



Welcome to [E-XFL.COM](https://www.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

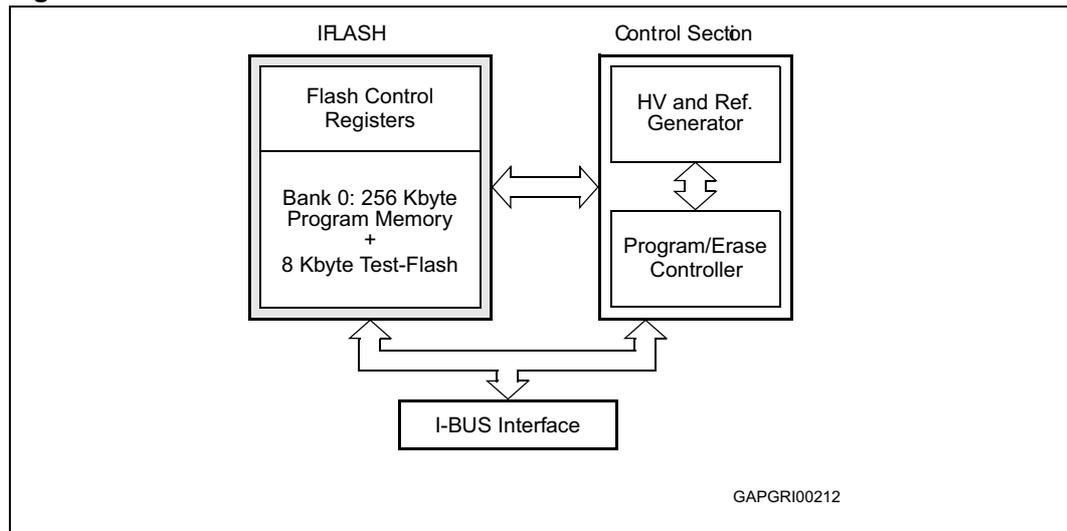
Product Status	Obsolete
Core Processor	ST10
Core Size	16-Bit
Speed	64MHz
Connectivity	ASC, CANbus, EBI/EMI, I ² C, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-BQFP
Supplier Device Package	144-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/f272-bag-p-tr

5 Internal Flash memory

5.1 Overview

The on-chip Flash is composed by one matrix module, 256 KBytes wide.
 This module is on ST10 Internal bus, so it is called IFLASH

Figure 5. Flash structure



The programming operations of the flash are managed by an embedded Flash Program/Erase Controller (FPEC). The High Voltages needed for Program/Erase operations are internally generated.

The Data bus is 32-bit wide for fetch accesses to IFLASH, while it is 16 bit wide for read accesses to IFLASH. Read/write accesses to IFLASH Control Registers area are 16 bit wide.

5.2 Functional description

5.2.1 Structure

[Table 3](#) shows the Address space reserved to the Flash module.

Table 3. Address space reserved to the Flash module

Description	Addresses	Size
IFLASH sectors	0x00 0000 to 0x04 FFFF	256 Kbyte
Reserved IBUS area	0x05 0000 to 0x07 FFFF	192 Kbyte
Registers and Flash internal reserved area	0x08 0000 to 0x08 FFFF	64 Kbyte

5.2.2 Modules structure

The IFLASH module is composed by a bank (Bank 0) of 256 Kbyte of Program Memory divided in 8 sectors (B0F0...B0F7). Bank 0 contains also a reserved sector named Test-Flash. The Addresses from 0x08 0000 to 0x08 FFFF are reserved for the Control Register Interface and other internal service memory space used by the Flash Program/Erase controller.

The following tables show the memory mapping of the Flash when it is accessed in read mode ([Table 4: Flash modules sectorization \(Read operations\)](#)), and when accessed in write or erase mode ([Table 5: Flash modules sectorization \(Write operations or with ROMS1='1' or BootStrap mode\)](#)): note that with this second mapping, the first four banks are remapped into code segment 1 (same as obtained setting bit ROMS1 in SYSCON register).

Table 4. Flash modules sectorization (Read operations)

Bank	Description	Addresses	Size	ST10 Bus size
B0	Bank 0 Flash 0 (B0F0)	0x0000 0000 - 0x0000 1FFF	8 KB	32-bit (I-BUS)
	Bank 0 Flash 1 (B0F1)	0x0000 2000 - 0x0000 3FFF	8 KB	
	Bank 0 Flash 2 (B0F2)	0x0000 4000 - 0x0000 5FFF	8 KB	
	Bank 0 Flash 3 (B0F3)	0x0000 6000 - 0x0000 7FFF	8 KB	
	Bank 0 Flash 4 (B0F4)	0x0001 8000 - 0x0001 FFFF	32 KB	
	Bank 0 Flash 5 (B0F5)	0x0002 0000 - 0x0002 FFFF	64 KB	
	Bank 0 Flash 6 (B0F6)	0x0003 0000 - 0x0003 FFFF	64 KB	
	Bank 0 Flash 7 (B0F7)	0x0004 0000 - 0x0004 FFFF	64 KB	

Table 5. Flash modules sectorization (Write operations or with ROMS1='1' or BootStrap mode)

Bank	Description	Addresses	Size	ST10 Bus size
B0	Bank 0 Test-Flash (B0TF)	0x0000 0000 - 0x0000 1FFF	8 KB	32-bit (I-BUS)
	Bank 0 Flash 0 (B0F0)	0x0001 0000 - 0x0001 1FFF	8 KB	
	Bank 0 Flash 1 (B0F1)	0x0001 2000 - 0x0001 3FFF	8 KB	
	Bank 0 Flash 2 (B0F2)	0x0001 4000 - 0x0001 5FFF	8 KB	
	Bank 0 Flash 3 (B0F3)	0x0001 6000 - 0x0001 7FFF	8 KB	
	Bank 0 Flash 4 (B0F4)	0x0001 8000 - 0x0001 FFFF	32 KB	
	Bank 0 Flash 5 (B0F5)	0x0002 0000 - 0x0002 FFFF	64 KB	
	Bank 0 Flash 6 (B0F6)	0x0003 0000 - 0x0003 FFFF	64 KB	
	Bank 0 Flash 7 (B0F7)	0x0004 0000 - 0x0004 FFFF	64 KB	

[Table 5](#) above refers to the configuration when bit ROMS1 of SYSCON register is set.

5.3 Write operation

The Flash module have one single register interface mapped in the memory space of the IBUS (0x08 0000 to 0x08 0015). All the operations are enabled through four 16-bit control registers: Flash Control Register 1-0 High/Low (FCR1H/L-FCR0H/L). Eight other 16-bit registers are used to store Flash Address and Data for Program operations (FARH/L and FDR1H/L-FDR0H/L) and Write Operation Error flags (FERH/L). All registers are accessible with 8 and 16-bit instructions (since operates in 16-bit mode when in read/ write).

Before accessing the IFlash module (and consequently also the Flash register to be used for program/erasing operations), bit ROMEN in SYSCON register shall be set.

During a Flash write operation any attempt to read the flash itself, that is under modification, will output invalid data (software trap 009Bh). This means that the Flash is not fetchable when a programming operation is active: the write operation commands must be executed from another memory (internal RAM or external memory), as in ST10F269 device. In fact, due to IBUS characteristics, it is not possible to perform a write operation on IFLASH, when fetching code from IFLASH.

Direct addressing is not allowed for write accesses to IFLASH Control Registers.

During a Write operation, when bit LOCK of FCR0 is set, it is forbidden to write into the Flash Control Registers.

Power supply drop

If during a write operation the internal low voltage supply drops below a certain internal voltage threshold, any write operation running is suddenly interrupted and the module is reset to Read mode. At following Power-on, the interrupted Flash write operation must be repeated.

7.2 Instruction set summary

The [Table 27](#) lists the instructions of the ST10F272. 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
CMP11/2	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

14 A/D converter

A 10-bit A/D converter with 16+8 multiplexed input channels and a sample and hold circuit is integrated on-chip. An automatic self-calibration adjusts the A/D converter module to process parameter variations at each reset event. The sample time (for loading the capacitors) and the conversion time is programmable and can be adjusted to the external circuitry.

The ST10F272E has 16+8 multiplexed input channels on Port 5 and Port 1. The selection between Port 5 and Port 1 is made via a bit in a XBus register. Refer to the User Manual for a detailed description.

A different accuracy is guaranteed (Total Unadjusted Error) on Port 5 and Port 1 analog channels (with higher restrictions when overload conditions occur); in particular, Port 5 channels are more accurate than the Port 1 ones. Refer to [Section 24: Electrical characteristics](#).

The A/D converter input bandwidth is limited by the achievable accuracy: supposing a maximum error of 0.5LSB (2mV) impacting the global TUE (TUE depends also on other causes), in worst case of temperature and process, the maximum frequency for a sine wave analog signal is around 7.5 kHz. Of course, to reduce the effect of the input signal variation on the accuracy down to 0.05LSB, the maximum input frequency of the sine wave shall be reduced to 800 Hz.

If static signal is applied during sampling phase, series resistance shall not be greater than 20k Ω (this taking into account eventual input leakage). It is suggested to not connect any capacitance on analog input pins, in order to reduce the effect of charge partitioning (and consequent voltage drop error) between the external and the internal capacitance: in case an RC filter is necessary the external capacitance must be greater than 10nF to minimize the accuracy impact.

Overrun error detection / protection is controlled by the ADDAT register. Either an interrupt request is generated when the result of a previous conversion has not been read from the result register at the time the next conversion is complete, or the next conversion is suspended until the previous result has been read. For applications which require less than 16+8 analog input channels, the remaining channel inputs can be used as digital input port pins.

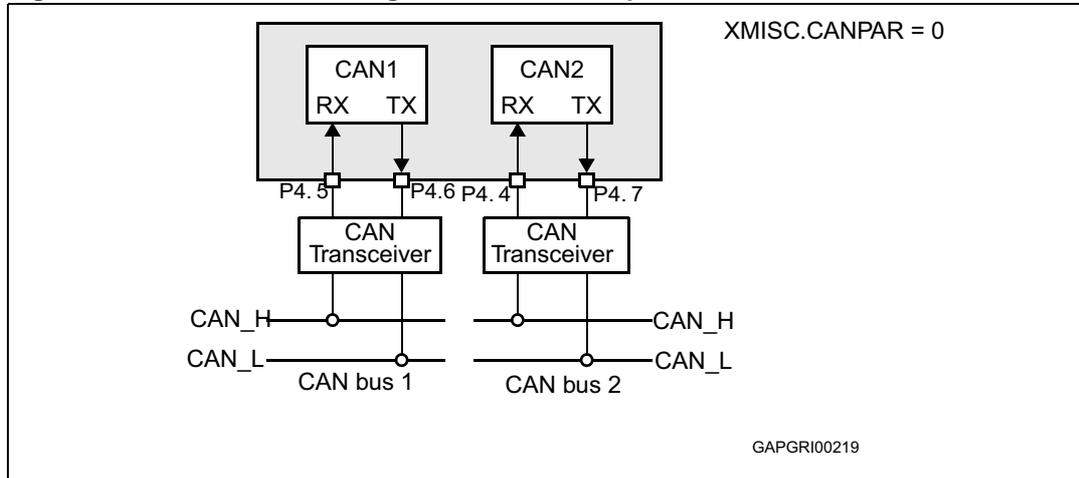
The A/D converter of the ST10F272 supports different conversion modes:

- **Single channel single conversion:** The analog level of the selected channel is sampled once and converted. The result of the conversion is stored in the ADDAT register.
- **Single channel continuous conversion:** The analog level of the selected channel is repeatedly sampled and converted. The result of the conversion is stored in the ADDAT register.
- **Auto scan single conversion:** The analog level of the selected channels are sampled once and converted. After each conversion the result is stored in the ADDAT register. The data can be transferred to the RAM by interrupt software management or using the powerful Peripheral Event Controller (PEC) data transfer.
- **Auto scan continuous conversion:** The analog level of the selected channels are repeatedly sampled and converted. The result of the conversion is stored in the ADDAT

Single CAN bus

The single CAN Bus multiple interfaces configuration may be implemented using two CAN transceivers as shown in [Figure 12](#).

Figure 12. Connection to single CAN bus via separate CAN transceivers



The ST10F272 also supports single CAN Bus multiple (dual) interfaces using the open drain option of the CANx_TxD output as shown in [Figure 13](#). Thanks to the OR-Wired Connection, only one transceiver is required. In this case the design of the application must take in account the wire length and the noise environment.

Figure 13. Connection to single CAN bus via common CAN transceivers

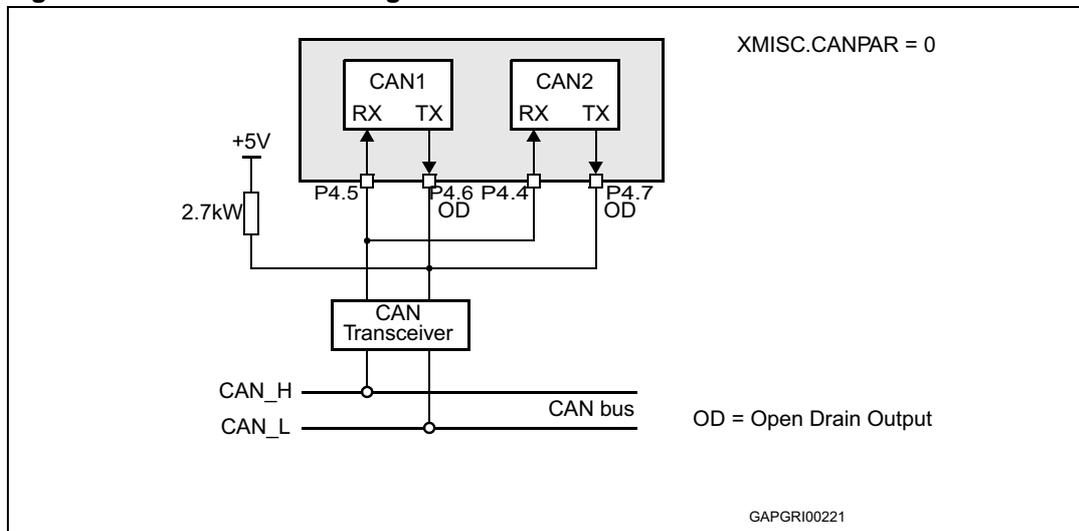
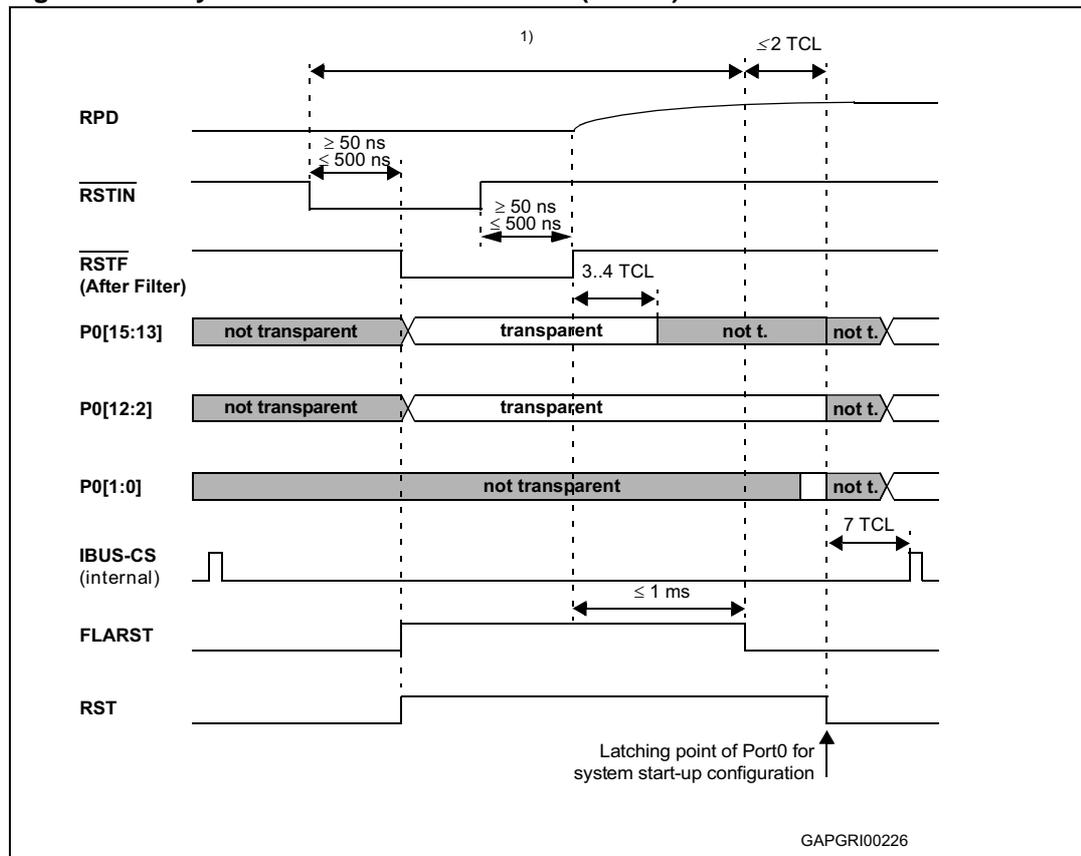
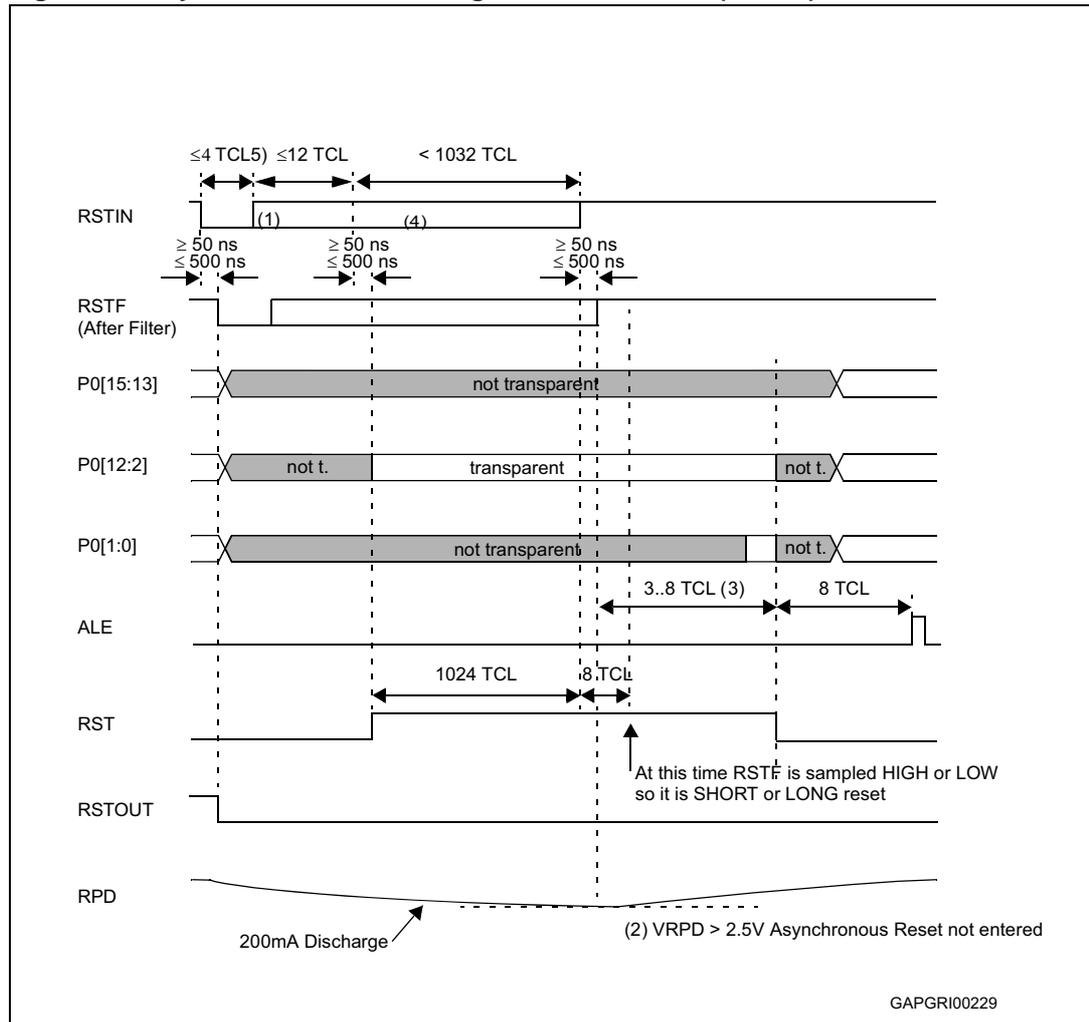


Figure 18. Asynchronous hardware RESET ($\overline{EA} = 1$)



1. Longer than Port0 settling time + PLL synchronization (if needed, that is P0 (15:13) changed).
2. Longer than 500ns to take into account of Input Filter on RSTIN pin.

Figure 21. Synchronous short / long hardware RESET ($\overline{EA} = 0$)



1. RSTIN assertion can be released there. Refer also to [Section 21.1](#) for details on minimum pulse duration.
2. If during the reset condition (RSTIN low), RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation), the asynchronous reset is then immediately entered.
3. 3 to 8 TCL depending on clock source selection.
4. RSTIN pin is pulled low if bit BDRSTEN (bit 3 of SYSCON register) was previously set by software. Bit BDRSTEN is cleared after reset.
5. Minimum RSTIN low pulse duration shall also be longer than 500ns to guarantee the pulse is not masked by the internal filter (refer to [Section 21.1](#)).

or Watchdog Reset become a Short Hardware Reset. On the contrary, if $\overline{\text{RSTF}}$ remains low for less than 4 TCL, the device simply exits reset state.

The Bidirectional reset is not effective in case RPD is held low, when a Software or Watchdog reset event occurs. On the contrary, if a Software or Watchdog Bidirectional reset event is active and RPD becomes low, the $\overline{\text{RSTIN}}$ pin is immediately released, while the internal reset sequence is completed regardless of RPD status change (1024 TCL).

Note: The bidirectional reset function is disabled by any reset sequence (bit BDRSTEN of SYSCON is cleared). To be activated again it must be enabled during the initialization routine.

WDTCN flags

Similarly to what already highlighted in the previous section when discussing about Short reset and the degeneration into Long reset, similar situations may occur when Bidirectional reset is enabled. The presence of the internal filter on $\overline{\text{RSTIN}}$ pin introduces a delay: when $\overline{\text{RSTIN}}$ is released, the internal signal after the filter (see $\overline{\text{RSTF}}$ in the drawings) is delayed, so it remains still active (low) for a while. It means that depending on the internal clock speed, a short reset may be recognized as a long reset: the WDTCN flags are set accordingly.

Besides, when either Software or Watchdog bidirectional reset events occur, again when the $\overline{\text{RSTIN}}$ pin is released (at the end of the internal reset sequence), the $\overline{\text{RSTF}}$ internal signal (after the filter) remains low for a while, and depending on the clock frequency it is recognized high or low: 8TCL after the completion of the internal sequence, the level of $\overline{\text{RSTF}}$ signal is sampled, and if recognized still low a Hardware reset sequence starts, and WDTCN will flag this last event, masking the previous one (Software or Watchdog reset). Typically, a Short Hardware reset is recognized, unless the $\overline{\text{RSTIN}}$ pin (and consequently internal signal $\overline{\text{RSTF}}$) is sufficiently held low by the external hardware to inject a Long Hardware reset. After this occurrence, the initialization routine is not able to recognize a Software or Watchdog bidirectional reset event, since a different source is flagged inside WDTCN register. This phenomenon does not occur when internal FLASH is selected during reset ($\text{EA} = 1$), since the initialization of the FLASH itself extend the internal reset duration well beyond the filter delay.

Figure 26, Figure 27 and Figure 28 summarize the timing for Software and Watchdog Timer Bidirectional reset events: in particular Figure 28 shows the degeneration into Hardware reset.

Figure 34. PORT0 bits latched into the different registers after reset

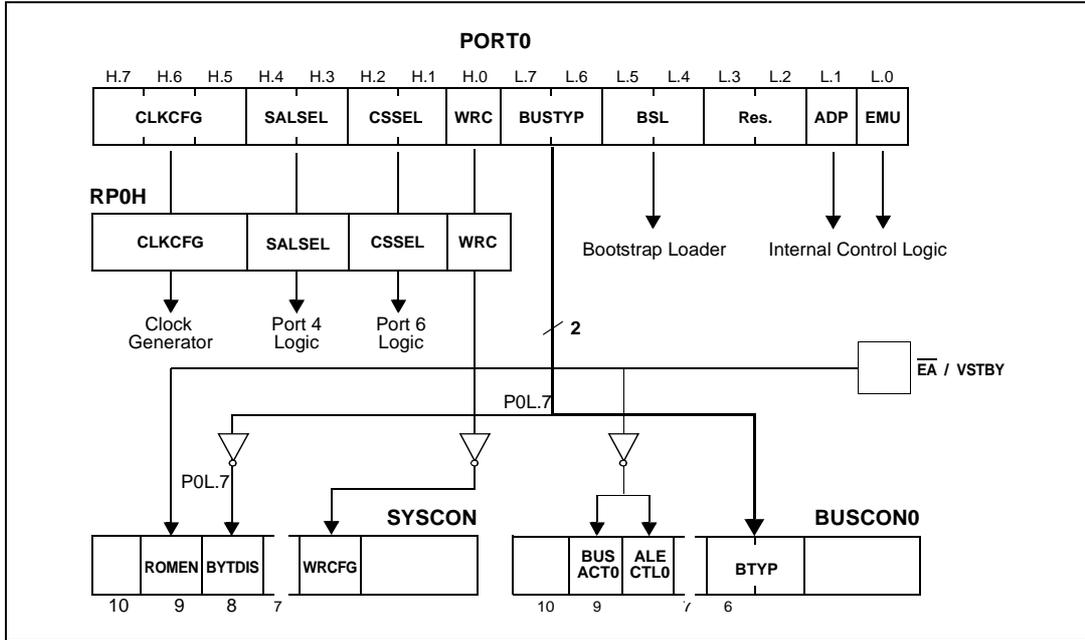


Table 53. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
T1	FE52h	29h	CAPCOM timer 1 register	0000h
T1ICb	FF9Eh	CFh	CAPCOM timer 1 interrupt control register	--00h
T1REL	FE56h	2Bh	CAPCOM timer 1 reload register	0000h
T2	FE40h	20h	GPT1 timer 2 register	0000h
T2CONb	FF40h	A0h	GPT1 timer 2 control register	0000h
T2ICb	FF60h	B0h	GPT1 timer 2 interrupt control register	--00h
T3	FE42h	21h	GPT1 timer 3 register	0000h
T3CONb	FF42h	A1h	GPT1 timer 3 control register	0000h
T3ICb	FF62h	B1h	GPT1 timer 3 interrupt control register	--00h
T4	FE44h	22h	GPT1 timer 4 register	0000h
T4CONb	FF44h	A2h	GPT1 timer 4 control register	0000h
T4ICb	FF64h	B2h	GPT1 timer 4 interrupt control register	--00h
T5	FE46h	23h	GPT2 timer 5 register	0000h
T5CONb	FF46h	A3h	GPT2 timer 5 control register	0000h
T5ICb	FF66h	B3h	GPT2 timer 5 interrupt control register	--00h
T6	FE48h	24h	GPT2 timer 6 register	0000h
T6CONb	FF48h	A4h	GPT2 timer 6 control register	0000h
T6ICb	FF68h	B4h	GPT2 timer 6 interrupt control register	--00h
T7	F050hE	28h	CAPCOM timer 7 register	0000h
T7CONb	FF20h	90h	CAPCOM timer 7 and 8 control register	0000h
T7ICb	F17AhE	BDh	CAPCOM timer 7 interrupt control register	--00h
T7REL	F054hE	2Ah	CAPCOM timer 7 reload register	0000h
T8	F052hE	29h	CAPCOM timer 8 register	0000h
T8ICb	F17ChE	BEh	CAPCOM timer 8 interrupt control register	--00h
T8REL	F056hE	2Bh	CAPCOM timer 8 reload register	0000h
TFR b	FFACh	D6h	Trap Flag register	0000h
WDT	FEAEh	57h	Watchdog timer register (read only)	0000h
WDTCONb	FFAEh	D7h	Watchdog timer control register	00xxh ²⁾
XADRS3	F01ChE	0Eh	XPER address select register 3	800Bh
XP0ICb	F186hE	C3h	See Section 9.1	--00h ³⁾
XP1ICb	F18EhE	C7h	See Section 9.1	--00h ³⁾
XP2ICb	F196hE	CBh	See Section 9.1	--00h ³⁾
XP3ICb	F19EhE	CFh	See Section 9.1	--00h ³⁾

Table 54. List of XBus registers (continued)

Name	Physical address	Description	Reset value
CAN2IF1MC	EE1Ch	CAN2: IF1 message control	0000h
CAN2IF2A1	EE48h	CAN2: IF2 arbitration 1	0000h
CAN2IF2A2	EE4Ah	CAN2: IF2 arbitration 2	0000h
CAN2IF2CM	EE42h	CAN2: IF2 command mask	0000h
CAN2IF2CR	EE40h	CAN2: IF2 command request	0001h
CAN2IF2DA1	EE4Eh	CAN2: IF2 data A 1	0000h
CAN2IF2DA2	EE50h	CAN2: IF2 data A 2	0000h
CAN2IF2DB1	EE52h	CAN2: IF2 data B 1	0000h
CAN2IF2DB2	EE54h	CAN2: IF2 data B 2	0000h
CAN2IF2M1	EE44h	CAN2: IF2 mask 1	FFFFh
CAN2IF2M2	EE46h	CAN2: IF2 mask 2	FFFFh
CAN2IF2MC	EE4Ch	CAN2: IF2 message control	0000h
CAN2IP1	EEA0h	CAN2: interrupt pending 1	0000h
CAN2IP2	EEA2h	CAN2: interrupt pending 2	0000h
CAN2IR	EE08h	CAN2: interrupt register	0000h
CAN2MV1	EEB0h	CAN2: message valid 1	0000h
CAN2MV2	EEB2h	CAN2: message valid 2	0000h
CAN2ND1	EE90h	CAN2: new data 1	0000h
CAN2ND2	EE92h	CAN2: new data 2	0000h
CAN2SR	EE02h	CAN2: status register	0000h
CAN2TR	EE0Ah	CAN2: test register	00x0h
CAN2TR1	EE80h	CAN2: transmission request 1	0000h
CAN2TR2	EE82h	CAN2: Transmission request 2	0000h
I2CCCR1	EA06h	I2C clock control register 1	0000h
I2CCCR2	EA0Eh	I2C clock control register 2	0000h
I2CCR	EA00h	I2C control register	0000h
I2CDR	EA0Ch	I2C data register	0000h
I2COAR1	EA08h	I2C own address register 1	0000h
I2COAR2	EA0Ah	I2C own address register 2	0000h
I2CSR1	EA02h	I2C status register 1	0000h
I2CSR2	EA04h	I2C status register 2	0000h
RTCAH	ED14h	RTC alarm register high byte	XXXXh
RTCAL	ED12h	RTC alarm register low byte	XXXXh
RTCCON	ED00H	RTC control register	000Xh

Example of external network sizing

The following hypothesis are formulated in order to proceed in designing the external network on A/D Converter input pins:

- Analog Signal Source Bandwidth (f_0):10kHz
- conversion Rate (f_C):25kHz
- Sampling Time (T_S):1 μ s
- Pin Input Capacitance (C_{P1}):5pF
- Pin Input Routing Capacitance (C_{P2}):1pF
- Sampling Capacitance (C_S):4pF
- Maximum Input Current Injection (I_{INJ}):3mA
- Maximum Analog Source Voltage (V_{AM}):12V
- Analog Source Impedance (R_S):100 Ω
- Channel Switch Resistance (R_{SW}):500 Ω
- Sampling Switch Resistance (R_{AD}):200 Ω

1. Supposing to design the filter with the pole exactly at the maximum frequency of the signal, the time constant of the filter is:

Equation 10

$$R_C C_F = \frac{1}{2\pi f_0} = 15.9\mu\text{s}$$

2. Using the relation between C_F and C_S and taking some margin (4000 instead of 2048), it is possible to define C_F :

Equation 11

$$C_F = 4000 C_S = 16\text{nF}$$

3. As a consequence of step 1 and 2, RC can be chosen:

Equation 12

$$R_F = \frac{1}{2\pi f_0 C_F} = 995\Omega \cong 1\text{k}\Omega$$

4. Considering the current injection limitation and supposing that the source can go up to 12V, the total series resistance can be defined as:

Equation 13

$$R_S + R_F + R_L = \frac{V_{AM}}{I_{INJ}} = 4\text{k}\Omega$$

from which is now simple to define the value of R_L :

Equation 14

$$R_L = \frac{V_{AM}}{I_{INJ}} - R_F - R_S = 2.9k\Omega$$

5. Now the three element of the external circuit R_F , C_F and R_L are defined. Some conditions discussed in the previous paragraphs have been used to size the component, the other must now be verified. The relation which allow to minimize the accuracy error introduced by the switched capacitance equivalent resistance is in this case:

Equation 15

$$R_{EQ} = \frac{1}{f_C C_S} = 10M\Omega$$

So the error due to the voltage partitioning between the real resistive path and C_S is less than half a count (considering the worst case when $V_A = 5V$):

Equation 16

$$V_A \cdot \frac{R_S + R_F + R_L + R_{SW} + R_{AD}}{R_{EQ}} = 2.35mV < \frac{1}{2}LSB$$

The other conditions to be verified is the time constants of the transients are really and significantly shorter than the sampling period duration T_S :

Equation 17

$$\begin{aligned} \tau_1 &= (R_{SW} + R_{AD}) \cdot C_S = 2.8ns \ll T_S = 1\mu s \\ 10 \tau_2 &= 10 R_L (C_S + C_{P1} + C_{P2}) = 290ns < T_S = 1\mu s \end{aligned}$$

For complete set of parameters characterizing the ST10F272 A/D Converter equivalent circuit, refer to [Section 24.7: A/D converter characteristics](#).

Table 70. Internal PLL divider mechanism

P0.15-13 (P0H.7-5)	XTAL Frequency	Input Prescaler	PLL		Output Prescaler	CPU Frequency $f_{CPU} = f_{XTAL} \times F$
			Multiply by	Divide by		
1 1 1	4 to 8MHz	$F_{XTAL} / 4$	64	4	–	$F_{XTAL} \times 4$
1 1 0	5.3 to 10.6MHz ¹⁾	$F_{XTAL} / 4$	48	4	–	$F_{XTAL} \times 3$
1 0 1	4 to 8MHz	$F_{XTAL} / 4$	64	2	–	$F_{XTAL} \times 8$
1 0 0	6.4 to 12MHz ¹⁾	$F_{XTAL} / 4$	40	2	–	$F_{XTAL} \times 5$
0 1 1	1 to 64MHz	–	PLL bypassed		–	$F_{XTAL} \times 1$
0 1 0	4 to 6.4MHz	$F_{XTAL} / 2$	40	2	–	$F_{XTAL} \times 10$
0 0 1	4 to 12MHz ¹⁾	–	PLL bypassed		$F_{PLL} / 2$	$F_{XTAL} / 2$
0 0 0	4MHz	$F_{XTAL} / 2$	64	2	–	$F_{XTAL} \times 16$

The PLL input frequency range is limited to 1 to 3.5MHz, while the VCO oscillation range is 64 to 128MHz. The CPU clock frequency range when PLL is used is 16 to 64MHz.

Example 1

- $F_{XTAL} = 4\text{MHz}$
- P0(15:13) = '110' (Multiplication by 3)
- PLL Input Frequency = 1MHz
- VCO frequency = 48MHz
- PLL Output Frequency = 12MHz
(VCO frequency divided by 4)
- $F_{CPU} = 12\text{MHz}$ (no effect of Output Prescaler)

Example 2

- $F_{XTAL} = 8\text{MHz}$
- P0(15:13) = '100' (Multiplication by 5)
- PLL Input Frequency = 2MHz
- VCO frequency = 80MHz
- PLL Output Frequency = 40MHz (VCO frequency divided by 2)
- $F_{CPU} = 40\text{MHz}$ (no effect of Output Prescaler)

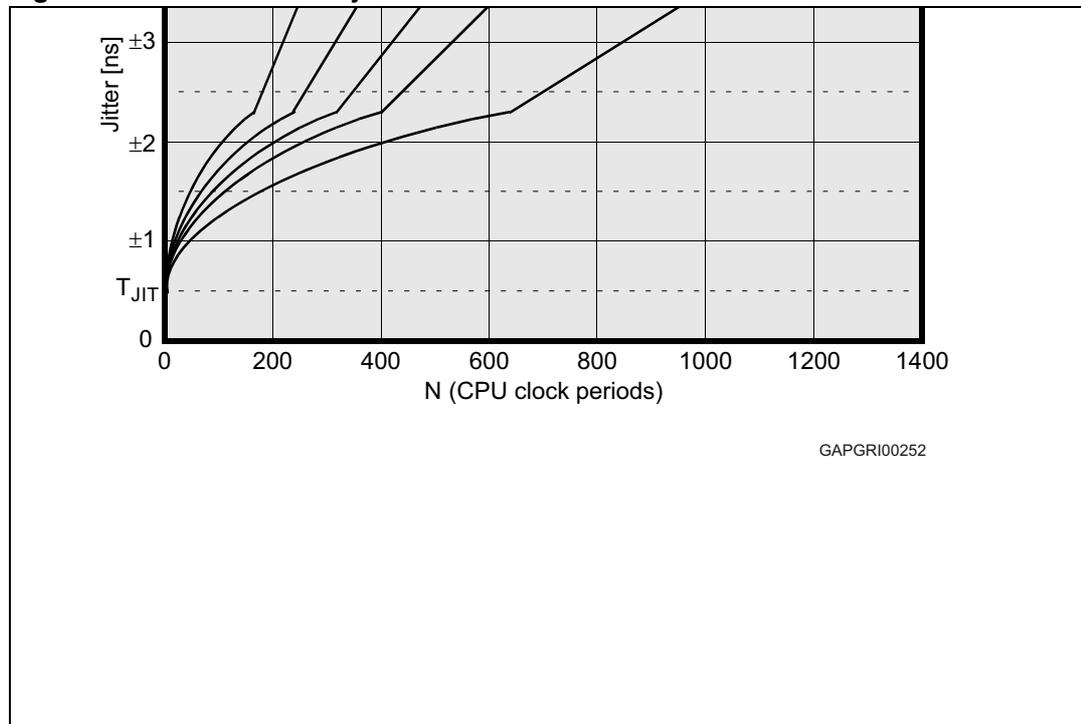
24.8.9 PLL Jitter

The following terminology is hereafter defined:

- **Self referred single period jitter**
Also called "Period Jitter", it can be defined as the difference of the T_{max} and T_{min} , where T_{max} is maximum time period of the PLL output clock and T_{min} is the minimum time period of the PLL output clock.
- **Self referred long term jitter**
Also called "N period jitter", it can be defined as the difference of T_{max} and T_{min} , where T_{max} is the maximum time difference between N+1 clock rising edges and T_{min} is the minimum time difference between N+1 clock rising edges. Here N should be kept

contribution of the digital noise to the global jitter is widely taken into account in the curves provided in [Figure 45](#).

Figure 45. ST10F272 PLL jitter



24.8.10 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{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).

Table 71. PLL characteristics ($V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, $T_A = -40$ to $+125^\circ C$)

Symbol	Parameter	Conditions	Value		Unit
			min.	max.	
T_{PSUP}	PLL Start-up time ¹⁾	Stable V_{DD} and reference clock	–	300	μs
T_{LOCK}	PLL Lock-in time	Stable V_{DD} and reference clock, starting from free-running mode	–	250	μs
T_{JIT}	Single Period Jitter ¹⁾ (cycle to cycle = 2 TCL)	6 sigma time period variation (peak to peak)	–500	+500	ps
F_{free}	PLL free running frequency	Multiplication Factors: 3, 4 Multiplication Factors: 5, 8, 10, 16	250 500	2000 4000	kHz

1. Not 100% tested, guaranteed by design characterization.

24.8.11 Main oscillator specifications

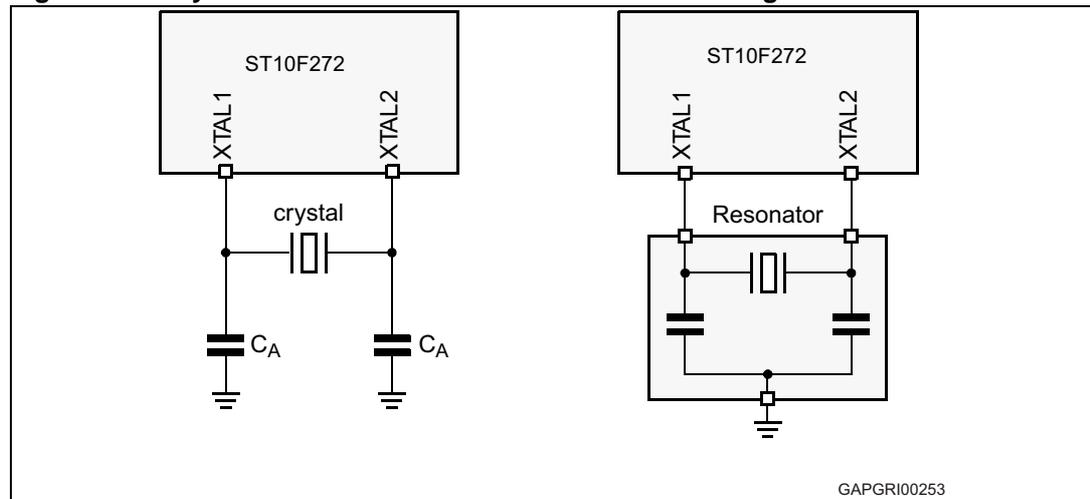
$V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, $T_A = -40$ to $+125^\circ C$

Table 72. Main oscillator characteristics

Symbol	Parameter	Conditions	Value			Unit
			min.	typ.	max.	
g_m	Oscillator Transconductance		1.4	2.6	4.2	mA/V
V_{OSC}	Oscillation Amplitude ¹⁾	Peak to Peak	–	1.5	–	V
V_{AV}	Oscillation Voltage level ¹⁾	Sine wave middle	–	0.8	–	V
t_{STUP}	Oscillator Start-up Time ¹⁾	Stable V_{DD} - Crystal	–	6	10	ms
		Stable V_{DD} - Resonator	–	1	2	ms

1. Not 100% tested, guaranteed by design characterization.

Figure 46. Crystal oscillator and resonator connection diagram



24.8.16 Multiplexed bus

$V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, $T_A = -40$ to $+125^\circ\text{C}$, $CL = 50\text{pF}$,
 ALE cycle time = $6 \text{ TCL} + 2t_A + t_C + t_F$ (75ns at 40MHz CPU clock without wait states)

Table 78. Multiplexed bus timings

Symbol	Parameter	$F_{\text{CPU}} = 40 \text{ MHz}$ $\text{TCL} = 12.5 \text{ ns}$		Variable CPU Clock $1/2 \text{ TCL} = 1$ to 64MHz		Unit
		min.	max.	min.	max.	
t_5 CC	ALE high time	$4 + t_A$	–	$\text{TCL} - 8.5 + t_A$	–	ns
t_6 CC	Address setup to ALE	$1.5 + t_A$	–	$\text{TCL} - 11 + t_A$	–	ns
t_7 CC	Address hold after ALE	$4 + t_A$	–	$\text{TCL} - 8.5 + t_A$	–	ns
t_8 CC	ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	$4 + t_A$	–	$\text{TCL} - 8.5 + t_A$	–	ns
t_9 CC	ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	$-8.5 + t_A$	–	$-8.5 + t_A$	–	ns
t_{10} CC	Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	–	6	–	6	ns
t_{11} CC	Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	–	18.5	–	$\text{TCL} + 6$	ns
t_{12} CC	$\overline{\text{RD}}$, $\overline{\text{WR}}$ low time (with RW-delay)	$15.5 + t_C$	–	$2\text{TCL} - 9.5 + t_C$	–	ns
t_{13} CC	$\overline{\text{RD}}$, $\overline{\text{WR}}$ low time (no RW-delay)	$28 + t_C$	–	$3\text{TCL} - 9.5 + t_C$	–	ns
t_{14} SR	$\overline{\text{RD}}$ to valid data in (with RW-delay)	–	$6 + t_C$	–	$2\text{TCL} - 19 + t_C$	ns
t_{15} SR	$\overline{\text{RD}}$ to valid data in (no RW-delay)	–	$18.5 + t_C$	–	$3\text{TCL} - 19 + t_C$	ns
t_{16} SR	ALE low to valid data in	–	$17.5 + t_A + t_C$	–	$3\text{TCL} - 20 + t_A + t_C$	ns
t_{17} SR	Address/Unlatched $\overline{\text{CS}}$ to valid data in	–	$20 + 2t_A + t_C$	–	$4\text{TCL} - 30 + 2t_A + t_C$	ns
t_{18} SR	Data hold after $\overline{\text{RD}}$ rising edge	0	–	0	–	ns
t_{19} SR	Data float after $\overline{\text{RD}}$	–	$16.5 + t_F$	–	$2\text{TCL} - 8.5 + t_F$	ns
t_{22} CC	Data valid to $\overline{\text{WR}}$	$10 + t_C$	–	$2\text{TCL} - 15 + t_C$	–	ns
t_{23} CC	Data hold after $\overline{\text{WR}}$	$4 + t_F$	–	$2\text{TCL} - 8.5 + t_F$	–	ns
t_{25} CC	ALE rising edge after $\overline{\text{RD}}$, $\overline{\text{WR}}$	$15 + t_F$	–	$2\text{TCL} - 10 + t_F$	–	ns
t_{27} CC	Address/Unlatched $\overline{\text{CS}}$ hold after $\overline{\text{RD}}$, $\overline{\text{WR}}$	$10 + t_F$	–	$2\text{TCL} - 15 + t_F$	–	ns
t_{38} CC	ALE falling edge to Latched $\overline{\text{CS}}$	$-4 - t_A$	$10 - t_A$	$-4 - t_A$	$10 - t_A$	ns

Table 79. Demultiplexed bus timings (continued)

Symbol	Parameter	$F_{CPU} = 40 \text{ MHz}$ $TCL = 12.5 \text{ ns}$		Variable CPU Clock 1/2 TCL = 1 to 64MHz		Unit
		min.	max.	min.	max.	
t_{38} CC	ALE falling edge to Latched \overline{CS}	$-4 - t_A$	$6 - t_A$	$-4 - t_A$	$6 - t_A$	ns
t_{39} SR	Latched \overline{CS} low to Valid Data In	–	$16.5 + t_C + 2t_A$	–	$3TCL - 21 + t_C + 2t_A$	ns
t_{41} CC	Latched \overline{CS} hold after \overline{RD} , \overline{WR}	$2 + t_F$	–	$TCL - 10.5 + t_F$	–	ns
t_{82} CC	Address setup to \overline{RdCS} , \overline{WrCS} (with RW-delay)	$14 + 2t_A$	–	$2TCL - 11 + 2t_A$	–	ns
t_{83} CC	Address setup to \overline{RdCS} , \overline{WrCS} (no RW-delay)	$2 + 2t_A$	–	$TCL - 10.5 + 2t_A$	–	ns
t_{46} SR	\overline{RdCS} to Valid Data In (with RW-delay)	–	$4 + t_C$	–	$2TCL - 21 + t_C$	ns
t_{47} SR	\overline{RdCS} to Valid Data In (no RW-delay)	–	$16.5 + t_C$	–	$3TCL - 21 + t_C$	ns
t_{48} CC	\overline{RdCS} , \overline{WrCS} Low Time (with RW-delay)	$15.5 + t_C$	–	$2TCL - 9.5 + t_C$	–	ns
t_{49} CC	\overline{RdCS} , \overline{WrCS} Low Time (no RW-delay)	$28 + t_C$	–	$3TCL - 9.5 + t_C$	–	ns
t_{50} CC	Data valid to \overline{WrCS}	$10 + t_C$	–	$2TCL - 15 + t_C$	–	ns
t_{51} SR	Data hold after \overline{RdCS}	0	–	0	–	ns
t_{53} SR	Data float after \overline{RdCS} (with RW-delay)	–	$16.5 + t_F$	–	$2TCL - 8.5 + t_F$	ns
t_{68} SR	Data float after \overline{RdCS} (no RW-delay)	–	$4 + t_F$	–	$TCL - 8.5 + t_F$	ns
t_{55} CC	Address hold after \overline{RdCS} , \overline{WrCS}	$-8.5 + t_F$	–	$-8.5 + t_F$	–	ns
t_{57} CC	Data hold after \overline{WrCS}	$2 + t_F$	–	$TCL - 10.5 + t_F$	–	ns

1. RW-delay and t_A refer to the next following bus cycle.
2. Read data are latched with the same clock edge that triggers the address change and the rising \overline{RD} edge. Therefore address changes before the end of \overline{RD} have no impact on read cycles.
3. Partially tested, guaranteed by design characterization.

Figure 55. External memory cycle: Demultipl. bus, with/without r/w delay, normal ALE, r/w CS

