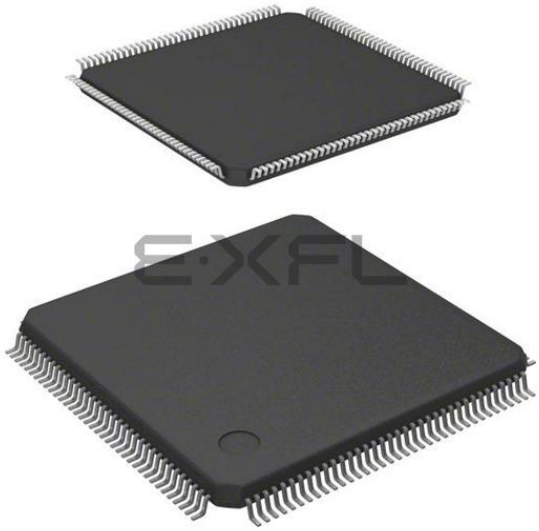


Welcome to E-XFL.COM

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"

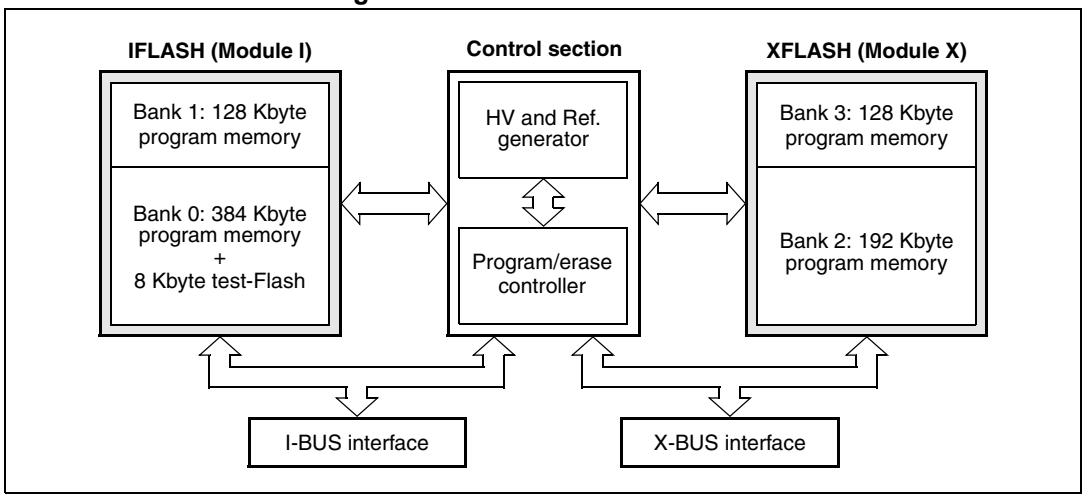
Details	
Product Status	Obsolete
Core Processor	ST10
Core Size	16-Bit
Speed	64MHz
Connectivity	ASC, CANbus, EBI/EMI, I²C, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	111
Program Memory Size	832KB (832K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	68K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-BQFP
Supplier Device Package	144-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st10f276z5q3tr

4 Internal Flash memory

4.1 Overview

The on-chip Flash is composed by two matrix modules each one containing one array divided in two banks that can be read and modified independently one of the other: one bank can be read while another bank is under modification.

Figure 4. Flash modules structure



The write operations of the 4 banks are managed by an embedded Flash program/erase controller (FPEC). The high voltages needed for program/erase operations are internally generated.

The data bus is 32-bit wide. Due to ST10 core architecture limitation, only the first 512 Kbytes are accessed at 32-bit (internal Flash bus, see I-BUS), while the remaining 320 Kbytes are accessed at 16-bit (see X-BUS).

4.2 Functional description

4.2.1 Structure

The following table shows the address space reserved to the Flash module.

Table 3. Flash modules absolute mapping

Description	Addresses	Size
IFLASH sectors	0x00 0000 to 0x08 FFFF	512 Kbyte
XFLASH sectors	0x09 0000 to 0x0D FFFF	320 Kbyte
Registers and Flash internal reserved area	0x0E 0000 to 0x0E FFFF	64 Kbyte

Table 27. Summary of access protection level (continued)

Access type	Read IFLASH / Jump to IFLASH	Read XFLASH /Jump to XFLASH	Read FLASH Registers	Write FLASH Registers
Fetching from IRAM	No / Yes	Yes / Yes	Yes	No
Fetching from XRAM	No / Yes	Yes / Yes	Yes	No
Fetching from External Memory	No / Yes	Yes / Yes	Yes	No

4.5.10 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 WyPx of FNVWPXRH/L-FNVWPIRH/L registers.

4.5.11 Temporary unprotection

Bits WyPx of FNVWPXRH/L-FNVWPIRH/L can be temporary unprotected by executing the Set Protection operation and writing 1 into these bits.

Bit ACCP can be temporary unprotected by executing the Set Protection operation and writing 1 into these bits, but only if these write instructions are executed from the Flash Modules.

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

It is not necessary to temporary unprotect an access protected Flash in order to update the code: it is, in fact, sufficient to execute the updating instructions from another Flash Bank.

In reality, when a temporary 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 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).

Table 28. 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

5 Bootstrap loader

The ST10F276Z5 features innovative boot capabilities in order to support:

- User defined bootstrap (see *Alternate bootstrap loader*)
- 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 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

5.5.2 Hardware aspects

Although the new bootstrap loader is designed to be compatible with the old bootstrap loader, there are a few hardware requirements for the new bootstrap loader as summarized in *Table 35*.

Table 35. Hardware topics summary

Actual bootstrap loader	New bootstrap loader	Comments
P4.5 can be used as output in BSL mode.	P4.5 cannot be used as user output in BSL mode, but only as CAN1_RxD or input or address segments.	
Level on CAN1_RxD can change during boot Step 2.	Level on CAN1_RxD must be stable at '1' during boot Step 2.	External pull-up on P4.5 needed.

5.6 Alternate boot mode (ABM)

5.6.1 Activation

Alternate boot is activated with the combination '01' on Port0L[5..4] at the rising edge of RSTIN.

5.6.2 Memory mapping

The ST10F276Z5 has the same memory mapping for standard boot mode and for alternate boot mode:

- Test-Flash: Mapped from 00'0000h. The Standard Bootstrap Loader can be started by executing a jump to the address of this routine (JMPS 00'xxxx; *address to be defined*).
- User Flash: The User Flash is divided in two parts: The IFLASH, visible only for memory reads and memory writes (no code fetch) and the XFLASH, visible for any ST10 access (memory read, memory write and code fetch).
- All device XRAM and Xperipherals modules can be accessed if enabled in XPERCON register.

Note: The alternate boot mode can be used to reprogram the whole content of the device User Flash (except Block 0 in Bank 2, where the alternate boot is mapped into).

5.6.3 Interrupts

The ST10 interrupt vector table is always mapped from address 00'0000h.

As a consequence, interrupts are not allowed in Alternate Boot mode; all maskable and non maskable interrupts must be disabled.

6 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 ST10F276Z5 instructions can be executed in one instruction cycle which requires 31.25ns at 64 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.

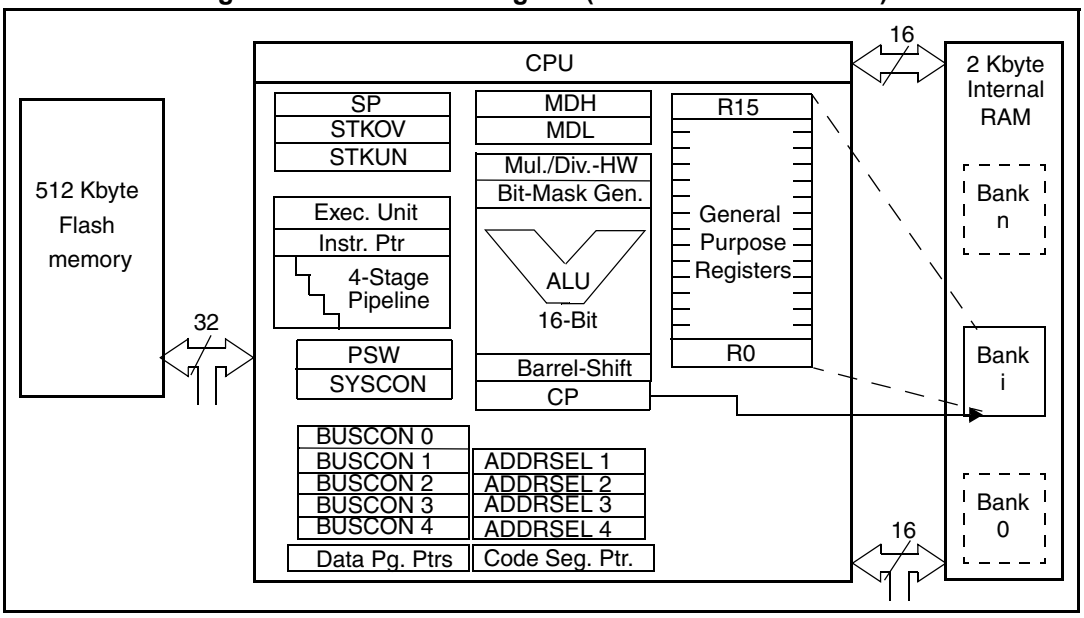
The CPU uses a bank of 16 word registers to run the current context. This bank of General Purpose Registers (GPR) is physically stored within the on-chip Internal RAM (IRAM) area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU.

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 14. CPU Block Diagram (MAC Unit not included)



15 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, like start or stop events, arbitration lost, etc.
- 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 may be selected between Standard mode (0 to 100 kHz) and Fast I²C mode (100 to 400 kHz).

16 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: Internal Flash memory*.
- 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 8 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).

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

18 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 pre-specified 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

The *Table 59* and *Table 60* show the watchdog time range for 40 MHz and 64 MHz CPU clock respectively.

Table 59. 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

Table 60. WDTREL reload value (f_{CPU} = 64 MHz)

Reload value in WDTREL	Prescaler for f_{CPU} = 64 MHz	
	2 (WDTIN = ‘0’)	128 (WDTIN = ‘1’)
FFh	8 µs	512 µs
00h	2.048 ms	131.1 ms

19 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 61*.

Table 61. Reset event definition(1)

Reset Source	Flag	RPD Status	Conditions
Power-on reset	PONR	Low	Power-on
Asynchronous Hardware reset	LHWR	Low	(2)
Synchronous Long Hardware reset		High	$t_{\overline{RSTIN}}$ > (1032 + 12) TCL + max(4 TCL, 500 ns)
Synchronous Short Hardware reset	SHWR	High	$t_{\overline{RSTIN}}$ > max(4 TCL, 500 ns) $t_{\overline{RSTIN}} \leq$(1032 + 12) TCL + max(4 TCL, 500 ns)
Watchdog Timer reset	WDTR	(3)	WDT overflow
Software reset	SWR		SRST instruction execution

1. See next *Section 19.1* for more details on minimum reset pulse duration.
2. $\overline{RSTIN}$ pulse should be longer than 500 ns (Filter) and than settling time for configuration of Port0.
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 *Section 19.4*, *Section 19.5* and *Section 19.6*).

19.1 Input filter

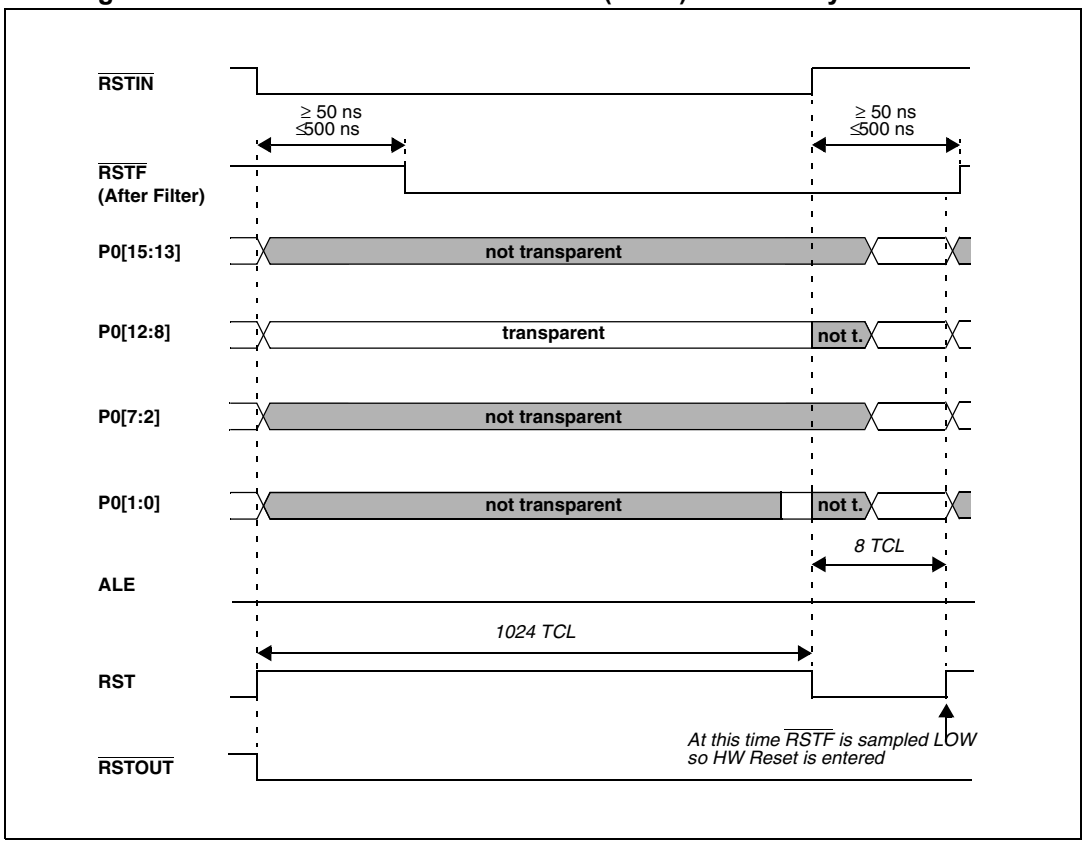
On $\overline{RSTIN}$ input pin an on-chip RC filter is implemented. It is sized to filter all the spikes shorter than 50 ns. On the other side, a valid pulse shall be longer than 500 ns to grant that ST10 recognizes a reset command. In between 50 ns and 500 ns a pulse can either be filtered or recognized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

For this reason all minimum durations mentioned in this Chapter for the different kind of reset events shall be carefully evaluated taking into account of 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 64 MHz, 4 TCL is 31.25 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.

Figure 36. SW / WDT bidirectional RESET ($\overline{EA}$=0) followed by a HW RESET



19.7 Reset circuitry

Internal reset circuitry is described in *Figure 39*. The $\overline{RSTIN}$ pin provides an internal pull-up resistor of 50kΩ to 250kΩ (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 $\overline{RSTOUT}$ pin.

This is the case of an external memory running codes before EINIT (end of initialization) instruction is executed. $\overline{RSTOUT}$ 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 200µA. 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 device is to insert a capacitor C1 between $\overline{RSTIN}$ 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 $\overline{RSTIN}$ provides an internal pull-up device equalling a resistor of 50kΩ to 250kΩ (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.

Figure 38. System reset circuit

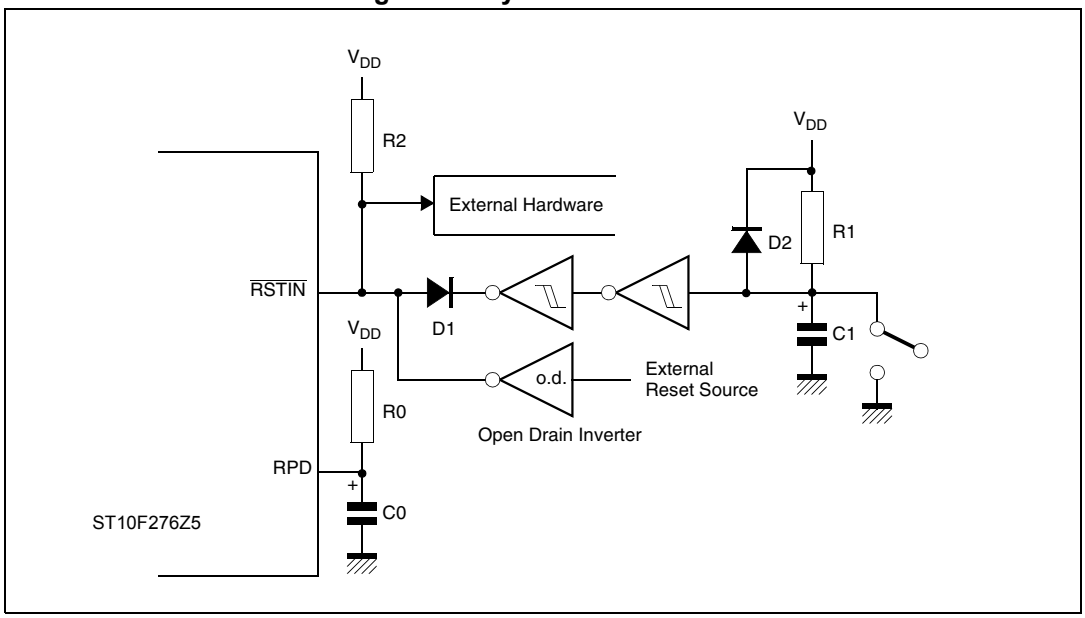


Figure 39. Internal (simplified) reset circuitry

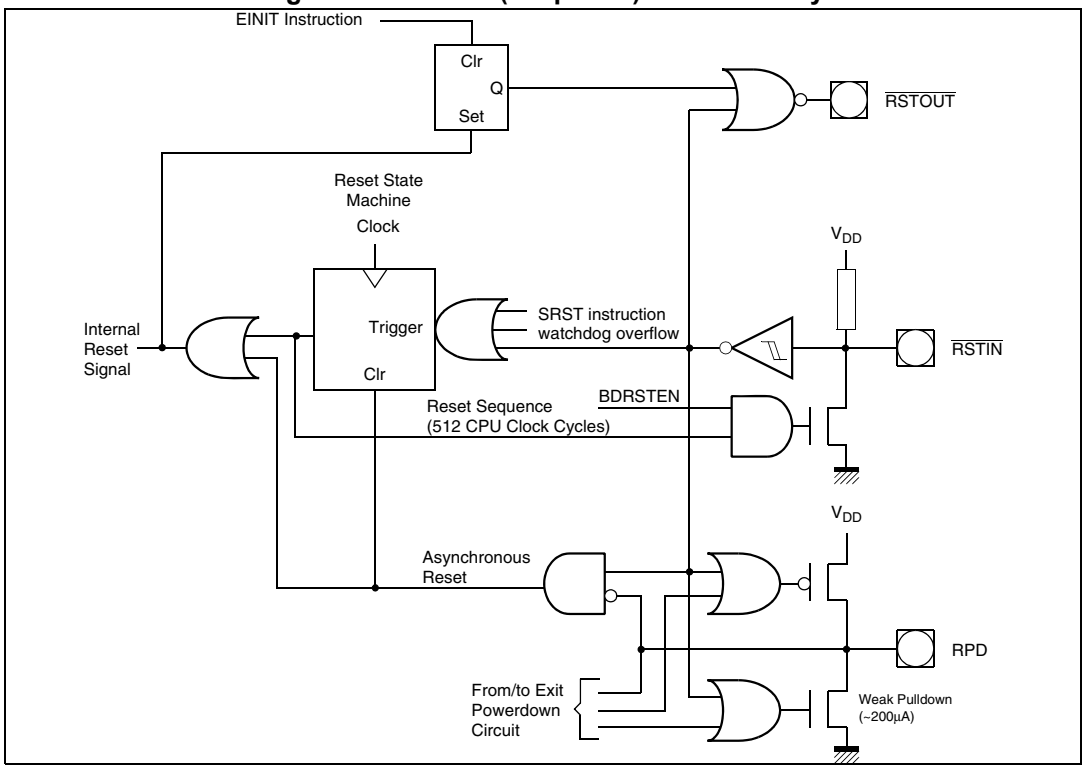


Table 70. X-Registers ordered by name (continued)

Name	Physical address	Description	Reset value
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
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

Table 79. SYSCON description (continued)

Bit	Function
WRCFG	Write configuration control (inverted copy of WRC bit of RP0H) '0': Pins WR and BHE retain their normal function. '1': Pin WR acts as WRL, pin BHE acts as WRH.
CLKEN	System clock output enable (CLKOUT) '0': CLKOUT disabled, pin may be used for general purpose I/O. '1': CLKOUT enabled, pin outputs the system clock signal or a prescaled value of system clock according to XCLKOUTDIV register setting.
BYTDIS	Disable/enable control for pin BHE (set according to data bus width) '0': Pin BHE enabled. '1': Pin BHE disabled, pin may be used for general purpose I/O.
ROMEN	Internal memory enable (set according to pin EA during reset) '0': Internal memory disabled: Accesses to the IFlash Memory area use the external bus. '1': Internal memory enabled.
SGTDIS	Segmentation disable/enable control '0': Segmentation enabled (CSP is saved/restored during interrupt entry/exit). '1': Segmentation disabled (Only IP is saved/restored).
ROMS1	Internal memory mapping '0': Internal memory area mapped to segment 0 (00'0000h...00'7FFFh). '1': Internal memory area mapped to segment 1 (01'0000h...01'7FFFh).
STKSZ	System stack size Selects the size of the system stack (in the internal I-RAM) from 32 to 1024 words.

BUSCON0 (FF0Ch / 86h) SFR Reset value: 0xx0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN0	CSREN0	RDYPOL0	RDYEN0	-	BUSACT0	ALECTL0	-	BTYP		MTTC0	RWDC0	MCTC			
RW	RW		RW		RW	RW		RW		RW	RW	RW			

BUSCON1 (FF14h / 8Ah) SFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN1	CSREN1	RDYPOL1	RDYEN1	-	BUSACT1	ALECTL1	-	BTYP		MTTC1	RWDC1	MCTC			
RW	RW	RW	RW		RW	RW		RW		RW	RW	RW			

BUSCON2 (FF16h / 8Bh) SFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN2	CSREN2	RDYPOL2	RDYEN2	-	BUSACT2	ALECTL2	-	BTYP		MTTC2	RWDC2	MCTC			
RW	RW		RW		RW	RW		RW		RW	RW	RW			

BUSCON3 (FF18h / 8Ch) SFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN3	CSREN3	RDYPOL3	RDYEN3	-	BUSACT3	ALECTL3	-	BTYP		MTTC3	RWDC3	MCTC			
RW	RW		RW		RW	RW		RW		RW	RW	RW			

23.8.3 Clock generation modes

The following table associates the combinations of these 3 bits with the respective clock generation mode.

Table 97. On-chip clock generator selections

P0.15-13 (P0H.7-5)			CPU frequency $f_{CPU} = f_{XTAL} \times F$	External clock input range (1)(2)	Notes
1	1	1	F_{XTAL} x 4	4 to 8 MHz	Default configuration
1	1	0	F_{XTAL} x 3	5.3 to 10.6 MHz	
1	0	1	F_{XTAL} x 8	4 to 8 MHz	
1	0	0	F_{XTAL} x 5	6.4 to 12 MHz	
0	1	1	F_{XTAL} x 1	1 to 64 MHz	Direct Drive (oscillator bypassed) (3)
0	1	0	F_{XTAL} x 10	4 to 6.4 MHz	
0	0	1	F_{XTAL} / 2	4 to 12 MHz	CPU clock via prescaler (3)
0	0	0	F_{XTAL} x 16	4 MHz	

1. The external clock input range refers to a CPU clock range of 1...64 MHz. Moreover, the PLL usage is limited to 4-12 MHz input frequency range. All configurations need a crystal (or ceramic resonator) to generate the CPU clock through the internal oscillator amplifier (apart from Direct Drive); on the contrary, 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 limits on input frequency are 4-12 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.
3. The maximum depends on the duty cycle of the external clock signal: When 64 MHz is used, 50% duty cycle shall be granted (low phase = high phase = 7.8 ns); when 32 MHz is selected, a 25% duty cycle can be accepted (minimum phase, high or low, again equal to 7.8ns).

23.8.4 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 can therefore 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.

23.8.5 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 the 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}.

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.

Figure 70. ADC injection theoretical operation

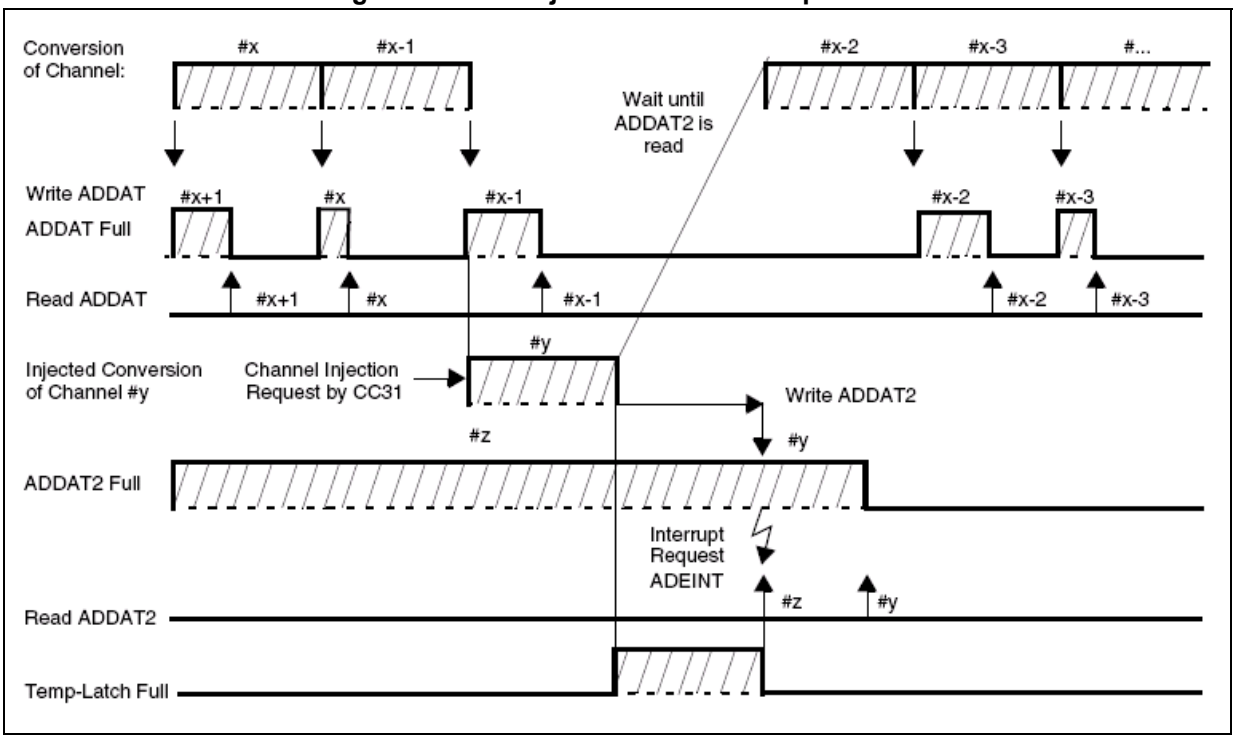
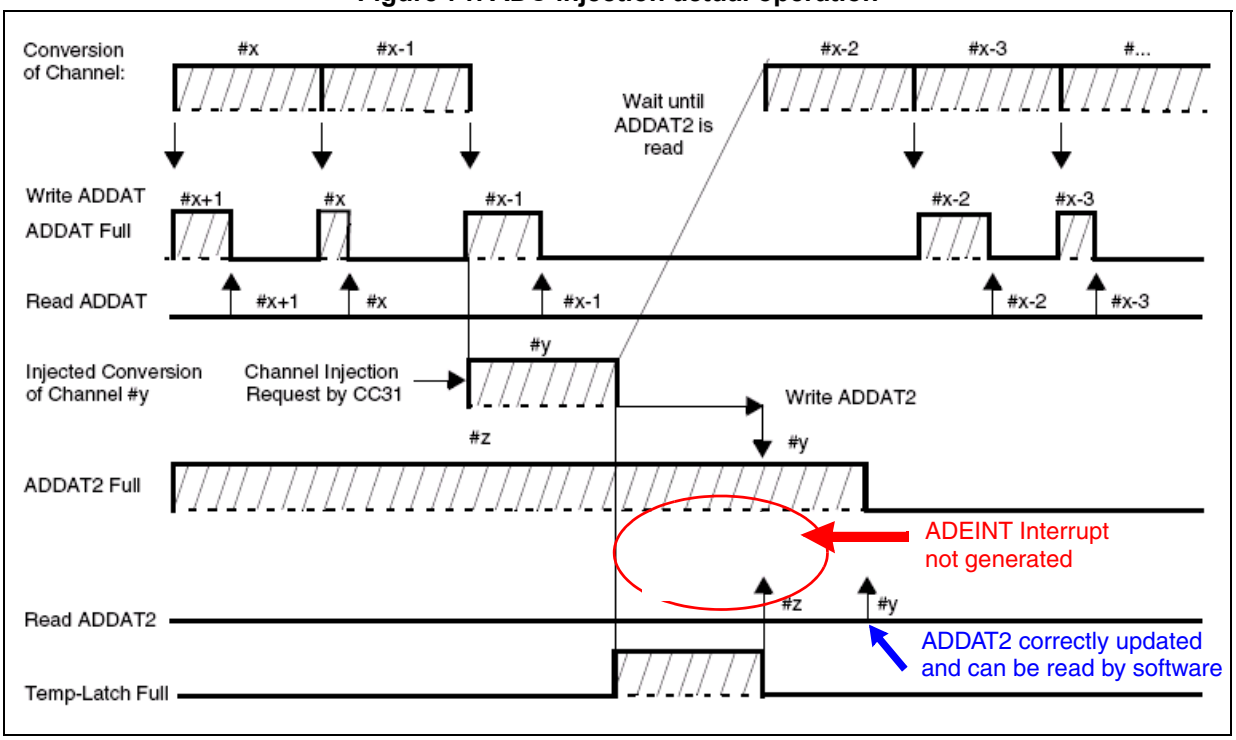


Figure 71. ADC injection actual operation



24.1.4 Spurious $\overline{\text{BREQ}}$ pulse in slave mode during external bus arbitration phase

Description

Sporadic bus errors may occur when the device operates as a slave and the $\overline{\text{HOLD}}$ signal is used for external bus arbitration.

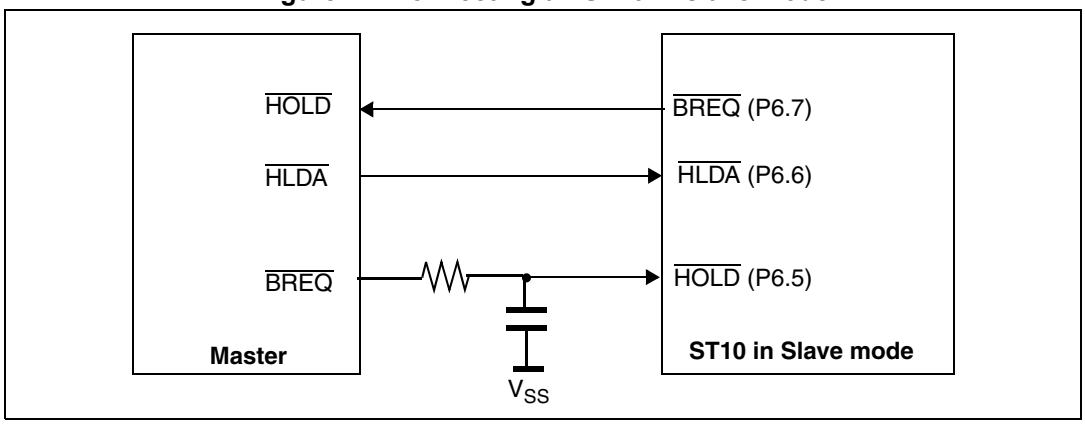
After the slave has been granted the bus, it deactivates sporadically $\overline{\text{BREQ}}$ signal for a short time, even though its access to the bus has not been completed. The master then starts accessing the bus, thus causing a bus conflict between master and slave.

Workaround

To avoid producing any spurious $\overline{\text{BREQ}}$ pulse during slave external bus arbitrations, it is necessary to ensure that the time between the $\overline{\text{HLDA}}$ assertion (Bus Acknowledge from Master device) and the following $\overline{\text{HOLD}}$ falling edge (Bus Request from Master) is longer than three clock cycles.

This can be achieved by delaying the $\overline{\text{HOLD}}$ signal with an RC circuit (see *Figure 72*).

Figure 72. Connecting an ST10 in slave mode



24.1.5 Executing PWRDN instructions

Description

The Power-down mode is not entered and the PWRDN instruction is ignored in the following cases:

- The PWRDN instruction is executed while $\overline{\text{NMI}}$ is high (PWDCFG bit of the SYSCON register cleared)
- The PWRDN instruction is executed while at least one of the Port 2 pins used to exit from Power-down mode (PWDCFG bit of the SYSCON register is set) is at the active level.

However, in certain cases, the PWRDN instruction is not ignored, no further instructions are fetched from external memory, and the CPU is in a quasi-idle state. This problem only occurs in the following situations:

1. The instructions following the PWRDN instruction are located in an external memory and a multiplexed bus configuration with memory tri-state waitstate (bit MTTCx = 0) is used.
2. The instruction preceding the PWRDN instruction writes to the external memory or to an XPeripheral (such as XRAM or CAN) and the instructions following the PWRDN instruction are located in external memory area. In this case, the problem occurs for all bus configurations.

Note: The on-chip peripherals, such as the watchdog timer, still operate properly. If the watchdog timer is not disabled, it resets the device upon an overflow event. However, interrupts and PEC transfers cannot be processed. Power-down mode is entered if the $\overline{\text{NMI}}$ signal is asserted low while the device is in this quasi-idle state.

No problem occurs and the device normally enters Power-down mode if the $\overline{\text{NMI}}$ pin is held low (PWDCFG = 0) or if all Port 2 pins used to exit from Power-down mode are at inactive level (PWDCFG = 1).

Workaround

To prevent this problem from occurring, the PWRDN instruction must be preceded by instructions performing write operations to external memory area or to an XPeripheral. Otherwise, it is recommended to insert a NOP instruction before PWRDN.

When using a multiplexed bus with memory tri-state wait state, the PWRDN instruction must be executed from internal RAM or XRAM.

24.1.6 Behavior of CAPCOM outputs in COMPARE mode 3

Description

When a CAPCOM channel is configured in compare mode 3, then the related output level switches to high when the allocated timer, Tx, matches the related CAPCOM register, CCy. When an overflow occurs on the CAPCOM timer Tx, it is reloaded with TxREL content and the output pin is cleared. The output pin level does not change if TxREL and CCy have the same value.

The related CAPCOM output stays low when the CAPCOM channel is configured in compare mode 3 and TxREL and Tx related timer registers are loaded with the same value as CCy. This is obtained by executing the following instructions:

```
MOV TxREL, #CCy value x =0,1,7,8
MOV Tx, #CCy value x = 0,1,7,8
MOV TxxCON, #data or bfldl/bfldh TxxCON , #mask, #data i.e. an
access is made to the T01CON or T78CON register.
```