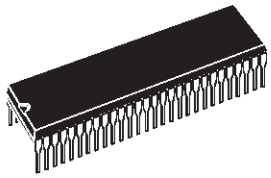
ST72334J/N, ST72314J/N, ST72124J

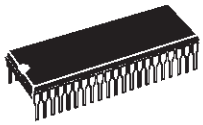
8-BIT MCU WITH SINGLE VOLTAGE FLASH MEMORY, ADC, 16-BIT TIMERS, SPI, SCI INTERFACES

PRODUCT PREVIEW

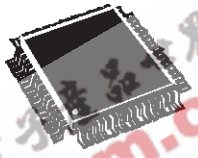
- 8K or 16K Program memory (ROM or Single voltage FLASH) with read-out protection
- 256-bytes EEPROM Data memory
- In-Situ Programming (Remote ISP)
- Enhanced Reset System
- Low voltage supply supervisor with 3 programmable levels
- Low consumption resonator or RC oscillators and by-pass for external clock source, with safe control capabilities
- 4 Power saving modes
- Standard Interrupt Controller
- 44 or 32 multifunctional bidirectional I/O lines:
  - External interrupt capability (4 vectors)
  - 21 or 19 alternate function lines
  - 12 or 8 high sink outputs
- Real time base, Beep and Clock-out capabilities
- Configurable watchdog reset
- Two 16-bit timers with:
  - 2 input captures (only one on timer A)
  - 2 output compares (only one on timer A)
  - External clock input on timer A
  - PWM and Pulse generator modes
- SPI synchronous serial interface
- SCI asynchronous serial interface
- 8-bit ADC with 8 input pins (6 only on ST72334Jx, not available on ST72124J2)



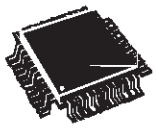
PSDIP56



PSDIP42



TQFP64
14 x 14



TQFP44
10 x 10

- 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	ST72124J2	ST72314J2	ST72314J4	ST72314N2	ST72314N4	ST72334J2	ST72334J4	ST72334N2	ST72334N4
Program memory - bytes	8K	8K	16K	8K	16K	8K	16K	8K	16K
RAM (stack) - bytes	384 (256)	384 (256)	512 (256)	384 (256)	512 (256)	384 (256)	512 (256)	384 (256)	512 (256)
EEPROM - bytes	-	-	-	-	-	256	256	256	256
Peripherals	Watchdog, 16-bit Timers, SPI, SCI	Watchdog, 16-bit Timers, SPI, SCI, ADC							
Operating Supply	3.0V to 5.5V								
CPU Frequency	500 kHz to 8 MHz (with 1 to 16 MHz oscillator)								
Operating Temperature	-40°C to +85°C (-40°C to +105/125°C optional)								
Packages	TQFP44 / SDIP42			TQFP64 / SDIP56		TQFP44 / SDIP42		TQFP64 / SDIP56	

Rev. 1.0

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1 PREAMBLE: ST72C334 VERSUS ST72E331 SPECIFICATION

New Features available on the ST72C334

- 8 or 16K FLASH/ROM with In-Situ Programming and Read-out protection
- New ADC with a better accuracy and conversion time
- New configurable Clock, Reset and Supply system
- New power saving mode with real time base: Active Halt
- Beep capability on PF1
- New interrupt source: Clock security system (CSS) or Main clock controller (MCC)

ST72C334 I/O Confuguration and Pinout

- Same pinout as ST72E331
- PA6 and PA7 are true open drain I/O ports without pull-up (same as ST72E331)
- PA3, PB3, PB4 and PF2 have no pull-up configuration (all IOs present on TQFP44)
- PA5:4, PC3:2, PE7:4 and PF7:6 have high sink capabilities (20mA on N-buffer, 2mA on P-buffer and pull-up). On the ST72E331, all these pads (except PA5:4) were 2mA push-pull pad without high sink capabilities. PA4 and PA5 were 20mA true open drain.

New Memory Locations in ST72C334

- 20h: MISCR register becomes MISCR1 register (naming change)
- 29h: new control/status register for the MCC module
- 2Bh: new control/status register for the Clock, Reset and Supply control. This register replaces the WDGSR register keeping the WDOGF flag compatibility.
- 40h: new MISCR2 register

2 GENERAL DESCRIPTION

2.1 INTRODUCTION

The ST72334J/N, ST72314J/N and ST72124J devices are members of the ST7 microcontroller family. They can be grouped as follows:

– ST72334J/N devices are designed for mid-range applications with Data EEPROM, ADC, SPI and SCI interface capabilities.

– ST72314J/N devices target the same range of applications but without Data EEPROM.

– ST72124J devices are for applications that do not need Data EEPROM and the ADC peripheral.

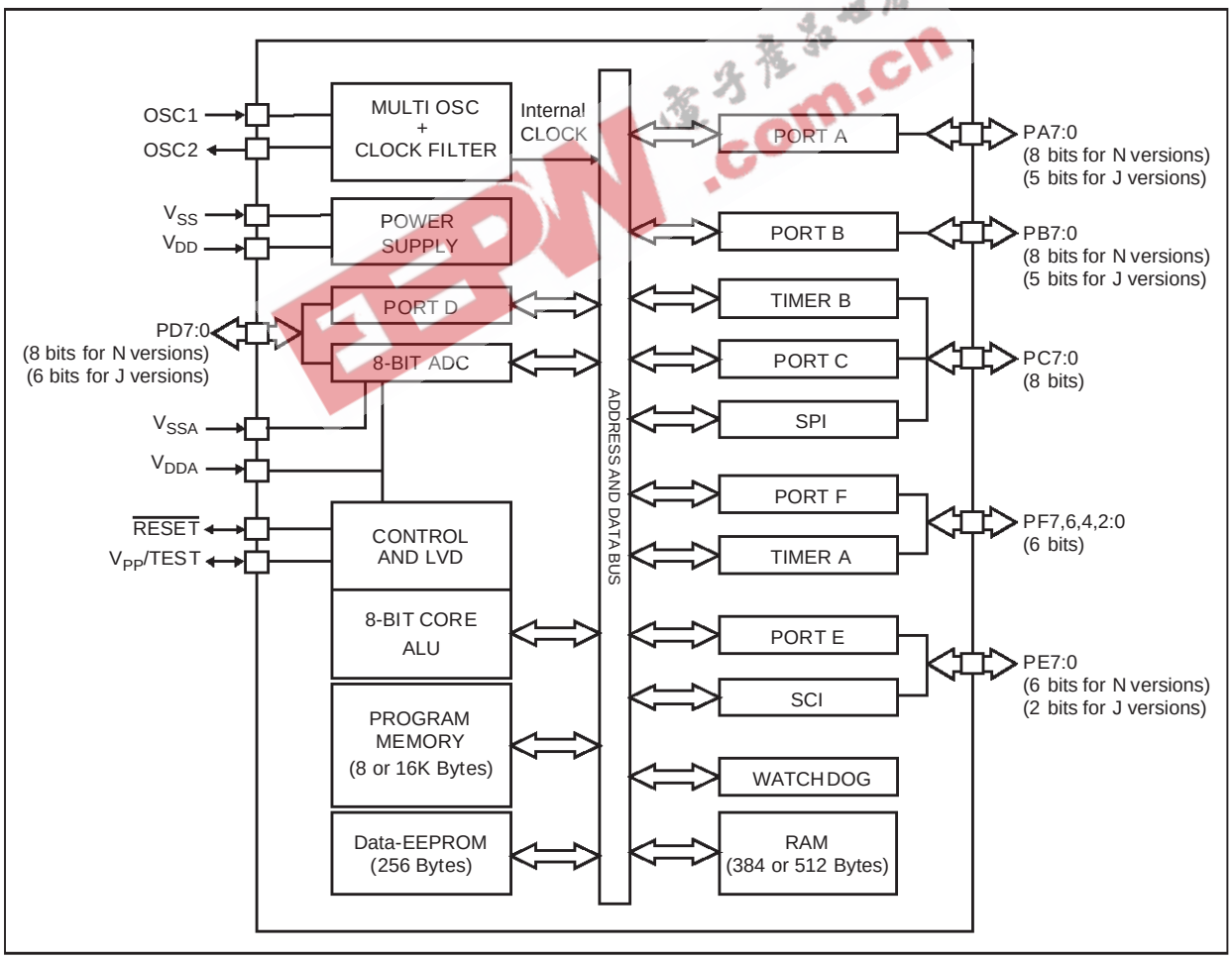
All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

The ST72C334J/N, ST72C314J/N and ST72C124J versions feature single-voltage FLASH memory with byte-by-byte In-Situ Programming (ISP) capability.

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.

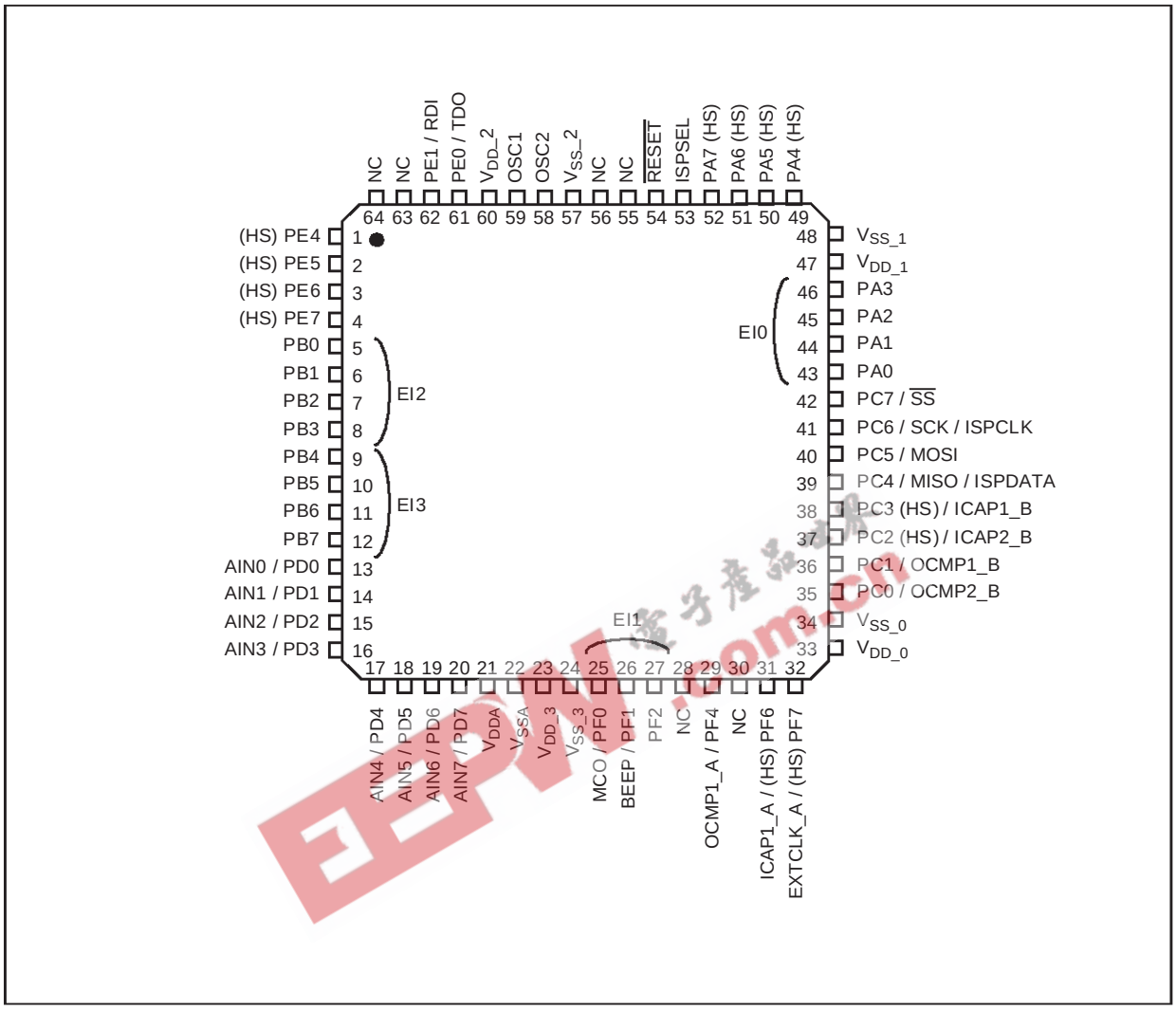
The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned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

Figure 1. Device Block Diagram



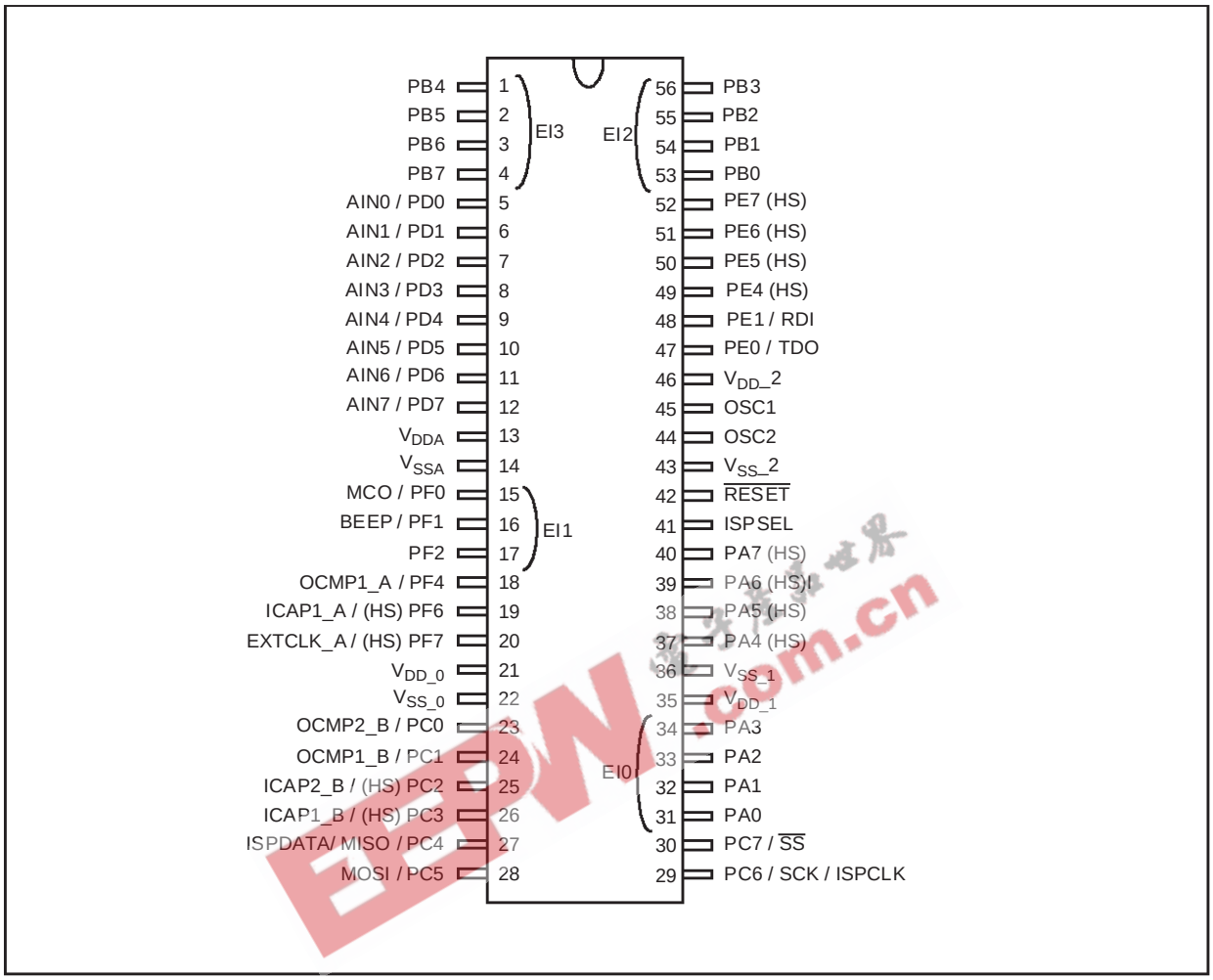
2.2 PIN DESCRIPTION

Figure 2. 64-Pin TQFP Package Pinout (N versions)



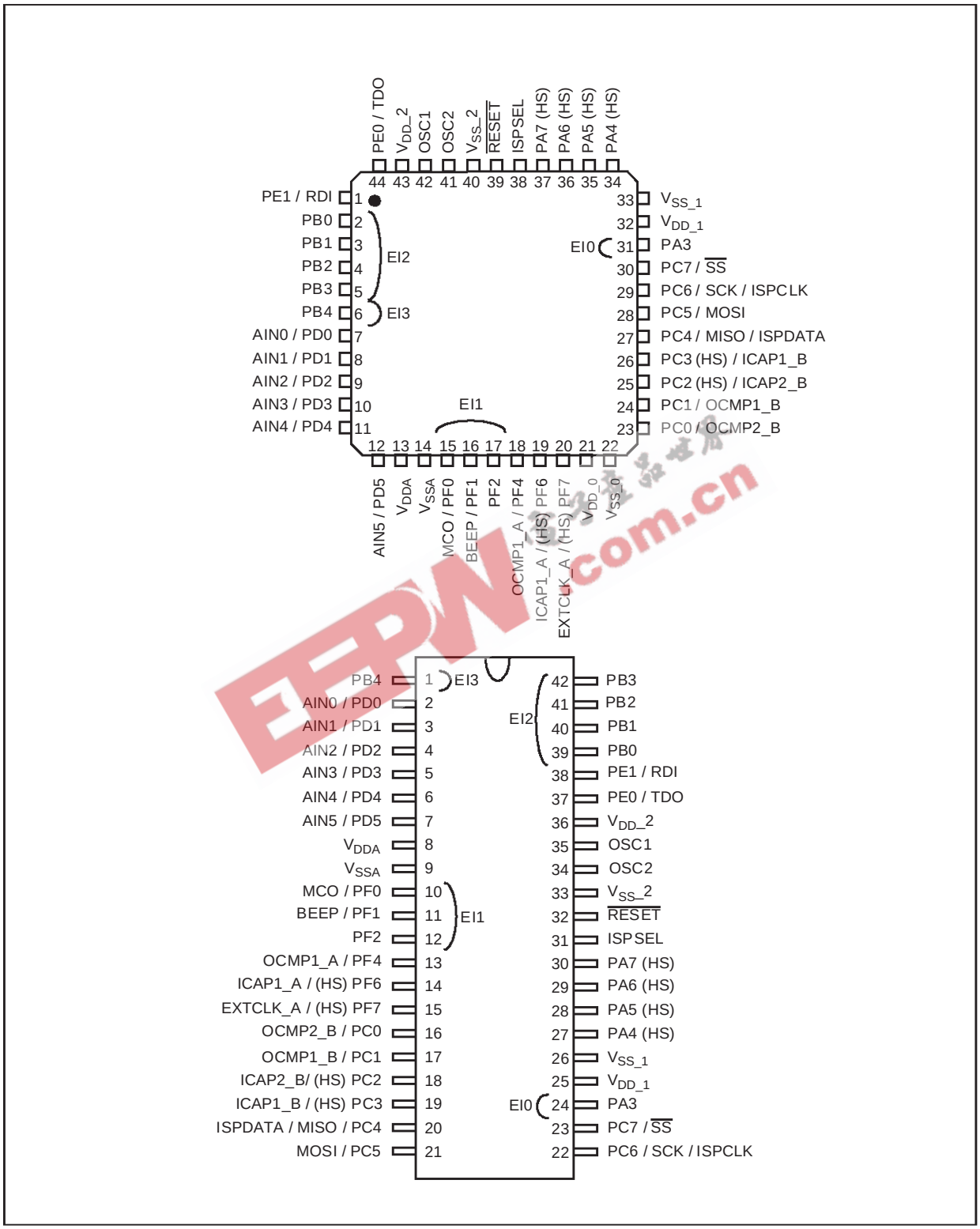
PIN DESCRIPTION (Cont'd)

Figure 3. 56-Pin SDIP Package Pinout (N versions)



PIN DESCRIPTION (Cont'd)

Figure 4. 44-Pin TQFP and 42-Pin SDIP Package Pinouts (J versions)



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	SDIP56	QFP44	SDIP42			Input	Output	float	wpu	int	ana	OD	PP		
1	49			PE4 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E4	
2	50			PE5 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E5	
3	51			PE6 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E6	
4	52			PE7 (HS)	I/O	C_T	HS	**X**	X			X	X	Port E7	
5	53	2	39	PB0	I/O	C_T		**X**	EI2			X	X	Port B0	
6	54	3	40	PB1	I/O	C_T		**X**	EI2			X	X	Port B1	
7	55	4	41	PB2	I/O	C_T		X	EI2			X	X	Port B2	
8	56	5	42	PB3	I/O	C_T		**X**		EI2		X	X	Port B3	
9	1	6	1	PB4	I/O	C_T		X		EI3		X	X	Port B4	
10	2			PB5	I/O	C_T		X	EI3			X	X	Port B5	
11	3			PB6	I/O	C_T		**X**	EI3			X	X	Port B6	
12	4			PB7	I/O	C_T		**X**	EI3			X	X	Port B7	
13	5	7	2	PD0/AIN0	I/O	C_T		**X**	X		X	X	X	Port D0	ADC Analog Input 0
14	6	8	3	PD1/AIN1	I/O	C_T		**X**	X		X	X	X	Port D1	ADC Analog Input 1
15	7	9	4	PD2/AIN2	I/O	C_T		**X**	X		X	X	X	Port D2	ADC Analog Input 2
16	8	10	5	PD3/AIN3	I/O	C_T		**X**	X		X	X	X	Port D3	ADC Analog Input 3
17	9	11	6	PD4/AIN4	I/O	C_T		**X**	X		X	X	X	Port D4	ADC Analog Input 4
18	10	12	7	PD5/AIN5	I/O	C_T		**X**	X		X	X	X	Port D5	ADC Analog Input 5
19	11			PD6/AIN6	I/O	C_T		**X**	X		X	X	X	Port D6	ADC Analog Input 6
20	12			PD7/AIN7	I/O	C_T		**X**	X		X	X	X	Port D7	ADC Analog Input 7
21	13	13	8	V_{DDA}	S									Analog Power Supply Voltage	
22	14	14	9	V_{SSA}	S									Analog Ground Voltage	
23				V_{DD_3}	S									Digital Main Supply Voltage	
24				V_{SS_3}	S									Digital Ground Voltage	
25	15	15	10	PF0/MCO	I/O	C_T		**X**	EI1			X	X	Port F0	Main clock output (f_{OSC}/2)
26	16	16	11	PF1/BEEP	I/O	C_T		**X**	EI1			X	X	Port F1	Beep signal output
27	17	17	12	PF2	I/O	C_T		**X**		EI1		X	X	Port F2	
28				NC	Not Connected										

Pin n°				Pin Name	Type	Level		Port						Main function (after reset)	Alternate function
								Input				Output			
TQFP64	SDIP56	QFP44	SDIP42			Input	Output	float	wpu	int	ana	OD	PP		
29	18	18	13	PF4/OCMP1_A	I/O	C_T		X	X			X	X	Port F4	Timer A Output Compare 1
30				NC	Not Connected										
31	19	19	14	PF6 (HS)/ICAP1_A	I/O	C_T	HS	X	X			X	X	Port F6	Timer A Input Capture 1
32	20	20	15	PF7 (HS)/EXTCLK_A	I/O	C_T	HS	X	X			X	X	Port F7	Timer A External Clock Source
33	21	21		V_{DD_0}	S									Digital Main Supply Voltage	
34	22	22		V_{SS_0}	S									Digital Ground Voltage	
35	23	23	16	PC0/OCMP2_B	I/O	C_T		X	X			X	X	Port C0	Timer B Output Compare 2
36	24	24	17	PC1/OCMP1_B	I/O	C_T		X	X			X	X	Port C1	Timer B Output Compare 1
37	25	25	18	PC2 (HS)/ICAP2_B	I/O	C_T	HS	X	X			X	X	Port C2	Timer B Input Capture 2
38	26	26	19	PC3 (HS)/ICAP1_B	I/O	C_T	HS	X	X			X	X	Port C3	Timer B Input Capture 1
39	27	27	20	PC4/MISO	I/O	C_T		X	X			X	X	Port C4	SPI Master In / Slave Out Data
40	28	28	21	PC5/MOSI	I/O	C_T		X	X			X	X	Port C5	SPI Master Out / Slave In Data
41	29	29	22	PC6/SCK	I/O	C_T		X	X			X	X	Port C6	SPI Serial Clock
42	30	30	23	PC7/$\overline{SS}$	I/O	C_T		X	X			X	X	Port C7	SPI Slave Select (active low)
43	31			PA0	I/O	C_T		X	EI0			X	X	Port A0	
44	32			PA1	I/O	C_T		X	EI0			X	X	Port A1	
45	33			PA2	I/O	C_T		X	EI0			X	X	Port A2	
46	34	31	24	PA3	I/O	C_T		X		EI0		X	X	Port A3	
47	35	32	25	V_{DD_1}	S									Digital Main Supply Voltage	
48	36	33	26	V_{SS_1}	S									Digital Ground Voltage	
49	37	34	27	PA4 (HS)	I/O	C_T	HS	X	X			X	X	Port A4	
50	38	35	28	PA5 (HS)	I/O	C_T	HS	X	X			X	X	Port A5	
51	39	36	29	PA6 (HS)	I/O	C_T	HS	X				T		Port A6	
52	40	37	30	PA7 (HS)	I/O	C_T	HS	X				T		Port A7	
53	41	38	31	ISPSEL	I									Must be tied low in user mode. In programming mode when available, this pin acts as In-Situ Programming mode selection.	
54	42	39	32	$\overline{RESET}$	I/O	C			X				X	Top priority non maskable interrupt (active low)	
55				NC	Not Connected										
56				NC											
57	43	40	33	V_{SS_3}	S									Digital Ground Voltage	
58	44	41	34	OSC2										These pins connect a parallel-resonant crystal or an external clock source to the on-chip main oscillator.	
59	45	42	35	OSC1											
60	46	43	36	V_{DD_3}	S									Digital Main Supply Voltage	
61	47	44	37	PE0/TDO	I/O	C_T		X	X			X	X	Port E0	SCI Transmit Data Out
62	48	1	38	PE1/RDI	I/O	C_T		X	X			X	X	Port E1	SCI Receive Data In
63				NC	Not Connected										
64				NC											

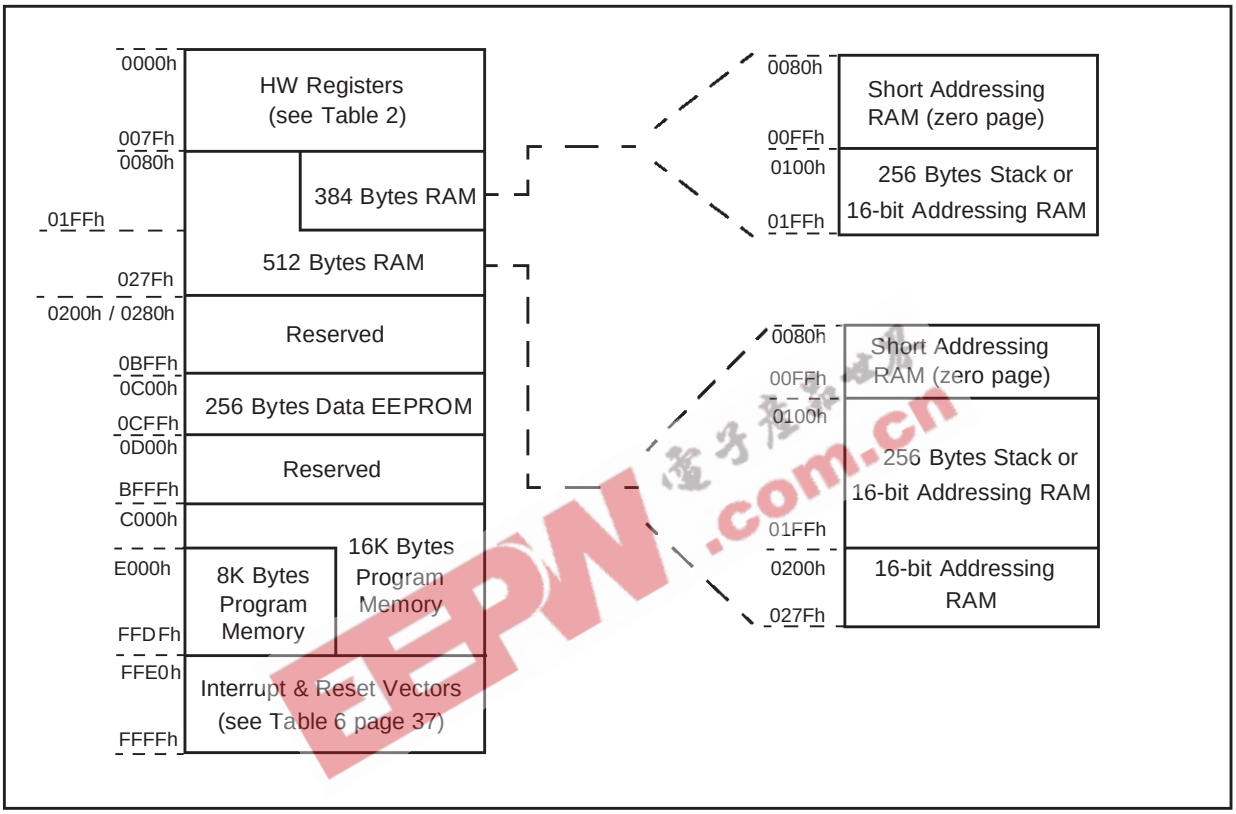
2.3 REGISTER & MEMORY MAP

As shown in the Figure 5, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, 384 or 512 bytes of RAM, up to 256 bytes of data EEPROM and 4 or 8 Kbytes 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 5. Memory Map



REGISTER & MEMORY MAP (Cont'd)

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 00h 00h	R/W R/W R/W 1)
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 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 00h 00h	R/W R/W R/W 1)
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 00h 00h	R/W R/W R/W 1)
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 00h 00h	R/W R/W R/W 1)
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 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 to 0028h	Reserved Area (5 Bytes)				
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		CRSR	Clock, Reset, Supply Control / Status Register	00h	R/W
002Ch	Data-EEPROM	EECSR	Data-EEPROM Control/Status Register	00h	R/W
002Dh 0030h	Reserved Area (4 Bytes)				
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh 003Ch 003Dh 003Eh 003Fh	TIMER A	TACR2 TACR1 TASR TAIC1HR TAIC1LR TAOC1HR TAOC1LR TACHR TACLR TAACHR TAACLR TAIC2HR TAIC2LR TAOC2HR TAOC2LR	Timer A Control Register 2 Timer A Control Register 1 Timer A Status Register Timer A Input Capture 1 High Register Timer A Input Capture 1 Low Register Timer A Output Compare 1 High Register Timer A Output Compare 1 Low Register Timer A Counter High Register Timer A Counter Low Register Timer A Alternate Counter High Register Timer A Alternate Counter Low Register Timer A Input Capture 2 High Register Timer A Input Capture 2 Low Register Timer A Output Compare 2 High Register Timer A Output Compare 2 Low Register	00h 00h xxh xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W Read Only Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only [2] Read Only [2] R/W [2] R/W [2]
0040h		MISCR2	Miscellaneous Register 2	00h	R/W
0041h 0042h 0043h 0044h 0045h 0046h 0047h 0048h 0049h 004Ah 004Bh 004Ch 004Dh 004Eh 004Fh	TIMER B	TBCR2 TBCR1 TBSR TBIC1HR TBIC1LR TBOC1HR TBOC1LR TBCHR TBCLR TBACHR TBACLR TBIC2HR TBIC2LR TBOC2HR TBOC2LR	Timer B Control Register 2 Timer B Control Register 1 Timer B Status Register Timer B Input Capture 1 High Register Timer B Input Capture 1 Low Register Timer B Output Compare 1 High Register Timer B Output Compare 1 Low Register Timer B Counter High Register Timer B Counter Low Register Timer B Alternate Counter High Register Timer B Alternate Counter Low Register Timer B Input Capture 2 High Register Timer B Input Capture 2 Low Register Timer B Output Compare 2 High Register Timer B Output Compare 2 Low Register	00h 00h xxh xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W Read Only Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only Read Only R/W R/W
0050h 0051h 0052h 0053h 0054h 0055h 0056h 0057h	SCI	SCISR SCIDR SCIBRR SCICR1 SCICR2 SCIERPR SCIETPR	SCI Status Register SCI Data Register SCI Baud Rate Register SCI Control Register 1 SCI Control Register 2 SCI Extended Receive Prescaler Register Reserved area SCI Extended Transmit Prescaler Register	C0h xxh 00xx xxxx xxh 00h 00h --- 00h	Read Only R/W R/W R/W R/W R/W R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
0058h 006Fh	Reserved Area (24 Bytes)				
0070h 0071h	ADC	ADCDR ADCCSR	Data Register Control/Status Register	xxh 00h	Read Only R/W
0072h to 007Fh	Reserved Area (14 Bytes)				

Notes:
1) The bits corresponding to unavailable pins are forced to 1 by hardware, this affects the reset status value.
2) External pin not available.
3) Not used in versions without Low Voltage Detector Reset.

2.4 FLASH PROGRAM MEMORY

2.4.1 Introduction

Flash devices have a single voltage non-volatile FLASH memory that may be programmed in-situ (or plugged in a programming tool) on a byte-by-byte basis.

2.4.2 Main features

- Remote In-Situ Programming (ISP) mode
- Up to 16 bytes programmed in the same cycle
- MTP memory (Multiple Time Programmable)
- Read-out memory protection against piracy

2.4.3 Structural organisation

The FLASH program memory is organised in a single 8-bit wide memory block which can be used for storing both code and data constants.

The FLASH program memory is mapped in the upper part of the ST7 addressing space (F000h-FFFFh) and includes the reset and interrupt user vector area .

2.4.4 In-Situ Programming (ISP) mode

The FLASH program memory can be programmed using Remote ISP mode. This ISP mode allows the contents of the ST7 program memory to be updated using a standard ST7 programming tools after the device is mounted on the application board. This feature can be implemented with a minimum number of added components and board area impact.

An example Remote ISP hardware interface to the standard ST7 programming tool is described below. For more details on ISP programming, refer to the ST7 Programming Specification.

Remote ISP Overview

The Remote ISP mode is initiated by a specific sequence on the dedicated ISPSEL pin.

The Remote ISP is performed in three steps:

- Selection of the RAM execution mode
- Download of Remote ISP code in RAM
- Execution of Remote ISP code in RAM to program the user program into the FLASH

Remote ISP hardware configuration

In Remote ISP mode, the ST7 has to be supplied with power (V_{DD} and V_{SS}) and a clock signal (oscillator and application crystal circuit for example).

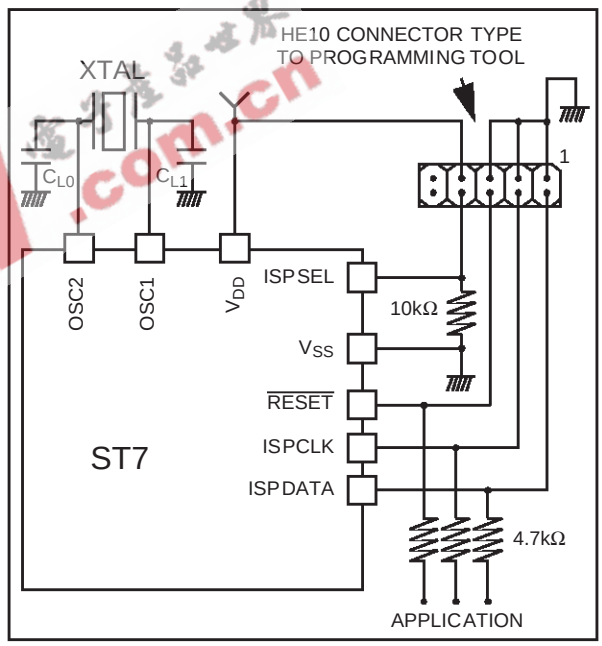
This mode needs five signals (plus the V_{DD} signal if necessary) to be connected to the programming tool. This signals are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device ground power supply
- ISPCLK: ISP output serial clock pin
- ISPDATA: ISP input serial data pin
- ISPSEL: Remote ISP mode selection. This pin must be connected to V_{SS} on the application board

If any of these pins are used for other purposes on the application, a serial resistor has to be implemented to avoid a conflict if the other device forces the signal level.

Figure 6 shows a typical hardware interface to a standard ST7 programming tool. For more details on the pin locations, refer to the device pinout description.

Figure 6. Typical Remote ISP Interface



2.5 Program Memory Read-out Protection

The read-out protection is enabled through an option bit.

For FLASH devices, when this option is selected, the program and data stored in the FLASH memory are protected against read-out piracy (including a re-write protection). When this protection option is removed the entire FLASH program memory is first automatically erased.

2.6 DATA EEPROM

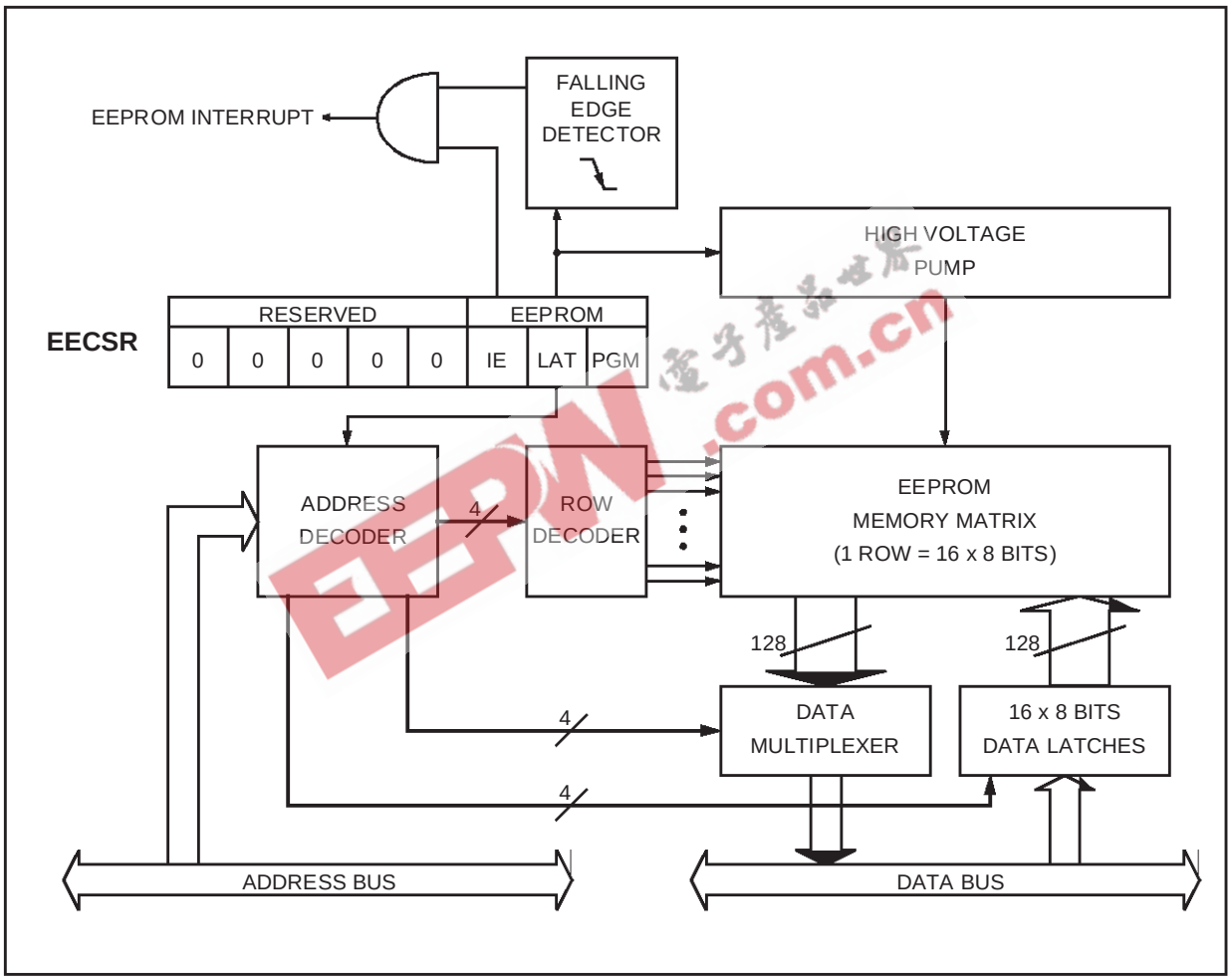
2.6.1 Introduction

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.

2.6.2 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 7. EEPROM Block Diagram



DATA EEPROM (Cont'd)

2.6.3 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 8 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)

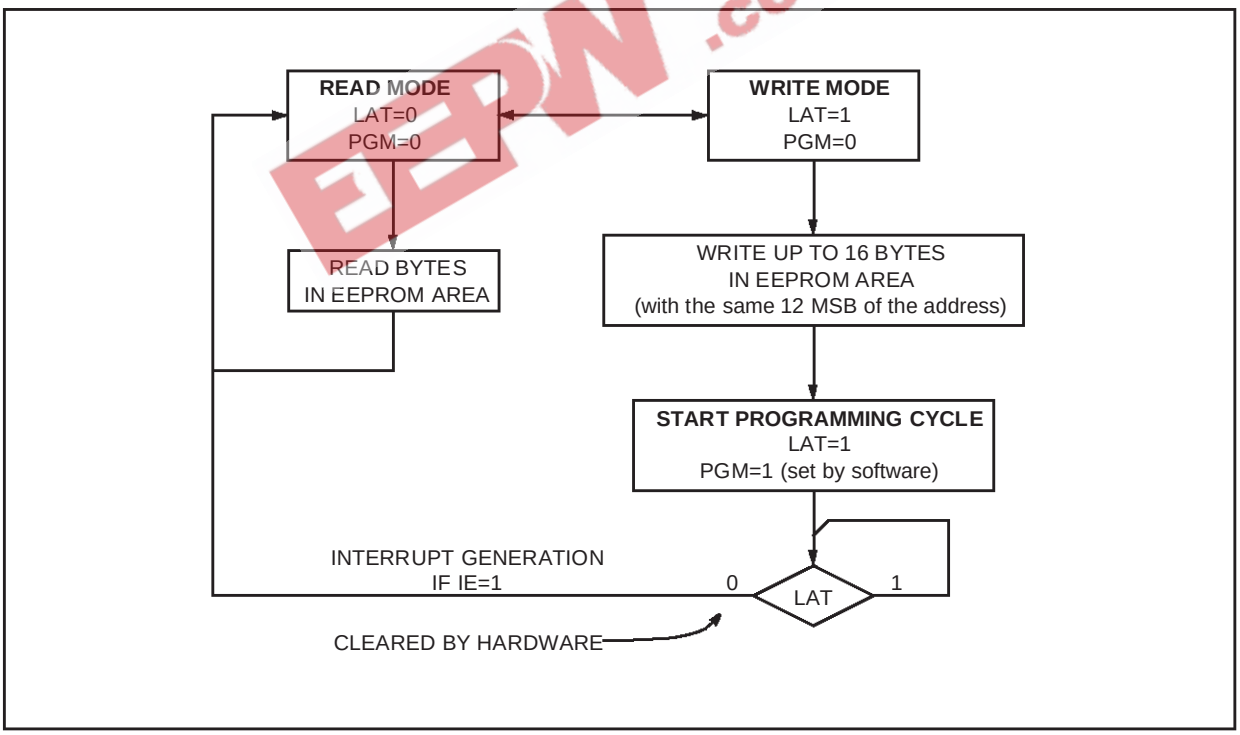
To access the write mode, the LAT bit has to be set by software (the PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 16 data latches according to its address.

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 illustrated by the Figure 9.

Figure 8. Data EEPROM Programming Flowchart



DATA EEPROM (Cont'd)

2.6.4 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.

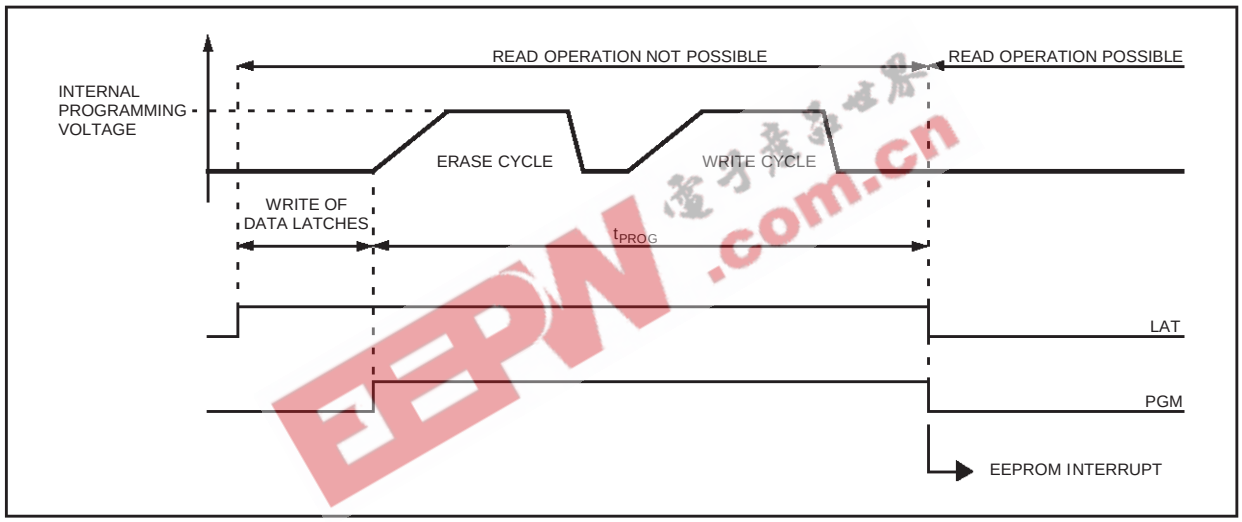
2.6.5 Data EEPROM Access Error Handling

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 9. Data EEPROM Programming Cycle



DATA EEPROM (Cont'd)

2.6.6 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**						IE	RWM	PGM
	Reset Value	0	0	0	0	0	0	0	0

3 CENTRAL PROCESSING UNIT

3.1 INTRODUCTION

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

3.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

3.3 CPU REGISTERS

The 6 CPU registers shown in Figure 10 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

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

Index Registers (X and Y)

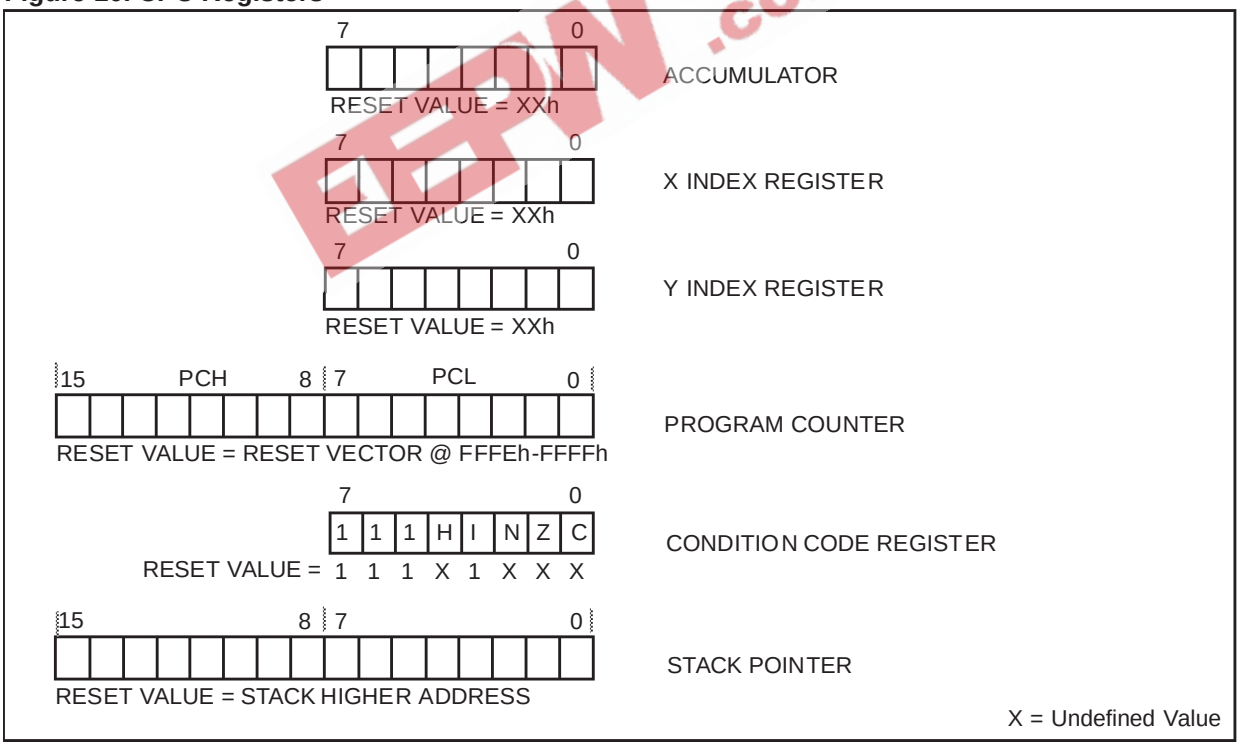
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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

Program Counter (PC)

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

Figure 10. CPU Registers



CENTRAL PROCESSING UNIT (Cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

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

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

Bit 4 = **H** *Half carry.*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.
0: No half carry has occurred.
1: A half carry has occurred.

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

Bit 3 = **I** *Interrupt mask.*

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
0: Interrupts are enabled.
1: Interrupts are disabled.

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

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptable because the I bit is set by hardware when you enter it and reset by the IRET instruction at the end of the interrupt routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = **N** *Negative.*

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero.*

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

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

Bit 0 = **C** *Carry/borrow.*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.

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

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh

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 11).

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.

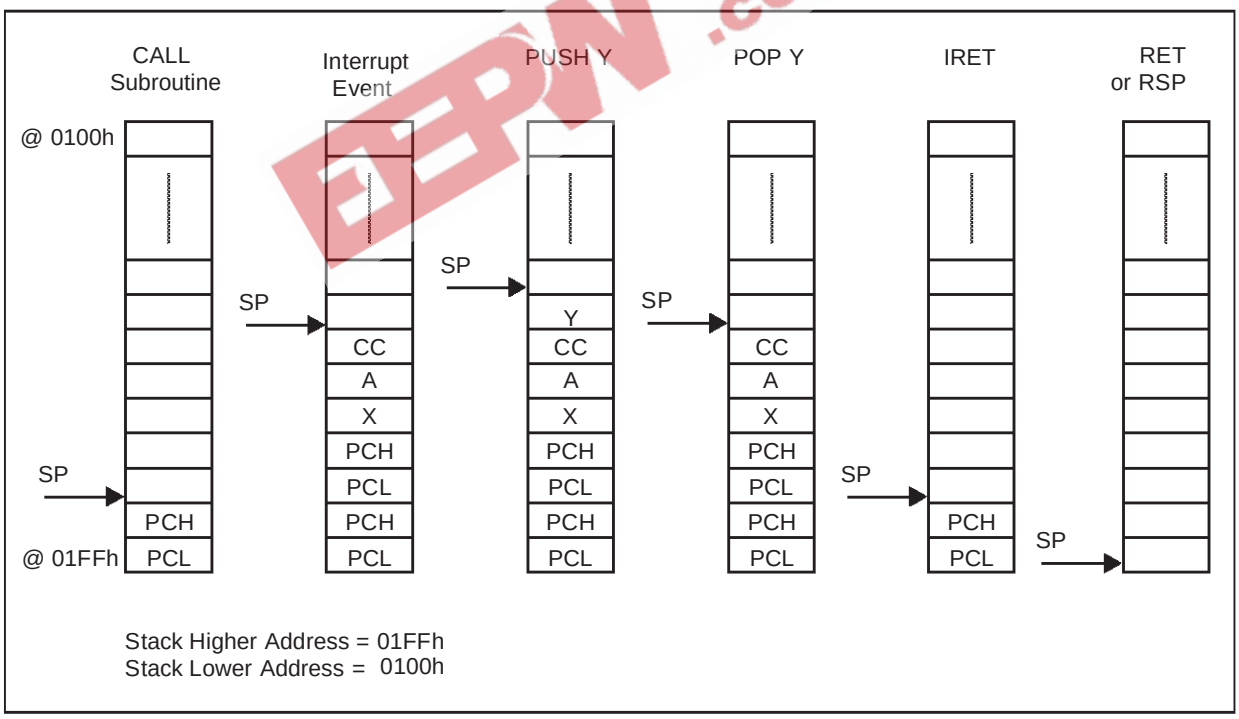
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 11.

– 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 11. Stack Manipulation Example



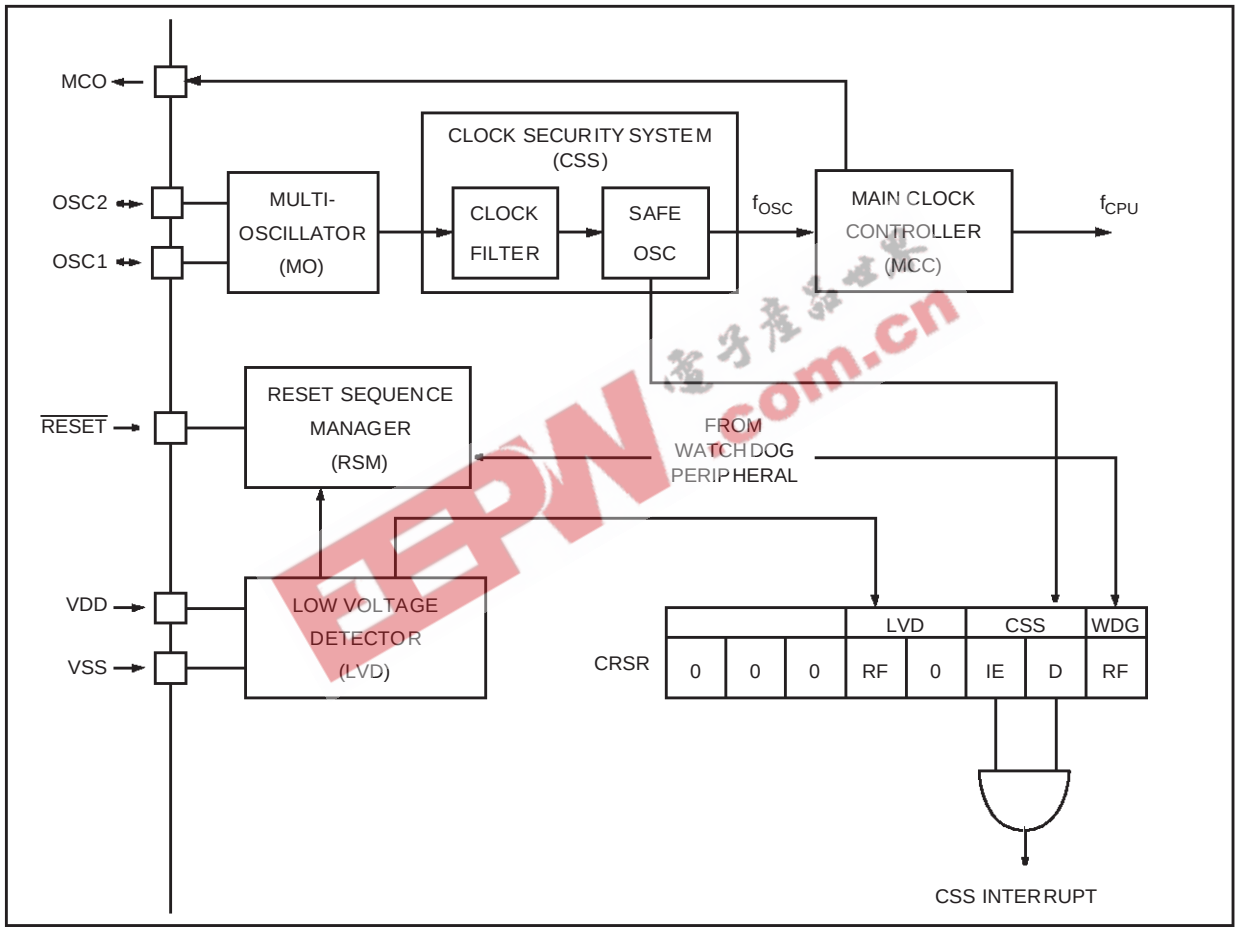
4 SUPPLY, RESET AND CLOCK MANAGEMENT

The ST72334J/N, ST72314J/N and ST72124J 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 12.

Main Features

- Supply Manager with Main supply Low voltage detection (LVD)
- Reset Sequence Manager (RSM)
- Multi-Oscillator (MO)
  - 4 Crystal/Ceramic resonator oscillators
  - 1 External RC oscillator
  - 1 Internal RC oscillator
- Clock Security System (CSS)
  - Clock Filter
  - Backup Safe Oscillator

Figure 12. Clock, Reset and Supply Block Diagram



4.1 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_{LVDf} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{LVDf} reference value for a voltage drop is lower than the V_{LVDr} 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_{LVDr} when V_{DD} is rising
– V_{LVDf} when V_{DD} is falling

The LVD function is illustrated in the Figure 13.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is below V_{LVDf}, 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:
1) the LVD allows the device to be used without any external RESET circuitry.

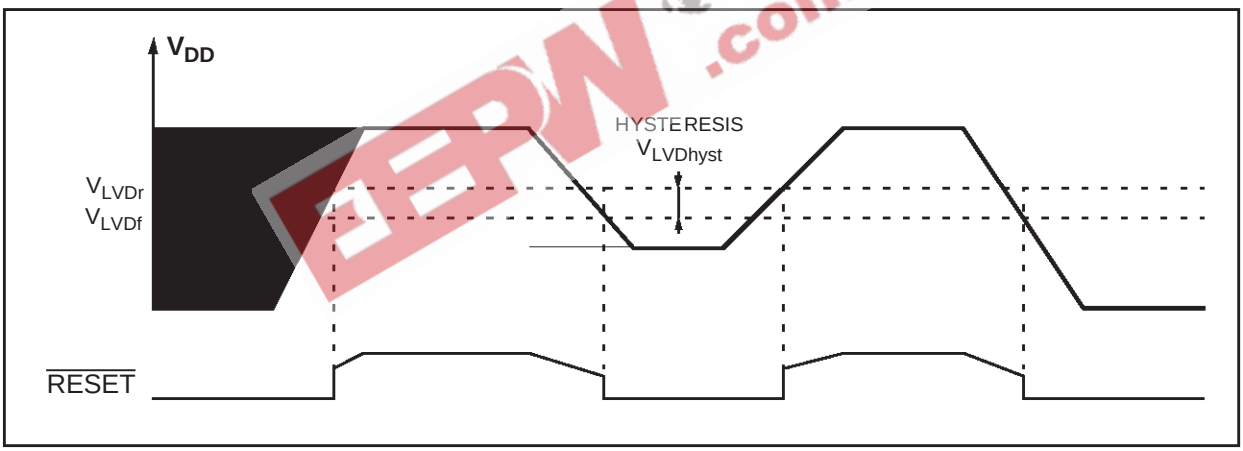
2) three different reference levels are selectable through the OPTION BYTE according to the application requirement.

LVD application note

Application software can detect a reset caused by the LVD by reading the LVDRF bit in the CRSR register.

This bit is set by hardware when a LVD reset is generated and cleared by software (writing zero).

Figure 13. Low Voltage Detector vs Reset



4.2 RESET SEQUENCE MANAGER (RSM)

The reset sequence manager includes three RESET sources as shown in Figure 15:

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

These sources act on the $\overline{\text{RESET}}$ PIN and it is always kept low during the delay phase.

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

The basic RESET sequence consists of 3 phases as shown in Figure 14:

- Delay depending on the RESET source
- 4096 CPU clock cycle delay
- RESET vector fetch

The 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 14. RESET Sequence Phases

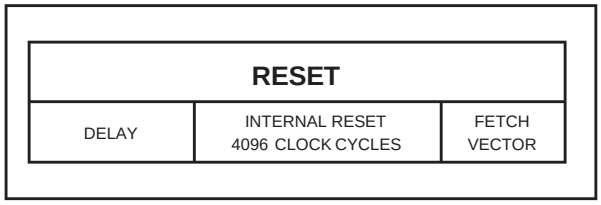
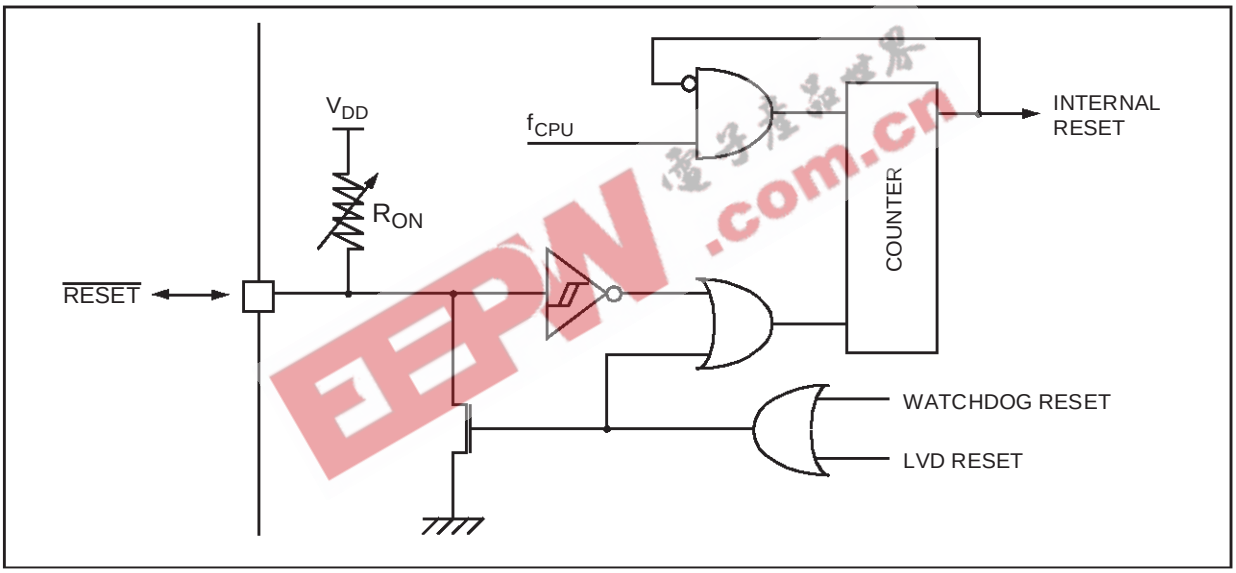


Figure 15. Reset Block Diagram



RESET SEQUENCE MANAGER (Cont'd)

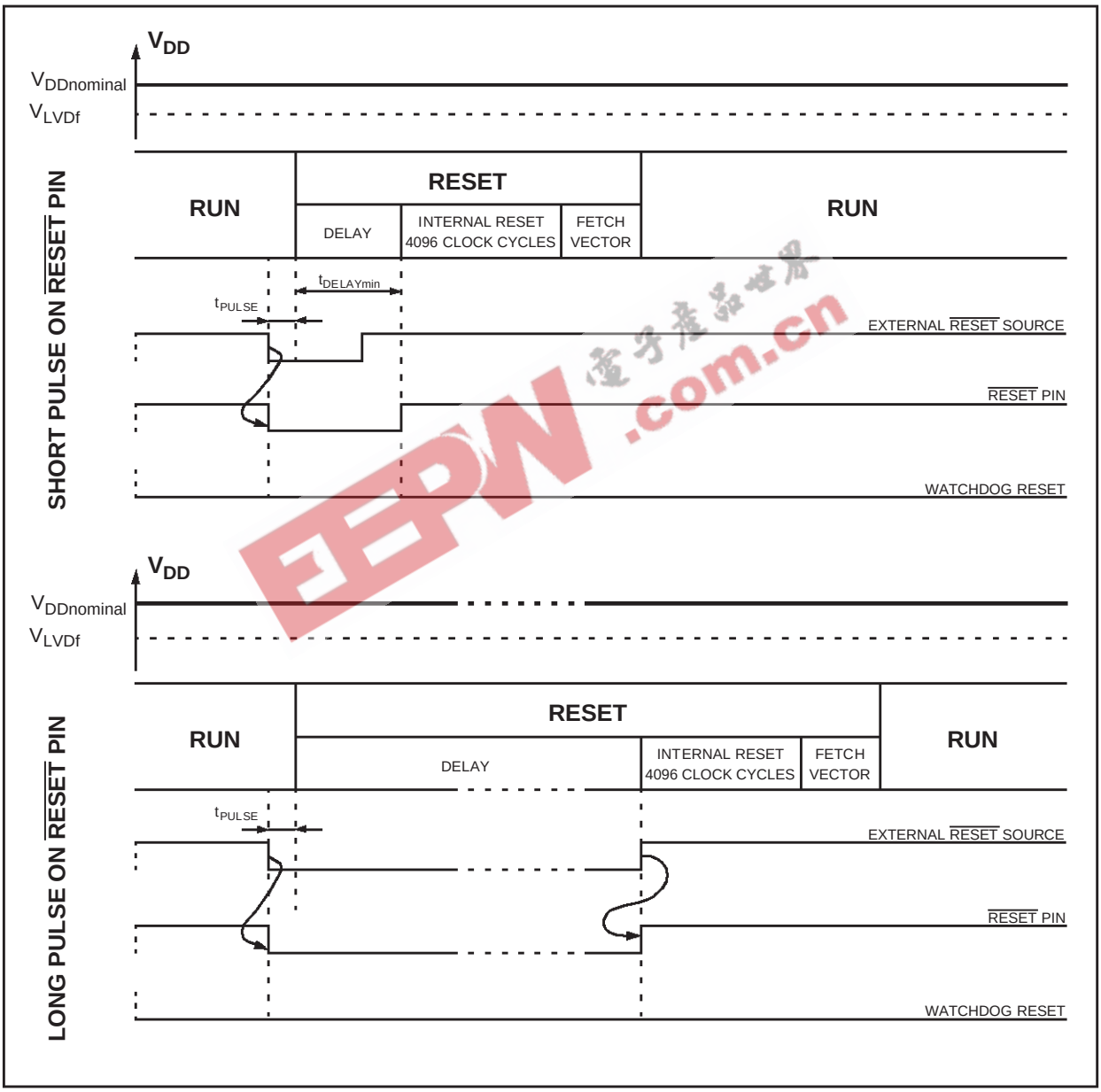
External $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. 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_{PULSE} in order to be recognized. Two RESET sequences can be associated with this RESET source as shown in Figure 16.

Starting from the external RESET pulse recognition, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{DELAYmin}$.

Figure 16. External RESET Sequences



RESET SEQUENCE MANAGER (Cont'd)

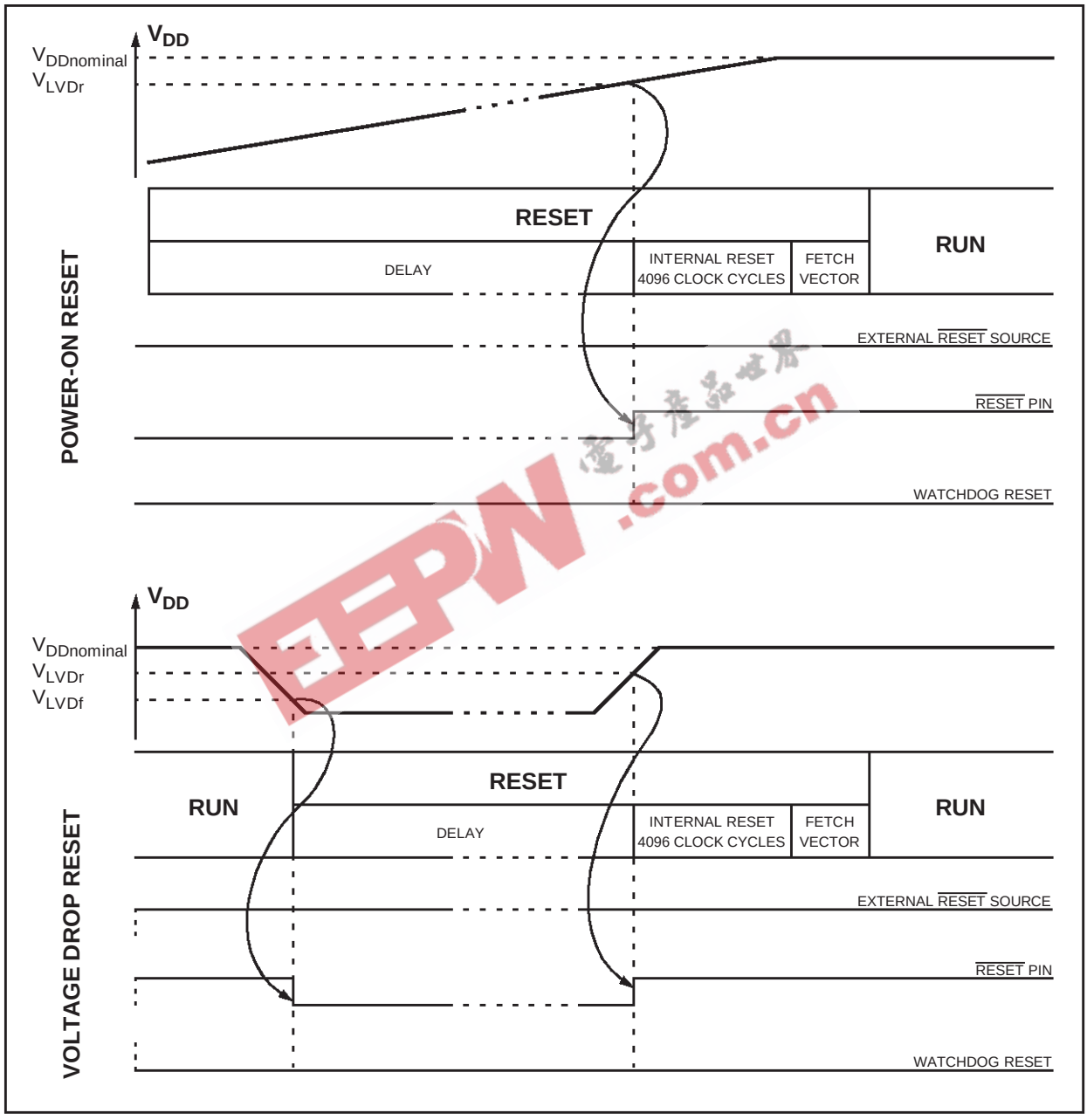
Internal Low Voltage Detection RESET

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:

- Power-On RESET
- Voltage Drop RESET

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}<V_{LVDr}$ (rising edge) or $V_{DD}<V_{LVDf}$ (falling edge) as shown in Figure 9.

Figure 17. LVD RESET Sequences



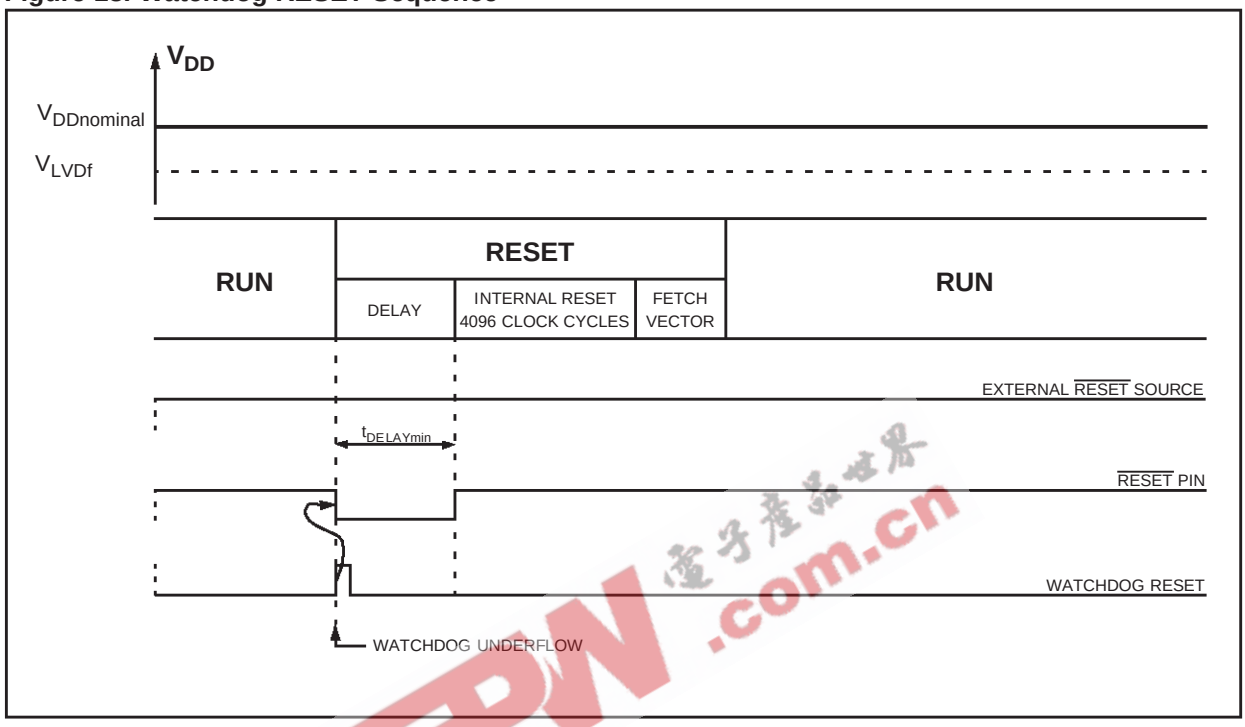
RESET SEQUENCE MANAGER (Cont'd)

Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 18.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{DELAYmin}$.

Figure 18. Watchdog RESET Sequence



MULTI-OSCILLATOR (MO)

The main clock of the ST7 can be generated by 7 different sources coming from the multi-oscillator block:

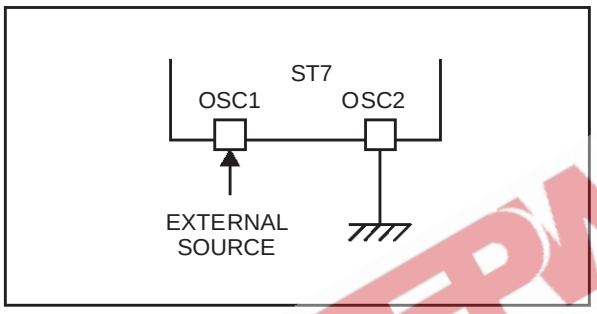
- an external source
- 4 crystal or ceramic resonator oscillators
- 1 external RC oscillator
- 1 internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the OPTION BYTE.

External Clock Source

The default OPTION BYTE value selects the External Clock in the MO block. In this mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to ground (see Figure 19).

Figure 19. MO External Clock

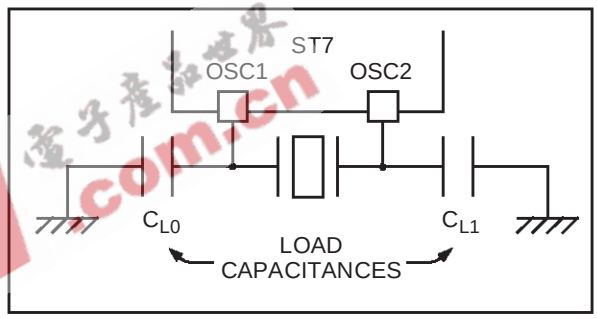


Crystal/Ceramic Oscillators

This family of oscillators has the advantage of producing a high accuracy on the main clock of the ST7. The selection within a list of 4 oscillators with different frequency ranges has to be done by OPTION BYTE in order to reduce the consumption. In this mode of the MO block, the resonator and the load capacitances have to be connected as shown in Figure 20 and have to be mounted as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators, when selected via the OPTION BYTE, are not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

Figure 20. MO Crystal/Ceramic Resonator



MULTI-OSCILLATOR (Cont'd)

External RC Oscillator

This oscillator allows a low cost solution for the main clock of the ST7 using only an external resistor and an external capacitor (see Figure 21). The selection of the external RC oscillator has to be done by OPTION BYTE.

The frequency of the external RC oscillator (in the range of some MHz.) is fixed by the resistor and the capacitor values:

$$f_{OSC} \sim \frac{4}{R_{EX} \cdot C_{EX}} \quad 1)$$

The previous formula shows that in this MO mode, the accuracy of the clock is directly linked to the accuracy of the discrete components.

Internal RC Oscillator

The Internal RC oscillator mode is based on the same principle as the External RC oscillator including the resistance and the capacitance of the device. This mode is the most cost effective one with the drawback of a lower frequency accuracy. Its frequency is in the range of several MHz.

In this mode, the two oscillator pins have to be tied to ground as shown in Figure 22.

The selection of the internal RC oscillator has to be done by OPTION BYTE.

Figure 21. MO External RC

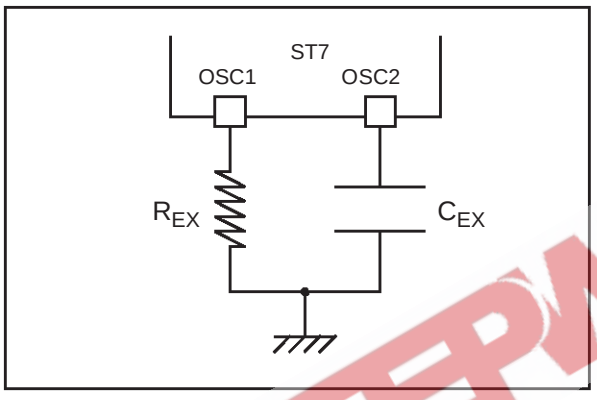
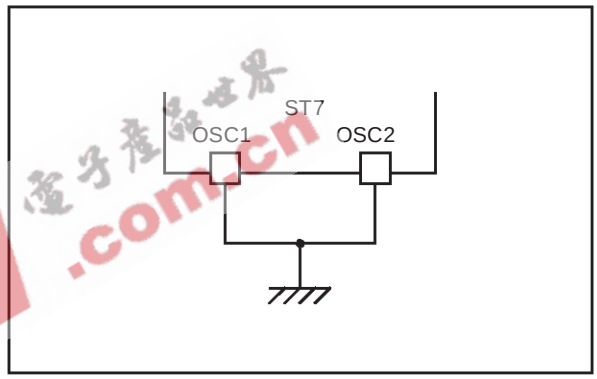


Figure 22. MO Internal RC



Note:

1) This formula provides an approximation of the frequency with typical R_{EX} and C_{EX} values at V_{DD}=5V. It is given only as design guidelines.

4.3 CLOCK SECURITY SYSTEM (CSS)

The Clock Security System (CSS) protects the ST7 against main clock problems. To allow the integration of the security features in the applications, it is based on a clock filter control and an Internal Safe Oscillator. The CSS can be disabled by OPTION BYTE.

4.3.1 Clock Filter Control

The Clock Filter is based on a clock frequency limitation function.

This filter function is able to detect and filter high frequency spikes on the ST7 main clock.

If the oscillator is not working properly (e.g. working at a harmonic frequency of the resonator), the current active oscillator clock can be totally filtered, and then no clock signal is available for the ST7 from this oscillator anymore. If the original clock source recovers, the filtering is stopped automatically and the oscillator supplies the ST7 clock.

4.3.2 Safe Oscillator Control

The Safe Oscillator of the CSS block is a low frequency back-up clock source (see Figure 24).

If the clock signal disappears (due to a broken or disconnected resonator...) during a Safe Oscillator period, the Safe oscillator delivers a low frequency clock signal which allows the ST7 to perform some rescue operations.

Automatically, the ST7 clock source switches back from the Safe Oscillator if the original clock source recovers.

Limitation detection

The automatic Safe Oscillator selection is notified by hardware setting the CSSD bit of the CRSR register. An interrupt can be generated if the CSSIE bit has been previously set.
These two bits are described in the CRSR register description.

Figure 23. Clock Filter Function

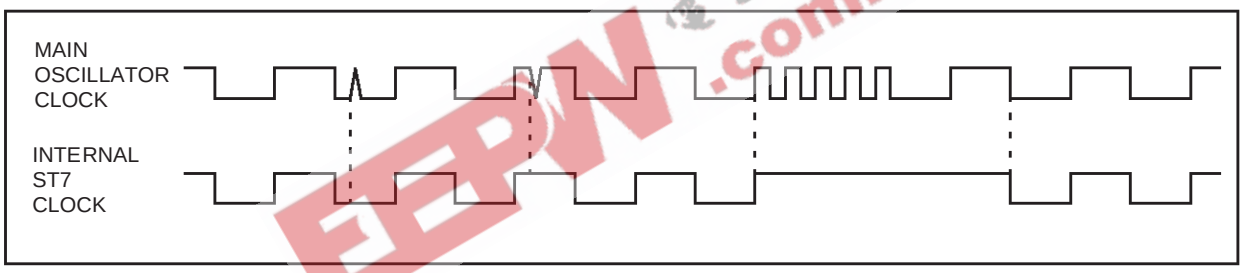
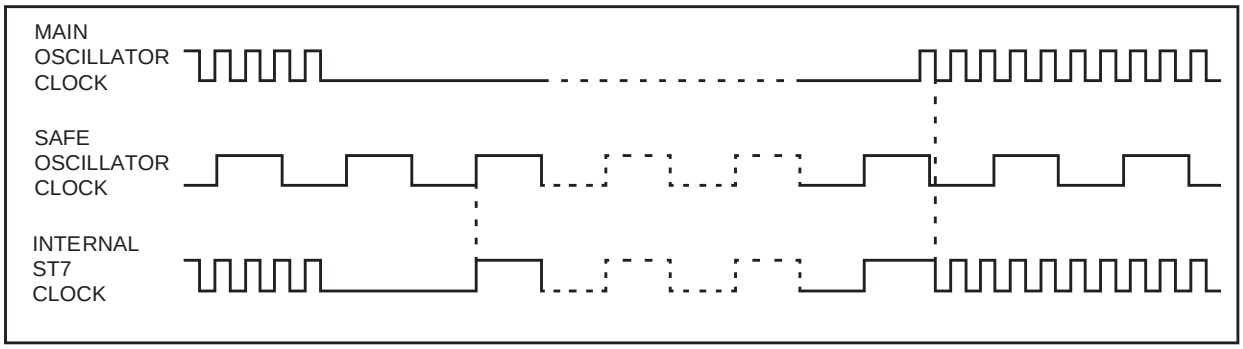


Figure 24. Safe Oscillator Function



4.4 SUPPLY, RESET AND CLOCK REGISTER DESCRIPTION

CLOCK RESET AND SUPPLY REGISTER (CRSR)

Read/Write

Reset Value: 000x 000x (00h)

7							0
0	0	0	LVD RF	0	CSS IE	CSS D	WDG RF

Bit 7:5 = **Reserved**, always read as 0.

Bit 4 = **LVDRF** *LVD reset flag*
This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero). See WDGRF flag description for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Bit 3 = **Reserved**, always read as 0.

Bit 2 = **CSSIE** *Clock security syst interrupt enable*
This bit enables the interrupt when a disturbance is detected by the Clock Security System (CSSD bit set). It is set and cleared by software.
0: Clock security system interrupt disabled
1: Clock security system interrupt enabled
When the CSS is disabled by OPTION BYTE, the CSSIE bit has no effect.

Bit 1 = **CSSD** *Clock security system detection*
This bit indicates that the safe oscillator of the Clock Security System block has been selected by hardware due to a disturbance on the main clock signal (f_{OSC}). It is set by hardware and cleared by a read of the CRSR register when the original oscillator recovers.
0: Safe oscillator is not active
1: Safe oscillator has been activated
When the CSS is disabled by OPTION BYTE, the CSSD bit value is forced to 0.

Bit 0 = **WDGRF** *Watchdog reset flag*
This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or a LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts).
Combined with the LVDRF flag information, the flag description is given by the following table.

RESET Sources	LVDRF	WDGRF
External RESET pin	0	0
Watchdog	0	1
LVD	1	X

Application notes

In case the LVDRF flag is not cleared upon another RESET type occurs (extern or watchdog), the LVDRF flag remains set to keep trace of the original failure.
In this condition, a watchdog reset can be detected by the software while an external reset not.

Table 4. Clock, Reset and Supply Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Bh	**CRSR** Reset Value	0	0	0	LVDRF x	0	CFIE 0	CSSD 0	WDGRF x

4.5 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:

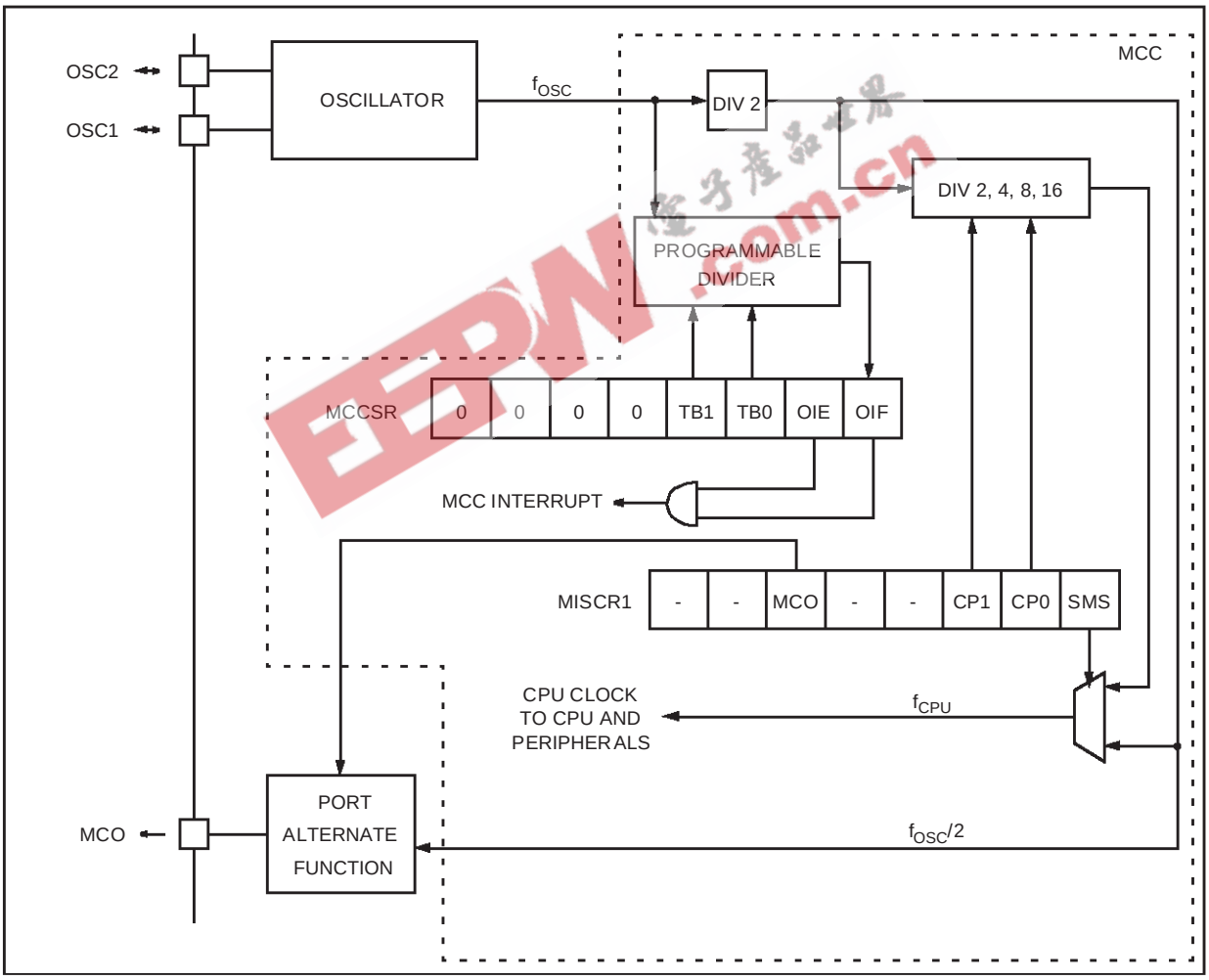
- a programmable CPU clock prescaler
- a time base counter with interrupt capability
- a clock-out signal to supply external devices

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 25. Main Clock Controller (MCC) Block Diagram



MAIN CLOCK CONTROLLER (Cont'd)

MISCELLANEOUS REGISTER 1 (MISCR1)

See section 6.2 on page 47.

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 5. 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

5 INTERRUPTS & POWER SAVING MODES

5.1 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in the Interrupt Mapping Table and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 26.
The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

– Normal processing is suspended at the end of the current instruction execution.

– The PC, X, A and CC registers are saved onto the stack.

– The I bit of the CC register is set to prevent additional interrupts.

– The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping Table for vector addresses).

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

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

Priority management

By default, the interrupt being serviced cannot be interrupted because the I bit is set by hardware when entering an interrupt routine.

If several interrupts are simultaneously pending, a hardware priority defines which one will be serviced first (see the Interrupt Mapping Table).

Non Maskable Software Interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on Figure 26.

Interrupts and Low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific mentioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT“ column in the Interrupt Mapping Table).

External Interrupts

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

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

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

If several input pins, connected to the same interrupt vector, are configured as interrupts, their signals are logically ANDed before entering the edge/level detection block.

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

Peripheral Interrupts

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

– The I bit of the CC register is cleared.

– The corresponding enable bit is set in the control register.

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

Clearing an interrupt request is done by:

– writing “0” to the corresponding bit in the status register or

– an access to the status register while the flag is set followed by a read or write of an associated register.

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

INTERRUPTS (Cont'd)

Figure 26. Interrupt Processing Flowchart

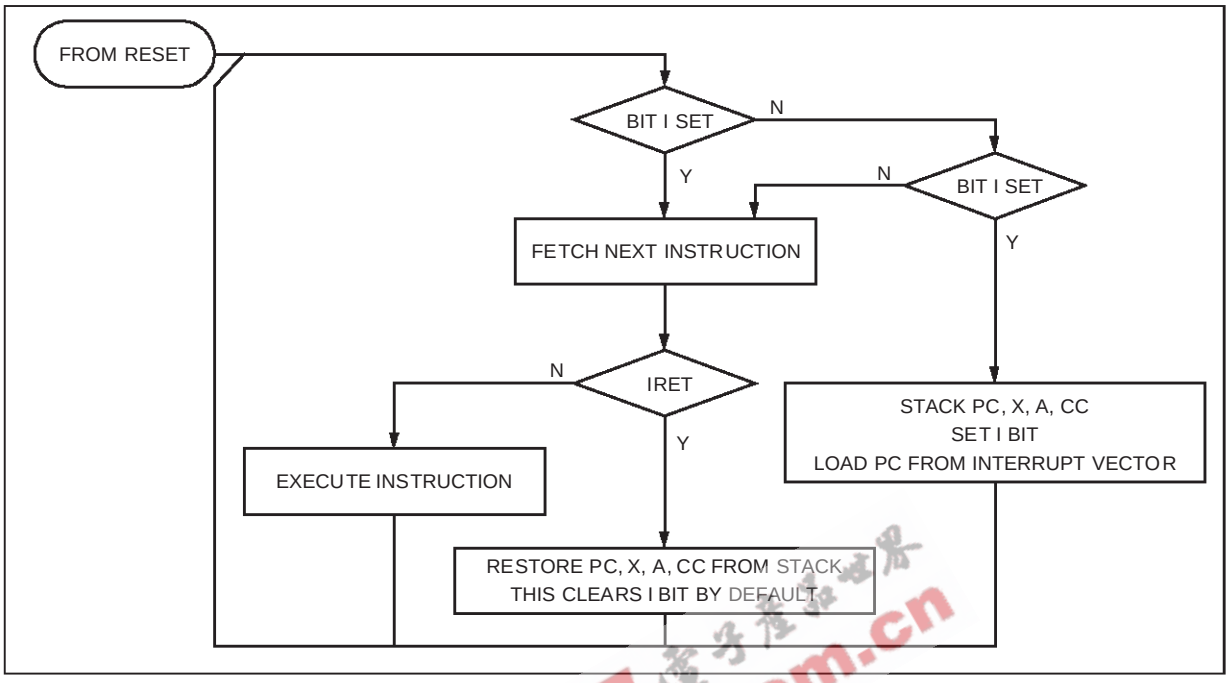


Table 6. Interrupt Mapping

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		Not used				FFFAh-FFFBh
1	MCC CSS	Main Clock Controller Time Base Interrupt or Clock Security System Interrupt	MCCSR CRSR		yes	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		Not used				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	Data-EEPROM	Data EEPROM Interrupt	EECSR			FFE4h-FFE5h
12		Not used		Lowest Priority		FFE2h-FFE3h
13						FFE0h-FFE1h

5.2 POWER SAVING MODES

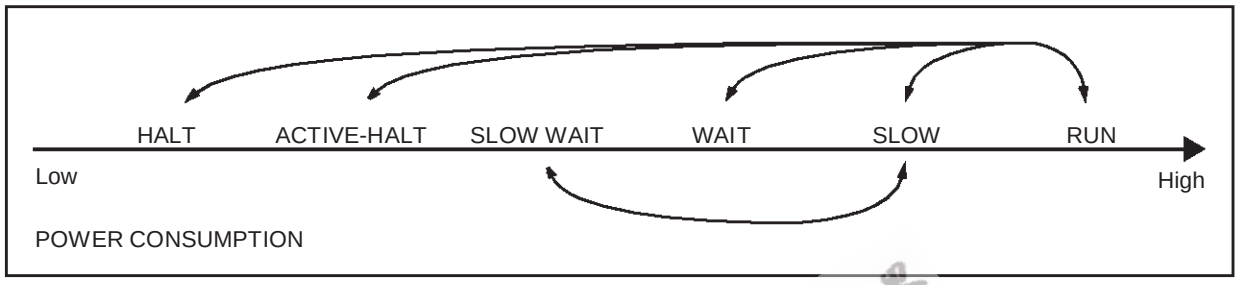
5.2.1 Introduction

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 27. Power saving mode consumption / transitions



5.2.2 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 29).

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 7).

When entering HALT modes, the I bit in the CC register is forced to 0 to enable interrupts.

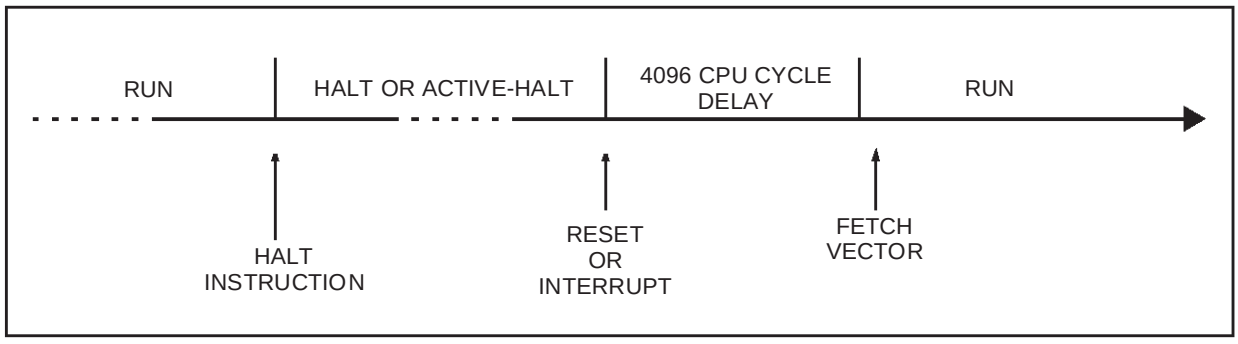
The MCU can exit HALT or ACTIVE-HALT modes on reception of an interrupt with Exit from Halt Mode capability or a reset (see Table 6 page 37). A 4096 CPU clock cycles delay is performed before the CPU operation resumes (see Figure 28).

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

Table 7. 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 28. 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 dedicated section 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.

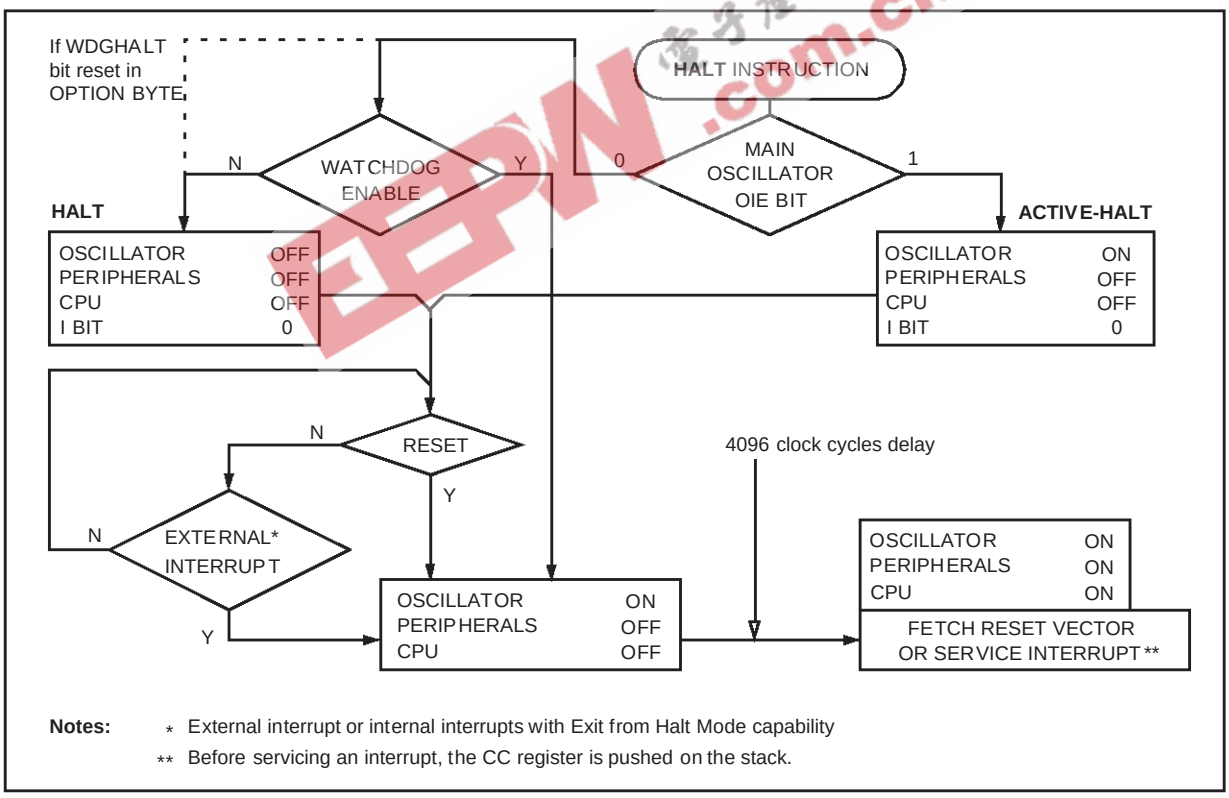
Specific ACTIVE-HALT mode

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 29. 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.

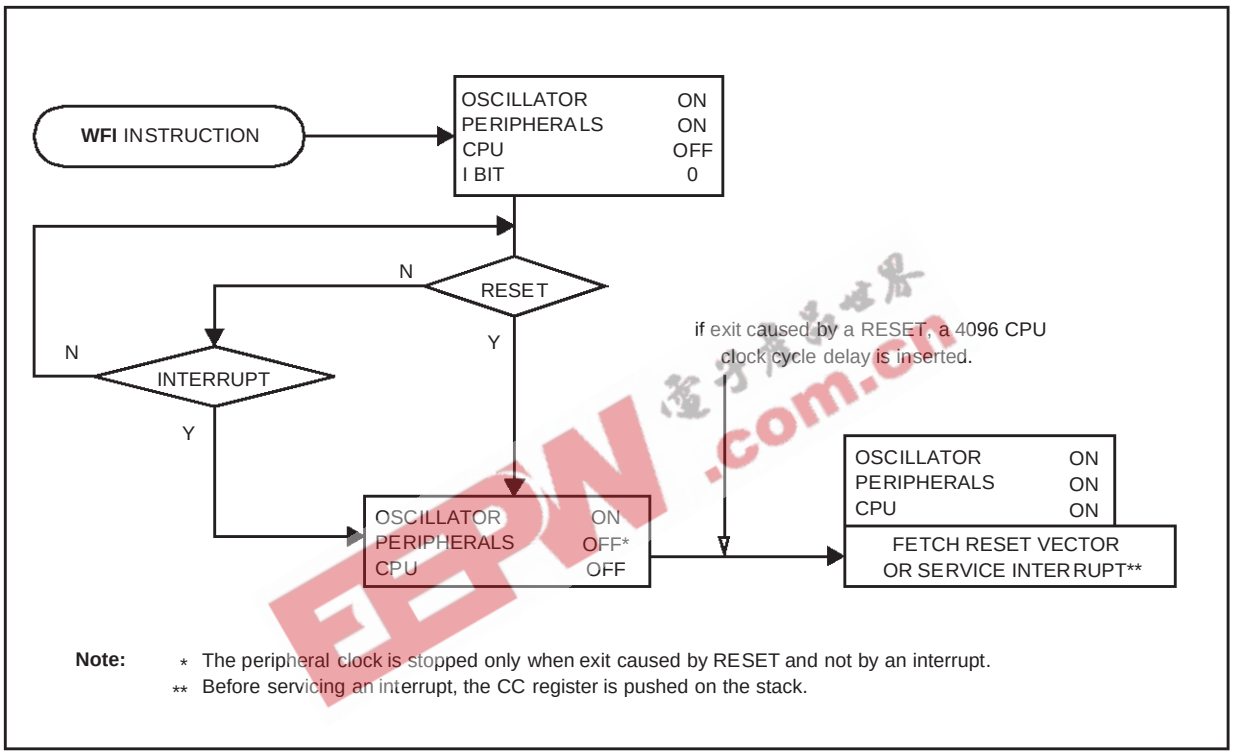
POWER SAVING MODES (Cont'd)

5.2.3 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 I bit of the CC register is forced to 0 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 30.

Figure 30. WAIT mode flow-chart



POWER SAVING MODES (Cont'd)

5.2.4 SLOW Mode

This mode has two targets:

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

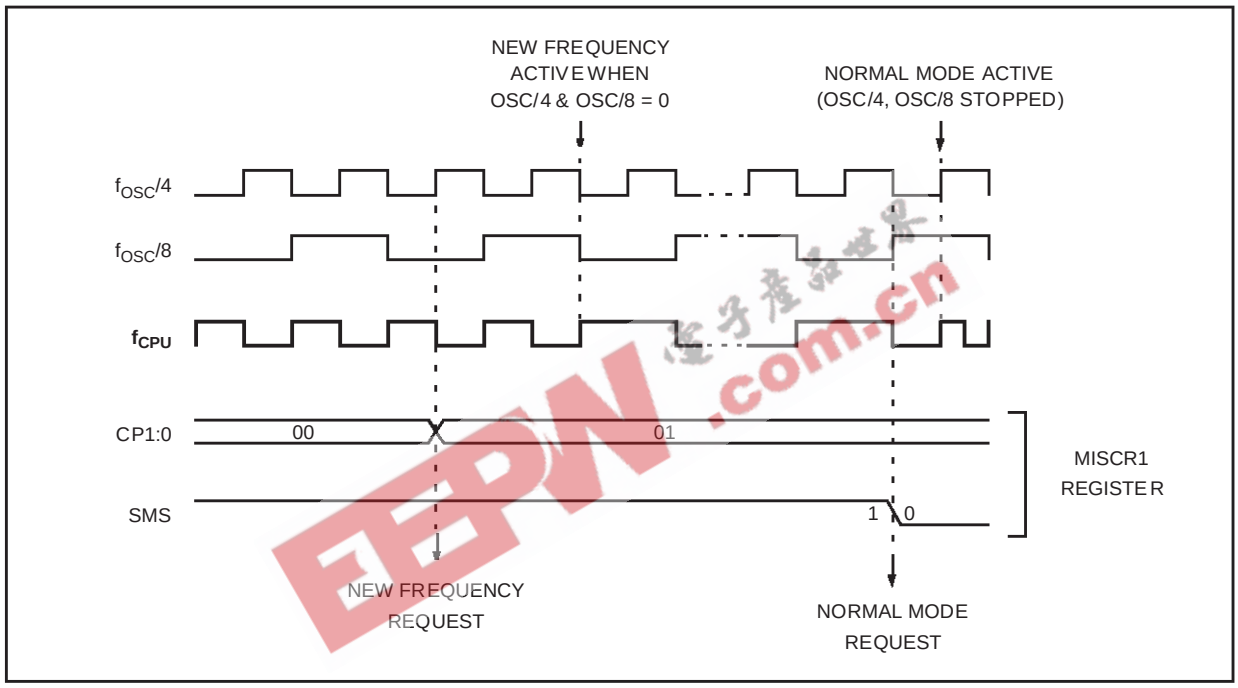
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by three bits in the 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 are clocked at this lower frequency.

Note: SLOW-WAIT mode is activated when entering the WAIT mode while the device is already in SLOW mode.

Figure 31. SLOW Mode: timing diagram for internal CPU clock transitions



6 ON-CHIP PERIPHERALS

6.1 I/O PORTS

6.1.1 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.

6.1.2 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 32

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 32. I/O Block Diagram

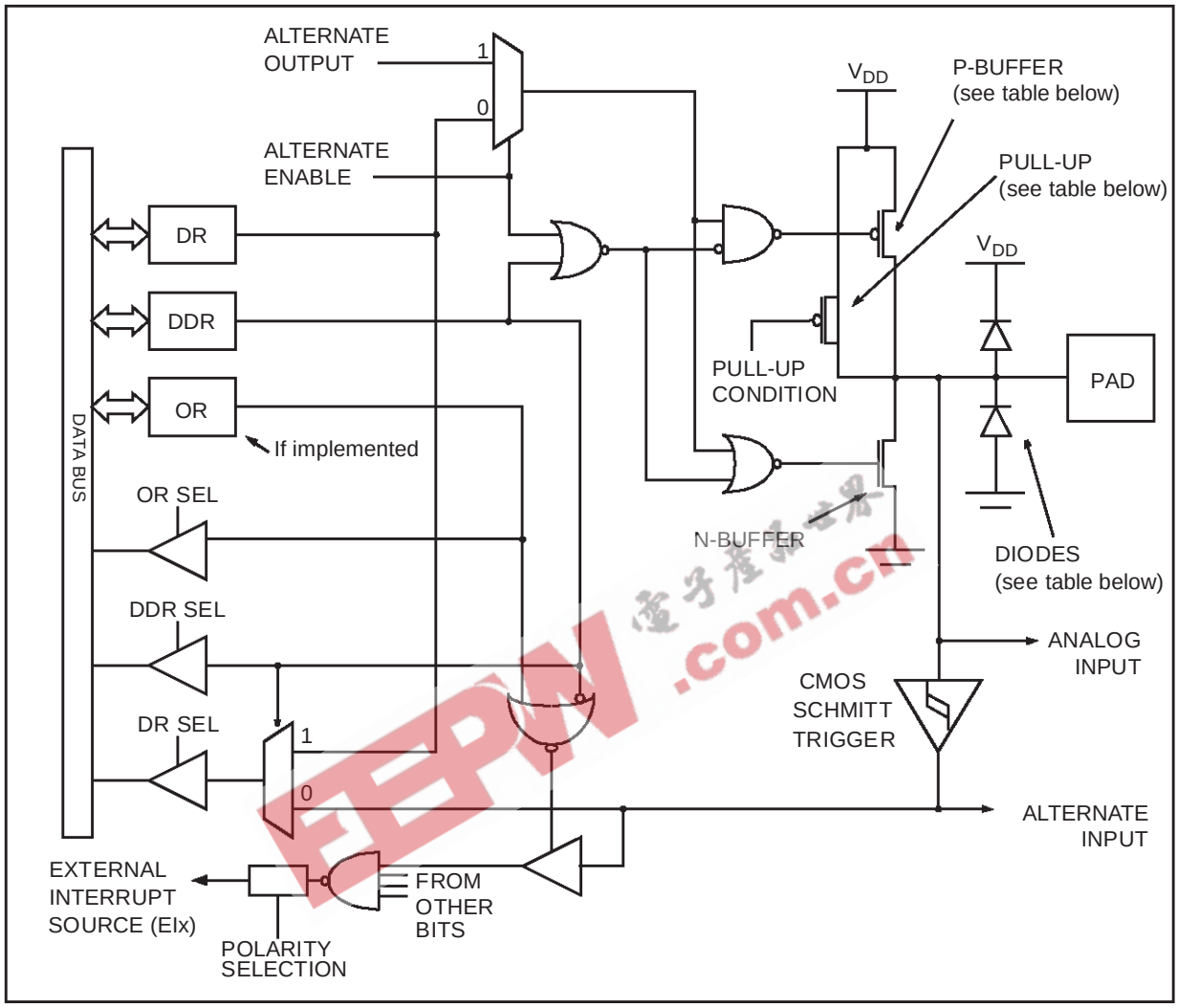


Table 8. Port Mode Options

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)

6.1.3 I/O Port Implementation

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

Standard Ports

PA5:4, PC7:0, PD7:0, PE7:4, PE1:0, PF7:6, PF4

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, PB6:4, PB2:0, PF1:0 (with pull-up)

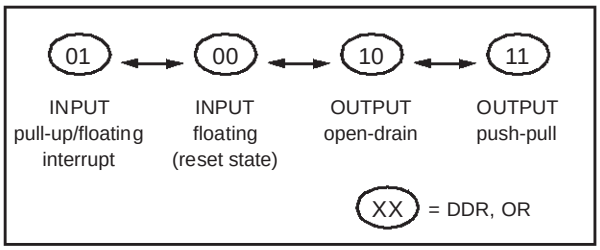
MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

PA3, PB7, 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 33 Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 33. Interrupt I/O Port State Transition



True Open Drain Ports

PA7:6

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

Table 9. Port Configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:6	floating		true open-drain	
	PA5:4	floating	pull-up	open drain	push-pull
	PA3	floating	floating interrupt	open drain	push-pull
	PA2:0	floating	pull-up interrupt	open drain	push-pull
Port B	PB7, PB3	floating	floating interrupt	open drain	push-pull
	PB6:4, PB2:0	floating	pull-up interrupt	open drain	push-pull
Port C	PC7:0	floating	pull-up	open drain	push-pull
Port D	PD7:0	floating	pull-up	open drain	push-pull
Port E	PE7:4, PE1:0	floating	pull-up	open drain	push-pull
Port F	PF7:6, PF4	floating	pull-up	open drain	push-pull
	PF2	floating	floating interrupt	open drain	push-pull
	PF1:0	floating	pull-up interrupt	open drain	push-pull

I/O PORTS (Cont'd)

6.1.4 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 10. 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** 1)								
0004h	**PCDR**	MSB							LSB
0005h	**PCDDR**								
0006h	**PCOR**								
0008h	**PBDR**	MSB							LSB
0009h	**PBDDR**								
000Ah	**PBOR** 1)								
000Ch	**PEDR**	MSB							LSB
000Dh	**PEDDR**								
000Eh	**PEOR** 1)								
0010h	**PDDR**	MSB							LSB
0011h	**PDDDR**								
0012h	**PDOR** 1)								
0014h	**PFDR**	MSB							LSB
0015h	**PFDDR**								
0016h	**PFOR**								

Notes:
1) The bits corresponding to unavailable pins are forced to 1 by hardware, this affects the reset status value.

6.2 MISCELLANEOUS REGISTERS

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

6.2.1 I/O Port Interrupt Sensitivity Description

The external interrupt sensitivity is controlled by the ISxx bits of the MISCR1 miscellaneous register. This control allows to have two fully independent external interrupt source sensitivities.

Each external interrupt source can be generated on four different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level

To guarantee correct functionality, the sensitivity bits in the MISCR1 register must be modified only when the I bit of the CC register is set to 1 (interrupt masked). See I/O port register and Miscellaneous register descriptions for more details on the programming.

6.2.2 I/O Port Alternate Functions

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

- Main clock signal (f_{CPU}) output on PF0
- A beep signal output on PF1 (with 3 selectable audio frequencies)
- SPI pin configuration:
  - $\overline{SS}$ pin internal control to use the PC7 I/O port function while the SPI is active.

These functions are described in detail in the Section 6.2.3 Miscellaneous Registers Description.

MISCELLANEOUS REGISTERS (Cont'd)

6.2.3 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) and EI3 (port B7..4). These 2 bits can be written only when the I bit of the CC register is set to 1 (interrupt disabled).

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

Bit 5 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the 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}/2 on 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) and EI1 (port F2..0). These 2 bits can be written only when the I bit of the CC register is set to 1 (interrupt disabled).

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

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

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
-	-	BC1	BC0	-	-	SSM	SSI

Bit 7:6 = **Reserved** *Must always be cleared*

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 3:2 = **Reserved** *Must always be cleared*

Bit 1 = **SSM** *$\overline{SS}$ mode selection*
It is set and cleared by software.
0: Normal mode - $\overline{SS}$ uses information coming from the $\overline{SS}$ pin of the SPI.
1: I/O mode, the SPI uses the information stored into bit SSI.

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.

Table 11. 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	0	0	BC1 0	BC0 0	0	0	SSM 0	SSI 0

6.3 WATCHDOG TIMER (WDG)

6.3.1 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.

6.3.2 Main Features

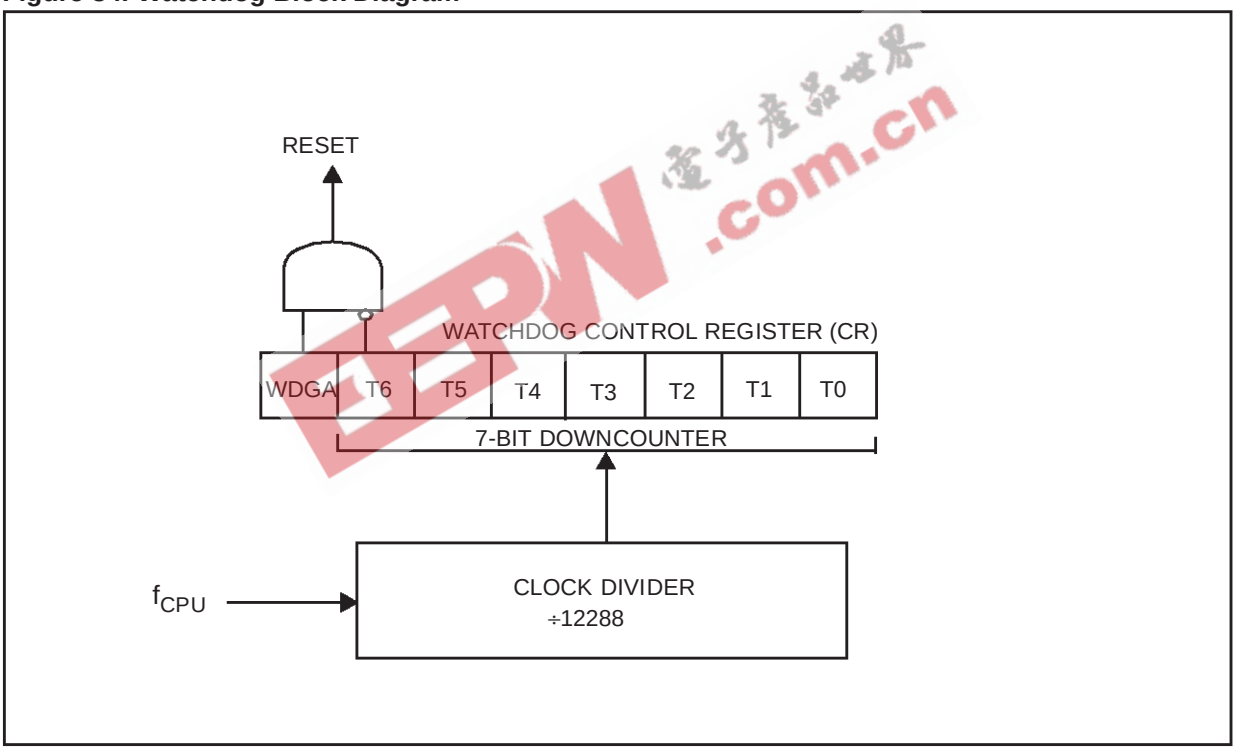
- Programmable timer (64 increments of 12288 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) after a HALT instruction or when the T6 bit reaches zero
- Hardware Watchdog selectable by option byte
- Watchdog Reset indicated by status flag (in versions with Safe Reset option only)

6.3.3 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 34. 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 12 .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 12.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.

6.3.4 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.

6.3.5 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).

6.3.6 Interrupts

None.

6.3.7 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 13. Watchdog Timer Register Map and Reset Values

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

6.4 16-BIT TIMER

6.4.1 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.

6.4.2 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 35.

***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'.

6.4.3 Functional Description

6.4.3.1 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).

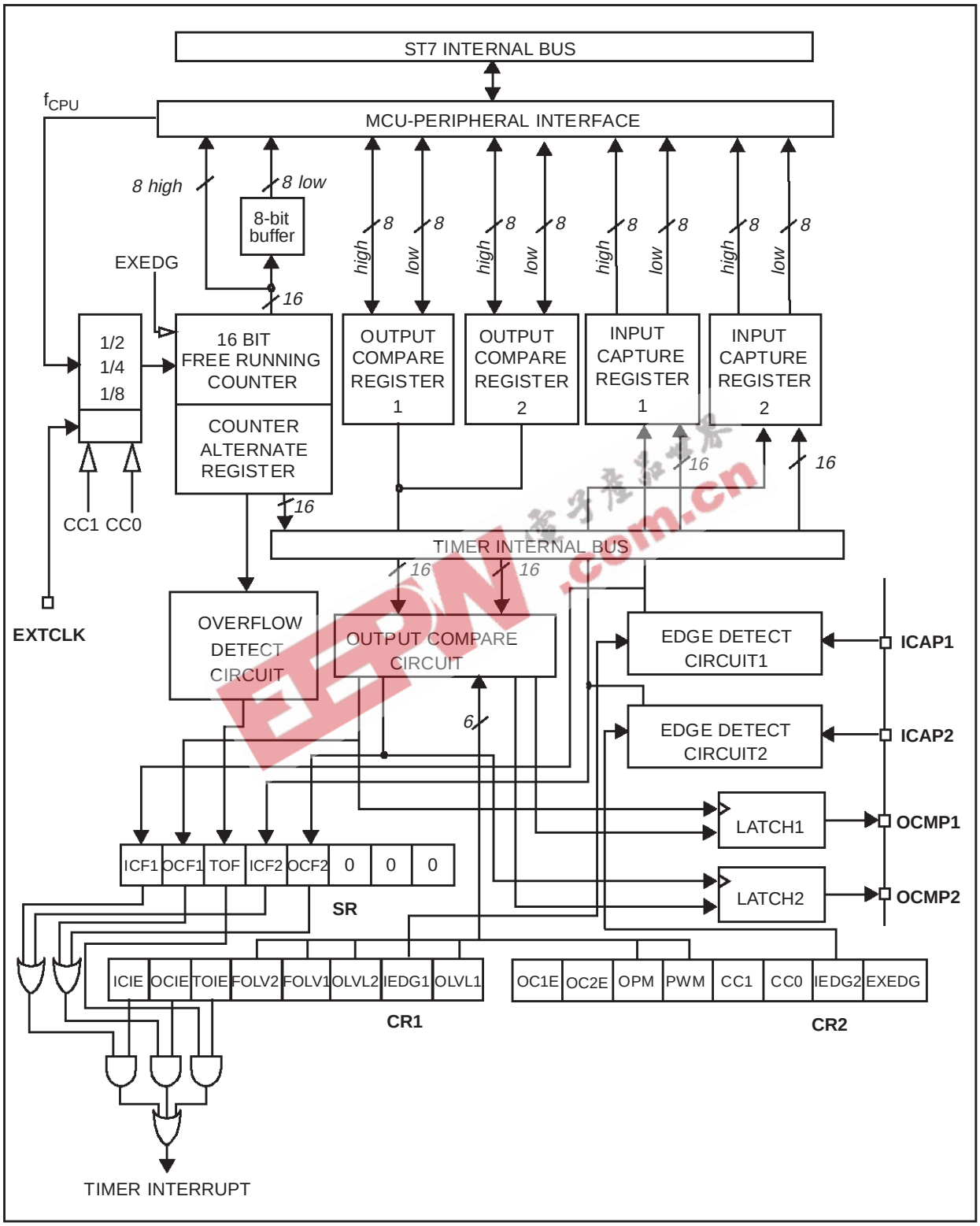
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (overflow flag), (see note at the end of paragraph titled 16-bit read sequence).

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 14 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.

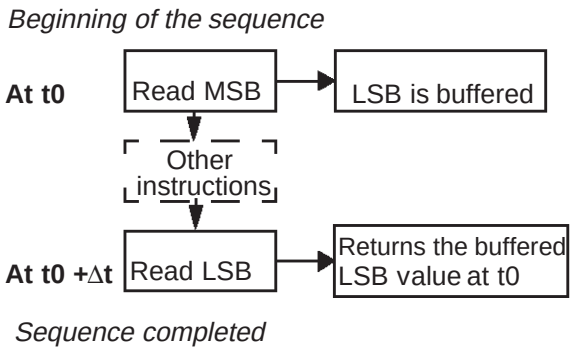
16-BIT TIMER (Cont'd)

Figure 35. 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).

6.4.3.2 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 36. Counter Timing Diagram, internal clock divided by 2

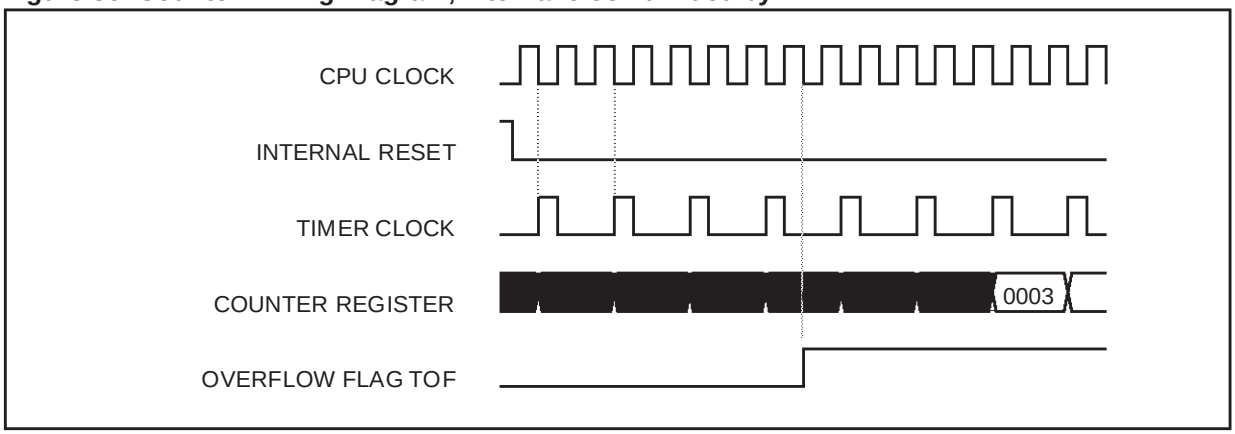


Figure 37. Counter Timing Diagram, internal clock divided by 4

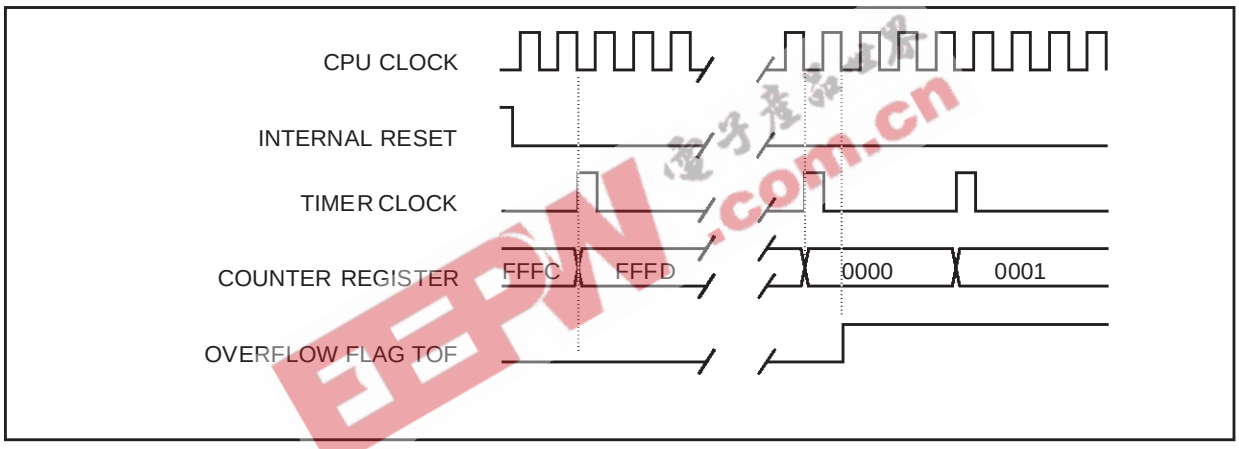
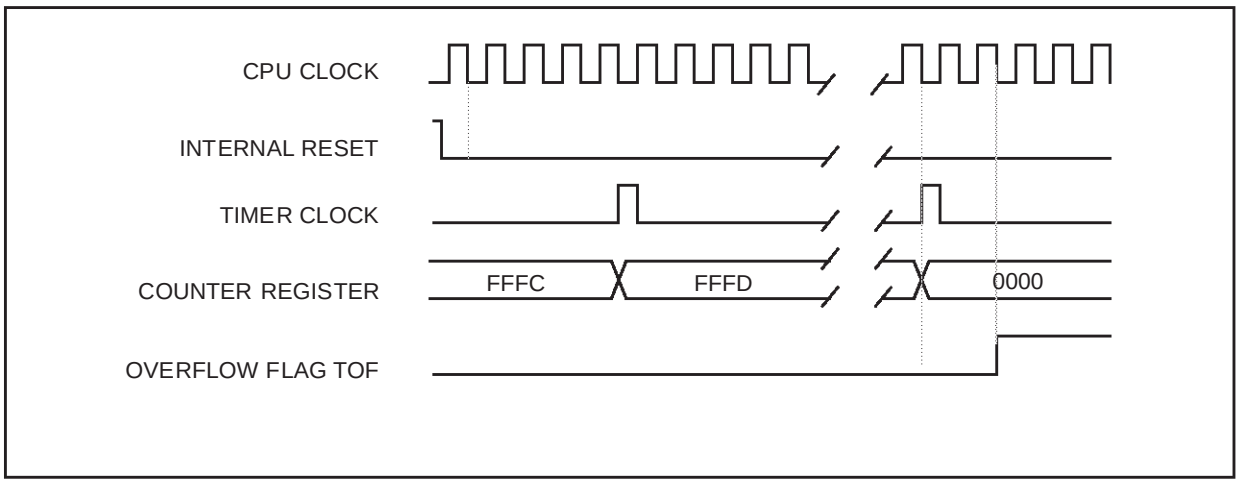


Figure 38. Counter Timing Diagram, internal clock divided by 8



16-BIT TIMER (Cont'd)

6.4.3.3 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 14 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 40).

– 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:

2. After reading the IC*i*HR register, transfer of input capture data is inhibited until the IC*i*LR register is also read.
3. The IC*i*R register always contains the free running counter value which corresponds to the most recent input capture.
4. The 2 input capture functions can be used together even if the timer also uses the output compare mode.
5. In One pulse Mode and PWM mode only the input capture 2 can be used.
6. The alternate inputs (ICAP1 & ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture process.
7. 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.
8. 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 39. Input Capture Block Diagram

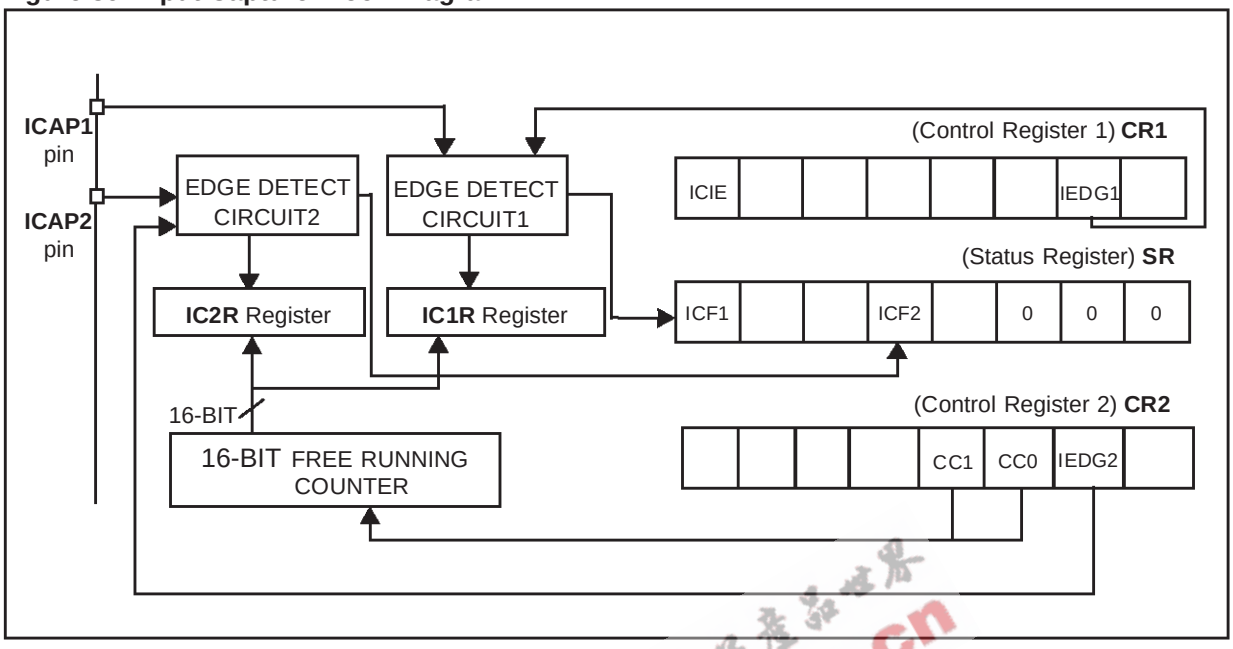
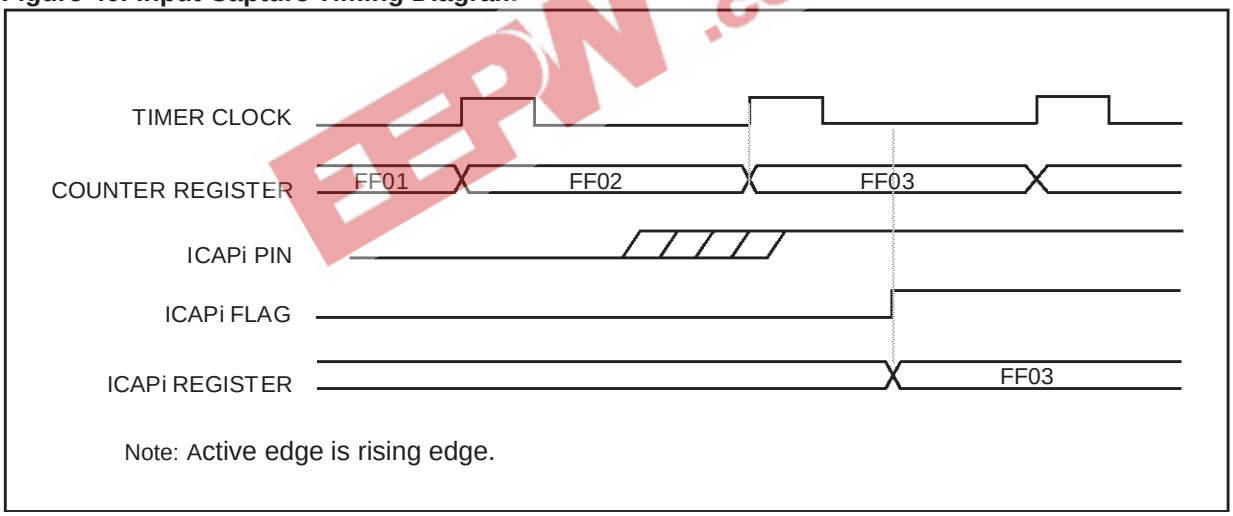


Figure 40. Input Capture Timing Diagram



Note: Active edge is rising edge.

16-BIT TIMER (Cont'd)

6.4.3.4 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 14 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:

$$\Delta OCiR = \frac{\Delta t * f_{CPU}}{PRESC}$$

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 14 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 42). 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 43).
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 41. Output Compare Block Diagram

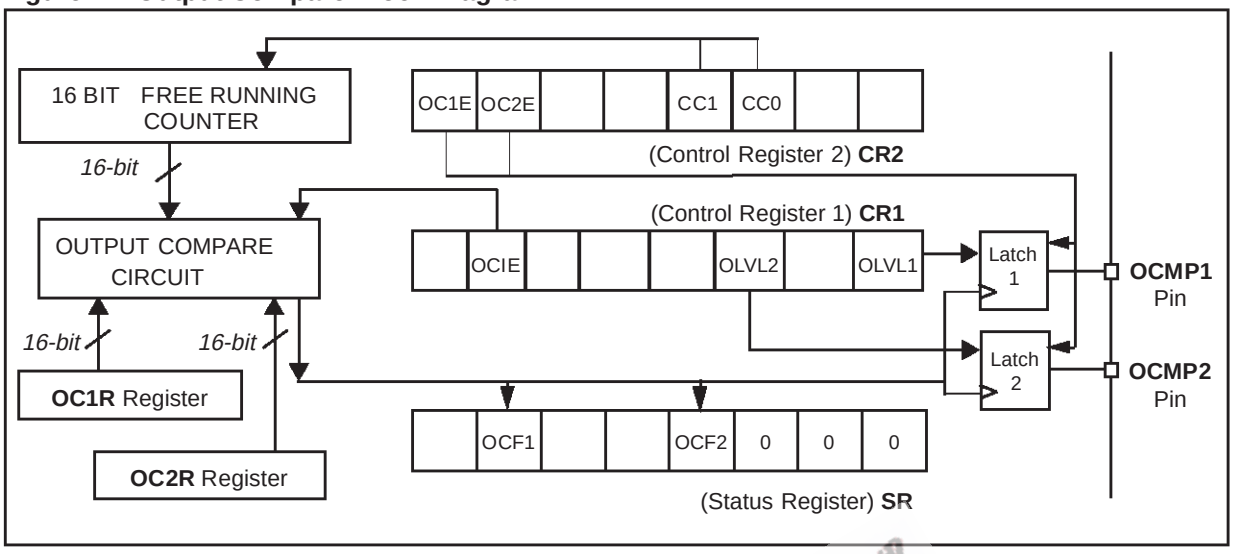


Figure 42. Output Compare Timing Diagram, Internal Clock Divided by 2

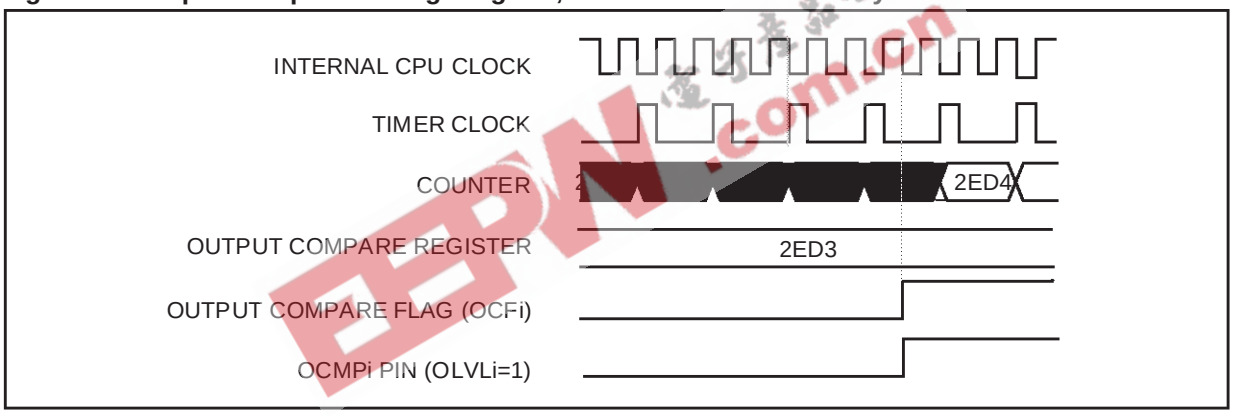
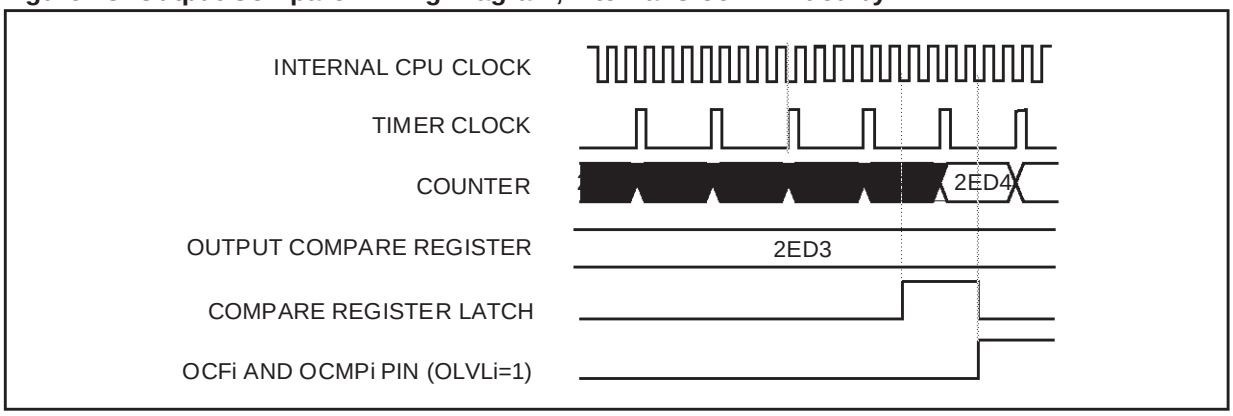


Figure 43. Output Compare Timing Diagram, Internal Clock Divided by 4



16-BIT TIMER (Cont'd)

6.4.3.5 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.

6.4.3.6 One Pulse Mode

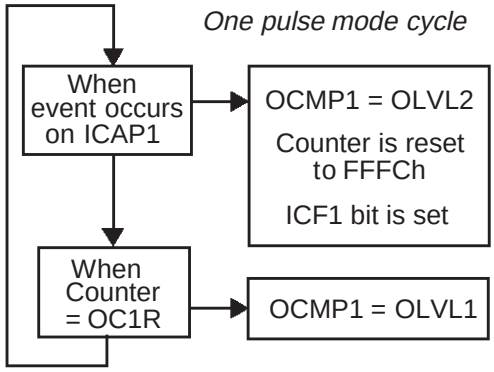
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

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 6.4.3.7).
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 14 Clock Control Bits).



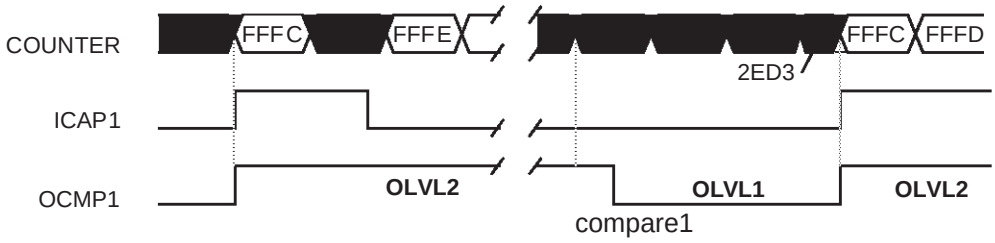
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

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 44).

Notes:

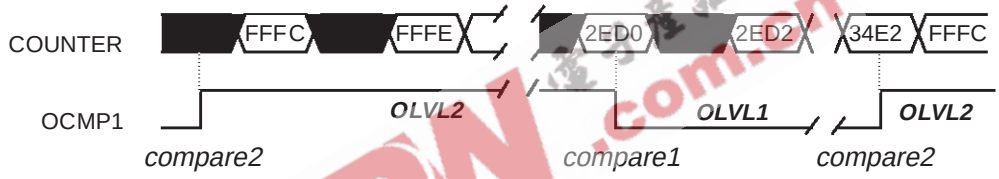
1. The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.
2. The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.
3. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both 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 44. One Pulse Mode Timing Example



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

Figure 45. Pulse Width Modulation Mode Timing Example



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

16-BIT TIMER (Cont'd)

6.4.3.7 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 14 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:

$$\text{OC}i\text{R Value} = \frac{t * f_{CPU}}{\text{PRESC}} - 5$$

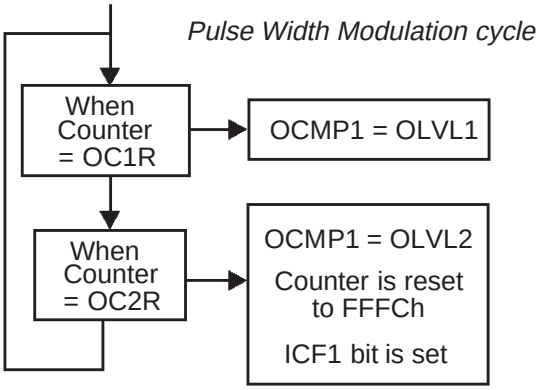
Where:

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 14 Clock Control Bits)

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 45).



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)

6.4.4 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.

6.4.5 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)

6.4.6 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 14. 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.

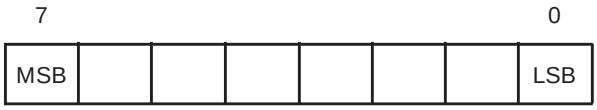
7							0
MSB							LSB

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.



OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

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

COUNTER HIGH REGISTER (CHR)

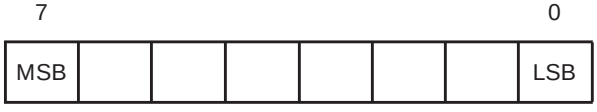
Read Only
Reset Value: 1111 1111 (FFh)

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

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

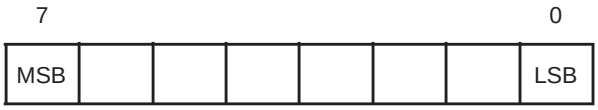
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the SR register clears the TOF bit.



ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

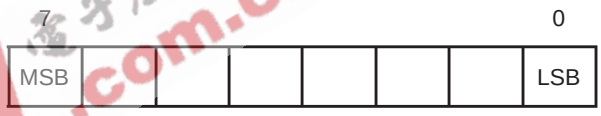
This is an 8-bit register that contains the high part of the counter value.



ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

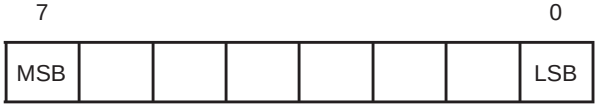
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to SR register does not clear the TOF bit in SR register.



INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

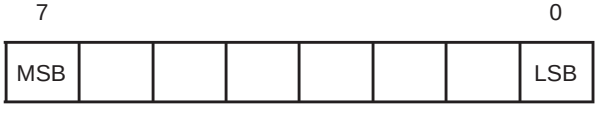
This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).



INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

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



16-BIT TIMER (Cont'd)

Table 15. 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 -

6.5 SERIAL PERIPHERAL INTERFACE (SPI)

6.5.1 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.

6.5.2 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.

6.5.3 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 46.

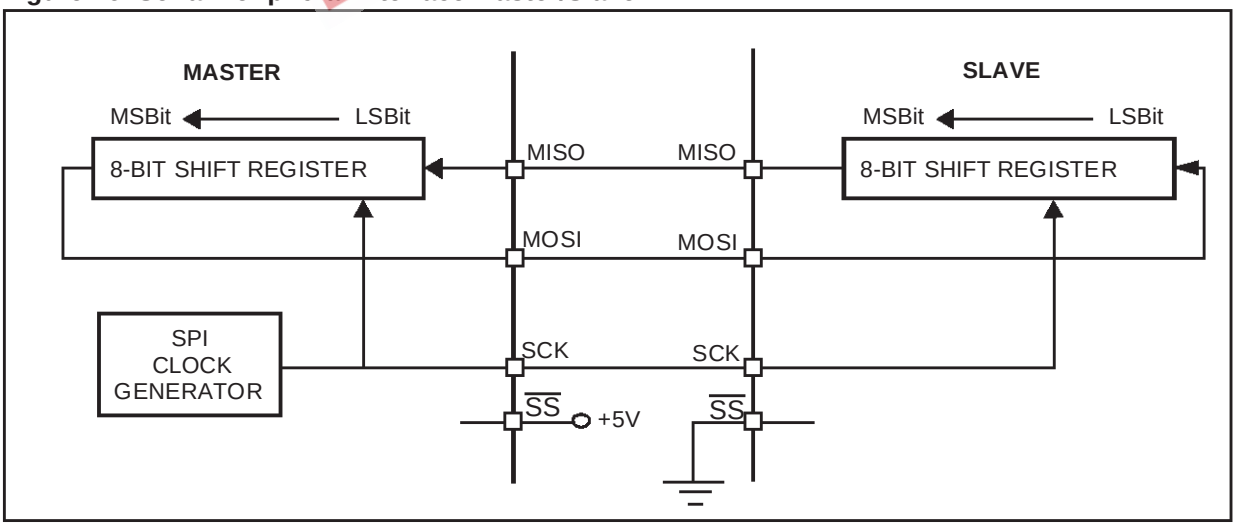
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

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.

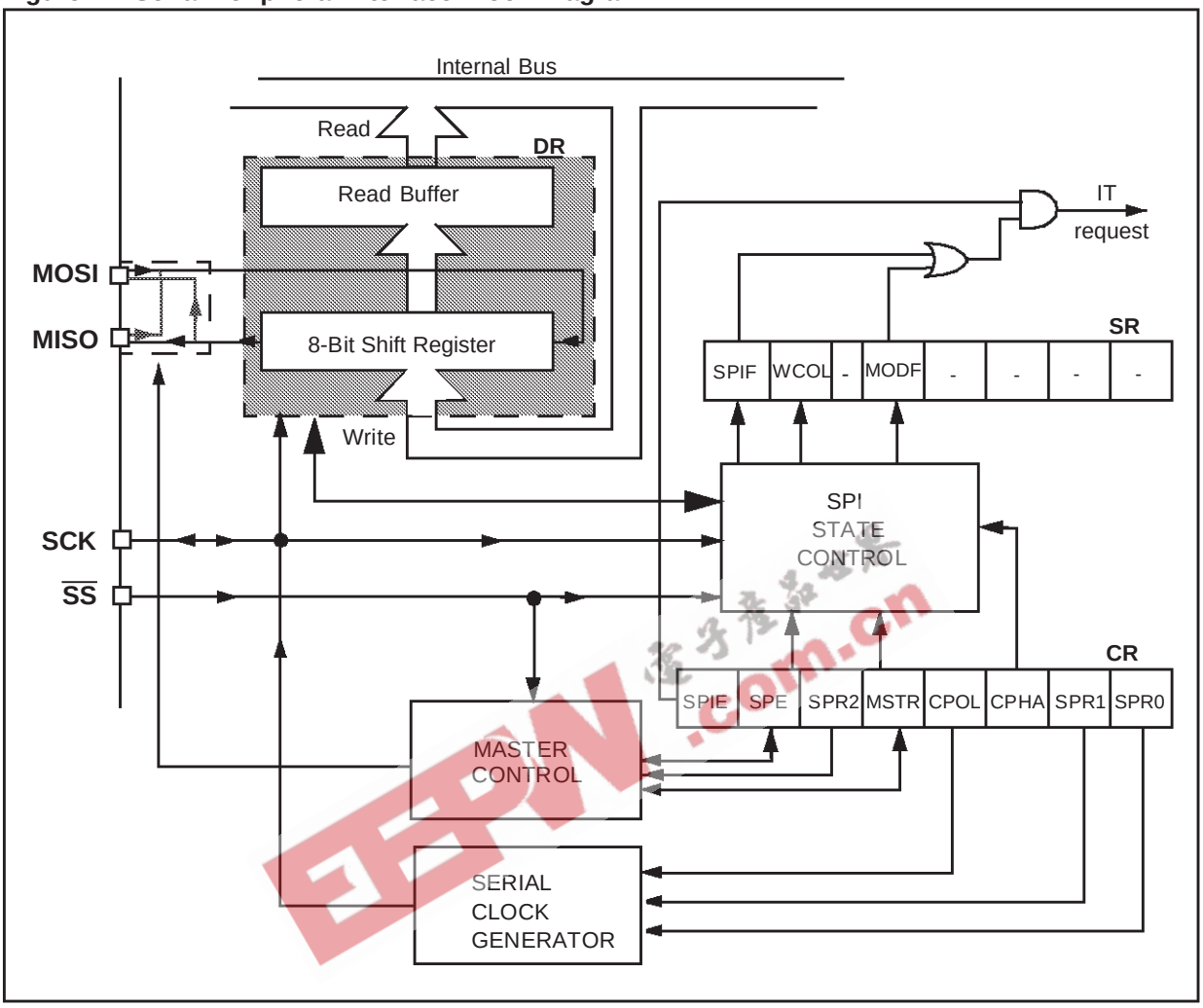
Four possible data/clock timing relationships may be chosen (see Figure 49) but master and slave must be programmed with the same timing mode.

Figure 46. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 47. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

6.5.4 Functional Description

Figure 46 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 6.5.7for the bit definitions.

6.5.4.1 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 49).
- The $\overline{SS}$ 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 $\overline{SS}$ 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 register are inhibited until the SR register is read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

6.5.4.2 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 49.

– 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 6.5.4.6).

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 6.5.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

6.5.4.3 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 49, 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 48).

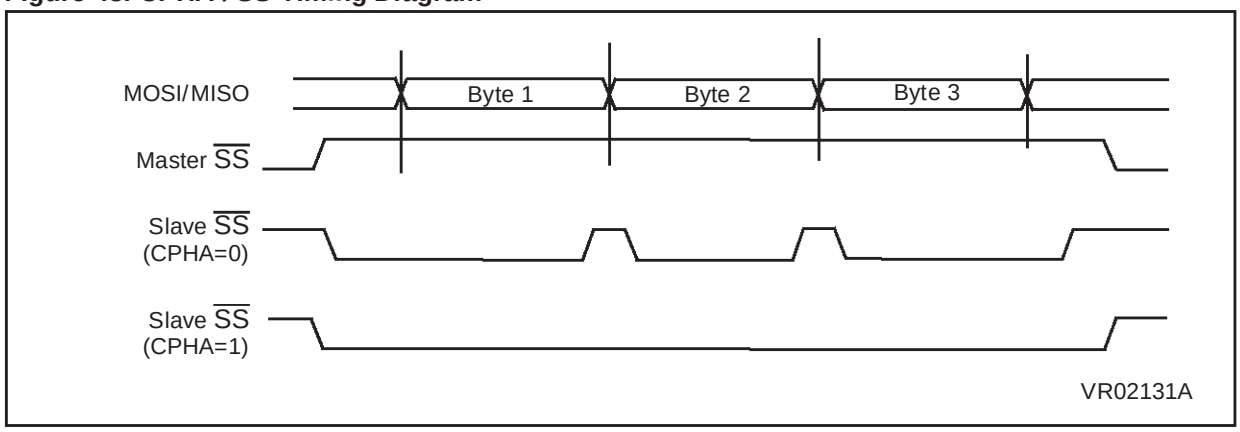
CPHA bit is reset

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 48).

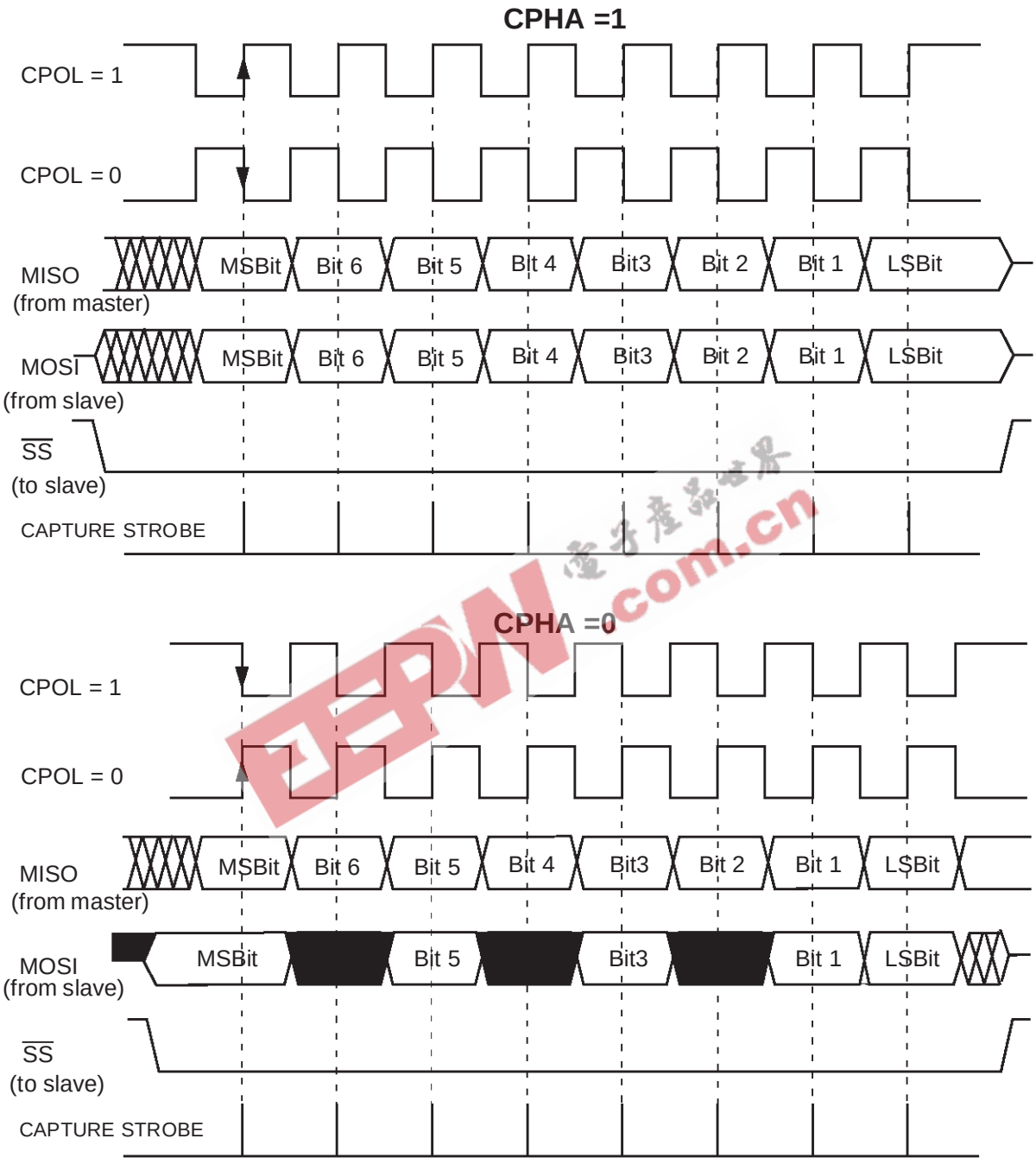
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its DR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the DR without producing a write collision.

Figure 48. CPHA / $\overline{SS}$ Timing Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 49. 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)

6.5.4.4 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.

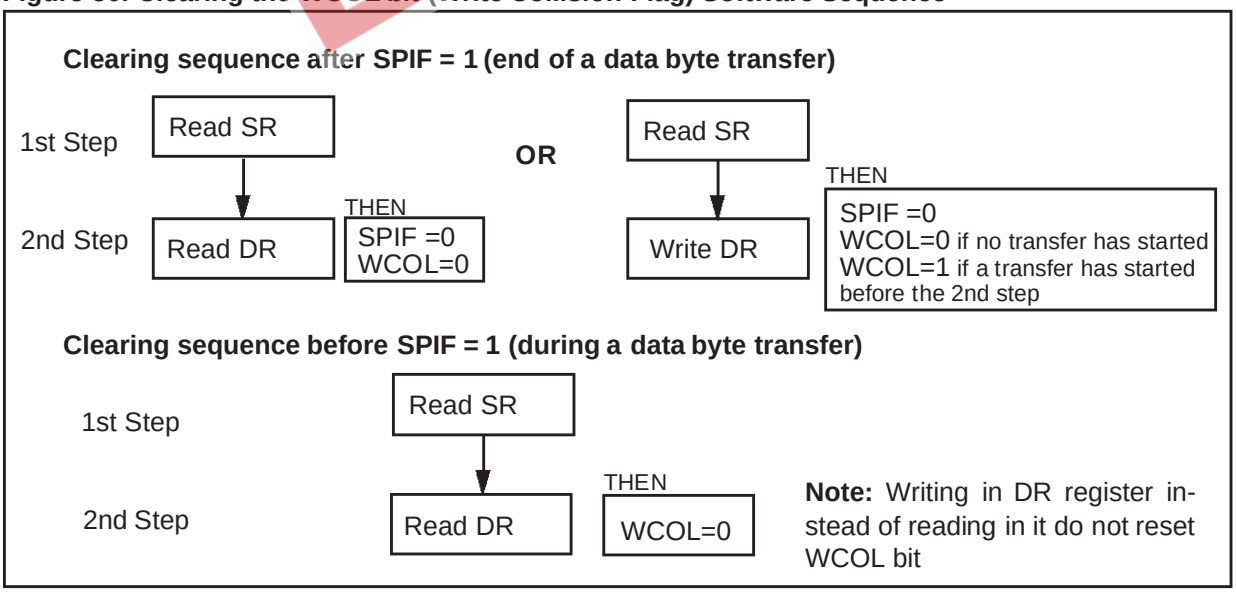
WCOL bit

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 50).

Figure 50. Clearing the WCOL bit (Write Collision Flag) Software Sequence



SERIAL PERIPHERAL INTERFACE (Cont'd)

6.5.4.5 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.

6.5.4.6 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)

6.5.4.7 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 51).

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.

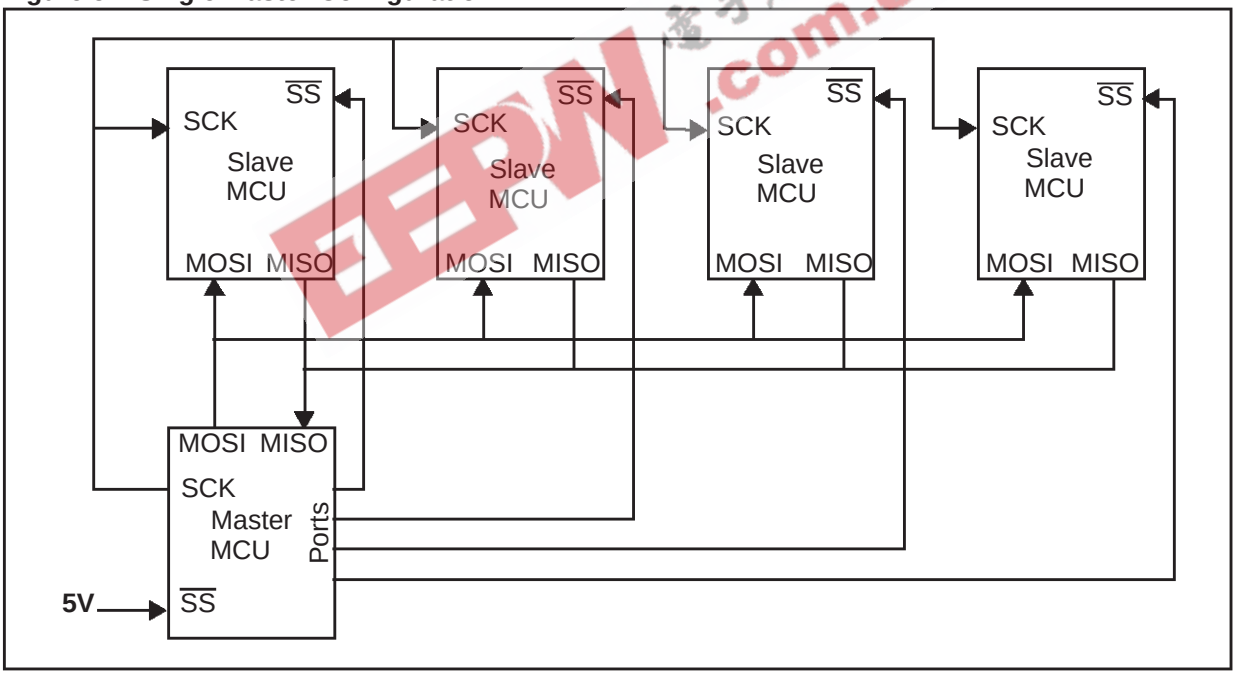
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

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 51. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

6.5.5 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.

6.5.6 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)

6.5.7 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 6.5.4.5 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 16.
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 6.5.4.5 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 16. 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 50).
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 6.5.4.5 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 47).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 17. 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

6.6 SERIAL COMMUNICATIONS INTERFACE (SCI)

6.6.1 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.

6.6.2 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

6.6.3 General Description

The interface is externally connected to another device by two pins (see Figure 53):

– 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.

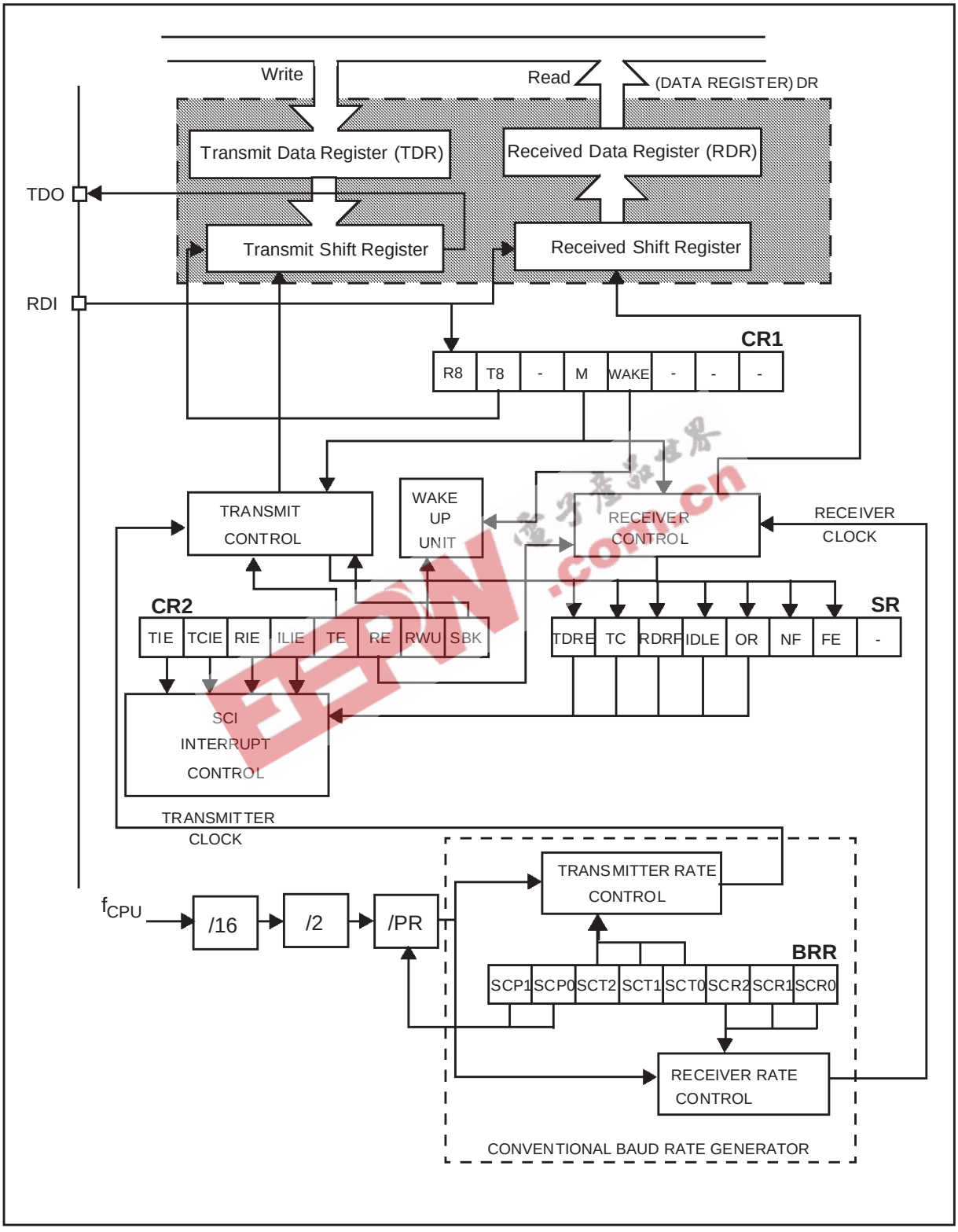
This interface uses two types of baud rate generator:

– 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 52. SCI Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

6.6.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 52. 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 6.6.7for the definitions of each bit.

6.6.4.1 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 52).

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

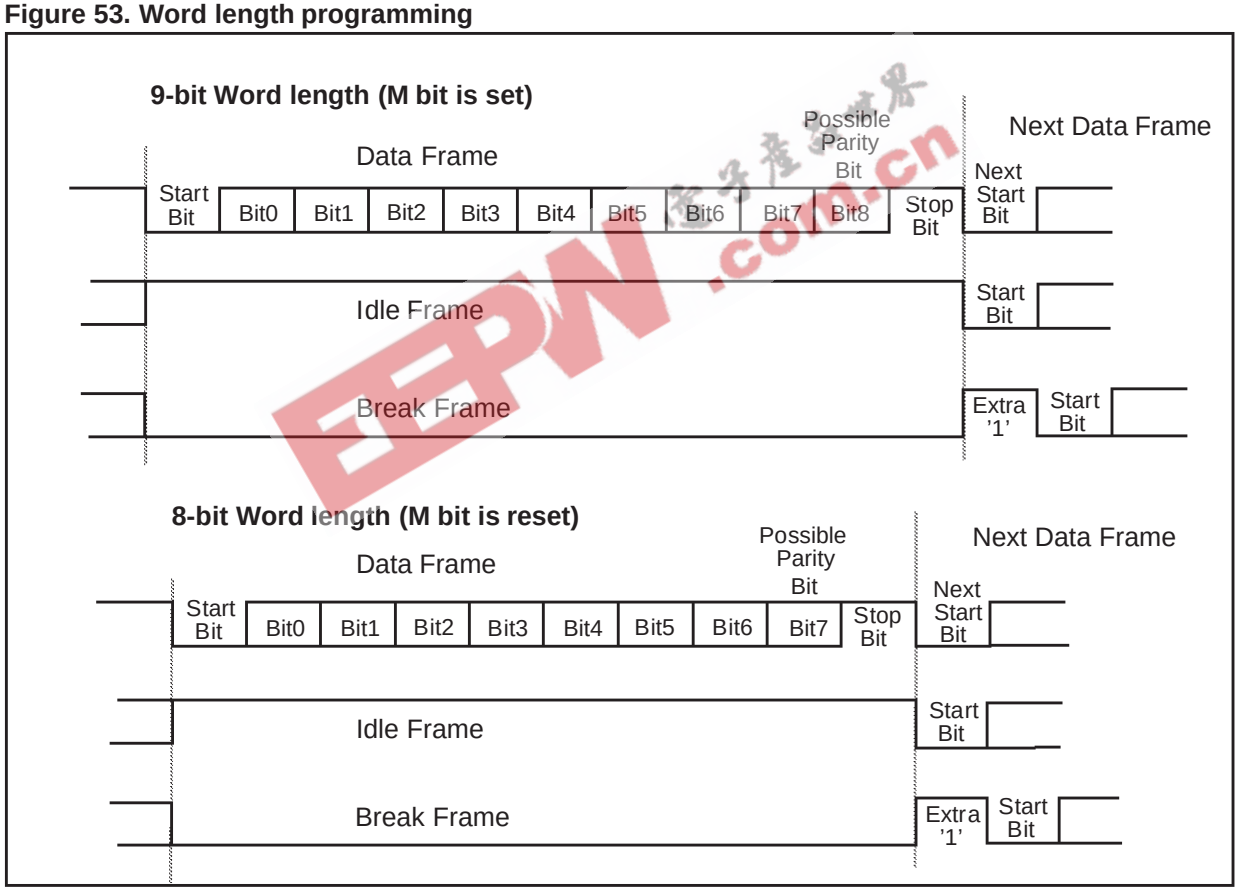
The TDO pin is in high state during the stop bit.

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 53. Word length programming



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

6.6.4.2 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 52).

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 53).

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)

6.6.4.3 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 52).

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

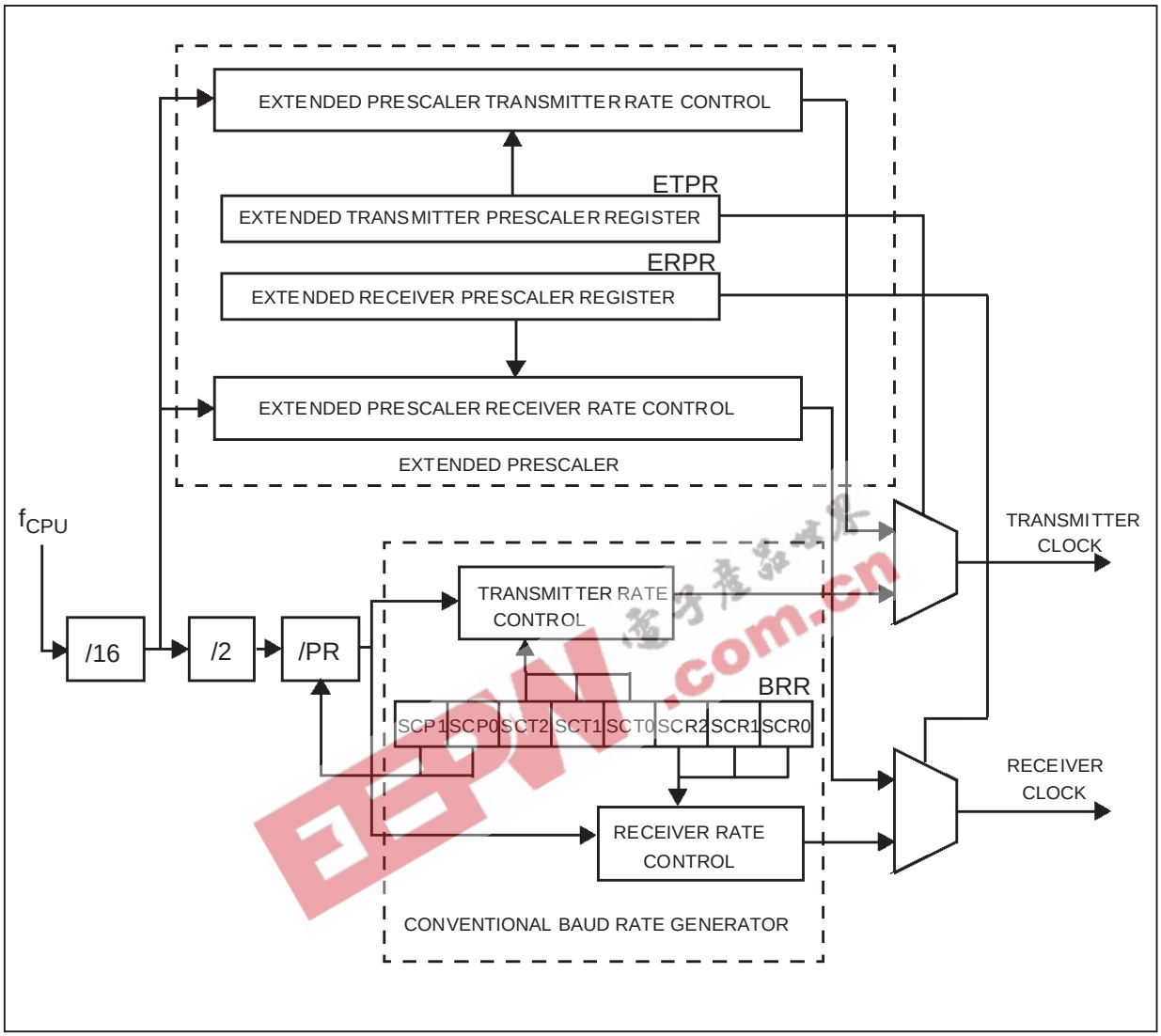
– Data is transferred from the Shift register to the DR register.

– 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 54. SCI Baud Rate and Extended Prescaler Block Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

6.6.4.4 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.

6.6.4.5 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 54.

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)

6.6.4.6 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)

6.6.5 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.

6.6.6 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)

6.6.7 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 52).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 52).

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 54) 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 54) 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 18. 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	SOG 0	0	VPOL x	2FHDET x	HVSEL x	VCORDIS x	CLPINV x	BLKINV x
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	**SCIPBRR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	**SCIPBRT** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

6.7 8-BIT A/D CONVERTER (ADC)

6.7.1 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.

6.7.2 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 56.

6.7.3 Functional Description

6.7.3.1 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 55. Recommended Ext. Connections

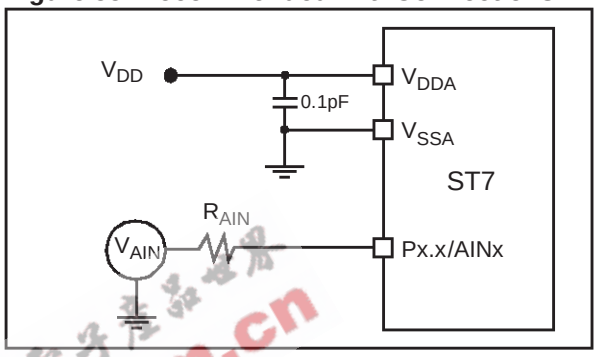
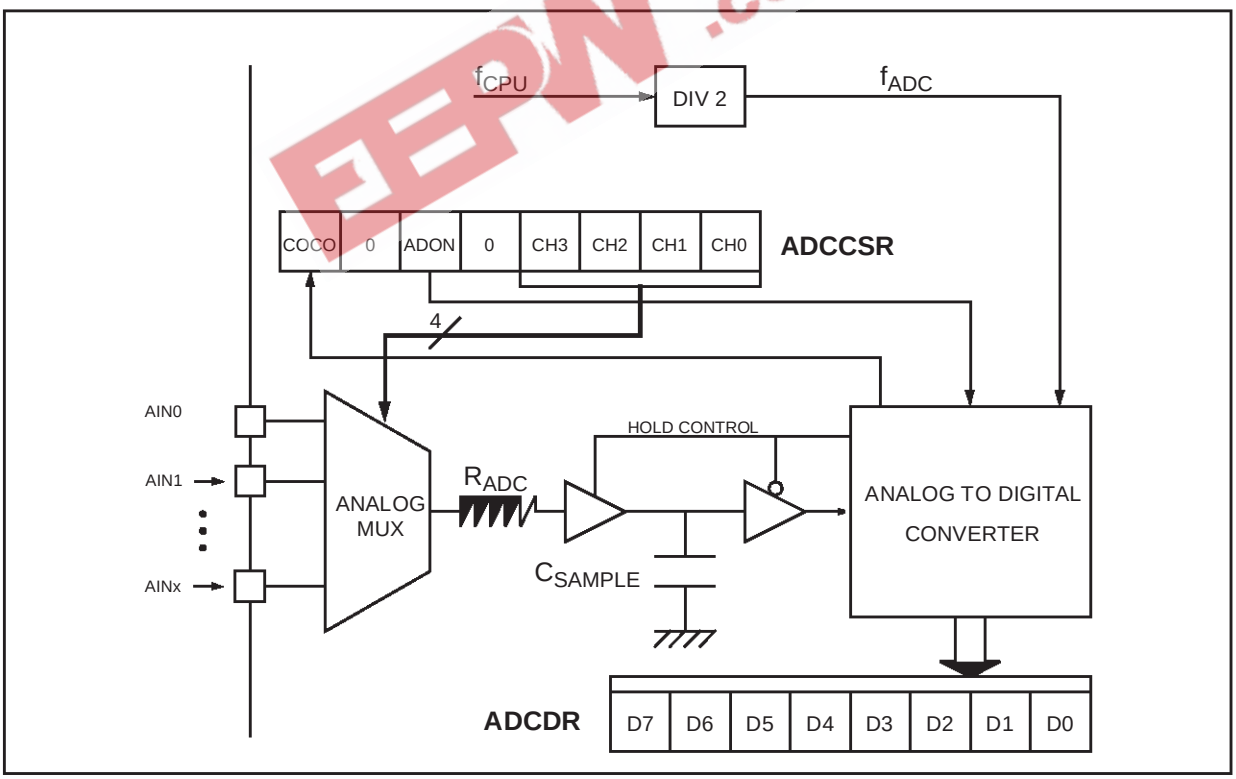


Figure 56. ADC Block Diagram



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

6.7.3.2 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.

6.7.3.3 A/D Conversion Phases

The A/D conversion is based on two conversion phases as shown in Figure 57:

- 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.

6.7.3.4 Software Procedure

Refer to the control/status register (CSR) and data register (DR) in Section 6.7.6 for the bit definitions and to Figure 57 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:

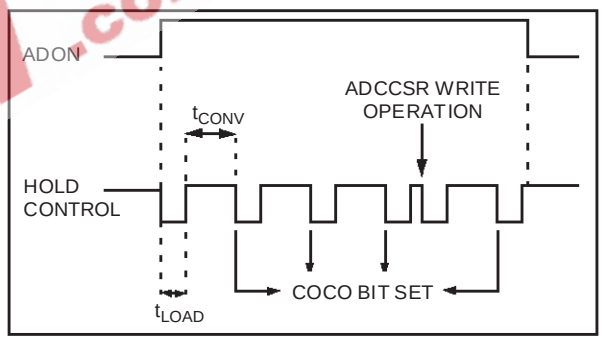
– Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

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 57. ADC Conversion Timings



6.7.4 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.

6.7.5 Interrupts

None

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

6.7.6 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 CONVERTER (ADC) (Cont'd)

Table 19. 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** Reset Value	COCO 0	0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0

7 INSTRUCTION SET

7.1 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 20. ST7 Addressing Mode Overview

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

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

ST7 ADDRESSING MODES (Cont'd)

7.1.1 Inherent

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

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

7.1.2 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

7.1.3 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.

7.1.4 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.

7.1.5 Indirect (Short, Long)

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

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

Indirect (short)

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

Indirect (long)

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

ST7 ADDRESSING MODES (Cont'd)

7.1.6 Indirect Indexed (Short, Long)

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

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

Indirect Indexed (Short)

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

Indirect Indexed (Long)

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

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

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

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

7.1.7 Relative mode (Direct, Indirect)

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

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

The relative addressing mode consists of two sub-modes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

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

7.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

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

Using a pre-byte

The instructions are described with one to four bytes.

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

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

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

8 ELECTRICAL CHARACTERISTICS

8.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices for protecting the inputs against damage due to high static voltages, however it is advisable to take normal precautions to avoid applying any voltage higher than the specified maximum rated voltages.

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

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

$T_J = T_A + P_D \times RthJA$

Where:
T_A = Ambient Temperature.
RthJA = Package thermal resistance (junction-to ambient).
$P_D = P_{INT} + P_{PORT}$.
$P_{INT} = I_{DD} \times V_{DD}$ (chip internal power).
P_{PORT} = Port power dissipation determined by the user)

Symbol	Ratings	Value	Unit
$V_{DD} - V_{SS}$	Supply voltage	6.5	V
$V_{DDA} - V_{SSA}$	Analog reference voltage $V_{DDA} > V_{SS}$	6.5	V
$\|V_{DD\_i} - V_{DD\_j}\|$ $\|V_{DD\_i} - V_{DDA}\|$	Max. variations (power line)	50	mV
$\|V_{SS\_i} - V_{SS\_j}\|$ $\|V_{SS\_i} - V_{SSA}\|$	Max. variations (ground line)	50	mV
V_{IN}	Input voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_{OUT}	Output voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
ESD	ESD susceptibility	2000	V
I_{VDD_i}	Total current into V_{DD_i} (source)	150	mA
I_{VSS_i}	Total current out of V_{SS_i} (sink)	150	

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

General Warning:
Directly connecting the RESET and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or the program counter is corrupted (by an expected change to the I/O configuration). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (10KΩ typical).

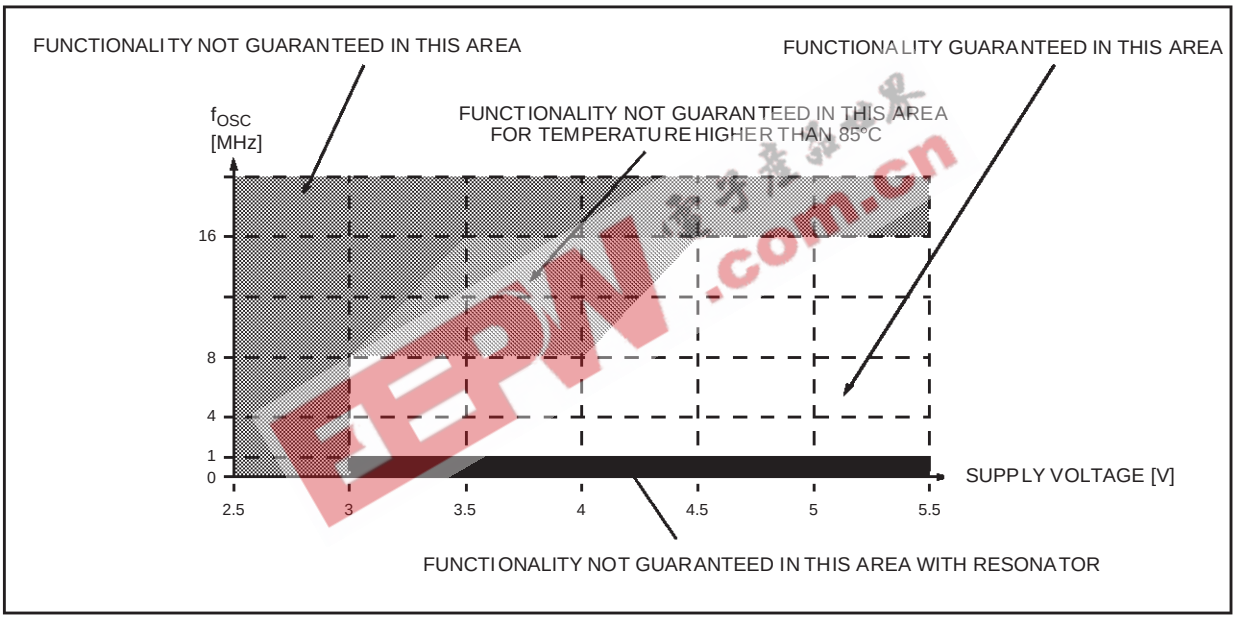
Thermal Characteristics

Symbol	Ratings		Value	Unit
R_{thJA}	Package thermal resistance	TQFP64 SDIP56 TQFP44 SDIP42	60 TBD TBD TBD	°C/W
T_{Jmax}	Max. junction temperature		150	°C
T_{STG}	Storage temperature range		-65 to +150	°C
PD	Power dissipation		500	mW

8.2 RECOMMENDED OPERATING CONDITIONS

GENERAL						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{DD}	Supply voltage	see Figure 58	3.0		5.5	V
f_{OSC}	Resonator oscillator frequency	$V_{DD} \geq 3.5V$	1		16	MHz
		$V_{DD} \geq 3.0V$	1		8	
	External clock source	$V_{DD} \geq 3.5V$	0 [2]		16	
		$V_{DD} \geq 3.0V$	0 [2]		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 58. f_{OSC} Maximum Operating Frequency Versus V_{DD} Supply Voltage [3]



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) A/D operation and resonator oscillator start-up are not guaranteed below 1MHz.

3) Operating conditions T_A=-40 to +85°C. The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

8.3 DC ELECTRICAL CHARACTERISTICS

Recommended operating conditions with T_A=-40 to +85°C, V_{DD}-V_{SS}=5V unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
I_{DD}	Supply current in RUN mode [2]	f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz			TBD	mA
	Supply current in SLOW mode [2]	f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz			TBD	
	Supply current in WAIT mode [3]	f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8MHz, f_{CPU} = 4 MHz f_{OSC} = 16MHz, f_{CPU} = 8 MHz			TBD	
	Supply current in SLOW WAIT mode [3]	f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz			TBD	
	Supply current in HALT mode [4]	I_{LOAD} = 0mA (current on I/Os)			TBD	µA
V_{RM}	Data retention mode [5]	HALT mode	2			V

8.4 GENERAL TIMING CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{INST}	Instruction time		2		12	t_{CPU}
t_{IRT}	Interrupt reaction time	$t_{IRT} = \Delta t_{INST} + 10$ [6]	10		22	t_{CPU}

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) CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS}, all peripherals switched off; clock input (OSC1) driven by external square wave.

3) All I/O pins in input mode with a static value at V_{DD} or V_{SS}, all peripherals switched off; clock input (OSC1) driven by external square wave.

4) All I/O pins in input mode with a static value at V_{DD} or V_{SS}, LVD disabled.

5) Data based on characterization results, not tested in production.

6) Δt_{INST} is the number of t_{CPU} to finish the current instruction execution.

8.5 I/O PORT CHARACTERISTICS

Recommended operating conditions
with T_A=-40 to +85°C and 4.5V<V_{DD}-V_{SS}<5.5V unless otherwise specified.

I/O PORT PINS						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{IL}	Input low level voltage [2]	3V<V_{DD}-V_{SS}<5.5V			0.3xV_{DD}	V
V_{IH}	Input high level voltage [2]		0.7xV_{DD}			
V_{HYS}	Schmitt trigger voltage hysteresis [3]			400		mV
V_{OL}	Output low level voltage for Standard I/O port pins	I=-5mA			1.3	V
		I=-2mA			0.5	
	Output low level voltage for high sink I/O port pins	I=-20mA			1.3	
		I=-8mA			0.5	
V_{OH}	Output high level voltage	I=-5mA	V_{DD}-2.0			
		I=-2mA	V_{DD}-0.8			
R_{PU}	Pull-up equivalent resistor	V_{IN} > V_{IH} V_{IN} < V_{IL}	20 60	35 100	50 140	kΩ
I_L	Input leakage current	V_{SS}<V_{PIN}<V_{DD}			1	µA
I_{SV}	Static current consumption [2]	Floating input mode			200	
I_{PINJ}	Single pin injected current	Positive [5]: V_{EXT}>V_{DD}			5	mA
		Negative [6]: V_{EXT}<V_{SS}			-5	
I_{INJ}	Total injected current [7] (sum of all I/O and control pins)	Positive: V_{EXT}>V_{DD}			tbd	
		Negative: V_{EXT}<V_{SS}			tbd	
t_{OHL}	Output high to low level fall time	C_l=50pF	14.8 [4]	25	45.6 [4]	ns
t_{OLH}	Output low to high rise time		14.4 [4]	25	45.9 [4]	
t_{ITEXT}	External interrupt pulse time [8]		1			t_{CPU}

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) Data based on design simulation and/or technology characteristics, not tested in production.

3) Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.

4) Data based on characterization results, not tested in production.

5) Positive injection (I_{INJ+})
The I_{INJ+} is performed through protection diodes insulated from the substrate of the die.
The true open-drain pins do not accept positive injection. In this case the maximum voltage rating must be respected.

6) ADC accuracy reduced by negative injection (I_{INJ-})
The I_{INJ-} is performed through protection diodes NOT INSULATED from the substrate of the die. The drawback is a small leakage (a few µA) induced inside the die when a negative injection is performed. This leakage is tolerated by the digital structure, but it acts on the analog line depending on the impedance versus a leakage current of a few µA (if the MCU has an AD converter). The effect depends on the pin which is submitted to the injection. Of course, external digital signals applied to the component must have a maximum impedance close to 50KΩ.
Location of the negative current injection:
- Pins with analog input capability are the most sensitive. I_{INJ-} maximum is 0.8 mA (assuming that the impedance of the analog voltage is lower than 25KΩ)
- Pure digital pins can tolerate 1.6mA. In addition, the best choice is to inject the current as far as possible from the analog input pins.

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

8) 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.

8.6 SUPPLY, RESET AND CLOCK CHARACTERISTICS

8.6.1 Supply Manager

Recommended operating conditions
with T_A=-40 to +85°C and voltage are referred to V_{SS} unless otherwise specified.

LOW VOLTAGE DETECTOR (LVD)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{LVDr}	Reset release threshold (V_{DD} rise)	High Threshold Med. Threshold (f_{OSC}≤16MHz) Low Threshold (f_{OSC}≤8MHz)		4.30 3.90 3.35	4.50 4.05 3.45	V
V_{LVDf}	Reset generation threshold (V_{DD} fall)	High Threshold Med. Threshold (f_{OSC}≤16MHz) Low Threshold (f_{OSC}≤8MHz)	3.85 3.50 3.00	4.05 3.65 3.10	4.25 3.80 3.20	
$V_{LVDhyst}$	V_{LVD} Hysteresis [2]	V_{LVDr} - V_{LVDf}		250*		mV
I_{DD}	LVD Supply Current	HALT mode		100	150 [3]	µA

8.6.2 Reset Sequence Manager

Recommended operating conditions
with T_A=-40...+85°C and 4.5V<V_{DD}-V_{SS}<5.5V unless otherwise specified.

RESET SEQUENCE MANAGER (RSM)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [4]	**Max**	**Unit**
R_{ON}	Reset weak pull-up resistance	V_{IN} > V_{IH} V_{IN} < V_{IL}	5 40	10 80	20 160	kΩ
$t_{DELAYmin}$	Reset delay for external and watchdog reset sources			6 30		$1/f_{SFOSC}$ µs
t_{PULSE}	External $\overline{RESET}$ pin Pulse time		20			µs

8.6.3 Multi-Oscillator, Clock Security System

Recommended operating conditions
with T_A=-40 to +85°C and voltage are referred to V_{SS} unless otherwise specified.

EXTERNAL CLOCK SOURCE						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{OSC1h}	OSC1 input pin high level voltage	Square wave signal with ~50% Duty Cycle	0.7xV_{DD}		V_{DD}	V
V_{OSC1l}	OSC1 input pin low level voltage		V_{SS}		0.3xV_{DD}	

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 $V_{LVDhyst}$ hysteresis is constant.

3) Data based on characterization results, not tested in production.

4) 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.

SUPPLY, RESET AND CLOCK CHARACTERISTICS (Cont'd)

CRYSTAL AND CERAMIC RESONATOR OSCILLATORS

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{OSC}	Oscillator Frequency [2]	Low speed resonator	1		2	MHz
		Medium-low speed resonator	>2		4	
		Medium-high speed resonator	>4		8	
		High speed resonator	>8		16	
C_{Li}	Load Capacitance Low speed	R_{Smax}=200Ω [3]	38 [4]	47	56 [4]	pF
	Load Capacitance Medium-low speed	R_{Smax}=200Ω [3]	32 [4]	39	46 [4]	
	Load Capacitance Medium-high speed	R_{Smax}=200Ω [3]	18 [4]	22	26 [4]	
	Load Capacitance High speed	R_{Smax}=100Ω [3]	15 [4]	18	21 [4]	
I_{DD}	Supply Current Low speed			150	700 [5]	µA
	Supply Current Medium-low speed			200	700 [5]	
	Supply Current Medium-high speed			400	750 [5]	
	Supply Current High speed			700	1100 [5]	
t_{START}	Oscillator start-up time	Depends on resonator quality. A typical value is 10ms				

EXTERNAL RC OSCILLATOR

Symbol	Parameter	Conditions	Min [4]	Typ [1]	Max	Unit
f_{OSC}	External RC Oscillator Frequency	V_{DD}=5V	1		14 [4]	MHz
R_{EX}	Oscillator External Resistance		10	33	47 [4]	kΩ
C_{EX}	Oscillator External Capacitance		0 [6]	47	470 [4]	pF
I_{DD}	Supply Current			525	750 [5]	µA

INTERNAL RC OSCILLATOR

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{OSC}	Internal RC Oscillator Frequency	V_{DD}=5.5V	3.50	4.25	5.00	MHz
		V_{DD}=3.0V	3.35	4.10	4.85	
I_{DD}	Supply Current			500	750 [5]	µA

CLOCK SECURITY SYSTEM (CSS)

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{SFOSC}	Safe Oscillator Frequency	V_{DD}=5.5V	250	340	430	kHz
		V_{DD}=3.0V	190	260	330	
f_{CFL}	Clock Filter Frequency limitation			30		MHz
I_{DD}	Supply Current			150	350 [5]	µA

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) These data are based on typical R_{Smax}. The oscillator selection can be optimized in terms of supply current with high quality resonator.

3) R_{Smax} is the equivalent serial resistance of the crystal or ceramic resonator.

4) Data based on design simulation and/or technology characteristics, not tested in production.

5) Data based on characterization results, not tested in production.

6) In this condition, the capacitor to be considered is the global parasitic capacitor. In this case, the RC oscillator frequency tuning has to be done by trying out several resistor values.

8.7 MEMORY AND PERIPHERAL CHARACTERISTICS

Recommended operating conditions
with T_A=-40 to +85°C and 3V<V_{DD}-V_{SS}<5.5V unless otherwise specified.

FLASH Program Memory						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
$t_{ISPPROG}$	Typical programming time	8 or 16kBytes FLASH		2.1	6.4	sec
t_{RET}	Data retention	T_A=55°C	20			years
N_{RW}	Write erase cycles		1000			cycles

Data-EEPROM						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
t_{PROG}	Programming time				25	ms
t_{RET}	Data retention		10			Years
N_{RW}	Write erase cycles		100 000			Cycles

WATCHDOG						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
t_{DOG}	Watchdog time-out	f_{CPU} = 8MHz	12,288 1.54		786,432 98.3	t_{CPU} ms

Note:

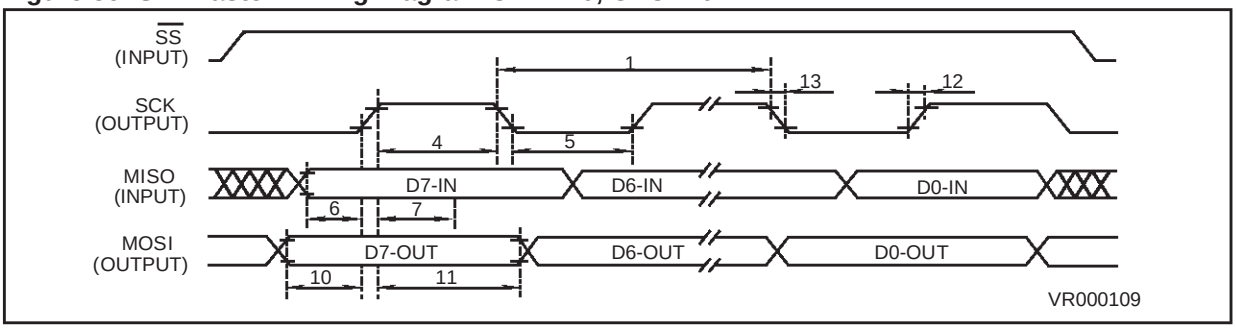
1) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

SPI Serial Peripheral Interface

Ref.	Symbol	Parameter	Condition	Value [1]		Unit
				Min	Max	
	f_{SPI}	SPI frequency	Master Slave	1/128 dc	1/4 1/2	f_{CPU}
1	t_{SPI}	SPI clock period	Master Slave	4 2		t_{CPU}
2	t_{Lead}	Enable lead time	Slave	120		ns
3	t_{Lag}	Enable lag time	Slave	120		ns
4	t_{SPI_H}	Clock (SCK) high time	Master Slave	100 90		ns
5	t_{SPI_L}	Clock (SCK) low time	Master Slave	100 90		ns
6	t_{SU}	Data set-up time	Master Slave	100 100		ns
7	t_H	Data hold time (inputs)	Master Slave	100 100		ns
8	t_A	Access time (time to data active from high impedance state)	Slave	0	120	ns
9	t_{Dis}	Disable time (hold time to high impedance state)			240	ns
10	t_V	Data valid	Master (before capture edge) Slave (after enable edge)	0.25	120	t_{CPU} ns
11	t_{Hold}	Data hold time (outputs)	Master (before capture edge) Slave (after enable edge)	0.25 0		t_{CPU} ns
12	t_{Rise}	Rise time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs
13	t_{Fall}	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs

Figure 59. SPI Master Timing Diagram CPHA=0, CPOL=0 [2]



Notes:

1) Data based on characterization results, not tested in production.

2) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

Figure 60. SPI Master Timing Diagram CPHA=0, CPOL=1 1)

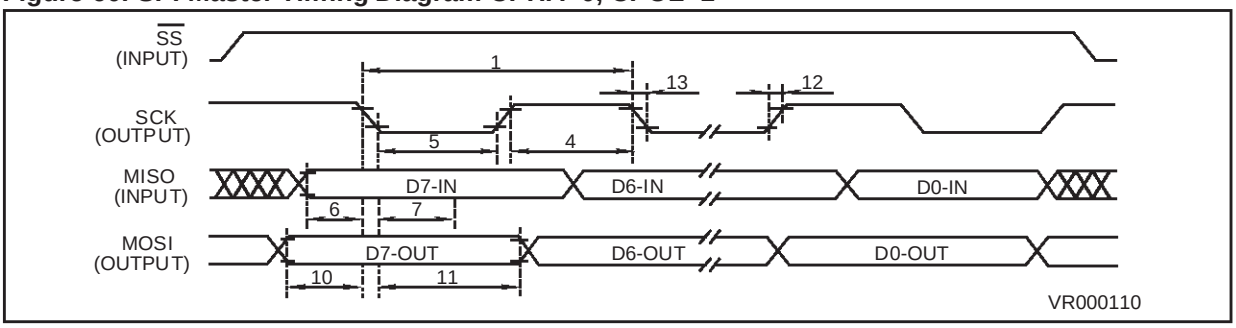


Figure 61. SPI Master Timing Diagram CPHA=1, CPOL=0 1)

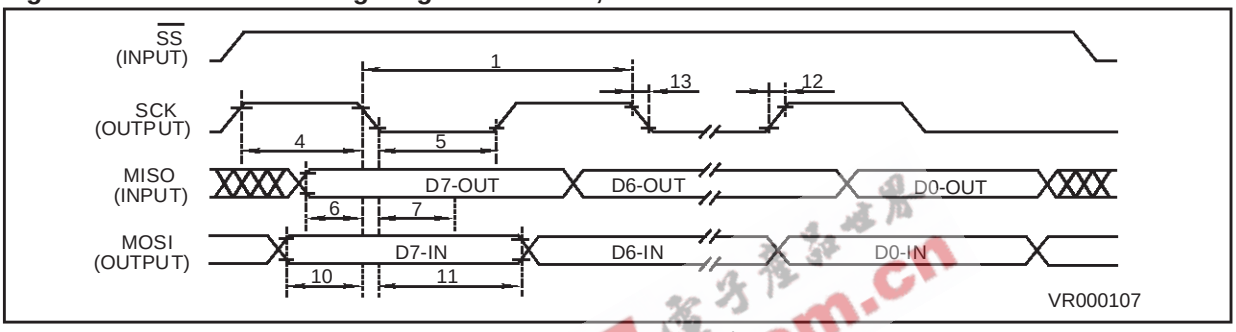
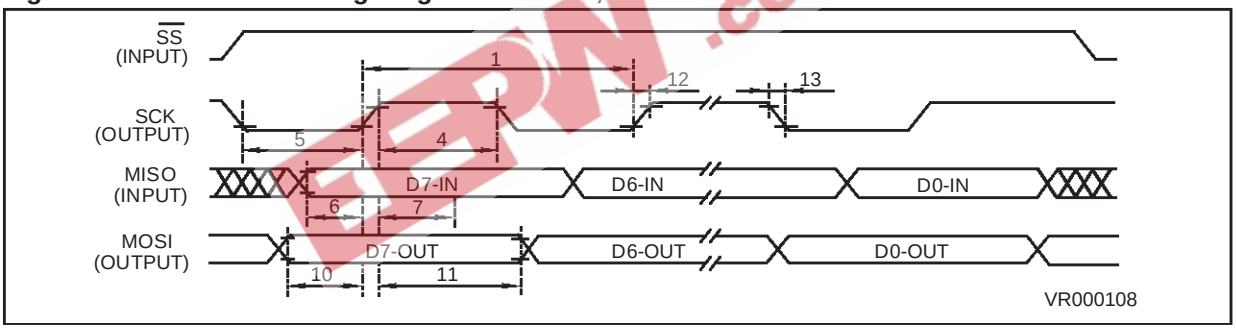


Figure 62. SPI Master Timing Diagram CPHA=1, CPOL=1 1)



Note:
1) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI Timing Diagram

Figure 63. SPI Slave Timing Diagram CPHA=0, CPOL=0 1)

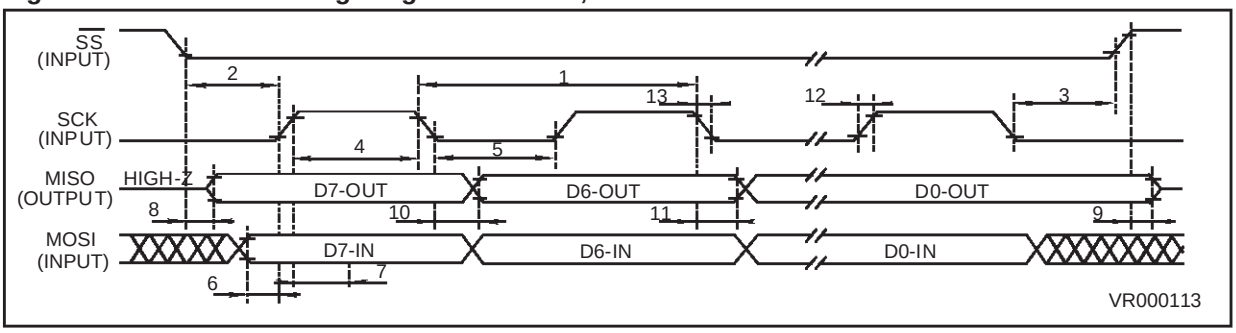


Figure 64. SPI Slave Timing Diagram CPHA=0, CPOL=1 1)

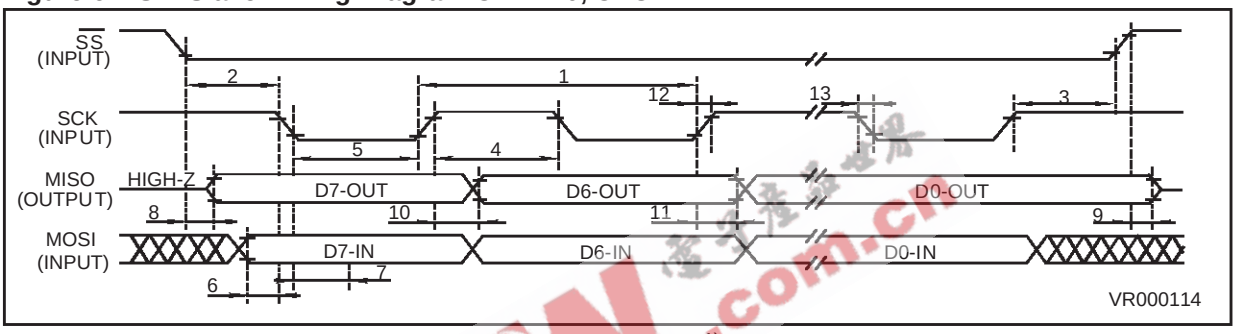


Figure 65. SPI Slave Timing Diagram CPHA=1, CPOL=0 1)

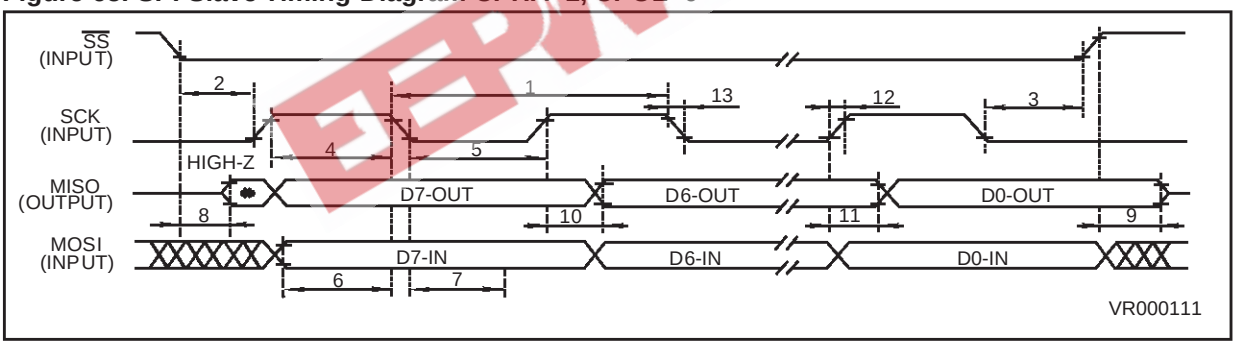
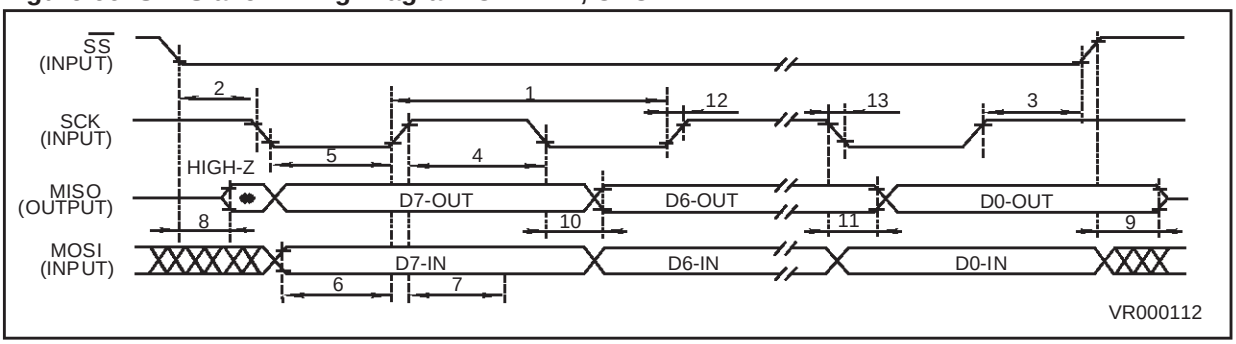


Figure 66. SPI Slave Timing Diagram CPHA=1, CPOL=1 1)



Note:
1) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

SCI Serial Communication Interface					
Symbol	**Parameter**	**Conditions**		**Typ** [1]	**Unit**
f_{Tx} or f_{Rx}	Communication frequency (precision vs. standard ~0.16%)	f_{CPU}=8MHz	Standard Mode TR (resp.RR)=64, PR=13 TR (resp.RR)=16, PR=13 TR (resp.RR)= 8, PR=13 TR (resp.RR)= 4, PR=13 TR (resp.RR)= 2, PR=13 TR (resp.RR)= 8, PR= 3 TR (resp.RR)= 1, PR=13	 ~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230,77	Hz
			Extended Mode ETPR (resp.ERPR) = 13	~38461.54	
See "STANDARD I/O PORT PINS" description for more details.					

Note:

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.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

ADC Analog to Digital Converter (8-bit)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
f_{ADC}	Analog control frequency	$V_{DD}=V_{DDA}=5V$			4 [2]	MHz
|TUE|	Total unadjusted error [4]	$T_A=25°C, V_{DD}=V_{DDA}=5V$, [3] $f_{CPU}=8MHz, f_{ADC}=4MHz$			1	LSB
OE	Offset error [4]		-0.5		0.5	
GE	Gain Error [4]		-0.5		0.5	
|DLE|	Differential linearity error [4]				0.5	
|ILE|	Integral linearity error [4]				0.5	
V_{AIN}	Conversion range voltage		V_{SSA}		V_{DDA}	V
I_{ADC}	A/D conversion supply current	$f_{CPU}=8MHz, f_{ADC}=4MHz$ $V_{DD}=V_{DDA}=5V$		1		mA
t_{STAB}	Stabilization time after ADC enable				1	µs
t_{LOAD}	Sample capacitor loading time			1 4		µs $1/f_{ADC}$
t_{CONV}	Hold conversion time			2 8		µs $1/f_{ADC}$
R_{AIN}	External input resistor				15 [2]	kΩ
R_{ADC}	Internal input resistor			1.5		kΩ
C_{SAMPLE}	Sample capacitor			6		pF

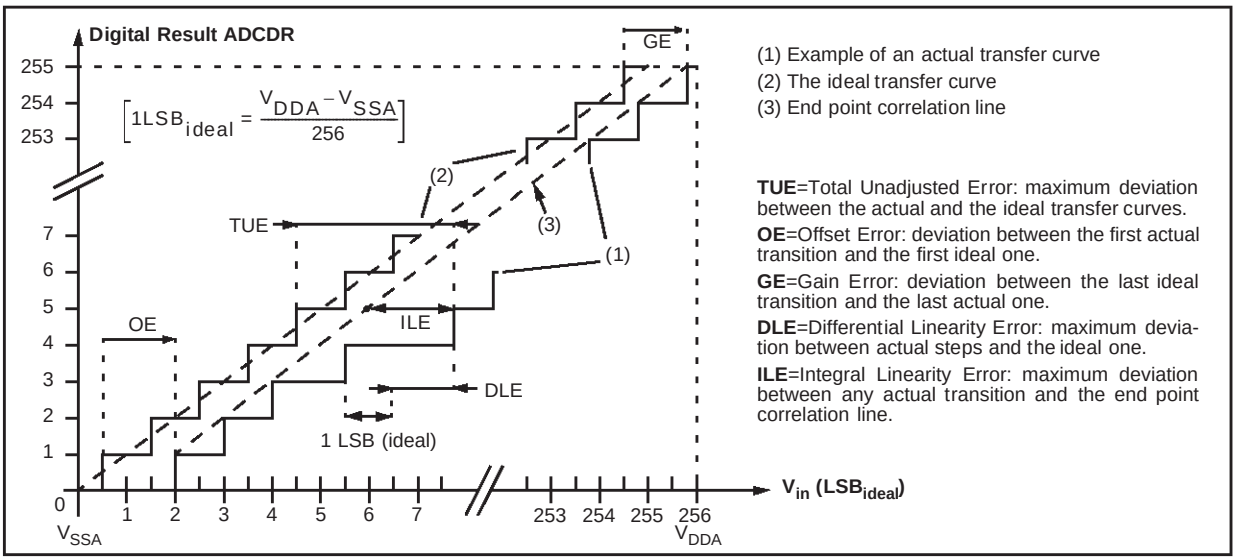
Notes:

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) Data based on characterization results, not tested in production.

3) Tested in production at T_A=25°C, characterized over the whole temperature range.

4) 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.

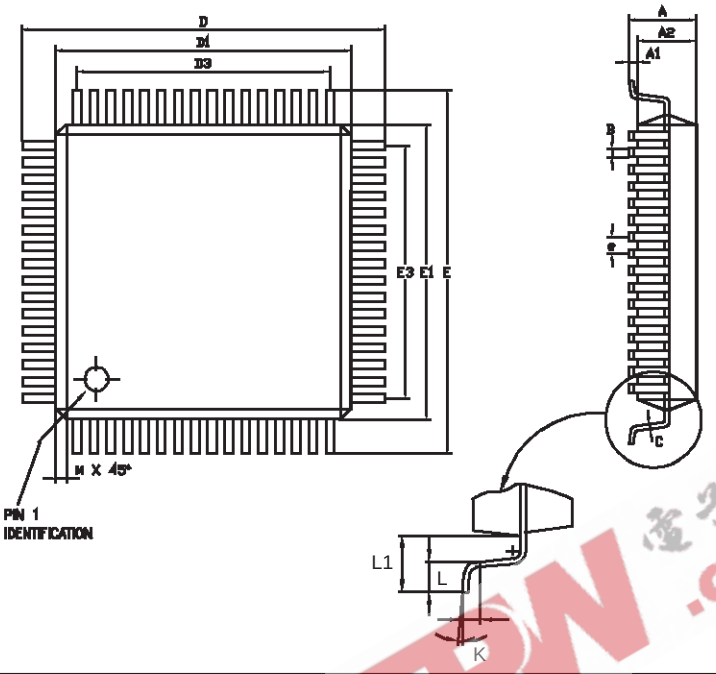


9 GENERAL INFORMATION

9.1 PACKAGES

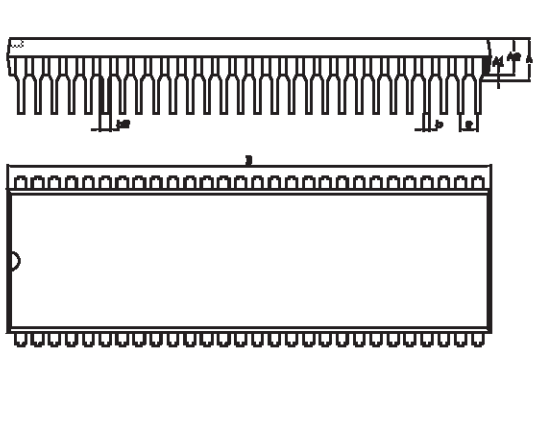
9.1.1 Package Mechanical Data

Figure 67. 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 68. 56-Pin Shrink Plastic Dual In-Line Package, 600-mil Width

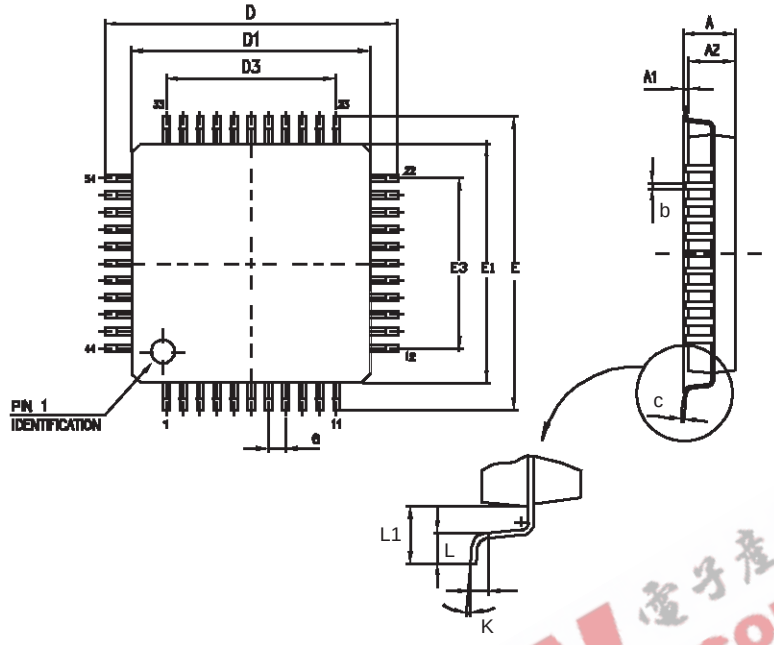


PDIP56S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			6.35			0.250
A1	0.38			0.015		
A2	3.18		4.95	0.125		0.195
b		0.41			0.016	
b2		0.89			0.035	
C	0.20		0.38	0.008		0.015
D	50.29		53.21	1.980		2.095
E		15.01			0.591	
E1	12.32		14.73	0.485		0.580
e		1.78			0.070	
eA		15.24			0.600	
eB			17.78			0.700
L	2.92		5.08	0.115		0.200
	Number of Pins					
N	56					

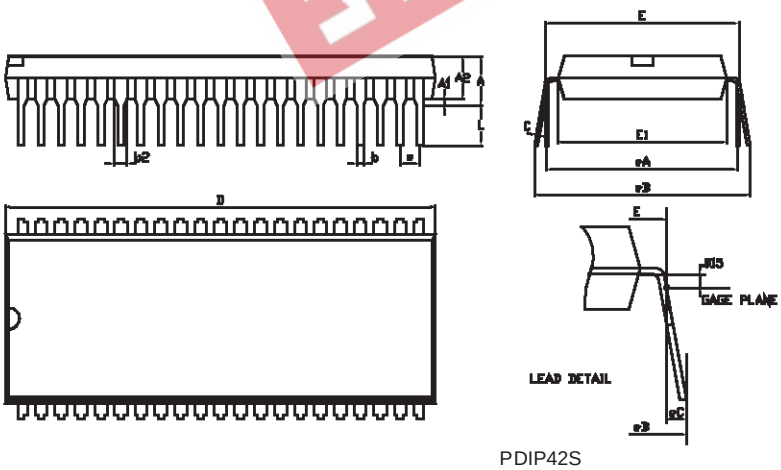
PACKAGES (Cont'd)

Figure 69. 44-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		12.00			0.472	
D1		10.00			0.394	
D3		8.00			0.315	
E		12.00			0.472	
E1		10.00			0.394	
E3		8.00			0.315	
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	44					

Figure 70. 42-Pin Shrink Plastic Dual In-Line Package, 600-mil Width



PDIP42S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
A2	3.05	3.81	4.57	0.120	0.150	0.180
b		0.46	0.56		0.018	0.022
b2		1.02	1.14		0.040	0.045
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.58	36.83	37.08	1.440	1.450	1.460
E	15.24		16.00	0.600		0.630
E1	12.70	13.72	14.48	0.500	0.540	0.570
e		1.78			0.070	
eA		15.24			0.600	
eB			18.54			0.730
eC	0.00		1.52	0.000		0.060
L	2.54	3.30	3.56	0.100	0.130	0.140
	Number of Pins					
N	42					

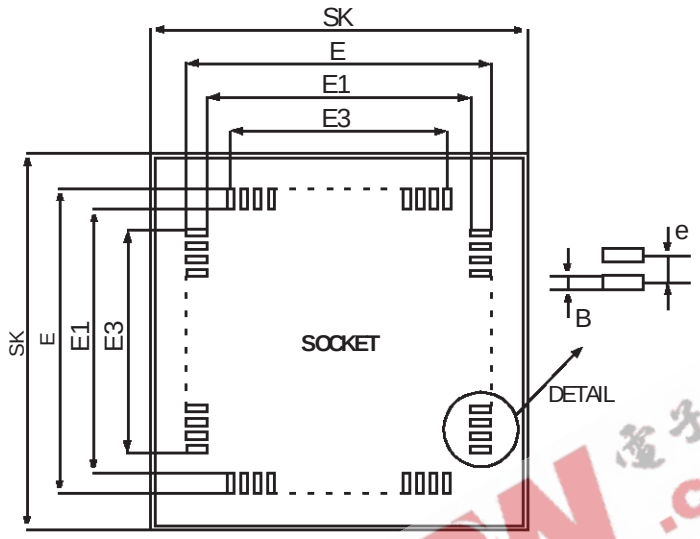
PACKAGES (Cont'd)

9.1.2 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 71. 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.
  * Not compatible with TQFP64 package.

Figure 71. 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 22. 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

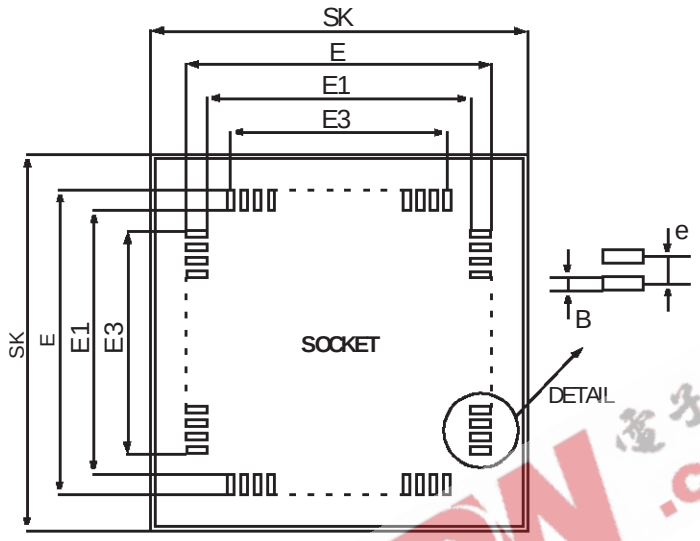
PACKAGES (Cont'd)

9.1.3 User-supplied TQFP44 Adaptor / Socket

To solder the TQFP44 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 72. This footprint allows both configurations:

- Direct TQFP44 soldering
- YAMAICHI IC149-044-*52-S5 socket soldering to plug either the emulator probe or an adaptor board with an TQFP44 clamshell socket.

Figure 72. TQFP44 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	13.40			0.527		
E1			10.00			0.394
E3	7.95	8.00	8.05	0.313	0.315	0.317
e	0.75	0.80	0.85	0.029	0.031	0.033
SK*		24.2			0.953	
Number of Pins						
N	44 (4x11)					

* SK: Plastic socket overall dimensions.

Table 23. Suggested List of TQFP44 Socket Types

Package / Probe	Adaptor / Socket Reference		Socket type
TQFP44	ENPLAS	OTQ-44-0.8-04	Open Top
	YAMAICHI	IC51-0444-467-KS-11787	Clamshell
TQFP44 EMU PROBE	YAMAICHI	IC149-044-*52-S5	SMC

9.2 DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (FLASH) as well as in factory coded versions (ROM).

FLASH 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 FLASH devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

9.2.1 Option Bytes

The two Option Bytes allow the hardware configuration of the microcontroller to be selected.
The Option Bytes have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 4programming tool). The default contents of the FLASH 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.

USER OPTION BYTE 1

7							0
1	1	1	1	1	1	56/42	FMP

Bit 7:2 = Reserved, must always be 1.

Bit 1 = **56/42** *Package configuration.*
This option bit allows to configure the device according to the package.
0: 42 and 44 pin.
1: 56 and 64 pin.

Bit 0 = **FMP** *Full memory protection.*
This option bit enables or disables external access to the internal program memory (read-out protection). Clearing this bit causes the erasing (to 00h) of the whole memory (including the option byte).
0: Program memory not read-out protected
1: Program memory read-out protected

USER OPTION BYTE 2

7							0
CFC	OSC2	OSC1	OSC0	LVD1	LVD0	WDG HALT	WDG SW

Bit 7 = **CFC** *Clock filter control on/off*
This option bit enables or disables the clock filter (CF) features.
0: Clock filter enabled
1: Clock filter disabled

Bit 6:4 = **OSC[2:0]** *Oscillator selection*
These three option bits can be used to select the main oscillator as shown in Table 24.

Table 24. Main Oscillator Configuration

Selected Oscillator	OSC2	OSC1	OSC0
External Clock (Stand-by)	1	1	1
Internal RC	1	1	0
External RC	1	0	1
	1	0	0
Low Speed Resonator	0	1	1
Medium-low Speed Resonator	0	1	0
Medium-high Speed Resonator	0	0	1
High Speed Resonator	0	0	0

Bit 3:2 = **LVD[1:0]** *Low voltage detection selection*
These option bits enable the LVD block with a selected threshold as shown in Table 25.

Table 25. LVD Threshold Configuration

Configuration	LVD1	LVD0
LVD Off	1	1
Highest Voltage Threshold (V_{DD}~5V)	1	0
Medium Voltage Threshold (f_{OSC}≤16MHz)	0	1
Lowest Voltage Threshold (f_{OSC}≤8MHz)	0	0

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)

DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

9.2.2 Transfer Of Customer Code

Customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file in .S19 format 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 73. ROM Factory Coded Device Types

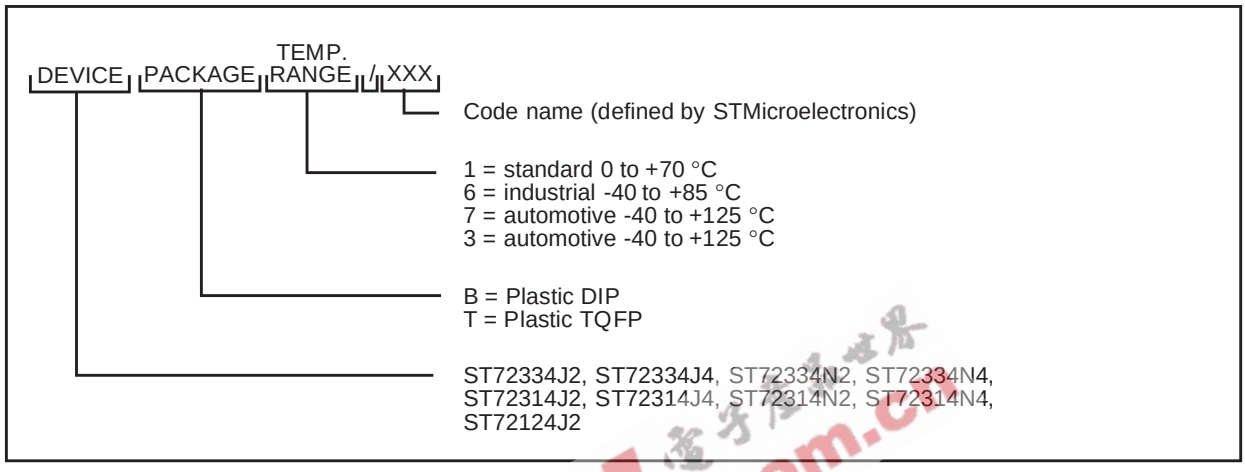
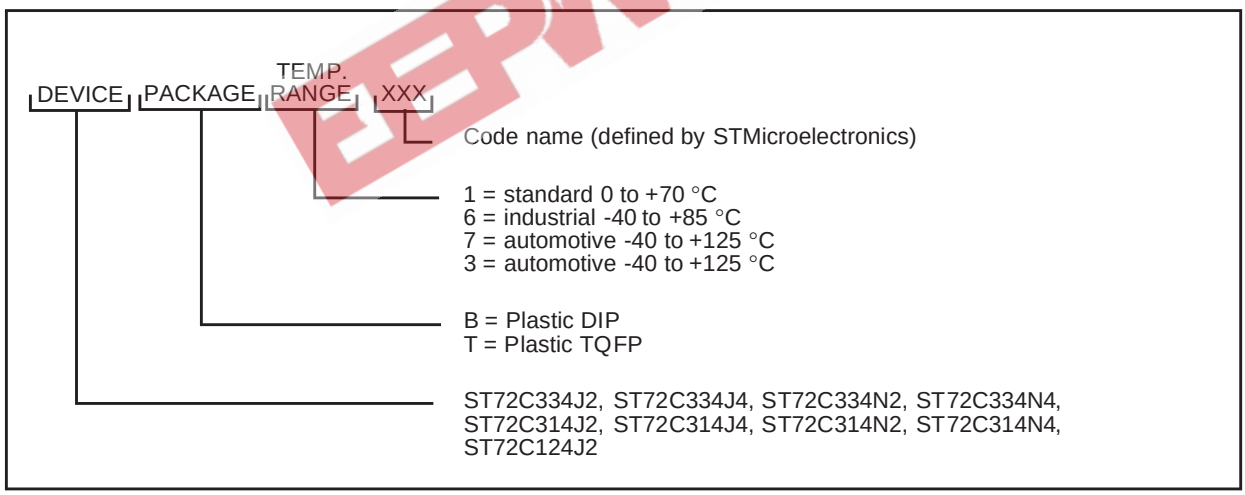


Figure 74. FLASH User Programmable Device Types



MICROCONTROLLER OPTION LIST

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

STMicroelectronics references

Device:
[ ] ST72334J2 [ ] ST72314J2 [ ] ST72124J2
[ ] ST72334J4 [ ] ST72314J4
[ ] ST72334N2 [ ] ST72314N2
[ ] ST72334N4 [ ] ST72314N4

Package:
[ ] TQFP64 [ ] SDIP56
[ ] TQFP44 [ ] SDIP42

Temperature Range:
[ ] 0°C to + 70°C [ ] - 40°C to + 85°C [ ] - 40°C to + 125°C

Clock Source Selection:
[ ] Resonator:
[ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] RC Network:
[ ] Internal
[ ] External
[ ] External Clock

Clock Security System:
[ ] Disabled
[ ] Enabled

Watchdog Selection:
[ ] Software Activation
[ ] Hardware Activation

Halt when Watchdog on:
[ ] Reset
[ ] No reset

Readout Protection:
[ ] Disabled
[ ] Enabled

LVD Reset
[ ] Disabled [ ] Enabled:
[ ] Highest threshold (4.30V/4.05V)
[ ] Medium threshold (3.90V/3.65V)
[ ] Lowest threshold (3.35V/3.10V)

Comments : . . . . . .
Supply Operating Range in the application: . . . . . .
Notes . . . . . .
Signature . . . . . .
Date . . . . . .

10 SUMMARY OF CHANGES

Description of the changes between the current release of the specification and the previous one.

Revision	Main changes	Date
1.0	New chapter to compare ST72334 versus ST72331 (section 2.1 on page 6) Correction of the address of the CRSR register to 2Bh instead of 25h (Table 4 page 33) Correction of port A pin name column in Table 9 page 44 (PA2:0 instead of PA3:0) Correction of MISCR2 register description (section 6.2.3 on page 48) Correction of the FLASH and data EEPROM programming time (section 8.7 on page 111) Correction of the TQFP44 socket proposal (Table 23 page 120) More information on the FMP option bit (section 9.2.1 on page 121) Added .S19 format in transfer of Code (section 9.2.2 on page 122) Correction of the microcontroller option list (section 9.2.2 on page 122) History page added (section 10 on page 124)	Sept-99

Notes: