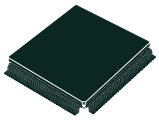
ST10F272M

16-bit MCU with 256 Kbyte Flash memory and 20 Kbyte RAM

Datasheet – production data

Features

- 16-bit CPU with DSP functions
  - 50ns instruction cycle time at 40 MHz max CPU clock
  - Multiply/accumulate unit (MAC) 16 x 16-bit multiplication, 40-bit accumulator
  - Enhanced boolean bit manipulations
  - Single-cycle context switching support
- On-chip memories
  - 256 Kbyte Flash memory (32-bit fetch)
  - Single voltage Flash memories with erase/program controller and 100 K erasing/programming cycles.
  - Up to 16 Mbyte linear address space for code and data (5 Mbytes with CAN or I^2C)
  - 2 Kbyte internal RAM (IRAM)
  - 18 Kbyte extension RAM (XRAM)
  - Programmable external bus configuration & characteristics for different address ranges
  - 5 programmable chip-select signals
  - Hold-acknowledge bus arbitration support
- Interrupt
  - 8-channel peripheral event controller for single cycle interrupt driven data transfer
  - 16-priority-level interrupt system with 56 sources, sampling rate down to 25 ns
- Timers
  - 2 multi-functional general purpose timer units with 5 timers
- Two 16-channel capture/compare units
- Serial channels
  - 2 synch./asynch. serial channels
  - 2 high-speed synchronous channels
  - One I^2C standard interface



LQFP144
20 x 20 x 1.4mm

GAPGRI00084

- 24-channel A/D converter
  - 16-channel 10-bit, accuracy ± 2 LSB
  - 8-channel 10-bit, accuracy ± 5 LSB
  - 4.85 µs minimum conversion time
- 4-channel PWM unit + 4-channel XPWM
- 2 CAN 2.0B interfaces operating on 1 or 2 CAN busses (64 or 2x32 message, C-CAN version)
- Fail-safe protection
  - Programmable watchdog timer
  - Oscillator watchdog
- On-chip bootstrap loader
- Clock generation
  - On-chip PLL with 4 to 8 MHz oscillator
  - Direct or prescaled clock input
- Real-time clock and 32 kHz on-chip oscillator
- Up to 111 general purpose I/O lines
  - Individually programmable as input, output or special function
  - Programmable threshold (hysteresis)
- Idle, power-down and stand-by modes
- Single voltage supply: 5 V ± 10 % (embedded regulator for 1.8 V core supply)
- Temperature range: -40 to +125 °C

Contents

List of tables

List of figures

1 Introduction

1.1 Description

The ST10F272M device is a new derivative of the STMicroelectronics ST10 family of 16-bit single-chip CMOS microcontrollers.

The ST10F272M combines high CPU performance (up to 20 million instructions per second) with high peripheral functionality and enhanced I/O capabilities. It also provides on-chip high-speed single voltage Flash memory, on-chip high-speed RAM, and clock generation via PLL.

The ST10F272M is processed in 0.18mm CMOS technology. The MCU core and the logic is supplied with a 5 V to 1.8 V on-chip voltage regulator. The part is supplied with a single 5 V supply and I/Os work at 5 V.

The ST10F272M is an optimized version of the ST10F272E, upward compatible with the following set of differences:

- Maximum CPU frequency is 40 MHz
- Reduced range for the Standby Voltage: V_{Stby} must be in the range of 4.5 to 5.5 V.
- Identification registers: the IDMEM register reflects the Flash type difference and can be used to differentiate the two devices by software
- Improved EMC behavior thanks to the introduction of an internal RC filter on the 5 V for the ballast transistors

1.2 Special characteristics

1.2.1 1.2.1 X-peripheral clock gating

This new feature have been implemented on the ST10F272M: Once the EINIT instruction has been executed, only the X-peripherals enabled in the XPERCON register will be clocked.

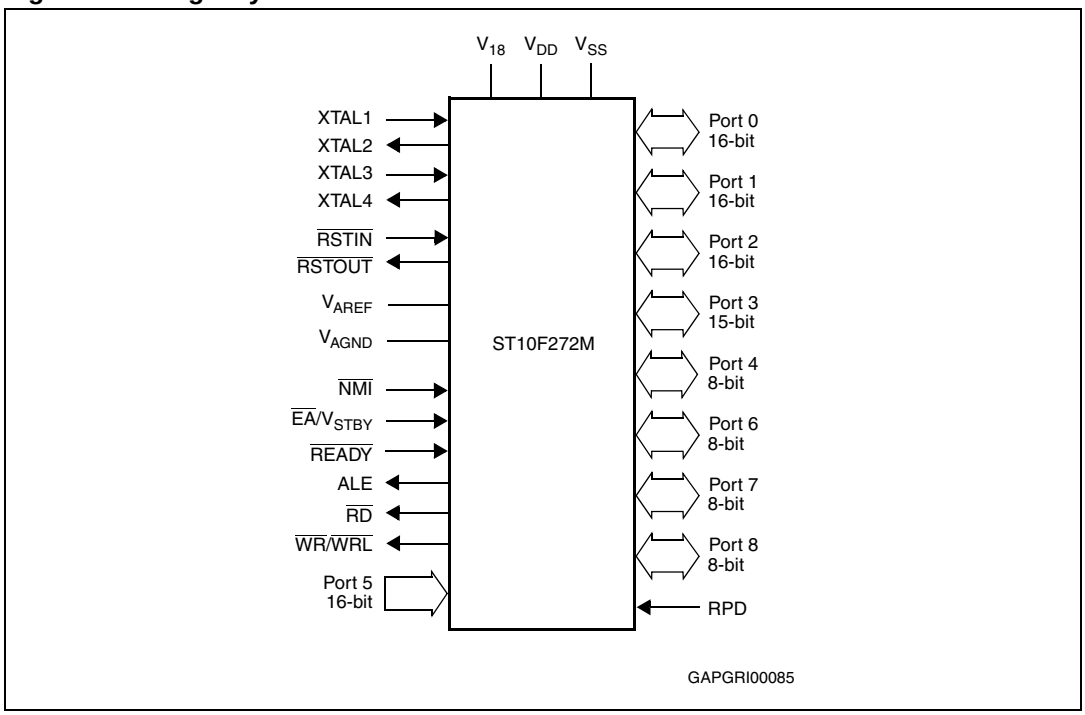
The new feature allows to reduce the power consumption and also should improve the emissions as it avoids to propagate useless clock signals across the device.

1.2.2 1.2.2 Improved supply ring

An RC filter has been introduced in the 5 V power supply ring of the ballast transistor. In addition, the supply rings for the internal voltage regulators and the I/Os have been split.

These two modifications should improve the behavior of the device regarding conducted emissions.

Figure 1. Logic symbol



2 Pin data

Figure 2. Pin configuration (top view)

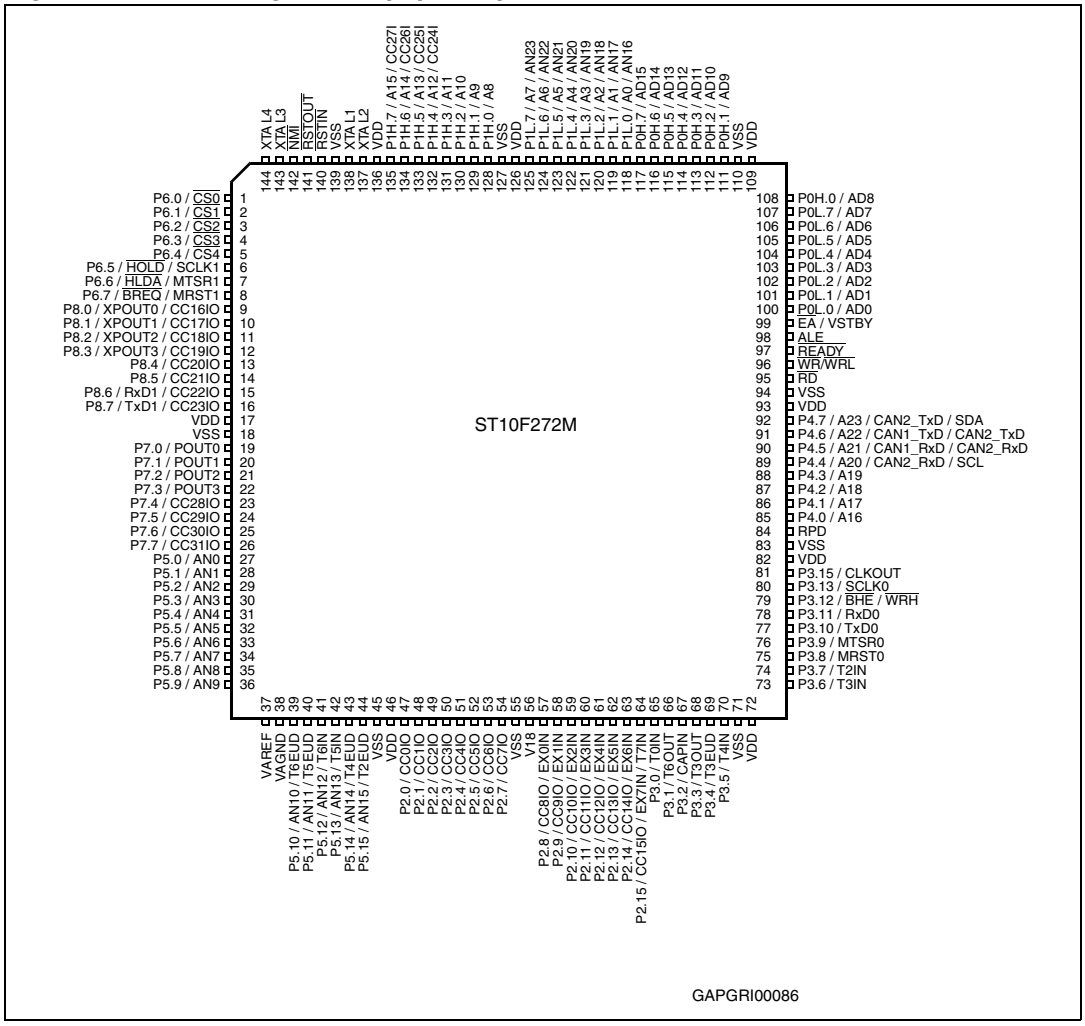


Table 1. Pin description

Symbol	Pin	Type	Function		
P6.0 - P6.7	1 - 8	I/O	8-bit bidirectional I/O port, bitwise 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 I/O
	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, bitwise 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)

Table 1. Pin description (continued)

Symbol	Pin	Type	Function		
P7.0 - P7.7	19-26	I/O	8-bit bidirectional I/O port, bitwise 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 7 outputs can be configured as push-pull or open drain drivers. The input threshold of port 7 is selectable (TTL or CMOS). The following port 7 pins have alternate functions:		
	19	O	P7.0	POUT0	PWM0: channel 0 output
	...	...	...	...	...
	22	O	P7.3	POUT3	PWM0: channel 3 output
	23	I/O	P7.4	CC28IO	CAPCOM2: CC28 capture input/compare output
	...	...	...	...	...
	26	I/O	P7.7	CC31IO	CAPCOM2: CC31 capture input/compare output
P5.0 - P5.9 P5.10 - P5.15	27-36 39-44	I I	16-bit input-only port with Schmitt-trigger characteristics. The pins of port 5 can be the analog input channels (up to 16) for the A/D converter, where P5.x equals ANx (Analog input channel x), or they are timer inputs. The input threshold of Port 5 is selectable (TTL or CMOS). The following port 5 pins have alternate functions:		
	39	I	P5.10	T6EUD	GPT2: timer T6 external up/down control input
	40	I	P5.11	T5EUD	GPT2: timer T5 external up/down control input
	41	I	P5.12	T6IN	GPT2: timer T6 count input
	42	I	P5.13	T5IN	GPT2: timer T5 count input
	43	I	P5.14	T4EUD	GPT1: timer T4 external up/down control input
	44	I	P5.15	T2EUD	GPT1: timer T2 external up/down control input
P2.0 - P2.7 P2.8 - P2.15	47-54 57-64	I/O	16-bit bidirectional I/O port, bitwise 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 2 outputs can be configured as push-pull or open drain drivers. The input threshold of Port 2 is selectable (TTL or CMOS). The following port 2 pins have alternate functions:		
	47	I/O	P2.0	CC0IO	CAPCOM: CC0 capture input/compare output
	...	...	...	...	...
	54	I/O	P2.7	CC7IO	CAPCOM: CC7 capture input/compare output
	57	I/O	P2.8	CC8IO	CAPCOM: CC8 capture input/compare output
		I		EX0IN	Fast external interrupt 0 input
	...	...	...	...	...
	64	I/O	P2.15	CC15IO	CAPCOM: CC15 capture input/compare output
		I		EX7IN	Fast external interrupt 7 input
		I		T7IN	CAPCOM2: timer T7 count input

Table 1. Pin description (continued)

Symbol	Pin	Type	Function		
P3.0 - P3.5 P3.6 - P3.13, P3.15	65-70, 73-80, 81	I/O I/O I/O	15-bit (P3.14 is missing) bidirectional I/O port, bitwise 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 3 outputs can be configured as push-pull or open drain drivers. The input threshold of port 3 is selectable (TTL or CMOS). The following port 3 pins have alternate functions:		
	65	I	P3.0	T0IN	CAPCOM1: timer T0 count input
	66	O	P3.1	T6OUT	GPT2: timer T6 toggle latch output
	67	I	P3.2	CAPIN	GPT2: register CAPREL capture input
	68	O	P3.3	T3OUT	GPT1: timer T3 toggle latch output
	69	I	P3.4	T3EUD	GPT1: timer T3 external up/down control input
	70	I	P3.5	T4IN	GPT1; timer T4 input for count/gate/reload/capture
	73	I	P3.6	T3IN	GPT1: timer T3 count/gate input
	74	I	P3.7	T2IN	GPT1: timer T2 input for count/gate/reload/capture
	75	I/O	P3.8	MRST0	SSC0: master-receiver/slave-transmitter I/O
	76	I/O	P3.9	MTSR0	SSC0: master-transmitter/slave-receiver I/O
	77	O	P3.10	TxD0	ASC0: clock/data output (asynchronous/synchronous)
	78	I/O	P3.11	RxD0	ASC0: data input (asynchronous) or I/O (synchronous)
	79	O	P3.12	$\overline{\text{BHE}}$	External memory high byte enable signal
				$\overline{\text{WRH}}$	External memory high byte write strobe
	80	I/O	P3.13	SCLK0	SSC0: master clock output/slave clock input
	81	O	P3.15	CLKOUT	System clock output (programmable divider on CPU clock)

Table 1. Pin description (continued)

Symbol	Pin	Type	Function		
P4.0 - P4.7	85-92	I/O	Port 4 is an 8-bit bidirectional I/O port. It is bitwise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. The input threshold is selectable (TTL or CMOS). Port 4.4, 4.5, 4.6 and 4.7 outputs can be configured as push-pull or open drain drivers. In case of an external bus configuration, port 4 can be used to output the segment address lines:		
	85	O	P4.0	A16	Segment address line
	86	O	P4.1	A17	Segment address line
	87	O	P4.2	A18	Segment address line
	88	O	P4.3	A19	Segment address line
	89	O	P4.4	A20	Segment address line
		I		CAN2_RxD	CAN2: receive data input
		I/O		SCL	I^2C Interface: serial clock
	90	O	P4.5	A21	Segment address line
		I		CAN1_RxD	CAN1: receive data input
		I		CAN2_RxD	CAN2: receive data input
	91	O	P4.6	A22	Segment address line
		O		CAN1_TxD	CAN1: transmit data output
		O		CAN2_TxD	CAN2: transmit data output
	92	O	P4.7	A23	Most significant segment address line
		O		CAN2_TxD	CAN2: transmit data output
		I/O		SDA	I^2C Interface: serial data
$\overline{RD}$	95	O	External memory read strobe. $\overline{RD}$ is activated for every external instruction or data read access.		
$\overline{WR}/\overline{WRL}$	96	O	External memory write strobe. In $\overline{WR}$-mode this pin is activated for every external data write access. In $\overline{WRL}$ mode this pin is activated for low byte data write accesses on a 16-bit bus, and for every data write access on an 8-bit bus. See WRCFG in the SYSCON register for mode selection.		
READY/ $\overline{READY}$	97	I	Ready input. The active level is programmable. When the ready function is enabled, the selected inactive level at this pin, during an external memory access, will force the insertion of waitstate cycles until the pin returns to the selected active level.		
ALE	98	O	Address latch enable output. In case of use of external addressing or of multiplexed mode, this signal is the latch command of the address lines.		

Table 1. Pin description (continued)

Symbol	Pin	Type	Function
$\overline{EA}$ / V_{STBY}	99	I	External access enable pin. A low level applied to this pin during and after reset forces the ST10F272M to start the program from the external memory space. A high level forces ST10F272M to start in the internal memory space. This pin is also used (when Stand-by mode is entered, that is ST10F272M under reset and main V_{DD} turned off) to bias the 32 kHz oscillator amplifier circuit and to provide a reference voltage for the low-power embedded voltage regulator which generates the internal 1.8V supply for the RTC module (when not disabled) and to retain data inside the Stand-by portion of the XRAM (16 Kbyte). It can range from 4.5 to 5.5 V. In running mode, this pin can be tied low during reset without affecting 32 kHz oscillator, RTC and XRAM activities, since the presence of a stable V_{DD} guarantees the proper biasing of all those modules.
P0L.0 - P0L.7, P0H.0, P0H.1 - P0H.7	100-107, 108, 111-117	I/O	Two 8-bit bidirectional I/O ports P0L and P0H, bitwise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. The input threshold of Port 0 is selectable (TTL or CMOS). In case of an external bus configuration, PORT0 serves as the address (A) and as the address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. Demultiplexed bus modes Data path width: 8-bit / 16-bit P0L.0 – P0L.7: D0 – D7 / D0 - D7 P0H.0 – P0H.7: I/O / D8 - D15 Multiplexed bus modes Data path width: 8-bit / 16-bit P0L.0 – P0L.7: AD0 – AD7 / AD0 - AD7 P0H.0 – P0H.7: A8 – A15 / AD8 - AD15
P1L.0 - P1L.7, P1H.0 - P1H.7	118-125 128-135	I/O	Two 8-bit bidirectional I/O ports P1L and P1H, bitwise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. port1 is used as the 16-bit address bus (A) in demultiplexed bus modes: If at least BUSCONx is configured such that the demultiplexed mode is selected, the pins of port1 are not available for general purpose I/O function. The input threshold of port 1 is selectable (TTL or CMOS). The pins of P1L also serve as the additional (up to eight) analog input channels for the A/D converter, where P1L.x equals ANy (Analog input channel y, where y = x + 16). This additional function has a higher priority on demultiplexed bus function. The following port1 pins have alternate functions:
	132	I	P1H.4 / CC24IO / CAPCOM2: CC24 capture input
	133	I	P1H.5 / CC25IO / CAPCOM2: CC25 capture input
	134	I	P1H.6 / CC26IO / CAPCOM2: CC26 capture input
	135	I	P1H.7 / CC27IO / CAPCOM2: CC27 capture input
XTAL1	138	I	XTAL1 / Main oscillator amplifier circuit and/or external clock input
XTAL2	137	O	XTAL2 / Main oscillator amplifier circuit output

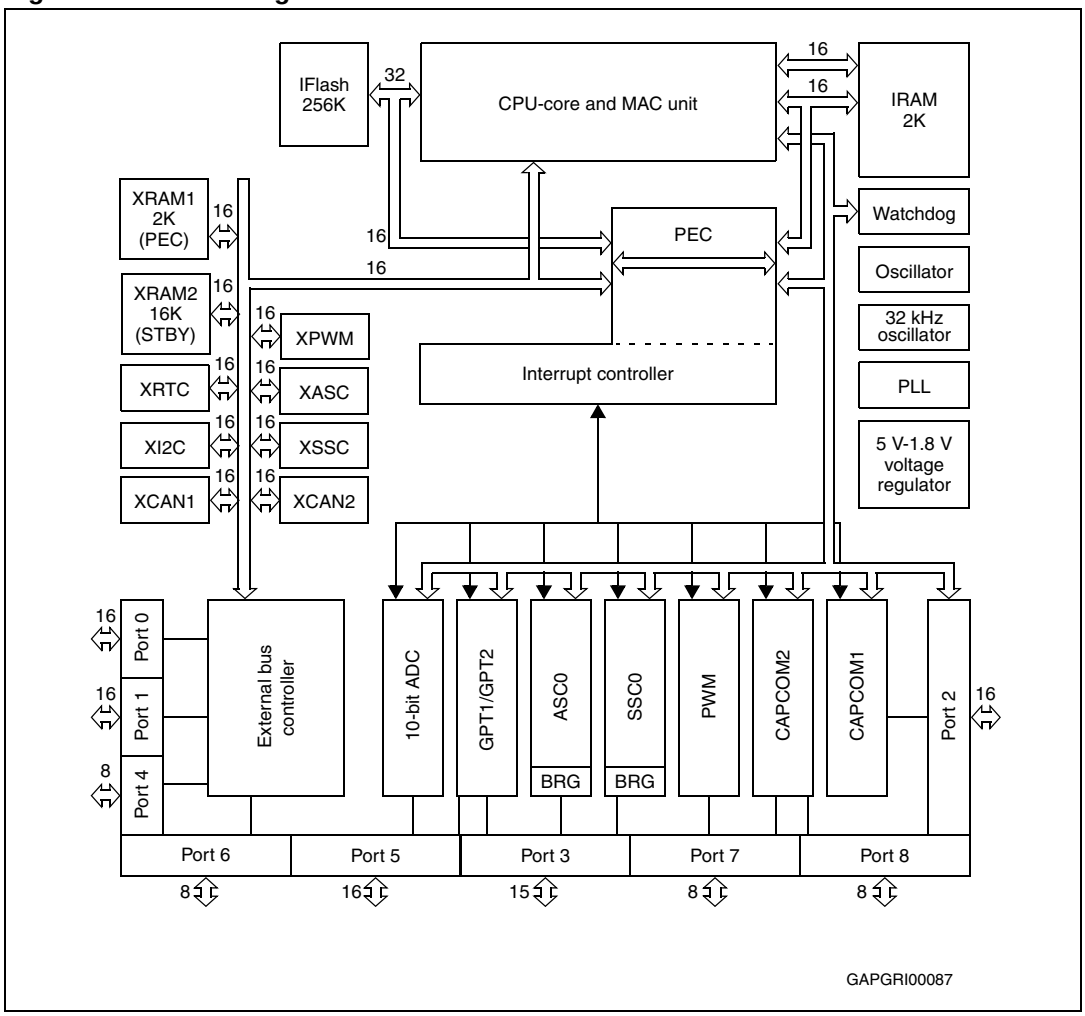
Table 1. Pin description (continued)

Symbol	Pin	Type	Function	
			To clock the device from an external source, drive XTAL1 while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC characteristics must be observed.	
XTAL3	143	I	XTAL3	32 kHz oscillator amplifier circuit input
XTAL4	144	O	XTAL4	32 kHz oscillator amplifier circuit output
			When 32 kHz oscillator amplifier is not used, to avoid spurious consumption, XTAL3 must be tied to ground while XTAL4 has to be left open. Additionally, bit OFF32 in RTCCON register must be set. 32 kHz oscillator can only be driven by an external crystal, and not by a different clock source.	
$\overline{\text{RSTIN}}$	140	I	Reset input with CMOS Schmitt-trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the ST10F272M. An internal pull-up resistor permits power-on reset using only a capacitor connected to V_{SS}. In bidirectional reset mode (enabled by setting bit BDRSTEN in SYSCON register), the $\overline{\text{RSTIN}}$ line is pulled low for the duration of the internal reset sequence.	
$\overline{\text{RSTOUT}}$	141	O	Internal reset indication output. This pin is driven to a low level during hardware, software or watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed.	
$\overline{\text{NMI}}$	142	I	Non-maskable interrupt input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. If bit PWDCFG = '0' in SYSCON register, when the PWRDN (power-down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the ST10F272M to go into power-down mode. If $\overline{\text{NMI}}$ is high and PWDCFG ='0', when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally.	
V_{AREF}	37	-	A/D converter reference voltage and analog supply	
V_{AGND}	38	-	A/D converter reference and analog ground	
RPD	84	-	Timing pin for the return from interruptible power-down mode and synchronous/ asynchronous reset selection.	
V_{DD}	17, 46, 72,82,93, 109, 126, 136	-	Digital supply voltage = +5 V during normal operation, idle and power-down modes. It can be turned off when stand-by RAM mode is selected.	
V_{SS}	18,45, 55,71, 83,94, 110, 127, 139	-	Digital ground	
V_{18}	56	-	1.8 V decoupling pin: a decoupling capacitor (typical value of 10 nF, max 100 nF) must be connected between this pin and nearest V_{SS} pin.	

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



4 Memory organization

The memory space of the ST10F272M is configured in a unified memory architecture. Code memory, data memory, registers and I/O ports are organized within the same linear address space of 16 Mbytes. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

IFlash: 256 Kbytes of on-chip Flash memory. It is divided in eight blocks (B0F0...B0F7) that constitute the bank 0. When bootstrap mode is selected, the test-Flash block B0TF (4 Kbytes) appears at address 00'0000h: refer to *Section 5: Internal Flash memory* for more details on memory mapping in boot mode. The summary of address range for IFlash is the following:

Table 2. Summary of IFlash address range

Blocks	User mode	Size (bytes)
B0TF	Not visible	4K
B0F0	00'0000h - 00'1FFFh	8K
B0F1	00'2000h - 00'3FFFh	8K
B0F2	00'4000h - 00'5FFFh	8K
B0F3	00'6000h - 00'7FFFh	8K
B0F4	01'8000h - 01'FFFFh	32K
B0F5	02'0000h - 02'FFFFh	64K
B0F6	03'0000h - 03'FFFFh	64K
B0F7	04'0000h - 04'FFFFh	64K

IRAM: 2 Kbytes of on-chip internal RAM (dual-port) is provided as a storage for data, system stack, general purpose register banks and code. A register bank is 16 wordwide (R0 to R15) and/or bytewide (RL0, RH0, …, RL7, RH7) general purpose registers group.

XRAM: 16K + 2K bytes of on-chip extension RAM (single port XRAM) is provided as a storage for data, user stack and code. The XRAM is divided into two areas, the first 2 Kbytes named XRAM1 and the second 16 Kbytes named XRAM2, connected to the internal XBUS and are accessed like an external memory in 16-bit demultiplexed bus-mode without wait state or read/write delay (50 ns access at 40 MHz CPU clock). Byte and word accesses are possible.

The XRAM1 address range is 00'E000h - 00'E7FFh if XPEN (bit 2 of SYSCON register), and XRAM1EN (bit 2 of XPERCON register) are set. If XRAM1EN or XPEN is cleared, then any access in the address range 00'E000h - 00'E7FFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register.

The XRAM2 address range is the one selected programming XADRS3 register, if XPEN (bit 2 of SYSCON register), and XRAM2EN (bit 3 of XPERCON register) are set. If bit XPEN is cleared, then any access in the address range programmed for XRAM2 will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register.

After reset, the XRAM2 address range is 09'0000h - 09'3FFFh and is mirrored every 16 Kbyte boundary until 0F'FFFFh.

XRAM2 also represents the stand-by RAM, which can be maintained biased through $\overline{EA}$ / V_{STBY} pin when main supply VDD is turned off. As the XRAM appears like external memory, it cannot be used as system stack or as register banks. The XRAM is not provided for single bit storage and therefore is not bit addressable.

SFR/ESFR: 1024 bytes (2 x 512 bytes) of address space is reserved for the special function register areas. SFRs are wordwide registers which are used to control and to monitor the function of the different on-chip units.

CAN1: Address range 00'EF00h - 00'EFFFh is reserved for the CAN1 module access. The CAN1 is enabled by setting XPEN bit 2 of the SYSCON register and by setting CAN1EN bit 0 of the XPERCON register. Accesses to the CAN module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two wait states give an access time of 100ns at 40 MHz CPU clock. No tri-state wait states are used.

CAN2: Address range 00'EE00h - 00'EEFFh is reserved for the CAN2 Module access. The CAN2 is enabled by setting XPEN bit 2 of the SYSCON register and by setting CAN2EN bit 1 of the new XPERCON register. Accesses to the CAN module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two wait states give an access time of 100ns at 40 MHz CPU clock. No tri-state wait states are used.

Note: If one or the two CAN modules are used, port 4 cannot be programmed to output all eight segment address lines. Thus, only four segment address lines can be used, reducing the external memory space to 5 Mbytes (1 Mbyte per CS line).

RTC: Address range 00'ED00h - 00'EDFFh is reserved for the RTC Module access. The RTC is enabled by setting XPEN bit 2 of the SYSCON register and bit 4 of the XPERCON register. Accesses to the RTC Module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100ns at 40 MHz CPU clock. No tristate waitstate is used.

PWM1: Address range 00'EC00h - 00'ECFFh is reserved for the PWM1 module access. The PWM1 is enabled by setting XPEN bit 2 of the SYSCON register and bit 6 of the XPERCON register. Accesses to the PWM1 Module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100ns at 40 MHz CPU clock. No tristate waitstate is used. Only word access is possible.

ASC1: Address range 00'E900h - 00'E9FFh is reserved for the ASC1 module access. The ASC1 is enabled by setting XPEN bit 2 of the SYSCON register and bit 7 of the XPERCON register. Accesses to the ASC1 module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100ns at 40 MHz CPU clock. No tristate waitstate is used.

SSC1: Address range 00'E800h - 00'E8FFh is reserved for the SSC1 Module access. The SSC1 is enabled by setting XPEN bit 2 of the SYSCON register and bit 8 of the XPERCON register. Accesses to the SSC1 module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100ns at 40 MHz CPU clock. No tristate waitstate is used.

I2C: Address range 00'EA00h - 00'EAFFh is reserved for the I^2C module access. The I^2C is enabled by setting XPEN bit 2 of the SYSCON register and bit 9 of the XPERCON register. Accesses to the I^2C module use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100 ns at 40 MHz CPU clock. No tristate waitstate is used.

X-miscellaneous: Address range 00'EB00h - 00'EBFFh is reserved for the access to a set of XBUS additional features. They are enabled by setting XPEN bit 2 of the SYSCON register and bit 10 of the XPERCON register. Accesses to this additional features use demultiplexed addresses and a 16-bit data bus (only word accesses are possible). Two waitstates give an access time of 100ns at 40 MHz CPU clock. No tristate waitstate is used. The following set of features are provided:

- CLKOUT programmable divider
- XBUS interrupt management registers
- ADC multiplexing on P1L register
- Port1L digital disable register for extra ADC channels
- CAN2 multiplexing on P4.5/P4.6
- CAN1-2 main clock prescaler
- Main voltage regulator disable for power-down mode
- TTL/CMOS threshold selection for port0, port1, and port5
- Flash temporary unprotection

In order to meet the needs of designs where more memory is required than is provided on chip, up to 16 Mbytes of external memory can be connected to the microcontroller.

Visibility of XBUS peripherals

In order to keep the ST10F272M compatible with the ST10F168/ST10F269, the XBUS peripherals can be selected to be visible on the external address / data bus. Different bits for X-peripheral enabling in XPERCON register must be set. If these bits are cleared before the global enabling with XPEN bit in SYSCON register, the corresponding address space, port pins and interrupts are not occupied by the peripherals, thus the peripheral is not visible and not available. Refer to *Chapter 23: Register set on page 108*.

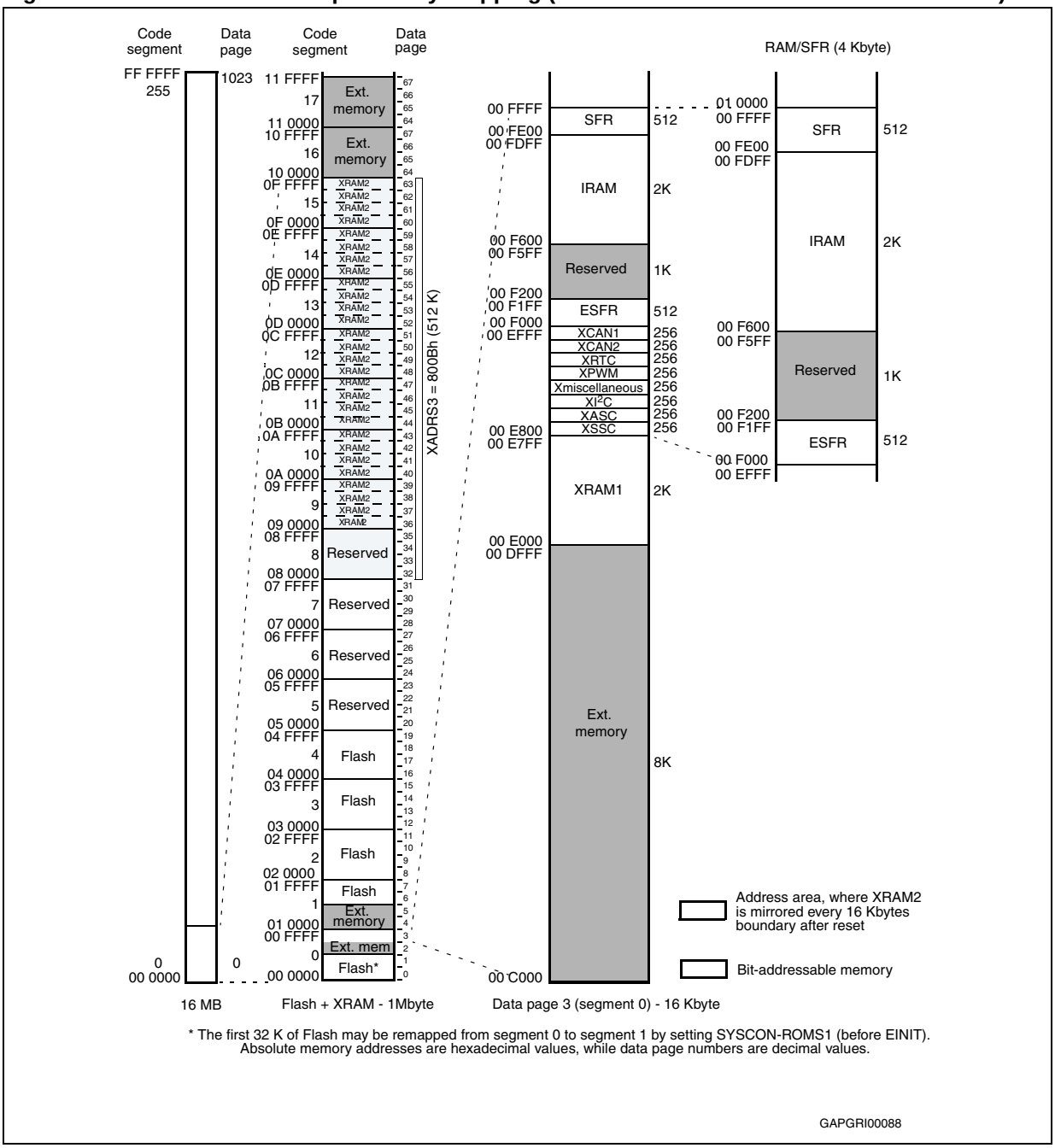
XPERCON and X-peripheral clock gating

As already mentioned, the XPERCON register must be programmed to enable the single X-bus modules separately. The XPERCON is a read/write ESFR register.

The new feature of clock gating has been implemented by means of this register: Once the EINIT instruction has been executed, all the peripherals (except RAMs and XMISC) not enabled in the XPERCON register are not be clocked. The clock gating can reduce power consumption and improve EMI when the user doesn't use all X-peripherals.

Note: When the clock has been gated in the disabled peripherals, no reset will be raised once the EINIT instruction has been executed.

Figure 4. ST10F272M on-chip memory mapping (ROMEN = 1/XADRS = 800Bh - reset value)

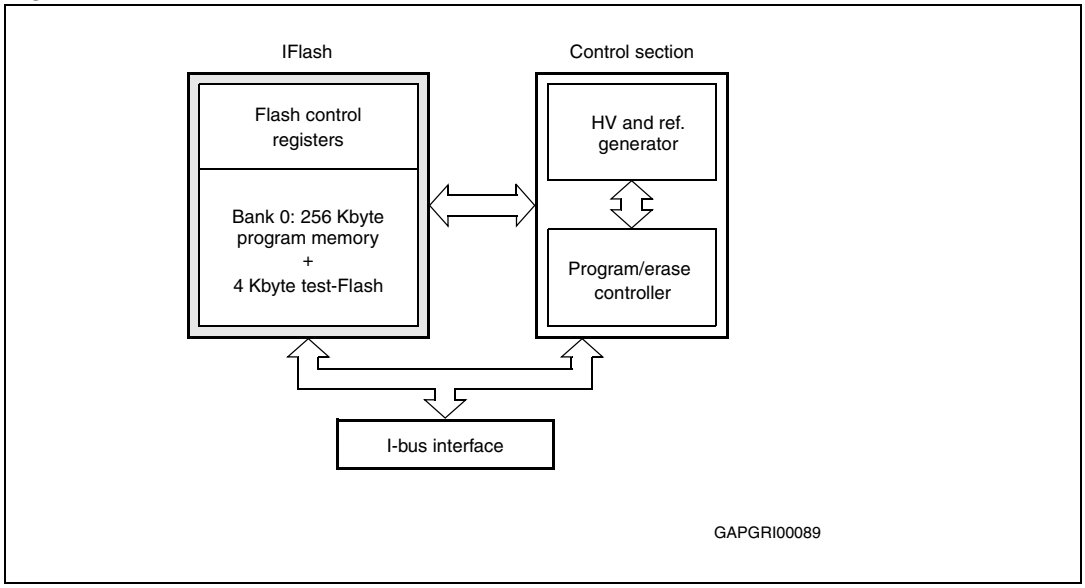


5 Internal Flash memory

5.1 Overview

The on-chip Flash is composed of one matrix module, 256 Kbytes wide. This module is called IFlash because it is on the ST10 internal bus.

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 generated internally.

The data bus is 32-bit wide for fetch accesses to IFlash, whereas it is 16-bit wide for read accesses to IFlash control registers. Write accesses are possible only in the IFlash control registers area.

5.2 Functional description

5.2.1 Structure

Table 3 below shows the address space reserved for the Flash module.

Table 3. Address space reserved for the Flash module

Description	Addresses	Size
IFlash sectors	0x00 0000 to 0x04 FFFF	256 Kbytes
Reserved IBus area	0x05 0000 to 0x07 FFFF	192 Kbytes
Registers and Flash internal reserved area	0x08 0000 to 0x08 FFFF	64 Kbytes

5.2.2 Modules structure

The IFlash module is composed of a bank (bank 0) of 256 Kbytes of program memory divided in eight sectors (B0F0...B0F7). Bank 0 also contains 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: 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 (bytes)	ST10 bus size
B0	Bank 0 Flash 0 (B0F0)	0x0000 0000 - 0x0000 1FFF	8K	32-bit (IBus)
	Bank 0 Flash 1 (B0F1)	0x0000 2000 - 0x0000 3FFF	8K	
	Bank 0 Flash 2 (B0F2)	0x0000 4000 - 0x0000 5FFF	8K	
	Bank 0 Flash 3 (B0F3)	0x0000 6000 - 0x0000 7FFF	8K	
	Bank 0 Flash 4 (B0F4)	0x0001 8000 - 0x0001 FFFF	32K	
	Bank 0 Flash 5 (B0F5)	0x0002 0000 - 0x0002 FFFF	64K	
	Bank 0 Flash 6 (B0F6)	0x0003 0000 - 0x0003 FFFF	64K	
	Bank 0 Flash 7 (B0F7)	0x0004 0000 - 0x0004 FFFF	64K	

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

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

Table 5 above refers to the configuration when bit ROMS1 of SYSCON register is set. Refer to the device user manual for more details on the memory mapping during bootstrap mode. In particular, when bootstrap mode is entered:

- Test-Flash is seen and available for code fetches (address 00'0000h)
- User IFlash is only available for read and write accesses
- Write accesses must be made with addresses starting in segment 1 from 01'0000h, whatever ROMS1 bit in SYSCON value
- Read accesses are made in segment 0 or in segment 1 depending of ROMS1 value.

In bootstrap mode, by default ROMS1 = 0, so the first 32 Kbytes of IFlash are mapped in segment 0.

Example:
In default configuration, to program address 0, the user must put the value 01'0000h in the FARL and FARH registers but to verify the content of the address 0, a read to 00'0000h must be performed.

The next *Table 6* shows the control register interface composition: This set of registers can be addressed by the CPU.

Table 6. Control register interface

Name	Description	Addresses	Size
FCR1-0	Flash control registers 1-0	0x0008 0000 - 0x0008 0007	8 bytes
FDR1-0	Flash data registers 1-0	0x0008 0008 - 0x0008 000F	8 bytes
FAR	Flash address registers	0x0008 0010 - 0x0008 0013	4 bytes
FER	Flash error register	0x0008 0014 - 0x0008 0015	2 bytes
FNVWPIR	Flash non-volatile protection i register	0x0008 DFB0 - 0x0008 DFB1	2 bytes
FNVPIR-Mirror	Mirror of Flash non-volatile protection i register	0x0008 DFB4 - 0x0008 DFB5	2 bytes
FNVAPR0	Flash non-volatile access protection register 0	0x0008 DFB8 - 0x0008 DFB9	2 bytes
FNVAPR1	Flash non-volatile access protection register 1	0x0008 DFBC - 0x0008 DFBF	4 bytes
XFVTAUR0	X-bus Flash volatile temporary access unprotection register 0	0x0000 EB50 - 0x0000 EB51	2 bytes

5.2.3 Low power mode

The Flash module is automatically switched off executing PWRDN instruction. The consumption is drastically reduced, but exiting this state can require a long time (t_{PD}).

Recovery time from power-down mode for the Flash modules is anyway shorter than the main oscillator start-up time. To avoid any problem in restarting to fetch code from the Flash, it is important to size properly the external circuit on RPD pin.

Note: PWRDN instruction must not be executed while a Flash program/erase operation is in progress.

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	-

Table 7. Flash control register 0 low

Bit	Name	Function
4	LOCK	Flash registers access locked When this bit is set, it means that the access to the Flash control registers FCR0H/-FCR1H/L, FDR0H/L-FDR1H/L, FARH/L and FER is locked by the FPEC: any read access to the registers will output invalid data (software trap 009Bh) and any write access will be ineffective. LOCK bit is automatically set when the Flash bit WMS is set. This is the only bit the user can always access to detect the status of the Flash: once it is found low, the rest of FCR0L and all the other Flash registers are accessible by the user as well. *Note that FER content can be read when LOCK is low, but its content is updated only when the BSY0 bit is reset.*
1	BSY0	Bank 0 busy (IFlash) This bit indicates that a write operation is running on bank 0 (IFlash). It is automatically set when bit WMS is set. Setting protection operation sets bit BSY0 (since protection registers are in this block). When this bit is set, every read access to bank 0 will output invalid data (software trap 009bh), while every write access to the bank will be ignored. At the end of the write operation or during a program or erase suspend this bit is automatically reset and the bank returns to read mode. After a program or erase Resume this bit is automatically set again.

5.4.2 Flash control register 0 high (FCR0H)

The Flash control register 0 high (FCR0H) together with the Flash control register 0 low (FCR0L) is 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.

FCR0H (0x08 0002) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WMS	SUSP	WPG	DWPG	SER	Reserved		SPR	Reserved							
RW	RW	RW	RW	RW	-		RW	-							

Table 8. Flash control register 0 high

Bit	Name	Function
15	WMS	Write mode start This bit must be set to start every write operation in the Flash module. At the end of the write operation or during a suspend, this bit is automatically reset. To resume a suspended operation, this bit must be set again. It is forbidden to set this bit if bit ERR of FER is high (the operation is not accepted). It is also forbidden to start a new write (program or erase) operation (by setting WMS high) when bit SUSP of FCR0 is high. Resetting this bit by software has no effect.

Table 8. Flash control register 0 high (continued)

Bit	Name	Function
14	SUSP	Suspend This bit must be set to suspend the current program (word or double word) or sector erase operation in order to read data in one of the sectors of the bank under modification or to program data in another bank. The suspend operation resets the Flash bank to normal read mode (automatically resetting bit BSY0). When in program suspend, the Flash module accepts only the following operations: Read and program resume. when in erase suspend the module accepts only the following operations: read, erase resume and program (word or double word; program operations cannot be suspended during erase suspend). To resume a suspended operation, the WMS bit must be set again, together with the selection bit corresponding to the operation to resume (WPG, DWPG, SER). *Note: It is forbidden to start a new write operation with bit SUSP already set.*
13	WPG	Word program This bit must be set to select the word (32 bits) program operation in the Flash module. The word program operation can be used to program 0s in place of 1s. The Flash address to be programmed must be written in the FARH/L registers, while the Flash data to be programmed must be written in the FDR0H/L registers before starting the execution by setting bit WMS. WPG bit is automatically reset at the end of the word program operation.
12	DWPG	Double word program This bit must be set to select the double word (64 bits) program operation in the Flash module. The double word program operation can be used to program 0s in place of 1s. The Flash address in which to program (aligned with even words) must be written in the FARH/L registers, while the two Flash data words to be programmed must be written in the FDR0H/L registers (even word) and FDR1H/L registers (odd word) before starting the execution by setting bit WMS. DWPG bit is automatically reset at the end of the double word program operation.
11	SER	Sector erase This bit must be set to select the sector erase operation in the Flash modules. The sector erase operation can be used to erase all the Flash locations to value 0xFF. From 1 to all the sectors of the same bank (excluded test-Flash for bank B0) can be selected to be erased through bits BxFy of FCR1H/L registers before starting the execution by setting bit WMS. It is not necessary to preprogram the sectors to 0x00, because this is done automatically. SER bit is automatically reset at the end of the sector erase operation.
8	SPR	Set protection This bit must be set to select the set protection operation. The set protection operation can be used to program 0s in place of 1s in the Flash non-volatile protection registers. The Flash address in which to program must be written in the FARH/L registers, while the Flash data to be programmed must be written in the FDR0H/L before starting the execution by setting bit WMS. A sequence error is flagged by bit SEQER of FER if the address written in FARH/L is not in the range of 0x0E8FB0 to 0x08DFBF. SPR bit is automatically reset at the end of the set protection operation.

5.4.3 Flash control register 1 low (FCR1L)

The Flash control register 1 Low (FCR1L), together with Flash control register 1 high (FCR1H), is used to select the sectors to erase, or during any write operation to monitor the status of each sector and bank.

FCR1L (0x08 0004) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								B0F7	B0F6	B0F5	B0F4	B0F3	B0F2	B0F1	B0F0
-								RS	RS	RS	RS	RS	RS	RS	RS

Table 9. Flash control register 1 low

Bit	Name	Function
7:0	B0F[7:0]	Bank 0 IFlash sectors 7-0 status These bits must be set during a sector erase operation to select the sectors to erase in bank 0. Moreover, during any erase operation, these bits are automatically set and give the status of the eight sectors of Bank 0 (B0F7-B0F0). The meaning of B0Fy bit for sector y of bank 0 is given in *Table 11: Banks (BxS) and sectors (BxFy) status bits meaning*. These bits are automatically reset at the end of a write operation if no errors are detected.

5.4.4 Flash control register 1 high (FCR1H)

The Flash control register 1 high (FCR1H), together with Flash control register 1 low (FCR1L), is used to select the sectors to erase, or during any write operation to monitor the status of each sector and bank.

FCR1H (0x08 0006) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							B0S	Reserved							
-							RS	-							

Table 10. Flash control register 1 high

Bit	Name	Function
8	B0S	Bank 0 status (IFlash) During any erase operation, this bit is automatically modified and gives the status of the bank 0. The meaning of B0Fy bit for sector y of bank 0 is given in *Table 11: Banks (BxS) and sectors (BxFy) status bits meaning*. This bit is automatically reset at the end of an erase operation if no errors are detected.

During any erase operation, this bit is automatically set and gives the status of the bank 0. The meaning of B0Fy bit for sector y of bank 0 is given in *Table 11: Banks (BxS) and sectors (BxFy) status bits meaning*. These bits are automatically reset at the end of an erase operation if no errors are detected.

Table 11. Banks (BxS) and sectors (BxFy) status bits meaning

ERR	SUSP	B0S = 1 meaning	B0Fy = 1 meaning
1	-	Erase error in bank0	Erase error in sector y of bank0
0	1	Erase suspended in bank0	Erase suspended in sector y of bank0
0	0	Don't care	Don't care

5.4.5 Flash data register 0 low (FDR0L)

During program operations, the Flash address registers (FARH/L) are used to store the Flash address in which to program and the Flash data registers (FDR1H/L-FDR0H/L) are used to store the Flash data to program.

FDR0L (0x08 0008) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 12. Flash data register 0 low

Bit	Name	Function
15:0	DIN[15:0]	Data Input 15:0 These bits must be written with the data to program the Flash with the following operations: Word program (32-bit), double word program (64-bit) and set protection.

5.4.6 Flash data register 0 high (FDR0H)

FDR0H (0x08 000A) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN31	DIN30	DIN29	DIN28	DIN27	DIN26	DIN25	DIN24	DIN23	DIN22	DIN21	DIN20	DIN19	DIN18	DIN17	DIN16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 13. Flash data register 0 high

Bit	Name	Function
15:0	DIN[31:16]	Data input 31:16 These bits must be written with the data to program the Flash with the following operations: Word program (32-bit), double word program (64-bit) and set protection.

5.4.7 Flash data register 1 low (FDR1L)

FDR1L (0x08 000C) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 14. Flash data register 1 low

Bit	Name	Function
15:0	DIN[15:0]	Data Input 15:0 These bits must be written with the data to program the Flash with the following operations: Word program (32-bit), double word program (64-bit) and set protection.

5.4.8 Flash data register 1 high (FDR1H)

FDR1H (0x08 000E) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN31	DIN30	DIN29	DIN28	DIN27	DIN26	DIN25	DIN24	DIN23	DIN22	DIN21	DIN20	DIN19	DIN18	DIN17	DIN16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 15. Flash data register 1 high

Bit	Name	Function
15:0	DIN[31:16]	Data input 31:16 These bits must be written with the data to program the Flash with the following operations: Word program (32-bit), double word program (64-bit) and set protection.

5.4.9 Flash address register low (FARL)

FARL (0x08 0010) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD15	ADD14	ADD13	ADD12	ADD11	ADD10	ADD9	ADD8	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	Reserved	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	

Table 16. Flash address register low

Bit	Name	Function
15:2	ADD[15:2]	Address 15:2 These bits must be written with the address of the Flash location to program in the following operations: Word program (32-bit) and double word program (64-bit). in double word program bit add2 must be written to ‘0’.

5.4.10 Flash address register high (FARH)

FARH (0x08 0012) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ADD20	ADD19	ADD18	ADD17	ADD16
-											RW	RW	RW	RW	RW

Table 17. Flash address register high

Bit	Name	Function
4:0	ADD[20:16]	Address 20:16 These bits must be written with the address of the Flash location to program in the following operations: Word program and double word program.

5.4.11 Flash error register (FER)

The Flash error register, as well as all the other Flash registers, can be read only once the LOCK bit of register FCR0L is low. Nevertheless, the FER content is updated after completion of the Flash operation, that is, when BSY0 is reset. Therefore, the FER content can only be read once the LOCK and BSY0 bits 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 18. Flash error register

Bit	Name	Function
0	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.
1	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.
2	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.
3	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.

Table 18. Flash error register (continued)

Bit	Name	Function
6	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.
7	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.
8	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.

5.5 Protection strategy

The protection bits are stored in non-volatile Flash cells inside IFlash module, that are read once at reset and stored in four volatile registers. Before they are read from the non-volatile cells, all the available protections are forced active during reset.

The protections can be programmed using the set protection operation (see Flash control registers paragraph), that can be executed from all the internal or external memories except from the Flash itself.

Two kind of protections are available: write protections to avoid unwanted writings and access protections to avoid piracy. In next paragraphs all different level of protections are shown, and architecture limitations are highlighted as well.

5.5.1 Protection registers

The four non-volatile protection registers are one time programmable for the user.

One register (FNVWPIR) is used to store the write protection fuses respectively for each sector IFlash module. The other three registers (FNVAPR0 and FNVAPR1L/H) are used to store the access protection fuses.

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

Table 21. Flash non-volatile access protection register 1 low

Bit	Name	Function
15:0	PDS[15:0]	Protections disable 15-0 If bit PDSx is programmed at 0 and bit PENx is erased at 1, the action of bit ACCP is disabled. Bit PDS0 can be programmed at 0 only if both bits DBGP and ACCP have already been programmed at 0. Bit PDSx can be programmed at 0 only if bit PENx-1 has already been programmed at 0.

5.5.5 Flash non-volatile access protection register 1 high (FNVAPR1H)

FNVAPR1H (0x08 DFBE) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN9	PEN8	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 22. Flash non-volatile access protection register 1 high

Bit	Name	Function
15:0	PEN[15:0]	Protections enable 15-0 If bit PENx is programmed at 0 and bit PDSx+1 is erased at 1, the action of bit ACCP is enabled again. Bit PENx can be programmed at 0 only if bit PDSx has already been programmed at 0.

5.5.6 X-bus Flash volatile temporary access unprotection register (XFVTAUR0)

XFVTAUR0 (0x00 EB50) NVR Reset value: 0000h

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

Table 23. X-bus Flash volatile temporary access unprotection register

Bit	Name	Function
0	TAUB	Temporary access unprotection bit If this bit is set to 1, the access protection is temporary disabled. The fact that this bit can be written only while executing from IFlash guarantees that only a code executed in IFlash can unprotect the IFlash when it is access protected.

5.5.7 Access protection

The IFlash module has one level of access protection (access to data both in reading and writing).

When bit ACCP of FNVAPR0 is programmed at 0 and bit TAUB in XFVTAUR0 is set at 0, the IFlash module becomes access protected (data in the IFlash module can be read only if the current execution is from the IFlash module itself).

Trying to read into the access protected Flash from internal RAM or external memories will output a dummy data (software trap 009Bh).

When the Flash module is protected in access, data access through PEC transfers is also forbidden. To read/write data through PEC in a protected bank, first it is necessary to temporarily unprotect the Flash module.

To enable access protection, the following sequence of operations is recommended:

- Execution from external memory or internal RAM
- Program TAUB bit at 1 in XFVTAUR0 register
- Program ACCP bit in FNVAPR0 to 0 using set protection operation
- Program TAUB bit at 0 in XFVTAUR0 register
- Access protection is active when both ACCP bit and TAUB bit are set to 0

Protection can be permanently disabled by programming bit PDS0 of FNVAPR1H, in order to analyze rejects. Protection can be permanently enabled again by programming bit PEN0 of FNVAPR1L. The action to disable and enable again access protections in a permanent way can be executed a maximum of 16 times. To execute the above described operations, the Flash has to be temporarily unprotected (see *Section 5.5.9: Temporary unprotection*).

Trying to write into the access protected Flash from internal RAM or external memories will be unsuccessful. Trying to read into the access protected Flash from internal RAM or external memories will output dummy data (software trap 0x009Bh).

When the Flash module is protected in access, data access through PEC of a peripheral is also forbidden. To read/write data in PEC mode from/to a protected bank, it is necessary to first temporarily unprotect the Flash module.

The following table summarizes all possible access protection levels: In particular, it shows what is possible and not possible to do when fetching from a memory (see fetch location column) supposing all possible access protections are enabled.

Table 24. Summary of access protection level

Fetch location	Read IFlash / jump to IFlash	Read XRAM or external memory/ jump to XRAM or external memory	Read Flash registers	Write Flash registers
Fetching from IFlash	Yes/yes	Yes/yes	Yes	No
Fetching from IRAM	No/yes	Yes/yes	No	No
Fetching from XRAM	No/yes	Yes/yes	No	No
Fetching from external memory	No/yes	Yes/yes	No	No

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.

Example of indirect addressing mode

```
MOV  RWm, #ADDRESS;        /*Load Add in RWm*/
MOV  RWn, #DATA;           /*Load Data in RWn*/
MOV  [RWm], RWn;           /*Indirect addressing*/
```

Word program

Example: 32-bit word program of data 0xAAAAAAAA at address 0x025554

```
FCR0H|= 0x2000;            /*Set WPG in FCR0H*/
FARL  = 0x5554;            /*Load Add in FARL*/
FARH  = 0x0002;            /*Load Add in FARH*/
FDR0L = 0xAAAA;            /*Load Data in FDR0L*/
FDR0H = 0xAAAA;            /*Load Data in FDR0H*/
FCR0H|= 0x8000;            /*Operation start*/
```

Double word program

Example: Double word program (64-bit) of data 0x55AA55AA at address 0x035558 and data 0xAA55AA55 at address 0x03555C in IFlash module.

```
FCR0H   |= 0x1000;         /*Set DWPG/
FARL    = 0x5558;          /*Load Add in FARL*/
FARH    = 0x0003;          /*Load Add in FARH*/
FDR0L   = 0x55AA;          /*Load Data in FDR0L*/
FDR0H   = 0x55AA;          /*Load Data in FDR0H*/
FDR1L   = 0xAA55;          /*Load Data in FDR1L*/
FDR1H   = 0xAA55;          /*Load Data in FDR1H*/
FCR0H  |= 0x8000;          /*Operation start*/
```

Double word program is always performed on the double word aligned on an even word: bit ADD2 of FARL is ignored.

Sector erase

Example: Sector erase of sectors B0F1 and B0F0 of Bank 0 in IFlash module.

```
FCR0H  |= 0x0800;          /*Set SER in FCR0H*/
FCR1L  |= 0x0003;          /*Set B0F1, B0F0*/
FCR0H  |= 0x8000;          /*Operation start*/
```

Suspend and resume

Word program, double word program, and sector erase operations can be suspended in the following way:

```
FCR0H  |= 0x4000;          /*Set SUSP in FCR0H*/
```

Then the operation can be resumed in the following way:

```
FCR0H  |= 0x0800;          /*Set SER in FCR0H*/
FCR0H  |= 0x8000;          /*Operation resume*/
```

Before resuming a suspended erase, FCR1H/FCR1L must be read to check if the erase is already completed (FCR1H = FCR1L = 0x0000 if erase is complete). Original setup of select operation bits in FCR0H/L must be restored before the operation resume, otherwise the operation is aborted and bit RESER of FER is set.

Set protection

Example 1: Enable write protection of sectors B0F3-0 of Bank 0 in IFlash module.

```
FCR0H  |= 0x0100;       /*Set SPR in FCR0H*/
FARL    = 0xDFB4;       /*Load Add of register FNVWPIR in FARL*/
FARH    = 0x0008;       /*Load Add of register FNVWPIR in FARH*/
FDR0L   = 0xFFF0;       /*Load Data in FDR0L*/
FDR0H   = 0xFFFF;       /*Load Data in FDR0H*/
FCR0H  |= 0x8000;       /*Operation start*/
```

Example 2: Enable access and debug protection.

```
FCR0H  |= 0x0100;       /*Set SPR in FCR0H*/
FARL    = 0xDFB8;       /*Load Add of register FNVAPR0 in FARL*/
FARH    = 0x0008;       /*Load Add of register FNVAPR0 in FARH*/
FDR0L   = 0xFFFC;       /*Load Data in FDR0L*/
FCR0H  |= 0x8000;       /*Operation start*/
```

Example 3: Disable in a permanent way access and debug protection.

```
XFVTAUR0   = 0x0001;    /*Set TAUB in XFVTAUR0*/
FCR0H     |= 0x0100;    /*Set SPR in FCR0H*/
FARL       = 0xDFBC;    /*Load Add of register FNVAPR1L in FARL*/
FARH       = 0x0008;    /*Load Add of register FNVAPR1L in FARH*/
FDR0L      = 0xFFFE;    /*Load Data in FDR0L for clearing PDS0*/
FCR0H     |= 0x8000;    /*Operation start*/
```

Example 4: Enable again in a permanent way access and debug protection, after having disabled them.

```
XFVTAUR0   = 0x0001;    /*Set TAUB in XFVTAUR0*/
FCR0H     |= 0x0100;    /*Set SPR in FCR0H*/
FARL       = 0xDFBC;    /*Load Add register FNVAPR1H in FARL*/
FARH       = 0x0008;    /*Load Add register FNVAPR1H in FARH*/
FDR0H      = 0xFFFE;    /*Load Data in FDR0H for clearing PEN0*/
FCR0H     |= 0x8000;    /*Operation start*/
XFVTAUR0  = 0x0000;     /*Reset TAUB in XFVTAUR0*/
```

Disable and re-enable of access and debug protection in a permanent way (as shown by examples 3 and 4) can be done for a maximum of 16 times.

5.7 Write operation summary

In general, each write operation is started through a sequence of three 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.

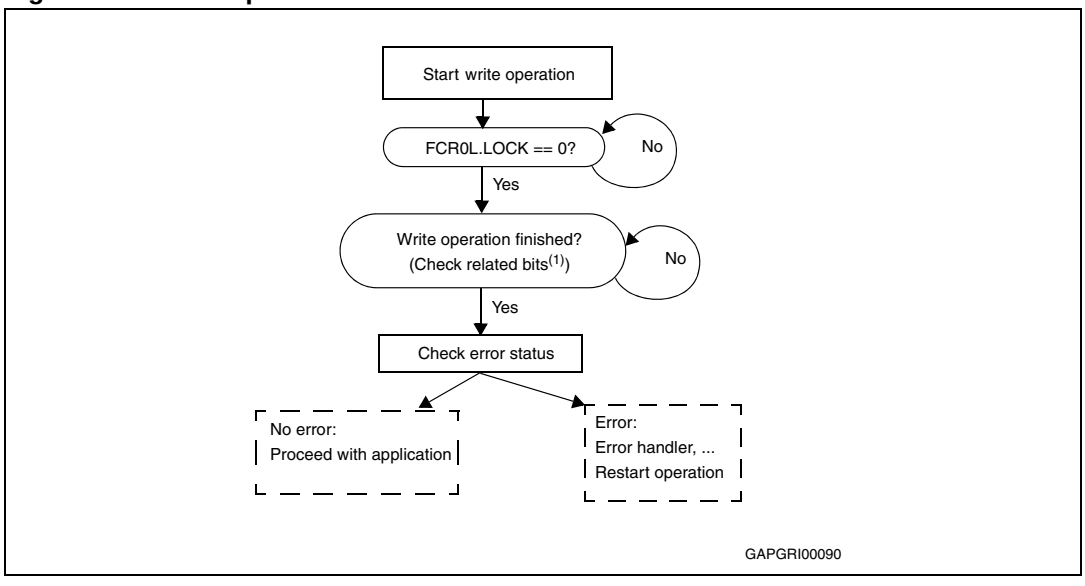
Available Flash module write operations are summarized in the following *Table 25*.

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

Figure 6 shows the complete flow needed for a write operation.

Figure 6. Write operation control flow



1. The following bits must be checked:
   - Corresponding BSYx bit in FCR0L register
   - WMS bit in FCR0H register
   - Related command bit in FCR0H register"

6 Bootstrap loader

The ST10F272M implements boot capabilities in order to:

- Support bootstrap via UART or bootstrap via CAN for the standard bootstrap
- Support a selective bootstrap loader, to manage the bootstrap sequence in a different way

6.1 Selection among user-code, standard or selective bootstrap

The boot modes are triggered with a special combination set on port0l[5...4]. Those signals, as other configuration signals, are latched on the rising edge of RSTIN pin.

- Decoding of reset configuration (P0L.5 = 1, P0L.4 = 1) selects the normal mode (also called User mode) and selects the user Flash to be mapped from address 00’0000h.
- Decoding of reset configuration (P0L.5 = 1, P0L.4 = 0) selects ST10 standard bootstrap mode (test-Flash is active and overlaps user Flash for code fetches from address 00'0000h; user Flash is active and available for read accesses).
- Decoding of reset configuration (P0L.5 = 0, P0L.4 = 1) activates new verifications to select which bootstrap software to execute:
  - if the user mode signature in the user Flash is programmed correctly, then a software reset sequence is selected and the user code is executed;
  - if the User mode signature is not programmed correctly in the user Flash, then the user key location is read again. Its value determines which communication channel will be enabled for bootstrapping.

Table 26. ST10F272M boot mode selection

P0.5	P0.4	ST10 decoding
1	1	User mode: User Flash mapped at 00’0000h
1	0	Standard bootstrap loader: User Flash mapped from 00’0000h, code fetches redirected to test-Flash at 00’0000h
0	1	Selective boot mode: User Flash mapped from 00’0000h, code fetches redirected to test-Flash at 00’0000h (different sequence execution compared to standard bootstrap loader)
0	0	Reserved

6.2 Standard bootstrap loader

After entering the standard BSL mode and the respective initialization, the ST10F272M scans the RxD0 line and the CAN1_RxD line to receive either a valid dominant bit from the CAN interface or a start condition from the UART line.

Start condition on UART RxD: ST10F272M starts standard bootstrap loader. This bootstrap loader is identical to that of other ST10 devices (example: ST10F269, ST10F168).

Valid dominant bit on CAN1 RxD: ST10F272M start bootstrapping via CAN1.

Caution: As both UART_RxD and CAN1_RxD lines are polled to detect a start of communication, ensure a stable level on the unused channel by adding a pull-up resistor.

6.3 Alternate and selective boot mode (ABM and SBM)

6.3.1 Activation of the ABM and SBM

Alternate boot is activated with the combination ‘01’ on Port0L[5..4] at the rising edge of $\overline{\text{RSTIN}}$.

6.3.2 User mode signature integrity check

The behavior of the selective boot mode is based on the computing of a signature between the content of two memory locations and a comparison with a reference signature. This requires that users who use selective boot have reserved and programmed the Flash memory locations.

6.3.3 Selective boot mode

When the user signature is not correct, instead of executing the standard bootstrap loader (triggered by P0L.4 low at reset), additional check is made.

Depending on the value at the user key location, the following behavior occurs:

- A jump is performed to the standard bootstrap loader
- Only UART is enabled for bootstrapping
- Only CAN1 is enabled for bootstrapping
- The device enters an infinite loop

7 Central processing unit (CPU)

The CPU includes a 4-stage instruction pipeline, a 16-bit arithmetic and logic unit (ALU) and dedicated SFRs. Additional hardware has been added for a separate multiply and divide unit, a bit-mask generator and a barrel shifter.

Most of the ST10F272M's instructions can be executed in one instruction cycle which requires 50 ns at 40 MHz CPU clock. For example, shift and rotate instructions are processed in one instruction cycle independent of the number of bits to be shifted.

Multiple-cycle instructions have been optimized: branches are carried out in 2 cycles, 16 x 16-bit multiplication in 5 cycles and a 32-/16-bit division in 10 cycles.

The jump cache reduces the execution time of repeatedly performed jumps in a loop, from 2 cycles to 1 cycle.

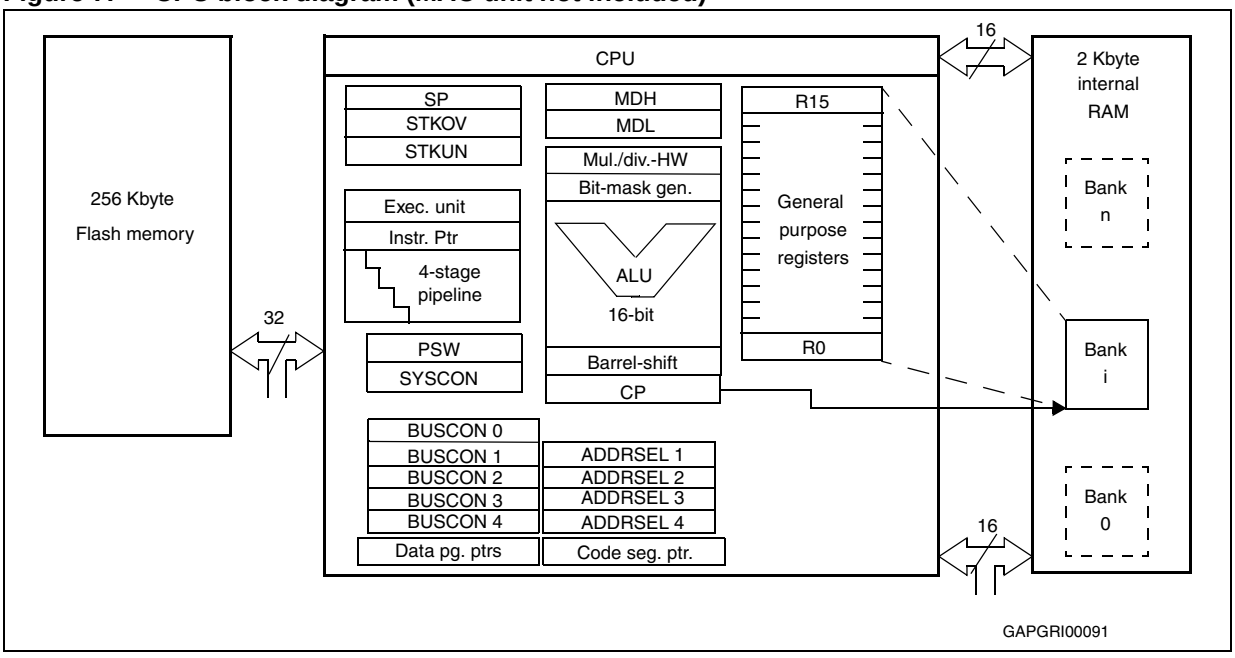
The CPU uses a bank of 16 word registers to run the current context. This bank of general purpose registers (GPR) is physically stored within the on-chip internal RAM (IRAM) area. A context pointer (CP) register determines the base address of the active register bank to be accessed by the CPU.

The number of register banks is only restricted by the available internal RAM space. For easy parameter passing, a register bank may overlap others.

A system stack of up to 2048 bytes is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register.

Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

Figure 7. CPU block diagram (MAC unit not included)



7.1 Multiplier-accumulator unit (MAC)

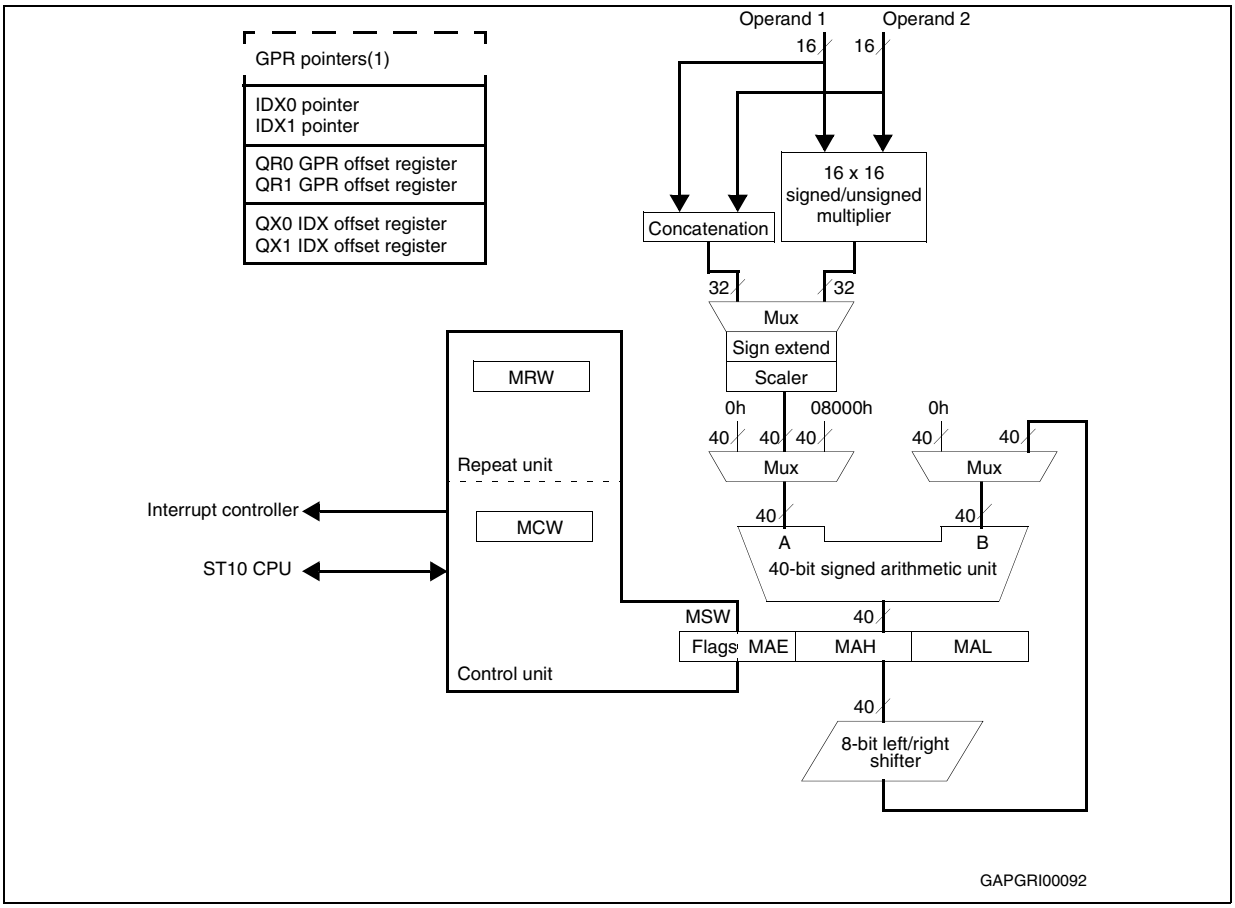
The MAC co-processor is a specialized co-processor added to the ST10 CPU core in order to improve the performances of the ST10 family in signal processing algorithms.

The standard ST10 CPU has been modified to include new addressing capabilities which enable the CPU to supply the new co-processor with up to 2 operands per instruction cycle.

This new co-processor (so-called MAC) contains a fast multiply-accumulate unit and a repeat unit.

The co-processor instructions extend the ST10 CPU instruction set with multiply, multiply-accumulate, 32-bit signed arithmetic operations.

Figure 8. MAC unit architecture



1. Shared with standard ALU

7.2 Instruction set summary

Table 27 lists the instructions of the ST10F272M. 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)	Bitwise AND, (word/byte operands)	2/4
OR(B)	Bitwise OR, (word/byte operands)	2/4
XOR(B)	Bitwise 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	Bitwise 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

Table 27. Standard instruction set summary (continued)

Mnemonic	Description	Bytes
J(N)B	Jump relative if direct bit is (not) set	4
JBC	Jump relative and clear bit if direct bit is set	4
JNBS	Jump relative and set bit if direct bit is not set	4
CALLA, CALLI, CALLR	Call absolute/indirect/relative subroutine if condition is met	4
CALLS	Call absolute subroutine in any code segment	4
PCALL	Push direct word register onto system stack and call absolute subroutine	4
TRAP	Call interrupt service routine via immediate trap number	2
PUSH, POP	Push/pop direct word register onto/from system stack	2
SCXT	Push direct word register onto system stack and update register with word operand	4
RET	Return from intra-segment subroutine	2
RETS	Return from inter-segment subroutine	2
RETP	Return from intra-segment subroutine and pop direct word register from system stack	2
RETI	Return from interrupt service subroutine	2
SRST	Software reset	4
IDLE	Enter idle mode	4
PWRDN	Enter power-down mode (supposes $\overline{\text{NMI}}$-pin being low)	4
SRVWDT	Service watchdog timer	4
DISWDT	Disable watchdog timer	4
EINIT	Signify end-of-initialization on RSTOUT-pin	4
ATOMIC	Begin ATOMIC sequence	2
EXTR	Begin EXTended register sequence	2
EXTP(R)	Begin EXTended page (and register) sequence	2/4
EXTS(R)	Begin EXTended segment (and register) sequence	2/4
NOP	Null operation	2

7.3 MAC co-processor specific instructions

Table 28 lists the MAC instructions of the ST10F272M. 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 and optional round
CoCMP	Compare accumulator with operands
CoLOAD(-,2)	Load accumulator with operands
CoMAC(R,u,s,-,rnd)	(Un)signed/(un)signed multiply-accumulate and optional round
CoMACM(R)(u,s,-,rnd)	(Un)signed/(un)signed multiply-accumulate with parallel data move and optional round
CoMAX / CoMIN	Maximum/minimum of operands and accumulator
CoMOV	Memory to memory move
CoMUL(u,s,-,rnd)	(Un)signed/(un)signed multiply and optional round
CoNEG(rnd)	Negate accumulator and optional round
CoNOP	No operation
CoRND	Round accumulator
CoSHL / CoSHR	Accumulator logical shift left/right
CoSTORE	Store a MAC unit register
CoSUB(2,R)	Subtraction

8 External bus controller

All of the external memory accesses are performed by the on-chip external bus controller.

The EBC can be programmed to single chip mode when no external memory is required, or to one of four different external memory access modes:

- 16-/18-/20-/24-bit addresses and 16-bit data, demultiplexed
- 16-/18-/20-/24-bit addresses and 16-bit data, multiplexed
- 16-/18-/20-/24-bit addresses and 8-bit data, multiplexed
- 16-/18-/20-/24-bit addresses and 8-bit data, demultiplexed

In demultiplexed bus modes addresses are output on PORT1 and data is input/output on PORT0 or P0L, respectively. In the multiplexed bus modes both addresses and data use PORT0 for input/output.

Timing characteristics of the external bus interface (memory cycle time, memory tri-state time, length of ALE and read/write delay) are programmable giving the choice of a wide range of memories and external peripherals.

Up to four independent address windows may be defined (using register pairs ADDRSELx/BUSCONx) to access different resources and bus characteristics.

These address windows are arranged hierarchically where BUSCON4 overrides BUSCON3 and BUSCON2 overrides BUSCON1.

All accesses to locations not covered by these four address windows are controlled by BUSCON0. Up to five external $\overline{CS}$ signals (four windows plus default) can be generated in order to save external glue logic. Access to very slow memories is supported by a ‘ready’ function.

A $\overline{HOLD}$/$\overline{HLDA}$ protocol is available for bus arbitration which shares external resources with other bus masters.

The bus arbitration is enabled by setting bit HLDEN in register PSW. After setting HLDEN once, pins P6.7...P6.5 ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$) are automatically controlled by the EBC. In master mode (default after reset) the $\overline{HLDA}$ pin is an output. By setting bit DP6.7 to’1’ the slave mode is selected where pin $\overline{HLDA}$ is switched to input. This directly connects the slave controller to another master controller without glue logic.

For applications which require less external memory space, the address space can be restricted to 1 Mbyte, 256 Kbytes or to 64 Kbytes. Port 4 outputs all eight address lines if an address space of 16 Mbytes is used, otherwise four, two or no address lines.

Chip select timing can be made programmable. By default (after reset), the $\overline{CSx}$ lines change half a CPU clock cycle after the rising edge of ALE. With the CSCFG bit set in the SYSCON register the $\overline{CSx}$ lines change with the rising edge of ALE.

The active level of the $\overline{READY}$ pin can be set by bit RDYPOL in the BUSCONx registers. When the READY function is enabled for a specific address window, each bus cycle within the window must be terminated with the active level defined by bit RDYPOL in the associated BUSCON register.

9 Interrupt system

The interrupt response time for internal program execution is from 125 ns to 300 ns at 40 MHz CPU clock.

The ST10F272M architecture supports several mechanisms for fast and flexible response to service requests that can be generated from various sources (internal or external) to the microcontroller. Any of these interrupt requests can be serviced by the interrupt controller or by the peripheral event controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is 'stolen' from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source or destination pointer. An individual PEC transfer counter is implicitly decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited to perform the transmission or the reception of blocks of data. The ST10F272M has eight PEC channels, each of them offers such fast interrupt-driven data transfer capabilities.

An interrupt control register which contains an interrupt request flag, an interrupt enable flag and an interrupt priority bit-field is dedicated to each existing interrupt source. Thanks to its related register, each source can be programmed to one of sixteen interrupt priority levels. Once starting to be processed by the CPU, an interrupt service can only be interrupted by a higher prioritized service request. For the standard interrupt processing, each of the possible interrupt sources has a dedicated vector location.

Software interrupts are supported by means of the 'TRAP' instruction in combination with an individual trap (interrupt) number.

Fast external interrupt inputs are provided to service external interrupts with high precision requirements. These fast interrupt inputs feature programmable edge detection (rising edge, falling edge or both edges).

Fast external interrupts may also have interrupt sources selected from other peripherals; for example, the CANx controller receives signals (CANx_RxD) and I^2C serial clock signal can be used to interrupt the system.

Table 29 shows all the available ST10F272M interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers.

Table 29. Interrupt sources

Source of interrupt or PEC service request	Request flag	Enable flag	Interrupt vector	Vector location	Trap number
CAPCOM register 0	CC0IR	CC0IE	CC0INT	00'0040h	10h
CAPCOM register 1	CC1IR	CC1IE	CC1INT	00'0044h	11h
CAPCOM register 2	CC2IR	CC2IE	CC2INT	00'0048h	12h
CAPCOM register 3	CC3IR	CC3IE	CC3INT	00'004Ch	13h
CAPCOM register 4	CC4IR	CC4IE	CC4INT	00'0050h	14h
CAPCOM register 5	CC5IR	CC5IE	CC5INT	00'0054h	15h

Table 29. Interrupt sources (continued)

Source of interrupt or PEC service request	Request flag	Enable flag	Interrupt vector	Vector location	Trap number
CAPCOM register 6	CC6IR	CC6IE	CC6INT	00'0058h	16h
CAPCOM register 7	CC7IR	CC7IE	CC7INT	00'005Ch	17h
CAPCOM register 8	CC8IR	CC8IE	CC8INT	00'0060h	18h
CAPCOM register 9	CC9IR	CC9IE	CC9INT	00'0064h	19h
CAPCOM register 10	CC10IR	CC10IE	CC10INT	00'0068h	1Ah
CAPCOM register 11	CC11IR	CC11IE	CC11INT	00'006Ch	1Bh
CAPCOM register 12	CC12IR	CC12IE	CC12INT	00'0070h	1Ch
CAPCOM register 13	CC13IR	CC13IE	CC13INT	00'0074h	1Dh
CAPCOM register 14	CC14IR	CC14IE	CC14INT	00'0078h	1Eh
CAPCOM register 15	CC15IR	CC15IE	CC15INT	00'007Ch	1Fh
CAPCOM register 16	CC16IR	CC16IE	CC16INT	00'00C0h	30h
CAPCOM register 17	CC17IR	CC17IE	CC17INT	00'00C4h	31h
CAPCOM register 18	CC18IR	CC18IE	CC18INT	00'00C8h	32h
CAPCOM register 19	CC19IR	CC19IE	CC19INT	00'00CCh	33h
CAPCOM register 20	CC20IR	CC20IE	CC20INT	00'00D0h	34h
CAPCOM register 21	CC21IR	CC21IE	CC21INT	00'00D4h	35h
CAPCOM register 22	CC22IR	CC22IE	CC22INT	00'00D8h	36h
CAPCOM register 23	CC23IR	CC23IE	CC23INT	00'00DCh	37h
CAPCOM register 24	CC24IR	CC24IE	CC24INT	00'00E0h	38h
CAPCOM register 25	CC25IR	CC25IE	CC25INT	00'00E4h	39h
CAPCOM register 26	CC26IR	CC26IE	CC26INT	00'00E8h	3Ah
CAPCOM register 27	CC27IR	CC27IE	CC27INT	00'00ECh	3Bh
CAPCOM register 28	CC28IR	CC28IE	CC28INT	00'00F0h	3Ch
CAPCOM register 29	CC29IR	CC29IE	CC29INT	00'0110h	44h
CAPCOM register 30	CC30IR	CC30IE	CC30INT	00'0114h	45h
CAPCOM register 31	CC31IR	CC31IE	CC31INT	00'0118h	46h
CAPCOM timer 0	T0IR	T0IE	T0INT	00'0080h	20h
CAPCOM timer 1	T1IR	T1IE	T1INT	00'0084h	21h
CAPCOM timer 7	T7IR	T7IE	T7INT	00'00F4h	3Dh
CAPCOM timer 8	T8IR	T8IE	T8INT	00'00F8h	3Eh
GPT1 timer 2	T2IR	T2IE	T2INT	00'0088h	22h
GPT1 timer 3	T3IR	T3IE	T3INT	00'008Ch	23h
GPT1 timer 4	T4IR	T4IE	T4INT	00'0090h	24h
GPT2 timer 5	T5IR	T5IE	T5INT	00'0094h	25h

Table 29. Interrupt sources (continued)

Source of interrupt or PEC service request	Request flag	Enable flag	Interrupt vector	Vector location	Trap number
GPT2 timer 6	T6IR	T6IE	T6INT	00'0098h	26h
GPT2 CAPREL register	CRIR	CRIE	CRINT	00'009Ch	27h
A/D conversion complete	ADCIR	ADCIE	ADCINT	00'00A0h	28h
A/D overrun error	ADEIR	ADEIE	ADEINT	00'00A4h	29h
ASC0 transmit	S0TIR	S0TIE	S0TINT	00'00A8h	2Ah
ASC0 transmit buffer	S0TBIR	S0TBIE	S0TBINT	00'011Ch	47h
ASC0 receive	S0RIR	S0RIE	S0RINT	00'00ACh	2Bh
ASC0 error	S0EIR	S0EIE	S0EINT	00'00B0h	2Ch
SSC transmit	SCTIR	SCTIE	SCTINT	00'00B4h	2Dh
SSC receive	SCRIR	SCRIE	SCRINT	00'00B8h	2Eh
SSC error	SCEIR	SCEIE	SCEINT	00'00BCh	2Fh
PWM channel 0...3	PWMIR	PWMIE	PWMINT	00'00FCh	3Fh
See *Section 9.1*	XP0IR	XP0IE	XP0INT	00'0100h	40h
See *Section 9.1*	XP1IR	XP1IE	XP1INT	00'0104h	41h
See *Section 9.1*	XP2IR	XP2IE	XP2INT	00'0108h	42h
See *Section 9.1*	XP3IR	XP3IE	XP3INT	00'010Ch	43h

Hardware traps are exceptions or error conditions that arise during run-time. They cause immediate non-maskable system reaction similar to a standard interrupt service (branching to a dedicated vector table location).

The occurrence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). A hardware trap will interrupt any other program execution except when another higher prioritized trap service is in progress. Hardware trap services cannot not be interrupted by a standard interrupt or by PEC interrupts.

9.1 X-peripheral interrupt

The limited number of X-bus interrupt lines of the present ST10 architecture, imposes some constraints on the implementation of the new functionality. In particular, the additional X-peripherals SSC1, ASC1, I^2C, PWM1 and RTC need some resources to implement interrupt and PEC transfer capabilities. For this reason, a multiplexed structure for the interrupt management is proposed. In *Figure 9*, the principle is explained through a simple diagram, which shows the basic structure replicated for each of the four X-interrupt available vectors (XP0INT, XP1INT, XP2INT and XP3INT).

It is based on a set of 16-bit registers XIRxSEL (x = 0,1,2,3), divided in two portions each:

- Byte high XIRxSEL[15:8] Interrupt enable bits
- Byte low XIRxSEL[7:0] Interrupt flag bits

When different sources submit an interrupt request, the enable bits (byte high of XIRxSEL register) define a mask which controls which sources will be associated with the unique 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 they are masked through the enable bits), allowing an effective software management even if the related interrupt request cannot be served: A periodic polling of the flag bits may be implemented inside the user application.

Figure 9. X-interrupt basic structure

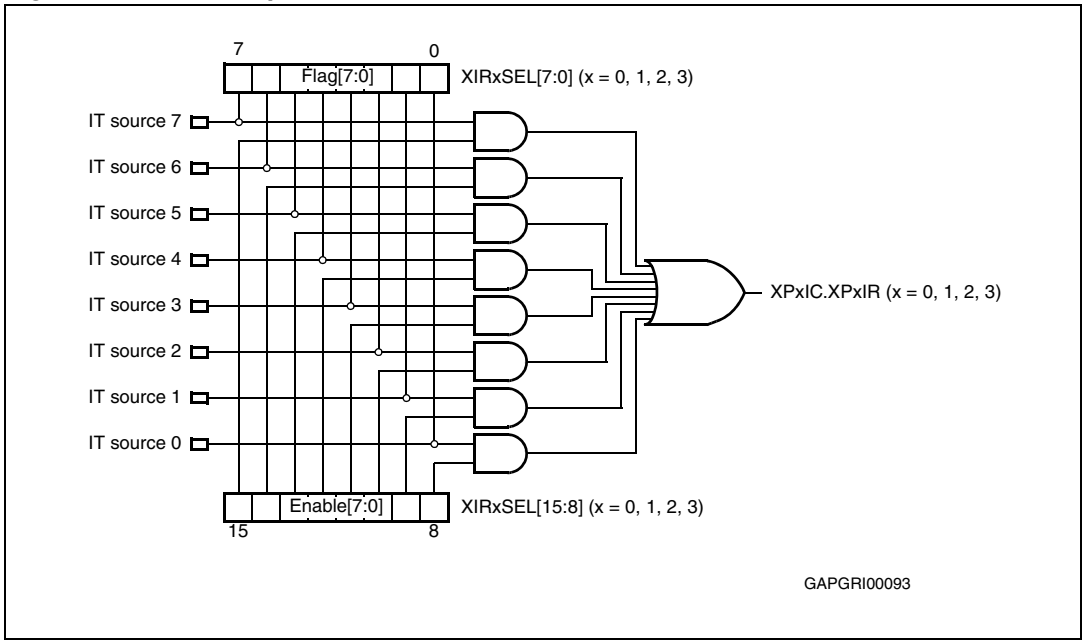


Table 30 summarizes the mapping of the different interrupt sources which shares the four X-interrupt vectors.

Table 30. X-interrupt detailed mapping

Interrupt source	XP0INT	XP1INT	XP2INT	XP3INT
CAN1 interrupt	x			x
CAN2 interrupt		x		x
I^2C receive	x	x	x	
I^2C transmit	x	x	x	
I^2C 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	

Table 30. X-interrupt detailed mapping (continued)

Interrupt source	XP0INT	XP1INT	XP2INT	XP3INT
ASC1 transmit buffer	x	x	x	
ASC1 error				x
PLL unlock/OWD				x
PWM1 channel 3...0			x	x

9.2 Exception and error traps list

Table 31 shows all of the possible exceptions or error conditions that can arise during run-time.

Table 31. Trap priorities

Exception condition	Trap flag	Trap vector	Vector location	Trap number	Trap priority(1)
Reset functions: Hardware reset Software reset Watchdog timer overflow		 Reset Reset Reset	 00'0000h 00'0000h 00'0000h	 00h 00h 00h	 III III III
Class A hardware traps: Non-maskable interrupt Stack overflow Stack underflow	 NMI STKOF STKUF	 NMITRAP STOTRAP STUTRAP	 00'0008h 00'0010h 00'0018h	 02h 04h 06h	 II II II
Class B hardware traps: Undefined opcode MAC interruption Protected instruction fault Illegal word operand access Illegal instruction access Illegal external bus access	 UNDOPC MACTRP PRTFLT ILLOPA ILLINA ILLBUS	 BTRAP BTRAP BTRAP BTRAP BTRAP BTRAP	 00'0028h 00'0028h 00'0028h 00'0028h 00'0028h 00'0028h	 0Ah 0Ah 0Ah 0Ah 0Ah 0Ah	 I I I I I I
Reserved			[002Ch - 003Ch]	[0Bh - 0Fh]	
Software traps TRAP instruction			Any 0000h – 01FCh in steps of 4h	Any [00h - 7Fh]	Current CPU priority

1. - All the class B traps have the same trap number (and vector) and the same lower priority compared to the class A traps and to the resets.
   - Each class A trap has a dedicated trap number (and vector). They are prioritized in the second priority level.
   - The resets have the highest priority level and the same trap number.
   - The PSW.ILVL CPU priority is forced to the highest level (15) when these exceptions are serviced.

10 Capture/compare (CAPCOM) units

The ST10F272M has two 16-channel CAPCOM units which support generation and control of timing sequences on up to 32 channels with a maximum resolution of 125 ns at 40 MHz CPU clock.

The CAPCOM units are typically used to handle high speed I/O tasks such as pulse and waveform generation, pulse width modulation (PMW), digital to analog (D/A) conversion, software timing, or time recording relative to external events.

Four 16-bit timers (T0/T1, T7/T8) with reload registers provide two independent time bases for the capture/compare register array.

The input clock for the timers is programmable to several prescaled values of the internal system clock, or may be derived from an overflow/underflow of timer T6 in module GPT2.

This provides a wide range of variation for the timer period and resolution, and allows precise adjustments to application specific requirements. In addition, external count inputs for CAPCOM timers T0 and T7 allow event scheduling for the capture/compare registers relative to external events.

Each of the two capture/compare register arrays contain 16 dual purpose capture/compare registers, each of which may be individually allocated to either CAPCOM timer T0 or T1 (T7 or T8, respectively), and programmed for capture or compare functions. Each of the 32 registers has one associated port pin which serves as an input pin for triggering the capture function, or as an output pin to indicate the occurrence of a compare event.

When a capture/compare register has been selected for capture mode, the current contents of the allocated timer will be latched (captured) into the capture/compare register in response to an external event at the port pin which is associated with this register. In addition, a specific interrupt request for this capture/compare register is generated.

Either a positive, a negative, or both a positive and a negative external signal transition at the pin can be selected as the triggering event. The contents of all registers which have been selected for one of the five compare modes are continuously compared with the contents of the allocated timers.

When a match occurs between the timer value and the value in a capture/compare register, specific actions will be taken based on the selected compare mode.

The input frequencies f_{Tx}, for the timer input selector Tx, are determined as a function of the CPU clocks. The timer input frequencies, resolution and periods which result from the selected prescaler option in TxI when using a 40 MHz CPU clock are listed in *Table 33*.

The numbers for the timer periods are based on a reload value of 0000h. Note that some numbers may be rounded off to three significant figures.

Table 32. Compare modes

Compare modes	Function
Mode 0	Interrupt-only compare mode; several compare interrupts per timer period are possible
Mode 1	Pin toggles on each compare match; several compare events per timer period are possible
Mode 2	Interrupt-only compare mode; only one compare interrupt per timer period is generated
Mode 3	Pin set ‘1’ on match; pin reset ‘0’ on compare time overflow; only one compare event per timer period is generated
Double register mode	Two registers operate on one pin; pin toggles on each compare match; several compare events per timer period are possible.

Table 33. CAPCOM timer input frequencies, resolutions and periods at 40 MHz

f_{CPU} = 40 MHz	Timer input selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Prescaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input frequency	5 MHz	2.5 MHz	1.25 MHz	625 kHz	312.5 kHz	156.25 kHz	78.125 kHz	39.1 kHz
Resolution	200 ns	400 ns	0.8 µs	1.6 µs	3.2 µs	6.4 µs	12.8 µs	25.6 µs
Period	13.1 ms	26.2 ms	52.4 ms	104.8 ms	209.7 ms	419.4 ms	838.9 ms	1.678 s

11 General purpose timer unit

The GPT unit is a flexible multifunctional timer/counter structure which is used for time related tasks such as event timing and counting, pulse width and duty cycle measurements, pulse generation, or pulse multiplication. The GPT unit contains five 16-bit timers organized into two separate modules GPT1 and GPT2. Each timer in each module may operate independently in several different modes, or may be concatenated with another timer of the same module.

11.1 GPT1

Each of the three timers T2, T3, T4 of the GPT1 module can be configured individually for one of four basic modes of operation: timer, gated timer, counter mode and incremental interface mode.

In timer mode, the input clock for a timer is derived from the CPU clock, divided by a programmable prescaler.

In counter mode, the timer is clocked in reference to external events.

Pulse width or duty cycle measurement is supported in gated timer mode where the operation of a timer is controlled by the ‘gate’ level on an external input pin. For these purposes, each timer has one associated port pin (TxIN) which serves as gate or clock input.

Table 34 lists the timer input frequencies, resolution and periods for each prescaler option at 40 MHz CPU clock.

In incremental interface mode, the GPT1 timers (T2, T3, T4) can be directly connected to the incremental position sensor signals A and B by their respective inputs TxIN and TxEUD.

Direction and count signals are internally derived from these two input signals so that the contents of the respective timer Tx corresponds to the sensor position. The third position sensor signal TOP0 can be connected to an interrupt input.

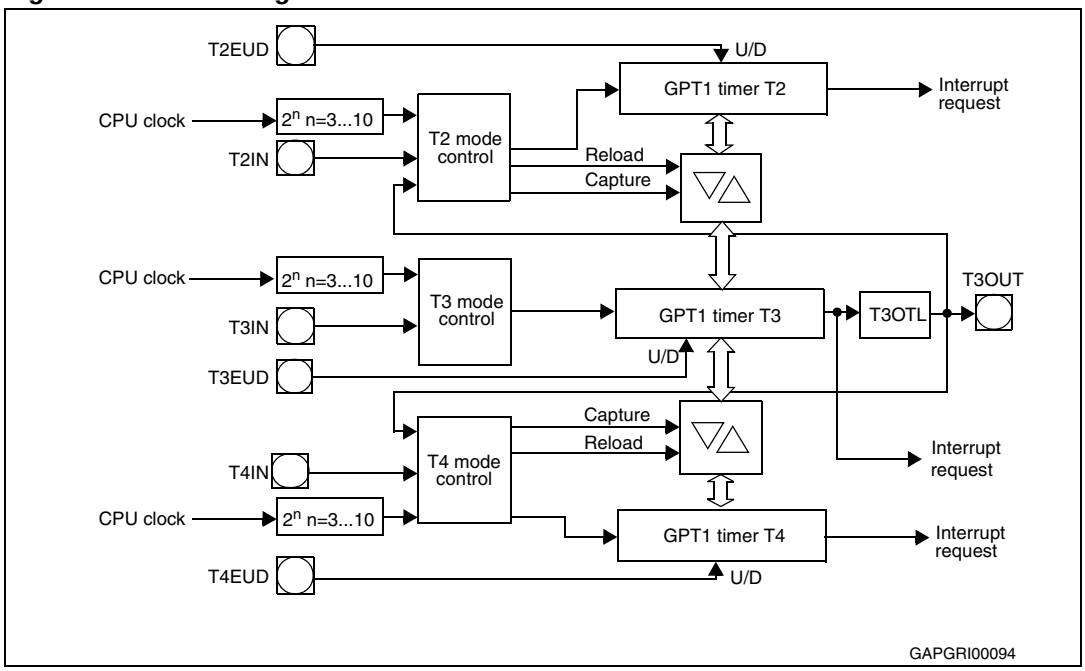
Timer T3 has output toggle latches (TxOTL) which changes state on each timer over flow / underflow. The state of this latch may be output on port pins (TxOUT) for time out monitoring of external hardware components, or may be used internally to clock timers T2 and T4 for high resolution of long duration measurements.

In addition to their basic operating modes, timers T2 and T4 may be configured as reload or capture registers for timer T3.

Table 34. GPT1 timer input frequencies, resolutions and periods at 40 MHz

f_{CPU} = 40 MHz	Timer input selection T2I / T3I / T4I							
	000b	001b	010b	011b	100b	101b	110b	111b
Prescaler factor	8	16	32	64	128	256	512	1024
Input frequency	5 MHz	2.5 MHz	1.25 MHz	625 kHz	312.5 kHz	156.25 kHz	78.125 kHz	39.1 kHz
Resolution	200 ns	400 ns	0.8 µs	1.6 µs	3.2 µs	6.4 µs	12.8 µs	25.6 µs
Period maximum	13.1 ms	26.2 ms	52.4 ms	104.8 ms	209.7 ms	419.4 ms	838.9 ms	1.678 s

Figure 10. Block diagram of GPT1



11.2 GPT2

The GPT2 module provides precise event control and time measurement. It includes two timers (T5, T6) and a capture/reload register (CAPREL). Both timers can be clocked with an input clock which is derived from the CPU clock via a programmable prescaler or with external signals. The count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD). Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6 which changes its state on each timer overflow/underflow.

The state of this latch may be used to clock timer T5, or it may be output on a port pin (T6OUT). The overflow / underflow of timer T6 can additionally be used to clock the CAPCOM timers T0 or T1, and to cause a reload from the CAPREL register. The CAPREL register may capture the contents of timer T5 based on an external signal transition on the corresponding port pin (CAPIN), and timer T5 may optionally be cleared after the capture procedure. This allows absolute time differences to be measured or pulse multiplication to be performed without software overhead.

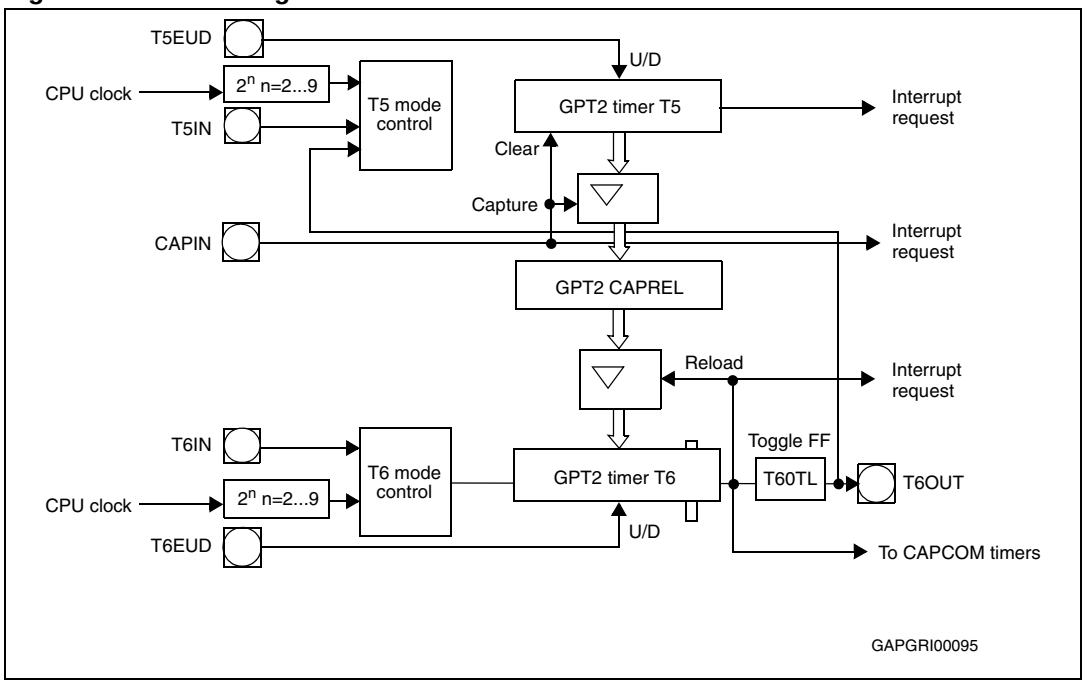
The capture trigger (timer T5 to CAPREL) may also be generated upon transitions of GPT1 timer T3 inputs T3IN and/or T3EUD. This is advantageous when T3 operates in Incremental Interface mode.

Table 35 lists the timer input frequencies, resolution and periods for each prescaler option at 40 MHz CPU clock.

Table 35. GPT2 timer input frequencies, resolutions and periods at 40 MHz

f_{CPU} = 40 MHz	Timer input selection T5I/T6I							
	000b	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
Prescaler factor	4	8	16	32	64	128	256	512
Input frequency	10 MHz	5 MHz	2.5 MHz	1.25 MHz	625 kHz	312.5 kHz	156.25 kHz	78.125 kHz
Resolution	100 ns	200 ns	400 ns	0.8 µs	1.6 µs	3.2 µs	6.4 µs	12.8 µs
Period maximum	6.55 ms	13.1 ms	26.2 ms	52.4 ms	104.8 ms	209.7 ms	419.4 ms	838.9 ms

Figure 11. Block diagram of GPT2



12 PWM modules

Two pulse width modulation modules are available on ST10F272M: standard PWM0 and XBUS PWM1. They can generate up to four PWM output signals each, using edge-aligned or center-aligned PWM. In addition, the PWM modules can generate PWM burst signals and single shot outputs. *Table 36* shows the PWM frequencies for different resolutions. The level of the output signals is selectable and the PWM modules can generate interrupt requests.

Figure 12. Block diagram of PWM module

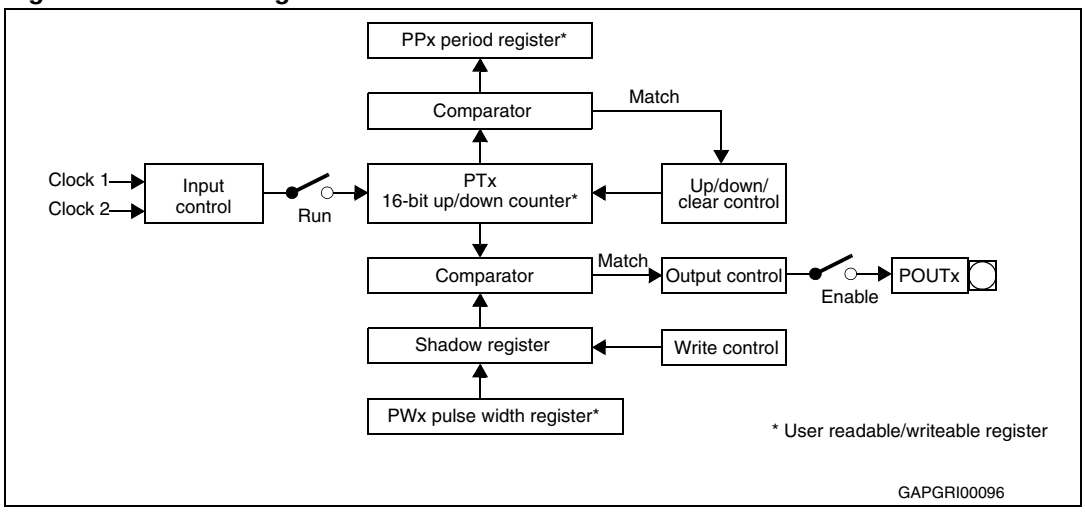


Table 36. PWM unit frequencies and resolutions at 40 MHz CPU clock

Mode 0	Resolution	8-bit	10-bit	12-bit	14-bit	16-bit
CPU clock/1	25 ns	156.25 kHz	39.1 kHz	9.77 kHz	2.44 Hz	610 Hz
CPU clock/64	1.6 µs	2.44 kHz	610Hz	152.6 Hz	38.15 Hz	9.54 Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU clock/1	25 ns	78.12 kHz	19.53 kHz	4.88 kHz	1.22 kHz	305.2 Hz
CPU clock/64	1.6 µs	1.22 kHz	305.17 Hz	76.29 Hz	19.07 Hz	4.77 Hz

13 Parallel ports

13.1 Introduction

The ST10F272M MCU provides up to 111 I/O lines with programmable features. These capabilities permit this MCU to be adapted to a wide range of applications.

ST10F272M has nine groups of I/O lines gathered as follows:

- Port 0 is a two time 8-bit port named P0L (low as less significant byte) and P0H (high as most significant byte)
- Port 1 is a two time 8-bit port named P1L and P1H
- Port 2 is a 16-bit port
- Port 3 is a 15-bit port (P3.14 line is not implemented)
- Port 4 is an 8-bit port
- Port 5 is a 16-bit port input only
- Port 6, Port 7 and Port 8 are 8-bit ports

These ports may be used as general purpose bidirectional input or output, software controlled with dedicated registers.

For example, the output drivers of six of the ports (2, 3, 4, 6, 7, 8) can be configured (bitwise) for push-pull or open drain operation using ODPx registers.

The input threshold levels are programmable (TTL/CMOS) for all the ports. The logic level of a pin is clocked into the input latch once per state time, regardless whether the port is configured for input or output. The threshold is selected with PICON and XPICON registers control bits.

A write operation to a port pin configured as an input causes the value to be written into the port output latch, while a read operation returns the latched state of the pin itself. A read-modify-write operation reads the value of the pin, modifies it, and writes it back to the output latch.

Writing to a pin configured as an output (DPx.y = ‘1’) causes the output latch and the pin to have the written value, since the output buffer is enabled. Reading this pin returns the value of the output latch. A read-modify-write operation reads the value of the output latch, modifies it, and writes it back to the output latch, thus also modifying the level at the pin.

I/O lines support an alternate function which is detailed in the following description of each port.

13.2 I/O’s special features

13.2.1 Open drain mode

Some of the I/O ports of ST10F272M support the open drain capability. This programmable feature may be used with an external pull-up resistor, in order to provide an AND wired logical function.

This feature is implemented for ports P2, P3, P4, P6, P7 and P8 (see respective sections), and is controlled through the respective open drain control registers ODPx.

13.2.2 Input threshold control

The standard inputs of the ST10F272M determine the status of input signals according to TTL levels. In order to accept and recognize noisy signals, CMOS input thresholds can be selected instead of the standard TTL thresholds for all the pins. These CMOS thresholds are defined above the TTL thresholds and feature a higher hysteresis to prevent the inputs from toggling while the respective input signal level is near the thresholds.

The port input control registers PICON and XPICON are used to select these thresholds for each byte of the indicated ports, this means the 8-bit ports P0L, P0H, P1L, P1H, P4, P7 and P8 are controlled by one bit each while ports P2, P3 and P5 are controlled by two bits each.

All options for individual direction and output mode control are available for each pin, independent of the selected input threshold.

13.3 Alternate port functions

Each port line has one associated programmable alternate input or output function.

- Port0 and port1 may be used as address and data lines when accessing external memory. Additionally, port1 provides:
  - Input capture lines
  - 8 additional analog input channels to the A/D converter
- Port 2, port 7 and port 8 are associated with the capture inputs or compare outputs of the CAPCOM units and/or with the outputs of the PWM0 module, of the PWM1 module and of the ASC1.
  Port 2 is also used for fast external interrupt inputs and for timer 7 input.
- Port 3 includes the alternate functions of timers, serial interfaces, the optional bus control signal $\overline{BHE}$ and the system clock output (CLKOUT).
- Port 4 outputs the additional segment address bit A23...A16 in systems where more than 64 Kbytes of memory are to be access directly. In addition, CAN1, CAN2 and I^2C lines are provided.
- Port 5 is used as analog input channels of the A/D converter or as timer control signals.
- Port 6 provides optional bus arbitration signals ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$) and chip select signals and the SSC1 lines.

If the alternate output function of a pin is to be used, the direction of this pin must be programmed for output (DPx.y = ‘1’), except for some signals that are used directly after reset and are configured automatically. Otherwise the pin remains in the high-impedance state and is not effected by the alternate output function. The respective port latch should hold a ‘1’, because its output is ANDed with the alternate output data (except for PWM output signals).

If the alternate input function of a pin is used, the direction of the pin must be programmed for input (DPx.y = ‘0’) if an external device is driving the pin. The input direction is the default after reset. If no external device is connected to the pin, however, the direction for this pin can also be set to output. In this case, the pin reflects the state of the port output latch. Thus, the alternate input function reads the value stored in the port output latch. This can be used for testing purposes to allow a software trigger of an alternate input function by writing to the port output latch.

On most of the port lines, the user software is responsible for setting the proper direction when using an alternate input or output function of a pin.

This is done by setting or clearing the direction control bit DPx.y of the pin before enabling the alternate function.

There are port lines, however, where the direction of the port line is switched automatically.

For instance, in the multiplexed external bus modes of port0, the direction must be switched several times for an instruction fetch in order to output the addresses and to input the data.

Obviously, this cannot be done through instructions. In these cases, the direction of the port line is switched automatically by hardware if the alternate function of such a pin is enabled.

To determine the appropriate level of the port output latches, check how the alternate data output is combined with the respective port latch output.

There is one basic structure for all port lines with only an alternate input function. Port lines with only an alternate output function, however, have different structures due to the way the direction of the pin is switched and depending on whether the pin is accessible by the user software or not in the alternate function mode.

All port lines that are not used for these alternate functions may be used as general purpose I/O lines.

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.

- **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 register. The data can be transferred to the RAM by interrupt software management or using the PEC data transfer.
- **Wait for ADDAT read mode**: When using continuous modes, in order to avoid to overwrite the result of the current conversion by the next one, the ADWR bit of ADCON control register must be activated. Then, until the ADDAT register is read, the new result is stored in a temporary buffer and the conversion is on hold.
- **Channel injection mode**: When using continuous modes, a selected channel can be converted in between without changing the current operating mode. The 10-bit data of the conversion are stored in ADRES field of ADDAT2. The current continuous mode remains active after the single conversion is completed.

A full calibration sequence is performed after a reset. This full calibration lasts up to 40630 CPU clock cycles. During this time, the busy flag ADBSY is set to indicate the operation. It compensates the capacitance mismatch, so the calibration procedure does not need any update during normal operation.

No conversion can be performed during this time: The bit ADBSY has to be polled to verify that the calibration is over, and the module is able to start a conversion.

15 Serial channels

Serial communication with other microcontrollers, microprocessors, terminals or external peripheral components is provided by up to four serial interfaces: Two asynchronous/ synchronous serial channels (ASC0 and ASC1) and two high-speed synchronous serial channel (SSC0 and SSC1). Dedicated baudrate generators set up all standard baudrates without the requirement of oscillator tuning. For transmission, reception and erroneous reception, separate interrupt vectors are provided for ASC0 and SSC0 serial channel. A more complex mechanism of interrupt sources multiplexing is implemented for ASC1 and SSC1 (XBUS mapped).

15.1 Asynchronous/synchronous serial interfaces

The asynchronous / synchronous serial interfaces (ASC0 and ASC1) provides serial communication between the ST10F272M and other microcontrollers, microprocessors or external peripherals.

15.2 ASCx in asynchronous mode

In asynchronous mode, 8- or 9-bit data transfer, parity generation and the number of stop bits can be selected. Parity framing and overrun error detection is provided to increase the reliability of data transfers. Transmission and reception of data is double-buffered. Full-duplex communication up to 1.25 Mbaud (at 40 MHz of f_{CPU}) is supported in this mode.

Table 37. ASC asynchronous baudrates by reload value and deviation errors (f_{CPU} = 40 MHz)

S0BRS = '0', f_{CPU} = 40 MHz			S0BRS = '1', f_{CPU} = 40 MHz		
Baudrate (baud)	Deviation error	Reload value (hex)	Baudrate (baud)	Deviation error	Reload value (hex)
1 250 000	0.0%/0.0%	0000/0000	833 333	0.0%/0.0%	0000/0000
112 000	+1.5%/-7.0%	000A/000B	112 000	+6.3%/-7.0%	0006/0007
56 000	+1.5%/-3.0%	0015/0016	56 000	+6.3%/-0.8%	000D/000E
38 400	+1.7%/-1.4%	001F/0020	38 400	+3.3%/-1.4%	0014/0015
19 200	+0.2%/-1.4%	0040/0041	19 200	+0.9%/-1.4%	002A/002B
9 600	+0.2%/-0.6%	0081/0082	9 600	+0.9%/-0.2%	0055/0056
4 800	+0.2%/-0.2%	0103/0104	4 800	+0.4%/-0.2%	00AC/00AD
2 400	+0.2%/0.0%	0207/0208	2 400	+0.1%/-0.2%	015A/015B
1 200	0.1%/0.0%	0410/0411	1 200	+0.1%/-0.1%	02B5/02B6
600	0.0%/0.0%	0822/0823	600	+0.1%/0.0%	056B/056C
300	0.0%/0.0%	1045/1046	300	0.0%/0.0%	0AD8/0AD9
153	0.0%/0.0%	1FE8/1FE9	102	0.0%/0.0%	1FE8/1FE9

Note: The deviation errors given in Table 37 are rounded off. To avoid deviation errors use a baudrate crystal (providing a multiple of the ASC0 sampling frequency).

15.3 ASCx in synchronous mode

In synchronous mode, data is transmitted or received synchronously to a shift clock which is generated by the ST10F272M. Half-duplex communication up to 5 Mbaud (at 40 MHz of f_{CPU}) is possible in this mode.

Table 38. ASC synchronous baudrates by reload value and deviation errors (f_{CPU} = 40 MHz)

S0BRS = '0', f_{CPU} = 40 MHz			S0BRS = '1', f_{CPU} = 40 MHz		
Baudrate (baud)	Deviation error	Reload value (hex)	Baudrate (baud)	Deviation error	Reload value (hex)
5 000 000	0.0%/0.0%	0000/0000	3 333 333	0.0%/0.0%	0000/0000
112 000	+1.5%/-0.8%	002B/002C	112 000	+2.6%/-0.8%	001C/001D
56 000	+0.3%/-0.8%	0058/0059	56 000	+0.9%/-0.8%	003A/003B
38 400	+0.2%/-0.6%	0081/0082	38 400	+0.9%/-0.2%	0055/0056
19 200	+0.2%/-0.2%	0103/0104	19 200	+0.4%/-0.2%	00AC/00AD
9 600	+0.2%/0.0%	0207/0208	9 600	+0.1%/-0.2%	015A/015B
4 800	+0.1%/0.0%	0410/0411	4 800	+0.1%/-0.1%	02B5/02B6
2 400	0.0%/0.0%	0822/0823	2 400	+0.1%/0.0%	056B/056C
1 200	0.0%/0.0%	1045/1046	1 200	0.0%/0.0%	0AD8/0AD9
900	0.0%/0.0%	15B2/15B3	600	0.0%/0.0%	15B2/15B3
612	0.0%/0.0%	1FE8/1FE9	407	0.0%/0.0%	1FFD/1FFE

Note: The deviation errors given in the Table 38 are rounded off. To avoid deviation errors use a baudrate crystal (providing a multiple of the ASC0 sampling frequency).

15.4 High speed synchronous serial interfaces

The high-speed synchronous serial interfaces (SSC0 and SSC1) provides flexible high-speed serial communication between the ST10F272M and other microcontrollers, microprocessors or external peripherals.

The SSCx supports full-duplex and half-duplex synchronous communication. The serial clock signal can be generated by the SSCx itself (master mode) or be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable.

This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A 16-bit baudrate generator provides the SSCx with a separate serial clock signal. The serial channel SSCx has its own dedicated 16-bit baudrate generator with 16-bit reload capability, allowing baudrate generation independent from the timers.

Table 39 lists some possible baudrates against the required reload values and the resulting bit times for the 40 MHz CPU clock. The maximum is limited to 8 Mbaud.

Table 39. Synchronous baudrate and reload values (f_{CPU} = 40 MHz)

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

16 I²C interface

The integrated I²C bus module handles the transmission and reception of frames over the two-line SDA/SCL in accordance with the I²C Bus specification. The I²C module can operate in slave mode, in master mode or in multi-master mode. It can receive and transmit data using 7-bit or 10-bit addressing. Data can be transferred at speeds up to 400 Kbit/s (both standard and fast I²C bus modes are supported).

The module can generate three different types of interrupt:

- requests related to bus events, such as start or stop events, or arbitration lost
- requests related to data transmission
- requests related to data reception

These requests are issued to the interrupt controller by three different lines, and identified as error, transmit, and receive interrupt lines.

When the I²C module is enabled by setting bit XI2CEN in XPERCON register, pins P4.4 and P4.7 (where SCL and SDA are respectively mapped as alternate functions) are automatically configured as bidirectional open-drain: the value of the external pull-up resistor depends on the application. P4, DP4 and ODP4 cannot influence the pin configuration.

When the I²C cell is disabled (clearing bit XI2CEN), P4.4 and P4.7 pins are standard I/ O controlled by P4, DP4 and ODP4.

The speed of the I²C interface can be selected between standard mode (0 to 100 kHz) and fast I²C mode (100 to 400 kHz).

17 CAN modules

The two integrated CAN modules (CAN1 and CAN2) are identical and handle the completely autonomous transmission and reception of CAN frames according to the CAN specification V2.0 part B (active). It is based on the C-CAN specification.

Each on-chip CAN module can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

Because of duplication of the CAN controllers, the following adjustments are to be considered:

- Same internal register addresses of both CAN controllers, but with base addresses differing in address bit A8; separate chip select for each CAN module. Refer to *Chapter 4: Memory organization on page 21*.
- The CAN1 transmit line (CAN1_TxD) is the alternate function of the Port P4.6 pin and the receive line (CAN1_RxD) is the alternate function of the Port P4.5 pin.
- The CAN2 transmit line (CAN2_TxD) is the alternate function of the Port P4.7 pin and the receive line (CAN2_RxD) is the alternate function of the Port P4.4 pin.
- Interrupt request lines of the CAN1 and CAN2 modules are connected to the XBUS interrupt lines together with other X-peripherals sharing the four vectors.
- The CAN modules must be selected with corresponding CANxEN bit of XPERCON register before the bit XPEN of SYSCON register is set.
- The reset default configuration is: CAN1 enabled, CAN2 disabled.

Note: If one or both CAN modules is used, port 4 cannot be programmed to output all eight segment address lines. Thus, only four segment address lines can be used, reducing the external memory space to 5 Mbytes (1 Mbyte per $\overline{CS}$ line).

17.1 Configuration support

It is possible that both CAN controllers are working on the same CAN bus, supporting together up to 64 message objects. In this configuration, both receive signals and both transmit signals are linked together when using the same CAN transceiver. This configuration is especially supported by providing open drain outputs for the CAN1_Txd and CAN2_TxD signals. The open drain function is controlled with the ODP4 register for port P4: in this way it is possible to connect together P4.4 with P4.5 (receive lines) and P4.6 with P4.7 (transmit lines configured to be configured as open-drain).

The user may also internally map both CAN modules on the same pins P4.5 and P4.6. In this way, P4.4 and P4.7 can be used either as general purpose I/O lines, or used for I^2C interface. This is possible by setting bit CANPAR of the XMISC register. To access this register it is necessary to set bit XMISCEN of the XPERCON register and bit XPEN of the SYSCON register.

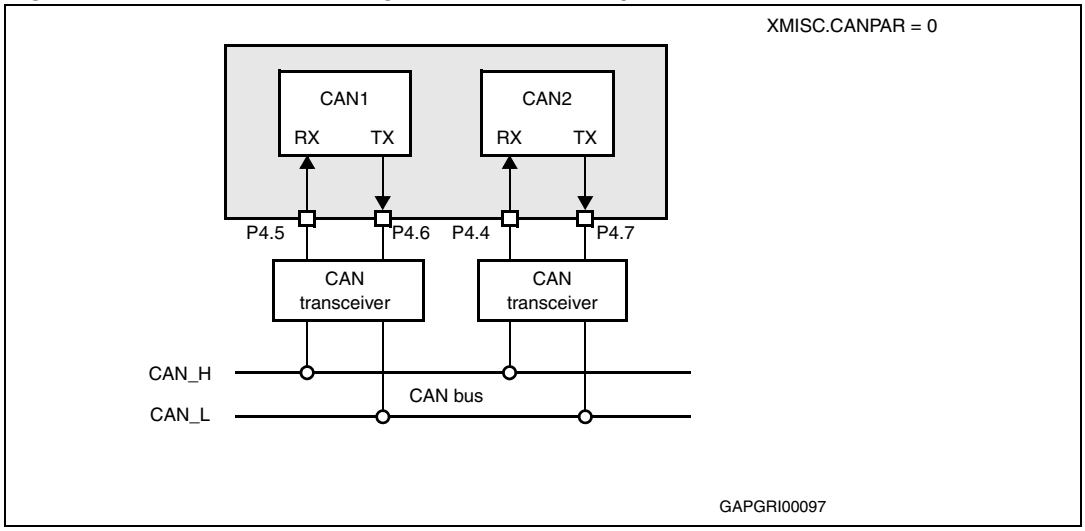
17.2 CAN bus configurations

Depending on the application, CAN bus configuration may be one single bus with a single or multiple interfaces or a multiple bus with a single or multiple interfaces. The ST10F272M can support both configurations.

17.2.1 Single CAN bus

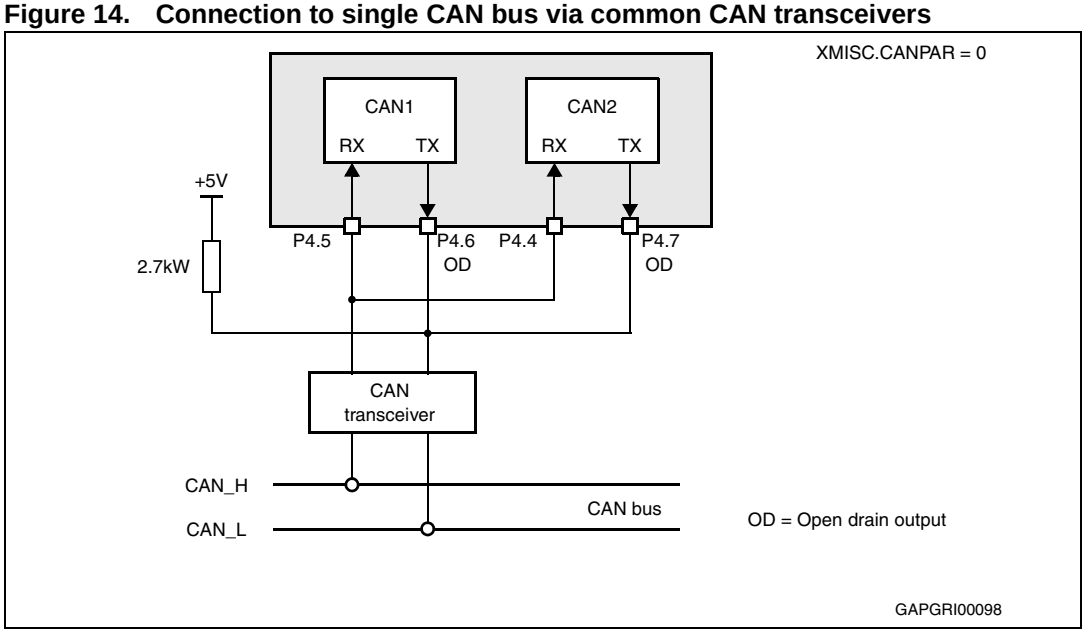
The single CAN bus multiple interfaces configuration may be implemented using two CAN transceivers as shown in *Figure 13*.

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



The ST10F272M also supports single CAN bus multiple (dual) interfaces using the open drain option of the CANx_TxD output as shown in *Figure 14*. 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.

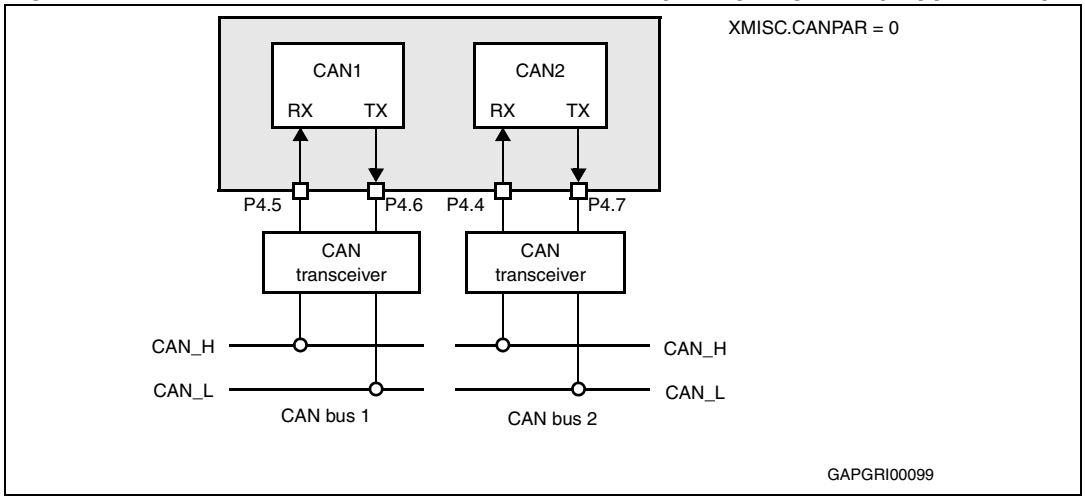
Figure 14. Connection to single CAN bus via common CAN transceivers



17.2.2 Multiple CAN bus

The ST10F272M provides two CAN interfaces to support the kind of bus configuration in *Figure 15*.

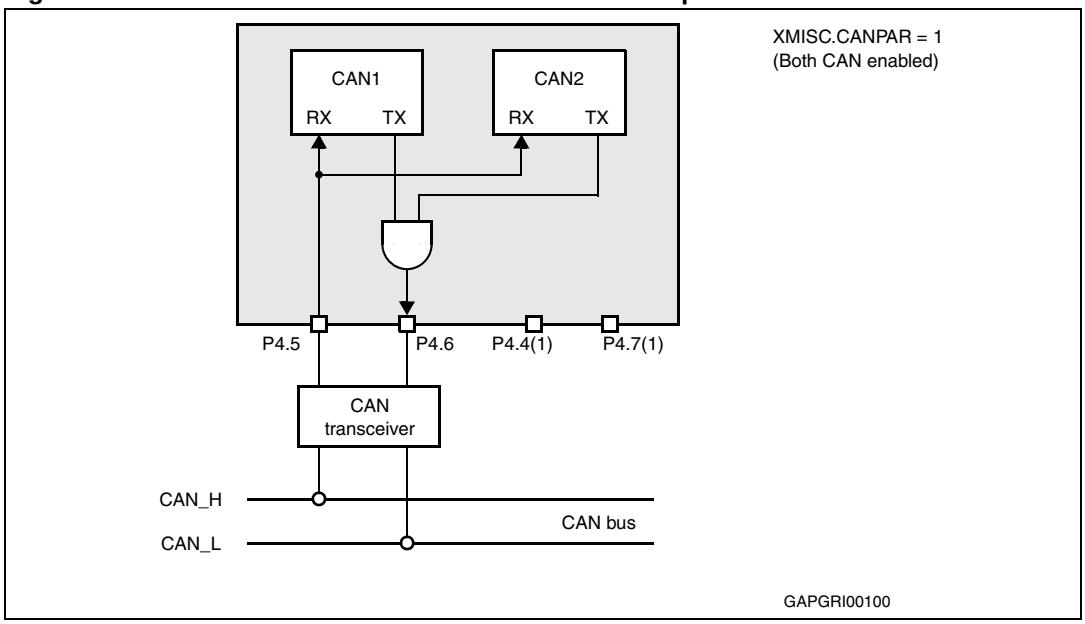
Figure 15. Connection to two different CAN buses (example, gateway application)



17.2.3 Parallel mode

In addition to previous configurations, a parallel mode is supported. This is shown in *Figure 16*.

Figure 16. Connection to one CAN bus with internal parallel mode enabled



1. P4.4 and P4.7 when not used as CAN functions can be used as general purpose I/O while they cannot be used as external bus address lines.

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).

19 Watchdog timer

The watchdog timer is a fail-safe mechanism which prevents the microcontroller from malfunctioning for long periods of time.

The watchdog timer is always enabled after a reset of the chip and can only be disabled in the time interval until the EINIT (end of initialization) instruction has been executed.

Therefore, the chip start-up procedure is always monitored. The software must be designed to service the watchdog timer before it overflows. If, due to hardware or software related failures, the software fails to do so, the watchdog timer overflows and generates an internal hardware reset. It pulls the $\overline{\text{RSTOUT}}$ pin low in order to allow external hardware components to be reset.

Each of the different reset sources is indicated in the WDTCON register:

- Watchdog timer reset in case of an overflow
- Software reset in case of execution of the SRST instruction
- Short, long and power-on reset in case of hardware reset (and depending of reset pulse duration and RPD pin configuration)

The indicated bits are cleared with the EINIT instruction. The source of the reset can be identified during the initialization phase.

The watchdog timer is 16-bit, clocked with the system clock divided by 2 or 128. The high byte of the watchdog timer register can be set to a prespecified reload value (stored in WDTREL).

Each time it is serviced by the application software, the high byte of the watchdog timer is reloaded. For security, rewrite WDTCON each time before the watchdog timer is serviced

Table 40 shows the watchdog time range for 40 MHz CPU clock.

Table 40. WDTREL reload value (f_{CPU} = 40 MHz)

Reload value in WDTREL	Prescaler for f_{CPU} = 40 MHz	
	2 (WDTIN = '0')	128 (WDTIN = '1')
FFh	12.8 µs	819.2 µs
00h	3.277 ms	209.7 ms

20 System reset

System reset initializes the MCU in a predefined state. There are six ways to activate a reset state. The system start-up configuration is different for each case as shown in *Table 41*.

Table 41. Reset event definition

Reset source	Flag	RPD status	Conditions
Power-on reset	PONR	Low	Power-on
Asynchronous hardware reset	LHWR	Low	$t_{\overline{RSTIN}}$ >(1)
Synchronous long hardware reset		High	$t_{\overline{RSTIN}}$ > (1032 + 12) TCL + max(4 TCL, 500ns)(2)
Synchronous short hardware reset	SHWR	High	$t_{\overline{RSTIN}}$ > max(4 TCL, 500ns)(2) $t_{\overline{RSTIN}}$ ≤ (1032 + 12) TCL + max(4 TCL, 500ns)(2)
Watchdog timer reset	WDTR	(3)	WDT overflow
Software reset	SWR	(3)	SRST instruction execution

1. $\overline{RSTIN}$ pulse should be longer than 500 ns (filter) and than settling time for configuration of Port0.
2. See next *Section 20.1* for more details on minimum reset pulse duration.
3. The RPD status has no influence unless bidirectional reset is activated (bit BDRSTEN in SYSCON): RPD low inhibits the bidirectional reset on SW and WDT reset events, that is $\overline{RSTIN}$ is not activated (refer to Sections *20.4*, *20.5* and *20.6*).

The figures in the upcoming sections *20.2*, *20.3*, *20.5* and *20.6* use the following terminology:

- Transparent: Level of the pin affects the internal reset logic
- Not transparent: Level of the pin does not affect internal logic

20.1 Input filter

On the $\overline{RSTIN}$ input pin an on-chip RC filter is implemented. It is sized to filter all spikes shorter than 50ns. On the other hand, a valid pulse longer than 500 ns is required for the ST10 to recognize a reset command. In between 50 ns and 500 ns a pulse can either be filtered or recognized as valid, depending on the operating conditions and process variations.

For this reason all minimum durations mentioned in this chapter for the different kinds of reset events must be carefully evaluated, taking into account the above requirements.

In particular, for short hardware reset, where only 4 TCL is specified as minimum input reset pulse duration, the operating frequency is a key factor.

Examples:

- For a CPU clock of 40 MHz, 4 TCL is 50 ns, so it would be filtered. In this case the minimum becomes the one imposed by the filter (that is 500 ns).
- For a CPU clock of 4 MHz, 4 TCL is 500 ns. In this case the minimum from the formula is coherent with the limit imposed by the filter.

20.2 Asynchronous reset

An asynchronous reset is triggered when $\overline{\text{RSTIN}}$ pin is pulled low while RPD pin is at low level. Then the ST10F272M is immediately (after the input filter delay) forced in reset default state. It pulls low $\overline{\text{RSTOUT}}$ pin, it cancels pending internal hold states if any, it aborts all internal/external bus cycles, it switches buses (data, address and control signals) and I/O pin drivers to high-impedance, it pulls high port0 pins.

Note: If an asynchronous reset occurs during a read or write phase in internal memories, the content of the memory itself could be corrupted: to avoid this, synchronous reset usage is strongly recommended.

Power-on reset

The asynchronous reset must be used during the power-on of the device. Depending on crystal or resonator frequency, the on-chip oscillator needs about 1ms to 10 ms to stabilize (refer to *Section 24: Electrical characteristics*), with an already stable V_{DD}. The logic of the ST10F272M does not need a stabilized clock signal to detect an asynchronous reset, so it is suitable for power-on conditions. To ensure a proper reset sequence, the $\overline{\text{RSTIN}}$ pin and the RPD pin must be held at low level until the device clock signal is stabilized and the system configuration value on port0 is settled.

At power-on it is important to respect some additional constraints introduced by the start-up phase of the different embedded modules.

In particular, the on-chip voltage regulator needs at least 1ms to stabilize the internal 1.8 V for the core logic: This time is computed from when the external reference (V_{DD}) becomes stable (inside specification range, that is, at least 4.5 V). This is a constraint for the application hardware (external voltage regulator): The $\overline{\text{RSTIN}}$ pin assertion has to be extended to guarantee the voltage regulator stabilization.

A second constraint is imposed by the embedded Flash. When booting from internal memory, starting from $\overline{\text{RSTIN}}$ releasing, it needs a maximum of 1ms for its initialization: before that, the internal reset (RST signal) is not released, so the CPU does not start code execution in internal memory.

Note: This is not true if external memory is used (pin $\overline{EA}$ held low during reset phase). In this case, once the $\overline{RSTIN}$ pin is released, and after a few CPU clock cycles (Filter delay plus 3...8 TCL), the internal reset signal RST is released as well, so the code execution can start immediately after. Obviously, an eventual access to the data in internal Flash is forbidden before its initialization phase is completed: An eventual access during starting phase will return FFFFh (just at the beginning), while later 009Bh (an illegal opcode trap can be generated).

At power-on, the $\overline{\text{RSTIN}}$ pin must be tied low for a minimum time that also includes the start-up time of the main oscillator (t_{STUP} = 1ms for resonator, 10ms for crystal) and PLL synchronization time (t_{PSUP} = 200 µs): This means if the internal Flash is used, the $\overline{\text{RSTIN}}$ pin could be released before the main oscillator and PLL are stable to recover some time in the start-up phase (Flash initialization only needs stable V_{18}, but does not need stable system clock since an internal dedicated oscillator is used).

Warning: **It is recommended to provide the external hardware with a current limitation circuitry. This is necessary to avoid permanent damage of the device during the power-on transient, when the capacitance on V_{18} pin is charged. For the on-chip voltage regulator functionality 10nF is sufficient: In any case, a maximum of 100 nF on V_{18} pin should not generate problems of over-current (higher value is allowed if current is limited by the external hardware). External current limitation is nevertheless also recommended to avoid risks of damage in case of a temporary short between V_{18} and ground: The internal 1.8 V drivers are sized to drive currents of several tens of Amps, so the current must be limited by the external hardware. The limit of current is imposed by power dissipation considerations (refer to *Section 24: Electrical characteristics*).**

In figures *17* and *18* below, asynchronous power-on timing diagrams are shown, with boot from internal or external memory respectively, highlighting the reset phase extension introduced by the embedded Flash module when selected.

Caution: Never power the device without keeping the RSTIN pin grounded: The device could enter into unpredictable states, risking also permanent damage.

Figure 17. Asynchronous power-on reset ($\overline{EA}$ = 1)

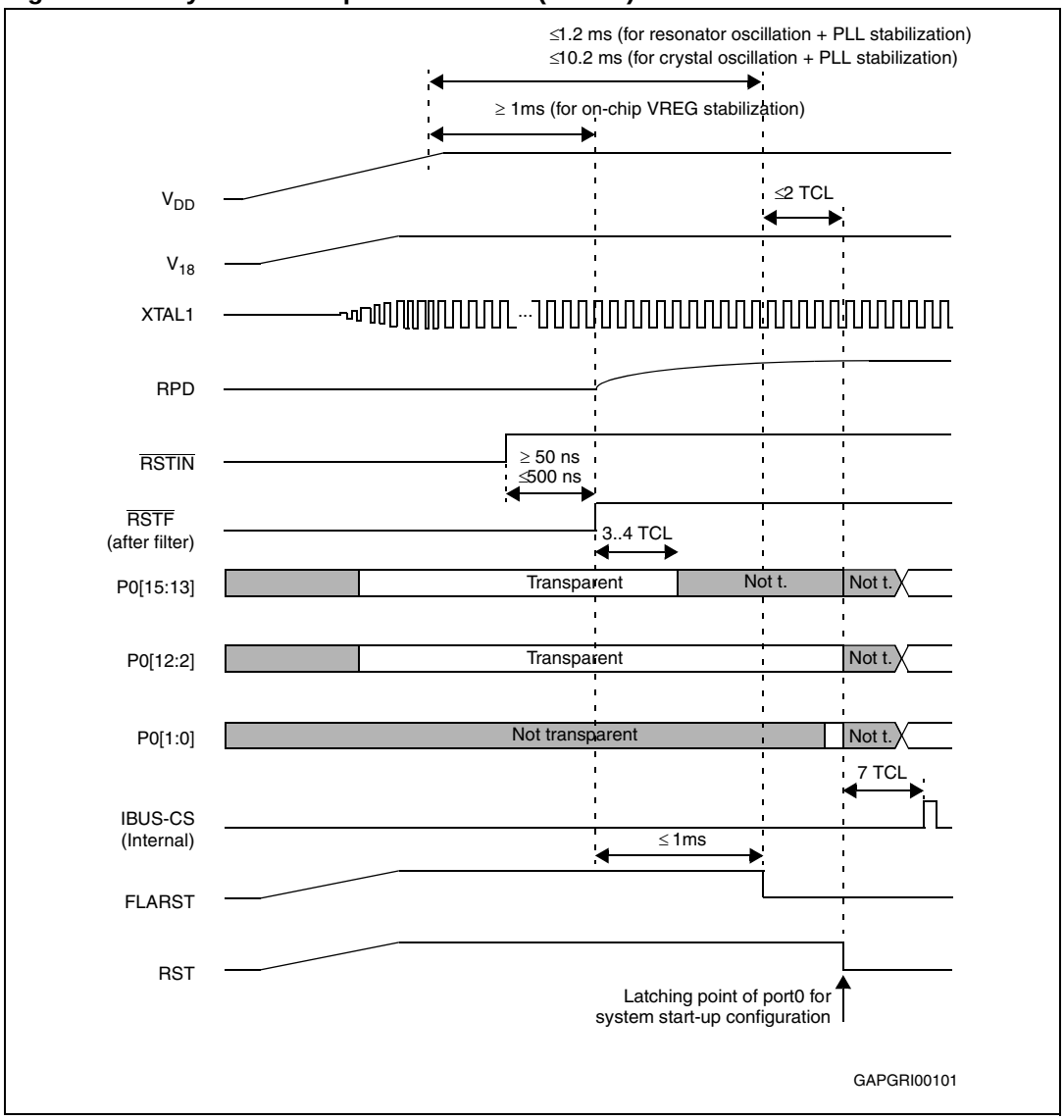
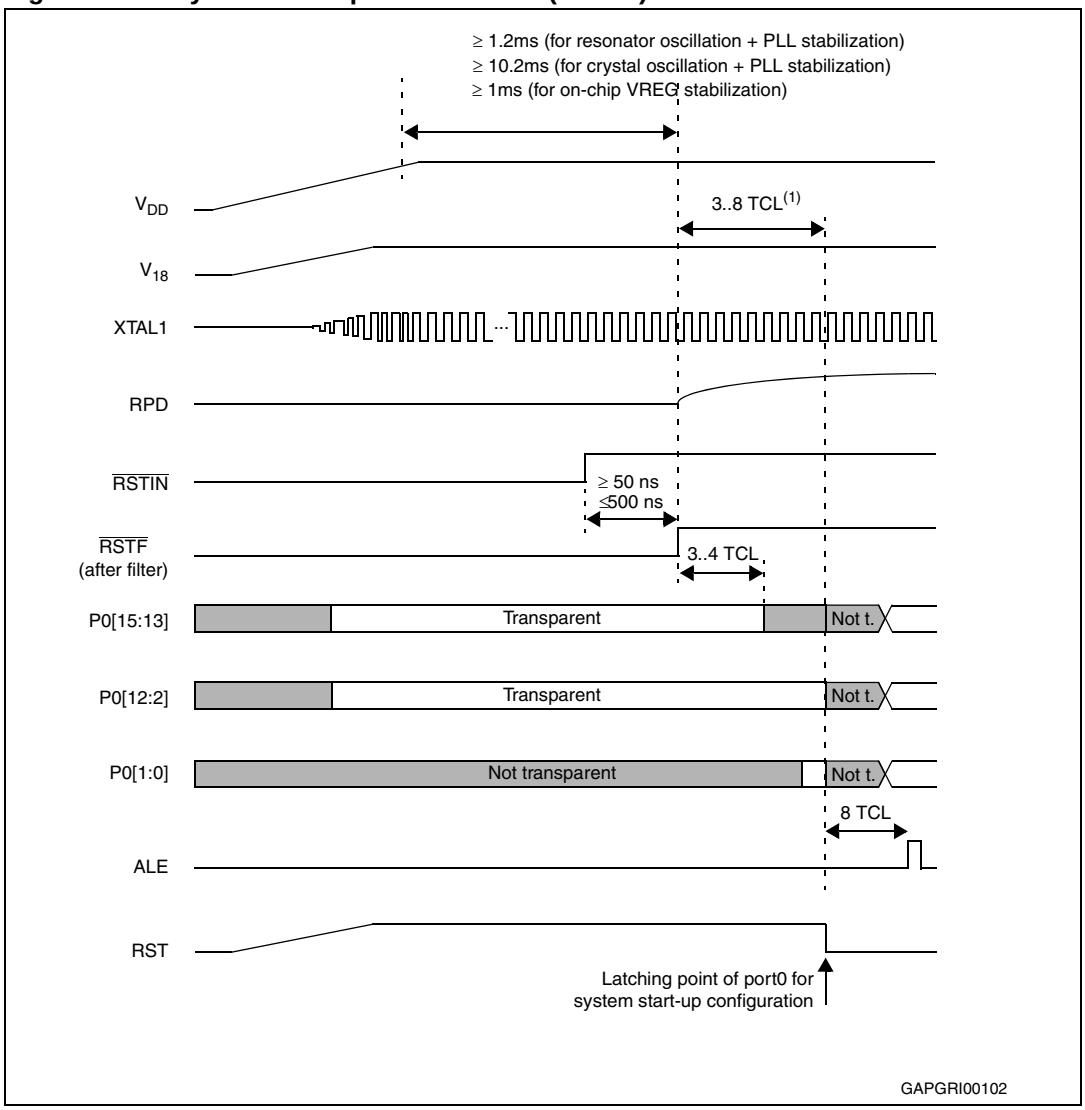


Figure 18. Asynchronous power-on reset ($\overline{EA}$ = 0)

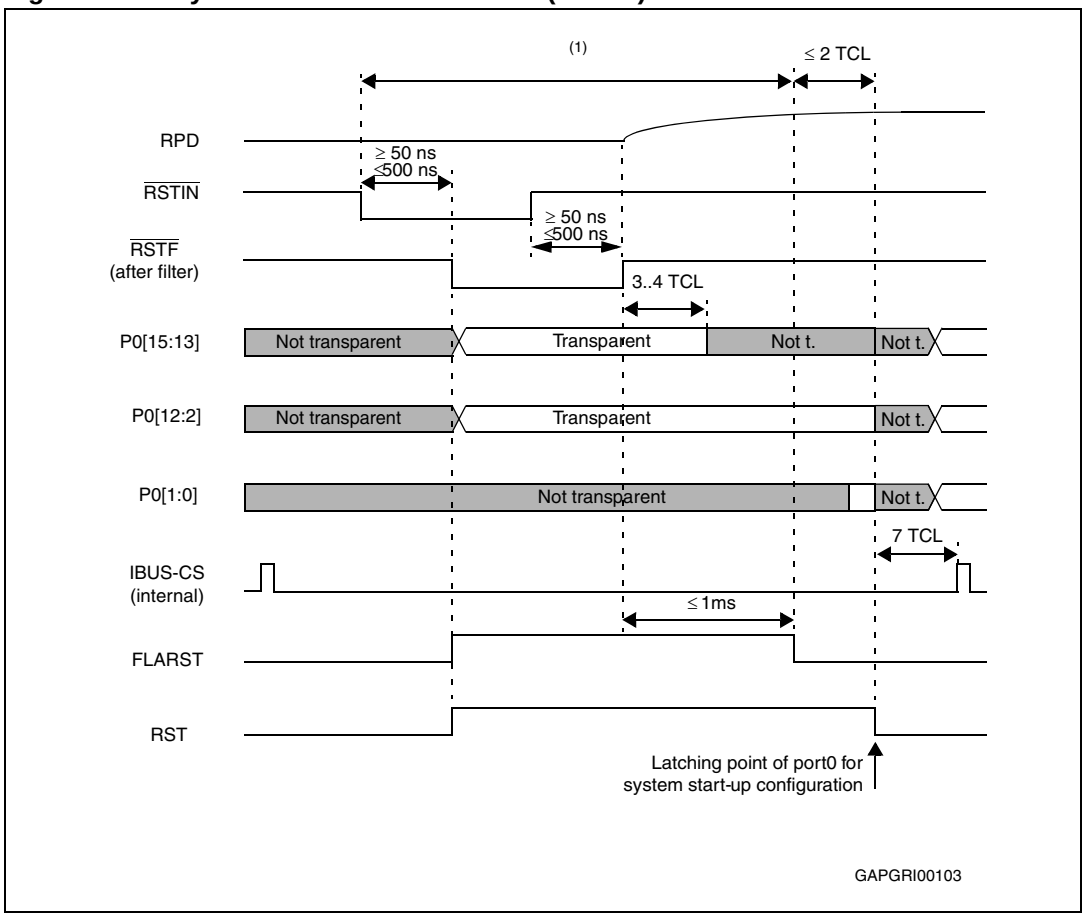


1. 3 to 8 TCL depending on clock source selection

Hardware reset

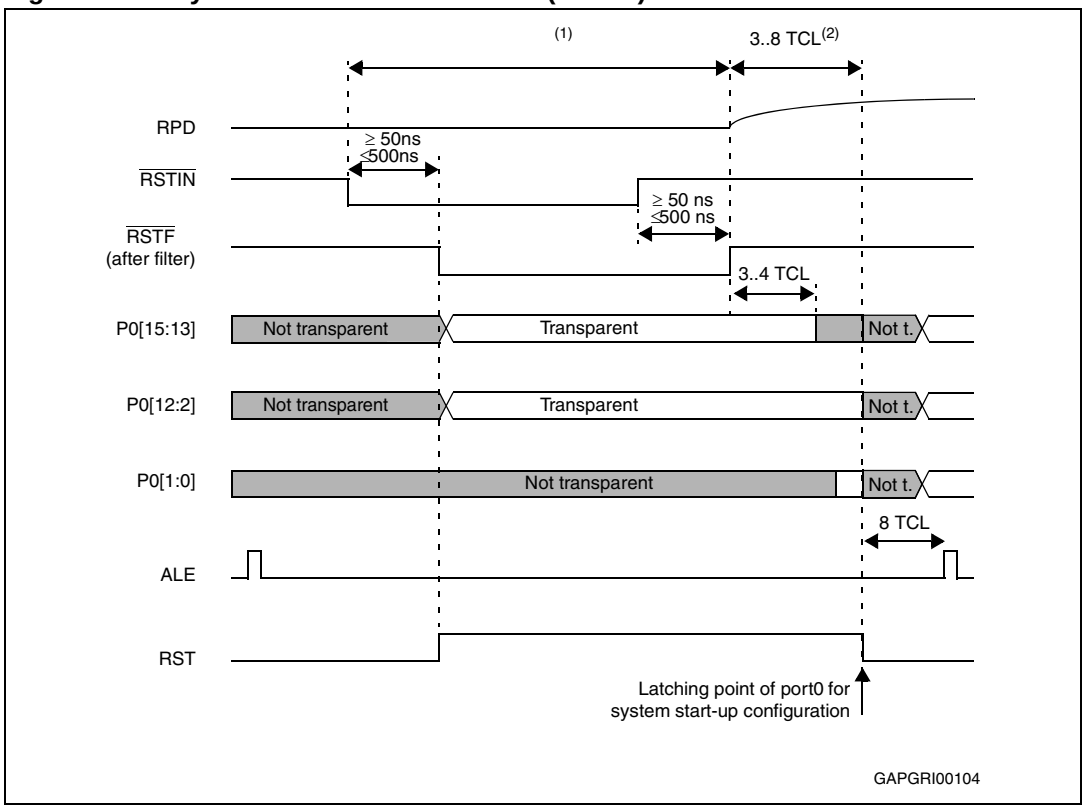
The asynchronous reset must be used to recover from catastrophic situations of the application. It may be triggered by the hardware of the application. Internal hardware logic and application circuitry are described in the reset circuitry chapter and in figures *30*, *31* and *32*. It occurs when $\overline{RSTIN}$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

Figure 19. Asynchronous hardware reset ($\overline{EA}$ = 1)



1. Longer than port0 settling time + PLL synchronization (if needed, that is P0(15:13) changed).
Longer than 500 ns to take account of input filter on RSTIN pin.

Figure 20. Asynchronous hardware reset ($\overline{EA}$ = 0)



1. Longer than port0 settling time + PLL synchronization (if needed, that is P0(15:13) changed). Longer than 500 ns to take account input filter on RSTIN pin.
2. 3 to 8 TCL depending on clock source selection.

Exit from asynchronous reset state

When the $\overline{RSTIN}$ pin is pulled high, the device restarts: As already mentioned, if internal Flash is used, the restarting occurs after the embedded Flash initialization routine is completed. The system configuration is latched from Port0: ALE, $\overline{RD}$ and $\overline{WR}$/$\overline{WRL}$ pins are driven to their inactive level. The ST10F272M starts program execution from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. The timings of asynchronous hardware reset sequence are summarized in *Figure 19* and *Figure 20*.

20.3 Synchronous reset (warm reset)

A synchronous reset is triggered when $\overline{RSTIN}$ pin is pulled low while RPD pin is at high level. In order to properly activate the internal reset logic of the device, the $\overline{RSTIN}$ pin must be held low, at least, during 4 TCL (two periods of CPU clock): Refer also to *Section 20.1* for details on minimum reset pulse duration. The I/O pins are set to high impedance and $\overline{RSTOUT}$ pin is driven low. After $\overline{RSTIN}$ level is detected, a short duration of a maximum of 12 TCL (six periods of CPU clock) elapses, during which pending internal hold states are cancelled and the current internal access cycle if any is completed. External bus cycle is aborted. The internal pull-down of $\overline{RSTIN}$ pin is activated if bit BDRSTEN of SYSCON register was previously set by software. Note that this bit is always cleared on power-on or after a reset sequence.

Short and long synchronous reset

Once the first maximum 16 TCL are elapsed (4+12 TCL), the internal reset sequence starts. It is 1024 TCL cycles long. At the end of it, and after another 8 TCL the level of $\overline{\text{RSTIN}}$ is sampled (after the filter, see $\overline{\text{RSTF}}$ in the drawings). If it is already at high level, only a short reset is flagged (refer to *Chapter 19* for details on reset flags). If it is still low, a long reset is flagged as well. The major difference between long and short resets is that during the long reset, P0(15:13) becomes transparent, so it is possible to change the clock options.

Warning: **In case of a short pulse on $\overline{\text{RSTIN}}$ pin, and when bidirectional reset is enabled, the RSTIN pin is held low by the internal circuitry. At the end of the 1024 TCL cycles, the RTSIN pin is released, but due to the presence of the input analog filter the internal input reset signal (RSTF in the drawings) is released later (from 50 to 500 ns). This delay is in parallel with the additional 8 TCL, at the end of which the internal input reset line (RSTF) is sampled, to decide if the reset event is short or long.**

- If 8 TCL > 500 ns (f_{CPU} < 8 MHz), the reset event is always recognized as short
- If 8 TCL < 500 ns (f_{CPU} > 8 MHz), the reset event could be recognized either as short or long, depending on the real filter delay (between 50 and 500 ns) and the CPU frequency ($\overline{\text{RSTF}}$ sampled high means short reset, $\overline{\text{RSTF}}$ sampled low means long reset). Note that in case a long reset is recognized, once the 8 TCL are elapsed, the P0(15:13) pins becomes transparent, so the system clock can be reconfigured. The port returns not transparent 3-4 TCL after the internal $\overline{\text{RSTF}}$ signal becomes high.

The same behavior just described, occurs also when unidirectional reset is selected and $\overline{\text{RSTIN}}$ pin is held low till the end of the internal sequence (exactly 1024 TCL + max 16 TCL) and released exactly at that time.

Note: When running with CPU frequency lower than 40 MHz, the minimum valid reset pulse to be recognized by the CPU (4 TCL) could be longer than the minimum analog filter delay (50 ns); so it might happen that a short reset pulse is not filtered by the analog input filter, but on the other hand it is not long enough to trigger a CPU reset (shorter than 4 TCL): this would generate a Flash reset but not a system reset. In this condition, the Flash answers always with FFFFh, which leads to an illegal opcode and consequently a trap event is generated.

Exit from synchronous reset state

The reset sequence is extended until the $\overline{\text{RSTIN}}$ level becomes high. Moreover, it is internally prolonged by the Flash initialization when $\overline{\text{EA}}$ = 1 (internal memory selected). Then, the code execution restarts. The system configuration is latched from port0, and ALE, $\overline{\text{RD}}$ and $\overline{\text{WR}}$/$\overline{\text{WRL}}$ pins are driven to their inactive level. The ST10F272M starts program execution from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. Timing of synchronous reset sequence are summarized in figures *21* and *22* where a short reset event is shown, with particular emphasis on the fact that it can degenerate into long reset. The two figures show the behavior when booting from internal or external memory respectively. Figures *23* and *24* report the timing of a typical synchronous long reset, again when booting from internal or external memory.

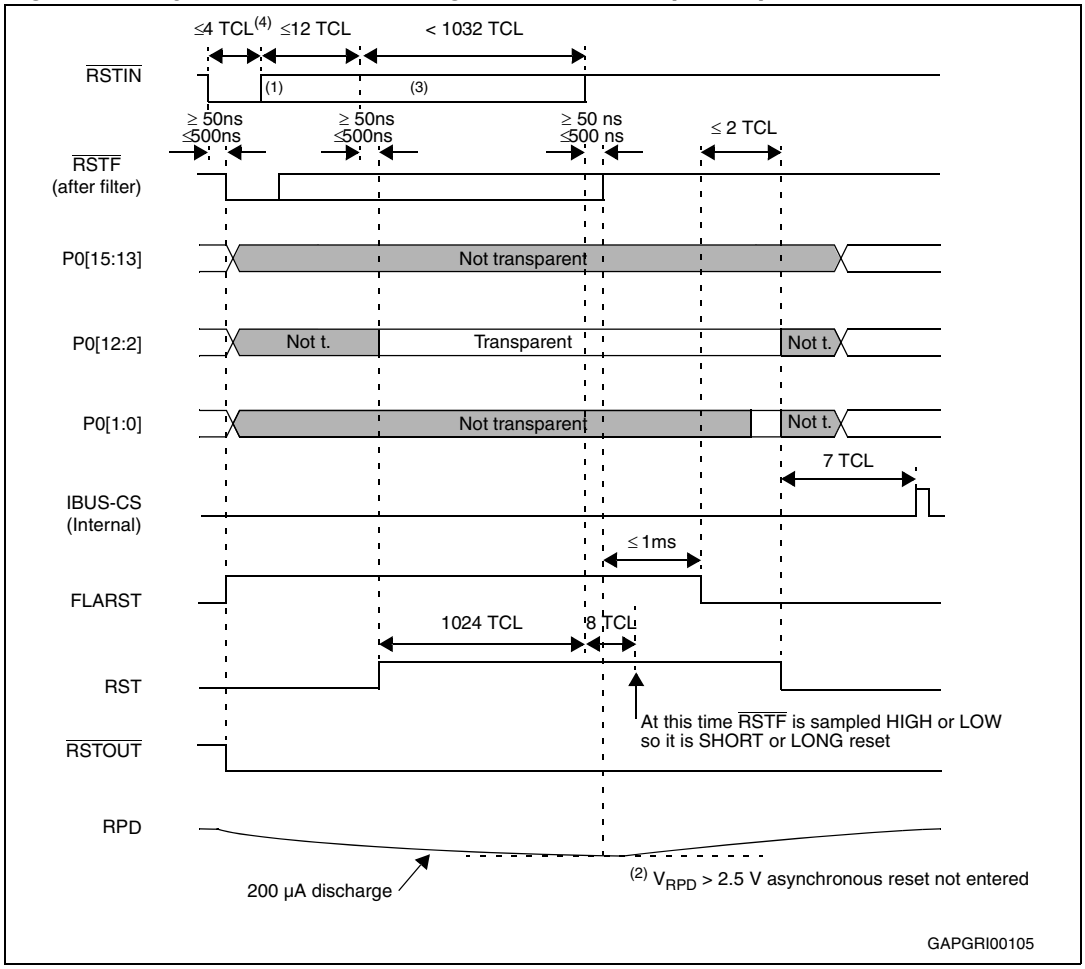
Synchronous reset and RPD pin

Whenever the RSTIN pin is pulled low (by external hardware or as a consequence of a Bidirectional reset), the RPD internal weak pull-down is activated. The external capacitance (if any) on RPD pin is slowly discharged through the internal weak pull-down. If the voltage level on RPD pin reaches the input low threshold (approximately 2.5 V), the reset event becomes immediately asynchronous. In case of hardware reset (short or long) the situation goes immediately to the one illustrated in *Figure 19*. There is no effect if RPD comes again above the input threshold: the asynchronous reset is completed coherently. To correctly complete a synchronous reset, the value of the capacitance must be big enough to maintain a sufficiently high voltage on the RPD pin for the duration of the internal reset sequence.

For a software or watchdog reset events, an active synchronous reset is completed regardless of the RPD status.

It is important to highlight that the signal that makes RPD status transparent under reset is the internal RSTF (after the noise filter).

Figure 21. Synchronous short/long hardware reset (EA = 1)

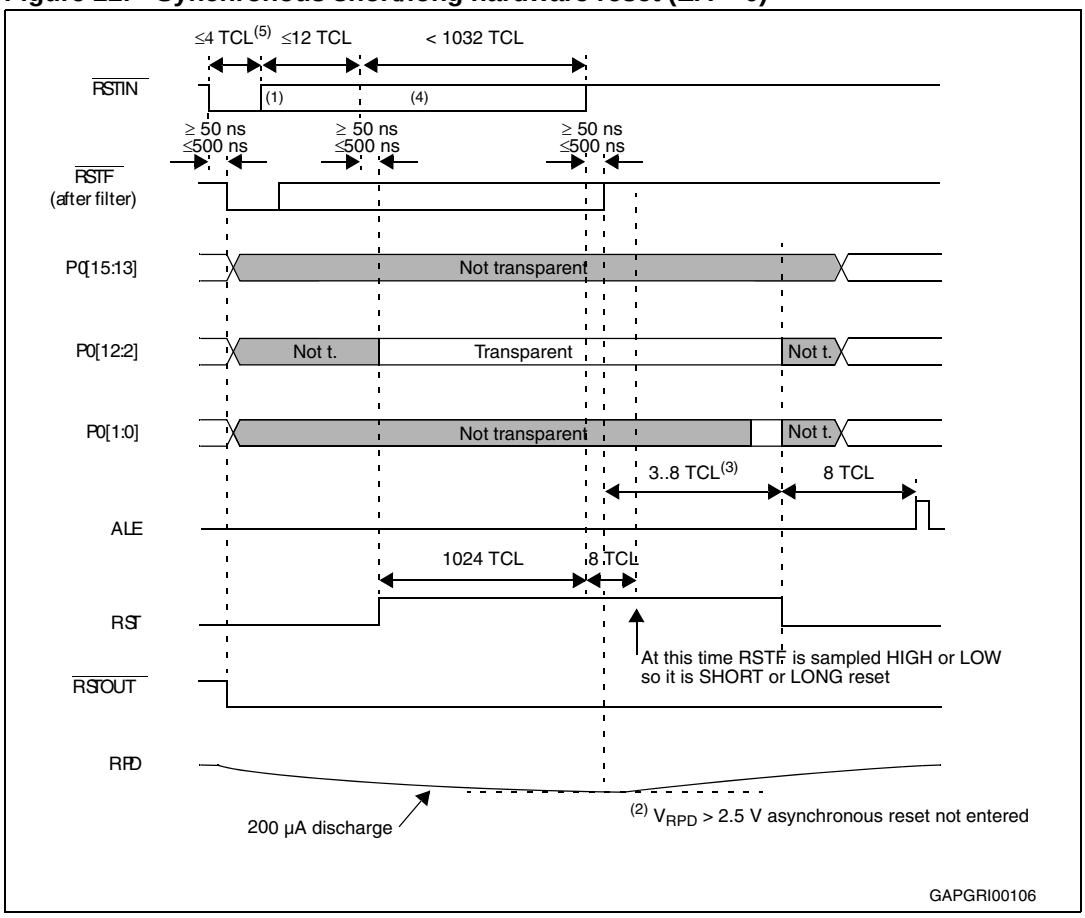


1. RSTIN assertion can be released there. Refer also to *Section 21.1* for details on minimum pulse duration.
2. If during the reset condition (RSTIN low), RPD voltage drops below the threshold voltage (about 2.5V for 5V operation), the asynchronous reset is then immediately entered.
3. RSTIN pin is pulled low if bit BDRSTEN (bit 3 of SYSCON register) was previously set by software. Bit

BDRSTEN is cleared after reset.

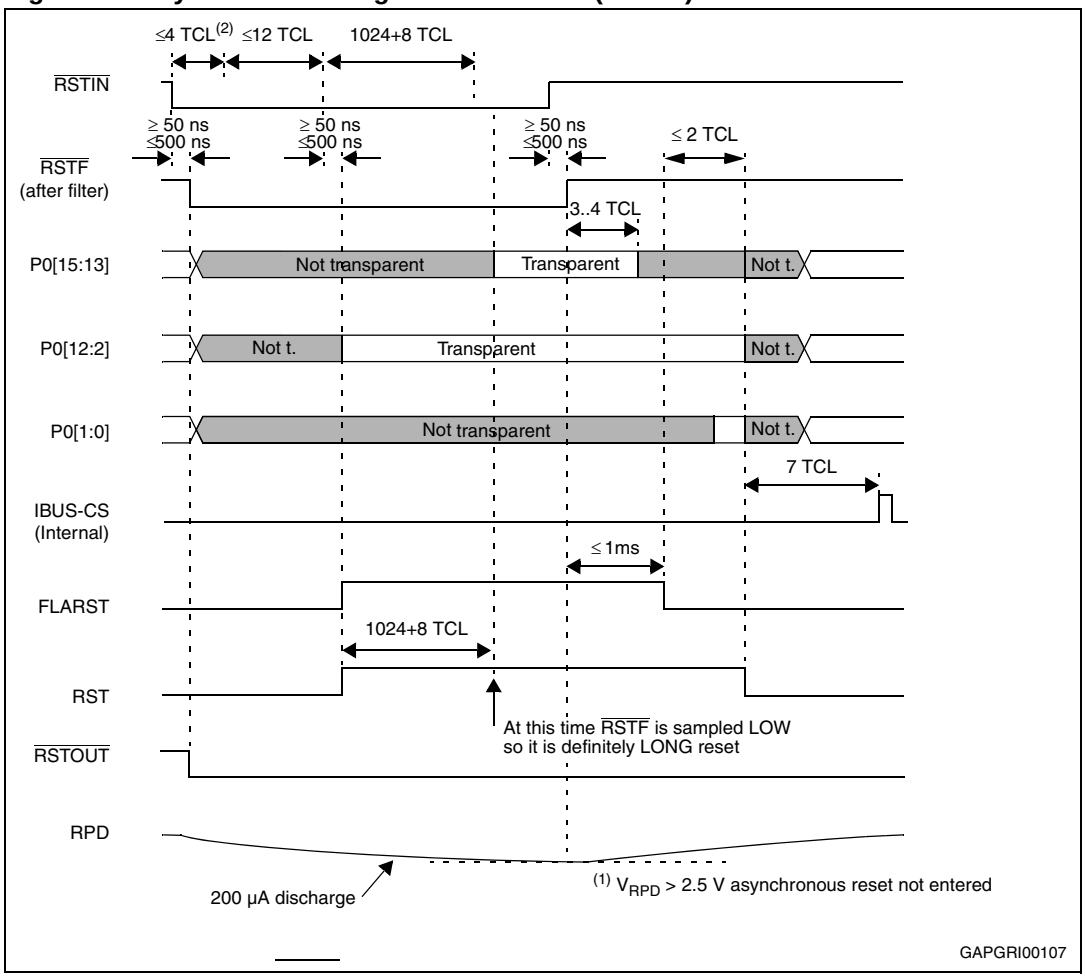
4. Minimum RSTIN low pulse duration shall also be longer than 500ns to guarantee the pulse is not masked by the internal filter (refer to *Section 21.1*)

Figure 22. Synchronous short/long hardware reset ($\overline{EA}$ = 0)



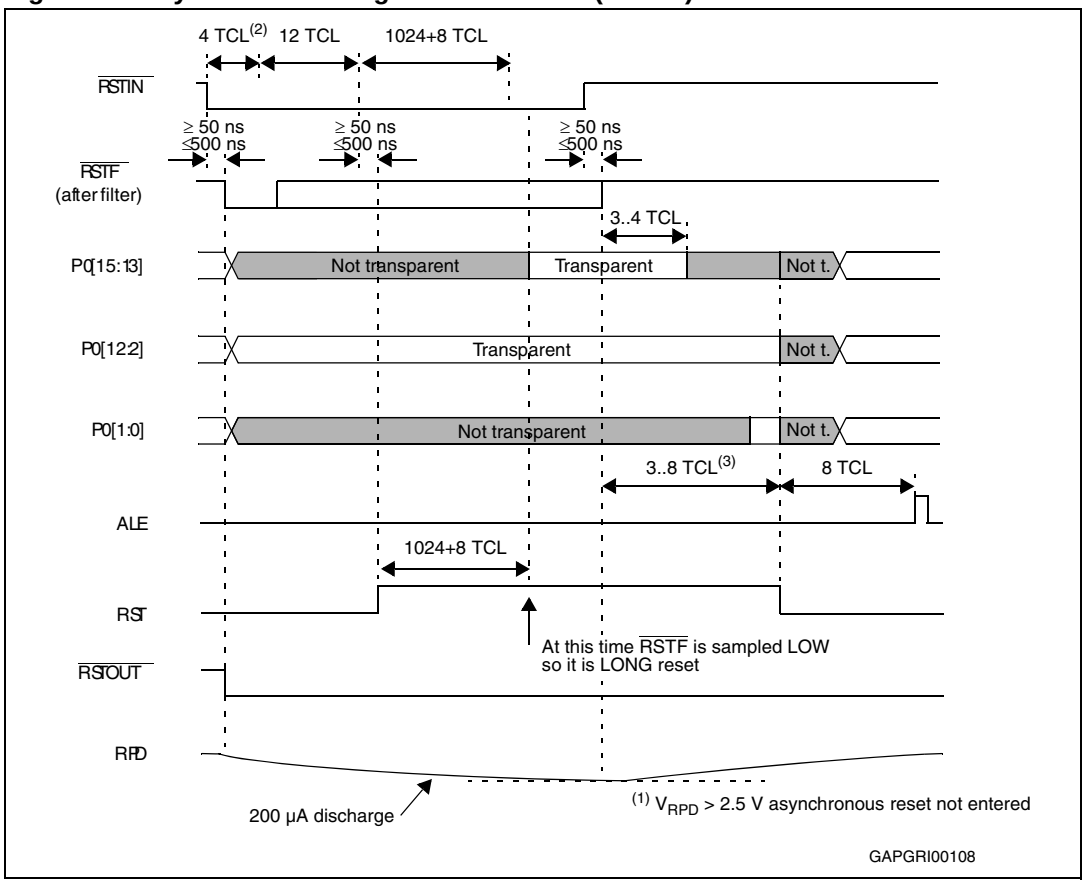
1. RSTIN assertion can be released there. Refer also to *Section 21.1* for details on minimum pulse duration.
2. If during the reset condition (RSTIN low), RPD voltage drops below the threshold voltage (about 2.5V for 5V operation), the asynchronous reset is then immediately entered.
3. 3 to 8 TCL depending 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*).

Figure 23. Synchronous long hardware reset ($\overline{EA}$ = 1)



1. If during the reset condition ($\overline{RSTIN}$ low), RPD voltage drops below the threshold voltage (about 2.5 V for 5V operation), the asynchronous reset is then immediately entered. Even if RPD returns above the threshold,the reset is defnitively taken as asynchronous.
2. Minimum $\overline{RSTIN}$ low pulse duration shall also be longer than 500 ns to guarantee the pulse is not masked by theinternal filter (refer to *Section 21.1*).

Figure 24. Synchronous long hardware reset ($\overline{EA}$ = 0)



1. If during the reset condition ($\overline{RSTIN}$ low), RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation), the asynchronous reset is then immediately entered.
2. Minimum $\overline{RSTIN}$ low pulse duration shall also be longer than 500 ns to guarantee the pulse is not masked by theinternal filter (refer to *Section 21.1*).
3. 3 to 8 TCL depending on clock source selection.

20.4 Software reset

A software reset sequence can be triggered at any time by the protected SRST (software reset) instruction. This instruction can be deliberately executed within a program, for example, to leave bootstrap loader mode, or on a hardware trap that reveals system failure.

On execution of the SRST instruction, the internal reset sequence is started. The microcontroller behavior is the same as for a synchronous short reset, except that only bits P0.12...P0.8 are latched at the end of the reset sequence, while previously latched, bits P0.7...P0.2 are cleared (that is written at ‘1’).

A software reset is always taken as synchronous: there is no influence on Software Reset behavior with RPD status. In case Bidirectional Reset is selected, a Software Reset event pulls $\overline{RSTIN}$ pin low: this occurs only if RPD is high; if RPD is low, $\overline{RSTIN}$ pin is not pulled low even though Bidirectional Reset is selected.

Refer to the figures *Figure 25* and *Figure 26* for unidirectional SW reset timing, and to figures *Figure 27*, *Figure 28* and *Figure 29* for bidirectional.

20.5 Watchdog timer reset

When the watchdog timer is not disabled during the initialization, or serviced regularly during program execution, it will overflow and trigger the reset sequence.

Unlike hardware and software resets, the watchdog reset completes a running external bus cycle if this bus cycle either does not use $\overline{\text{READY}}$, or if $\overline{\text{READY}}$ is sampled active (low) after the programmed wait states.

When $\overline{\text{READY}}$ is sampled inactive (high) after the programmed wait states the running external bus cycle is aborted. Then the internal reset sequence is started.

Bit P0.12...P0.8 are latched at the end of the reset sequence and bit P0.7...P0.2 are cleared (that is written at '1').

A Watchdog reset is always taken as synchronous: there is no influence on watchdog reset behavior with RPD status. In case bidirectional reset is selected, a watchdog reset event pulls $\overline{\text{RSTIN}}$ pin low: this occurs only if RPD is high; if RPD is low, $\overline{\text{RSTIN}}$ pin is not pulled low even though bidirectional reset is selected.

Refer to figures *Figure 25* and *Figure 26* for unidirectional SW reset timing, and to figures *Figure 27*, *Figure 28* and *Figure 29* for bidirectional.

Figure 25. SW/WDT unidirectional reset ($\overline{\text{EA}}$ = 1)

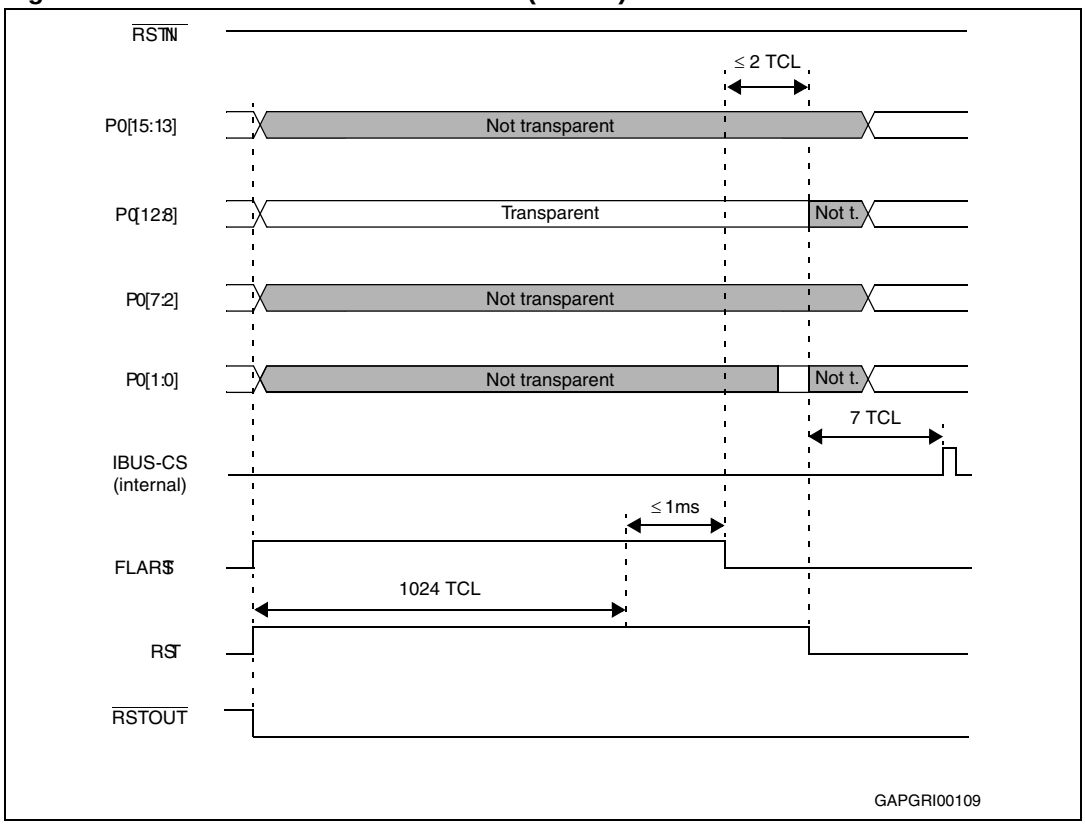
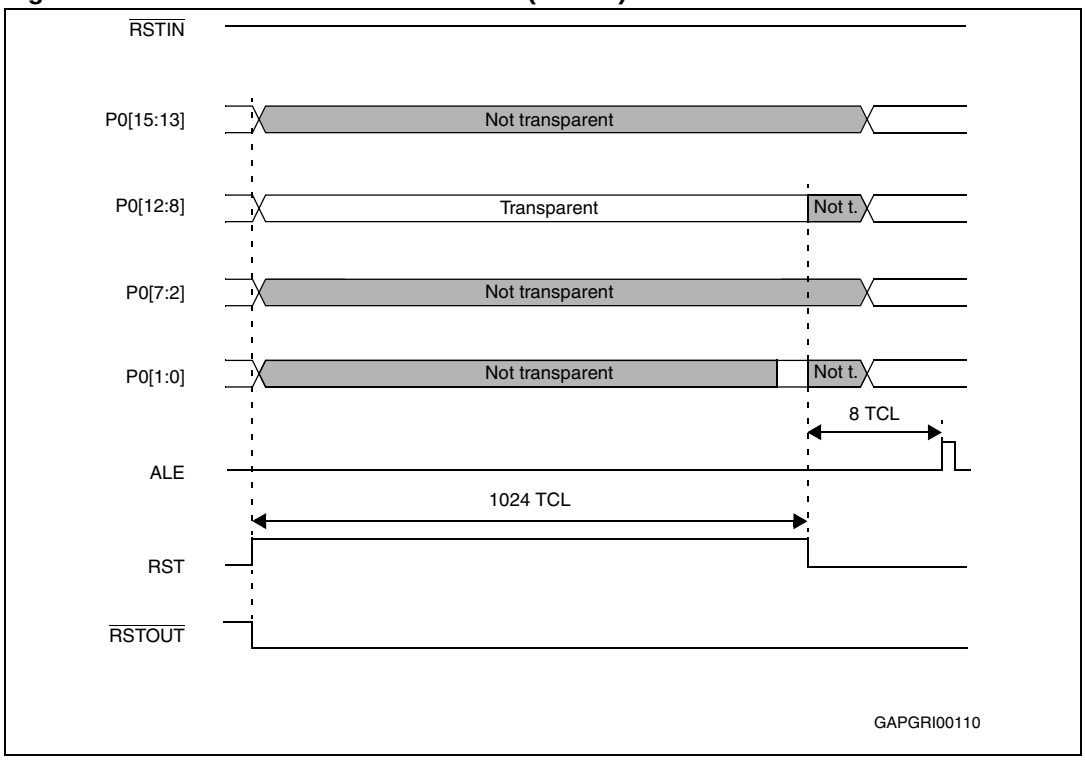


Figure 26. SW/WDT unidirectional reset ($\overline{EA}$ = 0)



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

or watchdog reset become a short hardware reset. On the contrary, if $\overline{RSTF}$ 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 $\overline{RSTIN}$ 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.

WDTCON flags

Similar to what is highlighted in the previous section, when discussing short reset and the degeneration into long reset, comparable situations may occur when bidirectional reset is enabled. The presence of the internal filter on $\overline{RSTIN}$ pin introduces a delay: When $\overline{RSTIN}$ is released, the internal signal after the filter (see $\overline{RSTF}$ 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 WDTCON flags are set accordingly.

Moreover, when either software or watchdog bidirectional reset events occur, when the $\overline{RSTIN}$ pin is released (at the end of the internal reset sequence), the $\overline{RSTF}$ 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 $\overline{RSTF}$ signal is sampled, and if recognized still low a hardware reset sequence starts, and WDTCON will flag this last event, masking the previous one (software or watchdog reset). Typically, a short hardware reset is recognized, unless the $\overline{RSTIN}$ pin (and consequently internal signal $\overline{RSTF}$) 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 WDTCON register. This phenomenon does not occur when internal Flash is selected during reset ($\overline{EA}$ = 1), since the initialization of the Flash itself extend the internal reset duration well beyond the filter delay.

Figures *Figure 27*, *Figure 28* and *Figure 29* summarize the timing for software and watchdog timer bidirectional reset events: In particular *Figure 29* shows the degeneration into hardware reset.

Figure 27. SW/WDT bidirectional reset ($\overline{EA}$ = 1)

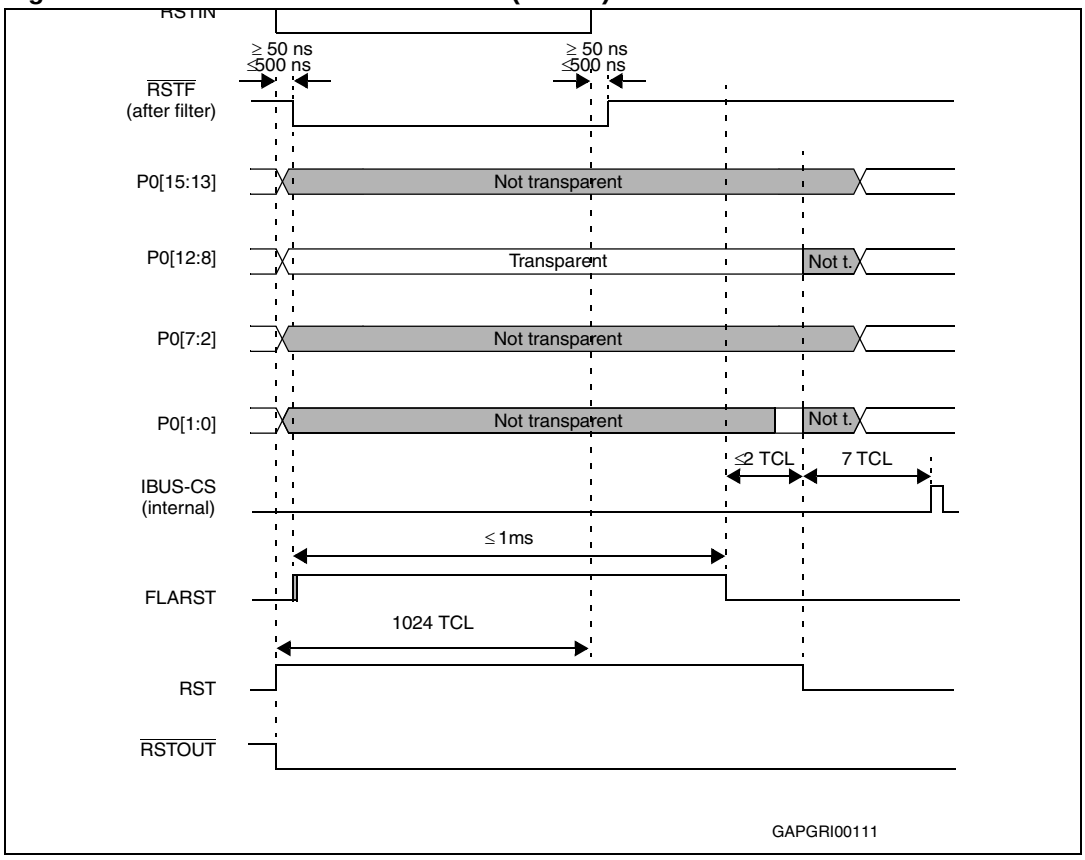


Figure 28. SW/WDT bidirectional reset ($\overline{EA}$ = 0)

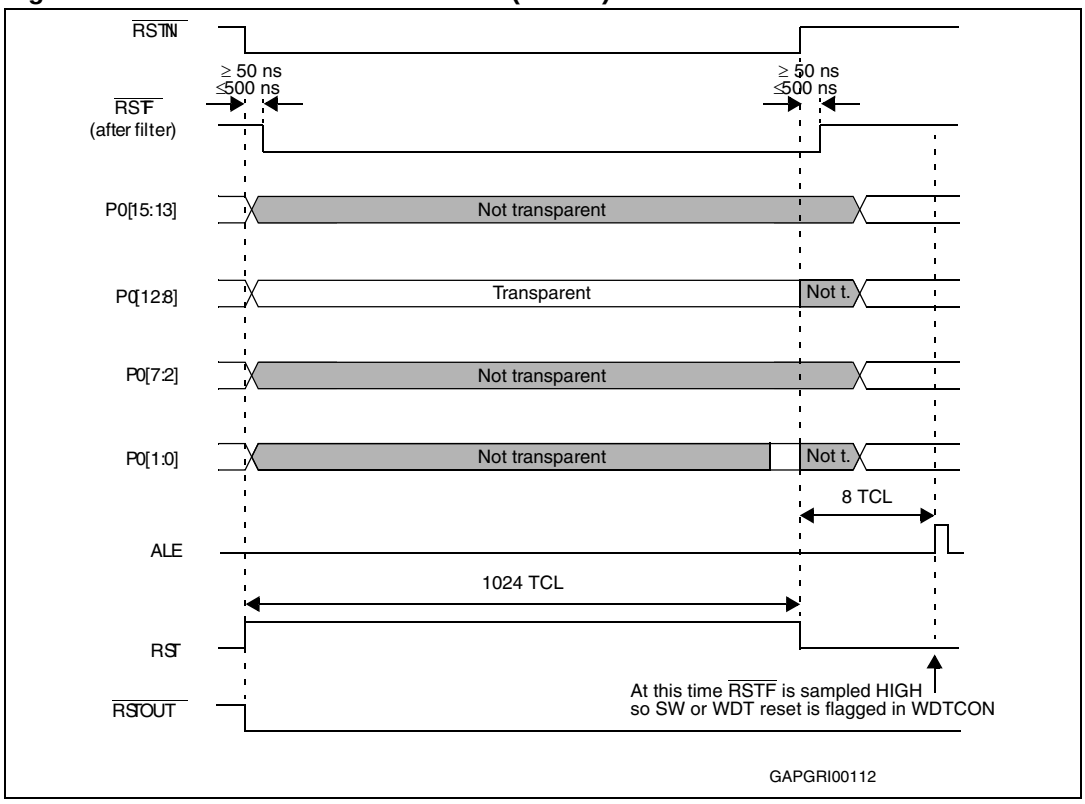
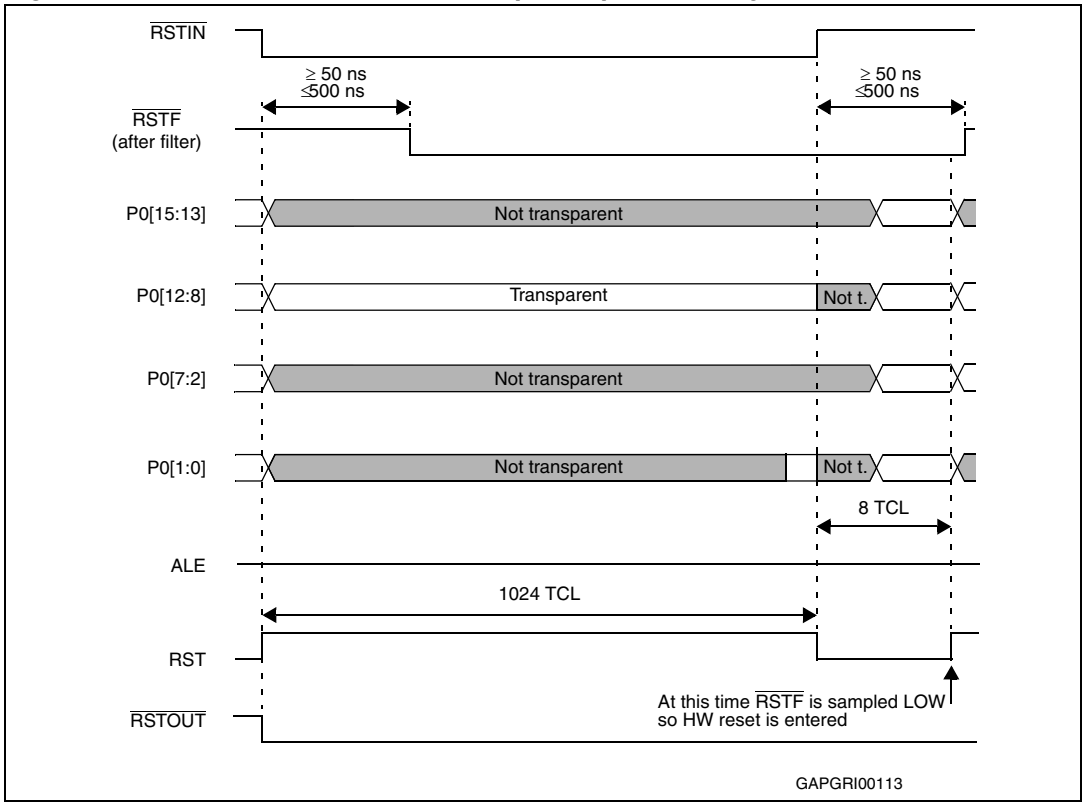


Figure 29. SW/WDT bidirectional reset ($\overline{EA}$ = 0) followed by a HW reset



20.7 Reset circuitry

Internal reset circuitry is described in *Figure 32*. The $\overline{\text{RSTIN}}$ pin provides an internal pull-up resistor of 50 kΩ to 250 kΩ (The minimum reset time must be calculated using the lowest value).

It also provides a programmable (BDRSTEN bit of SYSCON register) pull-down to output internal reset state signal (synchronous reset, watchdog timer reset or software reset).

This bidirectional reset function is useful in applications where external devices require a reset signal but cannot be connected to $\overline{\text{RSTOUT}}$ pin.

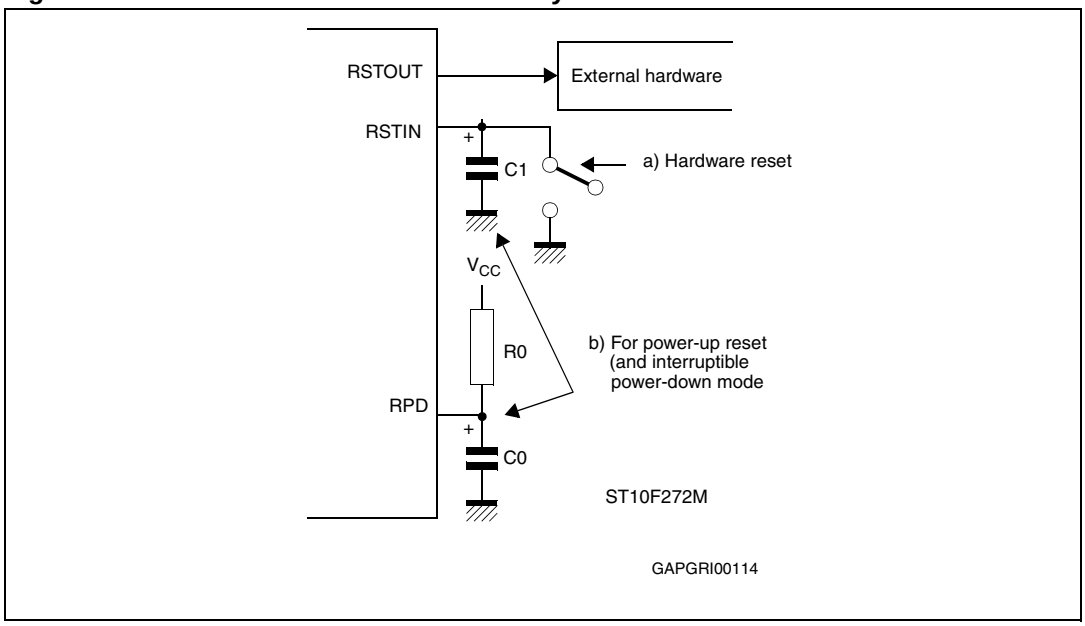
This is the case of an external memory running codes before EINIT (end of initialization) instruction is executed. $\overline{\text{RSTOUT}}$ pin is pulled high only when EINIT is executed.

The RPD pin provides an internal weak pull-down resistor which discharges external capacitor at a typical rate of 200 µA. If bit PWDCFG of SYSCON register is set, an internal pull-up resistor is activated at the end of the reset sequence. This pull-up will charge any capacitor connected on RPD pin.

The simplest way to reset the ST10F272M is to insert a capacitor C1 between $\overline{\text{RSTIN}}$ pin and V_{SS}, and a capacitor between RPD pin and V_{SS} (C0) with a pull-up resistor R0 between RPD pin and V_{DD}. The input $\overline{\text{RSTIN}}$ provides an internal pull-up device equalling a resistor of 50 kΩ to 250 kΩ (the minimum reset time must be determined by the lowest value). Select C1 that produce a sufficient discharge time to permit the internal or external oscillator and / or internal PLL and the on-chip voltage regulator to stabilize.

To ensure correct power-up reset with controlled supply current consumption, specially if clock signal requires a long period of time to stabilize, an asynchronous hardware reset is required during power-up. For this reason, it is recommended to connect the external R0-C0 circuit shown in *Figure 30* to the RPD pin. On power-up, the logical low level on RPD pin forces an asynchronous hardware reset when $\overline{\text{RSTIN}}$ is asserted low. The external pull-up R0 will then charge the capacitor C0. Note that an internal pull-down device on RPD pin is turned on when $\overline{\text{RSTIN}}$ pin is low, and causes the external capacitor (C0) to begin discharging at a typical rate of 100-200 µA. With this mechanism, after power-up reset, short low pulses applied on $\overline{\text{RSTIN}}$ produce synchronous hardware reset. If $\overline{\text{RSTIN}}$ is asserted longer than the time needed for C0 to be discharged by the internal pull-down device, then the device is forced in an asynchronous reset. This mechanism insures recovery from catastrophic failure.

Figure 30. Minimum external reset circuitry



The minimum reset circuit of *Figure 30* is not adequate when the $\overline{\text{RSTIN}}$ pin is driven from the ST10F272M itself during software or watchdog triggered resets, because of the capacitor C1 that will keep the voltage on $\overline{\text{RSTIN}}$ pin above V_{IL} after the end of the internal reset sequence, and thus will trigger an asynchronous reset sequence.

Figure 31 shows an example of a reset circuit. In this example, R1-C1 external circuit is only used to generate power-up or manual reset, and R0-C0 circuit on RPD is used for power-up reset and to exit from power-down mode. Diode D1 creates a wired-OR gate connection to the reset pin and may be replaced by open-collector Schmitt trigger buffer. Diode D2 provides a faster cycle time for repetitive power-on resets.

R2 is an optional pull-up for faster recovery and correct biasing of TTL open collector drivers.

Figure 31. System reset circuit

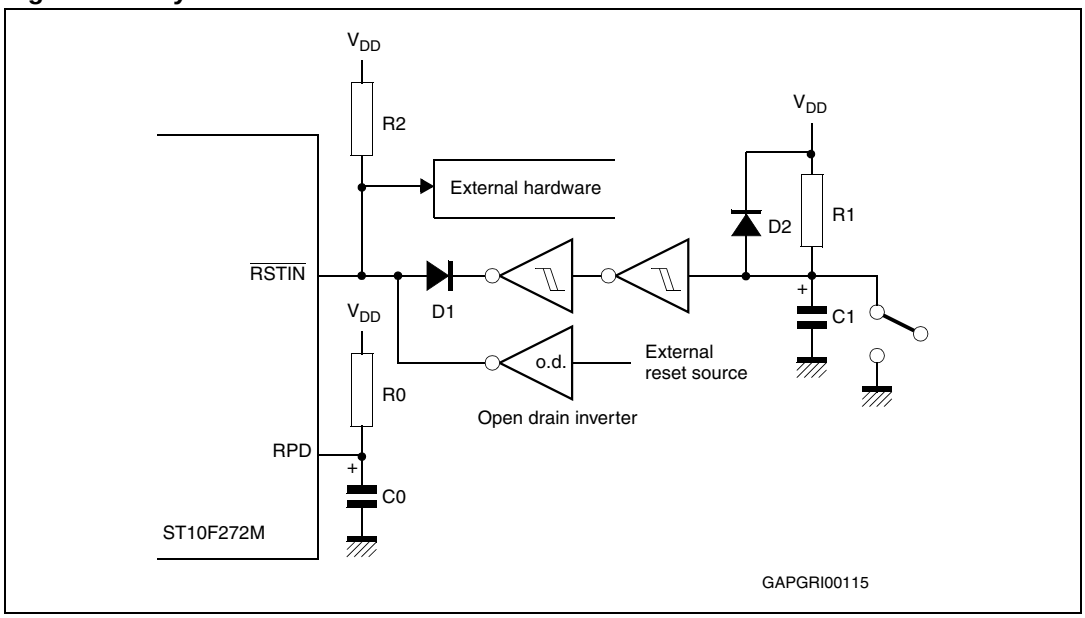
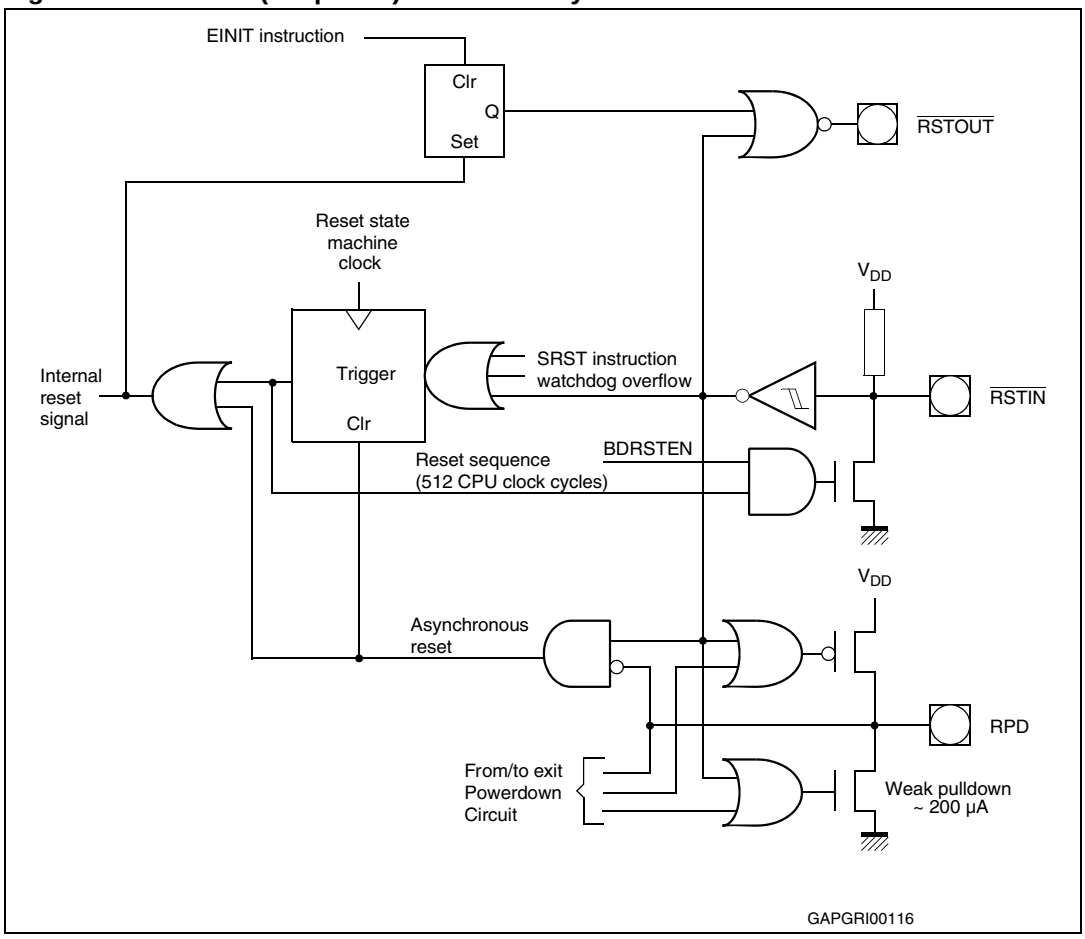


Figure 32. Internal (simplified) reset circuitry



20.8 Reset application examples

The next two timing diagrams (*Figure 33* and *Figure 34*) provide additional examples of bidirectional internal reset events (software and watchdog) including in particular the external capacitances charge and discharge transients (refer also to *Figure 31* for the external circuit scheme).

Figure 33. Example of software or watchdog bidirectional reset ($\overline{EA}$ = 1)

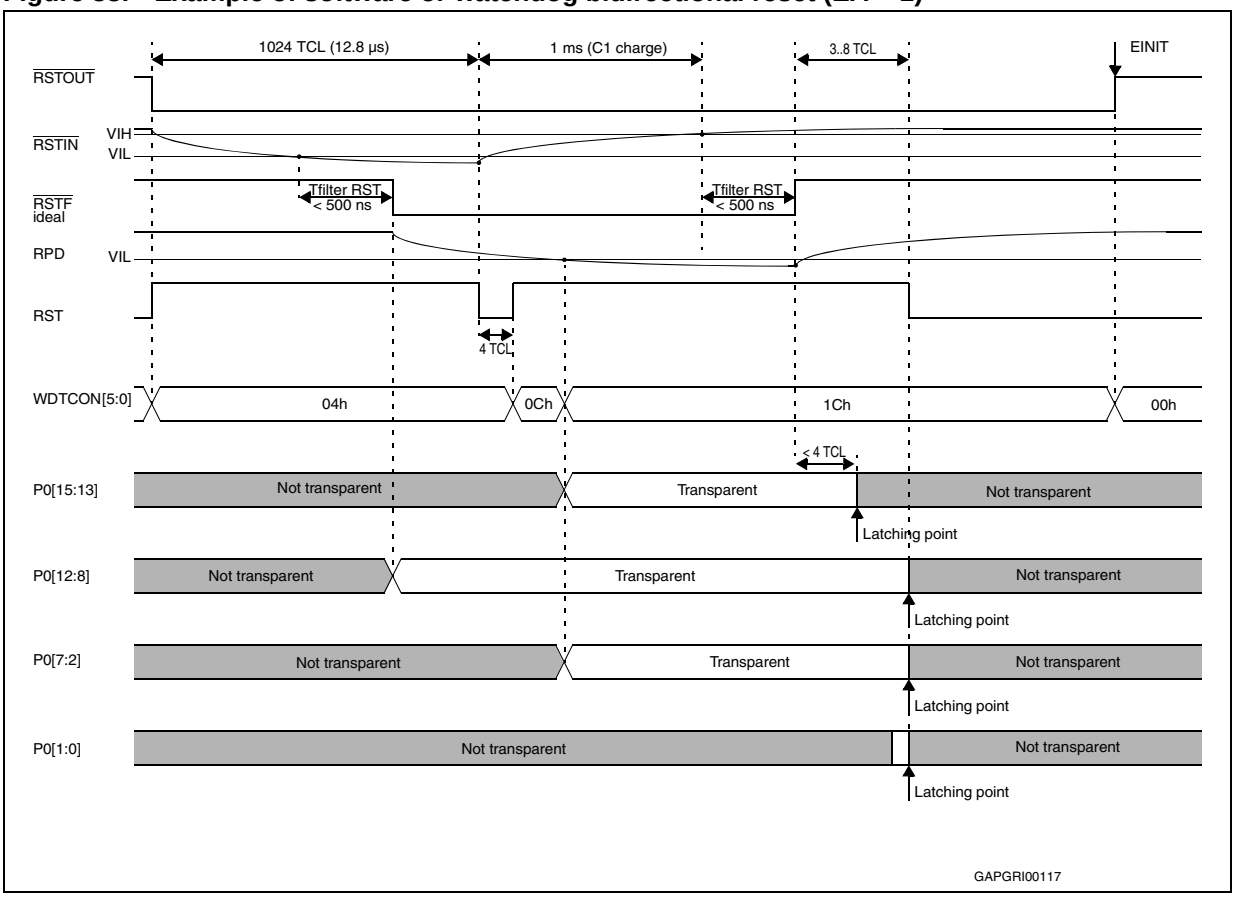
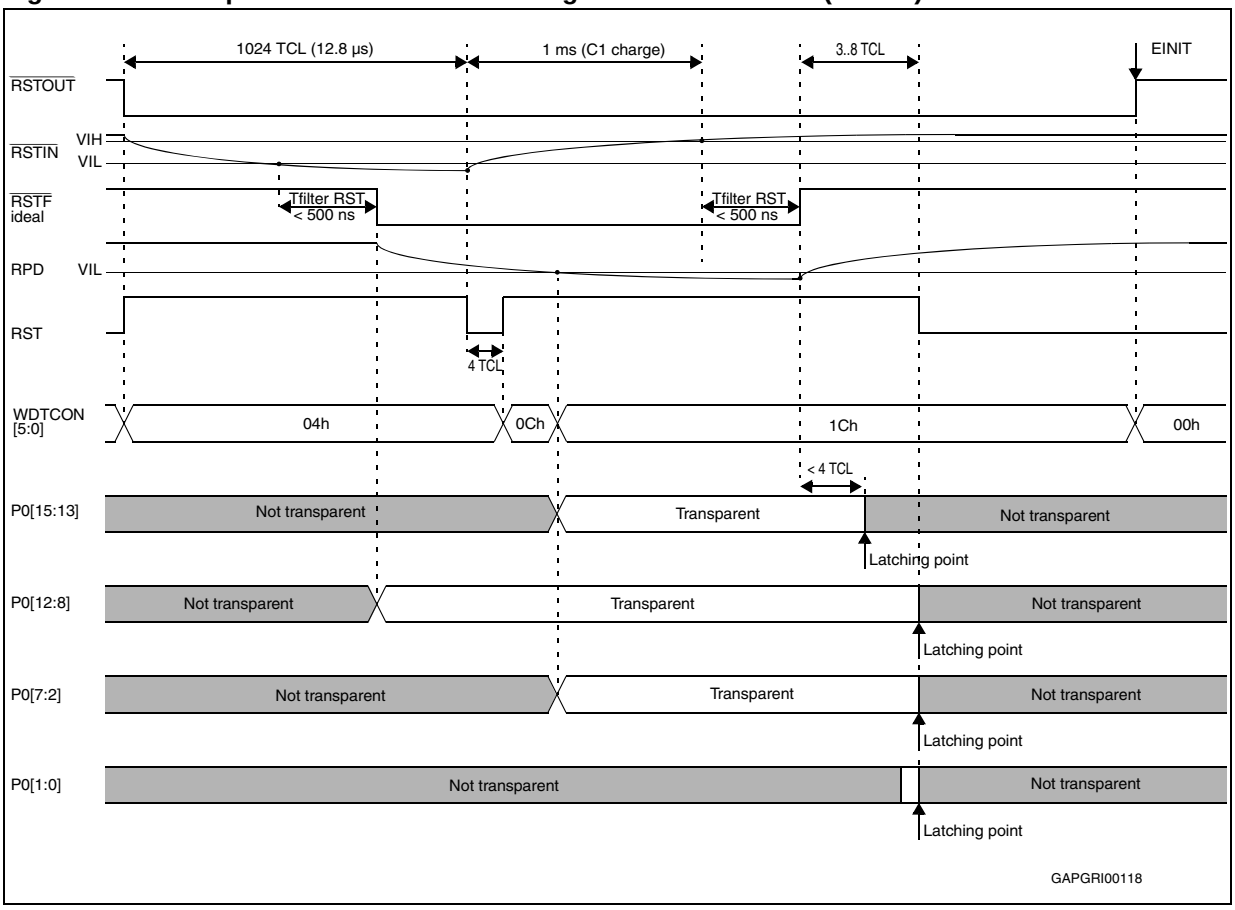


Figure 34. Example of software or watchdog bidirectional reset ($\overline{EA}$ = 0)



20.9 Reset summary

The following table summarizes the different reset events.

Table 42. Reset event

Event	RPD	$\overline{EA}$	Bidir	Synch. asynch.	RSTIN		WDTCON flags				
					Min	Max	PONR	LHWR	SHWR	SWR	WDTR
Power-on reset	0	0	N	Asynch.	1 ms (VREG) 1.2 ms (reson. + PLL) 10.2 ms (crystal + PLL)	-	1	1	1	1	0
	0	1	N	Asynch.	1ms (VREG)	-	1	1	1	1	0
	1	x	x	Forbidden							
	x	x	Y	-							
Hardware reset (asynchronous)	0	0	N	Asynch.	500 ns	-	0	1	1	1	0
	0	1	N	Asynch.	500 ns	-	0	1	1	1	0
	0	0	Y	Asynch.	500 ns	-	0	1	1	1	0
	0	1	Y	Asynch.	500 ns	-	0	1	1	1	0
Short hardware reset (synchronous) (1)	1	0	N	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max(4 TCL, 500 ns)	0	0	1	1	0
	1	1	N	Synch.	max (4 TCL, 500 ns)	1032 + 12 TCL + max(4 TCL, 500 ns)	0	0	1	1	0
	1	0	Y	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max(4 TCL, 500 ns)	0	0	1	1	0
					Activated by internal logic for 1024 TCL						
	1	1	Y	Synch.	Max (4 TCL, 500ns)	1032 + 12 TCL + max(4 TCL, 500 ns)	0	0	1	1	0
					Activated by internal logic for 1024 TCL						
Long hardware reset (synchronous)	1	0	N	Synch.	1032 + 12 TCL + Max(4 TCL, 500 ns)	-	0	1	1	1	0
	1	1	N	Synch.	1032 + 12 TCL + Max(4 TCL, 500ns)	-	0	1	1	1	0
	1	0	Y	Synch.	1032 + 12 TCL + Max(4 TCL, 500 ns)	-	0	1	1	1	0
					Activated by internal logic only for 1024 TCL						
	1	1	Y	Synch.	1032 + 12 TCL + Max(4 TCL, 500 ns)	-	0	1	1	1	0
					Activated by internal logic only for 1024 TCL						

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).

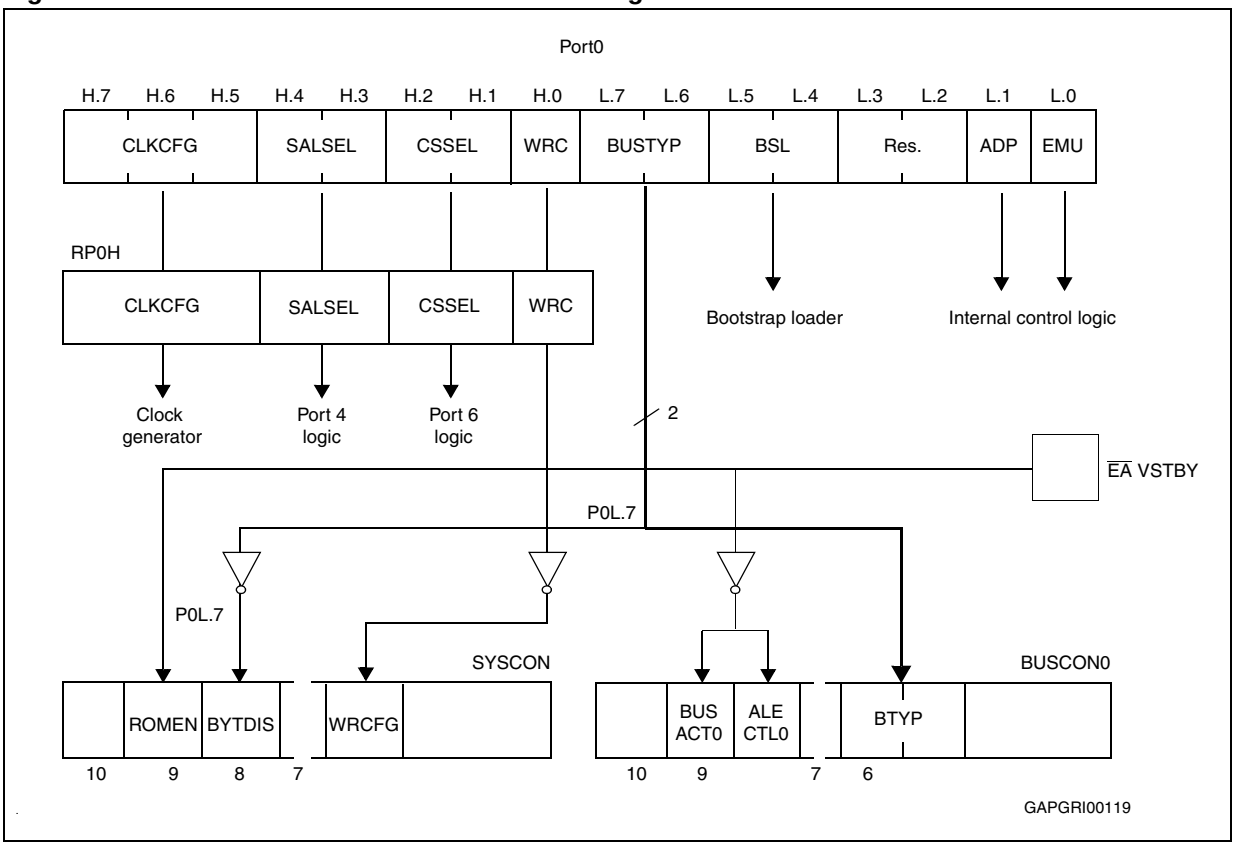
The start-up configurations and some system features are selected on reset sequences as described in *Table 43* and *Figure 35*.

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

Figure 35. Port0 bits latched into the different registers after reset



21 Power reduction modes

Three different power reduction modes with different levels of power reduction have been implemented in the ST10F272M. In idle mode only CPU is stopped, while peripheral still operates. In power-down mode both CPU and peripherals are stopped. In stand-by mode the main power supply (V_{DD}) can be turned off while a portion of the internal RAM remains powered via V_{STBY} dedicated power pin.

Idle and power-down modes are software activated by a protected instruction and are terminated in different ways as described in the following sections.

Stand-by mode is entered simply removing V_{DD}, holding the MCU under reset state.

*Note: All external bus actions are completed before idle or power-down mode is entered. However, idle or power-down mode is **not** entered if READY is enabled, but has not been activated (driven low for negative polarity, or driven high for positive polarity) during the last bus access.*

21.1 Idle mode

Idle mode is entered by running IDLE protected instruction. The CPU operation is stopped and the peripherals still run.

Idle mode is terminated by any interrupt request. Whatever the interrupt is serviced or not, the instruction following the IDLE instruction will be executed after return from interrupt (RETI) instruction, then the CPU resumes the normal program.

21.2 Power-down mode

Power-down mode starts by running PWRDN protected instruction. Internal clock is stopped, all MCU parts are on hold including the watchdog timer. The only exception could be the real-time clock if opportunely programmed and one of the two oscillator circuits as a consequence (either the main or the 32 kHz on-chip oscillator).

When real-time clock module is used, when the device is in power-down mode a reference clock is needed. In this case, two possible configurations may be selected by the user application according to the desired level of power reduction:

- A 32 kHz crystal is connected to the on-chip low-power oscillator (pins XTAL3 / XTAL4) and running. In this case the main oscillator is stopped when power-down mode is entered, while the real-time clock continue counting using 32 kHz clock signal as reference. The presence of a running low-power oscillator is detected after the power-on: this clock is immediately assumed (if present, or as soon as it is detected) as reference for the real-time clock counter and it will be maintained forever (unless specifically disabled via software).
- Only the main oscillator is running (XTAL1 / XTAL2 pins). In this case the main oscillator is not stopped when power-down is entered, and the real-time clock continue counting using the main oscillator clock signal as reference.

There are two different operating power-down modes: protected mode and interruptible mode.

Before entering power-down mode (by executing the instruction PWRDN), bit VREGOFF in XMISC register must be set.

Note: Leaving the main voltage regulator active during power-down may lead to unexpected behavior (example: CPU wake-up) and power consumption higher than what is specified.

21.2.1 Protected power-down mode

This mode is selected when PWDCFG (bit 5) of SYSCON register is cleared. The protected power-down mode is only activated if the $\overline{\text{NMI}}$ pin is pulled low when executing PWRDN instruction (this means that the PWRD instruction belongs to the $\overline{\text{NMI}}$ software routine). This mode is only deactivated with an external hardware reset on $\overline{\text{RSTIN}}$ pin.

21.2.2 Interruptible power-down mode

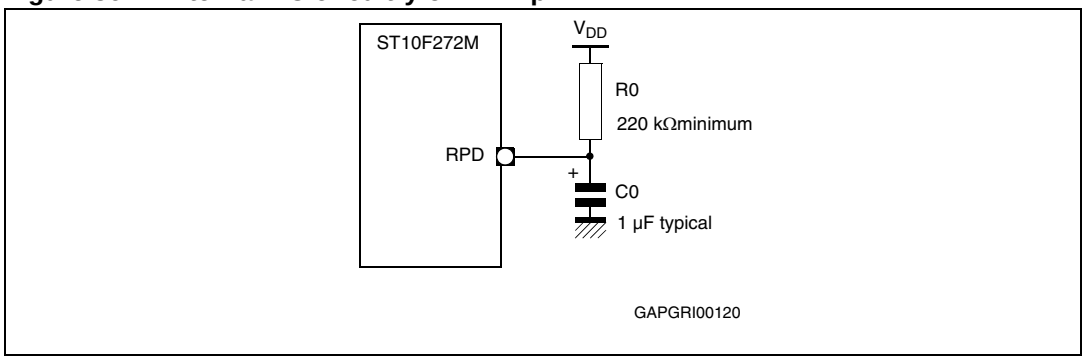
This mode is selected when PWDCFG (bit 5) of SYSCON register is set.

The interruptible power-down mode is only activated if all the enabled fast external interrupt pins are in their inactive level.

This mode is deactivated with an external reset applied to $\overline{\text{RSTIN}}$ pin or with an interrupt request applied to one of the fast external interrupt pins, or with an interrupt generated by the real-time clock, or with an interrupt generated by the activity on CAN's and I^2C module interfaces. To allow the internal PLL and clock to stabilize, the $\overline{\text{RSTIN}}$ pin must be held low according the recommendations described in *Chapter 20: System reset on page 77*.

An external RC circuit must be connected to RPD pin, as shown in the *Figure 36*.

Figure 36. External RC circuitry on RPD pin



To exit power-down mode with an external interrupt, an EXxIN (x = 7...0) pin has to be asserted for at least 40 ns.

21.3 Stand-by mode

In stand-by mode, it is possible to turn off the main V_{DD} provided that V_{STBY} is available through the dedicated pin of the ST10F272M.

To enter stand-by mode it is mandatory to held the device under reset: once the device is under reset, the RAM is disabled (see XRAM2EN bit of XPERCON register), and its digital interface is frozen in order to avoid any kind of data corruption.

A dedicated embedded low-power voltage regulator is implemented to generate the internal low voltage supply (about 1.65 V in stand-by mode) to bias all those circuits that shall

remain active: the portion of XRAM (16 Kbytes for ST10F272M), the RTC counters and 32 kHz on-chip oscillator amplifier.

In normal running mode (that is when main V_{DD} is on) the V_{STBY} pin can be tied to V_{SS} during reset to exercise the $\overline{EA}$ functionality associated with the same pin: the voltage supply for the circuitries which are usually biased with V_{STBY} (see in particular the 32 kHz oscillator used in conjunction with real-time clock module), is granted by the active main V_{DD}.

It must be noted that stand-by mode can generate problems associated with the usage of different power supplies in CMOS systems; particular attention must be paid when the ST10F272M I/O lines are interfaced with other external CMOS integrated circuits: if V_{DD} of ST10F272M becomes (for example, in stand-by mode) lower than the output level forced by the I/O lines of these external integrated circuits, the ST10F272M could be directly powered through the inherent diode existing on ST10F272M output driver circuitry. The same is valid for ST10F272M interfaced to active/inactive communication buses during stand-by mode: current injection can be generated through the inherent diode.

Furthermore, the sequence of turning on/off of the different voltage could be critical for the system (not only for the ST10F272M device). The device stand-by mode current (I_{STBY}) may vary while V_{DD} to V_{STBY} (and vice versa) transition occurs: some current flows between V_{DD} and V_{STBY} pins. System noise on both V_{DD} and V_{STBY} can contribute to increase this phenomenon.

21.3.1 Entering stand-by mode

As already stated, to enter stand-by mode the XRAM2EN bit in the XPERCON register must be cleared: This allows the RAM interface to be frozen immediately, avoiding any data corruption. As a consequence of a reset event, the RAM power supply is switched to the internal low-voltage supply V_{18SB} (derived from V_{STBY} through the low-power voltage regulator). The RAM interface remains frozen until the bit XRAM2EN is set again by software initialization routine (at next exit from main V_{DD} power-on reset sequence).

Since V_{18} is falling down (as a consequence of V_{DD} turning off), it can happen that the XRAM2EN bit is no longer able to guarantee its content (logic “0”), being the XPERCON Register powered by internal V_{18}. This does not generate any problem, because the stand-by mode switching dedicated circuit continues to confirm the RAM interface freezing, irrespective the XRAM2EN bit content; XRAM2EN bit status is considered again when internal V_{18} comes back over internal stand-by reference V_{18SB}.

If internal V_{18} becomes lower than internal stand-by reference (V_{18SB}) of about 0.3 to 0.45V with bit XRAM2EN set, the RAM supply switching circuit is not active: in case of a temporary drop on internal V_{18} voltage versus internal V_{18SB} during normal code execution, no spurious stand-by mode switching can occur (the RAM is not frozen and can still be accessed).

The ST10F272M core module, generating the RAM control signals, is powered by internal V_{18} supply; during turning off transient these control signals follow the V_{18}, while RAM is switched to V_{18SB} internal reference. It could happen that a high level of RAM write strobe from ST10F272M core (active low signal) is low enough to be recognized as a logic “0” by the RAM interface (due to V_{18} lower than V_{18SB}): The bus status could contain a valid address for the RAM and an unwanted data corruption could occur. For this reason, an extra interface, powered by the switched supply, is used to prevent the RAM from this kind of potential corruption mechanism.

Warning: **During power-off phase, it is important that the external hardware maintains a stable ground level on RSTIN pin, without any glitch, in order to avoid spurious exiting from reset status with unstable power supply.**

21.3.2 Exiting stand-by mode

After the system has entered the stand-by mode, the procedure to exit this mode consists of a standard power-on sequence, with the only difference that the RAM is already powered through V_{18SB} internal reference (derived from V_{STBY} pin external voltage).

It is recommended to held the device under reset ($\overline{RSTIN}$ pin forced low) until external V_{DD} voltage pin is stable. Even though, at the very beginning of the power-on phase, the device is maintained under reset by the internal low voltage detector circuit (implemented inside the main voltage regulator) till the internal V_{18} becomes higher than about 1.0 V, there is no guaranty that the device stays under reset status if $\overline{RSTIN}$ is at high level during power ramp up. So, it is important the external hardware is able to guarantee a stable ground level on $\overline{RSTIN}$ along the power-on phase, without any temporary glitch.

The external hardware is responsible for driving the $\overline{RSTIN}$ pin low until the V_{DD} is stable, even though the internal LVD is active.

Once the internal reset signal goes low, the RAM (still frozen) power supply is switched to the main V_{18}.

At this time, everything becomes stable, and the execution of the initialization routines can start: XRAM2EN bit can be set, enabling the RAM.

21.3.3 Real-time clock and stand-by mode

When stand-by mode is entered (turning off the main supply V_{DD}), the real-time clock counting can be maintained running in case the on-chip 32 kHz oscillator is used to provide the reference to the counter. This is not possible if the main oscillator is used as reference for the counter: Being the main oscillator powered by V_{DD}, once this is switched off, the oscillator is stopped.

21.3.4 Power reduction modes summary

The different power reduction modes are summarized in the following *Table 44*.

Table 44. Power reduction modes summary

Mode	V_{DD}	V_{STBY}	CPU	Peripherals	RTC	Main OSC	32 kHz OSC	STBY XRAM	XRAM
Idle	On	On	Off	On	Off	Run	Off	Biased	Biased
	On	On	Off	On	On	Run	On	Biased	Biased
Power-down	On	On	Off	Off	Off	Off	Off	Biased	Biased
	On	On	Off	Off	On	On	Off	Biased	Biased
	On	On	Off	Off	On	Off	On	Biased	Biased
Stand-by	Off	On	Off	Off	Off	Off	Off	Biased	Off
	Off	On	Off	Off	On	Off	On	Biased	Off

22 Programmable output clock divider

A specific register mapped on the XBUS can be used to choose the division factor on the CLKOUT signal (P3.15). This register is mapped on X-miscellaneous memory address range.

When CLKOUT function is enabled by setting bit CLKEN of register SYSCON, by default the CPU clock is output on P3.15. Setting bit XMISCEN of register XPERCON and bit XPEN of register SYSCON, it is possible to program the clock prescaling factor. In this way on P3.15 a prescaled value of the CPU clock can be output.

When CLKOUT function is not enabled (bit CLKEN of register SYSCON cleared), P3.15 does not output any clock signal, even though XCLKOUTDIV register is programmed.

23 Register set

This section summarizes all registers implemented in the ST10F272M, ordered by name.

23.1 Special function registers

Table 45 lists all SFRs which are implemented in the ST10F272M in alphabetical order. Bit-addressable SFRs are marked with the letter '**b**' in the column 'Name'.

SFRs within the extended sfr-space (ESFRs) are marked with the letter '**E**' in the 'Physical address' comumn.

Table 45. 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
CC4	FE88h	44h	CAPCOM register 4	0000h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
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
CC21	FE6Ah	35h	CAPCOM register 21	0000h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
CC21IC**b**	F16Ah**E**	B5h	CAPCOM register 21 interrupt control register	- - 00h
CC22	FE6Ch	36h	CAPCOM register 22	0000h
CC22IC**b**	F16Ch**E**	B6h	CAPCOM register 22 interrupt control register	- - 00h
CC23	FE6Eh	37h	CAPCOM register 23	0000h
CC23IC**b**	F16Eh**E**	B7h	CAPCOM register 23 interrupt control register	- - 00h
CC24	FE70h	38h	CAPCOM register 24	0000h
CC24IC**b**	F170h**E**	B8h	CAPCOM register 24 interrupt control register	- - 00h
CC25	FE72h	39h	CAPCOM register 25	0000h
CC25IC**b**	F172h**E**	B9h	CAPCOM register 25 interrupt control register	- - 00h
CC26	FE74h	3Ah	CAPCOM register 26	0000h
CC26IC**b**	F174h**E**	BAh	CAPCOM register 26 interrupt control register	- - 00h
CC27	FE76h	3Bh	CAPCOM register 27	0000h
CC27IC**b**	F176h**E**	BBh	CAPCOM register 27 interrupt control register	- - 00h
CC28	FE78h	3Ch	CAPCOM register 28	0000h
CC28IC**b**	F178h**E**	BCh	CAPCOM register 28 interrupt control register	- - 00h
CC29	FE7Ah	3Dh	CAPCOM register 29	0000h
CC29IC**b**	F184h**E**	C2h	CAPCOM register 29 interrupt control register	- - 00h
CC30	FE7Ch	3Eh	CAPCOM register 30	0000h
CC30IC**b**	F18Ch**E**	C6h	CAPCOM register 30 interrupt control register	- - 00h
CC31	FE7Eh	3Fh	CAPCOM register 31	0000h
CC31IC**b**	F194h**E**	CAh	CAPCOM register 31 interrupt control register	- - 00h
CCM0**b**	FF52h	A9h	CAPCOM mode control register 0	0000h
CCM1**b**	FF54h	AAh	CAPCOM mode control register 1	0000h
CCM2**b**	FF56h	ABh	CAPCOM mode control register 2	0000h
CCM3**b**	FF58h	ACh	CAPCOM mode control register 3	0000h
CCM4**b**	FF22h	91h	CAPCOM mode control register 4	0000h
CCM5**b**	FF24h	92h	CAPCOM mode control register 5	0000h
CCM6**b**	FF26h	93h	CAPCOM mode control register 6	0000h
CCM7**b**	FF28h	94h	CAPCOM mode control register 7	0000h
CP	FE10h	08h	CPU context pointer register	FC00h
CRIC**b**	FF6Ah	B5h	GPT2 CAPREL interrupt control register	- - 00h
CSP	FE08h	04h	CPU code segment pointer register (read only)	0000h
DP0L**b**	F100h**E**	80h	P0L direction control register	- - 00h
DP0H**b**	F102h**E**	81h	P0h direction control register	- - 00h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
DP1L**b**	F104h**E**	82h	P1L direction control register	- - 00h
DP1H**b**	F106h**E**	83h	P1h direction control register	- - 00h
DP2 **b**	FFC2h	E1h	Port 2 direction control register	0000h
DP3 **b**	FFC6h	E3h	Port 3 direction control register	0000h
DP4 **b**	FFCAh	E5h	Port 4 direction control register	- - 00h
DP6 **b**	FFCEh	E7h	Port 6 direction control register	- - 00h
DP7 **b**	FFD2h	E9h	Port 7 direction control register	- - 00h
DP8 **b**	FFD6h	EBh	Port 8 direction control register	- - 00h
DPP0	FE00h	00h	CPU data page pointer 0 register (10-bit)	0000h
DPP1	FE02h	01h	CPU data page pointer 1 register (10-bit)	0001h
DPP2	FE04h	02h	CPU data page pointer 2 register (10-bit)	0002h
DPP3	FE06h	03h	CPU data page pointer 3 register (10-bit)	0003h
EMUCON	FE0Ah	05h	Emulation control register	- - XXh
EXICON**b**	F1C0h**E**	E0h	External interrupt control register	0000h
EXISEL**b**	F1DAh**E**	EDh	External interrupt source selection register	0000h
IDCHIP	F07Ch**E**	3Eh	Device identifier register (n is the device revision)	110nh
IDMANUF	F07Eh**E**	3Fh	Manufacturer identifier register	0403h
IDMEM	F07Ah**E**	3Dh	On-chip memory identifier register	2040h
IDPROG	F078h**E**	3Ch	Programming voltage identifier register	0040h
IDX0**b**	FF08h	84h	MAC unit address pointer 0	0000h
IDX1**b**	FF0Ah	85h	MAC unit address pointer 1	0000h
MAH	FE5Eh	2Fh	MAC unit accumulator - high word	0000h
MAL	FE5Ch	2Eh	MAC unit accumulator - low word	0000h
MCW**b**	FFDCh	EEh	MAC unit control word	0000h
MDC**b**	FF0Eh	87h	CPU multiply divide control register	0000h
MDH	FE0Ch	06h	CPU multiply divide register – high word	0000h
MDL	FE0Eh	07h	CPU multiply divide register – low word	0000h
MRW**b**	FFDAh	EDh	MAC unit repeat word	0000h
MSW**b**	FFDEh	EFh	MAC unit status word	0200h
ODP2**b**	F1C2h**E**	E1h	Port 2 open drain control register	0000h
ODP3**b**	F1C6h**E**	E3h	Port 3 open drain control register	0000h
ODP4**b**	F1CAh**E**	E5h	Port 4 open drain control register	- - 00h
ODP6**b**	F1CEh**E**	E7h	Port 6 open drain control register	- - 00h
ODP7**b**	F1D2h**E**	E9h	Port 7 open drain control register	- - 00h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
ODP8**b**	F1D6h**E**	EBh	Port 8 open drain control register	- - 00h
ONES**b**	FF1Eh	8Fh	Constant value 1's register (read only)	FFFFh
P0L **b**	FF00h	80h	PORT0 low register (lower half of PORT0)	- - 00h
P0H **b**	FF02h	81h	PORT0 high register (upper half of PORT0)	- - 00h
P1L **b**	FF04h	82h	PORT1 low register (lower half of PORT1)	- - 00h
P1H **b**	FF06h	83h	PORT1 high register (upper half of PORT1)	- - 00h
P2 **b**	FFC0h	E0h	Port 2 register	0000h
P3 **b**	FFC4h	E2h	Port 3 register	0000h
P4 **b**	FFC8h	E4h	Port 4 register (8-bit)	- - 00h
P5 **b**	FFA2h	D1h	Port 5 register (read only)	XXXXh
P6 **b**	FFCCh	E6h	Port 6 register (8-bit)	- - 00h
P7 **b**	FFD0h	E8h	Port 7 register (8-bit)	- - 00h
P8 **b**	FFD4h	EAh	Port 8 register (8-bit)	- - 00h
P5DIDIS**b**	FFA4h	D2h	Port 5 digital disable register	0000h
PECC0	FEC0h	60h	PEC channel 0 control register	0000h
PECC1	FEC2h	61h	PEC channel 1 control register	0000h
PECC2	FEC4h	62h	PEC channel 2 control register	0000h
PECC3	FEC6h	63h	PEC channel 3 control register	0000h
PECC4	FEC8h	64h	PEC channel 4 control register	0000h
PECC5	FECAh	65h	PEC channel 5 control register	0000h
PECC6	FECCh	66h	PEC channel 6 control register	0000h
PECC7	FECEh	67h	PEC channel 7 control register	0000h
PICON**b**	F1C4h**E**	E2h	Port input threshold control register	- - 00h
PP0	F038h**E**	1Ch	PWM module period register 0	0000h
PP1	F03Ah**E**	1Dh	PWM module period register 1	0000h
PP2	F03Ch**E**	1Eh	PWM module period register 2	0000h
PP3	F03Eh**E**	1Fh	PWM module period register 3	0000h
PSW**b**	FF10h	88h	CPU program status word	0000h
PT0	F030h**E**	18h	PWM module up/down counter 0	0000h
PT1	F032h**E**	19h	PWM module up/down counter 1	0000h
PT2	F034h**E**	1Ah	PWM module up/down counter 2	0000h
PT3	F036h**E**	1Bh	PWM module up/down counter 3	0000h
PW0	FE30h	18h	PWM module pulse width register 0	0000h
PW1	FE32h	19h	PWM module pulse width register 1	0000h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
PW2	FE34h	1Ah	PWM module pulse width register 2	0000h
PW3	FE36h	1Bh	PWM module pulse width register 3	0000h
PWMCON0**b**	FF30h	98h	PWM module control register 0	0000h
PWMCON1**b**	FF32h	99h	PWM module control register 1	0000h
PWMIC**b**	F17Eh**E**	BFh	PWM module interrupt control register	- - 00h
QR0	F004h**E**	02h	MAC unit offset register r0	0000h
QR1	F006h**E**	03h	MAC unit offset register R1	0000h
QX0	F000h**E**	00h	MAC unit offset register X0	0000h
QX1	F002h**E**	01h	MAC unit offset register X1	0000h
RP0H**b**	F108h**E**	84h	System start-up configuration register (read only)	- - XXh
S0BG	FEB4h	5Ah	Serial channel 0 baudrate generator reload register	0000h
S0CON**b**	FFB0h	D8h	Serial channel 0 control register	0000h
S0EIC**b**	FF70h	B8h	Serial channel 0 error interrupt control register	- - 00h
S0RBUF	FEB2h	59h	Serial channel 0 receive buffer register (read only)	- - XXh
S0RIC**b**	FF6Eh	B7h	Serial channel 0 receive interrupt control register	- - 00h
S0TBIC**b**	F19Ch**E**	CEh	Serial channel 0 transmit buffer interrupt control reg.	- - 00h
S0TBUF	FEB0h	58h	Serial channel 0 transmit buffer register (write only)	0000h
S0TIC**b**	FF6Ch	B6h	Serial channel 0 transmit interrupt control register	- - 00h
SP	FE12h	09h	CPU system stack pointer register	FC00h
SSCBR	F0B4h**E**	5Ah	SSC baudrate register	0000h
SSCCON**b**	FFB2h	D9h	SSC control register	0000h
SSCEIC**b**	FF76h	BBh	SSC error interrupt control register	- - 00h
SSCRB	F0B2h**E**	59h	SSC receive buffer (read only)	XXXXh
SSCRIC**b**	FF74h	BAh	SSC receive interrupt control register	- - 00h
SSCTB	F0B0h**E**	58h	SSC transmit buffer (write only)	0000h
SSCTIC**b**	FF72h	B9h	SSC transmit interrupt control register	- - 00h
STKOV	FE14h	0Ah	CPU stack overflow pointer register	FA00h
STKUN	FE16h	0Bh	CPU stack underflow pointer register	FC00h
SYSCON**b**	FF12h	89h	CPU system configuration register	0xx0h(1)
T0	FE50h	28h	CAPCOM timer 0 register	0000h
T01CON**b**	FF50h	A8h	CAPCOM timer 0 and timer 1 control register	0000h
T0IC**b**	FF9Ch	CEh	CAPCOM timer 0 interrupt control register	- - 00h
T0REL	FE54h	2Ah	CAPCOM timer 0 reload register	0000h
T1	FE52h	29h	CAPCOM timer 1 register	0000h

Table 45. List of special function registers (continued)

Name	Physical address	8-bit address	Description	Reset value
T1IC**b**	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
T2CON**b**	FF40h	A0h	GPT1 timer 2 control register	0000h
T2IC**b**	FF60h	B0h	GPT1 timer 2 interrupt control register	- - 00h
T3	FE42h	21h	GPT1 timer 3 register	0000h
T3CON**b**	FF42h	A1h	GPT1 timer 3 control register	0000h
T3IC**b**	FF62h	B1h	GPT1 timer 3 interrupt control register	- - 00h
T4	FE44h	22h	GPT1 timer 4 register	0000h
T4CON**b**	FF44h	A2h	GPT1 timer 4 control register	0000h
T4IC**b**	FF64h	B2h	GPT1 timer 4 interrupt control register	- - 00h
T5	FE46h	23h	GPT2 timer 5 register	0000h
T5CON**b**	FF46h	A3h	GPT2 timer 5 control register	0000h
T5IC**b**	FF66h	B3h	GPT2 timer 5 interrupt control register	- - 00h
T6	FE48h	24h	GPT2 timer 6 register	0000h
T6CON**b**	FF48h	A4h	GPT2 timer 6 control register	0000h
T6IC**b**	FF68h	B4h	GPT2 timer 6 interrupt control register	- - 00h
T7	F050h**E**	28h	CAPCOM timer 7 register	0000h
T78CON**b**	FF20h	90h	CAPCOM timer 7 and 8 control register	0000h
T7IC**b**	F17Ah**E**	BDh	CAPCOM timer 7 interrupt control register	- - 00h
T7REL	F054h**E**	2Ah	CAPCOM timer 7 reload register	0000h
T8	F052h**E**	29h	CAPCOM timer 8 register	0000h
T8IC**b**	F17Ch**E**	BEh	CAPCOM timer 8 interrupt control register	- - 00h
T8REL	F056h**E**	2Bh	CAPCOM timer 8 reload register	0000h
TFR **b**	FFACh	D6h	Trap flag register	0000h
WDT	FEAEh	57h	Watchdog timer register (read only)	0000h
WDTCON**b**	FFAEh	D7h	Watchdog timer control register	00xxh(2)
XADRS3	F01Ch**E**	0Eh	XPER address select register 3	800Bh
XP0IC**b**	F186h**E**	C3h	See *Section 9.1*	- - 00h(3)
XP1IC**b**	F18Eh**E**	C7h	See *Section 9.1*	- - 00h(3)
XP2IC**b**	F196h**E**	CBh	See *Section 9.1*	- - 00h(3)
XP3IC**b**	F19Eh**E**	CFh	See *Section 9.1*	- - 00h(3)
XPERCON**b**	F024h**E**	12h	XPER configuration register	- - 05h
ZEROS**b**	FF1Ch	8Eh	Constant value 0's register (read only)	0000h

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 ST10F272M ordered by their name.

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

Table 46. List of X-bus 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
CAN1IF2CR	EF40h	CAN1: IF2 command request	0001h
CAN1IF2DA1	EF4Eh	CAN1: IF2 data A 1	0000h
CAN1IF2DA2	EF50h	CAN1: IF2 data A 2	0000h
CAN1IF2DB1	EF52h	CAN1: IF2 data B 1	0000h
CAN1IF2DB2	EF54h	CAN1: IF2 data B 2	0000h
CAN1IF2M1	EF44h	CAN1: IF2 Mask 1	FFFFh
CAN1IF2M2	EF46h	CAN1: IF2 mask 2	FFFFh

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
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	I^2C clock control register 1	0000h
I2CCCR2	EA0Eh	I^2C clock control register 2	0000h
I2CCR	EA00h	I^2C control register	0000h
I2CDR	EA0Ch	I^2C data register	0000h
I2COAR1	EA08h	I^2C own address register 1	0000h
I2COAR2	EA0Ah	I^2C own address register 2	0000h
I2CSR1	EA02h	I^2C status register 1	0000h
I2CSR2	EA04h	I^2C 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
RTCDH	ED0Ch	RTC divider counter high byte	XXXXh
RTCDL	ED0Ah	RTC divider counter low byte	XXXXh
RTCH	ED10h	RTC programmable counter high byte	XXXXh
RTCL	ED0Eh	RTC programmable counter low byte	XXXXh
RTCPH	ED08h	RTC prescaler register high byte	XXXXh
RTCPL	ED06h	RTC prescaler register low byte	XXXXh
XCLKOUTDIV	EB02h	CLKOUT divider control register	- - 00h
XEMU0	EB76h	XBUS emulation register 0 (write only)	XXXXh
XEMU1	EB78h	XBUS emulation register 1 (write only)	XXXXh
XEMU2	EB7Ah	XBUS emulation register 2 (write only)	XXXXh

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

Name	Physical address	Description	Reset value
XEMU3	EB7Ch	X-bus emulation register 3 (write only)	XXXXh
XIR0CLR	EB14h	X-interrupt 0 clear register (write only)	0000h
XIR0SEL	EB10h	X-interrupt 0 selection register	0000h
XIR0SET	EB12h	X-interrupt 0 set register (write only)	0000h
XIR1CLR	EB24h	X-interrupt 1 clear register (write only)	0000h
XIR1SEL	EB20h	X-interrupt 1 selection register	0000h
XIR1SET	EB22h	X-interrupt 1 set register (write only)	0000h
XIR2CLR	EB34h	X-interrupt 2 clear register (write only)	0000h
XIR2SEL	EB30h	X-interrupt 2 selection register	0000h
XIR2SET	EB32h	X-interrupt 2 set register (write only)	0000h
XIR3CLR	EB44h	X-interrupt 3 clear selection register (write only)	0000h
XIR3SEL	EB40h	X-interrupt 3 selection register	0000h
XIR3SET	EB42h	X-interrupt 3 set selection register (write only)	0000h
XMISC	EB46h	X-bus miscellaneous features register	0000h
XP1DIDIS	EB36h	Port 1 digital disable register	0000h
XPEREMU	EB7Eh	XPERCON copy for emulation (write only)	XXXXh
XPICON	EB26h	Extended port input threshold control register	- - 00h
XPOLAR	EC04h	XPWM module channel polarity register	0000h
XPP0	EC20h	XPWM module period register 0	0000h
XPP1	EC22h	XPWM module period register 1	0000h
XPP2	EC24h	XPWM module period register 2	0000h
XPP3	EC26h	XPWM module period register 3	0000h
XPT0	EC10h	XPWM module up/down counter 0	0000h
XPT1	EC12h	XPWM module up/down counter 1	0000h
XPT2	EC14h	XPWM module up/down counter 2	0000h
XPT3	EC16h	XPWM module up/down counter 3	0000h
XPW0	EC30h	XPWM module pulse width register 0	0000h
XPW1	EC32h	XPWM module pulse width register 1	0000h
XPW2	EC34h	XPWM module pulse width register 2	0000h
XPW3	EC36h	XPWM module pulse width register 3	0000h
XPWMCON0	EC00h	XPWM module control register 0	0000h
XPWMCON0CLR	EC08h	XPWM module clear control reg. 0 (write only)	0000h
XPWMCON0SET	EC06h	XPWM module set control register 0 (write only)	0000h
XPWMCON1	EC02h	XPWM module control register 1	0000h
XPWMCON1CLR	EC0Ch	XPWM module clear control reg. 0 (write only)	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

23.3 Flash registers ordered by name

The following table lists all Flash control registers which are implemented in the ST10F272M ordered by their name. These registers are physically mapped on the IBus, except for XFVTAUR0, which is mapped on X-bus. Note that these registers are not bit-addressable.

Table 47. 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	X-bus Flash volatile temporary access unprotection register 0	0000h

Note: XFVTAUR0 register is mapped on the X-bus in the XMiscellaneous window. Therefore XMISCEN, bit 10 of XPERCON register, must be set in order to access this register.

23.4 Identification registers

The ST10F272M has four identification registers, mapped in ESFR space. These registers contain:

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

IDMANUF (F07Eh / 3Fh) ESFR Reset value: 0403h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MANUF											0	0	0	1	1
RO											RO	RO	RO	RO	RO

Table 48. IDMANUF register description

Bit	Name	Function
15:5	MANUF	Manufacturer identifier 020h: STMicroelectronics manufacturer (JTAG worldwide normalization)

IDCHIP (F07Ch / 3Eh) ESFR Reset value: 110Xh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDCHIP												REVID			
RO												RO			

Table 49. IDCHIP register description

Bit	Name	Function
15:4	IDCHIP	Device identifier 110h: ST10F272M identifier (272)
3:0	REVID	Device revision identifier Xh: According to revision number

IDMEM (F07Ah / 3Dh) ESFR Reset value: 2040h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMTYP				MEMSIZE											
RO				RO											

Table 50. IDMEM register description

Bit	Name	Function
15:12	MEMSIZE	Internal memory size Internal memory size is 4 x (MEMSIZE) (in Kbyte) 040h for 256 Kbytes (ST10F272M)
11:0	MEMTYP	Internal memory type 0h: ROM-Less 1h: (M) ROM memory 2h: (S) Standard Flash memory (ST10F272M) 3h: (H) High performance Flash memory 4h...Fh: Reserved

IDPROG (F078h / 3Ch) ESFR Reset value: 0040h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROGVPP								PROGVDD							
RO								RO							

Table 51. IDPROG register description

Bit	Name	Function
15:8	PROGVPP	Programming V_{PP} voltage (no need of external V_{PP}) - 00h
7:0	PROGVDD	Programming V_{DD} voltage V_{DD} voltage when programming EPROM or Flash devices is calculated using the following formula: V_{DD} = 20 x [PROGVDD] / 256 (volts) - 40h for ST10F272M (5V).

Note: All identification words are read-only registers.

24 Electrical characteristics

24.1 Absolute maximum ratings

Table 52. Absolute maximum ratings

Symbol	Parameter	Values	Unit
V_{DD}	Voltage on V_{DD} pins with respect to ground (V_{SS})	-0.5 to +6.5	V
V_{STBY}	Voltage on V_{STBY} pin with respect to ground (V_{SS})	-0.5 to +6.5	V
V_{AREF}	Voltage on V_{AREF} pins with respect to ground (V_{SS})	-0.5 to V_{DD} + 0.5	V
V_{AGND}	Voltage on V_{AGND} pins with respect to ground (V_{SS})	V_{SS}	V
V_{IO}	Voltage on any pin with respect to ground (V_{SS})	-0.5 to V_{DD} + 0.5	V
I_{OV}	Input current on any pin during overload condition	± 10	mA
I_{TOV}	Absolute sum of all input currents during overload condition	| 75 |	mA
T_{ST}	Storage temperature	-65 to +150	°C
ESD	ESD susceptibility (Human body model)	2000	V

Note: Stresses above those listed under 'absolute maximum ratings' may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During overload conditions ($V_{IN} > V_{DD}$ or $V_{IN} < V_{SS}$) the voltage on pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.

During power-on and power-off transients (including standby entering/exiting phases), the relationships between voltages applied to the device and the main V_{DD} must always be respected. In particular, power-on and power-off of V_{AREF} must be coherent with V_{DD} transient, in order to avoid undesired current injection through the on-chip protection diodes.

24.2 Recommended operating conditions

Table 53. Recommended operating conditions

Symbol	Parameter	Value		Unit
		Min	Max	
V_{DD}	Operating supply voltage	4.5	5.5	V
V_{STBY}	Operation stand-by supply voltage(1)			
V_{AREF}	Operating analog reference voltage(2)	0	V_{DD} + 0.1	
T_A	Ambient temperature under bias	-40	+125	°C
T_J	Junction temperature under bias		+150	

1. The value of the V_{STBY} voltage is specified in the range of 4.5 to 5.5 volts. When V_{STBY} voltage is lower than main V_{DD}, the input section of V_{STBY}/EA pin can generate a spurious static consumption on V_{DD} 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 24.7*.

24.3 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 Watts. This is the chip internal power

$P_{I/O}$ represents the power dissipation on input and output pins; user determined.

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

Table 54. Thermal characteristics

Symbol	Description	Value (typical)	Unit
Θ_{JA}	Thermal resistance junction-ambient LQFP 144 - 20 x 20 mm/0.5 mm pitch LQFP 144 - 20 x 20 mm/0.5 mm pitch on four-layer FR4 board (2 layers signals/2 layers power)	40 35	°C/W

Based on thermal characteristics of the package and with reference to the power consumption figures provided in the next tables and diagrams, the following product classification can be proposed. However, the exact power consumption of the device inside the application must be computed according to different working conditions, thermal profiles, real thermal resistance of the system (including printed circuit board or other substrata), I/O activity, and so on.

Table 55. Package characteristics

Package	Ambient temperature range	CPU frequency range
LQFP 144	-40 to +125°C	1 to 40 MHz

24.4 Parameter interpretation

The parameters listed in the following tables represent the characteristics of the ST10F272M and its demands on the system.

Where the ST10F272M logic provides signals with their respective timing characteristics, the symbol '**CC**' for controller characteristics, is included in the 'Symbol' column. Where the external system must provide signals with their respective timing characteristics to the ST10F272M, the symbol '**SR**' for system requirement, is included in the 'Symbol' column.

24.5 DC characteristics

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

Table 56. DC characteristics

Parameter	Symbol		Limit values		Unit	Test condition
			Min	Max		
Input low voltage (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1, READY)	V_{IL}	SR	-0.3	0.8	V	–
Input low voltage (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1, READY)	V_{ILS}	SR	-0.3	0.3 V_{DD}	V	–
Input low voltage $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD	V_{IL1}	SR	-0.3	0.3 V_{DD}	V	–
Input low voltage XTAL1 (CMOS only)	V_{IL2}	SR	-0.3	0.3 V_{DD}	V	Direct drive mode
Input low voltage READY (TTL only)	V_{IL3}	SR	-0.3	0.8	V	–
Input high voltage (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1)	V_{IH}	SR	2.0	V_{DD} + 0.3	V	–
Input high voltage (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1)	V_{IHS}	SR	0.7 V_{DD}	V_{DD} + 0.3	V	–
Input high voltage $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD	V_{IH1}	SR	0.7 V_{DD}	V_{DD} + 0.3	V	–
Input high voltage XTAL1 (CMOS only)	V_{IH2}	SR	0.7 V_{DD}	V_{DD} + 0.3	V	Direct drive mode
Input high voltage READY (TTL only)	V_{IH3}	SR	2.0	V_{DD} + 0.3	V	–
Input hysteresis (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, XTAL1, RPD)	V_{HYS}	CC	400	700	mV	(1)
Input hysteresis (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, XTAL1, RPD)	V_{HYSS}	CC	750	1400	mV	(1)
Input hysteresis $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$	V_{HYS1}	CC	750	1400	mV	(1)
Input hysteresis XTAL1	V_{HYS2}	CC	0	50	mV	(1)
Input hysteresis READY (TTL only)	V_{HYS3}	CC	400	700	mV	(1)
Input hysteresis RPD	V_{HYS4}	CC	500	1500	mV	(1)
Output low voltage (P6[7:0], ALE, $\overline{RD}$, $\overline{WR}$/$\overline{WRL}$, $\overline{BHE}$/$\overline{WRH}$, CLKOUT, $\overline{RSTIN}$, $\overline{RSTOUT}$)	V_{OL}	CC	–	0.4 0.05	V	I_{OL} = 8 mA I_{OL} = 1 mA
Output low voltage (P0[15:0], P1[15:0], P2[15:0], P3[15,13:0], P4[7:0], P7[7:0], P8[7:0])	V_{OL1}	CC	–	0.4 0.05	V	I_{OL1} = 4 mA I_{OL1} = 0.5 mA
Output low voltage RPD	V_{OL2}	CC	–	V_{DD} 0.5 V_{DD} 0.3 V_{DD}	V	I_{OL2} = 85 µA I_{OL2} = 80 µA I_{OL2} = 60 µA
Output high voltage (P6[7:0], ALE, $\overline{RD}$, $\overline{WR}$/$\overline{WRL}$, $\overline{BHE}$/$\overline{WRH}$, CLKOUT, $\overline{RSTOUT}$)	V_{OH}	CC	V_{DD} - 0.8 V_{DD} - 0.08	–	V	I_{OH} = – 8mA I_{OH} = – 1mA

Table 56. DC characteristics (continued)

Parameter	Symbol		Limit values		Unit	Test condition
			Min	Max		
Output high voltage[2] (P0[15:0], P1[15:0], P2[15:0], P3[15,13:0], P4[7:0], P7[7:0], P8[7:0])	V_{OH1}	CC	V_{DD} - 0.8 V_{DD} - 0.08	–	V	I_{OH1} = – 4 mA I_{OH1} = – 0.5 mA
Output high voltage RPD	V_{OH2}	CC	0 0.3 V_{DD} 0.5 V_{DD}	–	V	I_{OH2} = – 2 mA I_{OH2} = – 750 µA I_{OH2} = – 150 µA
Input leakage current (P5[15:0])[3]	$\|I_{OZ1}\|$	CC	–	±0.2	µA	–
Input leakage current (all except P5[15:0], P2[0], RPD, P3[12], P3[15])	$\|I_{OZ2}\|$	CC	–	±0.5	µA	–
Input leakage current (P2[0])[4]	$\|I_{OZ3}\|$	CC	–	+1.0 -0.5	µA	–
Input leakage current (RPD)	$\|I_{OZ4}\|$	CC	–	±3.0	µA	–
Input leakage current (P3[12], P3[15])	$\|I_{OZ5}\|$	CC	–	±1.0	µA	–
Overload current (all except P2[0])	$\|I_{OV1}\|$	SR	–	±5	mA	(1)(5)
Overload current (P2[0])[4]	$\|I_{OV2}\|$	SR	–	+5 -1	mA	(1)(5)
$\overline{\text{RSTIN}}$ pull-up resistor	R_{RST}	CC	50	250	kΩ	100 kΩ nominal
Read/write inactive current[6][7]	I_{RWH}		–	-40	µA	V_{OUT} = 2.4 V
Read/write active current[6][8]	I_{RWL}		-500	–	µA	V_{OUT} = 0.4 V
ALE inactive current[6][7]	I_{ALEL}		20	–	µA	V_{OUT} = 0.4 V
ALE active current[6][8]	I_{ALEH}		–	300	µA	V_{OUT} = 2.4 V
Port 6 inactive current (P6[4:0])[6][7]	I_{P6H}		–	-40	µA	V_{OUT} = 2.4 V
Port 6 active current (P6[4:0])[6][8]	I_{P6L}		-500	–	µA	V_{OUT} = 0.4 V
PORT0 configuration current[6]	I_{P0H}[6]		–	-10	µA	V_{IN} = 2.0 V
	I_{P0L}[7]		-100	–	µA	V_{IN} = 0.8 V
Pin capacitance (digital inputs/outputs)	C_{IO}	CC	–	10	pF	(1)(6)
Run mode power supply current[9] (execution from internal RAM)	I_{CC1}		–	15 + 1.5 f_{CPU}	mA	–
Run mode power supply current[1][9] (execution from internal Flash)	I_{CC2}		–	15 + 1.5 f_{CPU}	mA	–
Idle mode supply current[10]	I_{ID}		–	15 + 0.6 f_{CPU}	mA	–
Power-down supply current[11] *(RTC off, oscillators off, Main voltage regulator off)*	I_{PD1}		–	150	µA	T_A = 25 °C
Power-down supply current[11] *(RTC on, main oscillator on, Main voltage regulator off)*	I_{PD2}		–	400 typical value	µA	T_A = 25 °C

Table 56. DC characteristics (continued)

Parameter	Symbol		Limit values		Unit	Test condition
			Min	Max		
Power-down supply current(11) *(RTC on, 32 kHz oscillator on, Main voltage regulator off)*	I_{PD3}		–	200	µA	T_A = 25 °C
Stand-by supply current(11) *(RTC off, oscillators off, V_{DD} off, V_{STBY} on)*	I_{SB1}		–	120	µA	V_{STBY} = 5.5 V $T_A = T_J$ = 25 °C
			–	500	µA	V_{STBY} = 5.5 V $T_A = T_J$ = 125 °C
Stand-by supply current(11) *(RTC on, 32 kHz oscillator on, main V_{DD} off, V_{STBY} on)*	I_{SB2}		–	120	µA	V_{STBY} = 5.5 V $T_A = T_J$ = 25 °C
			–	500	µA	V_{STBY} = 5.5 V $T_A = T_J$ = 125 °C
Stand-by supply current(1)(11) *(V_{DD} transient condition)*	I_{SB3}		–	2.5	mA	–

1. Not 100% tested, guaranteed by design characterization.
2. This specification is not valid for outputs which are switched to open drain mode. In this case the respective output will float and the voltage is imposed by the external circuitry.
3. Port 5 leakage values are granted for not selected A/D converter channel. One channel is always selected (by default, after reset, P5.0 is selected). For the selected channel the leakage value is similar to that of other port pins.
4. The leakage of P2.0 is higher than other pins due to the additional logic (pass gates active only in specific test modes) implemented on input path. Pay attention to not stress P2.0 input pin with negative overload beyond the specified limits: failures in Flash reading may occur (sense amplifier perturbation). Refer to next *Figure 37* for a scheme of the input circuitry.
5. Overload conditions occur if the standard operating conditions are exceeded, that is, the voltage on any pin exceeds the specified range (that is, $V_{OV} > V_{DD} + 0.3$ V or $V_{OV} < -0.3$ V). The absolute sum of input overload currents on all port pins may not exceed 50 mA. The supply voltage must remain within the specified limits.
6. This specification is only valid during reset, or during hold- or adapt-mode. Port 6 pins are only affected, if they are used for CS output and the open drain function is not enabled.
7. The maximum current may be drawn while the respective signal line remains inactive.
8. The minimum current must be drawn in order to drive the respective signal line active.
9. The power supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 38* below. This parameter is tested at V_{DDmax} and at maximum CPU clock frequency with all outputs disconnected and all inputs at V_{IL} or V_{IH}, RSTIN pin at V_{IH1min}. This implies I/O current is not considered. The device is doing the following:
   Fetching code from IRAM and XRAM1, accessing in read and write to both XRAM modules
   Watchdog timer is enabled and regularly serviced
   RTC is running with main oscillator clock as reference, generating a tick interrupts every 192 clock cycles
   Four channel of XPWM are running (waves period: 2, 2.5, 3 and 4 CPU clock cycles): no output toggling
   Five general purpose timers are running in timer mode with prescaler equal to 8 (T2, T3, T4, T5, T6)
   ADC is in autoscan continuous conversion mode on all 16 channels of port5
   All interrupts generated by XPWM, RTC, timers and ADC are not serviced
10. The idle mode supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 37* below. These parameters are tested and at maximum CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}, RSTIN pin at V_{IH1min}.
11. This parameter is tested including leakage currents. All inputs (including pins configured as inputs) at 0 V to 0.1 V or at V_{DD} - 0.1V to V_{DD}, V_{AREF} = 0V, all outputs (including pins configured as outputs) disconnected. Also, the main voltage regulator is assumed to be off; if it is not, an additional 1mA must be added.

Figure 37. Port2 test mode structure

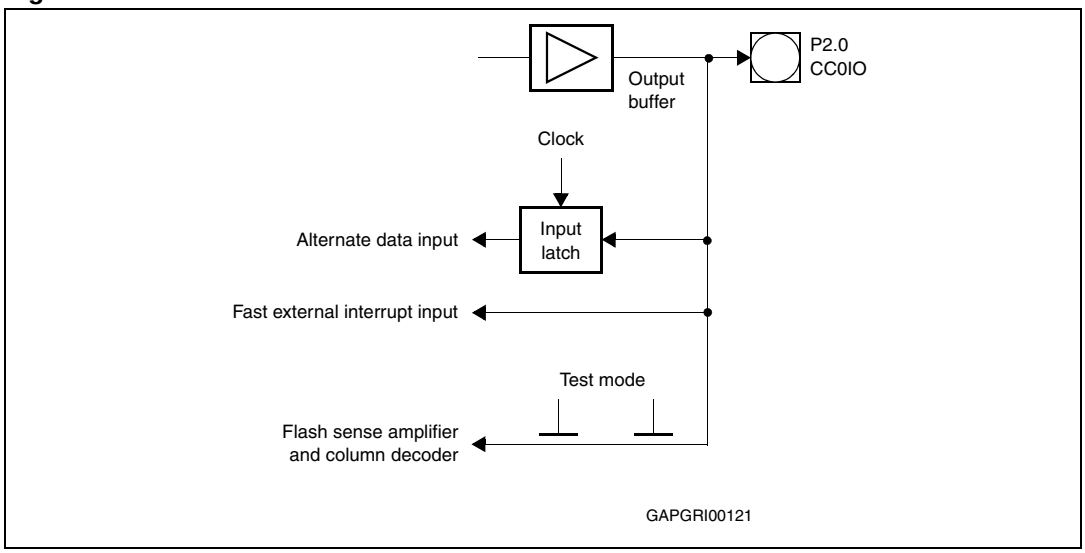
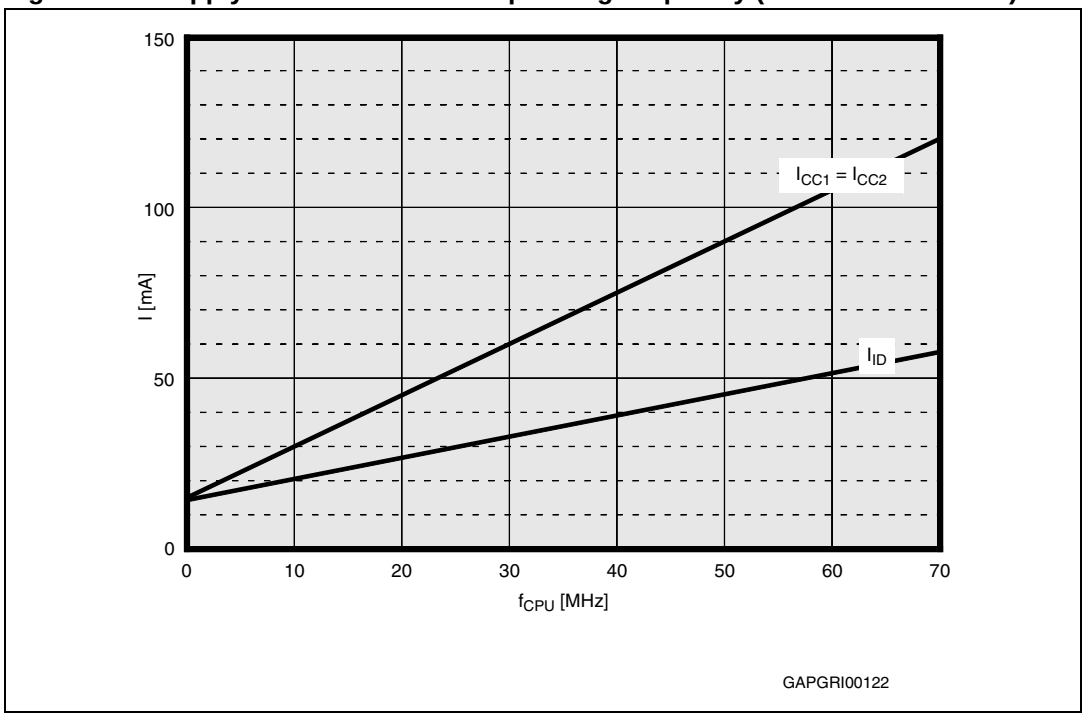


Figure 38. Supply current versus the operating frequency (run and idle modes)



24.6 Flash characteristics

V_{DD} = 5 V ± 10 %, V_{SS} = 0 V

Table 57. Flash characteristics

Parameter	Typical	Maximum		Unit	Notes
	T_A = 25 °C	T_A = 125 °C			
	0 cycles(1)	0 cycles(1)	100 k cycles		
Word program (32-bit) (2)	35	80	290	µs	–
Double word program (64-bit)(2)	60	150	570	µs	–
Bank 0 program (256 Kbyte) (double word program)	1.6	2.0	3.9	s	–
Sector erase (8 Kbyte)	0.6 0.5	0.9 0.8	1.0 0.9	s	Not preprogrammed Preprogrammed
Sector erase (32 Kbyte)	1.1 0.8	2.0 1.8	2.7 2.5	s	Not preprogrammed Preprogrammed
Sector erase (64 Kbyte)	1.7 1.3	3.7 3.3	5.1 4.7	s	Not preprogrammed Preprogrammed
Bank 0 erase (256 Kbyte)(3)	5.6 4.0	13.6 11.9	19.2 17.5	s	Not preprogrammed Preprogrammed
Recovery from power-down (t_{PD})	–	40	40	µs	(4)
Program suspend latency(4)	–	10	10	µs	–
Erase suspend latency(4)	–	30	30	µs	–
Erase suspend request rate(4)	20	20	20	ms	Minimum delay between two requests
Set protection(4)	40	90	300	µs	–

1. The figures are given after about 100 cycles due to testing routines (0 cycles at the final customer).
2. Word and double word programming times are provided as average values derived from a full sector programming time. Absolute value of a word or double word programming time could be longer than the average value.
3. Bank erase is obtained through a multiple sector erase operation (setting bits related to all sectors of the bank). As ST10F272M implements only one bank, the bank erase operation is equivalent to module and chip erase operations.
4. Not 100% tested, guaranteed by design characterization.

Table 58. Flash data retention characteristics

Number of program/erase cycles (-40 °C ≤ T_A ≤ 125 °C)	Data retention time (average ambient temperature 60 °C)	
	256 Kbyte (code store)	64 Kbyte (EEPROM emulation)(1)
0 - 100	> 20 years	> 20 years
1000	-	> 20 years
10000	-	10 years
100000	-	1 year

1. Two 64 Kbyte Flash sectors may be typically used to emulate up to 4, 8 or 16 Kbytes of EEPROM. Therefore, in case of an emulation of a 16 Kbyte EEPROM, 100,000 Flash program/erase cycles are equivalent to 800,000 EEPROM Program/Erase cycles. For an efficient use of the EEPROM emulation please refer to dedicated application note document (*AN2061 - "EEPROM Emulation with ST10F2xx"*). Contact your local field service, local sales person or STMicroelectronics representative to obtain a copy of such a guideline document.

24.7 A/D converter characteristics

V_{DD} = 5 V ± 10 %, V_{SS} = 0 V, T_A = -40 to +125 °C, 4.5 V ≤ V_{AREF} ≤ V_{DD}, $V_{SS} \le V_{AGND} \le V_{SS}$ + 0.2 V

Table 59. A/D converter characteristics

Parameter	Symbol		Limit values		Unit	Test condition
			Min	Max		
Analog reference voltage(1)	V_{AREF}	SR	4.5	V_{DD}	V	
Analog ground voltage	V_{AGND}	SR	V_{SS}	V_{SS} + 0.2	V	
Analog input voltage(2)	V_{AIN}	SR	V_{AGND}	V_{AREF}	V	
Reference supply current	I_{AREF}	CC	– –	5 1	mA µA	Running mode(3) Power-down mode
Sample time	t_S	CC	1	–	µs	(4)
Conversion time	t_C	CC	3	–	µs	(5)
Differential nonlinearity(6)	DNL	CC	-1	+1	LSB	No overload
Integral nonlinearity(6)	INL	CC	-1.5	+1.5	LSB	No overload
Offset error(6)	OFS	CC	-1.5	+1.5	LSB	No overload
Total unadjusted error(6)	TUE	CC	-2.0 -5.0 -7.0	+2.0 +5.0 +7.0	LSB	Port5 Port1 - no overload(3) Port1 - overload(3)
Coupling factor between inputs(3)(7)	K	CC	–	10^{-6}	–	On both Port5 and Port1
Input pin capacitance(3)(8)	C_{P1}	CC	–	3	pF	
	C_{P2}	CC	–	4 6	pF	Port5 Port1
Sampling capacitance(3)(8)	C_S	CC	–	3.5	pF	

Table 59. A/D converter characteristics (continued)

Parameter	Symbol		Limit values		Unit	Test condition
			Min	Max		
Analog switch resistance(3)(8)	R_{SW}	CC	– –	600 1600	W	Port5 Port1
	R_{AD}	CC	–	1300	W	

1. V_{AREF} can be tied to ground when A/D converter is not in use. There is increased consumption (approximately 200 µA) on main V_{DD} due to internal analog circuitry not being completely turned off. Therefore, it is suggested to maintain the V_{AREF} at V_{DD} level even when not in use, and to eventually switch off the A/D converter circuitry setting bit ADOFF in ADCON register.
2. V_{AIN} may exceed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases will be $0x000_H$ or $0x3FF_H$, respectively
3. Not 100% tested, guaranteed by design characterization
4. During the sample time the input capacitance C_{AIN} can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach its final voltage level within t_S. After the end of the sample time t_S, changes of the analog input voltage have no effect on the conversion result. Values for the sample clock t_S depend on programming and can be taken from *Table 60: A/D converter programming*.
5. This parameter includes the sample time t_S, the time for determining the digital result and the time to load the result register with the conversion result. Values for the conversion clock t_{CC} depend on programming and can be taken from the next *Table 60*.
6. DNL, INL, OFS and TUE are tested at V_{AREF} = 5.0 V, V_{AGND} = 0 V, V_{DD} = 5.0 V. It is guaranteed by design characterization for all other voltages within the defined voltage range. 'LSB' has a value of V_{AREF}/1024. For port5 channels, the specified TUE (± 2 LSB) is guaranteed also with an overload condition (see I_{OV} specification) occurring on maximum 2 not selected analog input pins of port5 and the absolute sum of input overload currents on all Port5 analog input pins does not exceed 10 mA. For port1 channels, the specified TUE is guaranteed when no overload condition is applied to port1 pins: when an overload condition occurs on maximum 2 not selected analog input pins of port1 and the input positive overload current on all analog input pins does not exceed 10 mA (either dynamic or static injection), the specified TUE is degraded (± 7 LSB). To obtain the same accuracy, the negative injection current on port1 pins must not exceed -1mA in case of both dynamic and static injection.
7. The coupling factor is measured on a channel while an overload condition occurs on the adjacent not selected channels with the overload current within the different specified ranges (for both positive and negative injection current).
8. Refer to scheme shown in *Figure 40*.

24.7.1 Conversion timing control

When a conversion is started, first the capacitances of the converter are loaded via the respective analog input pin to the current analog input voltage. The time to load the capacitances is referred to as sample time. Next the sampled voltage is converted to a digital value several successive steps, which correspond to the 10-bit resolution of the ADC. During these steps the internal capacitances are repeatedly charged and discharged via the V_{AREF} pin.

The current that has to be drawn from the sources for sampling and changing charges depends on the time that each respective step takes, because the capacitors must reach their final voltage level within the given time, at least with a certain approximation. The maximum current, however, that a source can deliver, depends on its internal resistance.

The time that the two different actions during conversion take (sampling, and converting) can be programmed within a certain range in the ST10F272M relative to the CPU clock. The absolute time that is consumed by the different conversion steps therefore is independent from the general speed of the controller. This allows adjustment of the ST10F272M A/D converter to the system's properties:

Fast conversion can be achieved by programming the respective times to their absolute possible minimum. This is preferable for scanning high frequency signals. The internal resistance of analog source and analog supply must be sufficiently low, however.

High internal resistance can be achieved by programming the respective times to a higher value, or the possible maximum. This is preferable when using analog sources and supply with a high internal resistance in order to keep the current as low as possible. The conversion rate in this case may be considerably lower, however.

The conversion times are programmed via the upper four bits of register ADCON. Bit fields ADCTC and ADSTC are used to define the basic conversion time and in particular the partition between sample phase and comparison phases. The table below lists the possible combinations. The timings refer to the unit TCL, where f_{CPU} = 1/2 TCL. A complete conversion time includes the conversion itself, the sample time and the time required to transfer the digital value to the result register.

Table 60. A/D converter programming

ADCTC	ADSTC	Sample	Comparison	Extra	Total conversion
00	00	TCL * 120	TCL * 240	TCL * 28	TCL * 388
00	01	TCL * 140	TCL * 280	TCL * 16	TCL * 436
00	10	TCL * 200	TCL * 280	TCL * 52	TCL * 532
00	11	TCL * 400	TCL * 280	TCL * 44	TCL * 724
11	00	TCL * 240	TCL * 480	TCL * 52	TCL * 772
11	01	TCL * 280	TCL * 560	TCL * 28	TCL * 868
11	10	TCL * 400	TCL * 560	TCL * 100	TCL * 1060
11	11	TCL * 800	TCL * 560	TCL * 52	TCL * 1444
10	00	TCL * 480	TCL * 960	TCL * 100	TCL * 1540
10	01	TCL * 560	TCL * 1120	TCL * 52	TCL * 1732
10	10	TCL * 800	TCL * 1120	TCL * 196	TCL * 2116
10	11	TCL * 1600	TCL * 1120	TCL * 164	TCL * 2884

Note: The total conversion time is compatible with the formula valid for ST10F269, while the meaning of the bit fields ADCTC and ADSTC is no longer compatible: the minimum conversion time is 388 TCL, which at 40 MHz CPU frequency corresponds to 4.85 µs (see ST10F269).

24.7.2 A/D conversion accuracy

The A/D converter compares the analog voltage sampled on the selected analog input channel to its analog reference voltage (V_{AREF}) and converts it into 10-bit digital data. The absolute accuracy of the A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error (OFS)
- Gain error (GE)
- Quantization error
- Nonlinearity error (differential and integral)

These four error quantities are explained below using *Figure 39*.

Offset error

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 00 to 01 (*Figure 39*, see OFS).

Gain error

Gain error is the deviation between the actual and ideal A/D conversion characteristics when the digital output value changes from the 3FE to the maximum 3FF, once offset error is subtracted. Gain error combined with offset error represents the so-called full-scale error (*Figure 39*, OFS + GE).

Quantization error

Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB.

Nonlinearity error

Nonlinearity error is the deviation between actual and the best-fitting A/D conversion characteristics (see *Figure 39*):

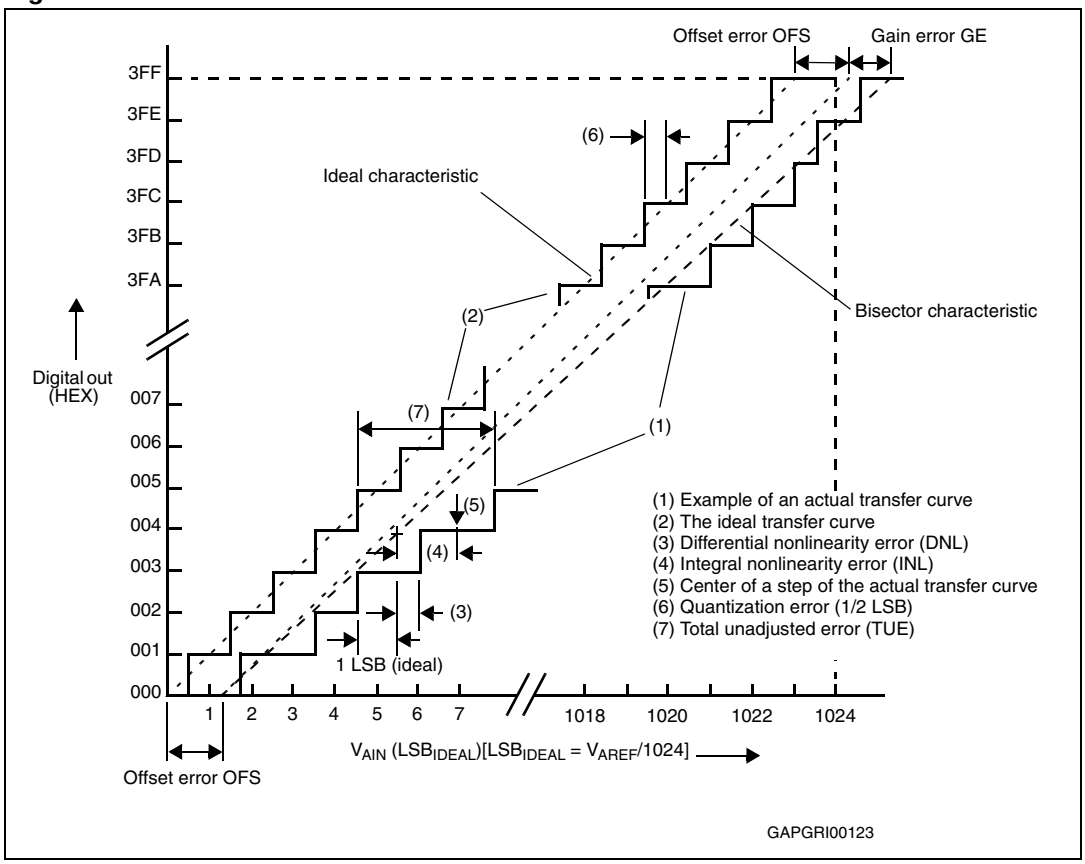
- Differential nonlinearity error is the actual step dimension versus the ideal one (1 LSB_{IDEAL}).
- Integral nonlinearity error is the distance between the center of the actual step and the center of the bisector line, in the actual characteristics. Note that for integral nonlinearity error, the effect of offset, gain and quantization errors is not included.

Note: Bisector characteristic is obtained drawing a line from 1/2 LSB before the first step of the real characteristic, and 1/2 LSB after the last step again of the real characteristic.

24.7.3 Total unadjusted error

The total unadjusted error specifies the maximum deviation from the ideal characteristic: the number provided in the data sheet represents the maximum error with respect to the entire characteristic. It is a combination of the offset, gain and integral linearity errors. The different errors may compensate each other depending on the relative sign of the offset and gain errors. Refer to *Figure 39*, see TUE.

Figure 39. A/D conversion characteristics



24.7.4 Analog reference pins

The accuracy of the A/D converter depends on how accurate is its analog reference: a noise in the reference results in at least that much error in a conversion. A low pass filter on the A/D converter reference source (supplied through pins V_{AREF} and V_{AGND}), is recommended in order to clean the signal, minimizing the noise. A simple capacitive bypassing may be sufficient in most of the cases; in presence of high RF noise energy, inductors or ferrite beads may be necessary.

In this architecture, V_{AREF} and V_{AGND} pins represents also the power supply of the analog circuitry of the A/D converter: there is an effective DC current requirement from the reference voltage by the internal resistor string in the R-C DAC array and by the rest of the analog circuitry.

An external resistance on V_{AREF} could introduce error under certain conditions: for this reasons, series resistance are not advisable, and more in general any series devices in the filter network should be designed to minimize the DC resistance.

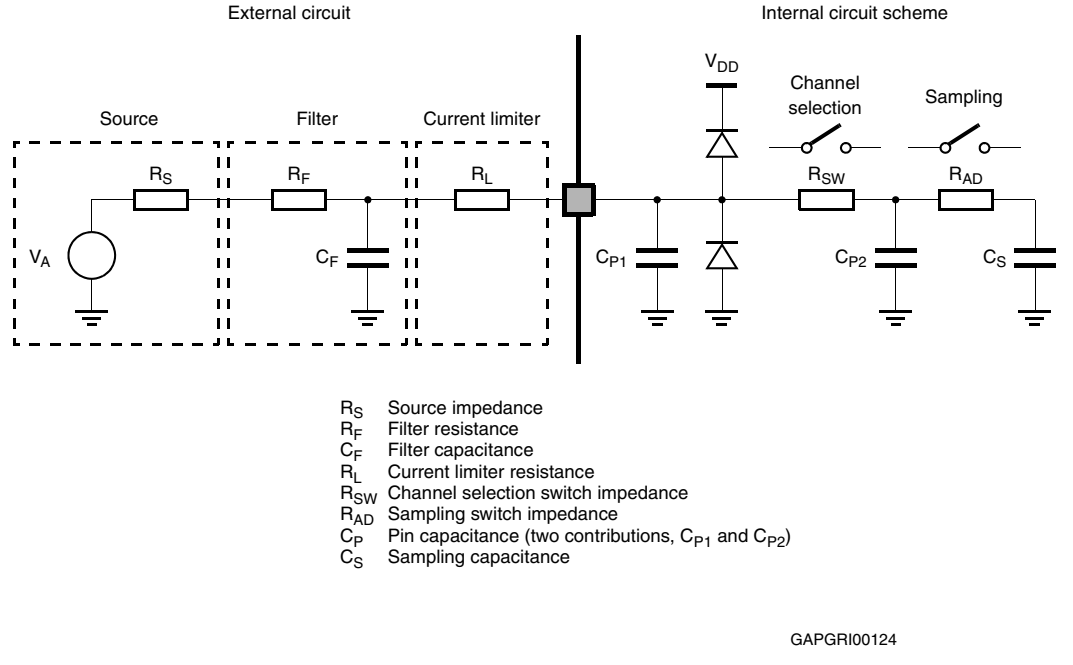
Analog input pins

To improve the accuracy of the A/D converter, it is definitively necessary that analog input pins have low AC impedance. Placing a capacitor with good high frequency characteristics at the input pin of the device, can be effective: the capacitor should be as large as possible, ideally infinite. This capacitor contributes to attenuating the noise present on the input pin.

Moreover, it sources charge during the sampling phase, when the analog signal source is a high-impedance source.

A real filter, can typically be obtained by using a series resistance with a capacitor on the input pin (simple RC Filter). The RC filtering may be limited according to the value of source impedance of the transducer or circuit supplying the analog signal to be measured. The filter at the input pins must be designed taking into account the dynamic characteristics of the input signal (bandwidth).

Figure 40. A/D converter input pins scheme



Input leakage and external circuit

The series resistor utilized to limit the current to a pin (see R_L in *Figure 40*), in combination with a large source impedance can lead to a degradation of A/D converter accuracy when input leakage is present.

Data about maximum input leakage current at each pin is provided in *Table 56: DC characteristics on page 126*. Input leakage is greatest at high operating temperatures, and in general it decreases by one half for each 10°C decrease in temperature.

Considering that, for a 10-bit A/D converter one count is about 5mV (assuming V_{AREF} = 5 V), an input leakage of 100 nA acting though an R_L = 50 kΩ of external resistance leads to an error of exactly one count (5 mV); if the resistance were 100 kΩ the error would become two counts.

Eventual additional leakage due to external clamping diodes must also be taken into account in computing the total leakage affecting the A/D converter measurements. Another contribution to the total leakage is represented by the charge sharing effects with the sampling capacitance: being C_S substantially a switched capacitance, with a frequency equal to the conversion rate of a single channel (maximum when fixed channel continuous conversion mode is selected), it can be seen as a resistive path to ground. For instance, assuming a conversion rate of 250 kHz, with C_S equal to 4 pF, a resistance of 1 MΩ is obtained ($R_{EQ} = 1/f_C C_S$, where f_C represents the conversion rate at the considered

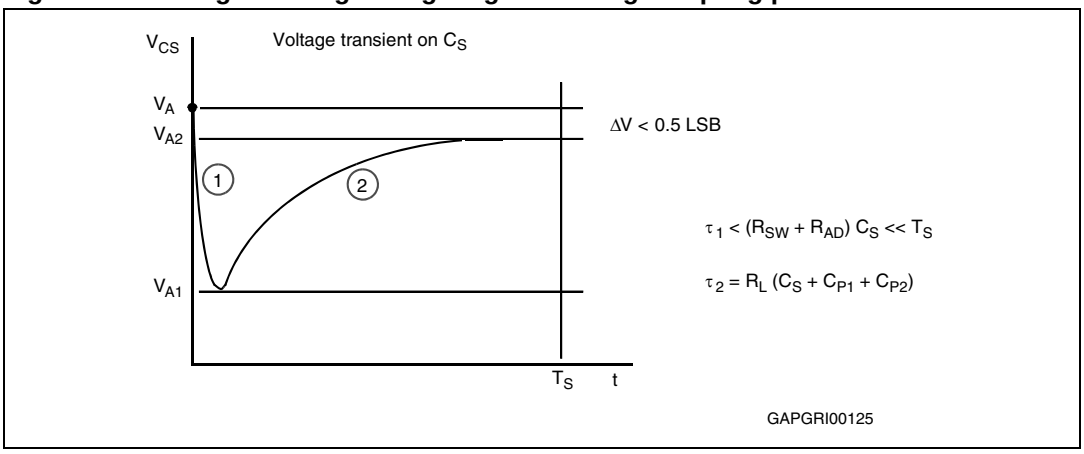
channel). To minimize the error induced by the voltage partitioning between this resistance (sampled voltage on C_S) and the sum of $R_S + R_F + R_L + R_{SW} + R_{AD}$, the external circuit must be designed to respect the following relation:

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

The formula above provides constraints for external network design, in particular on resistive path.

A second aspect involving the capacitance network must be considered. Assuming the three capacitances C_F, C_{P1} and C_{P2} initially charged at the source voltage V_A (refer to the equivalent circuit shown in *Figure 40*), when the sampling phase is started (A/D switch close), a charge sharing phenomena is installed.

Figure 41. Charge sharing timing diagram during sampling phase



In particular two different transient periods can be distinguished (see *Figure 41*):

- A first and quick charge transfer from the internal capacitance C_{P1} and C_{P2} to the sampling capacitance C_S occurs (C_S is supposed initially completely discharged): considering a worst case (since the time constant in reality would be faster) in which C_{P2} is shown in parallel to C_{P1} (call $C_P = C_{P1} + C_{P2}$), the two capacitance C_P and C_S are in series, and the time constant is:

$$\tau_1 = (R_{SW} + R_{AD}) \cdot \frac{C_P \cdot C_S}{C_P + C_S}$$

This relation can again be simplified considering only C_S as an additional worst condition. In reality, the transient is faster, but the A/D converter circuitry has been designed to be robust also in the very worst case: the sampling time T_S is always much longer than the internal time constant:

$$\tau_1 < (R_{SW} + R_{AD}) \cdot C_S < \quad < T_S$$

The charge of C_{P1} and C_{P2} is redistributed also on C_S, determining a new value of the voltage V_{A1} on the capacitance according to the following equation:

$$V_{A1} \cdot (C_S + C_{P1} + C_{P2}) = V_A \cdot (C_{P1} + C_{P2})$$

- A second charge transfer involves also C_F (that is typically bigger than the on-chip capacitance) through the resistance R_L: again considering the worst case in which C_{P2} and C_S were in parallel to C_{P1} (since the time constant in reality would be faster), the time constant is:

$$\tau_2 < R_L \cdot (C_S + C_{P1} + C_{P2})$$

In this case, the time constant depends on the external circuit: in particular imposing that the transient is completed well before the end of sampling time T_S, a constraint on R_L sizing is obtained:

$$10 \cdot \tau_2 = 10 \cdot R_L \cdot (C_S + C_{P1} + C_{P2}) \leq T_S$$

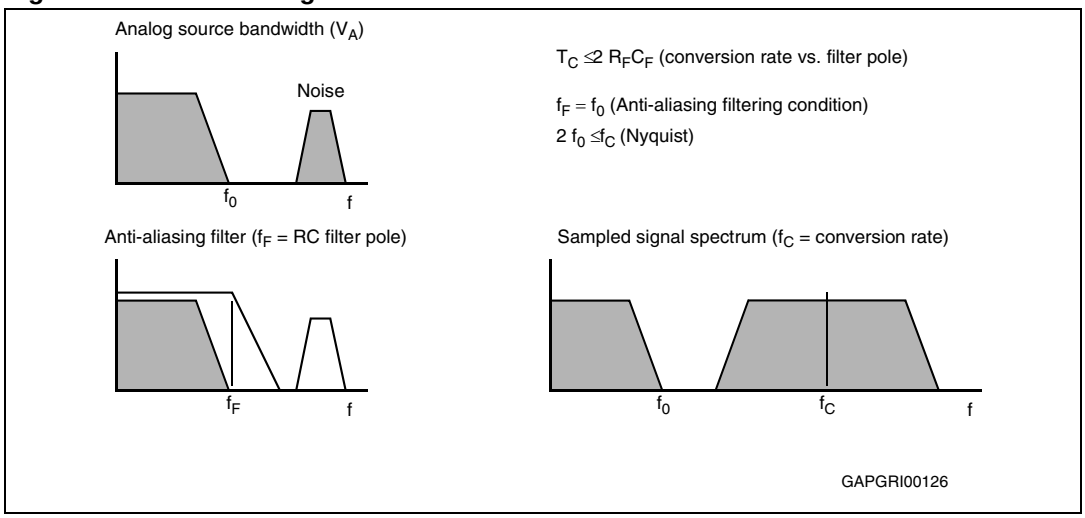
Of course, R_L must also be sized according to the current limitation constraints, in combination with R_S (source impedance) and R_F (filter resistance). Being C_F definitively bigger than C_{P1}, C_{P2} and C_S, then the final voltage V_{A2} (at the end of the charge transfer transient) will be much higher than V_{A1}. The following equation must be respected (charge balance assuming now C_S already charged at V_{A1}):

$$V_{A2}(C_S + C_{P1} + C_{P2} + C_F) = V_A \cdot C_F + V_{A1}(C_{P1} + C_{P2} + C_S)$$

The two transients above are not influenced by the voltage source that, due to the presence of the R_FC_F filter, is not able to provide the extra charge to compensate the voltage drop on C_S with respect to the ideal source V_A; the time constant R_FC_F of the filter is very high with respect to the sampling time (T_S). The filter is typically designed to act as anti-aliasing (see *Figure 42*).

Calling f_0 the bandwidth of the source signal (and as a consequence the cut-off frequency of the anti-aliasing filter, f_F), according to Nyquist theorem the conversion rate f_C must be at least $2f_0$; it means that the constant time of the filter is greater than or at least equal to twice the conversion period (T_C). Again the conversion period T_C is longer than the sampling time T_S, which is just a portion of it, even when fixed channel continuous conversion mode is selected (fastest conversion rate at a specific channel): in conclusion it is evident that the time constant of the filter R_FC_F is definitively much higher than the sampling time T_S, so the charge level on C_S cannot be modified by the analog signal source during the time in which the sampling switch is closed.

Figure 42. Anti-aliasing filter and conversion rate



The considerations above imposes new constraints on the external circuit, to reduce the accuracy error due to the voltage drop on C_S. From the two charge balance equations above, it is simple to derive the following relation between the ideal and real sampled voltage on C_S:

$$\frac{V_A}{V_{A2}} = \frac{C_{P1} + C_{P2} + C_F}{C_{P1} + C_{P2} + C_F + C_S}$$

From this formula, in the worst case (when V_A is maximum, that is for instance 5 V), assuming to accept a maximum error of half a count (~2.44 mV), a constraint is immediately evident on C_F value:

$$C_F > 2048\ C_S$$

In the next section an example of how to design the external network is provided, assuming some reasonable values for the internal parameters and making a hypothesis on the characteristics of the analog signal to be sampled.

Example of external network sizing

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

- Analog signal source bandwidth (f_0): 10 kHz
- Conversion rate (f_C): 25 kHz
- Sampling time (T_S): 1µs
- Pin input capacitance (C_{P1}): 5 pF
- Pin input routing capacitance (C_{P2}): 1 pF
- Sampling capacitance (C_S): 4pF
- Maximum input current injection (I_{INJ}): 3 mA
- Maximum analog source voltage ($V_{AM)}$): 12 V
- Analog source impedance (R_S): 100 Ω
- Channel switch resistance (R_{SW}): 500 Ω
- Sampling switch resistance (R_{AD}): 200 Ω

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

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

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:

$$C_F = 4000\, C_S = 16nF$$

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

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

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

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

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

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

5. Now the three elements 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 allows minimization of the accuracy error introduced by the switched capacitance equivalent resistance is in this case:

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

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

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

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

$$\tau_1 = (R_{SW} + R_{AD}) \cdot C_S = 2.8ns \quad << T_S = 1\mu s$$

$$10\,\tau_2 = 10\,R_L (C_S + C_{P1} + C_{P2}) = 290ns \quad < T_S = 1\mu s$$

For the complete set of parameters characterizing the ST10F272M A/D converter equivalent circuit, refer to *Section 24.7: A/D converter characteristics on page 131*.

24.8 AC characteristics

24.8.1 Test waveforms

Figure 43. Input/output waveforms

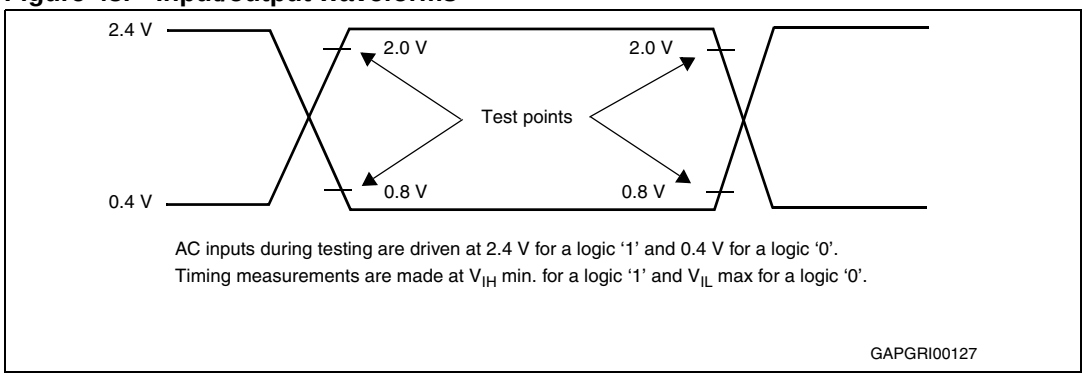
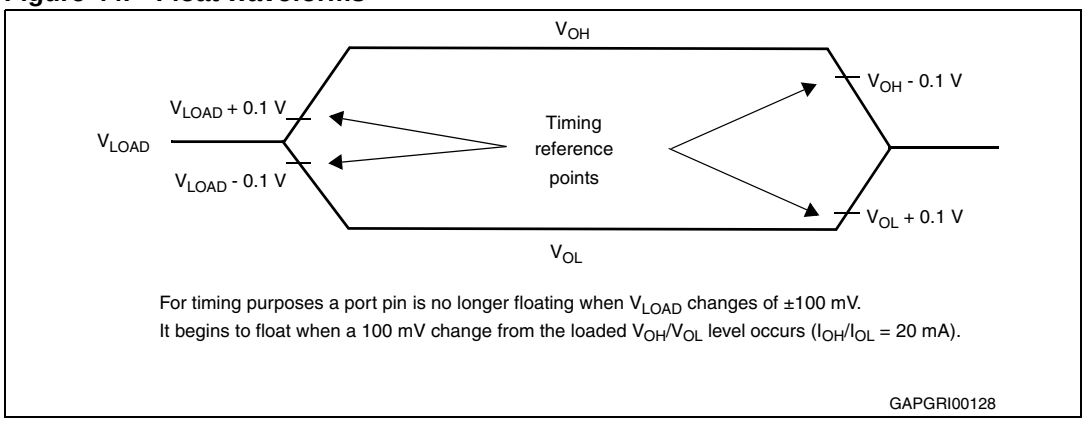


Figure 44. Float waveforms



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'.

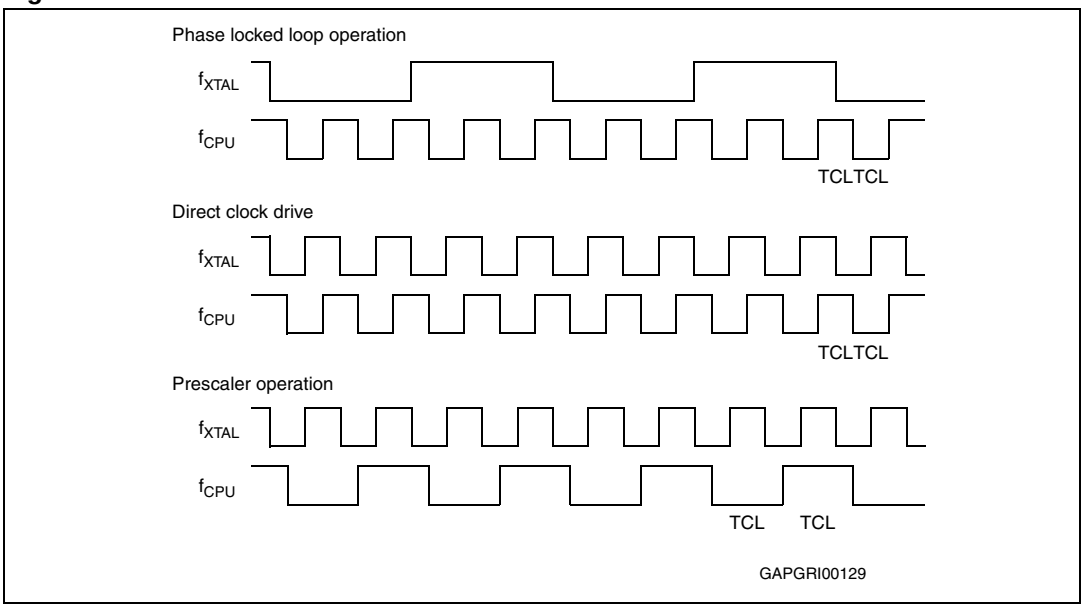
The CPU clock signal can be generated by different mechanisms. The duration of TCL and its variation (and also the derived external timing) depends on the mechanism used to generate f_{CPU}.

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).

Figure 45. Generation mechanisms for the CPU clock



24.8.3 Clock generation modes

The next *Table 61* associates the combinations of these three bits with the respective clock generation mode.

Table 61. On-chip clock generator selections

P0.15-13 (P0H.7-5)			CPU frequency $f_{CPU} = f_{XTAL} \times F$	External clock input range[1][3]	Notes
1	1	1	$f_{XTAL} \times 4$	4 to 8 MHz	Default configuration
1	1	0	$f_{XTAL} \times 3$	5.3 to 8 MHz	
1	0	1	$f_{XTAL} \times 8$	4 to 5 MHz	
1	0	0	$f_{XTAL} \times 5$	6.4 to 8 MHz	
0	1	1	$f_{XTAL} \times 1$	1 to 40 MHz	Direct drive (oscillator bypassed)[2]
0	1	0	$f_{XTAL} \times 10$	4 MHz	
0	0	1	$f_{XTAL}/2$	4 to 8 MHz	CPU clock via prescaler[3]
0	0	0	-	-	Reserved

1. The external clock input range refers to a CPU clock range of 1...40 MHz. Moreover, the PLL usage is limited to 4-8 MHz. All configurations need a crystal (or ceramic resonator) to generate the CPU clock through the internal oscillator amplifier (apart from direct drive). Vice versa, the clock can be forced through an external clock source only in direct drive mode (on-chip oscillator amplifier disabled, so no crystal or resonator can be used).
2. The maximum depends on the duty cycle of the external clock signal. When 40 MHz is used, 50% duty cycle is granted (low phase = high phase = 12.5 ns); when 20 MHz is selected a 25 % duty cycle can be accepted (minimum phase, high or low, again equal to 12.5 ns).
3. The limits on input frequency are 4-8 MHz since the usage of the internal oscillator amplifier is required. Also when the PLL is not used and the CPU clock corresponds to $f_{XTAL}/2$, an external crystal or resonator must be used: It is not possible to force any clock though an external clock source.

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.

24.8.8 Voltage controlled oscillator

The ST10F272M implements a PLL which combines different levels of frequency dividers with a voltage controlled oscillator (VCO) working as frequency multiplier. The following table gives a detailed summary of the internal settings and VCO frequency.

Table 62. 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 8 MHz	$f_{XTAL}/4$	64	4	–	$f_{XTAL} \times 4$
1	1	0	5.3 to 8 MHz(1)	$f_{XTAL}/4$	48	4	–	$f_{XTAL} \times 3$
1	0	1	4 to 5 MHz	$f_{XTAL}/4$	64	2	–	$f_{XTAL} \times 8$
1	0	0	6.4 to 8 MHz(1)	$f_{XTAL}/4$	40	2	–	$f_{XTAL} \times 5$
0	1	1	1 to 40 MHz	–	PLL bypassed		–	$f_{XTAL} \times 1$
0	1	0	4 MHz	$f_{XTAL}/2$	40	2	–	$f_{XTAL} \times 10$
0	0	1	4 to 8 MHz(1)	–	PLL bypassed		$f_{PLL}/2$	$f_{XTAL}/2$
0	0	0	–					

1. The PLL input frequency range is limited to 1 to 3.5 MHz, while the VCO oscillation range is 64 to 128 MHz. The CPU clock frequency range when PLL is used is 16 to 40 MHz.

Example 1

- f_{XTAL} = 4 MHz
- P0(15:13) = ‘110’ (multiplication by 3)
- PLL input frequency = 1 MHz
- VCO frequency = 48 MHz: **NOT VALID**, must be 64 to 128 MHz
- f_{CPU} = **NOT VALID**

Example 2

- f_{XTAL} = 8 MHz
- P0(15:13) = ‘100’ (multiplication by 5)
- PLL input frequency = 2 MHz
- VCO frequency = 80 MHz
- PLL output frequency = 40 MHz (VCO frequency divided by 2)
- f_{CPU} = 40 MHz (no effect of output prescaler)

24.8.9 PLL jitter

The following terminology is defined below:

- **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 sufficiently large to have the long term jitter. For N = 1, this becomes the single period jitter.

Jitter at the PLL output can be due to the following reasons:

- Jitter in the input clock
- Noise in the PLL loop

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 the loop bandwidth is, the lower the jitter is due to noise in the loop. Moreover, the long term jitter is practically independent of 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 noise, thus, 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 RMS 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 RMS value of the accumulated jitter is *proportional to N*, where N is the number of clock periods within the considered time interval.

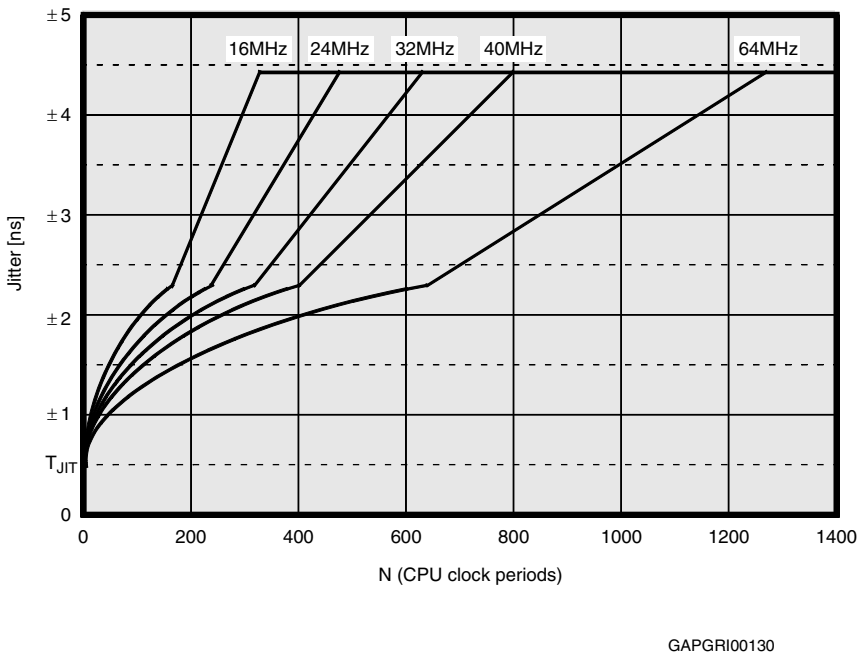
The jitter in the PLL loop can be modelized as dominated by the $i1/f^2$ noise for N smaller than a certain value depending on the PLL output frequency and on the bandwidth characteristics of loop. Above this first value, the jitter becomes dominated by the $i1/f^3$ noise component. Lastly, for N greater 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 46* the maximum jitter trend versus the number of clock periods N (for some typical CPU frequencies) is shown: 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 of the multiplication factor. Its effects are strongly reduced thanks to the particular care used in the physical implementation and integration of the PLL module inside the device. Nonetheless, the contribution of the digital noise to the global jitter is widely taken into account in the curves provided in *Figure 46*.

Figure 46. ST10F272M 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 recovery from a crystal failure occurrence without risking to go into 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 $\overline{\text{RSTIN}}$ pin must be properly sized in order to extend the duration of the low pulse to lock the PLL before the level at $\overline{\text{RSTIN}}$ pin is recognized high: A

bidirectional reset internally drives the $\overline{RSTIN}$ pin low for just 1024 TCL (definitly not sufficient to lock the PLL starting from free-running mode).

Table 63. PLL characteristics (V_{DD} = 5 V ± 10 %, V_{SS} = 0 V, T_A = -40 to +125 °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	250	2000	kHz
		Multiplication factors: 5, 8, 10, 16	500	4000	

1. Not 100 % tested, guaranteed by design characterization

24.8.11 Main oscillator specifications

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

Table 64. 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 47. Crystal oscillator and resonator connection diagram

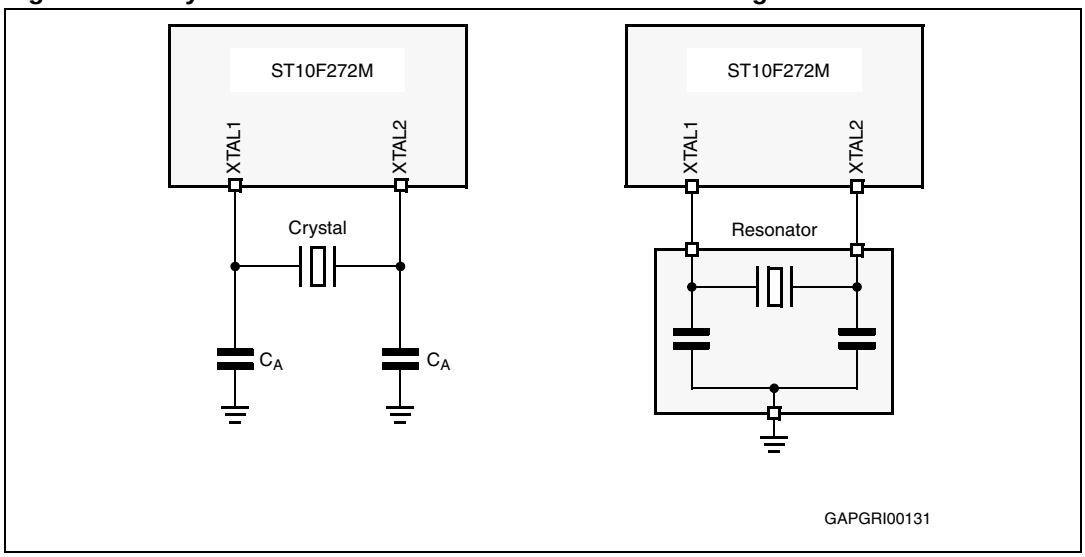


Table 65. Main oscillator negative resistance (module)

	C_A = 15 pF			C_A = 25 pF			C_A = 35 pF		
	Min	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
4 MHz	545 Ω	1035 Ω	–	550 Ω	1050 Ω	–	430 Ω	850 Ω	–
8 MHz	240 Ω	450 Ω	–	170 Ω	350 Ω	–	120 Ω	250 Ω	–

The given values of C_A do not include the stray capacitance of the package and of the printed circuit board: the negative resistance values are calculated assuming additional 5 pF to the values in the table. The crystal shunt capacitance (C_0) and the package capacitance between XTAL1 and XTAL2 pins is globally assumed equal to 10 pF.

The external resistance between XTAL1 and XTAL2 is not necessary, since already present on the silicon.

24.8.12 32 kHz oscillator specifications

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

Table 66. 32 kHz oscillator characteristics

Symbol	Parameter	Conditions	Value			Unit
			Min	Typ	Max	
g_{m32}	Oscillator transconductance(1)	Startup	20	31	50	µA/V
		Normal run	8	17	30	µA/V
V_{OSC32}	Oscillation amplitude(2)	Peak to peak	0.5	1.0	2.4	V
V_{AV32}	Oscillation voltage level(2)	Sine wave middle	0.7	0.9	1.2	V
t_{STUP32}	Oscillator startup time(2)	Stable V_{DD}	–	1	5	s

1. At power-on a high current biasing is applied for faster oscillation start-up. Once the oscillation is started, the current biasing is reduced to lower the power consumption of the system.
2. Not 100% tested, guaranteed by design characterization.

Figure 48. 32 kHz crystal oscillator connection diagram

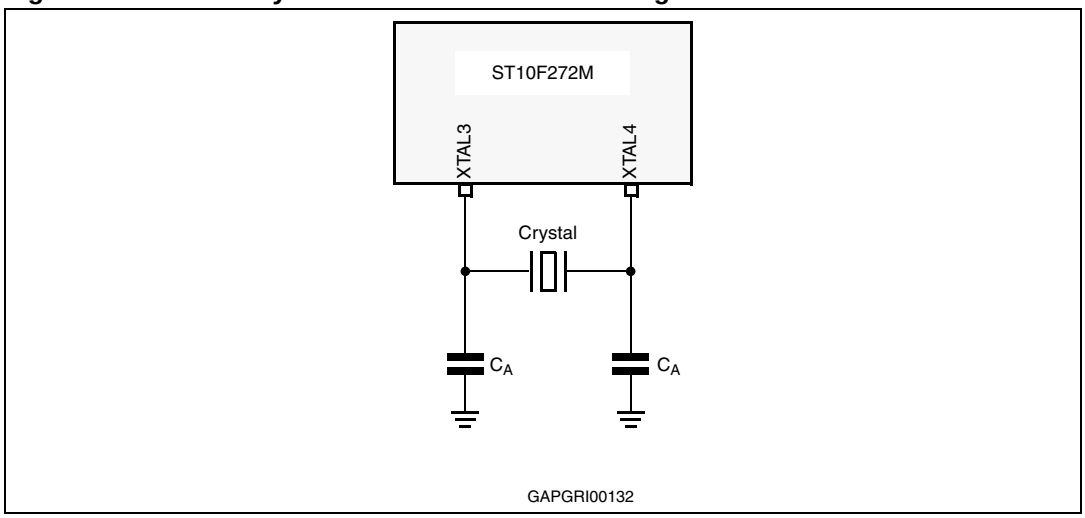


Table 67. Minimum values of negative resistance (module) for 32 kHz oscillator

Frequency	C_A = 6 pF	C_A = 12 pF	C_A = 15 pF	C_A = 18 pF	C_A = 22 pF	C_A = 27 pF	C_A = 33 pF
32 kHz	-	-	-	-	150 kΩ	120 kΩ	90 kΩ

The given values of C_A do not include the stray capacitance of the package and of the printed circuit board: the negative resistance values are calculated assuming additional 5 pF to the values in the table. The crystal shunt capacitance (C_0) and the package capacitance between XTAL3 and XTAL4 pins is globally assumed equal to 4 pF. The external resistance between XTAL3 and XTAL4 is not necessary, since already present on the silicon.

Warning: Direct driving on XTAL3 pin is not supported. Always use a 32 kHz crystal oscillator.

24.8.13 External clock drive XTAL1

When direct drive configuration is selected during reset, it is possible to drive the CPU clock directly from the XTAL1 pin, without particular restrictions on the maximum frequency, since the on-chip oscillator amplifier is bypassed. The speed limit is imposed by internal logic that targets a maximum CPU frequency of 40 MHz.

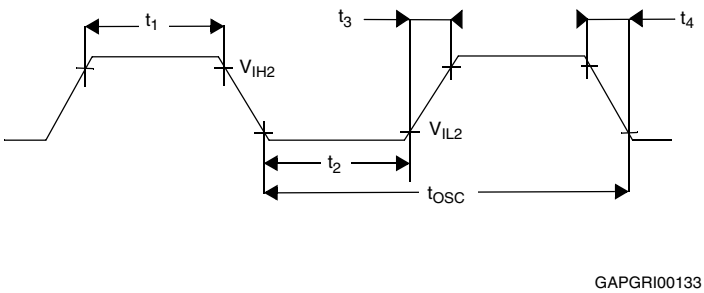
In all other clock configurations (direct drive with prescaler or PLL usage) the on-chip oscillator amplifier is not bypassed, so it determines the input clock speed limit. Then, when the on-chip oscillator is enabled it is forbidden to use any external clock source different from crystal or ceramic resonator.

Table 68. External clock drive

Parameter	Symbol		Direct drive $f_{CPU} = f_{XTAL}$		Direct drive with prescaler $f_{CPU} = f_{XTAL}/2$		PLL usage $f_{CPU} = f_{XTAL} \times F$		Unit
			Min	Max	Min	Max	Min	Max	
XTAL1 period(1)(2)	t_{OSC}	SR	25	–	83.3	250	83.3	250	ns
High time(3)	t_1	SR	6	–	3	–	6	–	ns
Low time(3)	t_2	SR	6	–	3	–	6	–	ns
Rise time(3)	t_3	SR	–	2	–	2	–	2	ns
Fall time(3)	t_4	SR	–	2	–	2	–	2	ns

1. The minimum value for the XTAL1 signal period is considered the theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal.
2. 4 to 8 MHz is the input frequency range when using an external clock source. 40 MHz can be applied with an external clock source only when direct drive mode is selected: in this case, the oscillator amplifier is bypassed so it does not limit the input frequency.
3. The input clock signal must reach the defined levels V_{IL2} and V_{IH2}.

Figure 49. 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 40 MHz is used, 50% duty cycle is granted (low phase = high phase = 12.5 ns); when for instance 20 MHz is used, a 25 % duty cycle can be accepted (minimum phase, high or low, again equal to 12.5 ns).

24.8.14 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 69. 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])

24.8.15 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.

Note: All external memory bus timings and SSC timings listed in the following tables are granted by design characterization and not fully tested in production.

24.8.16 Multiplexed bus

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

Table 70. Multiplexed bus timings

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 40 MHz		Unit
			Min	Max	Min	Max	
t_5	CC	ALE high time	4 + t_A	–	TCL – 8.5 + t_A	–	ns
t_6	CC	Address setup to ALE	1.5 + t_A	–	TCL – 11 + t_A	–	ns
t_7	CC	Address hold after ALE	4 + t_A	–	TCL – 8.5 + t_A	–	ns
t_8	tCC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	4 + t_A	–	TCL – 8.5 + t_A	–	ns
t_9	CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	– 8.5 + t_A	–	– 8.5 + t_A	–	ns
t_{10}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay)	–	6	–	6	ns
t_{11}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)	–	18.5	–	TCL + 6	ns
t_{12}	CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2TCL – 9.5 + t_C	–	ns
t_{13}	CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3TCL – 9.5 + t_C	–	ns
t_{14}	SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2TCL – 19 + t_C	ns
t_{15}	SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3TCL – 19 + t_C	ns
t_{16}	SR	ALE low to valid data in	–	17.5 + t_A + t_C	–	3TCL – 20 + t_A + t_C	ns
t_{17}	SR	Address/unlatched $\overline{CS}$ to valid data in	–	20 + $2t_A$ + t_C	–	4TCL – 30 + $2t_A$ + t_C	ns
t_{18}	SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{19}	SR	Data float after $\overline{RD}$	–	16.5 + t_F	–	2TCL – 8.5 + t_F	ns
t_{22}	CC	Data valid to $\overline{WR}$	10 + t_C	–	2TCL – 15 + t_C	–	ns
t_{23}	CC	Data hold after $\overline{WR}$	4 + t_F	–	2TCL – 8.5 + t_F	–	ns
t_{25}	CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	15 + t_F	–	2TCL – 10 + t_F	–	ns
t_{27}	CC	Address/unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	10 + t_F	–	2TCL – 15 + t_F	–	ns
t_{38}	CC	ALE falling edge to latched $\overline{CS}$	– 4 – t_A	10 – t_A	– 4 – t_A	10 – 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_{40}	CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	27 + t_F	–	3TCL - 10.5 + t_F	–	ns

Table 70. Multiplexed bus timings (continued)

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 40 MHz		Unit
			Min	Max	Min	Max	
t_{42}	CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	7 + t_A	–	TCL - 5.5 + t_A	–	ns
t_{43}	CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)	-5.5 + t_A	–	-5.5 + t_A	–	ns
t_{44}	CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	–	1.5	–	1.5	ns
t_{45}	CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)	–	14	–	TCL + 1.5	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_{52}	SR	Data float after $\overline{RdCS}$	–	16.5 + t_F	–	2TCL - 8.5 + t_F	ns
t_{54}	CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	6 + t_F	–	2TCL - 19 + t_F	–	ns
t_{56}	CC	Data hold after $\overline{WrCS}$	6 + t_F	–	2TCL - 19 + t_F	–	ns

Figure 50. External memory cycle: Multiplexed bus, with/without read/write delay, normal ALE

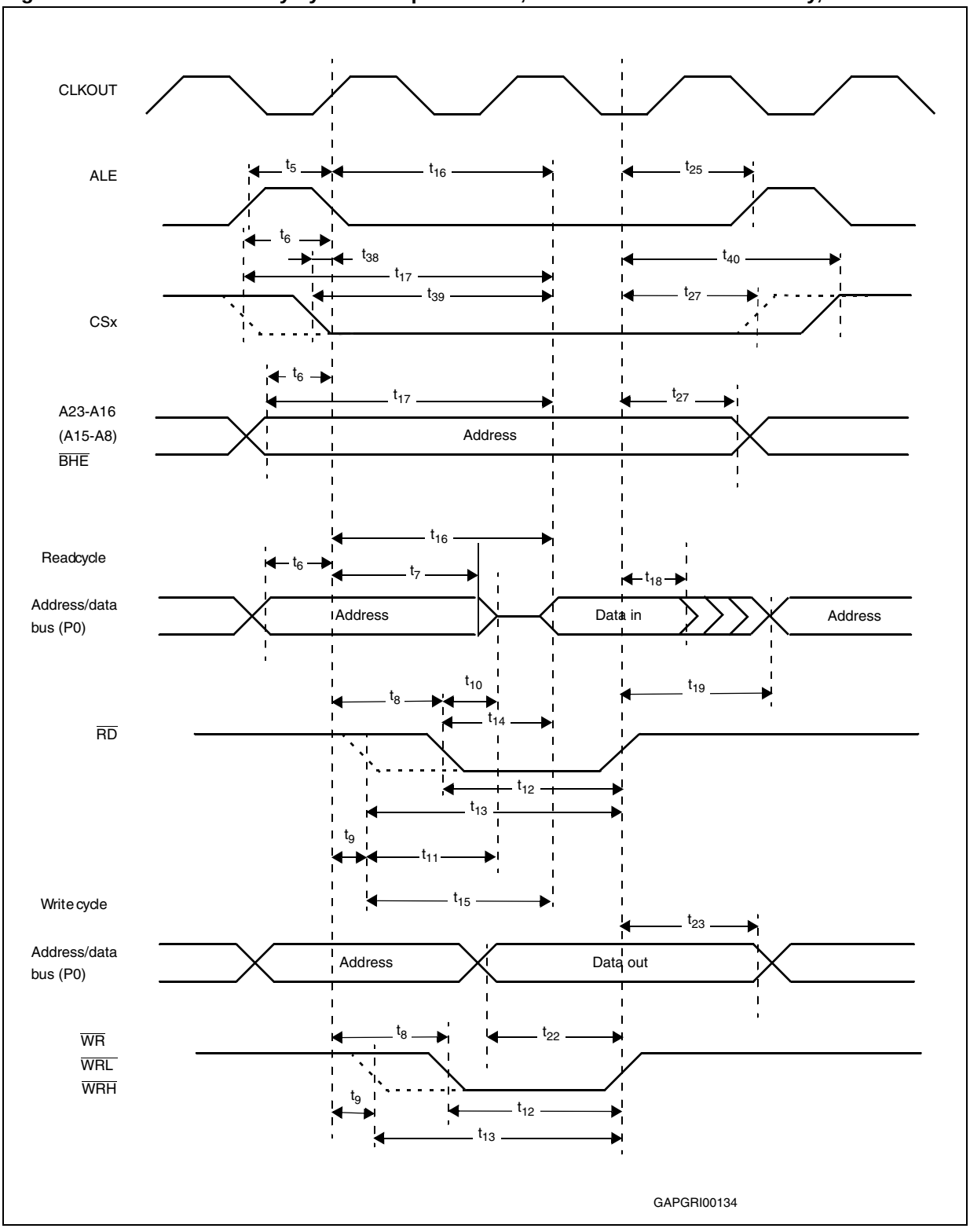


Figure 51. External memory cycle: Multiplexed bus, with/without read/write delay, extended ALE

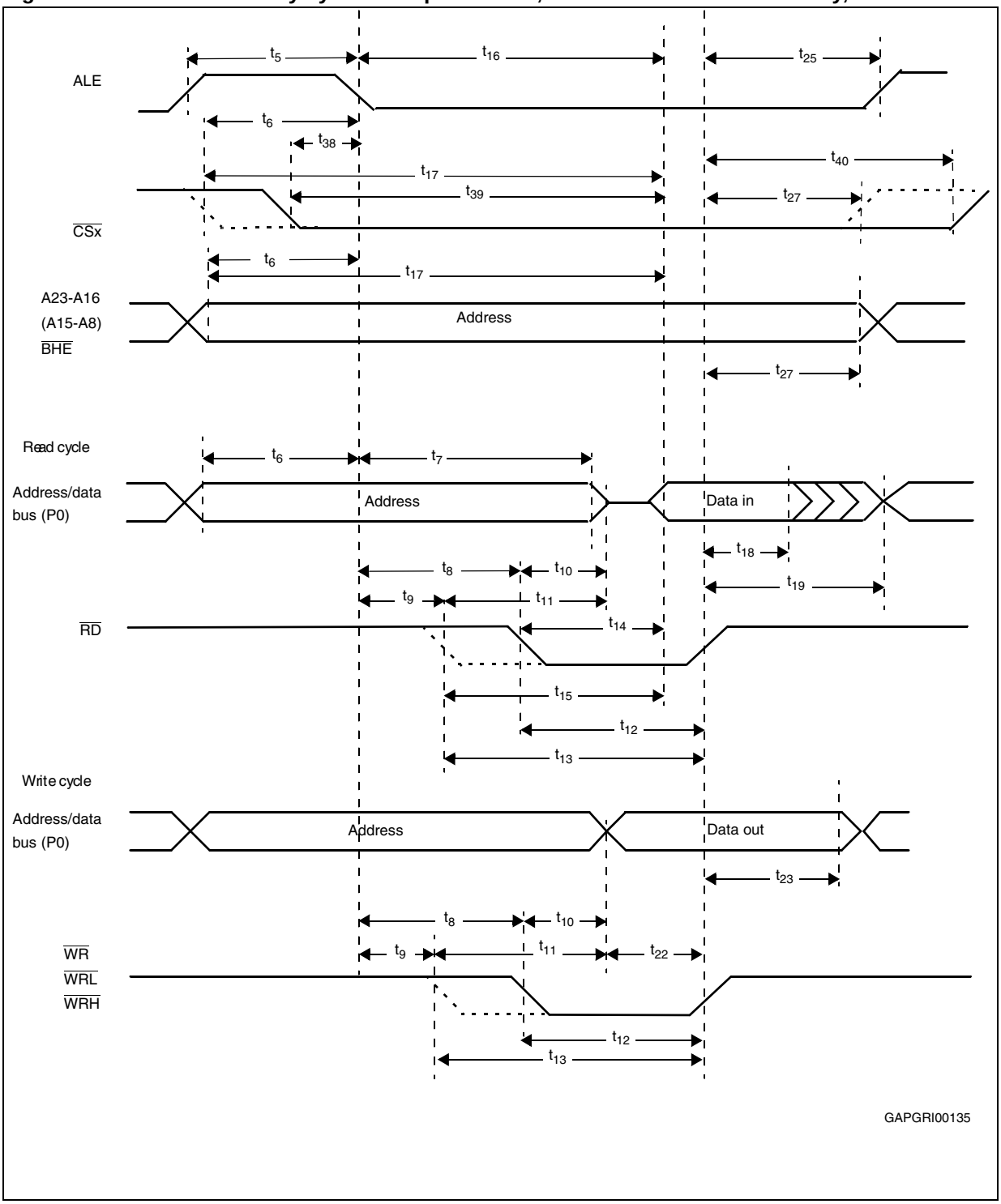


Figure 52. External memory cycle: Multiplexed bus, with/without r/w delay, normal ALE, r/w CS

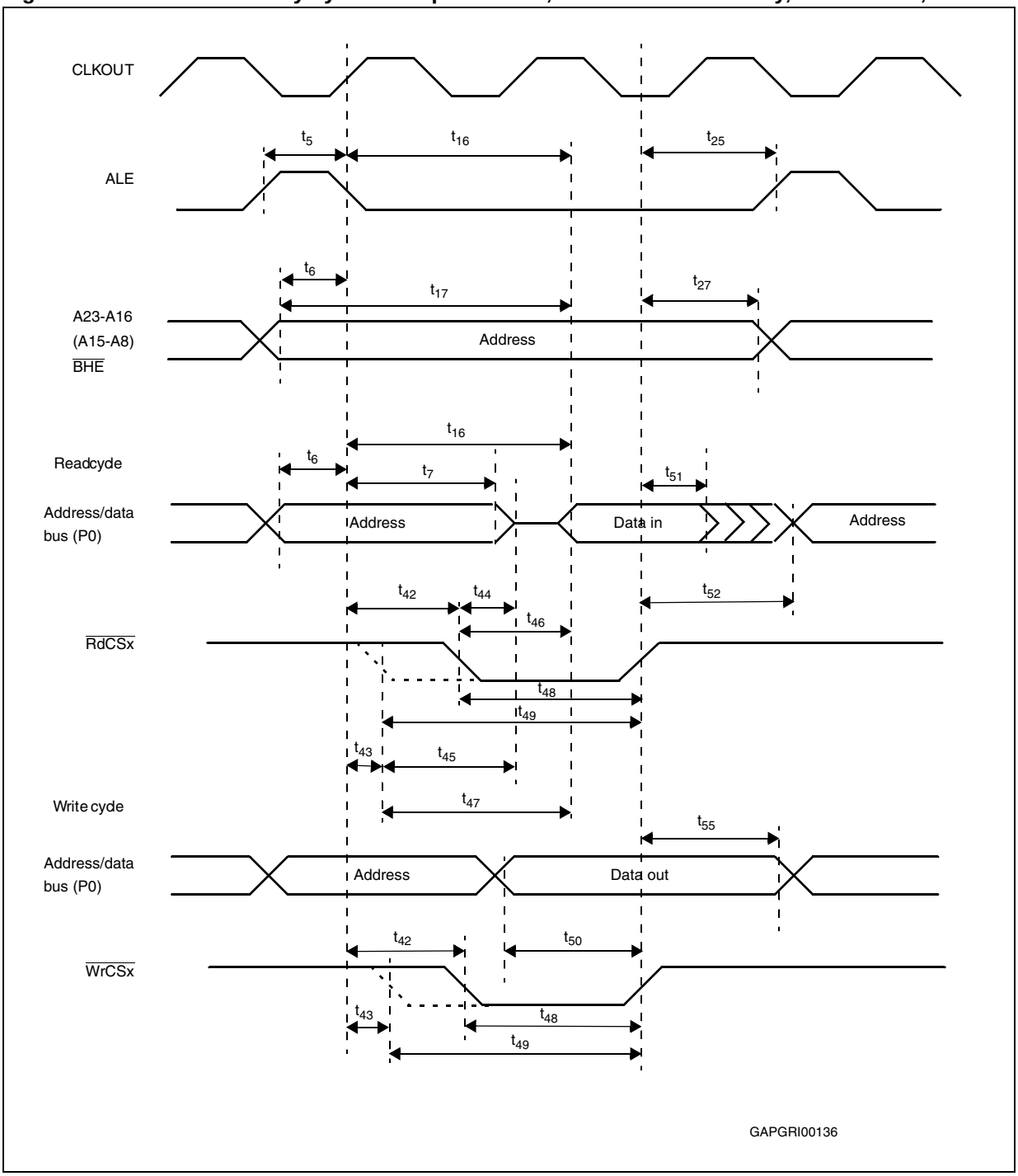
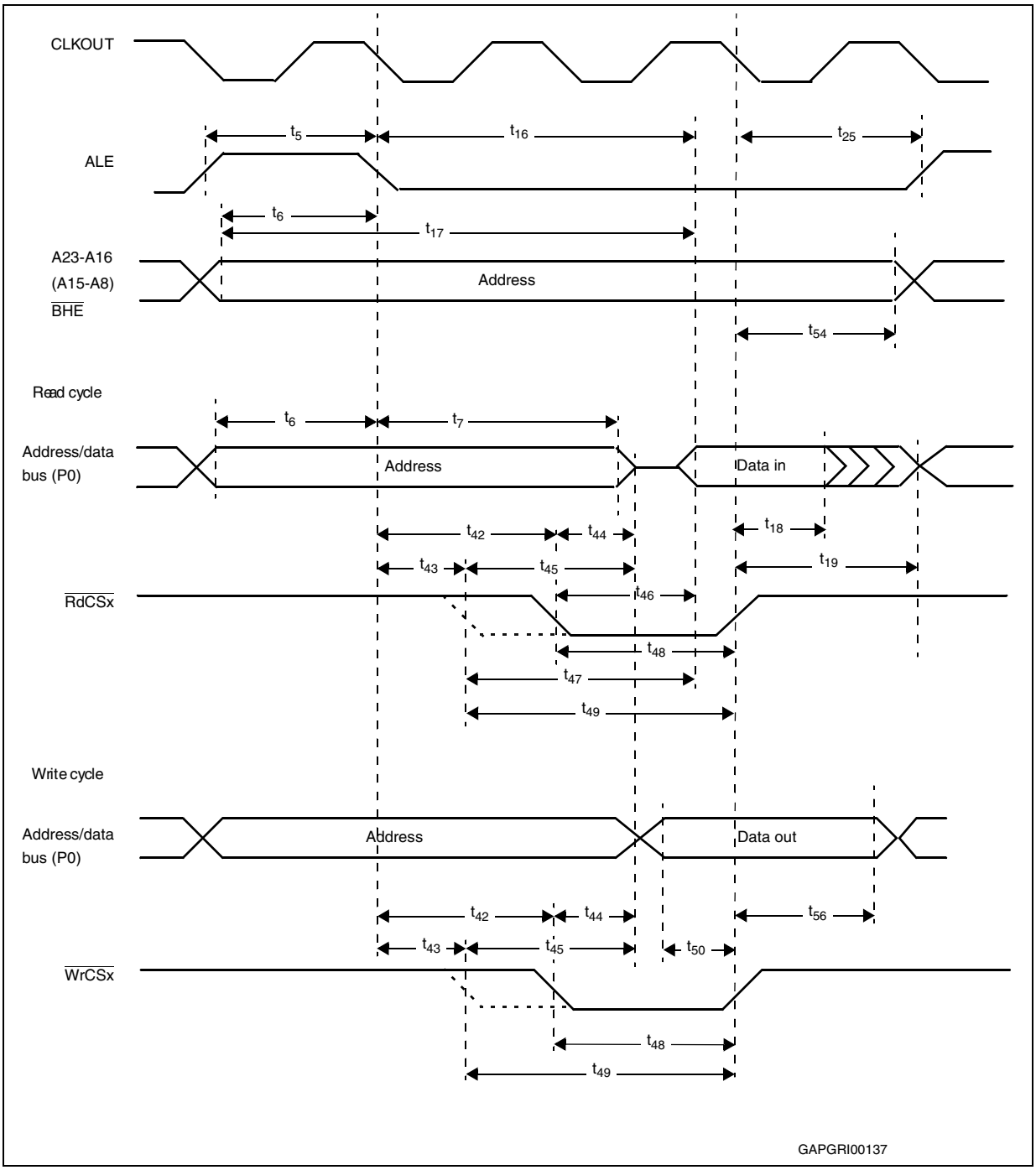


Figure 53. External memory cycle: Multiplexed bus, with/without r/w delay, extended ALE, r/w CS



24.8.17 Demultiplexed bus

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +125 °C, CL = 50 pF,
ALE cycle time = 4 TCL + 2 t_A + t_C + t_F (50 ns at 40 MHz CPU clock without wait states).

Table 71. Demultiplexed bus timings

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 40 MHz		Unit
			Min	Max	Min	Max	
t_5	CC	ALE high time	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_6	CC	Address setup to ALE	1.5 + t_A	–	TCL - 11 + t_A	–	ns
t_{80}	CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	12.5 + 2t_A	–	2TCL - 12.5 + 2t_A	–	ns
t_{81}	CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	0.5 + 2t_A	–	TCL - 12 + 2t_A	–	ns
t_{12}	CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2TCL - 9.5 + t_C	–	ns
t_{13}	CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3TCL - 9.5 + t_C	–	ns
t_{14}	SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2TCL - 19 + t_C	ns
t_{15}	SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3TCL - 19 + t_C	ns
t_{16}	SR	ALE low to valid data in	–	17.5 + t_A + t_C	–	3TCL - 20 + t_A + t_C	ns
t_{17}	SR	Address/Unlatched $\overline{CS}$ to valid data in	–	20 + 2t_A + t_C	–	4TCL - 30 + 2t_A + t_C	ns
t_{18}	SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{20}	SR	Data float after $\overline{RD}$ rising edge (with RW-delay)(1)	–	16.5 + t_F	–	2TCL - 8.5 + t_F + 2t_A	ns
t_{21}	SR	Data float after $\overline{RD}$ rising edge (no RW-delay)(1)	–	4 + t_F	–	TCL - 8.5 + t_F + 2t_A	ns
t_{22}	CC	Data valid to $\overline{WR}$	10 + t_C	–	2TCL - 15 + t_C	–	ns
t_{24}	CC	Data hold after $\overline{WR}$	4 + t_F	–	TCL - 8.5 + t_F	–	ns
t_{26}	CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	-10 + t_F	–	-10 + t_F	–	ns
t_{28}	CC	Address/unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$(2)	0 + t_F	–	0 + t_F	–	ns
t_{28h}	CC	Address/unlatched $\overline{CS}$ hold after $\overline{WRH}$	-5 + t_F	–	-5 + t_F	–	ns
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

Table 71. Demultiplexed bus timings (continued)

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 40 MHz		Unit
			Min	Max	Min	Max	
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 + 2 t_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)(3)	–	16.5 + t_F	–	2TCL - 8.5 + t_F	ns
t_{68}	SR	Data float after $\overline{RdCS}$ (no RW-delay)(3)	–	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 is 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 54. External memory cycle: Demultiplexed bus, with/without r/w delay, normal ALE

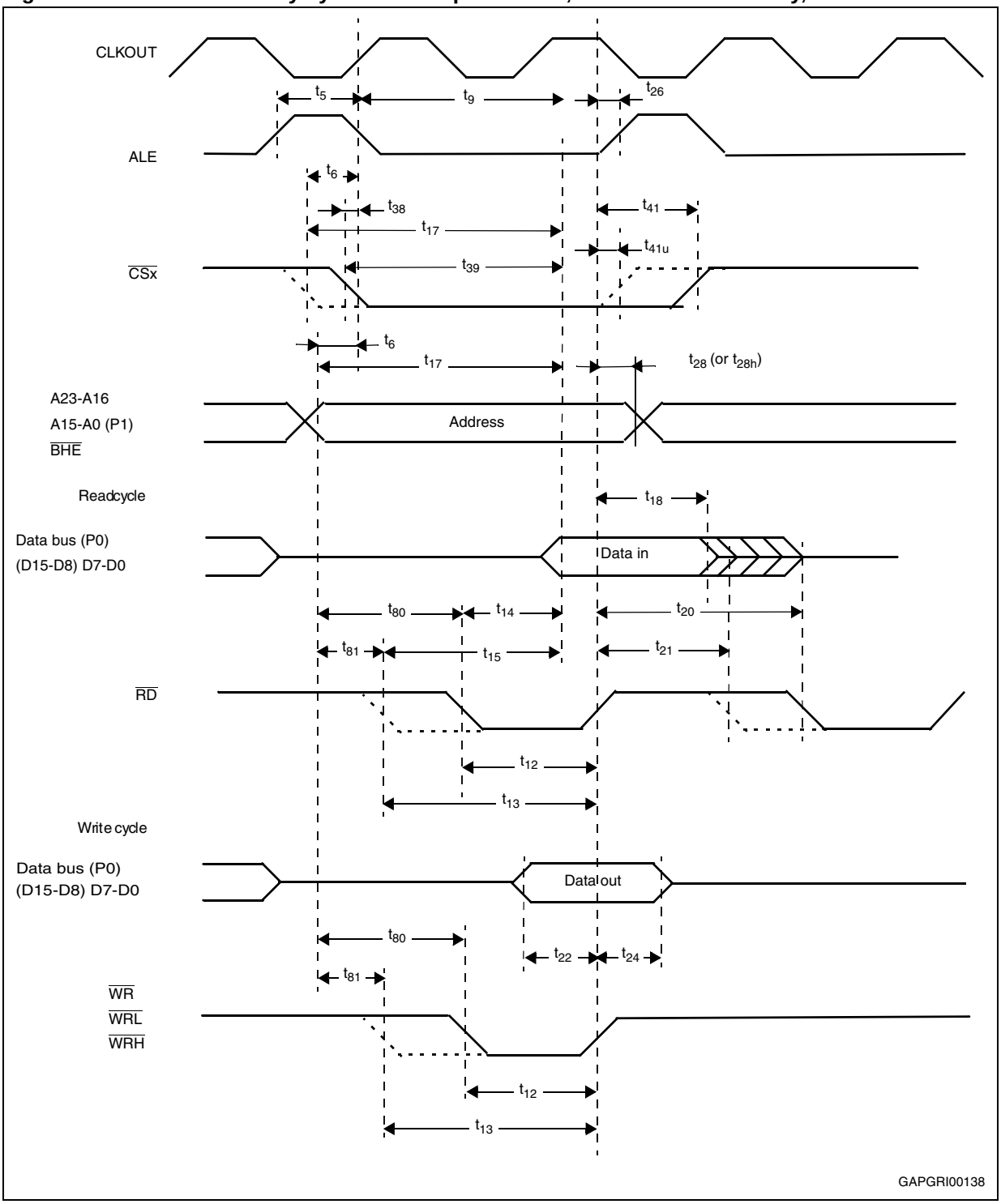


Figure 55. External memory cycle: Demultiplexed bus, with/without r/w delay, extended ALE

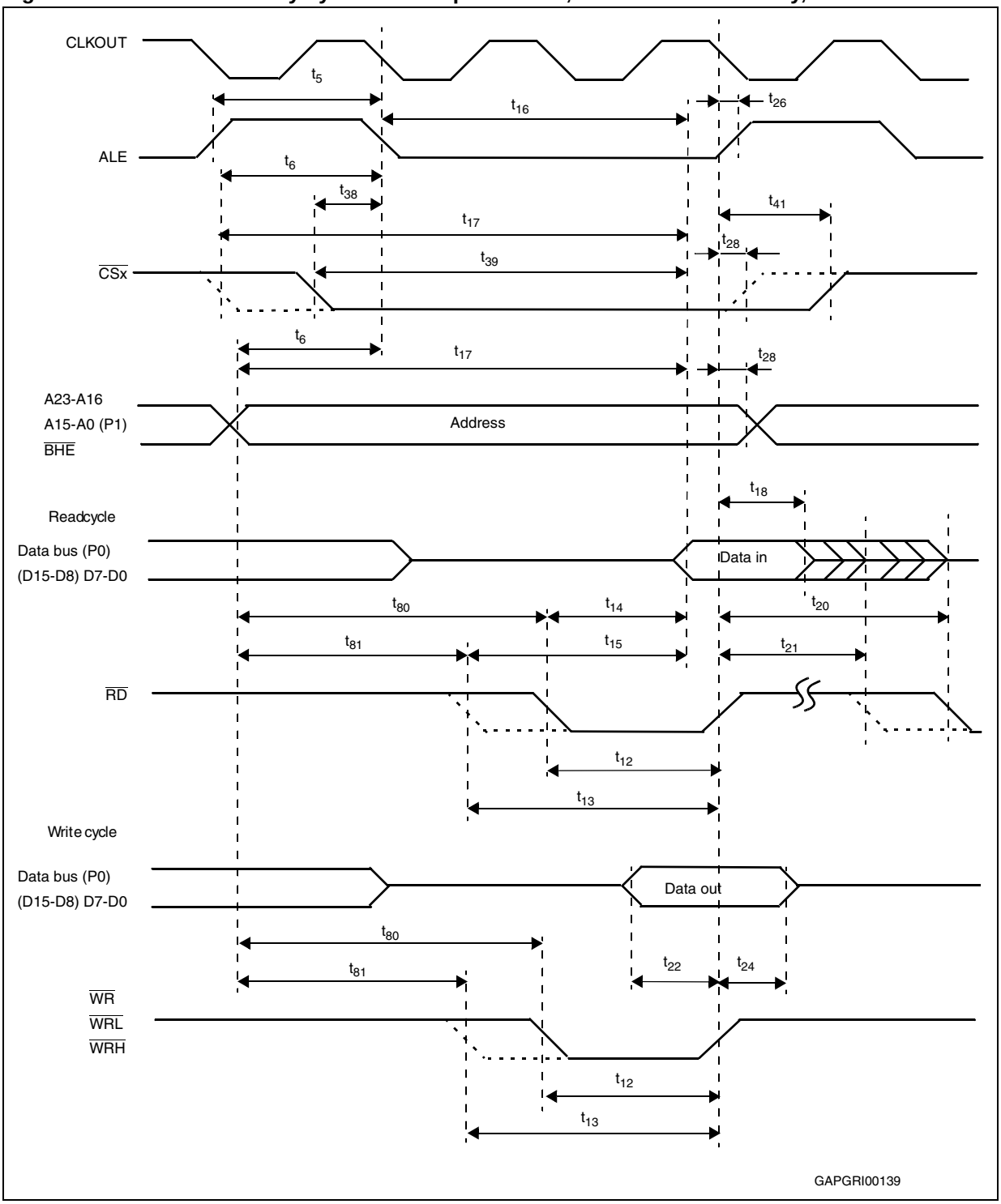


Figure 56. External memory cycle: Demultipl. bus, with/without r/w delay, normal ALE, r/w CS

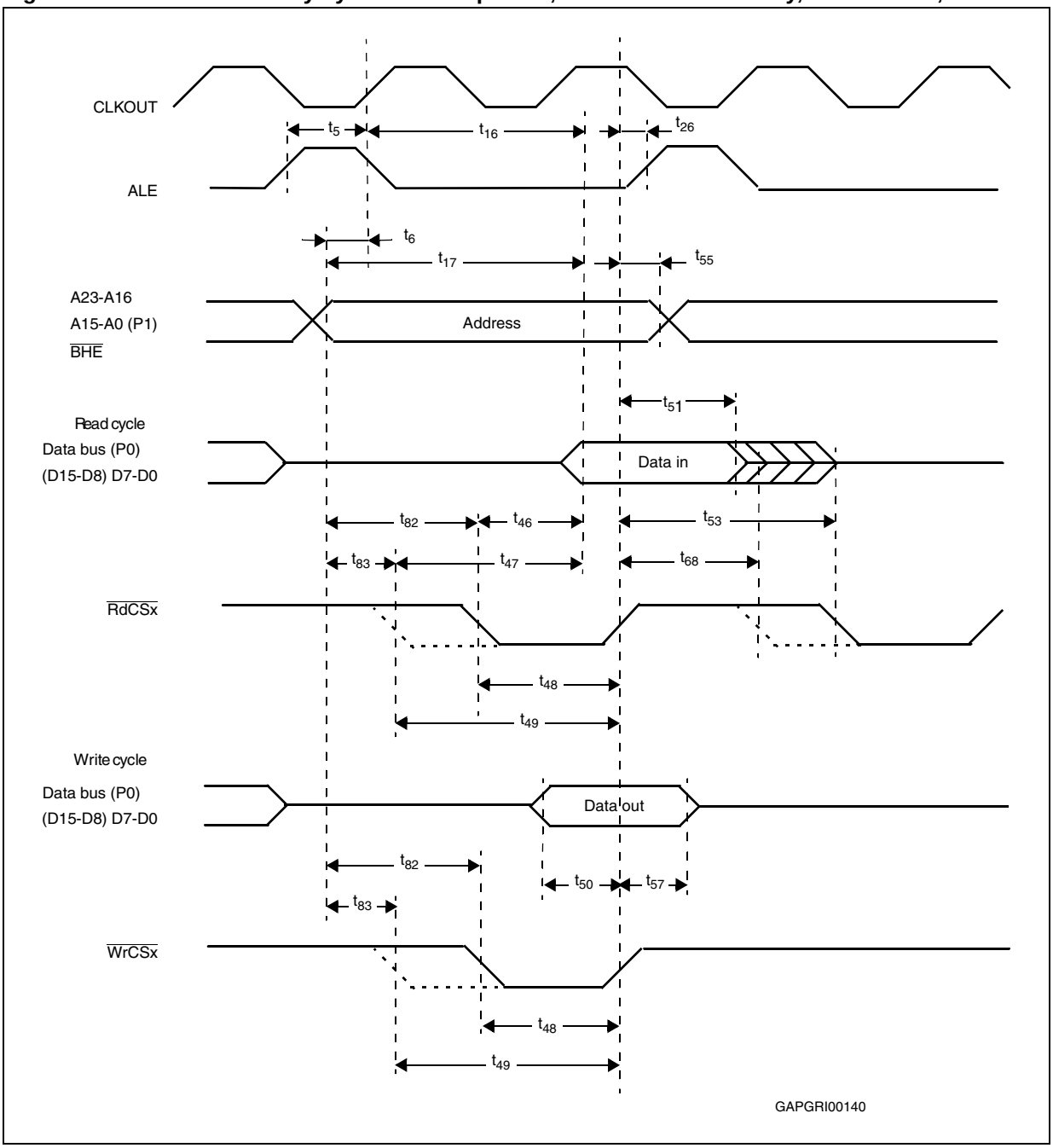
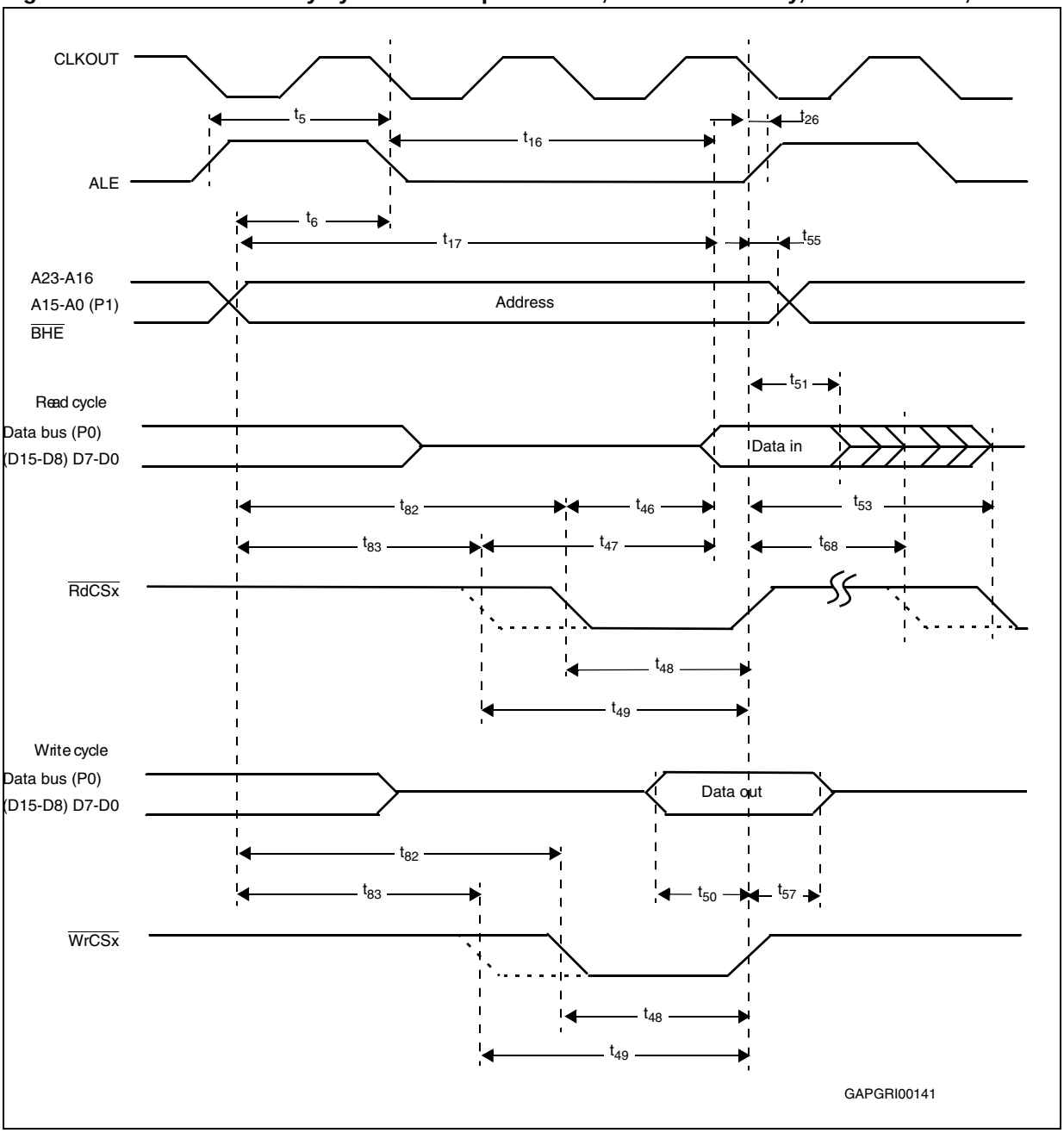


Figure 57. External memory cycle: Demultiplexed bus, without r/w delay, extended ALE, r/w CS



24.8.18 CLKOUT and $\overline{\text{READY}}$

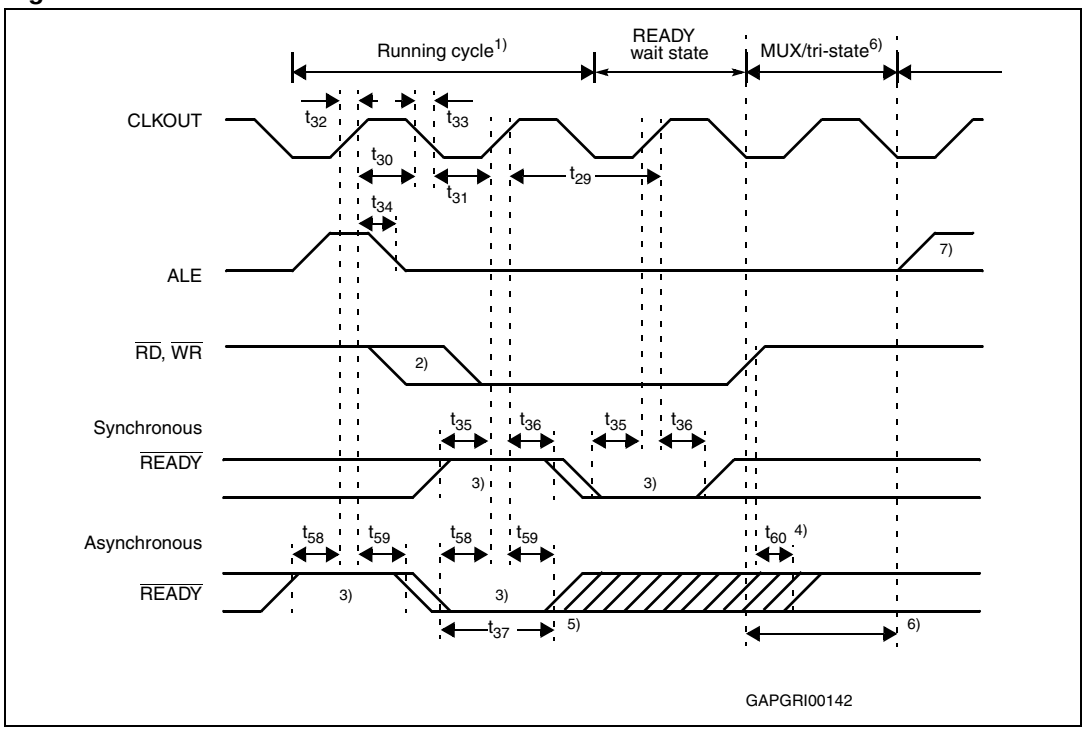
V_{DD} = 5 V ± 10 %, V_{SS} = 0 V, T_A = -40 to + 125 °C, CL = 50 pF

Table 72. CLKOUT and $\overline{\text{READY}}$ 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_{29}	CC	CLKOUT cycle time	25	25	2TCL	2TCL	ns
t_{30}	CC	CLKOUT high time	9	–	TCL - 3.5	–	ns
t_{31}	CC	CLKOUT low time	10	–	TCL - 2.5	–	ns
t_{32}	CC	CLKOUT rise time	–	4	–	4	ns
t_{33}	CC	CLKOUT fall time	–	4	–	4	ns
t_{34}	CC	CLKOUT rising edge to ALE falling edge	-2 + t_A	8 + t_A	-2 + t_A	8 + t_A	ns
t_{35}	SR	Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	17	–	17	–	ns
t_{36}	SR	Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	2	–	2	–	ns
t_{37}	SR	Asynchronous $\overline{\text{READY}}$ low time	35	–	2TCL + 10	–	ns
t_{58}	SR	Asynchronous $\overline{\text{READY}}$ setup time(1)	17	–	17	–	ns
t_{59}	SR	Asynchronous $\overline{\text{READY}}$ hold time(1)	2	–	2	–	ns
t_{60}	SR	Asynchronous $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (demultiplexed Bus)(2)	0	$2t_A + t_C + t_F$	0	$2t_A + t_C + t_F$	ns

1. These timings are given for characterization purposes only, in order to assure recognition at a specific clock edge.
2. Demultiplexed bus is the worst case. For multiplexed bus 2 TCL are to be added to the maximum values. This adds even more time for deactivating READY. $2t_A$ and t_C refer to the next following bus cycle, t_F refers to the current bus cycle.

Figure 58. CLKOUT and $\overline{\text{READY}}$



1. Cycle as programmed, including MCTC wait states (example shows 0 MCTC WS).
2. The leading edge of the respective command depends on RW-delay.
3. $\overline{\text{READY}}$ sampled HIGH at this sampling point generates a READY controlled wait state, $\overline{\text{READY}}$ sampled LOW at this sampling point terminates the currently running bus cycle.
4. $\overline{\text{READY}}$ may be deactivated in response to the trailing (rising) edge of the corresponding command ($\overline{\text{RD}}$ or $\overline{\text{WR}}$).
5. If the Asynchronous $\overline{\text{READY}}$ signal does not fulfill the indicated setup and hold times with respect to CLKOUT (for example, because CLKOUT is not enabled), it must fulfill t_{37} in order to be safely synchronized. This is guaranteed, if READY is removed in response to the command (see Note 4).
6. Multiplexed bus modes have a MUX wait state added after a bus cycle, and an additional MTTC wait state may be inserted here. For a multiplexed bus with MTTC wait state this delay is two CLKOUT cycles, for a demultiplexed bus without MTTC wait state this delay is zero.
7. The next external bus cycle may start here.

24.8.19 External bus arbitration

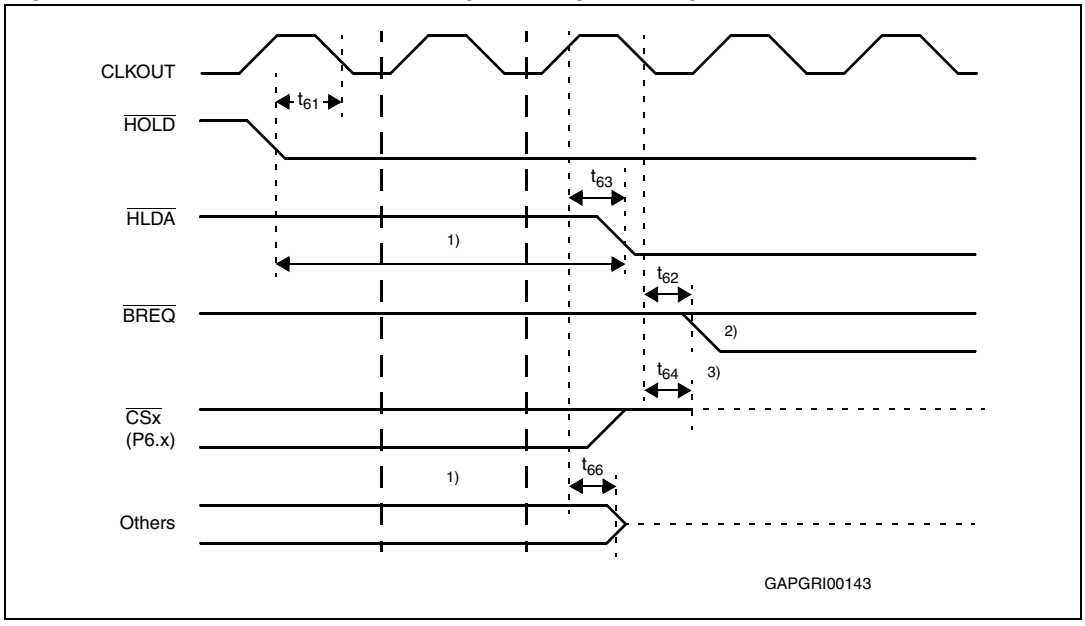
V_{DD} = 5 V ± 10 %, V_{SS} = 0 V, T_A = -40 to +125 °C, C_L = 50 pF

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

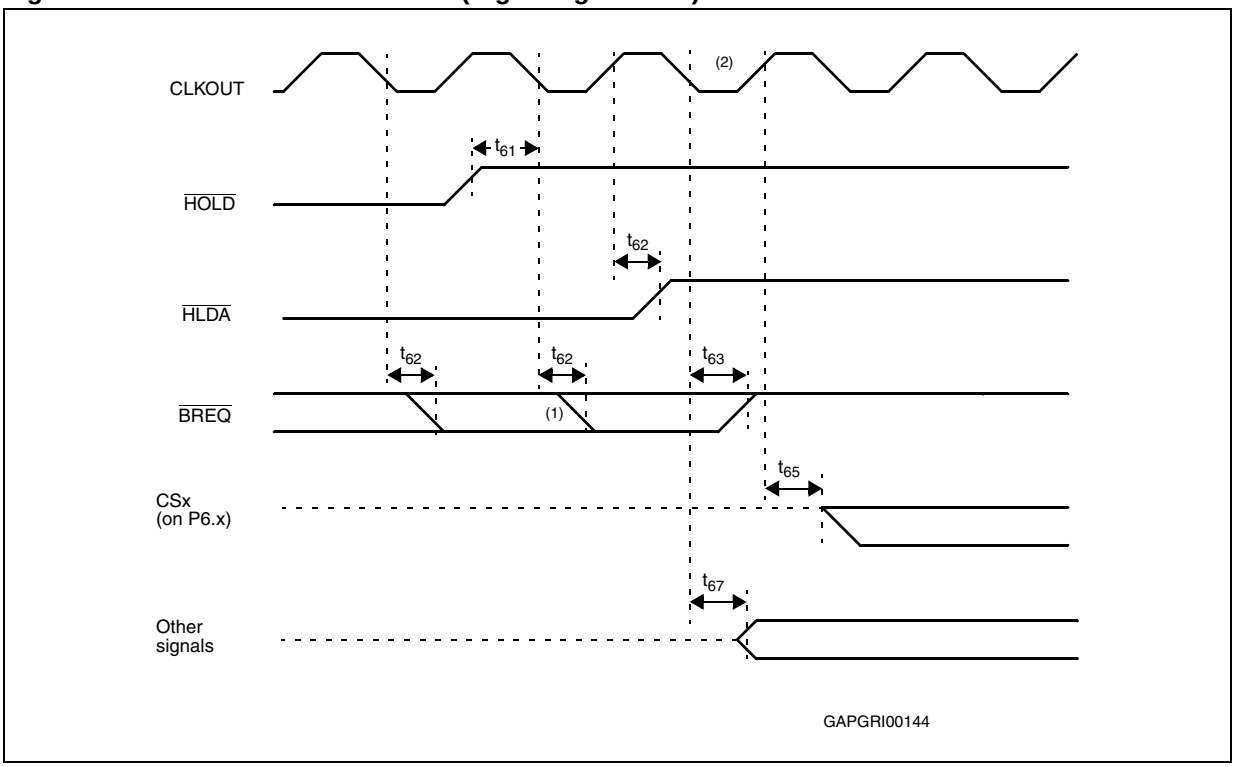
1. Partially tested, guaranteed by design characterization

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}.

Figure 60. External bus arbitration (regaining the bus)



1. This is the last chance for $\overline{BREQ}$ to trigger the indicated regain-sequence. Even if $\overline{BREQ}$ is activated earlier, the regain-sequence is initiated by $\overline{HOLD}$ going high. Please note that $\overline{HOLD}$ may also be deactivated without the ST10F272M requesting the bus.
2. The next ST10F272M driven bus cycle may start here.

24.8.20 High-speed synchronous serial interface (SSC) timing

24.8.20.1 Master mode

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

Table 74. SSC master mode timings

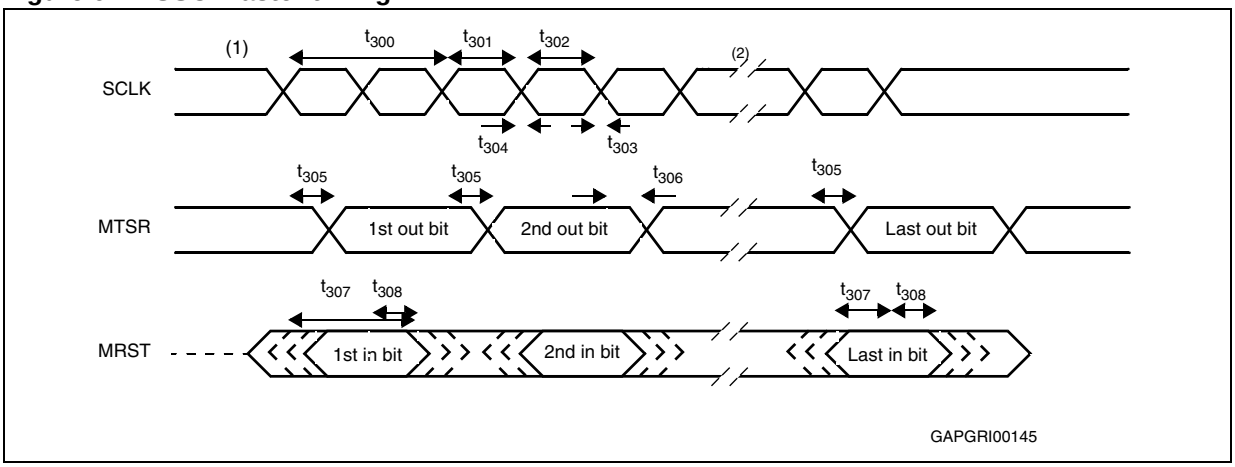
Symbol		Parameter	Maximum baudrate 6.6 Mbaud(1) @ f_{CPU} = 40 MHz (<SSCBR> = 0002h)		Variable baudrate (<SSCBR> = 0001h - FFFFh)		Unit
			Min	Max	Min	Max	
t_{300}	CC	SSC clock cycle time(2)	150	150	8TCL	262144 TCL	ns
t_{301}	CC	SSC clock high time	63	–	t_{300}/2 - 12	–	ns
t_{302}	CC	SSC clock low time	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

Table 74. SSC master mode timings (continued)

Symbol		Parameter	Maximum baudrate 6.6 Mbaud(1) @ f_{CPU} = 40 MHz (<SSCBR> = 0002h)		Variable baudrate (<SSCBR> = 0001h - FFFFh)		Unit
			Min	Max	Min	Max	
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. When 40 MHz 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) 32 MHz.
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 125 ns (corresponding to 8 Mbaud).
3. Partially tested, guaranteed by design characterization.

Figure 61. SSC master timing



1. The phase and polarity of shift and latch edge of SCLK is programmable. This figure uses the leading clock edge as shift edge (drawn in bold), with latch on trailing edge (SSCPH = 0b), Idle clock line is low, leading clock edge is low-to-high transition (SSCPO = 0b).
2. The bit timing is repeated for all bits to be transmitted or received.

24.8.20.2 Slave mode

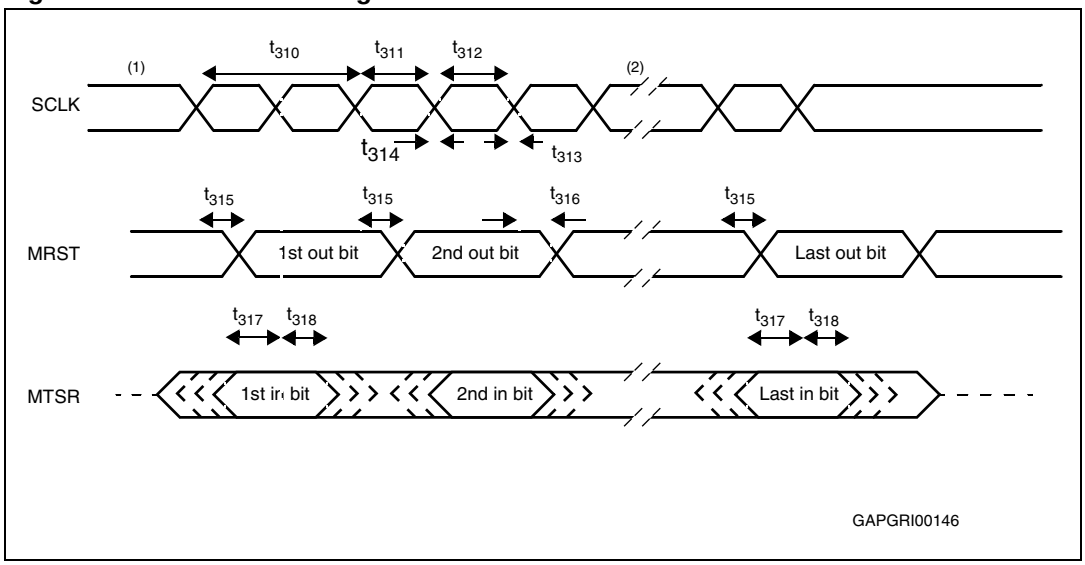
V_{DD} = 5 V ±10 %, V_{SS} = 0 V, T_A = -40 to +125 °C, C_L = 50 pF

Table 75. SSC slave mode timings

Symbol		Parameter	Maximum baudrate 6.6 Mbaud(1) @ f_{CPU} = 40 MHz (<SSCBR> = 0002h)		Variable baudrate (<SSCBR> = 0001h - FFFFh)		Unit
			Min	Max	Min	Max	
t_{310}	SR	SSC clock cycle time(2)	150	150	8TCL	262144 TCL	ns
t_{311}	SR	SSC clock high time	63	–	t_{310}/2 - 12	–	ns
t_{312}	SR	SSC clock low time	63	–	t_{310}/2 - 12	–	ns
t_{313}	SR	SSC clock rise time	–	10	–	10	ns
t_{314}	SR	SSC clock fall time	–	10	–	10	ns
t_{315}	CC	Write data valid after shift edge	–	55	–	2TCL + 30	ns
t_{316}	CC	Write data hold after shift edge	0	–	0	–	ns
t_{317p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	62	–	4TCL + 12	–	ns
t_{318p}	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	87	–	6TCL + 12	–	ns
t_{317}	SR	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	6	–	6	–	ns
t_{318}	SR	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	31	–	2TCL + 6	–	ns

1. When 40 MHz 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> may be used only with CPU clock lower than 32 MHz (after checking that resulting timings are suitable for the master).
2. Formula for SSC clock cycle time: t_{310} = 4 TCL * (<SSCBR> + 1)
Where <SSCBR> represents the content of the SSC baudrate register, taken as unsigned 16-bit integer.
Minimum limit allowed for t_{310} is 125ns (corresponding to 8 Mbaud).

Figure 62. SSC slave timing



1. The phase and polarity of shift and latch edge of SCLK is programmable. This figure uses the leading clock edge as shift edge (drawn in bold), with latch on trailing edge (SSCPH = 0b), Idle clock line is low, leading clock edge is low-to-high transition (SSCPO = 0b).
2. The bit timing is repeated for all bits to be transmitted or received.

25 Package information

25.1 ECOPACK® packages

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com.

ECOPACK® is an ST trademark.

25.2 LQFP144 mechanical data

Figure 63. LQFP144 package dimension

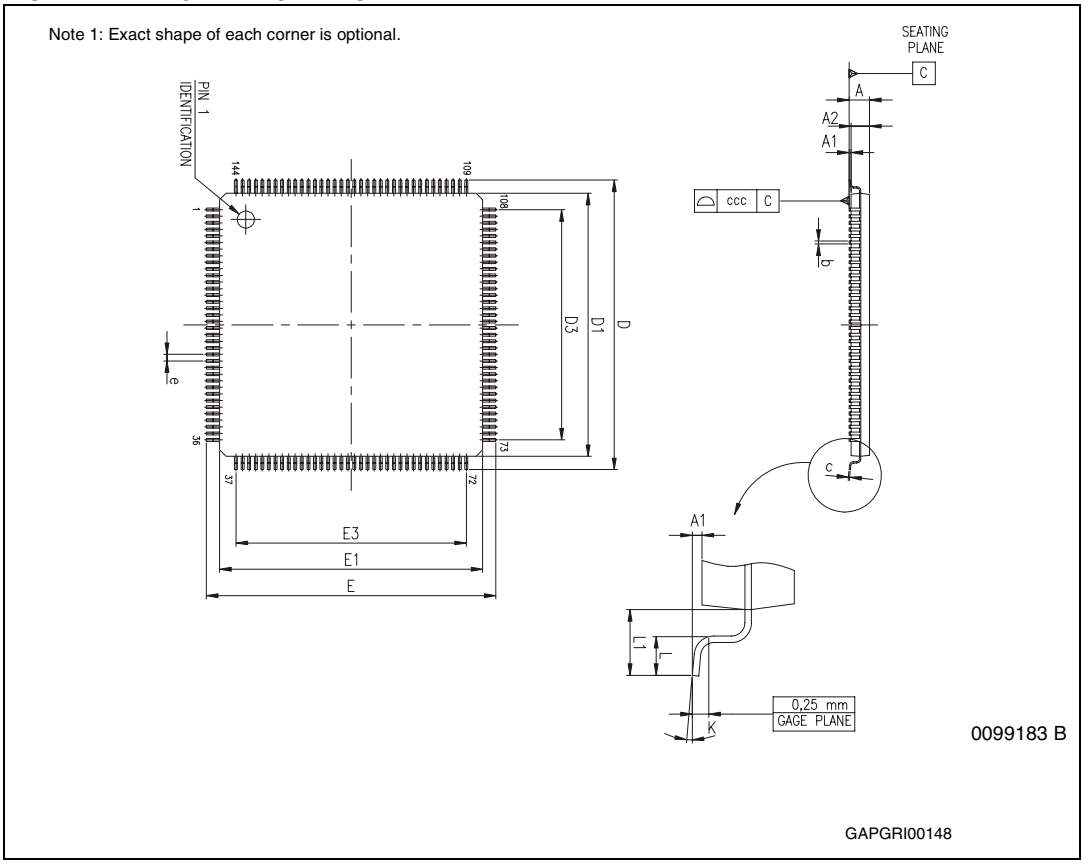


Table 76. LQFP144 mechanical data

Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.600			0.0630
A1	0.050		0.150	0.0019		0.0059
A2	1.350	1.400	1.450	0.0531	0.0551	0.0571

Table 76. LQFP144 mechanical data (continued)

Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
B	0.170	0.220	0.270	0.0067	0.0087	0.0106
C	0.090		0.200	0.0035		0.0089
D		22.000			0.8661	
D1		20.000			0.7874	
D3		17.500			0.6890	
e		0.500			0.0197	
E		22.000			0.8661	
E1		20.000			0.7874	
E3		17.500			0.6890	
L	0.450	0.600	0.750	0.0177	0.0236	0.0295
L1		1.000			0.0394	
K	3.5° min, 7° max					

26 Ordering information

Table 77. Device summary

Order code	Package	Packing	Temperature range (°C)	CPU frequency range (MHz)
ST10F272MR-4T3	LQFP144	Tray	-40 to +125	1 to 40
ST10F272MR-4TX3		Tape and reel		

27 Revision history

Table 78. Document revision history

Date	Revision	Changes
09-May-2007	1	Initial release
04-Jan-2008	2	Changed document status from Preliminary Data to Datasheet*Section 4: Memory organization on page 21*: Changed size of B0TF from 8 to 4Kbytes *Table 2: Summary of IFlash address range on page 21*: Changed size of B0TF from 8 to 4Kbytes *Figure 5: Flash structure on page 25*: Changed Test-Flash size from 8 to 4Kbytes *Table 5: Flash modules sectorization (write operations or with ROMS1 = '1' or bootstrap mode) on page 26*: Changed B0TF address and size (8 to 4Kbytes) *Section 14: A/D converter on page 66*: Replaced '40.630 CPU clock cycles' with '40630 CPU clock cycles' in end of section *Table 52: Absolute maximum ratings on page 123*: Changed V_{AREF} value from "-0.3 to V_{DD}" to "-0.5 to V_{DD} + 0.5" *Table 53: Recommended operating conditions on page 124*: Added missing V_{AREF} values. *Table 56: DC characteristics on page 126*: Changed max value for I_{PD1} from 200µA to 150µA *Table 61: On-chip clock generator selections on page 142*: - changed external clock input range for f_{XTAL} x 8 - changed external clock input range for f_{XTAL} x 1 - changed external clock input range for f_{XTAL} x 10 - changed configuration from f_{XTAL} x 16 to Reserved *Table 62: Internal PLL divider mechanism on page 145*: - changed external clock input range for f_{XTAL} x 3 - changed external clock input range for f_{XTAL} x 8 Table 76: PQFP144 mechanical data on page 173 and *Table 76: LQFP144 mechanical data on page 172*: Package mechanical data for inches converted to 4 decimal places
15-May-2012	3	Removed PQFP144 package . Update *Table 77: Device summary*.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com