SIEMENS

# Stand Alone Full CAN Controller

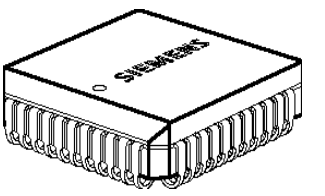
**SAE 81C90**
**SAE 81C91**

**Preliminary Data**

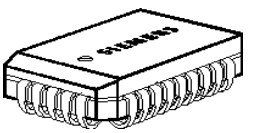
**CMOS IC**

## Features

- Full CAN controller for data rate up to 1 Mbaud
- Complies with CAN specification V2.0 part A (part B passive)
- Up to 16 messages at one time (each with maximum data length)
- Message identifier reprogrammable at any time without chip reset
- Several transmit jobs can be sent with a single command
- Transmit check
- Basic CAN feature
- Time stamp for eight messages
- Two host interfaces (parallel and serial)
- User-configurable outputs for different bus concepts
- Programmable clock output
- Two 8 bit I/O-Port extension (P-LCC-44-1 only)




**P-LCC-44-1**



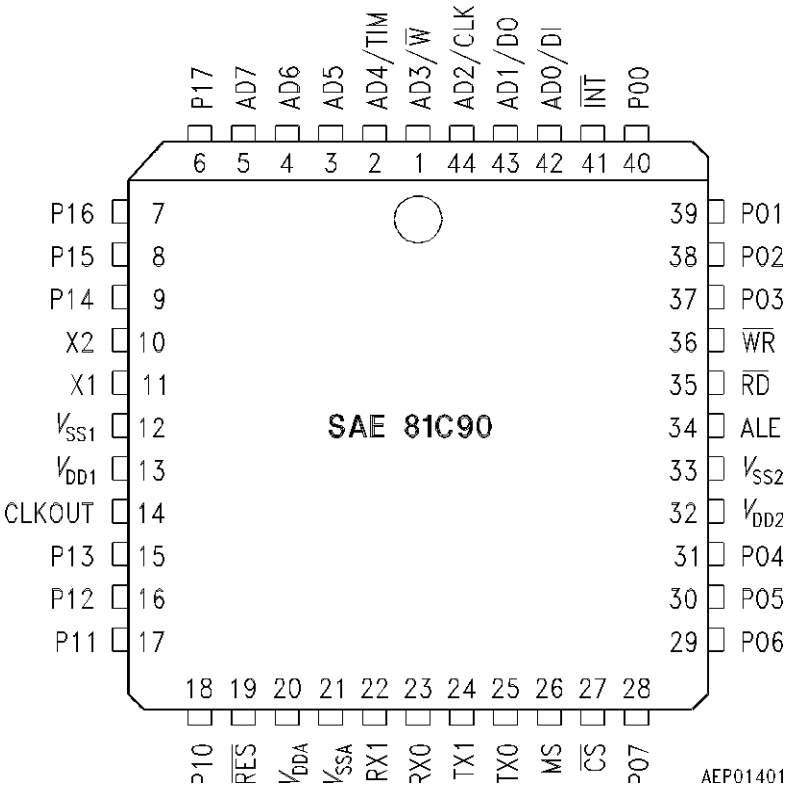

**P-LCC-28-1**

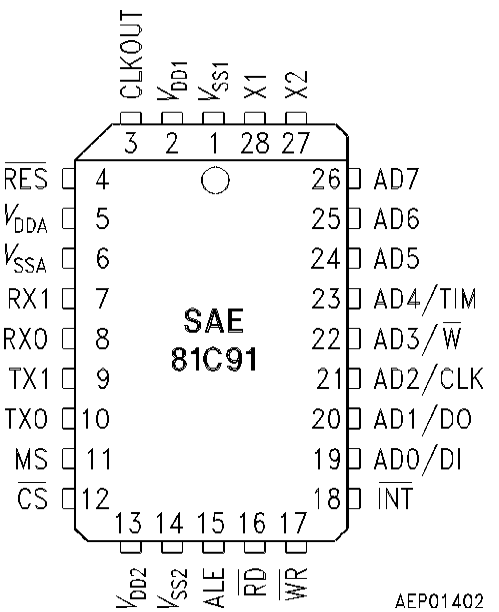
| Type | Ordering Code | Package |
|---|---|---|
| SAE 81C90 | Q67100-H9038 | P-LCC-44-1 (SMD) |
| SAE 81C91 | Q67106-H9037 | P-LCC-28-1 (SMD) |

The Siemens Stand Alone Full CAN (SFCAN) circuit incorporates all the parts for completely independent transmission and reception of messages using the CAN protocol. The flexible, programmable interface allows connection to different implementations of the physical layer. The link to a host controller can be made either by a multiplexed 8-bit address/data bus or by a high-speed, serial synchronous interface.

CAN: License of Robert Bosch GmbH

**Pin Configuration**
(top view)



## 1 Pin Definitions and Functions

| Pin No. | | Symbol | Input (I) Output (O) | Function |
|---|---|---|---|---|
| P-LCC-44-1 | P-LCC-28-1 | | | |
| 11 | 28 | X1 | I | Crystal oscillator |
| 10 | 27 | X2 | I | Crystal oscillator |
| 14 | 3 | CLKOUT | O | Clock output |
| 19 | 4 | $\overline{\text{RES}}$ | I | Reset |
| 42 | 19 | AD0/DI | I/O | PI: Address/Data bus/SI: Data input |
| 43 | 20 | AD1/DO | I/O | PI: Address/Data bus/SI: Data output |
| 44 | 21 | AD2/CLK | I/O | PI: Address/Data bus/SI: Clock input |
| 1 | 22 | AD3/$\overline{\text{W}}$ | I/O | PI: Address/Data bus/SI: Write select |
| 2 | 23 | AD4/TIM | I/O | PI: Address/Data bus/SI:TIM = 0 : Timing A<br>TIM = 1 : Timing B |
| 3 | 24 | AD5 | I/O | PI: Address/Data bus |
| 4 | 25 | AD6 | I/O | PI: Address/Data bus |
| 5 | 26 | AD7 | I/O | PI: Address/Data bus |
| 35 | 16 | $\overline{\text{RD}}$ | I | PI: Read/SI: no Function |
| 36 | 17 | $\overline{\text{WR}}$ | I | PI: Write/SI: no Function |
| 34 | 15 | ALE | I | PI: Address Latch Enable/SI: no Function |
| 27 | 12 | $\overline{\text{CS}}$ | I | Chip Select |
| 41 | 18 | $\overline{\text{INT}}$ | O | Interrupt |
| 26 | 11 | MS | I | Mode Select (PI $\leftrightarrow$ SI) |
| 28, 29, 30, 31,<br>37, 38, 39, 40 | – | P00, … P03,<br>P04, … P07 | I/O | Port 0 |
| 6, 7, 8, 9,<br>15, 16, 17, 18 | – | P10, … P13,<br>P14, … P17 | I/O | Port 1 |
| 25 | 10 | TX0 | O | Transmitter output 0 |
| 24 | 9 | TX1 | O | Transmitter output 1 |
| 23 | 8 | RX0 | I | Comparator input 0, digital input |
| 22 | 7 | RX1 | I | Comparator input 1 |
| 20 | 5 | $V_{DDA}$ | I | analogue power supply of comparator (may be unconnected using the digital mode) |
| 21 | 6 | $V_{SSA}$ | I | analogue power ground of comparator (must always be connected) |
| 13 | 2 | $V_{DD1}$ | I | digital power supply |
| 32 | 13 | $V_{DD2}$ | I | digital power supply |
| 12 | 1 | $V_{SS1}$ | I | digital power ground |
| 33 | 14 | $V_{SS2}$ | I | digital power ground |

## 2 General Description

The Siemens stand-alone Full-CAN (SFCAN) circuit is a large-scale-integrated peripheral device that executes the entire protocol of an automobile or industrial network.

Bus communication is based on the controller-area-network (CAN) protocol. With features like short message length, guaranteed reaction time for messages of appropriate importance, priority of bus access defined by message identifiers, powerful error detection and treatment plus ease of operation, the CAN protocol is optimally designed for the requirements of automobile and industrial electronic networks.

The SFCAN circuit incorporates all the parts for completely independent transmission and reception of messages using the CAN protocol. The flexible, programmable interface allows connection to different implementations of the physical layer. The link to a host controller can be made either by a multiplexed 8-bit address/data bus or by a high-speed, serial synchronous interface.

The device comes in two versions: SAE 81C90 in a P-LCC-44-1 package with two 8-bit l/O ports, and SAE 81C91 in a P-LCC-28-1 package without l/O ports.

### 2.1 Main Features

- Multi-master capability
- Assignment of message priorities for high priority messages
- Guaranteed reaction time
- Flexible configuration
- Several receivers simultaneously
- System-wide data consistency
- Resolution of bus accesses without time losses (arbitration)
- High data rates up to 1 Mbaud
- Error detection with automatic repeated transmission of disturbed messages
- Discrimination between temporary and permanent errors, automatic, disconnection of defective stations
- Complies with CAN specification V2.0 part A (part B passive= messages according to part B are tolerated but not latched)
- Extra check of data from CPU and transmitted data for equality (transmit check)
- Up to 16 messages at one time with maximum data length
- Message identifier reprogrammable at any time without chip reset
- Several transmit jobs can be sent with a single command
- One message memory can be configured so that all messages are received except those covered by other memory locations (monitor function, basic CAN feature)
- Time stamp for eigth messages for rating actuality or repetition rate of messages
- User-configurable outputs for different bus concepts
- Programmable clock output

## 3 Circuit Description

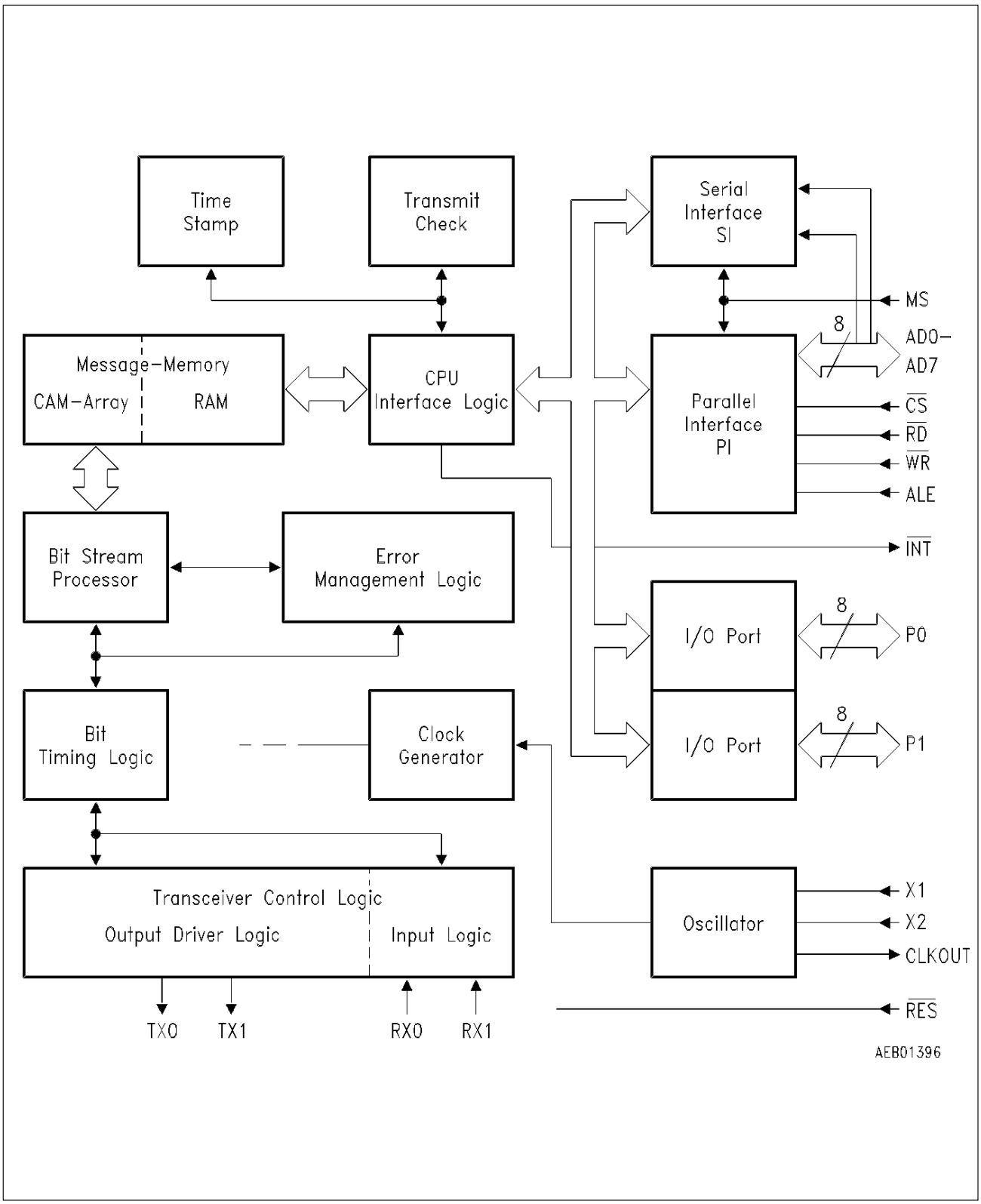



**Figure 1**
**Block Diagram**

### 3.1 Message Memory

The SFCAN circuit filters incoming messages with an associative memory (CAM = content-addressable memory). For this the identifier and RTR bits of the required message must be written to the appropriate memory location.

The identifier of each incoming message is compared to the identifiers held in CAM. When equality is detected, the received data bytes are written into the matching RAM location. At the same time the corresponding receive-ready bit is set and a receive interrupt is generated, if it is enabled. If no match is detected, the received message is rejected.

Identifiers can be reprogrammed at any time, although it is possible that data of the old or new identifier may be lost during reprogramming.

An incoming transmit request will only be satisfied automatically by the hardware if the RTR bit of the particular identifier is set in CAM.

To ensure data consistency when reading or writing several data bytes, these data are buffered in a 64-bit shadow register (**see Figure 2** 'CAM, Message Memory and Time-Stamp Registers' on the next page).

Writing must start with the most-significant data byte. When data byte 0 is written, the contents of the shadow register are transferred in parallel into the RAM of the appropriate memory location. In readout the data are transferred into the shadow register automatically by interpreting the address of the byte that has just been read. In this way, for example, reading address $83_H$ insures that the data of address $80_H$ through $87_H$ go into the shadow register. But transfer is only made if a date of another message has been read out beforehand. Upon readout of the higher-level data byte (e.g. address $87_H$) the shadow register is newly loaded in any case.

For these reasons it is absolutely essential to insure the writing of data is not interrupted by a read operation and vice versa, a read operation should not be interrupted by a write.

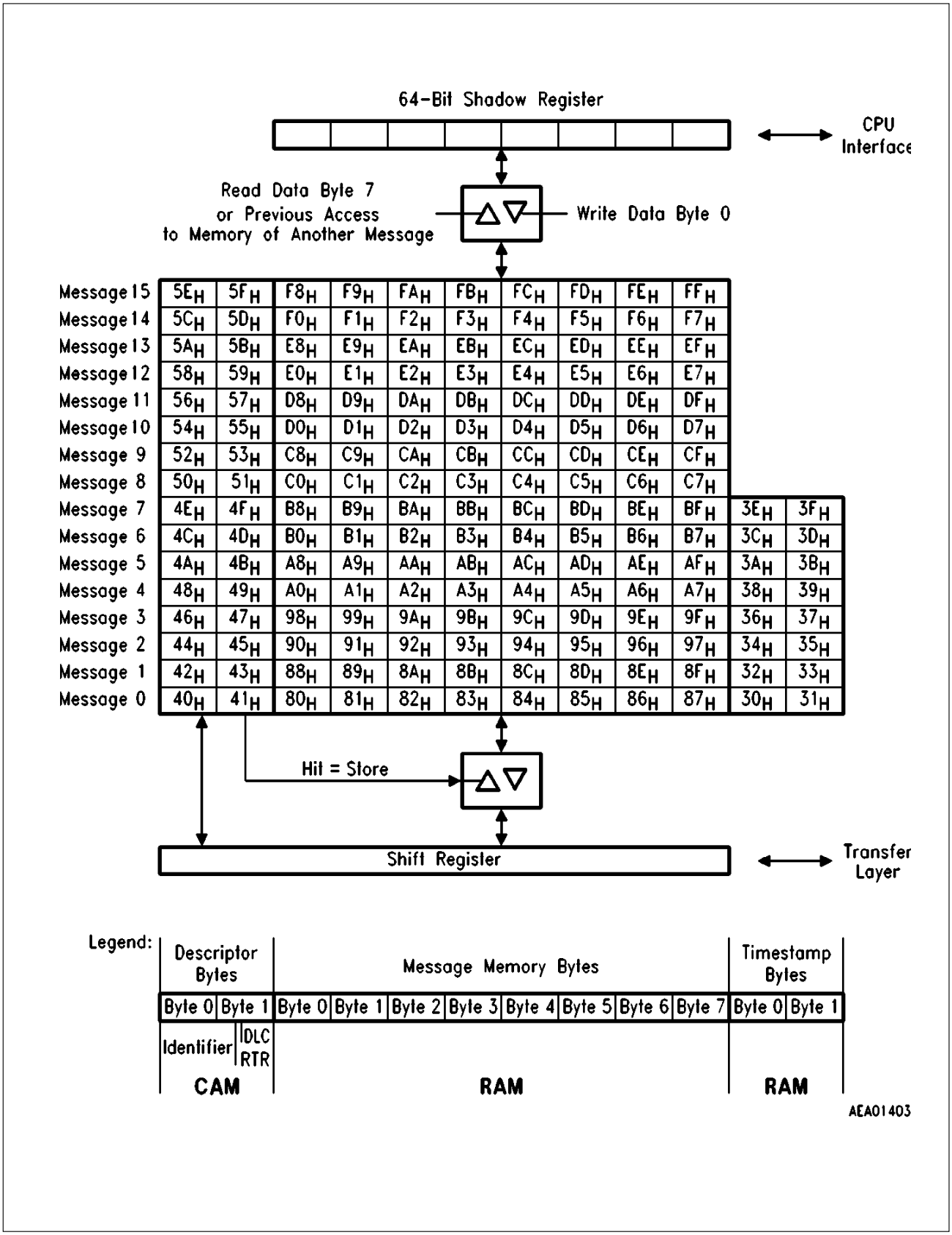



**Figure 2**
**CAM, Message Memory and Time-Stamp Registers**

### 3.2 Bit Stream Processor (BSP)

The bit-stream processor controls the entire protocol, differentiates between the frames types and detects frame errors.

### 3.3 Error Management Logic (EML)

The error-management logic receives error messages from the BSP and, in turn, sends back information about error state to the BSP and CIL.

### 3.4 Bit Timing Logic (BTL)

The bit-timing logic determines the timing of the bits and synchronises with the edges of the bit stream on the CAN bus.

### 3.5 Transceiver Control Logic (TCL)

The transceiver-control logic consists of programmable output driver, input comparator and input multiplexer.

### 3.6 Clock Generator (CG)

The clock generator consists of an oscillator and a programmable divider. The oscillator can be fed from an external quartz crystal, ceramic resonator or an external timing source. The permissible crystal frequency is 1 to 20 MHz, and the external clock may be between 0 and 20 MHz. A programmable frequency, dependent on the crystal clock, is available with the CLKOUT pin, e.g. for the clocking of a host controller.

### 3.7 CPU Interface Logic (CIL)

The CPU interface logic controls the access of the host via the parallel or serial interface, interprets the commands and outputs status and interrupt information.

### 3.8 Transmit Check

The CAN protocol insures a very large integrity for the data transferred over the bus. The on-chip path from the data stored in parallel to the serial bit stream is not protected by the protocol however. To eliminate any possible uncertainties at this point too, the SFCAN circuit incorporates a transmit-check unit. This reads back a transmitted message via the normal receive path from the bus interface and compares the data to those written into the message memory by the host controller. If any inconsistency of the data is detected, the current message will be invalidated by an error frame.

The transmit-check error counter TCEC is then incremented by 1. If this counter reaches 4, an error interrupt (bit TCI in the INT register) is generated, provided that this has not been masked (bit ETCI in the IMSK register). This count will also produce the bus-off status.

The TCEC is set to 0 after a reset and can be read and also written for test purposes at any time.

**Note:** The transmit-check is an additional feature of the Siemens Full CAN Chip and is not part of CAN protocol.

### 3.9 Time Stamp

Normally it is impossible to determine from the received data in the message memory when they were received. So the host controller is unable to derive any information about the actuality or the repetition rate of the data.

To enable an indication of the time of reception for at least some of the messages, there is a 16-bit timer implemented on the SFCAN circuit. The content of this is written into the time-stamp registers of the particular message when it is received in the memory locations of messages 0 through 7. There are two time-stamp bytes for each of the messages 0 through 7, and these hold the value of the 16-bit timer.

The actuality of a message is determined by subtracting the old time-stamp of a message, stored in the host controller, from the new one, with respect to the timer overflow bit.

Overflow of the timer can be detected by bit TSOV in the CTRL register. This bit does not trigger an interrupt and has to be reset by the host controller. Depending on the setting of bits TSP0 and TSP1 in the CTRL register, the counter is fed with 1/32, 1/64, 1/128 or 1/256 of the bus clock. The momentary timer status can be read and set at any time. The timer starts at 0 after a reset and cannot be stopped.

### 3.10 I/O-Ports

There are two parallel I/O ports in the SAE 81C90, each with eight pins. These ports are configured pin by pin as input or output, depending on the contents of the port-direction register.

The output data for the port pins can be written (latched) into the port-latch register. Reading this register reproduces the contents of the latch. The levels on the port pins can be read from the port-pin register.

For the SAE 81C91 in its P-LCC-28-1 package, the pads of the I/O ports are not bonded and therefore unavailable to the user.

### 3.11 Control Registers

The SFCAN circuit is controlled by registers. These can be accessed by the addresses listed below. If not otherwise noted in the detailed description, they are readable and writeable.

**Address Assignments: Control Registers**

| Register Name | Address | Function |
|---|---|---|
| BL1 | $00_H$ | Bit-length register 1 |
| BL2 | $01_H$ | Bit-length register 2 |
| OC | $02_H$ | Output-control register |
| BRP | $03_H$ | Baud-rate prescaler |
| RR1 | $04_H$ | Receive-ready register 1 |
| RR2 | $05_H$ | Receive-ready register 2 |
| RIM1 | $06_H$ | Receive-interrupt-mask register 1 |
| RIM2 | $07_H$ | Receive-interrupt-mask register 2 |
| TRS1 | $08_H$ | Transmit-request-set register 1 |
| TRS2 | $09_H$ | Transmit-request-set register 2 |
| IMSK | $0A_H$ | Interrupt-mask register |
| REC | $0C_H$ | Receive-error counter (for test only) |
| TEC | $0D_H$ | Transmit-error counter (for test only) |
| MOD | $10_H$ | Mode/status register |
| INT | $11_H$ | Interrupt register |
| CTRL | $12_H$ | Control register |
| CC | $14_H$ | Clock-control register |
| TCEC | $15_H$ | Transmit-check error counter |
| TCD | $16_H$ | Transmit-check data register |
| TRR1 | $18_H$ | Transmit-request-reset register 1 |
| TRR2 | $19_H$ | Transmit-request-reset register 2 |
| RRP1 | $1A_H$ | Remote-request-pending register 1 |
| RRP2 | $1B_H$ | Remote-request-pending register 2 |
| TSCL | $1C_H$ | Time-Stamp counter Low byte |
| TSCH | $1D_H$ | Time-Stamp counter High byte |
| P0PDR | $28_H$ | Port 0 port-direction register |
| P0PR | $29_H$ | Port 0 pin register |
| P0LR | $2A_H$ | Port 0 latch register |
| P1PDR | $2C_H$ | Port 1 port-direction register |
| P1PR | $2D_H$ | Port 1 pin register |
| P1LR | $2E_H$ | Port 1 latch register |

Addresses not listed above are not available and must not be written in initialisation mode.

### 3.11.1 Message-Memory Layout

**Time Stamp registers**

| Address | Function | |
|---|---|---|
| $30_H$ | Byte 0 | Time-Stamp 0 |
| $31_H$ | Byte 1 | |
| $32_H$ | Byte 0 | Time-Stamp 1 |
| $33_H$ | Byte 1 | |
| : | : | : |
| $3C_H$ | Byte 0 | Time-Stamp 6 |
| $3D_H$ | Byte 1 | |
| $3E_H$ | Byte 0 | Time-Stamp 7 |
| $3F_H$ | Byte 1 | |

**Descriptor Registers**

| Address | Function | |
|---|---|---|
| $40_H$ | Byte 0 | Descriptor 0 |
| $41_H$ | Byte 1 | |
| $42_H$ | Byte 0 | Descriptor 1 |
| $43_H$ | Byte 1 | |
| : | : | : |
| $5C_H$ | Byte 0 | Descriptor 14 |
| $5D_H$ | Byte 1 | |
| $5E_H$ | Byte 0 | Descriptor 15 |
| $5F_H$ | Byte 1 | |

**Data Registers**

| Address | Function | |
|---|---|---|
| $80_H$ | Byte 0 | Message 0 |
| $81_H$ | Byte 1 | |
| $82_H$ | Byte 2 | |
| $83_H$ | Byte 3 | |
| $84_H$ | Byte 4 | |
| $85_H$ | Byte 5 | |
| $86_H$ | Byte 6 | |
| $87_H$ | Byte 7 | |
| $88_H$ | Byte 0 | Message 1 |
| $89_H$ | Byte 1 | |
| $8A_H$ | Byte 2 | |
| $8B_H$ | Byte 3 | |
| $8C_H$ | Byte 4 | |
| $8D_H$ | Byte 5 | |
| $8E_H$ | Byte 6 | |
| $8F_H$ | Byte 7 | |
| : | : | : |
| $F8_H$ | Byte 0 | Message 15 |
| $F9_H$ | Byte 1 | |
| $FA_H$ | Byte 2 | |
| $FB_H$ | Byte 3 | |
| $FC_H$ | Byte 4 | |
| $FD_H$ | Byte 5 | |
| $FE_H$ | Byte 6 | |
| $FF_H$ | Byte 7 | |

### 3.11.2 Descriptor Registers

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **ID.10** | **ID.9** | **ID.8** | **ID.7** | **ID.6** | **ID.5** | **ID.4** | **ID.3** | Contents |

Descriptor byte 0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **ID.2** | **ID.1** | **ID.0** | **RTR** | **DLC.3** | **DLC.2** | **DLC.1** | **DLC.0** | Contents |

Descriptor byte 1

These two registers contain the eleven bits of the message identifier (ID.0 through ID.10), the remote-transmission-request bit (RTR) and the data-length-code (DLC.0 through DLC.3) of a message.

| **DLC.3** | **DLC.2** | **DLC.1** | **DLC.0** | **data length** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 byte |
| 0 | 0 | 0 | 1 | 1 byte |
| 0 | 0 | 1 | 0 | 2 byte |
| 0 | 0 | 1 | 1 | 3 byte |
| 0 | 1 | 0 | 0 | 4 byte |
| 0 | 1 | 0 | 1 | 5 byte |
| 0 | 1 | 1 | 0 | 6 byte |
| 0 | 1 | 1 | 1 | 7 byte |
| 1 | 0 | 0 | 0 | 8 byte |

All other bit combinations are not permitted.

**Note:** For the transmission of remote frames (RTR = 1) the data-length-code must be set like the DLC-bits of the corresponding data frame.

### 3.11.3 Time-Stamp Registers

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TS15** | **TS14** | **TS13** | **TS12** | **TS11** | **TS10** | **TS9** | **TS8** | Contents |

Time-Stamp byte 0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TS7** | **TS6** | **TS5** | **TS4** | **TS3** | **TS2** | **TS1** | **TS0** | Contents |

Time-Stamp byte 1

The two registers contain the time-stamp of the corresponding message. Both registers can only be read. For more information see chapter 3.9.

### 3.11.4 Mode/Status-Register MOD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **ADE** | **RS** | **TC** | **TWL** | **RWL** | **BS** | **RES** | **IM** | Contents |
| | | | | | | | | MOD |

The register can be read and written except otherwise noted; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| IM | MOD.0 | Init Mode<br>IM = 1: Setting this bit from 0 to 1 starts the initialisation mode:<br>– all bus activities are stopped<br>– all registers are set to their reset value<br>– access to read/write protected registers is enabled<br>If the bit stays set, the chip enters the normal mode, with enabled access to read/write protected registers.<br>IM = 0: Normal mode. |
| RES | MOD.1 | Reset Request<br>RES = 1: The chip enters the reset state:<br>– all bus activities are stopped<br>– all registers are set to their reset value<br>– the bus-off state is cancelled, if the bit is set for more than 1.5 x bit length<br>RES = 0: Normal mode |
| BS | MOD.2 | Bus State (read only)<br>BS = 1: Bus-off state, the IC does not participate in bus activities<br>BS = 0: Normal mode |
| RWL | MOD.3 | Receiver Warning Level (read only)<br>RWL = 1: Receive-error counter larger or equal 96<br>RWL = 0: Receive-error counter smaller 96 |
| TWL | MOD.4 | Transmit Warning Level (read only)<br>TWL = 1: Transmit-error counter larger or equal 96<br>TWL = 0: Transmit-error counter smaller 96 |
| TC | MOD.5 | Transmission Complete (read only)<br>TC = 1: The last transmission request was executed successfully<br>TC = 0: The last transmission request was not executed successfully or interrupted |
| RS | MOD.6 | Receive State (read only)<br>RS = 1: Currently a message is received<br>RS = 0: No receive |
| ADE | MOD.7 | Auto Decrement Enable<br>ADE = 1: With every read or write access using the serial synchronous interface SI the address is decremented by one automatically. So one can access data continuously without the need of writing a new address<br>ADE = 0: No automatic address decrement |

**Notes on Bit TC**

Scanning this bit is particularly useful if only one transmission is active. If there are several transmission jobs at the same time, it is better to scan the transmit-request register, because bit TC may possibly only be set very briefly between acknowledgement of the previous message and the start of the next one.

**Notes on Bit RES and IM**

There is a difference between a hardware reset (pin $\overline{\text{RES}}$ set to 0) and a software reset (bit RES set to 1). The bus-off state is cancelled immediately after a hardware reset. After a software reset the time of 128 idle phases is waited, determined by the CAN protocol (1 idle phase = 11 recessive bits in sequence).

The registers BL1, BL2, BRP and OC are set to $00_H$ after a hardware reset or after both bits RES and IM are set to 1 (reset state of the chip).

### 3.11.5 Control Register CTRL

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| – | **TST** | **TSP1** | **TSP0** | **TSOV** | **SME** | **TCE** | **MM** | Contents |
| | | | | | | | | CTRL |

The register can be read and written; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| MM | CTRL.0 | Monitor Mode<br>MM = 1: The memory location of message 0 receives all identifiers that are not accepted by other memory locations (corresponds to a Basic CAN receive register).<br>MM = 0: The above memory location responds like all others. |
| TCE | CTRL.1 | Transmit Check Enable<br>TCE = 1: If the transmit check detects an error, the message is invalidated by an error frame and the error counter TCEC is incremented by 1. If the counter reaches 4, the bus-off status is initiated and, if enabled, an interrupt (TCI) is generated.<br>TCE = 0: If the transmit check detects an error, there is no intervention. |
| SME | CTRL.2 | Sleep Mode Enable<br>SME = 1: The sleep mode is enabled: the crystal oscillator is deactivated, all other activities are inhibited.<br>The wake up is done also by a reset signal or by an active signal at the $\overline{CS}$ pin or by an input edge going from recessive to dominant at pin Rx0 respectively Rx1.<br>SME = 0: Normal operation. |
| TSOV | CTRL.3 | Time Stamp Overflow<br>TSOV = 1:There was at least one overflow of the time-stamp timer.<br>TSOV = 0:There has been no overflow. |
| TSP0<br>TSP1 | CTRL.4<br>CTRL.5 | Time Stamp Prescaler<br>TSP1 TSP0 Clock of time-stamp timer<br>0 0 $f_{BL}$ / 32<br>0 1 $f_{BL}$ / 64<br>1 0 $f_{BL}$ / 128<br>1 1 $f_{BL}$ / 256<br>For $f_{BL}$ see baud-rate prescaler BRP |
| TST | CTRL.6 | Time Stamp Test<br>TST = 1: The time-stamp prescaler is deactivated. Only for test, bit IM (MOD.0) must be set to 1.<br>TST = 0: The prescaler is activated. |
| – | CTRL.7 | Reserved, must be 0. |

### 3.11.6 Interrupt Register INT

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TCI** | **EPI** | **BOI** | **WUPI** | **RFI** | **WLI** | **TI** | **RI** | Contents |
| | | | | | | | | INT |

Register bits can be reset by writing 0 to the respective bit, writing 1 has no effect.

| Symbol | Position | Function |
|---|---|---|
| RI | INT.0 | Receive Interrupt<br>After a valid message has been received and filed, this bit is set and an interrupt generated. The bit will be set until all bits of the registers RR1 and RR2 are reset. |
| TI | INT.1 | Transmit Interrupt<br>This bit is set and an interrupt generated as soon as a transmit request has been processed. |
| WLI | INT.2 | Warning Level Interrupt<br>If at least one of the two error counters is greater than or equals 96, this bit is set and an interrupt generated. |
| RFI | INT.3 | Remote Frame Interrupt<br>This interrupt is generated after reception of a remote frame. |
| WUPI | INT.4 | Wake Up Interrupt<br>After a wake-up the bit is set and an interrupt generated. |
| BOI | INT.5 | Bus Off Interrupt<br>This bit is set and an interrupt generated in bus-off status. |
| EPI | INT.6 | Error Passive Interrupt<br>If at least one of the two error counters is greater than or equals 128, this bit is set and an interrupt generated. |
| TCI | INT.7 | Transmit Check Interrupt<br>If the transmit-check error counter reaches 4, this bit is set and an interrupt generated. |

All bits of this register must be reset by software.

**Note:** An interrupt is only generated, if the respective IMSK bit is set. The bits in this register are set independent of the IMSK-Register.

### 3.11.7 Bit-Length Register 1 BL1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **SAM** | **TS2.2** | **TS2.1** | **TS2.0** | **TS1.3** | **TS1.2** | **TS1.1** | **TS1.0** | Contents |
| | | | | | | | | BL1 |

The register can always be read but only written when bits IM (MOD.0) and RES (MOD.1) are set; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| TS1.0<br>TS1.1<br>TS1.2<br>TS1.3 | BL1.0<br>BL1.1<br>BL1.2<br>BL1.3 | Determine the length of segment 1 (TSEG1).<br><br>$TSEG1 = (8 \times TS1.3 + 4 \times TS1.2 + 2 \times TS1.1 + 1 \times TS1.0 + 1) \times t_{SCL}$ |
| TS2.0<br>TS2.1<br>TS2.2 | BL1.4<br>BL1.5<br>BL1.6 | Determine the length of segment 2 (TSEG2).<br><br>$TSEG2 = (4 \times TS2.2 + 2 \times TS2.1 + 1 \times TS2.0 + 1) \times t_{SCL}$ |
| SAM | BL1.7 | Sample Rate<br>SAM = 1: Input signal is sampled three times per bit.<br>SAM = 0: Input signal is sampled once per bit.<br><br>Note: The bit SAM should only be set to 1 using very low baud rates |

For $t_{SCL}$ see baud-rate prescaler BRP.

### 3.11.8 Bit-Length Register 2 BL2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **IPOL** | **DI** | – | – | – | **SM** | **SJW.1** | **SJW.0** | Contents |
| | | | | | | | | BL2 |

The register can always be read but only written when bits IM (MOD.0) and RES (MOD.1) are set; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| SJW.0<br>SJW.1 | BL2.0<br>BL2.1 | Determine the maximum Synchronisation Jump Width.<br>SJW = (2 × SJW.1 + 1 × SJW.0 + 1) × $t_{SCL}$ |
| SM | BL.2.2 | Speed Mode<br>Determines what edges are used for synchronisation.<br>SM = 1: Both edges are used.<br>SM = 0: Recessive to dominant is used.<br><br>Note: According to the CAN specification this bit should not be set to 1. |
| – | BL2.3 | Reserved, must be 0 (read value is not defined when IM is set). |
| – | BL2.4 | Reserved, must be 0 (read value is not defined when IM is set). |
| – | BL2.5 | Reserved, must be 0 (read value is not defined when IM is set). |
| DI | BL2.6 | Digital Input<br>DI = 1: The input signal on pin RX0 is evaluated digitally. The input comparator is inactive. Pin RX1 should be on $V_{SS.}$<br>DI = 0: The input signal is applied to the input comparator. |
| IPOL | BL2.7 | Input Polarity<br>IPOL = 1: The input level is inverted.<br>IPOL = 0: The input level remains unaltered. |

For $t_{SCL}$ see baud-rate prescaler BRP.

### 3.11.9 Interrupt-Mask Register IMSK

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **ETCI** | **EEPI** | **EBOI** | **EWUPI** | **ERFI** | **EWLI** | **ETI** | **ERI** | Contents |
| | | | | | | | | IMSK |

The register can be read and written; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| ERI | IMSK.0 | Enable Receive Interrupt<br>ERI = 1: Receive interrupts are enabled.<br>ERI = 1: No interrupt enabled. |
| ETI | IMSK.1 | Enable Transmit Interrupt<br>ETI = 1: Completed transmit jobs generate interrupts.<br>ETI = 0: No interrupt enabled. |
| EWLI | IMSK.2 | Enable Warning Level Interrupt<br>EWLI = 1: There is an interrupt when the warning level is reached.<br>EWLI = 0: No interrupt enabled. |
| ERFI | IMSK.3 | Enable Remote Frame Interrupt<br>ERFI = 1: A receive interrupt is generated after receiving a remote frame.<br>ERFI = 0: No interrupt enabled. |
| EWUPI | IMSK.4 | Enable Wake Up Interrupt<br>EWUPI = 1: Wake-up is enabled as interrupt.<br>EWUPI = 0: No interrupt enabled. |
| EBOI | IMSK.5 | Enable Bus-Off Interrupt<br>EBOI = 1: Bus off is enabled as interrupt.<br>EBOI = 0: No interrupt enabled. |
| EEPI | IMSK.6 | Enable Error Passive Interrupt<br>EEPI = 1: Error passive is enabled as interrupt.<br>EEPI = 0: No interrupt enabled. |
| ETCI | IMSK.7 | Enable Transmit Check Error Interrupt<br>ETCI = 1: Transmit-check error is enabled as interrupt.<br>ETCI = 0: No interrupt enabled. |

**Note:**The above bits control, if an event activates the INT pin. They dont influence the INT register.

### 3.11.10 Baud Rate Prescaler BRP

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| – | – | **BRP5** | **BRP4** | **BRP3** | **BRP2** | **BRP1** | **BRP0** | Contents |
| | | | | | | | | BRP |

The register is not readable and can only be written when bits IM (MOD.0) and RES (MOD.1) are set; the reset value is $00_H$.

| Symbol | Position | Function |
|---|---|---|
| BRP0<br>BRP1<br>BRP2<br>BRP3<br>BRP4<br>BRP5 | BRP.0<br>BRP.1<br>BRP.2<br>BRP.3<br>BRP.4<br>BRP.5 | Baud Rate Prescaler<br>This register determines the system clock<br>$t_{SCL}$ = (32 × BRP.5 + 16 × BRP.4 + 8 × BRP.3 + 4 × BRP.2 + 2 × BRP.1 + 1 × BRP.0 + 1) × $2t_{OSC}$ |
| – | BRP.6 | Reserved, must be 0 |
| – | BRP.7 | Reserved, must be 0 |

$t_{OSC}$ = 1/quartz clock

The bit length $t_{BL}$ is computed as follows:

$$t_{BL} = T_{SEG1} + T_{SEG2} + 1\ t_{SCL}$$

The baudrate BR is:

$$BR = f_{crystal} / (2 \times (BRP + 1) \times (TS1 + TS2 + 3))$$

**Note:**

BRP = BRP0..BRP5 see Baud Rate Prescaler
TS1 = TS1.0..TS1.3 see Bit Length Register 1
TS2 = TS2.0..TS2.2 see Bit Length Register 1

### 3.11.11 Output-Control Register OC

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **OCTP.1** | **OCTN.1** | **OCP.1** | **OCTP.0** | **OCTN.0** | **OCP.0** | **OCM.1** | **OCM.0** | Contents |
| | | | | | | | | OC |

The register can always be read but only written when bits IM (MOD.0) and RES (MOD.1) are set; the reset value is $00_H$.

**Output Modes**

| OCM.1 | OCM.0 | Output Mode |
|---|---|---|
| 0 | X | Normal Mode: TX0 = Bit Sequence, TX1 = Bit Sequence |
| 1 | 0 | Test Mode: TX0 = Bit Sequence, TX1 = RX0 |
| 1 | 1 | Clock Mode: TX0 = Bit Sequence, TX1 = Bit Clock |

**Output Programming**

| OCTP.x | OCTN.x | OCP.x | Data | TxP | TxN | TXx-Level |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | OFF | OFF | float |
| 0 | 0 | 0 | 1 | OFF | OFF | float |
| 0 | 0 | 1 | 0 | OFF | OFF | float |
| 0 | 0 | 1 | 1 | OFF | OFF | float |
| 0 | 1 | 0 | 0 | OFF | ON | LOW |
| 0 | 1 | 0 | 1 | OFF | OFF | float |
| 0 | 1 | 1 | 0 | OFF | OFF | float |
| 0 | 1 | 1 | 1 | OFF | ON | LOW |
| 1 | 0 | 0 | 0 | OFF | OFF | float |
| 1 | 0 | 0 | 1 | ON | OFF | HIGH |
| 1 | 0 | 1 | 0 | ON | OFF | HIGH |
| 1 | 0 | 1 | 1 | OFF | OFF | float |
| 1 | 1 | 0 | 0 | OFF | ON | LOW |
| 1 | 1 | 0 | 1 | ON | OFF | HIGH |
| 1 | 1 | 1 | 0 | ON | OFF | HIGH |
| 1 | 1 | 1 | 1 | OFF | ON | LOW |

TxP is the output transistor switching to $V_{DD}$, TxN switches to $V_{SS}$.
TXx is the output level at the transmit pin.

### 3.11.12 Receive-Ready Registers RR1 and RR2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RR7** | **RR6** | **RR5** | **RR4** | **RR3** | **RR2** | **RR1** | **RR0** | Contents |
| | | | | | | | | RR1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RR15** | **RR14** | **RR13** | **RR12** | **RR11** | **RR10** | **RR9** | **RR8** | Contents |
| | | | | | | | | RR2 |

The register bits can be reset by writing 0 to the respective bit, writing 1 has no effect; the reset value is $00_H$.

Bit RRx is set when a message has arrived and been written into the memory location of message x. Setting this bit by hardware can generate a receive interrupt, which can be blocked by bit RIMx in the receive-interrupt-mask register.

Bits RRx must be reset by software.

### 3.11.13 Receive-Interrupt-Mask Registers RIM1 and RIM2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RIM7** | **RIM6** | **RIM5** | **RIM4** | **RIM3** | **RIM2** | **RIM1** | **RIM0** | Contents |
| | | | | | | | | RIM1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RIM15** | **RIM14** | **RIM13** | **RIM12** | **RIM11** | **RIM10** | **RIM9** | **RIM8** | Contents |
| | | | | | | | | RIM2 |

These registers can be read and written; the reset value is $00_H$.

Setting bit RIMx enables a receive interrupt to be generated if the receive-ready bit RRx has been set, i.e. a message has arrived and was written into the memory location of message x.

Bit ERI in the interrupt-mask register IM blocks all receive interrupts, even if bits RIMx are set.

### 3.11.14 Transmit-Request Registers TR1 and TR2

The transmit-request registers are each divided in the addresses for setting and resetting of the transmission request. In this manner it is prevented that when writing these registers bits become set again which meanwhile were reset because of a completed transmission.

#### 3.11.14.1 Transmit-Request-Set Registers TRS1 and TRS2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TRS7** | **TRS6** | **TRS5** | **TRS4** | **TRS3** | **TRS2** | **TRS1** | **TRS0** | Contents |
| | | | | | | | | TRS1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TRS15** | **TRS14** | **TRS13** | **TRS12** | **TRS11** | **TRS10** | **TRS9** | **TRS8** | Contents |
| | | | | | | | | TRS2 |

These registers can be read. Writing 1 to a bit takes effect, writing 0 has no effect; the reset value is $00_H$.

Setting bit TRx causes the particular message x to be transmitted. The bit is reset by hardware after transmission. Several bits can be set simultaneously. In this way all messages whose request bits are set are transmitted in turn, starting with the memory location with the highest number.

#### 3.11.14.2 Transmit-Request-Reset Registers TRR1 and TRR2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TRR7** | **TRR6** | **TRR5** | **TRR4** | **TRR3** | **TRR2** | **TRR1** | **TRR0** | Contents |
| | | | | | | | | TRR1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TRR15** | **TRR14** | **TRR13** | **TRR12** | **TRR11** | **TRR10** | **TRR9** | **TRR8** | Contents |
| | | | | | | | | TRR2 |

These registers can not be read and only writing 1 to a bit takes effect.

Setting bit TRRx causes a transmission request, initiated by the corresponding bit TRSx, to be cancelled, provided that it is not currently processed. Bit TRSx is set to 0 by this action.

### 3.11.15 Remote-Request-Pending Registers RRP1 and RRP2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RRP7** | **RRP6** | **RRP5** | **RRP4** | **RRP3** | **RRP2** | **RRP1** | **RRP0** | Contents |
| | | | | | | | | RRP1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **RRP15** | **RRP14** | **RRP13** | **RRP12** | **RRP11** | **RRP10** | **RRP9** | **RRP8** | Contents |
| | | | | | | | | RRP2 |

These registers can only be read; the reset value $00_H$.

If bit RRPx by is set a remote frame was received for the particular message x, and could not be answered yet by the transmission of the corresponding message.

### 3.11.16 Port Register

**Port-Latch Registers**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | Contents |

P0LR, P1LR

**Port-Pin Registers**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | Contents |

P0PR, P1PR

**Port-Direction Registers**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | Contents |

P0PDR, P1PDR

Except the port-pin registers, which can only be read, all registers can be read and written; the reset value of the port registers is $00_H$.

After a reset, the ports are switched as inputs. The state of the pin can be read via the port register. Setting a bit of the port-direction register to 1 causes the associated port pin to be switched as an output. The output data can then be written (latched) into the port register. The latch can be read from the port-latch register, the levels on the port pins from the port-pin register,

In parallel to the standard CMOS structure there is an additional internal pull up resistor of about 250 kΩ at each port pin.

### 3.11.17 Clock-Control Register CC

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| – | – | – | – | **CC3** | **CC2** | **CC1** | **CC0** | Contents |
| | | | | | | | | CC |

This register can only be written; the reset value is $01_H$.

This register determines the output freuency at pin CLK.

| **CC3** | **CC2** | **CC1** | **CC0** | **output frequency** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $f_{osc}$ |
| 0 | 0 | 0 | 1 | $f_{osc}/2$ |
| 0 | 0 | 1 | 0 | $f_{osc}/4$ |
| 0 | 0 | 1 | 1 | $f_{osc}/6$ |
| 0 | 1 | 0 | 0 | $f_{osc}/8$ |
| 0 | 1 | 0 | 1 | $f_{osc}/10$ |
| 0 | 1 | 1 | 0 | $f_{osc}/12$ |
| 0 | 1 | 1 | 1 | $f_{osc}/14$ |
| 1 | X | X | X | switched off (Low level) |

Bits 4 through 7 must be 0.

### 3.11.18 Transmit-Check Error Counter TCEC

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| – | – | – | – | – | **TCEC2** | **TCEC1** | **TCEC0** | Contents |
| | | | | | | | | TCEC |

This register can be read and written; the reset value is $00_H$.

This register counts the errors detected by the transmit check. When a count of 4 is reached (TCEC = 1), an interrupt is generated if enabled. Furthermore, if the bit TCE (CTRL.1) is set to 1, the bus-off status is initiated.

Bits 3 through 7 must be 0.

### 3.11.19 Transmit-Check Data Register TCD

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Contents |
| | | | | | | | | TCD |

This register can only be read; the reset value is undefined.

This register, when a transmit-check error appears, contains the data byte that led to the error. By reading the contents of the register one can see which byte actually being sent. This way an error analysis can be attempted.

### 3.11.20 Time-Stamp Counter TSCL and TSCH

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | Contents |
| | | | | | | | | TSCL |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **D15** | **D14** | **D13** | **D12** | **D11** | **D10** | **D9** | **D8** | Contents |
| | | | | | | | | TSCH |

Via these registers the time-stamp counter can be read and written, the reset value is $00_H$.

### 3.11.21 Receive and Transmit Error Counter REC and TEC

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **REC7** | **REC6** | **REC5** | **REC4** | **REC3** | **REC2** | **REC1** | **REC0** | Contents |
| | | | | | | | | REC |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit |
|---|---|---|---|---|---|---|---|---|
| **TEC7** | **TEC6** | **TEC5** | **TEC4** | **TEC3** | **TEC2** | **TEC1** | **TEC0** | Contents |
| | | | | | | | | TEC |

Via these registers the receive and transmit error counters can be read, provided that the bit IM (MOD.0) is set to 1, the reset value is $00_H$.

## 4 Communication and Protocol

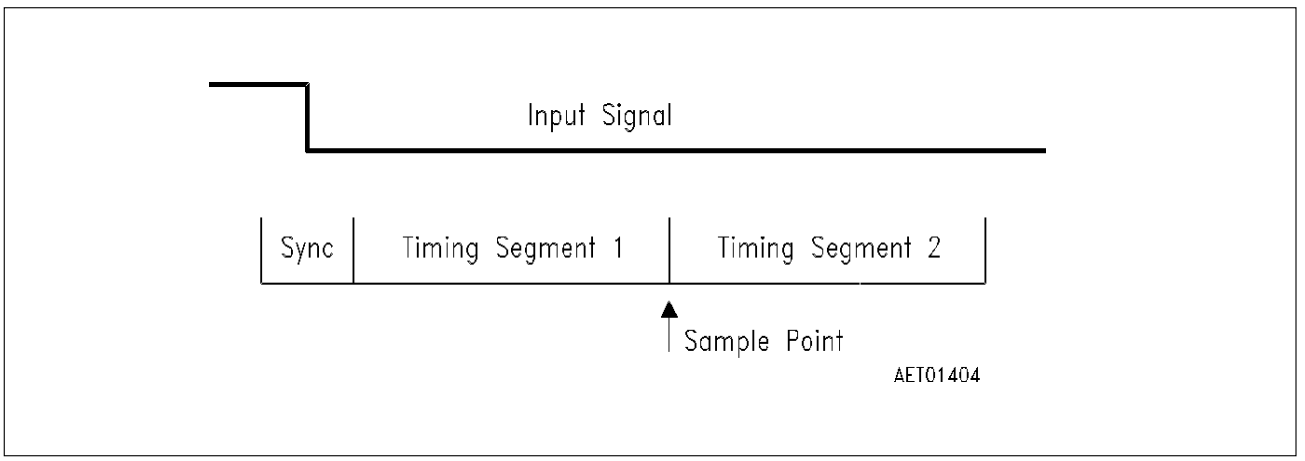
Details of this item will be added later. But the following points should be dealt with.

### 4.1 Bit Timing

A regular period is composed of the following three segments

- synchronisation segment,
- timing segment-1,
- timing segment 2.

Between timing segment 1 and timing segment 2 there is the sampling point.




**Synchronisation**

The edge of the input signal is expected during the sync segment (duration = 1 system clock cycle = 1 $t_{SCL}$).

**Timing Segment 1**

Timing segment 1 determines the sampling point within a bit period. This point is always at the end of segment 1. The segment is programmable from 1 to 16 $t_{SCL}$ (see bit-length register BL1).

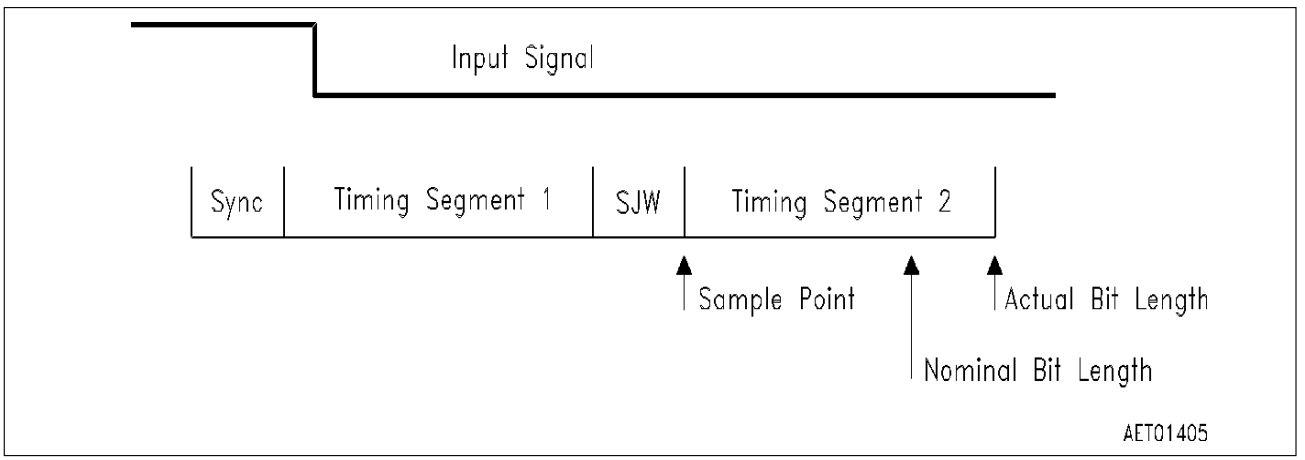
**Timing Segment 2**

Timing segment 2 provides extra time for internal processing after the sampling point. The segment is programmable from 1 to 8 $t_{SCL}$ (see bit-length register BL1).

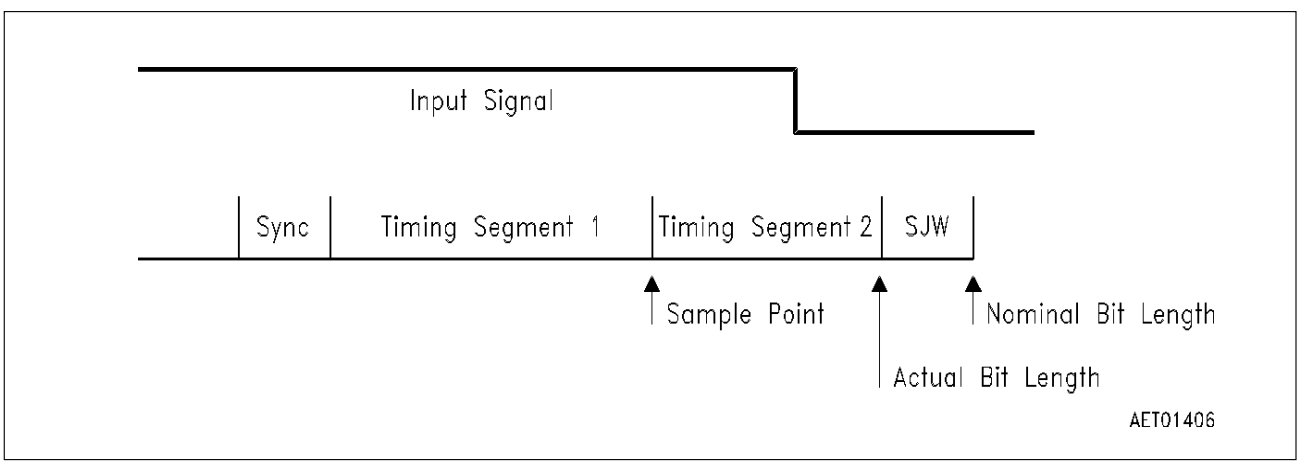
**Synchronisation Jump Width**

To compensate for phase shifts between the oscillator frequencies of the different bus stations, each CAN controller must be able to synchronise to the relevant signal edge of the incoming signal. The synchronisation jump width (SJW) determines the maximum number of system clock pulses by which the bit period can be lengthened or shortened for resynchronisation. The synchronisation jump width is programmable from 1 to 4 $t_{SCL}$ (see bit-length register BL2).

**Example of Lengthening Bit Period:**




**Example of Shortening Bit Period:**




**Delay Times**

The total delay is calculated from the following single delays:

- 2 × physical bus $t_{Bus}$ (max. 100 ns acc. to CAN specification),
- 2 × input comporator $t_{Comp}$ (depends on application circuit),
- 2 × output driver $t_{Driver}$ (depends on application circuit),
- 1 × input to output of CAN controller $t_{InOut}$ (max. 1 $t_{SCL}$ + 80 ns)

$$t_{Delay} = 2 \times (t_{Bus} + t_{Comp} + t_{Driver}) + t_{InOut}$$

**Recommendations**

On the basis of the stated conditions, there are the following essential requirements to be maintained:

$t_{TSEG1} \geq t_{SEG2}$
$t_{TSEG1} \geq t_{Delay}$
$t_{TSEG2} > t_{SJW}$
$t_{TSEG1} = 2 \times t_{TSEG2}$
if bit SAM = 1 then $T_{SEG2} \geq 3 \times t_{SCL}$ + SJW

## 5 Host Interfaces

There are two different host interfaces implemented in the SAE 81C90 / SAE 81C91.

Data and addresses on a multiplexed 8-bit bus, compatible with Siemens microcontrollers (SAB 80C5xx, SAB 80C16x), can be transferred via the parallel interface (PI). Using the serial synchronous interface (SI), any host controller with a serial three-lead interface can be connected with.

The interface is selected by hardware through the wiring of the MS (Mode Select) pin. This pin may not be switched during operation. If there is a High level on the MS pin, the SI and thus pins DI, DO, CLK, $\overline{W}$ and TIM are activated, while pins AD5 through AD7, $\overline{RD}$, $\overline{WR}$ and ALE are inactive. A Low level on the MS pin switches to the Pl and thus activates pins AD0 through AD7, $\overline{RD}$, $\overline{WR}$ and ALE.

### 5.1 Parallel Interface Pl

The parallel interface consists of a multiplexed 8-bit address/data bus. First the address of the required register is applied to the pins AD0 through AD7. A falling edge on pin ALE means that this address is transferred to an on-chip latch. After this, data can either be written into the selected register (pin $\overline{WR}$ = 0) or read from it (pin $\overline{RD}$ = 0) on the address/data bus. Pin $\overline{CS}$ must be 0 for the entire duration of the $\overline{RD}$/$\overline{WR}$ active time so that the circuit is activated.

### 5.2 Serial Synchronous Interface Sl

If the SI is used, the unused pins of the PI must be set to inactive levels ($\overline{RD}$, $\overline{WR}$ to $V_{DD}$ and ALE, AD5, AD6, AD7 to $V_{SS}$).

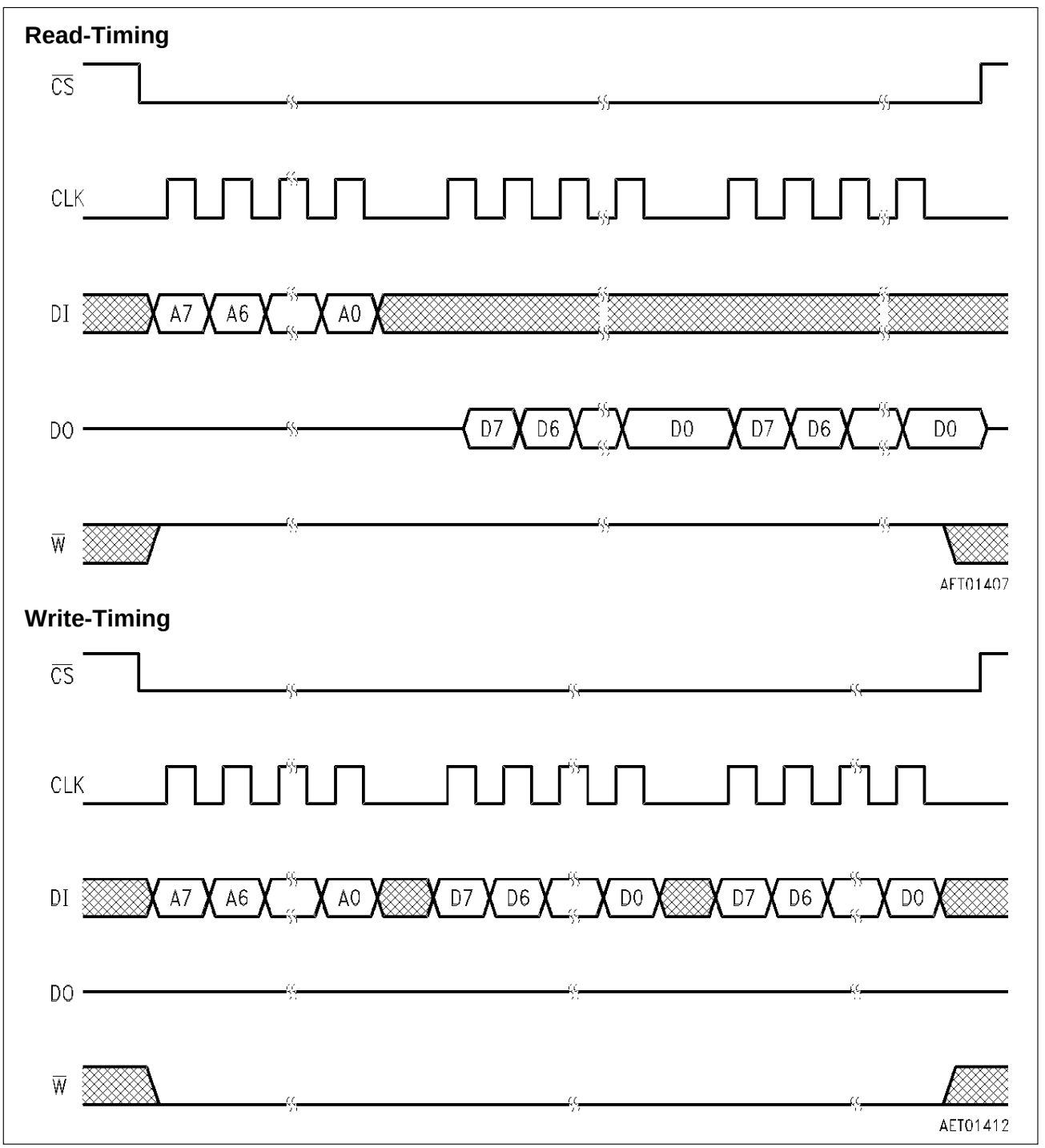
Communication on the SI is according to the following procedure:

Each access to the stand-alone Full-CAN circuit has to be started by activating the device ($\overline{CS}$ = 0). After the beginning of access, an address must be written first and then data can be read or written. The required function is determined by pin $\overline{W}$ ($\overline{W}$ = 1: read; $\overline{W}$ = 0: write). If the automatic decrementing of the address is activated (bit ADE in the MOD register), any number of data bytes can be accessed in succession. Finally the device has to be deactivated.

**Procedure:**

- Activate device ($\overline{CS}$ = 0)
- Set pin $\overline{W}$ to 1 for read, to 0 for write
- Write in address of first data byte
- Read out/write in one or more data bytes
- Deactivate device ($\overline{CS}$ = 1)

The most-significant bit is always output as the first bit of an address or a data byte.
Data from pin DI are transferred into the internal shift register with the **raising** edge of the clock. The active clock edge of pin DO is selectable via the pin TIM. If this pin is 0 the data are output from the shift register to pin DO with the **raising** clock edge (Timing A). If the pin TIM is 1, the output of data is done with the **falling** edge (Timing B).
The timing for the reading and writing of two data bytes with the automatic incrementing activated is illustrated below.




**SI timing (2 Data Bytes)**

## 6 Absolute Maximum Ratings

$T_A$ = – 40 ˚C to + 110 ˚C

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Supply voltage | $V_{DD}/V_{DDA}$ | – 0.5 | 6.0 | V |
| Input voltage | $V_I$ | – 0.5 | $V_{DD}$ + 0.5 | V |
| Open voltage | $V_O$ | $V_{SS}$ | $V_{DD}$ | V |
| Storage temperature | $T_{stg}$ | – 50 | 150 | ˚C |

**Operating Range**

| | | | | |
|---|---|---|---|---|
| Supply voltage | $V_{DD}/V_{DDA}$ | 4.5 | 5.5 | V |
| Operating current | $I_{DD}$ | | 15 | mA |
| Operating temperature | $T_A$ | – 40 | 110 | ˚C |
| Clock frequency | $f_{OSC}$ | 0 | 20 | MHz |

### 6.1 DC Characteristics

| Parameter | Symbol | Limit Values | | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | min. | typ. | max. | | |
| **Inputs** | | | | | | |
| Port0 (P00 … P07) | | | | | | |
| Port1 (P10 … P17) | | | | | | |
| AD0 … AD7 | | | | | | |
| DI, DO, CLK, $\overline{W}$, TIM | | | | | | |
| $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, ALE | | | | | | |
| MS, RX0 (digital mode) | | | | | | |
| H input voltage | $V_{IH}$ | $0.7 \times V_{DD}$ | | $V_{DD}$ | V | |
| L input voltage | $V_{IL}$ | 0 | | $0.3 \times V_{DD}$ | V | |
| Input capacitance | $C_I$ | | 10 | | pF | |
| Input current | $I_I$ | 0 | | 1 | µA | $0\ V < V_{IN} < V_{DD}$ |

The pins of Port 0 and Port 1 are provided with a pull-up resistor of about 50 kΩ.

| **Input** | | | | | | |
|---|---|---|---|---|---|---|
| $\overline{RES}$ | | | | | | |
| H input voltage | $V_{IH}$ | $0.7 \times V_{DD}$ | | $V_{DD}$ | V | |
| L input voltage | $V_{IL}$ | 0 | | $0.3 \times V_{DD}$ | V | |
| Input capacitance | $C_I$ | | 10 | | pF | |
| Input current | $I_I$ | 0 | | 1 | µA | $0\ V < V_{IN} < V_{DD}$ |
| Reset pulse width | $t_{RES}$ | 2 | | | $1/f_{OSC}$ | |

The input $\overline{RES}$ has Schmitt-Trigger characteristic.

| **Inputs** | | | | | | |
|---|---|---|---|---|---|---|
| XTAL1 | | | | | | |
| XTAL2 | | | | | | |
| H input voltage | $V_{IH}$ | $V_{DD} - 1.0$ | | $V_{DD}$ | V | |
| L input voltage | $V_{IL}$ | 0 | | 0.5 | V | |
| Input capacitance | $C_I$ | | | 10 | pF | |
| Low-end capacitance | $C_L$ | 27 | | 68 | pF | |

The size of the low-end capacitance must correspond to the specification of the crystal producer. For external clocking pin X1 must be unconnected. The external digital signal, which has to be input at pin X2, must have a duty cycle of 1:1 (± 5 %)

**DC Characteristics** (cont'd)

| Parameter | Symbol | Limit Values | | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | min. | typ. | max. | | |
| **Comparator Inputs** | | | | | | |
| RX0, RX1 | | | | | | |
| Input voltage | $V_I$ | 0.5 | | $V_{DD}$ + 0.5 | V | |
| Common mode voltage*) | $V_{ICOM}$ | 1.5 | | $V_{DD}$ – 1.5 | V | |
| Input capacitance*) | $C_I$ | | | 10 | pF | |
| Input current | $I_L$ | | | 1 | µA | |
| Hysteresis*) | $V_{HYS}$ | | 100 | | mV | |
| Offset voltage*) | $V_{Off}$ | | 100 | | mV | |

**Note:** If the bus lines work according to the ISO specification, additional circuitry is necessary for interconnection of the input comparator to the bus lines.

| | | | | | | |
|---|---|---|---|---|---|---|
| **Outputs** | | | | | | |
| Port0 (P00 … P07) | | | | | | |
| Port1 (P10 … P17) | | | | | | |
| AD0 … AD7 | | | | | | |
| $\overline{INT}$ | | | | | | |
| H output voltage | $V_{OH}$ | $0.8 \times V_{DD}$ | | $V_{DD}$ | V | $I_{OH}$ = 1.6 mA |
| L output voltage | $V_{OL}$ | 0 | | $0.2 \times V_{DD}$ | V | $I_{OL}$ = 1.6 mA |
| Rise time*) | $t_{QR}$ | | | 40 | ns | $C_L$ = 30 pF |
| Fall time*) | $t_{QF}$ | | | 40 | ns | $C_L$ = 30 pF |
| **Output** | | | | | | |
| CLKOUT | | | | | | |
| H output voltage | $V_{OH}$ | $0.8 \times V_{DD}$ | | $V_{DD}$ | V | $I_{OH}$ = 10 mA |
| L output voltage | $V_{OL}$ | 0 | | 0.4 | V | $I_{OL}$ = 10 mA |
| Rise time*) | $t_{QR}$ | | | 20 | ns | $C_L$ = 50 pF |
| Fall time*) | $t_{QF}$ | | | 20 | ns | $C_L$ = 50 pF |
| **Driver Outputs** | | | | | | |
| TX0, TX1 | | | | | | |
| Source output current | $I_{SRC}$ | 5 | | | mA | $V_O = 0.8 \times V_{DD}$ |
| Sink output current | $I_{SNK}$ | 5 | | | mA | $V_O = 0.2 \times V_{DD}$ |

*) are not tested

## 6.2 AC Characteristics

The AC characteristics are insured over the entire operating range.

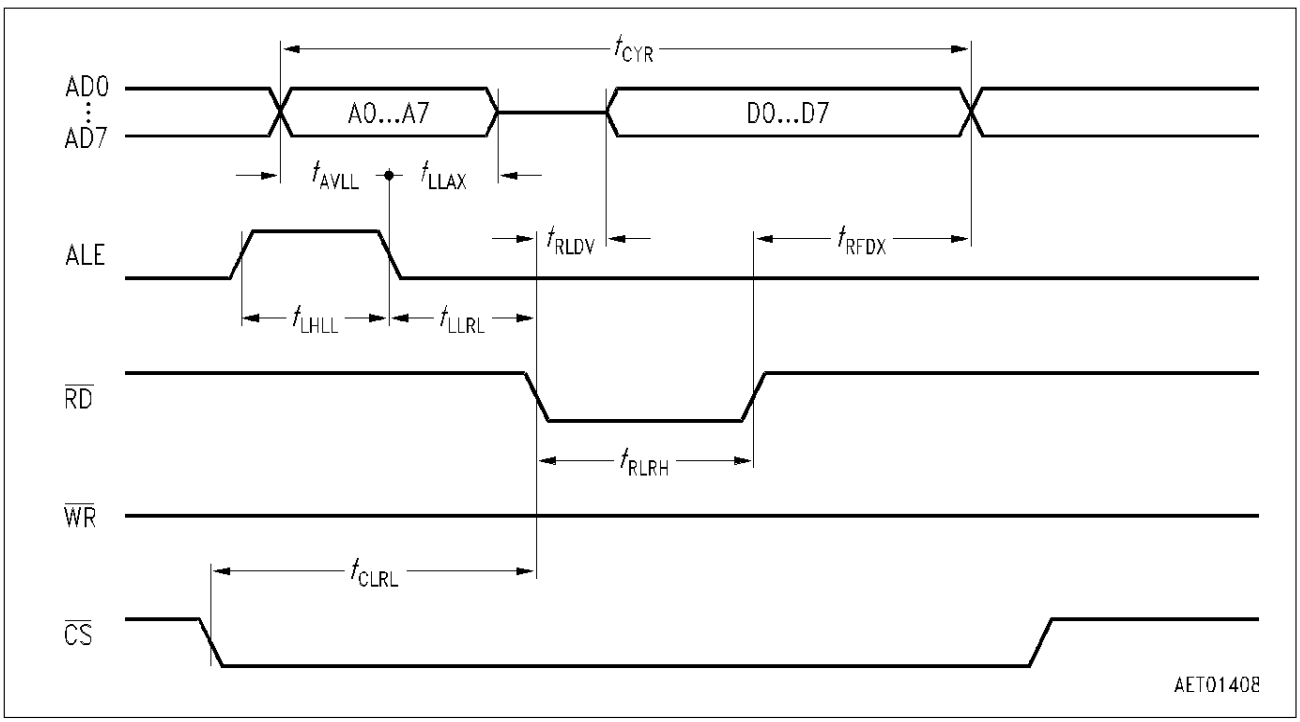
**PI Characteristics** $C_L$ = 30 pF

| Parameter | Symbol | Limit Values | | Unit |
|---|---|---|---|---|
| | | min. | max. | |
| Read-Cycletime | $t_{CYR}$ | 4 $t_{OSC}$ | | ns |
| Write-Cycletime | $t_{CYW}$ | 4 $t_{OSC}$ | | ns |
| ALE pulse width | $t_{LHLL}$ | 30 | | ns |
| Address setup to ALE low | $t_{AVLL}$ | 20 | | ns |
| Address hold after ALE low | $t_{LLAX}$ | 20 | | ns |
| $\overline{RD}$ pulse width | $t_{RLRH}$ | 2 $t_{OSC}$ + 30 | | ns |
| $\overline{WR}$ pulse width | $t_{WLWH}$ | 2 $t_{OSC}$ + 30 | | ns |
| ALE low to $\overline{WR}$ active | $t_{LLWL}$ | 20 | | ns |
| ALE low to $\overline{RD}$ active | $t_{LLRL}$ | 20 | | ns |
| Data float after $\overline{RD}$ high | $t_{RFDX}$ | 0 | 20 | ns |
| $\overline{RD}$ low to data valid | $t_{RLDV}$ | | 2 $t_{OSC}$ + 20 | ns |
| Data setup before $\overline{WR}$ high | $t_{QVWH}$ | 30 | | ns |
| Data hold after $\overline{WR}$ high | $t_{WHQX}$ | 12 | | ns |
| $\overline{CS}$ low to $\overline{RD}$ low | $t_{CLRL}$ | 0 | | ns |
| $\overline{CS}$ low to $\overline{WR}$ low | $t_{CLWL}$ | 0 | | ns |

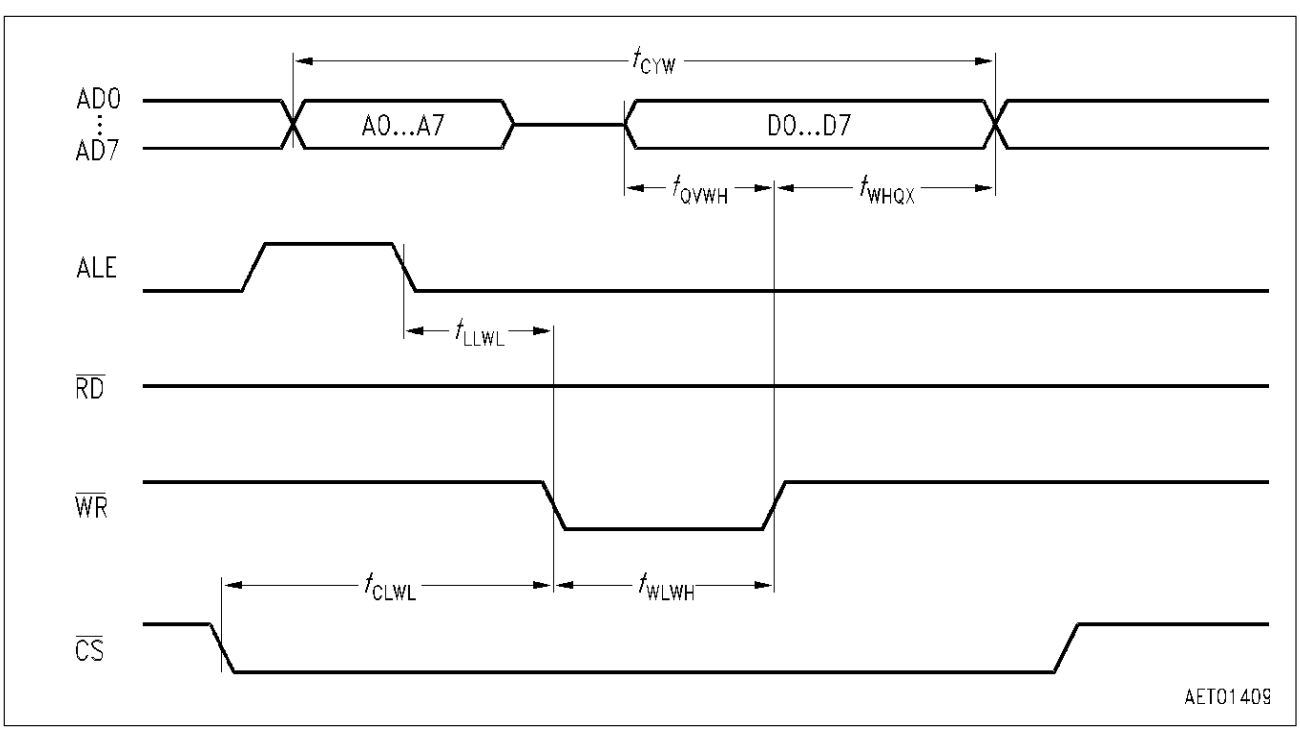
**SI Characteristics** $C_L$ = 30 pF

| | | | | |
|---|---|---|---|---|
| Chip Select Setup | $t_{CSS}$ | 10 | | ns |
| Clock High Time | $t_{CH}$ | 1.5 $t_{OSC}$ + 10 | | ns |
| Clock Low Time | $t_{CL}$ | 1.5 $t_{OSC}$ + 10 | | ns |
| Clock Period | $t_C$ | 4 $t_{OSC}$ | | ns |
| DI Setup | $t_{DIS}$ | 10 | | ns |
| DI Hold | $t_{DIH}$ | 0 | | ns |
| Address to Data Out | $t_{ADO}$ | 3 $t_{OSC}$ | | ns |
| Output Delay | $t_{OD}$ | | 25 | ns |
| Data Float after $\overline{CS}$ high | $t_{DF}$ | | 25 | ns |
| Chip Select Hold | $t_{CSH}$ | 1 $t_{OSC}$ | | ns |
| Write to Clock | $t_{WC}$ | 0 | | ns |
| $\overline{W}$ to $\overline{CS}$ high | $t_{WCS}$ | 0 | | ns |
| Address to Data In | $t_{ADI}$ | 0 | | ns |

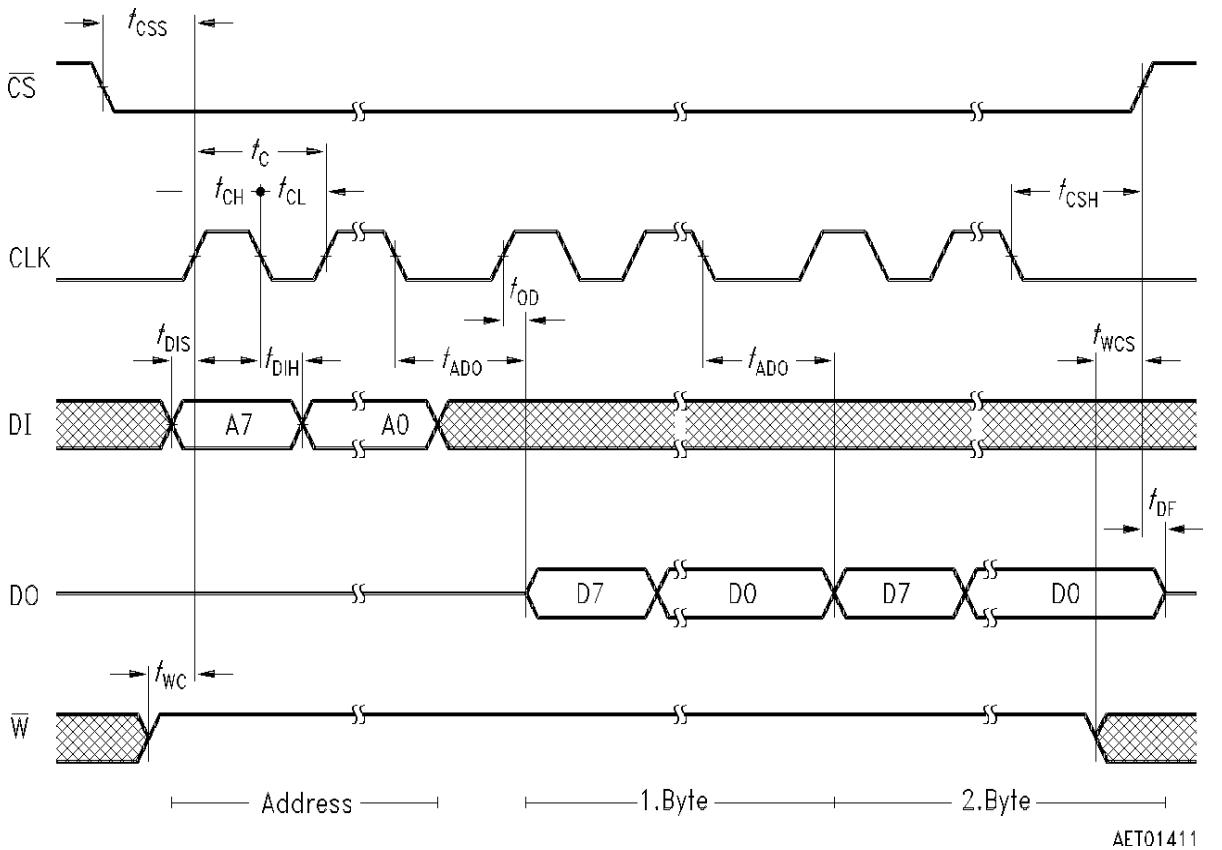
## 7 PI Timing



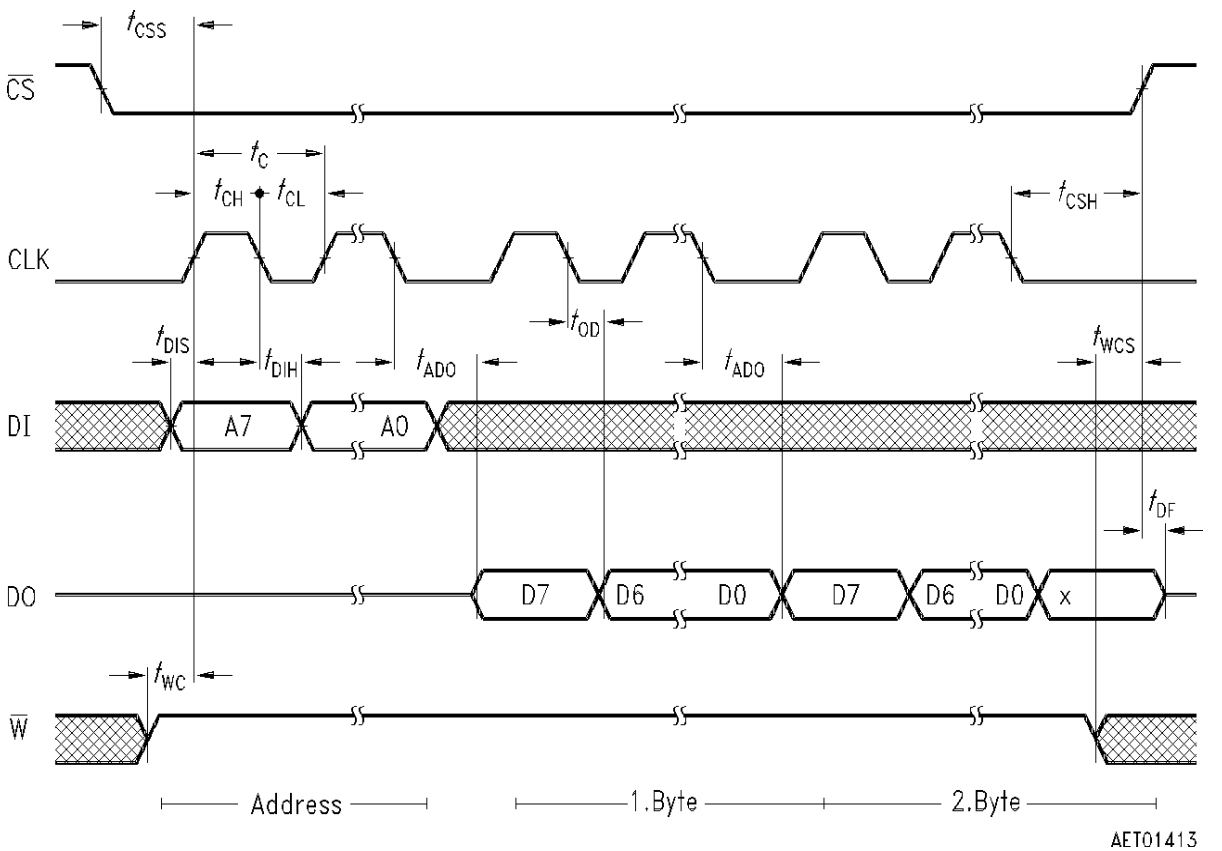

**Read-Cycle-Timing**



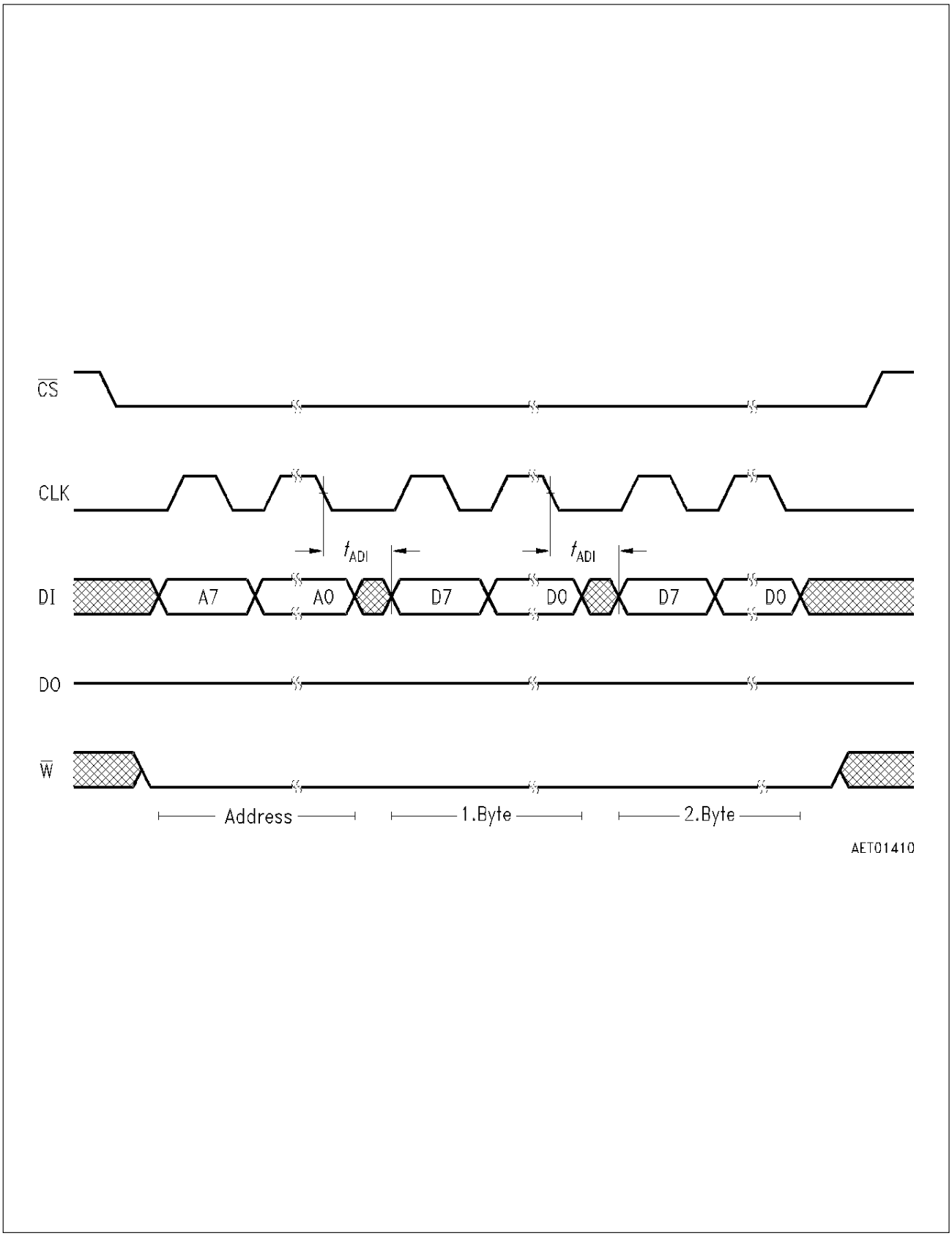

**Write-Cycle-Timing**




**SI-Read-Timing**
(Timing A: Pin TIM = 0)




**SI-Read-Timing**
(Timing B: Pin TIM = 1)

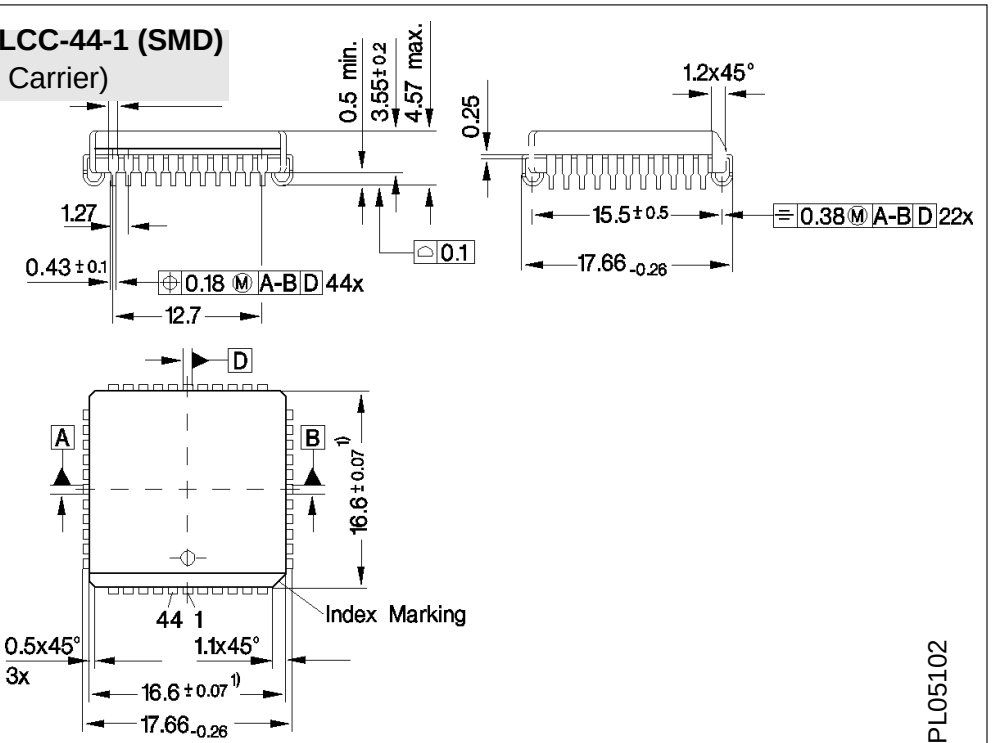



**SI-Write-Timing**

## 8 Package Outlines

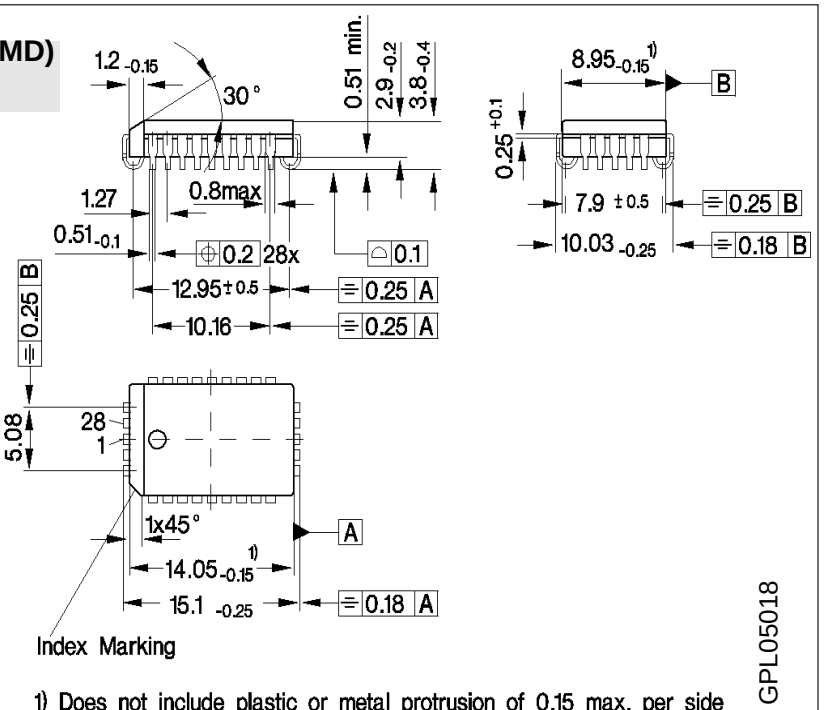
The SFCAN-IC is available in a P-LCC-44-1 package named as SAE 81C90 and in a P-LCC-28-1 package named as SAE 81C91. In the last package the pads of the I/O-ports are not bonded and so they are not available.

**Plastic-Package, P-LCC-44-1 (SMD)**
(Plastic Leaded Chip Carrier)




1) Does not include plastic or metal protrusion of 0.15 max. per side

**Plastic-Package, P-LCC-28-1 (SMD)**
(Plastic Leaded Chip Carrier)




1) Does not include plastic or metal protrusion of 0.15 max. per side

SMD = Surface Mounted Device

Dimensions in mm