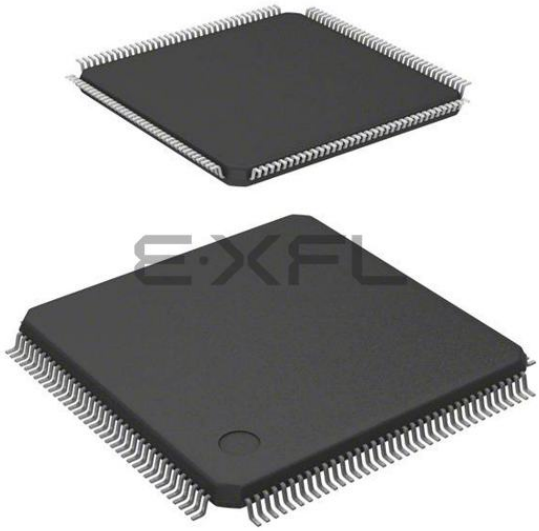


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST10
Core Size	16-Bit
Speed	48MHz
Connectivity	ASC, CANbus, EBI/EMI, I²C, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	832KB (832K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	68K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/f276-ceg-t-tr

Table 5. Control register interface

Bank	Description	Addresses	Size	ST10 bus size
FCR1-0	Flash control registers 1-0	0x000E 0000 - 0x000E 0007	8 byte	16-bit (X-BUS)
FDR1-0	Flash data registers 1-0	0x000E 0008 - 0x000E 000F	8 byte	
FAR	Flash address registers	0x000E 0010 - 0x000E 0013	4 byte	
FER	Flash error register	0x000E 0014 - 0x000E 0015	2 byte	
FNVWPXR	Flash non-volatile protection X register	0x000E DFB0 - 0x000E DFB3	4 byte	
FNVWPIR	Flash non-volatile protection I register	0x000E DFB4 - 0x000E DFB7	4 byte	
FNVAPR0	Flash non-volatile access protection register 0	0x000E DFB8 - 0x000E DFB9	2 byte	
FNVAPR1	Flash non-volatile access protection register 1	0x000E DFBC - 0x000E DFBF	4 byte	
XFICR	XFlash interface control register	0x000E E000 - 0x000E E001	2 byte	

4.2.3 Low power mode

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

Note: 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.

Power-off Flash mode is entered only at the end of the eventually running Flash write operation.

4.2.4 Write operation

The Flash modules have one single register interface mapped in the memory space of the XFlash module (0x0E 0000 to 0x0E 0013). 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 mapped on ST10 XBUS).

Note: Before accessing the XFlash module (and consequently also the Flash register to be used for program/erasing operations), bit XFLASHEN in XPERCON register and bit XPEN in SYSCON register shall be set.

The four banks have their own dedicated sense amplifiers, so that any bank can be read while any other bank is written. However simultaneous write operations (“write” means either Program or Erase) on different banks are forbidden: when there is a write operation on going (Program or Erase) anywhere in the Flash, no other write operation can be performed.

During a Flash write operation any attempt to read the bank under modification will output invalid data (software trap 009Bh). This means that the Flash bank is not fetchable when a

4.3.10 Flash address register high

FARH (0x0E 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 16. Flash address register high

Bit	Function
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.

4.3.11 Flash error register

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

FER (0xE 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 17. Flash error register

Bit	Function
ERR	Write error This bit is automatically set when an error occurs during a Flash write operation or when a bad write operation setup is done. Once the error has been discovered and understood, ERR bit must be software reset.
ERER	Erase error This bit is automatically set when an erase error occurs during a Flash write operation. This error is due to a real failure of a Flash cell, that can no more be erased. This kind of error is fatal and the sector where it occurred must be discarded. This bit has to be software reset.
PGER	Program error This bit is automatically set when a program error occurs during a Flash write operation. This error is due to a real failure of a Flash cell, that can no more be programmed. The word where this error occurred must be discarded. This bit has to be software reset.
10ER	1 over 0 error This bit is automatically set when trying to program at 1 bits previously set at 0 (this does not happen when programming the protection bits). This error is not due to a failure of the Flash cell, but only flags that the desired data has not been written. This bit has to be software reset.

5 Bootstrap loader

ST10F276E implements innovative boot capabilities in order to

- support a user defined bootstrap (see *Alternate bootstrap loader*);
- support bootstrap via UART or bootstrap via CAN for the standard bootstrap.

5.1 Selection among user-code, standard or alternate bootstrap

The selection among user-code, standard bootstrap or alternate bootstrap is made by special combinations on Port0L[5...4] during the time the reset configuration is latched from Port0.

The alternate boot mode is triggered with a special combination set on Port0L[5...4]. Those signals, as other configuration signals, are latched on the rising edge of $\overline{\text{RSTIN}}$ pin.

The alternate boot function is divided in two functional parts (which are independent from each other):

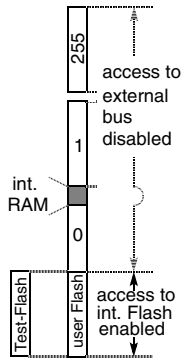
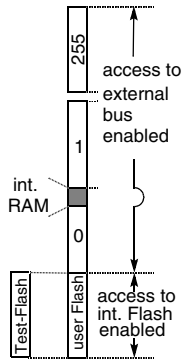
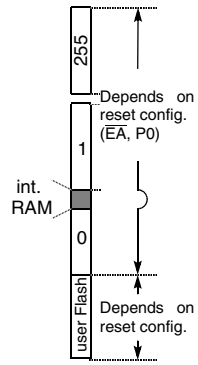
Part 1: Selection of reset sequence according to the Port0 configuration: User mode and alternate mode signatures

- Decoding of reset configuration (P0L.5 = 1, P0L.4 = 1) selects the normal mode and 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 and program).
- 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 but the alternate mode signature in the user Flash is programmed correctly, then the alternate boot mode is selected;
  - if both the user and the alternate mode signatures are not programmed correctly in the user Flash, then the user key location is read again. Its value will determine the behavior of the selected bootstrap loader.

Part 2: Running of user selected reset sequence

- Standard bootstrap loader: Jump to a predefined memory location in Test-Flash (controlled by ST)
- Alternate boot mode: Jump to address 09’0000h
- Selective bootstrap loader: Jump to a predefined location in Test-Flash (controlled by ST) and check which communication channel is selected
- User code: Make a software reset and jump to 00’0000h

Figure 8. Memory configuration after reset

	16 Mbytes	16 Mbytes	16 Mbytes
			
BSL mode active	**Yes** (P0L.4 = '0')	**Yes** (P0L.4 = '0')	**No** (P0L.4 = '1')
$\overline{EA}$ pin	High	Low	According to application
Code fetch from internal Flash area	Test-Flash access	Test-Flash access	User IFLASH access
Data fetch from internal Flash area	User IFLASH access	User IFLASH access	User IFLASH access

Note: As long as ST10F276E is in BSL, the user's software should not try to execute code from the internal IFlash, as the fetches are redirected to the Test-Flash.

5.2.6 Loading the start-up code

After the serial link initialization sequence (see following chapters), the BSL enters a loop to receive 32 bytes (boot via UART) or 128 bytes (boot via CAN).

These bytes are stored sequentially into ST10F276E Dual-Port RAM from location 00'FA40h.

To execute the loaded code, the BSL then jumps to location 00'FA40h. The bootstrap sequence running from the Test-Flash is now terminated; however, the microcontroller remains in BSL mode.

Most probably, the initially loaded routine, being the first level user code, will load additional code and data. This first level user code may use the pre-initialized interface (UART or CAN) to receive data and a second level of code, and store it in arbitrary user-defined locations.

This second level of code may be

- the final application code
- another, more sophisticated, loader routine that adds a transmission protocol to enhance the integrity of the loaded code or data
- a code sequence to change the system configuration and enable the bus interface to store the received data into external memory

In all cases, the ST10F276E still runs in BSL mode, that is, with the watchdog timer disabled and limited access to the internal IFLASH area.

5.2.7 Exiting bootstrap loader mode

To execute a program in normal mode, the BSL mode must first be terminated. The ST10F276E exits BSL mode at a software reset (level on P0L.4 is ignored) or a hardware reset (P0L.4 must be high in this case). After the reset, the ST10F276E starts executing from location $00'0000_H$ of the internal Flash (User Flash) or the external memory, as programmed via pin $\overline{EA}$.

Note: If a bidirectional software reset is executed and external memory boot is selected ($\overline{EA}$ = 0), a degeneration of the software reset event into a hardware reset can occur (refer to Section 19.3: Synchronous reset (warm reset) for details). This implies that P0L.4 becomes transparent, so to exit from Bootstrap mode it would be necessary to release pin P0L.4 (it is no longer ignored).

5.2.8 Hardware requirements

Although the new bootstrap loader is designed to be compatible with the old bootstrap loader, there are a few hardware requirements relative to the new bootstrap loader:

- External Bus configuration: Must have four or less segment address lines (keep CAN I/Os available);
- Usage of CAN pins (P4.5 and P4.6): Even in bootstrap via UART, P4.5 (CAN1_RxD) can be used as Port input but not as output. The pin P4.6 (CAN1_TxD) can be used as input or output.
- Level on UART RxD and CAN1_RxD during the bootstrap phase (see *Figure 6* - Step 2): Must be 1 (external pull-ups recommended).

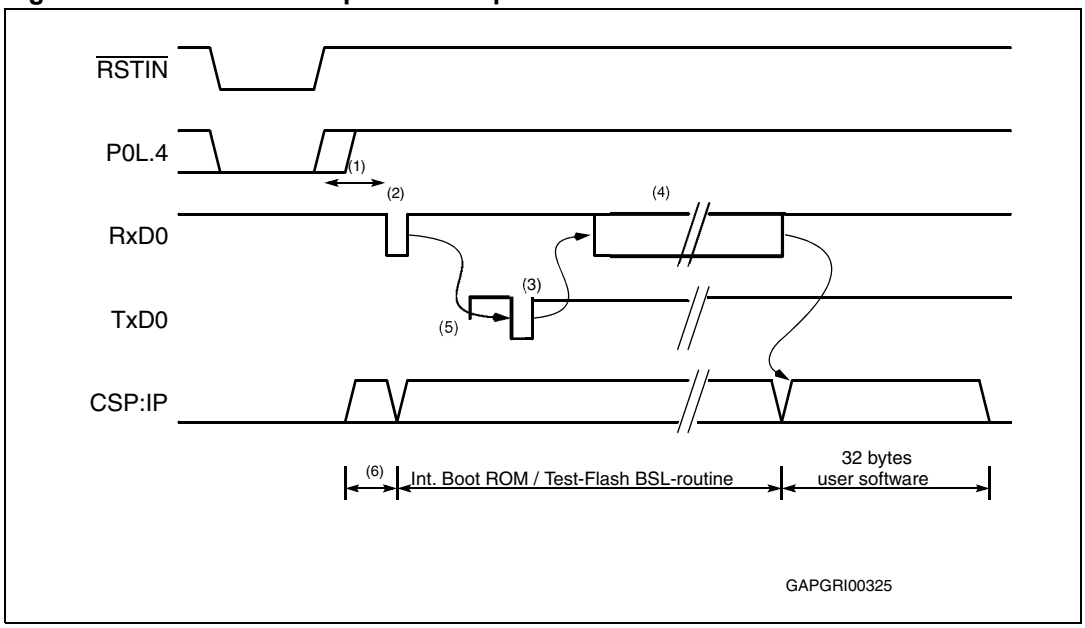
5.3 Standard bootstrap with UART (RS232 or K-Line)

5.3.1 Features

ST10F276E bootstrap via UART has the same overall behavior as the old ST10 bootstrap via UART:

- Same bootstrapping steps
- Same bootstrap method: Analyze the timing of a predefined byte, send back an acknowledge byte, load a fixed number of bytes and run
- Same functionalities: Boot with different crystals and PLL ratios

Figure 9. UART bootstrap loader sequence



1. BSL initialization time, > 1ms @ f_{CPU} = 40 MHz
2. Zero byte (1 start bit, eight '0' data bits, 1 stop bit), sent by host
3. Acknowledge byte, sent by ST10F276E
4. 32 bytes of code / data, sent by host
5. Caution: TxD0 is only driven a certain time after reception of the zero byte (1.3ms @ f_{CPU} = 40 MHz).
6. Internal Boot ROM / Test-Flash

5.3.2 Entering bootstrap via UART

The ST10F276E enters BSL mode if pin P0L.4 is sampled low at the end of a hardware reset. In this case, the built-in bootstrap loader is activated independently of the selected bus mode. The bootstrap loader code is stored in a special Test-Flash; no part of the standard mask ROM or Flash memory area is required for this.

After entering BSL mode and the respective initialization, the ST10F276E scans the RxD0 line to receive a zero byte, that is, 1 start bit, eight '0' data bits and 1 stop bit. From the duration of this zero byte, it calculates the corresponding baud rate factor with respect to the current CPU clock, initializes the serial interface ASC0 accordingly and switches pin TxD0 to output. Using this baud rate, an acknowledge byte is returned to the host that provides the loaded data.

The acknowledge byte is **D5h** for the ST10F276E.

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.

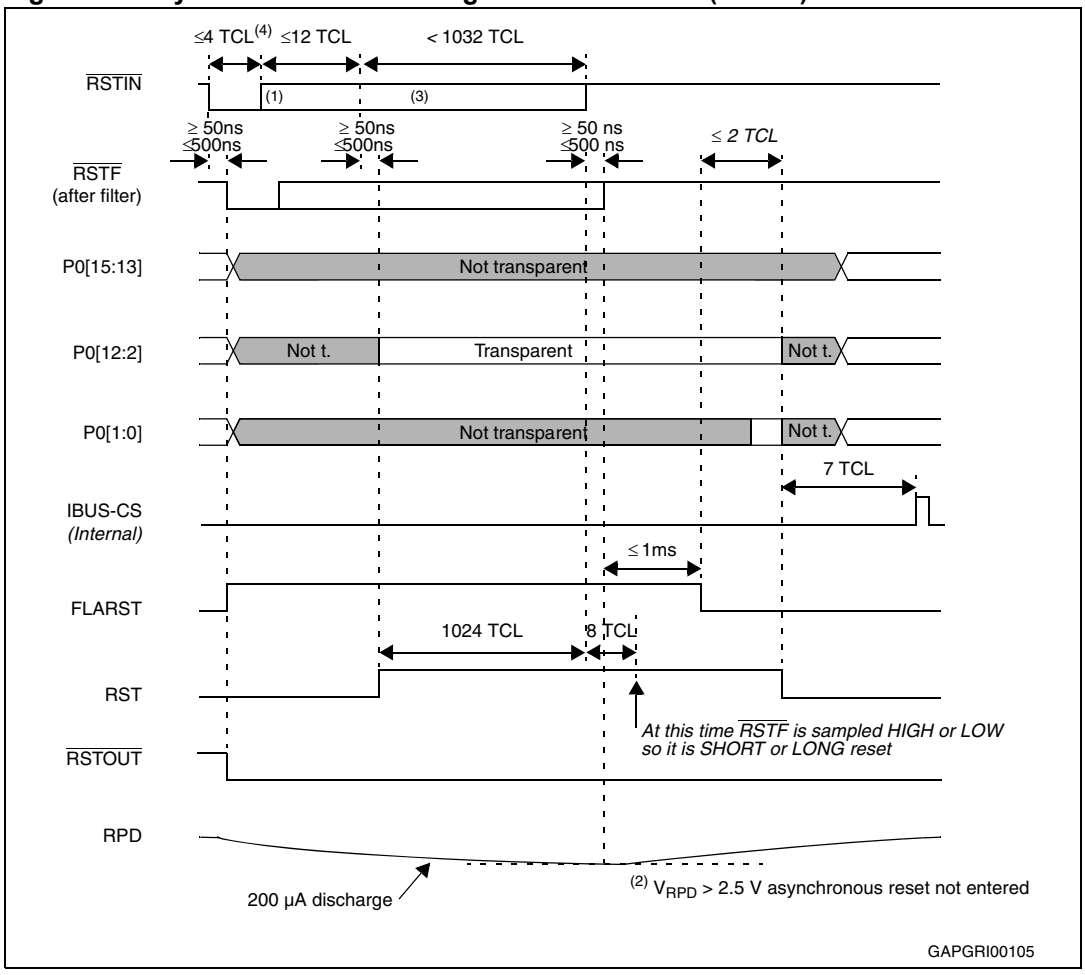
Synchronous reset and RPD pin

Whenever the $\overline{\text{RSTIN}}$ 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 (around 2.5V), the reset event becomes immediately asynchronous. In case of hardware reset (short or long) the situation goes immediately to the one illustrated in *Figure 26*. There is no effect if RPD comes again above the input threshold: the asynchronous reset is completed coherently. To grant the normal completion of a synchronous reset, the value of the capacitance shall be big enough to maintain the voltage on RPD pin sufficient high along the duration of the internal reset sequence.

For a Software or Watchdog reset event, 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 $\overline{\text{RSTF}}$ (after the noise filter).

Figure 28. Synchronous short / long hardware RESET ($\overline{\text{EA}}$ = 1)

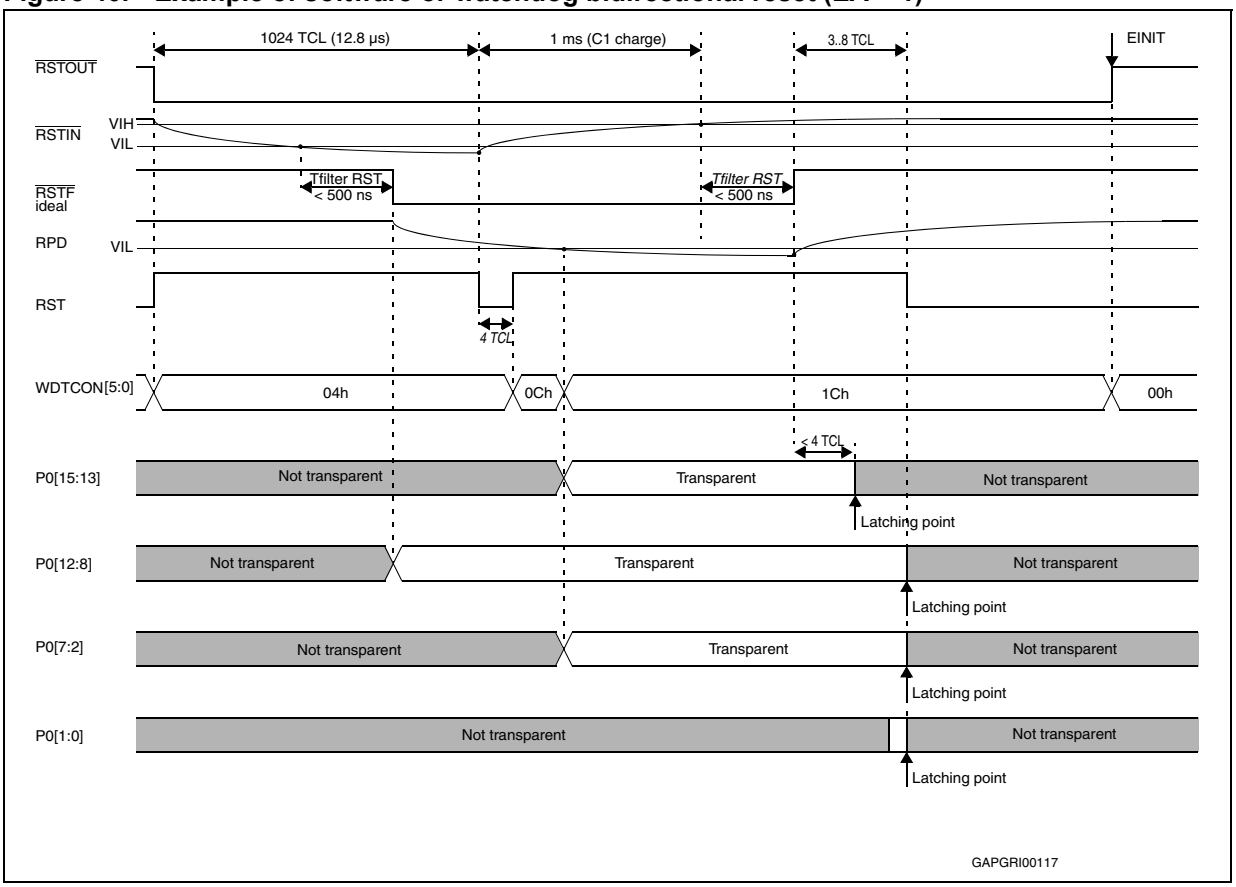


1. RSTIN assertion can be released there. Refer also to *Section 19.1* for details on minimum pulse duration.
2. If during the reset condition (RSTIN low), RPD voltage drops below the threshold voltage (about 2.5V for

19.8 Reset application examples

Next two timing diagrams (*Figure 40* and *Figure 41*) provides additional examples of bidirectional internal reset events (Software and Watchdog) including in particular the external capacitances charge and discharge transients (refer also to *Figure 38* for the external circuit scheme).

Figure 40. Example of software or watchdog bidirectional reset ($\overline{EA}$ = 1)



20 Power reduction modes

Three different power reduction modes with different levels of power reduction have been implemented in the ST10F276E. In Idle mode only CPU is stopped, while peripheral still operate. 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.*

20.1 Idle mode

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

Idle mode is terminate 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.

20.2 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.

Table 67. Special function registers ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
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
XADRS3	F01Ch**E**	0Eh	XPER address select register 3	800Bh
XP0IC **b**	F186h**E**	C3h	See *Section 8.1*	- - 00h
XP1IC **b**	F18Eh**E**	C7h	See *Section 8.1*	- - 00h
XP2IC **b**	F196h**E**	CBh	See *Section 8.1*	- - 00h
XP3IC **b**	F19Eh**E**	CFh	See *Section 8.1*	- - 00h
XPERCON	F024h**E**	12h	XPER configuration register	- - 05h
ZEROS b	FF1Ch	8Eh	Constant value 0's register (read-only)	0000h

Table 68. Special function registers ordered by address (continued)

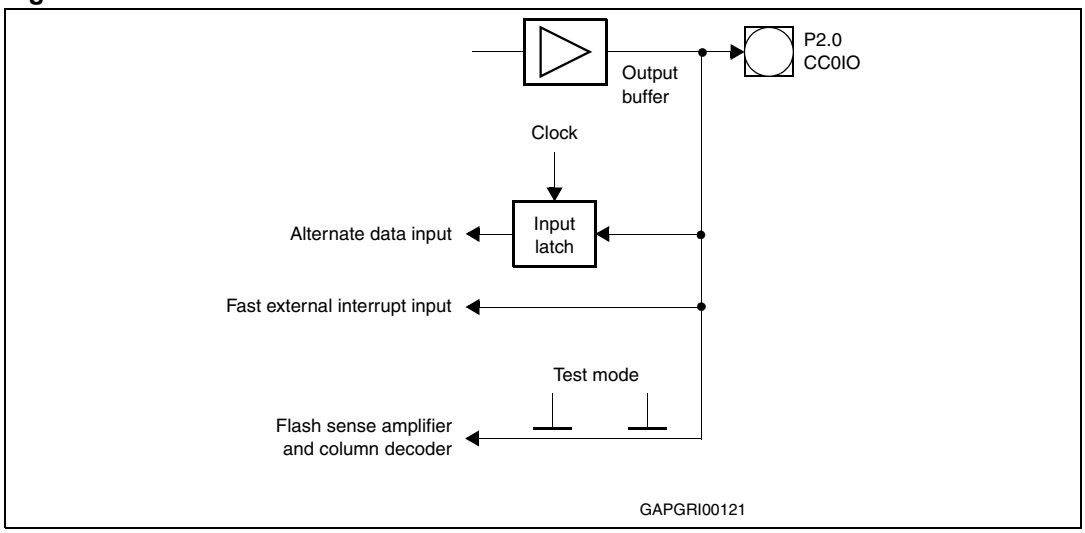
Name	Physical address	8-bit address	Description	Reset value
CC8	FE90h	48h	CAPCOM register 8	0000h
CC9	FE92h	49h	CAPCOM register 9	0000h
CC10	FE94h	4Ah	CAPCOM register 10	0000h
CC11	FE96h	4Bh	CAPCOM register 11	0000h
CC12	FE98h	4Ch	CAPCOM register 12	0000h
CC13	FE9Ah	4Dh	CAPCOM register 13	0000h
CC14	FE9Ch	4Eh	CAPCOM register 14	0000h
CC15	FE9Eh	4Fh	CAPCOM register 15	0000h
ADDAT	FEA0h	50h	A/D converter result register	0000h
WDT	FEAEh	57h	Watchdog timer register (read-only)	0000h
S0TBUF	FEB0h	58h	Serial channel 0 transmit buffer register (write-only)	0000h
S0RBUF	FEB2h	59h	Serial channel 0 receive buffer register (read-only)	- - XXh
S0BG	FEB4h	5Ah	Serial channel 0 baud rate generator reload 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
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
IDX0 b	FF08h	84h	MAC unit address pointer 0	0000h
IDX1 b	FF0Ah	85h	MAC unit address pointer 1	0000h
BUSCON0 b	FF0Ch	86h	Bus configuration register 0	0xx0h
MDC b	FF0Eh	87h	CPU multiply divide control register	0000h
PSW b	FF10h	88h	CPU program status word	0000h
SYSCON b	FF12h	89h	CPU system configuration register	0xx0h
BUSCON1 b	FF14h	8Ah	Bus configuration register 1	0000h

Table 70. X-registers ordered by address (continued)

Name	Physical address	Description	Reset value
XPICON	EB26h	Extended port input threshold control register	- - 00h
XIR2SEL	EB30h	X-Interrupt 2 selection register	0000h
XIR2SET	EB32h	X-Interrupt 2 set register (write-only)	0000h
XIR2CLR	EB34h	X-Interrupt 2 clear register (write-only)	0000h
XP1DIDIS	EB36h	Port 1 digital disable register	0000h
XIR3SEL	EB40h	X-Interrupt 3 selection register	0000h
XIR3SET	EB42h	X-Interrupt 3 set selection register (write-only)	0000h
XIR3CLR	EB44h	X-Interrupt 3 clear selection register (write-only)	0000h
XMISC	EB46h	XBUS miscellaneous features register	0000h
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
XEMU3	EB7Ch	XBUS emulation register 3 (write-only)	XXXXh
XPEREMU	EB7Eh	XPERCON copy for emulation (write-only)	XXXXh
XPWMCON0	EC00h	XPWM module control register 0	0000h
XPWMCON1	EC02h	XPWM module control register 1	0000h
XPOLAR	EC04h	XPWM module channel polarity register	0000h
XPWMCON0SET	EC06h	XPWM module set control register 0 (write-only)	0000h
XPWMCON0CLR	EC08h	XPWM module clear control reg. 0 (write-only)	0000h
XPWMCON1SET	EC0Ah	XPWM module set control register 0 (write-only)	0000h
XPWMCON1CLR	EC0Ch	XPWM module clear control reg. 0 (write-only)	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
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
XPW0	EC30h	XPWM module pulse width register 0	0000h

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 $\overline{CS}$ 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 45* 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}, $\overline{RSTIN}$ 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 channels 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 Auto Scan Continuous Conversion mode on all 16 channels of Port5
   - All interrupts generated by XPWM, RTC, Timers and ADC are not serviced
10. The power supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 45* 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}, $\overline{RSTIN}$ pin at V_{IH1min}: **This implies I/O current is not considered**. The device is doing the following:
    - Fetching code from all sectors of both IFlash and XFlash, accessing in read (few fetches) and write to XRAM
    - 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 channels 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 Auto Scan Continuous Conversion mode on all 16 channels of Port5
    - All interrupts generated by XPWM, RTC, Timers and ADC are not serviced
11. The Idle mode supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 44* below. These parameters are tested and at maximum CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}, $\overline{RSTIN}$ pin at V_{IH1min}.
12. This parameter is tested including leakage currents. All inputs (including pins configured as inputs) at 0 to 0.1V or at V_{DD} - 0.1V to V_{DD}, V_{AREF} = 0V, all outputs (including pins configured as outputs) disconnected. Furthermore, the Main Voltage Regulator is assumed off: In case it is not, additional 1mA shall be assumed.

Figure 44. Port2 test mode structure



1 For Port2 complete structure refer also to *Figure 44*.

6. DNL, INL, OFS and TUE are tested at V_{AREF} = 5.0V, V_{AGND} = 0V, V_{DD} = 5.0V. 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 (± 2LSB) is also guaranteed with an overload condition (see I_{OV} specification) occurring on a maximum of 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 a maximum of 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 (± 7LSB). To obtain the same accuracy, the negative injection current on Port1 pins shall 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 47*

23.7.1 Conversion timing control

When a conversion starts, 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 in several successive steps into a digital value, which corresponds 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 must be drawn from the sources for sampling and changing charges depends on the duration of each step because the capacitors must reach their final voltage level within the given time, at least with a certain approximation. However, the maximum current that a source can deliver depends on its internal resistance.

The time that the two different actions take during conversion (sampling and converting) can be programmed within a certain range in the ST10F276E relative to the CPU clock. The absolute time consumed by the different conversion steps is therefore independent from the general speed of the controller. This allows adjusting the ST10F276E A/D converter to the properties of the system:

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

High internal resistance can be achieved by programming the respective times to a higher value or to 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. However, the conversion rate in this case may be considerably lower.

The conversion times are programmed via the upper 4 bits of register ADCON. Bit fields ADCTC and ADSTC define the basic conversion time and in particular the partition between the sample phase and comparison phases. The table below lists the possible combinations. The timings refer to the unit TCL, where f_{CPU} = 1/2TCL. A complete conversion time includes the conversion itself, the sample time and the time required to transfer the digital value to the result register.

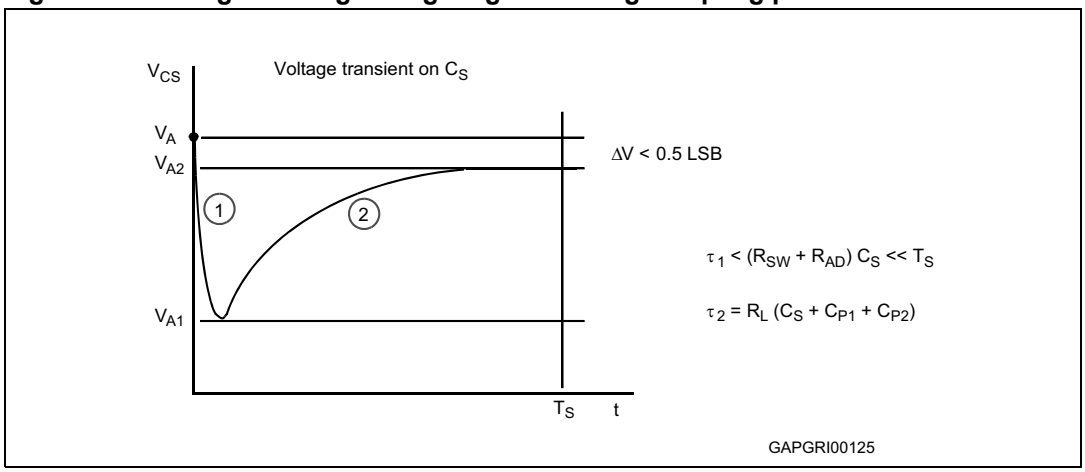
Table 94. A/D converter programming

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

The formula above places constraints on 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} are initially charged at the source voltage V_A (refer to the equivalent circuit shown in *Figure 47*), when the sampling phase is started (A/D switch close), a charge sharing phenomena is installed.

Figure 48. Charge sharing timing diagram during sampling phase



In particular two different transient periods can be distinguished (see *Figure 48*):

1. A first and quick charge transfer from the internal capacitances 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 reported in parallel to C_{P1} (call $C_P = C_{P1} + C_{P2}$), the two capacitances 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 also be robust in the very worst case: The sampling time T_S is always much longer than the internal time constant:

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

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

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

2. A second charge transfer also involves 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

1. Supposing to design 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 12V, the total series resistance can be defined as:

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

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

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

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 others must now be verified. The relation which allows to minimize 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 = 5V):

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

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

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

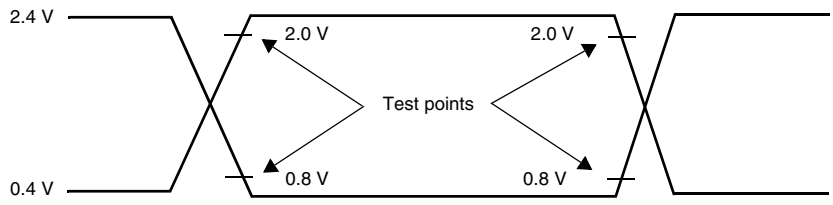
$$10\tau_2 = 10R_L(C_S + C_{P1} + C_{P2}) = 290ns < T_S = 1\mu s$$

For a complete set of parameters characterizing the ST10F276E A/D converter equivalent circuit, refer to *Table 93: A/D converter characteristics on page 187*.

23.8 AC characteristics

23.8.1 Test waveforms

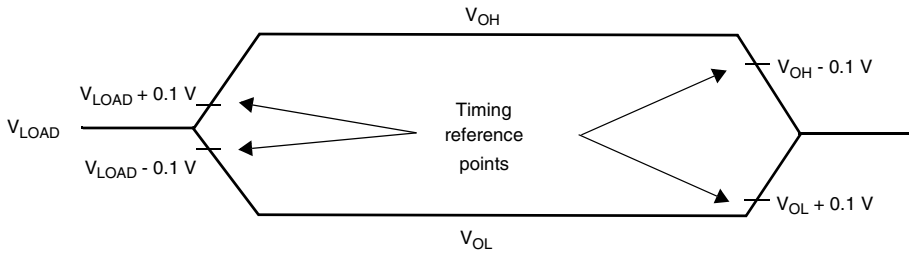
Figure 50. Input/output waveforms



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

GAPGRI00127

Figure 51. Float waveforms



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

GAPGRI00128

23.8.17 External memory bus timing

In the next sections the external memory bus timings are described. The given values are computed for a maximum CPU clock of 40 MHz.

It is evident that when higher CPU clock frequency is used (up to 64 MHz), some numbers in the timing formulas become zero or negative, which in most cases is not acceptable or meaningful. In these cases, the speed of the bus settings t_A, t_C and t_F must be correctly adjusted.

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

23.8.18 Multiplexed bus

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

Table 104. Multiplexed bus

Symbol		Parameter	f_{CPU} = 40 MHz TCL = 12.5ns		Variable CPU clock 1/2 TCL = 1 to 64 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	CC	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)(1)	-	6	-	6	ns
t_{11}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)(1)		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