**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 16MHz |
| Connectivity | CANbus, EBI/EMI, UART/USART |
| Peripherals | DMA, POR, PWM, WDT |
| Number of I/O | 48 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-BQFP |
| Supplier Device Package | 80-PQFP (14x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/p80ce598ffb-00-518 |

## 5 PINNING

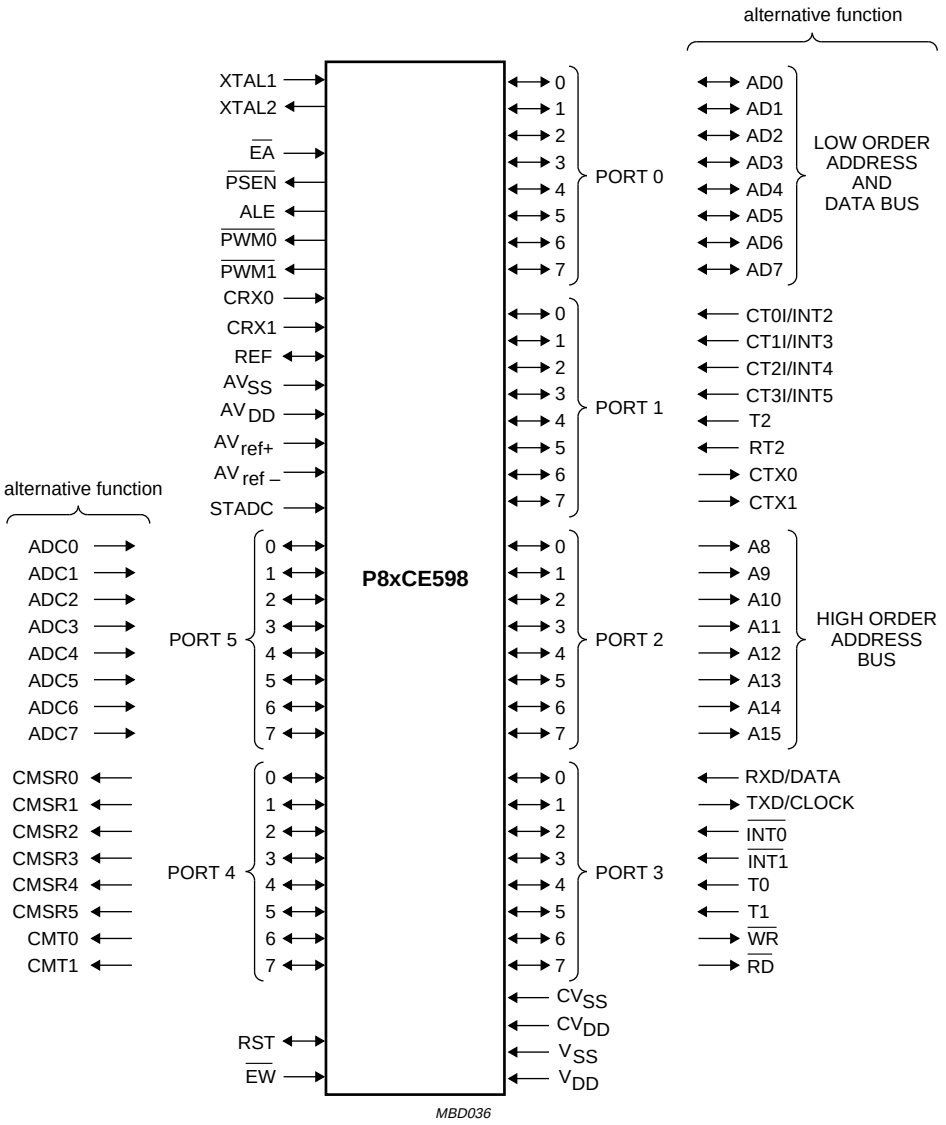



Fig.2 Pin functions.

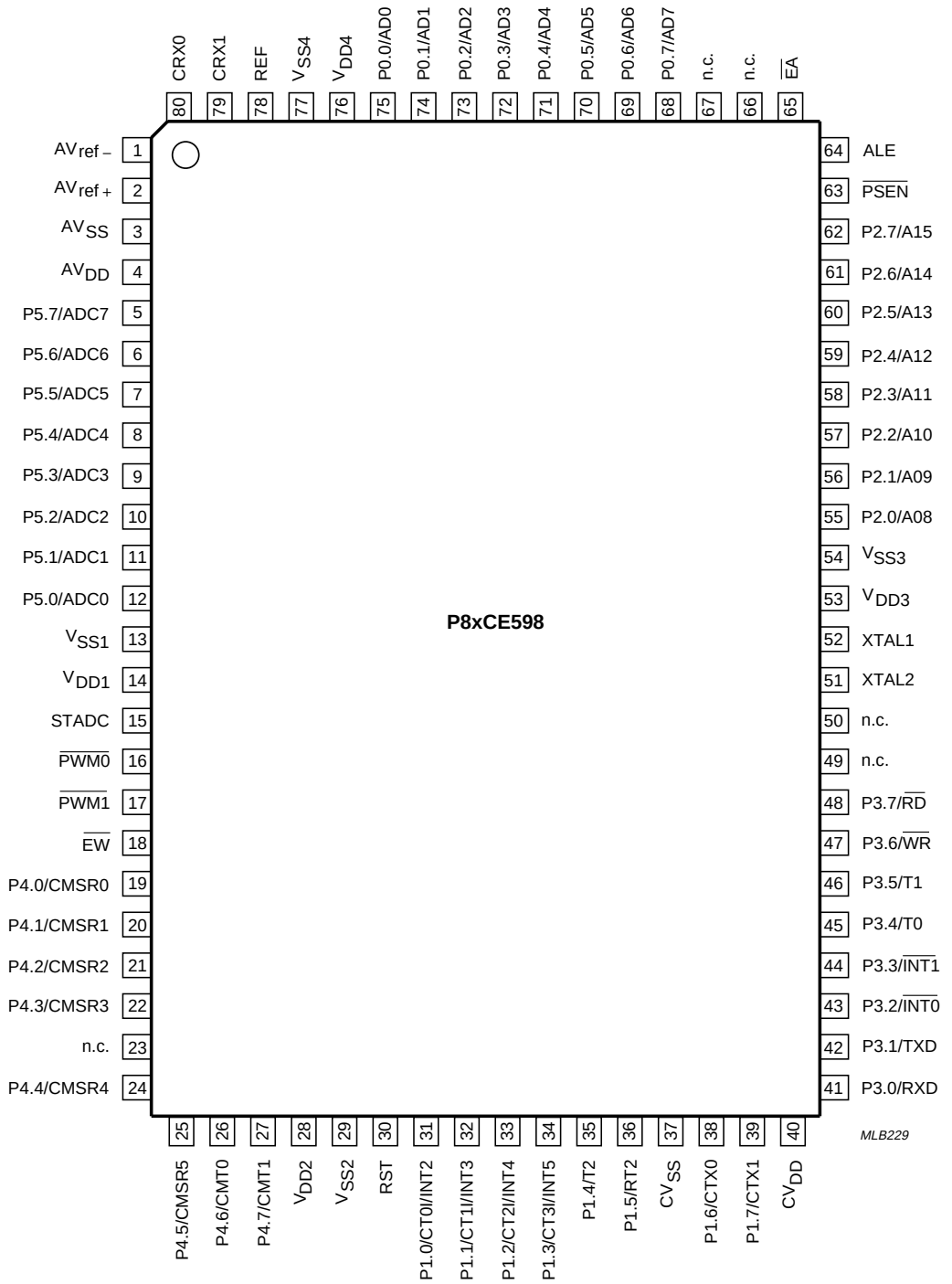



Fig.3 Pin configuration QFP80/SOT318-1.

| SYMBOL | | PIN | DESCRIPTION |
|---|---|---|---|
| **DEFAULT** | **ALTERNATIVE** | | |
| **Port 3** | | | |
| P3.0 to P3.7 | | 41 to 48 | **8-bit quasi-bidirectional I/O port**. |
| | RXD | 41 | **Serial Input Port**. |
| | TXD | 42 | **Serial Output Port**. |
| | $\overline{\text{INT0}}$ | 43 | **External interrupt input 0.** |
| | $\overline{\text{INT1}}$ | 44 | **External interrupt input 1.** |
| | T0 | 45 | **Timer 0 external input**. |
| | T1 | 46 | **Timer 1 external input**. |
| | $\overline{\text{WR}}$ | 47 | **External Data Memory Write strobe**. |
| | $\overline{\text{RD}}$ | 48 | **External Data Memory Read strobe**. |
| **Port 2 (Sink/source: 1 × TTL = 4 × LSTTL inputs)** | | | |
| P2.0 to P2.7 | | 55 to 62 | **8-bit quasi-bidirectional I/O port**. |
| | A08 to A15 | | **High-order address byte for external memory**. |
| **Port 0 (Sink/source: 8 × LSTTL inputs)** | | | |
| P0.7 to P0.0 | | 68 to 75 | **8-bit open drain bidirectional I/O port**. |
| | AD7 to AD0 | | **Multiplexed Low-order address and Data bus for external memory**. |
| **Port 5** | | | |
| P5.7 to P5.0 | | 5 to 12 | **8-bit input port**. |
| | ADC7 to ADC0 | | **8 input channels to ADC**. |

**Notes**

1. To avoid a ‘latch up’ effect at power-on: $V_{SS} - 0.5\ V <$ ‘voltage on any pin at any time’ $< V_{DD} + 0.5\ V$.
2. If the CAN-controller is in the reset state (e.g. after a power-up reset; CAN Control Register bit CR.0; see Section 13.5.3 Table 32, the CAN transmitter outputs are floating and the pins P1.6 and P1.7 can be used as open-drain port pins. After a power-up reset the port data is HIGH, leaving the pins P1.6 and P1.7 floating.

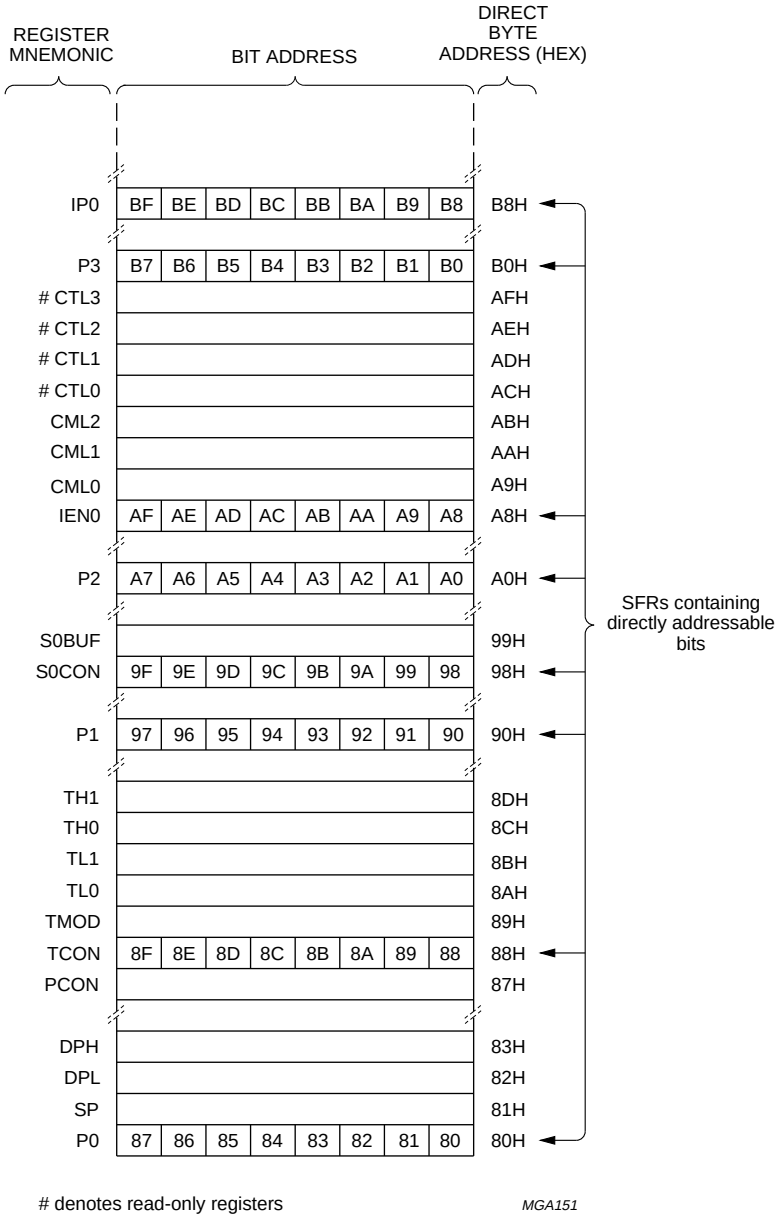



Fig.7 Special Function Register memory map (b).

## 11 TIMERS/COUNTERS

The P8xCE598 contains:

- Three 16-bit timer/event counters: Timer 0, Timer 1 and Timer 2
- One 8-bit timer, T3 (Watchdog WDT).

### 11.1 Timer 0 and Timer 1

Timer 0 and Timer 1 may be programmed to carry out the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupt requests.

Timer 0 and Timer 1 can be programmed independently to operate in 3 modes:

Mode 0 8-bit timer or 8-bit counter each with divide-by-32 prescaler.

Mode 1 16-bit timer-interval or event counter.

Mode 2 8-bit timer-interval or event counter with automatic reload upon overflow.

Timer 0 can be programmed to operate in an additional mode as follows:

Mode 3 one 8-bit time-interval or event counter and one 8-bit timer-interval counter.

When Timer 0 is in Mode 3, Timer 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt flag or generate an interrupt. However, the overflow from Timer 1 can be used to pulse the Serial Port baud-rate generator.

The frequency handling range of these counters with a 16 MHz crystal is as follows:

- In the timer function, the timer is incremented at a frequency of 1.33 MHz ($1/_{12}$ of the oscillator frequency)
- 0 Hz to an upper limit of 0.66 MHz ($1/_{24}$ of the oscillator frequency) when programmed for external inputs.

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations. When configured as a counter, the register is incremented on every falling edge on the corresponding input pin, T0 or T1.

The earliest moment, when the incremented register value can be read is during the second machine cycle following the machine cycle within which the incrementing pulse occurred.The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all HIGHs to all LOWs (or automatic reload value), with the exception of Mode 3 as previously described.

### 11.2 Timer T2 Capture and Compare Logic

Timer T2 is a 16-bit timer/counter which has capture and compare facilities (see Fig.11).

The 16-bit timer/counter is clocked via a prescaler with a programmable division factor of 1, 2, 4 or 8. The input of the prescaler is clocked with $1/_{12}$ of the oscillator frequency, or by an external source connected to the T2 input, or it is switched off. The maximum repetition rate of the external clock source is $1/_{12}f_{CLK}$, twice that of Timer 0 and Timer 1. The prescaler is incremented on a rising edge. It is cleared if its division factor or its input source is changed, or if the timer/counter is reset.

T2 is readable ‘on the fly’, without any extra read latches; this means that software precautions have to be taken against misinterpretation at overflow from least to most significant byte while T2 is being read. T2 is not loadable and is reset by the RST signal or at the positive edge of the input signal RT2, if enabled. In the Idle mode the timer/counter and prescaler are reset and halted.

T2 is connected to four 16-bit Capture Registers: CT0, CT1, CT2 and CT3. A rising or falling edge on the inputs CT0I, CT1I, CT2I or CT3I (alternative function of Port 1) results in loading the contents of T2 into the respective Capture Registers and an interrupt request.

Using the Capture Register CTCON, these inputs may invoke capture and interrupt request on a positive edge, a negative edge or on both edges. If neither a positive nor a negative edge is selected for capture input, no capture or interrupt request can be generated by this input.

The contents of the Compare Registers CM0, CM1 and CM2 are continually compared with the counter value of Timer T2. When a match occurs, an interrupt may be invoked. A match of CM0 sets the bits 0 to 5 of Port 4, a CM1 match resets these bits and a CM2 match toggles bits 6 and 7 of Port 4, provided these functions are enabled by the STE/RTE registers. A match of CM0 and CM1 at the same time results in resetting bits 0 to 5 of Port 4. CM0, CM1 and CM2 are reset by the RST signal.

Port 4 can be read and written by software without affecting the toggle, set and reset signals. At a byte overflow of the least significant byte, or at a 16-bit overflow of the timer/counter, an interrupt sharing the same interrupt vector is requested. Either one or both of these overflows can be programmed to request an interrupt. All interrupt flags must be reset by software.

**Table 30** CPU/CAN Register map

| BIT | | | | | | | |
|---|---|---|---|---|---|---|---|
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Control Segment** | | | | | | | |
| ADDRESS 0: CONTROL REGISTER | | | | | | | |
| TM | S | RA | OIE | EIE | TIE | RIE | RR |
| ADDRESS 1: COMMAND REGISTER | | | | | | | |
| RX0A | RX1A | WUM | SLP | COS | RRB | AT | TR |
| ADDRESS 2: STATUS REGISTER | | | | | | | |
| BS | ES | TS | RS | TCS | TBS | DO | RBS |
| ADDRESS 3: INTERRUPT REGISTER | | | | | | | |
| Reserved | Reserved | Reserved | WUI | OI | EI | TI | RI |
| ADDRESS 4: ACCEPTANCE CODE REGISTER | | | | | | | |
| AC.7 | AC.6 | AC.5 | AC.4 | AC.3 | AC.2 | AC.1 | AC.0 |
| ADDRESS 5: ACCEPTANCE MASK REGISTER | | | | | | | |
| AM.7 | AM.6 | AM.5 | AM.4 | AM.3 | AM.2 | AM.1 | AM.0 |
| ADDRESS 6: BUS TIMING REGISTER 0 | | | | | | | |
| SJW.1 | SJW.0 | BRP.5 | BRP.4 | BRP.3 | BRP.2 | BRP.1 | BRP.0 |
| ADDRESS 7: BUS TIMING REGISTER 1 | | | | | | | |
| SAM | TSEG2.2 | TSEG2.1 | TESG2.0 | TSEG1.3 | TSEG1.2 | TSEG1.1 | TSEG1.0 |
| ADDRESS 8: OUTPUT CONTROL REGISTER | | | | | | | |
| OCTP1 | OCTN1 | OCPOL1 | OCTP0 | OCTN0 | OCPOL0 | OCMODE1 | OCMODE0 |
| ADDRESS 9: TEST REGISTER (note 1) | | | | | | | |
| Reserved | Reserved | Map Internal Register | Connect RX Buffer 0 CPU | Connect TX Buffer CPU | Access Internal Bus | Normal RAM Connect | Float Output Driver |

**Table 40** Effects of setting the Reset Request bit HIGH (present)

| TYPE | BIT | SYMBOL | FUNCTION | EFFECT |
|---|---|---|---|---|
| Control | CR.7 | TM | Test Mode | LOW (disabled) |
| | CR.5 | RA | Reference Active | HIGH (output); note 1 |
| Command | CMR.7 | RX0A | RX0 Active | HIGH (RX0 = CRX0); note 1 |
| | CMR.6 | RX1A | RX1 Active | HIGH (RX1 = CRX1); note 1 |
| | CMR.4 | SLP | Sleep | LOW (wake-up) |
| | CMR.3 | COS | Clear Overrun Status | HIGH (clear) |
| | CMR.2 | RRB | Release Receive Buffer | HIGH (released) |
| | CMR.1 | AT | Abort Transmission | LOW (absent) |
| | CMR.0 | TR | Transmission Request | LOW (absent) |
| Status | SR.7 | BS | Bus Status | LOW (Bus-On); note 1 |
| | SR.6 | ES | Error Status | LOW (no error); note 1 |
| | SR.5 | TS | Transmit Status | LOW (idle) |
| | SR.4 | RS | Receive Status | LOW (idle) |
| | SR.3 | TCS | Transmission Complete Status | HIGH (complete) |
| | SR.2 | TBS | Transmit Buffer Access | HIGH (released) |
| | SR.1 | DO | Data Overrun | LOW (absent) |
| | SR.0 | RBS | Receive Buffer Status | LOW (empty) |
| Interrupt | IR.3 | OI | Overrun Interrupt | LOW (reset) |
| | IR.1 | TI | Transmit Interrupt | LOW (reset) |
| | IR.0 | RI | Receive Interrupt | LOW (reset) |

**Note**

1. Only after an external reset; see note 5 to Table 37 “Description of the SR bits”.

*13.5.15.3 Special Function Register CANCON*

**Table 63** SFR CANCON in **Read** access (address D9H)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | WUI | OI | EI | TI | RI |

**Table 64** Description of the CANCON bits in **Read** access
When reading CANCON the Interrupt Register of the CAN-controller is accessed.

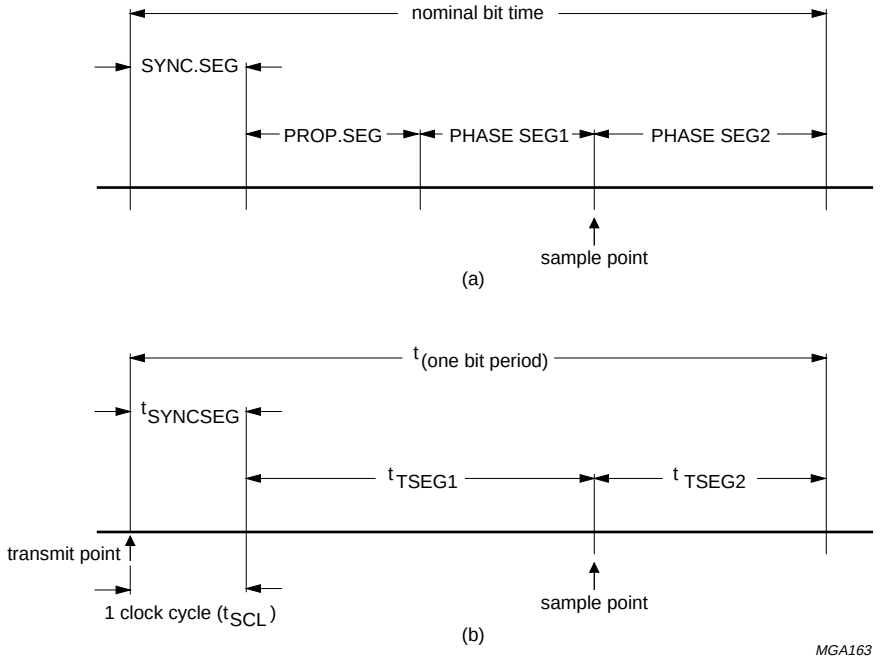
| BIT | SYMBOL | FUNCTION |
|---|---|---|
| 7 | – | Reserved; bits are read as HIGH. |
| 6 | – | |
| 5 | – | |
| 4 | WUI | Wake-Up Interrupt (see Table 39). |
| 3 | OI | Overrun Interrupt (see Table 39). |
| 2 | EI | Error Interrupt (see Table 39). |
| 1 | TI | Transmit Interrupt (see Table 39). |
| 0 | RI | Receive Interrupt (see Table 39). |

**Table 65** SFR CANCON in **Write** access (address D9H)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RX0A | RX1A | WUM | SLP | COS | RRB | AT | TR |

**Table 66** Description of the CANCON bits in **Write** access
When writing to CANCON then the Command Register of the CAN-controller is accessed.

| BIT | SYMBOL | FUNCTION |
|---|---|---|
| 7 | RX0A | RX0 Active (see Table 34). |
| 6 | RX1A | RX1 Active (see Table 34). |
| 5 | WUM | Wake-Up Mode (see Table 34). |
| 4 | SLP | Sleep (see Table 34). |
| 3 | COS | Clear Overrun Status (see Table 34). |
| 2 | RRB | Release Receive Buffer (see Table 34). |
| 1 | AT | Abort Transmission (see Table 34). |
| 0 | TR | Transmission Request (see Table 34). |




(a) As defined by the CAN-protocol.
(b) As implemented in the P8xCE598's on-chip CAN-controller.

Fig.18 Bit period.

*13.5.19.2 Time Segment 1 (TSEG1)*

This segment determines the location of the sampling point within a bit period, which is at the end of TSEG1. TSEG1 is programmable from 1 to 16 system clock cycles (see Section 13.5.10).

The correct location of the sample point is essential for the correct functioning of a transmission. The following points must be taken into consideration:

- A Start-Of-Frame (see Section 13.6.2) causes all CAN-controllers to perform a 'hard synchronization' (see Section 13.5.20) on the first recessive-to-dominant edge. During arbitration, however, several CAN-controllers may simultaneously transmit. Therefore it may require twice the sum of bus-line, input comparator and the output driver delay times until the bus is stable. This is the propagation delay time.
- To avoid sampling at an incorrect position, it is necessary to include an additional synchronization buffer on both sides of the sample point. The main reasons for incorrect sampling are:
  - incorrect synchronization due to spikes on the bus-line
  - slight variations in the oscillator frequency of each CAN-controller in the network, which results in a phase error.
- Time Segment 1 consists of the segment for compensation of propagation delays and the synchronization buffer segment directly before the sample point (see Fig.18).

#### *13.5.19.3 Time Segment 2 (TSEG2)*

This time segment provides:

- additional time at the sample point for calculation of the subsequent bit levels (e.g. arbitration)
- synchronization buffer segment directly after the sample point.

TSEG2 is programmable from 1 to 8 system clock cycles (see Section 13.5.10.

#### *13.5.19.4 Synchronisation Jump Width (SJW)*

SJW defines the maximum number of clock cycles ($t_{SCL}$) a period may be reduced or increased by one resynchronization. SJW is programmable from 1 to 4 system clock cycles, see Section 13.5.2.

#### *13.5.19.5 Propagation Delay Time ($t_{prop}$)*

The Propagation Delay Time is:

$$t_{prop} = 2 \times (\text{physical bus delay} + \text{input comparator delay} + \text{output driver delay}).$$

$t_{prop}$ is rounded up to the nearest multiple of $t_{SCL}$.

#### *13.5.19.6 Bit Timing Restrictions*

Restrictions on the configuration of the bit timing are based on internal processing. The restrictions are:

- $t_{TSEG2} \geq 2t_{SCL}$
- $t_{TSEG2} \geq t_{SJW}$
- $t_{TSEG1} \geq t_{SEG2}$
- $t_{TSEG1} \geq t_{SJW} + t_{prop}$.

The three sample mode (SAM = HIGH) has the effect of introducing a delay of one system clock cycle on the bus-line. This must be taken into account for the correct calculation of TSEG1 and TSEG2:

- $t_{TSEG1} \geq t_{SJW} + t_{prop} + 2t_{SCL}$
- $t_{TSEG2} \geq 3t_{SCL}$.

### 13.5.20 SYNCHRONIZATION

Synchronization is performed by a state machine which compares the incoming edge with its actual bit timing and adapts the bit timing by hard synchronization or resynchronization.

This type of synchronization occurs only at the beginning of a message.

The CAN-controller synchronizes on the first incoming recessive-to-dominant edge of a message (being the leading edge of a message's Start-Of-Frame bit; see Section 13.6.2.

Resynchronization occurs during the transmission of a message's bit stream to compensate for:

- Variations in individual CAN-controller oscillator frequencies
- Changes introduced by switching from one transmitter to another (e.g. during arbitration).

As a result of resynchronization either $t_{TSEG1}$ may be increased by up to a maximum of $t_{SJW}$ or $t_{TSEG2}$ may be decreased by up to a maximum of $t_{SJW}$:

- $t_{TSEG1} \leq t_{SCL}$ [(TSEG1 + 1) + (SJW + 1)]
- $t_{TSEG2} \geq t_{SCL}$ [(TSEG2 + 1) – (SJW + 1)].

TSEG1, TSEG2 and SJW are the programmed numerical values.

The phase error (e) of an edge is given by the position of the edge relative to SYNCSEG, measured in system clock cycles ($t_{SCL}$).

The value of the phase error is defined as:

- e = 0, if the edge occurs within SYNCSEG
- e > 0, if the edge occurs within TSEG1
- e < 0, if the edge occurs within TSEG2.

The effect of resynchronization is:

- The same as that of a hard synchronization, if the magnitude of the phase error (e) is less or equal to the programmed value of $t_{SJW}$
- To increase a bit period by the amount of $t_{SJW}$,if the phase error is positive and the magnitude of the phase error is larger than $t_{SJW}$
- To decrease a bit period by the amount of $t_{SJW}$ if the phase error is negative and the magnitude of the phase error is larger than $t_{SJW}$.

#### *13.5.20.1 Synchronization Rules*

The synchronization rules are as follows:

- Only one synchronization within one bit time is used.
- An edge is used for synchronization only if the value detected at the previous sample point differs from the bus value immediately after the edge.
- Hard synchronization is performed whenever there is a recessive-to-dominant edge during Bus-Idle (see Section 13.6.6).
- All other edges (recessive-to-dominant and optionally dominant-to recessive edges if the Sync bit is set HIGH (see Section 13.5.3) which are candidates for resynchronization will be used with the following exception:
  - A transmitting CAN-controller will not perform a resynchronization as a result of a recessive-to-dominant edge with positive phase error, if only these edges are used for resynchronization. This ensures that the delay times of the output driver and input comparator do not cause a permanent increase in the bit time.

## 13.6 CAN 2.0A Protocol description

### 13.6.1 FRAME TYPES

The P8xCE598's CAN-controller supports the four different CAN-protocol frame types for communication:

- Data Frame, to transfer data
- Remote Frame, request for data
- Error Frame, globally signal a (locally) detected error condition
- Overload Frame, to extend delay time of subsequent frames (an Overload Frame is not initiated by the P8xCE598 CAN-Controller).

#### *13.6.1.1 Bit representation*

There are two logical bit representations used in the CAN-protocol:

- A recessive bit on the bus-line appears only if all connected CAN-controllers send a recessive bit at that moment.
- Dominant bits always overwrite recessive bits i.e. the resulting bit level on the bus-line is dominant.

### 13.6.2 DATA FRAME

A Data Frame carries data from a transmitting CAN-controller to one or more receiving ones.
A Data Frame is composed of seven different bit-fields:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field (may have a length of zero)
- CRC Field
- Acknowledge Field
- End-Of-Frame.

#### *13.6.2.1 Start-Of-Frame bit*

Signals the start of a Data Frame or Remote Frame. It consists of a single dominant bit use for hard synchronization of a CAN-controller in receive mode.

#### *13.6.2.2 Arbitration Field*

Consists of the message Identifier and the RTR bit. In the case of simultaneous message transmissions by two or more CAN-controllers the bus access conflict is solved by bit-wise arbitration, which is active during the transmission of the Arbitration Field.

#### *13.6.2.3 Identifier*

This 11-bit field is used to provide information about the message, as well as the bus access priority. It is transmitted in the order ID.10 to ID.0 (LSB). The situation that the seven most significant bits (ID.10 to ID.4) are all recessive must not occur.

An Identifier does not define which particular CAN-controller will receive the frame because a CAN based communication network does not differentiate between a point-to-point, multicast or broadcast communication.

#### *13.6.8.3 CRC Error*

To ensure the validity of a transmitted message all receivers perform a CRC check. Therefore, in addition to the (destuffed) information digits (Start-Of-Frame up to Data Field), every message includes some control digits (CRC Sequence; generated by the transmitting CAN-controller of the respective message) used for error detection.

The code used by all CAN-controllers is a (shortened) BCH code, extended by a parity check and has the following attributes:

- 127 bits as maximum length of the code.
- 112 bits as maximum number of information digits (max. 83 bits are used by the CAN-controller).
- Length of the CRC Sequence amounts to 15 bits.
- Hamming distance d = 6.

As a result, '(d–1)' random errors are detectable (some exceptions exist).

The CRC Sequence is determined (calculated) by the following procedure:

1. The destuffed bit stream consisting of Start-Of-Frame up to the Data Field (if present) is interpreted as polynomial with coefficients 0 or 1.
2. This polynomial is divided (modulo-2) by the following generator polynomial, which includes a parity check:
   $f(x) = (x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1)$
   $(x + 1) = 1100010110011001$ B.
3. The remainder of this polynomial division is the CRC sequence.

Burst errors are detected up to a length of 15 [degree of f(x)]. Multiple errors (number of disturbed bits at least d = 6) are not detected with a residual error probability of $2^{-15}$ $(3 \times 10^{-5})$ by CRC check only.

#### *13.6.8.4 Form Error*

Form Errors result from violations of the fixed form of the following bit fields:

- CRC Delimiter
- Acknowledge Delimiter
- End-Of-Frame
- Error Delimiter
- Overload Delimiter.

During the transmission of these bit fields an error condition is recognized if a dominant bit level instead of a recessive one is detected.

#### *13.6.8.5 Acknowledgement Error*

This is detected by a transmitter whenever it does not monitor a dominant bit during the Acknowledge Slot.

#### *13.6.8.6 Error detection by an Error Flag from another CAN-controller*

The detection of an error is signalled by transmitting an Error Flag. An Active Error Flag causes a Stuff Error, a Bit Error or a Form Error at all other CAN-controllers.

#### *13.6.8.7 Error Detection Capabilities*

Errors which occur at all CAN-controllers (global errors) are 100% detected. For local errors, i.e. for errors occurring at some CAN-controllers only, the shortened BCH code, extended by a parity check, has the following error detection capabilities:

- Up to five single Bit Errors are 100% detected, even if they are distributed randomly within the code.
- All single Bit Errors are detected if their total number (within the code) is odd.
- The residual error probability of the CRC check amounts to $(3 \times 10^{-5})$. As an error may be detected not only by CRC check but also by other detection processes described above the residual error probability is several magnitudes less than $(3 \times 10^{-5})$.

### 13.6.9 Error confinement definitions

#### *13.6.9.1 Bus-OFF*

A CAN-controller which has too many unsuccessful transmissions, relative to the number of successful transmissions, will enter the Bus-OFF state. It remains in this state, neither receiving nor transmitting messages until the Reset Request bit is set LOW (absent) and both Error Counters set to 0 (see Section 13.6.10).

#### *13.6.9.2 Acknowledge*

A CAN-controller which has received a valid message correctly, indicates this to the transmitter by transmitting a dominant bit level on the bus during the Acknowledge Slot, independent of accepting or rejecting the message.

#### *13.6.9.3 Error-Active*

An error-active CAN-controller in its normal operating state is able to receive and to transmit normally and also to transmit an Active Error Flag (see Section 13.6.10).

**Table 88** Description of the mnemonics in the Instruction set

| MNEMONIC | DESCRIPTION |
|---|---|
| **Data addressing modes** | |
| Rr | Working register R0-R7. |
| direct | 128 internal RAM locations and any special function register (SFR). |
| @Ri | Indirect internal RAM location addressed by register R0 or R1 of the actual register bank. |
| #data | 8-bit constant included in instruction. |
| #data 16 | 16-bit constant included as bytes 2 and 3 of instruction. |
| bit | Direct addressed bit in internal RAM or SFR. |
| addr16 | 16-bit destination address. Used by LCALL and LJMP.<br>The branch will be anywhere within the 64 kbytes Program Memory address space. |
| addr11 | 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 kbytes page of Program Memory as the first byte of the following instruction. |
| rel | Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps.<br>Range is –128 to +127 bytes relative to first byte of the following instruction. |
| **Hexadecimal opcode cross-reference** | |
| * | 8, 9, A, B, C, D, E, F. |
| • | 1, 3, 5, 7, 9, B, D, F. |
| ♦ | 0, 2, 4, 6, 8, A, C, E. |

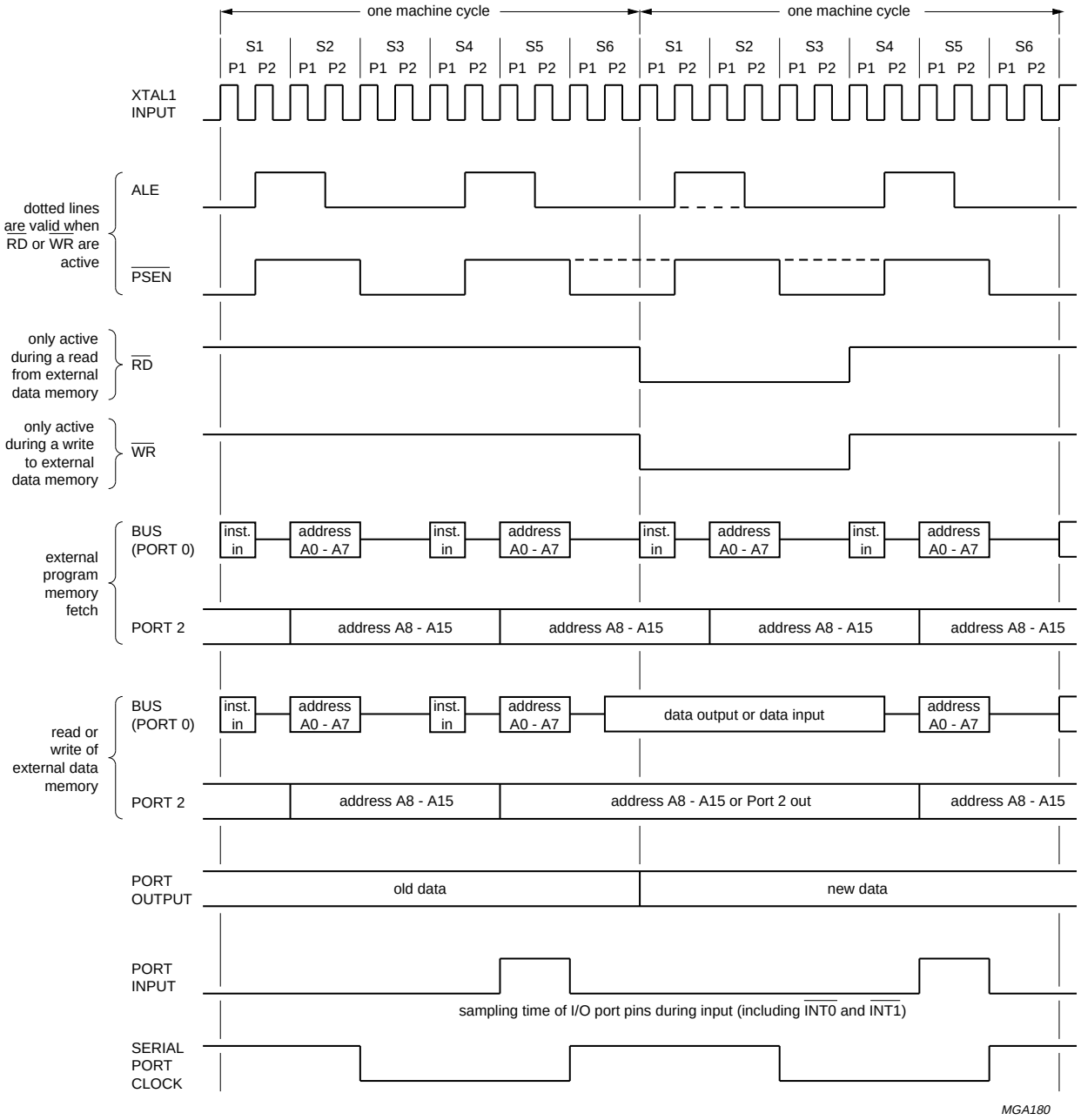
## 21 AC CHARACTERISTICS

See notes 1 and 2; $C_L$ = 100 pF for Port 0, ALE and $\overline{PSEN}$; $C_L$ = 80 pF for all other outputs unless otherwise specified.

| SYMBOL | PARAMETER | $f_{CLK}$ = 16 MHz | | $f_{CLK}$ = 12 MHz | | VARIABLE CLOCK 1.2 to 16 MHz | | UNIT |
|---|---|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| **External Program Memory** | | | | | | | | |
| $t_{LHLL}$ | ALE pulse width | 85 | – | 127 | – | $2t_{CLK} - 40$ | – | ns |
| $t_{AVLL}$ | address valid to ALE LOW | 8 | – | 28 | – | $t_{CLK} - 55$ | – | ns |
| $t_{LLAX}$ | address hold after ALE LOW | 28 | – | 48 | – | $t_{CLK} - 35$ | – | ns |
| $t_{LLIV}$ | ALE LOW to valid instruction in | – | 150 | – | 233 | – | $4t_{CLK} - 100$ | ns |
| $t_{LLPL}$ | ALE LOW to $\overline{PSEN}$ LOW | 23 | – | 43 | – | $t_{CLK} - 40$ | – | ns |
| $t_{PLPH}$ | $\overline{PSEN}$ pulse width | 143 | – | 205 | – | $3t_{CLK} - 45$ | – | ns |
| $t_{PLIV}$ | $\overline{PSEN}$ LOW to valid instruction in | – | 83 | – | 145 | – | $3t_{CLK} - 105$ | ns |
| $t_{PXIX}$ | input instruction hold after $\overline{PSEN}$ | 0 | – | 0 | – | 0 | – | ns |
| $t_{PXIZ}$ | input instruction float after $\overline{PSEN}$ | – | 38 | – | 59 | – | $t_{CLK} - 25$ | ns |
| $t_{AVIV}$ | address to valid instruction in | – | 208 | – | 312 | – | $5t_{CLK} - 105$ | ns |
| $t_{PLAZ}$ | $\overline{PSEN}$ LOW to address float | – | 10 | – | 10 | – | 10 | ns |
| **External data memory** | | | | | | | | |
| $t_{RLRH}$ | $\overline{RD}$ pulse width | 275 | – | 400 | – | $6t_{CLK} - 100$ | – | ns |
| $t_{WLWH}$ | $\overline{WR}$ pulse width | 275 | – | 400 | – | $6t_{CLK} - 100$ | – | ns |
| $t_{AVLL}$ | address valid to ALE LOW | 8 | – | 28 | – | $t_{CLK} - 55$ | – | ns |
| $t_{LLAX}$ | address hold after ALE LOW | 33 | – | 53 | – | $t_{CLK} - 30$ | – | ns |
| $t_{RLDV}$ | $\overline{RD}$ LOW to valid data in | – | 148 | – | 252 | – | $5t_{CLK} - 165$ | ns |
| $t_{RHDX}$ | data hold after $\overline{RD}$ | 0 | – | 0 | – | 0 | – | ns |
| $t_{RHDZ}$ | data float after $\overline{RD}$ | – | 55 | – | 97 | – | $2t_{CLK} - 70$ | ns |
| $t_{LLDV}$ | ALE LOW to valid data in | – | 350 | – | 517 | – | $8t_{CLK} - 150$ | ns |
| $t_{AVDV}$ | address to valid data in | – | 398 | – | 585 | – | $9t_{CLK} - 165$ | ns |
| $t_{LLWL}$ | ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW | 138 | 238 | 200 | 300 | $3t_{CLK} - 50$ | $3t_{CLK} + 50$ | ns |
| $t_{AVWL}$ | address valid to $\overline{RD}$ or $\overline{WR}$ LOW | 120 | – | 203 | – | $4t_{CLK} - 130$ | – | ns |
| $t_{WHLH}$ | $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH | 23 | 103 | 43 | 123 | $t_{CLK} - 40$ | $t_{CLK} + 40$ | ns |
| $t_{QVWX}$ | data valid to $\overline{WR}$ transition | 3 | – | 23 | – | $t_{CLK} - 60$ | – | ns |
| $t_{QVWH}$ | data valid time $\overline{WR}$ HIGH | 288 | – | 433 | – | $7t_{CLK} - 150$ | – | ns |
| $t_{WHQX}$ | data hold after $\overline{WR}$ | 13 | – | 33 | – | $t_{CLK} - 50$ | – | ns |
| $t_{RLAZ}$ | $\overline{RD}$ LOW to address float | – | 0 | – | 0 | – | 0 | ns |

**Notes**

1. For the AC Characteristics the following conditions are valid:**P8xCE598 FxB**: $V_{DD}$ = 5 V ± 10%; $T_{amb}$ = –40 to +85 °C (125 °C); $f_{CLK}$ = 1.2 to 16 MHz.
2. $t_{CLK} = \frac{1}{f_{CLK}}$ = one oscillator clock period; $t_{CLK}$ = 62.5 ns at $f_{CLK}$ = 16 MHz.




The Port 5 input buffers have a maximum propagation delay of 300 ns.
As a result Port 5 sample time begins 300 ns before state S5 and ends when S5 has finished.

Fig.30 Instruction cycle timing.

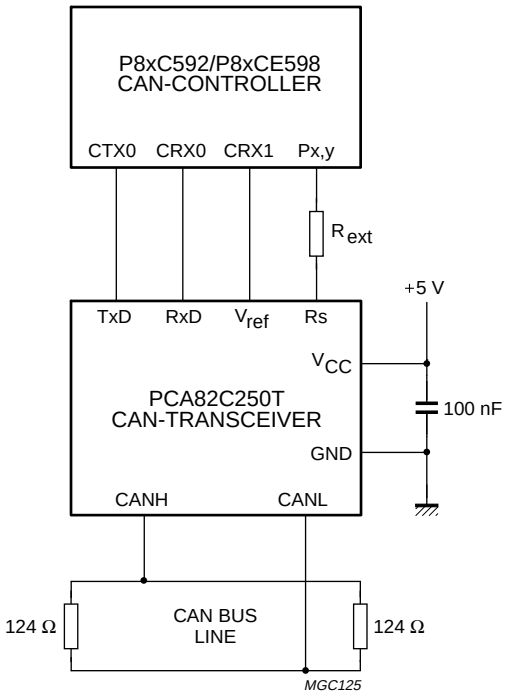



Fig.36 Application of a CAN-Transceiver (PCA82C250T).

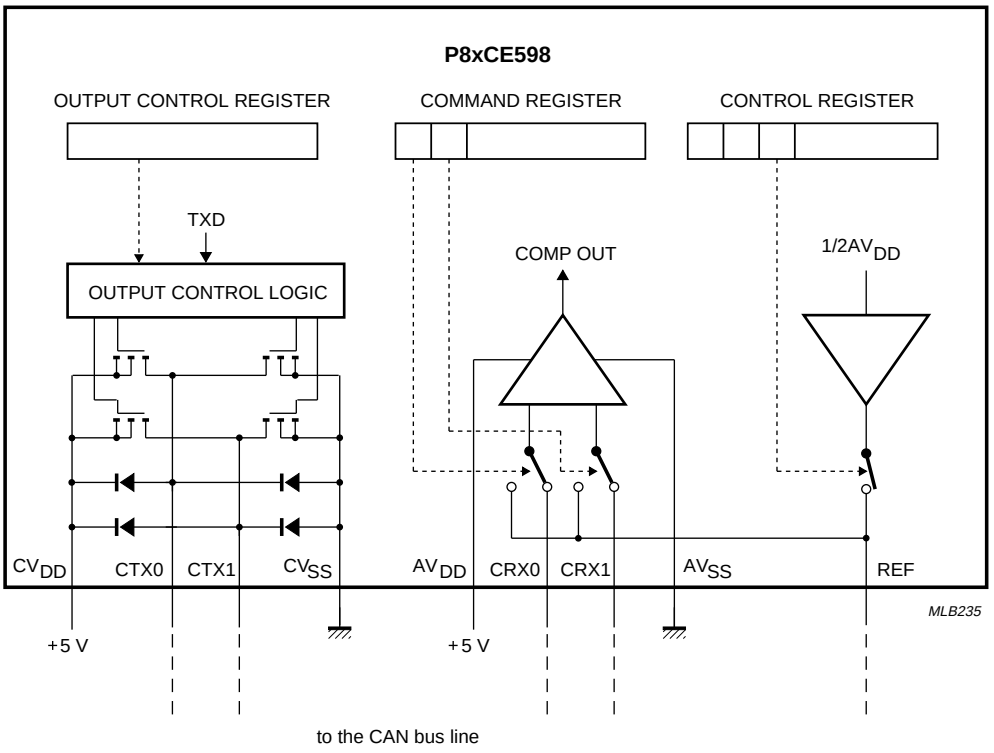



Fig.37 Structure of on-chip CAN-Transceiver.

### 22.2.5 P8xCE598 CAN Interrupt Handler Software example (including fast DMA transfer)

**MCS-51 MACRO ASSEMBLER CAN interrupt-handler**

```
LOC  OBJ   LINE  SOURCE

           1     $TITLE 8XCE598 (CAN interrupt-handler)
00A0       2     $NOSYMBOLS NOPAGING
00A1       3
           4     ;*****************************************************************************************************
           5     ;
           6     ;Very fast receive-routine for the 8XCE598 CAN Interface. It:
           7     • is embedded in the interrupt-handler for the CAN Controller,
           8     • uses the DMA-logic and
           9     • handles up to eight different messages
00A2       10    ;(if these have the same leading 8 identifier-bits).
           11    ;
           12    ;To allow for faster receive-routine, it is assumed that all other routines
           13    ;accessing the CAN Controller, disable the interrupt of the CAN Controller
           14    ;(IEN0.5) during their execution.
00A5       15    ;
00A7       16    ;Version:    1.0
           17    ;Date:       12-April-90
           18    ;Author:     Bernhard Reckels
           19    ;at:         Philips Components Application Lab., Hamburg (PCALH)
00A9       20
00AB       21    ;*****************************************************************************************************
00AD       22
           23    ;*****************************************************************************************************
           24    ;initial stuff
           25    ;*****************************************************************************************************
           26
           27    ;equatas
           28
           29                ;addresses of Special Function Registers
00AE       30                CANADR      EQU     0DBH
00AF       31                CANDAT      EQU     0DAH
           32                CANCON      EQU     0D9H
00B0       33                CANSTA      EQU     0D8H
           34
```

```
LOC   OBJ      LINE  SOURCE
00A0           72    CAN_INT_HANDLER:
00A1           73
               74            ; first save used registers
      C0D0     75            PUSH        PSW
      C0E0     76            PUSH        ACC
               77
               78            ; store the CAN Controller's Interrupt Register contents
               79            ; (here: at a bit-addressable location).
00A2           80            ; This is necessary because after reading the Interrupt Register
               81            ; its contents is cleared, but – on the other hand – several flags
               82            ; may be set in coincidence.
      E5D9     83            MOV         A, CANON
      541F     84            ANL         A, #INT_FLAG_MASK     ; only interrupt-flags
00A5  F520     85            MOV         CAN_INT_IMAGE, A
00A7           86
               87
               88    ;dispatcher----------------------------------------------------------------------------------------
               89    INT_TEST0:
00A9  100000   90            JBC     CAN_INT_RX,CAN_RX_SERV    ;receive-interrupt?
00AB           91
00AD           92    INT_TEST1:
               93            ; here the dispatcher has to be completed according
               94            ; to the application-specific requirements
               95            ; ...
               96            ; ...
               97    ; end of dispatcher--------------------------------------------------------------------------------
               98
               99    ;Rx-serve------------------------------------------------------------------------------------------
00AE           100   ; copy message (Data-Field only) from CAN- to CPU memory
00AF           101
               102   CAN_RX_SERVE
00B0           103           ; read 2nd Descriptor-Byte from the Rx-Buffer (address 21)
      75DB15   104           MOV         CANADR, #CAN_REF + 1
      E5DA     105           MOV         A, CANDAT
               106
```

# *Philips Semiconductors – a worldwide company*


**Argentina:** see South America

**Australia:** 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

**Austria:** Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

**Belarus:** Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

**Belgium:** see The Netherlands

**Brazil:** see South America

**Bulgaria:** Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

**Canada:** PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

**China/Hong Kong:** 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

**Colombia:** see South America

**Czech Republic:** see Austria

**Denmark:** Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 1949

**Finland:** Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 615 800, Fax. +358 615 80920

**France:** 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

**Germany:** Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 52 60, Fax. +49 40 23 536 300

**Greece:** No. 15, 25th March Street, GR 17778 TAVROS,
Tel. +30 1 4894 339/911, Fax. +30 1 4814 240

**Hungary:** see Austria

**India:** Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

**Indonesia:** see Singapore

**Ireland:** Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

**Israel:** RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,
Tel. +972 3 645 0444, Fax. +972 3 648 1007

**Italy:** PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

**Japan:** Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

**Korea:** Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

**Malaysia:** No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

**Mexico:** 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

**Middle East:** see Italy

**Netherlands:** Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 83749, Fax. +31 40 27 88399

**New Zealand:** 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

**Norway:** Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

**Philippines:** Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

**Poland:** Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

**Portugal:** see Spain

**Romania:** see Italy

**Russia:** Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 926 5361, Fax. +7 095 564 8323

**Singapore:** Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

**Slovakia:** see Austria

**Slovenia:** see Italy

**South Africa:** S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

**South America:** Rua do Rocio 220, 5th floor, Suite 51,
04552-903 São Paulo, SÃO PAULO - SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 829 1849

**Spain:** Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

**Sweden:** Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

**Switzerland:** Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 481 7730

**Taiwan:** PHILIPS TAIWAN Ltd., 23-30F, 66,
Chung Hsiao West Road, Sec. 1, P.O. Box 22978,
TAIPEI 100, Tel. +886 2 382 4443, Fax. +886 2 382 4444

**Thailand:** PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

**Turkey:** Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

**Ukraine:** PHILIPS UKRAINE, 2A Akademika Koroleva str., Office 165,
252148 KIEV, Tel. +380 44 476 0297/1642, Fax. +380 44 476 6991

**United Kingdom:** Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

**United States:** 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

**Uruguay:** see South America

**Vietnam:** see Singapore

**Yugoslavia:** PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 825 344, Fax.+381 11 635 777

**For all other countries apply to:** Philips Semiconductors, Marketing & Sales Communications,
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

**Internet:** http://www.semiconductors.philips.com/ps/





Printed in The Netherlands 617021/1200/04/pp108 Date of release: 1996 Jun 27 Document order number: 9397 750 00915


*Let's make things better.*

**Philips Semiconductors**

PHILIPS