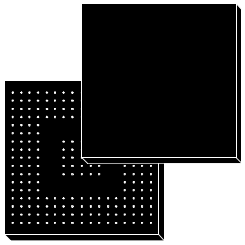
ST10F280

16-BIT MCU WITH MAC UNIT, 512K BYTE FLASH MEMORY AND 18K BYTE RAM

PRODUCT PREVIEW

- HIGH PERFORMANCE CPU WITH DSP FUNCTIONS
  - 16-BIT CPU WITH 4-STAGE PIPELINE.
  - 50ns INSTRUCTION CYCLE TIME AT 40MHz CPU CLOCK.
  - MULTIPLY/ACCUMULATE UNIT (MAC) 16 X 16-BIT MULTIPLICATION, 40-BIT ACCUMULATOR
  - REPEAT UNIT.
  - ENHANCED BOOLEAN BIT MANIPULATION FACILITIES.
  - ADDITIONAL INSTRUCTIONS TO SUPPORT HLL AND OPERATING SYSTEMS.
  - SINGLE-CYCLE CONTEXT SWITCHING SUPPORT.
- MEMORY ORGANIZATION
  - 512K BYTE ON-CHIP FLASH MEMORY SINGLE VOLTAGE WITH ERASE/PROGRAM CONTROLLER.
  - 100K ERASING/PROGRAMMING CYCLES.
  - 20 YEAR DATA RETENTION TIME
  - UP TO 16M BYTE LINEAR ADDRESS SPACE FOR CODE AND DATA (5M BYTE WITH CAN).
  - 2K BYTE ON-CHIP INTERNAL RAM (IRAM).
  - 16K BYTE EXTENSION RAM (XRAM).
- FAST AND FLEXIBLE BUS
  - PROGRAMMABLE EXTERNAL BUS CHARACTERISTICS FOR DIFFERENT ADDRESS RANGES.
  - 8-BIT OR 16-BIT EXTERNAL DATA BUS.
  - MULTIPLEXED OR DEMULTIPLEXED EXTERNAL ADDRESS/DATA BUSES.
  - 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, SAMPLE-RATE DOWN TO 25ns.
- TWO MULTI-FUNCTIONAL GENERAL PURPOSE TIMER UNITS WITH 5 TIMERS.
- TWO 16-CHANNEL CAPTURE/COMPARE UNITS
- A/D CONVERTER
  - 2X16-CHANNEL 10-BIT.
  - 4.85µS CONVERSION TIME
  - ONE TIMER FOR ADC CHANNEL INJECTION
- 8-CHANNEL PWM UNIT
- SERIAL CHANNELS
  - SYNCHRONOUS/ASYNC SERIAL CHANNEL
  - HIGH-SPEED SYNCHRONOUS CHANNEL.
- FAIL-SAFE PROTECTION
  - PROGRAMMABLE WATCHDOG TIMER.
  - OSCILLATOR WATCHDOG.



PBGA208 (23 x 23 x 1.96 - Pitch 1.27 mm)
(Plastic Bold Grid Array)

ORDER CODE: ST10F280-JT3

- TWO CAN 2.0b INTERFACES OPERATING ON ONE OR TWO CAN BUSSES (30 OR 2X15 MESSAGE OBJECTS)
- ON-CHIP BOOTSTRAP LOADER
- CLOCK GENERATION
  - ON-CHIP PLL.
  - DIRECT OR PRESCALED CLOCK INPUT.
- UP TO 143 GENERAL PURPOSE I/O LINES
  - INDIVIDUALLY PROGRAMMABLE AS INPUT, OUTPUT OR SPECIAL FUNCTION.
  - PROGRAMMABLE THRESHOLD (HYSTERESIS).
- IDLE AND POWER DOWN MODES
- MAXIMUM CPU FREQUENCY 40MHz
- PACKAGE PBGA 208 BALLS (23mm x 23mm x 1.96 mm - PITCH 1.27mm).
- SINGLE VOLTAGE SUPPLY: 5V ±10% (EMBEDDED REGULATOR FOR 3.3 V CORE SUPPLY).
- TEMPERATURE RANGE: -40 +125°C

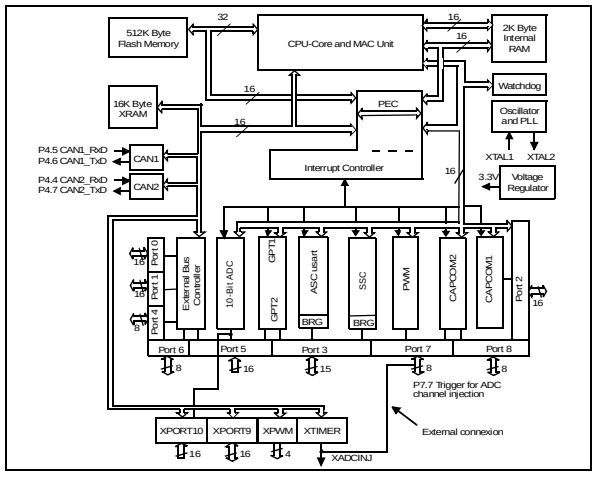


TABLE OF CONTENTS

1 - INTRODUCTION

The ST10F280 is a new derivative of the ST Microelectronics ST10 family of 16-bit single-chip CMOS microcontrollers. It combines high CPU performance (up to 20 million instructions per second) with high peripheral functionality and enhanced I/O-capabilities. It also provides on-chip high-speed single voltage FLASH memory, on-chip high-speed RAM, and clock generation via PLL.

ST10F280 is processed in 0.35µm CMOS technology. The MCU core and the logic is supplied with a 5V to 3.3V on chip voltage regulator. The part is supplied with a single 5V supply and I/Os work at 5V.

The device is upward compatible with the ST10F269 device, with the following set of differences:

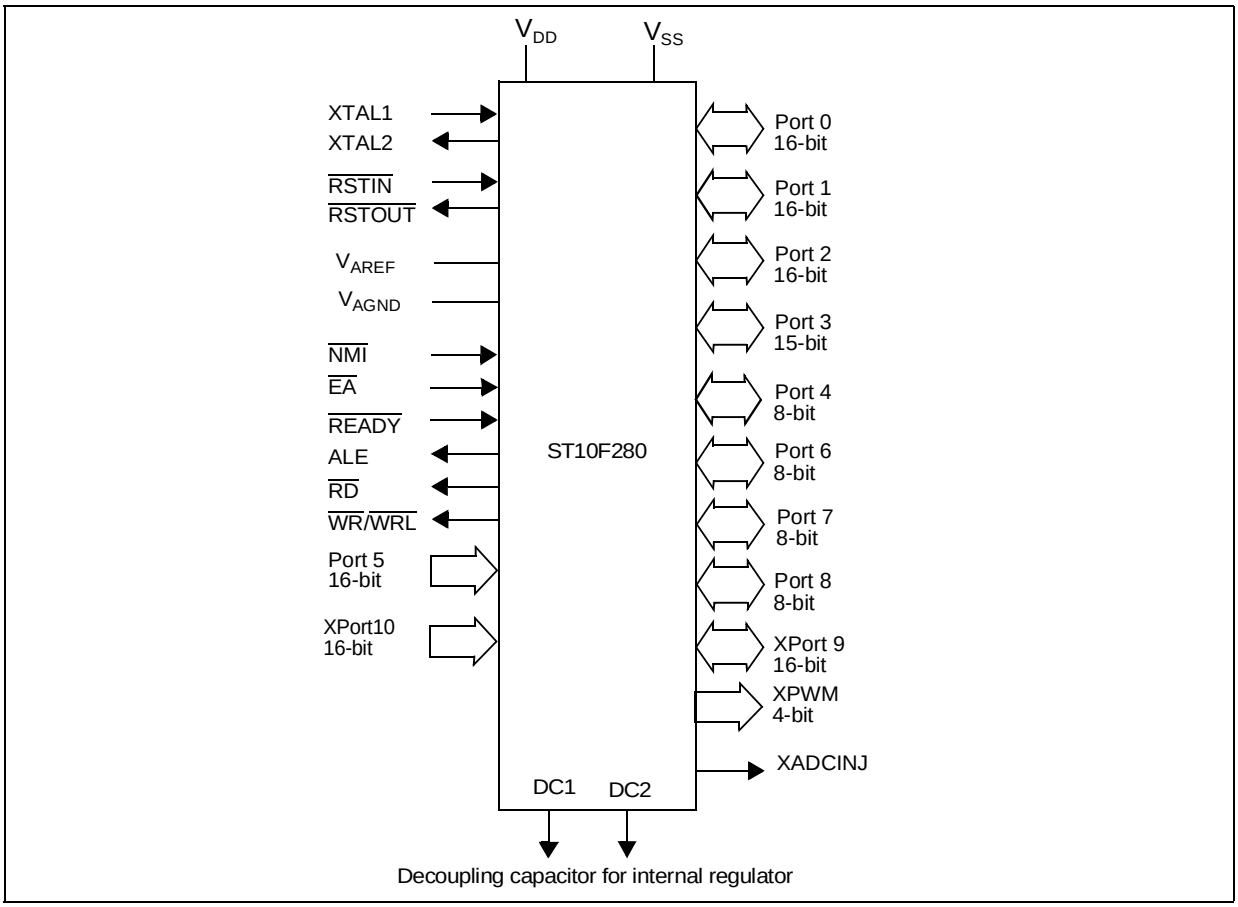
– Two supply pins (DC1,DC2) on the PBGA-208 package are used for decoupling the internally generated 3.3V core logic supply. Do not connect these two pins to 5.0V external supply. Instead, these pins should be connected to a decoupling capacitor (ceramic type, value ≥ 330nF).

– The A/D Converter characteristics stay identical but 16 new input channel are added. A bit in a new register (XADCMUX) control the multiplexage between the first block of 16 channel (on Port5) and the second block (on XPort10). The conversion result registers stay identical and the software management can determine the block in use. A new dedicated timer controls now the ADC channel injection mode on the input CC31 (P7.7). The output of this timer is visible on a dedicated pin (XADCINJ) to emulate this new functionnality.

– A second XPWM peripheral (4 new channels) is added. Four dedicated pins are reserved for the outputs (XPWM[0:3])

– A new general purpose I/O port named XPORT9 (16 bits) is added. Due to the bit addressing management, it will be different from other standard general purpose I/O ports.

Figure 1 : Logic Symbol



2 - BALL DATA

The ST10F280 package is a PBGA of 23 x 23 x 1.96 mm. The pitch of the balls is 1.27 mm. The signal assignment of the 208 balls is described in Figure 2 for the configuration and in Table 1 for the ball signal assignment. This package has 25 additional thermal balls.

Figure 2 : Ball 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 **DC2**	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 $\overline{READY}$	J17 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 P0.2	H15 P0.1	H16 P0.0	H17 $\overline{EA}$	H
G	G1 **DC1**	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 P0.5	G15 P0.4	G16 P0.3	G17 **V_{DD}**	G
F	F1 **V_{SS}**	F2 P8.2	F3 P8.1	F4 P6.6										F14 P0.10	F15 P0.8	F16 P0.6	F17 **V_{SS}**	F
E	E1 **V_{DD}**	E2 P8.0	E3 P6.5	E4 P6.0										E14 P0.15	E15 P0.12	E16 P0.9	E17 P0.7	E
D	D1 P6.7	D2 P6.4	D3 P6.1	D4 xpwm.0	D5 **V_{SS}**	D6 **V_{SS}**	D7 P1.13	D8 P1.9	D9 P1.6	D10 P1.2	D11 XP9.14	D12 XP9.11	D13 XP9.5	D14 XP9.2	D15 XP9.0	D16 P0.13	D17 P0.11	D
C	C1 P6.3	C2 xpwm.3	C3 xpwm.1	C4 $\overline{NMI}$	C5 P1.14	C6 P1.15	C7 P1.12	C8 P1.8	C9 P1.7	C10 P1.3	C11 P1.0	C12 XP9.13	C13 XP9.10	C14 XP9.6	C15 XP9.3	C16 XP9.1	C17 P0.14	C
B	B1 P6.2	B2 xpwm.2	B3 **V_{SS}**	B4 $\overline{RSTOUT}$	B5 **V_{SS}**	B6 **V_{SS}**	B7 P1.11	B8 **V_{SS}**	B9 **V_{SS}**	B10 P1.4	B11 P1.1	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 P1.10	A8 **V_{SS}**	A9 **V_{DD}**	A10 P1.5	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 1 : Ball Description

Symbol	Ball Number	Type	Function		
P6.0 – P6.7		I/O	Port 6 is an 8-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. Port 6 outputs can be configured as push/pull or open drain drivers.		
			The following Port 6 pins also serve for alternate functions:		
	E4	O	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
	C1	O	P6.3	$\overline{CS3}$	Chip Select 3 Output
	D2	O	P6.4	$\overline{CS4}$	Chip Select 4 Output
	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 – P8.7		I/O	Port 8 is an 8-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. Port 8 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 8 is selectable (TTL or special).		
			The following Port 8 pins also serve for alternate functions:		
	E2	I/O	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
	H2	I/O	P8.7	CC23IO	CAPCOM2: CC23 Capture Input / Compare Output
P7.0 – P7.7		I/O	Port 7 is an 8-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. Port 7 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 7 is selectable (TTL or special).		
			The following Port 7 pins also serve for alternate functions:		
	J4	O	P7.0	POUT0	PWM Channel 0 Output
	J3	O	P7.1	POUT1	PWM Channel 1 Output
	J2	O	P7.2	POUT2	PWM Channel 2 Output
	J1	O	P7.3	POUT3	PWM 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

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function		
XP10.0 – XP10.15		I	XPort 10 is a 16-bit input-only port with Schmitt-Trigger characteristics.		
			The pins of XPort10 also serve as the analog input channels (up to 16) for the A/D converter, where XP10.X equals ANx (Analog input channel x).		
	M4	I	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	I	XP10.12		
	R1	I	XP10.13		
	T1	I	XP10.14		
	U1	I	XP10.15		
P5.0 – P5.15		I	Port 5 is a 16-bit input-only port with Schmitt-Trigger characteristics.		
			The pins of Port 5 also serve as the analog input channels (up to 16) for the A/D converter, where P5.x equals ANx (Analog input channel x),		
			or they serve as timer inputs:		
	T2	I	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

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function		
P2.0 – P2.15		I/O	Port 2 is a 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. Port 2 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 2 is selectable (TTL or special). The following Port 2 pins also serve for alternate functions:		
	T7	I/O	P2.0	CC0IO	CAPCOM: CC0 Capture Input / Compare Output
	P8	I/O	P2.1	CC1IO	CAPCOM: CC1 Capture Input / Compare Output
	R8	I/O	P2.2	CC2IO	CAPCOM: CC2 Capture Input / Compare Output
	T8	I/O	P2.3	CC3IO	CAPCOM: CC3 Capture Input / Compare Output
	T9	I/O	P2.4	CC4IO	CAPCOM: CC4 Capture Input / Compare Output
	P9	I/O	P2.5	CC5IO	CAPCOM: CC5 Capture Input / Compare Output
	R9	I/O	P2.6	CC6IO	CAPCOM: CC6 Capture Input / Compare Output
	U9	I/O	P2.7	CC7IO	CAPCOM: CC7 Capture Input / Compare Output
	T10	I/O	P2.8	CC8IO	CAPCOM: CC8 Capture Input / Compare Output,
		I		EX0IN	Fast External Interrupt 0 Input
	R10	I/O	P2.9	CC9IO	CAPCOM: CC9 Capture Input / Compare Output,
		I		EX1IN	Fast External Interrupt 1 Input
	P10	I/O	P2.10	CC10IO	CAPCOM: CC10 Capture Input / Compare Output,
		I		EX2IN	Fast External Interrupt 2 Input
	T11	I/O	P2.11	CC11IO	CAPCOM: CC11 Capture Input / Compare Output,
		I		EX3IN	Fast External Interrupt 3 Input
	R11	I/O	P2.12	CC12IO	CAPCOM: CC12 Capture Input / Compare Output,
		I		EX4IN	Fast External Interrupt 4 Input
	U12	I/O	P2.13	CC13IO	CAPCOM: CC13 Capture Input / Compare Output,
		I		EX5IN	Fast External Interrupt 5 Input
	P11	I/O	P2.14	CC14IO	CAPCOM: CC14 Capture Input / Compare Output,
		I		EX6IN	Fast External Interrupt 6 Input
	T12	I/O	P2.15	CC15IO	CAPCOM: CC15 Capture Input / Compare Output,
		I		EX7IN	Fast External Interrupt 7 Input
		I	T7IN	CAPCOM2	Timer T7 Count Input

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function		
P3.0 - P3.13, P3.15		I/O	Port 3 is a 15-bit (P3.14 is missing) 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. Port 3 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 3 is selectable (TTL or special). The following Port 3 pins also serve for alternate functions:		
	R12	I	P3.0	T0IN	CAPCOM 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	MRST	SSC Master-Receive / Slave-Transmit I/O
	R16	I/O	P3.9	MTSR	SSC 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	$\overline{\text{BHE}}$ WRH	External Memory High Byte Enable Signal, External Memory High Byte Write Strobe
	M14	I/O	P3.13	SCLK	SSC Master Clock Output / Slave Clock Input
	T17	O	P3.15	CLKOUT	System Clock Output (=CPU Clock)
P4.0 – P4.7		I/O	Port 4 is an 8-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. The input threshold is selectable (TTL or special). P4.6 & P4.7 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:		
	N16	O	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
	L16	O	P4.5	A21	Segment Address Line
		I		CAN1_RxD	CAN1 Receive Data Input
	K14	O	P4.6	A22	Segment Address Line, CAN_TxD
		O		CAN1_TxD	CAN1 Transmit Data Output
	K15	O	P4.7	A23	Most Significant Segment Address Line
		O		CAN2_TxD	CAN2 Transmit Data Output

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function
$\overline{RD}$	J14	O	External Memory Read Strobe. $\overline{RD}$ is activated for every external instruction or data read access.
$\overline{WR}$/$\overline{WRL}$	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 accesses 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/ $\overline{READY}$	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 will force 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.
$\overline{EA}$	H17	I	External Access Enable pin. A low level at this pin during and after Reset forces the ST10F280 to begin instruction execution out of external memory. A high level forces execution out of the internal Flash Memory.
PORT0: P0L.0 - P0L.7, P0H.0 - P0H.7		I/O	PORT0 consists of the two 8-bit bidirectional I/O ports P0L and P0H. 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. In case of an external bus configuration, PORT0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: D0 - D7 D0 - D7 P0H.0 – P0H.7: I/O D8 - D15 **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: AD0 - AD7 AD0 - AD7 P0H.0 – P0H.7: A8 - A15 AD8 - AD15
	H16	I/O	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 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function
XPORT9.0 - XPORT9.15		I/O	XPort 9 is a 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.
	D15	I/O	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

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function
PORT1: P1L.0 - P1L.7, P1H.0 - P1H.7		I/O	PORT1 consists of the two 8-bit bidirectional I/O ports P1L and P1H. 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. PORT1 is used as the 16-bit address bus (A) in demultiplexed bus modes and also after switching from a demultiplexed bus mode to a multiplexed bus mode. The following PORT1 pins also serve for alternate functions:
	C11	I/O	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 CC24IO CAPCOM2: CC24 Capture Input
	D7	I	P1H.5 CC25IO CAPCOM2: CC25 Capture Input
	C5	I	P1H.6 CC26IO CAPCOM2: CC26 Capture Input
	C6	I	P1H.7 CC27IO CAPCOM2: CC27 Capture Input
XTAL1	A5	I	XTAL1: Input to the oscillator amplifier and input to the internal clock generator
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 Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the ST10F280. 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 set to a low level when the part is executing either a hardware, a software or a 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 SYSCON register, when the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the ST10F280 to go into power down mode. If $\overline{\text{NMI}}$ is high and PWDCFG ='0', when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally.

Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function
XPWM.0	D4	O	XPWM Channel 0 Output
XPWM.1	C3	O	XPWM Channel 1 Output
XPWM.2	B2	O	XPWM Channel 2 Output
XPWM.3	C2	O	XPWM Channel 3 Output
XADCINJ	L3	O	Output trigger for ADC channel injection
V_{AREF}	U2	-	Reference voltage for the A/D converter.
V_{AGND}	U3	-	Reference ground for the A/D converter.
RPD	M17	I/O	Timing pin for the return from powerdown circuit and synchronous/asynchronous reset selection.
DC1	G1	O	3.3V Decoupling pin: a decoupling capacitor of ~330 nF must be connected between this pin and nearest V_{SS} pin.
DC2	U11	O	3.3V Decoupling pin: a decoupling capacitor of ~330 nF must be connected between this pin and V_{SS} nearest pin.
V_{DD}	A2 A9 A12 A14 E1 K1 U8 U15 P17 L17 G17	-	Digital Supply Voltage: + 5 V during normal operation, idle mode and power down mode

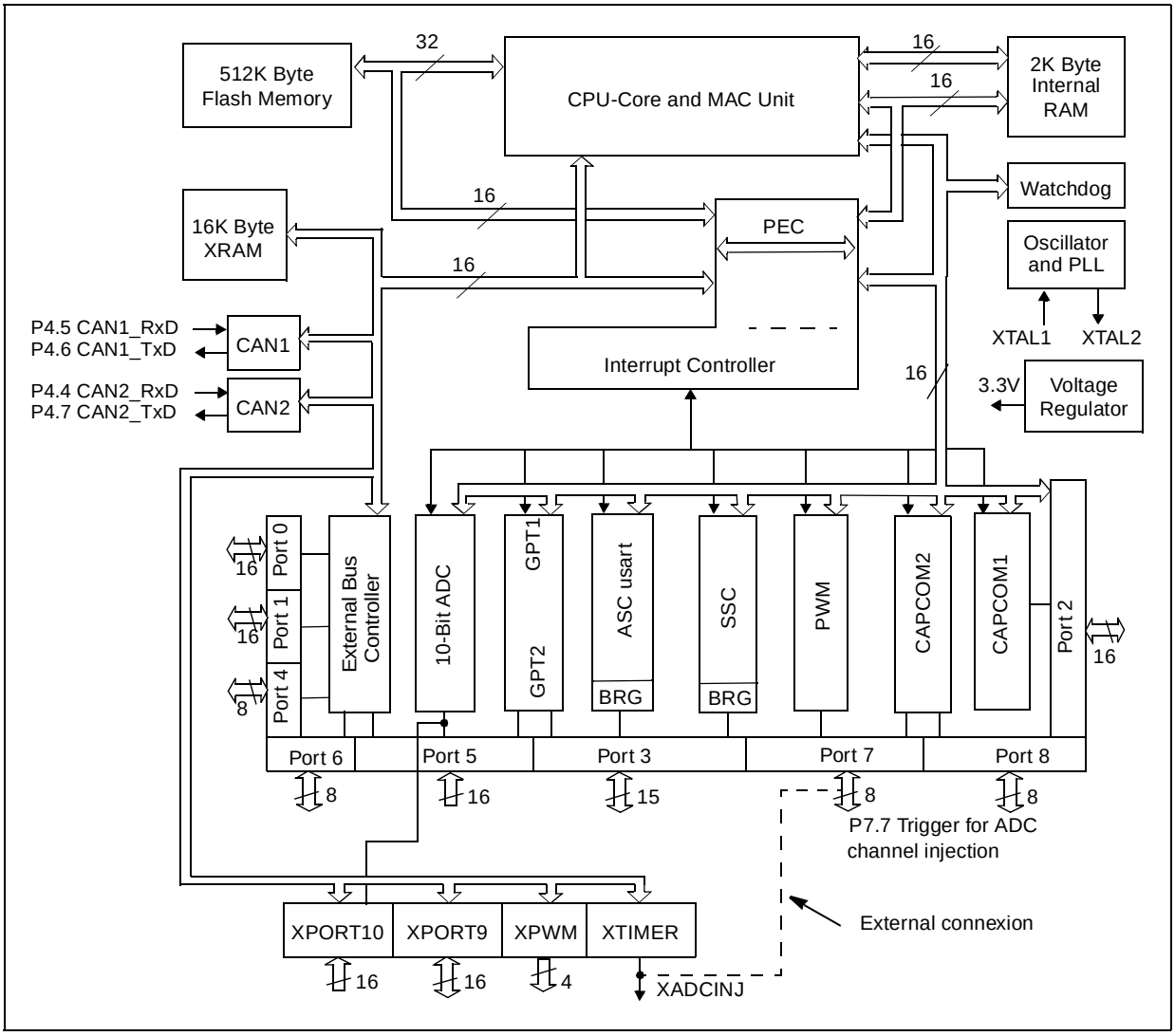
Table 1 : Ball Description (continued)

Symbol	Ball Number	Type	Function
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 ST10F280 combines advantages of both RISC and CISC processors and an advanced peripheral subsystem. The block diagram gives an overview of the different on-chip components and the high bandwidth internal bus structure of the ST10F280.

Figure 3 : Block Diagram



4 - MEMORY ORGANIZATION

The memory space of the ST10F280 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 16M Bytes. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

FLASH: 512K Bytes of on-chip single voltage FLASH memory.

IRAM: 2K Bytes of on-chip internal RAM (dual-port) is provided as a storage for data, system stack, general purpose register banks and code. The register bank can consist of up to 16 wordwide (R0 to R15) and/or bytewide (RL0, RH0, …, RL7, RH7) general purpose registers. Base address is 00'F600h, upper address is 00'FDFFh.

XRAM: 16K Bytes of on-chip extension RAM (single port XRAM) is provided as a storage for data, user stack and code. The XRAM is a single bank, connected to the internal XBUS and are accessed like an external memory in 16-bit demultiplexed bus-mode without waitstate or read/write delay (50ns access at 40MHz CPU clock). Byte and word access is allowed.

The XRAM address range is 00'8000h - 00'BFFFh if enabled (XPEN set bit 2 of SYSCON register-, and XRAMEN set bit 2 of XPERCON register-). If bit XRAMEN or XPEN is cleared, then any access in the address range 00'8000h 00'BFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register
As the XRAM appears like external memory, it cannot be used for the ST10F280's system stack or register banks. The XRAM is not provided for single bit storage and therefore is not bit addressable.

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

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

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

In order to meet the needs of designs where more memory is required than is provided on chip, up to 16M Bytes of external RAM and/or ROM can be connected to the microcontroller. If one or the two CAN modules are used, Port 4 can not be programmed to output all 8 segment address lines. Thus, only 4 segment address lines can be used, reducing the external memory space to 5M Bytes (1M Byte per $\overline{CS}$ line).

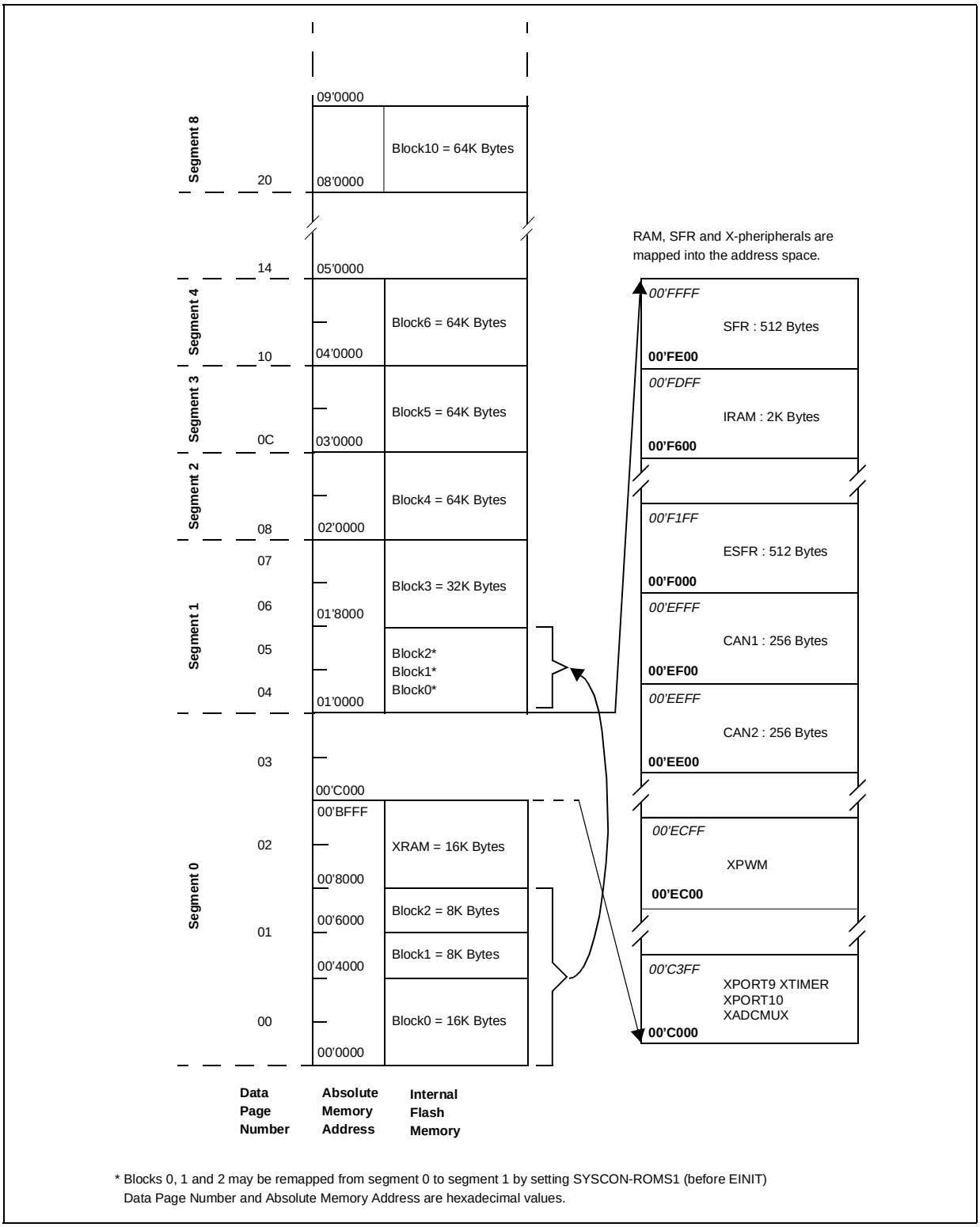
XPWM: Address range 00'EC00h 00'ECFFh is reserved for the XPWM Module access. The XPWM is enabled by setting XPEN bit 2 of the SYSCON register and bit 4 of the new XPERCON register. Accesses to the XPWM Module use demultiplexed addresses and a 16-bit data bus (byte accesses are possible). Two waitstates give an access time of 100 ns at 40MHz CPU clock. No tristate waitstate is used.

XPORT9, XTIMER, XPORT10, XADCMUX : Address range 00'C000h 00'C3FFh is reserved for the XPORT9, XPORT10, XTIMER and XADCMUX peripherals access. The XPORT9, XTIMER, XPORT10, XADCMUX are enabled by setting XPEN bit 2 of the SYSCON register and the bit 3 of the new XPERCON register. Accesses to the XPORT9, XTIMER, XPORT10 and XADCMUX modules use a 16-bit demultiplexed bus mode without waitstate or read/write delay (50ns access at 40MHz CPU clock). Byte and word access is allowed.

Visibility of XBUS Peripherals

The XBUS peripherals can be separately selected for being visible to the user by means of corresponding selection bits in the XPERCON register. If not selected (not activated with XPERCON bit) before the global enabling with XPEN-bit in SYSCON register, the corresponding address space, port pins and interrupts are not occupied by the peripheral, thus the peripheral is not visible and not available. SYSCON register is described in Section 19.2 - System Configuration Registers.

Figure 4 : ST10F280 On-chip Memory Mapping



* Blocks 0, 1 and 2 may be remapped from segment 0 to segment 1 by setting SYSCON-ROMS1 (before EINIT)
Data Page Number and Absolute Memory Address are hexadecimal values.

XPERCON (F024h / 12h) ESFR Reset Value: - - 05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	XPWMEN	XPERCONEN3	XRAMEN	CAN2EN	CAN1EN
											RW	RW	RW	RW	RW

Bit	Function
CAN1EN	**CAN1 Enable Bit** 0 Accesses to the on-chip CAN1 XPeripheral and its functions are disabled. P4.5 and P4.6 pins can be used as general purpose I/Os. Address range 00'EF00h-00'EFFFh is only directed to external memory if CAN2EN and XPWM bits are cleared also. 1 The on-chip CAN1 XPeripheral is enabled and can be accessed.
CAN2EN	**CAN2 Enable Bit** 0 Accesses to the on-chip CAN2 XPeripheral and its functions are disabled. P4.4 and P4.7 pins can be used as general purpose I/Os. Address range 00'EE00h-00'EEFFh is only directed to external memory if CAN1EN and XPWM bits are cleared also. 1 The on-chip CAN2 XPeripheral is enabled and can be accessed.
XRAMEN	**XRAM Enable Bit** 0 Accesses to the on-chip 16K Byte XRAM are disabled, external access performed. 1 The on-chip 16K Byte XRAM is enabled and can be accessed.
XPERCONEN3	**XPORT9, XTIMER, XPORT10, XADCMUX Enable Bit** 0 Accesses to the XPORT9, XTIMER, XPORT10, XADCMUX peripherals are disabled, external access performed. 1 The on-chip XPORT9, XTIMER, XPORT10, XADCMUX peripherals are enabled and can be accessed.
XPWMEN	**XPWM Enable Bit** 0 Accesses to the on-chip XPWM are disabled, external access performed. Address range 00'EC00h-00'ECFFh is only directed to external memory if CAN1EN and CAN2EN are '0' also 1 The on-chip XPWM is enabled and can be accessed.

Note: - When both CAN and XPWM are disabled via XPERCON setting, then any access in the address range 00'EC00h 00'EFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register. P4.4 and P4.7 can be used as General Purpose I/O when CAN2 is not enabled, and P4.5 and P4.6 can be used as General Purpose I/O when CAN1 is not enabled.

- The default XPER selection after Reset is : XCAN1 is enabled, XCAN2 is disabled, XRAM is enabled, XPORT9, XTIMER, XPORT10, XPWM, XADCMUX are disabled.

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

5 - INTERNAL FLASH MEMORY

5.1 - Overview

– 512K Byte on-chip Flash memory

– Two possibilities of Flash mapping into the CPU address space

– Flash memory can be used for code and data storage

– 32-bit, zero waitstate read access (50ns cycle time at f_{CPU} = 40MHz)

– Erase-Program Controller (EPC) similar to M29F400B STM's stand-alone Flash memory

  • Word-by-Word Programmable (16µs typical)

  • Data polling and Toggle Protocol for EPC Status

  • Internal Power-On detection circuit

– Memory Erase in blocks

  • One 16K Byte, two 8K Byte, one 32K Byte, seven 64K Byte blocks

  • Each block can be erased separately (1.5 second typical)

  • Chip erase (8.5 second typical)

  • Each block can be separately protected against programming and erasing

  • Each protected block can be temporary unprotected

  • When enabled, the read protection prevents access to data in Flash memory using a program running out of the Flash memory space. Access to data of internal Flash can only be performed with an inner protected program

– Erase Suspend and Resume Modes

  • Read and Program another Block during erase suspend

– Single Voltage operation , no need of dedicated supply pin

– Low Power Consumption:

  • 45mA max. Read current

  • 60mA max. Program or Erase current

  • Automatic Stand-by-mode (50µA maximum)

– 100,000 Erase-Program Cycles per block, 20 year data retention time

– Operating temperature: -40 to +125^{o}C

5.2 - Operational Overview

Read Mode

In standard mode (the normal operating mode) the Flash appears like an on-chip ROM with the same timing and functionality. The Flash module offers a fast access time, allowing zero waitstate access with CPU frequency up to 40MHz. Instruction fetches and data operand reads are performed with all addressing modes of the ST10F280 instruction set.

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

Table 2 : 512K Byte Flash Memory Block Organisation

Block	Addresses (Segment 0)	Addresses (Segment 1)	Size (K Byte)
0	00'0000h to 00'3FFFh	01'0000h to 01'3FFFh	16
1	00'4000h to 00'5FFFh	01'4000h to 01'5FFFh	8
2	00'6000h to 00'7FFFh	01'6000h to 01'7FFFh	8
3	01'8000h to 01'FFFFh	01'8000h to 01'FFFFh	32
4	02'0000h to 02'FFFFh	02'0000h to 02'FFFFh	64
5	03'0000h to 03'FFFFh	03'0000h to 03'FFFFh	64
6	04'0000h to 04'FFFFh	04'0000h to 04'FFFFh	64
7	05'0000h to 05'FFFFh	05'0000h to 05'FFFFh	64
8	06'0000h to 06'FFFFh	06'0000h to 06'FFFFh	64
9	07'0000h to 07'FFFFh	07'0000h to 07'FFFFh	64
10	08'0000h to 08'FFFFh	08'0000h to 08'FFFFh	64

Instructions and Commands

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

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

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

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

Notes: 1. As it is not possible to perform write operations in the Flash while fetching code from Flash, the Flash commands must be written by instructions executed from internal RAM or external memory.

2. Command write cycles do not need to be consecutively received, pauses are allowed, save for Block Erase command. During this operation all Erase Confirm commands must be sent to complete any block erase operation before time-out period expires (typically 96µs). Command sequencing must be followed exactly. Any invalid combination of commands will reset the Command Interface to Read Mode.

Status Register

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

Erase Operation

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

Erase Suspend

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

In-System Programming

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

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

Read/Write Protection

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

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

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

Power Supply, Reset

The Flash module uses a single power supply for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations from 5V supply. Once a program or erase cycle has been completed, the device resets to the standard read mode. At power-on, the Flash memory has a setup phase of some microseconds (dependent on the power supply ramp-up). During this phase, Flash can not be read. Thus, if $\overline{EA}$ pin is high (execution will start from Flash memory), the CPU will remains in reset state until the Flash can be accessed.

5.3 - Architectural Description

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

5.3.1 - Read Mode

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

– After Reset command

– After every completed erase operation

– After every completed programming operation

– After every other completed command execution

– Few microseconds after a CPU-reset has started

– After incorrect address and data values of command sequences or writing them in an improper sequence

– After incorrect write access to a read protected Flash memory

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

– erase one or several blocks

– program a word into Flash array

– protect / temporary unprotect a block.

In the standard read mode read accesses are directly controlled by the Flash memory array, delivering a 32-bit double Word from the addressed position. Read accesses are always aligned to double Word boundaries. Thus, both low order address bit A1 and A0 are not used in the Flash array for read accesses. The high order address bit A18/A17/A16 define the physical 64K Bytes segment being accessed within the Flash array.

5.3.2 - Command Mode

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

– The Status Register,

– The Ready/Busy signal.

5.3.3 - Flash Status Register

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

Flash Status (see note for address)

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

FSB.7	**Flash Status bit 7: Data Polling Bit** Programming Operation: this bit outputs the complement of the bit 7 of the word being programmed, and after completion, will output the bit 7 of the word programmed. Erasing Operation: outputs a ‘0’ during erasing, and ‘1’ after erasing completion. If the block selected for erasure is (are) protected, FSB.7 will be set to ‘0’ for about 100 µs, and then return to the previous addressed memory data value. FSB.7 will also flag the Erase Suspend Mode by switching from ‘0’ to ‘1’ at the start of the Erase Suspend. During Program operation in Erase Suspend Mode, FSB.7 will have the same behaviour as in normal Program execution outside the Suspend mode.
FSB.6	**Flash Status bit 6: Toggle Bit** Programming or Erasing Operations: successive read operations of Flash Status register will deliver complementary values. FSB.6 will toggle each time the Flash Status register is read. The Program operation is completed when two successive reads yield the same value. The next read will output the bit last programmed, or a ‘1’ after Erase operation FSB.6 will be set to‘1’ if a read operation is attempted on an Erase Suspended block. In addition, an Erase Suspend/Resume command will cause FSB.6 to toggle.
FSB.5	**Flash Status bit 5: Error Bit** This bit is set to ‘1’ when there is a failure of Program, block or chip erase operations.This bit will also be set if a user tries to program a bit to ‘1’ to a Flash location that is currently programmed with ‘0’. The error bit resets after Read/Reset instruction. In case of success, the Error bit will be set to ‘0’ during Program or Erase and then will output the bit last programmed or a ‘1’ after erasing
FSB.3	**Flash Status bit 3: Erase Time-out Bit** This bit is cleared by the EPC when the last Block Erase command has been entered to the Command Interface and it is awaiting the Erase start. When the time-out period is finished, after 96 µs, FSB.3 returns back to ‘1’.
FSB.2	**Flash Status bit 2: Toggle Bit** This toggle bit, together with FSB.6, can be used to determine the chip status during the Erase Mode or Erase Suspend Mode. It can be used also to identify the block being Erased Suspended. A Read operation will cause FSB.2 to Toggle during the Erase Mode. If the Flash is in Erase Suspend Mode, a Read operation from the Erase suspended block or a Program operation into the Erase suspended block will cause FSB.2 to toggle. When the Flash is in Program Mode during Erase Suspend, FSB.2 will be read as ‘1’ if address used is the address of the word being programmed. After Erase completion with an Error status, FSB.2 will toggle when reading the faulty sector.

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

5.3.4 - Flash Protection Register

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

Flash Protection Register (PR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP	-	-	-	-	**BP10**	**BP9**	**BP8**	**BP7**	**BP6**	**BP5**	**BP4**	**BP3**	**BP2**	**BP1**	**BP0**
RW					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

BPx	**Block x Protection bit (x = 0...10)** '0': the Block Protection is enabled for block x. Programming or erasing the block is not possible, unless a Block Temporary Unprotection command is issued. 1': the Block Protection is disabled for block x. Bit is '1' by default, and can be programmed permanently to '0' using the Set Protection command but then cannot be set to '1' again. It is therefore possible to temporally disable the Block Protection using the Block Temporary Unprotection instruction.
CP	**Code Protection Bit** '0': the Flash Code Protection is enabled. Read accesses to the Flash for execution not performed in the Flash itself are not allowed, the returned value will be 009Bh, whatever the content of the Flash is. 1': the Flash Code Protection is disabled: read accesses to the Flash from external or internal RAM are allowed Bit is '1' by default, and can be programmed permanently to '0' using the Set Protection command but then cannot be set to '1' again. It is therefore possible to temporarily disable the Code Protection using the Code Temporary Unprotection instruction.

5.3.5 - Instructions Description

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

Read/Reset (RD). The Read/Reset instruction consist of one write cycle with data XXF0h . it can be optionally preceded by two CI enable coded cycles (data xxA8h at address 1554h + data xx54h at address 2AA8h). Any successive read cycle following a Read/Reset instruction will read the memory array. A Wait cycle of 10µs is necessary after a Read/Reset command if the memory was in program or Erase mode.

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

Block Erase (BE). This instruction uses a minimum of six command cycles. The erase enable command xx80h is written at address 1554h after the two-cycle CI enable sequence.

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

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

If the second command given is not an erase confirm or if the coded cycles are wrong, the instruction aborts, and the device is reset to Read Mode.

It is not necessary to program the block with 0000h as the EPC will do this automatically before the erasing to FFFFh. Read operations after the EPC has started, output the Flash Status Register. During the execution of the erase by the EPC, the device accepts only the Erase Suspend and Read/Reset instructions. Data Polling bit FSB.7 returns '0' while the erasure is in progress, and '1' when it has completed. The Toggle bit FSB.2 and FSB.6 toggle during the erase operation. They stop when erase is completed. After completion, the Error bit FSB.5 returns '1' if there has been an erase failure because erasure has not completed even after the maximum number of erase cycles have been executed by the EPC, in this case, it will be necessary to input a Read/Reset to the Command Interface in order to reset the EPC.

Chip Erase (CE). This instruction uses six write cycles. The Erase Enable command xx80h, must be written at address 1554h after CI-Enable cycles. The Chip Erase command xx10h must be given on the sixth cycle after a second CI-Enable sequence. An error in command sequence will reset the CI to Read mode. It is NOT necessary to program the block with 0000h as the EPC will do this automatically before the erasing to FFFFh. Read operations after the EPC has started output the Flash Status Register. During the execution of the erase by the EPC, Data Polling bit FSB.7 returns '0' while the erasure is in progress, and '1' when it has completed. The FSB.2 and FSB.6 bit toggle during the erase operation. They stop when erase is finished. The FSB.5 error bit returns "1" in case of failure of the erase operation. The error flag is set after the maximum number of erase cycles have been executed by the EPC. In this case, it will be necessary to input a Read/Reset to the Command Interface in order to reset the EPC.

Erase Suspend (ES). This instruction can be used to suspend a Block Erase operation by giving the command xxB0h without any specific address. No CI-Enable cycles is required. Erase Suspend operation allows reading of data from another block and/or the programming in another block while erase is in progress. If this command is given during the time-out period, it will terminate the time-out period in addition to erase Suspend. The Toggle Bit FSB.6, when monitored at an address that belongs to the block being erased, stops toggling when Erase Suspend Command is effective, It happens between 0.1µs and 15µs after the Erase Suspend Command has been written. The Flash will then go in normal Read Mode, and read from blocks not being erased is valid, while read from block being erased will output FSB.2 toggling. During a Suspend phase the only instructions valid are Erase Resume and Program Word. A Read / Reset instruction during Erase suspend will definitely abort the Erase and result in invalid data in the block being erased.

Erase Resume (ER). This instruction can be given when the memory is in Erase Suspend State. Erase can be resumed by writing the command xx30h at any address without any CI-enable sequence.

Program during Erase Suspend. The Program Word instruction during Erase Suspend is allowed only on blocks that are not Erase-suspended. This instruction is the same than the Program Word instruction.

Set Protection (SP). This instruction can be used to enable both Block Protection (to protect each block independently from accidental Erasing-Programming Operation) and Code Protection (to avoid code dump). The Set Protection Command must be given after a special CI-Protection Enable cycles (see instruction table). The following Write cycle, will program the Protection Register. To protect the block x (x = 0 to 10), the data bit x must be at '0'. To protect the code, bit 15 of the data must be '0'. Enabling Block or Code Protection is **permanent** and can be cleared only by STM. Block Temporary Unprotection and Code Temporary Unprotection instructions are available to allow the customer to update the code.

Note: 1. The new value programmed in protection register will only become active after a reset.

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

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

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

Block Temporary Unprotection (BTU). This Instruction can be used to temporary unprotect all the blocks from Program / Erase protection. The Unprotection is disabled after a Reset cycle. The Block Temporary Unprotection command xxC1h must be given to enable Block Temporary Unprotection. The Command must be preceded by the CI-Protection Enable cycles and followed by the Read/Reset command xxF0h.

Set Code Protection (SCP). This kind of protection allows the customer to protect the proprietary code written in Flash. If installed and active, Flash Code Protection prevents data operand accesses and program branches into the on-chip Flash area from any location outside the Flash memory itself. Data operand accesses and branches to Flash locations are only and exclusively allowed for instructions executed from the Flash memory itself. Every read or jump to Flash performed from another memory (like internal RAM, external memory) while Code Protection is enabled, will give the opcode 009Bh related to TRAP #00 illegal instruction. The CI-Protection Enable cycles must be sent to set the Code Protection. By writing data 7FFFh at any odd word address, the Code Protected status is stored in the Flash Protection Register (PR). Protection is permanent and cannot be cleared by the user. It is possible to temporarily disable the Code Protection using Code Temporary Unprotection instruction.

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

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

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

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

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

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

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

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

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

Table 3 : Instructions

Instruction	Mne	Cycle		1st Cycle	2nd Cycle	3rd Cycle	4th Cycle	5th Cycle	6th Cycle	7th Cycle
Read/Reset	RD	1+	Addr.[1]	X [2]	Read Memory Array until a new write cycle is initiated					
			Data	xxF0h						
Read/Reset	RD	3+	Addr.[1]	x1554h	x2AA8h	xxxxxh	Read Memory Array until a new write cycle is initiated			
			Data	xxA8h	xx54h	xxF0h				
Program Word	PW	4	Addr.[1]	x1554h	x2AA8h	x1554h	WA [3]	Read Data Polling or Toggle Bit until Program completes.		
			Data	xxA8h	xx54h	xxA0h	WD [4]			
Block Erase	BE	6	Addr.[1]	x1554h	x2AA8h	x1554h	x1554h	x2AA8h	BA	BA' [5]
			Data	xxA8h	xx54h	xx80h	xxA8h	xx54h	xx30h	xx30h
Chip Erase	CE	6	Addr.[1]	x1554h	x2AA8h	x1554h	x1554h	x2AA8h	x1554h	Note [6]
			Data	xxA8h	xx54h	xx80h	xxA8h	xx54h	xx10h	
Erase Suspend	ES	1	Addr.[1]	X[2]	Read until Toggle stops, then read or program all data needed from block(s) not being erased then Resume Erase.					
			Data	xxB0h						
Erase Resume	ER	1	Addr.[1]	X[2]	Read Data Polling or Toggle bit until Erase completes or Erase is supended another time.					
			Data	xx30h						
Set Block/Code Protection	SP	4	Addr.[1]	x2A54h	x15A8h	x2A54h	Any odd word address [9]			
			Data	xxA8h	xx54h	xxC0h	WPR [7]			
Read Protection Status	RP	4	Addr.[1]	x2A54h	x15A8h	x2A54h	Any odd word address [9]	Read Protection Register until a new write cycle is initiated.		
			Data	xxA8h	xx54h	xx90h	Read PR			
Block Temporary Unprotection	BTU	4	Addr.[1]	x2A54h	x15A8h	x2A54h	X[2]			
			Data	xxA8h	xx54h	xxC1h	xxF0h			
Code Temporary Unprotection	CTU	1	Addr.[1]	MEM [8]	Write cycles must be executed from Flash.					
			Data	FFFFh						
Code Temporary Protection	CTP	1	Addr.[1]	MEM [8]	Write cycles must be executed from Flash.					
			Data	FFFBh						

Notes 1. Address bit A14, A15 and above are don't care for coded address inputs.

2. X = Don't Care.

3. WA = Write Address: address of memory location to be programmed.

4. WD = Write Data: 16-bit data to be programmed

5. Optional, additional blocks addresses must be entered within a time-out delay (96 µs) after last write entry, timeout status can be verified through FSB.3 value. When full command is entered, read Data Polling or Toggle bit until Erase is completed or suspended.

6. Read Data Polling or Toggle bit until Erase completes.

7. WPR = Write protection register. To protect code, bit 15 of WPR must be '0'. To protect block N (N=0,1,...), bit N of WPR must be '0'. Bit that are already at '0' in protection register must also be '0' in WPR, else a writing error will occurs (it is not possible to write a '1' in a bit already programmed at '0').

8. MEM = any address inside the Flash memory space. Absolute addressing mode must be used (MOV MEM, Rn), and instruction must be executed from Flash memory space.

9. Odd word address = 4n-2 where n = 0, 1, 2, 3..., ex. 0002h, 0006h...

– Generally, command sequences cannot be written to Flash by instructions fetched from the Flash itself. Thus, the Flash commands must be written by instructions, executed from internal RAM or external memory.
– Command cycles on the CPU interface need not to be consecutively received (pauses allowed). The CPU interface delivers dummy read data for not used cycles within command sequences.
– All addresses of command cycles shall be defined only with **Register-indirect** addressing mode in the according move instructions. Direct addressing is not allowed for command sequences. Address segment or data page pointer are taken into account for the command address value.

5.3.6 - Reset Processing and Initial State

The Flash module distinguishes two kinds of CPU reset types

The lengthening of CPU reset:

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

5.4 - Flash Memory Configuration

The default memory configuration of the ST10F280 Memory is determined by the state of the $\overline{EA}$ pin at reset. This value is stored in the Internal ROM Enable bit (named ROMEN) of the SYSCON register.

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

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

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

Bit ROMS1 only affects the mapping of the first 32K Bytes of the Flash memory. All other parts of the Flash memory (addresses 01’8000h 08’FFFFh) remain unaffected.

The SGTDIS Segmentation Disable / Enable must also be set to 0 to allow the use of the full 512K Bytes of on-chip memory in addition to the external boot memory. The correct procedure on changing the segmentation registers must also be observed to prevent an unwanted trap condition:

– Instructions that configure the internal memory must only be executed from external memory or from the internal RAM.

– An Absolute Inter-Segment Jump (JMPS) instruction must be executed after Flash enabling, to the next instruction, even if this next instruction is located in the consecutive address.

– Whenever the internal Memory is disabled, enabled or remapped, the DPPs must be explicitly (re)loaded to enable correct data accesses to the internal memory and/or external memory.

5.5 - Application Examples

5.5.1 - Handling of Flash Addresses

All command, Block, Data and register addresses to the Flash have to be located within the active Flash memory space. The active space is that address range to which the physical Flash addresses are mapped as defined by the user. When using data page pointer (DPP) for block addresses make sure that address bit A15 and A14 of the block address are reflected in both LSBs of the selected DPPS.

Note: - For Command Instructions, address bit A14, A15, A16, A17 and A18 are don’t care. This simplify a lot the application software, because it minimize the use of DPP registers when using Command in the Command Interface.

- Direct addressing is not allowed for Command sequence operations to the Flash. Only Register-indirect addressing can be used for command, block or write-data accesses.

5.5.2 - Basic Flash Access Control

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

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

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

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

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

Principle example of address generation for Flash commands and registers:

When using data page pointer (DPP0 is this example)

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

When using the extended segment instruction:

```
MOV   Rwm,#ADDRESS       ;ADDRESS could be a dedicated command sequence
                         ;address (2AA8h, 1554h ... ) or the Flash write
                         ;address
MOV   Rwo,#DATA          ;DATA could be a dedicated command sequence data
                         ;(xxA0h,xx80h ... ) or data to be programmed
MOV   Rwn,#SEGMENT       ;the value of SEGMENT represents the segment
                         ;number and could be 0, 1, 2, 3 or 4 (depending
                         ;on sector mapping) for 256KByte Flash.
EXTS  Rwn,#LENGTH        ;the value of Rwn determines the 8-bit segment
                         ;valid for the corresponding data access for any
                         ;long or indirect address in the following(s)
                         ;instruction(s). LENGTH defines the number of
                         ;the effected instruction(s) and has to be a value
                         ;between 1...4
MOV   [Rwm],Rwo          ;indirect addressing with segment number from
                         ;EXTS
```

5.5.3 - Programming Examples

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

Example 1 Performing the command Read/Reset

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

According to the usual way of ST10 data addressing with data page pointers, address bit A15 and A14 of a 16-bit command write address select the data page pointer (DPP) which contains the upper 10-bit for building the 24-bit physical data address. Address bit A13...A0 represent the address offset. As the bit A14...A18 are "don't care" when written a Flash command in the Command Interface (CI), we can choose the most conveniant DPPx register for address handling.

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

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

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

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

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

Example 2 Performing a Program Word command

We assume that in the initialization phase the lowest 32K Bytes of Flash memory (sector 0) have been mapped to segment 1.The data to be written is loaded in register R13, the address to be programmed is loaded in register R11/R12 (segment number in R11, segment offset in R12).

```
MOV   R5, #01554h          ;load auxilary register R5 with command address
                           ;(used in command cycle 1)
MOV   R6, #02AA8h          ;load auxilary register R6 with command address
                           ;(used in command cycle 2)
SXCT  DPPO, #08h           ;push data page pointer 0 and load it to point to
                           ;segment 2
MOV   R7, #0A8h            ;load register R7 with 1st CI enable command
MOV   [R5], R7             ;command cycle 1
MOV   R7, #054h            ;load register R7 with 2cd CI enable command
MOV   [R6], R7             ;command cycle 2
MOV   R7, #0A0h            ;load register R7 with Program Word command
MOV   [R5], R7             ;command cycle 3
```

```
POP   DPP0                ;restore DPP0: following addressing to the Flash
                          ;will use EXTended instructions
                          ;R11 contains the segment to be programmed
                          ;R12 contains the segment offset address to be
                          ;programmed
                          ;R13 contains the data to be programmed
EXTS  R11, #1             ;use EXTended addressing for next MOV instruction
MOV   [R12], R13          ;command cycle 4: the EPC starts execution of
                          ;Programming Command
Data_Polling:
EXTS  R11, #1             ;use EXTended addressing for next MOV instruction
MOV   R7, [R12]           ;read Flash Status register (FSB) in R7
MOV   R6, R7              ;save it in R6 register
                          ;Check if FSB.7 = Data.7 (i.e. R7.7 = R13.7)
XOR   R7, R13
JNB   R7.7, Prog_OK

                          ;Check if FSB.5 = 1 (Programming Error)
JNB   R6.5, Data_Polling
                          ;Programming Error: verify is Flash programmed
                          ;data is OK
EXTS  R11, #1             ;use EXTended addressing for next MOV instruction
MOV   R7, [R12]           ;read Flash Status register (FSB) in R7
                          ;Check if FSB.7 = Data.7
XOR   R7, R13
JNB   R7.7, Prog_OK
                          ;Programming failed: Flash remains in Write
                          ;Operation.
                          ;To go back to normal Read operations, a Read/Reset
                          ;command
                          ;must be performed
Prog_Error:
MOV   R7, #0F0h           ;load register R7 with Read/Reset command
EXTS  R11, #1             ;use EXTended addressing for next MOV instruction
MOV   [R12], R7           ;address is don't care for Read/Reset command
...                       ;here place specific Error handling code
...
...
                          ;When programming operation finished succesfully,
                          ;Flash is set back automatically to normal Read Mode
Prog_OK:
....
....
```

Example 3 Performing the Block Erase command

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

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

                           ;Check if FSB.5 = 1 (Erasing Error)
JNB   R7.5, Erase_Polling
                           ;Programming failed: Flash remains in Write
                           ;Operation.
                           ;To go back to normal Read operations, a Read/Reset
                           ;command
                           ;must be performed
Erase_Error:
MOV   R7, #0F0h            ;load register R7 with Read/Reset command
EXTS  R11, #1              ;use EXTended addressing for next MOV instruction
MOV   [R12], R7            ;address is don't care for Read/Reset command
...                        ;here place specific Error handling code
...
...

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

5.6 - Bootstrap Loader

The built-in bootstrap loader (BSL) of the ST10F280 provides a mechanism to load the startup program through the serial interface after reset. In this case, no external memory or internal Flash memory is required for the initialization code starting at location 00'0000h (see Figure 5).

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

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

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

5.6.1 - Entering the Bootstrap Loader

The ST10F280 enters BSL mode when pin P0L.4 is sampled low at the end of a hardware reset. In this case the built-in bootstrap loader is activated independent of the selected bus mode.

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

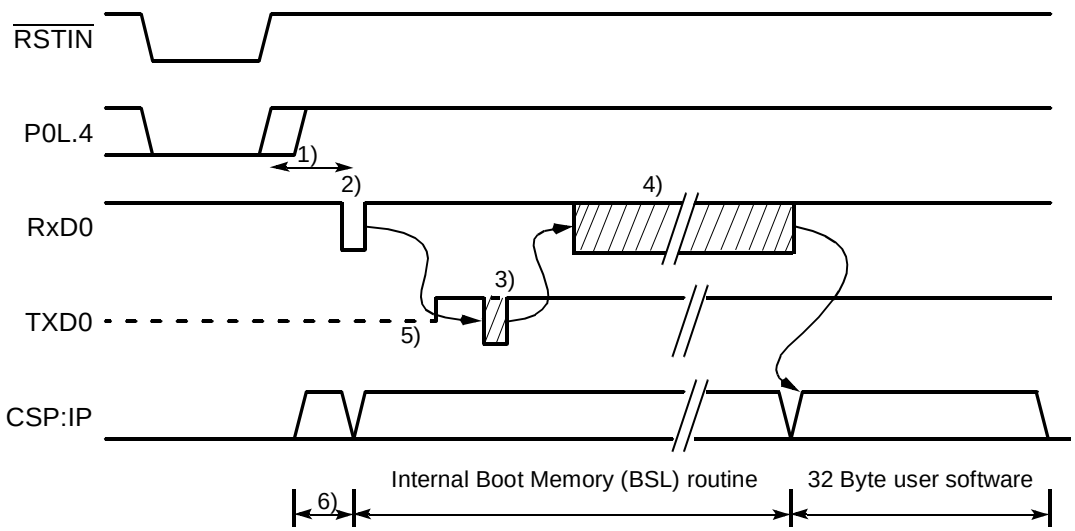
After entering BSL mode and the respective initialization the ST10F280 scans the RxD0 line to receive a zero Byte, one start Bit, eight '0' data Bits and one stop Bit.

From the duration of this zero Byte it calculates the corresponding Baud rate factor with respect to the current CPU clock, initializes the serial interface ASC0 accordingly and switches pin TXD0 to output.

Using this Baud rate, an identification Byte is returned to the host that provides the loaded data.

This identification Byte identifies the device to be booted. Identification byte is D5h for the ST10F280.

Figure 5 : Bootstrap Loader Sequence



1) BSL initialization time

2) Zero Byte (1 start Bit, eight '0' data Bits, 1 stop Bit), sent by host.

3) Identification Byte (D5h), sent by ST10F280.

4) 32 Bytes of code / data, sent by host.

5) Caution: TXD0 is only driven a certain time after reception of the zero Byte.

6) Internal Boot ROM.

When the ST10F280 has entered BSL mode, the following configuration is automatically set (values that deviate from the normal reset values, are ***marked***):

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

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

Pin TXD0 is configured as output, so the ST10F280 can return the identification Byte.

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

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

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

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

5.6.2 - Memory Configuration After Reset

The configuration (and the accessibility) of the ST10F280's memory areas after reset in Bootstrap-Loader mode differs from the standard case. Pin $\overline{EA}$ is not evaluated when BSL mode is selected, and accesses to the internal Flash area are partly redirected, while the ST10F280 is in BSL mode (see Figure 7). All code fetches are made from the special Boot-ROM, while data accesses read from the internal user Flash. Data accesses will return undefined values on ROMless devices.

The code in the Boot-ROM is not an invariant feature of the ST10F280. User software should not try to execute code from the internal Flash area while the BSL mode is still active, as these fetches will be redirected to the Boot-ROM. The Boot-ROM will also "move" to segment 1, when the internal Flash area is mapped to segment 1 (see Figure 7).

Figure 6 : Hardware Provisions to Activate the BSL

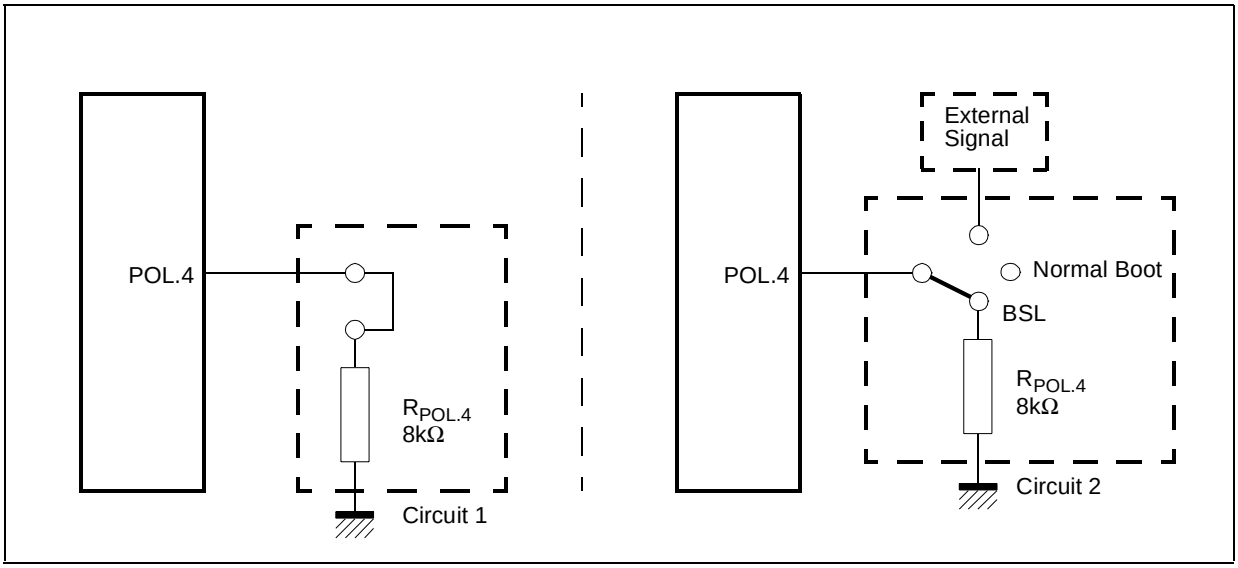
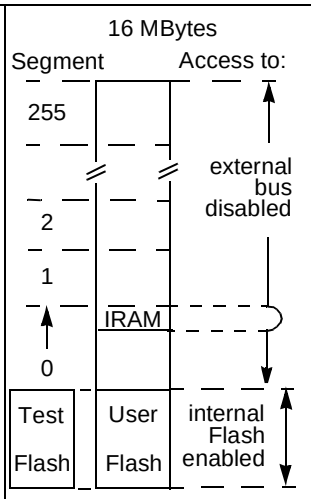
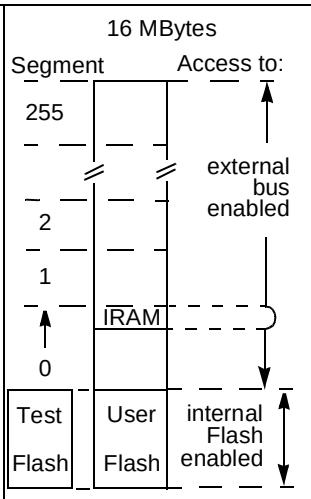
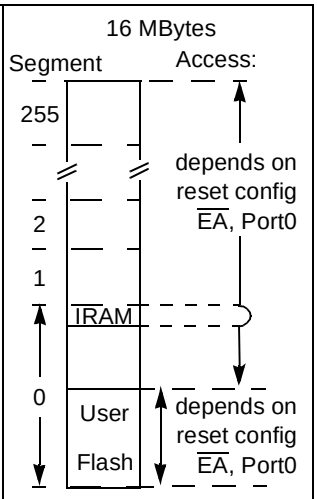


Figure 7 : Memory Configuration After Reset

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

5.6.3 - Loading the Startup Code

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

The bootstrap loading sequence is now terminated, the ST10F280 remains in BSL mode, however. Most probably the initially loaded routine will load additional code or data, as an average application is likely to require substantially more than 16 instructions. This second receive loop may directly use the pre-initialized interface ASC0 to receive data and store it to arbitrary user-defined locations.

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

This process may go through several iterations or may directly execute the final application. In all cases the ST10F280 will still run in BSL mode, that means with the watchdog timer disabled and limited access to the internal Flash area.

All code fetches from the internal Flash area (00'0000h...00'7FFFh or 01'0000h...01'7FFFh, if mapped to segment 1) are redirected to the special Boot-ROM. Data fetches access will access the internal Boot-ROM of the ST10F280, if any is available, but will return undefined data on ROMless devices.

5.6.4 - Exiting Bootstrap Loader Mode

In order to execute a program in normal mode, the BSL mode must be terminated first. The ST10F280 exits BSL mode upon a software reset (ignores the level on P0L.4) or a hardware reset (P0L.4 must be high). After a reset the ST10F280 will start executing from location 00'0000h of the internal Flash or the external memory, as programmed via pin $\overline{EA}$.

5.6.5 - Choosing the Baud Rate for the BSL

The calculation of the serial Baud rate for ASC0 from the length of the first zero Byte that is received, allows the operation of the bootstrap loader of the ST10F280 with a wide range of Baud rates. However, the upper and lower limits have to be kept, in order to insure proper data transfer.

$$B_{ST10F280} = \frac{f_{CPU}}{32 \times (S0BRL + 1)}$$

The ST10F280 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. The formula below shows the association:

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

For a correct data transfer from the host to the ST10F280 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 ST10F280 Baud rate can be calculated via the formula below:

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

$$F_B \leq 2.5 \ \%$$

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

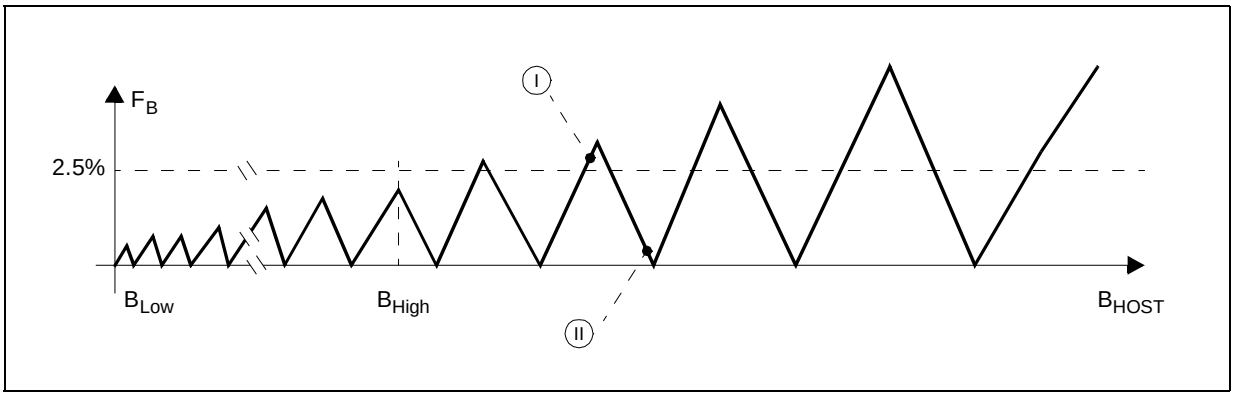
This Baud rate deviation is a nonlinear function depending on the CPU clock and the Baud rate of the host. The maxima of the function (F_B) increase with the host Baud rate due to the smaller Baud rate pre-scaler factors and the implied higher quantization error (see Figure 8).

The minimum Baud rate (B_{Low} in the Figure 8) is determined by the maximum count capacity of timer T6, when measuring the zero Byte, and it depends on the CPU clock. Using the maximum T6 count 2^{16} in the formula the minimum Baud rate can be calculated. The lowest standard Baud rate in this case would be 1200 Baud. Baud rates below B_{Low} would cause T6 to overflow. In this case ASC0 cannot be initialized properly.

The maximum Baud rate (B_{High} in the Figure 8) is the highest Baud rate where the deviation still does not exceed the limit, so all Baud rates between B_{Low} and B_{High} are below the deviation limit. The maximum standard Baud rate that fulfills this requirement is 19200 Baud.

Higher Baud rates, however, may be used as long as the actual deviation does not exceed the limit. A certain Baud rate (marked 'I' in Figure 8) may violate the deviation limit, while an even higher Baud rate (marked 'II' in Figure 8) stays very well below it. This depends on the host interface.

Figure 8 : Baud Rate Deviation Between Host and ST10F280



6 - CENTRAL PROCESSING UNIT (CPU)

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

Most of the ST10F280's instructions can be executed in one instruction cycle which requires 50ns at 40MHz CPU clock. For example, shift and rotate instructions are processed in one instruction cycle independent of the number of bits to be shifted.

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

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

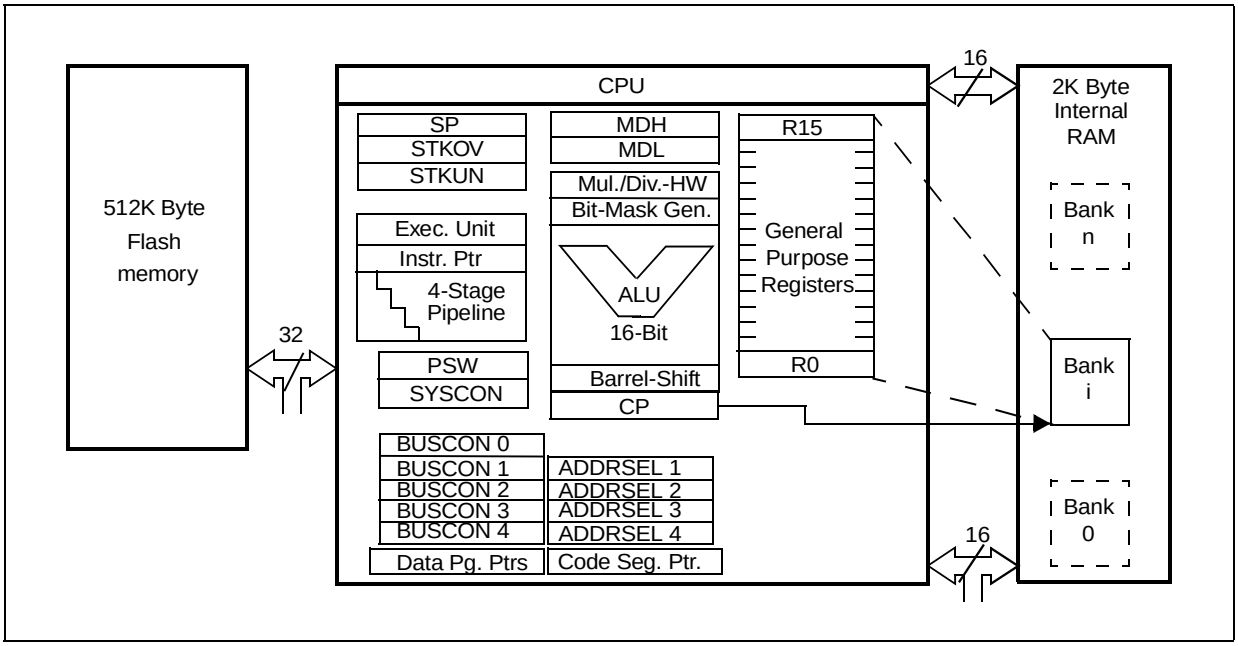
The CPU uses a bank of 16 word registers to run the current context. This bank of General Purpose Registers (GPR) is physically stored within the on-chip Internal RAM (IRAM) area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU.

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

A system stack of up to 1024 bytes is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register.

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

Figure 9 : CPU Block Diagram (MAC Unit not included)



The System Configuration Register SYSCON

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

SYSCON (FF12h / 89h) SFR Reset Value: 0xx0h

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

Notes: 1. These bit are set directly or indirectly according to PORT0 and $\overline{EA}$ pin configuration during reset sequence.
2. Register SYSCON cannot be changed after execution of the EINIT instruction.

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

6.1 - Multiplier-accumulator Unit (MAC)

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

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

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

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

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

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

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

6.1.1 - Features

6.1.1.1 - Enhanced Addressing Capabilities

– New addressing modes including a double indirect addressing mode with pointer post-modification.

– Parallel Data Move : this mechanism allows one operand move during Multiply-Accumulate instructions without penalty.

– New tranfer instructions CoSTORE (for fast access to the MAC SFRs) and CoMOV (for fast memory to memory table transfer).

6.1.1.2 - Multiply-Accumulate Unit

– One-cycle execution for all MAC operations.

– 16 x 16 signed/unsigned parallel multiplier.

– 40-bit signed arithmetic unit with automatic saturation mode.

– 40-bit accumulator.

– 8-bit left/right shifter.

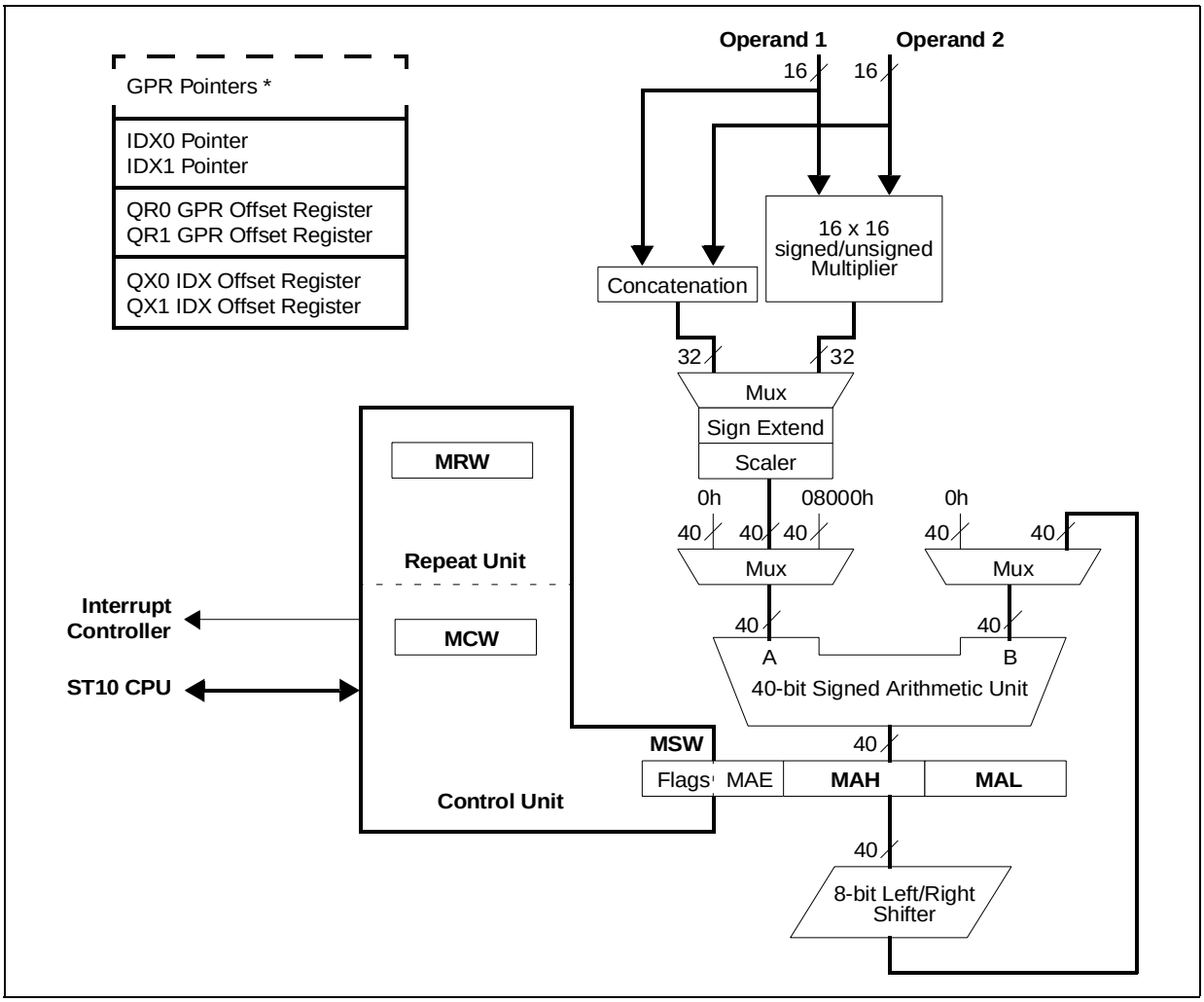
– Full instruction set with multiply and multiply-accumulate, 32-bit signed arithmetic and compare instructions.

6.1.1.3 - Program Control

– Repeat Unit : allows some MAC co-processor instructions to be repeated up to 8192 times. Repeated instructions may be interrupted.

– MAC interrupt (Class B Trap) on MAC condition flags.

Figure 10 : MAC Unit Architecture



*Note: * Shared with standard ALU.*

6.2 - Instruction Set Summary

The Table 4 lists the instructions of the ST10F280. The various addressing modes, instruction operation, parameters for conditional execution of instructions, opcodes and a detailed description of each instruction can be found in the “ST10 Family Programming Manual”.

Table 4 : Instruction Set Summary

Mnemonic	Description	Bytes
ADD(B)	Add word (byte) operands	2 / 4
ADDC(B)	Add word (byte) operands with Carry	2 / 4
SUB(B)	Subtract word (byte) operands	2 / 4
SUBC(B)	Subtract word (byte) operands with Carry	2 / 4
MUL(U)	(Un)Signed multiply direct GPR by direct GPR (16-16-bit)	2
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	Bitwise AND, (word/byte operands)	2 / 4
OR(B)	Bitwise OR, (word/byte operands)	2 / 4
XOR(B)	Bitwise XOR, (word/byte operands)	2 / 4
BCLR	Clear direct bit	2
BSET	Set direct bit	2
BMOV(N)	Move (negated) direct bit to direct bit	4
BAND, BOR, BXOR	AND/OR/XOR direct bit with direct bit	4
BCMP	Compare direct bit to direct bit	4
BFLDH/L	Bitwise modify masked high/low byte of bit-addressable direct word memory with immediate data	4
CMP(B)	Compare word (byte) operands	2 / 4
CMPD1/2	Compare word data to GPR and decrement GPR by 1/2	2 / 4
CMPI1/2	Compare word data to GPR and increment GPR by 1/2	2 / 4
PRIOR	Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR	2
SHL / SHR	Shift left/right direct word GPR	2
ROL / ROR	Rotate left/right direct word GPR	2
ASHR	Arithmetic (sign bit) shift right direct word GPR	2
MOV(B)	Move word (byte) data	2 / 4
MOVBS	Move byte operand to word operand with sign extension	2 / 4
MOVBZ	Move byte operand to word operand with zero extension	2 / 4
JMPA, JMPI, JMPR	Jump absolute/indirect/relative if condition is met	4
JMPS	Jump absolute to a code segment	4
J(N)B	Jump relative if direct bit is (not) set	4
JBC	Jump relative and clear bit if direct bit is set	4

Table 4 : Instruction Set Summary

Mnemonic	Description	Bytes
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

6.3 - MAC Coprocessor Specific Instructions

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

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

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

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

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

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

The Table 5 shows the various combinations of pointer post-modification for each of these 2 new addressing modes. In this document the symbols "[$Rw_n\otimes$]" and "[$IDX_i\otimes$]" refer to these addressing modes.

Table 5 : Pointer Post-modification Combinations for IDXi and Rwn

Symbol	Mnemonic	Address Pointer Operation
"[$IDX_i\otimes$]" stands for	[IDX_i]	(IDX_i) ← (IDX_i) (no-op)
	[IDX_i+]	(IDX_i) ← (IDX_i) +2 (i=0,1)
	[IDX_i]	(IDX_i) ← (IDX_i)2 (i=0,1)
	[$IDX_i + QX_j$]	(IDX_i) ← (IDX_i) + (QX_j) (i, j =0,1)
	[IDX_i QX_j]	(IDX_i) ← (IDX_i) (QX_j) (i, j =0,1)
"[$Rw_n\otimes$]" stands for	[Rwn]	(Rwn) ← (Rwn) (no-op)
	[Rwn+]	(Rwn) ← (Rwn) +2 (n=0-15)
	[Rwn-]	(Rwn) ← (Rwn)2 (k=0-15)
	[Rwn+QR_j]	(Rwn) ← (Rwn) + (QR_j) (n=0-15;j =0,1)
	[Rwn QR_j]	(Rwn) ← (Rwn) (QR_j) (n=0-15; j =0,1)

Table 6 : MAC Registers Referenced as 'CoReg'

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

7 - EXTERNAL BUS CONTROLLER

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

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

– 16-/18-/20-/24-bit addresses 16-bit data, demultiplexed

– 16-/18-/20-/24-bit addresses 16-bit data, multiplexed

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

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

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

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

Up to 4 independent address windows may be defined (using register pairs ADDRSELx / BUSCONx) to access different resources and bus characteristics.

These address windows are arranged hierarchically where BUSCON4 overrides BUSCON3 and BUSCON2 overrides BUSCON1. All accesses to locations not covered by these 4 address windows are controlled by BUSCON0.

Up to 5 external $\overline{CS}$ signals (4 windows plus default) can be generated in order to save external glue logic. Access to very slow memories is supported by a 'Ready' function.

A $\overline{HOLD}$/$\overline{HLDA}$ protocol is available for bus arbitration which shares external resources with other bus masters. The bus arbitration is enabled by setting bit HLDEN in register PSW. After setting HLDEN once, pins P6.7...P6.5 ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$) are automatically controlled by the EBC. In master mode (default after reset) the $\overline{HLDA}$ pin is an output.

By setting bit DP6.7 to'1' the slave mode is selected where pin $\overline{HLDA}$ is switched to input. This directly connects the slave controller to another master controller without glue logic.

For applications which require less external memory space, the address space can be restricted to 1 MByte, 256 KByte or to 64 KByte. Port 4 outputs all 8 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 CSx lines change half a CPU clock cycle after the rising edge of ALE. With the CSCFG bit set in the SYSCON register the CSx lines change with the rising edge of ALE.

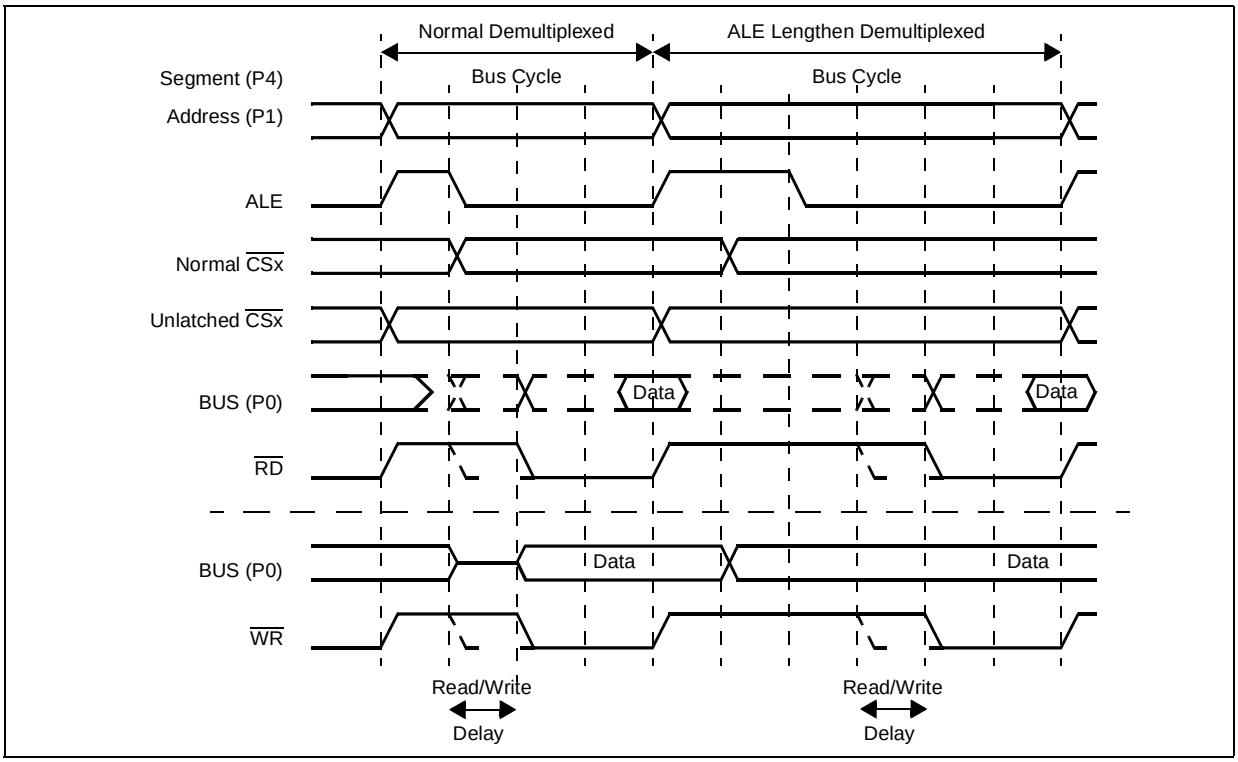
The active level of the READY pin can be set by bit RDYPOL in the BUSCONx registers. When the READY function is enabled for a specific address window, each bus cycle within the window must be terminated with the active level defined by bit RDYPOL in the associated BUSCON register.

7.1 - Programmable Chip Select Timing Control

The ST10F280 allows the user to adjust the position of the CSx lines changes. By default (after reset), the CSx lines are changing half a CPU clock cycle (12.5 ns at f_{CPU} = 40MHz) after the rising edge of ALE.

With the CSCFG bit set in the SYSCON register, the CSx lines are changing with the rising edge of ALE, thus the CSx lines are changing at the same time the address lines are changing. See Section 19.2 - System Configuration Registers for detailled description of SYSCON register.

Figure 11 : Chip Select Delay



7.2 - READY Programmable Polarity

The active level of the READY pin can be selected by software via the RDYPOL bit in the BUSCONx registers. When the READY function is enabled for a specific address window, each bus cycle within this window must be terminated with the active level defined by this RDYPOL bit in the associted BUSCON 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	CSRE N0	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 (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 (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 (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 (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			

Bit	Function
RDYPOLx	**Ready Active Level Control** 0 The active level on the READY pin is low, bus cycle terminates with a ‘0’ on READY pin, 1 The active level on the READY pin is high, bus cycle terminates with a ‘1’ on READY pin.

8 - INTERRUPT SYSTEM

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

The ST10F280 architecture supports several mechanisms for fast and flexible response to service requests that can be generated from various sources (internal or external) to the microcontroller. Any of these interrupt requests can be serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is 'stolen' from the current CPU activity to perform a PEC service. A PEC service implies a single Byte or Word data transfer between any two memory locations with an additional increment of either the PEC source or destination pointer. An individual PEC transfer counter is implicitly decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited to perform the transmission or the reception of blocks of data. The ST10F280 has 8 PEC channels, each of them offers such fast interrupt-driven data transfer capabilities.

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

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

8.1 - External Interrupts

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

Fast external interrupts may also have interrupt sources selected from other peripherals; for example the CANx controller receive signal (CANx_RxD) can be used to interrupt the system. This new function is controlled using the 'External Interrupt Source Selection' register EXISEL.

EXISEL (F1DAh / EDh) ESFR Reset Value: 0000h

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

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

EXIxSS	Port 2 pin	Alternate Source
0	P2.8	CAN1_RxD
1	P2.9	CAN2_RxD
2...7	P2.10...15	Not used (zero)

EXICON (**F1C0h** / **E0h**) ESFR Reset Value: 0000h

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

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

8.2 - Interrupt Registers and Vectors Location List

Table 7 shows all the available ST10F280 interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers:

Table 7 : 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

Table 7 : Interrupt Sources (continued)

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
CAPCOM Register 21	CC21IR	CC21IE	CC21INT	00'00D4h	35h
CAPCOM Register 22	CC22IR	CC22IE	CC22INT	00'00D8h	36h
CAPCOM Register 23	CC23IR	CC23IE	CC23INT	00'00DCh	37h
CAPCOM Register 24	CC24IR	CC24IE	CC24INT	00'00E0h	38h
CAPCOM Register 25	CC25IR	CC25IE	CC25INT	00'00E4h	39h
CAPCOM Register 26	CC26IR	CC26IE	CC26INT	00'00E8h	3Ah
CAPCOM Register 27	CC27IR	CC27IE	CC27INT	00'00ECh	3Bh
CAPCOM Register 28	CC28IR	CC28IE	CC28INT	00'00F0h	3Ch
CAPCOM Register 29	CC29IR	CC29IE	CC29INT	00'0110h	44h
CAPCOM Register 30	CC30IR	CC30IE	CC30INT	00'0114h	45h
CAPCOM Register 31	CC31IR	CC31IE	CC31INT	00'0118h	46h
CAPCOM Timer 0	T0IR	T0IE	T0INT	00'0080h	20h
CAPCOM Timer 1	T1IR	T1IE	T1INT	00'0084h	21h
CAPCOM Timer 7	T7IR	T7IE	T7INT	00'00F4h	3Dh
CAPCOM Timer 8	T8IR	T8IE	T8INT	00'00F8h	3Eh
GPT1 Timer 2	T2IR	T2IE	T2INT	00'0088h	22h
GPT1 Timer 3	T3IR	T3IE	T3INT	00'008Ch	23h
GPT1 Timer 4	T4IR	T4IE	T4INT	00'0090h	24h
GPT2 Timer 5	T5IR	T5IE	T5INT	00'0094h	25h
GPT2 Timer 6	T6IR	T6IE	T6INT	00'0098h	26h
GPT2 CAPREL Register	CRIR	CRIE	CRINT	00'009Ch	27h
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	00'00A0h	28h
A/D Overrun Error	ADEIR	ADEIE	ADEINT	00'00A4h	29h
ASC0 Transmit	S0TIR	S0TIE	S0TINT	00'00A8h	2Ah
ASC0 Transmit Buffer	S0TBIR	S0TBIE	S0TBINT	00'011Ch	47h
ASC0 Receive	S0RIR	S0RIE	S0RINT	00'00ACh	2Bh
ASC0 Error	S0EIR	S0EIE	S0EINT	00'00B0h	2Ch
SSC Transmit	SCTIR	SCTIE	SCTINT	00'00B4h	2Dh
SSC Receive	SCRIR	SCRIE	SCRINT	00'00B8h	2Eh
SSC Error	SCEIR	SCEIE	SCEINT	00'00BCh	2Fh
PWM Channel 0...3	PWMIR	PWMIE	PWMINT	00'00FCh	3Fh
CAN1 Interface	XP0IR	XP0IE	XP0INT	00'0100h	40h
CAN2 Interface	XP1IR	XP1IE	XP1INT	00'0104h	41h
XPWM	XP2IR	XP2IE	XP2INT	00'0108h	42h
PLL Unlock/OWD	XP3IR	XP3IE	XP3INT	00'010Ch	43h

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

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

8.3 - Interrupt Control Registers

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

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

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

xxIC (yyyyh / zzh) SFR Area Reset Value: - - 00h

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

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

8.4 - Exception and Error Traps List

Table 8 shows all of the possible exceptions or error conditions that can arise during run-time :

Table 8 : Exceptions or Error Conditions that Can Arise During Run-time

Exception Condition	Trap Flag	Trap Vector	Vector Location	Trap Number	Trap * Priority
Reset Functions					MAXIMUM
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					
Non-Maskable Interrupt	NMI	NMITRAP	00'0008h	02h	II
Stack Overflow	STKOF	STOTRAP	00'0010h	04h	II
Stack Underflow	STKUF	STUTRAP	00'0018h	06h	II
Class B Hardware Traps					
Undefined Opcode	UNDOPC	BTRAP	00'0028h	0Ah	I
Protected Instruction Fault	PRTFLT	BTRAP	00'0028h	0Ah	I
Illegal Word Operand Access	ILLOPA	BTRAP	00'0028h	0Ah	I
Illegal Instruction Access	ILLINA	BTRAP	00'0028h	0Ah	I
Illegal External Bus Access	ILLBUS	BTRAP	00'0028h	0Ah	I
MAC Trap	MACTRP	BTRAP	00'0028h	0Ah	I
					MINIMUM
Reserved			[2Ch –3Ch]	[0Bh – 0Fh]	
Software Traps TRAP Instruction			Any [00'0000h– 00'01FCh] in steps of 4h	Any [00h – 7Fh]	Current CPU Priority

* *- All the class B traps have the same trap number (and vector) and the same lower priority compare to the class A traps and to the resets.*

- Each class A traps has a dedicated trap number (and vector). They are prioritized in the second priority level.

- The resets have the highest priority level and the same trap number.

- The PSW.ILVL CPU priority is forced to the highest level (15) when these exeptions are serviced.

9 - CAPTURE/COMPARE (CAPCOM) UNITS

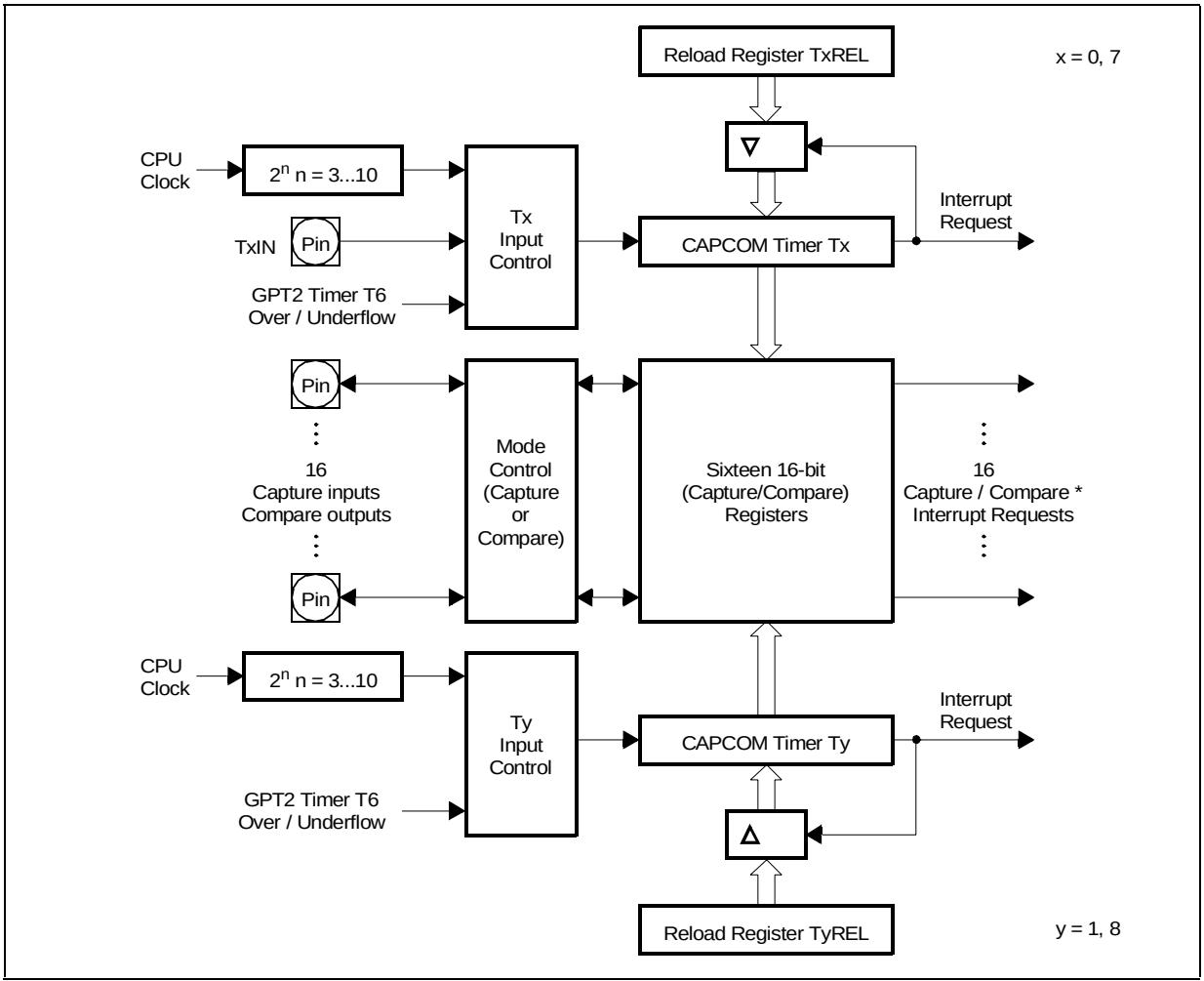
The ST10F280 has two 16 channels CAPCOM units as described in Figure 12. These support generation and control of timing sequences on up to 32 channels with a maximum resolution of 200ns at 40MHz CPU clock. The CAPCOM units are typically used to handle high speed I/O tasks such as pulse and waveform generation, pulse width modulation (PMW), Digital to Analog (D/A) conversion, software timing, or time recording relative to external events.

Four 16-bit timers (T0/T1, T7/T8) with reload registers provide two independent time bases for the capture/compare register array (See Figure 13 and Figure 14).

The input clock for the timers is programmable to several prescaled values of the internal system clock, or may be derived from an overflow/ underflow of timer T6 in module GPT2. This provides a wide range of variation for the timer period and resolution and allows precise adjustments to application specific requirements. In addition, external count inputs for CAPCOM timers T0 and T7 allow event scheduling for the capture/compare registers relative to external events.

Each of the two capture/compare register arrays contain 16 dual purpose capture/compare registers, each of which may be individually allocated to either CAPCOM timer T0 or T1 (T7 or T8, respectively), and programmed for capture or compare functions. Each of the 32 registers has one associated port pin which serves as an input pin for triggering the capture function, or as an output pin to indicate the occurrence of a compare event. Figure 12 shows the basic structure of the two CAPCOM units.

Figure 12 : CAPCOM Unit Block Diagram



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

Figure 13 : Block Diagram of CAPCOM Timers T0 and T7

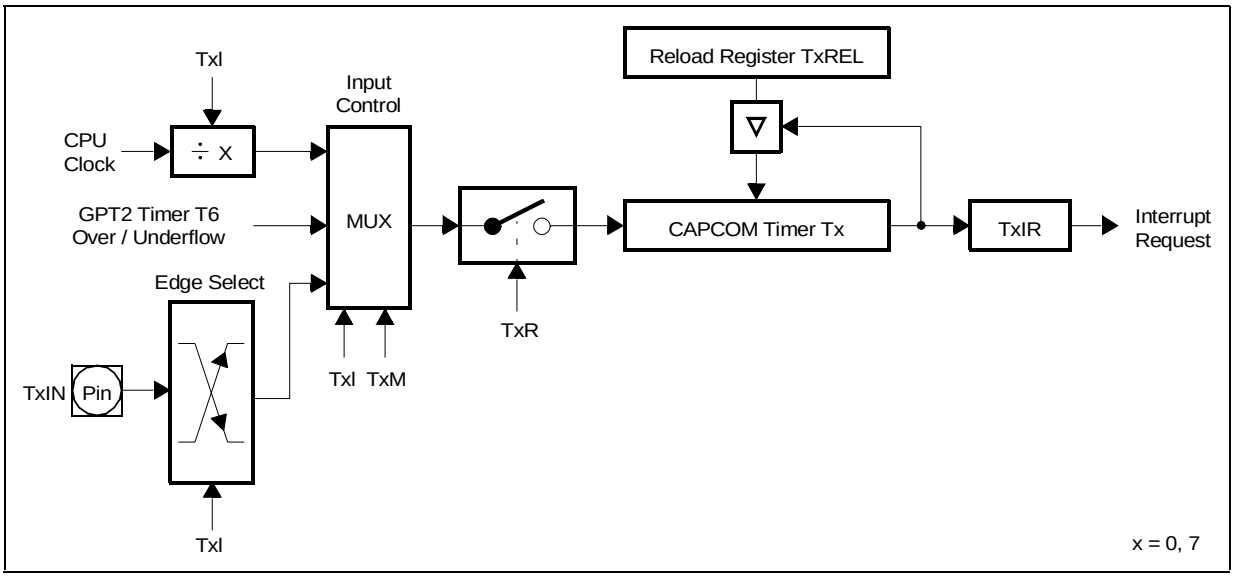
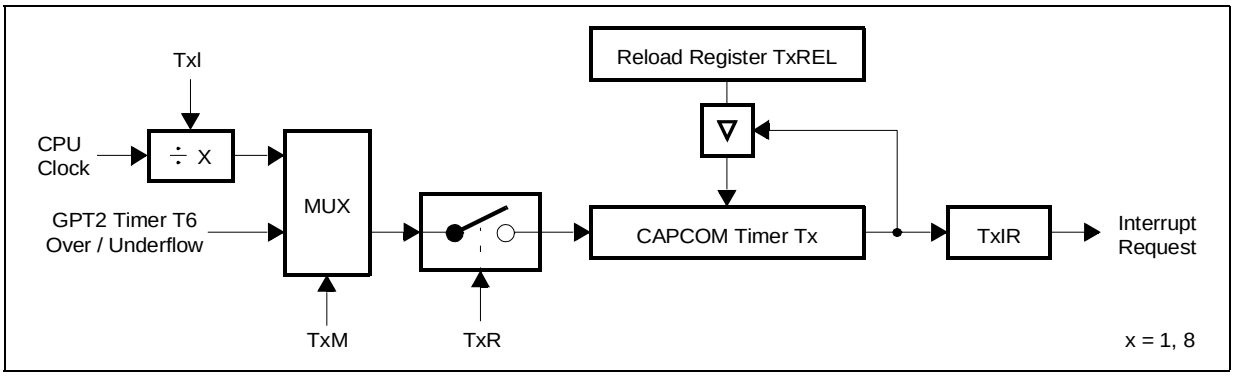


Figure 14 : Block Diagram of CAPCOM Timers T1 and T8



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

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

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

When a match occurs between the timer value and the value in a capture /compare register, specific actions will be taken based on the selected compare mode (see Table 9).

The input frequencies f_{Tx}, for the timer input selector Tx, are determined as a function of the CPU clocks. The timer input frequencies, resolution and periods which result from the selected pre-scaler option in TxI when using a 40MHz CPU clock are listed in the Table 10.

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

Table 9 : Compare Modes

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

Table 10 : CAPCOM Timer Input Frequencies, Resolution and Periods

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

10 - GENERAL PURPOSE TIMER UNIT

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

10.1 - GPT1

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

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

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

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

Table 11 lists the timer input frequencies, resolution and periods for each pre-scaler option at 40MHz CPU clock. This also applies to the Gated Timer Mode of T3 and to the auxiliary timers T2 and T4 in Timer and Gated Timer Mode. The count direction (up/down) for each timer is programmable by software or may be altered dynamically by an external signal on a port pin (TxEUD).

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

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

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

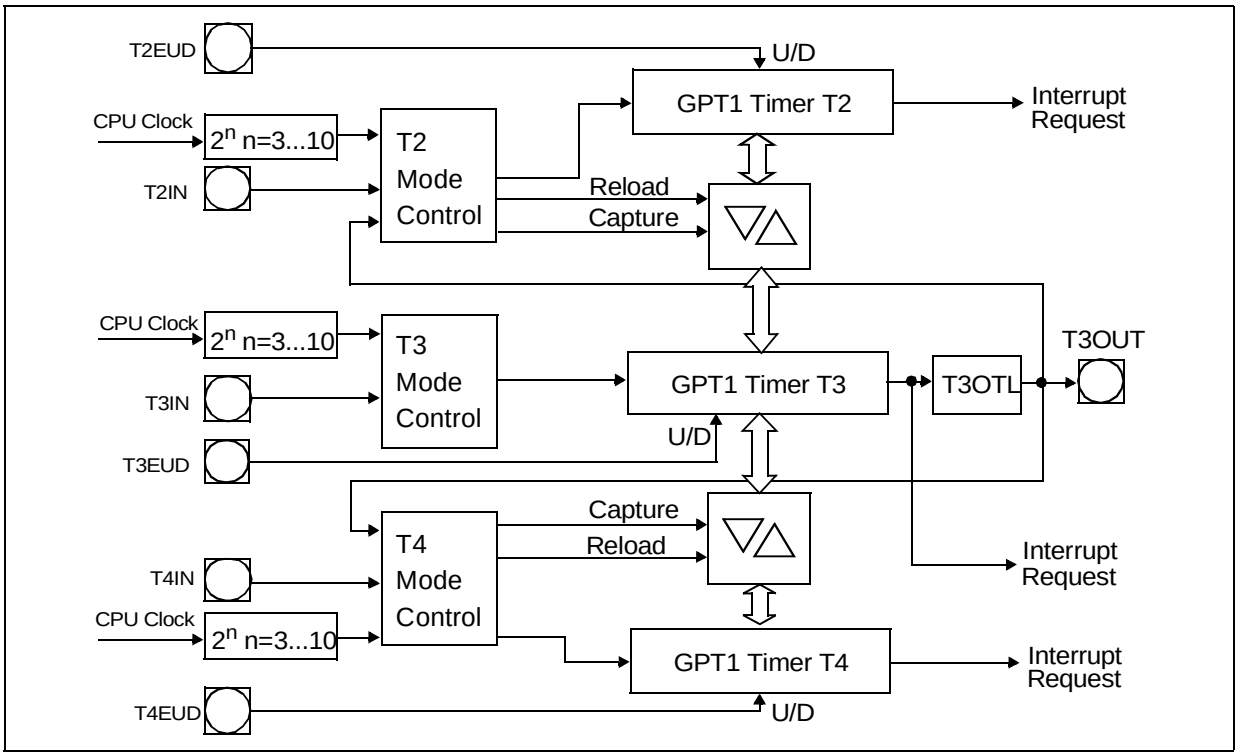
In addition to their basic operating modes, timers T2 and T4 may be configured as reload or capture registers for timer T3. When used as capture or reload registers, timers T2 and T4 are stopped. The contents of timer T3 is captured into T2 or T4 in response to a signal at their associated input pins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

Timer T3 is reloaded with the contents of T2 or T4 triggered either by an external signal or by a selectable state transition of its toggle latch T3OTL. When both T2 and T4 are configured to alternately reload T3 on opposite state transitions of T3OTL with the low and high times of a PWM signal, this signal can be constantly generated without software intervention.

Table 11 : GPT1 Timer Input Frequencies, Resolution and Periods

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

Figure 15 : Block Diagram of GPT1



10.2 - GPT2

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

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

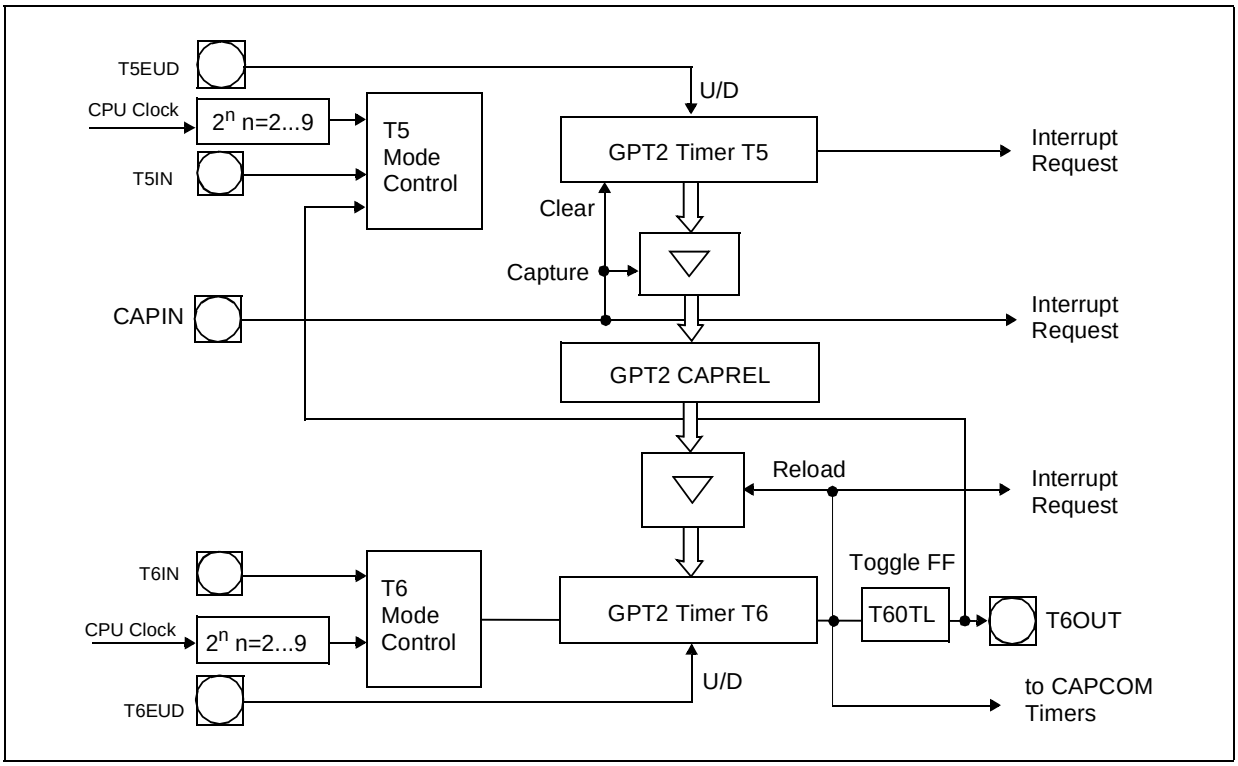
The capture trigger (timer T5 to CAPREL) may also be generated upon transitions of GPT1 timer T3 inputs T3IN and/or T3EUD. This is advantageous when T3 operates in Incremental Interface Mode.

Table 12 lists the timer input frequencies, resolution and periods for each pre-scaler option at 40MHz CPU clock. This also applies to the Gated Timer Mode of T6 and to the auxiliary timer T5 in Timer and Gated Timer Mode.

Table 12 : GPT2 Timer Input Frequencies, Resolution and Period

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

Figure 16 : Block Diagram of GPT2



11 - PWM MODULE

11.1 - Standard PWM Module

The pulse width modulation module can generate up to four PWM output signals using edge-aligned or centre-aligned PWM. In addition, the PWM module can generate PWM burst signals and single shot outputs. The Table 13 shows the PWM frequencies for different resolutions.

The level of the output signals is selectable and the PWM module can generate interrupt requests.

Figure 17 : Block Diagram of PWM Module

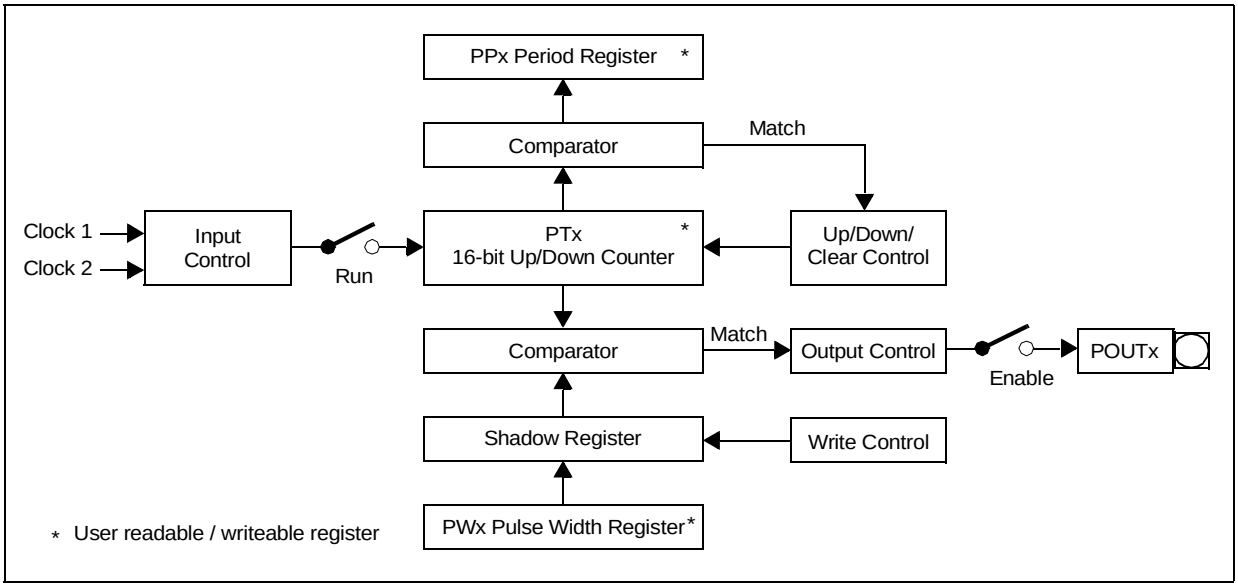


Table 13 : PWM Unit Frequencies and Resolution at 40MHz CPU Clock

Mode 0	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU Clock/1	25ns	156.25kHz	39.06kHz	9.77kHz	2.44Hz	610.35Hz
CPU Clock/64	1.6µs	2.44Hz	610.35Hz	152.58Hz	38.15Hz	9.54Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU Clock/1	25ns	78.12kHz	19.53kHz	4.88kHz	1.22kHz	305.17Hz
CPU Clock/64	1.6µs	1.22kHz	305.17Hz	76.29Hz	19.07Hz	4.77Hz

11.2 - New PWM Module : XPWM

The new Pulse Width Modulation (XPWM) Module of the ST10F280 is mapped on the XBUS interface (Address range 00'EC00h-00'ECFFh) and allows the generation of up to 4 independent PWM signals.The XPWM is enabled by setting XPEN bit 2 of the SYSCON register and bit 4 of the new XPERCON register. The frequency range of these XPWM signals for a 40MHz CPU clock is from 9.6Hz up to 20MHz for edge aligned signals. For center aligned signals the frequency range is 4.8Hz up to 10MHz (see detailed description). The minimum values depend on the width (16 bit) and the resolution (CLK/1 or CLK/64) of the XPWM timers. The maximum values assume that the XPWM output signal changes with every cycle of the respective timer. In a real application the maximum XPWM frequency will depend on the required resolution of the XPWM output signal (see Figure 18).

The Pulse Width Modulation Module consists of 4 independent PWM channels. Each channel has a 16-bit up/down counter XPTx, a 16-bit period register XPPx with a shadow latch, a 16-bit pulse width register XPWx with a shadow latch, two comparators, and the necessary control logic. The operation of all four channels is controlled by two common control registers, XPWMCON0 and XPWMCON1, and the interrupt control and status is handled by one interrupt control register XP2IC, which is also common for all channels (see Figure 19).

Figure 18 : SFRs and Port Pins Associated with the XPWM Module

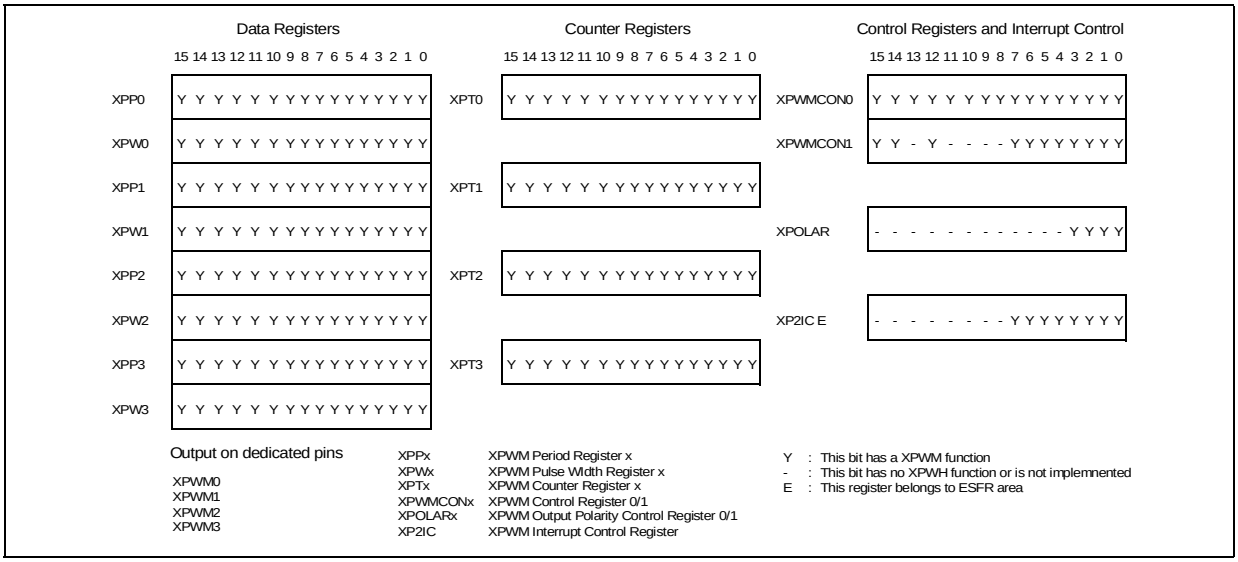
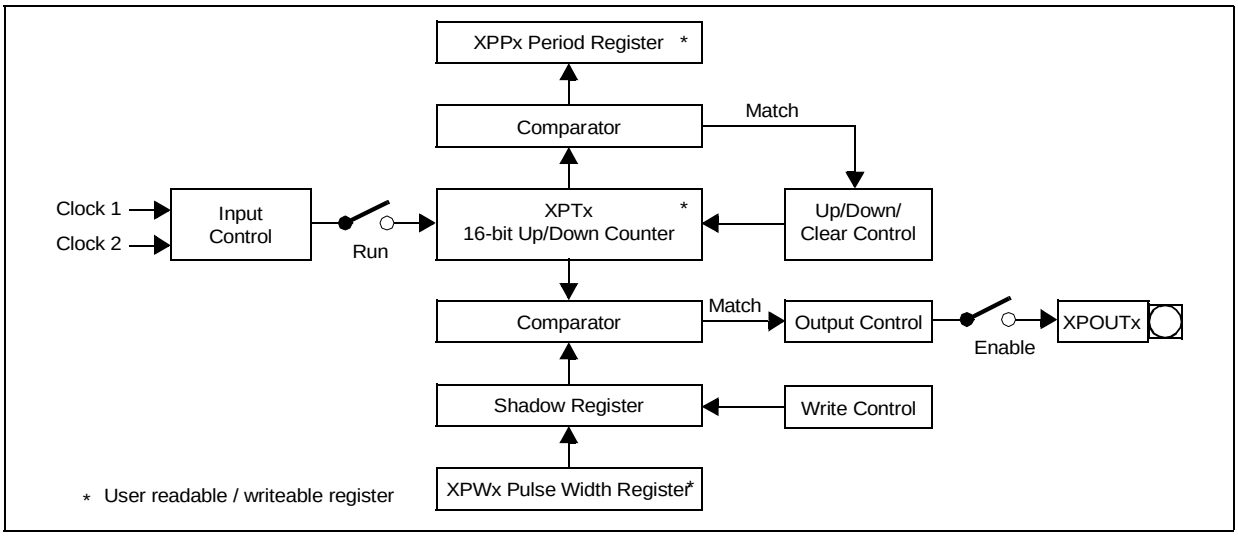


Figure 19 : XPWM Channel Block Diagram



11.2.1 - Operating Modes

The XPWM module provides four different operating modes:

– **Mode 0 Standard PWM** generation (edge aligned PWM) available on all four channels
– **Mode 1 Symmetrical PWM** generation (center aligned PWM) available on all four channels
– **Burst mode** combines channels 0 and 1
– **Single shot mode** available on channels 2 and 3

Note: The output signals of the XPWM module are XORed with the outputs of the respective bits of XPOLAR register. After reset these bits are cleared, so the PWM signals are directly driven to the output pins. By setting the respective bits of XPOLAR register to '1' the PWM signal may be inverted (XORed with '1') before being driven to the output pin. The descriptions below refer to the standard case after reset, i.e. direct driving.

11.2.1.1 - Mode 0: Standard PWM Generation (Edge Aligned PWM)

Mode 0 is selected by clearing the respective bit XPMx in register XPWMCON1 to '0'. In this mode the timer XPTx of the respective XPWM channel is always counting up until it reaches the value in the associated period shadow register. Upon the next count pulse the timer is reset to 0000h and continues counting up with subsequent count pulses. The XPWM output signal is switched to high level when the timer contents are equal to or greater than the contents of the pulse width shadow register. The signal is switched back to low level when the respective timer is reset to 0000h, i.e. below the pulse width shadow register. The period of the resulting PWM signal is determined by the value of the respective XPPx shadow register plus 1, counted in units of the timer resolution.

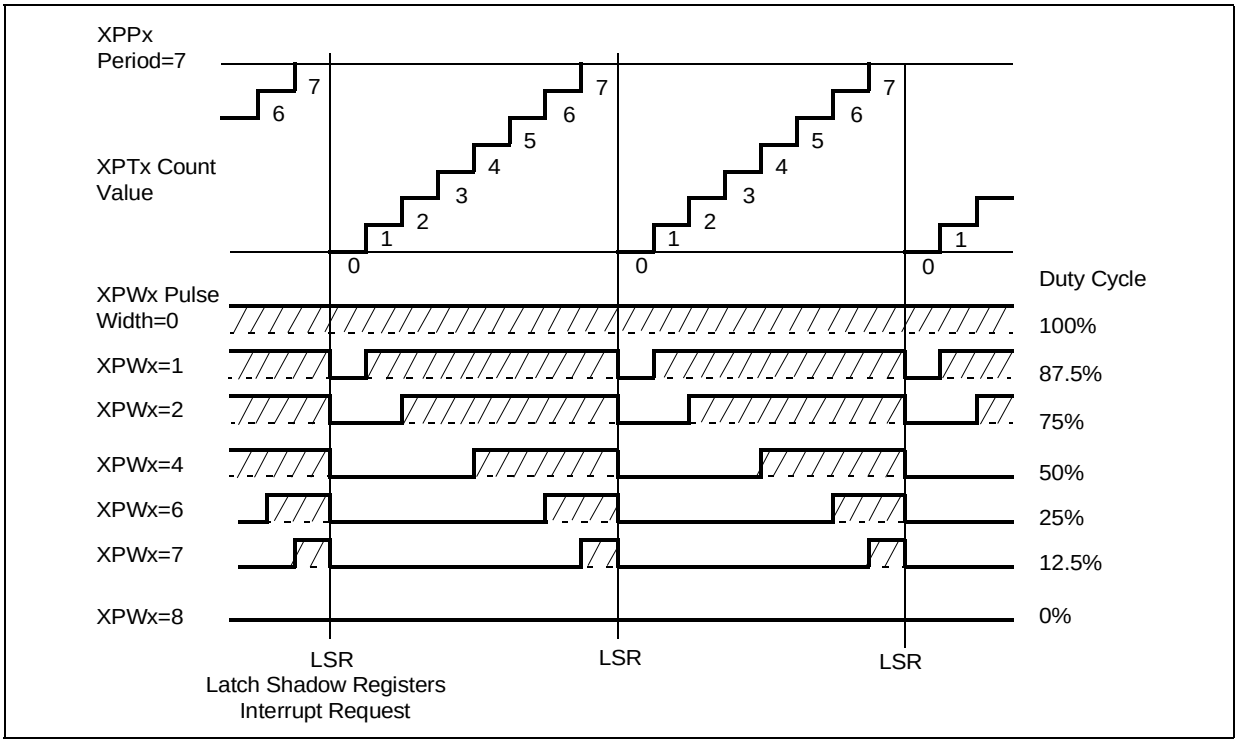
$$PWM\_Period_{Mode0} = [XPPx] + 1$$

The duty cycle of the XPWM output signal is controlled by the value in the respective pulse width shadow register. This mechanism allows the selection of duty cycles from 0% to 100% including the boundaries. For a value of 0000h the output will remain at a high level, representing a duty cycle of 100%. For a value higher than the value in the period register the output will remain at a low level, which corresponds to a duty cycle of 0%.

The Figure 20 illustrates the operation and output waveforms of a XPWM channel in mode 0 for different values in the pulse width register.

This mode is referred to as Edge Aligned PWM, because the value in the pulse width (shadow) register only effects the positive edge of the output signal. The negative edge is always fixed and related to the clearing of the timer.

Figure 20 : Operation and Output Waveform in Mode 0



11.2.1.2 - Mode 1: Symmetrical PWM Generation (Center Aligned PWM)

Mode 1 is selected by setting the respective bit XPMx in register XPWMCON1 to '1'. In this mode the timer XPTx of the respective XPWM channel is counting up until it reaches the value in the associated period shadow register.

Upon the next count pulse the count direction is reversed and the timer starts counting down now with subsequent count pulses until it reaches the value 0000_H. Upon the next count pulse the count direction is reversed again and the count cycle is repeated with the following count pulses.

The XPWM output signal is switched to a high level when the timer contents are equal to or greater than the contents of the pulse width shadow register while the timer is counting up.

The signal is switched back to a low level when the respective timer has counted down to a value below the contents of the pulse width shadow register. So in mode 1 this PWM value controls both edges of the output signal.

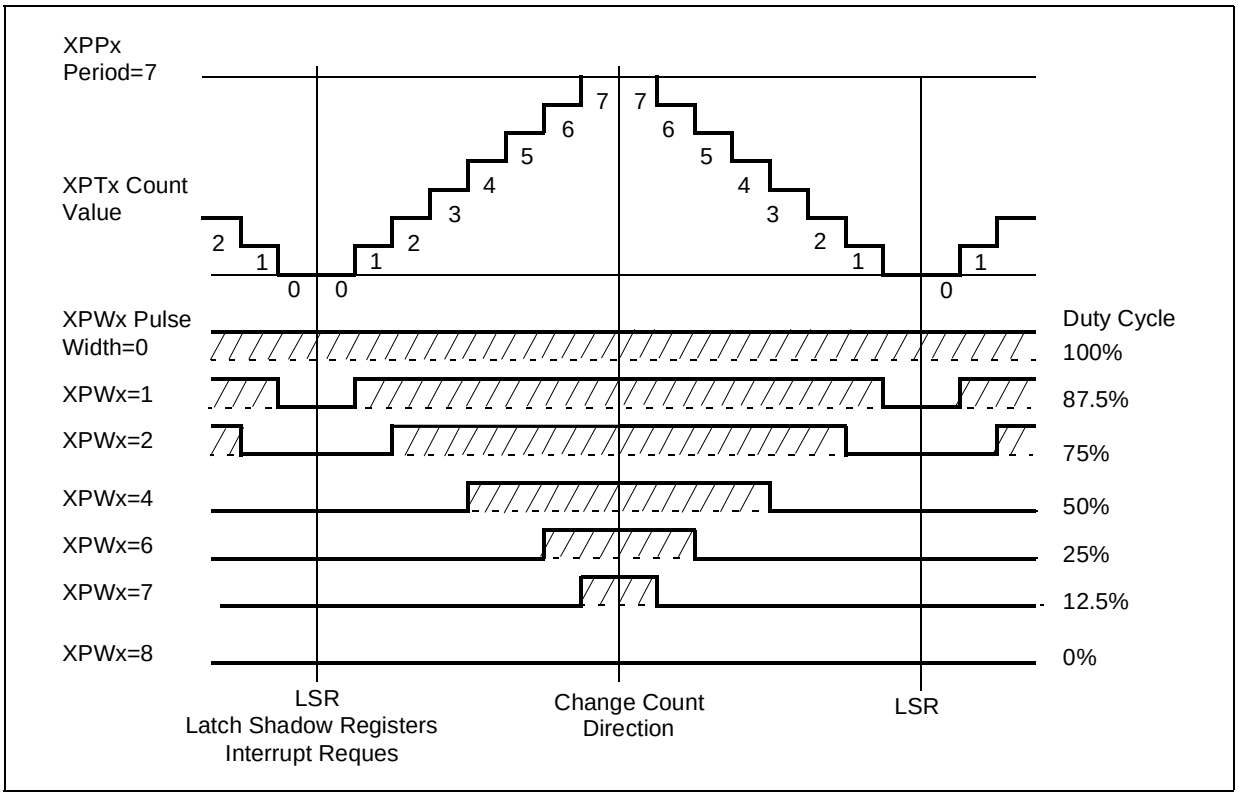
Note that in mode 1 the period of the PWM signal is twice the period of the timer:

$$PWM\_Period_{Mode1} = 2 * ([XPPx] + 1)$$

The figure below illustrates the operation and output waveforms of a XPWM channel in mode 1 for different values in the pulse width register.

This mode is referred to as Center Aligned PWM, because the value in the pulse width (shadow) register effects both edges of the output signal symmetrically.

Figure 21 : Operation and Output Waveform in Mode 1

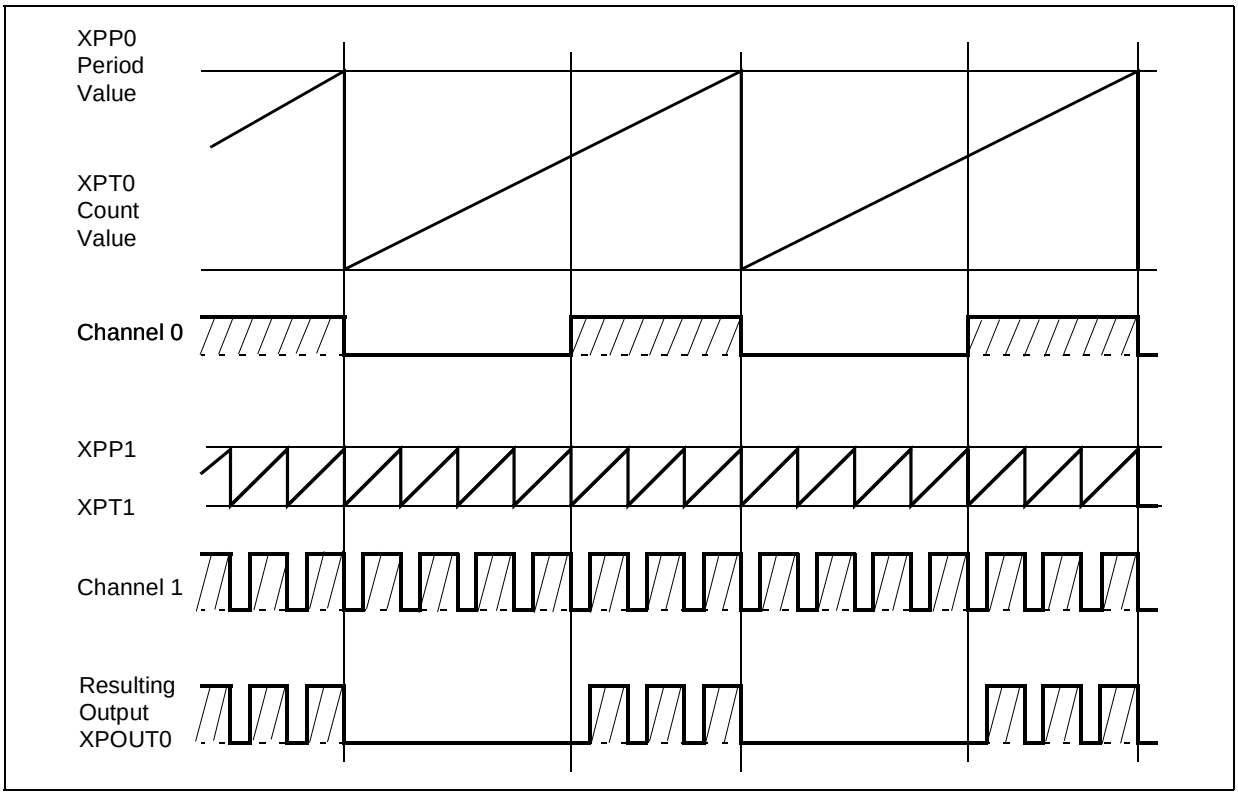


11.2.1.3 - Burst Mode

Burst mode is selected by setting bit PB01 in register XPWMCON1 to '1'. This mode combines the signals from XPWM channels 0 and 1 onto the port pin of channel 0. The output of channel 0 is replaced with the logical AND of channels 0 and 1. The output of channel 1 can still be used at its associated output pin (if enabled).
Each of the two channels can either operate in mode 0 or 1.

Note: It is guaranteed by design, that no spurious spikes will occur at the output pin of channel 0 in this mode. The output of the AND gate will be transferred to the output pin synchronously to internal clocks.
XORing of the PWM signal and the port output latch value is done after the ANDing of channel 0 and 1.

Figure 22 : Operation and Output Waveform in Burst Mode



11.2.1.4 - Single Shot Mode

Single shot mode is selected by setting the respective bit PSx in register XPWMCON1 to '1'. This mode is available for XPWM channels 2 and 3.

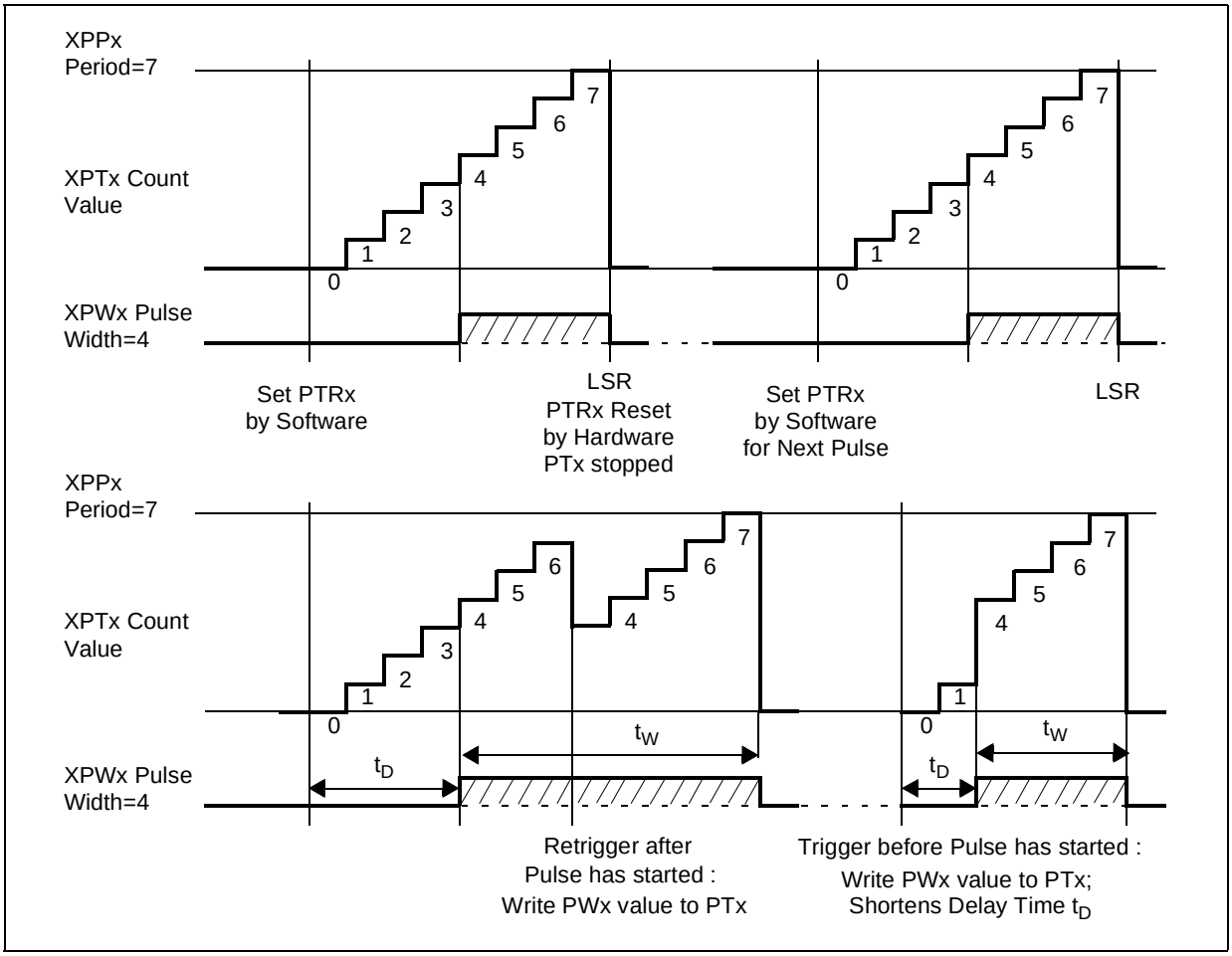
In this mode the timer XPTx of the respective XPWM channel is started via software and is counting up until it reaches the value in the associated period shadow register. Upon the next count pulse the timer is cleared to 0000h and stopped via hardware, i.e. the respective PTRx bit is cleared. The XPWM output signal is switched to high level when the timer contents are equal to or greater than the contents of the pulse width shadow register. The signal is switched back to low level when the respective timer is cleared, i.e. is below the pulse width shadow register. Thus starting a XPWM timer in single shot mode produces one single pulse on the respective port pin, provided that the pulse width value is between 0000h and the period value. In order to generate a further pulse, the timer has to be started again via software by setting bit PTRx (see Figure 23).

After starting the timer (i.e. PTRx = '1') the output pulse may be modified via software. Writing to timer XPTx changes the positive and/or negative edge of the output signal, depending on whether the pulse has already started (i.e. the output is high) or not (i.e. the output is still low). This (multiple) re-triggering is always possible while the timer is running, i.e. after the pulse has started and before the timer is stopped.

Loading counter XPTx directly with the value in the respective XPPx shadow register will abort the current PWM pulse upon the next clock pulse (counter is cleared and stopped by hardware).

By setting the period (XPPx), the timer start value (XPTx) and the pulse width value (XPWx) appropriately, the pulse width (tw) and the optional pulse delay (td) may be varied in a wide range.

Figure 23 : Operation and Output Waveform in Single Shot Mode



11.2.2 - XPWM Module Registers

The XPWM module is controlled via two sets of registers. The waveforms are selected by the channel specific registers XPTx (timer), XPPx (period) and XPWx (pulse width).

Three common registers control the operating modes and the general functions (XPWMCON0 and XPWMCON1) of the PWM module as well as the interrupt behavior (XP2IC).

Up/Down Counters XPTx

Each counter XPTx of a PWM channel is clocked either directly by the CPU clock or by the CPU clock divided by 64. Bit PTIx in register XPWMCON0 selects the respective clock source. A XPWM counter counts up or down (controlled by hardware), while its respective run control bit PTRx is set. A timer is started (PTRx = '1') via software and is stopped (PTRx = '0') either via hardware or software, depending on its operating mode. Control bit PTRx enables or disables the clock input of counter XPTx rather than controlling the XPWM output signal.

Note For the register locations please refer to the Table 14.

Table 15 summarizes the XPWM frequencies that result from various combinations of operating mode, counter resolution (input clock) and pulse width resolution.

Period Registers XPPx

The 16-bit period register XPPx of a XPWM channel determines the period of a PWM cycle, i.e. the frequency of the PWM signal. This register is buffered with a shadow register. The shadow register is loaded from the respective XPPx register at the beginning of every new PWM cycle, or upon a write access to XPPx, while the timer is stopped. The CPU accesses the XPPx register while the hardware compares the contents of the shadow register with the contents of the associated counter XPTx. When a match is found between counter and XPPx shadow register, the counter is either reset to 0000h, or the count direction is switched from counting up to counting down, depending on the selected operating mode of that XPWM channel. For the register locations refer to the Table 14.

Pulse Width Registers XPWx

This 16-bit register holds the actual PWM pulse width value which corresponds to the duty cycle of the PWM signal. This register is buffered with a shadow register. The CPU accesses the XPWx register while the hardware compares the contents of the shadow register with the contents of the associated counter XPTx. The shadow register is loaded from the respective XPWx register at the beginning of every new PWM cycle, or upon a write access to XPWx, while the timer is stopped.When the counter value is greater than or equal to the shadow register value, the PWM signal is set, otherwise it is reset. The output of the comparators may be described by the boolean formula:

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

This type of comparison allows a flexible control of the PWM signal. For the register locations refer to theTable 14.

Table 14 : XPWM Module Channel Specific Register Addresses

Register	Address	Register	Address
XPW0	EC30h	XPT0	EC10h
XPW1	EC32h	XPT1	EC12h
XPW2	EC34h	XPT2	EC14h
XPW3	EC36h	XPT3	EC16h
These registers are not bit-addressable.		XPP0	EC20h
		XPP1	EC22h
		XPP2	EC24h
		XPP3	EC26h

Table 15 : XPWM Frequency

Mode 0	Resolution	8-bit	10-bit	12-bit	14-bit	16-bit
CPU Clock/1	25ns	156.25kHz	39.06kHz	9.77kHz	2.44Hz	610.35Hz
CPU Clock/64	1.6µs	2.44Hz	610.35Hz	152.58Hz	38.15Hz	9.54Hz
Mode 1	**Resolution**	**8-bit**	**10-bit**	**12-bit**	**14-bit**	**16-bit**
CPU Clock/1	25ns	78.12kHz	19.53kHz	4.88kHz	1.22kHz	305.17Hz
CPU Clock/64	1.6µs	1.22kHz	305.17Hz	76.29Hz	19.07Hz	4.77Hz

XPWM Control Registers

Register XPWMCON0 controls the function of the timers of the four XPWM channels and the channel specific interrupts. Having the control bits organized in functional groups allows e.g. to start or stop all 4 XPWM timers simultaneously with one bitfield instruction. Note: This register is not bit-addressable.

XPWMCON0 (EC00h) Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	**PIR2**	**PIR1**	**PIR0**	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Function
PTRx	**XPWM Timer x Run Control Bit** 0 Timer XPTx is disconnected from its input clock 1 Timer XPTx is running
PTIx	**XPWM Timer x Input Clock Selection** 0 Timer XPTx clocked with CLK_{CPU} 1 TimerX PTx clocked with CLK_{CPU} / 64
PIEx	**XPWM Channel x Interrupt Enable Flag** 0 Interrupt from channel x disabled 1 Interrupt from channel x enabled
PIRx	**XPWM Channel x Interrupt Request Flag** 0 No interrupt request from channel x 1 Channel x interrupt pending (must be reset via software)

Register XPWMCON1 controls the operating modes and the outputs of the four XPWM channels. The basic operating mode for each channel (standard=edge aligned, or symmetrical=center aligned PWM mode) is selected by the mode bits XPMx. Burst mode (channels 0 and 1) and single shot mode (channel 2 or 3) are selected by separate control bits. The output signal of each XPWM channel is individually enabled by bit PENx. If the output is not enabled the respective pin can only be used to generate an interrupt request. Note: This register is not bit-addressable.

XPWMCON1 (EC02h) Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PS3	PS2	-	PB01	-	-	-	-	PM3	PM2	PM1	PM0	PEN3	PEN2	PEN1	PEN0
RW	RW	-	RW	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Function
PENx	**XPWM Channel x Output Enable Bit** 0 Channel x output signal disabled, generate interrupt only 1 Channel x output signal enabled
PMx	**XPWM Channel x Mode Control Bit** 0 Channel x operates in mode 0, edge aligned PWM 1 Channel x operates in mode 1, center aligned PWM
PB01	**XPWM Channel 0/1 Burst Mode Control Bit** 0 Channels 0 and 1 work independently in respective standard mode 1 Outputs of channels 0 and 1 are ANDed to XPWM0 in burst mode
PSx	**XPWM Channel x Single Shot Mode Control Bit** 0 Channel x works in respective standard mode 1 Channel x operates in single shot mode

11.2.3 - Interrupt Request Generation

Each of the four channels of the XPWM module can generate an individual interrupt request. Each of these “channel interrupts” can activate the common “module interrupt”, which actually interrupts the CPU. This common module interrupt is controlled by the XPWM Module Interrupt Control register XP2IC(Xperipherals 2 control register). The interrupt service routine can determine the active channel interrupt(s) from the channel specific interrupt request flags PIRx in register XPWMCON0. The interrupt request flag PIRx of a channel is set at the beginning of a new PWM cycle, i.e. upon loading the shadow registers. This indicates that registers XPPx and XPWx are now ready to receive a new value. If a channel interrupt is enabled via its respective PIEx bit, also the common interrupt request flag XP2IR in register XP2IC is set, provided that it is enabled via the common interrupt enable bit XP2IE.

Note: The channel interrupt request flags (PIRx in register XPWMCON0) are not automatically cleared by hardware upon entry into the interrupt service routine, so they must be cleared via software. The module interrupt request flag XP2IR is cleared by hardware upon entry into the service routine, regardless of how many channel interrupts were active. However, it will be set again if during execution of the service routine a new channel interrupt request is generated.

XP2IC (F196h / CBh) ESFR Reset Value: - - 00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	**XP2IR**	XP2IE	ILVL				GLVL	
								RW	RW	RW				RW	

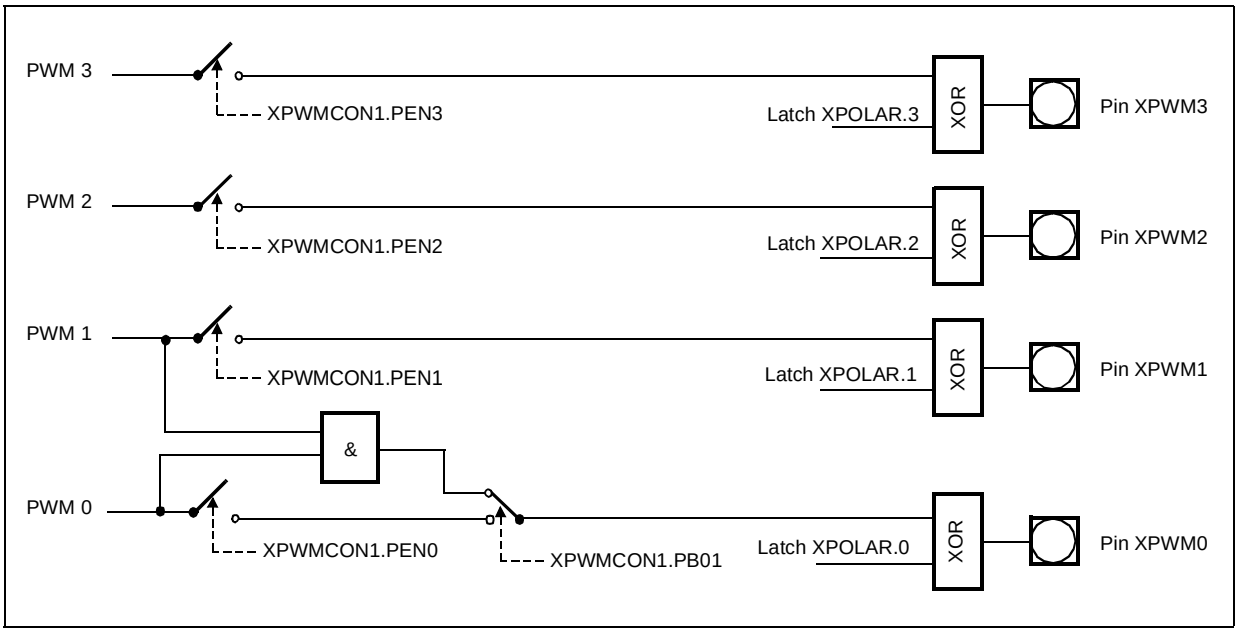
Note: Refer to the general Interrupt Control Register description for an explanation of the control fields.

11.2.4 - XPWM Output Signals

The output signals of the four XPWM channels are XPWM3...XPWM0. The output signal of each PWM channel is individually enabled by control bit PENx in register XPWMCON1.

The XPWM signals are XORed with the outputs of the register XPOLAR(3...0) before being driven to the output pins. This allows driving the XPWM signal directly to the output pin (XPOLAR.x=’0’) or driving the inverted XPWM signal (XPOLAR.x=’1’).

Figure 24 : XPWM Output Signal Generation



11.2.5 - XPOLAR Register (polarity of the XPWM channel)

XPOLAR (EC04h) Reset Value: 0000h

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

Bit	Function
XPOLAR.x	**XPOLAR Channel x polarity Bit**
	0 Polarity of Channel x is normal
	1 Polarity of Channel x is inverted

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 on some extraordinary condition, e.g. 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 = [PTx] ≥ [PWx shadow latch].

So whenever software changes registers XPTx, the respective output will reflect 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, a XPTx value below the XPWx value clears the respective output.

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

12 - PARALLEL PORTS

In order to accept or generate single external control signals or parallel data, the ST10F280 provides up to 143 parallel I/O lines, organized into two 16-bit I/O port (Port 2, XPort9), eight 8-bit I/O ports (PORT0 made of P0H and P0L, PORT1 made of P1H and P1L, Port 4, Port 6, Port 7, Port 8) , one 15-bit I/O port (Port 3) and two 16-bit input port (Port 5, XPort10).

These port lines may be used for general purpose Input/Output, controlled via software, or may be used implicitly by ST10F280's integrated peripherals or the External Bus Controller.

All port lines are bit addressable, and all input/output lines are individually (bit-wise) programmable as inputs or outputs via direction registers (except Port 5, XPort10). The I/O ports are true bidirectional ports which are switched to high impedance state when configured as inputs. The output drivers of seven I/O ports (2, 3, 4, 6, 7, 8, 9) can be configured (pin by pin) for push/pull operation or open-drain operation via ODPx control registers.

The output driver of the pads are programmable to adapt the edge characteristics to the application requirement and to improve the EMI behaviour.

This is possible using the POCONx registers for Ports P0L, P0H, P1L, P1H, P2, P3, P4, P6, P7, P8. The output driver capabilities of ALE, $\overline{RD}$ and $\overline{WR}$ control lines are programmable with the dedicated bits of POCON20 control register.

The input threshold levels are programmable (TTL/CMOS) for five ports (2, 3, 4, 7, 8) with the PICON register control bits. The logic level of a pin is clocked into the input latch once per state time, regardless whether the port is configured for input or output.

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.

Note: The new I/O ports (XPort9, XPort10) are not mapped on the SFR space but on the internal XBUS interface . The XPort9 and XPort10 are enabled by setting XPEN bit 2 of the SYSCON register and bit 3 of the new XPERCON register. On the XBUS interface, the registers are not bit-addressable.

Figure 25 : SFRs Associated with the Parallel Ports

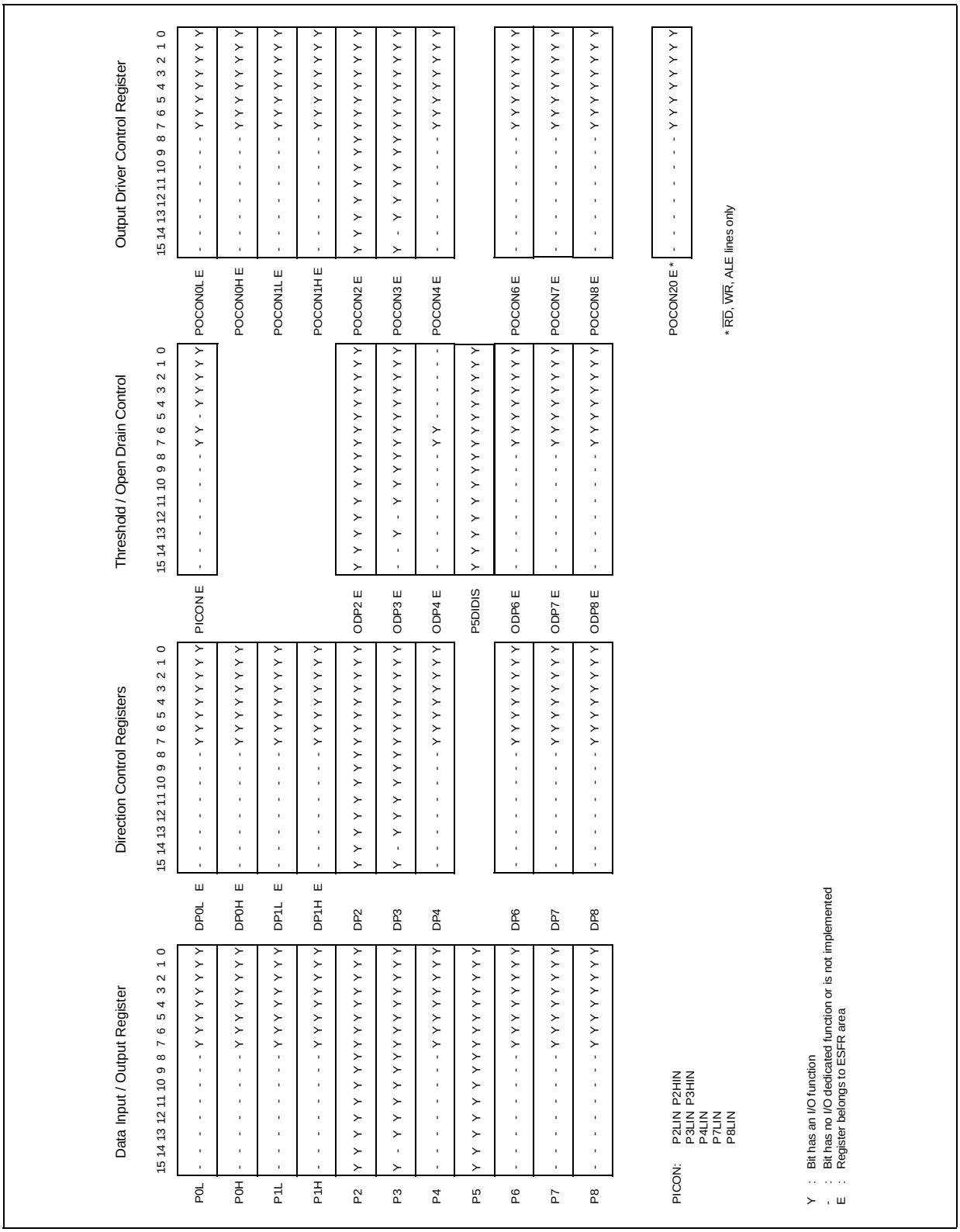
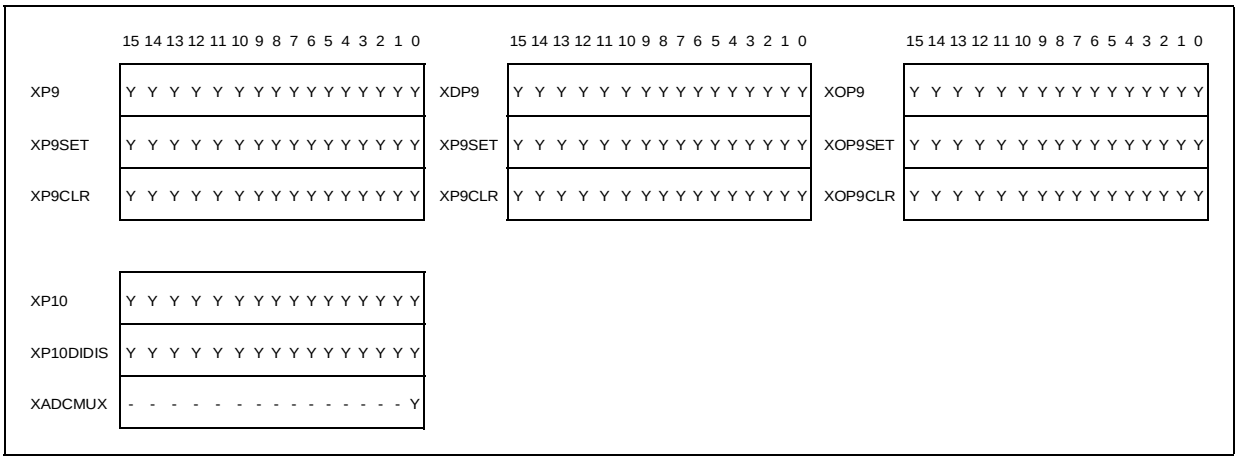


Figure 26 : XBUS Registers Associated with the Parallel Ports



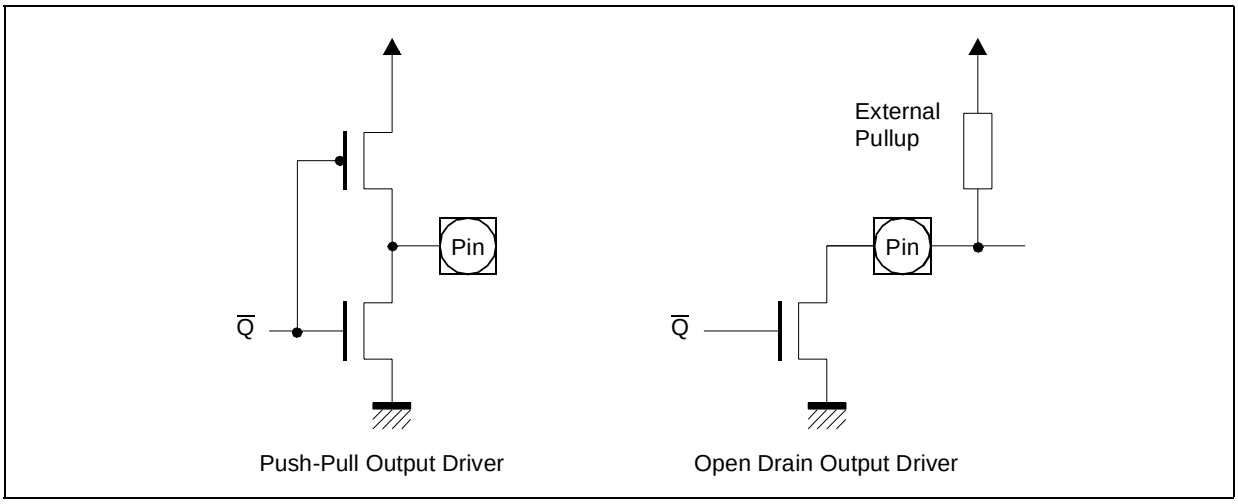
12.1 - Introduction

12.1.1 - Open Drain Mode

In the ST10F280 some ports provide Open Drain Control. This make is possible to switch the output driver of a port pin from a push/pull configuration to an open drain configuration. In push/pull mode a port output driver has an upper and a lower transistor, thus it can actively drive the line either to a high or a low level. In open drain mode the upper transistor is always switched off, and the output driver can only actively drive the line to a low level. When writing a '1' to the port latch, the lower transistor is switched off and the output enters a high-impedance state. The high level must then be provided by an external pull-up device. With this feature, it is possible to connect several port pins together to a Wired-AND configuration, saving external glue logic and/or additional software overhead for enabling/disabling output signals.

This feature is implemented for ports P2, P3, P4, P6, P7 and P8 (see respective sections), and is controlled through the respective Open Drain Control Registers ODPx. These registers allow the individual bit-wise selection of the open drain mode for each port line. If the respective control bit ODPx.y is '0' (default after reset), the output driver is in the push/pull mode. If ODPx.y is '1', the open drain configuration is selected. Note that all ODPx registers are located in the ESFR space.

Figure 27 : Output Drivers in Push/Pull Mode and in Open Drain Mode



12.1.2 - Input Threshold Control

The standard inputs of the ST10F280 determine the status of input signals according to TTL levels. In order to accept and recognize noisy signals, CMOS-like input thresholds can be selected instead of the standard TTL thresholds for all pins of Port 2, Port 3, Port4, Port 7 and Port 8. These special thresholds are defined above the TTL thresholds and feature a defined hysteresis to prevent the inputs from toggling while the respective input signal level is near the thresholds.

The Port Input Control register PICON is used to select these thresholds for each byte of the indicated ports, i.e. the 8-bit ports P7 and P8 are controlled by one bit each while ports P2 and P3 are controlled by two bits each.

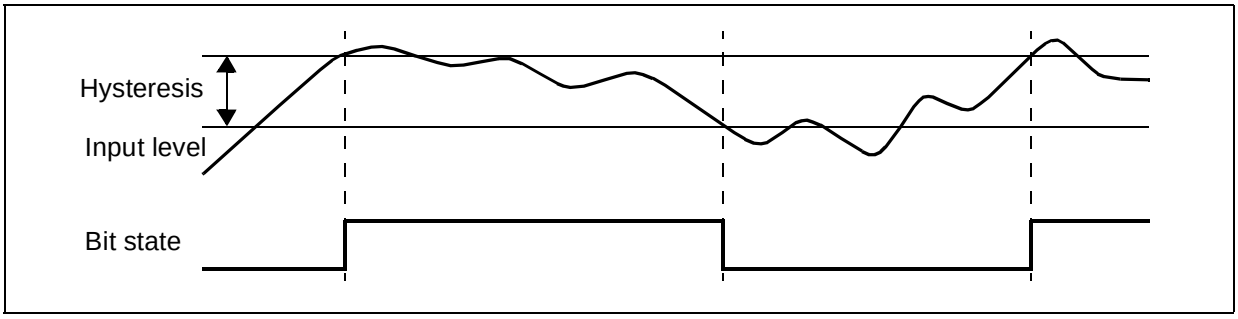
PICON (F1C4h / E2h) ESFR Reset Value:-00h

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

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

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.

Figure 28 : Hysteresis for Special Input Thresholds



12.1.3 - Output Driver Control

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

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

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

For each feature, a 2-bit control field (ie. 4 bits) is provided for each group of 4 port pads (ie. a port nibble), in port output control registers POCONx.

POCONx (F0yyh / zzh) for 8-bit Ports ESFR Reset Value: - - 00h

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

POCONx (F0yyh / zzh) for 16-bit Ports ESFR Reset Value: 0000h

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

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

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

Port Control Register Allocation

The table below lists the defined POCON registers and the allocation of control bitfields and port pins:

Control Register	Physical Address	8-Bit Address	Controlled Port			
POCON0L	F080h	40h			P0L.7...4	P0L.3...0
POCON0H	F082h	41h			P0H.7...4	P0H.3...0
POCON1L	F084h	42h			P1L.7...4	P1L.3...0
POCON1H	F086h	43h			P1H.7...4	P1H.3...0
POCON2	F088h	44h	P2.15...12	P2.11...8	P2.7...4	P2.3...0
POCON3	F08Ah	45h	P3.15, P3.13...12	P3.11...8	P3.7...4	P3.3...0
POCON4	F08Ch	46h			P4.7...4	P4.3...0
POCON6	F08Eh	47h			P6.7...4	P6.3...0
POCON7	F090h	48h			P7.7...4	P7.3...0
POCON8	F092h	49h			P8.7...4	P8.3...0

Dedicated Pins Output Control

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

POCON20 (F0AAh / 5h) ESFR Reset Value: 0000h

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

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

12.1.4 - Alternate Port Functions

Each port line has one associated programmable alternate input or output function. PORT0 and PORT1 may be used as the address and data lines when accessing external memory.

Port 4 outputs the additional segment address bits A23/A19/A18/A16 in systems where more than 64 KBytes of memory are to be accessed directly.

Port 6 provides the optional chip select outputs and the bus arbitration 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 PWM module.

Port 2 is also used for fast external interrupt inputs and for timer 7 input.

Port 3 includes alternate input/output functions of timers, serial interfaces, the optional bus control signal $\overline{BHE}$/$\overline{WRH}$ and the system clock output (CLKOUT).

Port 5 is used for the analog input channels to the A/D converter or timer control signals.

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

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

On most of the port lines, the user software is responsible for setting the proper direction when using an alternate input or output function of a pin. This is done by setting or clearing the direction control bit DPx.y of the pin before enabling the alternate function. There are port lines, however, where the direction of the port line is switched automatically. For instance, in the multiplexed external bus modes of PORT0, the direction must be switched several times for an instruction fetch in order to output the addresses and to input the data. Obviously, this cannot be done through instructions. In these cases, the direction of the port line is switched automatically by hardware if the alternate function of such a pin is enabled.
To determine the appropriate level of the port output latches check how the alternate data output is combined with the respective port latch output.

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

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

```
SINGLE_BIT: 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 more pins of one port, these pairs must be separated by instructions, which do not reference the respective port (see “Particular Pipeline Effects” in Chapter 6 - Central Processing Unit (CPU)).

12.2 - PORT0

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

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

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

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

P0H (FF02h / 81h) SFR Reset Value: - - 00h

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

Bit	Function
P0X.y	Port data register P0H or P0L bit y

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

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

DP0H (F102h / 81h) ESFR Reset Value: - - 00h

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

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

12.2.1 - Alternate Functions of PORT0

When an external bus is enabled, PORT0 is used as data bus or address/data bus.

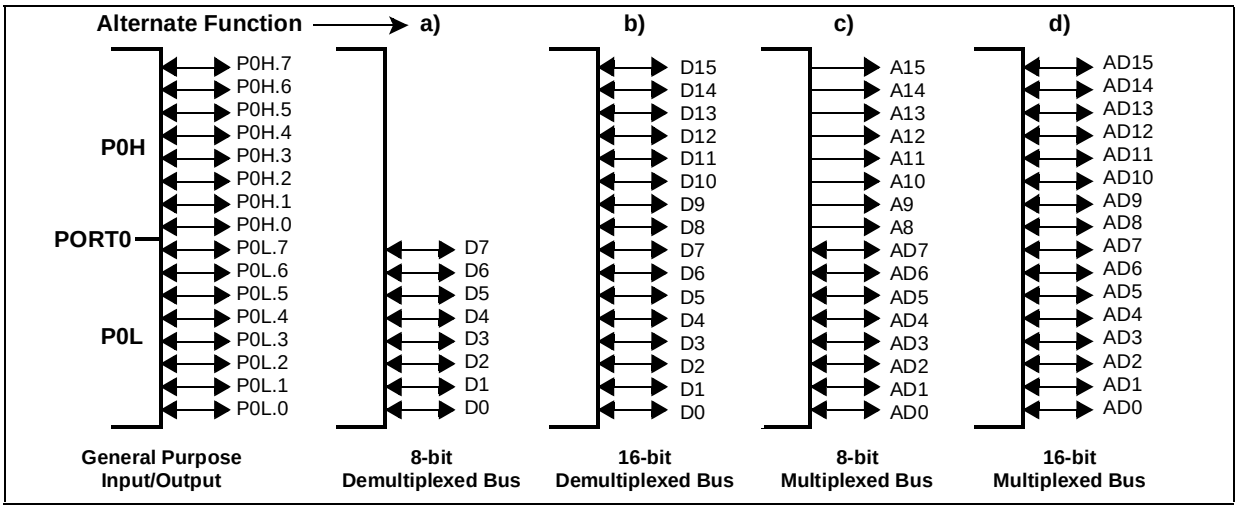
Note that an external 8-bit de-multiplexed bus only uses P0L, while P0H is free for I/O (provided that no other bus mode is enabled).

PORT0 is also used to select the system start-up configuration. During reset, PORT0 is configured to input, and each line is held high through an internal pull-up device. Each line can now be individually pulled to a low level (see DC-level specifications) through an external pull-down device. A default configuration is selected when the respective PORT0 lines are at a high level. Through pulling individual lines to a low level, this default can be changed according to the needs of the applications.

The internal pull-up devices are designed such that an external pull-down resistors can be used to apply a correct low level. These external pull-down resistors can remain connected to the PORT0 pins also during normal operation, however, care has to be taken such that they do not disturb the normal function of PORT0 (this might be the case, for example, if the external resistor is too strong). With the end of reset, the selected bus configuration will be written to the BUSCON0 register. The configuration of the high byte of PORT0, will be copied into the special register RP0H. This read-only register holds the selection for the number of chip selects and segment addresses. Software can read this register in order to react according to the selected configuration, if required. When the reset is terminated, the internal pull-up devices are switched off, and PORT0 will be switched to the appropriate operating mode.

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

Figure 29 : PORT0 I/O and Alternate Functions

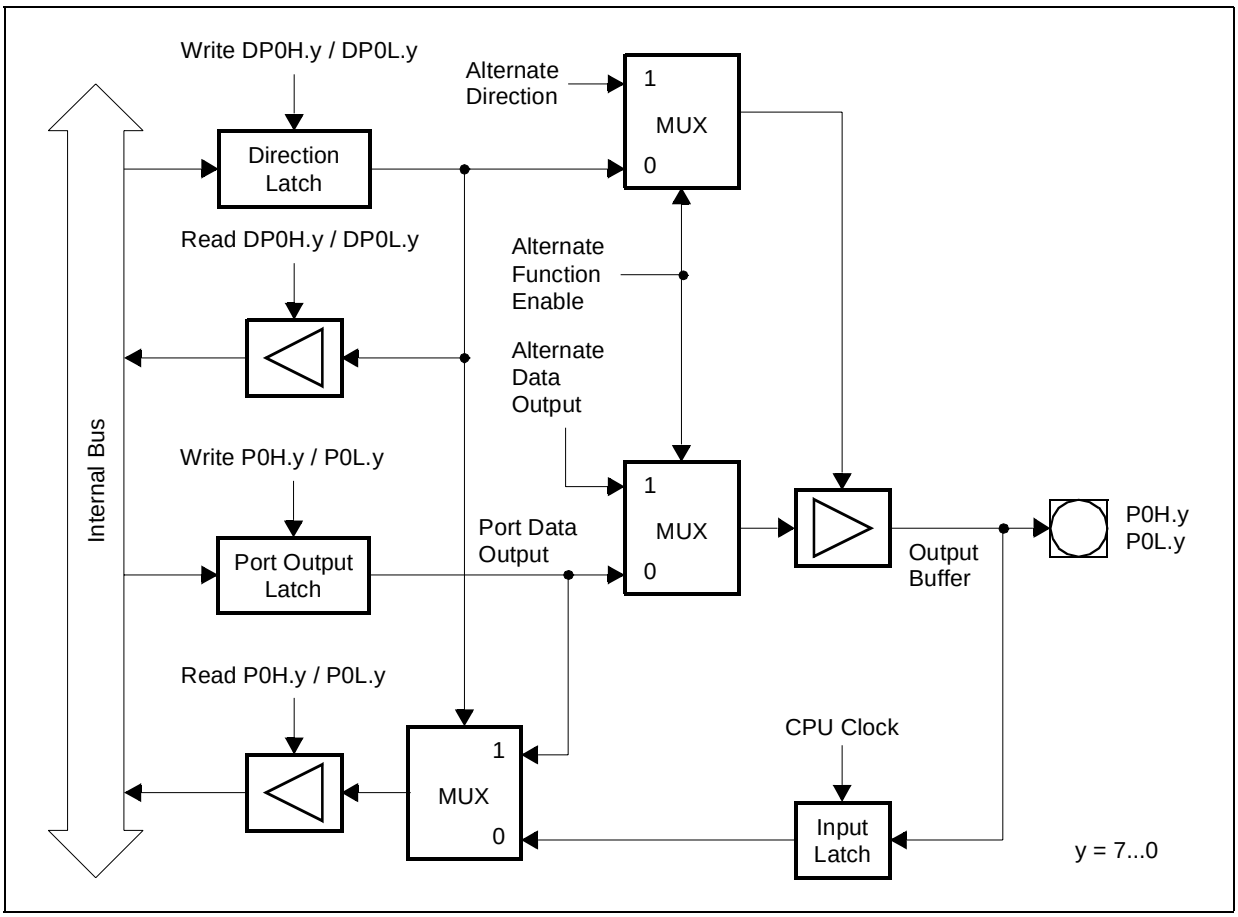


When an external bus mode is enabled, the direction of the port pin and the loading of data into the port output latch are controlled by the bus controller hardware.

The input of the port output 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 PORT0 is read on the line “Alternate Data Input”. While an external bus mode is enabled, the user software should not write to the port output latch, otherwise unpredictable results may occur. When the external bus modes are disabled, the contents of the direction register last written by the user becomes active.

The Figure 30 shows the structure of a PORT0 pin.

Figure 30 : Block Diagram of a PORT0 Pin



12.3 - PORT1

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

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

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

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

Bit	Function
P1X.y	Port data register P1H or P1L bit y

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

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

DP1H (F106h / 83h) ESFR Reset Value: - - 00h

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

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

12.3.1 - Alternate Functions of PORT1

When a de-multiplexed external bus is enabled, PORT1 is used as address bus.

Note that de-multiplexed bus modes use PORT1 as a 16-bit port. Otherwise all 16 port lines can be used for general purpose I/O.

The upper four pins of PORT1 (P1H.7...P1H.4) also serve as capture input lines for the CAPCOM2 unit (CC27IO...CC24IO).

As all other capture inputs, the capture input function of pins P1H.7...P1H.4 can also be used as external interrupt inputs (200 ns sample rate at 40MHz CPU clock).

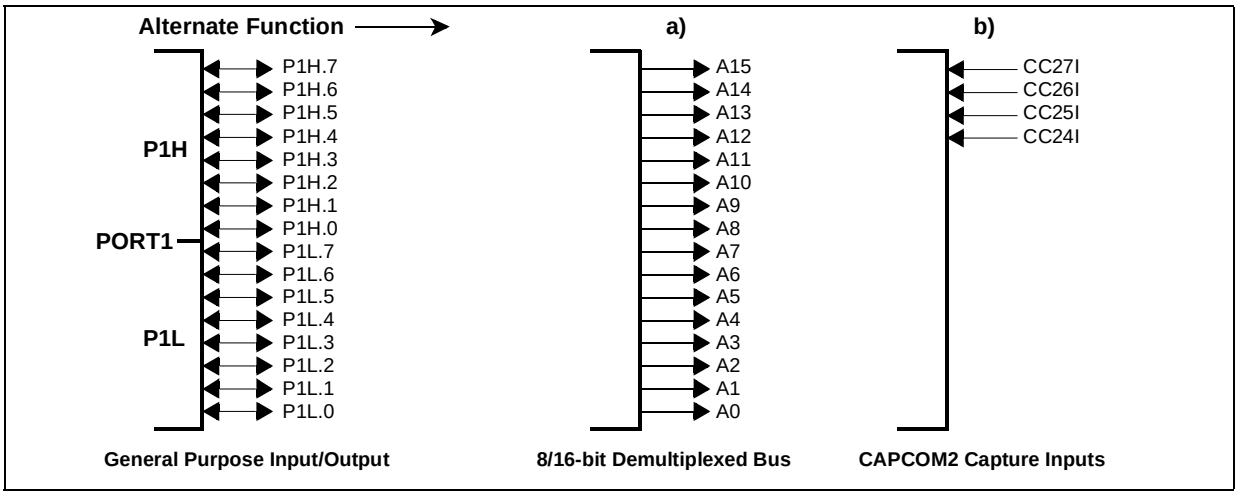
During external accesses in de-multiplexed bus modes PORT1 outputs the 16-bit intra-segment address as an alternate output function.

During external accesses in multiplexed bus modes, when no BUSCON register selects a de-multiplexed bus mode, PORT1 is not used and is available for general purpose I/O (see Figure 31).

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.

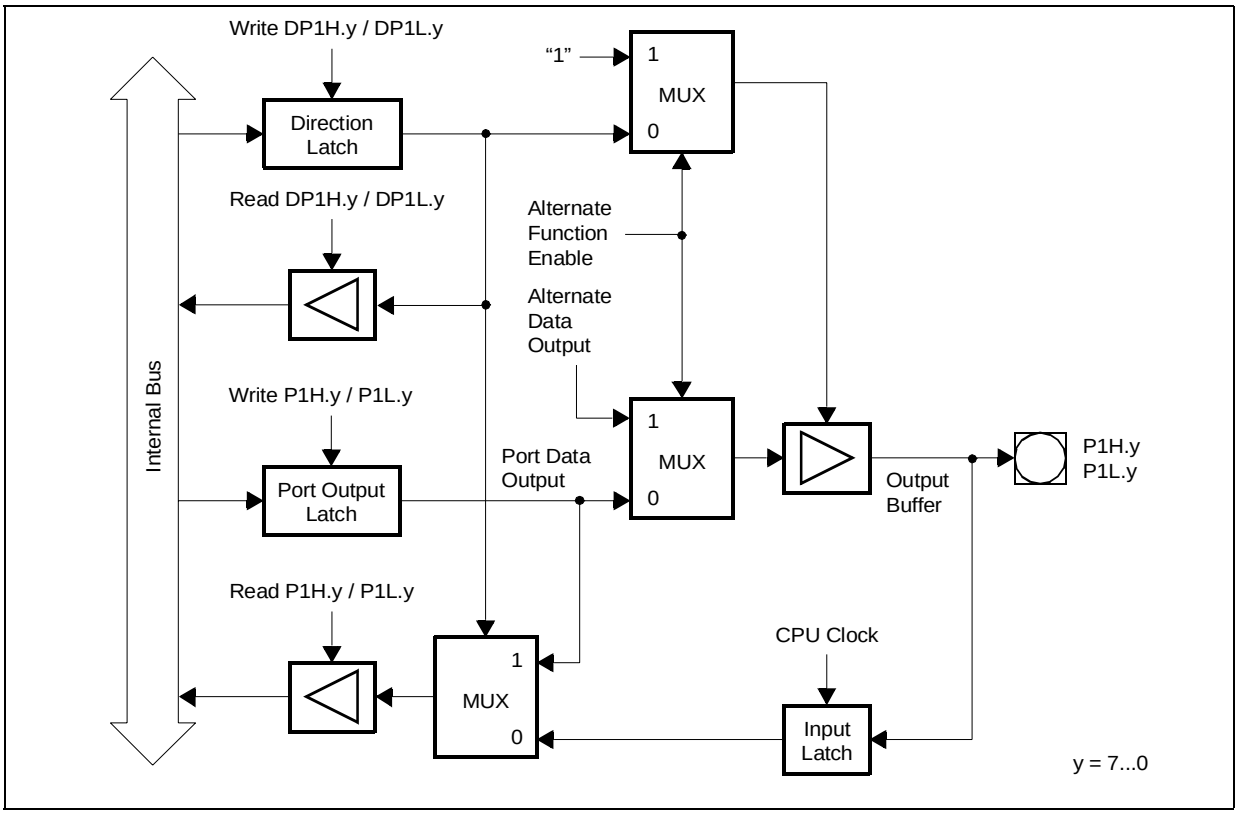
While an external bus mode is enabled, the user software should not write to the port output latch, otherwise unpredictable results may occur. When the external bus modes are disabled, the contents of the direction register last written by the user becomes active.

Figure 31 : PORT1 I/O and Alternate Functions



The figure below shows the structure of a PORT1 pin.

Figure 32 : Block Diagram of a PORT1 Pin



12.4 - 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.

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

Bit	Function
P2.y	Port data register P2 bit y

DP2 (FFC2h / E1h) SFR Reset Value: 0000h

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

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

ODP2 (F1C2h / E1h) ESFR Reset Value: 0000h

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

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

12.4.1 - Alternate Functions of Port 2

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

When a Port 2 line is used as a capture input, the state of the input latch, which represents the state of the port pin, is directed to the CAPCOM unit via the line “Alternate Pin Data Input”. If an external capture trigger signal is used, the direction of the respective pin must be set to input. If the direction is set to output, the state of the port output latch will be read since the pin represents the state of the output latch. This can be used to trigger a capture event through software by setting or clearing the port latch. Note that in the output configuration, no external device may drive the pin, otherwise conflicts would occur.

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

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

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

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

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

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

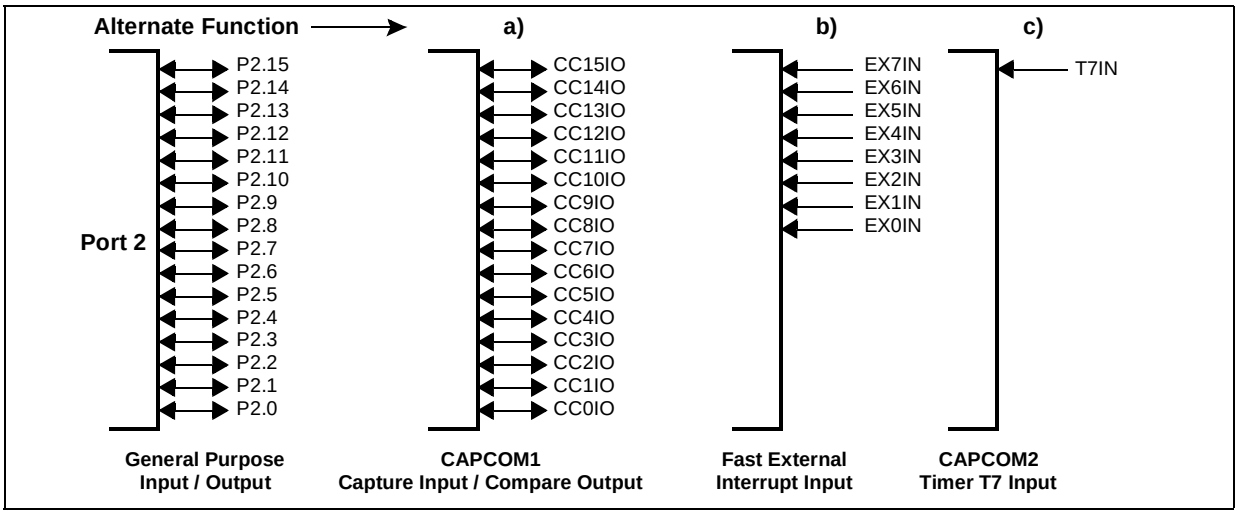
The upper eight Port 2 lines (P2.15...P2.8) also can serve as Fast External Interrupt inputs from EX0IN to EX7IN. (Fast external interrupt sampling rate is 25ns at 40MHz CPU clock).

P2.15 in addition serves as input for CAPCOM2 timer T7 (T7IN).

The table below summarizes the alternate functions of Port 2.

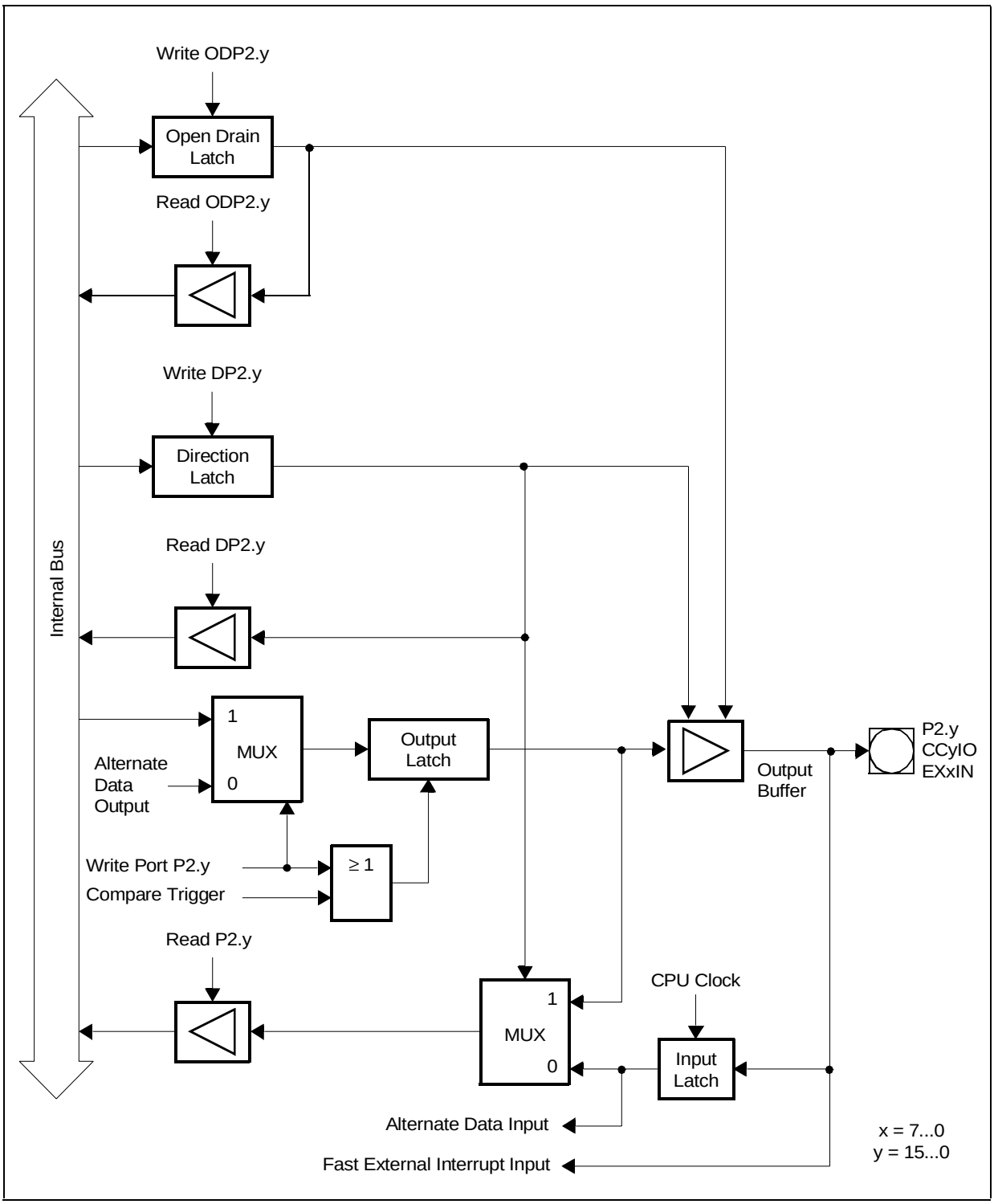
Port 2 Pin	Alternate Function a)	Alternate Function b)		Alternate Function c)
P2.0	CC0IO	-		-
P2.1	CC1IO	-		-
P2.2	CC2IO	-		-
P2.3	CC3IO	-		-
P2.4	CC4IO	-		-
P2.5	CC5IO	-		-
P2.6	CC6IO	-		-
P2.7	CC7IO	-		-
P2.8	CC8IO	EX0IN	Fast External Interrupt 0 Input	-
P2.9	CC9IO	EX1IN	Fast External Interrupt 1 Input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

Figure 33 : Port 2 I/O and Alternate Functions



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

Figure 34 : Block Diagram of a Port 2 Pin



12.5 - 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, P3.14 and P3.12 do not support open drain mode).

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

P3 (FFC4h / E2h) SFR Reset Value: 0000h

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

Bit	Function
P3.y	Port data register P3 bit y

DP3 (FFC6h / E3h) SFR Reset Value: 0000h

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

Bit	Function
DP3.y	**Port direction register DP3 bit y** DP3.y = 0: Port line P3.y is an input (high-impedance) DP3.y = 1: Port line P3.y is an output

ODP3 (F1C6h / E3h) SFR Reset Value: 0000h

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

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

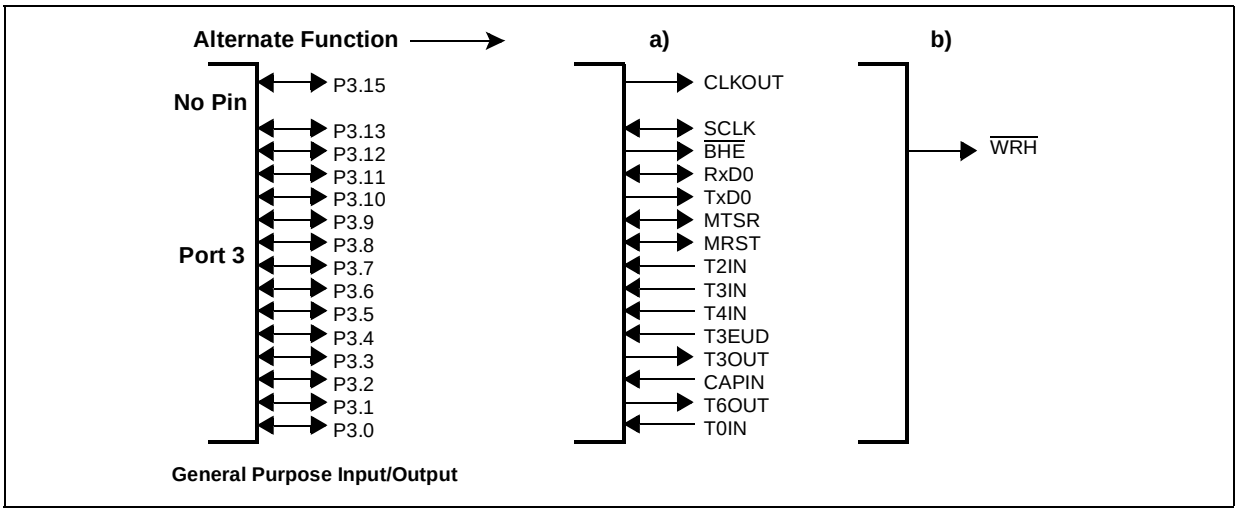
12.5.1 - Alternate Functions of Port 3

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

Table 16 : 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	MRST	SSC Master Receive / Slave Transmit
P3.9	MTSR	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	SCLK	SSC Shift Clock Input/Output
P3.14	---	No pin assigned!
P3.15	CLKOUT	System Clock Output

Figure 35 : Port 3 I/O and Alternate Functions



The port structure of the Port 3 pins depends on their alternate function (see Figure 36).

When the on-chip peripheral associated with a Port 3 pin is configured to use the alternate input function, it reads the input latch, which represents the state of the pin, via the line labeled “Alternate Data Input”. Port 3 pins with alternate input functions are:

T0IN, T2IN, T3IN, T4IN, T3EUD and CAPIN.

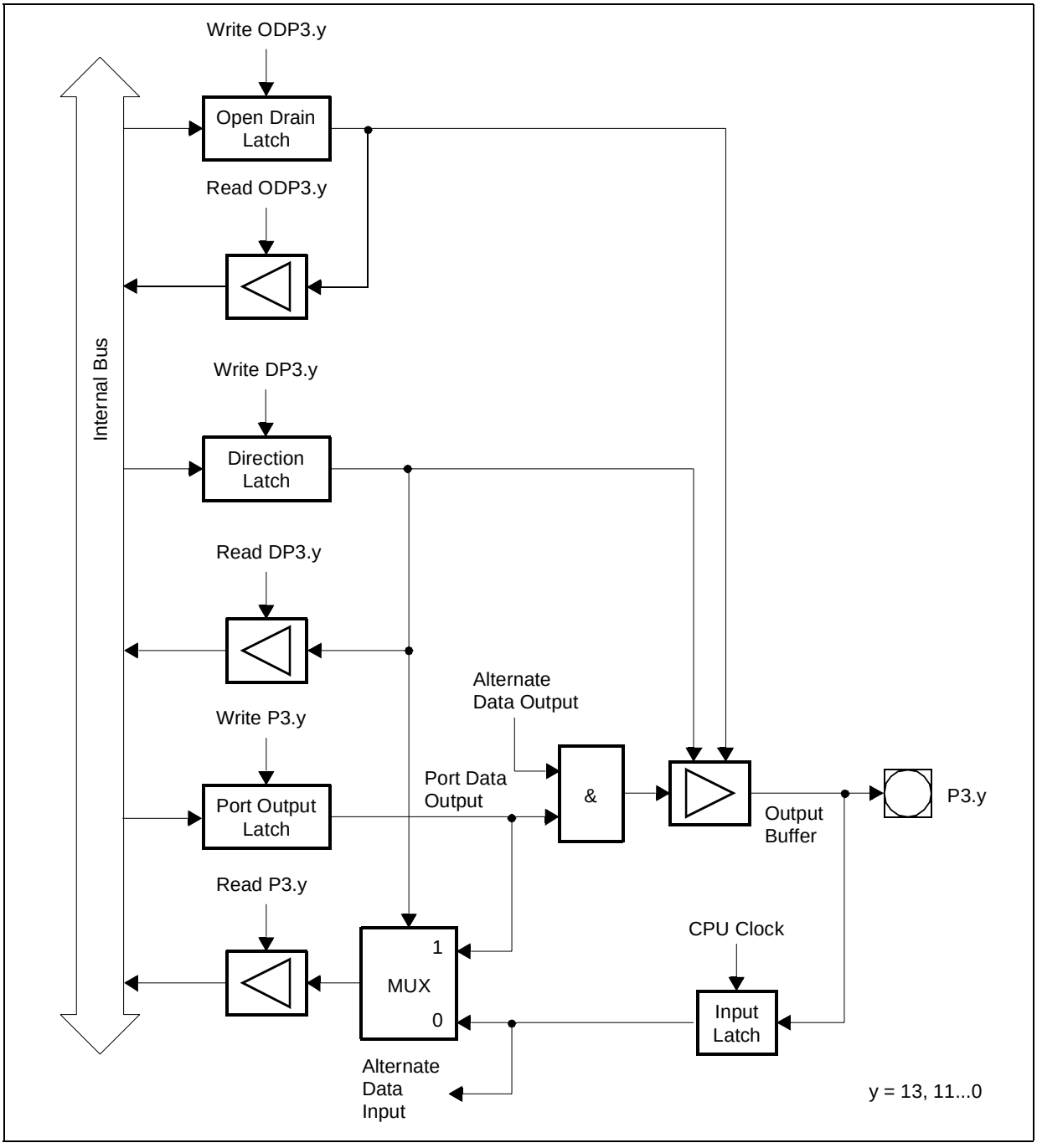
When the on-chip peripheral associated with a Port 3 pin is configured to use the alternate output function, its “Alternate Data Output” line is ANDed with the port output latch line. When using these alternate functions, the user must set the direction of the port line to output (DP3.y=1) and must set the port output latch (P3.y=1). Otherwise the pin is in its high-impedance state (when configured as input) or the pin is stuck at '0' (when the port output latch is cleared).

When the alternate output functions are not used, the “Alternate Data Output” line is in its inactive state, which is a high level ('1'). Port 3 pins with alternate output functions are: T6OUT, T3OUT, TxD0 and CLKOUT.

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

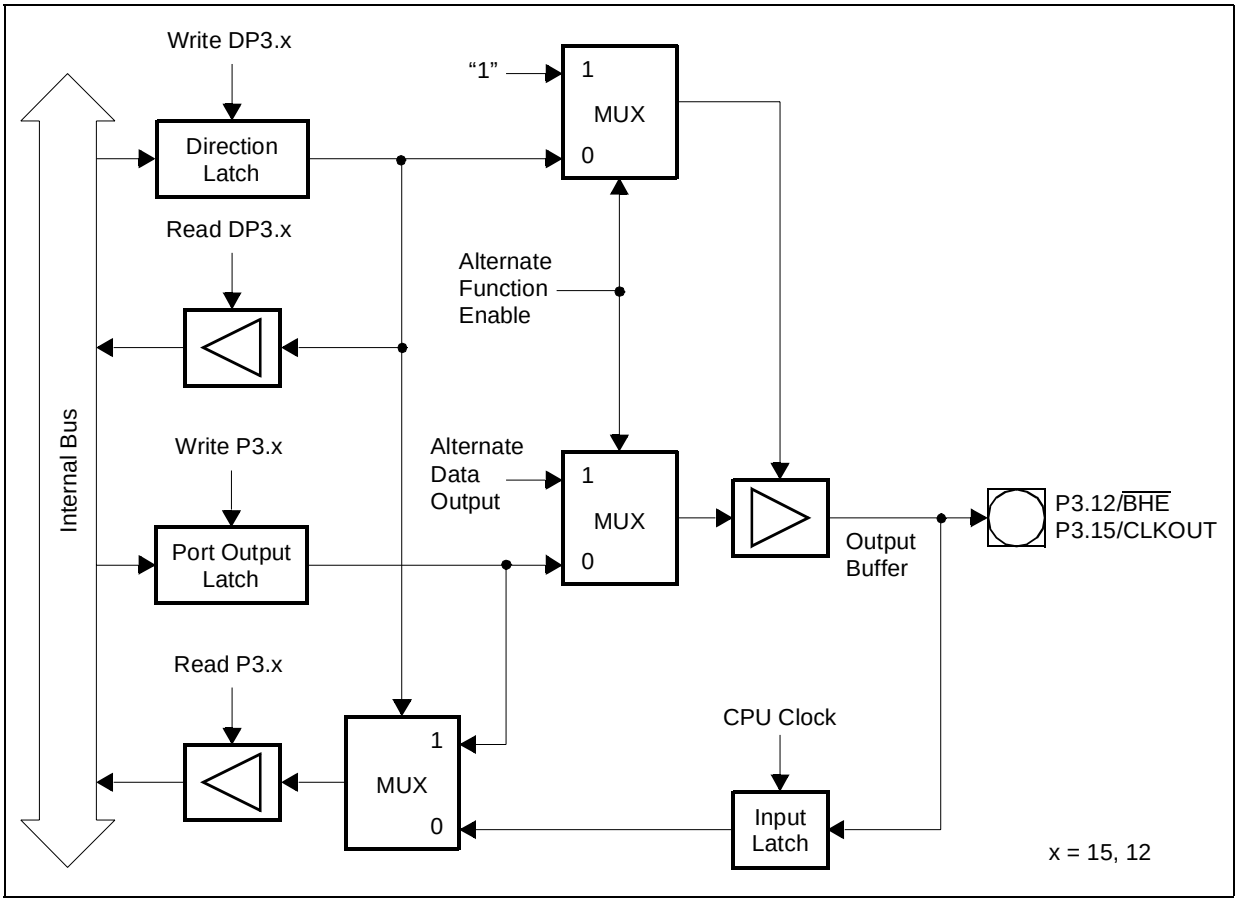
Note: Enabling the CLKOUT function automatically enables the P3.15 output driver. Setting bit DP3.15='1' is not required.

Figure 36 : Block Diagram of Port 3 Pin with Alternate Input or Alternate Output Function



Pin P3.12 (BHE/WRH) is another pin with an alternate output function, however, its structure is slightly different (see figure Figure 37). After reset the BHE or WRH function must be used depending on the system start-up configuration. In either of these cases, there is no possibility to program any port latches before. Thus, the appropriate alternate function is selected automatically. If BHE/WRH is not used in the system, this pin can be used for general purpose I/O by disabling the alternate function (BYTDIS = '1' / WRCFG='0').

Figure 37 : Block Diagram of Pins P3.15 (CLKOUT) and P3.12 (BHE/WRH)



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

12.6 - 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.

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

Bit	Function
P4.y	Port data register P4 bit y

DP4 (FFCAh / E5h) SFR Reset Value: - - 00h

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

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

For CAN configuration support (see Chapter 15 - CAN Modules), Port 4 has a new open drain function, controlled with the new ODP4 register:

ODP4 (F1CAh / E5h) SFR Reset Value: - - 00h

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

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

Note: Only bits 6 and 7 are implemented, all other bits will be read as “0”.

12.6.1 - Alternate Functions of Port 4

During external bus cycles that use segmentation (i.e. an address space above 64K Bytes) a number of Port 4 pins may output the segment address lines. The number of pins used for segment address output determines the directly accessible external address space.

The other pins of Port 4 may be used for general purpose I/O. If segment address lines are selected, the alternate function of Port 4 may be necessary to access e.g. external memory directly after reset. For this reason Port 4 will be switched to this alternate function automatically.

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

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

The table below summarizes the alternate functions of Port 4 depending on the number of selected segment address lines (coded via bitfield SALSEL)..

Port 4 Pin	Std. Function SALSEL=01 64 KB	Altern. Function SALSEL=11 256KB	Altern. Function SALSEL=00 1MB	Altern. Function SALSEL=10 16MB
P4.0	GPIO	Seg. Address A16	Seg. Address A16	Seg. Address A16
P4.1	GPIO	Seg. Address A17	Seg. Address A17	Seg. Address A17
P4.2	GPIO	GPIO	Seg. Address A18	Seg. Address A18
P4.3	GPIO	GPIO	Seg. Address A19	Seg. Address A19
P4.4	GPIO/CAN2_RxD	GPIO/CAN2_RxD	GPIO/CAN2_RxD	Seg. Address A20
P4.5	GPIO/CAN1_RxD	GPIO/CAN1_RxD	GPIO/CAN1_RxD	Seg. Address A21
P4.6	GPIO/CAN1_TxD	GPIO/CAN1_TxD	GPIO/CAN1_TxD	Seg. Address A22
P4.7	GPIO/CAN2_TxD	GPIO/CAN2_TxD	GPIO/CAN2_TxD	Seg. Address A23

Figure 38 : Port 4 I/O and Alternate Functions

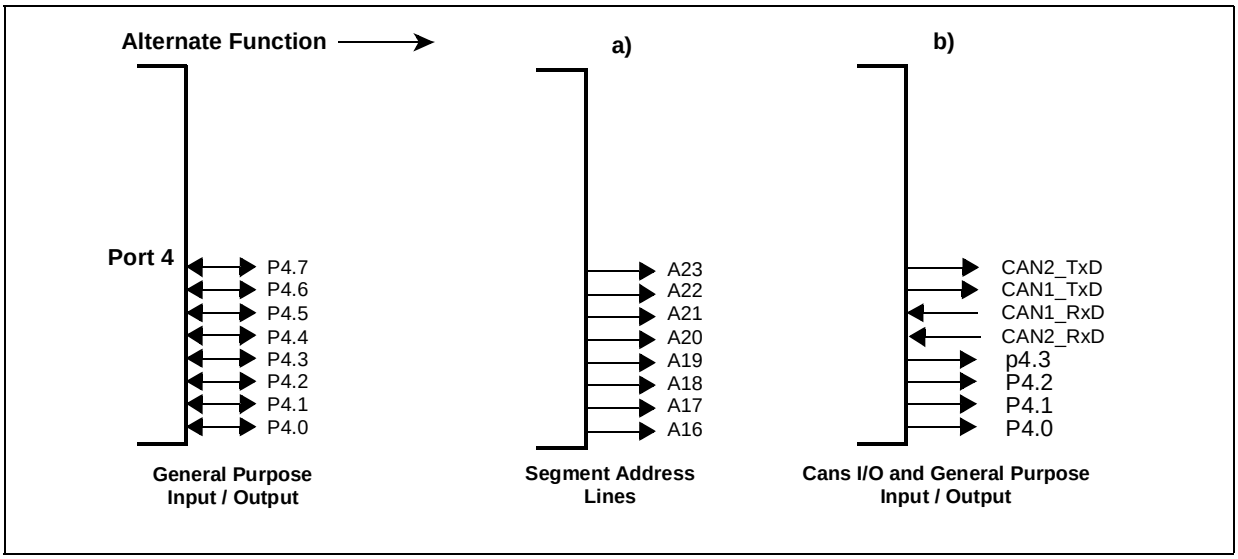


Figure 39 : Block Diagram of a Port 4 Pin

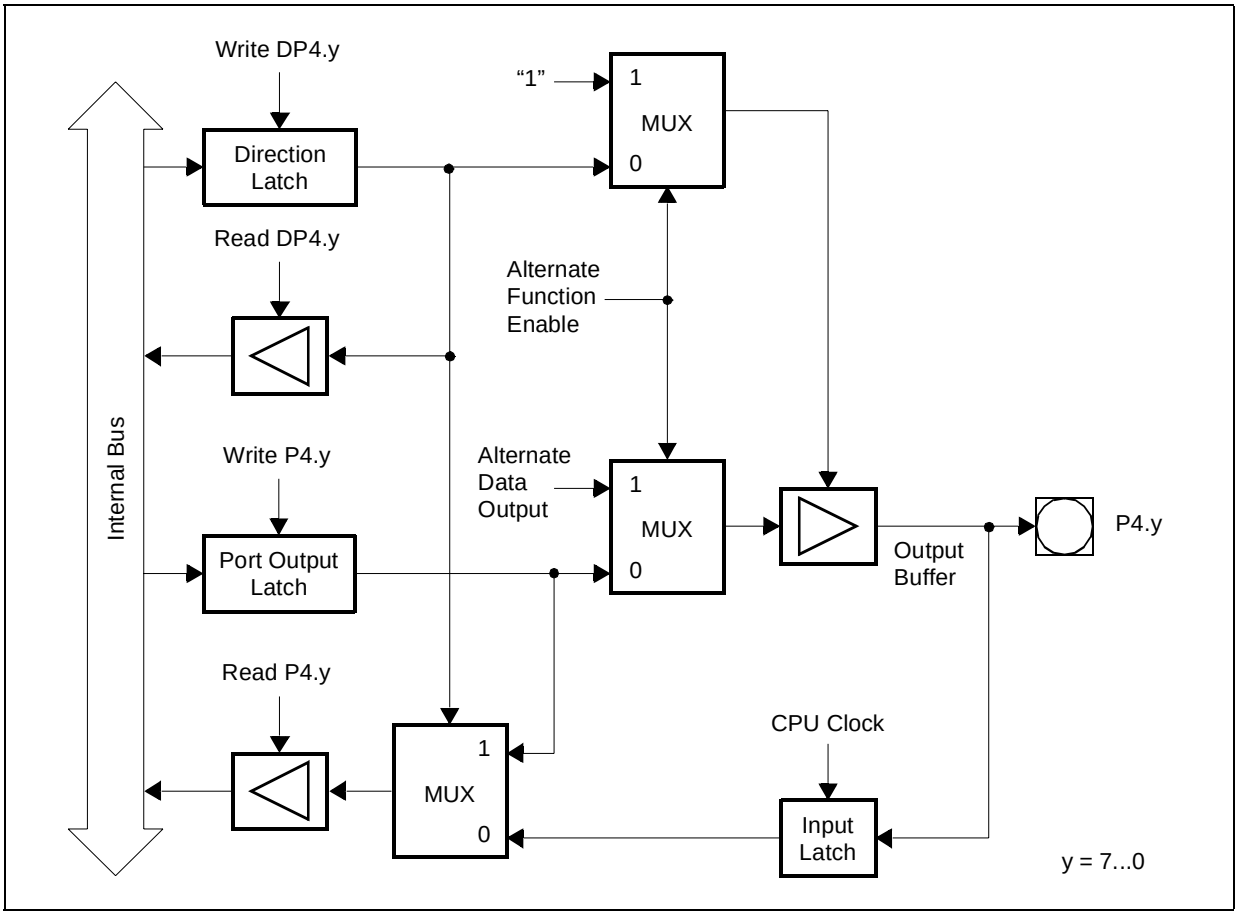


Figure 40 : Block Diagram of P4.4 and P4.5 Pins

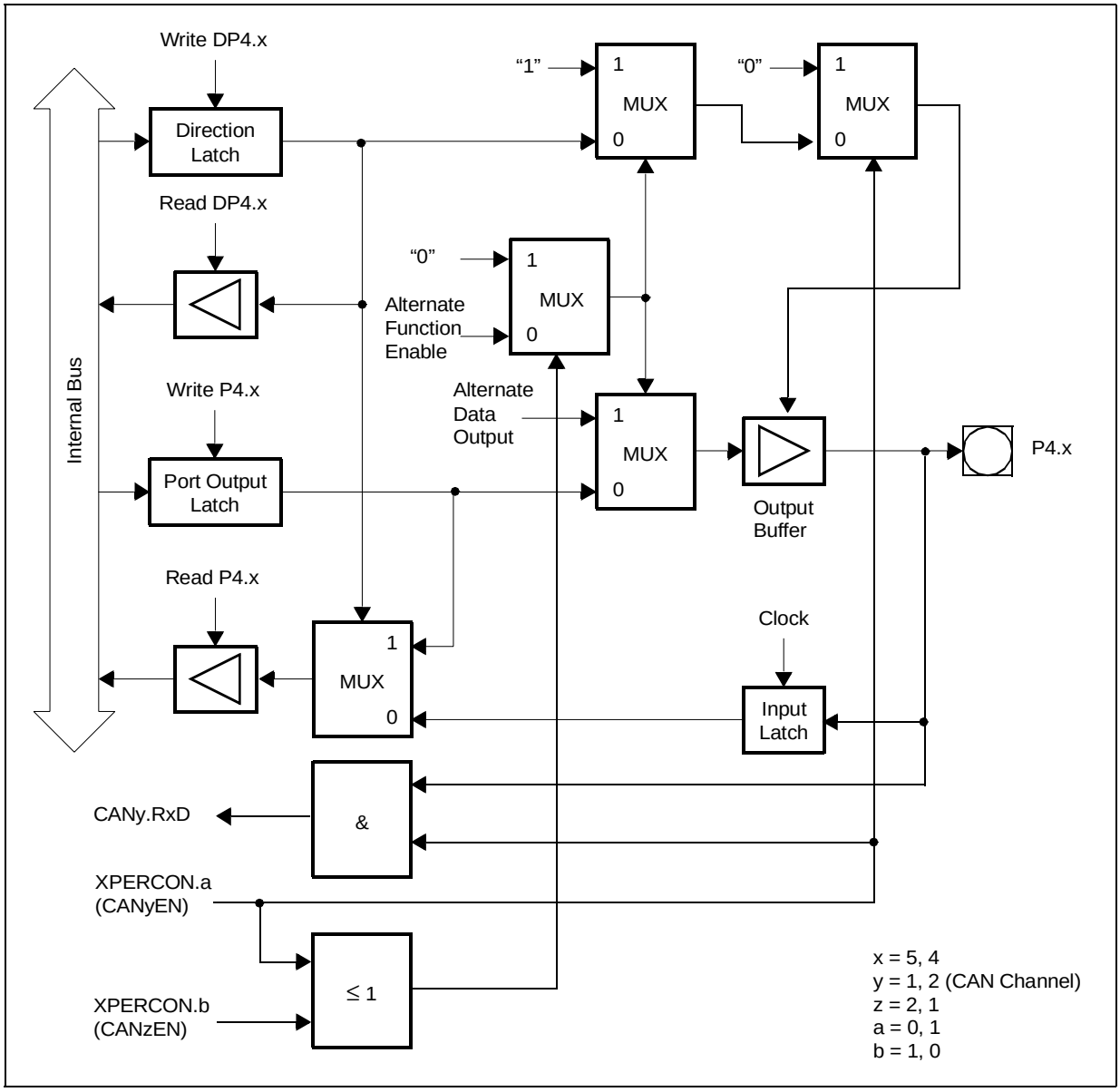
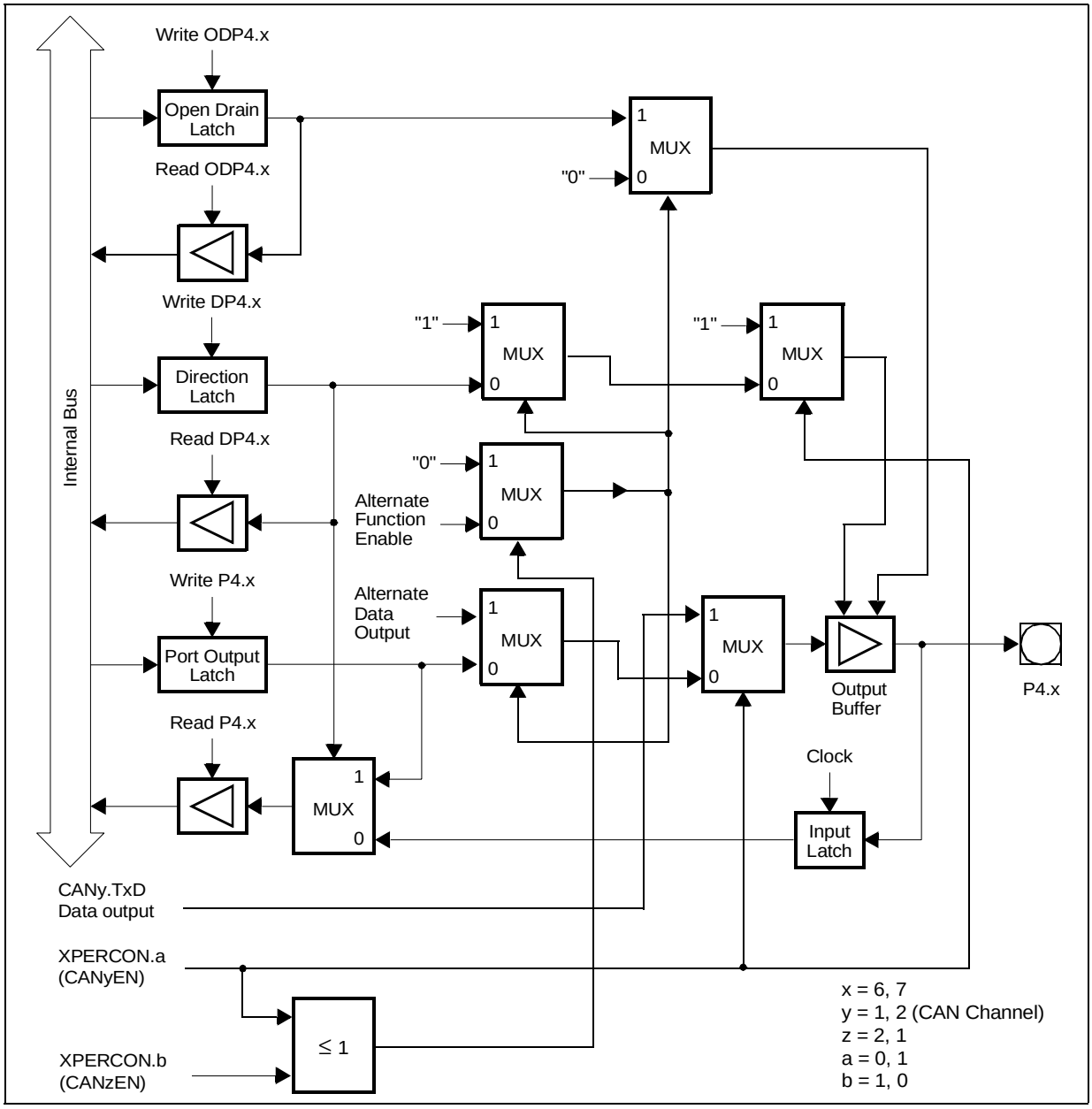


Figure 41 : Block Diagram of P4.6 and P4.7 Pins



12.7 - Port 5

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

P5 (FFA2h / D1h) SFR Reset Value: XXXXh

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

Bit	Function
P5.y	Port data register P5 bit y (Read only)

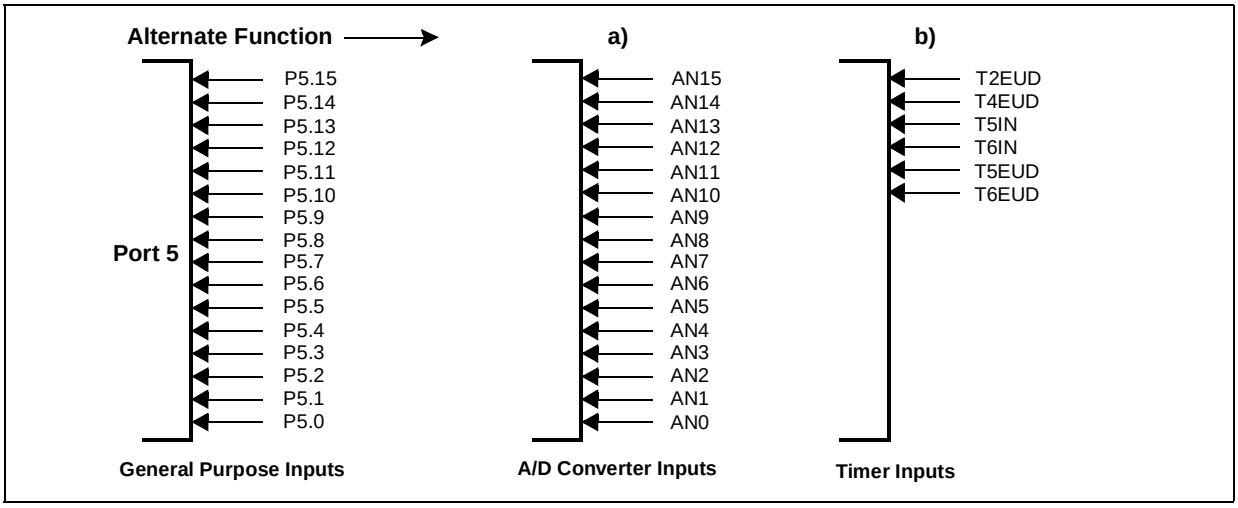
Alternate Functions of Port 5

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

Table 17 : Port 5 Alternate Functions

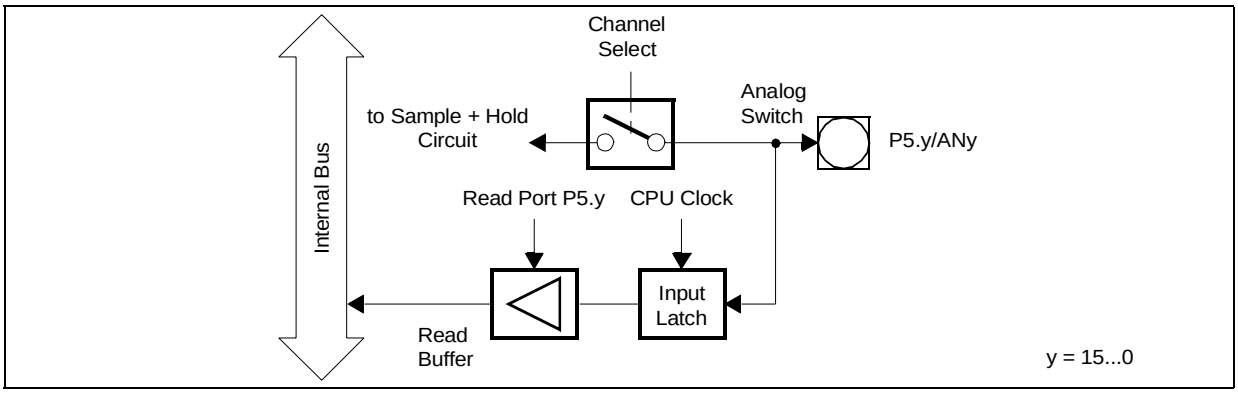
Port 5 Pin	Alternate Function a)	Alternate Function b)	
P5.0	Analog Input AN0	-	
P5.1	Analog Input AN1	-	
P5.2	Analog Input AN2	-	
P5.3	Analog Input AN3	-	
P5.4	Analog Input AN4	-	
P5.5	Analog Input AN5	-	
P5.6	Analog Input AN6	-	
P5.7	Analog Input AN7	-	
P5.8	Analog Input AN8	-	
P5.9	Analog Input AN9	-	
P5.10	Analog Input AN10	T6EUD	Timer 6 ext. Up/Down Input
P5.11	Analog Input AN11	T5EUD	Timer 5 ext. 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 ext. Up/Down Input
P5.15	Analog Input AN15	T2EUD	Timer 2 ext. Up/Down Input

Figure 42 : Port 5 I/O and Alternate Functions



Port 5 pins have a special port structure (see Figure 43), first because it is an input only port, and second because the analog input channels are directly connected to the pins rather than to the input latches.

Figure 43 : Block Diagram of a Port 5 Pin



12.7.1 - Port 5 Schmitt Trigger Analog Inputs

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

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

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

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

12.8 - Port 6

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

P6 (FFCCh / E6h) SFR Reset Value: - - 00h

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

Bit	Function
P6.y	Port data register P6 bit y

DP6 (FFCEh / E7h) SFR Reset Value: - - 00h

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

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

ODP6 (F1CEh / E7h) ESFR Reset Value: - - 00h

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

Bit	Function
ODP6.y	**Port 6 Open Drain control register bit y** ODP6.y = 0: Port line P6.y output driver in push/pull mode ODP6.y = 1: Port line P6.y output driver in open drain mode

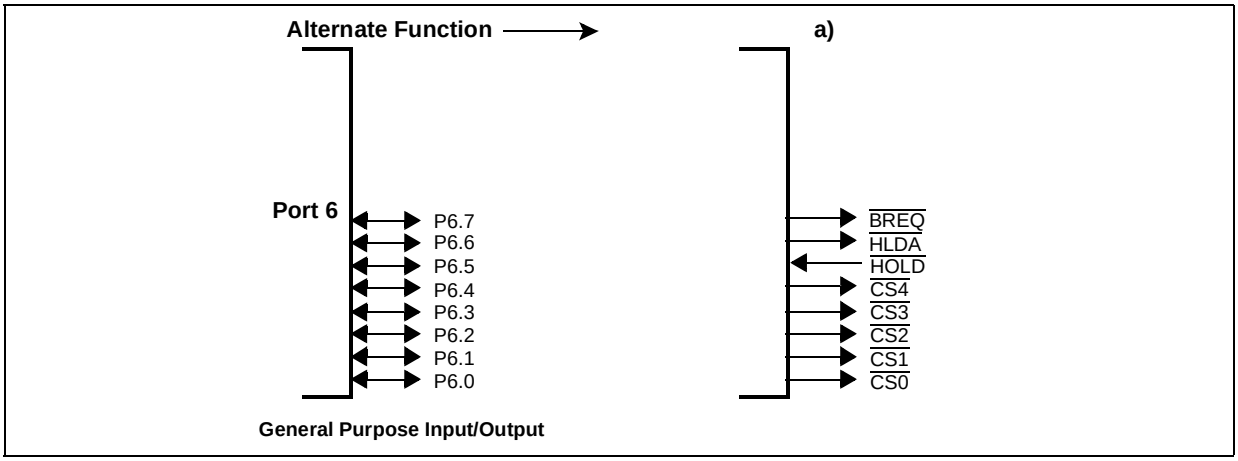
12.8.1 - Alternate Functions of Port 6

A programmable number of chip select signals (CS4...CS0) derived from the bus control registers (BUSCON4...BUSCON0) can be output on the 5 pins of Port 6. The number of chip select signals is selected via PORT0 during reset. The selected value can be read from bitfield CSSEL in register RP0H (read only) e.g. in order to check the configuration during run time. The table below summarizes the alternate functions of Port 6 depending on the number of selected chip select lines (coded via bitfield CSSEL).

Table 18 : Port 6 Alternate Functions

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

Figure 44 : Port 6 I/O and Alternate Functions

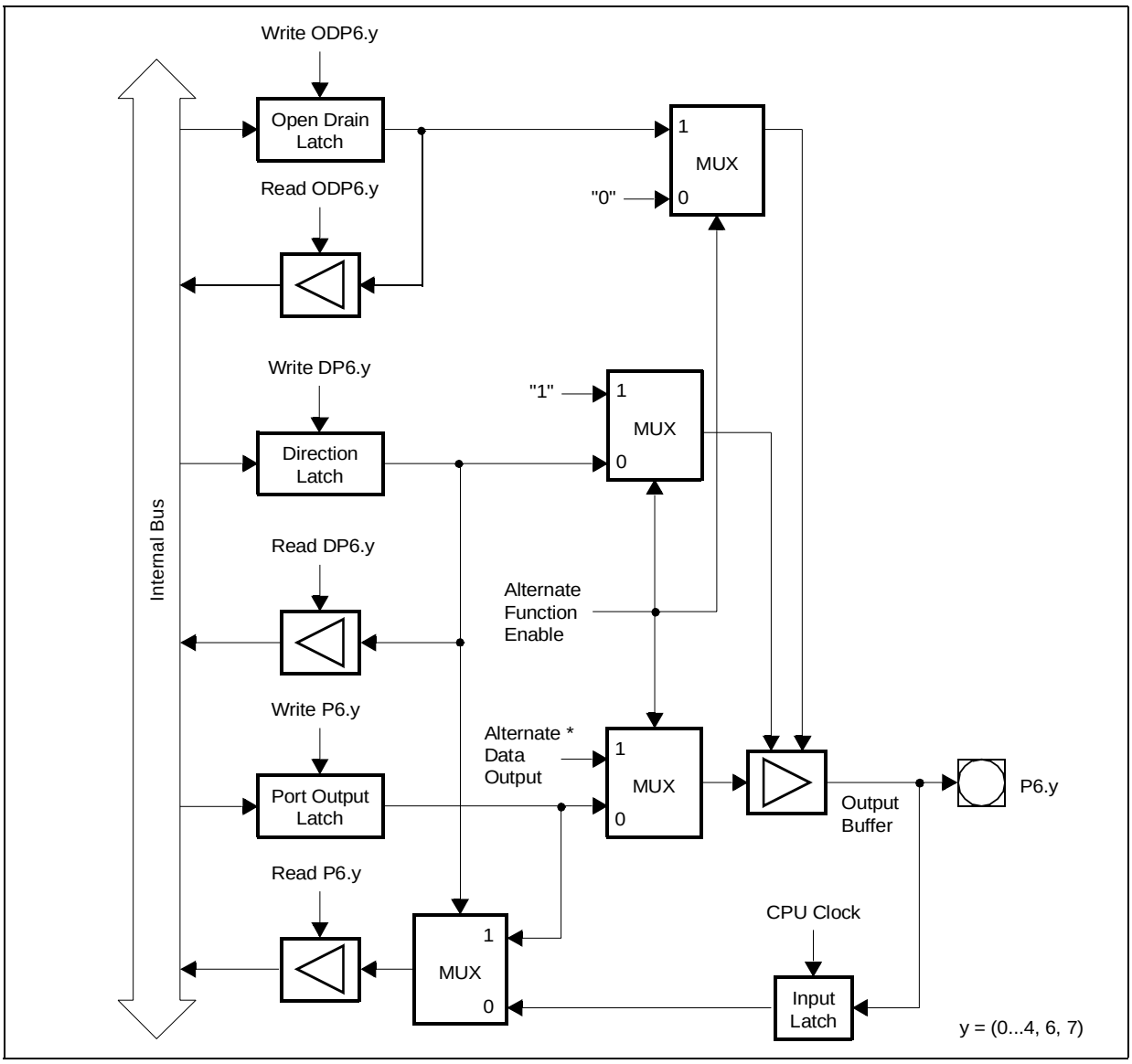


The chip select lines of Port 6 have an internal weak pull-up device. This device is switched on during reset. This feature is implemented to drive the chip select lines high during reset in order to avoid multiple chip selection.

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

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

Figure 45 : Block Diagram of Port 6 Pins with an Alternate Output Function



** P6.5 has only alternate input function.*

12.9 - Port 7

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

P7 (FFD0h / E8h) SFR Reset Value: - - 00h

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

P7.y	**Port data register P7 bit y**

DP7 (FFD2h / E9h) SFR Reset Value: - - 00h

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

DP7.y	**Port direction register DP7 bit y** DP7.y = 0: Port line P7.y is an input (high impedance) DP7.y = 1: Port line P7.y is an output

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

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

ODP7.y	**Port 7 Open Drain control register bit y** ODP7.y = 0: Port line P7.y output driver in push-pull mode ODP7.y = 1: Port line P7.y output driver in open drain mode

12.9.1 - Alternate Functions of Port 7

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

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

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

The lower 4 lines of Port 7 (P7.3...P7.0) serve as outputs from the PWM module (POUT3...POUT0). At these pins the value of the respective port output latch is XORed with the value of the PWM output rather than ANDed, as the other pins do. This allows to use the alternate output value either as it is (port latch holds a '0') or invert its level at the pin (port latch holds a '1').

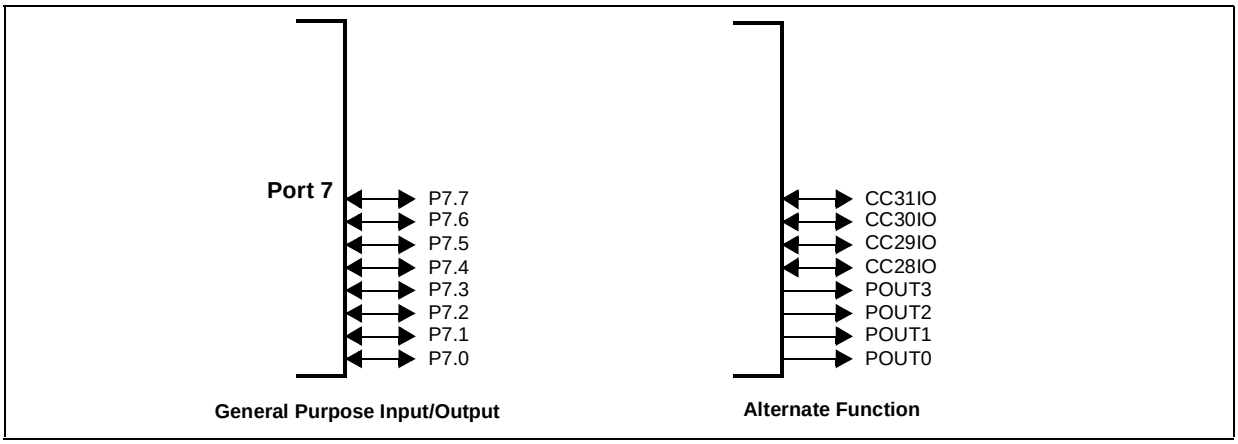
Note that the PWM outputs must be enabled via the respective PENx bits in PWMCON1.

The table below summarizes the alternate functions of Port 7.

Table 19 : 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	CC28IO	Capture input / compare output channel 28
P7.5	CC29IO	Capture input / compare output channel 29
P7.6	CC30IO	Capture input / compare output channel 30
P7.7	CC31IO	Capture input / compare output channel 31

Figure 46 : Port 7 I/O and Alternate Functions



The port structures of Port 7 differ in the way the output latches are connected to the internal bus and to the pin driver (see the two Figure 47). Pins P7.3...P7.0 (POUT3...POUT0) XOR the alternate data output with the port latch output, which allows to use the alternate data directly or inverted at the pin driver.

Figure 47 : Block Diagram of Port 7 Pins P7.3...P7.0

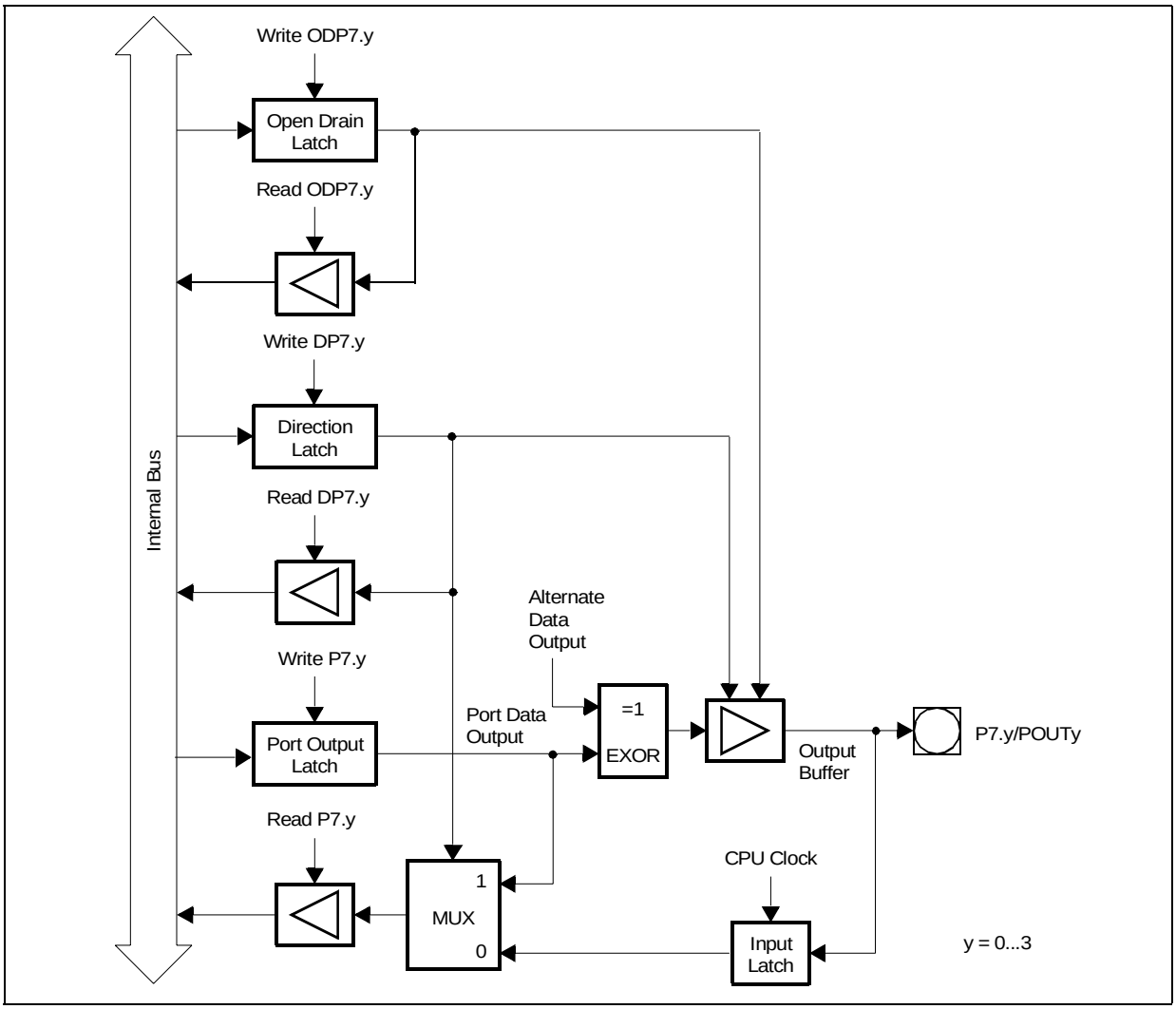
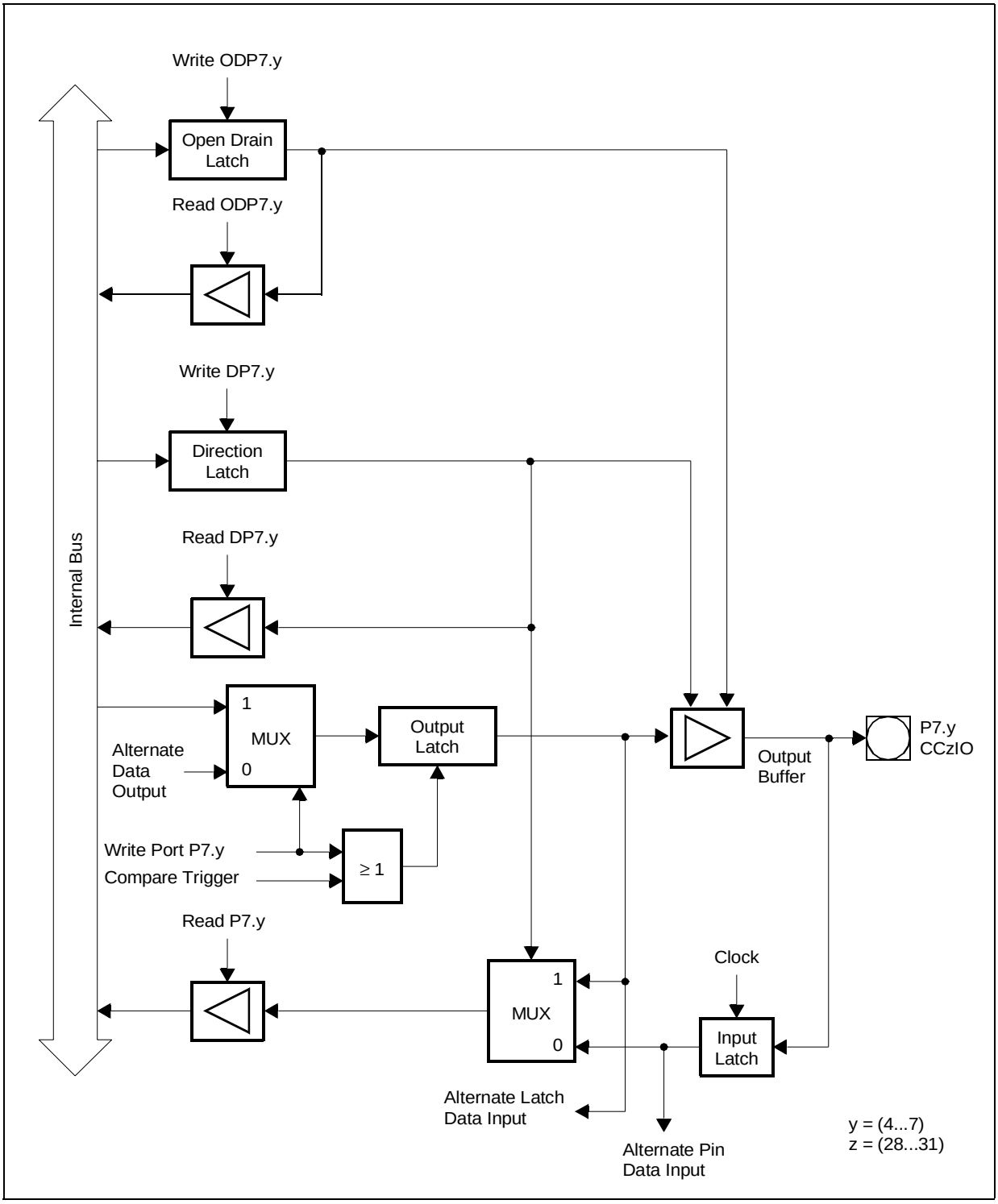


Figure 48 : Block Diagram of Port 7 Pins P7.7...P7.4



12.10 - Port 8

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

P8 (FFD4h / EAh) SFR Reset Value: - - 00h

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

P8.y	**Port data register P8 bit y**

DP8 (FFD6h / EBh) SFR Reset Value: - - 00h

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

DP8.y	**Port direction register DP8 bit y** DP8.y = 0: Port line P8.y is an input (high impedance) DP8.y = 1: Port line P8.y is an output

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

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

ODP8.y	**Port 8 Open Drain control register bit y** ODP8.y = 0: Port line P8.y output driver in push-pull mode ODP8.y = 1: Port line P8.y output driver in open drain mode

12.10.1 - Alternate Functions of Port 8

The 8 lines of Port 8 (P8.7...P8.0) serve as capture inputs or compare outputs (CC23IO...CC16IO) for the CAPCOM2 unit.

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

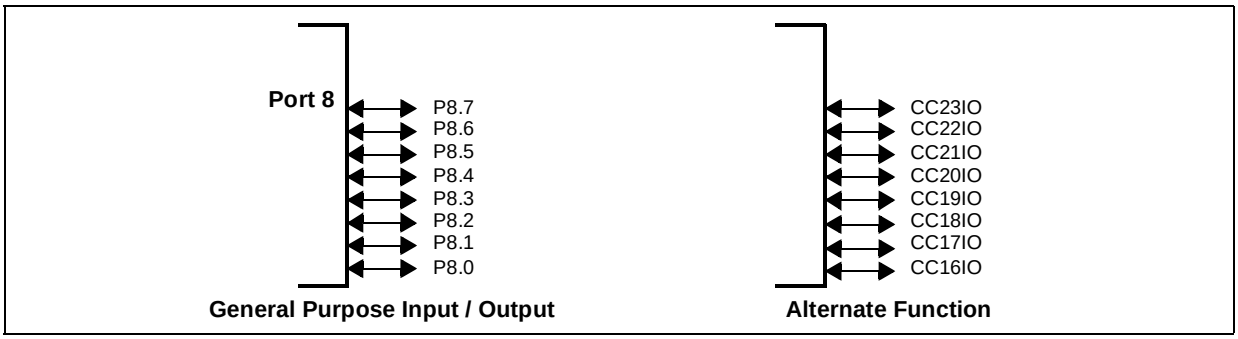
As all other capture inputs, the capture input function of pins P8.7...P8.0 can also be used as external interrupt inputs (200 ns sample rate at 40MHz CPU clock).

The Table 20 summarizes the alternate functions of Port 8.

Table 20 : Port 8 Alternate Functions

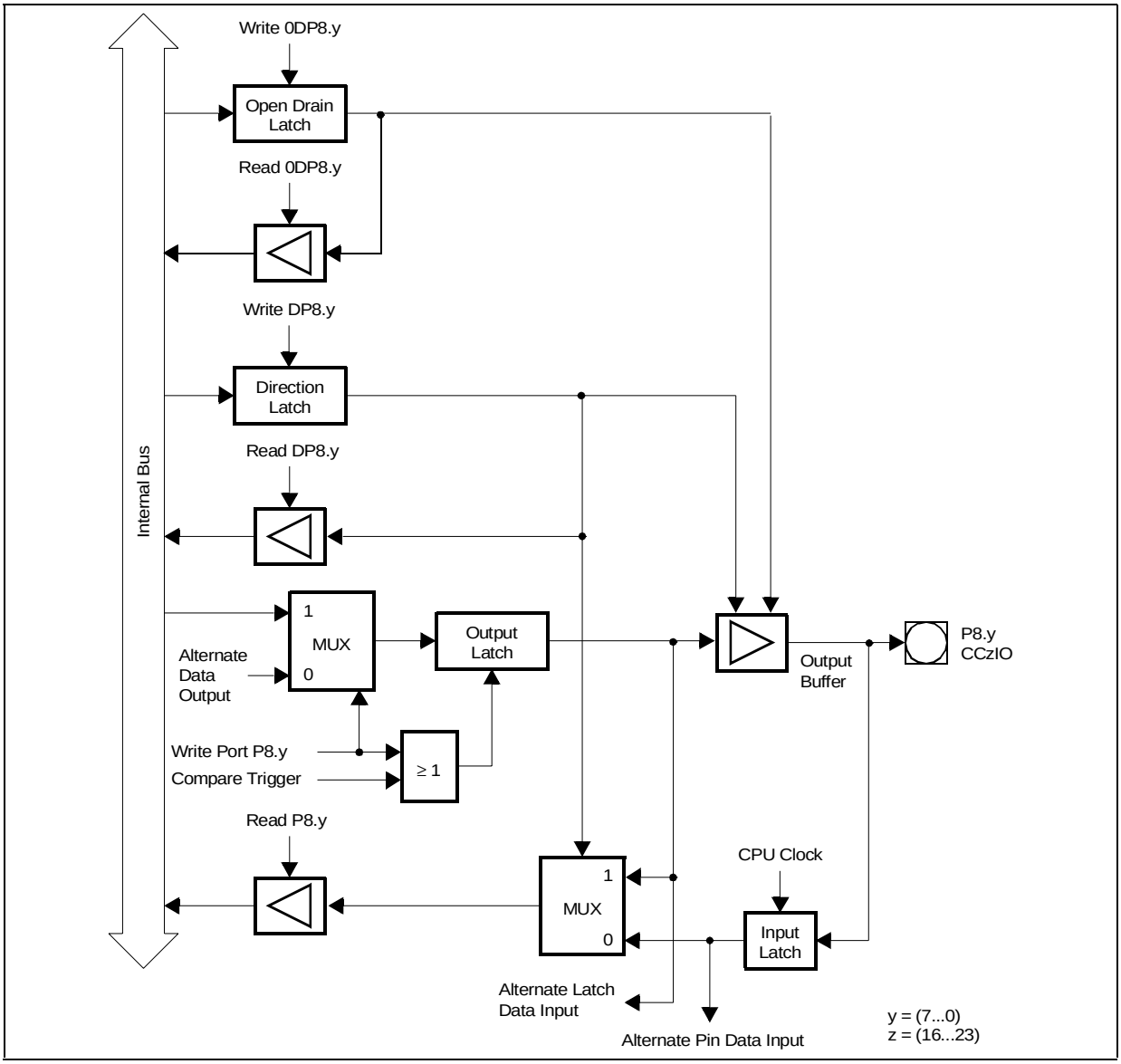
Port 7	Alternate Function	
P8.0	CC16IO	Capture input / compare output channel 16
P8.1	CC17IO	Capture input / compare output channel 17
P8.2	CC18IO	Capture input / compare output channel 18
P8.3	CC19IO	Capture input / compare output channel 19
P8.4	CC20IO	Capture input / compare output channel 20
P8.5	CC21IO	Capture input / compare output channel 21
P8.6	CC22IO	Capture input / compare output channel 22
P8.7	CC23IO	Capture input / compare output channel 23

Figure 49 : Port 8 I/O and Alternate Functions



The port structures of Port 8 differ in the way the output latches are connected to the internal bus and to the pin driver (see the Figure 50). Pins P8.7...P8.0 (CC23IO...CC16IO) combine internal bus data and alternate data output before the port latch input, as do the Port 2 pins.

Figure 50 : Block Diagram of Port 8 Pins P8.7...P8.0



12.11 - XPort 9

The XPort9 is enabled by setting XPEN bit 2 of the SYSCON register and XPORT9EN bit 3 of the new XPERCON register. On the XBUS interface, the register 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 mode via the open drain control register XODP9.

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.

XP9 (C100h) 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

Bit	Function
XP9.y	**Port data register XP9 bit y**

XP9SET (C102h) 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

Bit	Function
XP9SET.y	Writing a ‘1’ will set the corresponding bit in XP9 register, Writing a ‘0’ has no effect.

XP9CLR (C104h) 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

Bit	Function
XP9CLR.y	Writing a ‘1’ will clear the corresponding bit in XP9 register, Writing a ‘0’ has no effect.

XDP9 (C200h) 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

Bit	Function
XDP9.y	**Port direction register XDP9 bit y** XDP9.y = 0: Port line XP9.y is an input (high-impedance) XDP9.y = 1: Port lineX P9.y is an output

XDP9SET (C202h) Reset Value: 0000h

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

Bit	Function
XDP9SET.y	Writing a ‘1’ will set the corresponding bit in XDP9 register, Writing a ‘0’ has no effect.

XDP9CLR (C204h) Reset Value: 0000h

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

Bit	Function
XDP9CLR.y	Writing a ‘1’ will clear the corresponding bit in XDP9 register, Writing a ‘0’ has no effect.

XODP9 (C300h) 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**	**XODP 9.2**	**XODP9 .1**	**XODP9 .0**
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Function
XODP9.y	**Port 9 Open Drain control register bit y** XODP9.y = 0: Port line XP9.y output driver in push/pull mode XODP9.y = 1: Port line XP9.y output driver in open drain mode

XODP9SET (C302h) Reset Value: 0000h

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

Bit	Function
XODP9SET.y	Writing a ‘1’ will set the corresponding bit in XODP9 register, Writing a ‘0’ has no effect.

XODP9CLR (C304h) Reset Value: 0000h

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

Bit	Function
XODP9CLR.y	Writing a ‘1’ will clear the corresponding bit in XODP9 register, Writing a ‘0’ has no effect.

12.12 - XPort 10

The XPort10 is enabled by setting XPEN bit 2 of the SYSCON register and bit 3 of the new XPERCON register. 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 will be lost.

XP10 (C380h) 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

Bit	Function
XP10.y	**Port data register XP10 bit y (Read only)**

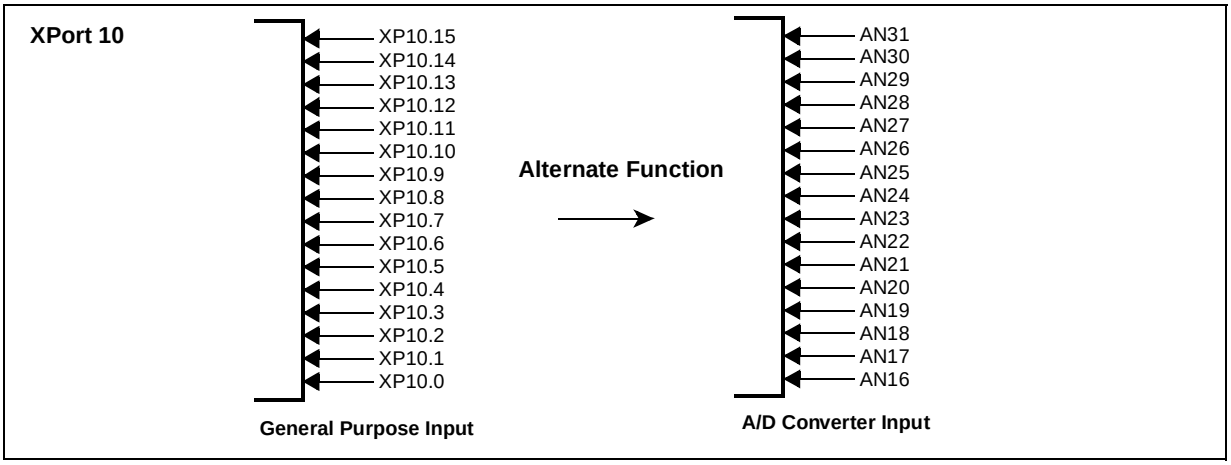
12.12.1 - Alternate Functions of XPort 10

Each line of XPort 10 is also connected to one of the multiplexer of the Analog/Digital Converter. All port lines (XP10.15...XP10.0) can accept analog signals (AN31...AN16) that can be converted by the ADC. No special programming is required for pins that shall be used as analog inputs. The Table 21 summarizes the alternate functions of XPort 10.

Table 21 : XPort 10 Alternate Functions

XPort 10 Pin	Alternate Function
P10.0	Analog Input AN16
P10.1	Analog Input AN17
P10.2	Analog Input AN18
P10.3	Analog Input AN19
P10.4	Analog Input AN20
P10.5	Analog Input AN21
P10.6	Analog Input AN22
P10.7	Analog Input AN23
P10.8	Analog Input AN24
P10.9	Analog Input AN25
P10.10	Analog Input AN26
P10.11	Analog Input AN27
P10.12	Analog Input AN28
P10.13	Analog Input AN29
P10.14	Analog Input AN30
P10.15	Analog Input AN31

Figure 51 : PORT10 I/O and Alternate Functions



12.12.2 - New Disturb Protection on Analog Inputs

A new register is provided for additional disturb protection support on analog inputs for Port XP10:

XP10DIDIS (C382h) Reset Value: 0000h

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

Bit	Function	
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)

13 - A/D CONVERTER

13.1 - A/D Converter Module

A 10-bit A/D converter with 2 x 16 multiplexed input channels and a sample and hold circuit is integrated on-chip. This A/D Converter does not have the self-calibration feature. Thus, guaranteed Total Unadjusted Error is ± 2 LSB. Refer to Section 20.3.1 - A/D Converter Characteristics for detailled characteristics. The sample time (for loading the capacitors) and the conversion time is programmable and can be adjusted to the external circuitry. Convertion time is fully equivalent to the one of previous generation A/D self-calibrated Converter.

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

Overrun error detection/protection is controlled by the ADDAT register. Either an interrupt request is generated when the result of a previous conversion has not been read from the result register at the time the next conversion is complete, or the next conversion is suspended until the previous result has been read. For applications which require less than 16 analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converter of the ST10F280 supports four different conversion modes:

Single channel conversion mode the analog level on a specified channel is sampled once and converted to a digital result.

Single channel continuous mode the analog level on a specified channel is repeatedly sampled and converted without software intervention.

Auto scan mode the analog levels on a pre-specified number of channels are sequentially sampled and converted.

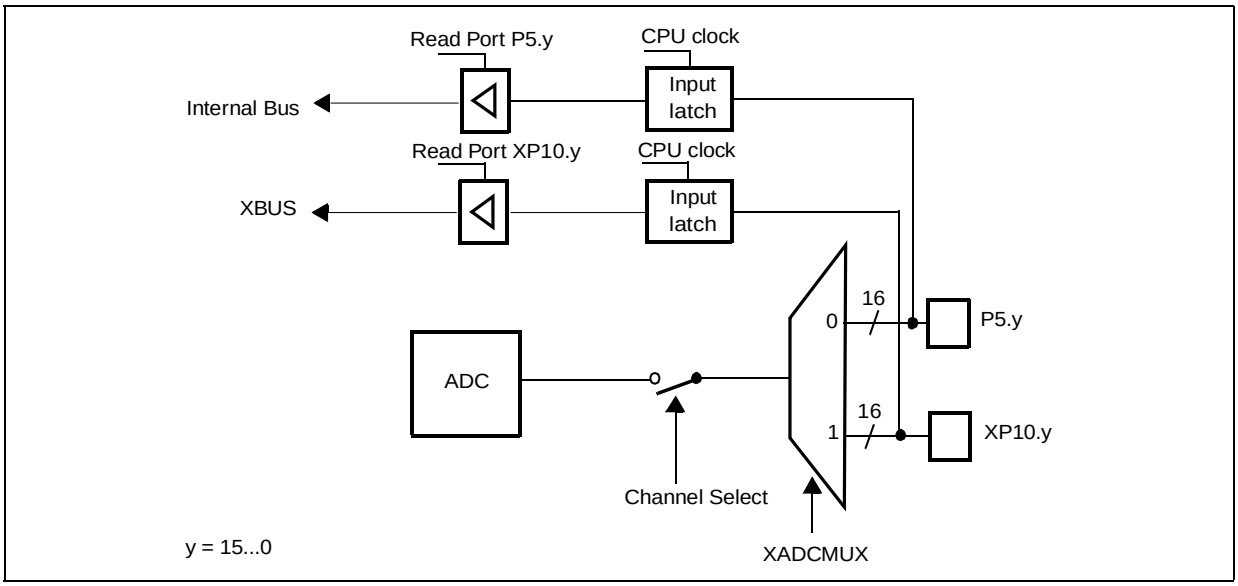
Auto scan continuous mode the number of pre-specified channels is repeatedly sampled and converted.

Channel Injection Mode injects a channel into a running sequence without disturbing this sequence. The peripheral event controller stores the conversion results in memory without entering and exiting interrupt routines for each data transfer.

13.2 - Multiplexage of two blocks of 16 Analog Inputs

The ADC can manage 16 analog inputs, so to increase its capability, a new XADCMUX register is added to control the multiplexage between the first block of 16 channels on Port5 and the second block of 16 channels on XPort10. The conversion result register stays identical and only a software management can determine the block in use.

Figure 52 : Block Diagram



The XADCMUX register is enabled by setting XPEN bit 2 of the SYSCON register and bit 3 of the new XPERCON register

XADCMUX (C384h) Reset Value: 0000h

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

Bit	Function	
XADCMUX.0	0	Default configuration,analog inputs on port P5.y can be converted
	1	Analog inputs on port XP10.y can be converted

13.3 - XTIMER Peripheral (trigger for ADC channel injection)

This new peripheral is dedicated for the Channel Injection Mode of the A/D converter. This mode injects a channel into a running sequence without disturbing this sequence. The peripheral event controller stores the conversion results in memory without entering and exiting interrupt routines for each data transfer.

A channel injection can be triggered by an event on Capture/Compare CC31 (Port P7.7) of the CAPCOM2 unit.

The dedicated output XADCINJ of the XTIMER must be connected externally on the input P7.7/ CC31.

Due to the multiplexed inputs, at a time, the ADC exclusively converts the Port5 inputs or the XPort10 inputs. If one "y" channel has to be used continuously in injection mode, it must be externally hardware connected to the Port5.y and XPort10.y inputs.

The XTIMER peripheral is enabled by setting XPEN bit 2 of the SYSCON register and bit 3 of the new XPERCON register.

13.3.1 - Main Features

The XTIMER features are :

– 16 bits linear timer / 4 bits exponential prescaler

– Counting between 16 bits “start value” and 16 bits “end value”

– Counting period between 4 cycles and 2**33 cycles (100 ns and 214s using 40MHz CPU clock)

– 1 trigger ouput (XADCINJ)

– Programmable functions :

- Internal clock XCLK is derivated from the CPU clock and has the same period
- Up counting / down counting
- Reload enable
- Continue / stop modes

– 4 memory mapped registers :

- Control / prescaler
- Start value
- End value
- Current value

Table 22 : The 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

13.3.2 - Register Description

13.3.2.1 - TCR : Timer Control Register

XTCR (C000h) Reset Value: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	TFP[3:0]				TCM	TIE	TCS	TLE	TUD	TEN
R	R	R	R	R	R	RW				RW	RW	RW	RW	RW	RW

Bit	Function
TEN	**Timer Enable** When TEN = '0', the Timer is **disabled** (reset value). To avoid glitches, it is recommended to modify TCR in 2 steps, first with new values and and second by setting TEN.
TUD	**Timer Up / Down Counting** When TUD = '0', the Timer is counting **"down"** (reset value), ie the TCVR ('current value') register content is decremented. When TUD = '1', the Timer is counting **"up"**, ie the TCVR ('current value') register content is incremented.
TLE	**Timer Load Enable** When the counter has reached its end value (TCVR = TEVR), TCVR is **(re)loaded** with TSVR ('start value') register content when TLE = '1'. When TLE = '0' (reset value), the next state of TCVR depends on TCS bit.
TCS	**Timer Continue / Stop** When TLE = '0' (no load) and when the counter has reached its end value (TCVR = TEVR), the TCVR content continues to increment / decrement according to TUD bit when TCS = '1' (**continue** mode). When TCS = '0' (**stop** mode reset value), TCVR is stopped and its content is frozen.
TIE	**Timer Output Enable** When the counter has reached its end value (TCVR = TEVR), the XADCINJ output is set when TIE = '1'. When TIE = '0' (reset value), XADCINJ output is disabled (= '0').
TCM	**Timer Clock Mode Must be Cleared** TCM = '0' (reset value), the TCVR clock is derived from **internal** XCLK clock according to TFP bits.
TFP[3:0]	**Timer Frequency Prescaler** When TCM = '0' (internal clock), the TCVR register clock is derived from the XCLK clock input by dividing XCLK by 2**(2+ TFP). The coding is as follows : - 0000 : prescaler by 2 (reset value), XCLK divided by 4 - 0001 : prescaler by 4, XCLK divided by 8 - 0010 : prescaler by 8, XCLK divided by 16 - ... - 1111 : prescaler by 2**16, XCLK divided by 2**17

13.3.2.2 - XTSVR :Timer Start Value Register

XTSVR (C002h) Reset Value: 0000h

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

Bit	Function
TSVR[15:0]	**Timer Start Value** TSVR contains the data to be transferred to the TCVR 'Current Value' register when : 1) - TEN = '1' (TIM enable), - TLE = '1' (TIM Load enable), - TCVR = TEVR (count period finished), - TCS = '1' (stop mode disabled). 2) - first counting clock rising edge after the timer start (the timer starts on TEN rising edge).

13.3.2.3 - XTEVR : Timer End Value Register

XTEVR (C004h) Reset Value: 0000h

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

Bit	Function
TEVR[15:0]	**Timer End Value** TEVR contains the data to be compared to the TCVR 'Current Value' register.

13.3.2.4 - XTCVR : Timer Current Value Register

XTCVR (C006h) Reset Value: 0000h

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

Bit	Function
TCVR[15:0]	**Timer Current Value** TCVR contains the current counting value. When TCVR = TEVR, TCVR content is changed according to Table 22. The TCVR clock is derived from internal XCLK clock according to TFP bits when TCM = '0'.

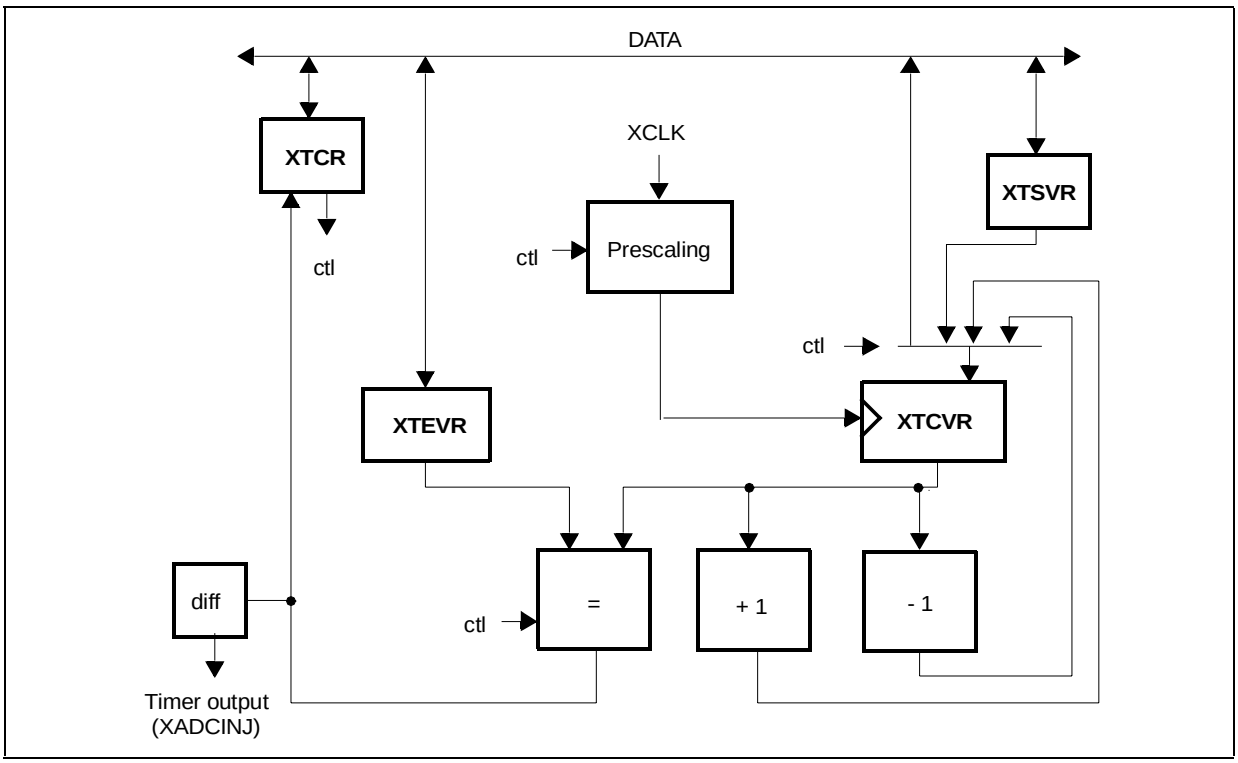
13.3.2.5 - Registers Mapping

Table 23 : Timer Registers Mapping

Address (Hexa)	Register Name	Reset Value (Hexa)	Access
C000h	**XTCR** : Control	0000h	RW
C002h	**XTSVR** : Start Value	0000h	RW
C004h	**XTEVR** : End Value	0000h	RW
C006h	**XTCVR** : Current Value	0000h	R

13.3.3 - Block Diagram

Figure 53 : XTIMER Block Diagram



13.3.3.1 - Clocks

The XTCVR register clock is the prescaler output. The prescaler allows to divide the basic register frequency in order to offer a wide range of counting period, from 2**2 to 2**33 cycles (note that 1 cycle = 1 XCLK periods).

13.3.3.2 - 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.

– The selection between the sources is made through the XTCR control register.

When starting the timer, by setting TEN bit of TCR to '1', XTCVR will be loaded with XTSVR value on the first rising edge of the counting clock. That's to say that for counting from 0000h to 0009h for example, 10 counting clock rising edges are required.

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 actions are made depending on the XTCR control register content :

– The output XADCINJ is conditionally generated,

– XTCVR is loaded with XTSVR or stops or continues to count (see Table 22).

XTEVR, XTSVR and all TCR bits except TEN must not be modified while the timer is counting, ie while TEN bit of TCR = '1'. The timer behaviour is not guaranteed if this rule is not respected. It implies that the timer can be configured only when stopped (TEN = '0'). When programming the timer, XTEVR, XTSVR and XTCR bits except TEN can be modified, with TEN = '0'; then the timer is started by modifying only TEN bit of TCR. To stop the timer, only TEN bit should be modified, from '1' to '0'.

13.3.3.3 - Timer output (XADCINJ)

The XADCINJ output is the result of the (XTCVR = XTEVR) flag after differentiation. The duration of the output lasts two cycles (50ns at 40MHz).

Figure 54 : XADCINJ Timer Output

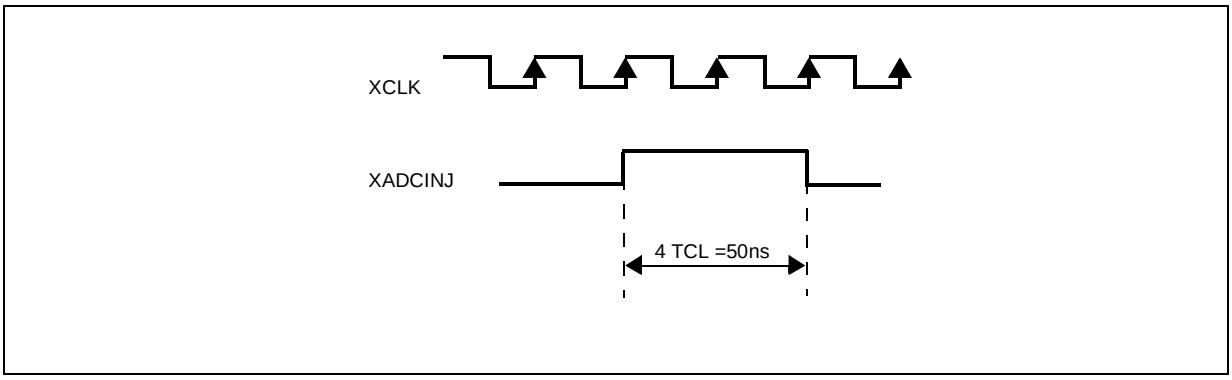
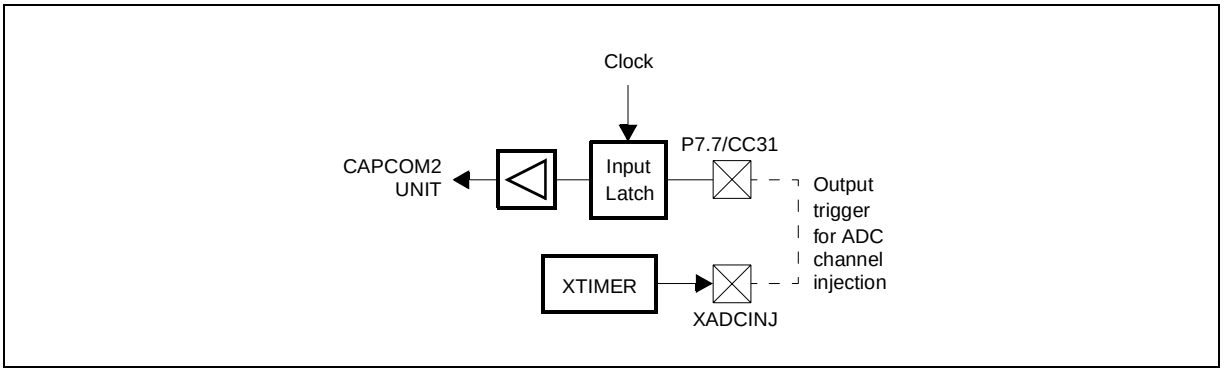


Figure 55 : External Connection for ADC Channel Injection



14 - SERIAL CHANNELS

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

14.1 - Asynchronous / Synchronous Serial Interface (ASCO)

The asynchronous / synchronous serial interface (ASCO) provides serial communication between the ST10F280 and other microcontrollers, microprocessors or external peripherals.

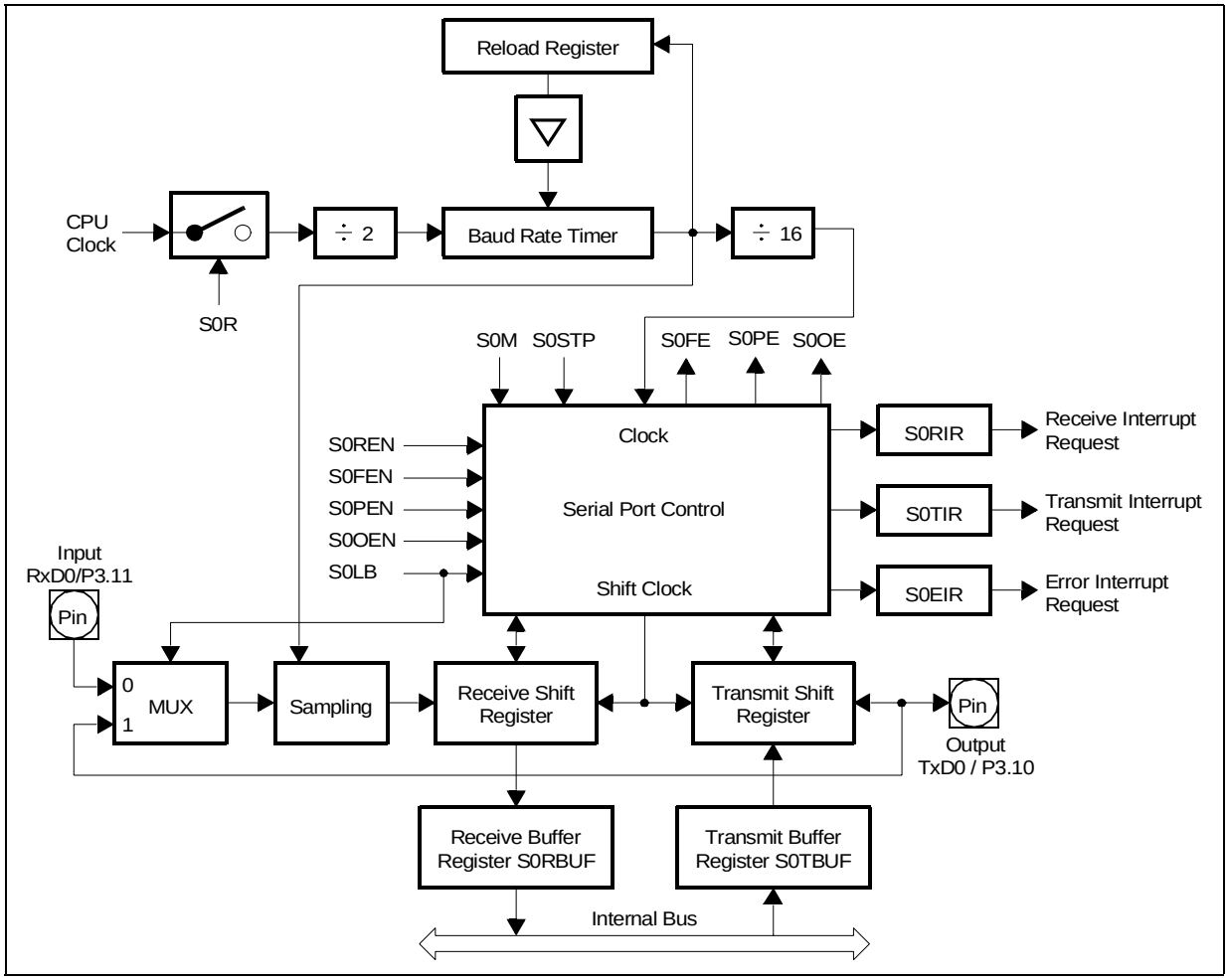
A set of registers is used to configure and to control the ASCO serial interface:

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

14.1.1 - ASCO in Asynchronous Mode

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

Figure 56 : Asynchronous Mode of Serial Channel ASC0



Asynchronous Mode Baud rates

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

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

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

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

Using the above equation, the maximum Baud rate can be calculated for any given clock speed. Baud rate versus reload register value (SOBRS=0 and SOBRS=1) is described in Table 24.

Table 24 : Commonly Used Baud Rates by Reload Value and Deviation Errors

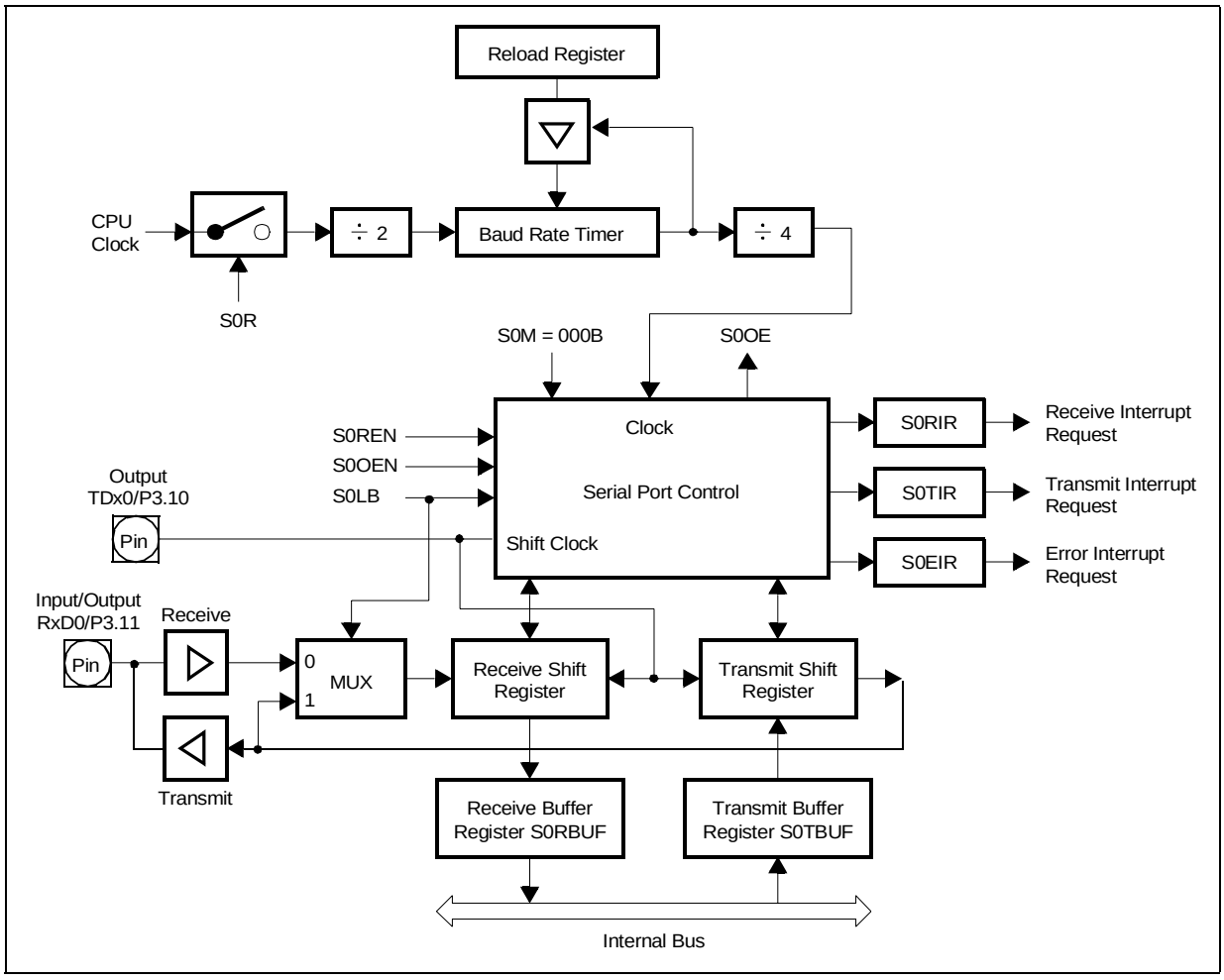
S0BRS = ‘0’, f_{CPU} = 40MHz			**S0BRS = ‘1’, f_{CPU} = 40MHz**		
Baud Rate (Baud)	**Deviation Error**	**Reload Value (hexa)**	**Baud Rate (Baud)**	**Deviation Error**	**Reload Value (hexa)**
1 250 000	0.0% / 0.0%	0000 / 0000	833 333	0.0% / 0.0%	0000 / 0000
112 000	+1.5% /7.0%	000A / 000B	112 000	+6.3%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

Note: The deviation errors given in the Table 24 are rounded. To avoid deviation errors use a Baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency).

14.1.2 - ASCO in Synchronous Mode

In synchronous mode, data are transmitted or received synchronously to a shift clock which is generated by the ST10F280. Half-duplex communication up to 5M Baud (at 40MHz of f_{CPU}) is possible in this mode.

Figure 57 : Synchronous Mode of Serial Channel ASC0



Synchronous Mode Baud Rates

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

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

(S0BRL) represents the content of the reload register, taken as unsigned 13 Bit integers,
(S0BRS) represents the value of Bit S0BRS ('0' or '1'), taken as integer.

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

Using the above equation, the maximum Baud rate can be calculated for any clock speed as given in Table 25.

Table 25 : Commonly Used Baud Rates by Reload Value and Deviation Errors

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

Note: The deviation errors given in the Table 25 are rounded. To avoid deviation errors use a Baud rate crystal (providing a multiple of the ASC0/SSC sampling frequency)

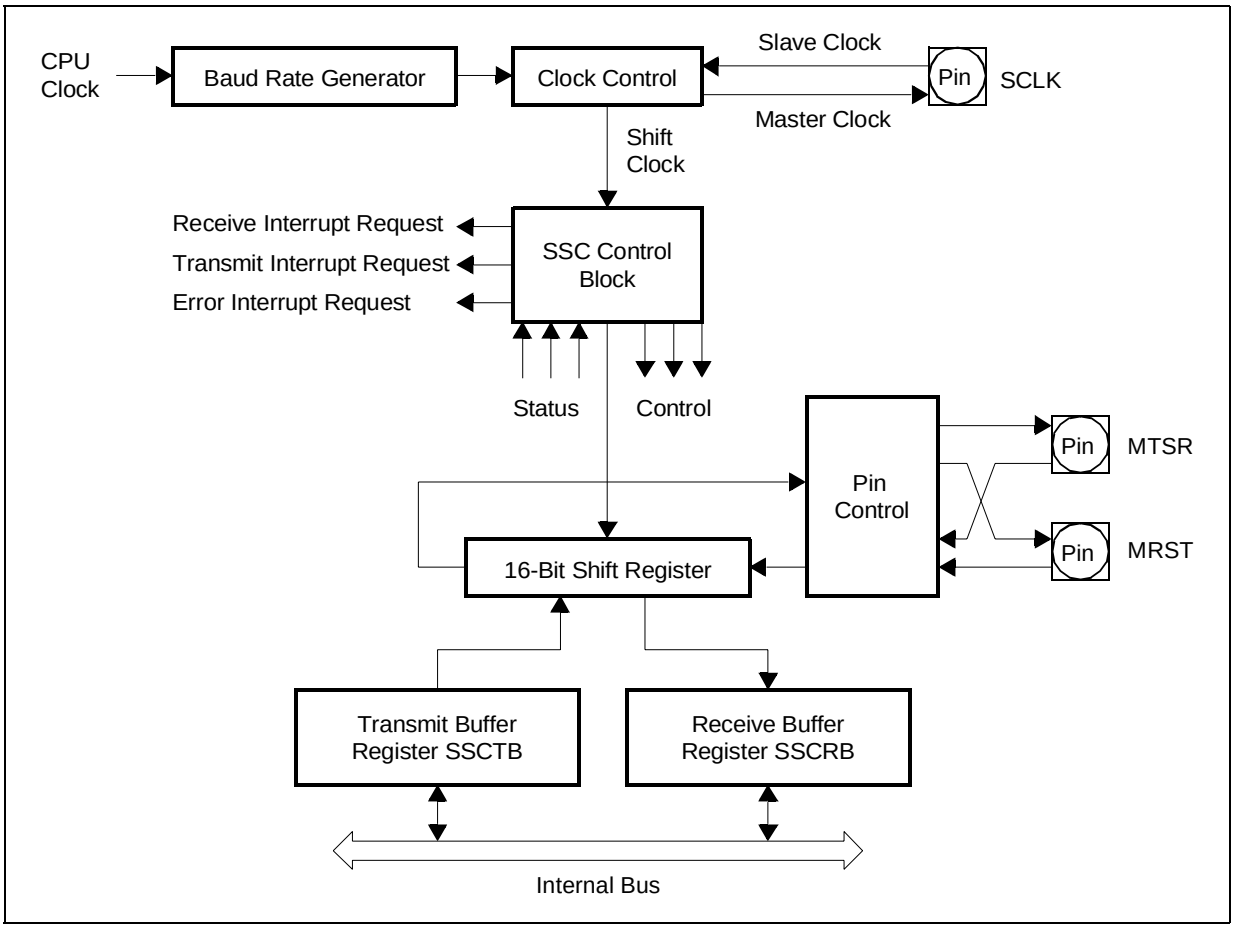
14.2 - High Speed Synchronous Serial Channel (SSC)

The High-Speed Synchronous Serial Interface SSC provides flexible high-speed serial communication between the ST10F280 and other microcontrollers, microprocessors or external peripherals.

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

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

Figure 58 : Synchronous Serial Channel SSC Block Diagram



Baud Rate Generation

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

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

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

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

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

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

Table 26 lists some possible Baud rates against the required reload values and the resulting bit times for a 40MHz CPU clock.

Table 26 : Synchronous Baud Rate and Reload Values

Baud Rate	Bit Time	Reload Value
Reserved use a reload value > 0.	---	---
10M Baud	100ns	0001h
5M Baud	200ns	0003h
2.5M Baud	400ns	0007h
1M Baud	1µs	0013h
100K Baud	10µs	00C7h
10K Baud	100µs	07CFh
1K Baud	1ms	4E1Fh
306 Baud	3.26ms	FF4Eh

15 - CAN MODULES

The two integrated CAN modules (CAN1 and CAN2) are identical and handle the completely autonomous transmission and reception of CAN frames in accordance with the CAN specification V2.0 part B (active) i.e. the on-chip CAN module can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

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

– The same internal register addresses both CAN controllers, but with the base addresses differing in address bit A8 and separate chip select for each CAN module. For address mapping, see Chapter 4.
– The CAN1 transmit line (CAN1_TxD) is the alternate function of the port P4.6 and the receive line (CAN1_RxD) is P4.5.
– The CAN2 transmit line (CAN2_TxD) is the alternate function of the port P4.7 and the receive line (CAN2_RxD) is the alternate function of the port P4.4.
– Interrupt of CAN2 is connected to the XBUS interrupt line XP1 (CAN1 is on XP0).
– Because of the new XPERCON register, both CAN modules have to be selected, before the bit XPEN is set in SYSCON register.
– After reset, the CAN1 is selected with the related control bit in the XPERCON register. The CAN2 is not selected.

15.1 - Memory Mapping

15.1.1 - CAN1

Address range 00'EF00h 00'EFFFh is reserved for the CAN1 Module access. The CAN1 is enabled by setting bit 0 of the new XPERCON register before setting XPEN bit 2 of the SYSCON register. Accesses to the CAN Module use demultiplexed addresses and a 16-bit data bus (byte accesses are possible). Two waitstates give an access time of 100 ns at 40MHz CPU clock. No tristate waitstate is used.

15.1.2 - CAN2

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

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

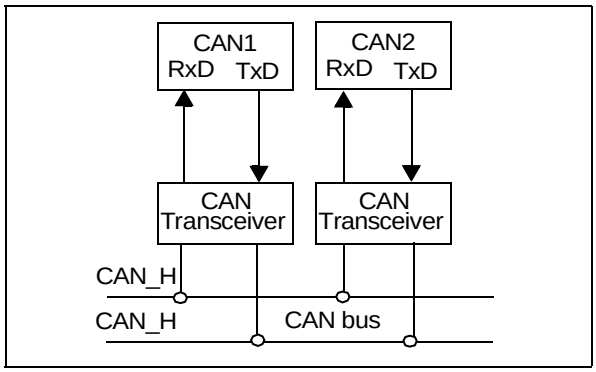
15.2 - CAN Bus Configurations

Depending on application, CAN bus configuration may be one single bus with a single or multiple interfaces or a multiple bus with a single or multiple interfaces. The ST10F280 is able to support these 2 cases.

Single CAN Bus

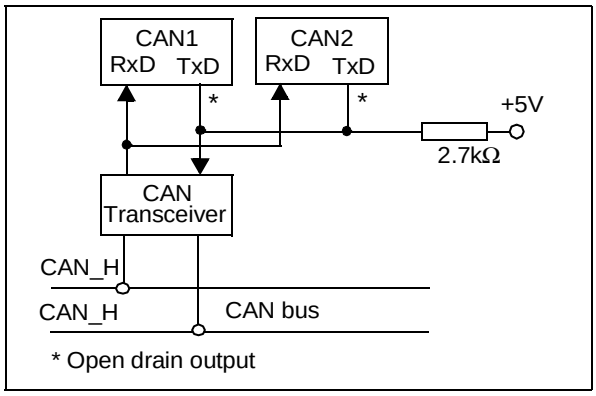
The single CAN Bus multiple interfaces configuration may be implemented using 2 CAN transceives as shown in Figure 59.

Figure 59 : Single CAN Bus Multiple Interfaces - Multiple Transceivers



The ST10F280 also supports single CAN Bus multiple (dual) interfaces using the open drain option of the CANx_TxD output as shown in Figure 60. Thanks to the OR-Wired Connection, only one transceiver is required. In this case the design of the application must take in account the wire length and the noise environment.

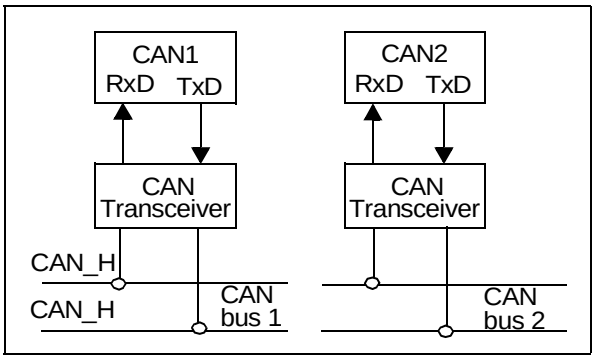
Figure 60 : Single CAN Bus Dual Interfaces - Single Transceiver



Multiple CAN Bus

The ST10F280 provides 2 CAN interfaces to support the kind of bus configuration shown in Figure 61.

Figure 61 : Connection to Two Different CAN Buses (e.g. for gateway application)

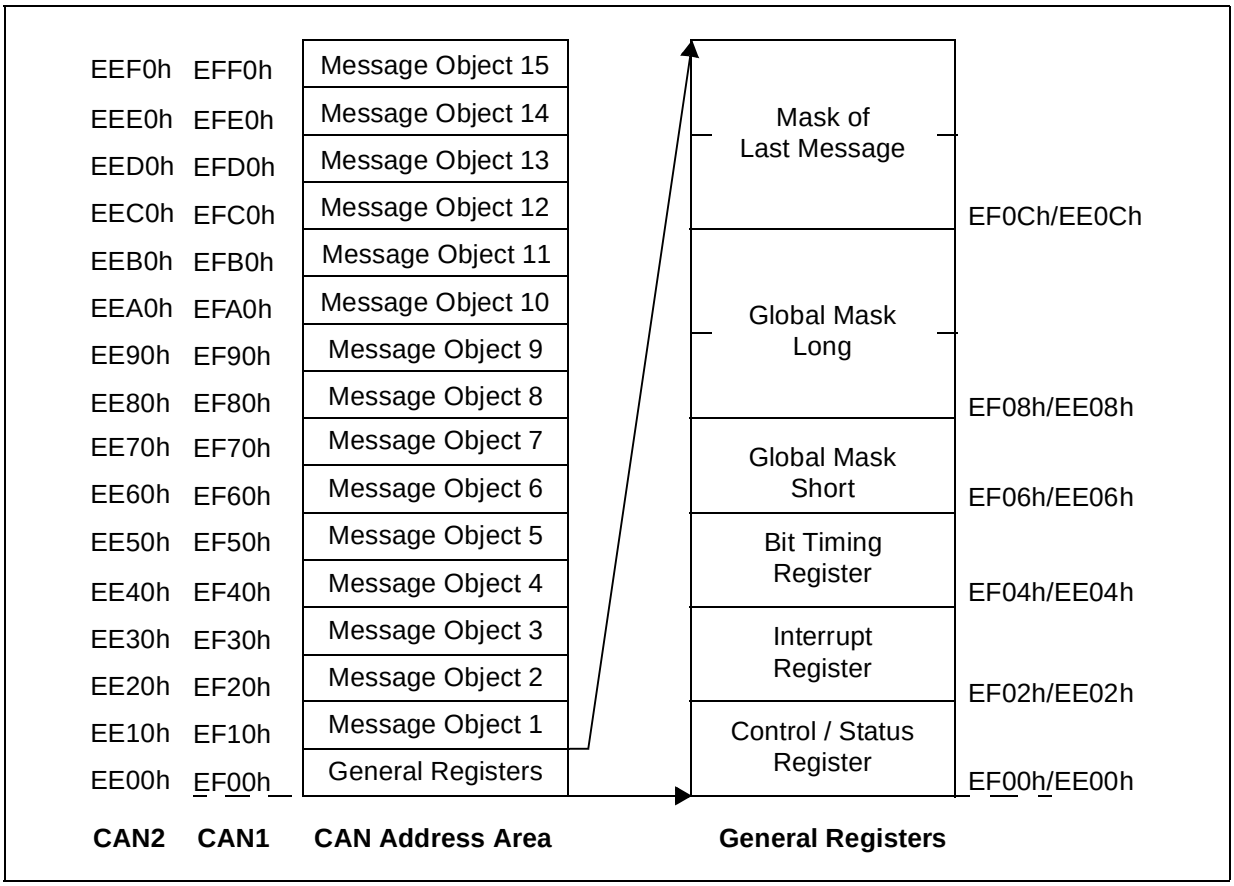


15.3 - Register and Message Object Organization

All registers and message objects of the CAN controller are located in the special CAN address area of 256 bytes, which is mapped into segment 0 and uses addresses 00'EE00h through 00'EFFFh. All registers are organized as 16 bit registers, located on word addresses. However, all registers may be accessed byte wise in order to select special actions without effecting other mechanisms.

Note The address map shown in Figure 62 lists the registers which are part of the CAN controller. There are also ST10F280 specific registers that are associated with the CAN Module. These registers, however, control the access to the CAN Module rather than its function.

Figure 62 : CAN Module Address Map



Control / Status Register (EF00h/EE00h) XReg Reset Value: XX01h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOFF	**EWRN**	-	**RXOK**	**TXOK**	**LEC**			TST	CCE	0	0	EIE	SIE	IE	**INIT**
R	R		RW	RW		RW		RW	RW	R	R	RW	RW	RW	RW

Table 27 : CAN Control/Status Register

Bit	Function (Control Bit)
INIT	**Initialization** 1: Software initialization of the CAN controller. While init is set, all message transfers are stopped. Setting init does not change the configuration registers and does not stop transmission or reception of a message in progress. The INIT bit is also set by hardware, following a busoff condition; the CPU then needs to reset INIT to start the bus recovery sequence. 0: Disable software initialization of the CAN controller; on INI completion, the CAN waits for 11 consecutive recessive bit before taking part in bus activities.
IE	**Interrupt Enable** Does not affect status updates. 1: Global interrupt enable from CAN module. 0: Global interrupt disable from CAN module.
SIE	**Status Change Interrupt Enable** 1: Enables interrupt generation when a message transfer (reception or transmision is successfully completed) or CAN bus error is detected and registered in LEC is the status partition. 0: Disable status change interrupt.
EIE	**Error Interrupt Enable** 1: Enables interrupt generation on a change of bit BOFF or EWARN in the status partition. 0: Disable error interrupt.
CCE	**Configuration Change Enable** 1: Allows CPU access to the bit timing register 0: Disables CPU access to the bit timing register
TST	**Test Mode (Bit 7)** Make sure that bit 7 is cleared when writing to the Control Register. Writing a 1 during normal operation may lead erroneous device behaviour.
LEC	**Last Error Code** This field holds a code which indicates the type of the last error occurred on the CAN bus. If a message has been transferred (reception or transmission) without error, this field will be cleared. Code “7” is unused and may be written by the CPU to check for updates. 0: **No Error** 1: **Stuff Error:** More than 5 equal bit in a sequence have occurred in a part of a received message where this is not allowed. 2: **Form Error:** A fixed format part of a received frame has the wrong format. 3: **AckError:** The message this CAN controller transmitted was not acknowledged by another node 4: **Bit1Error:** During the transmission of a message (with the exception of the arbitration field), the device wanted to send a *recessive* level (“1”), but the monitored bus value was *dominant* 5: **Bit0Error:** During the transmission of a message (or acknowledge bit, active error flag, or overload flag), the device wanted to send a *dominant* level (“0”), but the monitored bus value was *recessive*. During *busoff* recovery this status is set each time a sequence of 11 *recessive* bit has been monitored. This enables the CPU to monitor the proceeding of the busoff recovery sequence (indicating the bus is not stuck at *dominant* or continuously disturbed). 6: **CRCError:** The CRC check sum was incorrect in the message received.

Bit	Function (Control Bit)
TXOK	**Transmitted Message Successfully** Indicates that a message has been transmitted successfully (error free and acknowledged by at least one other node), since this bit was last reset by the CPU (the CAN controller does not reset this bit!).
RXOK	**Received Message Successfully** Indicates that a message has been received successfully, since this bit was last reset by the CPU (the CAN controller does not reset this bit!).
EWRN	**Error Warning Status** Indicates that at least one of the error counters in the EML has reached the error warning limit of 96.
BOFF	**Busoff Status** Indicates when the CAN controller is in busoff state (see EML).

Note Reading the upper half of the Control Register (status partition) will clear the Status Change Interrupt value in the Interrupt Register, if it is pending. Use byte accesses to the lower half to avoid this.

15.4 - CAN Interrupt Handling

The on-chip CAN Module has one interrupt output, which is connected (through a synchronization stage) to a standard interrupt node in the ST10F280 in the same manner as all other interrupts of the standard on-chip peripherals. The control register for this interrupt is XP0IC (located at address F186h/C3h for CAN1 and F18Eh/C7h for CAN2 in the ESFR range). The associated interrupt vector is called XP0INT at location 100h (trap number 40h) and XP1INT at location 104h (trap number 41h). With this configuration, the user has all control options available for this interrupt, such as enabling/ disabling, level and group priority, and interrupt or PEC service (see note below).

As for all other interrupts, the interrupt request flag XP0IR/XP1IR in register XP0IC/XP1IC is cleared automatically by hardware when this interrupt is serviced (either by standard interrupt or PEC service).

Note As a rule, CAN interrupt requests can be serviced by a PEC channel. However, because PEC channels only can execute single predefined data transfers (there are no conditional PEC transfers), PEC service can only be used, if the respective request is known to be generated by one specific source, and that no other interrupt request will be generated in between. In practice this seems to be a rare case.

Since an interrupt request of the CAN Module can be generated due to different conditions, the appropriate CAN interrupt status register must be read in the service routine to determine the cause of the interrupt request. The Interrupt Identifier INTID (a number) in the Interrupt Register indicates the cause of an interrupt. When no interrupt is pending, the identifier will have the value 00h. If the value in INTID is not 00h, then there is an interrupt pending. If bit IE in the Control Register is set, also the interrupt line to the CPU is activated. The interrupt line remains active until either INTID gets 00h (after the interrupt requester has been serviced) or until IE is reset (if interrupts are disabled).

The interrupt with the lowest number has the highest priority. If a higher priority interrupt (lower number) occurs before the current interrupt is processed, INTID is updated and the new interrupt overrides the last one. The Table 28 lists the valid values for INTID and their corresponding interrupt sources.

Interrupt Register (EF02h/EE02h) XReg Reset Value: - - XXh

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

Bit	Function
INTID	**Interrupt Identifier** This number indicates the cause of the interrupt. When no interrupt is pending, the value will be “00”.

Table 28 : INTID values and Corresponding Interrupt Sources

INTID	Cause of the Interrupt
00	**Interrupt Idle:** There is no interrupt request pending.
01	**Status Change Interrupt:** The CAN controller has updated (not necessarily changed) the status in the Control Register. This can refer to a change of the error status of the CAN controller (EIE is set and BOFF or EWRN change) or to a CAN transfer incident (SIE must be set), like reception or transmission of a message (RXOK or TXOK is set) or the occurrence of a CAN bus error (LEC is updated). The CPU may clear RXOK, TXOK, and LEC, however, writing to the status partition of the Control Register can never generate or reset an interrupt. To update the INTID value the status partition of the Control Register must be read.
02	**Message 15 Interrupt:** Bit INTPND in the Message Control Register of message object 15 (last message) has been set. The last message object has the highest interrupt priority of all message objects. [1)]
(2+N)	**Message N Interrupt:** Bit INTPND in the Message Control Register of message object ‘N’ has been set (N = 1...14). [1) 2)]

Notes 1) Bit INTPND of the corresponding message object has to be cleared to give messages with a lower priority the possibility to update INTID or to reset INTID to 00h (idle state).

2) A message interrupt code is only displayed, if there is no other interrupt request with a higher priority.

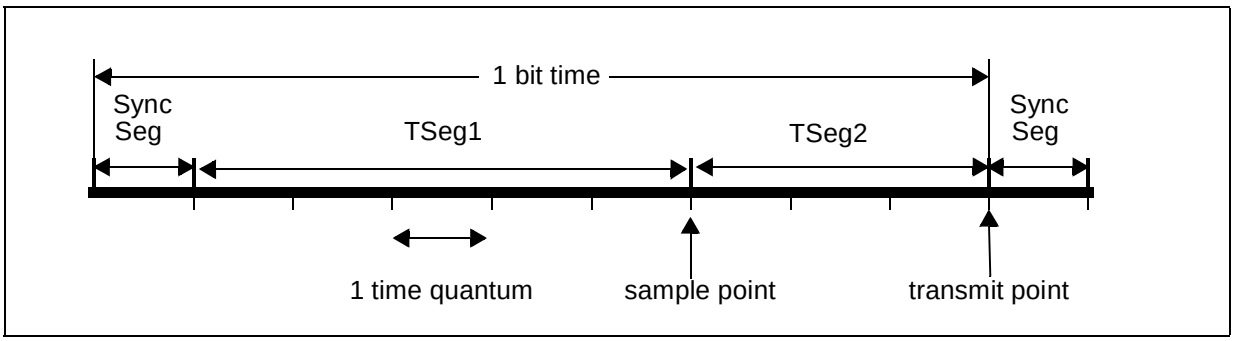
Bit Timing Configuration

According to the CAN protocol specification, a bit time is subdivided into four segments:
Sync segment, propagation time segment, phase buffer segment 1 and phase buffer segment 2.

Each segment is a multiple of the time quantum t_q with $t_q = (BRP + 1) \times 2 \times t_{XCLK}$
The Synchronization Segment (Sync seg) is always 1 t_q long. The Propagation Time Segment and the Phase Buffer Segment1 (combined to Tseg1) defines the time before the sample point, while Phase Buffer Segment2 (Tseg2) defines the time after the sample point. The length of these segments is programmable (except Sync-Seg).

Note For exact definition of these segments please refer to the CAN Specification.

Figure 63 : Bit Timing Definition



Bit Timing Register (EF04h/EE04h) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TSEG2			TSEG1				SJW		BRP					
R	RW			RW				RW		RW					

Bit	Function
BRP	**Baud Rate Prescaler** For generating the bit time quanta the CPU frequency is divided by 2 x (BRP+1).
SJW	**(Re)Synchronization Jump Width** Adjust the bit time by maximum (SJW+1) time quanta for re-synchronization.
TSEG1	**Time Segment before sample point** There are (TSEG1+1) time quanta before the sample point. Valid values for TSEG1 are “2...15”.
TSEG2	**Time Segment after sample point** There are (TSEG2+1) time quanta after the sample point. Valid values for TSEG2 are “1...7”.

Note This register can only be written, if the configuration change enable bit (CCE) is set.

Mask Registers

Messages can use standard or extended identifiers. Incoming frames are masked with their appropriate global masks. Bit IDE of the incoming message determines whether the standard 11 bit mask in Global Mask Short or the 29 bit extended mask in Global Mask Long is to be used. Bit holding a “0” mean “don’t care”, so do not compare the message’s identifier in the respective bit position.

The last message object (15) has an additional individually programmable acceptance mask (Mask of Last Message) for the complete arbitration field. This allows classes of messages to be received in this object by masking some bits of the identifier.

Note The Mask of Last Message is ANDed with the Global Mask that corresponds to the incoming message.

Global Mask Short (EF06h/EE06h) XReg Reset Value: UFUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID20...18			1	1	1	1	1	ID28...21							
RW			R	R	R	R	R	RW							

Bit	Function
ID28...18	**Identifier (11 Bit)** Mask to filter incoming messages with standard identifier.

Upper Global Mask Long (EF08h/EE08h) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID20...13								ID28...21							
RW								RW							

Lower Global Mask Long (EF0Ah/EE0Ah) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID4...0					0	0	0	ID12...5							
RW					R	R	R	RW							

Bit	Function
ID28...0	**Identifier (29 bit)** Mask to filter incoming messages with extended identifier.

Upper Mask of Last Message (EF0Ch/EE0Ch) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID20...18			ID17...13					ID28...21							
RW			RW					RW							

Lower Mask of Last Message (EF0Eh/EE0Eh) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID4...0					0	0	0	ID12...5							
RW					R	R	R	RW							

Bit	Function
ID28...0	**Identifier (29 bit)** Mask to filter the last incoming message (Nr. 15) with standard or extended identifier (as configured).

15.5 - The Message Object

The message object is the primary means of communication between CPU and CAN controller. Each of the 15 message objects uses 15 consecutive bytes (see Figure 64) and starts at an address that is a multiple of 16.

Note All message objects must be initialized by the CPU, even those which are not going to be used, before clearing the INIT bit.

Each element of the Message Control Register is made of two complementary bits.

This special mechanism allows the selective setting or resetting of specific elements (leaving others unchanged) without requiring read-modify-write cycles. None of these elements will be affected by reset.

The Table 29 shows how to use and to interpret these 2 bit-fields.

Figure 64 : Message Object Address Map

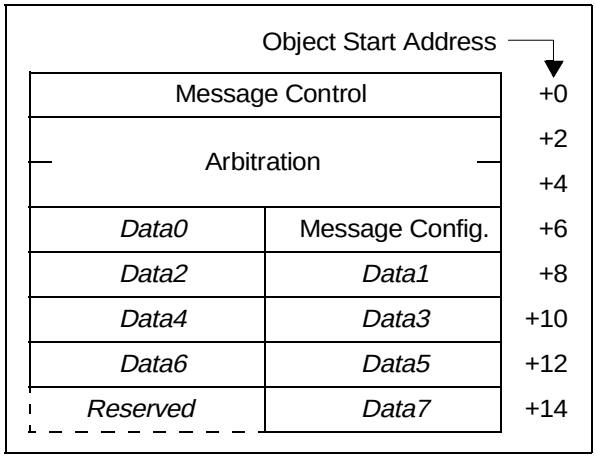


Table 29 : Functions of Complementary Bit of Message Control Register

Value	Function on Write	Meaning on Read
00	Reserved	Reserved
01	Reset element	Element is reset
10	Set element	Element is set
11	Leave element unchanged	Reserved

Message Control Register (EFn0h/EEn0h) XReg Reset Value: UUUUh

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RMTPND	**TXRQ**	**MSGLST CPUUPD**	**NEWDAT**	**MSGVAL**	TXIE	RXIE	**INTPND**
RW	RW	RW	RW	RW	RW	RW	RW

Bit	Function
INTPND	**Interrupt Pending** Indicates, if this message object has generated an interrupt request (see TXIE and RXIE), since this bit was last reset by the CPU, or not.
RXIE	**Receive Interrupt Enable** Defines, if bit INTPND is set after successful reception of a frame.
TXIE	**Transmit Interrupt Enable** Defines, if bit INTPND is set after successful transmission of a frame. [1]
MSGVAL	**Message Valid** Indicates, if the corresponding message object is valid or not. The CAN controller only operates on valid objects. Message objects can be tagged invalid, while they are changed, or if they are not used at all.
NEWDAT	**New Data** Indicates, if new data has been written into the data portion of this message object by CPU (transmit-objects) or CAN controller (receive-objects) since this bit was last reset, or not. [2]
MSGLST (Receive) CPUUPD (Transmit)	**Message Lost** (This bit applies to *receive*-objects only) Indicates that the CAN controller has stored a new message into this object, while NEWDAT was still set, i.e. the previously stored message is lost. **CPU Update** (This bit applies to *transmit*-objects only) Indicates that the corresponding message object may not be transmitted now. The CPU sets this bit in order to inhibit the transmission of a message that is currently updated, or to control the automatic response to remote requests.
TXRQ	**Transmit Request** Indicates that the transmission of this message object is requested by the CPU or via a remote frame and is not yet done. TXRQ can be disabled by CPUUPD. [1] [3]
RMTPND	**Remote Pending** (Used for transmit-objects) Indicates that the transmission of this message object has been requested by a remote node, but the data has not yet been transmitted. When RMTPND is set, the CAN controller also sets TXRQ. RMTPND and TXRQ are cleared, when the message object has been successfully transmitted.

Notes 1. In message object 15 (last message) these bits are hardwired to "0" (inactive) in order to prevent transmission of message 15.

2. When the CAN controller writes new data into the message object, unused message bytes will be overwritten by non specified values. Usually the CPU will clear this bit before working on the data, and verify that the bit is still cleared once it has finished working to ensure that it has worked on a consistent set of data and not part of an old message and part of the new message.
For transmit-objects the CPU will set this bit along with clearing bit CPUUPD. This will ensure that, if the message is actually being transmitted during the time the message was being updated by the CPU, the CAN controller will not reset bit TXRQ. In this way bit TXRQ is only reset once the actual data has been transferred.

3. When the CPU requests the transmission of a receive-object, a remote frame will be sent instead of a data frame to request a remote node to send the corresponding data frame. This bit will be cleared by the CAN controller along with bit RMTPND when the message has been successfully transmitted, if bit NEWDAT has not been set. If there are several valid message objects with pending transmission request, the message with the lowest message number is transmitted first.

15.6 - Arbitration Registers

The arbitration Registers are used for acceptance filtering of incoming messages and to define the identifier of outgoing messages.

Upper Arbitration Reg (EFn2h/EEn2h) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID20...18			ID17...13					ID28...21							
RW			RW					RW							

Lower Arbitration Reg (EFn4h/EEn4h) XReg Reset Value: UUUUh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID4...0					0	0	0	ID12...5							
RW					R	R	R	RW							

Bit	Function
ID28...0	**Identifier (29 bit)** Identifier of a standard message (ID28...18) or an extended message (ID28...0). For standard identifiers bit ID17...0 are “don’t care”.

16 - WATCHDOG TIMER

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

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

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

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

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

WDTCON (FFAEh / D7h) SFR Reset Value: 00xxh

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

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

Notes: 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 Vcc = 0 V to Vcc > 2.0 V is recognized.

The PONR flag of WDTCON register is set if the output voltage of the internal 3.3V supply falls below the threshold (typically 2V) of the power-on detection circuit. This circuit is efficient to detect major failures of the external 5V supply but if the internal 3.3V supply does not drop under 2 volts, the PONR flag is not set. This could be the case on fast switch-off / switch-on of the 5V 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 30 : WDTCON Bits Value on Different Resets

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

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

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

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

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

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

The Table 31 shows the watchdog time range for 40MHz CPU clock.

Table 31 : WDTREL Reload Value

Reload value in WDTREL	Prescaler for f_{CPU} = 40MHz	
	2 (WDTIN = '0')	128 (WDTIN = '1')
FFh	12.8µs	819.2ms
00h	3.276ms	209.7ms

The watchdog timer period is calculated with the following formula:

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

17 - SYSTEM RESET

Table 32 : Reset Event Definition

Reset Source	Short-cut	Conditions
Power-on reset	PONR	Power-on
Long Hardware reset (synchronous & asynchronous)	LHWR	t_{RSTIN} > 1032 TCL
Short Hardware reset (synchronous reset)	SHWR	4 TCL < t_{RSTIN} ≤ 1032 TCL
Watchdog Timer reset	WDTR	WDT overflow
Software reset	SWR	SRST execution

System reset initializes the MCU in a predefined state. There are five ways to activate a reset state. The system start-up configuration is different for each case as shown in Table 32.

17.1 - Asynchronous Reset (Long Hardware Reset)

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

Power-on reset

The asynchronous reset must be used during the power-on of the MCU. Depending on crystal frequency, the on-chip oscillator needs about 10ms to 50ms to stabilize. The logic of the MCU does not need a stabilized clock signal to detect an asynchronous reset, so it is suitable for power-on conditions. To ensure a proper reset sequence, the $\overline{RSTIN}$ pin and the RPD pin must be held at low level until the MCU clock signal is stabilized and the system configuration value on PORT0 is settled.

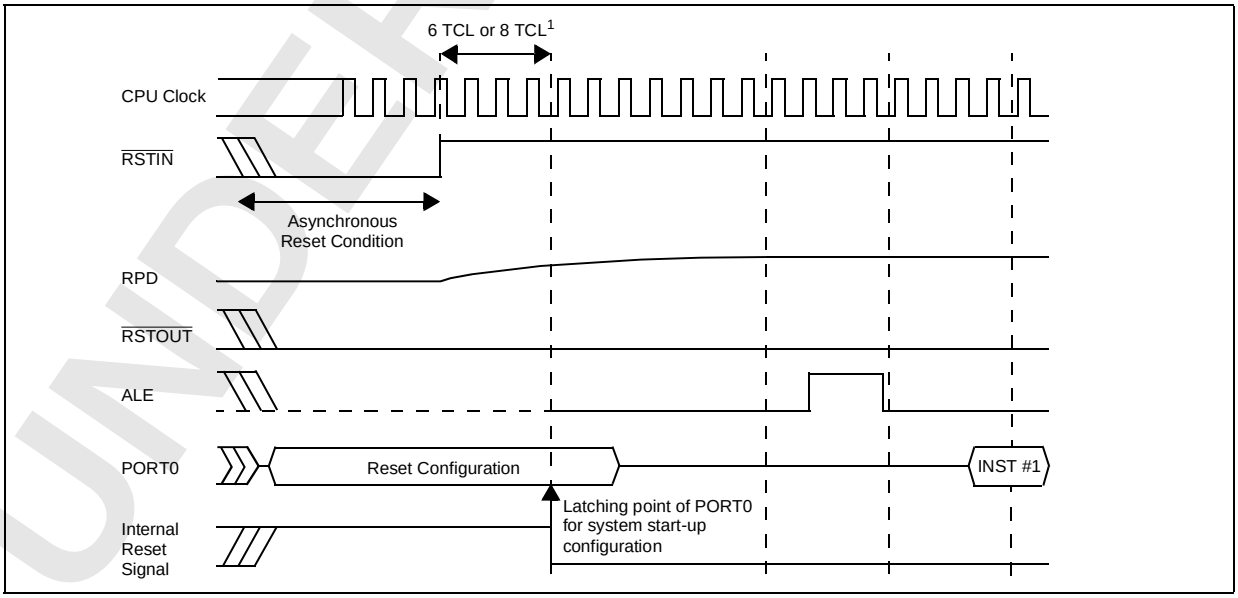
Hardware reset

The asynchronous reset must be used to recover from catastrophic situations of the application. It may be triggerred by the hardware of the application. Internal hardware logic and application circuitry are described in Reset circuitry chapter and Figures Figure 68 :, Figure 69 : and Figure 70 :.

Exit of asynchronous reset state

When the $\overline{RSTIN}$ pin is pulled high, the MCU restarts. The system configuration is latched from PORT0 and ALE, $\overline{RD}$ and $\overline{R/W}$ pins are driven to their inactive level. The MCU starts program execution from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. Timing of asynchronous reset sequence are summarized in Figure 65.

Figure 65 : Asynchronous Reset Timing



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

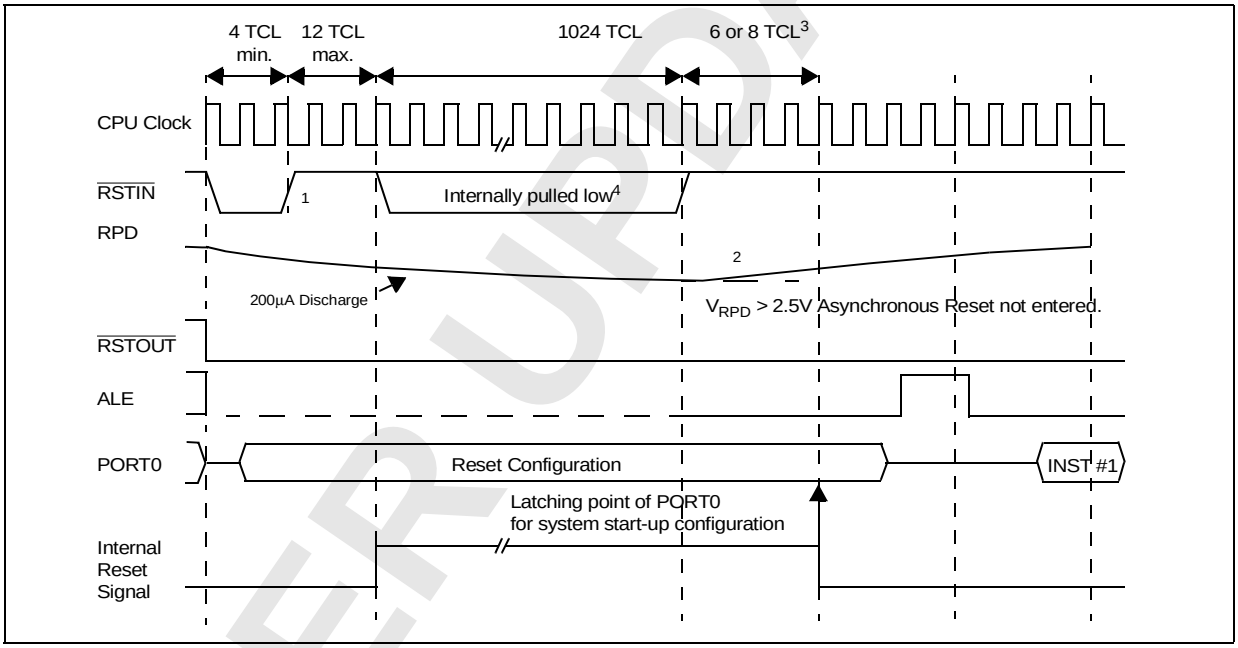
17.2 - Synchronous Reset (Warm Reset)

A synchronous reset is triggered when $\overline{\text{RSTIN}}$ pin is pulled low while RPD pin is at high level. In order to properly activate the internal reset logic of the MCU, the $\overline{\text{RSTIN}}$ pin must be held low, at least, during 4 TCL (2 periods of CPU clock). The I/O pins are set to high impedance and $\overline{\text{RSTOUT}}$ pin is driven low. After $\overline{\text{RSTIN}}$ level is detected, a short duration of 12 TCL (approximately 6 periods of CPU clock) elapes, during which pending internal hold states are cancelled and the current internal access cycle if any is completed. External bus cycle is aborted. The internal pull-down of $\overline{\text{RSTIN}}$ pin is activated if bit BDRSTEN of SYSCON register was previously set by software. This bit is always cleared on power-on or after a reset sequence.

Exit of synchronous reset state

The internal reset sequence starts for 1024 TCL (512 periods of CPU clock) and $\overline{\text{RSTIN}}$ pin level is sampled. The reset sequence is extended until $\overline{\text{RSTIN}}$ level becomes high. Then, the MCU restarts. The system configuration is latched from PORT0 and ALE, $\overline{\text{RD}}$ and $\overline{\text{R/W}}$ pins are driven to their inactive level. The MCU starts program execution from memory location 00'0000h in code segment 0. This starting location will typically point to the general initialization routine. Timing of synchronous reset sequence are summarized in Figure 66 and Figure 67.

Figure 66 : Synchronous Warm Reset (Short low pulse on $\overline{\text{RSTIN}}$)



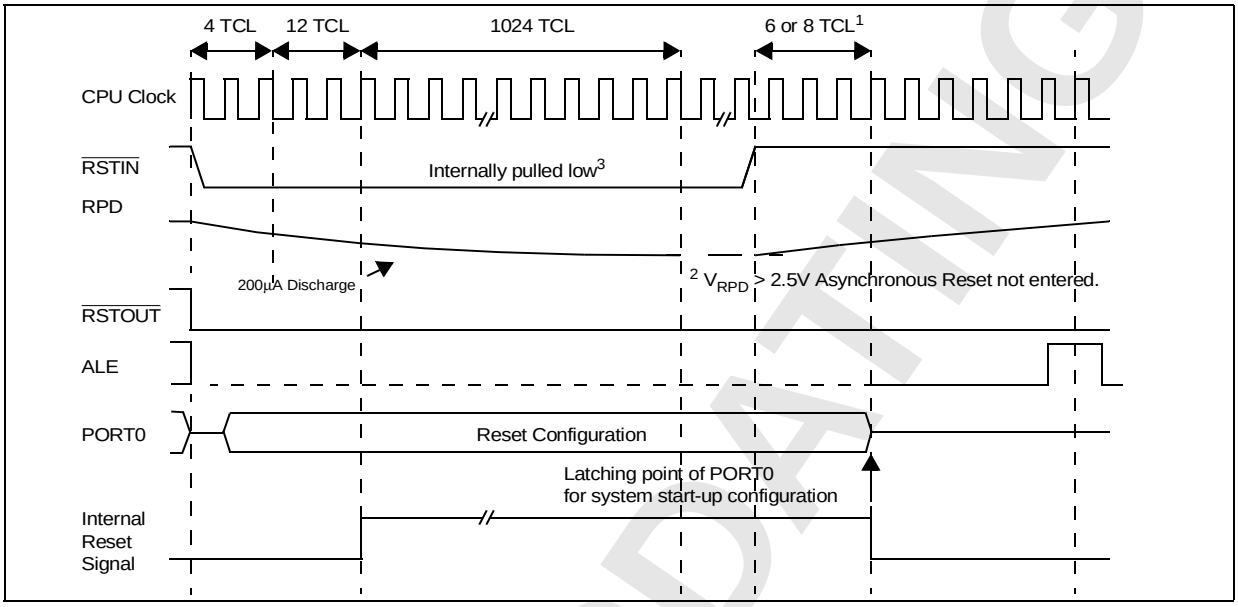
Notes: 1. $\overline{\text{RSTIN}}$ assertion can be released there.

2. If during the reset condition ($\overline{\text{RSTIN}}$ low), V_{RPD} voltage drops below the threshold voltage (about 2.5V for 5V operation), the asynchronous reset is then immediately entered.

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

4) $\overline{\text{RSTIN}}$ pin is pulled low if bit BDRSTEN (bit 5 of SYSCON register) was previously set by software. Bit BDRSTEN is cleared after reset.

Figure 67 : Synchronous Warm Reset (Long low pulse on $\overline{\text{RSTIN}}$)



Notes: 1. $\overline{\text{RSTIN}}$ rising edge to internal latch of PORT0 is 3 CPU (6 TCL) clock cycles if the PLL is bypassed and the prescaler is on ($f_{CPU} = f_{XTAL} / 2$), else it is 4 CPU clock cycles (8 TCL).
2. If during the reset condition ($\overline{\text{RSTIN}}$ low), V_{RPD} voltage drops below the threshold voltage (about 2.5V for 5V operation), the asynchronous reset is then immediately entered.
3. $\overline{\text{RSTIN}}$ pin is pulled low if bit BDRSTEN (bit 5 of SYSCON register) was previously set by soft-ware. Bit BDRSTEN is cleared after reset.

17.3 - Software Reset

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

Upon execution of the SRST instruction, the internal reset sequence (1024 TCL) is started. The microcontroller behaviour is the same as for a Short Hardware reset, except that only P0.12...P0.6 bit are latched at the end of the reset sequence, while P0.5...P0.2 bit are cleared.

17.4 - Watchdog Timer Reset

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

Unlike hardware and software resets, the watchdog reset completes a running external bus cycle if this bus cycle either does not use $\overline{\text{READY}}$, 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. At the end of the internal reset sequence (1024 TCL), only P0.12...P0.6 bit are latched, while previously latched values of P0.5...P0.2 are cleared.

17.5 - $\overline{\text{RSTOUT}}$ Pin and 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). $\overline{\text{RSTOUT}}$ pin stays active low beyond the end of the initialization routine, until the protected EINIT instruction (End of Initialization) is completed.

The Bidirectional Reset function is useful when external devices require a reset signal but cannot be connected to $\overline{\text{RSTOUT}}$ pin, because $\overline{\text{RSTOUT}}$ signal lasts during initialisation. It is, for instance, the case of external memory running initialization routine before the execution of EINIT instruction.

Bidirectional reset function is enabled by setting bit 3 (BDRSTEN) in SYSCON register. It only can be enabled during the initialization routine, before EINIT instruction is completed.

When enabled, the open drain of the RSTIN pin is activated, pulling down the reset signal, for the duration of the internal reset sequence (synchronous/asynchronous hardware, software and watchdog timer resets). At the end of the internal reset sequence the pull down is released and the RSTIN pin is sampled 8 TCL periods later.

– If signal is sampled low, a hardware reset is triggered again.
– If it is sampled high, the chip exits reset state according to the running reset way (synchronous/ asynchronous hardware, software and watchdog timer resets).

Note: The bidirectional reset function is disabled by any reset sequence (Bit BDRSTEN of SYSCON is cleared). To be activated again it must be enabled during the initialization routine.

17.6 - Reset Circuitry

The internal reset circuitry is described in Figure 68.

An internal pull-up resistor is implemented on RSTIN pin. (50kΩ minimum, to 250kΩ maximum). The minimum reset time must be calculated using the lowest value. In addition, a programmable pull-down (bit BDRSTEN of SYSCON register) drives the RSTIN pin according to the internal reset state as explained in Section 17.5 - RSTOUT Pin and Bidirectional Reset.

The RSTOUT pin provides a signals to the application as described in Section 17.5 - RSTOUT Pin and Bidirectional Reset.

A weak internal pull-down is connected to the RPD pin to discharge external capacitor to Vss at a rate of 100µA to 200µA. This Pull-down is turned on when RSTIN pin is low

If bit PWDCFG of SYSCON register is set, an internal pull-up resistor is activated at the end of the reset sequence. This pull-up charges the capacitor connected to RPD pin.

If Bidirectional Reset function is not used, the simplest way to reset ST10F280 is to connect external components as shown in Figure 69. It works with reset from application (hardware or manual) and with power-on. The value of C1 capacitor, connected on RSTIN pin with internal pull-up resistor (50kΩ to 250kΩ), must lead to a charging time long enough to let the internal or external oscillator and / or the on-chip PLL to stabilize.

The R0-C0 components on RPD pin are mainly implemented to provide a time delay to exit Power down mode (see Chapter 18 - Power Reduction Modes). Nervertheless, they drive RPD pin level during resets and they lead to different reset modes as explained hereafter. On power-on, C0 is totaly discharged, a low level on RPD pin forces an asynchronous hardware reset. C0 capacitor starts to charge throught R0 and at the end of reset sequence ST10F280 restarts. RPD pin threshold is typically 2.5V.

Depending on the delay of the next applied reset, the MCU can enter a synchronous reset or an asynchronous reset. If RPD pin is below 2.5V an asynchronous reset starts, if RPD pin is above 2.5V a synchronous reset starts. (see Section 17.1 - Asynchronous Reset (Long Hardware Reset) and Section 17.2 - Synchronous Reset (Warm Reset)).

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

In order to properly use the Bidirectional reset features, the schematic (or equivalent) of Figure 70 must be implemented. R1-C1 only work for power-on or manual reset in the same way as explained previously. D1 diode brings a faster discharge of C1 capacitor at power-off during repetitive switch-on / switch-off sequences. D2 diode performs an OR-wired connection, it can be replaced with an open drain buffer. R2 resistor may be added to increase the pull-up current to the open drain in order to get a faster rise time on RSTIN pin when bidirectional function is activated.

The start-up configurations and some system features are selected on reset sequences as described in Table 33 and Table 34.

Table 33 describes what is the system configuration latched on PORT0 in the five different reset ways. Table 34 summarizes the bit state of PORT0 latched in RP0H, SYSCON, BUSCON0 registers. RPOH register is described in Section 19.2 - System Configuration Registers.

Figure 68 : Internal (simplified) Reset Circuitry.

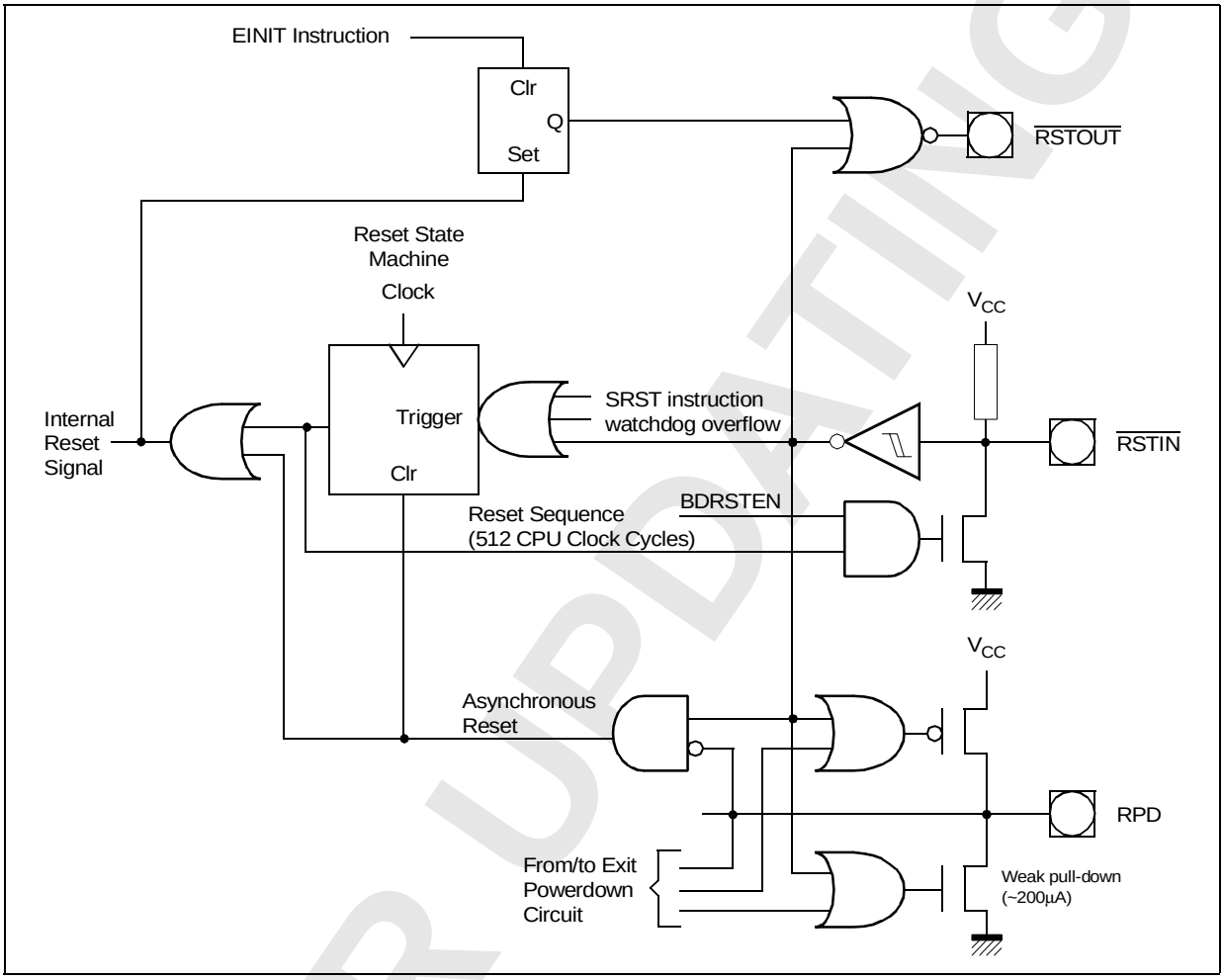


Figure 69 : Minimum External Reset Circuitry

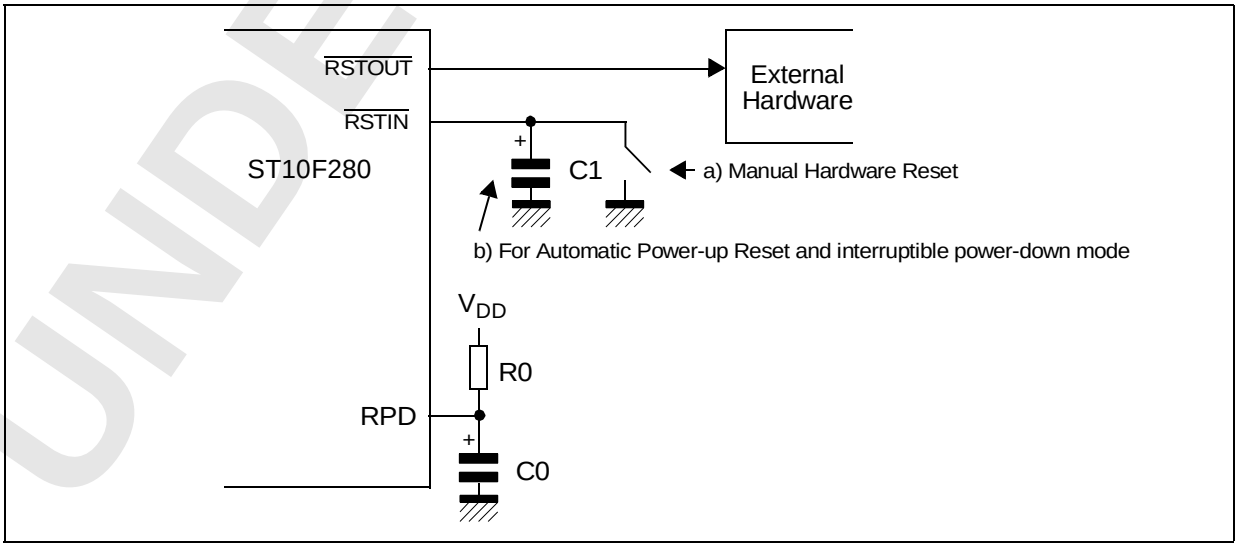


Figure 70 : External Reset Hardware Circuitry

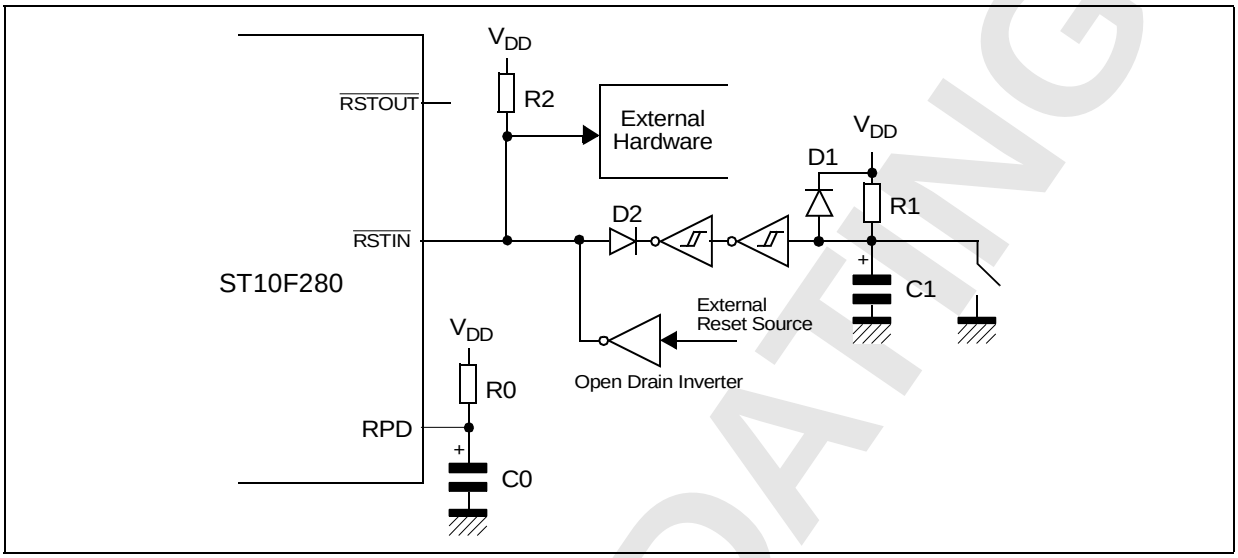


Table 33 : PORT0 Latched Configuration for the Different Resets

X : Pin is sampled - : Pin is not sampled	PORT0															
	Clock Options			Segm. Addr. Lines		Chip Selects		WR config.	Bus Type		Reserved	BSL	Reserved	Reserved	Adapt Mode	Emu Mode
Sample event	**P0H.7**	**P0H.6**	**P0H.5**	**P0H.4**	**P0H.3**	**P0H.2**	**P0H.1**	**P0H.0**	**P0L.7**	**P0L.6**	**P0L.5**	**P0L.4**	**P0L.3**	**P0L.2**	**P0L.1**	**P0L.0**
Software Reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Watchdog Reset	-	-	-	X	X	X	X	X	X	X	-	-	-	-	-	-
Short Hardware Reset	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X
Long Hardware Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Power-On Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 34 : PORT0 bit latched into the different registers after reset

PORT0 bit nber	**h7**	**h6**	**h5**	**h4**	**h3**	**h2**	**h1**	**h0**	**l7**	**l6**	**l5**	**l4**	**l3**	**l2**	**l1**	**l0**
PORT0 bit Name	CLKCFG	CLKCFG	CLKCFG	SALSEL	SALSEL	CSSEL	CSSEL	WRC	BUSTYP	BUSTYP	R	BSL	R	R	ADP	EMU
RP0H [2]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	CLKCFG	CLKCFG	CLKCFG	SALSEL	SALSEL	CSSEL	CSSEL	WRC
SYSCON	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	BYTDIS [3]	X [1]	WRCFG [3]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]
BUSCON0	X [1]	X [1]	X [1]	X [1]	-	BUS ACT0 [4]	ALE CTL0 [4]	-	BTYP	BTYP	X [1]	X [1]	X [1]	X [1]	X [1]	X [1]
Internal Logic	To Clock Generator			To Port 4 Logic		To Port 6 Logic		X [1]	X [1]	X [1]	X [1]	Internal	X [1]	X [1]	Internal	Internal

Notes: 1. Not latched from PORT0.

2. Only RP0H low byte is used and the bit-fields are latched from PORT0 high byte to RP0H low byte.

3. Indirectly depend on PORT0.

4. Bits set if $\overline{EA}$ pin is 1.

18 - POWER REDUCTION MODES

Two different power reduction modes with different levels of power reduction have been implemented in the ST10F280, which may be entered under software control.

In **Idle mode** the CPU is stopped, while the peripherals continue their operation. Idle mode can be terminated by any reset or interrupt request.

In **Power Down mode** both the CPU and the peripherals are stopped. Power Down mode can now be configured by software in order to be terminated only by a hardware reset or by a transition on enabled fast external interrupt pins.

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

18.1 - Idle Mode

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

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

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

18.2 - Power Down Mode

Power Down mode starts by running PWRDN protected instruction. Internal clock is stopped, all MCU parts are on hold including the watchdog timer.

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

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

SYSCON (FF12h / 89h) SFR Reset Value: 0xx0h

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

Bit	Function
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 occurs by asserting the $\overline{RSTIN}$ pin. 1 Power Down Mode can only be entered during PWRDN instruction execution if all enabled FastExternal Interrupt (EXxIN) pins are in their inactive level. Exiting this mode can be done by asserting one enabled EXxIN pin.

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

18.2.1 - Protected Power Down Mode

This mode is selected by clearing the bit PWDCFG in register SYSCON to '0'.

In this mode, the Power Down mode can **only** be entered if the NMI (Non Maskable Interrupt) pin is externally pulled low while the PWRDN instruction is executed.

This feature can be used in conjunction with an external power failure signal which pulls the NMI pin low when a power failure is imminent. The microcontroller will enter the NMI trap routine which can save the internal state into RAM. After the internal state has been saved, the trap routine may set a flag or write a certain bit pattern into specific RAM locations, and then execute the PWRDN instruction. If the NMI pin is still low at this time, Power Down mode will be entered, otherwise program execution continues. During power down the voltage delivered by the on-chip voltage regulator automatically lowers the internal logic supply down to 2.5 V, saving the power while the contents of the internal RAM and all registers will still be preserved.

Exiting Power Down Mode

In this mode, the **only** way to exit Power Down mode is with an external hardware reset.

The initialization routine (executed upon reset) can check the identification flag or bit pattern within RAM to determine whether the controller was initially switched on, or whether it was properly restarted from Power Down mode.

18.2.2 - Interruptable Power Down Mode

This mode is selected by setting the bit bit PWDCFG in register SYSCON to '1'.

In this mode, the Power Down mode can be entered if enabled Fast External Interrupt pins (EXxIN pins, alternate functions of Port 2 pins, with x = 7...0) are in their inactive level. This inactive level is configured with the EXIxES bit field in the EXICON register, as follow:

EXICON (F1C0h / E0h) ESFR Reset Value: 0000h

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

Bit	Function
EXIxES (x=7...0)	**External Interrupt x Edge Selection Field (x=7...0)** 00 Fast external interrupts disabled: standard mode EXxIN pin not taken in account for entering/exiting Power Down mode. 01 Interrupt on positive edge (rising) Enter Power Down mode if EXiIN = '0', exit if EXxIN = '1' (referred as 'high' active level) 10 Interrupt on negative edge (falling) Enter Power Down mode if EXiIN = '1', exit if EXxIN = '0' (referred as 'low' active level) 11 Interrupt on any edge (rising or falling) Always enter Power Down mode, exit if EXxIN level changed.

Exiting Power Down Mode

When Power Down mode is entered, the CPU and peripheral clocks are frozen, and the oscillator and PLL are stopped. Power Down mode can be exited by either asserting $\overline{\text{RSTIN}}$ or one of the enabled EXxIN pin (Fast External Interrupt).

$\overline{\text{RSTIN}}$ must be held low until the oscillator and PLL have stabilized.

EXxIN inputs are normally sampled interrupt inputs. However, the Power Down mode circuitry uses them as level-sensitive inputs. An EXxIN (x = 7...0) Interrupt Enable bit (bit CCxIE in respective CCxIC register) need not to be set to bring the device out of Power Down mode. An external RC circuit must be connected, as shown in the following figure:

Figure 71 : External RC Circuit on RPD Pin for Exiting Powerdown Mode with External Interrupt

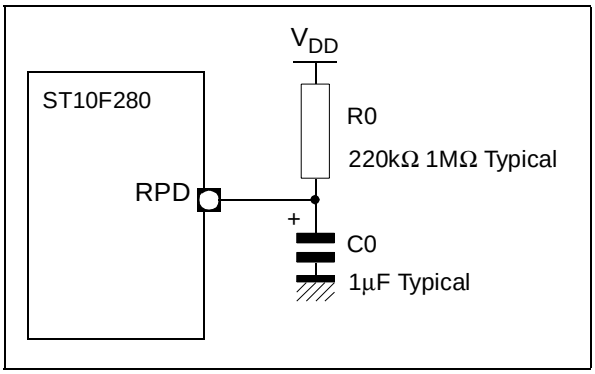


Figure 72 : Simplified Powerdown Exit Circuitry

To exit Power Down mode with external interrupt, an EXxIN pin has to be asserted for at least 40 ns (x = 7...0). This signal enables the internal oscillator and PLL circuitry, and also turns on the weak pull-down (see following figure). The discharging 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 Vpp voltage drops below the threshold voltage (about 2.5 V), the Schmitt trigger clears Q2 flip-flop, thus enabling the CPU and Peripheral clocks, and the device resumes code execution.

If the Interrupt was enabled (bit CCxIE='1' in the respective CCxIC register) before entering Power Down mode, the device executes the interrupt service routine, and then resumes execution after the PWRDN intruction (see note below). If the interrupt was disabled, the device executes the instruction following PWRDN instruction, and the Interrupt Request Flag (bit CCxIR in the respective CCxIC register) remains set until it is cleared by software.

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

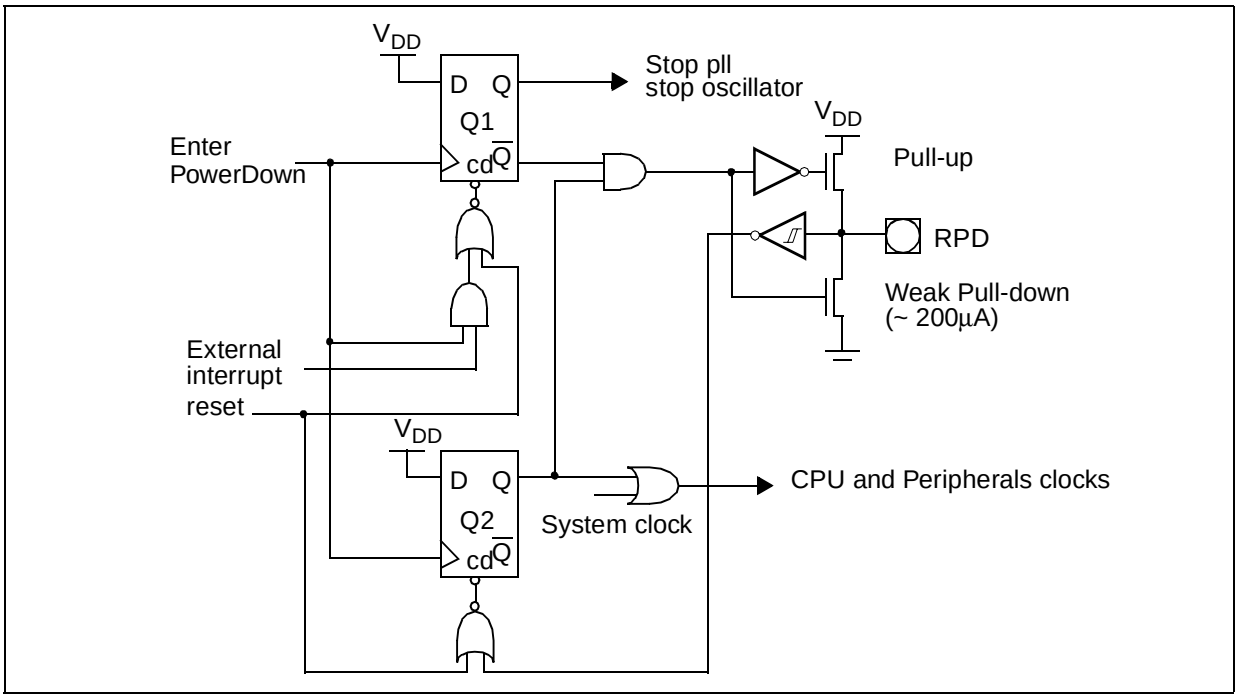
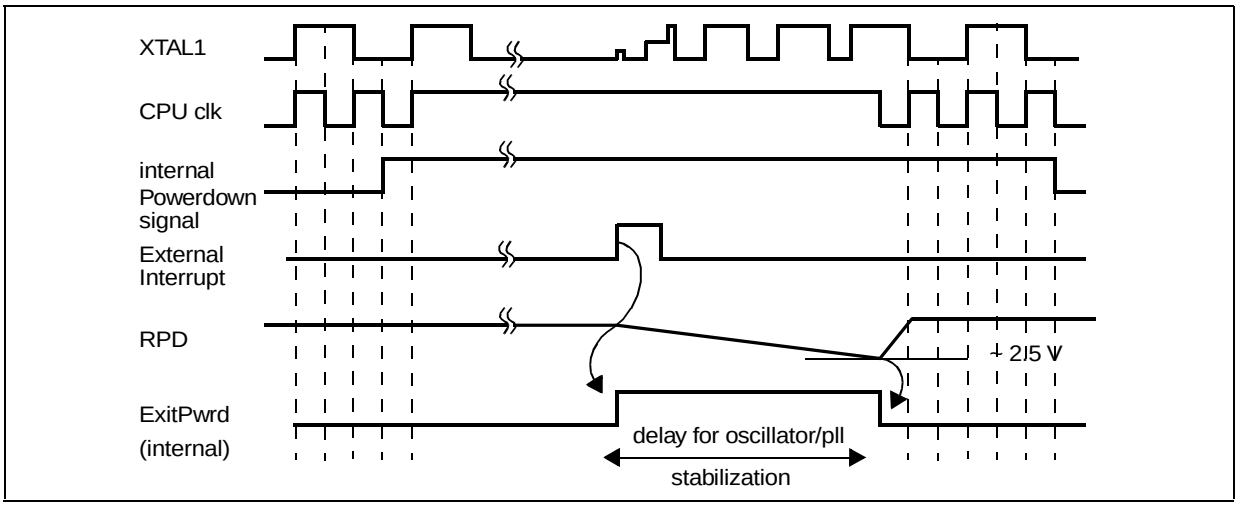


Figure 73 : Powerdown Exit Sequence when Using an External Interrupt (PLL x 2)



19 - SPECIAL FUNCTION REGISTER OVERVIEW

The following table lists all SFRs which are implemented in the ST10F280 in alphabetical order. Bit-addressable SFRs are marked with the letter “b” in column “Name”. SFRs within the Extended SFR-Space (ESFRs) are marked with the letter “E” in column “Physical Address”.

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

The reset value is defined as following:

X : Means the full nibble is not defined at reset.

x : Means some bit of the nibble are not defined at reset.

Table 35 : Special Function Registers Listed by Name

Name	Physical address	8-bit address	Description	Reset value
ADCIC b	FF98h	CCh	A/D Converter end of Conversion Interrupt Control Register	- - 00h
ADCON b	FFA0h	D0h	A/D Converter Control Register	0000h
ADDAT	FEA0h	50h	A/D Converter Result Register	0000h
ADDAT2	F0A0h E	50h	A/D Converter 2 Result Register	0000h
ADDRSEL1	FE18h	0Ch	Address Select Register 1	0000h
ADDRSEL2	FE1Ah	0Dh	Address Select Register 2	0000h
ADDRSEL3	FE1Ch	0Eh	Address Select Register 3	0000h
ADDRSEL4	FE1Eh	0Fh	Address Select Register 4	0000h
ADEIC b	FF9Ah	CDh	A/D Converter Overrun Error Interrupt Control Register	- - 00h
BUSCON0 b	FF0Ch	86h	Bus Configuration Register 0	0xx0h
BUSCON1 b	FF14h	8Ah	Bus Configuration Register 1	0000h
BUSCON2 b	FF16h	8Bh	Bus Configuration Register 2	0000h
BUSCON3 b	FF18h	8Ch	Bus Configuration Register 3	0000h
BUSCON4 b	FF1Ah	8Dh	Bus Configuration Register 4	0000h
CAPREL	FE4Ah	25h	GPT2 Capture/Reload Register	0000h
CC0	FE80h	40h	CAPCOM Register 0	0000h
CC0IC b	FF78h	BCh	CAPCOM Register 0 Interrupt Control Register	- - 00h
CC1	FE82h	41h	CAPCOM Register 1	0000h
CC1IC b	FF7Ah	BDh	CAPCOM Register 1 Interrupt Control Register	- - 00h
CC2	FE84h	42h	CAPCOM Register 2	0000h
CC2IC b	FF7Ch	BEh	CAPCOM Register 2 Interrupt Control Register	- - 00h
CC3	FE86h	43h	CAPCOM Register 3	0000h
CC3IC b	FF7Eh	BFh	CAPCOM Register 3 Interrupt Control Register	- - 00h
CC4	FE88h	44h	CAPCOM Register 4	0000h
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 35 : Special Function Registers Listed 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
CC23IC b	F16Eh E	B7h	CAPCOM Register 23 Interrupt Control Register	- - 00h
CC24	FE70h	38h	CAPCOM Register 24	0000h

Table 35 : Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
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
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

Table 35 : Special Function Registers Listed by Name (continued)

Name		Physical address		8-bit address	Description	Reset value
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
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)	118nh
IDMANUF		F07Eh	E	3Fh	Manufacturer Identifier Register	0401h
IDMEM		F07Ah	E	3Dh	On-chip Memory Identifier Register	3080h
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	PORT0 Low Register (Lower half of PORT0)	- - 00h
P0H	b	FF02h		81h	PORT0 High Register (Upper half of PORT0)	- - 00h
P1L	b	FF04h		82h	PORT1 Low Register (Lower half of PORT1)	- - 00h
P1H	b	FF06h		83h	PORT1 High Register (Upper half of PORT1)	- - 00h
P2	b	FFC0h		E0h	Port 2 Register	0000h
P3	b	FFC4h		E2h	Port 3 Register	0000h
P4	b	FFC8h		E4h	Port 4 Register (8-bit)	- - 00h
P5	b	FFA2h		D1h	Port 5 Register (read only)	XXXXh
P6	b	FFCCh		E6h	Port 6 Register (8-bit)	- - 00h

Table 35 : Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
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
POCON0L	F080h E	40h	PORT0 Low Outpout Control Register (8-bit)	- - 00h
POCON0H	F082h E	41h	PORT0 High Output Control Register (8-bit)	- - 00h
POCON1L	F084h E	42h	PORT1 Low Output Control Register (8-bit)	- - 00h
POCON1H	F086h E	43h	PORT1 High Output Control Register (8-bit)	- - 00h
POCON2	F088h E	44h	Port2 Output Control Register	0000h
POCON3	F08Ah E	45h	Port3 Output Control Register	0000h
POCON4	F08Ch E	46h	Port4 Output Control Register (8-bit)	- - 00h
POCON6	F08Eh E	47h	Port6 Output Control Register (8-bit)	- - 00h
POCON7	F090h E	48h	Port7 Output Control Register (8-bit)	- - 00h
POCON8	F092h E	49h	Port8 Output Control Register (8-bit)	- - 00h
POCON20	F0AAh E	55h	ALE, $\overline{RD}$, $\overline{WR}$ Output Control Register (8-bit)	- - 00h
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

Table 35 : Special Function Registers Listed by Name (continued)

Name	Physical address	8-bit address	Description	Reset value
PWMCON0 b	FF30h	98h	PWM Module Control Register 0	0000h
PWMCON1 b	FF32h	99h	PWM Module Control Register 1	0000h
PWMIC b	F17Eh E	BFh	PWM Module Interrupt Control Register	- - 00h
QR0	F004h E	02h	MAC Unit Offset Register QR0	0000h
QR1	F006h E	03h	MAC Unit Offset Register QR1	0000h
QX0	F000h E	00h	MAC Unit Offset Register QX0	0000h
QX1	F002h E	01h	MAC Unit Offset Register QX1	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 35 : Special Function Registers Listed 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	BEh	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	BFh	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]
XP0IC	b	F186h	E	C3h	CAN1 Module Interrupt Control Register	- - 00h [3]
XP1IC	b	F18Eh	E	C7h	CAN2 Module Interrupt Control Register	- - 00h [3]
XP2IC	b	F196h	E	CBh	XPWM Interrupt Control Register	- - 00h [3]
XP3IC	b	F19Eh	E	CFh	PLL unlock Interrupt Control Register	- - 00h [3]
XPERCON		F024h	E	12h	XPER Configuration Register	- - 05h
ZEROS	b	FF1Ch		8Eh	Constant Value 0's Register (read only)	0000h

Notes: 1. The system configuration is selected during reset.

2. Bit WDTR indicates a watchdog timer triggered reset.

3. The XPnIC Interrupt Control Registers control interrupt requests from integrated X-Bus peripherals. Some software controlled interrupt requests may be generated by setting the XPnIR bits (of XPnIC register) of the unused X-peripheral nodes.

Table 36 : X Registers Listed by Name

Name	Physical address	Description	Reset value
CAN1BTR	EF04h	CAN1 Bit Timing Register	XXXXh
CAN1CSR	EF00h	CAN1 Control/Status Register	XX01h
CAN1GMS	EF06h	CAN1 Global Mask Short	XFXXh
CAN1IR	EF02h	CAN1 Interrupt Register	- - XXh
CAN1LAR1--15	EF14--EFF4h	CAN1 Lower Arbitration register 1 to 15	XXXXh
CAN1LGML	EF0Ah	CAN1 Lower Global Mask Long	XXXXh
CAN1LMLM	EF0Eh	CAN1 Lower Mask Last Message	XXXXh
CAN1MCR1--15	EF10--EFF0h	CAN1 Message Control Register 1 to 15	XXXXh
CAN1MO1--15	EF1x--EFFxh	CAN1 Message Object 1 to 15	XXXXh
CAN1UAR1--15	EF12--EFF2h	CAN1 Upper Arbitration Register 1 to 15	XXXXh
CAN1UGML	EF08h	CAN1 Upper Global Mask Long	XXXXh
CAN1UMLM	EF0Ch	CAN1 Upper Mask Last Message	XXXXh
CAN2BTR	EE04h	CAN2 Bit Timing Register	XXXXh
CAN2CSR	EE00h	CAN2 Control/Status Register	XX01h
CAN2GMS	EE06h	CAN2 Global Mask Short	XFXXh
CAN2IR	EE02h	CAN2 Interrupt Register	- - XXh
CAN2LAR1--15	EE14--EEF4h	CAN2 Lower Arbitration register 1 to 15	XXXXh
CAN2LGML	EE0Ah	CAN2 Lower Global Mask Long	XXXXh
CAN2LMLM	EE0Eh	CAN2 Lower Mask Last Message	XXXXh
CAN2MCR1--15	EE10--EEF0h	CAN2 Message Control Register 1 to 15	XXXXh
CAN2MO1--15	EE1x--EEFxh	CAN2 Message Object 1 to 15	XXXXh
CAN2UAR1--15	EE12--EEF2h	CAN2 Upper Arbitration Register 1 to 15	XXXXh
CAN2UGML	EE08h	CAN2 Upper Global Mask Long	XXXXh
CAN2UMLM	EE0Ch	CAN2 Upper Mask Last Message	XXXXh
XADCMUX	C384h	Port5 or PortX10 ADC Input Selection (Read / Write)	0000h
XDP9	C200h	Direction Register Xport9 (Read / Write)	0000h
XDP9CLR	C204h	Bit Clear Direction Register Xport9 (Write only)	0000h
XDP9SET	C202h	Bit Set Direction Register Xport9 (Write only)	0000h
XODP9	C300H	Open Drain Control Register Xport9 (Read / Write)	0000h
XODP9CLR	C304H	Bit clear Open drain Control register Xport9 (Write only)	0000h
XODP9SET	C302H	Bit Set Open Drain Control Register Xport9 (Write only)	0000h
XP10	C380h	Read only Data register Xport10 (Read only)	0000h
XP10DIDIS	C382h	Xport10 Schmitt Trigger Input Selection (Read / Write)	0000h
XP9	C100h	Data Register Xport9 (Read / Write)	0000h
XP9CLR	C104h	Bit Clear Data Register Xport9 (Write only)	0000h
XP9SET	C102h	Bit Set Data Register Xport9 (Write only)	0000h

Name	Physical address	Description	Reset value
XPOLAR	EC04h	XPWM Channel Polarity Control Register	0000h
XPP0	EC20h	XPWM Period Register 0	0000h
XPP1	EC22h	XPWM Period Register 1	0000h
XPP2	EC24H	XPWM Period Register 2	0000h
XPP3	EC26h	XPWM Period Register 3	0000h
XPT0	EC10h	XPWM Timer Counter Register 0	0000h
XPT1	EC12h	XPWM Timer Counter Register 1	0000h
XPT2	EC14h	XPWM Timer Counter Register 2	0000h
XPT3	EC16h	XPWM Timer Counter Register 3	0000h
XPW0	EC30h	XPWM Pulse Width Register 0	0000h
XPW1	EC32h	XPWM Pulse Width Register 1	0000h
XPW2	EC34h	XPWM Pulse Width Register 2	0000h
XPW3	EC36h	XPWM Pulse Width Register 3	0000h
XPWMCON0	EC00h	XPWM Control Register 0	0000h
XPWMCON1	EC02h	XPWM Control Register 1	0000h
XTCR	C000h	Xtimer Control Register (Read / Write)	0000h
XTCVR	C006h	Xtimer Current Value Register (Read / Write)	0000h
XTEVR	C004h	Xtimer End Value Register (Read / Write)	0000h
XTSVR	C002h	Xtimer Start Value Register (Read / Write)	0000h

19.1 - Identification Registers

The ST10F280 has four Identification registers, mapped in ESFR space. These register contain:

– A manufacturer identifier,

– A chip identifier, with its revision,

– A internal memory and size identifier and programming voltage description.

IDMANUF (F07Eh / 3Fh) [1] ESFR Reset Value: 0401h

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

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

IDCHIP (F07Ch / 3Eh) [1] ESFR Reset Value: 118Xh

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

REVID	**Device Revision Identifier**
CHIPID	**Device Identifier** 118h: ST10F280 identifier.

IDMEM (F07Ah / 3Dh) [1] ESFR Reset Value: 3080h

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

MEMSIZE	**Internal Memory Size** is calculated using the following formula: Size = 4 x [MEMSIZE] (in K Byte) 080h for ST10F280 (512K Byte)
MEMTYP	**Internal Memory Type** 3h for ST10F280 (Flash memory).

IDPROG (F078h / 3Ch) [1] ESFR Reset Value: 0040h

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

PROGVDD	**Programming V_{DD} Voltage** V_{DD} voltage when programming EPROM or FLASH devices is calculated using the following formula: V_{DD} = 20 x [PROGVDD] / 256 (volts) 40h for ST10F280 (5V).
PROGVPP	**Programming V_{PP} Voltage** (no need of external V_{PP}) 00h

Note : 1. All identification words are read only registers.

19.2 - System Configuration Registers

The ST10F280 has registers used for different configuration of the overall system. These registers are described below.

SYSCON (FF12h / 89h) SFR Reset Value: 0xx0h

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

Notes: 1. These bit are set directly or indirectly according to PORT0 and $\overline{EA}$ pin configuration during reset sequence.
2. Register SYSCON cannot be changed after execution of the EINIT instruction.

XPER-SHARE	**XBUS Peripheral Share Mode Control** '0': External accesses to XBUS peripherals are disabled '1': XBUS peripherals are accessible via the external bus during hold mode
VISIBLE	**Visible Mode Control** '0': Accesses to XBUS peripherals are done internally '1': XBUS peripheral accesses are made visible on the external pins
XPEN	**XBUS Peripheral Enable bit** '0': Accesses to the on-chip X-Peripherals and XRAM are disabled '1': The on-chip X-Peripherals are enabled.
BDRSTEN	**Bidirectional Reset Enable** '0': $\overline{RSTIN}$ pin is an input pin only. (SW Reset or WDT Reset have no effect on this pin) '1': $\overline{RSTIN}$ pin is a bidirectional pin. This pin is pulled low during 1024 TCL during reset sequence.
OWDDIS	**Oscillator Watchdog Disable Control** '0': Oscillator Watchdog (OWD) is enabled. If PLL is bypassed, the OWD monitors XTAL1 activity. If there is no activity on XTAL1 for at least 1 µs, the CPU clock is switched automatically to PLL's base frequency (2 to 10MHz). '1': OWD is disabled. If the PLL is bypassed, the CPU clock is always driven by XTAL1 signal. The PLL is turned off to reduce power supply current.
PWDCFG	**Power Down Mode Configuration Control** '0': Power Down Mode can only be entered during PWRDN instruction execution if $\overline{NMI}$ pin is low, otherwise the instruction has no effect. Exit power down only with reset. '1': Power Down Mode can only be entered during PWRDN instruction execution if all enabled fast external interrupt EXxIN pins are in their inactive level. Exiting this mode can be done by asserting one enabled EXxIN pin or with external reset.
CSCFG	**Chip Select Configuration Control** '0': Latched Chip Select lines: CSx change 1 TCL after rising edge of ALE '1': Unlatched Chip Select lines: CSx change with rising edge of ALE.
WRCFG	**Write Configuration Control** (Inverted copy of bit WRC of RP0H) '0': Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function '1': Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$.
CLKEN	**System Clock Output Enable (CLKOUT)** '0': CLKOUT disabled: pin may be used for general purpose I/O '1': CLKOUT enabled: pin outputs the system clock signal.

BYTDIS	**Disable/Enable Control for Pin $\overline{BHE}$ (Set according to data bus width)** '0': Pin $\overline{BHE}$ enabled '1': Pin $\overline{BHE}$ disabled, pin may be used for general purpose I/O.
ROMEN	**Internal Memory Enable (Set according to pin $\overline{EA}$ during reset)** '0': Internal Memory disabled: accesses to the Memory area use the external bus '1': Internal Memory enabled.
SGTDIS	**Segmentation Disable/Enable Control** '0': Segmentation enabled (CSP is saved/restored during interrupt entry/exit) '1': Segmentation disabled (Only IP is saved/restored).
ROMS1	**Internal Flash Memory Mapping** '0': Internal Flash Memory area mapped to segment 0 (00'0000H...00'7FFFH) '1': Internal Flash Memory area mapped to segment 1 (01'0000H...01'7FFFH).
STKSZ	**System Stack Size** Selects the size of the system stack (in the internal RAM) from 32 to 1024 words.

Table 37 : Stack Size Selection

<STKSZ>	Stack Size (Words)	Internal RAM Addresses (Words) of Physical Stack	Significant Bits of Stack Pointer SP
0 0 0 b	256	00'FBFEh...00'FA00h (Default after Reset)	SP.8...SP.0
0 0 1 b	128	00'FBFEh...00'FB00h	SP.7...SP.0
0 1 0 b	64	00'FBFEh...00'FB80h	SP.6...SP.0
0 1 1 b	32	00'FBFEh...00'FBC0h	SP.5...SP.0
1 0 0 b	512	00'FBFEh...00'F800h (not for 1K Byte IRAM)	SP.9...SP.0
1 0 1 b	-	Reserved. Do not use this combination	-
1 1 0 b	-	Reserved. Do not use this combination	-
1 1 1 b	1024	00'FDFEh...00'FX00h (Note: No circular stack) 00'FX00h represents the lower IRAM limit, i.e. 1K Byte: 00'FA00h, 2K Byte: 00'F600h, 3K Byte: 00'F200h	SP.11...SP.0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN0	CSREN0	RDYPOL0	RDYEN0	-	**BUS ACT0**	**ALE CTL0**	-	**BTYP**		MTTC0	RWDC0	MCTC			
RW	RW	RW	RW		RW[2]	RW[2]		RW[1]		RW	RW	RW			

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

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

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

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

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

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

BUSCON4 (FF1Ah / 8Dh) SFR Reset Value: 0000h

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

Notes: 1. BTYP (bit 6 and 7) are set according to the configuration of the bit l1 and l2 of PORT0 latched at the end of the reset sequence.
2. BUSCON0 is initialized with 0000h, if $\overline{EA}$ pin is high during reset. If $\overline{EA}$ pin is low during reset, bit BUSACT0 and ALECTRL0 are set ('1') and bit field BTYP is loaded with the bus configuration selected via PORT0.

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

RP0H (F108h / 84h) ESFR Reset Value: - - XXH

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

WRC [2]	**Write Configuration Control** '0': Pin $\overline{WR}$ acts as $\overline{WRL}$, pin $\overline{BHE}$ acts as $\overline{WRH}$ '1': Pins $\overline{WR}$ and $\overline{BHE}$ retain their normal function
CSSEL [2]	**Chip Select Line Selection (Number of active $\overline{CS}$ outputs)** 0 0: 3 $\overline{CS}$ lines: $\overline{CS}$2...$\overline{CS}$0 0 1: 2 $\overline{CS}$ lines: $\overline{CS}$1...$\overline{CS}$0 1 0: No $\overline{CS}$ lines at all 1 1: 5 $\overline{CS}$ lines: $\overline{CS}$4...$\overline{CS}$0 (Default without pull-downs)
SALSEL [2]	**Segment Address Line Selection (Number of active segment address outputs)** 0 0: 4-bit segment address: A19...A16 0 1: No segment address lines at all 1 0: 8-bit segment address: A23...A16 1 1: 2-bit segment address: A17...A16 (Default without pull-downs)
CLKSEL [1 - 2]	**System Clock Selection** 000: $f_{CPU} = 2.5 \times f_{OSC}$ 001: $f_{CPU} = 0.5 \times f_{OSC}$ 010: $f_{CPU} = 10 \times f_{OSC}$ 011: $f_{CPU} = f_{OSC}$ 100: $f_{CPU} = 5 \times f_{OSC}$ 101: $f_{CPU} = 2 \times f_{OSC}$ 110: $f_{CPU} = 3 \times f_{OSC}$ 111: $f_{CPU} = 4 \times f_{OSC}$

Notes: 1. RP0H.7 to RP0H.5 bits are loaded only during a long hardware reset. As pull-up resistors are active on each Port P0H pins during reset, RP0H default value is "FFh".

2. These bits are set according to Port 0 configuration during any reset sequence.

EXICON (**F1C0h** / **E0h**) ESFR Reset Value: 0000h

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

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

EXISEL (F1DAh / EDh) ESFR Reset Value: 0000h

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

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

EXIxSS	**Port 2 pin**	**Alternate Source**
0	P2.8	CAN1_RxD
1	P2.9	CAN2_RxD
2...7	P2.10...15	Not used (zero)

XP3IC (F19Eh / CFh) [1] ESFR Reset Value: - - 00h

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

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

xxIC (yyyyh / zzh) SFR Area Reset Value: - - 00h

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

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

XPERCON (F024h / 12h) ESFR Reset Value: - - 05h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	XPWMEN	XPERCONEN3	XRAMEN	CAN2EN	CAN1EN
											RW	RW	RW	RW	RW

Bit	Function
CAN1EN	**CAN1 Enable Bit** 0 Accesses to the on-chip CAN1 XPeripheral and its functions are disabled. P4.5 and P4.6 pins can be used as general purpose I/Os. Address range 00'EF00h-00'EFFFh is only directed to external memory if CAN2EN and XPWM bits are cleared also. 1 The on-chip CAN1 XPeripheral is enabled and can be accessed.
CAN2EN	**CAN2 Enable Bit** 0 Accesses to the on-chip CAN2 XPeripheral and its functions are disabled. P4.4 and P4.7 pins can be used as general purpose I/Os. Address range 00'EE00h-00'EEFFh is only directed to external memory if CAN1EN and XPWM bits are cleared also. 1 The on-chip CAN2 XPeripheral is enabled and can be accessed.
XRAMEN	**XRAM Enable Bit** 0 Accesses to the on-chip 16K Byte XRAM are disabled, external access performed. 1 The on-chip 16K Byte XRAM is enabled and can be accessed.
XPERCONEN3	**XPORT9,XTIMER, XPORT10, XADCMUX Enable Bit** 0 Accesses to the XPORT9, XTIMER, XPORT10, XADCMUX peripherals are disabled, external access performed. 1 The on-chip XPORT9, XTIMER, XPORT10, XADCMUX peripherals are enabled and can be accessed.
XPWMEN	**XPWM Enable Bit** 0 Accesses to the on-chip XPWM are disabled, external access performed. Address range 00'EC00h-00'ECFFh is only directed to external memory if CAN1EN and CAN2EN are '0' also 1 The on-chip XPWM is enabled and can be accessed.

Note: - When both CAN and XPWM are disabled via XPERCON setting, then any access in the address range 00'EC00h 00'EFFFh will be directed to external memory interface, using the BUSCONx register corresponding to address matching ADDRSELx register. P4.4 and P4.7 can be used as General Purpose I/O when CAN2 is not enabled, and P4.5 and P4.6 can be used as General Purpose I/O when CAN1 is not enabled.

- The default XPER selection after Reset is : XCAN1 is enabled, XCAN2 is disabled, XRAM is enabled, XPORT9, XTIMER, XPORT10, XPWM are disabled.

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

20 - ELECTRICAL CHARACTERISTICS

20.1 - Absolute Maximum Ratings

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

Note: 1. Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During overload conditions ($V_{IN} > V_{DD}$ 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.

20.2 - Parameter Interpretation

The parameters listed in the following tables represent the characteristics of the ST10F280 and its demands on the system. Where the ST10F280 logic provides signals with their respective timing characteristics, the symbol "CC" for Controller Characteristics, is included in the "Symbol" column.

Where the external system must provide signals with their respective timing characteristics to the ST10F280, the symbol "SR" for System Requirement, is included in the "Symbol" column.

20.3 - DC Characteristics

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

Symbol		Parameter		Test Conditions	Min.	Max.	Unit
V_{IL}	SR	Input low voltage		–	-0.5	0.2 V_{DD} - 0.1	V
V_{ILS}	SR	Input low voltage (special threshold)		–	-0.5	2.0	V
V_{IH}	SR	Input high voltage (all except $\overline{RSTIN}$ and XTAL1)		–	0.2 V_{DD} + 0.9	V_{DD} + 0.5	V
V_{IH1}	SR	Input high voltage $\overline{RSTIN}$		–	0.6 V_{DD}	V_{DD} + 0.5	V
V_{IH2}	SR	Input high voltage XTAL1		–	0.7 V_{DD}	V_{DD} + 0.5	V
V_{IHS}	SR	Input high voltage (special threshold)		–	0.8 V_{DD} - 0.2	V_{DD} + 0.5	V
HYS		Input Hysteresis (special threshold)	3	–	400	–	mV
V_{OL}	CC	Output low voltage (PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT, $\overline{RSTOUT}$)	1	I_{OL} = 2.4mA	–	0.45	V
V_{OL1}	CC	Output low voltage (all other outputs)	1	I_{OL1} = 1.6mA	–	0.45	V

Symbol		Parameter		Test Conditions	Min.	Max.	Unit
V_{OH}	CC	Output high voltage (PORT0, PORT1, Port4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT, $\overline{RSTOUT}$)	1	I_{OH} = -500µA I_{OH} = -2.4mA	0.9 V_{DD} 2.4	– –	V
V_{OH1}	CC	Output high voltage (all other outputs)	1/2	I_{OH} = – 250µA I_{OH} = – 1.6mA	0.9 V_{DD} 2.4	– –	V V
$\lvert I_{OZ1} \rvert$	CC	Input leakage current (Port 5, XPort 10)		0V < V_{IN} < V_{DD}	–	0.2	µA
$\lvert I_{OZ2} \rvert$	CC	Input leakage current (all other)		0V < V_{IN} < V_{DD}	–	1	µA
$\lvert I_{OV} \rvert$	SR	Overload current	3/4		–	5	mA
R_{RST}	CC	$\overline{RSTIN}$ pull-up resistor	3	–	50	250	kΩ
I_{RWH}		Read / Write inactive current	5/6	V_{OUT} = 2.4V	–	-40	µA
I_{RWL}		Read / Write active current	5/7	V_{OUT} = V_{OLmax}	-500	–	µA
I_{ALEL}		ALE inactive current	5/6	V_{OUT} = V_{OLmax}	40	–	µA
I_{ALEH}		ALE active current	5/7	V_{OUT} = 2.4V	–	500	µA
I_{P6H}		Port 6 inactive current	5/6	V_{OUT} = 2.4V	–	-40	µA
I_{P6L}		Port 6 active current	5/7	V_{OUT} = V_{OL1max}	-500	–	µA
I_{P0H}		PORT0 configuration current	5/6	V_{IN} = V_{IHmin}	–	-10	µA
I_{P0L}			5/7	V_{IN} = V_{ILmax}	-100	–	µA
$\lvert I_{IL} \rvert$	CC	XTAL1 input current		0V < V_{IN} < V_{DD}	–	20	µA
C_{IO}	CC	Pin capacitance (digital inputs / outputs)	3/5	f = 1MHz, T_A = 25°C	–	10	pF
I_{CC}		Power supply current	8	RSTIN = V_{IH1} f_{CPU} in [MHz]	–	30 + 3.3 x f_{CPU}	mA
I_{ID}		Idle mode supply current	9	RSTIN = V_{IH1} f_{CPU} in [MHz]	–	20 + f_{CPU}	mA
I_{PD}		Power-down mode supply current	10	V_{DD} = 5.5V T_A = 55°C	–	200	µA

Notes: 1. ST10F280 pins are equipped with low-noise output drivers which significantly improve the device's EMI performance. These low-noise drivers deliver their maximum current only until the respective target output level is reached. After this, the output current is reduced. This results in increased impedance of the driver, which attenuates electrical noise from the connected PCB tracks. The current specified in column "Test Conditions" is delivered in any cases.

2. This specification is not valid for outputs which are switched to open drain mode. In this case the respective output will float and the voltage results from the external circuitry.
3. Partially tested, guaranteed by design characterization.

4. Overload conditions occur if the standard operating conditions are exceeded, i.e. the voltage on any pin exceeds the specified range (i.e. V_{OV} > V_{DD}+0.5V or V_{OV} <0.5V). The absolute sum of input overload currents on all port pins may not exceed 50mA. The supply voltage must remain within the specified limits.

5. This specification is only valid during Reset, or during Hold-mode or Adapt-mode. Port 6 pins are only affected if they are used for $\overline{CS}$ output and if their open drain function is not enabled.

6. The maximum current may be drawn while the respective signal line remains inactive.

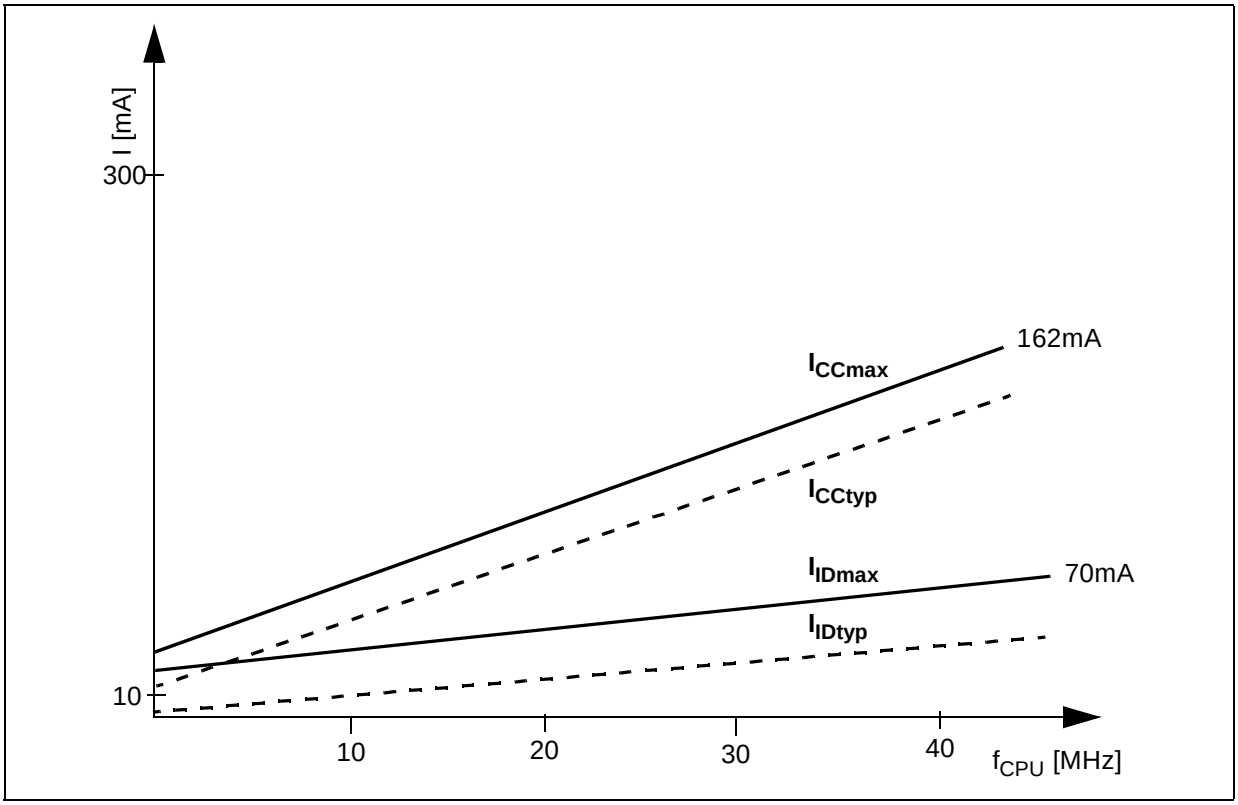
7. The minimum current must be drawn in order to drive the respective signal line active.

8. The power supply current is a function of the operating frequency. This dependency is illustrated in the Figure 74. These parameters are tested at V_{DD}max and 40MHz CPU clock with all outputs disconnected and all inputs at VIL or VIH. The chip is configured with a demultiplexed 16-bit bus, direct clock drive, 5 chip select lines and 2 segment address lines, $\overline{EA}$ pin is low during reset. After reset, PORT 0 is driven with the value '00CCh' that produces infinite execution of NOP instruction with 15 wait-state, R/W delay, memory tristate wait state, normal ALE. Peripherals are not activated.

9. Idle mode supply current is a function of the operating frequency. This dependency is illustrated in the Figure 74. These parameters are tested at V_{DD}max and 40MHz CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}.

10. This parameter value includes leakage currents. With all inputs (including pins configured as inputs) at 0 V to 0.1V or at V_{DD} – 0.1V to V_{DD}, V_{REF} = 0V, all outputs (including pins configured as outputs) disconnected.

Figure 74 : Supply / Idle Current as a Function of Operating Frequency



20.3.1 - A/D Converter Characteristics

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

Table 38 : A/D Converter Characteristics

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

Notes: 1. V_{AIN} may exceed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases will be X000h or X3FFh, respectively.

2. During the t_S sample time the input capacitance C_{ain} can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach its final voltage level within the t_S sample time. After the end of the t_S sample time, changes of the analog input voltage have no effect on the conversion result. Values for the t_{SC} sample clock depend on the programming. Referring to the t_C conversion time formula of section 20.3.2 and to the table 39 of page 156:

- t_S min = 2 t_{SC} min = 2 t_{CC} min = 2 x 24 x TCL = 48 TCL

- t_S max = 2 t_{SC} max = 2 x 8 t_{CC} max = 2 x 8 x 96 TCL = 1536 TCL

TCL is defined in section 20.4.5 at page 159.

3. The conversion time formula is:

- t_C = 14 t_{CC} + t_S + 4 TCL (= 14 t_{CC} + 2 t_{SC} + 4 TCL)

The t_C parameter includes the t_S sample time, the time for determining the digital result and the time to load the result register with the result of the conversion. Values for the t_{CC} conversion clock depend on the programming. Referring to the table 39 of page 156:

- t_C min = 14 t_{CC} min + t_S min + 4 TCL = 14 x 24 x TCL + 48 TCL + 4 TCL = 388 TCL

- t_C max = 14 t_{CC} max + t_S max + 4 TCL = 14 x 96 TCL + 1536 TCL + 4 TCL = 2884 TCL

4. This parameter is fixed by ADC control logic.

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

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

7. Partially tested, guaranteed by design characterization.

8.To remove noise and undesirable high frequency components from the analog input signal, a low-pass filter must be connected at the ADC input. The cut-off frequency of this filter should avoid 2 opposite transitions during the t_s sampling time of the ST10 ADC:

- $f_{cut\text{-}off}$ ≤ 1 / 5 t_s to 1/10 t_s

where t_s is the sampling time of the ST10 ADC and is not related to the Nyquist frequency determined by the t_c conversion time.

20.3.2 - Conversion Timing Control

When a conversion is started, first the capacitances of the converter are loaded via the respective analog input pin to the current analog input voltage. The time to load the capacitances is referred to as the sample time t_S. Next the sampled voltage is converted to a digital value in 10 successive steps, which correspond to the 10-bit resolution of the ADC. The next 4 steps are used for equalizing internal levels (and are keep for exact timing matching with the 10-bit A/D converter module implemented in ST10F168).

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

The sample time t_S (= 2 t_{SC}) and the conversion time t_C (= 14 t_{CC} + 2 t_{SC} + 4 TCL) can be programmed relatively to the ST10F280 CPU clock. This allows adjusting the A/D converter of the ST10F280 to the properties of the system:

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

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

The conversion times are programmed via the upper four bit of register ADCON. Bit field ADCTC (conversion time control) selects the basic conversion clock t_{CC}, used for the 14 steps of converting. The sample time t_S is a multiple of this conversion time and is selected by bit field ADSTC (sample time control). The table below lists the possible combinations. The timings refer to the unit TCL, where f_{CPU} = 1/2 TCL.

Table 39 : ADC Sampling and Conversion Timing

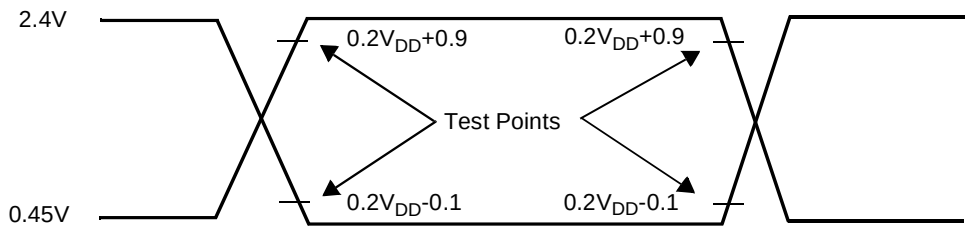
ADCTC	Conversion Clock t_{CC}		ADSTC	Sample Clock t_{SC}	
	TCL = 1/2 x f_{XTAL}	At f_{CPU} = 40MHz		t_{SC} =	At f_{CPU} = 40MHz and ADCTC = 00
00	TCL x 24	0.3µs	00	t_{CC}	0.3µs
01	Reserved, do not use	Reserved	01	t_{CC} x 2	0.6µs
10	TCL x 96	1.2 µs	10	t_{CC} x 4	1.2µs
11	TCL x 48	0.6 µs	11	t_{CC} x 8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

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

20.4 - AC characteristics

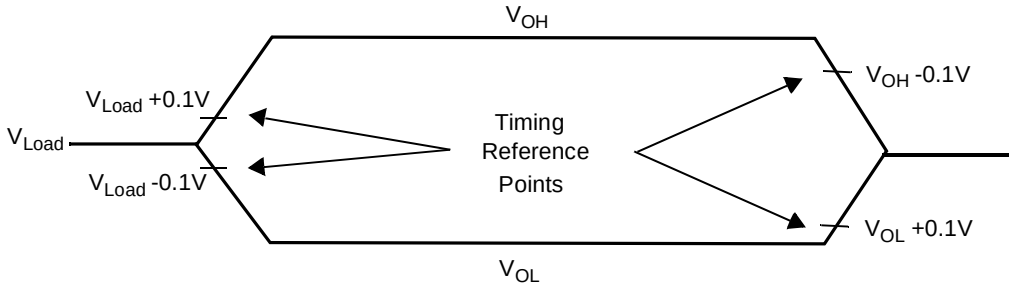
20.4.1 - Test Waveforms

Figure 75 : Input / Output Waveforms



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

Figure 76 : Float Waveforms



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

20.4.2 - Definition of Internal Timing

The internal operation of the ST10F280 is controlled by the internal CPU clock f_{CPU}. Both edges of the CPU clock can trigger internal (for example pipeline) or external (for example bus cycles) operations.

The specification of the external timing (AC Characteristics) therefore depends on the time between two consecutive edges of the CPU clock, called "TCL".

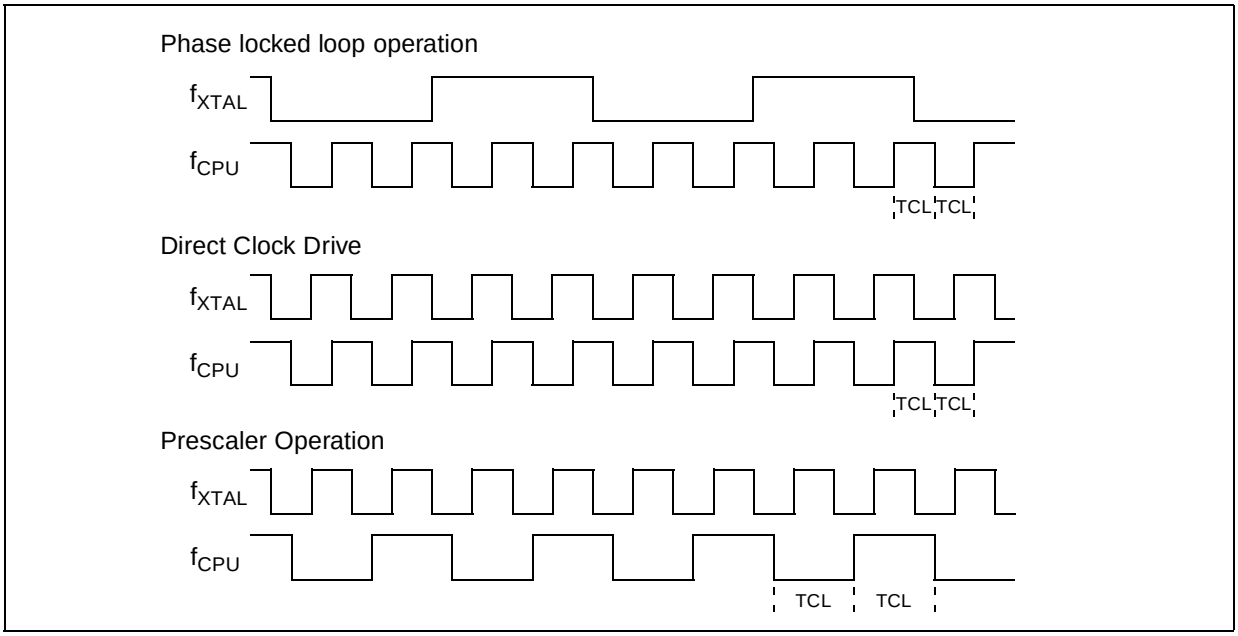
The CPU clock signal can be generated by different mechanisms. The duration of TCL and its variation (and also the derived external timing) depends on the mechanism used to generate f_{CPU}.

This influence must be regarded when calculating the timings for the ST10F280.

The example for PLL operation shown in Figure 77 refers to a PLL factor of 4.

The mechanism used to generate the CPU clock is selected during reset by the logic levels on pins P0.15-13 (P0H.7-5).

Figure 77 : Generation Mechanisms for the CPU Clock



20.4.3 - Clock Generation Modes

The Table 40 associates the combinations of these three bit with the respective clock generation mode.

Table 40 : CPU Frequency Generation

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

Notes: 1. The external clock input range refers to a CPU clock range of 1...40MHz.
2. The maximum depends on the duty cycle of the external clock signal.
3. The maximum input frequency is 25MHz when using an external crystal with the internal oscillator; providing that internal serial resistance of the crystal is less than 40Ω. However, higher frequencies can be applied with an external clock source on pin XTAL1, but in this case, the input clock signal must reach the defined levels V_{IL} and V_{IH2}.
4. The PLL free-running frequency is from 2 to 10MHz.

20.4.4 - Prescaler Operation

When pins P0.15-13 (P0H.7-5) equal '001' during reset, the CPU clock is derived from the internal oscillator (input clock signal) by a 2:1 prescaler.
The frequency of f_{CPU} is half the frequency of f_{XTAL} and the high and low time of f_{CPU} (i.e. the duration of an individual TCL) is defined by the period of the input clock f_{XTAL}.

The timings listed in the AC Characteristics that refer to TCL therefore can be calculated using the period of f_{XTAL} for any TCL.

Note that if the bit OWDDIS in SYSCON register is cleared, the PLL runs on its free-running frequency and delivers the clock signal for the Oscillator Watchdog. If bit OWDDIS is set, then the PLL is switched off.

20.4.5 - Direct Drive

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

The frequency of f_{CPU} directly follows the frequency of f_{XTAL} so the high and low time of f_{CPU} (i.e. the duration of an individual TCL) is defined by the duty cycle of the input clock f_{XTAL}.

Therefor, the timings given in this chapter refer to the minimum TCL. This minimum value can be calculated by the following formula:

$$TCL_{min} = 1/f_{XTAL} \times DC_{min}$$

$$DC = \text{duty cycle}$$

For two consecutive TCLs, the deviation caused by the duty cycle of f_{XTAL} is compensated, so the duration of 2 TCL is always $1/f_{XTAL}$.

The minimum value TCL_{min} has to be used only once for timings that require an odd number of TCLs (1,3,...). Timings that require an even number of TCLs (2,4,...) may use the formula:

$$2TCL = 1/f_{XTAL}$$

Note: 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 bit OWDDIS in SYSCON register is cleared, the PLL runs on its free-running frequency and delivers the clock signal for the Oscillator Watchdog. If bit OWDDIS is set, then the PLL is switched off.

20.4.6 - Oscillator Watchdog (OWD)

An on-chip watchdog oscillator is implemented in the ST10F280. This feature is used for safety operation with external crystal oscillator (using direct drive mode with or without prescaler). This watchdog oscillator operates as following :

The reset default configuration enables the watchdog oscillator. It can be disabled by setting the OWDDIS (bit 4) of SYSCON register.

When the OWD is enabled, the PLL runs at its free-running frequency, and it increments the watchdog counter. The PLL free-running frequency is from 2 to 10MHz. On each transition of external clock, the watchdog counter is cleared. If an external clock failure occurs, then the watchdog counter overflows (after 16 PLL clock cycles).

The CPU clock signal will be switched to the PLL free-running clock signal, and the oscillator watchdog Interrupt Request (XP3INT) is flagged. The CPU clock will not switch back to the external clock even if a valid external clock exits on XTAL1 pin. Only a hardware reset can switch the CPU clock source back to direct clock input.

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

20.4.7 - Phase Locked Loop

For all other combinations of pins P0.15-13 (P0H.7-5) during reset the on-chip phase locked loop is enabled and it provides the CPU clock (see Table 40). The PLL multiplies the input frequency by the factor F which is selected via the combination of pins P0.15-13 ($f_{CPU} = f_{XTAL} \times F$). With every F'th transition of f_{XTAL} the PLL circuit synchronizes the CPU clock to the input clock. This synchronization is done smoothly, so the CPU clock frequency does not change abruptly.

Due to this adaptation to the input clock the frequency of f_{CPU} is constantly adjusted so it is locked to f_{XTAL}. The slight variation causes a jitter of f_{CPU} which also effects the duration of individual TCLs.

The timings listed in the AC Characteristics that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances.

The real minimum value for TCL depends on the jitter of the PLL. The PLL tunes f_{CPU} to keep it locked on f_{XTAL}. The relative deviation of TCL is the maximum when it is refered to one TCL period. It decreases according to the formula and to the Figure 78 given below. For *N* periods of TCL the minimum value is computed using the corresponding deviation D_N:

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

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

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

D_3 = 4 - 3/15 = 3.8%

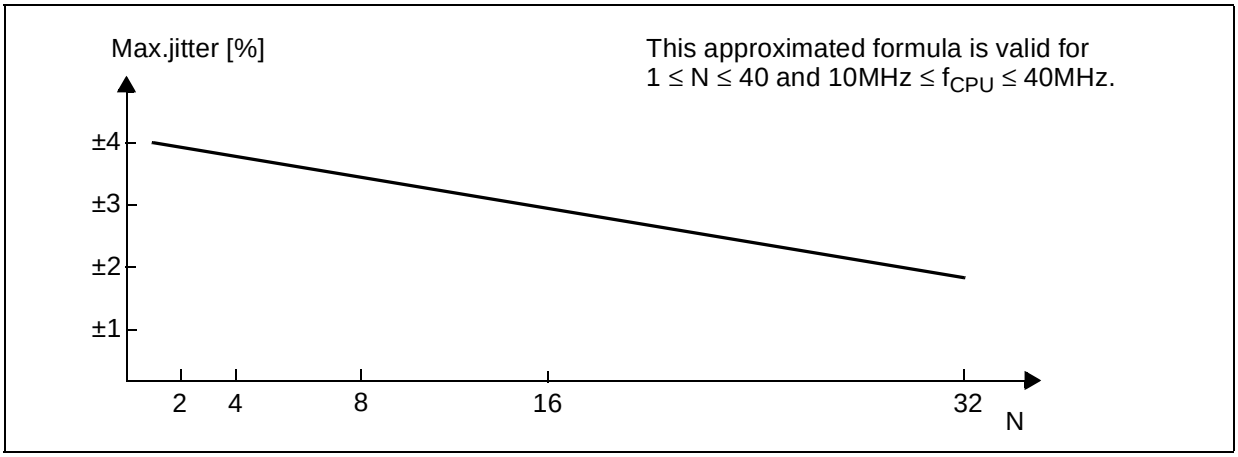
$3\ TCL_{min}$ = $3\ TCL_{NOM}$ x (1 - 3.8/100)

= $3\ TCL_{NOM}$ x 0.962

$3\ TCL_{min}$ = (36.075ns at f_{CPU} = 40MHz)

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

Figure 78 : Approximated Maximum PLL Jitter



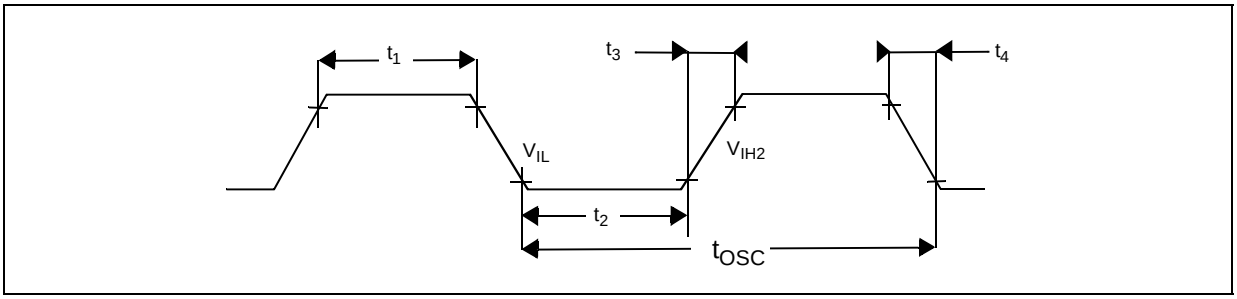
20.4.8 - External Clock Drive XTAL1

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

Parameter	Symbol		$f_{CPU} = f_{XTAL}$		$f_{CPU} = f_{XTAL} / 2$		$f_{CPU} = f_{XTAL} \times F$ F = 2 / 2.5 / 3 / 4 / 5 / 10		Unit
			min	max	min	max	min	max	
Oscillator period	t_{OSC}	SR	25 [1]	–	12.5	–	40 x N	100 x N	ns
High time	t_1	SR	10 [2]	–	5 [2]	–	10 [2]	–	ns
Low time	t_2	SR	10 [2]	–	5 [2]	–	10 [2]	–	ns
Rise time	t_3	SR	–	3 [2]	–	3 [3]	–	3 [2]	ns
Fall time	t_4	SR	–	3 [2]	–	3 [2]	–	3 [2]	ns

Notes: 1. Theoretical minimum. The real minimum value depends on the duty cycle of the input clock signal. 25MHz is the maximum input frequency when using an external crystal oscillator. Howevwer, 40MHz can be applied with an external clock source.
2. The input clock signal must reach the defined levels V_{IL} and V_{IH2}.

Figure 79 : External Clock Drive XTAL1



20.4.9 - Memory Cycle Variables

The tables below use three variables which are derived from the BUSCONx registers and represent the special characteristics of the programmed memory cycle. The following table describes, how these variables are to be computed.

Description	Symbol	Values
ALE Extension	t_A	TCL x [ALECTL]
Memory Cycle Time wait states	t_C	2 TCL x (15 - [MCTC])
Memory Tri-state Time	t_F	2 TCL x (1 - [MTTC])

20.4.10 - Multiplexed Bus

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

Table 41 : Multiplexed Bus Characteristics

Symbol		Parameter	Max. CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		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	2 + t_A	–	TCL - 10.5 + t_A	–	ns
t_7	CC	Address hold after ALE 1	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_8	CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_9	CC	ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	-8.5 + t_A	–	-8.5 + t_A	–	ns
t_{10}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay) 1	–	6	–	6	ns
t_{11}	CC	Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay) 1	–	18.5	–	TCL + 6	ns
t_{12}	CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2 TCL -9.5 + t_C	–	ns
t_{13}	CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3 TCL -9.5 + t_C	–	ns
t_{14}	SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2 TCL - 19 + t_C	ns
t_{15}	SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3 TCL - 19 + t_C	ns
t_{16}	SR	ALE low to valid data in	–	18.5 + t_A + t_C	–	3 TCL - 19 + t_A + t_C	ns
t_{17}	SR	Address/Unlatched $\overline{CS}$ to valid data in	–	22 + $2t_A$ + t_C	–	4 TCL - 28 + $2t_A$ + t_C	ns
t_{18}	SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{19}	SR	Data float after $\overline{RD}$ 1	–	16.5 + t_F	–	2 TCL - 8.5 + t_F	ns
t_{22}	CC	Data valid to $\overline{WR}$	10 + t_C	–	2 TCL -15 + t_C	–	ns
t_{23}	CC	Data hold after $\overline{WR}$	4 + t_F	–	2 TCL - 8.5 + t_F	–	ns
t_{25}	CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	15 + t_F	–	2 TCL -10 + t_F	–	ns
t_{27}	CC	Address/Unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	10 + t_F	–	2 TCL -15 + t_F	–	ns
t_{38}	CC	ALE falling edge to Latched $\overline{CS}$	-4 - t_A	10 - t_A	-4 - t_A	10 - t_A	ns
t_{39}	SR	Latched $\overline{CS}$ low to Valid Data In	–	18.5 + t_C + $2t_A$	–	3 TCL - 19 + t_C + $2t_A$	ns
t_{40}	CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	27 + t_F	–	3 TCL - 10.5 + t_F	–	ns

Table 41 : Multiplexed Bus Characteristics

Symbol		Parameter	Max. CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		Unit
			min.	max.	min.	max.	
t_{42}	CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	7 + t_A	–	TCL - 5.5+ t_A	–	ns
t_{43}	CC	ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay)	-5.5 + t_A	–	-5.5 + t_A	–	ns
t_{44}	CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay) 1	–	0	–	0	ns
t_{45}	CC	Address float after $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay) 1	–	12.5	–	TCL	ns
t_{46}	SR	$\overline{RdCS}$ to Valid Data In (with RW delay)	–	4 + t_C	–	2 TCL - 21 + t_C	ns
t_{47}	SR	$\overline{RdCS}$ to Valid Data In (no RW delay)	–	16.5 + t_C	–	3 TCL - 21 + t_C	ns
t_{48}	CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW delay)	15.5 + t_C	–	2 TCL - 9.5 + t_C	–	ns
t_{49}	CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW delay)	28 + t_C	–	3 TCL - 9.5 + t_C	–	ns
t_{50}	CC	Data valid to $\overline{WrCS}$	10 + t_C	–	2 TCL - 15+ t_C	–	ns
t_{51}	SR	Data hold after $\overline{RdCS}$	0	–	0	–	ns
t_{52}	SR	Data float after $\overline{RdCS}$ 1	–	16.5 + t_F	–	2 TCL - 8.5+t_F	ns
t_{54}	CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	6 + t_F	–	2 TCL - 19 + t_F	–	ns
t_{56}	CC	Data hold after $\overline{WrCS}$	6 + t_F	–	2 TCL - 19 + t_F	–	ns

Note: 1. Partially tested, guaranted by design characterization.

Figure 80 : External Memory Cycle : Multiplexed Bus, With / Without Read / Write Delay, Normal ALE

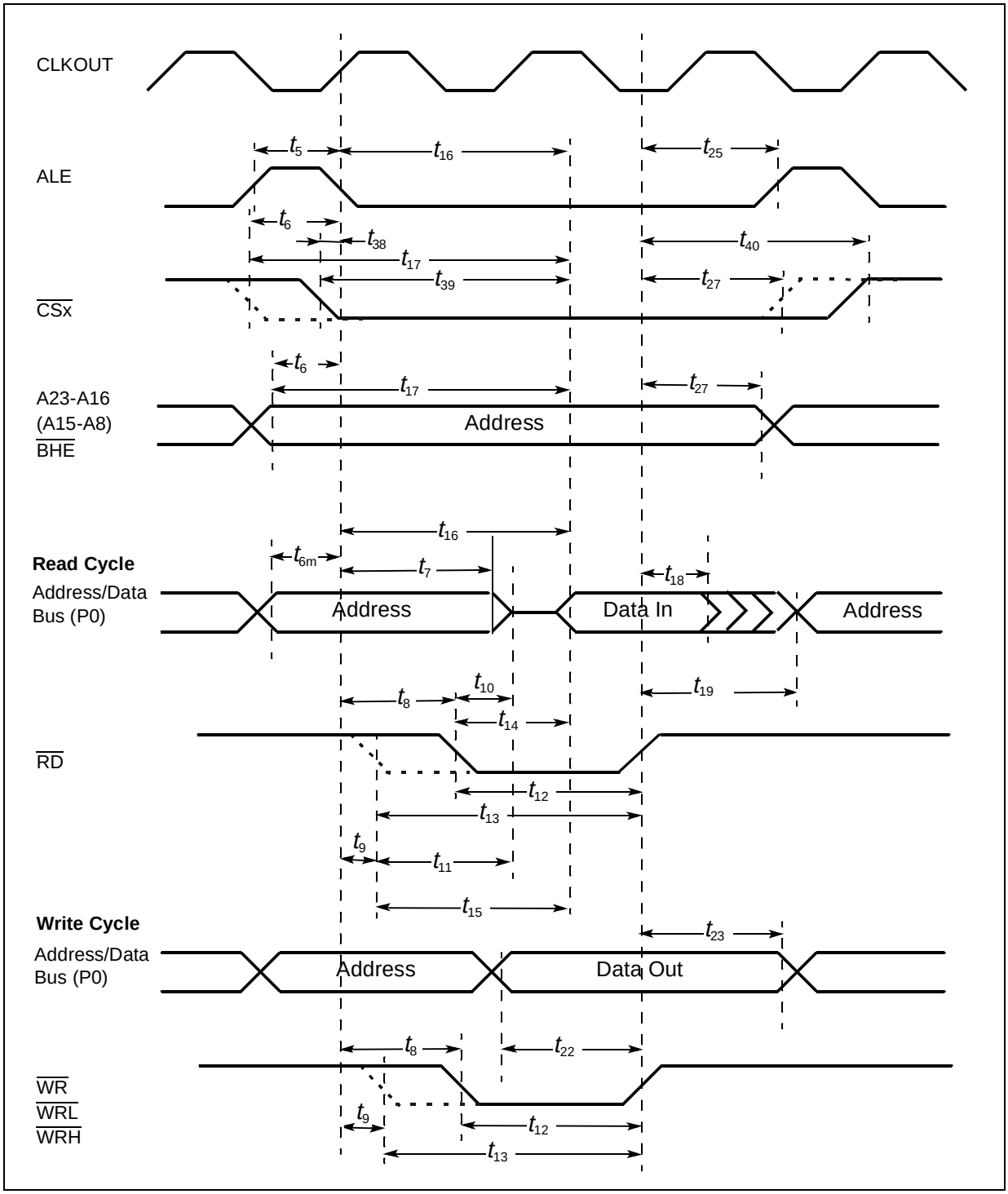


Figure 81 : External Memory Cycle: Multiplexed Mus, With / Without Read / Write Delay, Extended ALE

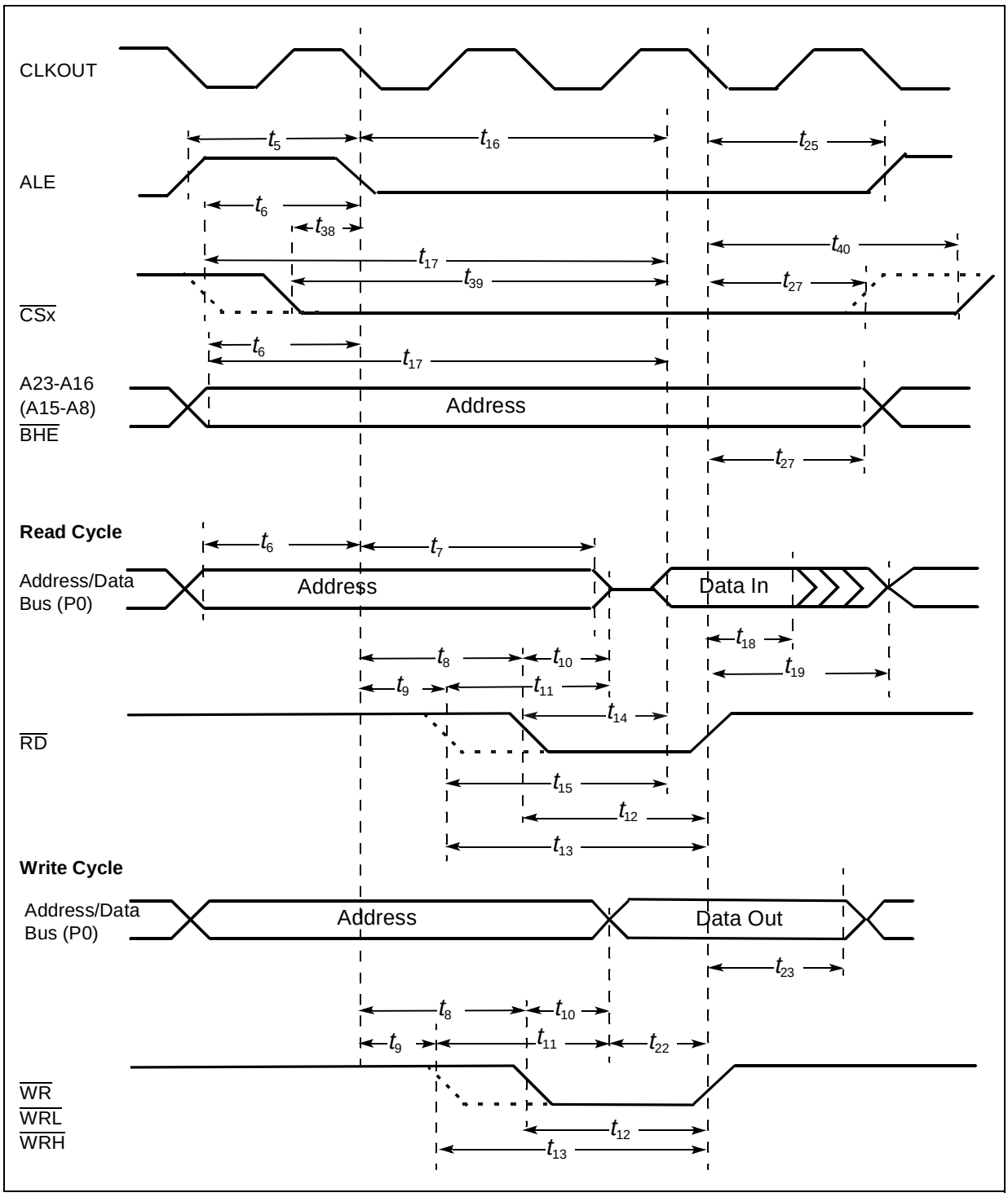


Figure 82 : External Memory Cycle: Multiplexed Bus, With / Without Read / Write Delay, Normal ALE, Read / Write Chip Select

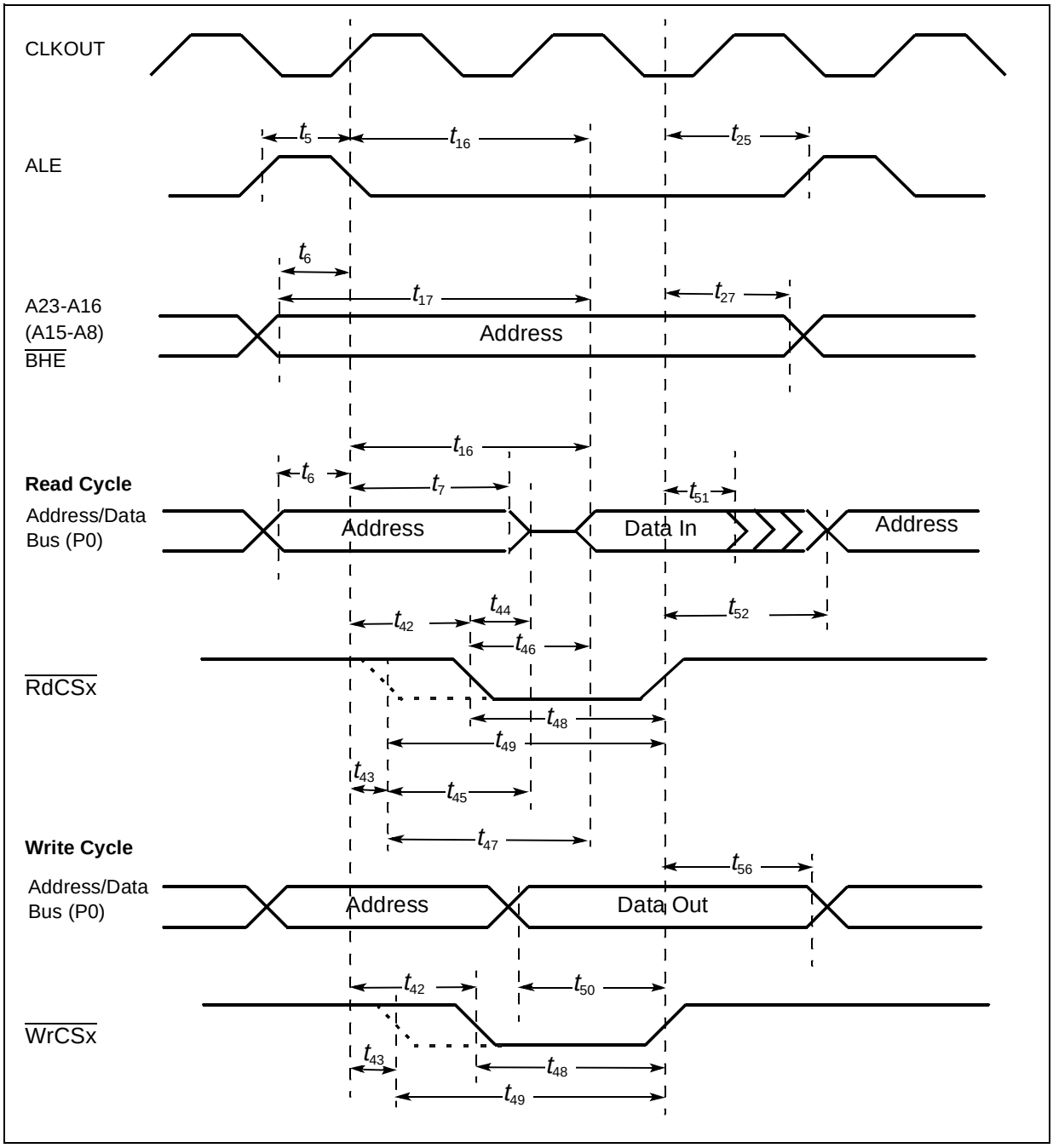
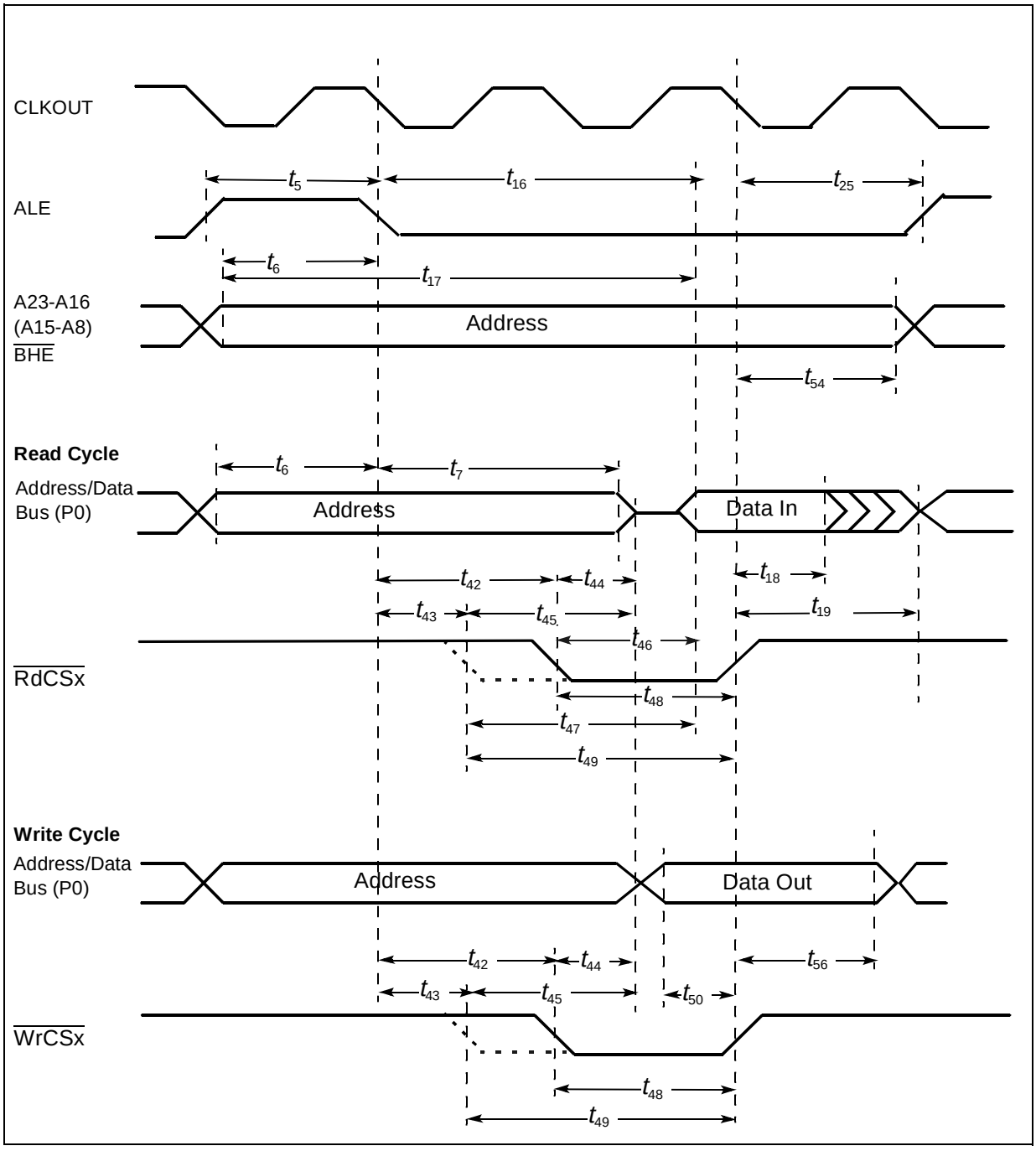


Figure 83 : External Memory Cycle: Multiplexed Bus, With / Without Read / Write Delay, Extended ALE, Read / Write Chip Select



20.4.11 - Demultiplexed Bus

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

Table 42 : Demultiplexed Bus Characteristics

Symbol		Parameter	Maximum CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		Unit
			Minimum	Maximum	Minimum	Maximum	
t_5	CC	ALE high time	4 + t_A	–	TCL - 8.5 + t_A	–	ns
t_6	CC	Address setup to ALE	2 + t_A	–	TCL - 10.5 + t_A	–	ns
t_{80}	CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	16.5 + $2t_A$	–	2 TCL - 8.5 + $2t_A$	–	ns
t_{81}	CC	Address/Unlatched $\overline{CS}$ setup to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	4 + $2t_A$	–	TCL - 8.5 + $2t_A$	–	ns
t_{12}	CC	$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	15.5 + t_C	–	2 TCL - 9.5 + t_C	–	ns
t_{13}	CC	$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	28 + t_C	–	3 TCL - 9.5 + t_C	–	ns
t_{14}	SR	$\overline{RD}$ to valid data in (with RW-delay)	–	6 + t_C	–	2 TCL - 19 + t_C	ns
t_{15}	SR	$\overline{RD}$ to valid data in (no RW-delay)	–	18.5 + t_C	–	3 TCL - 19 + t_C	ns
t_{16}	SR	ALE low to valid data in	–	18.5 + t_A + t_C	–	3 TCL - 19 + t_A + t_C	ns
t_{17}	SR	Address/Unlatched $\overline{CS}$ to valid data in	–	22 + $2t_A$ + t_C	–	4 TCL - 28 + $2t_A$ + t_C	ns
t_{18}	SR	Data hold after $\overline{RD}$ rising edge	0	–	0	–	ns
t_{20}	SR	Data float after $\overline{RD}$ rising edge (with RW-delay) [1 3]	–	16.5 + t_F	–	2 TCL - 8.5 + t_F + $2t_A$ [1]	ns
t_{21}	SR	Data float after $\overline{RD}$ rising edge (no RW-delay) [1 3]	–	4 + t_F	–	TCL - 8.5 + t_F + $2t_A$ [1]	ns
t_{22}	CC	Data valid to $\overline{WR}$	10 + t_C	–	2 TCL - 15 + t_C	–	ns
t_{24}	CC	Data hold after $\overline{WR}$	4 + t_F	–	TCL - 8.5 + t_F	–	ns
t_{26}	CC	ALE rising edge after $\overline{RD}$, $\overline{WR}$	-10 + t_F	–	-10 + t_F	–	ns
t_{28}	CC	Address/Unlatched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$ [2]	0 (no t_F) -5 + t_F (t_F > 0)	–	0 (no t_F) -5 + t_F (t_F > 0)	–	ns
t_{28h}	CC	Address/Unlatched $\overline{CS}$ hold after $\overline{WRH}$	-5 + t_F	–	-5 + t_F	–	ns
t_{38}	CC	ALE falling edge to Latched $\overline{CS}$	-4 - t_A	6 - t_A	-4 - t_A	6 - t_A	ns
t_{39}	SR	Latched $\overline{CS}$ low to Valid Data In	–	18.5 + t_C + $2t_A$	–	3 TCL - 19 + t_C + $2t_A$	ns

Table 42 : Demultiplexed Bus Characteristics

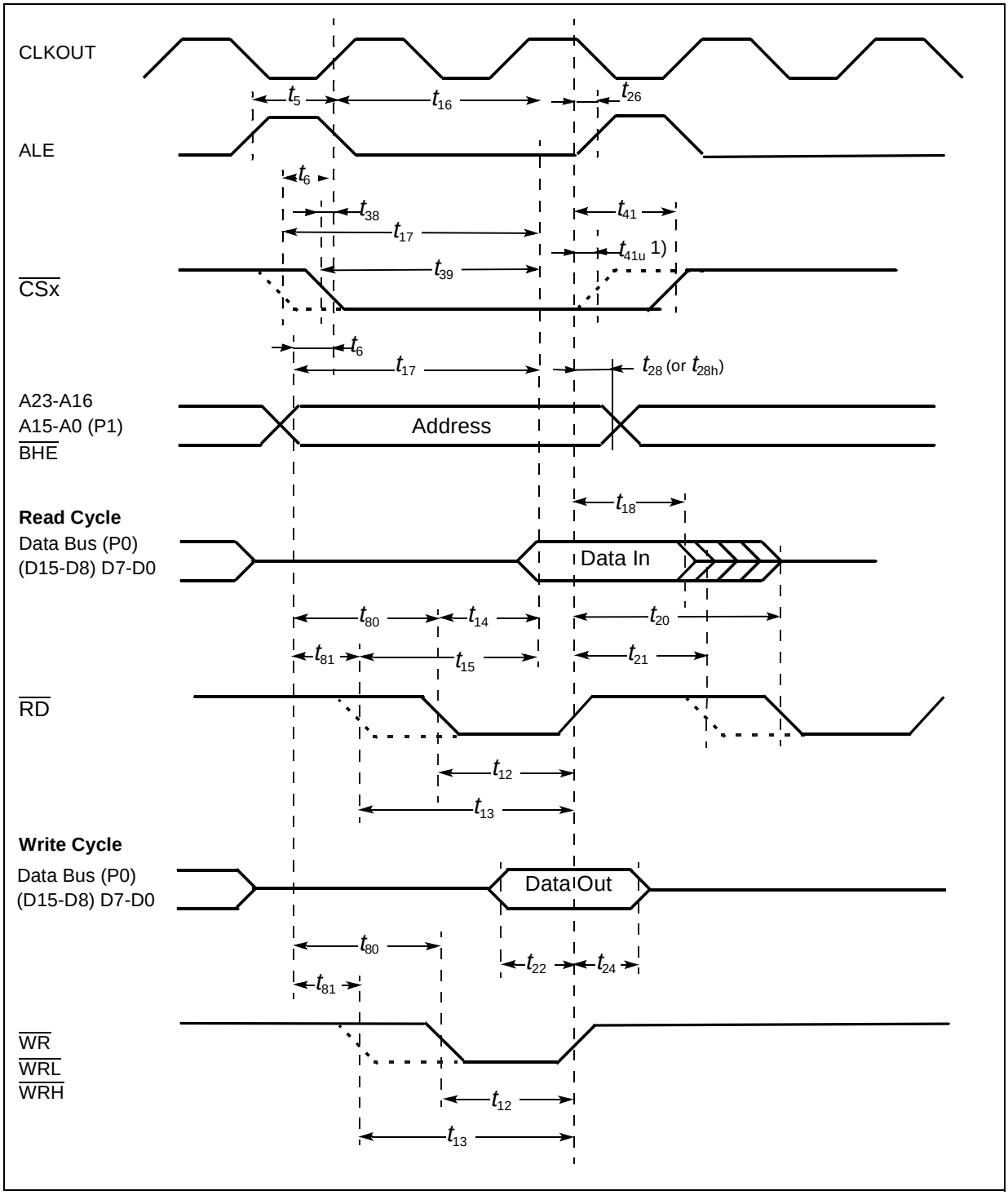
Symbol		Parameter	Maximum CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{41}	CC	Latched $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$	$2 + t_F$	–	$TCL - 10.5 + t_F$	–	ns
t_{82}	CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay)	$14.5 + 2t_A$	–	$2\ TCL - 10.5 + 2t_A$	–	ns
t_{83}	CC	Address setup to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay)	$2 + 2t_A$	–	$TCL - 10.5 + 2t_A$	–	ns
t_{46}	SR	$\overline{RdCS}$ to Valid Data In (with RW-delay)	–	$4 + t_C$	–	$2\ TCL - 21 + t_C$	ns
t_{47}	SR	$\overline{RdCS}$ to Valid Data In (no RW-delay)	–	$16.5 + t_C$	–	$3\ TCL - 21 + t_C$	ns
t_{48}	CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay)	$15.5 + t_C$	–	$2\ TCL - 9.5 + t_C$	–	ns
t_{49}	CC	$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay)	$28 + t_C$	–	$3\ TCL - 9.5 + t_C$	–	ns
t_{50}	CC	Data valid to $\overline{WrCS}$	$10 + t_C$	–	$2\ TCL - 15 + t_C$	–	ns
t_{51}	SR	Data hold after $\overline{RdCS}$	0	–	0	–	ns
t_{53}	SR	Data float after $\overline{RdCS}$ (with RW-delay) [3]	–	$16.5 + t_F$	–	$2\ TCL - 8.5 + t_F$	ns
t_{68}	SR	Data float after $\overline{RdCS}$ (no RW-delay) [3]	–	$4 + t_F$	–	$TCL - 8.5 + t_F$	ns
t_{55}	CC	Address hold after $\overline{RdCS}$, $\overline{WrCS}$	$-8.5 + t_F$	–	$-8.5 + t_F$	–	ns
t_{57}	CC	Data hold after $\overline{WrCS}$	$2 + t_F$	–	$TCL - 10.5 + t_F$	–	ns

Notes: 1. RW-delay and t_A refer to the next following bus cycle.

2. Read data are latched with the same clock edge that triggers the address change and the rising $\overline{RD}$ edge. Therefore address changes before the end of $\overline{RD}$ have no impact on read cycles.

3. Partially tested, guaranteed by design characterization.

Figure 84 : External Memory Cycle: Demultiplexed Bus, With / Without Read / Write Delay, Normal ALE



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

Figure 85 : External Memory Cycle: Demultiplexed Bus, With / Without Read / Write Delay, Extended ALE

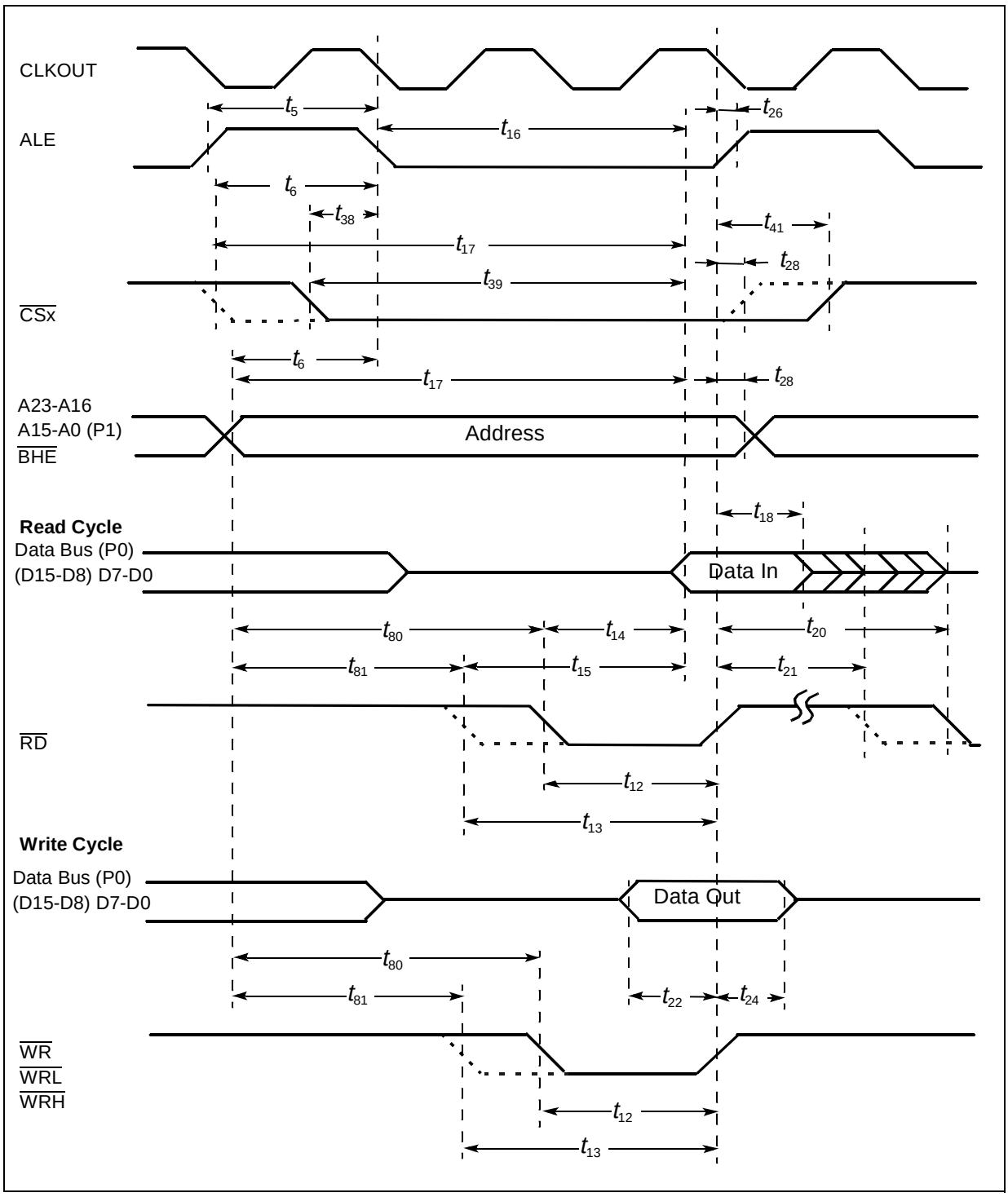


Figure 86 : External Memory Cycle: Demultiplexed Bus, With / Without Read / Write Delay, Normal ALE, Read / Write Chip Select

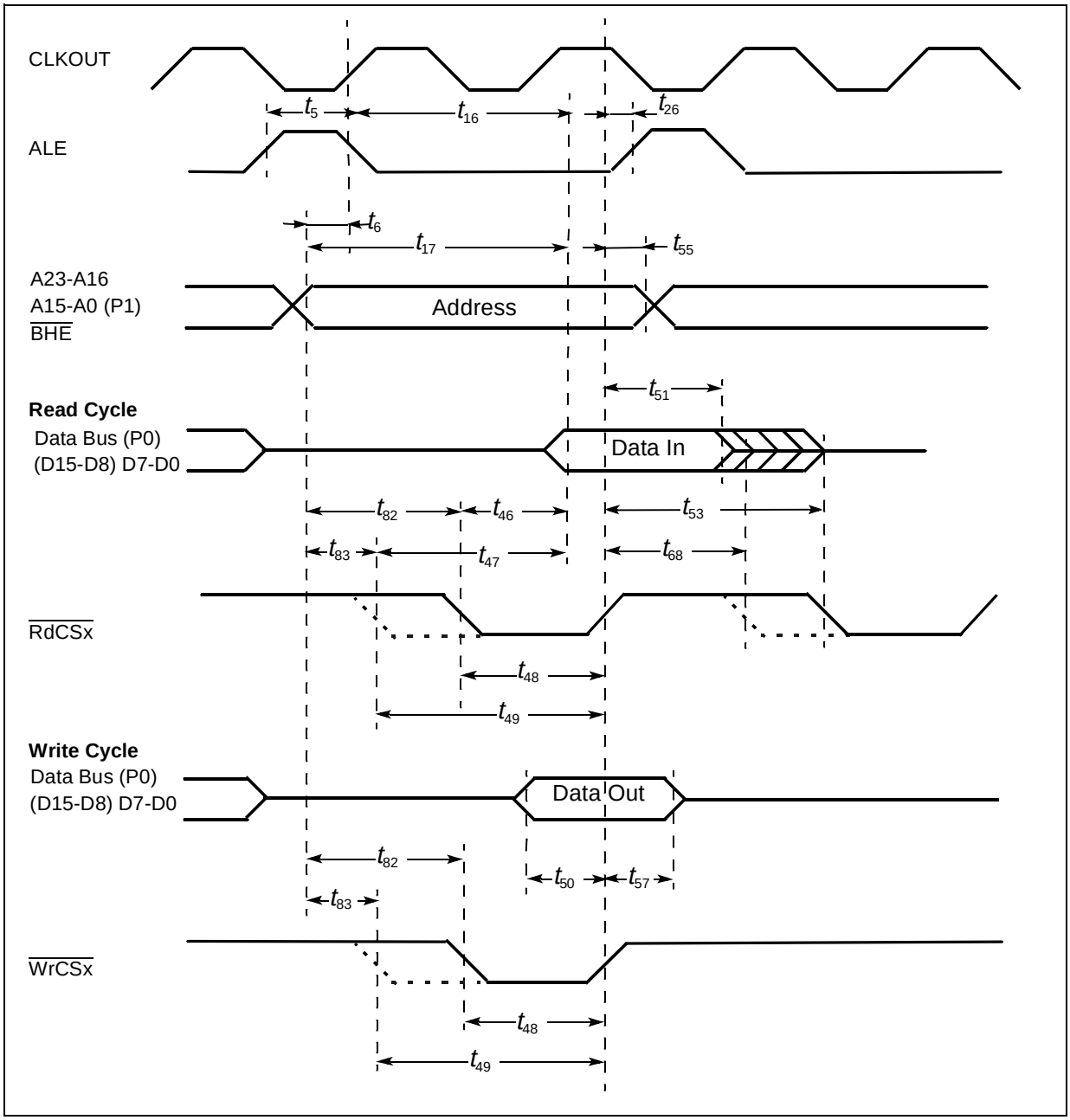
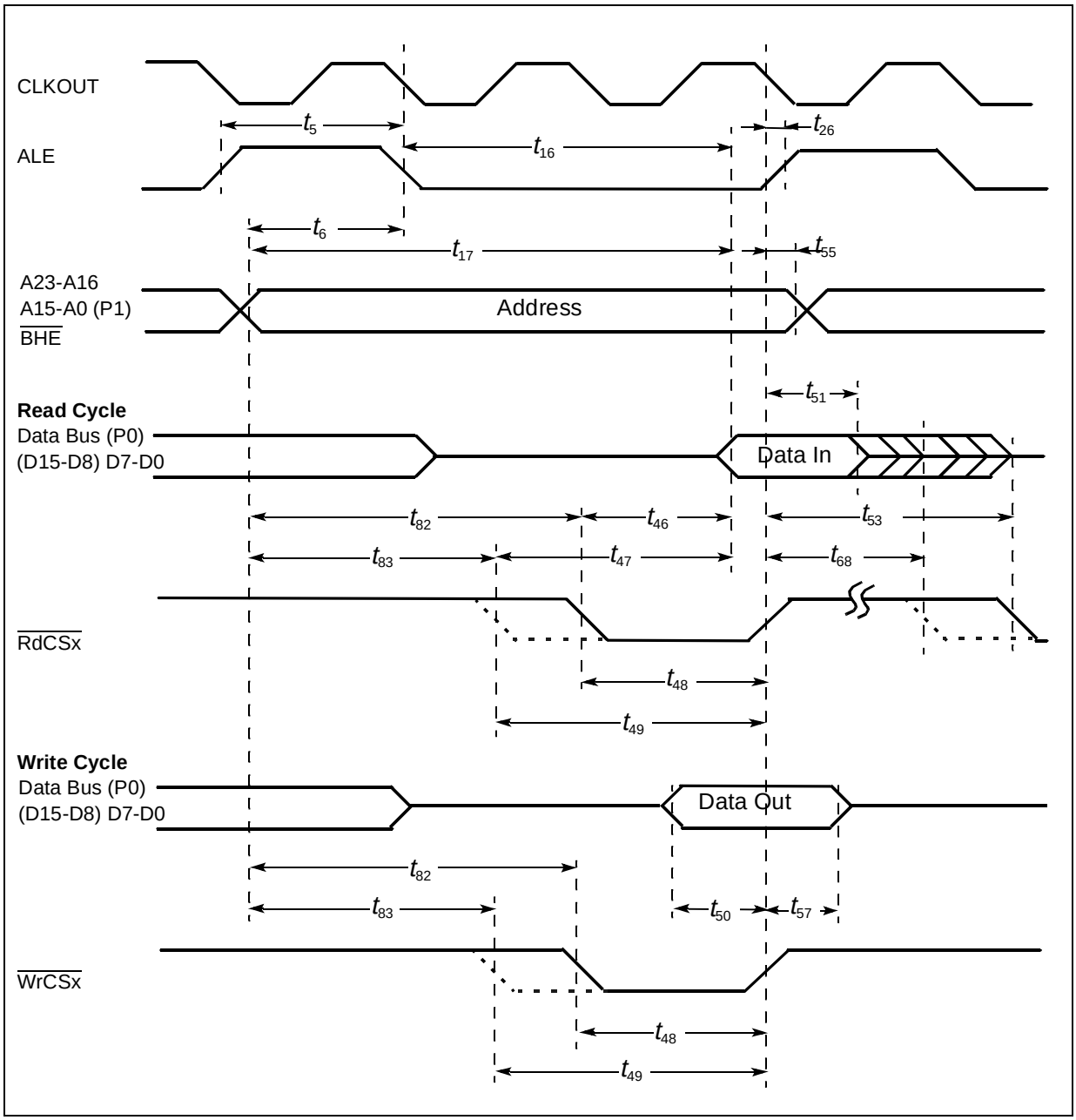


Figure 87 : External Memory Cycle: Demultiplexed Bus, no Read / Write Delay, Extended ALE, Read / Write Chip Select



20.4.12 - CLKOUT and $\overline{READY}$

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

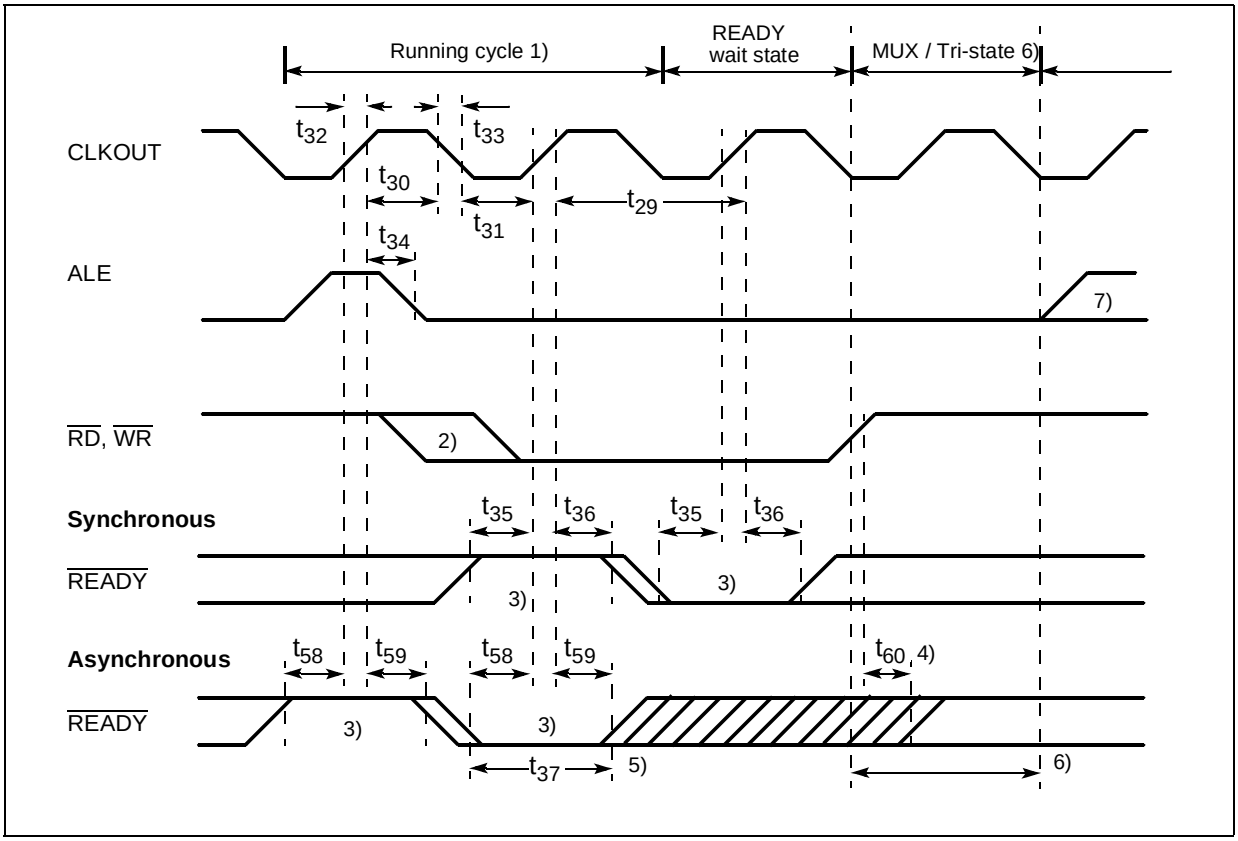
Table 43 : CLKOUT and $\overline{READY}$ Characteristics

Symbol		Parameter	Maximum CPU Clock = 40MHz		Variable CPU Clock 1/2 TCL = 1 to 40MHz		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{29}	CC	CLKOUT cycle time	25	25	2 TCL	2TCL	ns
t_{30}	CC	CLKOUT high time	4	–	TCL – 8.5	–	ns
t_{31}	CC	CLKOUT low time	3	–	TCL – 9.5	–	ns
t_{32}	CC	CLKOUT rise time	–	4	–	4	ns
t_{33}	CC	CLKOUT fall time	–	4	–	4	ns
t_{34}	CC	CLKOUT rising edge to ALE falling edge	$-2 + t_A$	$8 + t_A$	$-2 + t_A$	$8 + t_A$	ns
t_{35}	SR	Synchronous $\overline{READY}$ setup time to CLKOUT	12.5	–	12.5	–	ns
t_{36}	SR	Synchronous $\overline{READY}$ hold time after CLKOUT	2	–	2	–	ns
t_{37}	SR	Asynchronous $\overline{READY}$ low time	35	–	2 TCL + 10	–	ns
t_{58}	SR	Asynchronous $\overline{READY}$ setup time [1]	12.5	–	12.5	–	ns
t_{59}	SR	Asynchronous $\overline{READY}$ hold time [1]	2	–	2	–	ns
t_{60}	SR	Async. $\overline{READY}$ hold time after $\overline{RD}$, $\overline{WR}$ high (Demultiplexed Bus) [2]	0	$0 + 2t_A + t_C + t_F$ [2]	0	TCL - 12.5 $+ 2t_A + t_C + t_F$ [2]	ns

Notes: 1. These timings are given for test purposes only, in order to assure recognition at a specific clock edge.

2. Demultiplexed bus is the worst case. For multiplexed bus 2 TCL are to be added to the maximum values. This adds even more time for deactivating READY.
The $2t_A$ and t_C refer to the next following bus cycle, t_F refers to the current bus cycle.

Figure 88 : CLKOUT and $\overline{\text{READY}}$



Notes: 1. Cycle as programmed, including MCTC wait states (Example shows 0 MCTC WS).

2. The leading edge of the respective command depends on RW-delay.

3. $\overline{\text{READY}}$ sampled HIGH at this sampling point generates a READY controlled wait state, $\overline{\text{READY}}$ sampled LOW at this sampling point terminates the currently running bus cycle.

4. $\overline{\text{READY}}$ may be deactivated in response to the trailing (rising) edge of the corresponding command ($\overline{\text{RD}}$ or $\overline{\text{WR}}$).

5. If the Asynchronous $\overline{\text{READY}}$ signal does not fulfill the indicated setup and hold times with respect to CLKOUT (e.g. because CLKOUT is not enabled), it must fulfill t_{37} in order to be safely synchronized. This is guaranteed, if READY is removed in response to the command (see Note 4)).

6. Multiplexed bus modes have a MUX wait state added after a bus cycle, and an additional MTTC wait state may be inserted here. For a multiplexed bus with MTTC wait state this delay is 2 CLKOUT cycles, for a demultiplexed bus without MTTC wait state this delay is zero.

7. The next external bus cycle may start here.

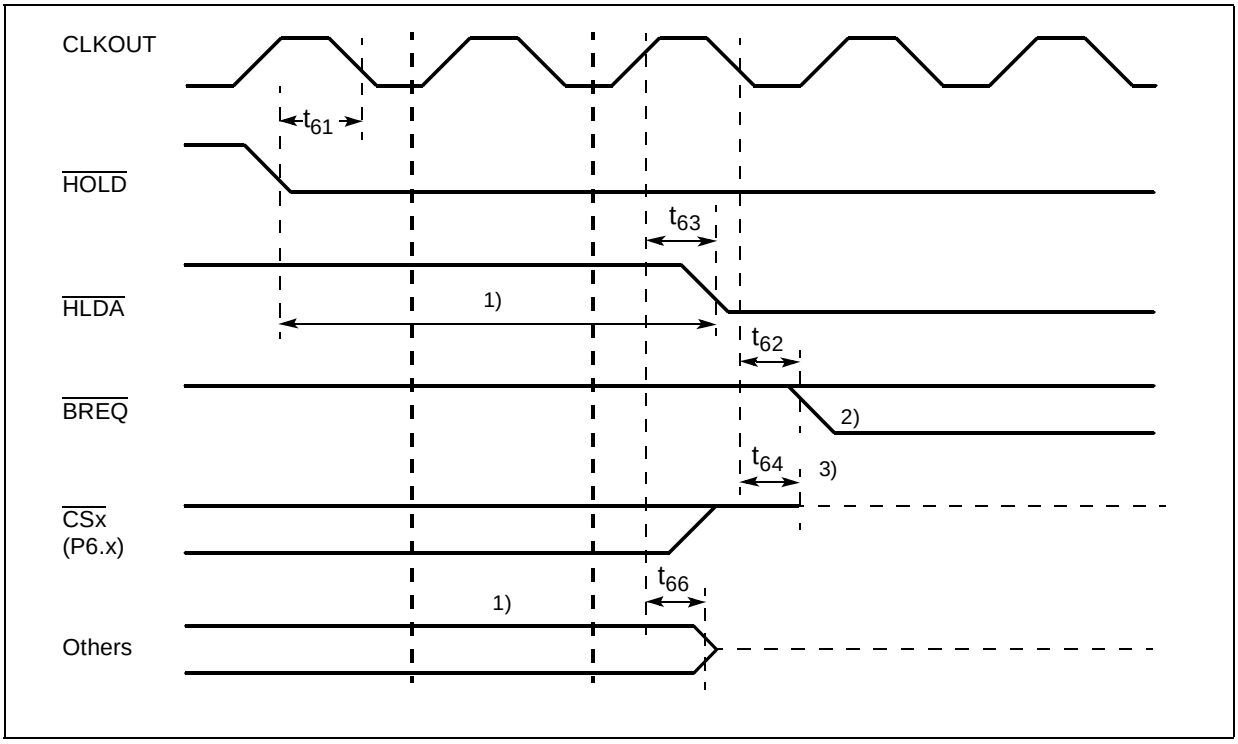
20.4.13 - External Bus Arbitration

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

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

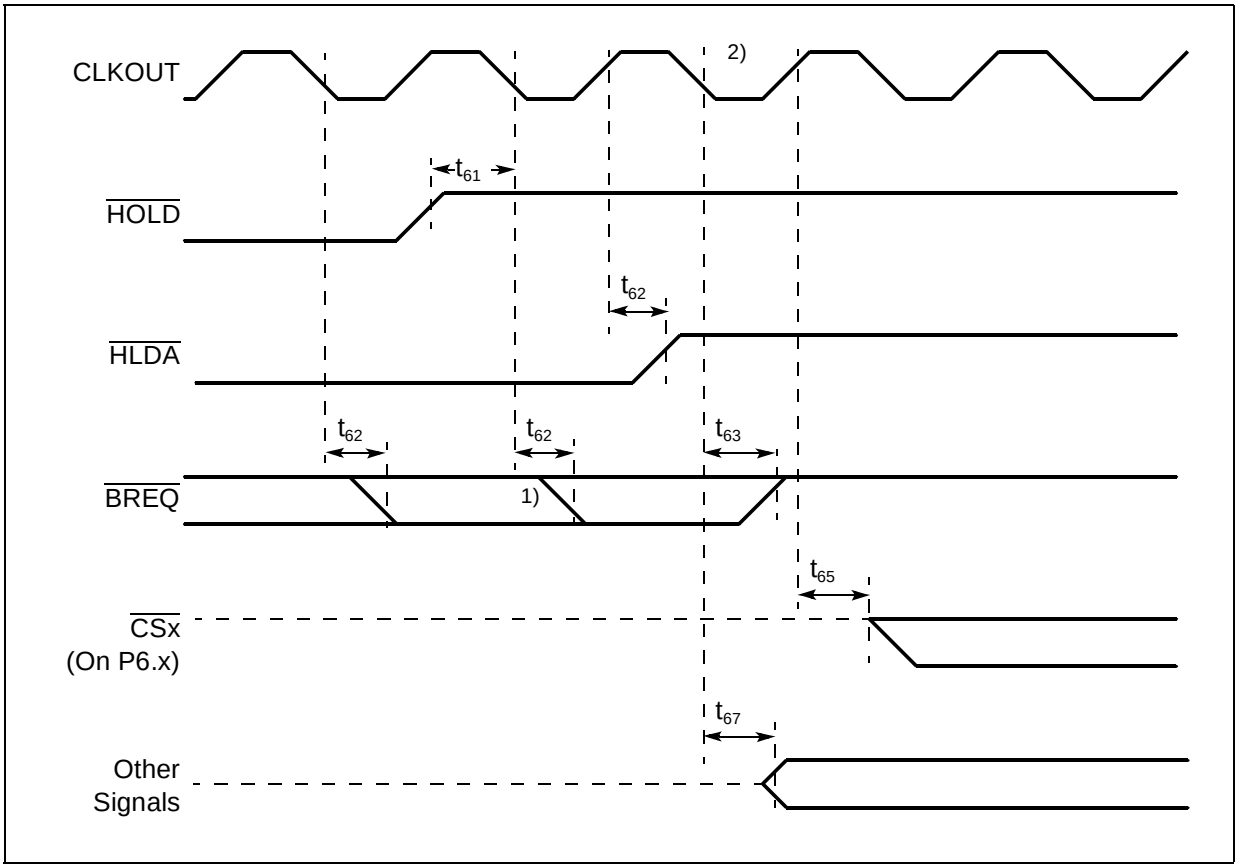
Note: 1. Partially tested, guaranteed by design characterization.

Figure 89 : External Bus Arbitration, Releasing the Bus



Notes: 1. The ST10F280 will complete the currently running bus cycle before granting bus access.
2. This is the first possibility for $\overline{BREQ}$ to become active.
3. The $\overline{CS}$ outputs will be resistive high (pull-up) after t_{64}.

Figure 90 : External Bus Arbitration, (regaining the bus)



Notes: 1. This is the last chance for BREQ to trigger the indicated regain-sequence. Even if BREQ is activated earlier, the regain-sequence is initiated by HOLD going high. Please note that HOLD may also be disactivated without the ST10F280 requesting the bus.

2. The next ST10F280 driven bus cycle may start here.

20.4.14 - High-Speed Synchronous Serial Interface (SSC) Timing

20.4.14.1 Master Mode

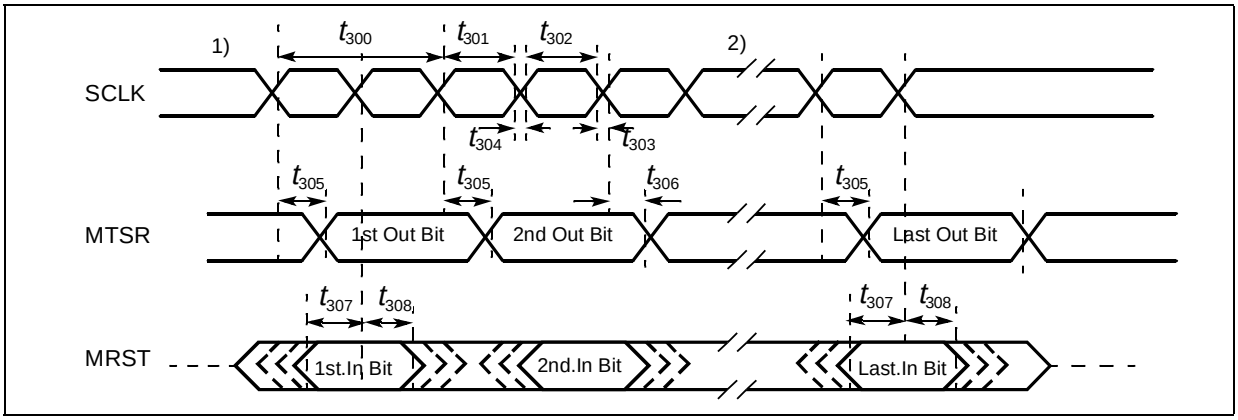
V_{CC} = 5V ±10%, V_{SS} = 0V, CPU clock = 40MHz, T_A = -40 to +125°C, C_L = 50pF

Symbol		Parameter	Maximum Baud rate = 10M Baud (<SSCBR> = 0001h)		Variable Baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{300}	**CC**	SSC clock cycle time	100	100	8 TCL	262144 TCL	ns
t_{301}	**CC**	SSC clock high time	40	–	$t_{300}/2$ - 10	–	ns
t_{302}	**CC**	SSC clock low time	40	–	$t_{300}/2$ - 10	–	ns
t_{303}	**CC**	SSC clock rise time	–	10	–	10	ns
t_{304}	**CC**	SSC clock fall time	–	10	–	10	ns
t_{305}	**CC**	Write data valid after shift edge	–	15	–	15	ns
t_{306}	**CC**	Write data hold after shift edge [1]	-2	–	-2	–	ns
t_{307p}	**SR**	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	37.5	–	2 TCL + 12.5	–	ns
t_{308p}	**SR**	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	50	–	4 TCL	–	ns
t_{307}	**SR**	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	25	–	2 TCL	–	ns
t_{308}	**SR**	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	0	–	0	–	ns

Note: 1. Timing guaranteed by design.

The formula for SSC Clock Cycle time is : t_{300} = 4 TCL * (<SSCBR> + 1)
Where <SSCBR> represents the content of the SSC Baud rate register, taken as unsigned 16-bit integer.

Figure 91 : SSC Master Timing



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

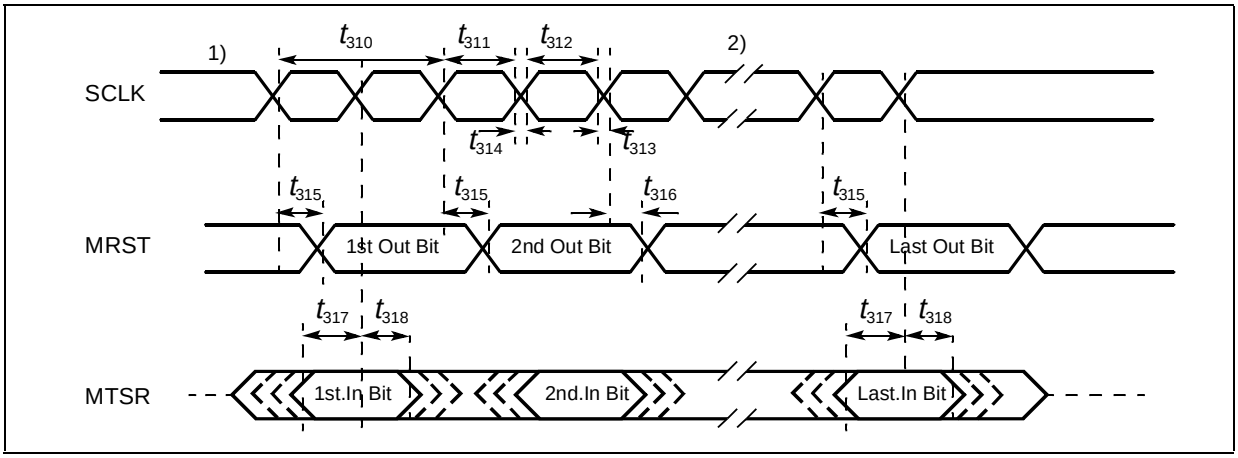
20.4.14.2 Slave mode

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

Symbol		Parameter	Maximum Baud rate=10MBd (<SSCBR> = 0001h)		Variable Baud rate (<SSCBR>=0001h-FFFFh)		Unit
			Minimum	Maximum	Minimum	Maximum	
t_{310}	**SR**	SSC clock cycle time	100	100	8 TCL	262144 TCL	ns
t_{311}	**SR**	SSC clock high time	40	–	t_{310}/2 - 10	–	ns
t_{312}	**SR**	SSC clock low time	40	–	t_{310}/2 - 10	–	ns
t_{313}	**SR**	SSC clock rise time	–	10	–	10	ns
t_{314}	**SR**	SSC clock fall time	–	10	–	10	ns
t_{315}	**CC**	Write data valid after shift edge	–	39	–	2 TCL + 14	ns
t_{316}	**CC**	Write data hold after shift edge	0	–	0	–	ns
t_{317p}	**SR**	Read data setup time before latch edge, phase error detection on (SSCPEN = 1)	62	–	4 TCL + 12	–	ns
t_{318p}[1]	**SR**	Read data hold time after latch edge, phase error detection on (SSCPEN = 1)	87	–	6 TCL + 12	–	ns
t_{317}	**SR**	Read data setup time before latch edge, phase error detection off (SSCPEN = 0)	6	–	6	–	ns
t_{318}	**SR**	Read data hold time after latch edge, phase error detection off (SSCPEN = 0)	31	–	2 TCL + 6	–	ns

The formula for SSC Clock Cycle time is: t_{310} = 4 TCL * (<SSCBR> + 1)
Where <SSCBR> represents the content of the SSC Baud rate register, taken as unsigned 16-bit integer.

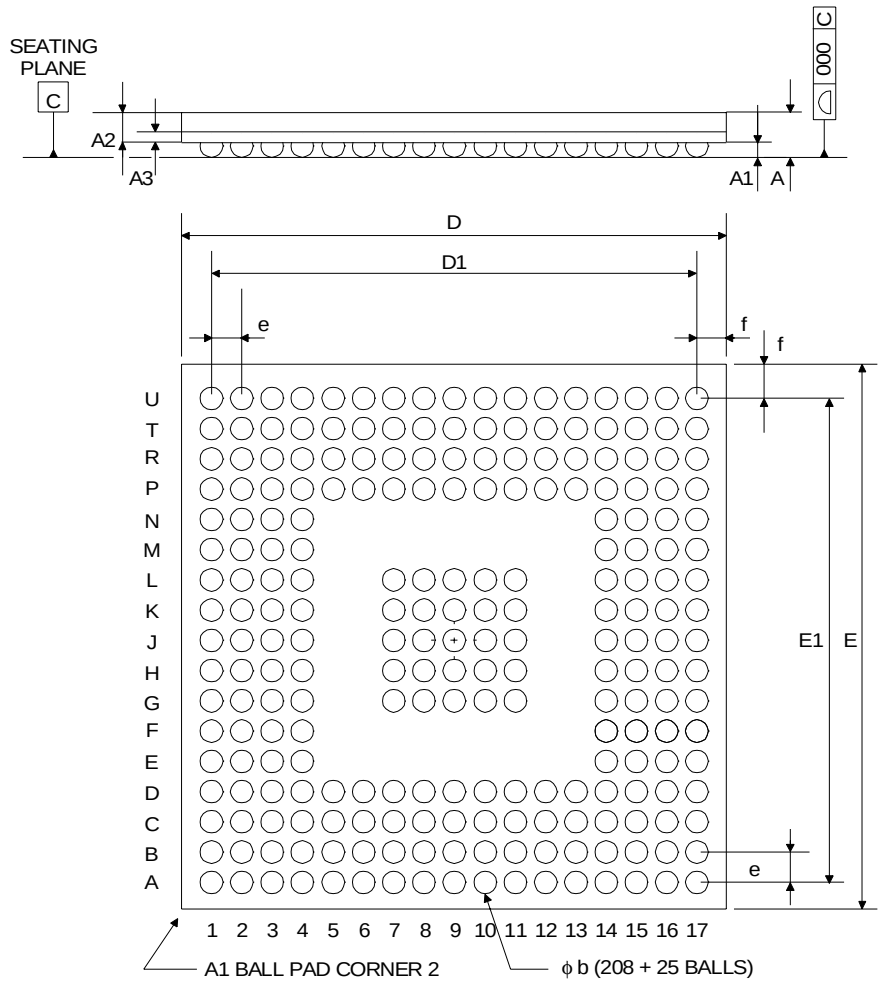
Figure 92 : SSC Slave Timing



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

21 - PACKAGE MECHANICAL DATA

Figure 93 : Package Outline PBGA 208 (23 x 23 x 1.96 mm)



Dimensions	Millimeters			Inches (approx)		
	Minimum	Typical	Maximum	Minimum	Typical	Maximum
A		1.960			0.077	
A1	0.500	0.600	0.700	0.019	0.024	0.028
A2		1.360			0.054	
A3		0.560			0.022	
ϕb	0.600	0.760	0.900	0.024	0.030	0.035
D	22.900	23.000	23.100	0.902	0.906	0.909
D1		20.320			0.800	
E	22.900	23.000	23.100	0.902	0.906	0.909
E1		20.320			0.800	
e		1.270			0.50	
f	1.240	1.340	1.440	0.049	0.053	0.057
aaa			0.150			0.006

Notes: 1. PBGA stands for Plastic Ball Grid Array.

2. The terminal A1 corner must be identified on the top surface of the package by using a corner chamfer, ink or metalized markings, identation or other feature of package body or integral heastslug. A distinguishing feature is allowable on the bottom of the package to identify the terminal A1 corner. Exact shape and size of this feature is optional.

22 - ORDERING INFORMATION

Salestype	Temperature range	Package
ST10F280-JT3	-40°C to +125°C	PBGA 208 (23 x 23 x 1.96 mm)