ST52T430/E430

8-BIT INTELLIGENT CONTROLLER UNIT (ICU)
Three Timer/PWMs, ADC, SCI

PRELIMINARY DATASHEET

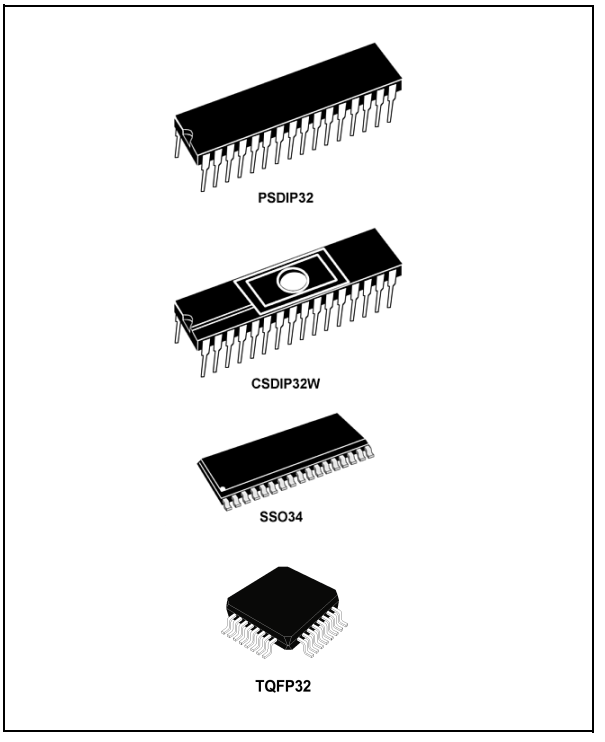
Memories
- Up to 8 Kbytes EPROM/OTP
- 256 bytes of RAM
- Readout Protection

Core
- Register File Based Architecture
- 55 instructions
- Hardware multiplication and division
- Decision Processor for the implementation of Fuzzy Logic algorithms

Clock and Power Supply
- Up to 20 MHz clock frequency.
- Power Saving features

Interrupts
- 6 interrupt vectors
- Top Level External Interrupt (INT)

Peripherals
- 3 Programmable 8-bit Timer/PWMs with internal 16-bit Prescaler featuring:
  - PWM output
  - Input capture
  - Output Compare
  - Pulse Generator mode
- Watchdog timer
- On-chip 8-bit Sample and Hold A/D Converter with 8-channel analog multiplexer
- Serial Communication Interface with asynchronous protocol (UART)



I/O Ports
- 23 I/O PINs configurable in Input and Output mode
- High current sink/source in all pins.

Development tools
- High level Software tools
- Emulator
- Low cost Programmer
- Gang Programmer

ST52x430 Devices Summary

Device	NVM	RAM	Timer PWM	ADC	SCI	Watchdog	Operating Supply	I/O	Package
ST52T430K1	2K OTP	256	3x8-bit	8-Ch	Yes	Yes	3.0-5.5 V	23	Sdip32 Sso34 Tqfp32
ST52T430K2	4K OTP	256	3x8-bit	8-Ch	Yes	Yes	3.0-5.5 V	23	Sdip32 Sso34 Tqfp32
ST52T430K3	8K OTP	256	3x8-bit	8-Ch	Yes	Yes	3.0-5.5 V	23	Sdip32 Sso34 Tqfp32
ST52E430K3	8K EPROM	256	3x8-bit	8-Ch	Yes	Yes	3.0-5.5 V	23	Csdip32w

TABLE OF CONTENTS

1 GENERAL DESCRIPTION

1.1 Introduction

ST52x430 is an 8-bit Intelligent Control Units (ICU) of the ST Five Family, which can perform both boolean and fuzzy algorithms in an efficient manner, in order to reach the best performances that the two methodologies allow.

ST52x430 is produced by STMicroelectronics using the reliable high performance CMOS process, including integrated-on-chip peripherals that allow maximization of system reliability, decreasing system costs and minimizing the number of external components.

The flexible I/O configuration of ST52x400/440 allows for an interface with a wide range of external devices, like D/A converters or power control devices.

ST52x430 pins are configurable, allowing the user to set the input or output signals on each single pin.

A hardware multiplier (8 bit by 8 bit with 16 bit result) and a divider (16 bit over 8 bit with 8 bit result and 8 bit remainder) are available to implement complex functions by using a single instruction. The program memory utilization and computational speed is optimized.

Fuzzy Logic dedicated structures in ST52x430 ICU's can be exploited to model complex systems with high accuracy in a useful and easy way.

Fuzzy Expert Systems for overall system management and fuzzy Real time Controls can be designed to increase performances at highly competitive costs.

The linguistic approach characterizing Fuzzy Logic is based on a set of IF-THEN rules, which describe the control behavior, as well as on Membership Functions, which are associated to input and output variables.

Up to 334 Membership Functions, with triangular and trapezoidal shapes, or singleton values are available to describe fuzzy variables.

The Timer/PWM peripheral allows the management of power devices and timing signals, implementing different operating modes and high frequency PWM (Pulse With Modulation) controls. Input Capture and Output Compare functions are available on the TIMER.

The programmable Timer has a 16 bit Internal Prescaler and an 8 bit Counter. It can use internal or external Start/Stop signals and clock.

An internal programmable Watchdog is available to avoid loop errors and to reset the ICU.

ST52x430 includes an 8-bit Analog to Digital Converter with an 8-analog channel Multiplexer. Single/Multiple channels and Single/Sequence conversion modes are supported.

A Serial Communication peripheral (SCI), which uses the UART protocol allows data transfer from the ST52x430 to other external devices.

In order to optimize energy consumption, two different power saving modes are available: Wait mode and Halt mode.

Program Memory (EPROM/OTP) addressing capability addresses up to 8 Kbytes of memory locations to store both program instructions and permanent data.

EPROM can be locked by the user to prevent external undesired operations.

Operations may be performed on data stored in RAM, allowing the direct combination of new input and feedback data. All bytes of RAM are used like Register File.

OTP (One Time Programmable) version devices are fully compatible with the EPROM windowed version, which may be used for prototyping and pre-production phases of development.

A powerful development environment consisting of a board and software tools allows an easy configuration and use of ST52x430.

The VISUAL FIVE^{TM} software tool allows development of projects through a user-friendly graphical interface and optimization of generated code.

1.2 Functional Description

ST52x430 ICU can work in two modes:

- Memory Programming Mode
- Working Mode

according to RESET and Vpp signals levels (see pins description).

Note: When RESET=0 it is advisable not to use the sequence "101010" to port PA (7 : 2).

1.2.1 Memory Programming Mode.

The ST52x430 memory is loaded in the Memory Programming Phase. All fuzzy and standard instructions are written inside the memory.

This phase starts by setting the control signals as illustrated below:

RESET	TEST	V_{PP}
V_{SS}	V_{SS}	12V/V_{DD}

When this phase starts, the ST52x430 core is set to RESET status; then 12V are applied to the Vpp

pin in order to start EPROM programming. A signal applied to PB1 is used to increment the memory address; the data is supplied to PORT A (see EPROM programming for further details).

1.2.2 Working mode.

Below are the control signals of this mode:

RESET	TEST	V_{PP}
V_{DD}	V_{SS}	V_{SS}

The processor starts the working phase following the instructions, which have been previously loaded in the memory.

ST52x430's internal structure includes a computational block, CONTROL UNIT (CU) / DATA PROCESSING UNIT (DPU), which allows processing of boolean functions and fuzzy algorithms.

The CU/DPU can manage up to 334 different Membership Functions for the fuzzy rules antecedent part. The rule consequents are "crisp" values (real numbers). The maximum number of rules that can be defined is limited by the dimensions of the implemented standard algorithm.

EPROM is then shared between fuzzy and standard algorithms. The Membership Function data is stored inside the first 1024 memory locations. The Fuzzy rules are parts of the program instructions.

The Control Unit (CU) reads the information and the status deriving from the peripherals.

Arithmetic calculus can be performed on these values by using the internal CU and the 128/256 bytes of RAM, which supports all computations. The peripheral input can be fuzzy and/or arithmetic output, or the values contained in Data RAM and EPROM locations.

Figure 1.1 TQFP32 Pin Configuration

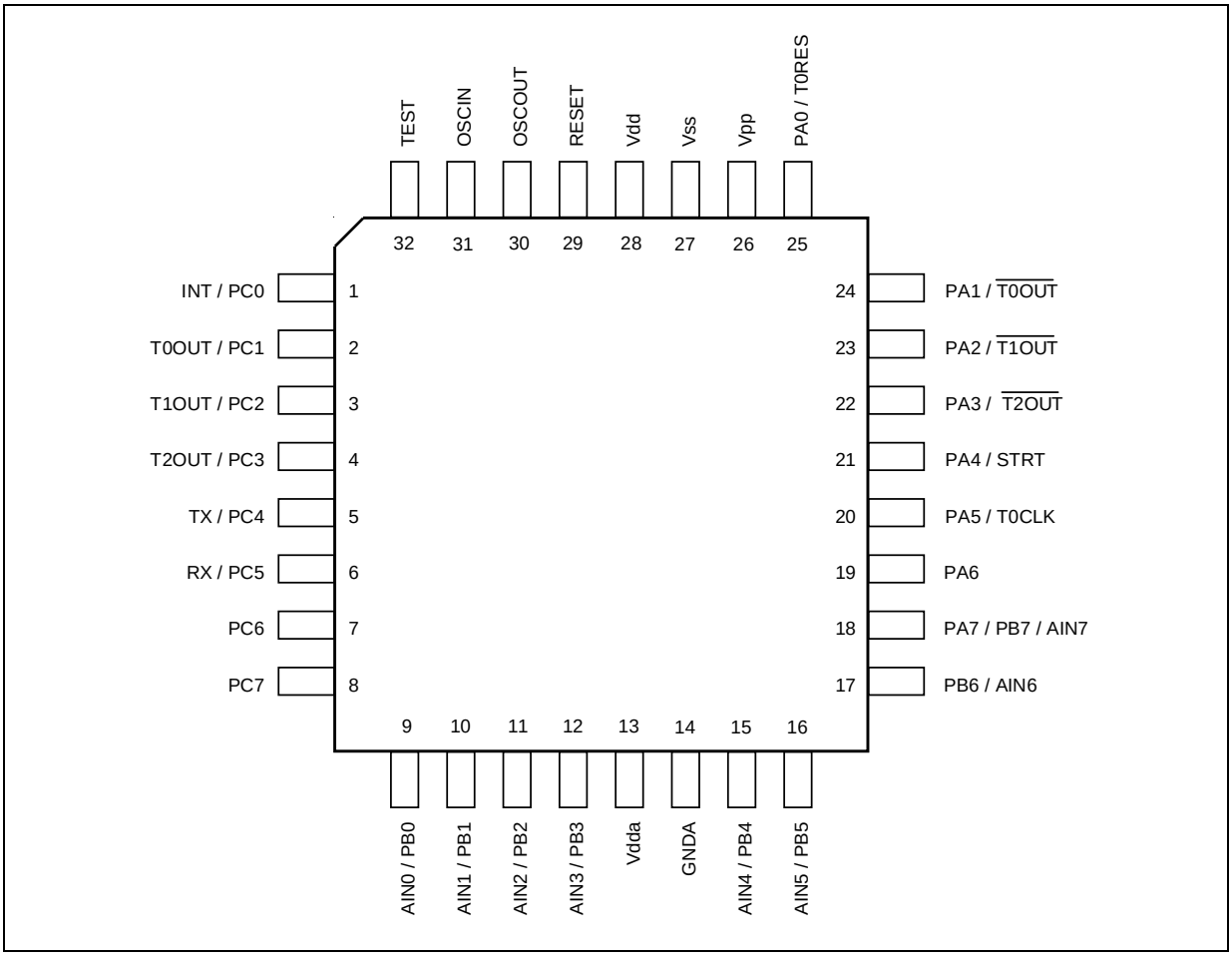


Figure 1.2 SSO34 Pin Configuration

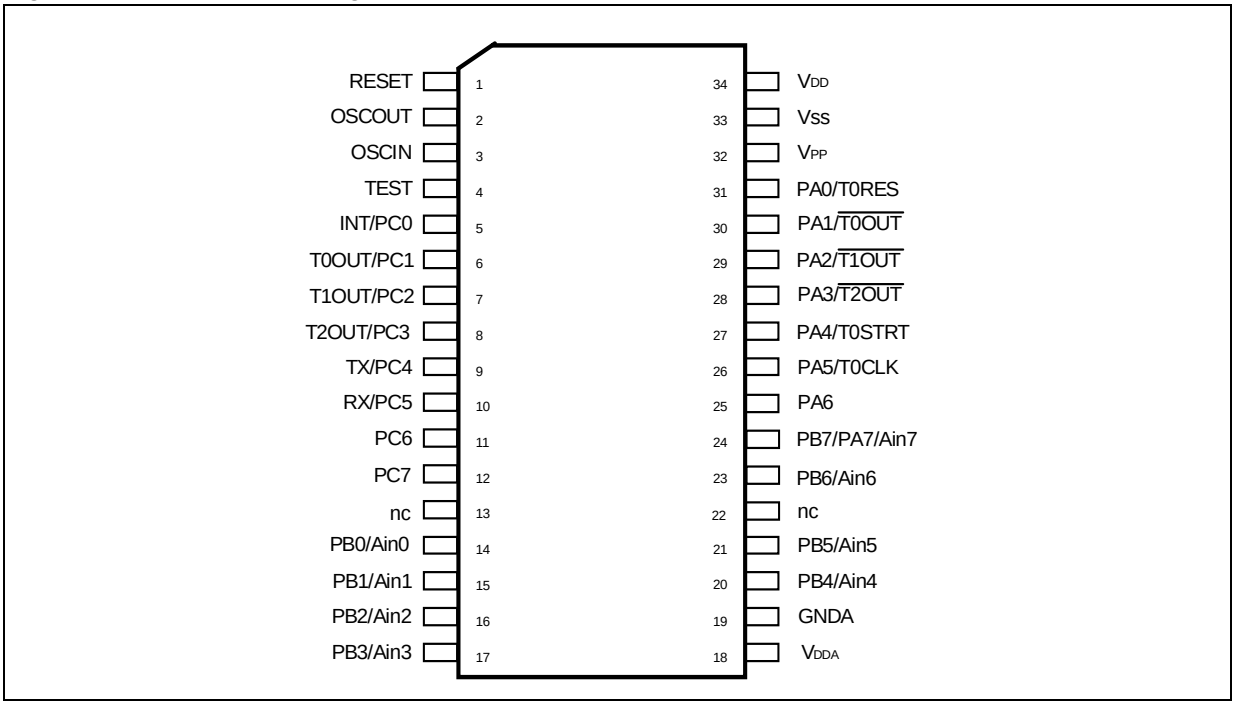


Figure 1.3 PSDIP32 Pin Configuration

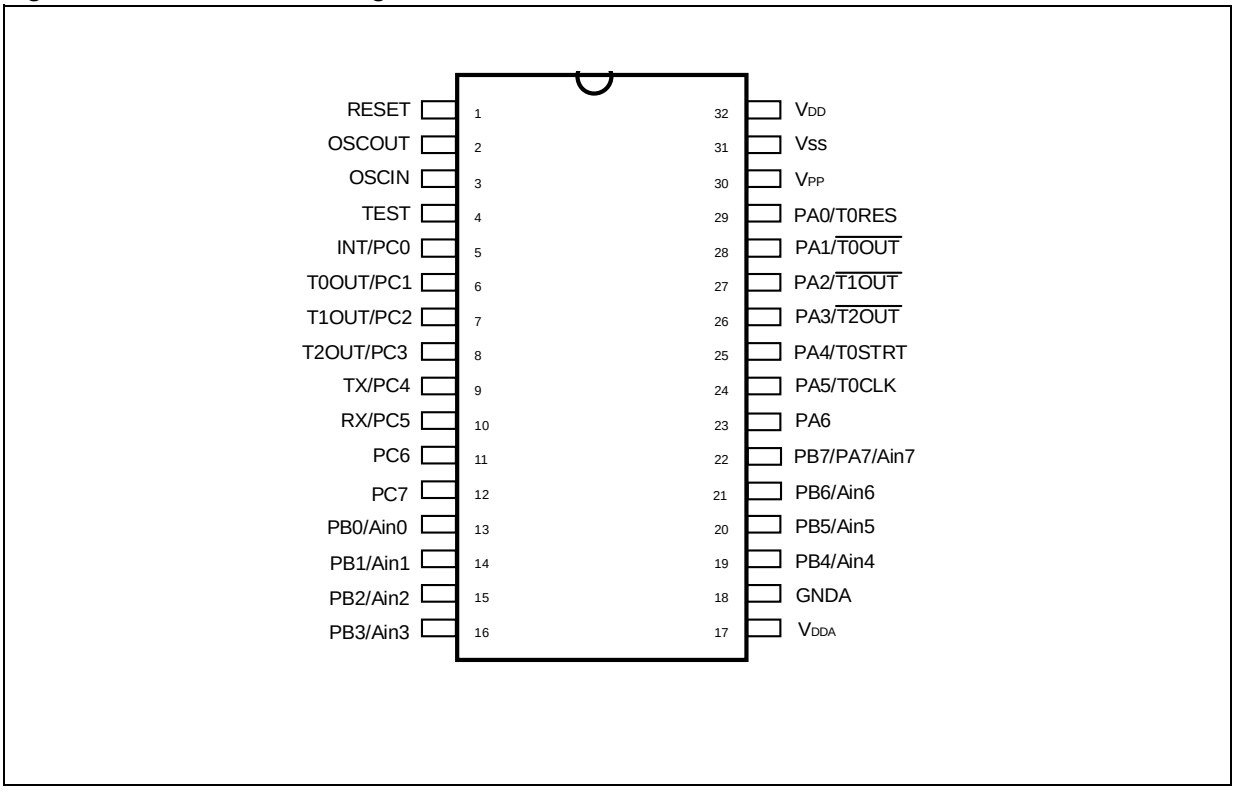


Table 1.1 ST52x430 SSO34 & PSDIP32 Pin list

SSO34 Pins	SDIP32 Pins	NAME	Programming Phase	Working Phase
1	1	RESET	General Reset	General Reset
2	2	OSCOUT		Oscillator Output
3	3	OSCIN		Oscillator Input
4	4	TEST	Must be tied to V_{SS}	Must be tied to V_{SS}
5	5	INT/PC0	PHASE signal (PHASE)	External interrupt, Digital I/O
6	6	T0OUT/PC1		Timer/PWM 0 output, Digital I/O
7	7	T1OUT/PC2		Timer/PWM 1 output, Digital I/O
8	8	T2OUT/PC3		Timer/PWM 2 output, Digital I/O
9	9	TX/PC4		SCI Output, Digital I/O
10	10	RX/PC5		SCI Input, Digital I/O
11	11	PC6		Digital I/O
12	12	PC7		Digital I/O
13		nc		
14	13	Ain0/PB0	Address Reset (RST_ADD)	Analog Input, Digital I/O
15	14	Ain1/PB1	Address Increment (INC_ADD)	Analog Input, Digital I/O
16	15	Ain2/PB2	Configuration Reset (RST_CONF)	Analog Input, Digital I/O
17	16	Ain3/PB3	Configuration Increment (INC_CONF)	Analog Input, Digital I/O
18	17	V_{DDA}	Analog Power Supply	Analog Power Supply
19	18	GNDA	Analog Ground	Analog Ground
20	19	Ain4/PB4		Analog Input, Digital I/O
21	20	Ain5/PB5		Analog Input, Digital I/O
22		nc		
23	21	Ain6/PB6		Analog Input, Digital I/O
24	22	Ain7/PB7/PA7	I/O EPROM Data	Analog Input, Digital I/O
25	23	PA6	I/O EPROM Data	Digital I/O
26	24	T0CLK/PA5	I/O EPROM Data	Timer/PWM 0 clock, Digital I/O
27	25	T0STRT/PA4	I/O EPROM Data	Timer/PWM 0 start/stop, Digital I/O
28	26	$\overline{\text{T2OUT}}$/PA3	I/O EPROM Data	Timer/PWM 2 compl. output, Digital I/O
29	27	$\overline{\text{T1OUT}}$/PA2	I/O EPROM Data	Timer/PWM 1 compl. output, Digital I/O
30	28	$\overline{\text{T0OUT}}$/PA1	I/O EPROM Data	Timer/PWM 0 compl. output, Digital I/O
31	29	T0RES/PA0	I/O EPROM Data	Timer/PWM 0 Reset, Digital I/O
32	30	V_{PP}	EPROM Programming Power supply (12V ± 5%)	EPROM V_{DD} or Vss
33	31	V_{SS}	Digital Ground	Digital Ground
34	32	V_{DD}	Digital Power Supply	Digital Power Supply

Table 1.2 ST52x430 TQFP32 Pin list

TQFP32 Pins	NAME	Programming Phase	Working Phase
1	INT/PC0	PHASE signal (PHASE)	External interrupt, Digital I/O
2	T0OUT/PC1		Timer/PWM 0 output, Digital I/O
3	T1OUT/PC2		Timer/PWM 1 output, Digital I/O
4	T2OUT/PC3		Timer/PWM 2 output, Digital I/O
5	TX/PC4		SCI Output, Digital I/O
6	RX/PC5		SCI Input, Digital I/O
7	PC6		Digital I/O
8	PC7		Digital I/O
9	Ain0/PB0	Address Reset (RST_ADD)	Analog Input, Digital I/O
10	Ain1/PB1	Address Increment (INC_ADD)	Analog Input, Digital I/O
11	Ain2/PB2	Configuration Reset (RST_CONF)	Analog Input, Digital I/O
12	Ain3/PB3	Configuration Increment (INC_CONF)	Analog Input, Digital I/O
13	V_{DDA}	Analog Power Supply	Analog Power Supply
14	GNDA	Analog Ground	Analog Ground
15	Ain4/PB4		Analog Input, Digital I/O
16	Ain5/PB5		Analog Input, Digital I/O
21	Ain6/PB6		Analog Input, Digital I/O
22	Ain7/PB7/PA7	I/O EPROM Data	Analog Input, Digital I/O
19	PA6	I/O EPROM Data	Digital I/O
20	T0CLK/PA5	I/O EPROM Data	Timer/PWM 0 clock, Digital I/O
21	T0STRT/PA4	I/O EPROM Data	Timer/PWM 0 start/stop, Digital I/O
22	$\overline{\text{T2OUT}}$/PA3	I/O EPROM Data	Timer/PWM 2 compl. output, Digital I/O
23	$\overline{\text{T1OUT}}$/PA2	I/O EPROM Data	Timer/PWM 1 compl. output, Digital I/O
24	$\overline{\text{T0OUT}}$/PA1	I/O EPROM Data	Timer/PWM 0 compl. output, Digital I/O
25	T0RES/PA0	I/O EPROM Data	Timer/PWM 0 Reset, Digital I/O
26	V_{PP}	EPROM Programming Power supply (12V ± 5%)	EPROM V_{DD} or Vss
27	V_{SS}	Digital Ground	Digital Ground
28	V_{DD}	Digital Power Supply	Digital Power Supply
29	RESET	General Reset	General Reset
30	OSCOUT		Oscillator Output
31	OSCIN		Oscillator Input
32	TEST	Must be tied to V_{SS}	Must be tied to V_{SS}

1.3 Pin Description

V_{DD}, V_{SS}, V_{DDA}, GNDA, V_{PP}. In order to avoid noise disturbances, the power supply of the digital part is kept separate from the power supply of the analog part.

V_{DD}. Main Power Supply Voltage (5V± 10%).

V_{SS}. Digital circuit ground.

V_{DDA}. Analog V_{DD} of the Analog to Digital Converter.

GNDA. Analog V_{SS} of the Analog to Digital Converter. **Must be tied to V_{SS}**.

V_{PP}. Main Power Supply for internal EPROM (12.5V±5%, in programming phase) and MODE selector. During the Programming phase (programming), V_{PP} must be set at 12V. In the Working phase V_{PP} must be equal to **V_{SS}**.

OSCin and **OSCout.** These pins are internally connected with the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operations of ST52x430 with various stability/cost trade-off. An external clock signal can be applied to OSCin, in this case OSCout must be floating.

RESET. This signal is used to restart ST52x430 at the beginning of its program. It also allows one to select the program mode for EPROM.

Ain0-Ain8. These 8 lines are connected to the input of the analog multiplexer. They allow the acquisition of 8 analog input. During the Programming phase, Ain0, Ain1, Ain2 and Ain3 are used to manage EPROM operation.

PA0-PA7, PB0-PB7, PC0-PC7. These lines are organized as I/O port. Each pin can be configured as input or output. PA7/PB7 are tied to the same output. During Programming phase PA port is used for EPROM read/write data.

T0RES, **T0CLK**, **T0STRT**. These pins are related with the internal Programmable Timer/PWM 0. This Timer can be reset externally by using T0RES. In Working Mode, T0RES resets the address counter of the Timer. T0RES is active at low level.

The Timer 0 Clock can be the internal clock or can be supplied externally by using pin T0CLK.

An external Start/Stop signal can be used to control the Timer through T0STRT pin.

T0OUT, T1OUT, T2OUT. The TIMER/PWM outputs are available on these pins.

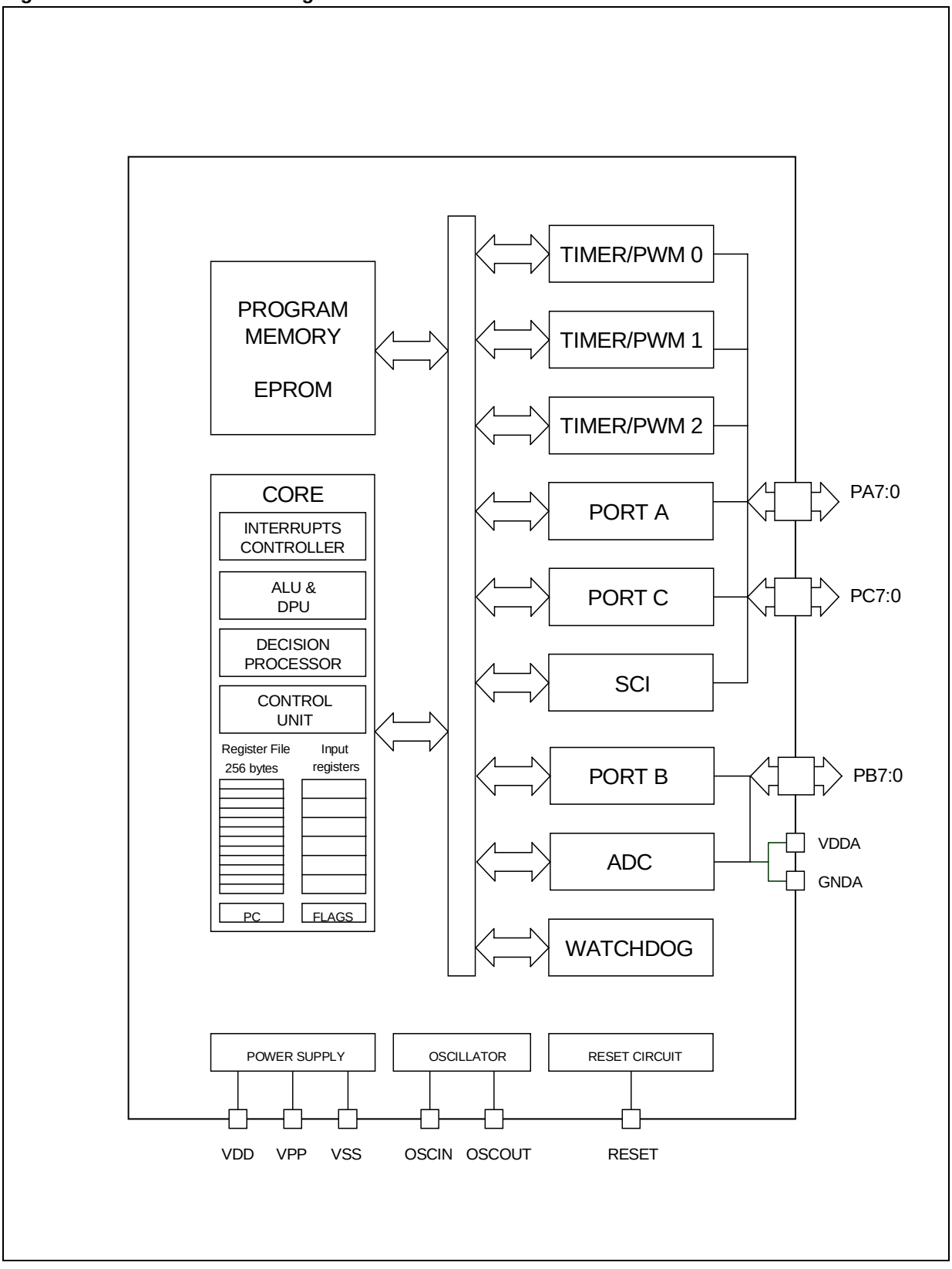
$\overline{\text{T0OUT}}$, **$\overline{\text{T1OUT}}$**, **$\overline{\text{T2OUT}}$**. The TIMER/PWM complementary outputs are available on these pins.

Tx. Serial data output of SCI transmitter block

Rx. Serial data input of the SCI receiver block.

TEST. During the Programming and Working phase it **must be set to Vss**.

Figure 1.4 ST52X430 Block Diagram



2 INTERNAL ARCHITECTURE

ST52x430 is made up of the following blocks and peripherals:

- Control Unit (CU) and Data Processing Unit (DPU)
- ALU / Fuzzy Core
- EPROM
- 256 Byte RAM
- Clock Oscillator
- Analog Multiplexer and A/D Converter
- 3 PWM / Timers
- SCI
- Digital I/O port

2.1 ST52x430 Operating Modes

ST52x430 works in two modes, Programming and Working Modes, depending on the control signals level RESET, TEST and V_{PP}

The Operating modes are selected by setting the control signal level as specified in the Control Signals Setting table.

Table 2.1 Control Signals Setting

Control Signal	Pro-gramming	Reset	Working
RESET	V_{SS}	V_{SS}	V_{DD}
TEST	V_{SS}	V_{SS}	V_{SS}
V_{PP}	12 V	V_{SS}	V_{SS}

2.2 Control Unit and Data Processing Unit

The Control Unit (CU) formally includes five main blocks. Each block decodes a set of instructions, generating the appropriate control signals. The main parts of the CU are illustrated in Figure 2.1.

The five different parts of the CU manage Loading, Logic/Arithmetic, Jump, Control and the Fuzzy instruction set.

The block called “Collector” manages the signals deriving from the different parts of the CU, defining the signals for the Data Processing Unit (DPU) and the different peripherals of the microcontroller.

The block called “Arbiter” manages the different-parts of the CU so that only one part of the system is activated during working mode.

The CU structure is very flexible. It was designed with the purpose of easily adapting the core of the microcontroller to market needs. New instruction sets or new peripherals can be easily included without changing the structure of the microcontroller, maintaining code compatibility.

The CU reads the instructions stored on EPROM (Fetch) and decodes them. According to the instruction types, the arbiter activates one of the main blocks of the CU. Afterwards, all the control signals for the DPU are generated.

A set of 46 different arithmetic, fuzzy and logic instructions is available. Each instruction requires 6 (fuzzy instructions) to 26 (DIVISION) clock pulses to be performed.

The DPU receives, stores and sends instructions deriving from EPROM, RAM or peripherals in order to execute them.

2.2.1 Program Counter.

The Program Counter (PC) is a 13-bit register that contains the address of the next memory location to be processed by the core. This memory location may be an opcode, operand, or an address of an operand.

The 13-bit length allows direct addressing of a maximum of 8,192 bytes in the program space.

After having read the current instruction address, the PC value is incremented. The result of this operation is shifted back into the PC.

The PC can be changed in the following ways:

- JP (Jump)PC = Jump Address
- InterruptPC = Interrupt Vector
- RETIPC = Pop (stack)
- RETPC = Pop (stack)
- CALLPC = Subroutines address
- ResetPC = Reset Vector
- Normal InstructionPC = PC + 1

2.2.2 Flags.

The ST52x430 core includes a different set of flags that correspond to 2 different modes: normal mode and interrupt mode. Each set of flags consists of a CARRY flag (C), ZERO flag (Z) and SIGN flag (S).

One set (CN, ZN, SN) is used during normal operation and one is used during interrupt mode (CI, ZI, SI). **Formally, the user has to manage only one set of flags: C, Z and S**.

Figure 2.1 Data Processing Unit (DPU)

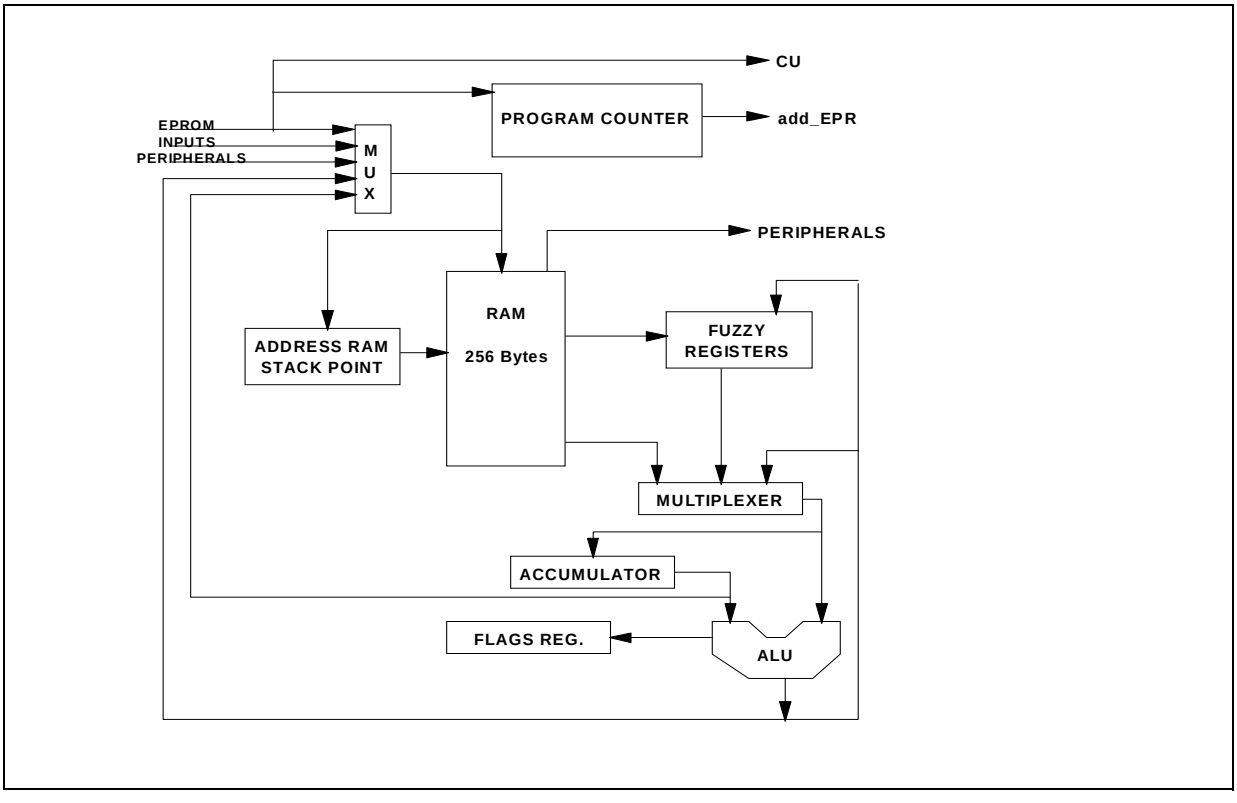
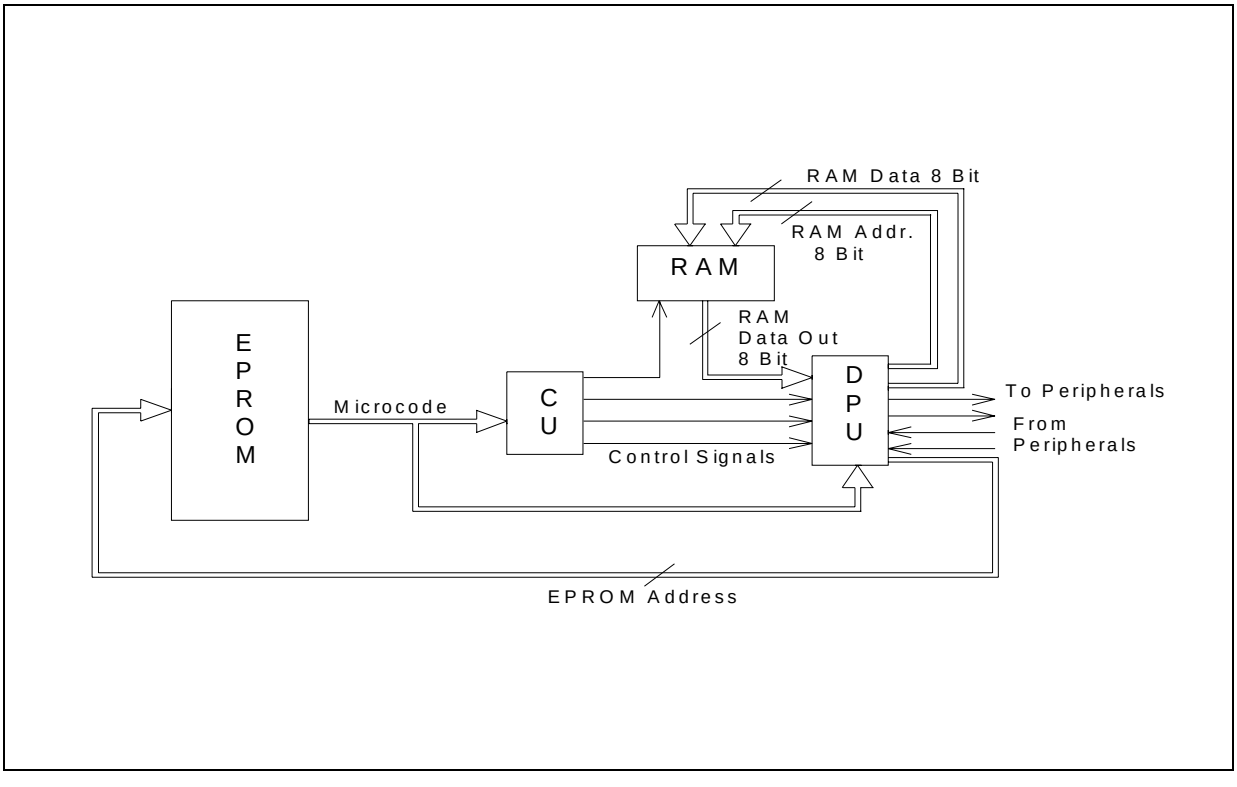


Figure 2.2 CU/DPU Block Diagram



The ST52x430 core uses flags that correspond to the actual mode. As soon as an interrupt is generated the ST52x430 core uses the interrupt flags instead of the normal flags.

Each interrupt level has its own set of flags, which is saved in the STACK together with the Program Counter. These flags are restored from the STACK automatically when a RETI instruction is executed.

If the MCU was in normal mode before an interrupt, the normal flags are restored when the RETI instruction is executed.

Note: *A CALL subroutine is a normal mode execution. For this reason, a RET instruction, consequent to a CALL instruction does not affect the normal mode set of flags.*

Flags are not cleared during context switching and remain in the state they were at the end of the last interrupt routine switching.

The Carry flag is set when an overflow occurs during arithmetic operations, otherwise it is cleared.

The Sign flag is set when an underflow occurs during arithmetic operations, otherwise it is cleared.

2.3 Address Spaces

ST52x430 has four separate address spaces:

- RAM: 256 Bytes
- Input Registers: 20 8-bit registers
- Output Registers 10 8-bit registers
- Configuration Registers: 21 8-bit registers
- Program memory up to 8K Bytes

Program memory will be described in further details in the MEMORY section

2.3.1 RAM and STACK.

RAM memory consists of 256 general purpose 8-bit RAM registers.

All the registers in RAM can be specified by using a decimal address. For example, 0 identifies the first register of RAM.

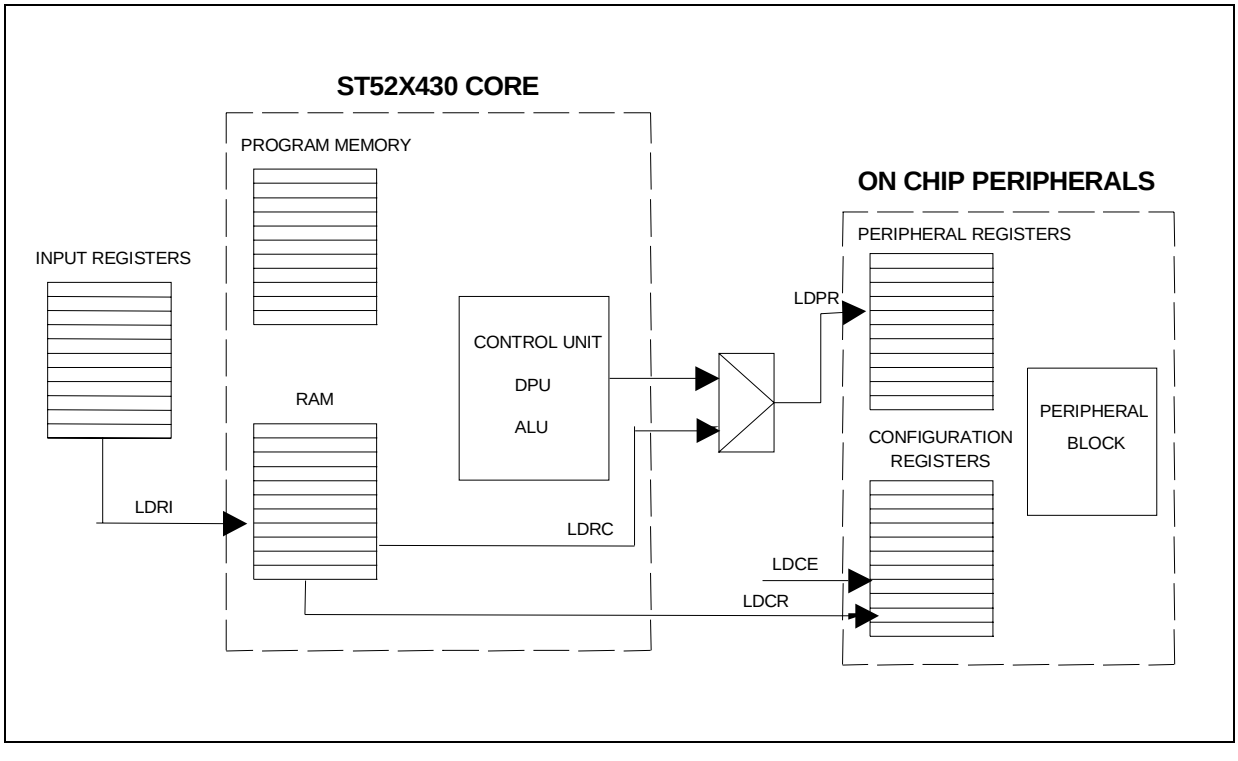
To read or write RAM registers LOAD instructions must be used. See Table 2.5

Each interrupt level has its own set of flags, which is saved in the STACK together with the Program Counter. These flags are restored from the STACK automatically when a RETI instruction is executed.

When the instructions like Interrupt request or CALL are executed, a STACK level is used to push the PC.

The STACK is located in RAM. For each level of stack, 2 bytes of RAM are used. The values of this stack are stored from the last RAM register (address 255). **The maximum level of stack must be less than 128**.

Figure 2.3 Address Spaces Description



The STACK POINTER indicates the first level available to store data. When a subroutine call or interrupt request occurs, the content of the PC and the current set of flags are stored into the level located by the STACK POINTER.

When a interrupt return occurs (RETI instruction), the data stored in the highest stack level is restored back into the PC and current flags.

Instead, when a subroutine return occurs (RET instruction) the data stored in the highest stack level are restored in the PC not affecting the flags.

These operating modes are illustrated in Figure 2.4.

Note: *The user must pay close attention to avoid overwriting RAM locations where the STACK could be stored*.

2.3.2 Input Registers Bench.

The Input Registers (IR) bench consists of 20 8-bit registers containing data or the status of the peripherals.

All the registers can be specified by using a decimal address (for example, 0 identifies the first register of the IR).

The assembler instruction:

```
LDRI RAM_Reg. IR_i
```

loads the value of the i-th IR in the RAM location identified by the RAM_Reg address.

The first input register is dedicated to store the value of the stack pointer. The next 8 registers (ADC_OUT_0:7) of the IR are dedicated to the 8 converted values deriving from the ADC. The last 9 Input Registers contain data from the I/O ports and PWM/Timers. The following table summarizes the IR address and the relative peripherals. In order to simplify the concept, a mnemonic name is assigned to the registers. The same name is used in VISUALSTUDIO® development tools

Figure 2.4 Stack Operation

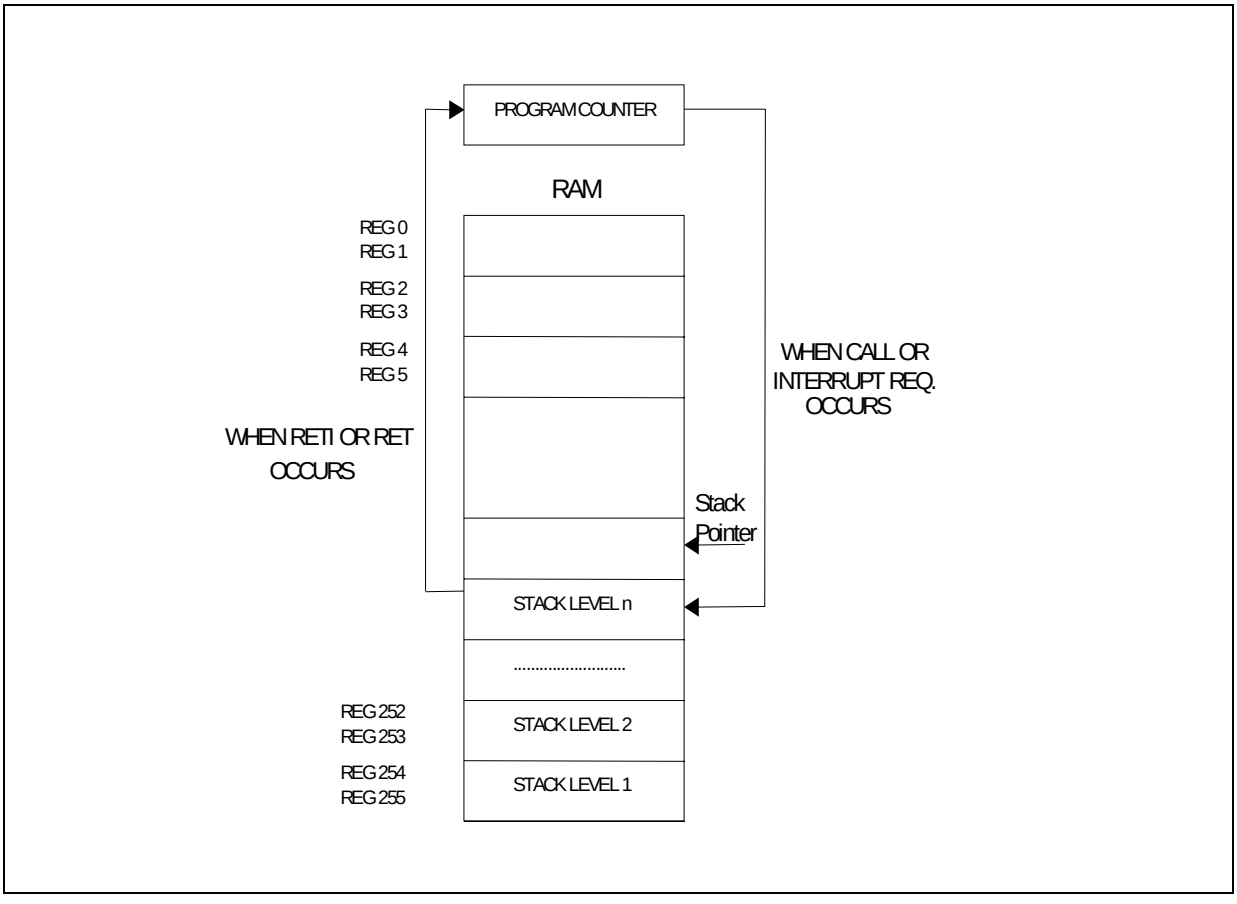


Table 2.2 Input Registers

IR MNEMONIC NAME	PERIPHERAL REGISTER	ADDRESS
STACK_POINTER	STACK POINTER	0
CHAN 0	A/D CHANNEL 0	1
CHAN 1	A/D CHANNEL 1	2
CHAN 2	A/D CHANNEL 2	3
CHAN 3	A/D CHANNEL 3	4
CHAN 4	A/D CHANNEL 4	5
CHAN 5	A/D CHANNEL 5	6
CHAN 6	A/D CHANNEL 6	7
CHAN 7	A/D CHANNEL 7	8
PORT_A	PORT A INPUT REGISTER	9
PORT_B	PORT B INPUT REGISTER	10
PORT_C	PORT C INPUT REGISTER	11
PWM_ 0_COUNT	PWM/TIMER 0 COUNTER	12
PWM_ 0_ STATUS	PWM/TIMER 0 STATUS REGISTER	13
PWM_ 1_ COUNT	PWM/TIMER 1 COUNTER	14
PWM_ 1_ STATUS	PWM/TIMER 1 STATUS REGISTER	15
PWM_ 2_ COUNT	PWM/TIMER 2 COUNTER	16
PWM_ 2_ STATUS	PWM/TIMER 2 STATUS REGISTER	17
SCI_RX	SCI DATA REGISTER	18
SCI_STATUS	SCI STATUS REGISTER	19

2.3.3 Configuration Registers.

The ST52x430 configuration Registers allow the configuration of all the blocks of the fuzzy microcontroller. Table 2.3 describes the functions and the related peripherals of each of the Configuration Registers. By using the load instructions, the Configuration Registers can be set by using values stored in the Program Memory (EPROM) or in RAM.

Use and meaning of each register will be described in further details in the corresponding section.

Table 2.3 Configuration Registers

CONFIGURATION REGISTER	PERIPHERAL	DESCRIPTION
REG_CONF 0	INTERRUPT MASK	Interrupts mask setting
REG_CONF 1	N.U.	N.U.
REG_CONF 2	WATCHDOG TIMER	Watchdog Timer Configuration
REG_CONF 3	A/D CONVERTER	A/D configuration

Table 2.3 Configuration Registers (continued)

CONFIGURATION REGISTER	PERIPHERAL	DESCRIPTION
REG_CONF 4	PORT A	Set the relative bit like digital input or digital output
REG_CONF 5	PWM/TIMER 0	PWM/Timer 0 Working mode Configuration
REG_CONF 6	PWM/TIMER 0	PWM/TIMER 0 Prescaler configuration and output waveform selection.
REG_CONF 7	PWM/TIMER 0	PWM/TIMER 0 Working Mode Configuration
REG_CONF 8	PWM/TIMER 1	PWM/TIMER 1 Working Mode Configuration
REG_CONF 9	PWM/TIMER 1	PWM/TIMER 1 Prescaler configuration and output waveform selection.
REG_CONF 10	PWM/TIMER 2	PWM/TIMER 2 Working Mode Configuration
REG_CONF 11	PWM/TIMER 2	PWM/Timer 2 Prescaler configuration and output waveform selection.
REG_CONF 12	PORT A	Set the bit 0,1 and 2 like Digital I/O or complementary Timers Output.
REG_CONF 13	PORT B	Set the relative bit like digital input or digital output.
REG_CONF 14	PORT B	Set the relative I/O like Digital or Analog.
REG_CONF 15	PORT C	Set the relative I/O like digital input or digital output
REG_CONF 16	PORT C	Set the relative I/O like Digital I/O or Timers Output
REG_CONF 17	Interrupt Priority	Set the Interrupts priority
REG_CONF 18	Interrupt Priority	Set the Interrupts priority
REG_CONF 19	SCI	Set the SCI working mode
REG_CONF 20	SCI	Set the SCI working mode

2.3.4 Output Registers.

The Output Registers (OR) consist of 10 registers containing data for the microcontroller peripherals including the I/O Ports.

All registers can be specified by using a decimal address (for example, 1 identifies the second OR).

By using LOAD instructions the Output Registers (OR) may be set by using values stored in the Program Memory (LDPE) or in RAM (LDPR)

The assembler instruction:

```
LDPR OR_i RAM_Reg.
```

loads the value of the RAM location identified by the address RAM_Reg in the OR i-th Table 2.4 describes OR.

In order to simplify the concept, a mnemonic name is assigned to OR. The same names are used in FUZZYSTUDIO™ 4.0 development tools.

Use and meaning of each register will be described in further details in the corresponding section.

Table 2.4 Output Registers

OR MNEMONIC NAME	PERIPHERAL REGISTER	ADDRESS
PORT_ A	PORT A OR	0
PORT_ B	PORT B OR	1
PORT_C	PORT C OR	2
PWM_0_COUNT	TIMER/PWM 0 COUNTER	3
PWM_0_RELOAD	TIMER/PWM 0 RELOAD REGISTER	4
PWM_1_COUNT	TIMER/PWM 1 COUNTER	5
PWM_1_RELOAD	TIMER/PWM 1 RELOAD REGISTER	6
PWM_ 2_ COUNT	TIMER/PWM 2 COUNTER	7
PWM_2_RELOAD	TIMER/PWM 2 RELOAD REGISTER	8
SCI_TX_DATA	SCI DATA REGISTER	9

2.4 Arithmetic Logic Unit

The 8-bit Arithmetic Logic Unit (ALU) allows the performance of arithmetic calculations and logic instructions, which can be divided into 5 groups: Load, Arithmetic, Jump, Interrupts and Program Control instructions (refer to the ST52x430 Assembler Set for further details).

The computational time required for each instruction consists of one clock pulse for each Cycle plus 3 clock pulses for the decoding phase.

The ALU of the ST52x430 can perform multiplication (MULT) and division (DIV). Multiplication is performed by using 8 bit operands storing the result in 2 registers (16 bit values), see Figure 2.5 and Figure 2.6.

WARNING: If the LSB of the multiplication result is 0, the Zero flag is set although the result is not 0.

Table 2.5 Load instructions

Load Instructions						
Mnemonic	**Instruction**	**Bytes**	**Cycles**	**Z**	**S**	**C**
LDCE	LDCE conf, EPROM	3	17	-	-	-
LDCR	LDCR conf, RAM	3	14	-	-	-
LDFR	LDFR FUZZY_i_RAM RAM	3	14	-	-	-
LDPE	LDPE per, EPROM	3	17	-	-	-
LDPE	LDPE per, (RAM)	3	17	-	-	-
LDPR	LDPR reg, RAM	3	14	-	-	-
LDRC	LDRC RAM, const	3	14	-	-	-
LDRE	LDRE RAMi, EPROMi	3	16	-	-	-
LDRE	LDRE (RAMi), (RAMj)	3	18	-	-	-
LDRI	LDRI RAM, inp_reg	3	15	-	-	-
LDRR	LDRR RAMi, RAMj	3	16	-	-	-

Table 2.5 Load instructions

PGSET	PGSET const	2	9	-	-	-

Table 2.6 Arithmetic & Logic instructions set

Arithmetic Instructions						
Mnemonic	**Instruction**	**Bytes**	**Cycles**	**Z**	**S**	**C**
ADD	ADD regi, regj	3	17	I	-	I
ADDO	ADDO regi, regj	3	20	I	I	I
AND	AND regi, regj	3	17	I	-	-
ASL	ASL regi	2	15	I	-	I
ASR	ASR regi	2	15	I	I	-
DEC	DEC regi	2	15	I	I	-
DIV	DIV regi, regj	3	26	I	I	I
INC	INC regi	2	15	I	-	I
MULT	MULT regi, regj	3	19	I	-	-
NOT	NOT regi	2	15	I	-	-
OR	OR regi, regj	3	17	I	-	-
SUB	SUB regi, regj	3	17	I	I	-
SUBO	SUBO regi, regj	3	20	I	I	I
MIRROR	MIRROR regi	2	15	I	-	-

Table 2.7 Jump Instruction Set

Jump instructions						
mnemonic	**instruction**	**bytes**	**cycles**	**z**	**s**	**c**
call	call addr	3	18	-	-	-
jp	jp addr	3	12	-	-	-
jpc	jpc addr	3	10/12	-	-	-
jpnc	jpnc addr	3	10/12	-	-	-
jpns	jpns addr	3	10/12	-	-	-
jpnz	jpnz addr	3	10/12	-	-	-
jps	jps addr	3	10/12	-	-	-
jpz	jpz addr	3	10/12	-	-	-
ret	ret	1	13	-	-	-

Table 2.8 Interrupt Instructions Set

Interrupt Instructions						
Mnemonic	**Instruction**	**Bytes**	**Cycles**	**Z**	**S**	**C**
HALT	HALT	1	7/15	-	-	-
MEGI	MEGI	1	7/15	-	-	-

Table 2.8 Interrupt Instructions Set (continued)

MDGI	MDGI	1	6	-	-	-
RETI	RETI	1	12	-	-	-
RINT	RINT INT	2	8	-	-	-
UDGI	UDGI	1	6	-	-	-
UEGI	UEGI	1	7/15	-	-	-
WAITI	WAITI	1	7/14	-	-	-

Table 2.9 Control Instructions Set

Control Instructions						
Mnemonic	**Instruction**	**Bytes**	**Cycles**	**Z**	**S**	**C**
FUZZY	FUZZY	1	5	-	-	-
NOP	NOP	1	6	-	-	-
WDTRFR	WDTRFR	1	7	-	-	-
WDTSLP	WDTSLP	1	6	-	-	-

Notes:
I affected
- not affected

Figure 2.5 Multiplication

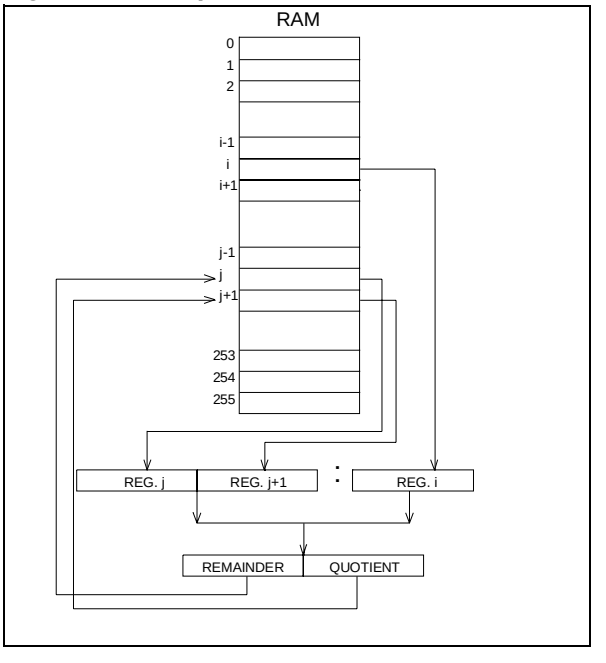
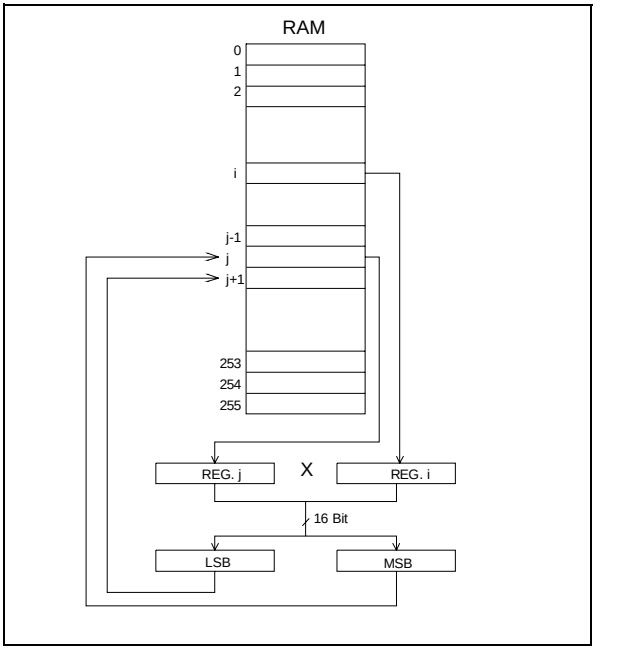


Figure 2.6 Division



3 EPROM

EPROM memory provides an on-chip user-programmable non-volatile memory, which allows fast and reliable storage of user data.

EPROM memory can be locked by the user. In fact, a memory location called Lock Cell is devoted to lock EPROM and avoid external operations. A software identification code, called ID CODE, distinguishes which software version is stored in the memory.

64 kbits of memory space with an 8-bit internal parallelism (up to 8 kbytes) addressed by an 13-bit bus are available. The data bus is 8 bits.

Memory has a double supply: V_{PP} is equal to 12V±5% in Programming Phase or to V_{SS} during Working Phase. V_{DD} is equal to 5V±10%.

ST52x430 EPROM memory is divided into three main blocks (see Figure):

- *Interrupt Vectors memory block* (3 through 20) contains the addresses for the interrupt routines. Each address is composed of three bytes.
- *Mbfs Setting memory block* (21 through *MemAdd*) contains the coordinates of the vertexes of every Mbf defined in the program.
- The maximum value of MemAdd is 1023. This area is dynamically assigned according to the size of the fuzzy routines. The unused memory area, if any, is assigned to the Program Instruction Set memory block.
- The Program Instructions Set memory block (MemAdd through MemAddx) contains the instruction set of the user program. The following table summarizes the values of Mem Addx for the different devices

Table 3.1 Mem Addx

	ST52T430K1	ST52T430	ST52T430
Mem	2047	4095	8191

Locations 0, 1 and 2 contain the address of the first microcode instruction.The operations that can be performed on EPROM during the Programming Phase are: Stand By, Memory Writing, Reading and Verify/Margin Mode, Memory Lock, IDCode Writing and Verify.

Figure 3.1 Program Memory Organization

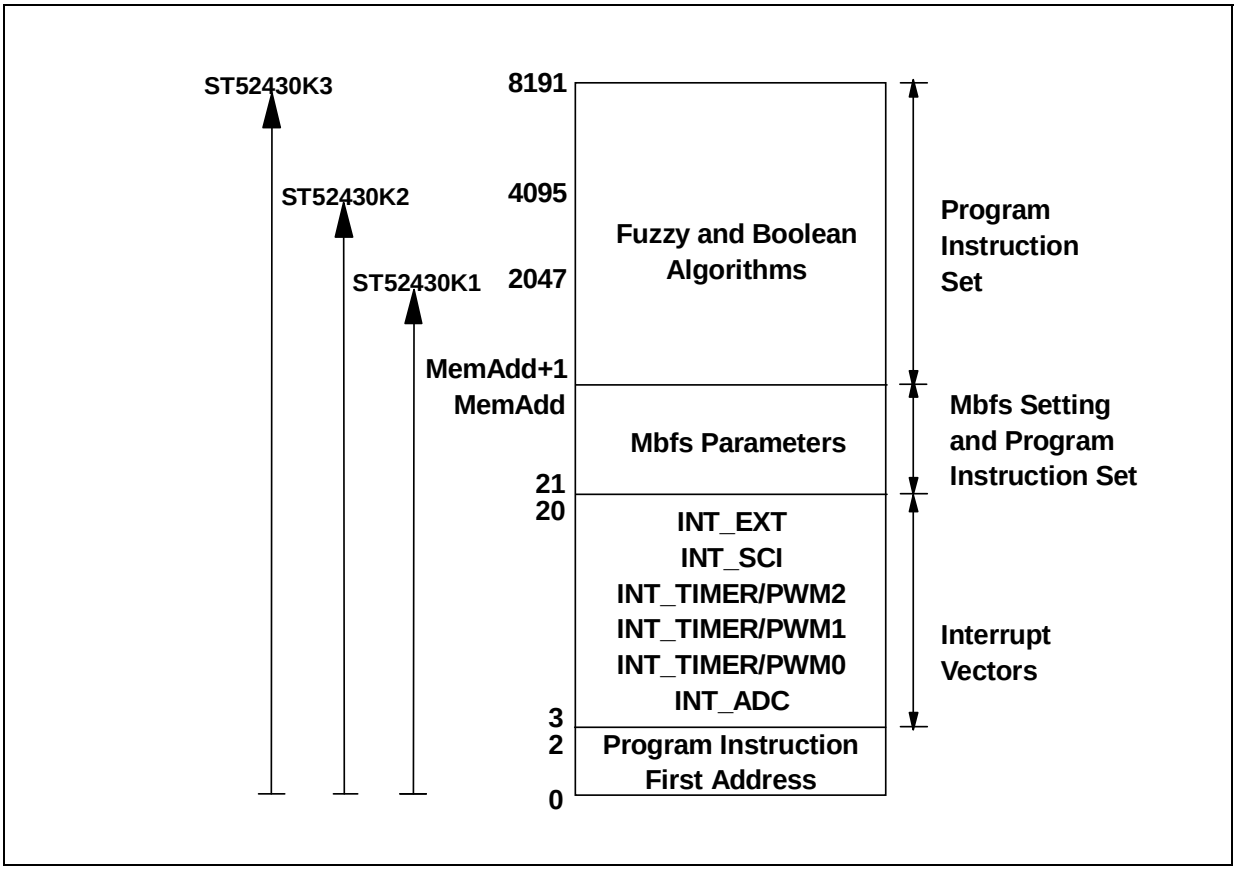


Table 3.2 EPROM Control Register

OPERATION	REGISTER VALUE
Stand By	0
Memory Reading/Verify	1
Memory Unlock and Lock Status Reading	2
Memory Writing	3
Memory Lock	4
ID CODE Writing	5
Memory Lock Status Reading/Verify	9
ID CODE Reading/ Verify	10

The operations above are managed by using the internal 4-bit EPROM Control Register. The reading phase is executed with V_{PP}= 5V±5%, while the verify/Margin Mode phase needs V_{PP}= 12V±5%. The Blank Check must be a reading operation with V_{PP}= 5V±5%.

Table 3.2 illustrates EPROM Control Register codes used to identify the operation running.

3.1 EPROM Programming Phase Procedure

The Programming mode is selected by applying 12V±5% voltage or 5V±5% voltage to the V_{PP} pin and setting the control signal as following:

RESET =Vss

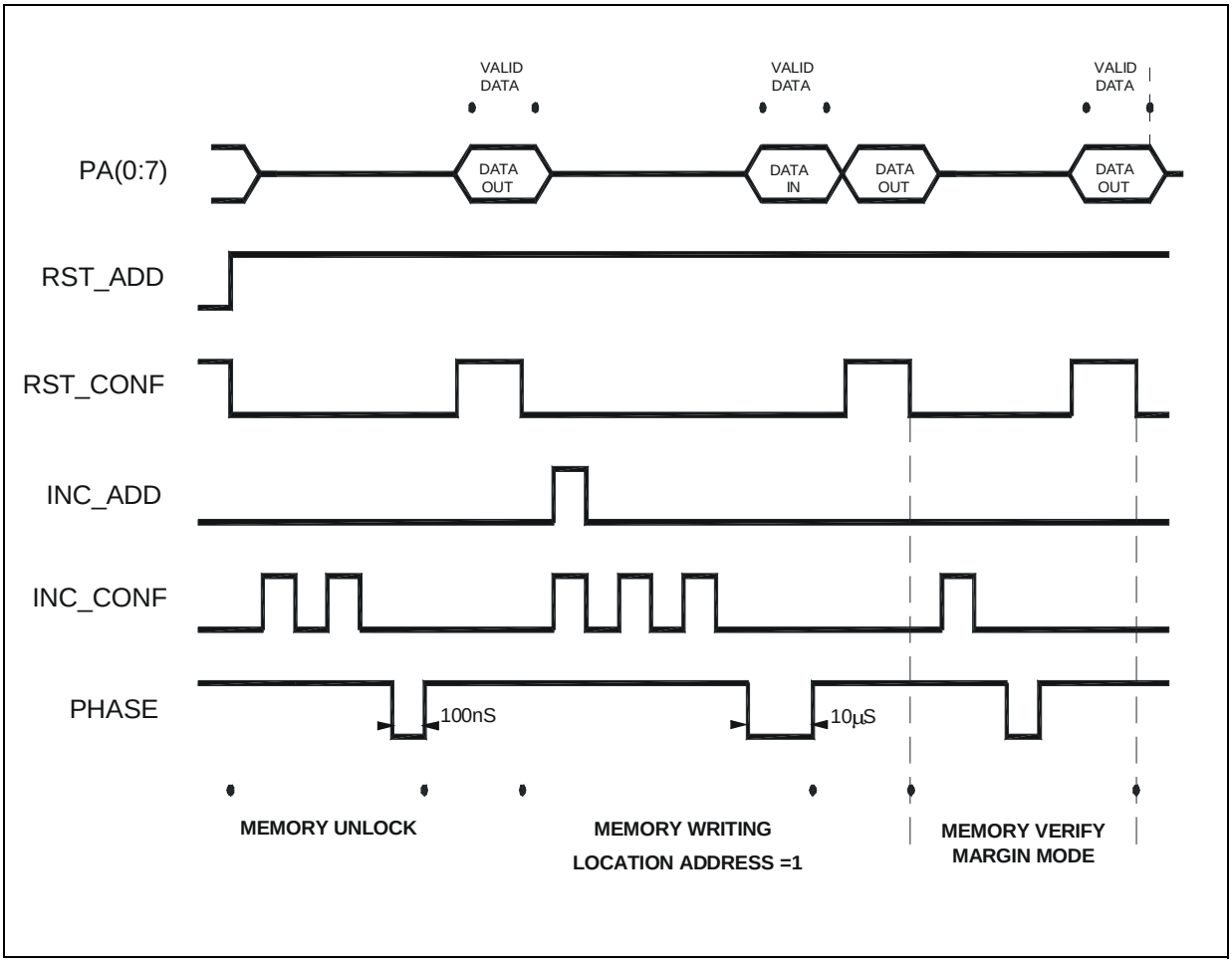
TEST =Vss

If the V_{PP} voltage is 5V±5% only reading may be performed.

RST_ADD, INC_ADD, RST_CONF, INC_CONF and PHASE are the control signals used during the Programming Mode.

PHASE, RST_CONF and RST_ADD signals are active on level, the others are active on rising edge.

Figure 3.2 Eprom Programming Timing



PHASE and RST_ADD signals are active low, RST_CONF signal is active high.

Port A is used for the memory data I/O. ***(See Table 3.2 for pin reference on the different packages)***.

Memory may be locked by means of the Memory Lock Status, which is a flag used to enable EPROM operations.

If Memory Lock Status is 1 all EPROM operations are enabled, otherwise the user may only read (and verify) the OTP code and the Memory Lock Status.

Only if EPROM is not locked by means of Lock Cell (see EPROM Locking may EPROM operations be enabled by changing the Memory Lock Status from 0 to 1.

RST_ADD signal resets the memory address register and the Memory Lock Status. When the RST_ADD becomes high, the memory must be unlocked in order to read or write.

INC_ADD signal increments the memory address.

RST_CONF signal resets the EPROM Control Register. **When RST_CONF is high, the DATA I/O Port A is in output, otherwise it is always in input.**

INC_CONF signal increments the EPROM Control Register value.

PHASE signal validates the operation selected by means of the EPROM Control Register value.

3.1.1 EPROM Operation.

In order to execute an EPROM operation (See Table 3.2), the corresponding identification value must be loaded in the EPROM Control Register. The signal timing is the following: RST_ADD= high and PHASE= high, RST_CONF changes from low to high level, to reset the EPROM Control Register, and INC_CONF signal generates a number of positive pulses equal to the value to be loaded. After this sequence, a negative pulse of the PHASE signal will validate the operation selected. The minimum PHASE signal pulse width must be 10 µs for EPROM Writing Operation and 100 ns for the others.

When RST_CONF is high, DATA I/O Port A is enabled in output and the reading/verifying operation results are available.

After a writing operation, when RST_CONF is high, Port A is in output without valid data.

3.1.2 EPROM Locking.

The Memory Lock operation, which is identified with the number 4 in the EPROM Control Register, writes “0" in the Memory Lock Cell.

At the beginning of an External Operation, when the RST_ADD signal changes from low level to high level, the Memory Lock Status is “0", therefore it must be unlocked before proceeding.

In order to unlock the Memory Lock Status the operation, which is identified by the number 2 in the EPROM Control Register must be executed (see Figure 3.2).

Memory Lock Status can be changed only if Memory Lock Cell is “1". After a Memory Lock operation external operations cannot be executed except to read (or verify) the OTP Code and the Memory Lock Status.

3.1.3 EPROM Writing.

When the memory is blank, all bits are at logic level “1". Data is introduced by programming only the zeros in the desired memory location. However, all input data must contain both ”1" and “0".

The only way to change “0" into ”1" is to erase the entire memory (by exposure to Ultra Violet light) and reprogram it.

The memory is in Writing mode when the EPROM Control Register value is 3.

The V_{PP} voltage must be 12V±5%, with stable data on the data bus PA(0:7).

The timing signals are the following (see Figure):

1) RST_ADD and RST_CONF change from low to high level,

2) two pulses on INC_CONF signal load the Memory Unlock operation code,

3) a negative pulse (100 ns) on the PHASE signal validates the Memory Unlock operation,

4) a negative pulse on RST_CONF signal resets the EPROM Control Register,

5) three positive pulses on INC_CONF load the Memory Writing operation code,

6) a train of positive pulses on INC_ADD signal increments the memory location address up to the requested value (generally this is a sequential operation and only one pulse is used),

7) a negative pulse (10 µs) on the PHASE signal validates the Memory Writing operation.

3.1.4 EPROM Read/Verify Margin Mode.

The read phase is executed with V_{PP}= 5V±5%, instead of the verify phase that needs V_{PP}= 12V±5%.

The Memory Verify operation is available in order to verify the accuracy of the data written. A Memory Verify Margin Mode operation can be executed immediately after writing each byte, in this case (see Figure):

1) a positive pulse on RST_CONF signal resets the EPROM Control Register, if it wasn't already reset;

2) one positive pulse on INC_CONF loads the Memory Read/Verify operation code;

3) a negative pulse (100 ns) on the PHASE signal validates the Memory Reading / Verify operation;

4) a negative pulse on RST_CONF signal puts in the PA(0:7) port the value stored in the actual memory address and resets the EPROM Control Register;

If an error occurred writing, the user has to repeat EPROM writing.

3.1.5 Stand by Mode.

EPROM has a standby mode, which reduces the active current from 10mA (Programming mode) to less than 100 µA. Memory is placed in standby mode by setting the PHASE signal at a high level or when the EPROM Control Register value is 0 and the PHASE signal is low.

3.1.6 ID code.

A software identification code, called ID code may be written in order to distinguish which software version is stored in the memory.

64 Bytes are dedicated to store this code by using the address values from 0 to 63.

The ID Code may be read or verified even if the Memory Lock Status is "0".

The timing signals are the same as that of a normal operation.

3.2 Eprom Erasure

The transparent window available in the CSDIP32W package, allows the memory contents to be erased by exposure to UV light.

Erasure begins when the device is exposed to light with a wavelength shorter than 4000Å. Sunlight, as well as some types of artificial light, includes wavelengths in the 3000-4000Å range which, on prolonged exposure can cause erasure of memory contents. Therefore, it is recommended that EPROM devices be fitted with an opaque label over the window area in order to prevent unintentional erasure.

The erasure procedure recommended for EPROM devices consists of exposure to short wave UV light having a wavelength of 2537Å. The minimum integrated dose recommended (intensity x exposure time) for complete erasure is 15Wsec/cm 2.

This is equivalent to an erasure time of 15-20 minutes using a UV source having an intensity of 12mW/cm 2 at a distance of 25mm (1 inch) from the device window.

4 INTERRUPTS

The Control Unit (CU) responds to peripheral events and external events via its interrupt channels.

When such an events occur, if the related interrupt is not masked and according to a priority order, the current program execution can be suspended to allow the CU to execute a specific response routine.

Each interrupt is associated with an interrupt vector that contains the memory address of the related interrupt service routine. Each vector is located in the Program Space (EPROM Memory) at a fixed address (see Interrupt Vectors Table 4.2).

4.1 Interrupt Operation

If there are pending interrupts at the end of an arithmetic or logic instruction, the one with the highest priority is passed. Passing an interrupt means storing the arithmetic flags and the current PC in the stack and executing the associated Interrupt routine, whose address is located in three bytes of the EPROM memory location between address 3 and 20.

The Interrupt routine is performed as a normal code, checking if a higher priority interrupt has to be passed at the end of each instruction. An Interrupt request with the higher priority stops the lower priority Interrupt. The Program Counter and the arithmetic flags are stored in the stack.

With the RETI (Return from Interrupt) instruction the arithmetic flags and Program Counter (PC) are restored from the top of the stack. This stack was already described in section RAM and STACK.

An Interrupt request cannot stop processing of the fuzzy rule, but this is passed only after the end of a fuzzy rule or at the end of a logic, or arithmetic instruction.

NOTE: A fuzzy routine can only be interrupted in the Main program. An interrupt request cannot stop a Fuzzy function that is running inside another interrupt routine. In order to use a Fuzzy function inside an interrupt routine, the user MUST include the Fuzzy function between an UDGI (MDGI) instruction and an UEGI (MEGI) instruction (see the following paragraphs), so that the interrupt request may be disabled during the execution of the fuzzy function.

4.2 Global Interrupt Request Enabling

When an Interrupt occurs, it generates a Global Interrupt Pending (GIP), that can be masked by software. After a GIP a Global Interrupt Request (GIR) will be generated and Interrupt service

Figure 4.1 Interrupt Flow

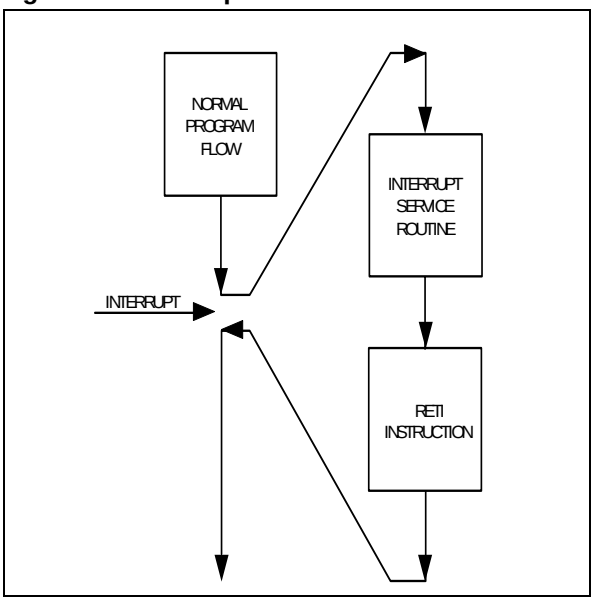


Figure 4.2 Interrupt Vectors mapping

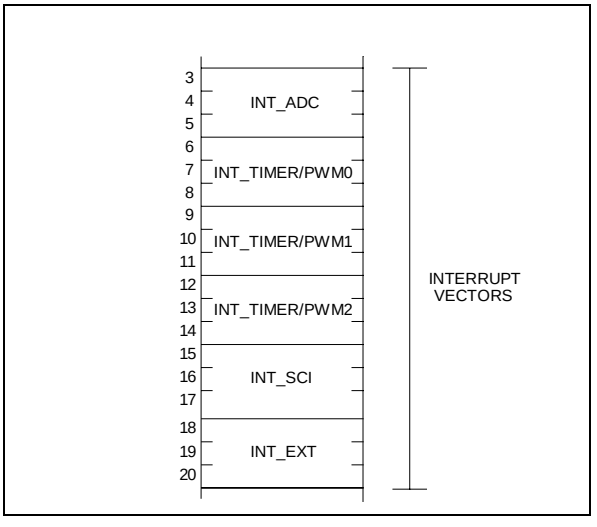
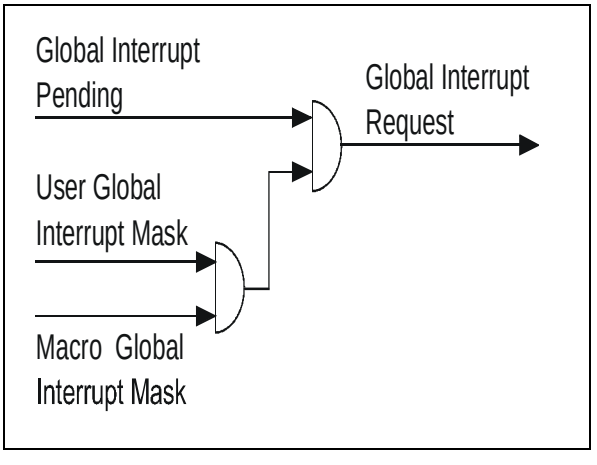


Figure 4.3 Global Interrupt Request generation



Routine associated to the interrupt with higher priority will start.

In order to avoid possible conflicts between interrupt masking set in the main program, or inside high level language compiler macros, the GIP is hung up through the User Global Interrupt Mask or the Macro Global Interrupt Mask (see Figure 4.2).

UEGI/UDGI instruction switches on/off the User Global Interrupt Mask, enabling/disabling the GIR for the main program.

MEGI/MDGI instructions switch the Macro Global Interrupt Mask on/off, in order to ensure that the macro will not be broken.

4.3 Interrupt Sources

ST52x430 manages interrupt signals generated by the internal peripherals (PWM/Timers, UART and Analog to Digital Converter) or coming from the INT/PC0 pin. The External Interrupt can be programmed to be active on the rising or falling edge of INT/PC0 signal by setting the PEXTINT bit of the Configuration Register to 0.

WARNING: *Changing the interrupt priority an interrupt request is generated.*

Each peripheral can be programmed in order to generate the associated interrupt; further details are described in the related chapter.

4.4 Interrupt Maskability

The interrupts can be masked by configuring the REG_CONF 0 by means of LDCR, or LDCE, instruction. The interrupt is enabled when the bit associated to the mask interrupt is “1". Viceversa, when the bit is ”0", the interrupt is masked and is kept pendent.

For example:

```
LDRC 10,6 //load the constant 6 in the
RAM Register 10
LDCR 0, 10 // set the CONF_REG 0 with
the value stored in the RAM Register
10
```

the result is CONF_REG0 =00000110 enabling the interrupts deriving from the ADC (INT_ADC) and from the PWM/TIMER 0 (INT_PWM/TIMER0).

Table 4.1 Configuration Register 0 Description

Bit	Name	Value	Description
0	MSKE	0	External Interrupt Masked
		1	External Interrupt Not Masked
1	MSKAD	0	A/D Converter Interrupt Masked
		1	A/D Converter Interrupt Not Masked
2	MSKTM0	0	PWM/TIMER 0Interrupt Masked
		1	PWM/TIMER 0 Interrupt Not Masked
3	MSKTM1	0	PWM/TIMER 1 Interrupt Masked
		1	PWM/TIMER 1 Interrupt Not Masked
4	MSKTM2	0	PWM/TIMER 2 Interrupt Masked
		1	PWM/TIMER 2 Interrupt Not Masked
5	MSCI	0	SCI Interrupt Masked
		1	SCI Interrupt Not Masked
6	PEXTINT	0	External Interrupt Polarity Active on Rising
		1	External Interrupt Polarity Active on Falling
7	Not used		

Reset Configuration ‘000000’

Table 4.2 Interrupts Description

Name	Description		Priority	Peripheral Code	Maskable	EPROM Locations
INT_ADC	ADC	Int	Programmable	000	yes	3-5
INT_PWM/ TIMER0	PWM/TIMER 0	Int	Programmable	001	yes	6-8
INT_PWM/ TIMER1	PWM/TIMER 1	Int	Programmable	010	yes	9-11
INT_PWM/ TIMER2	PWM/TIMER 2	Int	Programmable	011	yes	12-14
INT_SCI	SCI	Int	Programmable	100	yes	15-17
INT_EXT	External Interrupt (INT)	Ext	Highest	-	yes	18-20

Figure 4.4 Interrupt Configuration Register 0

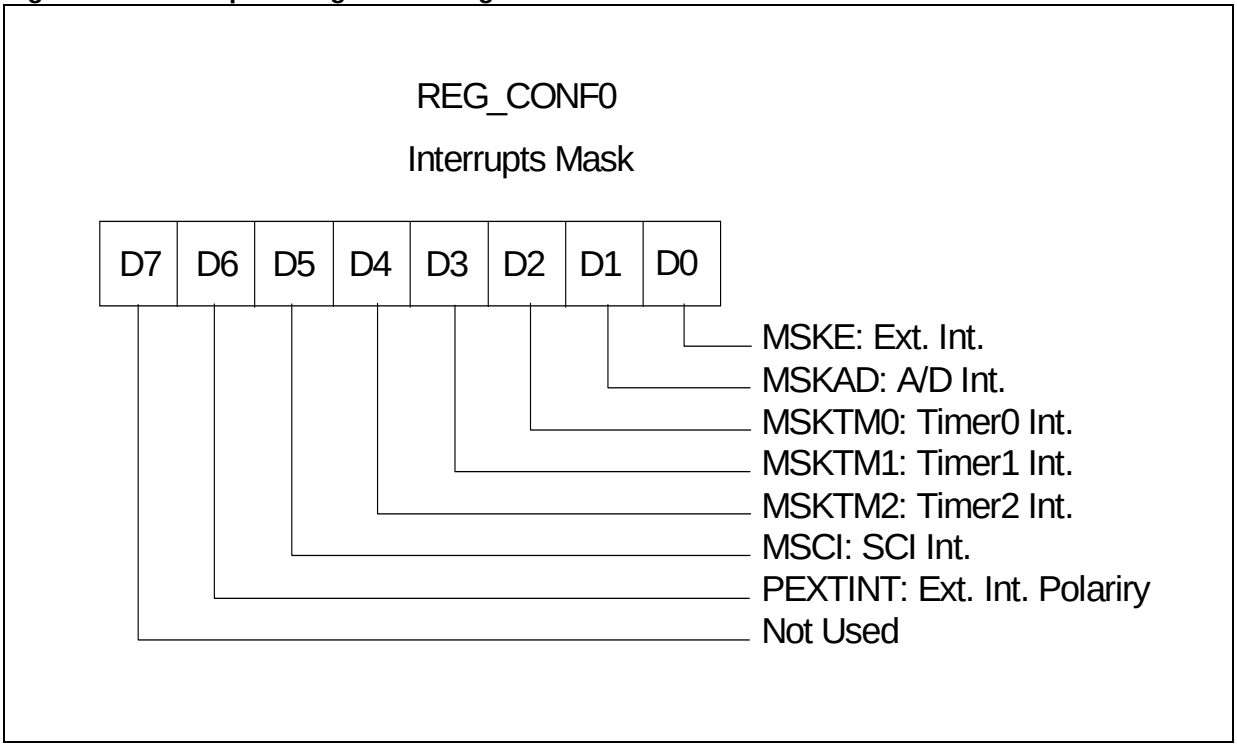
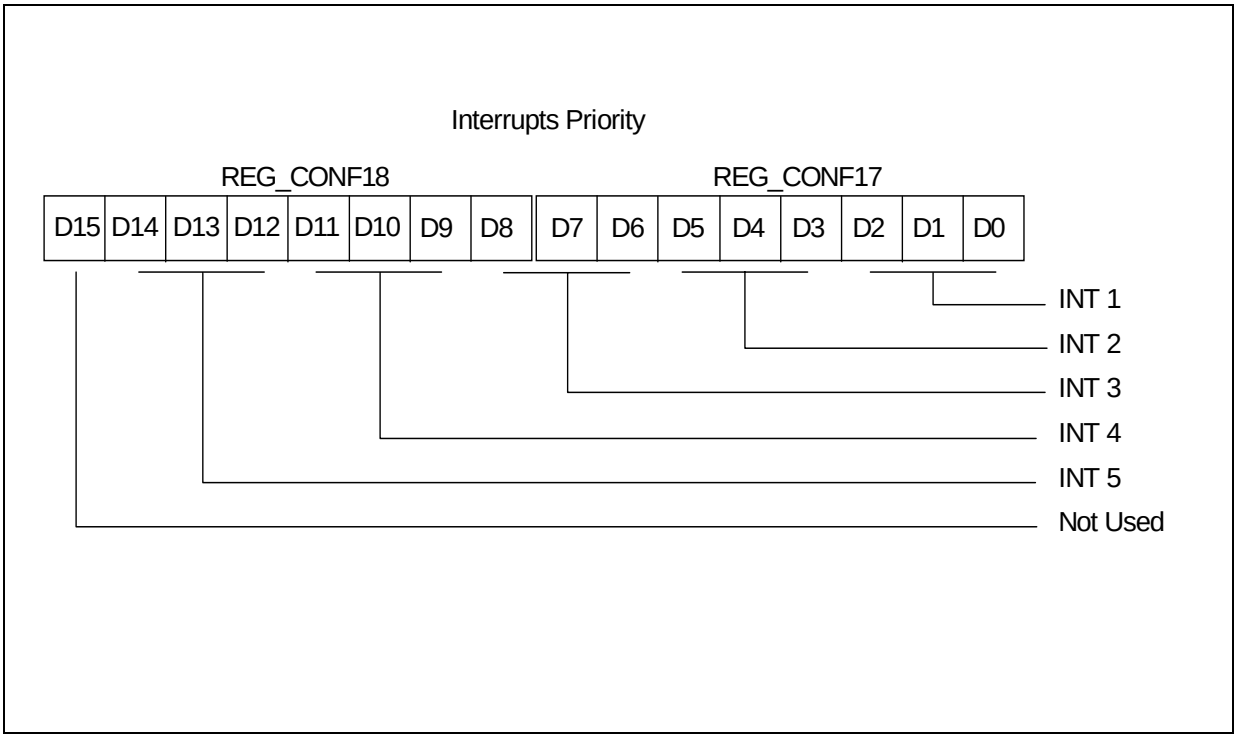


Figure 4.5 Interrupt Configuration Register 17 & 18



4.5 Interrupt Priority

Seven priority levels are available: level 6 has the lowest priority, level 0 has the highest priority.

Level 6 is associated to the Main Program, levels 5 to 1 are programmable by means of the priority registers called REG_CONF17 and REG_CONF18 (see Figure 4.5 and Table 4.3); whereas the higher level is related to the external interrupt (INT_EXT).

PWM/Timers, UART and ADC are identified by a three-bit Peripheral Codes (see Table 4.2); in order to set the *i*-th priority level the user must write the peripheral label *i* in the related INT*i* priority level.

i.e.

```
LDRC 10, 193 //(load the value 193='11000001' in the RAM Register 10)
LDRC 11, 168 //(load the value 168='10101000' in the RAM Register 11)
LDCR 17, 10 // set the REG_CONF17= '11000001'
LDCR 18, 11 // set the REG_CONF18= '10101000'
```

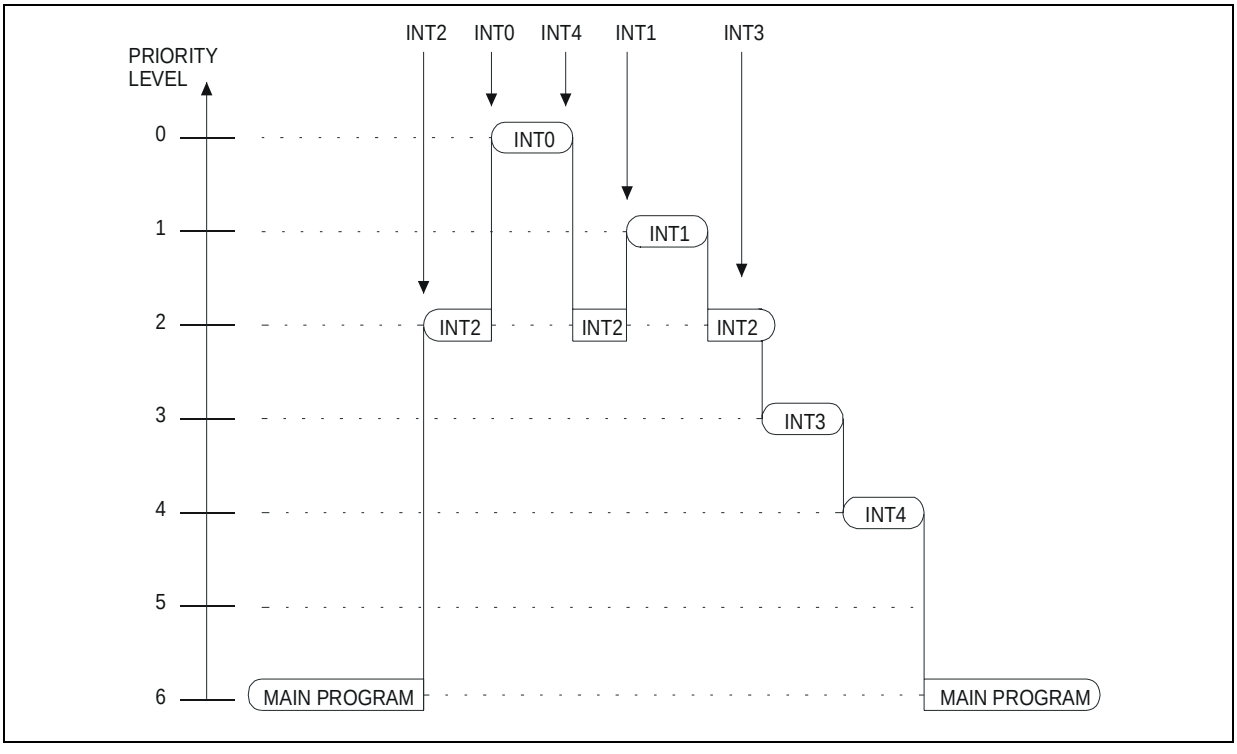
The following priority levels are defined:

- Level 1: INT_PWM/TIMER0 (PWM/TIMER 0 Code: 001)
- Level 2: INT_ADC (ADC Code: 000)
- Level 3: Int_PWM/Timer2 (PWM/TIMER 2 Code: 011)
- Level 4: INT_UART (UART Code: 100)
- Level 5: INT_PWM/TIMER1 (PWM/TIMER 1 Code: 010)

Table 4.3 Conf. Register 17 & 18 Description

Bit	Name	Value	Level
0, 1,2	INT1	Peripheral Code	High
3, 4,5	INT2	Peripheral Code	Medium-High
6,7,8	INT3	Peripheral Code	Medium-Low
9,10,11	INT4	Peripheral Code	Low
12,13,14	INT5	Peripheral Code	Very Low

Figure 4.6 Example of a sequence of Interrupt requests



REMARK: The Interrupt priority must be set at the beginning of the main program, because at the RESET REG_CONF1='00000000', this condition could generate wrong operations. Further, changing the priority levels must be avoided in interrupt service routines.

When a source provides an Interrupt request and the request processing is also enabled, the CU changes the normal sequential flow of a program by transferring program control to a selected service routine.

When an interrupt occurs the CU executes a JUMP instruction to the address loaded in the related location of the Interrupt Vector.

When the execution returns to the original program it immediately begins following the instruction that was interrupted.

Table 4.4 RINT Instruction code

Peripheral Name	Value
A/D Converter	0
PWM/TIMER 0	1

4.6 Interrupts and Low power mode

All interrupts allow the processor to leave the WAIT low power mode. Only the external Interrupt allows the processor to leave the HALT low power mode.

4.7 Interrupt RESET

An eventually pending interrupt can be reset with the instruction `RINT j`, which resets the interrupt *j*-th where *j* identifies the peripherals as described in the following table (see Table 4.4).

The assembler instruction:

```
RINT 2
```

Resets the PWM/Timer 1 interrupt.

Note: **The RINT command must be preceded from a UDGI (or MDGI) command and followed by a UEGI (or MEGI) command**.

WARNING: If an interrupt is reset, with the RINT instruction within its own interrupt routine, the priority level of the interrupt becomes the lowest and the routine can be immediately interrupted by a lower priority interrupt request.

5 CLOCK, RESET & POWER SAVING MODE

5.1 System Clock

The ST52x430 Clock Generator module generates the internal clock for the internal Control Unit, ALU and on-chip peripherals and it is designed to require a minimum number of external components.

The ST52x430 oscillator circuit generates an internal clock signal with the same period and phase as that of the OSCin input pin. The maximum frequency allowed is **20 Mhz**.

Note: When the SCI peripheral is used only a 5, 10, or 20 MHz system clock must be used.

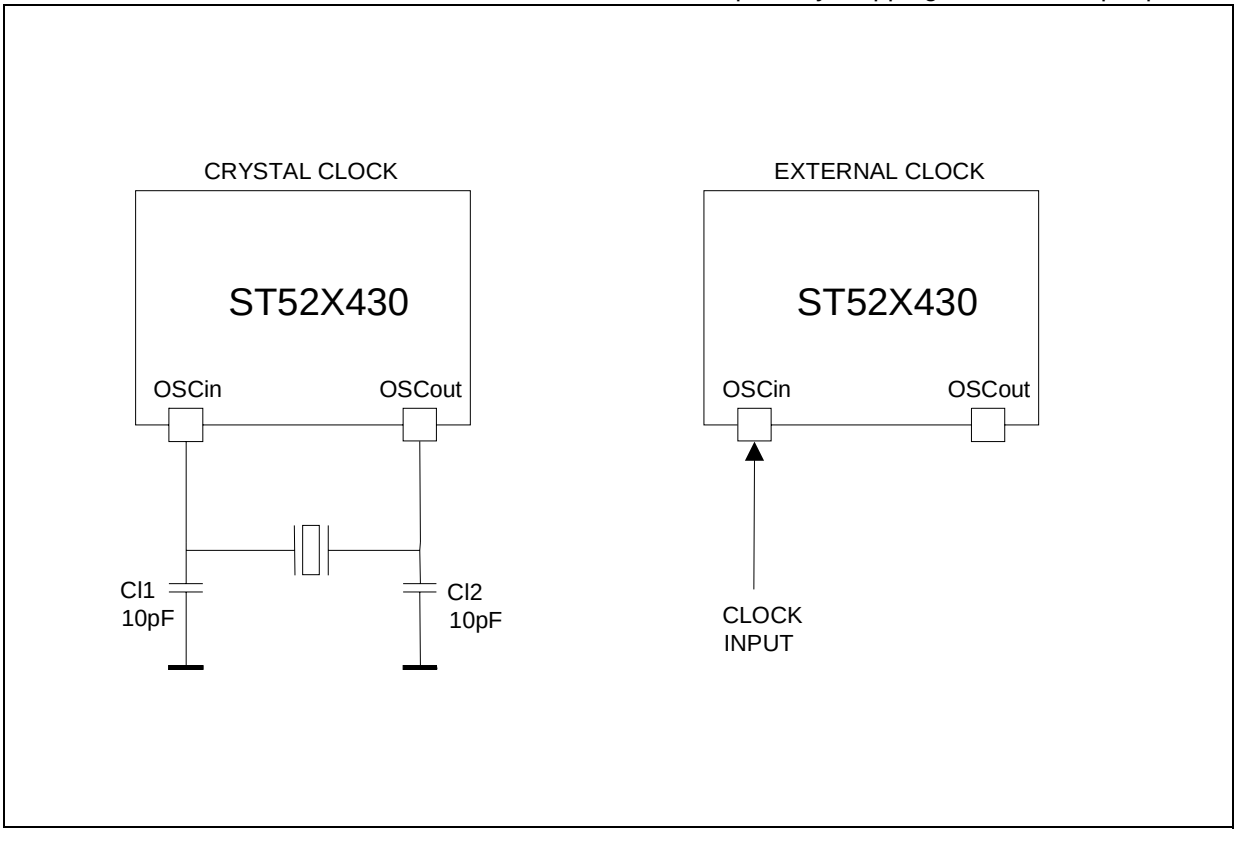
The system clock may be generated by using either a quartz crystal, ceramic resonator or an external clock.

The different methods of the clock generator are illustrated in Figure 5.1.

When an external clock is used, it must be connected on the OSCin pin, while OSCout can be floating.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (CL), IC parameters, environment temperature, supply voltage.

Figure 5.1 Oscillator Connections



Note: The crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1, CL2 are 10pF for a 20 MHz crystal.

5.2 RESET

There are two Reset sources:

- RESET pin (external source.)
- WATCHDOG (internal source)

When a Reset event happens, the user program restarts from the beginning.

The Reset pin is an input. An internal reset does not affect this pin.

A Reset signal originated by external sources is recognized instantaneously. The RESET pin may be used to ensure V_{DD} has risen to a point where the MCU can operate correctly before the user program runs. In working mode Reset must be set to '1' (see Table 2.1).

5.3 Power Saving Mode

There are two Power Saving modes: WAIT and HALT mode. These conditions may be entered using the WAIT or HALT instructions.

5.3.1 Wait Mode

Wait mode places the MCU in low power consumption by stopping the CPU. All peripherals

and the watchdog remain active. During WAIT mode, Interrupts are enabled. The MCU will remain in Wait mode until an Interrupt or a RESET occurs, whereupon the Program Counter jumps to the interrupt service routine or, if a RESET occurs, at the beginning of the user program.

REMARK: In Wait mode the CPU clock does not stop.

5.3.2 Halt Mode

Halt mode is MCU's lowest power consumption mode, which is entered by executing the HALT instruction. The internal oscillator is turned off, causing all internal processing to stop, including the operations of the on-chip peripherals.

Figure 5.2 Reset Block Diagram

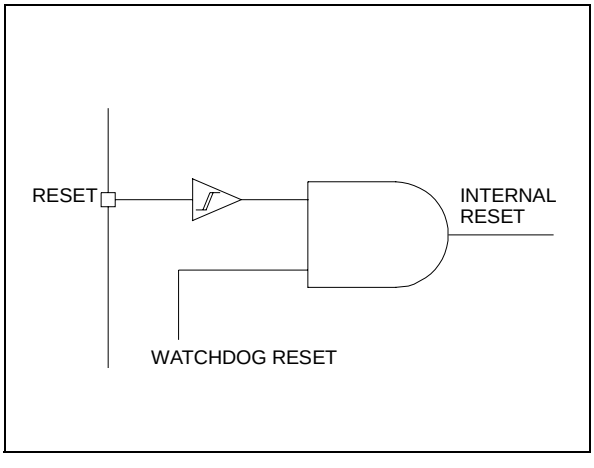
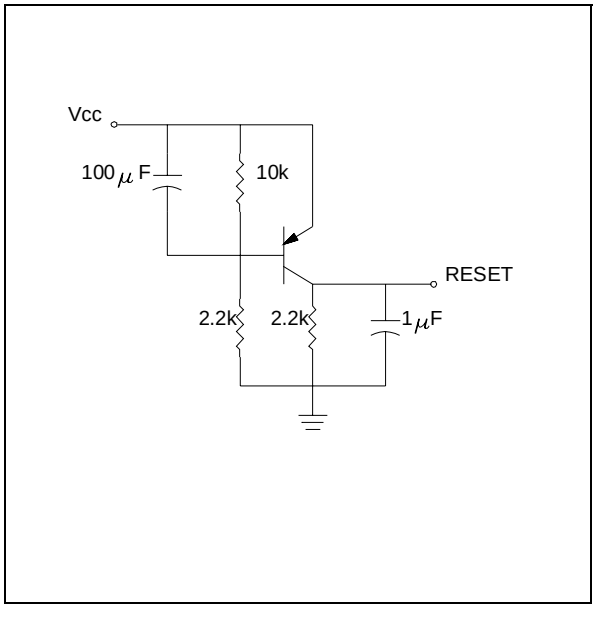


Figure 5.3 Simple Reset Circuit



Halt mode cannot be used when the watchdog is enabled.

If the HALT instruction is executed while the watchdog system is enabled, it will be skipped without modifying the normal CPU operations.

The ICU can exit Halt mode after an external interrupt or reset. The oscillator is then turned on and stabilization time is provided before restarting CPU operations. Stabilization time is 4096 CPU clock cycles after the interrupt and 1.000.000 after the Reset.

After the start up delay, the CPU restarts operations by serving the external interrupt routine. Reset makes the ICU exit from HALT mode and restart, after the delay, from the beginning of the user program after the delay.

Warning: if the External Interrupt is disabled, the ICU exits from the Halt mode and jumps **to the lower priority interrupt routine.**

Figure 5.4 WAIT Flow Chart

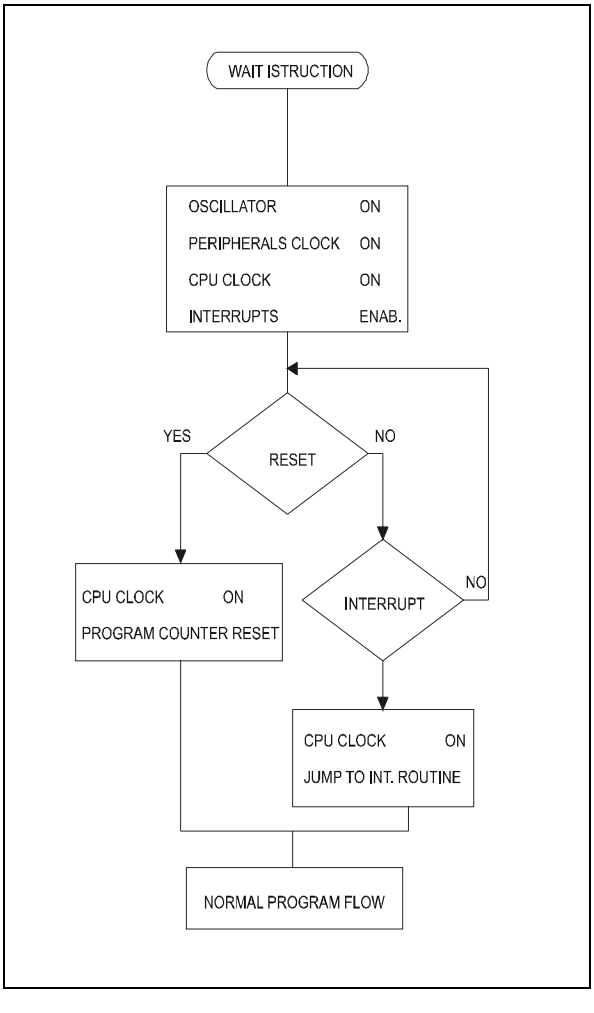
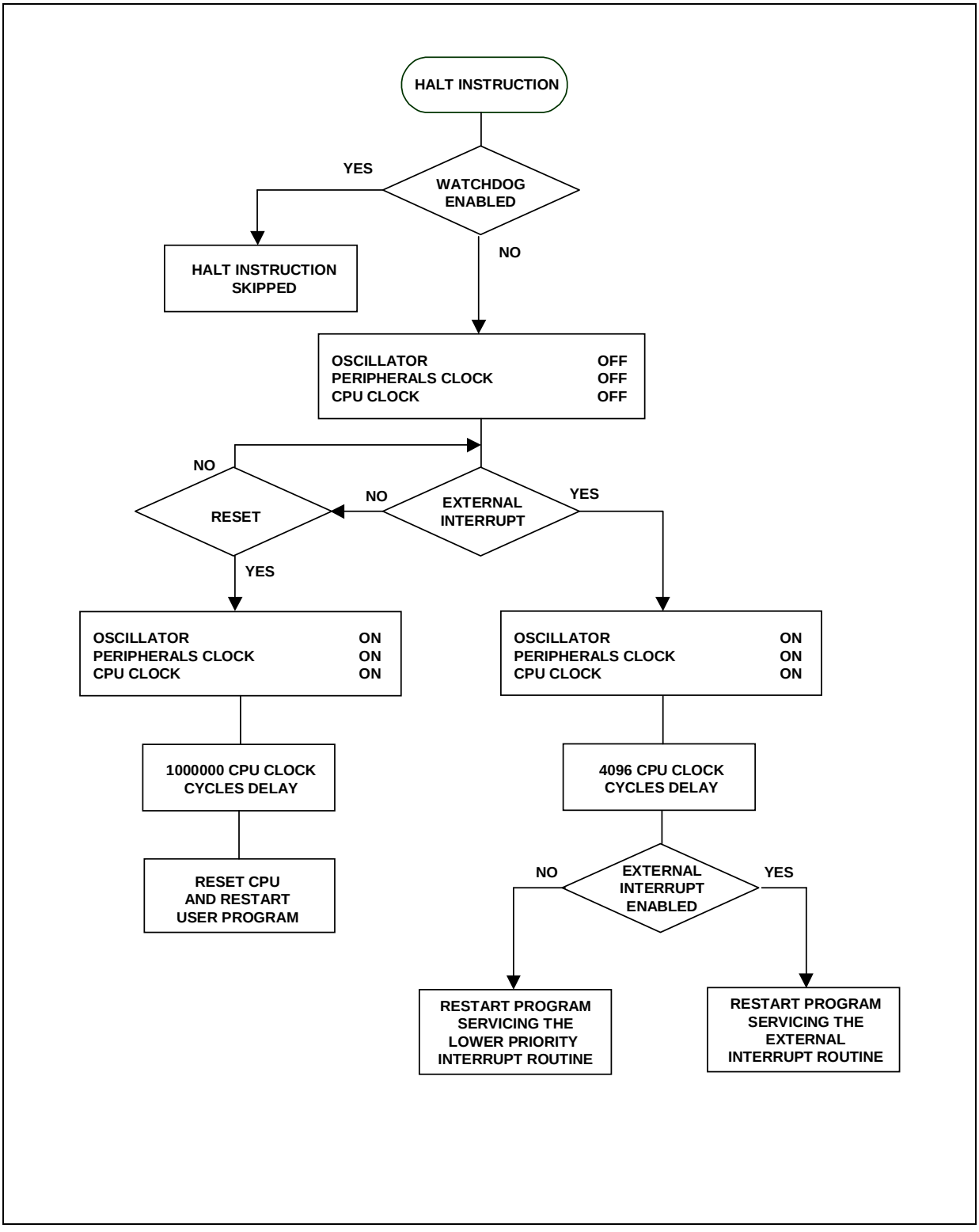


Figure 5.5 HALT Flow Chart



6 I/O PORTS

6.1 Introduction

ST52x430 devices feature flexible individually programmable multi-functional input/output lines. Refer to the following figure for specific pin allocations.

23 I/O lines, grouped in 3 different ports are available on the ST52x430:

PORT A = 7 or 8-bit ports (PA0 - PA7 pins)

PORT B = 7 or 8-bit ports (PB0 - PB7 pins)

PORT C = 8-bit port (PC0 - PC7 pins)

PIN 24 in the SO34 or PIN 22 in the PDIP32 can be configured to belong to port A or to port B.

These I/O lines can be programmed to provide digital input/output and analog input, or to connect input/output signals to the on-chip peripherals as alternate pin functions.

Input buffers are TTL compatible with Schmitt trigger in port A and C while port B is CMOS compatible without Schmitt trigger.

The output buffer can supply up to 8 mA.

The port cannot be configured to be used contemporaneously as input and output.

Each port is configured by using two configuration registers. The first is used to determine if a pin is an input or output, while the second defines the Alternate functions.

6.2 Input Mode

The input configuration is selected by setting the corresponding configuration register bit to “1” (REG_CONF 4, 13 and 15) (see paragraph I/O Port Configuration Registers). The ports are configured by using the configuration registers illustrated in the following table.

.

Table 6.1 I/O Port Configuration Registers.

PORT A	PORT B	PORT C
Reg_Conf 4	Reg_Conf 13	Reg_Conf 15

Digital input data is automatically stored in the Input Registers, but it cannot be read directly. In order to read a single bit of the IR its value must be copied in a RAM location. Digital data is stored in a RAM location by using the assembler instruction:

LDRI RAM_Reg Input_i

Figure 6.1 Ports A & C Functional Blocks

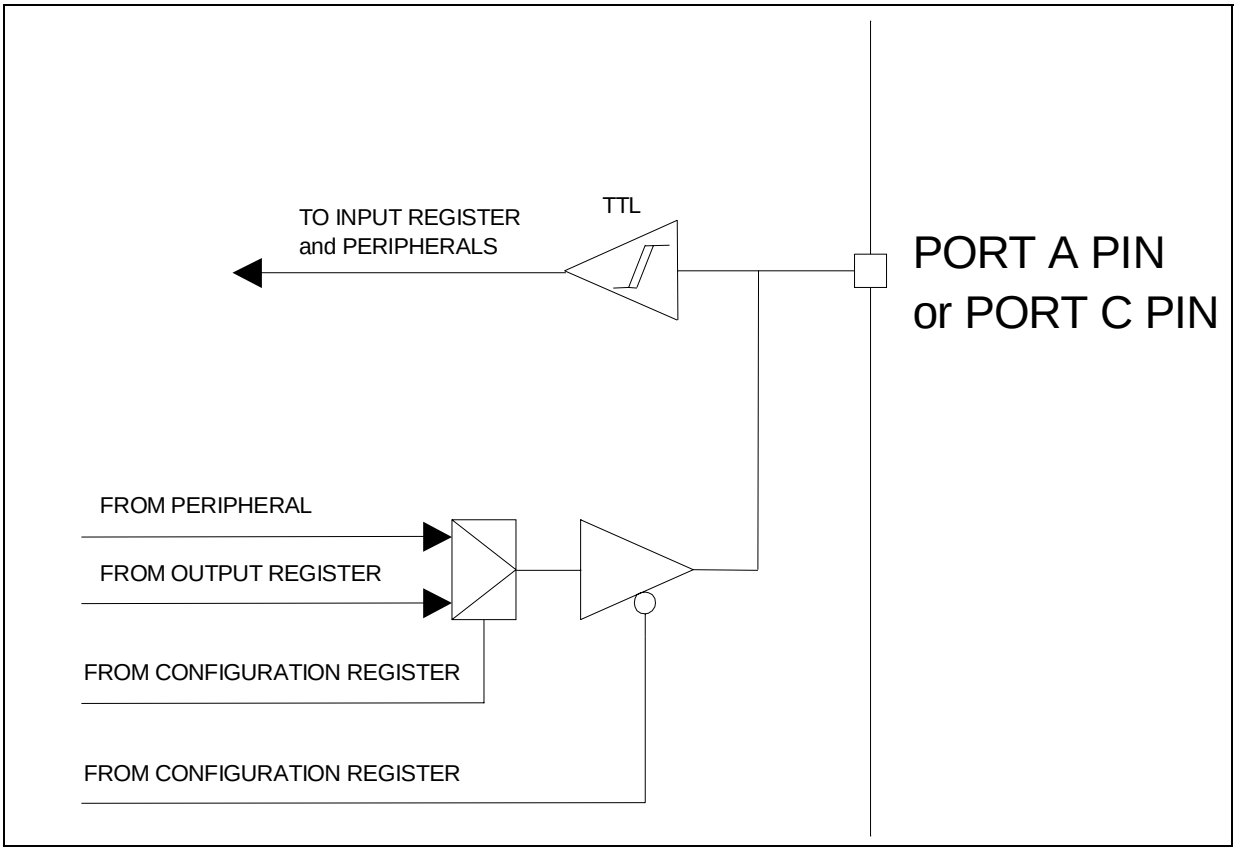


Figure 6.2 Port B Functional Blocks

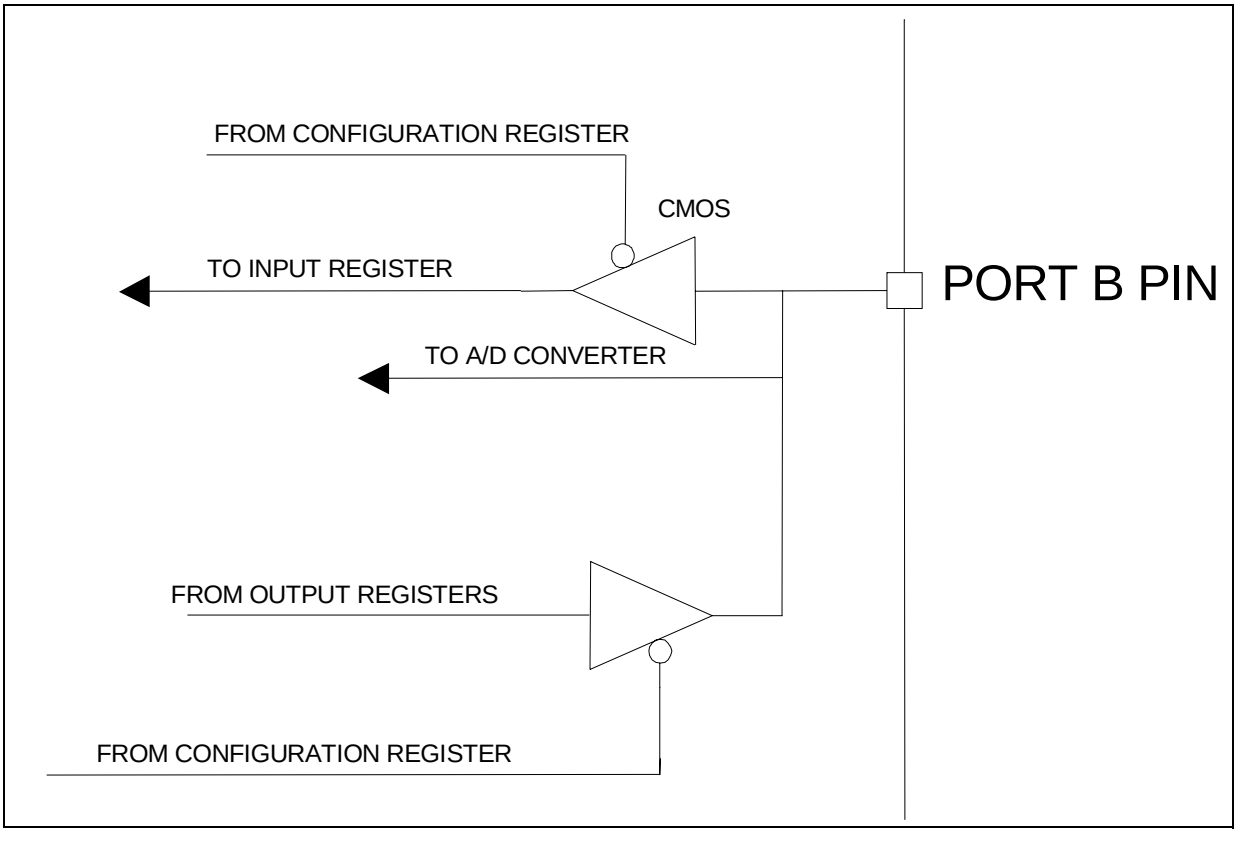


Table 6.2 Input Register and I/O Ports

PORT A	PORT B	PORT C
IR 9	IR 10	IR 11

6.3 Output Mode

The output configuration is selected by setting the corresponding configuration register bit to “0” (REG_CONF 4, 13 and 15) (see paragraph I/O Port Configuration Registers).

Digital data is transferred to the related I/O Port by means of the Output register via the assembler instructions LDPE or LDPR.

Table 6.3 Output Register and I/O Ports

PORT A	PORT B	PORT C
OR 0	OR 1	OR 2

6.4 Alternate Functions

Several ST52x430 pins are configurable to be used with different functions (see Table 1.1).

When an on-chip peripheral is configured to use a pin, the correct I/O mode of the related pin must be selected.

For example: if pin 26 (PA5/T0CLK in the SO34) has to be used as an external PWM/Timer0 clock, the Reg_Conf 4(5) bit must be set to ‘1’.

When the signal is an on-chip peripheral input the related I/O pin has to be configured in Input Mode.

When a pin is used as an A/D Converter input the related I/O pin is automatically set in tristate. The analog multiplexer (controlled by the A/D configuration Register) switches the analog voltage present on the selected pin to the common analog rail, which is connected to the ADC input.

It is recommended that the voltage level not be changed or that any port pins not be loaded while conversion is running. Furthermore, it is recommended that clocking pins not be located close to a selected analog pin.

6.5 I/O Port Configuration Registers

The I/O mode for each bit of the three ports is selected by using the Configuration Registers 4,

13 and 15 (See Table 6.1) The structure of these registers is illustrated in the following tables.

Each bit of the configuration registers determines the I/O mode of the related port pin.

Table 6.4 Ports A REG_CONF 4

Bit	Name	Value	Description
0	D0	0	Set the pin PA0/T0RES in Output Mode
		1	Set the pin PA0/T0RES in Input Mode
1	D1	0	Set the pin PA1/T0OUT in Output Mode
		1	Set the pin PA1/T0OUT in Input Mode
2	D2	0	Set the pin PA2/T1OUT in Output Mode
		1	Set the pin PA2/T1OUT in Input Mode
3	D3	0	Set the pin PA3/T2OUT in Output Mode
		1	Set the pin PA3/T2OUT in Input Mode
4	D4	0	Set the pin PA4/T0STRT in Output Mode
		1	Set the pin PA4/T0STRT in Input Mode
5	D5	0	Set the pin PA5/T0CLK in Output Mode
		1	Set the pin PA5/T0CLK in Input Mode
6	D6	0	Set the pin PA6 in Output Mode
		1	Set the pin PA6 in Input Mode
7	D7	0	Set the pin PB7/PA7/ Ain7 in Output Mode
		1	Set the pin PB7/PA7/ Ain7 in Input Mode
Reset Configuration ‘11111111’			

Table 6.5 Ports B REG_CONF 13

Bit	Name	Value	Description
0	D0	0	Set the pin PB0/Ain0 in Output Mode
		1	Set the pin PB0/Ain0 in Input Mode
1	D1	0	Set the pin PB1/Ain1 in Output Mode
		1	Set the pin PB1/Ain1 in Input Mode
2	D2	0	Set the pin PB2/Ain2 in Output Mode
		1	Set the pin PB2/Ain2 in Input Mode
3	D3	0	Set the pin PB3/Ain3 in Output Mode
		1	Set the pin PB3/Ain3 in Input Mode
4	D4	0	Set the pin PB4/Ain4 in Output Mode
		1	Set the pin PB4/Ain4 in Input Mode
5	D5	0	Set the pin PB5/Ain5 in Output Mode
		1	Set the pin PB5/Ain5 in Input Mode
6	D6	0	Set the pin PB6/Ain6 in Output Mode
		1	Set the pin PB6/Ain6 in Input Mode
7	D7	0	Set the pin PB7/PA7/ Ain7 in Output Mode
		1	Set the pin PB7/PA7/ Ain7 in Input Mode
Reset Configuration ‘11111111’			

Table 6.6 Port C REG_CONF 15

Bit	Name	Value	Description
0	D0	0	Set the pin INT/PC0 in Output Mode
		1	Set the pin INT/PC0 in Input Mode
1	D1	0	Set the pin T0OUT/ PC1 in Output Mode
		1	Set the pin T0OUT/ PC1 in Input Mode
2	D2	0	Set the pin T1OUT/ PC2 in Output Mode
		1	Set the pin T1OUT/ PC2 in Input Mode
3	D3	0	Set the pin T2OUT/ PC3 in Output Mode
		1	Set the pin T2OUT/ PC3 in Input Mode
4	D4	0	Set the pin Tx/PC4 in Output Mode
		1	Set the pin Tx/PC4 in Input Mode
5	D5	0	Set the pin Rx/PC5 in Output Mode
		1	Set the pin Rx/PC5 in Input Mode
6	D6	0	Set the pin PC6 in Output Mode
		1	Set the pin PC6 in Input Mode
7	D7	0	Set the pin PC7 in Output Mode
		1	Set the pin PC7 in Input Mode
Reset Configuration '11111111'			

Analog Input Option. The PB0-PB7 pins can be configured to be analog inputs according to the codes programmed in the configuration register REG_CONF 14 (See Table 6.7). These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter.

Table 6.7 Analog Inputs (REG_CONF 14)

Bit	Name	Value	Description
0	D0	0	pin PB0/Ain0 Digital I/O
		1	pin PB0/Ain0 Analog
1	D1	0	pin PB1/Ain1 Digital I/O
		1	pin PB1/Ain1 Analog
2	D2	0	pin PB2/Ain2 Digital I/O
		1	pin PB2/Ain2 Analog
3	D3	0	pin PB3/Ain3 Digital I/O
		1	pin PB3/Ain3 Analog
4	D4	0	pin PB4/Ain4 Digital I/O
		1	pin PB4/Ain4 Analog
5	D5	0	pin PB5/Ain5 Digital I/O
		1	pin PB5/Ain5 Analog
6	D6	0	pin PB6/Ain6 Digital I/O
		1	pin PB6/Ain6 Analog
7	D7	0	pin PB7/Ain7 Digital I/O
		1	pin PB7/Ain7 Analog
Reset Configuration '11111111'			

PWM/Timers Alternate Functions

The pins of Port A and C can be configured to be I/O of the three PWM/Timers available on the ST52x430. The configuration of these pins is performed by using the Configuration Registers REG_CONF 12 and REG_CONF 16 if the related pin has to be output. When the related pin has to be used as an input peripheral the configuration is performed by the relative peripheral configuration registers (See PWM/Timer Session).

Table 6.8 PWM/Timers REG_CONF 16

Bit	Name	Value	Description
0	PC1	1	Pin T0OUT/PC1 is configured as Port C Digital I/O
		0	Pin T0OUT/PC1 is configured as PWM/ Timer 0 output T0OUT
1	PC2	1	PinT1OUT/PC2 is configured as Port C Digital I/O
		0	Pin T1OUT/PC2 is configured as PWM/ Timer 1 output T1OUT
2	PC3	1	Pin T2OUT/PC3 is configured as Port C Digital I/O
		0	Pin T2OUT/PC3 is configured as PWM/ Timer 2 output T2OUT
3	PC4	1	Pin Tx/PC4 is configured as Port C Digital I/O
		0	Pin Tx/PC4 is configured as SCI output Tx
4-7	NC	X	Not Used
Reset Configuration '1111'			

Table 6.9 PWM/Timers REG_CONF 12

Bit	Name	Value	Description
0	PA1	1	Pin PA1/$\overline{\text{T0OUT}}$ is configured as PWM/Timer 0 complementary output
		0	Pin PA1/$\overline{\text{T0OUT}}$ is configured as Port A Digital I/O
1	PA2	1	Pin PA2/$\overline{\text{T1OUT}}$ is configured as PWM/Timer 1 complementary output
		0	Pin PA2/$\overline{\text{T1OUT}}$ is configured as Port A Digital I/O
2	PA3	1	Pin PA3/$\overline{\text{T2OUT}}$ is configured as PWM/Timer 2 complementary output
		0	Pin PA3/$\overline{\text{T2OUT}}$ is configured as Port A Digital I/O
3	PASZ	1	PORT A bits = 8
		0	PORT A bits = 7
4-7	NC	x	Not Used
Reset Configuration '0000'			

Figure 6.3 Configuration Register 12

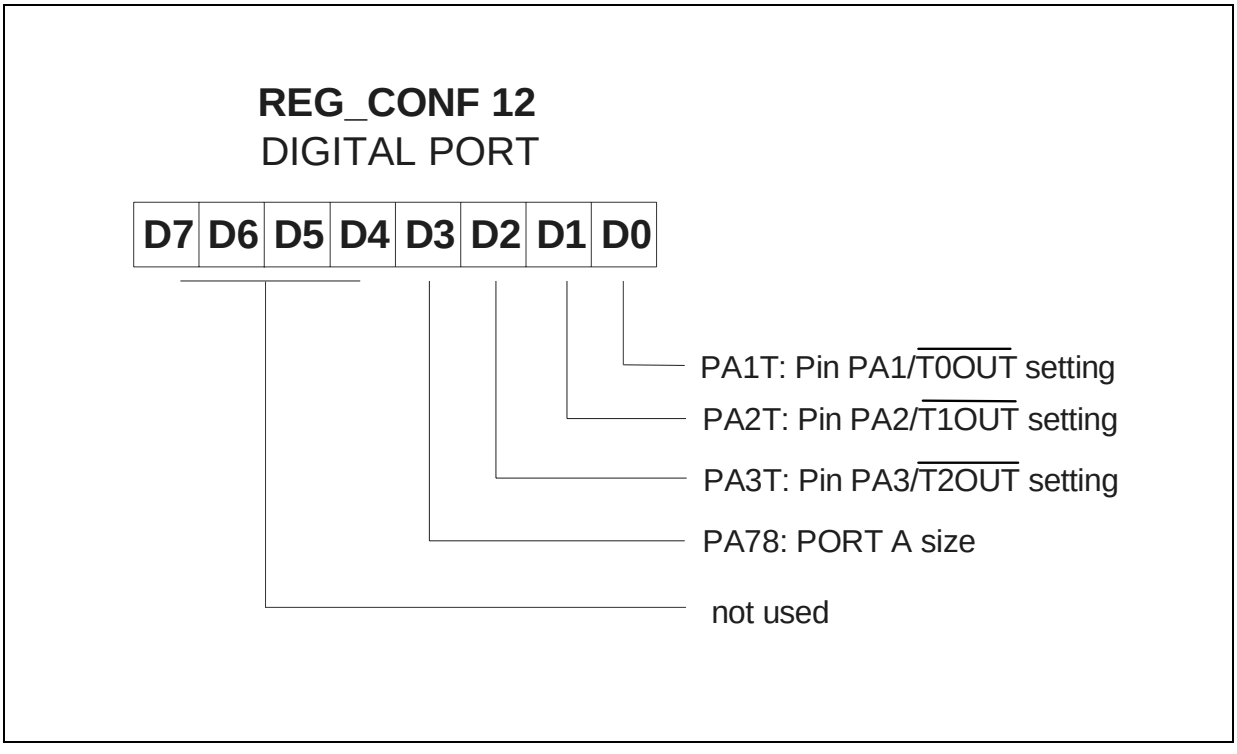
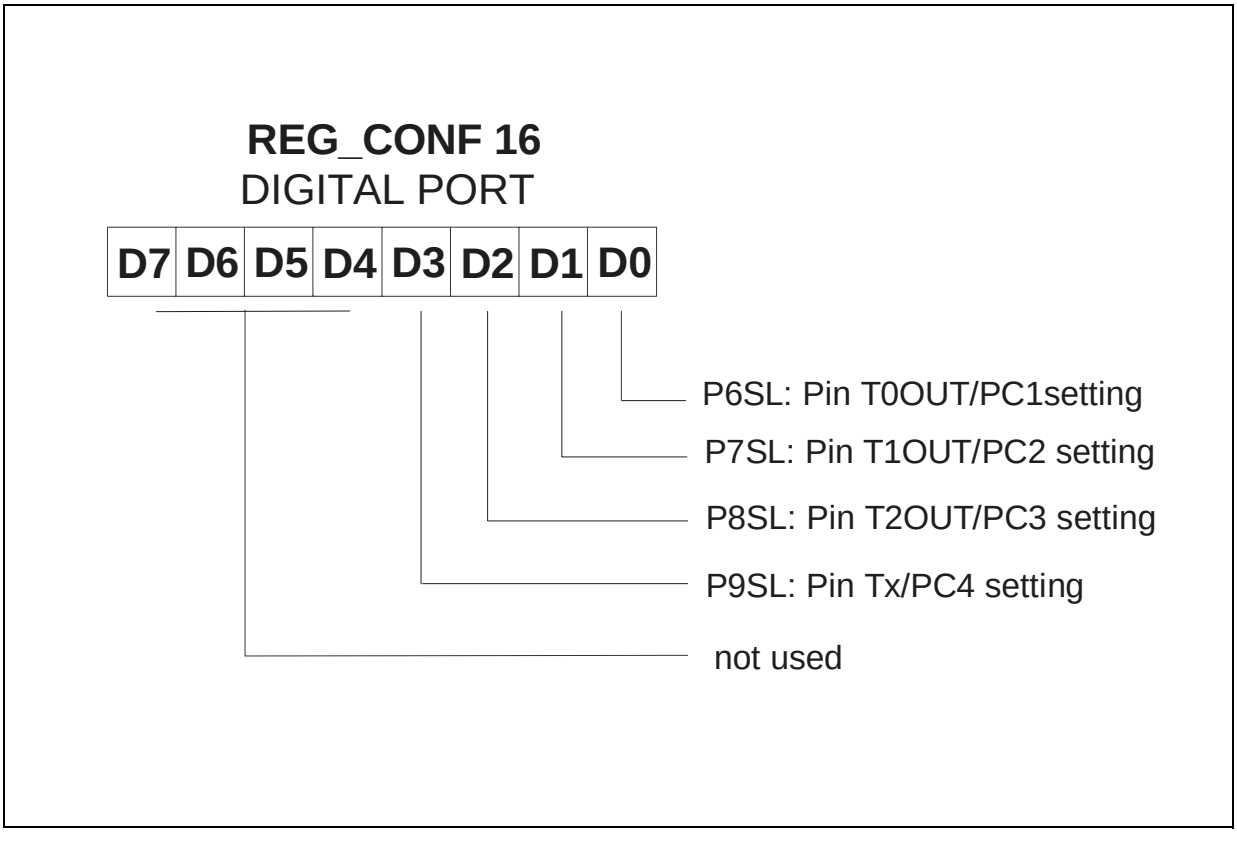


Figure 6.4 Configuration Register 16



7 FUZZY COMPUTATION (DP)

The ST52T430/E430 Decision Processor (DP) main features are:

- Up to 8 Inputs with 8-bit resolution;
- 1 Kbyte of Program/Data Memory available to store more than 300 to Membership Functions (Mbfs) for each Input;
- Up to 128 Outputs with 8-bit resolution;
- Possibility of processing fuzzy rules with an UNLIMITED number of antecedents;
- UNLIMITED number of Rules and Fuzzy Blocks.

The limits on the number of Fuzzy Rules and Fuzzy program blocks are only related to the Program/Data Memory size.

7.1 Fuzzy Inference

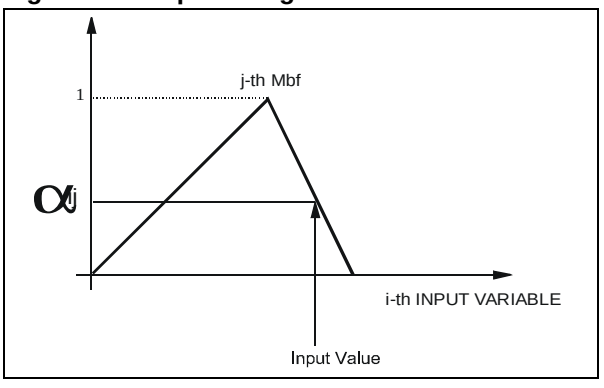
The block diagram shown in Figure 7.1 describes the different steps performed during a Fuzzy algorithm. The ST52T430/E430 Core allows for the implementation of a Mamdani type fuzzy inference with crisp consequents. Inputs for fuzzy inference are stored in 8 dedicated Fuzzy input registers. The LDFR instruction is used to set the Input Fuzzy registers with values stored in the Register File. The result of a Fuzzy inference is stored directly in a location of the Register File.

7.2 Fuzzyfication Phase

In this phase the intersection (alpha weight) between the input values and the related Mbfs (Figure 7.2) is performed.

Eight Fuzzy Input registers are available for Fuzzy inferences.

Figure 7.2 Alpha Weight Calculation



After loading the input values by using the LDFR assembler instruction, the user can start the fuzzy inference by using the FUZZY assembler instruction. During fuzzyfication: input data is transformed in the activation level (alpha weight) of the Mbf's.

7.3 Inference Phase

The Inference Phase manages the alpha weights obtained during the fuzzyfication phase to compute the truth value (ω) for each rule.

This is a calculation of the maximum (for the OR operator) and/or minimum (for the AND operator) performed on alpha values according to the logical connectives of Fuzzy Rules.

Several conditions may be linked together by linguistic connectives AND/OR, NOT operators and brackets.

The truth value ω and the related output singleton are used by the Defuzzyfication phase, in order to complete the inference calculation.

Figure 7.1 Fuzzy Inference

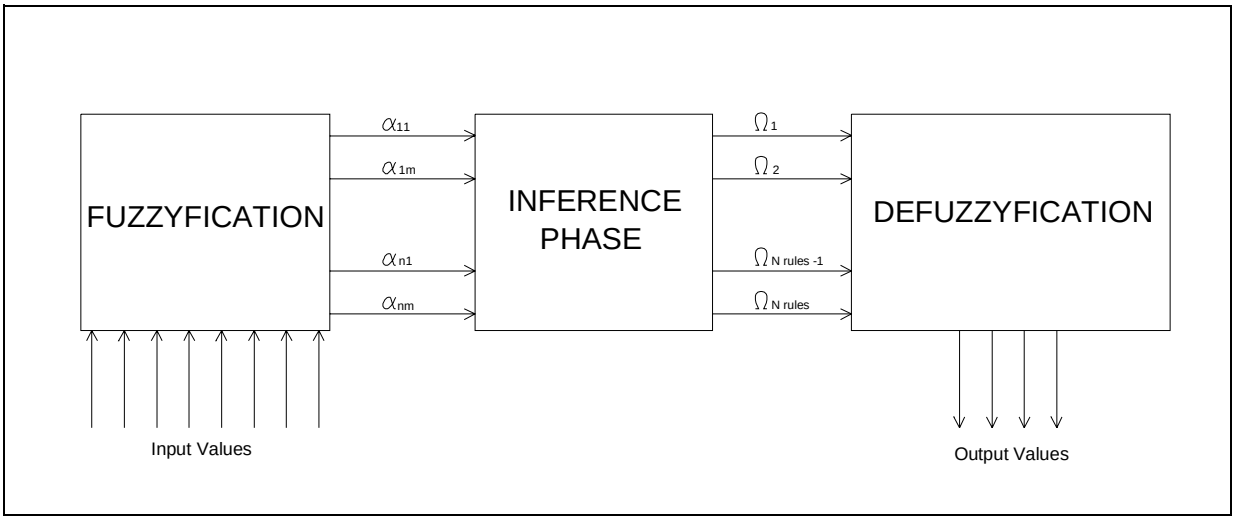
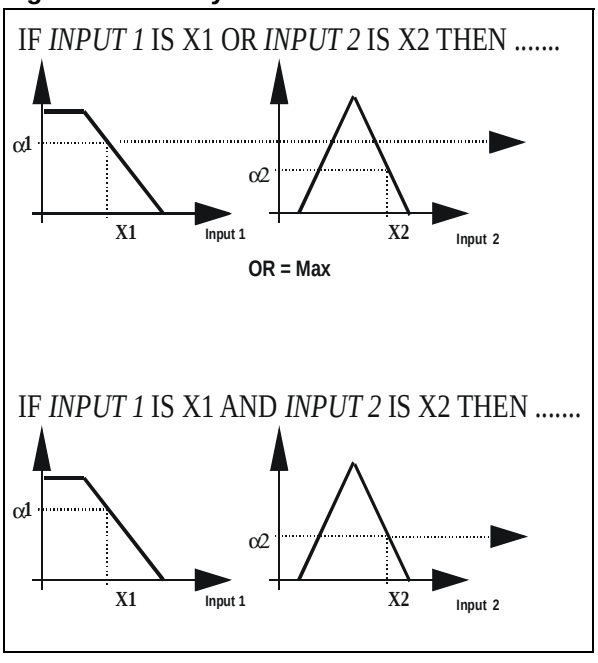


Figure 7.3 Fuzzyfication



7.4 Defuzzyfication

In this phase the output crisp values are determined by implementing the consequent part of the rules.

Each consequent Singleton X_i is multiplied by its weight values ω_i, calculated by the Decision processor, in order to compute the upper part of the Defuzzyfication formula.

Each output value is obtained from the consequent crisp values (X_i) by carrying out the following Defuzzyfication formula:

$$Y_i = \frac{\sum_{j}^{N} X_{ij}\omega_{ij}}{\sum_{j} \omega_{ij}}$$

where:

i = identifies the current output variable

N = number of the active rules on the current output

ω_{ij} = weight of the j-th singleton

X_{ij} = abscissa of the j-th singleton

The Decision Processor outputs are stored in the RAM location i-th specified in the assembler instruction OUT i.

7.5 Input Membership Function

The Decision Processor allows the management of triangular Mbfs. In order to define an Mbf, three different parameters must be stored on the Program/Data Memory (see Figure 7.4):

- the vertex of the Mbf: **V**;
- the length of the left semi-base: **LVD**;
- the length of the right semi-base: **RVD**;

In order to reduce the size of the memory area and the computational effort the vertical range of the vertex is fixed between 0 and 15 (4 bits)

By using the previous memorization method different kinds of triangular Membership Functions may be stored. Figure 7.5 shows some examples of valid Mbfs that can be defined in ST52T430/E430.

Each Mbf is then defined storing 3 bytes in the first Kbyte of the Program/Data Memory.

The Mbf is stored by using the following instruction:

MBF *n_mbf lvd v rvd*

where:

n_mbf is a tag number that identifies the Mbf

lvd, *v*, and *rvd* are the parameters that describe the Mbf's shape as described above.

Figure 7.4 Mbfs Parameters

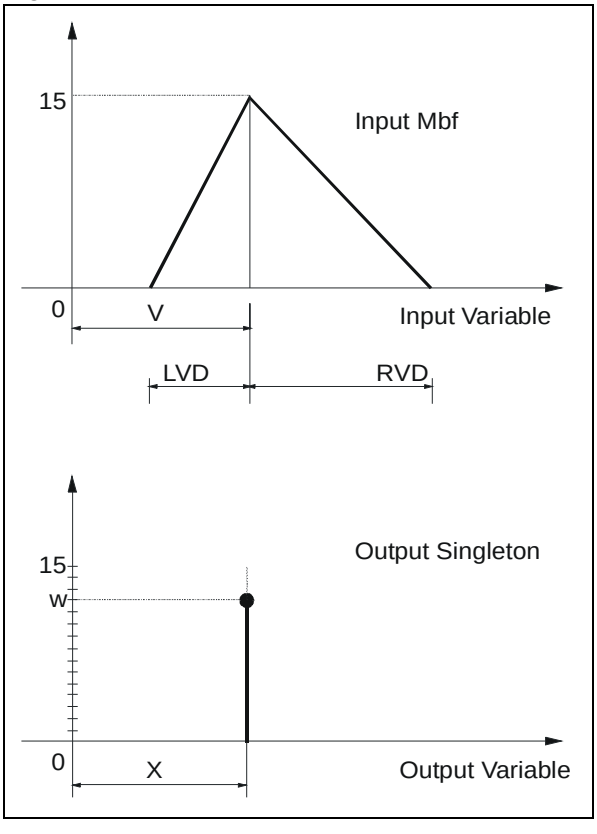


Figure 7.5 Example of valid Mbfs

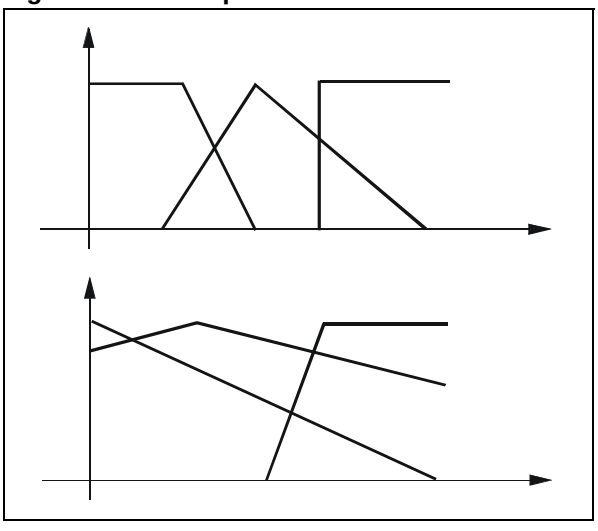
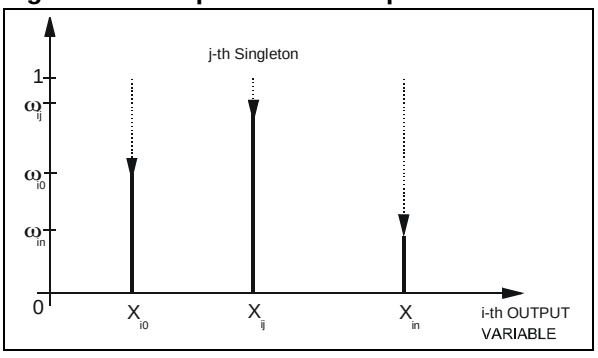


Figure 7.6 Output Membership Functions



7.6 Output Singleton

The Decision Processor uses a particular kind of membership function called Singleton for its output variables. A Singleton doesn't have a shape, like a traditional Mbf, and is characterized by a single point identified by the couple (X, w), where w is calculated by the Inference Unit as described earlier. Often, a Singleton is simply identified with its Crisp Value X.

7.7 Fuzzy Rules

Rules can have the following structures:

if A op B op C.....then Z

if (A op B) op (C op D op E...) .....then Z

where *op* is one of the possible linguistic operators (AND/OR)

In the first case the rule operators are managed sequentially; in the second one, the priority of the operator is fixed by the brackets.

Each rule is codified by using an instruction set, the inference time for a rule with 4 antecedents and 1 consequent is about 3 microseconds at 20 MHz.

The Assembler Instruction Set used to manage the Fuzzy operations is reported in the table below.

Table 7.1 Fuzzy Instructions Set

Instruction	Description
MBF *n_mbf Ivd v rvd*	Stores the Mbf *n_mbf* with the shape identified by the parameters *Ivd*, *v* and *rvd*
LDP *n m*	Fixes the alpha value of the input *n* with the Mbf *m* and stores it in internal registers
LDN *n m*	Calculates the complementary alpha value of the input *n* with the Mbf *m.* and stores the result in internal registers
FZAND	Implements the Fuzzy operation AND between the last two values stored in internal registers
FZOR	Implements the Fuzzy operation OR between the last two values stored in internal registers
LDK	Stores the result of the last Fuzzy operation executed in internal registers
SKM	Loads the result of the last performed Fuzzy operation (stored in the temporary register K) in the temporary buffer M.
LDM	Copies the value of register M in the data stack
CON *crisp*	Multiplies the *crisp* value with the last ω weight
OUT *n_out*	Performs Defuzzyfication and stores the currently Fuzzy output in the RAM *n_out* location
FUZZY	Starts the Fuzzy algorithm

Example 1:

IF Input$_1$ IS NOT Mbf$_1$ AND Input$_4$ is Mbf$_{12}$ OR Input$_3$ IS Mbf$_8$ THEN Crisp$_1$

is codified by the following instructions:

LDN 1 1 calculates the NOT α value of Input$_1$ with Mbf$_1$ and stores the result in internal registers

LDP 4 12 fixes the α value of Input$_4$ with Mbf$_{12}$ and stores the result in internal registers

FZAND implements the operation AND between the results obtained with the previous instructions

LDK stores the result of the previous operation in internal DPU registers

LDP 3 8 fixes the α value of Input$_3$ with Mbf$_8$ and stores the result in internal registers

FZOR implements the operation OR between the results obtained with the previous instructions

CON *crisp$_1$* multiplies the result of the last Ω operation with the crisp value *crisp$_1$*

Example 2, the priority of the operator is fixed by the brackets:

IF (Input$_3$ IS Mbf$_1$ AND Input$_4$ IS NOT Mbf$_{15}$) OR (Input$_1$ IS Mbf$_6$ OR Input$_6$ IS NOT Mbf$_{14}$) THEN Crisp$_2$

LDP 3 1 fixes the α value of Input$_3$ with Mbf$_1$ and stores the result in internal registers

LDN 4 15 calculates the NOT α value of Input$_4$ with Mbf$_{15}$ and stores the result in internal registers

FZAND implements the operation AND between the results obtained with the previous instructions

SKM stores the result of the previous operation in register M

LDP 1 6 fixes the α value of Input$_1$ with Mbf$_6$ and stores the result in internal registers

LDN 2 14 calculates the NOT α value of Input$_6$ with Mbf$_{14}$ and stores the result in internal registers

FZOR implements the operation OR between the results obtained with the previous instructions

LDK stores the result of the previous operation in internal DPU registers

LDM copies the value of the register M in internal DPU registers

FZOR implements the operation OR between the last two values stored in DPU registers

CON *crisp$_2$* multiplies the result of the last Ω operation with the crisp value *crisp$_2$*

At the end of the fuzzy rule, by using the instruction **OUT *RAM_reg***, a byte is written. Afterwards, the control of the algorithm returns to the CU.

8 A/D CONVERTER

8.1 Introduction

The A/D Converter of ST52x430 is an 8-bit analog to digital converter with up to 8 analog inputs offering 8 bit resolution with a total accuracy of 1 LSB and a typical conversion time of 8.2 µs with a 20 MHz clock. This period also includes the 5.1 µs of the integral Sample and Hold circuitry, which minimizes the need for external components and allows quick sampling of the signal for a minimum warping effect and Integral conversion error.

Conversion is performed in 82 A/D clock pulses.

The A/D clock is derived from the clock master. The maximum A/D clock frequency has to be 10 MHz. When the master clock is higher than 10 MHz it has to be divided by 2 using the SCK bit of the A/D configuration register REG_CONF 3 (See Table 8.1).

The A/D peripheral converts the input voltage with a process of successive approximations using a fixed clock frequency derived from the oscillator.

The conversion range is between the analog V_{SS} and V_{DD} references.

The converter uses a fully differential analog input configuration for the best noise immunity and precision performance, along with one separate supply (V_{DDA}), allowing the best supply noise rejection.

Up to 8 multiplexed Analog Inputs are available. A group of signals can be converted sequentially by simply programming the starting address of the last analog channel to be converted.

Single or continuous conversion mode are available.

The result of the conversion is stored in an 8-bit Input Register (from IR 1 to IR 8).

The A/D converter is controlled via the Configuration Register REG_CONF 3.

A Power-Down programmable bit allows the A/D converter to be set to a minimum consumption idle status.

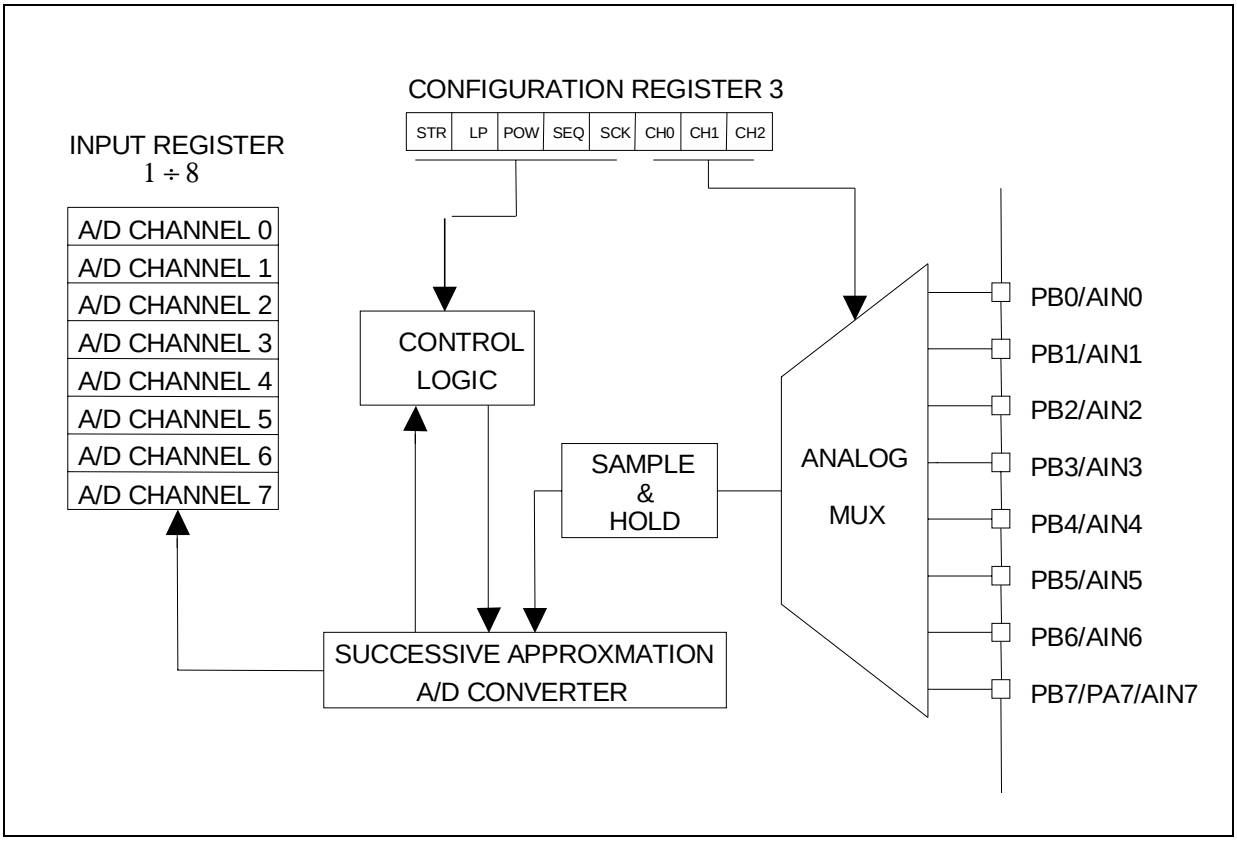
The ST52x430 Interrupt Unit provides one maskable channel for the End of Conversion (EOC).

8.2 Operational Description

The conversion is monotonic, meaning that the result never decreases if the analog input doesn't and never increases if the analog input doesn't.

If input voltage is greater than or equal to V_{dda} (Voltage Reference high) then the result is equal to FFh (full scale) without an overflow indication.

Figure 8.1 A/D Converter Structure



If input voltage is less than V_{SS} (voltage reference low) then the result is equal to 00h.

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

$$Digitalresult = \frac{255 inputVoltage}{referenceVoltage}$$

Where Reference Voltage is V_{dda} - V_{ss}.

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

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

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

8.2.1 Operating Modes.

Four main operating modes can be selected by setting the values of the LP and SEQ bit in the A/D configuration Register.

One Channel Single Mode

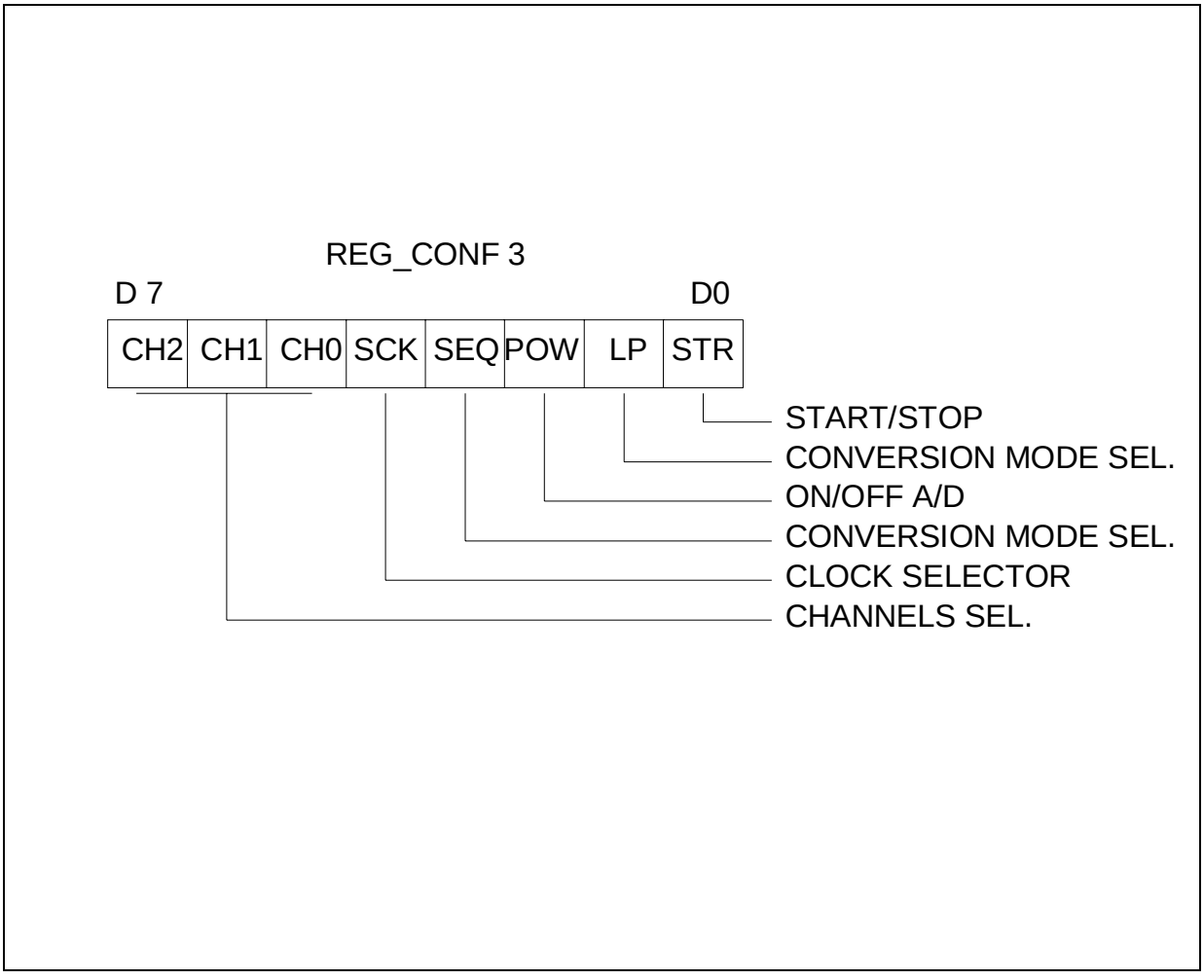
In this mode (**SEQ = '0'', LP = '0'**) the A/D provides an EOC signal after the end of channel i-th conversion; then the A/D waits for a new start event. Channel i-th is identified by the bit CH0, CH1, CH2.

i.e CH(2:0) = '011' means conversion of channel 3 then stop.

Multiple Channels Single Mode

In this mode (**SEQ = '1', LP = '0'**) the A/D provides an EOC signal after the end of the channels sequence conversion identified by the bit CH0, CH1, CH2; then the A/D waits for a new start event. i.e. CH(2:0) = '011' means conversion of channels 0,1,2 and 3 then stop.

Figure 8.2 Conf. Register (REG_CONF 3)



One Channel Continuous Mode

In this mode (**SEQ = '0'', LP = '1'**) a continuous conversion flow is entered by a starting event on the channel selected by the CH0, CH1, CH2 bits

For example: CH(2:0) = '011' means continuous conversion of channel 3. At the end of each conversion the relative IR is updated with the last conversion result, while the former value is lost.

To stop the conversion STR has to be set to '0'**.**

Multiple Channels Continuous Mode

In this mode (**SEQ = '1'', LP = '1'**) a continuous conversion flow is entered by a starting event on the channels selected by the CH0, CH1, CH2 bits.

i.e CH(2:0) = '011' means continuous conversion of channel 0,1,2 and 3.

At the end of each conversion the relative IRs are updated with the last conversion results, while the former values are lost.

To stop the conversion STR has to be set to '0'.

8.2.2 Power Down Mode.

Before enabling any A/D operation mode, set the POW bit of the A/D configuration register to '1' at least 60 µs before the first conversion starts to enable the biasing circuit inside the analog section of the converter. Clearing the POW bit (POW = '0') is useful when the A/D is not used, reducing the total chip power consumption. This state is also the reset configuration and it is forced by hardware when the core is in HALT state (after a HALT instruction execution).

8.3 A/D Registers Description

The result of the conversions of the 8 available channels are loaded in the 8 Input Register from decimal address 1 to decimal address 8. (IR (1:8) see Table 2.2)). Every IR(1:8) is reloaded with a new value at the end of the conversion of the correspondent analog input.

By using the assembler instruction:

LDRI RAM_Reg. IR_i

the value stored in the i-th IR is transferred on the RAM location RAM_Reg.

The A/D configuration register is the REG_CONF 3. Figure 6.2 illustrates the structure of this register, which manages the A/D logic operation. The A/D configuration register (REG_CONF 3) is programmable as following:

b7-b5 = **CH2, CH1, CH0**: Last Conversion Address. These 3 bits define the last analog input. The first analog input is converted, then the address is incremented for the successive conversion until the channel identified by CH0-CH2 is converted. The (CH2, CH1, CH0) bits define the group of channels to be scanned. When setting CH2=0 CH1=0 CH0=0 only channel 0 is converted.

b4 = **SCK**: Master clock divider. ST52x430 can work with a clock frequency up to 20 MHz. The SCK must be set to '1' when the ST52x430 clock is higher then 10 MHz. It is useful to set SCK = '1' even when the clock master is lower than 10 MHz and a high accuracy is required.

b3 = **SEQ**: Multiple/Single channel. When SEQ is set to '0' the channel identified by CH(2:0) is converted. If SEQ is set to '1' the group of channels identified by CH(2:0) are converted.

b2= **POW**: Power Up/ Power Down. A logical '1' enables the A/D logic and analog circuitry.

Logical level '0' disables all power consuming logic, allowing a low power idle status.

b1 = **LP**: Continuous/Single. When this bit is set to '1' (continuous mode), the first conversion sequences are started by the STR bit then a continuous conversion flow is processed.

When LP='0' (single mode) only one sequence of conversions is started when STR is set.

b0 = **STR**: Start/Stop. A logical level '1' enables starting a conversion sequence; a logical level '0' stops the conversion. When the A/D is running in the Single Modes (LP='0'), this bit is hardware reset at the end of a conversion sequence.

Table 8.1 A/D Conf. Register (Reg_Conf 3)

Bit	Name	Value	Description
0	STR	0	Stop Conversion
		1	Start Conversion
1	LP	0	Single Conversion
		1	Continuous
2	POW	0	A/D OFF
		1	A/D ON
3	SEQ	0	Single Channel Conv.
		1	Multiple Channels Conv
4	SCK	0	Clock not Divided
		1	Clock Divided
5	CH(2:0)	000	Channel 0
		001	Channel 1
		010	Channel 2
6		011	Channel 3
		100	Channel 4
7		101	Channel 5
		110	Channel 6
		111	Channel 7

9 WATCHDOG TIMER

9.1 Operational Description

The Watchdog Timer (WDT) is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which cause the application program to abandon its normal sequence. The WDT circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the WDT before the end of the programmed time delay.

16 different delays can be selected by using the WDT configuration register.

After the end of the delay programmed by the configuration register if the WDT is activated (by using the assembler instruction WDTSFR), it starts a reset cycle pulling the reset pin low.

Once the WDT has been activated the application program has to refresh this peripheral (by the WDTSFR instruction) at regular intervals during normal operation in order to prevent an MCU reset.

In order to stop the WDT during user program execution the instruction WDTSLP has to be used.

The working frequency of the WDT (PRES CLK in the Figure 9.1) is equal to the clock master. The clock master is divided by 500, obtaining the WDT CLK signal, which is used to fix the timeout of the WDT.

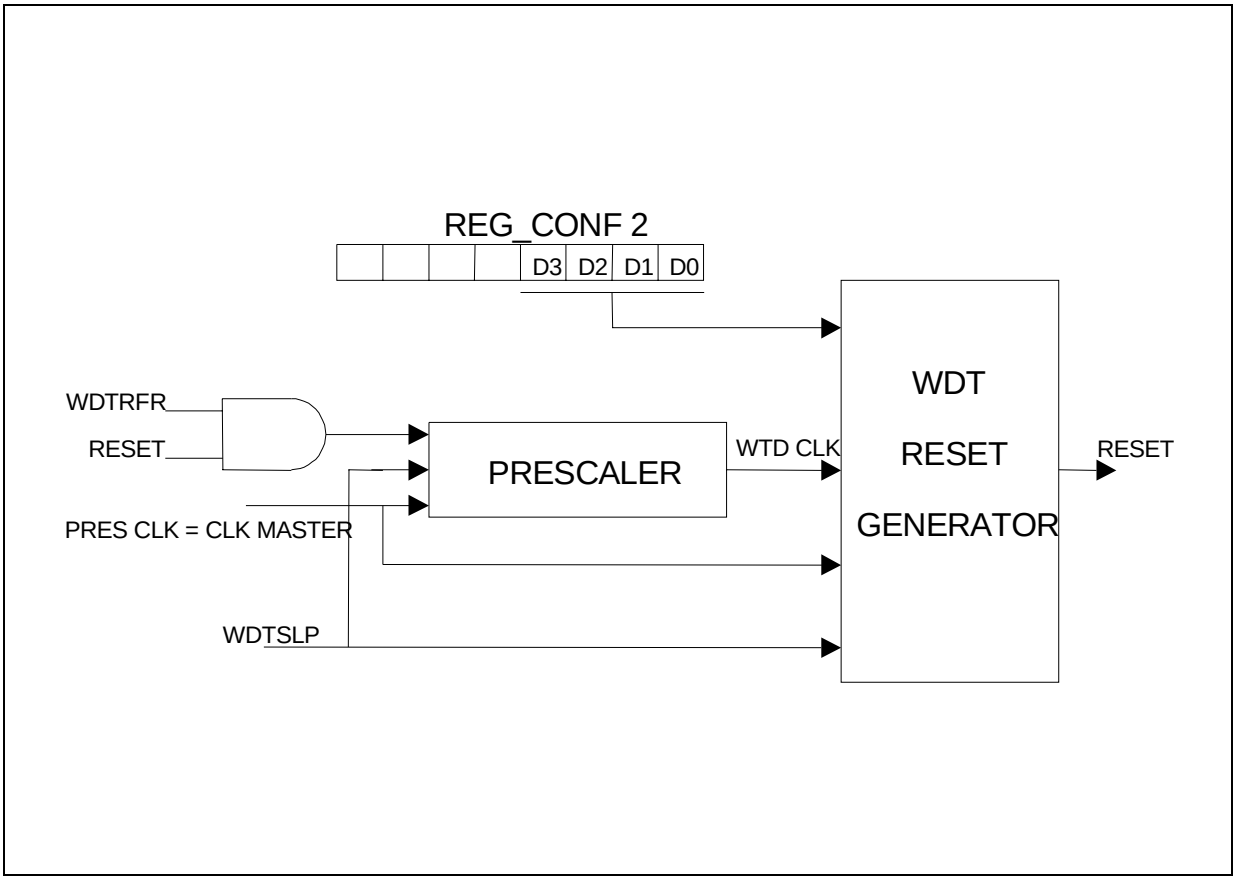
Table 9.1 Watchdog Timing range (CLK=5 MHz)

	WDT timeout period (ms)
min	0.1
max	937.5

According to the WDT configuration register values, a WDT delay may be defined between 0.1 ms and 937.5 mS when the clock master is 5 MHz. By changing the clock master frequency the timeout delay can be calculated according to the configuration register values REG_CONF 2, as described in the following section.

Warning: *changing the REG_CONF2 value when the WDT is active, a WDT reset is generated and the CPU is restarted. To avoid this side effect, use the* WDTSLP *instruction before changing the REG_CONF2.*

Figure 9.1 Watchdog Block Diagram



9.2 Register Description

The WDT timeout is defined by setting the value of the REG_CONF 2. The first 4 bits of this register are used, obtaining 16 different delays as illustrated in Table 9.2. In Table 9.2 timeout is expressed by using the number of WDT CLK. The WDT CLK is derived from the clock master by a division factor of 500. Timeout is obtained by multiplying the WDT CLK pulse length for the number of pulses defined by the configuration register REG_CONF 2. Table 9.4 illustrates the pulse lengths for typical values of the clock master.

Table 9.3 illustrates the timeout WDT values when the Master Clock is 5 MHz.

Table 9.2 WDT REG_CONF 2

Bit	Name	Value	Timeout Values (WDT
0	D(3:0)	0000	1
		0001	625
		0010	1250
		0011	1875
1		0100	2500
		0101	3125
		0110	3750
		0111	4375
2		1000	5000
		1001	5625
		1010	6250
		1011	6875
3		1100	7500
		1101	8125
		1110	8750
		1111	9375
4-7	NC	x	Not Used
Reset Configuration ‘0000’			

Table 9.3 Timeout Values with CLK = 5 MHz

Bit	Name	Value	Timeout Values (ms)
0	D (3:0)	0000	0.1
		0001	62.5
		0010	125
		0011	187.5
1		0100	250
		0101	312.5
		0110	375
		0111	437.5
2		1000	500
		1001	562.5
		1010	625
		1011	687.5
3		1100	750
		1101	812.5
		1110	875
		1111	937.5
4-7	NC	x	Not Used
Reset Configuration ‘0000’			

Table 9.4 Typical WDT CLK Pulse Length

MASTER CLK (MHz)	WDT CLK (KHz)	WDT CLK PULSE LENGTH (ms)
4	8	0.125
5	10	0.1
8	16	0.0625
10	20	0.05
20	40	0.025

10 PWM/TIMER

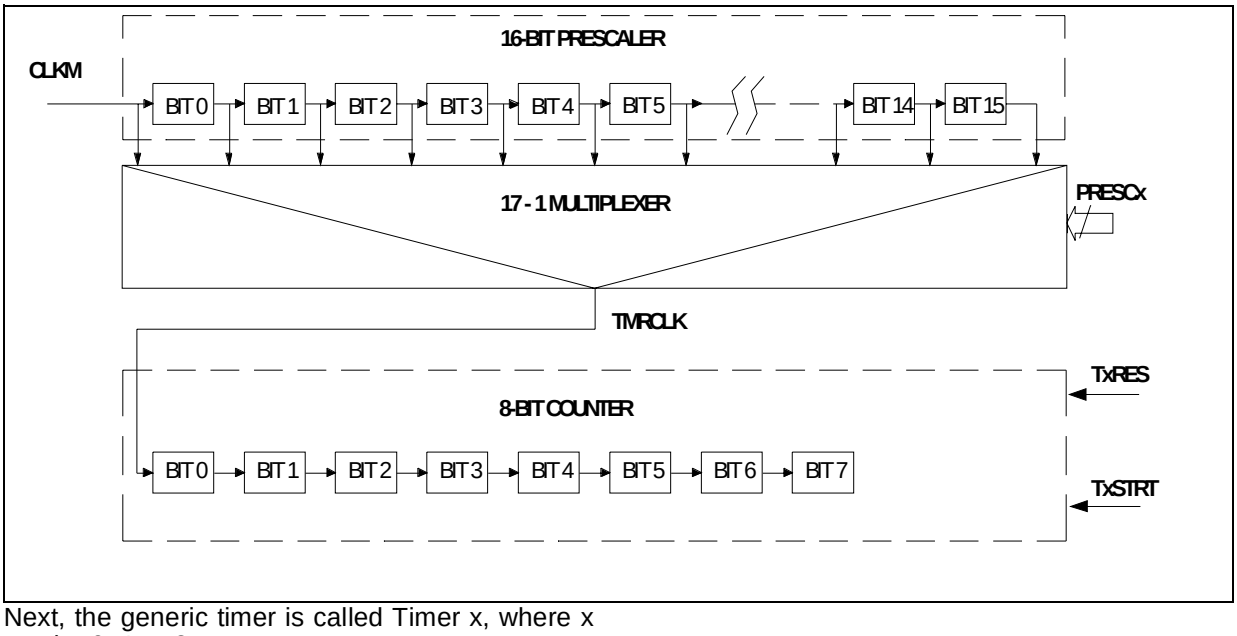
ST52x430 offers three on-chip PWM/Timer peripherals:TIMER0, TIMER1 and TIMER2.

The ST52x430 timers have the same internal structure. The timer consists of an 8-bit counter with a 16-bit programmable prescaler, giving a maximum count of 2^{24} (see Figure 10.1).

Note: In order to use T0RST, T0STR, T0CLK external signals the related pins must be configured in Input Mode by setting REG_CONF4 and REG_CONF7 registers (see Table 6.4 and Table 10.3)

For each timer, the content of the 8-bit counter is incremented on the Rising Edge of the 16-bit prescaler output (PRESCOUT) and it can be read at any instant of the counting phase, saved in a location of RAM memory. The PWM/Timer x Counter value can be read from the Input Register

Figure 10.1 Timer Peripheral Block Diagram



Next, the generic timer is called Timer x, where x can be 0, 1 or 2.

Each timer has two different working modes, which can be selected by setting the correspondent TxMODE bits of REG_CONF5, REG_CONF8 and REG_CONF10 registers: Timer Mode and PWM (Pulse Width Modulation) Mode.

All Timers have Autoreload Functions in PWM Mode.

Each timer output is available, with its complementary signal on external pins by setting PAx and PCx bits of REG_CONF12 and REG_CONF16 (see Table 10.8 and Table 10.9).

Note: In order to enable timer output (TxOUT or TxOUT) the related pin must be configured in Output Mode by setting REG_CONF4 and REG_CONF15 registers (see Table 6.4 and Table 6.6)

In particular, TIMER0 can also use external START/STOP signals (Input capture and Output compare), external RESET signal and external CLOCK: PA4/T0STRT, PA0/T0RES and PA5/ T0CLK pins.

PWM_x_COUNT (Input Registers 12, 14 or 16. See Table 2.2).

The PWM/Timer x Status can be read from the Input Register PWM_x_STATUS (Input Registers 13, 15 or 17. See Table 2.2 and Table 10.10).

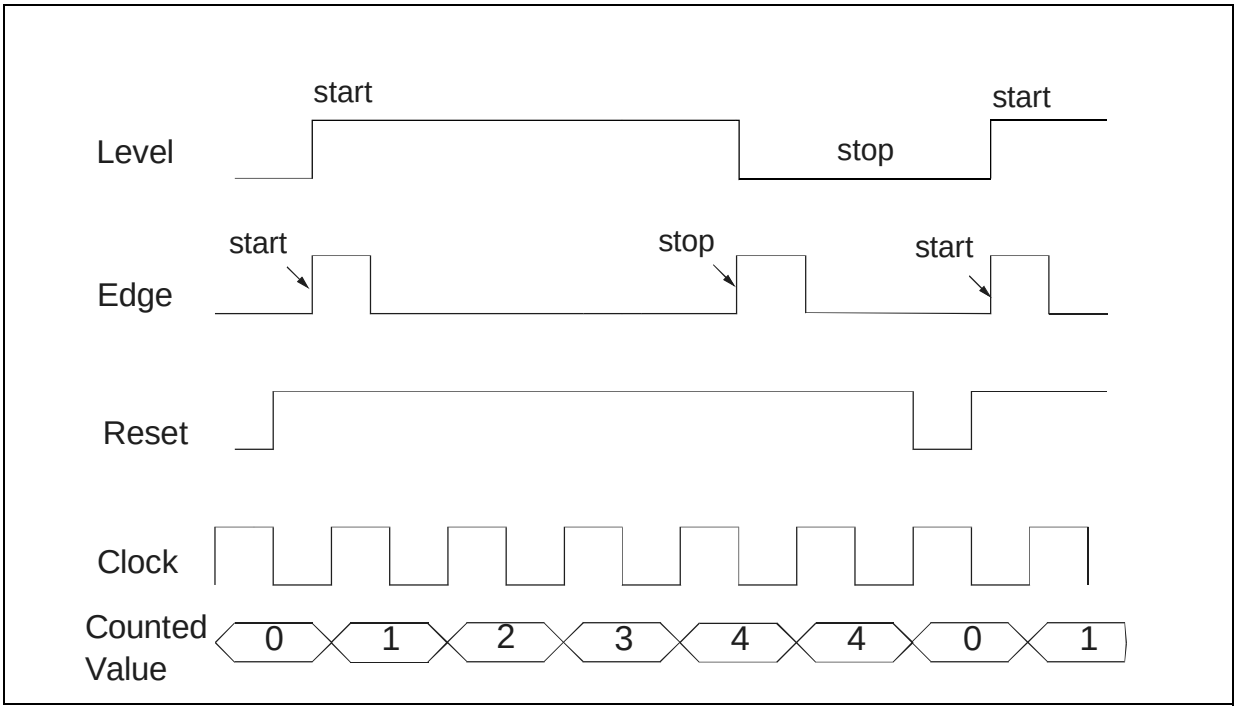
10.1 Timer Mode

Timer Mode is selected by fixing the TxMODE bit of REG_CONF5, REG_CONF8 and REG_CONF10 equal to 0 (see Table 10.1, Table 10.4 and Table 10.6).

Each TIMERx requires three signals: Timer Clock (TMRCLKx), Timer Reset (TxRES) and Timer Start (TxSTRT) (see Figure 10.1). Each of these signals can be generated internally, or, only for Timer 0, externally by setting T0RST, T0STR, T0CLK bits of REG_CONF7 register.

TMRCLKx is the Prescaler x output, which increments the Counter x value on the rising edge. TMRCLKx is obtained from the internal clock signal (CLKM) or, only for TIMER0, from the external signal provided on the PA5/T0CLK pin.

Figure 10.2 Timer 0 External START/STOP Mode



NOTE: The external clock signal applied on the T0CLK pin must have a frequency at least two times smaller than the internal master clock.

The prescaler output can be selected by setting the PRESCx bit of REG_CONF6, REG_CONF9 and REG_CONF11 registers (see Table 10.2, Table 10.5 and Table 10.7).

TxRES resets the content of the 8-bit counter x to zero. It is generated by the TIRSTx and TxMSK bits of REG_CONF5, REG_CONF7, REG_CONF8 and REG_CONF10 registers (see Table 10.1, Table 10.3, Table 10.4 and Table 10.6).

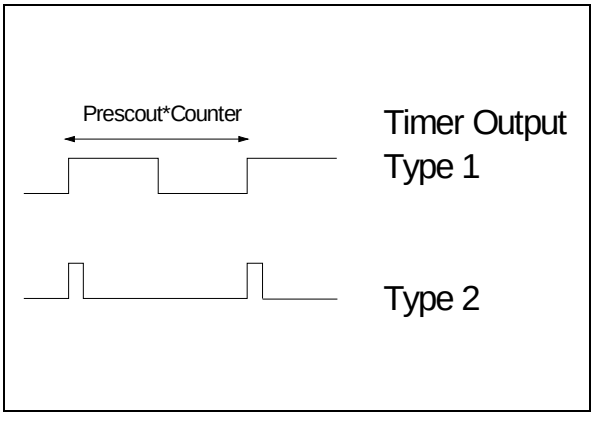
TxSTRT signal starts/stops Timer x counting only if the peripherals are configured in Timer mode. This signal is forced by setting the correspondent TISTRx bit of REG_CONF5, REG_CONF8 and REG_CONF10 registers (see Table 10.1, Table 10.4 and Table 10.6).

TxMSK bits mask the reset of each timer and can be utilized to synchronize a simultaneous start of the timers by means (for example), of the following procedure, which starts three timers:

1) TIRST0 = TIRST1 = TIRST2 = 0,
2) TISTR0 = TISTR1 = TISTR2 = 0,
3) T0MSK = T1MSK = T2MSK = 1,
4) TIRST0 = TIRST1 = TIRST2 = 1,
5) TISTR0 = TISTR1 = TISTR2 = 1,
6) T0MSK = T1MSK = T2MSK = 0,

When TxMSK is 1 the TIMER x is reset.

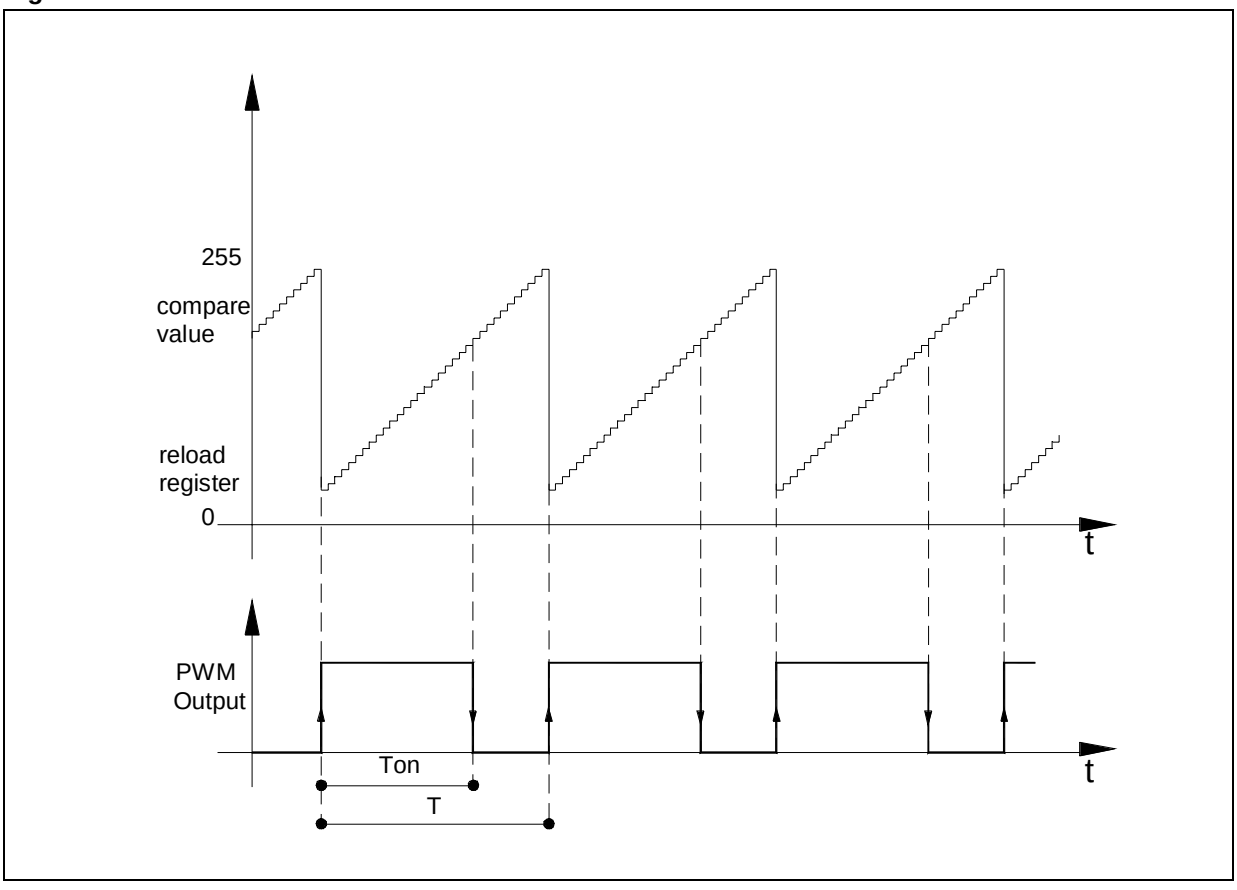
Figure 10.3 TIMEROUT Signal Type



TIMER 0 START/STOP can be provided externally on the T0STRT pin. In this case, the T0STRT signal allows the user to work in two different modes by setting the TESTR configuration bit of REG_CONF5 register (see Figure 10.2) (Input capture):

LEVEL (Time Counter): If the T0STRT signal is high the Timer starts counting. When T0STRT is low the counting ceases and the current value is stored in the PWM_0_COUNT Input Register.

Figure 10.4 PWM Mode with Auto Reload



EDGE(Period Counter): After reset, on the first T0STRT rising edge, TIMER 0 starts counting and at the next rising edge it stops. In this manner, the period of an external signal may be measured.

Timer x output signal, TIMERxOUT is a signal with a frequency equal to the 16 bit-Prescaler x output signal, TMRCLKx, divided by the Output Register PWM_x_COUNT value (8 bit) (Output Registers 3, 5 or 7. See Table 2.4), which is the value to count.

There can be two types of TIMERxOUT waveform:

type 1: TIMERxOUT waveform equal to a square wave with a 50% duty-cycle.

type 2: TIMERxOUT waveform equal to a pulse signal with the pulse duration equal to the Prescaler x output signal.

For each Timer x, the TIMERxOUT waveform type can be selected by setting the correspondent TMRWx bit of REG_CONF6, REG_CONF9 and REG_CONF11 registers (see Table 10.2, Table 10.5 and Table 10.7)

10.2 PWM Mode

For each timer, PWM working mode is obtained by setting the correspondent TxMODE bits of REG_CONF5, REG_CONF8 and REG_CONF10 registers to 1 (see Table 10.1, Table 10.4 and Table 10.6).

TIMERxOUT, in PWM Mode consists of a signal with a fixed period, whose duty cycle can be modified by the user.

The TIMERxOUT signal can be available on the TxOUT pin and the TIMERxOUT inverted signal can be available on the TxOUT pin by setting the PxSL bits of REG_CONF12 and REG_CONF16 (see Table 10.8 and Table 10.9)

The PWM TIMERxOUT period can be determined by setting the 16-bit prescaler x output and an initial autoreload 8-bit counter value stored in the Output Register PWM_x_RELOAD, as illustrated in Figure 10.4.

NOTE: the Start/Stop and Set/Reset signals should be moved together in PWM mode. If the Start/Stop bit is reset during the PWM mode working, the TxOUT signal keeps its status until the next start.

The Output Register PWM_x_RELOAD value is automatically reloaded when Counter x restarts counting.

The 16-bit Prescaler x divides the master clock, CLKM, or, only for TIMER0, the external T0CLK signal, by the 16-bit Prescaler x.

NOTE: The external clock signal, applied on T0CLK pin must have a frequency at least two times smaller than the internal master clock.

The Prescaler x output can be selected by setting PRESCx bit of REG_CONF6, REG_CONF9 and REG_CONF11 registers (see Table 10.2, Table 10.5 and Table 10.7).

When Counter x reaches the Peripheral Register PWM_x_COUNT value (Compare Value), TIMERxOUT signal changes from high to low level, up to the next counter start.

The period of the PWM signal is obtained by using the following equation:

T = (255 - PWM _x_RELOAD)x TMR CLKx

where TMRCLKx is the output of the 16-bit prescaler x.

The duty cycle of the PWM signal is controlled by the Output Register PWM_x_COUNT:

Ton =(PWM_x_COUNT- PWM_x_RELOAD)* TMRCLKx

If the Output Register PWM_x_COUNT value is 255 the TIMERxOUT signal is always at a high level.

If the Output Register PWM_x_COUNT is 0, or less than the PWM_x_RELOAD value, TIMERxOUT signal is always at a low level.

NOTE. If PWM_x_RELOAD value increases the duty cycle resolution decreases.

By using a 20 MHz clock master a PWM frequency in the range 1.2 Hz to 78.43 Khz can be obtained.

NOTE: loading new values of the counter in the PWM_x_COUNT register, in order to avoid side effects, the PWM/Timer counter is updated only at the end of the counting cycle.

WARNING: loading new values of the reload in the PWM_x_RELOAD registers, the PWM/Timer is immediately set on-fly. This can cause some side effects during the current counting cycle. The next cycles work normally. This occurs both in Timer and in PWM mode.

When the Timers are in Reset, or when the device is reset, TxOut pins go in threestate. If these outputs are used to drive external devices it is recommended to put a pull-up or a pull-down resistor.

10.3 Timer Interrupt

TIMERx can be programmed to generate an Interrupt request until the end of the count or when there is an external TSTART signal. The Timer can generate programmable Interrupts into 4 different modes:

Interrupt mode 1: Interrupt on counter Stop.

Interrupt mode 2: Interrupt on Rising Edge of TIMEROUT.

Interrupt mode 3: Interrupt on Falling Edge of TIMEROUT.

Interrupt mode 4: Interrupt on both edges of TIMEROUT.

Interrupt mode can be selected by means of INTSLx and INTEx bits of the REG_CONF5, REG_CONF8 and REG_CONF10 registers (see Table 10.1, Table 10.4 and Table 10.6).

NOTE: the interrupt on TIMEROUT rising edge is also generated after the Start.

WARNING: the first interrupt after starting PWM is not generated if the counter value is 0, 255, or lower than the reload value. If the PWM/ Timer is configured with the Interrupt on Stop and the Start/Stop is configured as external, a low signal in the STRT pin determines a PWM/ Timer interrupt even if the peripheral is off. If the interrupt is configured on falling edge, a reset signal generates an interrupt request.

Table 10.1 Configuration Register 5 Description

Bit	Name	Value	Description
0	TIRST0	0	PWM/TIMER 0 Internal RESET
		1	PWM/TIMER 0 Internal SET
1	TERST	0	External RESET on Level
		1	External RESET on Edge
2	TISTR0	0	PWM/TIMER 0 Internal STOP
		1	PWM/TIMER 0 Internal START
3	TESTR	0	External START on Level
		1	External START on Edge
4	INTE0	00	TIMER0 Interrupt on TIMER Interrupt on TIMEROUT Falling Edge
		01	TIMER0 Interrupt on TIMER0OUT Rising Edge
5		10	TIMER0 Interrupt on Both Edges of TIMER0OUT
		11	- not used
6	INTSL0	0	TIMER0 Interrupt on Counter Stop
		1	TIMER0 Interrupt on TIMER0OUT
7	T0MODE	0	TIMER MODE
		1	PWM MODE

Figure 10.5 Configuration Register 5

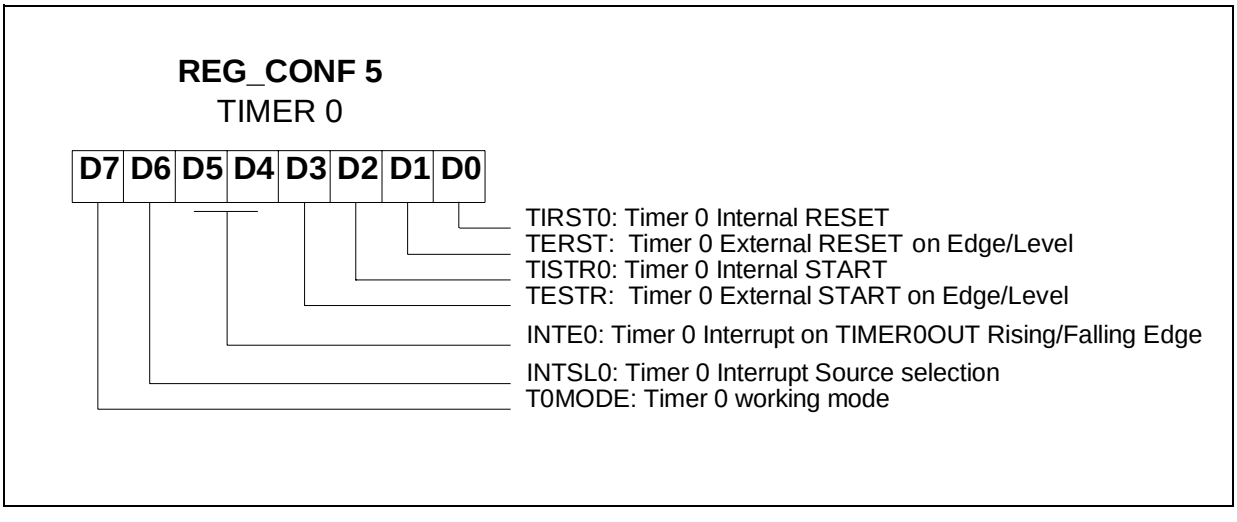


Table 10.2 Configuration Register 6 Description

Bit	Name	Value	Description
0	PRESC0	00000	TIMER0 Clock = CLKM / 1
		00001	TIMER0 Clock = CLKM / 2
		00010	TIMER0 Clock = CLKM / 4
		00011	TIMER0 Clock = CLKM / 8
1		00100	TIMER0 Clock = CLKM / 16
		00101	TIMER0 Clock = CLKM / 32
		00110	TIMER0 Clock = CLKM / 64
2		00111	TIMER0 Clock = CLKM / 128
		01000	TIMER0 Clock = CLKM / 256
		01001	TIMER0 Clock = CLKM / 512
3		01010	TIMER0 Clock = CLKM/1024
		01011	TIMER0 Clock = CLKM/2048
		01100	TIMER0 Clock = CLKM/4096
4		01101	TIMER0 Clock = CLKM/8192
		01110	TIMER0 Clock=CLKM/16384
		01111	TIMER0 Clock=CLKM/32768
		10000	TIMER0 Clock=CLKM /65536
5	TMRW0	0	TIMER0OUT Waveform equal to pulse wave
		1	TIMER0OUT Waveform equal to square wave
6	-	-	- not used
7	-	-	- not used

Figure 10.6 Configuration Register 6

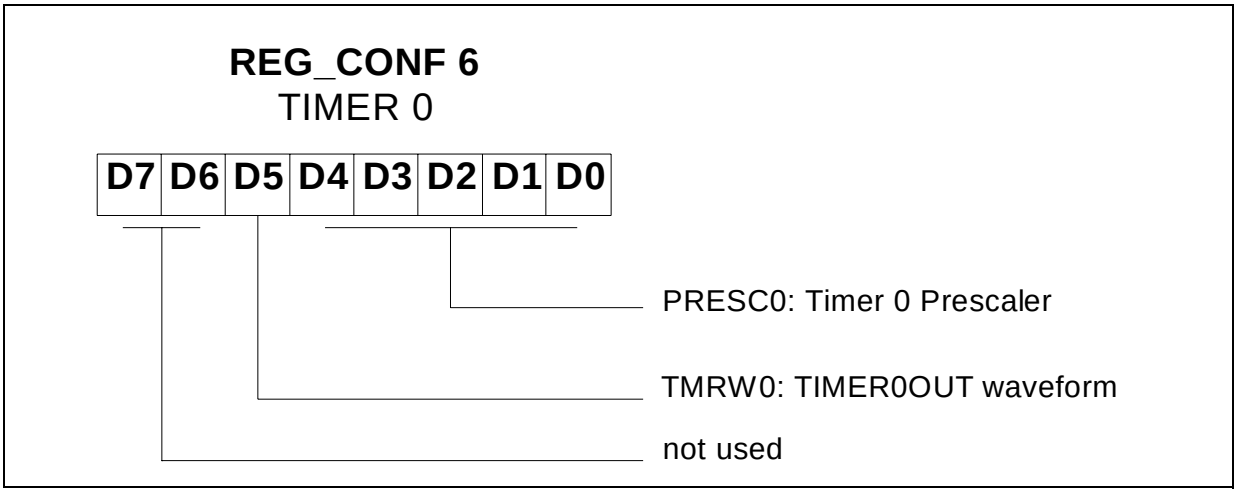


Table 10.3 Configuration Register 7 Description

Bit	Name	Value	Description
0	T0RST	00	TIMER0 RESET Internal
		01	TIMER0 RESET External
1		10	TIMER0 RESET External or Internal
		11	- not used
2	T0STR	00	TIMER0 START Internal
		01	TIMER0 START External
3		10	TIMER0 START External or Internal
		11	- not used
4	T0CLK	0	TIMER0 Clock Internal
		1	TIMER0 Clock External
5	T0MSK	0	TIMER 0 reset synchronization mask.
		1	TIMER0 reset synchronization mask.
6	T2MSK	0	TIMER2 reset synchronization mask.
		1	TIMER2 reset synchronization mask.
7	T1MSK	0	TIMER1 reset synchronization mask.
		1	TIMER1 reset synchronization mask.

Figure 10.7 Configuration Register 7

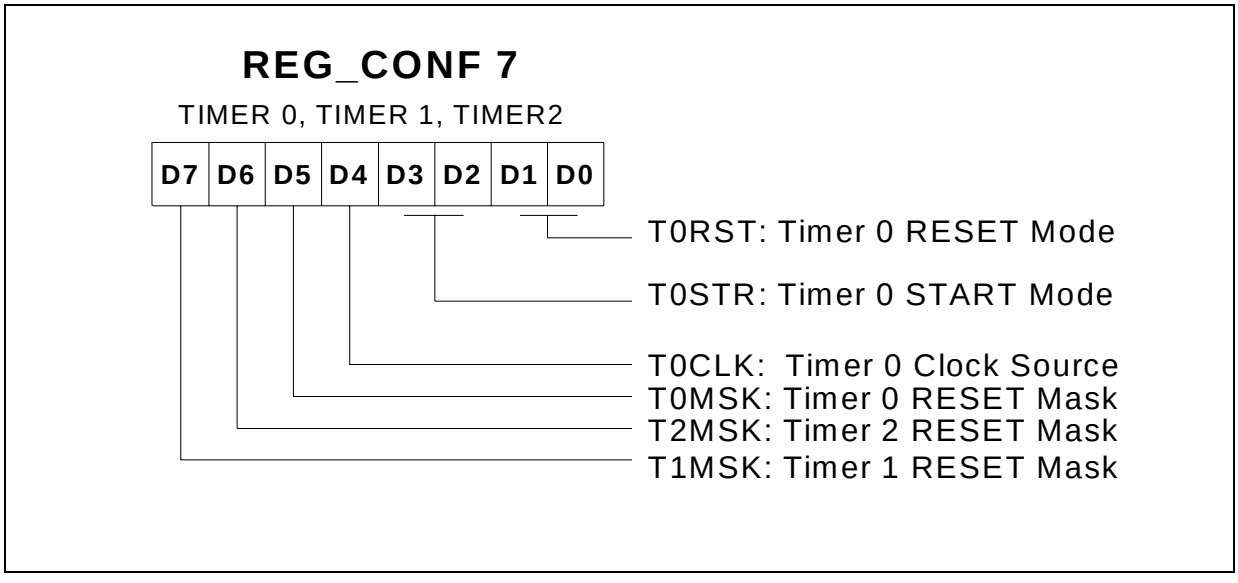


Table 10.4 Config. Register 8 Description

Bit	Name	Value	Description
0	TIRST1	0	PWM/TIMER 1 Internal RESET
		1	PWM/TIMER 1 Internal SET
1	-	-	- not used
2	TISTR1	0	PWM/TIMER 1 Internal STOP
		1	PWM/TIMER 1 Internal START
3	-	-	- not used
4	INTE1	00	TIMER1 Interrupt on TIMER1OUT Falling Edge
		01	TIMER1 Interrupt on TIMER1OUT Rising Edge
5		10	TIMER1 Interrupt on Both Edges of TIMER1OUT
		11	- not used
6	INTSL1	0	TIMER1 Interrupt on Counter Stop
		1	TIMER1 Interrupt on TIMER1OUT
7	T1MODE	0	TIMER MODE
		1	PWM MODE

Figure 10.8 Configuration Register 8

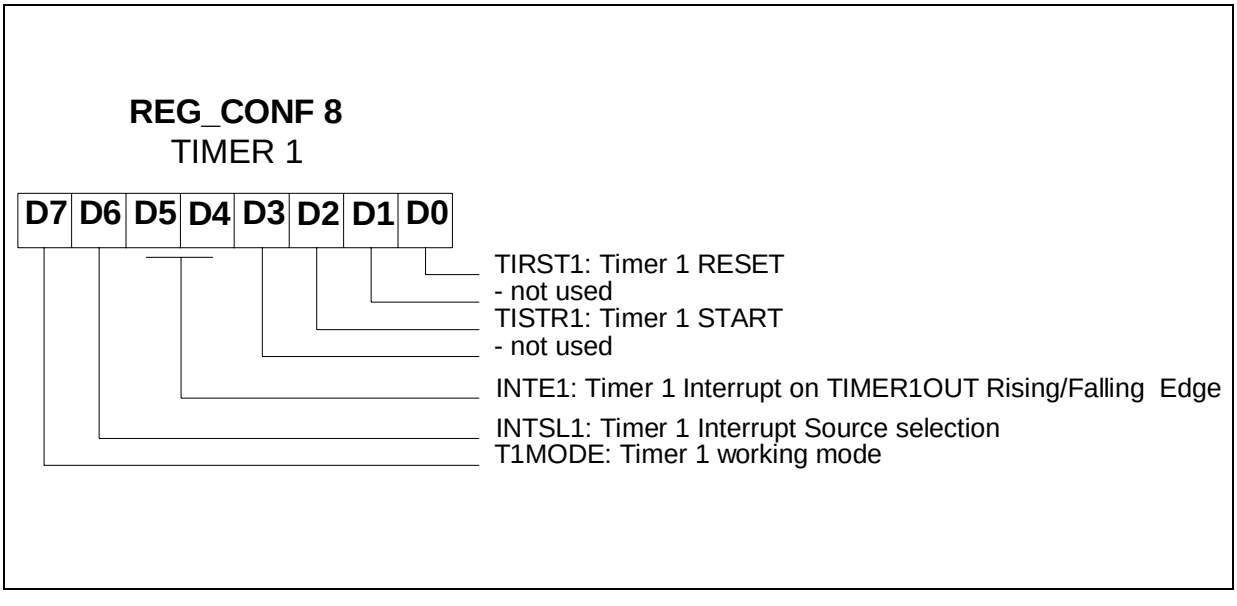


Table 10.5 Config. Register 9 Description

Bit	Name	Value	Description
0	PRESC1	00000	TIMER1 Clock = CLKM / 1
		00001	TIMER1 Clock = CLKM / 2
		00010	TIMER1 Clock = CLKM / 4
		00011	TIMER1 Clock = CLKM / 8
1		00100	TIMER1 Clock = CLKM / 16
		00101	TIMER1 Clock = CLKM / 32
		00110	TIMER1 Clock = CLKM / 64
2		00111	TIMER1 Clock = CLKM / 128
		01000	TIMER1 Clock = CLKM / 256
		01001	TIMER1 Clock = CLKM / 512
3		01010	TIMER1 Clock =CLKM / 1024
		01011	TIMER1 Clock =CLKM / 2048
		01100	TIMER1 Clock =CLKM / 4096
4		01101	TIMER1 Clock =CLKM / 8192
		01110	TIMER1 Clock =CLKM/16384
		01111	TIMER1 Clock=CLKM /32768
		10000	TIMER1 Clock=CLKM /65536
5	TMRW1	0	TIMER1OUT Waveform equal to pulse wave
		1	TIMER1OUT Waveform equal to square wave
6	-	-	- not used
7	-	-	- not used

Figure 10.9 Configuration Register 9

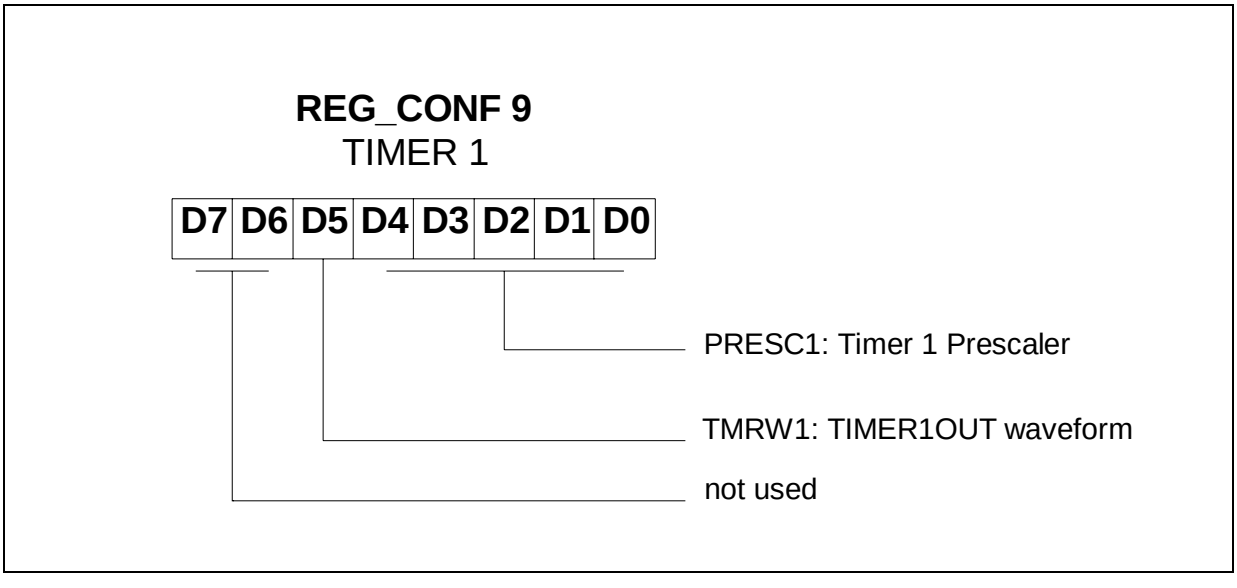


Table 10.6 Config. Register 10 Description

Bit	Name	Value	Description
0	TIRST2	0	PWM/TIMER 2 Internal RESET
		1	PWM/TIMER 2 Internal SET
1	-	-	- not used
2	TISTR2	0	PWM/TIMER 2 Internal STOP
		1	PWM/TIMER 2 Internal START
3	-	-	- not used
4	INTE2	00	TIMER2 Interrupt on TIMER2OUT Falling Edge
		01	TIMER2 Interrupt on TIMER2OUT Rising Edge
5		10	TIMER2 Interrupt on Both Edges of TIMER2OUT
		11	- not used
6	INTSL2	0	TIMER2 Interrupt on Counter Stop
		1	TIMER2 Interrupt on TIMER2OUT
7	T2MODE	0	TIMER MODE
		1	PWM MODE

REG_CONF 10
TIMER 2

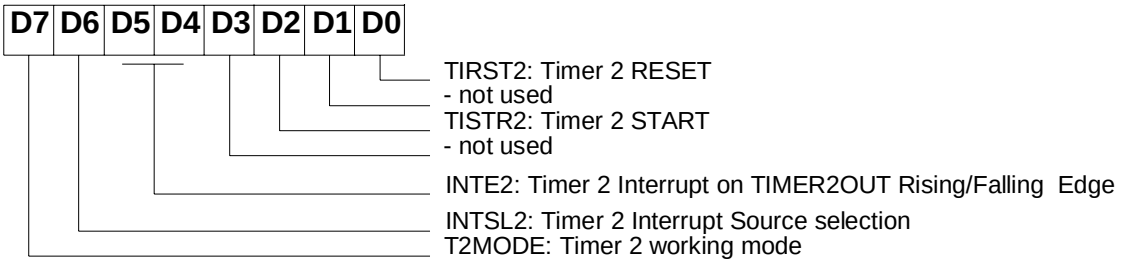


Table 10.7 Config. Register 11 Description

Bit	Name	Value	Description
0	PRESC2	00000	TIMER2 Clock = CLKM / 1
		00001	TIMER2 Clock = CLKM / 2
		00010	TIMER2 Clock = CLKM / 4
1		00011	TIMER2 Clock = CLKM / 8
		00100	TIMER2 Clock = CLKM / 16
		00101	TIMER2 Clock = CLKM / 32
		00110	TIMER2 Clock = CLKM / 64
2		00111	TIMER2 Clock = CLKM / 128
		01000	TIMER2 Clock = CLKM / 256
		01001	TIMER2 Clock = CLKM / 512
3		01010	TIMER2 Clock = CLKM /1024
		01011	TIMER2 Clock = CLKM/ 2048
		01100	TIMER2 Clock = CLKM/ 4096
4		01101	TIMER2 Clock = CLKM/ 8192
		01110	TIMER2 Clock= CLKM/16384
		01111	TIMER2 Clock =CLKM/32768
		10000	TIMER2 Clock =CLKM/65536
5	TMRW2	0	TIMER2OUT Waveform equal to pulse wave
		1	TIMER2OUT Waveform equal to square wave
6	-	-	- not used
7	-	-	- not used

Figure 10.11 Configuration register 11

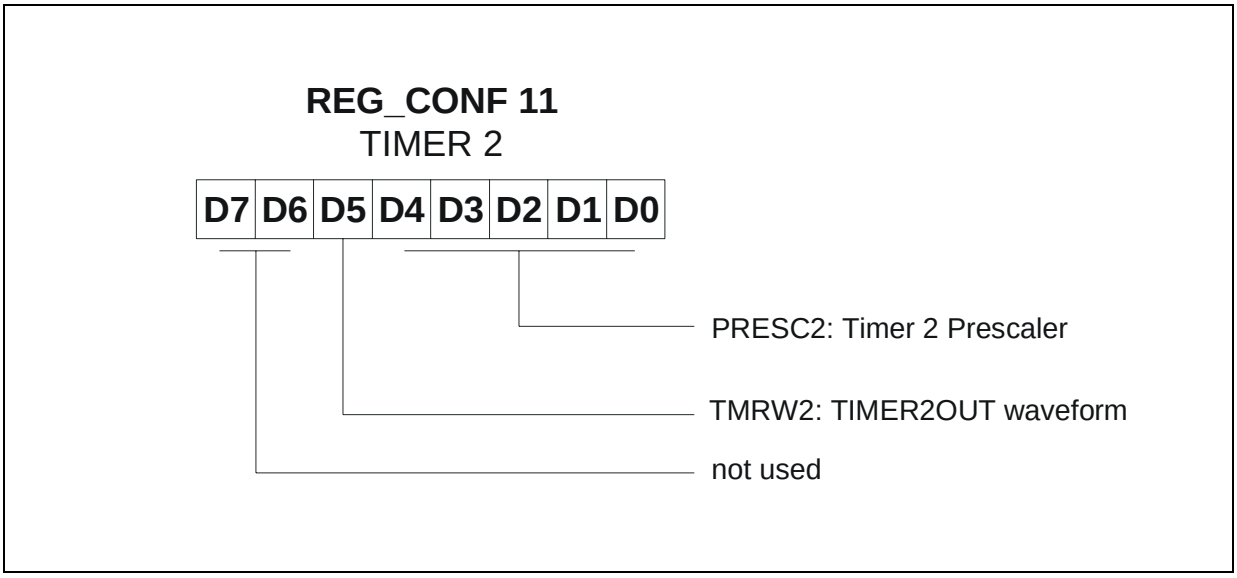


Table 10.8 Config. Register 12 Description

Bit	Name	Value	Description
0	PA1	0	Pin PA1/$\overline{\text{T0OUT}}$ equal to PORT A Digital I/O
		1	Pin PA1/ $\overline{\text{T0OUT}}$ equal to $\overline{\text{T0OUT}}$
1	PA2	0	Pin PA2/ $\overline{\text{T1OUT}}$ equal to PORT A Digital I/O
		1	Pin PA2/ $\overline{\text{T1OUT}}$ equal to $\overline{\text{T1OUT}}$
2	PA3	0	Pin PA3/ $\overline{\text{T2OUT}}$ equal to PORT A Digital I/O
		1	Pin PA3/ $\overline{\text{T2OUT}}$ equal to $\overline{\text{T2OUT}}$
3	PASZ	0	PORT A bits = 7
		1	PORT A bits = 8
4	-	-	- not used
5	-	-	- not used
6	-	-	- not used
7	-	-	- not used

Figure 10.12 Configuration Register 12

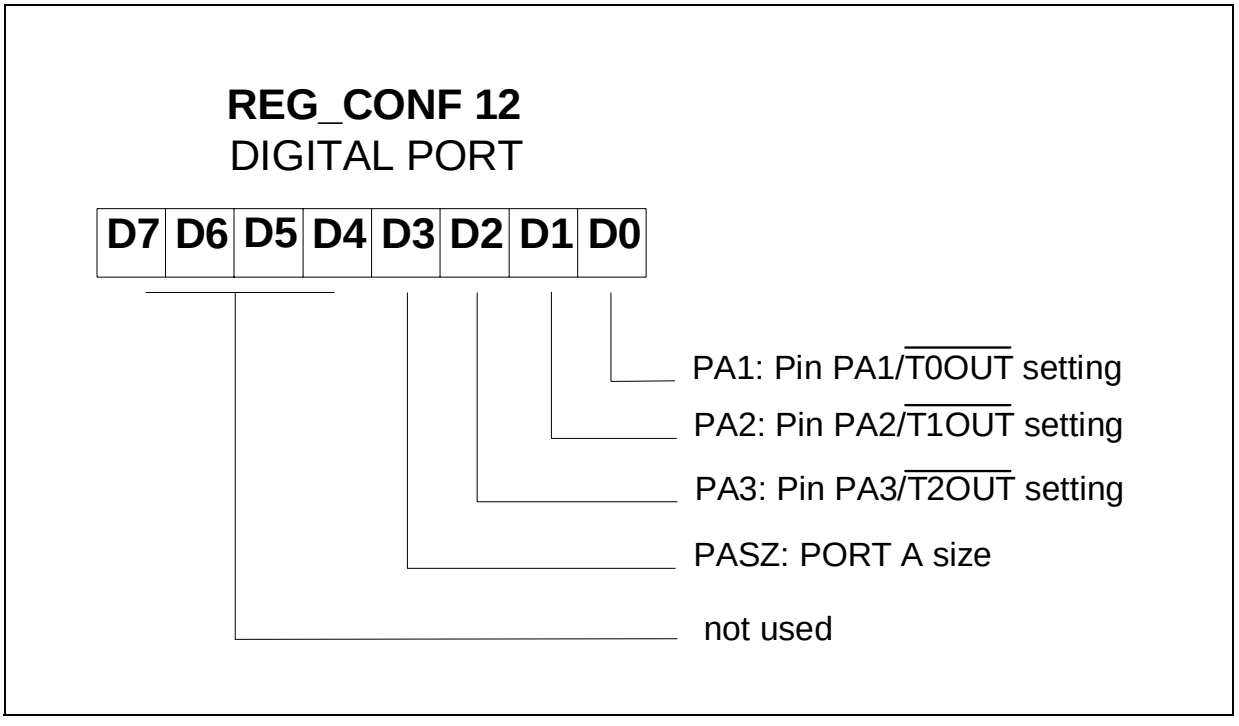


Table 10.9 Config. Register 16 Description

Bit	Name	Value	Description
0	PC1	1	Pin T0OUT/PC1 equal to PORT C Digital I/O
		0	Pin T0OUT/PC1 equal to T0OUT
1	PC2	1	Pin T1OUT/PC2 equal to PORT C Digital I/O
		0	Pin T1OUT/PC2 equal to T1OUT
2	PC3	1	Pin T2OUT/PC3 equal to PORT C Digital I/O
		0	Pin T2OUT/PC3 equal to T2OUT
3	PC4	1	Pin Tx/PC4 is configured as Port C Digital I/O
		0	Pin Tx/PC4 is configured as SCI output Tx
4-7	-	-	- not used

Figure 10.13 Configuration Register 16

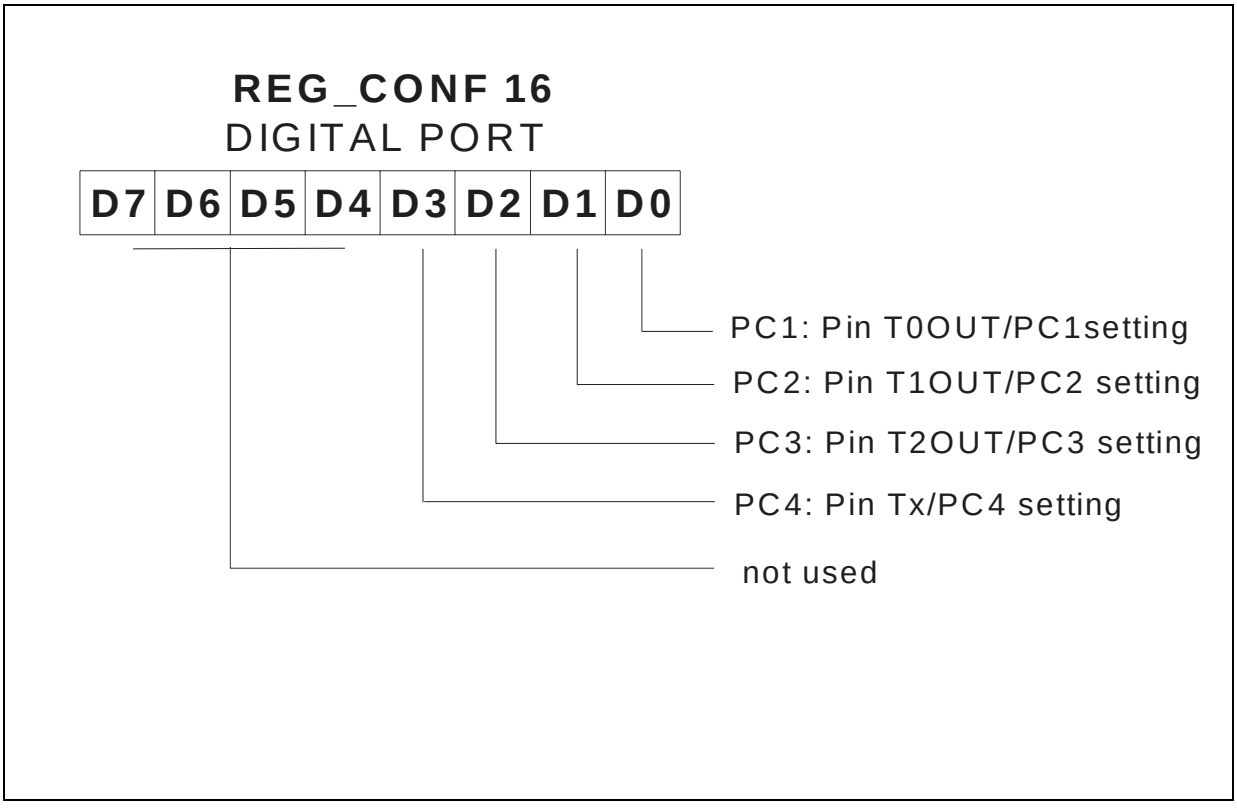


Table 10.10 Input Registers 13. PWM_0_STATUS

Bit	Name	Value	Description
0	STR0	0	TIMER 0 is STOP
		1	TIMER 0 START
1	RST0	0	TIMER 0 is RESET
		1	TIMER 0 is NOT
2	-	-	- not used
3	-	-	- not used
4	-	-	- not used
5	-	-	- not used
6	-	-	- not used
7	-	-	- not used

Table 10.11 Input Registers 15. PWM_1_STATUS

Bit	Name	Value	Description
0	STR1S	0	TIMER 1 is STOP
		1	TIMER 1 is START
1	RST1S	0	TIMER 1 is RESET
		1	TIMER 1 is NOT
2	-	-	- not used
3	-	-	- not used
4	-	-	- not used
5	-	-	- not used
6	-	-	- not used
7	-	-	- not used

Table 10.12 Input Registers 17. PWM_2_STATUS

Bit	Name	Value	Description
0	STR2	0	TIMER 2 is STOP
		1	TIMER 2 is START
1	RST2	0	TIMER 2 is RESET
		1	TIMER 2 is NOT
2	-	-	- not used
3	-	-	- not used
4	-	-	- not used
5	-	-	- not used
6	-	-	- not used
7	-	-	- not used

11 SERIAL COMMUNICATION INTERFACE

The Serial Communication Interface (SCI) integrated into the ST52x430 fuzzy processor provides a general purpose shift register peripheral, which links several widely distributed MCU's through their SCI subsystem. SCI offers a serial interface providing communication with common baud rates up to 38,400 and flexible character format.

SCI is a full-duplex UART-type asynchronous system with standard Non Return to Zero (NRZ) format for the transmitted/received bit. The length of the transmitted word is 10/11 bits (1 start bit, 8/9 data bits, 1 stop bit).

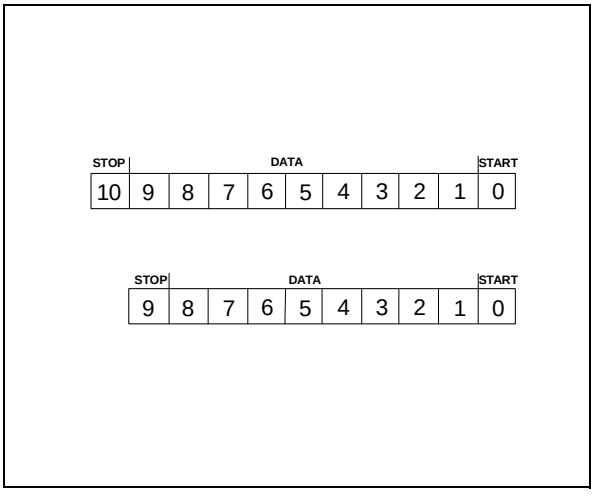
SCI is composed of three modules: Receiver, Transmitter and Baud-Rate Generator. It is configured by means of Configuration Registers 19 and 20.

WARNING: IN ORDER TO WORK PROPERLY WITH SCI PERIPHERALS MAINTAINING THE DESIRED BAUD RATE A SYSTEM CLOCK OF ONLY 5, 10 OR 20 MHz MUST BE USED.

11.1 SCI Receiver block

The SCI Receiver block manages the synchronization of the serial data stream and stores data characters. The SCI Receiver is mainly formed by two sub-systems: Recovery Buffer Block and SCDR_RX Block.

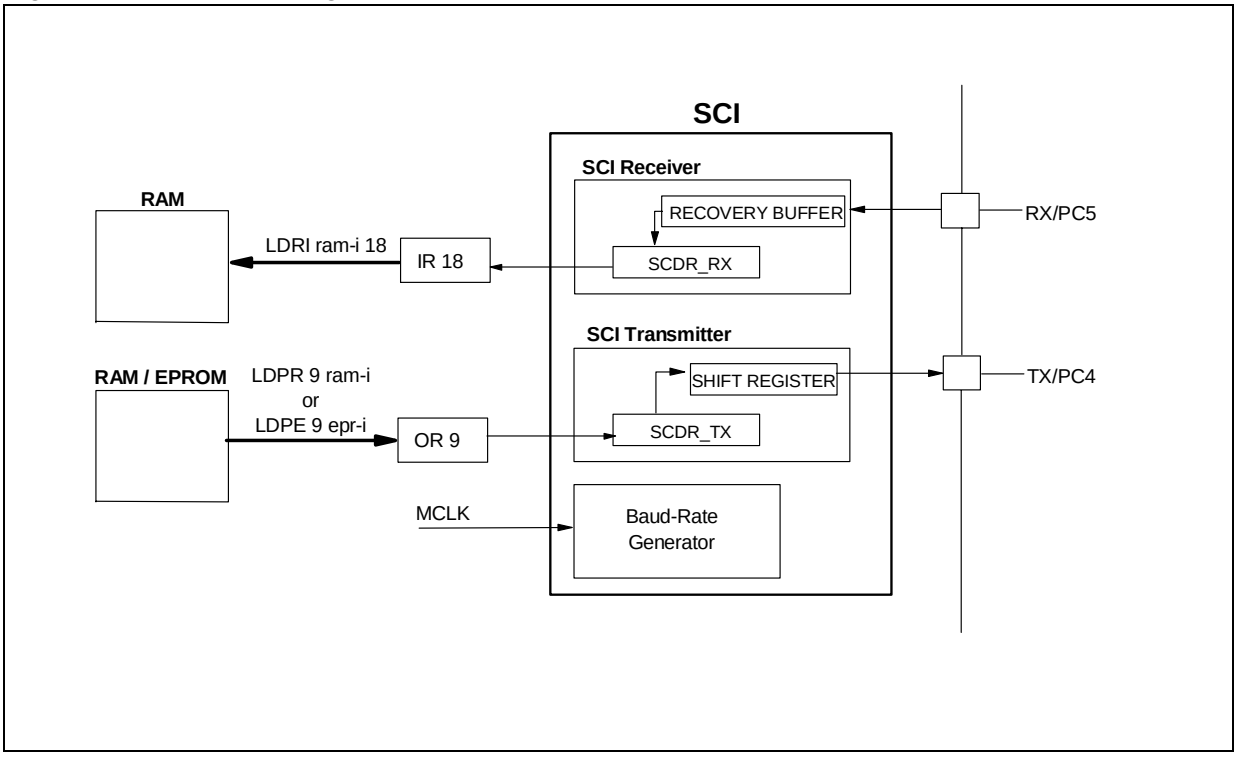
Figure 11.1 SCI transmitted word structures



The RE configuration bit (bit 1 of the Configuration Register 20) enables the SCI Receiver when it is set to "1".

SCI receives data deriving from the RX/PC5 pin and drives the Recovery Buffer Block, which is a high-speed shift register operating at a clock frequency (CLOCK_RX) that is 16 times higher than the fixed baud rate (CLOCK_TX). This sampling rate, higher than the Baud Rate clock allows the detection of the START condition, Noise error and Frame error.

Figure 11.2 SCI Block Diagram



When the SCI Receiver is in IDLE status, it is waiting for the START condition, which is obtained with a logic level 0, consecutive to a logic level 1.

This condition is detected if a logic level 0 is sampled after three logic levels 1 with the fixed sampling time.

The recognition of the START bit forces the SCI Receiver Block to enter in a data acquisition sequence. The data acquisition sequence is configured via Configuration Register 20 as follows.

The 2 bits, M, of the Configuration Register 20 allows the definition of the serial mode as illustrated in Table 11.1.

In the case that M=10, Bit T8 is used to set the parity check in order to perform (as indicated in Table 11.1).

Recognition of the STOP condition allows data received from the Recovery Buffer to be transferred to the SCDR_RX buffer, adding the eventual ninth data bit, according to the meaning illustrated in previous Table 11.1. After this operation, the RXF flag of the SCI Status Input Register 19 (Figure 11.3) is set to logic level 1. The Control Unit reads data from the SCDR_RX buffer (in read-only mode) with the LDRI instruction, addressing Input Register 18, and provides a reset at logic level 0 to the RXF flag.

If data of the Recovery Buffer is ready to be transferred into the SCDR_RX buffer, but the previous one was not read by the Core yet, an OVERRUN Error takes place: the status flag OVERR indicates the error condition. In this case, the information stored in the SCDR_RX buffer is not altered, but the one that has caused the OVERRUN error can be overwritten by new data deriving from the serial data line.

Recovery Buffer Block

This block is structured as a synchronized finite state machine on the CLOCK_RX signal.

When the Recovery Buffer Block is in IDLE state it waits for the reception of the correct 1 and 0 sequence representing START.

The recognition takes place by sampling the input RX/PC5 at CLOCK_RX frequency, which has a frequency that is 16 times higher than CLOCK_TX. While the external transmitter sends a single bit, the Recovery Buffer Block samples 16 states (from SAMPLE1 to SAMPLE16).

The analysis of the RX/PC5 input signal is carried out providing three samples for each bit received.

If these three samples are not equal, then the noise error flag, NSERR, of Input Register 19 is set to 1 and the data value received will be the one assumed by the majority of the samples.

The procedure above allows SCI not to become IDLE, because of a limited noise due to “an erroneous sampling”. The transmission is recognized as correct and the noise flag is set.

Table 11.1 Configuration Register 20 Setting

Bit	Name	Value	Description
0	TE	0	Transmission
		1	Transmission ENABLED
1	RE	0	Receiver DISABLED
		1	Receiver ENABLED
2	M	00	8, No Parity, 1 bit stop
		01	8, No Parity, 2 bit stop
3		10	8, Parity, 1 bit stop
		11	9, No Parity, 1 bit stop
4	T8	0	Parity Odd, if Parity is selected (M = 10); otherwise 9th Data bit
		1	Parity Even, if Parity is selected (M = 10); otherwise 9th Data bit
5	BRSL	000	600 baud
		001	1200 baud
		010	2400baud
6		011	4800 baud
		100	9600 baud
7		101	19200 baud
		110	38400 baud
		111	Not Used

At the end of the reception of a bit, Recovery Buffer Block will repeat the same step 9 times: one for the stop acquisition (10 times in case of 9-bit data, double stop or parity check).

At the end of data reception the Recovery Buffer Block will supply information on eventual frame errors by setting the FRERR flag bit of Input Register 19 to 1.

A frame error can occur if the parity check hasn't been successfully achieved or if the STOP bit hasn't been detected.

If the Recovery Buffer Block receives 10 consecutive bits at logic level 0, a break error occurs and an interrupt routine request starts.

SCDR_RX Block

It is a finite state machine synchronized with the clock master signal, CKM.

The SCDR_RX block waits for the signal of complete reception from the Recovery Buffer, in order to load the word received. Moreover, the SCDR_RX block loads the values of FRERR and NSERR flag bits (Input Register 19), and sets the RXF flag to 1.

Data is transferred to RAM and the RXF flag is reset to 0 by using the LDRI instruction in order to indicate that the SCDR_RX block is empty.

If new data arrives before the previous one has been transferred to Register File, an overrun error occurs and OVERR flag of Input Register 19 is set to 1.

11.2 SCI Transmitter Block

The SCI Transmitter Block consists of the following blocks: SCDR_TX and SHIFT REGISTER, synchronized, respectively, with the clock master signal (CKM) and the CLOCK_TX.

The whole block receives the settings for the following transmission modes (see Table 11.1) through Configuration Register 20 (M bits):

- 8-bit word and a single stop signal
- 8-bit word plus a parity bit and a single stop signal
- 8-bit word plus a double stop signal
- 9-bit word

In case of 9 bit frame transmission, the most significative bit arrives through T8 of the Configuration Register 20.

Instead, in an 8-bit transmission T8 is used to configure SCI according to information contained in M (see Table 11.1). In particular, it is used to choose the polarity control (even or odds) in order to implement the parity check.

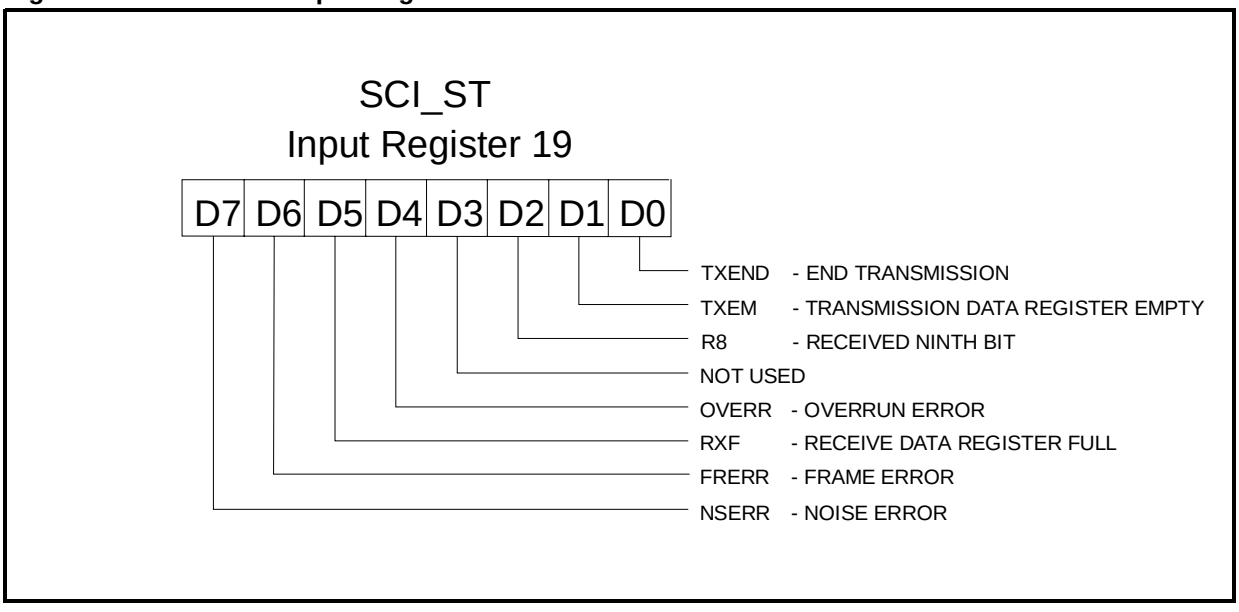
After a RESET signal RST, the SCDR_TX block is in IDLE state until it receives the enabling signal TE=1, of Configuration Register 20.

Data is loaded on the peripheral register (OR 9) by using the instruction LPPR or LDPE. If TE=1 the data to be transmitted is transferred from DR_TX block and flag of Input Register 19. TXEM is reset to 0 in order to indicate that the SCDR_TX block is full.

Table 11.2 Configuration Register 19 Setting

Bit	Name	Value	Description
0		-	Not used
1	ECKF		
		00	5 MHz
		01	10 MHz
2		10	20 MHz
		11	5 MHz
3	TXC	0	SCI End Transmission Interrupt Disabled
		1	SCI End Transmission Interrupt Enabled
4	TDRE	0	SCI Transmission Data Register Empty Interrupt Disabled
		1	SCI Transmission Data Register Empty Interrupt Enabled
5	BRK	0	SCI Break Error Interrupt Disabled
		1	SCI Break Error Interrupt Enabled
6	OVR	0	SCI Overrun Error Interrupt Disabled
		1	SCI Overrun Error Interrupt Enabled
7	RDRF	0	SCI Received Data Register Full Interrupt Disabled
		1	SCI Received Data Register Full Interrupt Enabled

Figure 11.3 SCI Status Input Register



If the core supplies new data it can't be loaded in the SCDR_TX block until the current data hasn't been unloaded on the Shift Register block. Therefore, data may be loaded in the SCDR_TX Block only when TXEM is 1.

When the SHIFT REGISTER Block loads data to be transmitted on an internal buffer, TXEND is reset to 0 in order to indicate the beginning of a new transmission. At the end of transmission TXEND is set to 1, allowing to load new data coming from SCDR_TX in the SHIFT REGISTER.

Note: TXEND = 1 does not mean SCDR_TX is ready to receive new data. For this reason it is better to utilize the TXEM signal in order to synchronize the LDPR instruction to the SCI TRANSMITTER block

If the ST52x430 core resets TE to 0, the transmission is interrupted, but the SCI Transmitter block completes the transmission in progress before reset.

Warning: *after the stop bit in SCI transmission an idle time is present before the next start bit. This time is equal to the duration of a bit transmission.*

11.3 Baud Rate Generator Block

The Baud Rate Generator Block performs the division of the clock master signal (CKM), in a set of synchronism frequencies for the serial bit reception/transmission on the external line.

Table 11.1 illustrates the set of frequencies selected by means of BRSL (Configuration Register 20).

Reception frequency (CLOCK_RX) is 16 times higher than transmission frequency (CLOCK_TX).

The following example illustrates a simple way to use SCI to receive and transmit data:

```
LDRC 1 155

LDCR 20 1      These instructions load value 155
               on the Configuration Register 20
               fixing the Baud Rate=9600, 8 bit
               data, TE=1, RE=1; Parity; 1 stop bit.
LDRC 1 252

LDCR 19 4      SCI Interrupts enabled, clock
               frequency 20 MHz

LDRC 1 170

LDPR 9 1       Send data to transmission buffer

WAITI

LDRI 6 19      Save the SCI status register on the
               RAM

LDRI 1 18      Save the received data on a RAM
               register
```

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 environment temperature, supply voltage and frequencies production testing on 100% of the devices with an environmental temperature at T_A=25°C and T_A=$T_A$max (given by the selected temperature range).

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

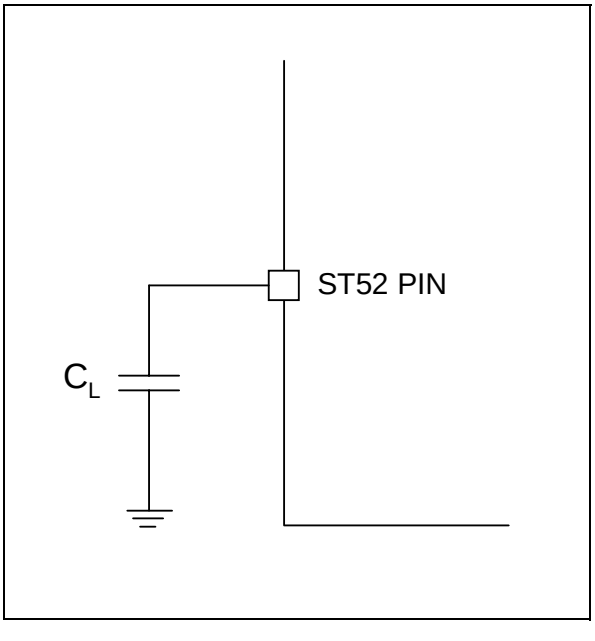
12.1.2 Typical values.

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

12.1.3 Typical curves.

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

Figure 12.1 Pin loading conditions



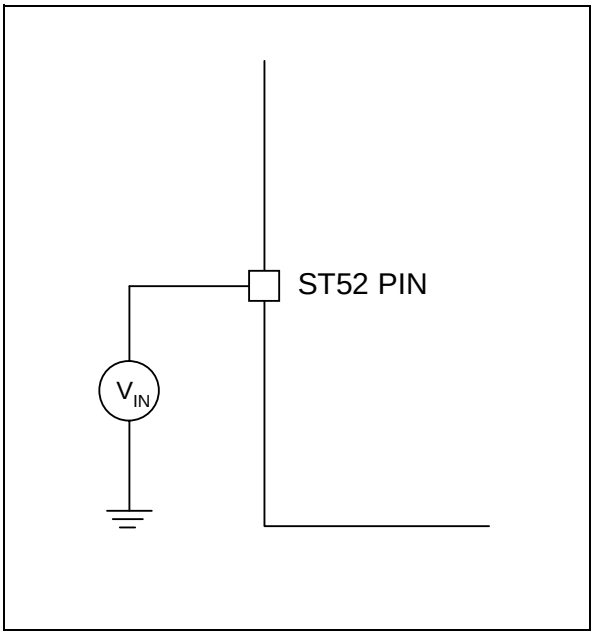
12.1.4 Loading capacitor.

The loading condition used for pin parameter measurement is illustrated in Figure 12.1.

12.1.5 Pin input voltage.

Input voltage measurement on a pin of the device is described in Figure 12.2

Figure 12.2 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.

Functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 12.1 Voltage Characteristics

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

Table 12.2 Current Characteristics

Symbol	Ratings	Maximum Value	Unit
I_{VDD}	Total current in V_{DD} power lines (source)3)	100	mA
I_{VSS}	Total current in V_{SS} ground lines (sink)3)	100	
I_{IO}	Output current sunk by any standard I/O and control pin	25	
	Output current source by any I/Os and control pin	-25	
$I_{INJ(PIN)}$	Injected current on V_{PP} pin	±5	
	Injected current on RESET pin	±5	
	Injected current on OSCin and OSCout pins	±5	
	Injected current on any other pin 4)	±5	
$\Sigma I_{INJ(PIN)}$	Total Injected current (sum of all I/O and control pins) 4)	±20	

Table 12.3 Thermal Characteristics

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

Notes:

1. Connecting RESET and I/O Pins directly to VDD or VSS could damage the device if the unintentional internal reset is generated or an unexpected change of I/O configuration occurs (for example, due to the corrupted program counter). In order to guarantee safe operation, this connection has to be performed via a pull-up or pull-down resistor (typical: 4.7k Ω for RESET, 10K Ω for I/Os). Unused I/O pins must be tied in the same manner to VDD or VSS according to their reset configuration.

2. When the current limitation is not possible, the VIN absolute maximum rating must be respected, otherwise refer to I INJ(PIN) specification. A positive injection is induced by VIN>VDD while a negative injection is induced by VIN<VSSto I INJ(PIN) specification. A positive injection is VIN>VDD while a negative injection is induced by VIN<VSS.

3. All power (VDD) and ground (VSS) lines must always be connected to the external supply.

4. 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).

12.3 Recommended Operating Condition

Operating condition: V_{DD}=5V±10%; T_A=0/125°C (unless otherwise specified).

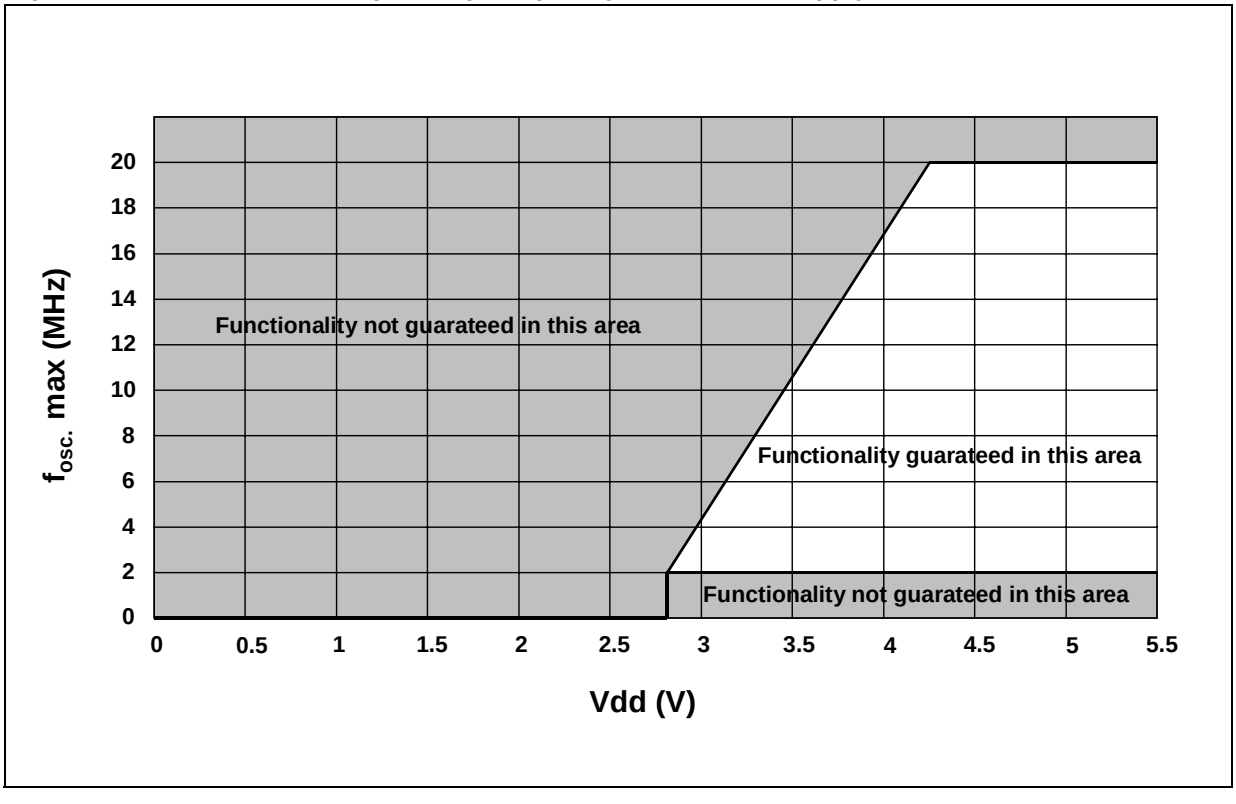
Table 12.4 Recommended Operating Conditions

Symbol	Parameter	Test Condition	Min.	Typ.	Max	Unit
V_{DD} 2)	Operating Supply	Refer to Figure 12.3	3.5		5.5	V
V_{PP}	Programming Voltage		11.4	12	12.6	
V_O	Output Voltage		V_{SS}		V_{DD}	
$V_{DDA,}$	Analog Supply Voltage		V_{DD}-0.3	V_{DD}	V_{DD}+0.3	
V_{SSA}	Analog Ground		V_{SS}-0.3	V_{SS}	V_{SS}+0.3	
f_{OSC} 1)2)	Oscillator Frequency		2		20	MHz

Notes:

1. It is reccomendend to insert a capacitor beetwen V_{DD} and V_{SS} for improving noise rejection. Reccomended values are 10 µF (electrolytic or tantalum) and/or 100 nF (ceramic).
2. In order to use SCI correctly maintaining the programmed baud rates, fosc must be set to 5, 10 or 20 Mhz.
3. A lower V_{DD} decreasing f_{osc} (see Figure 12.3). Data illustrated in the figure are characterized but not tested.

Figure 12.3 fosc Maximum Operating Frequency versus VDD supply



12.4 Supply Current Characteristics

Supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

The test condition in RUN mode for all the IDD measurements are:

OSCin = external square wave, from rail to rail;

OSCout = floating;

All I/O pins tristated pulled to VDD

T_A=90°C

Table 12.5 Supply Current in RUN and WAIT Mode

Symbol	Parameter	Conditions		Typ	Max[3)]	Unit
I_{DD}	Supply current in RUN mode [1)]	V_{DD}=5V±5% T_A=90°C	f_{osc}=2 Mhz	4.0	6.0	mA
			f_{osc}=4 Mhz	7.5	10.0	
			f_{osc}=5 Mhz,	9.0	12.0	
			f_{osc}=10	17.5	20.0	
			f_{osc}=20	33.5	37.0	
	Supply current in WAIT mode[2)]		f_{osc}=2 MHz	2.0	3.0	
			f_{osc}=4 MHz	4.0	5.0	
			f_{osc}=5 MHz	5.0	6.0	
			f_{osc}=10	10.0	12.0	
			f_{osc}=20	18.5	22.0	

Notes:

1. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSCin driven by external square wave).

2. CPU in WAIT mode with all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals switched off; clock input (OSCin driven by external square wave).

3. Data based on characterization results, tested in production at V_{DDmax} and f_{oscmax}.

Figure 12.4 Typical IDD in RUN vs fosc

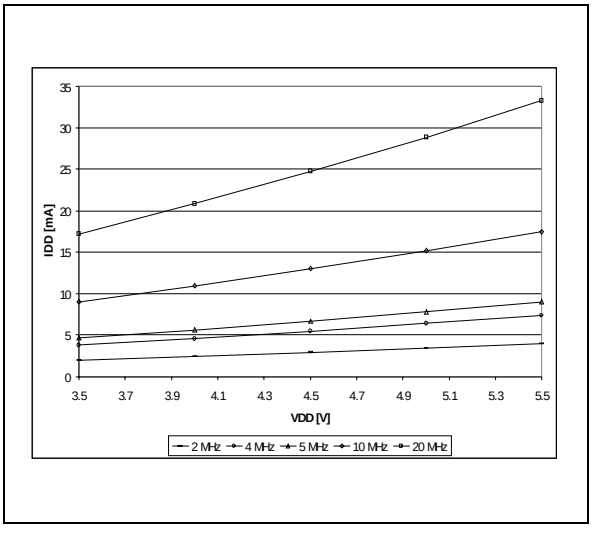


Figure 12.5 Typical IDD in WAIT vs fosc

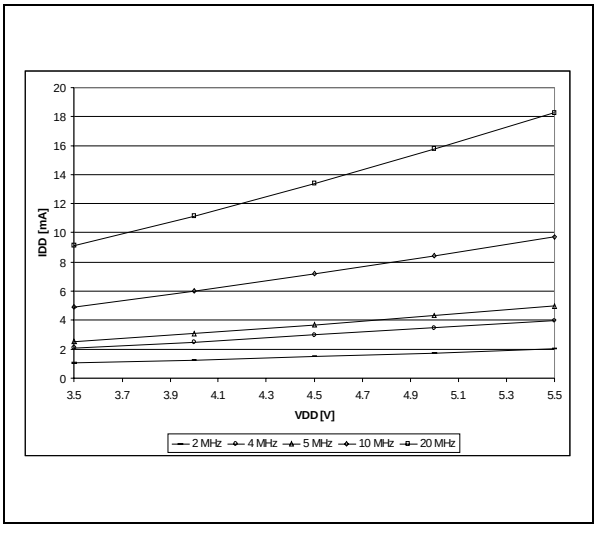


Table 12.6 Supply Current in HALT Mode

Symbol	Parameter	Conditions	Typ[1)]	Max	Unit
I_{DDA}	Supply current in HALT mode[2)]	3.0 V≤ VDD ≤ 5.5 V	1	10	µA

Notes:

1. Typical data is based on TA = 25 °C

2. All I/O pins in input mode with a static value at VDD or VSS (no load)

Table 12.7 On-Chip Peripheral

Symbol	Parameter	Conditions	Typ[3]	Max[4]	Unit
I_{DDA}	ADC Supply current when converting	fosc=20MHz, V_{DDA} = 5 ±5% V, VssA = Vss V_{SSA}=V_{SS}	1	2	mA

Notes:

3. Typical data is based on T_A=25°C, V_{DDA}=5 V.

4. Data is based on characterization results and isn't tested in production.

12.5 Clock and Timing Characteristics

Operating Conditions: V_{DD}=5V ±5%, TA=0/125°C, unless otherwise specified

Table 12.8 General Timing Parameters

Symbol	Parameters	Test Condition	Min	Typ.	Max	Unit
f_{osc}	Oscillator Frequency		1		20	MH
t_{CLH}	Clock High		25		250	nS
t_{CLL}	Clock Low		25		250	
t_{SET}	Setup	See Fig. 11.6		5		
t_{HLD}	Hold	See Fig. 11.6		5		
t_{WRESET}	Minimum Reset Pulse	f_{osc}=20MHz	100			
t_{WINT}	Minimum External	f_{osc}=20MHz	100			
t_{IR}	Input Rise Time	See Fig. 11.7			15	
t_{IF}	Input Fall Time	See Fig. 11.7			15	
t_{OR}	Output Rise Time	C_{LOAD}=10pF		10		
t_{OF}	Output Fall	C_{LOAD}=10pF		10		

Figure 12.6 Data Input Timing

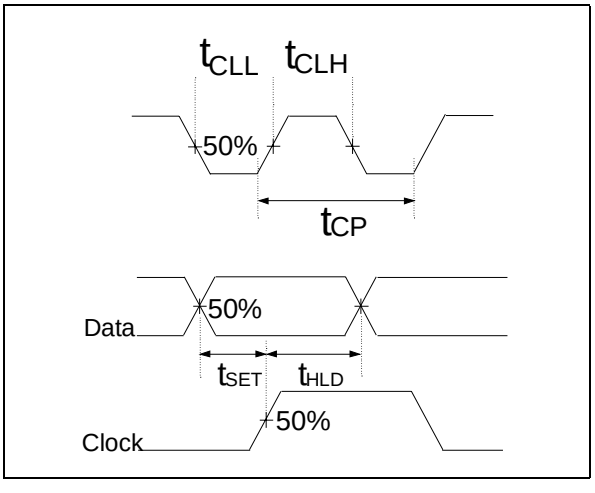
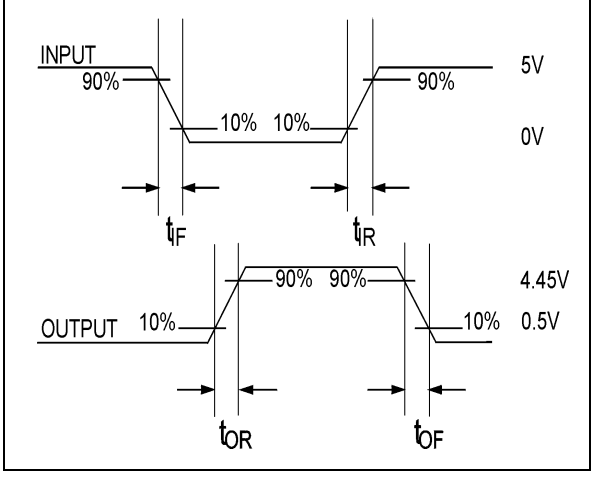


Figure 12.7 I/O Rise and Fall Timing



12.6 Memory Characteristics

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

Table 12.9 RAM and Registers

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

Table 12.10 EPROM Program Memory

Symbol	Parameter	Conditions	Min.	Typ.	Max	Unit
WERASE	UV lamp	Lamp wavelength 2537 A		15		Watt, sec/cm[2)]
tERASE	Erase time[2)]	UV lamp is placed 1 inch from the device window without any interposed filters	15		20	min.
t_{RET}	Data Retention	T_A =+55°C	20			years

Notes:

1. Minimum V_{DD} supply voltage without losing data stored into RAM (in HALT mode or under RESET) or into hardware registers (only in HALT mode). Guaranteed by construction, not tested in production.
2. Data is provided only as a guideline.

12.7 ESD Pin Protection Strategy

In order to protect an integrated circuit against Electro-Static Discharge the stress must be controlled to prevent degradation or destruction of the circuit elements. 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 that are to be protected must not receive excessive current, voltage, or heating within their structure.

An ESD network combines the different input and output protections. This network works by allowing safe discharge paths for the pins subject to ESD stress. Two critical ESD stress cases are presented in Figure 12.8 and Figure 12.9 for standard pins.

12.7.1 Standard Pin Protection

In order 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)

In order protect the input structure the following elements are added:

- A resistor in series with 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 12.8 Safe discharge path subjected to ESD stress

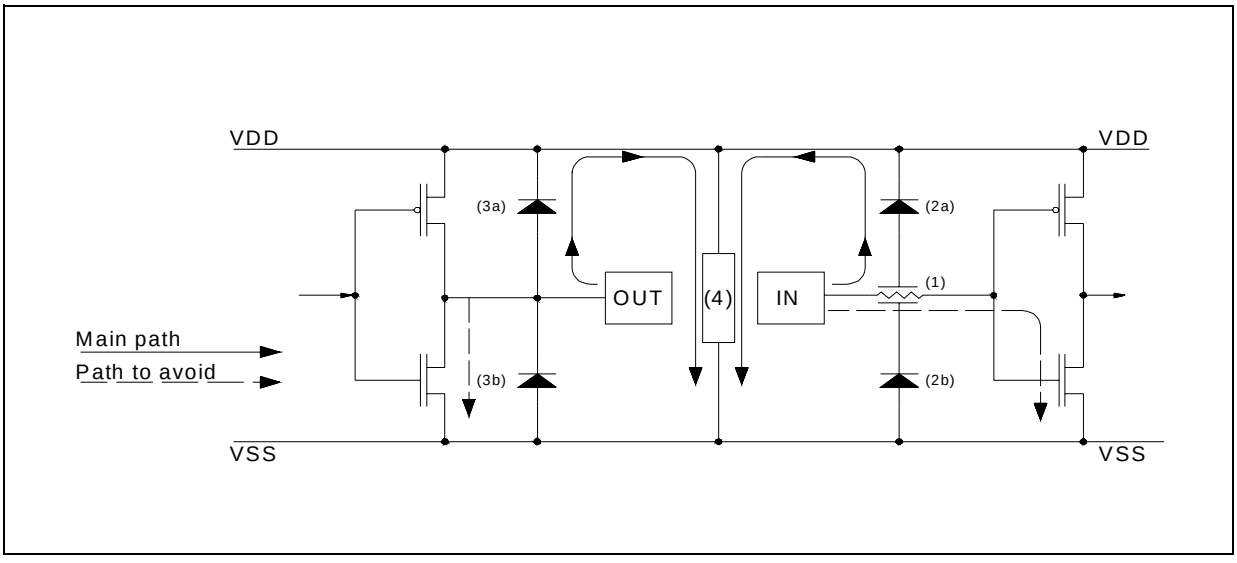
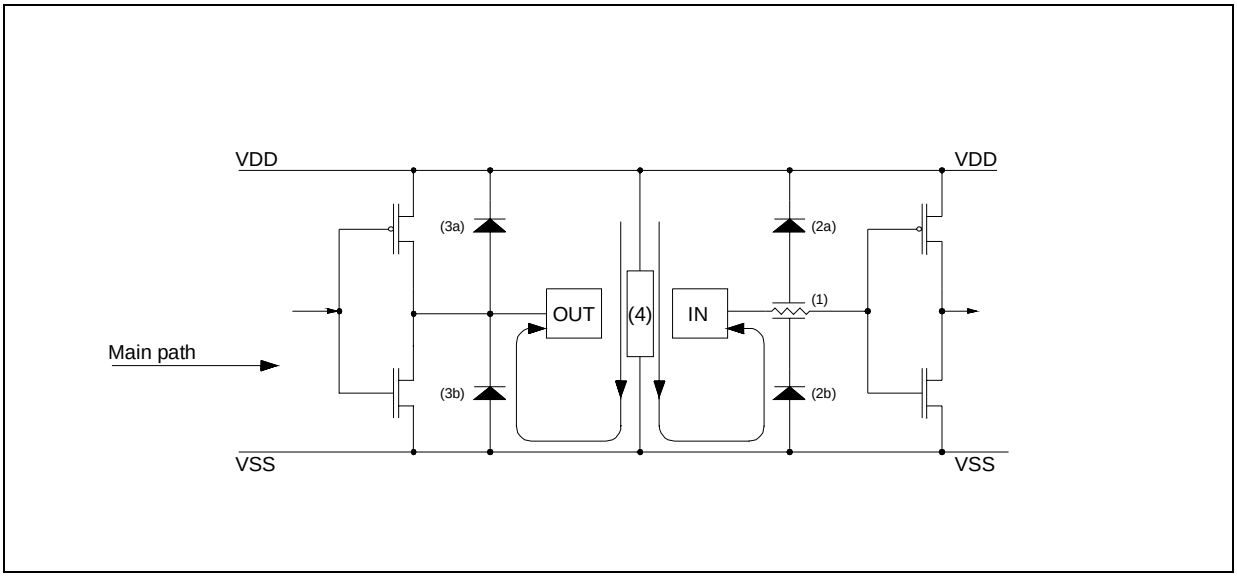


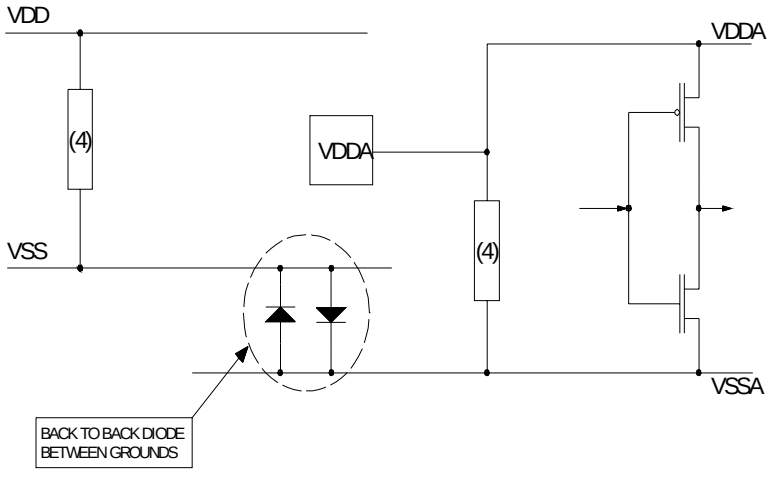
Figure 12.9 Negative Stress on a Standard Pad vs. VDD



12.7.2 Multi-supply 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 illustrated in Figure 12.10 is implemented in order to protect the device against ESD.

Figure 12.10 ESD Protection for Multisupply Configuration



12.8 Port Pin Characteristics

12.8.1 General Characteristics.

Subject to general operating condition for V_{DD}, $f_{osc,}$ and $T_{A\ ,}$ unless otherwise specified.

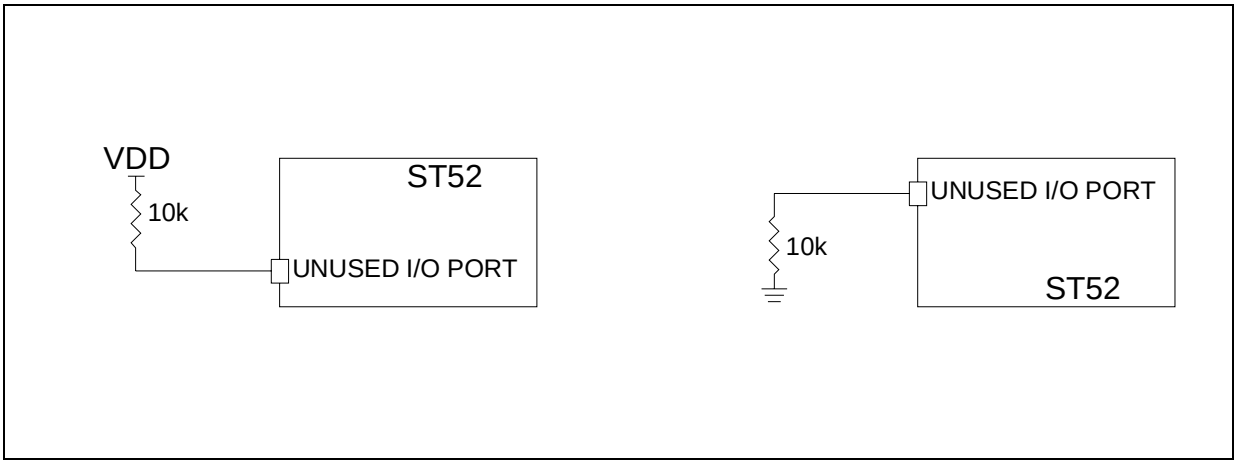
Symbol	Parameter	Condition	Min	Typ[1)]	Max	Unit
V_{IL}	CMOS type low level input voltage. Port B pins. (See Fig 11.13)				1.5	V
	TTL type Schmitt trigger low level input voltage. Port A and Port C pins. (See Fig. 11.12)				0.8	
V_{IH}	CMOS type high level input voltage. Port B pins. (See Fig 11.13)		3.3			
	TTL type Schmitt trigger high level input voltage. Port A and Port C pins. (See Fig. 11.12)		2.2			
V_{hys}	Schmitt trigger voltage hysteresis [2)]			1		
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			±1	µA
I_S	Static current consumption[3)]	Floating input mode			200	

Notes:

1. Unless otherwise specified, typical data is based on T_A=25 °C and V_{DD}=5 V

2. Hysteresis voltage between Schmitt trigger switching level. Based on characterization results, not tested in production.

3. Configuration is 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 12.11. Data based on design simulation and/or technology characteristics is not tested in production.



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

Table 12.11 Output Voltage Levels

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OL} [1)]	Output low level voltage for standard I/O pin when 8 pins are sunk at same time.	V_{DD}=5V, I_{IO}=+8mA			V_{SS}+ 0.4	V
V_{OH} [2)]	Output high level voltage for standard I/ O pin when 8 pins are sourced at same time.	V_{DD}=5V, I_{IO}=- 8mA	V_{DD}- 0.5			

Notes:

1. The I_{IO} current sunk must always respects 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} sourced current 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}.

Figure 12.12 TTL-Level input Schmitt Trigger

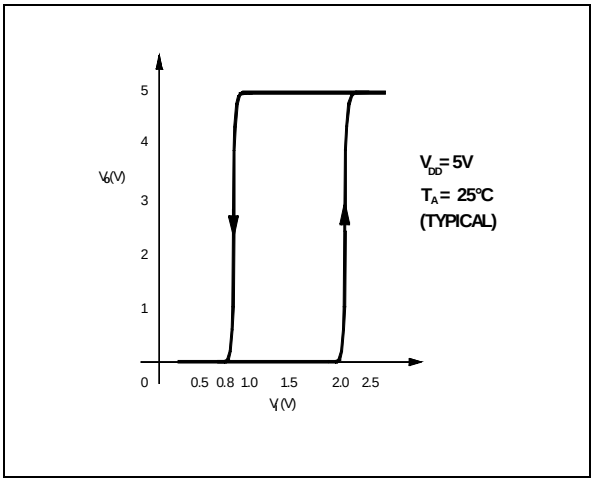
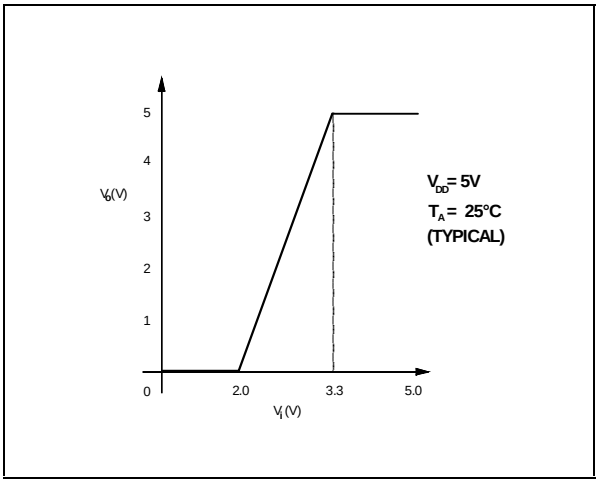


Figure 12.13 Port B pins CMOS-level input



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

Table 12.12 Output Driving Current

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
R_S	Input protection resistor	All input Pins		1		kΩ
C_S	Pin Capacitance	All input Pins		5		pF

Figure 12.14 Port A and Port C pin Equivalent Circuit

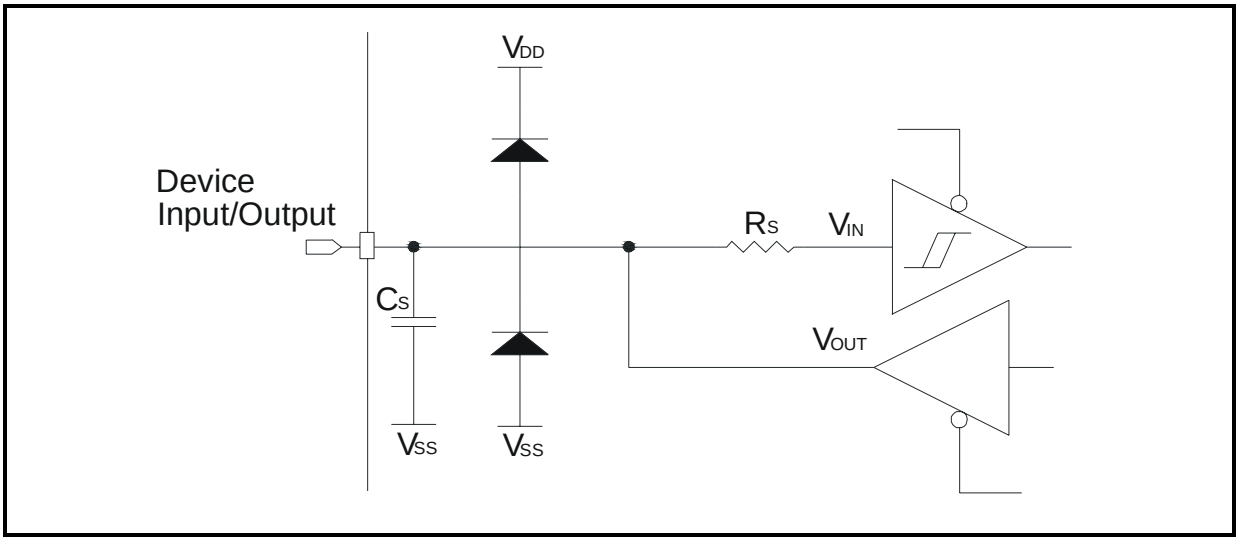
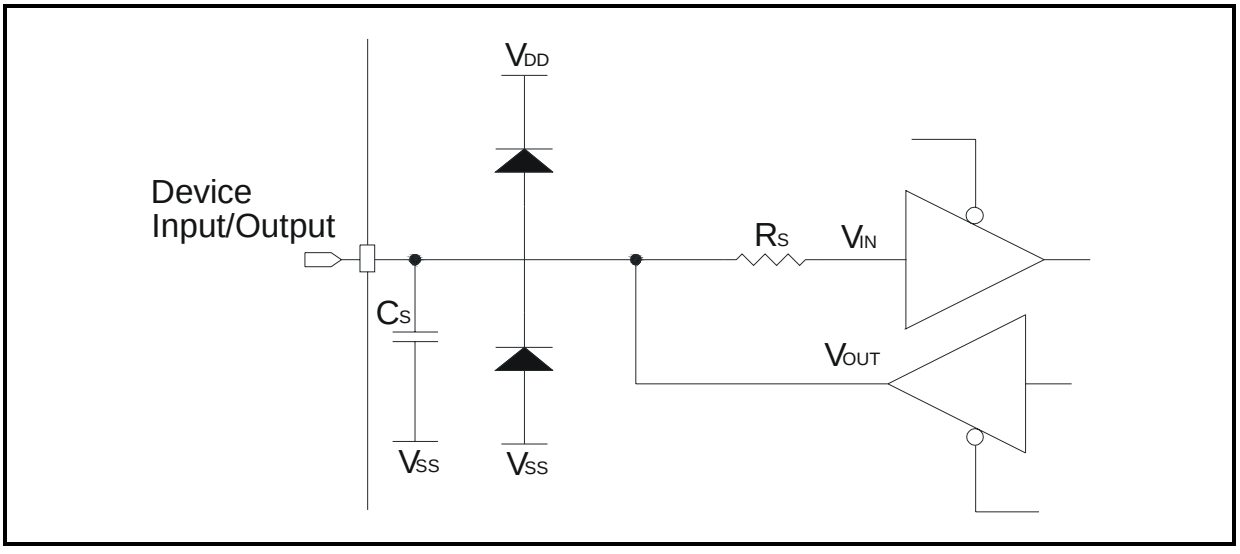


Figure 12.15 Port B Pin Equivalent Circuit



12.9 Control Pin Characteristics

12.9.1 RESET pin.

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

Table 12.13 Reset pin

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage[1)]	V_{DD}= 5 V			1.8	V
V_{IH}	Input high level voltage[1)]	V_{DD}= 5 V	2.8			
V_{hys}	Schmitt trigger voltage hysteresis[2)]	V_{DD}= 5 V		0.8		
$t_{w(RSTL)out}$	General reset pulse duration			30		µS
$t_{h(RSTL)int}$	External reset pulse hold time		20			

12.9.2 V_{PP} pin.

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

Table 12.14 V_{PP}[4)] pin

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage[3)]		V_{SS}		0.2	V
V_{IH}	Input high level voltage[3)]		V_{DD}-0.1		12.6	

Notes:

1. Data is based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching level.
   Based on characterization results not tested in production.
3. Data is based on design simulation and/or technology characteristics, not tested in production.
4. In working mode V_{PP} must be tied to V_{SS}

12.10 8-bit A/D Characteristics

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Res	Resolution			8		bit
A_{TOT}	Total Accuracy[1)]	1 MHz<f_{ADC}< 20 MHz		±1		LSB
t_C	Conversion Time			82/f_{ADC}		µS
V_{AN}	Conversion Range		V_{SSA}		V_{DDA}	V
V_{ZI}	Zero Scale Voltage	Conversion result = 00 Hex		V_{SSA}		V
V_{FS}	Full Scale Voltage	Conversion result = FF Hex		V_{DDA}		V
AD_I	Analog Input Current during Conversion	f_{ADC}=20MHz		0.1		µA
AC_{IN}	Analog Input Capacitance				25	pF
f_{ADC}	ADC Clock frequency		f_{osc}/2		f_{osc}	MHz

Notes:

1. Noise on V_{DDA}, V_{SSA} < 40 mV

Table 12.15 SS034 PACKAGE MECHANICAL DATA

DIM	mm			inch.		
	MIN	TYP.	MAX	MIN	TYP.	MAX
A	2.4638		2.6416	0.097		0.104
A1	0.127		0.2921	0.005		0.0115
B	0.3556		0.4826	0.014		0.019
C	0.23114		0.3175	0.0091		0.0125
D	17.7292		18.0594	0.698		0.711
E	7.4168		7.5946	0.292		0.299
e		1.016			0.040	
H	10.16		10.414	0.400		0.410
h	0.635		0.7366	0.025		0.029
k	0°		8°	0°		8°
l	0.6096		1.016	0.024		0.040

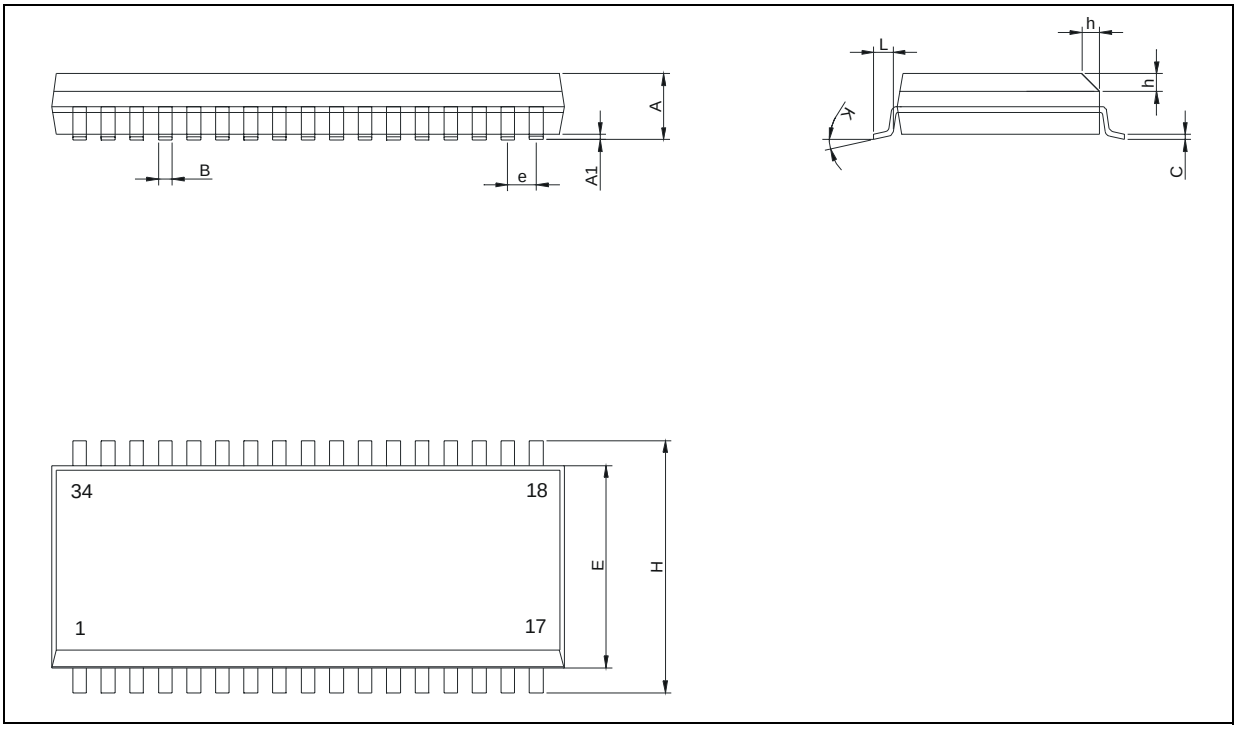


Table 12.16 PDIP32 Shrink PACKAGE MECHANICAL DATA

DIM	mm			inch		
	MIN	TYP.	MAX	MIN	TYP.	MAX
A				0.140	0.148	0.200
A1				0.02		
A2				0.120	0.140	0.18
b				0.014	0.018	0.023
b1				0.030	0.040	0.055
C				0.008	0.010	0.014
D				1.080	1.100	1.120
E				0.390	0.410	0.435
E1				0.300	0.350	0.370
e					0.070	
eAl					0.400	
eB						0.500
eC						0.055
L				0.100	0.120	0.150

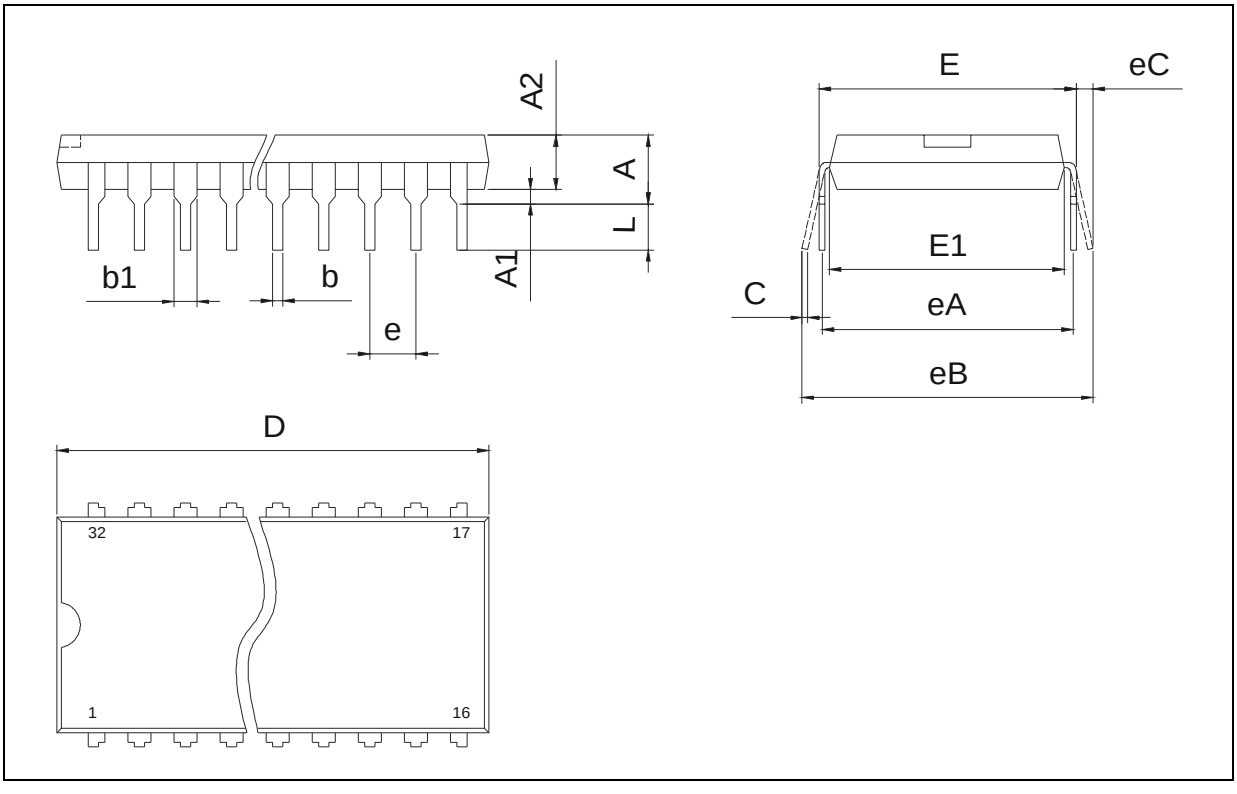


Table 12.17 CSDIP32W Shrink PACKAGE MECHANICAL DATA

DIM	mm			inch.		
	MIN	TYP.	MAX	MIN	TYP.	MAX
A				0.097	0.115	0.133
A1				0.025	0.035	0.045
B				0.016	0.018	0.020
B1				0.035		
C				0.008	0.010	0.012
D				1.168	1.180	1.192
D1				1.042	1.050	1.058
E1				0.382	0.390	0.398
e				0.065	0.070	0.075
G					0.375	
G1					0.580	
G2					0.044	
L				0.170	0.175	0.180
Q						0.290

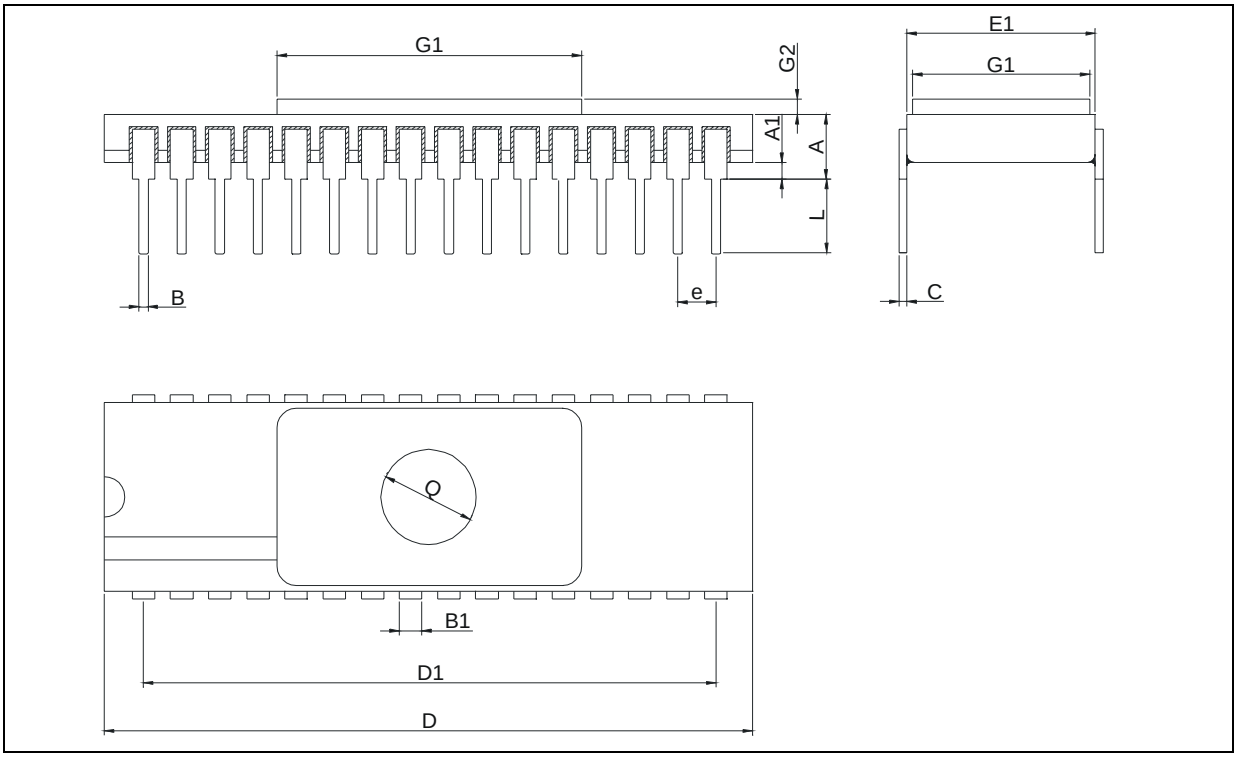
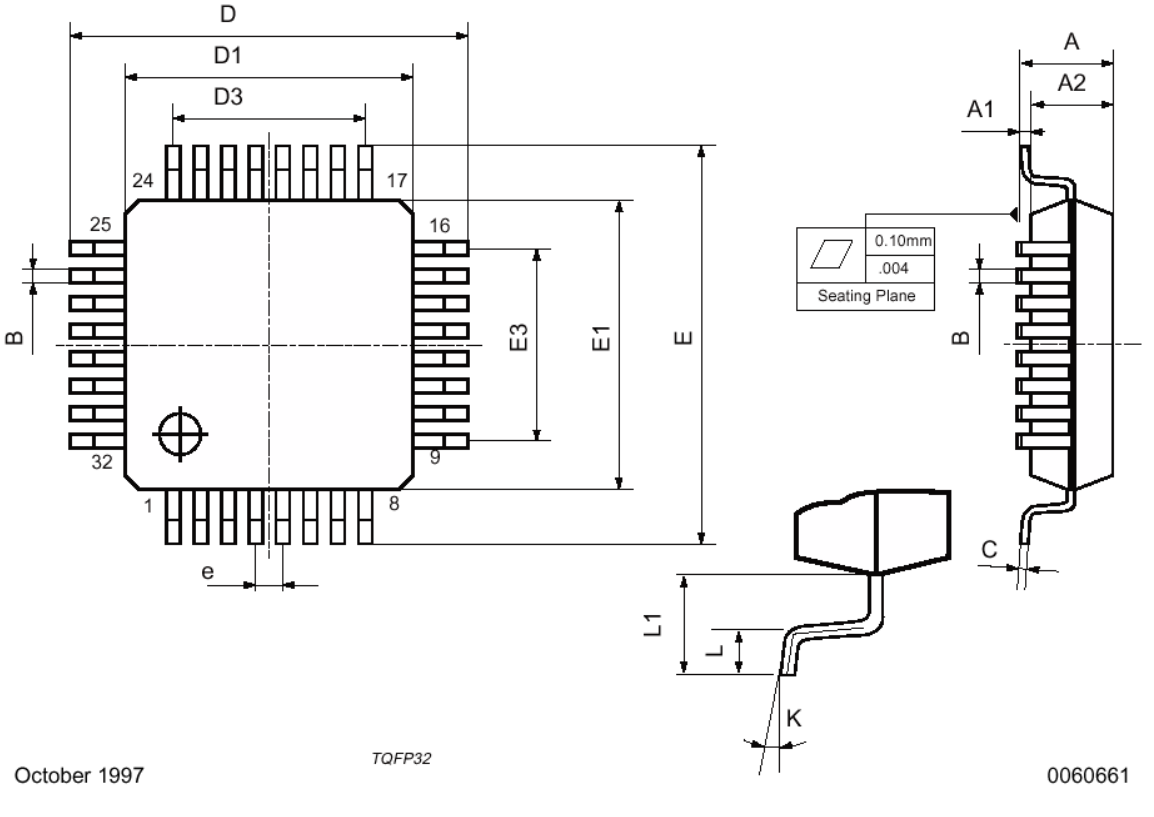


Table 12.18 TQFP32 PACKAGE MECHANICAL DATA

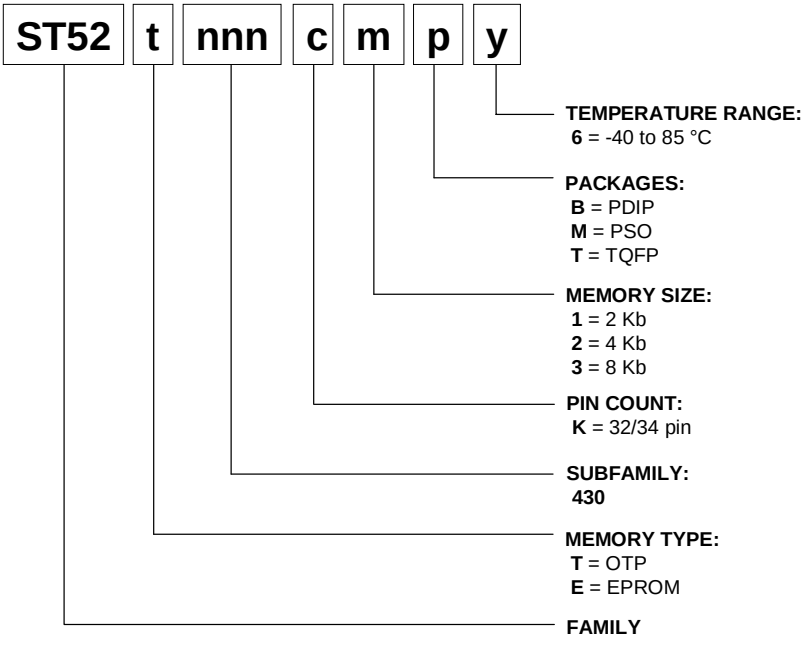
DIM	mm			inch.		
	MIN	TYP.	MAX	MIN	TYP.	MAX
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.30	0.37	0.45	0.012	0.015	0.018
C	0.09		0.20	0.004		0.008
D		9.00			0.354	
D1		7.00			0.276	
D3		5.60			0.220	
e		0.80			0.031	
E		9.00			0.354	
E1		7.00			0.276	
E3		5.60			0.220	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
K	0°(min.), 7°(max.)					



ORDERING INFORMATION

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

Figure 12.16 Device Types Selection Guide



PART NUMBER	TEMPERATURE RANGE	PACKAGE
ST52T430K1M6	-40 to +85 °C	SSO34
ST52T430K2M6	-40 to +85 °C	SSO34
ST52T430K3M6	-40 to +85 °C	SSO34
ST52T430K1B6	-40 to +85 °C	PSDIP32
ST52T430K2B6	-40 to +85 °C	PSDIP32
ST52T430K3B6	-40 to +85 °C	PSDIP32
ST52T430K1T6	-40 to +85 °C	TQFP32
ST52T430K2T6	-40 to +85 °C	TQFP32
ST52T430K3T6	-40 to +85 °C	TQFP32
ST52E430K3D6	-40 to +85 °C	CSDIP32W
ST52X430/KIT		DEVELOPMENT KIT

Full Product Information at http://mcu.st.com

STMicroelectronics GROUP OF COMPANIES
Australia - Brazil - China - Canada - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com