Contents

List of tables

List of figures

1 Introduction

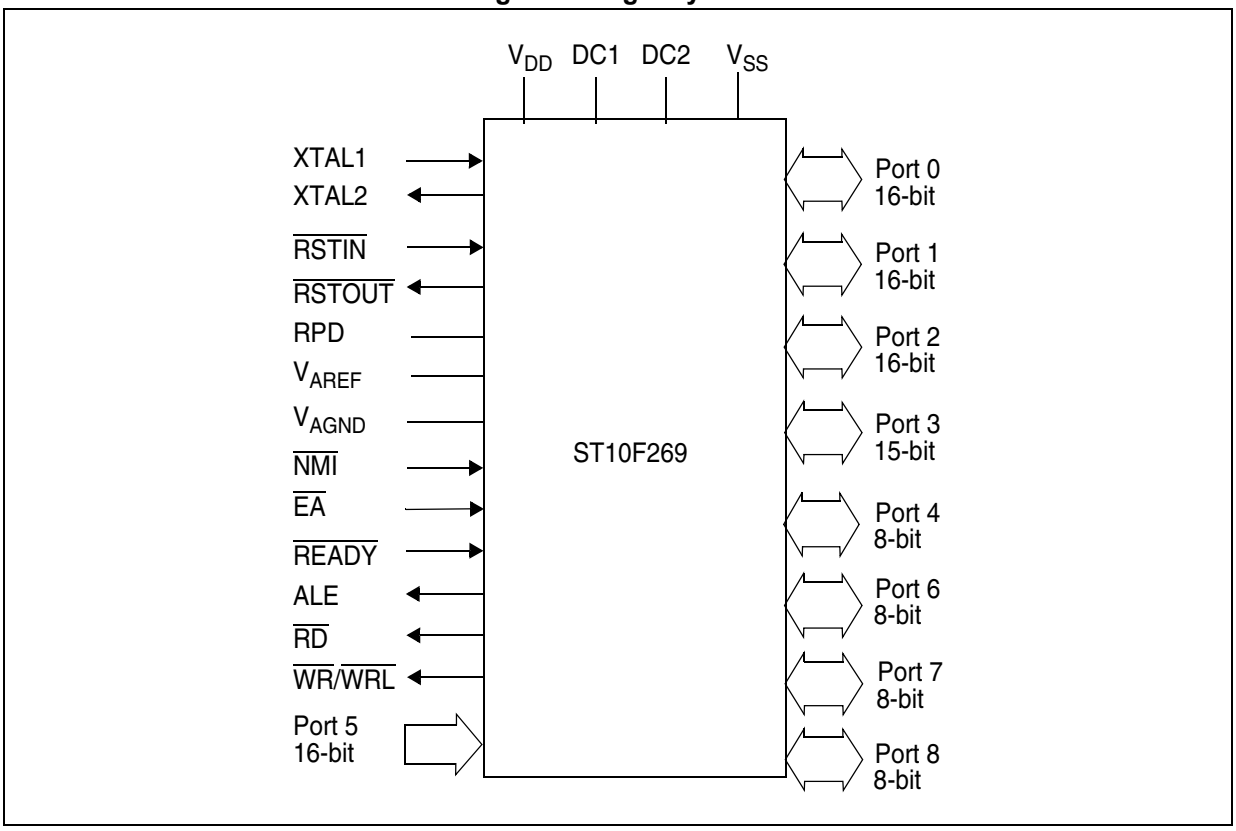
The ST10F269Z1-ST10F269Z2 are derivatives of the STMicroelectronics ST10 family of 16-bit single-chip CMOS microcontrollers. It combines high CPU performance (up to 20 million instructions per second) with high peripheral functionality and enhanced I/O-capabilities. It also provides on-chip high-speed single voltage Flash memory, on-chip high-speed RAM, and clock generation via PLL.

The ST10F269Z1-ST10F269Z2 are processed in 0.35 mm CMOS technology. The MCU core and the logic is supplied with a 5 V to 2.7 V on chip voltage regulator. The part is supplied with a single 5 V supply and I/Os work at 5 V.

The device is upward compatible with the ST10F168 device, with the following set of differences:

- The Multiply/Accumulate unit is available as standard. This MAC unit adds powerful DSP functions to the ST10 architecture, but maintains full compatibility for existing code.
- Flash control interface is now based on STMicroelectronics third generation of stand-alone Flash memories, with an embedded Erase/Program Controller. This completely frees up the CPU during programming or erasing the Flash.
- 128-Kbyte Flash Option
- Two dedicated pins (DC1 and DC2) on the 144-pin package are used for decoupling the internally generated 2.7V core logic supply. Do not connect these two pins to 5.0 V external supply. Instead, these pins should be connected to a decoupling capacitor (ceramic type, value ≥ 330 nF).
- The A/D Converter characteristics are different from previous ST10 derivatives ones. Refer to *Section 21.3.1: A/D Converter Characteristics*.
- The characterization is performed with C_L = 50 pF max on output pins. Refer to *Section 21.3: DC Characteristics*.
- In order to reduce EMC, the rise/fall time and the sink/source capability of the drivers of the I/O pads are programmable. Refer to *Section 12.2: I/Os Special Features*.
- The Real Time Clock functionality is added.
- The external interrupt sources can be selected with the EXISEL register.
- The reset source is identified by a dedicated status bit in the WDTCON register.

Figure 1. Logic Symbol



2 Pin Data

Figure 2. Pin Configuration (top view)

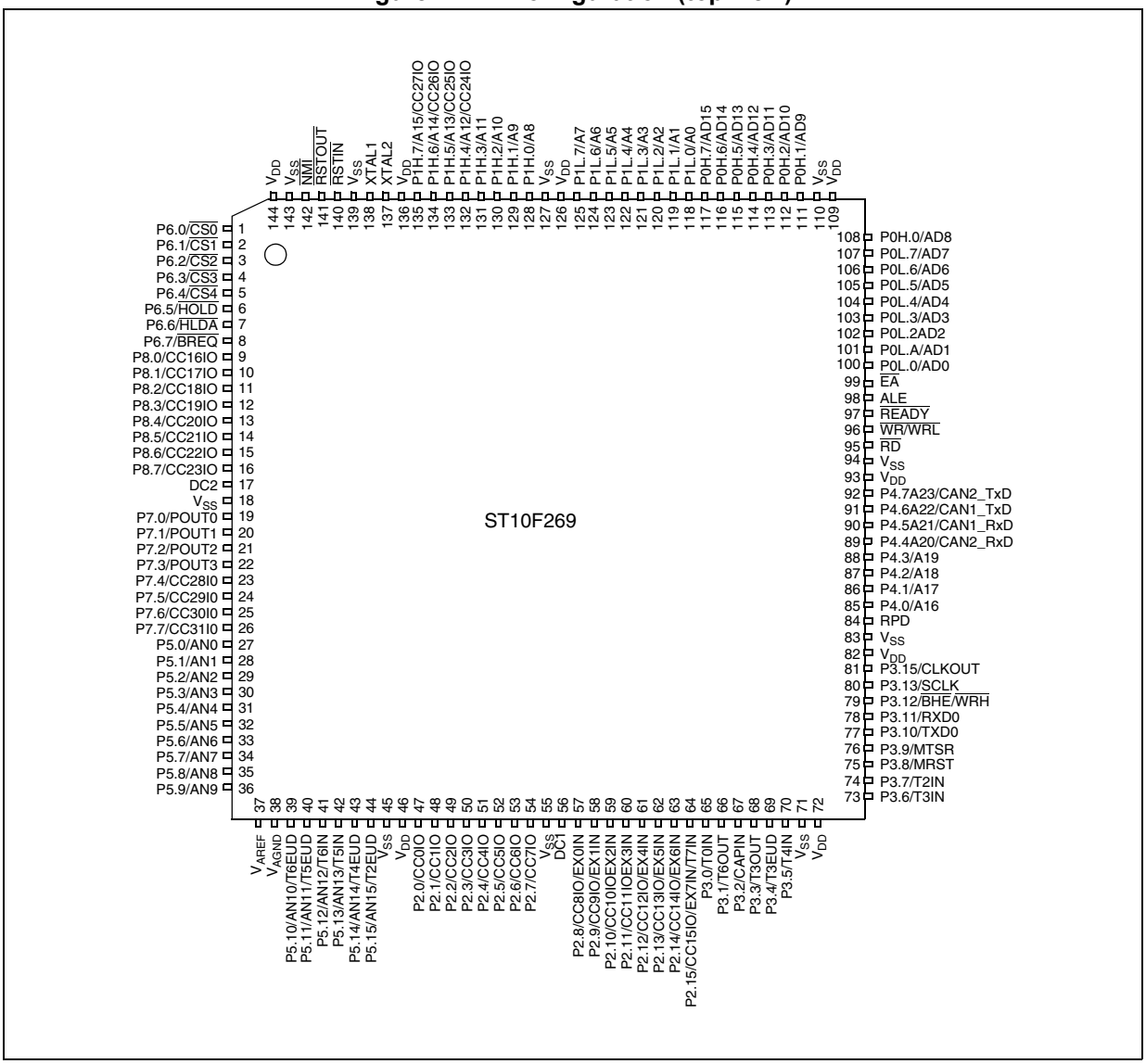


Table 1. Pin Description

Symbol	Pin	Type	Function		
P6.0 - P6.7	1 - 8	I/O	8-bit bidirectional I/O port, bit-wise 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 following Port 6 pins have alternate functions:		
	1	O	P6.0	CS0	Chip Select 0 Output
	...	...	...	...	...
	5	O	P6.4	CS4	Chip Select 4 Output
	6	I	P6.5	HOLD	External Master Hold Request Input
	7	O	P6.6	HLDA	Hold Acknowledge Output
	8	O	P6.7	BREQ	Bus Request Output
P8.0 - P8.7	9-16	I/O	8-bit bidirectional I/O port, bit-wise 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 special). The following Port 8 pins have alternate functions:		
	9	I/O	P8.0	CC16IO	CAPCOM2: CC16 Capture Input / Compare Output
	...	...	...	...	...
	16	I/O	P8.7	CC23IO	CAPCOM2: CC23 Capture Input / Compare Output
P7.0 - P7.7	19-26	I/O	8-bit bidirectional I/O port, bit-wise 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 special). The following Port 7 pins have alternate functions:		
	19	O	P7.0	POUT0	PWM Channel 0 Output
	...	...	...	...	...
	22	O	P7.3	POUT3	PWM 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:		
	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

Table 1. Pin Description (continued)

Symbol	Pin	Type	Function		
P2.0 - P2.7 P2.8 - P2.15	47-54 57-64	I/O	16-bit bidirectional I/O port, bit-wise 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 special). 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
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, bit-wise 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 special). The following Port 3 pins have alternate functions:		
	65	I	P3.0	T0IN	CAPCOM 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	MRST	SSC Master-Receiver / Slave-Transmitter I/O
	76	I/O	P3.9	MTSR	SSC Master-Transmitter / Slave-Receiver O/I
	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	SCLK	SSC Master Clock Output / Slave Clock Input
	81	O	P3.15	CLKOUT	System Clock Output (=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 bit-wise 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 special). Port 4.6 & 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
	90	O	P4.5	A21	Segment Address Line
		I		CAN1_RxD	CAN1 Receive Data Input
	91	O	P4.6	A22	Segment Address Line
		O		CAN1_TxD	CAN1 Transmit Data Output
	92	O	P4.7	A23	Most Significant Segment Address Line
		O		CAN2_TxD	CAN2 Transmit Data Output
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.		
$\overline{EA}$	99	I	External Access Enable pin. A low level applied to this pin during and after Reset forces the ST10F269Z1-ST10F269Z2 to start the program from the external memory space. A high level forces the MCU to start in the internal memory space.		

Table 1. Pin Description (continued)

Symbol	Pin	Type	Function
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, bit-wise programmable for input or output via direction bit. Programming an I/O pin as input forces the corresponding output driver to high impedance state. 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 – D7D0 - D7 P0H.0 – P0H.7I/OD8 - D15 **Multiplexed bus modes** Data Path Width:8-bit 16-bit P0L.0 – P0L.7:AD0 – AD7AD0 - AD7 P0H.0 – P0H.7A8 – A15AD8 - 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, bit-wise 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 and also after switching from a demultiplexed bus mode to a multiplexed bus mode. 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 Oscillator amplifier and/or external clock input. XTAL2 Oscillator amplifier circuit output. 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.
XTAL2	137	O	
$\overline{\text{RSTIN}}$	140	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the ST10F269Z1-ST10F269Z2. 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 ST10F269Z1-ST10F269Z2 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.
V_{AGND}	38	-	A/D converter reference ground.

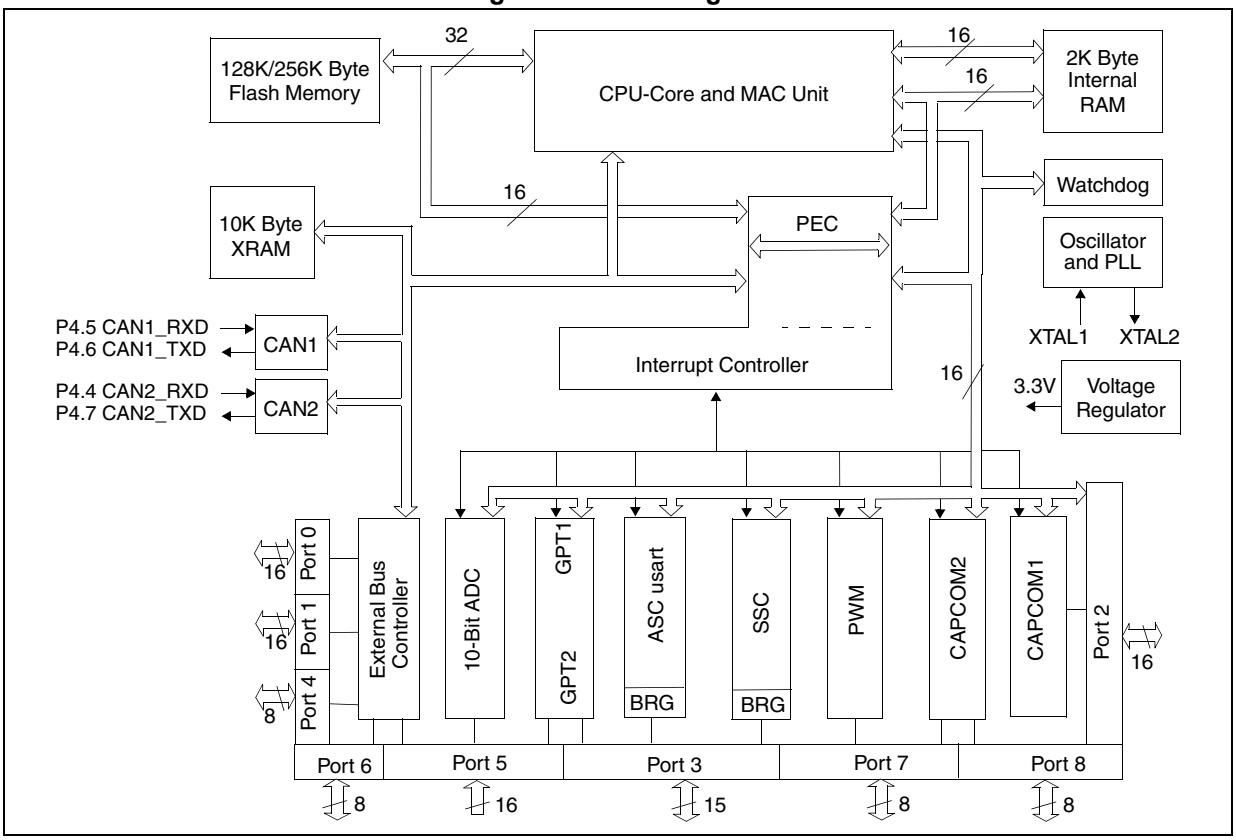
Table 1. Pin Description (continued)

Symbol	Pin	Type	Function
RPD	84	-	Timing pin for the return from interruptible powerdown mode and synchronous / asynchronous reset selection.
V_{DD}	46, 72, 82,93, 109, 126, 136, 144	-	Digital Supply Voltage = +5 V during normal operation and idle mode.
V_{SS}	18,45, 55,71, 83,94, 110, 127, 139, 143	-	Digital Ground.
DC1 DC2	56 17	-	2.7V Decoupling pin: a decoupling capacitor of ≥ 330 nF must be connected between this pin and nearest V_{SS} pin.

3 Functional Description

The architecture of ST10F269Z1-ST10F269Z2 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 ST10F269Z1-ST10F269Z2.

Figure 3. Block Diagram



4 Memory Organization

The memory space of the ST10F269Z1-ST10F269Z2 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 Byte wise or Word wise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

- **Flash**: 128 K or 256 Kbytes of on-chip Flash memory.
- **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.
- **XRAM**: 10 Kbytes of on-chip extension RAM (single port XRAM) is provided as a storage for data, user stack and code.

  The XRAM is divided into 2 areas, the first 2 Kbytes named XRAM1 and the second 8 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 and 62.5 ns access at 32 MHz CPU clock). Byte and Word accesses are allowed.

  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 00’C000h - 00’DFFFh if XPEN (bit 2 of SYSCON register), and XRAM2 (bit 3 of XPERCON register are set). If bit XRAM2EN or XPEN is cleared, then any access in the address range 00’C000h - 00’DFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register.
  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 new XPERCON register. Accesses to the CAN Module use demultiplexed addresses and a 16-bit data bus (byte accesses are possible). Two wait states give an access time of 100 ns at 40 MHz CPU clock (or 125 ns at 32 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 (byte accesses are possible). Two wait states give an access time of 100 ns at 40 MHz CPU clock (or 125 ns at 32 MHz CPU clock). No tri-state wait states are used.
  In order to meet the needs of designs where more memory is required than is provided

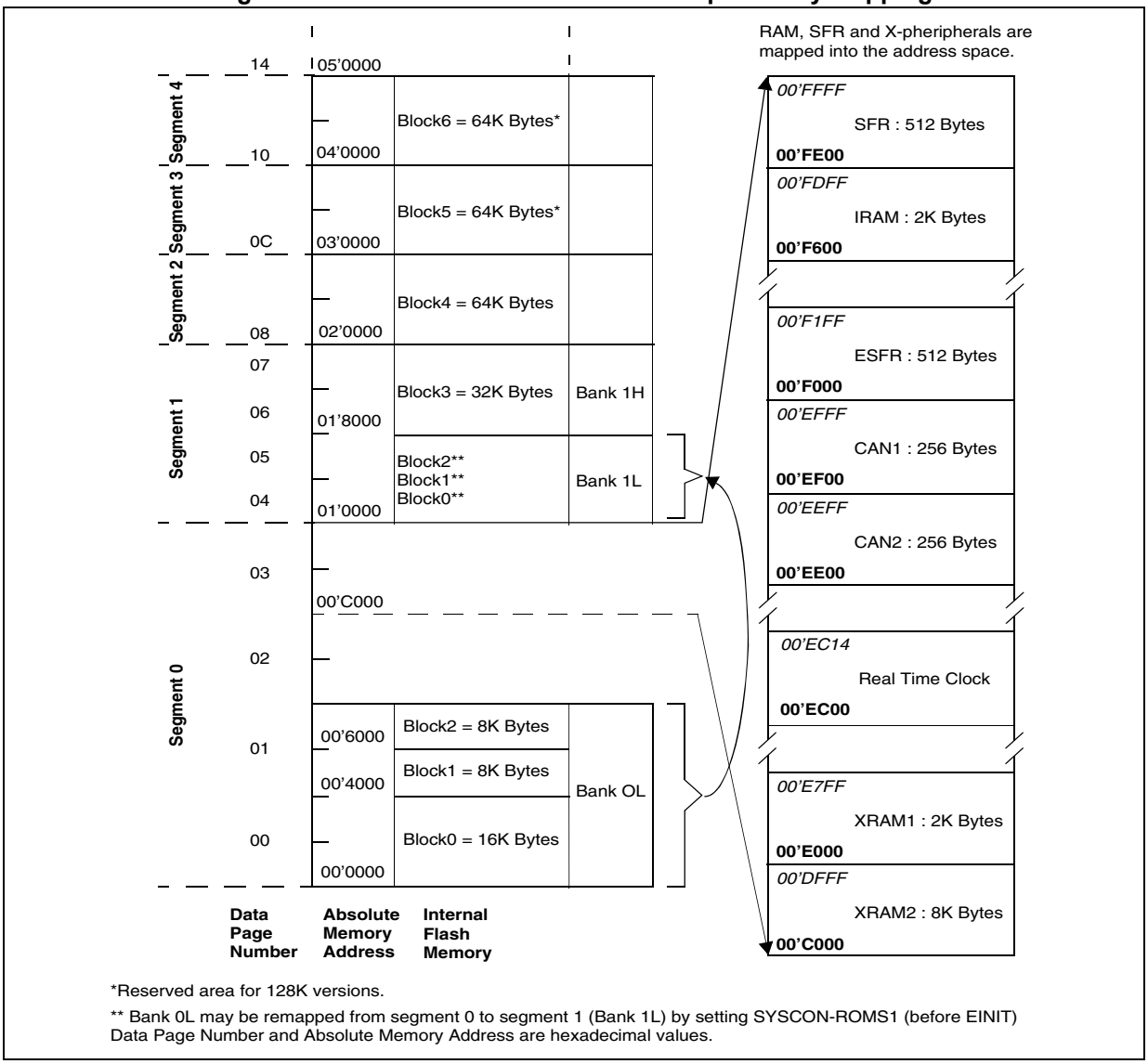
on chip, up to 16M Bytes of external RAM and/or ROM can be connected to the microcontroller.

Note: If one or the two CAN modules are used, Port 4 cannot be programmed to output all 8 segment address lines. Thus, only 4 segment address lines can be used, reducing the external memory space to 5 Mbytes (1 Mbyte per CS line).

- **Visibility of XBUS Peripherals**

  In order to keep the ST10F269Z1-ST10F269Z2 compatible with the ST10C167 and with the ST10F167, the XBUS peripherals can be selected to be visible and / or accessible on the external address / data bus. CAN1EN and CAN2EN bits of 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 peripheral, thus the peripheral is not visible and not available. Refer to *Section 20: Special Function Register Overview*.

Figure 4. ST10F269Z1-ST10F269Z2 On-chip Memory Mapping



*Reserved area for 128K versions.

** Bank 0L may be remapped from segment 0 to segment 1 (Bank 1L) by setting SYSCON-ROMS1 (before EINIT)
Data Page Number and Absolute Memory Address are hexadecimal values.

XPERCON (F024H / 12H)

ESFR Reset Value: - - 05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	RTCEN	XRAM2EN	XRAM1EN	CAN2EN	CAN1EN
											RW	RW	RW	RW	RW

CAN1EN	**CAN1 Enable Bit** ‘0’: Accesses to the on-chip CAN1 XPeripheral and its functions are disabled. P4.5 and P4.6 pins can be used as general purpose I/Os. Address range 00’EF00h-00’EFFFh is only directed to external memory if CAN2EN is also ‘0’. ‘1’: The on-chip CAN1 XPeripheral is enabled and can be accessed.
CAN2EN	**CAN2 Enable Bit** ‘0’: Accesses to the on-chip CAN2 XPeripheral and its functions are disabled. P4.4 and P4.7 pins can be used as general purpose I/Os. Address range 00’EE00h-00’EEFFh is only directed to external memory if CAN1EN is also ‘0’. ‘1’: The on-chip CAN2 XPeripheral is enabled and can be accessed.
XRAM1EN	**XRAM1 Enable Bit** ‘0’: Accesses to external memory within space 00’E000h to 00’E7FFh. The 2K Bytes of internal XRAM1 are disabled. ’1’: Accesses to the internal 2K Bytes of XRAM1.
XRAM2EN	**XRAM2 Enable Bit** ‘0’: Accesses to the external memory within space 00’C000h to 00’DFFFh. The 8K Bytes of internal XRAM2 are disabled. ’1’: Accesses to the internal 8K Bytes of XRAM2.
RTCEN	**RTC Enable Bit** ’0’: Accesses to the on-chip Real Time Clock are disabled, external access is performed. Address range 00’EC00h-00’ECFFh is only directed to external memory if CAN1EN and CAN2EN are ’0’ also ’1’: The on-chip Real Time Clock is enabled and can be accessed.

When both CAN are disabled via XPERCON setting, then any access in the address range 00’EE00h - 00’EFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register. P4.4 and P4.7 can be used as General Purpose I/O when CAN2 is disabled, and P4.5 and P4.6 can be used as General Purpose I/O when CAN1 is disabled.

The default XPER selection after Reset is identical to XBUS configuration of ST10C167: XCAN1 is enabled, XCAN2 is disabled, XRAM1 (2 Kbyte compatible XRAM) is enabled, XRAM2 (new 8 Kbyte XRAM) is disabled.

Register XPERCON cannot be changed after the global enabling of XPeripherals, i.e. after the setting of bit XPEN in the SYSCON register.

In EMUlation mode, all the XPERipherals are enabled (XPERCON bit are all set). The access to external memory and/or XBus is controlled by the bondout chip.

When the Real Time Clock is disabled (RTCEN = 0), the clock oscillator is switch-off if the ST10 enters in power-down mode. Otherwise, when the Real Time Clock is enabled, the bit RTCOFF of the RTCCON register allows to choose the power-down mode of the clock oscillator (See *Section 16: Real Time Clock*).

5 Internal Flash Memory

5.1 Overview

- 128-K or 256-Kbyte on-chip Flash memory
- Two possibilities of Flash mapping into the CPU address space
- Flash memory can be used for code and data storage
- 32-bit, zero waitstate read access (50 ns cycle time at f_{CPU} = 40 MHz and 62.5 ns cycle time at f_{CPU} = 32 MHz
- Erase-Program Controller (EPC) similar to M29F400B ST stand-alone Flash memory
  - Word-by-Word Programmable (16 ms typical)
  - Data polling and Toggle Protocol for EPC Status
  - Ready/Busy signal connected on XP2INT interrupt line
  - Internal Power-On detection circuit
- Memory Erase in blocks
  - One 16-Kbyte, two 8-Kbyte, one 32-Kbyte, one to three 64-Kbyte blocks
  - Each block can be erased separately (1.5 second typical)
  - Chip erase (8.5 second typical)
  - Each block can be separately protected against programming and erasing
  - Each protected block can be temporary unprotected
  - When enabled, the read protection prevents access to data in Flash memory using a program running out of the Flash memory space. Access to data of internal Flash can only be performed with an inner protected program
- Erase Suspend and Resume Modes
  - Read and Program another Block during erase suspend
- Single Voltage operation, no need of dedicated supply pin
- Low Power Consumption:
  - 45 mA max. Read current
  - 60 mA max. Program or Erase current
  - Automatic Stand-by-mode (50 mA maximum)
- 1000 Erase-Program Cycles per block, 20 years of data retention time
- Operating temperature: -40 to +125°C

5.2 Operational Overview

5.2.1 Read Mode

In standard mode (the normal operating mode) the Flash appears like an on-chip ROM with the same timing and functionality. The Flash module offers a fast access time, allowing zero waitstate access with CPU frequency up to 40 MHz. Instruction fetches and data operand reads are performed with all addressing modes of the ST10F269Z1-ST10F269Z2 instruction set.

In order to optimize the programming time of the internal Flash, blocks of 8 Kbytes, 16 Kbytes, 32 Kbytes, 64 Kbytes can be used. But the size of the blocks does not apply to the whole memory space, see details in *Table 2*.

Table 2. 128-Kbyte or 256-Kbyte Flash Memory Block Organization

Block	Addresses (Segment 0)	Addresses (Segment 1)	Size (Kbyte)
0	00'0000h to 00'3FFFh	01'0000h to 01'3FFFh	16
1	00'4000h to 00'5FFFh	01'4000h to 01'5FFFh	8
2	00'6000h to 00'7FFFh	01'6000h to 01'7FFFh	8
3	01'8000h to 01'FFFFh	01'8000h to 01'FFFFh	32
4	02'0000h to 02'FFFFh	02'0000h to 02'FFFFh	64
5(1)	03'0000h to 03'FFFFh(1)	03'0000h to 03'FFFFh(1)	64(1)
6(1)	04'0000h to 04'FFFFh(1)	04'0000h to 04'FFFFh(1)	64(1)

1. Not available on 128K versions (reserved areas).

5.2.2 Instructions and Commands

All operations besides normal read operations are initiated and controlled by command sequences written to the Flash Command Interface (CI). The Command Interface (CI) interprets words written to the Flash memory and enables one of the following operations:

- Read memory array
- Program Word
- Block Erase
- Chip Erase
- Erase Suspend
- Erase Resume
- Block Protection
- Block Temporary Unprotection
- Code Protection

Commands are composed of several write cycles at specific addresses of the Flash memory. The different write cycles of such command sequences offer a fail-safe feature to protect against an inadvertent write.

A command only starts when the Command Interface has decoded the last write cycle of an operation. Until that last write is performed, Flash memory remains in Read Mode

Note: As it is not possible to perform write operations in the Flash while fetching code from Flash, the Flash commands must be written by instructions executed from internal RAM or external memory.
Command write cycles do not need to be consecutively received, pauses are allowed, save for Block Erase command. During this operation all Erase Confirm commands must be sent to complete any block erase operation before time-out period expires (typically 96 ms).
Command sequencing must be followed exactly. Any invalid combination of commands will reset the Command Interface to Read Mode.

5.2.3 Status Register

This register is used to flag the status of the memory and the result of an operation. This register can be accessed by read cycles during the Erase-Program Controller (EPC) operation.

5.2.4 Erase Operation

This Flash memory features a block erase architecture with a chip erase capability too. Erase is accomplished by executing the six cycle erase command sequence. Additional command write cycles can then be performed to erase more than one block in parallel. When a time-out period elapses (96 ms) after the last cycle, the Erase-Program Controller (EPC) automatically starts and times the erase pulse and executes the erase operation. There is no need to program the block to be erased with ‘0000h’ before an erase operation. Termination of operation is indicated in the Flash status register. After erase operation, the Flash memory locations are read as 'FFFFh’ value.

5.2.5 Erase Suspend

A block erase operation is typically executed within 1.5 second for a 64-Kbyte block. Erasure of a memory block may be suspended, in order to read data from another block or to program data in another block, and then resumed.

5.2.6 In-System Programming

In-system programming is fully supported. No special programming voltage is required. Because of the automatic execution of erase and programming algorithms, write operations are reduced to transferring commands and data to the Flash and reading the status. Any code that programs or erases Flash memory locations (that writes data to the Flash) must be executed from memory outside the on-chip Flash memory itself (on-chip RAM or external memory).

A boot mechanism is provided to support in-system programming. It works using serial link via USART interface and a PC compatible or other programming host.

5.2.7 Read/Write Protection

The Flash module supports read and write protection in a very comfortable and advanced protection functionality. If Read Protection is installed, the whole Flash memory is protected against any "external" read access; read accesses are only possible with instructions fetched directly from program Flash memory. For update of the Flash memory a temporary disable of Flash Read Protection is supported.

The device also features a block write protection. Software locking of selectable memory blocks is provided to protect code and data. This feature will disable both program and erase operations in the selected block(s) of the memory. Block Protection is accomplished by block specific lock-bit which are programmed by executing a four cycle command sequence. The locked state of blocks is indicated by specific flags in the according block status registers. A block may only be temporarily unlocked for update (write) operations.

With the two possibilities for write protection - whole memory or block specific - a flexible installation of write protection is supported to protect the Flash memory or parts of it from unauthorized programming or erase accesses and to provide virus-proof protection for all system code blocks. All write protection also is enabled during boot operation.

5.2.8 Power Supply, Reset

The Flash module uses a single power supply for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations from 5 V supply. Once a program or erase cycle has been completed, the device resets to the

standard read mode. At power-on, the Flash memory has a setup phase of some microseconds (dependent on the power supply ramp-up). During this phase, Flash can not be read. Thus, if $\overline{EA}$ pin is high (execution will start from Flash memory), the CPU remains in reset state until the Flash can be accessed.

5.3 Architectural Description

The Flash module distinguishes two basic operating modes, the standard read mode and the command mode. The initial state after power-on and after reset is the standard read mode.

5.3.1 Read Mode

The Flash module enters the standard operating mode, the read mode:

- After Reset command
- After every completed erase operation
- After every completed programming operation
- After every other completed command execution
- Few microseconds after a CPU-reset has started
- After incorrect address and data values of command sequences or writing them in an improper sequence
- After incorrect write access to a read protected Flash memory

The read mode remains active until the last command of a command sequence is decoded which starts directly a Flash array operation, such as:

- erase one or several blocks
- program a word into Flash array
- protect / temporary unprotect a block.

In the standard read mode read accesses are directly controlled by the Flash memory array, delivering a 32-bit double Word from the addressed position. Read accesses are always aligned to double Word boundaries. Thus, both low order address bit A1 and A0 are not used in the Flash array for read accesses. The high order address bit A17/A16 define the physical 64-Kbyte segment being accessed within the Flash array.

5.3.2 Command Mode

Every operation besides standard read operations is initiated by commands written to the Flash command register. The addresses used for command cycles define in conjunction with the actual state the specific step within command sequences. With the last command of a command sequence, the Erase-Program Controller (EPC) starts the execution of the command. The EPC status is indicated during command execution by:

- The Status Register,
- The Ready/Busy signal.

5.3.3 Ready/Busy Signal

The Ready/Busy ($\overline{R}$/B) signal is connected to the XPER2 interrupt node (XP2IC). When $\overline{R}$/B is high, the Flash is busy with a Program or Erase operation and will not accept any additional program or erase instruction. When $\overline{R}$/B is Low, the Flash is ready for any

Read/Write or Erase operation. The $\overline{R}$/B will also be low when the memory is put in Erase Suspend mode.

This signal can be polled by reading XP2IC register, or can be used to trigger an interrupt when the Flash goes from Busy to Ready.

5.3.4 Flash Status Register

The Flash Status register is used to flag the status of the Flash memory and the result of an operation. This register can be accessed by Read cycles during the program-Erase Controller operations. The program or erase operation can be controlled by data polling on bit FSB.7 of Status Register, detection of Toggle on FSB.6 and FSB.2, or Error on FSB.5 and Erase Time-out on FSB.3 bit. Any read attempt in Flash during EPC operation will automatically output these five bits. The EPC sets bit FSB.2, FSB.3, FSB.5, FSB.6 and FSB.7. Other bits are reserved for future use and should be masked.

Flash Status

Note: The Address of Flash Status Register is the address of the word being programmed when Programming operation is in progress, or an address within block being erased when Erasing operation is in progress.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	FSB.7	FSB.6	FSB.5	-	FSB.3	FSB.2	-	-
								R	R	R		R	R		

FSB.7	**Flash Status bit 7: Data Polling Bit** Programming Operation: this bit outputs the complement of the bit 7 of the word being programmed, and after completion, will output the bit 7 of the word programmed. Erasing Operation: outputs a ‘0’ during erasing, and ‘1’ after erasing completion. If the block selected for erasure is (are) protected, FSB.7 will be set to ‘0’ for about 100 µs, and then return to the previous addressed memory data value. FSB.7 will also flag the Erase Suspend Mode by switching from ‘0’ to ‘1’ at the start of the Erase Suspend. During Program operation in Erase Suspend Mode, FSB.7 will have the same behavior as in normal Program execution outside the Suspend mode.
FSB.6	**Flash Status bit 6: Toggle Bit** Programming or Erasing Operations: successive read operations of Flash Status register will deliver complementary values. FSB.6 will toggle each time the Flash Status register is read. The Program operation is completed when two successive reads yield the same value. The next read will output the bit last programmed, or a ‘1’ after Erase operation FSB.6 will be set to‘1’ if a read operation is attempted on an Erase Suspended block. In addition, an Erase Suspend/Resume command will cause FSB.6 to toggle.
FSB.5	**Flash Status bit 5: Error Bit** This bit is set to ‘1’ when there is a failure of Program, block or chip erase operations.This bit will also be set if a user tries to program a bit to ‘1’ to a Flash location that is currently programmed with ‘0’. The error bit resets after Read/Reset instruction. In case of success, the Error bit will be set to ‘0’ during Program or Erase and then will output the bit last programmed or a ‘1’ after erasing
FSB.3	**Flash Status bit 3: Erase Time-out Bit** This bit is cleared by the EPC when the last Block Erase command has been entered to the Command Interface and it is awaiting the Erase start. When the time-out period is finished, after 96 µs, FSB.3 returns back to ‘1’.

FSB.2	**Flash Status bit 2: Toggle Bit** This toggle bit, together with FSB.6, can be used to determine the chip status during the Erase Mode or Erase Suspend Mode. It can be used also to identify the block being Erased Suspended. A Read operation will cause FSB.2 to Toggle during the Erase Mode. If the Flash is in Erase Suspend Mode, a Read operation from the Erase suspended block or a Program operation into the Erase suspended block will cause FSB.2 to toggle. When the Flash is in Program Mode during Erase Suspend, FSB.2 will be read as '1' if address used is the address of the word being programmed. After Erase completion with an Error status, FSB.2 will toggle when reading the faulty sector.

5.3.5 Flash Protection Register

The Flash Protection register is a non-volatile register that contains the protection status. This register can be read by using the Read Protection Status (RP) command, and programmed by using the dedicated Set Protection command.

Flash Protection Register (PR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP	-	-	-	-	-	-	-	-	BP6 (1)	BP5 (1)	BP4	BP3	BP2	BP1	BP0

1. Not available for 128K versions (reserved areas)

BPx	**Block x Protection Bit (x = 0...6)** '0': the Block Protection is enabled for block x. Programming or erasing the block is not possible, unless a Block Temporary Unprotection command is issued. 1': the Block Protection is disabled for block x. Bit is '1' by default, and can be programmed permanently to '0' using the Set Protection command but then cannot be set to '1' again. It is therefore possible to temporally disable the Block Protection using the Block Temporary Unprotection instruction.
CP	**Code Protection Bit** '0': the Flash Code Protection is enabled. Read accesses to the Flash for execution not performed in the Flash itself are not allowed, the returned value will be 009Bh, whatever the content of the Flash is. 1': the Flash Code Protection is disabled: read accesses to the Flash from external or internal RAM are allowed Bit is '1' by default, and can be programmed permanently to '0' using the Set Protection command but then cannot be set to '1' again. It is therefore possible to temporally disable the Code Protection using the Code Temporary Unprotection instruction.

5.3.6 Instructions Description

Twelve instructions dedicated to Flash memory accesses are defined as follows:

- **Read/Reset (RD)**. The Read/Reset instruction consist of one write cycle with data XXF0h. it can be optionally preceded by two CI enable coded cycles (data xxA8h at address 1554h + data xx54h at address 2AA8h). Any successive read cycle following a

Read/Reset instruction will read the memory array. A Wait cycle of 10 µs is necessary after a Read/Reset command if the memory was in program or Erase mode.

- **Program Word (PW)**. This instruction uses four write cycles. After the two CI enable coded cycles, the Program Word command xxA0h is written at address 1554h. The following write cycle will latch the address and data of the word to be programmed. Memory programming can be done only by writing 0's instead of 1's, otherwise an error occurs. During programming, the Flash Status is checked by reading the Flash Status bit FSB.2, FSB.5, FSB.6 and FSB.7 which show the status of the EPC. FSB.2, FSB.6 and FSB.7 determine if programming is on going or has completed, and FSB.5 allows a check to be made for any possible error.
- **Block Erase (BE)**. This instruction uses a minimum of six command cycles. The erase enable command xx80h is written at address 1554h after the two-cycle CI enable sequence.

  The erase confirm code xx30h must be written at an address related to the block to be erased preceded by the execution of a second CI enable sequence. Additional erase confirm codes must be given to erase more than one block in parallel. Additional erase confirm commands must be written within a defined time-out period. The input of a new Block Erase command will restart the time-out period.

  When this time-out period has elapsed, the erase starts. The status of the internal timer can be monitored through the level of FSB.3, if FSB.3 is ‘0’, the Block Erase command has been given and the time-out is running; if FSB.3 is ‘1’, the time-out has expired and the EPC is erasing the block(s).

  If the second command given is not an erase confirm or if the coded cycles are wrong, the instruction aborts, and the device is reset to Read Mode. It is not necessary to program the block with 0000h as the EPC will do this automatically before the erasing to FFFFh. Read operations after the EPC has started, output the Flash Status Register.

  During the execution of the erase by the EPC, the device accepts only the Erase Suspend and Read/Reset instructions. Data Polling bit FSB.7 returns ‘0’ while the erasure is in progress, and ‘1’ when it has completed. The Toggle bit FSB.2 and FSB.6 toggle during the erase operation. They stop when erase is completed. After completion, the Error bit FSB.5 returns ‘1’ if there has been an erase failure because erasure has not completed even after the maximum number of erase cycles have been executed by the EPC, in this case, it will be necessary to input a Read/Reset to the Command Interface in order to reset the EPC.
- **Chip Erase (CE)**. This instruction uses six write cycles. The Erase Enable command xx80h, must be written at address 1554h after CI-Enable cycles. The Chip Erase command xx10h must be given on the sixth cycle after a second CI-Enable sequence. An error in command sequence will reset the CI to Read mode. It is NOT necessary to program the block with 0000h as the EPC will do this automatically before the erasing to FFFFh. Read operations after the EPC has started output the Flash Status Register. During the execution of the erase by the EPC, Data Polling bit FSB.7 returns ‘0’ while the erasure is in progress, and ‘1’ when it has completed. The FSB.2 and FSB.6 bit toggle during the erase operation. They stop when erase is finished. The FSB.5 error bit returns "1" in case of failure of the erase operation. The error flag is set after the maximum number of erase cycles have been executed by the EPC. In this case, it will be necessary to input a Read/Reset to the Command Interface in order to reset the EPC.
- **Erase Suspend (ES)**. This instruction can be used to suspend a Block Erase operation by giving the command xxB0h without any specific address. No CI-Enable cycles is required. Erase Suspend operation allows reading of data from another block and/or

the programming in another block while erase is in progress. If this command is given during the time-out period, it will terminate the time-out period in addition to erase Suspend. The Toggle bit FSB.6, when monitored at an address that belongs to the block being erased, stops toggling when Erase Suspend Command is effective, It happens between 0.1 ms and 15 ms after the Erase Suspend Command has been written. The Flash will then go in normal Read Mode, and read from blocks not being erased is valid, while read from block being erased will output FSB.2 toggling. During a Suspend phase the only instructions valid are Erase Resume and Program Word. A Read / Reset instruction during Erase suspend will definitely abort the Erase and result in invalid data in the block being erased.

- **Erase Resume (ER)**. This instruction can be given when the memory is in Erase Suspend State. Erase can be resumed by writing the command xx30h at any address without any CI-enable sequence.
- **Program during Erase Suspend**. The Program Word instruction during Erase Suspend is allowed only on blocks that are not Erase-suspended. This instruction is the same than the Program Word instruction.
- **Set Protection (SP)**. This instruction can be used to enable both Block Protection (to protect each block independently from accidental Erasing-Programming Operation) and Code Protection (to avoid code dump). The Set Protection Command must be given after a special CI-Protection Enable cycles (see instruction table). The following Write cycle, will program the Protection Register. To protect the block x (x = 0 to 6), the data bit x must be at '0'. To protect the code, bit 15 of the data must be '0'. Enabling Block or Code Protection is permanent and can be cleared only by ST. Block Temporary Unprotection and Code Temporary Unprotection instructions are available to allow the customer to update the code.

Note: The new value programmed in protection register will only become active after a reset. Bit that are already at '0' in protection register must be confirmed at '0' also in data latched during the 4th cycle of set protection command, otherwise an error may occur.

- **Read Protection Status (RP)**. This instruction is used to read the Block Protection status and the Code Protection status. To read the protection register (see *Table 3*), the CI-Protection Enable cycles must be executed followed by the command xx90h at address x2A54h. The following Read Cycles at any odd word address will output the Block Protection Status. The Read/Reset command xxF0h must be written to reset the protection interface.

Note: After a modification of protection register (using Set Protection command), the Read Protection Status will return the new PR value only after a reset.

- **Block Temporary Unprotection (BTU)**. This Instruction can be used to temporary unprotect all the blocks from Program / Erase protection. The Unprotection is disabled after a Reset cycle. The Block Temporary Unprotection command xxC1h must be given to enable Block Temporary Unprotection. The Command must be preceded by the CI-Protection Enable cycles and followed by the Read/Reset command xxF0h.
- **Set Code Protection (SCP)**. This kind of protection allows the customer to protect the proprietary code written in Flash. If installed and active, Flash Code Protection prevents data operand accesses and program branches into the on-chip Flash area from any location outside the Flash memory itself. Data operand accesses and branches to Flash locations are only and exclusively allowed for instructions executed from the Flash memory itself. Every read or jump to Flash performed from another memory (like internal RAM, external memory) while Code Protection is enabled, will give the opcode 009Bh related to TRAP #00 illegal instruction. The CI-Protection Enable cycles must be sent to set the Code Protection. By writing data 7FFFh at any

odd word address, the Code Protected status is stored in the Flash Protection Register (PR). Protection is permanent and cannot be cleared by the user. It is possible to temporarily disable the Code Protection using Code Temporary Unprotection instruction.

Note: Bits that are already at '0' in protection register must be confirmed at '0' also in data latched during the 4th cycle of set protection command, otherwise an error may occur.

- **Code Temporary Unprotection (CTU)**. This instruction must be used to temporary disable Code Protection. This instruction is effective only if executed from Flash memory space. To restore the protection status, without using a reset, it is necessary to use a Code Temporary Protection instruction. System reset will reset also the Code Temporary Unprotected status. The Code Temporary Unprotection command consists of the following write cycle:

```
MOVMEM, Rn;This instruction MUST be executed from Flash memory space.
```

Where MEM is an absolute address inside memory space, Rn is a register loaded with data 0FFFFh.

- **Code Temporary Protection (CTP).** This instruction allows to restore Code Protection. This operation is effective only if executed from Flash memory and is necessary to restore the protection status after the use of a Code Temporary Unprotection instruction.

  The Code Temporary Protection command consists of the following write cycle:

```
MOVMEM, Rn ;This instruction MUST be executed from Flash memory space
```

Where MEM is an absolute address inside memory space, Rn is a register loaded with data 0FFFBh.

Note that Code Temporary Unprotection instruction must be used when it is necessary to modify the Flash with protected code (SCP), since the write/erase routines must be executed from a memory external to Flash space. Usually, the write/erase routines, executed in RAM, ends with a return to Flash space where a CTP instruction restore the protection.

Table 3. Instructions

Instruction	Mne	Cycle		1st Cycle	2nd Cycle	3rd Cycle	4th Cycle	5th Cycle	6th Cycle	7th Cycle
Read/Reset	RD	1+	Addr.[1]	X [2]	Read Memory Array until a new write cycle is initiated					
			Data	xxF0h						
Read/Reset	RD	3+	Addr.[1]	x1554h	x2AA8h	xxxxxh	Read Memory Array until a new write cycle is initiated			
			Data	xxA8h	xx54h	xxF0h				
Program Word	PW	4	Addr.[1]	x1554h	x2AA8h	x1554h	WA [3]	Read Data Polling or Toggle bit until Program completes.		
			Data	xxA8h	xx54h	xxA0h	WD [4]			
Block Erase	BE	6	Addr.[1]	x1554h	x2AA8h	x1554h	x1554h	x2AA8h	BA	BA' [5]
			Data	xxA8h	xx54h	xx80h	xxA8h	xx54h	xx30h	xx30h
Chip Erase	CE	6	Addr.[1]	x1554h	x2AA8h	x1554h	x1554h	x2AA8h	x1554h	Note [6]
			Data	xxA8h	xx54h	xx80h	xxA8h	xx54h	xx10h	
Erase Suspend	ES	1	Addr.[1]	X[2]	Read until Toggle stops, then read or program all data needed from block(s) not being erased then Resume Erase.					
			Data	xxB0h						
Erase Resume	ER	1	Addr.[1]	X[2]	Read Data Polling or Toggle bit until Erase completes or Erase is suspended another time.					
			Data	xx30h						

Table 3. Instructions (continued)

Instruction	Mne	Cycle		1st Cycle	2nd Cycle	3rd Cycle	4th Cycle	5th Cycle	6th Cycle	7th Cycle
Set Block/Code Protection	SP	4	Addr.[1]	x2A54h	x15A8h	x2A54h	Any odd word address [9]	-		
			Data	xxA8h	xx54h	xxC0h	WPR [7]			
Read Protection Status	RP	4	Addr.[1]	x2A54h	x15A8h	x2A54h	Any odd word address [9]	Read Protection Register until a new write cycle is initiated.		
			Data	xxA8h	xx54h	xx90h	Read PR			
Block Temporary Unprotection	BTU	4	Addr.[1]	x2A54h	x15A8h	x2A54h	X[2]	-		
			Data	xxA8h	xx54h	xxC1h	xxF0h			
Code Temporary Unprotection	CTU	1	Addr.[1]	MEM [8]	Write cycles must be executed from Flash.					
			Data	FFFFh						
Code Temporary Protection	CTP	1	Addr.[1]	MEM [8]	Write cycles must be executed from Flash.					
			Data	FFFBh						

1. Address bit A14, A15 and above are don't care for coded address inputs.
2. X = Don't Care.
3. WA = Write Address: address of memory location to be programmed.
4. WD = Write Data: 16-bit data to be programmed.
5. Optional, additional blocks addresses must be entered within a time-out delay (96 µs) after last write entry, time-out status can be verified through FSB.3 value. When full command is entered, read Data Polling or Toggle bit until Erase is completed or suspended.
6. Read Data Polling or Toggle bit until Erase completes.
7. WPR = Write protection register. To protect code, bit 15 of WPR must be '0'. To protect block N (N=0,1,...), bit N of WPR must be '0'. Bit that are already at '0' in protection register must also be '0' in WPR, else a writing error will occurs (it is not possible to write a '1' in a bit already programmed at '0').
8. MEM = any address inside the Flash memory space. Absolute addressing mode must be used (MOV MEM, Rn), and instruction must be executed from Flash memory space.
9. Odd word address = 4n-2 where n = 0, 1, 2, 3..., ex. 0002h, 0006h...

Generally, command sequences cannot be written to Flash by instructions fetched from the Flash itself. Thus, the Flash commands must be written by instructions, executed from internal RAM or external memory.

Command cycles on the CPU interface need not to be consecutively received (pauses allowed). The CPU interface delivers dummy read data for not used cycles within command sequences.

All addresses of command cycles shall be defined only with Register-indirect addressing mode in the according move instructions. Direct addressing is not allowed for command sequences. Address segment or data page pointer are taken into account for the command address value.

5.3.7 Reset Processing and Initial State

The Flash module distinguishes two kinds of CPU reset types

The lengthening of CPU reset:

- is not reported to external devices by bidirectional pin
- is not enabled in case of external start of CPU after reset.

5.4 Flash Memory Configuration

The default memory configuration of the ST10F269Z1-ST10F269Z2 Memory is determined by the state of the $\overline{EA}$ pin at reset. This value is stored in the Internal ROM Enable bit (named ROMEN) of the SYSCON register.

When ROMEN = 0, the internal Flash is disabled and external ROM is used for startup control. Flash memory can later be enabled by setting the ROMEN bit of SYSCON to 1. The code performing this setting must not run from a segment of the external ROM to be replaced by a segment of the Flash memory, otherwise unexpected behavior may occur.

For example, if external ROM code is located in the first 32 Kbytes of segment 0, the first 32 Kbytes of the Flash must then be enabled in segment 1. This is done by setting the ROMS1 bit of SYSCON to 0 before or simultaneously with setting of ROMEN bit. This must be done in the externally supplied program before the execution of the EINIT instruction.

If program execution starts from external memory, but access to the Flash memory mapped in segment 0 is later required, then the code that performs the setting of ROMEN bit must be executed either in the segment 0 but above address 00’8000h, or from the internal RAM.

Bit ROMS1 only affects the mapping of the first 32 Kbytes of the Flash memory. All other parts of the Flash memory (addresses 01’8000h - 04’FFFFh) remain unaffected.

The SGTDIS Segmentation Disable / Enable must also be set to 0 to allow the use of the full 256 Kbytes of on-chip memory in addition to the external boot memory. The correct procedure on changing the segmentation registers must also be observed to prevent an unwanted trap condition:

- Instructions that configure the internal memory must only be executed from external memory or from the internal RAM.
- An Absolute Inter-Segment Jump (JMPS) instruction must be executed after Flash enabling, to the next instruction, even if this next instruction is located in the consecutive address.
- Whenever the internal Memory is disabled, enabled or remapped, the DPPs must be explicitly (re)loaded to enable correct data accesses to the internal memory and/or external memory.

5.5 Application Examples

5.5.1 Handling of Flash Addresses

All command, Block, Data and register addresses to the Flash have to be located within the active Flash memory space. The active space is that address range to which the physical Flash addresses are mapped as defined by the user. When using data page pointer (DPP)

for block addresses make sure that address bit A15 and A14 of the block address are reflected in both LSBs of the selected DPPS.

Note: For Command Instructions, address bit A14, A15, A16 and A17 are don't care. This simplify a lot the application software, because it minimize the use of DPP registers when using Command in the Command Interface.
Direct addressing is not allowed for Command sequence operations to the Flash. Only Register-indirect addressing can be used for command, block or write-data accesses.

5.5.2 Basic Flash Access Control

When accessing the Flash all command write addresses have to be located within the active Flash memory space. The active Flash memory space is that logical address range which is covered by the Flash after mapping. When using data page pointer (DPP) for addressing the Flash, make sure that address bit A15 and A14 of the command addresses are reflected in both LSBs of the selected data page pointer (A15 - DPPx.1 and A14 - DPPx.0).

In case of the command write addresses, address bit A14, A15 and above are don't care. Thus, command writes can be performed by only using one DPP register. This allow to have a more simple and compact application software.

Another - advantageous - possibility is to use the extended segment instruction for addressing.

Note: The direct addressing mode is not allowed for write access to the Flash address/command register. Be aware that the C compiler may use this kind of addressing. For write accesses to Flash module always the indirect addressing mode has to be selected.

The following basic instruction sequences show examples for different addressing possibilities.

Principle example of address generation for Flash commands and registers:

When using data page pointer (DPP0 is this example)

```
MOVDPP0,#08h;adjust data page pointers according to the
;addresses: DPP0 is used in this example, thus
;ADDRESS must have A14 and A15 bit set to '0'.
MOVRwm,#ADDRESS;ADDRESS could be a dedicated command sequence
;address 2AA8h, 1554h ... ) or the Flash write
;address
MOVRwn,#DATA;DATA could be a dedicated command sequence data
;(xxA0h,xx80h ... ) or data to be programmed
MOV[Rwm],Rwn;indirect addressing
```

When using the extended segment instruction:

```
MOVRwm,#ADDRESS;ADDRESS could be a dedicated command sequence
;address (2AA8h, 1554h ... ) or the Flash write
;address
MOVRwo,#DATA;DATA could be a dedicated command sequence data
;(xxA0h,xx80h ... ) or data to be programmed
MOVRwn,#SEGMENT ;the value of SEGMENT represents the segment
;number and could be 0, 1, 2, 3 or 4 (depending
;on sector mapping) for 256KByte Flash.
```

```
EXTSRwn,#LENGTH;the value of Rwn determines the 8-bit segment
;valid for the corresponding data access for any
;long or indirect address in the following(s)
;instruction(s). LENGTH defines the number of
;the effected instruction(s) and has to be a value ;between 1...4
MOV[Rwm],Rwo;indirect addressing with segment number from
;EXTS
```

5.5.3 Programming Examples

Most of the microcontroller programs are written in the C language where the data page pointers are automatically set by the compiler. But because the C compiler may use the not allowed direct addressing mode for Flash write addresses, it is necessary to program the organizational Flash accesses (command sequences) with assembler in-line routines which use indirect addressing.

Example 1 Performing the command Read/Reset

We assume that in the initialization phase the lowest 32 Kbytes of Flash memory (sector 0) have been mapped to segment 1.

According to the usual way of ST10 data addressing with data page pointers, address bit A15 and A14 of a 16-bit command write address select the data page pointer (DPP) which contains the upper 10-bit for building the 24-bit physical data address. Address bit A13...A0 represent the address offset. As the bit A14...A17 are "don't care" when written a Flash command in the Command Interface (CI), we can choose the most convenient DPPx register for address handling.

The following examples are making usage of DPP0. We just have to make sure, that DPP0 points to active Flash memory space.

To be independent of mapping of sector 0 we choose for all DPPs which are used for Flash address handling, to point to segment 2.

For this reason we load DPP0 with value 08h (00 0000 l000b).

```
MOVR5, #01554h;load auxilary register R5 with command address ;(used in
command cycle 1)
MOVR6, #02AA8h;load auxilary register R6 with command address ;(used in
command cycle 2)
SCXTDPPO, #08h;push data page pointer 0 and load it to point to ;segment 2
MOVR7, #0A8h;load register R7 with 1st CI enable command
MOV[R5], R7;command cycle 1
MOVR7, #054h;load register R7 with 2cd CI enable command
MOV[R6], R7;command cycle 2
MOVR7, #0F0h;load register R7 with Read/Reset command
MOV[R5], R7;command cycle 3. Address is don't care
POPDPP0;restore DPP0 value
```

In the example above the 16-bit registers R5 and R6 are used as auxiliary registers for indirect addressing.

Example 2 Performing a Program Word command

We assume that in the initialization phase the lowest 32 Kbytes of Flash memory (sector 0) have been mapped to segment 1.The data to be written is loaded in register R13, the

address to be programmed is loaded in register R11/R12 (segment number in R11, segment offset in R12).

```
MOVR5, #01554h;load auxilary register R5 with command address ;(used in
command cycle 1)
MOVR6, #02AA8h;load auxilary register R6 with command address ;(used in
command cycle 2)
SXCTDPPO, #08h;push data page pointer 0 and load it to point to ;segment 2
MOVR7, #0A8h;load register R7 with 1st CI enable command
MOV[R5], R7;command cycle 1
MOVR7, #054h;load register R7 with 2cd CI enable command
MOV[R6], R7;command cycle 2
MOVR7, #0A0h;load register R7 with Program Word command
MOV[R5], R7;command cycle 3
POPDPP0;restore DPP0: following addressing to the Flash ;will use EXTended
instructions
     ;R11 contains the segment to be programmed
     ;R12 contains the segment offset address to be ;programmed
     ;R13 contains the data to be programmed
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOV[R12], R13;command cycle 4: the EPC starts execution of ;Programming
Command
Data_Polling:
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOVR7, [R12];read Flash Status register (FSB) in R7
MOVR6, R7;save it in R6 register
     ;Check if FSB.7 = Data.7 (i.e. R7.7 = R13.7)
XORR7, R13
JNBR7.7, Prog_OK

     ;Check if FSB.5 = 1 (Programming Error)
JNBR6.5, Data_Polling
     ;Programming Error: verify is Flash programmed ;data is OK
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOVR7, [R12];read Flash Status register (FSB) in R7
     ;Check if FSB.7 = Data.7
XORR7, R13
JNBR7.7, Prog_OK
     ;Programming failed: Flash remains in Write ;Operation.
     ;To go back to normal Read operations, a Read/Reset ;command
     ;must be performed
Prog_Error:
MOVR7, #0F0h;load register R7 with Read/Reset command
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOV[R12], R7;address is don't care for Read/Reset command
...  ;here place specific Error handling code
...
```

```
...
     ;When programming operation finished succesfully, ;Flash is set back
automatically to normal Read Mode
Prog_OK:
....
....
```

Example 3 Performing the Block Erase command

We assume that in the initialization phase the lowest 32 Kbytes of Flash memory (sector 0) have been mapped to segment 1.The registers R11/R12 contain an address related to the block to be erased (segment number in R11, segment offset in R12, for example R11 = 01h, R12 = 4000h will erase the block 1 - first 8-Kbyte block).

```
MOVR5, #01554h;load auxilary register R5 with command address ;(used in
command cycle 1)
MOVR6, #02AA8h;load auxilary register R6 with command address ;(used in
command cycle 2)
SXCTDPPO, #08h;push data page pointer 0 and load it to point ;to ;segment 2
MOVR7, #0A8h;load register R7 with 1st CI enable command
MOV[R5], R7;command cycle 1
MOVR7, #054h;load register R7 with 2cd CI enable command
MOV[R6], R7;command cycle 2
MOVR7, #080h;load register R7 with Block Erase command
MOV[R5], R7;command cycle 3
MOVR7, #0A8h;load register R7 with 1st CI enable command
MOV[R5], R7;command cycle 4
MOVR7, #054h;load register R7 with 2cd CI enable command
MOV[R6], R7;command cycle 5
POPDPP0;restore DPP0: following addressing to the Flash ;will use EXTended
instructions
     ;R11 contains the segment of the block to be erased
     ;R12 contains the segment offset address of the ;block to be erased
MOVR7, #030h;load register R7 with erase confirm code
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOV[R12], R7;command cycle 6: the EPC starts execution of ;Erasing Command
Erase_Polling:
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOVR7, [R12];read Flash Status register (FSB) in R7
     ;Check if FSB.7 = '1' (i.e. R7.7 = '1')
JB R7.7, Erase_OK

     ;Check if FSB.5 = 1 (Erasing Error)
JNBR7.5, Erase_Polling
     ;Programming failed: Flash remains in Write ;Operation.
     ;To go back to normal Read operations, a Read/Reset ;command
     ;must be performed
```

```
Erase_Error:
MOVR7, #0F0h;load register R7 with Read/Reset command
EXTSR11, #1;use EXTended addressing for next MOV instruction
MOV[R12], R7;address is don't care for Read/Reset command
...  ;here place specific Error handling code
...
...

     ;When erasing operation finished succesfully,
;Flash is set back automatically to normal Read Mode
Erase_OK:
....
....
```

5.6 Bootstrap Loader

The built-in bootstrap loader (BSL) of the ST10F269 provides a mechanism to load the startup program through the serial interface after reset. In this case, no external memory or internal Flash memory is required for the initialization code starting at location 00'0000h (see *Figure 5*).

The bootstrap loader moves code/data into the internal RAM, but can also transfer data via the serial interface into an external RAM using a second level loader routine. Flash Memory (internal or external) is not necessary, but it may be used to provide lookup tables or "core-code" like a set of general purpose subroutines for I/O operations, number crunching, system initialization, etc.

The bootstrap loader can be used to load the complete application software into ROMless systems, to load temporary software into complete systems for testing or calibration, or to load a programming routine for Flash devices.

The BSL mechanism can be used for standard system startup as well as for special occasions like system maintenance (firmer update) or end-of-line programming or testing.

5.6.1 Entering the Bootstrap Loader

The ST10F269 enters BSL mode when pin P0L.4 is sampled low at the end of a hardware reset. In this case the built-in bootstrap loader is activated independent of the selected bus mode.

The bootstrap loader code is stored in a special Boot-ROM. No part of the standard mask Memory or Flash Memory area is required for this.

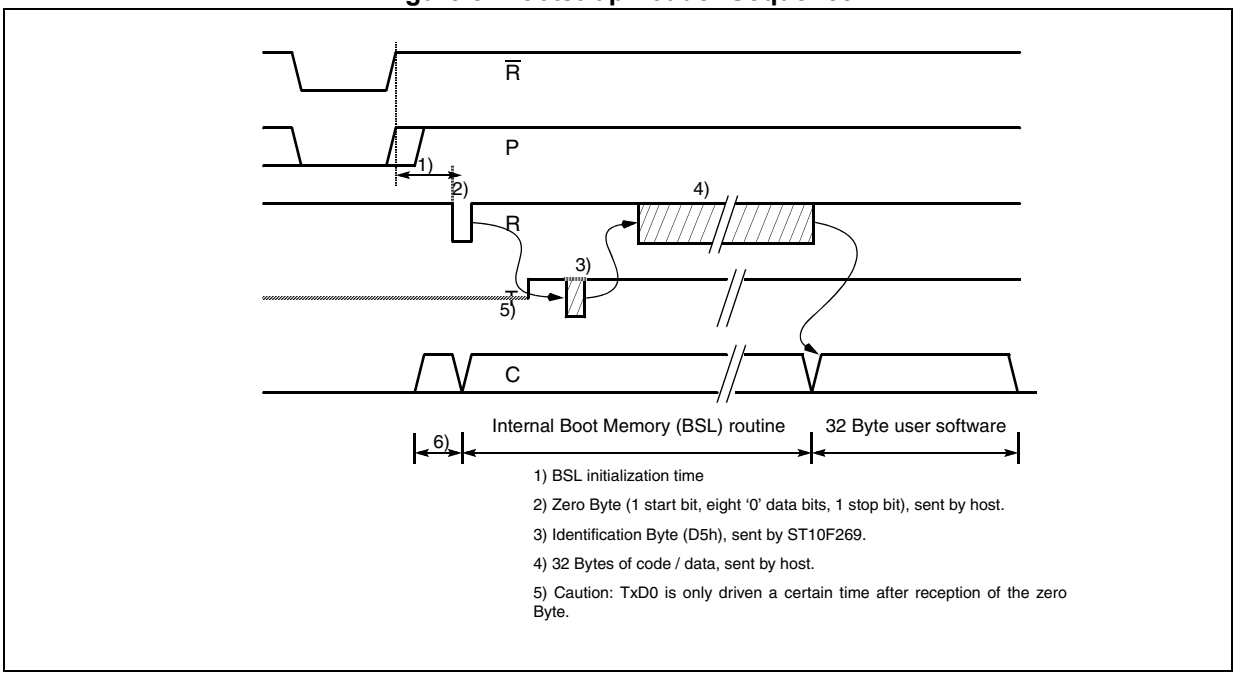
After entering BSL mode and the respective initialization the ST10F269 scans the RXD0 line to receive a zero Byte, one start bit, eight ‘0’ data bits and one stop bit.

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 identification Byte is returned to the host that provides the loaded data.

This identification byte identifies the device to be booted. The identification byte is D5h for ST10F269.

Figure 5. Bootstrap Loader Sequence



When the ST10F269 has entered BSL mode, the following configuration is automatically set (values that deviate from the normal reset values, are marked):

Watchdog Timer:	Disabled	Register SYSCON:	0E00h
Context Pointer CP:	FA00h	Register STKUN:	FA40h
Stack Pointer SP:	FA40h	Register STKOV:	FA0Ch 0<->C
Register S0CON:	8011h	Register BUSCON0:	acc. to startup configuration
Register S0BG:	Acc. to ‘00’ Byte	P3.10 / TXD0:	‘1’
		DP3.10:	‘1’

In this case, the watchdog timer is disabled, so the bootstrap loading sequence is not time limited.

Pin TXD0 is configured as output, so the ST10F269 can return the identification byte.

Even if the internal Flash is enabled, no code can be executed out of it.

The hardware that activates the BSL during reset may be a simple pull-down resistor on P0L.4 for systems that use this feature upon every hardware reset.

A switchable solution (via jumper or an external signal) can be used for systems that only temporarily use the bootstrap loader
(see *Figure 6*).

After sending the identification Byte the ASC0 receiver is enabled and is ready to receive the initial 32 Bytes from the host. A half duplex connection is therefore sufficient to feed the BSL.

5.6.2 Memory Configuration After Reset

The configuration (and the accessibility) of the ST10F269's memory areas after reset in Bootstrap-Loader mode differs from the standard case. Pin EA is not evaluated when BSL mode is selected, and accesses to the internal Flash area are partly redirected, while the ST10F269 is in BSL mode (see *Table 4*). All code fetches are made from the special Boot-ROM, while data accesses read from the internal user Flash. Data accesses will return undefined values on ROMless devices.

The code in the Boot-ROM is not an invariant feature of the ST10F269. User software should not try to execute code from the internal Flash area while the BSL mode is still active, as these fetches will be redirected to the Boot-ROM. The Boot-ROM will also "move" to segment 1, when the internal Flash area is mapped to segment 1 (see *Table 4*).

Figure 6. Hardware Provisions to Activate the BSL

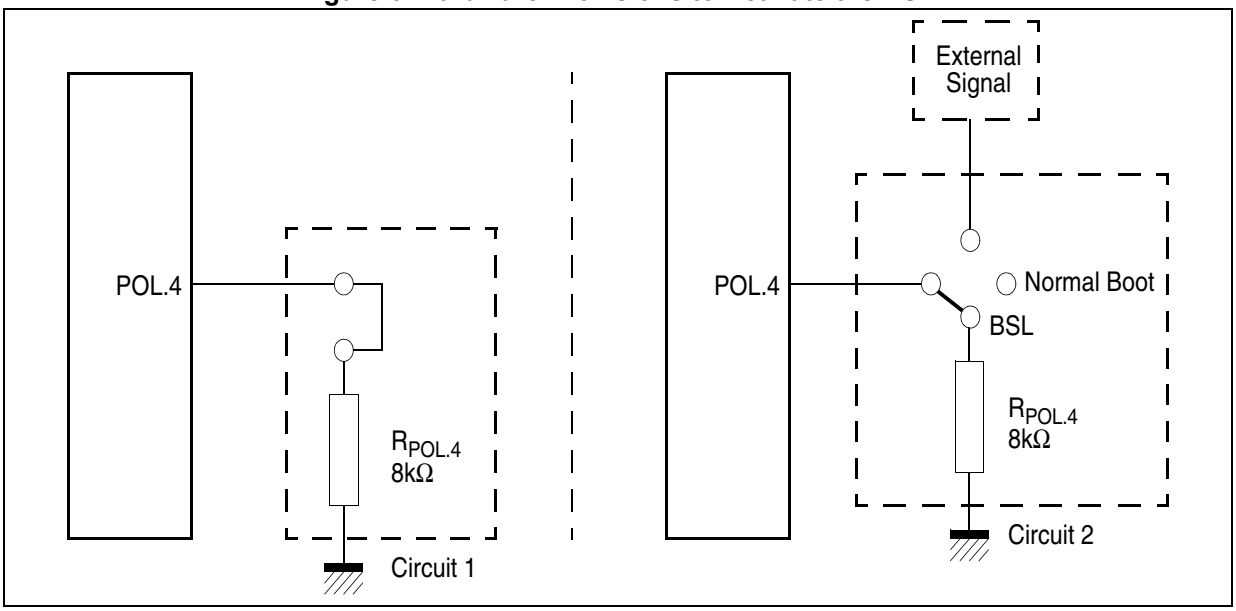
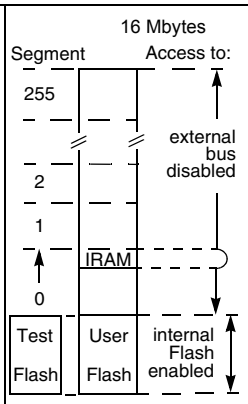
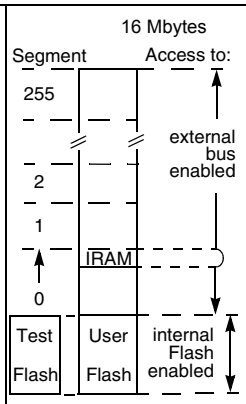
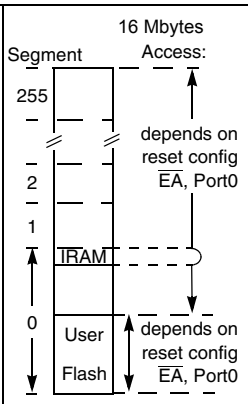


Table 4. Memory Configuration after Reset

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

5.6.3 Loading the Startup Code

After sending the identification byte the BSL enters a loop to receive 32 bytes via ASC0. These bytes are stored sequentially into locations 00'FA40h through 00'FA5Fh of the internal RAM. So up to 16 instructions may be placed into the RAM area. To execute the loaded code the BSL then jumps to location 00'FA40h, which is the first loaded instruction.

The bootstrap loading sequence is now terminated, the ST10F269Z1-ST10F269Z2 remain in BSL mode, however. Most probably the initially loaded routine will load additional code or data, as an average application is likely to require substantially more than 16 instructions. This second receive loop may directly use the pre-initialized interface ASC0 to receive data and store it to arbitrary user-defined locations.

This second level of loaded code may be the final application code. It may also be another, more sophisticated, loader routine that adds a transmission protocol to enhance the integrity of the loaded code or data. It may also contain a code sequence to change the system configuration and enable the bus interface to store the received data into external memory.

This process may go through several iterations or may directly execute the final application. In all cases the ST10F269 will still run in BSL mode, that means with the watchdog timer disabled and limited access to the internal Flash area.

All code fetches from the internal Flash area (00'0000h...00'7FFFh or 01'0000h...01'7FFFh, if mapped to segment 1) are redirected to the special Boot-ROM. Data fetches access will access the internal Boot-ROM of the ST10F269, if any is available, but will return undefined data on ROMless devices.

5.6.4 Exiting Bootstrap Loader Mode

In order to execute a program in normal mode, the BSL mode must be terminated first. The ST10F269 exits BSL mode upon a software reset (ignores the level on P0L.4) or a hardware

reset (P0L.4 must be high). After a reset the ST10F269Z1-ST10F269Z2 will start executing from location 00'0000h of the internal Flash or the external memory, as programmed via pin $\overline{EA}$.

5.6.5 Choosing the Baud Rate for the BSL

The calculation of the serial baud rate for ASC0 from the length of the first zero Byte that is received, allows the operation of the bootstrap loader of the ST10F269Z1-ST10F269Z2 with a wide range of baud rates. However, the upper and lower limits have to be kept, in order to insure proper data transfer.

$$\frac{f_{CPU}}{32 \times (S0BRL + 1)} B_S$$

The ST10F269Z1-ST10F269Z2 use timer T6 to measure the length of the initial zero Byte. The quantization uncertainty of this measurement implies the first deviation from the real baud rate, the next deviation is implied by the computation of the S0BRL reload value from the timer contents. The formula below shows the association:

$$S0BRL = \frac{T6 - 36}{72} \quad , \quad T6 = \frac{9}{4} \times \frac{f_{CPU}}{B_{Host}}$$

For a correct data transfer from the host to the ST10F269Z1-ST10F269Z2 the maximum deviation between the internal initialized baud rate for ASC0 and the real baud rate of the host should be below 2.5 %. The deviation (F_B, in percent) between host baud rate and ST10F269Z1-ST10F269Z2 baud rate can be calculated via the formula below:

$$F_B = \left| \frac{B_{Contr} - B_{Host}}{B_{Contr}} \right| \times 100 \ \% \ , F_B \leq 2.5 \ \%$$

Function (F_B) does not consider the tolerances of oscillators and other devices supporting the serial communication.

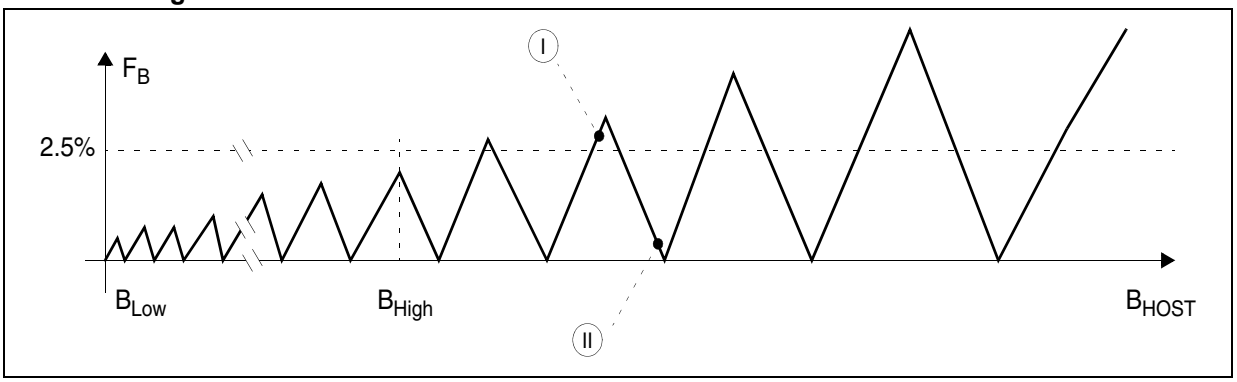
This baud rate deviation is a nonlinear function depending on the CPU clock and the baud rate of the host. The maximum of the function (F_B) increases with the host baud rate due to the smaller baud rate pre-scaler factors and the implied higher quantization error (see *Figure 7*).

The minimum baud rate (B_{Low} in *Figure 7*) is determined by the maximum count capacity of timer T6, when measuring the zero Byte, and it depends on the CPU clock. Using the maximum T6 count 216 in the formula the minimum baud rate can be calculated. The lowest standard baud rate in this case would be 1200 baud. Baud rates below B_{Low} would cause T6 to overflow. In this case ASC0 cannot be initialized properly.

The maximum baud rate (B_{High} in *Figure 7*) is the highest baud rate where the deviation still does not exceed the limit, so all baud rates between B_{Low} and B_{High} are below the deviation limit. The maximum standard baud rate that fulfills this requirement is 19200 baud.

Higher baud rates, however, may be used as long as the actual deviation does not exceed the limit. A certain baud rate (marked 'I' in *Figure 7*) may violate the deviation limit, while an even higher baud rate (marked 'II' in *Figure 7*) stays very well below it. This depends on the host interface.

Figure 7. Baud Rate Deviation Between Host and ST10F269Z1-ST10F269Z2



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 ST10F269Z1-ST10F269Z2 instructions can be executed in one instruction cycle which requires 50 ns at 25 MHz CPU clock and 62.5 ns at 32 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.

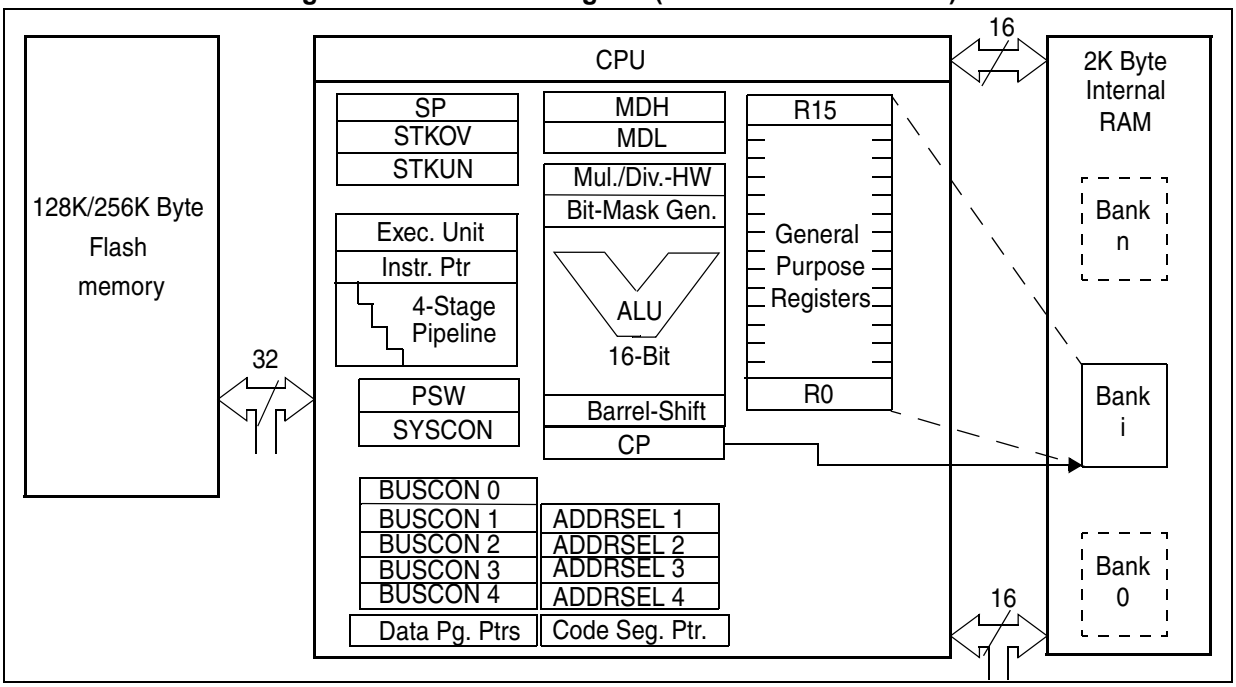
The CPU uses a bank of 16 word registers to run the current context. This bank of General Purpose Registers (GPR) is physically stored within the on-chip Internal RAM (IRAM) area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU.

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 1024 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 8. CPU Block Diagram (MAC Unit not included)



The System Configuration Register SYSCON

This bit-addressable register provides general system configuration and control functions. The reset value for register SYSCON depends on the state of the PORT0 pins during reset.

SYSCON (FF12H / 89H)

SFR Reset Value: 0xx0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STKSZ			ROM S1	SGT DIS	**ROM EN**	**BYT DIS**	CLK EN	**WR CFG**	CS CFG	PWD CFG	OWD DIS	BDR STEN	XPEN	VISI BLE	XPER-SHARE
RW			RW	RW	RW[1]	RW[1]	RW	RW[1]	RW	RW	RW	RW	RW	RW	RW

1. These bits are set directly or indirectly according to PORT0 and EA pin configuration during reset sequence.

2. Register SYSCON cannot be changed after execution of the EINIT instruction.

XPEN		**XBUS Peripheral Enable Bit**
	0	Accesses to the on-chip X-Peripherals and their functions are disabled
	1	The on-chip X-Peripherals are enabled and can be accessed.
BDRSTEN		**Bidirectional Reset Enable**
	0	$\overline{\text{RSTIN}}$ pin is an input pin only. SW Reset or WDT Reset have no effect on this pin
	1	$\overline{\text{RSTIN}}$ pin is a bidirectional pin. This pin is pulled low during 1024 TCL during reset sequence.
OWDDIS		**Oscillator Watchdog Disable Control**
	0	Oscillator Watchdog (OWD) is enabled. If PLL is bypassed, the OWD monitors XTAL1 activity. If there is no activity on XTAL1 for at least 1 ms, the CPU clock is switched automatically to PLL's base frequency (2 to 10MHz).
	1	OWD is disabled. If the PLL is bypassed, the CPU clock is always driven by XTAL1 signal. The PLL is turned off to reduce power supply current.
PWDCFG		**Power Down Mode Configuration Control**
	0	Power Down Mode can only be entered during PWRDN instruction execution if $\overline{\text{NMI}}$ pin is low, otherwise the instruction has no effect. To exit Power Down Mode, an external reset must occurs by asserting the $\overline{\text{RSTIN}}$ pin.
	1	Power Down Mode can only be entered during PWRDN instruction execution if all enabled fast external interrupt EXxIN pins are in their inactive level. Exiting this mode can be done by asserting one enabled EXxIN pin.
CSCFG		**Chip Select Configuration Control**
	0	Latched Chip Select lines: CSx change 1 TCL after rising edge of ALE
	1	Unlatched Chip Select lines: CSx change with rising edge of ALE

6.1 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.

Signal processing needs at least three specialized units operating in parallel to achieve maximum performance:

- A Multiply-Accumulate Unit,
- An Address Generation Unit, able to feed the MAC Unit with 2 operands per cycle,
- A Repeat Unit, to execute series of multiply-accumulate instructions.

The existing 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.

A new transfer instruction CoMOV has also been added to take benefit of the new addressing capabilities.

6.1.1 Features

Enhanced Addressing Capabilities

- New addressing modes including a double indirect addressing mode with pointer post-modification.
- Parallel Data Move: this mechanism allows one operand move during Multiply-Accumulate instructions without penalty.
- New transfer instructions CoSTORE (for fast access to the MAC SFRs) and CoMOV (for fast memory to memory table transfer).

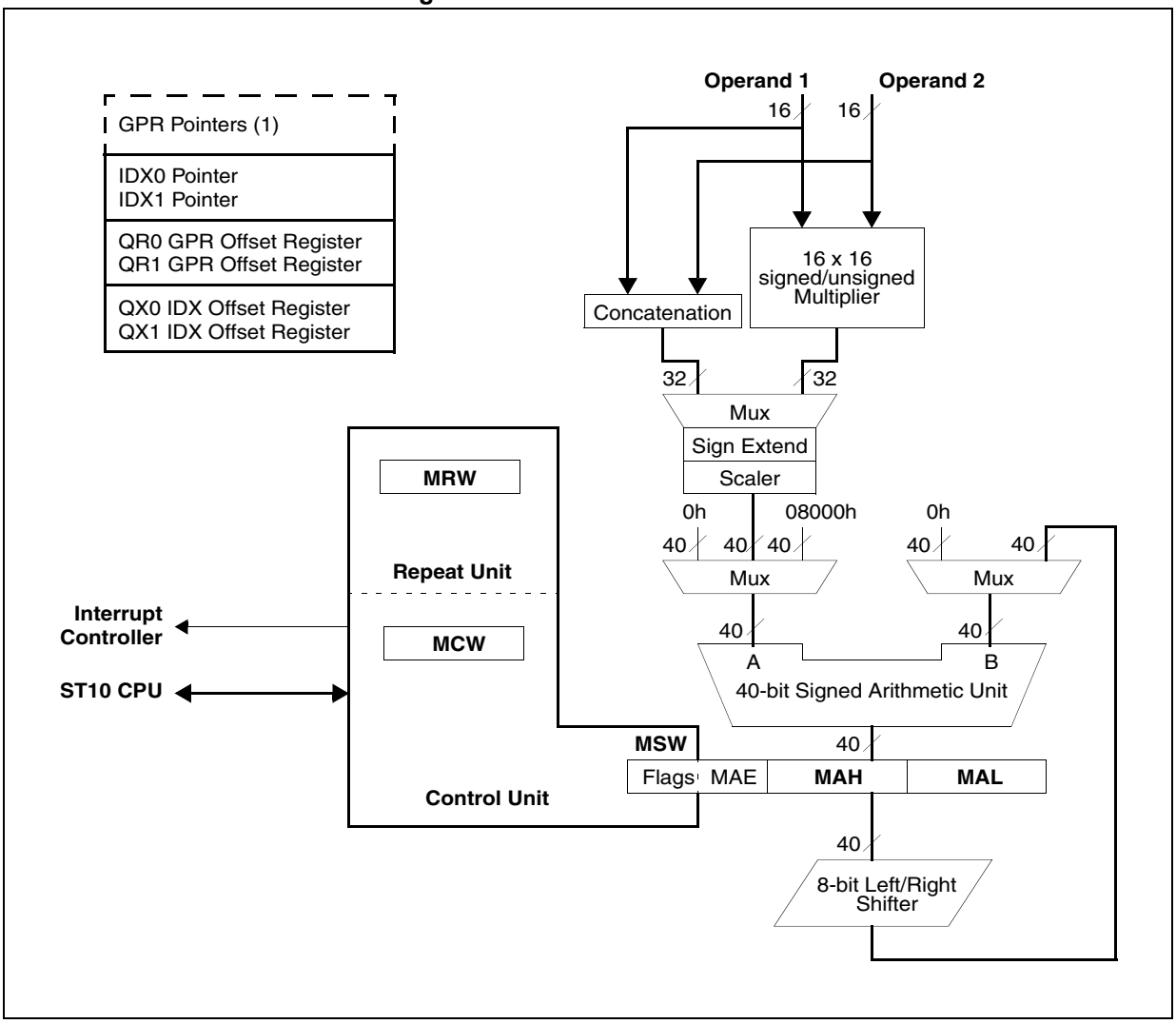
Multiply-Accumulate Unit

- One-cycle execution for all MAC operations.
- 16 x 16-bit signed/unsigned parallel multiplier.
- 40-bit signed arithmetic unit with automatic saturation mode.
- 40-bit accumulator.
- 8-bit left/right shifter.
- Full instruction set with multiply and multiply-accumulate, 32-bit signed arithmetic and compare instructions.

Program Control

- Repeat Unit: allows some MAC co-processor instructions to be repeated up to 8192 times. Repeated instructions may be interrupted.
- MAC interrupt (Class B Trap) on MAC condition flags.

Figure 9. MAC Unit Architecture



1. Shared with standard ALU.

6.2 Instruction Set Summary

Table 5 lists the instructions of the ST10F269Z1-ST10F269Z2. The various addressing modes, instruction operation, parameters for conditional execution of instructions, opcodes and a detailed description of each instruction can be found in the “ST10 Family Programming Manual”.

Table 5. 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

Table 5. Instruction Set Summary (continued)

Mnemonic	Description	Bytes
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	Bit-wise AND, (word/byte operands)	2 / 4
OR(B)	Bit-wise OR, (word/byte operands)	2 / 4
XOR(B)	Bit-wise XOR, (word/byte operands)	2 / 4
BCLR	Clear direct bit	2
BSET	Set direct bit	2
BMOV(N)	Move (negated) direct bit to direct bit	4
BAND, BOR, BXOR	AND/OR/XOR direct bit with direct bit	4
BCMP	Compare direct bit to direct bit	4
BFLDH/L	Bit-wise modify masked high/low byte of bit-addressable direct word memory with immediate data	4
CMP(B)	Compare word (byte) operands	2 / 4
CMPD1/2	Compare word data to GPR and decrement GPR by 1/2	2 / 4
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
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

Table 5. Instruction Set Summary (continued)

Mnemonic	Description	Bytes
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

6.3 MAC Coprocessor Specific Instructions

Table 6 gives an overview of the MAC instruction set. All the mnemonics are listed with the addressing modes that can be used with each instruction.

For each combination of mnemonic and addressing mode this table indicates if it is repeatable or not.

New addressing capabilities enable the CPU to supply the MAC with up to 2 operands per instruction cycle. MAC instructions: multiply, multiply-accumulate, 32-bit signed arithmetic operations and the CoMOV transfer instruction have been added to the standard instruction set. Full details are provided in the ‘ST10 Family Programming Manual’. Double indirect addressing requires two pointers. Any GPR can be used for one pointer, the other pointer is provided by one of two specific SFRs IDX0 and IDX1. Two pairs of offset registers QR0/QR1 and QX0/QX1 are associated with each pointer (GPR or IDXi).

The GPR pointer allows access to the entire memory space, but IDXi are limited to the internal Dual-Port RAM, except for the CoMOV instruction.

Table 6. MAC specific instructions

Mnemonic	Addressing Modes	Repeatability
CoMUL CoMULu CoMULus CoMULsu CoMUL- CoMULu- CoMULus- CoMULsu- CoMUL, rnd CoMULu, rnd CoMULus, rnd CoMULsu, rnd	Rw_n, Rw_m $[IDX_i\otimes]$, $[Rw_m\otimes]$ Rw_n, $[Rw_m\otimes]$	No No No
CoMAC CoMACu CoMACus CoMACsu CoMAC- CoMACu- CoMACus- CoMACsu- CoMAC, rnd CoMACu, rnd CoMACus, rnd CoMACsu, rnd CoMACR CoMACRu	Rw_n, Rw_m $[IDX_i\otimes]$, $[Rw_m\otimes]$ Rw_n, $[Rw_m\otimes]$	No Yes Yes
CoMACRus CoMACRsu CoMACR, rnd CoMACRu, rnd CoMACRus, rnd CoMACRsu, rnd	Rw_n, Rw_m $[IDX_i\otimes]$, $[Rw_n\otimes]$ Rw_n, $[RW_m\otimes]$	No No No
CoNOP	$[Rw_m\otimes]$ $[IDX_i\otimes]$ $[IDX_i\otimes]$, $[Rw_m\otimes]$	Yes Yes Yes
CoNEG CoNEG, rnd CoRND	-	No
CoSTORE	Rw_n, CoReg $[Rw_n\otimes]$, Coreg	No Yes
CoMOV	$[IDX_i\otimes]$, $[Rw_m\otimes]$	Yes

Table 6. MAC specific instructions (continued)

Mnemonic	Addressing Modes	Repeatability
CoMACM CoMACMu CoMACMus CoMACMsu CoMACM- CoMACMu- CoMACMus- CoMACMsu- CoMACM, rnd CoMACMu, rnd CoMACMus, rnd CoMACMsu, rnd CoMACMR CoMACMRu CoMACMRus CoMACMRsu CoMACMR, rnd CoMACMRu, rnd CoMACMRus, rnd CoMACMRsu, rnd	[IDX_i⊗], [Rw_m⊗]	Yes
CoADD CoADD2 CoSUB CoSUB2 CoSUBR CoSUB2R CoMAX CoMIN	Rw_n, Rw_m [IDX_i⊗], [Rw_m⊗] Rw_n, [Rw_m⊗]	No Yes Yes
CoLOAD CoLOAD- CoLOAD2 CoLOAD2- CoCMP	Rw_n, Rw_m [IDX_i⊗], [Rw_m⊗] Rw_n, [Rw_m⊗]	No No No
CoSHL CoSHR CoASHR CoASHR, rnd	Rw_m #data4 [Rw_m⊗]	Yes No Yes
CoABS	- Rw_n, Rw_m [IDX_i⊗], [Rw_m⊗] Rw_n, [Rw_m⊗]	No No No

Table 7 shows the various combinations of pointer post-modification for each of these 2 new addressing modes. In this document the symbols “[RwnÄ]” and “[IDXiÄ]” refer to these addressing modes.

Table 7. Pointer Post-modification Combinations for IDXi and Rwn

Symbol	Mnemonic	Address Pointer Operation	
“[IDX_i⊗]” stands for	[IDX_i]	(IDX_i) ←(IDX_i)	(no-op)
	[IDX_i+]	(IDX_i) ←(IDX_i) + 2	(i=0,1)
	[IDX_i-]	(IDX_i) ←(IDX_i) - 2	(i=0,1)
	[IDX_i +QX_j]	(IDX_i) ←(IDX_i) + (QX_j)	(i, j =0,1)
	[IDX_i - QX_j]	(IDX_i) ←(IDX_i) - (QX_j)	(i, j =0,1)
“[Rw_n⊗]” stands for	[Rwn]	(Rwn) ←(Rwn)	(no-op)
	[Rwn+]	(Rwn) ←(Rwn) + 2	(n=0-15)
	[Rwn-]	(Rwn) ←(Rwn) - 2	(n=0-15)
	[Rwn +QR_j]	(Rwn) ←(Rwn) + (QR_j)	(n=0-15; j =0,1)
	[Rwn - QR_j]	(Rwn) ←(Rwn) - (QR_j)	(n=0-15; j =0,1)

Table 8. MAC Registers Referenced as ‘CoReg‘

Registers	Description	Address in Opcode
MSW	MAC-Unit Status Word	00000b
MAH	MAC-Unit Accumulator High	00001b
MAS	“limited” MAH /signed	00010b
MAL	MAC-Unit Accumulator Low	00100b
MCW	MAC-Unit Control Word	00101b
MRW	MAC-Unit Repeat Word	00110b

7 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 4 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 4 address windows are controlled by BUSCON0. Up to 5 external CS signals (4 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 $\overline{\text{HOLD}}$ / $\overline{\text{HLDA}}$ 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 ($\overline{\text{BREQ}}$, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$) are automatically controlled by the EBC. In master mode (default after reset) the $\overline{\text{HLDA}}$ pin is an output. By setting bit DP6.7 to’1’ the slave mode is selected where pin $\overline{\text{HLDA}}$ 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 Mbytes, 256 Kbytes or to 64 Kbytes. Port 4 outputs all 8 address lines if an address space of 16M Bytes is used, otherwise four, two or no address lines.

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

The active level of the 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.

7.1 Programmable Chip Select Timing Control

The ST10F269 allows the user to adjust the position of the CSx line changes. By default (after reset), the CSx lines change half a CPU clock cycle (12.5 ns at 40 MHz of CPU clock and 31.25 ns at 32 MHz of CPU clock) after the rising edge of ALE. With the CSCFG bit set in the SYSCON register the CSx lines change with the rising edge of ALE, thus the CSx lines and the address lines change at the same time (see *Figure 10*).

7.2 READY Programmable Polarity

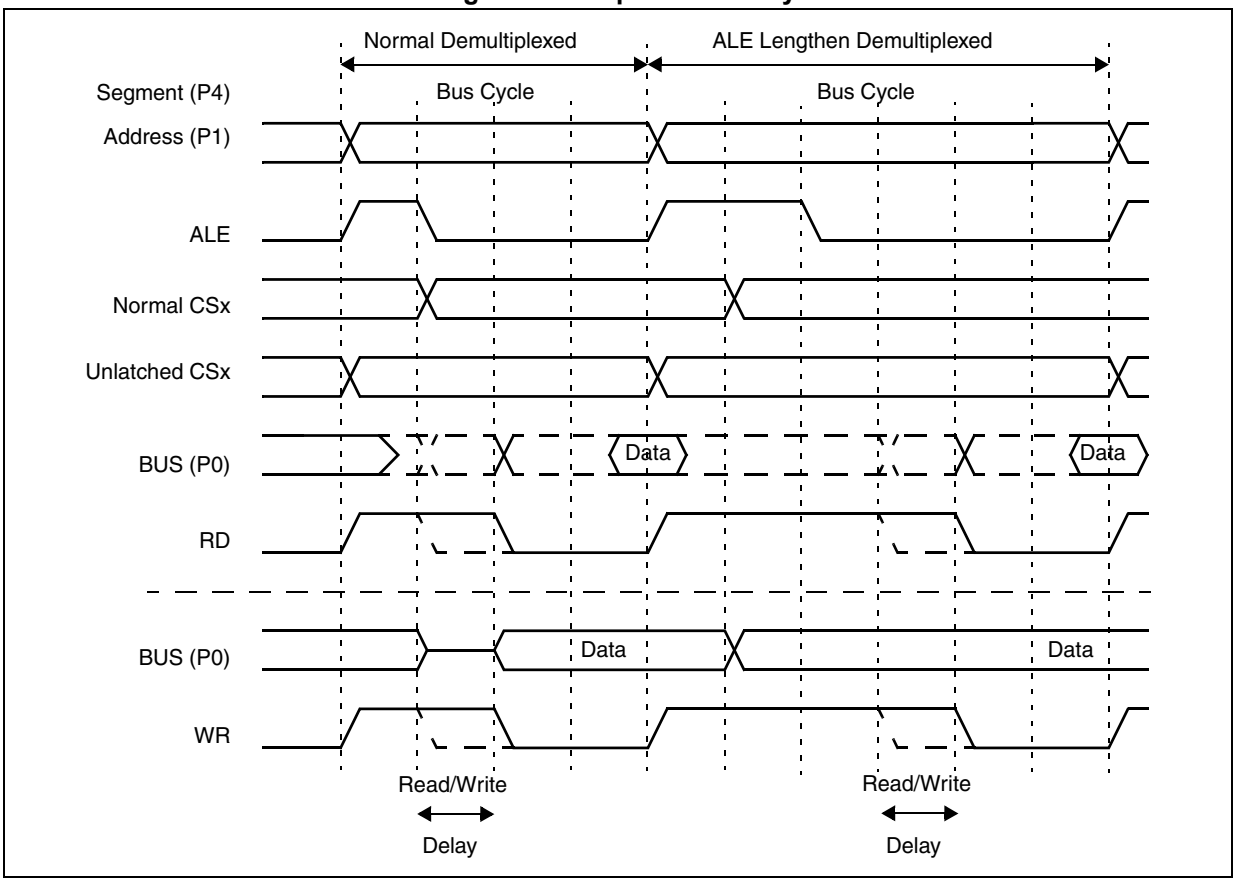
The active level of the READY pin can be selected by software via the RDYPOL bit in the BUSCONx registers.

When the READY function is enabled for a specific address window, each bus cycle within this window must be terminated with the active level defined by this RDYPOL bit in the associated BUSCON register.

BUSCONx registers are described in *Section 20.2: System Configuration Registers*.

Note: ST10F269Z1-ST10F269Z2 have no internal pull-up resistor on READY pin.

Figure 10. Chip Select Delay



8 Interrupt System

The interrupt response time for internal program execution is from 125 ns to 300 ns at 40 MHz CPU clock and from 156.25 ns to 375 ns at 32 MHz CPU clock.

The ST10F269Z1-ST10F269Z2 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 ST10F269 has 8 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.

8.1 External Interrupts

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 receive signal (CANx_RxD) can be used to interrupt the system. This new function is controlled using the ‘External Interrupt Source Selection’ register EXISEL.

EXISEL (F1DAH / EDH)

ESFR Reset Value: 0000H

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
EXI7SS	EXI6SS	EXI5SS	EXI4SS	EXI3SS	EXI2SS	EXI1SS	EXI0SS
RW	RW	RW	RW	RW	RW	RW	RW

EXIxSS	**External Interrupt x Source Selection (x=7...0)** ‘00’: Input from associated Port 2 pin. ‘01’: Input from “alternate source”. ‘10’: Input from Port 2 pin ORed with “alternate source”. ‘11’: Input from Port 2 pin ANDed with “alternate source”.

EXIxSS	Port 2 pin	Alternate Source
0	P2.8	CAN1_RxD
1	P2.9	CAN2_RxD
2	P2.10	RTCSI (Timed)
3	P2.11	RTCAI (Alarm)
4...7	P2.12...15	Not used (zero)

8.2 Interrupt Registers and Vectors Location List

Table 9 shows all the available ST10F269 interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers:

Table 9. 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
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

Table 9. Interrupt Sources (continued)

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
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
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
CAN1 Interface	XP0IR	XP0IE	XP0INT	00'0100h	40h
CAN2 Interface	XP1IR	XP1IE	XP1INT	00'0104h	41h
FLASH Ready / Busy	XP2IR	XP2IE	XP2INT	00'0108h	42h
PLL Unlock/OWD	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). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any other program execution. Hardware trap services cannot not be interrupted by standard interrupt or by PEC interrupts.

8.3 Interrupt Control Registers

All interrupt control registers are identically organized. The lower 8 bits of an interrupt control register contain the complete interrupt status information of the associated source, which is required during one round of prioritization, the upper 8 bits of the respective register are reserved. All interrupt control registers are bit addressable and all bits can be read or written via software.

This allows each interrupt source to be programmed or modified with just one instruction. When accessing interrupt control registers through instructions which operate on Word data types, their upper 8 bits (15...8) will return zeros, when read, and will discard written data.

The layout of the Interrupt Control registers shown below applies to each xxIC register, where xx stands for the mnemonic for the respective source.

xxIC (yyyyh / zzh)

SFR Area Reset Value: - - 00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	**xxIR**	xxIE	ILVL				GLVL	
								RW	RW	RW				RW	

GLVL	**Group Level** Defines the internal order for simultaneous requests of the same priority. 3: Highest group priority 0: Lowest group priority
ILVL	**Interrupt Priority Level** Defines the priority level for the arbitration of requests. Fh: Highest priority level 0h: Lowest priority level
xxIE	**Interrupt Enable Control Bit** (individually enables/disables a specific source) ‘0’: Interrupt Request is disabled ‘1’: Interrupt Request is enabled
xxIR	**Interrupt Request Flag** ‘0’: No request pending ‘1’: This source has raised an interrupt request

8.4 Exception and Error Traps List

Table 10 shows all of the possible exceptions or error conditions that can arise during run-time.

Table 10. Trap Priorities

Exception Condition	Trap Flag	Trap Vector	Vector Location	Trap Number	Trap[(1)] Priority
Reset Functions:					
– Hardware Reset	-	RESET	00’0000h	00h	III
– Software Reset		RESET	00’0000h	00h	III
– Watchdog Timer Overflow		RESET	00’0000h	00h	III

Table 10. Trap Priorities (continued)

Exception Condition	Trap Flag	Trap Vector	Vector Location	Trap Number	Trap[1] Priority
Class A Hardware Traps:					
– Non-Maskable Interrupt	NMI	NMITRAP	00’0008h	02h	II
– Stack Overflow	STKOF	STOTRAP	00’0010h	04h	II
– Stack Underflow	STKUF	STUTRAP	00’0018h	06h	II
Class B Hardware Traps:					
Undefined Opcode	UNDOPC	BTRAP	00’0028h	0Ah	I
Protected Instruction Fault	PRTFLT	BTRAP	00’0028h	0Ah	I
Illegal word Operand Access	ILLOPA	BTRAP	00’0028h	0Ah	I
Illegal Instruction Access	ILLINA	BTRAP	00’0028h	0Ah	I
Illegal External Bus Access	ILLBUS	BTRAP	00’0028h	0Ah	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 compare to the class A traps and to the resets.
Each class A traps 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.

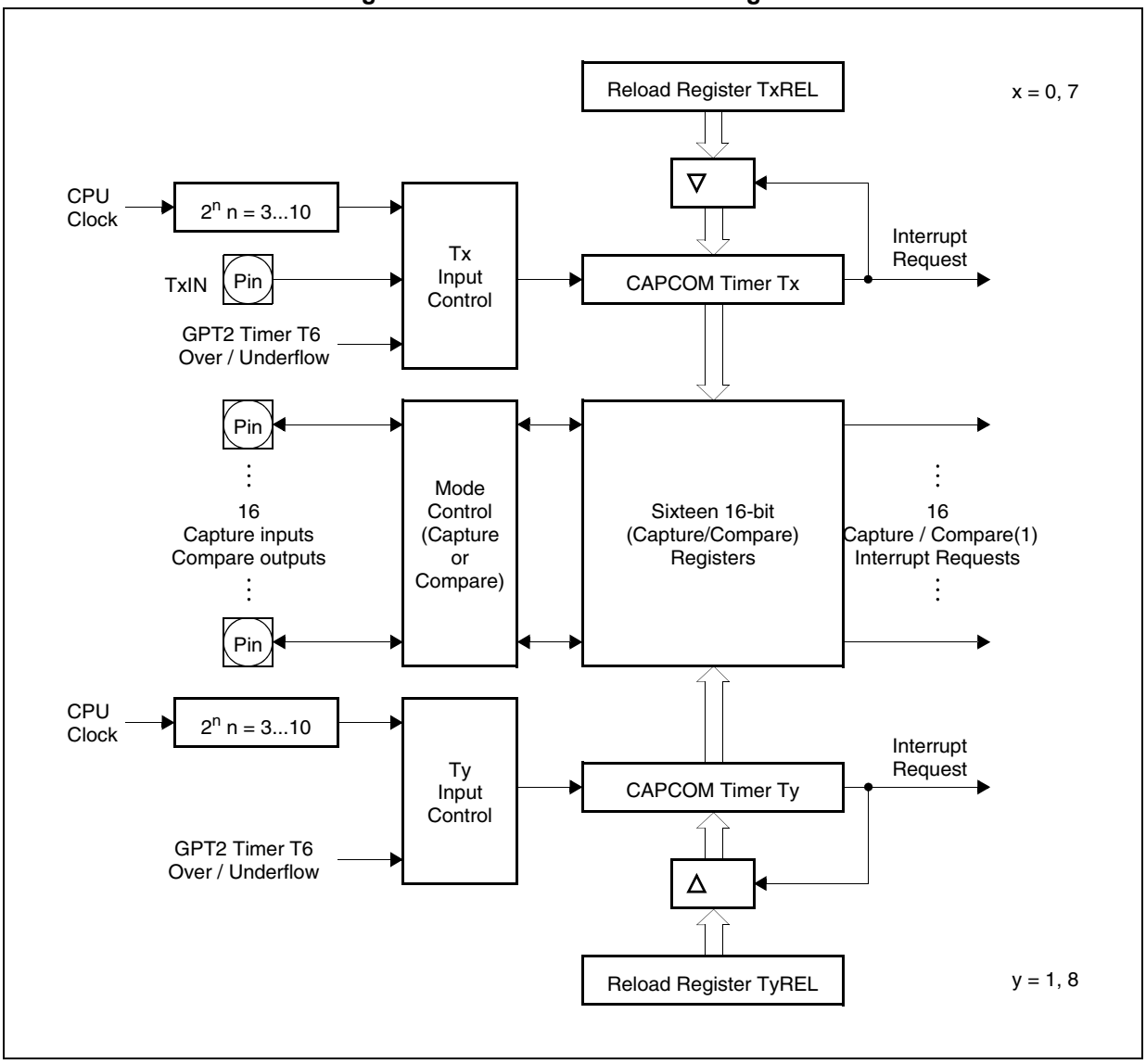
9 Capture/Compare (CAPCOM) Units

The ST10F269 has two 16 channels CAPCOM units as described in *Figure 11*. These support generation and control of timing sequences on up to 32 channels with a maximum resolution of 200 ns at 40 MHz CPU clock and 250 ns at 32 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
(See *Figure 12* and *Figure 13*).

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. *Figure 11* shows the basic structure of the two CAPCOM units.

Figure 11. CAPCOM Unit Block Diagram



1. The CAPCOM2 unit provides 16 capture inputs, but only 12 compare outputs. CC24I to CC27I are inputs only.

Figure 12. Block Diagram of CAPCOM Timers T0 and T7

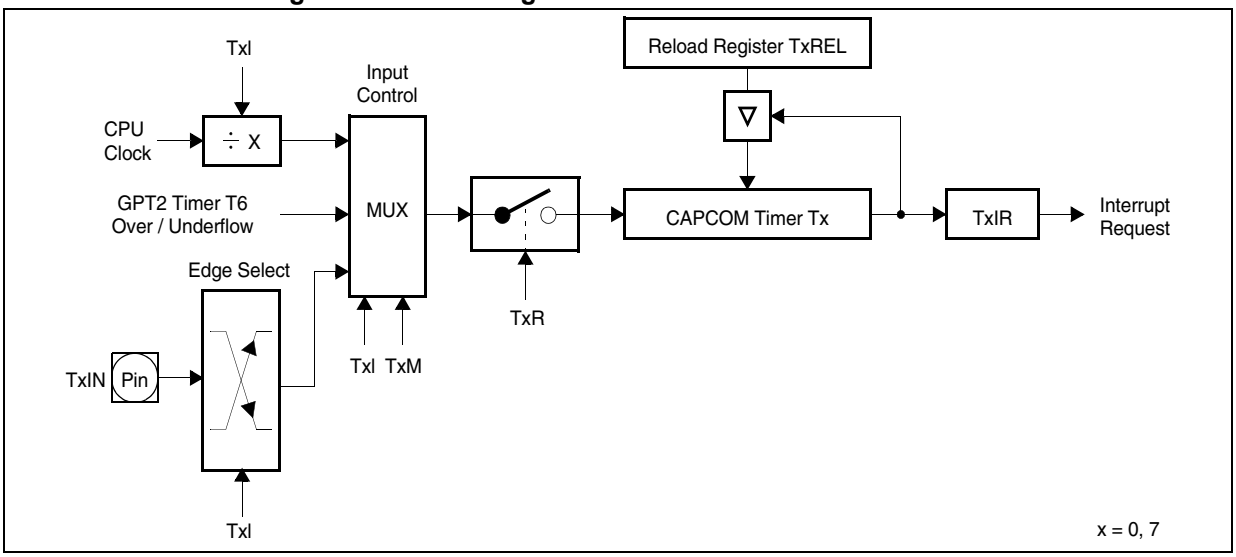
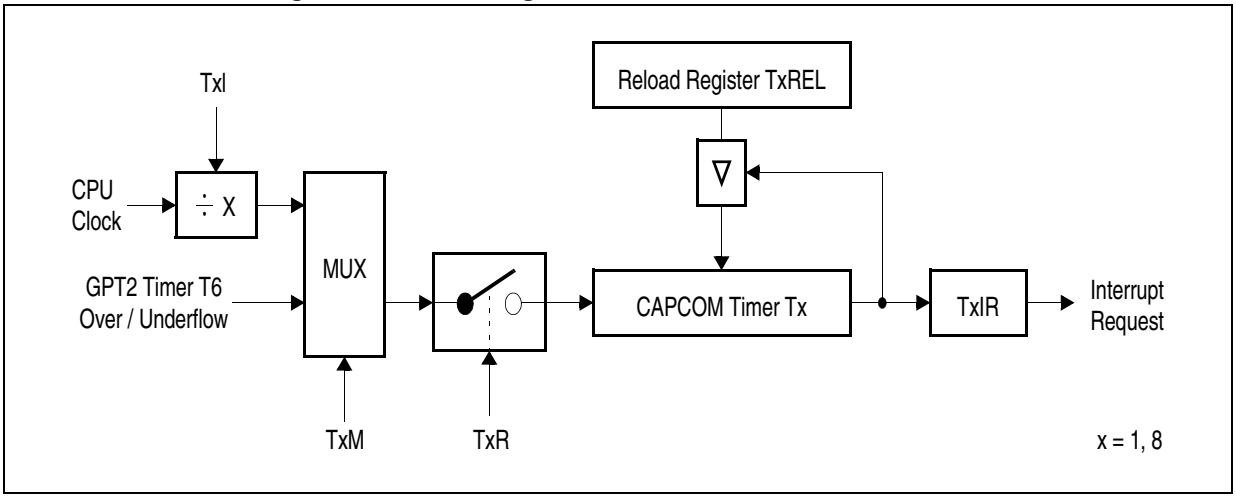


Figure 13. Block Diagram of CAPCOM Timers T1 and T8



Note: When an external input signal is connected to the input lines of both T0 and T7, these timers count the input signal synchronously. Thus the two timers can be regarded as one timer whose contents can be compared with 32 capture registers.

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 (see *Table 11*).

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 pre-scaler option in TxI when using a 40 MHz or a 32 MHz CPU clock, are listed in *Table 12* and *Table 13*.

The numbers for the timer periods are based on a reload value of 0000h. Note that some numbers may be rounded to 3 significant figures.

Table 11. 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 12. CAPCOM Timer Input Frequencies, Resolution and Periods (f_{CPU} = 40 MHz)

f_{CPU} = 25 MHz	Timer Input Selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler 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	156.25kHz	78.125kHz	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

Table 13. CAPCOM Timer Input Frequencies, Resolution and Periods (f_{CPU} = 32MHz)

f_{CPU} = 32 MHz	Timer Input Selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input Frequency	4 MHz	2 MHz	1 MHz	500 KHz	250 KHz	125 KHz	62.5KHz	31.125KHz
Resolution	250 ns	500 ns	1 µs	2 µs	4 µs	8 µs	16 µs	32 µs
Period	16.4 ms	32.8 ms	65.5 ms	131 ms	262.1 ms	524.3 ms	1.05 s	2.1 s

10 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.

10.1 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 14 and *Table 15* list the timer input frequencies, resolution and periods for each pre-scaler option at 40 MHz or 32 MHz CPU clock. This also applies to the Gated Timer Mode of T3 and to the auxiliary timers T2 and T4 in Timer and Gated Timer Mode. The count direction (up/down) for each timer is programmable by software or may be altered dynamically by an external signal on a port pin (TxEUD).

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. When used as capture or reload registers, timers T2 and T4 are stopped. The contents of timer T3 is captured into T2 or T4 in response to a signal at their associated input pins (TxIN).

Timer T3 is reloaded with the contents of T2 or T4 triggered either by an external signal or by a selectable state transition of its toggle latch T3OTL. When both T2 and T4 are configured to alternately reload T3 on opposite state transitions of T3OTL with the low and

high times of a PWM signal, this signal can be constantly generated without software intervention.

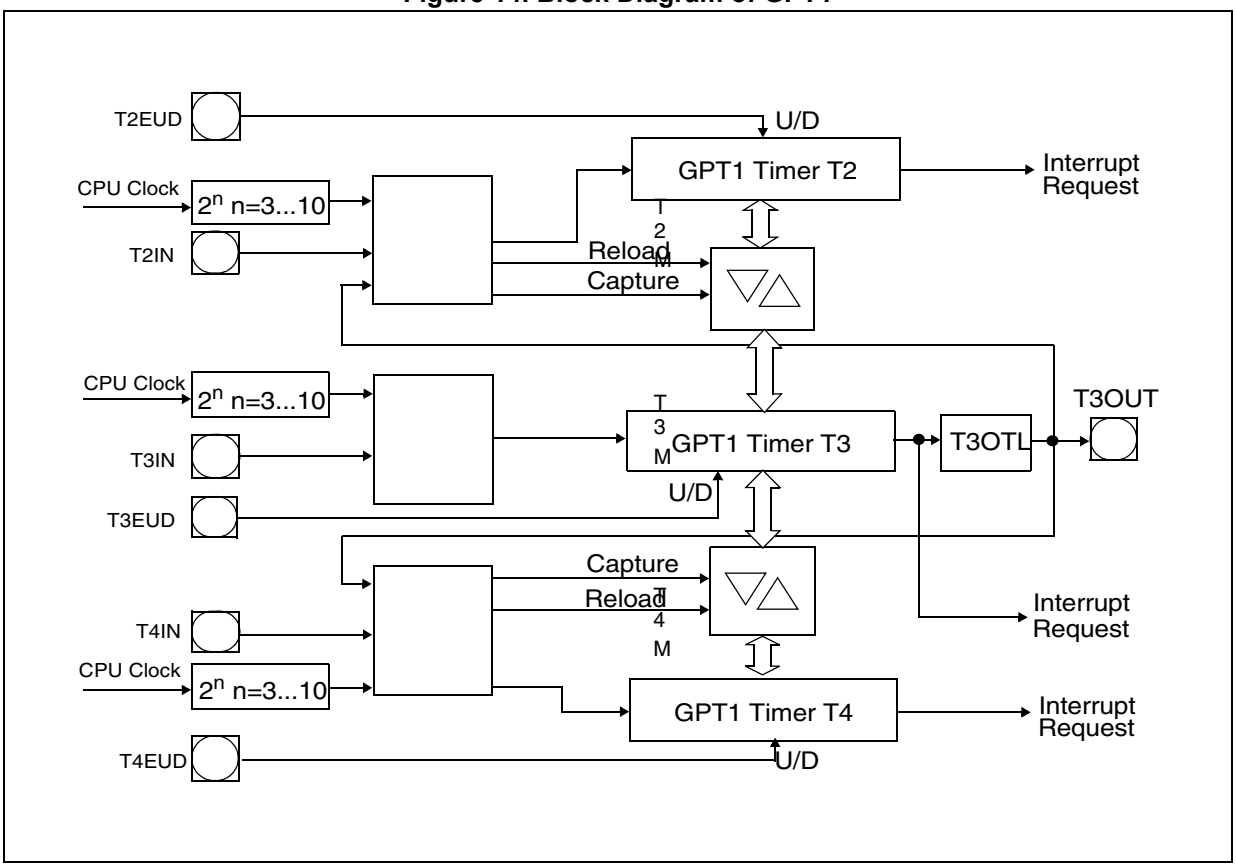
Table 14. GPT1 Timer Input Frequencies, Resolution and Periods (f_{CPU} = 40 MHz)

f_{CPU} = 40 MHz	Timer Input Selection T2I / T3I / T4I							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler factor	8	16	32	64	128	256	512	1024
Input Freq	5 MHz	2.5 MHz	1.25 MHz	625 kHz	312.5kHz	156.25 kHz	78.125kHz	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

Table 15. PT1 Timer Input Frequencies, Resolution and Periods (f_{CPU} = 32 MHz)

f_{CPU} = 32 MHz	Timer Input Selection T2I / T3I / T4I							
	000b	001b	010b	011b	100b	101b	110b	111b
Pre-scaler factor	8	16	32	64	128	256	512	1024
Input Freq	4 MHz	2 MHz	1 MHz	500 KHz	250 KHz	125 KHz	62.5KHz	31.125KHz
Resolution	250 ns	500 ns	1 µs	2 µs	4 µs	8 µs	16 µs	32 µs
Period maximum	16.4 ms	32.8 ms	65.5 ms	131 ms	262.1 ms	524.3 ms	1.05 s	2.1 s

Figure 14. Block Diagram of GPT1



10.2 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 16 and *Table 17* list the timer input frequencies, resolution and periods for each pre-scaler option at 40 MHz or 32 MHz CPU clock.

This also applies to the Gated Timer Mode of T6 and to the auxiliary timer T5 in Timer and Gated Timer Mode.

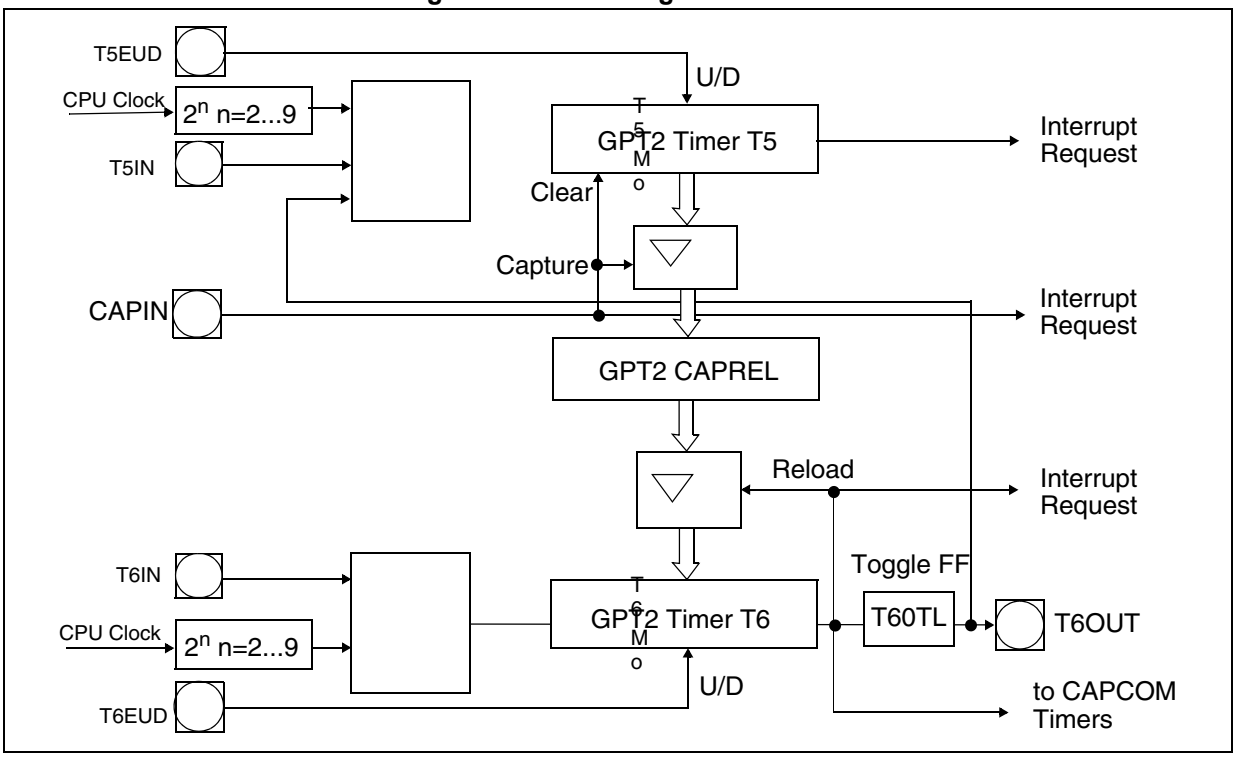
Table 16. GPT2 Timer Input Frequencies, Resolution and Period (f_{CPU} = 40 MHz)

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

Table 17. GPT2 Timer Input Frequencies, Resolution and Period (f_{CPU} = 32 MHz)

f_{CPU} = 32MHz	Timer Input Selection T5I / T6I							
	000b	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
Pre-scaler factor	4	8	16	32	64	128	256	512
Input Freq	8MHz	4MHz	2MHz	1MHz	500KHz	250KHz	125KHz	62.5KHz
Resolution	125ns	250ns	500ns	1µs	2µs	4µs	8µs	16µs
Period maximum	8.19ms	16.4ms	32.8ms	65.5ms	131ms	262.1ms	524.3ms	1.05s

Figure 15. Block Diagram of GPT2



11 PWM Module

The pulse width modulation module can generate up to four PWM output signals using edge-aligned or center-aligned PWM. In addition, the PWM module can generate PWM burst signals and single shot outputs. *Table 18* and *Table 19* show the PWM frequencies for different resolutions. The level of the output signals is selectable and the PWM module can generate interrupt requests.

Figure 16. Block Diagram of PWM Modules

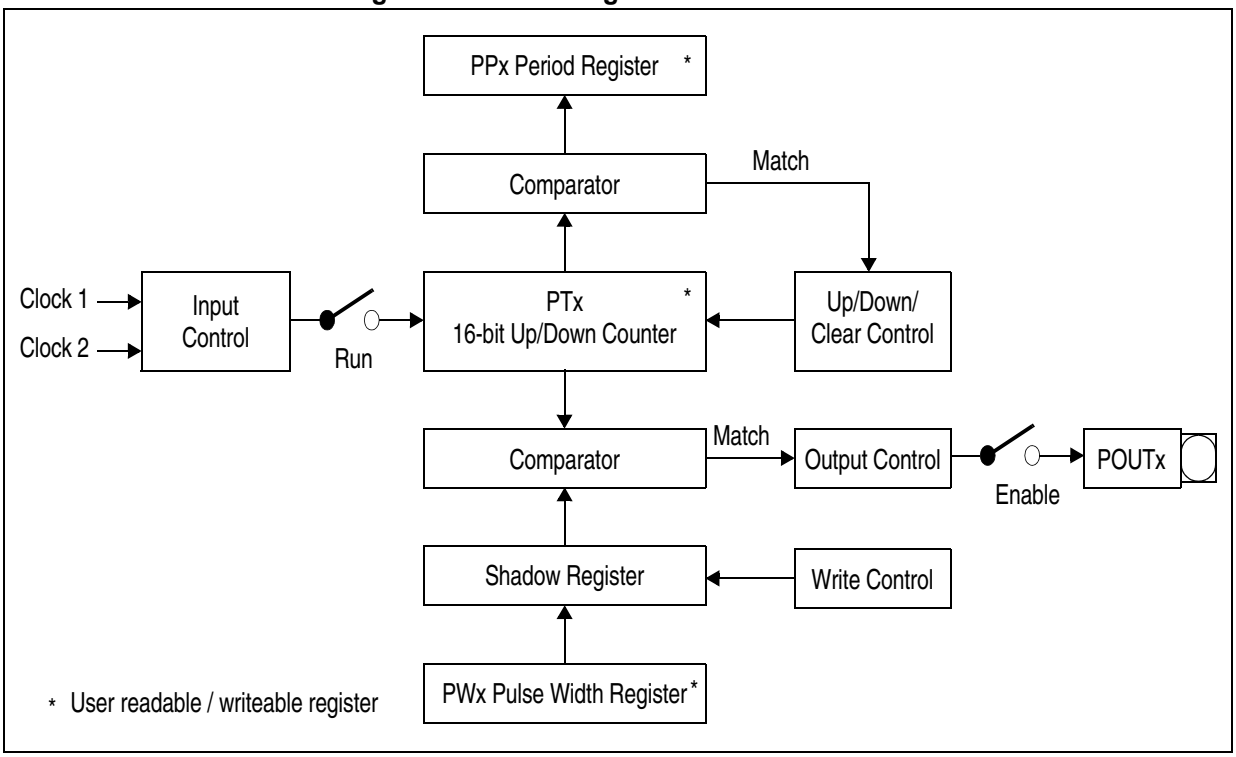


Table 18. PWM Unit Frequencies and Resolution at 40 MHz CPU Clock

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

Table 19. PWM Unit Frequencies and Resolution at 32 MHz CPU Clock

Mode 0	Resolution	8-bit	10-bit	12-bit	14-bit	16-bit
CPU Clock/1	31.25ns	125KHz	31.25KHz	7.81KHz	1.953KHz	976.6Hz
CPU Clock/64	2.00µs	1.953KHz	488.3Hz	122.1Hz	30.52Hz	7.63Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU Clock/1	31.25ns	62.5KHz	15.62KHz	3.90KHz	976.6Hz	244.1Hz
CPU Clock/64	2.00µs	976.6Hz	244.1Hz	61Hz	15.26Hz	3.81Hz

12 Parallel Ports

12.1 Introduction

The ST10F269 MCU provides up to 111 I/O lines with programmable features. These capabilities bring very flexible adaptation of this MCU to wide range of applications.

ST10F269 has 9 groups of I/O lines gathered as following:

- Port 0 is a 2 time 8-bit port named P0L (Low as less significant byte) and P0H (high as most significant byte)
- Port 1 is a 2 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 a 8-bit port
- Port 5 is a 16-bit port input only
- Port 6, Port 7 and Port 8 are 8-bit port

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 (bit-wise) for push-pull or open drain operation using ODPx registers.

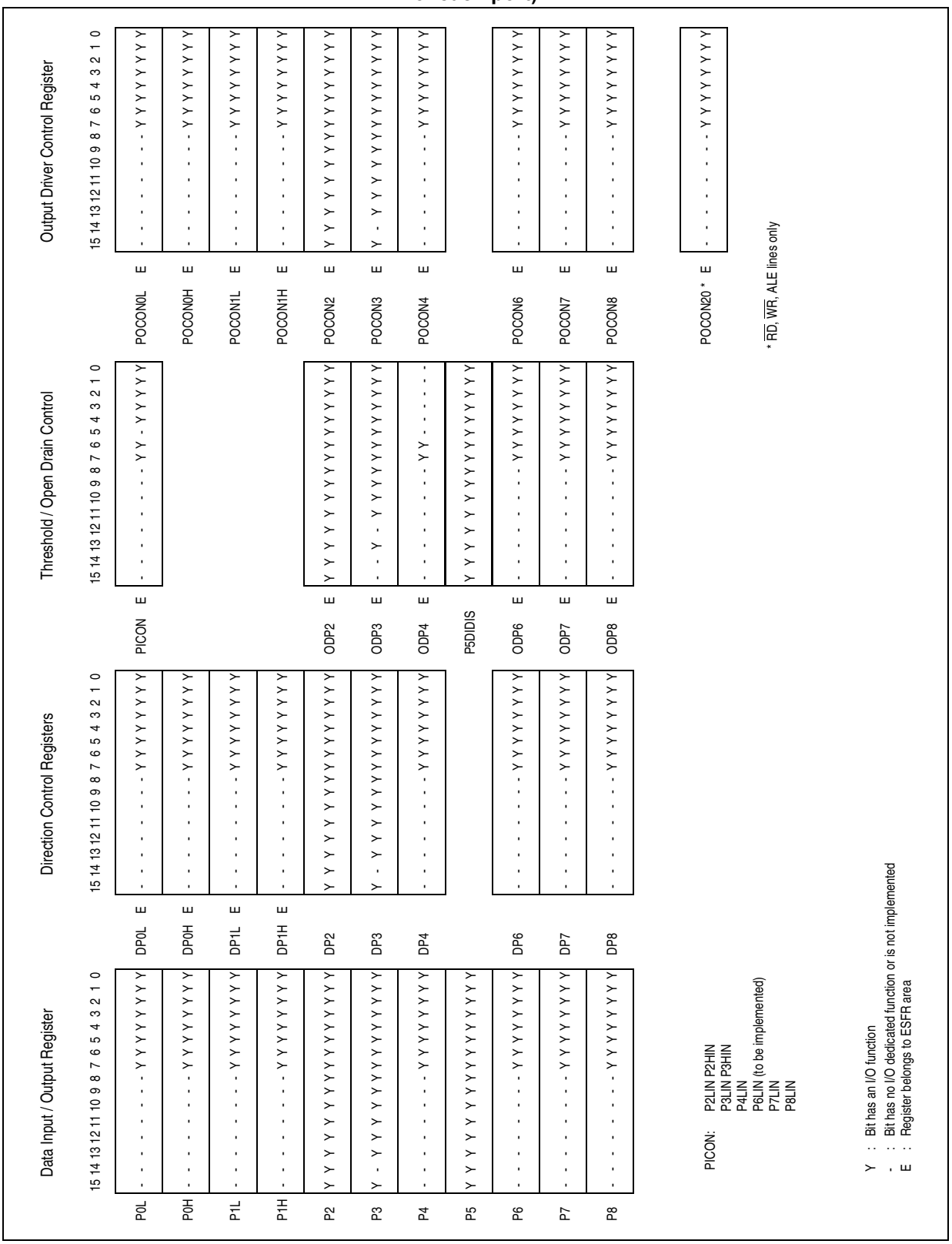
In addition, the sink and the source capability and the rise / fall time of the transition of the signal of some of the push-pull buffers can be programmed to fit the driving requirements of the application and to minimize EMI. This feature is implemented on Port 0, 1, 2, 3, 4, 6, 7 and 8 with the control registers POCONx. The output drivers capabilities of ALE, RD, WR control lines are programmable with the dedicated bits of POCON20 control register.

The input threshold levels are programmable (TTL/CMOS) for 5 ports (2, 3, 4, 7, 8). 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 the PICON register 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.

Figure 17. I/O lines support an alternate function (detailed in the following description of each port)



12.2 I/Os Special Features

12.2.1 Open Drain Mode

Some of the I/O ports of ST10F269 support the open drain capability. This programmable feature may be used with an external pull-up resistor, in order to get 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. These registers allow the individual bit-wise selection of the open drain mode for each port line. If the respective control bit ODPx.y is '0' (default after reset), the output driver is in the push-pull mode. If ODPx.y is '1', the open drain configuration is selected. Note that all ODPx registers are located in the ESFR space (See *Figure 18*).

12.2.2 Input Threshold Control

The standard inputs of the ST10F269 determine the status of input signals according to TTL levels. In order to accept and recognize noisy signals, CMOS-like input thresholds can be selected instead of the standard TTL thresholds for all pins of Port 2, Port 3, Port 4, Port 7 and Port 8. These special thresholds are defined above the TTL thresholds and feature a defined hysteresis to prevent the inputs from toggling while the respective input signal level is near the thresholds.

The Port Input Control register PICON is used to select these thresholds for each byte of the indicated ports, this means the 8-bit ports P4, P7 and P8 are controlled by one bit each while ports P2 and P3 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. The input hysteresis provides stable inputs from noisy or slowly changing external signals (See *Figure 19*).

PICON (F1C4h / E2h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P8LIN	P7LIN	-	P4LIN	P3HIN	P3LIN	P2HIN	P2LIN
								RW	RW		RW	RW	RW	RW	RW

PxLIN	**Port x Low Byte Input Level Selection** 0: Pins Px.7...Px.0 switch on standard TTL input levels 1: Pins Px.7...Px.0 switch on special threshold input levels
PxHIN	**Port x High Byte Input Level Selection** 0: Pins Px.15...Px.8 switch on standard TTL input levels 1: Pins Px.15...Px.8 switch on special threshold input levels

Figure 18. Output Drivers in Push-pull Mode and in Open Drain Mode

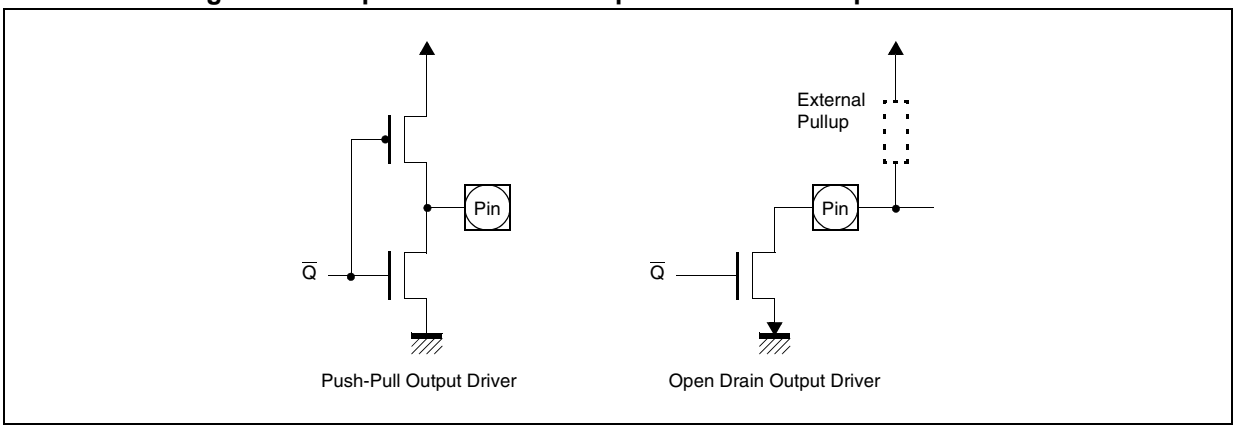
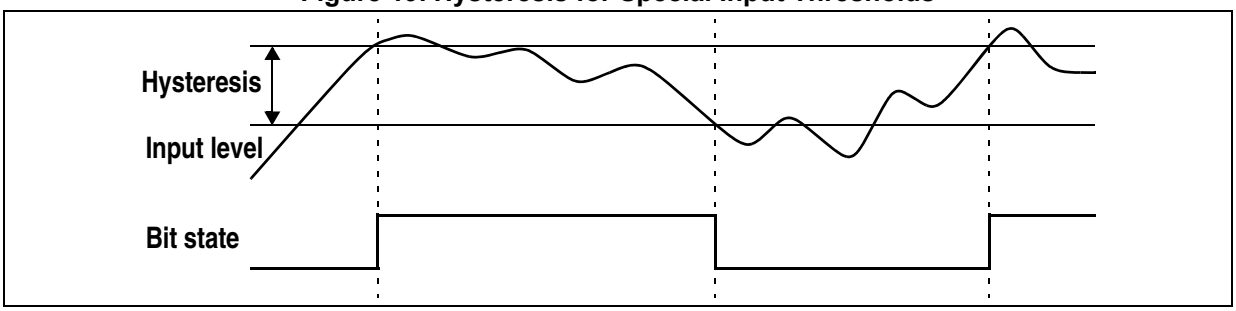


Figure 19. Hysteresis for Special Input Thresholds



12.2.3 Output Driver Control

The port output control registers POCONx allow to select the port output driver characteristics of a port. The aim of these selections is to adapt the output drivers to the application's requirements, and to improve the EMI behavior of the device. Two characteristics may be selected:

Edge characteristic defines the rise/fall time for the respective output. Slow edges reduce the peak currents that are sinked/sourced when changing the voltage level of an external capacitive load. For a bus interface or pins that are changing at frequency higher than 1MHz, however, fast edges may still be required.

Driver characteristic defines either the general driving capability of the respective driver, or if the driver strength is reduced after the target output level has been reached or not. Reducing the driver strength increases the output's internal resistance, which attenuates noise that is imported via the output line. For driving LEDs or power transistors, however, a stable high output current may still be required as described below.

This rise / fall time of 4 I/O pads (a nibble) is selected using 2-bit named PNxEC. That means Port Nibble (x = nibble number, it could be 3 as for Port 2.15 to 2.12) Edge Characteristic.

The sink / source capability of the same 4 I/O pads is selected using 2-bit named PNxDC. That means Port Nibble (x = nibble number) Drive Characteristic (See *Table 20*).

POCONx (F0yyh / zzh) for 8-bit Ports

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	PN1DC		PN1EC		PN0DC		PN0EC	
								RW		RW		RW		RW	

POCONx (F0yyh / zzh) for 16-bit Ports

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PN3DC		PN3EC		PN2DC		PN2EC		PN1DC		PN1EC		PN0DC		PN0EC	
RW		RW		RW		RW		RW		RW		RW		RW	

PNxEC	**Port Nibble x Edge Characteristic (rise/fall time)** 00: Fast edge mode, rise/fall times depend on the size of the driver. 01: Slow edge mode, rise/fall times ~60 ns 10: Reserved 11: Reserved
PNxDC	**Port Nibble x Driver Characteristic (output current)** 00: High Current mode: Driver always operates with maximum strength. 01: Dynamic Current mode: Driver strength is reduced after the target level has been reached. 10: Low Current mode: Driver always operates with reduced strength. 11: Reserved

Note: In case of reading an 8 bit P0CONX register, high Byte (bit 15..8) is read as 00h

Table 20 lists the defined POCON registers and the allocation of control bit-fields and port pins.

Table 20. Port Control Register Allocation

Control Register	Physical Address	8-bit Address	Controlled Port Nibble			
			3	2	1	0
POCON0L	F080h	40h			P0L.7...4	P0L.3...0
POCON0H	F082h	41h			P0H.7...4	P0H.3...0
POCON1L	F084h	42h			P1L.7...4	P1L.3...0
POCON1H	F086h	43h			P1H.7...4	P1H.3...0
POCON2	F088h	44h	P2.15...12	P2.11...8	P2.7...4	P2.3...0
POCON3	F08Ah	45h	P3.15, 3.13, 3.12	P3.11...8	P3.7...4	P3.3...0
POCON4	F08Ch	46h			P4.7...4	P4.3...0
POCON6	F08Eh	47h			P6.7...4	P6.3...0
POCON7	F090h	48h			P7.7...4	P7.3...0
POCON8	F092h	49h			P8.7...4	P8.3...0

Dedicated Pins Output Control

Programmable pad drivers also are supported for the dedicated pins ALE, RD and WR. For these pads, a special POCON20 register is provided.

POCON20 (F0AAh / 55h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	PN1DC		PN1EC		PN0DC		PN0EC	
								RW		RW		RW		RW	

PN0EC	**$\overline{RD}$, $\overline{WR}$ Edge Characteristic (rise/fall time)** 00: Fast edge mode, rise/fall times depend on the size of the driver. 01: Slow edge mode, rise/fall times ~60 ns 10: Reserved 11: Reserved
PN0DC	**$\overline{RD}$, $\overline{WR}$ Driver Characteristic (output current)** 00: High Current mode: Driver always operates with maximum strength. 01: Dynamic Current mode: Driver strength is reduced after the target level has been reached. 10: Low Current mode: Driver always operates with reduced strength. 11: Reserved
PN1EC	**ALE Edge Characteristic (rise/fall time)** 00: Fast edge mode, rise/fall times depend on the size of the driver. 01: Slow edge mode, rise/fall times ~60 ns 10: Reserved 11: Reserved
PN1DC	**ALE Driver Characteristic (output current)** 00: High Current mode: Driver always operates with maximum strength. 01: Dynamic Current mode: Driver strength is reduced after the target level has been reached. 10: Low Current mode: Driver always operates with reduced strength. 11: Reserved

12.2.4 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.
- 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 PWM module.
  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 BHE and the system clock output (CLKOUT).
- Port 4 outputs the additional segment address bit A16 to A23 in systems where segmentation is enabled to access more than 64 Kbytes of memory.
- 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.

If an 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 an 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, one can also set the direction for this pin 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 application software must set 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 supporting only one alternate input function. Port lines with only one alternate output function, however, have different structures. It has to be adapted to support the normal and the alternate function features.

All port lines that are not used for these alternate functions may be used as general purpose I/O lines. When using port pins for general purpose output, the initial output value should be written to the port latch prior to enabling the output drivers, in order to avoid undesired transitions on the output pins. This applies to single pins as well as to pin groups (see examples below).

```
SINGLE_BIT:BSETP4.7; Initial output level is "high"
   BSETDP4.7; Switch on the output driver
BIT_GROUP:BFLDHP4, #24H, #24H; Initial output level is "high"
   BFLDHDP4, #24H, #24H; Switch on the output drivers
```

Note: When using several BSET pairs to control more pins of one port, these pairs must be separated by instructions, which do not apply to the respective port (See Section 6: Central Processing Unit (CPU).

12.3 PORT0

The two 8-bit ports P0H and P0L represent the higher and lower part of PORT0, respectively. Both halves of PORT0 can be written (via a PEC transfer) without effecting the other half.

If this port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction registers DP0H and DP0L.

P0L (FF00h / 80h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P0L.7	P0L.6	P0L.5	P0L.4	P0L.3	P0L.2	P0L.1	P0L.0
								RW	RW	RW	RW	RW	RW	RW	RW

P0H (FF02h / 81h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P0H.7	P0H.6	P0H.5	P0H.4	P0H.3	P0H.2	P0H.1	P0H.0
								RW	RW	RW	RW	RW	RW	RW	RW

P0X.y	**Port Data Register P0H or P0L Bit y**

DP0L (F100h / 80h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP0L.7	DP0L.6	DP0L.5	DP0L.4	DP0L.3	DP0L.2	DP0L.1	DP0L.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP0H (F102h / 81h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP0H.7	DP0H.6	DP0H.5	DP0H.4	DP0H.3	DP0H.2	DP0H.1	DP0H.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP0X.y	**Port Direction Register DP0H or DP0L Bit y** DP0X.y = 0: Port line P0X.y is an input (high-impedance) DP0X.y = 1: Port line P0X.y is an output

12.3.1 Alternate Functions of PORT0

When an external bus is enabled, PORT0 is used as data bus or address/data bus. Note that an external 8-bit demultiplexed bus only uses P0L, while P0H is free for I/O (provided that no other bus mode is enabled).

PORT0 is also used to select the system start-up configuration. During reset, PORT0 is configured to input, and each line is held high through an internal pull-up device.

Each line can now be individually pulled to a low level (see *Section 21.3: DC Characteristics*) through an external pull-down device. A default configuration is selected when the respective PORT0 lines are at a high level. Through pulling individual lines to a low level, this default can be changed according to the needs of the applications.

The internal pull-up devices are designed in such way that an external pull-down resistors (see Data Sheet specification) can be used to apply a correct low level.

These external pull-down resistors can remain connected to the PORT0 pins also during normal operation, however, care has to be taken in order to not disturb the normal function of PORT0 (this might be the case, for example, if the external resistor value is too low). With the end of reset, the selected bus configuration will be written to the BUSCON0 register.

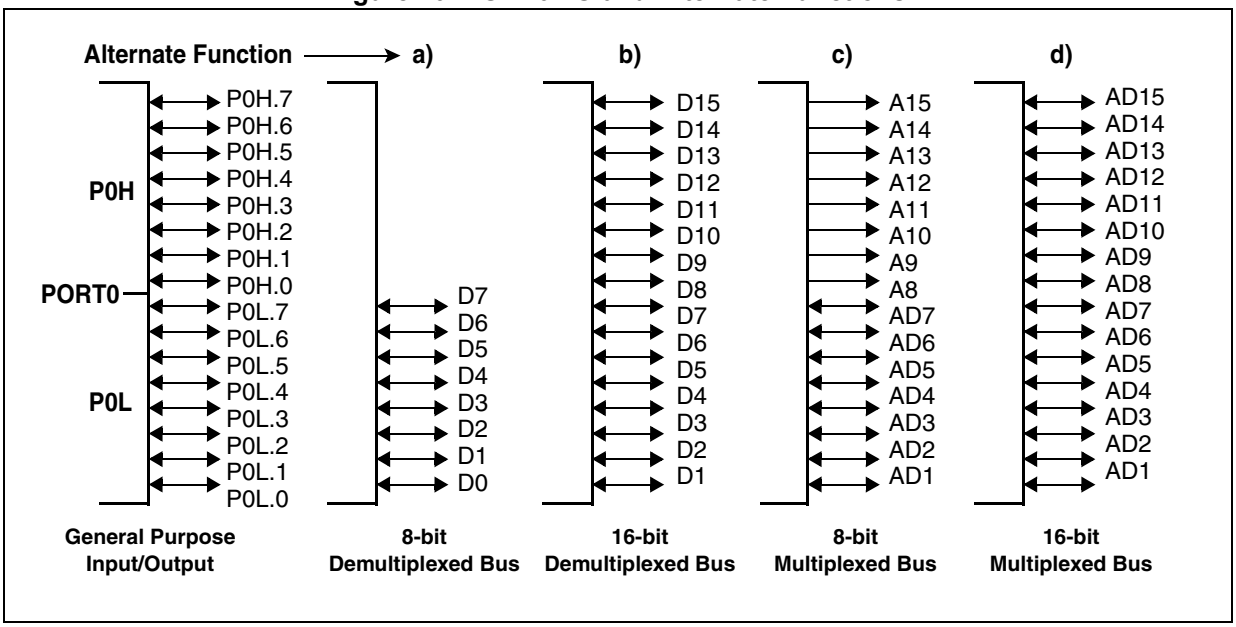
The configuration of the high byte of PORT0, will be copied into the special register RP0H. This read-only register holds the selection for the number of chip selects and segment addresses. Software can read this register in order to react according to the selected configuration, if required.

When the reset is terminated, the internal pull-up devices are switched off, and PORT0 will be switched to the appropriate operating mode.

During external accesses in multiplexed bus modes PORT0 first outputs the 16-bit intra-segment address as an alternate output function. PORT0 is then switched to high-impedance input mode to read the incoming instruction or data. In 8-bit data bus mode, two memory cycles are required for word accesses, the first for the low Byte and the second for the high Byte of the Word.

During write cycles PORT0 outputs the data Byte or Word after outputting the address. During external accesses in demultiplexed bus modes PORT0 reads the incoming instruction or data Word or outputs the data Byte or Word.

Figure 20. PORT0 I/O and Alternate Functions



When an external bus mode is enabled, the direction of the port pin and the loading of data into the port output latch are controlled by the bus controller hardware.

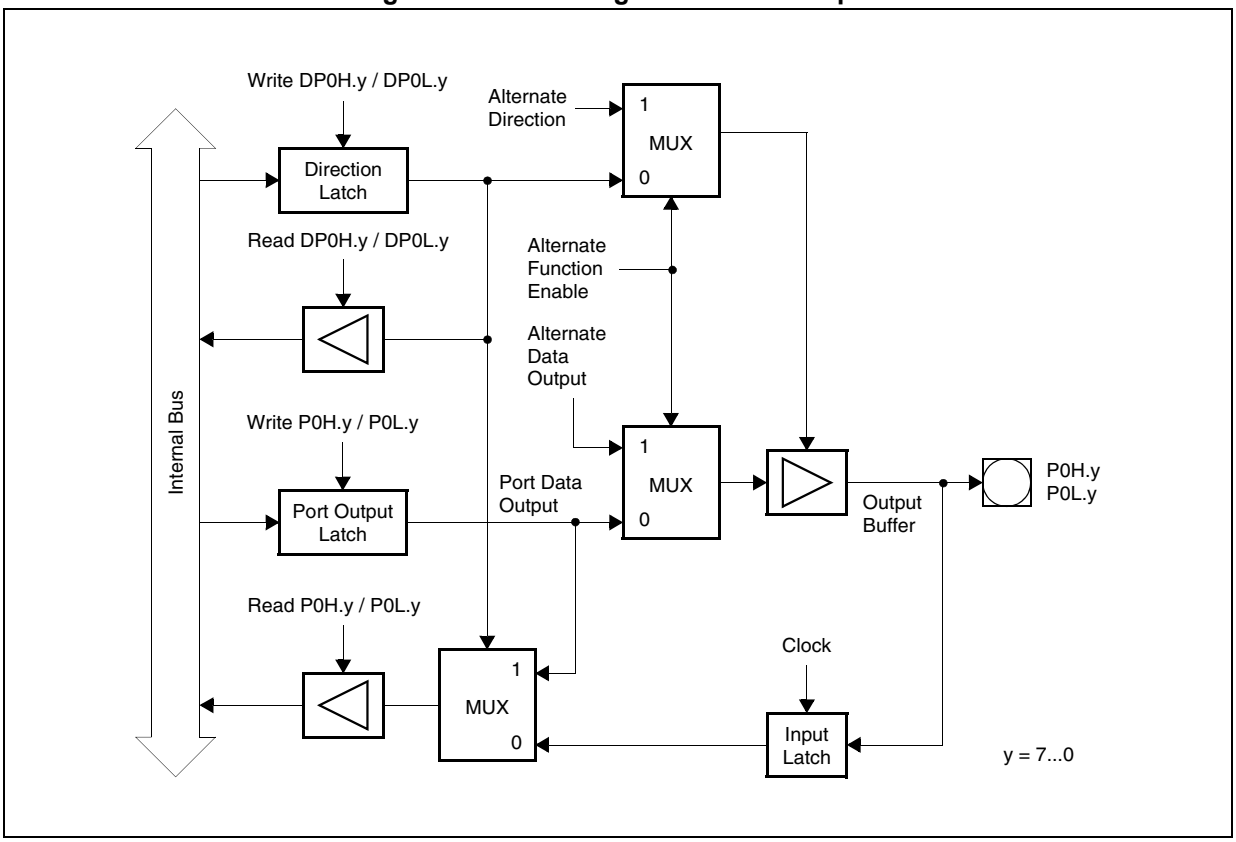
The input of the port output Buffer is disconnected from the internal bus and is switched to the line labeled “Alternate Data Output” via a multiplexer.

The alternate data can be the 16-bit intra-segment address or the 8/16-bit data information. The incoming data on PORT0 is read on the line “Alternate Data Input”. While an external bus mode is enabled, the user software should not write to the port output latch, otherwise unpredictable results may occur.

When the external bus modes are disabled, the contents of the direction register last written by the user becomes active.

Figure 21 shows the structure of a PORT0 pin.

Figure 21. Block Diagram of a PORT0 pin



12.4 PORT1

The two 8-bit ports P1H and P1L represent the higher and lower part of PORT1, respectively. Both halves of PORT1 can be written (via a PEC transfer) without effecting the other half.

If this port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction registers DP1H and DP1L.

P1L (FF04h / 82h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P1L.7	P1L.6	P1L.5	P1L4	P1L.3	P1L.2	P1L.1	P1L.0
								RW	RW	RW	RW	RW	RW	RW	RW

P1H (FF06h / 83h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P1H.7	P1H.6	P1H.5	P1H.4	P1H.3	P1H.2	P1H.1	P1H.0
								RW	RW	RW	RW	RW	RW	RW	RW

P1X.y	**Port Data Register P1H or P1L Bit y**

DP1L (F104h / 82h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP1L.7	DP1L.6	DP1L.5	DP1L.4	DP1L.3	DP1L.2	DP1L.1	DP1L.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP1H (F106h / 83h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP1H.7	DP1H.6	DP1H.5	DP1H.4	DP1H.3	DP1H.2	DP1H.1	DP1H.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP1X.y	**Port Direction Register DP1H or DP1L Bit y** DP1X.y = 0: Port line P1X.y is an input (high-impedance) DP1X.y = 1: Port line P1X.y is an output

12.4.1 Alternate Functions of PORT1

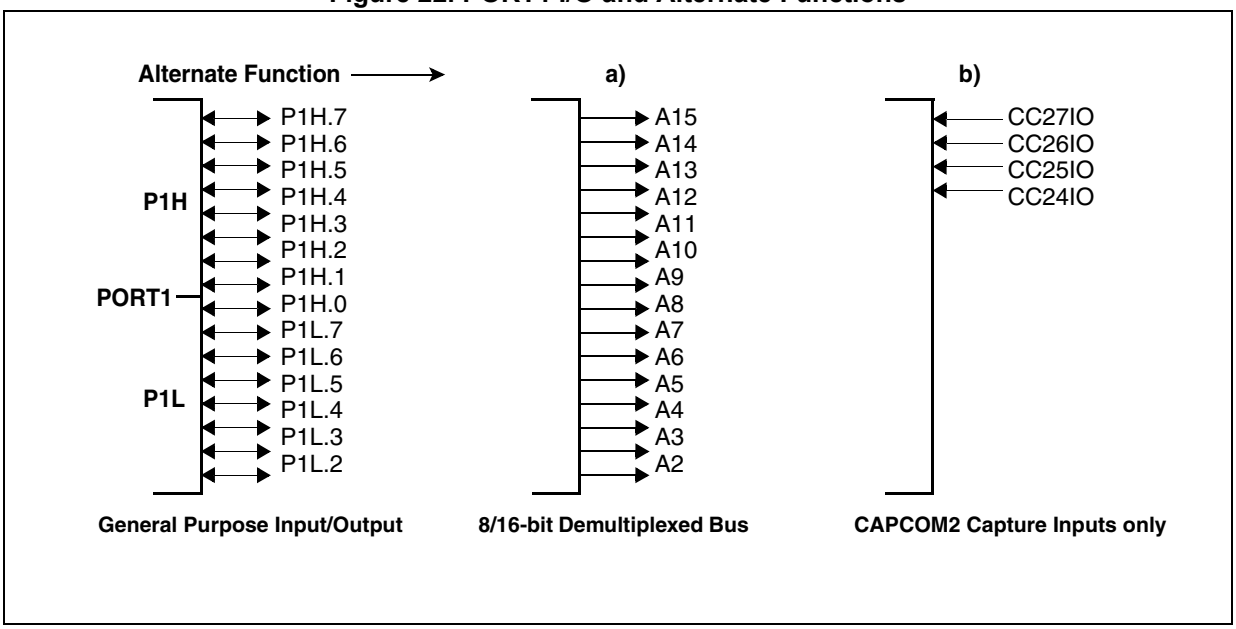
When a demultiplexed external bus is enabled, PORT1 is used as address bus.
Note: Demultiplexed bus modes use PORT1 as a 16-bit port. Otherwise all 16 port lines can be used for general purpose I/O.

The upper 4 pins of PORT1 (P1H.7...P1H.4) are used as capture input lines (CC27IO...CC24IO).

During external accesses in demultiplexed bus modes PORT1 outputs the 16-bit intra-segment address as an alternate output function.

During external accesses in multiplexed bus modes, when no BUSCON register selects a demultiplexed bus mode, PORT1 is not used and is available for general purpose I/O.

Figure 22. PORT1 I/O and Alternate Functions

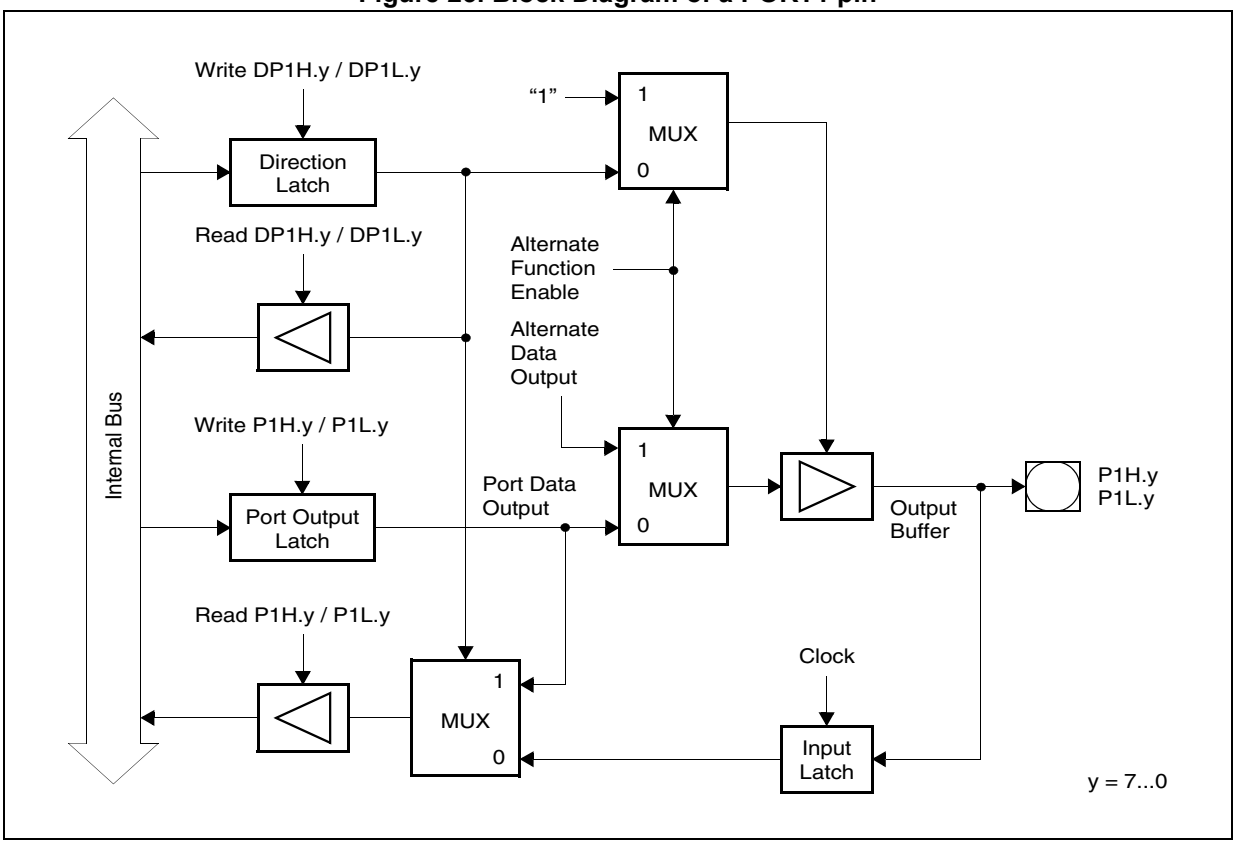


When an external bus mode is enabled, the direction of the port pin and the loading of data into the port output latch are controlled by the bus controller hardware.

The input of the port Buffer latch is disconnected from the internal bus and is switched to the line labeled “Alternate Data Output” via a multiplexer. The alternate data is the 16-bit intra-segment address. While an external bus mode is enabled, the user software should not write to the port output latch, otherwise unpredictable results may occur. When the external bus modes are disabled, the contents of the direction register last written by the user becomes active.

Figure 23 shows the structure of a PORT1 pin.

Figure 23. Block Diagram of a PORT1 pin



12.5 Port 2

If this 16-bit port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction register DP2. Each port line can be switched into push/pull or open drain mode via the open drain control register ODP2.

P2 (FFC0h / E0h)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P2.15	P2.14	P2.13	P2.12	P2.11	P2.10	P2.9	P2.8	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

P2.y	**Port Data Register P2 Bit y**

DP2 (FFC2h / E1h)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP2.15	DP2.14	DP2.13	DP2.12	DP2.11	DP2.10	DP2.9	DP2.8	DP2.7	DP2.6	DP2.5	DP2.4	DP2.3	DP2.2	DP2.1	DP2.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

DP2.y	**Port Direction Register DP2 Bit y** DP2.y = 0: Port line P2.y is an input (high-impedance) DP2.y = 1: Port line P2.y is an output

ODP2 (F1C2h / E1h)

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODP2 .15	ODP2 .14	ODP2 .13	ODP2 .12	ODP2 .11	ODP2 .10	ODP2 .9	ODP2 .8	ODP2 .7	ODP2 .6	ODP2 .5	ODP2 .4	ODP2 .3	ODP2 .2	ODP2 .1	ODP2 .0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ODP2.y	**Port 2 Open Drain Control Register Bit y** ODP2.y = 0: Port line P2.y output driver in push/pull mode ODP2.y = 1: Port line P2.y output driver in open drain mode

12.5.1 Alternate Functions of Port 2

All Port 2 lines (P2.15...P2.0) serve as capture inputs or compare outputs (CC15IO...CC0IO) for the CAPCOM1 unit.

When a Port 2 line is used as a capture input, the state of the input latch, which represents the state of the port pin, is directed to the CAPCOM unit via the line “Alternate Pin Data Input”. If an external capture trigger signal is used, the direction of the respective pin must be set to input.

If the direction is set to output, the state of the port output latch will be read since the pin represents the state of the output latch.

This can be used to trigger a capture event through software by setting or clearing the port latch. Note that in the output configuration, no external device may drive the pin, otherwise conflicts would occur.

When a Port 2 line is used as a compare output (compare modes 1 and 3), the compare event (or the timer overflow in compare mode 3) directly effects the port output latch. In compare mode 1, when a valid compare match occurs, the state of the port output latch is read by the CAPCOM control hardware via the line “Alternate Latch Data Input”, inverted, and written back to the latch via the line “Alternate Data Output”.

The port output latch is clocked by the signal “Compare Trigger” which is generated by the CAPCOM unit. In compare mode 3, when a match occurs, the value '1' is written to the port output latch via the line “Alternate Data Output”. When an overflow of the corresponding timer occurs, a '0' is written to the port output latch. In both cases, the output latch is clocked by the signal “Compare Trigger”.

The direction of the pin should be set to output by the user, otherwise the pin will be in the high-impedance state and will not reflect the state of the output latch.

As can be seen from the port structure in *Figure 25*, the user software always has free access to the port pin even when it is used as a compare output. This is useful for setting up the initial level of the pin when using compare mode 1 or the double-register mode. In these modes, unlike in compare mode 3, the pin is not set to a specific value when a compare match occurs, but is toggled instead.

When the user wants to write to the port pin at the same time a compare trigger tries to clock the output latch, the write operation of the user software has priority. Each time a CPU write access to the port output latch occurs, the input multiplexer of the port output latch is switched to the line connected to the internal bus. The port output latch will receive the value from the internal bus and the hardware triggered change will be lost.

As all other capture inputs, the capture input function of pins P2.15...P2.0 can also be used as external interrupt inputs (200 ns sample rate at 40 MHz CPU clock and 250 ns sample rate at 32 MHz CPU clock).

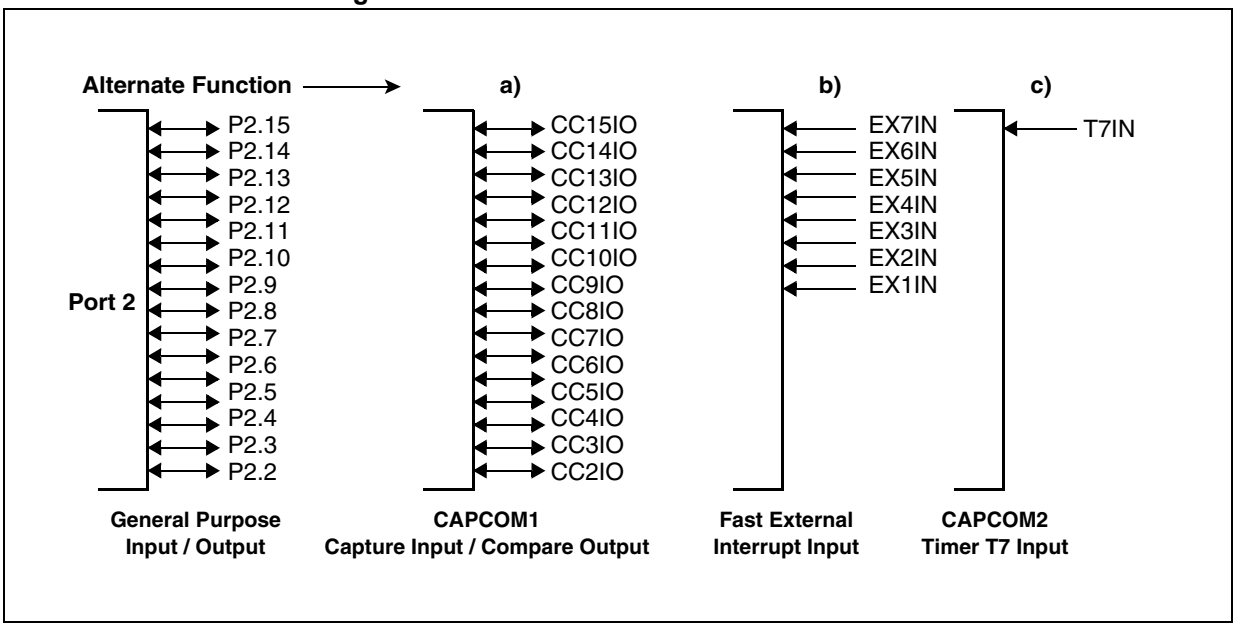
The upper eight Port 2 lines (P2.15...P2.8) also can serve as Fast External Interrupt inputs from EX0IN to EX7IN (Fast external interrupt sampling rate is 25 ns at 40 MHz CPU clock and 31.25 ns at 32 MHz CPU clock).

P2.15 in addition serves as input for CAPCOM2 timer T7 (T7IN). *Table 21* summarizes the alternate functions of Port 2.

Table 21. Alternate Functions of Port 2

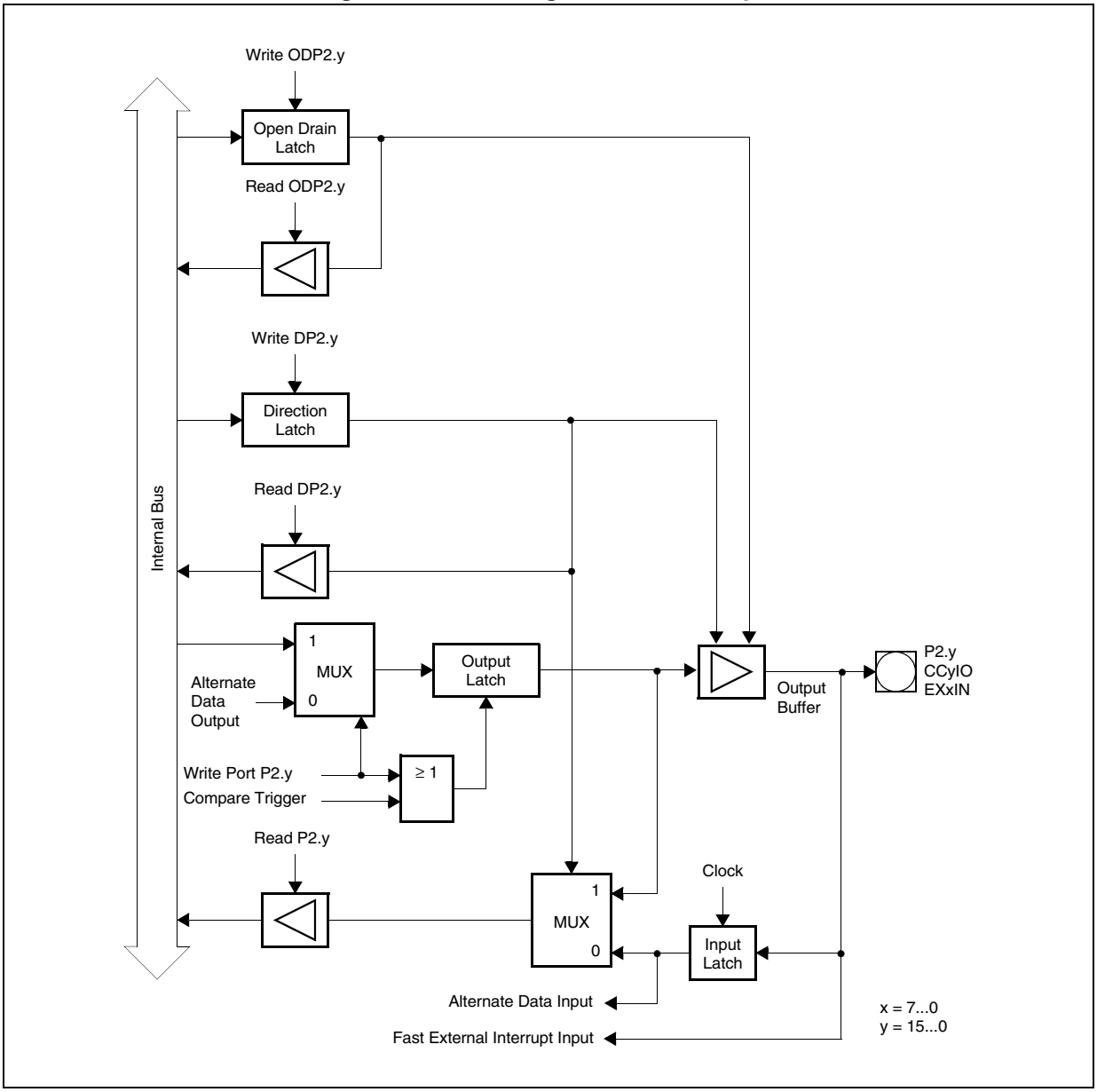
Port 2 Pin	Alternate Function a)	Alternate Function b)	Alternate Function c)
P2.0	CC0IO	-	-
P2.1	CC1IO	-	-
P2.2	CC2IO	-	-
P2.3	CC3IO	-	-
P2.4	CC4IO	-	-
P2.5	CC5IO	-	-
P2.6	CC6IO	-	-
P2.7	CC7IO	-	-
P2.8	CC8IO	EX0IN Fast External Interrupt 0 Input	-
P2.9	CC9IO	EX1IN Fast External Interrupt 1 Input	-
P2.10	CC10IO	EX2IN Fast External Interrupt 2 Input	-
P2.11	CC11IO	EX3IN Fast External Interrupt 3 Input	-
P2.12	CC12IO	EX4IN Fast External Interrupt 4 Input	-
P2.13	CC13IO	EX5IN Fast External Interrupt 5 Input	-
P2.14	CC14IO	EX6IN Fast External Interrupt 6 Input	-
P2.15	CC15IO	EX7IN Fast External Interrupt 7 Input	T7IN T7 External Count Input

Figure 24. PORT2 I/O and Alternate Functions



The pins of Port 2 combine internal bus data with alternate data output before the port latch input.

Figure 25. Block Diagram of a PORT2 pin



12.6 Port 3

If this 15-bit port is used for general purpose I/O, the direction of each line can be configured by the corresponding direction register DP3. Most port lines can be switched into push-pull

or open drain mode by the open drain control register ODP2 (pins P3.15, P3.14 and P3.12 do not support open drain mode).

Due to pin limitations register bit P3.14 is not connected to an output pin.

P3 (FFC4h / E2h)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P3.15	-	P3.13	P3.12	P3.11	P3.10	P3.9	P3.8	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

P3.y	**Port Data Register P3 Bit y**

DP3 (FFC6h / E3h)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP3 .15	-	DP3 .13	DP3 .12	DP3 .11	DP3 .10	DP3 .9	DP3 .8	DP3 .7	DP3 .6	DP3 .5	DP3 .4	DP3 .3	DP3 .2	DP3 .1	DP3 .0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

DP3.y	**Port Direction Register DP3 Bit y** DP3.y = 0: Port line P3.y is an input (high-impedance) DP3.y = 1: Port line P3.y is an output

ODP3 (F1C6h / E3h)

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	ODP3 .13	-	ODP3 .11	ODP3 .10	ODP3 .9	ODP3 .8	ODP3 .7	ODP3 .6	ODP3 .5	ODP3 .4	ODP3 .3	ODP3 .2	ODP3 .1	ODP3 .0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ODP3.y	**Port 3 Open Drain Control Register Bit y** ODP3.y = 0: Port line P3.y output driver in push-pull mode ODP3.y = 1: Port line P3.y output driver in open drain mode

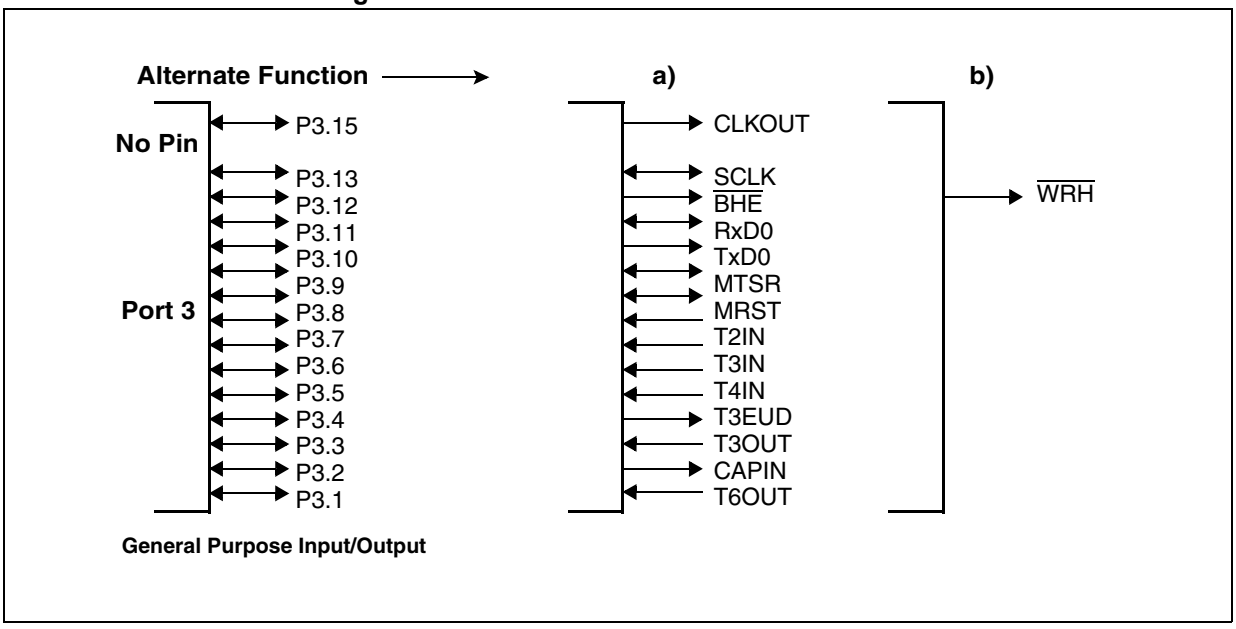
12.6.1 Alternate Functions of Port 3

The pins of Port 3 serve for various functions which include external timer control lines, the two serial interfaces and the control lines BHE/WRH and CLKOUT.

Table 22. Port 3 Alternative Functions

Port 3 Pin	Alternate Function
P3.0	T0INCAPCOM1 Timer 0 Count Input
P3.1	T6OUTTimer 6 Toggle Output
P3.2	CAPINGPT2 Capture Input
P3.3	T3OUTTimer 3 Toggle Output
P3.4	T3EUDTimer 3 External Up/Down Input
P3.5	T4INTimer 4 Count Input
P3.6	T3INTimer 3 Count Input
P3.7	T2INTimer 2 Count Input
P3.8	MRSTSSC Master Receive / Slave Transmit
P3.9	MTSRSSC Master Transmit / Slave Receive
P3.10	TxD0ASC0 Transmit Data Output
P3.11	RxD0ASC0 Receive Data Input (Output in synchronous mode)
P3.12	$\overline{BHE}$/$\overline{WRH}$Byte High Enable / Write High Output
P3.13	SCLKSSC Shift Clock Input/Output
P3.14	---No pin assigned
P3.15	CLKOUTSystem Clock Output

Figure 26. PORT3 I/O and Alternate Functions



The structure of the Port 3 pins depends on their alternate function (see *Figure 27* and *Figure 28*). When the on-chip peripheral associated with a Port 3 pin is configured to use the alternate input function, it reads the input latch, which represents the state of the pin, via the line labeled “Alternate Data Input”. Port 3 pins with alternate input functions are: T0IN, T2IN, T3IN, T4IN, T3EUD and CAPIN.

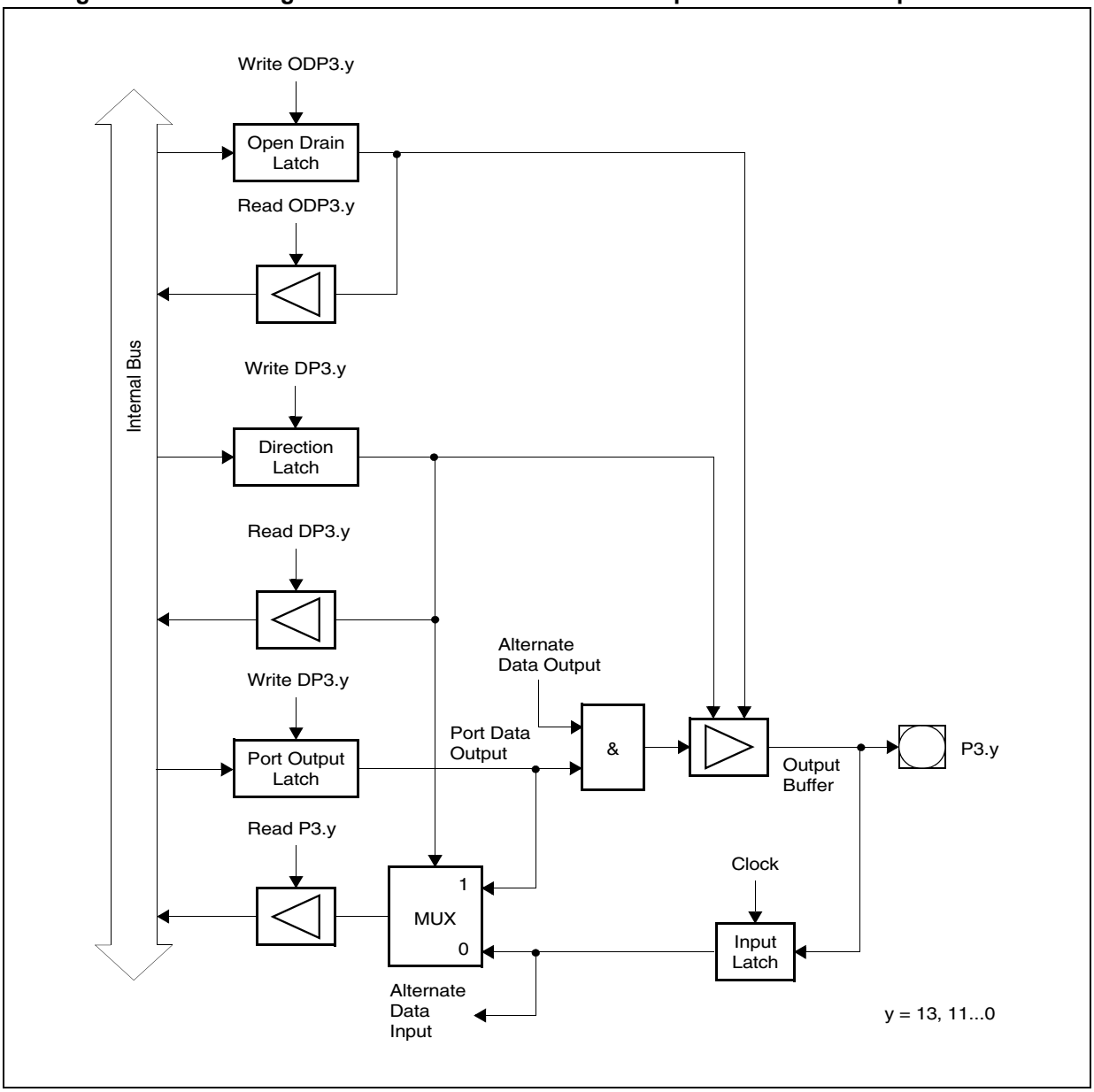
When the on-chip peripheral associated with a Port 3 pin is configured to use the alternate output function, its “Alternate Data Output” line is ANDed with the port output latch line. When using these alternate functions, the user must set the direction of the port line to output (DP3.y=1) and must set the port output latch (P3.y=1). Otherwise the pin is in its high-impedance state (when configured as input) or the pin is stuck at '0' (when the port output latch is cleared). When the alternate output functions are not used, the “Alternate Data Output” line is in its inactive state, which is a high level ('1').

Port 3 pins with alternate output functions are: T6OUT, T3OUT, TxD0, BHE and CLKOUT.

When the on-chip peripheral associated with a Port 3 pin is configured to use both the alternate input and output function, the descriptions above apply to the respective current operating mode. The direction must be set accordingly. Port 3 pins with alternate input/output functions are: MTSR, MRST, RxD0 and SCLK.

Note: Enabling the CLKOUT function automatically enables the P3.15 output driver. Setting bit DP3.15=’1’ is not required.

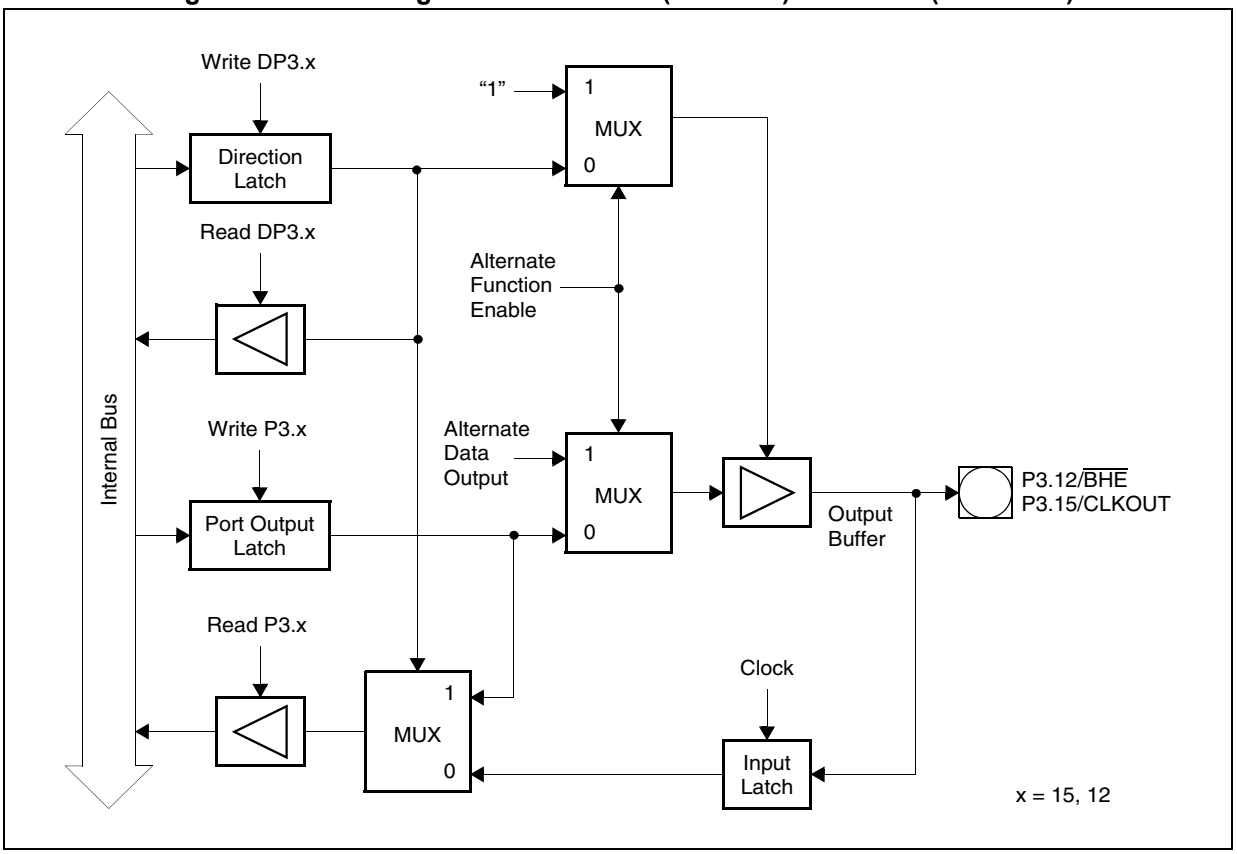
Figure 27. Block Diagram of Port 3 Pin with Alternate Input or Alternate Output Function



Pin P3.12 ($\overline{BHE}$/$\overline{WRH}$) is another pin with an alternate output function, however, its structure is slightly different.

After reset the $\overline{BHE}$ or $\overline{WRH}$ function must be used depending on the system start-up configuration. In either of these cases, there is no possibility to program any port latches before. Thus, the appropriate alternate function is selected automatically. If $\overline{BHE}$/$\overline{WRH}$ is not used in the system, this pin can be used for general purpose I/O by disabling the alternate function (BYTDIS = '1' / WRCFG='0').

Figure 28. Block Diagram of Pins P3.15 (CLKOUT) and P3.12 (BHE/WRH)



Note: Enabling the BHE or WRH function automatically enables the P3.12 output driver. Setting bit DP3.12='1' is not required.
During bus hold pin P3.12 is switched back to its standard function and is then controlled by DP3.12 and P3.12. Keep DP3.12 = '0' in this case to ensure floating in hold mode.

12.7 Port 4

If this 8-bit port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction register DP4.

P4 (FFC8h / E4h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
								RW	RW	RW	RW	RW	RW	RW	RW

P4.y	**Port Data Register P4 Bit y**

DP4 (FFCAh / E5h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP4.7	DP4.6	DP4.5	DP4.4	DP4.3	DP4.2	DP4.1	DP4.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP4.y	**Port Direction Register DP4 Bit y** DP4.y = 0: Port line P4.y is an input (high-impedance) DP4.y = 1: Port line P4.y is an output

For CAN configuration support (see section 15), Port 4 has an open drain function, controlled with the ODP4 register:

ODP4 (F1CAh / E5h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	ODP4.7	ODP4.6	-	-	-	-	-	-
								RW	RW						

ODP4.y	**Port 4 Open Drain Control Register Bit y** ODP4.y = 0: Port line P4.y output driver in push/pull mode ODP4.y = 1: Port line P4.y output driver in open drain mode if P4.y is not a segment address line output

Note: Only bit 6 and 7 are implemented, all other bit will be read as "0".

12.7.1 Alternate Functions of Port 4

During external bus cycles that use segmentation (address space above 64K Bytes) a number of Port 4 pins may output the segment address lines. The number of pins that is used for segment address output determines the external address space which is directly accessible. The other pins of Port 4 may be used for general purpose I/O. If segment address lines are selected, the alternate function of Port 4 may be necessary to access external memory directly after reset. For this reason Port 4 will be switched to this alternate function automatically.

The number of segment address lines is selected via PORT0 during reset. The selected value can be read from bitfield SALSEL in register RP0H (read only) in order to check the configuration during run time.

The CAN interfaces use 2 or 4 pins of Port 4 to interface each CAN Modules to an external CAN transceiver. In this case the number of possible segment address lines is reduced.

Table 23 summarizes the alternate functions of Port 4 depending on the number of selected segment address lines (coded via bitfield SALSEL)

Table 23. Port 4 Alternate Functions

Port 4	Standard Function SALSEL = 01 64 Kbytes	Alternate Function SALSEL = 11 256 Kbytes	Alternate Function SALSEL = 00 1 Mbyte	Alternate Function SALSEL = 10 16 Mbytes
P4.0	GPIO	Segment Address A16	Segment. Address A16	Segment Address A16
P4.1	GPIO	Segment Address A17	Segment Address A17	Segment Address A17
P4.2	GPIO	GPIO	Segment Address A18	Segment Address A18
P4.3	GPIO	GPIO	Segment Address A19	Segment Address A19
P4.4	GPIO/CAN2_RxD	GPIO/CAN2_RxD	GPIO/CAN2_RxD	Segment Address A20
P4.5	GPIO/CAN1_RxD	GPIO/CAN1_RxD	GPIO/CAN1_RxD	Segment Address A21
P4.6	GPIO/CAN1_TxD	GPIO/CAN1_TxD	GPIO/CAN1_TxD	Segment Address A22
P4.7	GPIO/CAN2_TxD	GPIO/CAN2_TxD	GPIO/CAN2_TxD	Segment Address A23

Figure 29. PORT4 I/O and Alternate Functions

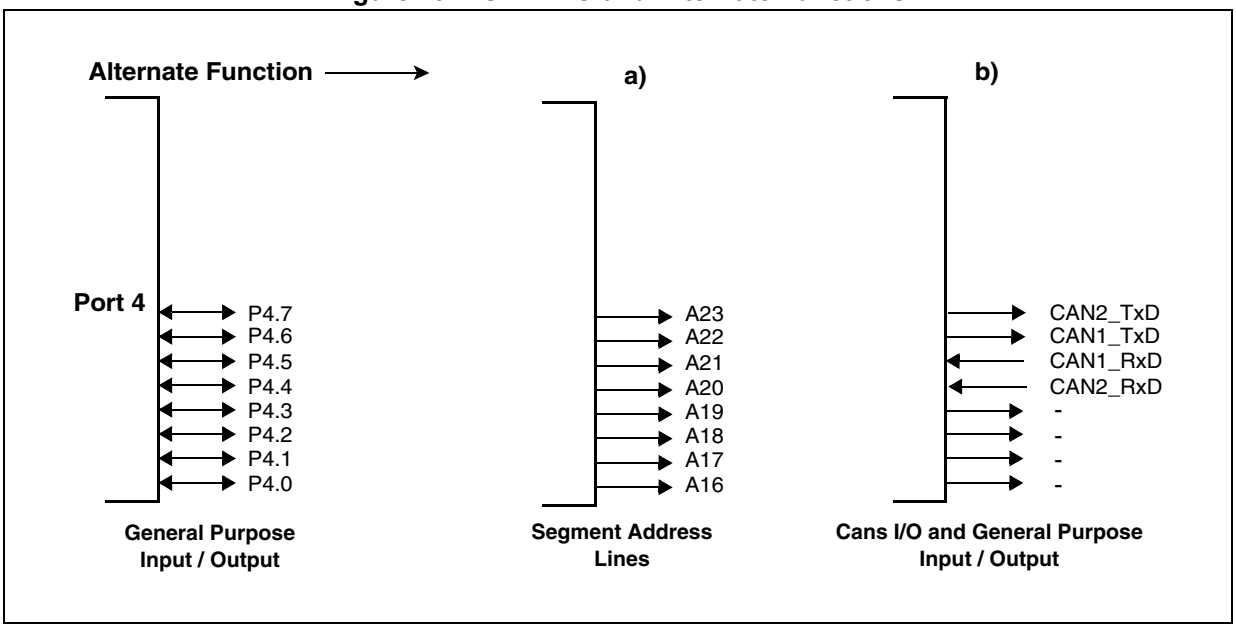


Figure 30. Block Diagram of a Port 4 Pin

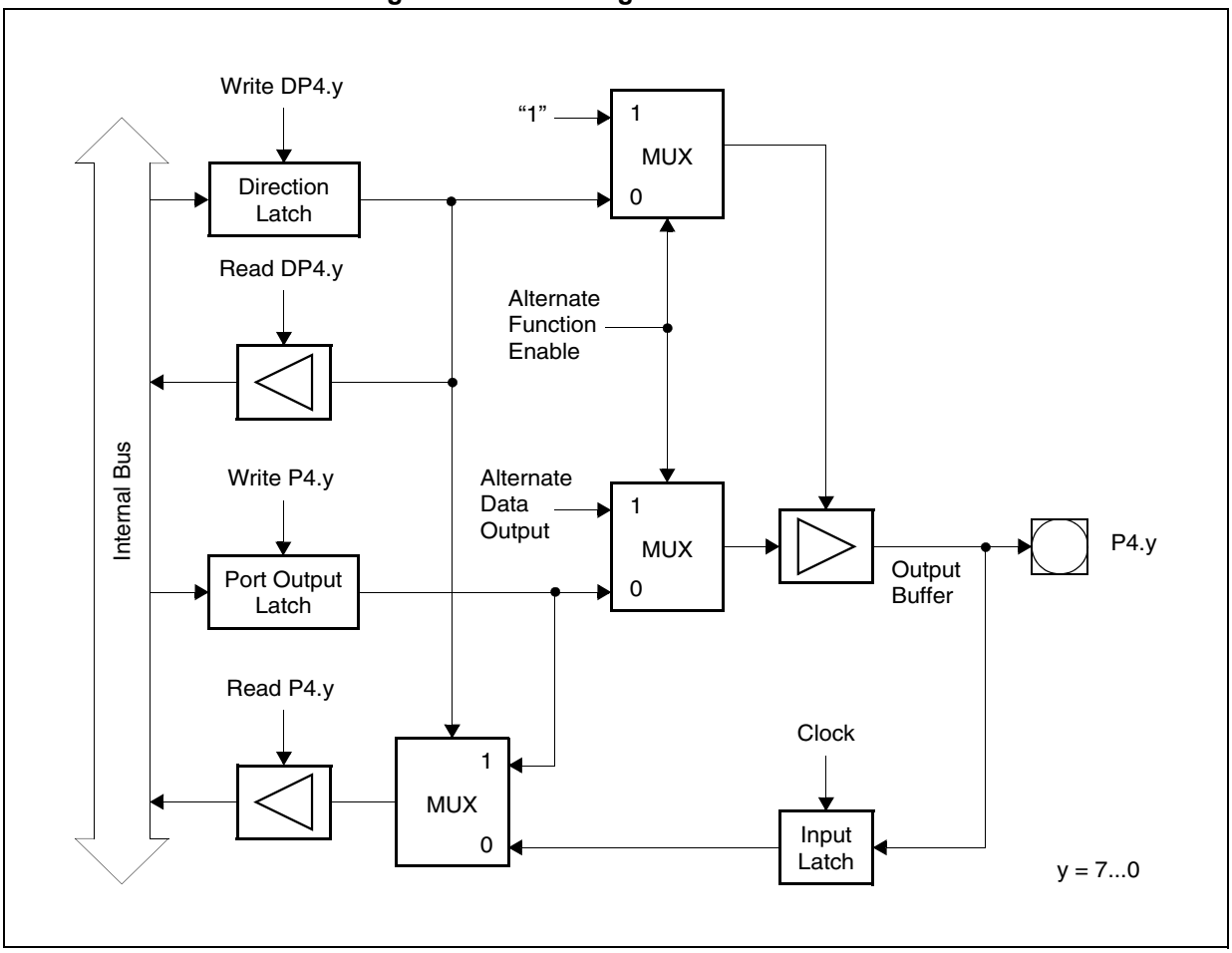


Figure 31. Block Diagram of P4.4 and P4.5 Pins

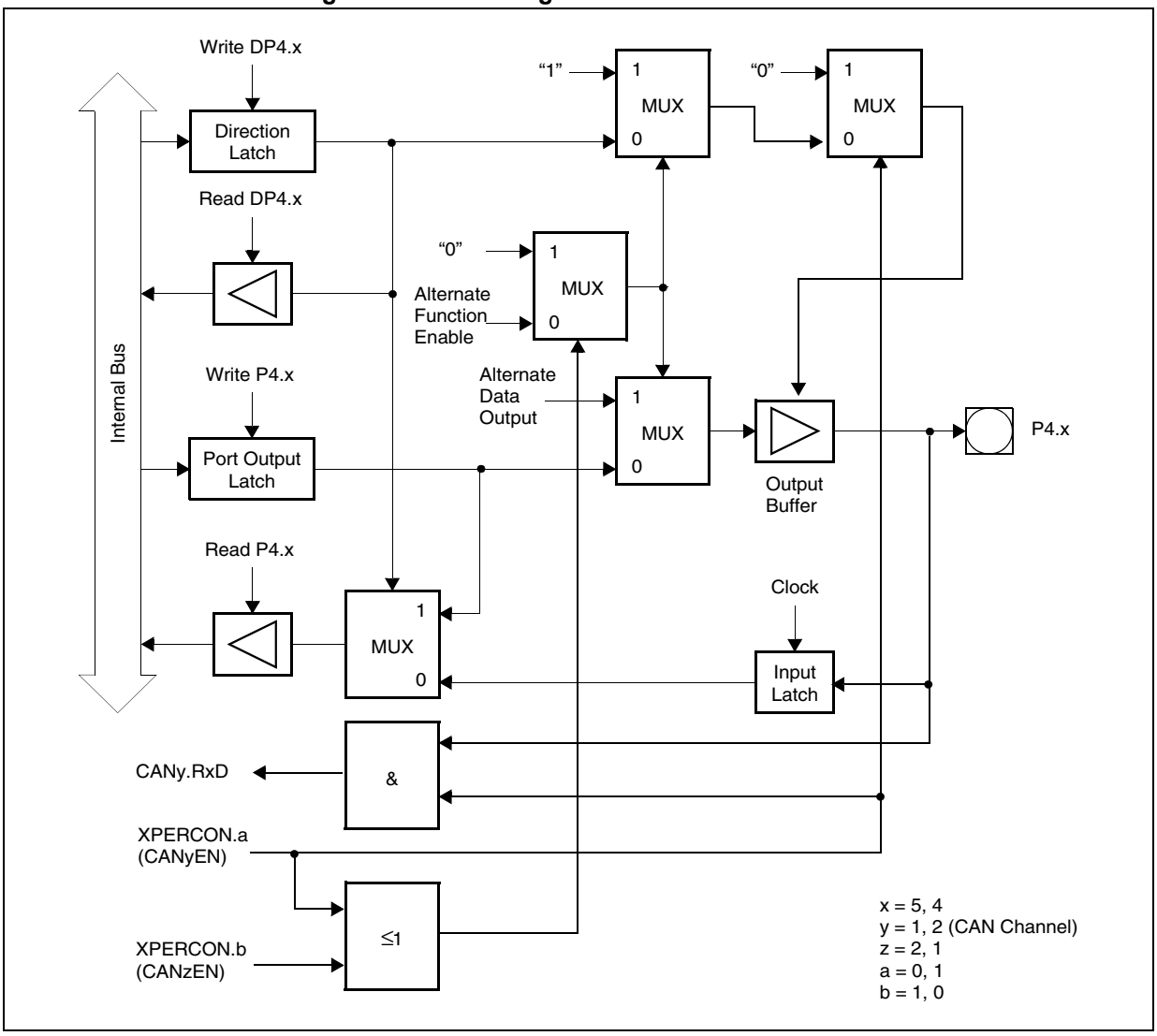
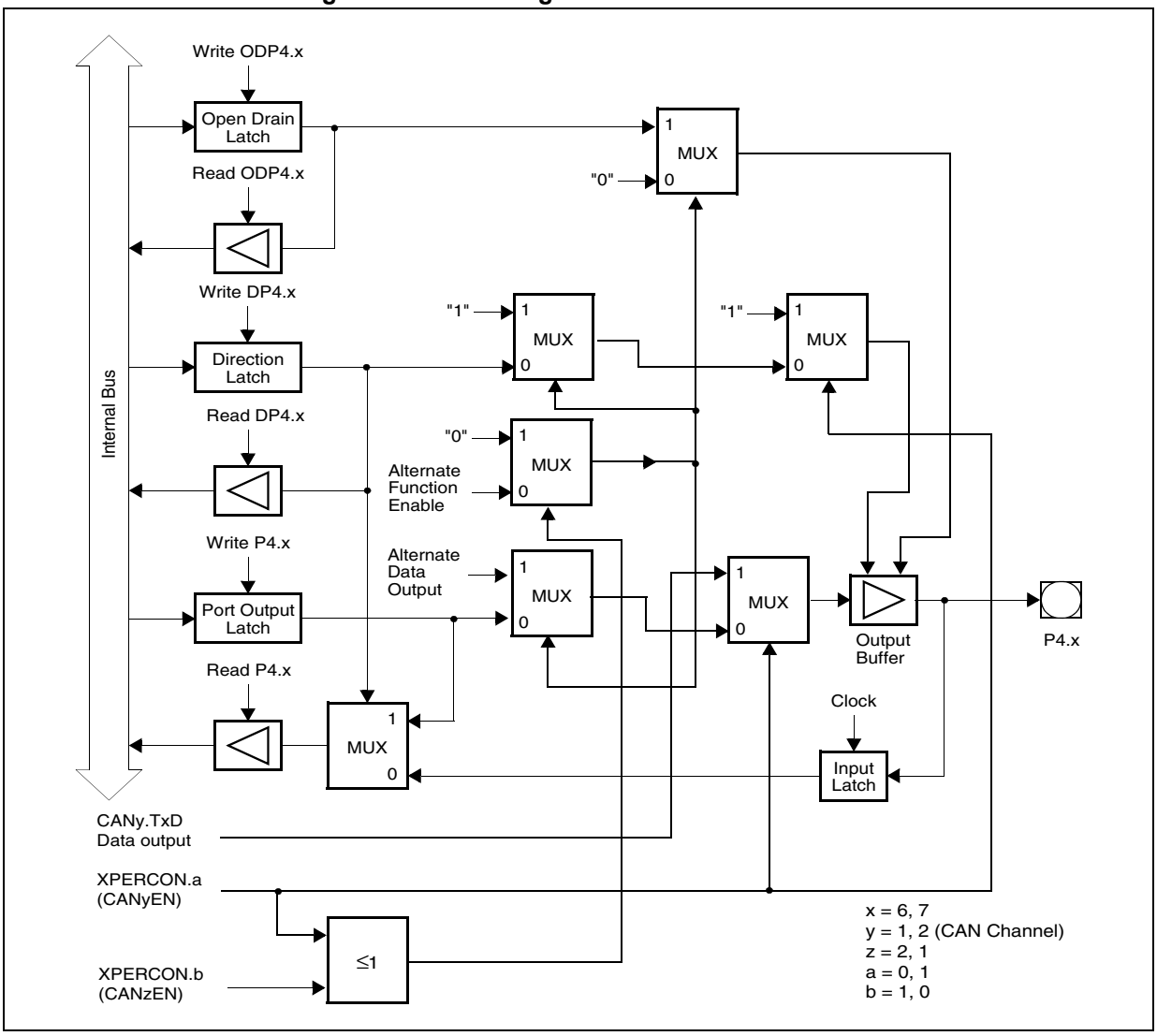


Figure 32. Block Diagram of P4.6 and P4.7 Pins



12.8 Port 5

This 16-bit input port can only read data. There is no output latch and no direction register. Data written to P5 will be lost.

P5 (FFA2h / D1h)

SFR Reset Value: XXXXH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P5.15	P5.14	P5.13	P5.12	P5.11	P5.10	P5.9	P5.8	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

P5.y	**Port Data Register P5 Bit y (Read only)**

12.8.1 Alternate Functions of Port 5

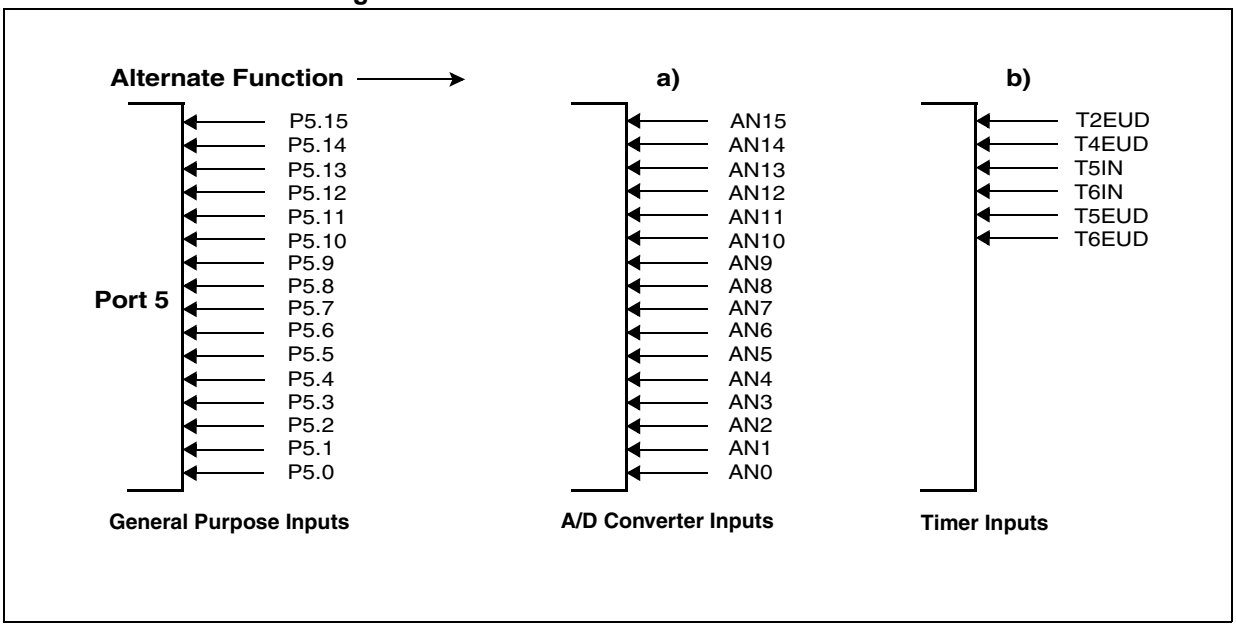
Each line of Port 5 is also connected to one of the multiplexer of the Analog/Digital Converter. All port lines (P5.15...P5.0) can accept analog signals (AN15...AN0) to be converted by the ADC. No special programming is required for pins that shall be used as analog inputs. Some pins of Port 5 also serve as external timer control lines for GPT1 and GPT2.

Table 24 summarizes the alternate functions of Port 5.

Table 24. Port 5 Alternate Functions

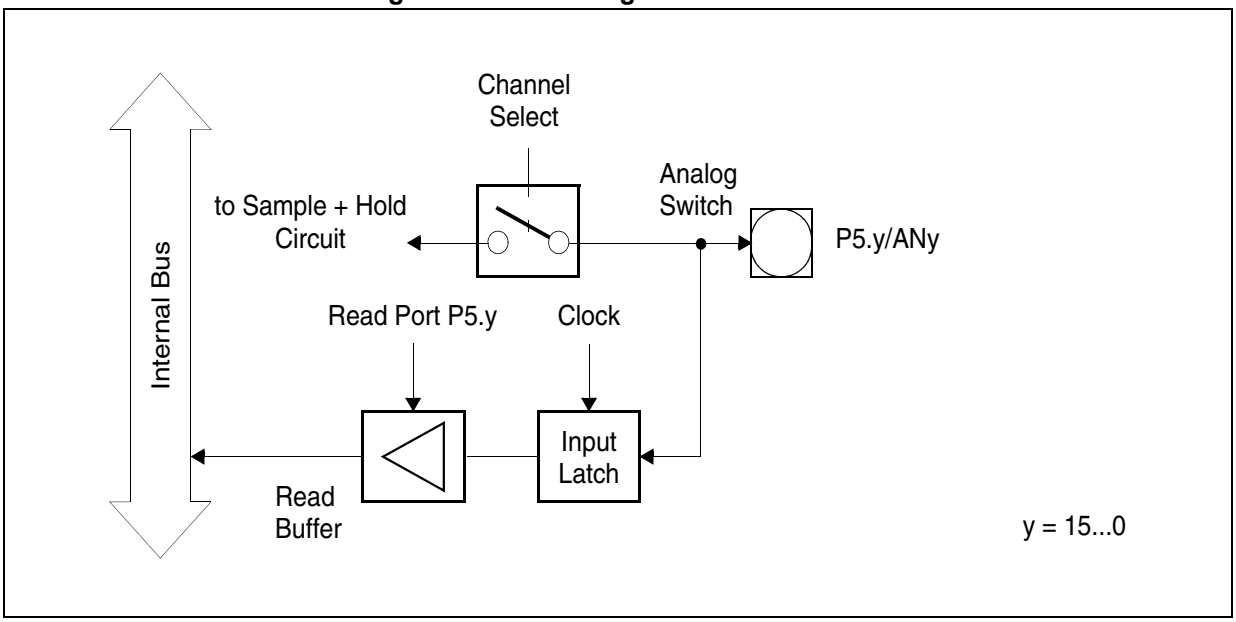
Port 5 Pin	Alternate Function a)	Alternate Function b)
P5.0	Analog Input AN0	-
P5.1	Analog Input AN1	-
P5.2	Analog Input AN2	-
P5.3	Analog Input AN3	-
P5.4	Analog Input AN4	-
P5.5	Analog Input AN5	-
P5.6	Analog Input AN6	-
P5.7	Analog Input AN7	-
P5.8	Analog Input AN8	-
P5.9	Analog Input AN9	-
P5.10	Analog Input AN10	T6EUDTimer 6 external Up/Down Input
P5.11	Analog Input AN11	T5EUDTimer 5 external Up/Down Input
P5.12	Analog Input AN12	T6INTimer 6 Count Input
P5.13	Analog Input AN13	T5INTimer 5 Count Input
P5.14	Analog Input AN14	T4EUDTimer 4 external Up/Down Input
P5.15	Analog Input AN15	T2EUDTimer 2 external Up/Down Input

Figure 33. PORT5 I/O and Alternate Functions



Port 5 pins have a special port structure (see *Figure 34*), first because it is an input only port, and second because the analog input channels are directly connected to the pins rather than to the input latches.

Figure 34. Block Diagram of a Port 5 Pin



12.8.2 Port 5 Schmitt Trigger Analog Inputs

A Schmitt trigger protection can be activated on each pin of Port 5 by setting the dedicated bit of register P5DIDIS.

P5DIDIS (FFA4H / D2H)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P5DI DIS.15	P5DI DIS.14	P5DI DIS.13	P5DI DIS.12	P5DI DIS.11	P5DI DIS.10	P5DI DIS.9	P5DI DIS.8	P5DI DIS.7	P5DI DIS.6	P5DI DIS.5	P5DI DIS.4	P5DI DIS.3	P5DI DIS.2	P5DI DIS.1	P5DI DIS.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

P5DIDIS.y	**Port 5 Digital Disable Register Bit y** P5DIDIS.y = 0: Port line P5.y digital input is enabled (Schmitt trigger enabled) P5DIDIS.y = 1: Port line P5.y digital input is disabled (Schmitt trigger disabled, necessary for input leakage current reduction)

12.9 Port 6

If this 8-bit port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction register DP6. Each port line can be switched into push/pull or open drain mode via the open drain control register ODP6.

P6 (FFCCh / E6h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0
								RW	RW	RW	RW	RW	RW	RW	RW

P6.y	**Port Data Register P6 Bit y**

DP6 (FFCEH / E7H)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP6.7	DP6.6	DP6.5	DP6.4	DP6.3	DP6.2	DP6.1	DP6.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP6.y	**Port Direction Register DP6 Bit y** DP6.y = 0: Port line P6.y is an input (high impedance) DP6.y = 1: Port line P6.y is an output

ODP6 (F1CEH / E7H)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	ODP6.7	ODP6.6	ODP6.5	ODP6.4	ODP6.3	ODP6.2	ODP6.1	ODP6.0
								RW	RW	RW	RW	RW	RW	RW	RW

ODP6.y	**Port 6 Open Drain Control Register Bit y** ODP6.y = 0: Port line P6.y output driver in push-pull mode ODP6.y = 1: Port line P6.y output driver in open drain mode

12.9.1 Alternate Functions of Port 6

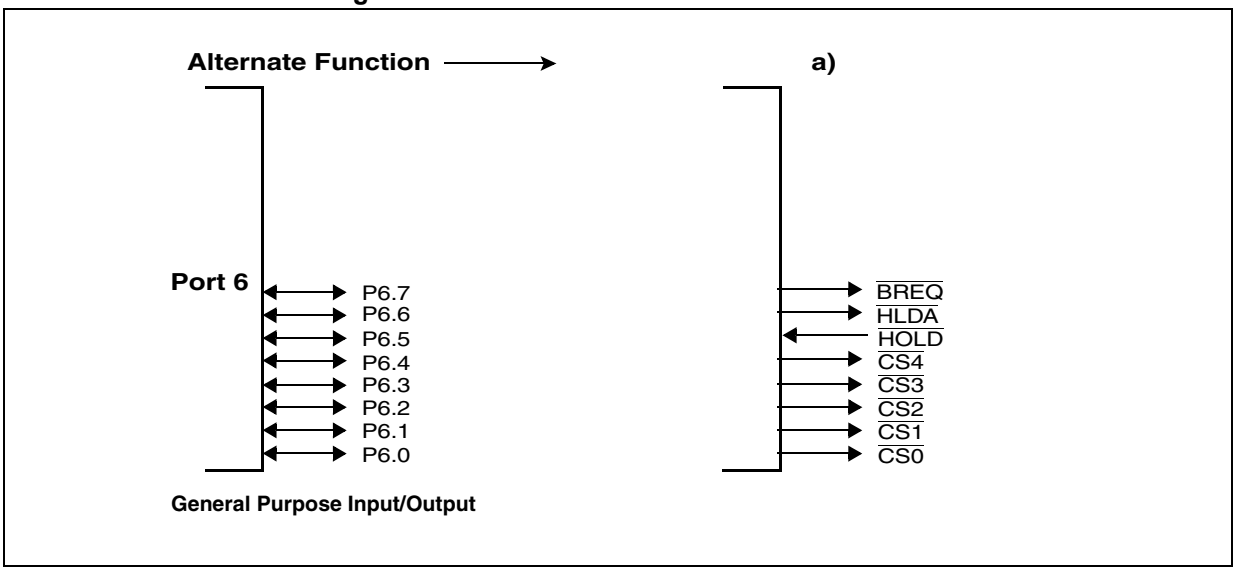
A programmable number of chip select signals ($\overline{CS4}$...$\overline{CS0}$) derived from the bus control registers (BUSCON4...BUSCON0) can be output on 5 pins of Port 6.
The number of chip select signals is selected via PORT0 during reset. The selected value can be read from bit-field CSSEL in register RP0H (read only) in order to check the configuration during run time.

Table 25 summarizes the alternate functions of Port 6 depending on the number of selected chip select lines (coded via bit-field CSSEL).

Table 25. Port 6 Alternate Functions

Port 6	Alternate Function CSSEL = 10	Alternate Function CSSEL = 01	Alternate Function CSSEL = 00	Alternate Function CSSEL = 11
P6.0	General purpose I/O	Chip select $\overline{CS0}$	Chip select $\overline{CS0}$	Chip select $\overline{CS0}$
P6.1	General purpose I/O	Chip select $\overline{CS1}$	Chip select $\overline{CS1}$	Chip select $\overline{CS1}$
P6.2	General purpose I/O	General purpose I/O	Chip select $\overline{CS2}$	Chip select $\overline{CS2}$
P6.3	General purpose I/O	General purpose I/O	General purpose I/O	Chip select $\overline{CS3}$
P6.4	General purpose I/O	General purpose I/O	General purpose I/O	Chip select $\overline{CS4}$
P6.5	$\overline{HOLD}$ External hold request input			
P6.6	$\overline{HLDA}$ Hold acknowledge output			
P6.7	$\overline{BREQ}$ Bus request output			

Figure 35. PORT6 I/O and Alternate Functions



The chip select lines of Port 6 have an internal weak pull-up device. This device is switched on during reset. This feature is implemented to drive the chip select lines high during reset in order to avoid multiple chip selection.

After reset the $\overline{CS}$ function must be used, if selected so. In this case there is no possibility to program any port latches before. Thus the alternate function ($\overline{CS}$) is selected automatically in this case.

Note: The open drain output option can only be selected via software earliest during the initialization routine; at least signal CS0 will be in push/pull output driver mode directly after reset.

Figure 36. Block Diagram of Port 6 Pins with an Alternate Output Function

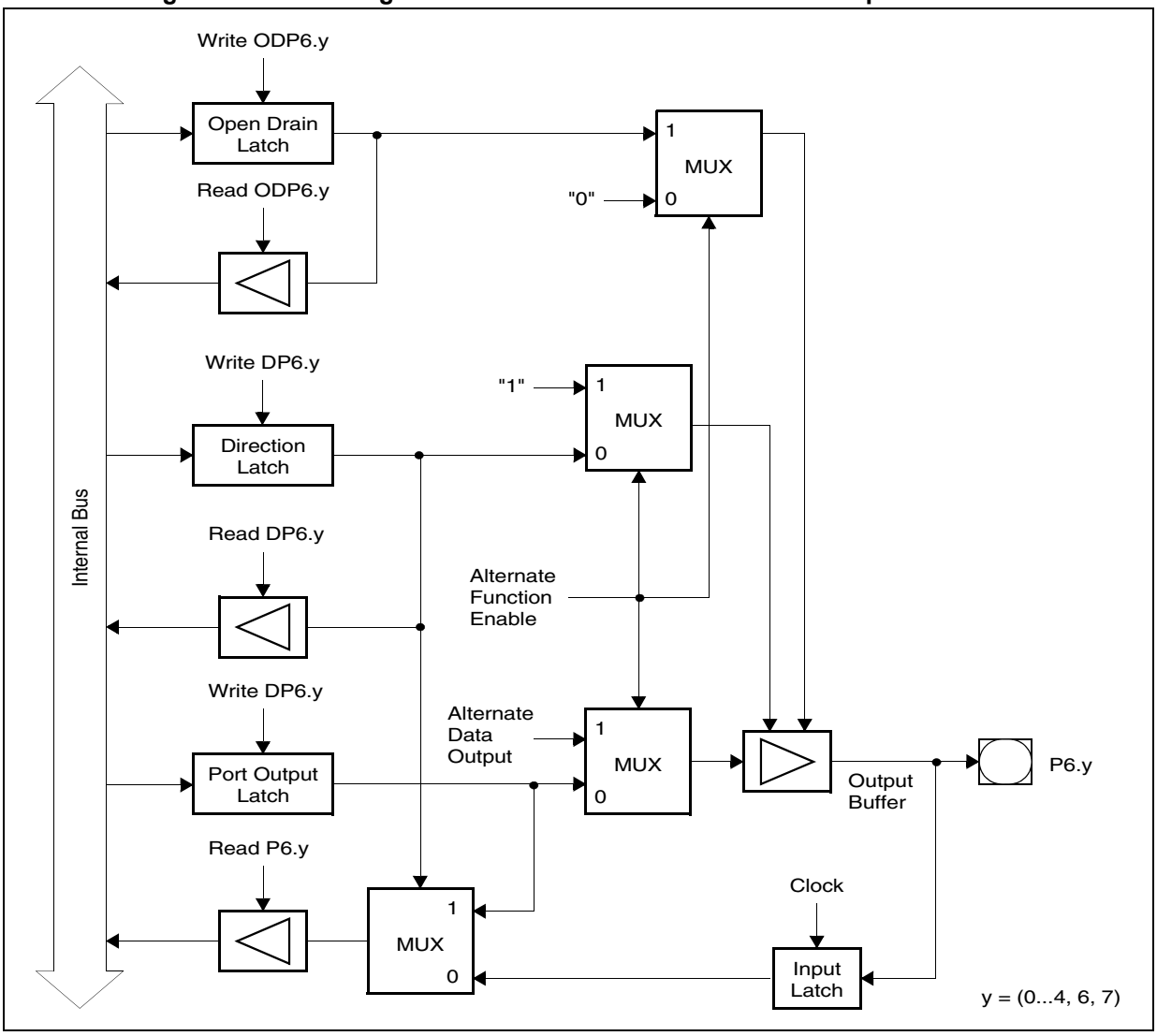
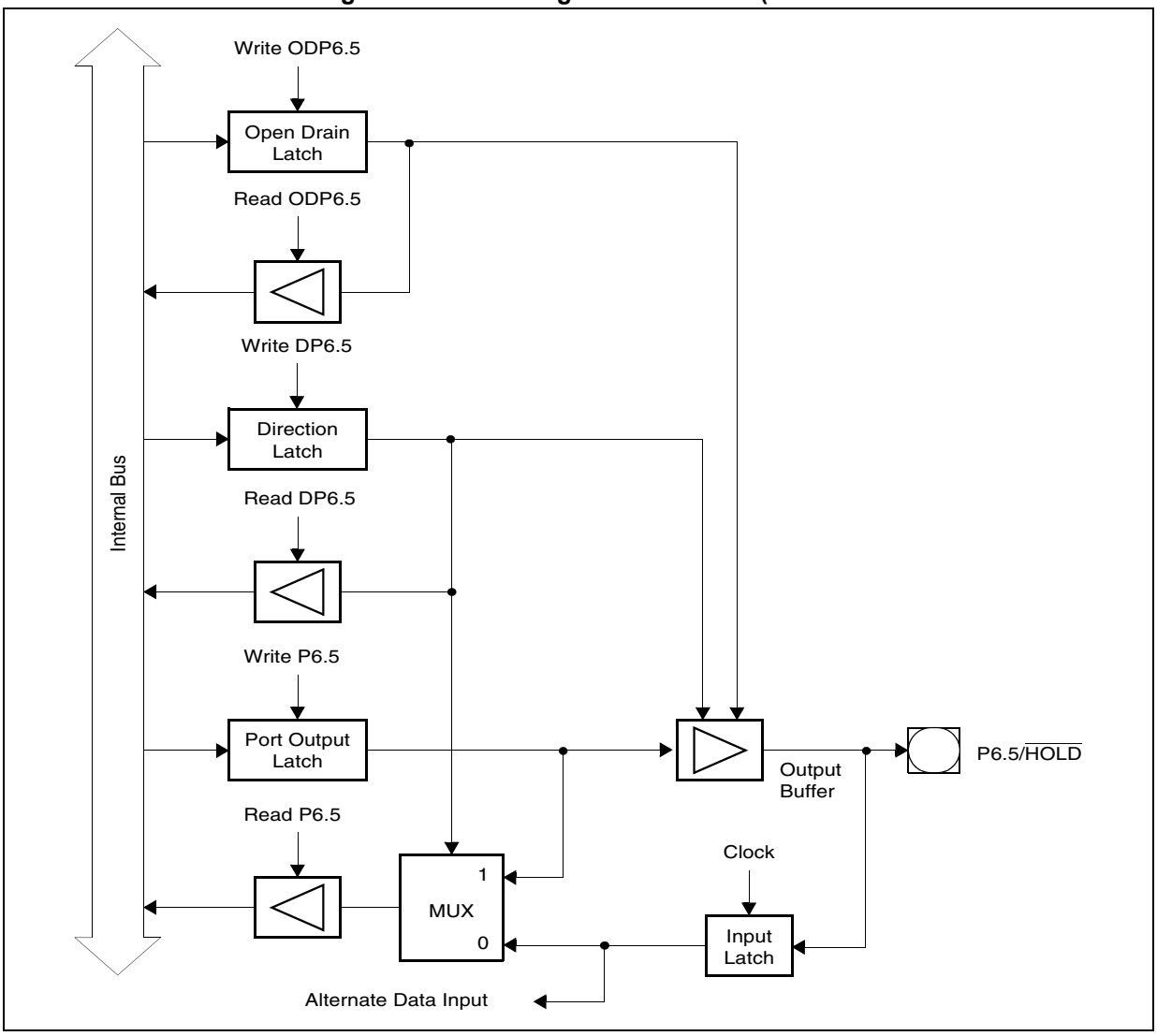


Figure 37. Block Diagram of Pin P6.5 ($\overline{\text{HOLD}}$)



12.10 Port 7

If this 8-bit port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction register DP7. Each port line can be switched into push-pull or open drain mode via the open drain control register ODP7.

P7 (FFD0h / E8h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0
								RW	RW	RW	RW	RW	RW	RW	RW

P7.y	**Port Data Register P7 Bit y**

DP7 (FFD2h / E9h)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP7.7	DP7.6	DP7.5	DP7.4	DP7.3	DP7.2	DP7.1	DP7.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP7.y	**Port Direction Register DP7 Bit y** DP7.y = 0: Port line P7.y is an input (high impedance) DP7.y = 1: Port line P7.y is an output

ODP7 (F1D2h / E9h)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	ODP7.7	ODP7.6	ODP7.5	ODP7.4	ODP7.3	ODP7.2	ODP7.1	ODP7.0
								RW	RW	RW	RW	RW	RW	RW	RW

ODP7.y	**Port 7 Open Drain Control Register Bit y** ODP7.y = 0: Port line P7.y output driver in push-pull mode ODP7.y = 1: Port line P7.y output driver in open drain mode

12.10.1 Alternate Functions of Port 7

The upper 4 lines of Port 7 (P7.7...P7.4) serve as capture inputs or compare outputs (CC31IO...CC28IO) for the CAPCOM2 unit.

The usage of the port lines by the CAPCOM unit, its accessibility via software and the precautions are the same as described for the Port 2 lines.

As all other capture inputs, the capture input function of pins P7.7...P7.4 can also be used as external interrupt inputs (200 ns sample rate at 40 MHz CPU clock and 250 ns sample rate at 32 MHz CPU clock).

The lower 4 lines of Port 7 (P7.3...P7.0) serve as outputs from the PWM module (POUT3...POUT0).

At these pins the value of the respective port output latch is EXORed with the value of the PWM output rather than ANDed, as the other pins do.

This allows to use the alternate output value either as it is (port latch holds a ‘0’) or to invert its level at the pin (port latch holds a ‘1’).

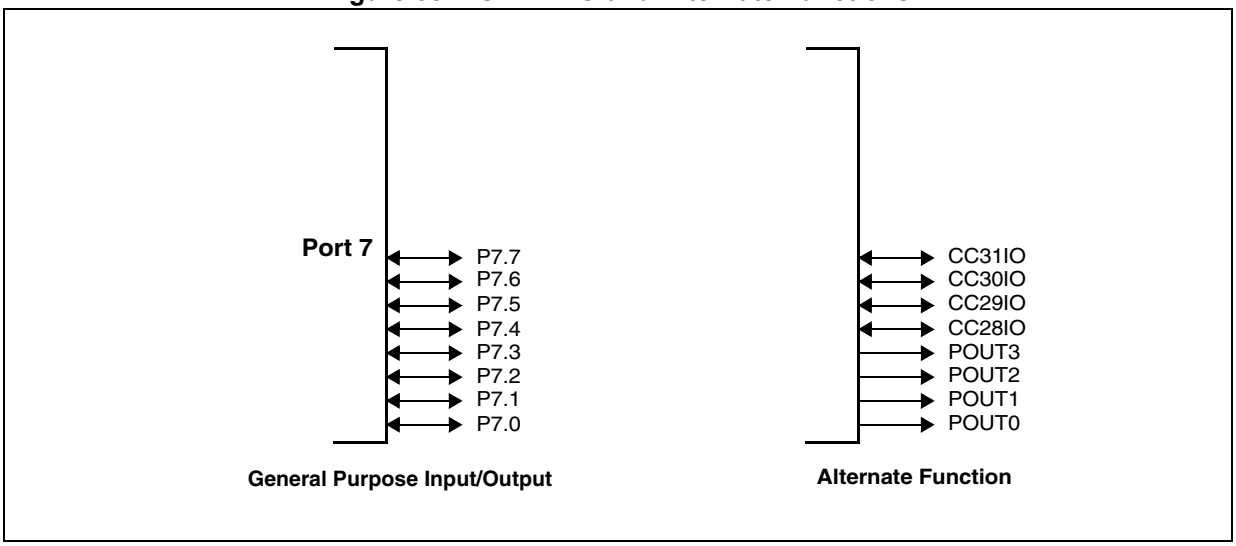
Note that the PWM outputs must be enabled via the respective PENx bit in PWMCON1.

Table 26 summarizes the alternate functions of Port 7.

Table 26. Port 7 Alternate Functions

Port 7	Alternate Function
P7.0	POUT0PWM mode channel 0 output
P7.1	POUT1PWM mode channel 1 output
P7.2	POUT2PWM mode channel 2 output
P7.3	POUT3PWM mode channel 3 output
P7.4	CC28IOCapture input / compare output channel 28
P7.5	CC29IOCapture input / compare output channel 29
P7.6	CC30IOCapture input / compare output channel 30
P7.7	CC31IOCapture input / compare output channel 31

Figure 38. PORT 7 I/O and Alternate Functions



The structure of Port 7 differs in the way the output latches are connected to the internal bus and to the pin driver. Pins P7.3...P7.0 (POUT3...POUT0) EXOR the alternate data output with the port latch output, which allows to use the alternate data directly or inverted at the pin driver.

Figure 39. Block Diagram of Port 7 Pins P7.3...P7.0

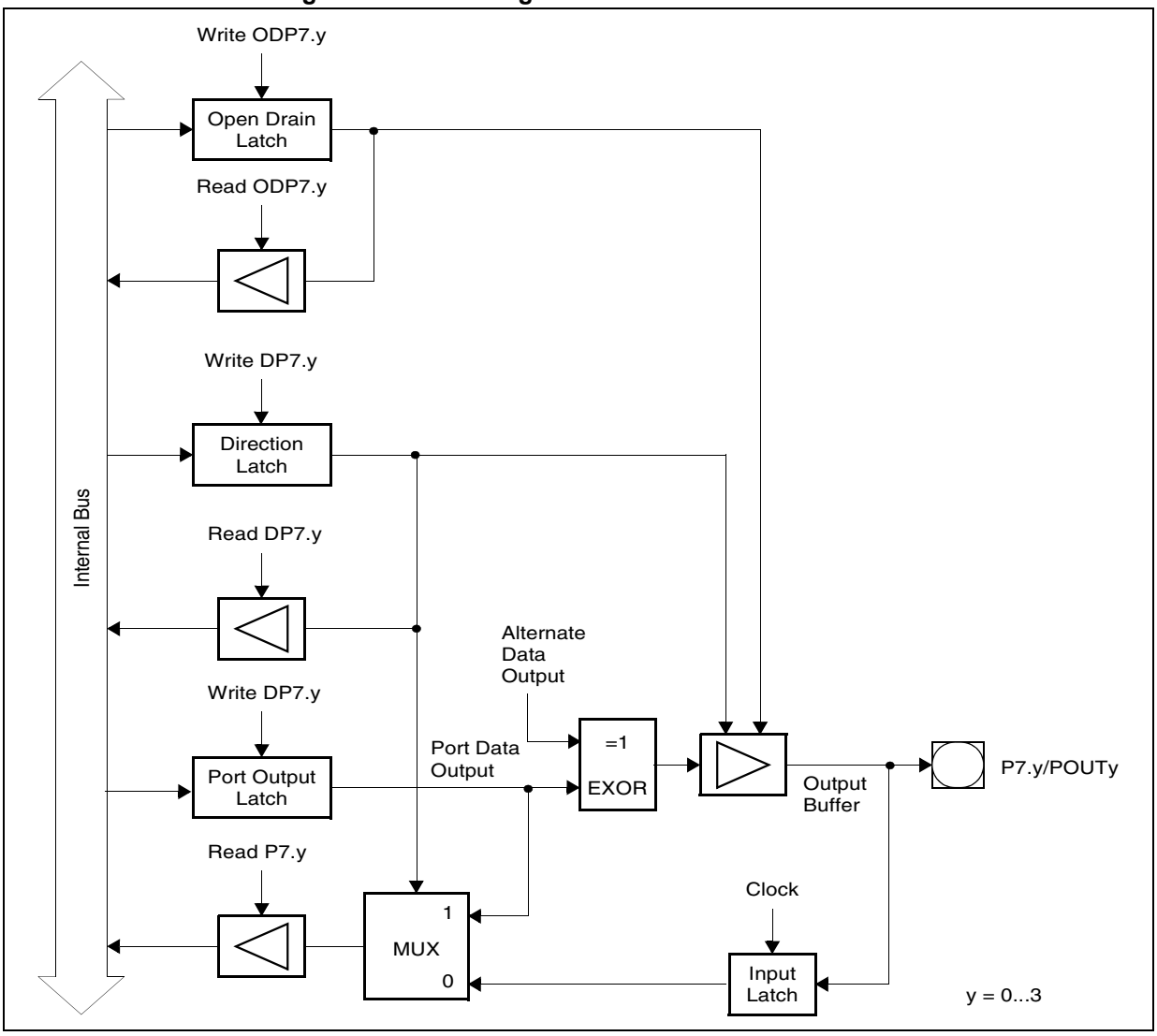
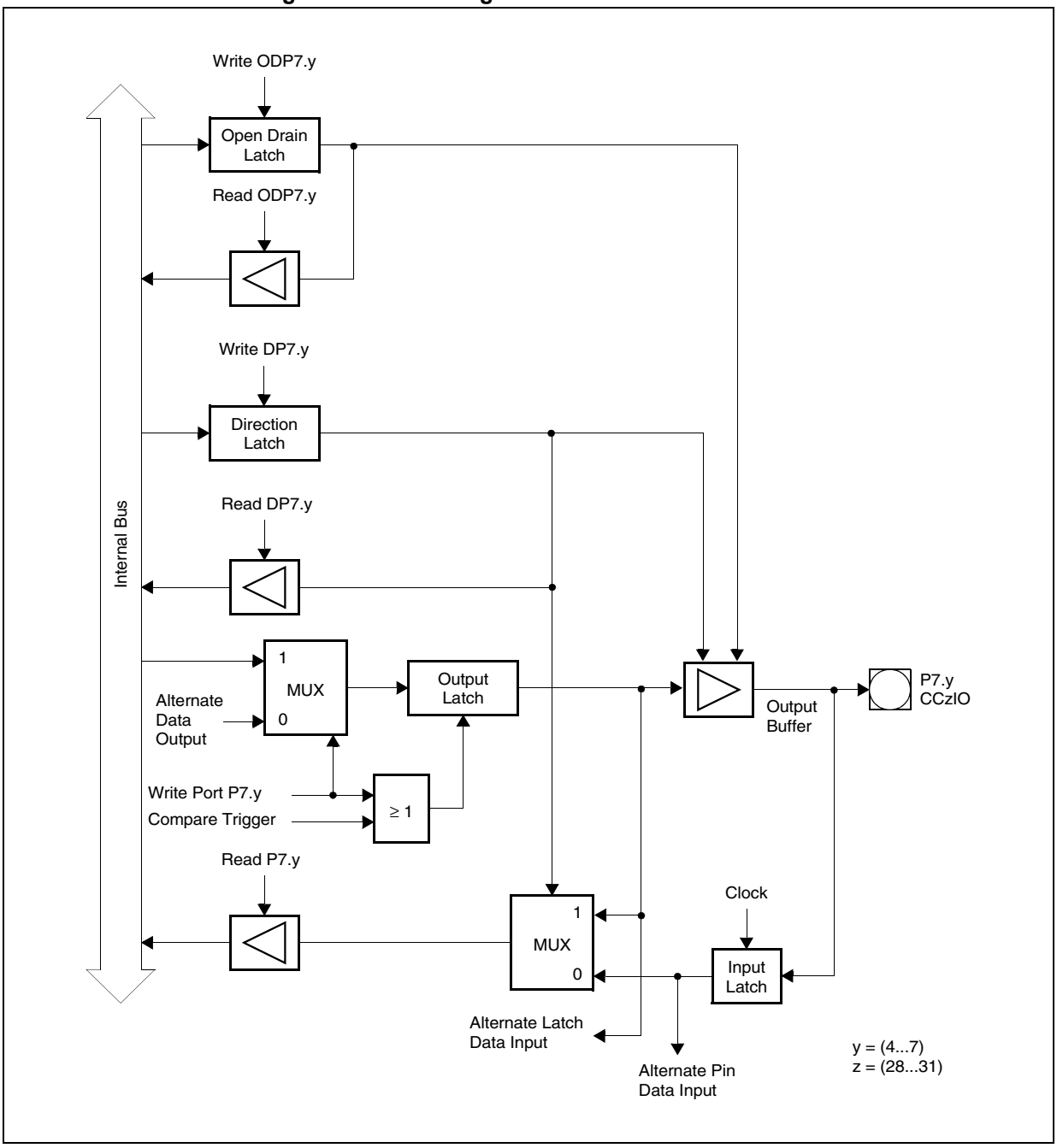


Figure 40. Block Diagram of Port 7 Pins P7.7...P7.



12.11 Port 8

If this 8-bit port is used for general purpose I/O, the direction of each line can be configured via the corresponding direction register DP8. Each port line can be switched into push/pull or open drain mode via the open drain control register ODP8.

P8 (FFD4h / EAh)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	P8.7	P8.6	P8.5	P8.4	P8.3	P8.2	P8.1	P8.0
								RW	RW	RW	RW	RW	RW	RW	RW

P8.y	**Port Data Register P8 Bit y**

DP8 (FFD6h / EBh)

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	DP8.7	DP8.6	DP8.5	DP8.4	DP8.3	DP8.2	DP8.1	DP8.0
								RW	RW	RW	RW	RW	RW	RW	RW

DP8.y	**Port Direction Register DP8 Bit y** DP8.y = 0: Port line P8.y is an input (high impedance) DP8.y = 1: Port line P8.y is an output

ODP8 (F1D6h / EBh)

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	ODP8.7	ODP8.6	ODP8.5	ODP8.4	ODP8.3	ODP8.2	ODP8.1	ODP8.0
								RW	RW	RW	RW	RW	RW	RW	RW

ODP8.y	**Port 8 Open Drain Control Register Bit y** ODP8.y = 0: Port line P8.y output driver in push-pull mode ODP8.y = 1: Port line P8.y output driver in open drain mode

12.11.1 Alternate Functions of Port 8

The 8 lines of Port 8 serve as capture inputs or as compare outputs (CC23IO...CC16IO) for the CAPCOM2 unit.

The usage of the port lines by the CAPCOM unit, its accessibility via software and the precautions are the same as described for the Port 2 lines.

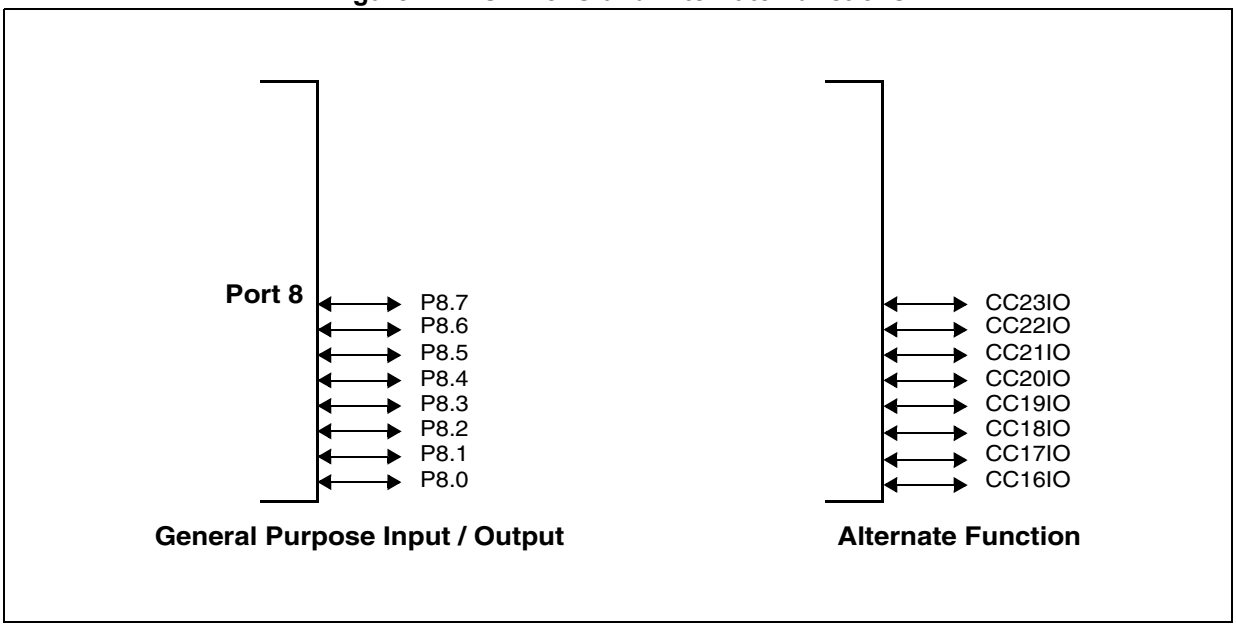
As all other capture inputs, the capture input function of pins P8.7...P8.0 can also be used as external interrupt inputs (200ns sample rate at 40MHz CPU clock and 250ns sample rate at 32MHz CPU clock).

Table 27 summarizes the alternate functions of Port 8.

Table 27. Port 8 Alternate Functions

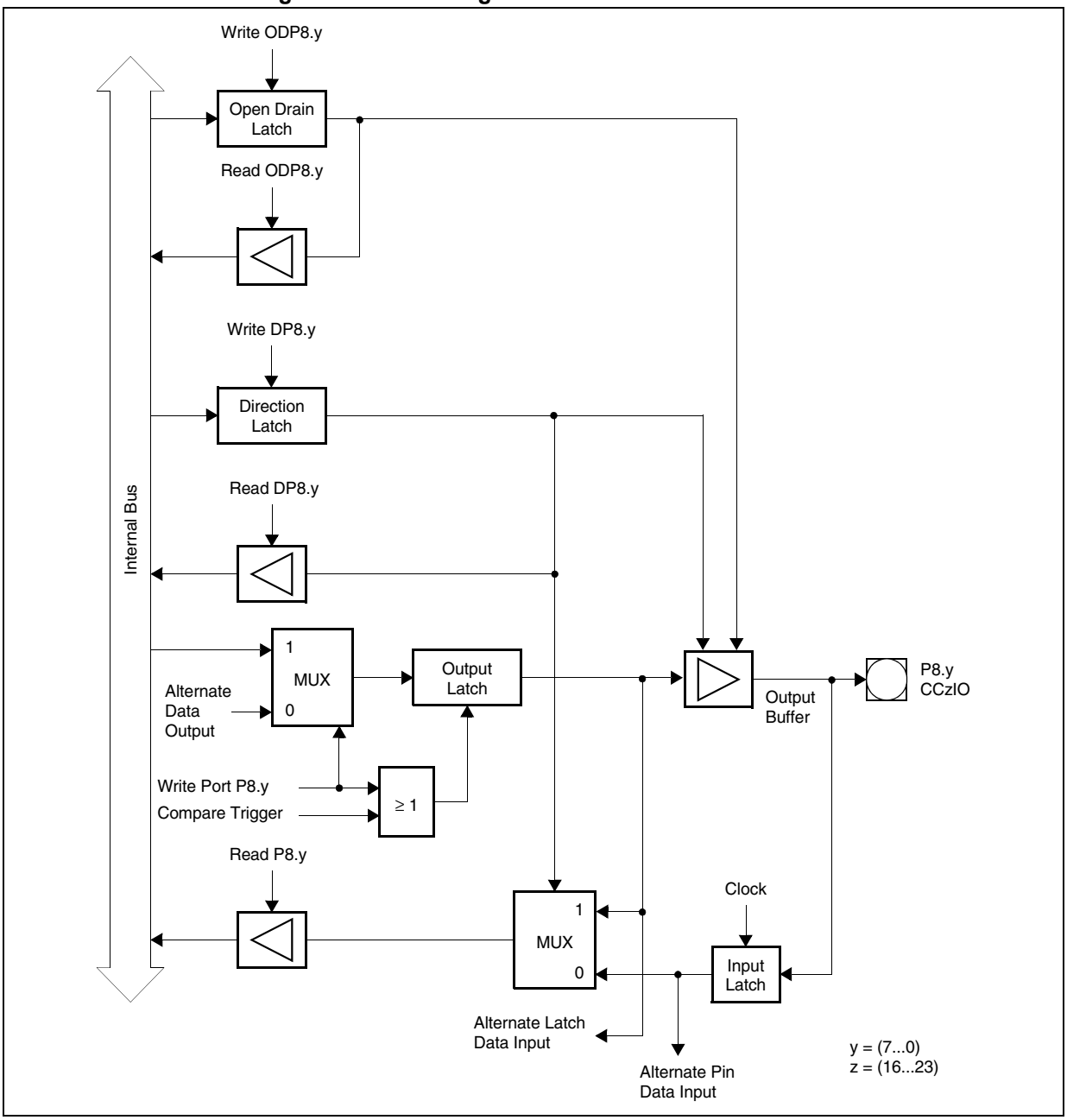
Port 7	Alternate Function
P8.0	CC16IOCapture input / compare output channel 16
P8.1	CC17IOCapture input / compare output channel 17
P8.2	CC18IOCapture input / compare output channel 18
P8.3	CC19IOCapture input / compare output channel 19
P8.4	CC20IOCapture input / compare output channel 20
P8.5	CC21IOCapture input / compare output channel 21
P8.6	CC22IOCapture input / compare output channel 22
P8.7	CC23IOCapture input / compare output channel 23

Figure 41. PORT 8I/O and Alternate Functions



The structure of Port 8 differs in the way the output latches are connected to the internal bus and to the pin driver (see *Figure 42*). Pins P8.7...P8.0 (CC23IO...CC16IO) combine internal bus data and alternate data output before the port latch input, as do the Port 2 pins

Figure 42. Block Diagram of Port 8 Pins P8.7...P8.0



13 A/D Converter

A 10-bit A/D converter with 16 multiplexed input channels and a sample and hold circuit is integrated on-chip. The sample time (for loading the capacitors) and the conversion time is programmable and can be adjusted to the external circuitry.

To remove high frequency components from the analog input signal, a low-pass filter must be connected at the ADC input.

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 analog input channels, the remaining channel inputs can be used as digital input port pins. The A/D converter of the ST10F269 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

Table 28. ADC Sample Clock and Conversion Clock at f_{CPU} = 40 MHz

ADCTC	Conversion Clock t_{CC}		ADSTC	Sample Clock t_{SC}	
	TCL[1] = 1/2 x f_{XTAL}	At f_{CPU} = 40 MHz		t_{SC} =	At f_{CPU} = 40 MHz
00	TCL x 24	0.3µs	00	t_{CC}	0.3µs[2]
01	Reserved, do not use	Reserved	01	t_{CC} x 2	0.6µs [2]
10	TCL x 96	1.2 µs	10	t_{CC} x 4	1.2µs [2]
11	TCL x 48	0.6 µs	11	t_{CC} x 8	2.4µs [2]

1. *Section 21.4.5* for TCL definition.
2. t_{CC} = TCL x 24.

Table 29. ADC Sample Clock and Conversion Clock at f_{CPU} = 32MHz

ADCON 15/14 ADCTC	Conversion Clock t_{CC}		ADCON 13/12 ADSTC	Sample Clock t_{SC}	
	TCL[1] = 1/2 x f_{XTAL}	At f_{CPU} = 32MHz		t_{SC} =	At f_{CPU} = 32MHz
00	TCL x 24	0.375µs	00	t_{CC}	0.375µs [2]
01	Reserved, do not use	Reserved	01	t_{CC} x 2	0.75µs [2]
10	TCL x 96	1.5 µs	10	t_{CC} x 4	1.50µs [2]
11	TCL x 48	0.75 µs	11	t_{CC} x 8	3.00µs [2]

1. *Section 21.4.5* for TCL definition.
2. t_{CC} = TCL x 24.

14 Serial Channels

Serial communication with other microcontrollers, microprocessors, terminals or external peripheral components is provided by two serial interfaces: the asynchronous / synchronous serial channel (ASCO) and the high-speed synchronous serial channel (SSC). Two dedicated baud rate generators set up all standard baud rates without the requirement of oscillator tuning. For transmission, reception and erroneous reception, 3 separate interrupt vectors are provided for each serial channel.

14.1 Asynchronous / Synchronous Serial Interface (ASCO)

The asynchronous / synchronous serial interface (ASCO) provides serial communication between the ST10F269 and other microcontrollers, microprocessors or external peripherals.

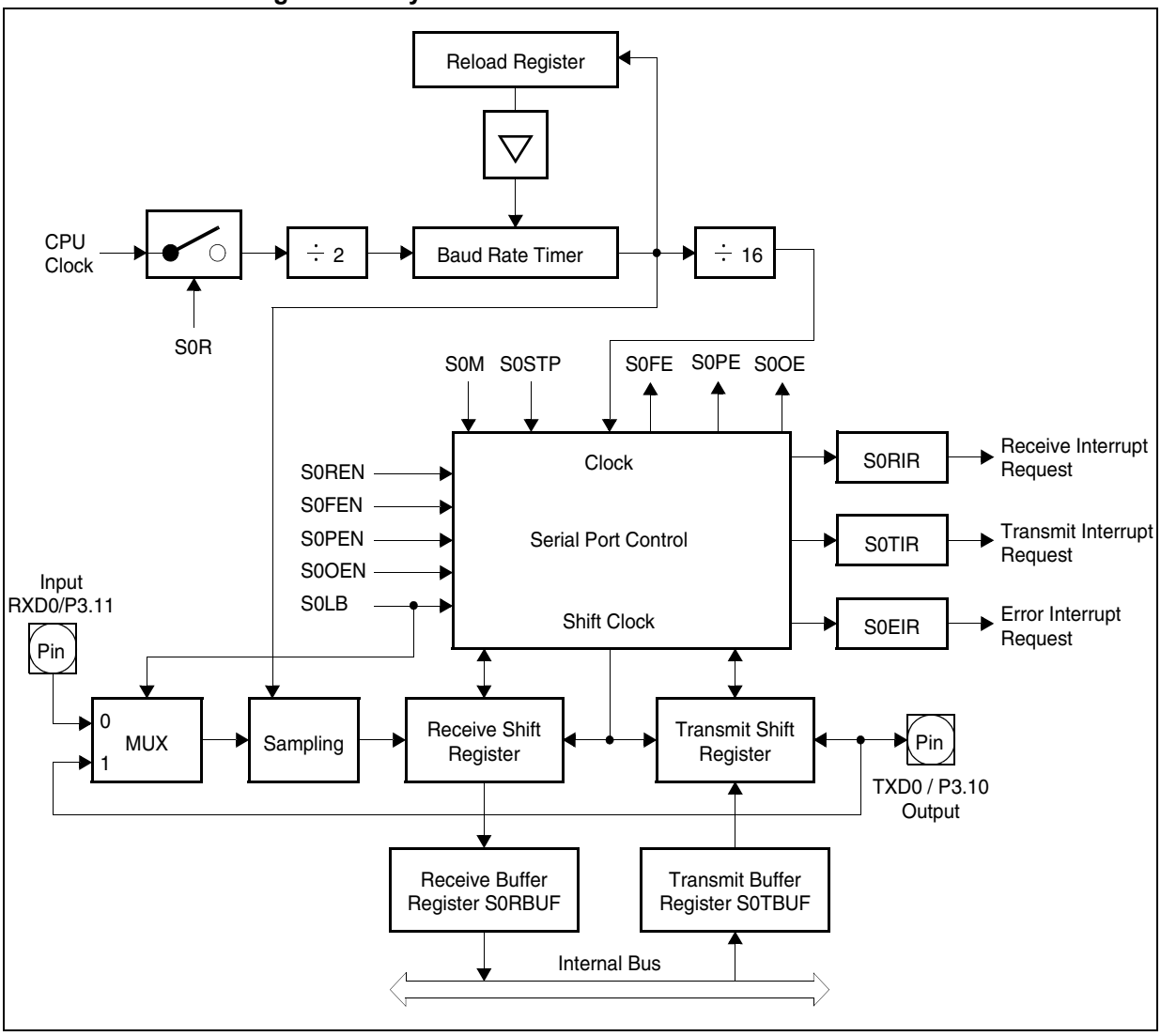
A set of registers is used to configure and to control the ASCO serial interface:

- P3, DP3, ODP3 for pin configuration
- SOBG for baud rate generator
- SOTBUF for transmit buffer
- SOTIC for transmit interrupt control
- SOTBIC for transmit buffer interrupt control
- SOCON for control
- SORBUF for receive buffer (read only)
- SORIC for receive interrupt control
- SOEIC for error interrupt control

14.1.1 ASCO in Asynchronous Mode

In asynchronous mode, 8 or 9-bit data transfer, parity generation and the number of stop bit 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 CPU Clock (and up to 1 Mbaud at 32 MHz CPU Clock) is supported in this mode.

Figure 43. Asynchronous Mode of Serial Channel ASC0



Asynchronous Mode baud rates

For asynchronous operation, the baud rate generator provides a clock with 16 times the rate of the established baud rate. Every received bit is sampled at the 7th, 8th and 9th cycle of this clock. The baud rate for asynchronous operation of serial channel ASC0 and the required reload value for a given baud rate can be determined by the following formulas:

$$B_{Async} = \frac{f_{CPU}}{16 \times [2 + (S0BRS)] \times [(S0BRL) + 1]}$$

$$S0BRL = \left(\frac{f_{CPU}}{16 \times [2 + (S0BRS)] \times B_{Async}}\right) - 1$$

(S0BRL) represents the content of the reload register, taken as unsigned 13-bit integer, (S0BRS) represents the value of bit S0BRS (‘0’ or ‘1’), taken as integer.

Using the above equation, the maximum baud rate can be calculated for any given clock speed. Baud rate versus reload register value (SOBRS=0 and SOBRS=1) is described in *Table 30* and *Table 31*.

Table 30. Commonly Used Baud Rates by Reload Value and Deviation Errors at f_{CPU} = 40 MHz

S0BRS = '0', f_{CPU} = 25 MHz			S0BRS = '1', f_{CPU} = 25 MHz		
Baud Rate (baud)	Deviation Error(1)	Reload Value (hexa)	Baud Rate (baud)	Deviation Error	Reload Value (hexa)
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

1. The deviation errors given in *Table 30* are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

Table 31. Commonly Used Baud Rates by Reload Value and Deviation Errors at f_{CPU} = 32 MHz

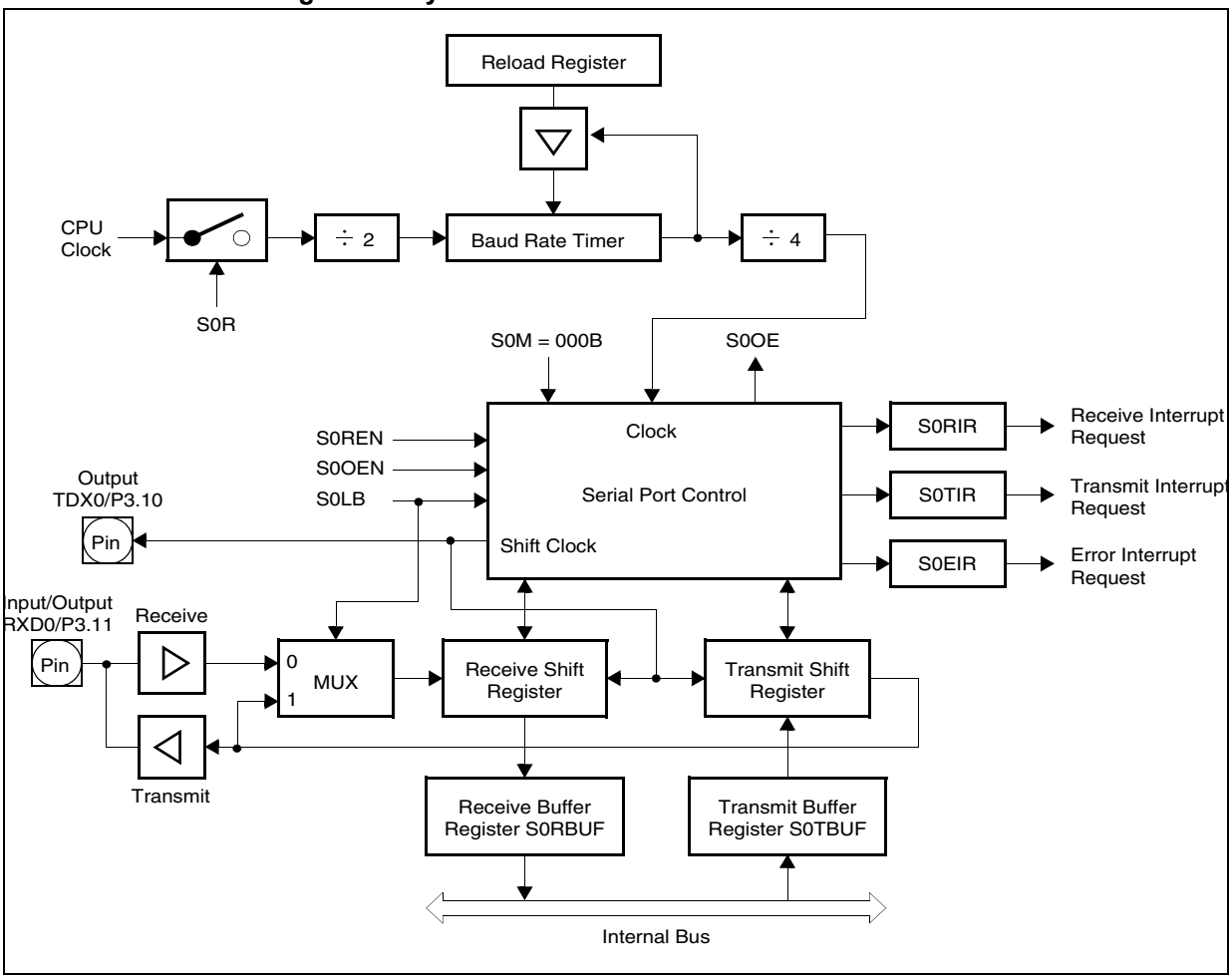
S0BRS = '0', f_{CPU} = 32MHz			S0BRS = '1', f_{CPU} = 32MHz		
Baud Rate (baud)	Deviation Error(1)	Reload Value	Baud Rate (baud)	Deviation Error	Reload Value
1000 000	±0.0%	0000h	666 667	±0.0%	0000h
56000	+5.0% / -0.8%	0010h / 001h	56000	+8.2% / -0.8%	000Ah / 000Bh
38400	+0.2% / -3.5%	0019h / 0020h	38400	+2.1% / -3.5%	0010h / 0011h
19200	+0.2% / -1.7%	0033h / 0034h	19200	+2.1% / -0.8%	0021h / 0022h
9600	+0.2% / -0.8%	0067h/ 0068h	9600	+0.6% / -0.8%	0044h / 0045h
4800	+0.5% / -0.3%	00CFh / 00CEh	4800	+0.6% / -0.1%	0089h / 008Ah
2400	+0.2% / -0.1%	019Fh / 01A0h	2400	+0.3% / -0.1%	0114h / 0115h
1200	+0.1% / -0.1%	0340h / 0341h	1200	+0.1% / -0.1%	022Ah / 022Bh
600	+0.1% / -0.1%	0681h / 0682h	600	+0.1% / -0.1%	0456h / 0457h
95	+0.1% / -0.1%	291Dh / 291Eh	75	+0.1% / 0.1%	22B7h / 22B8h
-	-	-	63	+0.1% / -0.1%	2955h / 2956h

1. The deviation errors given in *Table 31* are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

14.1.2 ASCO in Synchronous Mode

In synchronous mode, data are transmitted or received synchronously to a shift clock which is generated by the ST10F269. Half-duplex communication up to 5 Mbaud (at 40 MHz f_{CPU}) or 4 Mbaud (at 32 MHz) is possible in this mode.

Figure 44. Synchronous Mode of Serial Channel ASC0



Synchronous Mode Baud Rates

For synchronous operation, the baud rate generator provides a clock with 4 times the rate of the established baud rate. The baud rate for synchronous operation of serial channel ASC0 can be determined by the following formula:

$$B_{Sync} = \frac{f_{CPU}}{4 \times [2 + (S0BRS)] \times [(S0BRL) + 1]}$$

$$S0BRL = \left(\frac{f_{CPU}}{4 \times [2 + (S0BRS)] \times B_{Sync}} \right) - 1$$

(S0BRL) represents the content of the reload register, taken as unsigned 13-bit integers, (S0BRS) represents the value of bit S0BRS (‘0’ or ‘1’), taken as integer.

Using the above equation, the maximum baud rate can be calculated for any clock speed as given in *Table 32* and *Table 33*.

Table 32. Commonly Used Baud Rates by Reload Value and Deviation Errors (f_{CPU} = 40 MHz)

S0BRS = '0'			S0BRS = '1'		
Baud Rate (baud)	Deviation Error(1)	Reload Value (hexa)	Baud Rate (baud)	Deviation Error	Reload Value (hexa)
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

1. The deviation errors given in *Table 32* are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

Table 33. Commonly Used Baud Rates by Reload Value and Deviation Errors (f_{CPU} = 32 MHz)

S0BRS = '0'			S0BRS = '1'		
Baud Rate (baud)	Deviation Error(1)	Reload Value	Baud Rate (baud)	Deviation Error	Reload Value
4 000 000	±0.0%	0000h	2 666 667	±0.0%	0000h
224 000	+5.0% / -0.8%	0011h / 0012h	224 000	+8.2% / -0.8%	000Bh / 000Ch
112 000	+2.0% / -0.8%	0023h / 0024h	112 000	+3.5% / -0.8%	0017h / 0018h
56 000	+0.6% / -0.8%	0046h / 0047h	56 000	+1.3% / -0.8%	002Fh / 0030h
38 400	+0.2% / -0.85%	0077h / 0078h	38 400	+0.6% / -0.8%	0044h / 0045h
19 200	+0.2% / -0.3%	00BFh / 00C0h	19 200	+0.6% / -0.1%	008Ah / 008Bh
9 600	+0.2% / -0.1%	01A0h/ 01A1h	9 600	+0.3% / -0.1%	0115h / 0116h
4 800	+0.0% / -0.1%	0340h / 0341h	4 800	+0.1% / -0.1%	022Bh / 022Ch
2 400	+0.0% / -0.0%	0682h / 0683h	2 400	+0.0% / -0.1%	0456h / 0457h
1 200	+0.0% / -0.0%	004h / 0D05h	1 200	+0.0% / -0.0%	08ACh / 08ADh
600	+0.0% / -0.0%	1A0Ah / 1A0Bh	600	+0.0% / -0.0%	115Bh / 115C7h
490	+0.0% / -0.0%	1FE2h / 1FE3h	320	+0.2%	1FFFh

1. The deviation errors given in *Table 33* are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

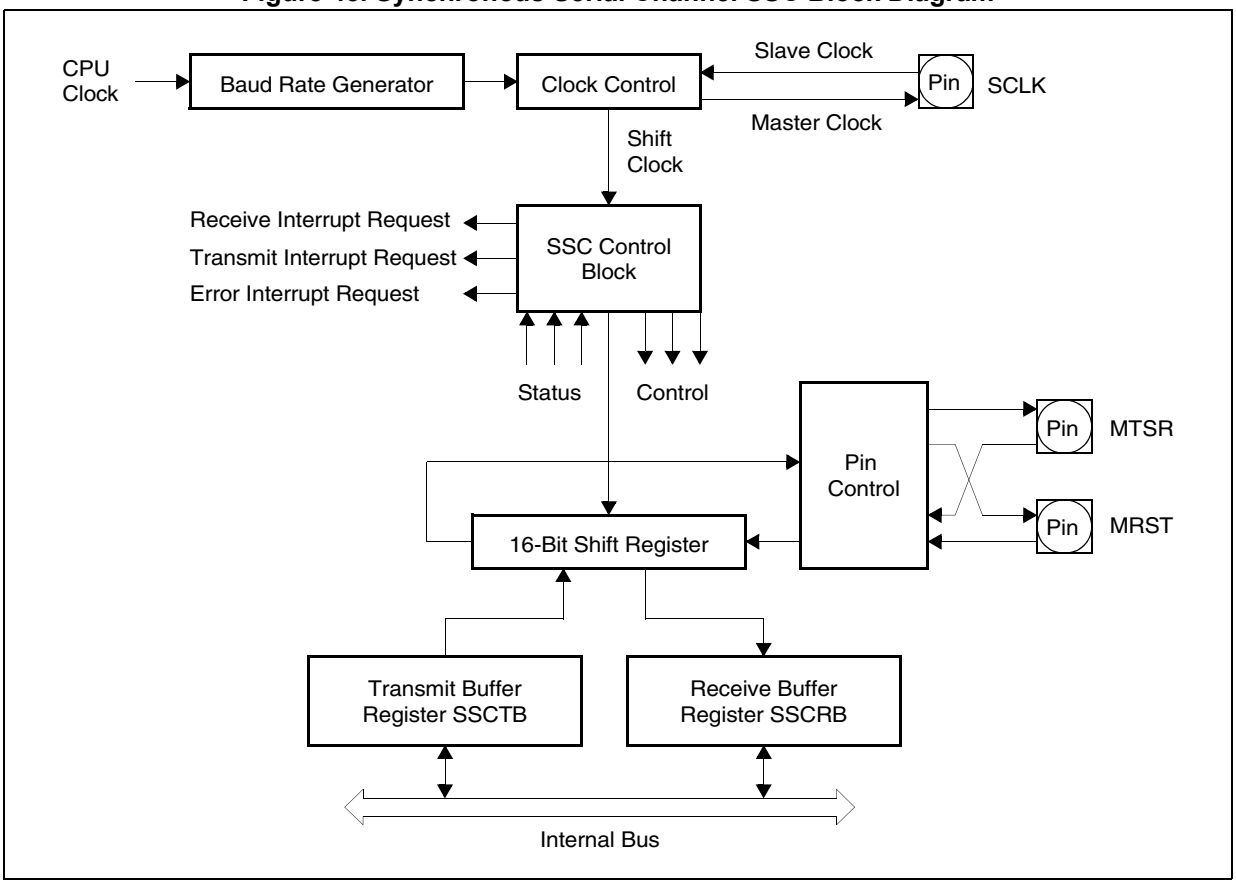
14.2 High Speed Synchronous Serial Channel (SSC)

The High-Speed Synchronous Serial Interface SSC provides flexible high-speed serial communication between the ST10F269 and other microcontrollers, microprocessors or external peripherals.

The SSC supports full-duplex and half-duplex synchronous communication. The serial clock signal can be generated by the SSC 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 baud rate generator provides the SSC with a separate serial clock signal. The serial channel SSC has its own dedicated 16-bit baud rate generator with 16-bit reload capability, allowing baud rate generation independent from the timers.

Figure 45. Synchronous Serial Channel SSC Block Diagram



Baud Rate Generation

The baud rate generator is clocked by $f_{CPU}/2$. The timer is counting downwards and can be started or stopped through the global enable bit SSCEN in register SSCCON. Register SSCBR is the dual-function Baud Rate Generator/Reload register. Reading SSCBR, while the SSC is enabled, returns the content of the timer. Reading SSCBR, while the SSC is disabled, returns the programmed reload value. In this mode the desired reload value can be written to SSCBR.

Note: Never write to SSCBR, while the SSC is enabled.

The formulas below calculate the resulting baud rate for a given reload value and the required reload value for a given baud rate:

$$\text{Baud rate}_{SSC} = \frac{f_{CPU}}{2 \times [(SSCBR) + 1]}$$

$$SSCBR = \left(\frac{f_{CPU}}{2 \times \text{Baud rate}_{SSC}} \right) - 1$$

(SSCBR) represents the content of the reload register, taken as unsigned 16-bit integer.

Table 34 lists some possible baud rates against the required reload values and the resulting bit times for a 25 MHz CPU clock.

Table 34. Synchronous Baud Rate and Reload Values (f_{CPU} = 40MHz)

Baud Rate	Bit Time	Reload Value
Reserved use a reload value > 0.	-	-
10 Mbaud	100 ns	0001h
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

Table 35 lists some possible baud rates against the required reload values and the resulting bit times for a 32MHz CPU clock.

Table 35. Synchronous Baud Rate and Reload Values (f_{CPU} = 32MHz)

Baud Rate	Bit Time	Reload Value
Reserved use a reload value > 0.	-	-
8 Mbaud	125 ns	0001h
4 Mbaud	250 s	0003h
2 Mbaud	500 ns	0007h
1 Mbaud	1 µs	000Fh
500 Kbaud	2 µs	001Fh
100 Kbaud	10 µs	009Fh
10 Kbaud	100 µs	030Ch
1 Kbaud	1 ms	3E7Fh
244.14 baud	5.24 ms	FFFFh

15 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).

Each on-chip CAN module can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers. These two CAN modules are both identical to the CAN module of the ST10F167.

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 *Section 4: Memory Organization*.
- 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 line of the CAN1 module is connected to the XBUS interrupt line XP0, interrupt of the CAN2 module is connected to the line XP1.
- 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 is enabled, CAN2 is disabled.

15.1 CAN Modules Memory Mapping

15.1.1 CAN1

Address range 00'EF00h - 00'EFFFh is reserved for the CAN1 Module access. CAN1 is enabled by setting XPEN bit 2 of the SYSCON register and by setting bit 0 of the XPERCON register. Accesses to the CAN Module use demultiplexed addresses and a 16-bit data bus (Byte accesses are possible). Two wait states give an access time of 125 ns at 40 MHz CPU clock or at 32 MHz CPU clock. No tri-state wait states are used.

15.1.2 CAN2

Address range 00'EE00h - 00'EEFFh is reserved for the CAN2 Module access. CAN2 is enabled by setting XPEN bit 2 of the SYSCON register and by setting bit 1 of the XPERCON register. Accesses to the CAN Module use demultiplexed addresses and a 16-bit data bus (Byte accesses are possible). Two wait states give an access time of 125 ns at 40 MHz or 32 MHz CPU clock. No tri-state wait states are used.

If one or both CAN modules is used, Port 4 cannot be programmed to output all 8 segment address lines. Thus, only 4 segment address lines can be used, reducing the external memory space to 5 Mbytes (1 Mbyte per CS line).

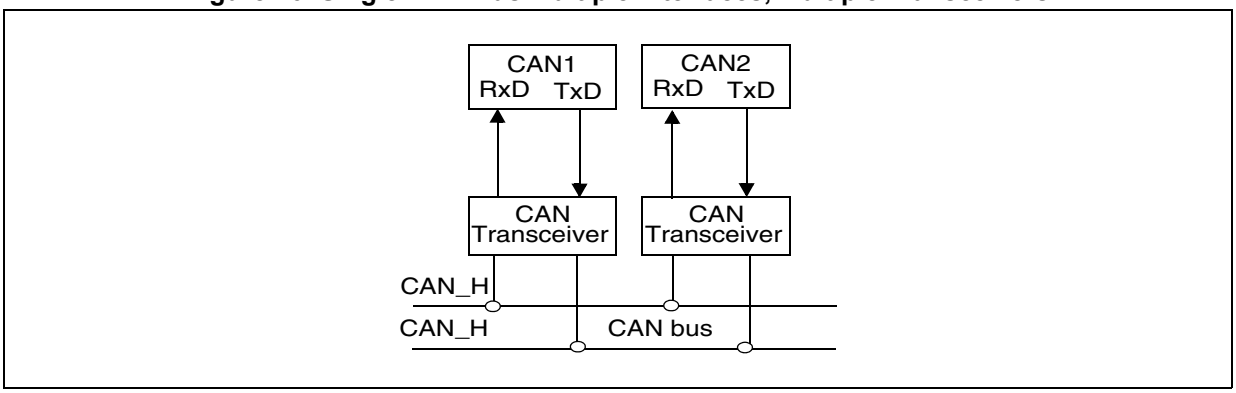
15.2 CAN Bus Configurations

Depending on 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 ST10F269 is able to support these 2 cases.

15.2.1 Single CAN Bus

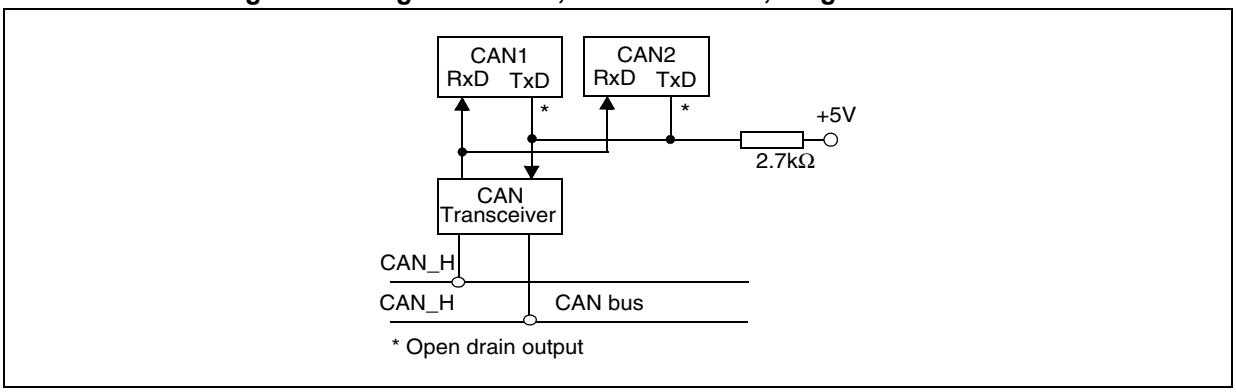
The single CAN Bus multiple interfaces configuration may be implemented using 2 CAN transceivers as shown in *Figure 46*.

Figure 46. Single CAN Bus Multiple Interfaces, Multiple Transceivers



The ST10F269 also supports single CAN Bus multiple (dual) interfaces using the open drain option of the CANx_TxD output as shown in *Figure 47*. 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.

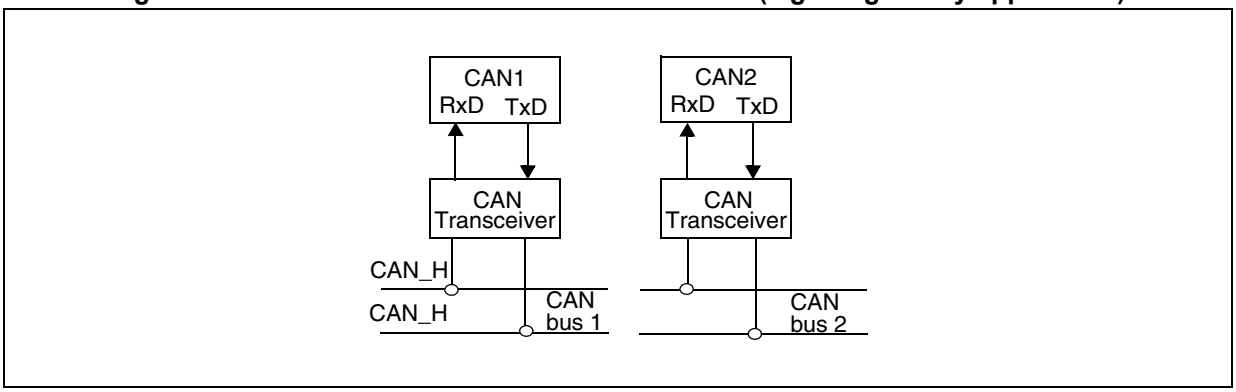
Figure 47. Single CAN Bus, Dual Interfaces, Single Transceiver



15.2.2 Multiple CAN Bus

The ST10F269 provides 2 CAN interfaces to support such kind of bus configuration as shown in *Figure 48*.

Figure 48. Connection to Two Different CAN Buses (e.g. for gateway application)



16 Real Time Clock

The Real Time Clock is an independent timer, which clock is directly derived from the clock oscillator on XTAL1 input so that it can keep on running even in Idle or Power down mode (if enabled to). Registers access is implemented onto the XBUS. This module is designed for the following purposes:

- Generate the current time and date for the system
- Cyclic time based interrupt, provides Port 2 external interrupts every second and every n seconds (n is programmable) if enabled.
- 58-bit timer for long term measurement
- Capable to exit the ST10 chip from power down mode (if PWDCFG of SYSCON set) after a programmed delay.

The real time clock is base 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 coming out of a 20-bit DIVIDER (4-bit MSB RTCDH counter and 16-bit LSB RTCDL counter). This 20-bit counter is driven by an input clock derived from the on-chip high frequency CPU clock, predivided by a 1/64 fixed counter (see *Figure 50*). This 20-bit counter is loaded at each basic reference clock period with the value of the 20-bit PRESCALER register (4-bit MSB RTCPH register and 16-bit LSB RTCPL 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 (16-bit RTCH and 16-bit RTCL). This counter may be initialized with the current system time. RTCH/RTCL counter is driven with the basic reference clock signal. In order to provide an alarm function the contents of RTCH/RTCL counter is compared with a 32-bit alarm register (16-bit RTCAH register and 16-bit RTCAL register). The alarm register may be loaded with a reference date. An alarm interrupt request (RTCAI), may be generated when the value of RTCH/RTCL counter matches the reference date of RTCAH/RTCAL register.

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

Figure 49. ESFRs and Port Pins Associated with the RTC

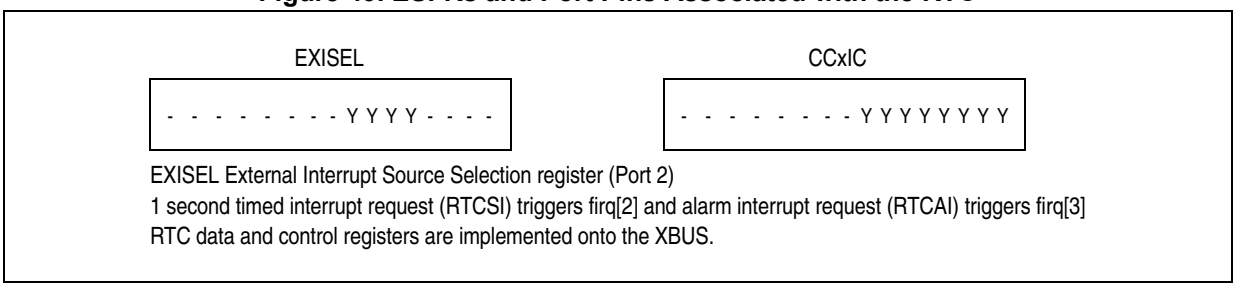
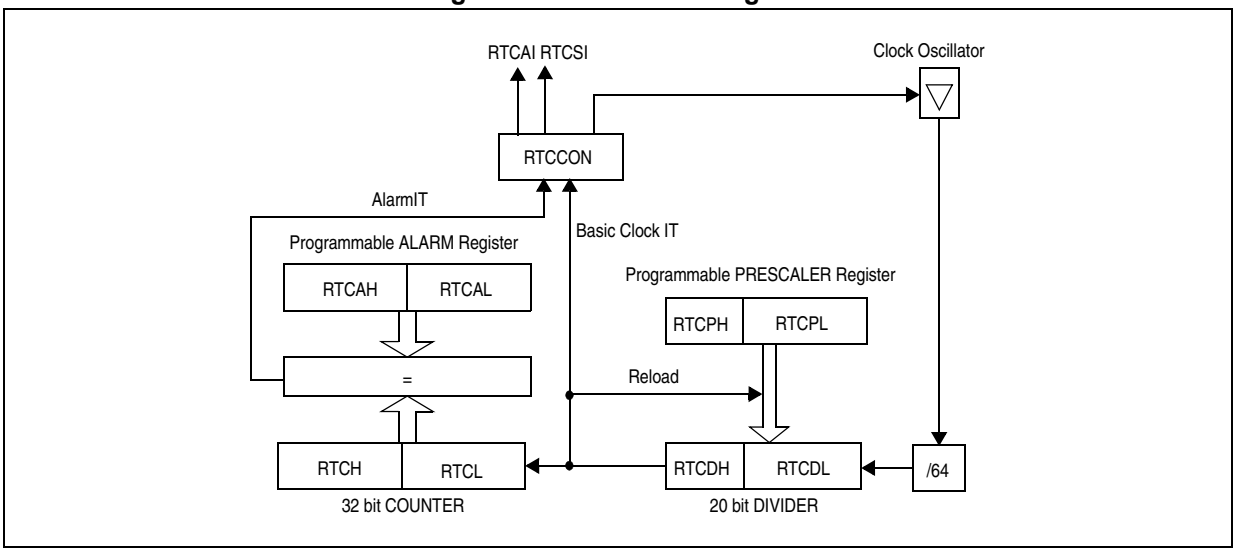


Figure 50. RTC Block Diagram



16.1 RTC registers

16.1.1 RTCCON: RTC Control Register

The functions of the RTC are controlled by the RTCCON control register. If the RTOFF bit is set, the RTC dividers and counters clock is disabled and registers can be written, when the ST10 chip enters power down mode the clock oscillator will be switch off. The RTC has 2 interrupt sources, one is triggered every basic clock period, the other one is the alarm. RTCCON includes an interrupt request flag and an interrupt enable bit for each of them. This register is read and written via the XBUS.

RTCCON (EC00h)

XBUS Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	RTCOFF	-	-	-	RTCAEN	RTCAIR	RTCSEN	RTCSIR
								RW				RW	RW	RW	RW

RTCOFF [2]	**RTC Switch Off Bit** '0': clock oscillator and RTC keep on running even if ST10 in power down mode'1': clock oscillator is switch off if ST10 enters power down mode, RTC dividers and counters are stopped and registers can be written
RTCAEN [2]	**RTC Alarm Interrupt ENable** '0': RTCAI is disabled '1': RTCAI is enabled, it is generated every n seconds
RTCAIR [1]	**RTC Alarm Interrupt Request flag (when the alarm is triggered)** '0': the bit was reseted less than a n seconds ago '1': the interrupt was triggered
RTCSEN [2]	**RTC Second interrupt ENable** '0': RTCSI is disabled '1': RTCSI is enabled, it is generated every second
RTCSIR [1]	**RTC Second Interrupt Request flag (every second)** '0': the bit was reseted less than a second ago '1': the interrupt was triggered

1. As RTCCON register is not bit-addressable, the value of these bits must be read by checking their associated CCxIC register. The 2 RTC interrupt signals are connected to Port2 in order to trigger an external interrupt that wake up the chip when in power down mode.

2. All the bit of RTCCON are active high.

16.1.2 RTCPH & RTCPL: RTC PRESCALER Registers

The 20-bit programmable prescaler divider is loaded with 2 registers.

The 4 most significant bit are stored into RTCPH and the 16 Less significant bit are stored in RTCPL. In order to keep the system clock, those registers are not reset.

They are write protected by bit RTOFF of RTCCON register, write operation is allowed if RTOFF is set.

RTCPL (EC06h)

XBUS Reset Value: XXXXH

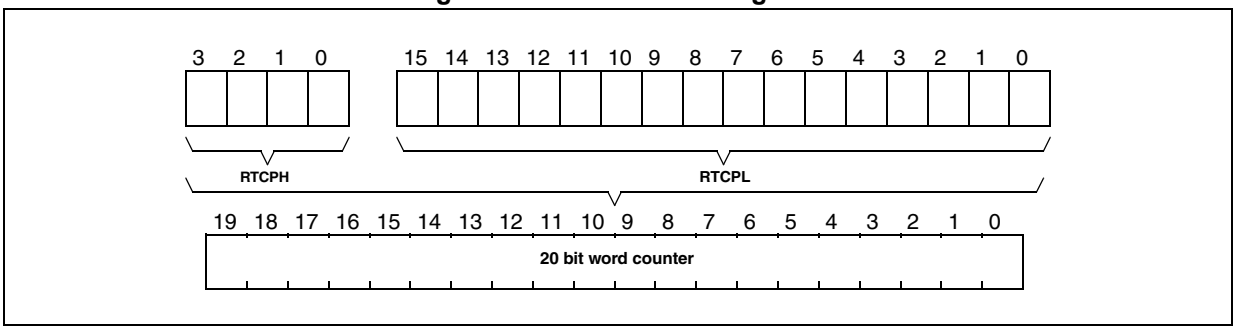
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCPL															
RW															

RTCPH (EC08h)

XBUS Reset Value: ---XH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RTCPH			
												RW			

Figure 51. PRESCALER Register



The value stored into RTCPH, RTCPL is called RTCP (coded on 20-bit). The dividing ratio of the Prescaler divider is: ratio = 64 x (RTCP)

16.1.3 RTCDH & RTCDL: RTC DIVIDER Counters

Every basic reference clock the DIVIDER counters are reloaded with the value stored RTCPH and RTCPL registers. To get an accurate time measurement it is possible to read the value of the DIVIDER, reading the RTCDH, RTCDL. Those counters are read only. After any bit changed in the programmable PRESCALER register, the new value is loaded in the DIVIDER.

RTCDL (EC0Ah)

XBUS Reset Value: XXXXH

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

RTCDH (EC0Ch)

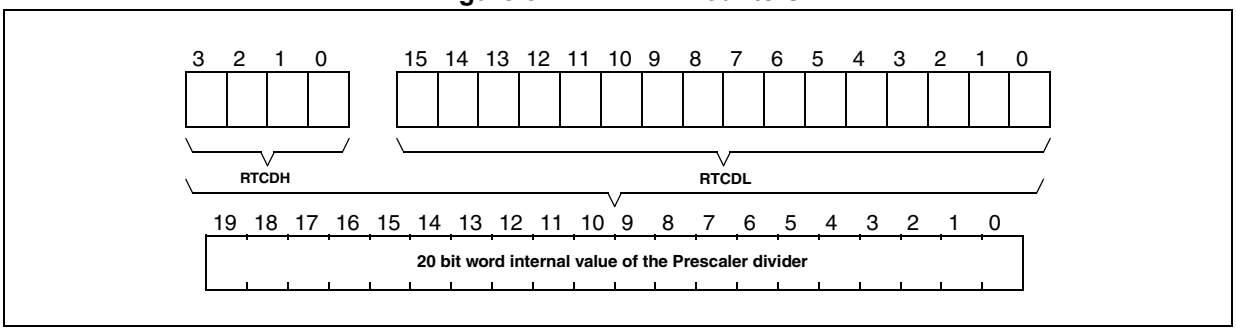
XBUS Reset Value: ---XH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RTCDH			
												R			

Those registers are not reset, and are read only.

When RTCD increments to reach 00000h, The 20-bit word stored into RTCPH, RTCPL registers is loaded in RTCD.

Figure 52. DIVIDER Counters



Bit 15 to bit 4 of RTCPH and RTCDH are not used. When reading, the return value of those bit will be zeros.

16.1.4 RTCH & RTCL: RTC Programmable COUNTER Registers

The RTC has 2 x 16-bit programmable counters which count rate is based on the basic time reference (for example 1 second). As the clock oscillator may be kept working, even in power down mode, the RTC counters may be used as a system clock. In addition RTC counters and registers are not modified at any system reset. The only way to force their value is to write them via the XBUS.

Those counters are write protected as well. The bit RTOFF of the RTCCON register must be set (RTC dividers and counters are stopped) to enable a write operation on RTCH or RTCL.

A write operation on RTCH or RTCL register loads directly the corresponding counter. When reading, the current value in the counter (system date) is returned.

The counters keeps on running while the clock oscillator is working.

RTCL (EC0Eh)

XBUS Reset Value: XXXXH

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

RTCH (EC10h)

XBUS Reset Value: XXXXH

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

Note: Those registers are nor reset.

16.1.5 RTCAH & RTCAL: RTC ALARM Registers

When the programmable counters reach the 32-bit value stored into RTCAH & RTCAL registers, an alarm is triggered and the interrupt request RTAIR is generated. Those registers are not protected.

RTCAL (EC12h)

XBUS Reset Value: XXXXH

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

RTCAH (EC14h)

XBUS Reset Value: XXXXH

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

Note: Those registers are not reset.

16.2 Programming the RTC

RTC interrupt request signals are connected to Port2, pad 10 (RTCSI) and pad 11 (RTCAI). An alternate function Port2 is to generate fast interrupts firq[7:0]. To trigger firq[2] and firq[3] the following configuration has to be set.

EXICON ESFR controls the external interrupt edge selection, RTC interrupt requests are rising edge active.

EXICON (F1C0h)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXI7ES		EXI6ES		EXI5ES		EXI4ES		EXI3ES [1 2]		EXI2ES [1 3]		EXI1ES		EXI0ES	
RW		RW		RW		RW		RW		RW		RW		RW	

1. EXI2ES and EXI3ES must be configured as "01b" because RCT interrupt request lines are rising edge active.
2. Alarm interrupt request line (RTCAI) is linked with EXI3ES.
3. Timed interrupt request line (RTCSI) is linked with EXI2ES.EXISEL ESFR enables the Port2 alternate sources. RTC interrupts are alternate sources 2 and 3.

EXISEL (F1DAh)

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXI7SS		EXI6SS		EXI5SS		EXI4SS		EXI3SS [2]		EXI2SS [3]		EXI1SS		EXI0SS	
RW		RW		RW		RW		RW		RW		RW		RW	

EXIxSS	**External Interrupt x Source Selection (x=7...0)** ‘00’: Input from associated Port 2 pin. ‘01’: Input from “alternate source”. [1] ‘10’: Input from Port 2 pin ORed with “alternate source”. [1] ‘11’: Input from Port 2 pin ANDed with “alternate source”.

1. Advised configuration.
2. Alarm interrupt request (RTCAI) is linked with EXI3SS.
3. Timed interrupt request (RTCSI) is linked with EXI2SS.

Interrupt control registers are common with CAPCOM1 Unit: CC10IC (RTCSI) and CC11IC (RTCAI).

CCxIC

SFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	CCxIR	CCxIE	ILVL				GLVL	
								RW	RW	RW				RW	

CC10IC: FF8Ch/C6h

CC11IC: FF8Eh/C7h

Source of interrupt	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
External interrupt 2	CC10IR	CC10IE	CC10INT	00’0068h	1Ah/26
External interrupt 3	CC11IR	CC11IE	CC11INT	00’006Ch	1Bh/27

17 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.

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

WDTCON (FFAEh / D7h)

SFR Reset Value: 00xxH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTREL								-	-	PONR	LHWR	SHWR	SWR	WDTR	WDTIN
RW										HR	HR	HR	HR	HR	RW

WDTIN	**Watchdog Timer Input Frequency Selection** '0': Input Frequency is $f_{CPU}/2$. '1': Input Frequency is $f_{CPU}/128$.
WDTR[1-3]	**Watchdog Timer Reset Indication Flag** Set by the watchdog timer on an overflow. Cleared by a hardware reset or by the SRVWDT instruction.
SWR[1-3]	**Software Reset Indication Flag** Set by the SRST execution. Cleared by the EINIT instruction.
SHWR[1-3]	**Short Hardware Reset Indication Flag** Set by the input $\overline{\text{RSTIN}}$. Cleared by the EINIT instruction.
LHWR[1-3]	**Long Hardware Reset Indication Flag** Set by the input $\overline{\text{RSTIN}}$. Cleared by the EINIT instruction.
PONR[1- 2-3]	**Power-On (Asynchronous) Reset Indication Flag** Set by the input $\overline{\text{RSTIN}}$ if a power-on condition has been detected. Cleared by the EINIT instruction.

1. More than one reset indication flag may be set. After EINIT, all flags are cleared.
2. Power-on is detected when a rising edge from V_{DD} = 0 V to V_{DD} > 2.0 V is recognized on the internal 3.3V supply.
3. These bits cannot be directly modified by software.

The PONR flag of WDTCON register is set if the output voltage of the internal 3.3V supply falls below the threshold (typically 2 V) of the power-on detection circuit. This circuit is efficient to detect major failures of the external 5 V supply but if the internal 3.3 V supply does not drop under 2 V, the PONR flag is not set. This could be the case on fast switch-off / switch-on of the 5 V supply. The time needed for such a sequence to activate the PONR flag depends on the value of the capacitors connected to the supply and on the exact value of the internal threshold of the detection circuit.

Table 36. WDTCON Bit Value on Different Resets

Reset Source	PONR	LHWR	SHWR	SWR	WDTR
Power On Reset	X	X	X	X	-
Power on after partial supply failure	1) 2)	X	X	X	-
Long Hardware Reset	-	X	X	X	-
Short Hardware Reset	-	-	X	X	-
Software Reset	-	-	-	X	-
Watchdog Reset	-	-	-	X	X

1. PONR bit may not be set for short supply failure.

2. For power-on reset and reset after supply partial failure, asynchronous reset must be used.

In case of bi-directional reset is enabled, and if the $\overline{\text{RSTIN}}$ pin is latched low after the end of the internal reset sequence, then a Short hardware reset, a software reset or a watchdog reset will trigger a Long hardware reset. Thus, Reset Indications flags will be set to indicate a Long Hardware Reset.

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

Table 37 shows the watchdog time range for 25 MHz CPU clock and *Table 38* shows the watchdog time range for 32 MHz CPU clock.

Table 37. WDTREL Reload Value (f_{CPU} = 40 MHz)

Reload value in WDTREL	Prescaler	
	2 (WDTIN = '0')	128 (WDTIN = '1')
FFh	12.8µs	819.2ms
00h	3.276ms	209.7ms

Table 38. WDTREL Reload Value (f_{CPU} = 3 2MHz)

Reload value in WDTREL	Prescaler	
	2 (WDTIN = '0')	128 (WDTIN = '1')
FFh	16.0µs	1.024ms
00h	4.096ms	262.1ms

The watchdog timer period is calculated with the following formula:

$$P_{WDT} = \frac{1}{f_{CPU}} \times 512 \times (1 + [WDTIN] \times 63) \times (256 - [WDTREL])$$

18 System Reset

System reset initializes the MCU in a predefined state. There are five ways to activate a reset state. The system start-up configuration is different for each case as shown in *Table 39*.

Table 39. Reset Event Definition

Reset Source	Short-cut	Conditions
Power-on reset	PONR	Power-on
Long Hardware reset (synchronous & asynchronous)	LHWR	$t_{\overline{RSTIN}}$ > 1040 TCL
Short Hardware reset (synchronous reset)	SHWR	4 TCL < $t_{\overline{RSTIN}}$ ≤ 1038 TCL
Watchdog Timer reset	WDTR	WDT overflow
Software reset	SWR	SRST execution

18.1 Long Hardware Reset

The reset is triggered when $\overline{RSTIN}$ pin is pulled low, then the MCU is immediately forced in reset default state. It pulls low $\overline{RSTOUT}$ pin, it cancels pending internal hold states if any, it aborts external bus cycle, it switches buses (data, address and control signals) and I/O pin drivers to high-impedance, it pulls high PORT0 pins and the reset sequence starts.

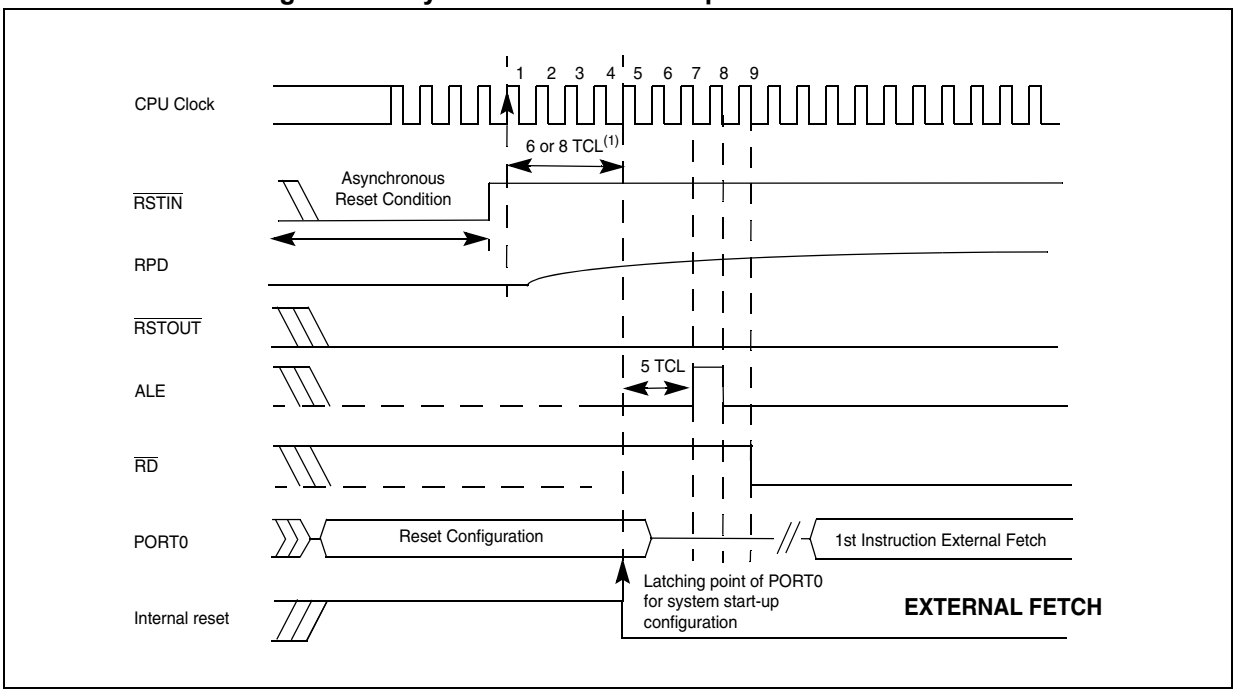
To get a long hardware reset, the duration of the external $\overline{RSTIN}$ signal must be longer than 1040 TCL. The level of RPD pin is sampled during the whole $\overline{RSTIN}$ pulse duration. A low level on RPD pin determines an asynchronous reset while a high level leads to a synchronous reset.

Note: A reset can be entered as synchronous and exit as asynchronous if V_{RPD} voltage drops below the RPD pin threshold (typically 2.5 V for V_{DD} = 5 V) when $\overline{RSTIN}$ pin is low or when $\overline{RSTIN}$ pin is internally pulled low.

18.1.1 Asynchronous Reset

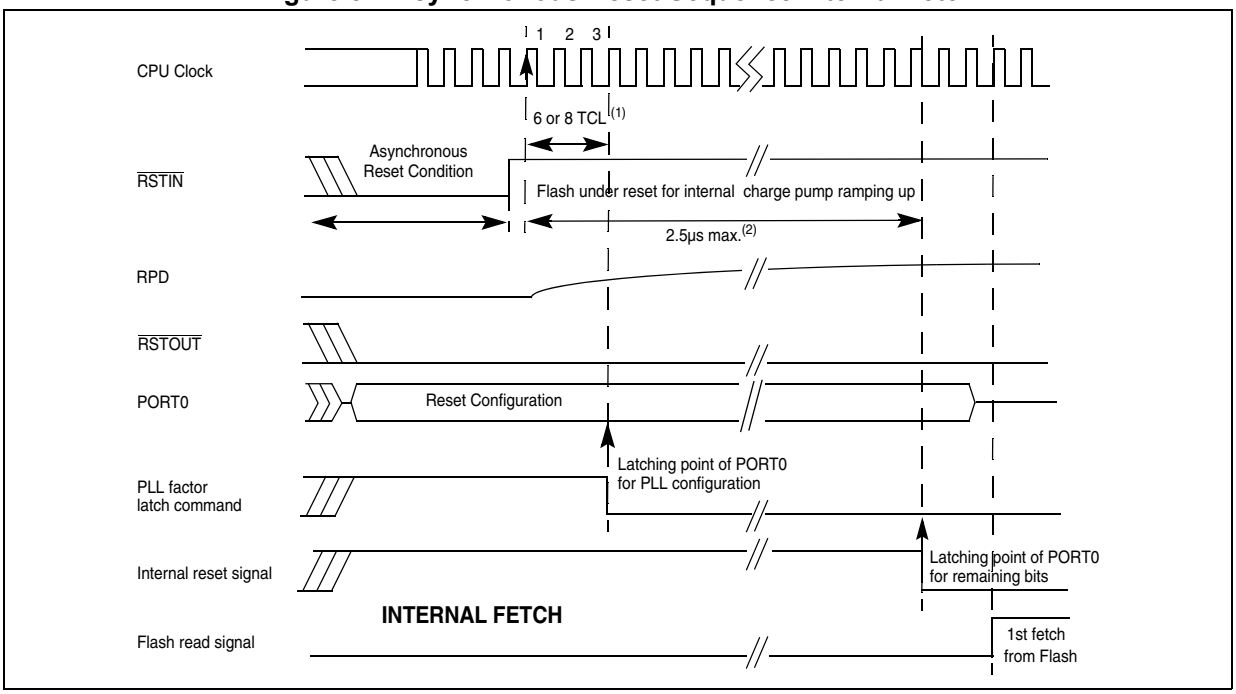
Figure 53 and *Figure 54* show asynchronous reset condition (RPD pin is at low level).

Figure 53. Asynchronous Reset Sequence External Fetch



1. $\overline{\text{RSTIN}}$ rising edge to internal latch of PORT0 is 3 CPU clock cycles (6 TCL) if the PLL is bypassed and the prescaler is on ($f_{CPU} = f_{XTAL} / 2$), else it is 4 CPU clock cycles (8 TCL).

Figure 54. Asynchronous Reset Sequence Internal Fetch



1. $\overline{\text{RSTIN}}$ rising edge to internal latch of PORT0 is 3 CPU clock cycles (6 TCL) if the PLL is bypassed and the prescaler is on ($f_{CPU} = f_{XTAL} / 2$), else it is 4 CPU clock cycles (8 TCL).
2. 2.1 µs typical value.

Power-on reset

The asynchronous reset must be used during the power-on of the MCU. Depending on the crystal frequency, the on-chip oscillator needs about 10ms to 50ms to stabilize. The logic of the MCU 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 RSTIN pin and the RPD pin must be held at low level until the MCU clock signal is stabilized and the system configuration value on PORT0 is settled.

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 *Section 18.6: Reset Circuitry*, *Figure 57*, *Figure 58* and *Figure 59*.

18.1.2 Synchronous Reset (RSTIN pulse > 1040TCL and RPD pin at high level)

The synchronous reset is a warm reset. It may be generated synchronously to the CPU clock. To be detected by the reset logic, the RSTIN pulse must be low at least for 4 TCL (2 periods of CPU clock).

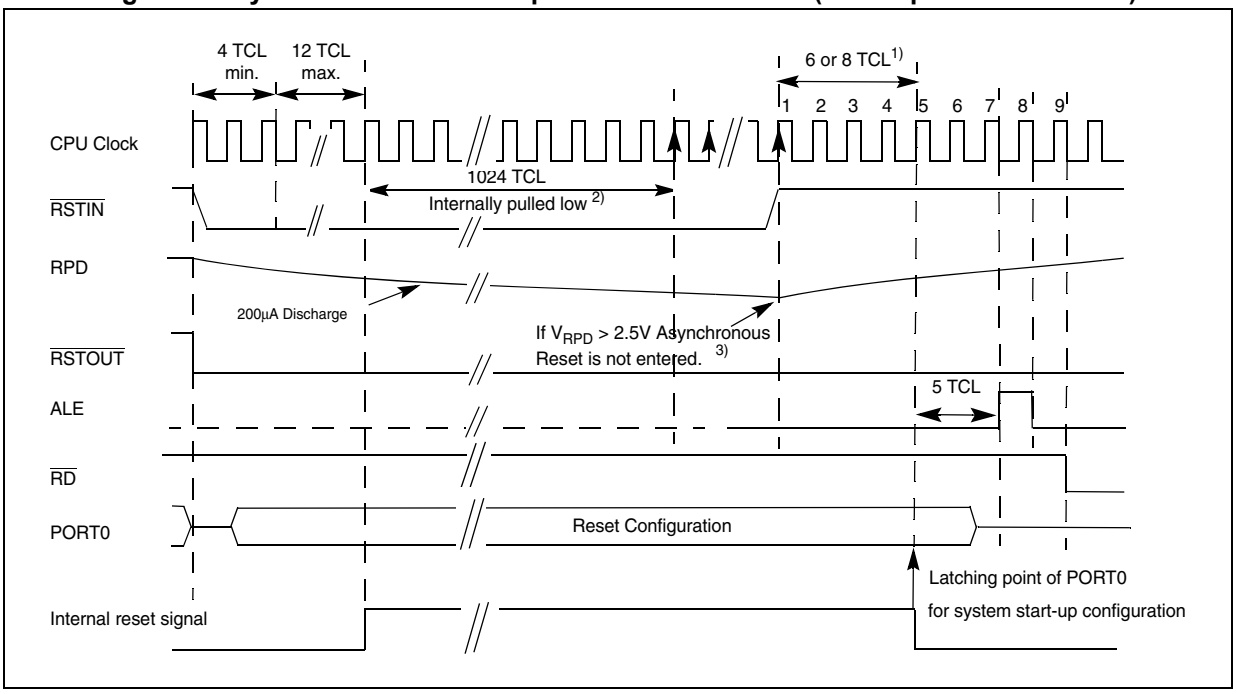
Then the I/O pins are set to high impedance and RSTOUT pin is driven low. After the RSTIN level is detected, a short duration of 12 TCL (6 CPU clocks) maximum elapses, during which pending internal hold states are canceled and the current internal access cycle, if any, is completed. External bus cycle is aborted.

The internal pull-down of RSTIN pin is activated if bit BDRSTEN of SYSCON register was previously set by software. This bit is always cleared on power-on or after any reset sequence.

The internal sequence lasts for 1024 TCL (512 periods of CPU clock). After this duration the pull-down of RSTIN pin for the bidirectional reset function is released and the RSTIN pin level is sampled. At this step the sequence lasts 1040 TCL (4 TCL + 12 TCL + 1024 TCL). If the RSTIN pin level is low, the reset sequence is extended until RSTIN level becomes high. Refer to *Figure 55*.

Note: If VRPD voltage drops below the RPD pin threshold (typically 2.5 V for V_{DD} = 5 V) when RSTIN pin is low or when RSTIN pin is internally pulled low, the ST10 reset circuitry disables the bidirectional reset function and RSTIN pin is no more pulled low. The reset is processed as an asynchronous reset.

Figure 55. Synchronous Reset Sequence External Fetch ($\overline{\text{RSTIN}}$ pulse > 1040 TCL)



1. $\overline{\text{RSTIN}}$ rising edge to internal latch of PORT0 is 3 CPU clock cycles (6 TCL) if the PLL is bypassed and the prescaler is on ($f_{CPU} = f_{XTAL} / 2$), else it is 4 CPU clock cycles (8 TCL).
2. $\overline{\text{RSTIN}}$ pin is pulled low if bit BDRSTEN (bit 3 of SYSCON register) was previously set by software. Bit BDRSTEN is cleared after reset.
3. If during the reset condition ($\overline{\text{RSTIN}}$ low), VRPD voltage drops below the threshold voltage (typically 2.5 V for 5 V operation), the ST10 reset circuitry disables the bidirectional reset function and RSTIN pin is no more pulled low.

18.1.3 Exit of Long Hardware Reset

If the RPD pin level is low when the $\overline{\text{RSTIN}}$ pin is sampled high, the MCU completes an asynchronous reset sequence.

If the RPD pin level is high when the $\overline{\text{RSTIN}}$ pin is sampled high, the MCU completes a synchronous reset sequence.

The system configuration is latched from PORT0 after a duration of 8 TCL / 4 CPU clocks (6 TCL / 3 CPU clocks if PLL is bypassed) and in case of external fetch, ALE, $\overline{\text{RD}}$ and $\overline{\text{R/W}}$ pins are driven to their inactive level. The MCU starts program execution from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. Refer to *Table 40* for PORT0 latched configuration.

18.2 Short Hardware Reset

A short hardware reset is a warm reset. It may be generated synchronously to the CPU clock (synchronous reset).

The short hardware is triggered when $\overline{\text{RSTIN}}$ signal duration is shorter or equal to 1038 TCL, the RPD pin must be pulled high.

To properly activate the internal reset logic of the MCU, the $\overline{\text{RSTIN}}$ pin must be held low, at least, during 4 TCL (2 periods of CPU clock). The I/O pins are set to high impedance and $\overline{\text{RSTOUT}}$ pin is driven low. After $\overline{\text{RSTIN}}$ level is detected, a short duration of 12 TCL (6 CPU

clocks) maximum elapses, during which pending internal hold states are canceled and the current internal access cycle if any is completed. External bus cycle is aborted. The internal pull-down of RSTIN pin is activated if bit BDRSTEN of SYSCON register was previously set by software. This bit is always cleared on power-on or after any reset sequence.

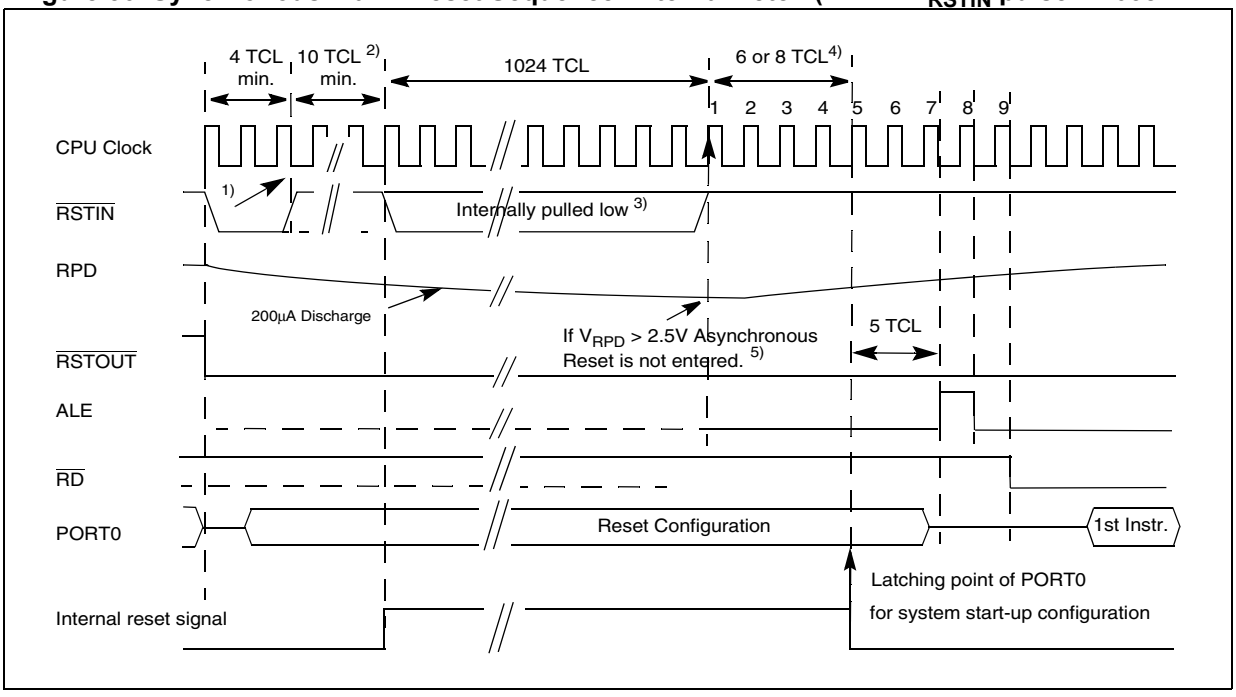
The internal reset sequence starts for 1024 TCL (512 periods of CPU clock).

After that duration the pull-down of RSTIN pin for the bidirectional reset function is released and the RSTIN pin level is sampled high while RPD level is high.

The short hardware reset ends and the MCU restarts.To be processed as a short hardware reset, the external RSTIN signal must last a maximum of 1038 TCL (4 TCL + 10 TCL + 1024 TCL). The system configuration is latched from PORT0 after a duration of 8 TCL / 4 CPU clocks (6 TCL / 3 CPU clocks if PLL is bypassed) and in case of external fetch, ALE, RD and R/W pins are driven to their inactive level. Program execution starts from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. Timings of synchronous reset sequence are summarized in *Figure 56*. Refer to *Table 40* for PORT0 latched configuration.

Note: If the RSTIN pin level is sampled low, the reset sequence is extended until RSTIN level becomes high leading to a long hardware reset (synchronous or asynchronous reset) because RSTIN signal duration has lasted longer than 1040TCL.
If the V_{RPD} voltage has dropped below the RPD pin threshold, the reset is processed as an asynchronous reset.

Figure 56. Synchronous Warm Reset Sequence External Fetch (4 TCL < $_{RSTIN}$ pulse < 1038 TCL



1. RSTIN assertion can be released there.
2. Maximum internal synchronization is 6 CPU cycles (12 TCL).
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. RSTIN rising edge to internal latch of PORT0 is 3 CPU clock cycles (6 TCL) if the PLL is bypassed and the prescaler is on ($f_{CPU} = f_{XTAL} / 2$), else it is 4 CPU clock cycles (8 TCL).
5. If during the reset condition (RSTIN low), VRPD voltage drops below the threshold voltage (typically 2.5 V for 5 V operation), the ST10 reset circuitry disables the bidirectional reset function and RSTIN pin is no more pulled low.

18.3 Software Reset

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

Upon execution of the SRST instruction, the internal reset sequence (1024 TCL) is started. The microcontroller behavior is the same as for a short hardware reset, except that only P0.12...P0.6 bits are latched at the end of the reset sequence, while previously latched values of P0.5...P0.2 are cleared.

18.4 Watchdog Timer Reset

When the watchdog timer is not disabled during the initialization or when it is not regularly serviced during program execution it will overflow and it will 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 READY, or if READY is sampled active (low) after the programmed wait states. When READY is sampled inactive (high) after the programmed wait states the running external bus cycle is aborted. Then the internal reset sequence (1024 TCL) is started.

The microcontroller behavior is the same as for a short hardware reset, except that only P0.12...P0.6 bits are latched, while previously latched values of P0.5...P0.2 are cleared.

18.5 RSTOUT, RSTIN, Bidirectional Reset

18.5.1 RSTOUT Pin

The RSTOUT pin is driven active (low level) at the beginning of any reset sequence (synchronous/asynchronous hardware, software and watchdog timer resets). RSTOUT pin stays active low beyond the end of the initialization routine, until the protected EINIT instruction (End of Initialization) is completed.

18.5.2 Bidirectional Reset

The bidirectional reset function is enabled by setting SYSCON.BDRSTEN (bit 3). This function is disabled by any reset sequence which always clears the SYSCON.BDRSTEN bit.

It can only be enabled during the initialization routine, before EINIT instruction is completed.

If VRPD voltage drops below the RPD pin threshold (typically 2.5V for VDD = 5V) when RSTIN pin is low or when RSTIN pin is internally pulled low, the ST10 reset circuitry disables the bidirectional reset function and RSTIN pin is no more pulled low. The reset is processed as an asynchronous reset.

The bidirectional reset function is useful for external peripherals with on-chip memory because the reset signal output on RSTIN pin is de-activated before the CPU starts its first instruction fetch.

18.5.3 $\overline{\text{RSTIN}}$ pin

When the bidirectional reset function is enabled, the open-drain of the $\overline{\text{RSTIN}}$ pin is activated, pulling down the reset signal, for the duration of the internal reset sequence. See *Figure 55* and *Figure 56*. At the end of the sequence the pull-down is released and the $\overline{\text{RSTIN}}$ pin gets back its input function.

The bidirectional reset function can be used:

- to convert SW or WD resets to a hardware reset so that the configuration can be (re-) latched from PORT0.
- to make visible SW or WDT resets at $\overline{\text{RSTIN}}$ pin whenever $\overline{\text{RSTIN}}$ is the only reset signal used by the application ($\overline{\text{RSTOUT}}$ not used).
- to get a die-activated reset signal before CPU starts its first instruction fetch.

The configuration latched from PORT0 is determined by the kind of reset generated by the application. (Refer to *Table 40*).

Converting a SW or WDT reset to a hardware reset allows the PLL to re-lock or the PLL configuration to be re-latched, provided a SW or WDT reset is generated by the application program is case of PLL unlock or input clock fail.

18.6 Reset Circuitry

The internal reset circuitry is described in *Figure 57*.

An internal pull-up resistor is implemented on $\overline{\text{RSTIN}}$ pin. (50 kΩ minimum, to 250 kΩ maximum). The minimum reset time must be calculated using the lowest value. In addition, a programmable pull-down (SYSCON.BDRSTEN bit 3) drives the $\overline{\text{RSTIN}}$ pin according to the internal reset state. The $\overline{\text{RSTOUT}}$ pin provides a signals to the application. (Refer to *Section 18.5: RSTOUT, RSTIN, Bidirectional Reset*).

A weak internal pull-down is connected to the RPD pin to discharge external capacitor to V_{SS} at a rate of 100 mA to 200 mA. This Pull-down is turned on when $\overline{\text{RSTIN}}$ pin is low.

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 charges the capacitor connected to RPD pin.

If the bidirectional reset function is not used, the simplest way to reset ST10F269 is to connect external components as shown in *Figure 58*. It works with reset from application (hardware or manual) and with power-on. The value of C1 capacitor, connected on RSTIN pin with internal pull-up resistor (50 kΩ to 250 kΩ), must lead to a charging time long enough to let the internal or external oscillator and / or the on-chip PLL to stabilize.

The R0-C0 components on RPD pin are mainly implemented to provide a time delay to exit Power down mode (see *Section 19: Power Reduction Modes*). Nevertheless, they drive RPD pin level during resets and they lead to different reset modes as explained hereafter. On power-on, C0 is total discharged, a low level on RPD pin forces an asynchronous hardware reset. C0 capacitor starts to charge through R0 and at the end of reset sequence ST10F269 restarts. RPD pin threshold is typically 2.5 V.

Depending on the delay of the next applied reset, the MCU can enter a synchronous reset or an asynchronous reset. If RPD pin is below 2.5 V an asynchronous reset starts, if RPD pin is above 2.5 V a synchronous reset starts (see *Section 18.1: Long Hardware Reset* and *Section 18.2: Short Hardware Reset*).

Note that an internal pull-down is connected to RPD pin and can drive a 100mA to 200mA current. This Pull-down is turned on when RSTIN pin is low.

To properly use the bidirectional reset features, the schematic (or equivalent) of *Figure 59* must be implemented. R1-C1 only work for power-on or manual reset in the same way as explained previously. D1 diode brings a faster discharge of C1 capacitor at power-off during repetitive switch-on / switch-off sequences. D2 diode performs an OR-wired connection, it can be replaced with an open drain buffer. R2 resistor may be added to increase the pull-up current to the open drain in order to get a faster rise time on RSTIN pin when bidirectional function is activated.

The start-up configurations and some system features are selected on reset sequences as described in *Table 40* and *Table 41*.

Table 40 describes what is the system configuration latched on PORT0 in the five different reset ways. *Table 41* summarizes the state of bits of PORT0 latched in RP0H, SYSCON, BUSCON0 registers.

Figure 57. Internal (simplified) Reset Circuitry

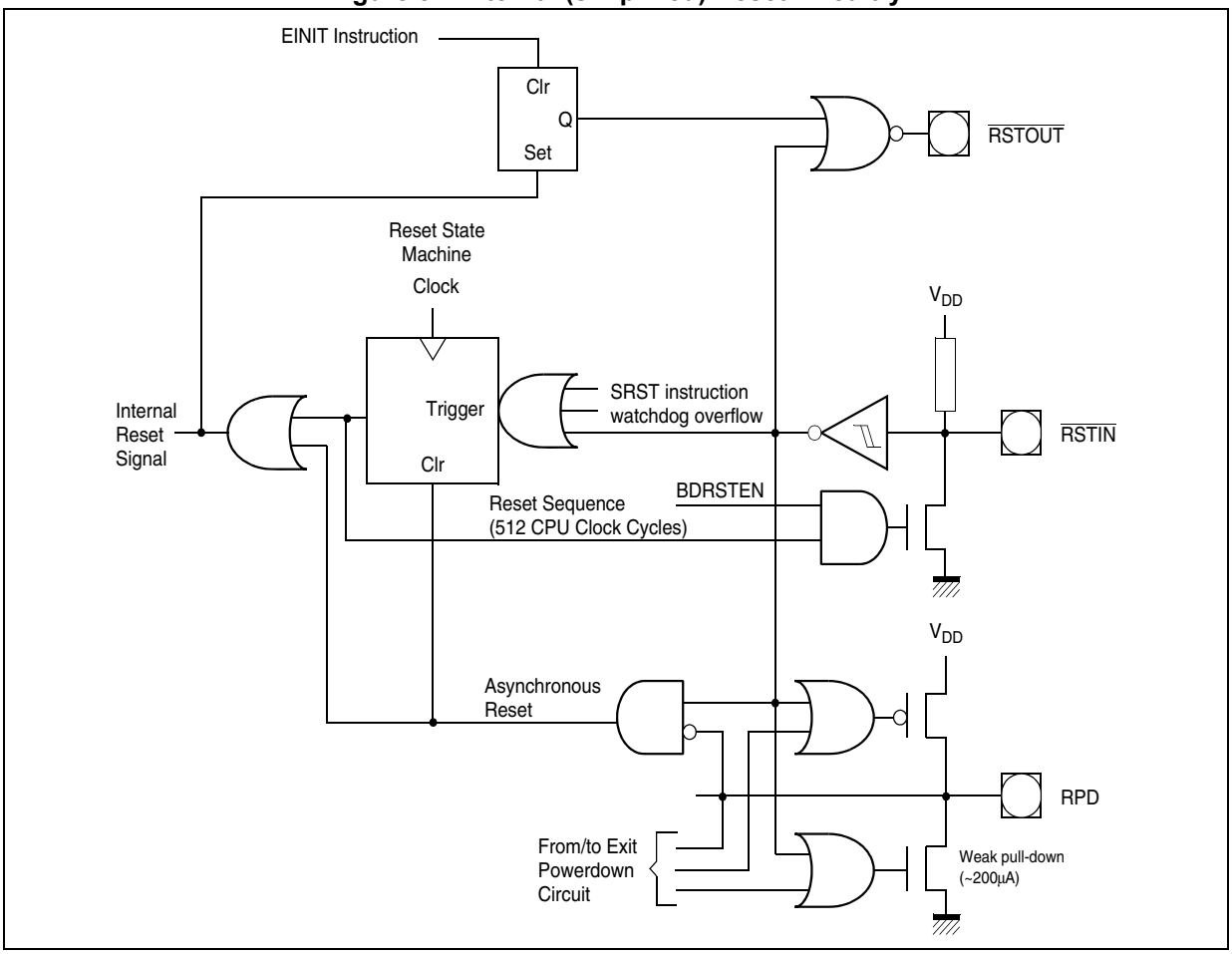


Figure 58. Minimum External Reset Circuitry

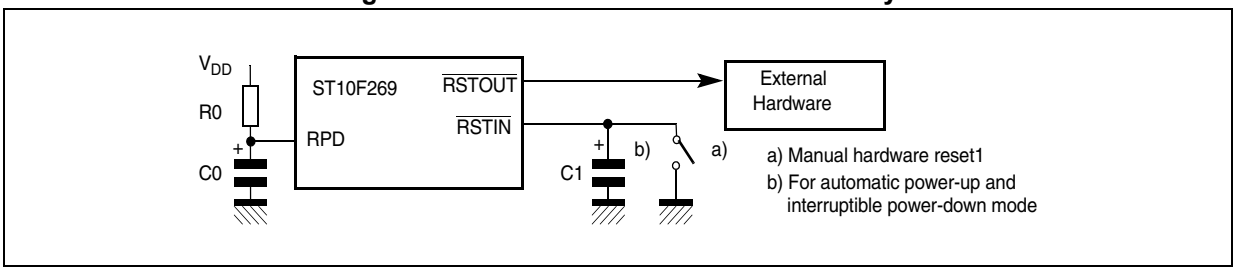


Figure 59. External Reset Hardware Circuitry

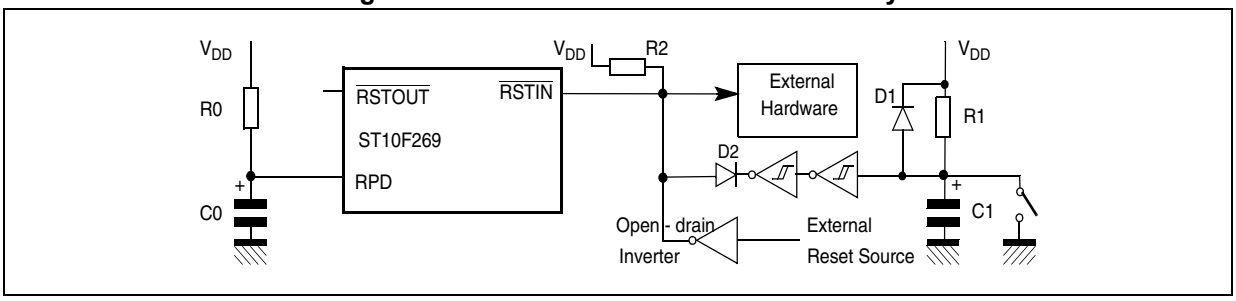


Table 40. PORT0 Latched Configuration for the Different Resets

X: Pin is sampled -: Pin is not sampled	PORT0															
	Clock Options			Seem. Add. Lines		Chip Selects		WR confide.	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	-	-	-	-	-	-
Short Hardware Reset	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X
Long Hardware Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Power-On Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 41. PORT0 Bits Latched into the Different Registers After Reset

PORT0 bit Nebr.	h7	h6	h5	h4	h3	h2	h1	h0	l7	l6	l5	l4	l3	l2	l1	l0
PORT0 bit Name	CLKCFG	CLKCFG	CLKCFG	SALSEL	SALSEL	CSSEL	CSSEL	WRC	BUSTYP	BUSTYP	R	BSL	R	R	ADP	EMU
RP0H [2]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	CLKCFG	CLKCFG	CLKCFG	SALSEL	SALSEL	CSSEL	CSSEL	WRC

Table 41. PORT0 Bits Latched into the Different Registers After Reset (continued)

PORT0 bit Nebr.	h7	h6	h5	h4	h3	h2	h1	h0	l7	l6	l5	l4	l3	l2	l1	l0
SYSCON	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	BYTDIS [3]	X [1]	WRCFG [3]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]
BUSCON0	X [1]	X [1]	X [1]	X [1]	-	BUS ACT0 [4]	ALE CTL0 [4]	-	BTYP	BTYP	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]
Internal Logic	To Clock Generator			To Port 4 Logic		To Port 6 Logic		X [1]	X [1]	X [1]	X [1]	Internal	X [1]	X [1]	Internal	Internal

1. Not latched from PORT0.
2. Only RP0H low byte is used and the bit-fields are latched from PORT0 high byte to RP0H low byte.
3. Indirectly depend on PORT0.
4. Bits set if $\overline{EA}$ pin is 1.

19 Power Reduction Modes

Two different power reduction modes with different levels of power reduction have been implemented in the ST10F269. In Idle mode only CPU is stopped, while peripheral still operate. In Power Down mode both CPU and peripherals are stopped.

Both mode are software activated by a protected instruction and are terminated in different ways as described in the following sections.

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.

19.1 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.

Note that a PEC transfer keep the CPU in Idle mode. If the PEC transfer does not succeed, the Idle mode is terminated. Watchdog timer must be properly programmed to avoid any disturbance during Idle mode.

19.2 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.

There are two different operating Power Down modes: protected mode and interruptible mode. The internal RAM contents can be preserved through the voltage supplied via the VDD pins. To verify RAM integrity, some dedicated patterns may be written before entering the Power Down mode and have to be checked after Power Down is resumed.

Caution: It is mandatory to keep V_{DD} = +5 V ± 10% during power-down mode, because the on-chip voltage regulator is turned in power saving mode and it delivers 2.5 V to the core logic, but it must be supplied at nominal V_{DD} = +5 V.

19.2.1 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 NMI pin is pulled low when executing PWRDN instruction (this means that the PWRD instruction belongs to the NMI software routine). This mode is only deactivated with an external hardware reset on RSTIN pin.

Note: During power down the on-chip voltage regulator automatically lowers the internal logic supply voltage to 2.5 V, to save power and to keep internal RAM and registers contents.

19.2.2 Interruptible Power Down Mode

This mode is selected when PWDCFG (bit 5) of SYSCON register is set (See *Section 20: Special Function Register Overview*).

The Interruptible Power Down mode is only activated if all the enabled Fast External Interrupt pins are in their inactive level (see EXICON register description below).

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. To allow the internal PLL and clock to stabilize, the $\overline{\text{RSTIN}}$ pin must be held low according the recommendations described in *Section 18: System Reset*.

EXICON (F1C0H / E0H)

XSFR Reset Value: 0000H

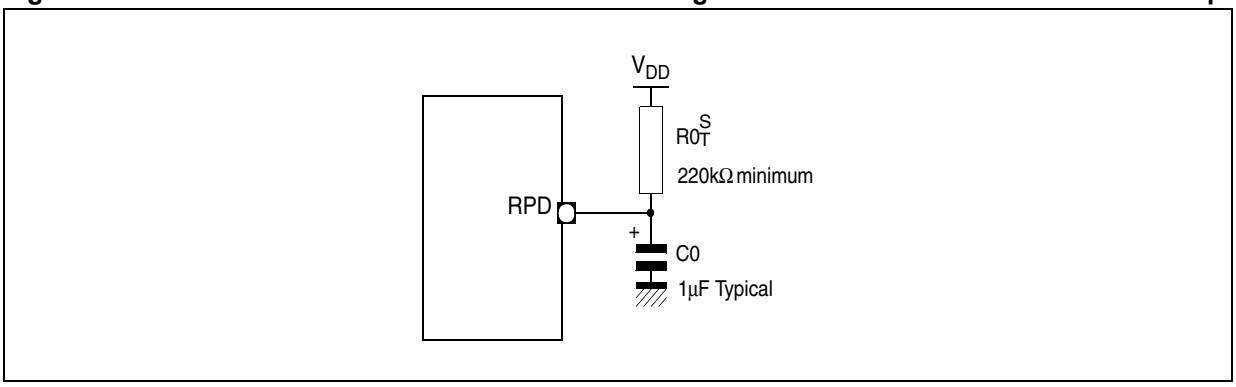
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
EXI7ES	EXI6ES	EXI5ES	EXI4ES	EXI3ES	EXI2ES	EXI1ES	EXI0ES
RW	RW	RW	RW	RW	RW	RW	RW

EXIxES(x=7...0)	**External Interrupt x Edge Selection Field (x=7...0)** 0 0: Fast external interrupts disabled: standard mode EXxIN pin not taken in account for entering/exiting Power Down mode. 0 1: Interrupt on positive edge (rising) Enter Power Down mode if EXiIN = '0', exit if EXxIN = '1' (referred as 'high' active level) 1 0: Interrupt on negative edge (falling) Enter Power Down mode if EXiIN = '1', exit if EXxIN = '0' (referred as 'low' active level) 1 1: Interrupt on any edge (rising or falling) Always enter Power Down mode, exit if EXxIN level changed.

EXxIN inputs are normally sampled interrupt inputs. However, the Power Down mode circuitry uses them as level-sensitive inputs.

An EXxIN (x = 3...0) Interrupt Enable bit (bit CCxIE in respective CCxIC register) need not be set to bring the device out of Power Down mode. An external RC circuit must be connected to RPD pin, as shown in *Figure 60*.

Figure 60. External R0C0 Circuit on RPD Pin For Exiting Powerdown Mode with External Interrupt



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

This signal enables the internal oscillator and PLL circuitry, and also turns on the weak pull-down (see *Figure 61*).

The discharge of the external capacitor provides a delay that allows the oscillator and PLL circuits to stabilize before the internal CPU and Peripheral clocks are enabled. When the RPD voltage drops below the threshold voltage (about 2.5 V), the Schmitt trigger clears Q2 flip-flop, thus enabling the CPU and Peripheral clocks, and the device resumes code execution.

If the Interrupt was enabled (bit CCxIE='1' in the respective CCxIC register) before entering Power Down mode, the device executes the interrupt service routine, and then resumes execution after the PWRDN instruction (see note below).

If the interrupt was disabled, the device executes the instruction following PWRDN instruction, and the Interrupt Request Flag (bit CCxIR in the respective CCxIC register) remains set until it is cleared by software.

Note: Due to the internal pipeline, the instruction that follows the PWRDN instruction is executed before the CPU performs a call of the interrupt service routine when exiting power-down mode

Figure 61. Simplified Powerdown Exit Circuitry

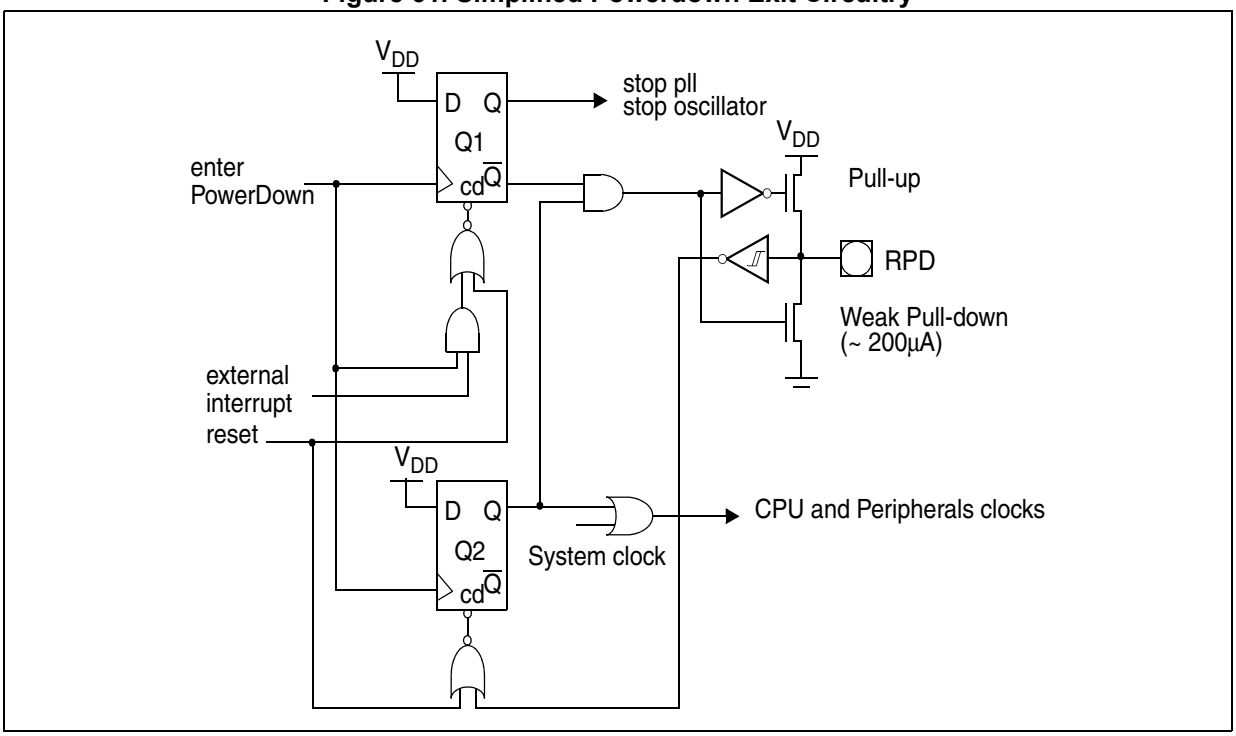
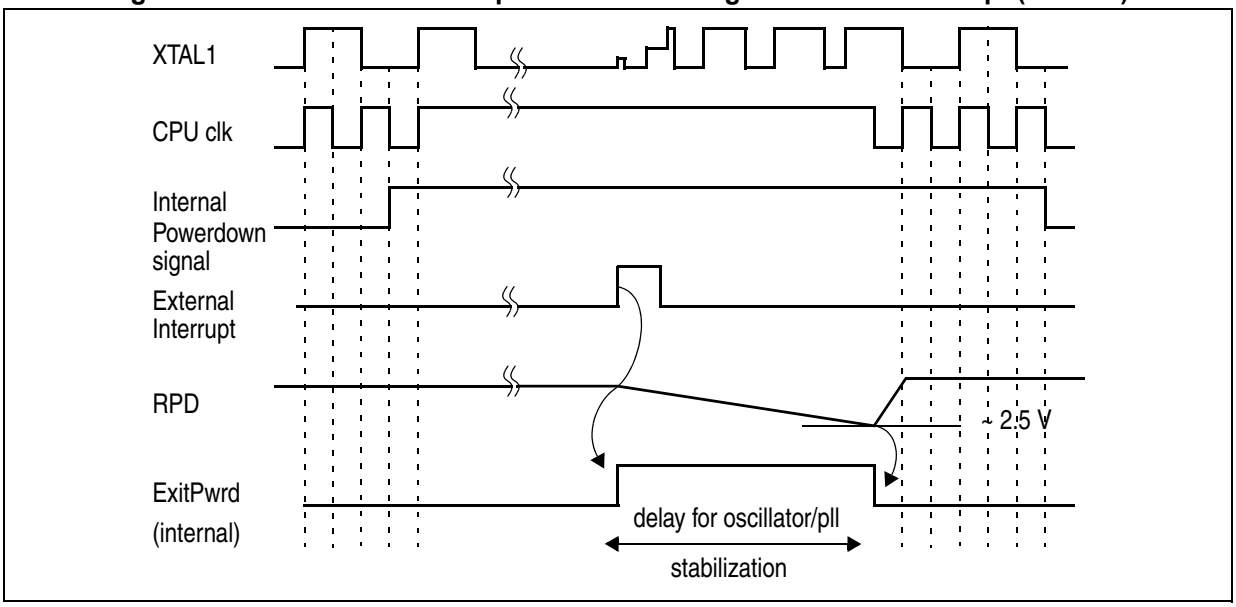


Figure 62. Powerdown Exit Sequence When Using an External Interrupt (PLL x 2)



20 Special Function Register Overview

The following table lists all SFRs which are implemented in the ST10F269 in alphabetical order. Bit-addressable SFRs are marked with the letter “b” in column “Name”. SFRs within the Extended SFR-Space (ESFRs) are marked with the letter “E” in column “Physical Address”.

A SFR can be specified by its individual mnemonic name. Depending on the selected addressing mode, a SFR can be accessed via its physical address (using the Data Page Pointers), or via its short 8-bit address (without using the Data Page Pointers).

The reset value is defined as following:

- X means the full nibble is not defined at reset.
- x means some bits of the nibble are not defined at reset.

Table 42. Special Function Registers Listed by Name

Name	Physical address	8-bit address	Description	Reset value
ADCICb	FF98h	CCh	A/D Converter end of Conversion Interrupt Control Register	- - 00h
ADCONb	FFA0h	D0h	A/D Converter Control Register	0000h
ADDAT	FEA0h	50h	A/D Converter Result Register	0000h
ADDAT2	F0A0hE	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
ADEICb	FF9Ah	CDh	A/D Converter Overrun Error Interrupt Control Register	- - 00h
BUSCON0b	FF0Ch	86h	Bus Configuration Register 0	0xx0h
BUSCON1b	FF14h	8Ah	Bus Configuration Register 1	0000h
BUSCON2b	FF16h	8Bh	Bus Configuration Register 2	0000h
BUSCON3b	FF18h	8Ch	Bus Configuration Register 3	0000h
BUSCON4b	FF1Ah	8Dh	Bus Configuration Register 4	0000h
CAPREL	FE4Ah	25h	GPT2 Capture/Reload Register	0000h
CC0	FE80h	40h	CAPCOM Register 0	0000h
CC0ICb	FF78h	BCh	CAPCOM Register 0 Interrupt Control Register	- - 00h
CC1	FE82h	41h	CAPCOM Register 1	0000h
CC1ICb	FF7Ah	BDh	CAPCOM Register 1 Interrupt Control Register	- - 00h
CC2	FE84h	42h	CAPCOM Register 2	0000h
CC2ICb	FF7Ch	BEh	CAPCOM Register 2 Interrupt Control Register	- - 00h
CC3	FE86h	43h	CAPCOM Register 3	0000h
CC3ICb	FF7Eh	BFh	CAPCOM Register 3 Interrupt Control Register	- - 00h
CC4	FE88h	44h	CAPCOM Register 4	0000h
CC4ICb	FF80h	C0h	CAPCOM Register 4 Interrupt Control Register	- - 00h
CC5	FE8Ah	45h	CAPCOM Register 5	0000h
CC5ICb	FF82h	C1h	CAPCOM Register 5 Interrupt Control Register	- - 00h
CC6	FE8Ch	46h	CAPCOM Register 6	0000h
CC6ICb	FF84h	C2h	CAPCOM Register 6 Interrupt Control Register	- - 00h

Table 42. Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
CC7	FE8Eh	47h	CAPCOM Register 7	0000h
CC7ICb	FF86h	C3h	CAPCOM Register 7 Interrupt Control Register	- - 00h
CC8	FE90h	48h	CAPCOM Register 8	0000h
CC8ICb	FF88h	C4h	CAPCOM Register 8 Interrupt Control Register	- - 00h
CC9	FE92h	49h	CAPCOM Register 9	0000h
CC9ICb	FF8Ah	C5h	CAPCOM Register 9 Interrupt Control Register	- - 00h
CC10	FE94h	4Ah	CAPCOM Register 10	0000h
CC10ICb	FF8Ch	C6h	CAPCOM Register 10 Interrupt Control Register	- - 00h
CC11	FE96h	4Bh	CAPCOM Register 11	0000h
CC11ICb	FF8Eh	C7h	CAPCOM Register 11 Interrupt Control Register	- - 00h
CC12	FE98h	4Ch	CAPCOM Register 12	0000h
CC12ICb	FF90h	C8h	CAPCOM Register 12 Interrupt Control Register	- - 00h
CC13	FE9Ah	4Dh	CAPCOM Register 13	0000h
CC13ICb	FF92h	C9h	CAPCOM Register 13 Interrupt Control Register	- - 00h
CC14	FE9Ch	4Eh	CAPCOM Register 14	0000h
CC14ICb	FF94h	CAh	CAPCOM Register 14 Interrupt Control Register	- - 00h
CC15	FE9Eh	4Fh	CAPCOM Register 15	0000h
CC15ICb	FF96h	CBh	CAPCOM Register 15 Interrupt Control Register	- - 00h
CC16	FE60h	30h	CAPCOM Register 16	0000h
CC16ICb	F160hE	B0h	CAPCOM Register 16 Interrupt Control Register	- - 00h
CC17	FE62h	31h	CAPCOM Register 17	0000h
CC17ICb	F162hE	B1h	CAPCOM Register 17 Interrupt Control Register	- - 00h
CC18	FE64h	32h	CAPCOM Register 18	0000h
CC18ICb	F164hE	B2h	CAPCOM Register 18 Interrupt Control Register	- - 00h
CC19	FE66h	33h	CAPCOM Register 19	0000h
CC19ICb	F166hE	B3h	CAPCOM Register 19 Interrupt Control Register	- - 00h
CC20	FE68h	34h	CAPCOM Register 20	0000h
CC20ICb	F168hE	B4h	CAPCOM Register 20 Interrupt Control Register	- - 00h
CC21	FE6Ah	35h	CAPCOM Register 21	0000h
CC21ICb	F16AhE	B5h	CAPCOM Register 21 Interrupt Control Register	- - 00h
CC22	FE6Ch	36h	CAPCOM Register 22	0000h
CC22ICb	F16ChE	B6h	CAPCOM Register 22 Interrupt Control Register	- - 00h
CC23	FE6Eh	37h	CAPCOM Register 23	0000h
CC23ICb	F16EhE	B7h	CAPCOM Register 23 Interrupt Control Register	- - 00h
CC24	FE70h	38h	CAPCOM Register 24	0000h
CC24ICb	F170hE	B8h	CAPCOM Register 24 Interrupt Control Register	- - 00h
CC25	FE72h	39h	CAPCOM Register 25	0000h
CC25ICb	F172hE	B9h	CAPCOM Register 25 Interrupt Control Register	- - 00h
CC26	FE74h	3Ah	CAPCOM Register 26	0000h
CC26ICb	F174hE	BAh	CAPCOM Register 26 Interrupt Control Register	- - 00h
CC27	FE76h	3Bh	CAPCOM Register 27	0000h
CC27ICb	F176hE	BBh	CAPCOM Register 27 Interrupt Control Register	- - 00h
CC28	FE78h	3Ch	CAPCOM Register 28	0000h
CC28ICb	F178hE	BCh	CAPCOM Register 28 Interrupt Control Register	- - 00h
CC29	FE7Ah	3Dh	CAPCOM Register 29	0000h

Table 42. Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
CC29ICb	F184hE	C2h	CAPCOM Register 29 Interrupt Control Register	- - 00h
CC30	FE7Ch	3Eh	CAPCOM Register 30	0000h
CC30ICb	F18ChE	C6h	CAPCOM Register 30 Interrupt Control Register	- - 00h
CC31	FE7Eh	3Fh	CAPCOM Register 31	0000h
CC31ICb	F194hE	CAh	CAPCOM Register 31 Interrupt Control Register	- - 00h
CCM0b	FF52h	A9h	CAPCOM Mode Control Register 0	0000h
CCM1b	FF54h	AAh	CAPCOM Mode Control Register 1	0000h
CCM2b	FF56h	ABh	CAPCOM Mode Control Register 2	0000h
CCM3b	FF58h	ACh	CAPCOM Mode Control Register 3	0000h
CCM4b	FF22h	91h	CAPCOM Mode Control Register 4	0000h
CCM5b	FF24h	92h	CAPCOM Mode Control Register 5	0000h
CCM6b	FF26h	93h	CAPCOM Mode Control Register 6	0000h
CCM7b	FF28h	94h	CAPCOM Mode Control Register 7	0000h
CP	FE10h	08h	CPU Context Pointer Register	FC00h
CRICb	FF6Ah	B5h	GPT2 CAPREL Interrupt Control Register	- - 00h
CSP	FE08h	04h	CPU Code Segment Pointer Register (read only)	0000h
DP0Lb	F100hE	80h	P0L Direction Control Register	- - 00h
DP0Hb	F102hE	81h	P0h Direction Control Register	- - 00h
DP1Lb	F104hE	82h	P1L Direction Control Register	- - 00h
DP1Hb	F106hE	83h	P1h Direction Control Register	- - 00h
DP2b	FFC2h	E1h	Port 2 Direction Control Register	0000h
DP3b	FFC6h	E3h	Port 3 Direction Control Register	0000h
DP4b	FFCAh	E5h	Port 4 Direction Control Register	00h
DP6b	FFCEh	E7h	Port 6 Direction Control Register	00h
DP7b	FFD2h	E9h	Port 7 Direction Control Register	00h
DP8b	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
EXICONb	F1C0hE	E0h	External Interrupt Control Register	0000h
EXISELb	F1DAhE	EDh	External Interrupt Source Selection Register	0000h
IDCHIP	F07ChE	3Eh	Device Identifier Register (n is the device revision)	10Dnh
IDMANUF	F07EhE	3Fh	Manufacturer Identifier Register	0401h
IDMEM	F07AhE	3Dh	On-chip Memory Identifier Register	3040h
IDPROG	F078hE	3Ch	Programming Voltage Identifier Register	0040h
IDX0b	FF08h	84h	MAC Unit Address Pointer 0	0000h
IDX1b	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
MCWb	FFDCh	EEh	MAC Unit Control Word	0000h
MDCb	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
MRWb	FFDAh	EDh	MAC Unit Repeat Word	0000h

Table 42. Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
MSWb	FFDEh	EFh	MAC Unit Status Word	0200h
ODP2b	F1C2hE	E1h	Port 2 Open Drain Control Register	0000h
ODP3b	F1C6hE	E3h	Port 3 Open Drain Control Register	0000h
ODP4b	F1CAhE	E5h	Port 4 Open Drain Control Register	- - 00h
ODP6b	F1CEhE	E7h	Port 6 Open Drain Control Register	- - 00h
ODP7b	F1D2hE	E9h	Port 7 Open Drain Control Register	- - 00h
ODP8b	F1D6hE	EBh	Port 8 Open Drain Control Register	- - 00h
ONESb	FF1Eh	8Fh	Constant Value 1's Register (read only)	FFFFh
P0Lb	FF00h	80h	PORT0 Low Register (Lower half of PORT0)	- - 00h
P0Hb	FF02h	81h	PORT0 High Register (Upper half of PORT0)	- - 00h
P1Lb	FF04h	82h	PORT1 Low Register (Lower half of PORT1)	- - 00h
P1Hb	FF06h	83h	PORT1 High Register (Upper half of PORT1)	- - 00h
P2b	FFC0h	E0h	Port 2 Register	0000h
P3b	FFC4h	E2h	Port 3 Register	0000h
P4b	FFC8h	E4h	Port 4 Register (8-bit)	00h
P5b	FFA2h	D1h	Port 5 Register (read only)	XXXXh
P6b	FFCCh	E6h	Port 6 Register (8-bit)	- - 00h
P7b	FFD0h	E8h	Port 7 Register (8-bit)	- - 00h
P8b	FFD4h	EAh	Port 8 Register (8-bit)	- - 00h
P5DIDISb	FFA4h	D2h	Port 5 Digital Disable Register	0000h
POCON0L	F080hE	40h	PORT0 Low Outpout Control Register (8-bit)	- - 00h
POCON0H	F082hE	41h	PORT0 High Output Control Register (8-bit)	- - 00h
POCON1L	F084hE	42h	PORT1 Low Output Control Register (8-bit)	- - 00h
POCON1H	F086hE	43h	PORT1 High Output Control Register (8-bit)	- - 00h
POCON2	F088hE	44h	Port2 Output Control Register	0000h
POCON3	F08AhE	45h	Port3 Output Control Register	0000h
POCON4	F08ChE	46h	Port4 Output Control Register (8-bit)	- - 00h
POCON6	F08EhE	47h	Port6 Output Control Register (8-bit)	- - 00h
POCON7	F090hE	48h	Port7 Output Control Register (8-bit)	- - 00h
POCON8	F092hE	49h	Port8 Output Control Register (8-bit)	- - 00h
POCON20	F0AAhE	55h	ALE, $\overline{RD}$, $\overline{WR}$ Output Control Register (8-bit)	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
PICONb	F1C4hE	E2h	Port Input Threshold Control Register	- - 00h
PP0	F038hE	1Ch	PWM Module Period Register 0	0000h
PP1	F03AhE	1Dh	PWM Module Period Register 1	0000h
PP2	F03ChE	1Eh	PWM Module Period Register 2	0000h
PP3	F03EhE	1Fh	PWM Module Period Register 3	0000h
PSWb	FF10h	88h	CPU Program Status Word	0000h

Table 42. Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
PT0	F030hE	18h	PWM Module Up/Down Counter 0	0000h
PT1	F032hE	19h	PWM Module Up/Down Counter 1	0000h
PT2	F034hE	1Ah	PWM Module Up/Down Counter 2	0000h
PT3	F036hE	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
PW2	FE34h	1Ah	PWM Module Pulse Width Register 2	0000h
PW3	FE36h	1Bh	PWM Module Pulse Width Register 3	0000h
PWMCON0b	FF30h	98h	PWM Module Control Register 0	0000h
PWMCON1b	FF32h	99h	PWM Module Control Register 1	0000h
PWMICb	F17EhE	BFh	PWM Module Interrupt Control Register	- - 00h
QR0	F004hE	02h	MAC Unit Offset Register QR0	0000h
QR1	F006hE	03h	MAC Unit Offset Register QR1	0000h
QX0	F000hE	00h	MAC Unit Offset Register QX0	0000h
QX1	F002hE	01h	MAC Unit Offset Register QX1	0000h
RP0Hb	F108hE	84h	System Start-up Configuration Register (read only)	- - XXh
S0BG	FEB4h	5Ah	Serial Channel 0 Baud Rate Generator Reload Register	0000h
S0CONb	FFB0h	D8h	Serial Channel 0 Control Register	0000h
S0EICb	FF70h	B8h	Serial Channel 0 Error Interrupt Control Register	- - 00h
S0RBUF	FEB2h	59h	Serial Channel 0 Receive Buffer Register (read only)	- - XXh
S0RICb	FF6Eh	B7h	Serial Channel 0 Receive Interrupt Control Register	- - 00h
S0TBICb	F19ChE	CEh	Serial Channel 0 Transmit Buffer Interrupt Control Register	- - 00h
S0TBUF	FEB0h	58h	Serial Channel 0 Transmit Buffer Register (write only)	0000h
S0TICb	FF6Ch	B6h	Serial Channel 0 Transmit Interrupt Control Register	- - 00h
SP	FE12h	09h	CPU System Stack Pointer Register	FC00h
SSCBR	F0B4hE	5Ah	SSC Baud Rate Register	0000h
SSCCONb	FFB2h	D9h	SSC Control Register	0000h
SSCEICb	FF76h	BBh	SSC Error Interrupt Control Register	- - 00h
SSCRB	F0B2hE	59h	SSC Receive Buffer (read only)	XXXXh
SSCRICb	FF74h	BAh	SSC Receive Interrupt Control Register	- - 00h
SSCTB	F0B0hE	58h	SSC Transmit Buffer (write only)	0000h
SSCTICb	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
SYSCONb	FF12h	89h	CPU System Configuration Register	0xx0h [1]
T0	FE50h	28h	CAPCOM Timer 0 Register	0000h
T01CONb	FF50h	A8h	CAPCOM Timer 0 and Timer 1 Control Register	0000h
T0ICb	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
T1ICb	FF9Eh	CFh	CAPCOM Timer 1 Interrupt Control Register	- - 00h
T1REL	FE56h	2Bh	CAPCOM Timer 1 Reload Register	0000h
T2	FE40h	20h	GPT1 Timer 2 Register	0000h
T2CONb	FF40h	A0h	GPT1 Timer 2 Control Register	0000h

Table 42. Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
T2ICb	FF60h	B0h	GPT1 Timer 2 Interrupt Control Register	- - 00h
T3	FE42h	21h	GPT1 Timer 3 Register	0000h
T3CONb	FF42h	A1h	GPT1 Timer 3 Control Register	0000h
T3ICb	FF62h	B1h	GPT1 Timer 3 Interrupt Control Register	- - 00h
T4	FE44h	22h	GPT1 Timer 4 Register	0000h
T4CONb	FF44h	A2h	GPT1 Timer 4 Control Register	0000h
T4ICb	FF64h	B2h	GPT1 Timer 4 Interrupt Control Register	- - 00h
T5	FE46h	23h	GPT2 Timer 5 Register	0000h
T5CONb	FF46h	A3h	GPT2 Timer 5 Control Register	0000h
T5ICb	FF66h	B3h	GPT2 Timer 5 Interrupt Control Register	- - 00h
T6	FE48h	24h	GPT2 Timer 6 Register	0000h
T6CONb	FF48h	A4h	GPT2 Timer 6 Control Register	0000h
T6ICb	FF68h	B4h	GPT2 Timer 6 Interrupt Control Register	- - 00h
T7	F050hE	28h	CAPCOM Timer 7 Register	0000h
T78CONb	FF20h	90h	CAPCOM Timer 7 and 8 Control Register	0000h
T7ICb	F17AhE	BEh	CAPCOM Timer 7 Interrupt Control Register	- - 00h
T7REL	F054hE	2Ah	CAPCOM Timer 7 Reload Register	0000h
T8	F052hE	29h	CAPCOM Timer 8 Register	0000h
T8ICb	F17ChE	BFh	CAPCOM Timer 8 Interrupt Control Register	- - 00h
T8REL	F056hE	2Bh	CAPCOM Timer 8 Reload Register	0000h
TFRb	FFACh	D6h	Trap Flag Register	0000h
WDT	FEAEh	57h	Watchdog Timer Register (read only)	0000h
WDTCONb	FFAEh	D7h	Watchdog Timer Control Register	00xxh [2]
XP0ICb	F186hE	C3h	CAN1 Module Interrupt Control Register	- - 00h [3]
XP1ICb	F18EhE	C7h	CAN2 Module Interrupt Control Register	- - 00h [3]
XP2ICb	F196hE	CBh	Flash ready/busy interrupt control register	- - 00h [3]
XP3ICb	F19EhE	CFh	PLL unlock Interrupt Control Register	- - 00h [3]
XPERCON	F024hE	12h	XPER Configuration Register	- - 05h
ZEROSb	FF1Ch	8Eh	Constant Value 0's Register (read only)	0000h

1. The system configuration is selected during reset.
2. Bit WDTR indicates a watchdog timer triggered reset.
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.

20.1 Identification Registers

The ST10F269 has four Identification registers, mapped in ESFR space. These registers contain:

- A manufacturer identifier,
- A chip identifier, with its revision,
- A internal memory and size identifier and programming voltage description.

Note: 256K and 128K versions of ST10F269 have the same IDMEM corresponding to 256K.

Both versions are based on the same device with the only difference that the two upper banks of Flash are not tested on 128K versions. Therefore, there is no way to detect by software if a device is a 128K version or a 256K version.

IDMANUF (F07Eh / 3Fh)[1]

ESFR Reset Value: 0401h

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

MANUF	**Manufacturer Identifier** - 020h: STMicroelectronics Manufacturer (JTAG worldwide normalization).

IDCHIP (F07Ch / 3Eh)[1]

ESFR Reset Value: 10DXh

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

REVID	**Device Revision Identifier**
CHIPID	**Device Identifier** - 10Dh: ST10F269 identifier.

IDMEM (F07Ah / 3Dh)[1]

ESFR Reset Value: 3040

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

MEMSIZE	**Internal Memory Size** is calculated using the following formula: Size = 4 x [MEMSIZE] (in Kbyte) - 040h for ST10F269 (256 Kbyte)
MEMTYP	**Internal Memory Type** - 3h for ST10F269 (Flash memory).

IDPROG (F078h / 3Ch)[1]

ESFR Reset Value: 0040h

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

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 ST10F269 (5V).
PROGVPP	**Programming V_{PP} Voltage** (no need of external V_{PP}) - 00h

1. All identification words are read only registers.

20.2 System Configuration Registers

The ST10F269 has registers used for different configuration of the overall system. These registers are described below.

SYSCON (FF12H / 89H)

SFR Reset Value: 0xx0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STKSZ			ROMS1	SGTDIS	ROMEN	BYTDIS	CLKEN	WRCFG	CSCFG	PWD CFG	OWD DIS	BDR STEN	XPEN	VISIBLE	XPER-SHARE
RW			RW	RW	RW[1]	RW[1]	RW	RW[1]	RW	RW	RW	RW	RW	RW	RW

1. These bit are set directly or indirectly according to PORT0 and $\overline{EA}$ pin configuration during reset sequence.

2. Register SYSCON cannot be changed after execution of the EINIT instruction.

XPER-SHARE	**XBUS Peripheral Share Mode Control** ‘0’: External accesses to XBUS peripherals are disabled ‘1’: XBUS peripherals are accessible via the external bus during hold mode
VISIBLE	**Visible Mode Control** ‘0’: Accesses to XBUS peripherals are done internally ‘1’: XBUS peripheral accesses are made visible on the external pins
XPEN	**XBUS Peripheral Enable bit** ‘0’: Accesses to the on-chip X-Peripherals and XRAM are disabled ‘1’: The on-chip X-Peripherals are enabled.
BDRSTEN	**Bidirectional Reset Enable** ‘0’: $\overline{RSTIN}$ pin is an input pin only. (SW Reset or WDT Reset have no effect on this pin) ‘1’: $\overline{RSTIN}$ pin is a bidirectional pin. This pin is pulled low during 1024 TCL during reset sequence.
OWDDIS	**Oscillator Watchdog Disable Control** ‘0’: Oscillator Watchdog (OWD) is enabled. If PLL is bypassed, the OWD monitors XTAL1 activity. If there is no activity on XTAL1 for at least 1 ms, the CPU clock is switched automatically to PLL’s base frequency (from 2 to 10MHz). ‘1’: OWD is disabled. If the PLL is bypassed, the CPU clock is always driven by XTAL1 signal. The PLL is turned off to reduce power supply current.
PWDCFG	**Power Down Mode Configuration Control** ‘0’: Power Down Mode can only be entered during PWRDN instruction execution if $\overline{NMI}$ pin is low, otherwise the instruction has no effect. Exit power down only with reset. ‘1’: Power Down Mode can only be entered during PWRDN instruction execution if all enabled fast external interrupt EXxIN pins are in their inactive level. Exiting this mode can be done by asserting one enabled EXxIN pin or with external reset.
CSCFG	**Chip Select Configuration Control** ‘0’: Latched Chip Select lines: CSx change 1 TCL after rising edge of ALE ‘1’: Unlatched Chip Select lines: CSx change with rising edge of ALE.

WRCFG	**Write Configuration Control** (Inverted copy of bit WRC of RP0H) '0': Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function '1': Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$.
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.
BYTDIS	**Disable/Enable Control for Pin $\overline{BHE}$ (Set according to data bus width)** '0': Pin $\overline{BHE}$ enabled '1': Pin $\overline{BHE}$ disabled, pin may be used for general purpose I/O.
ROMEN	**Internal Memory Enable (Set according to pin $\overline{EA}$ during reset)** '0': Internal Memory disabled: accesses to the 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 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	-	BUS ACT0	ALE CTL0	-	BTYP		MTTC0	RWDC0	MCTC			
RW	RW	RW	RW		RW[2]	RW[2]		RW[1]		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	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	RW

BUSCON4 (FF1AH / 8DH)

SFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7 6	5	4	3 2 1 0
CSWEN4	CSREN4	RDYPOL4	RDYEN4	-	BUSACT4	ALECTL4	-	BTYP	MTTC4	RWDC4	MCTC
RW	RW	RW	RW		RW	RW		RW	RW	RW	RW

1. BTYP (bit 6 and 7) are set according to the configuration of the bit I6 and I7 of PORT0 latched at the end of the reset sequence.
2. BUSCON0 is initialized with 0000h, if $\overline{EA}$ pin is high during reset. If $\overline{EA}$ pin is low during reset, bit BUSACT0 and ALECTRL0 are set ('1') and bit field BTYP is loaded with the bus configuration selected via PORT0.

MCTC	**Memory Cycle Time Control (Number of memory cycle time wait states)** 0 0 0 0: 15 wait states (Nber = 15 - [MCTC]) . . . 1 1 1 1: No wait state
RWDCx	**Read/Write Delay Control for BUSCONx** '0': With read/write delay: activate command 1 TCL after falling edge of ALE '1': No read/write delay: activate command with falling edge of ALE
MTTCx	**Memory Tristate Time Control** '0': 1 wait state '1': No wait state
BTYP	**External Bus Configuration** 0 0: 8-bit Demultiplexed Bus 0 1: 8-bit Multiplexed Bus 1 0: 16-bit Demultiplexed Bus 1 1: 16-bit Multiplexed Bus Note: For BUSCON0, BTYP bit-field is defined via PORT0 during reset.
ALECTLx	**ALE Lengthening Control** '0': Normal ALE signal '1': Lengthened ALE signal
BUSACTx	**Bus Active Control** '0': External bus disabled '1': External bus enabled (within the respective address window, see ADDRSEL)
RDYENx	**$\overline{READY}$ Input Enable** '0': External bus cycle is controlled by bit field MCTC only '1': External bus cycle is controlled by the $\overline{READY}$ input signal

RDYPOLx	**Ready Active Level Control** '0': Active level on the $\overline{\text{READY}}$ pin is low, bus cycle terminates with a '0' on READY pin, '1': Active level on the $\overline{\text{READY}}$ pin is high, bus cycle terminates with a '1' on $\overline{\text{READY}}$ pin.
CSRENx	**Read Chip Select Enable** '0': The CS signal is independent of the read command ($\overline{\text{RD}}$) '1': The CS signal is generated for the duration of the read command
CSWENx	**Write Chip Select Enable** '0': The $\overline{\text{CS}}$ signal is independent of the write command ($\overline{\text{WR}}$,$\overline{\text{WRL}}$,$\overline{\text{WRH}}$) '1': The $\overline{\text{CS}}$ signal is generated for the duration of the write command

RP0H (F108h / 84h)

ESFR Reset Value: --XXH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	CLKSEL			SALSEL		CSSEL		WRC
								R [1-2]			R [2]		R [2]		R [2]

WRC [2]	**Write Configuration Control** '0': Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$ '1': Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function
CSSEL [2]	**Chip Select Line Selection (Number of active $\overline{CS}$ outputs)** 0 0: 3 $\overline{CS}$ lines: $\overline{CS}$2...$\overline{CS}$0 0 1: 2 $\overline{CS}$ lines: $\overline{CS}$1...$\overline{CS}$0 1 0: No $\overline{CS}$ line at all 1 1: 5 $\overline{CS}$ lines: $\overline{CS}$4...$\overline{CS}$0 (Default without pull-downs)
SALSEL [2]	**Segment Address Line Selection (Number of active segment address outputs)** 0 0: 4-bit segment address: A19...A16 0 1: No segment address lines at all 1 0: 8-bit segment address: A23...A16 1 1: 2-bit segment address: A17...A16 (Default without pull-downs)
CLKSEL [1-2]	**System Clock Selection** 000: $f_{CPU} = 2.5 \times f_{OSC}$ 001: $f_{CPU} = 0.5 \times f_{OSC}$ 010: $f_{CPU} = 1.5 \times f_{OSC}$ 011: $f_{CPU} = f_{OSC}$ 100: $f_{CPU} = 5 \times f_{OSC}$ 101: $f_{CPU} = 2 \times f_{OSC}$ 110: $f_{CPU} = 3 \times f_{OSC}$ 111: $f_{CPU} = 4 \times f_{OSC}$

1. RP0H.7 to RP0H.5 bits are loaded only during a long hardware reset. As pull-up resistors are active on each Port P0H pins during reset, RP0H default value is "FFh".
2. These bits are set according to Port 0 configuration during any reset sequence.
3. RP0H is a read only register.

EXICON (F1C0H / E0H)

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXI7ES		EXI6ES		EXI5ES		EXI4ES		EXI3ES		EXI2ES		EXI1ES		EXI0ES	
RW		RW		RW		RW		RW		RW		RW		RW	

EXIxES(x=7...0)	**External Interrupt x Edge Selection Field (x=7...0)** 0 0: Fast external interrupts disabled: standard mode EXxIN pin not taken in account for entering/exiting Power Down mode. 0 1: Interrupt on positive edge (rising) Enter Power Down mode if EXiIN = '0', exit if EXxIN = '1' (referred as 'high' active level) 1 0: Interrupt on negative edge (falling) Enter Power Down mode if EXiIN = '1', exit if EXxIN = '0' (referred as 'low' active level) 1 1: Interrupt on any edge (rising or falling) Always enter Power Down mode, exit if EXxIN level changed.

EXISEL (F1DAH / EDH)

ESFR Reset Value: 0000H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXI7SS		EXI6SS		EXI5SS		EXI4SS		EXI3SS		EXI2SS		EXI1SS		EXI0SS	
RW		RW		RW		RW		RW		RW		RW		RW	

EXIxSS	**External Interrupt x Source Selection (x=7...0)** '00': Input from associated Port 2 pin. '01': Input from "alternate source". '10': Input from Port 2 pin ORed with "alternate source". '11': Input from Port 2 pin ANDed with "alternate source".

EXIxSS	Port 2 pin	Alternate Source
0	P2.8	CAN1_RxD
1	P2.9	CAN2_RxD
2	P2.10	RTCSI
3	P2.11	RTCAI
4...7	P2.12...15	Not used (zero)

XP3IC (F19EH / CFH)[1]

ESFR Reset Value: --00H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	XP3IR	XP3IE	XP3ILVL				GLVL	
								RW	RW	RW				RW	

1. XP3IC register has the same bit field as xxIC interrupt registers

xxIC (yyyyh / zzh)

SFR Area Reset Value: --00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	xxIR	xxIE	ILVL				GLVL	
								RW	RW	RW				RW	

GLVL	**Group Level** Defines the internal order for simultaneous requests of the same priority. 3: Highest group priority 0: Lowest group priority
ILVL	**Interrupt Priority Level** Defines the priority level for the arbitration of requests. Fh: Highest priority level 0h: Lowest priority level
xxIE	**Interrupt Enable Control Bit** (individually enables/disables a specific source) ‘0’: Interrupt Request is disabled ‘1’: Interrupt Request is enabled
xxIR	**Interrupt Request Flag** ‘0’: No request pending ‘1’: This source has raised an interrupt request

XPERCON (F024H / 12H)

ESFR Reset Value: --05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	RTCEN	XRAM2EN	XRAM1EN	CAN2EN	CAN1EN
											RW	RW	RW	RW	RW

CAN1EN	**CAN1 Enable Bit** ‘0’: Accesses to the on-chip CAN1 XPeripheral and its functions are disabled. P4.5 and P4.6 pins can be used as general purpose I/Os. Address range 00’EF00h-00’EFFFh is only directed to external memory if CAN2EN is also ‘0’. ‘1’: The on-chip CAN1 XPeripheral is enabled and can be accessed.
CAN2EN	**CAN2 Enable Bit** ‘0’: Accesses to the on-chip CAN2 XPeripheral and its functions are disabled. P4.4 and P4.7 pins can be used as general purpose I/Os. Address range 00’EE00h-00’EEFFh is only directed to external memory if CAN1EN is also ‘0’. ‘1’: The on-chip CAN2 XPeripheral is enabled and can be accessed.
XRAM1EN	**XRAM1 Enable Bit** ‘0’: Accesses to external memory within space 00’E000h to 00’E7FFh. The 2K Bytes of internal XRAM1 are disabled. ’1’: Accesses to the internal 2K Bytes of XRAM1.
XRAM2EN	**XRAM2 Enable Bit** ‘0’: Accesses to the external memory within space 00’C000h to 00’DFFFh. The 8K Bytes of internal XRAM2 are disabled. ’1’: Accesses to the internal 8K Bytes of XRAM2.
RTCEN	**RTC Enable Bit** ’0’: Accesses to the on-chip Real Time Clock are disabled, external access performed. Address range 00’EC00h-00’ECFFh is only directed to external memory if CAN1EN and CAN2EN are ’0’ also ’1’: The on-chip Real Time Clock is enabled and can be accessed.

When both CAN are disabled via XPERCON setting, then any access in the address range 00’EE00h - 00’EFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register. P4.4 and P4.7 can be used as General Purpose I/O when CAN2 is not enabled, and P4.5 and P4.6 can be used as General Purpose I/O when CAN1 is not enabled.

The default XPER selection after Reset is identical to XBUS configuration of ST10C167: XCAN1 is enabled, XCAN2 is disabled, XRAM1 (2-Kbyte compatible XRAM) is enabled, XRAM2 (new 8-Kbyte XRAM) is disabled.

Register XPERCON cannot be changed after the global enabling of XPeripherals, i.e. after setting of bit XPEN in SYSCON register.

In EMUlation mode, all the XPERipherals are enabled (XPERCON bit are all set).

When the Real Time Clock is disabled (RTCEN = 0), the clock oscillator is switch off if ST10 enters in power-down mode. Otherwise, when the Real Time Clock is enabled, the bit RTCOFF of the RTCCON register allows to choose the power-down mode of the clock oscillator.

21 Electrical Characteristics

21.1 Absolute Maximum Ratings

Table 43. Absolute Maximum Ratings

Symbol	Parameter	Value	Unit
V_{DD}	Voltage on V_{DD} pins with respect to ground(1)	-0.5, +6.5	V
V_{IO}	Voltage on any pin with respect to ground(1)	-0.5, (V_{DD} +0.5)	V
V_{AREF}	Voltage on V_{AREF} pin with respect to ground(1)	-0.3, (V_{DD} +0.3)	V
I_{OV}	Input Current on any pin during overload condition(1)	-10, +10	mA
I_{TOV}	Absolute Sum of all input currents during overload condition(1)	|100|	mA
P_{tot}	Power Dissipation(1)	1.5	W
T_A	Ambient Temperature under bias	-40, +125	°C
T_{stg}	Storage Temperature(1)	-65, +150	°C

1. 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 (VIN > VDD or VIN < VSS) the voltage on pins with respect to ground (VSS) must not exceed the values defined by the Absolute Maximum Ratings.

21.2 Parameter Interpretation

The parameters listed in the following tables represent the characteristics of the ST10F269 and its demands on the system. Where the ST10F269 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 ST10F269, the symbol “SR” for System Requirement, is included in the “Symbol” column.

21.3 DC Characteristics

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, Reset active, f_{CPU} = 40 MHz with T_A = -40 to +119°C or f_{CPU} = 32 MHz with T_A = -40 to +125°C

Table 44. DC Characteristics

Symbol		Parameter	Test Conditions	Min.	Max.	Unit
V_{IL}	SR	Input low voltage	–	-0.5	0.2 V_{DD} -0.1	V
V_{ILS}	SR	Input low voltage (special threshold)	–	-0.5	2.0	V
V_{IH}	SR	Input high voltage (all except $\overline{\text{RSTIN}}$ and XTAL1)	–	0.2 V_{DD} + 0.9	V_{DD} + 0.5	V
V_{IH1}	SR	Input high voltage $\overline{\text{RSTIN}}$	–	0.6 V_{DD}	V_{DD} + 0.5	V
V_{IH2}	SR	Input high voltage XTAL1	–	0.7 V_{DD}	V_{DD} + 0.5	V

Table 44. DC Characteristics (continued)

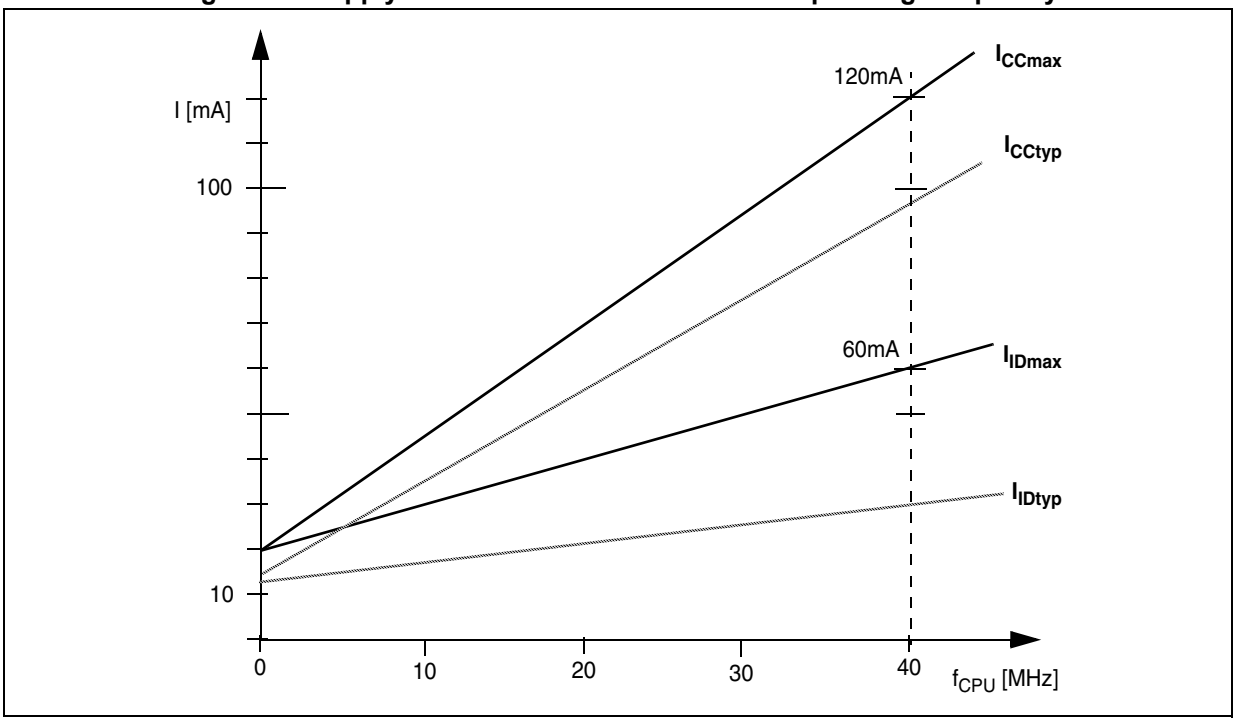
Symbol		Parameter		Test Conditions	Min.	Max.	Unit
V_{IHS}	SR	Input high voltage (special threshold)		–	0.8 V_{DD} - 0.2	V_{DD} + 0.5	V
HYS		Input Hysteresis (special threshold)	3	–	250	–	mV
V_{OL}	CC	Output low voltage (PORT0, PORT1, Port 4, ALE, RD, WR, BHE, CLKOUT, RSTOUT)	1	I_{OL} = 2.4mA	–	0.45	V
V_{OL1}	CC	Output low voltage (all other outputs)	1	I_{OL1} = 1.6mA	–	0.45	V
V_{OH}	CC	Output high voltage (PORT0, PORT1, Port4, ALE, RD, WR, BHE, CLKOUT, RSTOUT)	1	I_{OH} = -500µA I_{OH} = -2.4mA	0.9 V_{DD} 2.4	– –	V
V_{OH1}	CC	Output high voltage (all other outputs)	1/2	I_{OH} = – 250µA I_{OH} = – 1.6mA	0.9 V_{DD} 2.4	– –	V V
$\lvert I_{OZ1} \rvert$	CC	Input leakage current (Port 5)		0V < V_{IN} < V_{DD}	–	200	nA
$\lvert I_{OZ2} \rvert$	CC	Input leakage current (all other)		0V < V_{IN} < V_{DD}	–	1	µA
$\lvert I_{OV} \rvert$	SR	Overload current	3/4		–	5	mA
R_{RST}	CC	$\overline{\text{RSTIN}}$ pull-up resistor	3	–	50	250	kΩ
I_{RWH}		Read / Write inactive current	5/6	V_{OUT} = 2.4V	–	-40	µA
I_{RWL}		Read / Write active current	5/7	V_{OUT} = V_{OLmax}	-500	–	µA
I_{ALEL}		ALE inactive current	5/6	V_{OUT} = V_{OLmax}	40	–	µA
I_{ALEH}		ALE active current	5/7	V_{OUT} = 2.4V	–	500	µA
I_{P6H}		Port 6 inactive current	5/6	V_{OUT} = 2.4V	–	-40	µA
I_{P6L}		Port 6 active current	5/7	V_{OUT} = V_{OL1max}	-500	–	µA
I_{P0H}		PORT0 configuration current	5/6	V_{IN} = V_{IHmin}	–	-10	µA
I_{P0L}			5/7	V_{IN} = V_{ILmax}	-100	–	µA
$\lvert I_{IL} \rvert$	CC	XTAL1 input current		0V < V_{IN} < V_{DD}	–	20	µA
gm		On-chip oscillator transconductance	3		5	-	mA/V
C_{IO}	CC	Pin capacitance (digital inputs / outputs)	3/5	f = 1MHz, T_A = 25°C	–	10	pF
I_{CC}		Power supply current	8	RSTIN = V_{IH1} f_{CPU} in [MHz]	–	20 + 2.5 x f_{CPU}	mA
I_{ID}		Idle mode supply current	9	RSTIN = V_{IH1} f_{CPU} in [MHz]	–	20 + f_{CPU}	mA

Table 44. DC Characteristics (continued)

Symbol	Parameter		Test Conditions	Min.	Max.	Unit
I_{PD}	Power-down mode supply current	10	V_{DD} = 5.5V T_A = 25°C T_A = 85°C T_A = 125°C	– – –	15 [11] 50 [11] 190 [11]	µA µA µA
I_{PD2}	Power-down mode supply current (Real time clock enabled, oscillator enabled)	10 12	V_{DD} = 5.5V T_A = 55°C f_{OSC} = 25MHz	–	2 + f_{OSC} / 4	mA

1. ST10F269 pins are equipped with low-noise output drivers which significantly improve the device's EMI performance. These low-noise drivers deliver their maximum current only until the respective target output level is reached. After this, the output current is reduced. This results in increased impedance of the driver, which attenuates electrical noise from the connected PCB tracks. The current specified in column "Test Conditions" is delivered in any cases.
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 results from the external circuitry.
3. Partially tested, guaranteed by design characterization.
4. Overload conditions occur if the standard operating conditions are exceeded, i.e. the voltage on any pin exceeds the specified range (i.e. $V_{OV} > V_{DD}$+0.5V or $V_{OV} < -0.5V$). 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.
5. This specification is only valid during Reset, or during Hold-mode or Adapt-mode. Port 6 pins are only affected if they are used for CS output and if their open drain function is not enabled.
6. The maximum current may be drawn while the respective signal line remains inactive.
7. The minimum current must be drawn in order to drive the respective signal line active.
8. The power supply current is a function of the operating frequency. This dependency is illustrated in *Figure 63*. These parameters are tested at V_{DD}max and 40 MHz (or 32 MHz) CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}. The chip is configured with a demultiplexed 16-bit bus, direct clock drive, 5 chip select lines and 2 segment address lines, EA pin is low during reset. After reset, PORT 0 is driven with the value '00CCh' that produces infinite execution of NOP instruction with 15 wait-states, R/W delay, memory tristate wait state, normal ALE. Peripherals are not activated.
9. Idle mode supply current is a function of the operating frequency. These parameters are tested at V_{DD}max and 40 MHz (or 32 MHz) CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}.
10. This parameter value includes leakage currents. With all inputs (including pins configured as inputs) at 0V to 0.1V or at V_{DD} – 0.1V to V_{DD}, VREF = 0V, all outputs (including pins configured as outputs) disconnected.
11. Typical I_{PD} value is 5 µA @ T_A=25°C, 20 µA @ T_A=85°C and 60 µA @ T_A=125°C.
12. Partially tested, guaranteed by design characterization using 22 pF loading capacitors on crystal pins.

Figure 63. Supply / Idle Current as a Function of Operating Frequency



21.3.1 A/D Converter Characteristics

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +85°C or -40 to +125°C,
4.0 V ≤ V_{AREF} ≤ V_{DD} + 0.1V, VSS 0.1 V ≤ V_{AGND} ≤ VSS + 0.2 V

Table 45. A/D Converter Characteristics

Symbol	Parameter	Test Condition	Limit Values		Unit
			minimum	maximum	
V_{AREF}SR	Analog Reference voltage		4.0	V_{DD} + 0.1	V
V_{AIN}SR	Analog input voltage	1 - 8	V_{AGND}	V_{AREF}	V
I_{AREF}CC	Reference supply current running mode power-down mode	7	 – –	 500 1	 µA µA
C_{AIN}CC	ADC input capacitance Not sampling Sampling	7	 – –	 10 15	 pF pF
t_SCC	Sample time	2 - 4	48 TCL	1 536 TCL	
t_CCC	Conversion time	3 - 4	388 TCL	2 884 TCL	
DNLCC	Differential Nonlinearity	5	-0.5	+0.5	LSB
INLCC	Integral Nonlinearity	5	-1.5	+1.5	LSB
OFSCC	Offset Error	5	-1.0	+1.0	LSB
TUECC	Total unadjusted error	5	-2.0	+2.0	LSB
R_{ASRC}SR	Internal resistance of analog source	t_S in [ns] [2 - 7]	–	(t_S / 150) - 0.25	kΩ
KCC	Coupling Factor between inputs	[6 - 7]	–	1/500	

1. V_{AIN} may exceed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases will be

X000h or X3FFh, respectively.

2. During the t_S 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 the tS sample time. After the end of the tS sample time, changes of the analog input voltage have no effect on the conversion result. Values for the tSC sample clock depend on the programming. Referring to the tC conversion time formula of *Section 21.3.2: Conversion Timing Control* and to *Table 46*:
t_S min = 2 t_{SC} min = 2 t_{CC} min = 2 * 24 TCL = 48 TCL
t_S max = 2 t_{SC} max = 2 * 8 t_{CC} max = 2 * 8 * 96 TCL = 1536 TCL

TCL is defined in *Section 21.4.2: Definition of Internal Timing*, *Section 21.4.4: Prescaler Operation*, and *Section 21.4.5: Direct Drive*:

3. The conversion time formula is: t_C = 14 t_{CC} + t_S + 4 TCL (= 14 t_{CC} + 2 t_{SC} + 4 TCL).
The t_C parameter includes the t_S sample time, the time for determining the digital result and the time to load the result register with the result of the conversion. Values for the t_{CC} conversion clock depend on the programming. Referring to *Table 46*:
t_C min = 14 t_{CC} min. + t_S min + 4 TCL = 14 x 24 x TCL + 48 TCL + 4 TCL = 388 TCL
t_C max = 14 t_{CC} max +t_S max + 4 TCL = 14 x 96 TCL + 1536 TCL + 4 TCL = 2884 TCL

4. This parameter is fixed by ADC control logic.

5. DNL, INL, TUE are tested at V_{AREF} = 5.0 V, VA_{GND} = 0 V, V_{CC} = 4.9 V. It is guaranteed by design characterization for all other voltages within the defined voltage range.'LSB' has a value of V_{AREF} / 1024.
The specified TUE is guaranteed only if an overload condition (see IOV specification) occurs on maximum 2 not selected analog input pins and the absolute sum of input overload currents on all analog input pins does not exceed 10 mA.

6. The coupling factor is measured on a channel while an overload condition occurs on the adjacent not selected channel with an absolute overload current less than 10 mA.

7. Partially tested, guaranteed by design characterization.

8.To remove noise and undesirable high frequency components from the analog input signal, a low-pass filter must be connected at the ADC input. The cut-off frequency of this filter should avoid 2 opposite transitions during the ts sampling time of the ST10 ADC: $f_{cut\text{-}off} \leq 1 / 5\ t_s$ to $1/10\ t_s$, where t_s is the sampling time of the ST10 ADC and is not related to the Nyquist frequency determined by the t_c conversion time.

21.3.2 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 the sample time ts. Next the sampled voltage is converted to a digital value in 10 successive steps, which correspond to the 10-bit resolution of the ADC. The next 4 steps are used for equalizing internal levels (and are kept for exact timing matching with the 10-bit A/D converter module implemented in the ST10F168).

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 sample time t_S (= 2 t_{SC}) and the conversion time t_c (= 14 t_{CC} + 2 t_{SC} + 4 TCL) can be programmed relatively to the ST10F269 CPU clock. This allows adjusting the A/D converter of the ST10F269 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. 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. However the conversion rate in this case may be considerably lower.

The conversion times are programmed via the upper four bit of register ADCON. Bit field ADCTC (conversion time control) selects the basic conversion clock t_{CC}, used for the 14 steps of converting. The sample time tS is a multiple of this conversion time and is selected

by bit field ADSTC (sample time control). The table below lists the possible combinations. The timings refer to the unit TCL, where f_{CPU} = 1/2 TCL.

Table 46. ADC Sampling and Conversion Timing (f_{CPU} = 40 MHz)

ADCTC	Conversion Clock t_{CC}		ADSTC	Sample Clock t_{SC}	
	TCL = 1/2 x f_{XTAL}	At f_{CPU} = 40MHz		t_{SC} =	At f_{CPU} = 40MHz and ADCTC = 00
00	TCL x 24	0.3µs	00	t_{CC}	0.3µs
01	Reserved, do not use	Reserved	01	t_{CC} x 2	0.6µs
10	TCL x 96	1.2 µs	10	t_{CC} x 4	1.2µs
11	TCL x 48	0.6 µs	11	t_{CC} x 8	2.4µs

A complete conversion will take 14 t_{CC} + 2 t_{SC} + 4 TCL (fastest conversion rate = 4.85 ms at 40 MHz). This time includes the conversion itself, the sample time and the time required to transfer the digital value to the result register.

Table 47. ADC Sampling and Conversion Timing (f_{CPU} = 32 MHz)

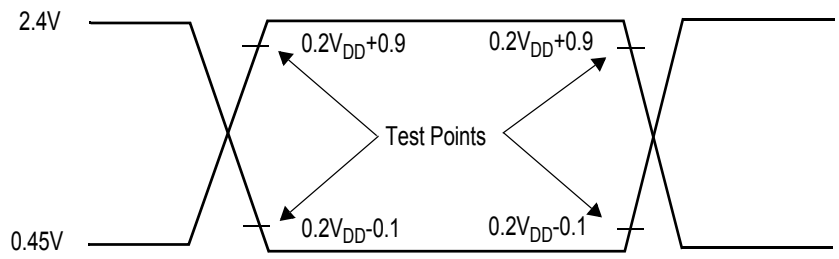
ADCON.15/14 ADCTC	Conversion Clock t_{CC}		ADCON.13/12 ADSTC	Sample Clock t_{SC}	
	TCL = 1/2 x f_{XTAL}	At f_{CPU} = 32MHz		t_{SC} =	At f_{CPU} = 32MHz and ADCTC = 00
00	TCL x 24	0.375µs	00	t_{CC}	0.375µs
01	Reserved, do not use	Reserved	01	t_{CC} x 2	0.75µs
10	TCL x 96	1.5 µs	10	t_{CC} x 4	1.50µs
11	TCL x 48	0.75 µs	11	t_{CC} x 8	3.00µs

A complete conversion will take 14 t_{CC} + 2 t_{SC} + 4 TCL (fastest conversion rate = 6.06 ms at 32 MHz). This time includes the conversion itself, the sample time and the time required to transfer the digital value to the result register.

21.4 AC characteristics

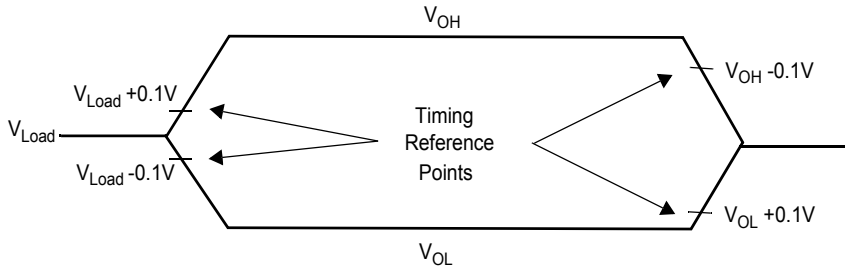
21.4.1 Test Waveforms

Figure 64. Input / Output Waveforms



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

Figure 65. Float Waveforms



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

21.4.2 Definition of Internal Timing

The internal operation of the ST10F269 is controlled by the internal CPU clock fCPU. 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".

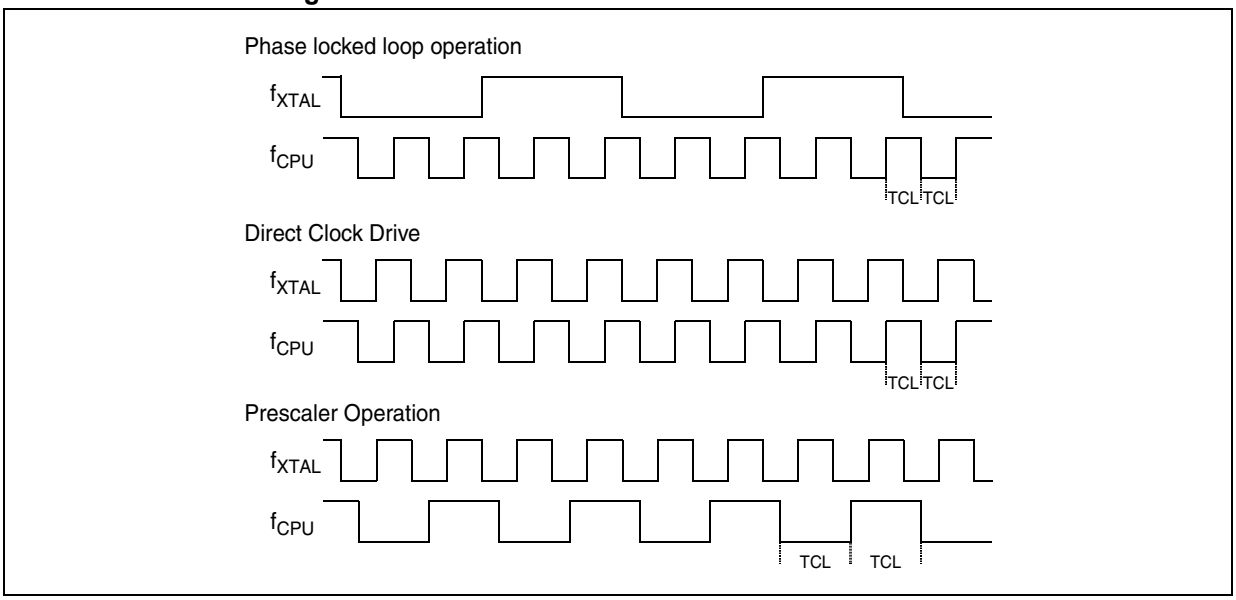
The CPU clock signal can be generated by different mechanisms. The duration of TCL and its variation (and also the derived external timing) depends on the mechanism used to generate f_{CPU}.

This influence must be regarded when calculating the timings for the ST10F269.

The example for PLL operation shown in *Figure 66* 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 66. Generation Mechanisms for the CPU Clock



21.4.3 Clock Generation Modes

Table 48 associates the combinations of these three bits with the respective clock generation mode.

Table 48. CPU Frequency Generation (CPU clock in the range 1 to 40 MHz)

P0H.7	P0H.6	P0H.5	CPU Frequency $f_{CPU} = f_{XTAL} \times F$	External Clock Input Range[1]	Notes
1	1	1	f_{XTAL} x 4	2.5 to 10MHz	Default configuration
1	1	0	f_{XTAL} x 3	3.33 to 13.33MHz	-
1	0	1	f_{XTAL} x 2	5 to 20MHz	-
1	0	0	f_{XTAL} x 5	2 to 8MHz	-
0	1	1	f_{XTAL} x 1	1 to 40MHz	Direct drive[2]
0	1	0	f_{XTAL} x 1.5	6.66 to 26.66MHz	-
0	0	1	f_{XTAL} x 0.5	2 to 80MHz	CPU clock via prescaler[3]
0	0	0	f_{XTAL} x 2.5	4 to 16MHz	-

1. The external clock input range refers to a CPU clock range of 1 to 40 MHz.
2. The maximum input frequency depends on the duty cycle of the external clock signal.
3. The maximum input frequency is 25 MHz when using an external crystal with the internal oscillator; providing that internal serial resistance of the crystal is less than 40Ω. However, higher frequencies can be applied with an external clock source on pin XTAL1, but in this case, the input clock signal must reach the defined levels V_{IL} and V_{IH2}.

Table 49. CPU Frequency Generation (CPU clock in the range 1 to 32 MHz)

P0H.7	P0H.6	P0H.5	CPU Frequency $f_{CPU} = f_{XTAL} \times F$	External Clock Input Range[1]	Notes
1	1	1	f_{XTAL} x 4	2.5 to 8MHz	Default configuration
1	1	0	f_{XTAL} x 3	3.33 to 10.67MHz	-

Table 49. CPU Frequency Generation (CPU clock in the range 1 to 32 MHz) (continued)

P0H.7	P0H.6	P0H.5	CPU Frequency $f_{CPU} = f_{XTAL} \times F$	External Clock Input Range[1]	Notes
1	0	1	f_{XTAL} x 2	5 to 16MHz	-
1	0	0	f_{XTAL} x 5	2 to 6.4MHz	-
0	1	1	f_{XTAL} x 1	1 to 32MHz	Direct drive[2]
0	1	0	f_{XTAL} x 1.5	6.67 to 21.33MHz	-
0	0	1	f_{XTAL} x 0.5	2 to 64MHz	CPU clock via prescaler[3]
0	0	0	f_{XTAL} x 2.5	4 to 12.8MHz	-

1. The external clock input range refers to a CPU clock range of 1 to 32 MHz.
2. The maximum input frequency depends on the duty cycle of the external clock signal.
3. The maximum input frequency is 32 MHz when using an external crystal with the internal oscillator; providing that internal serial resistance of the crystal is less than 40Ω. However, higher frequencies can be applied with an external clock source on pin XTAL1, but in this case, the input clock signal must reach the defined levels V_{IL} and V_{IH2}.

21.4.4 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} (i.e. 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.

21.4.5 Direct Drive

When pins P0.15-13 (P0H.7-5) equal ’011’ during reset the on-chip phase locked loop is disabled and the CPU clock is directly driven from the internal oscillator with the input clock signal.

The frequency of f_{CPU} directly follows the frequency of f_{XTAL} so the high and low time of f_{CPU} (i.e. 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 2TCL 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 (t11 and t45) use the maximum duration of TCL (TCL_{max} = 1 / f_{XTAL} x DC_{max}) instead of TCLmin.
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.

21.4.6 Oscillator Watchdog (OWD)

An on-chip watchdog oscillator is implemented in the ST10F269. This feature is used for safety operation with external crystal oscillator (using direct drive mode with or without prescaler). This watchdog oscillator operates as described below.

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. The PLL free-running frequency is between 2 and 10 MHz. 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 (XP3INT) 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 can switch the CPU clock source back to direct clock input.

When the OWD is disabled, the CPU clock is always external oscillator clock and the PLL is switched off to decrease consumption supply current.

21.4.7 Phase Locked Loop

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 48* and *Table 49*). 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. It decreases according to the formula and to *Figure 67* given below. For N periods of TCL the minimum value is computed using the corresponding deviation DN:

$$TCL_{MIN} = TCL_{NOM} \times \left(1 - \frac{\left[D_N\right]}{100}\right)$$

$$D_N = \pm(4 - N/15)[\%]$$

where N = number of consecutive TCL periods and 1 ≤ N ≤ 40. So for a period of 3 TCL periods (N = 3):

D3 =4 - 3/15 = 3.8%

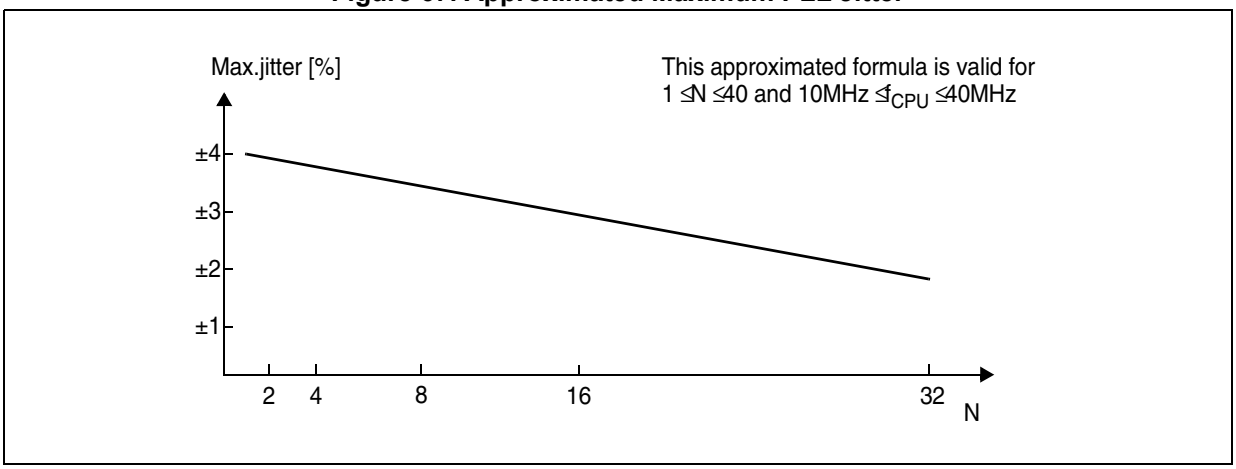
3TCLmin =3TCLNOM x (1 - 3.8/100)
=3TCLNOM x 0.962

3TCLmin=36.075ns (at fCPU = 40MHz)

3TCLmin=45.1ns (at fCPU = 32MHz)

This is especially important for bus cycles using wait states and e.g. for the operation of timers, serial interfaces, etc. For all slower operations and longer periods (e.g. pulse train generation or measurement, lower baud rates, etc.) the deviation caused by the PLL jitter is negligible.

Figure 67. Approximated Maximum PLL Jitter



21.4.8 External Clock Drive XTAL1

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +119 °C

Table 50. External Clock Drive XTAL1 (max f_{CPU} = 40 MHz)

Parameter	Symbol	$f_{CPU} = f_{XTAL}$		$f_{CPU} = f_{XTAL} / 2$		$f_{CPU} = f_{XTAL} \times F$ F = 1.5/2,/2.5/3/4/5		Unit
		Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	
Oscillator period	$t_{OSC}SR$	25 [1]	–	12.5	–	40 x N	100 x N	ns
High time	t_1SR	10 [2]	–	5 [2]	–	10 [2]	–	ns
Low time	t_2SR	10 [2]	–	5 [2]	–	10 [2]	–	ns
Rise time	t_3SR	–	3 [2]	–	3 [2]	–	3 [2]	ns
Fall time	t_4SR	–	3 [2]	–	3 [2]	–	3 [2]	ns

1. Theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal. 25 MHz is the maximum input frequency when using an external crystal oscillator. However, 40 MHz can be applied with an external clock source.

2. The input clock signal must reach the defined levels V_{IL} and V_{IH2}.

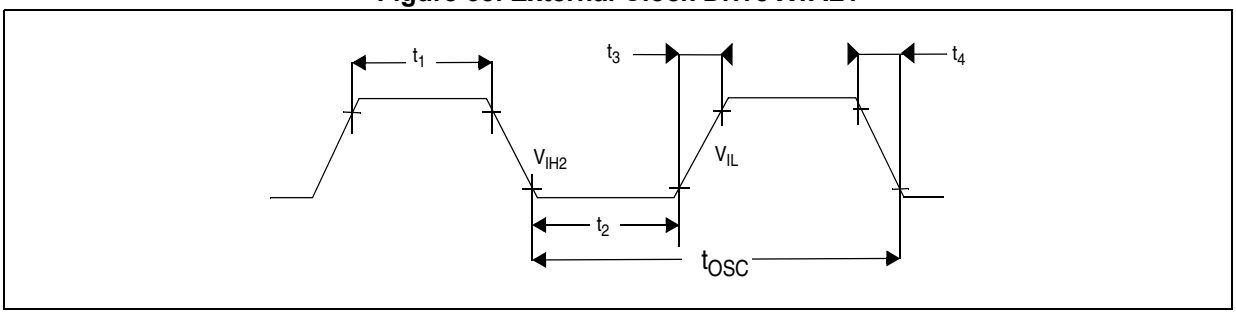
V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +125 °C

Table 51. External Clock Drive XTAL1 (max f_{CPU} = 32 MHz)

Parameter	Symbol	$f_{CPU} = f_{XTAL}$		$f_{CPU} = f_{XTAL} / 2$		$f_{CPU} = f_{XTAL} \times F$ F = 1.5/2,/2.5/3/4/5		Unit
		Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	
Oscillator period	$t_{OSC}SR$	31.25[1]	–	15.625	–	31.25 x N	–	ns
High time	t_1SR	12.5[2]	–	6.25[2]	–	12.5[2]	–	ns
Low time	t_2SR	12.5[2]	–	6.25[2]	–	12.5[2]	–	ns
Rise time	t_3SR	–	3.125[2]	–	1.56[2]	–	3.125[2]	ns
Fall time	t_4SR	–	3.125[2]	–	1.56[2]	–	3.125[2]	ns

1. Theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal. 25 MHz is the maximum input frequency when using an external crystal oscillator. However, 32 MHz can be applied with an external clock source.
2. The input clock signal must reach the defined levels V_{IL} and V_{IH2}.

Figure 68. External Clock Drive XTAL1



21.4.9 Memory Cycle Variables

The table below uses 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 computed.

Table 52. 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])

21.4.10 Multiplexed Bus

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +119°C, C_L = 50 pF,
ALE cycle time = 6 TCL + 2 t_A + t_C + t_F (75 ns at 40 MHz CPU clock without wait states).

Table 53. Multiplexed Bus Characteristics (max f_{CPU} = 40 MHz)

Symbol	Parameter	Max. CPU Clock = 40 MHz		Variable CPU Clock 1/2 TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_5CC	ALE high time	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_6CC	Address setup to ALE	2 + t_A	–	TCL - 10.5 + t_A	–	ns
t_7CC	Address hold after ALE[1]	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_8CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_9CC	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	–	2 TCL -9.5 + t_C	–	ns
t_{13}CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3 TCL -9.5 + t_C	–	ns
t_{14}SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2 TCL - 19 + t_C	ns
t_{15}SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3 TCL - 19 + t_C	ns
t_{16}SR	ALE low to valid data in	–	18.5 + t_A + t_C	–	3 TCL - 19 + t_A + t_C	ns
t_{17}SR	Address/Unlatched $\overline{CS}$ to valid data in	–	22 + 2t_A + t_C	–	4 TCL - 28 + 2t_A + t_C	ns
t_{18}SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{19}SR	Data float after $\overline{RD}$[1]	–	16.5 + t_F	–	2 TCL - 8.5 + t_F	ns
t_{22}CC	Data valid to $\overline{WR}$	10 + t_C	–	2 TCL -15 + t_C	–	ns
t_{23}CC	Data hold after $\overline{WR}$	4 + t_F	–	2 TCL - 8.5 + t_F	–	ns
t_{25}CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	15 + t_F	–	2 TCL -10 + t_F	–	ns
t_{27}CC	Address/Unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	10 + t_F	–	2 TCL -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	–	18.5 + t_C + 2t_A	–	3 TCL - 19 + t_C + 2t_A	ns
t_{40}CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	27 + t_F	–	3 TCL - 10.5 + t_F	–	ns
t_{42}CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	7 + t_A	–	TCL - 5.5+ t_A	–	ns

Table 53. Multiplexed Bus Characteristics (max f_{CPU} = 40 MHz) (continued)

Symbol	Parameter	Max. CPU Clock = 40 MHz		Variable CPU Clock 1/2 TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
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	–	0	–	0	ns
t_{45}CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)1	–	12.5	–	TCL	ns
t_{46}SR	$\overline{RdCS}$ to Valid Data In (with RW delay)	–	4 + t_C	–	2 TCL - 21 + t_C	ns
t_{47}SR	$\overline{RdCS}$ to Valid Data In (no RW delay)	–	16.5 + t_C	–	3 TCL - 21 + t_C	ns
t_{48}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW delay)	15.5 + t_C	–	2 TCL - 9.5 + t_C	–	ns
t_{49}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW delay)	28 + t_C	–	3 TCL - 9.5 + t_C	–	ns
t_{50}CC	Data valid to $\overline{WrCS}$	10 + t_C	–	2 TCL - 15+ t_C	–	ns
t_{51}SR	Data hold after $\overline{RdCS}$	0	–	0	–	ns
t_{52}SR	Data float after $\overline{RdCS}$1	–	16.5 + t_F	–	2 TCL - 8.5+t_F	ns
t_{54}CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	6 + t_F	–	2 TCL - 19 + t_F	–	ns
t_{56}CC	Data hold after $\overline{WrCS}$	6 + t_F	–	2 TCL - 19 + t_F	–	ns

1. Partially tested, guaranteed by design characterization.

V_{DD} = 5 V ± 10%, V_{SS} = 0 V, T_A = -40 to +125°C, C_L = 50 pF,
ALE cycle time = 6 TCL + 2 t_A + t_C + t_F (187.5 ns at 32 MHz CPU clock without wait states).

Table 54. Multiplexed Bus Characteristics (max f_{CPU} = 32 MHz)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{42}CC	ALE high time	5.625 + t_A	–	TCL - 10 + t_A	–	ns
t_6CC	Address setup to ALE	0.625 + t_A	–	TCL - 15+ t_A	–	ns
t_7CC	Address hold after ALE1	5.625 + t_A	–	TCL - 10 + t_A	–	ns
t_8CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	5.625 + t_A	–	TCL - 10 + t_A	–	ns
t_9CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	-10 + t_A	–	-10 + 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	–	21.625	–	TCL + 6	ns

Table 54. Multiplexed Bus Characteristics (max f_{CPU} = 32 MHz) (continued)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{12}CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	21.25 + t_C	–	2TCL - 10 + t_C	–	ns
t_{13}CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	36.875 + t_C	–	3TCL - 10 + t_C	–	ns
t_{14}SR	$\overline{RD}$ to valid data in (with RW-delay)	–	11.25 + t_C	–	2TCL - 20+ t_C	ns
t_{15}SR	$\overline{RD}$ to valid data in (no RW-delay)	–	26.875 + t_C	–	3TCL - 20+ t_C	ns
t_{16}SR	ALE low to valid data in	–	26.875 + t_A + t_C	–	3TCL - 20 + t_A + t_C	ns
t_{17}SR	Address/Unlatched $\overline{CS}$ to valid data in	–	32.5 + 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 $\overline{RD}$[1]	–	17.25 + t_F	–	2TCL - 14 + t_F	ns
t_{22}CC	Data valid to $\overline{WR}$	11.25 + t_C	–	2TCL - 20 + t_C	–	ns
t_{23}CC	Data hold after $\overline{WR}$	17.25 + t_F	–	2TCL - 14 + t_F	–	ns
t_{25}CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	17.25 + t_F	–	2TCL - 14 + t_F	–	ns
t_{27}CC	Address/Unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	17.25 + t_F	–	2TCL - 14 + 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	–	26.875 + t_C + 2t_A	–	3TCL - 20 + t_C + 2t_A	ns
t_{40}CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	32.875 + t_F	–	3TCL - 14 + t_F	–	ns
t_{42}CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	11.625 + t_A	–	TCL - **4** + t_A	–	ns
t_{43}CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)	-4 + t_A	–	-4 + t_A	–	ns
t_{44}CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)[1]	–	0	–	0	ns
t_{45}CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)[1]	–	15.625	–	TCL	ns
t_{46}SR	$\overline{RdCS}$ to Valid Data In (with RW delay)	–	7.25 + t_C	–	2TCL - 24 + t_C	ns
t_{47}SR	$\overline{RdCS}$ to Valid Data In (no RW delay)	–	22.875 + t_C	–	3TCL - 24 + t_C	ns
t_{48}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW delay)	21.25 + t_C	–	2TCL - 10 + t_C	–	ns
t_{49}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW delay)	36.875 + t_C	–	3TCL - 10 + t_C	–	ns
t_{50}CC	Data valid to $\overline{WrCS}$	17.25 + t_C	–	2TCL - 14+ t_C	–	ns

Table 54. Multiplexed Bus Characteristics (max f_{CPU} = 32 MHz) (continued)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{51}SR	Data hold after $\overline{RdCS}$	0	–	0	–	ns
t_{52}SR	Data float after $\overline{RdCS}$[1]	–	11.25 + t_F	–	2TCL - 20 + t_F	ns
t_{54}CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	11.25 + t_F	–	2TCL - 20 + t_F	–	ns
t_{56} CC	Data hold after $\overline{WrCS}$	11.25 + t_F	–	2TCL - 20 + t_F	–	ns

1. Partially tested, guaranteed by design characterization.

Figure 69. External Memory Cycle: Multiplexed Bus, With/Without Read/Write Delay, Normal ALE

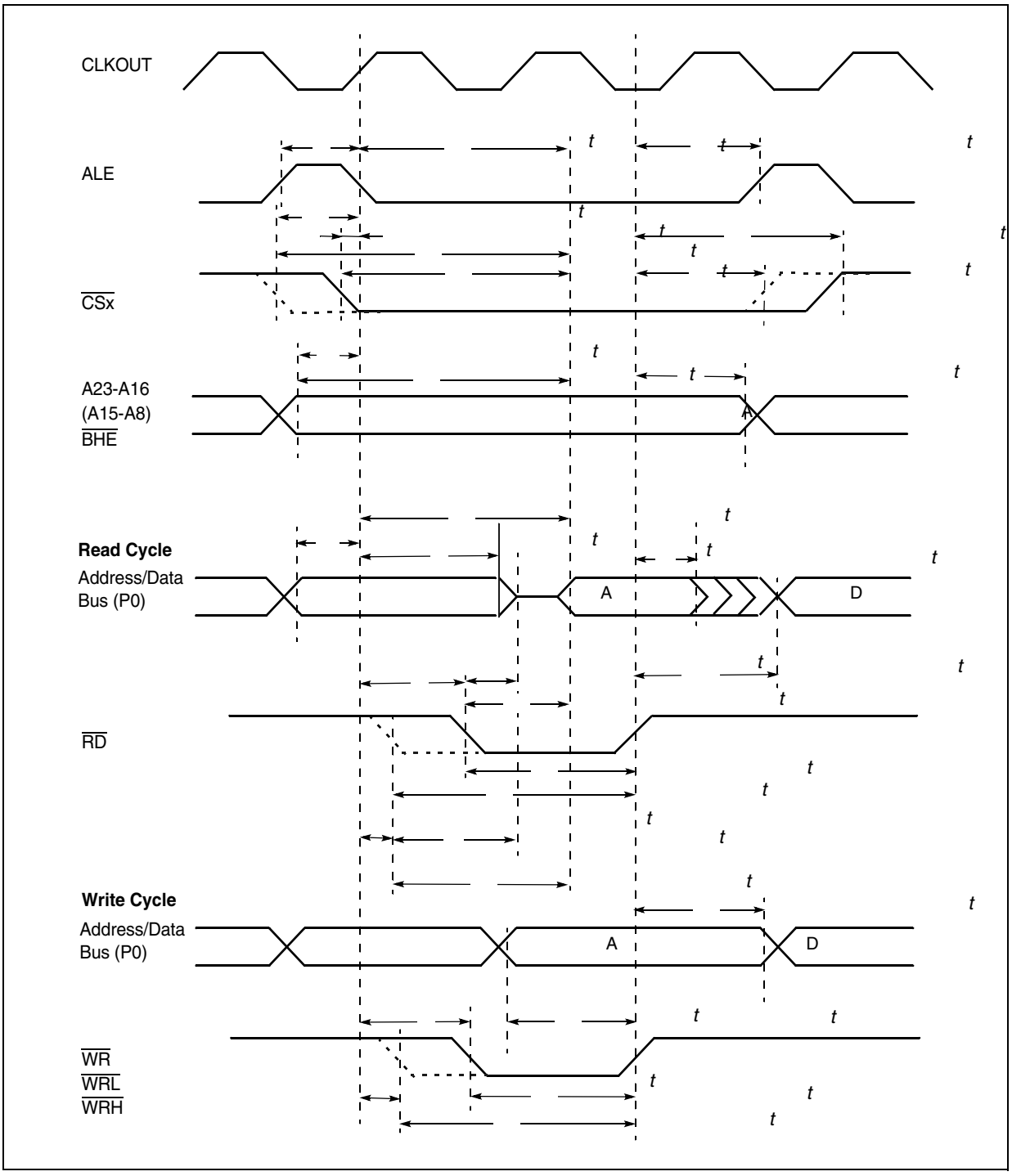


Figure 70. External Memory Cycle: Multiplexed Bus, With/Without Read/Write Delay, Extended ALE

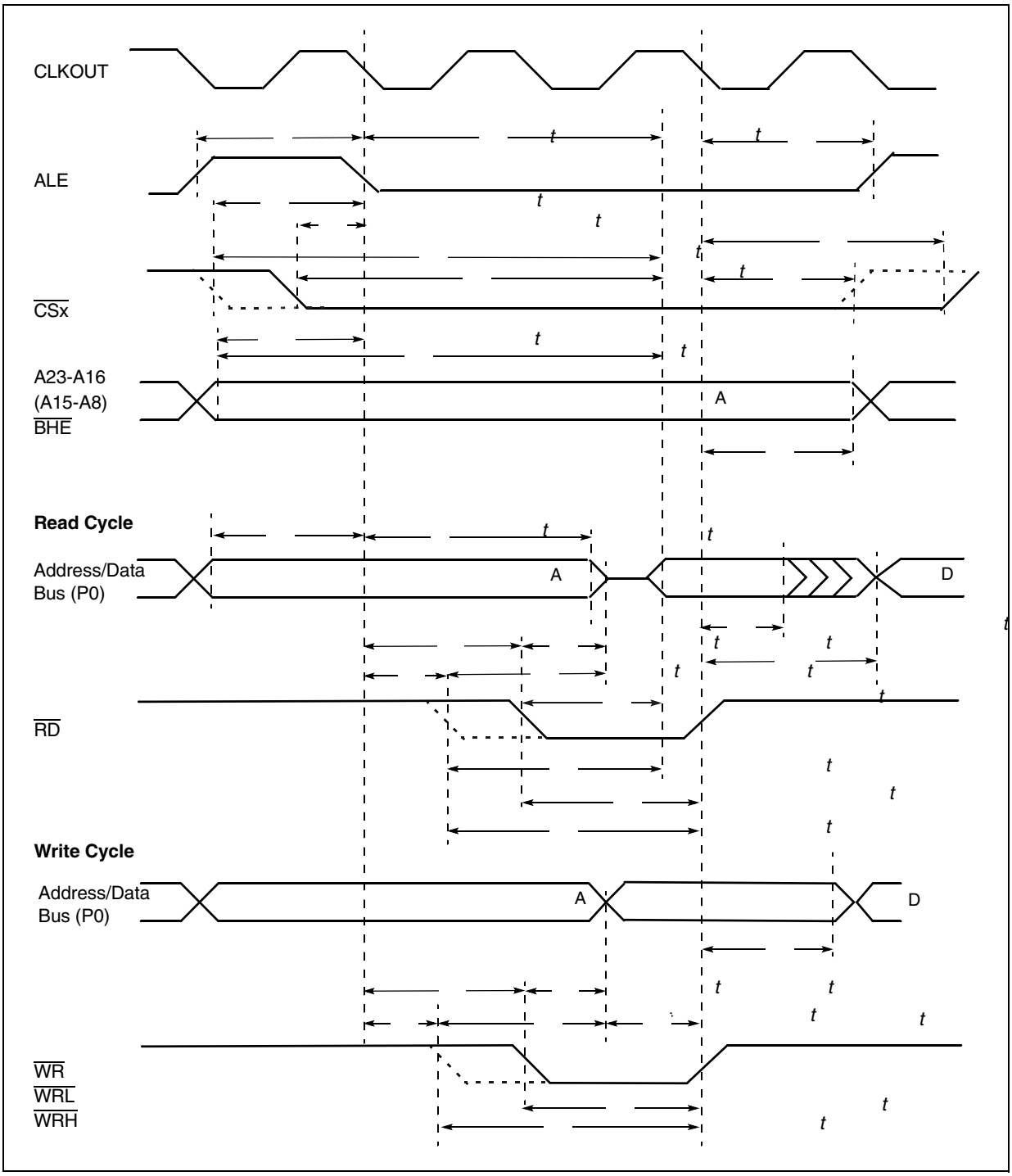


Figure 71. External Memory Cycle: Multiplexed Bus, With/Without Read/Write Delay, Normal ALE, Read / Write Chip Select

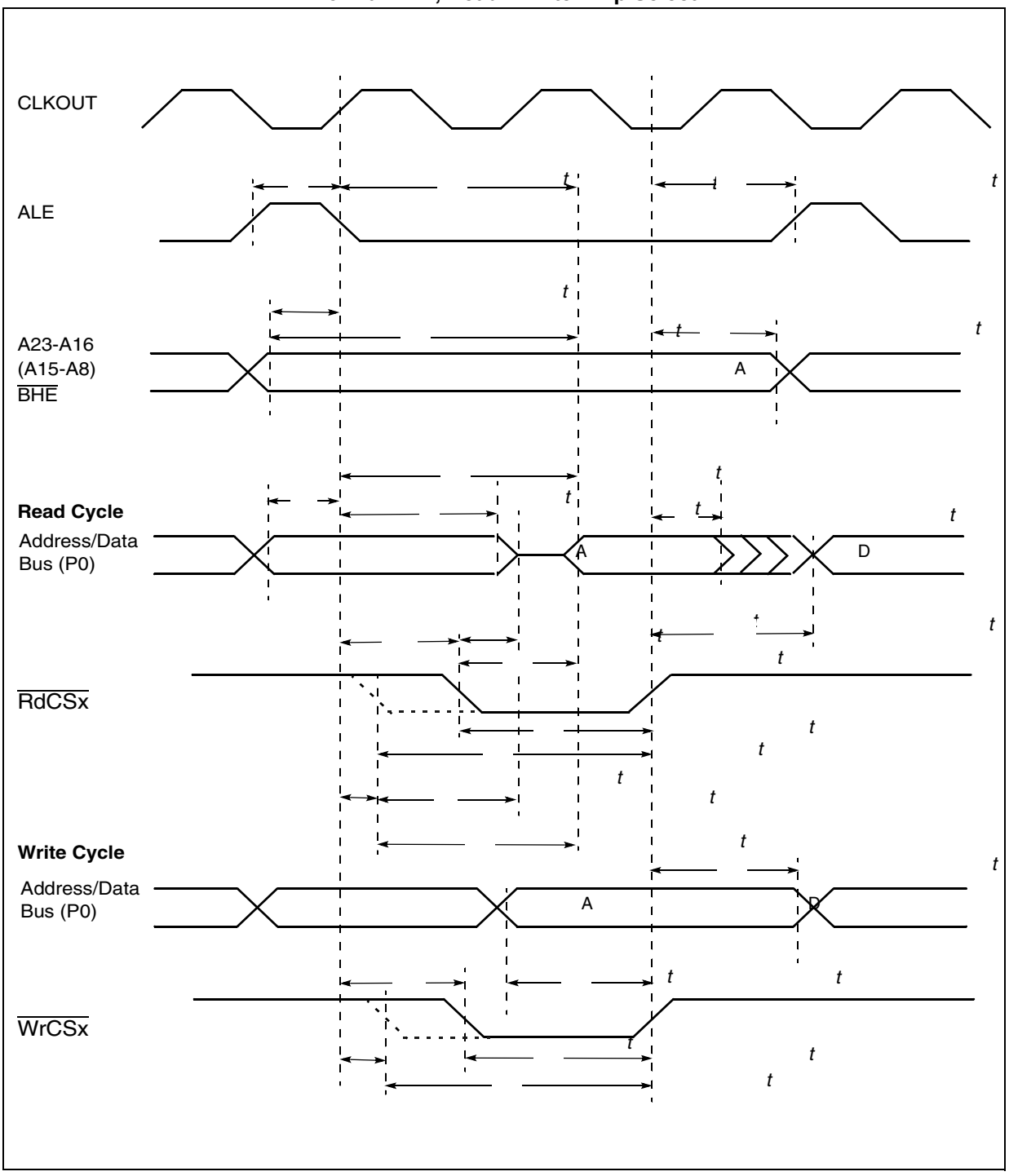
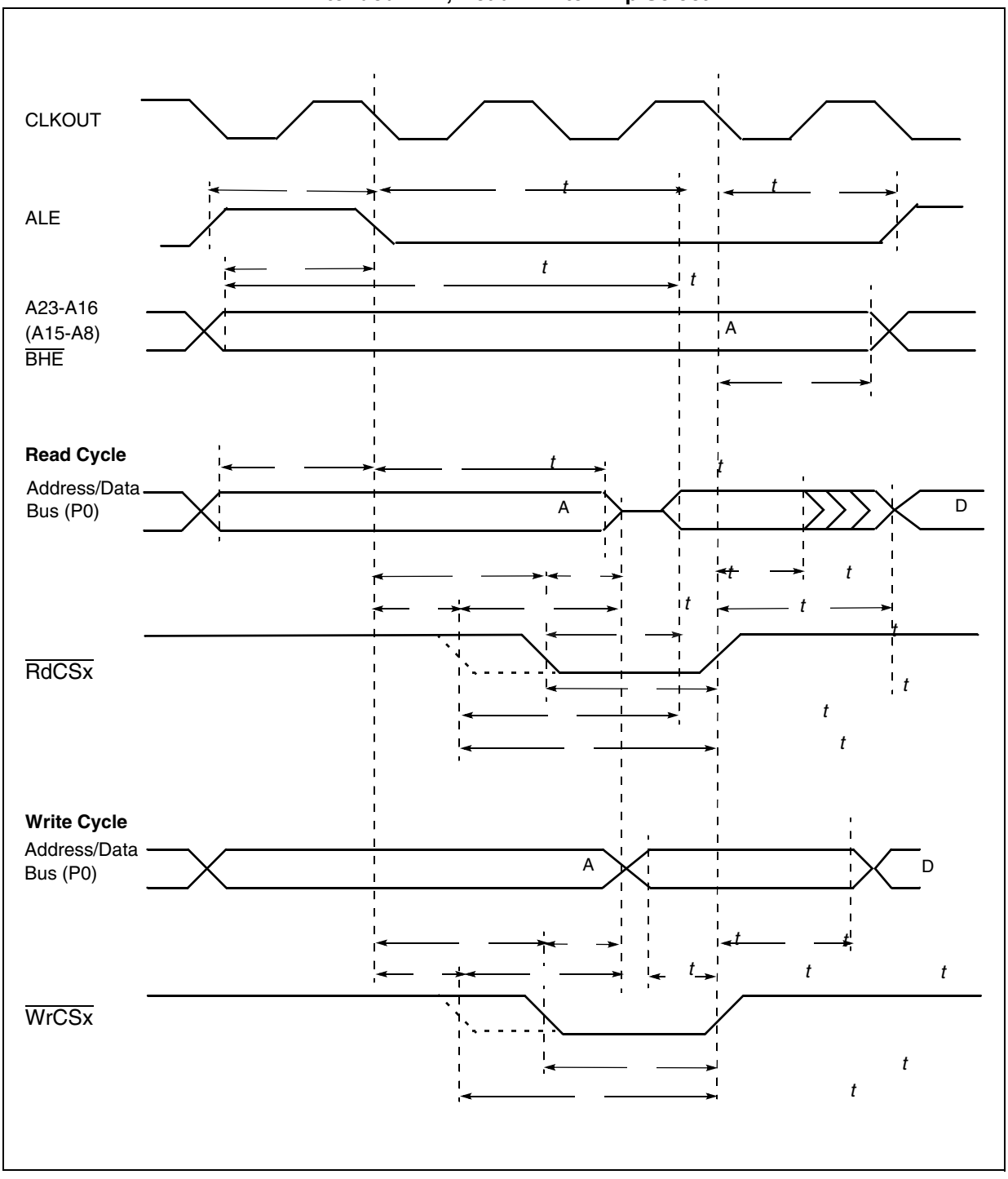


Figure 72. External Memory Cycle: Multiplexed Bus, With/Without Read/Write Delay, Extended ALE, Read / Write Chip Select



21.4.11 Demultiplexed Bus

V_{DD} = 5 V ± 10%, V_{SS} = 0V, T_A = -40 to +119°C, CL = 50pF,
ALE cycle time = 4 TCL + 2 t_A + t_C + t_F (50 ns at 40 MHz CPU clock without wait states)

Table 55. Demultiplexed Bus Characteristics (max f_{CPU} = 40 MHz)

Symbol	Parameter	Maximum CPU Clock = 40 MHz		Variable CPU Clock 1/2 TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_5CC	ALE high time	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_6CC	Address setup to ALE	2 + t_A	–	TCL - 10.5 + t_A	–	ns
t_{80}CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	16.5 + 2t_A	–	2 TCL - 8.5 + 2t_A	–	ns
t_{81}CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	4 + 2t_A	–	TCL - 8.5 + 2t_A	–	ns
t_{12}CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2 TCL - 9.5 + t_C	–	ns
t_{13}CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3 TCL - 9.5 + t_C	–	ns
t_{14}SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2 TCL - 19 + t_C	ns
t_{15}SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3 TCL - 19 + t_C	ns
t_{16}SR	ALE low to valid data in	–	18.5 + t_A + t_C	–	3 TCL - 19 + t_A + t_C	ns
t_{17}SR	Address/Unlatched $\overline{CS}$ to valid data in	–	22 + 2t_A + t_C	–	4 TCL - 28 + 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 3]	–	16.5 + t_F	–	2 TCL - 8.5 + t_F + 2t_A [1]	ns
t_{21}SR	Data float after $\overline{RD}$ rising edge (no RW-delay) [1 3]	–	4 + t_F	–	TCL - 8.5 + t_F + 2t_A [1]	ns
t_{22}CC	Data valid to $\overline{WR}$	10 + t_C	–	2 TCL - 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 (no t_F) -5 + t_F (t_F > 0)	–	0 (no t_F) -5 + t_F (t_F > 0)	–	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	–	18.5 + t_C + 2t_A	–	3 TCL - 19 + t_C + 2t_A	ns
t_{41}CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	2 + t_F	–	TCL - 10.5 + t_F	–	ns
t_{82}CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay)	14.5 + 2t_A	–	2 TCL - 10.5 + 2t_A	–	ns

Table 55. Demultiplexed Bus Characteristics (max f_{CPU} = 40 MHz) (continued)

Symbol	Parameter	Maximum CPU Clock = 40 MHz		Variable CPU Clock 1/2 TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{83}CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay)	2 + 2t_A	–	TCL - 10.5 + 2t_A	–	ns
t_{46}SR	$\overline{RdCS}$ to Valid Data In (with RW-delay)	–	4 + t_C	–	2 TCL - 21 + t_C	ns
t_{47}SR	$\overline{RdCS}$ to Valid Data In (no RW-delay)	–	16.5 + t_C	–	3 TCL - 21 + t_C	ns
t_{48}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay)	15.5 + t_C	–	2 TCL - 9.5 + t_C	–	ns
t_{49}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay)	28 + t_C	–	3 TCL - 9.5 + t_C	–	ns
t_{50}CC	Data valid to $\overline{WrCS}$	10 + t_C	–	2 TCL - 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	–	2 TCL - 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 are latched with the same clock edge that triggers the address change and the rising $\overline{RD}$ edge. Therefore address changes before the end of RD have no impact on read cycles.
3. Partially tested, guaranteed by design characterization.

V_{DD} = 5 V ± 10%, V_{SS} = 0V, T_A = -40 to +85°C, CL = 50pF,
ALE cycle time = 4 TCL + 2 t_A + t_C + t_F (125ns at 32MHz CPU clock without wait states)
RW-delay and t_A refer to the next following bus cycle.

Table 56. Demultiplexed Bus Characteristics (max f_{CPU} = 32 MHz)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_5CC	ALE high time	5.625 + t_A	–	TCL - 10+ t_A	–	ns
t_6CC	Address setup to ALE	0.625 + t_A	–	TCL - 15+ t_A	–	ns
t_{80}CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	21.25 + 2t_A	–	2TCL - 10 + 2t_A	–	ns
t_{81}CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	5.625 + 2t_A	–	TCL -10 + 2t_A	–	ns
t_{12}CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	21.25 + t_C	–	2TCL - 10 + t_C	–	ns

Table 56. Demultiplexed Bus Characteristics (max f_{CPU} = 32 MHz) (continued)

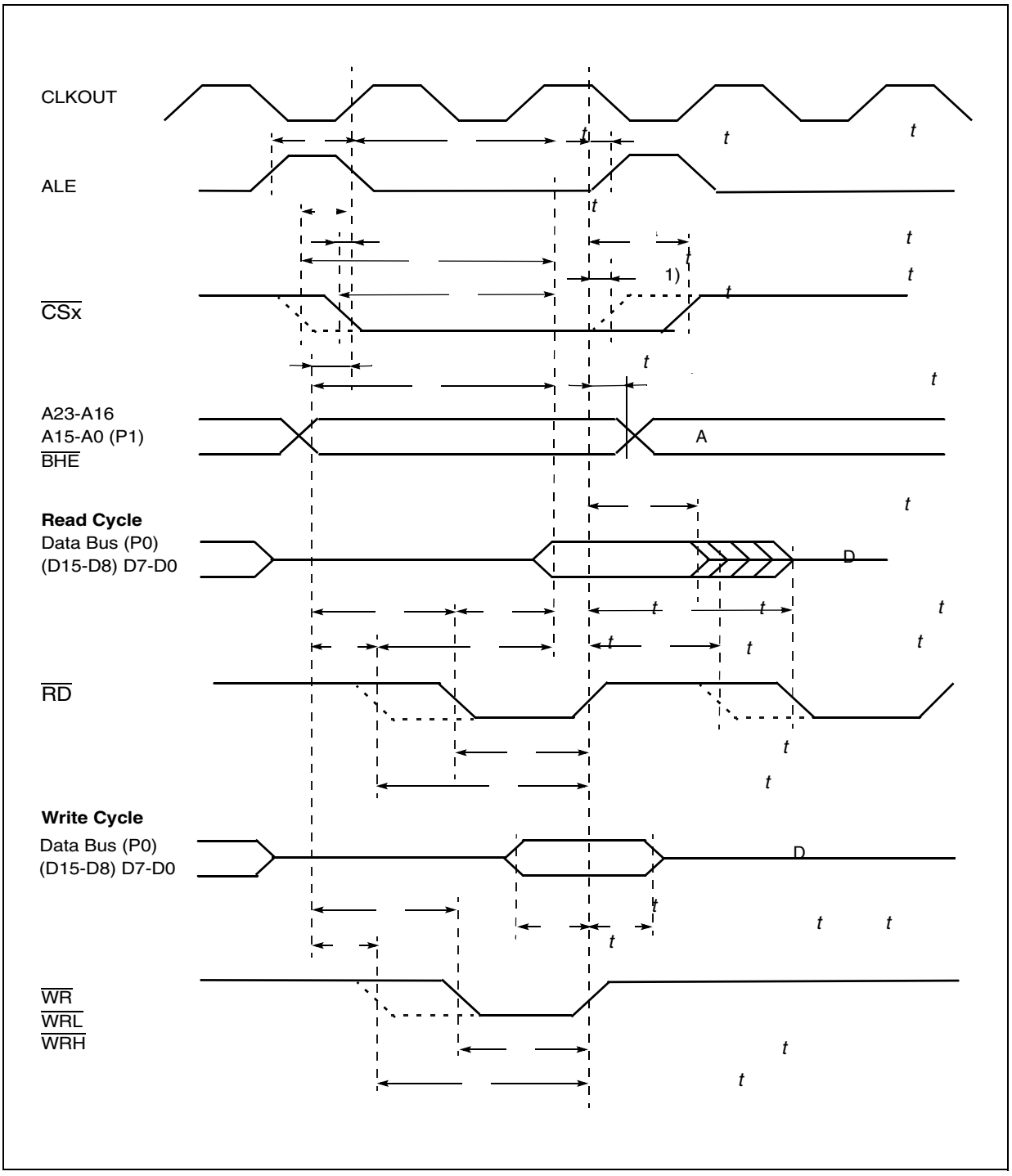
Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{13}CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	36.875 + t_C	–	3TCL - 10 + t_C	–	ns
t_{14}SR	$\overline{RD}$ to valid data in (with RW-delay)	–	11.25 + t_C	–	2TCL - 20 + t_C	ns
t_{15}SR	$\overline{RD}$ to valid data in (no RW-delay)	–	26.875 + t_C	–	3TCL - 20 + t_C	ns
t_{16}SR	ALE low to valid data in	–	26.875 + t_A + t_C	–	3TCL - 20 + t_A + t_C	ns
t_{17}SR	Address/Unlatched $\overline{CS}$ to valid data in	–	32.5 + 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 - 3]	–	26 + t_F	–	2TCL - 14 + t_F + 2t_A[1]	ns
t_{21}SR	Data float after $\overline{RD}$ rising edge (no RW-delay) [1 - 3]	–	5.625 + t_F	–	TCL - 10 + t_F + 2t_A[1]	ns
t_{22}CC	Data valid to $\overline{WR}$	11.25 + t_C	–	2TCL- 20 + t_C	–	ns
t_{24}CC	Data hold after $\overline{WR}$	5.625 + t_F	–	TCL - 10+ 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 (no t_F) -5 + t_F (t_F > 0)	–	0 (no t_F) -5 + t_F (t_F > 0)	–	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	10 - t_A	-4 - t_A	10 - t_A	ns
t_{39}SR	Latched $\overline{CS}$ low to Valid Data In	–	26.875 + t_C+ 2t_A	–	3TCL - 20 + t_C + 2t_A	ns
t_{41}CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	1.625 + t_F	–	TCL - 14 + t_F	–	ns
t_{82}CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay)	17.25 + 2t_A	–	2TCL - 14 + 2t_A	–	ns
t_{83}CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay)	1.625 + 2t_A	–	TCL -14 + 2t_A	–	ns
t_{46}SR	$\overline{RdCS}$ to Valid Data In (with RW-delay)	–	7.25 + t_C	–	2TCL - 24 + t_C	ns
t_{47}SR	$\overline{RdCS}$ to Valid Data In (no RW-delay)	–	22.875 + t_C	–	3TCL - 24 + t_C	ns
t_{48}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay)	21.25 + t_C	–	2TCL - 10 + t_C	–	ns
t_{49}CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay)	36.875 + t_C	–	3TCL - 10 + t_C	–	ns

Table 56. Demultiplexed Bus Characteristics (max f_{CPU} = 32 MHz) (continued)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2 TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{50}CC	Data valid to $\overline{WrCS}$	17.25 + t_C	–	2TCL - 14 + 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]	–	21.25 + t_F	–	2TCL - 10 + t_F	ns
t_{68}SR	Data float after $\overline{RdCS}$ (no RW-delay) [3]	–	0 + t_F	–	TCL - 10 + t_F	ns
t_{55}CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	-10 + t_F	–	-10 + t_F	–	ns
t_{57}CC	Data hold after $\overline{WrCS}$	1.625 + t_F	–	TCL - 14 + t_F	–	ns

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

Figure 73. External Memory Cycle: Demultiplexed Bus, With/Without Read/Write Delay, Normal ALE



1. 1. Un-latched CSx = t_{41u} = t_{41} TCL =10.5 + t_F.

Figure 74. External Memory Cycle: Demultiplexed Bus, With/Without Read/Write Delay, Extended ALE

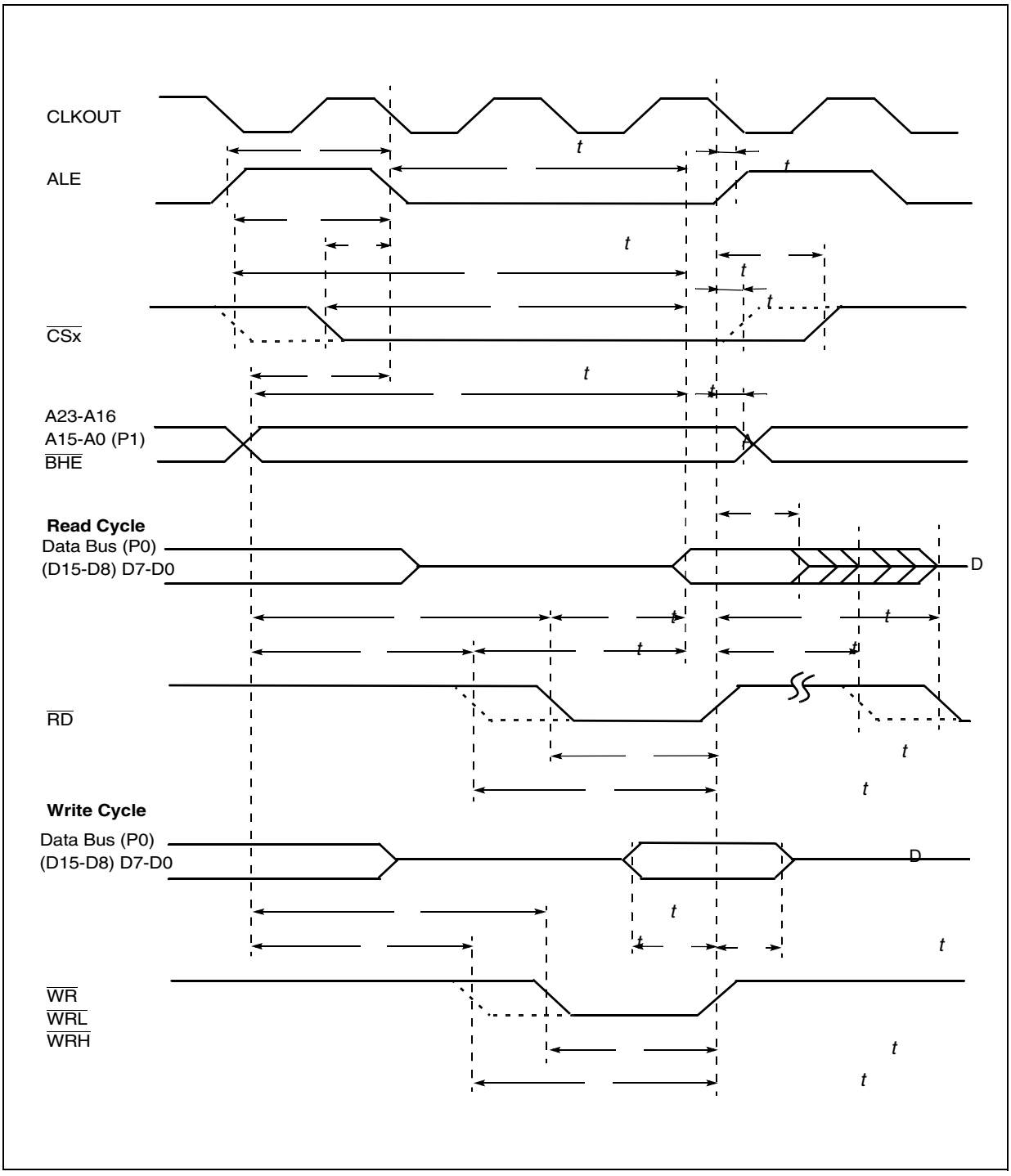


Figure 75. External Memory Cycle: Demultiplexed Bus, With/Without Read/Write Delay, Normal ALE, Read/Write Chip Select

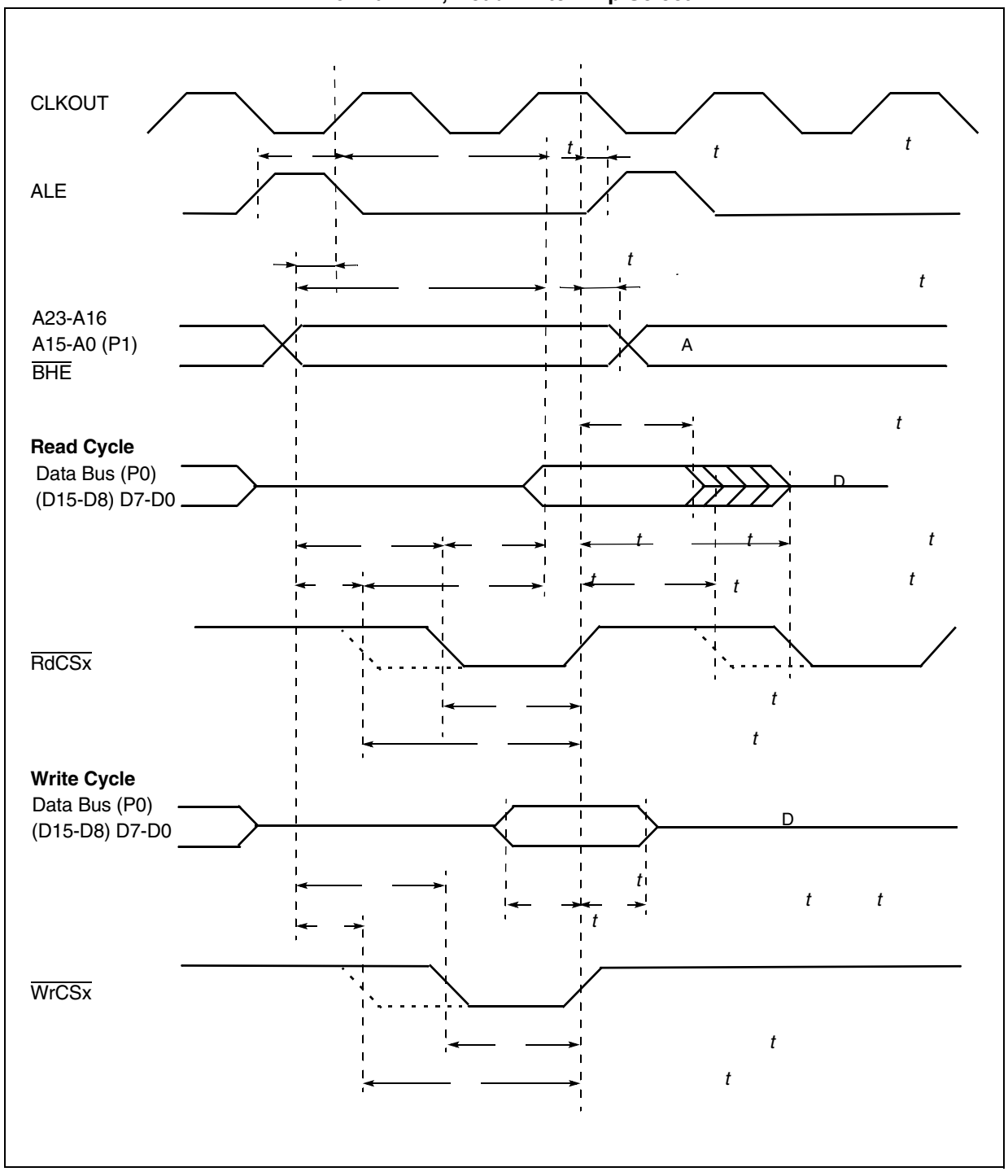
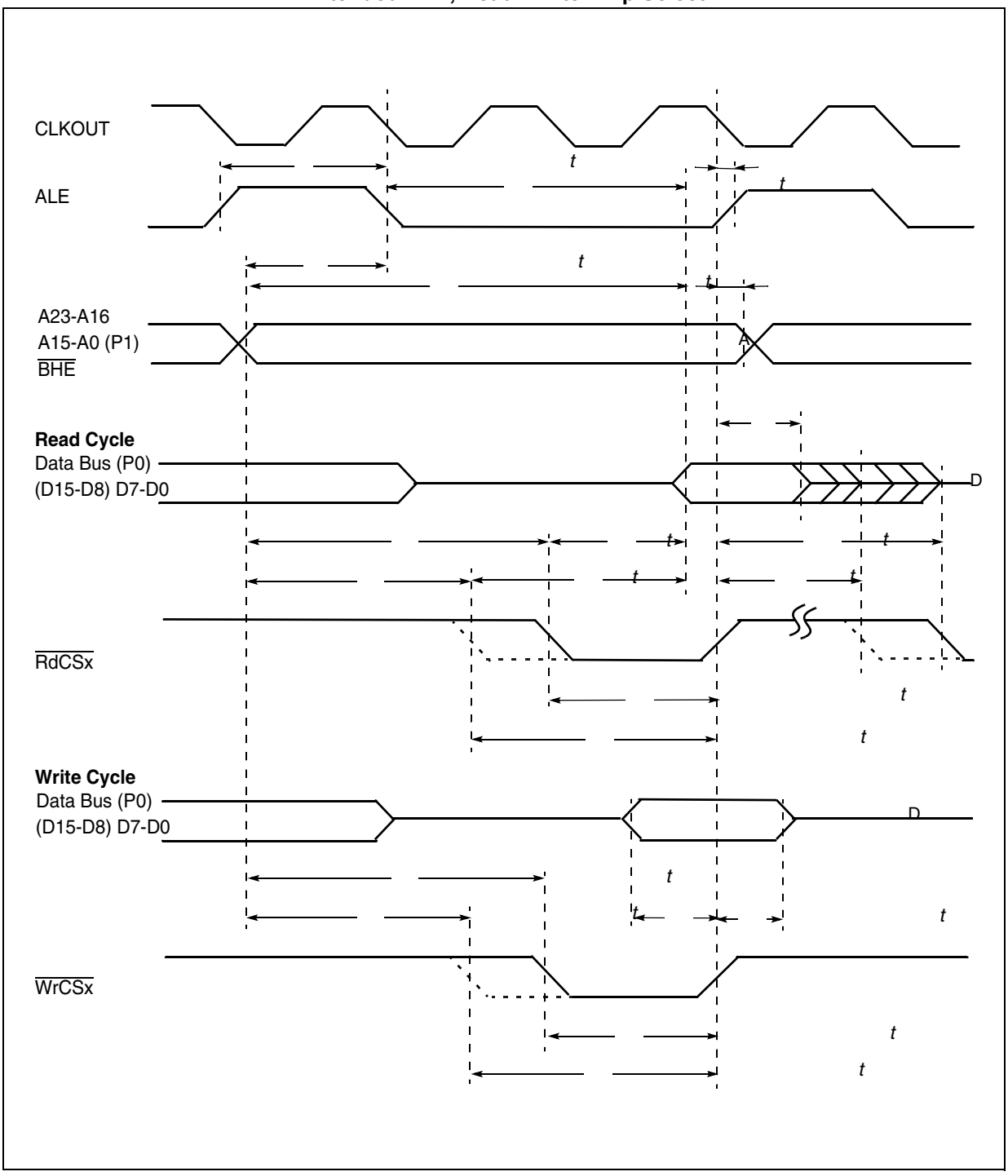


Figure 76. External Memory Cycle: Demultiplexed Bus, no Read/Write Delay, Extended ALE, Read /Write Chip Select



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

V_{DD} = 5V ± 10%, V_{SS} = 0V, T_A = -40 to + 119°C, C_L = 50 pF

Table 57. CLKOUT and $\overline{\text{READY}}$ Characteristics (max f_{CPU} = 40 MHz)

Symbol	Parameter	Maximum CPU Clock = 40 MHz		Variable CPU Clock 1/2TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{29}CC	CLKOUT cycle time	25	25	2TCL	2TCL	ns
t_{30}CC	CLKOUT high time	4	–	TCL – 8.5	–	ns
t_{31}CC	CLKOUT low time	3	–	TCL – 9.5	–	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	12.5	–	12.5	–	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)	12.5	–	12.5	–	ns
t_{59}SR	Asynchronous $\overline{\text{READY}}$ hold time 1)	2	–	2	–	ns
t_{60}SR	Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) 2)	0	0 + $2t_A$ + t_C + t_F 2)	0	TCL - 12.5 + $2t_A$ + t_C + t_F 2)	ns

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

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

Table 58. CLKOUT and $\overline{\text{READY}}$ Characteristics (max f_{CPU} = 32 MHz)

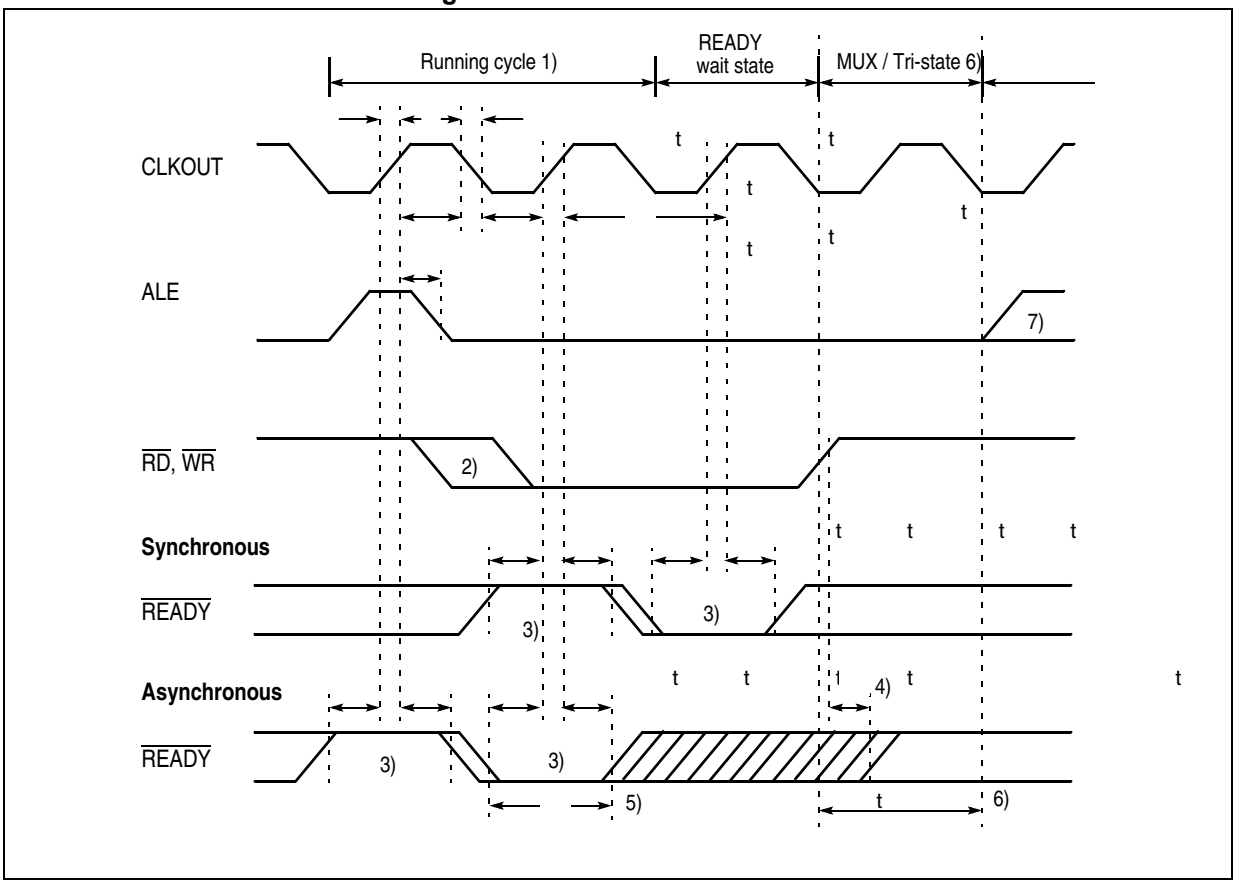
Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{29}CC	CLKOUT cycle time	31.25	31.25	2TCL	2TCL	ns
t_{30}CC	CLKOUT high time	9.625	–	TCL – 6	–	ns
t_{31}CC	CLKOUT low time	5.625	–	TCL – 10	–	ns
t_{32}CC	CLKOUT rise time	–	4	–	4	ns
t_{33}CC	CLKOUT fall time	–	4	–	4	ns
t_{34}CC	CLKOUT rising edge to ALE falling edge	-3 + t_A	+7 + t_A	-3 + t_A	+7 + t_A	ns

Table 58. CLKOUT and $\overline{\text{READY}}$ Characteristics (max f_{CPU} = 32 MHz) (continued)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{35}SR	Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	14	–	14	–	ns
t_{36}SR	Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	4	–	4	–	ns
t_{37}SR	Asynchronous $\overline{\text{READY}}$ low time	45.25	–	2TCL + 14	–	ns
t_{58}SR	Asynchronous $\overline{\text{READY}}$ setup time [1]	14	–	14	–	ns
t_{59}SR	Asynchronous $\overline{\text{READY}}$ hold time [1]	4	–	4	–	ns
t_{60}SR	Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) [2]	0	$0 + 2t_A + t_C + t_F$ [2]	0	TCL - 15.625 + $2t_A + t_C + t_F$ [2]	ns

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

Figure 77. 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 $\overline{\text{READY}}$ 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 (e.g. 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 2 CLKOUT cycles, for a demultiplexed bus without MTTC wait state this delay is zero.
7. The next external bus cycle may start here.

21.4.13 External Bus Arbitration

V_{DD} = 5 V ± 10%, V_{SS} = 0V, T_A = -40 to +119°C, C_L = 50pF

Table 59. external Bus Arbitration (max f_{CPU} = 40 MHz)

Symbol	Parameter	Maximum CPU Clock = 40 MHz		Variable CPU Clock 1/2TCL = 1 to 40 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{61SR}	$\overline{HOLD}$ input setup time to CLKOUT	15	–	15	–	ns
t_{62CC}	CLKOUT to $\overline{HLDA}$ high or $\overline{BREQ}$ low delay	–	12.5	–	12.5	ns
t_{63CC}	CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	–	12.5	–	12.5	ns
t_{64CC}	$\overline{CSx}$ release [1]	–	15	–	15	ns
t_{65CC}	$\overline{CSx}$ drive	-4	15	-4	15	ns
t_{66CC}	Other signals release [1]	–	15	–	15	ns
t_{67CC}	Other signals drive	-4	15	-4	15	ns

1. Partially tested, guaranteed by design characterization

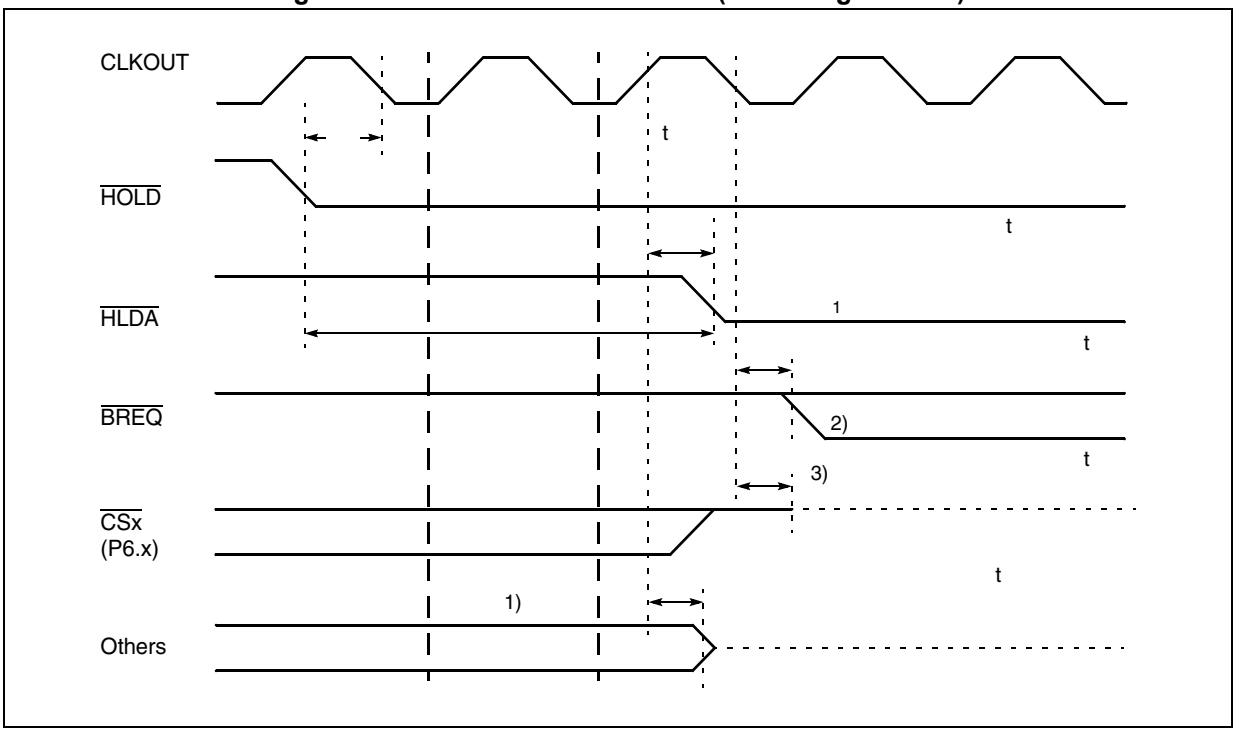
V_{DD} = 5 V ± 10%, V_{SS} = 0V, T_A = -40 to +125°C, C_L = 50 pF

Table 60. External Bus Arbitration (max f_{CPU} = 32 MHz)

Symbol	Parameter	Maximum CPU Clock = 32 MHz		Variable CPU Clock 1/2TCL = 1 to 32 MHz		Unit
		Minimum	Maximum	Minimum	Maximum	
t_{61SR}	$\overline{HOLD}$ input setup time to CLKOUT	20	–	20	–	ns
t_{62CC}	CLKOUT to $\overline{HLDA}$ high or $\overline{BREQ}$ low delay	–	15.625	–	15.625	ns
t_{63CC}	CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	–	15.625	–	15.625	ns
t_{64CC}	$\overline{CSx}$ release [1]	–	15	–	15	ns
t_{65CC}	$\overline{CSx}$ drive	-4	15	-4	15	ns
t_{66CC}	Other signals release [1]	–	15	–	15	ns
t_{67CC}	Other signals drive	-4	15	-4	15	ns

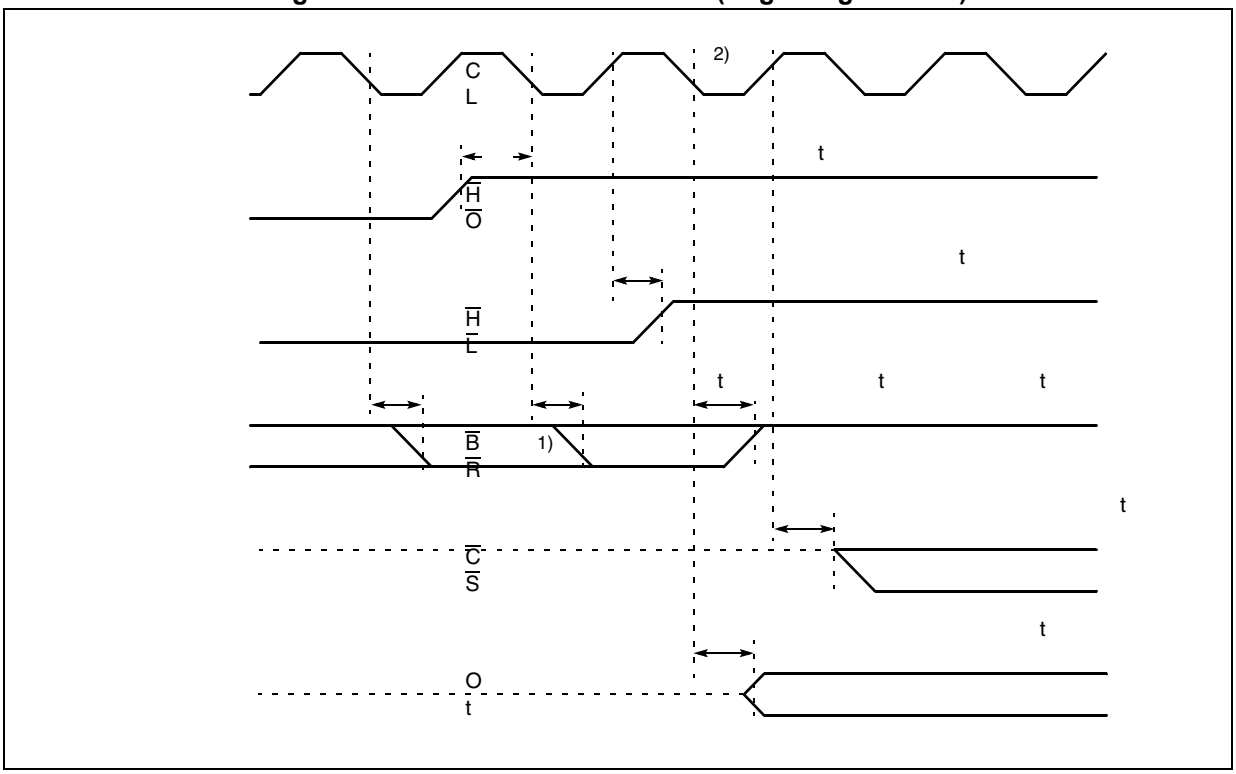
1. Partially tested, guaranteed by design characterization

Figure 78. External Bus Arbitration (Releasing the Bus)



1. The ST10F269 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 79. 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 HOLD going high. Please note that HOLD may also be deactivated without the ST10F269 requesting the bus.
2. The next ST10F269 driven bus cycle may start here.

21.4.14 High-Speed Synchronous Serial Interface (SSC) Timing

Master Mode

V_{CC} = 5 V ±10%, V_{SS} = 0V, CPU clock = 40 MHz, T_A = -40 to +119°C, C_L = 50pF

Table 61. Master Mode (max f_{CPU} = 40 MHz)

Symbol		Parameter	Maximum baud rate = 10 Mbaud (<SSCBR> = 0001h)		Variable baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{300}	CC	SSC clock cycle time	100	100	8 TCL	262144 TCL	ns
t_{301}	CC	SSC clock high time	40	–	t_{300}/2 - 10	–	ns
t_{302}	CC	SSC clock low time	40	–	t_{300}/2 - 10	–	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 [1]	-2	–	-2	–	ns
t_{307p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	37.5	–	2TCL+12.5	–	ns

Table 61. Master Mode (max f_{CPU} = 40 MHz) (continued)

Symbol		Parameter	Maximum baud rate = 10 Mbaud (<SSCBR> = 0001h)		Variable baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
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. Timing guaranteed by design.

The formula for SSC Clock Cycle time is: t_{300} = 4 TCL * (<SSCBR> + 1), where <SSCBR> represents the content of the SSC baud rate register, taken as unsigned 16-bit integer.

V_{CC} = 5 V ±10%, V_{SS} = 0V, CPU clock = 32 MHz, T_A = -40 to +125°C, C_L = 50 pF

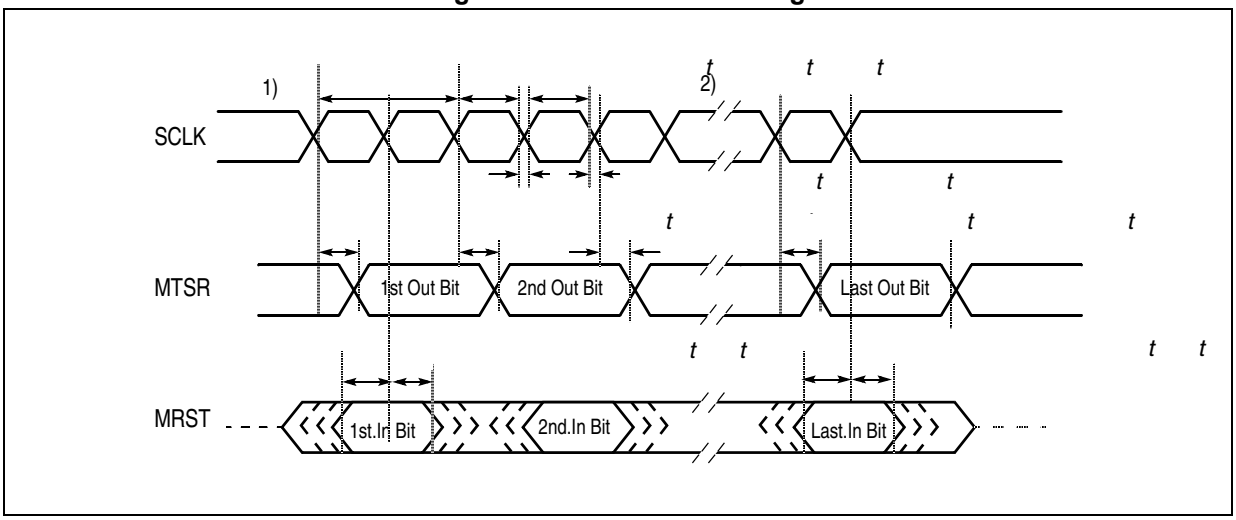
Table 62. Master Mode (max f_{CPU} = 32 MHz)

Symbol		Parameter	Maximum baud rate = 6.25 Mbaud (<SSCBR> = 0001h)		Variable baud rate (<SSCBR>=0001h-FFFFh)		Symbol
			Minimum	Maximum	Minimum	Maximum	
t_{310}	SR	SSC clock cycle time	125	–	8 TCL	262144 TCL	t_{310}
t_{311}	SR	SSC clock high time	52.5	–	t_{310}/2 - 10	–	t_{311}
t_{312}	SR	SSC clock low time	52.5	–	t_{310}/2 - 10	–	t_{312}
t_{313}	SR	SSC clock rise time	–	10	–	10	t_{313}
t_{314}	SR	SSC clock fall time	–	10	–	10	t_{314}
t_{315}	CC	Write data valid after shift edge	–	45.25	–	2 TCL + 14	t_{315}
t_{316}	CC	Write data hold after shift edge	0	–	0	–	t_{316}
t_{317p}	SR	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	78.125	–	4TCL + 15.625	–	t_{317p}
t_{318p}[1]	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	109.375	–	6TCL + 15.625	–	t_{318p}[1]
t_{317}	SR	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	6	–	6	–	t_{317}
t_{318}	SR	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	41.25	–	2TCL + 10	–	t_{318}

1. Timing guaranteed by design.

The formula for SSC Clock Cycle time is: t_{300} = 4 TCL * (<SSCBR> + 1), where <SSCBR> represents the content of the SSC baud rate register, taken as unsigned 16-bit integer.

Figure 80. 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.

Slave mode

V_{CC} = 5 V ±10%, V_{SS} = 0V, CPU clock = 40 MHz, T_A = -40 to +119°C, C_L = 50 pF

Table 63. Slave Mode (max f_{CPU} = 40 MHz)

Symbol		Parameter	Maximum baud rate = 10 Mbaud (<SSCBR> = 0001h)		Variable baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{310}	SR	SSC clock cycle time	100	100	8 TCL	262144 TCL	ns
t_{311}	SR	SSC clock high time	40	–	t_{310}/2 - 10	–	ns
t_{312}	SR	SSC clock low time	40	–	t_{310}/2 - 10	–	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	–	39	–	2 TCL + 14	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}[1]	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

The formula for SSC Clock Cycle time is: t_{310} = 4 TCL * (<SSCBR> + 1), where <SSCBR> represents the content of the SSC baud rate register, taken as unsigned 16-bit integer.

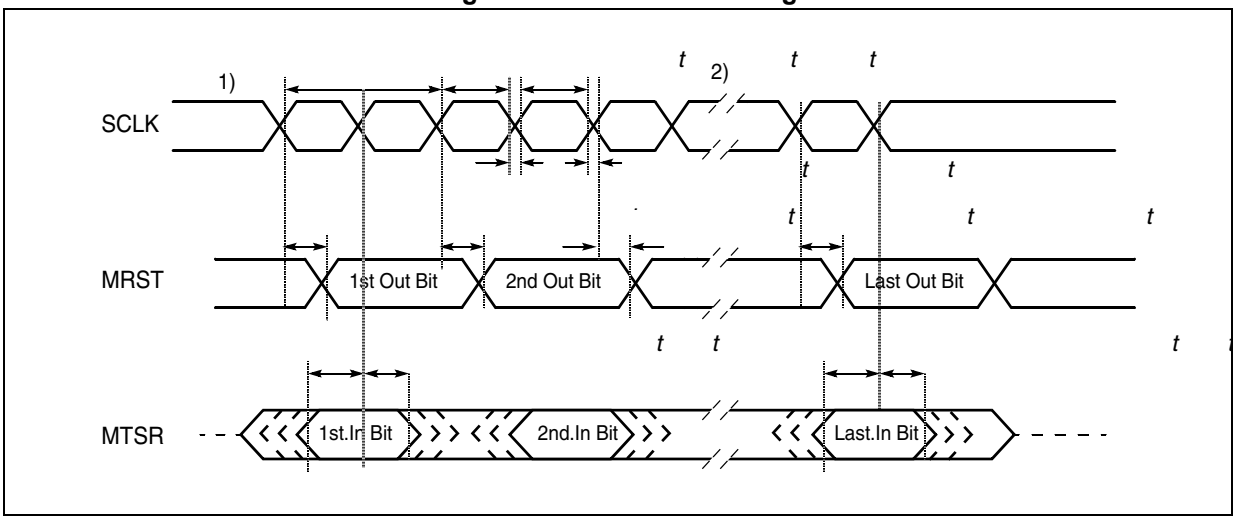
V_{CC} = 5 V ±10%, V_{SS} = 0V, CPU clock = 32 MHz, T_A = -40 to +125°C, C_L = 50 pF

Table 64. Slave Mode (max f_{CPU} = 32 MHz)

Symbol		Parameter	Maximum baud rate = 6.25 Mbaud (<SSCBR> = 0001h)		Variable baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{310}	SR	SSC clock cycle time	125	–	8 TCL	262144 TCL	ns
t_{311}	SR	SSC clock high time	52.5	–	t_{310}/2 - 10	–	ns
t_{312}	SR	SSC clock low time	52.5	–	t_{310}/2 - 10	–	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	–	45.25	–	2 TCL + 14	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)	78.125	–	4TCL + 15.625	–	ns
t_{318p}[1]	SR	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	109.375	–	6TCL + 15.625	–	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)	41.25	–	2TCL + 10	–	ns

The formula for SSC Clock Cycle time is: t_{310} = 4 TCL * (<SSCBR> + 1), where <SSCBR> represents the content of the SSC baud rate register, taken as unsigned 16-bit integer.

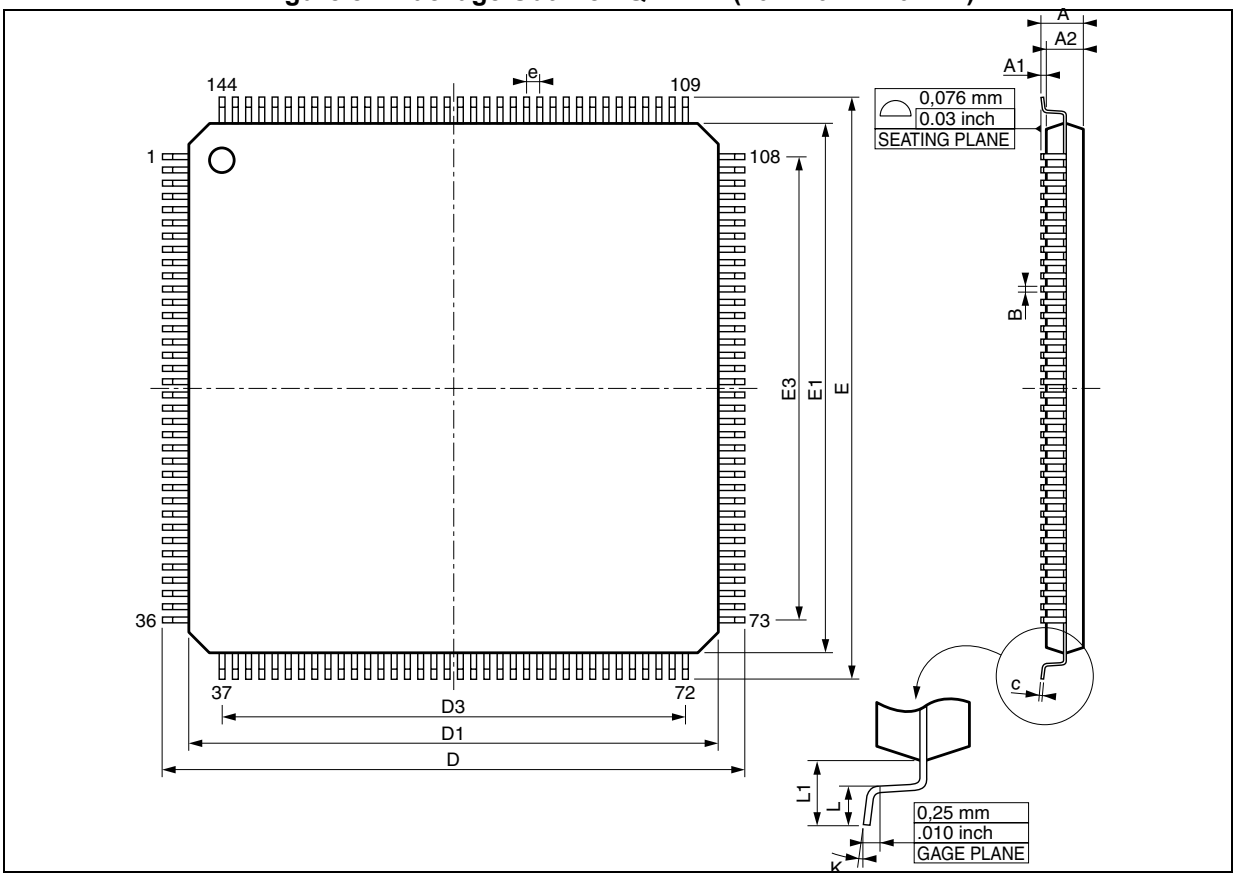
Figure 81. SSC Slave Timing



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.

22 Package Mechanical data

Figure 82. Package Outline TQFP144 (20 x 20 x 1.40 mm)



Dimensions	Millimeters [1]			Inches (approx)		
	Minimum	Typical	Maximum	Minimum	Typical	Maximum
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.17	0.22	0.27	0.0067	0.0087	0.011
C	0.09		0.20	0.0035		0.008
D		22.00			0.866	
D1		20.00			0.787	
D3		17.50			0.689	
e		0.50			0.020	
E		22.00			0.866	
E1		20.00			0.787	
E3		17.50			0.689	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
K	0° (Minimum), 7° (Maximum)					

1. Package dimensions are in mm. The dimensions quoted in inches are rounded.

23 Ordering Information

Table 65. Ordering information

Sales type	Flash Program Memory (bytes)	Temperature range	Package
ST10F269Z2T3	256K	-40°C to +125°C if CPU clock ≤ 32 MHz -40°C to +119°C if 32 MHz < CPU clock ≤ 40 MHz	TQFP144 (20 x 20 x 1.40 mm)
ST10F269Z2T6	256K	-40°C to +85°C if CPU clock ≤ 32 MHz -40°C to +79°C if 32 MHz < CPU clock ≤ 40 MHz	TQFP144 (20 x 20 x 1.40 mm)
ST10F269Z1T3	128K	-40°C to +125°C if CPU clock ≤ 32 MHz -40°C to +119°C if 32 MHz < CPU clock ≤ 40 MHz	TQFP144 (20 x 20 x 1.40 mm)
ST10F269Z1T6	128K	-40°C to +85°C if CPU clock ≤ 32 MHz -40°C to +79°C if 32 MHz < CPU clock ≤ 40 MHz	TQFP144 (20 x 20 x 1.40 mm)

24 Known limitations

24.1 Description

This section describes the functional and electrical problems known in the D revision of the ST10F269Zxxx-D.

The revision number can be found in the third line on the ST10F269 package. It looks like: 'xxxxxxxxx D', where "D" identifies the revision number.

24.2 Functional problems

The following malfunctions are known in this step:

24.2.1 PWRDN.1 - Execution of PWRDN Instruction

When instruction PWRDN is executed while pin NMI is at a high level (if PWRDCFG bit is clear in SYSCON register) or while at least one of the port 2 pins used to exit from power-down mode (if PWRDCFG bit is set in SYSCON register) is at the active level, power down mode is not entered, and the PWRDN instruction is ignored.

However, under the conditions described below, the PWRDN instruction is not ignored, and no further instructions are fetched from external memory, i.e. the CPU is in a quasi-idle state.

This problem only occurs in on of the following situations:

- The instructions following the PWRDN instruction are located in an external memory, and a multiplexed bus configuration with memory tristate waitstate (bit MT-TCx = 0) is used.
- The instruction preceding the PWRDN instruction writes to external memory or an XPeripheral (XRAM,CAN), and the instructions following the PWRDN instruction are located in external memory. In this case, the problem occurs for any bus configuration.

Note: The on-chip peripherals are still working correctly, in particular the Watchdog Timer, if not disabled, resets the device upon an overflow. Interrupts and PEC transfers, however, cannot be processed. In case NMI is asserted low while the device is in this quasi-idle state, power-down mode is entered. No problem occurs if the NMI pin is low (if PWRDCFG = 0) or if all P2 pins used to exit from power-down mode are at inactive level (if PWRDCFG = 1): the chip normally enters powerdown mode.

Workaround

Ensure that no instruction that writes to external memory or an XPeripheral precedes the PWRDN instruction, otherwise insert a NOP instruction in front of PWRDN. When a multiplexed bus with memory tristate wait state is used, the PWRDN instruction must be executed from internal RAM or XRAM.

24.2.2 MAC.9 - CoCMP Instruction Inverted Operands

The ST10 Family Programming Manual describes the CoCMP instruction as: subtracts a 40-bit signed operand from th 40-bit accumulator content (acc - op2\op1), and updates the N, Z and C flags in the MSW register, leaving the accumulator unchanged. On the device

the reverse operation (op2\op1 - acc) has been implemented in the Mac Unit. Therefore, the N and C flags are set according to the reverse operation (Z flag is not affected).

Workaround

Change interpretation of the N and C flags in the MSW register.

Example:

MOV R12, #07h

MOV R13, #06h

MOV R14, #0

CoLOAD R14, R12; Accumulator = 70000h

CoCMP R14, R13; Compares 70000h to 60000h

Here the content of MSW is 0500h, i.e. C = 1, Z = 0 and N = 1.

To test if the Accumulator was greater than or equal the compared value, the "normal" test, according to the description in the ST10 Programming Manual, would be:

JNB MSW.10, Greater ; If C flag cleared, then greater than or equal

With the implementation, this test does not provide the expected result.

To obtain the correct comparison, use instead:

JB MSW.10, Greater ; C flag set: 60000h < 70000h (60000h-70000h implemented)

; i.e. the accumulator is greater than or equal compared value

24.2.3 MAC.10 - E Flag Evaluation for CoSHR and CoASHR Instructions when Saturation Mode is Enabled

The Logical and the Arithmetic Right Shift instructions (CoSHR/CoASHR) are specified not to be affected by the saturation mode (MS bit of the MCW register): the shift operation is always made on the 40 bits of the accumulator. The result shifted in the accumulator is never saturated. Only when the saturation mode is enabled, the evaluation of the E Flag (in the MSW register) is erroneous.

Comment to the example:

In this example below (*Table 66*), the E Flag is kept cleared however MAE is used: bit 0 of MAE has been shifted into bit 15 of MAH. The MAE part has been used and it contents significant bits but the E Flag has not been set.

The content of the flags is given after the execution of the instruction.

Table 66. MAC.10 Example

MS Bit is Set, Saturation Mode is Enabled		Status of Flags After Instruction Execution						
Code	Accumulator value (Hexa)	SL	E	SV	C	Z	N	Remark
MOV R5, #5555h	0x-- ---- ----	-	-	-	-	-	-	-
CoLOAD R5, R5	0x00 5555 5555	0	0	0	0	0	0	Right
NOP	0x00 5555 5555	0	0	0	0	0	0	Right

Table 66. MAC.10 Example (continued)

MS Bit is Set, Saturation Mode is Enabled		Status of Flags After Instruction Execution						
Code	Accumulator value (Hexa)	SL	E	SV	C	Z	N	Remark
MOV MSW, #007Fh	7F 5555 5555	0	0	0	0	0	0	Right
NOP	7F 5555 5555	0	0	0	0	0	0	Right
CoSHR #1	3F AAAA AAAA	0	0*	0	0	0	0	*E is wrong

Workaround

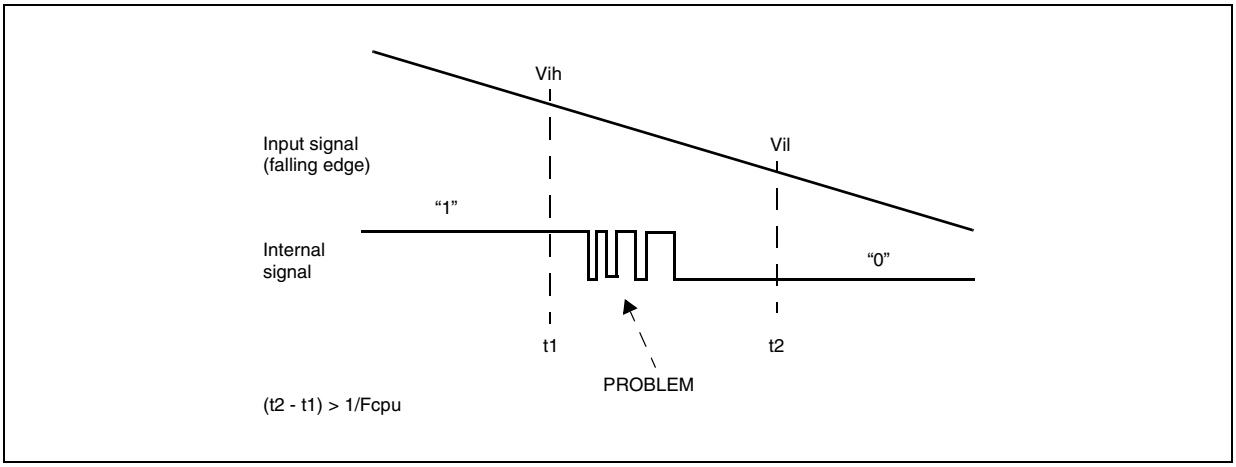
If the MAE flag is used, the saturation mode must be disabled before running Logical and/or Arithmetic Right Shift instructions and re-enable just after.

24.2.4 ST_PORT.3 - Bad Behavior of Hysteresis Function on Input Falling Edge

In the following conditions, a slow falling edge on a ST10F269 input may generate multiple events:

- A falling edge is occurring.
- AND the falling edge has a transition time between Vih and Vil longer than the CPU clock period.

Figure 83. Bad behavior of hysteresis function on input falling edge



Workaround

Add external hardware on the ST10 input in order to have a fast falling edge (lower than 1/Fcpu).

Table 67. History of Fixed Functional Problems of the ST10F269Zxxx-D

Name of modification	Short description	Fixed in step
ST_PORT.2	Wrong Port Direction after Return From Power Down Mode	D

Table 68. Summary of Remaining Functional Problems Known on the ST10F269Zxxx-D

Name	Short description
PWRDN.1	Execution of PWRDN Instruction
MAC.9	CoCMP Instruction Inverted Operands
MAC.10	E Flag Evaluation for CoSHR and CoASHR Instructions when Saturation Mode is Enabled
ST_PORT.3	Bad Behavior of Hysteresis Function on Input Falling Edge

24.3 Deviations from DC/AC preliminary specification

Note about on-chip oscillator

The XTAL2 output is not designed to provide a valid signal when XTAL1 is supplied by an external clock signal. It may happen, if the external clock signal is not perfectly symmetrical and centered on VDD / 2, that XTAL2 signal is not equal to XTAL1. This is due to the design of the oscillator, which has a auto-adaptation gain control dedicated to external crystal.

If an external clock signal is directly provided on XTAL1 pin, then leave XTAL2 pin disconnected to achieve the lowest consumption of the on-chip oscillator.

25 Revision history

Table 69. Document revision history

Date	Revision	Changes
15-Sep-2003	1	Initial release.
28-Sep-2017	2	– PQFP144 package removed (no more in production) – The limitations previously appearing a specific Errata Sheet part, are now available in *Section 24: Known limitations*.