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.

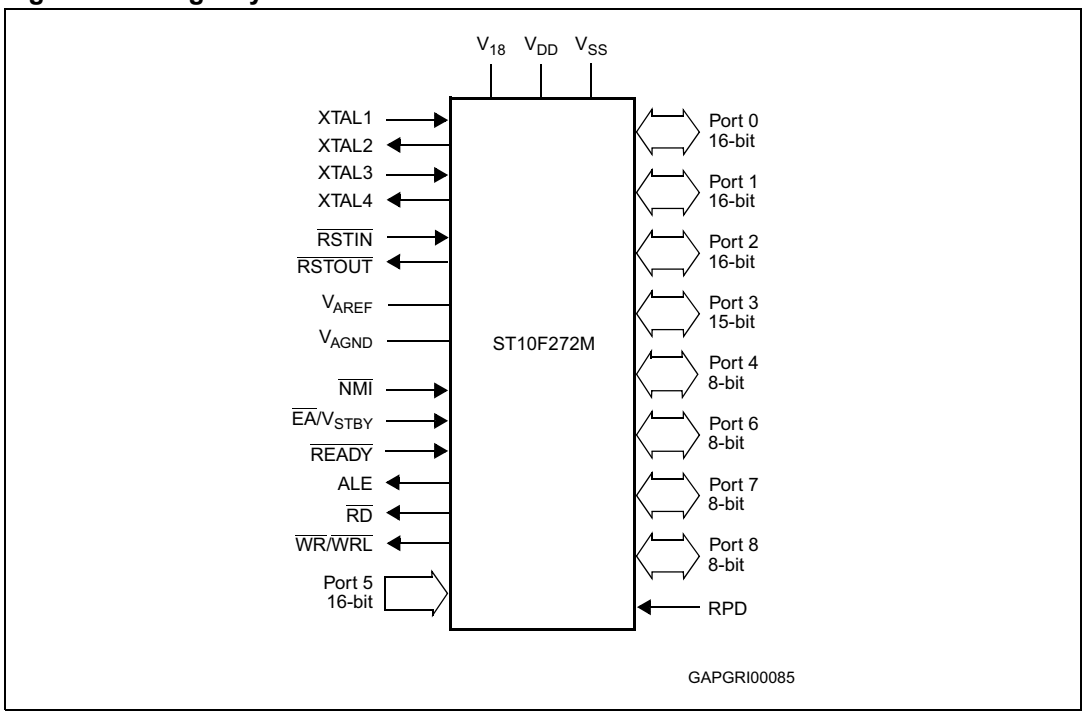
Applications of "Embedded - Microcontrollers"

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-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st10f272m-4qr3

Contents

List of figures

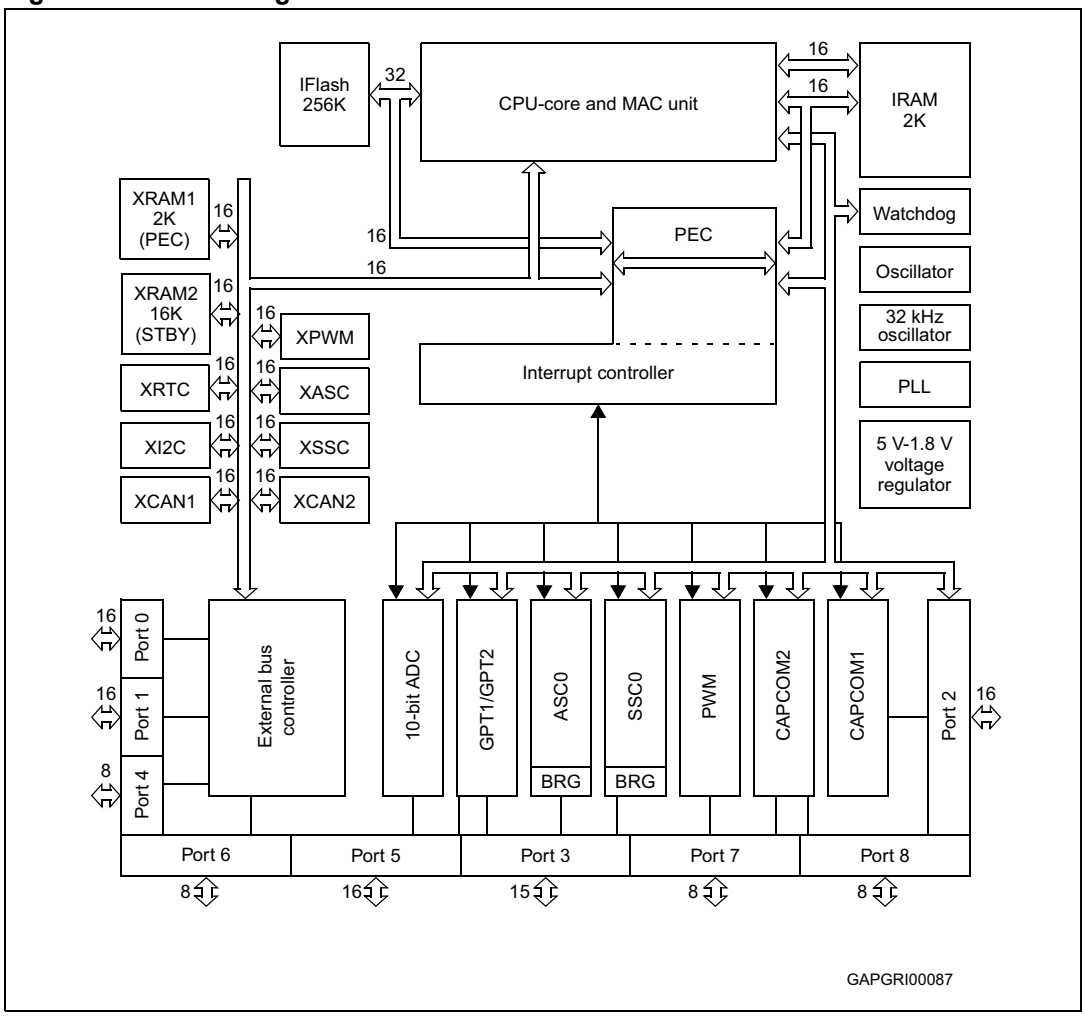
Figure 1. Logic symbol



3 Functional description

The architecture of the ST10F272M combines advantages of both RISC and CISC processors and an advanced peripheral subsystem. The block diagram gives an overview of the different on-chip components and the high bandwidth internal bus structure of the ST10F272M.

Figure 3. Block diagram



5.3 Write operation

The Flash module has one single register interface mapped in the memory space of the IBus (08'0000h - 08'0015h). 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 the IBUS operates in 16-bit mode for read/write accesses to data).

Note: The register that controls the temporary unprotection of the Flash is located on the X-bus at address 00'EB50h in the XMiscellaneous register area.

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

Caution: 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.

Warning: 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.

5.4 Registers description

5.4.1 Flash control register 0 low (FCR0L)

The Flash control register 0 low (FCR0L) together with the Flash control register 0 high (FCR0H) are used to enable and to monitor all the write operations on the IFlash. The user has no access in write mode to the test-Flash (B0TF). Moreover, the test-Flash block is seen by the user in bootstrap mode only.

FCR0L (0x08 0000) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											LOCK	Reserved		BSY0	Res.
-											RO	-		RO	-

5.5.2 Flash non-volatile write protection I register (FNVWPIR)

FNVWPIR (0x08 DFB0) NVR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								W0P7	W0P6	W0P5	W0P4	W0P3	W0P2	W0P1	W0P0
-								RW	RW	RW	RW	RW	RW	RW	RW

Table 19. Flash non-volatile write protection I register

Bit	Name	Function
7:0	W0P[7:0]	Write protection bank 0/sectors 7-0 (IFlash) These bits, if programmed at 0, disable any write access to the sectors of bank 0 (IFlash)

5.5.3 Flash non-volatile access protection register 0 (FNVAPR0)

FNVAPR0 (0x08 DFB8) NVR Delivery value: ACFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														DBGP	ACCP
-														RW	RW

Table 20. Flash non-volatile access protection register 0

Bit	Name	Function
0	ACCP	Access protection This bit, if programmed at 0, disables any access (read/write) to data mapped inside IFlash module address space, unless the current instruction is fetched from IFlash.
1	DBGP	Debug protection This bit, if erased at 1, can be used to by-pass all the protections using the debug features through the test interface. If programmed at 0, on the contrary, all the debug features, the test interface and all the Flash test modes are disabled. Even STMicroelectronics will not be able to access the device to run any eventual failure analysis.

5.5.4 Flash non-volatile access protection register 1 low (FNVAPR1L)

FNVAPR1L (0x08 DFBC) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDS15	PDS14	PDS13	PDS12	PDS11	PDS10	PDS9	PDS8	PDS7	PDS6	PDS5	PDS4	PDS3	PDS2	PDS1	PDS0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

When the access protection is enabled, Flash registers can not be written, so no program/erase operation can be run on IFlash. To enable the access to registers again, the temporary access unprotection procedure has to be followed (see *Section 5.5.9*).

5.5.8 Write protection

The Flash modules have one level of write protections: each sector of each bank of each Flash module can be software write protected by programming at 0 the related bit W0Px in FNVWPIRL register.

5.5.9 Temporary unprotection

Bits W0Px of FNVWPIRL can be temporarily unprotected by executing the set protection operation and by writing 1 into these bits.

To restore the write protection bits it is necessary to reset the microcontroller or to execute a set protection operation and write 0 into the desired bits.

In reality, when a temporary write unprotection operation is executed, the corresponding volatile register is written to 1, while the non-volatile registers bits previously written to 0 (for a protection set operation), will continue to maintain the 0. For this reason, the user software must be in charge to track the current write protection status (for instance using a specific RAM area), it is not possible to deduce it by reading the non-volatile register content (a temporary unprotection cannot be detected).

To temporarily unprotect the Flash when the access protection is active, it is necessary to set to ‘1’ the bit TAUB in XFVTAUR0. This bit can be set to ‘1’ only while executing from Flash: In this way only an instruction executed from Flash can unprotect the Flash itself.

To restore the access protection, it is necessary to reset the microcontroller or to write at 0 the bit TAUB in XFVTAUR0.

5.6 Write operation examples

In the following, examples for each kind of Flash write operation are presented.

Note: 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 write operation in Flash while fetching code from Flash.

Moreover, direct addressing is not allowed for write accesses to IFlash control registers. This means that both address and data for a writing operation must be loaded in one of ST10 GPR register (R0...R15).

Write operation on IBus registers is 16 bits wide.

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 Root part number 1 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 an X-bus 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 channels. Refer to *Section 24: Electrical characteristics* for details.

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

If static signal is applied during sampling phase, series resistance must not be greater than 20 kΩ (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 10 nF 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 Root part number 1 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.

18 Real-time clock

The real-time clock is an independent timer, in which the clock is derived directly from the clock oscillator on XTAL1 (main oscillator) input or XTAL3 input (32 kHz low-power oscillator) so that it can continue running even in Idle or power-down modes (if so enabled). Registers access is implemented onto the XBUS. This module is designed with the following characteristics:

- Generation of the current time and date for the system
- Cyclic time based interrupt, on port2 external interrupts every 'RTC basic clock tick' and after *n* 'RTC basic clock ticks' (*n* is programmable) if enabled
- 58-bit timer for long term measurement
- Capability to exit the ST10 chip from power-down mode (if PWDCFG of SYSCON set) after a programmed delay

The real-time clock is based on two main blocks of counters. The first block is a prescaler which generates a basic reference clock (for example, a 1 second period). This basic reference clock is provided by the 20-bit divider. This 20-bit counter is driven by an input clock derived from the on-chip CPU clock, predivided by a 1/64 fixed counter. This 20-bit counter is loaded at each basic reference clock period with the value of the 20-bit prescaler register. The value of the 20-bit RTCP register determines the period of the basic reference clock.

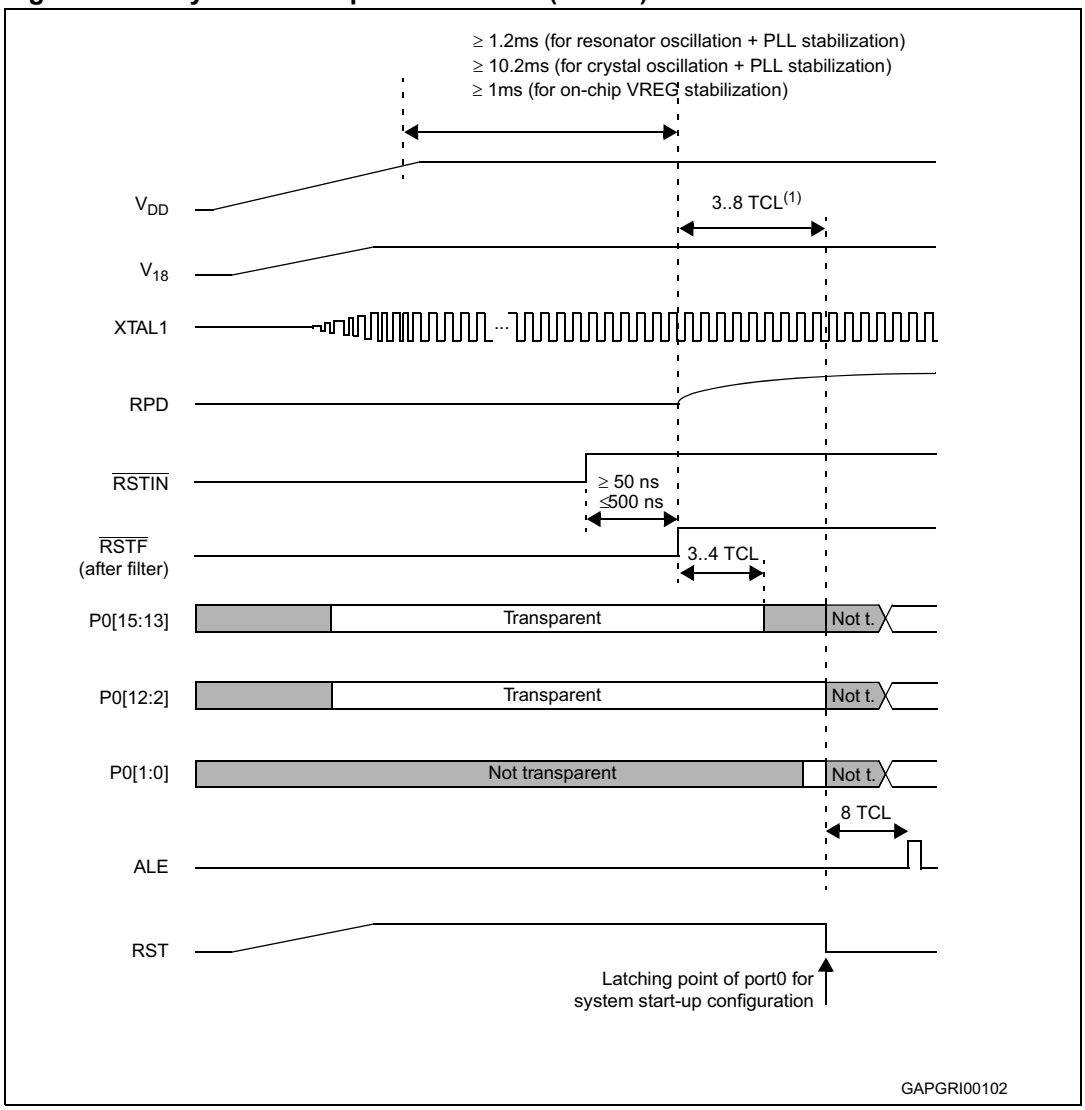
A timed interrupt request (RTCSI) may be sent on each basic reference clock period. The second block of the RTC is a 32-bit counter that may be initialized with the current system time. This counter is driven with the basic reference clock signal. In order to provide an alarm function the contents of the counter is compared with a 32-bit alarm register. The alarm register may be loaded with a reference date. An alarm interrupt request (RTCAI), may be generated when the value of the counter matches the alarm register.

The timed RTCSI and the alarm RTCAI interrupt requests can trigger a fast external interrupt via the EXISEL register of port 2 and wake up the ST10 chip when running power-down mode. Using the RTCOFF bit of the RTCCON register, the user may switch off the clock oscillator when entering the power-down mode.

The last function implemented in the RTC is to switch off the main on-chip oscillator and the 32 kHz on chip oscillator if the ST10 enters the power-down mode, so that the chip can be fully switched off (if RTC is disabled).

At power-on, and after reset phase, if the presence of a 32 kHz oscillation on XTAL3/XTAL4 pins is detected, then the RTC counter is driven by this low frequency reference clock: when Power-down mode is entered, the RTC can either be stopped or left running, and in both the cases the main oscillator is turned off, reducing the power consumption of the device to the minimum required to keep on running the RTC counter and relative reference oscillator. This is also valid if stand-by mode is entered (switching off the main supply V_{DD}), since both the RTC and the low power oscillator (32 kHz) are biased by the V_{STBY}. Vice versa, when at power on and after Reset, the 32 kHz is not present, the main oscillator drives the RTC counter, and since it is powered by the main power supply, it cannot be maintained running in stand-by mode, while in power-down mode the main oscillator is maintained running to provide the reference to the RTC module (if not disabled).

Figure 18. Asynchronous power-on reset ($\overline{EA}$ = 0)

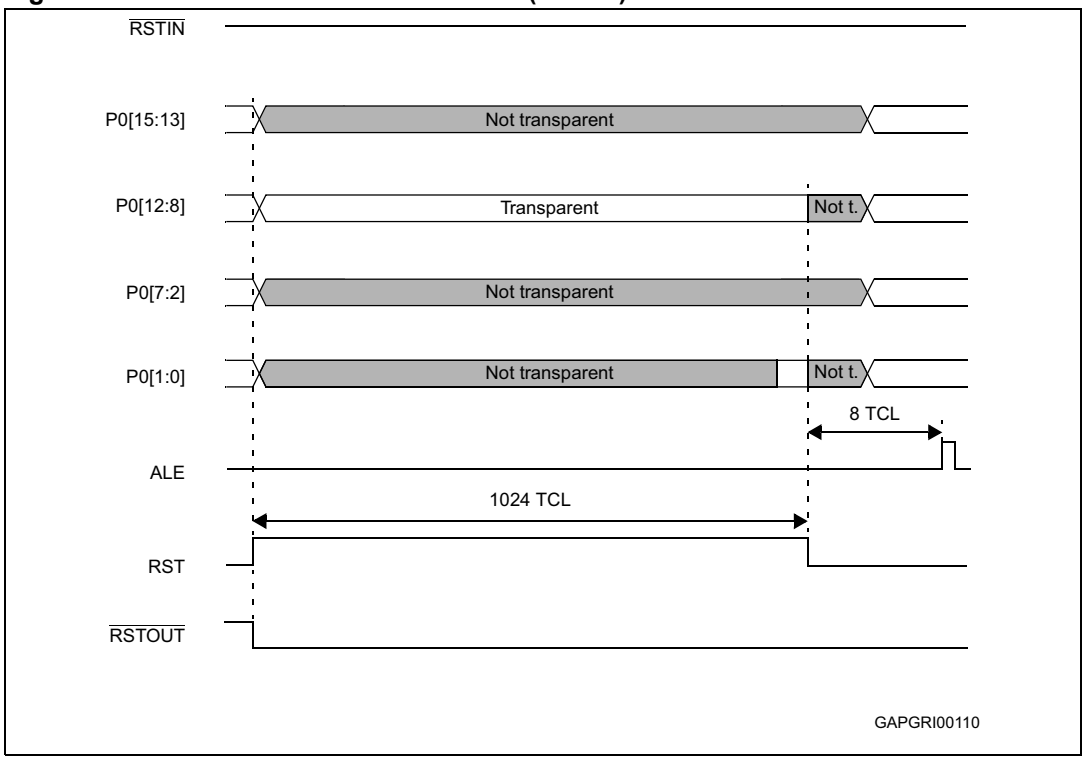


1. 3 to 8 TCL depending on clock source selection

Hardware reset

The asynchronous reset must be used to recover from catastrophic situations of the application. It may be triggered by the hardware of the application. Internal hardware logic and application circuitry are described in the reset circuitry chapter and in figures *30*, *31* and *32*. It occurs when $\overline{RSTIN}$ is low and RPD is detected (or becomes) low as well.

Figure 26. SW/WDT unidirectional reset ($\overline{EA}$ = 0)



20.6 Bidirectional reset

As shown in the previous sections, the $\overline{RSTOUT}$ pin is driven active (low level) at the beginning of any reset sequence (synchronous/asynchronous hardware, software and watchdog timer resets). $\overline{RSTOUT}$ pin stays active low beyond the end of the initialization routine, until the protected EINIT instruction (end of initialization) is completed.

The bidirectional reset function is useful when external devices require a reset signal but cannot be connected to $\overline{RSTOUT}$ pin, because $\overline{RSTOUT}$ signal lasts during initialization. It is, for instance, the case of external memory running initialization routine before the execution of EINIT instruction.

Bidirectional reset function is enabled by setting bit 3 (BDRSTEN) in SYSCON register. It only can be enabled during the initialization routine, before EINIT instruction is completed.

When enabled, the open drain of the $\overline{RSTIN}$ pin is activated, pulling down the reset signal, for the duration of the internal reset sequence (synchronous/asynchronous hardware, synchronous software and synchronous watchdog timer resets). At the end of the internal reset sequence the pull down is released and:

- After a short synchronous bidirectional hardware reset, if $\overline{RSTF}$ is sampled low eight TCL periods after the internal reset sequence completion (refer to *Figure 21* and *Figure 22*), the short reset becomes a long reset. On the contrary, if $\overline{RSTF}$ is sampled high the device simply exits reset state.
- After a software or watchdog bidirectional reset, the device exits from reset. If $\overline{RSTF}$ remains still low for at least four TCL periods (minimum time to recognize a short hardware reset) after the reset exiting (refer to *Figure 27* and *Figure 28*), the software

Table 42. Reset event (continued)

Event	RPD	$\overline{EA}$	Bidir	Synch. asynch.	RSTIN		WDTCON flags				
					Min	Max	PONR	LHWR	SHWR	SWR	WDTR
Software reset (2)	x	0	N	Synch.	Not activated		0	0	0	1	0
	x	0	N	Synch.	Not activated		0	0	0	1	0
	0	1	Y	Synch.	Not activated		0	0	0	1	0
	1	1	Y	Synch.	Activated by internal logic for 1024 TCL		0	0	0	1	0
Watchdog reset (2)	x	0	N	Synch.	Not activated		0	0	0	1	1
	x	0	N	Synch.	Not activated		0	0	0	1	1
	0	1	Y	Synch.	Not activated		0	0	0	1	1
	1	1	Y	Synch.	Activated by internal logic for 1024 TCL		0	0	0	1	1

1. It can degenerate into a long hardware reset and consequently differently flagged (see *Section 20.3* for details).
2. When bidirectional is active (and with RPD = 0), it can be followed by a short hardware reset and consequently differently flagged (see *Section 20.6* for details).

The start-up configurations and some system features are selected on reset sequences as described in *Table 43* and *Figure 35*.

Table 43 describes the system configuration latched on port0 in the six different reset modes. *Figure 35* summarizes the state of bits of PORT0 latched in RP0H, SYSCON, BUSCON0 registers.

Table 43. PORT0 latched configuration for the different reset events

X: Pin is sampled -: Pin is not sampled	Port0															
	Clock options			Segment address lines		Chip selects		WR configuration	Bus type		Reserved	BSL	Reserved	Reserved	Adapt mode	Emu mode
Sample event	P0H.7	P0H.6	P0H.5	P0H.4	P0H.3	P0H.2	P0H.1	P0H.0	P0L.7	P0L.6	P0L.5	P0L.4	P0L.3	P0L.2	P0L.1	P0L.0
Software reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Watchdog reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Synchronous short hardware reset	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X
Synchronous long hardware reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Asynchronous hardware reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Asynchronous power-on reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 46. List of X-bus registers (continued)

Name	Physical address	Description	Reset value
CAN1IF2MC	EF4Ch	CAN1: IF2 message control	0000h
CAN1IP1	EFA0h	CAN1: Interrupt pending 1	0000h
CAN1IP2	EFA2h	CAN1: Interrupt pending 2	0000h
CAN1IR	EF08h	CAN1: Interrupt register	0000h
CAN1MV1	EFB0h	CAN1: Message valid 1	0000h
CAN1MV2	EFB2h	CAN1: Message valid 2	0000h
CAN1ND1	EF90h	CAN1: New data 1	0000h
CAN1ND2	EF92h	CAN1: New data 2	0000h
CAN1SR	EF02h	CAN1: Status register	0000h
CAN1TR	EF0Ah	CAN1: Test register	00x0h
CAN1TR1	EF80h	CAN1: Transmission request 1	0000h
CAN1TR2	EF82h	CAN1: Transmission request 2	0000h
CAN2BRPER	EE0Ch	CAN2: BRP extension register	0000h
CAN2BTR	EE06h	CAN2: Bit timing register	2301h
CAN2CR	EE00h	CAN2: CAN control register	0001h
CAN2EC	EE04h	CAN2: Error counter	0000h
CAN2IF1A1	EE18h	CAN2: IF1 arbitration 1	0000h
CAN2IF1A2	EE1Ah	CAN2: IF1 arbitration 2	0000h
CAN2IF1CM	EE12h	CAN2: IF1 command mask	0000h
CAN2IF1CR	EE10h	CAN2: IF1 command request	0001h
CAN2IF1DA1	EE1Eh	CAN2: IF1 data A 1	0000h
CAN2IF1DA2	EE20h	CAN2: IF1 data A 2	0000h
CAN2IF1DB1	EE22h	CAN2: IF1 data B 1	0000h
CAN2IF1DB2	EE24h	CAN2: IF1 data B 2	0000h
CAN2IF1M1	EE14h	CAN2: IF1 mask 1	FFFFh
CAN2IF1M2	EE16h	CAN2: IF1 mask 2	FFFFh
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

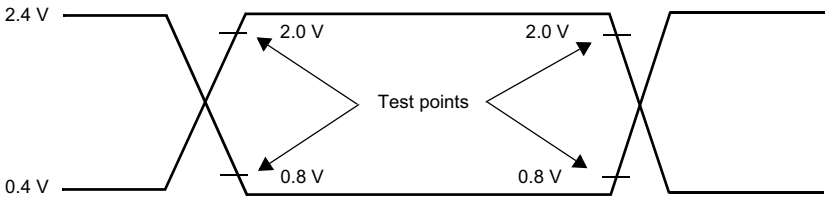
Table 46. List of X-bus registers (continued)

Name	Physical address	Description	Reset value
XPWMCON1SET	EC0Ah	XPWM module set control register 0 (write only)	0000h
XPWMPORT	EC80h	XPWM module port control register	0000h
XS1BG	E906h	XASC baudrate generator reload register	0000h
XS1CON	E900h	XASC control register	0000h
XS1CONCLR	E904h	XASC clear control register (write only)	0000h
XS1CONSET	E902h	XASC set control register (write only)	0000h
XS1PORT	E980h	XASC port control register	0000h
XS1RBUF	E90Ah	XASC receive buffer register	0000h
XS1TBUF	E908h	XASC transmit buffer register	0000h
XSSCBR	E80Ah	XSSC baudrate register	0000h
XSSCCON	E800h	XSSC control register	0000h
XSSCCONCLR	E804h	XSSC clear control register (write only)	0000h
XSSCCONSET	E802h	XSSC set control register (write only)	0000h
XSSCPORT	E880h	XSSC port control register	0000h
XSSCRB	E808h	XSSC receive buffer	XXXXh
XSSCTB	E806h	XSSC transmit buffer	0000h

24.8 AC characteristics

24.8.1 Test waveforms

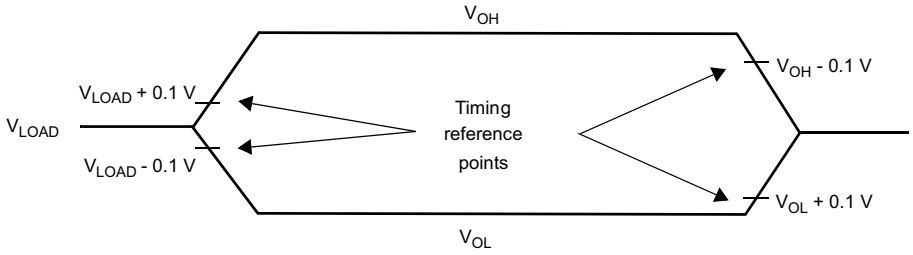
Figure 43. Input/output waveforms



AC inputs during testing are driven at 2.4 V for a logic '1' and 0.4 V for a logic '0'.
Timing measurements are made at V_{IH} min. for a logic '1' and V_{IL} max for a logic '0'.

GAPGRI00127

Figure 44. Float waveforms



For timing purposes a port pin is no longer floating when V_{LOAD} changes of ±100 mV.
It begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs (I_{OH}/I_{OL} = 20 mA).

GAPGRI00128

24.8.2 Definition of internal timing

The internal operation of the ST10F272M is controlled by the internal CPU clock f_{CPU}. Both edges of the CPU clock can trigger internal (for example, pipeline) or external (for example, bus cycles) operations.

The specification of the external timing (AC characteristics) therefore depends on the time between two consecutive edges of the CPU clock, called 'TCL'.

The CPU clock signal can be generated by different mechanisms. The duration of TCL and its variation (and also the derived external timing) depends on the mechanism used to generate f_{CPU}.

This influence must be regarded when calculating the timings for the ST10F272M.

The example for PLL operation shown in *Figure 45* refers to a PLL factor of 4.

The mechanism used to generate the CPU clock is selected during reset by the logic levels on pins P0.15-13 (P0H.7-5).

24.8.4 Prescaler operation

When pins P0.15-13 (P0H.7-5) equal ‘001’ during reset, the CPU clock is derived from the internal oscillator (input clock signal) by a 2:1 prescaler.

The frequency of f_{CPU} is half the frequency of f_{XTAL} and the high and low time of f_{CPU} (that is, the duration of an individual TCL) is defined by the period of the input clock f_{XTAL}.

The timings listed in the AC characteristics that refer to TCL therefore can be calculated using the period of f_{XTAL} for any TCL.

Note that if the bit OWDDIS in SYSCON register is cleared, the PLL runs on its free-running frequency and delivers the clock signal for the oscillator watchdog. If bit OWDDIS is set, then the PLL is switched off.

24.8.5 Direct drive

When pins P0.15-13 (P0H.7-5) equal ‘011’ during reset the on-chip phase locked loop is disabled, the on-chip oscillator amplifier is bypassed and the CPU clock is directly driven by the input clock signal on XTAL1 pin.

The frequency of CPU clock (f_{CPU}) directly follows the frequency of f_{XTAL} so the high and low time of f_{CPU} (that is, the duration of an individual TCL) is defined by the duty cycle of the input clock f_{XTAL}.

Therefore, the timings given in this chapter refer to the minimum TCL. This minimum value can be calculated by the following formula:

$$TCL_{min} = 1/f_{XTAL} \times DC_{min}$$

$$DC = \text{duty cycle}$$

For two consecutive TCLs, the deviation caused by the duty cycle of f_{XTAL} is compensated, so the duration of 2 TCL is always $1/f_{XTAL}$.

The minimum value TCL_{min} has to be used only once for timings that require an odd number of TCLs (1,3,...). Timings that require an even number of TCLs (2,4,...) may use the formula:

$$2TCL = 1/f_{XTAL}$$

The address float timings in Multiplexed bus mode (t_{11} and t_{45}) use the maximum duration of TCL ($TCL_{max} = 1/f_{XTAL} \times DC_{max}$) instead of TCL_{min}.

Similarly to what happen for prescaler operation, if the bit OWDDIS in SYSCON register is cleared, the PLL runs on its free-running frequency and delivers the clock signal for the oscillator watchdog. If bit OWDDIS is set, then the PLL is switched off.

24.8.6 Oscillator watchdog (OWD)

An on-chip watchdog oscillator is implemented in the ST10F272M. This feature is used for safety operation with external crystal oscillator (available only when using direct drive mode with or without prescaler, so the PLL is not used to generate the CPU clock multiplying the frequency of the external crystal oscillator). This watchdog oscillator operates as following.

The reset default configuration enables the watchdog oscillator. It can be disabled by setting the OWDDIS (bit 4) of SYSCON register.

When the OWD is enabled, the PLL runs at its free-running frequency, and it increments the watchdog counter. On each transition of external clock, the watchdog counter is cleared. If

an external clock failure occurs, then the watchdog counter overflows (after 16 PLL clock cycles).

The CPU clock signal will be switched to the PLL free-running clock signal, and the oscillator watchdog interrupt request is flagged. The CPU clock will not switch back to the external clock even if a valid external clock exits on XTAL1 pin. Only a hardware reset (or bidirectional software/watchdog reset) can switch the CPU clock source back to direct clock input.

When the OWD is disabled, the CPU clock is always the external oscillator clock (in direct drive or prescaler operation) and the PLL is switched off to decrease consumption supply current.

24.8.7 Phase locked loop (PLL)

For all other combinations of pins P0.15-13 (P0H.7-5) during reset the on-chip phase locked loop is enabled and it provides the CPU clock (see *Table 61*). The PLL multiplies the input frequency by the factor F which is selected via the combination of pins P0.15-13 (f_{CPU} = f_{XTAL} x F). With every F'th transition of f_{XTAL} the PLL circuit synchronizes the CPU clock to the input clock. This synchronization is done smoothly, so the CPU clock frequency does not change abruptly.

Due to this adaptation to the input clock the frequency of f_{CPU} is constantly adjusted so it is locked to f_{XTAL}. The slight variation causes a jitter of f_{CPU} which also effects the duration of individual TCLs.

The timings listed in the AC characteristics that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances.

The real minimum value for TCL depends on the jitter of the PLL. The PLL tunes f_{CPU} to keep it locked on f_{XTAL}. The relative deviation of TCL is the maximum when it is referred to one TCL period.

This is especially important for bus cycles using wait states and for example, such as for the operation of timers or serial interfaces. For all slower operations and longer periods (for example, pulse train generation or measurement, or lower baudrates) the deviation caused by the PLL jitter is negligible. Refer to next *Section 24.8.9: PLL jitter* for more details.

24.8.19 External bus arbitration

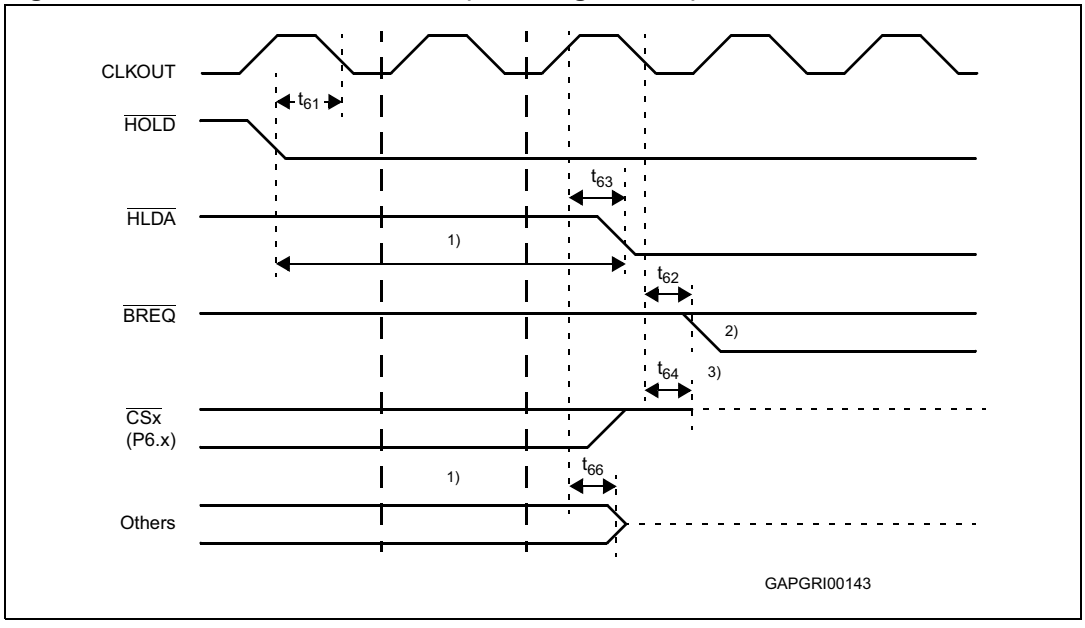
V_{DD} = 5 V ± 10 %, V_{SS} = 0 V, T_A = -40 to +125 °C, C_L = 50 pF

Table 73. External bus arbitration timings

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5ns		Variable CPU clock 1/2 TCL = 1 to 40 MHz		Unit
			Min	Max	Min	Max	
t_{61}	SR	$\overline{HOLD}$ input setup time to CLKOUT	18.5	–	18.5	–	ns
t_{62}	CC	CLKOUT to $\overline{HLDA}$ high or $\overline{BREQ}$ low delay	–	12.5	–	12.5	ns
t_{63}	CC	CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	–	12.5	–	12.5	ns
t_{64}	CC	$\overline{CSx}$ release(1)	–	20	–	20	ns
t_{65}	CC	$\overline{CSx}$ drive	-4	15	-4	15	ns
t_{66}	CC	Other signals release(1)	–	20	–	20	ns
t_{67}	CC	Other signals drive	-4	15	-4	15	ns

1. Partially tested, guaranteed by design characterization

Figure 59. External bus arbitration (releasing the bus)



1. The ST10F272M will complete the currently running bus cycle before granting bus access.
2. This is the first possibility for $\overline{BREQ}$ to become active.
3. The $\overline{CS}$ outputs will be resistive high (pull-up) after t_{64}.