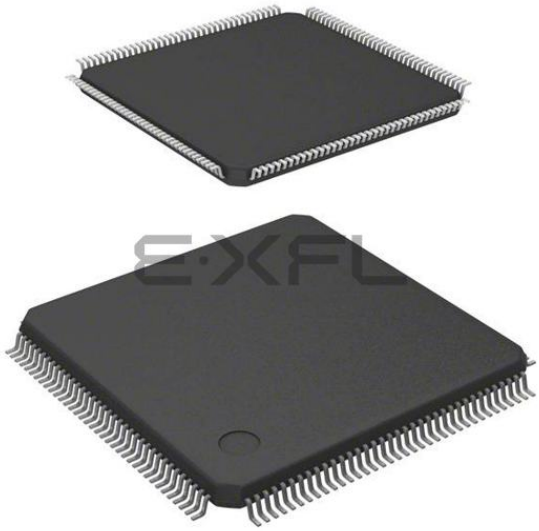


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST10
Core Size	16-Bit
Speed	40MHz
Connectivity	ASC, CANbus, EBI/EMI, I²C, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st10f272z2t3

Table 2. Pin description

Symbol	Pin	Type	Function		
P6.0 - P6.7	1 - 8	I/O	8-bit bidirectional I/O port, bit-wise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. Port 6 outputs can be configured as push-pull or open drain drivers. The input threshold of Port 6 is selectable (TTL or CMOS). The following Port 6 pins have alternate functions:		
	1	O	P6.0	$\overline{CS0}$	Chip select 0 output
	...	...	...	...	...
	5	O	P6.4	$\overline{CS4}$	Chip select 4 output
	6	I	P6.5	$\overline{HOLD}$	External master hold request input
		I/O		SCLK1	SSC1: master clock output / slave clock input
	7	O	P6.6	$\overline{HLDA}$	Hold acknowledge output
		I/O		MTSR1	SSC1: master-transmitter / slave-receiver O/I
	8	O	P6.7	$\overline{BREQ}$	Bus request output
		I/O		MRST1	SSC1: master-receiver / slave-transmitter I/O
P8.0 - P8.7	9-16	I/O	8-bit bidirectional I/O port, bit-wise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. Port 8 outputs can be configured as push-pull or open drain drivers. The input threshold of Port 8 is selectable (TTL or CMOS). The following Port 8 pins have alternate functions:		
	9	I/O	P8.0	CC16IO	CAPCOM2: CC16 capture input / compare output
		O		XPWM0	PWM1: channel 0 output
	...	...	...	...	...
	12	I/O	P8.3	CC19IO	CAPCOM2: CC19 capture input / compare output
		O		XPWM0	PWM1: channel 3 output
	13	I/O	P8.4	CC20IO	CAPCOM2: CC20 capture input / compare output
	14	I/O	P8.5	CC21IO	CAPCOM2: CC21 capture input / compare output
	15	I/O	P8.6	CC22IO	CAPCOM2: CC22 capture input / compare output
		I/O		RxD1	ASC1: Data input (Asynchronous) or I/O (Synchronous)
	16	I/O	P8.7	CC23IO	CAPCOM2: CC23 capture input / compare output
		O		TxD1	ASC1: Clock / Data output (Asynchronous/Synchronous)

5.4.11 Flash error register

Flash Error register, as well as all the other Flash registers, can be properly read only once LOCK bit of register FCR0L is low. Nevertheless, its content is updated when also BSY0 bit is reset as well; for this reason, it is definitively meaningful reading FER register content only when LOCK bit and BSY0 bit are cleared.

FER (0x8 0014h) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved							WPF	RESER	SEQER	reserved		10ER	PGER	ERER	ERR
							RC	RC	RC			RC	RC	RC	RC

Table 19. Flash error register

Bit	Function
ERR	Write Error This bit is automatically set when an error occurs during a Flash write operation or when a bad write operation setup is done. Once the error has been discovered and understood, ERR bit must be software reset.
ERER	Erase Error This bit is automatically set when an Erase error occurs during a Flash write operation. This error is due to a real failure of a Flash cell, that can no more be erased. This kind of error is fatal and the sector where it occurred must be discarded. This bit has to be software reset.
PGER	Program Error This bit is automatically set when a Program error occurs during a Flash write operation. This error is due to a real failure of a Flash cell, that can no more be programmed. The word where this error occurred must be discarded. This bit has to be software reset.
10ER	1 over 0 Error This bit is automatically set when trying to program at 1 bits previously set at 0 (this does not happen when programming the Protection bits). This error is not due to a failure of the Flash cell, but only flags that the desired data has not been written. This bit has to be software reset.
SEQER	Sequence Error This bit is automatically set when the control registers (FCR1H/L-FCR0H/L, FARH/L, FDR1H/L-FDR0H/L) are not correctly filled to execute a valid Write Operation. In this case no Write Operation is executed. This bit has to be software reset.
RESER	Resume Error This bit is automatically set when a suspended Program or Erase operation is not resumed correctly due to a protocol error. In this case the suspended operation is aborted. This bit has to be software reset.
WPF	Write Protection Flag This bit is automatically set when trying to program or erase in a sector write protected. In case of multiple Sector Erase, the not protected sectors are erased, while the protected sectors are not erased and bit WPF is set. This bit has to be software reset.

5.7 Write operation summary

In general, each write operation is started through a sequence of 3 steps:

1. The first instruction is used to select the desired operation by setting its corresponding selection bit in the Flash Control Register 0.
2. The second step is the definition of the Address and Data for programming or the Sectors or Banks to erase.
3. The last instruction is used to start the write operation, by setting the start bit WMS in the FCR0.

Once selected, but not yet started, one operation can be canceled by resetting the operation selection bit.

A summary of the available Flash Module Write Operations are shown in the following *Table 25*.

Table 25. Flash write operations

Operation	Select bit	Address and data	Start bit
Word Program (32-bit)	WPG	FARL/FARH FDR0L/FDR0H	WMS
Double Word Program (64-bit)	DWPG	FARL/FARH FDR0L/FDR0H FDR1L/FDR1H	WMS
Sector Erase	SER	FCR1L/FCR1H	WMS
Set Protection	SPR	FDR0L/FDR0H	WMS
Program/Erase Suspend	SUSP	None	None

7.2 Instruction set summary

The *Table 27* lists the instructions of the ST10F272Z2. The detailed description of each instruction can be found in the “ST10 Family Programming Manual”.

Table 27. Standard instruction set summary

Mnemonic	Description	Bytes
ADD(B)	Add word (byte) operands	2 / 4
ADDC(B)	Add word (byte) operands with Carry	2 / 4
SUB(B)	Subtract word (byte) operands	2 / 4
SUBC(B)	Subtract word (byte) operands with Carry	2 / 4
MUL(U)	(Un)Signed multiply direct GPR by direct GPR (16-16-bit)	2
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	Bit-wise AND, (word/byte operands)	2 / 4
OR(B)	Bit-wise OR, (word/byte operands)	2 / 4
XOR(B)	Bit-wise XOR, (word/byte operands)	2 / 4
BCLR	Clear direct bit	2
BSET	Set direct bit	2
BMOV(N)	Move (negated) direct bit to direct bit	4
BAND, BOR, BXOR	AND/OR/XOR direct bit with direct bit	4
BCMP	Compare direct bit to direct bit	4
BFLDH/L	Bit-wise modify masked high/low byte of bit-addressable direct word memory with immediate data	4
CMP(B)	Compare word (byte) operands	2 / 4
CMPD1/2	Compare word data to GPR and decrement GPR by 1/2	2 / 4
CMPI1/2	Compare word data to GPR and increment GPR by 1/2	2 / 4
PRIOR	Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR	2
SHL / SHR	Shift left/right direct word GPR	2
ROL / ROR	Rotate left/right direct word GPR	2
ASHR	Arithmetic (sign bit) shift right direct word GPR	2
MOV(B)	Move word (byte) data	2 / 4
MOVBS	Move byte operand to word operand with sign extension	2 / 4
MOVBZ	Move byte operand to word operand with zero extension	2 / 4
JMPA, JMPI, JMPR	Jump absolute/indirect/relative if condition is met	4
JMPS	Jump absolute to a code segment	4

7.3 MAC co-processor specific instructions

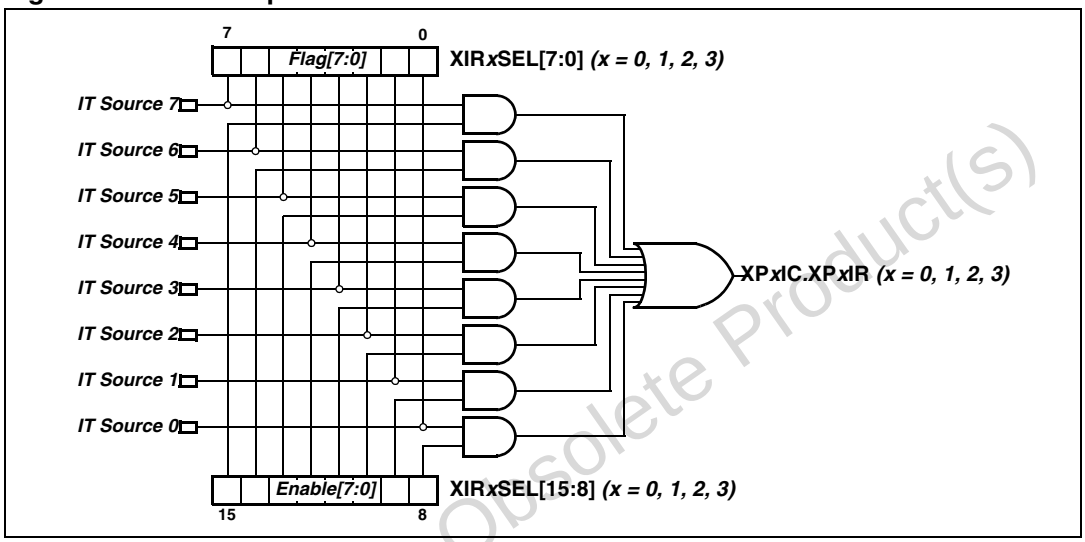
The *Table 28* lists the MAC instructions of the ST10F272Z2. The detailed description of each instruction can be found in the “ST10 Family Programming Manual”. Note that all MAC instructions are encoded on 4 Bytes.

Table 28. MAC instruction set summary

Mnemonic	Description
CoABS	Absolute Value of the Accumulator
CoADD(2)	Addition
CoASHR(rnd)	Accumulator Arithmetic Shift Right & Optional Round
CoCMP	Compare Accumulator with Operands
CoLOAD(-,2)	Load Accumulator with Operands
CoMAC(R,u,s,-,rnd)	(Un)Signed/(Un)Signed Multiply-Accumulate & Optional Round
CoMACM(R)(u,s,-,rnd)	(Un)Signed/(Un)Signed Multiply-Accumulate with Parallel Data Move & Optional Round
CoMAX / CoMIN	Maximum / Minimum of Operands and Accumulator
CoMOV	Memory to Memory Move
CoMUL(u,s,-,rnd)	(Un)Signed/(Un)Signed multiply & Optional Round
CoNEG(rnd)	Negate Accumulator & Optional Round
CoNOP	No-Operation
CoRND	Round Accumulator
CoSHL / CoSHR	Accumulator Logical Shift Left / Right
CoSTORE	Store a MAC Unit Register
CoSUB(2,R)	Substraction

available vector. If more than one source is enabled to issue the request, the service routine will have to take care to identify the real event to be serviced. This can easily be done by checking the flag bits (Byte Low of XIRxSEL register). Note that the flag bits can also provide information about events which are not currently serviced by the interrupt controller (since masked through the enable bits), allowing an effective software management also in absence of the possibility to serve the related interrupt request: a periodic polling of the flag bits may be implemented inside the user application.

Figure 9. X-Interrupt basic structure



The *Table 30* summarizes the mapping of the different interrupt sources which shares the four X-interrupt vectors.

Table 30. X-Interrupt detailed mapping

	XP0INT	XP1INT	XP2INT	XP3INT
CAN1 Interrupt	x			x
CAN2 Interrupt		x		x
I2C Receive	x	x	x	
I2C Transmit	x	x	x	
I2C Error				x
SSC1 Receive	x	x	x	
SSC1 Transmit	x	x	x	
SSC1 Error				x
ASC1 Receive	x	x	x	
ASC1 Transmit	x	x	x	
ASC1 Transmit Buffer	x	x	x	

Table 45 and *Table 46* list some possible Baud rates against the required reload values and the resulting bit times for 40 MHz and 64 MHz CPU clock respectively. The maximum is anyway limited to 8M Baud.

Table 45. Synchronous baud rate and reload values (f_{CPU} = 40 MHz)

Baud Rate	Bit Time	Reload Value
Reserved	---	0000h
Can be used only with f_{CPU} = 32 MHz (or lower)	---	0001h
6.6M Baud	150 ns	0002h
5M Baud	200 ns	0003h
2.5M Baud	400 ns	0007h
1M Baud	1 µs	0013h
100K Baud	10 µs	00C7h
10K Baud	100 µs	07CFh
1K Baud	1 ms	4E1Fh
306 Baud	3.26 ms	FF4Eh

Table 46. Synchronous baud rate and reload values (f_{CPU} = 64 MHz)

Baud Rate	Bit Time	Reload Value
Reserved	---	0000h
Can be used only with f_{CPU} = 32 MHz (or lower)	---	0001h
Can be used only with f_{CPU} = 48 MHz (or lower)	---	0002h
8M Baud	125 ns	0003h
4M Baud	250 ns	0007h
1M Baud	1 µs	001Fh
100K Baud	10 µs	013Fh
10K Baud	100 µs	0C7Fh
1K Baud	1 ms	7CFFh
489 Baud	2.04 ms	FF9Eh

23 Register set

This section summarizes all registers implemented in the ST10F272Z2, ordered by name.

23.1 Special function registers

The following table lists all SFRs which are implemented in the ST10F272Z2 in alphabetical order.

Bit-addressable SFRs are marked with the letter "**b**" in column "Name".

SFRs within the **Extended SFR-Space** (ESFRs) are marked with the letter "**E**" in column "Physical Address".

Table 53. List of special function registers

Name	Physical address	8-bit address	Description	Reset value
ADCIC **b**	FF98h	CCh	A/D converter end of conversion interrupt control register	- - 00h
ADCON **b**	FFA0h	D0h	A/D converter control register	0000h
ADDAT	FEA0h	50h	A/D converter result register	0000h
ADDAT2	F0A0h **E**	50h	A/D converter 2 result register	0000h
ADDRSEL1	FE18h	0Ch	Address select register 1	0000h
ADDRSEL2	FE1Ah	0Dh	Address select register 2	0000h
ADDRSEL3	FE1Ch	0Eh	Address select register 3	0000h
ADDRSEL4	FE1Eh	0Fh	Address select register 4	0000h
ADEIC **b**	FF9Ah	CDh	A/D converter overrun error interrupt control register	- - 00h
BUSCON0 **b**	FF0Ch	86h	Bus configuration register 0	0xx0h
BUSCON1 **b**	FF14h	8Ah	Bus configuration register 1	0000h
BUSCON2 **b**	FF16h	8Bh	Bus configuration register 2	0000h
BUSCON3 **b**	FF18h	8Ch	Bus configuration register 3	0000h
BUSCON4 **b**	FF1Ah	8Dh	Bus configuration register 4	0000h
CAPREL	FE4Ah	25h	GPT2 capture/reload register	0000h
CC0	FE80h	40h	CAPCOM register 0	0000h
CC0IC **b**	FF78h	BCh	CAPCOM register 0 interrupt control register	- - 00h
CC1	FE82h	41h	CAPCOM register 1	0000h
CC1IC **b**	FF7Ah	BDh	CAPCOM register 1 interrupt control register	- - 00h
CC2	FE84h	42h	CAPCOM register 2	0000h
CC2IC **b**	FF7Ch	BEh	CAPCOM register 2 interrupt control register	- - 00h
CC3	FE86h	43h	CAPCOM register 3	0000h
CC3IC **b**	FF7Eh	BFh	CAPCOM register 3 interrupt control register	- - 00h

Table 53. List of special function registers (continued)

Name		Physical address		8-bit address	Description	Reset value
CC4		FE88h		44h	CAPCOM register 4	0000h
CC4IC	b	FF80h		C0h	CAPCOM register 4 interrupt control register	- - 00h
CC5		FE8Ah		45h	CAPCOM register 5	0000h
CC5IC	b	FF82h		C1h	CAPCOM register 5 interrupt control register	- - 00h
CC6		FE8Ch		46h	CAPCOM register 6	0000h
CC6IC	b	FF84h		C2h	CAPCOM register 6 interrupt control register	- - 00h
CC7		FE8Eh		47h	CAPCOM register 7	0000h
CC7IC	b	FF86h		C3h	CAPCOM register 7 interrupt control register	- - 00h
CC8		FE90h		48h	CAPCOM register 8	0000h
CC8IC	b	FF88h		C4h	CAPCOM register 8 interrupt control register	- - 00h
CC9		FE92h		49h	CAPCOM register 9	0000h
CC9IC	b	FF8Ah		C5h	CAPCOM register 9 interrupt control register	- - 00h
CC10		FE94h		4Ah	CAPCOM register 10	0000h
CC10IC	b	FF8Ch		C6h	CAPCOM register 10 interrupt control register	- - 00h
CC11		FE96h		4Bh	CAPCOM register 11	0000h
CC11IC	b	FF8Eh		C7h	CAPCOM register 11 interrupt control register	- - 00h
CC12		FE98h		4Ch	CAPCOM register 12	0000h
CC12IC	b	FF90h		C8h	CAPCOM register 12 interrupt control register	- - 00h
CC13		FE9Ah		4Dh	CAPCOM register 13	0000h
CC13IC	b	FF92h		C9h	CAPCOM register 13 interrupt control register	- - 00h
CC14		FE9Ch		4Eh	CAPCOM register 14	0000h
CC14IC	b	FF94h		CAh	CAPCOM register 14 interrupt control register	- - 00h
CC15		FE9Eh		4Fh	CAPCOM register 15	0000h
CC15IC	b	FF96h		CBh	CAPCOM register 15 interrupt control register	- - 00h
CC16		FE60h		30h	CAPCOM register 16	0000h
CC16IC	b	F160h	E	B0h	CAPCOM register 16 interrupt control register	- - 00h
CC17		FE62h		31h	CAPCOM register 17	0000h
CC17IC	b	F162h	E	B1h	CAPCOM register 17 interrupt control register	- - 00h
CC18		FE64h		32h	CAPCOM register 18	0000h
CC18IC	b	F164h	E	B2h	CAPCOM register 18 interrupt control register	- - 00h
CC19		FE66h		33h	CAPCOM register 19	0000h
CC19IC	b	F166h	E	B3h	CAPCOM register 19 interrupt control register	- - 00h
CC20		FE68h		34h	CAPCOM register 20	0000h
CC20IC	b	F168h	E	B4h	CAPCOM register 20 interrupt control register	- - 00h

Table 53. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
XPERCON **b**	F024h **E**	12h	XPER configuration register	- - 05h
ZEROS **b**	FF1Ch	8Eh	Constant value 0's register (read only)	0000h

Note: 1. The system configuration is selected during reset. SYSCON reset value is 0000 0xx0 x000 0000b.

2. Reset Value depends on different triggered reset event.

3. The XPnIC Interrupt Control Registers control interrupt requests from integrated X-Bus peripherals. Some software controlled interrupt requests may be generated by setting the XPnIR bits (of XPnIC register) of the unused X-Peripheral nodes.

23.2 X-registers

The following table lists all X-Bus registers which are implemented in the ST10F272Z2 ordered by their name. The FLASH control registers are listed in a separate section, in spite of they also are physically mapped on X-Bus memory space.

Note: The X-Registers are not bit-addressable.

Table 54. List of XBus registers

Name	Physical address	Description	Reset value
CAN1BRPER	EF0Ch	CAN1: BRP extension register	0000h
CAN1BTR	EF06h	CAN1: Bit timing register	2301h
CAN1CR	EF00h	CAN1: CAN control register	0001h
CAN1EC	EF04h	CAN1: error counter	0000h
CAN1IF1A1	EF18h	CAN1: IF1 arbitration 1	0000h
CAN1IF1A2	EF1Ah	CAN1: IF1 arbitration 2	0000h
CAN1IF1CM	EF12h	CAN1: IF1 command mask	0000h
CAN1IF1CR	EF10h	CAN1: IF1 command request	0001h
CAN1IF1DA1	EF1Eh	CAN1: IF1 data A 1	0000h
CAN1IF1DA2	EF20h	CAN1: IF1 data A 2	0000h
CAN1IF1DB1	EF22h	CAN1: IF1 data B 1	0000h
CAN1IF1DB2	EF24h	CAN1: IF1 data B 2	0000h
CAN1IF1M1	EF14h	CAN1: IF1 mask 1	FFFFh
CAN1IF1M2	EF16h	CAN1: IF1 mask 2	FFFFh
CAN1IF1MC	EF1Ch	CAN1: IF1 message control	0000h
CAN1IF2A1	EF48h	CAN1: IF2 arbitration 1	0000h
CAN1IF2A2	EF4Ah	CAN1: IF2 arbitration 2	0000h
CAN1IF2CM	EF42h	CAN1: IF2 command mask	0000h

23.3 Flash registers ordered by name

The following table lists all Flash Control Registers which are implemented in the ST10F272Z2 ordered by their name. These registers are physically mapped on the IBus, except for XFVTAUR0, which is mapped on XBus. Note that these registers are not bit-addressable.

Table 55. List of Flash registers

Name	Physical address	Description	Reset value
FARH	0x0008 0012	Flash address register - high	0000h
FARL	0x0008 0010	Flash address register - low	0000h
FCR0H	0x0008 0002	Flash control register 0 - high	0000h
FCR0L	0x0008 0000	Flash control register 0 - low	0000h
FCR1H	0x0008 0006	Flash control register 1 - high	0000h
FCR1L	0x0008 0004	Flash control register 1 - low	0000h
FDR0H	0x0008 000A	Flash data register 0 - high	FFFFh
FDR0L	0x0008 0008	Flash data register 0 - low	FFFFh
FDR1H	0x0008 000E	Flash data register 1 - high	FFFFh
FDR1L	0x0008 000C	Flash data register 1 - low	FFFFh
FER	0x0008 0014	Flash error register	0000h
FNVAPR0	0x0008 DFB8	Flash non-volatile access protection reg.0	ACFFh
FNVAPR1H	0x0008 DFBE	Flash non-volatile access protection reg.1 - high	FFFFh
FNVAPR1L	0x0008 DFBC	Flash non-volatile access protection reg.1 - low	FFFFh
FNVWPIR	0x0008 DFB0	Flash non-volatile protection I register	FFFFh
XFVTAUR0	0x0000 EB50	XBus Flash volatile temporary access unprotection register 0	0000h

23.4 Identification registers

The ST10F272Z2 have four Identification registers, mapped in ESFR space. These registers contain:

- A manufacturer identifier
- A chip identifier with its revision
- A internal Flash and size identifier
- Programming voltage description

not know that a new converted value is ready to be read in ADDAT2 register. Therefore at the following injection request the ADC fills the temporary register again (without generating any ADEINT interrupt request) and then the ADC is stalled for any further conversion. The ADC stays in the “wait for read ADDAT2 register” condition forever.

Figure 37. ADC injection theoretical operation

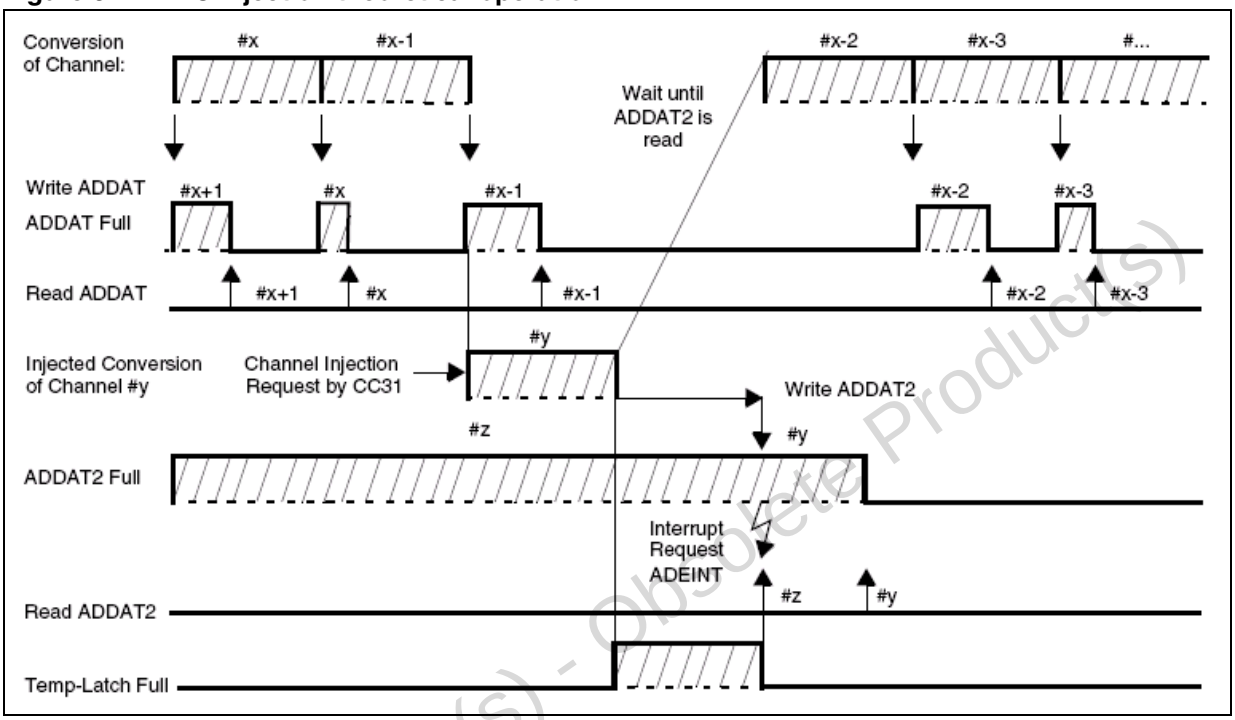
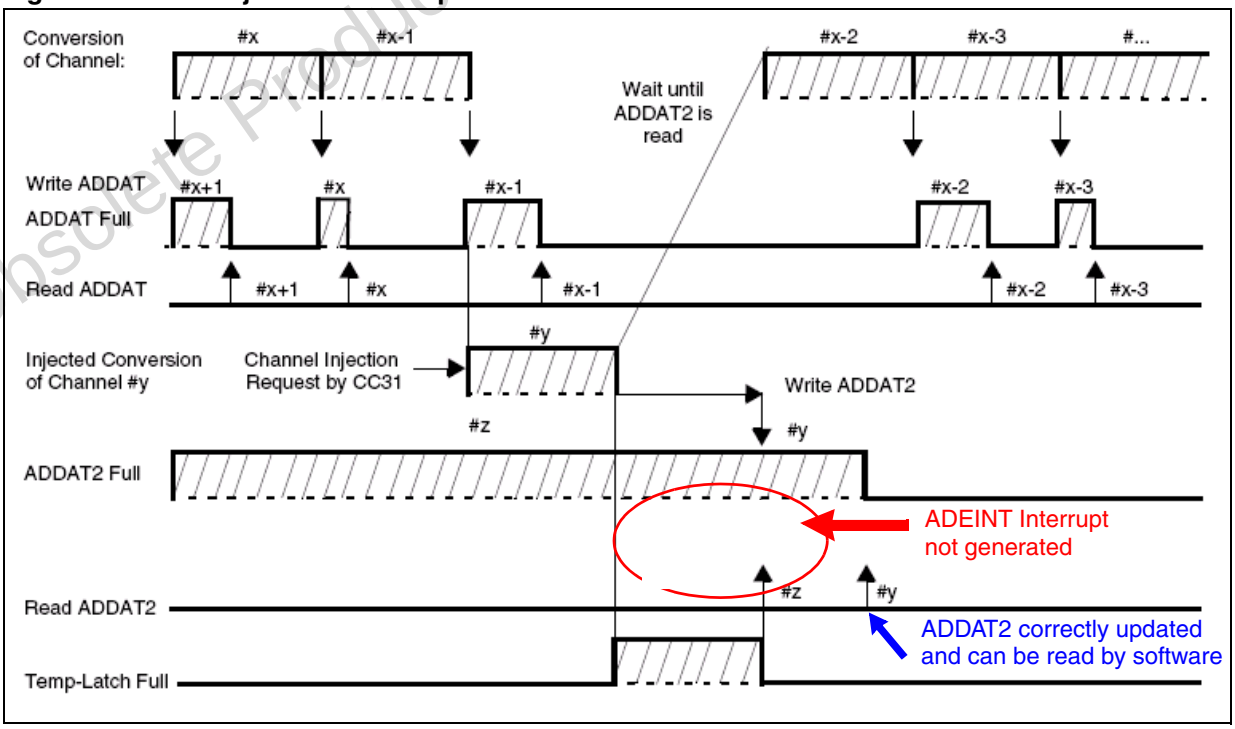


Figure 38. ADC injection actual operation



3) Load the Tx timer with the CCy register minus 1:

```
MOV TxREL, #value
MOV Tx, #( value-1)
bfldl/bfldh TxxCON , #mask, #data or MOV TxxCON, #data
```

25.2 Recommended operating conditions

Table 62. Recommended operating conditions

Symbol	Parameter	Value		Unit
		Min	Max	
V_{DD}	Operating supply voltage	4.5	5.5	V
V_{STBY}	Operating stand-by supply voltage (1)	4.5	5.5	V
V_{AREF}	Operating analog reference voltage (2)			
T_A	Ambient temperature under bias	-40	+125	°C
T_J	Junction temperature under bias	-40	+150	°C

1. The value of the V_{STBY} voltage is specified in the range 4.5 - 5.5 Volt. Nevertheless, it is acceptable to exceed the upper limit (up to 6.0 Volt) for a maximum of 100 hours over the global 300000 hours, representing the lifetime of the device (about 30 years). On the other hand, it is possible to exceed the lower limit (down to 4.0 Volt) whenever RTC and 32kHz on-chip oscillator amplifier are turned off (only Stand-by RAM powered through VSTBY pin in Stand-by mode). When VSTBY voltage is lower than main VDD, the input section of VSTBY/EA pin can generate a spurious static consumption on VDD power supply (in the range of tenth of µA).
2. For details on operating conditions concerning the usage of A/D Converter refer to *Section 25.7*.

25.3 Power considerations

The average chip-junction temperature, T_J, in degrees Celsius, may be calculated using the following equation:

$T_J = T_A + (P_D \times \Theta_{JA})$ (1)

Where:

T_A is the Ambient Temperature in °C,

Θ_{JA} is the Package Junction-to-Ambient Thermal Resistance, in °C/W,

P_D is the sum of P_{INT} and $P_{I/O}$ ($P_D = P_{INT} + P_{I/O}$),

P_{INT} is the product of I_{DD} and V_{DD}, expressed in Watt. This is the Chip Internal Power,

$P_{I/O}$ represents the Power Dissipation on Input and Output Pins; User Determined.

Most of the time for the applications $P_{I/O} < P_{INT}$ and may be neglected. On the other hand, $P_{I/O}$ may be significant if the device is configured to drive continuously external modules and/or memories.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is given by:

$P_D = K / (T_J + 273°C)$ (2)

Therefore (solving equations 1 and 2):

$K = P_D \times (T_A + 273°C) + \Theta_{JA} \times P_D^2$ (3)

Where:

K is a constant for the particular part, which may be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J may be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Jitter in the input clock

PLL acts like a low pass filter for any jitter in the input clock. Input Clock jitter with the frequencies within the PLL loop bandwidth is passed to the PLL output and higher frequency jitter (frequency > PLL bandwidth) is attenuated @20dB/decade.

Noise in the PLL loop

This contribution again can be caused by the following sources:

- Device noise of the circuit in the PLL
- Noise in supply and substrate.

Device noise of the circuit in the PLL

The long term jitter is inversely proportional to the bandwidth of the PLL: the wider is the loop bandwidth, the lower is the jitter due to noise in the loop. Besides, the long term jitter is practically independent on the multiplication factor.

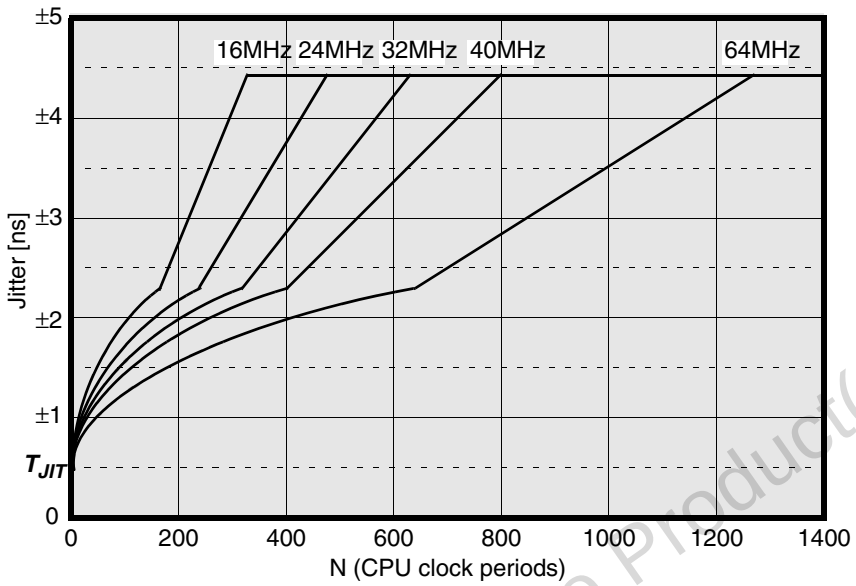
The most noise sensitive circuit in the PLL circuit is definitively the VCO (Voltage Controlled Oscillator). There are two main sources of noise: thermal (random noise, frequency independent so practically white noise) and flicker (low frequency noise, 1/f). For the frequency characteristics of the VCO circuitry, the effect of the thermal noise results in a $1/f^2$ region in the output noise spectrum, while the flicker noise in a $1/f^3$. Assuming a noiseless PLL input and supposing that the VCO is dominated by its $1/f^2$ noise, the R.M.S. value of the accumulated jitter is *proportional to the square root of N*, where N is the number of clock periods within the considered time interval.
On the contrary, assuming again a noiseless PLL input and supposing that the VCO is dominated by its $1/f^3$ noise, the R.M.S. value of the accumulated jitter is *proportional to N*, where N is the number of clock periods within the considered time interval.

The jitter in the PLL loop can be modelized as dominated by the $i1/f^2$ noise for N smaller than a certain value depending on the PLL output frequency and on the bandwidth characteristics of loop. Above this first value, the jitter becomes dominated by the $i1/f^3$ noise component. Lastly, for N greater than a second value of N, a saturation effect is evident, so the jitter does not grow anymore when considering a longer time interval (jitter stable increasing the number of clock periods N). The PLL loop acts as a high pass filter for any noise in the loop, with cutoff frequency equal to the bandwidth of the PLL. The saturation value corresponds to what has been called self referred long term jitter of the PLL. In *Figure 49* the maximum jitter trend versus the number of clock periods N (for some typical CPU frequencies) is reported: the curves represent the very worst case, computed taking into account all corners of temperature, power supply and process variations: the real jitter is always measured well below the given worst case values.

Noise in supply and substrate

Digital supply noise adds deterministic components to the PLL output jitter, independent on multiplication factor. Its effects is strongly reduced thanks to particular care used in the physical implementation and integration of the PLL module inside the device. Anyhow, the contribution of the digital noise to the global jitter is widely taken into account in the curves provided in *Figure 49*.

Figure 49. ST10F272Z2 PLL jitter



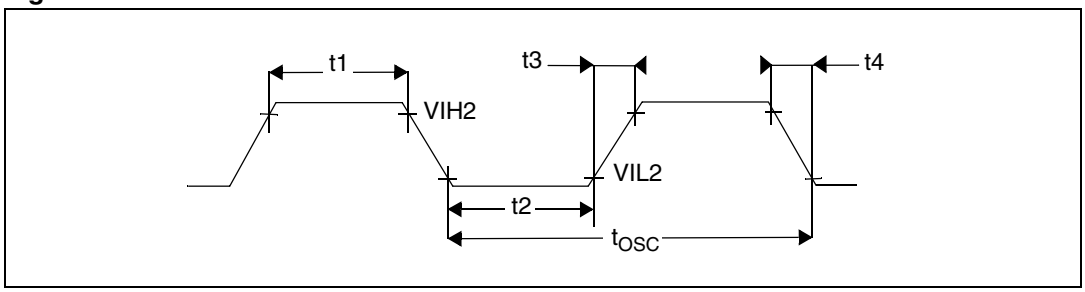
25.8.10 PLL lock / unlock

During normal operation, if the PLL gets unlocked for any reason, an interrupt request to the CPU is generated, and the reference clock (oscillator) is automatically disconnected from the PLL input: in this way, the PLL goes into free-running mode, providing the system with a backup clock signal (free running frequency F_{free}). This feature allows to recover from a crystal failure occurrence without risking to go in an undefined configuration: the system is provided with a clock allowing the execution of the PLL unlock interrupt routine in a safe mode.

The path between reference clock and PLL input can be restored only by a hardware reset, or by a bidirectional software or watchdog reset event that forces the $\overline{\text{RSTIN}}$ pin low.

Note: The external RC circuit on $\overline{RSTIN}$ pin shall be properly sized in order to extend the duration of the low pulse to grant the PLL gets locked before the level at $\overline{RSTIN}$ pin is recognized high: bidirectional reset internally drives $\overline{RSTIN}$ pin low for just 1024 TCL (definitively not sufficient to get the PLL locked starting from free-running mode).

Figure 52. External clock drive XTAL1



Note: When Direct Drive is selected, an external clock source can be used to drive XTAL1. The maximum frequency of the external clock source depends on the duty cycle: when 64MHz is used, 50% duty cycle shall be granted (low phase = high phase = 7.8 ns); when for instance 32 MHz is used, a 25% duty cycle can be accepted (minimum phase, high or low, again equal to 7.8ns).

25.8.14 Memory cycle variables

The tables below use three variables which are derived from the BUSCONx registers and represent the special characteristics of the programmed memory cycle. The following table describes, how these variables are to be computed.

Table 78. Memory cycle variables

Description	Symbol	Values
ALE Extension	t_A	TCL x [ALECTL]
Memory Cycle Time wait states	t_C	2TCL x (15 - [MCTC])
Memory Tri-state Time	t_F	2TCL x (1 - [MTTC])

25.8.15 External memory bus timing

The following sections include the External Memory Bus timings. The given values are computed for a maximum CPU clock of 40 MHz.

Obviously, when higher CPU clock frequency is used (up to 64 MHz), some numbers in the timing formulas become zero or negative which, in most cases is not acceptable or not meaningless at all. In these cases, it is necessary to relax the speed of the bus setting properly t_A, t_C and t_F.

Note: All External Memory Bus Timings and SSC Timings reported in the following tables are granted by Design Characterization and not fully tested in production.

25.8.20 High-speed synchronous serial interface (SSC) timing

25.8.20.1 Master mode

V_{DD} = 5 V ±10%, V_{SS} = 0 V, T_A = -40 to +125 °C, C_L = 50 pF

Table 83. SSC master mode timings

Symbol		Parameter	Max. Baudrate 6.6MBd (1)@F_{CPU} = 40MHz (<SSCBR> = 0002h)		Variable Baudrate (<SSCBR> = 0001h - FFFFh)		Unit
			min.	max.	min.	max.	
t_{300}	CC	SSC clock cycle time(2))	150	150	8TCL	262144 TCL	ns
t_{301}	CC	SSC clock high time	63	–	t_{300} / 2 – 12	–	ns
t_{302}	CC	SSC clock low time	63	–	t_{300} / 2 – 12	–	ns
t_{303}	CC	SSC clock rise time	–	10	–	10	ns
t_{304}	CC	SSC clock fall time	–	10	–	10	ns
t_{305}	CC	Write data valid after shift edge	–	15	–	15	ns
t_{306}	CC	Write data hold after shift edge(3)	– 2	–	– 2	–	ns
t_{307p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	37.5	–	2TCL + 12.5	–	ns
t_{308p}	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	50	–	4TCL	–	ns
t_{307}	SR	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	25	–	2TCL	–	ns
t_{308}	SR	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	0	–	0	–	ns

1. Maximum Baudrate is in reality 8Mbaud, that can be reached with 64MHz CPU clock and <SSCBR> set to '3h', or with 48MHz CPU clock and <SSCBR> set to '2h'. When 40MHz CPU clock is used the maximum baudrate cannot be higher than 6.6Mbaud (<SSCBR> = '2h') due to the limited granularity of <SSCBR>. Value '1h' for <SSCBR> can be used only with CPU clock equal to (or lower than) 32MHz.
2. Formula for SSC Clock Cycle time: t_{300} = 4 TCL x (<SSCBR> + 1) Where <SSCBR> represents the content of the SSC Baudrate register, taken as unsigned 16-bit integer. Minimum limit allowed for t_{300} is 125ns (corresponding to 8Mbaud).
3. Partially tested, guaranteed by design characterization.