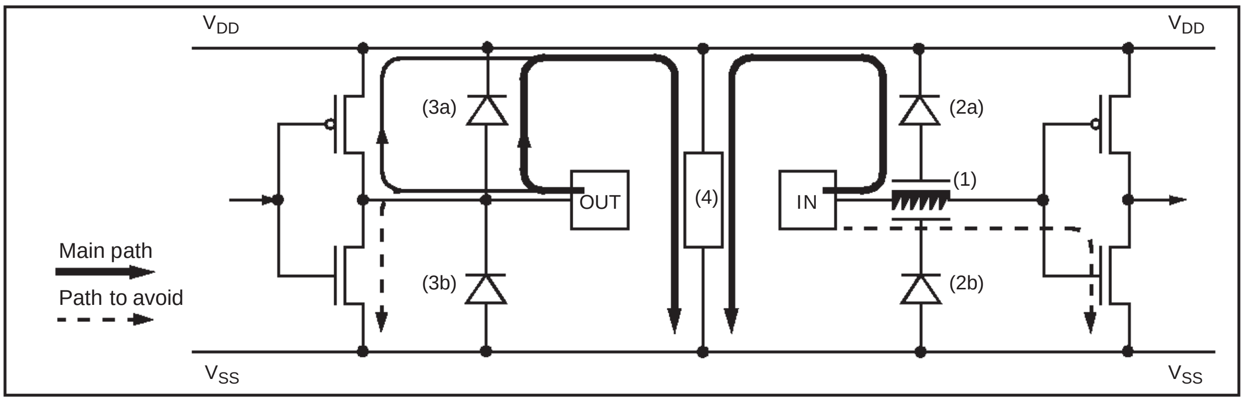
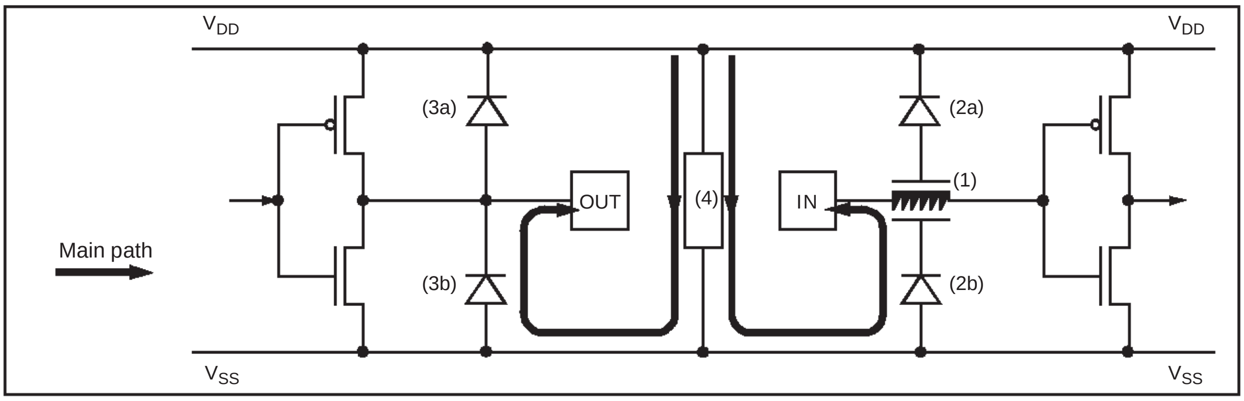


Figure 77. Negative Stress on a Standard Pad vs. V_{DD}



ESD PIN PROTECTION STRATEGY (Cont'd)

True Open Drain Pin Protection

The centralized protection (4) is not involved in the discharge of the ESD stresses applied to true open drain pads due to the fact that a P-Buffer and diode to V_{DD} are not implemented. An additional local protection between the pad and V_{SS} (5a & 5b) is implemented to completly absorb the positive ESD discharge.

Figure 78. Positive Stress on a True Open Drain Pad vs. V_{SS}

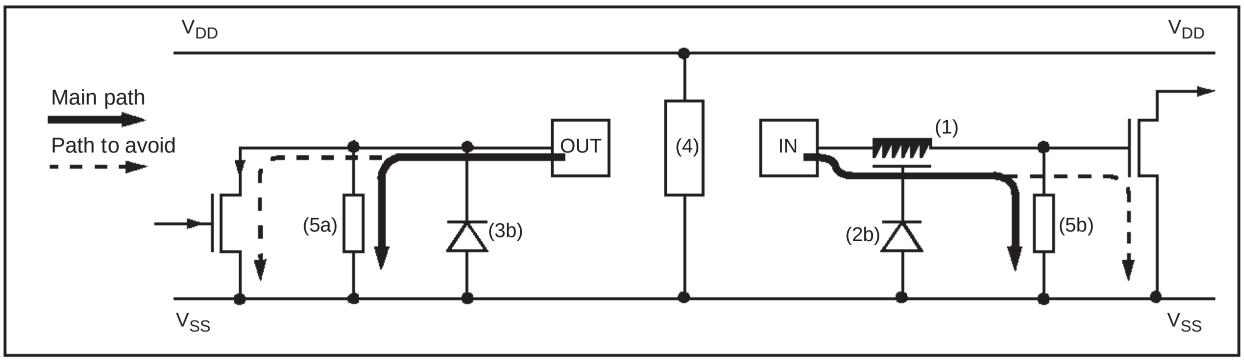
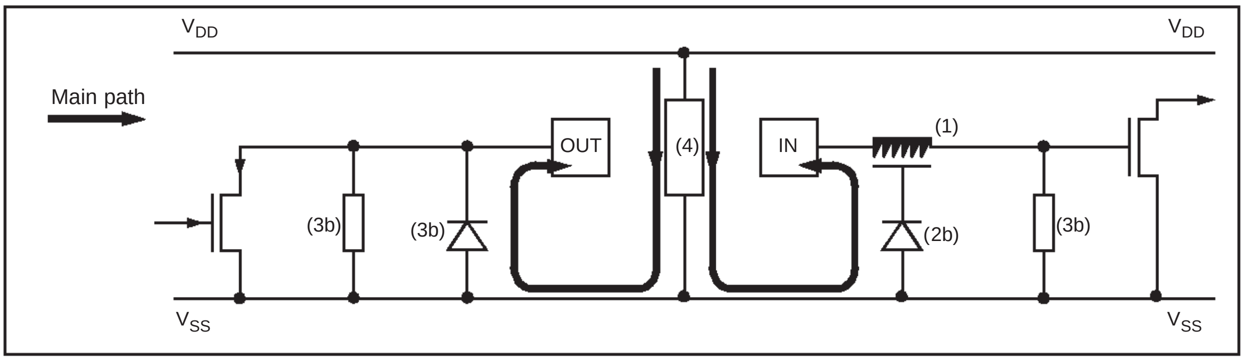


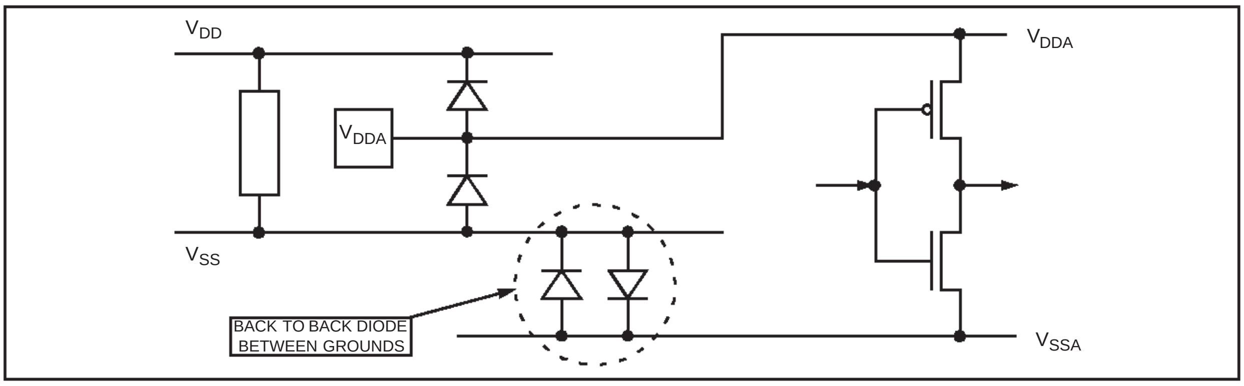
Figure 79. Negative Stress on a True Open Drain Pad vs. V_{DD}



Multisupply Configuration

When several types of ground (V_{SS}, V_{SSA}, ...) and power supply (V_{DD}, V_{DDA}, ...) are available for any reason (better noise immunity...), the structure shown in Figure 80 is implemented to protect the device against ESD.

Figure 80. Multisupply Configuration



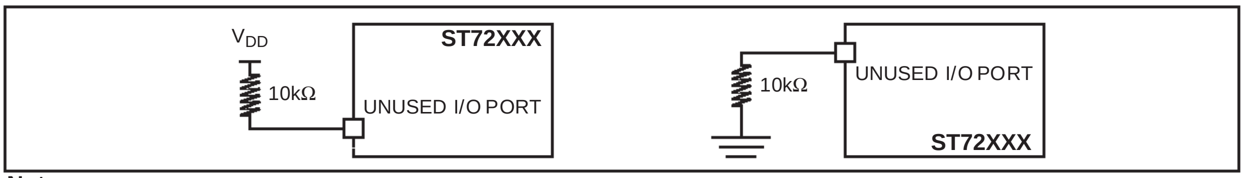
7.8 I/O PORT PIN CHARACTERISTICS

7.8.1 General Characteristics

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage [2]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$				±1	µA
I_S	Static current consumption [4]	Floating input mode				200	
R_{PU}	Weak pull-up equivalent resistor [5]	$V_{IN}=V_{SS}$	$V_{DD}=5V$	60		240	kΩ
C_{IO}	IO pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time [2]	$C_L=50pF$ Between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time [2]				25		
$t_{w(IT)in}$	External interrupt pulse time [6]			1			t_{CPU}

Figure 81. Two typical Applications with unused I/O Pin



Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.
2. Data based on characterization results, not tested in production.
3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
4. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see Figure 81). Data based on design simulation and/or technology characteristics, not tested in production.
5. The R_{PU} pull-up equivalent resistor is based on a resistive transistor. This data is based on characterization results, not tested in production.
6. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

I/O PORT PIN CHARACTERISTICS (Cont'd)

7.8.2 Output Driving Current

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} 1)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 82)	V_{DD}=5V	I_{IO}=+5mA		1.3	V
			I_{IO}=+2mA		0.4	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 83)		I_{IO}=+20mA		1.3	
			I_{IO}=+8mA		0.4	
V_{OH} 2)	Output high level voltage for an I/O pin when 8 pins are sourced at same time (see Figure 84)		I_{IO}=-5mA	V_{DD}-2.0		
			I_{IO}=-2mA	V_{DD}-0.8		

Figure 82. Typical V_{OL} at V_{DD}=5V (standard)

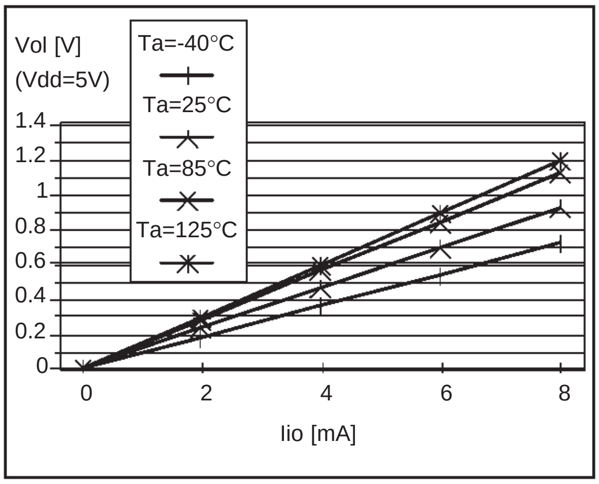


Figure 84. Typical V_{DD}-V_{OH} at V_{DD}=5V

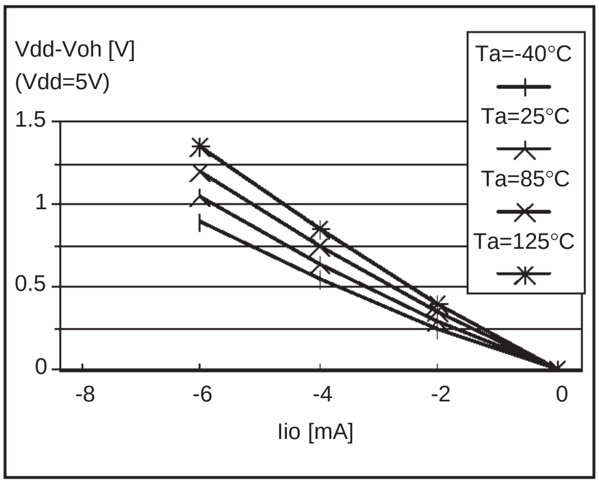
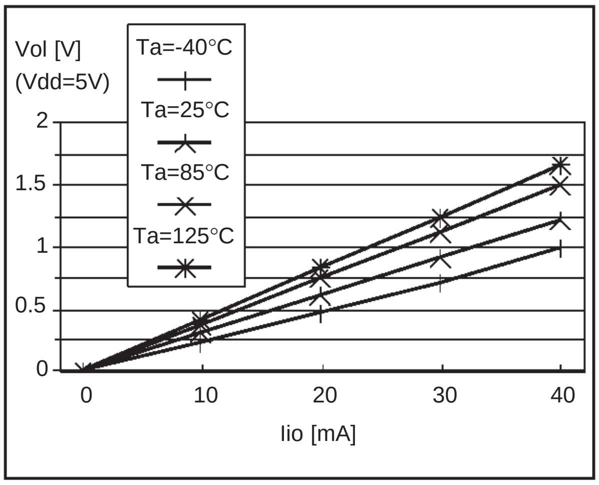


Figure 83. Typical V_{OL} at V_{DD}=5V (high-sink)



Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 7.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

2. The I_{IO} current sourced must always respect the absolute maximum rating specified in Section 7.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

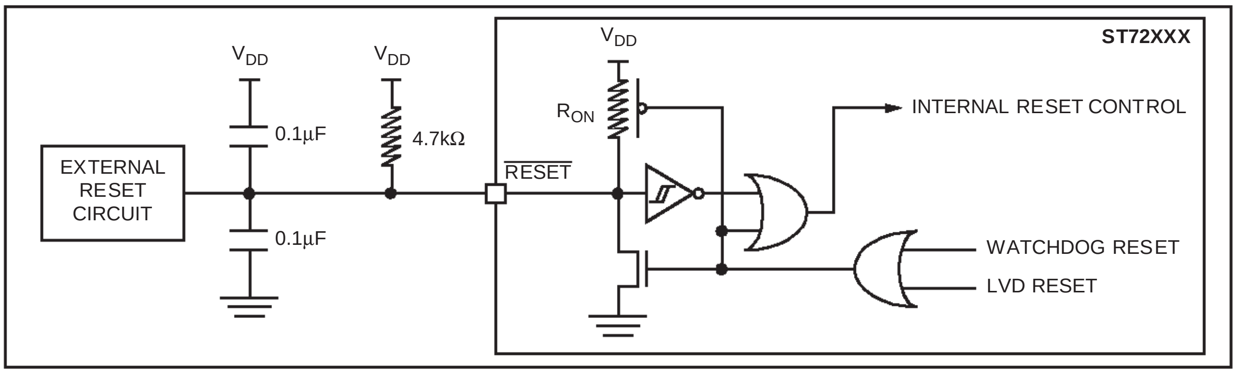
7.9 CONTROL PIN CHARACTERISTICS

7.9.1 Asynchronous $\overline{\text{RESET}}$ Pin

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage [2]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]				400		mV
R_{ON}	Weak pull-up equivalent resistor [4]	V_{IN}=V_{SS}	V_{DD}=5V	20		60	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or watchdog reset sources			6 30		$1/f_{SFOSC}$ µs
$t_{h(RSTL)in}$	External reset pulse hold time			20			µs

Figure 85. Typical Application with $\overline{\text{RESET}}$ pin [5]

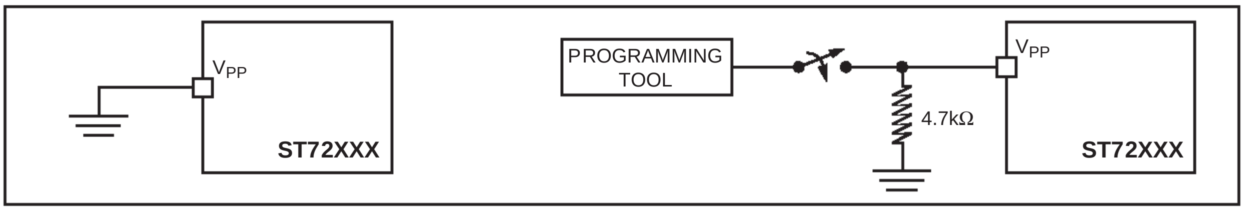


7.9.2 V_{PP} Pin

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage [6]		V_{SS}	0.2	V
V_{IH}	Input high level voltage [6]		V_{DD}-0.1	12.6	

Figure 86. Two typical Applications with V_{PP} Pin [7]



Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.
2. Data based on characterization results, not tested in production.
3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
4. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. This data is based on characterization results, not tested in production.
5. The reset network protects the device against parasitic resets, especially in a noisy environment.
6. Data based on design simulation and/or technology characteristics, not tested in production.
7. When the in-circuit programming mode is not required by the application V_{PP} pin must be tied to V_{SS}.

7.10 TIMER PERIPHERAL CHARACTERISTICS

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (outpu compare, input capture, external clock, PWM output...).

7.10.1 Watchdog Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(WDG)}$	Watchdog time-out duration		12,288		786,432	t_{CPU}
		f_{CPU}=8MHz	1.54		98.3	ms

7.10.2 8-Bit PWM-ART Auto-Reload Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{res(PWM)}$	PWM resolution time		1			t_{CPU}
		f_{CPU}=8MHz	125			ns
f_{EXT}	ART external clock frequency		0		f_{CPU}/2	MHz
f_{PWM}	PWM repetition rate		0		f_{CPU}/2	
Res_{PWM}	PWM resolution				8	bit
V_{OS}	PWM/DAC output step voltage	V_{DD}=5V, Res=8-bits		20		mV

7.10.3 16-Bit Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(ICAP)in}$	Input capture pulse time		1			t_{CPU}
$t_{res(PWM)}$	PWM resolution time		2			t_{CPU}
		f_{CPU}=8MHz	250			ns
f_{EXT}	Timer external clock frequency		0		f_{CPU}/4	MHz
f_{PWM}	PWM repetition rate		0		f_{CPU}/4	MHz
Res_{PWM}	PWM resolution				16	bit

7.11 COMUNICATION INTERFACE CHARACTERISTICS

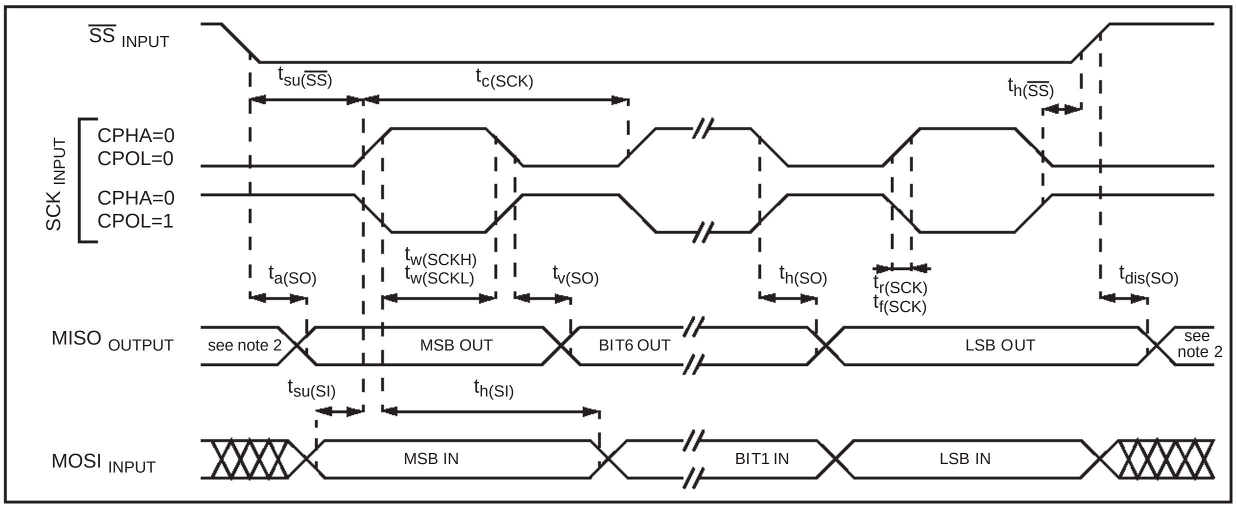
7.11.1 SPI - Serial Peripheral Interface

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU}=8MHz	f_{CPU}/128 0.0625	f_{CPU}/4 2	MHz
		Slave f_{CPU}=8MHz	0	f_{CPU}/2 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$	$\overline{SS}$ setup time	Slave	120		ns
$t_{h(\overline{SS})}$	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$	Data output access time	Slave	0	120	
$t_{dis(SO)}$	Data output disable time	Slave		240	
$t_{v(SO)}$	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$	Data output hold time		0		
$t_{v(MO)}$	Data output valid time	Master (before capture edge)	0.25		t_{CPU}
$t_{h(MO)}$	Data output hold time		0.25		

Figure 87. SPI Slave Timing Diagram with CPHA=0 3)



Notes:

1. Data based on design simulation and/or characterisation results, not tested in production.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.
3. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

COMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

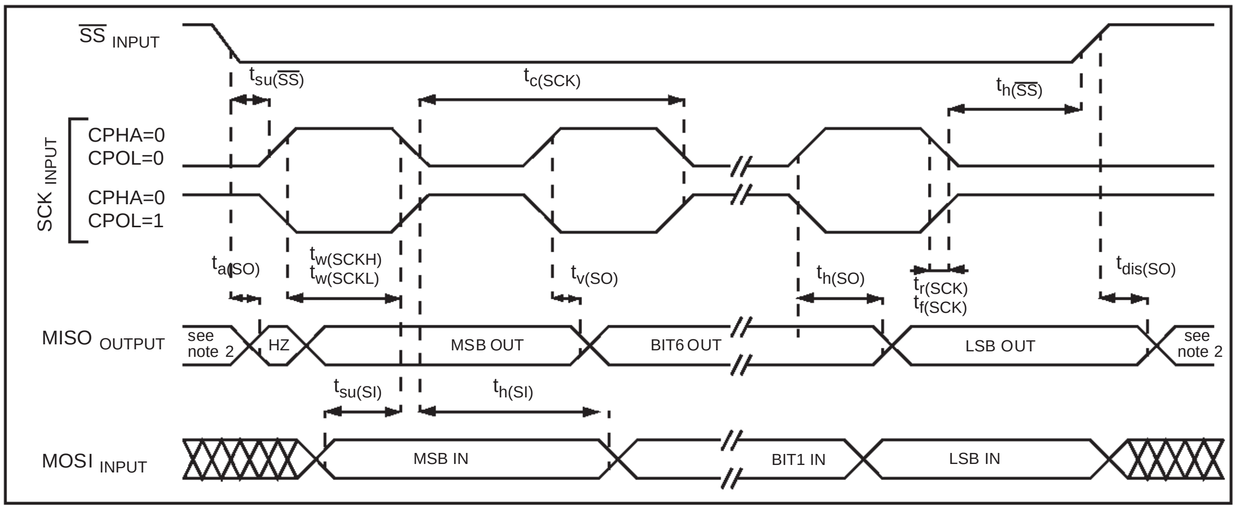


Figure 88. SPI Slave Timing Diagram with CPHA=1[1)]

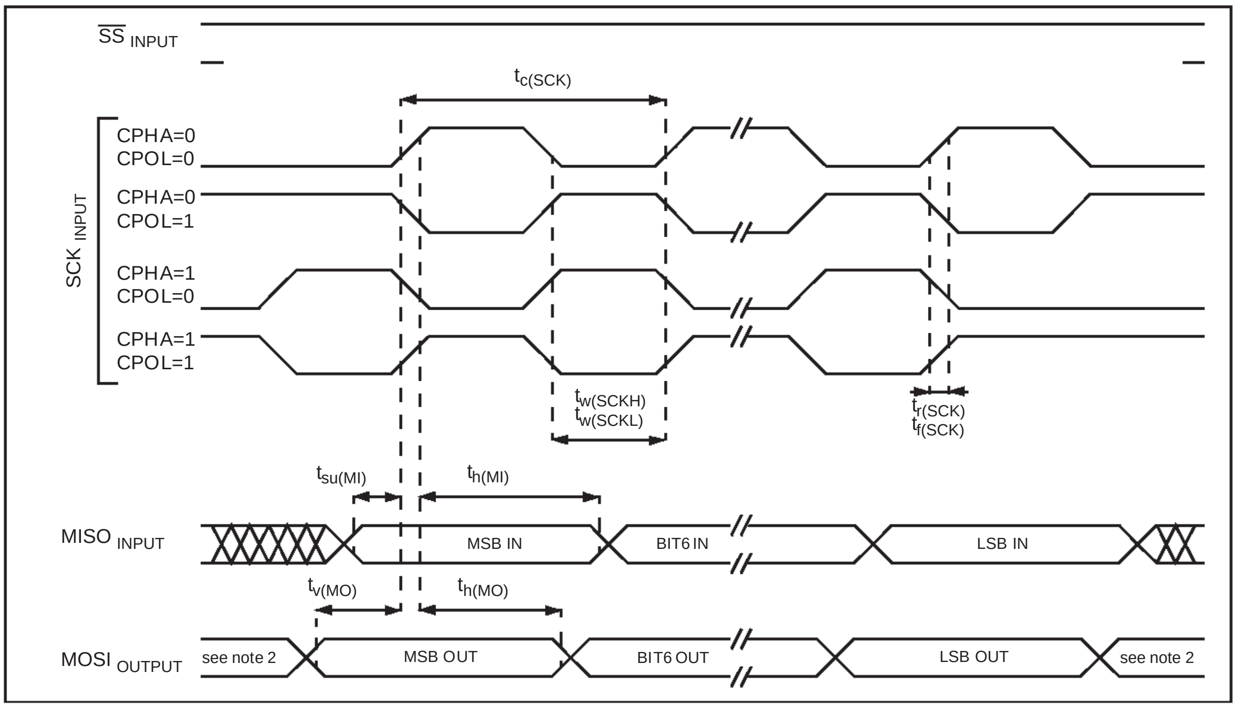


Figure 89. SPI Master Timing Diagram [1)]

Notes:

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

COMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

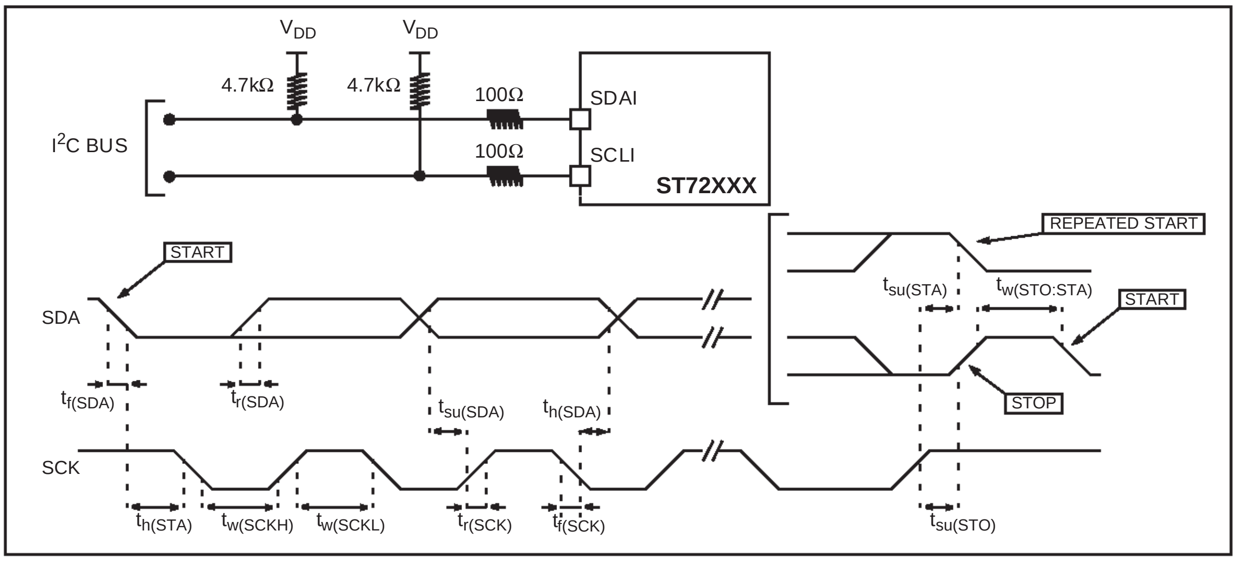
7.11.2 I²C - Inter IC Control Interface

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (SDAI and SCLI). The ST7 I²C interface fulfills the characteristics of the Standard I²C communication protocol described in the following table.

Symbol	Parameter	Standard mode I²C		Fast mode I²C		Unit
		Min [1)]	Max [1)]	Min [1)]	Max [1)]	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0 [3)]		0 [2)]	900 [3)]	
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	$20+0.1C_b$	300	
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300	$20+0.1C_b$	300	
$t_{h(STA)}$	START condition hold time	4.0		0.6		µs
$t_{su(STA)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(STO)}$	STOP condition setup time	4.0		0.6		ns
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		ms
C_b	Capacitive load for each bus line		400		400	pF

Figure 90. Typical Application with I²C Bus and Timing Diagram [4)]



Notes:

1. Data based on standard I²C protocol requirement, not tested in production.
2. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.
3. 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.
4. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

COMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

7.11.3 SCI - Serial Comunication Interface

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (RDI and TDO).

Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8MHz	~0.16%	Conventional Mode			Hz
				TR (or RR)=64, PR=13	300	~300.48	
				TR (or RR)=16, PR=13	1200	~1201.92	
				TR (or RR)= 8, PR=13	2400	~2403.84	
				TR (or RR)= 4, PR=13	4800	~4807.69	
				TR (or RR)= 2, PR=13	9600	~9615.38	
				TR (or RR)= 8, PR= 3	10400	~10416.67	
				TR (or RR)= 1, PR=13	19200	~19230.77	
				Extended Mode ETPR (or ERPR) = 13	38400	~38461.54	
			~0.79%	Extended Mode ETPR (or ERPR) = 35	14400	~14285.71	

7.11.4 CAN - Controller Area Network Interface

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.
Refer to I/O port characteristics for more details on the input/output alternate function characteristics (CANTX and CANRX).

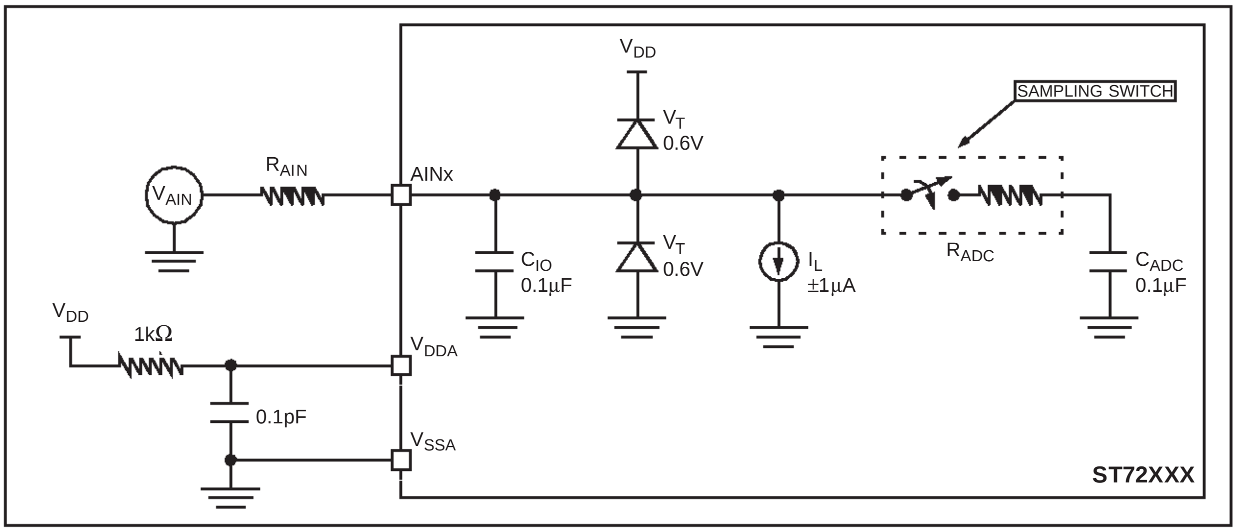
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{p(RX:TX)}$	CAN controller propagation time				60	ns

7.12 8-BIT ADC CHARACTERISTICS

Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ [1)]	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion range voltage [2)]		V_{SSA}		V_{DDA}	V
R_{AIN}	External input resistor				15 [3)]	kΩ
R_{ADC}	Internal input resistor			1.5		kΩ
C_{ADC}	Internal sample and hold capacitor			6		pF
t_{STAB}	Stabilization time after ADC enable	f_{CPU}=8MHz, f_{ADC}=4MHz		0 [4)]		µs
t_{LOAD}	Sample capacitor loading time			1 4		µs 1/f_{ADC}
t_{CONV}	Hold conversion time			2.250 9		µs 1/f_{ADC}

Figure 91. Typical Application with ADC



Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refer to V_{DD} and V_{SS}.

3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.

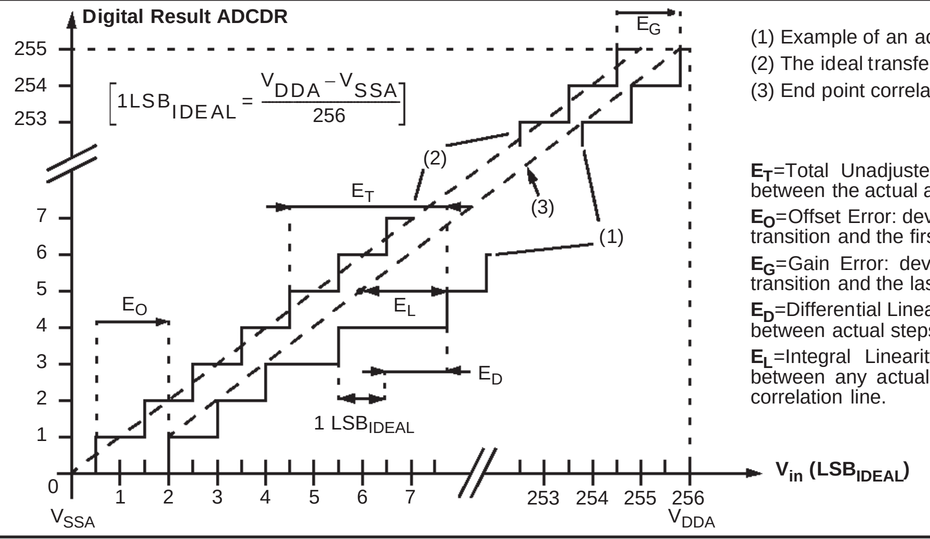
4. The stabilization time of the AD converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

COMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

ADC Accuracy with V_{DD}=5.0V

Symbol	Parameter	Conditions	Min	Max	Unit
$\lvert E_T \rvert$	Total unadjusted error [2)]	f_{CPU}=8MHz, f_{ADC}=4MHz [1)]		1.5	LSB
E_O	Offset error [2)]		-1	1	
E_G	Gain Error [2)]		-0.5	0.5	
$\lvert E_D \rvert$	Differential linearity error [2)]			1	
$\lvert E_L \rvert$	Integral linearity error [2)]			1	

Figure 92. ADC Accuracy Characteristics



(1) Example of an actual transfer curve
(2) The ideal transfer curve
(3) End point correlation line

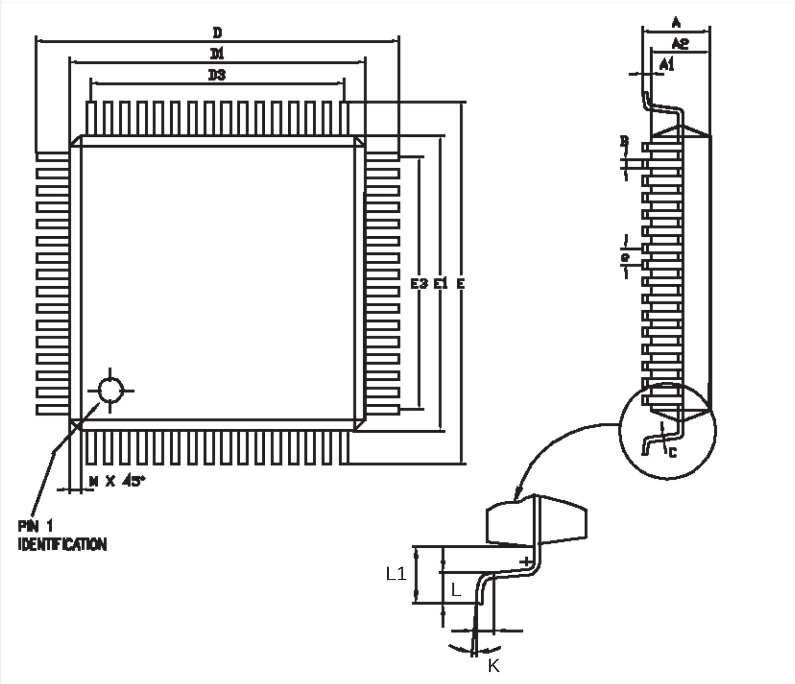
E_T=Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.
E_O=Offset Error: deviation between the first actual transition and the first ideal one.
E_G=Gain Error: deviation between the last ideal transition and the last actual one.
E_D=Differential Linearity Error: maximum deviation between actual steps and the ideal one.
E_L=Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

Notes:
1. Data based on characterization results over the whole temperature range, monitored in production.
2. ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 1 LSB for each 10KΩ increase of the external analog source impedance. This effect on the ADC accuracy has been observed under worst-case conditions for injection:
- negative injection
- injection to an Input with analog capability, adjacent to the enabled Analog Input
- at 5V V_{DD} supply, and worst case temperature.

8 PACKAGE CHARACTERISTICS

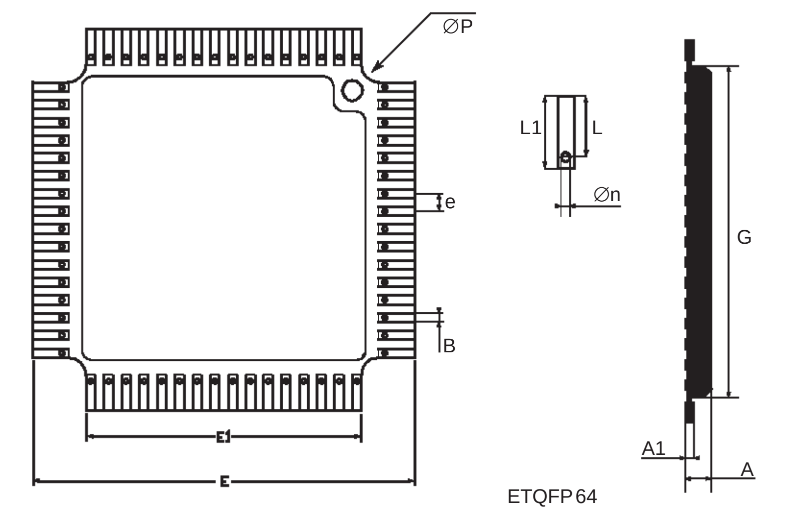
8.1 PACKAGE MECHANICAL DATA

Figure 93. 64-Pin Thin Quad Flat Package



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.30	0.37	0.45	0.012	0.015	0.018
C	0.09		0.20	0.004		0.008
D		16.00			0.630	
D1		14.00			0.551	
D3		12.00			0.472	
E		16.00			0.630	
E1		14.00			0.551	
E3		12.00			0.472	
e		0.80			0.031	
K	0°	3.5°	7°			
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	64		ND	16	NE	16

Figure 94. 64-Pin Epoxy Thin Quad Flat Package



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			2.40			0.095
A1			0.60			0.024
B	0.25	0.38	0.50	0.010	0.015	0.020
E	15.80	16.00	16.20	0.622	0.630	0.638
E1	12.20	12.35	12.50	0.480	0.486	0.492
e		0.80			0.031	
G		13.10			0.515	
L	0.50			0.020		
L1	1.10			0.043		
∅n			0.35			0.013
∅P			1.10			0.043
	Number of Pins					
N	64 (4x16)					

Note: “ QUALIFICATION OR VOLUME PRODUCTION OF DEVICES USING EPOXY PACKAGES (ESO/EDIL/EQFP) IS NOT AUTHORIZED It is expressly specified that qualification and/or volume production of devices using the package E.... in any applications is not authorized. Usage in any application is strictly restricted to development purpose. Similar devices are available in plastic package mechanically compatible to the epoxy package for qualification and volume production.”

8.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) TQFP64	60	°C/W
P_D	Power dissipation [1)]	500	mW
T_{Jmax}	Maximum junction temperature [2)]	150	°C

Notes:
1. The power dissipation is obtained from the formula $P_D=P_{INT}+P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD}xV_{DD}$) and P_{PORT} is the port power dissipation determined by the user.
2. The average chip-junction temperature can be obtained from the formula $T_J = T_A + P_D$ x RthJA.

8.3 SOLDERING AND GLUEABILITY INFORMATION

Recommended soldering information given only as design guidelines.

Figure 95. Recommended Wave Soldering Profile (with 37% Sn and 63% Pb)

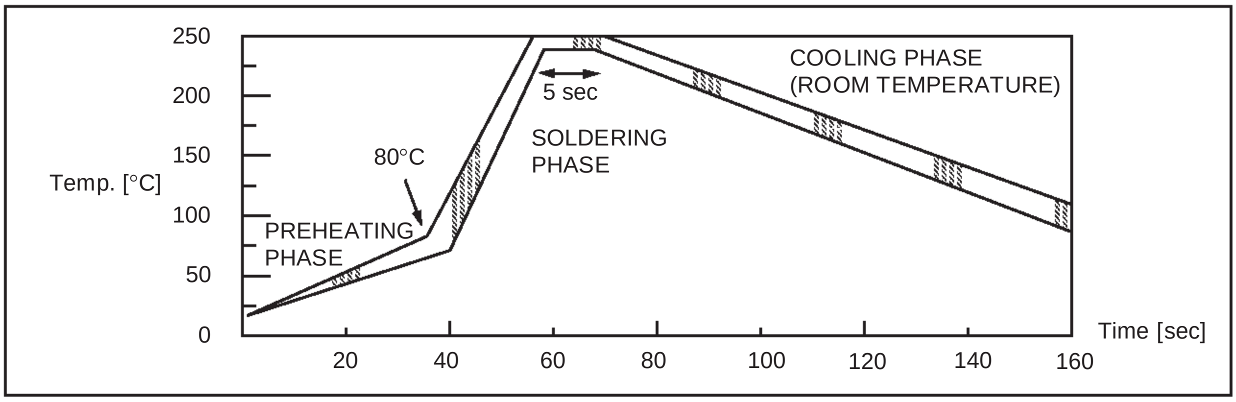
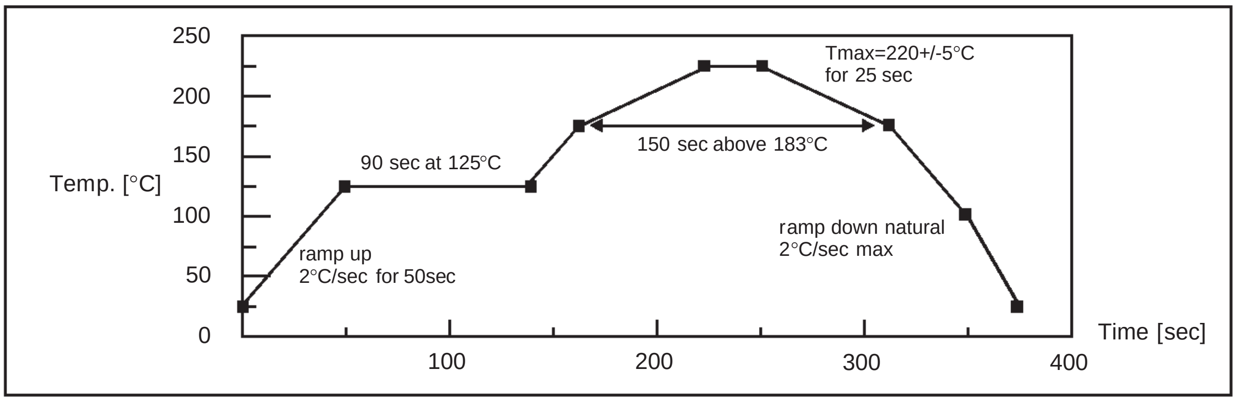


Figure 96. Recommended Reflow Soldering Oven Profile (MID JEDEC)



Recommended glue for SMD plastic packages dedicated to molding compound with silicone:

- Heraeus: PD945, PD955
- Loctite: 3615, 3298

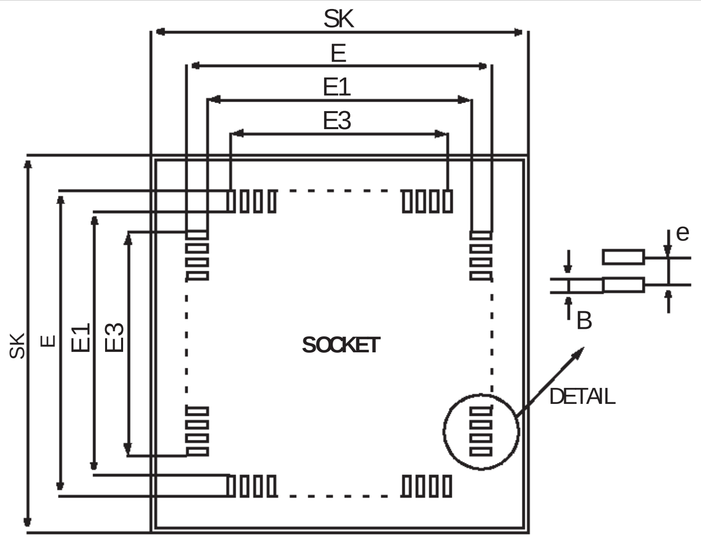
8.4 PACKAGE/SOCKET FOOTPRINT PROPOSAL

8.4.1 User-supplied TQFP64 Adaptor / Socket

To solder the TQFP64 device directly on the application board, or to solder a socket for connecting the emulator probe, the application board should provide the footprint described in Figure 97. This footprint allows both configurations:

- Direct TQFP64 soldering
- YAMAICHI IC149-064-008-S5 socket soldering to plug either the emulator probe or an adaptor board with an TQFP64 clamshell socket.
  This socket is not compatible with TQFP64 package.

Figure 97. TQFP64 Device And Emulator Probe Compatible Footprint



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
B	0.35	0.45	0.50	0.014	0.018	0.020
E	20.80			0.819		
E1		14.00			0.551	
E3	11.90	12.00	12.10	0.468	0.472	0.476
e	0.75	0.80	0.85	0.029	0.031	0.033
SK*		26			1.023	
	Number of Pins					
N	64 (4x16)					

* SK: Plastic socket overall dimensions.

Table 26. Suggested List of TQFP64 Socket Types

Package / Probe	Adaptor / Socket Reference		Socket type
TQFP64	ENPLAS	OTQ-64-0.8-02	Open Top
	YAMAICHI	IC51-0644-1240.KS-14584	Clamshell
EMU PROBE	YAMAICHI	IC149-064-008-S5	SMC

9 DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (OTP) as well as in factory coded versions (ROM). OTP devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that OTP devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

9.1 OPTION BYTES

The option byte allows the hardware configuration of the microcontroller to be selected.
The option byte has no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the OTP is fixed to FFh. This means that all the options have "1" as their default value.
In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

	USER OPTION BYTE 7							0
					FMP		WDG HALT	WDG SW
Default Value	1	1	1	1	1	1	1	1

USER OPTION BYTE

Bit 7:6,4 = **Reserved**, *must always be 1.*

Bit 5 = **Reserved**, *must always be 0.*

Bit 3 = **FMP** *Full memory protection*
This option bit allows the protection of the software contents against piracy (program or data). When the protection is activated, read-out of the EPROM or data EEPROM contents is prevented by hardware.
0: Read-out protection enabled
1: Read-out protection disabled

Bit 2 = **Reserved**, *must always be 1*

Bit 1 = **WDG HALT** *Watchdog and HALT mode*
This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

Bit 0 = **WDG SW** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

9.2 DEVICE ORDERING INFORMATION AND 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 S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

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

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

Figure 98. ROM Factory Coded Device Types

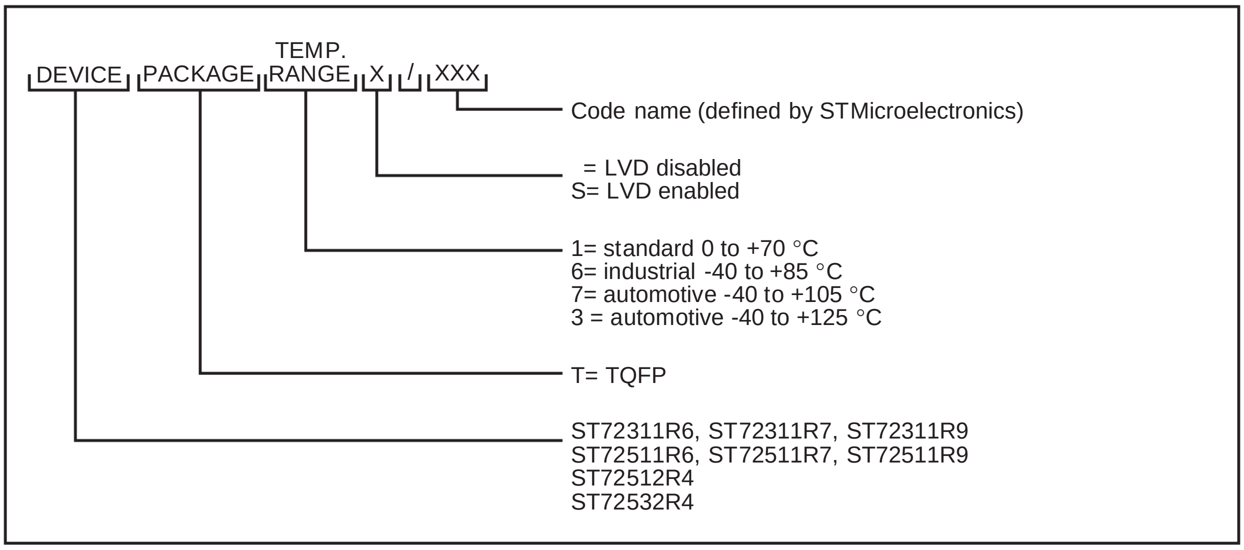
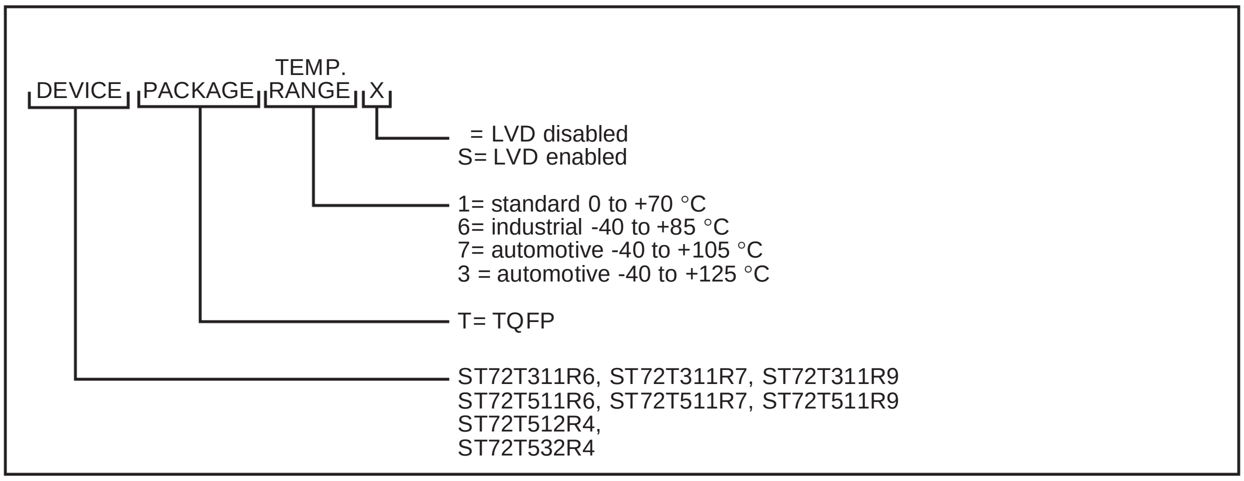

DEVICE
PACKAGE
TEMP. RANGE
X
/
XXX
Code name (defined by STMicroelectronics)
= LVD disabled
S= LVD enabled
1= standard 0 to +70 °C
6= industrial -40 to +85 °C
7= automotive -40 to +105 °C
3 = automotive -40 to +125 °C
T= TQFP
ST72311R6, ST72311R7, ST72311R9
ST72511R6, ST72511R7, ST72511R9
ST72512R4
ST72532R4

Figure 99. OTP User Programmable Device Types


DEVICE
PACKAGE
TEMP. RANGE
X
= LVD disabled
S= LVD enabled
1= standard 0 to +70 °C
6= industrial -40 to +85 °C
7= automotive -40 to +105 °C
3 = automotive -40 to +125 °C
T= TQFP
ST72T311R6, ST72T311R7, ST72T311R9
ST72T511R6, ST72T511R7, ST72T511R9
ST72T512R4,
ST72T532R4

TRANSFER OF CUSTOMER CODE (Cont'd)

MICROCONTROLLER OPTION LIST

Customer	. . . .
Address	. . . .
	. . . .
Contact	. . . .
Phone N°	. . . .
Reference	. . . .

STMicroelectronics references

Device:	[ ] ST72311R9	[ ] ST72511R9	[ ] ST72512R4
	[ ] ST72311R7	[ ] ST72511R7	[ ] ST72532R4
	[ ] ST72311R6	[ ] ST72511R6	
Package:	[ ] TQFP64		
Temperature Range:	[ ] 0°C to + 70°C	[ ] - 40°C to + 85°C	
	[ ] - 40°C to + 105°C	[ ] - 40°C to + 125°C	
Oscillator Source Selection:	[ ] Quartz Crystal/Ceramic resonator		
	[ ] External Clock		
Watchdog Selection:	[ ] Software Activation		
	[ ] Hardware Activation		
Watchdog Reset on Halt	[ ] Disabled		
	[ ] Enabled		
Readout Protection:	[ ] Disabled		
	[ ] Enabled		
LVD Reset	[ ] Disabled	[ ] Enabled:	

Comments :

Supply Operating Range in the application:

Notes	. . . .
Signature	. . . .
Date	. . . .

9.3 DEVELOPMENT TOOLS

STMicroelectronics offers a range of hardware and software development tools for the ST7 microcontroller family. Full details of tools available for the ST7 from third party manufacturers can be obtain from the STMicroelectronics Internet site ➟ http//st7.st.com.

Third Party Tools

- ACTUM
- BP
- COSMIC
- CMX
- DATA I/O
- HITEX
- HIWARE
- ISYSTEM
- KANDA
- LEAP

Tools from these manufacturers include C compilers, emulators and gang programmers.

STMicroelectronics Tools

Four types of development tool are offered by ST, all of which connect to a PC via a parallel (LPT) port:

	In-Circuit Emulation	Programming Capability[1)]	Software Included
ST7 Development Kit	Yes. (Same features as HDS2 emulator but no trace/ logic analyzer)	Yes (DIP packages only)	ST7 CD ROM with: – ST7 Assembly toolchain – WGDB7 powerful Source Level Debugger for Win 3.1, Win 95 and NT) – C compiler demo versions – ST Realizer for Win 3.1 and Win 95. – Windows Programming Tools for Win 3.1, Win 95 and NT
ST7 HDS2 Emulator	Yes, powerful emulation features including trace/ logic analyzer	No	
ST7 Programming Board	No	Yes (All packages)	

Note:
1. In Situ Programming (ISP) interface for FLASH devices.

Table 27. STMicroelectronics Development Tools

Supported Products	Development Kit	HDS2 Emulator	Programming Board
ST72311R6, ST72311R7, ST72311R9 ST72511R6, ST72511R7, ST72511R9 ST72512R4 ST72532R4	ST7MDT2-DVP2	ST7MDT2-EMU2B	ST7MDT2-EPB2

10 ST7 GENERIC APPLICATION NOTE

Identification	Description
PROGRAMMING AND TOOLS	
AN912	A simple guide to development tools
AN985	Executing code in ST7 RAM
AN986	Using the ST7 indirect addressing mode
AN987	ST7 in-circuit programming
AN988	Starting with ST7 assembly tool chain
AN989	Starting with ST7 Hiware C
AN1039	ST7 math utility routines
AN1064	Writing optimized hiware C language for ST7
EXAMPLE DRIVERS	
AN969	ST7 SCI communication between the ST7 and a PC
AN970	ST7 SPI communication between the ST7 and E PROM
AN971	ST7 I C communication between the ST7 and E PROM
AN972	ST7 software SPI master communication
AN973	SCI software communication with a PC using ST72251 16-bit timer
AN974	Real time clock with the ST7 timer output compare
AN976	Driving a buzzer using the ST7 PWM function
AN979	Driving an analog keyboard with the ST7 ADC
AN980	ST7 keypad decoding techniques, implementing wake-up on keystroke
AN1017	Using the ST7 USB microcontroller
AN1041	Using ST7 PWM signal to generate analog output (sinusoid)
AN1042	ST7 routine for I C slave mode management
AN1044	Multiple interrupt sources management for ST7 MCUs
AN1045	ST7 software implementation of I C bus master
AN1047	Managing reception errors with the ST7 SCI peripheral
AN1048	ST7 software LCD driver
AN1048	ST7 timer PWM duty cycle switch for true 0% or 100% duty cycle
PRODUCT OPTIMIZATION	
AN982	Using ceramic resonators with the ST7
AN1014	How to minimize the ST7 power consumption
AN1070	ST7 checksum selfchecking capability
PRODUCT EVALUATION	
AN910	ST7 and st9 performance benchmarking
AN990	ST7 benefits versus industry standard
APPLICATIONS EXAMPLES	
AN1086	ST7 / ST10U435 CAN-Do solutions for car multiplexing

TO GET MORE INFORMATION

To get the updated information on that product please refer to STMicroelectronics web server.

➟ http://st7.st.com/

11 SUMMARY OF CHANGES

Description of the changes between the current release of the specification and the previous one.

Revision	Main changes	Date
1.4	- Update of the pin description: more information: trigger oscillator... (Table 1 on page 8) - Section 1.3 on page 10: Interrupt vector map removed (see Table 7 on page 35) - Correction of interrupt vector Ei3 and Ei2 for PB7:4 and PB3:0 (Table 7 on page 35) - Modification of the title level for memory description (Section 1.4 and Section 1.5) - Correction of HALT and WAIT flowcharts (Figure 24 on page 38 and Figure 25 on page 39) - Correction of the SLOW mode timing (Figure 26 on page 39) - New information in the I/O block diagram (Figure 27 on page 41) - Correction of the I/O port configuration table for PB4 and PB7 (Table 11 on page 42) - New High-Sink information in I/O port configuration (Table 11 on page 42) - Correction of the MISCR2 $\overline{SS}$ function on PC7 and not PB7 (Section 5.2.2 on page 45) - Correction of the SCI register map description (Table 21 on page 100) - New electrical parametric data format (Section 7 on page 128) with JDEC naming - New tools and application note information (Section 9 on page 153 and Section 10 on page 157)	Aug-99
1.5	- 16-bit Timer: new notes concerning functional modes (Section 5.5 on page 59) - SPI: in SPICR register SPR2 bit reinstated (Section 5.6 on page 76)	Sept-99

Notes: