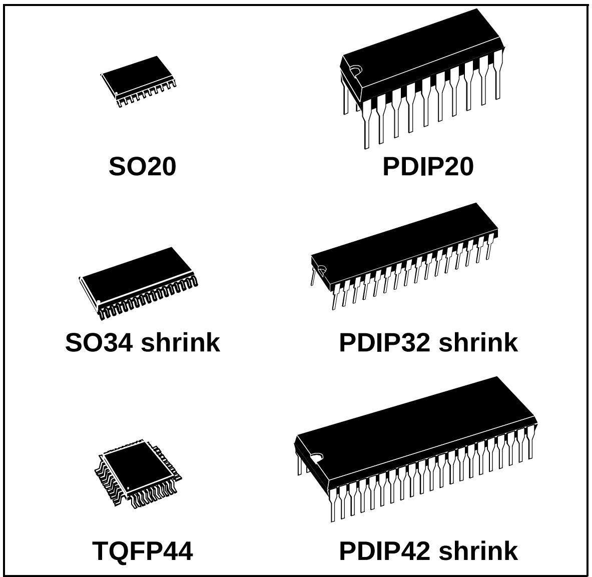
ST7262

LOW SPEED USB 8-BIT MCU WITH 3 ENDPOINTS, FLASH OR ROM MEMORY, LVD, WDG, 10-BIT ADC, 2 TIMERS, SCI, SPI

PRODUCT PREVIEW

- **Memories**
  - 8K or 16K Program memory (ROM, FASTROM or Dual voltage FLASH) with read-write protection
  - In-Application and In-Circuit Programming for FLASH versions
  - 384 to 768 bytes RAM memory (128-byte stack)
- **Clock, Reset and Supply Management**
  - Enhanced Reset System (Power On Reset)
  - Low Voltage Detector (LVD)
  - Clock-out capability
  - 6 or 12 MHz Oscillator (8, 4, 2, 1 MHz internal frequencies)
  - 3 Power saving modes
- **USB (Universal Serial Bus) Interface**
  - DMA for low speed applications compliant with USB 1.5 Mbs specification (v 1.1) and USB HID specification (v 1.0):
  - Integrated 3.3V voltage regulator and transceivers
  - Suspend and Resume operations
  - 3 Endpoints
- **Up to 31 I/O Ports**
  - Up to 31 multifunctional bidirectional I/O lines
  - Up to 12 External interrupts (3 vectors)
  - 13 alternate function lines
  - 8 high sink outputs (8 mA@0.4 V/20 mA@1.3 V)
  - 2 true open drain pins (N buffer 8 mA@0.4 V)
- **3 Timers**
  - Configurable watchdog timer (8 to 500 ms timeout)
  - 8-bit Auto Reload Timer (ART) with 2 Input Captures, 2 PWM outputs and External Clock Input
  - 8-bit Time Base Unit (TBU) for generating periodic interrupts cascadable with ART



- **Analog Peripheral**
  - 10-bit A/D Converter with up to 8 input pins.
- **2 Communications Interfaces**
  - Asynchronous Serial Communication interface
  - Synchronous Serial Peripheral Interface
- **Instruction Set**
  - 8-bit data manipulation
  - 63 basic instructions
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instruction
  - True bit manipulation
- **Development Tools**
  - Full hardware/software development package

Device Summary

Features	ST72623F2	ST72622K2	ST72621K4	ST72622L2	ST72621L4	ST72621J2	ST72621J4
Program memory - bytes	8K	8K	16K	8K	16K	8K	16K
RAM (stack) - bytes	384 (128)	384 (128)	768 (128)	384 (128)	768 (128)	384 (128)	768 (128)
Peripherals	USB, Watchdog, Low Voltage Detector, 8-bit Auto-Reload timer, Timebase unit, A/D Converter						
Serial I/O	-	SPI	SPI + SCI	SPI	SPI + SCI		
I/Os	11	21		23		31	
Operating Supply	3.0V to 5.5V						
Operating Temperature	0°C to +70°C						
Packages	PDIP20/SO20	PDIP32		SO34		PDIP42/TQFP44	

Rev. 1.0

This is preliminary information on a new product now in development. Details are subject to change without notice.

Table of Contents

1 INTRODUCTION 6

2 PIN DESCRIPTION 7

2.1 PCB LAYOUT RECOMMENDATION 12

3 REGISTER & MEMORY MAP 13

4 FLASH PROGRAM MEMORY 16

4.1 INTRODUCTION 16

4.2 MAIN FEATURES 16

4.3 STRUCTURE 16

4.4 PROGRAM MEMORY READ-OUT PROTECTION 16

4.5 ICP (IN-CIRCUIT PROGRAMMING) 17

4.6 IAP (IN-APPLICATION PROGRAMMING) 17

4.6.1 Register Description 18

5 CENTRAL PROCESSING UNIT 19

5.1 INTRODUCTION 19

5.2 MAIN FEATURES 19

5.3 CPU REGISTERS 19

6 CLOCKS AND RESET 22

6.1 CLOCK SYSTEM 22

6.1.1 General Description 22

6.1.2 External Clock input 22

6.1.3 Clock Output Pin (MCO) 22

6.2 RESET 23

6.2.1 Low Voltage Reset 23

6.2.2 Watchdog Reset 23

6.2.3 External Reset 23

7 INTERRUPTS 25

7.1 NON MASKABLE SOFTWARE INTERRUPT 25

7.2 EXTERNAL INTERRUPTS 25

7.3 PERIPHERAL INTERRUPTS 25

7.4 INTERRUPT REGISTER 27

8 POWER SAVING MODES 28

8.1 INTRODUCTION 28

8.1.1 Slow Mode 28

8.2 WAIT MODE 28

8.3 HALT MODE 29

9 I/O PORTS 30

9.1 INTRODUCTION 30

9.2 FUNCTIONAL DESCRIPTION 30

9.2.1 Input Modes 30

9.2.2 Output Mode 30

9.2.3 Alternate Functions 30

9.2.4 I/O Port Implementation 31

9.2.5 Port A 32

9.2.6 Port B 33

9.2.7 Port C 34
9.2.8 Port D 35
9.2.9 Register Description 36
9.3 MISCELLANEOUS REGISTER 38
10 ON-CHIP PERIPHERALS 39
10.1 WATCHDOG TIMER (WDG) 39
10.1.1 Introduction 39
10.1.2 Main Features 39
10.1.3 Functional Description 39
10.1.4 Software Watchdog Option 40
10.1.5 Hardware Watchdog Option 40
10.1.6 Low Power Modes 40
10.1.7 Interrupts 40
10.1.8 Register Description 40
10.2 PWM AUTO-RELOAD TIMER (ART) 41
10.2.1 Introduction 41
10.2.2 Functional Description 42
10.2.3 Register Description 46
10.3 TIMEBASE UNIT (TBU) 50
10.3.1 Introduction 50
10.3.2 Main Features 50
10.3.3 Functional Description 50
10.3.4 Programming Example 50
10.3.5 Low Power Modes 51
10.3.6 Interrupts 51
10.3.7 Register Description 51
10.4 SERIAL PERIPHERAL INTERFACE (SPI) 53
10.4.1 Introduction 53
10.4.2 Main Features 53
10.4.3 General Description 53
10.4.4 Functional Description 55
10.4.5 Register Description 62
10.4.6 Low Power Modes 64
10.4.7 Interrupts 64
10.5 SERIAL COMMUNICATIONS INTERFACE (SCI) 66
10.5.1 Introduction 66
10.5.2 Main Features 66
10.5.3 General Description 66
10.5.4 Functional Description 68
10.5.5 Low Power Modes 73
10.5.6 Interrupts 73
10.5.7 Register Description 74
10.6 USB INTERFACE (USB) 80
10.6.1 Introduction 80
10.6.2 Main Features 80
10.6.3 Functional Description 80
10.6.4 Register Description 81
10.6.5 Programming Considerations 86

Table of Contents

10.7 10-BIT A/D CONVERTER (ADC) 88
- 10.7.1 Introduction 88
- 10.7.2 Main Features 88
- 10.7.3 Functional Description 88
- 10.7.4 Low Power Modes 90
- 10.7.5 Interrupts 90
- 10.7.6 Register Description 91

11 INSTRUCTION SET 92

11.1 ST7 ADDRESSING MODES 92
- 11.1.1 Inherent 93
- 11.1.2 Immediate 93
- 11.1.3 Direct 93
- 11.1.4 Indexed (No Offset, Short, Long) 93
- 11.1.5 Indirect (Short, Long) 93
- 11.1.6 Indirect Indexed (Short, Long) 94
- 11.1.7 Relative Mode (Direct, Indirect) 94

11.2 INSTRUCTION GROUPS 95

12 ELECTRICAL CHARACTERISTICS 98

12.1 PARAMETER CONDITIONS 98
- 12.1.1 Minimum and Maximum Values 98
- 12.1.2 Typical Values 98
- 12.1.3 Typical Curves 98
- 12.1.4 Loading Capacitor 98
- 12.1.5 Pin Input Voltage 98

12.2 ABSOLUTE MAXIMUM RATINGS 99
- 12.2.1 Voltage Characteristics 99
- 12.2.2 Current Characteristics 99
- 12.2.3 Thermal Characteristics 100

12.3 OPERATING CONDITIONS 100
- 12.3.1 General Operating Conditions 100
- 12.3.2 Operating Conditions with Low Voltage Detector (LVD) 101

12.4 SUPPLY CURRENT CHARACTERISTICS 102

12.5 CLOCK AND TIMING CHARACTERISTICS 103
- 12.5.1 General Timings 103
- 12.5.2 CONTROL TIMING CHARACTERISTICS 103
- 12.5.3 External Clock Source 104

12.6 MEMORY CHARACTERISTICS 105
- 12.6.1 RAM and Hardware Registers 105
- 12.6.2 FLASH Memory 105

12.7 EMC CHARACTERISTICS 106
- 12.7.1 Functional EMS 106
- 12.7.2 Absolute Electrical Sensitivity 107
- 12.7.3 ESD Pin Protection Strategy 109

12.8 I/O PORT PIN CHARACTERISTICS 111
- 12.8.1 General Characteristics 111
- 12.8.2 Output Driving Current 112

12.9 CONTROL PIN CHARACTERISTICS 114

12.9.1 Asynchronous RESET Pin 114
12.10 TIMER PERIPHERAL CHARACTERISTICS 116
12.10.1 8-Bit PWM-ART Auto-Reload Timer 116
12.11 COMMUNICATION INTERFACE CHARACTERISTICS 117
12.11.1 USB - Universal Bus Interface 117
12.11.2 SPI - Serial Peripheral Interface 118
12.11.3 SCI - Serial Communications Interface 120
12.12 10-BIT ADC CHARACTERISTICS 121

13 PACKAGE CHARACTERISTICS 123

13.1 PACKAGE MECHANICAL DATA 123

14 DEVICE CONFIGURATION AND ORDERING INFORMATION 126

14.1 OPTION BYTE 126
14.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE 127
14.3 ST7 APPLICATION NOTES 131
14.4 TO GET MORE INFORMATION 131

15 SUMMARY OF CHANGES 132

1 INTRODUCTION

The ST7262, ST72P62 and ST72F62 devices are members of the ST7 microcontroller family designed for USB applications.

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

The ST7262 devices are ROM versions.

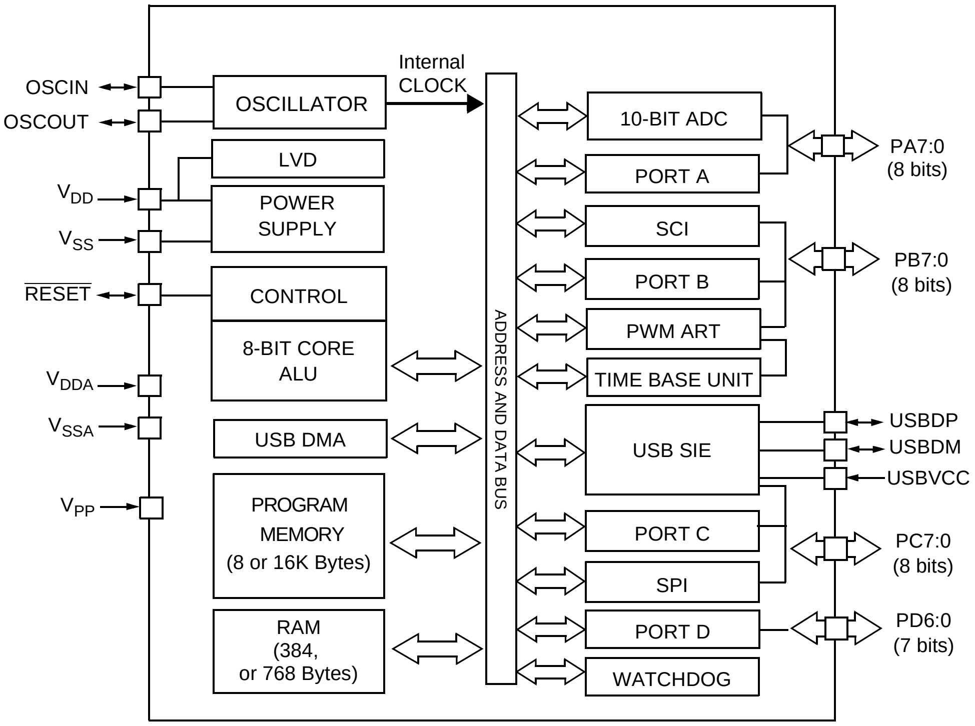
The ST72P62 devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory-programmed and are not reprogrammable.

The ST72F62 versions feature dual-voltage FLASH memory with FLASH Programming capability.

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

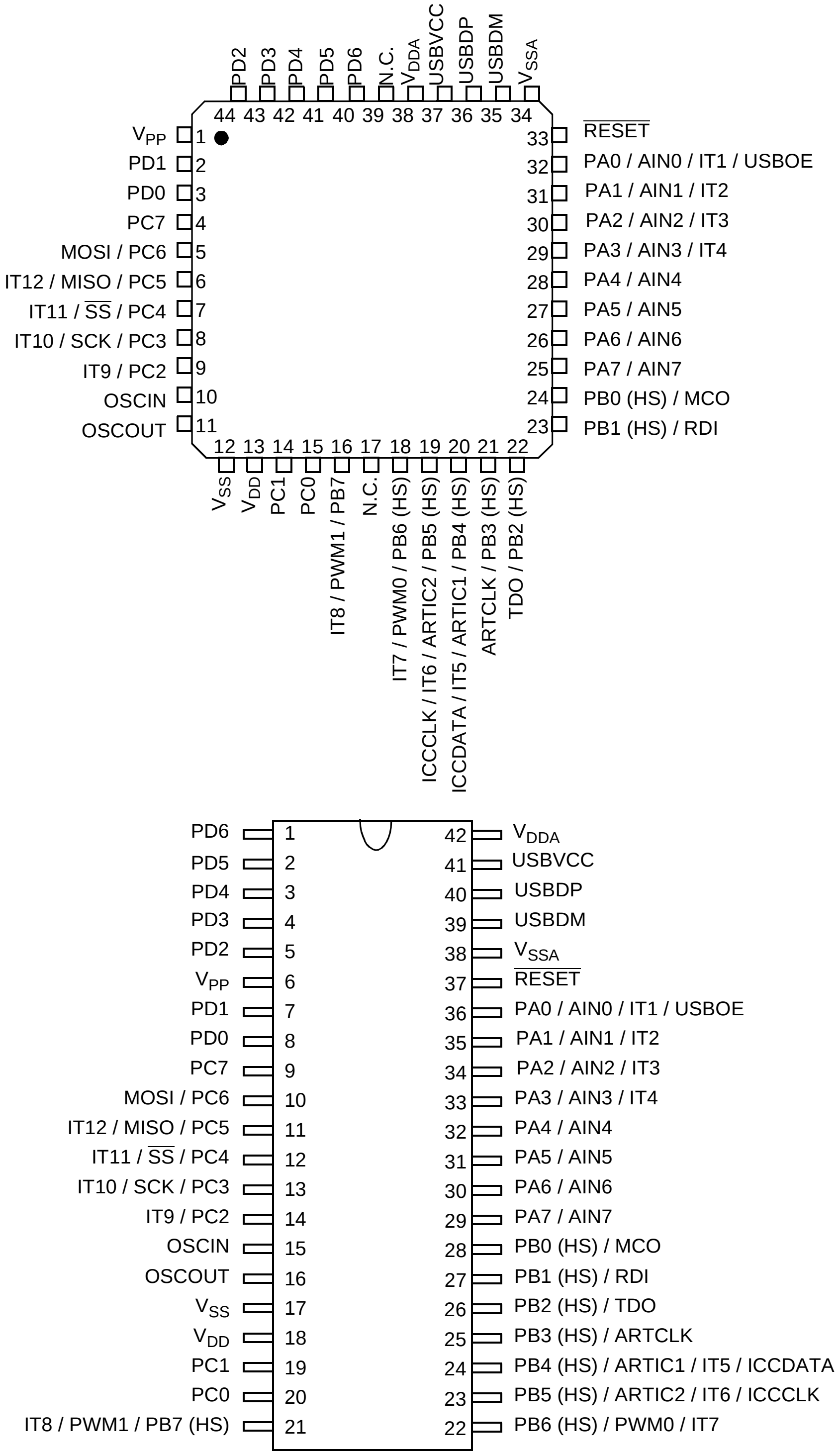
The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned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

Figure 1. General Block Diagram


OSCIN
OSCOUT
VDD
VSS
RESET
VDDA
VSSA
VPP
OSCILLATOR
Internal CLOCK
LVD
POWER SUPPLY
CONTROL
8-BIT CORE ALU
USB DMA
PROGRAM MEMORY (8 or 16K Bytes)
RAM (384, or 768 Bytes)
ADDRESS AND DATA BUS
10-BIT ADC
PORT A
SCI
PORT B
PWM ART
TIME BASE UNIT
USB SIE
PORT C
SPI
PORT D
WATCHDOG
PA7:0 (8 bits)
PB7:0 (8 bits)
USBDP
USBDM
USBVCC
PC7:0 (8 bits)
PD6:0 (7 bits)

2 PIN DESCRIPTION

Figure 2. 44-pin TQFP and 42-Pin SDIP Package* Pinouts


PD2
PD3
PD4
PD5
PD6
N.C.
V_{DDA}
USBVCC
USBDP
USBDM
V_{SSA}
44 43 42 41 40 39 38 37 36 35 34
V_{PP} 1
PD1 2
PD0 3
PC7 4
MOSI / PC6 5
IT12 / MISO / PC5 6
IT11 / $\overline{SS}$ / PC4 7
IT10 / SCK / PC3 8
IT9 / PC2 9
OSCIN 10
OSCOUT 11
33 $\overline{RESET}$
32 PA0 / AIN0 / IT1 / USBOE
31 PA1 / AIN1 / IT2
30 PA2 / AIN2 / IT3
29 PA3 / AIN3 / IT4
28 PA4 / AIN4
27 PA5 / AIN5
26 PA6 / AIN6
25 PA7 / AIN7
24 PB0 (HS) / MCO
23 PB1 (HS) / RDI
12 13 14 15 16 17 18 19 20 21 22
V_{SS}
V_{DD}
PC1
PC0
IT8 / PWM1 / PB7
N.C.
IT7 / PWM0 / PB6 (HS)
ICCCLK / IT6 / ARTIC2 / PB5 (HS)
ICCDATA / IT5 / ARTIC1 / PB4 (HS)
ARTCLK / PB3 (HS)
TDO / PB2 (HS)
PD6 1
PD5 2
PD4 3
PD3 4
PD2 5
V_{PP} 6
PD1 7
PD0 8
PC7 9
MOSI / PC6 10
IT12 / MISO / PC5 11
IT11 / $\overline{SS}$ / PC4 12
IT10 / SCK / PC3 13
IT9 / PC2 14
OSCIN 15
OSCOUT 16
V_{SS} 17
V_{DD} 18
PC1 19
PC0 20
IT8 / PWM1 / PB7 (HS) 21
42 V_{DDA}
41 USBVCC
40 USBDP
39 USBDM
38 V_{SSA}
37 $\overline{RESET}$
36 PA0 / AIN0 / IT1 / USBOE
35 PA1 / AIN1 / IT2
34 PA2 / AIN2 / IT3
33 PA3 / AIN3 / IT4
32 PA4 / AIN4
31 PA5 / AIN5
30 PA6 / AIN6
29 PA7 / AIN7
28 PB0 (HS) / MCO
27 PB1 (HS) / RDI
26 PB2 (HS) / TDO
25 PB3 (HS) / ARTCLK
24 PB4 (HS) / ARTIC1 / IT5 / ICCDATA
23 PB5 (HS) / ARTIC2 / IT6 / ICCCLK
22 PB6 (HS) / PWM0 / IT7

PIN DESCRIPTION (Cont'd)

Figure 3. 34-Pin SO and 32-Pin SDIP Package Pinouts

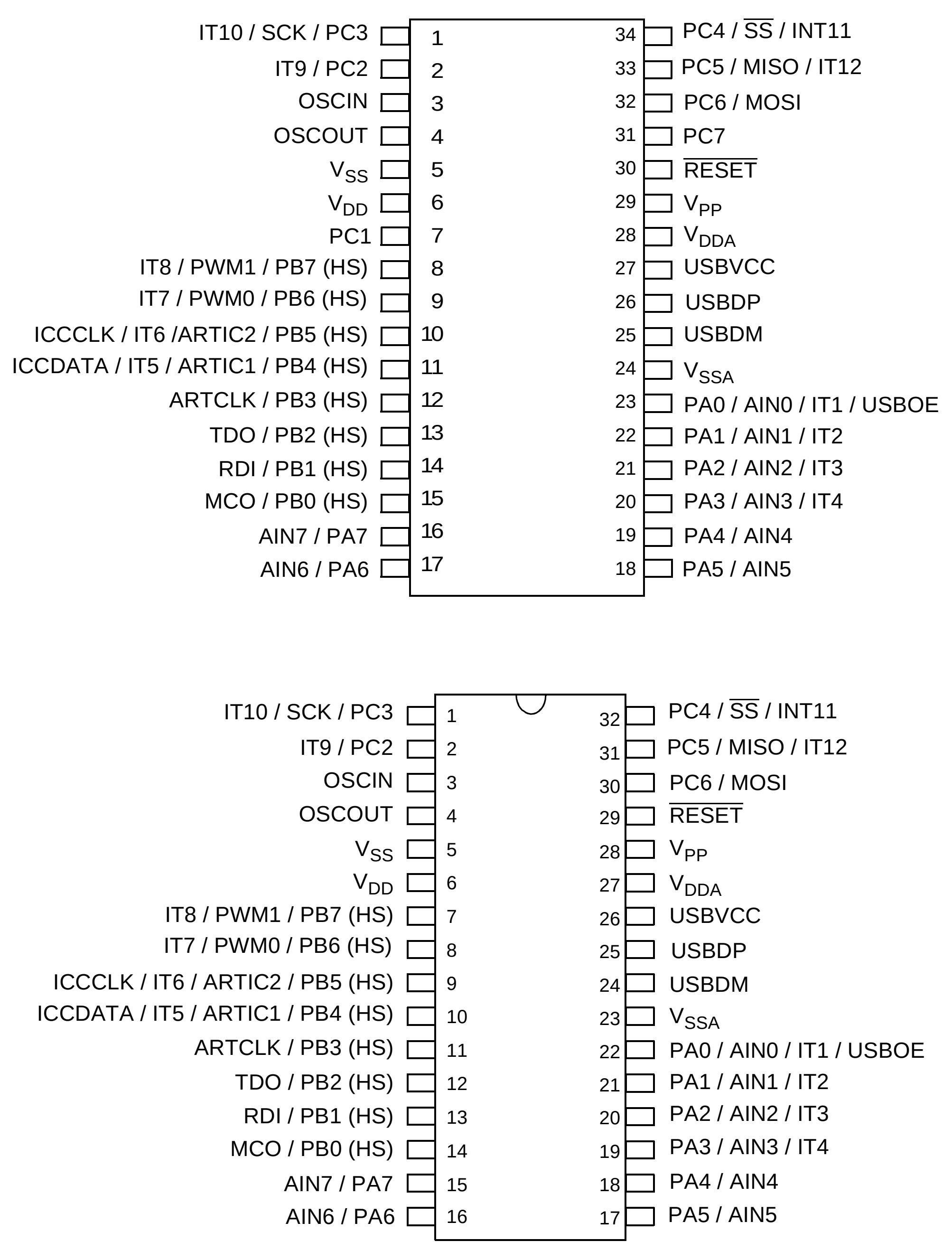

IT10 / SCK / PC3 1
IT9 / PC2 2
OSCIN 3
OSCOUT 4
V_{SS} 5
V_{DD} 6
PC1 7
IT8 / PWM1 / PB7 (HS) 8
IT7 / PWM0 / PB6 (HS) 9
ICCCLK / IT6 /ARTIC2 / PB5 (HS) 10
ICCDATA / IT5 / ARTIC1 / PB4 (HS) 11
ARTCLK / PB3 (HS) 12
TDO / PB2 (HS) 13
RDI / PB1 (HS) 14
MCO / PB0 (HS) 15
AIN7 / PA7 16
AIN6 / PA6 17
34 PC4 / $\overline{SS}$ / INT11
33 PC5 / MISO / IT12
32 PC6 / MOSI
31 PC7
30 $\overline{RESET}$
29 V_{PP}
28 V_{DDA}
27 USBVCC
26 USBDP
25 USBDM
24 V_{SSA}
23 PA0 / AIN0 / IT1 / USBOE
22 PA1 / AIN1 / IT2
21 PA2 / AIN2 / IT3
20 PA3 / AIN3 / IT4
19 PA4 / AIN4
18 PA5 / AIN5
IT10 / SCK / PC3 1
IT9 / PC2 2
OSCIN 3
OSCOUT 4
V_{SS} 5
V_{DD} 6
IT8 / PWM1 / PB7 (HS) 7
IT7 / PWM0 / PB6 (HS) 8
ICCCLK / IT6 / ARTIC2 / PB5 (HS) 9
ICCDATA / IT5 / ARTIC1 / PB4 (HS) 10
ARTCLK / PB3 (HS) 11
TDO / PB2 (HS) 12
RDI / PB1 (HS) 13
MCO / PB0 (HS) 14
AIN7 / PA7 15
AIN6 / PA6 16
32 PC4 / $\overline{SS}$ / INT11
31 PC5 / MISO / IT12
30 PC6 / MOSI
29 $\overline{RESET}$
28 V_{PP}
27 V_{DDA}
26 USBVCC
25 USBDP
24 USBDM
23 V_{SSA}
22 PA0 / AIN0 / IT1 / USBOE
21 PA1 / AIN1 / IT2
20 PA2 / AIN2 / IT3
19 PA3 / AIN3 / IT4
18 PA4 / AIN4
17 PA5 / AIN5

Figure 4. 20-pin SO20 Package Pinout

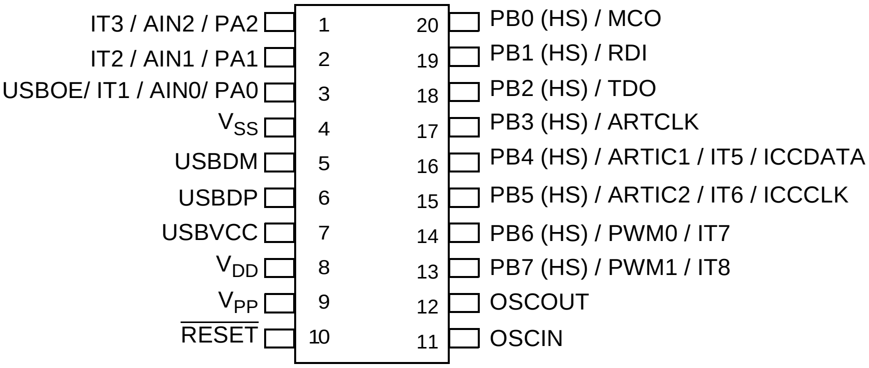
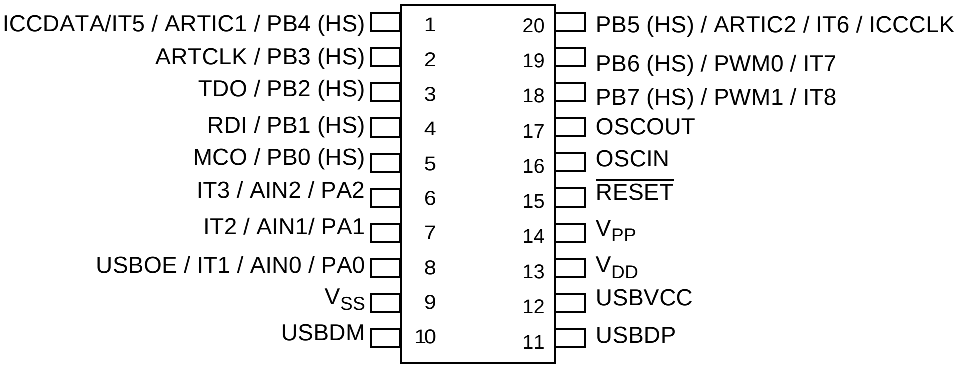

IT3 / AIN2 / PA2 1
IT2 / AIN1 / PA1 2
USBOE/ IT1 / AIN0/ PA0 3
V_{SS} 4
USBDM 5
USBDP 6
USBVCC 7
V_{DD} 8
V_{PP} 9
$\overline{\text{RESET}}$ 10
20 PB0 (HS) / MCO
19 PB1 (HS) / RDI
18 PB2 (HS) / TDO
17 PB3 (HS) / ARTCLK
16 PB4 (HS) / ARTIC1 / IT5 / ICCDATA
15 PB5 (HS) / ARTIC2 / IT6 / ICCCLK
14 PB6 (HS) / PWM0 / IT7
13 PB7 (HS) / PWM1 / IT8
12 OSCOUT
11 OSCIN

Figure 5. 20-pin DIP20 Package Pinout


ICCDATA/IT5 / ARTIC1 / PB4 (HS) 1
ARTCLK / PB3 (HS) 2
TDO / PB2 (HS) 3
RDI / PB1 (HS) 4
MCO / PB0 (HS) 5
IT3 / AIN2 / PA2 6
IT2 / AIN1/ PA1 7
USBOE / IT1 / AIN0 / PA0 8
V_{SS} 9
USBDM 10
20 PB5 (HS) / ARTIC2 / IT6 / ICCCLK
19 PB6 (HS) / PWM0 / IT7
18 PB7 (HS) / PWM1 / IT8
17 OSCOUT
16 OSCIN
15 $\overline{\text{RESET}}$
14 V_{PP}
13 V_{DD}
12 USBVCC
11 USBDP

PIN DESCRIPTION (Cont'd)

Legend / Abbreviations:

Type:	I = Input, O = Output, S = Supply
Input level:	A = Dedicated analog input
Input 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 (\ =falling edge, / =rising edge), ana = analog
- Output: OD = open drain, T = true open drain (N buffer 8mA@0.4 V), PP = push-pull

Note 1: Peripheral not present on all devices. Refer to "Device Summary" on page 1.

Table 1. Device Pin Description

Pin n°						Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
TQFP44	DIP42	SO34	DIP32	SO20	DIP20			Input	Output	Input				Output			
										float	wpu	int	ana	OD	PP		
1	6	29	28	9	14	V_{PP}	S				x					FLASH programming voltage (12V), must be tied low in user mode.	
2	7	-	-	-	-	PD1	I/O	C_T			x				x	**Port D1**	
3	8	-	-	-	-	PD0	I/O	C_T			x				x	**Port D0**	
4	9	31	-	-	-	PC7	I/O	C_T			x				x	**Port C7**	
5	10	32	30	-	-	PC6/MOSI	I/O	C_T			x				x	**Port C6**	SPI Master Out / Slave In 1)
6	11	33	31	-	-	PC5/MISO/IT12	I/O	C_T			x	x			x	**Port C5**	SPI Master In / Slave Out 1) / Interrupt 12 input
7	12	34	32	-	-	PC4/$\overline{SS}$/IT11	I/O	C_T			x	x			x	**Port C4**	SPI Slave Select (active low) 1)/ Interrupt 11 input
8	13	1	1	-	-	PC3/SCK/IT10	I/O	C_T			x	x			x	**Port C3**	SPI Serial Clock 1)/ Interrupt 10 input
9	14	2	2	-	-	PC2/IT9	I/O	C_T			x	x			x	**Port C2**	Interrupt 9 input
10	15	3	3	11	16	OSCIN										These pins are used connect an external clock source to the on-chip main oscillator.	
11	16	4	4	12	17	OSCOUT											
12	17	5	5	4	9	V_{SS}	S									Digital Ground Voltage	
13	18	6	6	8	13	V_{DD}	S									Digital Main Power Supply Voltage	
14	19	7	-	-	-	PC1	I/O	C_T		x				T		**Port C1**	
15	20	-	-	-	-	PC0	I/O	C_T		x				T		**Port C0**	
16	21	8	7	13	18	PB7/PWM1/IT8	I/O	C_T	HS	x		\			x	**Port B7**	ART PWM output 1/ Interrupt 8 input
17	-	-				N.C.										Not Connected	

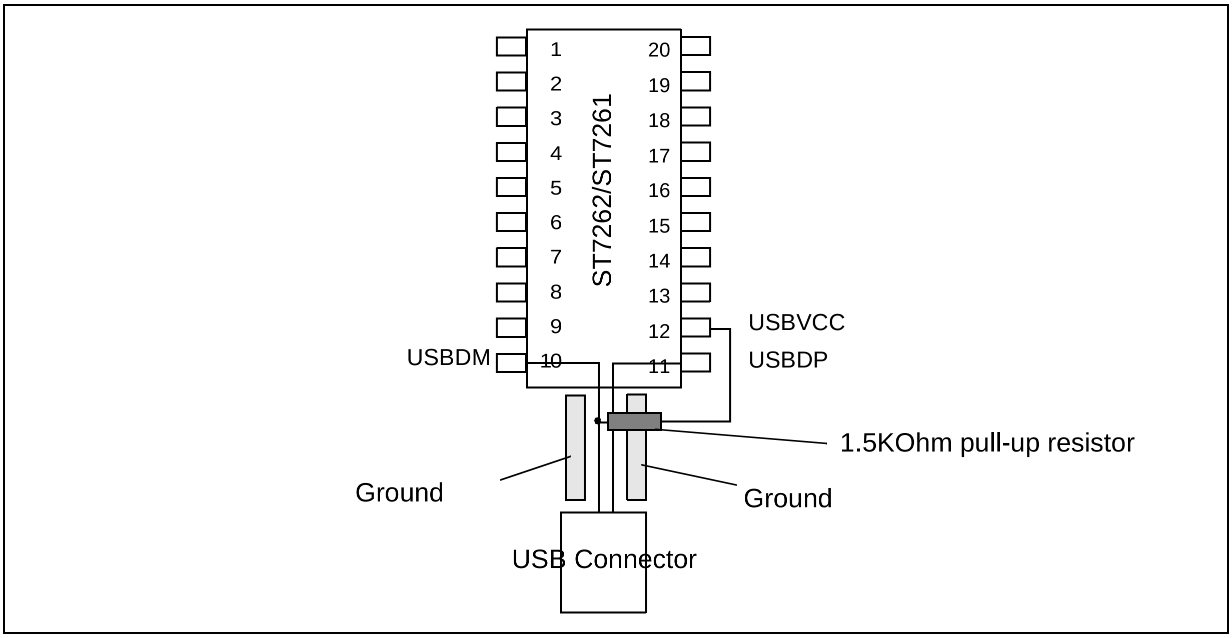
Pin n°						Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
TQFP44	DIP42	SO34	DIP32	SO20	DIP20			Input	Output	Input				Output			
										float	wpu	int	ana	OD	PP		
18	22	9	8	14	19	PB6/PWM0/IT7	I/O	C_T	HS	x		\			x	**Port B6**	ART PWM output 0/ Interrupt 7 input
19	23	10	9	15	20	PB5/ARTIC2/IT6/ ICCCLK	I/O	C_T	HS	x		/			x	**Port B5**	ART Input Capture 2/ Interrupt 6 input/ In-Circuit Communication Clock input
20	24	11	10	16	1	PB4/ARTIC1/IT5/ ICCDATA	I/O	C_T	HS	x		/			x	**Port B4**	ART Input Capture 1/Interrupt 5 input/ In-Circuit Communication Data input
21	25	12	11	17	2	PB3/ARTCLK	I/O	C_T	HS	x					x	**Port B3**	ART Clock input
22	26	13	12	18	3	PB2/TDO	I/O	C_T	HS	x					x	**Port B2**	SCI Transmit Data Output 1)
23	27	14	13	19	4	PB1/RDI	I/O	C_T	HS	x					x	**Port B1**	SCI Receive Data Input 1)
24	28	15	14	20	5	PB0/MCO	I/O	C_T	HS	x					x	**Port B0**	CPU clock output
25	29	16	15	-	-	PA7/AIN7	I/O	C_T		x			x		x	**Port A7**	ADC Analog Input 7
26	30	17	16	-	-	PA6/AIN6	I/O	C_T		x			x		x	**Port A6**	ADC Analog Input 6
27	31	18	17	-	-	PA5/AIN5	I/O	C_T		x			x		x	**Port A5**	ADC Analog Input 5
28	32	19	18	-	-	PA4/AIN4	I/O	C_T		x			x		x	**Port A4**	ADC Analog Input 4
29	33	20	19	-	-	PA3/AIN3/IT4	I/O	C_T		x		\	x		x	**Port A3**	ADC Analog Input 3/ Interrupt 4 input
30	34	21	20	1	6	PA2/AIN2/IT3	I/O	C_T		x		\	x		x	**Port A2**	ADC Analog Input 2/ Interrupt 3 input
31	35	22	21	2	7	PA1/AIN1/IT2	I/O	C_T		x		\	x		x	**Port A1**	ADC Analog Input 1/ Interrupt 2 input
32	36	23	22	3	8	PA0/AIN0/IT1/ USBOE	I/O	C_T		x		\	x		x	**Port A0**	ADC Analog Input 0/ Interrupt 1 input/ USB Output Enable
33	37	30	29	10	15	$\overline{\text{RESET}}$	I/O	C								Top priority non maskable interrupt (active low)	
34	38	24	23	-	-	V_{SSA}	S									Analog Ground Voltage, must be connected externally to V_{SS}.	
35	39	25	24	5	10	USBDM	I/O									USB bidirectional data (data -)	
36	40	26	25	6	11	USBDP	I/O									USB bidirectional data (data +)	
37	41	27	26	7	12	USBVCC	S									USB power supply 3.3V output	
38	42	28	27	-	-	V_{DDA}	S									Analog Power Supply Voltage, must be connected externally to V_{DD}.	
39	-	-	-	-	-	N.C.										Not Connected	
40	1	-	-	-	-	PD6	I/O	C_T			x				x	**Port D6**	
41	2	-	-	-	-	PD5	I/O	C_T			x				x	**Port D5**	
42	3	-	-	-	-	PD4	I/O	C_T			x				x	**Port D4**	

Pin n°						Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
										Input				Output			
TQFP44	DIP42	SO34	DIP32	SO20	DIP20			Input	Output	float	wpu	int	ana	OD	PP		
43	4	-	-	-	-	PD3	I/O	C_T			x				x	**Port D3**	
44	5	-	-	-	-	PD2	I/O	C_T			x				x	**Port D2**	

2.1 PCB LAYOUT RECOMMENDATION

In the case of DIP20 devices the user should layout the PCB so that the DIP20 ST7262/ST7261 device and the USB connector are centered on the same axis ensuring that the D- and D+ lines are of equal length. Refer to Figure 6

Figure 6. Recommended PCB Layout for USB Interface with DIP20 package


ST7262/ST7261
1
2
3
4
5
6
7
8
9
10
20
19
18
17
16
15
14
13
12
11
USBVCC
USBDM
USBDP
1.5KOhm pull-up resistor
Ground
Ground
USB Connector

3 REGISTER & MEMORY MAP

As shown in the Figure 7, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 64 bytes of register locations, 768 bytes of RAM and up to 16 Kbytes of user program memory. The RAM space includes up to 128 bytes for the stack from 0100h to 017Fh.

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

Figure 7. Memory Map

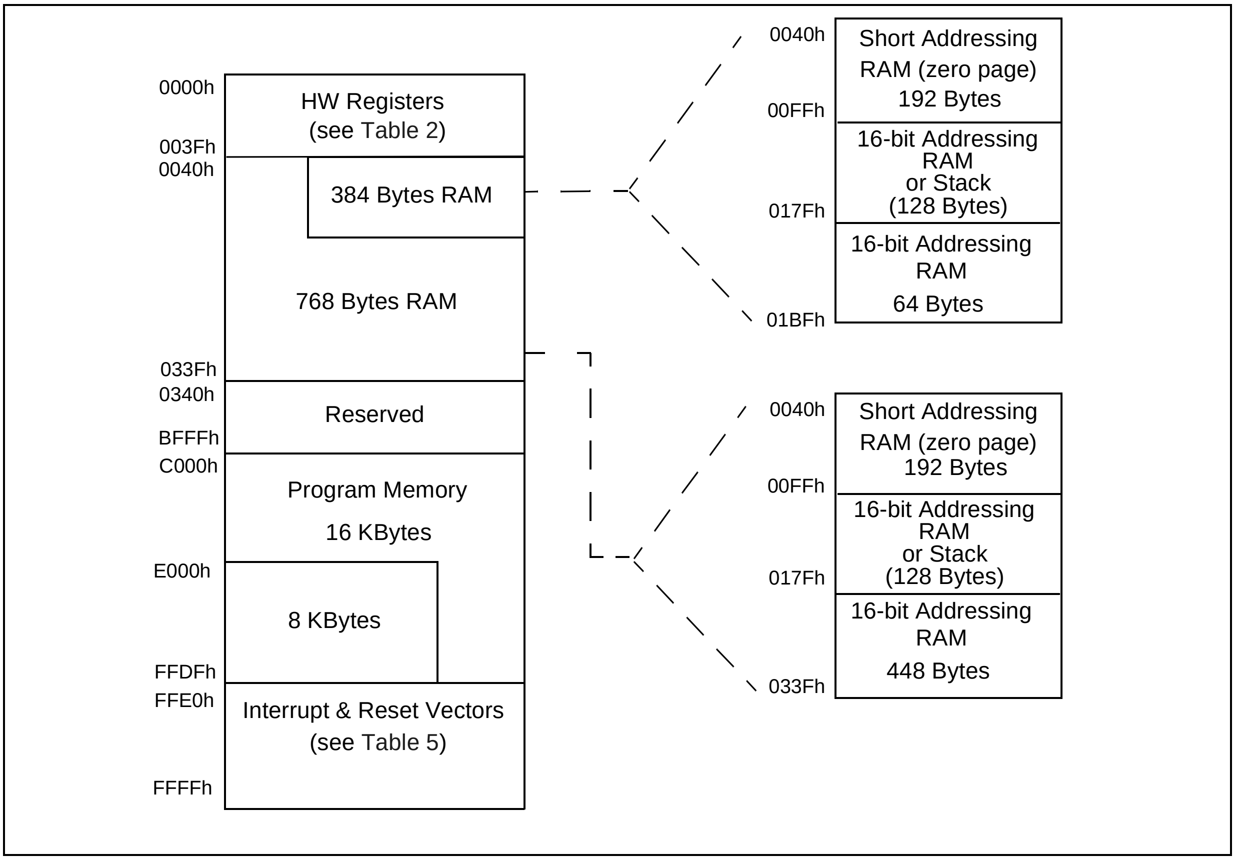

0000h
HW Registers
(see Table 2)
003Fh
0040h
384 Bytes RAM
768 Bytes RAM
033Fh
0340h
Reserved
BFFFh
C000h
Program Memory
16 KBytes
E000h
8 KBytes
FFDFh
FFE0h
Interrupt & Reset Vectors
(see Table 5)
FFFFh
0040h
Short Addressing RAM (zero page) 192 Bytes
00FFh
16-bit Addressing RAM or Stack (128 Bytes)
017Fh
16-bit Addressing RAM 64 Bytes
01BFh
0040h
Short Addressing RAM (zero page) 192 Bytes
00FFh
16-bit Addressing RAM or Stack (128 Bytes)
017Fh
16-bit Addressing RAM 448 Bytes
033Fh

Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h	Port A	PADR	Port A Data Register	00h	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h	Port B	PBDR	Port B Data Register	00h	R/W
0003h		PBDDR	Port B Data Direction Register	00h	R/W.
0004h	Port C	PCDR	Port C Data Register	00h	R/W
0005h		PCDDR	Port C Data Direction Register	00h	R/W
0006h	Port D	PDDR	Port D Data Register	00h	R/W
0007h		PDDDR	Port D Data Direction Register	00h	R/W
0008h		ITRFRE1	Interrupt Register 1	00h	R/W
0009h		MISC	Miscellaneous Register	00h	R/W
000Ah	ADC	ADCDRMSB	ADC Data Register (bit 9:2)	00h	Read Only
000Bh		ADCDRLSB	ADC Data Register (bit 1:0)	00h	Read Only
000Ch		ADCCSR	ADC Control Status Register	00h	R/W
000Dh	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
000Eh 0010h	Reserved Area (3 Bytes)				
0011h	SPI	SPIDR	SPI Data I/O Register	xxh	R/W
0012h		SPICR	SPI Control Register	0xh	R/W
0013h		SPICSR	SPI Control Status Register	00h	Read Only
0014h	PWM ART	PWMDCR1	PWM AR Timer Duty Cycle Register 1	00h	R/W
0015h		PWMDCR0	PWM AR Timer Duty Cycle Register 0	00h	R/W
0016h		PWMCR	PWM AR Timer Control Register	00h	R/W
0017h		ARTCSR	Auto-Reload Timer Control/Status Register	00h	R/W
0018h		ARTCAR	Auto-Reload Timer Counter Access Register	00h	R/W
0019h		ARTARR	Auto-Reload Timer Auto-Reload Register	00h	R/W
001Ah		ARTICCSR	ART Input Capture Control/Status Register	00h	R/W
001Bh		ARTICR1	ART Input Capture Register 1	00h	Read Only
001Ch		ARTICR2	ART Input Capture Register 2	00h	Read Only
001Dh	SCI	SCIERPR	SCI Extended Receive Prescaler register	00h	R/W
001Eh		SCIETPR	SCI Extended Transmit Prescaler Register	00h	R/W
001Fh			Reserved Area	--	
0020h		SCISR	SCI Status register	C0h	Read Only
0021h		SCIDR	SCI Data register	xxh	R/W
0022h		SCIBRR	SCI Baud Rate Register	00h	R/W
0023h		SCICR1	SCI Control Register 1	x000 0000b	R/W
0024h		SCICR2	SCI Control Register 2	00h	R/W

Address	Block	Register Label	Register Name	Reset Status	Remarks
0025h	USB	USBPIDR	USB PID Register	x0h	Read Only
0026h		USBDMAR	USB DMA Address register	xxh	R/W
0027h		USBIDR	USB Interrupt/DMA Register	x0h	R/W
0028h		USBISTR	USB Interrupt Status Register	00h	R/W
0029h		USBIMR	USB Interrupt Mask Register	00h	R/W
002Ah		USBCTLR	USB Control Register	06h	R/W
002Bh		USBDADDR	USB Device Address Register	00h	R/W
002Ch		USBEP0RA	USB Endpoint 0 Register A	0000 xxxxb	R/W
002Dh		USBEP0RB	USB Endpoint 0 Register B	80h	R/W
002Eh		USBEP1RA	USB Endpoint 1 Register A	0000 xxxxb	R/W
002Fh		USBEP1RB	USB Endpoint 1 Register B	0000 xxxxb	R/W
0030h		USBEP2RA	USB Endpoint 2 Register A	0000 xxxxb	R/W
0031h		USBEP2RB	USB Endpoint 2 Register B	0000 xxxxb	R/W
0032h to 0035h	Reserved Area (4 Bytes)				
0036h	TBU	TBUCV	TBU Counter Value Register	00h	R/W
0037h		TBUCSR	TBU Control/Status Register	00h	R/W
0038h	FLASH	FCSR	Flash Control Status Register	00h	R/W
0039h		ITRFRE2	Interrupt Register 2	00h	R/W
003Ah to 003Fh	Reserved Area (6 Bytes)				

4 FLASH PROGRAM MEMORY

4.1 Introduction

The ST7 dual voltage Flash is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a Byte-by-Byte basis using an external V_{PP} supply.

The Flash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (In-Circuit Programming) or IAP (In-Application Programming).

The array matrix organisation allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main Features

- Three Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (In-Circuit Programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (In-Application Programming) In this mode, all sectors except Sector 0, can be programmed or erased without removing the device from the application board and while the application is running.
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Read-out protection against piracy
- Register Access Security System (RASS) to prevent accidental programming or erasing

4.3 Structure

The Flash memory is organised in sectors and can be used for both code and data storage.

Depending on the overall FLASH memory size in the microcontroller device, there are up to three user sectors (see Table 3). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

The first two sectors have a fixed size of 4 Kbytes (see Figure 8). They are mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

Table 3. Sectors available in FLASH devices

Flash Memory Size (bytes)	Available Sectors
4K	Sector 0
8K	Sectors 0,1
> 8K	Sectors 0,1, 2

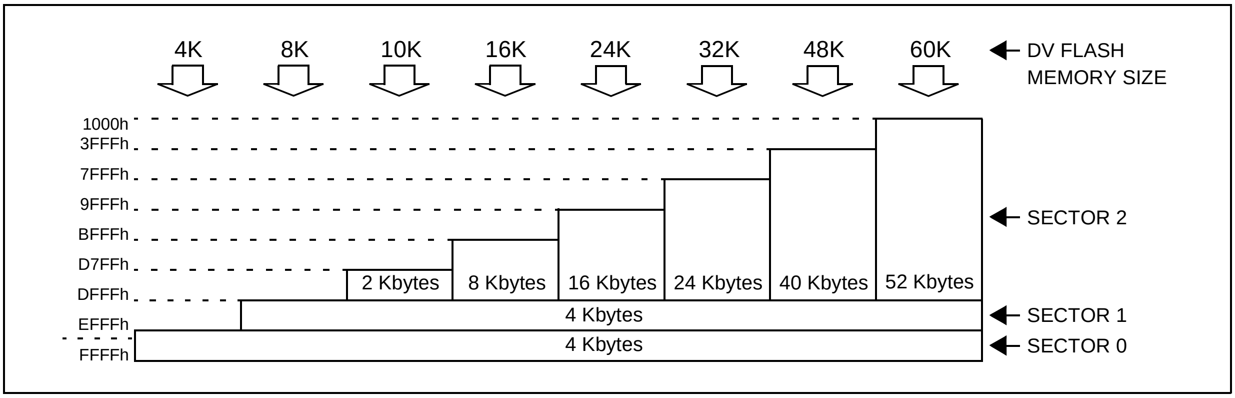
4.4 Program Memory Read-out Protection

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

When this option is selected, the programs and data stored in the program memory (Flash or ROM) are protected against read-out piracy (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire program memory is first automatically erased.

Refer to the Option Byte description for more details.

Figure 8. Memory Map and Sector Address



FLASH PROGRAM MEMORY (Cont'd)

4.5 ICP (In-Circuit Programming)

To perform ICP the microcontroller must be switched to ICC (In-Circuit Communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see Figure 9). For more details on the pin locations, refer to the device pinout description.

ICP needs five signals to be connected to the programming tool. These signals are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- V_{PP}: programming voltage

When the device is not yet configured to support the application clock source (option byte not yet programmed) or if the option bytes have to be programmed using ICP, one more pin has to be connected:

- OSCIN: main clock input for external source

When the device is not supplied (V_{DD}) by the application, one more pin has to be connected:

- V_{DD}: main power supply

CAUTIONS:

1. If $\overline{\text{RESET}}$, ICCCLK or ICCDATA pins are used for other purposes in the application, a serial resistor has to be implemented to avoid a conflict in case one of the other devices forces the signal level.

2. As soon as the external controller is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application.

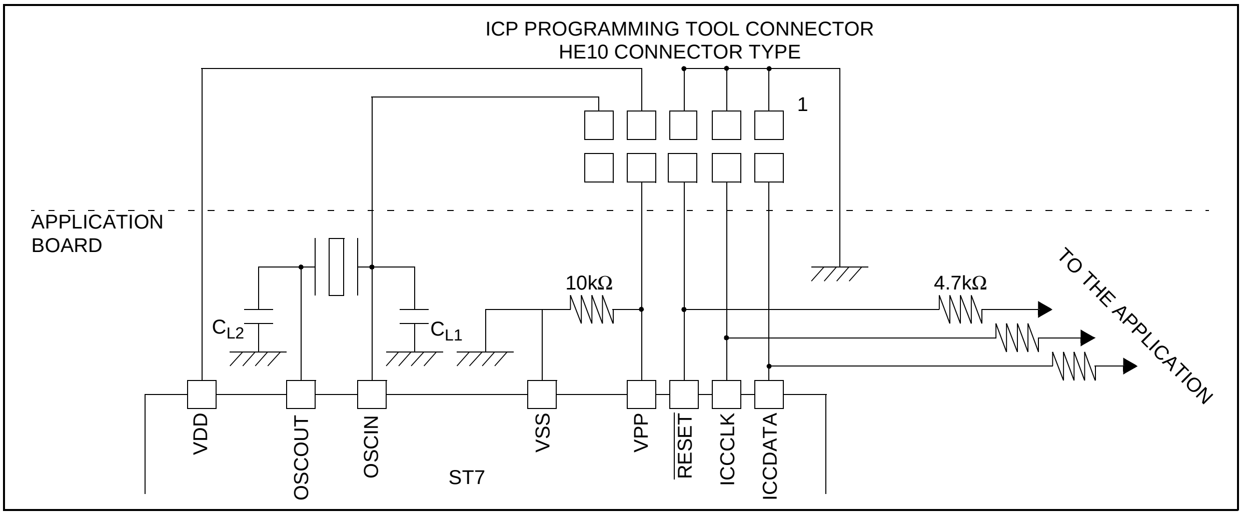
Note: To develop a custom programming tool, refer to the ST7 FLASH Programming and ICC Reference Manual which gives full details on the ICC protocol hardware and software.

4.6 IAP (In-Application Programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from the SPI, SCI, USB or CAN interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

Figure 9. Typical ICP Interface



FLASH PROGRAM MEMORY (Cont'd)

4.6.1 Register Description

FLASH CONTROL/STATUS REGISTER (FCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	0	0

This register is reserved for use by Programming Tool software. It controls the FLASH programming and erasing operations. For details on customizing FLASH programming methods and In-Circuit Testing, refer to the ST7 FLASH Programming and ICC Reference Manual.

5 CENTRAL PROCESSING UNIT

5.1 INTRODUCTION

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

5.2 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

5.3 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)

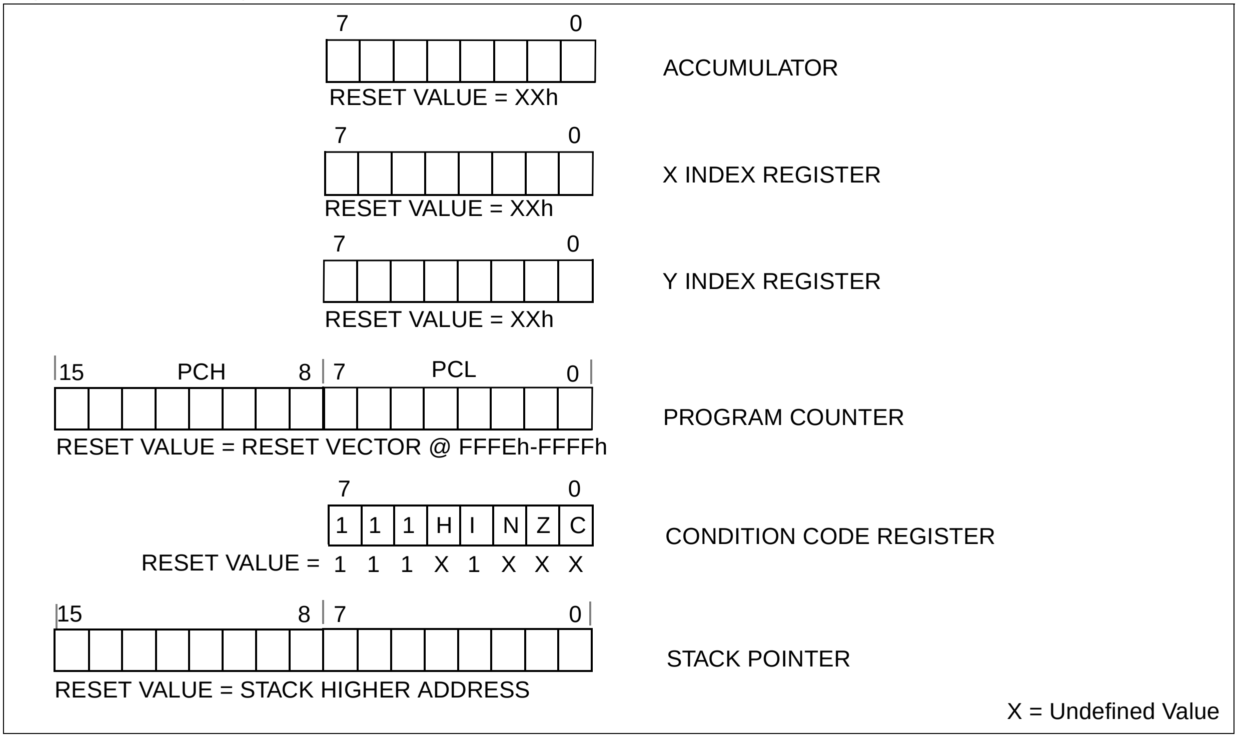
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

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


7 0
ACCUMULATOR
RESET VALUE = XXh
7 0
X INDEX REGISTER
RESET VALUE = XXh
7 0
Y INDEX REGISTER
RESET VALUE = XXh
15 PCH 8 7 PCL 0
PROGRAM COUNTER
RESET VALUE = RESET VECTOR @ FFFEh-FFFFh
7 0
1 1 1 H I N Z C
CONDITION CODE REGISTER
RESET VALUE = 1 1 1 X 1 X X X
15 8 7 0
STACK POINTER
RESET VALUE = STACK HIGHER ADDRESS
X = Undefined Value

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

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

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

This bit is accessed by the JRMI and JRPL instructions.

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

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

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

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

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

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

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 7Fh

15							8
0	0	0	0	0	0	0	1

7							0
0	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 128 bytes deep, the 9 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 SP6 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.

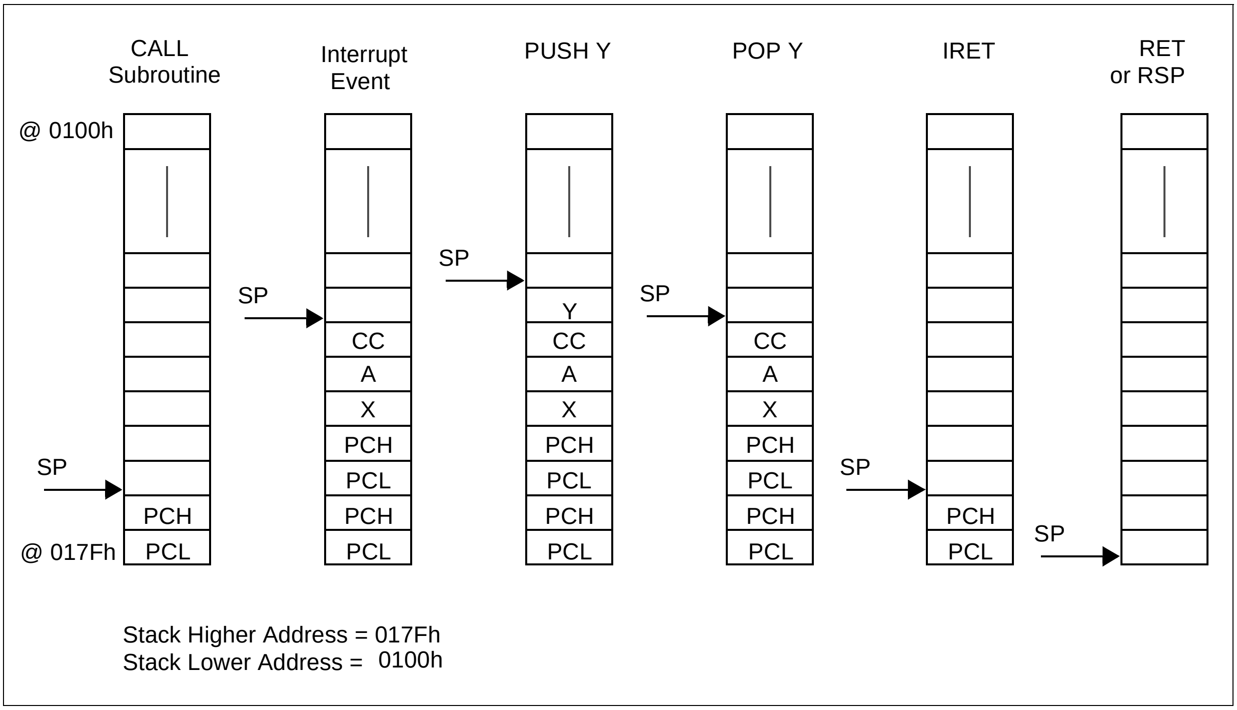
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the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



6 CLOCKS AND RESET

6.1 CLOCK SYSTEM

6.1.1 General Description

The MCU accepts either a Crystal or Ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}), by dividing by 3 and multiplying by 2. By setting the OSC12/6 bit in the option byte, a 12 MHz external clock can be used giving an internal frequency of 8 MHz while maintaining a 6 MHz clock for USB (refer to Figure 14).

The internal clock signal (f_{CPU}) consists of a square wave with a duty cycle of 50%.

It is further divided by 1, 2, 4 or 8 depending on the Slow Mode Selection bits in the Miscellaneous register (SMS[1:0])

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz or ceramic resonator in the frequency range specified for f_{osc}. The circuit shown in Figure 13 is recommended when using a crystal, and Table 4 lists the recommended capacitors. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilization time.

Table 4. Recommended Values for 12 MHz Crystal Resonator

R_{SMAX}	**20 Ω**	**25 Ω**	**70 Ω**
C_{OSCIN}	56pF	47pF	22pF
C_{OSCOUT}	56pF	47pF	22pF
R_P	1-10 MΩ	1-10 MΩ	1-10 MΩ

Note: R_{SMAX} is the equivalent serial resistor of the crystal (see crystal specification).

6.1.2 External Clock input

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected, as shown on Figure 12. The t_{OXOV} specifications does not apply when using an external clock input. The equivalent specification of the external clock source should be used instead of t_{OXOV} (see Electrical Characteristics).

6.1.3 Clock Output Pin (MCO)

The internal clock (f_{CPU}) can be output on Port B0 by setting the MCO bit in the Miscellaneous register.

Figure 12. External Clock Source Connections

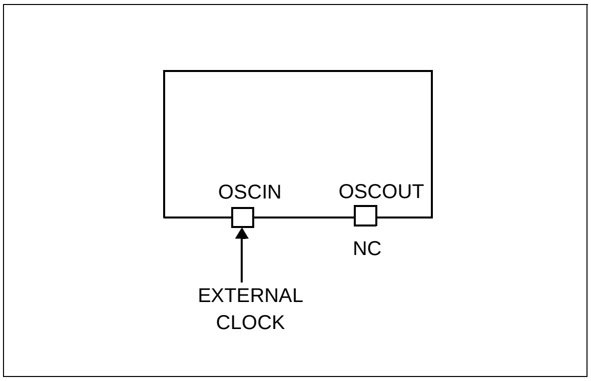


Figure 13. Crystal/Ceramic Resonator

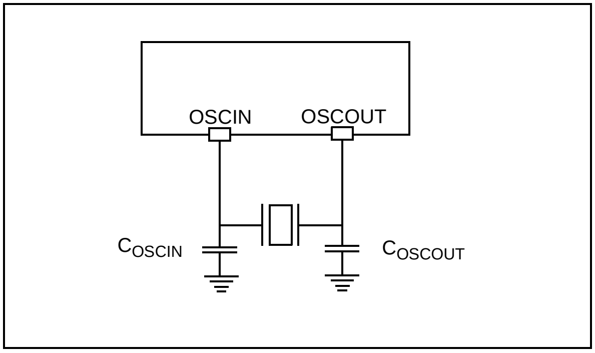
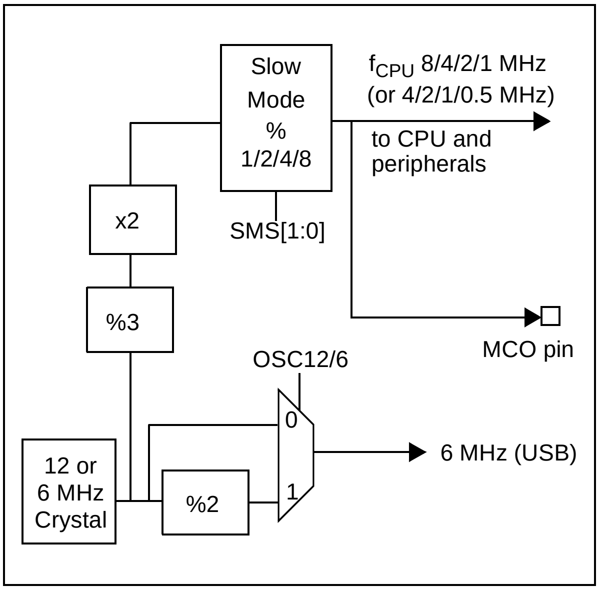


Figure 14. Clock block diagram



6.2 RESET

The Reset procedure is used to provide an orderly software start-up or to exit low power modes.

Three reset modes are provided: a low voltage reset, a watchdog reset and an external reset at the $\overline{\text{RESET}}$ pin.

A reset causes the reset vector to be fetched from addresses FFFEh and FFFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 514 CPU clock cycle delay from the time that the oscillator becomes active.

6.2.1 Low Voltage Reset

Low voltage reset circuitry generates a reset when V_{DD} is:

- below V_{IT+} when V_{DD} is rising,
- below V_{IT-} when V_{DD} is falling.

During low voltage reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

The Low Voltage Detector can be disabled by setting the LVD bit of the Option byte.

6.2.2 Watchdog Reset

When a watchdog reset occurs, the $\overline{\text{RESET}}$ pin is pulled low permitting the MCU to reset other devices as when low voltage reset (Figure 15).

6.2.3 External Reset

The external reset is an active low input signal applied to the $\overline{\text{RESET}}$ pin of the MCU.
As shown in Figure 18, the $\overline{\text{RESET}}$ signal must stay low for a minimum of one and a half CPU clock cycles.

An internal Schmitt trigger at the $\overline{\text{RESET}}$ pin is provided to improve noise immunity.

Figure 15. Low Voltage Reset functional Diagram

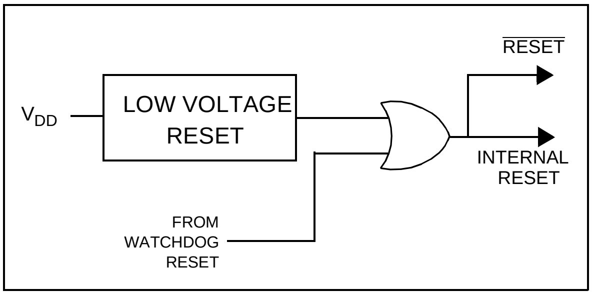
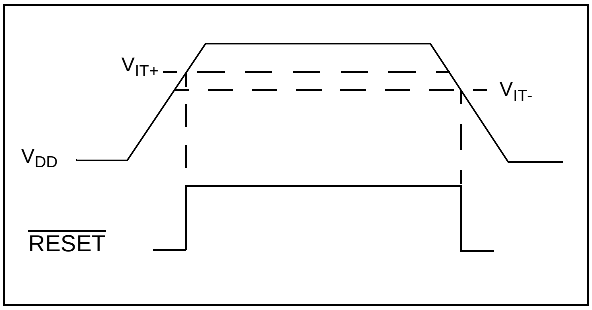


Figure 16. Low Voltage Reset **Signal Output**



Note: Typical hysteresis (V_{IT+}-V_{IT-}) of 250 mV is expected

Figure 17. Temporization Timing Diagram after an internal Reset

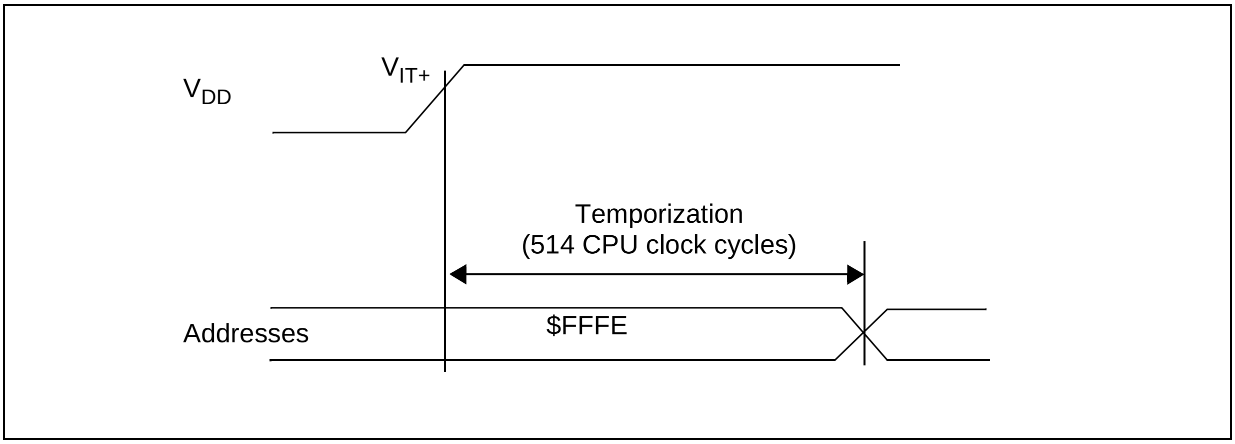
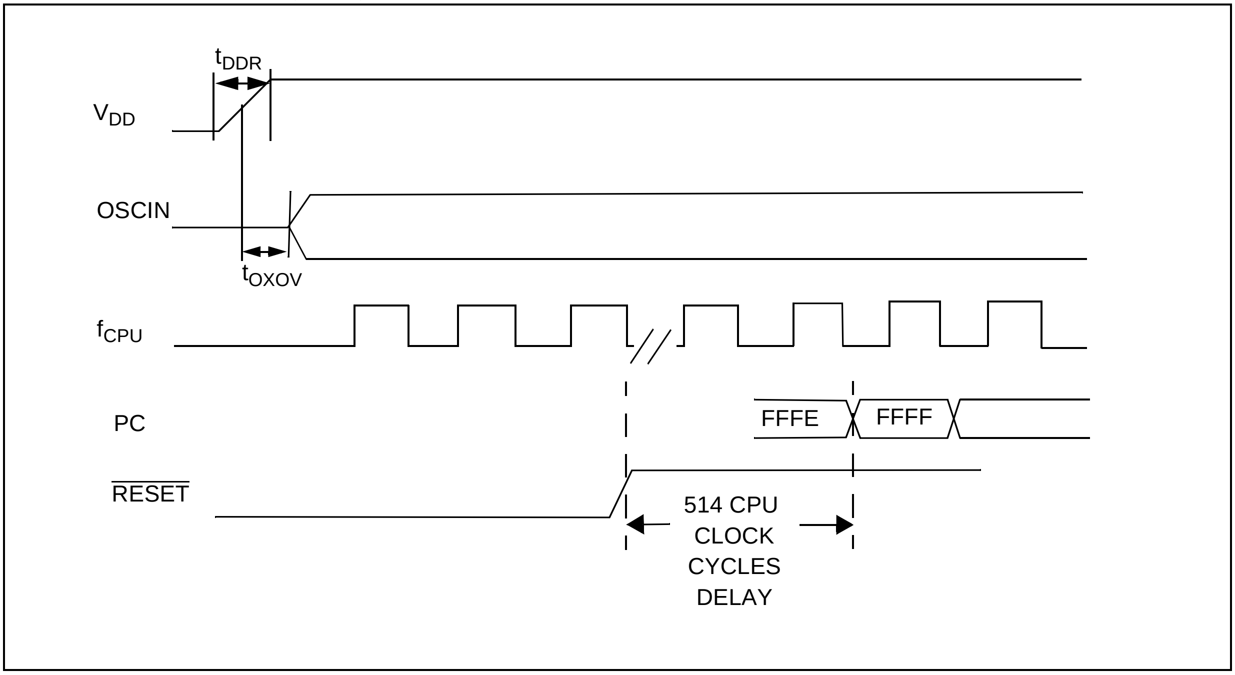
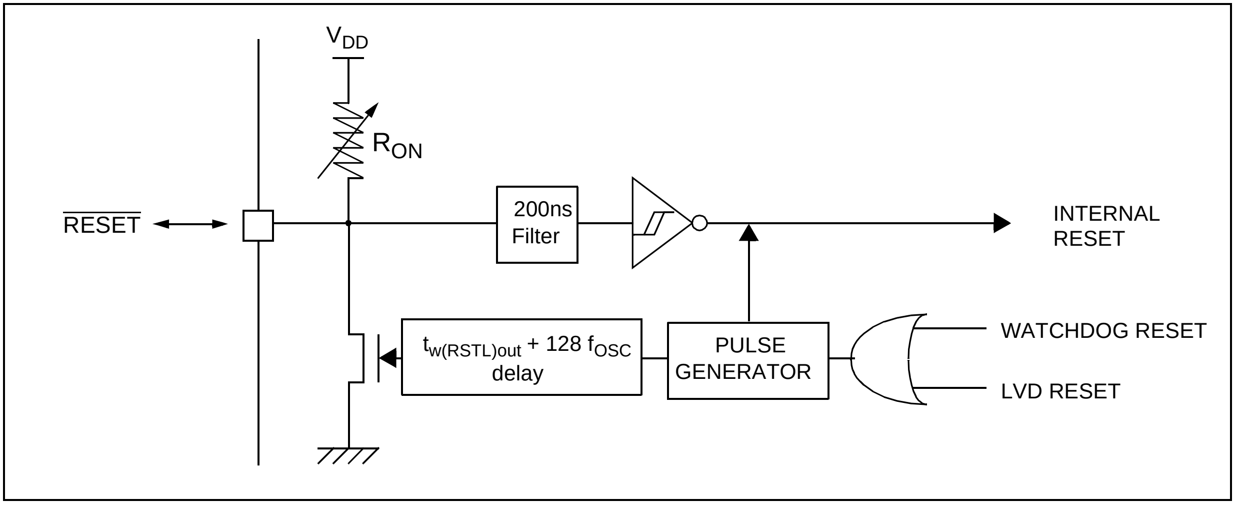


Figure 18. Reset Timing Diagram


t_{DDR}
V_{DD}
OSCIN
t_{OXOV}
f_{CPU}
PC
FFFE
FFFF
$\overline{\text{RESET}}$
514 CPU CLOCK CYCLES DELAY

Note: Refer to Electrical Characteristics for values of t_{DDR}, t_{OXOV}, V_{IT+} and V_{IT-}.

Figure 19. Reset Block Diagram


V_{DD}
R_{ON}
$\overline{\text{RESET}}$
200ns Filter
INTERNAL RESET
$t_{w(RSTL)out}$ + 128 f_{OSC} delay
PULSE GENERATOR
WATCHDOG RESET
LVD RESET

Note: The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).

7 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 20.
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, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case when several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping Table).

Interrupts and Low Power Mode

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

7.1 NON MASKABLE SOFTWARE INTERRUPT

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 20.

7.2 EXTERNAL INTERRUPTS

External interrupt vectors can be loaded into 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).

An 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.

Caution: 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.

7.3 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

– 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 20. Interrupt Processing Flowchart

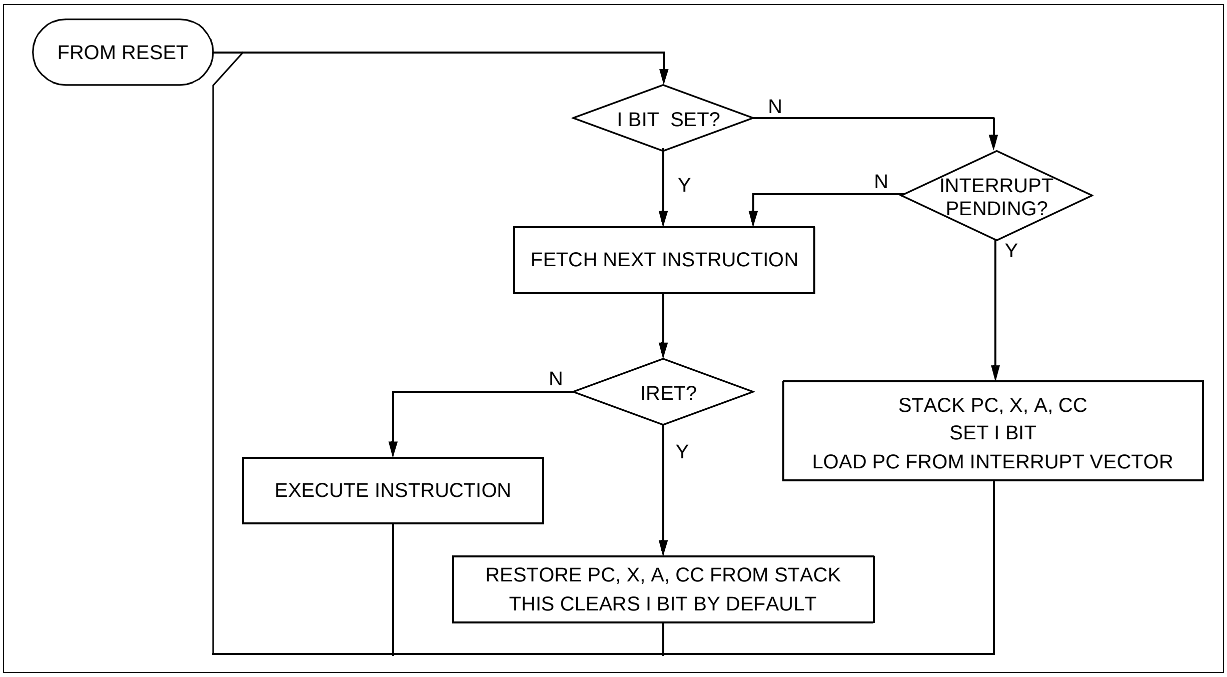

FROM RESET
I BIT SET?
N
Y
INTERRUPT PENDING?
N
Y
FETCH NEXT INSTRUCTION
IRET?
N
Y
STACK PC, X, A, CC
SET I BIT
LOAD PC FROM INTERRUPT VECTOR
EXECUTE INSTRUCTION
RESTORE PC, X, A, CC FROM STACK
THIS CLEARS I BIT BY DEFAULT

Table 5. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT	Address Vector
		Reset		Highest Priority	Yes	FFFEh-FFFFh
		TRAP software interrupt			No	FFFCh-FFFDh
0	ICP	FLASH Start programming NMI interrupt			Yes	FFFAh-FFFBh
1	USB	USB End Suspend interrupt	USBISTR		Yes	FFF8h-FFF9h
2	I/O Ports	Port A external interrupts IT[4:1]	ITRFRE1		Yes	FFF6h-FFF7h
3		Port B external interrupts IT[8:5]	ITRFRE1		Yes	FFF4h-FFF5h
4		Port C external interrupts IT[12:9]	ITRFRE2		Yes	FFF2h-FFF3h
5	TBU	Timebase Unit interrupt	TBUCSR		No	FFF0h-FFF1h
6	ART	ART/PWM Timer interrupt	ICCSR		Yes	FFEEh-FFEFh
7	SPI	SPI interrupt vector	SPISR		Yes	FFECh-FFEDh
8	SCI	SCI interrupt vector	SCISR		No	FFEAh-FFEBh
9	USB	USB interrupt vector	USBISTR	Lowest Priority	No	FFE8h-FFE9h
10	ADC	A/D End of conversion interrupt	ADCCSR		No	FFE6h-FFE7h

7.4 Interrupt Register

INTERRUPT REGISTER 1 (ITRFRE1)

Address: 0008h - Read/Write

Reset Value: 0000 0000 (00h)

7							0
IT8E	IT7E	IT6E	IT5E	IT4E	IT3E	IT2E	IT1E

Bit 7:0 = **ITiE** *Interrupt Enable*
0: I/O pin free for general purpose I/O
1: ITi external interrupt enabled.

Note: The corresponding interrupt is generated when:

– a rising edge occurs on the IT5/IT6 pins

– a falling edge occurs on the IT1, 2, 3, 4, 7 and 8 pins

INTERRUPT REGISTER 2 (ITRFRE2)

Address: 0039h - Read/Write

Reset Value: 0000 0000 (00h)

7							0
CTL3	CTL2	CTL1	CTL0	IT12E	IT11E	IT10E	IT9E

Bit 7:6 = **CTL[3:2]** *IT[12:11] Interrupt Sensitivity*

These bits are set and cleared by software.They are used to configure the edge and level sensitivity of the IT12 and IT11 external interrupt pins (this means that both must have the same sensitivity).

CTL3	**CTL2**	**IT[12:11] Sensitivity**
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Bit 5:4 = **CTL[1:0]** *IT[10:9]1nterrupt Sensitivity*

These bits are set and cleared by software.They are used to configure the edge and level sensitivity of the IT10 and IT9 external interrupt pins (this means that both must have the same sensitivity).

CTL1	**CTL0**	**IT[10:9] Sensitivity**
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Bit 3:0 = **ITiE** *Interrupt Enable*
0: I/O pin free for general purpose I/O
1: ITi external interrupt enabled.

8 POWER SAVING MODES

8.1 INTRODUCTION

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

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 3 and multiplied 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 oscillator status.

8.1.1 Slow Mode

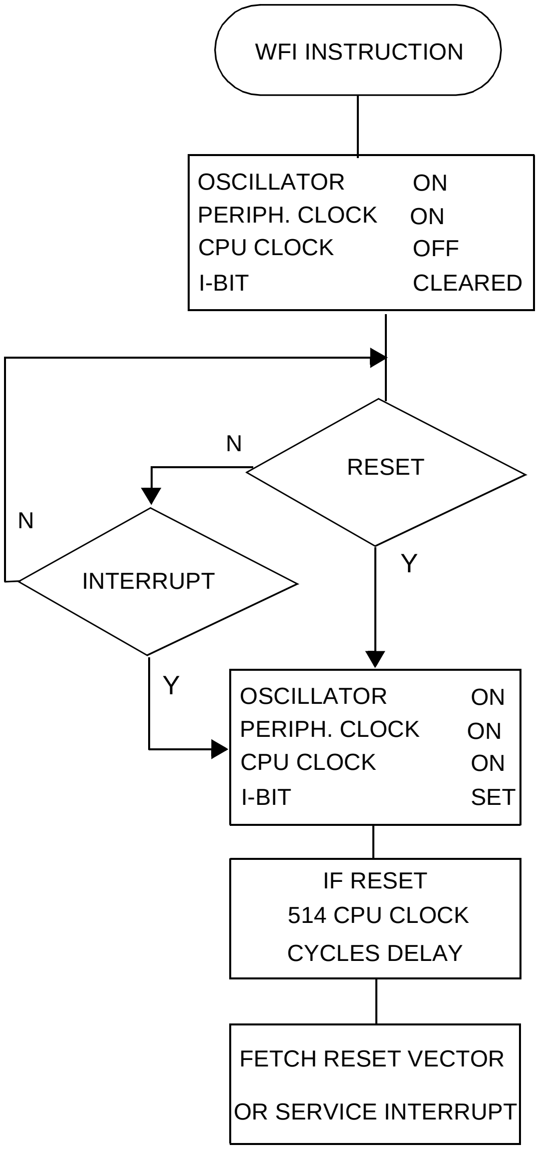
In Slow mode, the oscillator frequency can be divided by a value defined in the Miscellaneous Register. The CPU and peripherals are clocked at this lower frequency. Slow mode is used to reduce power consumption, and enables the user to adapt clock frequency to available supply voltage.

8.2 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 21.

Figure 21. WAIT Mode Flow Chart


WFI INSTRUCTION
OSCILLATOR ON
PERIPH. CLOCK ON
CPU CLOCK OFF
I-BIT CLEARED
N
RESET
N
INTERRUPT
Y
Y
OSCILLATOR ON
PERIPH. CLOCK ON
CPU CLOCK ON
I-BIT SET
IF RESET
514 CPU CLOCK
CYCLES DELAY
FETCH RESET VECTOR
OR SERVICE INTERRUPT

Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

POWER SAVING MODES (Cont'd)

8.3 HALT MODE

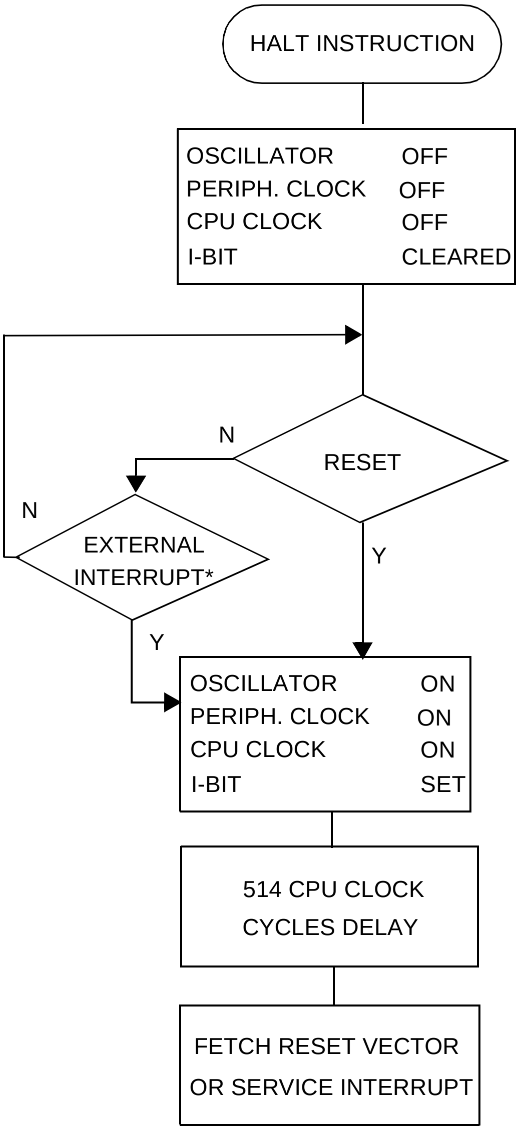
The HALT mode is the MCU lowest power consumption mode. The HALT mode is entered by executing the HALT instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals.

When entering HALT mode, the I bit in the Condition Code Register is cleared. Thus, any of the external interrupts (ITi or USB end suspend mode), are allowed and if an interrupt occurs, the CPU clock becomes active.

The MCU can exit HALT mode on reception of either an external interrupt on ITi, an end suspend mode interrupt coming from USB peripheral, or a reset. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 514 CPU clock cycles.

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

Figure 22. HALT Mode Flow Chart


HALT INSTRUCTION
OSCILLATOR OFF
PERIPH. CLOCK OFF
CPU CLOCK OFF
I-BIT CLEARED
N
RESET
N
EXTERNAL INTERRUPT*
Y
Y
OSCILLATOR ON
PERIPH. CLOCK ON
CPU CLOCK ON
I-BIT SET
514 CPU CLOCK CYCLES DELAY
FETCH RESET VECTOR OR SERVICE INTERRUPT

Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

9 I/O PORTS

9.1 INTRODUCTION

The I/O ports offer different functional modes:

transfer of data through digital inputs and outputs and for specific pins:

- Analog signal input (ADC)
- Alternate signal input/output for the on-chip peripherals.
- External interrupt generation

An I/O port is composed of up to 8 pins. Each pin can be programmed independently as digital input or digital output.

9.2 FUNCTIONAL DESCRIPTION

Each port is associated with 2 main registers:

- Data Register (DR)
- Data Direction Register (DDR)

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

Table 6. I/O Pin Functions

DDR	MODE
0	Input
1	Output

9.2.1 Input Modes

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

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

Notes:

1. All the inputs are triggered by a Schmitt trigger.
2. When switching from input mode to output mode, the DR register should be written first to output the correct value as soon as the port is configured as an output.

Interrupt function

When an external interrupt function of an I/O pin, is enabled using the ITFRE registers, an event on this I/O can generate an external Interrupt request to the CPU. The interrupt sensitivity is programmable, the options are given in the description of the ITRFRE interrupt registers.

Each pin can independently generate an Interrupt request.

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

9.2.2 Output Mode

The pin is configured in output mode by setting the corresponding DDR register bit (see Table 7).

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

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

9.2.3 Alternate Functions

Digital Alternate Functions

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

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

Notes:

1. Input pull-up configuration can cause an unexpected value at the alternate peripheral input.
2. When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

Warning: Alternate functions of peripherals must must not be activated when the external interrupts are enabled on the same pin, in order to avoid generating spurious interrupts.

I/O PORTS (Cont'd)

Analog Alternate Functions

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

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

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

9.2.4 I/O Port Implementation

The hardware implementation on each I/O port depends on the settings in the DDR register and specific features of the I/O port such as ADC Input or true open drain.

I/O PORTS (Cont'd)

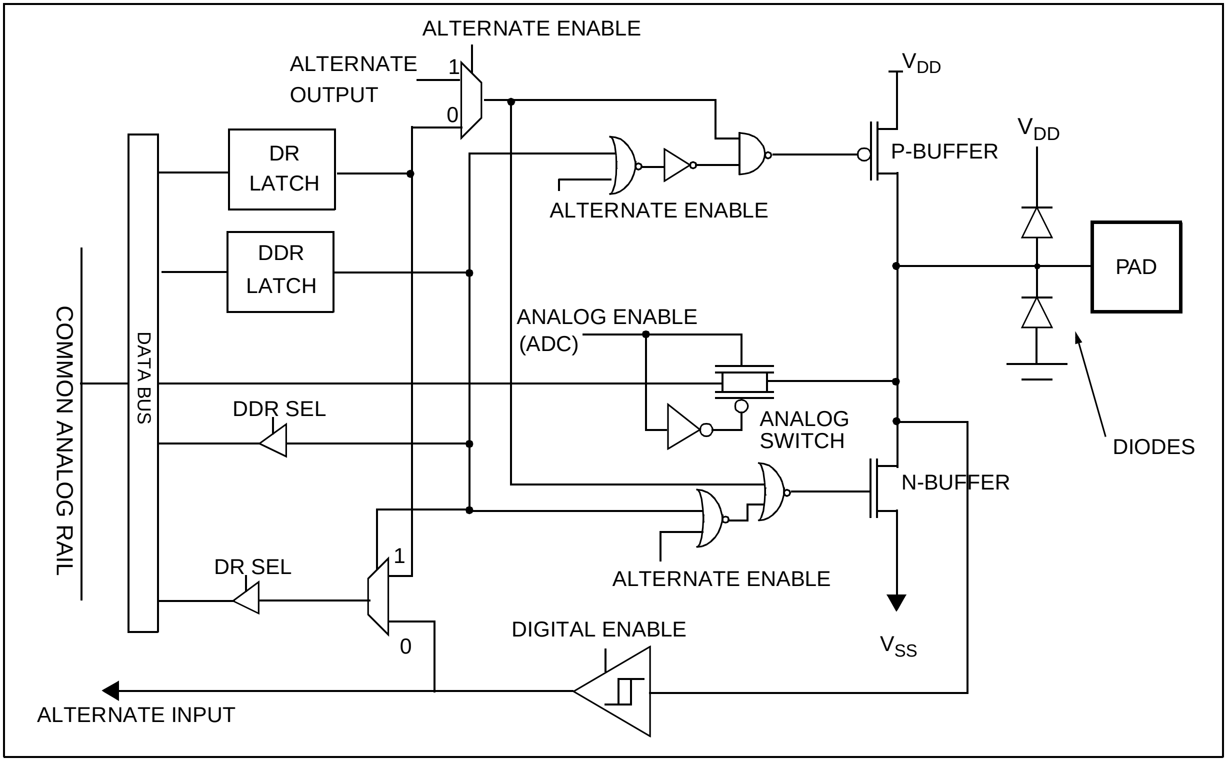
9.2.5 Port A

Table 7. Port A Description

PORT A	I/O		Alternate Function	
	Input*	Output	Signal	Condition
PA0	floating	push-pull	USBOE	USBOE = 1 (MISC)
			IT1 Schmitt triggered input	IT1E = 1 (ITRFRE1)
			AIN0 (ADC)	CS[2:0] = 000 (ADCCSR)
PA1	floating	push-pull	IT2 Schmitt triggered input	IT2E = 1 (ITRFRE1)
			AIN1 (ADC)	CS[2:0] = 001 (ADCCSR)
PA2	floating	push-pull	IT3 Schmitt triggered input	IT3E = 1 (ITRFRE1)
			AIN2 (ADC)	CS[2:0] = 010 (ADCCSR)
PA3	floating	push-pull	IT4 Schmitt triggered input	IT4E = 1 (ITRFRE1)
			AIN3 (ADC)	CS[2:0] = 011 (ADCCSR)
PA4	floating	push-pull	AIN4 (ADC)	CS[2:0] = 100 (ADCCSR)
PA5	floating	push-pull	AIN5 (ADC)	CS[2:0] = 101 (ADCCSR)
PA6	floating	push-pull	AIN6 (ADC)	CS[2:0] = 110 (ADCCSR)
PA7	floating	push-pull	AIN7 (ADC)	CS[2:0] = 111 (ADCCSR)

*Reset State

Figure 23. PA[7:0] Configuration


ALTERNATE ENABLE
ALTERNATE OUTPUT
1
0
V_{DD}
DR LATCH
P-BUFFER
V_{DD}
ALTERNATE ENABLE
DDR LATCH
PAD
ANALOG ENABLE (ADC)
COMMON ANALOG RAIL
DATA BUS
DDR SEL
ANALOG SWITCH
DIODES
N-BUFFER
1
DR SEL
ALTERNATE ENABLE
0
DIGITAL ENABLE
V_{SS}
ALTERNATE INPUT

I/O PORTS (Cont'd)

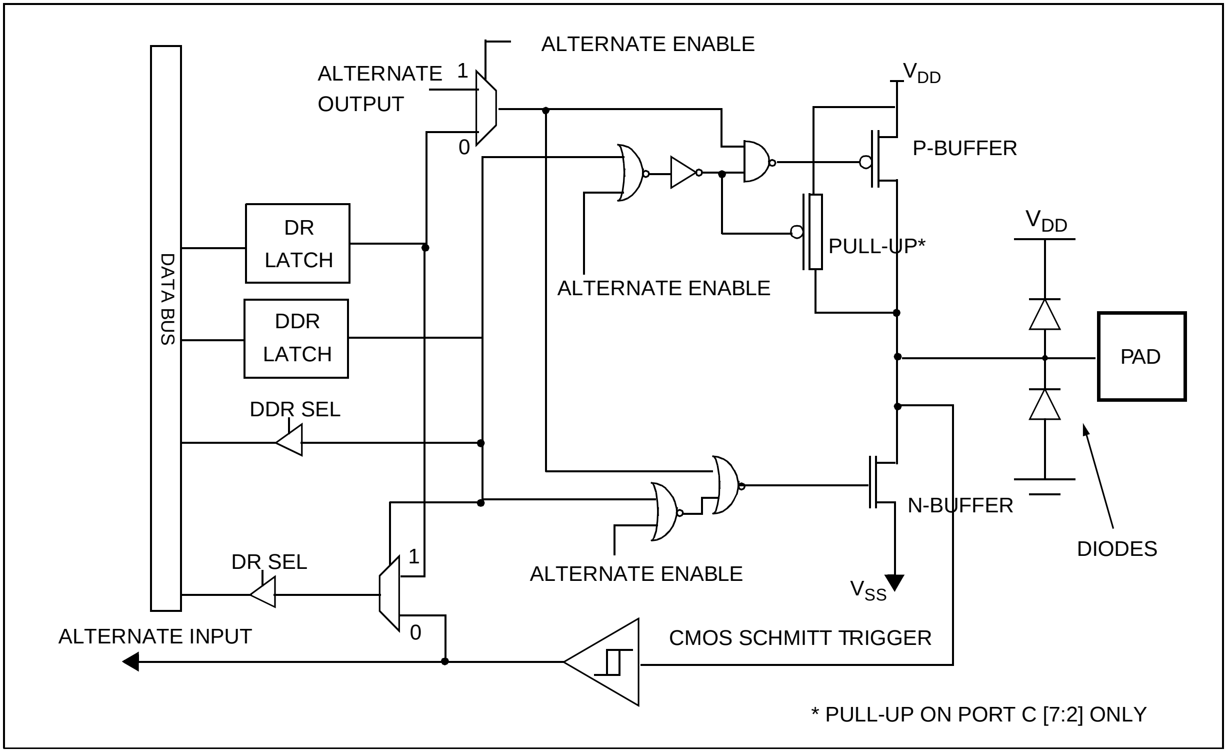
9.2.6 Port B

Table 8. Port B Description

PORT B	I/O		Alternate Function	
	Input*	**Output**	**Signal**	**Condition**
PB0	floating	push-pull (high sink)	MCO (Main Clock Output)	MCO = 1 (MISCR)
PB1	floating	push-pull (high sink)	RDI	SCI enabled
PB2	floating	push-pull (high sink)	TDO	TE = 1 (SCICR2)
PB3	floating	push-pull (high sink)	ARTCLK	EXCL = 1 (ARTCSR)
PB4	floating	push-pull (high sink)	ARTIC1	ART Timer enabled
			IT5 Schmitt triggered input	IT5E = 1 (ITRFRE1)
PB5	floating	push-pull (high sink)	ARTIC2	ART Timer enabled
			IT6 Schmitt triggered input	IT6E = 1 (ITRFRE1)
PB6	floating	push-pull (high sink)	PWM1	OE0 = 1 (PWMCR)
			IT7 Schmitt triggered input	IT7E = 1 (ITRFRE1)
PB7	floating	push-pull (high sink)	PWM2	OE1 = 1 (PWMCR)
			IT8 Schmitt triggered input	IT8E = 1 (ITRFRE1)

*Reset State

Figure 24. Port B and Port C [7:2] Configuration


ALTERNATE ENABLE
ALTERNATE OUTPUT
1
0
V_{DD}
P-BUFFER
DR LATCH
DDR LATCH
DATA BUS
PULL-UP*
ALTERNATE ENABLE
V_{DD}
PAD
DDR SEL
N-BUFFER
DIODES
DR SEL
1
ALTERNATE ENABLE
V_{SS}
ALTERNATE INPUT
0
CMOS SCHMITT TRIGGER
* PULL-UP ON PORT C [7:2] ONLY

I/O PORTS (Cont'd)

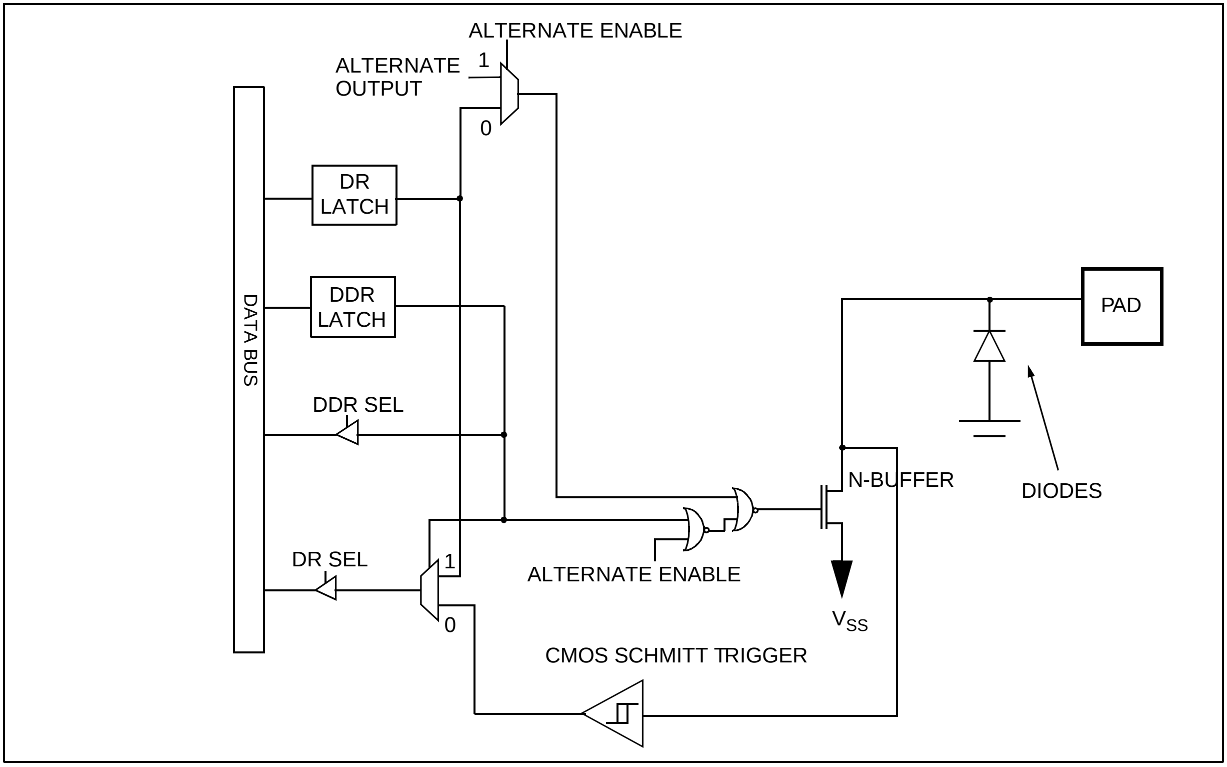
9.2.7 Port C

Table 9. Port C Description

PORT C	I/O		Alternate Function	
	Input*	**Output**	**Signal**	**Condition**
PC0	floating	true open drain		
PC1	floating	true open drain		
PC2	with pull-up	push-pull	IT9 Schmitt triggered input	IT9E = 1 (ITRFRE2)
PC3	with pull-up	push-pull	SCK	SPI enabled
			IT10 Schmitt triggered input	IT10E = 1 (ITRFRE2)
PC4	with pull-up	push-pull	$\overline{SS}$	SPI enabled
			IT11 Schmitt triggered input	IT11E = 1 (ITRFRE2)
PC5	with pull-up	push-pull	MISO	SPI enabled
			IT12 Schmitt triggered input	IT12E = 1 (ITRFRE2)
PC6	with pull-up	push-pull	MOSI	SPI enabled
PC7	with pull-up	push-pull		

*Reset State

Figure 25. Port C[1:0] Configuration


ALTERNATE ENABLE
ALTERNATE OUTPUT
1
0
DR LATCH
DDR LATCH
DATA BUS
DDR SEL
DR SEL
1
0
ALTERNATE ENABLE
N-BUFFER
V_{SS}
CMOS SCHMITT TRIGGER
PAD
DIODES

I/O PORTS (Cont'd)

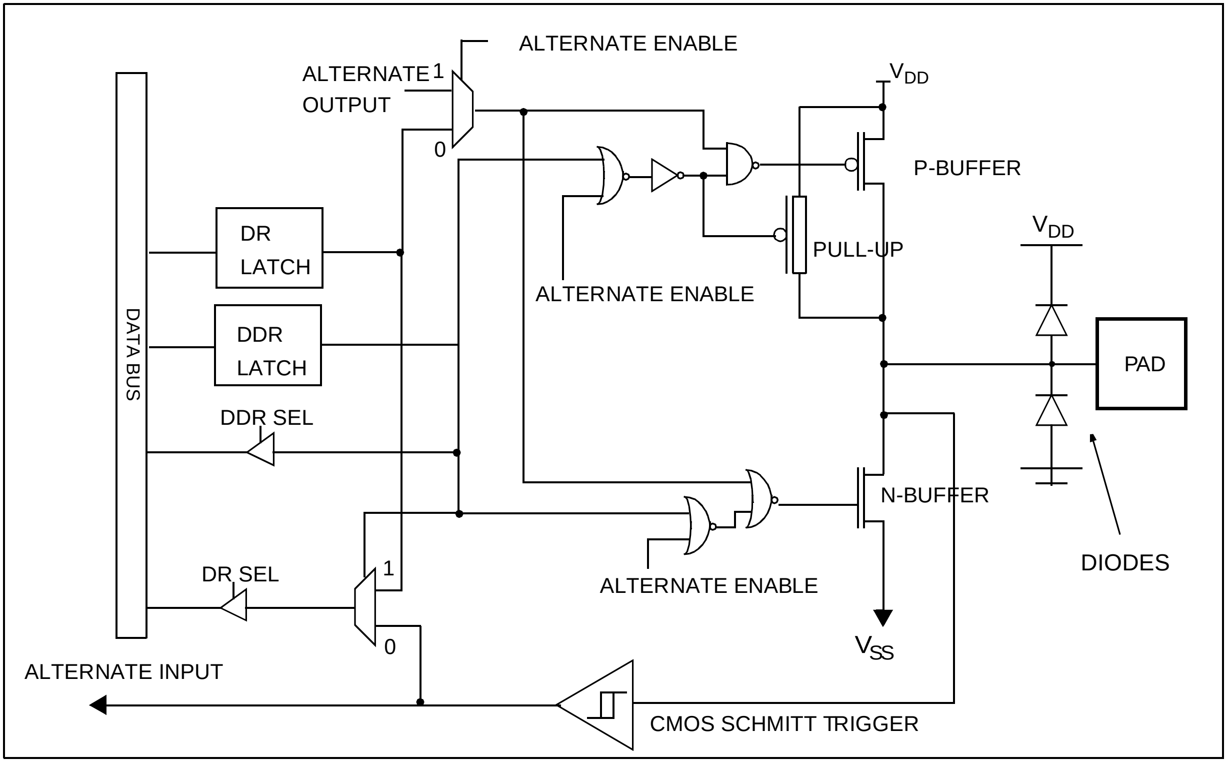
9.2.8 Port D

Table 10. Port D Description

PORT D	I/O		Alternate Function	
	Input*	**Output**	**Signal**	**Condition**
PD0	with pull-up	push-pull		
PD1	with pull-up	push-pull		
PD2	with pull-up	push-pull		
PD3	with pull-up	push-pull		
PD4	with pull-up	push-pull		
PD5	with pull-up	push-pull		
PD6	with pull-up	push-pull		

*Reset State

Figure 26. Port D Configuration


ALTERNATE ENABLE
ALTERNATE OUTPUT
1
0
V_DD
P-BUFFER
DR LATCH
PULL-UP
ALTERNATE ENABLE
V_DD
DATA BUS
DDR LATCH
PAD
DDR SEL
N-BUFFER
DR SEL
1
ALTERNATE ENABLE
DIODES
0
V_SS
ALTERNATE INPUT
CMOS SCHMITT TRIGGER

I/O PORTS (Cont'd)

9.2.9 Register Description

DATA REGISTER (DR)

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

Read/Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bits 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 or D.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

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

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

0: Input mode
1: Output mode

I/O PORTS (Cont'd)

Table 11. I/O Port Register Map and Reset Values

Address (Hex.)	**Register Label**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset Value of all I/O port registers		0	0	0	0	0	0	0	0
0000h	**PADR**	MSB							LSB
0001h	**PADDR**								
0002h	**PBDR**	MSB							LSB
0003h	**PBDDR**								
0004h	**PCDR**	MSB							LSB
0005h	**PCDDR**								
0006h	**PDDR**	MSB							LSB
0007h	**PDDDR**								

9.3 MISCELLANEOUS REGISTER

MISCELLANEOUS REGISTER
Read Write
Reset Value - 0000 0000 (00h)

7							0
-	-	-	-	SMS1	SMS0	US-BOE	MCO

Bits 7:4 = Reserved

Bits 3:2 = **SMS[1:0]** *Slow Mode Selection*
These bits select the Slow Mode frequency (depending on the oscillator frequency configured by option byte).

OSC12/6	**SMS1**	**SMS0**	**Slow Mode Frequency (MHz.)**
f_{OSC}= 6 MHz.	0	0	4
	0	1	2
	1	0	1
	1	1	0.5
f_{OSC}= 12 MHz.	0	0	8
	0	1	4
	1	0	2
	1	1	1

Bit 1 = **USBOE** *USB Output Enable*
0: PA0 port free for general purpose I/O
1: USBOE alternate function enabled. The USB output enable signal is output on the PA0 port (at “1” when the ST7 USB is transmitting data).

Bit 0 = **MCO** *Main Clock Out*
0: PB0 port free for general purpose I/O
1: MCO alternate function enabled (f_{CPU} output on PB0 I/O port)

10 ON-CHIP PERIPHERALS

10.1 WATCHDOG TIMER (WDG)

10.1.1 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.

10.1.2 Main Features

- Programmable timer (64 increments of 65536 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Hardware Watchdog selectable by option byte

10.1.3 Functional Description

The counter value stored in the CR register (bits T[6:0]), is decremented every 65,536 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.

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):

– The WDGA bit is set (watchdog enabled)

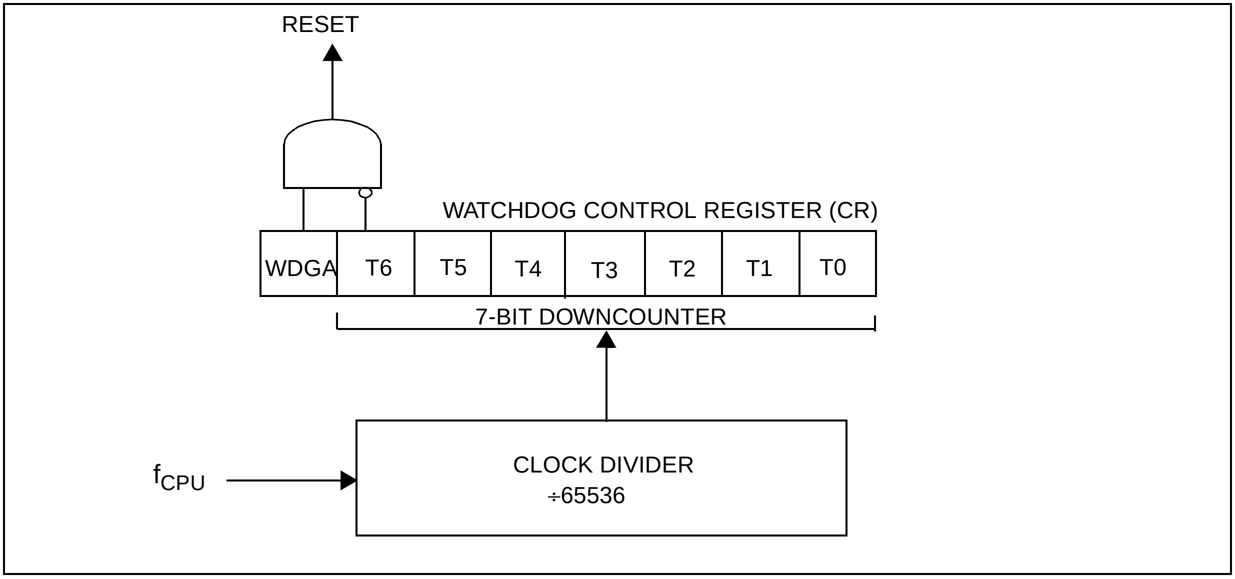
– The T6 bit is set to prevent generating an immediate reset

– 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	524.288
Min	C0h	8.192

Figure 27. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

10.1.4 Software Watchdog Option

If Software Watchdog is selected by option byte, the watchdog is disabled following a reset. 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).

10.1.5 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.

10.1.6 Low Power Modes

WAIT Instruction

No effect on Watchdog.

HALT Instruction

Halt mode can be used when the watchdog is enabled. When the oscillator is stopped, the WDG stops counting and is no longer able to generate a reset until the microcontroller receives an external interrupt or a reset.

If an external interrupt is received, the WDG restarts counting after 514 CPU clocks. In the case of the Software Watchdog option, if a reset is generated, the WDG is disabled (reset state).

Recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.
- When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as Input before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in ROM with the value 0x8E.
- As the HALT instruction clears the I bit in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

10.1.7 Interrupts

None.

10.1.8 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.

Bits 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).

Table 13. Watchdog Timer Register Map and Reset Values

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

10.2 PWM AUTO-RELOAD TIMER (ART)

10.2.1 Introduction

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

These resources allow five possible operating modes:

– Generation of up to 4 independent PWM signals

– Output compare and Time base interrupt

– Up to two input capture functions

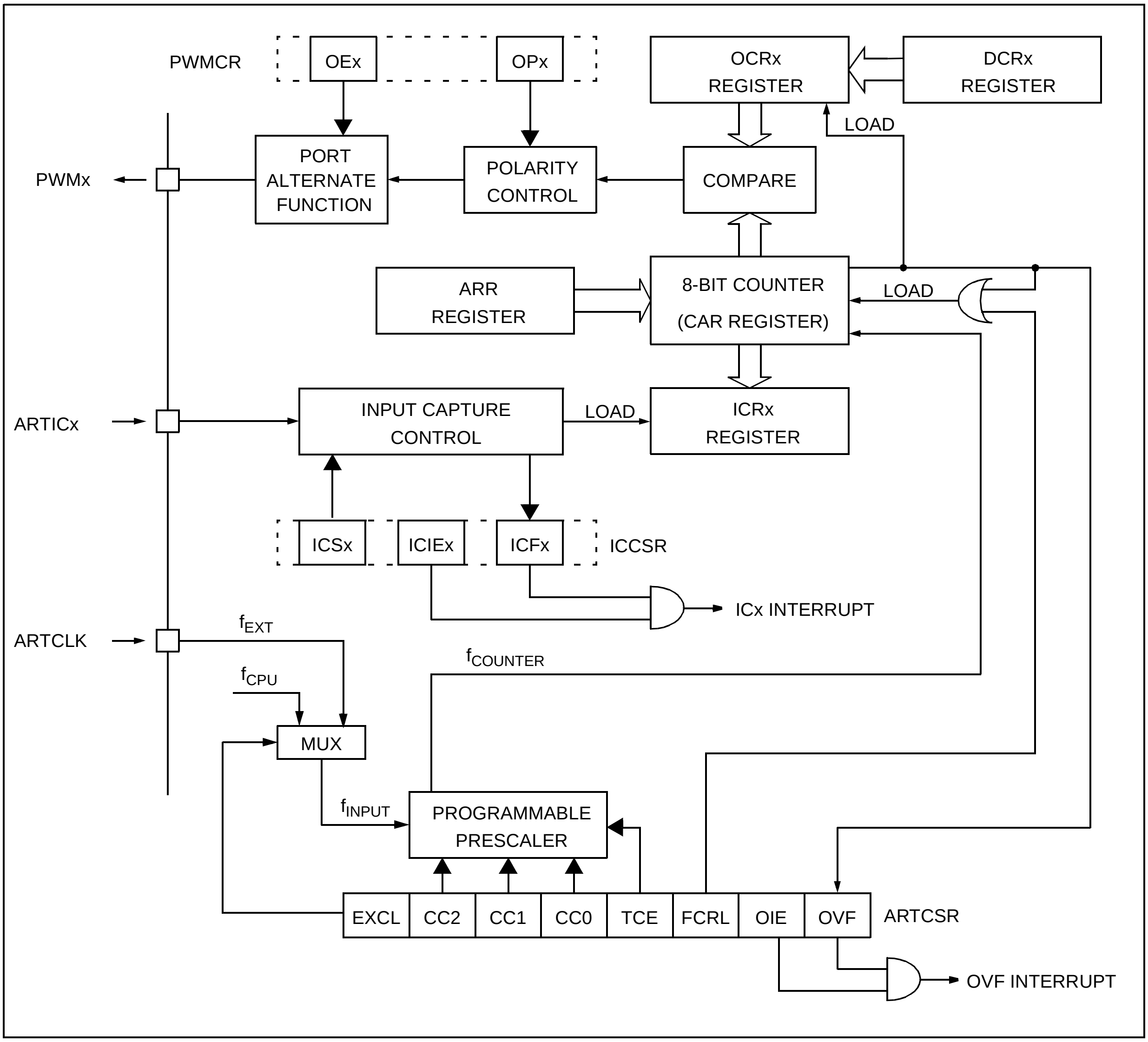
– External event detector

– Up to two external interrupt sources

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

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

Figure 28. PWM Auto-Reload Timer Block Diagram


PWMCR
OEx
OPx
OCRx REGISTER
DCRx REGISTER
LOAD
PWMx
PORT ALTERNATE FUNCTION
POLARITY CONTROL
COMPARE
ARR REGISTER
8-BIT COUNTER (CAR REGISTER)
LOAD
ARTICx
INPUT CAPTURE CONTROL
LOAD
ICRx REGISTER
ICSx
ICIEx
ICFx
ICCSR
ICx INTERRUPT
ARTCLK
f_{EXT}
f_{CPU}
$f_{COUNTER}$
MUX
f_{INPUT}
PROGRAMMABLE PRESCALER
EXCL
CC2
CC1
CC0
TCE
FCRL
OIE
OVF
ARTCSR
OVF INTERRUPT

PWM AUTO-RELOAD TIMER (Cont'd)

10.2.2 Functional Description

Counter

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

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

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

Counter clock and prescaler

The counter clock frequency is given by:

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

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

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

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

Counter and Prescaler Initialization

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

The counter can be initialized by:

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

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

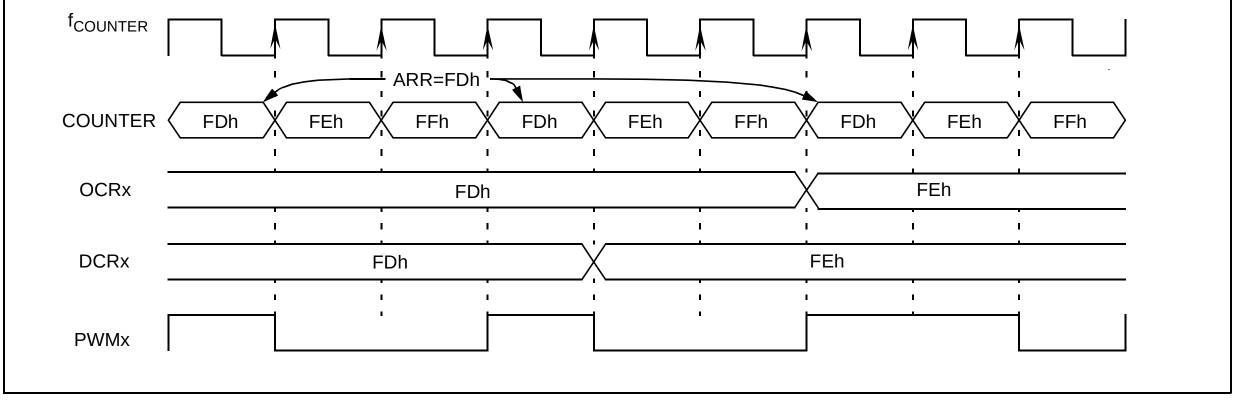
Direct access to the prescaler is not possible.

Output compare control

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

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

Figure 29. Output compare control



PWM AUTO-RELOAD TIMER (Cont'd)

Independent PWM signal generation

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

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

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

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

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

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

The maximum available resolution for the PWMx duty cycle is:

$$\text{Resolution} = 1 / (256 - ARR)$$

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

Figure 30. PWM Auto-reload Timer Function

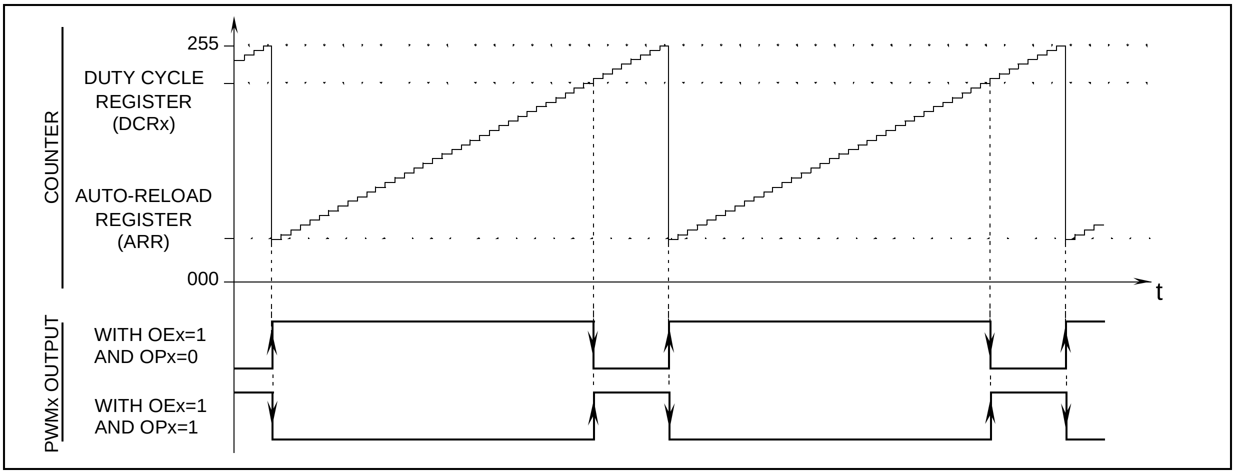
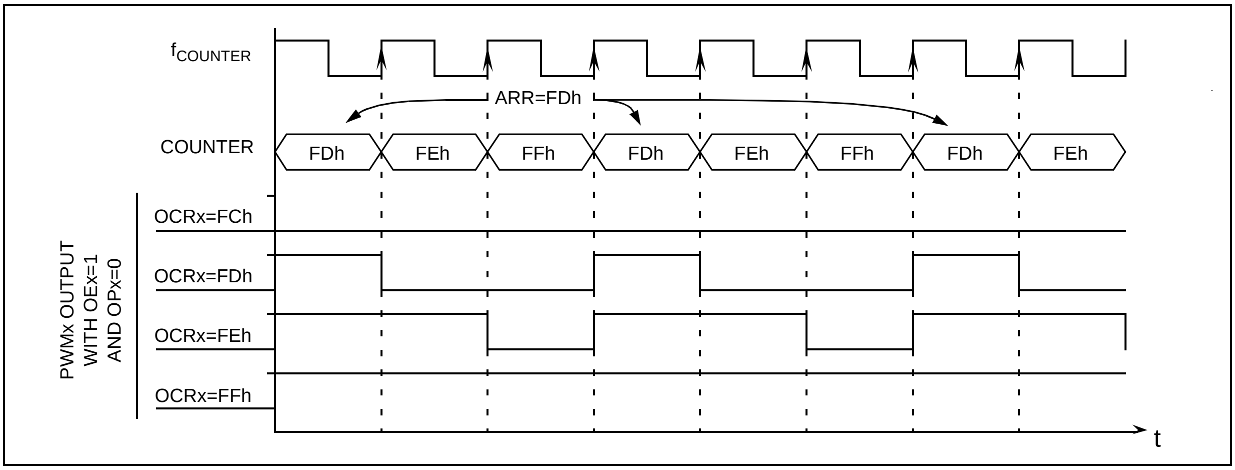

COUNTER
255
DUTY CYCLE REGISTER (DCRx)
AUTO-RELOAD REGISTER (ARR)
000
t
PWMx OUTPUT
WITH OEx=1 AND OPx=0
WITH OEx=1 AND OPx=1

Figure 31. PWM Signal from 0% to 100% Duty Cycle


fCOUNTER
ARR=FDh
COUNTER
FDh
FEh
FFh
FDh
FEh
FFh
FDh
FEh
OCRx=FCh
OCRx=FDh
OCRx=FEh
OCRx=FFh
PWMx OUTPUT WITH OEx=1 AND OPx=0
t

PWM AUTO-RELOAD TIMER (Cont'd)

Output compare and Time base interrupt

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

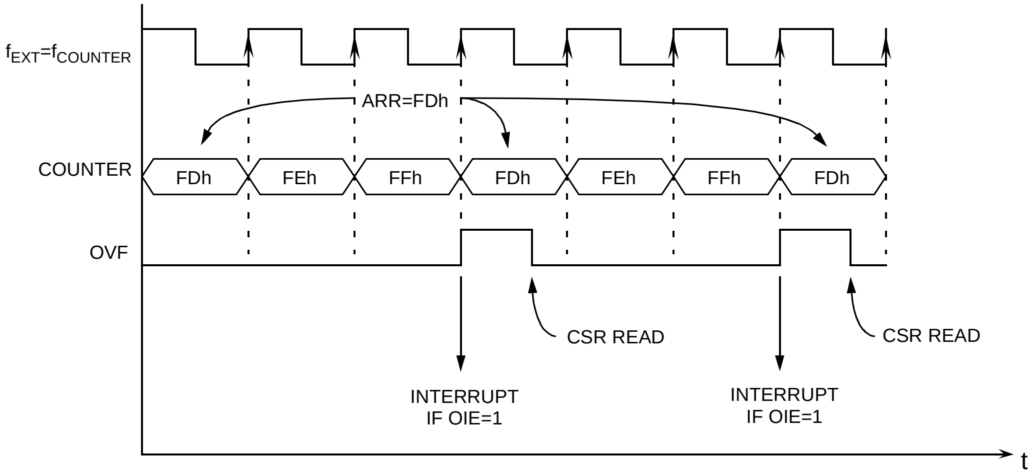
External clock and event detector mode

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

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

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

Figure 32. External Event Detector Example (3 counts)



fEXT=fCOUNTER
ARR=FDh
COUNTER
FDh
FEh
FFh
FDh
FEh
FFh
FDh
OVF
CSR READ
CSR READ
INTERRUPT
IF OIE=1
INTERRUPT
IF OIE=1
t

PWM AUTO-RELOAD TIMER (Cont'd)

Input capture function

This mode allows the measurement of external signal pulse widths through ICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ICCSR).

These input capture interrupts are enabled through the CIEx bits of the ICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ICCSR register.

The read only input capture registers (ICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ICRx register is inhibited until it is read (clearing the CFx bit).
The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means, the ICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time ($1/f_{COUNTER}$).

During HALT mode, input capture is inhibited (the ICRx is never re-loaded) and only the external interrupt capability can be used.

External interrupt capability

This mode allows the Input capture capabilities to be used as external interrupt sources.

The edge sensitivity of the external interrupts is programmable (CSx bit of ICCSR register) and they are independently enabled through CIEx bits of the ICCSR register. After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

The interrupts are synchronized on the counter clock rising edge (Figure 33).

During HALT mode, the external interrupts can still be used to wake up the micro (if CIEx bit is set).

Figure 33. ART External Interrupt

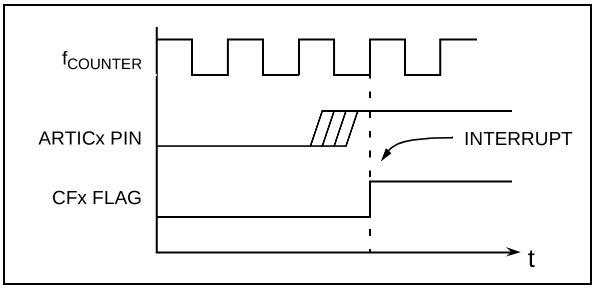
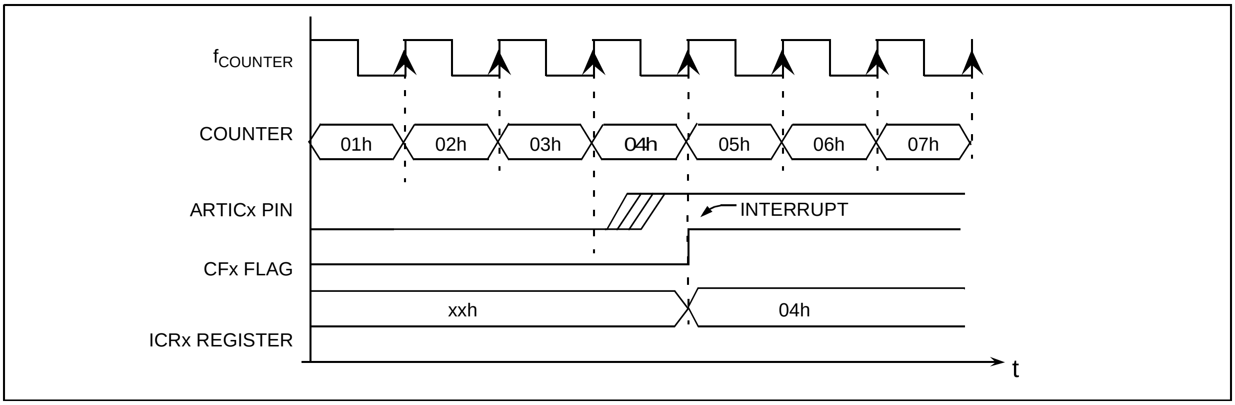


Figure 34. Input Capture Timing Diagram



PWM AUTO-RELOAD TIMER (Cont'd)

10.2.3 Register Description

CONTROL / STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
EXCL	CC2	CC1	CC0	TCE	FCRL	OIE	OVF

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

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

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

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

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

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

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

COUNTER ACCESS REGISTER (CAR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

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

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

AUTO-RELOAD REGISTER (ARR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

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

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

This register has two PWM management functions:

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

PWM Frequency vs. Resolution:

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

PWM AUTO-RELOAD TIMER (Cont'd)

PWM CONTROL REGISTER (PWMCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	OE1	OE0	0	0	OP1	OP0

Bit 7:6 = Reserved.

Bit 5:4 = **OE[1:0]** *PWM Output Enable*
These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin.
0: PWM output disabled.
1: PWM output enabled.

Bit 3:2 = Reserved.

Bit 1:0 = **OP[1:0]** *PWM Output Polarity*
These bits are set and cleared by software. They independently select the polarity of the two PWM output signals.

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

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

DUTY CYCLE REGISTERS (DCRx)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

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

These bits are set and cleared by software.

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

PWM AUTO-RELOAD TIMER (Cont'd)

INPUT CAPTURE CONTROL / STATUS REGISTER (ICCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	CS2	CS1	CIE2	CIE1	CF2	CF1

Bit 7:6 = Reserved, always read as 0.

Bit 5:4 = **CS[2:1]** *Capture Sensitivity*
These bits are set and cleared by software. They determine the trigger event polarity on the corresponding input capture channel.
0: Falling edge triggers capture on channel x.
1: Rising edge triggers capture on channel x.

Bit 3:2 = **CIE[2:1]** *Capture Interrupt Enable*
These bits are set and cleared by software. They enable or disable the Input capture channel interrupts independently.
0: Input capture channel x interrupt disabled.
1: Input capture channel x interrupt enabled.

Bit 1:0 = **CF[2:1]** *Capture Flag*
These bits are set by hardware and cleared by software reading the corresponding ICRx register. Each CFx bit indicates that an input capture x has occurred.
0: No input capture on channel x.
1: An input capture has occured on channel x.

INPUT CAPTURE REGISTERS (ICRx)

Read only

Reset Value: 0000 0000 (00h)

7							0
IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

Bit 7:0 = **IC[7:0]** *Input Capture Data*

These read only bits are set and cleared by hardware. An ICRx register contains the 8-bit auto-reload counter value transferred by the input capture channel x event.

PWM AUTO-RELOAD TIMER (Cont'd)

Table 14. PWM Auto-Reload Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0014h	**PWMDCR1** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0015h	**PWMDCR0** Reset Value	DC7 0	DC6 0	DC5 0	DC4 0	DC3 0	DC2 0	DC1 0	DC0 0
0016h	**PWMCR** Reset Value	0 0	0 0	OE1 0	OE0 0	0 0	0 0	OP1 0	OP0 0
0017h	**ARTCSR** Reset Value	EXCL 0	CC2 0	CC1 0	CC0 0	TCE 0	FCRL 0	OIE 0	OVF 0
0018h	**ARTCAR** Reset Value	CA7 0	CA6 0	CA5 0	CA4 0	CA3 0	CA2 0	CA1 0	CA0 0
0019h	**ARTARR** Reset Value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0
001Ah	**ARTICCSR** Reset Value	0	0	CE2 0	CE1 0	CS2 0	CS1 0	CF2 0	CF1 0
001Bh	**ARTICR1** Reset Value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0
001Ch	**ARTICR2** Reset Value	IC7 0	IC6 0	IC5 0	IC4 0	IC3 0	IC2 0	IC1 0	IC0 0

10.3 TIMEBASE UNIT (TBU)

10.3.1 Introduction

The Timebase unit (TBU) can be used to generate periodic interrupts.

10.3.2 Main Features

- 8-bit upcounter
- Programmable prescaler
- Period between interrupts: max. 8.1ms (at 8 MHz f_{CPU})
- Maskable interrupt
- Cascadable with PWM/ART TImer

10.3.3 Functional Description

The TBU operates as a free-running upcounter.

When the TCEN bit in the TBUCSR register is set by software, counting starts at the current value of the TBUCV register. The TBUCV register is incremented at the clock rate output from the prescaler selected by programming the PR[2:0] bits in the TBUCSR register.

When the counter rolls over from FFh to 00h, the OVF bit is set and an interrupt request is generated if ITE is set.

The user can write a value at any time in the TBUCV register.

If the cascading option is selected (CAS bit=1 in the TBUCSR register), the TBU and the the ART TImer counters act together as a 16-bit counter. In this case, the TBUCV register is the high order byte, the ART counter (ARTCAR register) is the low order byte. Counting is clocked by the ART timer clock (Refer to the description of the ART Timer ARTCSR register).

10.3.4 Programming Example

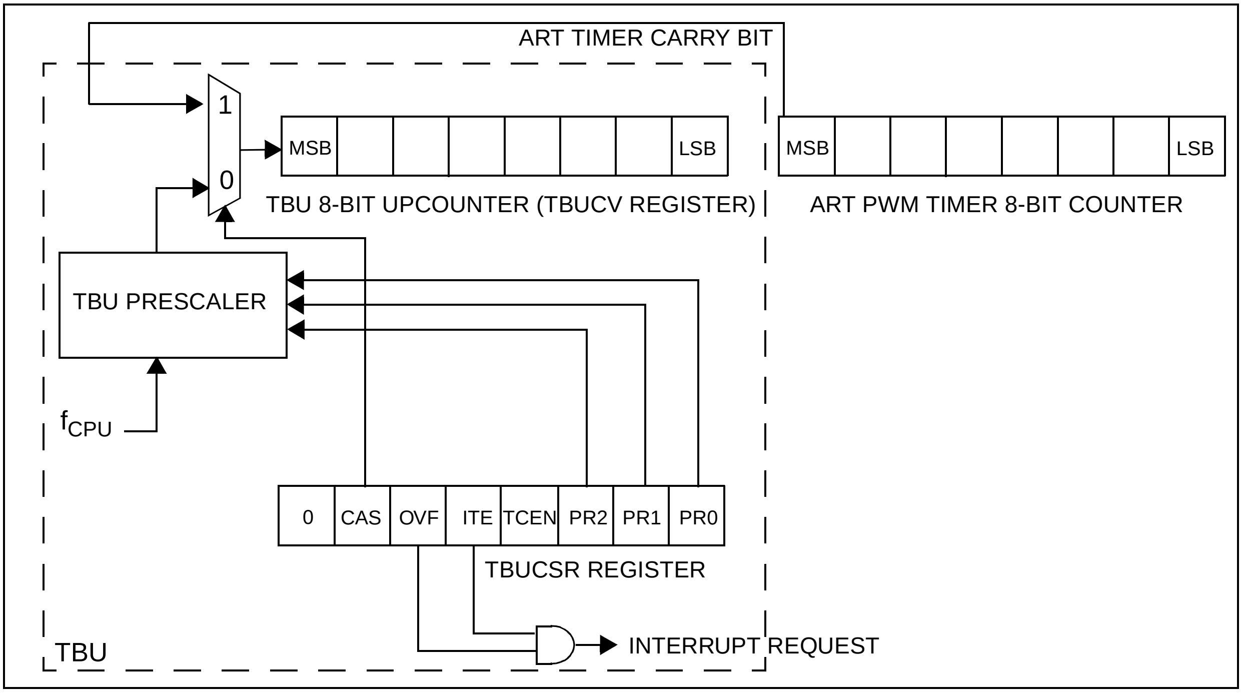
In this example, timer is required to generate an interrupt after a delay of 1 ms.

Assuming that f_{CPU} is 8 MHz and a prescaler division factor of 256 will be programmed using the PR[2:0] bits in the TBUCSR register, 1 ms = 32 TBU timer ticks.

In this case, the initial value to be loaded in the TBUCV must be (256-32) = 224 (E0h).

```
ld A, E0h
ld TBUCV, A   ; Initialize counter value
ld A 1Fh      ;
ld TBUCSR, A  ; Prescaler factor = 256,
              ; interrupt enable,
              ; TBU enable
```

Figure 35. TBU Block Diagram


ART TIMER CARRY BIT
1
0
MSB
LSB
TBU 8-BIT UPCOUNTER (TBUCV REGISTER)
MSB
LSB
ART PWM TIMER 8-BIT COUNTER
TBU PRESCALER
f_{CPU}
0
CAS
OVF
ITE
TCEN
PR2
PR1
PR0
TBUCSR REGISTER
INTERRUPT REQUEST
TBU

TIMEBASE UNIT (Cont'd)

10.3.5 Low Power Modes

Mode	Description
WAIT	No effect on TBU
HALT	TBU halted.

10.3.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Counter Overflow Event	OVF	ITE	Yes	No

Note: The OVF interrupt event is connected to an interrupt vector (see Interrupts chapter).
It generates an interrupt if the ITE bit is set in the TBUCSR register and the I-bit in the CC register is reset (RIM instruction).

10.3.7 Register Description

TBU COUNTER VALUE REGISTER (TBUCV)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
CV7	CV6	CV5	CV4	CV3	CV2	CV1	CV0

Bit 7:0 = **CV[7:0]** *Counter Value*

This register contains the 8-bit counter value which can be read and written anytime by software. It is continuously incremented by hardware if TCEN=1.

TBU CONTROL/STATUS REGISTER (TBUCSR)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
0	CAS	OVF	ITE	TCEN	PR2	PR1	PR0

Bit 7 = Reserved. Forced by hardware to 0.

Bit 6 = **CAS** *Cascading Enable*
This bit is set and cleared by software. It is used to cascade the TBU and the PWM/ART timers.
0: Cascading disabled
1: Cascading enabled

Bit 5 = **OVF** *Overflow Flag*

This bit is set only by hardware, when the counter value rolls over from FFh to 00h. It is cleared by software reading the TBUCSR register. Writing to this bit does not change the bit value.
0: No overflow
1: Counter overflow

Bit 4 = **ITE** *Interrupt enabled.*

This bit is set and cleared by software.
0: Overflow interrupt disabled
1: Overflow interrupt enabled. An interrupt request is generated when OVF=1.

Bit 3 = **TCEN** *TBU Enable.*

This bit is set and cleared by software.
0: TBU counter is frozen and the prescaler is reset.
1: TBU counter and prescaler running.

Bit 2:0 = **PR[2:0]** *Prescaler Selection*

These bits are set and cleared by software to select the prescaling factor.

PR2	PR1	PR0	Prescaler Division Factor
0	0	0	2
0	0	1	4
0	1	1	8
1	0	0	16
1	0	1	32
1	0	1	64
1	1	0	128
1	1	1	256

TIMEBASE UNIT (Cont'd)

Table 15. TBU Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0036h	**TBUCV** Reset Value	CV7 0	CV6 0	CV5 0	CV4 0	CV3 0	CV2 0	CV1 0	CV0 0
0037h	**TBUSR** Reset Value	- 0	CAS 0	OVF 0	ITE 0	TCEN 0	PR2 0	PR1 0	PR0 0

10.4 SERIAL PERIPHERAL INTERFACE (SPI)

10.4.1 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.

10.4.2 Main Features

- Full duplex, three-wire synchronous transfers
- Simplex, two wire synchronous transfers
- Master or slave operation selectable either through software or hardware
- Six master mode frequencies
- Maximum slave mode frequency = $f_{CPU}/2$.
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability.
- Overrun error detection flag.

10.4.3 General Description

The SPI is connected to external devices through 4(3) alternate pins:

– MISO: Master In / Slave Out pin

– MOSI: Master Out / Slave In pin

– SCK: Serial Clock pin

– $\overline{\text{SS}}$: Slave select pin (if not done through software)

A basic example of interconnections between a single master and a single slave is illustrated on Figure 36.

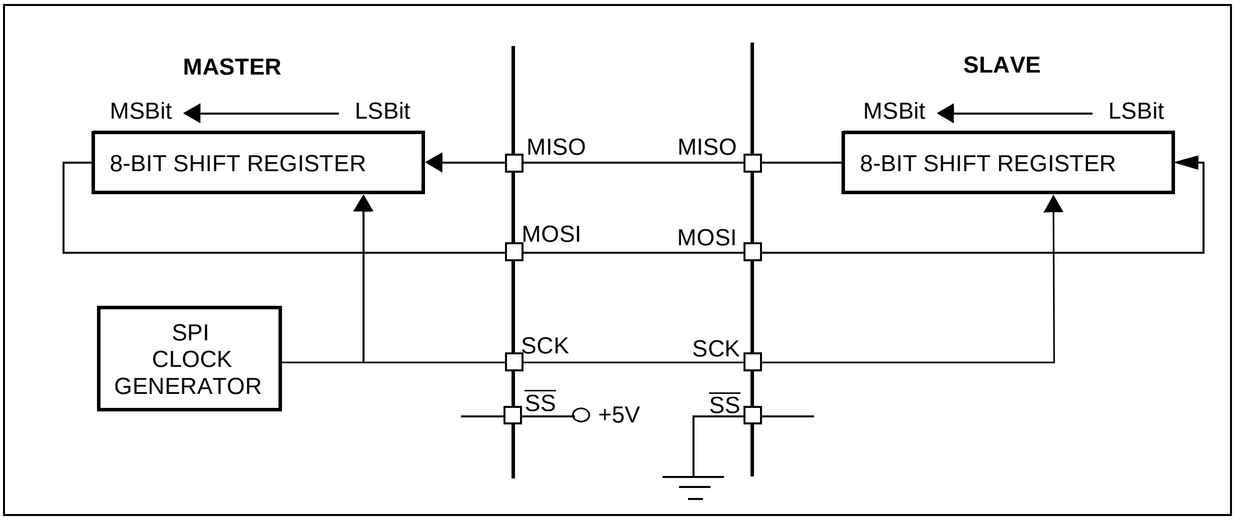
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

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.

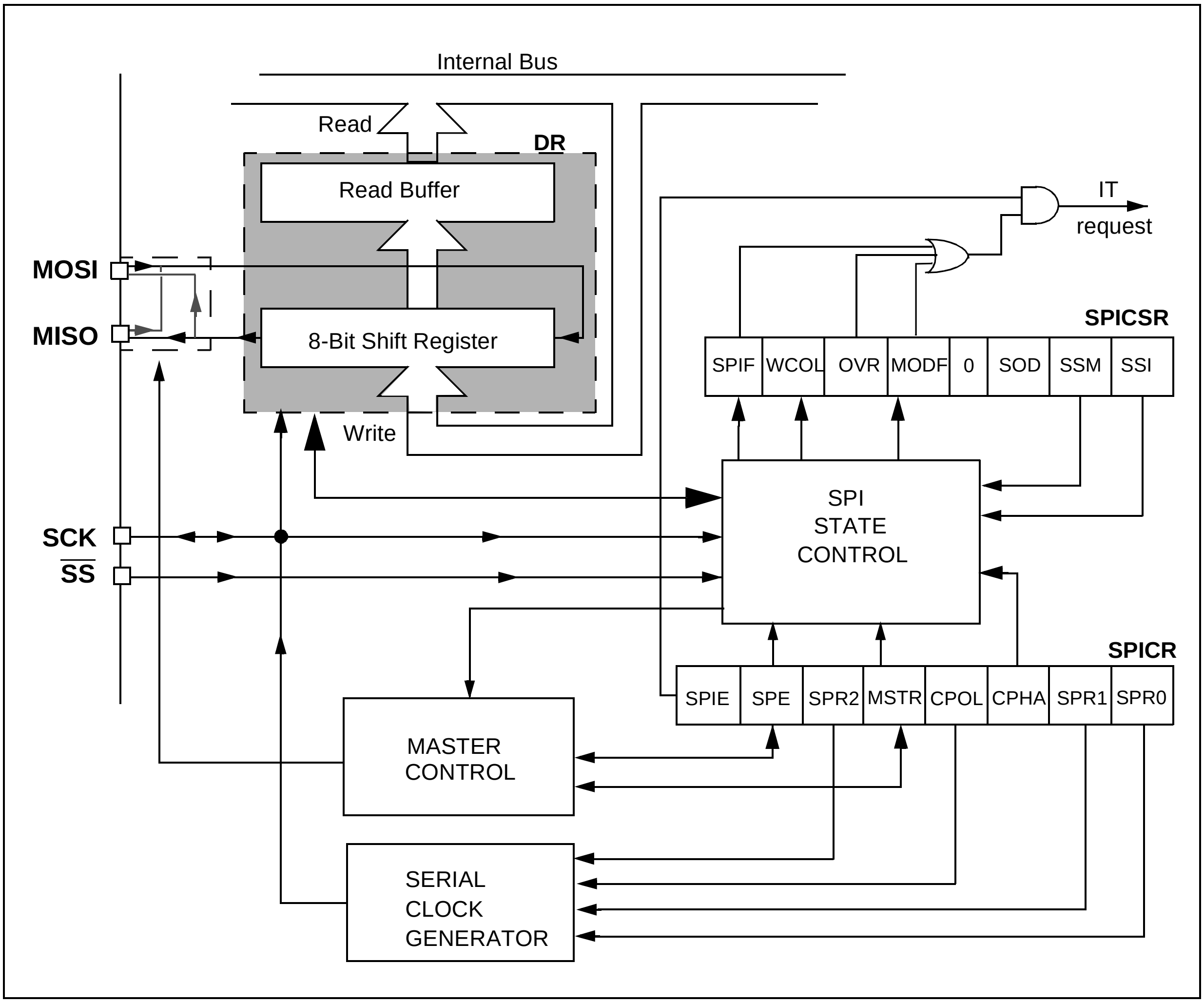
Four possible data/clock timing relationships may be chosen (see Figure 39) but master and slave must be programmed with the same timing mode.

Figure 36. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 37. Serial Peripheral Interface Block Diagram


Internal Bus
Read
DR
Read Buffer
8-Bit Shift Register
Write
MOSI
MISO
SCK
SS
IT
request
SPICSR
SPIF
WCOL
OVR
MODF
0
SOD
SSM
SSI
SPI
STATE
CONTROL
SPICR
SPIE
SPE
SPR2
MSTR
CPOL
CPHA
SPR1
SPR0
MASTER
CONTROL
SERIAL
CLOCK
GENERATOR

SERIAL PERIPHERAL INTERFACE (Cont’d)

10.4.4 Functional Description

Figure 36 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

- A Control Register (SPICR)
- A Control Status Register (SPICSR)
- A Data Register (SPIDR)

Refer to the SPICR, SPICSR and SPIDR registers in Section 10.4.5for the bit definitions.

10.4.4.1 $\overline{\text{SS}}$ Signal in Hardware/Software Mode

The $\overline{\text{SS}}$ signal can obtained in two modes:

- Hardware mode (through the $\overline{\text{SS}}$ pin)
- Software mode (through the SSI bit in the SPICSR register)

The mode (hardware or software) is selected by the Slave Selection Mode (SSM) bit in the SPICSR register.

Note: In this document, wherever $\overline{\text{SS}}$ signal selection is done using the $\overline{\text{SS}}$ pin (hardware mode), this can also be done in software mode, using the SSM and SSI bits.

10.4.4.2 Master Configuration

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

Procedure

1. Select the SPR[2:0] bits to define the serial clock baud rate (see SPICR register).
2. Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 39).
3. Connect the $\overline{\text{SS}}$ pin to a high level signal during the complete byte transmit sequence or, in software mode, set the SSI bit in the SPICSR register.
4. The MSTR and SPE bits must be set (they remain set only if the $\overline{\text{SS}}$ pin is connected to a high level signal).

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

Transmit Sequence

The transmit sequence begins when a byte is written in 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 SPIDR 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 SPICSR register while the SPIF bit is set
2. A read to the SPIDR register.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.4.3 Slave Configuration

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

The value of the SPR[2:0] bits is not used for the data transfer.

Procedure

1. For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 39.
2. The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence or, in software mode, clear the SSI bit in the SPICSR register.
3. 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 SPIDR 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 SPICSR 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 SPIDR register are inhibited until the SPICSR 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 10.4.4.8).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the SPIDR register between each data byte transfer to avoid a write collision (see Section 10.4.4.6).

Slave in Halt mode

In slave configuration, the SPI is able to exit the ST7 device from HALT mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from Halt mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from Halt mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the ST7 from Halt mode only if the Slave Select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the ST7 enters Halt mode. So if Slave selection is configured as external (see Section 10.4.4.1), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters Halt mode.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.4.4 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 39, 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 38).

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 38).

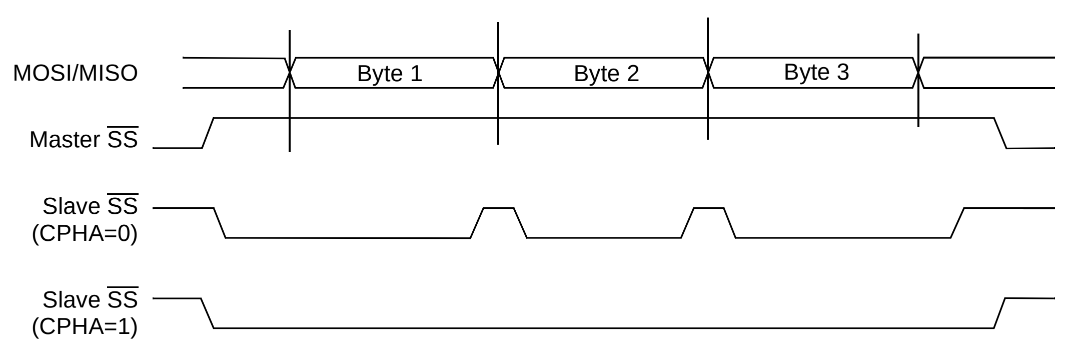
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its SPIDR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the SPIDR without producing a write collision.

Note: The SPI must be disabled by resetting the SPE bit if the CPOL bit is changed at the communication byte boundaries.

10.4.4.5 Output Disable

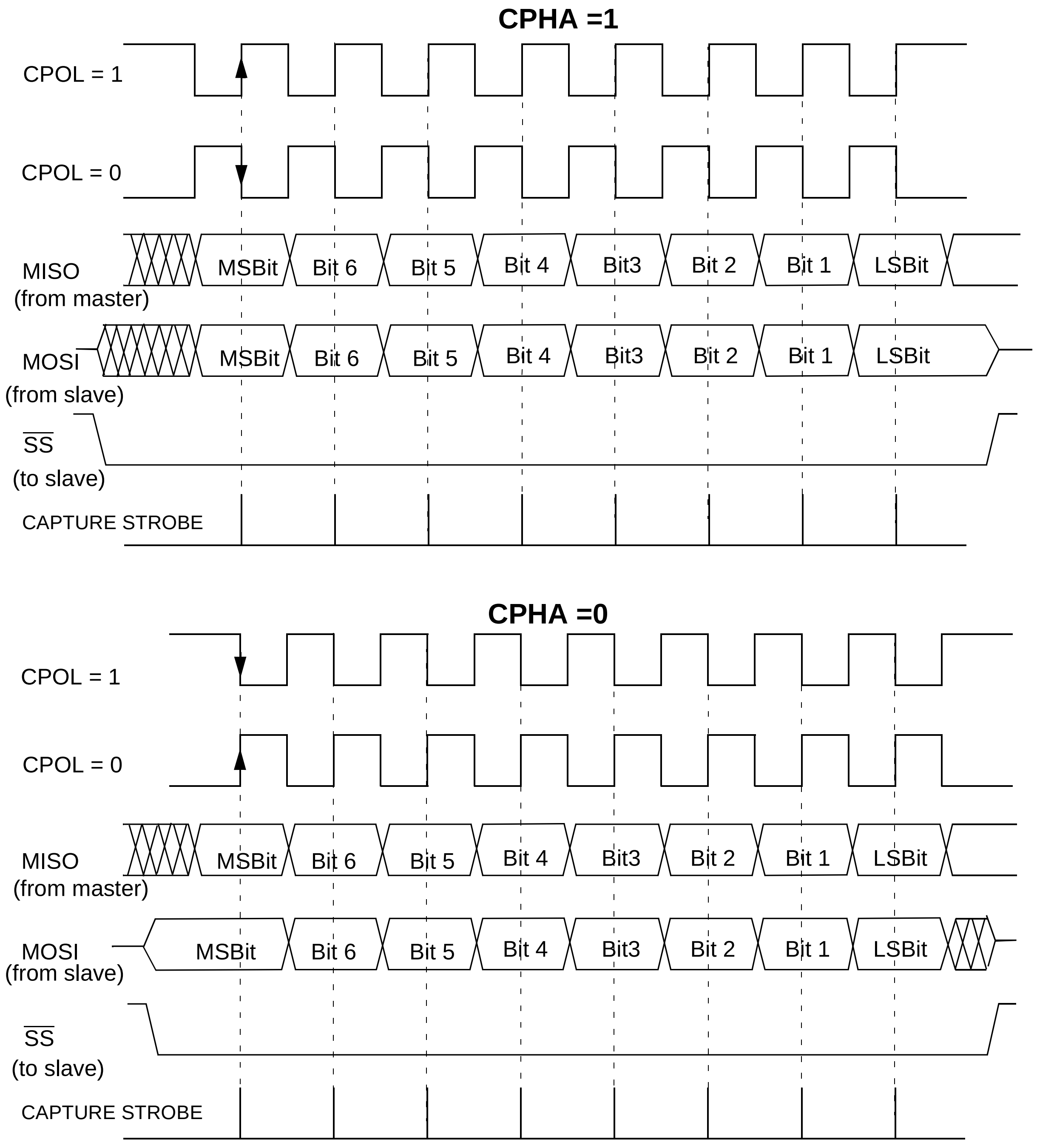
In order to free the I/O pin so it can be used for other purposes, it is possible to disable the SPI output function by setting the SOD bit in the SPICSR register.

Figure 38. CPHA / $\overline{SS}$ Timing Diagram


MOSI/MISO
Byte 1
Byte 2
Byte 3
Master $\overline{SS}$
Slave $\overline{SS}$ (CPHA=0)
Slave $\overline{SS}$ (CPHA=1)

SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 39. Data Clock Timing Diagram


CPHA =1
CPOL = 1
CPOL = 0
MISO
(from master)
MSBit
Bit 6
Bit 5
Bit 4
Bit3
Bit 2
Bit 1
LSBit
MOSI
(from slave)
MSBit
Bit 6
Bit 5
Bit 4
Bit3
Bit 2
Bit 1
LSBit
$\overline{SS}$
(to slave)
CAPTURE STROBE
CPHA =0
CPOL = 1
CPOL = 0
MISO
(from master)
MSBit
Bit 6
Bit 5
Bit 4
Bit3
Bit 2
Bit 1
LSBit
MOSI
(from slave)
MSBit
Bit 6
Bit 5
Bit 4
Bit3
Bit 2
Bit 1
LSBit
$\overline{SS}$
(to slave)
CAPTURE STROBE

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

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.4.6 Write Collision Error

A write collision occurs when the software tries to write to the SPIDR 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 SPIDR 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 SPIDR 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 SPIDR 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.

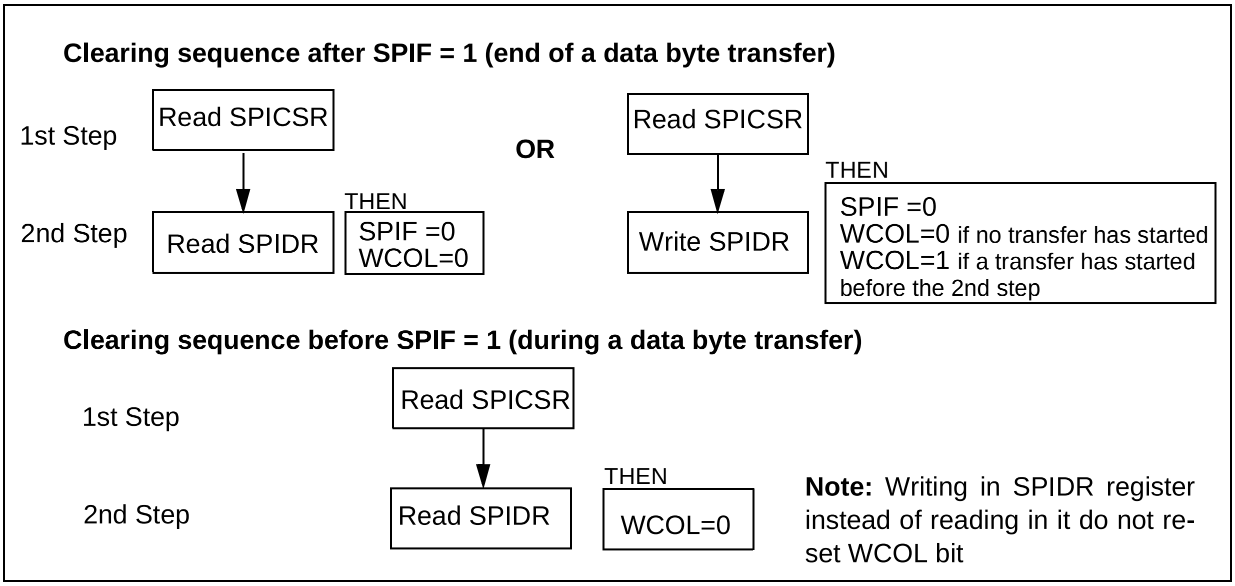
WCOL bit

The WCOL bit in the SPICSR 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 40).

Figure 40. Clearing the WCOL bit (Write Collision Flag) Software Sequence


Clearing sequence after SPIF = 1 (end of a data byte transfer)
1st Step
Read SPICSR
2nd Step
Read SPIDR
THEN
SPIF =0
WCOL=0
OR
Read SPICSR
Write SPIDR
THEN
SPIF =0
WCOL=0 if no transfer has started
WCOL=1 if a transfer has started before the 2nd step
Clearing sequence before SPIF = 1 (during a data byte transfer)
1st Step
Read SPICSR
2nd Step
Read SPIDR
THEN
WCOL=0

Note: Writing in SPIDR register instead of reading in it do not reset WCOL bit

SERIAL PERIPHERAL INTERFACE (Cont’d)

10.4.4.7 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.

10.4.4.8 Overrun Condition

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previous data byte transmitted, then the OVR bit is set and an interrupt is generated if the SPIE bit is set.

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

The OVR bit is cleared just after it has been read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.4.9 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 41).

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 active 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.

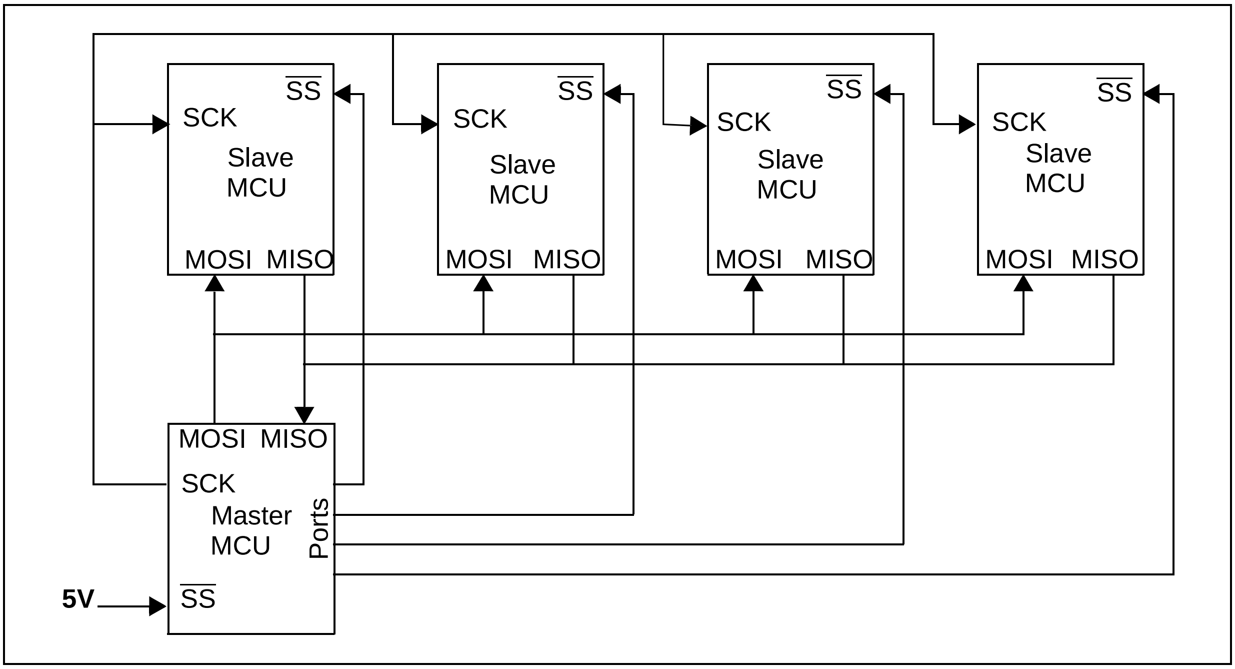
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

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 SPICR register and the MODF bit in the SPICSR register.

Figure 41. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.5 Register Description

CONTROL REGISTER (SPICR)

Read/Write

Reset Value: 0000 xxxx (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 10.4.4.7 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 Serial Peripheral Baud Rate.
0: Divider by 2 enabled
1: Divider by 2 disabled

Note: This bit has no effect in slave mode.

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 10.4.4.7 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.

Note: The SPI must be disabled by resetting SPE bit if CPOL is changed at the communication byte boundaries.

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.

Bits 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
$f_{CPU}/4$	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)

CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only)

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only).*
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 (Read only).*
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 40).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** *SPI Overrun error (Read only).*
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 10.4.4.8). An interrupt is generated if SPIE = 1 in SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only).*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 10.4.4.7 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF=1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable.*
This bit is set and cleared by software. When set it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output not disabled
1: SPI output disabled

Bit 1 = **SSM** *$\overline{SS}$ Mode Selection.*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI Slave Select pin and use the SSI bit value instead.
0: Hardware mode ($\overline{SS}$ pin is used for Master/ Slave Selection).
1: Software mode (SSI bit is used for Master/Slave Selection). $\overline{SS}$ is free to be used as a standard I/O pin

Note: Master mode selection also depends on the value of the MSTR bit in the SPICR register

Bit 0 = **SSI** *$\overline{SS}$ Internal Mode.*
This bit is set and cleared by software. It replaces pin $\overline{SS}$ of the SPI when bit SSM is set to 1. SSI bit is active low slave select signal when SSM is set to 1.
0 : Slave selected
1 : Slave not selected

DATA I/O REGISTER (SPIDR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR 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 SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the contents of the shift register (see Figure 37).

SERIAL PERIPHERAL INTERFACE (Cont'd)

10.4.6 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. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wake-up event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

10.4.7 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	Yes
Master Mode Fault Event	MODF		Yes	No
Overrun Error	OVR		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 interrupt mask in the CC register is reset (RIM instruction).

Table 17. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0011h	**SPIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0012h	**SPICR** Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0013h	**SPICSR** Reset Value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

10.5 SERIAL COMMUNICATIONS INTERFACE (SCI)

10.5.1 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.

10.5.2 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
- Four error detection flags:
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- Five interrupt sources with flags:
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error detected
- Parity control:
  - Transmits parity bit
  - Checks parity of received data byte
- Reduced power consumption mode

10.5.3 General Description

The interface is externally connected to another device by two pins (see Figure 43):

– 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 these 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.

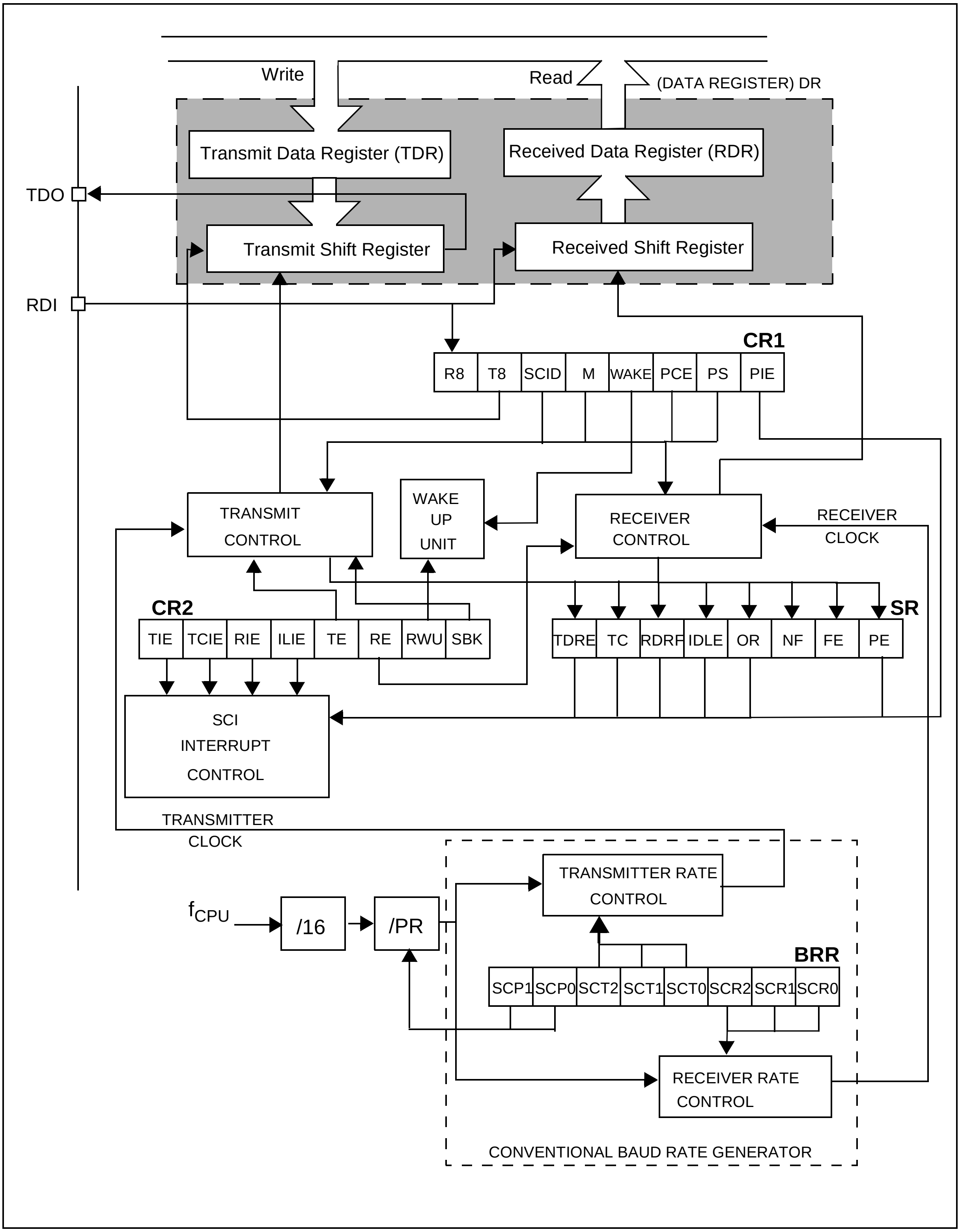
This interface uses two types of baud rate generator:

– 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 42. SCI Block Diagram


Write
Read
(DATA REGISTER) DR
Transmit Data Register (TDR)
Received Data Register (RDR)
TDO
Transmit Shift Register
Received Shift Register
RDI
CR1
R8
T8
SCID
M
WAKE
PCE
PS
PIE
TRANSMIT CONTROL
WAKE UP UNIT
RECEIVER CONTROL
RECEIVER CLOCK
CR2
TIE
TCIE
RIE
ILIE
TE
RE
RWU
SBK
SR
TDRE
TC
RDRF
IDLE
OR
NF
FE
PE
SCI INTERRUPT CONTROL
TRANSMITTER CLOCK
TRANSMITTER RATE CONTROL
f_{CPU}
/16
/PR
BRR
SCP1
SCP0
SCT2
SCT1
SCT0
SCR2
SCR1
SCR0
RECEIVER RATE CONTROL
CONVENTIONAL BAUD RATE GENERATOR

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.5.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 42. It contains 6 dedicated registers:

– Two control registers (SCICR1 & SCICR2)

– A status register (SCISR)

– A baud rate register (SCIBRR)

– An extended prescaler receiver register (SCIERPR)

– An extended prescaler transmitter register (SCIETPR)

Refer to the register descriptions in Section 10.5.7for the definitions of each bit.

10.5.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 42).

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

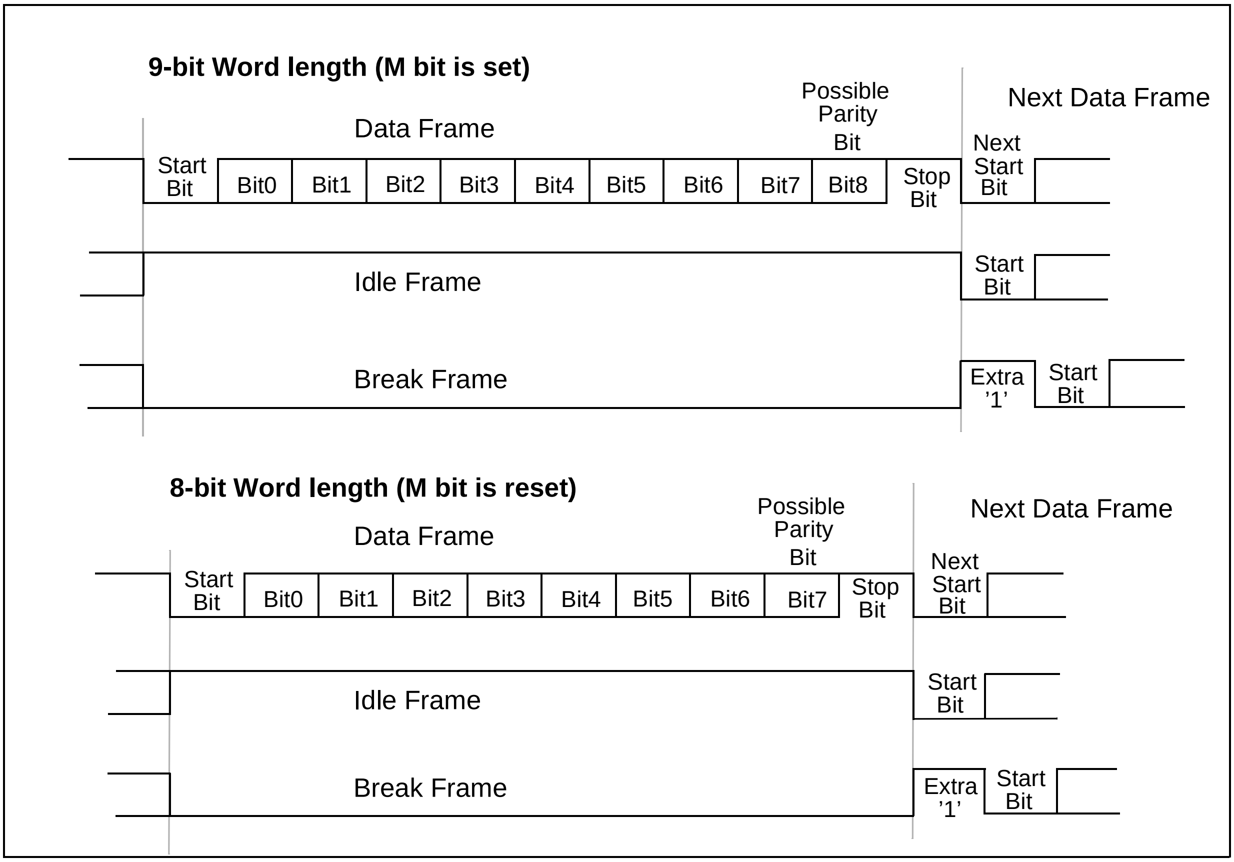
The TDO pin is in high state during the stop bit.

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 43. Word Length Programming


9-bit Word length (M bit is set)
Data Frame
Possible Parity Bit
Next Data Frame
Start Bit
Bit0
Bit1
Bit2
Bit3
Bit4
Bit5
Bit6
Bit7
Bit8
Stop Bit
Next Start Bit
Idle Frame
Start Bit
Break Frame
Extra '1'
Start Bit
8-bit Word length (M bit is reset)
Data Frame
Possible Parity Bit
Next Data Frame
Start Bit
Bit0
Bit1
Bit2
Bit3
Bit4
Bit5
Bit6
Bit7
Stop Bit
Next Start Bit
Idle Frame
Start Bit
Break Frame
Extra '1'
Start Bit

SERIAL COMMUNICATIONS INTERFACE (Cont’d)

10.5.4.2 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 SCICR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 42).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the SCIBRR and the SCIETPR 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 SCISR register and write the data to send in the SCIDR 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 SCISR register
2. A write to the SCIDR 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 SCIDR 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 SCIDR 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 SCIDR 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 SCISR register
2. A write to the SCIDR 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 43).

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 SCIDR.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.5.4.3 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 SCICR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the SCIDR register consists or a buffer (RDR) between the internal bus and the received shift register (see Figure 42).

Procedure

- Select the M bit to define the word length.
- Select the desired baud rate using the SCIBRR and the SCIERPR 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 SCISR register
2. A read to the SCIDR 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 RDR 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 SCISR register followed by a SCIDR 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 SCIDR 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 SCISR register read operation followed by a SCIDR 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.

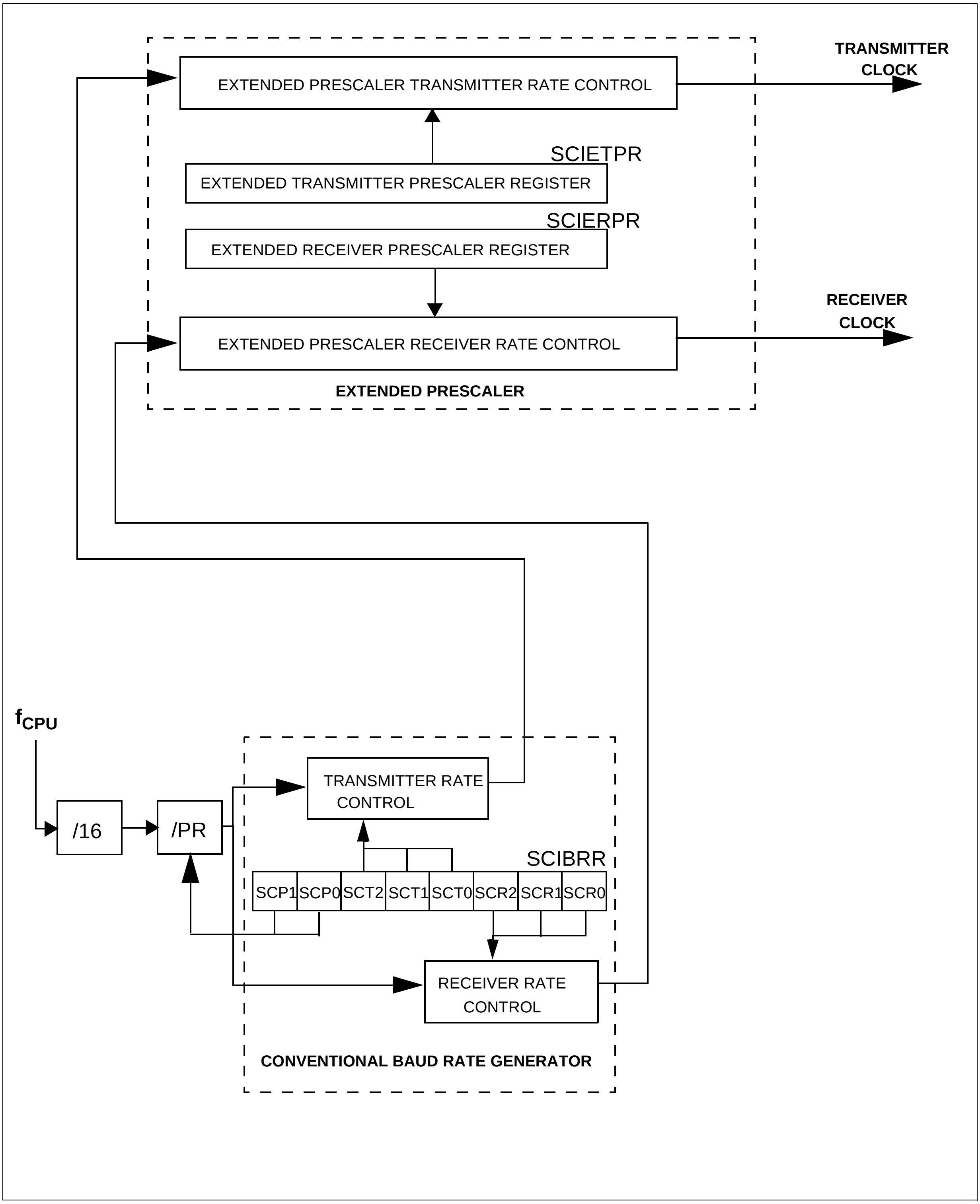
When the framing error is detected:

- the FE bit is set by hardware
- Data is transferred from the Shift register to the SCIDR 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 SCISR register read operation followed by a SCIDR register read operation.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 44. SCI Baud Rate and Extended Prescaler Block Diagram


TRANSMITTER CLOCK
EXTENDED PRESCALER TRANSMITTER RATE CONTROL
SCIETPR
EXTENDED TRANSMITTER PRESCALER REGISTER
SCIERPR
EXTENDED RECEIVER PRESCALER REGISTER
RECEIVER CLOCK
EXTENDED PRESCALER RECEIVER RATE CONTROL
EXTENDED PRESCALER
f_{CPU}
/16
/PR
TRANSMITTER RATE CONTROL
SCIBRR
SCP1
SCP0
SCT2
SCT1
SCT0
SCR2
SCR1
SCR0
RECEIVER RATE CONTROL
CONVENTIONAL BAUD RATE GENERATOR

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.5.4.4 Conventional Baud Rate Generation

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

$$Tx = \frac{f_{CPU}}{(16_*PR)_*TR} \qquad Rx = \frac{f_{CPU}}{(16_*PR)_*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

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

(see SCT[2:0] bits)

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

(see SCR[2:0] bits)

All these bits are in the SCIBRR 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 38400 baud.

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

10.5.4.5 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 44.

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 SCIERPR or the SCIETPR register.

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

$$Tx = \frac{f_{CPU}}{16_*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16_*ERPR*(PR*RR)}$$

with:

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

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

10.5.4.6 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 interrupts 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)

10.5.4.7 Parity Control

Parity control (generation of parity bit in trasmission and and parity chencking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 18.

Table 18. Frame Formats

M bit	PCE bit	SCI frame
0	0	| SB | 8 bit data | STB |
0	1	| SB | 7-bit data | PB | STB |
1	0	| SB | 9-bit data | STB |
1	1	| SB | 8-bit data PB | STB |

Legend: SB = Start Bit, STB = Stop Bit,
PB = Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: the parity bit is calculated to obtain an even number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 0 if even parity is selected (PS bit = 0).

Odd parity: the parity bit is calculated to obtain an odd number of “1s” inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Ex: data=00110101; 4 bits set => parity bit will be 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of “1s” if even parity is selected (PS=0) or an odd number of “1s” if odd parity is selected (PS=1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.5.5 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.

10.5.6 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
Parity Error	PE	PIE	Yes	No

The SCI interrupt events are connected to the same interrupt vector.

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.5.7 Register Description

STATUS REGISTER (SCISR)
Read Only
Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE

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 bit=1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR 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 unless the TDRE bit is cleared.

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 SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR 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 to the SCIDR register. An interrupt is generated if RIE=1 in the SCICR2 register. It is cleared by hardware when RE=0 or by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 SCICR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 SCICR2 register. It is cleared by hardware when RE=0 by a software sequence (an access to the SCISR register followed by a read to the SCIDR 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 SCISR register followed by a read to the SCIDR 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 SCISR register followed by a read to the SCIDR 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 = **PE** *Parity error.*
This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE=1 in the SCICR1 register.
0: No parity error
1: Parity error

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

CONTROL REGISTER 1 (SCICR1)

Read/Write

Reset Value: x000 0000 (x0h)

7							0
R8	T8	SCID	M	WAKE	PCE	PS	PIE

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 5 = **SCID** *Disabled for low power consumption*
When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption.This bit is set and cleared by software.
0: SCI enabled
1: SCI prescaler and outputs disabled

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

Note: The M bit must not be modified during a data transfer (both transmission and reception).

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

Bit 2 = **PCE** *Parity control enable.*
This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M=1; 8th bit if M=0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission).
0: Parity control disabled
1: Parity control enabled

Bit 1 = **PS** *Parity selection.*
This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte.
0: Even parity
1: Odd parity

Bit 0 = **PIE** *Parity interrupt enable.*
This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software.
0: Parity error interrupt disabled
1: Parity error interrupt enabled.

SERIAL COMMUNICATIONS INTERFACE (Cont’d)

CONTROL REGISTER 2 (SCICR2)

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 SCISR 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 SCISR 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 SCISR 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 SCISR 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 SCISR 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

Note: Before selecting Mute mode (setting the RWU bit), the SCI must receive some data first, otherwise it cannot function in Mute mode with wakeup by idle line detection.

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 (SCIDR)

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 42).

The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 42).

BAUD RATE REGISTER (SCIBRR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bits 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

Bits 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

Bits 2:0 = **SCR[2:0]** *SCI Receiver rate divisor.*
These 3 bits, in conjunction with the SCP[1:0] 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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

EXTENDED RECEIVE PRESCALER DIVISION REGISTER (SCIERPR)

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

Bits 7:0 = **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 44) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255).

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

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (SCIETPR)

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

Bits 7:0 = **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 44) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255).

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

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Table 19. SCI Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
1D	**SCIERPR** Reset Value	ERPR7 0	ERPR6 0	ERPR5 0	ERPR4 0	ERPR3 0	ERPR2 0	ERPR1 0	ERPR0 0
1E	**SCIETPR** Reset Value	ETPR7 0	ETPR6 0	ETPR5 0	ETPR4 0	ETPR3 0	ETPR2 0	ETPR1 0	ETPR0 0
20	**SCISR** Reset Value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
21	**SCIDR** Reset Value	DR7 x	DR6 x	DR5 x	DR4 x	DR3 x	DR2 x	DR1 x	DR0 x
22	**SCIBRR** Reset Value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
23	**SCICR1** Reset Value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
24	**SCICR2** Reset Value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0

10.6 USB INTERFACE (USB)

10.6.1 Introduction

The USB Interface implements a low-speed function interface between the USB and the ST7 microcontroller. It is a highly integrated circuit which includes the transceiver, 3.3 voltage regulator, SIE and DMA. No external components are needed apart from the external pull-up on USBDM for low speed recognition by the USB host. The use of DMA architecture allows the endpoint definition to be completely flexible. Endpoints can be configured by software as in or out.

10.6.2 Main Features

- USB Specification Version 1.1 Compliant
- Supports Low-Speed USB Protocol
- Two or Three Endpoints (including default one) depending on the device (see device feature list and register map)
- CRC generation/checking, NRZI encoding/ decoding and bit-stuffing
- USB Suspend/Resume operations
- DMA Data transfers
- On-Chip 3.3V Regulator
- On-Chip USB Transceiver

10.6.3 Functional Description

The block diagram in Figure 45, gives an overview of the USB interface hardware.

For general information on the USB, refer to the "Universal Serial Bus Specifications" document available at http//:www.usb.org.

Serial Interface Engine

The SIE (Serial Interface Engine) interfaces with the USB, via the transceiver.

The SIE processes tokens, handles data transmission/reception, and handshaking as required by the USB standard. It also performs frame formatting, including CRC generation and checking.

Endpoints

The Endpoint registers indicate if the microcontroller is ready to transmit/receive, and how many bytes need to be transmitted.

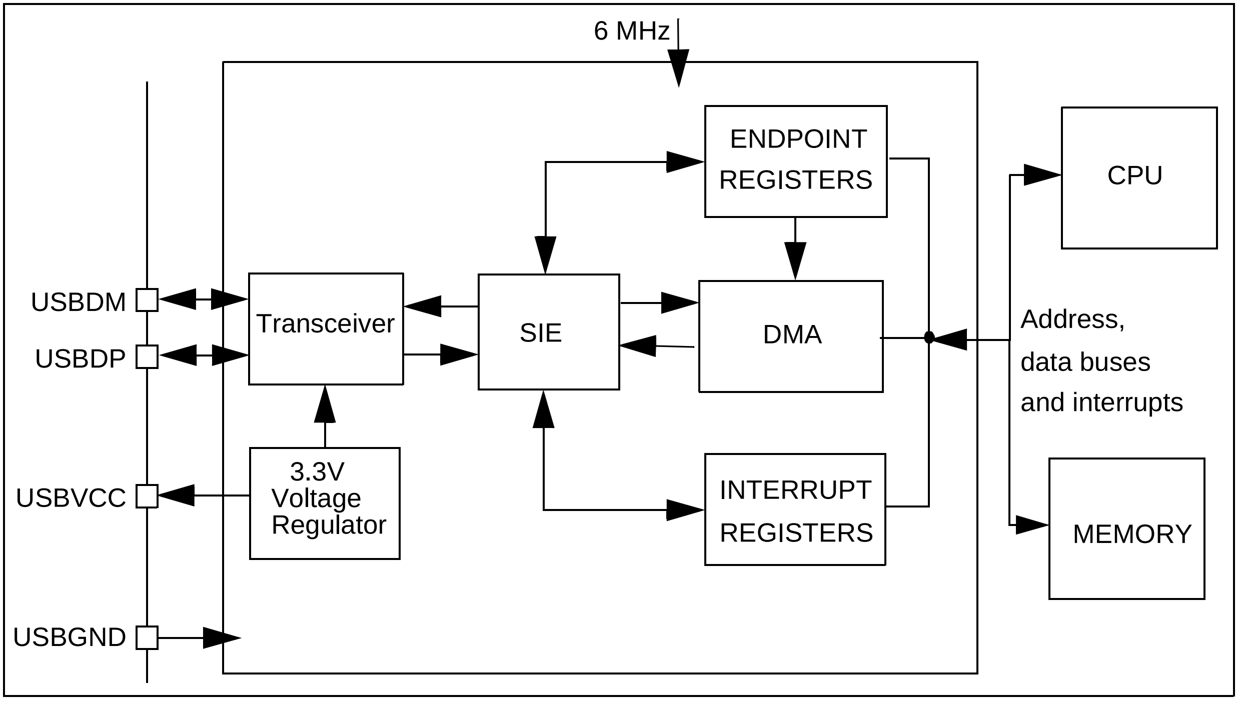
DMA

When a token for a valid Endpoint is recognized by the USB interface, the related data transfer takes place, using DMA. At the end of the transaction, an interrupt is generated.

Interrupts

By reading the Interrupt Status register, application software can know which USB event has occurred.

Figure 45. USB Block Diagram



USB INTERFACE (Cont'd)

10.6.4 Register Description

DMA ADDRESS REGISTER (DMAR)

Read / Write

Reset Value: Undefined

7							0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8

Bits 7:0=**DA[15:8]** *DMA address bits 15-8.* Software must write the start address of the DMA memory area whose most significant bits are given by DA15-DA6. The remaining 6 address bits are set by hardware. See the description of the IDR register and Figure 46.

INTERRUPT/DMA REGISTER (IDR)

Read / Write

Reset Value: xxxx 0000 (x0h)

7							0
DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0

Bits 7:6 = **DA[7:6]** *DMA address bits 7-6.* Software must reset these bits. See the description of the DMAR register and Figure 46.

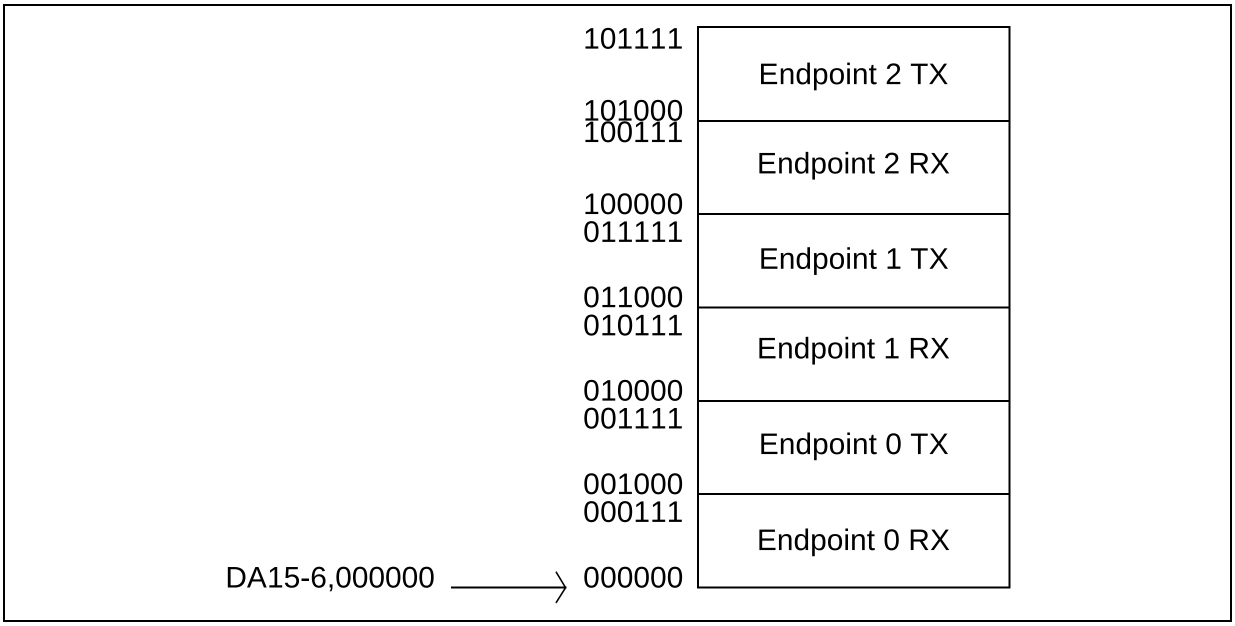
Bits 5:4 = **EP[1:0]** *Endpoint number* (read-only). These bits identify the endpoint which required attention.
00: Endpoint 0
01: Endpoint 1
10: Endpoint 2

When a CTR interrupt occurs (see register ISTR) the software should read the EP bits to identify the endpoint which has sent or received a packet.

Bits 3:0 = **CNT[3:0]** *Byte count* (read only). This field shows how many data bytes have been received during the last data reception.

Note: Not valid for data transmission.

Figure 46. DMA Buffers


101111
Endpoint 2 TX
101000
100111
Endpoint 2 RX
100000
011111
Endpoint 1 TX
011000
010111
Endpoint 1 RX
010000
001111
Endpoint 0 TX
001000
000111
Endpoint 0 RX
DA15-6,000000
000000

USB INTERFACE (Cont'd)

PID REGISTER (PIDR)

Read only

Reset Value: xx00 0000 (x0h)

7							0
TP3	TP2	0	0	0	RX_ SEZ	RXD	0

Bits 7:6 = **TP[3:2]** *Token PID bits 3 & 2*.
USB token PIDs are encoded in four bits. **TP[3:2]** correspond to the variable token PID bits 3 & 2.
Note: PID bits 1 & 0 have a fixed value of 01.
When a CTR interrupt occurs (see register ISTR) the software should read the TP3 and TP2 bits to retrieve the PID name of the token received.
The USB standard defines TP bits as:

TP3	TP2	PID Name
0	0	OUT
1	0	IN
1	1	SETUP

Bits 5:3 Reserved. Forced by hardware to 0.

Bit 2 = **RX_SEZ** *Received single-ended zero*
This bit indicates the status of the RX_SEZ transceiver output.
0: No SE0 (single-ended zero) state
1: USB lines are in SE0 (single-ended zero) state

Bit 1 = **RXD** *Received data*
0: No K-state
1: USB lines are in K-state

This bit indicates the status of the RXD transceiver output (differential receiver output).

Note: If the environment is noisy, the RX_SEZ and RXD bits can be used to secure the application. By interpreting the status, software can distinguish a valid End Suspend event from a spurious wake-up due to noise on the external USB line. A valid End Suspend is followed by a Resume or Reset sequence. A Resume is indicated by RXD=1, a Reset is indicated by RX_SEZ=1.

Bit 0 = Reserved. Forced by hardware to 0.

INTERRUPT STATUS REGISTER (ISTR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF

When an interrupt occurs these bits are set by hardware. Software must read them to determine the interrupt type and clear them after servicing.
Note: These bits cannot be set by software.

Bit 7 = **SUSP** *Suspend mode request*.
This bit is set by hardware when a constant idle state is present on the bus line for more than 3 ms, indicating a suspend mode request from the USB bus. The suspend request check is active immediately after each USB reset event and its disabled by hardware when suspend mode is forced (FSUSP bit of CTLR register) until the end of resume sequence.

Bit 6 = **DOVR** *DMA over/underrun*.
This bit is set by hardware if the ST7 processor can't answer a DMA request in time.
0: No over/underrun detected
1: Over/underrun detected

Bit 5 = **CTR** *Correct Transfer.* This bit is set by hardware when a correct transfer operation is performed. The type of transfer can be determined by looking at bits TP3-TP2 in register PIDR. The Endpoint on which the transfer was made is identified by bits EP1-EP0 in register IDR.
0: No Correct Transfer detected
1: Correct Transfer detected

Note: A transfer where the device sent a NAK or STALL handshake is considered not correct (the host only sends ACK handshakes). A transfer is considered correct if there are no errors in the PID and CRC fields, if the DATA0/DATA1 PID is sent as expected, if there were no data overruns, bit stuffing or framing errors.

Bit 4 = **ERR** *Error.*
This bit is set by hardware whenever one of the errors listed below has occurred:
0: No error detected
1: Timeout, CRC, bit stuffing or nonstandard framing error detected

USB INTERFACE (Cont'd)

Bit 3 = **IOVR** *Interrupt overrun.*
This bit is set when hardware tries to set ERR, or SOF before they have been cleared by software.
0: No overrun detected
1: Overrun detected

Bit 2 = **ESUSP** *End suspend mode*.
This bit is set by hardware when, during suspend mode, activity is detected that wakes the USB interface up from suspend mode.

This interrupt is serviced by a specific vector, in order to wake up the ST7 from HALT mode.
0: No End Suspend detected
1: End Suspend detected

Bit 1 = **RESET** *USB reset.*
This bit is set by hardware when the USB reset sequence is detected on the bus.
0: No USB reset signal detected
1: USB reset signal detected

Note: The DADDR, EP0RA, EP0RB, EP1RA, EP1RB, EP2RA and EP2RB registers are reset by a USB reset.

Bit 0 = **SOF** *Start of frame.*
This bit is set by hardware when a low-speed SOF indication (keep-alive strobe) is seen on the USB bus. It is also issued at the end of a resume sequence.
0: No SOF signal detected
1: SOF signal detected

Note: To avoid spurious clearing of some bits, it is recommended to clear them using a load instruction where all bits which must not be altered are set, and all bits to be cleared are reset. Avoid read-modify-write instructions like AND , XOR..

INTERRUPT MASK REGISTER (IMR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM

Bits 7:0 = These bits are mask bits for all interrupt condition bits included in the ISTR. Whenever one of the IMR bits is set, if the corresponding ISTR bit is set, and the I bit in the CC register is cleared, an interrupt request is generated. For an explanation of each bit, please refer to the corresponding bit description in ISTR.

CONTROL REGISTER (CTLR)

Read / Write

Reset Value: 0000 0110 (06h)

7							0
0	0	0	0	RESUME	PDWN	FSUSP	FRES

Bits 7:4 = Reserved. Forced by hardware to 0.

Bit 3 = **RESUME** *Resume*.
This bit is set by software to wake-up the Host when the ST7 is in suspend mode.
0: Resume signal not forced
1: Resume signal forced on the USB bus.

Software should clear this bit after the appropriate delay.

Bit 2 = **PDWN** *Power down*.
This bit is set by software to turn off the 3.3V on-chip voltage regulator that supplies the external pull-up resistor and the transceiver.
0: Voltage regulator on
1: Voltage regulator off

Note: After turning on the voltage regulator, software should allow at least 3 µs for stabilisation of the power supply before using the USB interface.

Bit 1 = **FSUSP** *Force suspend mode*.
This bit is set by software to enter Suspend mode. The ST7 should also be halted allowing at least 600 ns before issuing the HALT instruction.
0: Suspend mode inactive
1: Suspend mode active

When the hardware detects USB activity, it resets this bit (it can also be reset by software).

Bit 0 = **FRES** *Force reset.*
This bit is set by software to force a reset of the USB interface, just as if a RESET sequence came from the USB.
0: Reset not forced
1: USB interface reset forced.

The USB is held in RESET state until software clears this bit, at which point a "USB-RESET" interrupt will be generated if enabled.

USB INTERFACE (Cont'd)

DEVICE ADDRESS REGISTER (DADDR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Bit 7 = Reserved. Forced by hardware to 0.

Bits 6:0 = **ADD[6:0]** *Device address, 7 bits.*

Software must write into this register the address sent by the host during enumeration.

Note: This register is also reset when a USB reset is received from the USB bus or forced through bit FRES in the CTLR register.

ENDPOINT n REGISTER A (EPnRA)

Read / Write

Reset Value: 0000 xxxx (0xh)

7							0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC 3	TBC 2	TBC 1	TBC 0

These registers (**EP0RA**, **EP1RA** and **EP2RA**) are used for controlling data transmission. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RA register are not available on some devices (see device feature list and register map).

Bit 7 = **ST_OUT** *Status out.*
This bit is set by software to indicate that a status out packet is expected: in this case, all nonzero OUT data transfers on the endpoint are STALLed instead of being ACKed. When ST_OUT is reset, OUT transactions can have any number of bytes, as needed.

Bit 6 = **DTOG_TX** *Data Toggle, for transmission transfers.*
It contains the required value of the toggle bit (0=DATA0, 1=DATA1) for the next transmitted data packet. This bit is set by hardware at the reception of a SETUP PID. DTOG_TX toggles only when the transmitter has received the ACK signal from the USB host. DTOG_TX and also DTOG_RX (see EPnRB) are normally updated by hardware, at the receipt of a relevant PID. They can be also written by software.

Bits 5:4 = **STAT_TX[1:0]** *Status bits, for transmission transfers.*
These bits contain the information about the endpoint status, which are listed below:

STAT_TX1	STAT_TX0	Meaning
0	0	**DISABLED:** transmission transfers cannot be executed.
0	1	**STALL**: the endpoint is stalled and all transmission requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all transmission requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for transmission.

These bits are written by software. Hardware sets the STAT_TX bits to NAK when a correct transfer has occurred (CTR=1) related to a IN or SETUP transaction addressed to this endpoint; this allows the software to prepare the next set of data to be transmitted.

Bits 3:0 = **TBC[3:0]** *Transmit byte count for Endpoint n.*
Before transmission, after filling the transmit buffer, software must write in the TBC field the transmit packet size expressed in bytes (in the range 0-8).

USB INTERFACE (Cont'd)

ENDPOINT n REGISTER B (EPnRB)

Read / Write

Reset Value: 0000 xxxx (0xh)

7							0
CTRL	DTOG _RX	STAT _RX1	STAT _RX0	EA3	EA2	EA1	EA0

These registers (**EP1RB** and **EP2RB**) are used for controlling data reception on Endpoints 1 and 2. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RB register are not available on some devices (see device feature list and register map).

Bit 7 = **CTRL** *Control.*
This bit should be 0.

Note: If this bit is 1, the Endpoint is a control endpoint. (Endpoint 0 is always a control Endpoint, but it is possible to have more than one control Endpoint).

Bit 6 = **DTOG_RX** *Data toggle, for reception transfers*.
It contains the expected value of the toggle bit (0=DATA0, 1=DATA1) for the next data packet. This bit is cleared by hardware in the first stage (Setup Stage) of a control transfer (SETUP transactions start always with DATA0 PID). The receiver toggles DTOG_RX only if it receives a correct data packet and the packet's data PID matches the receiver sequence bit.

Bits 5:4 = **STAT_RX [1:0]** *Status bits, for reception transfers.*
These bits contain the information about the endpoint status, which are listed below:

STAT_RX1	STAT_RX0	Meaning
0	0	**DISABLED**: reception transfers cannot be executed.
0	1	**STALL:** the endpoint is stalled and all reception requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all reception requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for reception.

These bits are written by software. Hardware sets the STAT_RX bits to NAK when a correct transfer has occurred (CTR=1) related to an OUT or SETUP transaction addressed to this endpoint, so the software has the time to elaborate the received data before acknowledging a new transaction.

Bits 3:0 = **EA[3:0]** *Endpoint address*.
Software must write in this field the 4-bit address used to identify the transactions directed to this endpoint. Usually EP1RB contains "0001" and EP2RB contains "0010".

ENDPOINT 0 REGISTER B (EP0RB)

Read / Write

Reset Value: 1000 0000 (80h)

7							0
1	DTOG RX	STAT RX1	STAT RX0	0	0	0	0

This register is used for controlling data reception on Endpoint 0. It is also reset by the USB bus reset.

Bit 7 = Forced by hardware to 1.

Bits 6:4 = Refer to the EPnRB register for a description of these bits.

Bits 3:0 = Forced by hardware to 0.

USB INTERFACE (Cont'd)

10.6.5 Programming Considerations

The interaction between the USB interface and the application program is described below. Apart from system reset, action is always initiated by the USB interface, driven by one of the USB events associated with the Interrupt Status Register (ISTR) bits.

10.6.5.1 Initializing the Registers

At system reset, the software must initialize all registers to enable the USB interface to properly generate interrupts and DMA requests.

1. Initialize the DMAR, IDR, and IMR registers (choice of enabled interrupts, address of DMA buffers). Refer the paragraph titled initializing the DMA Buffers.
2. Initialize the EP0RA and EP0RB registers to enable accesses to address 0 and endpoint 0 to support USB enumeration. Refer to the paragraph titled Endpoint Initialization.
3. When addresses are received through this channel, update the content of the DADDR.
4. If needed, write the endpoint numbers in the EA fields in the EP1RB and EP2RB register.

10.6.5.2 Initializing DMA buffers

The DMA buffers are a contiguous zone of memory whose maximum size is 48 bytes. They can be placed anywhere in the memory space to enable the reception of messages. The 10 most significant bits of the start of this memory area are specified by bits DA15-DA6 in registers DMAR and IDR, the remaining bits are 0. The memory map is shown in Figure 46.

Each buffer is filled starting from the bottom (last 3 address bits=000) up.

10.6.5.3 Endpoint Initialization

To be ready to receive:

Set STAT_RX to VALID (11b) in EP0RB to enable reception.

To be ready to transmit:

1. Write the data in the DMA transmit buffer.
2. In register EPnRA, specify the number of bytes to be transmitted in the TBC field
3. Enable the endpoint by setting the STAT_TX bits to VALID (11b) in EPnRA.

Note: Once transmission and/or reception are enabled, registers EPnRA and/or EPnRB (respectively) must not be modified by software, as the hardware can change their value on the fly. When the operation is completed, they can be accessed again to enable a new operation.

10.6.5.4 Interrupt Handling

Start of Frame (SOF)

The interrupt service routine may monitor the SOF events for a 1 ms synchronization event to the USB bus. This interrupt is generated at the end of a resume sequence and can also be used to detect this event.

USB Reset (RESET)

When this event occurs, the DADDR register is reset, and communication is disabled in all endpoint registers (the USB interface will not respond to any packet). Software is responsible for reenabling endpoint 0 within 10 ms of the end of reset. To do this, set the STAT_RX bits in the EP0RB register to VALID.

Suspend (SUSP)

The CPU is warned about the lack of bus activity for more than 3 ms, which is a suspend request. The software should set the USB interface to suspend mode and execute an ST7 HALT instruction to meet the USB-specified power constraints.

End Suspend (ESUSP)

The CPU is alerted by activity on the USB, which causes an ESUSP interrupt. The ST7 automatically terminates HALT mode.

Correct Transfer (CTR)

1. When this event occurs, the hardware automatically sets the STAT_TX or STAT_RX to NAK.
   Note: Every valid endpoint is NAKed until software clears the CTR bit in the ISTR register, independently of the endpoint number addressed by the transfer which generated the CTR interrupt.
   Note: If the event triggering the CTR interrupt is a SETUP transaction, both STAT_TX and STAT_RX are set to NAK.
2. Read the PIDR to obtain the token and the IDR to get the endpoint number related to the last transfer.
   Note: When a CTR interrupt occurs, the TP3-TP2 bits in the PIDR register and EP1-EP0 bits in the IDR register stay unchanged until the CTR bit in the ISTR register is cleared.
3. Clear the CTR bit in the ISTR register.

USB INTERFACE (Cont'd)

Table 20. USB Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
25	PIDR	TP3	TP2	0	0	0	RX_SEZ	RXD	0
	Reset Value	x	x	0	0	0	0	0	0
26	DMAR	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
	Reset Value	x	x	x	x	x	x	x	x
27	IDR	DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0
	Reset Value	x	x	x	x	0	0	0	0
28	ISTR	SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF
	Reset Value	0	0	0	0	0	0	0	0
29	IMR	SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM
	Reset Value	0	0	0	0	0	0	0	0
2A	CTLR	0	0	0	0	RESUME	PDWN	FSUSP	FRES
	Reset Value	0	0	0	0	0	1	1	0
2B	DADDR	0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	Reset Value	0	0	0	0	0	0	0	0
2C	EP0RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
2D	EP0RB	1	DTOG_RX	STAT_RX1	STAT_RX0	0	0	0	0
	Reset Value	1	0	0	0	0	0	0	0
2E	EP1RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
2F	EP1RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset Value	0	0	0	0	x	x	x	x
30	EP2RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
31	EP2RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset Value	0	0	0	0	x	x	x	x

10.7 10-BIT A/D CONVERTER (ADC)

10.7.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 8 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 8 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.7.2 Main Features

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

The block diagram is shown in Figure 47.

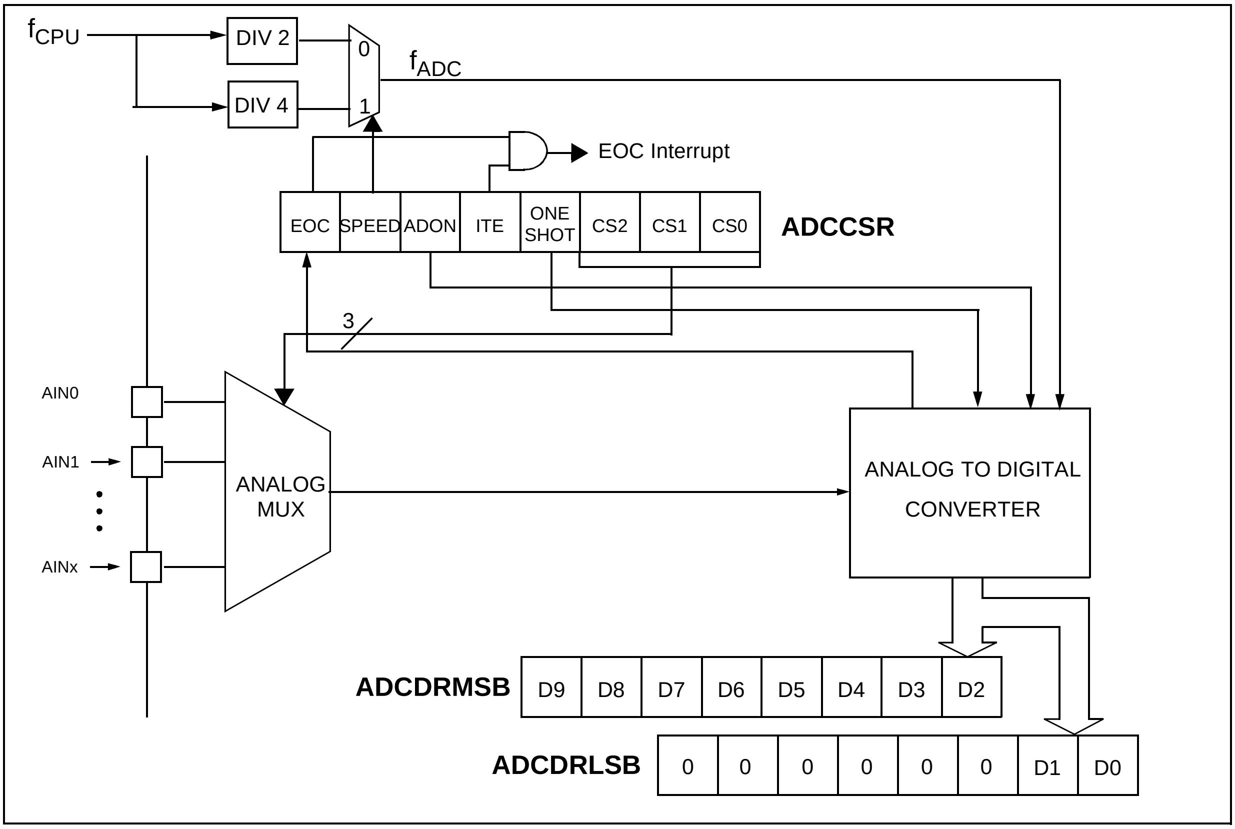
10.7.3 Functional Description

10.7.3.1 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 47. ADC Block Diagram


fCPU
DIV 2
DIV 4
0
1
fADC
EOC Interrupt
EOC
SPEED
ADON
ITE
ONE SHOT
CS2
CS1
CS0
ADCCSR
3
AIN0
AIN1
AINx
ANALOG MUX
ANALOG TO DIGITAL CONVERTER
ADCDRMSB
D9
D8
D7
D6
D5
D4
D3
D2
ADCDRLSB
0
0
0
0
0
0
D1
D0

10-BIT A/D CONVERTER (ADC) (Cont’d)

10.7.3.2 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 V_{DDA} (high-level voltage reference) then the conversion result is FFh in the ADCDRMSB register and 03h in the ADCDRLSB register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRMSB and ADCDRLSB registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRMSB and ADCDRLSB registers. 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.

10.7.3.3 A/D Conversion

Conversion can be performed in One-Shot or Continuous mode. Continuous mode is typically used for monitoring a single channel. One-shot mode should be used when the application requires inputs from several channels.

ADC Configuration

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 ADCCSR register:

– Select the CS[2:0] bits to assign the analog channel to convert.

ADC One-Shot Conversion mode

In the ADCCSR register:

1.Set the ONE SHOT bit to put the A/D converter in one shot mode.

2.Set the ADON bit to enable the A/D converter and to start the conversion. The EOC bit is kept low by hardware during the conversion.

Note: Changing the A/D channel during conversion will stop the current conversion and start conversion of the newly selected channel.

When a conversion is complete:

– The EOC bit is set by hardware.
– An interrupt request is generated if the ITE bit is set.
– The ADON bit is reset by hardware.
– The result is in the ADCDR registers.

To read the 10 bits, perform the following steps:

1. Wait for interrupt or poll the EOC bit
2. Read ADCDRLSB
3. Read ADCDRMSB

The EOC bit is reset by hardware once the ADCDRMSB is read.

To read only 8 bits, perform the following steps:

1. Wait for interrupt or poll the EOC bit
2. Read ADCDRMSB

The EOC bit is reset by hardware once the ADCDRMSB is read.

To start another conversion, user should set the ADON bit once again.

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

ADC Continuous Conversion mode

In the ADCCSR register:

1. Reset the ONE SHOT bit to put the A/D converter in continuous mode.
2. 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.

Note: Changing the A/D channel during conversion will stop the current conversion and start conversion of the newly selected channel.

When a conversion is complete:

- The EOC bit is set by hardware.
- An interrupt request is generated if the ITE bit is set.
- The result is in the ADCDR registers and remains valid until the next conversion has ended.

To read the 10 bits, perform the following steps:

1. Wait for interrupt or poll the EOC bit
2. Read ADCDRLSB
3. Read ADCDRMSB

The EOC bit is reset by hardware once the ADCDRMSB is read.

To read only 8 bits, perform the following steps:

1. Wait for interrupt
2. Read ADCDRMSB

The EOC bit is reset by hardware once the ADCDRMSB is read.

Changing the conversion channel

The application can change channels during conversion. In this case the current conversion is stopped and the A/D converter starts converting the newly selected channel.

ADCCR consistency

If an End Of Conversion event occurs after software has read the ADCDRLSB but before it has read the ADCDRMSB, there would be a risk that the two values read would belong to different samples.

To guarantee consistency:

- The ADCDRMSB and the ADCDRLSB are locked when the ADCCRLSB is read
- The ADCDRMSB and the ADCDRLSB are unlocked when the MSB is read or when ADON is reset.

Thus, it is mandatory to read the ADCDRMSB just after reading the ADCDRLSB. This is especially important in continuous mode, as the ADCDR register will not be updated until the ADCDRMSB is read.

10.7.4 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 t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

10.7.5 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
End of Conversion	EOC	ITE	Yes	No

Note: The EOC interrupt event is connected to an interrupt vector (see Interrupts chapter).
It generates an interrupt if the ITE bit is set in the ADCCSR register and the interrupt mask in the CC register is reset (RIM instruction).

10-BIT A/D CONVERTER (ADC) (Cont’d)

10.7.6 Register Description

CONTROL/STATUS REGISTER (ADCCSR)

Read/Write (Except bit 7 read only)

Reset Value: 0000 0000 (00h)

7							0
EOC	SPEED	ADON	ITE	ONE SHOT	CS2	CS1	CS0

Bit 7 = **EOC** *End of Conversion*
This bit is set by hardware. It is cleared by software reading the ADCDRMSB register.
0: Conversion is not complete
1: Conversion complete

Bit 6 = **SPEED** *ADC clock selection*
This bit is set and cleared by software.
0: $f_{ADC} = f_{CPU}/2$
1: $f_{ADC} = f_{CPU}/4$

Bit 5 = **ADON** *A/D Converter on*
This bit is set and cleared by software or by hardware after the end of a one shot conversion.
0: Disable ADC and stop conversion
1: Enable ADC and start conversion

Bit 4 = **ITE** *Interrupt Enable*

This bit is set and cleared by software.
0: EOC Interrupt disabled
1: EOC Interrupt enabled

Bit 3 = **ONESHOT** *One Shot Conversion Selection*
This bit is set and cleared by software.
0: Continuous conversion mode
1: One Shot conversion mode

Bit 2:0 = **CS[2:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert.

Channel*	CS2	CS1	CS0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

*The number of channels is device dependent. Refer to the device pinout description.

DATA REGISTER (ADCDRMSB)

Read Only

Reset Value: 0000 0000 (00h)

7							0
D9	D8	D7	D6	D5	D4	D3	D2

Bit 7:0 = **D[9:2]** *MSB of Analog Converted Value*
This register contains the MSB of the converted analog value.

DATA REGISTER (ADCDRLSB)

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	D1	D0

Bit 7:2 = Reserved. Forced by hardware to 0.

Bit 1:0 = **D[1:0]** *LSB of Analog Converted Value*
This register contains the LSB of the converted analog value.

11 INSTRUCTION SET

11.1 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 21. 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)

11.1.1 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

11.1.2 Immediate

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

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

11.1.3 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.

11.1.4 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.

11.1.5 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)

11.1.6 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 22. 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

11.1.7 Relative Mode (Direct, Indirect)

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

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

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

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

11.2 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	

12 ELECTRICAL CHARACTERISTICS

12.1 PARAMETER CONDITIONS

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

12.1.1 Minimum and Maximum Values

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

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

12.1.2 Typical Values

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

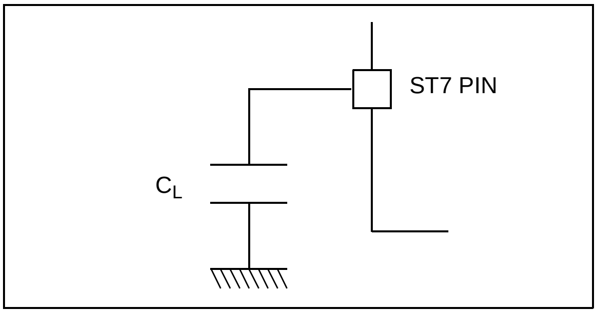
12.1.3 Typical Curves

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

12.1.4 Loading Capacitor

The loading conditions used for pin parameter measurement are shown in Figure 48.

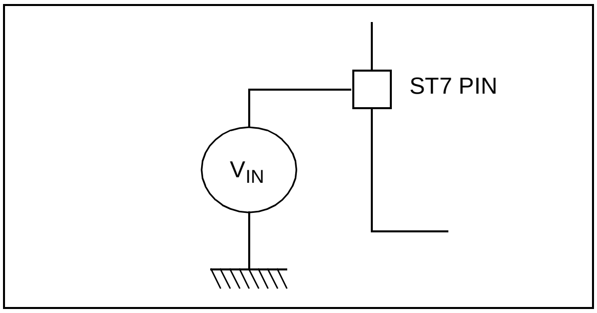
Figure 48. Pin Loading Conditions



12.1.5 Pin Input Voltage

The input voltage measurement on a pin of the device is described in Figure 49.

Figure 49. Pin Input Voltage



12.2 ABSOLUTE MAXIMUM RATINGS

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

12.2.1 Voltage Characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.0	V
V_{DDA} - V_{SSA}	Analog Reference Voltage	6.0	
V_{IN} 1) & 2)	Input voltage on true open drain pin	V_{SS}-0.3 to 6.0	
	Input voltage on any other pin	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	See “Absolute Electrical Sensitivity” on page 107.	

12.2.2 Current Characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source) 3)	80	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) 3)	80	
I_{IO}	Output current sunk by any standard I/O and control pin	25	
	Output current sunk by any high sink I/O pin	50	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$ 2) & 4)	Injected current on V_{PP} pin	75	
	Injected current on $\overline{\text{RESET}}$ pin	± 5	
	Injected current on OSCIN and OSCOUT pins	± 5	
	Injected current on any other pin 5) & 6)	± 5	
$\Sigma I_{INJ(PIN)}$ 2)	Total injected current (sum of all I/O and control pins) 5)	± 20	

Notes:

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

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

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

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

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

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

12.2.3 Thermal Characteristics

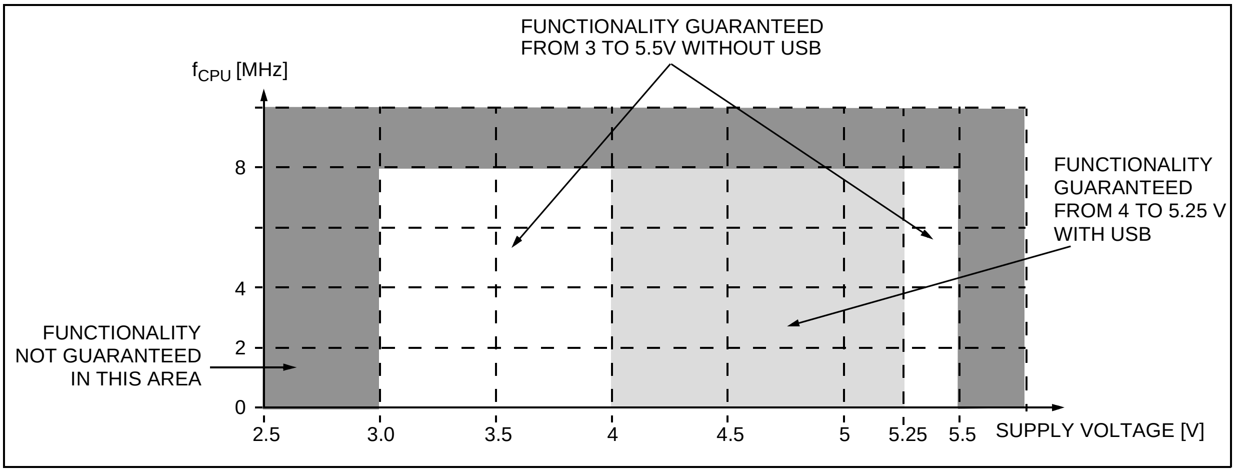
Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature	175	°C

12.3 OPERATING CONDITIONS

12.3.1 General Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating Supply Voltage (USB Full Spec.)	f_{CPU} = 8 MHz	4	5	5.25	V
	Operating Supply Voltage (No USB)	f_{CPU} = 8 MHz	4	5	5.5	
	Operating Supply Voltage (No USB, No LVD)	f_{CPU} = TBD	TBD	5	5.5	
V_{DDA}	Analog reference voltage		V_{DD}		V_{DD}	
V_{SSA}	Analog reference voltage		V_{SS}		V_{SS}	
f_{CPU}	Operating frequency	f_{OSC} = 12MHz			8	MHz
		f_{OSC} = 6MHz			4	
T_A	Ambient temperature range		0		70	°C

Figure 50. f_{CPU} Maximum Operating Frequency Versus V_{DD} Supply Voltage



OPERATING CONDITIONS (Cont'd)

12.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A. Refer to Figure 15.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IT+}	Low Voltage Reset Threshold (V_{DD} rising)	V_{DD} Max. Variation 50V/ms	3.75	3.85	3.95	V
V_{IT-}	Low Voltage Reset Threshold (V_{DD} falling)	V_{DD} Max. Variation 50V/ms	3.6	3.7	3.8	V
V_{hyst}	Hysteresis (V_{IT+} - V_{IT-})		120	150		mV
Vt_{POR}	V_{DD} rise time rate [2]		0.5		50	V/ms
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}[1]	Not detected by the LVD			40	ns

Notes:

1. Not tested, guaranteed by design.
2. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. Not tested in production.

12.4 SUPPLY CURRENT CHARACTERISTICS

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

Symbol	Parameter	Conditions		Typ 1)	Max	Unit
$\Delta I_{DD(\Delta Ta)}$	Supply current variation vs. temperature	Constant V_{DD} and f_{CPU}			10	%
I_{DD}	CPU RUN mode	I/Os in input mode	f_{CPU} = 4 MHz	6	TBD 2)	mA
			f_{CPU} = 8 MHz	12	TBD 2)	
	CPU WAIT mode		f_{CPU} = 8 MHz	10	TBD 2)	mA
	CPU HALT mode	with LVD		90	120 3)	µA
		without LVD			10 3)	
	USB Suspend mode4)			90	120	µA
$I_{DD(ADC)}$	ADC supply current when converting	f_{ADC}=4MHz		10005)		µA

Note 1: Typical data are based on T_A=25°C and not tested in production

Note 2: Oscillator and watchdog running.
All others peripherals disabled.

Note 3: USB Transceiver and ADC are powered down.

Note 4: Low voltage reset function enabled.
CPU in HALT mode.
Current consumption of external pull-up (1.5Kohms to USBVCC) and pull-down (15Kohms to V_{SSA}) not included.

Figure 51. Typ. I_{DD} in RUN at 4 and 8 MHz f_{CPU}

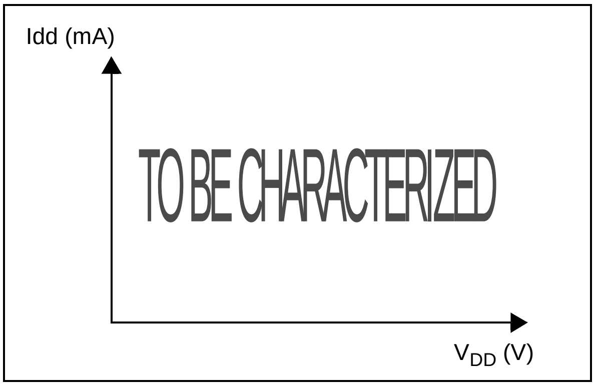
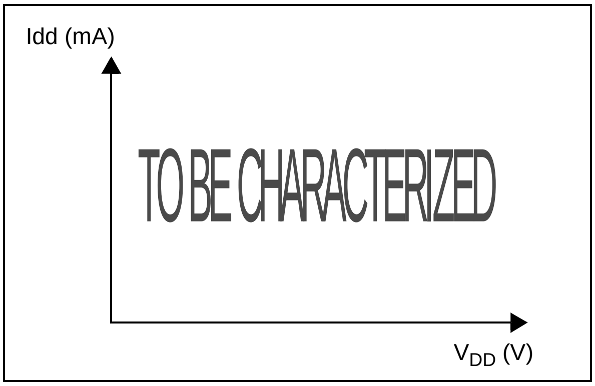


Figure 52. Typ. I_{DD} in WAIT at 4 and 8 MHz f_{CPU}



12.5 CLOCK AND TIMING CHARACTERISTICS

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A.

12.5.1 General Timings

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
$t_{c(INST)}$	Instruction cycle time	f_{CPU}=8MHz	2	3	12	t_{CPU}
			250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time [2] $t_{v(IT)} = \Delta t_{c(INST)} + 10\ t_{CPU}$	f_{CPU}=8MHz	10		22	t_{CPU}
			1.25		2.75	µs

1. Data based on typical application software.

2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

12.5.2 CONTROL TIMING CHARACTERISTICS

CONTROL TIMINGS						
Symbol	Parameter	Conditions	Value			Unit
			Min	Typ.	Max	
f_{OSC}	Oscillator Frequency				12	MHz
f_{CPU}	Operating Frequency				8	MHz
t_{RL}	External RESET Input pulse Width		1.5			t_{CPU}
t_{PORL}	Internal Power Reset Duration		514			t_{CPU}
T_{RSTL}	Reset Pin Output Pulse Width		10			µs
t_{WDG}	Watchdog Time-out		65536		4194304	t_{CPU}
		f_{cpu} = 8MHz	8.192		524.288	ms
t_{OXOV}	Crystal Oscillator Start-up Time		20	30	40	ms
t_{DDR}	Power up rise time	from V_{DD} = 0 to 4V			100	ms

Note 1: The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 cycles.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

12.5.3 External Clock Source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSCINH}	OSCIN input pin high level voltage	see Figure 53	$0.7xV_{DD}$		V_{DD}	V
V_{OSCINL}	OSCIN input pin low level voltage		V_{SS}		$0.3xV_{DD}$	
$t_{w(OSCINH)}$ $t_{w(OSCINL)}$	OSCIN high or low time [1)]		15			ns
$t_{r(OSCIN)}$ $t_{f(OSCIN)}$	OSCIN rise or fall time[1)]				15	
I_L	OSCx Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			±1	µA

Figure 53. Typical Application with an External Clock Source

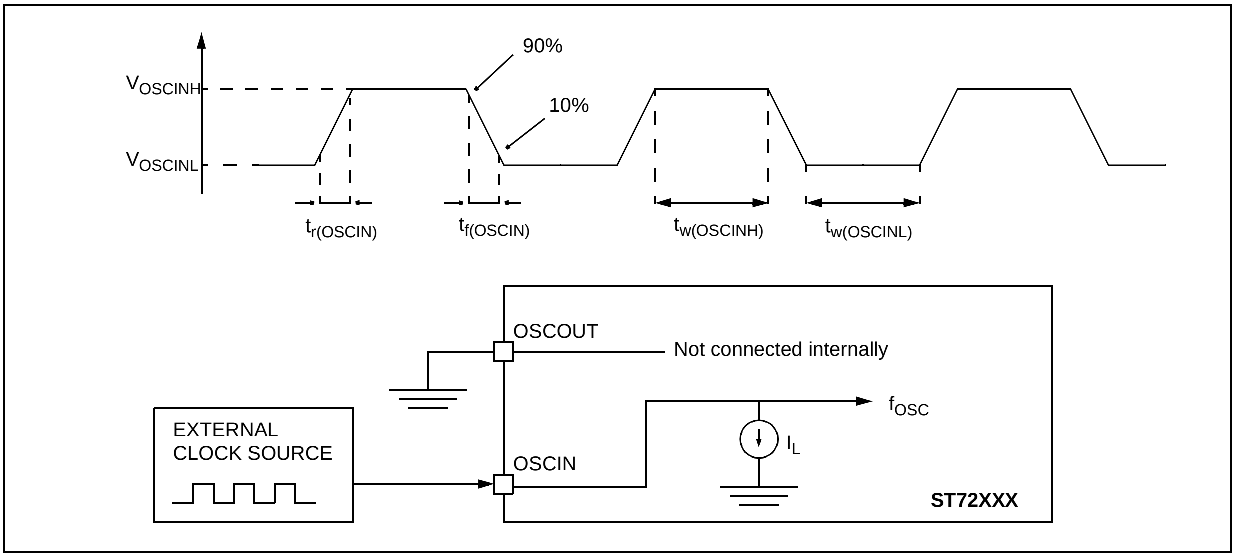
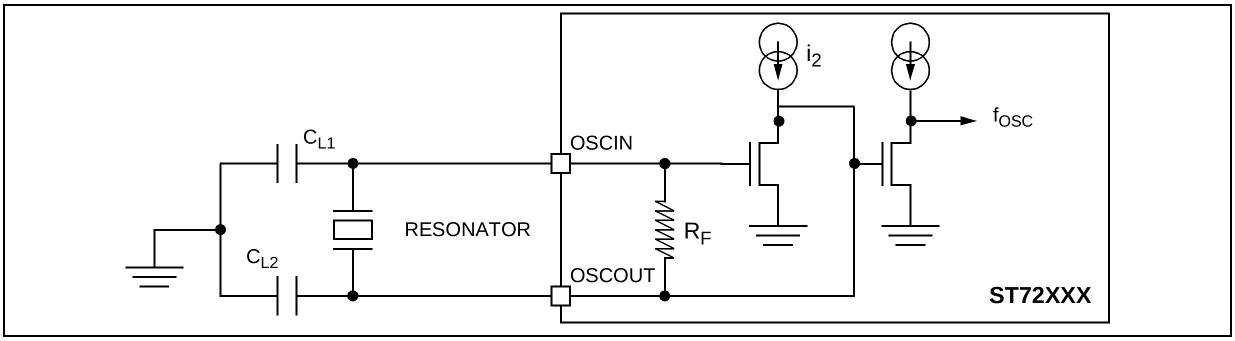

90%
10%
V_{OSCINH}
V_{OSCINL}
$t_{r(OSCIN)}$
$t_{f(OSCIN)}$
$t_{w(OSCINH)}$
$t_{w(OSCINL)}$
OSCOUT
Not connected internally
f_{OSC}
I_L
EXTERNAL CLOCK SOURCE
OSCIN
ST72XXX

Figure 54. Typical Application with a Crystal Resonator


i_2
f_{OSC}
C_{L1}
OSCIN
RESONATOR
R_F
C_{L2}
OSCOUT
ST72XXX

12.6 MEMORY CHARACTERISTICS

Subject to general operating conditions for f_{CPU}, and T_A unless otherwise specified.

12.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	HALT mode (or RESET)	2.0			V

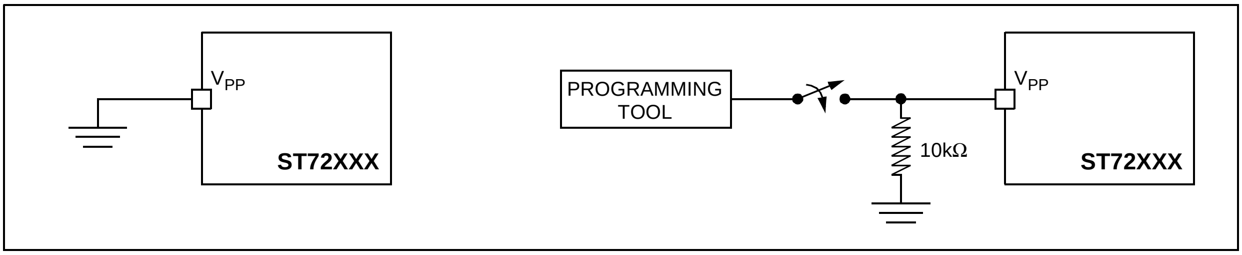
Note 1: Guaranteed by design. Not tested in production.

12.6.2 FLASH Memory

Operating Conditions: f_{CPU} = 8 MHz.

DUAL VOLTAGE FLASH MEMORY						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
f_{CPU}	Operating Frequency	Read mode			8	MHz
		Write / Erase mode			8	
V_{PP}	Programming Voltage	4.0V <= V_{DD} <= 5.5V	11.4		12.6	V
I_{PP}	V_{PP} Current	Write / Erase			30	mA
t_{PROG}	Byte Programming Time			100	TBD	µs
t_{ERASE}	Sector Erasing Time			0.5	TBD	sec
	Device Erasing Time			2	10	
		T_A=25°C			3	
t_{VPP}	Internal V_{PP} Stabilization Time			10		µs
t_{RET}	Data Retention	$T_A \leq 55°C$	20			years
N_{RW}	Write Erase Cycles		100			cycles

Figure 55. Two typical Applications with V_{PP} Pin[1]



Note 1: When the ICP mode is not required by the application, V_{PP} pin must be tied to V_{SS}.

12.7 EMC CHARACTERISTICS

Susceptibility tests are performed on a sample basis during product characterization.

12.7.1 Functional EMS

(Electro Magnetic Susceptibility)

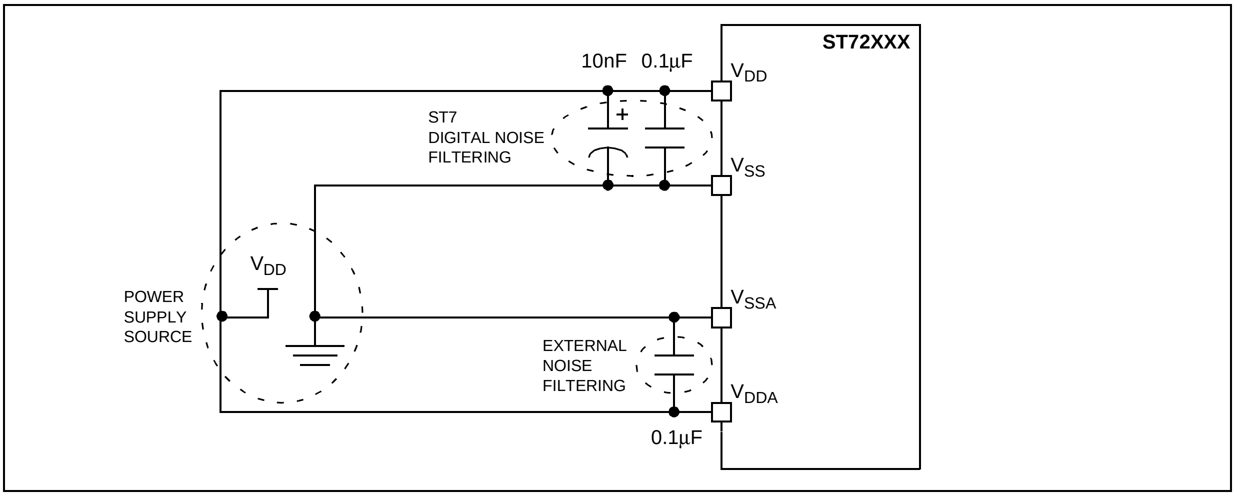
Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electro magnetic events until a failure occurs (indicated by the LEDs).

- ESD: Electro-Static Discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- FTB: A Burst of Fast Transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed.

Symbol	Parameter	Conditions	Neg 1)	Pos 1)	Unit
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD}=5V, T_A=+25°C, f_{OSC}=8MHz conforms to IEC 1000-4-2	TBD	TBD	kV
V_{FFTB}	Fast transient voltage burst limits to be applied through 100pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD}=5V, T_A=+25°C, f_{OSC}=8MHz conforms to IEC 1000-4-4	TBD	TBD	

Figure 56. EMC Recommended star network power supply connection 2)



Notes:

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

2. The suggested 10nF and 0.1µF decoupling capacitors on the power supply lines are proposed as a good price vs. EMC performance trade-off. They have to be put as close as possible to the device power supply pins. Other EMC recommendations are given in other sections (I/Os, RESET, OSCx pin characteristics).

EMC CHARACTERISTICS (Cont'd)

12.7.2 Absolute Electrical Sensitivity

Based on three different tests (ESD, LU and DLU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the AN1181 ST7 application note.

12.7.2.1 Electro-Static Discharge (ESD)

Electro-Static Discharges (1 positive then 1 negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends of the number of supply pins of the device (3 parts*(n+1) supply pin). The Human Body Model is simulated. This test conforms to the JESD22-A114A standard. See Figure 57 and the following test sequences.

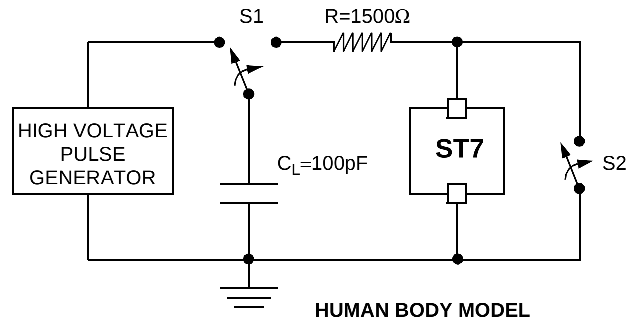
Human Body Model Test Sequence

- C_L is loaded through S1 by the HV pulse generator.
- S1 switches position from generator to R.
- A discharge from C_L through R (body resistance) to the ST7 occurs.
- S2 must be closed 10 to 100ms after the pulse delivery period to ensure the ST7 is not left in charge state. S2 must be opened at least 10ms prior to the delivery of the next pulse.

Absolute Maximum Ratings

Symbol	Ratings	Conditions	Maximum value [1]	Unit
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human Body Model)	T_A=+25°C	2000	V

Figure 57. Typical Equivalent ESD Circuits



Notes:

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

EMC CHARACTERISTICS (Cont'd)

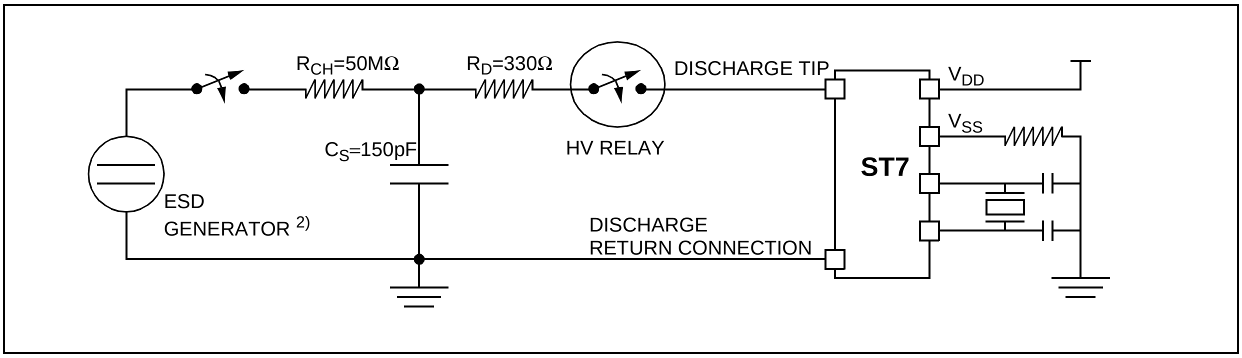
12.7.2.2 Static and Dynamic Latch-Up

- LU: 3 complementary static tests are required on 10 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin), a current injection (applied to each input, output and configurable I/O pin) and a power supply switch sequence are performed on each sample. This test conforms to the EIA/ JESD 78 IC latch-up standard. For more details, refer to the AN1181 ST7 application note.
- DLU: Electro-Static Discharges (one positive then one negative test) are applied to each pin of 3 samples when the micro is running to assess the latch-up performance in dynamic mode. Power supplies are set to the typical values, the oscillator is connected as near as possible to the pins of the micro and the component is put in reset mode. This test conforms to the IEC1000-4-2 and SAEJ1752/3 standards and is described in Figure 58. For more details, refer to the AN1181 ST7 application note.

Electrical Sensitivities

Symbol	Parameter	Conditions	Class [1]
LU	Static latch-up class	T_A=+25°C	A
DLU	Dynamic latch-up class	V_{DD}=5.5V, f_{OSC}=4MHz, T_A=+25°C	A

Figure 58. Simplified Diagram of the ESD Generator for DLU



Notes:

1. Class description: A Class is an STMicroelectronics internal specification. All its limits are higher than the JEDEC specifications, that means when a device belongs to Class A it exceeds the JEDEC standard. B Class strictly covers all the JEDEC criteria (international standard).
2. Schaffner NSG435 with a pointed test finger.

EMC CHARACTERISTICS (Cont'd)

12.7.3 ESD Pin Protection Strategy

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

An ESD network combines the different input and output ESD protections. This network works, by allowing safe discharge paths for the pins subjected to ESD stress. Two critical ESD stress cases are presented in Figure 59 and Figure 60 for standard pins and in Figure 61 and Figure 62 for true open drain pins.

Standard Pin Protection

To protect the output structure the following elements are added:

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

To protect the input structure the following elements are added:

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

Figure 59. Positive Stress on a Standard Pad vs. V_{SS}

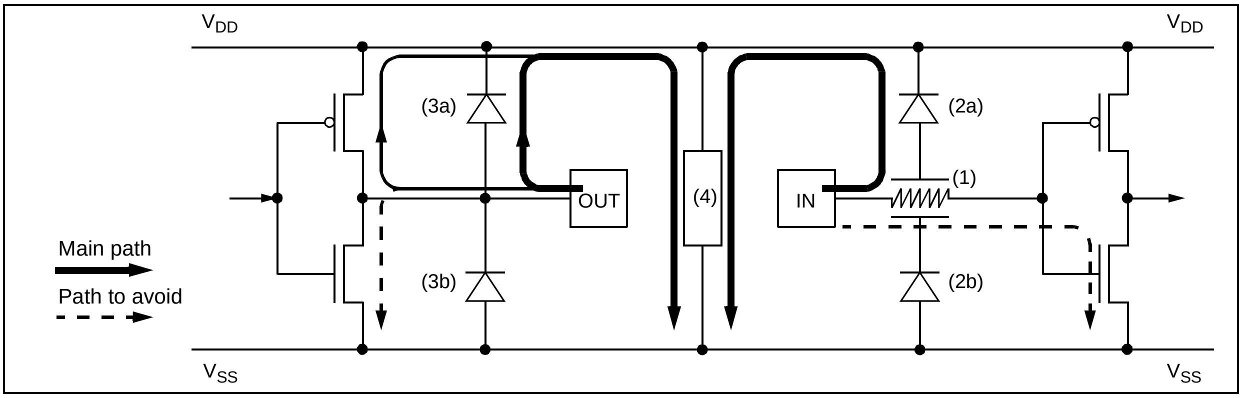
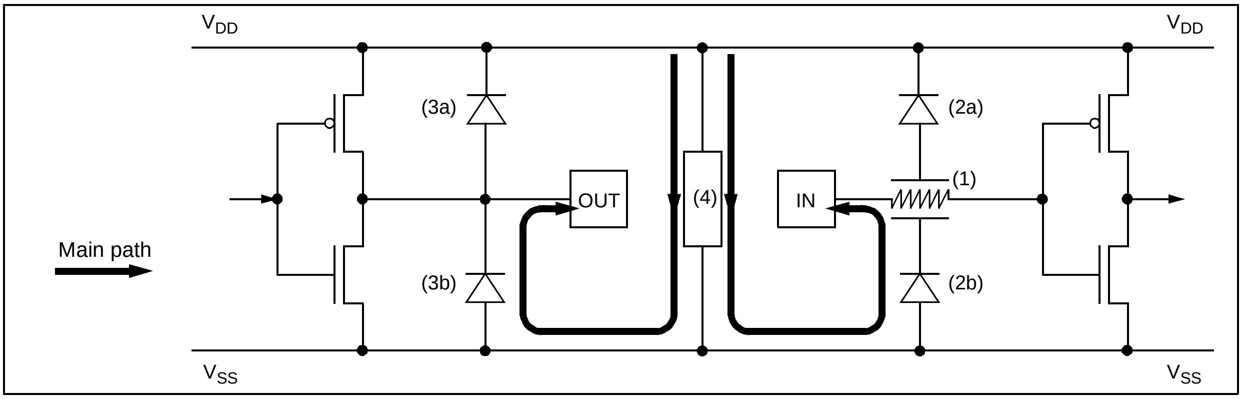


Figure 60. Negative Stress on a Standard Pad vs. V_{DD}



EMC CHARACTERISTICS (Cont'd)

True Open Drain Pin Protection

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

Multisupply Configuration

When several types of ground (V_{SS}, V_{SSA}, ...) and power supply (V_{DD}, V_{DDA}, ...) are available for any reason (better noise immunity...), the structure shown in Figure 63 is implemented to protect the device against ESD.

Figure 61. Positive Stress on a True Open Drain Pad vs. V_{SS}

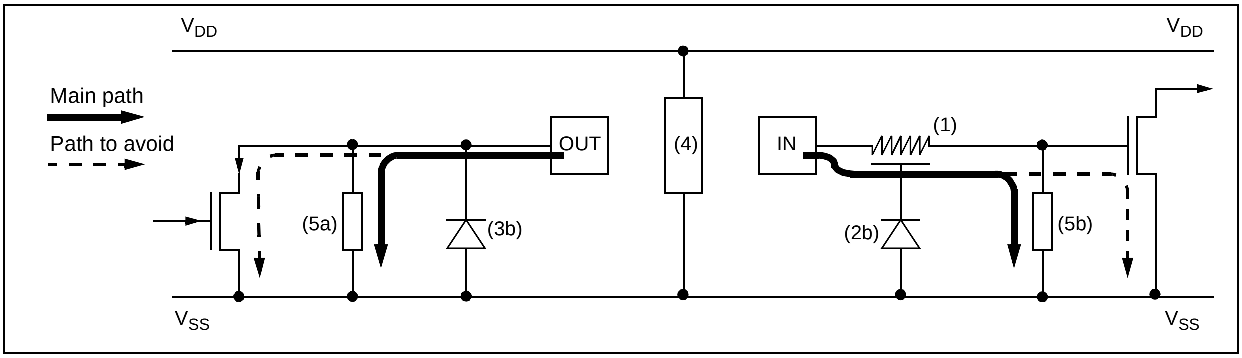

VDD
VDD
Main path
Path to avoid
OUT
(4)
IN
(1)
(5a)
(3b)
(2b)
(5b)
VSS
VSS

Figure 62. Negative Stress on a True Open Drain Pad vs. V_{DD}

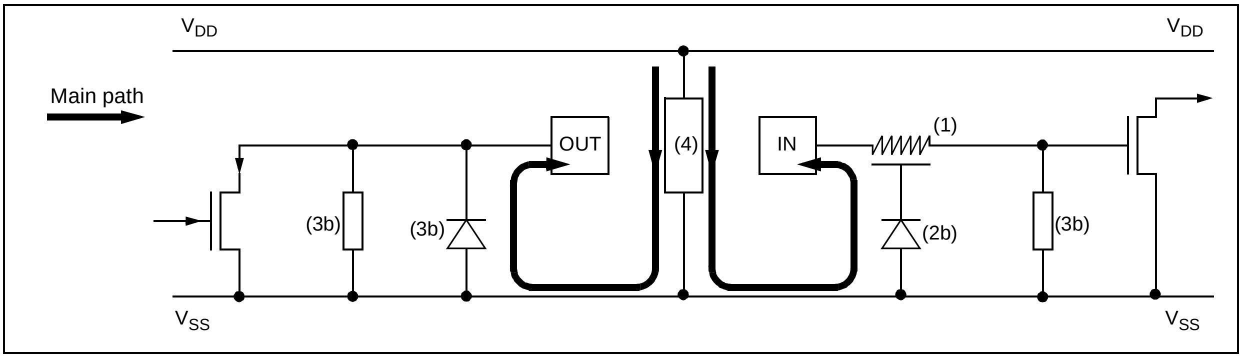
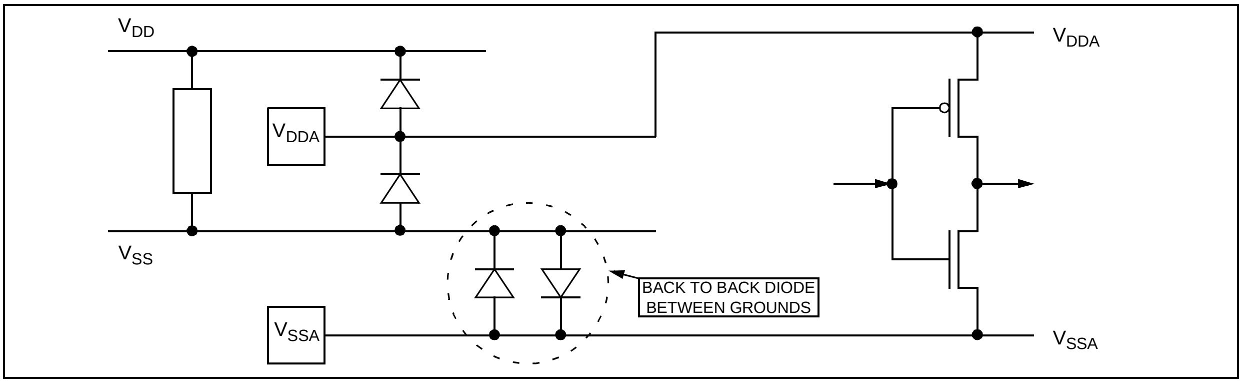

VDD
VDD
Main path
OUT
(4)
IN
(1)
(3b)
(3b)
(2b)
(3b)
VSS
VSS

Figure 63. Multisupply Configuration


VDD
VDDA
VDDA
VSS
BACK TO BACK DIODE BETWEEN GROUNDS
VSSA
VSSA

12.8 I/O PORT PIN CHARACTERISTICS

12.8.1 General Characteristics

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

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage				$0.3xV_{DD}$	V
V_{IH}	Input high level voltage		$0.7xV_{DD}$			
V_{IN}	Input voltage	True Open Drain I/O pins	V_{SS}		6.0	V
		Other I/O pins			V_{DD}	
V_{hys}	Schmitt trigger voltage hysteresis			400		mV
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			±1	µA
I_S	Static current consumption [2]	Floating input mode			200	
R_{PU}	Weak pull-up equivalent resistor [3]	$V_{IN}=V_{SS}$ V_{DD}=5V	50	80	120	kΩ
C_{IO}	I/O pin capacitance			5		pF
$t_{f(IO)out}$	Output high to low level fall time	C_L=50pF Between 10% and 90%		25		ns
$t_{r(IO)out}$	Output low to high level rise time			25		
$t_{w(IT)in}$	External interrupt pulse time [4]		1			t_{CPU}

Figure 64. Two typical Applications with unused I/O Pin

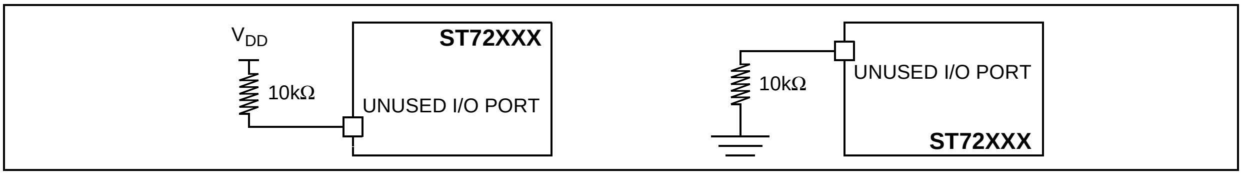


Figure 65. Typical I_{PU} vs. V_{DD} with $V_{IN}=V_{SS}$

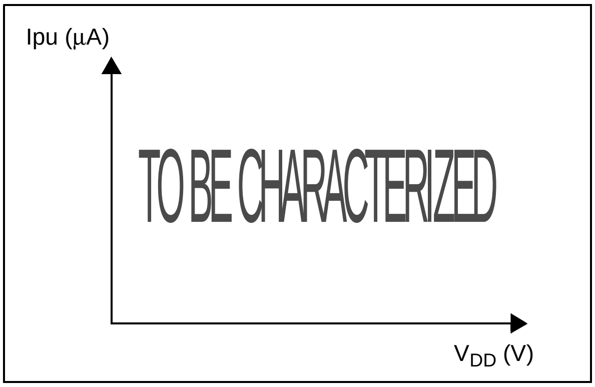
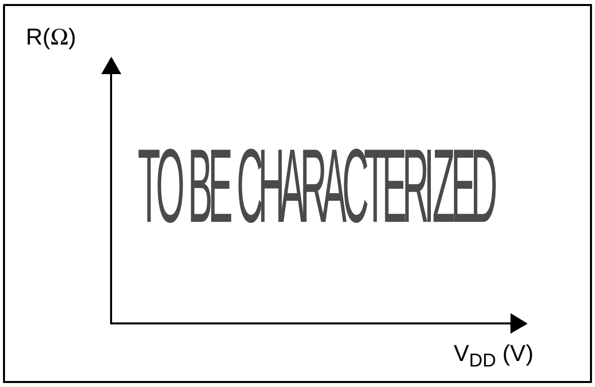


Figure 66. Typical R_{PU} vs. V_{DD} with $V_{IN}=V_{SS}$



Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V, not tested in production.
2. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see Figure 64). Data based on design simulation and/or technology characteristics, not tested in production.
3. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in Figure 65). This data is based on characterization results.
4. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

I/O PORT PIN CHARACTERISTICS (Cont'd)

12.8.2 Output Driving Current

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

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} 1)	Output low level voltage for a standard I/O pin when up to 8 pins are sunk at same time (see Figure 67)	V_{DD}=5V	I_{IO}=+5mA		1.3	V
			I_{IO}=+2mA		0.4	
	Output low level voltage for a high sink I/O pin when up to4 pins are sunk at same time (see Figure 68)		I_{IO}=+20mA		1.3	
			I_{IO}=+8mA		0.4	
V_{OH} 2)	Output high level voltage for an I/O pin when up to 8 pins are sourced at same time (see Figure 69)		I_{IO}=-5mA	V_{DD}-2.0		
			I_{IO}=-2mA	V_{DD}-0.8		

Figure 67. Typical V_{OL} at V_{DD}=5V (standard)

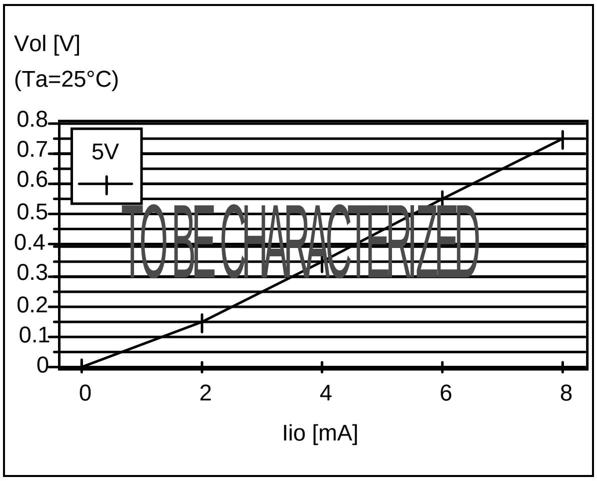


Figure 69. Typical V_{DD}-V_{OH} at V_{DD}=5V

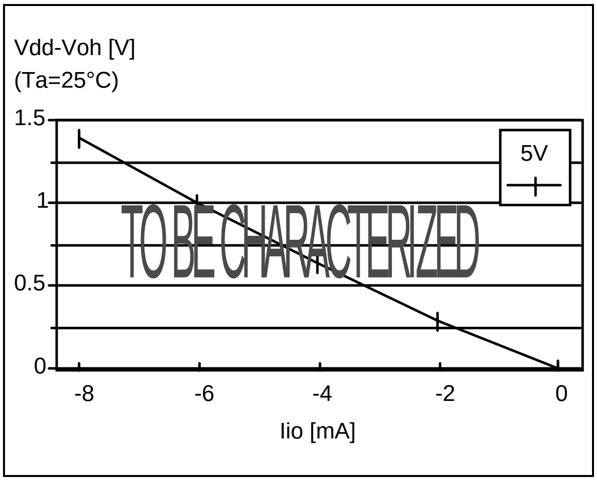
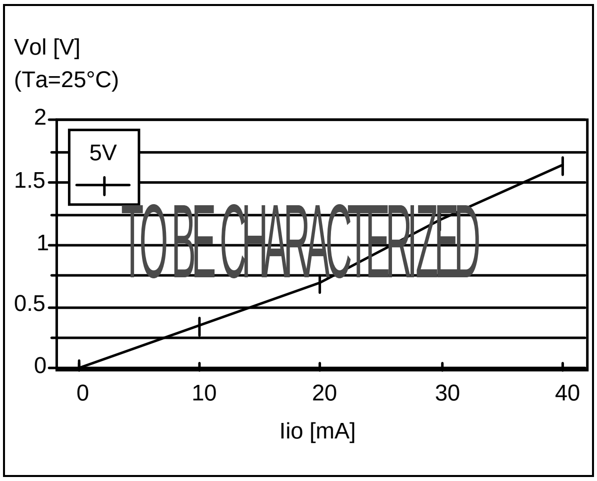


Figure 68. Typical V_{OL} at V_{DD}=5V (high-sink)



Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

2. The I_{IO} current sourced must always respect the absolute maximum rating specified in Section 12.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 70. Typical V_{OL} vs. V_{DD} (standard I/Os)

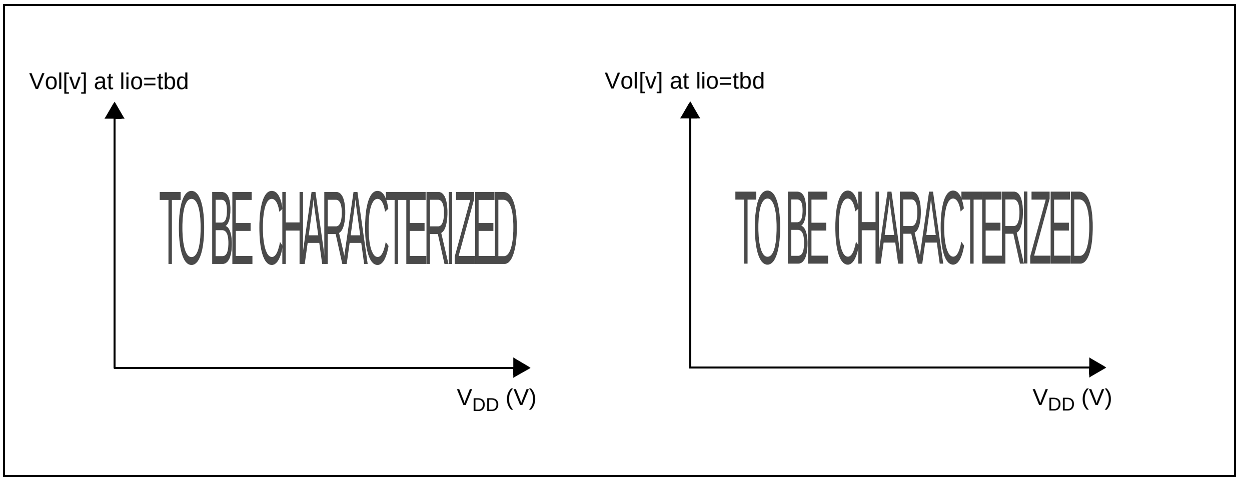


Figure 71. Typical V_{OL} vs. V_{DD} (high-sink I/Os)

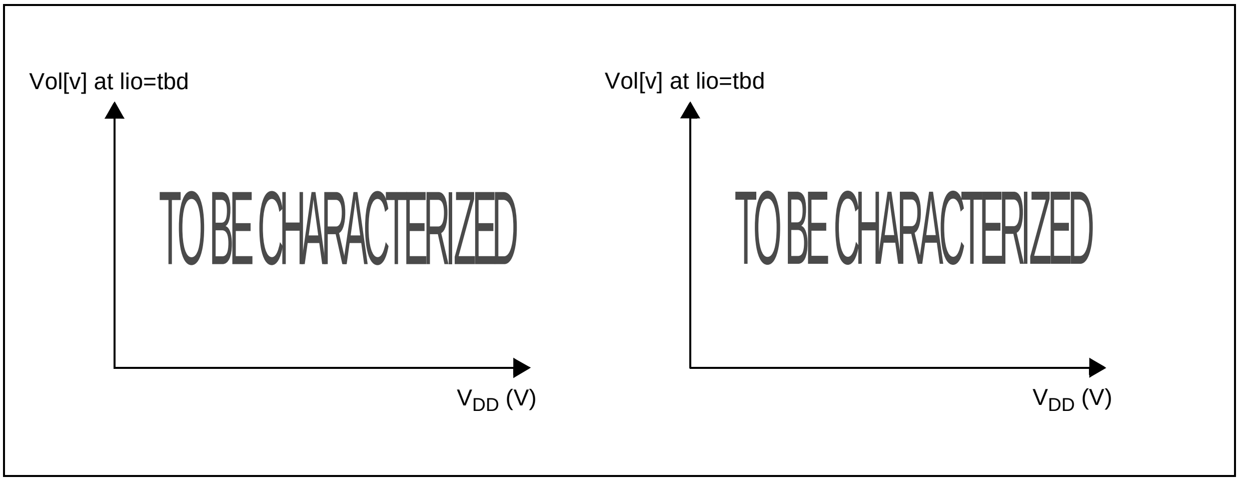
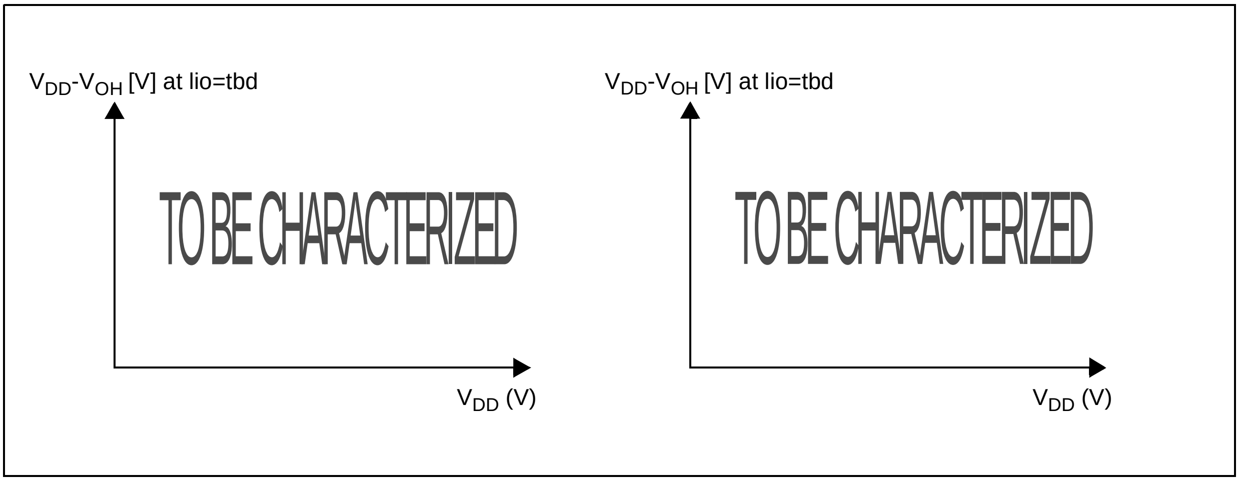


Figure 72. Typical V_{DD}-V_{OH} vs. V_{DD}



12.9 CONTROL PIN CHARACTERISTICS

12.9.1 Asynchronous $\overline{\text{RESET}}$ Pin

Subject to general operating conditions for V_{DD}, $f_{CPÜ}$, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ 1)	Max	Unit
V_{IH}	Input High Level Voltage			0.7xV_{DD}		V_{DD}	V
V_{IL}	Input Low Voltage			V_{SS}		0.3xV_{DD}	V
V_{hys}	Schmitt trigger voltage hysteresis 3)				400		mV
V_{OL}	Output low level voltage 4) (see Figure 74, Figure 75)	V_{DD}=5V	I_{IO}=TBD			TBD	V
			I_{IO}=TBD			TBD	
R_{ON}	Weak pull-up equivalent resistor 5)	V_{IN}=V_{SS}	V_{DD}=5V	TBD	40	TBD	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or internal reset sources			6 30		1/f_{SFOSC} µs
$t_{h(RSTL)in}$	External reset pulse hold time 6)			10			µs

Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V, not tested in production.
2. Data based on characterization results, not tested in production.
3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
4. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
5. The R_{ON} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{ON} current characteristics described in Figure 73). This data is based on characterization results, not tested in production.
6. To guarantee the reset of the device, a minimum pulse has to be applied to $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

CONTROL PIN CHARACTERISTICS (Cont'd)

Figure 73. Typical I_{ON} vs. V_{DD} with V_{IN}=V_{SS}

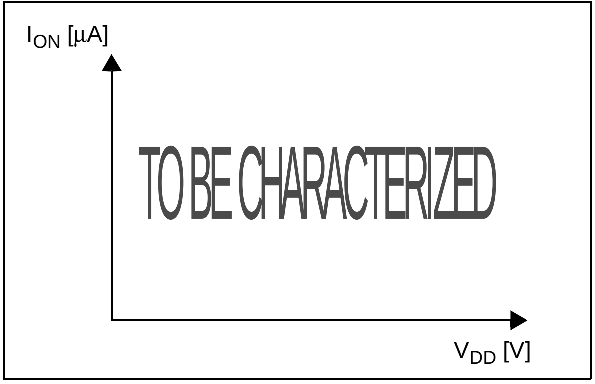


Figure 74. Typical V_{OL} at V_{DD}=5V ($\overline{\text{RESET}}$)

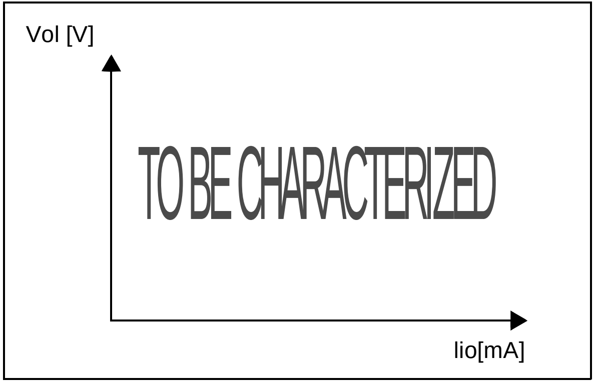
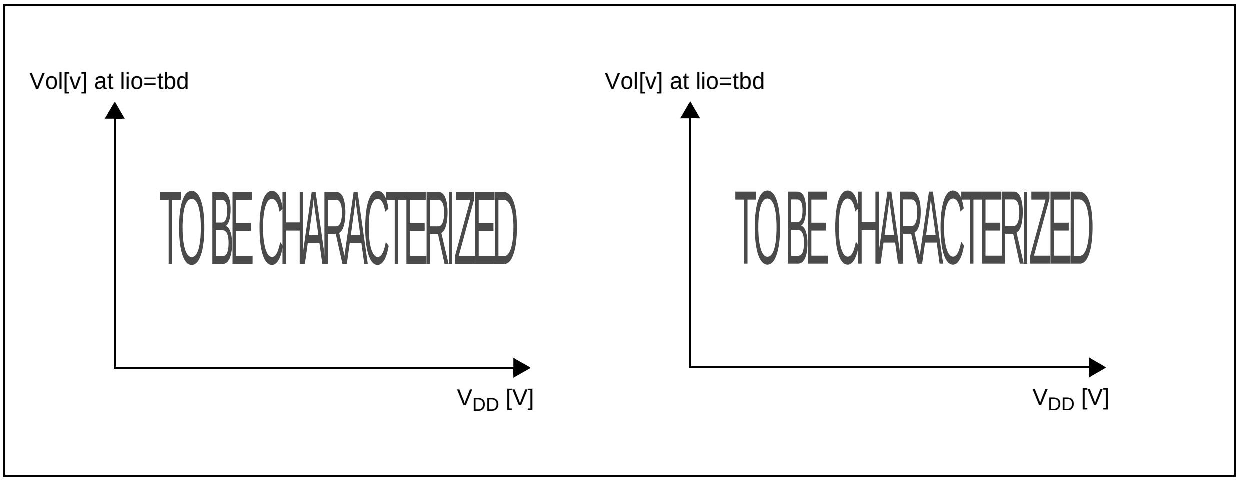


Figure 75. Typical V_{OL} vs. V_{DD} ($\overline{\text{RESET}}$)



12.10 TIMER PERIPHERAL CHARACTERISTICS

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (output compare, input capture, external clock, PWM output...).

12.10.1 8-Bit PWM-ART Auto-Reload Timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{res(PWM)}$	PWM resolution time		1			t_{CPU}
		f_{CPU}=8MHz	125			ns
f_{EXT}	ART external clock frequency		0		$f_{CPU}/2$	MHz
f_{PWM}	PWM repetition rate		0		$f_{CPU}/2$	
Res_{PWM}	PWM resolution				8	bit
V_{OS}	PWM/DAC output step voltage	V_{DD}=5V, Res=8-bits		20		mV

12.11 COMMUNICATION INTERFACE CHARACTERISTICS

12.11.1 USB - Universal Bus Interface

(Operating conditions T_A = 0 to +70°C, V_{DD} = 4.0 to 5.25V unless otherwise specified)

USB DC Electrical Characteristics					
Parameter	**Symbol**	**Conditions**	**Min.**	**Max.**	**Unit**
Differential Input Sensitivity	VDI	I(D+, D-)	0.2		V
Differential Common Mode Range	VCM	Includes VDI range	0.8	2.5	V
Single Ended Receiver Threshold	VSE		0.8	2.0	V
Static Output Low	VOL	RL of 1.5K ohms to 3.6v		0.3	V
Static Output High	VOH	RL of 15K ohms to V_{SS}	2.8	3.6	V
USBVCC: voltage level	USBV	V_{DD}=5v	3.00	3.60	V

Note 1: RL is the load connected on the USB drivers.

Note 2: All the voltages are measured from the local ground potential.

Figure 76. USB: Data Signal Rise and Fall Time

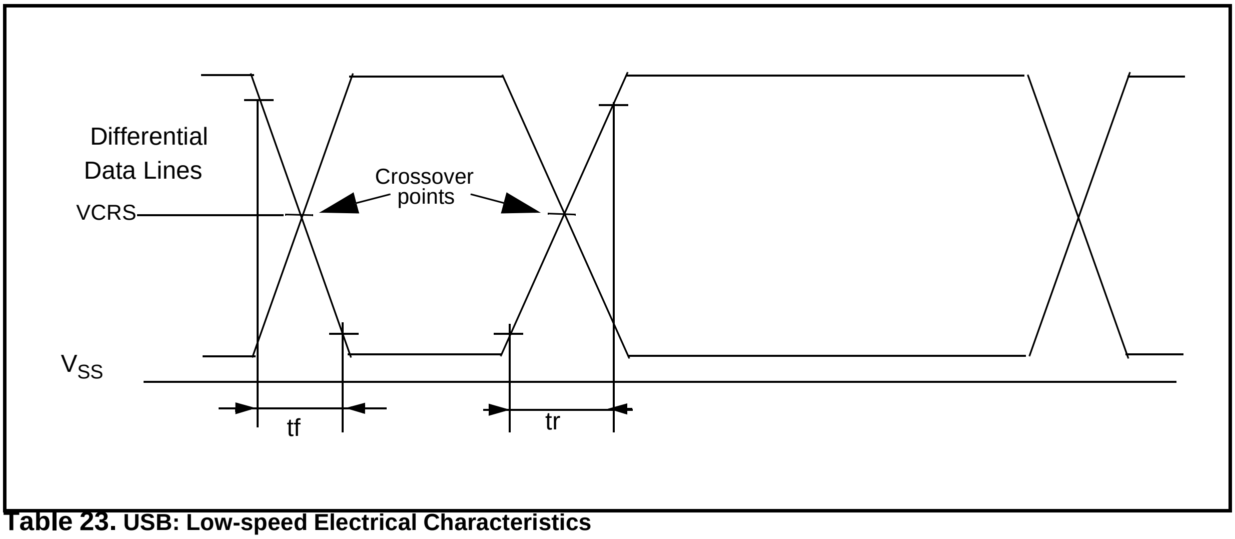


Table 23. USB: Low-speed Electrical Characteristics

Parameter	Symbol	Conditions	Min	Max	Unit
Driver characteristics:					
Rise time	tr	Note 1,CL=50 pF	75		ns
		Note 1, CL=600 pF		300	ns
Fall Time	tf	Note 1, CL=50 pF	75		ns
		Note 1, CL=600 pF		300	ns
Rise/ Fall Time matching	trfm	tr/tf	80	120	%
Output signal Crossover Voltage	VCRS		1.3	2.0	V

Note 1: Measured from 10% to 90% of the data signal. For more detailed informations, please refer to Chapter 7 (Electrical) of the USB specification (version 1.1).

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

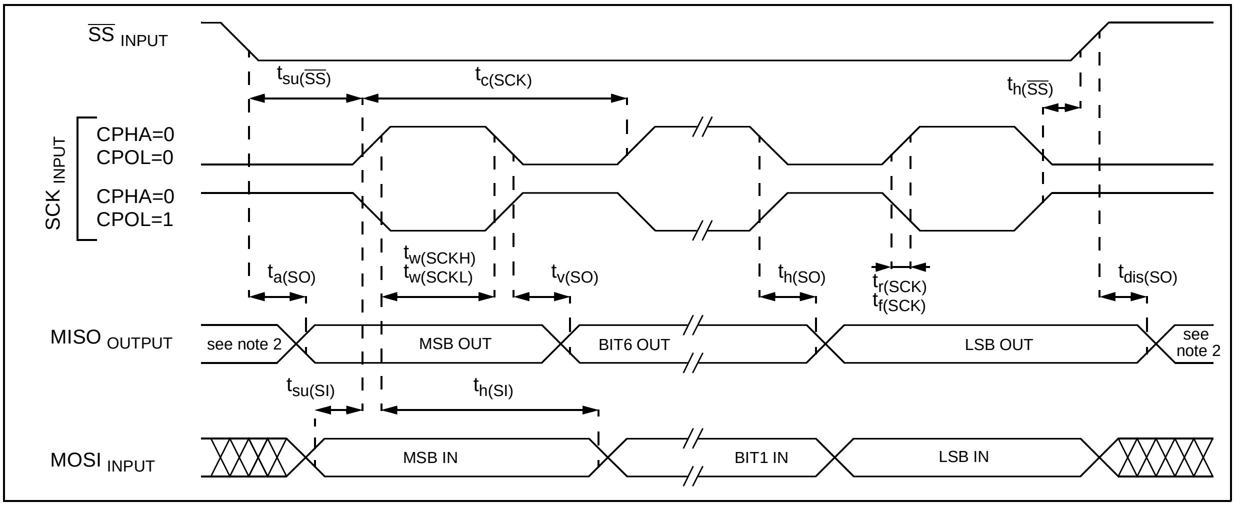
12.11.2 SPI - Serial Peripheral Interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU}=8MHz	$f_{CPU}/128$ 0.0625	$f_{CPU}/4$ 2	MHz
		Slave f_{CPU}=8MHz	0	$f_{CPU}/2$ 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$	$\overline{SS}$ setup time	Slave	120		ns
$t_{h(\overline{SS})}$	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$	Data output access time	Slave	0	120	
$t_{dis(SO)}$	Data output disable time	Slave		240	
$t_{v(SO)}$	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$	Data output hold time		0		
$t_{v(MO)}$	Data output valid time	Master (before capture edge)	0.25		t_{CPU}
$t_{h(MO)}$	Data output hold time		0.25		

Figure 77. SPI Slave Timing Diagram with CPHA=0 3)


$\overline{SS}$ INPUT
$t_{su(\overline{SS})}$
$t_{c(SCK)}$
$t_{h(\overline{SS})}$
SCK INPUT
CPHA=0
CPOL=0
CPHA=0
CPOL=1
$t_{a(SO)}$
$t_{w(SCKH)}$
$t_{w(SCKL)}$
$t_{v(SO)}$
$t_{h(SO)}$
$t_{r(SCK)}$
$t_{f(SCK)}$
$t_{dis(SO)}$
MISO OUTPUT
see note 2
MSB OUT
BIT6 OUT
LSB OUT
see note 2
$t_{su(SI)}$
$t_{h(SI)}$
MOSI INPUT
MSB IN
BIT1 IN
LSB IN

Notes:

1. Data based on design simulation and/or characterisation results, not tested in production.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.
3. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

Figure 78. SPI Slave Timing Diagram with CPHA=1[1)]

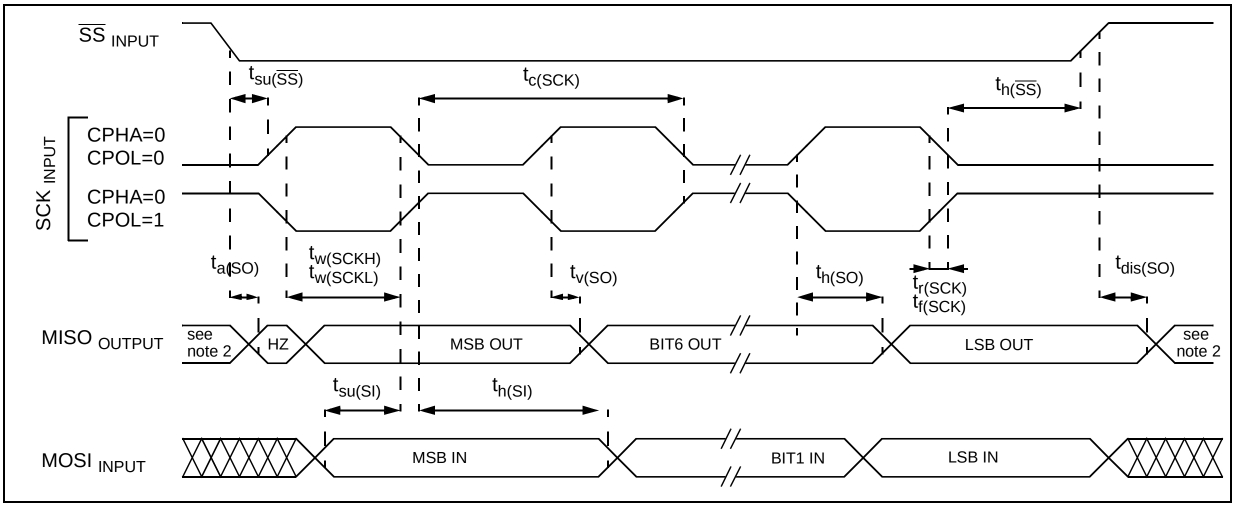
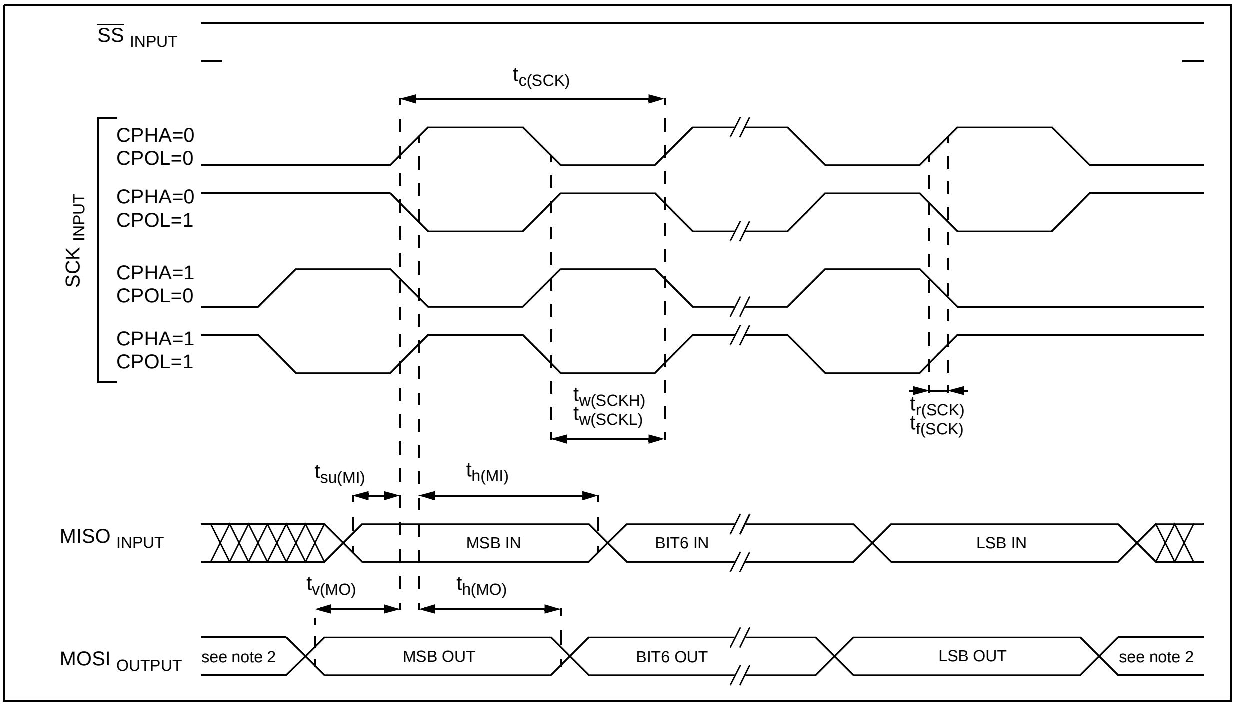

SS INPUT
tsu(SS)
tc(SCK)
th(SS)
SCK INPUT
CPHA=0
CPOL=0
CPHA=0
CPOL=1
ta(SO)
tw(SCKH)
tw(SCKL)
tv(SO)
th(SO)
tr(SCK)
tf(SCK)
tdis(SO)
MISO OUTPUT
see note 2
HZ
MSB OUT
BIT6 OUT
LSB OUT
see note 2
tsu(SI)
th(SI)
MOSI INPUT
MSB IN
BIT1 IN
LSB IN

Figure 79. SPI Master Timing Diagram [1)]


SS INPUT
tc(SCK)
SCK INPUT
CPHA=0
CPOL=0
CPHA=0
CPOL=1
CPHA=1
CPOL=0
CPHA=1
CPOL=1
tw(SCKH)
tw(SCKL)
tr(SCK)
tf(SCK)
tsu(MI)
th(MI)
MISO INPUT
MSB IN
BIT6 IN
LSB IN
tv(MO)
th(MO)
MOSI OUTPUT
see note 2
MSB OUT
BIT6 OUT
LSB OUT
see note 2

Notes:

1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)

12.11.3 SCI - Serial Communications Interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics (RDI and TDO).

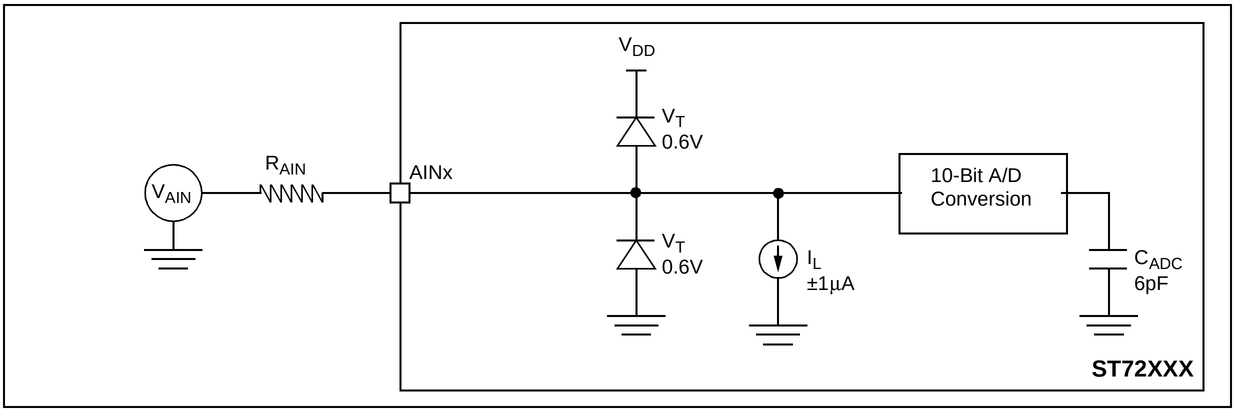
Symbol	Parameter	Conditions			Standard	Baud Rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8MHz	~0.16%	Conventional Mode			Hz
				TR (or RR)=128, PR=13	300	~300.48	
				TR (or RR)= 32, PR=13	1200	~1201.92	
				TR (or RR)= 16, PR=13	2400	~2403.84	
				TR (or RR)= 8, PR=13	4800	~4807.69	
				TR (or RR)= 4, PR=13	9600	~9615.38	
				TR (or RR)= 16, PR= 3	10400	~10416.67	
				TR (or RR)= 2, PR=13	19200	~19230.77	
				TR (or RR)= 1, PR=13	38400	~38461.54	
			~0.79%	Extended Mode ETPR (or ERPR) = 35, TR (or RR)= 1, PR=1	14400	~14285.71	

12.12 10-BIT ADC CHARACTERISTICS

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

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion voltage range [2]		V_{SSA}		V_{DDA}	V
R_{AIN}	External input impedance				10 [3]	kΩ
C_{ADC}	Internal sample and hold capacitor			6		pF
t_{CONV}	Conversion time	f_{ADC}=4MHz	28			µs
			112			$1/f_{ADC}$
I_{ADC}	Analog Part	Sunk on V_{DDA} [2]			3.6	mA
	Digital Part	Sunk on V_{DD}			0.2	

Figure 80. Typical Application with ADC



Notes:

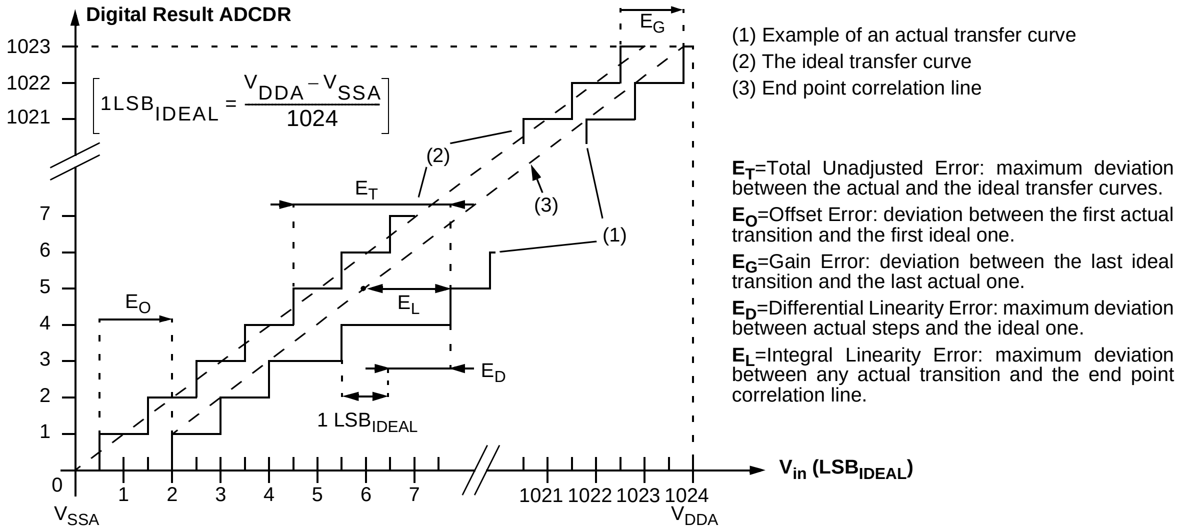
1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.
2. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refers to V_{DD} and V_{SS}.
3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.

ADC CHARACTERISTICS (Cont'd)

ADC Accuracy with f_{CPU}=8 MHz, f_{ADC}=4 MHz R_{AIN}< 10kΩ

Symbol	Parameter	Conditions	Typ	Max	Unit
$\lvert E_T \rvert$	Total unadjusted error [1)]	V_{DD}=5V	2	±4	LSB
		V_{DD}=3V		TBD	
E_O	Offset error [1)]	V_{DD}=5V		TBD	
		V_{DD}=3V		TBD	
E_G	Gain Error [1)]	V_{DD}=5V		TBD	
		V_{DD}=3V		TBD	
$\lvert E_D \rvert$	Differential linearity error[1)]	V_{DD}=5V	1.5	±3	
		V_{DD}=3V		TBD	
$\lvert E_L \rvert$	Integral linearity error [1)]	V_{DD}=5V		TBD	
		V_{DD}=3V		TBD	

Figure 81. ADC Accuracy Characteristics


Digital Result ADCDR
1023
1022
1021
7
6
5
4
3
2
1
0
V_{SSA}
$\left[1LSB_{IDEAL} = \frac{V_{DDA} - V_{SSA}}{1024}\right]$
E_G
E_T
E_O
E_L
E_D
(1)
(2)
(3)
1 LSB_{IDEAL}
1
2
3
4
5
6
7
1021
1022
1023
1024
V_{DDA}
V_{in} (LSB_{IDEAL})

(1) Example of an actual transfer curve
(2) The ideal transfer curve
(3) End point correlation line

E_T=Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.

E_O=Offset Error: deviation between the first actual transition and the first ideal one.

E_G=Gain Error: deviation between the last ideal transition and the last actual one.

E_D=Differential Linearity Error: maximum deviation between actual steps and the ideal one.

E_L=Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

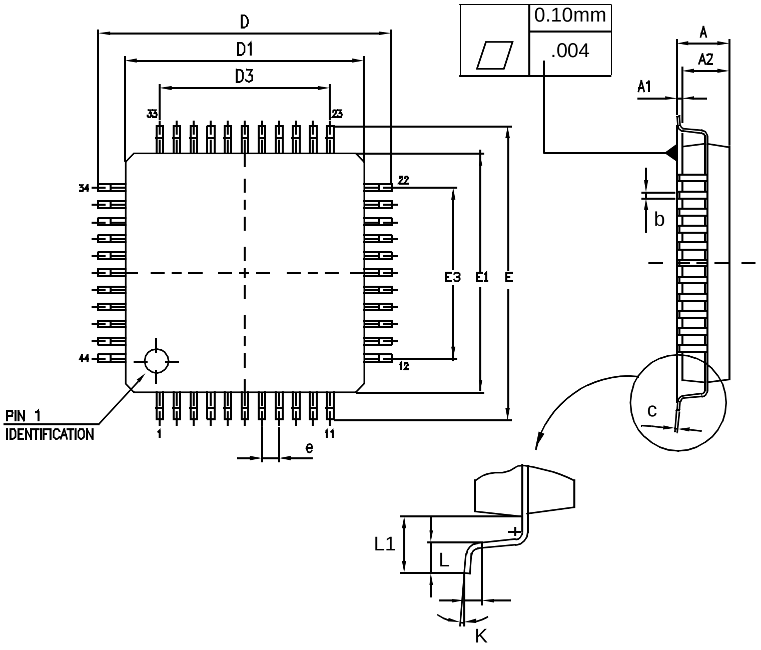
Notes:

1) 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.

13 PACKAGE CHARACTERISTICS

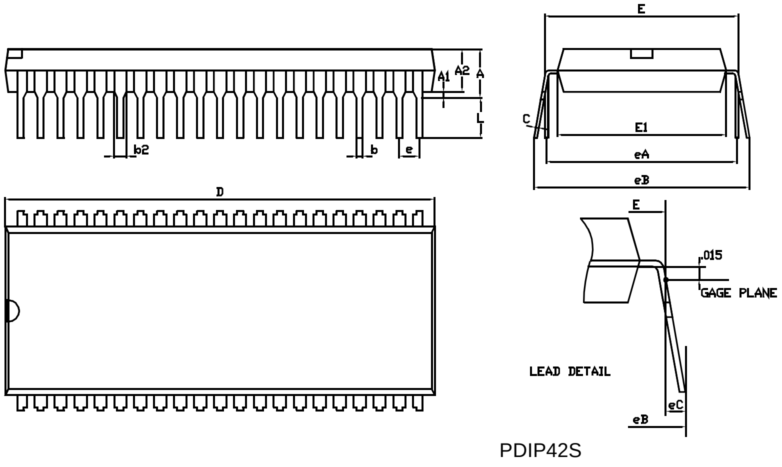
13.1 PACKAGE MECHANICAL DATA

Figure 82. 44-Pin Thin Quad Flat Package


D
D1
D3
0.10mm
.004
A
A2
A1
33
23
34
22
b
E3
E1
E
44
12
PIN 1
IDENTIFICATION
1
11
e
c
L1
L
K

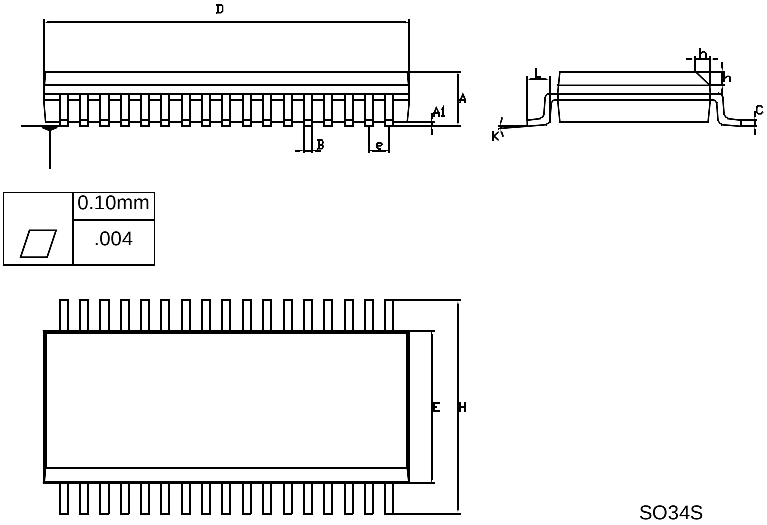
Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.015	0.018
c	0.09		0.20	0.004		0.008
D		12.00			0.472	
D1		10.00			0.394	
D3		8.00			0.315	
E		12.00			0.472	
E1		10.00			0.394	
E3		8.00			0.315	
e		0.80			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 83. 42-Pin Shrink Plastic Dual In-Line Package, 600-mil Width


A1
A2
A
L
b2
b
e
D
E
C
E1
eA
eB
E
.015
GAGE PLANE
LEAD DETAIL
eB
eC
PDIP42S

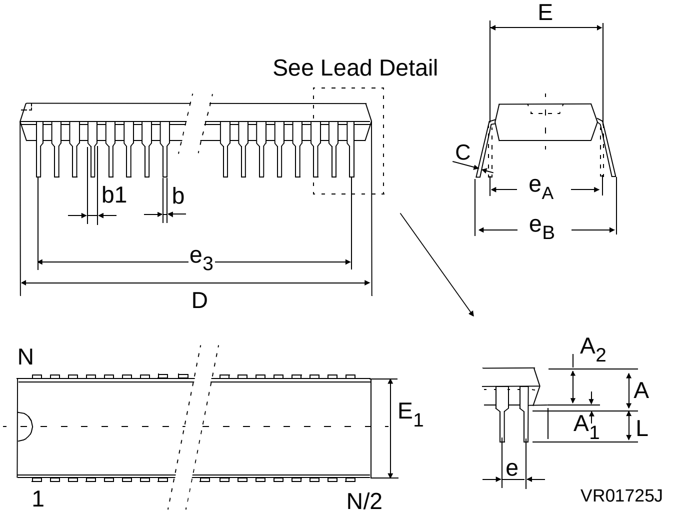
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
A2	3.05	3.81	4.57	0.120	0.150	0.180
b		0.46	0.56		0.018	0.022
b2		1.02	1.14		0.040	0.045
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.58	36.83	37.08	1.440	1.450	1.460
E	15.24		16.00	0.600		0.630
E1	12.70	13.72	14.48	0.500	0.540	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					

Figure 84. 34-Pin Plastic Small Outline Package


D
A
A1
B
e
L
h
C
K
0.10mm
.004
E
H
SO34S

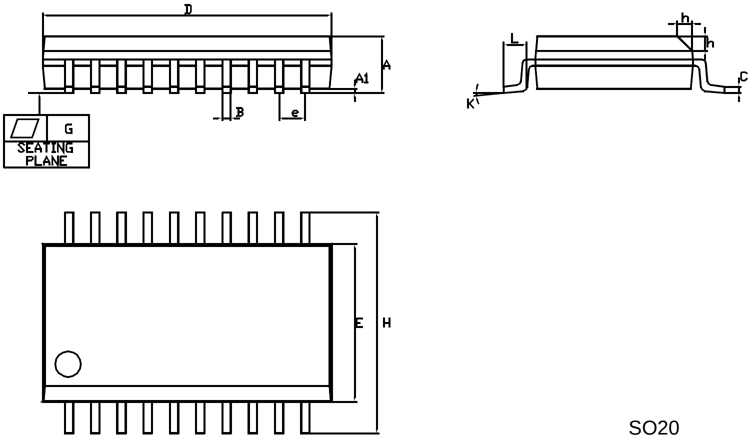
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.46		2.64	0.097		0.104
A1	0.13		0.29	0.005		0.0115
B	0.36		0.48	0.014		0.019
C	0.23		0.32	0.0091		0.0125
D	17.73		18.06	0.698		0.711
E	7.42		7.59	0.292		0.299
e		1.02			0.040	
H	10.16		10.41	0.400		0.410
h	0.64		0.74	0.025		0.029
K				0°		8°
L	0.61		1.02	0.024		0.040
	Number of Pins					
N	34					

Figure 85. 32-Pin Shrink Plastic Dual In Line Package


E
See Lead Detail
C
e_A
e_B
b1
b
e_3
D
N
E_1
1
N/2
A_2
A
A_1
L
e
VR01725J

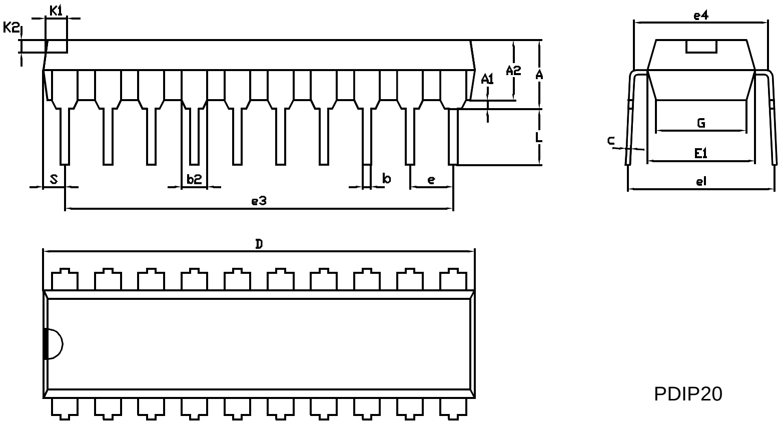
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.140	0.148	0.200
A1	0.51			0.020		
A2	3.05	3.56	4.57	0.120	0.140	0.180
b	0.36	0.46	0.58	0.014	0.018	0.023
b1	0.76	1.02	1.40	0.030	0.040	0.055
C	0.20	0.25	0.36	0.008	0.010	0.014
D	27.43	27.94	28.45	1.080	1.100	1.120
E	9.91	10.41	11.05	0.390	0.410	0.435
E1	7.62	8.89	9.40	0.300	0.350	0.370
e		1.78			0.070	
eA		10.16			0.400	
eB			12.70			0.500
L	2.54	3.05	3.81	0.100	0.120	0.150
	Number of Pins					
N	32					

Figure 86. 20-Pin Plastic Small Outline Package, 300-mil Width


D
A
A1
B
e
G
SEATING PLANE
E
H
L
h
h
K
C
SO20

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35		2.65	0.0926		0.1043
A1	0.10			0.0040		
B	0.33		0.51	0.0130		0.0200
C			0.32			0.0125
D	4.98		13.00	0.1961		0.5118
E	7.40		7.60	0.2914		0.2992
e		1.27			0.050	
H	10.01		10.64	0.394		0.419
h	0.25		0.74	0.010		0.029
K	0°		8°	0°		8°
L	0.41		1.27	0.016		0.050
G			0.10			0.004
	Number of Pins					
N	20					

Figure 87. 20-Pin Plastic Dual In-Line Package, 300-mil Width


K1
K2
A1
A2
A
L
S
b2
b
e
e3
D
e4
G
c
E1
e1
PDIP20

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.33			0.210
A2	2.92	3.30	4.95	0.115	0.130	0.195
b	0.36	0.46	0.56	0.014	0.018	0.022
b2	1.14	1.52	1.78	0.045	0.060	0.070
c	0.20	0.25	0.36	0.008	0.010	0.014
D	24.89		26.92	0.980		1.060
e		2.54			0.100	
E1	6.10	6.35	7.11	0.240	0.250	0.280
L	2.92	3.30	3.81	0.115	0.130	0.150
	Number of Pins					
N	20					

14 DEVICE CONFIGURATION AND ORDERING INFORMATION

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

FLASH devices are shipped to customers with a default content (FFh), while FASTROM 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 Byte while the ROM devices are factory-configured.

14.1 Option Byte

The Option Byte allows the hardware configuration of the microcontroller to be selected.
The Option Byte has no address in the memory map and can be accessed only in programming mode using a standard ST7 programming tool. The default contents of the FLASH is fixed to FFh. This means that all the options have “1” as their default value.
In FASTROM devices, the Option Byte is fixed in hardware by the FASTROM code.

OPTION BYTE

7							0
-	-	WDG	1	LVD	-	OSC 12/6	ROP

Bits 7:6 = Reserved.

Bit 5 = **WDG** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

Bit 4= Reserved.

Bit 3 = **LVD** *Low Voltage Detector selection*
This option bit selects the LVD.
0: LVD enabled
1: LVD disabled

Bit 2= Reserved.

Bit 1 = **OSC12/6** *Oscillator selection*
This option bit selects the clock divider used to drive the USB interface at 6MHz.
0: 6 MHz oscillator (no divider for USB)
1: 12 Mhz oscillator (2 divider for USB)

Bit 0 = **ROP** *Read out protection*
This option bit allows the protection of the software contents against piracy (program or data). When the protection is activated, read/write access is prevented by hardware. If the protection is deactivated, the memory is erased first.
0: Read-out protection enabled
1: Read-out protection disabled

14.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE

Customer code is made up of the FASTROM or ROM contents and the list of the selected options (if any). The FASTROM or 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 88. FLASH User Programmable Device Types

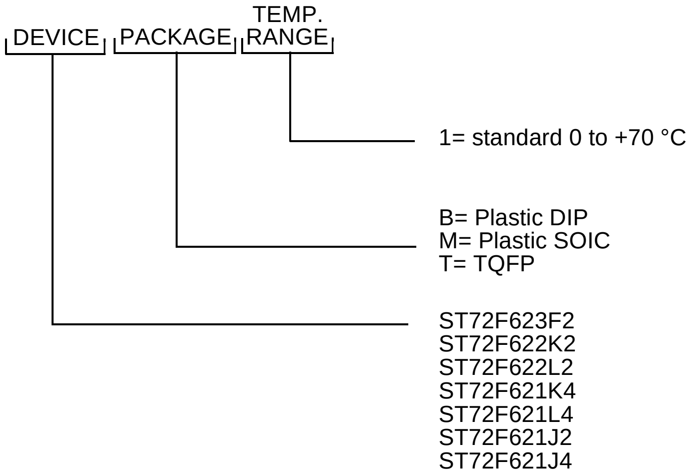

DEVICE
PACKAGE
TEMP. RANGE
1= standard 0 to +70 °C
B= Plastic DIP
M= Plastic SOIC
T= TQFP
ST72F623F2
ST72F622K2
ST72F622L2
ST72F621K4
ST72F621L4
ST72F621J2
ST72F621J4

Figure 89. FASTROM Factory-Programmed Device Types

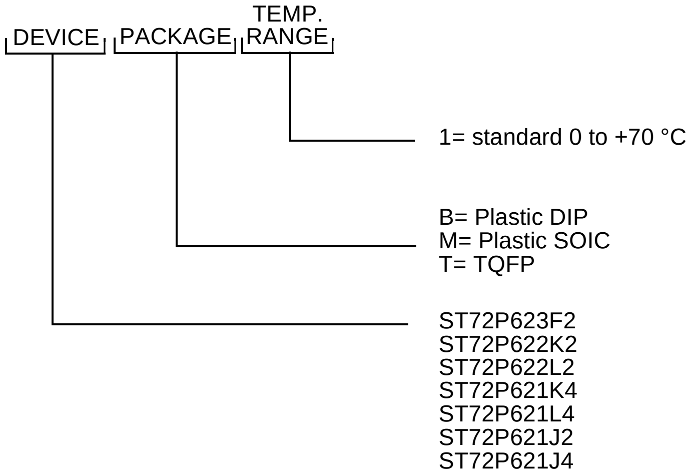
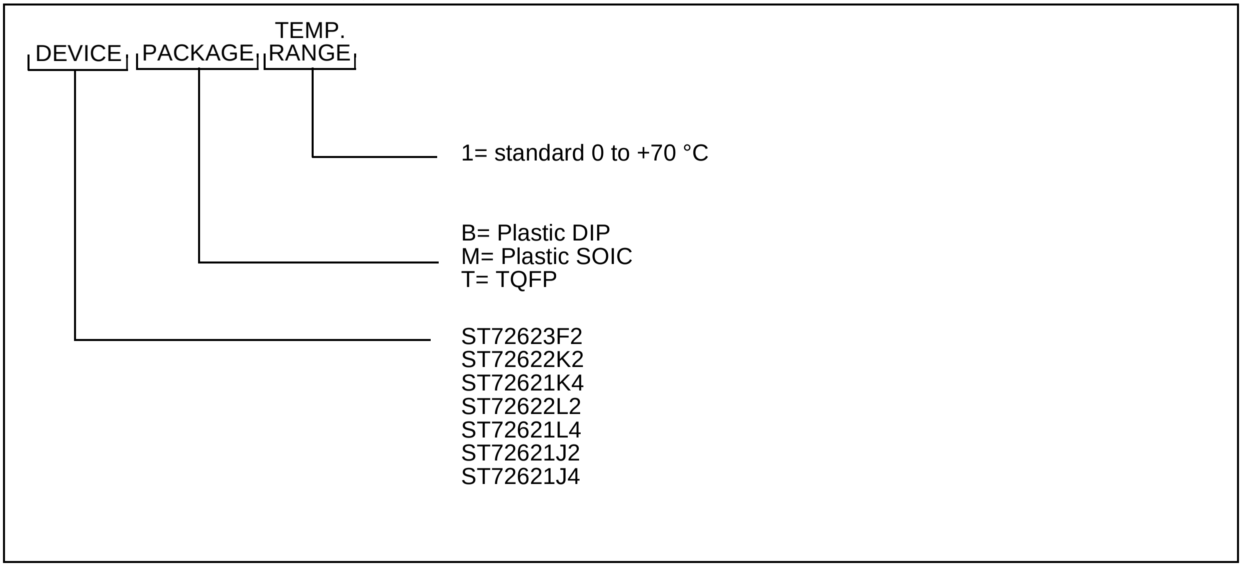

DEVICE
PACKAGE
TEMP. RANGE
1= standard 0 to +70 °C
B= Plastic DIP
M= Plastic SOIC
T= TQFP
ST72P623F2
ST72P622K2
ST72P622L2
ST72P621K4
ST72P621L4
ST72P621J2
ST72P621J4

Figure 90. ROM Factory-Programmed Device Types


DEVICE
PACKAGE
TEMP. RANGE
1= standard 0 to +70 °C
B= Plastic DIP
M= Plastic SOIC
T= TQFP
ST72623F2
ST72622K2
ST72621K4
ST72622L2
ST72621L4
ST72621J2
ST72621J4

STMicroelectronics OPTION LIST
ST7262 MICROCONTROLLER FAMILY (ROM DEVICE)

Customer: . . . .
Address: . . . .
. . . .
Contact: . . . .
Phone No: . . . .
Reference/ROM Code* : . . . .
*The ROM code name is assigned by STMicroelectronics.
ROM code must be sent in .S19 format. .Hex extension cannot be processed.

Watchdog: [ ] Software activation
[ ] Hardware activation

Low Voltage Detector: [ ] Disabled
[ ] Enabled

Oscillator Selection: [ ] 6 MHz.
[ ] 12 MHz.

Readout protection: [ ] Enabled
[ ] Disabled

Special Marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _"
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count: S020 8

DIP20	10
DIP32	10
SO34	13
DIP42	16
TQFP44	10

We have checked the ROM code verification file returned to us by STMicroelectronics. It conforms exactly with the ROM code file orginally supplied. We therefore authorize STMicroelectronics to proceed with device manufacture.

Signature . . . .

Date . . . .

STMicroelectronics OPTION LIST

ST72P62 MICROCONTROLLER FAMILY (FASTROM DEVICE)

Customer: . . . .
Address: . . . .
. . . .
Contact: . . . .
Phone No: . . . .
Reference/FASTROM Code* :. . . . .
*The FASTROM code name is assigned by STMicroelectronics.
FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Watchdog: [ ] Software activation
[ ] Hardware activation

Low Voltage Detector: [ ] Disabled
[ ] Enabled

Oscillator Selection: [ ] 6 MHz.
[ ] 12 MHz.

Readout protection: [ ] Enabled
[ ] Disabled

Special Marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _"
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count:

S020	8
DIP20	10
DIP32	10
SO34	13
DIP42	16
TQFP44	10

We have checked the FASTROM code verification file returned to us by STMicroelectronics. It conforms exactly with the FASTROM code file orginally supplied. We therefore authorize STMicroelectronics to proceed with device manufacture.

Signature . . . .

Date . . . .

14.3 ST7 APPLICATION NOTES

IDENTIFICATION	DESCRIPTION
PROGRAMMING AND TOOLS	
AN985	EXECUTING CODE IN ST7 RAM
AN986	USING THE ST7 INDIRECT ADDRESSING MODE
AN987	ST7 IN-CIRCUIT PROGRAMMING
AN988	STARTING WITH ST7 ASSEMBLY TOOL CHAIN
AN989	STARTING WITH ST7 HIWARE C
AN1039	ST7 MATH UTILITY ROUTINES
AN1064	WRITING OPTIMIZED HIWARE C LANGUAGE FOR ST7
AN1106	TRANSLATING ASSEMBLY CODE FROM HC05 TO ST7
EXAMPLE DRIVERS	
AN969	ST7 SCI COMMUNICATION BETWEEN THE ST7 AND A PC
AN970	ST7 SPI COMMUNICATION BETWEEN THE ST7 AND E²PROM
AN971	ST7 I²C COMMUNICATION BETWEEN THE ST7 AND E²PROM
AN972	ST7 SOFTWARE SPI MASTER COMMUNICATION
AN973	SCI SOFTWARE COMMUNICATION WITH A PC USING ST72251 16-BIT TIMER
AN974	REAL TIME CLOCK WITH THE ST7 TIMER OUTPUT COMPARE
AN976	DRIVING A BUZZER USING THE ST7 PWM FUNCTION
AN979	DRIVING AN ANALOG KEYBOARD WITH THE ST7 ADC
AN980	ST7 KEYPAD DECODING TECHNIQUES, IMPLEMENTING WAKE-UP ON KEYSTROKE
AN1017	USING THE ST7 USB MICROCONTROLLER
AN1041	USING ST7 PWM SIGNAL TO GENERATE ANALOG OUTPUT (SINUSOID)
AN1042	ST7 ROUTINE FOR I²C SLAVE MODE MANAGEMENT
AN1044	MULTIPLE INTERRUPT SOURCES MANAGEMENT FOR ST7 MCUS
AN1045	ST7 SOFTWARE IMPLEMENTATION OF I²C BUS MASTER
AN1046	ST7 UART EMULATION SOFTWARE
AN1047	MANAGING RECEPTION ERRORS WITH THE ST7 SCI PERIPHERAL
AN1048	ST7 SOFTWARE LCD DRIVER
AN1078	ST7 TIMER PWM DUTY CYCLE SWITCH FOR TRUE 0% or 100% DUTY CYCLE
AN1082	DESCRIPTION OF THE ST72141 MOTOR CONTROL
AN1083	ST72141 BLDC MOTOR CONTROL SOFTWARE AND FLOWCHART EXAMPLE
AN1129	PWM MANAGEMENT FOR BLDC MOTOR DRIVES USING THE ST72141
AN1130	BRUSHLESS DC MOTOR DRIVE WITH ST72141
AN1148	USING THE ST7263 FOR DESIGNING A USB MOUSE
AN1149	HANDLING SUSPEND MODE ON A USB MOUSE
AN1180	USING THE ST7263 KIT TO IMPLEMENT A USB GAME PAD
AN1182	USING THE ST7 USB LOW-SPEED FIRMWARE
PRODUCT OPTIMIZATION	
AN982	USING CERAMIC RESONATORS WITH THE ST7
AN1014	HOW TO MINIMIZE THE ST7 POWER CONSUMPTION
AN1070	ST7 CHECKSUM SELFCHECKING CAPABILITY
AN1179	PROGRAMMING ST7 FLASH MICROCONTROLLERS IN REMOTE ISP
PRODUCT EVALUATION	
AN910	ST7 AND ST9 PERFORMANCE BENCHMARKING
AN990	ST7 BENEFITS VERSUS INDUSTRY STANDARD
AN1086	ST7 / ST10U435 CAN-do SOLUTIONS FOR CAR MULTIPLEXING
AN1150	BENCHMARK ST72 VS PC16
AN1151	PERFORMANCE COMPARISON BETWEEN ST72254 & PC16F8

14.4 TO GET MORE INFORMATION

To get the latest information on this product please use the ST web server: http://mcu.st.com/

15 SUMMARY OF CHANGES

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

Revision	Main Changes	Date
1.0	SCI updated with new peripheral features in section 10.5 on page 66 SPI updated with new peripheral features in section 10.4 on page 53 ADC consistancy description modified 'ADCDRMSB and ADCDRLSB are unlocked when the MSB is read" in section 10.7.3.3 on page 89 Low Voltage Reset Thresholds section 12.3.2 on page 101 Modification of Absolute Max. Voltage Characteristics in section 12.2.1 on page 99. Added V_{IN} to I/O port characteristics in section 12.8.1 on page 111 A/D Converter TUE changed from 3 to 4 LSB in section 12.12 on page 121 Changed polarity of WDG option bit in section 14.1 on page 126 Added note in Table 1, "Device Pin Description," on page 10 V_{DDA} must be connected externally to V_{DD}	Dec 00

Notes:

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

The ST logo is a registered trademark of STMicroelectronics

©2001 STMicroelectronics - All Rights Reserved.

Purchase of I^{2}C Components by STMicroelectronics conveys a license under the Philips I^{2}C Patent. Rights to use these components in an I^{2}C system is granted provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies
Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain
Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com