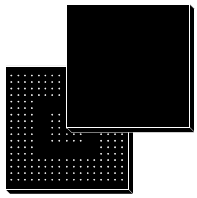
ST10F296E

16-bit MCU with MAC unit, 832 Kbyte Flash memory and 68 Kbyte RAM

Features

- High performance 16-bit CPU with DSP functions
  - 31.25 ns instruction cycle time at 64 MHz max CPU clock
  - Multiply/accumulate unit (MAC) 16 x 16-bit multiplication, 40-bit accumulator
  - Enhanced boolean bit manipulation facilities
  - Single-cycle context switching support
- Memory organization
  - 512 Kbyte Flash memory (32-bit fetch)
  - 320 Kbyte extension Flash memory (16-bit fetch)
  - 100 k erasing/programming cycles
  - Up to 16 Mbyte linear address space for code and data (5 Mbytes with CAN or I^2C)
  - 2 Kbyte on-chip internal RAM (IRAM)
  - 66 Kbyte on-chip extension RAM (XRAM)
  - Programmable external bus characteristics for different address ranges
  - Five programmable chip-select signals
  - Hold-acknowledge bus arbitration support
- Interrupt
  - 8-channel peripheral event controller for single cycle interrupt driven data transfer
  - 16-priority-level interrupt system with 56 sources, sampling rate down to 15.6 ns
- Timers
  - Two multi-functional general purpose timer units with 5 timers
- Two 16-channel capture/compare units
- Analog-to-digital converter (ADC)
  - 32-channel 10-bit
  - 3 µs minimum conversion time
  - TImer for ADC channel injection
- 4-channel PWM unit and 4-channel XPWM



PBGA 208
(23 x 23 x 1.96 mm)

- Serial channels
  - Two synchronous/asynch. serial channels
  - Two high-speed synchronous channels
  - I^2C standard interface
- Two CAN 2.0B interfaces operating on one or two CAN busses (64 or 2 x 32 message objects, C-CAN version)
- Fail-safe protection
  - Programmable watchdog timer
  - Oscillator watchdog
- On-chip bootstrap loader
- Clock generation
  - On-chip PLL and 4-12 MHz oscillator
  - Direct or prescaled clock input
- Real-time clock
- Up to 143 general purpose I/O lines
  - Individually programmable as input, output or special function
  - Programmable threshold (hysteresis)
- Idle, power-down and stand-by modes
- single voltage supply: 5 V ±10% (embedded regulator for 1.8 V core supply).

Table 1. Device summary

Order codes	Temp. range (°C)	CPU freq. range (MHz)
ST10F296	-40 to 125	1 to 64
ST10F296TR		

Contents

List of tables

List of figures

1 Description

The ST10F296E is a derivative of the STMicroelectronics ST10 family of 16-bit single-chip CMOS microcontrollers. It combines high CPU performance (up to 32 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 the phase-locked loop (PLL).

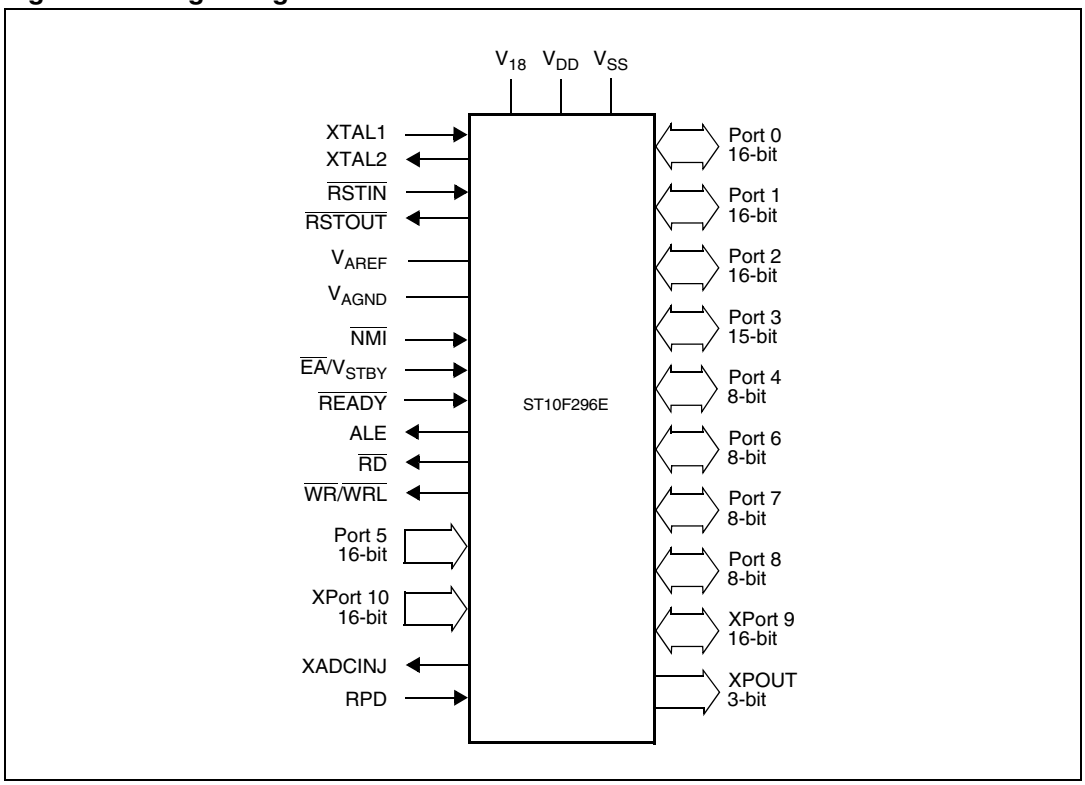
ST10F296E is processed in 0.18 µm CMOS technology. The MCU core and the logic is supplied with a 5 V to 1.8 V on-chip voltage regulator. The part is supplied with a single 5 V supply and I/Os work at 5 V.

The device is upwardly compatible with the ST10F280 device, with the following differences:

- The Flash control interface is now based on STMicroelectronics third generation of standalone Flash memories (M29F400 series), with an embedded program/erase controller. This completely frees up the CPU during programming or erasing of the Flash.
- Pins DC1 and DC2 of ST10F280, are renamed as V_{18}. Do not connect these pins to 5.0 V external supply. Instead, these pin should be connected to a decoupling capacitor (ceramic type, typical value 10 nF, maximum value 100 nF).
- The AC and DC parameters are modified due to a difference in the maximum CPU frequency.
- The $\overline{EA}$ pin has assumed a new, alternate functionality: It is also used to provide a dedicated power supply (see V_{STBY}) to maintain a portion of the XRAM (16 Kbytes) biased when the main power supply of the device (V_{DD} and consequently the internally generated V_{18}) is turned off for low power mode, thereby allowing data retention. V_{STBY} voltage is in the range 4.5-5.5 V, and a dedicated embedded low power voltage regulator provides the 1.8 V for the RAM. The upper limit of up to 6 V may be exceeded for a very short period of time during the global life of the device. The lower limit of 4 V may also be exceeded.
- A second SSC, mapped on the XBus, has been added (SSC of ST10F280 becomes SSC0, while the new SSC is referred to as XSSC or SSC1). There are some restrictions and functional differences due to peculiarities present in the XBus between the classic SSC and the new XSSC.
- A second ASC, mapped on the XBus, has been added (ASC0 of ST10F280 remains ASC0, while the new one is referred to as XASC or ASC1). Some restrictions and functional differences due to peculiarities present in the XBus between the classic ASC, and the new XASC.
- The second PWM (XPWM), mapped on the XBus, has been improved adding set/clear command for safe management of the control register. Memory mapping is thus slightly different.
- An I^2C interface on the XBus has been added (see X-I^2C or simply I^2C interface).
- The CLKOUT function can output either the CPU clock (as in ST10F280) or a software programmable prescaled value of the CPU clock.
- the embedded memory size has been significantly increased (both Flash and RAM).
- PLL multiplication factors have been adapted to new frequency range.

- The ADC is not fully compatible with the ST10F280 (timing and programming model). The formula for the convertion time is still valid, while the sampling phase programming model is different.
- The external memory bus potential limitations on maximum speed and maximum capacitance load are under evaluation and may be introduced: ST10F296E will probably not be able to address an external memory at 64 MHz with 0 wait states.
- The XPERCON register bit mapping has been modified according to new peripheral implementation (which is not fully compatible with ST10F280).
- The bondout chip for emulation (ST10R201) cannot achieve more than 50 MHz at room temperature (so, no real-time emulation is possible at maximum speed).
- Input section characteristics are different. The threshold programmability is extended to all port pins (additional XPICON register); it is possible to select standard TTL (with up to 400 mV of hysteresis) and standard CMOS (with up to 750 mV of hysteresis).
- Output transition is not programmable.
- An RTC module has been added.
- The CAN module has been enhanced: ST10F296E implements two C-CAN modules, so the programming model is slightly different. The possibility to map both CAN modules simultaneously has been added (on P4.5/P4.6).
- The on-chip main oscillator input frequency range has been reshaped, reducing it from 1-25 MHz to 4-12 MHz. This is a high performance oscillator amplifier, that provides a very high negative resistance and wide oscillation amplitude. When this on-chip amplifier is used as a reference for the RTC module, the power-down consumption is dominated by the consumption of the oscillator amplifier itself. A metal option is added to offer a low power oscillator amplifier working in the range 4-8 MHz which allows a power consumption reduction when the RTC is running in power-down mode using the on-chip main oscillator clock as a reference.
- The possibility to reprogram the internal XBus chip select window characteristics (XRAM2 and XFlash address window) has been added.

Figure 1. Logic diagram



2 Ball data

The ST10F296E package is a PBGA measuring 23 x 23 x 1.96 mm. Ball pitch is 1.27 mm. Pin configuration is shown in *Figure 2* while the signal assignment of the balls is given in *Table 2*. This package has 25 additional thermal balls.

Figure 2. Pin configuration (bottom view)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
U	U1 XP10.15	U2 V_{AREF}	U3 V_{AGND}	U4 P5.5	U5 P5.9	U6 P5.13	U7 V_{SS}	U8 V_{DD}	U9 P2.7	U10 V_{SS}	U11 V_{18}	U12 P2.13	U13 V_{SS}	U14 V_{SS}	U15 V_{DD}	U16 V_{SS}	U17 V_{SS}	U
T	T1 XP10.14	T2 P5.0	T3 P5.2	T4 P5.4	T5 P5.8	T6 P5.12	T7 P2.0	T8 P2.3	T9 P2.4	T10 P2.8	T11 P2.11	T12 P2.15	T13 P3.1	T14 P3.4	T15 V_{SS}	T16 V_{SS}	T17 P3.15	T
R	R1 XP10.13	R2 XP10.12	R3 P5.1	R4 P5.3	R5 P5.7	R6 P5.11	R7 P5.15	R8 P2.2	R9 P2.6	R10 P2.9	R11 P2.12	R12 P3.0	R13 P3.3	R14 P3.6	R15 P3.8	R16 P3.9	R17 V_{SS}	R
P	P1 XP10.11	P2 XP10.10	P3 XP10.9	P4 XP10.8	P5 P5.6	P6 P5.10	P7 P5.14	P8 P2.1	P9 P2.5	P10 P2.10	P11 P2.14	P12 P3.2	P13 P3.5	P14 P3.7	P15 P3.11	P16 P3.12	P17 V_{DD}	P
N	N1 XP10.7	N2 XP10.6	N3 XP10.5	N4 XP10.4										N14 P3.10	N15 V_{SS}	N16 P4.0	N17 V_{SS}	N
M	M1 XP10.3	M2 XP10.2	M3 XP10.1	M4 XP10.0										M14 P3.13	M15 P4.1	M16 P4.3	M17 RPD	M
L	L1 V_{SS}	L2 P7.7	L3 XADCINJ	L4 V_{SS}			L7 V_{SS}	L8 V_{SS}	L9 V_{SS}	L10 V_{SS}	L11 V_{SS}			L14 P4.2	L15 P4.4	L16 P4.5	L17 V_{DD}	L
K	K1 V_{DD}	K2 P7.4	K3 P7.5	K4 P7.6			K7 V_{SS}	K8 V_{SS}	K9 V_{SS}	K10 V_{SS}	K11 V_{SS}			K14 P4.6	K15 P4.7	K16 V_{SS}	K17 V_{SS}	K
J	J1 P7.3	J2 P7.2	J3 P7.1	J4 P7.0			J7 V_{SS}	J8 V_{SS}	J9 V_{SS}	J10 V_{SS}	J11 V_{SS}			J14 $\overline{RD}$	J15 $\overline{WR}$	J16 $\overline{READY}$	17 ALE	J
H	H1 V_{SS}	H2 P8.7	H3 P8.6	H4 P8.5			H7 V_{SS}	H8 V_{SS}	H9 V_{SS}	H10 V_{SS}	H11 V_{SS}			H14 P0L.2	H15 P0L.1	H16 P0L.0	H17 $\overline{EA}$	H
G	G1 V_{18}	G2 P8.4	G3 P8.3	G4 V_{SS}			G7 V_{SS}	G8 V_{SS}	G9 V_{SS}	G10 V_{SS}	G11 V_{SS}			G14 P0L.5	G15 P0L.4	G16 P0L.3	G17 V_{DD}	G
F	F1 V_{SS}	F2 P8.2	F3 P8.1	F4 P6.6										F14 P0H.2	F15 P0H.0	F16 P0L.6	F17 V_{SS}	F
E	E1 V_{DD}	E2 P8.0	E3 P6.5	E4 P6.0										E14 P0H.7	E15 P0H.4	E16 P0H.1	E17 P0L.7	E
D	D1 P6.7	D2 P6.4	D3 P6.1	D4 XPOUT0	D5 V_{SS}	D6 V_{SS}	D7 P1H.5	D8 P1H.1	D9 P1L.6	D10 P1L.2	D11 XP9.14	D12 XP9.11	D13 XP9.5	D14 XP9.2	D15 XP9.0	D16 P0H.5	D17 P0H.3	D
C	C1 P6.3	C2 XPOUT3	C3 XPOUT1	C4 $\overline{NMI}$	C5 P1H.6	C6 P1H.7	C7 P1H.4	C8 P1H.0	C9 P1L.7	C10 P1L.3	C11 P1L.0	C12 XP9.13	C13 XP9.10	C14 XP9.6	C15 XP9.3	C16 XP9.1	C17 P0H.6	C
B	B1 P6.2	B2 XPOUT2	B3 V_{SS}	B4 $\overline{RSTOUT}$	B5 V_{SS}	B6 V_{SS}	B7 P1H.3	B8 V_{SS}	B9 V_{SS}	B10 P1L.4	B11 P1L.1	B12 XP9.15	B13 XP9.12	B14 XP9.9	B15 XP9.7	B16 XP9.4	B17 V_{SS}	B
A	A1 V_{SS}	A2 V_{DD}	A3 $\overline{RSTIN}$	A4 V_{SS}	A5 XTAL1	A6 XTAL2	A7 P1H.2	A8 V_{SS}	A9 V_{DD}	A10 P1L.5	A11 V_{SS}	A12 V_{DD}	A13 V_{SS}	A14 V_{DD}	A15 XP9.8	A16 V_{SS}	A17 V_{SS}	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Table 2. Ball description

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P6.0 to P6.7	E4	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 input threshold of Port 6 is selectable (TTL or CMOS).	P6.0	$\overline{CS0}$	Chip select 0 output
	D3	O		P6.1	$\overline{CS1}$	Chip select 1 output
	B1	O		P6.2	$\overline{CS2}$	Chip select 2 output
		I/O			SCLK1	SSC1: Master clock output/slave clock input
	C1	O		P6.3	$\overline{CS3}$	Chip select 3 output
		I/O			MTSR1	SSC1: Master-transmitter/slave-receiver O/I
	D2	O		P6.4	$\overline{CS4}$	Chip select 4 output
		I/O			MRST1	SSC1: Master-receiver/slave-transmitter I/O
	E3	I		P6.5	$\overline{HOLD}$	External master hold request input
	F4	O		P6.6	$\overline{HLDA}$	Hold acknowledge output
	D1	O		P6.7	$\overline{BREQ}$	Bus request output
P8.0 to P8.7	E2	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 CMOS).	P8.0	CC16IO	CAPCOM2: CC16 capture input/compare output
	F3	I/O		P8.1	CC17IO	CAPCOM2: CC17 capture input/compare output
	F2	I/O		P8.2	CC18IO	CAPCOM2: CC18 capture input/compare output
	G3	I/O		P8.3	CC19IO	CAPCOM2: CC19 capture input/compare output
	G2	I/O		P8.4	CC20IO	CAPCOM2: CC20 capture input/compare output
	H4	I/O		P8.5	CC21IO	CAPCOM2: CC21 capture input/compare output
	H3	I/O		P8.6	CC22IO	CAPCOM2: CC22 capture input/compare output
					RxD1	ASC1: Data input (asynchronous) or I/O (synchronous)
	H2	I/O		P8.7	CC23IO	CAPCOM2: CC23 capture input/compare output
		O			TxD1	ASC1: Clock/data output (asynchronous/synchronous)

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P7.0 to P7.7	J4	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 CMOS).	P7.0	POUT0	PWM0: Channel 0 output
	J3	O		P7.1	POUT1	PWM0: Channel 1 output
	J2	O		P7.2	POUT2	PWM0: Channel 2 output
	J1	O		P7.3	POUT3	PWM0: Channel 3 output
	K2	I/O		P7.4	CC28IO	CAPCOM2: CC28 capture input/compare output
	K3	I/O		P7.5	CC29IO	CAPCOM2: CC29 capture input/compare output
	K4	I/O		P7.6	CC30IO	CAPCOM2: CC30 capture input/compare output
	L2	I/O		P7.7	CC31IO	CAPCOM2: CC31 capture input/compare output
XP10.0 to XP10.15	M4	I	16-bit input-only port with Schmitt-Trigger characteristics. The pins of XPort 10 can be the analog input channels (up to 16) for the ADC, where XP10.x equals ANy (analog input channel y, where y = x + 16). The input threshold of XPort 10 is selectable (TTL or CMOS).	XP10.0		
	M3	I		XP10.1		
	M2	I		XP10.2		
	M1	I		XP10.3		
	N4	I		XP10.4		
	N3	I		XP10.5		
	N2	I		XP10.6		
	N1	I		XP10.7		
	P4	I		XP10.8		
	P3	I		XP10.9		
	P2	I		XP10.10		
	P1	I		XP10.11		
	R2			XP10.12		
	R1			XP10.13		
	T1			XP10.14		
	U1			XP10.15		

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P5.0 to P5.15	T2	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 ADC, where P5.x equals ANx (analog input channel x), or they are timer inputs. The input threshold of Port 5 is selectable (TTL or CMOS).	P5.0		
	R3	I		P5.1		
	T3	I		P5.2		
	R4	I		P5.3		
	T4	I		P5.4		
	U4	I		P5.5		
	P5	I		P5.6		
	R5	I		P5.7		
	T5	I		P5.8		
	U5	I		P5.9		
	P6	I		P5.10	T6EUD	GPT2: Timer T6 external up/down control input
	R6	I		P5.11	T5EUD	GPT2: Timer T5 external up/down control input
	T6	I		P5.12	T6IN	GPT2: Timer T6 count input
	U6	I		P5.13	T5IN	GPT2: Timer T5 count input
	P7	I		P5.14	T4EUD	GPT1: Timer T4 external up/down control input
	R7	I		P5.15	T2EUD	GPT1: Timer T2 external up/down control input
P2.0 to P2.15	T7	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 CMOS).	P2.0	CC0IO	CAPCOM1: CC0 capture input/compare output
	P8	I/O		P2.1	CC1IO	CAPCOM1: CC1 capture input/compare output
	R8	I/O		P2.2	CC2IO	CAPCOM1: CC2 capture input/compare output
	T8	I/O		P2.3	CC3IO	CAPCOM1: CC3 capture input/compare output
	T9	I/O		P2.4	CC4IO	CAPCOM1: CC4 capture input/compare output
	P9	I/O		P2.5	CC5IO	CAPCOM1: CC5 capture input/compare output
	R9	I/O		P2.6	CC6IO	CAPCOM1: CC6 capture input/compare output
	U9	I/O		P2.7	CC7IO	CAPCOM1: CC7 capture input/compare output

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P2.0 to P2.15 cont'd	T10	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 CMOS).	P2.8	CC8IO	CAPCOM1: CC8 capture input/compare output
		I			EX0IN	Fast external interrupt 0 input
	R10	I/O		P2.9	CC9IO	CAPCOM1: CC9 capture input/compare output
		I			EX1IN	Fast external interrupt 1 input
	P10	I/O		P2.10	CC10IO	CAPCOM1: CC10 capture input/compare output
		I			EX2IN	Fast external interrupt 2 input
	T11	I/O		P2.11	CC11IO	CAPCOM1: CC11 capture input/compare output
		I			EX3IN	Fast external interrupt 3 input
	R11	I/O		P2.12	CC12IO	CAPCOM1: CC12 capture input/compare output
		I			EX4IN	Fast external interrupt 4 input
	U12	I/O		P2.13	CC13IO	CAPCOM1: CC13 capture input/compare output
		I			EX5IN	Fast external interrupt 5 input
	P11	I/O		P2.14	CC14IO	CAPCOM1: CC14 capture input/compare output
		I			EX6IN	Fast external interrupt 6 input
	T12	I/O		P2.15	CC15IO	CAPCOM1: CC15 capture input/compare output
		I			EX7IN	Fast external interrupt 7 input
		I			T7IN	CAPCOM2: Timer T7 count input

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P3.0 to P3.13, P3.15	R12	I	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 CMOS).	P3.0	T0IN	CAPCOM1: Timer T0 count input
	T13	O		P3.1	T6OUT	GPT2: Timer T6 toggle latch output
	P12	I		P3.2	CAPIN	GPT2: Register caprel capture input
	R13	O		P3.3	T3OUT	GPT1: Timer T3 toggle latch output
	T14	I		P3.4	T3EUD	GPT1: Timer T3 external up/down control input
	P13	I		P3.5	T4IN	GPT1: Timer T4 input for count/gate/reload/capture
	R14	I		P3.6	T3IN	GPT1: Timer T3 count/gate input
	P14	I		P3.7	T2IN	GPT1: Timer T2 input for count/gate/reload/capture
	R15	I/O		P3.8	MRST0	SSC0: Master receive/slave transmit I/O
	R16	I/O		P3.9	MTSR0	SSC0: Master transmit/slave receive O/I
	N14	I/O		P3.10	TxD0	ASC0: Clock/data output (asynchronous/synchronous)
	P15	O		P3.11	RxD0	ASC0: Data input (asynchronous) or I/O (synchronous)
	P16	O		P3.12	$\overline{BHE}$	External memory high byte enable signal
					$\overline{WRH}$	External memory high byte write strobe
	M14	I/O		P3.13	SCLK0	SSC0: Master clock output/slave clock input
	T17	O		P3.15	CLKOUT	Clock output (programmable divider on CPU clock)

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P4.0 to P4.7	N16	O	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 CMOS). Port 4.4, 4.5, 4.6 and 4.7 outputs can be configured as push-pull or open-drain drivers. In case of an external bus configuration, Port 4 can be used to output the segment address lines.	P4.0	A16	Least significant segment address line
	M15	O		P4.1	A17	Segment address line
	L14	O		P4.2	A18	Segment address line
	M16	O		P4.3	A19	Segment address line
	L15	O		P4.4	A20	Segment address line
		I			CAN2_RxD	CAN2: Receive data input
		I/O			SCL	I^2C interface: Serial clock
	L16	O		P4.5	A21	Segment address line
		I			CAN1_RxD	CAN1: Receive data input
		I			CAN2_RxD	CAN2: Receive Data Input
	K14	O		P4.6	A22	Segment address line
		O			CAN1_TxD	CAN1: Transmit data output
		O			CAN2_TxD	CAN2: Transmit data output
	K15	O		P4.7	A23	Most significant segment address line
		O			CAN2_TxD	CAN2: Transmit data output
		I/O			SDA	I^2C interface: Serial data
$\overline{RD}$	J14	O	External memory read strobe: $\overline{RD}$ is activated for every external instruction or data read access.			
$\overline{WR}$ and $\overline{WRL}$	J15	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 access on a 16-bit bus, and, for every data write access on an 8-bit bus. See WRCFG in register SYSCON for mode selection.			
READY and $\overline{READY}$	J16	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 forces the insertion of memory cycle time waitstates until the pin returns to the selected active level.			
ALE	J17	O	Address latch enable output: Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.			
$\overline{EA}$ and V_{STBY}	H17	I	External access enable pin: A low level applied to this pin during and after reset forces the ST10F296E to start the program from the external memory space. A high level forces ST10F296E to start in the internal memory space. This pin is also used (when standby mode is entered: ST10F296E under reset and main V_{DD} turned off) to provide a reference voltage for the low-power embedded voltage regulator which generates the internal 1.8 V supply to retain data inside the standby portion of the XRAM (16 Kbyte). It can range from 4.5 to 5.5 V (6 V for a reduced amount of time during the device life). In running mode, this pin can be tied low during reset without affecting XRAM activities, since the presence of a stable V_{DD} guarantees the proper biasing of this module.			

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
POL.0 to POL.7 and POH.0 to POH.7	H16	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. The input threshold of Port 0 is selectable (TTL or CMOS). In case of an external bus configuration, Port 0 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** P0L.0-P0L.7: D0-D7 (8-bit), D0-D7 (16-bit). P0H.0-P0H.7: I/O (8-bit), D8-D15 (16-bit). **Multiplexed bus modes** P0L.0-P0L.7: AD0-AD7 (8-bit), AD0-AD7 (16-bit). P0H.0-P0H.7: A8-A15 (8-bit), AD8-AD15 (16-bit).	P0L.0		
	H15	I/O		P0L.1		
	H14	I/O		P0L.2		
	G16	I/O		P0L.3		
	G15	I/O		P0L.4		
	G14	I/O		P0L.5		
	F16	I/O		P0L.6		
	E17	I/O		P0L.7		
	F15	I/O		P0H.0		
	E16	I/O		P0H.1		
	F14	I/O		P0H.2		
	D17	I/O		P0H.3		
	E15	I/O		P0H.4		
	D16	I/O		P0H.5		
	C17	I/O		P0H.6		
	E14	I/O		P0H.7		

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
P1L.0 to P1L.7 and P1H.0 to P1H.7	C11	I/O	Port 1 output pins: 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. Port 1 is used as the 16-bit address bus (A) in demultiplexed bus modes: if at least BUSCONx is configured so that the demultiplexed mode is selected, the pis of Port 1 are not available for general purpose I/O function. The input threshold of Port 1 is selectable (TTL or CMOS).	P1L.0		
	B11	I/O		P1L.1		
	D10	I/O		P1L.2		
	C10	I/O		P1L.3		
	B10	I/O		P1L.4		
	A10	I/O		P1L.5		
	D9	I/O		P1L.6		
	C9	I/O		P1L.7		
	C8	I/O		P1H.0		
	D8	I/O		P1H.1		
	A7	I/O		P1H.2		
	B7	I/O		P1H.3		
	C7	I		P1H.4	CC24I	CAPCOM2: CC24 capture input
	D7	I		P1H.5	CC25I	CAPCOM2: CC25 capture input
	C5	I		P1H.6	CC26I	CAPCOM2: CC26 capture input
	C6	I		P1H.7	CC27I	CAPCOM2: CC27 capture input

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)			
XPORT9.0 to XPORT9.15	D15	I/O	16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. XPort 9 outputs can be configured as push-pull or open-drain drivers.The input threshold of XPort 9 is selectable (TTL or CMOS).	XPORT9.0		
	C16	I/O		XPORT9.1		
	D14	I/O		XPORT9.2		
	C15	I/O		XPORT9.3		
	B16	I/O		XPORT9.4		
	D13	I/O		XPORT9.5		
	C14	I/O		XPORT9.6		
	B15	I/O		XPORT9.7		
	A15	I/O		XPORT9.8		
	B14	I/O		XPORT9.9		
	C13	I/O		XPORT9.10		
	D12	I/O		XPORT9.11		
	B13	I/O		XPORT9.12		
	C12	I/O		XPORT9.13		
	D11	I/O		XPORT9.14		
	B12	I/O		XPORT9.15		
XTAL1	A5	I	XTAL1: Input to the oscillator amplifier and/or external clock input.			
XTAL2	A6	O	XTAL2: Output of the oscillator amplifier circuit. 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			
$\overline{\text{RSTIN}}$	A3	I	Reset input with CMOS Schmitt-Trigger characteristics: A low level at this pin for a specified duration while the oscillator is running resets ST10F296E. 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.			
$\overline{\text{RSTOUT}}$	B4	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.			
$\overline{\text{NMI}}$	C4	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 the SYSCON register, when the PWRDN (power-down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the ST10F296E 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 being used, pin $\overline{\text{NMI}}$ should be pulled high externally.			
XPOUT.0	D4	O	XPWM: Channel 0 output			
XPOUT.1	C3	O	XPWM: Channel 1 output			
XPOUT.2	B2	O	XPWM: Channel 2 output			
XPOUT.3	C2	O	XPWM: Channel 3 output			
XADCINJ	L3	O	Output trigger for ADC channel injection			

Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)
V_{AREF}	U2	-	ADC reference voltage and analog supply
V_{AGND}	U3	-	ADC reference and analog ground
RPD	M17	I/O	Timing pin for the return from power-down circuit and synchronous/ asynchronous reset selection.
V_{18}	G1, U11	O	1.8 V decoupling pin: A decoupling capacitor (typical value of 10 nF, max 100 nF) must be connected between this pin and nearest V_{SS} pin.
V_{DD}	A2 A9 A12 A14 E1 K1 U8 U15 P17 L17 G17	-	Digital supply voltage: 5 V during normal operation, idle and power-down modes. It can be turned off when standby RAM mode is selected.

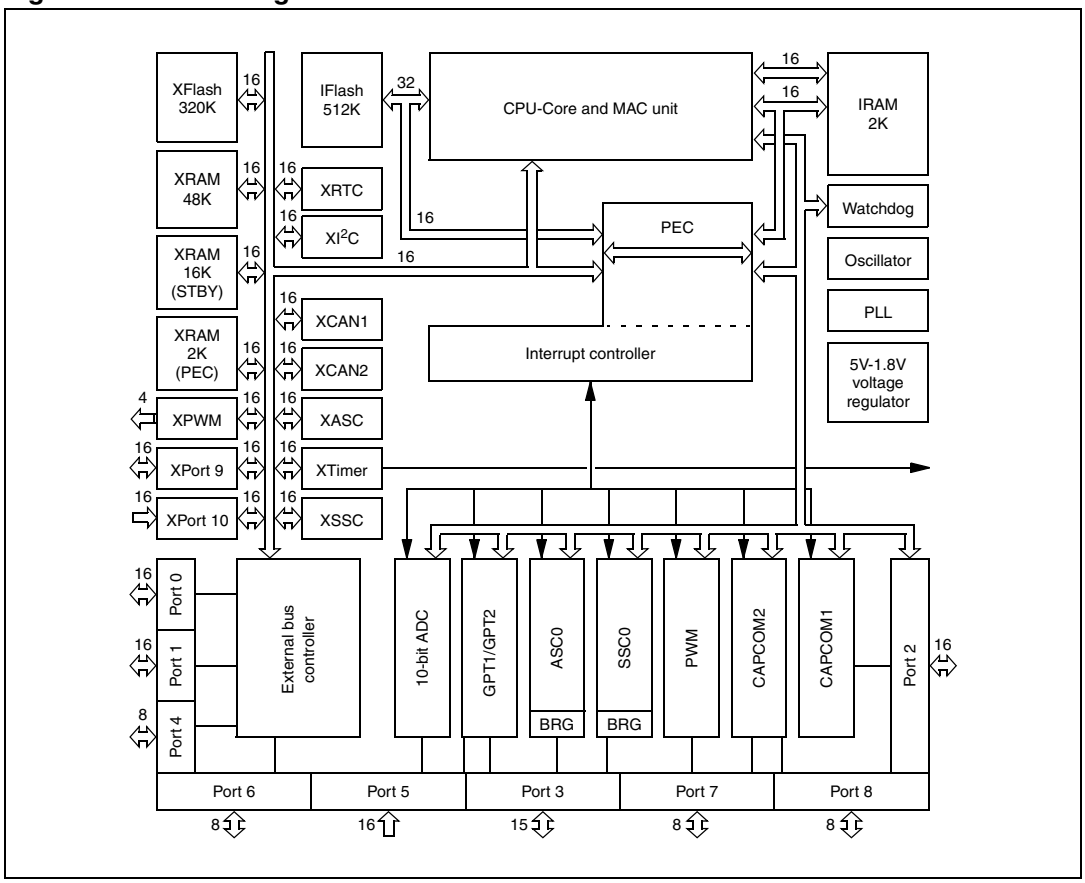
Table 2. Ball description (continued)

Symbol	Ball no.	Type	Function (including port, pin and alternate function where applicable)
V_{SS}	A1, A4 A8, A11, A13, A16 A17, B3, B5 B6, B8 B9, B17, D5, D6 F1, F17, G4, H1 K16, K17, L1, L4 N15, N17, R17, T15, T16, U7, U10, U13, U14, U16, U17	-	Digital ground

3 Functional description

The architecture of the ST10F296E combines advantages of both RISC and CISC processors and an advanced peripheral subsystem. The block diagram of *Figure 3* gives an overview of the different on-chip components and the high bandwidth internal bus structure of the ST10F296E.

Figure 3. Block diagram



4 Memory organization

The memory space of the ST10F296E 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.

The organization of the ST10F296E memory is described in the sections below and shown in *Figure 4: ST10F296E on-chip memory mapping on page 38*.

4.1 IFlash

IFlash comprises 512 Kbytes of on-chip Flash memory. It is divided into 10 blocks (B0F0...B0F9) of Bank 0, and two blocks of Bank 1 (B1F0, B1F1). Read-while-write operations inside the same bank are not allowed. When bootstrap mode is selected, the Test-Flash Block B0TF (8 Kbyte) appears at address 00'0000h. Refer to *Section 5: Internal Flash memory on page 42* for more details on memory mapping in boot mode. The summary of address ranges for IFlash is given in *Table 3*.

Table 3. Address ranges for IFlash

Blocks	User mode	Size (Kbytes)
B0TF	Not visible	8
B0F0	00'0000h - 00'1FFFh	8
B0F1	00'2000h - 00'3FFFh	8
B0F2	00'4000h - 00'5FFFh	8
B0F3	00'6000h - 00'7FFFh	8
B0F4	01'8000h - 01'FFFFh	32
B0F5	02'0000h - 02'FFFFh	64
B0F6	03'0000h - 03'FFFFh	64
B0F7	04'0000h - 04'FFFFh	64
B0F8	05'0000h - 05'FFFFh	64
B0F9	06'0000h - 06'FFFFh	64
B1F0	07'0000h - 07'FFFFh	64
B1F1	08'0000h - 08'FFFFh	64

4.2 XFlash

XFlash comprises 320 Kbytes of on-chip extension Flash memory. The XFLASH address range is 09'0000h - 0E'FFFFh if enabled (if the XPEN bit, bit 2, of the SYSCON register and the XFLASHEN bit, bit 5, of the XPERCON register are set). If the XPEN bit is cleared, any access in the address range 09'0000h - 0E'FFFFh is directed to the external memory interface, using the BUSCONx register corresponding to an address matching the ADDRSELx register. When the XPEN bit is set, but the XFLASHEN and XRAM2EN bits are cleared.

Note: When the Flash control registers are not accessible, no program/erase operations are possible.

XFlash is divided into 3 blocks (B2F0...B0F2) of Bank 2, and two blocks of Bank 3 (B3F0, B3F1). Read-while-write operations inside the same bank are not allowed. Flash control registers are mapped in the range 0E'0000h - 0E'FFFFh. The summary of address range for XFLASH is given in *Table 4*.

Table 4. Address ranges for IFlash

Blocks	User Mode	Size (Kbytes)
B2F0	09'0000h - 09'FFFFh	64
B2F1	0A'0000h - 0A'FFFFh	64
B2F2	0B'0000h - 0B'FFFFh	64
B3F0	0C'0000h - 0C'FFFFh	64
B3F1	0D'0000h - 0D'FFFFh	64
CTRL Registers	0E'0000h - 0E'FFFFh	64

The XFlash is accessed like an external memory in 16-bit demultiplexed bus-mode without read/write delay. The user must set the proper number of waitstates according to the system frequency (1 waitstate for f_{CPU} higher than 40 MHz, 0 waitstates otherwise). Refer to the XFICR register described in *Section 5: Internal Flash memory on page 42*). Byte and word access is allowed.

Note: When the ROMEN and XPEN bits in the SYSCON register are set, together with at least one of the XFLASHEN or XRAM2EN bits in the XPERCON register, the address 08'0000h - 08'FFFFh must be reserved (no external memory access is enabled).

4.3 Internal RAM (IRAM)

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

4.4 Extension RAM (XRAM)

64 Kbyte and 2 Kbytes of on-chip XRAM (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 and second 64 Kbytes, called XRAM1 and XRAM2 respectively, are 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 (31.25 ns access at 64 MHz CPU clock). Byte and word access is allowed.

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

The XRAM2 address range is 0F'0000h - 0F'FFFFh if the XPEN bit and XRAM2EN bit (bit 3 of the XPERCON register) are set. If the XRAM2EN or XPEN bits are cleared, any access in the address range 00'C000h - 00'DFFFh is directed to the external memory interface, using the BUSCONx register corresponding to an address matching the ADDRSELx register. The same thing happens when the XPEN bit is set, but both the XRAM2EN and XFLASHEN bits are cleared.

The lower 16 Kbyte portion of XRAM2 (address range 0F'0000h-0F'3FFFh) represents the standby RAM which can be maintained biased through $\overline{EA}$ / V_{STBY} pin when the main supply V_{DD} is turned off.

As the XRAM appears as external memory, it cannot be used as a system stack or as a register bank. The XRAM is not provided for single bit storage and therefore is not bit addressable.

Note: When the ROMEN bit in the SYSCON register is low, and the XPEN bit is set, and at least one of the two bits XFLASHEN or XRAM2EN in the XPERCON register are also set, the address 08'0000h - 08'FFFFh must be reserved (no external memory access is enabled).

4.5 Special function register (SFR) areas

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

4.6 CAN1

Address range 00'EF00h - 00'EFFFh is reserved for the CAN1 module access. CAN1 is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and the CAN1EN bit (bit 0 of the XPERCON register). Access to the CAN module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two wait states give an access time of 62.5 ns at 64 MHz CPU clock. No tristate wait states are used.

4.7 CAN2

Address range 00'EE00h - 00'EEFFh is reserved for the CAN2 module access. CAN2 is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and the CAN2EN bit (bit 1 of the new XPERCON register). Access to the CAN module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two wait states give an access time of 62.5 ns at 64 MHz CPU clock. No tristate wait states are used.

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

4.8 Real-time clock (RTC)

Address range 00'ED00h - 00'EDFFh is reserved for the RTC module access. The RTC is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 4 of the XPERCON register. Access to the RTC module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used.

4.9 Pulse-width modulation 1 (PWM1)

Address range 00'EC00h - 00'ECFFh is reserved for the PWM1 module access. The PWM1 is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 6 of the XPERCON register. Access to the PWM1 module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used. Only word access is allowed.

4.10 ASC1

Address range 00'E900h - 00'E9FFh is reserved for the ASC1 module access. The ASC1 is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 7 of the XPERCON register. Access to the ASC1 module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used.

4.11 SSC1

Address range 00'E800h - 00'E8FFh is reserved for the SSC1 module access. The SSC1 is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 8 of the XPERCON register. Access to the SSC1 module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used.

4.12 I^2C

Address range 00'EA00h - 00'EAFFh is reserved for the I^2C module access. The I^2C is enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 9 of the XPERCON register. Access to the I^2C module use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used.

4.13 XTimer/XMiscellaneous

Address range 00'EB00h - 00'EB7Fh is reserved for the access to XTimer and to a set of XBus additional features (XMiscellaneous). They are enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 10 of the XPERCON register. Access to these additional modules and features use demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used. In addition to the XTimer module control registers, the following set of features are provided:

- CLKOUT programmable divider
- XBus interrupt management registers
- ADC multiplexing on the P1L register
- Port 1L digital disable register for extra ADC channels
- CAN2 multiplexing on P4.5/P4.6
- CAN1-2 main clock prescaler
- Main voltage regulator disable for power-down mode
- TTL/CMOS threshold selection for Port 0, Port 1, Port 5, XPort 9 and XPort 10.

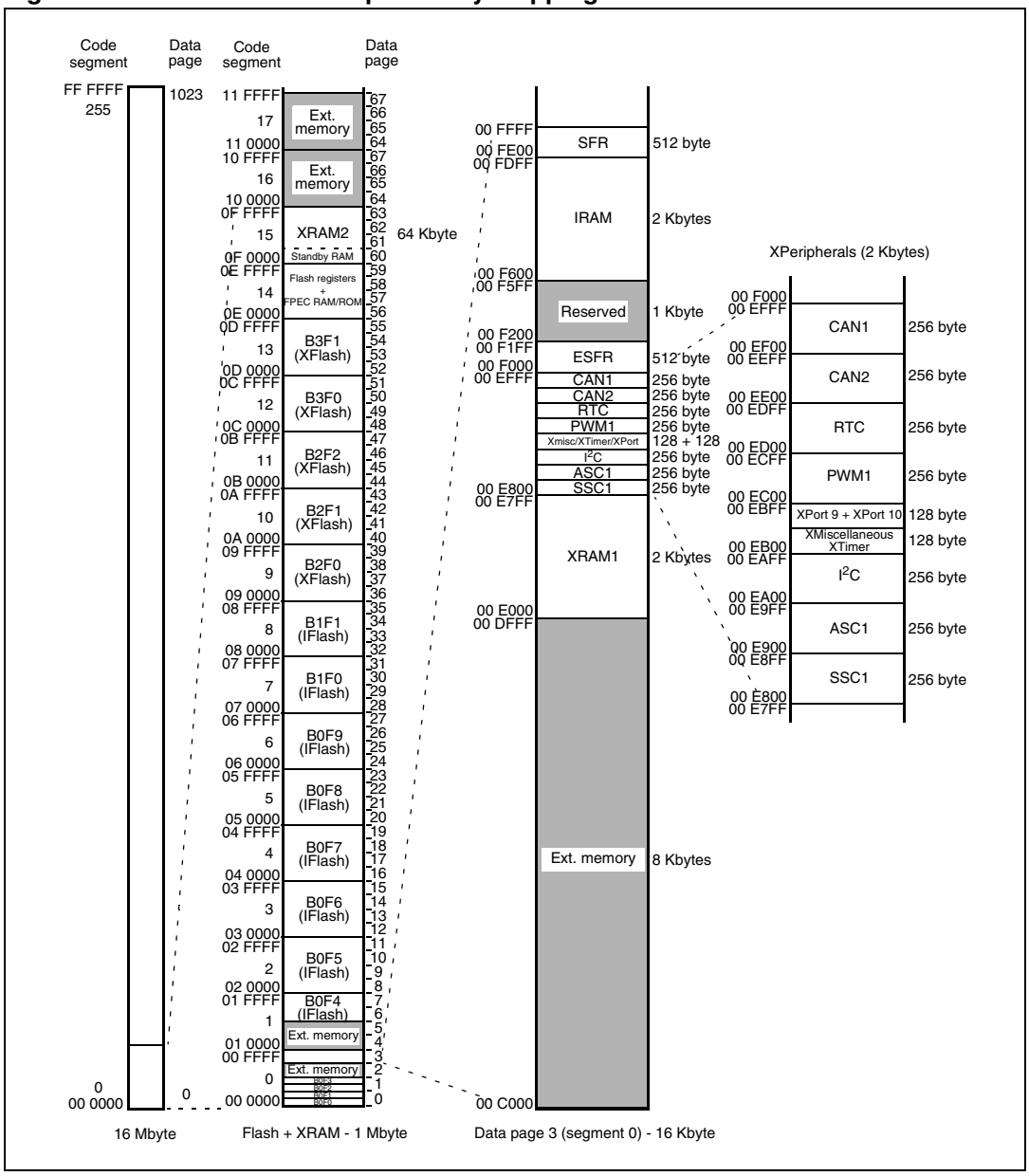
4.14 XPort 9/XPort 10

Address range 00'EB80h - 00'EBFFh is reserved for access to XPort 9 and XPort 10. They are enabled by setting the XPEN bit (bit 2 of the SYSCON register) and bit 11 of the XPERCON register. These additional modules are accessed by demultiplexed addresses and a 16-bit data bus (only word access is possible). Two waitstates give an access time of 62.5 ns at 64 MHz CPU clock. No tristate waitstate is used.

4.15 Visibility of XBus peripherals

To retain compatibility between the ST10F296E and the ST10F280, the XBus peripherals can be selected to be visible and/or accessible on the external address/data bus. Different bits must be set in the XPERCON register to enable the XPeripherals. If these bits are cleared before global enabling with the XPEN bit (in the SYSCON register), the corresponding address space, port pins and interrupts are not occupied by the peripherals, and the peripheral is not visible and not available. Refer to *Section 23: Register set on page 248*.

Figure 4. ST10F296E on-chip memory mapping



1. Blocks B0F0, B0F1, B0F2, B0F3 may be remapped from segment 0 to segment 1 by setting SYSCON-ROMS1 (before EINIT).
2. Data page number and absolute memory address are hexadecimal values.

4.16 XPeripheral configuration registers

XPERCON register

XPERCON (F024h/12h) ESFR Reset value: 005h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	XPORT EN	XMISC EN	XI2C EN	XSSC EN	XASC EN	XPWM EN	XFLASH EN	XRTC EN	XRAM 2EN	XRAM 1EN	CAN 2EN	CAN 1EN
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 5. XPERCON register description

Bit	Bit name	Function
11	XPORTEN	XPort 9 and XPort 10 enable bit 0: Access to the on-chip XPort 9 and XPort 10 modules is disabled. Address range 00'EB80h to 00'EBFFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XI2CEN and XMISCEN are also 0. 1: The on-chip XPort 9 and XPort 10 are enabled and can be accessed.
10	XMISCEN	XBus additional features and XTimer enable bit 0: Access to the additional miscellaneous features is disabled. Address range 00'EB00h to 00'EB7Fh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XI2CEN and XPORTEN are also 0. 1: The additional features and XTimer are enabled and can be accessed.
9	XI2CEN	I^2C enable bit 0: Access to the on-chip I^2C is disabled, external access performed. Address range 00'EA00h to 00'EAFFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip I^2C is enabled and can be accessed.
8	XSSCEN	SSC1 enable bit 0: Access to the on-chip SSC1 is disabled, external access performed. Address range 00'E800h to 00'E8FFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip SSC1 is enabled and can be accessed.
7	XASCEN	ASC1 enable bit 0: Access to the on-chip ASC1 is disabled, external access performed. Address range 00'E900h to 00'E9FFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip ASC1 is enabled and can be accessed.
6	XPWMEN	XPWM enable 0: Access to the on-chip PWM1 module is disabled, external access is performed. Address range 00'EC00h to 00'ECFF is directed to the external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XRTCEN, XMISCEN and XPORTEN are also 0. 1: The on-chip PWM1 module is enabled and can be accessed.

Table 5. XPERCON register description

Bit	Bit name	Function
5	XFLASHEN	XFlash enable bit 0: Access to the on-chip XFlash is disabled, external access is performed. Address range 09'0000h to 0E'FFFFh is directed to the external memory only if XRAM2EN is also 0. 1: The on-chip XFlash is enabled and can be accessed.
4	XRTCEN	RTC enable 0: Access to the on-chip RTC module is disabled, external access is performed. Address range 00'ED00h to 00'EDFF is directed to the external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip RTC module is enabled and can be accessed.
3	XRAM2EN	XRAM2 enable bit 0: Access to the on-chip 64 KByte XRAM is disabled, external access is performed. Address range 0F'0000h to 0F'FFFFh is directed to the external memory only if XFLASHEN is also 0. 1: The on-chip 64 Kbyte XRAM is enabled and can be accessed.
2	XRAM1EN	XRAM1 enable bit 0: Access to the on-chip 2 KByte XRAM is disabled. Address range 00'E000h to 00'E7FFh is directed to the external memory. 1: The on-chip 2 Kbyte XRAM is enabled and can be accessed.
1	CAN2EN	CAN2 enable bit 0: Access to the on-chip CAN2 XPeripheral and its functions is disabled (P4.4 and P4.7 pins can be used as general purpose IOs, but, address range 00'EC00h to 00'EFFFh is directed to the external memory only if CAN1EN, XRTCEN, XASCEN, XSSCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0). 1: The on-chip CAN2 XPeripheral is enabled and can be accessed.
0	CAN1EN	CAN1 enable bit 0: Access to the on-chip CAN1 XPeripheral and its functions is disabled (P4.5 and P4.6 pins can be used as general purpose IOs, but, address range 00'EC00h to 00'EFFFh is directed to the external memory only if CAN2EN, XRTCEN, XASCEN, XSSCEN, XI2CEN, XPWMEN an XMISCEN are also 0). 1: The on-chip CAN1 XPeripheral is enabled and can be accessed.

When CAN1, CAN2, RTC, ASC1, SSC1, I^2C, PWM1, XBus additional features, XTimer and XPort modules are disabled via XPERCON settings, any access in the address range 00'E800h to 00'EFFFh is directed to the external memory interface, using the BUSCONx register associated with the ADDRSELx register matching the target address. All pins involved with the XPeripherals can be used as general purpose IOs whenever the related module is not enabled.

The default XPER selection after reset is identical to configuration of the XBus in the ST10F280. CAN1 and XRAM1 are enabled, CAN2 and XRAM2 are disabled, all other XPeripherals are disabled after reset.

the XPERCON register cannot be changed after globally enabling the XPeripherals (after setting the XPEN bit in the SYSCON register).

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

Reserved bits of the XPERCON register must always be written to 0.

When the RTC is disabled (RTCEN = 0) the main clock oscillator is switched off if the ST10 enters power-down mode. When the RTC is enabled, the RTCOFF bit of the RTCCON register allows the power-down mode of the main clock oscillator to be chosen (eee *Section 18: Real-time clock (RTC) on page 203*).

Table 6 summarizes the address range mapping on segment 8 for programming the ROMEN and XPEN bits (of the SYSCON register) and the XRAM2EN and XFLASHEN bits (of the XPERCON register).

Table 6. Segment 8 address range mapping

ROMEN	XPEN	XRAM2EN	XFLASHEN	Segment 8
0	0	x(1)	x(1)	External memory
0	1	0	0	External memory
0	1	1	x(1)	Reserved
0	1	x(1)	1	Reserved
1	x(1)	x(1)	x(1)	IFlash (B1F1)

1. Don't care

XPEREMU register

The XPEREMU register is a write-only register that is mapped on the XBus memory space at address EB7Eh. It contrasts with the XPERCON register, a read/write ESFR register, which must be programmed to enable the single XBus modules separately.

Once the XPEN bit of the SYSCON register is set and at least one of the XPeripherals (except the memories) is activated, the XPEREMU register must be written with the same content as the XPERCON register. This is to allow a correct emulation of the new set of features introduced on the XBus for the new ST10 generation. The following instructions must be added inside the initialization routine:

```
if (SYSCON.XPEN && (XPERCON & 0x07D3))
then { XPEREMU = XPERCON }
```

XPEREMU must be programmed after both the XPERCON and SYSCON registers in such a way that the final configuration for the XPeripherals is stored in the XPEREMU register and used for the emulation hardware setup.

XPEREMU (EB7Eh) XBus Reset value: xxxxh

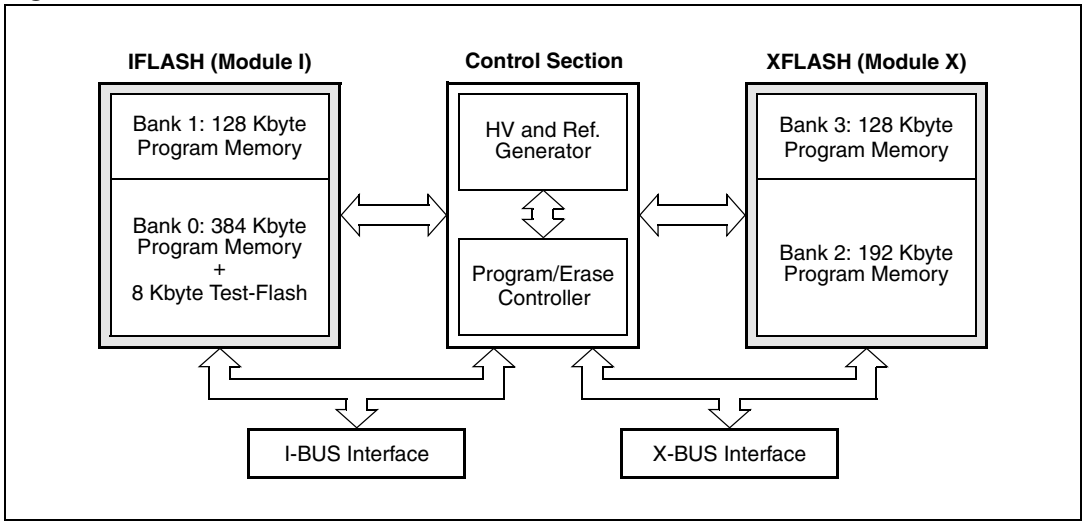
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	XPORT EN	XMISC EN	XI2C EN	XSSC EN	XASC EN	XPWM EN	XFLASH EN	XRTC EN	XRAM 2EN	XRAM 1EN	CAN 2EN	CAN 1EN
-	-	-	-	W	W	W	W	W	W	W	W	W	W	W	W

XPEREMU bit descriptition follows the XPERCON register (see *Table 5: XPERCON register description on page 39*).

5 Internal Flash memory

The on-chip Flash is composed of two matrix modules each one containing one array divided into two banks that can be read and modified independently of the other (i.e. one bank can be read while the other is under modification).

Figure 5. Flash modules structure



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

The data bus is 32 bits wide. Due to ST10 core architecture limitations, only the first 512 Kbytes are accessed at 32-bit (internal Flash bus, also known as IBus), while the remaining 320 Kbytes are accessed at 16-bit (also known as XBus).

5.1 Functional description

5.1.1 Structure

Table 7 shows the address space reserved for the Flash module.

Table 7. Flash module absolute mapping

Description	Addresses	Size (Kbytes)
IFlash sectors	0x00 0000 to 0x08 FFFF	512
XFlash sectors	0x09 0000 to 0x0D FFFF	320
Registers and Flash internal reserved area	0x0E 0000 to 0x0E FFFF	64

5.1.2 Module structure

The IFlash module is composed by two banks. Bank 0 contains 384 Kbytes of program memory divided into 10 sectors. Bank 0 also contains a reserved sector named 'Test-Flash'. Bank 1 contains 128 Kbyte of program memory divided into two sectors (64 Kbytes each).

The XFlash module is also composed of two banks. Bank 2 contains 192 Kbytes of program memory divided into 3 sectors. Bank 3 contains 128 Kbytes of program memory divided into two sectors (64 Kbytes each).

Addresses from 0x0E 0000 to 0x0E FFFF are reserved for the control register interface and other internal service memory space used by the Flash program/erase controller.

Table 8 shows the memory mapping of the Flash when it is accessed in read mode and *Table 9* when it is accessed in write or erase mode.

Note: With this second mapping, the first three banks are remapped into code segment 1 (same result as setting ROMS1 bit in the SYSCON register).

Table 8. Sectorization of the Flash modules (read operations)

Bank	Description	Addresses	Size (Kbytes)	ST10 bus size
B0	Bank 0 Flash 0 (B0F0)	0x0000 0000 - 0x0000 1FFF	8	32-bit (IBus)
	Bank 0 Flash 1 (B0F1)	0x0000 2000 - 0x0000 3FFF	8	
	Bank 0 Flash 2 (B0F2)	0x0000 4000 - 0x0000 5FFF	8	
	Bank 0 Flash 3 (B0F3)	0x0000 6000 - 0x0000 7FFF	8	
	Bank 0 Flash 4 (B0F4)	0x0001 8000 - 0x0001 FFFF	32	
	Bank 0 Flash 5 (B0F5)	0x0002 0000 - 0x0002 FFFF	64	
	Bank 0 Flash 6 (B0F6)	0x0003 0000 - 0x0003 FFFF	64	
	Bank 0 Flash 7 (B0F7)	0x0004 0000 - 0x0004 FFFF	64	
	Bank 0 Flash 8 (B0F8)	0x0005 0000 - 0x0005 FFFF	64	
	Bank 0 Flash 9 (B0F9)	0x0006 0000 - 0x0006 FFFF	64	
B1	Bank 1 Flash 0 (B1F0)	0x0007 0000 - 0x0007 FFFF	64	
	Bank 1 Flash 1 (B1F1)	0x0008 0000 - 0x0008 FFFF	64	
B2	Bank 2 Flash 0 (B2F0)	0x0009 0000 - 0x0009 FFFF	64	16-bit (X-BUS)
	Bank 2 Flash 1 (B2F1)	0x000A 0000 - 0x000A FFFF	64	
	Bank 2 Flash 2 (B2F2)	0x000B 0000 - 0x000B FFFF	64	
B3	Bank 3 Flash 0 (B3F0)	0x000C 0000 - 0x000C FFFF	64	
	Bank 3 Flash 1 (B3F1)	0x000D 0000 - 0x000D FFFF	64	

Table 9. Sectorization of the Flash modules (write operations or with ROMS1 = 1)

Bank	Description	Addresses	Size (Kbytes)	ST10 bus size
B0	Bank 0 Test-Flash (B0TF)	0x0000 0000 - 0x0000 1FFF	8	32-bit (IBus)
	Bank 0 Flash 0 (B0F0)	0x0001 0000 - 0x0001 1FFF	8	
	Bank 0 Flash 1 (B0F1)	0x0001 2000 - 0x0001 3FFF	8	
	Bank 0 Flash 2 (B0F2)	0x0001 4000 - 0x0001 5FFF	8	
	Bank 0 Flash 3 (B0F3)	0x0001 6000 - 0x0001 7FFF	8	
	Bank 0 Flash 4 (B0F4)	0x0001 8000 - 0x0001 FFFF	32	
	Bank 0 Flash 5 (B0F5)	0x0002 0000 - 0x0002 FFFF	64	
	Bank 0 Flash 6 (B0F6)	0x0003 0000 - 0x0003 FFFF	64	
	Bank 0 Flash 7 (B0F7)	0x0004 0000 - 0x0004 FFFF	64	
	Bank 0 Flash 8 (B0F8)	0x0005 0000 - 0x0005 FFFF	64	
	Bank 0 Flash 9 (B0F9)	0x0006 0000 - 0x0006 FFFF	64	
B1	Bank 1 Flash 0 (B1F0)	0x0007 0000 - 0x0007 FFFF	64	
	Bank 1 Flash 1 (B1F1)	0x0008 0000 - 0x0008 FFFF	64	
B2	Bank 2 Flash 0 (B2F0)	0x0009 0000 - 0x0009 FFFF	64	16-bit (XBus)
	Bank 2 Flash 1 (B2F1)	0x000A 0000 - 0x000A FFFF	64	
	Bank 2 Flash 2 (B2F2)	0x000B 0000 - 0x000B FFFF	64	
B3	Bank 3 Flash 0 (B3F0)	0x000C 0000 - 0x000C FFFF	64	
	Bank 3 Flash 1 (B3F1)	0x000D 0000 - 0x000D FFFF	64	

Table 9 refers to the configuration when bit ROMS1 of the SYSCON register is set. When bootstrap mode is entered:

- Test-Flash is seen and is available for code fetches (address 00’0000h)
- User IFlash is only available for read and write access
- Write access must be made using addresses in segment 1 that start at 01'0000h, irrespective of the ROMS1 bit value in the SYSCON register. Note that the user must not rely on the ROMS1 bit because it is ‘don't care’ for write operations.
- Read access is made in segment 0 or in segment 1 depending on the ROMS1 value.

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

Example

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

Table 10 shows the composition of the control register interface. These registers can be addressed by the CPU

Table 10. Control register interface

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

5.1.3 Low power mode

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

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

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

5.2 Write operation

The Flash modules have a single register interface mapped in the memory space of the XFlash module (0x0E 0000 to 0x0E 0013). All operations are enabled through four 16-bit control registers: Flash control register 1-0 high/low (FCR1H/L-FCR0H/L). Eight other 16-bit registers are used to store Flash addresses and data for program operations (FARH/L and FDR1H/L-FDR0H/L) and write operation error flags (FERH/L). All registers are accessible with 8 and 16-bit instructions (since they are mapped on the ST10 XBus).

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

The four Flash module banks have their own dedicated sense amplifiers, so that any bank can be read while any other bank is written. However simultaneous write operations ('write' meaning either program or erase) on different banks are forbidden. When a write operation is occurring in the Flash, no other write operations can be performed.

During a Flash write operation any attempt to read the bank under modification outputs invalid data (software trap 009Bh). This means that the Flash bank is not fetchable when a write operation is active. The write operation commands must be executed from another bank, from the other module or from another memory (internal RAM or external memory).

Note: During a write operation, when the LOCK bit of the FCR0 register is set, it is forbidden to write into the Flash control registers.

5.2.1 Power supply drop

If, during a write operation, the internal low voltage supply drops below a certain internal voltage threshold, any write operation that is running is suddenly interrupted and the modules are reset to read mode. Following power-on, an interrupted Flash write operation must be repeated.

5.3 Internal Flash memory registers

Flash control register 0 low (FCR0L)

The Flash control register 0 low (FCR0L) together with the Flash control register 0 high (FCR0H) is used to enable and to monitor all the write operations for both Flash modules. The user has no access in write mode to the Test-Flash (B0TF). The Test-Flash block is only seen by the user in bootstrap mode.

FCR0L (0x0E 0000) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									BSY1	BSY0	LOCK	Res.	BSY3	BSY2	Res.
-									R	R	R		R	R	

Table 11. FCR0L register decription

Bit	Bit name	Function
15-7	-	Reserved
6-5	BSY[1:0]	Bank 1:0 busy (IFlash) These bits indicate that a write operation is running in the corresponding bank of IFlash. They are automatically set when the WMS bit of the FCR0H register is set. When the BSY [1:0] bits are set every read access to the corresponding bank outputs invalid data (software trap 009Bh), while every write access to the bank is ignored. At the end of a write operation or during a program or erase suspend these bits are automatically reset and the bank returns to read mode. After a program or erase resume these bits are automatically reset.

Table 11. FCR0L register decription (continued)

Blt	Bit name	Function
4	LOCK	Flash registers access locked When this bit is set, access to the Flash control registers FCR0H/L-FCR1H/L, FDR0H/L-FDR1H/L, FARH/L and FER is locked by the FPEC. Any read access to the registers outputs invalid data (software trap 009Bh) and any write access is ineffective. The LOCK bit is automatically set when the Flash bit WMS of the FCR0H register is set. The LOCK bit is the only bit the user can always access to detect the status of the Flash. If it is low, the remainder of the FCR0L and all other Flash registers are accessible by the user. *Note: When the LOCK bit is low, the FER register content can be read, but, its content is updated only when the BSY bits are reset.*
3	-	Reserved
2-1	BSY[3:2]	Bank 3:2 busy (XFlash) These bits indicate that a write operation is running on the corresponding bank of XFlash. They are automatically set when bit WMS in the FCR0H register is set. Setting the protection operation automatically sets the BSY2 bit (since the protection registers are in Block B2). When both busy (XFlash) bits are set, every read access to the corresponding bank outputs invalid data (software trap 009Bh), while every write access to the bank is ignored. At the end of a write operation or during a program or erase suspend these bits are automatically reset and the bank returns to read mode. After a program or erase resume these bits are automatically reset.
0	-	Reserved

Flash control register 0 high (FCR0H)

The Flash control register 0 high (FCR0H) together with the Flash control register 0 low (FCR0L) is used to enable and monitor write operations for both the Flash modules. The user has no access in write mode to the Test-Flash (B0TF). The Test-Flash block is only seen by the user in bootstrap mode.

FCR0H (0x0E 0002) FCR Reset value: 0000h

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

Table 12. FCR0H register decription

Bit	Bit name	Function
15	WMS	Write mode start This bit must be set to start every write operation in the Flash modules. At the end of the write operation or during a suspend, this bit is automatically reset. To resume a suspended operation, this bit must be set again. It is forbidden to set this bit if the ERR bit of the FER register is high (the operation is not accepted). It is also forbidden to start a new write (program or erase) operation (by setting the WMS bit high) when the SUSP bit of the FCR0 register is high. Resetting this bit by software has no effect.
14	SUSP	Suspend This bit must be set to suspend the current program (word or double word) or sector erase operation to read data in one of the sectors of the bank under modification or to program data in another bank. The suspend operation resets the Flash bank to normal read mode (automatically resetting bits BSYx). When in program suspend, the two Flash modules accept only read and program resume operations. When in erase suspend, the modules accept only read, erase resume, and program (word or double word) operations. Program operations cannot be suspended during erase suspend. To resume the suspended operation, the WMS bit must be set again, together with the selection bit corresponding to the operation to resume (WPG, DWPG, SER). *Note: It is forbidden to start a new write operation with the SUSP bit already set.*
13	WPG	Word program This bit must be set to select the word (32 bits) program operation in the Flash modules. The word program operation allows 0s to be programmed instead of 1s. The Flash address to be programmed must be written in the FARH/L registers, while the Flash data to be programmed must be written in the FDR0H/L registers before starting the execution by setting the WMS bit. The WPG bit is automatically reset at the end of the word program operation.

Table 12. FCR0H register decription (continued)

Bit	Bit name	Function
12	DWPG	Double word program This bit must be set to select the double word (64 bits) program operation in the Flash modules. The double word program operation allows 0s to be programmed instead of 1s. The Flash address in which to program (aligned with even words) must be written in the FARH/L registers, while the two Flash data to be programmed must be written in the FDR0H/L registers (even word) and FDR1H/L registers (odd word) before starting the execution by setting the WMS bit. The DWPG bit is automatically reset at the end of the double word program operation.
11	SER	Sector erase This bit must be set to select the sector erase operation in the Flash modules. The sector erase operation allows all Flash locations to 0xFF to be erased. 1 to all sectors of the same bank (excluding the Test-Flash for Bank B0) can be erased through bits BxFy of the FCR1H/L registers before starting the execution by setting the WMS bit. Preprogramming the sectors to 0x00 is done automatically. The SER bit is automatically reset at the end of the sector erase operation.
10-9	-	Reserved
8	SPR	Set protection This bit must be set to select the set protection operation. The set protection operation allows 0s to be programmed instead of 1s in the Flash non-volatile protection registers. The Flash address in which to program must be written in the FARH/L registers, while the Flash data to be programmed must be written in the FDR0H/L before starting the execution by setting the WMS bit. A sequence error is flagged by the SEQER bit of the FER register if the address written in FARH/L is not in the range 0x0EDFB0-0x0EDFBF. The SPR bit is automatically reset at the end of the set protection operation.
7	SMOD	Select module If this bit is reset, a write operation is performed on the XFlash module. if this bit is set, a write operation is performed on IFlash module. The SMOD bit is automatically reset at the end of the write operation.
6-0	-	Reserved

Flash control register 1 low (FCR1L)

The Flash control register 1 low (FCR1L) and the Flash control register 1 high (FCR1H) are used to select the sectors to erase or they are used, during any write operation, to monitor the status of each sector of the module selected by the SMOD bit of the FCR0H register. FCR1L is shown below when SMOD = 0 and when SMOD = 1.

FCR1L (0x0E 0004) SMOD = 0 FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													B2F2	B2F1	B2F0
-													R	R	R

Table 13. FCR1L register description (SMOD = 0, XFlash selected)

Bit	Bit name	Function
15-3	-	Reserved
2-0	B2F[2:0]	Bank 2 XFlash sector 2:0 status These bits must be set during a sector erase operation to select the sectors to be erased in Bank 2. During any erase operation, these bits are automatically set and give the status of the three sectors of Bank 2 (B2F2-B2F0). The meaning of B2Fy bit for sector y of Bank 2 is given in *Table 17*. The BTF [2:0] bits are automatically reset at the end of a write operation if no errors are detected.

FCR1L (0x0E 0004) SMOD = 1 FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						B0F9	B0F8	B0F7	B0F6	B0F5	B0F4	B0F3	B0F2	B0F1	B0F0
-						RS	RS	RS	RS	RS	RS	RS	RS	RS	RS

Table 14. FCR1L register description (SMOD = 1, IFlash selected)

Bit	Bit name	Function
15-10	-	Reserved
9-0	B0F[9:0]	Bank 0 IFlash sector 9:0 status These bits must be set during a sector erase operation to select the sectors to be erased in Bank 0. During any erase operation, these bits are automatically set and give the status of the 10 sectors of Bank 0 (B0F9-B0F0). The meaning of B0Fy bit for sector y of Bank 0 is given in *Table 17*. The B0F [9:0] bits are automatically reset at the end of a write operation if no errors are detected.

Flash control register 1 high (FCR1H)

The Flash control register 1 high (FCR1H) and the Flash control register 1 low (FCR1L) are used to select the sectors to erase, or they are used, during any write operation, to monitor the status of each sector and each bank of the module selected by the SMOD bit of the FCR0H register. FCR1H is shown below when SMOD = 0 and when SMOD = 1.

FCR1H (0x0E 0006) SMOD = 0 FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						B3S	B2S	Reserved						B3F1	B3F0
-						RS	RS	-						RS	RS

Table 15. FCR1H register description (SMOD = 0, XFlash selected)

Bit	Bit name	Function
15-10	-	Reserved
9-8	B[3:2]S	Bank 3-2 status (XFlash) During any erase operation, these bits are automatically modified and give the status of the two banks, B3-B2. The meaning of the BxS bit for Bank x is given in *Table 17*. Bits B[3:2]S are automatically reset at the end of a erase operation if no errors are detected.
7-2	-	Reserved
1-0	B3F[1:0]	Bank 3 XFlash sector 1:0 status During any erase operation, these bits are automatically set and give the status of the two sectors of Bank 3 (B3F1-B3F0). The meaning of B3Fy bit for sector y of Bank 1 is given in *Table 17*. Bits B3F[1:0] are automatically reset at the end of a erase operation if no errors are detected.

FCR1H (0x0E 0006) SMOD = 1 FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						B1S	B0S	Reserved						B1F1	B1F0
-						RS	RS	-						RS	RS

Table 16. FCR1H register description (SMOD = 1, IFlash selected)

Bit	Bit name	Function
15-10	-	Reserved
9-8	B[1:0]S	Bank 1-0 status (IFlash) During any erase operation, these bits are automatically modified and give the status of the two banks, B1-B0. The meaning of BxS bit for Bank x is given in *Table 17*. Bits B[1:0]S are automatically reset at the end of a erase operation if no errors are detected.
7-2	-	Reserved
1-0	B1F[1:0]	Bank 1 IFlash sector 1:0 status During any erase operation, these bits are automatically set and give the status of the two sectors of Bank 1 (B1F1-B1F0). The meaning of B1Fy bit for sector y of Bank 1 is given in *Table 17*. These bits are automatically reset at the end of a erase operation if no errors are detected.

Table 17. Banks (BxS) and sectors (BxFy) status bits meaning

ERR	SUSP	BxS = 1 meaning	BxFy = 1 meaning
1	-	Erase error in Bank x	Erase error in sector y of Bank x
0	1	Erase suspended in Bank x	Erase suspended in sector y of Bank x
0	0	Don't care	Don't care

Flash data register 0 low (FDR0L)

The Flash address registers (FARH/L) and the Flash data registers (FDR1H/L-FDR0H/L) are used during program operations to store Flash addresses and data to program.

FDR0L (0x0E 0008) FCR Reset value: FFFFh

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

Table 18. FDR0L register description

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

Flash data register 0 high (FDR0H)

FDR0H (0x0E 000A) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN 31	DIN 30	DIN 29	DIN 28	DIN 27	DIN 26	DIN 25	DIN 24	DIN 23	DIN 22	DIN 21	DIN 20	DIN 19	DIN 18	DIN 17	DIN 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 19. FDR0H register description

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

Flash data register 1 low (FDR1L)

FDR1L (0x0E 000C) FCR Reset value: FFFFh

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

Table 20. FDR1L register description

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

Flash data register 1 high (FDR1H)

FDR1H (0x0E 000E) FCR Reset value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN 31	DIN 30	DIN 29	DIN 28	DIN 27	DIN 26	DIN 25	DIN 24	DIN 23	DIN 22	DIN 21	DIN 20	DIN 19	DIN 18	DIN 17	DIN 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 21. FDR1H register description

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

Flash address register low (FARL)

FARL (0x0E 0010) FCR Reset value: 0000h

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

Table 22. FARL register description

Bit	Bit name	Function
15-2	ADD[15:2]	Address 15:2 These bits must be written with the address of the Flash location to program in the following operations: Word program (32-bit) and double word program (64-bit). In double word program the ADD2 bit must be written to 0.
1-0	-	Reserved

Flash address register high (FARH)

FARH (0x0E 0012) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ADD 20	ADD 19	ADD 18	ADD 17	ADD 16
-											RW	RW	RW	RW	RW

Table 23. FARH register description

Bit	Bit name	Function
15-5	-	Reserved
4-0	ADD[20:16]	Address 20:16 These bits must be written with the address of the Flash location to program in the following operations: Word program and double word program.

Flash error register (FER)

The Flash error register (and all other Flash registers) can only be properly read once the LOCK bit of the FCR0L register is low. Nevertheless, its content is updated when the BSY bits are reset. For this reason, it is meaningful to read the FER register content only when the LOCK bit and all BSY bits are cleared.

FER (0xE 0014) FCR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WPF	RESER	SEQER	Reserved		10ER	PGER	ERER	ERR
-							RC	RC	RC	-		RC	RC	RC	RC

Table 24. FER register description

Bit	Bit name	Function
15-9	-	Reserved
8	WPF	Write protection flag This bit is automatically set when trying to program or erase in a sector that is write protected. In the case of a multiple sector erase, unprotected sectors are erased, protected sectors are not erased, and the WPF bit is set. The WPF bit has to be reset by software.
7	RESER	Resume error This bit is automatically set when a suspended program or erase operation is not resumed correctly due to a protocol error. In this case the suspended operation is aborted. This bit has to be reset by software.
6	SEQER	Sequence error This bit is automatically set when the control registers (FCR1H/L-FCR0H/L, FARH/L, FDR1H/L-FDR0H/L) are not correctly filled to execute a valid write operation. In this case no write operation is executed. This bit has to be reset by software.
5-4	-	Reserved
3	10ER	1 over 0 error This bit is automatically set when trying to program bits to 1 that have previously been set to 0 (this does not happen when programming the protection bits). This error is not due to a failure of the Flash cell. It flags that the desired data has not been written. The 10ER bit has to be reset by software.
2	PGER	Program error This bit is automatically set when a program error occurs during a Flash write operation. This error is due to a failure of a Flash cell that can no longer be programmed. The word where this error occurred must be discarded. This bit has to be reset by software.

Table 24. FER register description (continued)

Bit	Bit name	Function
1	ERER	Erase error This bit is automatically set when an erase error occurs during a Flash write operation. This error is due to a failure of a Flash cell that can no longer be erased. This kind of error is fatal and the sector where it occurred must be discarded. This bit has to be reset by software.
0	ERR	Write error This bit is automatically set when an error occurs during a Flash write operation or when a bad operation setup is written. Once the error has been discovered and understood, the ERR bit must be reset by software.

XFlash interface control register (XFICR)

This register is used to configure the XFlash interface behavior on the XBus. It allows the number of wait states introduced on the XBus to be set before the internal $\overline{\text{READY}}$ signal is given to the ST10 bus master.

XFICR (0xE E000h) XBus Reset value: 000Fh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												WS3	WS2	WS1	WS0
-												RW	RW	RW	RW

Table 25. XFICR register description

Bit	Bit name	Function
15-4	-	Reserved
3-0	WS[3:0]	Wait state setting These three bits are the binary coding of the wait state number introduced by the XFlash interface through the internal $\overline{\text{READY}}$ signal of the XBus. The default value after reset is 1111, where up to 15 wait states are set. Recommendations for the ST10F296E include: For f_{CPU} > 40 MHz: 1 wait state WS[3:0] = 0001 For f_{CPU} ≤40 MHz: 0 wait state WS[3:0] = 0000

5.4 Protection strategy

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

Protection can be programmed using the set protection operation (see the Flash control registers of *Section 5.3*), that can be executed from all the internal or external memories except from the Flash bank, B2.

Two kinds of protection are available:

- Write protections to avoid unwanted writings
- Access protections to avoid piracy

5.4.1 Protection registers

This section describes the seven non-volatile protection registers and their architectural limitations. These registers are one time programmable.

Four registers (FNVWPXRL/H-FNVWPIRL/H) are used to store the write protection fuses respectively for each sector of the XFlash module (see 'X' in the sections below) and IFlash module (see 'I' in the sections below). The other three registers (FNVAPR0 and FNVAPR1L/H) are used to store the access protection fuses (common to both Flash modules, though, with some limitations).

Flash non-volatile write protection X register low (FNVWPXRL)

FNVWPXRL (0x0E DFB0) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W2PPR	Reserved												W2P2	W2P1	W2P0
RW	-												RW	RW	RW

Table 26. FNVWPXRL register description

Bit	Bit name	Function
15	W2PPR	Write protection Bank 2 non-volatile cells This bit, if programmed at 0, disables any write access to the non-volatile cells of Bank 2. Since these non-volatile cells are dedicated to protection registers, once the W2PPR bit is set, the configuration of protection setting is frozen, and can only be modified by executing a temporary write unprotection operation.
14-3	-	Reserved
2-0	W2P[2:0]	Write protection Bank 2 sectors 2-0 (XFlash) These bits, if programmed at 0, disable any write access to the sectors of Bank 2 (XFlash).

Flash non-volatile write protection X register high (FNVWPXRH)

FNVWPXRH (0x0E DFB2) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														W3P1	W3P0
-														RW	RW

Table 27. FNVWPXRH register description

Bit	Bit name	Function
15-2	-	Reserved
1-0	W3P[1:0]	Write protection Bank 3/sectors 1-0 (XFlash) These bits, if programmed at 0, disable any write access to the sectors of Bank 3 (XFlash).

Flash non-volatile write protection I register low (FNVWPIRL)

FNVWPIRL (0x0E DFB4) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						W0P9	W0P8	W0P7	W0P6	W0P5	W0P4	W0P3	W0P2	W0P1	W0P0
-						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 28. FNVWPIRL register description

Bit	Bit name	Function
15-10	-	Reserved
9-0	W0P[9:0]	Write protection Bank 0/sectors 9-0 (IFlash) These bits, if programmed at 0, disable any write access to the sectors of Bank 0 (IFlash).

Flash non-volatile write protection I register high (FNVWPIRH)

FNVWPIRH (0x0E DFB6) NVR Delivery value: FFFFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														W1P1	W1P0
-														RW	RW

Table 29. FNVWPIRH register description

Bit	Bit name	Function
15-2	-	Reserved
1-0	W1P[1:0]	Write protection Bank 1/sectors 1-0 (IFlash) These bits, if programmed at 0, disable any write access to the sectors of Bank 1 (IFlash).

Flash non-volatile access protection register 0 (FNVAPR0)

Because of the ST10 architecture, the XFlash is seen as external memory. For this reason, it is impossible to access protect it from the real external memory or internal RAM.

FNVAPR0 (0x0E DFB8) NVR Delivery value: ACFFh

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

Table 30. FNVAPR0 register description

Bit	Bit name	Function
15-2	-	Reserved
1	DBGP	Debug protection This bit, if erased at 1, allows all protections to be by-passed using the debug features through the test interface. If programmed at 0, all the debug features and Flash test modes, and the test interface are disabled. STMicroelectronics will be unable to access the device to run any eventual failure analysis.
0	ACCP	Access protection This bit, if programmed at 0, disables any access (read/write) to data mapped inside the IFlash module address space, unless the current instruction is fetched from one of the two Flash modules.

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

FNVAPR1L (0x0E DFBC) NVR Delivery value: FFFFh

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

Table 31. FNVAPR1L register description

Bit	Bit name	Function
15-0	PDS[15:0]	Protections disable 15-0 If bit PDSx is programmed at 0 and bit PENx (of the FNVAPR1H register) is erased at 1, the ACCP bit action is disabled. Bit PDS0 can be programmed at 0 only if bits DBGP and ACCP (of the FNVAPR0 register) have already been programmed at 0. Bit PDSx can be programmed at 0 only if bit PENx-1 has already been programmed at 0.

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

FNVAPR1H (0x0E DFBE) NVR Delivery value: FFFFh

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

Table 32. FNVAPR1H register description

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

5.4.2 Access protection

The Flash modules have one level of access protection (access to data both when reading and writing). If bit ACCP of the FNVAPR0 register is programmed at 0, the IFlash module becomes access protected: Data in the IFlash module can be read/written only if the current execution is from the IFlash module itself.

Protection can be permanently disabled by programming bit PDS0 of the FNVAPR1H register to analyze rejects. Allowing PDS0 bit programming only when the ACCP bit is programmed, guarantees that only an execution from the Flash itself can disable the protections.

Protection can be permanently enabled again by programming bit PEN0 of the FNVAPR1L register. Access protection can be permanantly disabled and enabled again up to 16 times.

Trying to write into the access protected Flash from internal RAM is unsuccessful. Trying to read into the access protected Flash from internal RAM outputs dummy data.

When the Flash module is access protected, data access through the program erase controller (PEC) of a peripheral is forbidden. To read/write data in PEC mode from/to a protected bank, the Flash module must be temporarily unprotected.

Due to the ST10 architecture, the XFlash is seen as external memory. For this reason, it is impossible to access protect it from real external memory or internal RAM. *Table 33* summarizes the different access protection levels. In particular, it shows what is possible (and not possible) if trying to enable all access protections when fetching from a memory (see column 1).

Table 33. Summary of access protection levels

	Read IFlash/ jump to IFlash	Read XFlash/ jump to XFlash	Read Flash registers	Write Flash registers
Fetching from IFlash	Yes/Yes	Yes/Yes	Yes	Yes
Fetching from XFlash	No/Yes	Yes/Yes	Yes	No
Fetching from IRAM	No/Yes	Yes/Yes	Yes	No
Fetching from XRAM	No/Yes	Yes/Yes	Yes	No
Fetching from external memory	No/Yes	Yes/Yes	Yes	No

5.4.3 Write protection

The Flash modules have one level of write protection. Each sector of each bank of each Flash module can be software write protected by programming the related WyPx bit of the FNVWPXRH/L-FNVWPIRH/L registers at 0.

5.4.4 Temporary unprotection

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

Bit ACCP can be temporarily unprotected by executing the set protection operation and writing 1 into these bits, bu,t only if these write instructions are executed from the Flash modules.

To restore the write and access protection bits or to execute a set protection operation and write 0 into the desired bits, the microcontroller must be reset.

It is not necessary to temporarily unprotect an access protected Flash to update the code. it is sufficient to execute the updating instructions from another Flash bank.

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

5.5 Write operation examples

Examples are presented below for each kind of Flash write operation.

5.5.1 Word program

Example: 32-bit word program of data 0xAAAAAAAA at address 0x0C5554 in XFlash module.

```
FCR0H|= 0x2000;     /*Set WPG in FCR0H */
FARL = 0x5554;      /*Load Add in FARL*/
FARH = 0x000C;      /*Load Add in FARH*/
FDR0L = 0xAAAA;     /*Load Data in FDR0L*/
FDR0H = 0xAAAA;     /*Load Data in FDR0H*/
FCR0H|= 0x8000;     /*Operation start*/
```

5.5.2 Double word program

Example: Double word program (64-bit) of data 0x55AA55AA at address 0x095558 and data 0xAA55AA55 at address 0x09555C in IFlash module.

```
FCR0H|= 0x1080;     /*Set DWPG, SMOD*/
FARL = 0x5558;      /*Load Add in FARL*/
FARH = 0x0009;      /*Load Add in FARH*/
FDR0L = 0x55AA;     /*Load Data in FDR0L*/
FDR0H = 0x55AA;     /*Load Data in FDR0H*/
FDR1L = 0xAA55;     /*Load Data in FDR1L*/
FDR1H = 0xAA55;     /*Load Data in FDR1H*/
FCR0H|= 0x8000;     /*Operation start*/
```

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

5.5.3 Sector erase

Example: Sector erase of sectors B3F1 and B3F0 of Bank 3 in XFlash module.

```
FCR0H|= 0x0800;     /*Set SER in FCR0H*/
FCR1H|= 0x0003;     /*Set B3F1, B3F0*/
FCR0H|= 0x8000;     /*Operation start*/
```

5.5.4 Suspend and resume

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

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

The operation can be resumed in the following way:

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

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

5.5.5 Erase suspend, program and resume

A sector erase operation can be suspended in order to program (word or double word) another sector.

Example: Sector erase of sector B3F1 of Bank 3 in XFlash module.

```
FCR0H|= 0x0800;     /*Set SER in FCR0H*/
FCR1H|= 0x0002;     /*Set B3F1*/
FCR0H|= 0x8000;     /*Operation start*/
```

Example: Sector erase suspend

```
FCR0H|= 0x4000;     /*Set SUSP in FCR0H*/
do                  /* Loop to wait for LOCK=0 and BSY bit(s)=0 */
{tmp = FCR0L ;
} while( (tmp && 0x00E6) );
```

Example: Word program of data 0x5555AAAA at address 0x0C5554 in XFlash module.

```
FCR0H&= 0xBFFF;     /*Rst SUSP in FCR0H*/
FCR0H|= 0x2000;     /*Set WPG in FCR0H*/
FARL = 0x5554;      /*Load Add in FARL*/
FARH = 0x000C;      /*Load Add in FARH*/
FDR0L = 0xAAAA;     /*Load Data in FDR0L*/
FDR0H = 0x5555;     /*Load Data in FDR0H*/
FCR0H|= 0x8000;     /*Operation start*/
```

Once the program operation is finished, the erase operation can be resumed in the following way:

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

During the program operation in erase suspend, bits SER and SUSP are low. A word or double word program during erase suspend cannot be suspended.

To summarize:

- A sector erase can be suspended by setting SUSP bit
- To perform a word program operation during erase suspend, bits SUSP and SER must first be reset, then bits WPG and WMS can be set.
- To resume the sector erase operation bit SER must be set again
- It is forbidden to start any write operation when the SUSP bit is set

5.5.6 Set protection

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

```
FCR0H|= 0x0100;     /*Set SPR in FCR0H*/
FARL = 0xDFB4;      /*Load Add of register FNVWPIRL in FARL*/
FARH = 0x000E;      /*Load Add of register FNVWPIRL in FARH*/
FDR0L = 0xFFF0;     /*Load Data in FDR0L*/
FDR0H = 0xFFFF;     /*Load Data in FDR0H*/
FCR0H|= 0x8000;     /*Operation start*/
```

Bit SMOD of FCR0H must not be set as the write protection bits of the IFlash module are stored in the Test-Flash (XFlash module).

Example 2: Enable access and debug protection.

```
FCR0H|= 0x0100;     /*Set SPR in FCR0H*/
FARL = 0xDFB8;      /*Load Add of register FNVAPR0 in FARL*/
FARH = 0x000E;      /*Load Add of register FNVAPR0 in FARH*/
FDR0L = 0xFFFC;     /*Load Data in FDR0L*/
FCR0H|= 0x8000;     /*Operation start*/
```

Example 3: Disable access and debug protection permanently.

```
FCR0H|= 0x0100;     /*Set SPR in FCR0H*/
FARL = 0xDFBC;      /*Load Add of register FNVAPR1L in FARL*/
FARH = 0x000E;      /*Load Add of register FNVAPR1L in FARH*/
FDR0L = 0xFFFE;     /*Load Data in FDR0L for clearing PDS0*/
FCR0H|= 0x8000;     /*Operation start*/
```

Example 4: Re- enable access and debug protection permanently .

```
FCR0H|= 0x0100;     /*Set SPR in FCR0H*/
FARL = 0xDFBC;      /*Load Add register FNVAPR1H in FARL*/
FARH = 0x000E;      /*Load Add register FNVAPR1H in FARH*/
FDR0H = 0xFFFE;     /*Load Data in FDR0H for clearing PEN0*/
FCR0H|= 0x8000;     /*Operation start*/
```

Disabling and re-enabling access and debug protection permanently way (as shown above) can be done up to a maximum of 16 times.

5.6 Write operation summary

Write operations are generally started with the following three steps:

1. The first instruction is used to select the desired operation by setting its corresponding selection bit in the Flash control register 0. This instruction is also used to select in which Flash Module to apply the write operation (by setting/resetting the SMOD bit).
2. The second step is the definition of the address and data for programming or the sectors or banks to erase.
3. The third instruction is used to start the write operation, by setting the start bit, WMS, in the FCR0 register.

Once selected, but not yet started, one operation can be canceled by resetting the operation selection bit.

A summary of the available Flash module write operations are shown in *Table 34*.

Table 34. Flash write operations

Operation	Select bit	Address and data	Start bit
Word program (32-bit)	WPG	FARL/FARH FDR0L/FDR0H	WMS
Double word program (64-bit)	DWPG	FARL/FARH FDR0L/FDR0H FDR1L/FDR1H	WMS
Sector erase	SER	FCR1L/FCR1H	WMS
Set protection	SPR	FDR0L/FDR0H	WMS
Program/erase suspend	SUSP	None	None

6 The bootstrap loader

The ST10F296E implements innovative boot capabilities to:

- Support a user defined bootstrap (see ‘alternate bootstrap loader’);
- Support bootstrap via UART or bootstrap via CAN for the standard bootstrap.

6.1 Selection among user-code, standard or alternate bootstrap

The selection among user-code, standard bootstrap or alternate bootstrap is made by special combinations on Port 0L[5...4] during the time the reset configuration is latched from Port 0.

The alternate boot mode is triggered with a special combination set on Port 0L[5...4]. These signals, as with other configuration signals are latched on the rising edge of the $\overline{\text{RSTIN}}$ pin.

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

Part 1

Selection of the reset sequence according to Port 0 configuration, user mode, and alternate mode signatures:

- Decoding reset configuration P0L.5 = 1 and P0L.4 = 1 selects normal mode and selects that user Flash is mapped from address 00’0000h.
- Decoding reset configuration P0L.5 = 1 and P0L.4 = 0 selects ST10 standard bootstrap mode (Test-Flash is active and overlaps user Flash for code fetches from address 00'0000h; user Flash is active and available for read and program).
- Decoding reset configuration P0L.5 = 0 and P0L.4 = 1 activates new verifications to select which bootstrap software to execute:
  - If the user mode signature in the user Flash is programmed correctly, a software reset sequence is selected and the user code is executed.
  - if the user mode signature is not programmed correctly, but, the alternate mode signature in the user Flash is programmed correctly, alternate boot mode is selected.
  - If both the user and alternate mode signatures are not programmed correctly in the user Flash, the user key location is read again. Its value determines the behavior of the selective bootstrap loader.

Part 2

Running of user selected reset sequences:

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

Table 35. ST10F296E boot mode selection

P0.5	P0.4	ST10 decoding
1	1	User mode: User Flash is mapped at 00'0000h
1	0	Standard bootstrap loader: User Flash is mapped from 00'0000h, code fetches redirected data to Test-Flash at 00'0000h
0	1	Alternate boot mode: Flash mapping depends on signature integrity check
0	0	Reserved

6.2 Standard bootstrap loader (BSL)

The built-in bootstrap loader of the ST10F296E provides a mechanism to load the startup program, which is executed after reset, via the serial interface. In this case no external (ROM) memory or internal ROM is required for the initialization code starting at location $00'0000_H$. The bootstrap loader moves code/data into the IRAM, but it is also possible to transfer data via the serial interface into an external RAM using a second level loader routine. ROM memory (internal or external) is not necessary. However, it may be used to provide lookup tables or may provide 'core-code', a set of general purpose subroutines, for I/O operations, number crunching, system initialization, etc.

The bootstrap loader may be used to load the complete application software into ROMless systems. It may also load temporary software into complete systems for testing or calibration. in addition, it may be used to load a programming routine for Flash devices.

The BSL mechanism may be used for standard system startup as well as for special occasions such as system maintenance (firmware update), end-of-line programming, or testing.

6.2.1 Entering the standard bootstrap loader

The ST10F296E enters BSL mode if pin P0L.4 is sampled low at the end of a hardware reset. In this case the built-in bootstrap loader is activated independently of the selected bus mode. The bootstrap loader code is stored in a special Test-Flash: No part of the standard Flash memory area is required for this.

After entering BSL mode and completing the respective initialization steps, the ST10F296E scans the RxD0 line and the CAN1_RxD line to receive either a valid dominant bit from the CAN interface, or a start condition from the UART line.

Start condition on UART RxD: The ST10F296E starts the standard bootstrap loader. This bootstrap loader is identical to other ST10 devices (for example, the ST10F280). See *Section 6.3: Standard bootstrap with UART (RS232 or K-line) on page 73* for details.

Valid dominant bit on CAN1 RxD: The ST10F296E starts bootstrapping via CAN1. This bootstrapping method is new and is described in *Section 6.4: Standard bootstrap with CAN on page 78*. *Figure 6: ST10F296E new standard bootstrap loader program flow on page 69* shows the program flow of the new bootstrap loader. It illustrates how new functionalities are implemented, which is as follows:

- UART: UART has priority over CAN after a falling edge on CAN1_RxD untill the first valid rising edge on CAN1_RxD.
- CAN: Pulses on CAN1_RxD which are shorter than 20*CPU-cycles, are filtered.

6.2.2 ST10 configuration in BSL

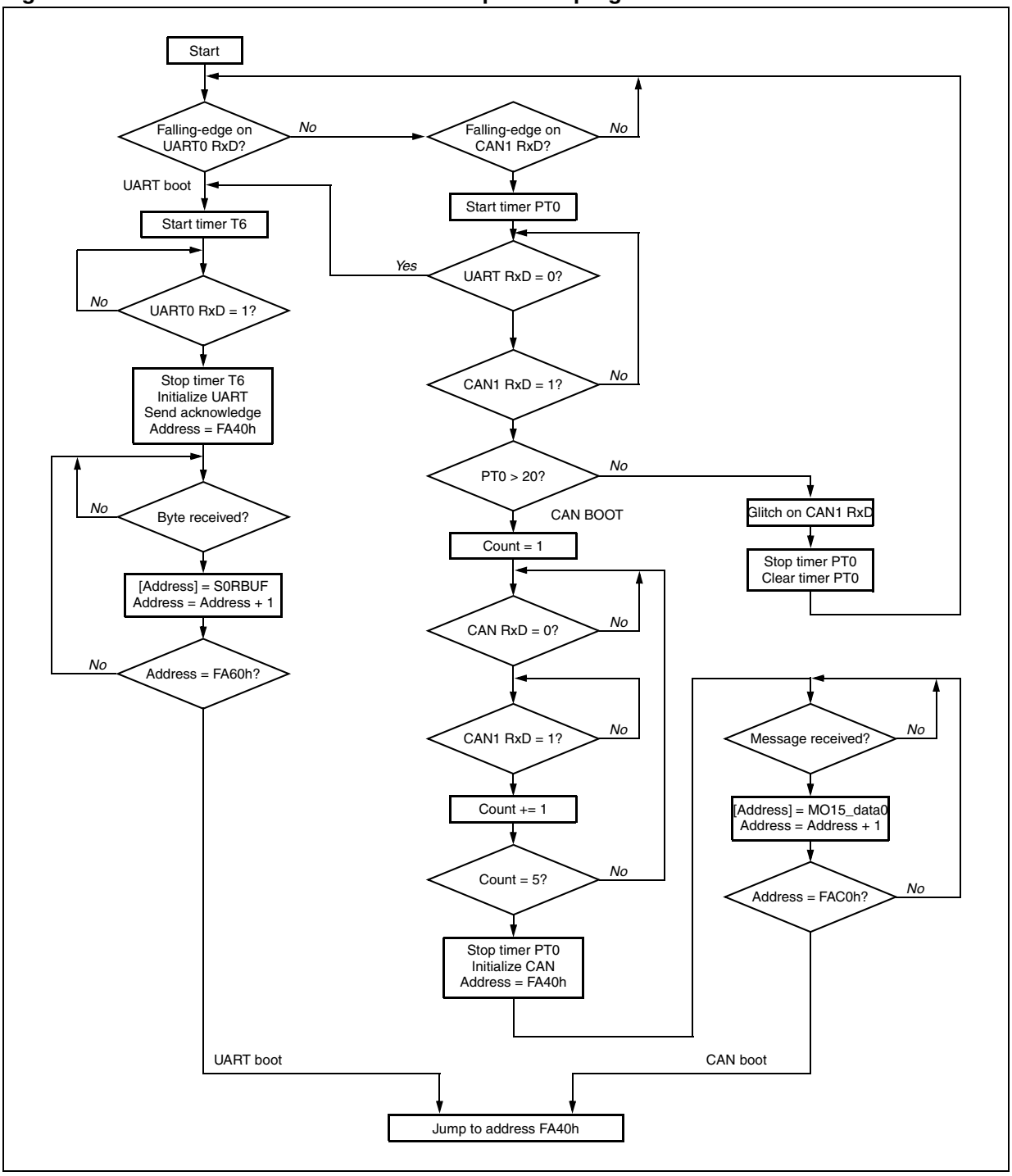
When the ST10F296E has entered BSL mode, the configuration shown in *Table 36* is automatically set (values that deviate from the normal reset values, are highlighted in ***bold italic***).

Table 36. ST10 configuration in BSL mode

Watchdog timer	***Disabled***	
Register SYSCON	***0404_H***[1]	XPEN bit set for bootstrap via CAN or alternate boot mode
Context pointer CP	$FA00_H$	
Register STKUN	$FC00_H$	
Stack pointer SP	***$FA40_H$***	
Register STKOV	$FA00_H$	
Register BUSCON0	Acc. to startup config.[2]	
Register S0CON	***8011_H***	Initialized only if bootstrap is run via UART
Register S0BG	Acc. to '00' byte	Initialized only if bootstrap is run via UART
P3.10/TXD0	***1***	Initialized only if bootstrap is run via UART
DP3.10	***1***	Initialized only if bootstrap is run via UART
CAN1 status/control register	***0000_H***	Initialized only if bootstrap is run via CAN
CAN1 bit timing register	Acc. to 0 frame	Initialized only if bootstrap is run via CAN
XPERCON	***$042D_H$***	XRAM1-2, XFlash, CAN1 and XMISC enabled. Initialized only if bootstrap is run via CAN
P4.6/CAN1_TxD	***1***	Initialized only if bootstrap is run via CAN
DP4.6	***1***	Initialized only if bootstrap is run via CAN

1. In bootstrap modes (standard or alternate) the ROMEN bit, bit 10 of the SYSCON register, is always set regardless of the $\overline{EA}$ pin level. The BYTDIS bit, bit 9 of the SYSCON register, is set according to the data bus width selection via Port 0 configuration.
2. BUSCON0 is initialized with 0000h which disables the external bus if pin $\overline{EA}$ is high during reset. If pin $\overline{EA}$ is low during reset, the BUSACT0 bit, bit 10, and the ALECTL0 bit, bit 9, are set, enabling the external bus with a lengthened ALE signal. BTYP field, bit 7 and 6, is set according to Port 0 configuration.

Figure 6. ST10F296E new standard bootstrap loader program flow



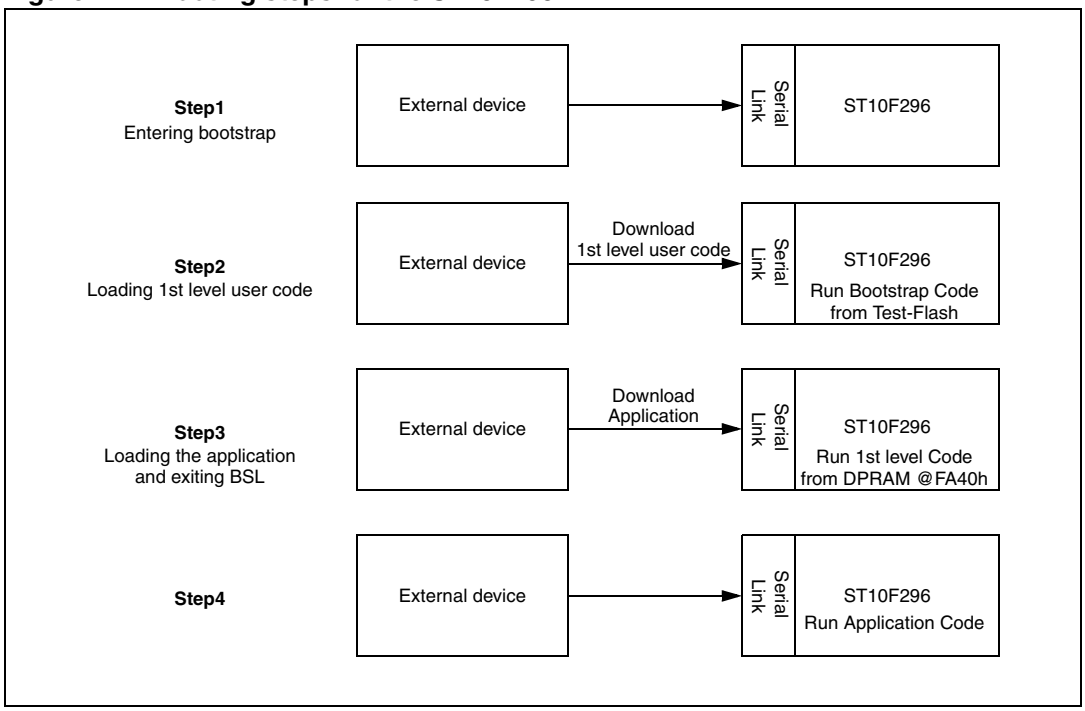
The watchdog timer is disabled, except after a normal reset, so the bootstrap loading sequence is not time limited. Depending on the selected serial link (UART0 or CAN1), pin TxD0 or CAN1_TxD is configured as output, so the ST10F296E can return the acknowledge byte. Even if the internal IFlash is enabled, no code can be executed out of it.

6.2.3 Booting steps

There are four steps to booting the ST10F296E with the boot loader code (see *Figure 7*):

1. The ST10F296E is reset with P0L.4 low
2. The internal new bootstrap code runs on the ST10 and a first level user code is downloaded from the external device, via the selected serial link (UART0 or CAN1). The bootstrap code is contained in the ST10F296E Test-Flash and is automatically run when ST10F296E is reset with P0L.4 low. After loading a preselected number of bytes, ST10F296E begins executing the downloaded program.
3. The first level user code is run on ST10F296E. Typically, this user code is another loader that is used to download the application software into the ST10F296E.
4. The loaded application software is now running

Figure 7. Booting steps for the ST10F296E

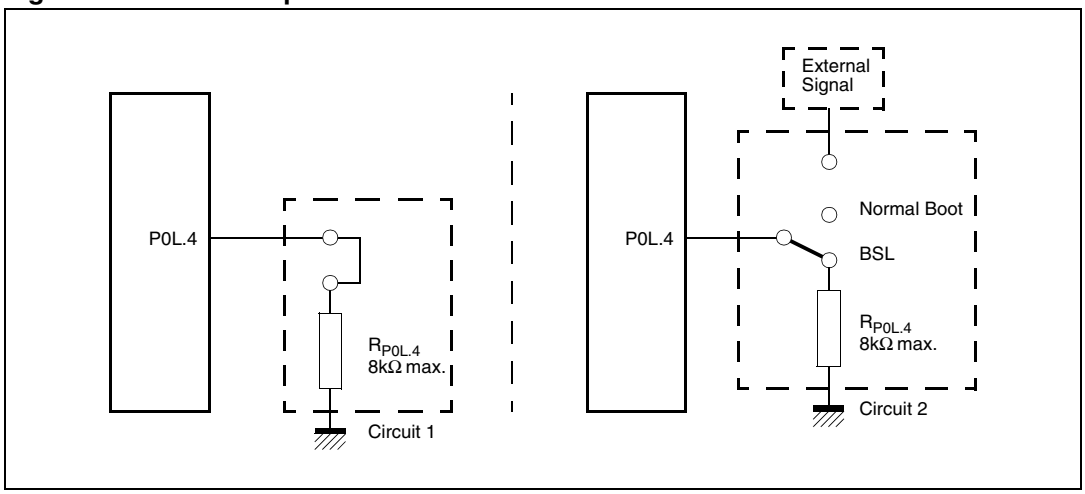


6.2.4 Hardware to activate BSL

The hardware that activates the BSL during every hardware reset may be a simple pull-down resistor on P0L.4. switchable solution (via jumper or an external signal) may be used for systems that only temporarily use the BSL.

Note: The CAN alternate function on Port 4 lines is not activated if the user has selected eight address segments (Port 4 pins have three functions: I/O port, address-segment, and CAN). Bootstrapping via CAN requires that four address segments or less are selected.

Figure 8. Hardware provisions to activate the BSL



6.2.5 Memory configuration in bootstrap loader mode

The configuration (i.e. the accessibility) of the ST10F296E's memory areas after reset in bootstrap loader mode differs from the standard case. Pin $\overline{EA}$ is evaluated when BSL mode is selected to enable the external bus or not:

- If $\overline{EA}$ = 1, the external bus is disabled (BUSACT0 = 0 in BUSCON0 register);
- If $\overline{EA}$ = 0, the external bus is enabled (BUSACT0 = 1 in BUSCON0 register).

Moreover, while in BSL mode, access to the internal IFlash area are partly redirected:

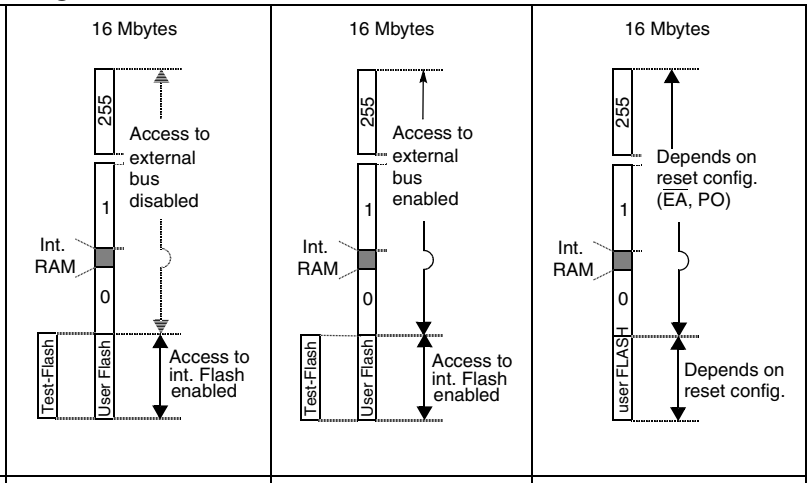
- Code access is made from the special Test-Flash seen in the range 00'0000h to 00'01FFFh.
- User IFlash is only available for read and write access (Test-Flash cannot be read nor written).
- Write access must be made with addresses starting in segment 1 from 01'0000h, whatever the value of the ROMS1 bit in the SYSCON register.
- Read access is made in segment 0 or in segment 1 depending on the ROMS1 bit value.
- In BSL mode, by default, ROMS1= 0 so the first 32 Kbytes of IFlash are mapped in segment 0.

Example

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

Figure 9 shows the memory configuration after reset.

Figure 9. Memory configuration after reset



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

1. As long as the ST10F296E is in BSL, user software should not try to execute code from the internal IFlash as the fetches are redirected to the Test-Flash.

6.2.6 Loading the startup code

After the serial link initialization sequence (see *Section 6.3* and *Section 6.4*), the BSL enters a loop to receive 32 bytes (boot via UART) or 128 bytes (boot via CAN).

These bytes are stored sequentially into the ST10F296E dual-port RAM from location 00'FA40h.

To execute the loaded code, the BSL jumps to location 00'FA40h. The bootstrap sequence running from the Test-Flash terminates. However, the microcontroller remains in BSL mode.

The initially loaded routine (the first level user code) most probably loads additional code and data. This first level user code may use the pre-initialized interface (UART or CAN) to receive data, a second level code, and store it to arbitrary user-defined locations.

This second level 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.

In all cases, the ST10F296E runs in BSL mode, i.e. with the watchdog timer disabled and with limited access to the internal IFlash area.

6.2.7 Exiting bootstrap loader mode

To execute a program in normal mode, the BSL mode must first be terminated. The ST10F296E exits BSL mode upon a software reset (level on P0L.4 is ignored) or a hardware reset (P0L.4 must be high in this case). After the reset, the ST10F296E starts executing from location $00'0000_H$ of the internal Flash (user Flash) or the external memory, as programmed via pin $\overline{EA}$.

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

6.2.8 Hardware requirements

Although the new bootstrap loader has been designed to be compatible with the old one, there are a few hardware requirements related to the new one:

- External bus configuration: Four segment address lines or less (keep CAN I/O's available) are required.
- Use of CAN pins (P4.5 and P4.6): P4.5 (CAN1_RxD) can only be used as a port input. Pin P4.6 (CAN1_TxD) can be used as input or output.
- Level on UART RxD and CAN1_RxD during the bootstrap phase (see step 2 of *Figure 7: Booting steps for the ST10F296E on page 70*): Must be 1 (external pull-up's recommended).

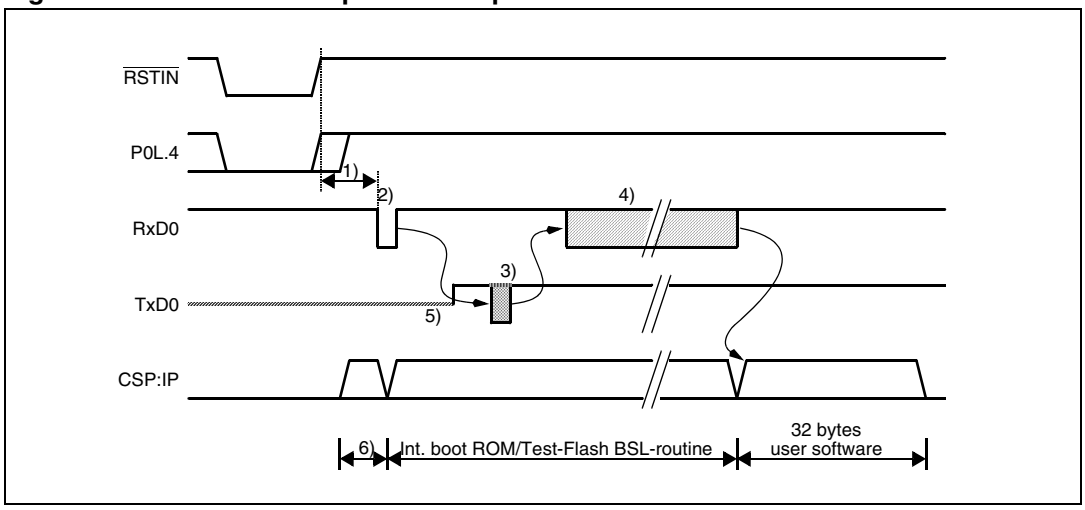
6.3 Standard bootstrap with UART (RS232 or K-line)

6.3.1 Features

ST10F296E bootstrap via UART has the same overall behavior as the old ST10 bootstrap via UART:

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

Figure 10. UART bootstrap loader sequence



1. BSL initialization time > 1ms @ f_{CPU} = 40 MHz.
2. Zero byte (1 start bit, eight 0 data bits, 1 stop bit), sent by host.
3. Acknowledge byte, sent by ST10F296E.
4. 32 bytes of code / data, sent by host.
5. TxD0 is only driven a certain time after reception of the zero byte (1.3 ms @ f_{CPU} = 40 MHz).
6. Internal boot ROM / Test-Flash.

6.3.2 Entering bootstrap via UART

The ST10F296E enters BSL mode at the end of a hardware reset if pin P0L.4 is sampled low. 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 Test-Flash, for which no part of the standard mask ROM or Flash memory area is required.

After entering BSL mode and the respective initialization, the ST10F296E scans the RxD0 line to receive a zero byte (one start bit, eight 0 data bits and one stop bit). From 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 the TxD0 pin to output. Using this baud rate, an acknowledge byte is returned to the host that provides the loaded data.

The acknowledge byte for the ST10F296E is D5h**.**

6.3.3 ST10 configuration in UART BSL (RS232 or K-line)

When the ST10F296E has entered BSL mode on the UART, the configuration shown in *Table 37* is automatically set (values that deviate from the normal reset values, are ***highlighted in bold italic***).

Table 37. ST10 configuration in UART BSL mode (RS232 or K-line)

Watchdog timer	***Disabled***	
Register SYSCON	***0400_H***[1]	
Context pointer CP	$FA00_H$	
Register STKUN	$FA00_H$	
Stack pointer SP	***$FA40_H$***	
Register STKOV	$FC00_H$	
Register BUSCON0	Acc. to startup config.[2]	
Register S0CON	***8011_H***	Initialized only if bootstrap is run via UART
Register S0BG	Acc. to 00 byte	Initialized only if Bootstrap is run via UART
P3.10/TXD0	***1***	Initialized only if Bootstrap is run via UART
DP3.10	***1***	Initialized only if Bootstrap is run via UART

1. In bootstrap modes (standard or alternate) the ROMEN bit, bit 10 of the SYSCON register, is always set regardless of the $\overline{EA}$ pin level. The BYTDIS bit, bit 9 of the SYSCON register, is set according to the data bus width selection via Port 0 configuration.
2. BUSCON0 is initialized with 0000h which disables the external bus if pin $\overline{EA}$ is high during reset. If pin $\overline{EA}$ is low during reset, the BUSACT0 bit, bit 10, and the ALECTL0 bit, bit 9, are set, enabling the external bus with a lengthened ALE signal. BTYP field, bit 7 and 6, is set according to Port 0 configuration.

The watchdog timer is disabled, except after a normal reset, so the bootstrap loading sequence is not time limited. Pin TxD0 is configured as output, so the ST10F296E can return the acknowledge byte. Even if the internal IFlash is enabled, no code can be executed out of it.

6.3.4 Loading the startup code

After sending the acknowledge byte the BSL enters a loop to receive 32 bytes via ASC0. These bytes are stored sequentially into locations $00'FA40_H$ through $00'FA5F_H$ of the IRAM. Up to 16 instructions may be placed into the RAM area. To execute the loaded code the BSL jumps to location $00'FA40_H$, i.e. the first loaded instruction. The bootstrap loading sequence then terminates, however, the ST10F296E remains in BSL mode. It is likely that the initially loaded routine loads 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. In addition, it may contain a code sequence to change the system configuration and enable the bus interface to store the received data into the external memory.

This process may go through several iterations or may directly execute the final application. In all cases, the ST10F296E runs in BSL mode, i.e. with the watchdog timer disabled and limited access to the internal Flash area. All code fetches from the internal IFlash area ($01'0000_H$...$08'FFFF_H$) are redirected to the special Test-Flash. Data read operations access the internal Flash of the ST10F296E, if any is available, but return undefined data on ROM-less devices.

6.3.5 Choosing the baud rate for the BSL via UART

The calculation of the serial baud rate for ASC0 from the length of the first zero byte that is received, allows the bootstrap loader of the ST10F296E to operate with a wide range of baud rates. However, upper and lower limits have to be respected to insure proper data transfer.

Equation 1

$$B_{ST10F296} = f_{CPU} / 32 \times (SOBRL + 1)$$

The ST10F296E uses 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. *Equation 2* below shows the association:

Equation 2

$$SOBRL = (T6 - 36) / 72$$

Where:

$$T6 = 9 / 4 \times f_{CPU} / B_{Host}$$

For correct data transfer from the host to the ST10F296E, 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 the ST10F296E baud rate can be calculated via *Equation 3*:

Equation 3

$$F_B = (B_{Contr} - B_{Host}) / B_{Contr} \times 100$$

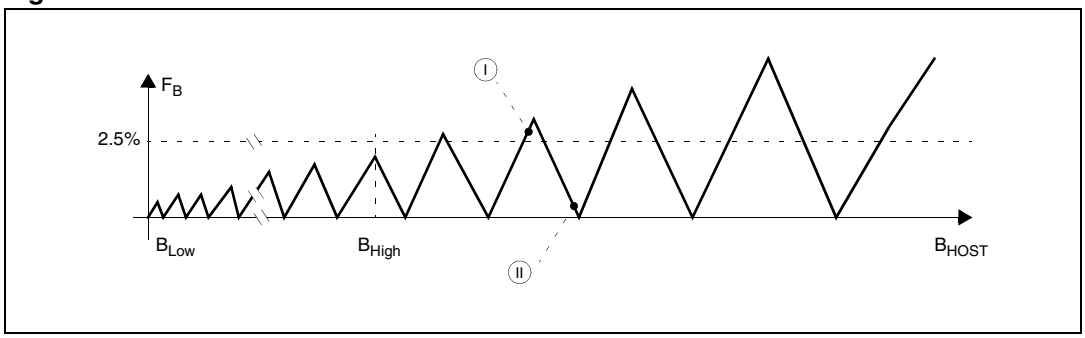
where:

$FB \leq 2.5\ \%$

Note: 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 maxima of 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 11*).

Figure 11. Baud rate deviation between the host and ST10F296E



The minimum baud rate (B_{Low} in *Figure 11*) is determined by the maximum count capacity of Timer T6, when measuring the zero byte, i.e. it depends on the CPU clock. Using the maximum T6 count as 2^{16} in the formula, the minimum baud rate can be calculated. The lowest standard baud rate in this case is 1200 Baud. Baud rates below B_{Low} cause T6 to overflow. In this case ASC0 cannot be initialized properly.

The maximum baud rate (B_{High} in *Figure 11*) is the highest baud rate where the deviation does not exceed the limit, i.e. 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. The baud rate marked 'I' in *Figure 11* may violate the deviation limit, while the higher baud rate, marked 'II', in *Figure 11* stays well below it. This depends on the host interface.

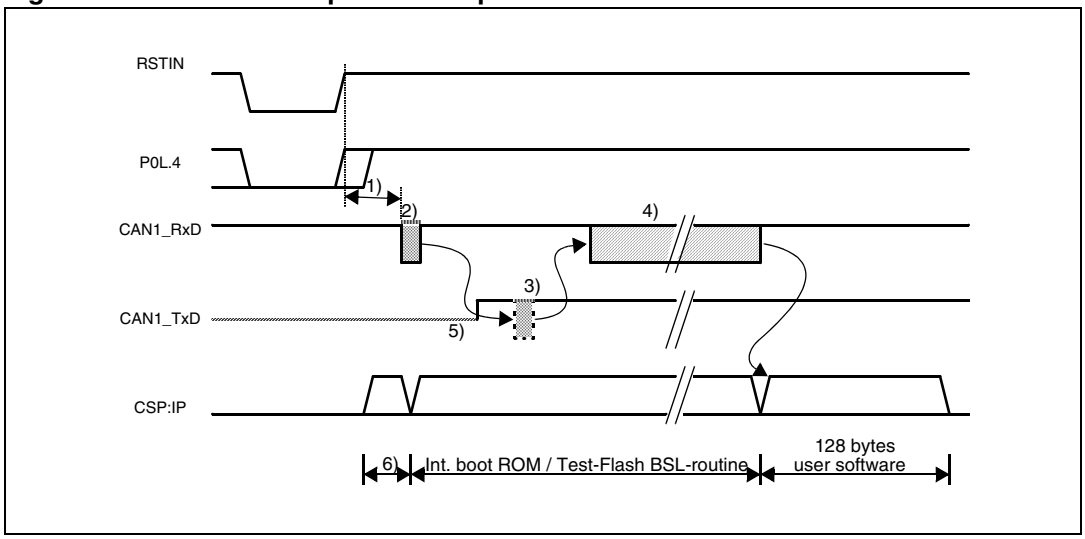
6.4 Standard bootstrap with CAN

6.4.1 Features

The bootstrap via CAN has the same overall behavior as the bootstrap via UART:

- Same bootstrapping steps
- Same bootstrap method: To analyze the timing of a predefined frame, send back an acknowledge frame (on request only), load a fixed number of bytes and then run.
- Same functionalities: To boot with different crystals and PLL ratios.

Figure 12. CAN bootstrap loader sequence



1. BSL initialization time > 1ms @ f_{CPU} = 40 MHz
2. Zero frame (CAN message: standard ID = 0, DLC = 0) sent by host
3. CAN message (standard ID = E6h, DLC = 3, Data0 = D5h, Data1-Data2 = IDCHIP_low-high) sent by ST10F296E on request.
4. 128 bytes of code/data, sent by host
5. CAN1_TxD is only driven a certain time after reception of the zero byte (1.3 ms @ f_{CPU} = 40 MHz).
6. Internal boot ROM/Test-Flash

The bootstrap loader may be used to load the complete application software into ROM-less systems. It may also load temporary software into complete systems for testing or calibration. In addition, it may be used to load a programming routine for Flash devices.

The BSL mechanism may be used for standard system start-ups as well as for special occasions like system maintenance (firmware update), end-of-line programming or testing.

6.4.2 Entering the CAN bootstrap loader

The ST10F296E enters BSL mode, if pin P0L.4 is sampled low at the end of a hardware reset. In this case, the built-in bootstrap loader is activated independent of the selected bus mode. The bootstrap loader code is stored in a special Test-Flash, no part of the standard mask ROM or Flash memory area is required for this.

After entering BSL mode and the respective initialization the ST10F296E scans the CAN1_TxD line to receive the following initialization frame:

- Standard identifier = 0h
- DLC = 0h

As all the bits to be transmitted are dominant bits, a succession of five dominant bits and one stuff bit on the CAN network is used. From the duration of this frame it calculates the corresponding baud rate factor with respect to the current CPU clock, initializes the CAN1 interface accordingly, switches pin CAN1_TxD to output and enables the CAN1 interface to take part in the network communication. Using this baud rate, a message object is configured to send an acknowledge frame. The ST10F296E does send this message object, but, the host can request it by sending remote frame.

The acknowledge frame is the following for the ST10F296E:

- Standard identifier = E6h
- DLC = 3h
- Data0 = D5h (generic acknowledge of the ST10 devices)
- Data1 = IDCHIP least significant byte
- Data2 = IDCHIP most significant byte

For the ST10F296E, IDCHIP = 128Xh.

Note: Two behaviors can be distinguished regarding acknowledgement of the ST10 by the host. If the host is behaving according to CAN protocol, as long as the ST10 CAN module is not configured, the host is alone on the CAN network and does not receive acknowledgement. It automatically resends the zero frame. As soon as the ST10 CAN is configured, the host acknowledges the zero frame. The 'acknowledge frame', with identifier 0xE6, is configured, but, the transmit request is not set. The host can request this frame to be sent, and therefore get the IDCHIP, by sending a remote frame.

As the IDCHIP is sent in the acknowledge frame, Flash programming software now has the possibility to know immediately the exact type of device to be programmed.

6.4.3 ST10 configuration in CAN BSL

When the ST10F296E has entered BSL mode via CAN, the configuration shown in *Table 38* is automatically set (values that deviate from the normal reset values, are marked in ***bold italic***)

Table 38. ST10 configuration in CAN BSL mode

Watchdog timer	Disabled	
Register SYSCON	***0404_H***[1]	XPEN bit set
Context pointer CP	$FA00_H$	
Register STKUN	$FA00_H$	
Stack pointer SP	***$FA40_H$***	
Register STKOV	$FC00_H$	
Register BUSCON0	Acc. to startup config.[2]	
CAN1 status/control register	0000_H	Initialized only if bootstrap is run via UART
CAN1 bit timing register	Acc. to 0 frame	Initialized only if bootstrap is run via CAN
XPERCON	***$042D_H$***	XRAM1-2, XFlash, CAN1 and XMISC enabled
P4.6/CAN1_TxD	***1***	Initialized only if bootstrap is run via CAN
DP4.6	***1***	Initialized only if bootstrap is run via CAN

1. In bootstrap modes (standard or alternate) the ROMEN bit, bit 10 of the SYSCON register, is always set regardless of the $\overline{EA}$ pin level. The BYTDIS bit, bit 9 of the SYSCON register, is set according to the data bus width selection via Port 0 configuration.
2. BUSCON0 is initialized with 0000h which disables the external bus if pin $\overline{EA}$ is high during reset. If pin $\overline{EA}$ is low during reset, the BUSACT0 bit, bit 10, and the ALECTL0 bit, bit 9, are set, enabling the external bus with a lengthened ALE signal. BTYP field, bit 7 and 6, is set according to Port 0 configuration.

The watchdog timer is disabled, except after a normal reset, so the bootstrap loading sequence is not time limited. The CAN1_TxD1 pin is configured as output, so the ST10F296E can return the identification frame. Even if the internal IFlash is enabled, no code can be executed out of it.

6.4.4 Loading the startup code via CAN

After sending the acknowledge byte the BSL enters a loop to receive 128 bytes via CAN1.

Note: The number of bytes loaded when booting via the CAN interface has been extended to 128 bytes to allow re-configuration of the CAN bit timing register with the best timings (synchronization window, ...). This can be achieved by the following sequence of instructions:

```
ReconfigureBaudRate:
   MOV R1,#041h
   MOV DPP3:0EF00h,R1 ; Put CAN in Init, enable Configuration Change
   MOV R1,#01600h
   MOV DPP3:0EF06h,R1 ; 1MBaud at Fcpu = 20 MHz
```

These 128 bytes are stored sequentially into locations 00'FA40$_H$ to 00'FABF$_H$ of the IRAM. So, up to 64 instructions may be placed into the RAM area. To execute the loaded code the BSL jumps to location 00'FA40$_H$, the first loaded instruction. The bootstrap loading sequence is now terminated, however, the ST10F296E remains in BSL mode. It is likely that the initially loaded routine loads additional code or data (because an average application is likely to require substantially more than 64 instructions). This second receive loop may directly use the pre-initialized CAN interface to receive data and store it to arbitrary user-defined locations.

The 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. In addition, it may contain a code sequence to change the system configuration and enable the bus interface to store the received data into the external memory.

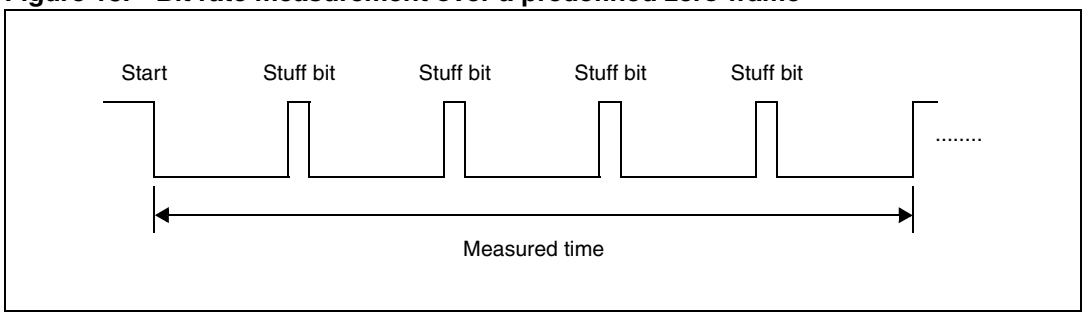
This process may go through several iterations or may directly execute the final application. In all cases the ST10F296E runs in BSL mode, with the watchdog timer disabled and limited access to the internal Flash area. All code fetches from the internal Flash area (01'0000$_H$ ...08'FFFF$_H$) are redirected to the special Test-Flash. Data read operations access the internal Flash of the ST10F296E, if any is available, but return undefined data on ROM-less devices.

6.4.5 Choosing the baud rate for the BSL via CAN

The bootstrap via CAN acts in the same way as the UART bootstrap mode. When the ST10F296E is started in BSL mode, it polls the RxD0 and CAN1_RxD lines. When polling a low level on one of these lines, a timer is launched that is stopped when the line goes back to high level.

For CAN communication, the algorithm is made to receive a zero frame, where the standard identifier is 0x0 and DLC is 0. This frame produces the following levels on the network: 5D, 1R, 5D, 1R, 5D, 1R, 5D, 1R, 5D, 1R, 4D, 1R, 1D, 11R. The algorithm lets the timer run until detection of the 5th recessive bit. In this way, the bit timing is calculated over 29 bit time durations. This minimizes the error introduced by the polling.

Figure 13. Bit rate measurement over a predefined zero-frame



Error induced by the polling

The code used for polling is as follows:

```
WaitCom:
    JNB   P4.5,CAN_Boot           ; if SOF detected on CAN, then go to CAN
                                  ; loader
    JB    P3.11,WaitCom           ; Wait for start bit at RxD0
    BSET  T6R                     ; Start Timer T6
....
CAN_Boot:
    BSET  PWMCON0.0               ; Start PWM Timer0
                                  ; (resolution is 1 CPU clk cycle)
    JMPR  cc_UC,WaitRecessiveBit
WaitDominantBit:
    JB    P4.5,WaitDominantBit    ; wait for end of stuff bit
WaitRecessiveBit:
    JNB   P4.5,WaitRecessiveBit   ; wait for 1st dominant bit = Stuff bit
    CMPI1 R1,#5                   ; Test if 5th stuff bit detected
    JMPR  cc_NE,WaitDominantBit   ; No, go back to count more
    BCLR  PWMCON.0                ; Stop timer
                                  ; here the 5th stuff bit is detected:
                                  ; PT0 = 29 Bit_Time (25D and 4R)
```

The maximum error at detection of communication on the CAN pin is: (1 not taken + 1 taken jumps) + 1 taken jump + 1 bit set: (6) + 6 CPU clock cycles

The error at detection of the 5th recessive bit is: (1 taken jump) + 1 not taken jump + 1 compare + 1 bit clear: (4) + 6 CPU cycles

In the worst case scenario, the induced error is 6 CPU clock cycles. So, polling could induce an error of 6 timer ticks.

Error induced by the baud rate calculation

The content of the PT0 timer counter corresponds to 29 bit times. This gives the following equation:

Equation 4

$$PT0 = 58 \times (BRP + 1) \times (1 + Tseg1 + Tseg2)$$

where BRP (bit rate prescaler), Tseg1 and Tseg2 are the field of the CAN bit timing register.

The CAN protocol specification recommends implementing a bit time composed of at least eight time quantum (tq). This recommendation has been applied above. The maximum bit time length is 25 tq. To achieve good precision, the target must have the smallest BRP and the maximum number of tq in a bit time.

The ranges for PT0 according to BRP are given in *Equation 5*.

Equation 5

$$8 \leq 1 + Tseg1 + Tseg2 \leq 25$$

$$464 \times (1 + BRP) \leq PT0 \leq 1450 \times (1 + BRP)$$

Table 39. Timer content ranges of BRP value in *Equation 5*

BRP	PT0_min	PT0_max	Comments
0	464	1450	
1	1451	2900	
2	2901	4350	
3	4351	5800	
4	5801	7250	
5	7251	8700	
..	..	..	
43	20416	63800	
44	20880	65250	
45	21344	66700	Possible timer overflow
..	..	..	
63	X	X	

The error coming from the measurement of bit 29 is:

$$e_1 = 6 / [PTO]$$

It is maximal for the smallest BRP value and the smallest number of ticks in PT0.

Therefore:

$$e_{1\ Max} = 1.29\%$$

For the best precision possible, the target must have the smallest BRP, which minimises errors when calculating time quanta in a bit time.

To achieve this, the PT0 value is divided into ranges of 1450 ticks. In the bootstrap algorithm, PT0 is divided by 1451 and the result gives the BRP value.3

This calculated BRP value is then divided into PT0 to give the *'1+ Tseg1 + Tseg2' value*. A table is then made to set the values for *Tseg1* and *Tseg2* according to the *'1+ Tseg1 + Tseg2' value*. The *Tseg1* and *Tseg2* values are chosen to reach a sample point between 70% and 80% of the bit time.

During the calculation of *'1+ Tseg1 + Tseg2'*, an error, e_2, can be introduced. The maximum value of this error is 1 time quantum.

To compensate for any possible errors on the bit rate, the (re)synchronization jump width is fixed to two time quanta.

6.4.6 How to compute the baud rate error

An example of the baud rate error computation is as follows:

Conditions:

- CPU frequency: 20 MHz
- Target bit rate: 1 Mbit/s

The content of the PTO timer for bit 29 is given in *Equation 6*:

Equation 6

$$[PT0] = 29 \times f_{CPU} / (BitRate) = 29 \times 20 \times 6 / 1 \times 10^6 = 580$$

Therefore:

574 < [PT0] < 586

This gives:

- BRP = 0
- tq = 100 ns

Computation of *1 + Tseg1 + Tseg2* considering *Equation 4* is given in *Equation 7*:

Equation 7

$$9 = \frac{574}{58} \leq Tseg1 + Tseg2 \leq \frac{586}{58} = 10$$

In the algorithm, a rounding to the superior value is made if the remainder of the division is greater than half of the divisor. This would have been the case above, if the PT0 content was 574. Thus in this example, *1+Tseg1+Tseg2 = 10*, giving a bit time of exactly 1µs => no error in bit rate.

Note: In most cases (24 MHz, 32 MHz, and 40 MHz of CPU frequency and 125, 250, 500 or 1Mbyte/s of bit rate) there is no error. However, it is better to check the error with real application parameters.

The content of the bit timing register is : *0x1640*. This gives a sample point of 80%.

Note: The (re)synchronization jump width is fixed to 2 time quanta.

6.4.7 Bootstrap via CAN

After the bootstrap phase, the ST10F296E CAN module is configured as follows:

- Pin P4.6 is configured as output (the latch value is: 1 = recessive) to assume CAN1_TxD function.
- The MO2 is configured to output the acknowledge of the bootstrap with the standard identifier E6h, DLC = 3, Data0 = D5h, and Data1&2 = IDCHIP.
- The MO1 is configured to receive messages with the standard identifier 5h. Its acceptance mask is set in order that all bits must match. The DLC received is not checked: The ST10 expects only 1 byte of data at a time.

No other message is sent by the ST10F296E after the acknowledge.

Note: The CAN bootstrap loader waits for 128 bytes of data instead of 32 bytes (see Section 6.3: Standard bootstrap with UART (RS232 or K-line) on page 73). This is to allow the user to reconfigure the CAN bit rate as soon as possible.

6.5 Comparing the old and the new bootstrap loader

Table 40 and *Table 41* summarize the differences between bootstrapping via UART only (old ST10 method) and bootstrapping via UART or CAN (new ST10F296E method).

Table 40. Software topics summary

Old bootstrap loader	New bootstrap loader	Comments
Uses only 32 bytes in dual-port RAM from 00'FA40h	Uses up to 128 bytes in dual-port RAM from 00'FA40h	For compatibility between bootstrapping via UART and bootstrapping via CAN1, avoid loading the application software in the 00'FA60h/00'FABFh range
Loads 32 bytes from the UART	Loads 32 bytes from UART (bootstrapping via UART mode)	Same files can be used for bootstrapping via UART
User selected XPeripherals can be enabled during bootstrapping (see steps 3 and 4 of *Section 6.2.3: Booting steps on page 70*)	XPeripherals selection is fixed.	User can change the XPeripheral selections through a specific code

6.5.1 Software aspects

As CAN1 is needed, the XPERCON register is configured by the bootstrap loader code and the XPEN bit of the SYSCON register is set. This is done as follows:

- Disable the XPeripherals by clearing the XPEN bit in the SYSCON register. ***Caution: This part of code must not be located in the XRAM, because if so, it is disabled.***
- Enable the XPeripherals that are needed by writing the correct value in the XPERCON register.
- Set the XPEN bit in the SYSCON.

Note: The settings can be modified if the EINIT instruction is not executed (and is not in the bootstrap loader code).

6.5.2 Hardware aspects

The new bootstrap loading method via UART and CAN is compatible with the old method via UART only. However, some additional hardware is required with the new method which is summarized in *Table 41*.

Table 41. Hardware topics summary

Actual bootstrap loader	New bootstrap loader	Comments
P4.5 can be used as output in BSL mode	P4.5 cannot be used as user output in BSL mode. It can only be used as CAN1_RxD, input, or address-segments.	
The level on CAN1_RxD can change during step 2 of the booting steps (see *Section 6.2.3 on page 70*)	The level on CAN1_RxD must be stable at 1 during step 2 of the booting steps (see *Section 6.2.3 on page 70*)	External pull-up on P4.5 needed

6.6 Alternate boot mode (ABM)

6.6.1 Activation

Alternate boot mode is activated with the combination 01 on Port 0L[5..4] at the rising edge of $\overline{\text{RSTIN}}$.

6.6.2 Memory mapping

ST10F296E has the same memory mapping for standard and alternate boot mode:

- Test-Flash: Mapped from 00'0000h. The standard bootstrap loader can be started by executing a jump to the address of this routine (JMPS 00'xxxx; address to be defined).
- User Flash: The user Flash is divided into two parts: The IFlash, visible only for memory reads and memory writes (no code fetch) and the XFlash, visible for any ST10 access (memory read, memory write, code fetch).
- All ST10F296E XRAM and XPeripheral modules can be accessed if enabled in the XPERCON register.

Note: The alternate boot mode can be used to reprogram the whole content of ST10F296E user Flash (except Block 0 in Bank 2).

6.6.3 Interrupts

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

As a consequence, interrupts are not allowed in alternate boot mode. All maskable and non maskable interrupts must be disabled.

6.6.4 ST10 configuration in alternate boot mode

When the ST10F296E has entered BSL mode via CAN, the configuration shown in *Table 42* is automatically set (values that deviate from the normal reset values, are marked in ***bold italic***).

Table 42. ST10 configuration in alternate boot mode

Watchdog timer	***Disabled***	
Register SYSCON	***0404_H***(1)	XPEN bit set
Context pointer CP	$FA00_H$	
Register STKUN	$FA00_H$	
Stack pointer SP	***$FA40_H$***	
Register STKOV	$FC00_H$	
Register BUSCON0	Acc. to startup config.(2)	
XPERCON	***$002D_H$***	XRAM1-2, XFlash, CAN1 enabled

1. In bootstrap modes (standard or alternate) the ROMEN bit, bit 10 of the SYSCON register, is always set regardless of the $\overline{EA}$ pin level. The BYTDIS bit, bit 9 of the SYSCON register, is set according to the data bus width selection via Port 0 configuration.
2. BUSCON0 is initialized with 0000h which disables the external bus if pin $\overline{EA}$ is high during reset. If pin $\overline{EA}$ is low during reset, the BUSACT0 bit, bit 10, and the ALECTL0 bit, bit 9, are set, enabling the external bus with a lengthened ALE signal. BTYP field, bit 7 and 6, is set according to Port 0 configuration.

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

Warning: As the XFlash is needed, the XPERCON register is configured by the ABM loader code and the XPEN bit of the SYSCON register

To do this:

- Copy a function into DPRAM that can do the following:
  - Disable the XPeripherals by clearing the XPEN bit in the SYSCON register
  - Enable the XPeripherals that are needed by writing the correct value in the XPERCON register
  - Set the XPEN bit in the SYSCON register
  - Return to the calling address
- Call the function from XFlash

Changing the XPERCON value can not be executed from the XFlash because the XFlash is disabled when the XPEN bit in the SYSCON register is cleared.

Note: The settings can be modified if the EINIT instruction is not executed (and is not in the bootstrap loader code).

6.6.5 Watchdog

The watchdog timer remains disabled during both standard and alternate boot mode. If a watchdog reset occurs, a software reset is generated.

Note: See note concerning software reset in Section 6.2.7 on page 73.

6.6.6 Exiting alternate boot mode

Once the ABM mode is entered, it can be exited only with a software or hardware reset.

Note: See note concerning software reset in Section 6.2.7 on page 73.

6.6.7 Alternate boot user software

Users can write the software they want to execute in alternate boot user mode if the rules concerning the following items are met:

- Mapping variables
- Disabling interrupts
- Exiting conditions
- Predefining vectors in Block 0 of Bank 2
- Using the watchdog

The starting address is 09’0000h.

6.6.8 User/alternate boot mode signature check

To operate user/alternate boot mode, the signature of two memory location contents are calculated and compared to a reference signature. Flash memory locations must be reserved and programmed as follows:

User mode signature

00'0000h: Memory address of *operand0* for the signature computing

00’1FFCh: Memory address of *operand1* for the signature computing

00’1FFEh: Memory address for the signature reference

Alternate mode signature

09'0000h: Memory address of *operand0* for the signature computing

09’1FFCh: Memory address of *operand1* for the signature computing

09’1FFEh: Memory address for the signature reference

Correct values for *operand0, operand1* and the reference signature allow the sequence in *Figure 14* to execute successfully.

Figure 14. Reference signature computation

```
MOV     Rx, CheckBlock1Addr          ; 00’0000h for standard reset
ADD     Rx, CheckBlock2Addr          ; 00’1FFCh for standard reset
CPLB    RLx                          ; 1s complement of the lower
                                     ;byte of the sum
CMP     Rx, CheckBlock3Addr          ; 00’1FFEh for standard reset
```

6.6.9 Alternate boot user software aspects

User defined alternate boot code must start at 09'0000h. A new SFR has been created on ST10F296E to indicate that the device is running in alternate boot mode. Bit 5 of the EMUCON register (mapped at 0xFE0Ah) is set when the alternate boot is selected by the reset configuration. All other bits must be ignored when checking the content of this register to read the value of bit5.

This bit is a read-only bit. It remains set until the next software or hardware reset.

EMUCON register

EMUCON (FE0Ah/05h) SFR Reset value: xxh

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

Table 43. EMUCON register description

Bit	Bit name	Function
15-6	-	Reserved
5	ABM	ABM Flag (or TMOD3) 0: Alternate boot mode is not selected by reset configuration on P0L[5..4] 1: Alternate boot mode is selected by reset configuration on P0L[5..4]. This bit is set if P0L[5..4] = 01 during hardware reset.
4-0	-	Reserved

6.6.10 Internal decoding of test modes

The test mode decoding logic is located inside the ST10F296E bus controller.

The decoding is as follows:

- Alternate boot mode decoding: ($\overline{\text{P0L.5}}$ & P0L.4)
- Standard bootstrap decoding: (P0L.5 & $\overline{\text{P0L.4}}$)
- Normal operation: (P0L.5 & P0L.4)

The other configurations select ST internal test modes.

6.6.11 Example of alternate boot mode operation

- The reset configuration is latched on the rising edge of the $\overline{\text{RSTIN}}$ pin.
- If bootstrap loader mode is not enabled (P0L[5..4] = 11), ST10F296E hardware starts a standard hardware reset procedure.
- If standard bootstrap loader is enabled (P0L[5..4] = 10), the standard ST10 bootstrap loader is enabled and a variable is cleared to indicate that ABM is not enabled.
- If alternate boot mode is selected (P0L[5..4] = 01), a predefined reset sequence may be activated. This depends on the user/alternate boot mode signature check.

6.7 Selective boot mode

Selective boot mode is a sub-case of alternate boot mode.

The following additional check is made when no signature of the alternate boot mode signature check is correct:

Address 00'1FFCh is read again.

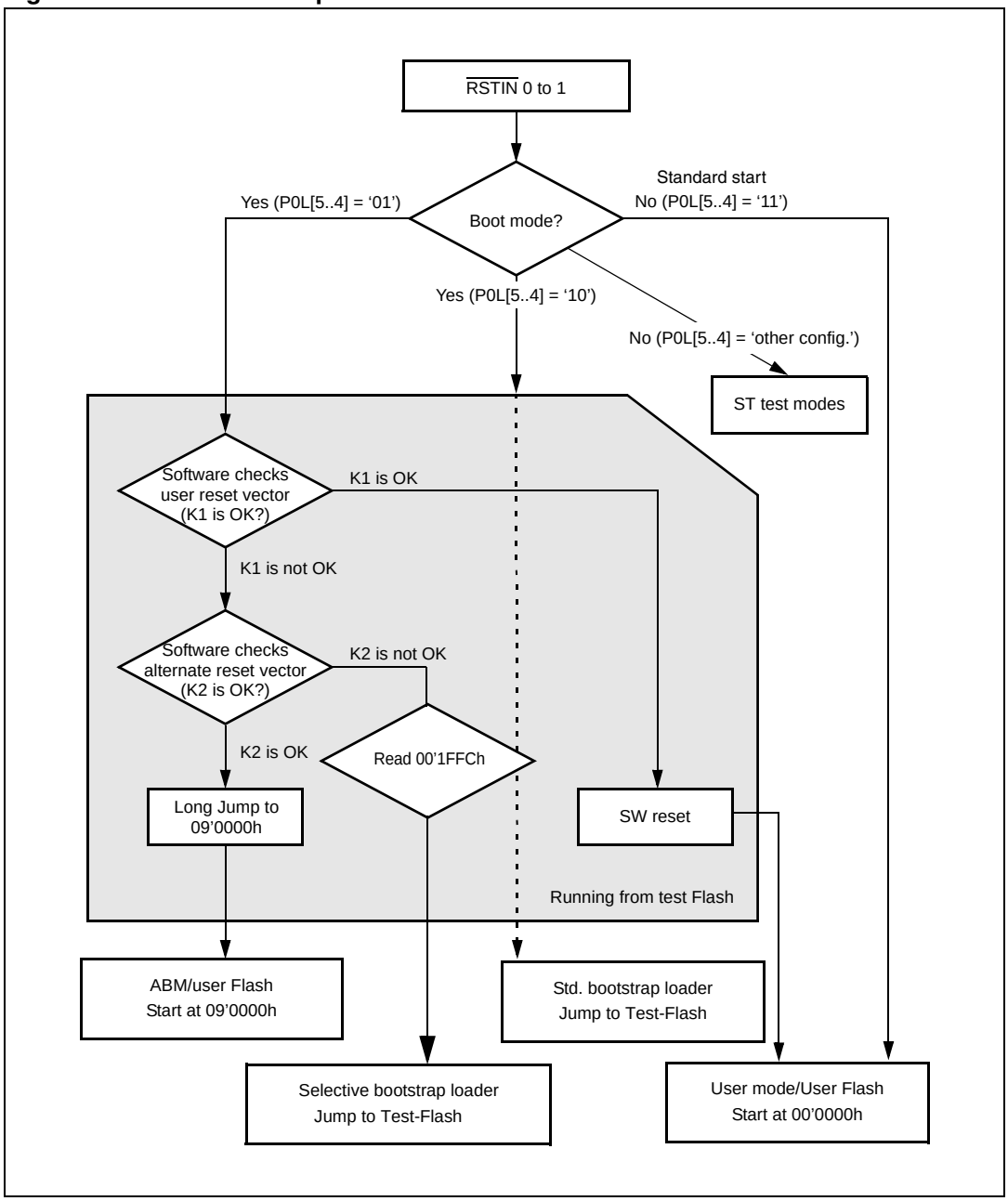
- If a value 0000h or FFFFh is obtained, a jump is performed to the standard bootstrap loader.
- If the value obtained is not 0000h or FFFFh:
  - High byte bits are disregarded
  - Low byte bits select which communication channel is enabled (see *Table 44*).

Table 44. Selective boot mode configurations

Bit	Function
0	UART selection 0: UART not watched for a start condition 1: UART is watched for a start condition
1	CAN1 selection 0: CAN1 not watched for a start condition 1: CAN1 is watched for a start condition
2-7	Reserved Must be programmed to 0 for upward compatibility

- 0xXX03 configures the selective bootstrap loader to poll for RxD0 and CAN1_RxD.
- 0xXX01 configures the selective bootstrap loader to poll RxD0 only (no bootloading via CAN).
- 0xXX02 configures the selective bootstrap loader to poll CAN1_RxD only (no bootloading via UART).
- other values will let the ST10F296E executing an endless loop into the Test-Flash.

Figure 15. Reset boot sequence



7 Central processing unit (CPU)

The CPU includes a four-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 instructions of the ST10F296E can be executed in one instruction cycle which requires 31.25 ns at 25 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 two cycles, 16 x 16-bit multiplication in five 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 two cycles to one cycle.

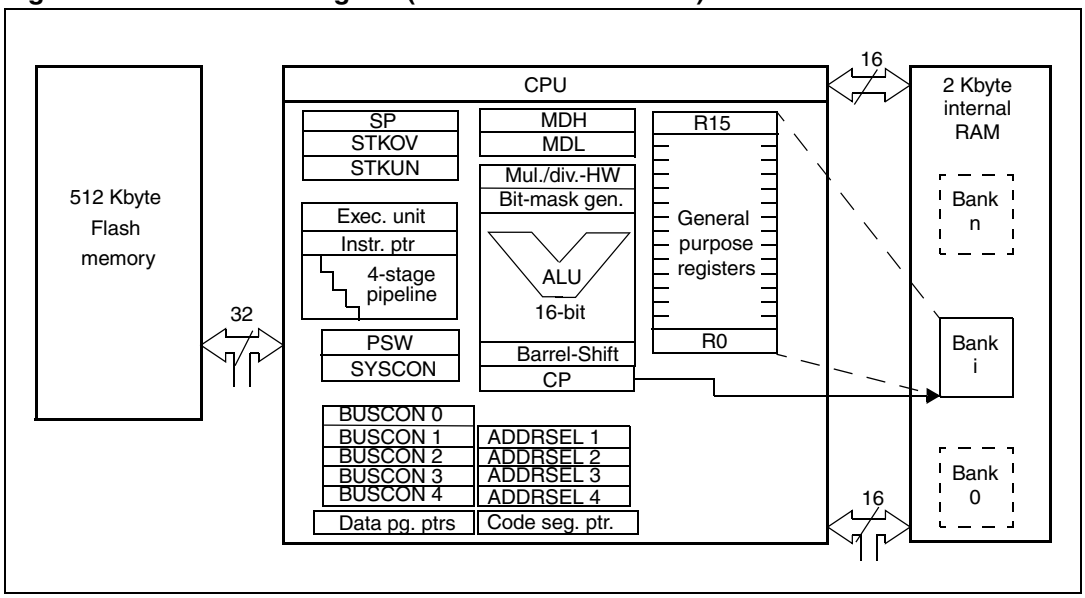
The CPU uses a bank of 16 word registers to run the current context. This bank of general purpose registers (GPR) is physically stored within the on-chip internal RAM (IRAM) area. A context pointer (CP) register determines the base address of the active register bank to be accessed by the CPU.

The number of register banks is 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 16. CPU block diagram (MAC unit not included)



7.1 Multiplier-accumulator unit (MAC)

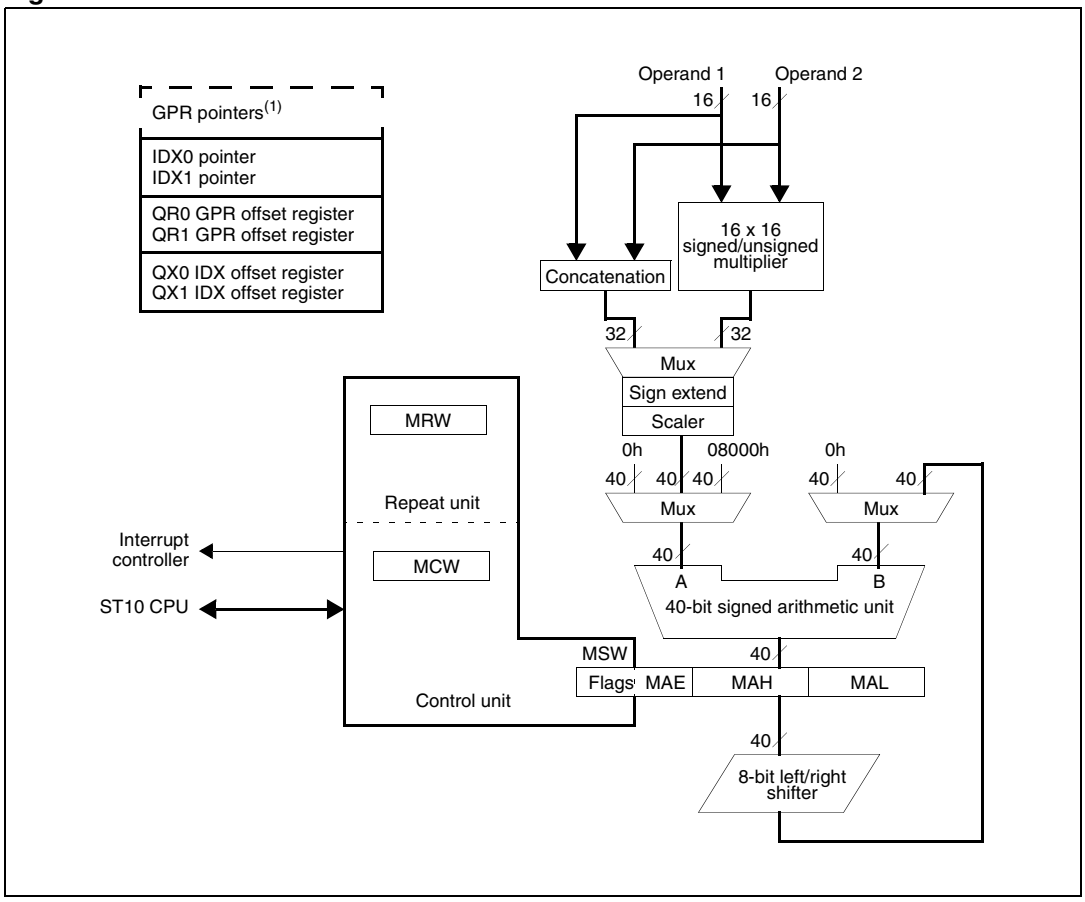
The specialized MAC coprocessor has been added to the ST10 CPU core to improve the computing performances of the ST10 device during signal processing tasks.

The standard ST10 CPU has been modified to include new addressing capabilities which enable the CPU to supply the new coprocessor with up to two operands per instruction cycle.

This new MAC coprocessor contains a fast multiply-accumulate unit and a repeat unit.

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

Figure 17. MAC unit architecture



1. Shared with standard ALU

7.2 Instruction set summary

Table 45 lists the instructions of the ST10F296E. A detailed description of each instruction can be found in the ST10 family programming manual (PM0036).

Table 45. Instruction set summary

Mnemonic	Description	Bytes
ADD(B)	Add word (byte) operands	2/4
ADDC(B)	Add word (byte) operands with carry	2/4
SUB(B)	Subtract word (byte) operands	2/4
SUBC(B)	Subtract word (byte) operands with carry	2/4
MUL(U)	(Un)Signed multiply direct GPR by direct GPR (16-16-bit)	2
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	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

Table 45. Instruction set summary (continued)

Mnemonic	Description	Bytes
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
RETP	Return from intra-segment subroutine and pop direct word register from system stack	2
RETI	Return from interrupt service subroutine	2
SRST	Software reset	4
IDLE	Enter idle mode	4
PWRDN	Enter power-down mode (supposes $\overline{\text{NMI}}$-pin being low)	4
SRVWDT	Service watchdog timer	4
DISWDT	Disable watchdog timer	4
EINIT	Signify end-of-initialization on RSTOUT pin	4
ATOMIC	Begin ATOMIC sequence	2
EXTR	Begin EXTended register sequence	2
EXTP(R)	Begin EXTended page (and register) sequence	2/4
EXTS(R)	Begin EXTended segment (and register) sequence	2/4
NOP	Null operation	2

7.3 MAC coprocessor specific instructions

Table 46 lists the MAC instructions of the ST10F296E. A detailed description of each instruction can be found in the ST10 family programming manual (PM0036). Note that all MAC instructions are encoded on four bytes.

Table 46. MAC instruction set summary

Mnemonic	Description
CoABS	Absolute value of the accumulator
CoADD(2)	Addition
CoASHR(rnd)	Accumulator arithmetic shift right and optional round
CoCMP	Compare accumulator with operands
CoLOAD(-,2)	Load accumulator with operands
CoMAC(R,u,s,-,rnd)	(Un)signed/(un)signed multiply-accumulate and optional round
CoMACM(R)(u,s,-,rnd)	(Un)signed/(un)signed multiply-accumulate with parallel data move and optional round
CoMAX/CoMIN	Maximum/minimum of operands and accumulator
CoMOV	Memory to memory move
CoMUL(u,s,-,rnd)	(Un)signed/(un)signed multiply and optional round
CoNEG(rnd)	Negate accumulator and optional round
CoNOP	No-operation
CoRND	Round accumulator
CoSHL/CoSHR	Accumulator logical shift left/right
CoSTORE	Store a MAC unit register
CoSUB(2,R)	Substraction

8 External bus controller (EBC)

All external memory access is 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 Port 1 and data is input/output on Port 0 or P0L, respectively. In the multiplexed bus modes both addresses and data use Port 0 for input/output.

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

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

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

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

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

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

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

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

The active level of the READY pin can be set by the RDYPOL bit 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 the RDYPOL bit in the associated BUSCON register.

8.1 Programmable chip select timing control

The ST10F296E allows the user to adjust the position of the $\overline{CSx}$ line changes. By default (after reset), the $\overline{CSx}$ lines change half a CPU clock cycle (7.8 ns at 64 MHz of CPU clock) after the rising edge of ALE. With the CSCFG bit set in the SYSCON register the $\overline{CSx}$ lines change with the rising edge of ALE, thus the $\overline{CSx}$ lines and the address lines change at the same time (see *Figure 18*).

8.2 $\overline{READY}$ programmable polarity

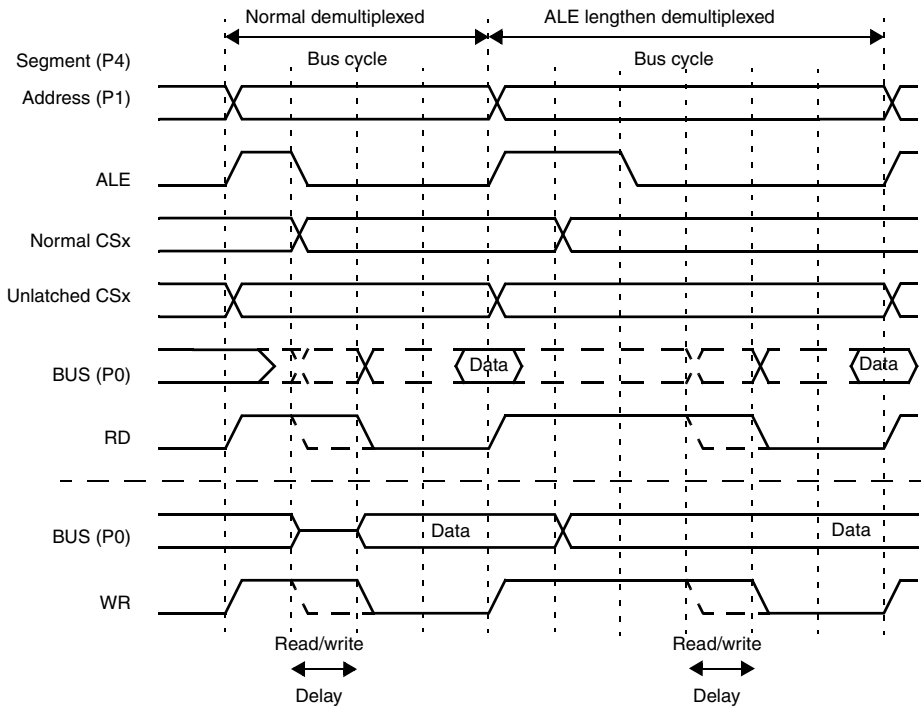
The active level of the $\overline{READY}$ pin can be selected by software via the RDYPOL bit in the BUSCONx registers.

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 the RDYPOL bit in the associated BUSCON register.

BUSCONx registers are described in *Section 23.10: System configuration registers on page 280*.

Note: ST10F296E has no internal pull-up resistor on the READY pin.

Figure 18. Chip select delay



8.3 $\overline{EA}$ functionality

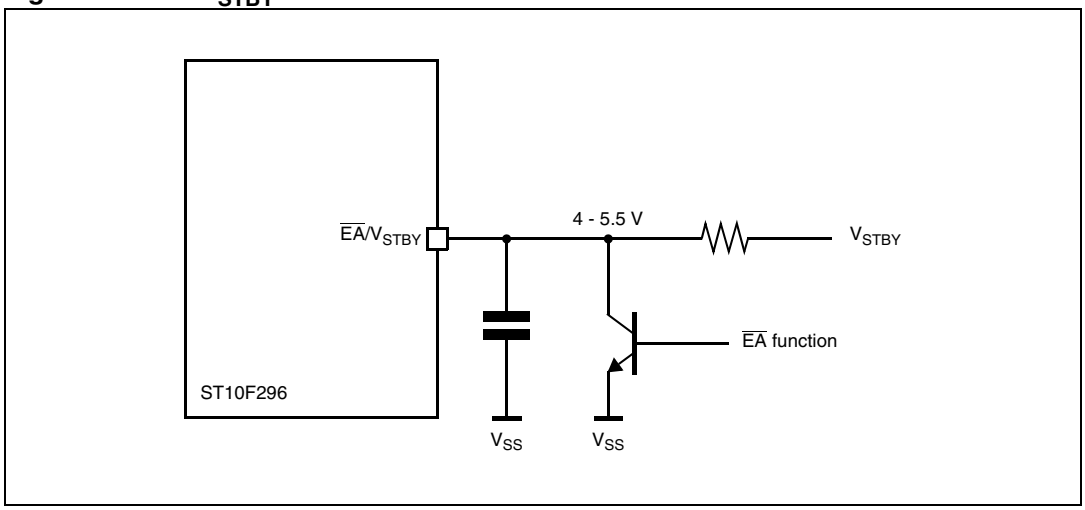
The $\overline{EA}$ pin of the ST10F296E is shared with the V_{STBY} supply pin. When V_{DD} main is on and stable, V_{STBY} can be temporarily grounded: The logic that in standby mode is powered by V_{STBY} (that is standby voltage regulator and 16 Kbyte portion of XRAM), is powered by V_{DD} main. This allows the $\overline{EA}$ pin to be driven low during reset, as requested, to configure the system to start from the external memory.

An appropriate external circuit must be provided to manage dynamically both functionalities associated with the $\overline{EA}$ pin. During reset and with stable V_{DD}, the pin can be tied low, while after reset (or before turning off the main V_{DD} to enter in standby mode) the V_{STBY} supply is applied.

Figure 19 shows a diagram of a possible external circuit. Care should be taken when implementing the resistance for current limitation of bipolar. The resistance should not disturb standby mode when some current (in the order of hundreds of µA) is provided to the device by the V_{STBY} voltage supply source. The voltage at the $\overline{EA}$ pin of ST10F296E should not become lower than 4.5 V.

To reduce the effect of current consumption transients on the V_{STBY} pin (refer to I_{SB3} in *Section 24: Electrical characteristics*) which may create voltage drops if a very low power external voltage regulator is used, it is suggested to add an external capacitance which can filter the eventual current peaks. Additional care must be paid to external hardware to limit the current peaks due to the presence of the capacitance (when $\overline{EA}$ functionality is used and the external bipolar is turned on, see *Figure 19*).

Figure 19. $\overline{EA}/V_{STBY}$ external circuit



9 Interrupt system

The interrupt response time for internal program execution is from 78 ns to 187.5 ns at 64 MHz CPU clock.

The ST10F296E architecture supports several mechanisms for fast, flexible responses 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 ST10F296E has eight PEC channels, each of them offers such fast interrupt-driven data transfer capabilities.

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

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

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

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

Table 47 shows all the available ST10F296E interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers.

Table 47. 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
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

Table 47. Interrupt sources (continued)

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

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

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

9.1 XPeripheral interrupt

The limited number of XBus interrupt lines of the present ST10 architecture, imposes some constraints on the implementation of the new functionality. In particular, the additional XPeripherals SSC1, ASC1, I^2C, PWM1, and RTC need some resources to implement interrupt and PEC transfer capabilities. For this reason, a sophisticated but very flexible multiplexed structure for the interrupt management is proposed (see *Figure 20*). It shows the basic structure replicated for each of the four XInterrupt available vectors (XP0INT, XP1INT, XP2INT, and XP3INT).

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

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

When different sources submit an interrupt request, the enable bits (byte high of the XIRxSEL register) define a mask which controls which sources are associated with the unique available vector. If more than one source is enabled to issue the request, the service routine has to identify the real event to be serviced. This can be done by checking the flag bits (byte low of the XIRxSEL register). Note that the flag bits can also provide information about events which are not currently serviced by the interrupt controller (since they are masked through the enable bits). This allows effective software management in the absence of the possibility to serve the related interrupt request. A periodic polling of the flag bits may be implemented inside the user application.

Note: The XIRxSEL registers are mapped into the XMiscellaneous area. Therefore, they can be accessed only if the XMISCEN and XPEN bits are set in the XPERCON and SYSCON registers respectively.

Figure 20. XInterrupt basic structure

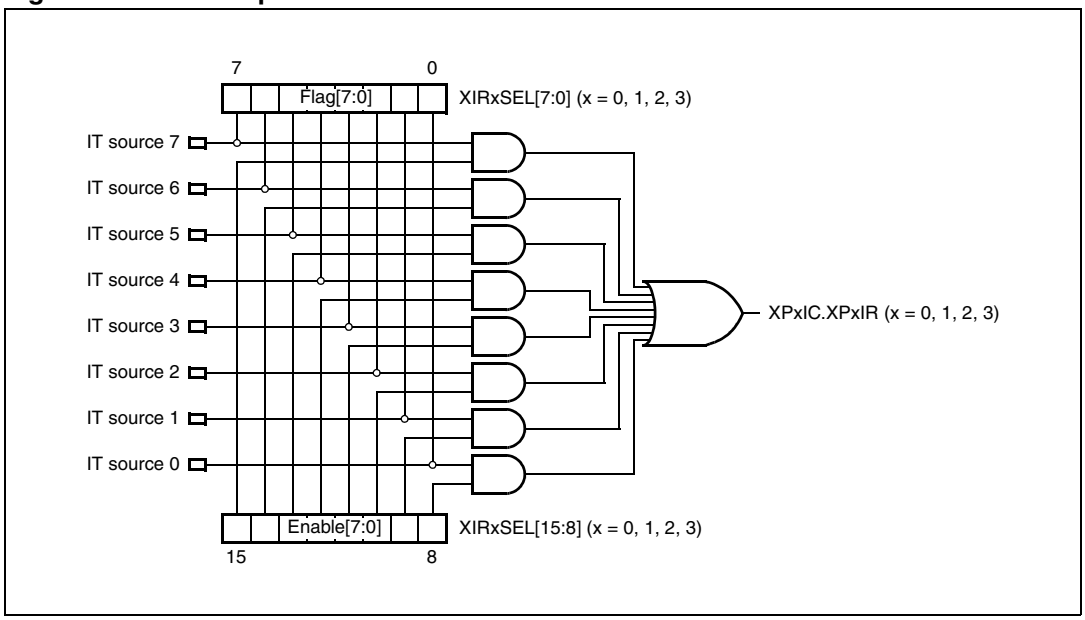


Table 48 summarizes the mapping of the different interrupt sources which share the four XInterrupt vectors.

Table 48. XInterrupt detailed mapping

Source	XP0INT	XP1INT	XP2INT	XP3INT
CAN1 interrupt	x			x
CAN2 interrupt		x		x
I^2C receive	x	x	x	
I^2C transmit	x	x	x	
I^2C error				x
SSC1 receive	x	x	x	
SSC1 transmit	x	x	x	
SSC1 error				x
ASC1 receive	x	x	x	
ASC1 transmit	x	x	x	
ASC1 transmit buffer	x	x	x	
ASC1 error				x
PLL unlock/OWD				x
PWM1 channel 3...0			x	x

9.2 Exception and error traps list

Table 49 shows all of the possible exceptions or error conditions that can arise during run-time.

Table 49. Trap priorities

Exception condition	Trap flag	Trap vector	Vector location	Trap number	Trap priority(1)
Reset functions:					
Hardware reset		RESET	00'0000h	00h	III
Software reset		RESET	00'0000h	00h	III
Watchdog timer overflow		RESET	00'0000h	00h	III
Class A hardware traps:					
Nonmaskable 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
MAC Interruption	MACTRP	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 class B traps have the same trap number, trap vector, and lower priority compared to class A traps and resets.
Each class A trap has a dedicated trap number (and vector). They are prioritized in the second priority level.
Resets have the highest priority level and the same trap number.
The PSW.ILVL CPU priority is forced to the highest level (15) when these exceptions are serviced.

10 Capture/compare (CAPCOM) units

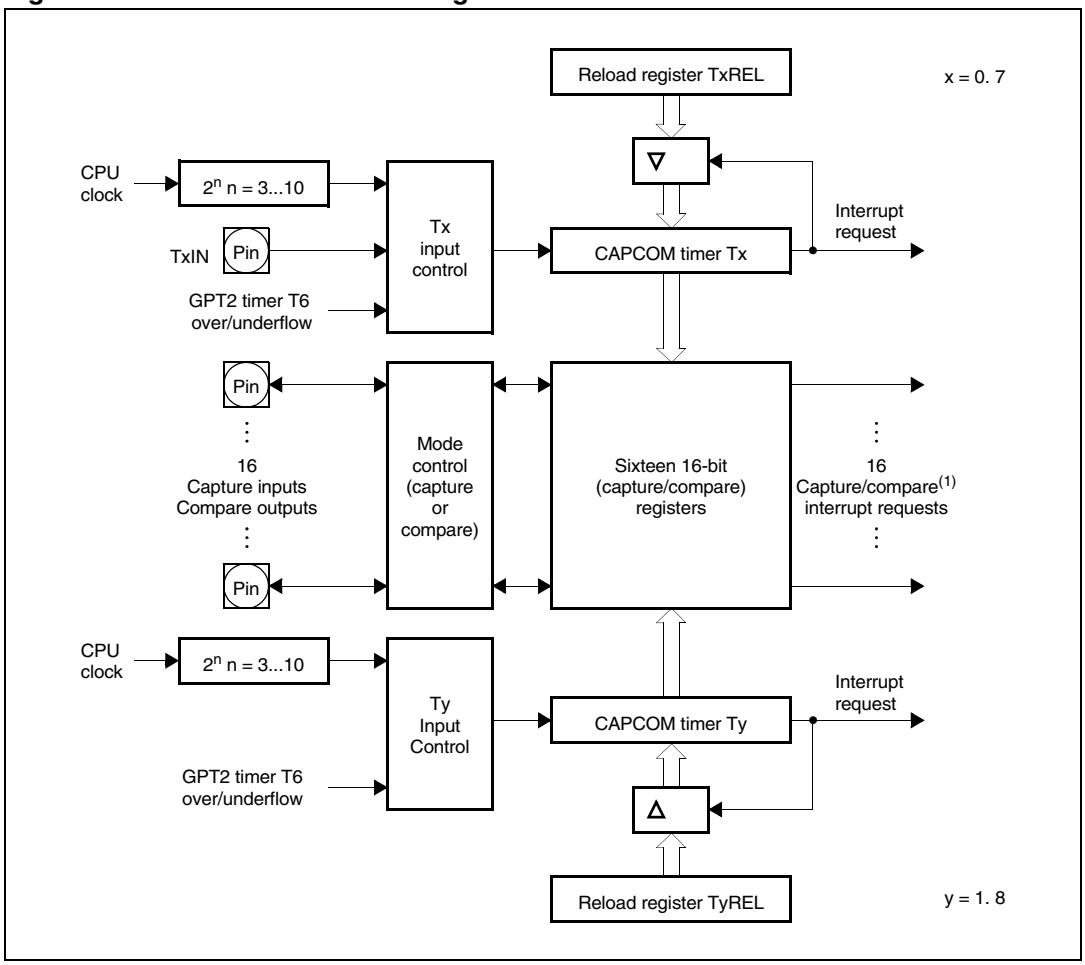
The ST10F296E has two 16 channel CAPCOM units as shown in *Figure 21: CAPCOM unit block diagram*. These support generation and control of timing sequences on up to 32 channels with a maximum resolution of 125 ns at 64 MHz CPU clock. The CAPCOM units are typically used to handle high speed I/O tasks such as pulse and waveform generation, 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 22* and *Figure 23*).

The input clock for the timers is programmable to several prescaled values of the internal system clock, or it 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 for 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 21* shows the basic structure of the two CAPCOM units.

Figure 21. CAPCOM unit block diagram



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

Figure 22. Block diagram of CAPCOM timers T0 and T7

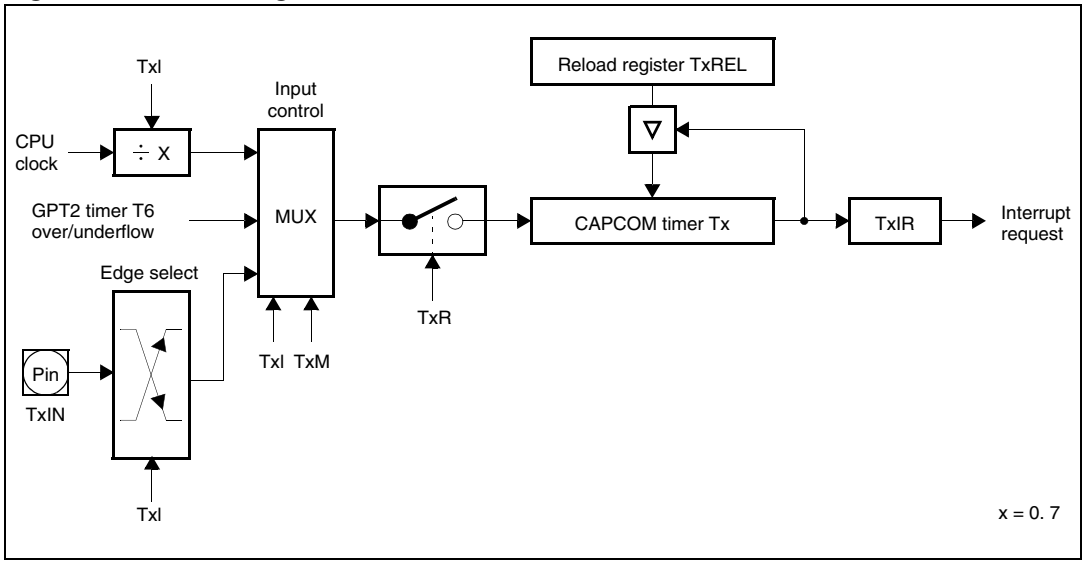
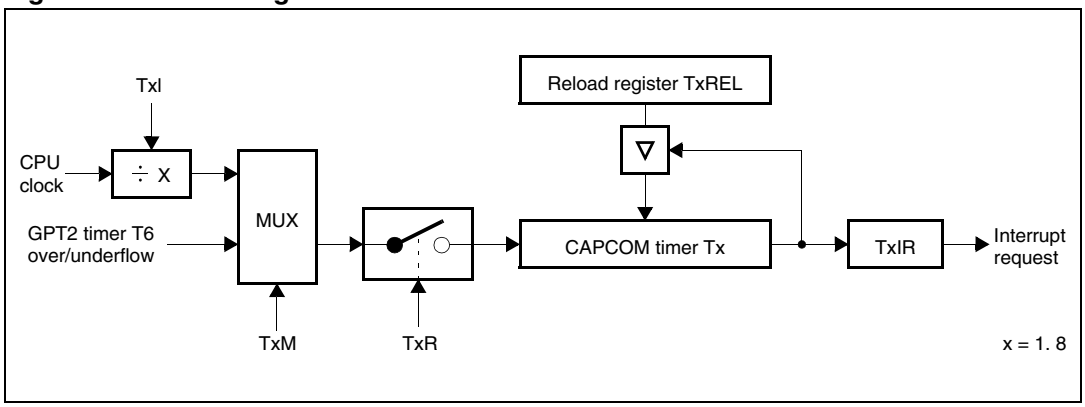


Figure 23. Block diagram of CAPCOM timers T1 and T8



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 are 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 are taken based on the selected compare mode (see *Table 50*).

The input frequencies f_{Tx}, for the timer input selector Tx, are determined as a function of the CPU clocks. The timer input frequencies, resolution and periods which result from the selected prescaler option in TxI when using a 40 MHz and 64 MHz CPU clock are listed in *Table 51* and *Table 52* respectively.

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

Table 50. 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 to 1 on match pin reset to 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 51. CAPCOM timer input frequencies, resolution, and periods at 40 MHz

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

Table 52. CAPCOM timer input frequencies, resolution, and periods at 64 MHz

f_{CPU} = 25 MHz	Timer input selection TxI							
	000b	001b	010b	011b	100b	101b	110b	111b
Prescaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input frequency	8 MHz	4 MHz	2 MHz	1 kHz	500 kHz	250 kHz	128 kHz	64 kHz
Resolution	125 ns	250 ns	0.5 µs	1.0 µs	2.0 µs	4.0 µs	8.0 µs	16.0 µs
Period	8.2 ms	16.4 ms	32.8 ms	65.5 ms	131.1 ms	262.1 ms	524.3 ms	1.049 s

11 General purpose timer unit

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

11.1 GPT1

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

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

In counter mode, the timer is clocked with 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 53 and *Table 54* list the timer input frequencies, resolution and periods for each prescaler option at 40 MHz and 64 MHz CPU clock respectively. 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 change 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 are 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.

Figure 24 shows the block diagram of the GPT1.

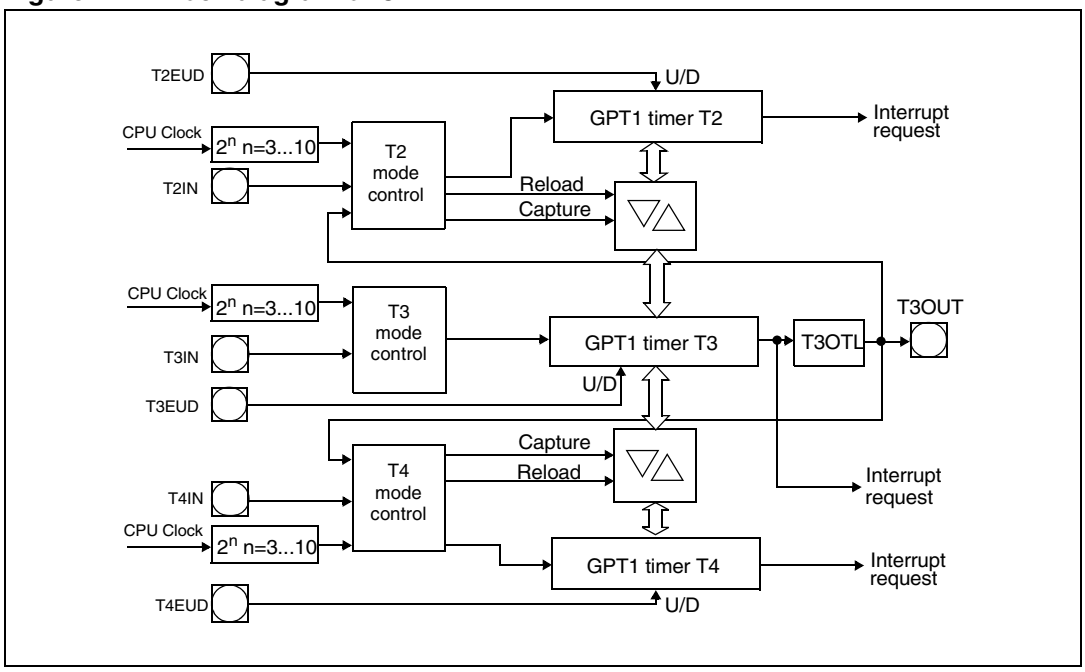
Table 53. GPT1 timer input frequencies, resolution, and periods at 40 MHz

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

Table 54. GPT1 timer input frequencies, resolution, and periods at 64 MHz

f_{CPU} = 64 MHz	Timer input selection T2I/T3I /T4I							
	000b	001b	010b	011b	100b	101b	110b	111b
Prescaler factor	8	16	32	64	128	256	512	1024
Input frequency	8 MHz	4 MHz	2 MHz	1 kHz	500 kHz	250 kHz	128 kHz	64 kHz
Resolution	125 ns	250 ns	0.5 µs	1.0 µs	2.0 µs	4.0 µs	8.0 µs	16.0 µs
Period maximum	8.2 ms	16.4 ms	32.8 ms	65.5 ms	131.1 ms	262.1 ms	524.3 ms	1.049 s

Figure 24. Block diagram of GPT1



11.2 GPT2

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

The state of this latch may be used to clock timer T5, or it may be output on a port pin (T6OUT). The overflow/underflow of timer T6 can also 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 a software overhead.

The capture trigger (timer T5 to CAPREL) may also be generated upon transition of the GPT1 timer T3 inputs, T3IN and/or T3EUD. This is advantageous when T3 operates in incremental interface mode.

Table 55 and *Table 56* list the timer input frequencies, resolution and periods for each pre-scaler option at 40 MHz and 64 MHz CPU clock respectively. This also applies to the gated timer mode of T6 and to the auxiliary timer T5 in timer and gated timer mode.

Figure 25 shows the block diagram of the GPT2.

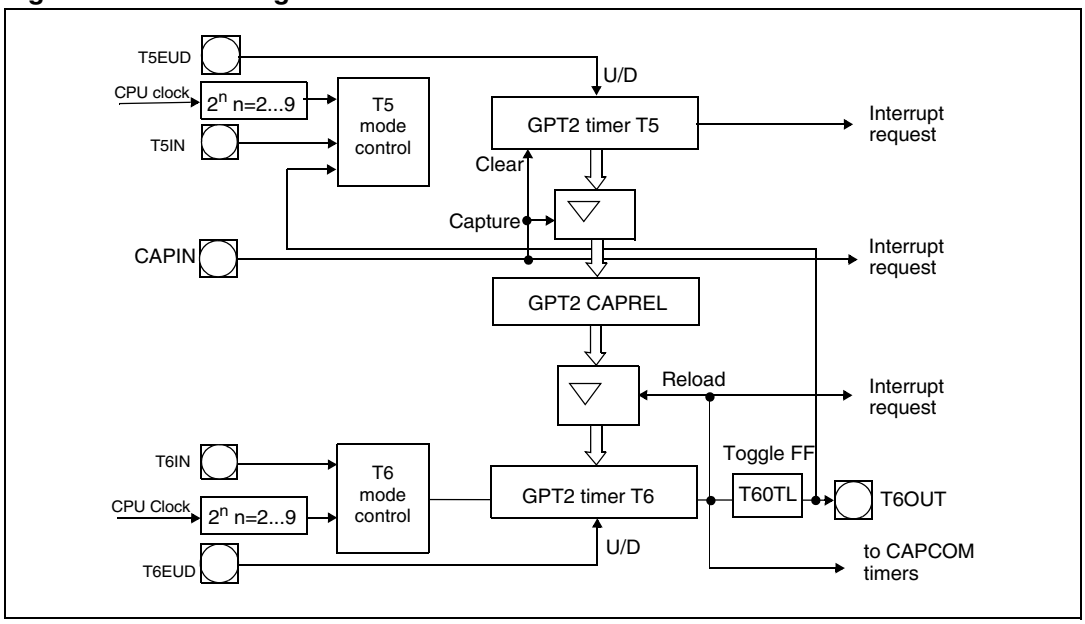
Table 55. GPT2 timer input frequencies, resolution, and period at 40 MHz

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

Table 56. GPT2 timer input frequencies, resolution, and period at 64 MHz

f_{CPU} = 64 MHz	Timer Input Selection T5I / T6I							
	000b	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
Prescaler factor	4	8	16	32	64	128	256	512
Input frequency	16 MHz	8 MHz	4 MHz	2 MHz	1 kHz	500 kHz	250 kHz	128 kHz
Resolution	62.5 ns	125 ns	250 ns	0.5 µs	1.0 µs	2.0 µs	4.0 µs	8.0 µs
Period maximum	4.1 ms	8.2 ms	16.4 ms	32.8 ms	65.5 ms	131.1 ms	262.1 ms	524.3 ms

Figure 25. Block diagram of GPT2

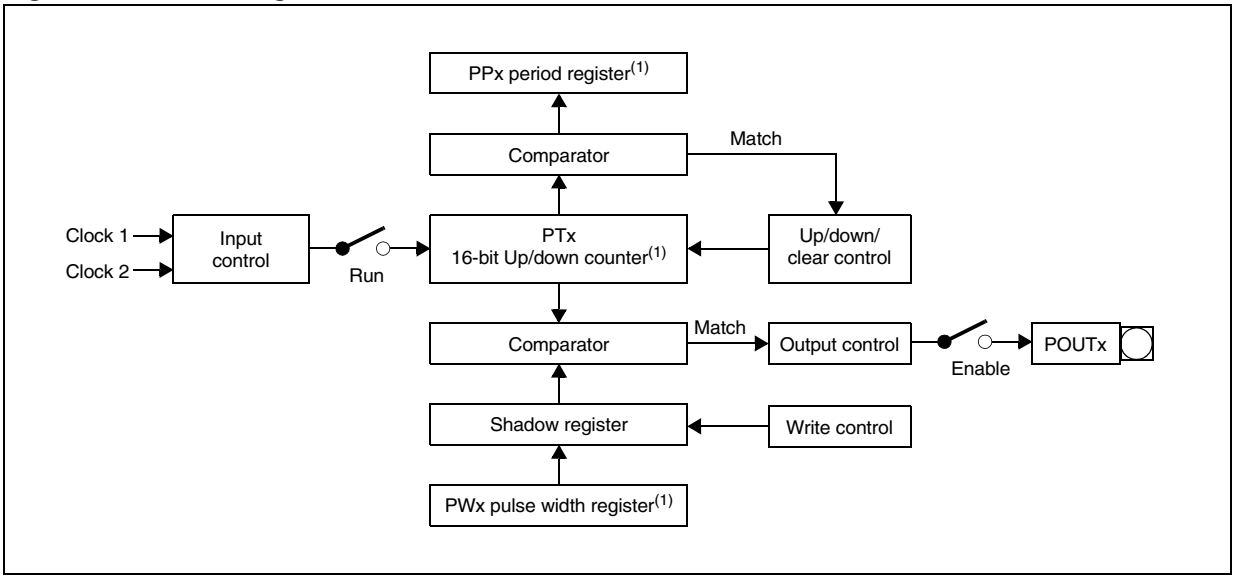


12 Pulse-width modulation (PWM) modules

Two PWM modules are available on ST10F296E: Standard PWM0 and XBus PWM1. They can generate up to four PWM output signals each, using edge-aligned or centre-aligned PWM. In addition, the PWM modules can generate PWM burst signals and single shot outputs. *Table 57* and *Table 58* show the PWM frequencies for different resolutions. The level of the output signals is selectable and the PWM modules can generate interrupt requests.

Figure 26 shows the block diagram of the PWM module.

Figure 26. Block diagram of PWM module



1. User readable/writeable register

Table 57. 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	25 ns	156.25 kHz	39.1 kHz	9.77 kHz	2.44 Hz	610 Hz
CPU clock/64	1.6 µs	2.44kHz	610 Hz	152.6 Hz	38.15Hz	9.54 Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU clock/1	25 ns	78.12 kHz	19.53 kHz	4.88 kHz	1.22 kHz	305.2 Hz
CPU clock/64	1.6 µs	1.22 kHz	305.17 Hz	76.29 Hz	19.07 Hz	4.77 Hz

Table 58. PWM unit frequencies and resolution at 64 MHz CPU clock

Mode 0	Resolution	8-bit	10-bit	12-bit	14-bit	16-bit
CPU clock/1	15.6 ns	250 kHz	62.5 kHz	15.63 kHz	3.91 Hz	977 Hz
CPU clock/64	1.0 µs	3.91 kHz	976.6 Hz	244.1 Hz	61.01 Hz	15.26 Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU clock/1	15.6 ns	125 kHz	31.25 kHz	7.81 kHz	1.95 kHz	488.3 Hz
CPU clock/64	1.0 µs	1.95 kHz	488.28 Hz	122.07 Hz	30.52 Hz	7.63 Hz

12.1 XPWM output signals

The output signals of the four XPWM channels (XPOUT3...XPOUT0) are available as dedicated pins. The XPWM signals are XORed with the outputs of the XPOLAR register before being driven to the dedicated pins. This allows the XPWM signal (XPOLAR.x = 0) or the inverted XPWM signal (XPOLAR.x = 1) to be driven directly.

Note: Using open-drain mode allows two or more XPWM outputs to be combined through an AND-wired configuration, using an external pull-up device. This provides a type of burst mode for any XPWM channel.

12.2 XPWM registers

XPOLAR register

The XPWMPORT register controls the specific XPWM output pins. Each output can be enabled/disabled which allows the XPWM to be configured as a push-pull or open-drain driver. In addition, the signal coming from the XPOLAR register is inverted. If both XPOLAR.Y and XP.y are set, no inversion is achieved.

XPOLAR (EC04h) XBus Reset value: 0000h

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

Table 59. XPOLAR register description

Bit	Bit name	Function
15-4	-	Reserved
3-0	XPOLAR.Y	XPWM channel Y polarity bit 0: Polarity of channel Y is normal 1: Polarity of channel Y is inverted

XPWMPORT register

XPWMPORT (EC80h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				XODP.3	XP.3	XDP.3	XODP.2	XP.2	XDP.2	XODP8.1	XP.1	XDP.1	XODP.0	XP.0	XDP.0
-				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 60. XPWMPORT register description

Bit	Bit name	Function
15-12	-	Reserved
11, 8, 5, 2	XODP.y	Port open-drain control register bit y 0: Port line XPOUT.y output driver in push-pull mode 1: Port line XPOUT.y output driver in open-drain mode
10, 7, 4, 1	XP.y	Port data register bit y
9, 6, 3, 0	XDP.y	Port direction register bit y 0: Port line XPOUT.y is an input (high impedance) 1: Port line XPOUT.y is an output

The XPWMPORT register is enabled and visible only when the XPEN and XPWMEN bits of the SYSCON and XPERCON registers respectively are set.

12.2.1 Software control of the XPWM outputs

In an application, the XPWM output signals are generally controlled by the XPWM module. However, it may be necessary to influence the level of the XPWM output pins via software, either to initialize the system or to react to some extraordinary conditions such as a system fault or an emergency.

Clearing the timer run bit PTRx stops the associated counter and leaves the respective output at its current level.

The individual XPWM channel outputs are controlled by comparators according to the formula:

PWM output signal = [XPTx] ≥ [XPWx shadow latch]

Whenever software changes register XPTx, the respective output reflects the condition after the change. Loading timer XPTx with a value greater than or equal to the value in XPWx immediately sets the respective output, an XPTx value below the XPWx value clears the respective output.

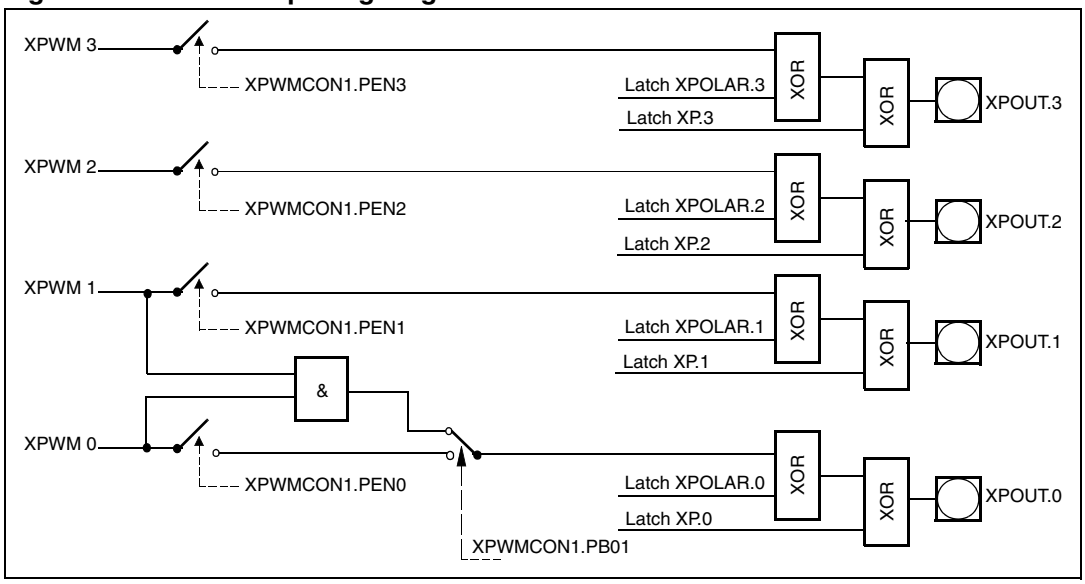
By clearing or setting the respective XPWMPORT output latch the XPWM channel signal is driven directly or inverted to the port pin.

Clearing the enable bit PENx disconnects the XPWM channel and switches the respective pin to the value in the port output latch XP.y.

Note: To prevent further PWM pulses from occurring after such a software intervention the respective counter must be stopped first.

Figure 27 shows the XPWM output signal generation.

Figure 27. XPWM output signal generation



13 Parallel ports

The ST10F296E MCU provides up to 143 I/O lines with programmable features. The MCU is therefore very flexible for a wide range of applications.

The ST10F296E has 11 groups of I/O lines organized as follows:

- Port 0 is a two-time, 8-bit port named P0L (low is the least significant byte) and P0H (high is the most significant byte)
- Port 1 is a two-time, 8-bit port named P1L and P1H
- Port 2 is a 16-bit port
- Port 3 is a 15-bit port (P3.14 line is not implemented)
- Port 4 is an 8-bit port
- Port 5 is a 16-bit input only port
- Port 6, Port 7 and Port 8 are 8-bit ports
- XPort 9 is a 16-bit general purpose port
- XPort 10 is a 16-bit input only port

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

For example the output drivers of seven of the ports (2, 3, 4, 6, 7, 8, and 9) can be configured (bit-wise) for push-pull or open-drain operation using the ODPx registers (and the XODP9 register for XPort 9).

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

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

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

I/O lines support an alternate function which is detailed in *Section 13.1.4*, *Section 13.2.2*, *Section 13.3.2*, and *Section 13.4.2*.

Note: The I/O ports XPort 9 and XPort10 are not mapped on the SFR space but on the internal XBus interface. They are enabled by setting the XPEN bit, bit 2, of the SYSCON register and bit 11 of the XPERCON register. On the XBus interface, the registers are not bit-addressable

Figure 28. SFRs and pins associated with the parallel ports (A)

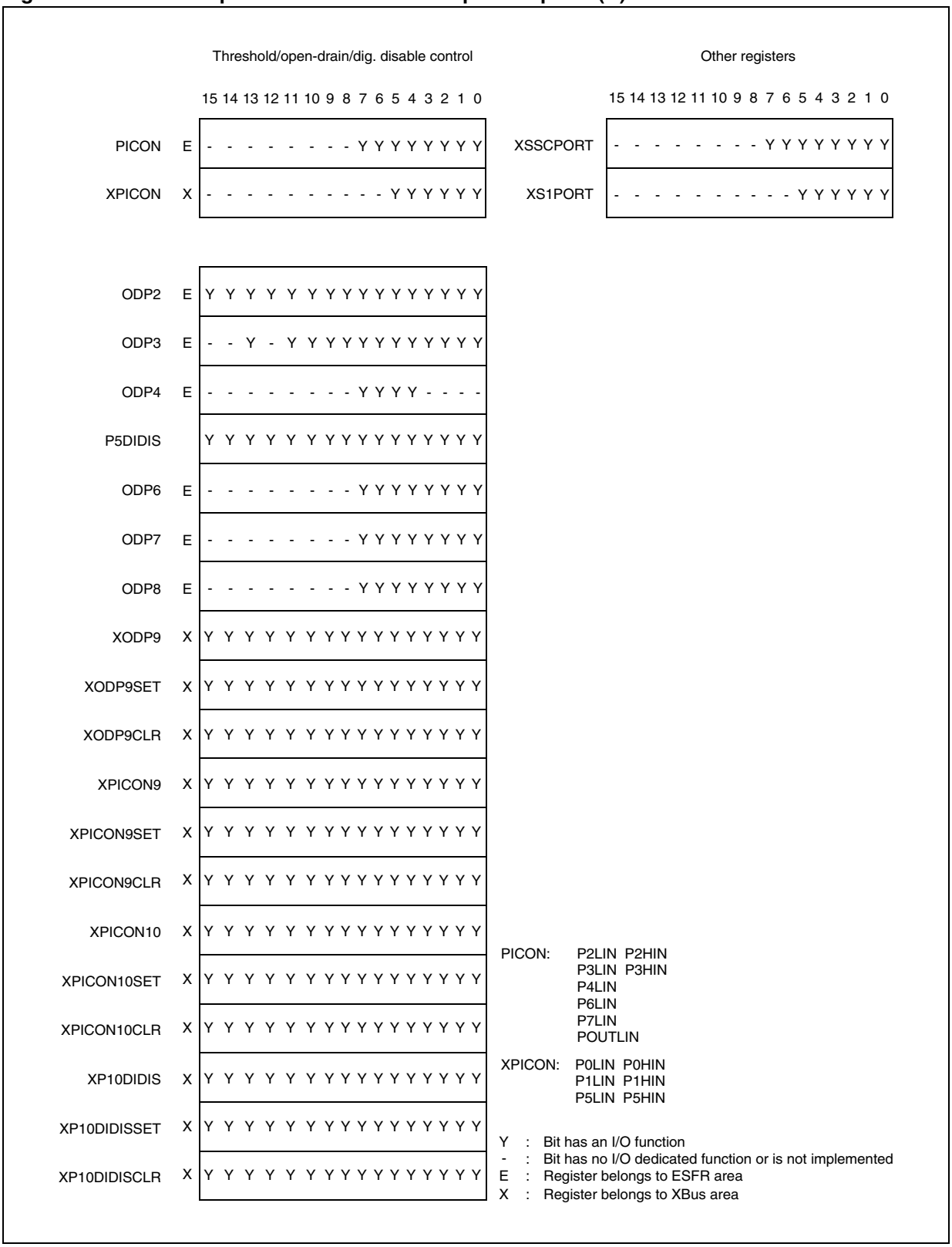
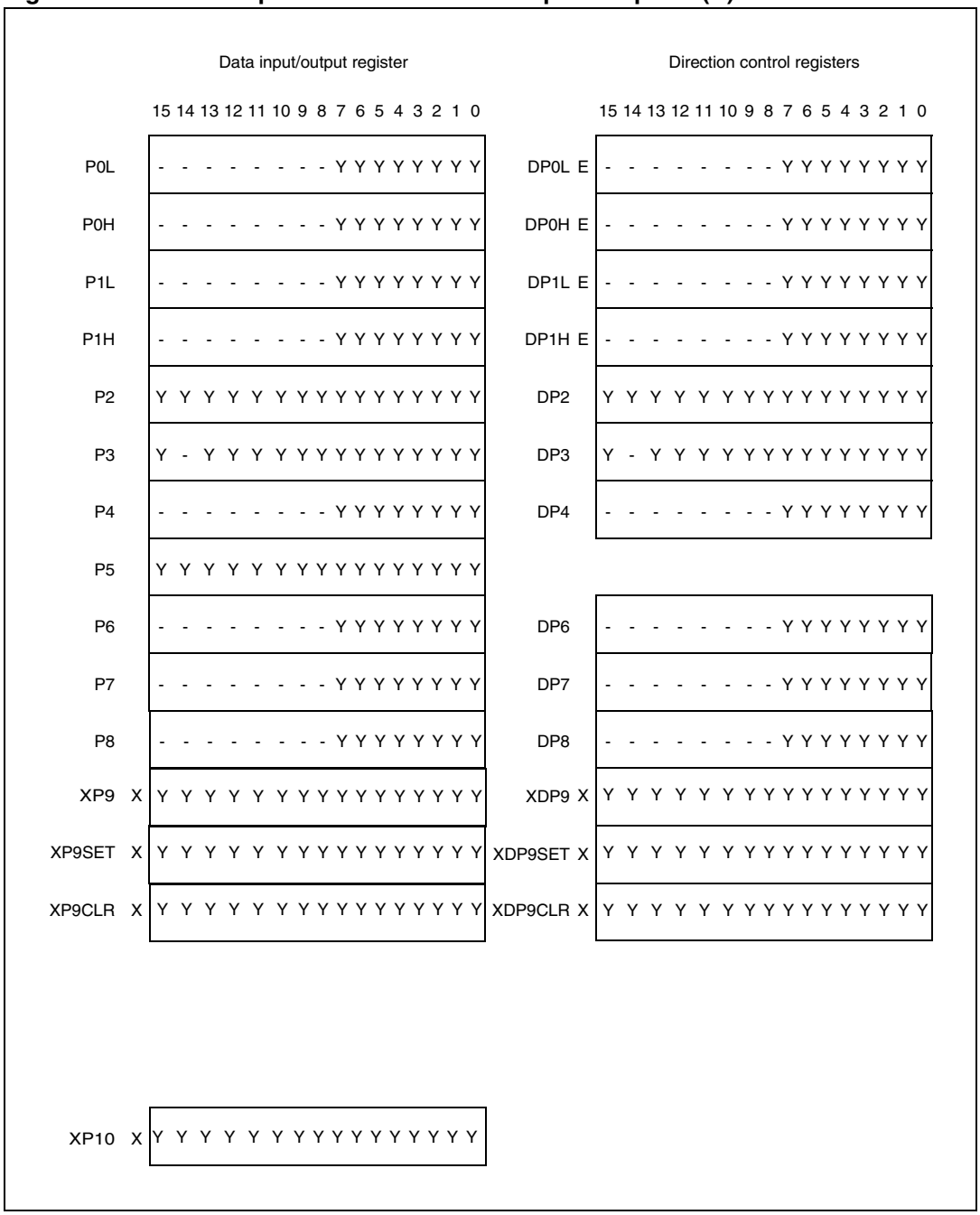


Figure 29. SFRs and pins associated with the parallel ports (B)



13.1 I/O special features

13.1.1 Open-drain mode

Some of the I/O ports of the ST10F296E support the open-drain capability. This programmable feature may be used with an external pull-up resistor to get an AND wired logical function.

This feature is implemented for ports P2, P3, P4, P6, P7, P8, and XP9 (see *Section 13.4*, *Section 13.5*, *Section 13.6*, *Section 13.8*, *Section 13.9*, *Section 13.10*, and *Section 13.11*) and is controlled through the respective open-drain control registers ODPx (and the XODP9 register for XP9). 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 (see *Figure 30*). All ODPx registers are located in the ESFR space. The XODP9 register is in the XBus space.

13.1.2 Input threshold control

The standard inputs of the ST10F296E determine the status of input signals according to TTL levels. To accept and recognize noisy signals, CMOS input thresholds can be selected instead of standard TTL thresholds for all pins. CMOS thresholds are defined above the TTL thresholds and feature a higher hysteresis to prevent inputs from toggling while the respective input signal level is near its threshold.

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 31*).

13.1.3 I/0 port registers

The port input control registers, PICON and XPICON, are used to select thresholds for each byte of the indicated ports. This means the 8-bit ports P0L, P0H, P1L, P1H, P4, P7, and P8 are controlled by one bit each while ports P2, P3, and P5 are controlled by two bits each. In addition, the registers XPICON9 and XPICON10 allow single bit input threshold control for XP9 and XP10 respectively.

For XPort 9 and XPort 10, the bit-addressable feature is available via specific ‘set’ and ‘clear’ registers. These are:

- XPICON9SET and XPICON9CLR for XPICON9
- XPICON10SET and XPICON10CLR for XPICON10

PICON register

PICON (F1C4h/E2h) ESFR Reset value: --00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P8 LIN	P7 LIN	P6 LIN	P4 LIN	P3 HIN	P3 LIN	P2 HIN	P2 LIN
-								RW	RW	RW	RW	RW	RW	RW	RW

Table 61. PICON register description

Bit	Bit name	Function
15-8	-	Reserved
7, 6, 5, 4, 2, 0	PxLIN	Port x low byte input level selection 0: Pins Px.7 to Px.0 switch on standard TTL input levels 1: Pins Px.7 to Px.0 switch on standard CMOS input levels
3, 1	PxHIN	Port x high byte input level selection 0: Pins Px.15 to Px.8 switch on standard TTL input levels 1: Pins Px.15 to Px.8 switch on standard CMOS input levels

XPICON register

XPICON (EB26h) XBus Reset value: --00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P5 HIN	P5 LIN	P1 HIN	P1 LIN	P0 HIN	P0 LIN
-										RW	RW	RW	RW	RW	RW

Table 62. XPICON register description

Bit	Bit name	Function
15-6	-	Reserved
5, 3, 1	PxHIN	Port x high byte input level selection 0: Pins Px.15 to Px.8 switch on standard TTL input levels 1: Pins Px.15 to Px.8 switch on standard CMOS input levels
4, 2, 0	PxLIN	Port x low byte input level selection 0: Pins Px.7 to Px.0 switch on standard TTL input levels 1: Pins Px.7 to Px.0 switch on standard CMOS input levels

XPICON9 register

XPICON9 (EB98h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9IN.15	XP9IN.14	XP9IN.13	XP9IN.12	XP9IN.11	XP9IN.10	XP9IN.9	XP9IN.8	XP9IN.7	XP9IN.6	XP9IN.5	XP9IN.4	XP9IN.3	XP9IN.2	XP9IN.1	XP9IN.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 63. XPICON9 register description

Bit	Bit name	Function
15-0	XP9IN.y	Port 9 bit y input level selection 0: Port line XP9.y switch on standard TTL input levels 1: Port line XP9.y switch on standard CMOS input levels

XPICON9SET register

XPICON9SET (EB9Ah) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9INSET.15	XP9INSET.14	XP9INSET.13	XP9INSET.12	XP9INSET.11	XP9INSET.10	XP9INSET.9	XP9INSET.8	XP9INSET.7	XP9INSET.6	XP9INSET.5	XP9INSET.4	XP9INSET.3	XP9INSET.2	XP9INSET.1	XP9INSET.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 64. XPICON9SET register description

Bit	Bit name	Function
15-0	XP9INSET.y	Writing a 1 sets the corresponding bit of the XPICON9.y register. Writing a 0 has no effect

XPICON9CLR register

XPICON9CLR (EB9Ch) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9INCLR.15	XP9INCLR.14	XP9INCLR.13	XP9INCLR.12	XP9INCLR.11	XP9INCLR.10	XP9INCLR.9	XP9INCLR.8	XP9INCLR.7	XP9INCLR.6	XP9INCLR.5	XP9INCLR.4	XP9INCLR.3	XP9INCLR.2	XP9INCLR.1	XP9INCLR.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 65. XPICON9CLR register description

Bit	Bit name	Function
15-0	XP9INCLR.y	Writing a 1 clears the corresponding bit of the XPICON9.y register. Writing a 0 has no effect.

XPICON10 register

XPICON10 (EBD8h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP10IN.15	XP10IN.14	XP10IN.13	XP10IN.12	XP10IN.11	XP10IN.10	XP10IN.9	XP10IN.8	XP10IN.7	XP10IN.6	XP10IN.5	XP10IN.4	XP10IN.3	XP10IN.2	XP10IN.1	XP10IN.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 66. XPICON10 register description

Bit	Bit name	Function
15-0	XP10IN.y	Port 10 bit y input level selection 0: Port line XP10.y switches on standard TTL input levels 1: Port line XP10.y switches on standard CMOS input levels

Figure 30. Output drivers in push-pull mode and in open-drain mode

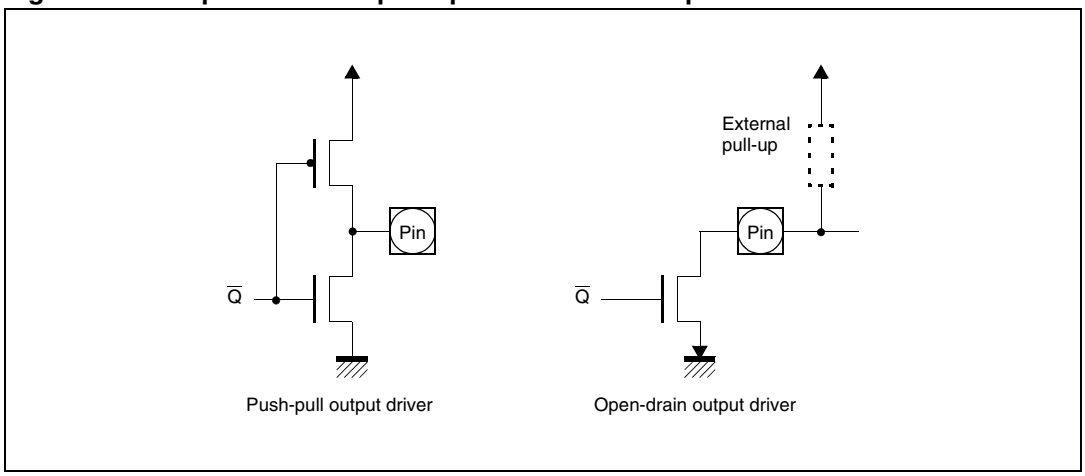
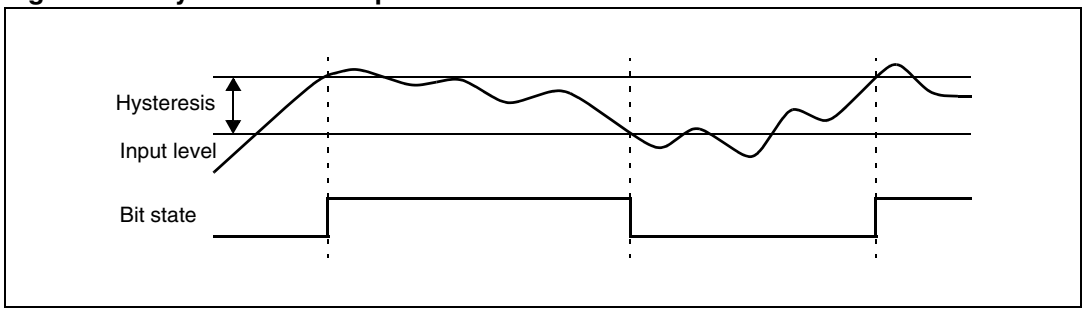


Figure 31. Hysteresis concept



13.1.4 Alternate port functions

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

- Port 0 and Port 1 may be used for address and data lines when accessing the external memory. Port 1 also provides input capture lines.
- Port 2, Port 7 and Port 8 are associated with the capture inputs or compare outputs of the CAPCOM units and/or with the outputs of the PWM0 module, the PWM1 module, and the ASC1. Port 2 is also used for fast external interrupt inputs and for timer 7 input.
- Port 3 includes the alternate functions of timers, serial interfaces, the optional bus control signal $\overline{BHE}$ and the system clock output (CLKOUT).
- Port 4 outputs the additional segment address bit A23 to A16 in systems where more than 64 Kbytes of memory are accessed directly. In addition, CAN1, CAN2 and I^2C lines are provided.
- Port 5 is used for the analog input channels of the ADC or for the timer control signals.
- Port 6 provides optional bus arbitration signals ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$), chip select signals, and SSC1 lines.
- XPort 9 is a general purpose input/output port
- XPort 10 is used for additional analog input channels of the ADC

If the alternate output function of a pin is being 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 affected by the alternate output function. The respective port latch should hold a 1, because its output is ANDed with the alternate output data (except for PWM output signals).

If the alternate input function of a pin is being 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, the direction of the pin can also be set to output. In this case, the pin reflects the state of the port output latch. Thus, the alternate input function reads the value stored in the port output latch. This can be used for testing purposes to allow a software trigger of an alternate input function by writing to the port output latch.

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

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

However, there are port lines where the direction of the port line is switched automatically.

For instance, in the multiplexed external bus modes of Port 0, the direction must be switched several times for an instruction fetch to output the addresses and to input the data.

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

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

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

All port lines that are not used for 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 to avoid undesired transitions on the output pins. This applies to single pins as well as to pin groups (see example below).

```
SINGLE_BIT:     BSET     P4.7              ; Initial output level is "high"
                BSET     DP4.7             ; Switch on the output driver
BIT_GROUP:      BFLDH    P4, #24H, #24H    ; Initial output level is "high"
                BFLDH    DP4, #24H, #24H   ; Switch on the output drivers
```

Note: When using several BSET pairs to control several pins of one port, the pairs must be separated by instructions which do not apply to the respective port (see Section 7: Central processing unit (CPU) on page 92).

13.2 Port 0

The two 8-bit ports, P0H and P0L, represent the higher and lower part of Port 0, respectively. Both halves of Port 0 can be written (for example via a PEC transfer) without affecting 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.

13.2.1 Port 0 registers

P0L and P0H registers

P0L (FF00h/80h) SFR Reset value: --00h

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

Table 67. P0L and P0H register description

Bit	Bit name	Function
15-8	-	Reserved
7-0	P0X.y	Port data register P0L or P0H bit y

DP0L and DP0H registers

DP0L (F100h/80h) ESFR Reset value: --00h

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

Table 68. DP0L and DP0H register description

Bit	Bit name	Function
15-8	-	Reserved
7-0	DP0X.y	Port direction register DP0L or DP0H bit y 0: Port line P0X.y is an input (high impedance) 1: Port line P0X.y is an output

13.2.2 Alternate functions of Port 0

When an external bus is enabled, Port 0 is used as a data bus or an 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).

Port 0 is also used to select the system startup configuration. During reset, Port 0 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 24.5: DC characteristics*) through an external pull-down device. A default configuration is selected when the respective Port 0 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.

Internal pull-up devices are designed so that external pull-down resistors (see *Section 24.5: DC characteristics*) can be used to apply a correct low level.

Such external pull-down resistors can remain connected to Port 0 pins during normal operation. However, care has to be taken that they do not disturb the normal function of Port 0 (for example, if the external resistor is too strong).

At the end of reset, the selected bus configuration is written to the BUSCON0 register. The configuration of the high byte of Port 0, is copied into the RP0H register.

RPOH is a read-only register that holds the selection for the number of chip selects and segment addresses. Software can read this register if required. When the reset is terminated, the internal pull-up devices are switched off, and Port 0 is switched to the appropriate operating mode.

During external access in multiplexed bus modes, Port 0 first outputs the 16-bit intra-segment address as an alternate output function. Port 0 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 access. The first memory cycle is for the low byte of the word and the second is for the high byte. During write cycles Port 0 outputs the data byte or word after outputting the address. During external access in de-multiplexed bus modes Port 0 reads the incoming instruction or data word or outputs the data byte or word (see *Figure 32*).

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 latch 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 Port 0 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 33* shows the structure of a Port 0 pin.

Figure 32. Port 0 I/O and alternate functions

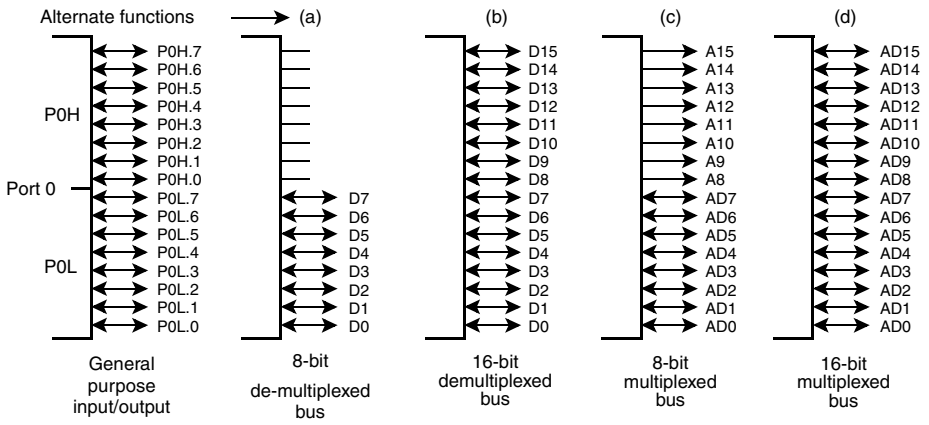
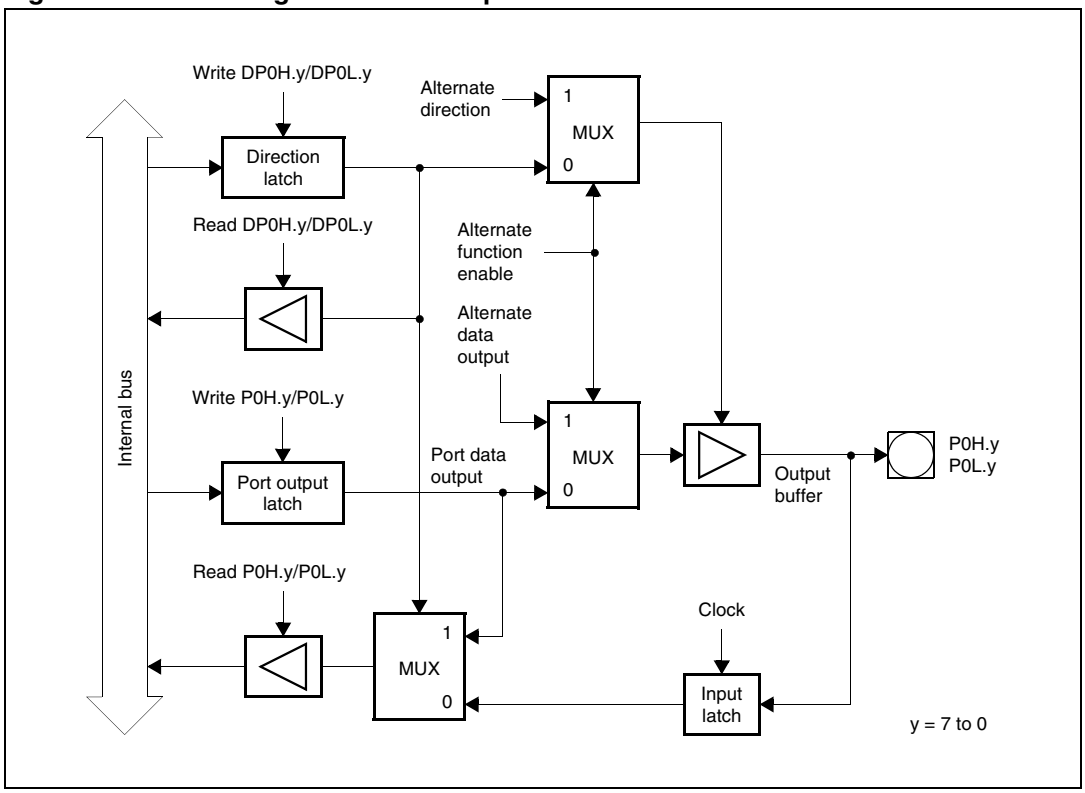


Figure 33. Block diagram of a Port 0 pin



13.3 Port 1

The two 8-bit ports P1H and P1L represent the higher and lower part of Port 1, respectively. Both halves of Port 1 can be written (for example 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.

13.3.1 Port 1 registers

P1L and P1H registers

P1L (FF04h/82h) SFR Reset value: --00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P1L.7	P1L.6	P1L.5	P1L.4	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
Reserved								P1H.7	P1H.6	P1H.5	P1H.4	P1H.3	P1H.2	P1H.1	P1H.0
-								RW	RW	RW	RW	RW	RW	RW	RW

Table 69. P1L and P1H register description

Bit	Bit name	Function
15-8	-	Reserved
7-0	P1X.y	Port data register P1L or P1H or bit y

DP1L and DP1H registers

DP1L (F104h/82h) ESFR Reset value: --00h

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

Table 70. DP1L and DP1H register description

Bit	Bit name	Function
15-8	-	Reserved
7-0	DP1X.y	Port direction register DP1L or DP1H bit y 0: Port line P1X.y is an input (high impedance) 1: Port line P1X.y is an output

13.3.2 Alternate functions of Port 1

When a demultiplexed external bus is enabled, Port 1 is used as an address bus. Note that demultiplexed bus modes use Port 1 as a 16-bit port. Otherwise all 16 port lines can be used for general purpose I/O. The upper four pins of Port 1 (P1H.7 to P1H.4) are also capture input lines for the CAPCOM2 unit (CC27-24 I).

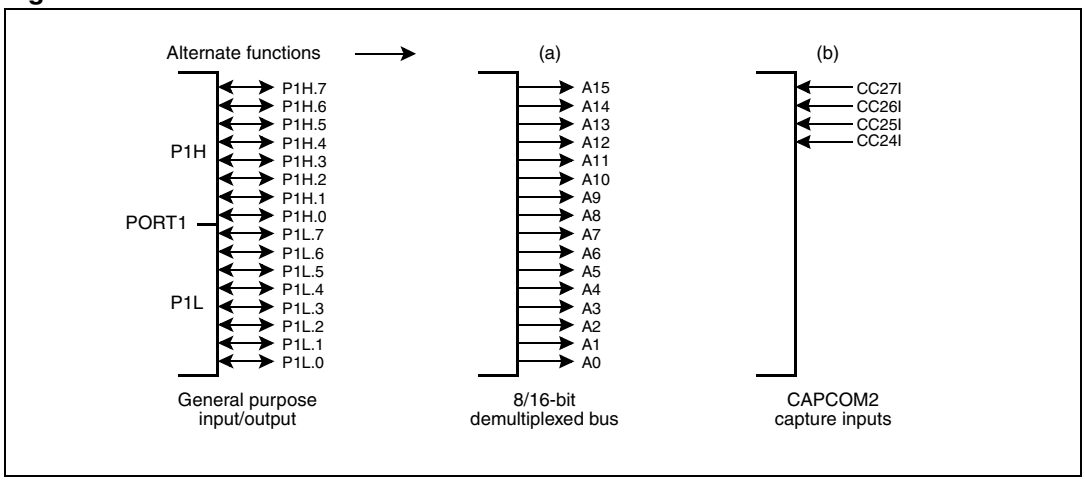
The capture input functions of pins P1H.7 to P1H.4 can be used for external interrupt inputs with a sample rate of eight CPU clock cycles.

As a side effect, the capture input capability of these lines can also be used in the address bus mode. Changes of the upper address lines may be detected, thereby triggering an interrupt request that performs some special service routines. External capture signals can only be applied if no address output is selected for Port 1.

During external access in demultiplexed bus modes, Port 1 outputs the 16-bit intra-segment address as an alternate output function.

During external access in multiplexed bus modes, when no BUSCON register selects a de-multiplexed bus mode, Port 1 is not used and is available for general purpose I/O.

Figure 34. Port 1 I/O and alternate functions

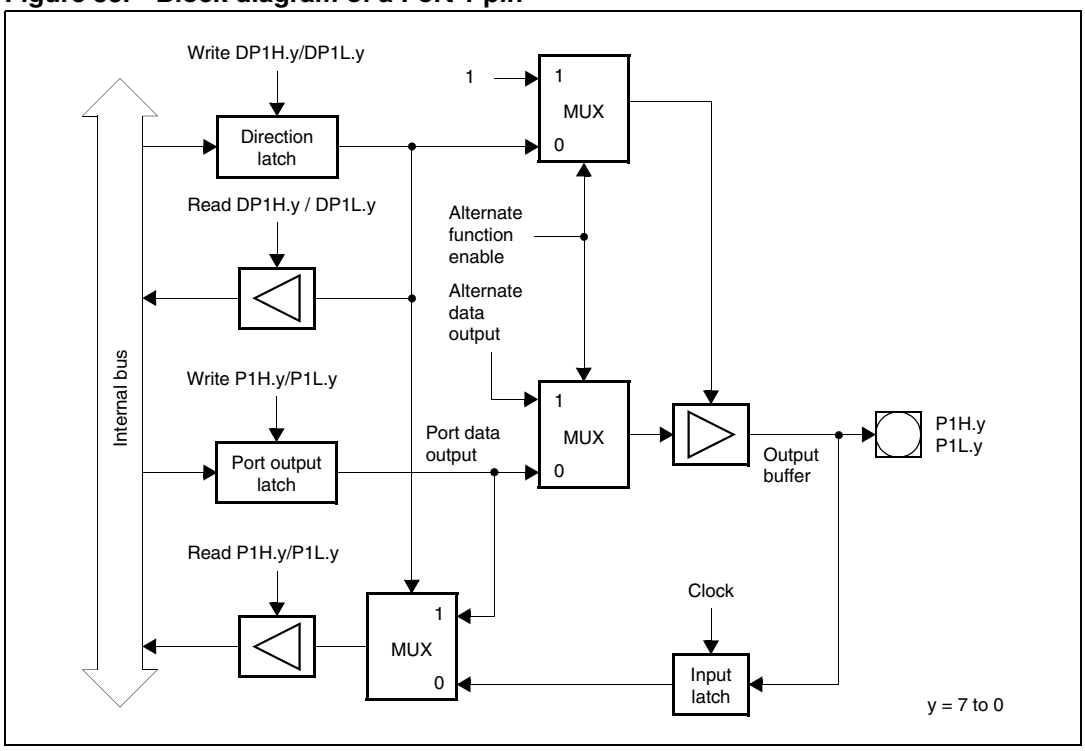


When an external bus mode is enabled, the direction of the port pin and the loading of data into the port output latch are controlled by the bus controller hardware. The input of the port output latch is disconnected from the internal bus and is switched to the line labeled 'alternate data output' via 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 that was last written by the user becomes active.

Figure 35 shows the structure of a Port 1 pin.

Figure 35. Block diagram of a Port 1 pin



13.4 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.

13.4.1 Port 2 registers

P2 register

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

Table 71. P2 register description

Bit	Bit name	Function
15-0	P2.y	Port data register P2 bit y

DP2 register

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

Table 72. DP2 register description

Bit	Bit name	Function
15-0	DP2.y	Port direction register DP2 bit y 0: Port line P2.y is an input (high impedance) 1: Port line P2.y is an output

ODP2 register

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

Table 73. ODP2 register description

Bit	Bit name	Function
15-0	ODP2.y	Port open-drain control register ODP2 bit y 0: Port line P2.y output driver in push-pull mode 1: Port line P2.y output driver in open-drain mode

13.4.2 Alternate functions of Port 2

All Port 2 lines (P2.15 to P2.0) can be configured as capture inputs or compare outputs (CC15IO to 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 is read since the pin represents the state of the output latch. This may 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 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 affects 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’. It is 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 is in the high impedance state and does not reflect the state of the output latch.

As can be seen from the port structure (*Figure 37*), 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. It 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 receives the value from the internal bus and the hardware triggered change is lost.

The capture input function of pins P2.7 to P2.0 can be used for external interrupt inputs with a sample rate of eight CPU clock cycles.

For pins P2.15 to P2.8, the sampling rate is eight CPU clock cycles when used as capture input, and one CPU clock cycle if used as fast external input.

The upper eight Port 2 lines (P2.15 to P2.8) also support fast external interrupt inputs (EX7IN to EX0IN). In addition, P2.15 is the input for CAPCOM2 timer T7 (T7IN). *Table 74* summarizes the alternate functions of Port 2. The pins of this port combine internal capture input bus data with compare output alternate data which is output before the port latch input.

Table 74. Alternate functions of Port 2

P2 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 Timer T7 external count input

Figure 36. Port 2 I/O and alternate functions

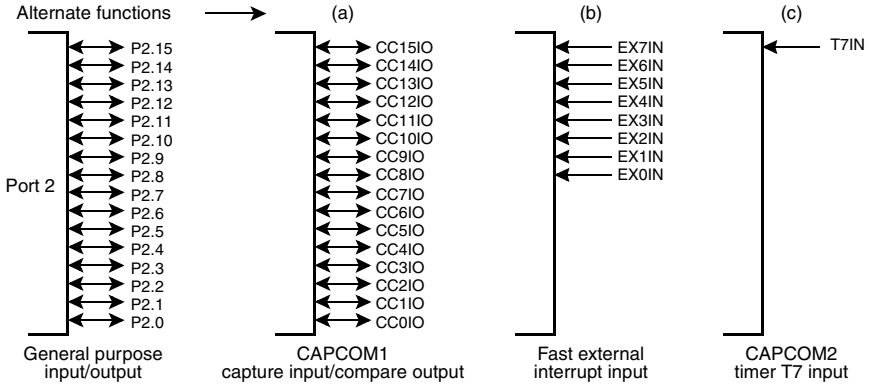
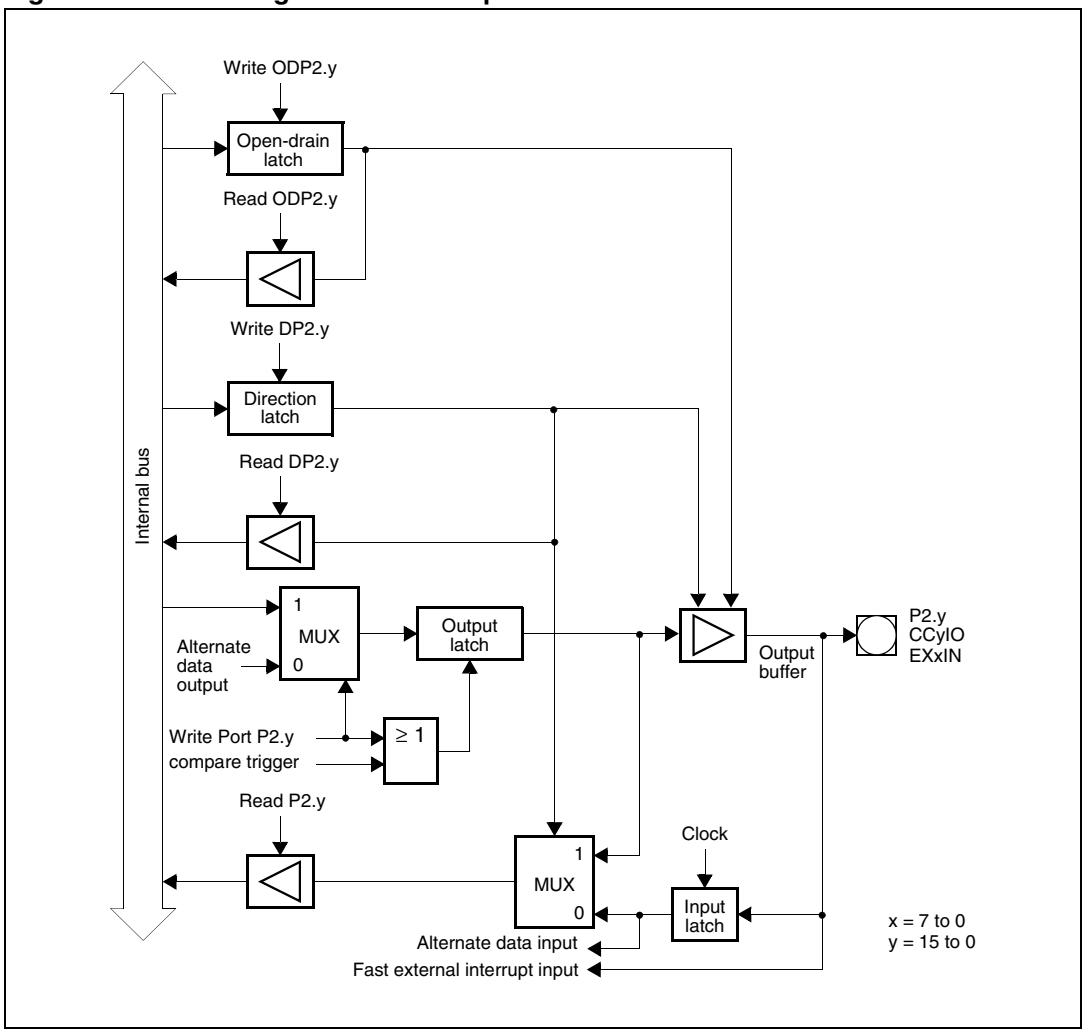


Figure 37. Block diagram of a Port 2 pin



13.4.3 Port 2 and external interrupts

These interrupt inputs are provided to service external interrupts which have 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 of the ST10F296E is controlled using the 'external interrupt source selection' register (EXISEL).

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(1)		EXI2SS(2)		EXI1SS		EXI0SS	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

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

Table 75. EXISEL register description

Bit	Bit name	Function
15-0	EXIxSS	External interrupt x source selection (x = 7 to 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

Table 76. External interrupt selection

EXIxSS	Port 2 pin	Alternate source	
0	P2.8	CAN1_RxD	P4.5
1	P2.9	CAN2_RxD/SCL	P4.4
2	P2.10	RTCSI (second)	Internal MUX
3	P2.11	RTCAI (alarm)	Internal MUX
4 to 7	P2.12 to 15	Not used (zero)	-

13.5 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 ODP3 (pins P3.15 and P3.12 do not support open-drain mode).

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

13.5.1 Port 3 registers

P3 register

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

Table 77. P3 register description

Bit	Bit name	Function
15, 13-0	P3.y	Port data register P3 bit y

DP3 register

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

Table 78. DP3 register description

Bit	Bit name	Function
15, 13-0	DP3.y	Port direction register DP3 bit y 0: Port line P3.y is an input (high impedance) 1: Port line P3.y is an output

ODP3 register

DP3 (F1C6h/E3h) ESFR Reset value: 0000h

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

Table 79. ODP3 register description

Bit	Bit name	Function
13, 11-0	ODP3.y	Port open-drain control register ODP3 bit y 0: Port line P3.y output driver in push-pull mode 1: Port line P3.y output driver in open-drain mode

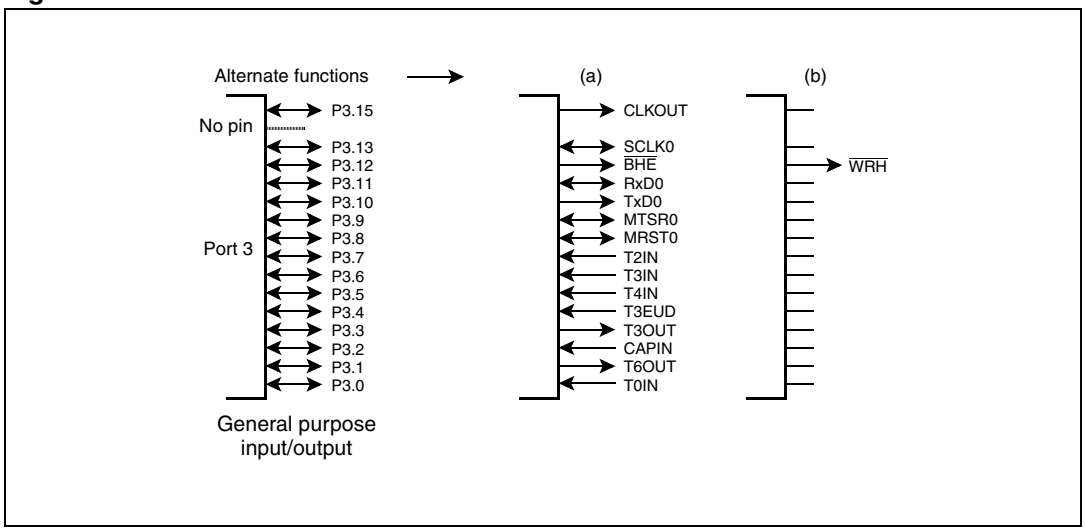
13.5.2 Alternate functions of Port 3

The pins of Port 3 are used for various functions which include external timer control lines, the two serial interfaces and the control lines $\overline{BHE}$ / $\overline{WRH}$ and CLKOUT.

Table 80. Port 3 alternative functions

Port 3 pin	Alternate function	
P3.0	T0IN	CAPCOM1 timer 0 count input
P3.1	T6OUT	Timer 6 toggle output
P3.2	CAPIN	GPT2 capture input
P3.3	T3OUT	Timer 3 toggle output
P3.4	T3EUD	Timer 3 external up/down input
P3.5	T4IN	Timer 4 count input
P3.6	T3IN	Timer 3 count input
P3.7	T2IN	Timer 2 count input
P3.8	MRST0	SSC master receive/slave transmit
P3.9	MTSR0	SSC master transmit/slave receive
P3.10	TxD0	ASC0 transmit data output
P3.11	RxD0	ASC0 receive data input (/output in synchronous mode)
P3.12	$\overline{BHE}$/$\overline{WRH}$	Byte high enable/write high output
P3.13	SCLK0	SSC shift clock input/output
P3.14	---	No pin assigned
P3.15	CLKOUT	System clock output (either prescaled or not through register XCLKOUTDIV)

Figure 38. Port 3 I/O and alternate functions



The structure of the Port 3 pins depends on their alternate function (see *Figure 39*).

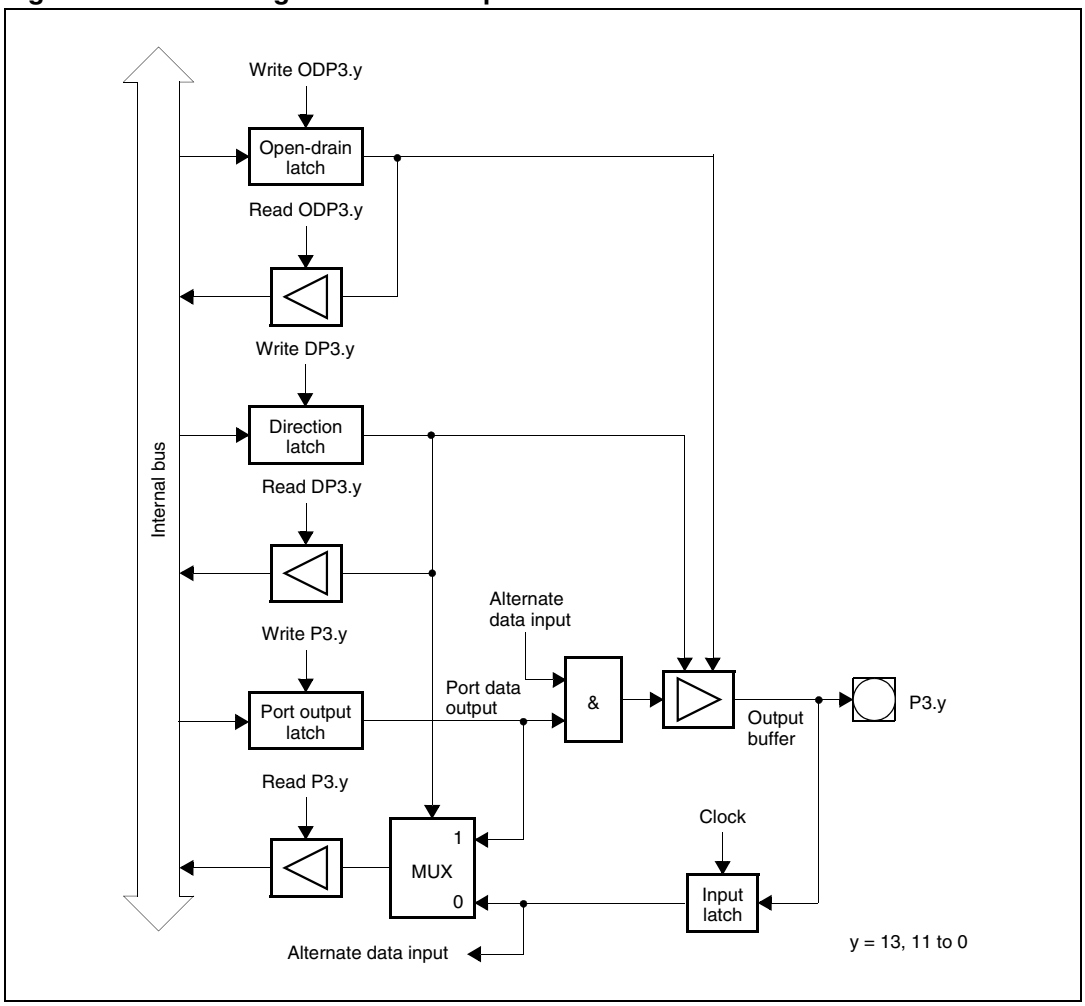
When the on-chip peripheral associated with a Port 3 pin is configured to use the alternate input function, it reads the input latch, which represents the state of the pin, via the line labeled ‘alternate data input’. Port 3 pins with alternate input functions are: T0IN, T2IN, T3IN, T4IN, T3EUD and CAPIN.

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 the port to output latch (P3.y = 1).If this is not done, 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, $\overline{\text{BHE}}$ 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: MTSR0, MRST0, RxD0 and SCLK0.

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

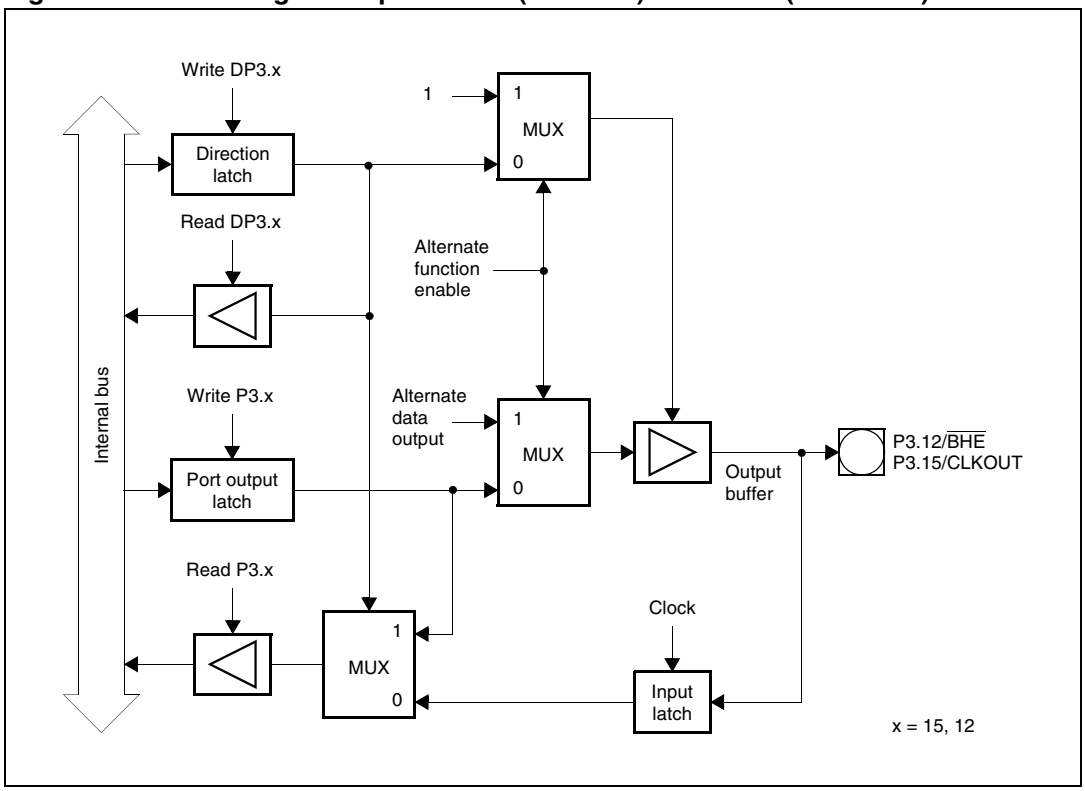
Figure 39. Block diagram of a Port 3 pin



Pin P3.12 ($\overline{BHE}$/$\overline{WRH}$) is another pin with an alternate output function. However, its structure is slightly different to the Port 3 pin (see *Figure 40*). After reset, the $\overline{BHE}$ or $\overline{WRH}$ function must be used. In either case, port latches cannot be programmed 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).

Note: Enabling the $\overline{BHE}$ or $\overline{WRH}$ 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. In this case, keep DP3.12 = 0 to ensure floating in hold mode.

Figure 40. Block diagram of pins P3.15 (CLKOUT) and P3.12 (BHE/WRH)



13.6 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.

13.6.1 Port 4 registers

P4 register

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

Table 81. P4 register description

Bit	Bit name	Function
7-0	P4.y	Port data register P4 bit y

Only bits 7 to 0 of the P4 register are implemented. All other bits are read as 0.

DP4 register

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

Table 82. DP4 register description

Bit	Bit name	Function
7-0	DP4.y	Port direction register DP4 bit y 0: Port line P4.y is an input (high impedance) 1: Port line P4.y is an output

Only bits 7 to 0 of the DP4 register are implemented. All other bits are read as 0.

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
-	-	-	-	-	-	-	-	OD P4.7	OD P4.6	OD P4.5	OD P4.4	-	-	-	-
-	-	-	-	-	-	-	-	RW	RW	RW	RW	-	-	-	-

Table 83. ODP4 register description

Bit	Bit name	Function
7-4	ODP4.y	Port open-drain control register ODP4 bit y 0: Port line P4.y output driver in push-pull mode 1: Port line P4.y output driver in open-drain mode if P4.y is not a segment address line output

Only bits 7 to 4 of the ODP4 register are implemented. All other bits are read as 0.

Note: When I^2C is enabled by setting the XPEN and XI2CEN bits of the SYSCON and XPERCON registers respectively, pins P4.4 and P4.7 become fully dedicated to the I^2C interface. All alternate functions are bypassed (external memory and CAN2 functions). The pins are also automatically configured as open-drain as requested by the I^2C bus standard. The Port 4 control registers P4, DP4 , and ODP4 can no longer control pins P4.7 and P4.4, as writing in the bits corresponding to P4.4 and P4.7 of these registers has no effect on pin behavior.

13.6.2 Alternate functions of Port 4

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

The number of segment address lines is selected via Port 0 during reset. The selected value can be read from the bit-field, SALSEL, in register RP0H (a read-only register) to check configuration during run time.

The CAN interfaces use two or four pins of Port 4 to interface the CAN module to the external CAN transceiver. In this case the number of possible segment address lines is reduced. The case is the same when the I^2C interface module is used.

Table 84 summarizes the alternate functions of Port 4 depending on the number of selected segment address lines (coded via bit-field SALSEL).

Table 84. 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(1)(2) 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/SCL	GPIO/CAN2_RxD/SCL	GPIO/CAN2_RxD/SCL	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/SDA	GPIO/CAN2_TxD/SDA	GPIO/CAN2_TxD/SDA	Segment address A23

1. When SALSEL = 10, CAN1 and CAN2 cannot be used and the external memory has a higher priority on the CAN alternate functions. Once the I^2C is enabled, P4.4 and P4.7 are dedicated to it and it has higher priority on the CAN alternate functions and on segment address functions.
2. If SALSEL= 10 and I^2C is enabled, P4.5 and P4.6 continue to output address lines.

Figure 41. Port 4 I/O and alternate functions

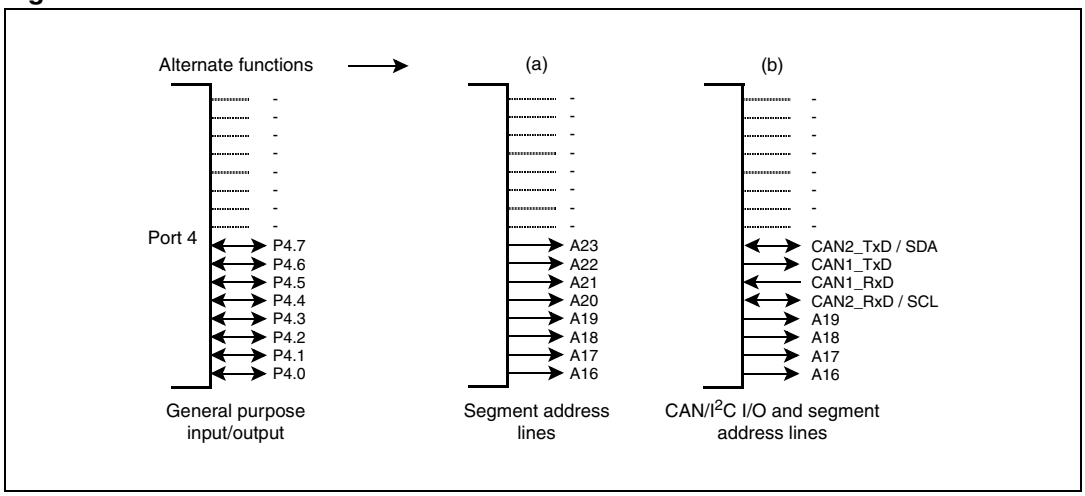


Figure 42. Block diagram of Port 4 pins 3 to 0

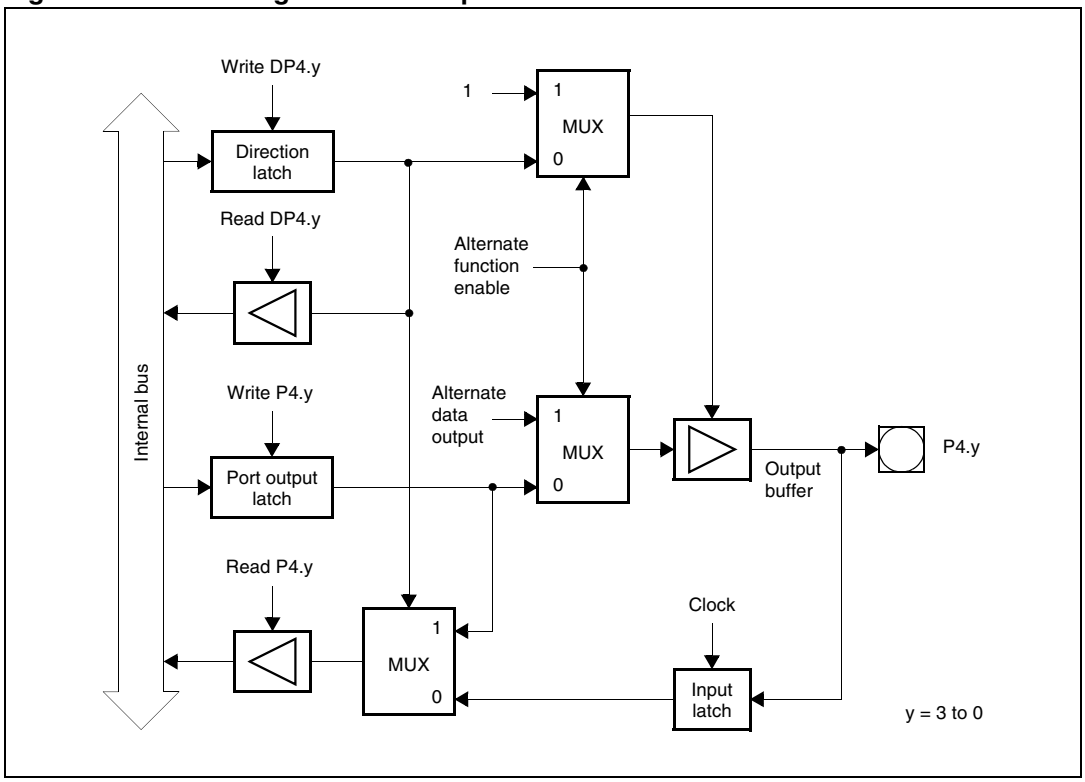
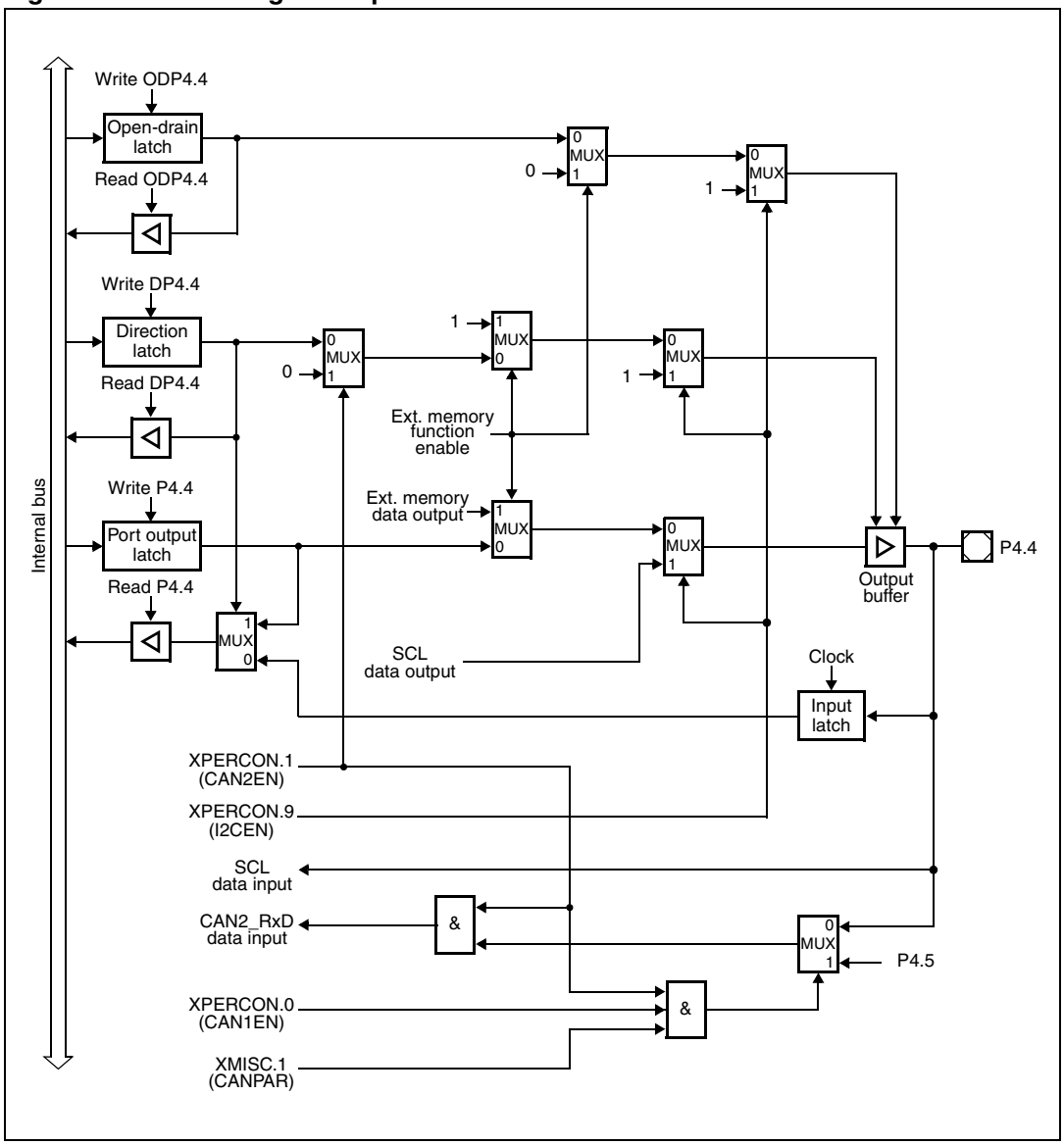
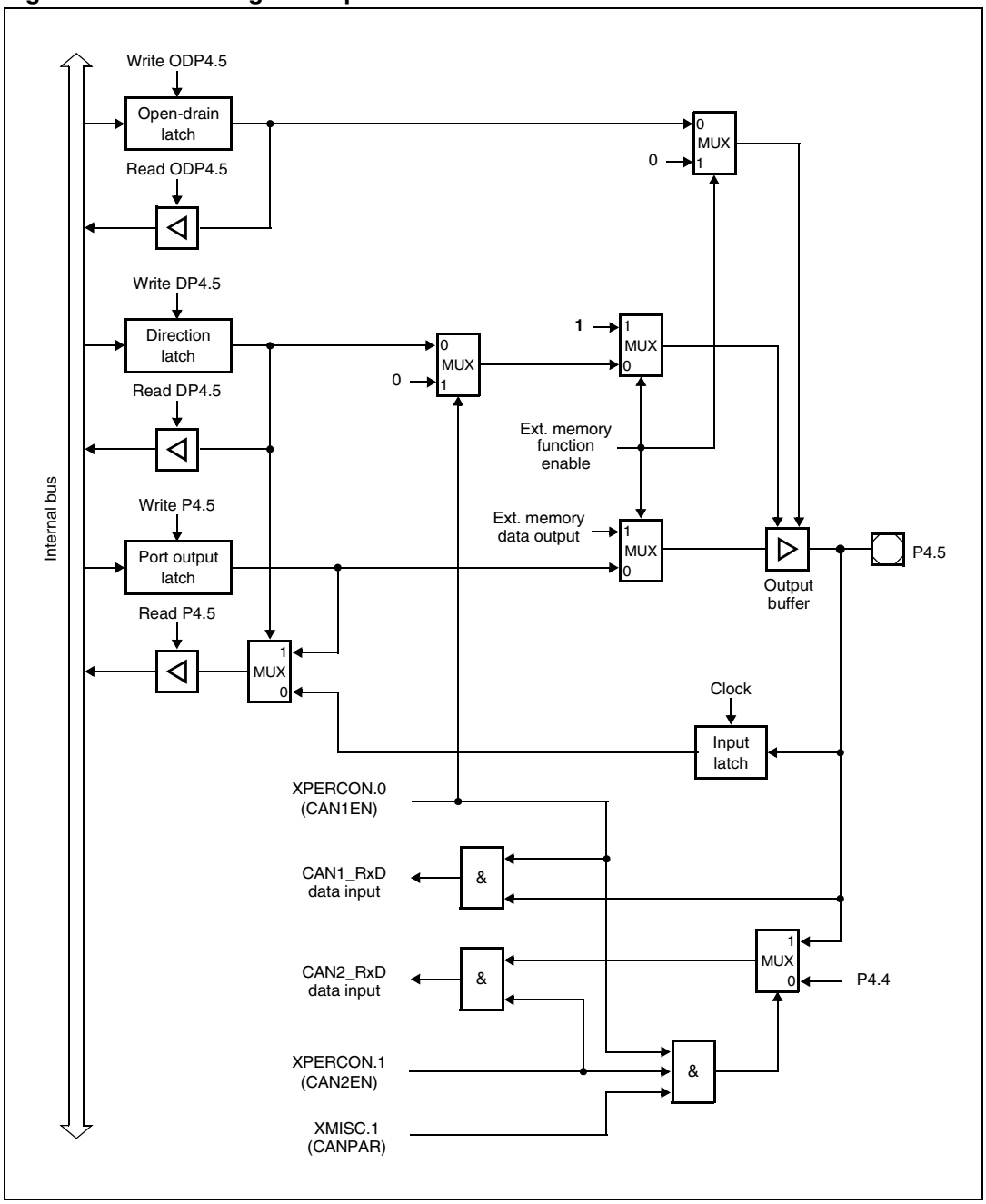


Figure 43. Block diagram of pin P4.4



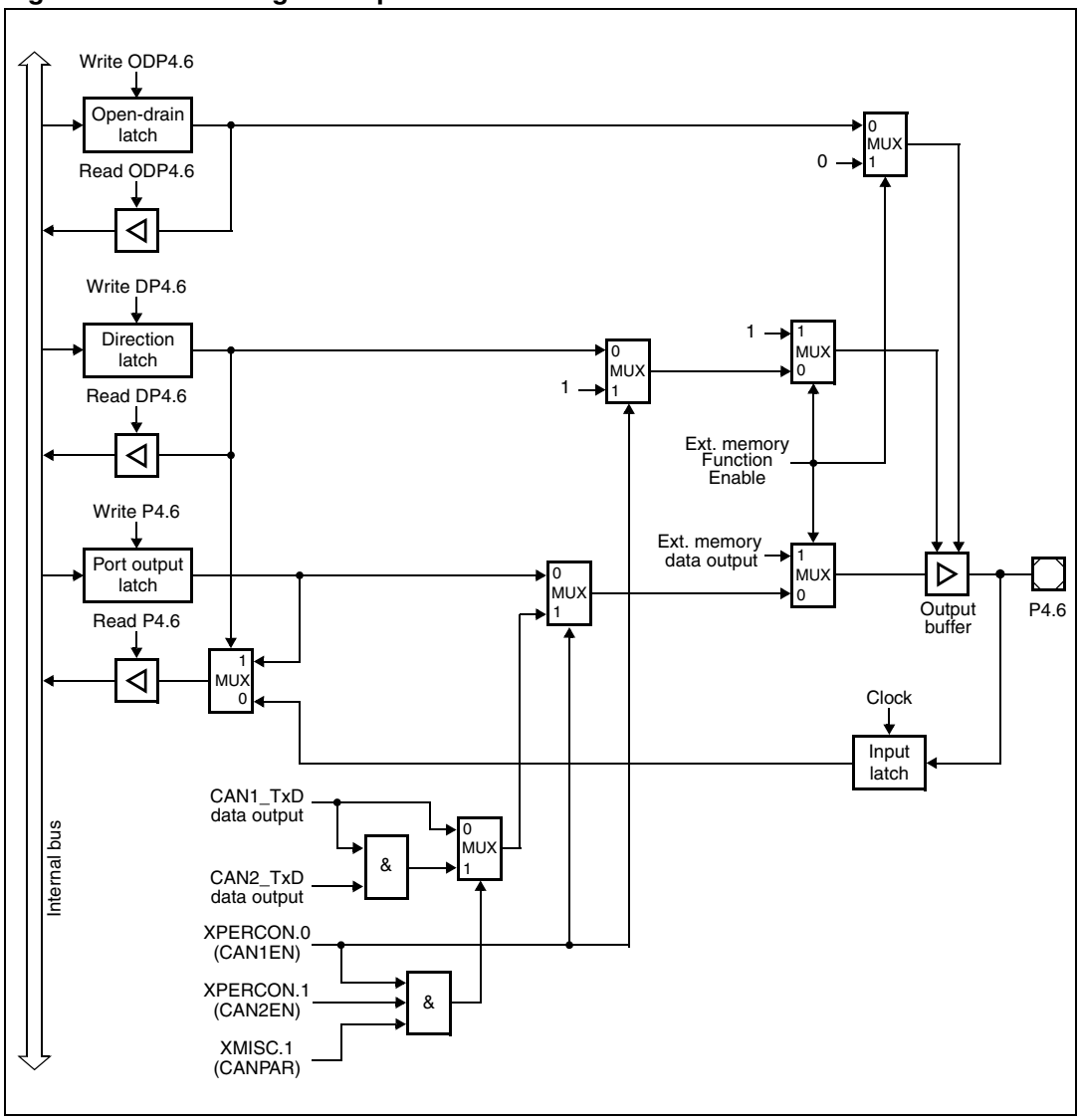
1. When SALSEL = 10, 8-bit segment address lines are selected and P4.4 outputs the address. Any attempt to use the CAN2 on P4.4 is masked. However, by enabling the I^2C, the segment function is masked, pin P4.4 is automatically configured as open-drain and used to input and output the SCL alternate function.
2. When CAN parallel mode is selected, CAN2_RxD is remapped on P4.5. This occurs only if CAN1 is also enabled. If CAN1 is disabled, no remapping occurs.

Figure 44. Block diagram of pin P4.5



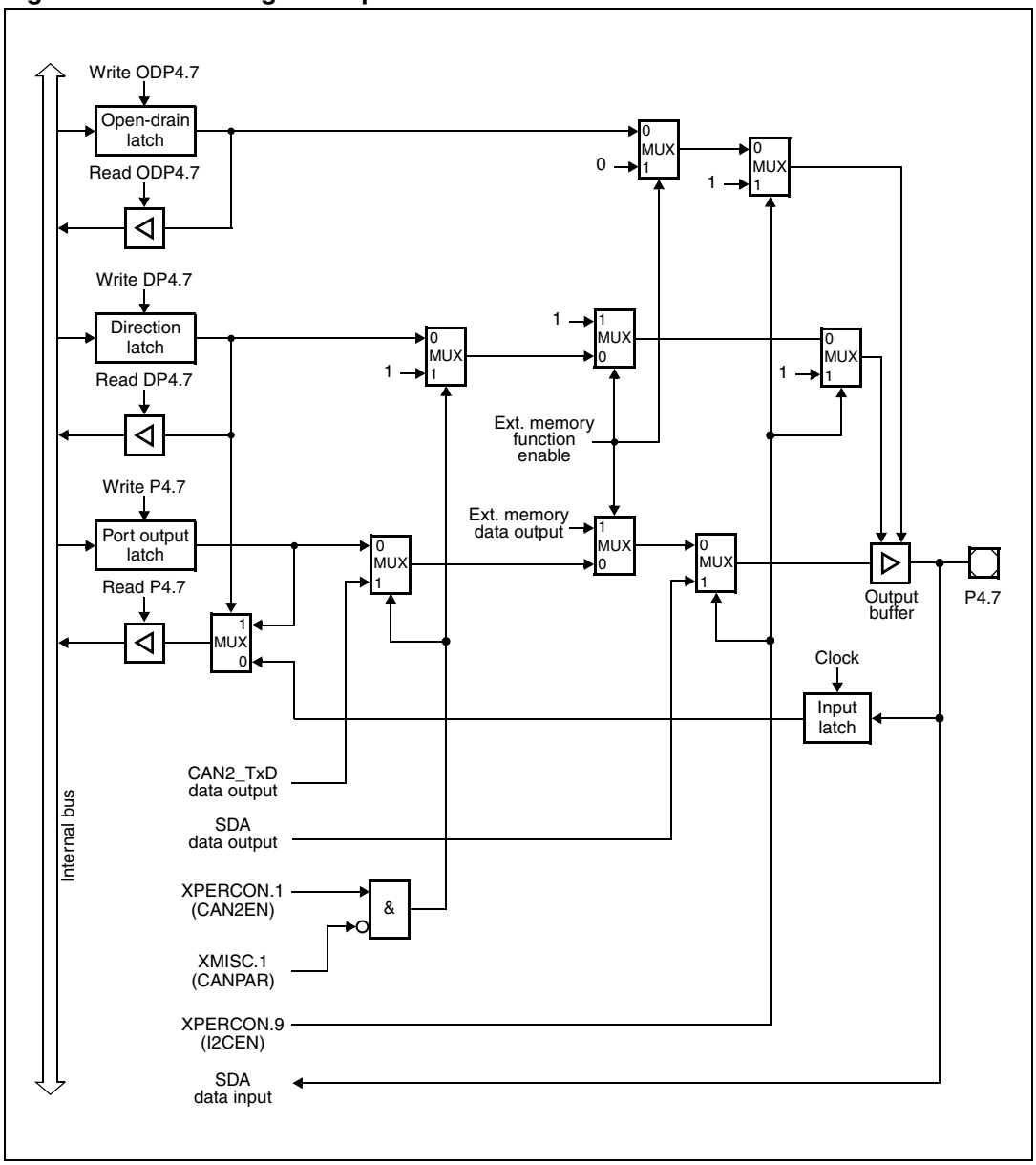
1. When SALSEL = 10, 8-bit segment address lines are selected and P4.5 outputs the address. Any attempt to use the CAN1 on P4.5 is masked.
2. When CAN parallel mode is selected, CAN2_RxD is remapped on P4.5. This occurs only if CAN1 is also enabled. If CAN1 is disabled, no remapping occurs.

Figure 45. Block diagram of pin P4.6



1. When SALSEL = 10, 8-bit segment address lines are selected and P4.6 outputs the address. Any attempt to use the CAN1 on P4.6 is masked.
2. When CAN parallel mode is selected, CAN2_TxD is remapped on P4.6. This occurs only if CAN1 is also enabled. If CAN1 is disabled, no remapping occurs.

Figure 46. Block diagram of pin P4.7



1. When SALSEL = 10, 8-bit segment address lines are selected and P4.7 outputs the address. Any attempt to use the CAN2 on P4.7 is masked. However, by enabling the I^2C, the segment function is masked, pin P4.7 is automatically configured as open-drain and used to input and output the SDA alternate function.
2. When CAN parallel mode is selected, CAN2_TxD is remapped on P4.6. This occurs only if CAN1 is also enabled. If CAN1 is disabled, no remapping occurs.

13.7 Port 5

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

13.7.1 Port 5 registers

P5 register

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

Table 85. P5 register description

Bit	Bit name	Function
15-0	P5.y	Port data register P5 bit y (read-only)

13.7.2 Alternate functions of port 5

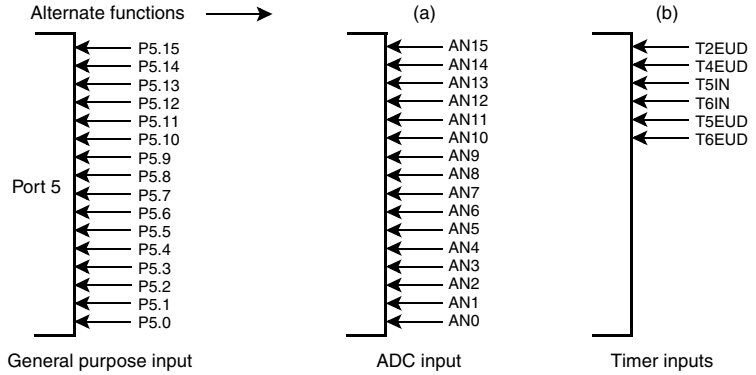
Each line of Port 5 is connected to the input multiplexer of the ADC. All port lines (P5.15 to P5.0) can accept analog signals (AN15 to AN0) which can then be converted by the ADC. No special programming is required for pins that are used as analog inputs. The upper 6 pins of Port 5 also serve as external timer control lines for GPT1 and GPT2.

Table 86 summarizes the alternate functions of Port 5.

Table 86. Port 5 alternate functions

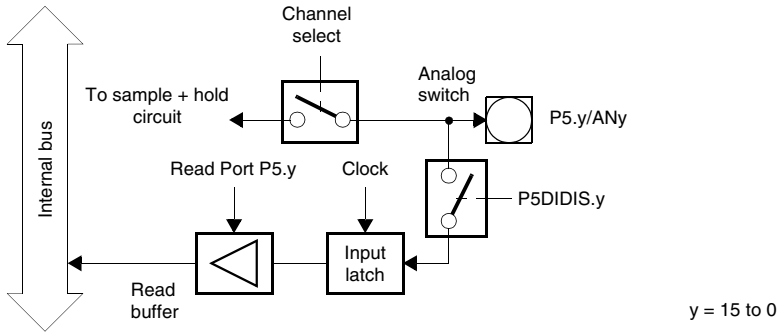
Port 5 pin	Alternate function (a)	Alternate function (b)
P5.0	Analog input AN0	-
P5.1	Analog input AN1	-
P5.2	Analog input AN2	-
P5.3	Analog input AN3	-
P5.4	Analog input AN4	-
P5.5	Analog input AN5	-
P5.6	Analog input AN6	-
P5.7	Analog input AN7	-
P5.8	Analog input AN8	-
P5.9	Analog input AN9	-
P5.10	Analog input AN10	T6EUD timer 6 external up/down input
P5.11	Analog input AN11	T5EUD timer 5 external up/down input
P5.12	Analog input AN12	T6IN timer 6 count input
P5.13	Analog input AN13	T5IN timer 5 count input
P5.14	Analog input AN14	T4EUD timer 4 external up/down input
P5.15	Analog input AN15	T2EUD timer 2 external up/down input

Figure 47. Port 5 I/O and alternate functions



Port 5 is an input only port where the analog input channels are directly connected to the pins rather than to the input latches. For these reasons, Port 5 pins have a special port structure (see *Figure 48*).

Figure 48. Block diagram of a Port 5 pin



13.7.3 Port 5 analog inputs disturb protection

A Schmitt trigger protection can be activated on each pin of Port 5 by setting the dedicated bit of register P5DIDIS. This allows the input leakage effect to be reduced.

P5DIDIS register

P5DIDIS (FFA4h/D2h) SFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P5DIDIS.15	P5DIDIS.14	P5DIDIS.13	P5DIDIS.12	P5DIDIS.11	P5DIDIS.10	P5DIDIS.9	P5DIDIS.8	P5DIDIS.7	P5DIDIS.6	P5DIDIS.5	P5DIDIS.4	P5DIDIS.3	P5DIDIS.2	P5DIDIS.1	P5DIDIS.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 87. P5DIDIS register description

Bit	Bit name	Function
15-0	P5DIDIS.y	Port 5 digital disable register bit y 0: Port line P5.y digital input is enabled (Schmitt trigger enabled) 1: Port line P5.y digital input is disabled (Schmitt trigger disabled)

13.8 Port 6

Port 6 is an 8-bit port. If it 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. In the ST10F296E, SSC1 is implemented on pins P6.5, P6.6, and P6.7. When the module is enabled through the XPERCON register, the corresponding bits P6, DP6 and ODP6 are overwritten by the new XSSCPORT register (mapped on the XBus). This allows the user to program pins P6.5, P6.6, and P6.7 according to the SSC1 configuration.

13.8.1 Port 6 registers

P6 register

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

Table 88. P6 register description

Bit	Bit name	Function
7-0	P6.y	Port data register P6 bit y

DP6 register

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

Table 89. DP6 register description

Bit	Bit name	Function
7-0	DP6.y	Port direction register DP6 bit y 0: Port line P6.y is an input (high impedance) 1: Port line P6.y is an output

ODP6 register

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

Table 90. ODP6 register description

Bit	Bit name	Function
7-0	ODP6.y	Port open-drain control register ODP6 bit y 0: Port line P6.y output driver in push-pull mode 1: Port line P6.y output driver in open-drain mode

XSSCPORT register

This register is enabled and visible only when the XPEN and XSSCEN bits of the SYSCON and XPERCON registers respectively are set. However, when SSC1 is disabled, P6, DP6 and ODP6 registers must be used to configure pins P6.2, P6.3, and P6.4.

XSSCPORT (E880h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	XODP 6.4	XP 6.4	XDP 6.4	XODP 6.3	XP 6.3	XDP 6.3	XODP 6.2	XP 6.2
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Table 91. ODP6 register description

Bit	Bit name	Function
7, 4, 1	XODP6.y	Port open-drain control register XODP6 bit y (y = 2, 3, and 4 only) 0: Port line P6.y output driver in push-pull mode 1: Port line P6.y output driver in open-drain mode
6, 3, 0	XP6.y	Port data register bit XP6 y (y = 2, 3, and 4 only)
5, 2	XDP6.y	Port direction register XDP6 bit y (y = 2, 3, and 4 only) 0: Port line P6.y is an input (high impedance) 1: Port line P6.y is an output

13.8.2 Alternate functions of Port 6

A programmable number of chip select signals (CS4 to CS0) derived from the bus control registers (BUSCON4 to BUSCON0) can be output on five pins of Port 6. The other three pins may be used for bus arbitration to accommodate additional masters in a ST10F296E system.

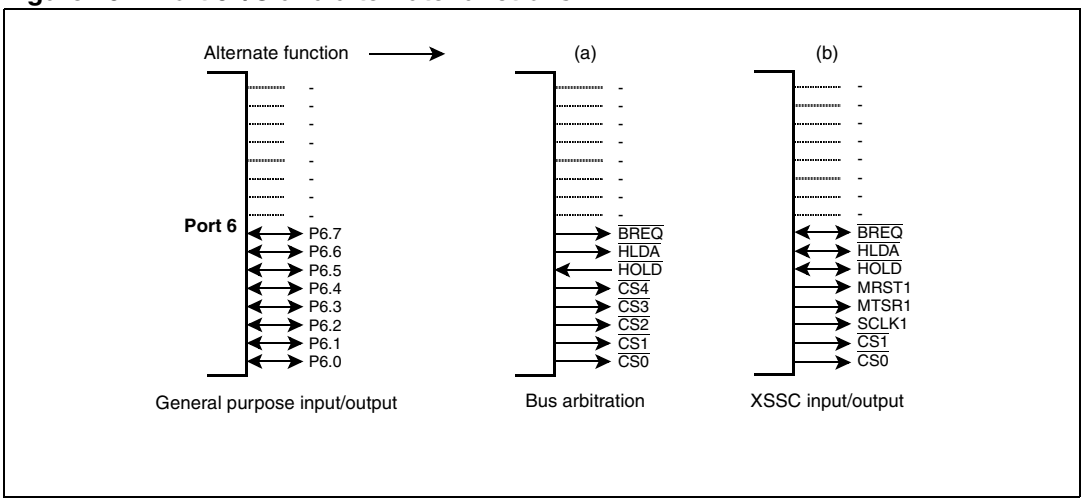
The number of chip select signals are selected via Port 0 during reset. The selected value can be read from bit-field CSSEL in the RP0H register to check the configuration during run time.

Table 92 summarizes the alternate functions of Port 6 depending on the number of chip select lines (coded via bit-field CSSEL) that are selected.

Table 92. Port 6 alternate functions

Port 6 pin	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 SCLK1	General purpose I/O SCLK1	Chip select $\overline{CS2}$ SCLK1	Chip select $\overline{CS2}$ SCLK1
P6.3	General purpose I/O MTSR1	General purpose I/O MTSR1	General purpose I/O MTSR1	Chip select $\overline{CS3}$ MTSR1
P6.4	General purpose I/O MRST1	General purpose I/O MRST1	General purpose I/O MRST1	Chip select $\overline{CS4}$ MRST1
P6.5	HOLD external hold request input			
P6.6	HLDA hold acknowledge output			
P6.7	$\overline{BREQ}$ bus request output			

Figure 49. Port 6 I/O and alternate functions



The chip select lines of Port 6 have an internal weak pull-up device. This device is switched on under the following conditions:

- During reset
- If Port 6 line is used as a chip select output, the ST10F296E is in hold mode, and the respective pin driver is in push-pull mode (ODP6.x = 0).

The pull-up device is implemented to drive the chip select lines high during reset to avoid multiple chip selection and to allow another master to access the external memory via the same chip select lines (AND-wired) while the ST10F296E is in hold mode.

When ODP6.x = 1 (open-drain output selected), the internal pull-up device is active during hold mode and external pull-up devices must be used in this case. When entering hold mode the $\overline{CS}$ lines are actively driven high for one clock phase, at which point the output level is controlled by the pull-up devices (if activated).

After reset the $\overline{CS}$ function must be used. In this case, the port latches cannot be programmed and the alternate function ($\overline{CS}$) is selected automatically.

Note: The open-drain output option can only be selected via software during the initialization routine. The CS0 signal is in push-pull output driver mode directly after reset (see Figure 50 on page 156).

The bus arbitration signals HOLD, HLDA and BREQ are selected with the HLDEN bit in the PSW register. When the bus arbitration signals are enabled via HLDEN, these pins are switched automatically to the appropriate direction. The pin drivers for HLDA and BREQ are automatically enabled while the pin driver for HOLD is automatically disabled (see *Figure 51 on page 157* and *Figure 52 on page 158*).

Figure 50. Block diagram of Port 6 pins 7, 6, 1, 0

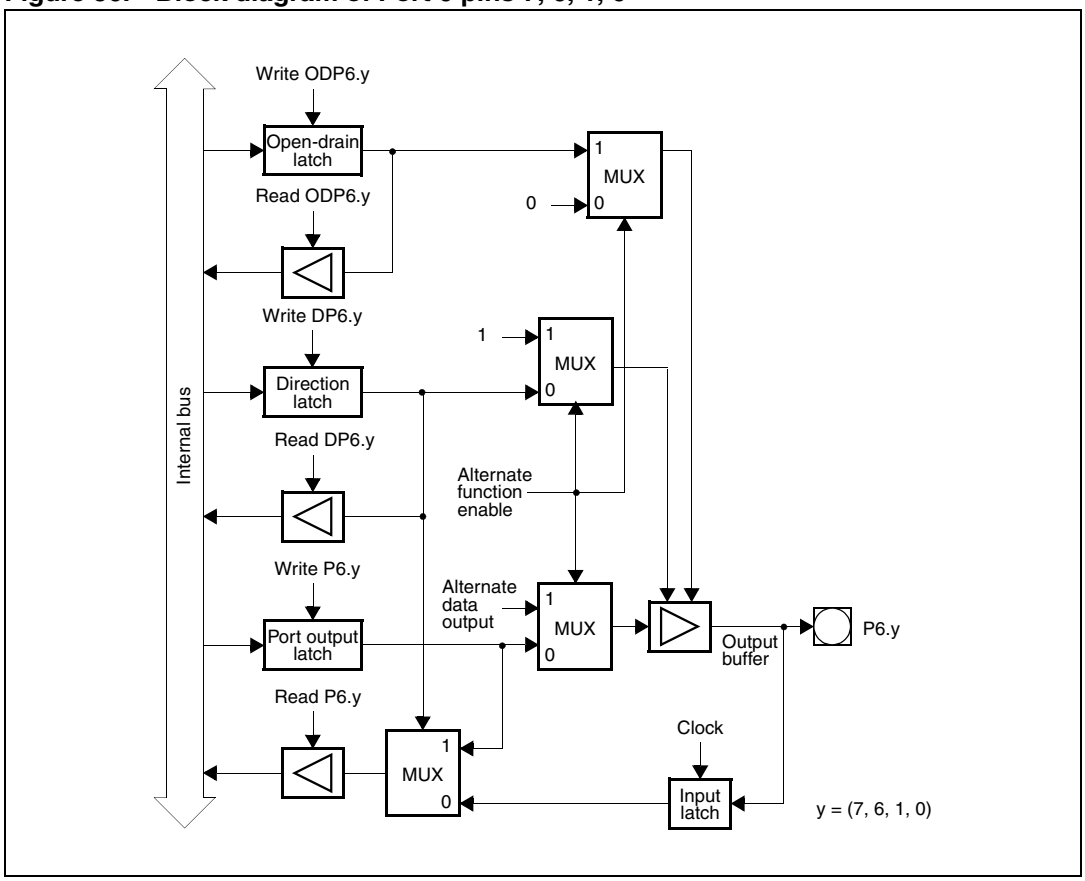


Figure 51. Block diagram of pin P6.5

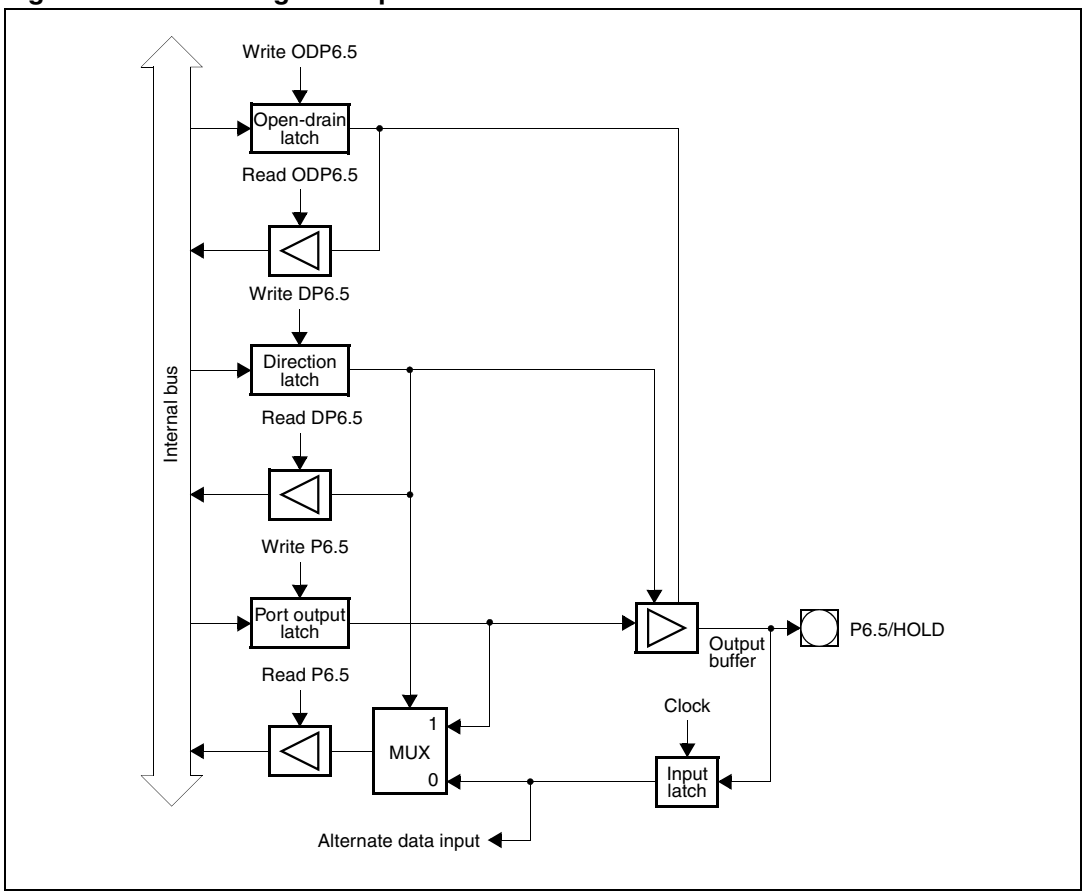
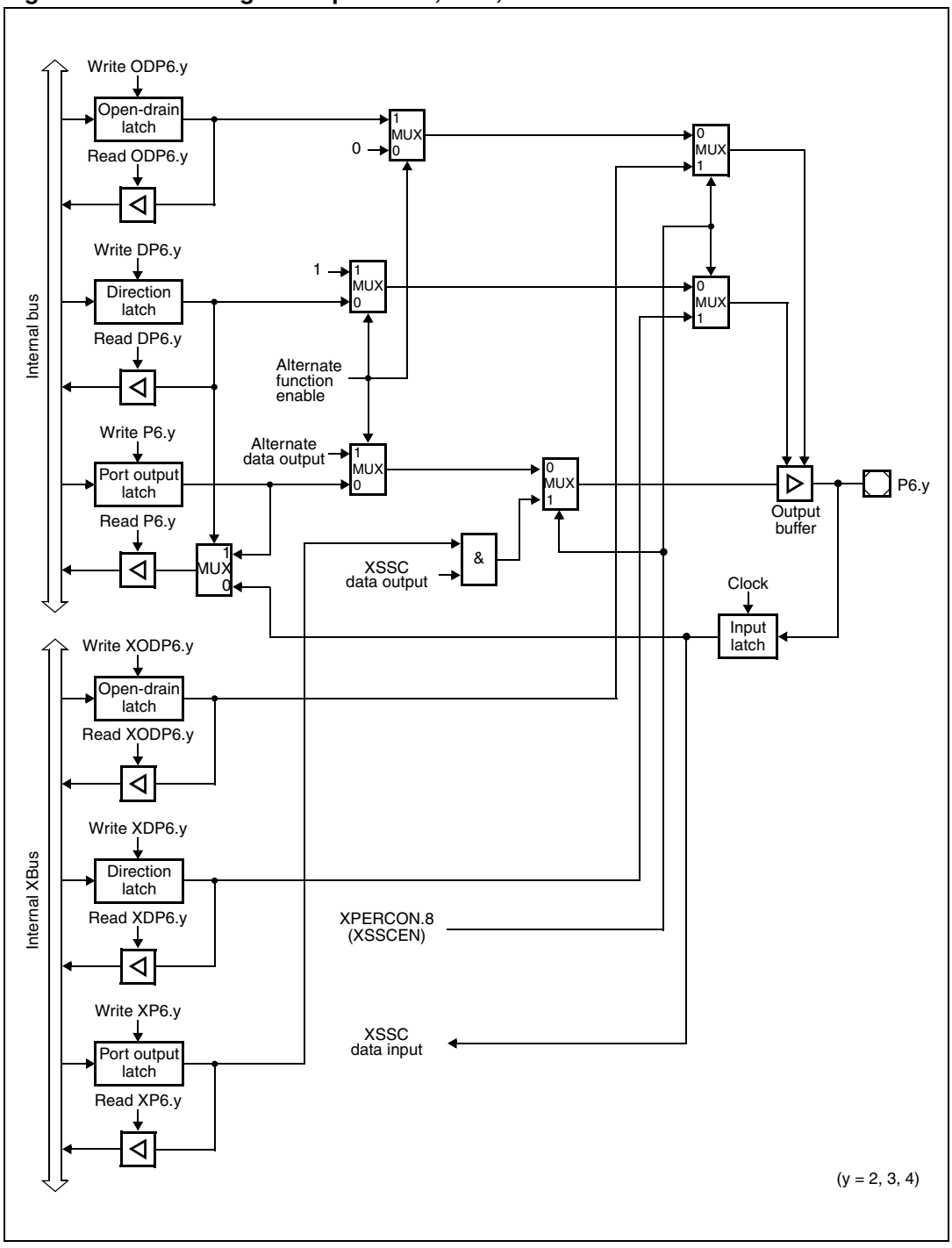


Figure 52. Block diagram of pins P6.2, P6.3, and P6.4



13.9 Port 7

This is an 8-bit port. If it 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.

13.9.1 Port 7 registers

P7 register

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

Table 93. P7 register description

Bit	Bit name	Function
7-0	P7.y	Port data register P7 bit y

DP7 register

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

Table 94. DP7 register description

Bit	Bit name	Function
7-0	DP7.y	Port direction register DP7 bit y 0: Port line P7.y is an input (high impedance) 1: Port line P7.y is an output

ODP7 register

ODP7 (F1D2h/E9h) ESFR Reset value: --00h

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

Table 95. ODP7 register description

Bit	Bit name	Function
7-0	ODP7.y	Port open-drain control register ODP7 bit y 0: Port line P7.y output driver in push-pull mode 1: Port line P7.y output driver in open-drain mode

13.9.2 Alternate functions of Port 7

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

Port 7 lines are connected to the CAPCOM2 unit and handled by software in a similar way to Port 2 lines (see *Section 13.4.2: Alternate functions of Port 2 on page 134*.

The capture input function of pins P7.7 to P7.4 can be used as external interrupt inputs with a sample rate of eight CPU clock cycles.

The lower 4 lines of Port 7 (P7.3 to P7.0) supports outputs of the PWM module (POUT3 to POUT0). At these pins, the value of the respective port output latch is XORed with the value of the PWM output rather than ANDed. This allows the alternate output value to be used as it is (port latch holds a 0) or to be inverted at the pin (port latch holds a 1).

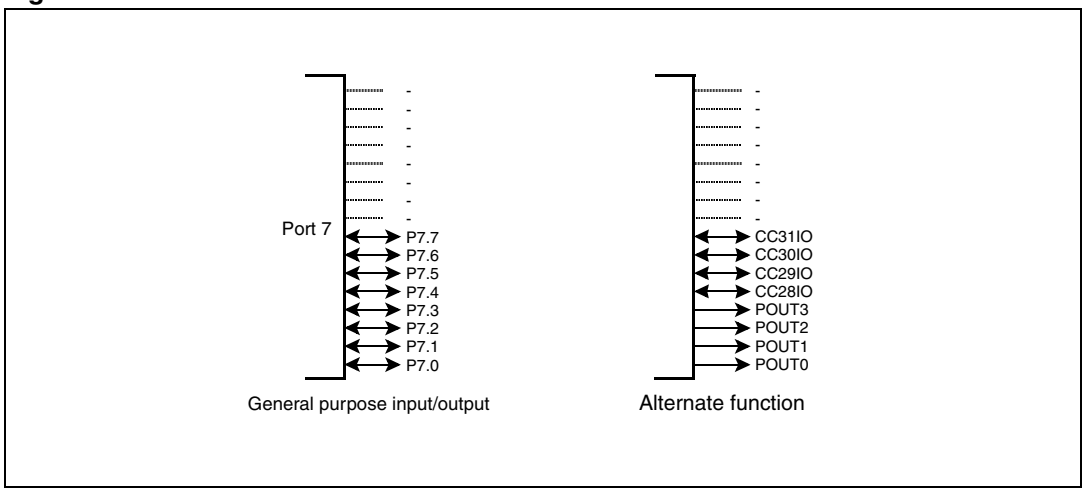
The PWM outputs must be enabled via the respective PENx bit in the PWMCON1 register.

Table 96 summarizes the alternate functions of Port 7.

Table 96. Port 7 alternate functions

Port 7 pin	Alternate function	
P7.0	POUT0	PWM mode channel 0 output
P7.1	POUT1	PWM mode channel 1 output
P7.2	POUT2	PWM mode channel 2 output
P7.3	POUT3	PWM mode channel 3 output
P7.4	CC28 I/O	Capture input/compare output channel 28
P7.5	CC29 I/O	Capture input/compare output channel 29
P7.6	CC30 I/O	Capture input/compare output channel 30
P7.7	CC31 I/O	Capture input/compare output channel 31

Figure 53. Port 7 I/O and alternate functions



The structure of Port 7 differs from the other ports in the way the output latches are connected to the internal bus and to the pin driver (see *Figure 54 on page 161* and *Figure 55 on page 162*).

Pins P7.3 to P7.0 (POUT3 to POUT0) XOR the alternate data output with the port latch output. This allows alternate data to be used directly or inverted at the pin driver.

Pins P7.7 to P7.4 (CC31IO to CC28IO) combine internal bus data and alternate data output before the port latch input.

Figure 54. Block diagram of Port 7 pins 3 to 0

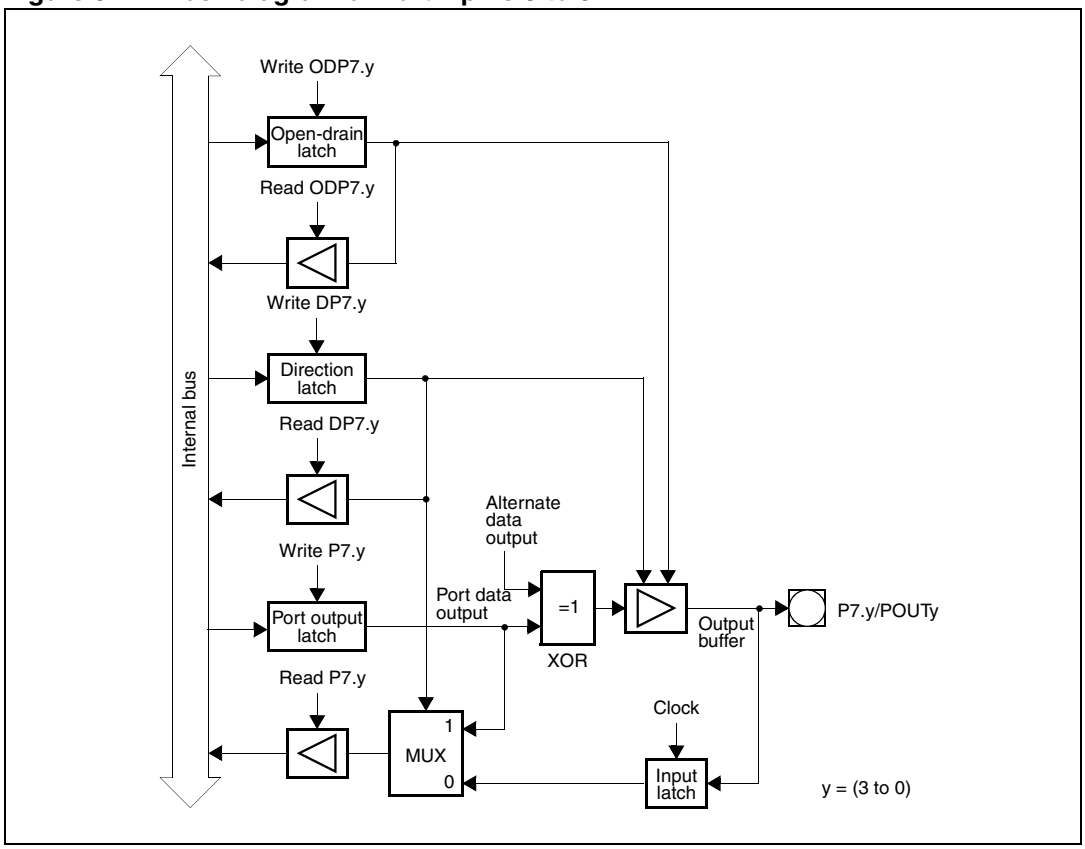
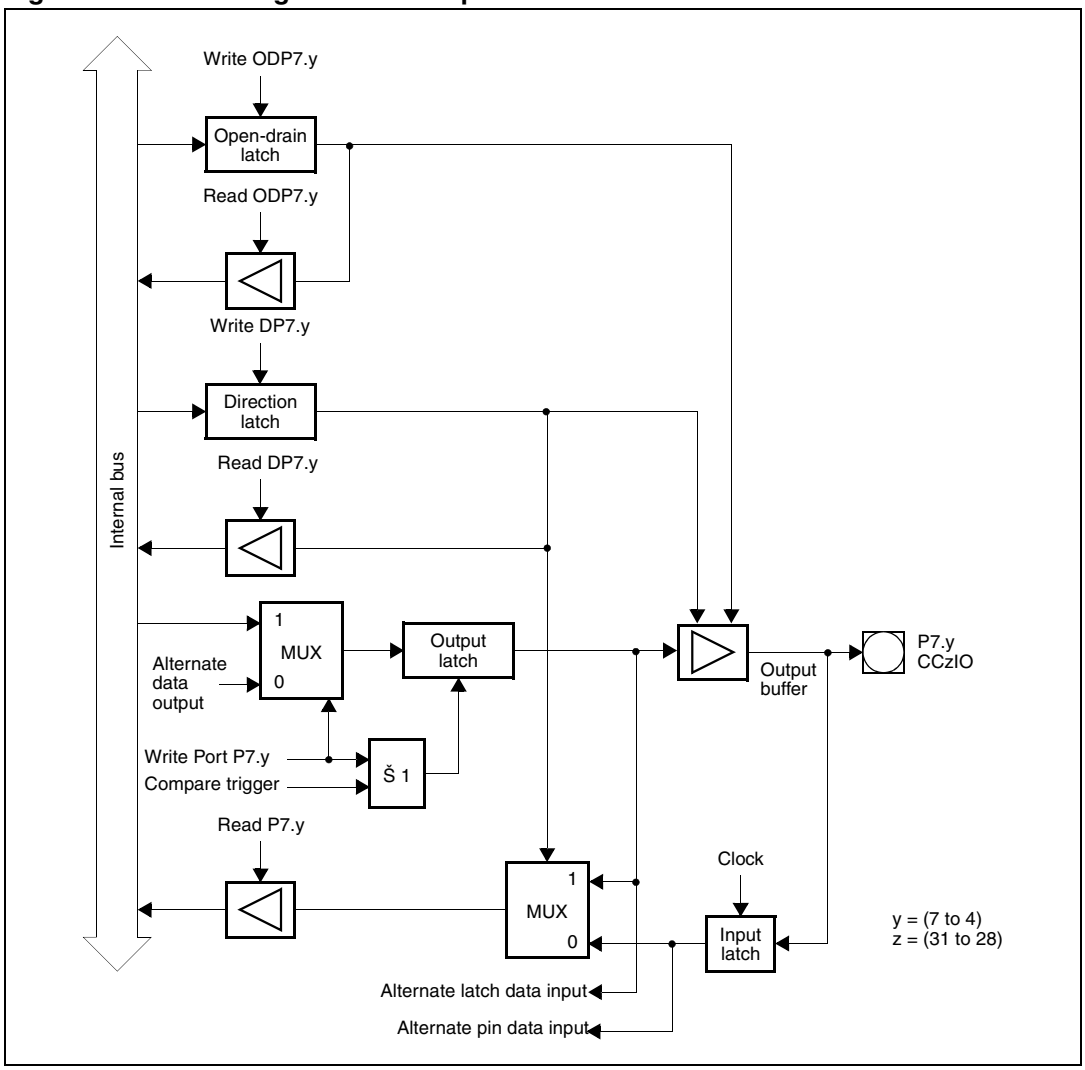


Figure 55. Block diagram of Port 7 pins 7 to 4



13.10 Port 8

This is an 8-bit port. If it 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.

In the ST10F296E, XASC (or ASC1) is implemented on pins P8.6 and P8.7. When the module is enabled through the XPERCON register, the corresponding bits P8, DP8 and ODP8 are overwritten by the new XS1PORT register (mapped on the XBus). This allows the user to program pins P8.6 and P8.7 according to the ASC1 configuration.

13.10.1 Port 8 registers

P8 register

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

Table 97. P8 register description

Bit	Bit name	Function
7-0	P8.y	Port data register P8 bit y

DP8 register

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

Table 98. DP8 register description

Bit	Bit name	Function
7-0	DP8.y	Port direction register DP8 bit y 0: Port line P8.y is an input (high impedance) 1: Port line P8.y is an output

ODP8 register

ODP8 (F1D6h/EBh) ESFR Reset value: --00h

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

Table 99. ODP8 register description

Bit	Bit name	Function
7-0	ODP8.y	Port open-drain control register ODP8 bit y 0: Port line P8.y output driver in push-pull mode 1: Port line P8.y output driver in open-drain mode

XS1PORT register

This register is enabled and visible only when the XPEN and XASCEN bits of the SYSCON and XPERCON registers respectively are set. However, when ASC1 is disabled, the standard P8, DP8 and ODP8 registers must be used to configure pins P8.6 and P8.7.

XS1PORT (E980h) XBus Reset value: 0000h

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

Table 100. XS1PORT register description

Bit	Bit name	Function
5, 2	XODP8.y	Port open-drain control register bit y (y = 6, 7 only) 0: Port line P8.y output driver in push-pull mode 1: Port line P8.y output driver in open-drain mode
4, 1	XP8.y	Port data register bit y (y = 6, 7 only)
3, 0	XDP8.y	Port direction register bit y (y = 6, 7 only) 0: Port line P8.y is an input (high impedance) 1: Port line P8.y is an output

13.10.2 Alternate functions of Port 8

All Port 8 lines (P8.7 to P8.0) support capture inputs or compare outputs (CC23IO to CC16IO) for the CAPCOM2 unit (see *Table 101*). See *Section 13.4.2: Alternate functions of Port 2 on page 134* for the use of the port lines by the CAPCOM unit, its accessibility via software, and precautions, all of which are the same as described for Port 2 lines.

The capture input function of pins P8.7 to P8.0 can be used as external interrupt inputs with a sample rate of eight CPU clock cycles. The pins of Port 8 combine internal bus data and alternate data output before the port latch input.

Table 101. Port 8 alternate functions

Port 8 pin	Alternate function (a)		Alternate function (b)	
P8.0	CC16IO	Capture input/compare output ch. 16	-	
P8.1	CC17IO	Capture input/compare output ch. 17	-	
P8.2	CC18IO	Capture input/compare output ch. 18	-	
P8.3	CC19IO	Capture input/compare output ch. 19	-	
P8.4	CC20IO	Capture input/compare output ch. 20	-	
P8.5	CC21IO	Capture input/compare output ch. 21	-	
P8.6	CC22IO	Capture input/compare output ch. 22	RxD1	XASC receive data input/output
P8.7	CC23IO	Capture input/compare output ch. 23	TxD1	XASC transmit data output

Figure 56. Port 8 I/O and alternate functions

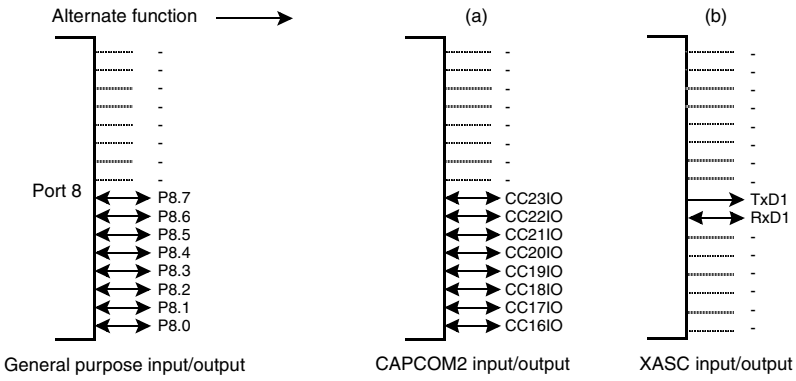


Figure 57. Block diagram of P8 pins 5 to 0

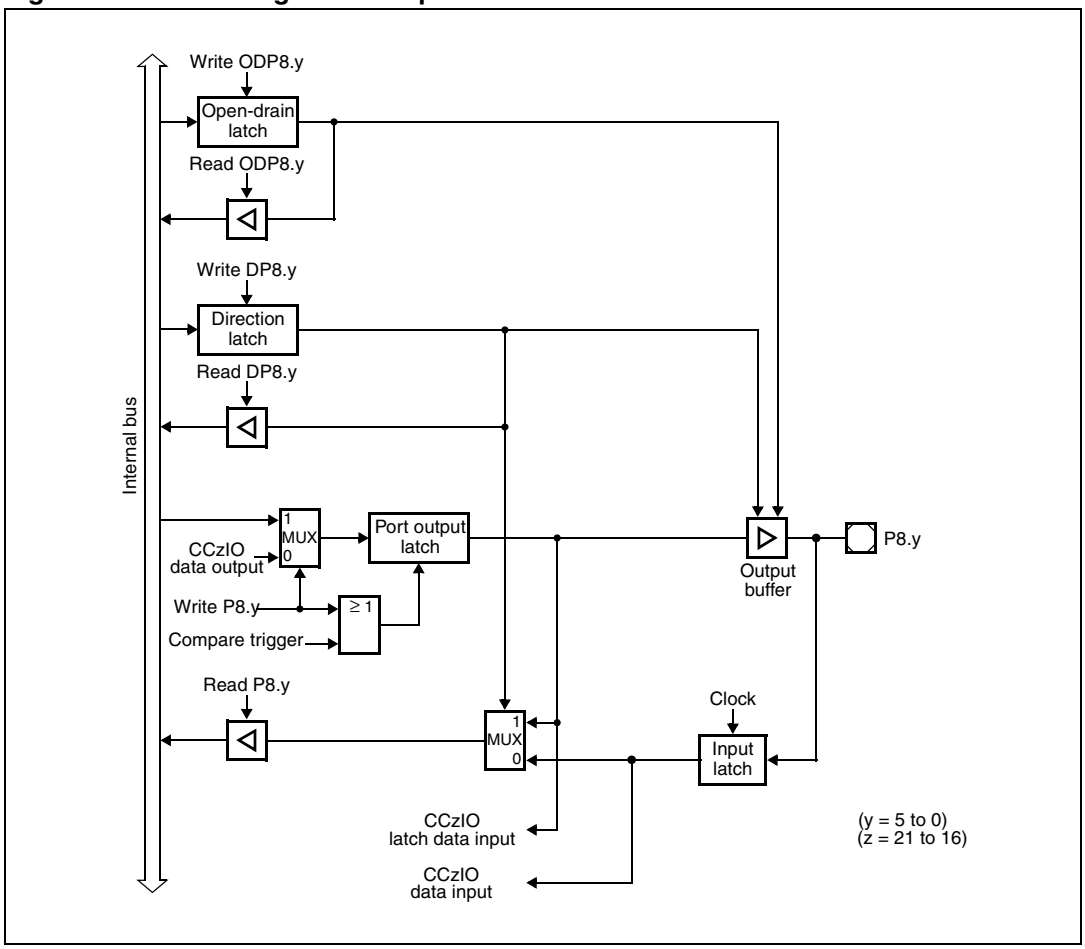


Figure 58. Block diagram of pin P8.6

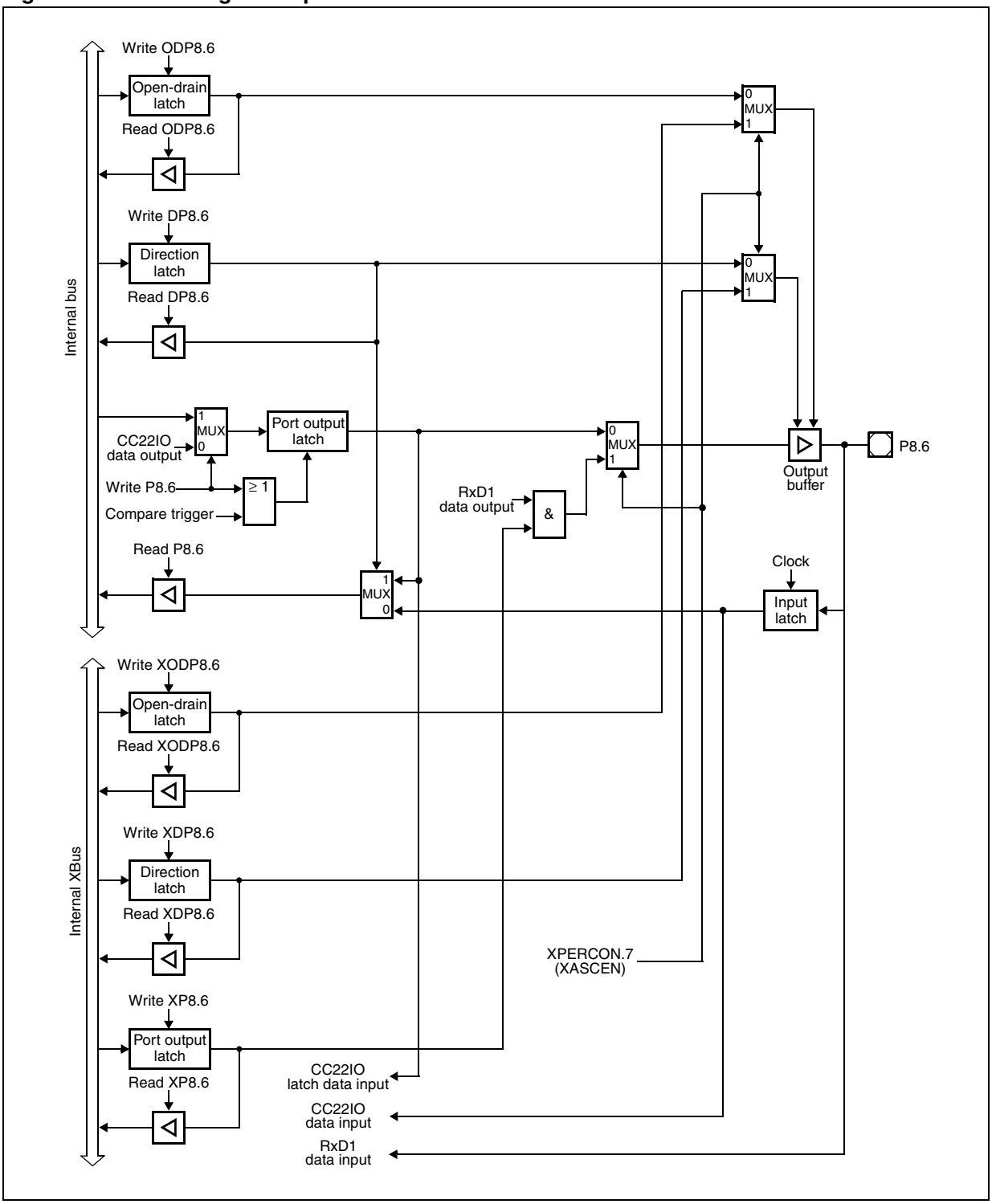
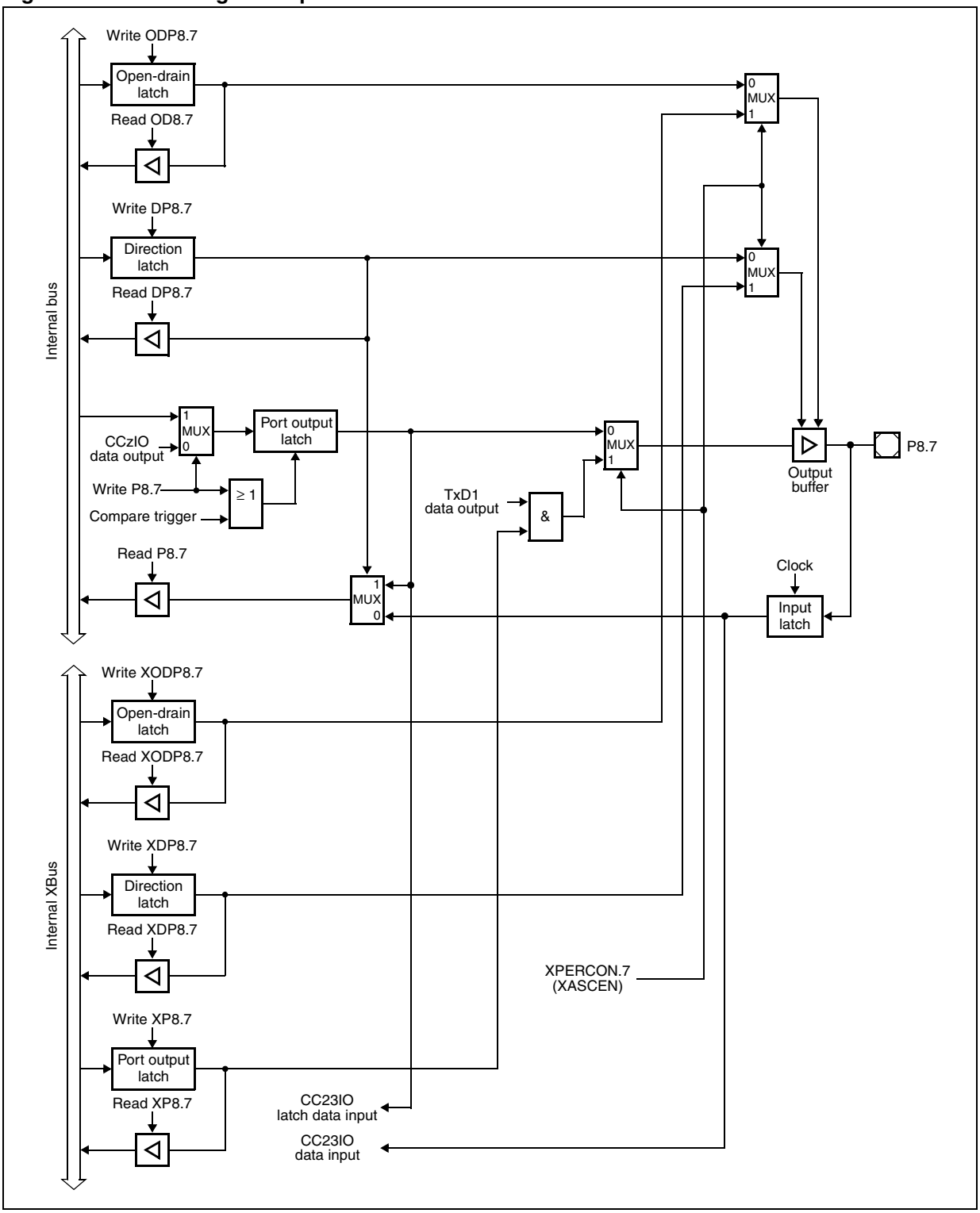


Figure 59. Block diagram of pin P8.7



13.11 XPort 9

XPort 9 is enabled by setting the XPEN and XPORTEN bits of the SYSCON and XPERCON registers respectively. On the XBus interface, the registers are not bit-addressable.

This 16-bit port is used for general purpose I/O. The direction of each line can be configured via the corresponding direction register XDP9. Each port line can be switched into push-pull or open-drain output mode via the open-drain control register XODP9. The port lines can also be switeched into TTL/CMOS input through the input threshold control register XPICON9 (*Section 13.1.2: Input threshold control on page 121*).

All port lines can be individually (bit-wise) programmed. The 'bit-addressable' feature is available via specific 'set' and 'clear' registers: XP9SET, XP9CLR, XDP9SET, XDP9CLR, XODP9SET, XODP9CLR, XPICON9SET, and XPICON9CLR.

13.11.1 XPort 9 registers

XP9 register

XP9 (EB80h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9.15	XP9.14	XP9.13	XP9.12	XP9.11	XP9.10	XP9.9	XP9.8	XP9.7	XP9.6	XP9.5	XP9.4	XP9.3	XP9.2	XP9.1	XP9.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 102. XP9 register description

Bit	Bit name	Function
15-0	XP9.y	Port data register XP9 bit y

XP9SET register

XP9SET (EB82h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9SET.15	XP9SET.14	XP9SET.13	XP9SET.12	XP9SET.11	XP9SET.10	XP9SET.9	XP9SET.8	XP9SET.7	XP9SET.6	XP9SET.5	XP9SET.4	XP9SET.3	XP9SET.2	XP9SET.1	XP9SET.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 103. XP9SET register description

Bit	Bit name	Function
15-0	XP9SET.y	Writing a 1 sets the corresponding bit of the XP9.y register. Writing a 0 has no effect.

XP9CLR register

XP9CLR (EB84h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP9CLR.15	XP9CLR.14	XP9CLR.13	XP9CLR.12	XP9CLR.11	XP9CLR.10	XP9CLR.9	XP9CLR.8	XP9CLR.7	XP9CLR.6	XP9CLR.5	XP9CLR.4	XP9CLR.3	XP9CLR.2	XP9CLR.1	XP9CLR.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 104. XP9CLR register description

Bit	Bit name	Function
15-0	XP9CLR.y	Writing a 1 clears the corresponding bit of the XP9.y register. Writing a 0 has no effect.

XDP9 register

XDP9 (EB86h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XDP9.15	XDP9.14	XDP9.13	XDP9.12	XDP9.11	XDP9.10	XDP9.9	XDP9.8	XDP9.7	XDP9.6	XDP9.5	XDP9.4	XDP9.3	XDP9.2	XDP9.1	XDP9.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 105. XDP9 register description

Bit	Bit name	Function
15-0	XDP9.y	Port direction register XDP9 bit y 0: Port line XP9.y is an input (high impedance) 1: Port line XP9.y is an output

XDP9SET register

XDP9SET (EB88h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XDP9SET.15	XDP9SET.14	XDP9SET.13	XDP9SET.12	XDP9SET.11	XDP9SET.10	XDP9SET.9	XDP9SET.8	XDP9SET.7	XDP9SET.6	XDP9SET.5	XDP9SET.4	XDP9SET.3	XDP9SET.2	XDP9SET.1	XDP9SET.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 106. XDP9SET register description

Bit	Bit name	Function
15-0	XDP9SET.y	Writing a 1 sets the corresponding bit of the XDP9.y register. Writing a 0 has no effect.

XDP9CLR register

XDP9CLR (EB8Ah) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XDP9CLR.15	XDP9CLR.14	XDP9CLR.13	XDP9CLR.12	XDP9CLR.11	XDP9CLR.10	XDP9CLR.9	XDP9CLR.8	XDP9CLR.7	XDP9CLR.6	XDP9CLR.5	XDP9CLR.4	XDP9CLR.3	XDP9CLR.2	XDP9CLR.1	XDP9CLR.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 107. XDP9CLR register description

Bit	Bit name	Function
15-0	XDP9CLR.y	Writing a 1 clears the corresponding bit of the XDP9.y register. Writing a 0 has no effect.

XODP9 register

XODP9 (EB8Ch) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XODP9.15	XODP9.14	XODP9.13	XODP9.12	XODP9.11	XODP9.10	XODP9.9	XODP9.8	XODP9.7	XODP9.6	XODP9.5	XODP9.4	XODP9.3	XODP9.2	XODP9.1	XODP9.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 108. XODP9 register description

Bit	Bit name	Function
15-0	XODP9.y	Port open-drain control register XODP9 bit y 0: Port line XP9.y output driver in push-pull mode 1: Port line XP9.y output driver in open-drain mode

XODP9SET register

XODP9SET (EB8Eh) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XODP9SET.15	XODP9SET.14	XODP9SET.13	XODP9SET.12	XODP9SET.11	XODP9SET.10	XODP9SET.9	XODP9SET.8	XODP9SET.7	XODP9SET.6	XODP9SET.5	XODP9SET.4	XODP9SET.3	XODP9SET.2	XODP9SET.1	XODP9SET.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 109. XODP9SET register description

Bit	Bit name	Function
15-0	XODP9SET.y	Writing a 1 sets the corresponding bit of the XODP9.y register. Writing a 0 has no effect.

XODP9CLR register

XODP9CLR (EB90h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XODP9CLR.15	XODP9CLR.14	XODP9CLR.13	XODP9CLR.12	XODP9CLR.11	XODP9CLR.10	XODP9CLR.9	XODP9CLR.8	XODP9CLR.7	XODP9CLR.6	XODP9CLR.5	XODP9CLR.4	XODP9CLR.3	XODP9CLR.2	XODP9CLR.1	XODP9CLR.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 110. XODP9CLR register description

Bit	Bit name	Function
15-0	XODP9CLR.y	Writing a 1 clears the corresponding bit of the XODP9.y register. Writing a 0 has no effect.

13.12 XPort 10

XPort 10 is enabled by setting the XPEN and XPORT10EN/XPORT9EN bits of the SYSCON and XPERCON registers respectively. On the XBus interface, the register are not bit-addressable. This 16-bit input port can only read data. There is no output latch and no direction register. Data written to XP10 are lost.

13.12.1 XPort 10 registers

XP10 register

XP10 (EBC0h) XBus Reset value: XXXXh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP10.15	XP10.14	XP10.13	XP10.12	XP10.11	XP10.10	XP10.9	XP10.8	XP10.7	XP10.6	XP10.5	XP10.4	XP10.3	XP10.2	XP10.1	XP10.0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 111. XP10 register description

Bit	Bit name	Function
15-0	XP10.y	Port data register XP10 bit y

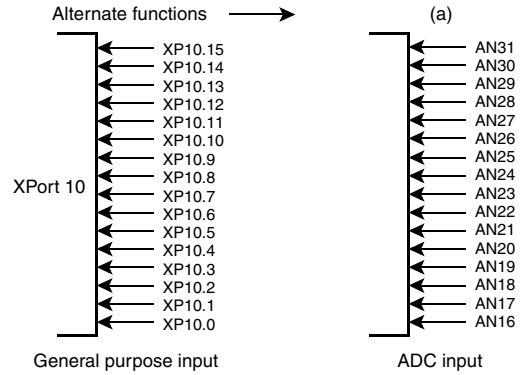
13.12.2 Alternate functions of XPort 10

Each line of XPort 10 is also connected to a multiplexer of the ADC. All port lines (XP10.15 to XP10.0) can accept analog signals (AN31 to AN16) that can be converted by the ADC. No special programming is required for pins that are used as analog inputs. *Table 112* summarizes the alternate functions of XPort 10.

Table 112. XPort 10 alternate functions

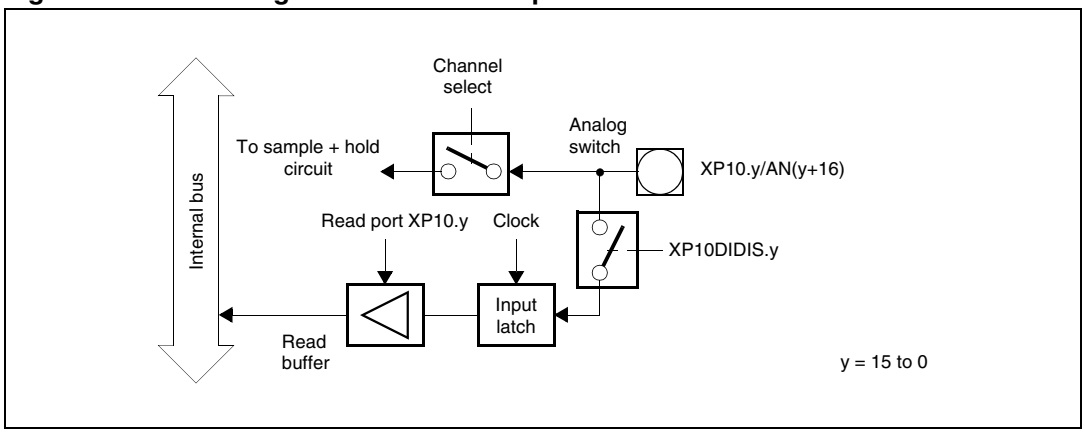
XPort 10 pin	Alternate function
XP10.0	Analog input AN16
XP10.1	Analog input AN17
XP10.2	Analog input AN18
XP10.3	Analog input AN19
XP10.4	Analog input AN20
XP10.5	Analog input AN21
XP10.6	Analog input AN22
XP10.7	Analog input AN23
XP10.8	Analog input AN24
XP10.9	Analog input AN25
XP10.10	Analog input AN26
XP10.11	Analog input AN27
XP10.12	Analog input AN28
XP10.13	Analog input AN29
XP10.14	Analog input AN30
XP10.15	Analog input AN31

Figure 60. XPort 10 I/O and alternate functions



XPort 10 pins have a special port structure (see *Figure 61*) because the port is input only and because the analog input channels are directly connected to the pins rather than to the input latches.

Figure 61. Block diagram of an XPort 10 pin



13.12.3 XPort 10 analog inputs disturb protection

The XP10DIDIS, XP10DIDISSET, and XP10DIDISCLR registers are provided for additional disturb protection support on the analog inputs. Once one bit of any of the registers is set, the corresponding pin can no longer be used as general purpose input.

XP10DIDIS register

XP10DIDIS (EBD2h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP10DIDIS.15	XP10DIDIS.14	XP10DIDIS.13	XP10DIDIS.12	XP10DIDIS.11	XP10DIDIS.10	XP10DIDIS.9	XP10DIDIS.8	XP10DIDIS.7	XP10DIDIS.6	XP10DIDIS.5	XP10DIDIS.4	XP10DIDIS.3	XP10DIDIS.2	XP10DIDIS.1	XP10DIDIS.0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 113. XP10DIDIS register description

Bit	Bit name	Function
15-0	XP10DIDIS.y	XPort 10 digital disable register bit y 0: Port line XP10.y digital input is enabled (Schmitt trigger enabled) 1: Port line XP10.y digital input is disabled (Schmitt trigger disabled, necessary for input leakage current reduction).

XP10DIDISSET register

XP10DIDISSET (EBD4h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP10DIDISSET.15	XP10DIDISSET.14	XP10DIDISSET.13	XP10DIDISSET.12	XP10DIDISSET.11	XP10DIDISSET.10	XP10DIDISSET.9	XP10DIDISSET.8	XP10DIDISSET.7	XP10DIDISSET.6	XP10DIDISSET.5	XP10DIDISSET.4	XP10DIDISSET.3	XP10DIDISSET.2	XP10DIDISSET.1	XP10DIDISSET.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 114. XP10DIDISSET register description

Bit	Bit name	Function
15-0	XP10DIDISSET.y	Writing a 1 sets the corresponding bit of the XP10DIDIS.y register. Writing a 0 has no effect.

XP10DIDISCLR register

XP10DIDISCLR (EBD6h) XBus Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XP10DIDISCLR.15	XP10DIDISCLR.14	XP10DIDISCLR.13	XP10DIDISCLR.12	XP10DIDISCLR.11	XP10DIDISCLR.10	XP10DIDISCLR.9	XP10DIDISCLR.8	XP10DIDISCLR.7	XP10DIDISCLR.6	XP10DIDISCLR.5	XP10DIDISCLR.4	XP10DIDISCLR.3	XP10DIDISCLR.2	XP10DIDISCLR.1	XP10DIDISCLR.0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 115. XP10DIDISCLR register description

Bit	Bit name	Function
15-0	XP10DIDISCLR.y	Writing a 1 clears the corresponding bit of the XP10DIDIS.y register. Writing a 0 has no effect.

14 Analog-to-digital converter (ADC)

A 10-bit ADC with 16+16 multiplexed input channels and a sample and hold circuit is integrated on-chip. An automatic self-calibration adjusts the ADC module to process parameter variations at each reset event. The sample time (for loading the capacitors) and the conversion time is programmable and can be adjusted to the external circuitry.

The ADC input bandwidth is limited by the achievable accuracy as follows: If a maximum error of 0.5 LSB (2 mV) impacts the global TUE (TUE also depends on other causes) in the worst case of temperature and process, the maximum frequency for a sine wave analog signal is around 7.5 kHz. To reduce the effect of the input signal variation on the accuracy to 0.05 LSB, the maximum input frequency of the sine wave should be reduced to 800 Hz.

If static signal is applied during the sampling phase, the series resistance should not be greater than 20 kΩ (takingeventual input leakage into account). It is suggested not to connect any capacitance on analog input pins, to reduce the effect of charge partitioning (and consequent voltage drop error) between the external and the internal capacitance. If an RC filter is necessary the external capacitance must be greater than 10 nF to minimize the accuracy impact.

Overrun error detection/protection is controlled by the ADDAT register. Either, an interrupt request is generated when the result of a previous conversion has not been read from the result register at the time the next conversion is complete, or, the next conversion is suspended until the previous result has been read. For applications which require less than 16 analog input channels, the remaining channel inputs can be used as digital input port pins. The ADC of the ST10F296E 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 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**: The ADWR bit of the ADCON control register must be activated to avoid overwriting the result of a current conversion by the next one, when using continuous modes. This is because 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 between two of the continuous conversions without changing the current operating mode. The 10-bit data of the conversions are stored in the ADRES field of the ADDAT2 register. The current continuous mode remains active after the single conversion is completed.

14.1 Mode selection and operation

The analog input channels AN0 to AN15 are alternate functions of Port 5 which is a 16-bit input-only port (see *Section 13.7.2: Alternate functions of port 5 on page 150*). Port 5 lines may either be used as analog or digital inputs. No special action is required to configure the lines as analog inputs. An additional register P5DIDIS can be used to protect the ADC input analog section from disabling the digital input section.

The analog input channels AN16 to AN31 are alternate functions of XPort 10 which is also a 16-bit input-only port (see *Section 13.12.2: Alternate functions of XPort 10 on page 173*). XPort 10 lines may also be used as either analog or digital inputs and the additional XP10DIDIS register can be used to protect the ADC input analog section from disabling the digital input section.

To configure XPort 10 lines as analog inputs, it is first recommended to set register XP10DIDIS. Next, bit ADCMUX of register XMISC must be set. This ensures that all analog input channels of Por t5 are disabled and that the analog signal to the ADC is provided through the XPort 10 pins.

Note: Both the XMISC and XP10DIDIS registers can be accessed only after the XMISCEN and XPEN bits of the XPERCON and SYSCON registers have been set.

Figure 62. ADC block diagram

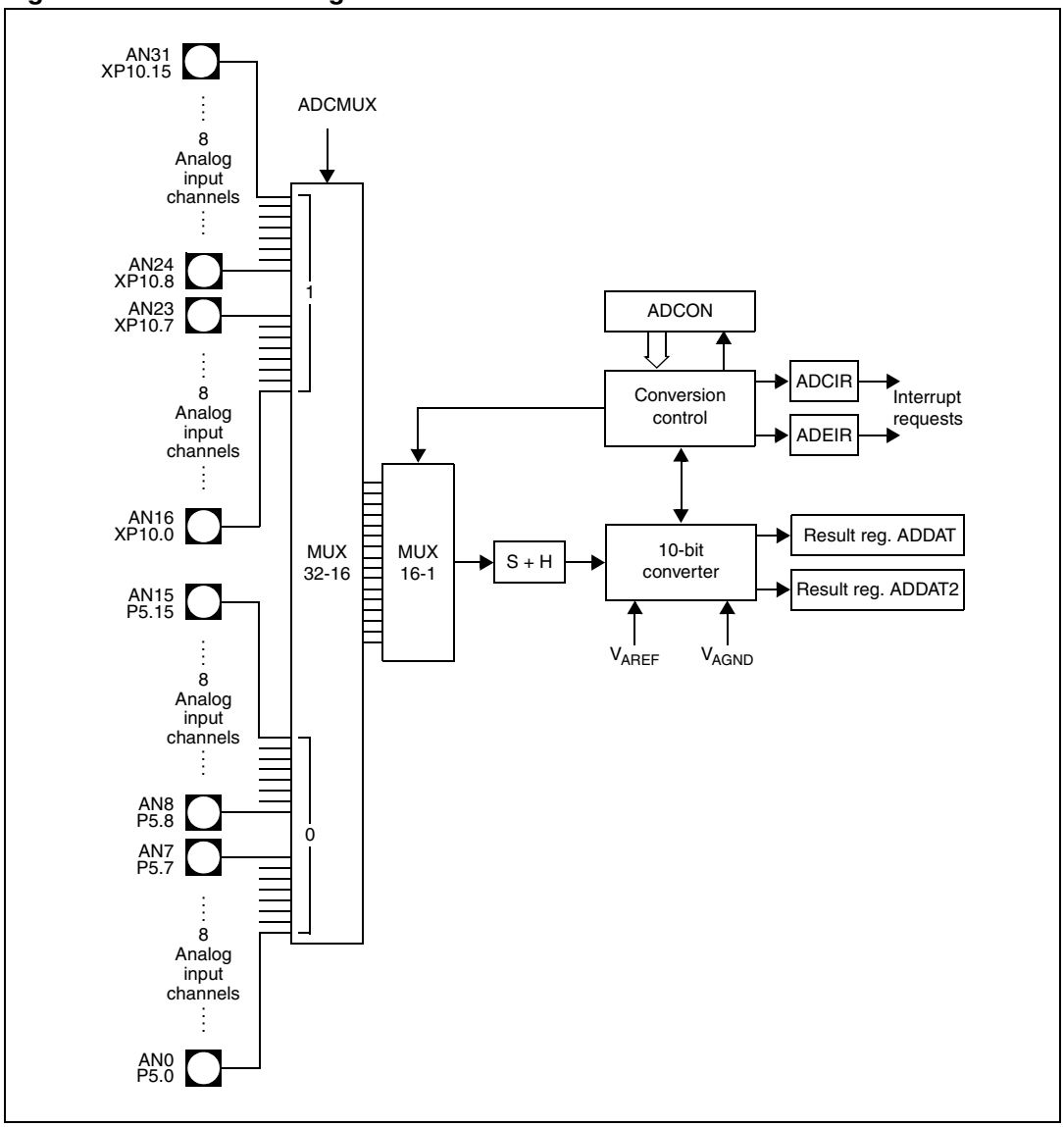


Table 116. ADC programming at f_{CPU} = 64 MHz

ADCTC	ADSTC	Sample	Comparison	Extra	Total convertion
00	00	0.94 µs	1.88 µs	0.22 µs	3.03 µs
00	01	1.09 µs	2.19 µs	0.13 µs	3.41 µs
00	10	1.56 µs	2.19 µs	0.41 µs	4.16 µs
00	11	3.13 µs	2.19 µs	0.34 µs	5.66 µs
11	00	1.88 µs	3.75 µs	0.41 µs	6.03 µs
11	01	2.19 µs	4.38 µs	0.22 µs	6.78 µs
11	10	3.13 µs	4.38 µs	0.78 µs	8.28 µs
11	11	6.25 µs	4.38 µs	0.41 µs	11.28 µs
10	00	3.75 µs	7.50 µs	0.78 µs	12.03 µs
10	01	4.38 µs	8.75 µs	0.41 µs	13.53 µs
10	10	6.25 µs	8.75 µs	1.53 µs	16.53 µs
10	11	12.5 µs	8.75 µs	1.28 µs	22.53 µs

Note: Total conversion time is compatible with the formula valid for the ST10F280, but, the meaning of the bit fields ADCTC and ADSTC is not. The minimum conversion time is 388 TCL, which at 40 MHz CPU frequency corresponds to 4.85 µs (see the ST10F280 datasheet). ST10F296E devices can target a maximum CPU frequency of 64 MHz. This means that the minimum conversion time is around 3 µs.

14.2 Calibration

A full calibration sequence is performed after a reset. It lasts 40.629 ± 1 CPU clock cycles. During this time, the busy flag, ADBSY, is set to indicate the operation. It compensates the capacitance mismatch, so, the calibration procedure does not need to be updated during normal operation.

No conversion can be performed during the calibration time. The ADBSY bit must be polled to verify when calibration is over, and the module can start a convertion.

At the end of the calibration, both the ADCIR and ADEIR flags are set, because the calibration process repeatedly writes spurious conversion results inside the ADDAT register. Consequently, before starting a conversion, the application performs a dummy read of the ADDAT register and clears the two flags in the ADC initialization routine.

Note: If the ADDAT register is not read before starting the first conversion, and if a 'wait for read mode' is entered (by setting the ADWR bit), the ADC is stacked waiting for the register ADDAT read. This is because the result of the current conversion cannot be immediately written inside the ADDAT register which contains the results of the calibration.

14.3 XTimer module

The XTimer module is a 16-bit up/down counter with a 4-bit exponential scaler dedicated to the channel injection mode of the ADC. This mode injects channel into a running sequence without disturbing it. The PEC stores the conversion results in the memory without entering and exiting interrupt routines for each data transfer.

A channel injection can be triggered by an event on the capture/compare CC31 (Port P7.7) of the CAPCOM2 unit by externally connecting the dedicated output XADCINJ of the XTimer to the input P7.7/CC31. The ADC exclusively converts Port 5 or XPort 10 inputs. If one 'y' channel has to be used continuously in injection mode, it must be externally connected by hardware to Port5.y and XPort10.y inputs.

The XTimer peripheral is enabled by setting the XPEN bit of the SYSCON register and bit 10 of the XPERCON register.

14.3.1 Main features

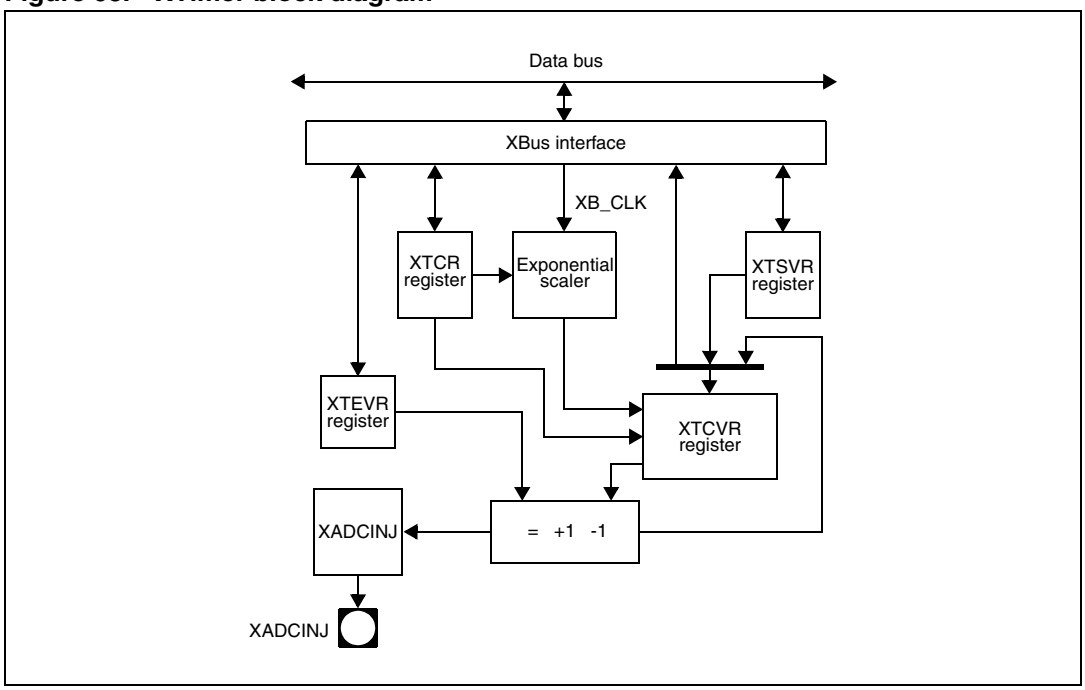
Main features include:

- 16-bit linear timer with 4-bit exponential prescaler
- Counting between 16-bit 'start value' and 16-bit 'end value'
- Counting period between four cycles and 2^{33} cycles (62.5 ns and 134 s using 64 MHz CPU clock)
- 1 trigger output (XADCINJ)

Programmable functions include:

- Up/down counting
- Reload enable
- Continue/stop modes

Figure 63. XTimer block diagram



Clock

The XTCVR register clock is the prescaler output. The prescaler allows the basic register frequency to be divided, therebyoffering a wide range of counting periods, from 2^2 to 2^{33} cycles.

Registers

The XTCVR register input is linked to several sources:

- XTSVR register (start value) for reload when the period is finished, or for load when the timer is starting.
- Incrementer output when the ‘up’ mode is selected
- Decrementer output when the ’down’ mode is selected
- Selection between sources is made through the XTCR control register.

By setting the TEN bit of the XTCR register to 1 when starting the timer, XTCVR is loaded with the XTSVR value on the first rising edge of the counting clock (XB_CLK in *Figure 63*).

The XTCVR register output is continuously compared to the XTEVR register to detect the end of the counting period. When the registers are equal, several things are done depending on the XTCR control register content:

- The output XADCINJ trigger event is generated conditionally depending on the TOE control bit.
- XTCVR is loaded with XTSVR, or it stops, or it continues to count (see *Table 117: Different counting modes on page 181*).

XTEVR, XTSVR and all the XTCR bits except TEN must not be modified while the timer is counting (while XTCR.TEN = 1). The timer can be configured only when it is stopped (TEN = 0). If this rule is not respected, timer behavior is not guaranteed. When programming the timer, the XTEVR, XTSVR and XTCR bits (except TEN) can be modified, with TEN = 0. The timer is started by modifying the TEN bit. To stop the timer, the TEN bit is modified from 1 to 0. To avoid any problems, it is recommended to modify the XTCR register in two steps: First, by writing the new value without setting the TEN bit and second, by re-writing the new value with the the TEN bit set.

Table 117. Different counting modes

TLE	TCS	TCVR(n) = TEVR	TUD	TEN	TCVR (n+1)	comments
x	x	x	x	0	TCVR (n)	Timer disable
x	0	1	x	1		Stop
x	x	0	0		TCVR (n)-1	Decrement
0	1	1				Decrement (continue)
x	x	0	1		TCVR (n)+1	Increment
0	1	1				Increment (continue)
1	1	1	x		TSVR	Load

Note: Setting the TEN bit to 1 loads the XTCVR register with the TSVR value. If the ‘down’ counter mode is selected and XTSVR is less then XTEVR, the XTCVR is loaded with the XTSVR value, but, the timer does not start to count (the current value is hold). The same behavior occurs in up counter mode (TUD = 1) if TSVR > TEVR.

Timer output

The trigger output, XADCINJ, is generated when the current value of the timer (XTCVR) matches the end value stored in the XTEVR register and when the output enable bit is set (XTCR.TOE = 1). If the output enable bit is reset, no event is generated regardless of the timer status (the XADINJ pin is kept at high impedance state).

The XADCINJ output trigger event is a positive pulse of 12 CPU clock cycles width (187 ns @64 MHz). To generate an ADC channel injection it has to be externally connected to the input P7.7/CC31 (CAPCOM2 capture/compare).

The ADC exclusively converts Port 5 or XPort 10 inputs. If one ‘y’ channel has to be used continuously in injection mode, it must be externally connected by hardware to Port5.y and XPort10.y inputs.

15 Serial channels

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

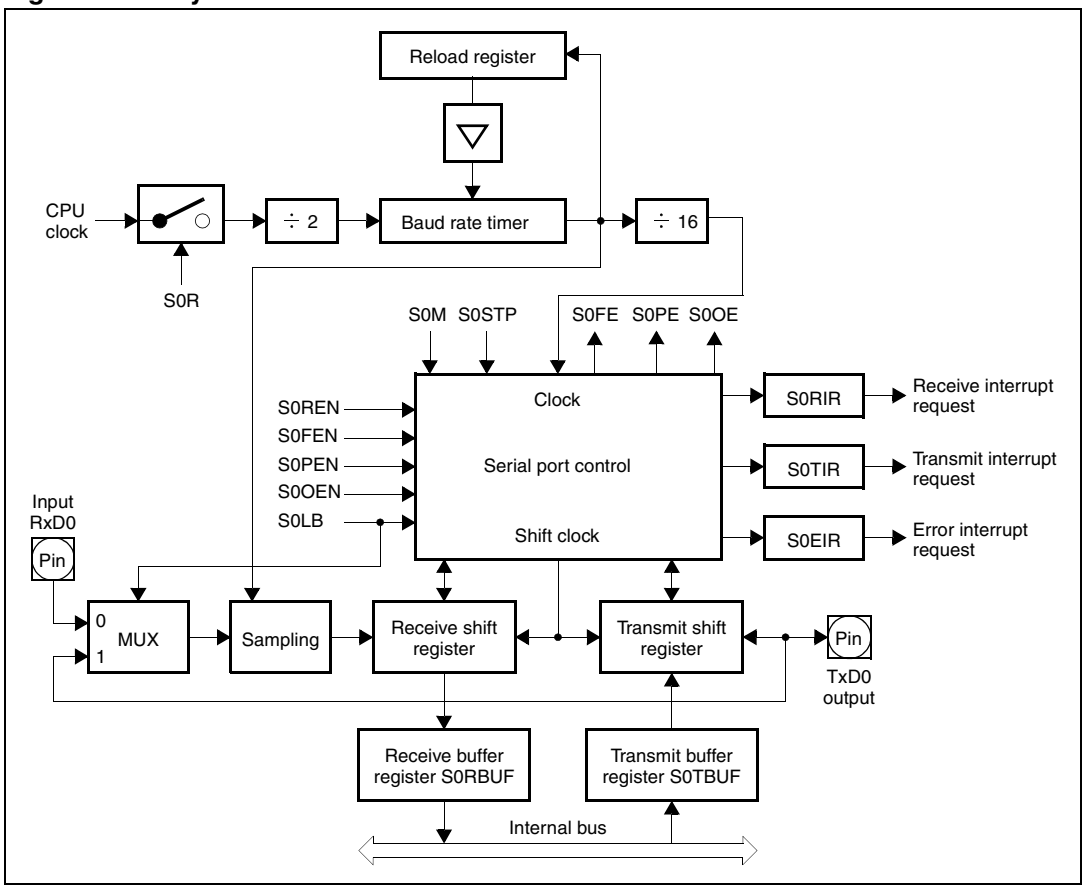
15.1 Asynchronous/synchronous serial interface (ASC0)

The asynchronous/synchronous serial interfaces (ASC0) provides serial communication between the ST10F296E and other microcontrollers, microprocessors or external peripherals.

15.1.1 ASC0 in asynchronous mode

In asynchronous mode, 8- or 9-bit data transfer, parity generation and the number of stop bits can be selected. Parity framing and overrun error detection is provided to increase the reliability of data transfers. Transmission and reception of data is double-buffered. Full-duplex communication up to 2 Mbauds (at 64 MHz of f_{CPU}) is supported in this mode. For reference, see *Figure 64*.

Figure 64. Asynchronous mode of serial channel ASC0



15.1.2 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 operations of serial channel ASC0 and the required reload value for a given baud rate can be determined by the following formulae:

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

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

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

Using the above equations, the maximum baud rate can be calculated for any given clock speed. Baud rate versus the reload register value (for both S0BRS = 0 and S0BRS = 1) is described in *Table 118* and *Table 119* for a CPU clock frequency equal to 40 MHz and 64 MHz respectively.

Table 118. Commonly used baud rates by reload value and deviation error (f_{CPU} = 40 MHz)

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

1. The deviation errors given above are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0 sampling frequency).

Table 119. Commonly used baud rates by reload value and deviation error (f_{CPU} = 64 MHz)

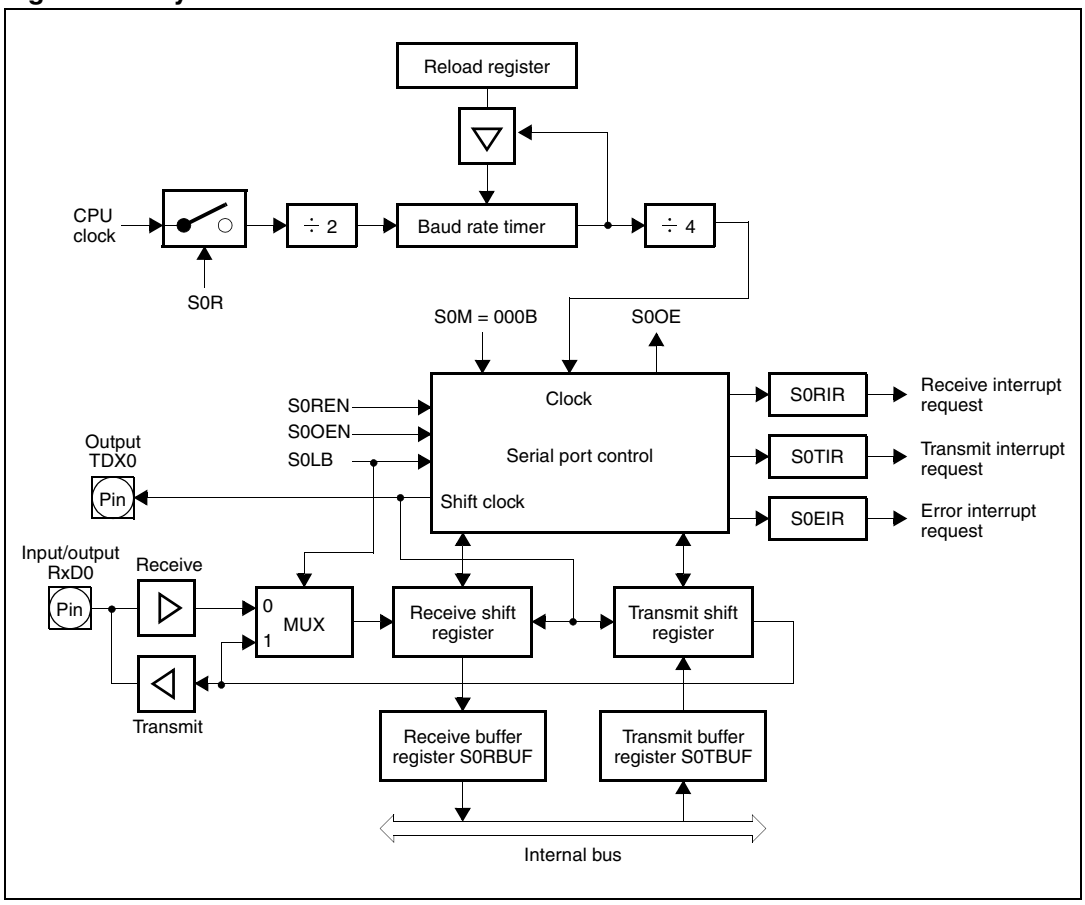
S0BRS = 0, f_{CPU} = 64 MHz			S0BRS = 1, f_{CPU} = 64 MHz		
Baud rate (baud)	Deviation error(1) (%)	Reload value (hex)	Baud rate (baud)	Deviation error(1) (%)	Reload value (hex)
2 000 000	0.0/0.0	0000/0000	1 333 333	0.0/0.0	0000/0000
112 000	1.5/-7.0	0010/0011	112 000	6.3/-7.0	000A/000B
56 000	1.5/-3.0	0022/0023	56 000	6.3/-0.8	0016/0017
38 400	1.7/-1.4	0033/0034	38 400	3.3/-1.4	0021/0022
19 200	0.2/-1.4	0067/0068	19 200	0.9/-1.4	0044/0045
9 600	0.2/-0.6	00CF/00D0	9 600	0.9/-0.2	0089/008A
4 800	0.2/-0.2	019F/01A0	4 800	0.4/-0.2	0114/0115
2 400	0.2/0.0	0340/0341	2 400	0.1/-0.2	022A/015B
1 200	0.1/0.0	0681/0682	1 200	0.1/-0.1	0456/0457
600	0.0/0.0	0D04/0D05	600	0.1/0.0	08AD/08AE
300	0.0/0.0	1A09/1A0A	300	0.0/0.0	115B/115C
245	0.0/0.0	1FE2/1FE3	163	0.0/0.0	1FF2/1FF3

1. The deviation errors given above are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0 sampling frequency).

15.1.3 ASC0 in synchronous mode

In synchronous mode, data are transmitted or received synchronously to a shift clock which is generated by the ST10F296E. Half-duplex communication up to 8 Mbaud (at 40 MHz of f_{CPU}) is possible in this mode. See *Figure 65*.

Figure 65. Synchronous mode of serial channel ASC0



15.1.4 Synchronous mode baud rates

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

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

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

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

Using the above equations, the maximum baud rate can be calculated for any clock speed as given in *Table 121* and *Table 120* for a CPU clock frequency equal to 40 MHz and 64 MHz respectively.

Table 120. Commonly used baud rates by reload value and deviation error (f_{CPU} = 40 MHz)

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

1. The deviation errors given above are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0 sampling frequency).

Table 121. Commonly used baud rates by reload value and deviation errors (f_{CPU} = 64 MHz)

S0BRS = 0, f_{CPU} = 64 MHz			S0BRS = 1, f_{CPU} = 64 MHz		
Baud rate (baud)	Deviation error(1) (%)	Reload value (hex)	Baud rate (baud)	Deviation error(1) (%)	Reload value (hex)
8 000 000	0.0/0.0	0000/0000	5 333 333	0.0/0.0	0000/0000
112 000	0.6/-0.8	0046/0047	112 000	1.3/-0.8	002E/002F
56 000	0.6/-0.1	008D/008E	56 000	0.3/-0.8	005E/005F
38 400	0.2/-0.3	00CF/00D0	38 400	0.6/0.1	0089/008A
19 200	0.2/-0.1	019F/01A0	19 200	0.3/-0.1	0114/0115
9 600	0.0/-0.1	0340/0341	9 600	0.1/-0.1	022A/022B
4 800	0.0/0.0	0681/0682	4 800	0.0/-0.1	0456/0457
2 400	0.0/0.0	0D04/0D05	2 400	0.0/0.0	08AD/08AE
1 200	0.0/0.0	1A09/1A0A	1 200	0.0/0.0	115B/115C
977	0.0/0.0	1FFB/1FFC	900	0.0/0.0	1724/1725
			652	0.0/0.0	1FF2/1FF3

1. The deviation errors given above are rounded. To avoid deviation errors use a baud rate crystal (providing a multiple of the ASC0 sampling frequency).

15.2 Asynchronous/synchronous serial interface (ASC1)

The XBus asynchronous/synchronous serial interfaces (ASC1) provides the same features as ASC0. Baud rate formulae are the same. The main differences are the register interface and interrupt management.

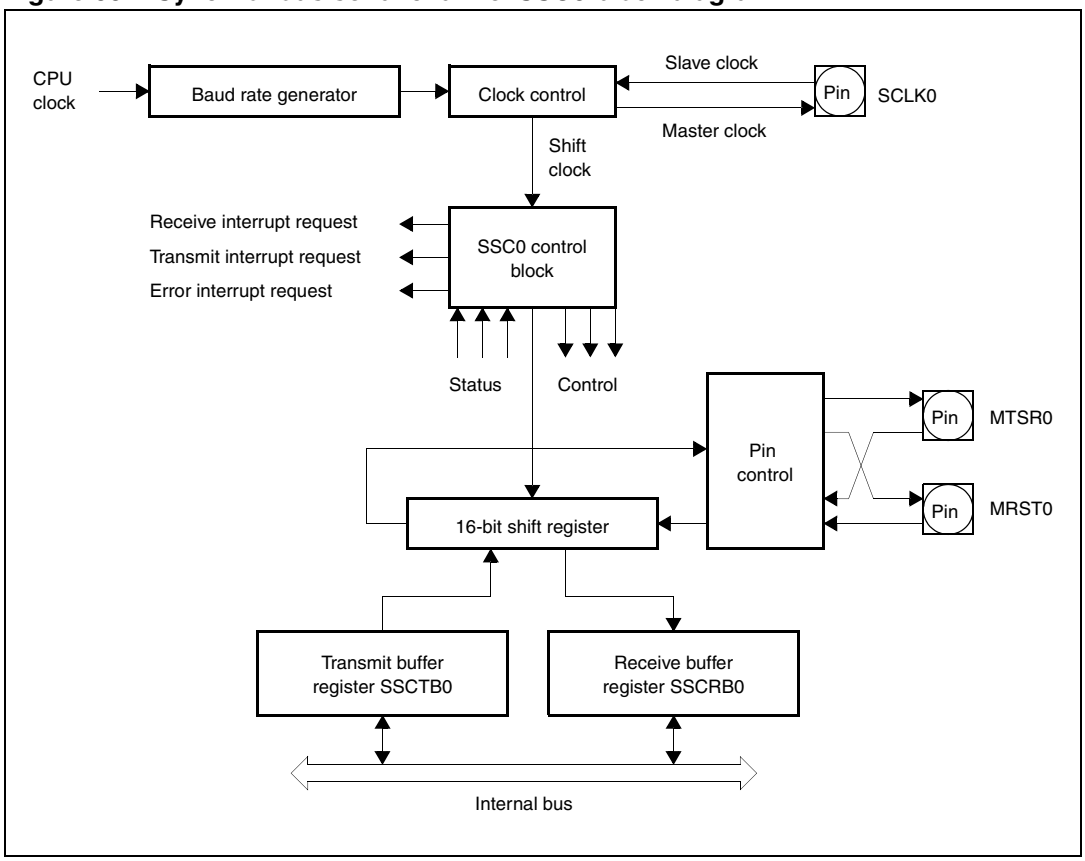
15.3 High speed synchronous serial interface (SSC0)

The high-speed synchronous serial interface, SSC0, provides flexible high-speed serial communication between the ST10F296E and other microcontrollers, microprocessors or external peripherals.

The SSC0 supports full-duplex and half-duplex synchronous communication. The serial clock signal can be generated by the SSC0 itself (master mode) or be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable.

The SSC0 allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A 16-bit baud rate generator provides the SSC0 with a separate serial clock signal. The SSC0 serial channel has its own dedicated 16-bit baud rate generator with 16-bit reload capability, allowing the baud rate to be generated independently from the timers.

Figure 66. Synchronous serial channel SSC0 block diagram



15.3.1 Baud rate generation

The baud rate generator is clocked by $f_{CPU}/2$. The timer counts downwards and can be started or stopped through the global enable bit SSCEN in the SSCCON0 register. The SSCBR0 is a dual-function register for baud rate generation and reloading. Reading SSCBR0, while the SSC0 is enabled, returns the content of the timer. Reading SSCBR0, while the SSC0 is disabled, returns the programmed reload value. In this mode the desired reload value can be written to SSCBR0.

Note: Never write to SSCBR0 while the SSC0 is enabled

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

$$Baudrate_{SSC} = f_{CPU} / 2 \times [(SSCBR) + 1]$$

$$SSCBR = (f_{CPU} / 2 \times Baudrate_{SSC}) - 1$$

Where (SSCBR) represents the content of the reload register, taken as an unsigned 16-bit integer. *Table 122* and *Table 123* list some possible baud rates against the required reload values and the resulting bit times for 40 MHz and 64 MHz CPU clock respectively. Maximum baud rate is limited to 8 Mbaud.

Table 122. Synchronous baud rate and reload values (f_{CPU} = 40 MHz)

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

Table 123. Synchronous baud rate and reload values (f_{CPU} = 64 MHz)

Baud rate	Bit time	Reload value
Reserved	-	0000h
Can be used only with f_{CPU} = 32 MHz (or lower)	-	0001h
Can be used only with f_{CPU} = 48 MHz (or lower)	-	0002h
8 Mbaud	125 ns	0003h
4 Mbaud	250 ns	0007h
1 Mbaud	1 µs	001Fh
100 Kbaud	10 µs	013Fh
10 Kbaud	100 µs	0C7Fh
1 Kbaud	1 ms	7CFFh
489 baud	2.04 ms	FF9Eh

15.4 High speed synchronous serial interface (SSC1)

The XBus high-speed synchronous serial interface, SSC1, provides the same features as the SSC0. Baud rate formulae are the same. The main differences are the register interface and interrupt management.

16 I²C interface

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

The module can generate three different types of interrupt:

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

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

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

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

16.1 I²C bus speed selection

The speed of the I²C interface may be selected between standard mode (0-100 kHz) and fast I²C mode (100-400 kHz). The selection is provided through the FM/SM bit in the clock control register 1 (CCR1).

Once bus mode is selected, the frequency of the serial clock line can be defined by setting prescaler bits CC0 to CC11 (CCR1 and CCR2). Different formulae are used according to the mode selected:

- Standard mode (FM/SM = 0): $F_{SCL} \leq 100$ kHz

  $F_{SCL} = F_{CPU} / (2 \times [CC11..CC0] + 7)$
- Fast mode (FM/SM = 1): $F_{SCL} > 100$ kHz

  $F_{SCL} = F_{CPU} / (3 \times [CC11..CC0] + 9)$

17 CAN modules

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

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

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

- Use the same internal register addresses for both CAN controllers, but, with base addresses differing in address bit A8. Also, use a separate chip select for each CAN module. Refer to *Section 4: Memory organization on page 33*.
- 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.
- The interrupt request lines of the CAN1 and CAN2 modules are connected to the XBus interrupt lines with other XPeripherals sharing the four vectors.
- The CAN modules must be selected with the CANxEN bit of the XPERCON register before the XPEN bit of the SYSCON register is set.
- The reset default configuration is: CAN1 enabled, CAN2 disabled.

17.1 CAN module memory mapping

17.1.1 CAN1

Address range 00’EF00h - 00’EFFFh is reserved for CAN1 module access. CAN1 is enabled by setting the XPEN bit 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 62.5 ns at 64 MHz CPU clock. No tri-state wait states are used.

17.1.2 CAN2

Address range 00’EE00h - 00’EEFFh is reserved for CAN2 module access. CAN2 is enabled by setting the XPEN bit 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 62.5 ns at 64 MHz CPU clock. No tri-state wait states are used.

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

17.2 Configuration support

Both CAN controllers can work on the same CAN bus and support up to 64 message objects. In this configuration, both receive and transmit signals are linked together when using the same CAN transceiver. This configuration is particularly supported by providing open-drain outputs for the CAN1_Txd and CAN2_TxD signals. The open-drain function is controlled with the ODP4 register for Port P4. In this way it is possible to connect pins P4.4 with P4.5 (receive lines) and pins P4.6 with P4.7 (transmit lines configured to be open-drain).

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

Note: CAN parallel mode is effective only if both CAN1 and CAN2 are enabled by setting bits CAN1EN and CAN2EN in the XPERCON register. If CAN1 is disabled, CAN2 remains on P4.4/P4.7 even if bit CANPAR is set.

17.3 Clock prescaling

The XMISC register also provides a bit (CANCK2) which modifies the clock frequency driving both the CAN modules. Due to architectural limitations of the CAN module, when the CPU frequency is higher than 40 MHz, it is recommended to provide each CAN module with the CPU clock divided by 2. It is sufficient to supply 20 MHz for the CAN module to produce the maximum baud rate defined by the protocol standard. The CPU frequency can be reduced down to 8 MHz. It is still possible to obtain the maximum CAN speed (1Mbaud) by feeding the CAN module directly with the CPU clock disabling the prescaler factor.

After reset, the prescaler is enabled, the CPU clock is divided by two, and provides the CAN modules. According to the system clock frequency, the application can disable the prescaler to obtain the required baud rate.

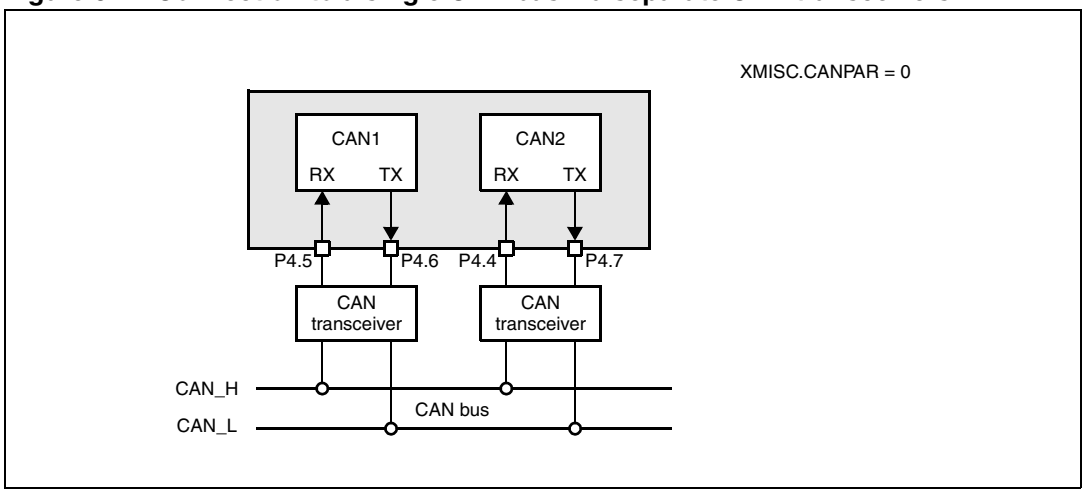
17.4 CAN bus configurations

Depending on the application, CAN bus configuration may be one single bus with single or multiple interfaces or a multiple bus with single or multiple interfaces. The ST10F296E is able to support both situations.

17.4.1 Single CAN bus

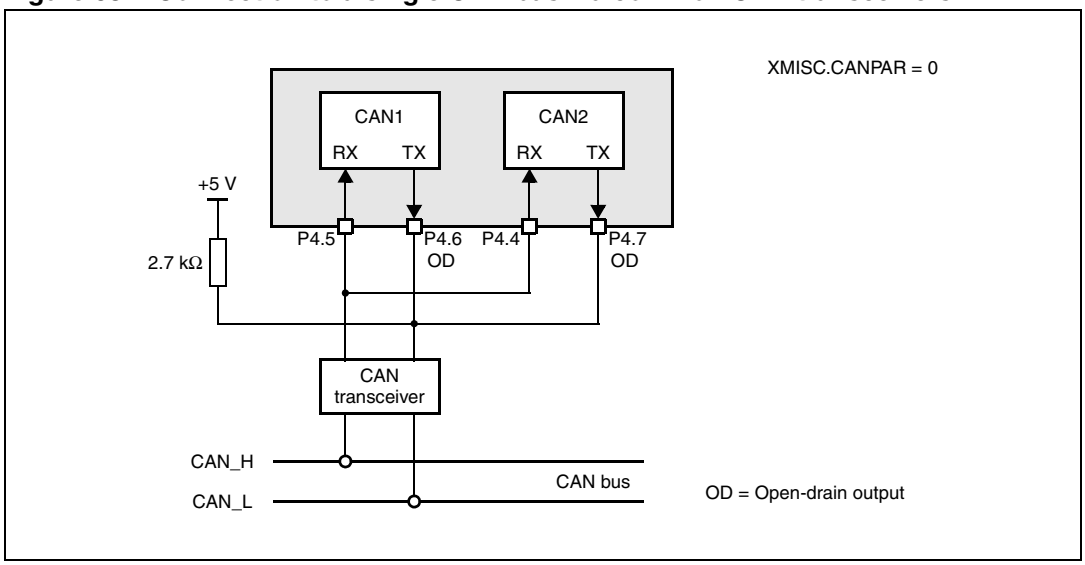
The single CAN bus multiple interface configuration may be implemented using two CAN transceivers as shown in *Figure 67*.

Figure 67. Connection to a single CAN bus via separate CAN transceivers



The ST10F296E also supports single CAN bus multiple (dual) interfaces using the open-drain option of the CANx_TxD output as shown in *Figure 68*. Due to the OR-wired connection, only one transceiver is required. In this case the design of the application must take into account the wire length and the noise environment.

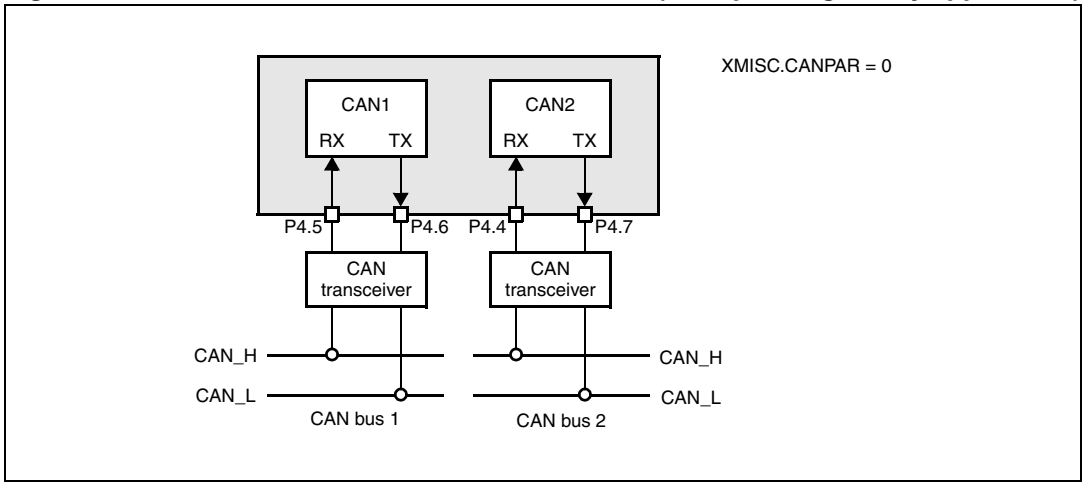
Figure 68. Connection to a single CAN bus via common CAN transceivers



17.4.2 Multiple CAN bus

The ST10F296E provides two CAN interfaces to support the bus configuration shown in *Figure 69*.

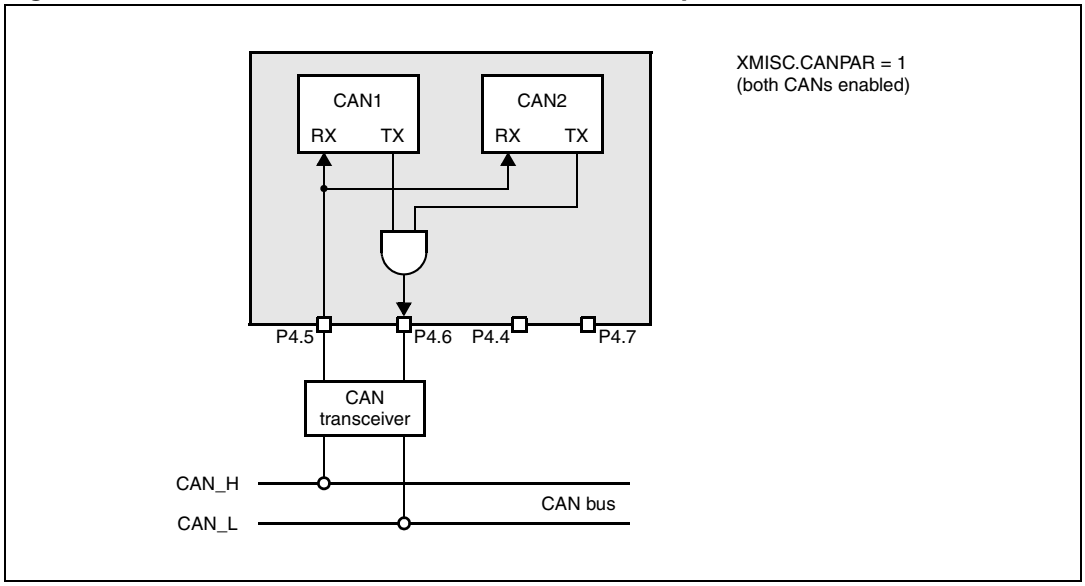
Figure 69. Connection to two different CAN buses (example for gateway application)



17.4.3 Parallel mode

A parallel mode configuration is also supported, as shown in *Figure 70*.

Figure 70. Connection to one CAN bus with internal parallel mode enabled



1. When P4.4 and P4.7 are not used as CAN functions, they can be used as general purpose I/Os, but, they cannot be used as external bus address lines. Refer to *Section 13.6.2: Alternate functions of Port 4 on page 144* for more details.

17.5 System clock tolerance range

The CAN system clock for the different nodes in the network is typically derived from a different clock generator source. The actual CAN system clock frequency for each node (and consequently the actual bit time), is affected by a tolerance. The CAN system clock for the ST10F296E is derived (prescaled) from the CPU clock, typically generated by the on-chip PLL multiplying the frequency of the main oscillator.

For communication to be effective, all CAN nodes in the network should sample the correct value for each transmitted bit. In addition, those nodes with the largest propagation delay (typically at opposite ends of the network), and working with system clocks that are at opposite limits of the frequency tolerance, must be able to correctly receive and decode every message transmitted on the network.

Considering the effect of the system clock discrepancy between two CAN nodes, and assuming no bus errors are detected (for example, due to electrical disturbances), bit stuffing guarantees that, in the worst case condition for the accumulation of phase error (during normal communication), the maximum time between two re-synchronization edges is 10 bit periods (five dominant bits followed by five recessive bits are always followed by a dominant bit).

Calling the CAN bit time, t_{BT}, the maximum time, t_J, between two re-synchronization edges can be expressed as follows:

Equation 8

$$t_J = 10 \times t_{BT}$$

Then, assuming that the two CAN nodes have opposite system clock generator tolerances for their respective system clocks, the accumulated phase error, Δt_J, at the re-synchronization instant becomes:

Equation 9

$$\Delta t_J = (2 \times df) \times 10 \times t_{BT}$$

Where df is the system clock relative tolerance which can be calculated from *Equation 10*:

Equation 10

$$df = |f - f_N| / f_N$$

f = actual frequency and f_N = nominal frequency.

The error in *Equation 10* must be compensated. It must be less than the programmed resynchronization jump width (SJW). Calling t_{SJW} the duration of the resynchronization segment (programmable from 1 to 4 time quanta), *Equation 11* can be written.

Equation 11

$$(2 \times df) \times 10 \times t_{BT} < t_{SJW}$$

Equation 11 can be seen as a condition for the CAN system clock tolerance, df, as shown in *Equation 12*.

Equation 12

$$df < t_{SJW} / 2 \times 10 \times t_{BT}$$

However, considering that real systems typically operate in the presence of electrical disturbances, errors on the CAN bus may occur. If an error is detected, an error flag is transmitted on the bus. If the error is local, only the node which detected it transmits the error flag on the bus; the other nodes receive the error flag and transmit their own error flags as an echo. If the error is global, all nodes detect it within the same bit time and they transmit their own error flags simultaneously. In this way, each node can recognize if the error is local or global simply by detecting whether there is an echo. However, this is possible only if each node can sample the first bit after the error flag has been transmitted.

The error flag from an error active node is composed of six dominant bits. In the worst case situation of a bit stuffing error, an additional six dominant bits could be received before the error flag. This means that the first bit after the error flag is the 13^{th} bit after the last synchronization. This bit, must be correctly sampled.

Calling t_{BT} the CAN bit time, the maximum time, t_S (with correct sampling), between two re-synchronization edges can be expressed as shown in *Equation 13*.

Equation 13

$$t_S = 13 \times t_{BT} - t_{PB2}$$

Where t_{PB2} corresponds to the duration of Phase_Seg2 (PB = phase buffer).

Assuming that the two CAN nodes have opposite system clock generator tolerances for their respective system clocks, *Equation 14* shows the accumulated phase error, Δt_J, at the re-synchronization instant.

Equation 14

$$\Delta t_S = (2 \times df) \times (13 \times t_{BT} - t_{PB2})$$

For correct sampling, the accumulated phase error must not lead the re-synchronization edge outside the interval Phase_seg1 + Phase_Seg2. This condition can be expressed as shown in *Equation 15*.

Equation 15

$$t_{PB1} < (2 \times df) \times (13 \times t_{BT} - t_{Seg2}) < t_{PB2}$$

This expression can be translated to a condition for the CAN system clock tolerance, df, as shown in *Equation 16*.

Equation 16

$$df < \min(t_{PB1}, t_{PB2}) / 2 \times (13 \times t_{BT} - t_{PB2})$$

In conclusion, there are two conditions to be satisfied on the CAN system clock tolerance.

If the CAN node generates its system clock through a PLL, the maximum clock tolerance allowed must also be a function of the PLL jitter. This results in a more severe quality requirement for the oscillator (crystal or resonator).

The phase error introduced by the PLL jitter is a function of the number of clock periods. In particular, the jitter increases with the clock period number up to a maximum saturation value which consists of the long term jitter. Refer to *Section 24.8.7: Phase-locked loop (PLL) on page 315* for more details about the ST10F296E PLL jitter.

Considering the PLL effect, *Equation 17* and *Equation 18* below are modified for the two CAN conditions to give the phase error:

Equation 17

$$\Delta t_J = 2 \times (df \times 10 \times t_{BT} + \delta_{PLL})$$

Equation 18

$$\Delta t_S = 2 \times [df \times (13 \times t_{BT} - t_{PB2}) + \delta_{PLL}]$$

Where δ_{PLL} represents the absolute deviation introduced by the PLL jitter.

In *Equation 17* and *Equation 18* the value of δ_{PLL} must be evaluated for different numbers of clock periods. For the first clock period, the jitter corresponding to 10 bit time periods must be considered, while for the second clock period, the jitter corresponding to 13 bit time periods must be considered. The number of clock periods must be computed taking account of the baud rate prescaler setting. A factor of two, which multiplies the single CAN node phase deviation, is considered to take account of the worst case scenario where two communicating nodes are at the opposite limits of the specified frequency tolerance.

From *Equation 17* and *Equation 18*, the new constraints for the CAN system clock tolerance can be translated into new quality requirements for the oscillator as shown in *Equation 19* and *Equation 20*.

Equation 19

$$df < t_{SJW} - 2 \times \delta_{PLL} / \ 2 \times 10 \times t_{BT}$$

Equation 20

$$df < \min(t_{PB1}, \ t_{PB2}) - 2 \times \delta_{PLL} / \ 2 \times (13 \times t_{BT} - t_{PB2})$$

It is obvious that the PLL jitter imposes more stringent constraints on oscillator tolerance than what can be accepted when no PLL is used. The ST10F296E PLL characteristics are such that the oscillator requirements are acceptably impacted by the jitter for the majority of the worst CAN bus network configurations.

Oscillator tolerance range was increased when the CAN protocol was developed from version 1.1 to version 1.2 (version 1.0 was never implemented in silicon). The option to synchronize on edges from dominant to recessive became obsolete and only edges from recessive to dominant are now considered for synchronization. Protocol update to version 2.0 (A and B) has had no influence on oscillator tolerance.

It must be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits the part of the bit time that may be used for the phase buffer segments.

The combination below allows the largest possible frequency tolerance of 1.58 % (in the absence of PLL jitter):

- Prop_Seg = 1
- Phase_Seg1 = Phase_Seg2 = SJW = 4

This combination with a propagation time segment of only 10 % of the bit time is not suitable for short bit times. It can be used for bit rates of up to 125 Kbit/s (bit time = 8 µs) with a bus length of 40 m.

17.6 Configuration of the CAN controller

In the C-CAN and in most CAN implementations, the bit timing configuration is programmed in two register bytes. The sum of Prop_Seg and Phase_Seg1 (as TSeg1) is combined with Phase_Seg2 (as TSeg2) in one byte, and SJW and BRP are combined in the second byte.

In these bit timing registers (CANxBTR), the four components TSeg1, TSeg2, SJW, and BRP have to be programmed to a numerical value that is one less than its functional value. Therefore, instead of values in the range of [1...n], values are programmed in the range [0...n-1]. Consequently, SJW (functional range of [1...4]) is represented by only two bits.

The length of the bit time is [TSeg1 + TSeg2 + 3] t_q (programmed values) or [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q (functional values).

The data in the bit timing registers are the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRP) defines the length of the time quantum and the basic time unit of the bit time. The bit timing logic (configured by TSeg1, TSeg2, and SJW) defines the number of time quanta in the bit time.

Processing of the bit time, calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing logic (BTL) state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP.

The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the cyclic redundancy check (CRC) code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (for example, data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT).

The IPT is application specific but may not be longer than 2 t_q. The C-CAN's IPT is 0 t_q. Its length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

17.7 Calculation of the bit timing parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1/ bit rate) must be an integer multiple of the system clock period.

The bit time may consist of four to 25 time quanta. The length of the time quantum, t_q, is defined by the baud rate prescaler, with t_q = (baud rate prescaler) / f_{sys}. Several combinations may lead to the desired bit time, allowing iterations of the steps described below.

The first part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two phase buffer segments. If the number of the remaining t_q is even, the phase buffer segments have the same length:

Phase_Seg2 = Phase_Seg1

else:

Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has also to be considered. Phase_Seg2 should not be shorter than the CAN controller's information processing time, which, depending on the actual implementation, is in the range of [0...2] t_q.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four times quanta and the value defined by the Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulae given in *Section 17.5: System clock tolerance range on page 196*.

If more than one configuration is possible, the configuration allowing the highest oscillator or PLL tolerance range should be chosen.

CAN nodes with different system clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is made once for the whole network.

The CAN system's oscillator (or PLL) tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased, or that the oscillator frequency stability has to be increased to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register:

(Phase_Seg2 - 1) &
(Phase_Seg1 + Prop_Seg - 1) &
(SynchronisationJumpWidth - 1) &
(Prescaler - 1)

17.7.1 Example of bit timing at high baud rate

In this example, the CPU frequency (CAN module clock) is 10 MHz, BRP is 0, and the bit rate is 1 Mbit/s.

t_q	100 ns	= t_{CPU}
Delay of bus driver	50 ns	
Delay of receiver circuit	30 ns	
Delay of bus line (40 m)	220 ns	
t_{Prop}	600 ns	= 6 x t_q
t_{SJW}	100 ns	= 1 x t_q
t_{PB1}	100 ns	= 1 x t_q
$t_{Seg1} = t_{Prop} + t_{PB1}$	700 ns	= 7 x t_q
$t_{Seg2} = t_{PB2}$	200 ns	= Information processing time + 1 x t_q = 2 x t_q
$t_{Sync\text{-}Seg}$	100 ns	= 1 x t_q
t_{BT}	1000 ns	= $t_{Sync\text{-}Seg} + t_{Seg1} + t_{Seg2}$ = 10 x t_q
Tolerance for CAN clock	0.39 %	=

$$\min(t_{PB1}, t_{PB2}) \div (13 \times t_{BT} - t_{PB2}) = 0.1(\mu s) / \ 2 \times (13 \times 1(\mu s) - 0.2(\mu s))$$

δ_{PLL} (13 x t_{BT} = 13 x 10 x t_q = 130 t_{CPU} 5 ns = Data from PLL jitter characteristics

Tolerance for oscillator (no PLL effect)0.35% =

$$\min(t_{PB1}, t_{PB2}) - 2 \times \delta_{PLL} / \ 2 \times (13 \times t_{BT} - t_{PB2})$$

In this example, the concatenated bit time parameters are $(2\text{-}1)_3$ & $(7\text{-}1)_4$ & $(1\text{-}1)_2$ & $(1\text{-}1)_6$, the bit timing register CANxBTR is programmed to = 0x1600h.

17.7.2 Example of bit timing at low baud rate

In this example, the frequency of the CAN module clock is 2 MHz, **BRP** is 1, the bit rate is 100 Kbit/s.

t_q	1	µs = 2 x t_{CPU}
Delay of bus driver	200	ns
Delay of receiver circuit	80	ns
Delay of bus line (40m)	220	ns
t_{Prop}	1	µs = 1 x t_q
t_{SJW}	4	µs = 4 x t_q
t_{PB1}	4	µs = 4 x t_q
$t_{Seg1} = t_{Prop} + t_{PB1}$	5	µs = 5 x t_q
$t_{Seg2} = t_{PB2}$	4	µs = Information processing time + 3 x t_q = 4 x t_q
$t_{Sync-Seg}$	1	µs = 1 x t_q
t_{BT}	10	µs = $t_{Sync-Seg} + t_{Seg1} + t_{Seg2}$ = 10 x t_q
Tolerance for CAN clock	1.58	% =

$$\min(t_{PB1},t_{PB2}) \div (13 \times t_{BT} - t_{PB2}) = 4(\mu s) / \ 2 \times (13 \times 10(\mu s) - 4(\mu s))$$

δ_{PLL} (13 x t_{BT} = 13 x 10 x t_q = 260 t_{CPU}) 10 ns = Data from PLL jitter characteristics

Tolerance for oscillator (no PLL effect 1.57 % =

$$\min(t_{PB1},t_{PB2}) - 2 \times \delta_{PLL} / \ 2 \times (13 \times t_{BT} - t_{PB2})$$

In this example, the concatenated bit time parameters are $(4\text{-}1)_3$ & $(5\text{-}1)_4$ & $(4\text{-}1)_2$ & $(2\text{-}1)_6$, the bit timing register CANxBTR is programmed to = 0x34C1h.

18 Real-time clock (RTC)

The RTC is an independent timer. It is directly derived from the clock oscillator on XTAL1 (main oscillator) input, so that it can be kept running even in idle or power-down mode (if it is enabled). Register access is implemented onto the XBus. This module is designed with the following characteristics:

- Generation of the current time and date for the system
- Cyclic time based interrupt on Port 2 external interrupts every 'RTC basic clock tick' and after *n* 'RTC basic clock ticks' if enabled (*n* is programmable).
- 58-bit timer for long-term measurements
- Capability to exit the ST10 chip from power-down mode (if the PWDCFG bit of the SYSCON register is set) after a programmed delay.

The RTC is based on two main blocks of counters. The first block is a prescaler which generates a basic reference clock (for example a one-second period). This basic reference clock comes out of a 20-bit divider (4-bit MSB RTCDH counter and 16-bit LSB RTCDL counter). The 20-bit divider is driven by an input clock which is derived from the on-chip high frequency CPU clock and pre-divided by a 1/64 fixed counter (see *Figure 72*). The divider 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).

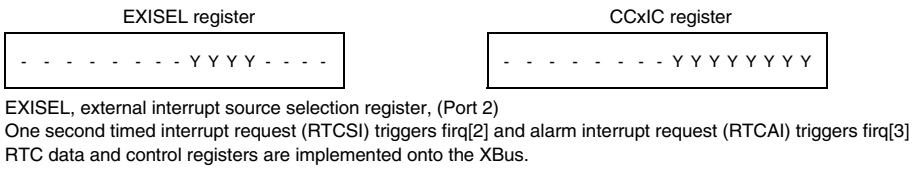
The value of the 20-bit RTCP register determines the period of the basic reference clock. A timed interrupt request (RTCSI) may be sent on each basic reference clock period. The second block of the RTC is a 32-bit counter (16-bit RTCH and 16-bit RTCL). This counter may be initialized with the current system time. The RTCH/RTCL counter is driven with the basic reference clock signal. To provide an alarm function, the contents of the 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 the RTCH/RTCL counter matches the reference date of the RTCAH/RTCAL register.

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

Since the RTC counter is driven by the main oscillator (powered by the main power supply), it cannot be maintained running in stand-by mode. The opposite is true in power-down mode, where the main oscillator can be maintained running to provide the reference to the RTC module (if not disabled).

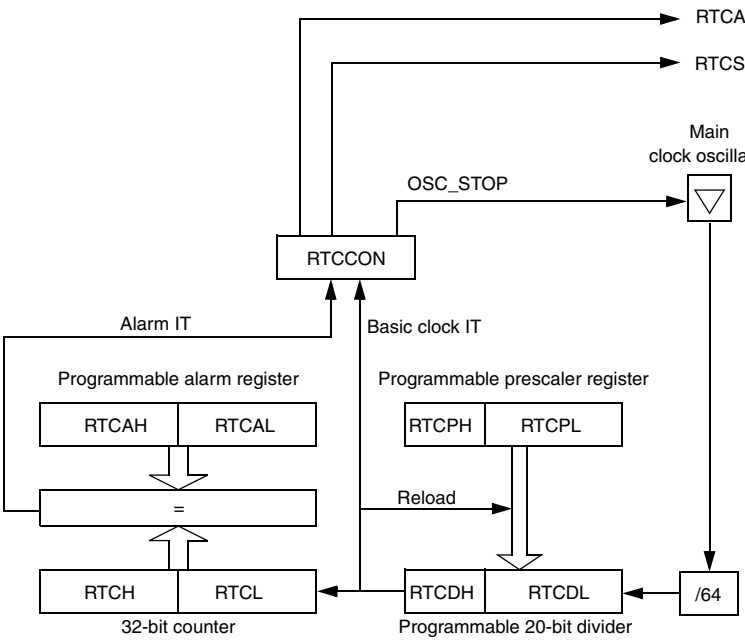
Figure 71 below shows the ESFRs and port pins associated with the RTC.

Figure 71. ESFRs and port pins associated with the RTC



EXISEL, external interrupt source selection register, (Port 2)
One second timed interrupt request (RTCSI) triggers firq[2] and alarm interrupt request (RTCAI) triggers firq[3]
RTC data and control registers are implemented onto the XBus.

Figure 72. RTC block diagram



18.1 RTC registers

RTC control register (RTCCON)

The functions of the RTC are controlled by the RTCCON control register (see register table and description below). If the RTOFF bit is set, the RTC dividers and counter clocks are disabled and the registers can be written. When the ST10 chip enters power-down mode, the clock oscillator is switched off. The RTC has two interrupt sources: One is triggered every basic clock period, the other is the alarm.

Note: The RTC registers are not bit-addressable.

The RTCCON register includes an interrupt request flag and an interrupt enable bit for each interrupt source. This register is read and written via the XBus.

RTCCON (ED00h) XBus Reset value: --00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	RTC OFF	-	-	-	RTC AEN	RTC AIR	RTC SEN	RTC SIR
-	-	-	-	-	-	-	-	RW	-	-	-	RW	RW	RW	RW

Table 124. RTCCON register description

Bit	Bit name	Function
7	RTCOFF(1)	RTC switch off bit 0: Clock oscillator and RTC keep running even if ST10 is in power-down mode. 1: If ST10 enters power-down mode, clock oscillator is switched off, RTC dividers and counters are stopped, and registers can be written.
3	RTCAEN	RTC alarm interrupt enable 0: RTCAI is disabled 1: RTCAI is enabled; it is generated when the counters reach the alarm value.
2	RTCAIR(2)(3)	RTC alarm interrupt request flag (when the alarm is triggered) 0: RTCAIR bit is reset in less than an *n* basic clock tick. 1: An interrupt is triggered
1	RTCSEN	RTC second interrupt enable 0: RTCSI is disabled 1: RTCSI is enabled; it is generated every basic clock tick
0	RTCSIR(2)(3)	RTC second interrupt request flag (every second) 0: RTCSIR bit is reset in less than an a basic clock tick. 1: An interrupt is triggered

1. The two RTC interrupt signals are connected to Port 2 to trigger an external interrupt that can wake up the chip when in power-down mode.
2. *To clear the RTC interrupt request flags (bit 0 and bit 2 of the RTCCON register) it is necessary to write a 1 to the corresponding bit of the RTCCON register.*
3. As the RTCCON register is not bit-addressable, the value of its bits must be read by checking their associated CCxIC register.

RTC prescaler registers (RTCPH and RTCPL)

The 20-bit programmable prescaler divider is loaded with two registers: The RTC prescaler high (RTCPH) and RTC prescaler low (RTCPL).

The four most significant bits are stored in the RTCPH and the 16 least significant bits are stored in the RTCPL.

To maintain the system clock, these registers are not reset. They are write protected by the RTCOFF bit of the RTCCON register. Write operation is allowed when RTCOFF is set.

RTCPH (ED08h) XBus Reset value: ---Xh

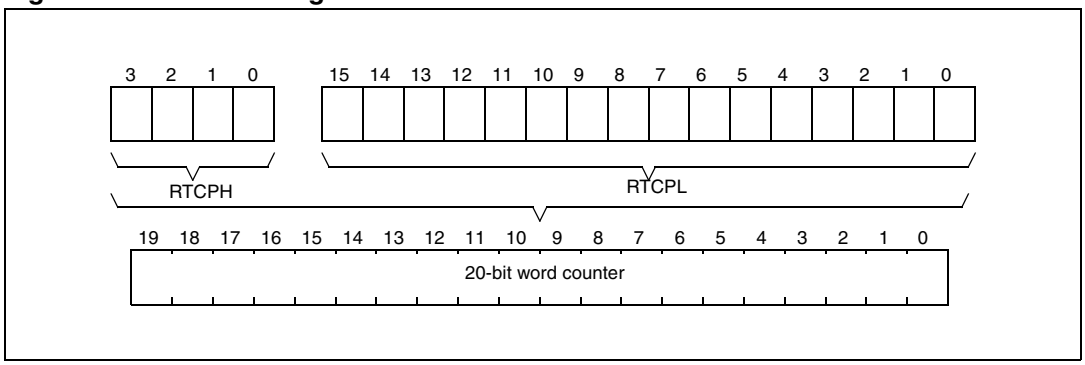
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												RTCPH			
-												RW			

Note: Bits 15 to 4 of the RTCPH are not used. When reading this register, the return value of these bit is zero.

RTCPL (ED06h) XBus Reset value: XXXXh

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

Figure 73. Prescaler registers



The value stored in RTCPH and RTCPL is called RTCP (coded on 20-bit). The dividing ratio of the prescaler divider is: 64 x (RTCP).

RTC divider counter registers (RTCDH and RTCDL)

The divider counter registers (the basic reference clocks) include the RTC divider high (RTCDH) and RTC divider low (RTCDL). These registers are read-only. They are reloaded with the value stored in the prescaler registers, RTCPH and RTCPL. To get accurate time measurements, the value of the divider can be read by reading the RTCDH and RTCDL. When a bit is changed in the prescaler register, the value is loaded into the divider. When the divider increments to reach 00000h, the 20-bit word stored in RTCPH or RTCPL is loaded into it.

RTCDH (ED0Ch) XBus Reset value: ---Xh

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

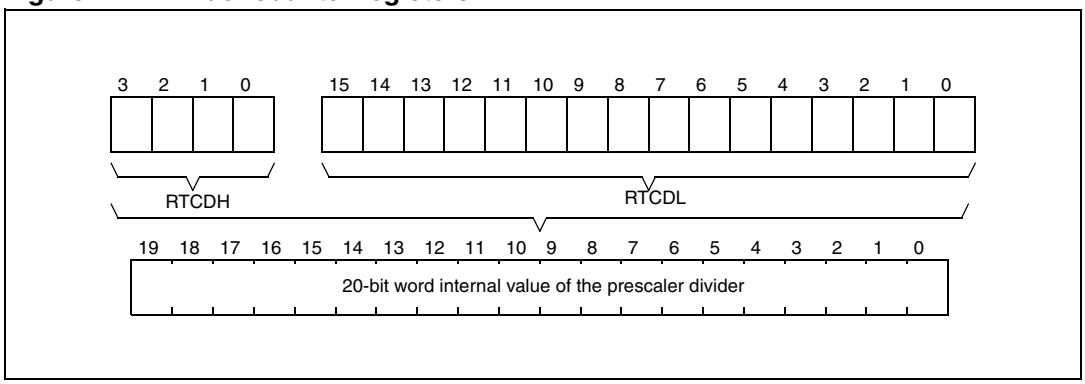
Note: Bits 15 to 4 of the RTCDH are not used. When reading this register, the return value of these bit is zero.

RTCDL (ED0Ah) XBus Reset value: XXXXh

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

Note: Neither the RTCDH nor the RTCDL counter registers can be reset.

Figure 74. Divider counter registers



RTC programmable counter registers (RTCH and RTCL)

The RTC has two x 16-bit programmable counters which are controlled by two counter registers: The RTC counter high register (RTCH) and the RTC counter low register (RTCL).

The count rate of the counters is based on a basic time reference (for example, 1 s). 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 a system reset. The only way to force their value is to write them via the XBus.

The RTC counter registers are write protected. The RTCOFF bit of the RTCCON register (see *Table 124*) 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 the corresponding counter directly. When reading, the current value in the counter (system date) is returned.

The counters keep running while the clock oscillator is working.

RTCH (ED10h) XBus Reset value: XXXXh

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

RTCL (ED0Eh) XBus Reset value: XXXXh

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

Note: Neither the RTCL nor the RTCH registers can be reset.

RTC alarm registers (RTCAH and RTCAL)

The RTC alarm registers include the RTC alarm high (RTCAH) and RTC alarm low (RTCAL). When the counters reach the 32-bit value stored in the RTCAH and RTCAL registers, an alarm is triggered and the interrupt request, RTAIR, is generated. These registers are not protected.

RTCAH (ED14h) XBus Reset value: XXXXh

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

RTCAL (ED12h) XBus Reset value: XXXXh

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

Note: Neither the RTCAL nor the RTCAH registers can be reset.

18.2 Programming the RTC

RTC interrupt request signals are connected to Port 2, pad 10 (RTCSI) and pad 11 (RTCAI). An alternate function of Port 2 is to generate fast interrupts, firq[7:0]. To trigger firq[2] and firq[3] the EXICON register must be used. RTC interrupt requests are rising edge active and the EXICON register controls the external interrupt edge selection.

The EXISEL register enables Port 2 alternate sources. RTC interrupts are alternate sources 2 and 3.

The following Interrupt control registers are common with the CAPCOM1 unit: CC10IC (RTCSI) and CC11IC (RTCAI).

EXICON 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(1)(2)		EXI2ES(1)(3)		EXI1ES		EXI0ES	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

1. EXI2ES and EXI3ES must be configured as 01b because RTC 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 register

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(1)		EXI2SS(2)		EXI1SS		EXI0SS	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

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

Table 125. EXISEL register description

Bit	Bit name	Function
15-0	EXIxSS	External interrupt x source selection (x = 7 to 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

CCxIC registers

CC10IC: FF8Ch/C6h

CC11IC: FF8Eh/C7h

CCxIC SFR Reset value: --00h

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

Table 126. Interrupt sources associated with the RTC

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

19 Watchdog timer

The watchdog timer is a fail-safe mechanism which prevents the microcontroller from malfunctioning over 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. 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 to allow external hardware components to be reset.

Each of the different reset sources is indicated in the watchdog control register (WDTCON).

The bits indicated in *Table 127* are cleared with the EINIT instruction. The source of the reset can be identified during the initialization phase.

Watchdog control register (WDTCON)

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

Table 127. WDTCON register description

Bit	Bit name	Function
15-8	WDTREL	Watchdog reload value
5	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.
4	LHWR(1)(2)(3)	Long hardware reset indication flag Set by the input $\overline{\text{RSTIN}}$. Cleared by the EINIT instruction.
3	SHWR(1)(2)(3)	Short hardware reset indication flag Set by the input $\overline{\text{RSTIN}}$. Cleared by the EINIT instruction.
2	SWR(1)(2)(3)	Software reset indication flag Set by the SRST execution. Cleared by the EINIT instruction.
1	WDTR(1)(2)(3)	Watchdog timer reset indication flag Set by the watchdog timer on an overflow. Cleared by a hardware reset or by the SRVWDT instruction.
0	WDTIN	Watchdog timer input frequency selection 0: Input frequency is $f_{CPU}/2$. 1: Input frequency is $f_{CPU}/128$.

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_{18} = 0 V to V_{18} > 1.5 V is recognized on the internal 1.8 V supply.
3. Bit cannot be modified directly by software.

The PONR flag of the WDTCON register is set if the output voltage of the internal 1.8 V supply falls below the threshold of the power-on detection circuit (typically 1.5 V). This circuit can detect major failures of the external 5 V supply, but, if the internal 1.8 V supply does not drop below 1.5 V, the PONR flag is not set.

This could occur with a 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 128. WDTCON bit values 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 cannot be set because of a short supply failure.
2. For power-on reset and resets after supply partial failure, asynchronous resets must be used.

If a bidirectional reset is enabled, and if the $\overline{\text{RSTIN}}$ pin is latched low at the end of an internal reset sequence, a short hardware reset, a software reset or a watchdog reset triggers a long hardware reset. Thus, reset indications flags are set to indicate a long hardware reset.

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

Each time the high byte of the watchdog timer is serviced by the application software, it is reloaded. For security reasons, the WDTCON register should be rewritten each time before the watchdog timer is serviced

Table 129 and *Table 130* show the watchdog time range for 40 MHz and 64 MHz CPU clock respectively.

Table 129. WDTREL reload value (f_{CPU} = 40 MHz)

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

Table 130. WDTREL reload value (f_{CPU} = 64 MHz)

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

The watchdog timer period is calculated using the following formula:

Equation 21

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

20 System reset

System reset initializes the MCU in a predefined state. There are six ways to activate a reset state. The system start-up configuration is different for each case as shown in *Table 131*

Table 131. Reset event definition

Reset source	Flag	RPD status	Conditions
Power-on reset	PONR	Low	Power-on
Asynchronous hardware reset	**LHWR**	Low	$t_{\overline{RSTIN}}$ > 500 ns and > Port 0 set-up time(1)
Synchronous long hardware reset		High	$t_{\overline{RSTIN}}$ > (1032 + 12) TCL + max (4 TCL, 500 ns)
Synchronous short hardware reset	SHWR	High	$t_{\overline{RSTIN}}$ > max(4 TCL, 500 ns) $t_{\overline{RSTIN}}$ ≤(1032 + 12) TCL + max (4 TCL, 500ns)
Watchdog timer reset	WDTR	(2)	WDT overflow
Software reset	SWR	(2)	SRST instruction execution

1. The $\overline{RSTIN}$ pulse should be > 500 ns (filter) and > Port 0 set-up time. If Port 0 set-up time is below 500 ns, there is no additional settling time. See *Section 20.1* for more details on minimum and reset pulse duration.
2. The RPD pin status has no influence unless a bidirectional reset is activated (BDRSTEN bit in the SYSCON register). When RPD is low, bidirectional resets on software and watchdog timer reset events are inhibited (that is, $\overline{RSTIN}$ is not activated) Refer to *Section 20.4*, *Section 20.5* and *Section 20.6*).

20.1 Input filter

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

For this reason, all minimum durations for the different types of reset events in this section, should be carefully evaluated taking account of the above requirements.

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

- For a CPU clock of 64 MHz, 4 TCL is 31.25 ns, so it is filtered: In this case, the minimum becomes the value imposed by the filter (500 ns).
- For a CPU clock of 4 MHz, 4 TCL is 500 ns: In this case, the minimum value from the formula (see conditions column in *Table 131*) is coherent with the limit imposed by the filter.

20.2 Asynchronous reset

An asynchronous reset is triggered when the $\overline{\text{RSTIN}}$ pin is pulled low while the RPD pin is at low level. The ST10F296E device is immediately (after the input filter delay) forced into a reset default state. It pulls the $\overline{\text{RSTOUT}}$ pin low, it cancels pending internal hold states (if any), it aborts all internal/external bus cycles, it switches buses (data, address and control signals) and I/O pin drivers to high-impedance, and it pulls the Port 0 pins high.

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

20.2.1 Power-on reset

The asynchronous reset must be used during the power-on of the device. Depending on the crystal or resonator frequency, the on-chip oscillator needs about 1 ms to 10 ms to stabilize (refer to *Section 24: Electrical characteristics*), with an already stable V_{DD}. The logic of the ST10F296E does not need a stabilized clock signal to detect an asynchronous reset, so it is suitable for power-on conditions. To ensure a proper reset sequence, the $\overline{\text{RSTIN}}$ pin and the RPD pin must be held low until the device clock signal is stabilized and the system configuration value on Port 0 has settled.

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

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

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

Note: The above is not true if the external memory is used (pin $\overline{EA}$ held low during reset phase). In this case, once the $\overline{RSTIN}$ pin is released, and after a few CPU clock (filter delay plus 3...8 TCL), the internal reset signal RST is released, afterwhich code execution can start immediately. Eventual access to the data in the internal Flash is forbidden before its initialization phase is complete. An eventual access during the starting phase returns FFFFh at the beginning and 009Bh later on (an illegal opcode trap can be generated).

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

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

Figure 75 and *Figure 76* show the asynchronous power-on timing diagrams with boot from internal or external memory respectively. The reset phase extension that is introduced by the embedded Flash module, is highlighted.

Note: Never power the device without keeping the $\overline{RSTIN}$ pin grounded as the device could enter unpredictable states which could permanently damage it.

Figure 75. Asynchronous power-on reset ($\overline{EA}$ = 1)

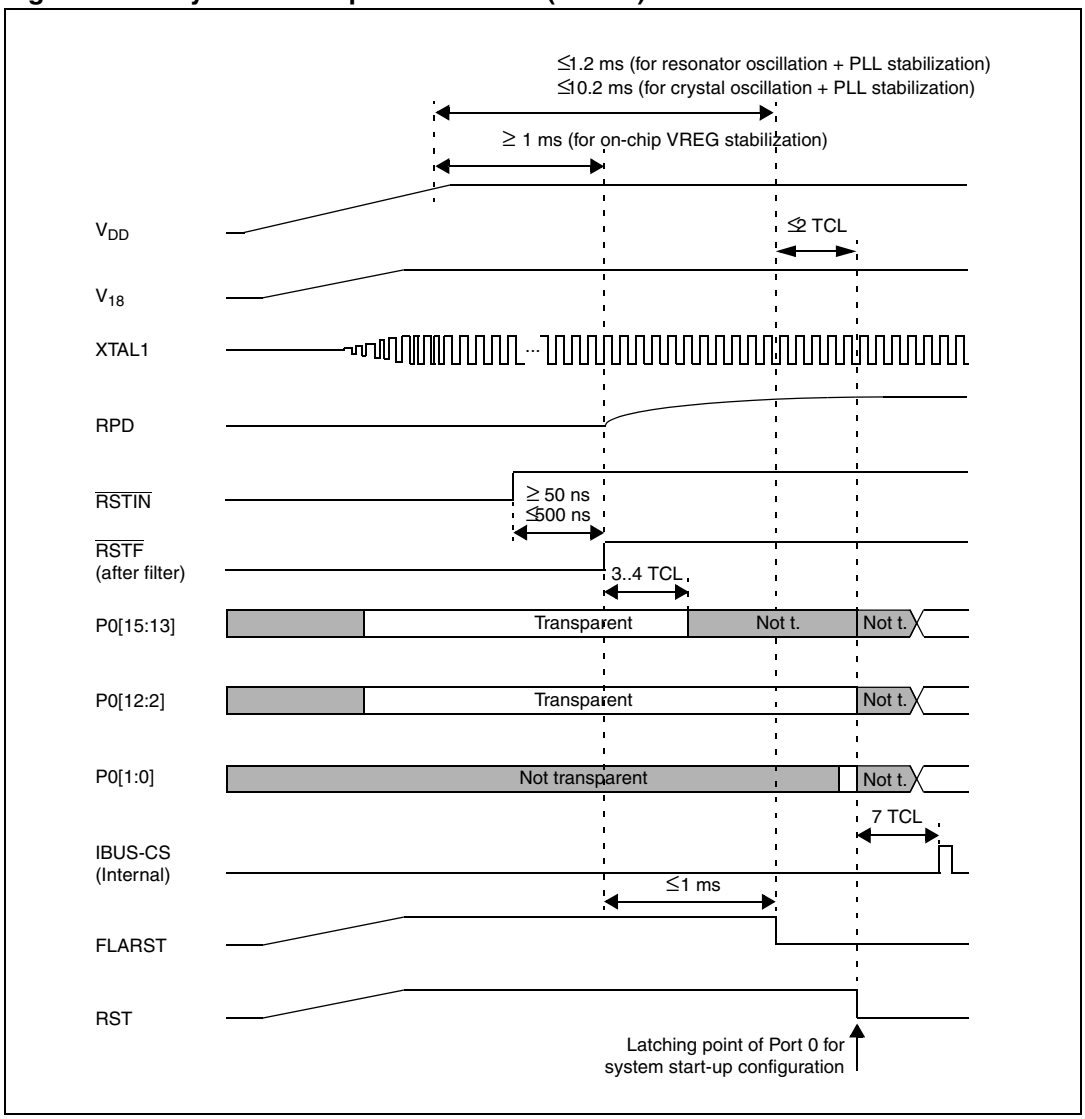
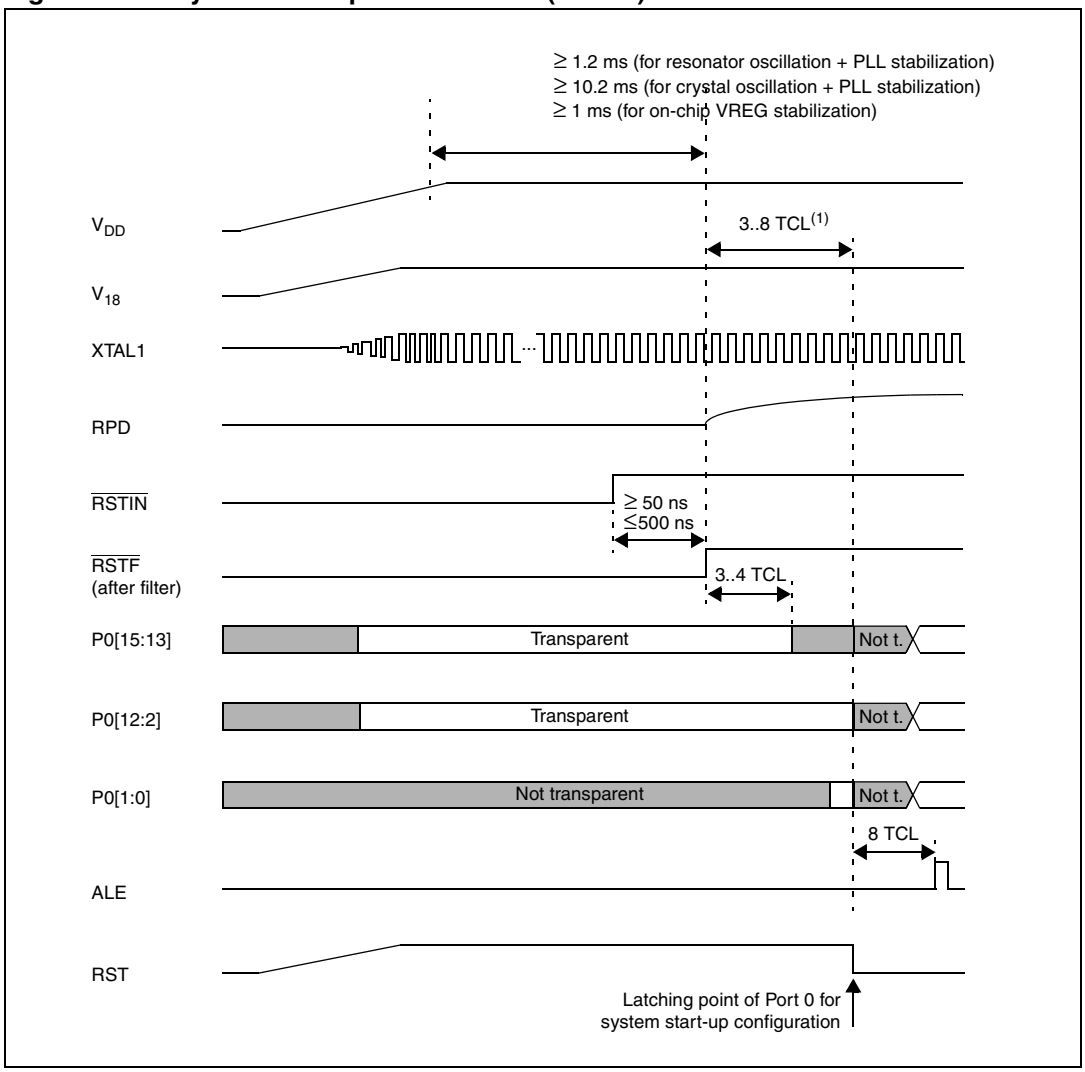


Figure 76. Asynchronous power-on reset ($\overline{EA}$ = 0)



1. Three to eight TCL depending on clock source selection.

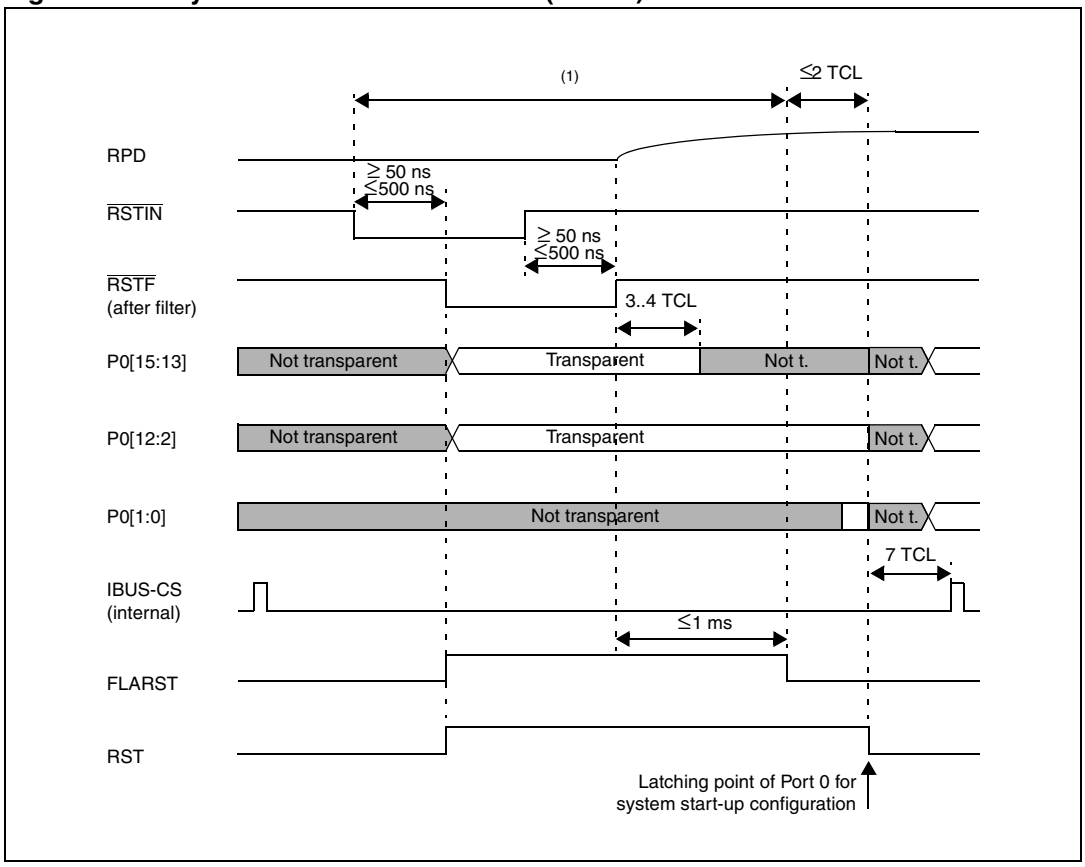
20.2.2 Hardware reset

An asynchronous reset is 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 20.7: Reset circuitry on page 233* and in *Figure 88*, *Figure 89* and *Figure 91*. Asynchronous resets occur when the $\overline{RSTIN}$ pin is low and the RPD pin is detected (or becomes) low.

20.2.3 Exit from asynchronous reset state

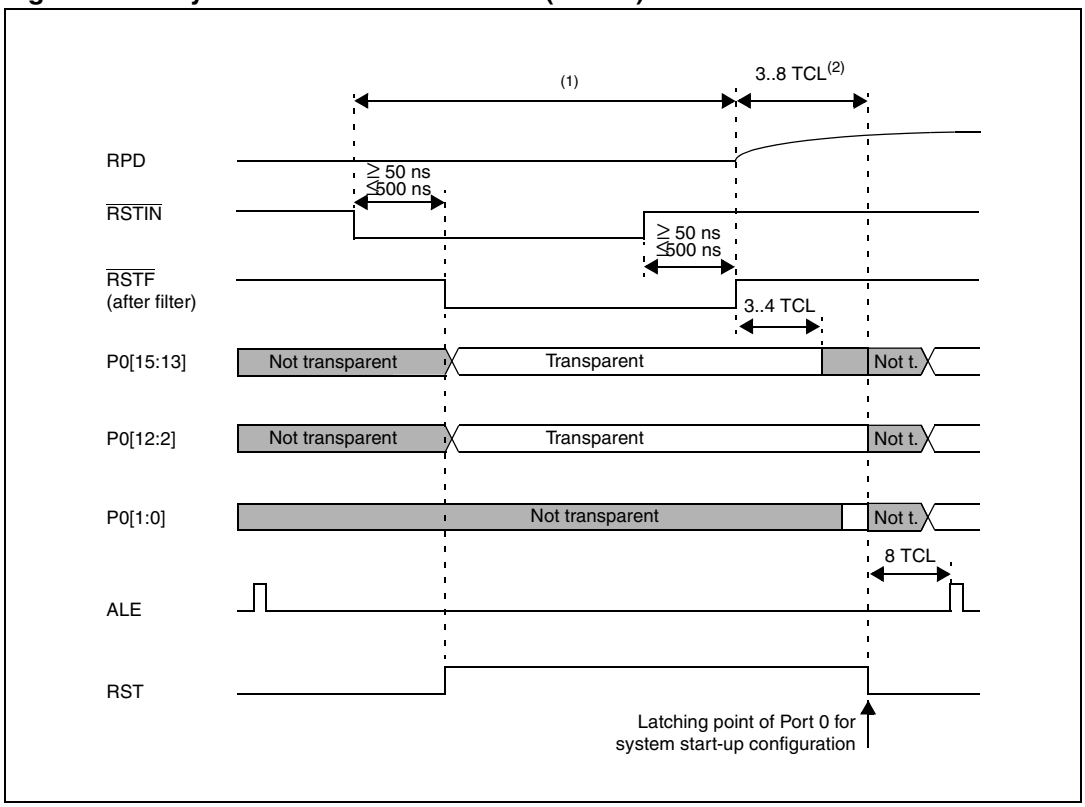
When the $\overline{RSTIN}$ pin is pulled high, the device restarts. If the internal Flash is used, restarting occurs after the embedded Flash initialization routine is completed. The system configuration is latched from Port 0. ALE, $\overline{RD}$ and $\overline{WR}$/$\overline{WRL}$ pins are driven to their inactive level. The ST10F296E starts program execution from memory location 00'0000h in code segment 0. This starting location typically points to the general initialization routine. Timing of asynchronous hardware reset sequences are summarized in *Figure 77* and *Figure 78*.

Figure 77. Asynchronous hardware reset ($\overline{EA}$ = 1)



1. Longer than Port 0 settling time + PLL synchronization (if needed, that is P0(15:13) changed) Longer than 500 ns to take account of input filter on $\overline{RSTIN}$ pin.

Figure 78. Asynchronous hardware reset ($\overline{EA}$ = 0)



1. Longer than Port 0 settling time + PLL synchronization (if needed, that is P0(15:13) changed). Longer than 500 ns to take account of input filter on $\overline{RSTIN}$ pin.
2. Three to eight TCL depending on clock source selection.

20.3 Synchronous reset (warm reset)

A synchronous reset is triggered when the $\overline{RSTIN}$ pin is pulled low while the RPD pin is at high level. To activate the internal reset logic of the device, the $\overline{RSTIN}$ pin must be held low, at least, during 4 TCL (2 CPU clock periods). Refer to *Section 20.1: Input filter on page 214* for details on minimum reset pulse duration. The I/O pins are set to high impedance and the $\overline{RSTOUT}$ pin is driven low. Once the $\overline{RSTIN}$ level is detected, a short duration of 12 TCL maximum (6 CPU clock periods) elapses, during which time pending internal hold states are cancelled and the current internal access cycle (if any) is completed. The external bus cycle is aborted. The internal pull-down of $\overline{RSTIN}$ pin is activated if bit BDRSTEN of the SYSCON register was previously set by software. Note that this bit is always cleared at power-on or after a reset sequence.

20.3.1 Short and long synchronous reset

Once the first 16 TCL elapse (4 TCL + 12 TCL), the internal reset sequence, of 1024 TCL cycles, starts. When it is finished and when an additional 8 TCL have elapsed, the level of the $\overline{\text{RSTIN}}$ pin is sampled (after the filter, see $\overline{\text{RSTF}}$ in *Figure 75*, *Figure 76*, *Figure 77*, and *Figure 78*). If the $\overline{\text{RSTIN}}$ pin is high, a short reset is flagged (see *Section 19: Watchdog timer* for details on reset flags). If the $\overline{\text{RSTIN}}$ pin is low, a long reset is flagged. The major difference between long and short resets is that during a long reset, P0(15:13) also become transparent, so it is possible to change the clock options.

Warning: **When there is a short pulse on the $\overline{\text{RSTIN}}$ pin, and when a bidirectional reset is enabled, the $\overline{\text{RSTIN}}$ pin is held low by the internal circuitry. At the end of 1024 TCL cycles, the $\overline{\text{RTSIN}}$ pin is released, but due to the presence of the input analog filter, the internal input reset signal ($\overline{\text{RSTF}}$ in *Figure 75*, *Figure 76*, *Figure 77*, and *Figure 78*) is released after it (50 to 500 ns after). This delay corresponds with the additional 8 TCL. At the end of this delay, the internal input reset line ($\overline{\text{RSTF}}$) is sampled to elucidate if the reset event is short or long.**

Short or long reset events

- If 8 TCL delay is > 500 ns (F_{CPU} < 8 MHz), the reset event is always recognized as short.
- If 8 TCL delay is < 500 ns (F_{CPU} > 8 MHz), the reset event could be recognized as either short or long, depending on the real filter delay (between 50 and 500 ns) and the CPU frequency. If $\overline{\text{RSTF}}$ samples high, a short reset is recognized. If $\overline{\text{RSTF}}$ samples low, a long reset is recognized. Once the 8 TCL delay has elapsed with a long reset, the P0(15:13) pins become transparent, and the system clock can be re-configured. After the internal $\overline{\text{RSTF}}$ signal becomes high, Port 0 returns 3-4 TCL which are not transparent.

The P0 pins become transparent and Port 0 returns 3-4 TCL which are not transparent when a unidirectional reset is selected and when the $\overline{\text{RSTIN}}$ pin is held low untill the end of an internal sequence (1024 TCL + max 16 TCL) and released at that time.

Note: When the device runs with a CPU frequency lower than 40 MHz, the minimum valid reset pulse recognized by the CPU (4 TCL) may be longer than the minimum analog filter delay (50 ns). Consequently, a short reset pulse may not be filtered by the analog input filter. However, this pulse is not long enough to trigger a CPU reset (as it is shorter than 4 TCL). It generates a Flash reset, but, not a system reset. In this condition, the Flash always answers with FFFFh, which leads to an illegal opcode and consequently a trap event is generated.

20.3.2 Exit from synchronous reset state

The reset sequence is extended until the $\overline{\text{RSTIN}}$ level becomes high. It is also internally prolonged by the Flash initialization when $\overline{\text{EA}}$ = 1 (internal memory selected). Then, the code execution restarts. The system configuration is latched from Port 0, and the ALE, $\overline{\text{RD}}$ and $\overline{\text{WR}}$/$\overline{\text{WRL}}$ pins are driven to their inactive level. The device starts program execution from memory location 00'0000h in code segment 0. This starting location typically points to the general initialization routine.

Figure 79 and *Figure 80* show the timing of synchronous reset sequences when booting from internal or external memory respectively. They emphasize a short reset event degenerating into a long reset.

Figure 81 and *Figure 82* shows the timing of a typical synchronous long reset when booting from internal or external memory respectively.

20.3.3 Synchronous reset and the RPD pin

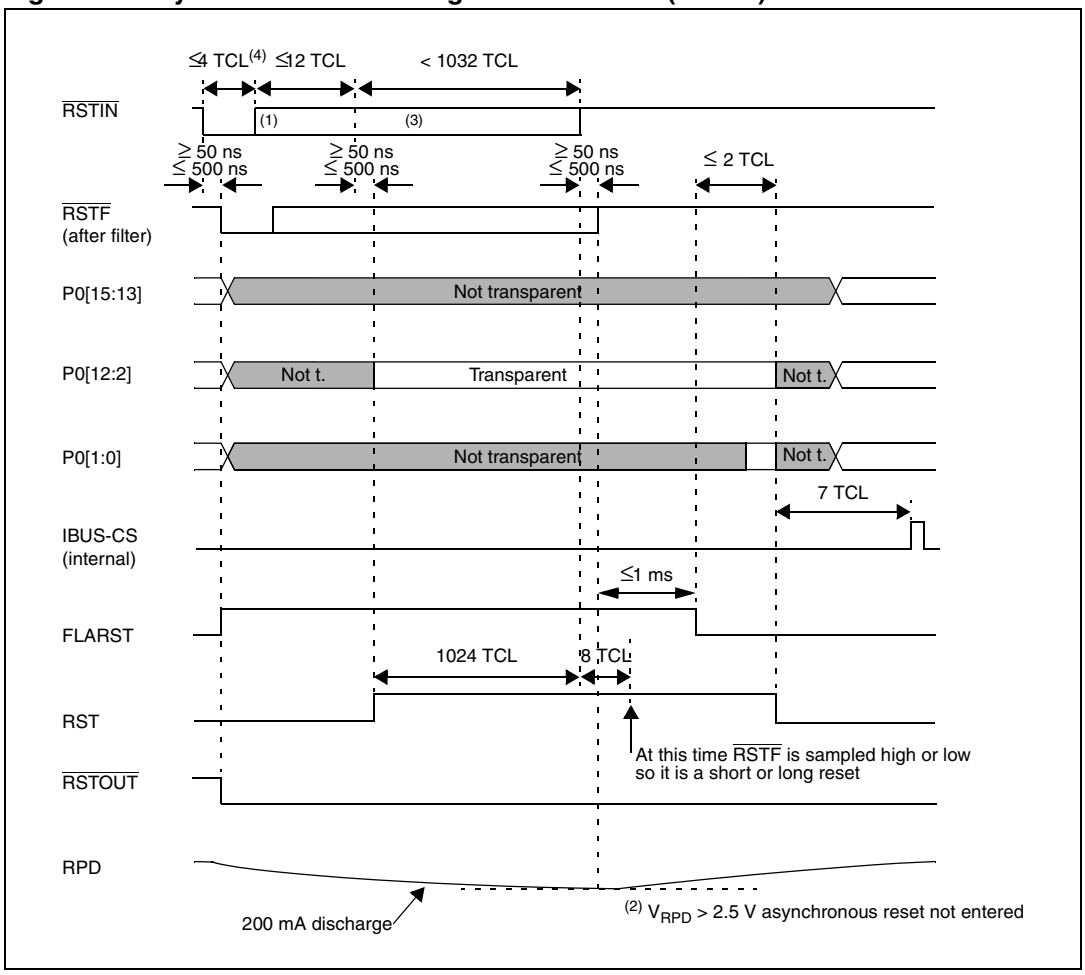
When the $\overline{\text{RSTIN}}$ pin is pulled low (by external hardware or as a consequence of a bidirectional reset), the RPD internal weak pull-down is activated. The external capacitance (if any) on the RPD pin is slowly discharged through the internal weak pull-down. If the voltage level on the RPD pin reaches the input low threshold (c. 2.5 V), the reset event becomes immediately asynchronous.If a short or long hardware reset occurs, the situation illustrated in *Figure 77* takes place.

If RPD rises above the input threshold, the asynchronous reset is completed. To complete a synchronous reset normally, the capacitance must be big enough to maintain the voltage on the RPD pin sufficiently high for the duration of the internal reset sequence.

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

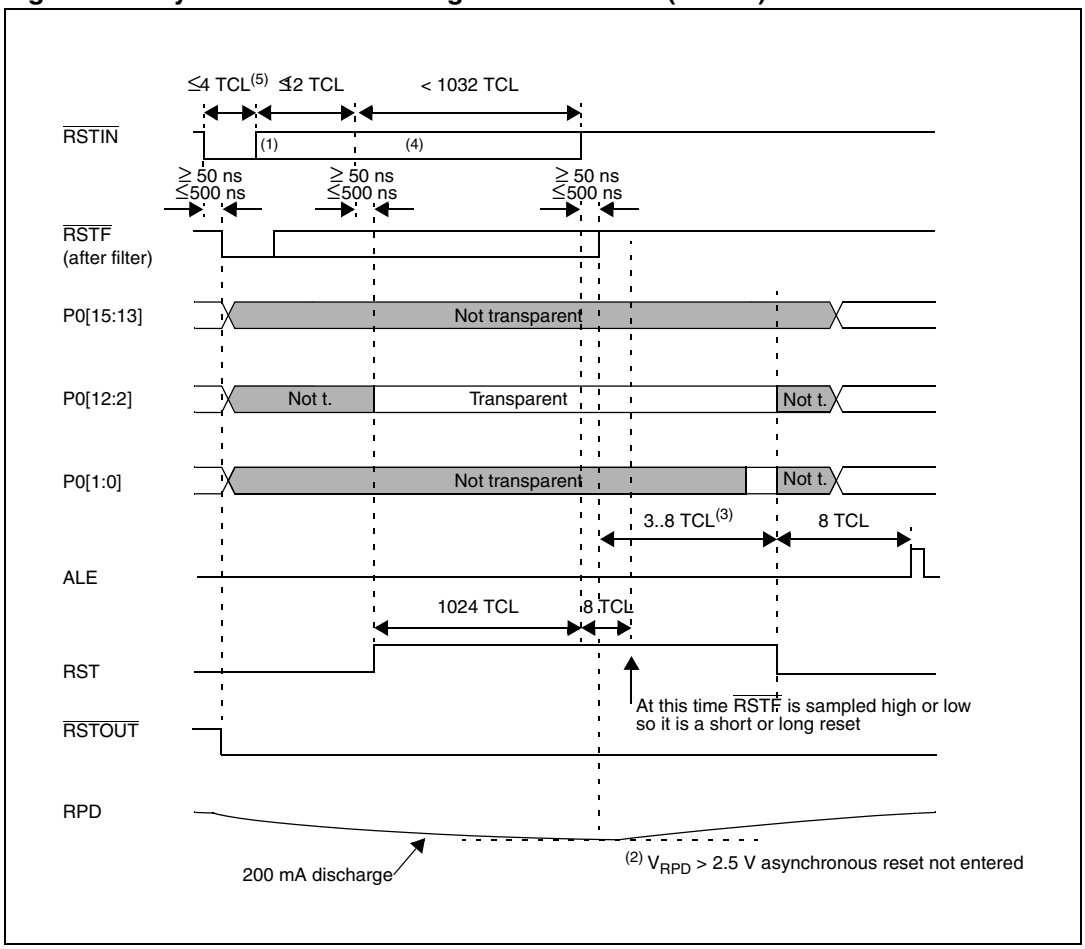
The signal that makes that RPD status transparent under reset is the internal $\overline{\text{RSTF}}$ (after the noise filter).

Figure 79. Synchronous short/long hardware reset ($\overline{EA}$ = 1)



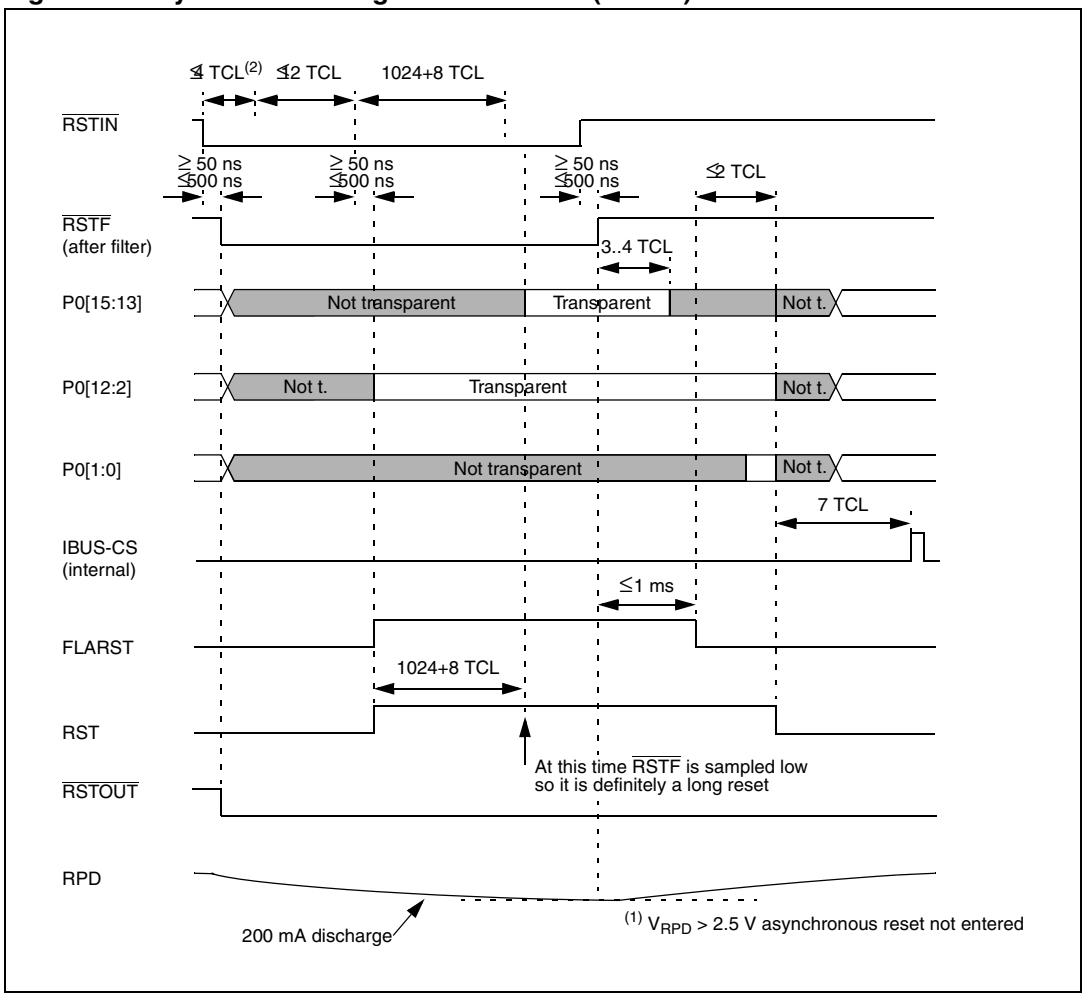
1. $\overline{RSTIN}$ assertion can be released here. See *Section 21.1: Idle mode on page 240* for details on minimum pulse duration.
2. If RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation) during the reset condition ($\overline{RSTIN}$ low), an asynchronous reset is entered immediately.
3. The $\overline{RSTIN}$ pin is pulled low if the BDRSTEN bit (of the SYSCON register) was previously set by software. The BDRSTEN bit is cleared after reset.
4. The minimum $\overline{RSTIN}$ low pulse duration must be longer than 500 ns, to guarantee the pulse is not masked by the internal filter (see *Section 21.1: Idle mode on page 240*).

Figure 80. Synchronous short/long hardware reset ($\overline{EA}$ = 0)



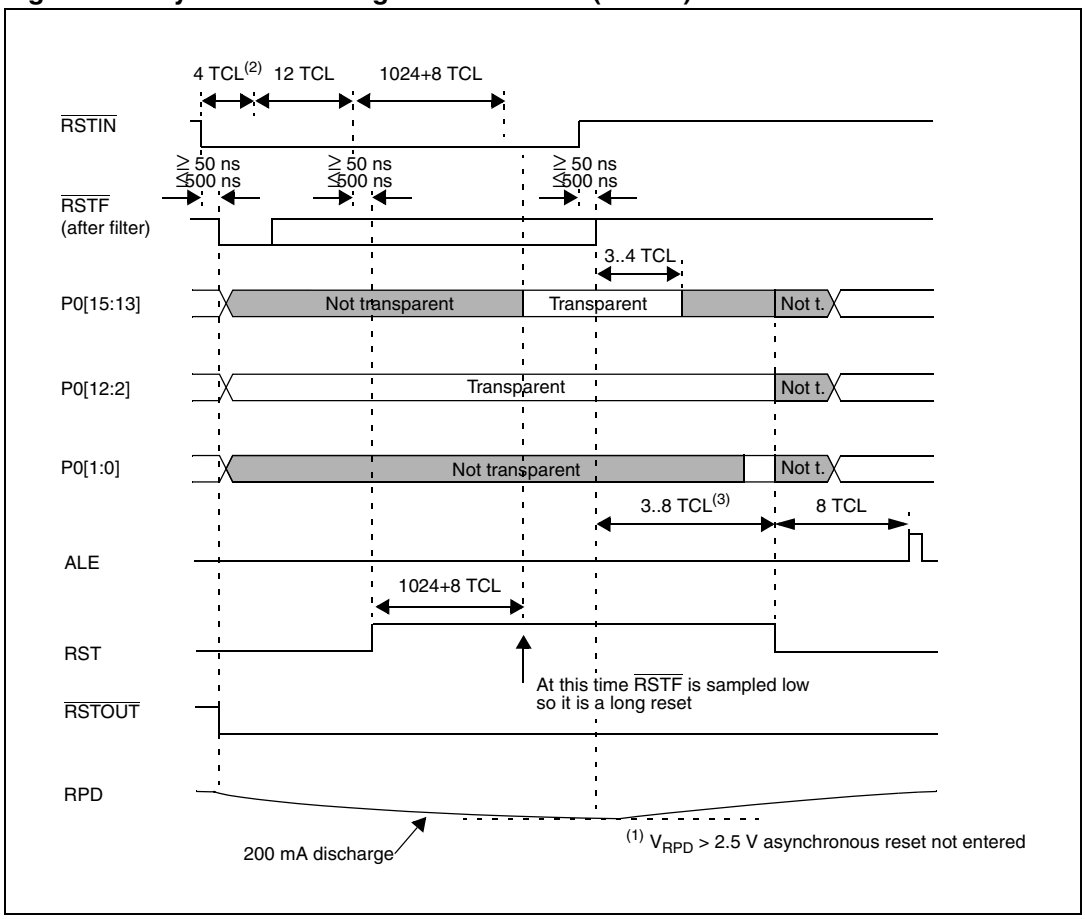
1. $\overline{RSTIN}$ assertion can be released here. See *Section 21.1: Idle mode on page 240* for details on minimum pulse duration.
2. If RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation) during the reset condition ($\overline{RSTIN}$ low), an asynchronous reset is entered immediately.
3. Three to eight TCL depending on clock source selection.
4. The $\overline{RSTIN}$ pin is pulled low if the BDRSTEN bit (of the SYSCON register) was previously set by software. The BDRSTEN bit is cleared after reset.
5. The minimum $\overline{RSTIN}$ low pulse duration must be longer than 500 ns, to guarantee the pulse is not masked by the internal filter (see *Section 21.1: Idle mode on page 240*).

Figure 81. Synchronous long hardware reset ($\overline{EA}$ = 1)



1. If RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation) during the reset condition ($\overline{RSTIN}$ low), an asynchronous reset is entered immediately. Even if RPD returns above teh threshold, the reset is taken as asychronous.
2. The minimum $\overline{RSTIN}$ low pulse duration must be longer than 500 ns, to guarantee the pulse is not masked by the internal filter (see *Section 21.1: Idle mode on page 240*).

Figure 82. Synchronous long hardware reset ($\overline{EA}$ = 0)



1. If RPD voltage drops below the threshold voltage (about 2.5 V for 5 V operation) during the reset condition ($\overline{RSTIN}$ low), an asynchronous reset is entered immediately.
2. The minimum $\overline{RSTIN}$ low pulse duration must be longer than 500 ns, to guarantee the pulse is not masked by the internal filter (see *Section 21.1: Idle mode on page 240*).
3. Three to eight TCL depending on clock source selection.

20.4 Software reset

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

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

A software reset is always taken as synchronous. There is no influence on software reset behavior with RPD status. If a bidirectional reset is selected, a software reset event pulls the $\overline{RSTIN}$ pin low. This occurs only if RPD is high. If RPD is low, the $\overline{RSTIN}$ pin is not pulled low even though a bidirectional reset is selected.

See *Figure 83* and *Figure 84* which shows unidirectional software reset timing. See *Figure 85*, *Figure 86*, and *Figure 87* for bidirectional software reset timing.

20.5 Watchdog timer reset

When the watchdog timer is not disabled during initialization, or if it is not serviced regularly during program execution, it overflows and triggers the reset sequence.

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

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

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

A watchdog reset is always taken as synchronous. There is no influence on watchdog reset behavior with RPD status. If a bidirectional reset is selected, a watchdog reset event pulls the $\overline{\text{RSTIN}}$ pin low. This occurs only if RPD is high. If RPD is low, the $\overline{\text{RSTIN}}$ pin is not pulled low even though a bidirectional reset is selected.

See *Figure 83* and *Figure 84* which shows unidirectional software reset timing. See *Figure 85*, *Figure 86*, and *Figure 87* for bidirectional software reset timing.

Figure 83. Software/watchdog timer unidirectional reset ($\overline{\text{EA}}$ = 1)

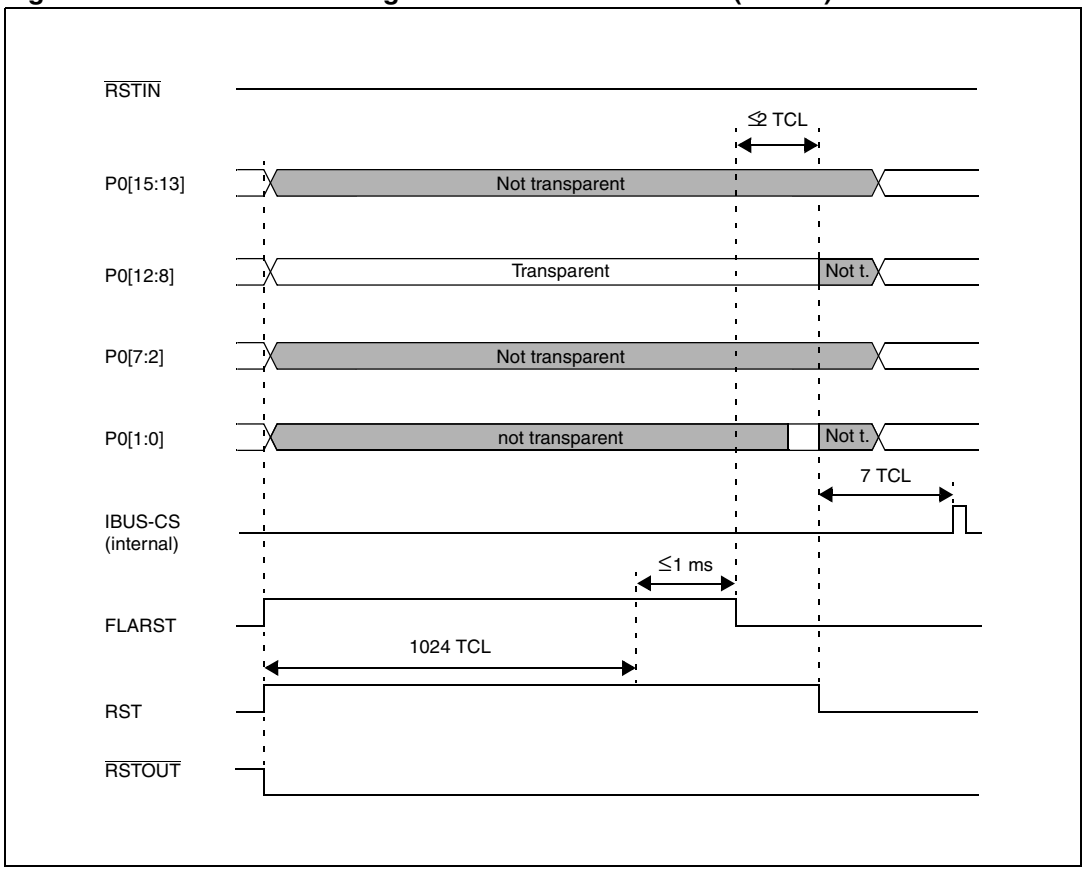
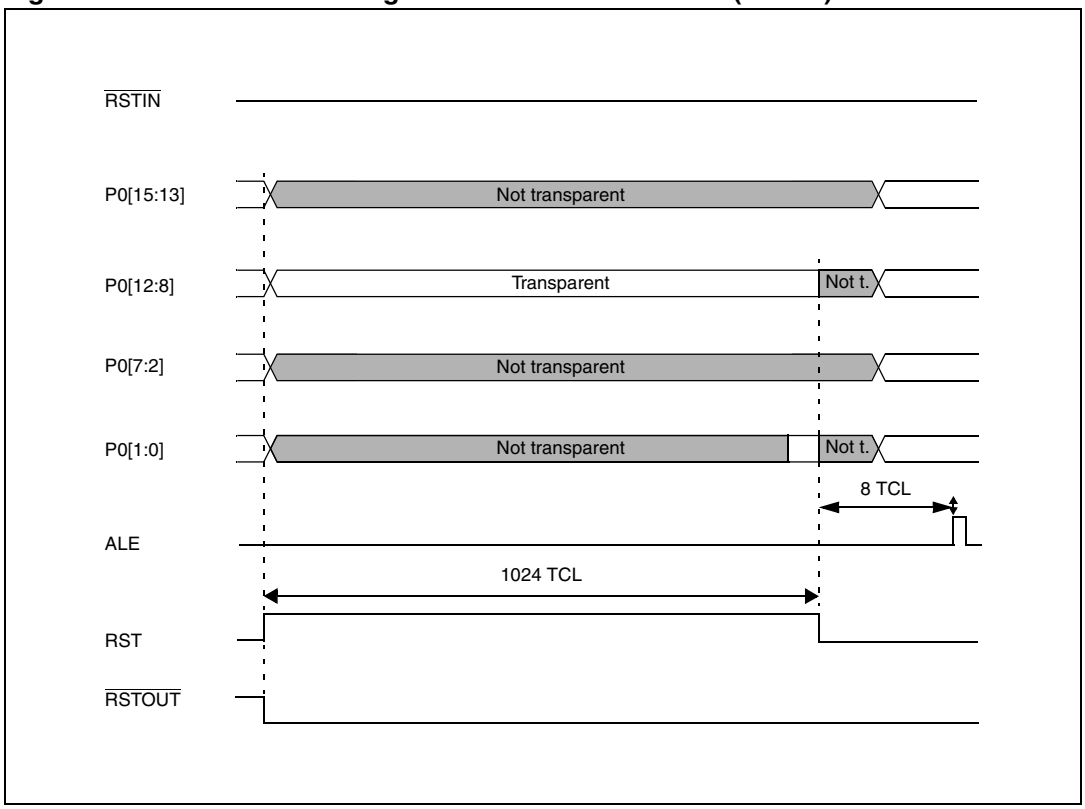


Figure 84. Software/watchdog timer unidirectional reset ($\overline{EA}$ = 0)



20.6 Bidirectional reset

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

The bidirectional reset function is useful when external devices require a reset signal, but, it cannot be connected to the $\overline{\text{RSTOUT}}$ pin, because the $\overline{\text{RSTOUT}}$ signal continues during initialization. In this case, the external memory can run the initialization routine before the execution of the EINIT instruction.

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

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

- If $\overline{\text{RSTF}}$ is sampled low (8 TCL periods after the internal reset sequence completion, see *Figure 79* and *Figure 80*) after a short synchronous bidirectional hardware reset, the short reset becomes a long reset. On the contrary, if $\overline{\text{RSTF}}$ is sampled high, the device simply exits reset state.
- After a software or watchdog bidirectional reset, the device exits from reset. If $\overline{\text{RSTF}}$ remains low for at least 4 TCL periods after exiting reset (minimum time to recognize a short hardware reset, see *Figure 85* and *Figure 86*), the software or watchdog reset become a short hardware reset. On the contrary, if $\overline{\text{RSTF}}$ remains low for less than 4 TCL, the device exits the reset state.

The bidirectional reset is not effective when RPD is held low or when a software or watchdog reset event occurs. On the contrary, if a software or watchdog bidirectional reset event is active and RPD becomes low, the $\overline{\text{RSTIN}}$ pin is immediately released, while the internal reset sequence is completed regardless of the RPD status change (1024 TCL).

Note: The bidirectional reset function is disabled by any reset sequence (when the BDRSTEN bit of the SYSCON register is cleared). To be activated again, it must be enabled during the initialization routine.

20.6.1 WDTCON flags

When a bidirectional reset is enabled, a short reset may degenerate into a long reset due to the presence of the internal filter on the $\overline{\text{RSTIN}}$ pin (see *Section 20.3.1: Short and long synchronous reset on page 221*). When the $\overline{\text{RSTIN}}$ pin is released, the internal signal after the filter (see $\overline{\text{RSTF}}$ in *Figure 75* to *Figure 78*) is delayed, so $\overline{\text{RSTIN}}$ remains active (low) for a while. Consequently, a short reset may be recognized as a long reset, depending on the internal clock speed.

When either a software or watchdog bidirectional reset event occurs, the $\overline{\text{RSTIN}}$ pin is released (at the end of the internal reset sequence), the $\overline{\text{RSTF}}$ internal signal (after the filter) remains low for a while, and $\overline{\text{RSTIN}}$ is recognized as high or low. Eight TCL after completion of the internal sequence, the level of the $\overline{\text{RSTF}}$ signal is sampled. If it is recognized as low, a hardware reset sequence starts, the WDTCON register flags this event, and masks the previous one (software or watchdog reset). Typically, a short hardware reset is recognized, unless the $\overline{\text{RSTIN}}$ pin (and consequently the internal $\overline{\text{RSTF}}$ signal) is held sufficiently low by the external hardware to inject a long hardware reset. The initialization routine is then unable to recognize a software or watchdog bidirectional reset event, since a different source is flagged inside the WDTCON register. This phenomenon does not occur when internal Flash is selected during reset ($\overline{\text{EA}}$ = 1), since the initialization of the Flash itself extends the internal reset duration beyond the filter delay.

Figure 85, *Figure 86*, and *Figure 87* show the timing for software and watchdog timer bidirectional reset events. *Figure 87* shows the degeneration into a hardware reset.

Figure 85. Software/watchdog timer bidirectional reset ($\overline{\text{EA}}$ = 1)

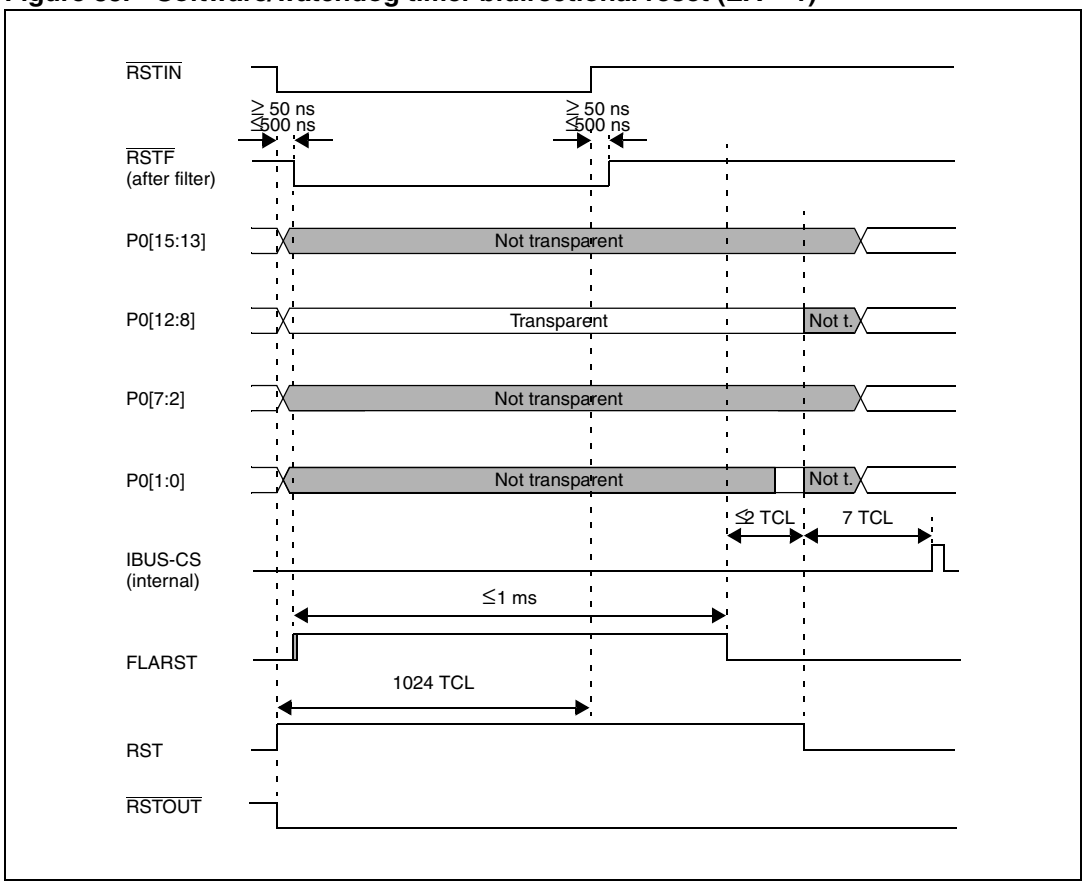


Figure 86. Software/watchdog timer bidirectional reset ($\overline{EA}$ = 0)

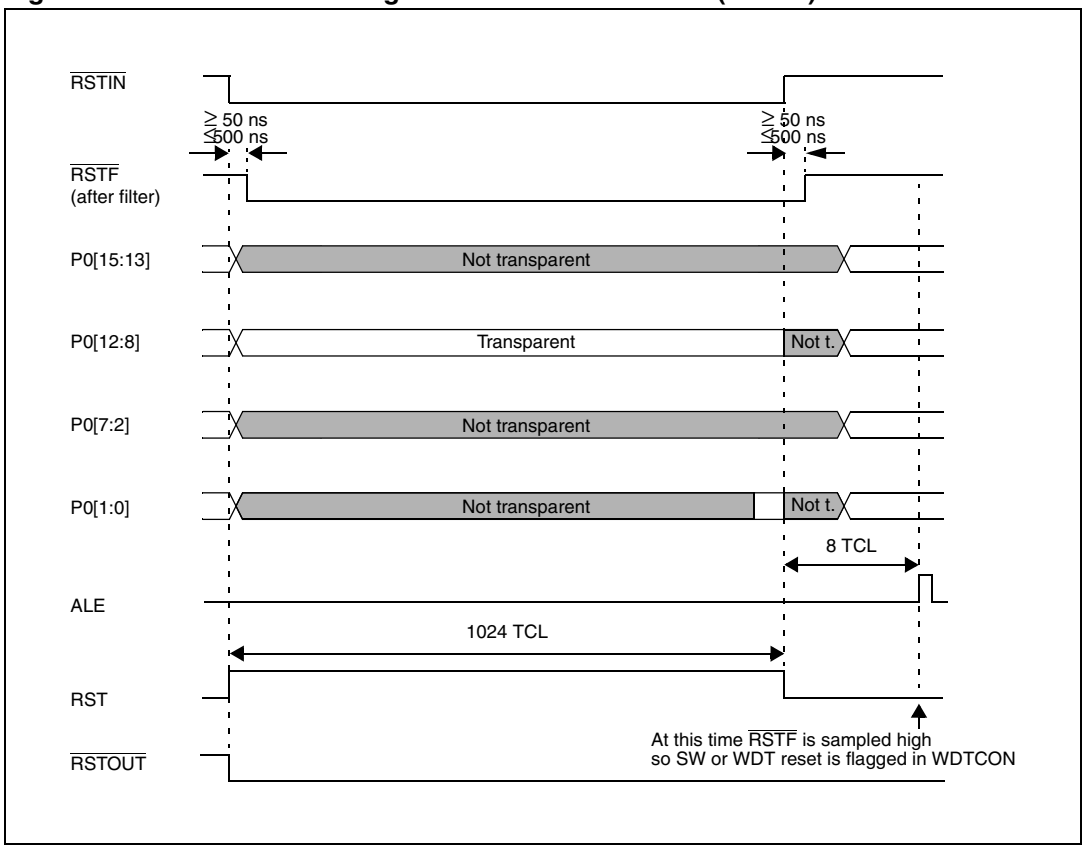
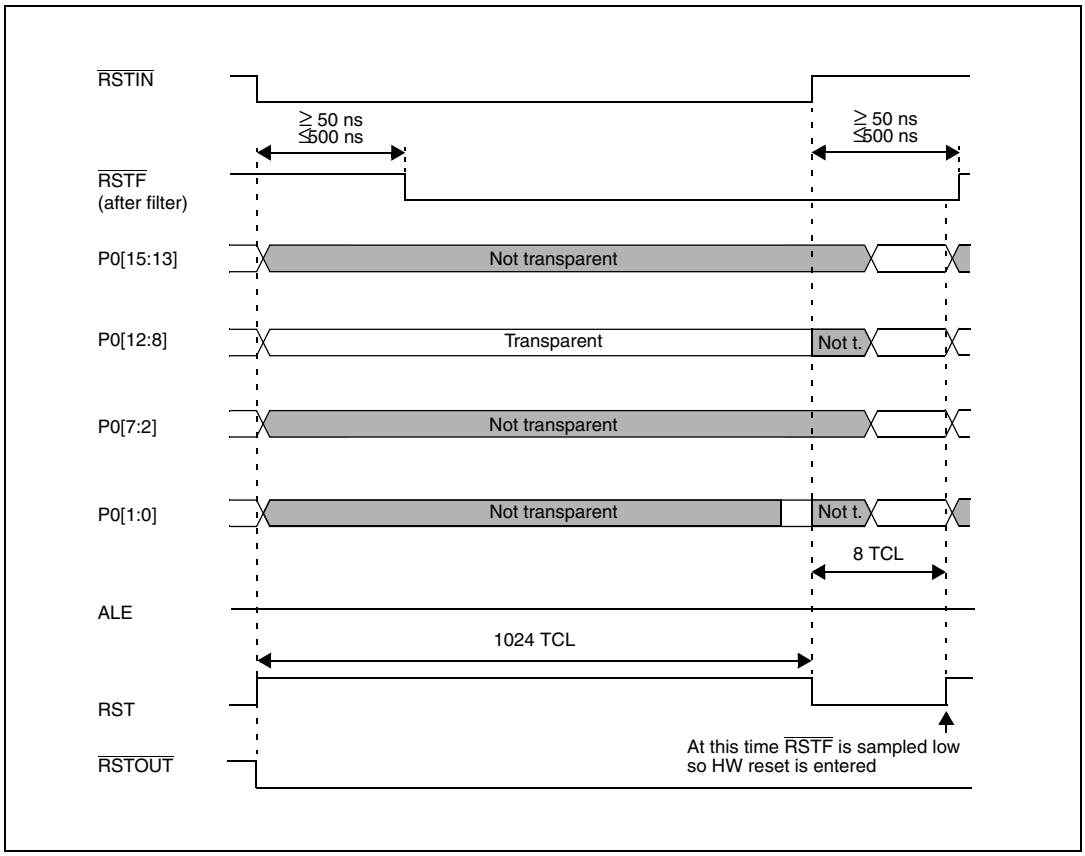


Figure 87. Software/watchdog timer bidirectional reset ($\overline{EA}$ = 0) followed by a hardware reset



20.7 Reset circuitry

The internal reset circuitry is described in *Figure 91*. The $\overline{\text{RSTIN}}$ pin provides an internal pull-up resistor of 50 kΩ to 250 kΩ (the minimum reset time must be calculated using the lowest value).

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

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

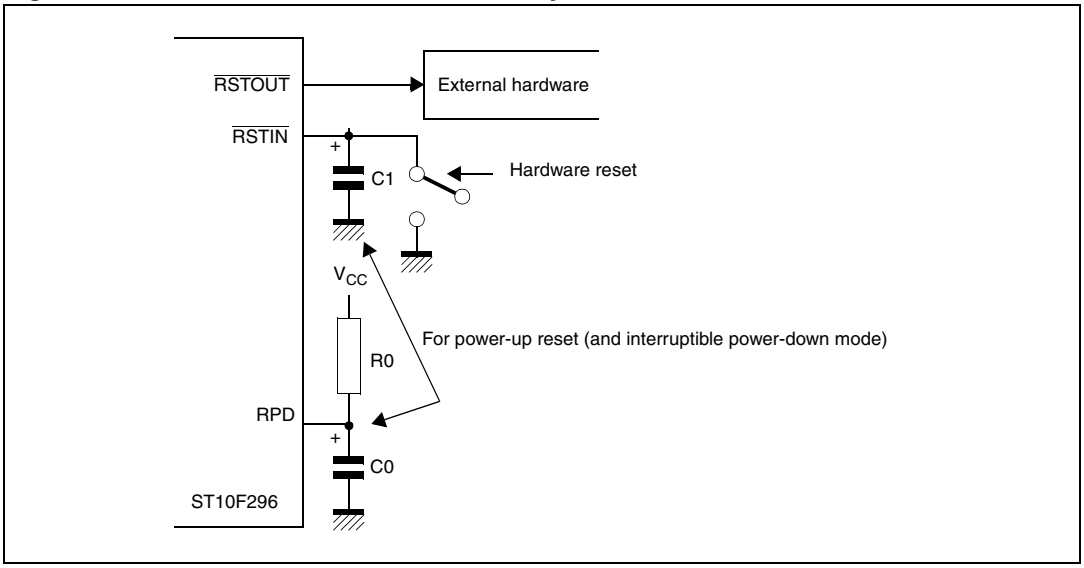
In this case, the external memory can run codes before the EINIT instruction is executed (end of initialization). The $\overline{\text{RSTOUT}}$ pin is pulled high only when EINIT is executed.

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

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

To ensure correct power-up reset with controlled supply current consumption, in particular if the clock signal requires a long period of time to stabilize, an asynchronous hardware reset is required during power-up. Consequently, it is recommended to connect the external R0-C0 circuit shown in *Figure 88* to the RPD pin. On power-up, the logical low level on the RPD pin, forces an asynchronous hardware reset when $\overline{\text{RSTIN}}$ is asserted low. The external pull-up, R0, then charges the capacitor, C0. Note that an internal pull-down device on the RPD pin is turned on when the $\overline{\text{RSTIN}}$ pin is low, and causes the external capacitor, C0, to begin discharging at a typical rate of 100-200 µA. With this mechanism, after power-up reset, short low pulses applied on $\overline{\text{RSTIN}}$ produce synchronous hardware resets. If $\overline{\text{RSTIN}}$ is asserted longer than the time needed for C0 to be discharged by the internal pull-down device, the device is forced into an asynchronous reset. This mechanism ensures recovery from catastrophic failures.

Figure 88. Minimum external reset circuitry

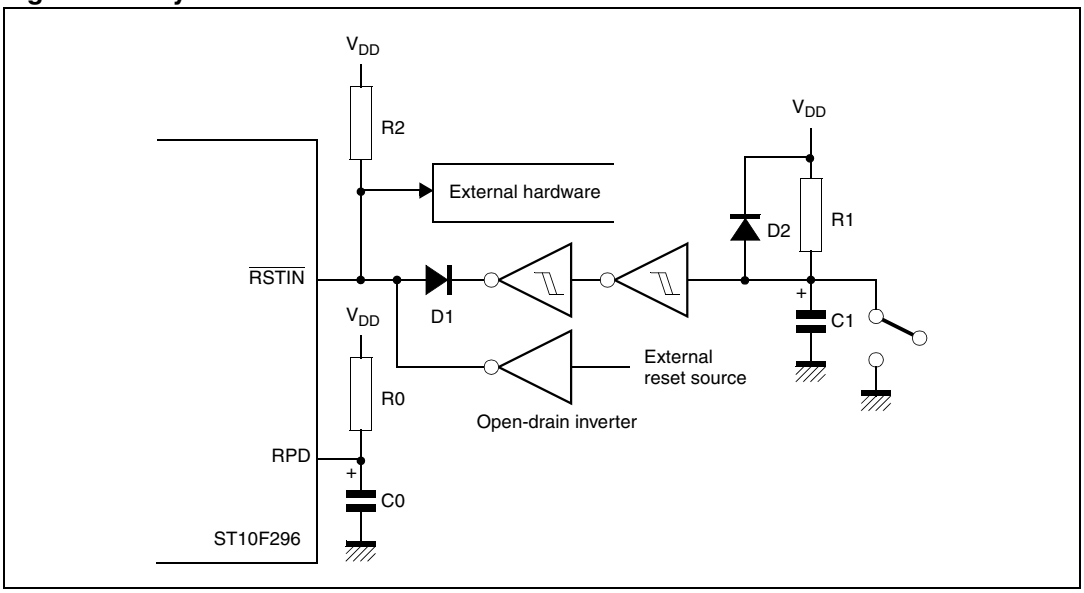


The minimum reset circuit of *Figure 88* is not adequate when the $\overline{\text{RSTIN}}$ pin is driven from the ST10F296E itself during software or watchdog triggered resets. This is because capacitor C1 keeps the voltage on the $\overline{\text{RSTIN}}$ pin above V_{IL} after the end of the internal reset sequence, thus triggering an asynchronous reset sequence.

Figure 89 shows an example of a reset circuit. The R1-C1 external circuit is used to generate power-up or manual reset and the R0-C0 circuit on RPD is used for power-up reset and to exit from power-down mode. Diode, D1, creates a wired-OR gate connection to the reset pin and may be replaced by an open-collector Schmitt trigger buffer. Diode, D2, provides a faster cycle time for repetitive power-on resets.

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

Figure 89. System reset circuit



20.8 Reset application examples

Figure 90 and *Figure 91* are timing diagrams that provide additional examples of bidirectional internal reset events (software and watchdog). They include the external capacitance charge and discharge transients. *Figure 89* shows the external circuit scheme.

Figure 90. Example of software or watchdog bidirectional reset ($\overline{EA}$ = 1)

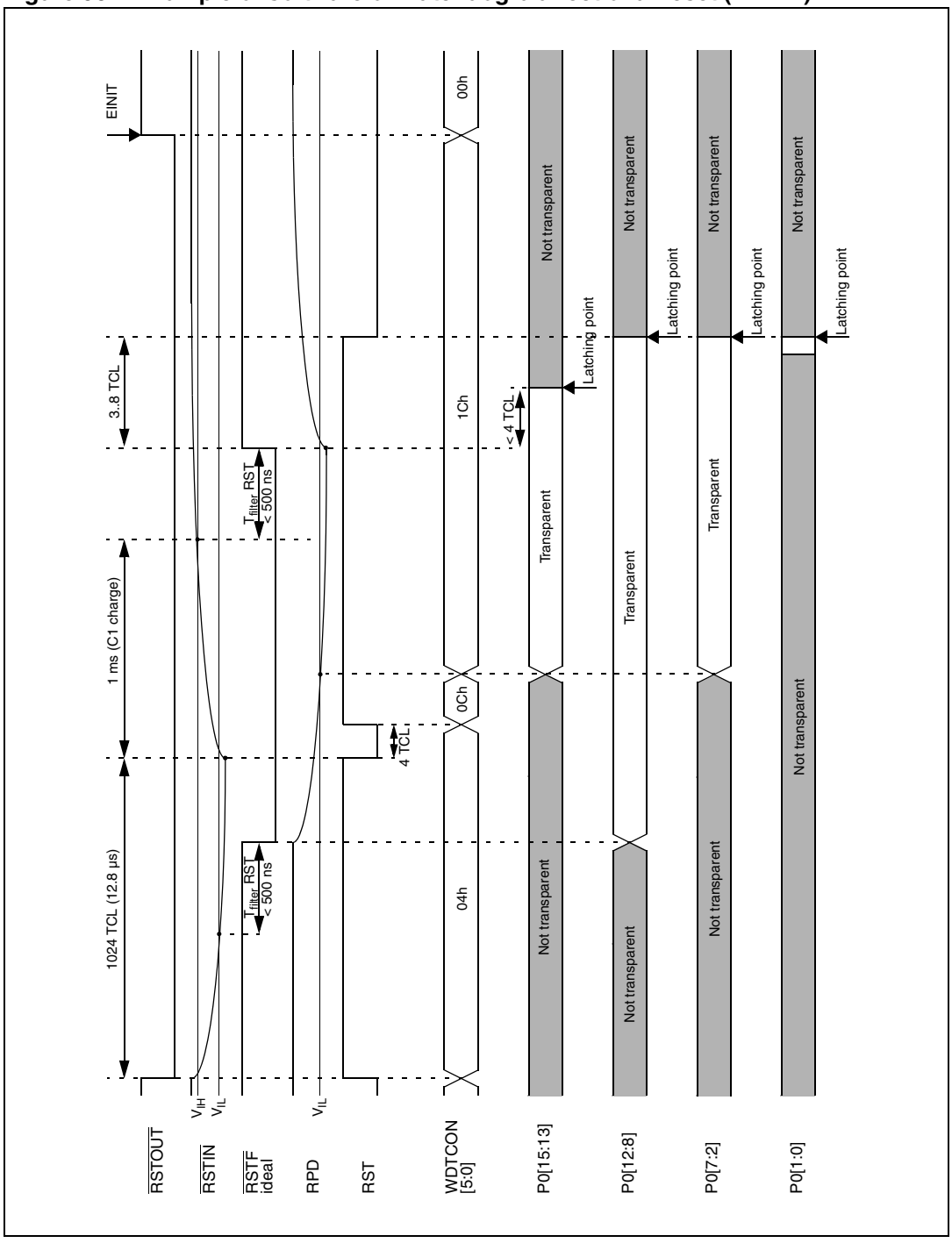
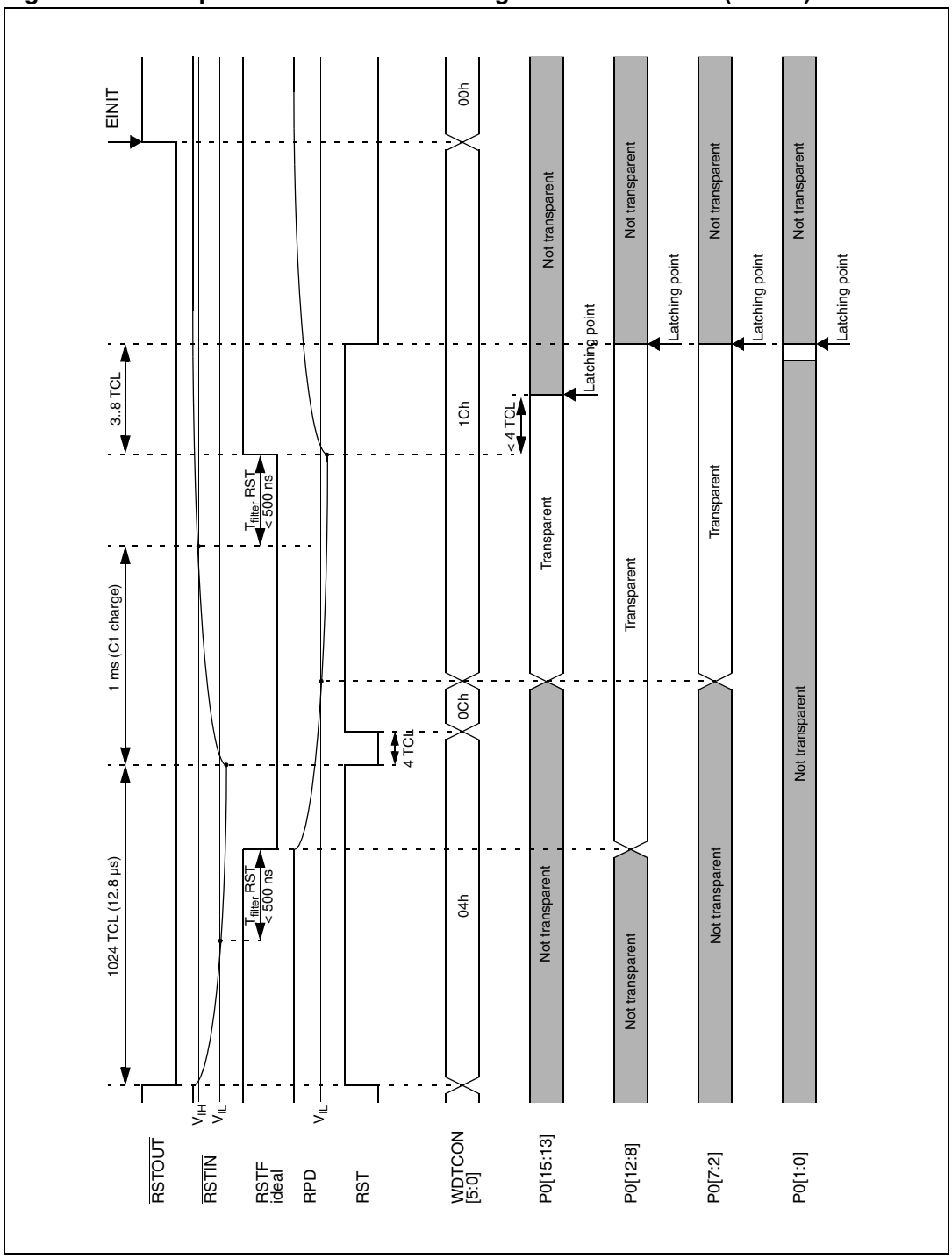


Figure 91. Example of software or watchdog bidirectional reset ($\overline{EA}$ = 0)



20.9 Reset summary

Table 132 summarizes the different reset events.

Table 132. Reset events summary

Event	RPD	$\overline{EA}$	Bidirectional	synchronous/ asynchronous	$\overline{RSTIN}$		WDTCON flags				
					Min	Max	PONR	LHWR	SHWR	SWR	WDTR
Power-on reset	0	0	N	Asynch.	1 ms (VREG) 1.2 ms (reson. + PLL) 10.2 ms (crystal + PLL)	-	1	1	1	1	0
	0	1	N	Asynch.	1 ms (VREG)	-	1	1	1	1	0
	1	x	x	Forbidden							
	x	x	Y	Not applicable							
Hardware reset (asynchronous)	0	0	N	Asynch.	500 ns	-	0	1	1	1	0
	0	1	N	Asynch.	500 ns	-	0	1	1	1	0
	0	0	Y	Asynch.	500 ns	-	0	1	1	1	0
	0	1	Y	Asynch.	500 ns	-	0	1	1	1	0
Short hardware reset (synchronous)(1)	1	0	N	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max (4 TCL, 500 ns)	0	0	1	1	0
	1	1	N	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max (4 TCL, 500ns)	0	0	1	1	0
	1	0	Y	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max (4 TCL, 500 ns)	0	0	1	1	0
					Activated by internal logic for 1024 TCL						
	1	1	Y	Synch.	Max (4 TCL, 500 ns)	1032 + 12 TCL + max (4 TCL, 500 ns)	0	0	1	1	0
					Activated by internal logic for 1024 TCL						
Long hardware reset (synchronous)	1	0	N	Synch.	1032 + 12 TCL + max (4 TCL, 500 ns)	-	0	1	1	1	0
	1	1	N	Synch.	1032 + 12 TCL + max(4 TCL, 500 ns)	-	0	1	1	1	0
	1	0	Y	Synch.	1032 + 12 TCL + max (4 TCL, 500 ns)	-	0	1	1	1	0
					Activated by internal logic only for 1024 TCL						
	1	1	Y	Synch.	1032 + 12 TCL + max (4 TCL, 500 ns)	-	0	1	1	1	0
					Activated by internal logic only for 1024 TCL						

Table 132. Reset events summary

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

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

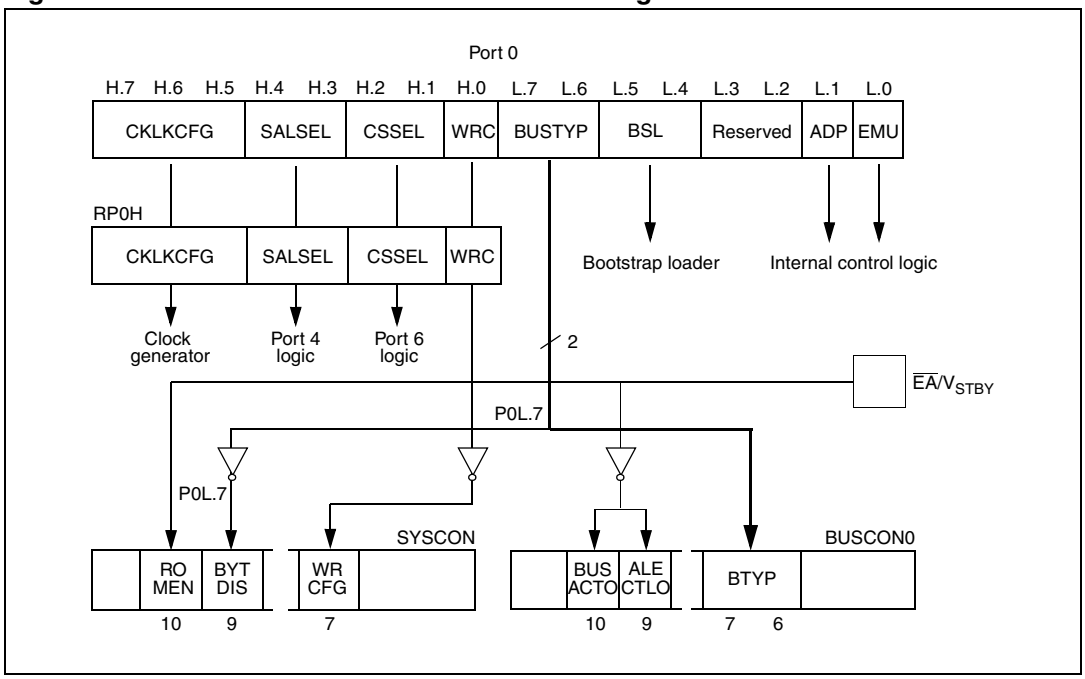
Table 133 and *Figure 92* shows the start-up configurations and some system features that are selected on reset sequences. *Table 133* describes the system configurations latched onto Port 0 in the six different reset modes. *Figure 92* summarizes the state of the Port 0 bits latched in the RP0H, SYSCON, and BUSCON0 registers.

Table 133. Latched configurations of Port 0 for the different reset events(1)

Sample event	Port 0															
	Clock options			Segm. addr. lines		Chip selects		$\overline{WR}$ config.	Bus type		Reserved	BSL	Reserved	Reserved	Adapt mode	EMU mode
	P0H.7	P0H.6	P0H.5	P0H.4	P0H.3	P0H.2	P0H.1	P0H.0	P0L.7	P0L.6	P0L.5	P0L.4	P0L.3	P0L.2	P0L.1	P0L.0
Software reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Watchdog reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Synchronous short hardware reset	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X
Synchronous long hardware reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Asynchronous hardware reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Asynchronous power-on reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

1. X: Pin is sampled; -: Pin is not sampled.

Figure 92. Port 0 bits latched into the different registers after reset



21 Power reduction modes

Three different power reduction modes with different levels of power saving have been implemented in the ST10F296E. In idle mode, the CPU is stopped, but, peripherals still operate. In power-down mode, both the CPU and peripherals are stopped. In stand-by mode, the main power supply (V_{DD}) can be turned off while a portion of the internal RAM remains powered via the dedicated power pin, V_{STBY}.

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

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

21.1 Idle mode

Idle mode is entered by running the IDLE protected instruction. CPU operation is stopped but, the peripherals continue to run.

Idle mode is terminated by any interrupt request. Whether the interrupt is serviced or not, the instruction following the IDLE instruction, is executed after a return from the interrupt instruction (RETI). The CPU then resumes normal programming.

Note that a PEC transfer keeps the CPU in idle mode. If the PEC transfer does not succeed, idle mode is terminated. The watchdog timer must be properly programmed to avoid any disturbance during idle mode.

21.2 Power-down mode

Power-down mode starts by running the PWRDN protected instruction. The internal clock is stopped, all MCU parts, including the watchdog timer, are put on hold. The only exception is the RTC, if it has been opportunely programmed, and consequently, the main oscillator circuits.

When the RTC module is used, and the device is in power-down mode, a reference clock is needed. Accordingly, the main oscillator is kept running (XTAL1/XTAL2 pins). In this way, the RTC continues counting using the main oscillator clock signal as a reference.

There are two different operating power-down modes:

- Protected mode
- Interruptible mode

The internal RAM contents can be preserved through the voltage that is supplied via the V_{DD} pins. To verify RAM integrity, some dedicated patterns may be written before entering power-down mode which must be checked after power-down is resumed.

Power-down mode is entered by executing the PWRDN instruction. Before entering it, the VREGOFF bit in the XMISC register must be set. In this way, as soon as the PWRDN command is executed, the main voltage regulator is turned off, and only the low power voltage regulator remains active.

Note: Leaving the main voltage regulator active during power-down may lead to unexpected behavior (example, CPU wake-up). Power consumption is also higher than that specified in Table 163: DC characteristics on page 295.

21.2.1 Protected power-down mode

This mode is selected when the PWDCFG bit of the SYSCON register is cleared. Protected power-down mode is only activated if the $\overline{NMI}$ pin is pulled low when executing the PWRDN instruction. This mode is only deactivated with an external hardware reset on the $\overline{RSTIN}$ pin.

21.2.2 Interruptible power-down mode

This mode is selected when the PWDCFG bit of the SYSCON register is set (see *Section 23: Register set on page 248*).

Interruptible power-down mode is only activated if all the enabled fast external interrupt pins are at their inactive level (see *Table 134: EXICON register description*).

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

EXICON 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
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

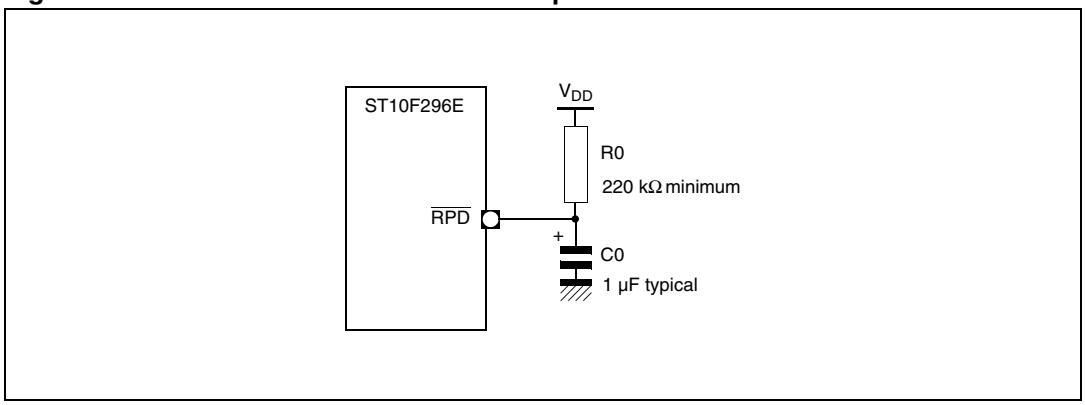
Table 134. EXICON register description

Bit	Bit name	Function
15-0	EXIxES	External interrupt x edge selection field (x = 7...0) 00: Fast external interrupts disabled (referred to as standard mode). The EXxIN pin is not taken into account for entering/exiting power-down mode. 01: Interrupt on positive edge (rising). Power-down mode is entered if EXiIN = 0 and exited if EXxIN = 1 (referred to as 'high' active level). 10: Interrupt on negative edge (falling). Power-down mode is entered if EXiIN = 1 and exited if EXxIN = 0 (referred to as 'low' active level). 11: Interrupt on any edge (rising or falling). Power-down mode is always entered and is exited if EXxIN level changes.

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 the CCxIC register) does not need to be set to bring the device out of power-down mode. An external RC circuit must be connected to the RPD pin, as shown in *Figure 93*.

Figure 93. External RC circuit on the $\overline{\text{RPD}}$ pin



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

This signal enables the internal oscillator and PLL circuitry. It also turns on the weak pull-down (see *Figure 94*).

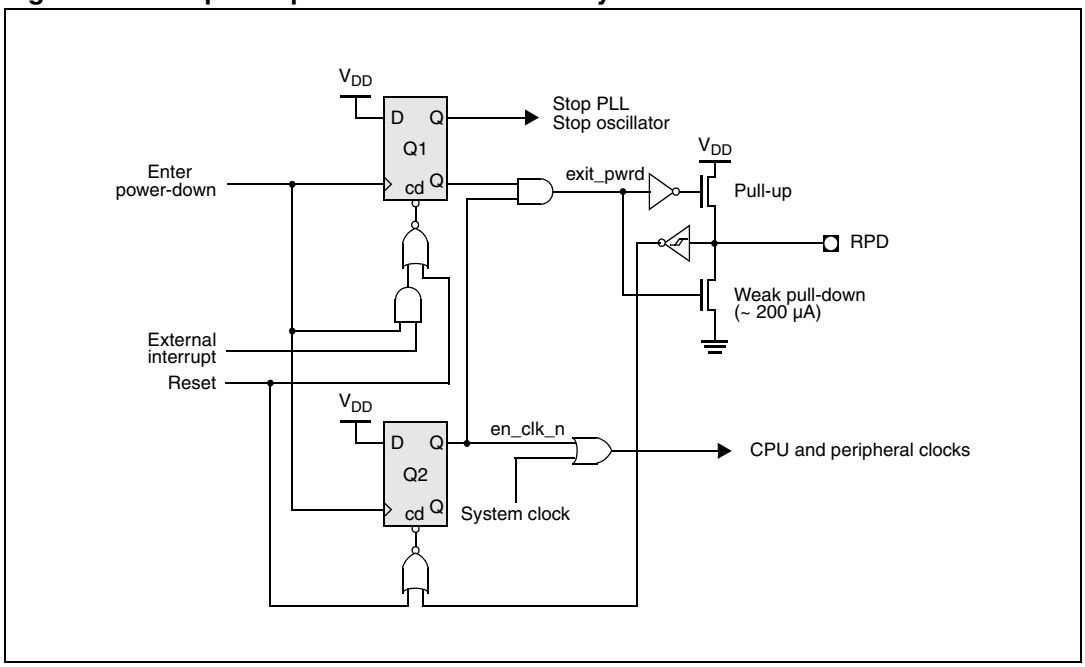
The discharge of the external capacitor provides a delay that allows the oscillator and PLL circuits to stabilize before the internal CPU and peripheral clocks are enabled. When the RPD voltage drops below the threshold voltage (about 2.5 V), the Schmitt trigger clears Q2 flip-flop, the CPU and peripheral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

If the interrupt is enabled (CCxIE bit = 1 in the CCxIC register) before entering power-down mode, the device executes the interrupt service routine and resumes execution after the PWRDN instruction (see note below).

If the interrupt is disabled, the device executes the instruction following the PWRDN instruction and the interrupt request flag remains set (using the CCxIR bit in the CCxIC register) until it is cleared by software.

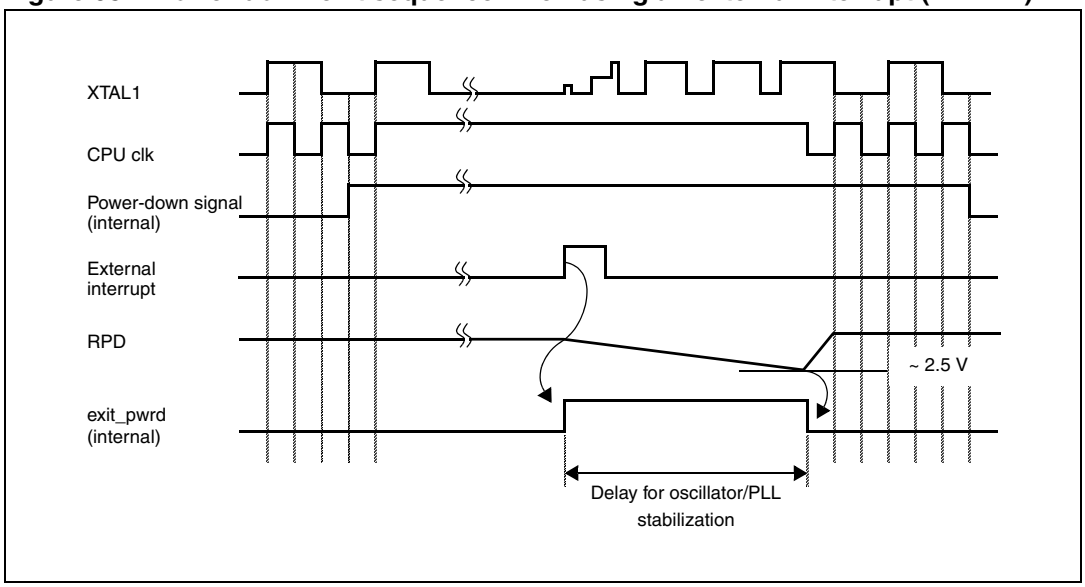
Note: Due to the internal pipeline, the instruction that follows the PWRDN instruction is executed before the CPU performs a call of the interrupt service routine when exiting power-down mode.

Figure 94. Simplified power-down exit circuitry



1. Legend:
exit_pwrd = exit power-down internal signal
en_clk_n = clock enable signal (negated: active low)

Figure 95. Power-down exit sequence when using an external interrupt (PLL x 2)



21.3 Standby mode

In stand-by mode, the RAM array is maintained powered through the dedicated pin, V_{STBY}, when the main power supply (V_{DD}) of the ST10F296E is turned off.

To enter stand-by mode, the device must be held under reset. In this way, the RAM is disabled (see XRAM2EN bit of XPERCON register, *Table 5*), and its digital interface is frozen to avoid any kind of data corruption. It is then possible to turn off the main V_{DD} provided that V_{STBY} is on.

A dedicated embedded low-power voltage regulator is implemented to generate the internal low voltage supply to bias the portion of XRAM (16 Kbytes).

In normal running mode (when V_{DD} is on), the V_{STBY} pin can be tied to V_{SS} during reset, to exercise the $\overline{EA}$ functionality associated with the same pin. The voltage supply for the circuits which are usually biased with V_{STBY} is granted by the active V_{DD}.

Standby mode can generate problems associated with the use of different power supplies in CMOS systems. Particular attention must be paid when the ST10F296E I/O lines are interfaced with other external CMOS integrated circuits. In standby mode, if the V_{DD} of the device falls below that of the output level forced by the I/O lines of the external integrated circuits, the device could be powered directly through the inherent diode existing on the device output driver circuit. The same is valid for the ST10F296E when it is interfaced to active/inactive communication buses during standby mode. Current injection can be generated through the inherent diode.

In addition, the sequence of turning on/off the different voltages could be critical for the system. The device standby mode current (I_{STBY}) may vary while the V_{DD} to V_{STBY} transition occurs (and vice versa) as some current flows between the V_{DD} and V_{STBY} pins. System noise on both the V_{DD} and V_{STBY} pins can increase this phenomenon.

21.3.1 Entering standby mode

To enter standby mode, the XRAM2EN bit in the XPERCON register must be cleared (this bit is automatically reset by any kind of reset event, see *Section 20: System reset*). This allows the RAM interface to be frozen immediately, thereby avoiding any data corruption. As a consequence of a reset event, the RAM power supply is switched to the internal low-voltage supply, V_{18SB} (derived from V_{STBY} through the low-power voltage regulator). The RAM interface remains frozen until the XRAM2EN bit is set again by the software initialization routine (at the next exit from V_{DD} power-on reset sequence).

When V_{18} falls (as a result of V_{DD} being turning off), the XRAM2EN bit is no longer be able to guarantee its content (logic 0), because the XPERCON register is powered by internal V_{18}. This does not generate a problem because the standby mode switching dedicated circuit continues to confirm that the RAM interface is freezing, irrespective of the XRAM2EN bit content. The XRAM2EN bit status is considered once more when the internal V_{18} starts again and replaces the internal stand-by reference V_{18SB}.

If internal V_{18} falls below the internal stand-by reference (V_{18SB}) by about 0.3 to 0.45 V when the XRAM2EN bit is set, the RAM supply switching circuit is inactive. If there is a temporary drop on the internal V_{18} voltage versus internal V_{18SB} during normal code execution, no spurious standby mode switching can occur (the RAM is not frozen and can still be accessed).

The ST10F296E core module, which generates the RAM control signals, is powered by the internal V_{18} supply. During turning off transient phase these control signals follow the V_{18}, while RAM is switched to V_{18SB} internal reference. A high level of RAM write strobe from the ST10F296E core (active low signal), may be low enough to be recognized as a logic 0 by the RAM interface (due to V_{18} being lower than V_{18SB}). The bus status may contain a valid address for the RAM and an unwanted data corruption may occur. For this reason, an extra interface, powered by the switched supply, is used to prevent the RAM from such potential corruption mechanisms.

Warning: During power-off phase, the external hardware must maintain a stable ground level on the $\overline{\text{RSTIN}}$ pin, with no glitches, to avoid spurious exits from reset status due to an unstable power supply.

21.3.2 Exiting standby mode

The procedure to exit standby mode consists of a standard power-on sequence where the RAM is powered through the V_{18SB} internal reference (derived from the V_{STBY} pin external voltage).

It is recommended to hold the device under reset ($\overline{\text{RSTIN}}$ pin forced low) until the external V_{DD} voltage pin is stable. At the beginning of the power-on phase, the device is maintained under reset by the internal low voltage detector circuit (implemented inside the main voltage regulator) until the internal V_{18} becomes higher than about 1.0 V. Despite this, **there is no warranty that the device stays under reset status if $\overline{\text{RSTIN}}$ is at high level during power ramp up.**

It is imperative that the external hardware guarantees a stable ground level on the $\overline{\text{RSTIN}}$ pin along the power-on phase, without any temporary glitches.

The external hardware is responsible for driving the $\overline{\text{RSTIN}}$ pin low until the V_{DD} is stable, even though the internal LVD is active. An additional time period of at least 1 ms is also requested to allow the internal voltage regulator to stabilize before releasing the $\overline{\text{RSTIN}}$ pin. This is necessary because the internal Flash has to begin its initialization phase (which starts when the $\overline{\text{RSTIN}}$ pin is released) with a stable V_{18}.

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

At this point, all voltages are stable, and the execution of the initialization routines can start. The XRAM2EN bit can be set and the RAM can be enabled.

21.4 Power reduction modes summary

Table 135 provides a summary of the different power reduction modes

Table 135. Power reduction modes summary

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

22 Programmable output clock divider

A specific register mapped on the XBus allows the division factor on the CLKOUT signal (P3.15) to be chosen. This register, XCLKOUTDIV, is mapped on the XMiscellaneous memory address range.

XCLKOUTDICV register

XCLKOUTDIV (EB02h) XBus Reset value: --00h

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

Table 136. XCLKOUTDIV register description

Bit	Bit name	Function
7-0	DIV	Clock divider setting 00h: $F_{CLKOUT} = F_{CPU}/DIV+1$

The CPU clock is output on P3.15, by default, when the CLKOUT function is enabled (setting the CLKEN bit of the SYSCON register).

By setting the XMISCEN and XPEN bits of the XPERCON and SYSCON registers respectively, the clock prescaling factor can be programmed. In this way, a prescaled value of the CPU clock can be output on P3.15.

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

23 Register set

This section summarizes the registers implemented in the ST10F296E, and explains the function and layout of the SFRs.

The registers (except the general purpose registers) are organized:

- By address, to check which register a given address references.
- By register name, to find the location of a specific register.

23.1 Register description format

Throughout the document, registers are laid out and described as follows:

Word registers

REG_NAME (A16h/A8h) SFR/ESFR/XBus Reset value: ****h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					Write only bit	HW bit	Read only bit	STD bit	HW bit	Bitfield			Bitfield		
-					W	RW	R	RW	RW	RW			RW		

Table 137. Word register description

Bit	Bit name	Function
Bit(field) number in register	bit(field) name	Explanation of bit(field) name Description of the functions controlled by the bit(field)

Byte registers

Byte registers do not contain reserved areas nor read-only/write-only bits.

REG_NAME (A16h/A8h) SFR/ESFR/XBus Reset value: --**h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	STD bit	HW bit	bitfield				bitfield	
-	-	-	-	-	-	-		RW	RW	RW				RW	

Elements

REG_NAME:	Name of the register
A16h/A8h:	Long address (16-bit)/ Short address (8-bit)
SFR/ESFR/XBus:	Register space (SFR, ESFR or XBus register)
(* *) * *	Register contents after reset
	0/1: Defined
	X: Undefined after power up)
	U: Unchanged
HW bit:	Bits that are set/cleared by hardware
STD bit:	Standard ‘normal’ bit (software rather than hardware bit)

23.2 General purpose registers (GPRs)

The GPRs form the register bank that the CPU works with. This register bank may be located anywhere within the internal RAM via the context pointer (CP). Due to the addressing mechanism, GPR banks can only reside within the internal RAM. All GPRs are bit-addressable.

Table 138. General purpose registers (GPRs)

Name	Physical address	8-bit address	Description	Reset value
R0	(CP) + 0	F0h	CPU general purpose (word) register R0	UUUUh
R1	(CP) + 2	F1h	CPU general purpose (word) register R1	UUUUh
R2	(CP) + 4	F2h	CPU general purpose (word) register R2	UUUUh
R3	(CP) + 6	F3h	CPU general purpose (word) register R3	UUUUh
R4	(CP) + 8	F4h	CPU general purpose (word) register R4	UUUUh
R5	(CP) + 10	F5h	CPU general purpose (word) register R5	UUUUh
R6	(CP) + 12	F6h	CPU general purpose (word) register R6	UUUUh
R7	(CP) + 14	F7h	CPU general purpose (word) register R7	UUUUh
R8	(CP) + 16	F8h	CPU general purpose (word) register R8	UUUUh
R9	(CP) + 18	F9h	CPU general purpose (word) register R9	UUUUh
R10	(CP) + 20	FAh	CPU general purpose (word) register R10	UUUUh
R11	(CP) + 22	FBh	CPU general purpose (word) register R11	UUUUh
R12	(CP) + 24	FCh	CPU general purpose (word) register R12	UUUUh
R13	(CP) + 26	FDh	CPU general purpose (word) register R13	UUUUh
R14	(CP) + 28	FEh	CPU general purpose (word) register R14	UUUUh
R15	(CP) + 30	FFh	CPU general purpose (word) register R15	UUUUh

The first eight GPRs (R7 to R0) may also be accessed byte wise. Writing to a GPR byte (except for SFRs) does not affect other bytes of the respective GPR. The respective halves of the byte-accessible registers receive special names listed in *Table 139*.

Table 139. General purpose registers (GPRs) bit wise addressing

Name	Physical address	8-bit address	Description	Reset value
RL0	(CP) + 0	F0h	CPU general purpose (byte) register RL0	UUh
RH0	(CP) + 1	F1h	CPU general purpose (byte) register RH0	UUh
RL1	(CP) + 2	F2h	CPU general purpose (byte) register RL1	UUh
RH1	(CP) + 3	F3h	CPU general purpose (byte) register RH1	UUh
RL2	(CP) + 4	F4h	CPU general purpose (byte) register RL2	UUh
RH2	(CP) + 5	F5h	CPU general purpose (byte) register RH2	UUh
RL3	(CP) + 6	F6h	CPU general purpose (byte) register RL3	UUh
RH3	(CP) + 7	F7h	CPU general purpose (byte) register RH3	UUh
RL4	(CP) + 8	F8h	CPU general purpose (byte) register RL4	UUh
RH4	(CP) + 9	F9h	CPU general purpose (byte) register RH4	UUh
RL5	(CP) + 10	FAh	CPU general purpose (byte) register RL5	UUh
RH5	(CP) + 11	FBh	CPU general purpose (byte) register RH5	UUh
RL6	(CP) + 12	FCh	CPU general purpose (byte) register RL6	UUh
RH6	(CP) + 13	FDh	CPU general purpose (byte) register RH6	UUh
RL7	(CP) + 14	FEh	CPU general purpose (byte) register RL7	UUh
RH7	(CP) + 15	FFh	CPU general purpose (byte) register RH7	UUh

23.3 SFRs ordered by name

Table 140 lists all SFR registers which are implemented in the ST10F296E. They are ordered by name in alphabetical order.

Bit-addressable SFRs are indicated by the bolded letter '**b**' in the 'Name' column.

SFRs within the ESFR space are indicated by the bolded letter '**E**' in the 'Physical address' column.

Table 140. SFRs ordered by name

Name	Physical address	8-bit address	Description	Reset value
ADCIC **(b)**	FF98h	CCh	ADC end of conversion interrupt control register	--00h
ADCON **(b)**	FFA0h	D0h	ADC control register	0000h
ADDAT	FEA0h	50h	ADC result register	0000h
ADDAT2	F0A0h **(E)**	50h	ADC 2 result register	0000h
ADDRSEL1	FE18h	0Ch	Address select register 1	0000h
ADDRSEL2	FE1Ah	0Dh	Address select register 2	0000h
ADDRSEL3	FE1Ch	0Eh	Address select register 3	0000h
ADDRSEL4	FE1Eh	0Fh	Address select register 4	0000h
ADEIC **(b)**	FF9Ah	CDh	ADC overrun error interrupt control register	--00h
BUSCON0 **(b)**	FF0Ch	86h	Bus configuration register 0	0xx0h
BUSCON1 **(b)**	FF14h	8Ah	Bus configuration register 1	0000h
BUSCON2 **(b)**	FF16h	8Bh	Bus configuration register 2	0000h
BUSCON3 **(b)**	FF18h	8Ch	Bus configuration register 3	0000h
BUSCON4 **(b)**	FF1Ah	8Dh	Bus configuration register 4	0000h
CAPREL	FE4Ah	25h	GPT2 capture/reload register	0000h
CC0	FE80h	40h	CAPCOM register 0	0000h
CC0IC **(b)**	FF78h	BCh	CAPCOM register 0 interrupt control register	--00h
CC1	FE82h	41h	CAPCOM register 1	0000h
CC1IC **(b)**	FF7Ah	BDh	CAPCOM register 1 interrupt control register	--00h
CC2	FE84h	42h	CAPCOM register 2	0000h
CC2IC **(b)**	FF7Ch	BEh	CAPCOM register 2 interrupt control register	--00h
CC3	FE86h	43h	CAPCOM register 3	0000h
CC3IC **(b)**	FF7Eh	BFh	CAPCOM register 3 interrupt control register	--00h
CC4	FE88h	44h	CAPCOM register 4	0000h
CC4IC **(b)**	FF80h	C0h	CAPCOM register 4 interrupt control register	--00h
CC5	FE8Ah	45h	CAPCOM register 5	0000h
CC5IC **(b)**	FF82h	C1h	CAPCOM register 5 interrupt control register	--00h
CC6	FE8Ch	46h	CAPCOM register 6	0000h

Table 140. SFRs ordered by name (continued)

Name	Physical address	8-bit address	Description	Reset value
CC6IC **(b)**	FF84h	C2h	CAPCOM register 6 interrupt control register	--00h
CC7	FE8Eh	47h	CAPCOM register 7	0000h
CC7IC **(b)**	FF86h	C3h	CAPCOM register 7 interrupt control register	--00h
CC8	FE90h	48h	CAPCOM register 8	0000h
CC8IC **(b)**	FF88h	C4h	CAPCOM register 8 interrupt control register	--00h
CC9	FE92h	49h	CAPCOM register 9	0000h
CC9IC **(b)**	FF8Ah	C5h	CAPCOM register 9 interrupt control register	--00h
CC10	FE94h	4Ah	CAPCOM register 10	0000h
CC10IC **(b)**	FF8Ch	C6h	CAPCOM register 10 interrupt control register	--00h
CC11	FE96h	4Bh	CAPCOM register 11	0000h
CC11IC **(b)**	FF8Eh	C7h	CAPCOM register 11 interrupt control register	--00h
CC12	FE98h	4Ch	CAPCOM register 12	0000h
CC12IC **(b)**	FF90h	C8h	CAPCOM register 12 interrupt control register	--00h
CC13	FE9Ah	4Dh	CAPCOM register 13	0000h
CC13IC **(b)**	FF92h	C9h	CAPCOM register 13 interrupt control register	--00h
CC14	FE9Ch	4Eh	CAPCOM register 14	0000h
CC14IC **(b)**	FF94h	CAh	CAPCOM register 14 interrupt control register	--00h
CC15	FE9Eh	4Fh	CAPCOM register 15	0000h
CC15IC **(b)**	FF96h	CBh	CAPCOM register 15 interrupt control register	--00h
CC16	FE60h	30h	CAPCOM register 16	0000h
CC16IC **(b)**	F160h **(E)**	B0h	CAPCOM register 16 interrupt control register	--00h
CC17	FE62h	31h	CAPCOM register 17	0000h
CC17IC **(b)**	F162h **(E)**	B1h	CAPCOM register 17 interrupt control register	--00h
CC18	FE64h	32h	CAPCOM register 18	0000h
CC18IC **(b)**	F164h **(E)**	B2h	CAPCOM register 18 interrupt control register	--00h
CC19	FE66h	33h	CAPCOM register 19	0000h
CC19IC **(b)**	F166h **(E)**	B3h	CAPCOM register 19 interrupt control register	--00h
CC20	FE68h	34h	CAPCOM register 20	0000h
CC20IC **(b)**	F168h **(E)**	B4h	CAPCOM register 20 interrupt control register	--00h
CC21	FE6Ah	35h	CAPCOM register 21	0000h
CC21IC **(b)**	F16Ah **(E)**	B5h	CAPCOM register 21 interrupt control register	--00h
CC22	FE6Ch	36h	CAPCOM register 22	0000h
CC22IC **(b)**	F16Ch **(E)**	B6h	CAPCOM register 22 interrupt control register	--00h
CC23	FE6Eh	37h	CAPCOM register 23	0000h

Table 140. SFRs ordered by name (continued)

Name	Physical address	8-bit address	Description	Reset value
CC23IC **(b)**	F16Eh **(E)**	B7h	CAPCOM register 23 interrupt control register	--00h
CC24	FE70h	38h	CAPCOM register 24	0000h
CC24IC **(b)**	F170h **(E)**	B8h	CAPCOM register 24 interrupt control register	--00h
CC25	FE72h	39h	CAPCOM register 25	0000h
CC25IC **(b)**	F172h **(E)**	B9h	CAPCOM register 25 interrupt control register	--00h
CC26	FE74h	3Ah	CAPCOM register 26	0000h
CC26IC **(b)**	F174h **(E)**	BAh	CAPCOM register 26 interrupt control register	--00h
CC27	FE76h	3Bh	CAPCOM register 27	0000h
CC27IC **(b)**	F176h **(E)**	BBh	CAPCOM register 27 interrupt control register	--00h
CC28	FE78h	3Ch	CAPCOM register 28	0000h
CC28IC **(b)**	F178h **(E)**	BCh	CAPCOM register 28 interrupt control register	--00h
CC29	FE7Ah	3Dh	CAPCOM register 29	0000h
CC29IC **(b)**	F184h **(E)**	C2h	CAPCOM register 29 interrupt control register	--00h
CC30	FE7Ch	3Eh	CAPCOM register 30	0000h
CC30IC **(b)**	F18Ch **(E)**	C6h	CAPCOM register 30 interrupt control register	--00h
CC31	FE7Eh	3Fh	CAPCOM register 31	0000h
CC31IC **(b)**	F194h **(E)**	CAh	CAPCOM register 31 interrupt control register	--00h
CCM0 **(b)**	FF52h	A9h	CAPCOM mode control register 0	0000h
CCM1 **(b)**	FF54h	AAh	CAPCOM mode control register 1	0000h
CCM2 **(b)**	FF56h	ABh	CAPCOM mode control register 2	0000h
CCM3 **(b)**	FF58h	ACh	CAPCOM mode control register 3	0000h
CCM4 **(b)**	FF22h	91h	CAPCOM mode control register 4	0000h
CCM5 **(b)**	FF24h	92h	CAPCOM mode control register 5	0000h
CCM6 **(b)**	FF26h	93h	CAPCOM mode control register 6	0000h
CCM7 **(b)**	FF28h	94h	CAPCOM mode control register 7	0000h
CP	FE10h	08h	CPU context pointer register	FC00h
CRIC **(b)**	FF6Ah	B5h	GPT2 CAPREL interrupt control register	--00h
CSP	FE08h	04h	CPU code segment pointer register (read-only)	0000h
DP0L **(b)**	F100h **(E)**	80h	P0L direction control register	--00h
DP0H **(b)**	F102h **(E)**	81h	P0h direction control register	--00h
DP1L **(b)**	F104h **(E)**	82h	P1L direction control register	--00h
DP1H **(b)**	F106h **(E)**	83h	P1h direction control register	--00h
DP2 **(b)**	FFC2h	E1h	Port 2 direction control register	0000h
DP3 **(b)**	FFC6h	E3h	Port 3 direction control register	0000h

Table 140. SFRs ordered by name (continued)

Name	Physical address	8-bit address	Description	Reset value
DP4 **(b)**	FFCAh	E5h	Port 4 direction control register	--00h
DP6 **(b)**	FFCEh	E7h	Port 6 direction control register	--00h
DP7 **(b)**	FFD2h	E9h	Port 7 direction control register	--00h
DP8 **(b)**	FFD6h	EBh	Port 8 direction control register	--00h
DPP0	FE00h	00h	CPU data page pointer 0 register (10-bit)	0000h
DPP1	FE02h	01h	CPU data page pointer 1 register (10-bit)	0001h
DPP2	FE04h	02h	CPU data page pointer 2 register (10-bit)	0002h
DPP3	FE06h	03h	CPU data page pointer 3 register (10-bit)	0003h
EMUCON	FE0Ah	05h	Emulation control register	--XXh
EXICON **(b)**	F1C0h **(E)**	E0h	External interrupt control register	0000h
EXISEL **(b)**	F1DAh **(E)**	EDh	External interrupt source selection register	0000h
IDCHIP	F07Ch **(E)**	3Eh	Device identifier register (n is the device revision)	128nh
IDMANUF	F07Eh **(E)**	3Fh	Manufacturer identifier register	0403h
IDMEM	F07Ah **(E)**	3Dh	On-chip memory identifier register	30D0h
IDPROG	F078h **(E)**	3Ch	Programming voltage identifier register	0040h
IDX0 **(b)**	FF08h	84h	MAC unit address pointer 0	0000h
IDX1 **(b)**	FF0Ah	85h	MAC unit address pointer 1	0000h
MAH	FE5Eh	2Fh	MAC unit accumulator - high word	0000h
MAL	FE5Ch	2Eh	MAC unit accumulator - low word	0000h
MCW **(b)**	FFDCh	EEh	MAC unit control word	0000h
MDC **(b)**	FF0Eh	87h	CPU multiply divide control register	0000h
MDH	FE0Ch	06h	CPU multiply divide register – high word	0000h
MDL	FE0Eh	07h	CPU multiply divide register – low word	0000h
MRW **(b)**	FFDAh	EDh	MAC unit repeat word	0000h
MSW **(b)**	FFDEh	EFh	MAC unit status word	0200h
ODP2 **(b)**	F1C2h **(E)**	E1h	Port 2 open-drain control register	0000h
ODP3 **(b)**	F1C6h **(E)**	E3h	Port 3 open-drain control register	0000h
ODP4 **(b)**	F1CAh **(E)**	E5h	Port 4 open-drain control register	--00h
ODP6 **(b)**	F1CEh **(E)**	E7h	Port 6 open-drain control register	--00h
ODP7 **(b)**	F1D2h **(E)**	E9h	Port 7 open-drain control register	--00h
ODP8 **(b)**	F1D6h **(E)**	EBh	Port 8 open-drain control register	--00h
ONES **(b)**	FF1Eh	8Fh	Constant value 1's register (read-only)	FFFFh
P0L **(b)**	FF00h	80h	Port 0 low register (lower half of Port 0)	--00h
P0H **(b)**	FF02h	81h	Port 0 high register (upper half of Port 0)	--00h

Table 140. SFRs ordered by name (continued)

Name	Physical address	8-bit address	Description	Reset value
P1L **(b)**	FF04h	82h	Port 1 low register (lower half of Port 1)	--00h
P1H **(b)**	FF06h	83h	Port 1 high register (upper half of Port 1)	--00h
P2 **(b)**	FFC0h	E0h	Port 2 register	0000h
P3 **(b)**	FFC4h	E2h	Port 3 register	0000h
P4 **(b)**	FFC8h	E4h	Port 4 register (8-bit)	--00h
P5 **(b)**	FFA2h	D1h	Port 5 register (read-only)	XXXXh
P6 **(b)**	FFCCh	E6h	Port 6 register (8-bit)	--00h
P7 **(b)**	FFD0h	E8h	Port 7 register (8-bit)	--00h
P8 **(b)**	FFD4h	EAh	Port 8 register (8-bit)	--00h
P5DIDIS **(b)**	FFA4h	D2h	Port 5 digital disable register	0000h
PECC0	FEC0h	60h	PEC channel 0 control register	0000h
PECC1	FEC2h	61h	PEC channel 1 control register	0000h
PECC2	FEC4h	62h	PEC channel 2 control register	0000h
PECC3	FEC6h	63h	PEC channel 3 control register	0000h
PECC4	FEC8h	64h	PEC channel 4 control register	0000h
PECC5	FECAh	65h	PEC channel 5 control register	0000h
PECC6	FECCh	66h	PEC channel 6 control register	0000h
PECC7	FECEh	67h	PEC channel 7 control register	0000h
PICON **(b)**	F1C4h **(E)**	E2h	Port input threshold control register	--00h
PP0	F038h **(E)**	1Ch	PWM module period register 0	0000h
PP1	F03Ah **(E)**	1Dh	PWM module period register 1	0000h
PP2	F03Ch **(E)**	1Eh	PWM module period register 2	0000h
PP3	F03Eh **(E)**	1Fh	PWM module period register 3	0000h
PSW **(b)**	FF10h	88h	CPU program status word	0000h
PT0	F030h **(E)**	18h	PWM module up/down counter 0	0000h
PT1	F032h **(E)**	19h	PWM module up/down counter 1	0000h
PT2	F034h **(E)**	1Ah	PWM module up/down counter 2	0000h
PT3	F036h **(E)**	1Bh	PWM module up/down counter 3	0000h
PW0	FE30h	18h	PWM module pulse width register 0	0000h
PW1	FE32h	19h	PWM module pulse width register 1	0000h
PW2	FE34h	1Ah	PWM module pulse width register 2	0000h
PW3	FE36h	1Bh	PWM module pulse width register 3	0000h
PWMCON0 **(b)**	FF30h	98h	PWM module control register 0	0000h
PWMCON1 **(b)**	FF32h	99h	PWM module control register 1	0000h

Table 140. SFRs ordered by name (continued)

Name	Physical address	8-bit address	Description	Reset value
PWMIC **(b)**	F17Eh **(E)**	BFh	PWM module interrupt control register	--00h
QR0	F004h **(E)**	02h	MAC unit offset register R0	0000h
QR1	F006h **(E)**	03h	MAC unit offset register r1	0000h
QX0	F000h **(E)**	00h	MAC unit offset register x0	0000h
QX1	F002h **(E)**	01h	MAC unit offset register x1	0000h
RP0H **(b)**	F108h **(E)**	84h	System start-up configuration register (read-only)	--XXh
S0BG	FEB4h	5Ah	Serial channel 0 baud rate generator reload register	0000h
S0CON **(b)**	FFB0h	D8h	Serial channel 0 control register	0000h
S0EIC **(b)**	FF70h	B8h	Serial channel 0 error interrupt control register	--00h
S0RBUF	FEB2h	59h	Serial channel 0 receive buffer register (read-only)	--XXh
S0RIC **(b)**	FF6Eh	B7h	Serial channel 0 receive interrupt control register	--00h
S0TBIC **(b)**	F19Ch **(E)**	CEh	Serial channel 0 transmit buffer interrupt control register	--00h
S0TBUF	FEB0h	58h	Serial channel 0 transmit buffer register (write-only)	0000h
S0TIC **(b)**	FF6Ch	B6h	Serial channel 0 transmit interrupt control register	--00h
SP	FE12h	09h	CPU system stack pointer register	FC00h
SSCBR	F0B4h **(E)**	5Ah	SSC baud rate register	0000h
SSCCON **(b)**	FFB2h	D9h	SSC control register	0000h
SSCEIC **(b)**	FF76h	BBh	SSC error interrupt control register	--00h
SSCRB	F0B2h **(E)**	59h	SSC receive buffer (read-only)	XXXXh
SSCRIC **(b)**	FF74h	BAh	SSC receive interrupt control register	--00h
SSCTB	F0B0h **(E)**	58h	SSC transmit buffer (write-only)	0000h
SSCTIC **(b)**	FF72h	B9h	SSC transmit interrupt control register	--00h
STKOV	FE14h	0Ah	CPU stack overflow pointer register	FA00h
STKUN	FE16h	0Bh	CPU stack underflow pointer register	FC00h
SYSCON **(b)**	FF12h	89h	CPU system configuration register	0xx0h(1)
T0	FE50h	28h	CAPCOM timer 0 register	0000h
T01CON **(b)**	FF50h	A8h	CAPCOM timer 0 and timer 1 control register	0000h
T0IC **(b)**	FF9Ch	CEh	CAPCOM timer 0 interrupt control register	--00h
T0REL	FE54h	2Ah	CAPCOM timer 0 reload register	0000h
T1	FE52h	29h	CAPCOM timer 1 register	0000h
T1IC **(b)**	FF9Eh	CFh	CAPCOM timer 1 interrupt control register	--00h
T1REL	FE56h	2Bh	CAPCOM timer 1 reload register	0000h
T2	FE40h	20h	GPT1 timer 2 register	0000h
T2CON **(b)**	FF40h	A0h	GPT1 timer 2 control register	0000h

Table 140. SFRs ordered by name (continued)

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

1. System configuration is selected during reset. The SYSCON reset value is 0000 0xx0 x000 0000b.
2. The reset value depends on different triggered reset events.
3. The XPnIC interrupt control register control interrupt requests from the integrated XBus peripherals. Some software controlled interrupt requests may be generated by setting the XPnIR bits (of the XPnIC register) of the unused XPeripheral nodes.

23.4 SFRs ordered by address

Table 141 lists all SFR registers which are implemented in the ST10F296E, ordered by their physical address.

Bit-addressable SFRs are indicated by the bolded letter '**b**' in the 'Name' column.

SFRs within the ESFR space are indicated by the bolded letter '**E**' in the 'Physical address' column.

Table 141. SFRs ordered by address

Name	Physical address	8-bit address	Description	Reset value
QX0	F000h **(E)**	00h	MAC unit offset register X0	0000h
QX1	F002h **(E)**	01h	MAC unit offset register X1	0000h
QR0	F004h **(E)**	02h	MAC unit offset register R0	0000h
QR1	F006h **(E)**	03h	MAC unit offset register R1	0000h
XADRS3	F01Ch **(E)**	0Eh	XPER address select register 3	800Bh
XPERCON	F024h **(E)**	12h	XPER configuration register	--05h
PT0	F030h **(E)**	18h	PWM module up/down counter 0	0000h
PT1	F032h **(E)**	19h	PWM module up/down counter 1	0000h
PT2	F034h **(E)**	1Ah	PWM module up/down counter 2	0000h
PT3	F036h **(E)**	1Bh	PWM module up/down counter 3	0000h
PP0	F038h **(E)**	1Ch	PWM module period register 0	0000h
PP1	F03Ah **(E)**	1Dh	PWM module period register 1	0000h
PP2	F03Ch **(E)**	1Eh	PWM module period register 2	0000h
PP3	F03Eh **(E)**	1Fh	PWM module period register 3	0000h
T7	F050h **(E)**	28h	CAPCOM timer 7 register	0000h
T8	F052h **(E)**	29h	CAPCOM timer 8 register	0000h
T7REL	F054h **(E)**	2Ah	CAPCOM timer 7 reload register	0000h
T8REL	F056h **(E)**	2Bh	CAPCOM timer 8 reload register	0000h
IDPROG	F078h **(E)**	3Ch	Programming voltage identifier register	0040h
IDMEM	F07Ah **(E)**	3Dh	On-chip memory identifier register	30D0h
IDCHIP	F07Ch **(E)**	3Eh	Device identifier register (n is the device revision)	128nh
IDMANUF	F07Eh **(E)**	3Fh	Manufacturer identifier register	0403h
ADDAT2	F0A0h **(E)**	50h	ADC 2 result register	0000h
SSCTB	F0B0h **(E)**	58h	SSC transmit buffer (write-only)	0000h
SSCRB	F0B2h **(E)**	59h	SSC receive buffer (read-only)	XXXXh
SSCBR	F0B4h **(E)**	5Ah	SSC baud rate register	0000h
DP0L **(b)**	F100h **(E)**	80h	P0L direction control register	--00h
DP0H **(b)**	F102h **(E)**	81h	P0H direction control register	--00h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
DP1L **(b)**	F104h **(E)**	82h	P1L direction control register	--00h
DP1H **(b)**	F106h **(E)**	83h	P1H direction control register	--00h
RP0H **(b)**	F108h **(E)**	84h	System startup configuration register (read-only)	--XXh
CC16IC **(b)**	F160h **(E)**	B0h	CAPCOM register 16 interrupt control register	--00h
CC17IC **(b)**	F162h **(E)**	B1h	CAPCOM register 17 interrupt control register	--00h
CC18IC **(b)**	F164h **(E)**	B2h	CAPCOM register 18 interrupt control register	--00h
CC19IC **(b)**	F166h **(E)**	B3h	CAPCOM register 19 interrupt control register	--00h
CC20IC **(b)**	F168h **(E)**	B4h	CAPCOM register 20 interrupt control register	--00h
CC21IC **(b)**	F16Ah **(E)**	B5h	CAPCOM register 21 interrupt control register	--00h
CC22IC **(b)**	F16Ch **(E)**	B6h	CAPCOM register 22 interrupt control register	--00h
CC23IC **(b)**	F16Eh **(E)**	B7h	CAPCOM register 23 interrupt control register	--00h
CC24IC **(b)**	F170h **(E)**	B8h	CAPCOM register 24 interrupt control register	--00h
CC25IC **(b)**	F172h **(E)**	B9h	CAPCOM register 25 interrupt control register	--00h
CC26IC **(b)**	F174h **(E)**	BAh	CAPCOM register 26 interrupt control register	--00h
CC27IC **(b)**	F176h **(E)**	BBh	CAPCOM register 27 interrupt control register	--00h
CC28IC **(b)**	F178h **(E)**	BCh	CAPCOM register 28 interrupt control register	--00h
T7IC **(b)**	F17Ah **(E)**	BDh	CAPCOM timer 7 interrupt control register	--00h
T8IC **(b)**	F17Ch **(E)**	BEh	CAPCOM timer 8 interrupt control register	--00h
PWMIC **(b)**	F17Eh **(E)**	BFh	PWM module interrupt control register	--00h
CC29IC **(b)**	F184h **(E)**	C2h	CAPCOM register 29 interrupt control register	--00h
XP0IC **(b)**	F186h **(E)**	C3h	See *Section 9.1: XPeripheral interrupt*	--00h
CC30IC **(b)**	F18Ch **(E)**	C6h	CAPCOM register 30 interrupt control register	--00h
XP1IC **(b)**	F18Eh **(E)**	C7h	See *Section 9.1: XPeripheral interrupt*	--00h
CC31IC **(b)**	F194h **(E)**	CAh	CAPCOM register 31 interrupt control register	--00h
XP2IC **(b)**	F196h **(E)**	CBh	See *Section 9.1: XPeripheral interrupt*	--00h
S0TBIC **(b)**	F19Ch **(E)**	CEh	Serial channel 0 transmit buffer interrupt control register	--00h
XP3IC **(b)**	F19Eh **(E)**	CFh	See *Section 9.1: XPeripheral interrupt*	--00h
EXICON **(b)**	F1C0h **(E)**	E0h	External interrupt control register	0000h
ODP2 **(b)**	F1C2h **(E)**	E1h	Port 2 open-drain control register	0000h
PICON **(b)**	F1C4h **(E)**	E2h	Port input threshold control register	--00h
ODP3 **(b)**	F1C6h **(E)**	E3h	Port 3 open-drain control register	0000h
ODP4 **(b)**	F1CAh **(E)**	E5h	Port 4 open-drain control register	--00h
ODP6 **(b)**	F1CEh **(E)**	E7h	Port 6 open-drain control register	--00h
ODP7 **(b)**	F1D2h **(E)**	E9h	Port 7 open-drain control register	--00h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
ODP8 **(b)**	F1D6h **(E)**	EBh	Port 8 open-drain control register	--00h
EXISEL **(b)**	F1DAh **(E)**	EDh	External interrupt source selection register	0000h
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
CSP	FE08h	04h	CPU code segment pointer register (read-only)	0000h
EMUCON	FE0Ah	05h	Emulation control register	--XXh
MDH	FE0Ch	06h	CPU multiply divide register – high word	0000h
MDL	FE0Eh	07h	CPU multiply divide register – low word	0000h
CP	FE10h	08h	CPU context pointer register	FC00h
SP	FE12h	09h	CPU system stack pointer register	FC00h
STKOV	FE14h	0Ah	CPU stack overflow pointer register	FA00h
STKUN	FE16h	0Bh	CPU stack underflow pointer register	FC00h
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
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
T2	FE40h	20h	GPT1 timer 2 register	0000h
T3	FE42h	21h	GPT1 timer 3 register	0000h
T4	FE44h	22h	GPT1 timer 4 register	0000h
T5	FE46h	23h	GPT2 timer 5 register	0000h
T6	FE48h	24h	GPT2 timer 6 register	0000h
CAPREL	FE4Ah	25h	GPT2 capture/reload register	0000h
T0	FE50h	28h	CAPCOM timer 0 register	0000h
T1	FE52h	29h	CAPCOM timer 1 register	0000h
T0REL	FE54h	2Ah	CAPCOM timer 0 reload register	0000h
T1REL	FE56h	2Bh	CAPCOM timer 1 reload register	0000h
MAL	FE5Ch	2Eh	MAC unit accumulator - low word	0000h
MAH	FE5Eh	2Fh	MAC unit accumulator - high word	0000h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
CC16	FE60h	30h	CAPCOM register 16	0000h
CC17	FE62h	31h	CAPCOM register 17	0000h
CC18	FE64h	32h	CAPCOM register 18	0000h
CC19	FE66h	33h	CAPCOM register 19	0000h
CC20	FE68h	34h	CAPCOM register 20	0000h
CC21	FE6Ah	35h	CAPCOM register 21	0000h
CC22	FE6Ch	36h	CAPCOM register 22	0000h
CC23	FE6Eh	37h	CAPCOM register 23	0000h
CC24	FE70h	38h	CAPCOM register 24	0000h
CC25	FE72h	39h	CAPCOM register 25	0000h
CC26	FE74h	3Ah	CAPCOM register 26	0000h
CC27	FE76h	3Bh	CAPCOM register 27	0000h
CC28	FE78h	3Ch	CAPCOM register 28	0000h
CC29	FE7Ah	3Dh	CAPCOM register 29	0000h
CC30	FE7Ch	3Eh	CAPCOM register 30	0000h
CC31	FE7Eh	3Fh	CAPCOM register 31	0000h
CC0	FE80h	40h	CAPCOM register 0	0000h
CC1	FE82h	41h	CAPCOM register 1	0000h
CC2	FE84h	42h	CAPCOM register 2	0000h
CC3	FE86h	43h	CAPCOM register 3	0000h
CC4	FE88h	44h	CAPCOM register 4	0000h
CC5	FE8Ah	45h	CAPCOM register 5	0000h
CC6	FE8Ch	46h	CAPCOM register 6	0000h
CC7	FE8Eh	47h	CAPCOM register 7	0000h
CC8	FE90h	48h	CAPCOM register 8	0000h
CC9	FE92h	49h	CAPCOM register 9	0000h
CC10	FE94h	4Ah	CAPCOM register 10	0000h
CC11	FE96h	4Bh	CAPCOM register 11	0000h
CC12	FE98h	4Ch	CAPCOM register 12	0000h
CC13	FE9Ah	4Dh	CAPCOM register 13	0000h
CC14	FE9Ch	4Eh	CAPCOM register 14	0000h
CC15	FE9Eh	4Fh	CAPCOM register 15	0000h
ADDAT	FEA0h	50h	ADC result register	0000h
WDT	FEAEh	57h	Watchdog timer register (read-only)	0000h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
S0TBUF	FEB0h	58h	Serial channel 0 transmit buffer register (write-only)	0000h
S0RBUF	FEB2h	59h	Serial channel 0 receive buffer register (read-only)	--XXh
S0BG	FEB4h	5Ah	Serial channel 0 baud rate generator reload register	0000h
PECC0	FEC0h	60h	PEC channel 0 control register	0000h
PECC1	FEC2h	61h	PEC channel 1 control register	0000h
PECC2	FEC4h	62h	PEC channel 2 control register	0000h
PECC3	FEC6h	63h	PEC channel 3 control register	0000h
PECC4	FEC8h	64h	PEC channel 4 control register	0000h
PECC5	FECAh	65h	PEC channel 5 control register	0000h
PECC6	FECCh	66h	PEC channel 6 control register	0000h
PECC7	FECEh	67h	PEC channel 7 control register	0000h
P0L **(b)**	FF00h	80h	Port 0 low register (lower half of Port 0)	--00h
P0H **(b)**	FF02h	81h	Port 0 high register (upper half of Port 0)	--00h
P1L **(b)**	FF04h	82h	Port 1 low register (lower half of Port 1)	--00h
P1H **(b)**	FF06h	83h	Port 1 high register (upper half of Port 1)	--00h
IDX0 **(b)**	FF08h	84h	MAC unit address pointer 0	0000h
IDX1 **(b)**	FF0Ah	85h	MAC unit address pointer 1	0000h
BUSCON0 **(b)**	FF0Ch	86h	Bus configuration register 0	0xx0h
MDC **(b)**	FF0Eh	87h	CPU multiply divide control register	0000h
PSW **(b)**	FF10h	88h	CPU program status word	0000h
SYSCON **(b)**	FF12h	89h	CPU system configuration register	0xx0h
BUSCON1 **(b)**	FF14h	8Ah	Bus configuration register 1	0000h
BUSCON2 **(b)**	FF16h	8Bh	Bus configuration register 2	0000h
BUSCON3 **(b)**	FF18h	8Ch	Bus configuration register 3	0000h
BUSCON4 **(b)**	FF1Ah	8Dh	Bus configuration register 4	0000h
ZEROS **(b)**	FF1Ch	8Eh	Constant value 0's register (read-only)	0000h
ONES **(b)**	FF1Eh	8Fh	Constant value 1's register (read-only)	FFFFh
T78CON **(b)**	FF20h	90h	CAPCOM timer 7 and 8 control register	0000h
CCM4 **(b)**	FF22h	91h	CAPCOM mode control register 4	0000h
CCM5 **(b)**	FF24h	92h	CAPCOM mode control register 5	0000h
CCM6 **(b)**	FF26h	93h	CAPCOM mode control register 6	0000h
CCM7 **(b)**	FF28h	94h	CAPCOM mode control register 7	0000h
PWMCON0 **(b)**	FF30h	98h	PWM module control register 0	0000h
PWMCON1 **(b)**	FF32h	99h	PWM module control register 1	0000h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
T2CON **(b)**	FF40h	A0h	GPT1 timer 2 control register	0000h
T3CON **(b)**	FF42h	A1h	GPT1 timer 3 control register	0000h
T4CON **(b)**	FF44h	A2h	GPT1 timer 4 control register	0000h
T5CON **(b)**	FF46h	A3h	GPT2 timer 5 control register	0000h
T6CON **(b)**	FF48h	A4h	GPT2 timer 6 control register	0000h
T01CON **(b)**	FF50h	A8h	CAPCOM timer 0 and timer 1 control register	0000h
CCM0 **(b)**	FF52h	A9h	CAPCOM mode control register 0	0000h
CCM1 **(b)**	FF54h	AAh	CAPCOM mode control register 1	0000h
CCM2 **(b)**	FF56h	ABh	CAPCOM mode control register 2	0000h
CCM3 **(b)**	FF58h	ACh	CAPCOM mode control register 3	0000h
T2IC **(b)**	FF60h	B0h	GPT1 timer 2 interrupt control register	--00h
T3IC **(b)**	FF62h	B1h	GPT1 timer 3 interrupt control register	--00h
T4IC **(b)**	FF64h	B2h	GPT1 timer 4 interrupt control register	--00h
T5IC **(b)**	FF66h	B3h	GPT2 timer 5 interrupt control register	--00h
T6IC **(b)**	FF68h	B4h	GPT2 timer 6 interrupt control register	--00h
CRIC **(b)**	FF6Ah	B5h	GPT2 CAPREL interrupt control register	--00h
S0TIC **(b)**	FF6Ch	B6h	Serial channel 0 transmit interrupt control register	--00h
S0RIC **(b)**	FF6Eh	B7h	Serial channel 0 receive interrupt control register	--00h
S0EIC **(b)**	FF70h	B8h	Serial channel 0 error interrupt control register	--00h
SSCTIC **(b)**	FF72h	B9h	SSC transmit interrupt control register	--00h
SSCRIC **(b)**	FF74h	BAh	SSC receive interrupt control register	--00h
SSCEIC **(b)**	FF76h	BBh	SSC error interrupt control register	--00h
CC0IC **(b)**	FF78h	BCh	CAPCOM register 0 interrupt control register	--00h
CC1IC **(b)**	FF7Ah	BDh	CAPCOM register 1 interrupt control register	--00h
CC2IC **(b)**	FF7Ch	BEh	CAPCOM register 2 interrupt control register	--00h
CC3IC **(b)**	FF7Eh	BFh	CAPCOM register 3 interrupt control register	--00h
CC4IC **(b)**	FF80h	C0h	CAPCOM register 4 interrupt control register	--00h
CC5IC **(b)**	FF82h	C1h	CAPCOM register 5 interrupt control register	--00h
CC6IC **(b)**	FF84h	C2h	CAPCOM register 6 interrupt control register	--00h
CC7IC **(b)**	FF86h	C3h	CAPCOM register 7 interrupt control register	--00h
CC8IC **(b)**	FF88h	C4h	CAPCOM register 8 interrupt control register	--00h
CC9IC **(b)**	FF8Ah	C5h	CAPCOM register 9 interrupt control register	--00h
CC10IC **(b)**	FF8Ch	C6h	CAPCOM register 10 interrupt control register	--00h
CC11IC **(b)**	FF8Eh	C7h	CAPCOM register 11 interrupt control register	--00h

Table 141. SFRs ordered by address (continued)

Name	Physical address	8-bit address	Description	Reset value
CC12IC **(b)**	FF90h	C8h	CAPCOM register 12 interrupt control register	--00h
CC13IC **(b)**	FF92h	C9h	CAPCOM register 13 interrupt control register	--00h
CC14IC **(b)**	FF94h	CAh	CAPCOM register 14 interrupt control register	--00h
CC15IC **(b)**	FF96h	CBh	CAPCOM register 15 interrupt control register	--00h
ADCIC **(b)**	FF98h	CCh	ADC end of conversion interrupt control register	--00h
ADEIC **(b)**	FF9Ah	CDh	ADC overrun error interrupt control register	--00h
T0IC **(b)**	FF9Ch	CEh	CAPCOM timer 0 interrupt control register	--00h
T1IC **(b)**	FF9Eh	CFh	CAPCOM timer 1 interrupt control register	--00h
ADCON **(b)**	FFA0h	D0h	ADC control register	0000h
P5 **(b)**	FFA2h	D1h	Port 5 register (read-only)	XXXXh
P5DIDIS **(b)**	FFA4h	D2h	Port 5 digital disable register	0000h
TFR **(b)**	FFACh	D6h	Trap flag register	0000h
WDTCON **(b)**	FFAEh	D7h	Watchdog timer control register	00xxh
S0CON **(b)**	FFB0h	D8h	Serial channel 0 control register	0000h
SSCCON **(b)**	FFB2h	D9h	SSC control register	0000h
P2 **(b)**	FFC0h	E0h	Port 2 register	0000h
DP2 **(b)**	FFC2h	E1h	Port 2 direction control register	0000h
P3 **(b)**	FFC4h	E2h	Port 3 register	0000h
DP3 **(b)**	FFC6h	E3h	Port 3 direction control register	0000h
P4 **(b)**	FFC8h	E4h	Port 4 register (8-bit)	--00h
DP4 **(b)**	FFCAh	E5h	Port 4 direction control register	--00h
P6 **(b)**	FFCCh	E6h	Port 6 register (8-bit)	--00h
DP6 **(b)**	FFCEh	E7h	Port 6 direction control register	--00h
P7 **(b)**	FFD0h	E8h	Port 7 register (8-bit)	--00h
DP7 **(b)**	FFD2h	E9h	Port 7 direction control register	--00h
P8 **(b)**	FFD4h	EAh	Port 8 register (8-bit)	--00h
DP8 **(b)**	FFD6h	EBh	Port 8 direction control register	--00h
MRW **(b)**	FFDAh	EDh	MAC unit repeat word	0000h
MCW **(b)**	FFDCh	EEh	MAC unit control word	0000h
MSW **(b)**	FFDEh	EFh	MAC unit status word	0200h

23.5 X registers ordered by name

Table 142 lists all XBus registers which are implemented in the ST10F296E ordered by their name. The Flash control registers are physically mapped on the XBus memory space, but, are listed in *Section 23.7: Flash registers ordered by name*. The X registers are not bit-addressable.

Table 142. X registers ordered by name

Name	Physical address	Description	Reset value
CAN1BRPER	EF0Ch	CAN1 BRP extension register	0000h
CAN1BTR	EF06h	CAN1 bit timing register	2301h
CAN1CR	EF00h	CAN1 CAN control register	0001h
CAN1EC	EF04h	CAN1 error counter	0000h
CAN1IF1A1	EF18h	CAN1 IF1 arbitration 1	0000h
CAN1IF1A2	EF1Ah	CAN1 IF1 arbitration 2	0000h
CAN1IF1CM	EF12h	CAN1 IF1 command mask	0000h
CAN1IF1CR	EF10h	CAN1 IF1 command request	0001h
CAN1IF1DA1	EF1Eh	CAN1 IF1 data A 1	0000h
CAN1IF1DA2	EF20h	CAN1 IF1 data A 2	0000h
CAN1IF1DB1	EF22h	CAN1 IF1 data B 1	0000h
CAN1IF1DB2	EF24h	CAN1 IF1 data B 2	0000h
CAN1IF1M1	EF14h	CAN1 IF1 mask 1	FFFFh
CAN1IF1M2	EF16h	CAN1 IF1 mask 2	FFFFh
CAN1IF1MC	EF1Ch	CAN1 IF1 message control	0000h
CAN1IF2A1	EF48h	CAN1 IF2 arbitration 1	0000h
CAN1IF2A2	EF4Ah	CAN1 IF2 arbitration 2	0000h
CAN1IF2CM	EF42h	CAN1 IF2 command mask	0000h
CAN1IF2CR	EF40h	CAN1 IF2 command request	0001h
CAN1IF2DA1	EF4Eh	CAN1 IF2 data A 1	0000h
CAN1IF2DA2	EF50h	CAN1 IF2 data A 2	0000h
CAN1IF2DB1	EF52h	CAN1 IF2 data B 1	0000h
CAN1IF2DB2	EF54h	CAN1 IF2 data B 2	0000h
CAN1IF2M1	EF44h	CAN1 IF2 mask 1	FFFFh
CAN1IF2M2	EF46h	CAN1 IF2 mask 2	FFFFh
CAN1IF2MC	EF4Ch	CAN1 IF2 message control	0000h
CAN1IP1	EFA0h	CAN1 interrupt pending 1	0000h
CAN1IP2	EFA2h	CAN1 interrupt pending 2	0000h
CAN1IR	EF08h	CAN1 interrupt register	0000h

Table 142. X registers ordered by name (continued)

Name	Physical address	Description	Reset value
CAN1MV1	EFB0h	CAN1 message valid 1	0000h
CAN1MV2	EFB2h	CAN1 message valid 2	0000h
CAN1ND1	EF90h	CAN1 new data 1	0000h
CAN1ND2	EF92h	CAN1 new data 2	0000h
CAN1SR	EF02h	CAN1 status register	0000h
CAN1TR	EF0Ah	CAN1 test register	00x0h
CAN1TR1	EF80h	CAN1 transmission request 1	0000h
CAN1TR2	EF82h	CAN1 transmission request 2	0000h
CAN2BRPER	EE0Ch	CAN2 BRP extension register	0000h
CAN2BTR	EE06h	CAN2 bit timing register	2301h
CAN2CR	EE00h	CAN2 CAN control register	0001h
CAN2EC	EE04h	CAN2 error counter	0000h
CAN2IF1A1	EE18h	CAN2 IF1 arbitration 1	0000h
CAN2IF1A2	EE1Ah	CAN2 IF1 arbitration 2	0000h
CAN2IF1CM	EE12h	CAN2 IF1 command mask	0000h
CAN2IF1CR	EE10h	CAN2 IF1 command request	0001h
CAN2IF1DA1	EE1Eh	CAN2 IF1 data A 1	0000h
CAN2IF1DA2	EE20h	CAN2 IF1 data A 2	0000h
CAN2IF1DB1	EE22h	CAN2 IF1 data B 1	0000h
CAN2IF1DB2	EE24h	CAN2 IF1 data B 2	0000h
CAN2IF1M1	EE14h	CAN2 IF1 mask 1	FFFFh
CAN2IF1M2	EE16h	CAN2 IF1 mask 2	FFFFh
CAN2IF1MC	EE1Ch	CAN2 IF1 message control	0000h
CAN2IF2A1	EE48h	CAN2 IF2 arbitration 1	0000h
CAN2IF2A2	EE4Ah	CAN2 IF2 arbitration 2	0000h
CAN2IF2CM	EE42h	CAN2 IF2 command mask	0000h
CAN2IF2CR	EE40h	CAN2 IF2 command request	0001h
CAN2IF2DA1	EE4Eh	CAN2 IF2 data A 1	0000h
CAN2IF2DA2	EE50h	CAN2 IF2 data A 2	0000h
CAN2IF2DB1	EE52h	CAN2 IF2 data B 1	0000h
CAN2IF2DB2	EE54h	CAN2 IF2 data B 2	0000h
CAN2IF2M1	EE44h	CAN2 IF2 mask 1	FFFFh
CAN2IF2M2	EE46h	CAN2 IF2 mask 2	FFFFh
CAN2IF2MC	EE4Ch	CAN2 IF2 message control	0000h

Table 142. X registers ordered by name (continued)

Name	Physical address	Description	Reset value
CAN2IP1	EEA0h	CAN2 interrupt pending 1	0000h
CAN2IP2	EEA2h	CAN2 interrupt pending 2	0000h
CAN2IR	EE08h	CAN2 interrupt register	0000h
CAN2MV1	EEB0h	CAN2 message valid 1	0000h
CAN2MV2	EEB2h	CAN2 message valid 2	0000h
CAN2ND1	EE90h	CAN2 new data 1	0000h
CAN2ND2	EE92h	CAN2 new data 2	0000h
CAN2SR	EE02h	CAN2 status register	0000h
CAN2TR	EE0Ah	CAN2 test register	00x0h
CAN2TR1	EE80h	CAN2 transmission request 1	0000h
CAN2TR2	EE82h	CAN2 transmission request 2	0000h
I2CCCR1	EA06h	I^2C clock control register 1	0000h
I2CCCR2	EA0Eh	I^2C clock control register 2	0000h
I2CCR	EA00h	I^2C control register	0000h
I2CDR	EA0Ch	I^2C data register	0000h
I2COAR1	EA08h	I^2C own address register 1	0000h
I2COAR2	EA0Ah	I^2C own address register 2	0000h
I2CSR1	EA02h	I^2C status register 1	0000h
I2CSR2	EA04h	I^2C status register 2	0000h
RTCAH	ED14h	RTC alarm register high byte	XXXXh
RTCAL	ED12h	RTC alarm register low byte	XXXXh
RTCCON	ED00H	RTC control register	000Xh
RTCDH	ED0Ch	RTC divider counter high byte	XXXXh
RTCDL	ED0Ah	RTC divider counter low byte	XXXXh
RTCH	ED10h	RTC programmable counter high byte	XXXXh
RTCL	ED0Eh	RTC programmable counter low byte	XXXXh
RTCPH	ED08h	RTC prescaler register high byte	XXXXh
RTCPL	ED06h	RTC prescaler register low byte	XXXXh
XCLKOUTDIV	EB02h	CLKOUT divider control register	- - 00h
XDP9	EB86h	XPort 9 direction control register	0000h
XDP9CLR	EB8Ah	XPort 9 direction control register clear	0000h
XDP9SET	EB88h	XPort 9 direction control register set	0000h
XEMU0	EB76h	XBus emulation register 0 (write-only)	XXXXh
XEMU1	EB78h	XBus emulation register 1 (write-only)	XXXXh

Table 142. X registers ordered by name (continued)

Name	Physical address	Description	Reset value
XEMU2	EB7Ah	XBus emulation register 2 (write-only)	XXXXh
XEMU3	EB7Ch	XBus emulation register 3 (write-only)	XXXXh
XIR0CLR	EB14h	XInterrupt 0 clear register (write-only)	0000h
XIR0SEL	EB10h	XInterrupt 0 selection register	0000h
XIR0SET	EB12h	XInterrupt 0 set register (write-only)	0000h
XIR1CLR	EB24h	XInterrupt 1 clear register (write-only)	0000h
XIR1SEL	EB20h	XInterrupt 1 selection register	0000h
XIR1SET	EB22h	XInterrupt 1 set register (write-only)	0000h
XIR2CLR	EB34h	XInterrupt 2 clear register (write-only)	0000h
XIR2SEL	EB30h	XInterrupt 2 selection register	0000h
XIR2SET	EB32h	XInterrupt 2 set register (write-only)	0000h
XIR3CLR	EB44h	XInterrupt 3 clear selection register (write-only)	0000h
XIR3SEL	EB40h	XInterrupt 3 selection register	0000h
XIR3SET	EB42h	XInterrupt 3 set selection register (write-only)	0000h
XMISC	EB46h	XBus miscellaneous features register	0000h
XODP9	EB8Ch	XPort 9 open-drain control register	0000h
XODP9CLR	EB90h	XPort 9 open-drain control register clear	0000h
XODP9SET	EB8Eh	XPort 9 open-drain control register set	0000h
XP10	EBC0h	XPort 10 register	0000h
XP10DIDIS	EBD2h	XPort 10 digital disable control register	0000h
XP10DIDISSET	EBD4h	XPort 10 digital disable control register set	0000h
XP10DIDISCLR	EBD6h	XPort 10 digital disable control register clear	0000h
XP9	EB80h	XPort 9 register	0000h
XP9CLR	EB84h	XPort 9 register clear	0000h
XP9SET	EB82h	XPort 9 register set	0000h
XPEREMU	EB7Eh	XPERCON copy for emulation (write-only)	XXXXh
XPICON	EB26h	Extended port input threshold control register	- - 00h
XPICON10	EBD8h	XPort 10 input control register	0000h
XPICON10CLR	EBDCh	XPort 10 input control register clear	0000h
XPICON10SET	EBDAh	XPort 10 input control register set	0000h
XPICON9	EB98h	XPort 9 input control register	0000h
XPICON9CLR	EB9Ch	XPort 9 input control register clear	0000h
XPICON9SET	EB9Ah	XPort 9 input control register set	0000h
XPOLAR	EC04h	XPWM module channel polarity register	0000h

Table 142. X registers ordered by name (continued)

Name	Physical address	Description	Reset value
XPP0	EC20h	XPWM module period register 0	0000h
XPP1	EC22h	XPWM module period register 1	0000h
XPP2	EC24h	XPWM module period register 2	0000h
XPP3	EC26h	XPWM module period register 3	0000h
XPT0	EC10h	XPWM module up/down counter 0	0000h
XPT1	EC12h	XPWM module up/down counter 1	0000h
XPT2	EC14h	XPWM module up/down counter 2	0000h
XPT3	EC16h	XPWM module up/down counter 3	0000h
XPW0	EC30h	XPWM module pulse width register 0	0000h
XPW1	EC32h	XPWM module pulse width register 1	0000h
XPW2	EC34h	XPWM module pulse width register 2	0000h
XPW3	EC36h	XPWM module pulse width register 3	0000h
XPWMCON0	EC00h	XPWM module control register 0	0000h
XPWMCON0CLR	EC08h	XPWM module clear control register 0 (write-only)	0000h
XPWMCON0SET	EC06h	XPWM module set control register 0 (write-only)	0000h
XPWMCON1	EC02h	XPWM module control register 1	0000h
XPWMCON1CLR	EC0Ch	XPWM module clear control register 0 (write-only)	0000h
XPWMCON1SET	EC0Ah	XPWM module set control register 0 (write-only)	0000h
XPWMPORT	EC80h	XPWM module port control register	0000h
XS1BG	E906h	XASC baud rate generator reload register	0000h
XS1CON	E900h	XASC control register	0000h
XS1CONCLR	E904h	XASC clear control register (write-only)	0000h
XS1CONSET	E902h	XASC set control register (write-only)	0000h
XS1PORT	E980h	XASC port control register	0000h
XS1RBUF	E90Ah	XASC receive buffer register	0000h
XS1TBUF	E908h	XASC transmit buffer register	0000h
XSSCBR	E80Ah	XSSC baud rate register	0000h
XSSCCON	E800h	XSSC control register	0000h
XSSCCONCLR	E804h	XSSC clear control register (write-only)	0000h
XSSCCONSET	E802h	XSSC set control register (write-only)	0000h
XSSCPORT	E880h	XSSC port control register	0000h
XSSCRB	E808h	XSSC receive buffer	XXXXh
XSSCTB	E806h	XSSC transmit buffer	0000h
XTCR	EB50h	XTimer control register	0000h

Table 142. X registers ordered by name (continued)

Name	Physical address	Description	Reset value
XTCVR	EB56h	XTimer current value register	0000h
XTEVR	EB54h	XTimer end value register	0000h
XTSVR	EB52h	XTimer start value register	0000h

23.6 X registers ordered by address

Table 143 lists all XBus registers which are implemented in the ST10F296E ordered by their physical address. The Flash control registers are physically mapped on the XBus memory space, but, are listed in *Section 23.7: Flash registers ordered by name*. The X registers are not bit-addressable.

Table 143. X registers ordered by address

Name	Physical address	Description	Reset value
XSSCCON	E800h	XSSC control register	0000h
XSSCCONSET	E802h	XSSC set control register (write-only)	0000h
XSSCCONCLR	E804h	XSSC clear control register (write-only)	0000h
XSSCTB	E806h	XSSC transmit buffer	0000h
XSSCRB	E808h	XSSC receive buffer	XXXXh
XSSCBR	E80Ah	XSSC baud rate register	0000h
XSSCPORT	E880h	XSSC port control register	0000h
XS1CON	E900h	XASC control register	0000h
XS1CONSET	E902h	XASC set control register (write-only)	0000h
XS1CONCLR	E904h	XASC clear control register (write-only)	0000h
XS1BG	E906h	XASC baud rate generator reload register	0000h
XS1TBUF	E908h	XASC transmit buffer register	0000h
XS1RBUF	E90Ah	XASC receive buffer register	0000h
XS1PORT	E980h	XASC port control register	0000h
I2CCR	EA00h	I^2C control register	0000h
I2CSR1	EA02h	I^2C status register 1	0000h
I2CSR2	EA04h	I^2C status register 2	0000h
I2CCCR1	EA06h	I^2C clock control register 1	0000h
I2COAR1	EA08h	I^2C own address register 1	0000h
I2COAR2	EA0Ah	I^2C own address register 2	0000h
I2CDR	EA0Ch	I^2C data register	0000h
I2CCCR2	EA0Eh	I^2C clock control register 2	0000h

Table 143. X registers ordered by address (continued)

Name	Physical address	Description	Reset value
XCLKOUTDIV	EB02h	CLKOUT divider control register	- - 00h
XIR0SEL	EB10h	XInterrupt 0 selection register	0000h
XIR0SET	EB12h	XInterrupt 0 set register (write-only)	0000h
XIR0CLR	EB14h	XInterrupt 0 clear register (write-only)	0000h
XIR1SEL	EB20h	XInterrupt 1 selection register	0000h
XIR1SET	EB22h	XInterrupt 1 set register (write-only)	0000h
XIR1CLR	EB24h	XInterrupt 1 clear register (write-only)	0000h
XPICON	EB26h	Extended port input threshold control register	- - 00h
XIR2SEL	EB30h	XInterrupt 2 selection register	0000h
XIR2SET	EB32h	XInterrupt 2 set register (write-only)	0000h
XIR2CLR	EB34h	XInterrupt 2 clear register (write-only)	0000h
XIR3SEL	EB40h	XInterrupt 3 selection register	0000h
XIR3SET	EB42h	XInterrupt 3 set selection register (write-only)	0000h
XIR3CLR	EB44h	XInterrupt 3 clear selection register (write-only)	0000h
XMISC	EB46h	XBus miscellaneous features register	0000h
XTCR	EB50h	XTimer control register	0000h
XTSVR	EB52h	XTimer start value register	0000h
XTEVR	EB54h	XTimer end value register	0000h
XTCVR	EB56h	XTimer current value register	0000h
XEMU0	EB76h	XBus emulation register 0 (write-only)	XXXXh
XEMU1	EB78h	XBus emulation register 1 (write-only)	XXXXh
XEMU2	EB7Ah	XBus emulation register 2 (write-only)	XXXXh
XEMU3	EB7Ch	XBus emulation register 3 (write-only)	XXXXh
XPEREMU	EB7Eh	XPERCON copy for emulation (write-only)	XXXXh
XP9	EB80h	XPort 9 register	0000h
XP9SET	EB82h	XPort 9 register set	0000h
XP9CLR	EB84h	XPort 9 register clear	0000h
XDP9	EB86h	XPort 9 direction control register	0000h
XDP9SET	EB88h	XPort 9 direction control register set	0000h
XDP9CLR	EB8Ah	XPort 9 direction control register clear	0000h
XODP9	EB8Ch	XPort 9 open-drain control register	0000h
XODP9SET	EB8Eh	XPort 9 open-drain control register set	0000h
XODP9CLR	EB90h	XPort 9 open-drain control register clear	0000h
XPICON9	EB98h	XPort 9 input control register	0000h

Table 143. X registers ordered by address (continued)

Name	Physical address	Description	Reset value
XPICON9SET	EB9Ah	XPort 9 input control register set	0000h
XPICON9CLR	EB9Ch	XPort 9 input control register clear	0000h
XP10	EBC0h	XPort 10 register	0000h
XP10DIDIS	EBD2h	XPort 10 digital disable control register	0000h
XP10DIDISSET	EBD4h	XPort 10 digital disable control register set	0000h
XP10DIDISCLR	EBD6h	XPort 10 digital disable control register clear	0000h
XPICON10	EBD8h	XPort 10 input control register	0000h
XPICON10SET	EBDAh	XPort 10 Input Control Register Set	0000h
XPICON10CLR	EBDCh	XPort 10 input control register clear	0000h
XPWMCON0	EC00h	XPWM module control register 0	0000h
XPWMCON1	EC02h	XPWM module control register 1	0000h
XPOLAR	EC04h	XPWM module channel polarity register	0000h
XPWMCON0SET	EC06h	XPWM module set control register 0 (write-only)	0000h
XPWMCON0CLR	EC08h	XPWM module clear control register 0 (write-only)	0000h
XPWMCON1SET	EC0Ah	XPWM module set control register 0 (write-only)	0000h
XPWMCON1CLR	EC0Ch	XPWM module clear control register 0 (write-only)	0000h
XPT0	EC10h	XPWM module up/down counter 0	0000h
XPT1	EC12h	XPWM module up/down counter 1	0000h
XPT2	EC14h	XPWM module up/down counter 2	0000h
XPT3	EC16h	XPWM module up/down counter 3	0000h
XPP0	EC20h	XPWM module period register 0	0000h
XPP1	EC22h	XPWM module period register 1	0000h
XPP2	EC24h	XPWM module period register 2	0000h
XPP3	EC26h	XPWM module period register 3	0000h
XPW0	EC30h	XPWM module pulse width register 0	0000h
XPW1	EC32h	XPWM module pulse width register 1	0000h
XPW2	EC34h	XPWM module pulse width register 2	0000h
XPW3	EC36h	XPWM module pulse width register 3	0000h
XPWMPORT	EC80h	XPWM module port control register	0000h
RTCCON	ED00H	RTC control register	000Xh
RTCPL	ED06h	RTC prescaler register low byte	XXXXh
RTCPH	ED08h	RTC prescaler register high byte	XXXXh
RTCDL	ED0Ah	RTC divider counter low byte	XXXXh
RTCDH	ED0Ch	RTC divider counter high byte	XXXXh

Table 143. X registers ordered by address (continued)

Name	Physical address	Description	Reset value
RTCL	ED0Eh	RTC programmable counter low byte	XXXXh
RTCH	ED10h	RTC programmable counter high byte	XXXXh
RTCAL	ED12h	RTC alarm register low byte	XXXXh
RTCAH	ED14h	RTC alarm register high byte	XXXXh
CAN2CR	EE00h	CAN2 CAN control register	0001h
CAN2SR	EE02h	CAN2 status register	0000h
CAN2EC	EE04h	CAN2 error counter	0000h
CAN2BTR	EE06h	CAN2 bit timing register	2301h
CAN2IR	EE08h	CAN2 interrupt register	0000h
CAN2TR	EE0Ah	CAN2 test register	00x0h
CAN2BRPER	EE0Ch	CAN2 BRP extension register	0000h
CAN2IF1CR	EE10h	CAN2 IF1 command request	0001h
CAN2IF1CM	EE12h	CAN2 IF1 command mask	0000h
CAN2IF1M1	EE14h	CAN2 IF1 mask 1	FFFFh
CAN2IF1M2	EE16h	CAN2 IF1 mask 2	FFFFh
CAN2IF1A1	EE18h	CAN2 IF1 arbitration 1	0000h
CAN2IF1A2	EE1Ah	CAN2 IF1 arbitration 2	0000h
CAN2IF1MC	EE1Ch	CAN2 IF1 message control	0000h
CAN2IF1DA1	EE1Eh	CAN2 IF1 data A 1	0000h
CAN2IF1DA2	EE20h	CAN2 IF1 data A 2	0000h
CAN2IF1DB1	EE22h	CAN2 IF1 data B 1	0000h
CAN2IF1DB2	EE24h	CAN2 IF1 data B 2	0000h
CAN2IF2CR	EE40h	CAN2 IF2 command request	0001h
CAN2IF2CM	EE42h	CAN2 IF2 command mask	0000h
CAN2IF2M1	EE44h	CAN2 IF2 mask 1	FFFFh
CAN2IF2M2	EE46h	CAN2 IF2 mask 2	FFFFh
CAN2IF2A1	EE48h	CAN2 IF2 arbitration 1	0000h
CAN2IF2A2	EE4Ah	CAN2 IF2 arbitration 2	0000h
CAN2IF2MC	EE4Ch	CAN2 IF2 message control	0000h
CAN2IF2DA1	EE4Eh	CAN2 IF2 data A 1	0000h
CAN2IF2DA2	EE50h	CAN2 IF2 data A 2	0000h
CAN2IF2DB1	EE52h	CAN2 IF2 data B 1	0000h
CAN2IF2DB2	EE54h	CAN2 IF2 data B 2	0000h
CAN2TR1	EE80h	CAN2 transmission request 1	0000h

Table 143. X registers ordered by address (continued)

Name	Physical address	Description	Reset value
CAN2TR2	EE82h	CAN2 transmission request 2	0000h
CAN2ND1	EE90h	CAN2 new data 1	0000h
CAN2ND2	EE92h	CAN2 new data 2	0000h
CAN2IP1	EEA0h	CAN2 interrupt pending 1	0000h
CAN2IP2	EEA2h	CAN2 interrupt pending 2	0000h
CAN2MV1	EEB0h	CAN2 message valid 1	0000h
CAN2MV2	EEB2h	CAN2 message valid 2	0000h
CAN1CR	EF00h	CAN1 CAN control register	0001h
CAN1SR	EF02h	CAN1 status register	0000h
CAN1EC	EF04h	CAN1 error counter	0000h
CAN1BTR	EF06h	CAN1 bit timing register	2301h
CAN1IR	EF08h	CAN1 interrupt register	0000h
CAN1TR	EF0Ah	CAN1 test register	00x0h
CAN1BRPER	EF0Ch	CAN1 BRP extension register	0000h
CAN1IF1CR	EF10h	CAN1 IF1 command request	0001h
CAN1IF1CM	EF12h	CAN1 IF1 command mask	0000h
CAN1IF1M1	EF14h	CAN1 IF1 mask 1	FFFFh
CAN1IF1M2	EF16h	CAN1 IF1 mask 2	FFFFh
CAN1IF1A1	EF18h	CAN1 IF1 arbitration 1	0000h
CAN1IF1A2	EF1Ah	CAN1 IF1 arbitration 2	0000h
CAN1IF1MC	EF1Ch	CAN1 IF1 message control	0000h
CAN1IF1DA1	EF1Eh	CAN1 IF1 data A 1	0000h
CAN1IF1DA2	EF20h	CAN1 IF1 data A 2	0000h
CAN1IF1DB1	EF22h	CAN1 IF1 data B 1	0000h
CAN1IF1DB2	EF24h	CAN1 IF1 data B 2	0000h
CAN1IF2CR	EF40h	CAN1 IF2 command request	0001h
CAN1IF2CM	EF42h	CAN1 IF2 command mask	0000h
CAN1IF2M1	EF44h	CAN1 IF2 mask 1	FFFFh
CAN1IF2M2	EF46h	CAN1 IF2 mask 2	FFFFh
CAN1IF2A1	EF48h	CAN1 IF2 arbitration 1	0000h
CAN1IF2A2	EF4Ah	CAN1 IF2 arbitration 2	0000h
CAN1IF2MC	EF4Ch	CAN1 IF2 message control	0000h
CAN1IF2DA1	EF4Eh	CAN1 IF2 data A 1	0000h
CAN1IF2DA2	EF50h	CAN1 IF2 data A 2	0000h

Table 143. X registers ordered by address (continued)

Name	Physical address	Description	Reset value
CAN1IF2DB1	EF52h	CAN1 IF2 data B 1	0000h
CAN1IF2DB2	EF54h	CAN1 IF2 data B 2	0000h
CAN1TR1	EF80h	CAN1 transmission request 1	0000h
CAN1TR2	EF82h	CAN1 transmission request 2	0000h
CAN1ND1	EF90h	CAN1 new data 1	0000h
CAN1ND2	EF92h	CAN1 new data 2	0000h
CAN1IP1	EFA0h	CAN1 interrupt pending 1	0000h
CAN1IP2	EFA2h	CAN1 interrupt pending 2	0000h
CAN1MV1	EFB0h	CAN1 message valid 1	0000h
CAN1MV2	EFB2h	CAN1 message valid 2	0000h

23.7 Flash registers ordered by name

Table 144 lists all Flash control registers which are implemented in the ST10F296E ordered by their name. As these registers are physically mapped on the XBus, they are not bit-addressable.

Table 144. Flash registers ordered by name

Name	Physical address	Description	Reset value
FARH	0x000E 0012	Flash address register high	0000h
FARL	0x000E 0010	Flash address register low	0000h
FCR0H	0x000E 0002	Flash control register 0 - high	0000h
FCR0L	0x000E 0000	Flash control register 0 - low	0000h
FCR1H	0x000E 0006	Flash control register 1 - high	0000h
FCR1L	0x000E 0004	Flash control register 1 - low	0000h
FDR0H	0x000E 000A	Flash data register 0 - high	FFFFh
FDR0L	0x000E 0008	Flash data register 0 - low	FFFFh
FDR1H	0x000E 000E	Flash data register 1 - high	FFFFh
FDR1L	0x000E 000C	Flash data register 1 - low	FFFFh
FER	0x000E 0014	Flash error register	0000h
FNVAPR0	0x000E DFB8	Flash non volatile access protection register 0	ACFFh
FNVAPR1H	0x000E DFBE	Flash non volatile access protection register 1 - high	FFFFh
FNVAPR1L	0x000E DFBC	Flash non volatile access protection register 1 - low	FFFFh
FNVWPIRH	0x000E DFB6	Flash non volatile protection I register high	FFFFh
FNVWPIRL	0x000E DFB4	Flash non volatile protection I register low	FFFFh
FNVWPXRH	0x000E DFB2	Flash non volatile protection X register high	FFFFh
FNVWPXRL	0x000E DFB0	Flash non volatile protection X register low	FFFFh
XFICR	0x000E E000	XFlash interface control register	000Fh

23.8 Flash registers ordered by address

Table 145 lists all Flash control registers which are implemented in the ST10F296E ordered by their physical address. As these registers are physically mapped on the XBus, they are not bit-addressable.

Table 145. Flash registers ordered by address

Name	Physical address	Description	Reset value
FCR0L	0x000E 0000	Flash control register 0 - low	0000h
FCR0H	0x000E 0002	Flash control register 0 - high	0000h
FCR1L	0x000E 0004	Flash control register 1 - low	0000h
FCR1H	0x000E 0006	Flash control register 1 - high	0000h
FDR0L	0x000E 0008	Flash data register 0 - low	FFFFh
FDR0H	0x000E 000A	Flash data register 0 - high	FFFFh
FDR1L	0x000E 000C	Flash data register 1 - low	FFFFh
FDR1H	0x000E 000E	Flash data register 1 - high	FFFFh
FARL	0x000E 0010	Flash address register low	0000h
FARH	0x000E 0012	Flash address register high	0000h
FER	0x000E 0014	Flash error register	0000h
FNVWPXRL	0x000E DFB0	Flash non volatile protection X register low	FFFFh
FNVWPXRH	0x000E DFB2	Flash non volatile protection X register high	FFFFh
FNVWPIRL	0x000E DFB4	Flash non volatile protection I register low	FFFFh
FNVWPIRH	0x000E DFB6	Flash non volatile protection I register high	FFFFh
FNVAPR0	0x000E DFB8	Flash non volatile access protection register 0	ACFFh
FNVAPR1L	0x000E DFBC	Flash non volatile access protection register 1 - low	FFFFh
FNVAPR1H	0x000E DFBE	Flash non volatile access protection register 1 - high	FFFFh
XFICR	0x000E E000	XFlash interface control register	000Fh

23.9 Identification registers

The ST10F296E has four identification registers, mapped in the ESFR space. These registers contain:

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

IDMANUF register

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

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

Table 146. IDMANUF register description

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

IDCHIP register

IDCHIP (F07Ch/3Eh) ESFR Reset value: 128Xh

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

Table 147. IDCHIP register description

Bit	Bit name	Function
15-14	PCONF	Peripheral configuration 00: (E) Enhanced (ST10F296E) 01: (B) Basic 10: (D) Dedicated 11: Reserved
13 - 4	IDCHIP	Device identifier 128h: ST10F296E identifier (128h = 296)
3-0	REVID	Device revision identifier Xh: According to revision number

IDMEM register

IDMEM (F07Ah/3Dh) ESFR Reset value: 30D0h

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

Table 148. IDMEM register description

Bit	Bit name	Function
15-12	MEMTYP	Internal memory type 0h: ROM-less 1h: (M) ROM memory 2h: (S) Standard Flash memory 3h: (H) High performance Flash memory (ST10F296E) 4h...Fh: Reserved
11 - 0	MEMSIZE	Internal memory size Internal memory size is 4 x (MEMSIZE) (in Kbyte). The 0D0h for the ST10F296E is 832 Kbytes

IDPROG register

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

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

Table 149. IDPROG register description

Bit	Bit name	Function
15-8	PROGVPP	Programming V_{PP} voltage (no need of external V_{PP}) - 00h No need for external V_{PP} (00h)
7-0	PROGVDD	Programming V_{DD} voltage When programming EPROM or Flash devices, V_{DD} voltage is calculated using the following formula for 5 V ST10F296E devices: V_{DD} = 20 x [PROGVDD] / 256 (volts) - 40h

Note: All identification registers are read-only registers.

The values written inside different identification register bits are valid only after the Flash initialization phase has been completed. When code execution starts from the internal memory (pin $\overline{EA}$ held high during reset), the Flash has completed initialization and the identification register bits can be read. When code execution starts from the external memory (pin $\overline{EA}$ held low during reset), Flash initialization has not been completed and the identification register bits cannot be read. The user can poll bits 15 and 14 of the IDMEM register. When both these bits are read low, Flash initialization can be completed and all identification register bits can be read.

Before Flash initialization completion, the default settings of the different identification registers are as follows:

- IDMANUF: 0403h
- IDCHIP: 128xh (x = silicon revision)
- IDMEM: F0D0h
- IDPROG: 0040h

23.10 System configuration registers

This section lists and describes 12 registers which are used for configuring various aspects of the ST10F296E system.

System configuration register (SYSCON)

SYSCON (FF12h/89h) SFR Reset value: 0xx0h(1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STKSZ			ROMS1	SGTDIS	ROMEN	BYTDIS	CLKEN	WRCFG	CSCFG	PWDCFG	OWDDIS	BDRSTEN	XPEN	VISIBLE	XPERSHARE
RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

1. SYSCON reset value is: 0000 0xx0 0x00 0000b.

Table 150. SYSCON register description

Bit	Bit name	Function
15-13	STKSZ	System stack size Selects the size of the system stack (in the internal IRAM) from 32 to 1024 words.
12	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).
11	SGTDIS	Segmentation disable/enable control 0: Segmentation enabled (CSP is saved/restored during interrupt entry/exit). 1: Segmentation disabled (only the IP is saved/restored).
10	ROMEN(1)	Internal memory enable (set according to the $\overline{EA}$ pin during reset) 0: Internal memory disabled. Accesses to the IFlash memory area is made through the external bus. 1: Internal memory enabled.
9	BYTDIS(1)	Disable/enable control for the BHE pin (set according to data bus width) 0: $\overline{BHE}$ pin enabled. 1: $\overline{BHE}$ pin disabled. Pin may be used for general purpose I/O.
8	CLKEN	System clock output enable (CLKOUT) 0: CLKOUT disabled. Pin may be used for general purpose I/O. 1: CLKOUT enabled. Pin outputs the system clock signal or a prescaled value of the system clock according to the XCLKOUTDIV register setting.

Table 150. SYSCON register description (continued)

Bit	Bit name	Function
7	WRCFG(1)	Write configuration control (inverted copy of the WRC bit of the RP0H register) 0: $\overline{WR}$ and $\overline{BHE}$ pins retain their normal function. 1: $\overline{WR}$ and $\overline{BHE}$ pins behave as the $\overline{WRL}$ and $\overline{WRH}$ pins respectively
6	CSCFG	Chip select configuration control 0: Latched chip select lines, $\overline{CSx}$ changes 1 TCL after rising edge of ALE 1: Unlatched chip select lines, $\overline{CSx}$ changes with rising edge of ALE
5	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. To exit power-down mode, an external reset must occur by asserting the $\overline{RSTIN}$ 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.
4	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 µs, the CPU clock is switched automatically to PLL's base frequency (250 Hz to 4 MHz). 1: OWD is disabled. If the PLL is bypassed, the CPU clock is always driven by the XTAL1 signal. The PLL is turned off to reduce power supply current.
3	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.
2	XPEN	XBus peripheral enable bit 0: Access to the on-chip XPeripherals and their functions are disabled. 1: The on-chip XPeripherals are enabled and can be accessed.
1	VISIBLE	Visible mode control 0: Access to the XBus peripherals is made internally. 1: Access to the XBus peripherals is made visible on the external pins.
0	XPERSHARE	XBus peripheral share mode control 0: External access to the XBus peripherals is disabled. 1: XRAM1 and XRAM2 are accessible via the external bus during hold mode. External access to other XBus peripherals is not guaranteed in terms of AC timings.

1. Bits are set directly or indirectly during the rest sequence according to Port 0 and the $\overline{EA}$ pin configuration.

BUSCON0 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN0	CSR EN0	RDY POL0	RDY EN0	-	BUS ACT0	ALE CTL0	-	BTYP		MTT C0	RWD C0	MCTC			
RW	RW	RW	RW	-	RW	RW	-	RW		RW	RW	RW			

BUSCON1 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN1	CSR EN1	RDY POL1	RDY EN1	-	BUS ACT1	ALE CTL1	-	BTYP		MTT C1	RWD C1	MCTC			
RW	RW	RW	RW	-	RW	RW	-	RW		RW	RW	RW			

BUSCON2 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN2	CSR EN2	RDY POL2	RDY EN2	-	BUS ACT2	ALE CTL2	-	BTYP		MTT C2	RWD C2	MCTC			
RW	RW	RW	RW	-	RW	RW	-	RW		RW	RW	RW			

BUSCON3 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN3	CSR EN3	RDY POL3	RDY EN3	-	BUS ACT3	ALE CTL3	-	BTYP		MTT C3	RWD C3	MCTC			
RW	RW	RW	RW	-	RW	RW	-	RW		RW	RW	RW			

BUSCON4 register

BUSCON4 (FF1Ah/8Dh) SFR Reset value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN4	CSR EN4	RDY POL4	RDY EN4	-	BUS ACT4	ALE CTL4	-	BTYP		MTT C4	RWD C4	MCTC			
RW	RW	RW	RW	-	RW	RW	-	RW		RW	RW	RW			

Table 151. BUSCONx register description

Bit	Bit name	Function
15	CSWENx	Write chip select enable 0: The $\overline{CS}$ signal is independent of the write command ($\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$). 1: The $\overline{CS}$ signal is generated for the duration of the write command.
14	CSRENx	Read chip select enable 0: The $\overline{CS}$ signal is independent of the read command ($\overline{RD}$). 1: The $\overline{CS}$ signal is generated for the duration of the read command.
13	RDYPOLx	Ready active level control 0: Active level on the READY pin is low and he bus cycle terminates with an 0 on this pin. 1: Active level on the READY pin is high and the bus cycle terminates with a 1 on this pin.
12	RDYENx	READY input enable 0: External bus cycle is controlled by the MCTC bit field. 1: External bus cycle is controlled by the $\overline{READY}$ input signal.
10	BUSACTx	Bus active control 0: External bus disabled. 1: External bus enabled (within the respective address window, see ADDRSEL register.
9	ALECTLx	ALE lengthening control 0: Normal ALE signal. 1: Lengthened ALE signal.
7-6	BTYP	External bus configuration 00: 8-bit demultiplexed bus 01: 8-bit multiplexed bus 10: 16-bit demultiplexed bus 11: 16-bit multiplexed bus *Note 1: BTYP bits of BUSCON0 are defined via Port 0 during reset. They are set according to the configuration of bit 6 and 7 of Port 0 latched at the end of the reset sequence.* *Note 2: If the $\overline{EA}$ pin is high during reset, the BUSCON0 register is initialized with 0000h. If $\overline{EA}$ pin is low during reset, the BUSACT0 and ALECTL0 bits are set (1) and the BTYP bit field is loaded with the bus configuration selected via Port 0.*
5	MTTCx	Memory tristate time control 0: 1 wait state. 1: No wait state.
4	RWDCx	Read/write delay control for BUSCONx 0: With read/write delay, the CPU inserts 1 TCL after falling edge of ALE. 1: No read/write delay; $\overline{RW}$ is activated after falling edge of ALE.
3-0	MCTC	Memory cycle time control (number of memory cycle time wait states) 0000: 15 wait states (number of wait states = 15 - [MCTC]). . . . 1111: No wait states.

RP0H register

RP0H is a read-only register.

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)

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

Table 152. RP0H register description

Bit	Bit name	Function
7-5	CLKSEL	System clock selection 000: $f_{CPU} = 16 \times f_{OSC}$ 001: $f_{CPU} = 0.5 \times f_{OSC}$ 010: $f_{CPU} = 10 \times f_{OSC}$ 011: $f_{CPU} = f_{OSC}$ 100: $f_{CPU} = 5 \times f_{OSC}$ 101: $f_{CPU} = 8 \times f_{OSC}$ 110: $f_{CPU} = 3 \times f_{OSC}$ 111: $f_{CPU} = 4 \times f_{OSC}$
4-3	SALSEL	Segment address line selection (number of active segment address outputs) 00: 4-bit segment addresses, A19 to A16. 01: No segment address lines. 10: 8-bit segment addresses, A23 to A16. 11: 2-bit segment address, A17 and A16 (default without pull-downs).
2-1	CSSEL	Chip select line selection (number of active $\overline{CS}$ outputs) 00: Three $\overline{CS}$ lines, $\overline{CS}2$ to $\overline{CS}0$. 01: Two $\overline{CS}$ lines, $\overline{CS}1$ and $\overline{CS}0$. 10: No $\overline{CS}$ lines. 11: Five $\overline{CS}$ lines, $\overline{CS}4$ to $\overline{CS}0$ (default without pull-downs)
0	WRC	Write configuration control 0: $\overline{WR}$ and $\overline{BHE}$ pins behave as $\overline{WRL}$ and $\overline{WRH}$ pins respectively. 1: $\overline{WR}$ and $\overline{BHE}$ pins retain their normal functioning.

EXICON 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(1)(2)		EXI2ES(1)(3)		EXI1ES		EXI0ES	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

1. EXI2ES and EXI3ES must be configured as 01b because RTC 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

Table 153. EXICON register description

Bit	Bit name	Function
15-0	EXIxES (x = 7 to 0)	External interrupt x edge selection field (x = 7...0) 00: Fast external interrupts disabled (standard mode). EXxIN pin not taken into account for entering/exiting power-down mode. 01: Interrupt on positive rising edge. Power-down mode is entered if EXiIN = 0 and exited if EXxIN = 1 (referred as 'high' active level). 10: Interrupt on negative falling edge. Power-down mode is entered if EXiIN = 1 and exited if EXxIN = 0 (referred as 'low' active level). 11: Interrupt on any edge (rising or falling). Power-down mode is always entered. It is exited if the EXxIN level changes.

EXISEL register

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(1)		EXI2SS(2)		EXI1SS		EXI0SS	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

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

Table 154. EXISEL register description

Bit	Bit name	Function
15-0	EXIxSS	External interrupt x source selection (x = 7 to 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

Table 155. External interrupt selection

EXIxSS	Port 2 pin	Alternate source	
0	P2.8	CAN1_RxD	P4.5
1	P2.9	CAN2_RxD/SCL	P4.4
2	P2.10	RTCSI (second)	Internal MUX
3	P2.11	RTCAI (alarm)	Internal MUX
4 to 7	P2.12 to 15	Not used (zero)	-

XP3IC register

This register has the same bit field as the xxIC interrupt register (see below).

XP3IC (F19Eh/CFh) 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	

xxIC register

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	

Table 156. xxIC register description

Bit	Bit name	Function
7	xxIR	Interrupt request flag 0: No request pending 1: This source has raised an interrupt request
6	xxIE	Interrupt enable control bit (individually enables/disables a specific source) 0: Interrupt request is disabled 1: Interrupt request is enabled
5-2	ILVL	Interrupt priority level Defines the priority level for the arbitration of requests. Fh: Highest priority level 0h: Lowest priority level
1-0	GLVL	Group level Defines the internal order for simultaneous requests of the same priority. 3: Highest group priority 0: Lowest group priority

XPERCON register

XPERCON (F024h/12h) ESFR Reset value: 005h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	XPORT EN	XMISC EN	XI2C EN	XSSC EN	XASC EN	XPWM EN	XFLASH EN	XRTC EN	XRAM 2EN	XRAM 1EN	CAN 2EN	CAN 1EN
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 157. XPERCON register description

Bit	Bit name	Function
11	XPORTEN	XPort 9 and XPort 10 enable bit 0: Access to the on-chip XPort 9 and XPort 10 modules is disabled. Address range 00'EB80h to 00'EBFFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XI2CEN and XMISCEN are also 0. 1: The on-chip XPort 9 and XPort 10 are enabled and can be accessed.
10	XMISCEN	XBus additional features and XTimer enable bit 0: Access to the additional miscellaneous features is disabled. Address range 00'EB00h to 00'EB7Fh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XI2CEN and XPORTEN are also 0. 1: The additional features and XTimer are enabled and can be accessed.
9	XI2CEN	I^2C enable bit 0: Access to the on-chip I^2C is disabled, external access performed. Address range 00'EA00h to 00'EAFFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XSSCEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip I^2C is enabled and can be accessed.
8	XSSCEN	SSC1 enable bit 0: Access to the on-chip SSC1 is disabled, external access performed. Address range 00'E800h to 00'E8FFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip SSC1 is enabled and can be accessed.
7	XASCEN	ASC1 enable bit 0: Access to the on-chip ASC1 is disabled, external access performed. Address range 00'E900h to 00'E9FFh is directed to the external memory only if CAN1EN, CAN2EN, XRTCEN, XASCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip ASC1 is enabled and can be accessed.
6	XPWMEN	XPWM enable 0: Access to the on-chip PWM1 module is disabled, external access is performed. Address range 00'EC00h to 00'ECFF is directed to the external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XRTCEN, XMISCEN and XPORTEN are also 0. 1: The on-chip PWM1 module is enabled and can be accessed.

Table 157. XPERCON register description

Bit	Bit name	Function
5	XFLASHEN	XFlash enable bit 0: Access to the on-chip XFlash is disabled, external access is performed. Address range 09'0000h to 0E'FFFFh is directed to the external memory only if XRAM2EN is also 0. 1: The on-chip XFlash is enabled and can be accessed.
4	XRTCEN	RTC enable 0: Access to the on-chip RTC module is disabled, external access is performed. Address range 00'ED00h to 00'EDFF is directed to the external memory only if CAN1EN, CAN2EN, XASCEN, XSSCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0. 1: The on-chip RTC module is enabled and can be accessed.
3	XRAM2EN	XRAM2 enable bit 0: Access to the on-chip 64 KByte XRAM is disabled, external access is performed. Address range 0F'0000h to 0F'FFFFh is directed to the external memory only if XFLASHEN is also 0. 1: The on-chip 64 Kbyte XRAM is enabled and can be accessed.
2	XRAM1EN	XRAM1 enable bit 0: Access to the on-chip 2 KByte XRAM is disabled. Address range 00'E000h to 00'E7FFh is directed to the external memory. 1: The on-chip 2 Kbyte XRAM is enabled and can be accessed.
1	CAN2EN	CAN2 enable bit 0: Access to the on-chip CAN2 XPeripheral and its functions is disabled (P4.4 and P4.7 pins can be used as general purpose IOs, but, address range 00'EC00h to 00'EFFFh is directed to the external memory only if CAN1EN, XRTCEN, XASCEN, XSSCEN, XI2CEN, XPWMEN, XMISCEN and XPORTEN are also 0). 1: The on-chip CAN2 XPeripheral is enabled and can be accessed.
0	CAN1EN	CAN1 enable bit 0: Access to the on-chip CAN1 XPeripheral and its functions is disabled (P4.5 and P4.6 pins can be used as general purpose IOs, but, address range 00'EC00h to 00'EFFFh is directed to the external memory only if CAN2EN, XRTCEN, XASCEN, XSSCEN, XI2CEN, XPWMEN an XMISCEN are also 0). 1: The on-chip CAN1 XPeripheral is enabled and can be accessed.

When CAN1, CAN2, RTC, ASC1, SSC1, I^2C, PWM1, XBus additional features, XTimer and XPort modules are disabled via XPERCON settings, any access in the address range 00'E800h to 00'EFFFh is directed to the external memory interface, using the BUSCONx register associated with the ADDRSELx register matching the target address. All pins involved with the XPeripherals can be used as general purpose IOs whenever the related module is not enabled.

The default XPER selection after reset is identical to configuration of the XBus in the ST10F280. CAN1 and XRAM1 are enabled, CAN2 and XRAM2 are disabled, all other XPeripherals are disabled after reset.

the XPERCON register cannot be changed after globally enabling the XPeripherals (after setting the XPEN bit in the SYSCON register).

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

Reserved bits of the XPERCON register must always be written to 0.

When the RTC is disabled (RTCEN = 0) the main clock oscillator is switched off if the ST10 enters power-down mode. When the RTC is enabled, the RTCOFF bit of the RTCCON register allows the power-down mode of the main clock oscillator to be chosen (eee *Section 18: Real-time clock (RTC) on page 203*).

Table 158 summarizes the address range mapping on segment 8 for programming the ROMEN and XPEN bits (of the SYSCON register) and the XRAM2EN and XFLASHEN bits (of the XPERCON register).

Table 158. Segment 8 address range mapping

ROMEN	XPEN	XRAM2EN	XFLASHEN	Segment 8
0	0	x(1)	x(1)	External memory
0	1	0	0	External memory
0	1	1	x(1)	Reserved
0	1	x(1)	1	Reserved
1	x(1)	x(1)	x(1)	IFlash (B1F1)

1. Don't care

23.10.1 XPEREMU register

The XPEREMU register is a write-only register that is mapped on the XBus memory space at address EB7Eh. It contrasts with the XPERCON register, a read/write ESFR register, which must be programmed to enable the single XBus modules separately.

Once the XPEN bit of the SYSCON register is set and at least one of the XPeripherals (except the memories) is activated, the XPEREMU register must be written with the same content as the XPERCON register. This is to allow a correct emulation of the new set of features introduced on the XBus for the new ST10 generation. The following instructions must be added inside the initialization routine:

```
if (SYSCON.XPEN && (XPERCON & 0x07D3))
then { XPEREMU = XPERCON }
```

XPEREMU must be programmed after both the XPERCON and SYSCON registers in such a way that the final configuration for the XPeripherals is stored in the XPEREMU register and used for the emulation hardware setup.

XPEREMU (EB7Eh) XBus Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	XPORT EN	XMISC EN	XI2C EN	XSSC EN	XASC EN	XPWM EN	XFLASH EN	XRTC EN	XRAM 2EN	XRAM 1EN	CAN 2EN	CAN 1EN
-	-	-	-	W	W	W	W	W	W	W	W	W	W	W	W

XPEREMU bit descriptition follows the XPERCON register (see *Table 5* and *Table 157*).

23.11 Emulation dedicated registers

Four write-only registers of the ST10F296E are described briefly below. These registers are used for emulation purposes only.

XEMU0 (EB76h) XBus Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU0(15:0)															
W															

XEMU1 (EB78h) XBus Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU1(15:0)															
W															

XEMU2 (EB7Ah) XBus Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU2(15:0)															
W															

XEMU3 (EB7Ch) XBus Reset value: xxxxh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XEMU3(15:0)															
W															

24 Electrical characteristics

24.1 Absolute maximum ratings

Table 159. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{DD}	Voltage on V_{DD} pins with respect to ground (V_{SS})	- 0.3 to +6.5	V
V_{STBY}	Voltage on V_{STBY} pin with respect to ground (V_{SS})		
V_{AREF}	Voltage on V_{AREF} pin with respect to ground (V_{SS})	- 0.3 to V_{DD} + 0.3	
V_{AGND}	Voltage on V_{AGND} pin with respect to ground (V_{SS})	V_{SS}	
V_{IO}	Voltage on any pin with respect to ground (V_{SS})	- 0.5 to V_{DD} + 0.5	
I_{OV}	Input current on any pin during overload condition	±10	mA
I_{TOV}	Absolute sum of all input currents during overload condition	| 75 |	
T_{ST}	Storage temperature	- 65 to +150	°C
ESD	ESD susceptibility (human body model)	2000	V

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 datasheet is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During overload conditions ($V_{IN} > V_{DD}$ or $V_{IN} < V_{SS}$) the voltage on pins with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.

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

24.2 Recommended operating conditions

Table 160. Recommended operating conditions

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

1. The value of the V_{STBY} voltage is in the range 4.5 to 5.5 volts. It is acceptable to exceed the upper limit (up to 6.0 volts) for a maximum of 100 hours over 300 000 hours (about 30 years), which represents the lifetime of the device. When V_{STBY} voltage is lower than main V_{DD}, the input section of V_{STBY}/$\overline{EA}$ pin can generate a spurious static consumption on V_{DD} power supply (in the range of a tenth of a µA).
2. For details on operating conditions concerning the use of the ADC, refer to *Section 24.7: ADC characteristics*.

24.3 Power considerations

The average chip-junction temperature, T_J, in degrees Celsius, may be calculated using the following equation:

Equation 22

$$T_J = T_A + (P_D \times \Theta_{JA})$$

Where:

T_A is the ambient temperature in °C.

Θ_{JA} is the package junction-to-ambient thermal resistance, in °C/W.

P_D is the sum of P_{INT} and $P_{I/O}$ ($P_D = P_{INT} + P_{I/O}$).

P_{INT} is the product of I_{DD} and V_{DD}, expressed in Watts. This is the chip internal power.

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

Usually, $P_{I/O} < P_{INT}$ can be neglected. $P_{I/O}$ may be significant if the device is configured to drive external modules and/or memories continuously.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is given by:

Equation 23

$$P_D = K / (T_J + 273° C)$$

Solving *Equation 22* and *Equation 23* gives *Equation 24*:

Equation 24

$$K = P_D \times (T_A + 273° C) + \Theta_{JA} \times P_D^2$$

Where:

K is a constant for the particular part, which may be determined from *Equation 24* by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J may be obtained by solving *Equation 22* and *Equation 23* iteratively for any value of T_A.

Table 161. Thermal characteristics

Symbol	Description	Value (typical)	Unit
Θ_{JA}	Thermal resistance junction-ambient PBGA 208 package (23 x 23 x 1.96 mm)	30	° C/W

Based on thermal characteristics of the package and with reference to the power consumption values provided in *Table 163: DC characteristics* and *Figure 97: Supply current versus the operating frequency (run and idle modes)*), the product classification in *Table 162* is suggested. The exact power consumption of the device inside the application must be computed according to different working conditions, thermal profiles, real thermal resistance of the system (including the printed circuit board or other substrata), I/O activity, and so on.

Table 162. Package characteristics

Package	Ambient temperature range	CPU frequency range
PBGA 208	-40 to 125 °C	1 to 64 MHz

24.4 Parameter interpretation

The parameters listed in *Table 163: DC characteristics* represent characteristics of the ST10F296E and its demands on the system.

Where the ST10F296E logic provides signals with their respective timing characteristics, the symbol for controller characteristics (CC) is included in the ‘Symbol’ column. Where the external system must provide signals with their respective timing characteristics to the ST10F296E, the symbol for system requirement (SR) is included in the ‘Symbol’ column.

24.5 DC characteristics

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

Table 163. DC characteristics

Symbol	Parameter	Test condition	Limit values		Unit
			Min	Max	
V_{IL} (SR)	Input low voltage (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1, READY)	-	-0.3	0.8	V
V_{ILS} (SR)	Input low voltage (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1, READY)	-	-0.3	0.3 V_{DD}	
V_{IL1} (SR)	Input low voltage $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD	-	-0.3	0.3 V_{DD}	
V_{IL2} (SR)	Input low voltage XTAL1 (CMOS only)	Direct drive mode	-0.3	0.3 V_{DD}	
V_{IL3} (SR)	Input low voltage READY (TTL only)	-	-0.3	0.8	
V_{IH} (SR)	Input high voltage (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1)	-	2.0	V_{DD} + 0.3	
V_{IHS} (SR)	Input high voltage (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD, XTAL1)	-	0.7 V_{DD}	V_{DD} + 0.3	
V_{IH1} (SR)	Input high voltage $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, RPD	-	0.7 V_{DD}	V_{DD} + 0.3	
V_{IH2} (SR)	Input high voltage XTAL1 (CMOS only)	Direct drive mode	0.7 V_{DD}	V_{DD} + 0.3	V
V_{IH3} (SR)	Input high voltage READY (TTL only)	-	2.0	V_{DD} + 0.3	
VHYS (CC)	Input hysteresis (TTL mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, XTAL1, RPD)	(1)	400	700	mV
VHYSS (CC)	Input hysteresis (CMOS mode) (except $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$, XTAL1, RPD)	(1)	750	1400	
VHYS1 (CC)	Input hysteresis $\overline{RSTIN}$, $\overline{EA}$, $\overline{NMI}$	(1)	750	1400	
VHYS2 (CC)	Input hysteresis XTAL1	(1)	0	50	
VHYS3 (CC)	Input hysteresis READY (TTL only)	(1)	400	700	
VHYS4 (CC)	Input hysteresis RPD	(1)	500	1500	

Table 163. DC characteristics (continued)

Symbol	Parameter	Test condition	Limit values		Unit
			Min	Max	
V_{OL} (CC)	Output low voltage (P6[7:0], ALE, $\overline{RD}$, $\overline{WR}$/$\overline{WRL}$, $\overline{BHE}$/$\overline{WRH}$, CLKOUT, $\overline{RSTIN}$, $\overline{RSTOUT}$)	I_{OL} = 8 mA I_{OL} = 1 mA	-	0.4 0.05	V
V_{OL1} (CC)	Output low voltage (P0[15:0], P1[15:0], P2[15:0], P3[15,13:0], P4[7:0], P7[7:0], P8[7:0])	I_{OL1} = 4 mA I_{OL1} = 0.5 mA	-	0.4 0.05	
V_{OL2} (CC)	Output low voltage RPD	I_{OL2} = 85 µA I_{OL2} = 80 µA I_{OL2} = 60 µA	-	V_{DD} 0.5 V_{DD} 0.3 V_{DD}	
V_{OH} (CC)	Output high voltage (P6[7:0], ALE, $\overline{RD}$, $\overline{WR}$/$\overline{WRL}$, $\overline{BHE}$/$\overline{WRH}$, CLKOUT, $\overline{RSTOUT}$)	I_{OH} = -8 mA I_{OH} = -1 mA	V_{DD} -0.8 V_{DD} -0.08	-	
V_{OH1} (CC)	Output high voltage(2) (P0[15:0], P1[15:0], P2[15:0], P3[15,13:0], P4[7:0], P7[7:0], P8[7:0])	I_{OH1} = -4 mA I_{OH1} = -0.5 mA	V_{DD} -0.8 V_{DD} -0.08	-	
V_{OH2} (CC)	Output high voltage RPD	I_{OH2} = -2 mA I_{OH2} = -750 µA I_{OH2} = -150 µA	0 0.3 V_{DD} 0.5 V_{DD}	-	
$\lvert I_{OZ1} \rvert$ (CC)	Input leakage current (P5[15:0])(3)	-	-	±0.2	µA
$\lvert I_{OZ2} \rvert$ (CC)	Input leakage current (all except P5[15:0], P2.0, RPD)	-	-	±0.5	
$\lvert I_{OZ3} \rvert$ (CC)	Input leakage current (P2.0)(4)	-	-	+1.0 -0.5	
$\lvert I_{OZ4} \rvert$ (CC)	Input leakage current (RPD)	-	-	±3.0	
$\lvert I_{OV1} \rvert$ (SR)	Overload current (all except P2.0)	(1)(5)	-	±5	mA
$\lvert I_{OV2} \rvert$ (SR)	Overload current (P2.0)(4)	(1)(5)	-	+5 -1	mA
R_{RST} (CC)	$\overline{RSTIN}$ pull-up resistor	100 kΩ nominal	50	250	kΩ
I_{RWH}	Read/write inactive current(6)(7)	V_{OUT} = 2.4 V	-	-40	µA
I_{RWL}	Read/write active current (6)(8)	V_{OUT} = 0.4 V	-500	-	
I_{ALEL}	ALE inactive current(6)(7)	V_{OUT} = 0.4 V	20	-	
I_{ALEH}	ALE active current(6)(8)	V_{OUT} = 2.4 V	-	300	
I_{P6H}	Port 6 inactive current (P6[4:0])(6)(7)	V_{OUT} = 2.4 V	-	-40	
I_{P6L}	Port 6 active current (P6[4:0])(6)(8)	V_{OUT} = 0.4 V	-500	-	

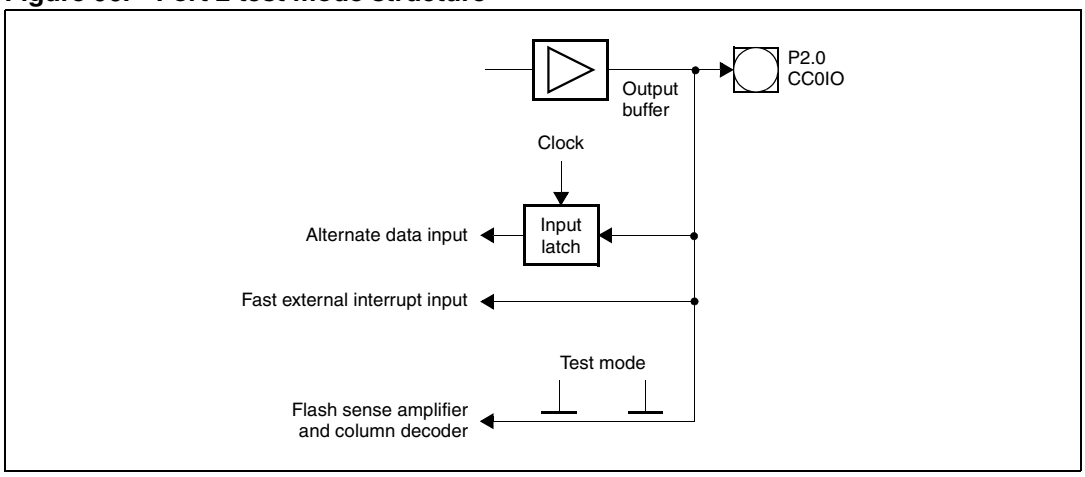
Table 163. DC characteristics (continued)

Symbol	Parameter	Test condition	Limit values		Unit
			Min	Max	
I_{P0H}[7]	Port 0 configuration current[6]	V_{IN} = 2.0 V	-	-10	µA
I_{P0L}[8]		V_{IN} = 0.8V	-100	-	
C_{IO} (CC)	Pin capacitance (digital inputs/outputs)	[1][6]	-	10	pF
I_{CC1}	Run mode power supply current (execution from internal RAM)[9]	-	-	20 + 2 f_{CPU}	mA
I_{CC2}	Run mode power supply current (execution from internal Flash)[1][9]	-	-	20 + 1.8 f_{CPU}	mA
I_{ID}	Idle mode supply current[10]	-	-	20 + 0.6 f_{CPU}	mA
I_{PD1}	Power-down supply current (RTC off, oscillators off, main voltage regulator off)[11]	T_A = 25 °C	-	1	mA
		T_A = 125 °C		3	
I_{PD2}	Power-down supply current[11] (RTC on, main oscillator on, main voltage regulator off)	T_A = 25 °C	-	8	mA
		T_A = 125 °C	-	10	
I_{SB1}	Standby supply current (RTC off, main oscillator off, V_{DD} off, V_{STBY} on)[11]	V_{STBY} = 5.5 V $T_A = T_J$ = 25 °C	-	250	µA
		V_{STBY} = 5.5 V $T_A = T_J$ = 125 °C	-	500	
		V_{STBY} = 5.5 V T_J = 150°C[4]	-	700	
I_{SB3}	Standby supply current (V_{DD} transient condition)[1][11]	-	-	2.5	mA

1. Not 100% tested, guaranteed by design characterization.
2. This specification is not valid for outputs which are switched to open-drain mode. In this case the respective output floats and the voltage is imposed by the external circuitry.
3. Port 5 and XPort 10 leakage values are granted for unselected ADC channels. One channel is always selected (by default, after reset, P5.0 is selected). For the selected channel the leakage value is similar to that of other port pins.
4. The leakage of P2.0 is higher than other pins due to the additional logic (pass gates active only in specific test modes) implemented on its input path. Do not stress P2.0 input pin with negative overload beyond the specified limits as failures in Flash reading may occur (sense amplifier perturbation). Refer to *Figure 96* for a scheme of the input circuitry.
5. Overload conditions occur if the standard operating conditions are exceeded, that is, the voltage on any pin exceeds the specified range ($V_{OV} > V_{DD}$ + 0.3 V or $V_{OV} < -0.3$ V). The absolute sum of input overload currents on all port pins must not exceed 50 mA. The supply voltage must remain within the specified limits.
6. This specification is only valid during reset, or during hold or adapt mode. Port 6 pins are only affected if they are used for $\overline{CS}$ output and the open drain function is not enabled.
7. The maximum current may be drawn while the respective signal line remains inactive.
8. The minimum current must be drawn to drive the respective signal line active.

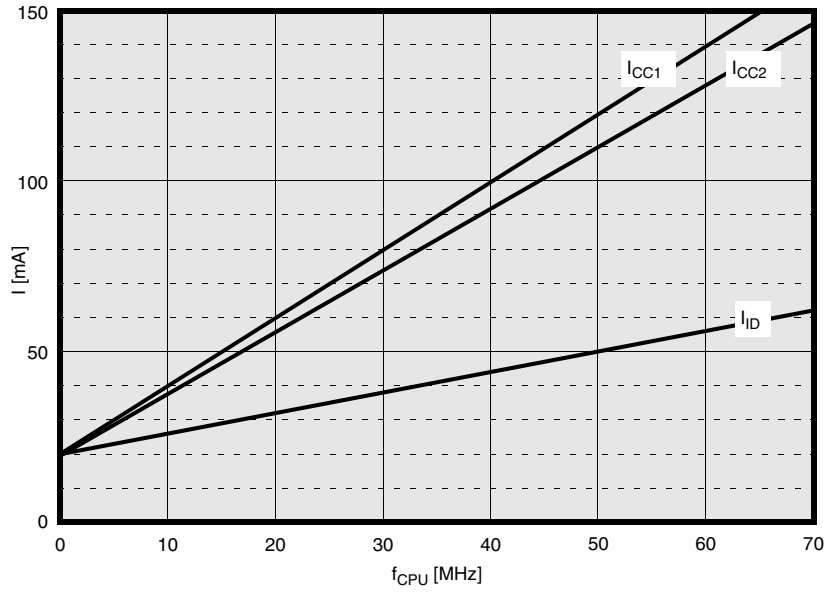
9. The power supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 97* below. This parameter is tested at V_{DDmax} and at maximum CPU clock frequency with all outputs disconnected, all inputs at V_{IL} or V_{IH}, and RSTIN pin at V_{IH1min}: **This implies that I/O current is not considered**. The device does the following:
   - Fetches code from IRAM and XRAM1, read and write accesses both XRAM modules
   - Enables watchdog timer and services it regularly
   - RTC runs with main oscillator clock as reference, generating a tick interrupt every 192 clock cycles
   - Four XPWM channels run (wave periods: 2, 2.5, 3, and 4 CPU clock cycles): No output toggling
   - Five general purpose timers run in timer mode with prescaler equal to 8 (T2, T3, T4, T5, and T6)
   - ADC is in auto scan continuous conversion mode on all 16 channels of Port 5
   - All interrupts generated by XPWM, RTC, timers and ADC are not serviced
10. The idle mode supply current is a function of the operating frequency (f_{CPU} is expressed in MHz). This dependency is illustrated in *Figure 97* below. These parameters are tested at maximum CPU clock with all outputs disconnected, all inputs at V_{IL} or V_{IH}, and the RSTIN pin at V_{IH1min}.
11. Testing of this parameter includes leakage currents. All inputs (including pins configured as inputs) are at 0 to 0.1 V or at V_{DD} - 0.1 V to V_{DD}, V_{AREF} = 0 V, all outputs (including pins configured as outputs) are disconnected. The main voltage regulator is assumed to be off. If this is not the case, an additional 1 mA must be assumed.

Figure 96. Port 2 test mode structure



1. For the complete structure of Port 2, see *Figure 37* in *Section 13.4*

Figure 97. Supply current versus the operating frequency (run and idle modes)



24.6 Flash characteristics

V_{DD} = 5 V ± 10%, V_{SS} = 0 V

Table 164. Flash characteristics

Parameter	Typical T_A = 25 °C	Maximum T_A = 125 °C		Unit	Notes
	0 cycles(1)	0 cycles(1)	100 k cycles(2)		
Word program (32-bit)(3)	35	80	290	µs	-
Double word program (64-bit)(3)	60	150	570	µs	-
Maximum word program (32-bit)	-	560	1385	µs	
Maximum double word program (64-bit)	-	1160	2760	µs	-
Bank 0 program (384 Kbyte) (double word program)	2.9	7.4	28.0	s	-
Bank 1 program (128 Kbyte) (double word program)	1.0	2.5	9.3	s	-
Bank 2 program (192 Kbyte) (double word program)	1.5	3.7	14.0	s	-
Bank 3 program (128 Kbyte) (double word program)	1.0	2.5	9.3	s	-
Sector erase (8 Kbyte)	0.6 0.5	0.9 0.8	1.0 0.9	s	Not preprogrammed Preprogrammed
Sector erase (32 Kbyte)	1.1 0.8	2.0 1.8	2.7 2.5	s	Not preprogrammed Preprogrammed
Sector erase (64K)	1.7 1.3	3.7 3.3	5.1 4.7	s	Not preprogrammed Preprogrammed
Bank 0 erase (384 Kbyte)(4)	8.2 5.8	20.2 17.7	28.6 26.1	s	Not preprogrammed Preprogrammed
Bank 1 erase (128 Kbyte)(4)	3.0 2.2	7.0 6.2	9.8 9.0	s	Not preprogrammed Preprogrammed
Bank 2 erase (192 Kbyte)(4)	4.3 3.1	10.3 9.1	14.5 13.3	s	Not preprogrammed Preprogrammed
Bank 3 erase (128 Kbyte)(4)	3.0 2.2	7.0 6.2	9.8 9.0	s	Not preprogrammed Preprogrammed
Imodule erase (512 Kbyte)(5)	11.2 7.6	27.2 23.5	38.4 34.7	s	Not preprogrammed Preprogrammed
Xmodule erase (320 Kbyte)(5)	7.3 4.9	17.3 14.8	24.3 21.8	s	Not preprogrammed Preprogrammed
Chip erase (832 Kbyte)(6)	18.5 12.0	44.4 37.9	62.6 56.1	s	Not preprogrammed Preprogrammed
Recovery from power-down (t_{PD})	–	40	40	µs	(7)
Program suspend latency(7)	-	10	10	µs	

Table 164. Flash characteristics (continued)

Parameter	Typical T_A = 25 °C	Maximum T_A = 125 °C		Unit	Notes
	0 cycles(1)	0 cycles(1)	100 k cycles(2)		
Erase suspend latency(7)	-	30	30	µs	
Erase suspend request rate(7)	20	20	20	ms	Min delay between two requests
Set protection(7)	40	170	170	µs	

1. Values are after about 100 cycles due to testing routines (0 cycles for the final customer).
2. The maximum program and erase times occur after the specified number of program/erase cycles. These maximum values are characterized but not guaranteed.
3. Word and double word programming times are provided as average values derived from a full sector programming time. The absolute value of a word or double word programming time may be longer than the provided average value.
4. Bank erase is obtained through a multiple sector erase operation (setting bits related to all sectors of the bank).
5. Module erase is obtained through a sequence of two bank erase operations (since each module is composed of two banks).
6. Chip erase is obtained through a sequence of two module erase operations on the Imodule and Xmodule.
7. Not 100% tested, guaranteed by design characterization.

Table 165. Data retention characteristics

Number of program/erase cycles (-40 °C ≤T_A ≤125 °C)	Data retention time (average ambient temperature 60 °C)	
	832 Kbyte (code store)	64 Kbyte (EEPROM emulation)(1)
0 - 100	> 20 years	> 20 years
1,000	-	> 20 years
10,000	-	10 years
100,000	-	1 year

1. Two 64 Kbyte Flash sectors may be typically used to emulate up to 4, 8, or 16 Kbytes of EEPROM. Therefore, in case of an emulation of a 16 Kbyte EEPROM, 100 000 Flash program/erase cycles are equivalent to 800 000 EEPROM program/erase cycles. For an efficient use of the read while write feature and/or EEPROM emulation, please refer to the dedicated application note (AN2061, EEPROM Emulation with ST10F2xx) on *www.st.com*.

24.7 ADC characteristics

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

Table 166. ADC characteristics

Symbol	Parameter	Test condition	Limit values		Unit
			Min	Max	
V_{AREF} (SR)	Analog reference voltage(1)		4.5	V_{DD}	V
V_{AGND} (SR)	Analog ground voltage		V_{SS}	V_{SS} + 0.2	V
V_{AIN} (SR)	Analog input voltage(2)		V_{AGND}	V_{AREF}	V
I_{AREF} (CC)	Reference supply current	Running mode(3) Power-down mode	- -	5 1	mA µA
t_S (CC)	Sample time	(4)	1	-	µs
t_C (CC)	Conversion time	(5)	3	-	µs
DNL (CC)	Differential nonlinearity(6)	No overload	-1	1	LSB
INL (CC)	Integral nonlinearity (6)	No overload	-1.5	1.5	LSB
OFS (CC)	Offset error (6)	No overload	-1.5	1.5	LSB
TUE (CC)	Total unadjusted error(6)		-2.0	2.0	LSB
K (CC)	Coupling factor between inputs(3)(7)	On both Port 5 and XPort 10	-	10^{-6}	-
C_{P1} (CC)	Input pin capacitance(3)(8)		-	3	pF
C_{P2} (CC)		Port 5 XPort 10	-	5 5	pF pF
C_S (CC)	Sampling capacitance(3)(8)		-	3.5	pF
R_{SW} (CC)	Analog switch resistance (3)(8)	Port 5 XPort 10	- -	600 1600	Ω Ω
R_{AD} (CC)			-	1300	Ω

1. V_{AREF} can be tied to ground when ADC is not in use: An extra consumption (around 200 µA) on main V_{DD} is added because the internal analog circuitry is not completely turned off. It is suggested to maintain the V_{AREF} at V_{DD} level even when not in use, and to eventually switch off the ADC circuitry setting bit, ADOFF, in the ADCON register.
2. V_{AIN} may exceed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases is $0x000_H$ or $0x3FF_H$, respectively.
3. Not 100% tested, guaranteed by design characterization.
4. During the sample time, t_S, the input capacitance, C_{AIN}, can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach its final voltage level within t_S. After the end of the sample time, changes of the analog input voltage have no effect on the conversion result. Values for the sample clock, t_S, depend on programming and can be taken from *Table 167*.
5. This parameter includes the sample time, t_S, the time for determining the digital result, and the time to load the result register with the conversion result. Values for the conversion clock, t_{CC}, depend on programming and can be taken from *Table 167*.

6. DNL, INL, OFS and TUE are tested at V_{AREF} = 5.0 V, V_{AGND} = 0 V, V_{DD} = 5.0 V. They are guaranteed by design characterization for all other voltages within the defined voltage range. 'LSB' has a value of V_{AREF}/1024. The specified TUE (±2 LSB) is also guaranteed with an overload condition (see I_{OV} specification) occurring on a maximum of two unselected analog input pins if the absolute sum of input overload currents on all analog input pins does not exceed 10 mA.
7. The coupling factor is measured on a channel while the overload condition occurs on the adjacent unselected channels with the overload current within the different specified ranges (for both positive and negative injection current).
8. Refer to *Figure 99*

24.7.1 Conversion timing control

When a conversion starts, the capacitances of the converter are first 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. Next, the sampled voltage is converted into a digital value in several successive steps which corresponds to the 10-bit resolution of the ADC. During these steps the internal capacitances are repeatedly charged and discharged via the V_{AREF} pin.

The current that must be drawn from the sources for sampling and changing charges depends on the duration of each step because the capacitors must reach their final voltage level as close to the given time as possible. However, the maximum current that a source can deliver depends on its internal resistance.

The amount of time that sampling and converting takes during conversion can be programmed within a certain range in the ST10F296E relative to the CPU clock. The absolute time consumed by the different conversion steps is therefore independent from the general speed of the controller. This allows the device ADC to be adjusted to the properties of the system.

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

High internal resistance can be achieved by programming the respective times to a higher value or to their possible maximum. This is preferable when using analog sources and supplies with a high internal resistance 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 bits of the ADCON register. Bit fields ADCTC and ADSTC define the basic conversion time and in particular the partition between the sample phase and comparison phases. *Table 167* lists the possible combinations. The timings refer to the unit TCL, where f_{CPU} = 1/2 TCL. A complete conversion time includes the conversion itself, the sample time and the time required to transfer the digital value to the result register.

Table 167. ADC programming

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

Note: The total conversion time is compatible with the formula valid for the ST10F280, while the meaning of the bit fields ADCTC and ADSTC is no longer compatible: The minimum conversion time is 388 TCL, which at 40 MHz CPU frequency corresponds to 4.85µsµ (see ST10F280).

24.7.2 ADC conversion accuracy

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

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

These errors are explained below using *Figure 98*.

Offset error

Offset error is the deviation between actual and ideal AD conversion characteristics when the digital output value changes from the minimum zero voltage, 00, to 01 (see OFS in *Figure 98*).

Gain error

Gain error is the deviation between the actual and ideal AD conversion characteristics when the digital output value changes from 3FE to 3FF, after subtracting offset error. Gain error combined with offset error represents full-scale error (see OFS + GE in *Figure 98*).

Quantization error

Quantization error is the intrinsic error of the ADC and is expressed as 1/2 LSB.

Nonlinearity error

Nonlinearity error is the deviation between the actual and the best-fitting AD conversion characteristics (see *Figure 98*):

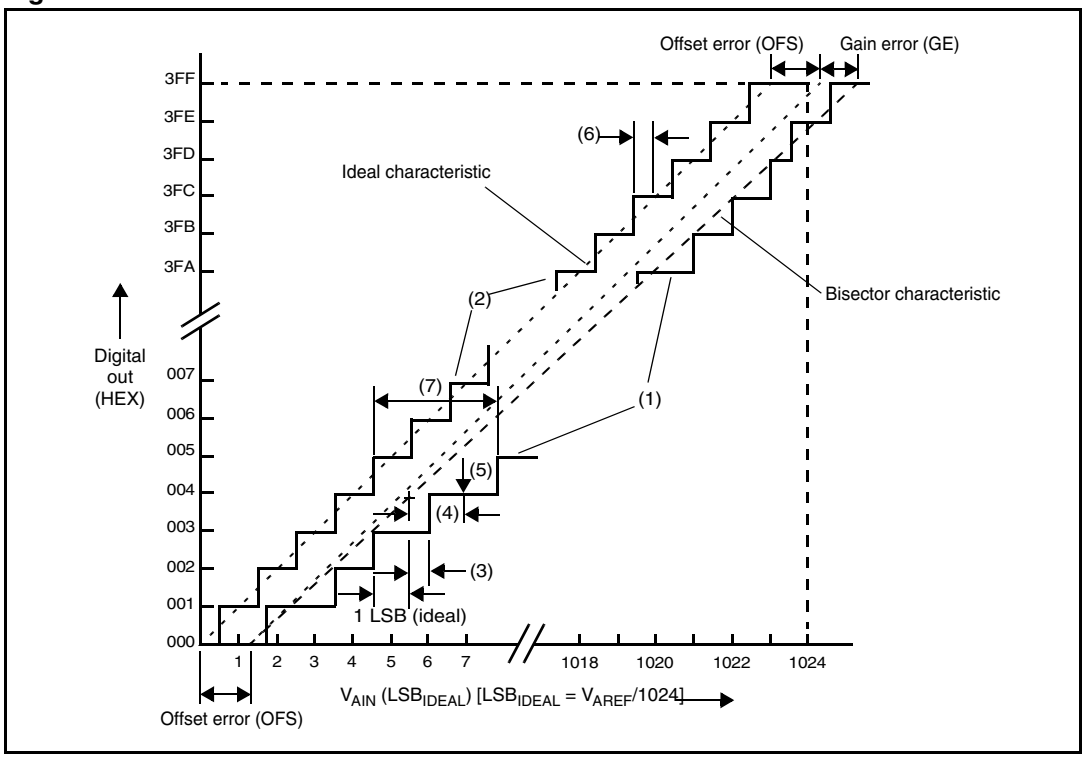
- Differential nonlinearity error is the actual step dimension versus the ideal one (1 LSB_{IDEAL}).
- Integral nonlinearity error is the distance between the center of the actual step and the center of the bisector line, in the actual characteristics. Note that for integral nonlinearity error, the effects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

Note: The bisector characteristic is obtained by drawing a line from 1/2 LSB to a point before the first step of the real characteristic, and another line from 1/2 LSB to a point after the last step of the real characteristic (see Figure 98).

Total unadjusted error

The total unadjusted error (TUE) specifies the maximum deviation from the ideal characteristic. The value provided in this datasheet represents the maximum error with respect to the entire characteristic. It is a combination of the offset, gain and integral linearity errors. The different errors may compensate each other depending on the relative sign of the offset and gain errors (see TUE in *Figure 98*).

Figure 98. AD conversion characteristic



1. Legend:
   (1) Example of an actual transfer curve
   (2) The ideal transfer curve
   (3) Differential Nonlinearity Error (DNL)
   (4) Integral Nonlinearity Error (INL)
   (5) Center of a step of the actual transfer curve
   (6) Quantization Error (1/2 LSB)
   (7) Total Unadjusted Error (TUE)

24.7.3 Analog reference pins

The accuracy of the ADC converter depends on the accuracy of its analog reference. A noise in the reference results in the same proportion of error in a conversion. A low pass filter on the ADC converter reference source (supplied through the V_{AREF} and V_{AGND} pins), is recommended to clean the signal thereby minimizing the noise. A simple capacitive bypassing may be sufficient in most cases. In the presence of high RF noise energy, inductors or ferrite beads may be necessary.

In the ST10F296E architecture, the V_{AREF} and V_{AGND} pins also represent the power supply of the analog circuitry of the ADC. An effective DC current is required from the reference voltage to the internal resistor string in the R-C DAC array and to the rest of the analog circuitry.

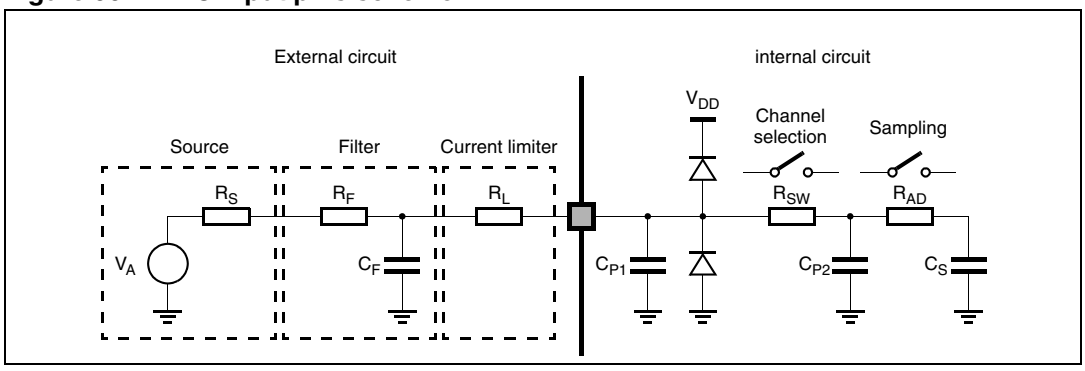
An external resistance on V_{AREF} could introduce error under certain conditions. For this reason, series resistance is not advisable. Any series devices in the filter network should be designed to minimize the DC resistance.

24.7.4 Analog input pins

To improve the accuracy of the ADC, analog input pins must have low AC impedance. Placing a capacitor with good high frequency characteristics at the input pin of the device can be effective. The capacitor should be as large as possible, ideally infinite. This capacitor contributes to attenuating the noise present on the input pin. Moreover, the source of the capacitor charges during the sampling phase, when the analog signal source is a high-impedance source.

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

Figure 99. ADC input pins scheme



1. Legend:
   R_S: Source impedance
   R_F: Filter resistance
   C_F: Filter capacitance
   R_L: Current limiter resistance
   R_{SW}: Channel selection switch impedance
   R_{AD}: Sampling switch impedance
   c_p: Pin capacitance (two contributions, CP1 and CP2)
   C_S: Sampling capacitance
   V_A: Source voltage

Input leakage and external circuit

The series resistor used to limit the current to a pin (see R_L in *Figure 99*), in combination with a large source of impedance, can lead to a degradation of the ADC accuracy when input leakage is present.

Data about maximum input leakage current at each pin is provided in *Section 24.5: DC characteristics*. Input leakage is greatest at high operating temperatures and generally decreases by one half a degree for each 10 °C decrease in temperature.

Considering that one count of a 10-bit ADC is about 5 mV (assuming V_{AREF} = 5 V), an input leakage of 100 nA acting though an R_L = 50 kΩ of external resistance, leads to an error of exactly one count (5 mV). If the resistance is 100 kΩ, the error is two counts (10 mV).

Additional leakage due to external clamping diodes must also be taken into account in computing the total leakage affecting the ADC measurements. Another contribution to the total leakage is represented by the charge sharing effects with the sampling capacitance. The sampling capacitance, C_S, is essentially a switched capacitance with a frequency equal to the conversion rate of a single channel (maximum when the fixed channel continuous conversion mode is selected). It can be seen as a resistive path to ground. For instance, assuming a conversion rate of 250 kHz and a C_S of 4 pF, a resistance of 1 MΩ is obtained ($R_{EQ} = 1/f_C C_S$, where f_C represents the conversion rate at the considered channel). To minimize the error induced by the voltage partitioning between this resistance (sampled voltage on C_S) and the sum of $R_S + R_F + R_L + R_{SW} + R_{AD}$, the external circuit must be designed to respect the following relation:

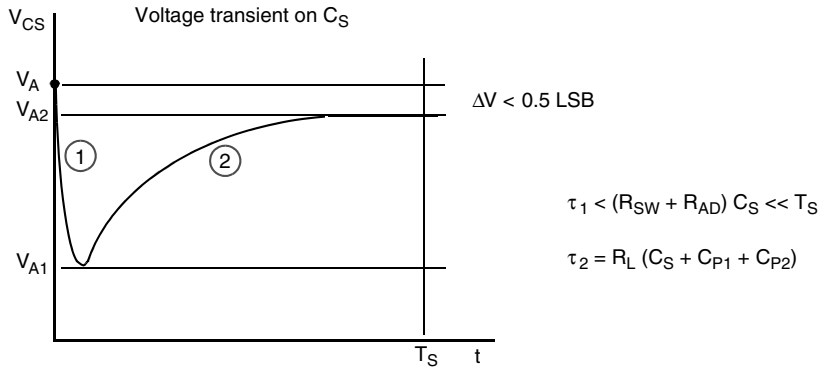
Equation 25

$$V_A \times (R_S + R_F + R_L + R_{SW} + R_{AD}) / R_{EQ} < (1 / 2) LSB$$

Equation 25 places constraints on the external network design, in particular on the resistive path.

A second aspect of the capacitance network must be considered. Assuming the three capacitances, C_F, C_{P1} and C_{P2}, are initially charged at the source voltage V_A (see *Figure 99*), when the sampling phase is started (ADC switch closed), a charge-sharing phenomena begins (see *Figure 100*).

Figure 100. Charge sharing timing diagram during sampling phase



Two different transient periods can be distinguished in *Figure 100*. They are described below.

First transient period

This is a quick charge transfer from the internal capacitances, C_{P1} and C_{P2}, to the sampling capacitance, C_S (initially C_S is supposed to be completely discharged). Considering the worst case scenario (since the time constant in reality is faster) in which C_{P2} is in parallel to C_{P1} (call $C_P = C_{P1} + C_{P2}$), the two capacitances, C_P and C_S, are in series and the time constant is:

Equation 26

$$\tau_1 = (R_{SW} + R_{AD}) \times (C_P \times C_S / C_P + C_S)$$

Equation 26 can be simplified if only C_S is considered as an additional worst condition. In reality, the transient is faster, but the ADC circuitry has been designed to be robust in the worst case situations. The sampling time, T_S, is always much longer than the internal time constant as in *Equation 27*.

Equation 27

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

The charge of C_{P1} and C_{P2} is also redistributed on C_S, which determines a new value for the V_{A1} voltage on the capacitance according to *Equation 28*.

Equation 28

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

Second transient period

A second charge transfer also involves C_F (that is typically greater than the on-chip capacitance) through the resistance R_L. Consideringagain the worst case scenario in which C_{P2} and C_S are in parallel to C_{P1} (since the time constant in reality is faster), the time constant is:

Equation 29

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

In *Equation 29*, the time constant depends on the external circuit. In particular, if the transient is completed well before the end of the sampling time, T_S, a constraint on R_L sizing is obtained, as shown in *Equation 30*.

Equation 30

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

R_L must also be sized, according to the current limitation constraints, in combination with R_S (source impedance) and R_F (filter resistance). As C_F is greater than C_{P1}, C_{P2} and C_S, the final voltage, V_{A2} (at the end of the charge transfer transient), is much higher than V_{A1}. *Equation 31* (the charge balance) must be respected assuming that C_S is already charged at V_{A1}.

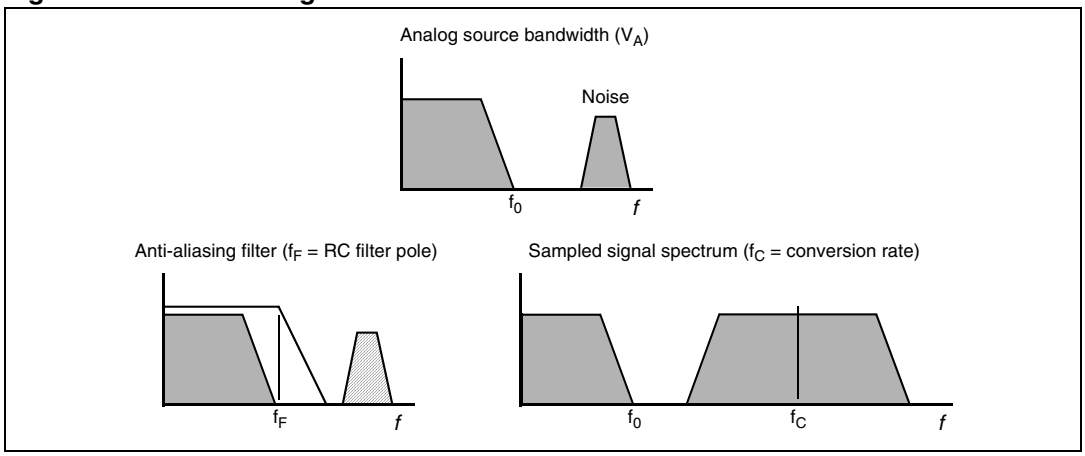
Equation 31

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

Transient periods one and two are not influenced by the voltage source that cannot provide the extra charge to compensate for the voltage drop on C_S with respect to the ideal source V_A (due to the presence of the R_FC_F filter). The time constant R_FC_F of the filter is very high with respect to the sampling time (T_S). The filter is typically designed to be anti-aliasing (see *Figure 101*).

If f_0 is the bandwidth of the source signal (and consequently is also the cut-off frequency of the anti-aliasing filter, f_F), then according to Nyquist's theorem, the conversion rate, f_C, must be at least $2f_0$. This means that the constant time of the filter is greater than or equal to twice the conversion period (T_C). The conversion period, T_C, is longer than the sampling time, T_S, even when fixed channel continuous conversion mode is selected (the fastest conversion rate at a specific channel). In conclusion, the time constant of the filter R_FC_F is much higher than the sampling time, T_S, so the charge level on C_S cannot be modified by the analog signal source during the time in which the sampling switch is closed.

Figure 101. Anti-aliasing filter and conversion rate



1. TC ≤2 R_FC_F (conversion rate vs. filter pole).
2. $f_F = f_0$ (anti-aliasing filtering condition).
3. 2 f0 ≤f_C (Nyquist's theorem).

The considerations above impose new constraints on the external circuit. Accuracy error, due to the voltage drop on C_S, must be reduced. Based on *Equation 30* and *Equation 31* above, *Equation 32* is derived to explain the relationship between the ideal and real sampled voltage on C_S.

Equation 32

$$V_A / V_{A2} = (C_{P1} + C_{P2} + C_F) / (C_{P1} + C_{P2} + C_F + C_S)$$

In the worst case scenario (V_A = 5 V), *Equation 32* assumes a maximum error of half a count (~2.44 mV) which leads to a constraint on the C_F value as shown in *Equation 33*.

Equation 33

$$C_F > 2048 \times C_S$$

24.7.5 Example of external network sizing

This section provides an example of how to design an external network, based on realistic values for the internal parameters and on a hypothesis concerning the characteristics of the analog signal to be sampled.

The following hypothesis is formulated to design the external network on the ADC input pins:

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

If designing a filter with the pole at the maximum frequency of the signal, the time constant of the filter is given in *Equation 34*:

Equation 34

$$R_C C_F = 1/(2\pi f_0) = 15.9\mu s$$

Using the relationship between C_F and C_S (*Equation 33*) and taking some margin (4000 instead of 2048), it is possible to define C_F as shown in *Equation 35*.

Equation 35

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

Equation 34 and *Equation 35* allow the RC to be calculated as shown in *Equation 36*.

Equation 36

$$R_F = 1/(2\pi f_0 C_F) = 995\Omega \cong 1k\Omega$$

Total series resistance can be calculated using *Equation 37* where the current injection limitation is considered and it is assumed that the source can go up to 12 V.

Equation 37

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

Equation 37 allows a value for R_L to be defined as shown in *Equation 38*.

Equation 38

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

Equation 36, and *Equation 38* define respectively the three elements of an external circuit, R_F, C_F, and R_L. Next, some conditions which are used to size the circuit must be verified. The first of these is a calculation which allows the accuracy error, introduced by the switched capacitance equivalent resistance, to be minimized. This is given in *Equation 39*.

Equation 39

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

The error due to the voltage partitioning between the real resistive path and C_S is less then half a count if considering the worst case when V_A = 5 V (see *Equation 40*).

Equation 40

$$V_A \times (R_S + R_F + R_L + R_{SW} + R_{AD}) / R_{EQ} = 2.35mV < (1 / 2)LSB$$

The other conditions to verify are if the time constants of the transients are shorter than the sampling period duration, T_S, and whether the distance is significant. These calculations are given in *Equation 41* and *Equation 42*.

Equation 41

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

Equation 42

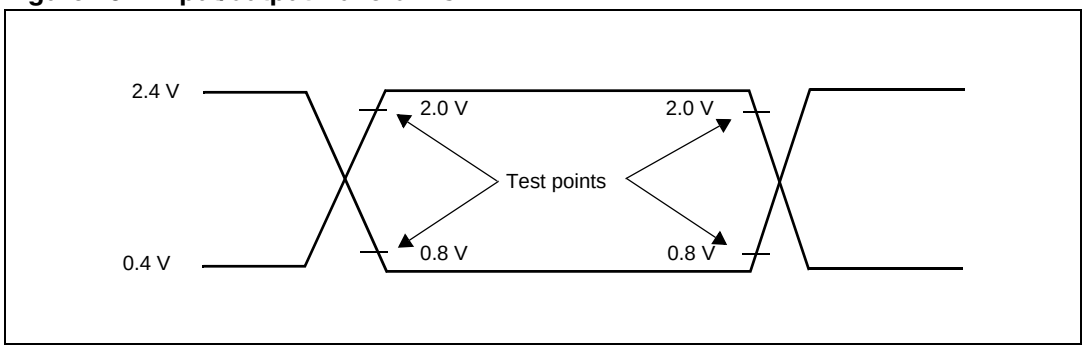
$$10 \times \tau_2 = 10 \times R_L \times (C_S + C_{P1} + C_{P2}) = 290ns < T_S = 1\mu s$$

For a complete set of parameter characterization of the ST10F296E ADC equivalent circuit, refer to *Table 166: ADC characteristics on page 302*.

24.8 AC characteristics

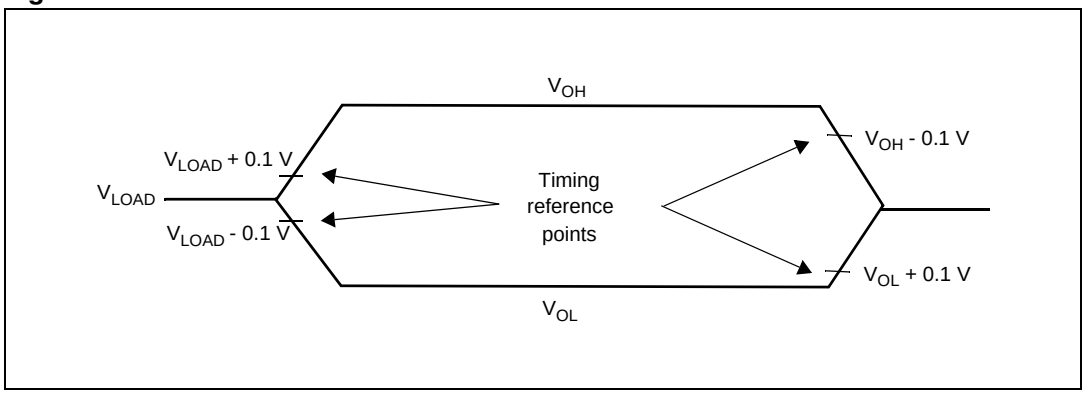
24.8.1 Test waveforms

Figure 102. Input/output waveforms



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

Figure 103. Float waveforms



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

24.8.2 Definition of internal timing

The internal operation of the ST10F296E is controlled by the internal CPU clock f_{CPU}. Both edges of the CPU clock can trigger internal (for example, pipeline) or external (for example, bus cycle) operations.

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

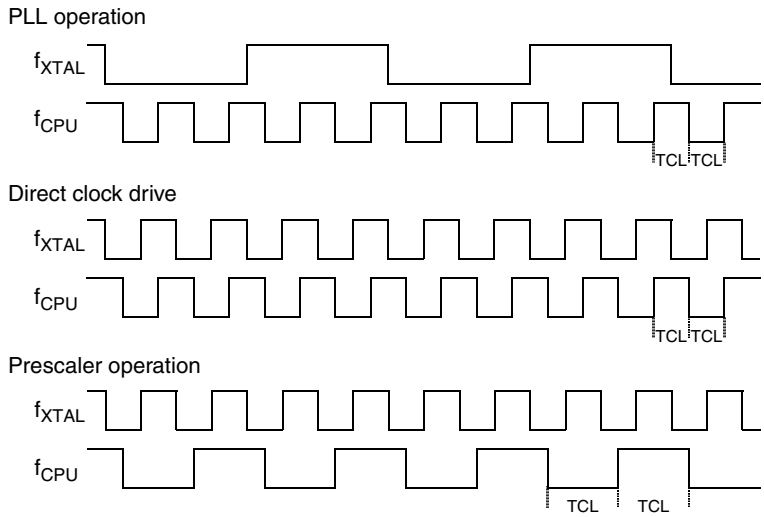
The CPU clock signal can be generated by different mechanisms. The duration of TCL and its variation (and also the derived external timing) depends on the mechanism used to generate f_{CPU}.

The f_{CPU} influence must be regarded when calculating the timings for the ST10F296E.

The example for PLL operation shown in *Figure 104* refers to a PLL factor of four.

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 104. Generation mechanisms for the CPU clock



24.8.3 Clock generation modes

Figure 168 associates combinations of the P0.15-13 (P0H.7-5) bits with the respective clock generation mode.

Table 168. On-chip clock generator selections

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

1. The maximum frequency of the external clock depends on the duty cycle of the external clock signal. When 64 MHz is used, 50 % duty cycle is granted (low phase = high phase = 7.8 ns). When 32 MHz is selected, a 25 % duty cycle can be accepted (minimum, high or low phase = 7.8 ns).

The external clock input range refers to a CPU clock range of 1 to 64 MHz. In addition, PLL use is limited to 4-12 MHz input frequency range. All configurations need a crystal (or ceramic resonator) to generate the CPU clock through the internal oscillator amplifier (apart from direct drive). On the contrary, the clock can be forced through an external clock source only in direct drive mode (on-chip oscillator amplifier disabled, so no crystal or resonator can be used).

The limits on input frequency are 4-12 MHz since use of the internal oscillator amplifier is required. When the PLL is not used and the CPU clock corresponds to $F_{XTAL}/2$, an external crystal or resonator must be used. It is not possible to force any clock though an external clock source.

24.8.4 Prescaler operation

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

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

The timings listed in this section that refer to TCL can be calculated using the f_{XTAL} period for any TCL.

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

24.8.5 Direct drive

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

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

Therefore, the timings given in this section refer to the minimum TCL. This minimum value can be calculated using *Equation 43*.

Equation 43

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

Where DC = 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}, is 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 *Equation 44*.

Equation 44

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

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

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

24.8.6 Oscillator watchdog (OWD)

An on-chip watchdog oscillator is implemented in the ST10F296E. This feature is used for safety reasons with an external crystal oscillator (available only when using direct drive mode with or without prescaler, so the PLL is not used to generate the CPU clock multiplying the frequency of the external crystal oscillator). The watchdog oscillator operates as follows:

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

When the OWD is enabled, the PLL runs at its free-running frequency and it increments the watchdog counter. At each transition of the external clock, the watchdog counter is cleared. If an external clock failure occurs, the watchdog counter overflows (after 16 PLL clock cycles).

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

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

24.8.7 Phase-locked loop (PLL)

For all combinations of pins P0.15-13 (P0H.7-5) other than 011, during reset, the on-chip PLL is enabled and it provides the CPU clock (see *Table 168*). 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 synchronization with the input clock, the frequency of f_{CPU} is constantly adjusted so it is locked to f_{XTAL}. The resulting slight variation causes a jitter of f_{CPU} which also effects the duration of individual TCLs.

The timings listed in this section that refer to TCLs must be calculated using the minimum possible TCL under the respective circumstances.

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

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

24.8.8 Voltage controlled oscillator

The ST10F296E implements a PLL which combines different levels of frequency dividers with a voltage controlled oscillator (VCO) working as a frequency multiplier. *Table 169* presents a summary of the internal PLL settings and VCO frequencies.

Table 169. Internal PLL divider mechanism

P0.15-13 (P0H.7-5)			XTAL frequency	Input prescaler	PLL		Output prescaler	CPU frequency $f_{CPU} = f_{XTAL} \times F$
					Multiply by	Divide by		
1	1	1	4 to 8 MHz	$F_{XTAL}/4$	64	4	-	$F_{XTAL} \times 4$
1	1	0	5.3 to 10.6 MHz	$F_{XTAL}/4$	48	4	-	$F_{XTAL} \times 3$
1	0	1	4 to 8 MHz	$F_{XTAL}/4$	64	2	-	$F_{XTAL} \times 8$
1	0	0	6.4 to 12 MHz	$F_{XTAL}/4$	40	2	-	$F_{XTAL} \times 5$
0	1	1	1 to 64 MHz	–	PLL bypassed		-	$F_{XTAL} \times 1$
0	1	0	4 to 6.4 MHz	$F_{XTAL}/2$	40	2	-	$F_{XTAL} \times 10$
0	0	1	4 to 12 MHz	–	PLL bypassed		$F_{PLL}/2$	$F_{XTAL}/2$
0	0	0	4 MHz	$F_{XTAL}/2$	64	2	–	$F_{XTAL} \times 16$

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

Example 1

- F_{XTAL} = 4 MHz
- P0(15:13) = 110 (multiplication by 3)
- PLL input frequency = 1 MHz
- VCO frequency = 48 MHz => Not valid
- PLL output frequency = Not Valid
- F_{CPU} = Not Valid

Example 2

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

24.8.9 PLL jitter

Two kinds of PLL jitter are defined:

Self referred single period jitter

Also called 'period jitter'. It can be defined as the difference between the T_{max} and T_{min}, where T_{max} is the maximum time period of the PLL output clock and T_{min} is the minimum time period of the PLL output clock.

Self referred long term jitter

Also called 'N period jitter'. It can be defined as the difference of T_{max} and T_{min}, where T_{max} is the maximum time difference between N + 1 clock rising edges and T_{min} is the minimum time difference between N + 1 clock rising edges. N should be kept sufficiently large to have obtain long term jitter. N = 1 becomes the single period jitter.

Jitter at the PLL output is caused by:

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

24.8.10 Jitter in the input clock

The PLL acts as a low pass filter for any jitter in the input clock. Input clock jitter, with the frequencies within the PLL loop bandwidth, is passed to the PLL output and higher frequency jitter (frequency > PLL bandwidth) is attenuated at 20 dB/decade.

24.8.11 Noise in the PLL loop

Noise is attributed to the following sources:

- Device noise of the circuit in the PLL
- Noise in the supply and substrate

Device noise of the circuit in the PLL

Long term jitter is inversely proportional to the bandwidth of the PLL. The wider the loop bandwidth, the lower the jitter, due to noise in the loop. Moreover, long term jitter is practically independent of the multiplication factor.

The most noise sensitive circuit in the PLL is the VCO. There are two main sources of noise: Thermal (random and frequency independent noise) and flicker (low frequency noise, 1/f). For the frequency characteristics of the VCO circuitry, the effect of the thermal noise results in a $1/f^2$ region in the output noise spectrum, while the flicker noise results in $1/f^3$. Assuming a noiseless PLL input and supposing that the VCO is dominated by its $1/f^2$ noise, the root mean square value of the accumulated jitter is proportional to the square root of N, where N is the number of clock periods within the considered time interval.

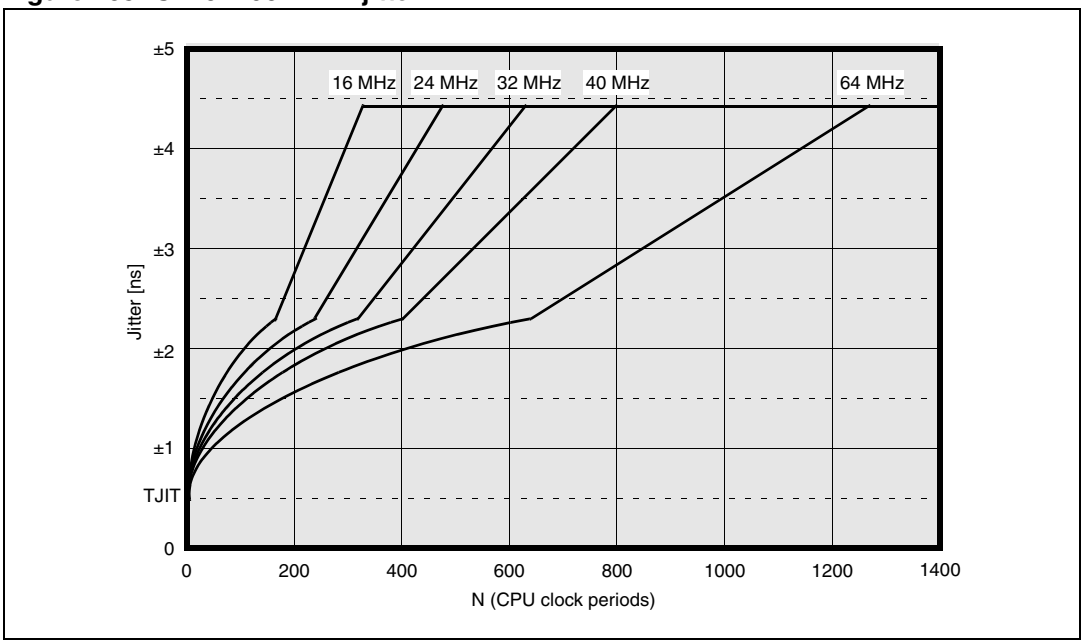
On the contrary, assuming a noiseless PLL input and supposing that the VCO is dominated by its $1/f^3$ noise, the RMS value of the accumulated jitter is proportional to N, where N is the number of clock periods within the considered time interval.

The jitter in the PLL loop can be modeled as being dominated by the $i1/f^2$ noise for N smaller than a 'certain' value that depends on the PLL output frequency and on the bandwidth characteristics of the program loop. Above this 'certain' value, the jitter becomes dominated by the $i1/f^3$ noise component. For N greater than a second value of N, the jitter does not increase with a longer time interval due to an apparent saturation effect (the jitter is stable, thereby increasing the number of clock periods, N). The PLL loop acts as a high pass filter for any noise in the loop, with a cutoff frequency equal to the bandwidth of the PLL. The saturation value corresponds to self referred long term jitter of the PLL. *Figure 105* shows the maximum jitter trend versus the number of clock periods N (for some typical CPU frequencies). The curves represent the worst case situations, as they are computed taking into account all temperature ranges, power supplies and process variations. 'Real' jitter is always measured well below the given worst case value.

Noise in supply and substrate

Digital supply noise adds determining elements to PLL output jitter, independent of the multiplication factor. Its effect is strongly reduced thanks to the particular care taken when integrating and implementing the PLL module inside the device. In addition, the contribution of digital noise to global jitter is widely taken into account in the curves provided in *Figure 105*.

Figure 105. ST10F296E PLL jitter



24.8.12 PLL lock/unlock

If the PLL is unlocked for any reason during normal operation, an interrupt request to the CPU is generated and the reference clock (oscillator) is automatically disconnected from the PLL input. In this way, the PLL goes into free-running mode, providing the system with a backup clock signal (free running frequency F_{free}). This feature allows the device to recover from a crystal failure occurrence without risking entering an undefined configuration. The system is provided with a clock allowing the execution of the PLL unlock interrupt routine in a safe mode.

The path between the reference clock and PLL input can be restored only by a hardware reset or by a bidirectional software or watchdog reset event that forces the $\overline{\text{RSTIN}}$ pin low.

Note: The external RC circuit on the $\overline{RSTIN}$ pin must be the right size to extend the duration of the low pulse that locks the PLL before the level at the $\overline{RSTIN}$ pin is recognized as being high. A bidirectional reset internally drives the $\overline{RSTIN}$ pin low for 1024 TCL (which is not sufficient to lock the PLL when starting from free-running mode).

Conditions: V_{DD} = 5 V ±10 %, T_A = -40/125 °C.

Table 170. PLL lock/unlock timing

Symbol	Parameter	Conditions	Value		Unit
			Min	Max	
T_{PSUP}	PLL start-up time(1)	Stable V_{DD} and reference clock	-	300	µs
T_{LOCK}	PLL lock-in time	Stable V_{DD} and reference clock, starting from free-running mode	-	250	
T_{JIT}	Single period jitter(1) (cycle to cycle = 2 TCL)	6 sigma time period variation (peak to peak)	-500	+500	ps
F_{free}	PLL free running frequency	Multiplication factors: 3, 4 Multiplication factors: 5, 8, 10, 16	250 500	2000 4000	kHz

1. Not 100% tested, guaranteed by design characterization.

24.8.13 Main oscillator specifications

Conditions: V_{DD} = 5 V ±10 %, T_A = -40/125 °C

Table 171. Main oscillator specifications

Symbol	Parameter	Conditions	Value			Unit
			Min	Typ	Max	
g_m	Oscillator transconductance		8	17	35	mA/V
V_{OSC}	Oscillation amplitude(1)	Peak to peak	-	V_{DD} -0.4	-	V
V_{AV}	Oscillation voltage level(1)	Sine wave middle	-	V_{DD}/2 - 0.25	-	
t_{STUP}	Oscillator start-up time(1)	Stable V_{DD} - crystal	-	3	4	ms
		Stable V_{DD}, resonator	-	2	3	

1. Not 100% tested, guaranteed by design characterization

Figure 106. ST10F296ECrystal oscillator and resonator connection diagram

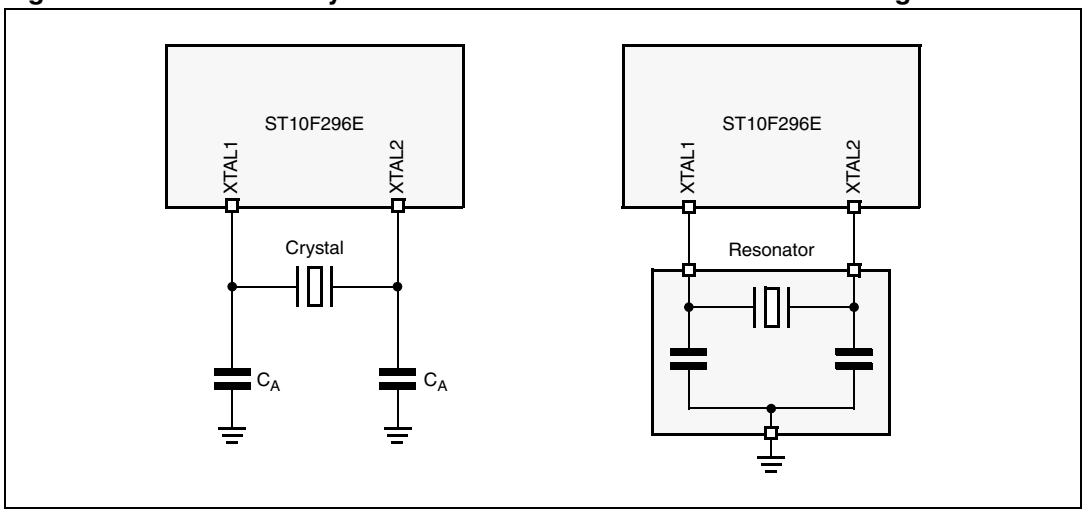


Table 172. Negative resistance (absolute min value @125 °C/V_{DD} = 4.5 V)

C_A (pF)	12	15	18	22	27	33	39	47
4 MHz	460 Ω	550 Ω	675 Ω	800 Ω	840 Ω	1000 Ω	1180 Ω	1200 Ω
8 MHz	380 Ω	460 Ω	540 Ω	640 Ω	580 Ω	-	-	-
12 MHz	370 Ω	420 Ω	360 Ω	-	-	-	-	-

The given values of C_A do not include the stray capacitance of the package or of the printed circuit board. The negative resistance values are calculated assuming an additional 5 pF to the values in *Table 172*. The crystal shunt capacitance (C_0), the package, and the stray capacitance between XTAL1 and XTAL2 pins is globally assumed to be 4 pF.

The external resistance between XTAL1 and XTAL2 does not have to be taken into account, since it is already present on the silicon.

24.8.14 External clock drive XTAL1

When direct drive configuration is selected during reset, it is possible to drive the CPU clock directly from the XTAL1 pin, without any particular restrictions on the maximum frequency, since the on-chip oscillator amplifier is bypassed. The speed limit is imposed by internal logic that targets a maximum CPU frequency of 64 MHz.

In all other clock configurations (direct drive with prescaler or PLL use) the on-chip oscillator amplifier is not bypassed, so it determines the input clock speed limit. In this case, an external clock source can be used, but it is limited in the range of frequencies defined for the use of crystal and resonator (see *Table 168 on page 313*).

External clock drive timing conditions: V_{DD} = 5 V ±10 %, V_{SS} = 0 V, T_A = -40 to 125 °C.

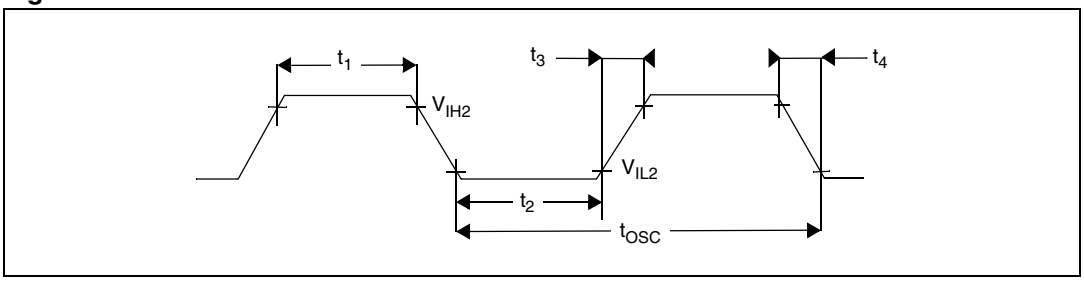
Table 173. External clock drive timing

Symbol	Parameter	Direct drive $f_{CPU} = f_{XTAL}$		Direct drive with prescaler $f_{CPU} = f_{XTAL}/2$		PLL use $f_{CPU} = f_{XTAL} \times F$		Unit
		Min	Max	Min	Max	Min	Max	
t_{OSC} (SR)	XTAL1 period(1)	15.625	-	83.3	250	83.3	250	ns
t_1 (SR)	High time(2)	6	-	3	-	6	-	
t_2 (SR)	Low time(2)							
t_3 (SR)	Rise time(2)	-	2	-	2	-	2	
t_4 (SR)	Fall time(2)							

1. The minimum value for the XTAL1 signal period is considered as the theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal.
2. The input clock signal must reach the defined levels V_{IL2} and V_{IH2}.

The input frequency range is 4-12 MHz when using an external clock source. With an external clock source, 64 MHz can be applied only when Direct Drive mode is selected. In this case, the oscillator amplifier is bypassed so it does not limit the input frequency.

Figure 107. External clock drive XTAL1



1. When direct drive is selected, an external clock source can be used to drive XTAL1. The maximum frequency of the external clock source depends on the duty cycle. When 64 MHz is used, 50 % duty cycle is granted (low phase = high phase = 7.8 ns). When 32 MHz is used, a 25 % duty cycle can be accepted (minimum, high or low phase = 7.8 ns).

24.8.15 Memory cycle variables

Table 174 describes how three variables derived from the BUSCONx registers are computed. These variables represent special characteristics of the programmed memory cycle.

Table 174. Memory cycle variables

Symbol	Description	Values
t_A	ALE extension	TCL x [ALECTL]
t_C	Memory cycle time wait states	2TCL x (15 - [MCTC])
t_F	Memory tri-state time	2TCL x (1 - [MTTC])

24.8.16 External memory bus timing

The next sections, *Multiplexed bus timings* and *Demultiplexed bus timings*, describe the external memory bus timings. The given values are computed for a maximum CPU clock of 40 MHz.

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

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

Multiplexed bus timings

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

ALE cycle time = 6 TCL + $2t_A$ + t_C + t_F (75 ns at 40 MHz CPU clock without wait states).

Table 175. Multiplexed bus timings

Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_5 (CC)	ALE high time	4 + t_A	-	TCL - 8.5 + t_A	-	ns
t_6 (CC)	Address setup to ALE	1.5 + t_A		TCL - 11 + t_A		
t_7 (CC)	Address hold after ALE	4 + t_A		TCL - 8.5 + t_A		
t_8 (CC)	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with R/W delay)	4 + t_A		TCL - 8.5 + t_A		
t_9 (CC)	ALE falling edge to $\overline{RD}$ and $\overline{WR}$ (no R/W delay)	-8.5 + t_A		-8.5 + t_A		
t_{10} (CC)	Address float after $\overline{RD}$ and $\overline{WR}$ (with R/W delay)(1)	-	6	-	6	
t_{11} (CC)	Address float after $\overline{RD}$ and $\overline{WR}$ (no R/W delay)(1)		18.5		TCL + 6	
t_{12} (CC)	$\overline{RD}$ and $\overline{WR}$ low time (with R/W delay)	15.5 + t_C	-	2TCL - 9.5 + t_C	-	
t_{13} (CC)	$\overline{RD}$ and $\overline{WR}$ low time (no R/W delay)	28 + t_C		3TCL - 9.5 + t_C		
t_{14} (SR)	$\overline{RD}$ to valid data in (with R/W delay)	-	6 + t_C	-	2TCL - 19 + t_C	
t_{15} (SR)	$\overline{RD}$ to valid data in (no R/W delay)		18.5 + t_C		3TCL - 19 + t_C	
t_{16} (SR)	ALE low to valid data in		17.5 + t_A + t_C		3TCL - 20 + t_A + t_C	
t_{17} (SR)	Address/unlatched $\overline{CS}$ to valid data in		20 + $2t_A$ + t_C		4TCL - 30 + $2t_A$ + t_C	
t_{18} (SR)	Data hold after $\overline{RD}$ rising edge	0	-	0	-	

Table 175. Multiplexed bus timings (continued)

Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_{19} (SR)	Data float after $\overline{RD}$(1)	-	16.5 + t_F	-	2TCL - 8.5 + t_F	ns
t_{22} (CC)	Data valid to $\overline{WR}$	10 + t_C	-	2TCL - 15 + t_C	-	
t_{23} (CC)	Data hold after $\overline{WR}$	4 + t_F		2TCL - 8.5 + t_F		
t_{25} (CC)	ALE rising edge after $\overline{RD}$ and $\overline{WR}$	15 + t_F		2TCL - 10 + t_F		
t_{27} (CC)	Address/unlatched $\overline{CS}$ hold after $\overline{RD}$ and $\overline{WR}$	10 + t_F		2TCL - 15 + t_F		
t_{38} (CC)	ALE falling edge to latched $\overline{CS}$	-4 - t_A	10 - t_A	-4 - t_A	10 - t_A	
t_{39} (SR)	Latched $\overline{CS}$ low to valid data in	-	16.5 + t_C+ 2t_A	-	3TCL - 21+ t_C+ 2t_A	
t_{40} (CC)	Latched $\overline{CS}$ hold after $\overline{RD}$ and $\overline{WR}$	27 + t_F	-	3TCL - 10.5 + t_F	-	
t_{42} (CC)	ALE falling edge to $\overline{RdCS}$ and $\overline{WrCS}$ (with R/W delay)	7 + t_A		TCL - 5.5 + t_A		
t_{43} (CC)	ALE falling edge to $\overline{RdCS}$ and $\overline{WrCS}$ (no R/W delay)	-5.5 + t_A		-5.5 + t_A		
t_{44} (CC)	Address float after $\overline{RdCS}$ and $\overline{WrCS}$ (with R/W delay)(1)	-	1.5	-	1.5	
t_{45} (CC)	Address float after $\overline{RdCS}$ and $\overline{WrCS}$ (no R/W delay)		14		TCL + 1.5	
t_{46} (SR)	$\overline{RdCS}$ to valid data in (with R/W delay)		4 + t_C		2TCL - 21 + t_C	
t_{47} (SR)	$\overline{RdCS}$ to valid data in (no R/W delay)		16.5 + t_C		3TCL - 21 + t_C	
t_{48} (CC)	$\overline{RdCS}$ and $\overline{WrCS}$ low time (with R/W delay)	15.5 + t_C	-	2TCL - 9.5 + t_C	-	
t_{49} (CC)	$\overline{RdCS}$ and $\overline{WrCS}$ low time (no R/W delay)	28 + t_C		3TCL - 9.5 + t_C		
t_{50} (CC)	Data valid to $\overline{WrCS}$	10 + t_C		2TCL - 15 + t_C		
t_{51} (SR)	Data hold after $\overline{RdCS}$	0		0		
t_{52} (SR)	Data float after $\overline{RdCS}$(1)	-	16.5 + t_F	-	2TCL - 8.5 + t_F	
t_{54} (CC)	Address hold after $\overline{RdCS}$ and $\overline{WrCS}$	6 + t_F	-	2TCL - 19 + t_F	-	
t_{56} (CC)	Data hold after $\overline{WrCS}$					

1. Partially tested, guaranteed by design characterization.

The following figures (*Figure 108* to *Figure 111*) present the different configurations of the external memory cycle for a multiplxed bus.

Figure 108. Multiplexed bus with/without R/W delay and normal ALE

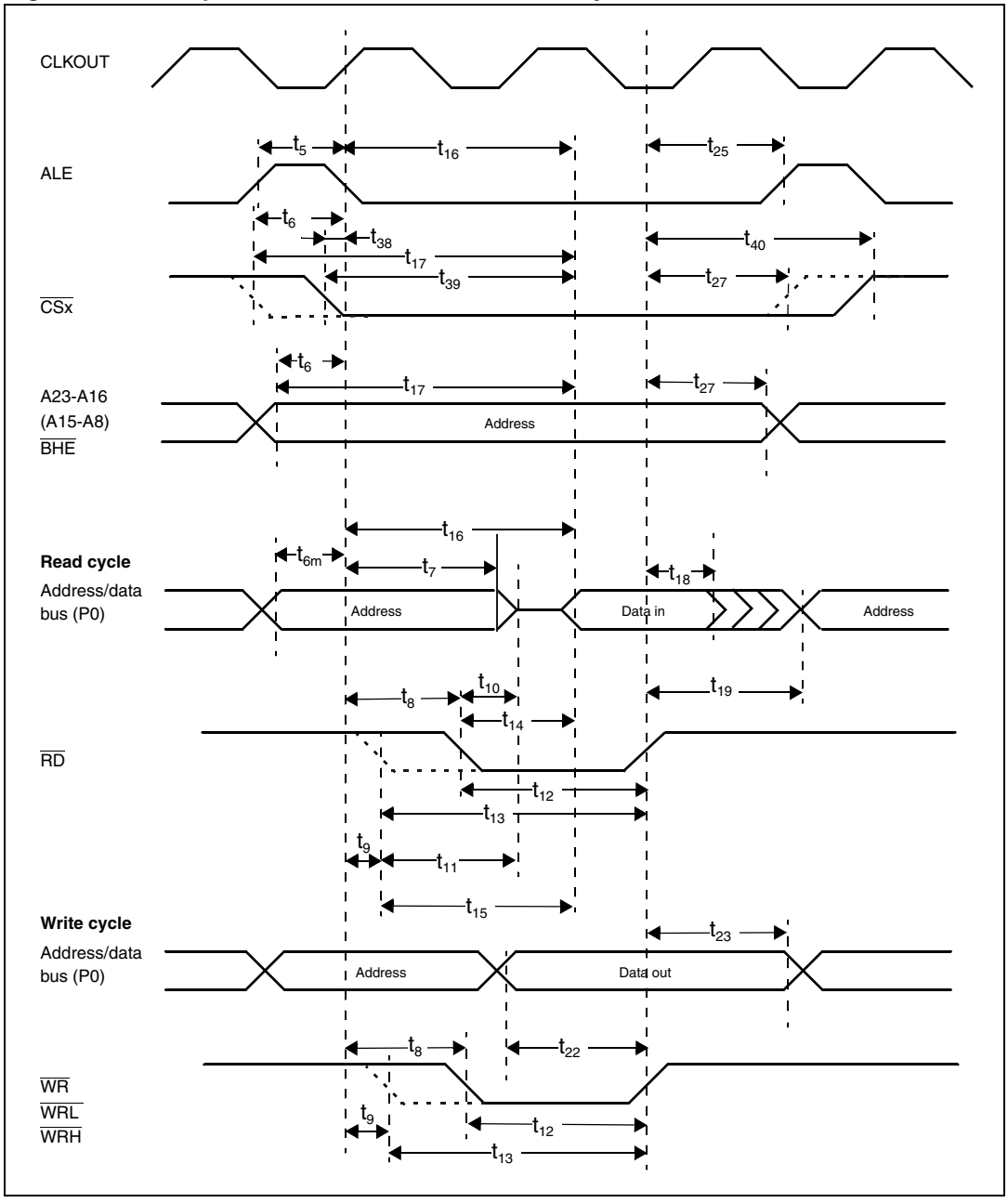


Figure 109. Multiplexed bus with/without R/W delay and extended ALE

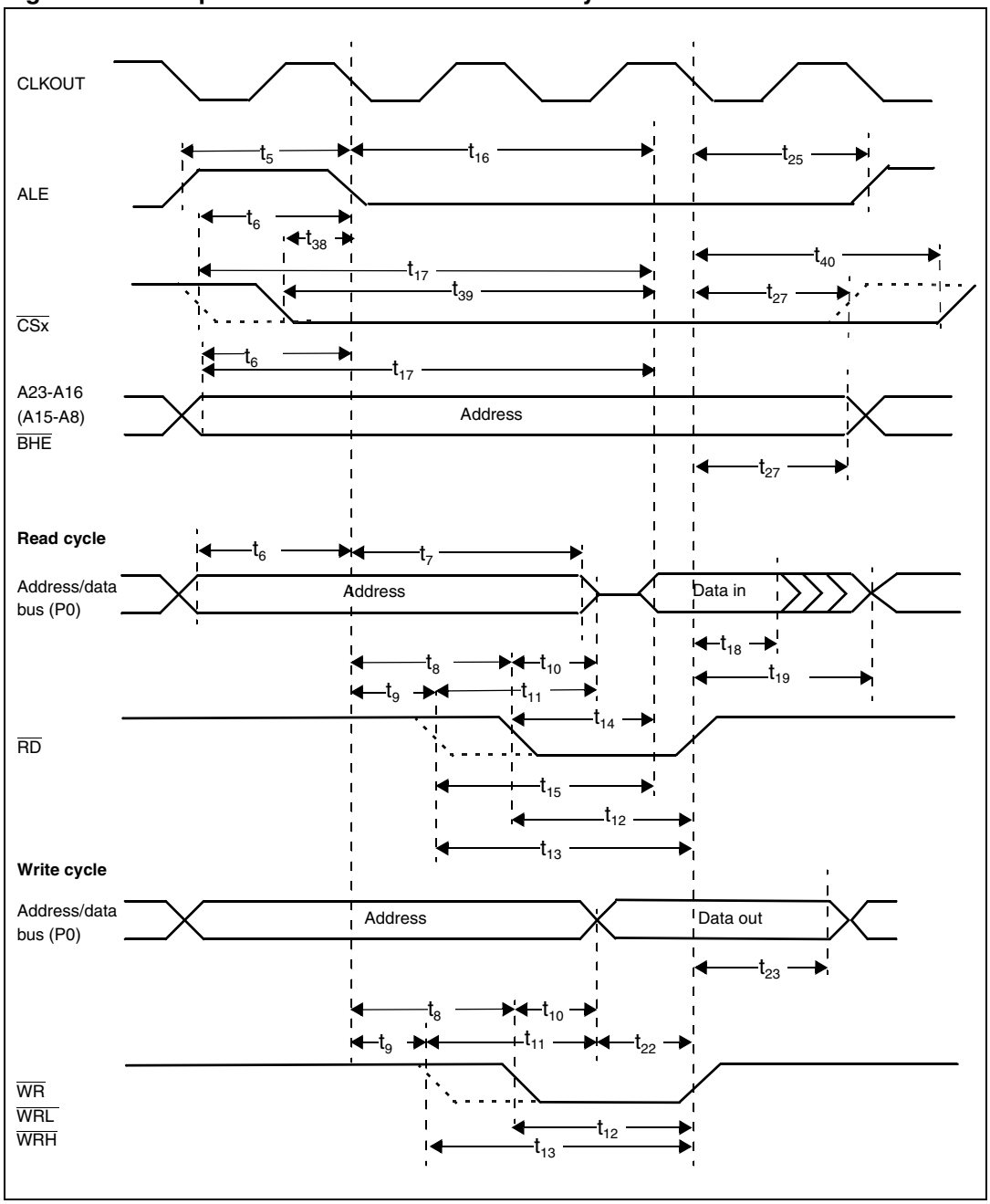


Figure 110. Multiplexed bus with/without R/W delay, normal ALE, R/W $\overline{CS}$

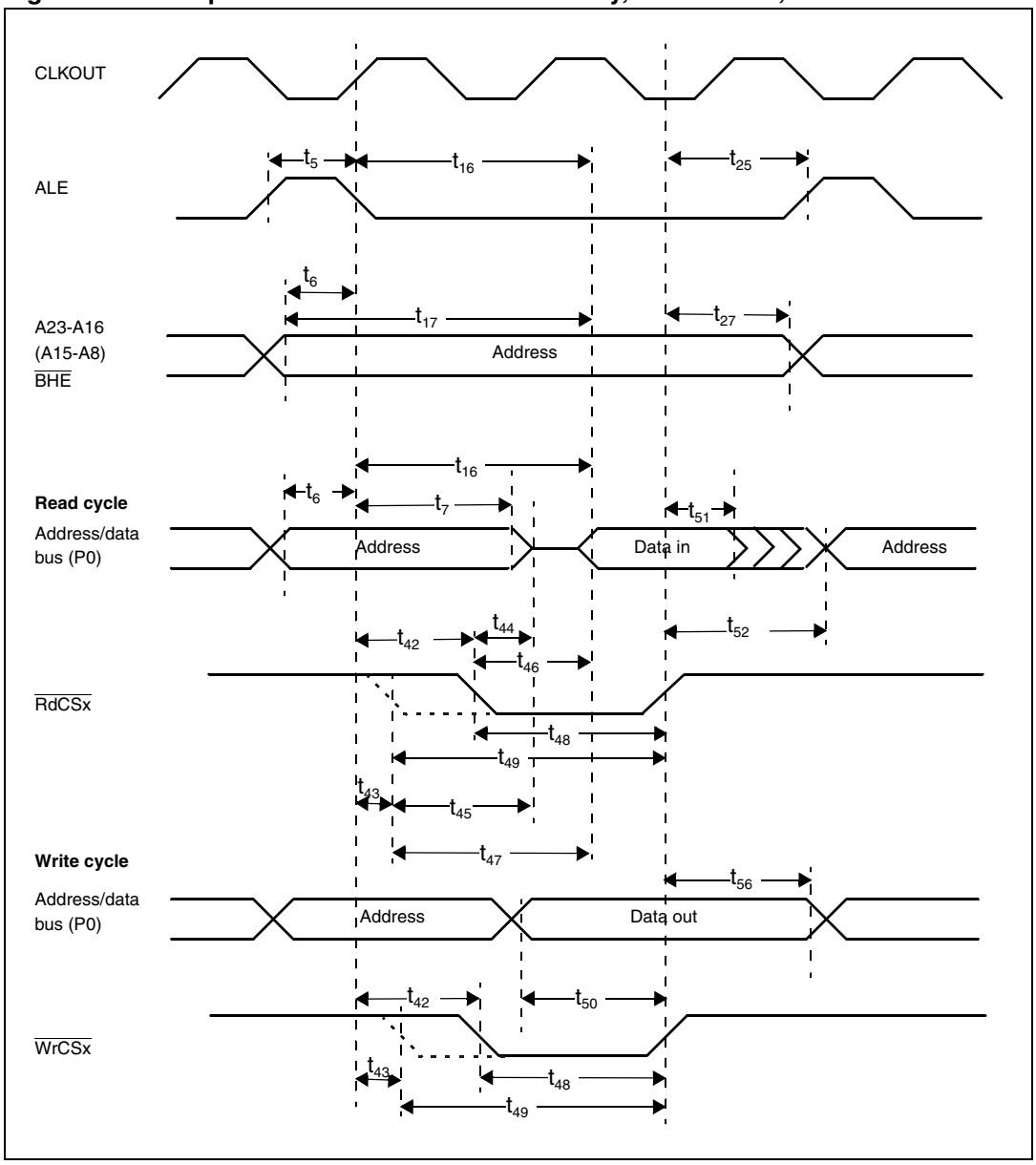
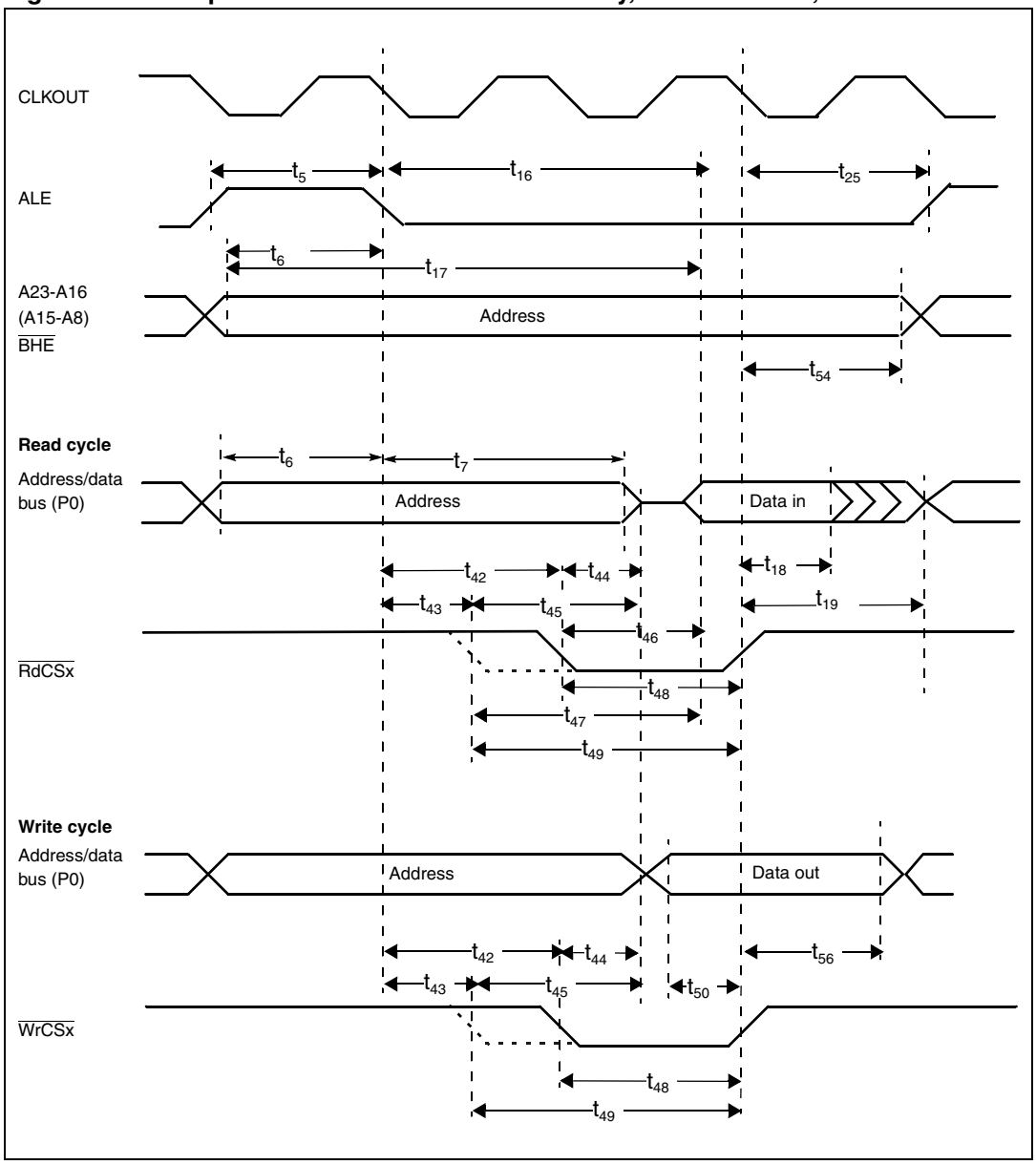


Figure 111. Multiplexed bus with/without R/ W delay, extended ALE, R/W $\overline{CS}$



Demultiplexed bus timings

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

ALE cycle time = 4 TCL + $2t_A$ + t_C + t_F (50 ns at 40 MHz CPU clock without wait states).

Table 176. Demultiplexed bus

Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_5 (CC)	ALE high time	4 + t_A	-	TCL - 8.5 + t_A	-	ns
t_6 (CC)	Address setup to ALE	1.5 + t_A	-	TCL - 11 + t_A	-	ns
t_{80} (CC)	Address/unlatched $\overline{CS}$ setup to $\overline{RD}$ and $\overline{WR}$ (with R/W delay)	12.5 + $2t_A$	-	2TCL - 12.5 + $2t_A$	-	ns
t_{81} (CC)	Address/unlatched $\overline{CS}$ setup to $\overline{RD}$ and $\overline{WR}$ (no R/W delay)	0.5 + $2t_A$	-	TCL - 12 + $2t_A$	-	ns
t_{12} (CC)	$\overline{RD}$ and $\overline{WR}$ low time (with R/W delay)	15.5 + t_C	-	2TCL - 9.5 + t_C	-	ns
t_{13} (CC)	$\overline{RD}$ and $\overline{WR}$ low time (no R/W delay)	28 + t_C	-	3TCL - 9.5 + t_C	-	ns
t_{14} (SR)	$\overline{RD}$ to valid data in (with R/W delay)	-	6 + t_C	-	2TCL - 19 + t_C	ns
t_{15} (SR)	$\overline{RD}$ to valid data in (no R/W delay)	-	18.5 + t_C	-	3TCL - 19 + t_C	ns
t_{16} (SR)	ALE low to valid data in	-	17.5 + t_A + t_C	-	3TCL - 20 + t_A + t_C	ns
t_{17} (SR)	Address/unlatched $\overline{CS}$ to valid data in	-	20 + $2t_A$ + t_C	-	4TCL - 30 + $2t_A$ + t_C	ns
t_{18} (SR)	Data hold after $\overline{RD}$ rising edge	0	-	0	-	ns
t_{20} (SR)	Data float after $\overline{RD}$ rising edge (with R/W delay)(1)	-	16.5 + t_F	-	2TCL - 8.5 + t_F + $2t_A$	ns
t_{21} (SR)	Data float after $\overline{RD}$ rising edge (no R/W delay)(1)	-	4 + t_F	-	TCL - 8.5 + t_F + $2t_A$	ns
t_{22} (CC)	Data valid to $\overline{WR}$	10 + t_C	-	2TCL - 15 + t_C	-	ns
t_{24} (CC)	Data hold after $\overline{WR}$	4 + t_F	-	TCL - 8.5 + t_F	-	ns
t_{26} (CC)	ALE rising edge after $\overline{RD}$ and $\overline{WR}$	-10 + t_F	-	-10 + t_F	-	ns
t_{28} (CC)	Address/unlatched $\overline{CS}$ hold after $\overline{RD}$ and $\overline{WR}$(2)	0 + t_F	-	0 + t_F	-	ns
t_{28h} (CC)	Address/unlatched $\overline{CS}$ hold after $\overline{WRH}$	-5 + t_F	-	-5 + t_F	-	ns

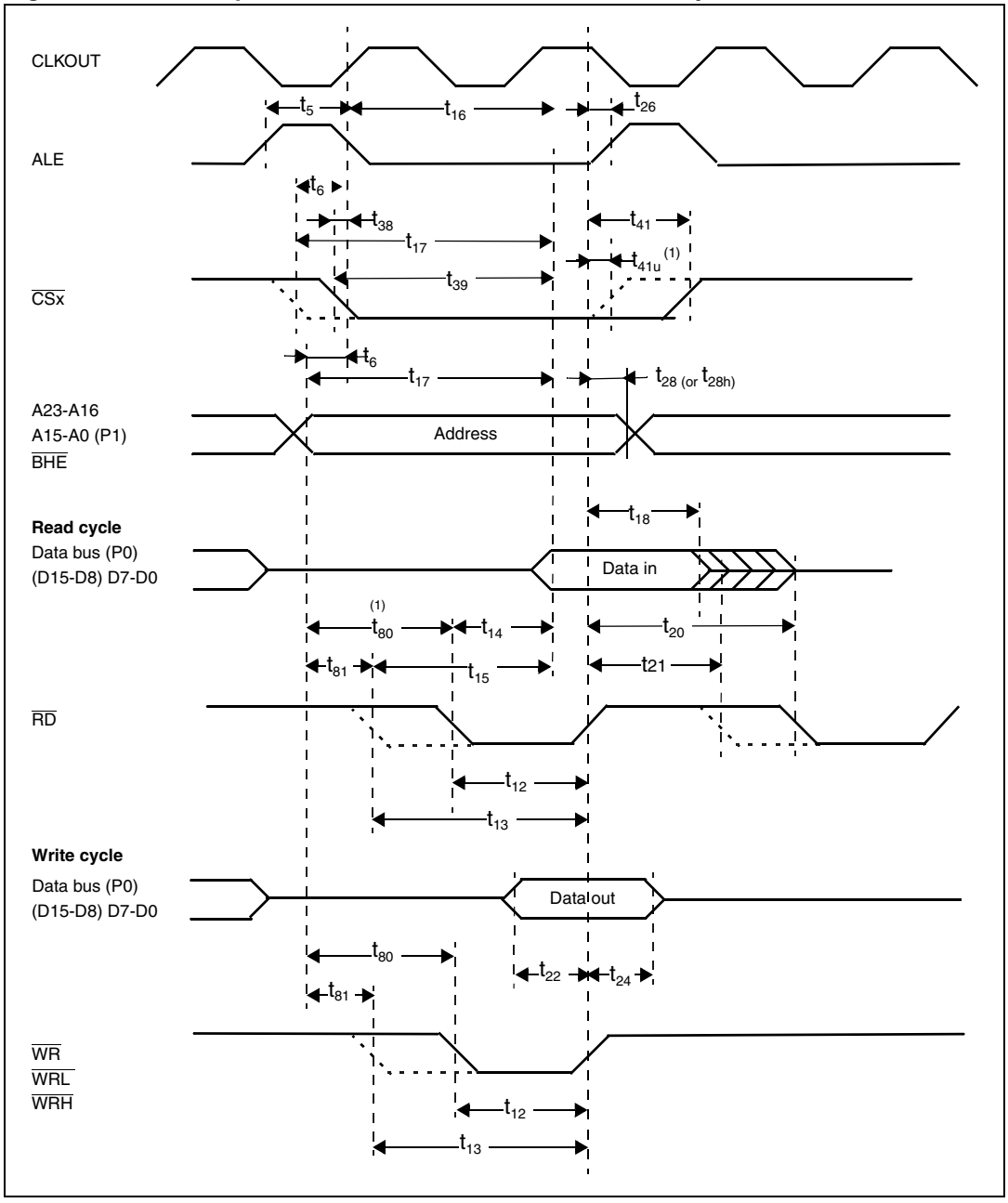
Table 176. Demultiplexed bus (continued)

Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_{38} (CC)	ALE falling edge to latched $\overline{CS}$	-4 - t_A	6 - t_A	-4 - t_A	6 - t_A	ns
t_{39} (SR)	Latched $\overline{CS}$ low to valid data in	-	16.5 + t_C + 2t_A	-	3TCL - 21+ t_C + 2t_A	ns
t_{41} (CC)	Latched $\overline{CS}$ hold after $\overline{RD}$ and $\overline{WR}$	2 + t_F	-	TCL - 10.5 + t_F	-	ns
t_{82} (CC)	Address setup to $\overline{RdCS}$ and $\overline{WrCS}$ (with R/W delay)	14 + 2t_A	-	2TCL - 11 + 2t_A	-	ns
t_{83} (CC)	Address setup to $\overline{RdCS}$ and $\overline{WrCS}$ (no R/W delay)	2 + 2t_A	-	TCL - 10.5 + 2t_A	-	ns
t_{46} (SR)	$\overline{RdCS}$ to valid data in (with R/W delay)	-	4 + t_C	-	2TCL - 21 + t_C	ns
t_{47} (SR)	$\overline{RdCS}$ to valid data in (no R/W delay)	-	16.5 + t_C	-	3TCL - 21 + t_C	ns
t_{48} (CC)	$\overline{RdCS}$ and $\overline{WrCS}$ low time (with RW-delay)	15.5 + t_C	-	2TCL - 9.5 + t_C	-	ns
t_{49} (CC)	$\overline{RdCS}$ and $\overline{WrCS}$ low time (no R/W delay)	28 + t_C	-	3TCL - 9.5 + t_C	-	ns
t_{50} (CC)	Data valid to $\overline{WrCS}$	10 + t_C	-	2TCL - 15 + t_C	-	ns
t_{51} (SR)	Data hold after $\overline{RdCS}$	0	-	0	-	ns
t_{53} (SR)	Data float after $\overline{RdCS}$ (with R/W delay)	-	16.5 + t_F	-	2TCL - 8.5 + t_F	ns
t_{68} (SR)	Data float after $\overline{RdCS}$ (no R/W delay)	-	4 + t_F	-	TCL - 8.5 + t_F	ns
t_{55} (CC)	Address hold after $\overline{RdCS}$ and $\overline{WrCS}$	-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. R/W delay and t_A refer to the next bus cycle.
2. Read data is latched with the same clock edge that triggers the address change and the rising $\overline{RD}$ edge. Therefore address changes which occur before the end of $\overline{RD}$ have no impact on read cycles.

The following figures (*Figure 112* to *Figure 115*) present the different configurations of external memory cycle for a demultiplxed bus.

Figure 112. Demultiplexed bus with/without read/write delay and normal ALE



1. Unlatched CSx = t_{41u} = t_{41} TCL = 10.5 + t_F.

Figure 113. Demultiplexed bus with/without R/W delay and extended ALE

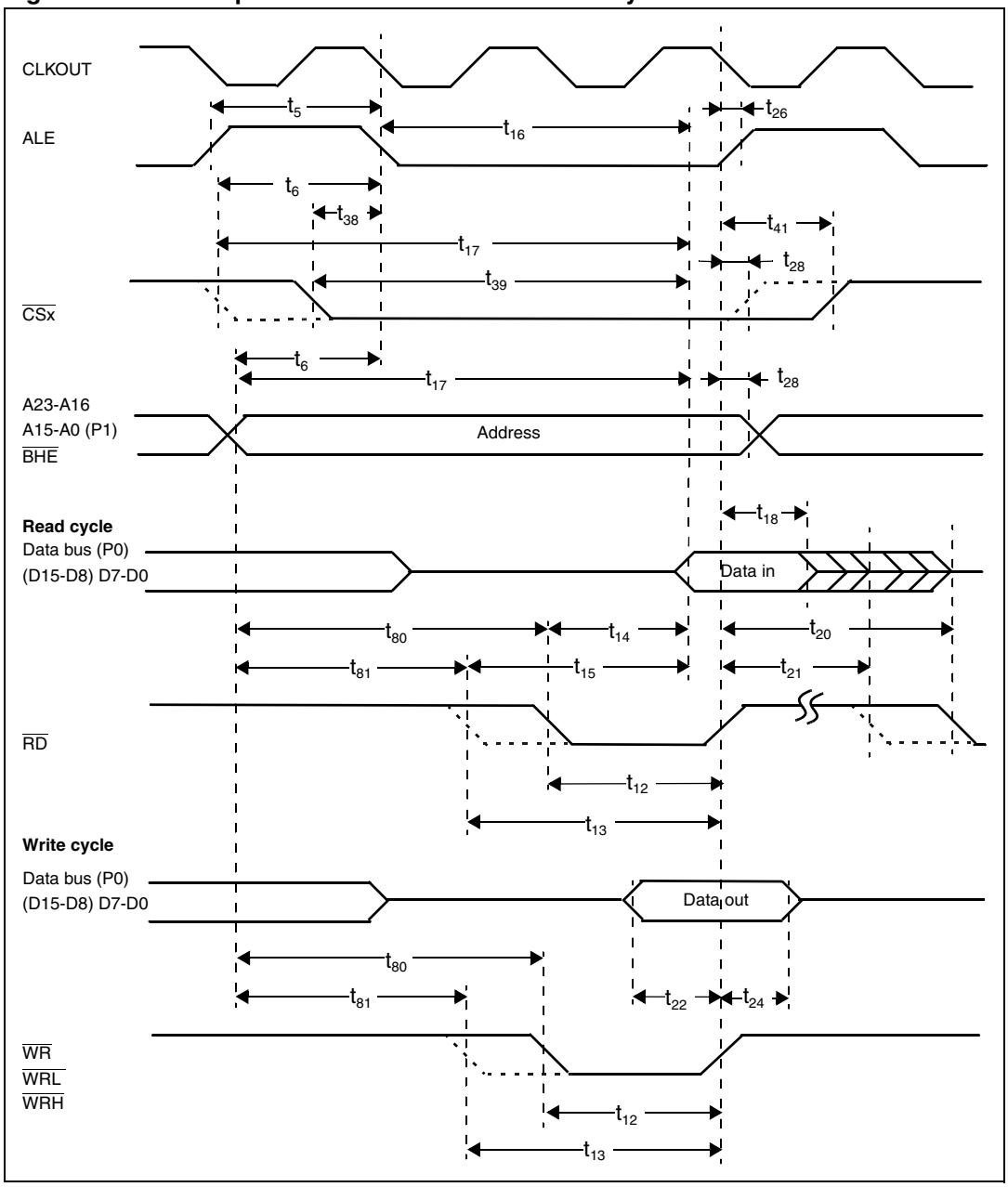


Figure 114. Demultiplexed bus with ALE and R/W $\overline{CS}$

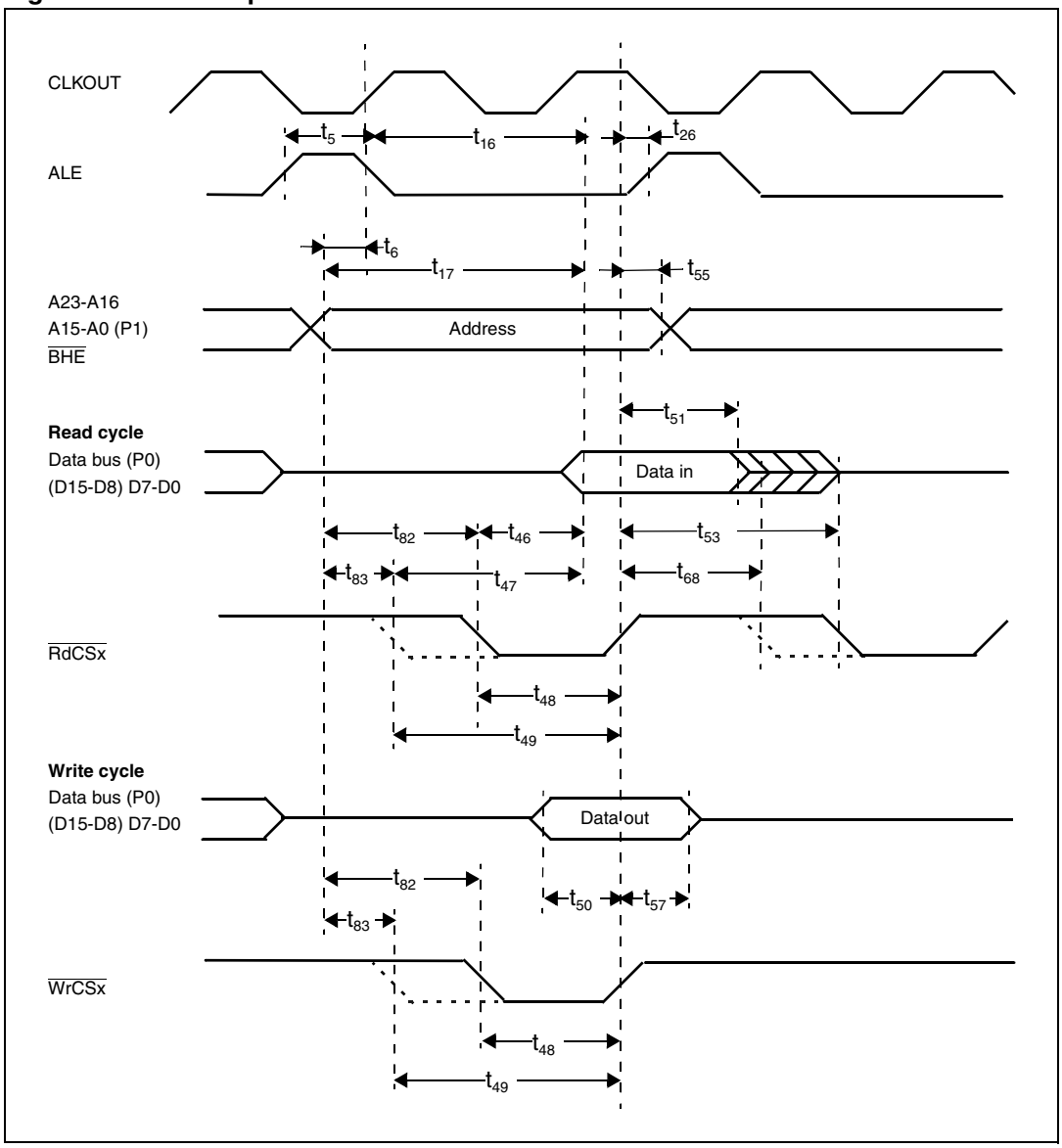
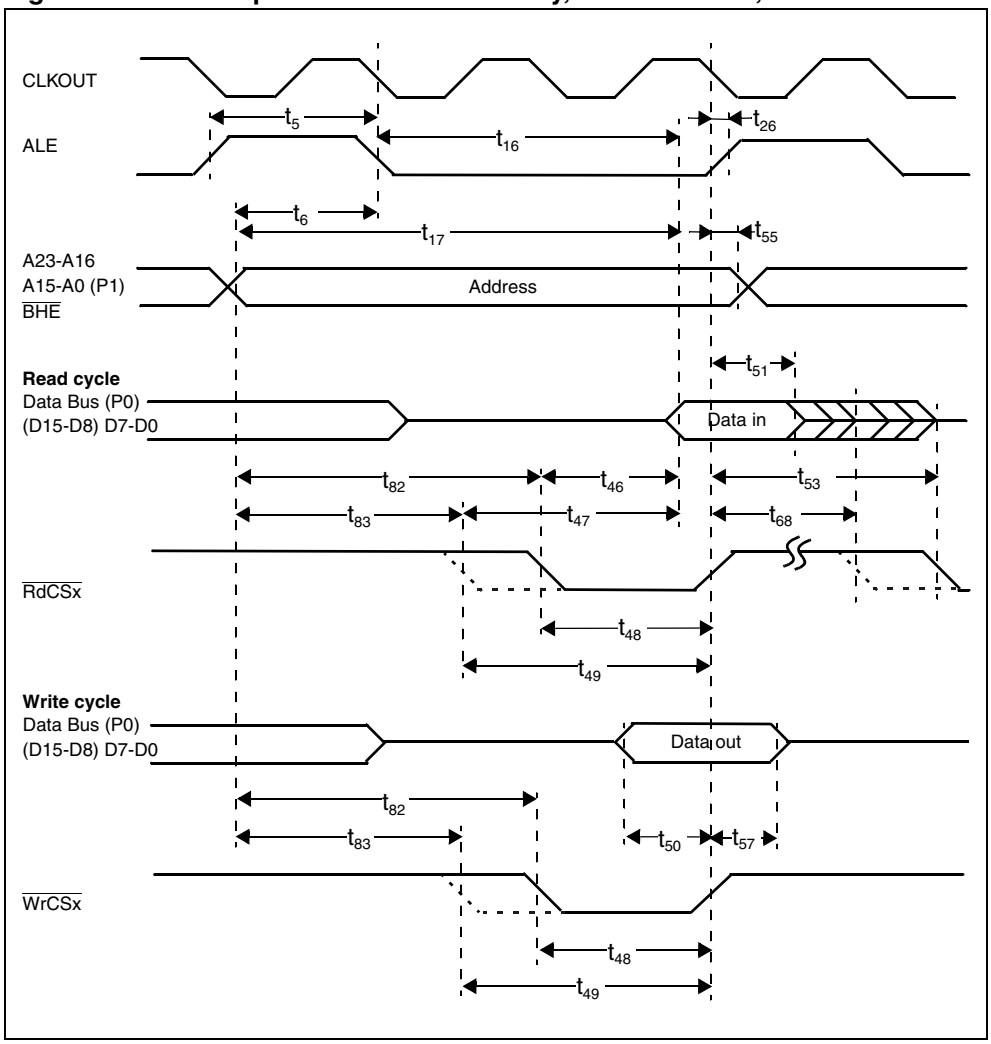


Figure 115. Demultiplexed bus no R/W delay, extended ALE, R/W $\overline{CS}$



24.8.17 $\overline{\text{READY}}$ and CLKOUT

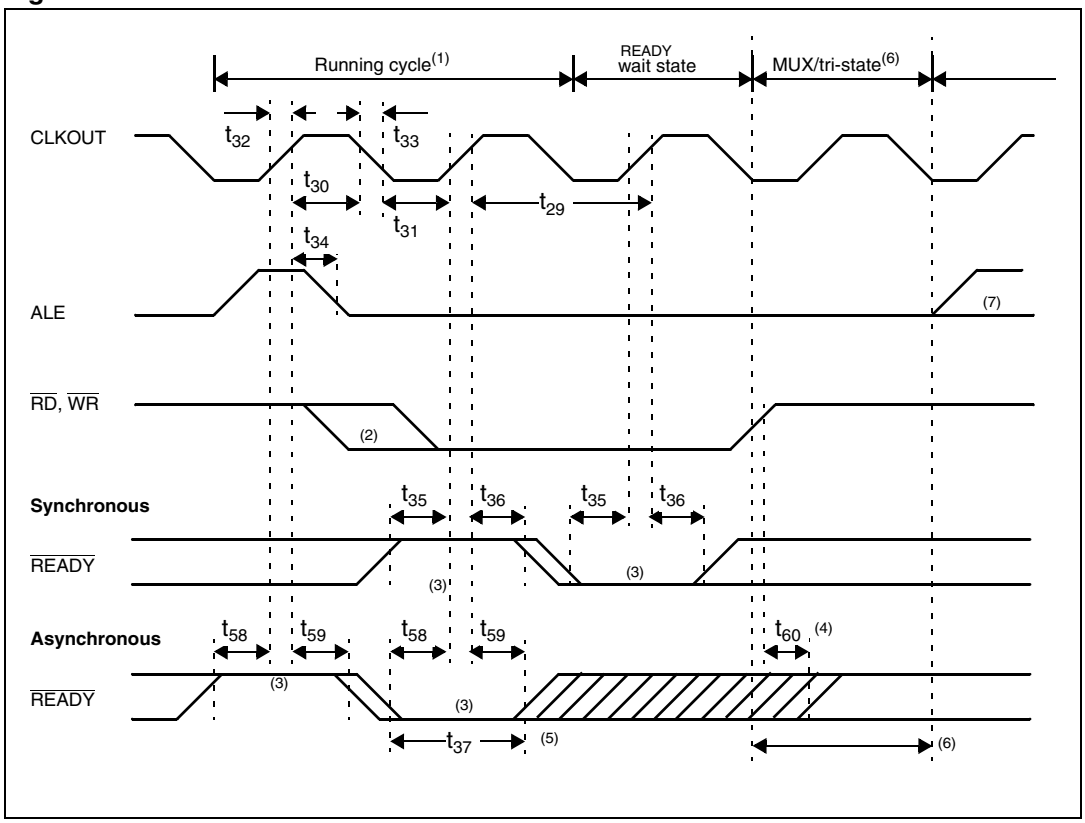
V_{DD} = 5 V ±10 %, V_{SS} = 0 V, T_A = -40 to 125°C, C_L = 50 pF.

Table 177. $\overline{\text{READY}}$ and CLKOUT

Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_{29} (CC)	CLKOUT cycle time	25	25	2 TCL	2 TCL	ns
t_{30} (CC)	CLKOUT high time	9	-	TCL - 3.5	-	
t_{31} (CC)	CLKOUT low time	10		TCL - 2.5		
t_{32} (CC)	CLKOUT rise time	-	4	-	4	
t_{33} (CC)	CLKOUT fall time					
t_{34} (CC)	CLKOUT rising edge to ALE falling edge	-2 + t_A	8 + t_A	-2 + t_A	8 + t_A	
t_{35} (SR)	Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	17	-	17	-	
t_{36} (SR)	Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	2		2		
t_{37} (SR)	Asynchronous $\overline{\text{READY}}$ low time	35		2 TCL + 10		
t_{58} (SR)	Asynchronous $\overline{\text{READY}}$ setup time(1)	17		17		
t_{59} (SR)	Asynchronous $\overline{\text{READY}}$ hold time(1)	2		2		
t_{60} (SR)	Asynchronous $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$ and $\overline{\text{WR}}$ high (demultiplexed bus)(2)	0	$2t_A$ + tC + t_F	0	$2t_A$ + tC + t_F	

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

Figure 116. $\overline{\text{READY}}$ and CLKOUT



1. Cycle as programmed, including MCTC wait states (example shows 0 MCTC wait states).
2. The leading edge of the respective command depends on R/W delay.
3. $\overline{\text{READY}}$ sampled high at this sampling point generates a READY controlled wait state, $\overline{\text{READY}}$ sampled low at this sampling point terminates the current running bus cycle.
4. $\overline{\text{READY}}$ may be deactivated in response to the trailing (rising) edge of the corresponding command ($\overline{\text{RD}}$ or $\overline{\text{WR}}$).
5. If the asynchronous $\overline{\text{READY}}$ signal does not fulfill the indicated setup and hold times with respect to CLKOUT (for example, because CLKOUT is not enabled), it must fulfill t_{37} 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.

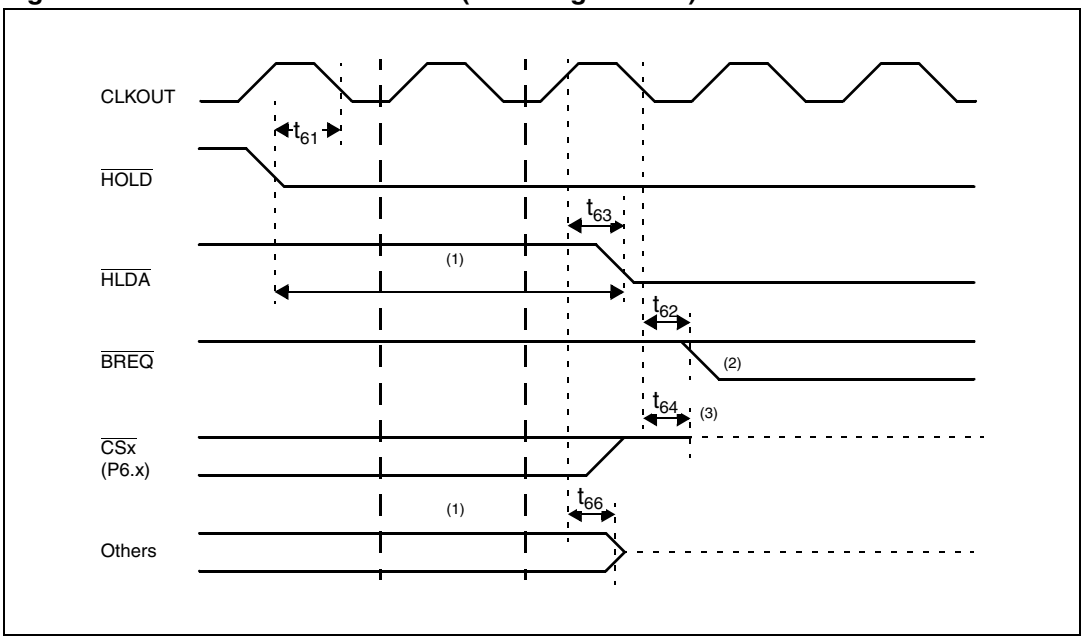
24.8.18 External bus arbitration

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

Table 178. External bus arbitration

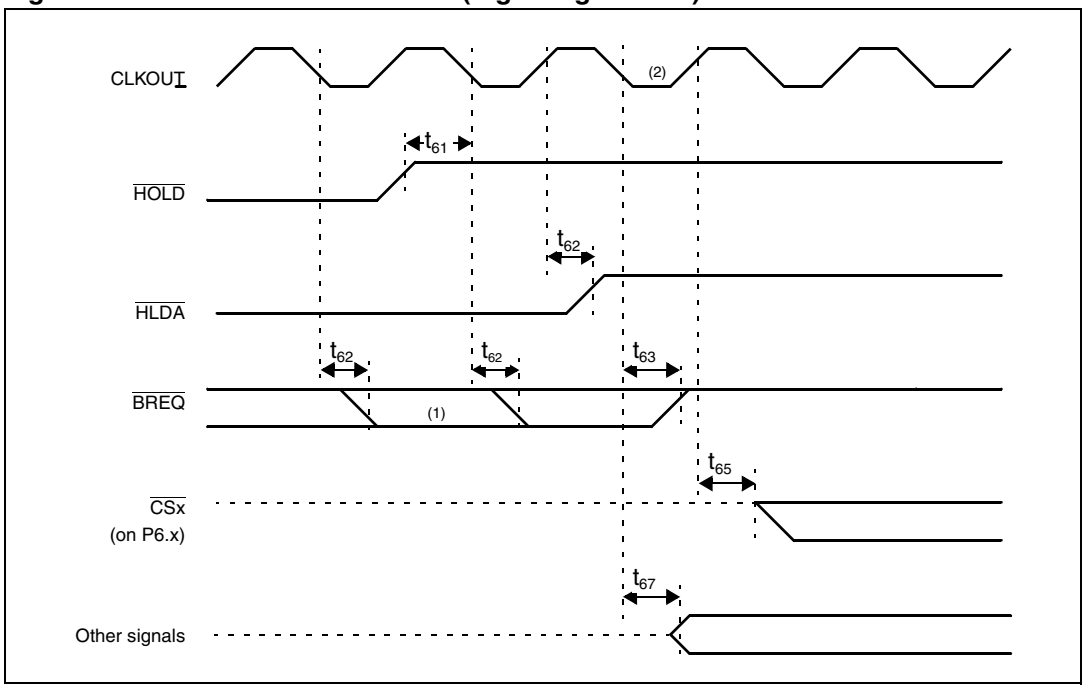
Symbol	Parameter	F_{CPU} = 40 MHz TCL = 12.5 ns		Variable CPU clock 1/2 TCL = 1 to 64 MHz		Unit
		Min	Max	Min	Max	
t_{61} (SR)	$\overline{HOLD}$ input setup time to CLKOUT	18.5	-	18.5	-	ns
t_{62} (CC)	CLKOUT to $\overline{HLDA}$ high or $\overline{BREQ}$ low delay	-	12.5	-	12.5	
t_{63} (CC)	CLKOUT to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

Figure 117. External bus arbitration (releasing the bus)



1. The ST10F296E completes the current running bus cycle before granting bus access.
2. This is the first possibility for $\overline{BREQ}$ to become active.
3. The $\overline{CS}$ outputs are resistive high (pull-up) after t_{64}.

Figure 118. External bus arbitration (regaining the bus)



1. This is the last chance for $\overline{\text{BREQ}}$ to trigger the indicated regain sequence. Even if $\overline{\text{BREQ}}$ is activated earlier, the regain sequence is initiated by $\overline{\text{HOLD}}$ going high. Note that $\overline{\text{HOLD}}$ may also be deactivated without the ST10F296E requesting the bus.
2. The next ST10F296E driven bus cycle may start here.

24.8.19 High-speed synchronous serial interface (SSC) timing modes

Master mode

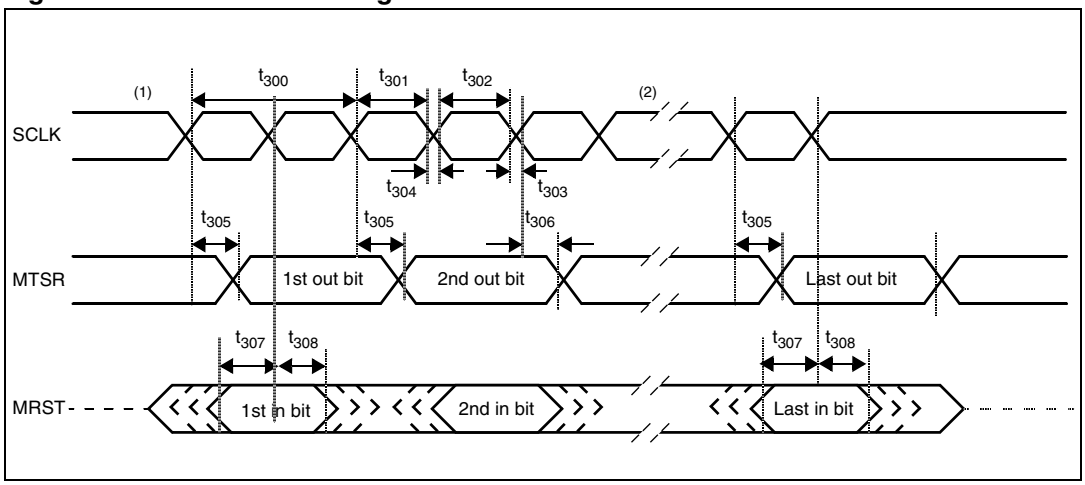
V_{DD} = 5 V ±10 %, V_{SS} = 0 V, T_A = -40 to 125°C, C_L = 50 pF.

Table 179. Master mode

Symbol	Parameter	Max baud rate 6.6 MBd @F_{CPU} = 40 MHz (<SSCBR> = 0002h)(1)		Variable baud rate (<SSCBR> = 0001h - FFFFh)		Unit
		Min	Max	Min	Max	
t_{300} (CC)	SSC clock cycle time(2)	150	150	8 TCL	262144 TCL	ns
t_{301} (CC)	SSC clock high time	63	–	t_{300}/2 - 12	-	
t_{302} (CC)	SSC clock low time					
t_{303} (CC)	SSC clock rise time	-	10	-	10	
t_{304} (CC)	SSC clock fall time					
t_{305} (CC)	Write data valid after shift edge		15		15	
t_{306} (CC)	Write data hold after shift edge	-2	-	-2	-	
t_{307p} (SR)	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	37.5		2TCL + 12.5		
t_{308p} (SR)	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	50		4 TCL		
t_{307} (SR)	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	25		2 TCL		
t_{308} (SR)	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	0		0		

1. Maximum baud rate is 8 Mbaud and can be reached with 64 MHz CPU clock and <SSCBR> set to 3h, or with 48 MHz CPU clock and <SSCBR> set to 2h. When 40 MHz CPU clock is used, the maximum baud rate cannot be higher than 6.6 Mbaud (<SSCBR> = 2h) due to the limited granularity of <SSCBR>. A value of 1h for <SSCBR> may be used only with a CPU clock equal to (or lower than) 32 MHz (after checking that timings are in line with the target slave).
2. The formula for the SSC clock cycle time is:
   t_{300} = 4 TCL x (<SSCBR> + 1)
   Where <SSCBR> represents the content of the SSC baud rate register, taken as an unsigned 16-bit integer. The minimum limit allowed for t_{300} is 125 ns (corresponding to 8 Mbaud).

Figure 119. SSC master timing



1. The phase and polarity of the shift and latch edges of SCLK are programmable. *Figure 119* uses the leading clock edge as the shift edge with the latch on the trailing edge (SSCPH = 0b). The idle clock line is low and the leading clock edge is low-to-high transition (SSCPO = 0b).
2. The bit timing is repeated for all bits that have to be transmitted or received.

Slave mode

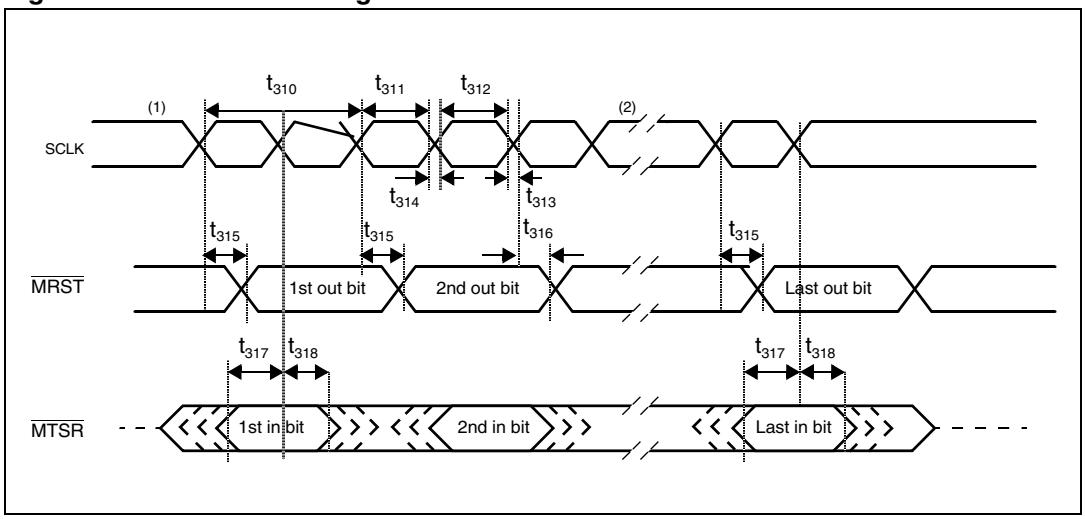
V_{DD} = 5 V ±10%, V_{SS} = 0 V, T_A = -40 to 125°C, C_L = 50 pF

Table 180. Slave mode

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

1. Maximum baud rate is 8 Mbaud and can be reached with 64 MHz CPU clock and <SSCBR> set to 3h, or with 48 MHz CPU clock and <SSCBR> set to 2h. When 40 MHz CPU clock is used, the maximum baud rate cannot be higher than 6.6 Mbaud (<SSCBR> = 2h) due to the limited granularity of <SSCBR>. A value of 1h for <SSCBR> may be used only with a CPU clock lower than 32 MHz (after checking that timings are in line with the target slave).
2. The formula for the SSC clock cycle time is:
   t_{310} = 4 TCL * (<SSCBR> + 1)
   Where <SSCBR> represents the content of the SSC baud rate register, taken as an unsigned 16-bit integer. The minimum limit allowed for t_{310} is 125 ns (corresponding to 8 Mbaud).

Figure 120. SSC slave timing



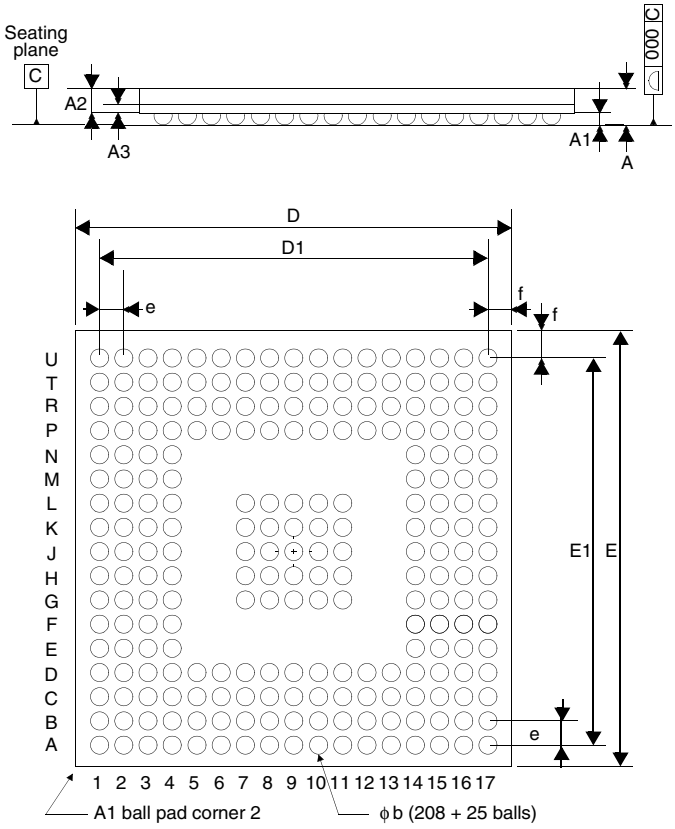
1. The phase and polarity of the shift and latch edges of SCLK are programmable. *Figure 120* uses the leading clock edge as the shift edge with the latch on the trailing edge (SSCPH = 0b). The idle clock line is low and the leading clock edge is low-to-high transition (SSCPO = 0b).
2. The bit timing is repeated for all bits that have to be transmitted or received.

25 Package mechanical data

To meet environmental requirements, ST offers the device in ECOPACK® packages. These packages have a lead-free second level interconnect. The category of second level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

ECOPACK is an ST trademark. ECOPACK® specifications are available at *www.st.com*..

Figure 121. PBGA 208 (23 x 23 x 1.96 mm) outline



1. PBGA stands for plastic ball grid array.
2. The terminal A1 corner of the package must be identified on the top surface by using a corner chamfer, ink or metallized marking, identation or other feature of the package body or an integral heastslug. A distinguishing feature is also allowable on the bottom of the package to identify the terminal A1 corner. The exact shape and size of this feature is optional.

Table 181. PBGA 208 (23 x 23 x 1.96 mm) mechanical data

Dimensions	Millimeters			Inches (approx)[1]		
	Minimum	Typical	Maximum	Minimum	Typical	Maximum
A		1.960			0.0772	
A1	0.500	0.600	0.700	0.0197	0.0236	0.0276
A2		1.360			0.0535	
A3		0.560			0.0220	
ϕ b	0.600	0.760	0.900	0.0236	0.0299	0.0354
D	22.900	23.000	23.100	0.9016	0.9055	0.9094
D1		20.320			0.8000	
E	22.900	23.000	23.100	0.9016	0.9055	0.9094
E1		20.320			0.8000	
e		1.270			0.0500	
f	1.240	1.340	1.440	0.0488	0.0528	0.0567

1. Values in inches are converted from mm and rounded to four decimal digits.

26 Ordering information

Table 182. Order codes

Order codes	Package	Packing	Temperature range (°C)	CPU frequency range (MHz)
ST10F296	PBGA208	Tray	-40 to 125	1 to 64
ST10F296TR		Tape and reel		

27 Revision history

Table 183. Document revision history

Date	Revision	Changes
24-Jan-2005	1	Initial release.
20-Oct-2008	2	Initial public release. Document reformatted; content of *Features* reworked to fit into one page (no technical changes); content of remaining document reworked to improve readability (no technical changes). Updated *Table 1: Device summary*. *Section 7: Central processing unit (CPU)*: Removed sections on the SYSCON register and MAC features; amended *Section 7.3*; removed table entitled *MAC coprocessor specific instructions* and replaced with *Table 46*; removed tables entitled *Pointer postmodification combinations for Rwn and IDXi* and *MAC registers referenced as 'CoReg'*. *Section 9: Interrupt system*: Updated introductory text; removed sections on *Extrenal interrupts* and *Interrupt control register*; removed some text from *Section 9.1: XPeripheral interrupt*. *Section 24: Electrical characteristics*: Updated *Table 164*, *Table 172*, *Table 176*, *Figure 99*, and *Figure 120*. *Section 25: Package mechanical data*: Added ECOPACK text. *Table 181: PBGA 208 (23 x 23 x 1.96 mm) mechanical data*: Converted values in inches to four decimal places.

Please Read Carefully:

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

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

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

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

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

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

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

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

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

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

STMicroelectronics group of companies

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

www.st.com