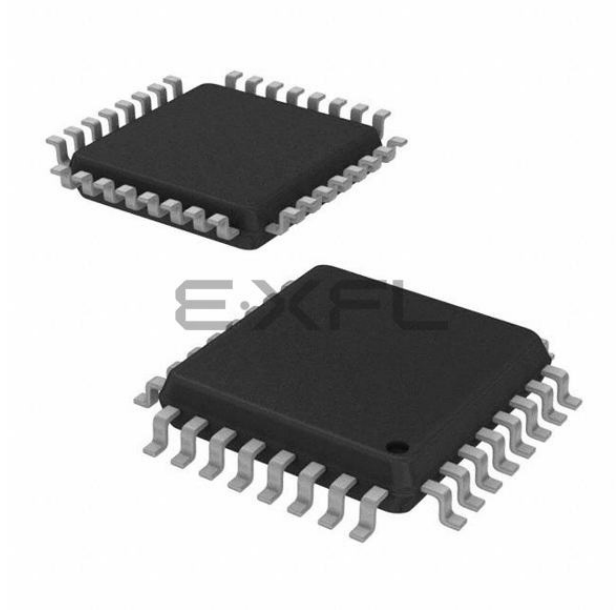


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7foxk1t6

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.

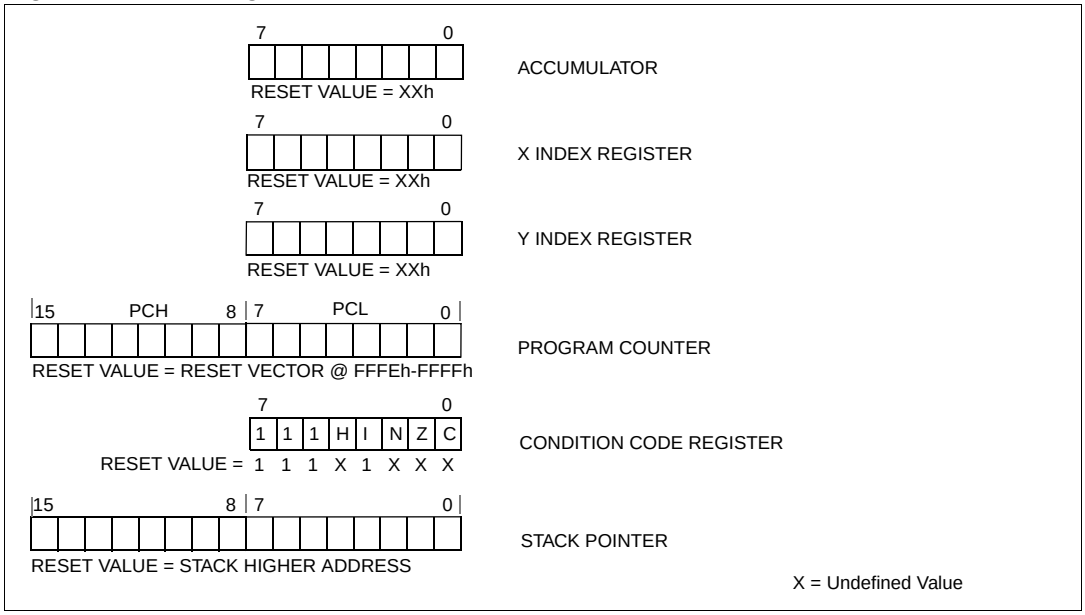
5.2 Main features

- 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 six CPU registers shown in *Figure 7*. They are not present in the memory mapping and are accessed by specific instructions.

Figure 7. CPU registers



Bit 3 = **I** *Interrupt mask bit*

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 interruptible 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 bit*

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

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

1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 **= Z** *Zero bit*

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 bit*

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

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

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

Interrupt management bits

Bits 5,3 = **I1, I0** *Interrupt bits*

The combination of the I1 and I0 bits gives the current interrupt software priority.

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions. See *Section 9.6: Interrupts* for more details.

Figure 8. Stack manipulation example

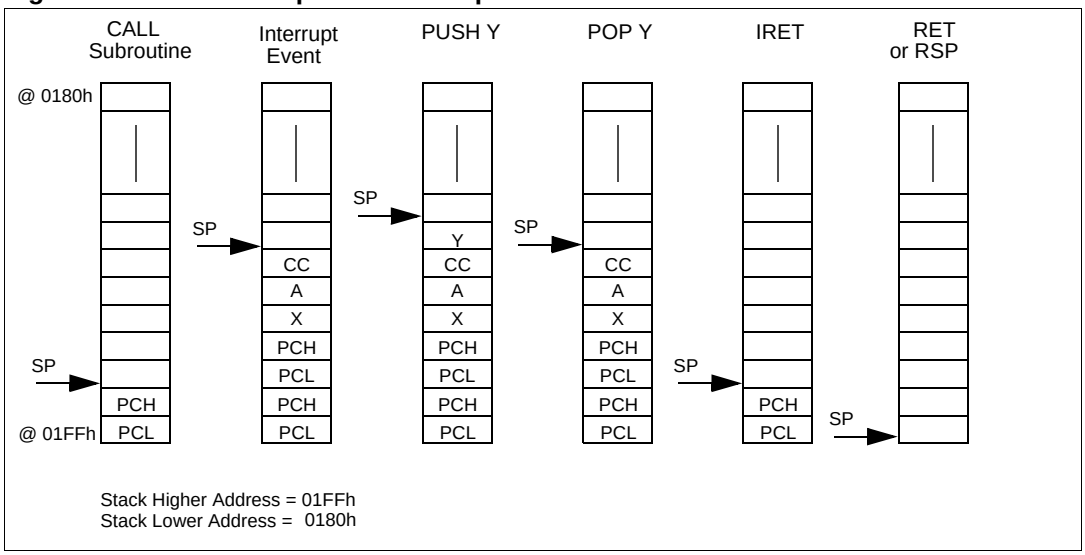
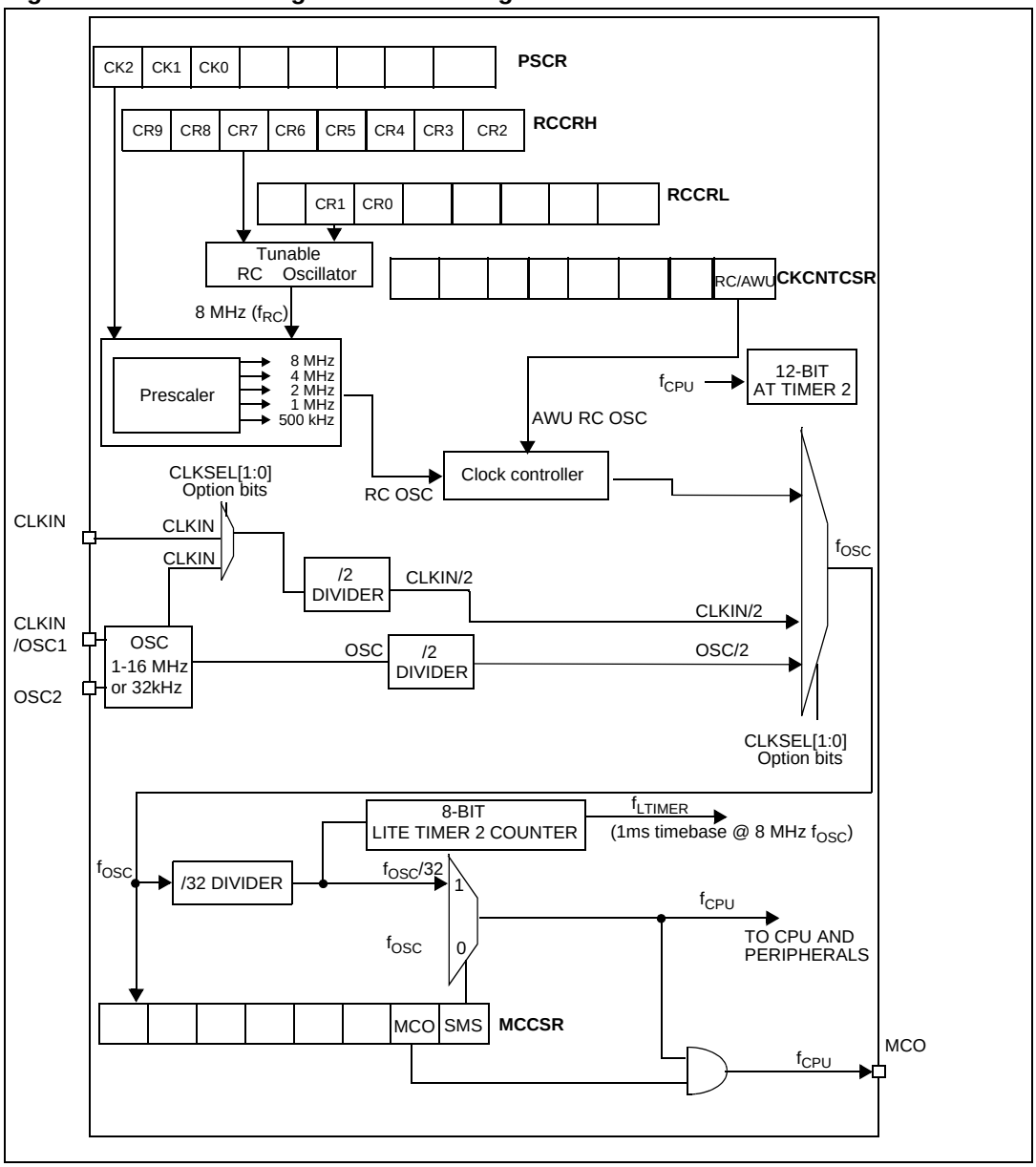


Figure 12. Clock management block diagram



6.2 Multi-oscillator (MO)

The main clock of the ST7 can be generated by four different source types coming from the multi-oscillator block (1 to 16 MHz):

- An external source
- 5 different configurations for crystal or ceramic resonator oscillators
- An internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in *Table 8*. Refer to the electrical characteristics section for more details.

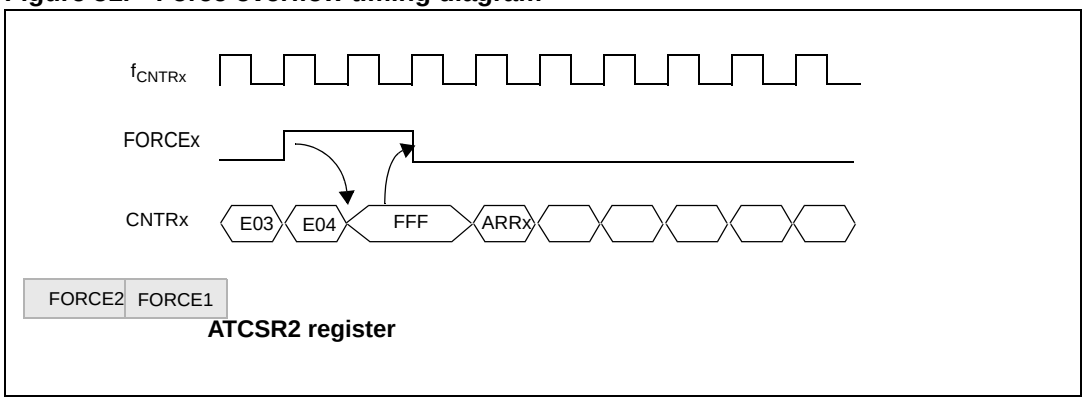
Force update

In order not to wait for the counter$_x$ overflow to load the value into active DCRx registers, a programmable counter$_x$ overflow is provided. For both counters, a separate bit is provided which when set, make the counters start with the overflow value, i.e. FFFh. After overflow, the counters start counting from their respective auto reload register values.

These bits are FORCE1 and FORCE2 in the ATCSR2 register. FORCE1 is used to force an overflow on Counter 1 and, FORCE2 is used for Counter 2. These bits are set by software and reset by hardware after the respective counter overflow event has occurred.

This feature can be used at any time. All related features such as PWM generation, Output Compare, Input Capture, One-pulse (refer to *Figure 50: Dynamic DCR2/3 update in one pulse mode*) etc. can be used this way.

Figure 51. Force overflow timing diagram



10.2.4 Low power modes

Table 35. Effect of low power modes on autoreload timer

Mode	Description
Wait	No effect on AT timer
Halt	AT timer halted.

10.2.5 Interrupts

Table 36. Description of interrupt events

Interrupt Event	Event Flag	Enable Control bit	Exit from Wait	Exit from Halt	Exit from Active-Halt
Overflow Event	OVF1	OVIE1	Yes	No	Yes
AT4 IC Event	ICF	ICIE	Yes	No	No
Overflow Event2	OVF2	OVIE2	Yes	No	No

Note: The AT4 IC is connected to an interrupt vector. The OVF event is mapped on a separate vector (see Interrupts chapter).
They generate an interrupt if the enable bit is set in the ATCSR register and the interrupt mask in the CC register is reset (RIM instruction).

Input Capture register Low (ATICRL)

Reset value: 0000 0000 (00h)

7							0
ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
Read only							

Bits 11:0 = **ICR[11:0]** *Input Capture Data*.

This is a 12-bit register which is readable by software and cleared by hardware after a reset. The ATICR register contains captured the value of the 12-bit CNTR1 register when a rising or falling edge occurs on the ATIC or LTIC pin (depending on ICS). Capture will only be performed when the ICF flag is cleared.

Break Enable register (BREAKEN)

Reset value: 0000 0011 (03h)

7							0
0	0	0	0	0	0	BREN2	BREN1
Read/write							

Bits 7:2 = Reserved, must be kept cleared.

Bit 1 = **BREN2** *Break Enable for Counter 2 bit*

This bit is read/write by software. It enables the break functionality for Counter2 if BA bit is set in BREAKCR. It controls PWM2/3 if ENCNTR2 bit is set.

0: No Break applied for CNTR2

1: Break applied for CNTR2

Bit 0 = **BREN1** *Break Enable for Counter 1 bit*

This bit is read/write by software. It enables the break functionality for Counter1. If BA bit is set, it controls PWM0/1 by default, and controls PWM2/3 also if ENCNTR2 bit is reset.

0: No Break applied for CNTR1

1: Break applied for CNTR1

Dead Time Generator register (DTGR)

Reset value: 0000 0000 (00h)

7							0
DTE	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Read/write							

Bit 7 = **DTE** *Dead Time Enable bit*

This bit is read/write by software. It enables a dead time generation on PWM0/PWM1.

0: No Dead time insertion.

1: Dead time insertion enabled.

Bits 6:0 = **DT[6:0]** *Dead Time value*

These bits are read/write by software. They define the dead time inserted between PWM0/PWM1. Dead time is calculated as follows:

Dead Time = DT[6:0] x Tcounter1

Note: If DTE is set and DT[6:0]=0, PWM output signals will be at their reset state.

Table 38. Register mapping and reset values

Add. (Hex)	Register label	7	6	5	4	3	2	1	0
0011	**ATCSR** Reset Value	0	ICF 0	ICIE 0	CK1 0	CK0 0	OVF1 0	OVFIE1 0	CMPIE 0
0012	**CNTR1H** Reset Value	0	0	0	0	CNTR1_11 0	CNTR1_10 0	CNTR1_9 0	CNTR1_8 0
0013	**CNTR1L** Reset Value	CNTR1_7 0	CNTR1_8 0	CNTR1_7 0	CNTR1_6 0	CNTR1_3 0	CNTR1_2 0	CNTR1_1 0	CNTR1_0 0
0014	**ATR1H** Reset Value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
0015	**ATR1L** Reset Value	ATR7 0	ATR6 0	ATR5 0	ATR4 0	ATR3 0	ATR2 0	ATR1 0	ATR0 0
0016	**PWMCR** Reset Value	0	OE3 0	0	OE2 0	0	OE1 0	0	OE0 0
0017	**PWM0CSR** Reset Value	0	0	0	0	0	0	OP0 0	CMPF0 0
0018	**PWM1CSR** Reset Value	0	0	0	0	0	0	OP1 0	CMPF1 0
0019	**PWM2CSR** Reset Value	0	0	0	0	0	0	OP2 0	CMPF2 0
001A	**PWM3CSR** Reset Value	0	0	0	0	OP_EN 0	OPEDGE 0	OP3 0	CMPF3 0
001B	**DCR0H** Reset Value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0

10.3 Lite timer 2 (LT2)

10.3.1 Introduction

The Lite timer can be used for general-purpose timing functions. It is based on two free-running 8-bit upcounters, a watchdog function and an 8-bit Input Capture register.

10.3.2 Main features

- Real-time Clock
  - One 8-bit upcounter 1 ms or 2 ms timebase period (@ 8 MHz f_{OSC})
  - One 8-bit upcounter with autoreload and programmable timebase period from 4µs to 1.024 ms in 4 µs increments (@ 8 MHz f_{OSC})
  - 2 Maskable timebase interrupts
- Input Capture
  - 8-bit Input Capture register (LTICR)
- Maskable interrupt with wakeup from Halt mode capability
- Watchdog
  - Enabled by hardware or software (configurable by option byte)
  - Optional reset on HALT instruction (configurable by option byte)
  - Automatically resets the device unless disable bit is refreshed
  - Software reset (Forced Watchdog reset)
  - Watchdog reset status flag

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

Figure 57. Counter timing diagram, internal clock divided by 2

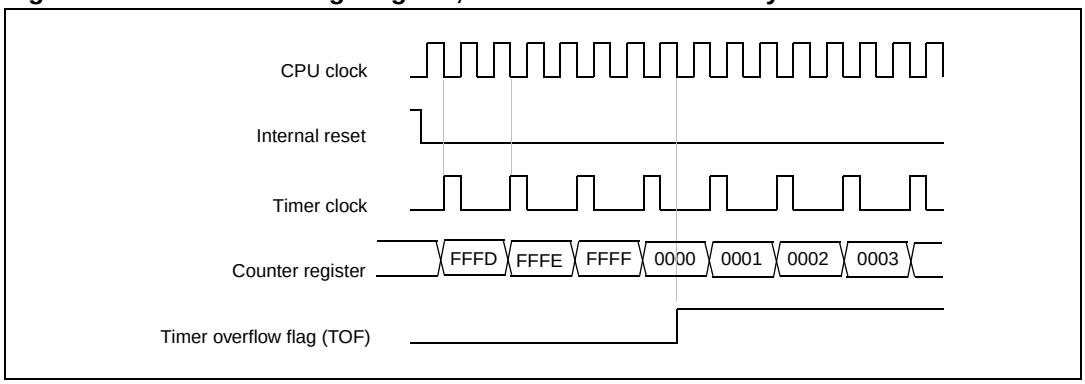


Figure 58. Counter timing diagram, internal clock divided by 4

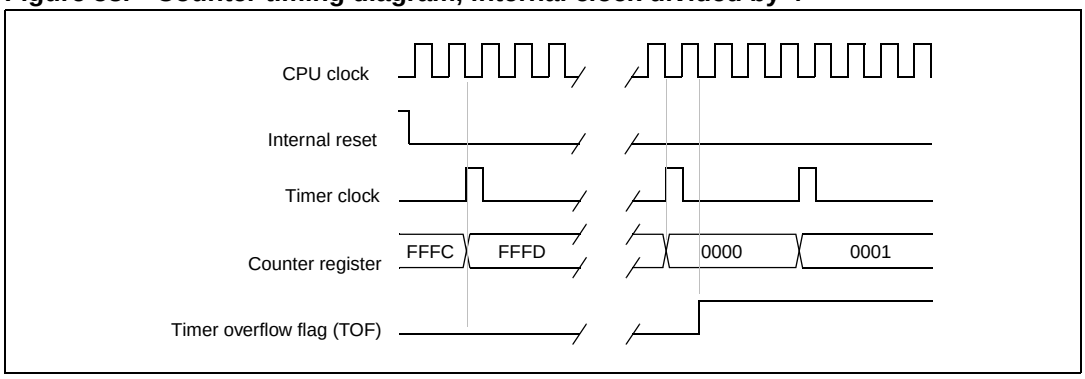
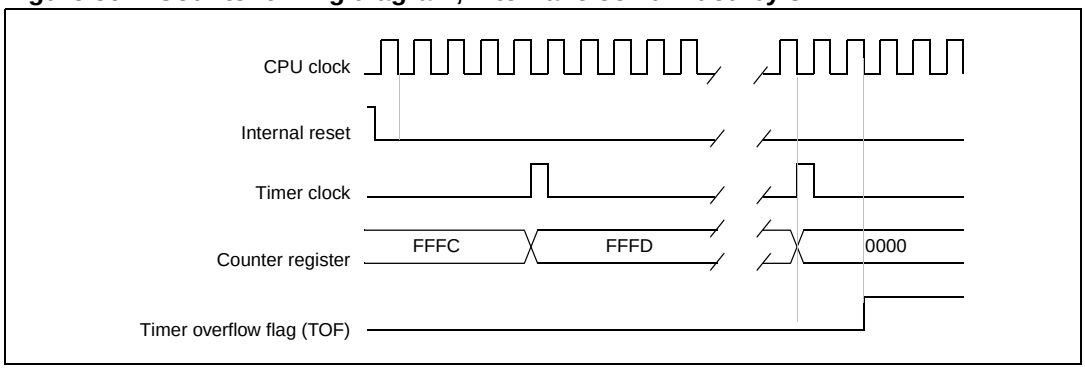


Figure 59. Counter timing diagram, internal clock divided by 8



Note: The device is in reset state when the internal reset signal is high, when it is low the device is running.

10.5 I²C bus interface (I²C)

10.5.1 Introduction

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

10.5.2 Main features

- Parallel-bus/I²C protocol converter
- Multi-master capability
- 7-bit/10-bit Addressing
- Transmitter/Receiver flag
- End-of-byte transmission flag
- Transfer problem detection

I²C master features:

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

I²C slave features:

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

10.5.3 General description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

When the I²C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

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

Figure 70. I²C interface block diagram

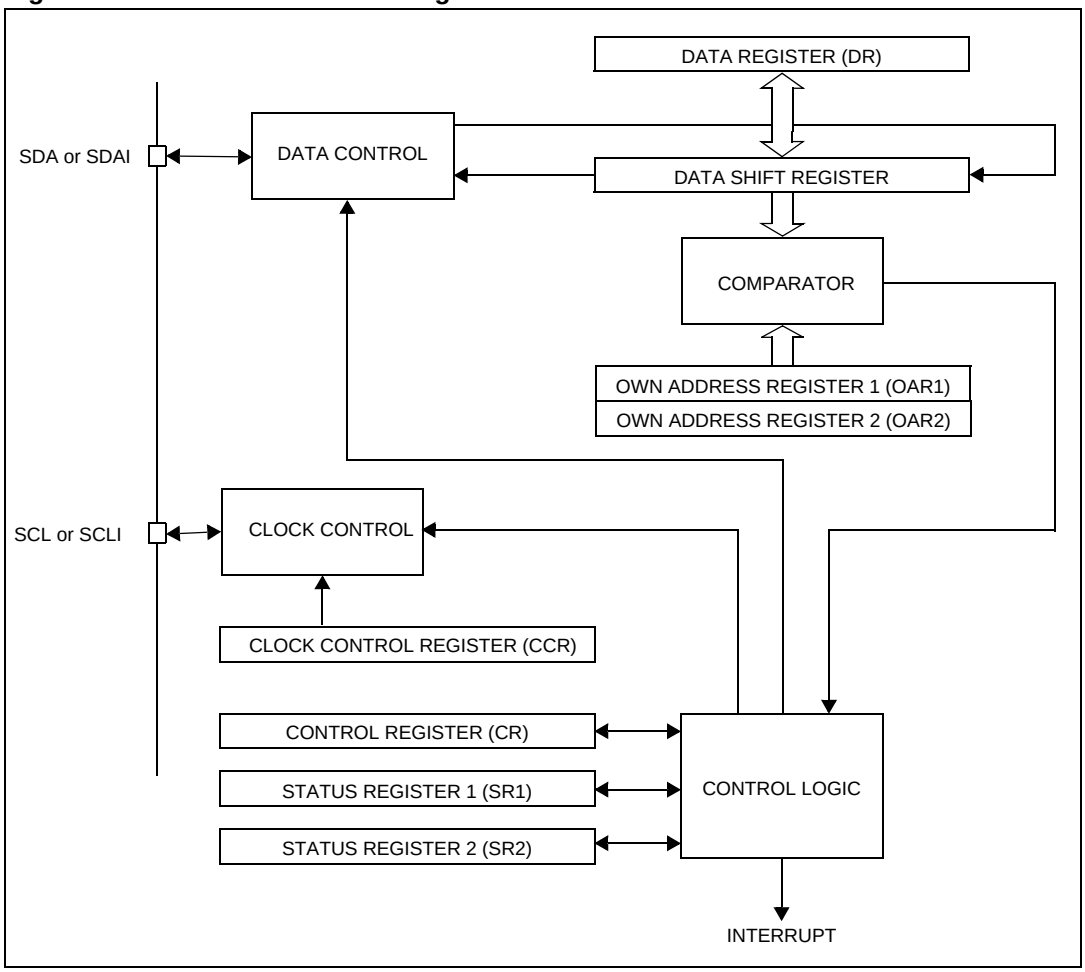
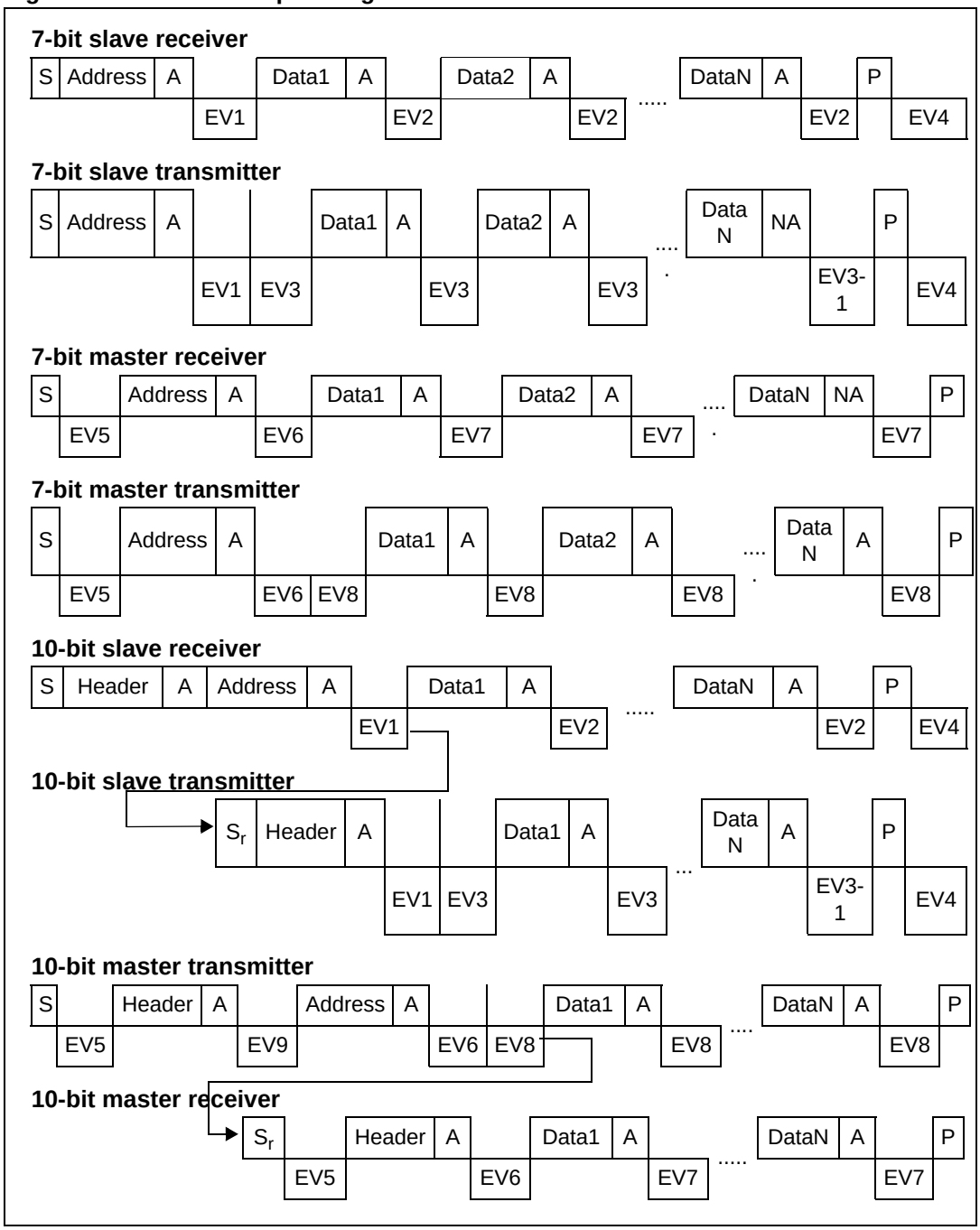
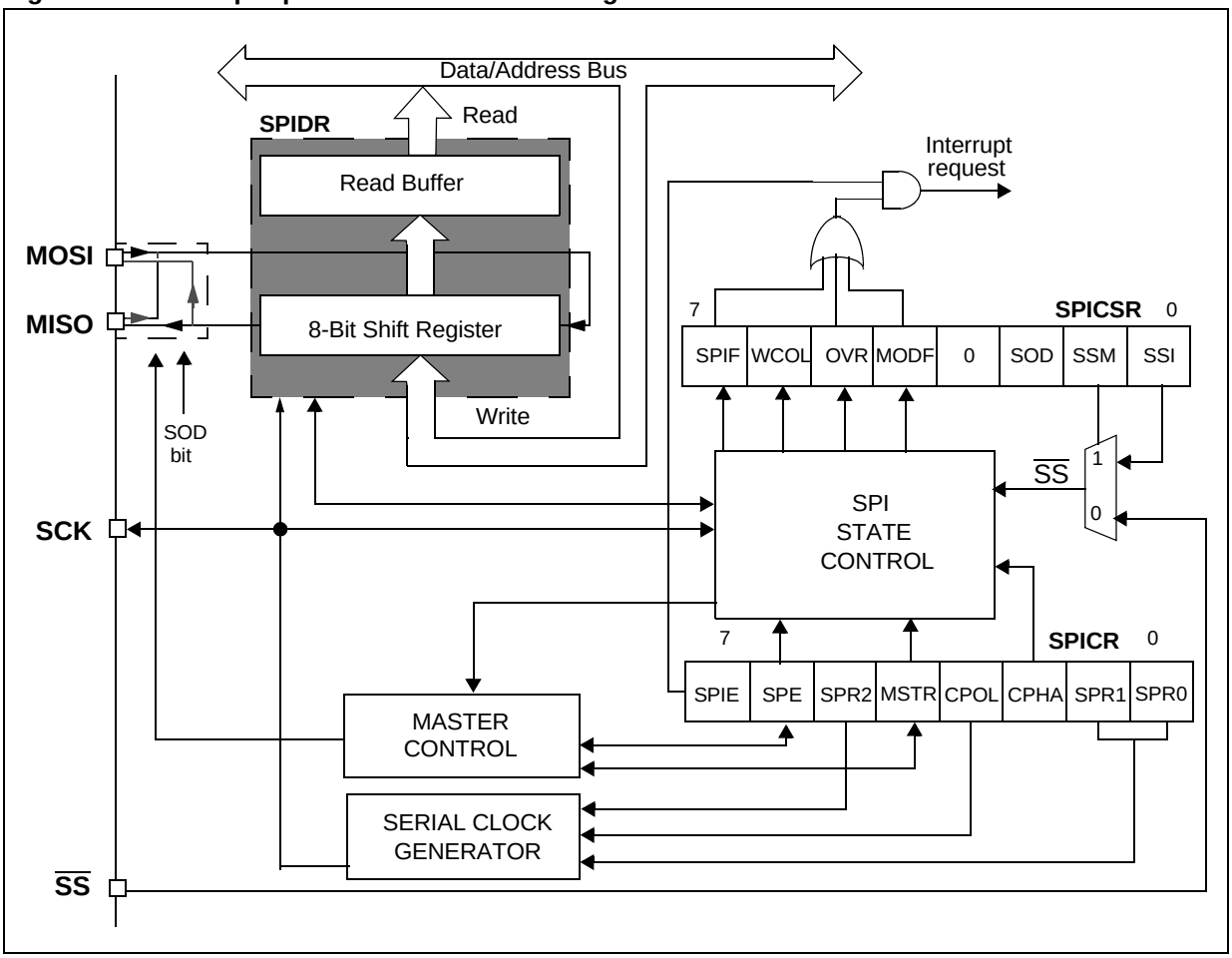


Figure 71. Transfer sequencing



1. S=Start, S_r = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1).
2. **EV1:** EVF=1, ADSL=1, cleared by reading SR1 register.
3. **EV2:** EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.
4. **EV3:** EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.
5. **EV3-1:** EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). If lines are released by STOP=1, STOP=0, the

Figure 73. Serial peripheral interface block diagram



10.6.4 Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 74*.

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

The communication is always initiated by the master. 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 communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see *Figure 77 on page 164*) but master and slave must be programmed with the same timing mode.

Data register High (ADCDRH)

Reset value: xxxx xxxx (xxh)

7							0
D9	D8	D7	D6	D5	D4	D3	D2
Read only							

Bits 7:0 = **D[9:2]** *MSB of Analog Converted Value*

ADC Control/data register Low (ADCDRL)

Reset value: 0000 00xx (0xh)

7							0
0	0	0	0	SLOW	0	D1	D0
Read/write							

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

Bit 3 = **SLOW** *Slow mode bit*

This bit is set and cleared by software. It is used together with the SPEED bit in the ADCCSR register to configure the ADC clock speed as shown on the table below.

Table 56. Configuring the ADC clock speed

f_{ADC}(1)	SLOW	SPEED
$f_{CPU}/2$	0	0
f_{CPU}	0	1
$f_{CPU}/4$	1	x

1. The maximum allowed value of f_{ADC} is 4 MHz (see *Section 12.11 on page 209*)

Bit 2 = Reserved. Forced by hardware to 0.

Bits 1:0 = **D[1:0]** *LSB of Analog Converted value*

Table 57. ADC register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0036h	**ADCCSR** Reset Value	EOC 0	SPEED 0	ADON 0	0 0	CH3 0	CH2 0	CH1 0	CH0 0
0037h	**ADCDRH** Reset Value	D9 x	D8 x	D7 x	D6 x	D5 x	D4 x	D3 x	D2 x
0038h	**ADCDRL** Reset Value	0 0	0 0	0 0	0	SLOW 0	0	D1 x	D0 x

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:

Table 64. ST7 instruction set

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 rotate	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				
Condition Code Flag modification	SIM	RIM	SCF	RCF				

Using a prebyte

The instructions are described with 1 to 4 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 by:

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

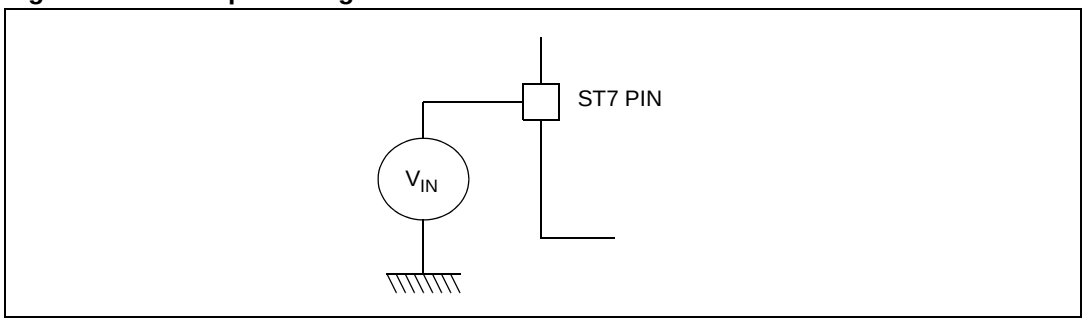
These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

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.

Figure 82. 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.

Table 66. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	7.0	V
V_{IN}	Input voltage on any pin(1)(2)	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human Body model)	see *Section 12.8.3 on page 203*	
$V_{ESD(CDM)}$	Electrostatic discharge voltage (Charge Device model)		

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: 4.7 kΩ for $\overline{\text{RESET}}$, 10 kΩ 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. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected

12.3.3 Internal RC oscillator

To improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100 nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device

Internal RC oscillator calibrated at 5.0 V

The ST7 internal clock can be supplied by an internal RC oscillator (selectable by option byte).

Table 71. Internal RC oscillator characteristics (5.0 V calibration)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC}	Internal RC oscillator frequency	RCCR = FF (reset value), T_A = 25 °C, V_{DD} = 5 V		5.5		MHz
		RCCR=RCCR0(1), T_A = 25 °C, V_{DD} = 5 V		8		
$f_{G(RC)}$	RC trimming granularity	T_A = 25 °C, V_{DD} = 5 V		6		kHz
ACC_{RC}	Accuracy of Internal RC oscillator with RCCR=RCCR0(1)	T_A = 25 °C, V_{DD} = 5 V (2) without user calibration		±7		%
		T_A = 25 °C, V_{DD} = 4.5 to 5.5 V (2) with user calibration	-2		2	%
		T_A= 0 to +85 °C, V_{DD} = 4.5 to 5.5 V(2) with user calibration	-2.5		4	%
		T_A = -40 to 0 °C, V_{DD} = 4.5 to 5.5 V(2) with user calibration	-4		2.5	%
$t_{su(RC)}$	RC oscillator setup time	T_A = 25 °C, V_{DD} = 5 V		4 (3)		µs

1. See *Section 6.1.1: Internal RC oscillator*
2. Guaranteed by characterization
3. Not tested in production

12.10 Control pin characteristics

12.10.1 Asynchronous $\overline{\text{RESET}}$ pin

T_A = -40 to 85 °C, unless otherwise specified.

Table 89. Asynchronous $\overline{\text{RESET}}$ pin characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage			V_{SS} - 0.3		$0.3V_{DD}$	V
V_{IH}	Input high level voltage			$0.7V_{DD}$		V_{DD}+0.3	
V_{hys}	Schmitt trigger voltage hysteresis(1)				2		V
V_{OL}	Output low level voltage (2)	V_{DD}= 5 V	I_{IO} = +2 mA		200		mV
R_{ON}	Pull-up equivalent resistor(3)	V_{IN}=V_{SS}	V_{DD} = 5 V	30	50	70	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset sources			90(1)		µs
$t_{h(RSTL)in}$	External reset pulse hold time(4)			20			µs
$t_{g(RSTL)in}$	Filtered glitch duration				200		ns

1. Data based on characterization results, not tested in production
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section Table 67. on page 189* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. Specified for voltages on $\overline{\text{RESET}}$ pin between V_{ILmax} and V_{DD}
4. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

Figure 95. 20-pin plastic dual in-line package, 300-mil width, package outline

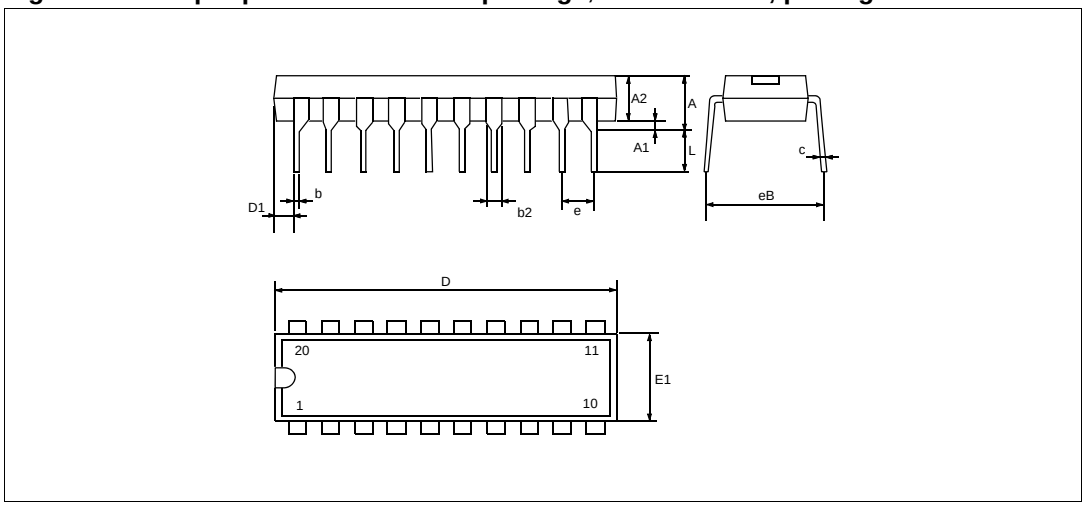


Table 98. 20-pin plastic dual in-line package, 300-mil width, mechanical data

Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			5.33			0.2098
A1	0.38			0.0150		
A2	2.92	3.30	4.95	0.1150	0.1299	0.1949
b	0.36	0.46	0.56	0.0142	0.0181	0.0220
b2	1.14	1.52	1.78	0.0449	0.0598	0.0701
c	0.20	0.25	0.36	0.0079	0.0098	0.0142
D	24.89	26.16	26.92	0.9799	1.0299	1.0598
D1	0.13			0.0051		
e		2.54			0.1000	
eB			10.92			0.4299
E1	6.10	6.35	7.11	0.2402	0.2500	0.2799
L	2.92	3.30	3.81	0.1150	0.1299	0.1500
	Number of Pins					
N	20					

1. Values in inches are converted from mm and rounded to 4 decimal digits.