

**SIEMENS**

# Microcomputer Components

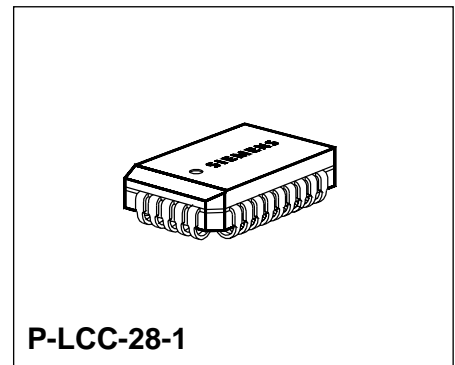
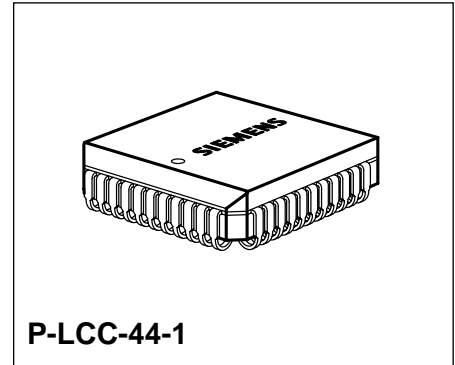
Standalone Full-CAN Controller

SAE 81C90/91

## Stand Alone Full CAN Controller

SAE 81C90/91

- 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 simultaneous (each with maximum data length)
- Message identifier reprogrammable "on the fly"
- 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 package only)



The device comes in two versions:

**SAE 81C90** in a P-LCC-44-1 package with two 8-bit I/O ports, and

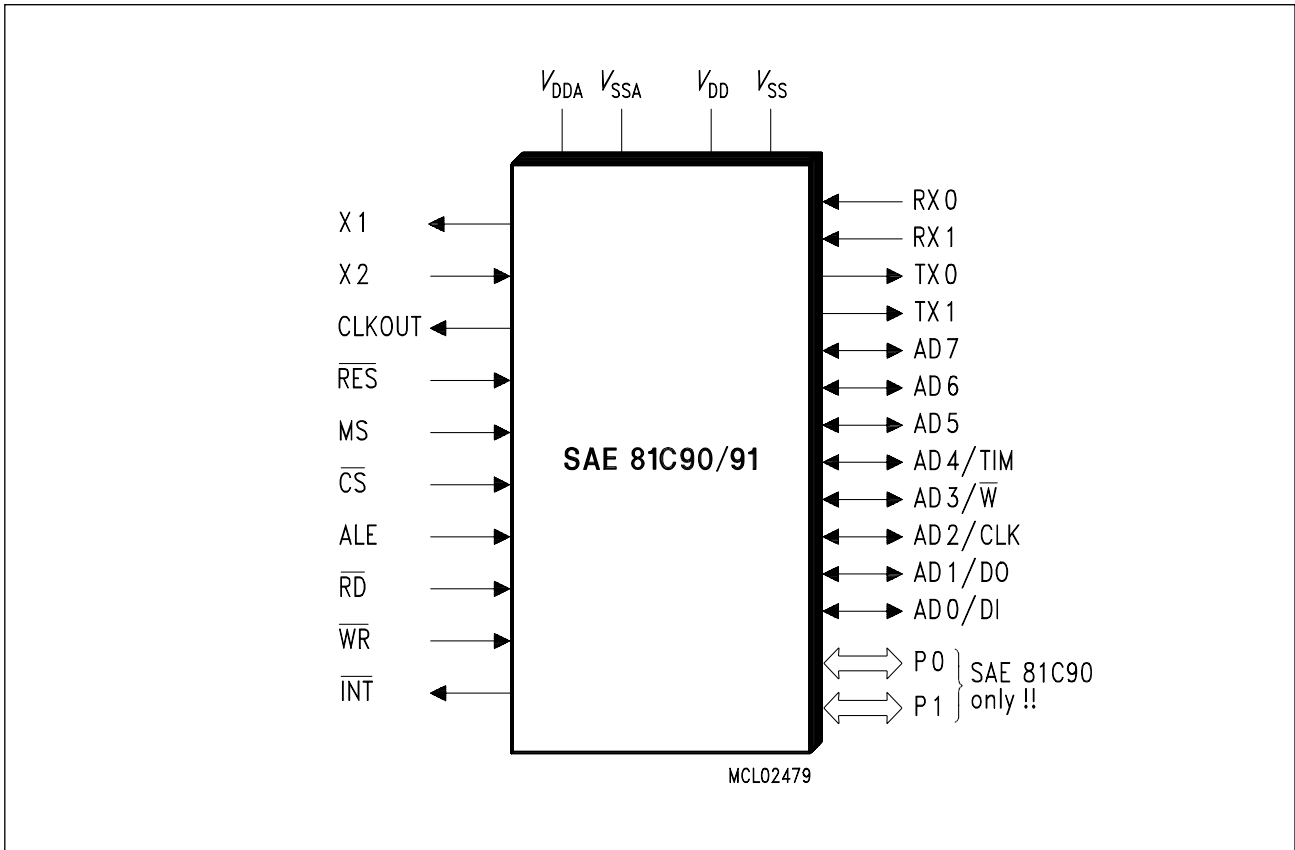
**SAE 81C91** in a P-LCC-28-1 package without I/O ports.

<b>SAE 81C90/91</b>	
<b>Revision History:</b>	<b>Version 01.97</b>
Previous Releases:	06.95 05.94 (Copy version)
Page	Subjects
1056	Figure 1 corrected.
1057	Figure 2 corrected.
1059	Notes updated.
1065 - 1084	Register description and arrangement improved.
1066, 1069	New register maps.
1093	$t_{AVLL}$ , $t_{LLAX}$ , $t_{DVWH}$ changed to 10 ns.
1093	$t_{WHDX}$ changed to 5 ns.

Controller Area Network (CAN): License of Robert Bosch GmbH

**Introduction**

The Siemens Stand Alone Full CAN (SFCAN) circuit incorporates all the parts for completely autonomous transmission and reception of messages using the CAN protocol. The flexible, programmable interface allows hookup 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.



**Figure 1**  
**Logic Symbol**

**Ordering Information**

Type	Ordering Code	Package	Function
SAE 81C91	Q67121-F0001	P-LCC-28-1	Stand Alone Full CAN Controller Temperature range – 40 to + 110 °C
SAE 81C90	Q67121-H9038	P-LCC-44-1	Stand Alone Full CAN Controller Temperature range – 40 to + 110 °C

Pin Configurations  
(top view)

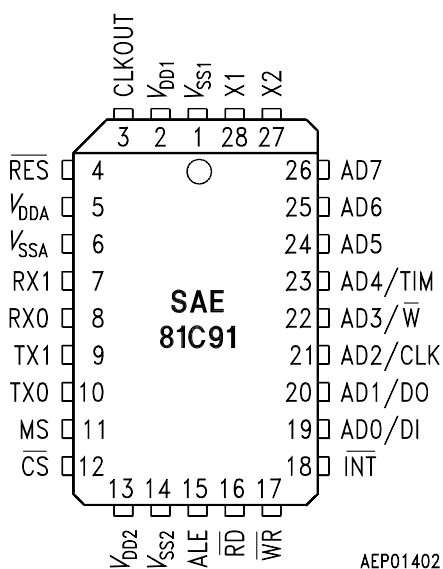
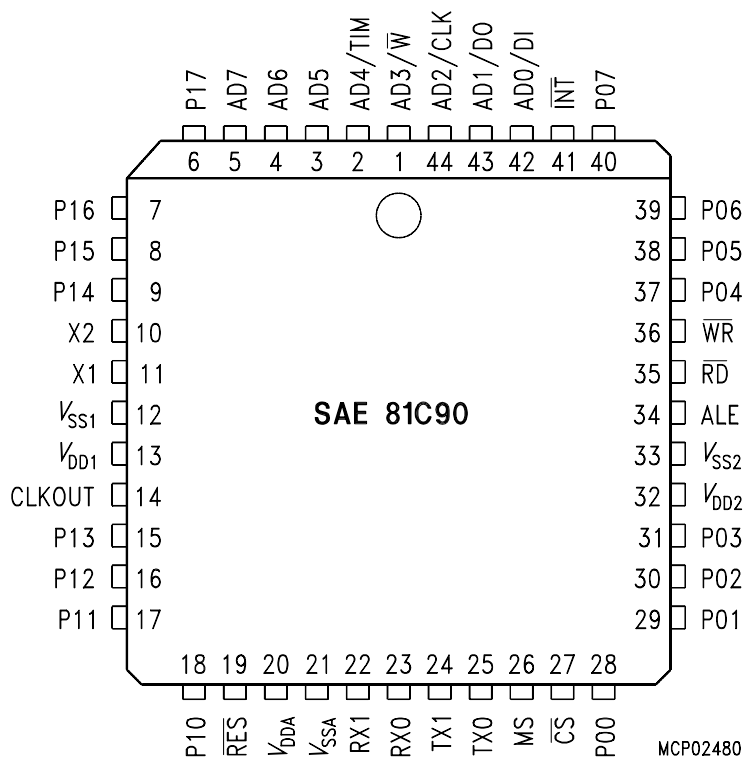


Figure 2

## Pin Definitions and Functions

Symbol	Pin Number		Input (I) Output (O)	Function
	PLCC-48	PLCC-28		
X1 <sup>1)</sup>	11	28	O	Crystal oscillator output. Must be unconnected for external clock input.
X2 <sup>1)</sup>	10	27	I	Crystal oscillator input. Used for external clock input.
CLKOUT <sup>1)</sup>	14	3	O	Clock output
$\overline{\text{RES}}$	19	4	I	Reset. (Schmitt trigger characteristic)
AD0/DI	42	19	I/O	PI: Address / Data bus / SI: Data input
AD1/DO	43	20	I/O	PI: Address / Data bus / SI: Data output
AD2/CLK	44	21	I/O	PI: Address / Data bus / SI: Clock input
AD3/ $\overline{\text{W}}$	1	22	I/O	PI: Address / Data bus / SI: Write select
AD4/TIM	2	23	I/O	PI: Address / Data bus / SI: TIM = 0: Timing A; TIM = 1: Timing B
AD5	3	24	I/O	PI: Address/Data bus
AD6	4	25	I/O	PI: Address/Data bus
AD7	5	26	I/O	PI: Address/Data bus
$\overline{\text{RD}}$	35	16	I	PI: Read / SI: no Function
$\overline{\text{WR}}$	36	17	I	PI: Write / SI: no Function
ALE	34	15	I	PI: Address Latch Enable / SI: no Function
$\overline{\text{CS}}$	27	12	I	Chip Select
$\overline{\text{INT}}$	41	18	O	Interrupt
MS	26	11	I	Mode Select (PI ↔ SI)
P00 ... P03, P04 ... P07	28, 29, 30, 31, 37, 38, 39, 40	–	I/O	Port 0 These pins provide internal pullup resistors of about 10...200 kΩ.
P10 ... P13, P14 ... P17	18, 17, 16, 15, 9, 8, 7, 6	–	I/O	Port 1 These pins provide internal pullup resistors of about 10...200 kΩ.
TX0	25	10	O	Transmitter output 0
TX1	24	9	O	Transmitter output 1
RX0	23	8	I	Comparator input 0 / Digital input <sup>2)</sup>
RX1	22	7	I	Comparator input 1 <sup>2)</sup>
V <sub>DDA</sub>	20	5	I	Analog power supply for comparator (may be unconnected using the digital mode)

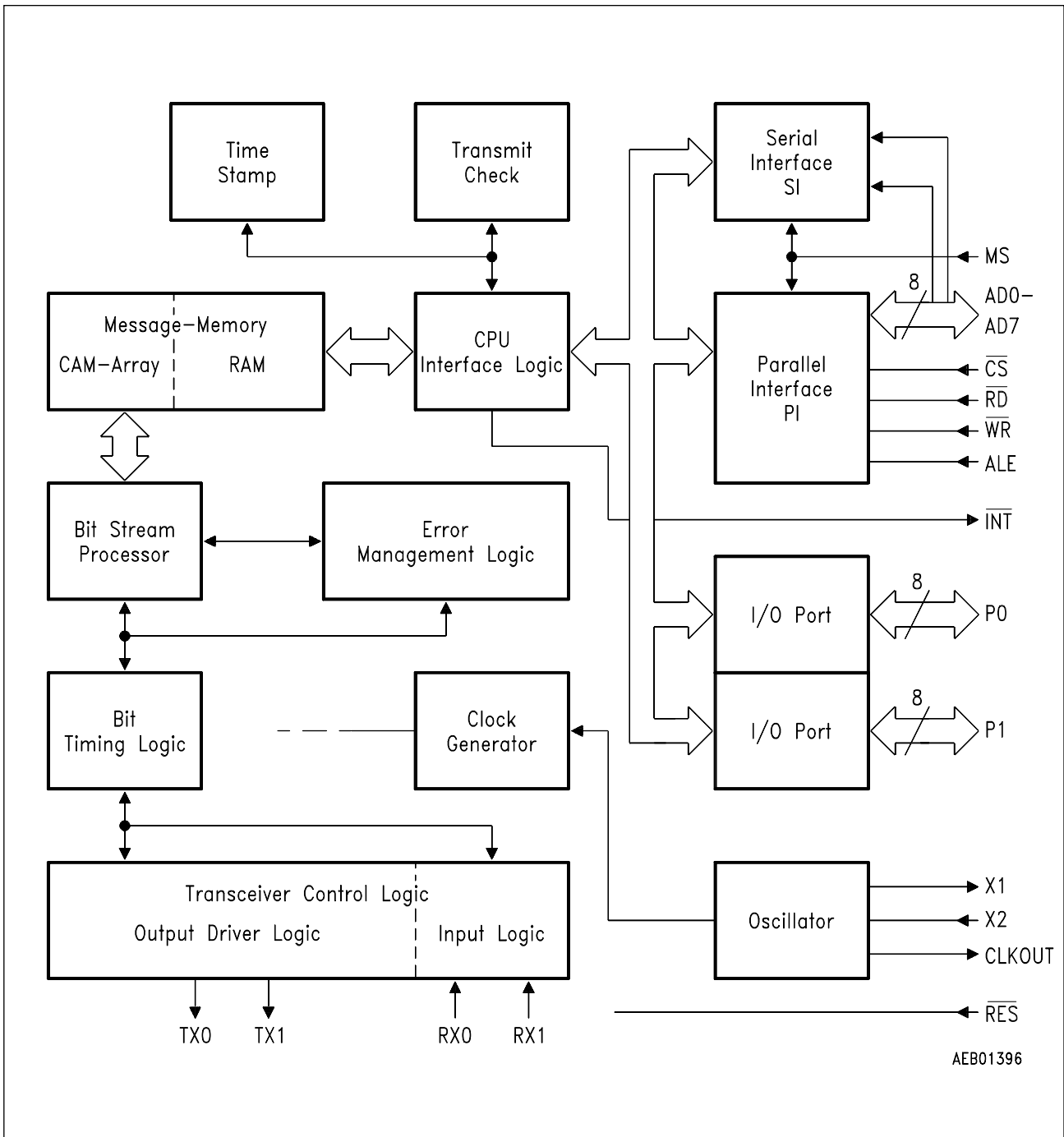
## Pin Definitions and Functions (cont'd)

Symbol	Pin Number		Input (I) Output (O)	Function
	PLCC-48	PLCC-28		
$V_{SSA}$	21	6	I	Analog power ground for comparator (must always be connected)
$V_{DD1}$	13	2	I	Digital power supply <sup>3)</sup>
$V_{DD2}$	32	13	I	Digital power supply
$V_{SS1}$	12	1	I	Digital power ground <sup>3)</sup>
$V_{SS2}$	33	14	I	Digital power ground

- 1) For best results keep the crystal circuitry connections as short as possible and keep the CLKOUT line away from it.
- 2) If the bus lines work according to the ISO specification, additional circuitry is necessary for interconnection of the input comparator to the bus lines.  
The digital mode is enabled by setting bit DI in register BL2. When using the digital mode pin RX1 should be on  $V_{SS}$ .
- 3) It is recommended to decouple these supply pins close to the device using a 10 pF capacitor in addition to the standard 100 nF capacitor.

**Functional 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.



**Figure 3**  
**Block Diagram**

Bus communication is based on the controller-area-network (CAN) protocol. With features like short message length, guaranteed reaction time for messages of appropriate priority, which is defined by the message identifiers. Also included are powerful error detection and treatment capabilities plus ease of operation. The CAN protocol is especially 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.

### 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 with the identifiers stored in the CAM. Upon a match the received data bytes are written into the RAM buffer of the matching message. 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 of a specific message the message objects are not accessed directly but via a 64-bit shadow register (see figure below).

This shadow register stores the complete data field of a certain message object for both reading and writing.

**For read accesses** the message's data field is copied to the shadow register...

...with the 1st read access to the respective data field (e.g. 80<sub>H</sub> ... 87<sub>H</sub> for message 0), or

...with any read access to byte 7 of the respective data field (e.g. 87<sub>H</sub> for message 0).

This ensures that all bytes read via the shadow register belong to the same message, even though a new one might have been received in the meantime.

**For write accesses** the shadow register is copied to the respective message data field...

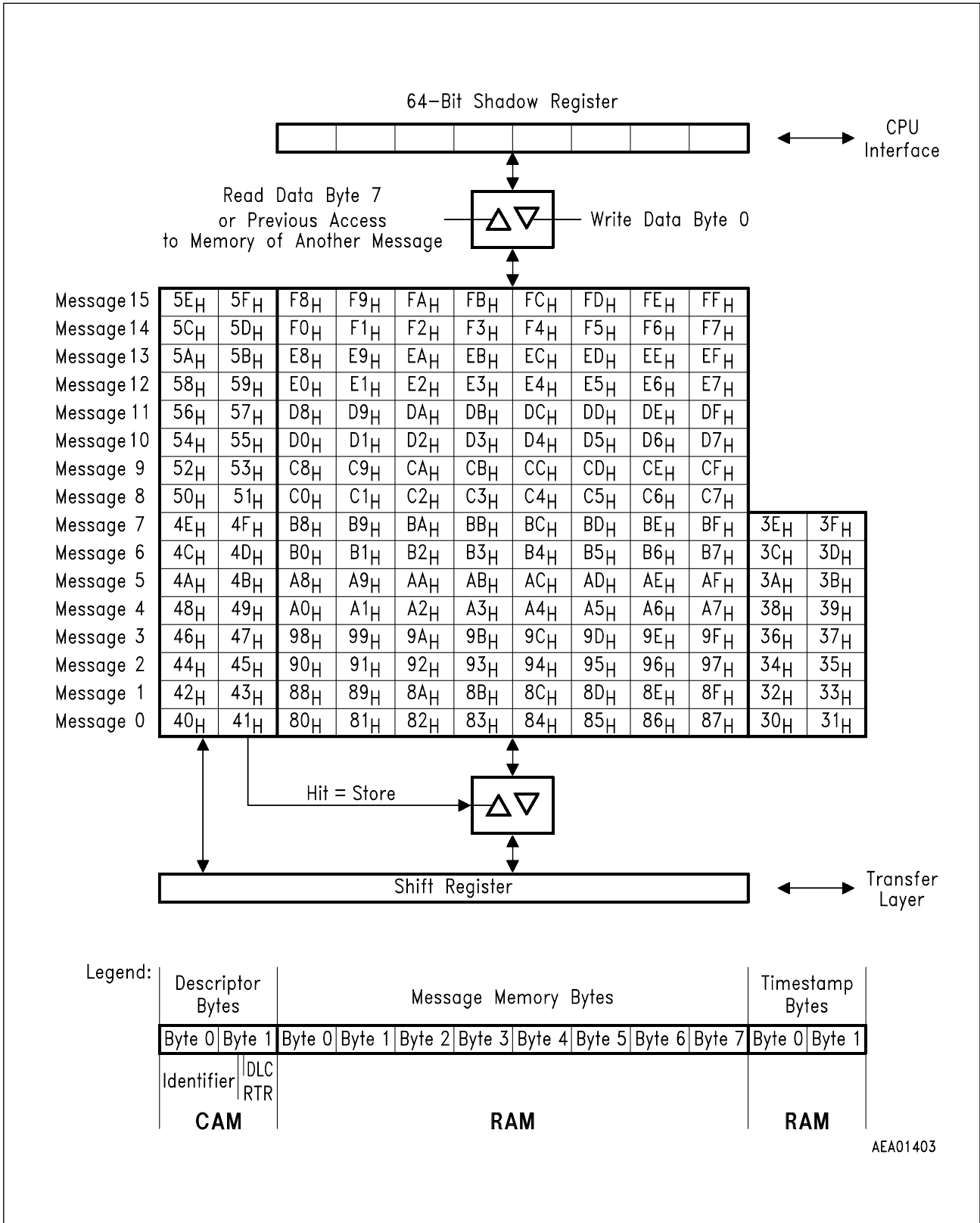
...with any write access to byte 0 of the respective data field (e.g. 80<sub>H</sub> for message 0).

This ensures that only completely updated message are transmitted.

It is therefore recommended to begin all read and write accesses with the most-significant data byte of a message and end with data byte 0. This ensures operations on consistent data and correct transfers between the shadow register and the message RAM.

**Note:** For these reasons it is absolutely essential to ensure that 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 4**  
**CAM, Message Memory and Time-Stamp Registers**

**Bit Stream Processor (BSP)**

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

**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.

**Bit Timing Logic (BTL)**

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

**Transceiver Control Logic (TCL)**

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

**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.

**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.

**Transmit Check**

The CAN protocol ensures a very high 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. To eliminate any possible uncertainties at this point too, the SFCAN circuit incorporates a transmit-check unit. This unit reads back a transmitted message via the normal receive path from the bus interface and compares the data with 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 ETCl 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 the CAN protocol.

### Time Stamp

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, a 16-bit timer is implemented on the SAE 81C90/91. The content of this gets written into the time-stamp registers of the particular message when it is received (for the 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 bitfield TSP in register CTRL, 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.

### 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.

**Note:** Registers PxPDR and PxPL may be used for general purpose storage if the ports are not used.

## Device Control and Registers

The operation of the SAE 81C90/91 is controlled via a number of registers. These registers allow initialization and function control, provide status information and configure the message objects. The upper part of the address space provides access to the data buffers of the message objects. The data buffers are ordered sequentially as shown in the table below.

The register map on the next page summarizes the other registers (i.e. except the data registers) ordered by their address, while the following pages describe these registers in more detail from a functional point of view.

**Note:** Locations marked “*Reserved*” in the register map must not be written in initialization mode. This also applies to locations 60<sub>H</sub> through 7F<sub>H</sub>.

## Data Registers

Address	Function	
80 <sub>H</sub>	Byte 0	Message 0
81 <sub>H</sub>	Byte 1	
82 <sub>H</sub>	Byte 2	
83 <sub>H</sub>	Byte 3	
84 <sub>H</sub>	Byte 4	
85 <sub>H</sub>	Byte 5	
86 <sub>H</sub>	Byte 6	
87 <sub>H</sub>	Byte 7	
88 <sub>H</sub>	Byte 0	Message 1
89 <sub>H</sub>	Byte 1	
:	:	:
F6 <sub>H</sub>	Byte 6	Message 14
F7 <sub>H</sub>	Byte 7	
F8 <sub>H</sub>	Byte 0	Message 15
F9 <sub>H</sub>	Byte 1	
FA <sub>H</sub>	Byte 2	
FB <sub>H</sub>	Byte 3	
FC <sub>H</sub>	Byte 4	
FD <sub>H</sub>	Byte 5	
FE <sub>H</sub>	Byte 6	
FF <sub>H</sub>	Byte 7	

## Register Map (ordered by address)

Addr.	Reg. Name	Reset	Addr.	Reg. Name	Reset	Addr.	Reg. Name	Reset
00 <sub>H</sub>	BL1	00 <sub>H</sub>	20 <sub>H</sub>	<i>Reserved</i>	---	40 <sub>H</sub>	DR0H	UU <sub>H</sub>
01 <sub>H</sub>	BL2	00 <sub>H</sub>	21 <sub>H</sub>	<i>Reserved</i>	---	41 <sub>H</sub>	DR0L	UU <sub>H</sub>
02 <sub>H</sub>	OC	00 <sub>H</sub>	22 <sub>H</sub>	<i>Reserved</i>	---	42 <sub>H</sub>	DR1H	UU <sub>H</sub>
03 <sub>H</sub>	BRP	00 <sub>H</sub>	23 <sub>H</sub>	<i>Reserved</i>	---	43 <sub>H</sub>	DR1L	UU <sub>H</sub>
04 <sub>H</sub>	RRR1	00 <sub>H</sub>	24 <sub>H</sub>	<i>Reserved</i>	---	44 <sub>H</sub>	DR2H	UU <sub>H</sub>
05 <sub>H</sub>	RRR2	00 <sub>H</sub>	25 <sub>H</sub>	<i>Reserved</i>	---	45 <sub>H</sub>	DR2L	UU <sub>H</sub>
06 <sub>H</sub>	RIMR1	00 <sub>H</sub>	26 <sub>H</sub>	<i>Reserved</i>	---	46 <sub>H</sub>	DR3H	UU <sub>H</sub>
07 <sub>H</sub>	RIMR2	00 <sub>H</sub>	27 <sub>H</sub>	<i>Reserved</i>	---	47 <sub>H</sub>	DR3L	UU <sub>H</sub>
08 <sub>H</sub>	TRS1	00 <sub>H</sub>	28 <sub>H</sub>	P0PDR	00 <sub>H</sub>	48 <sub>H</sub>	DR4H	UU <sub>H</sub>
09 <sub>H</sub>	TRS2	00 <sub>H</sub>	29 <sub>H</sub>	P0PR	XX <sub>H</sub>	49 <sub>H</sub>	DR4L	UU <sub>H</sub>
0A <sub>H</sub>	IMSK	00 <sub>H</sub>	2A <sub>H</sub>	P0LR	00 <sub>H</sub>	4A <sub>H</sub>	DR5H	UU <sub>H</sub>
0B <sub>H</sub>	<i>Reserved</i>	---	2B <sub>H</sub>	<i>Reserved</i>	---	4B <sub>H</sub>	DR5L	UU <sub>H</sub>
0C <sub>H</sub>	<i>Reserved</i>	---	2C <sub>H</sub>	P1PDR	00 <sub>H</sub>	4C <sub>H</sub>	DR6H	UU <sub>H</sub>
0D <sub>H</sub>	<i>Reserved</i>	---	2D <sub>H</sub>	P1PR	XX <sub>H</sub>	4D <sub>H</sub>	DR6L	UU <sub>H</sub>
0E <sub>H</sub>	<i>Reserved</i>	---	2E <sub>H</sub>	P1LR	00 <sub>H</sub>	4E <sub>H</sub>	DR7H	UU <sub>H</sub>
0F <sub>H</sub>	<i>Reserved</i>	---	2F <sub>H</sub>	<i>Reserved</i>	---	4F <sub>H</sub>	DR7L	UU <sub>H</sub>
10 <sub>H</sub>	MOD	00 <sub>H</sub>	30 <sub>H</sub>	TSR0H	UU <sub>H</sub>	50 <sub>H</sub>	DR8H	UU <sub>H</sub>
11 <sub>H</sub>	INT	00 <sub>H</sub>	31 <sub>H</sub>	TSR0L	UU <sub>H</sub>	51 <sub>H</sub>	DR8L	UU <sub>H</sub>
12 <sub>H</sub>	CTRL	00 <sub>H</sub>	32 <sub>H</sub>	TSR1H	UU <sub>H</sub>	52 <sub>H</sub>	DR9H	UU <sub>H</sub>
13 <sub>H</sub>	<i>Reserved</i>	---	33 <sub>H</sub>	TSR1L	UU <sub>H</sub>	53 <sub>H</sub>	DR9L	UU <sub>H</sub>
14 <sub>H</sub>	CC	01 <sub>H</sub>	34 <sub>H</sub>	TSR2H	UU <sub>H</sub>	54 <sub>H</sub>	DR10H	UU <sub>H</sub>
15 <sub>H</sub>	TCEC	00 <sub>H</sub>	35 <sub>H</sub>	TSR2L	UU <sub>H</sub>	55 <sub>H</sub>	DR10L	UU <sub>H</sub>
16 <sub>H</sub>	TCD	XX <sub>H</sub>	36 <sub>H</sub>	TSR3H	UU <sub>H</sub>	56 <sub>H</sub>	DR11H	UU <sub>H</sub>
17 <sub>H</sub>	<i>Reserved</i>	---	37 <sub>H</sub>	TSR3L	UU <sub>H</sub>	57 <sub>H</sub>	DR11L	UU <sub>H</sub>
18 <sub>H</sub>	TRR1	00 <sub>H</sub>	38 <sub>H</sub>	TSR4H	UU <sub>H</sub>	58 <sub>H</sub>	DR12H	UU <sub>H</sub>
19 <sub>H</sub>	TRR2	00 <sub>H</sub>	39 <sub>H</sub>	TSR4L	UU <sub>H</sub>	59 <sub>H</sub>	DR12L	UU <sub>H</sub>
1A <sub>H</sub>	RRP1	00 <sub>H</sub>	3A <sub>H</sub>	TSR5H	UU <sub>H</sub>	5A <sub>H</sub>	DR13H	UU <sub>H</sub>
1B <sub>H</sub>	RRP2	00 <sub>H</sub>	3B <sub>H</sub>	TSR5L	UU <sub>H</sub>	5B <sub>H</sub>	DR13L	UU <sub>H</sub>
1C <sub>H</sub>	TSCH	00 <sub>H</sub>	3C <sub>H</sub>	TSR6H	UU <sub>H</sub>	5C <sub>H</sub>	DR14H	UU <sub>H</sub>
1D <sub>H</sub>	TSCL	00 <sub>H</sub>	3D <sub>H</sub>	TSR6L	UU <sub>H</sub>	5D <sub>H</sub>	DR14L	UU <sub>H</sub>
1E <sub>H</sub>	<i>Reserved</i>	---	3E <sub>H</sub>	TSR7H	UU <sub>H</sub>	5E <sub>H</sub>	DR15H	UU <sub>H</sub>
1F <sub>H</sub>	<i>Reserved</i>	---	3F <sub>H</sub>	TSR7L	UU <sub>H</sub>	5F <sub>H</sub>	DR15L	UU <sub>H</sub>

**Note:** The locations marked “UU<sub>H</sub>” are not changed upon a reset. After a power on reset they are undefined (XX<sub>H</sub>), of course.

**Descriptor Registers**

A descriptor register is available for each message object and contains the eleven bits of the message identifier (ID.0 through ID.10), the remote-transmission-request bit (RTR) and the data length code (DLC) of a message.

<b>DRnH</b>	7	6	5	4	3	2	1	0
Address: XX <sub>H</sub>	<b>ID.10</b>	<b>ID.9</b>	<b>ID.8</b>	<b>ID.7</b>	<b>ID.6</b>	<b>ID.5</b>	<b>ID.4</b>	<b>ID.3</b>
Reset Value: UU <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>DRnL</b>	7	6	5	4	3	2	1	0
Address: XX <sub>H</sub>	<b>ID.2</b>	<b>ID.1</b>	<b>ID.0</b>	<b>RTR</b>	<b>DLC</b>			
Reset Value: UU <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>DLC</b>	<b>Data Length Code</b> Defines the number of data bytes of message n. Defined values are 0000...1000, i.e. 0...8 bytes. Other values are not permitted.
<b>RTR</b>	<b>Remote Transmission Request Bit</b> '0': This message operates as a data frame. '1': This message operates as a remote frame. <b>Note:</b> See description and table below.
<b>ID.10-0</b>	<b>Identifier</b> Identifier associated with message n, controls the acceptance of received frames and is inserted into transmitted frames.

n = 0...15

Bit RTR determines the function of the corresponding message object when it is transmitted, and its reaction on a received data frame or remote frame.

The table below summarizes the message object's behaviour in the different cases.

RTR bit	Object is transmitted	Matching Data Frame received	Matching Remote Frame received
'0' (data frame)	The message object is transmitted as a standard data frame.	The data frame is stored in the message object.	The remote frame is ignored.
'1' (remote frame)	The message object is transmitted as a remote frame (i.e. a request).	The data frame is ignored and not stored.	The message object is sent as a data frame.

**Note:** For the transmission of remote frames (RTR = 1) the data-length-code should be set to '0'.

## Descriptor Register Arrangement

Address	Function	
40 <sub>H</sub>	High Byte	Descriptor Register for Message Object 0
41 <sub>H</sub>	Low Byte	
42 <sub>H</sub>	High Byte	Descriptor Register for Message Object 1
43 <sub>H</sub>	Low Byte	
:	:	:
5C <sub>H</sub>	High Byte	Descriptor Register for Message Object 14
5D <sub>H</sub>	Low Byte	
5E <sub>H</sub>	High Byte	Descriptor Register for Message Object 15
5F <sub>H</sub>	Low Byte	

## Control Register Summary

Register Name	Address	Function	Reset Value	Read Write <sup>1)</sup>
OC	02 <sub>H</sub>	Output-control register	00 <sub>H</sub>	r/w, I
CC	14 <sub>H</sub>	Clock-control register	01 <sub>H</sub>	wo
CTRL	12 <sub>H</sub>	Control register	00 <sub>H</sub>	r/w
MOD	10 <sub>H</sub>	Mode/status register	00 <sub>H</sub>	r/w
INT	11 <sub>H</sub>	Interrupt register	00 <sub>H</sub>	r/w
IMSK	0A <sub>H</sub>	Interrupt-mask register	00 <sub>H</sub>	r/w
BL1	00 <sub>H</sub>	Bit-length register 1	00 <sub>H</sub>	r/w, I
BL2	01 <sub>H</sub>	Bit-length register 2	00 <sub>H</sub>	r/w, I
BRP	03 <sub>H</sub>	Baud-rate prescaler	00 <sub>H</sub>	wo, I
RRR1	04 <sub>H</sub>	Receive-ready register 1	00 <sub>H</sub>	r/w
RRR2	05 <sub>H</sub>	Receive-ready register 2	00 <sub>H</sub>	r/w
RIMR1	06 <sub>H</sub>	Receive-interrupt-mask register 1	00 <sub>H</sub>	r/w
RIMR2	07 <sub>H</sub>	Receive-interrupt-mask register 2	00 <sub>H</sub>	r/w
TRSR1	08 <sub>H</sub>	Transmit-request-set register 1	00 <sub>H</sub>	r/w
TRSR2	09 <sub>H</sub>	Transmit-request-set register 2	00 <sub>H</sub>	r/w
TRRR1	18 <sub>H</sub>	Transmit-request-reset register 1	00 <sub>H</sub>	wo
TRRR2	19 <sub>H</sub>	Transmit-request-reset register 2	00 <sub>H</sub>	wo
RRPR1	1A <sub>H</sub>	Remote-request-pending register 1	00 <sub>H</sub>	ro
RRPR2	1B <sub>H</sub>	Remote-request-pending register 2	00 <sub>H</sub>	ro
TSCH	1C <sub>H</sub>	Time-Stamp counter high byte	00 <sub>H</sub>	r/w
TSCL	1D <sub>H</sub>	Time-Stamp counter low byte	00 <sub>H</sub>	r/w
TCEC	15 <sub>H</sub>	Transmit-check error counter	00 <sub>H</sub>	r/w
TCD	16 <sub>H</sub>	Transmit-check data register	XX	ro
P0PDR	28 <sub>H</sub>	Port 0 port-direction register	00 <sub>H</sub>	r/w
P1PDR	2C <sub>H</sub>	Port 1 port-direction register	00 <sub>H</sub>	r/w
P0LR	2A <sub>H</sub>	Port 0 latch register	00 <sub>H</sub>	r/w
P1LR	2E <sub>H</sub>	Port 1 latch register	00 <sub>H</sub>	r/w
P0PR	29 <sub>H</sub>	Port 0 pin register	XX <sub>H</sub>	ro
P1PR	2D <sub>H</sub>	Port 1 pin register	XX <sub>H</sub>	ro

**Note:** <sup>1)</sup> **ro:** read only, **r/w:** read and write access, **wo:** write only, **I:** access only with bit IM set.



**Output-Control Register**

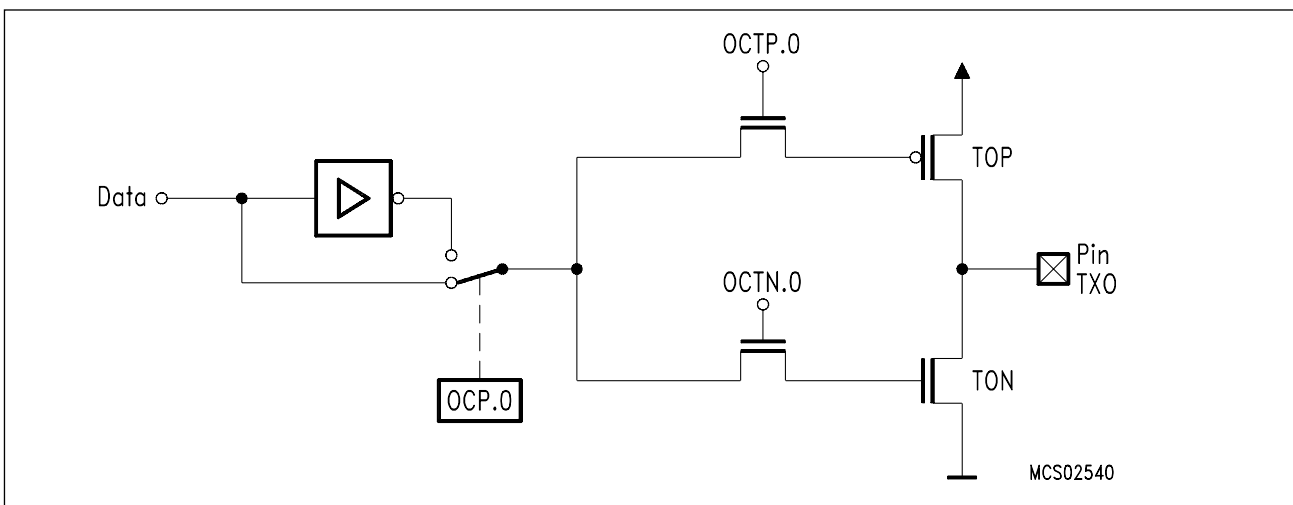
The output drivers of the SAE 81C90/91's transmit pins (TXn) can be individually configured. Thus they can be adapted to the requirements of the external bs system.

<b>OC</b>	7	6	5	4	3	2	1	0
Address: 02 <sub>H</sub>	<b>OCTP1</b>	<b>OCTN1</b>	<b>OCP1</b>	<b>OCTP0</b>	<b>OCTN0</b>	<b>OCP0</b>	<b>OCM</b>	
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>OCM</b>	<p><b>Output Mode</b></p> <p>'0X': <b>Normal Mode:</b> TX0 = Bit Sequence, TX1 = Bit Sequence.</p> <p>'10': <b>Test Mode:</b> TX0 = Bit Sequence, TX1 = RX0.</p> <p>'11': <b>Clock Mode:</b> TX0 = Bit Sequence, TX1 = Bit Clock.</p>
<b>OCPn</b>	<p><b>Output Polarity</b></p> <p>'0': Output is driven directly with CAN data.</p> <p>'1': Output is driven with inverted CAN data.</p>
<b>OCTNn</b>	<p><b>Negative Output Transistor Control</b></p> <p>'0': The low side output transistor TnN is disabled.</p> <p>'1': The low side output transistor TnN drives the pin according to data.</p>
<b>OCTPn</b>	<p><b>Positive Output Transistor Control</b></p> <p>'0': The high side output transistor TnP is disabled.</p> <p>'1': The high side output transistor TnP drives the pin according to data.</p>

n = 0, 1

**Note:** This register can only be written when bit IM (MOD.0) is set.



**Figure 5**  
**Output Control Circuitry**

**Output Programming**

OCTP.n	OCTN.n	OCP.n	Data	TnP	TnN	TXn-Level
0	0	0	0 = dominant	OFF	OFF	float
0	0	0	1 = recessive	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

TnP is the output transistor switching to  $V_{DD}$ , TnN switches to  $V_{SS}$ .  
 TXn is the output level at the transmit pin.

**Clock Control Register**

The Clock Control Register determines the output frequency at pin CLKOUT which is derived from the oscillator frequency.

<b>CCR</b>	7	6	5	4	3	2	1	0
Address: 14 <sub>H</sub>	-	-	-	-	<b>CC</b>			
Reset Value: 01 <sub>H</sub>	-	-	-	-	W	W	W	W

Bit(field)	Function
<b>CC</b>	<p><b>Clock Output Control</b></p> <p>'0000': <math>f_{CLKOUT} = f_{OSC}</math></p> <p>'0001': <math>f_{CLKOUT} = f_{OSC} / 2</math></p> <p>'0010': <math>f_{CLKOUT} = f_{OSC} / 4</math></p> <p>'0011': <math>f_{CLKOUT} = f_{OSC} / 6</math></p> <p>'0100': <math>f_{CLKOUT} = f_{OSC} / 8</math></p> <p>'0101': <math>f_{CLKOUT} = f_{OSC} / 10</math></p> <p>'0110': <math>f_{CLKOUT} = f_{OSC} / 12</math></p> <p>'0111': <math>f_{CLKOUT} = f_{OSC} / 14</math></p> <p>'1XXX': <math>f_{CLKOUT} = \text{LOW}</math> (clock output is switched off)</p>

The Clock Control Register requires a special protocol for writing in order to prevent the clock output from being changed inadvertently:

- Step 1: Write 80<sub>H</sub> to CC
- Step 2: Write desired value to CC (bits 7...4 must be '0000')

**Note:** Not defined bit positions must be '0' for write accesses.

## Control Register

<b>CTRL</b>	7	6	5	4	3	2	1	0
Address: 12 <sub>H</sub>	<b>RX</b>	<b>TST</b>	<b>TSP</b>		<b>TSOV</b>	<b>SME</b>	<b>TCE</b>	<b>MM</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>MM</b>	<p><b>Monitor Mode</b></p> <p>'0': Message object 0 operates like all other objects.</p> <p>'1': Message object 0 receives all identifiers that are not accepted by other objects (corresponds to a Basic CAN receive register).</p>
<b>TCE</b>	<p><b>Transmit Check Enable</b></p> <p>'0': If the transmit check detects an error, there is no intervention.</p> <p>'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.</p>
<b>SME</b>	<p><b>Sleep Mode Enable</b></p> <p>'0': Normal operation.</p> <p>'1': The sleep mode is enabled: the crystal oscillator is deactivated, all other activities are inhibited.</p> <p>The wake up is done by a reset signal or by an active signal at the <math>\overline{CS}</math> pin or by an input edge going from recessive to dominant at pin Rx0 or Rx1.</p>
<b>TSOV</b>	<p><b>Time Stamp Overflow</b></p> <p>'0': There has been no overflow</p> <p>'1': There was at least one overflow of the time-stamp timer.</p>
<b>TSP</b>	<p><b>Time Stamp Prescaler</b> (Defines the input clock of the time-stamp timer)</p> <p>'00': <math>f_{BL} / 32</math></p> <p>'01': <math>f_{BL} / 64</math></p> <p>'10': <math>f_{BL} / 128</math></p> <p>'11': <math>f_{BL} / 256</math> (For <math>f_{BL}</math> see baud-rate prescaler BRP).</p>
<b>TST</b>	<p><b>Time Stamp Test</b></p> <p>'0': The prescaler is activated.</p> <p>'1': The time-stamp prescaler is deactivated.</p> <p>(Only for testing purposes, bit IM = MOD.0 must be set to '1').</p>
<b>RX</b>	<p><b>Input Monitor RX</b></p> <p>This bit monitors the actual state of the digital input pin RX0.</p>

**Mode/Status-Register**

<b>MOD</b>	7	6	5	4	3	2	1	0
Address: 10 <sub>H</sub>	<b>ADE</b>	<b>RS</b>	<b>TC</b>	<b>TWL</b>	<b>RWL</b>	<b>BS</b>	<b>RES</b>	<b>IM</b>
Reset Value: 00 <sub>H</sub>	rw	r	r	r	r	r	rw	rw

Bit(field)	Function
<b>IM</b>	<p><b>Init Mode</b></p> <p>'0': Normal mode.</p> <p>'1': Initialization mode: write access to the configuration registers BL1, BL2, OC, BRP is enabled. If the bit stays set, the chip enters the normal mode, with enabled access to the configuration registers. If this bit is set in conjunction with bit RES a hard software reset is activated.</p>
<b>RES</b>	<p><b>Reset Request</b></p> <p>'0': Normal mode.</p> <p>'1': The chip enters the reset state: – if bit IM = '0' a soft software reset takes place. – if bit IM = '1' a hard software reset takes place. Further details see below.</p>
<b>BS</b>	<p><b>Bus State (read only)</b></p> <p>'0': Normal mode.</p> <p>'1': Bus Off state, the IC does not participate in bus activities.</p>
<b>RWL</b>	<p><b>Receiver Warning Level (read only)</b></p> <p>'0': Receive-error counter below 96.</p> <p>'1': Receive-error counter equal or above 96.</p>
<b>TWL</b>	<p><b>Transmit Warning Level (read only)</b></p> <p>'0': Transmit-error counter below 96.</p> <p>'1': Transmit-error counter equal or above 96.</p>
<b>TC</b>	<p><b>Transmission Complete (read only)</b></p> <p>'0': The last transmission request is not yet executed successfully.</p> <p>'1': The last transmission request was executed successfully.</p>
<b>RS</b>	<p><b>Receive State (read only)</b></p> <p>'0': No reception active.</p> <p>'1': Currently the SAE 81C90/91 is in receive mode.</p>
<b>ADE</b>	<p><b>Auto Decrement Enable</b></p> <p>'0': No automatic address decrement.</p> <p>'1': With every read or write access using the serial synchronous interface SI the address is automatically decremented by one. So data can be accessed sequentially without the need of writing a new address.</p>

**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 acknowledgment of the previous message and the start of the next one.

**Notes on Bit RES and IM and reset modes**

There are three different reset modes implemented in the SAE 81C90/91:

hardware reset (activated by low level on pin  $\overline{\text{RES}}$ )

hard software reset (activated by setting both bits RES and IM to 1)

soft software reset (activated by setting bit RES to 1 and bit IM to 0)

The only difference between hardware and hard software reset affect bits RES and IM, that are not changed by software reset.

With soft software reset the registers RRR1, RRR2, TRSR1, TRSR2, RRPR1 and RRPR2 are cleared, all bus activities are stopped, the error counters are not cleared, the Bus Off state is cancelled only after 128 idle phases (according to the CAN protocol 1 idle phase = 11 recessive bits in sequence). Simply spoken a soft software reset interrupts and cancels all bus activities and - if necessary - recovers from Bus Off state.

**Notes on Bit RS**

Bit RS directly reflects the internal status.

RS is '0' during transmission or when the SAE 81C90/91 is idle.

RS is '1' during reception or during the synchronization after a reset.

**Interrupt Register**

<b>INT</b>	7	6	5	4	3	2	1	0
Address: 11 <sub>H</sub>	<b>TCI</b>	<b>EPI</b>	<b>BOI</b>	<b>WUPI</b>	<b>RFI</b>	<b>WLI</b>	<b>TI</b>	<b>RI</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>RI</b>	<b>Receive Interrupt</b> After a valid message has been received and filed, this bit is set and an interrupt generated. This bit will remain set until all bits of the registers RRR1 and RRR2 are reset.
<b>TI</b>	<b>Transmit Interrupt</b> This bit is set and an interrupt generated as soon as a transmit request has been processed.
<b>WLI</b>	<b>Warning Level Interrupt</b> If at least one of the two error counters is greater than or equals 96, this bit is set and an interrupt generated.
<b>RFI</b>	<b>Remote Frame Interrupt</b> This interrupt is generated after reception of a remote frame.
<b>WUPI</b>	<b>Wake Up Interrupt</b> After a wake-up this bit is set and an interrupt generated.
<b>BOI</b>	<b>Bus Off Interrupt</b> This bit is set and an interrupt generated when the Bus Off status is entered.
<b>EPI</b>	<b>Error Passive Interrupt</b> If at least one of the two error counters is greater than or equals 128, this bit is set and an interrupt generated.
<b>TCI</b>	<b>Transmit Check Interrupt</b> If the transmit-check error counter reaches 4, this bit is set and an interrupt generated.

**Note:** All bits of this register must be reset by software. This is done by writing '0' to the respective bit location, writing '1' has no effect.  
An interrupt is only generated if the respective IMSK bit is set. The bits in this register are set independent of register IMSK (see below).  
The interrupt output is active for at least one bit time. The interrupt output is deactivated when all enabled request bits are cleared. A request bit is enabled by setting its corresponding mask bit. Masked request bits do not activate the interrupt output.

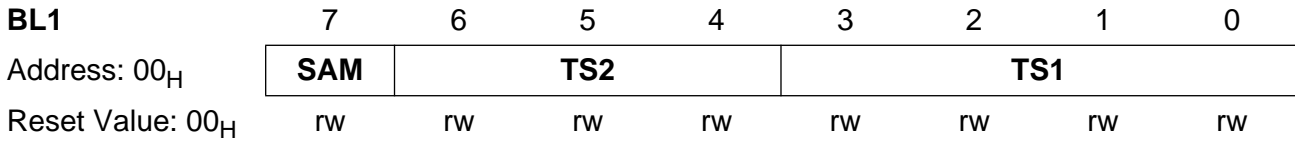
**Interrupt-Mask Register**

These mask bits determine if an event activates the INT pin. They do not influence the INT register.

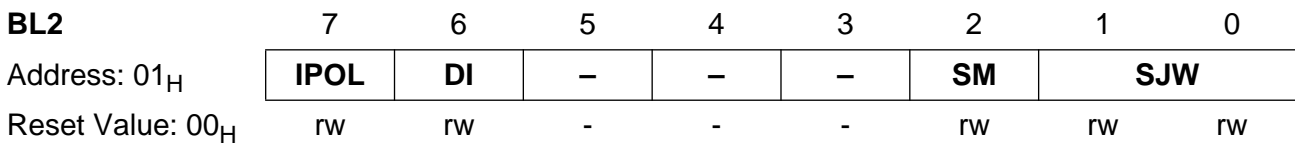
<b>IMSK</b>	7	6	5	4	3	2	1	0
Address: 0A <sub>H</sub>	<b>ETCI</b>	<b>EEPI</b>	<b>EBOI</b>	<b>EWUPI</b>	<b>ERFI</b>	<b>EWLI</b>	<b>ETI</b>	<b>ERI</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>ERI</b>	<b>Enable Receive Interrupt</b> '0': No receive interrupt enabled. '1': Receive interrupts are enabled.
<b>ETI</b>	<b>Enable Transmit Interrupt</b> '0': No transmit interrupt enabled. '1': Completed transmit jobs generate interrupts.
<b>EWLI</b>	<b>Enable Warning Level Interrupt</b> '0': No warning level interrupt enabled. '1': There is an interrupt when the warning level is reached.
<b>ERFI</b>	<b>Enable Remote Frame Interrupt</b> '0': No remote frame interrupt enabled. '1': A receive interrupt is generated after receiving a remote frame
<b>EWUPI</b>	<b>Enable Wake Up Interrupt</b> '0': No wake up interrupt enabled. '1': Wake-up interrupt is enabled.
<b>EBOI</b>	<b>Enable Bus Off Interrupt</b> '0': No bus off interrupt enabled. '1': Bus off interrupt is enabled.
<b>EEPI</b>	<b>Enable Error Passive Interrupt</b> '0': No error passive interrupt enabled. '1': Error passive interrupt is enabled.
<b>ETCI</b>	<b>Enable Transmit Check Error Interrupt</b> '0': No transmit check interrupt enabled. '1': Transmit-check error interrupt is enabled.

**Bit-Length Registers**



Bit(field)	Function
<b>TS1</b>	<b>Length of Timing Segment 1 (TSeg1).</b> $t_{\text{TSeg1}} = (\text{TS1} + 1) \times t_{\text{SCL}}$ . For $t_{\text{SCL}}$ see baud-rate prescaler BRP.
<b>TS2</b>	<b>Length of Timing Segment 2 (TSeg2).</b> $t_{\text{TSeg2}} = (\text{TS2} + 1) \times t_{\text{SCL}}$ . For $t_{\text{SCL}}$ see baud-rate prescaler BRP.
<b>SAM</b>	<b>Sample Rate</b> '0': Input signal is sampled once per bit. '1': Input signal is sampled three times per bit. <b>Note:</b> Bit SAM should only be set to '1' using very low baud rates.



Bit(field)	Function
<b>SJW</b>	<b>Maximum Synchronization Jump Width.</b> $t_{\text{SJWidth}} = (\text{SJW} + 1) \times t_{\text{SCL}}$ . For $t_{\text{SCL}}$ see baud-rate prescaler BRP.
<b>SM</b>	<b>Speed Mode (Defines edge used for synchronization)</b> '0': Recessive to dominant is used. '1': Both edges are used. <b>Note:</b> According to the CAN specification this bit should not be set to '1'.
<b>DI</b>	<b>Digital Input</b> '0': The input signal is applied to the input comparator. <sup>1)</sup> '1': The input signal on pin RX0 is evaluated digitally. The input comparator is inactive. Pin RX1 should be on $V_{\text{SS}}$ .
<b>IPOL</b>	<b>Input Polarity</b> '0': The input level remains unaltered. '1': The input level is inverted.

**Note:** Not defined bit positions must be '0' for write accesses.  
The Bit Length Registers BL1 and BL2 can only be written while bit IM (MOD.0) is set.  
<sup>1)</sup> If the bus lines work according to the ISO specification, additional circuitry is necessary for interconnection of the input comparator to the bus lines.



**Baud Rate Prescaler Register**

The register is not readable and can only be written when bit IM (MOD.0) is set.

<b>BRPR</b>	7	6	5	4	3	2	1	0
Address: 03 <sub>H</sub>	-	-	<b>BRP</b>					
Reset Value: 00 <sub>H</sub>	-	-	W	W	W	W	W	W

Bit(field)	Function
<b>BRP</b>	<p><b>Baud Rate Prescaler</b>                      This prescaler determines the period of the system clock:  <math>t_{SCL} = (BRP + 1) \times 2 \times t_{OSC}</math>, where <math>t_{OSC} = 1 / f_{crystal}</math>.</p>

**Note:** Not defined bit positions must be '0' for write accesses.

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

$$t_{BL} = t_{TSeg1} + t_{TSeg2} + 1 t_{SCL}$$

The baudrate BR can be computed with the following formula:

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

- Note:** BRP      see Baud Rate Prescaler Register  
 TS1            see Bit Length Register 1  
 TS2            see Bit Length Register 1

**Receive-Ready Registers**

<b>RRR2</b>	7	6	5	4	3	2	1	0
Address: 05 <sub>H</sub>	<b>RR15</b>	<b>RR14</b>	<b>RR13</b>	<b>RR12</b>	<b>RR11</b>	<b>RR10</b>	<b>RR9</b>	<b>RR8</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>RRR1</b>	7	6	5	4	3	2	1	0
Address: 04 <sub>H</sub>	<b>RR7</b>	<b>RR6</b>	<b>RR5</b>	<b>RR4</b>	<b>RR3</b>	<b>RR2</b>	<b>RR1</b>	<b>RR0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>RRn</b>	<b>Receive Ready Bit</b> '0': No new message received in object n. '1': A new message has been received and stored in object n.

These register bits can be reset by writing '0' to the respective bit, writing '1' has no effect. Bit RRn is set when a message has arrived and been written into the memory location of message n. Setting this bit by hardware can generate a receive interrupt, which can be blocked by bit RIMn in the receive-interrupt-mask register. Bits RRn must be reset by software.

**Receive-Interrupt-Mask Registers**

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

<b>RIMR2</b>	7	6	5	4	3	2	1	0
Address: 07 <sub>H</sub>	<b>RIM15</b>	<b>RIM14</b>	<b>RIM13</b>	<b>RIM12</b>	<b>RIM11</b>	<b>RIM10</b>	<b>RIM9</b>	<b>RIM8</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>RIMR1</b>	7	6	5	4	3	2	1	0
Address: 06 <sub>H</sub>	<b>RIM7</b>	<b>RIM6</b>	<b>RIM5</b>	<b>RIM4</b>	<b>RIM3</b>	<b>RIM2</b>	<b>RIM1</b>	<b>RIM0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>RIMn</b>	<b>Receive Interrupt Mask Bit</b> '0': No interrupt upon reception of object n. '1': When a new message is stored in object n an interrupt is generated.

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

**Transmit Request Registers**

The **Transmit Request Set Registers** provide a transmission request bit (TRS<sub>n</sub>) for each message object. Setting a transmission request bit causes the respective message x to be transmitted. The bit is cleared 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.

**Note:** A transmission request bit is set by writing '1' to the respective bit location (TRS<sub>n</sub>). Writing '0' has no effect.

<b>TRSR2</b>	7	6	5	4	3	2	1	0
Address: 09 <sub>H</sub>	<b>TRS15</b>	<b>TRS14</b>	<b>TRS13</b>	<b>TRS12</b>	<b>TRS11</b>	<b>TRS10</b>	<b>TRS9</b>	<b>TRS8</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>TRSR1</b>	7	6	5	4	3	2	1	0
Address: 08 <sub>H</sub>	<b>TRS7</b>	<b>TRS6</b>	<b>TRS5</b>	<b>TRS4</b>	<b>TRS3</b>	<b>TRS2</b>	<b>TRS1</b>	<b>TRS0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
TRS <sub>n</sub>	<p><b>Transmit Request Set Bit</b></p> <p>'0': No change of the respective transmit request bit.</p> <p>'1': The respective transmit request bit is cleared.</p>

n = 0...15

The **Transmit Request Reset Registers** provide a transmit request reset bit (TRR<sub>n</sub>) for each transmit request bit TRS<sub>n</sub>, i.e. for each message object.

Writing '1' to a TRR<sub>n</sub> bit clears the corresponding transmission request bit TRS<sub>n</sub>. This causes a transmission request, initiated by the corresponding bit TRS<sub>n</sub>, to be cancelled, provided that it is not currently processed.

This scheme avoids conflicts when writing to register bits while they are cleared by hardware because of a completed transmission.

**Note:** Registers TRRR<sub>x</sub> cannot be read.

<b>TRRR2</b>	7	6	5	4	3	2	1	0
Address: 19 <sub>H</sub>	<b>TRR15</b>	<b>TRR14</b>	<b>TRR13</b>	<b>TRR12</b>	<b>TRR11</b>	<b>TRR10</b>	<b>TRR9</b>	<b>TRR8</b>
Reset Value: 00 <sub>H</sub>	w	w	w	w	w	w	w	w

<b>TRRR1</b>	7	6	5	4	3	2	1	0
Address: 18 <sub>H</sub>	<b>TRR7</b>	<b>TRR6</b>	<b>TRR5</b>	<b>TRR4</b>	<b>TRR3</b>	<b>TRR2</b>	<b>TRR1</b>	<b>TRR0</b>
Reset Value: 00 <sub>H</sub>	w	w	w	w	w	w	w	w

Bit(field)	Function
<b>TRRx</b>	<b>Transmit Request Reset Bit</b> '0': No change of the respective transmit request bit. '1': The respective transmit request bit is cleared.

**Remote-Request-Pending Registers**

<b>RRPR2</b>	7	6	5	4	3	2	1	0
Address: 1B <sub>H</sub>	<b>RRP15</b>	<b>RRP14</b>	<b>RRP13</b>	<b>RRP12</b>	<b>RRP11</b>	<b>RRP10</b>	<b>RRP9</b>	<b>RRP8</b>
Reset Value: 00 <sub>H</sub>	r	r	r	r	r	r	r	r

<b>RRPR1</b>	7	6	5	4	3	2	1	0
Address: 1A <sub>H</sub>	<b>RRP7</b>	<b>RRP6</b>	<b>RRP5</b>	<b>RRP4</b>	<b>RRP3</b>	<b>RRP2</b>	<b>RRP1</b>	<b>RRP0</b>
Reset Value: 00 <sub>H</sub>	r	r	r	r	r	r	r	r

Bit(field)	Function
<b>RRPn</b>	<b>Remote Request Pending Bit</b> '0': No remote request pending. '1': A remote request (remote frame) for message n was received but is not yet answered by the transmission of the corresponding data frame.

n = 0...15

**Message Time Stamp**

This mechanism stores the time at which a specific message was received, i.e. it assigns a time stamp to that message. For this purpose the contents of the free-running time stamp counter TSC is copied to the time stamp register TSRn of the respective message object upon reception of this message.

**The Time Stamp Counter Registers** provide access to the free-running time stamp counter.

<b>TSCH</b>	7	6	5	4	3	2	1	0
Address: 1C <sub>H</sub>	<b>TSC.15</b>	<b>TSC.14</b>	<b>TSC.13</b>	<b>TSC.12</b>	<b>TSC.11</b>	<b>TSC.10</b>	<b>TSC.9</b>	<b>TSC.8</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>TSCL</b>	7	6	5	4	3	2	1	0
Address: 1D <sub>H</sub>	<b>TSC.7</b>	<b>TSC.6</b>	<b>TSC.5</b>	<b>TSC.4</b>	<b>TSC.3</b>	<b>TSC.2</b>	<b>TSC.1</b>	<b>TSC.0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

Bit(field)	Function
<b>TSC</b>	<b>Time Stamp Counter</b> Current contents of the free running time stamp counter.

**The Time-Stamp Registers** are available for each of message objects 0...7 (see table below) and contain the time-stamp of the corresponding message. These registers can only be read.

<b>TSRnH</b>	7	6	5	4	3	2	1	0
Address: 3X <sub>H</sub>	<b>TSn.15</b>	<b>TSn.14</b>	<b>TSn.13</b>	<b>TSn.12</b>	<b>TSn.11</b>	<b>TSn.10</b>	<b>TSn.9</b>	<b>TSn.8</b>
Reset Value: UU <sub>H</sub>	r	r	r	r	r	r	r	r

<b>TSRnL</b>	7	6	5	4	3	2	1	0
Address: 3X <sub>H</sub>	<b>TSn.7</b>	<b>TSn.6</b>	<b>TSn.5</b>	<b>TSn.4</b>	<b>TSn.3</b>	<b>TSn.2</b>	<b>TSn.1</b>	<b>TSn.0</b>
Reset Value: UU <sub>H</sub>	r	r	r	r	r	r	r	r

Bit(field)	Function
<b>TSn</b>	<b>Time Stamp n</b> A 16-bit timer value to identify the time of reception of message n.

n = 0...7

**Time Stamp Register Table**

Address	Function	
30 <sub>H</sub>	High Byte	Time-Stamp 0
31 <sub>H</sub>	Low Byte	
32 <sub>H</sub>	High Byte	Time-Stamp 1
33 <sub>H</sub>	Low Byte	
:	:	:
3C <sub>H</sub>	High Byte	Time-Stamp 6
3D <sub>H</sub>	Low Byte	
3E <sub>H</sub>	High Byte	Time-Stamp 7
3F <sub>H</sub>	Low Byte	

**Transmit Check Error Counter**

<b>TCEC</b>	7	6	5	4	3	2	1	0
Address: 15 <sub>H</sub>	-	-	-	-	-	<b>TCECV</b>		
Reset Value: 00 <sub>H</sub>	-	-	-	-	-	rw	rw	rw

Bit(field)	Function
<b>TCECV</b>	<b>Transmit Check Error Counter Value</b> Number of errors detected by the transmit check unit. When a count of 4 is reached an interrupt is generated if enabled. If bit TCE (CTRL.1) is set to '1' the Bus Off status is entered in this case.

**Note:** Not defined bit positions must be '0' for write accesses.

**Transmit Check Data Register**

This register supports an error analysis when a transmit check error is encountered. Reading TCD provides the byte which was actually being sent when the error occurred.

<b>TCD</b>	7	6	5	4	3	2	1	0
Address: 16 <sub>H</sub>	<b>Data Byte</b>							
Reset Value: XX <sub>H</sub>	r	r	r	r	r	r	r	r

Bit(field)	Function
<b>Data Byte</b>	The data byte which was attempted to be sent while a transmit check error was encountered.

## Port Control Registers

These registers control the parallel ports P0 and P1 which are provided in the SAE 81C90.

The **Port Direction Registers PxDPR** select each port pin separately for input (PxDPR.n='0') or output (PxDPR.n='1'). After reset the ports are switched as inputs.

<b>P1PDR</b>	7	6	5	4	3	2	1	0
Address: 2C <sub>H</sub>	<b>P1PD.7</b>	<b>P1PD.6</b>	<b>P1PD.5</b>	<b>P1PD.4</b>	<b>P1PD.3</b>	<b>P1PD.2</b>	<b>P1PD.1</b>	<b>P1PD.0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>P0PDR</b>	7	6	5	4	3	2	1	0
Address: 28 <sub>H</sub>	<b>P0PD.7</b>	<b>P0PD.6</b>	<b>P0PD.5</b>	<b>P0PD.4</b>	<b>P0PD.3</b>	<b>P0PD.2</b>	<b>P0PD.1</b>	<b>P0PD.0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

The **Port Latch Registers PxLR** store the output data for those port pins that are switched to output.

<b>P1LR</b>	7	6	5	4	3	2	1	0
Address: 2E <sub>H</sub>	<b>P1L.7</b>	<b>P1L.6</b>	<b>P1L.5</b>	<b>P1L.4</b>	<b>P1L.3</b>	<b>P1L.2</b>	<b>P1L.1</b>	<b>P1L.0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

<b>P0LR</b>	7	6	5	4	3	2	1	0
Address: 2A <sub>H</sub>	<b>P0L.7</b>	<b>P0L.6</b>	<b>P0L.5</b>	<b>P0L.4</b>	<b>P0L.3</b>	<b>P0L.2</b>	<b>P0L.1</b>	<b>P0L.0</b>
Reset Value: 00 <sub>H</sub>	rw	rw	rw	rw	rw	rw	rw	rw

The **Port Pin Registers PxPR** provide the current level of the port pins. These registers can only be read.

<b>P1PR</b>	7	6	5	4	3	2	1	0
Address: 2D <sub>H</sub>	<b>P1P.7</b>	<b>P1P.6</b>	<b>P1P.5</b>	<b>P1P.4</b>	<b>P1P.3</b>	<b>P1P.2</b>	<b>P1P.1</b>	<b>P1P.0</b>
Reset Value: 00 <sub>H</sub>	r	r	r	r	r	r	r	r

<b>P0PR</b>	7	6	5	4	3	2	1	0
Address: 29 <sub>H</sub>	<b>P0P.7</b>	<b>P0P.6</b>	<b>P0P.5</b>	<b>P0P.4</b>	<b>P0P.3</b>	<b>P0P.2</b>	<b>P0P.1</b>	<b>P0P.0</b>
Reset Value: 00 <sub>H</sub>	r	r	r	r	r	r	r	r

In parallel to the standard CMOS structure there are additional internal pullup devices of about 10...200 kΩ at each port pin.

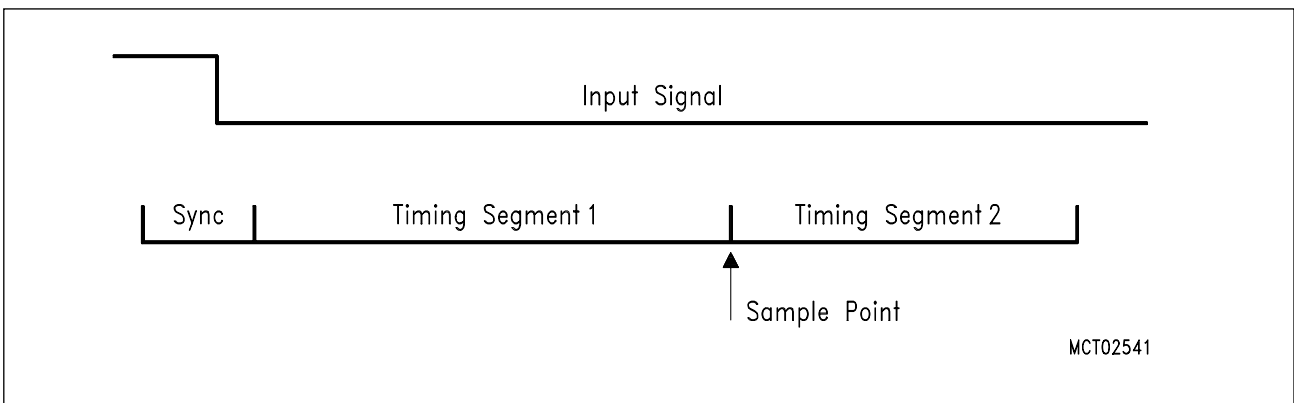
**Note:** Registers PxDPR and PxPL may be used for general purpose storage if the ports are not used.

**Bit Timing**

A regular bit period is composed of the following three segments:

- synchronization segment
- timing segment 1
- timing segment 2.

The sampling point is between timing segment 1 and timing segment 2.



**Figure 6**  
**Bit Time Segments**

**Synchronization**

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

**Timing Segment 1 (TSeg1)**

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).

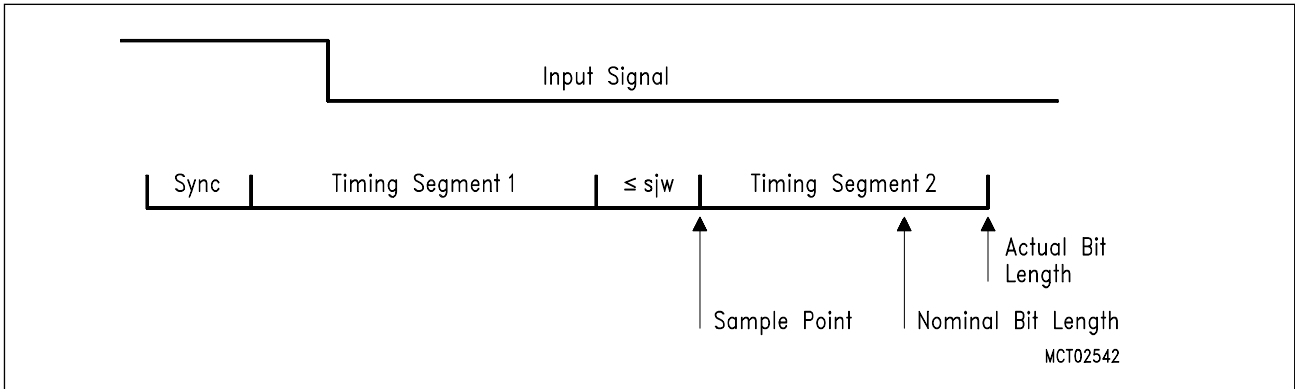
**Timing Segment 2 (TSeg2)**

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).

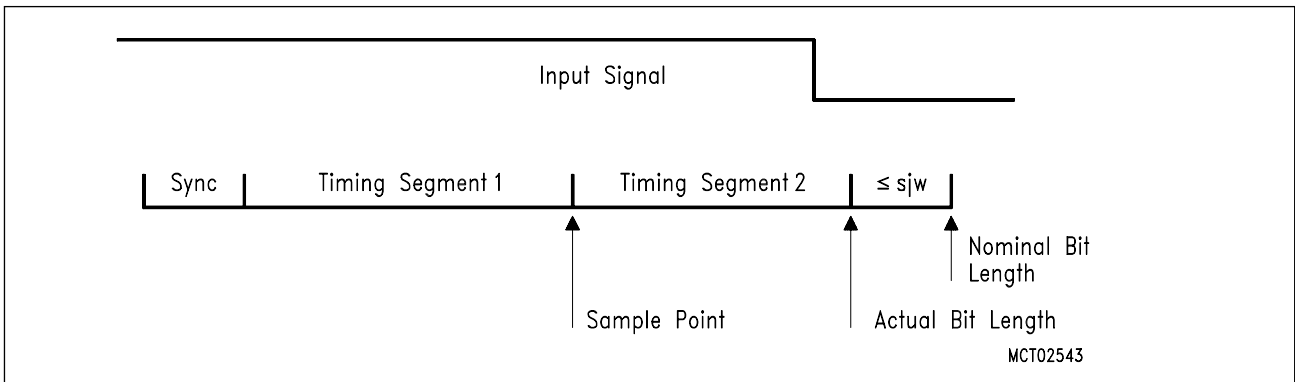
**Synchronization Jump Width (SJW)**

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





**Figure 7**  
Lengthening a Bit Period



**Figure 8**  
Shortening a Bit Period

**Delay Times**

The total delay is calculated from the following single delays:

- $2 \times$  physical bus  $t_{Bus}$  (max. 100 ns acc. to CAN specification)
- $2 \times$  input comparator  $t_{Comp}$  (depends on application circuit)
- $2 \times$  output driver  $t_{Driver}$  (depends on application circuit)
- $1 \times$  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 premise of the stated conditions, there are the following essential requirements to be maintained:

- $t_{TSeg1} \geq t_{Seg2}$
- $t_{TSeg1} \geq t_{Delay}$
- $t_{TSeg2} > t_{SJW}$
- $t_{TSeg2} \geq 3 \times t_{SCL} + t_{SJW}$  if bit SAM = 1 (otherwise bit recognition does not work).

## Host Interfaces

There are two different host interfaces implemented in the SAE 81C90/91.

Data and addresses on a multiplexed 8-bit bus, compatible with Siemens microcontrollers (C5xx, C16x), 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 PI and thus activates pins AD0 through AD7,  $\overline{RD}$ ,  $\overline{WR}$  and ALE.

## Parallel Interface PI

The parallel interface uses 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  $\overline{RD} = 0$ ) via 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.

## Serial Synchronous Interface SI

If the SI is used the unused pins of 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 accomplished 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.

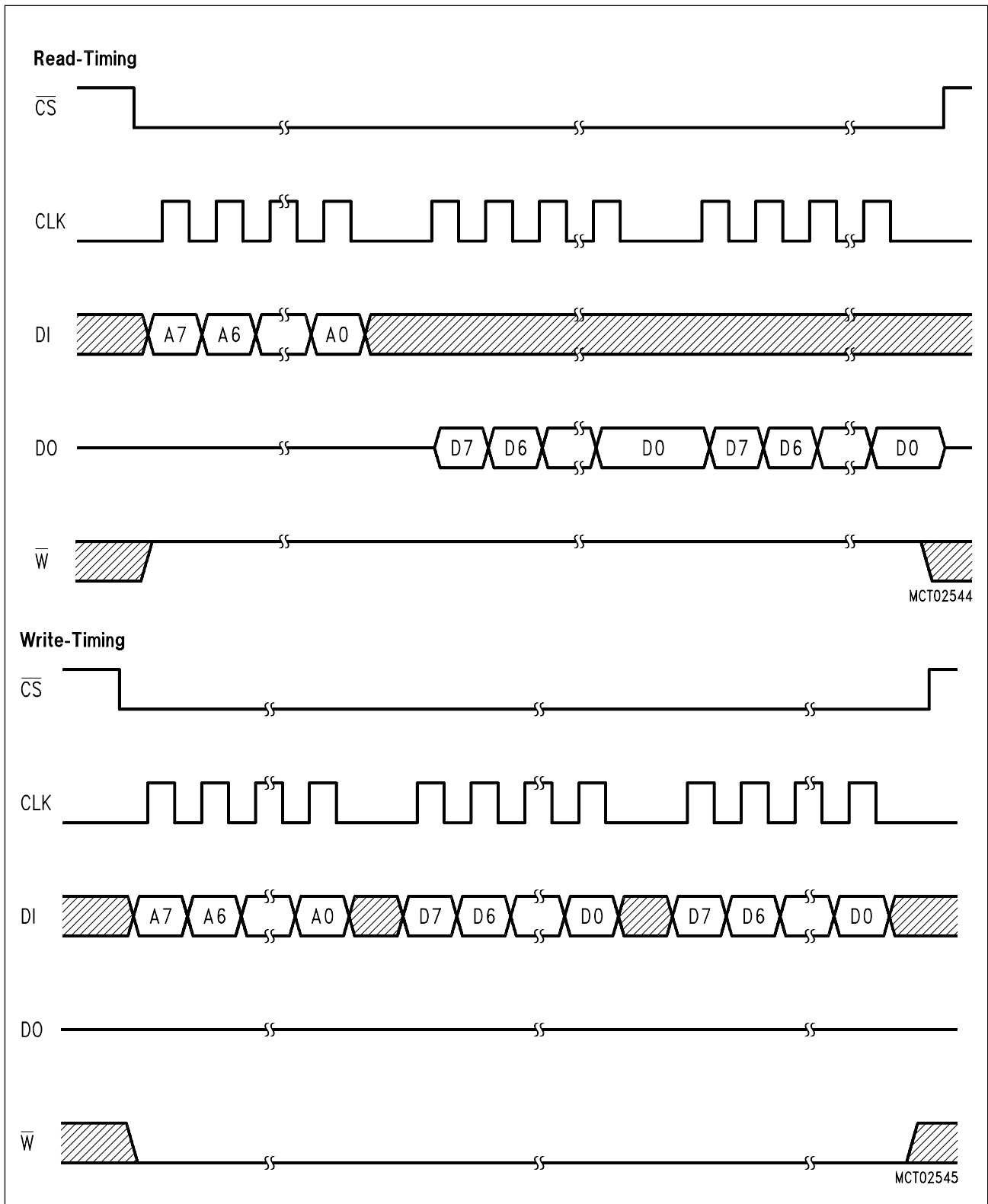
### Procedure:

- 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 **rising** 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 **rising** 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 reading and writing of two data bytes with automatic decrementing activated is illustrated below.



**Figure 9**  
Serial Interface Timing (for 2 Data Bytes)

**Absolute Maximum Ratings**

Ambient temperature under bias ( $T_A$ ): .....	- 40 to + 110 °C
Storage temperature ( $T_{ST}$ ).....	- 50 to + 150 °C
Voltage on $V_{CC}$ pins with respect to ground ( $V_{SS}$ ) .....	- 0.5 to + 6.0 V
Voltage on any pin with respect to ground ( $V_{SS}$ ) .....	- 0.5 to $V_{CC} + 0.5$ V
Input current on any pin during overload condition .....	- 10 to + 10 mA
Absolute sum of all input currents during overload condition .....	100 mA
Power dissipation.....	0.5 W

**Note:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During overload conditions ( $V_{IN} > V_{CC}$  or  $V_{IN} < V_{SS}$ ) the voltage on pins with respect to ground ( $V_{SS}$ ) must not exceed the values defined by the Absolute Maximum Ratings.

**Parameter Interpretation**

The parameters listed in the following partly represent the characteristics of the SAE 81C90/91 and partly its demands on the system. To aid in interpreting the parameters right, when evaluating them for a design, they are marked in column “Symbol”:

**DC (Device Characteristics):**

The logic of the SAE 81C90/91 will provide signals with the respective timing characteristics.

**SR (System Requirement):**

The external system must provide signals with the respective timing characteristics to the SAE 81C90/91.

## DC Characteristics

$$V_{CC} = 5 \text{ V} \pm 10 \% ; \quad V_{SS} = 0 \text{ V}$$

$$T_A = -40 \text{ to } +110 \text{ }^\circ\text{C}$$

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (all except XTAL1 and XTAL2)	$V_{IL}$ SR	0	$0.3 V_{CC}$	V	–
Input low voltage (XTAL1 and XTAL2)	$V_{ILX}$ SR	0	0.5	V	–
Input high voltage (all except XTAL1 and XTAL2)	$V_{IH}$ SR	$0.7 V_{CC}$	$V_{CC}$	V	–
Input high voltage (XTAL1 and XTAL2)	$V_{IHx}$ SR	$V_{CC} - 1.0$	$V_{CC}$	V	–
Comparator input voltage <sup>1)</sup>	$V_{CI}$ SR	0.5	$V_{CC} + 0.5$	V	–
Common mode voltage <sup>2)</sup>	$V_{ICOM}$ SR	1.5	$V_{CC} - 1.5$	V	–
Hysteresis <sup>2)</sup>	$V_{HYS}$ DC	–	100 <sup>3)</sup>	mV	–
Offset voltage <sup>2)</sup>	$V_{OFF}$ DC	–	100 <sup>3)</sup>	mV	–
Output low voltage (all except CLKOUT, TX0, TX1)	$V_{OL}$ DC	–	$0.2 V_{CC}$	V	$I_{OL} = 1.6 \text{ mA}$
Output low voltage (CLKOUT)	$V_{OLC}$ DC	–	0.4	V	$I_{OL1} = 10 \text{ mA}$
Output high voltage (all except CLKOUT, TX0, TX1)	$V_{OH}$ DC	$0.8 V_{CC}$	$V_{CC}$	V	$I_{OH} = -1.6 \text{ mA}$
Output high voltage (CLKOUT)	$V_{OHC}$ DC	$V_{CC} - 0.8$	$V_{CC}$	V	$I_{OH} = -10 \text{ mA}$
Input leakage current	$I_I$ DC	–	$\pm 1$	$\mu\text{A}$	$0 \text{ V} < V_{IN} < V_{CC}$ <sup>4)</sup>
Source output current (TX0, TX1)	$I_{SRC}$ DC	5	–	mA	$V_O = V_{CC} - 1 \text{ V}$
Sink output current (TX0, TX1)	$I_{SNK}$ DC	5	–	mA	$V_O = 1 \text{ V}$
Low end capacitance <sup>5)</sup>	$C_L$ DC	6.8	12	pF	
Pin capacitance <sup>2)</sup>	$C_I$ DC	–	10	pF	$f = 1 \text{ MHz}$ $T_A = 25 \text{ }^\circ\text{C}$
Power supply current	$I_{CC}$	–	30	mA	

### Notes

- 1) If the bus lines work according to the ISO specification, additional circuitry is necessary for interconnection of the input comparator to the bus lines.
- 2) Not 100% tested, guaranteed by design characterization.
- 3) This value is a typical value!
- 4) This specification does not apply to the port pins (P00...P07, P10...P17) due to the implemented pullups!
- 5) In oscillator mode the size of the low-end capacitance must correspond to the specification of the crystal manufacturer. The optimum values depend on the selected crystal, the intended frequency and the actual application hardware (stray capacitances). 10 pF are recommended for  $C_L$ .  
For best results keep the crystal circuitry connections as short as possible and keep the CLKOUT line away from it. If the CLKOUT signal is not required by the system it should be switched off.

## AC Characteristics (General Timing)

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$   
 $T_A = -40\text{ to }+110\text{ }^\circ\text{C}$

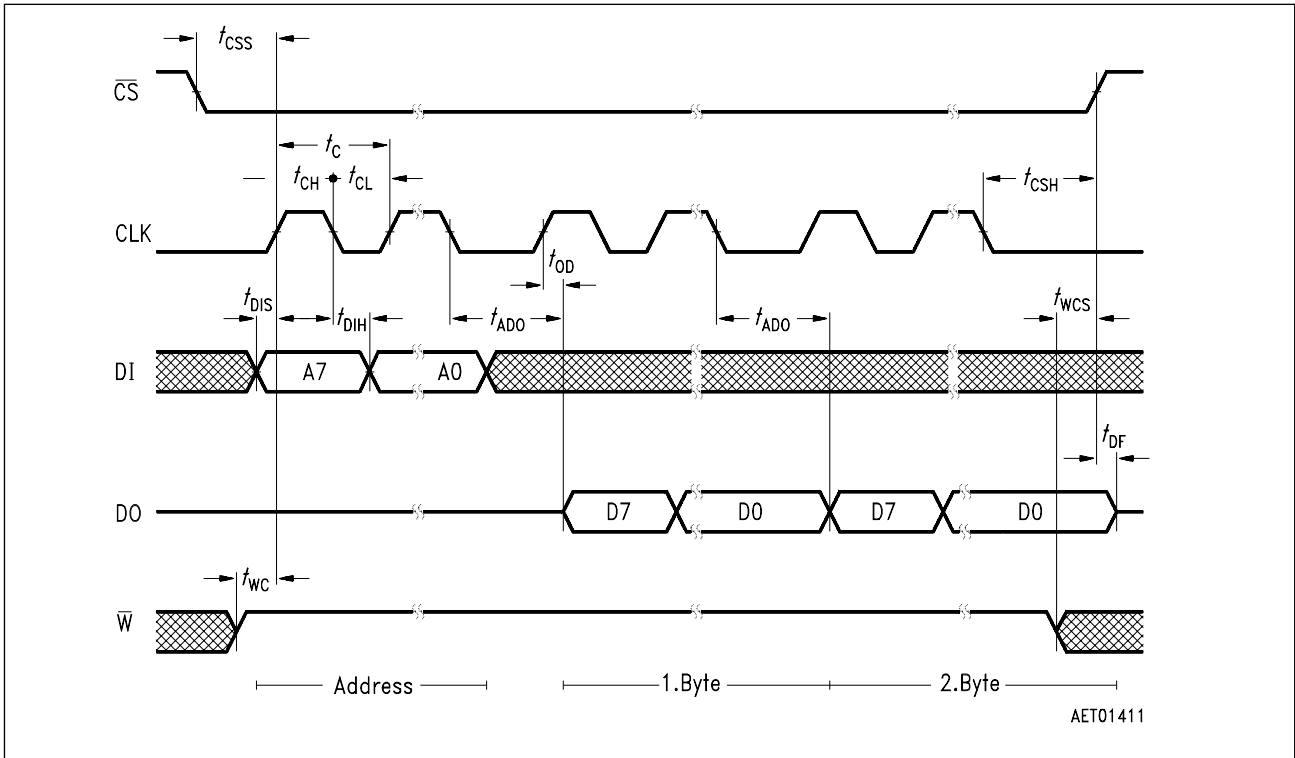
Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Oscillator period	$t_{OSC}$ SR	50	-	ns	
Clock input high time	$t_H$ SR	23.5	-	ns	
Clock input low time	$t_L$ SR	23.5	-	ns	
Reset pulse width	$t_{RES}$ SR	2	-	$t_{OSC}$	
Output rise time <sup>1)</sup>	$t_{QR}$ DC	-	40	ns	$C_L = 70\text{ pF}$
Output fall time <sup>1)</sup>	$t_{QF}$ DC	-	40	ns	$C_L = 70\text{ pF}$
CLKOUT rise time <sup>1)</sup>	$t_{QRC}$ DC	-	20	ns	$C_L = 50\text{ pF}$
CLKOUT fall time <sup>1)</sup>	$t_{QFC}$ DC	-	20	ns	$C_L = 50\text{ pF}$

<sup>1)</sup> Not 100% tested, guaranteed by design characterization.

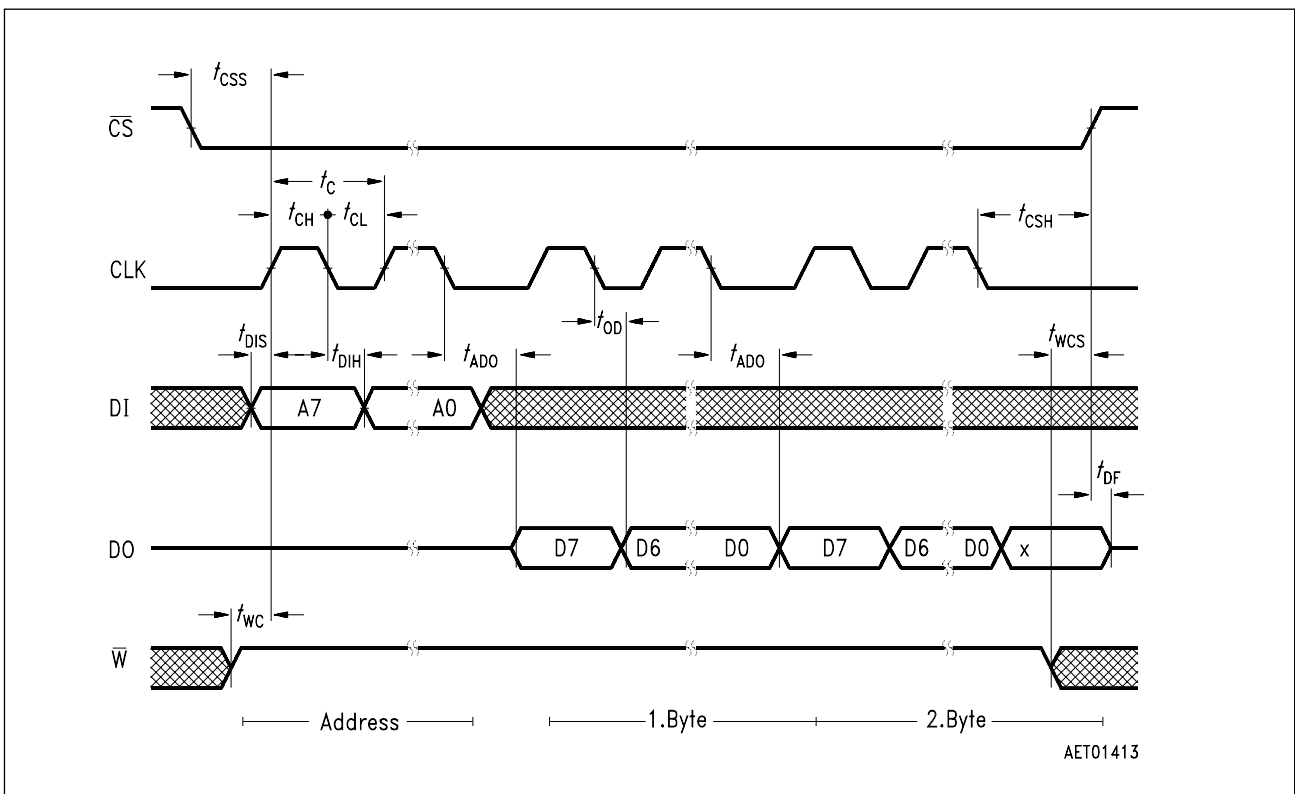
## AC Characteristics (SI Timing)

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$   
 $T_A = -40\text{ to }+110\text{ }^\circ\text{C}$ ;  $C_L = 50\text{ pF}$

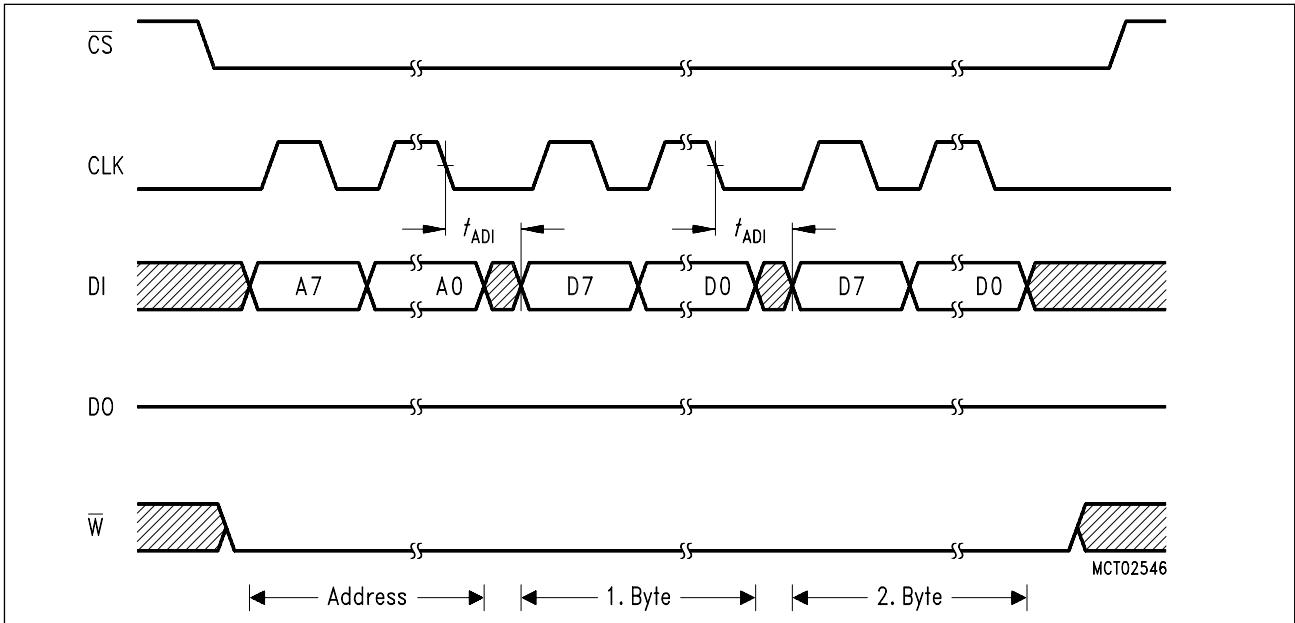
Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Chip Select Setup	$t_{CSS}$ SR	10		ns	
Clock High Time	$t_{CH}$ SR	$1.5 t_{OSC} + 10$		ns	
Clock Low Time	$t_{CL}$ SR	$1.5 t_{OSC} + 10$		ns	
Clock Period	$t_C$ SR	$4 t_{OSC}$		ns	
DI Setup	$t_{DIS}$ SR	10		ns	
DI Hold	$t_{DIH}$ SR	0		ns	
Address to Data Out	$t_{ADO}$ DC	$3 t_{OSC}$		ns	
Output Delay	$t_{OD}$ DC		25	ns	
Data Float after $\overline{CS}$ high	$t_{DF}$ DC		25	ns	
Chip Select Hold	$t_{CSH}$ SR	$1 t_{OSC}$		ns	
Write to Clock	$t_{WC}$	0		ns	
$\overline{W}$ to $\overline{CS}$ high	$t_{WCS}$ SR	0		ns	
Address to Data In	$t_{ADI}$ DC	0		ns	



**Figure 10**  
SI-Read-Timing (Timing A: Pin TIM = 0)



**Figure 11**  
SI-Read-Timing (Timing B: Pin TIM = 1)



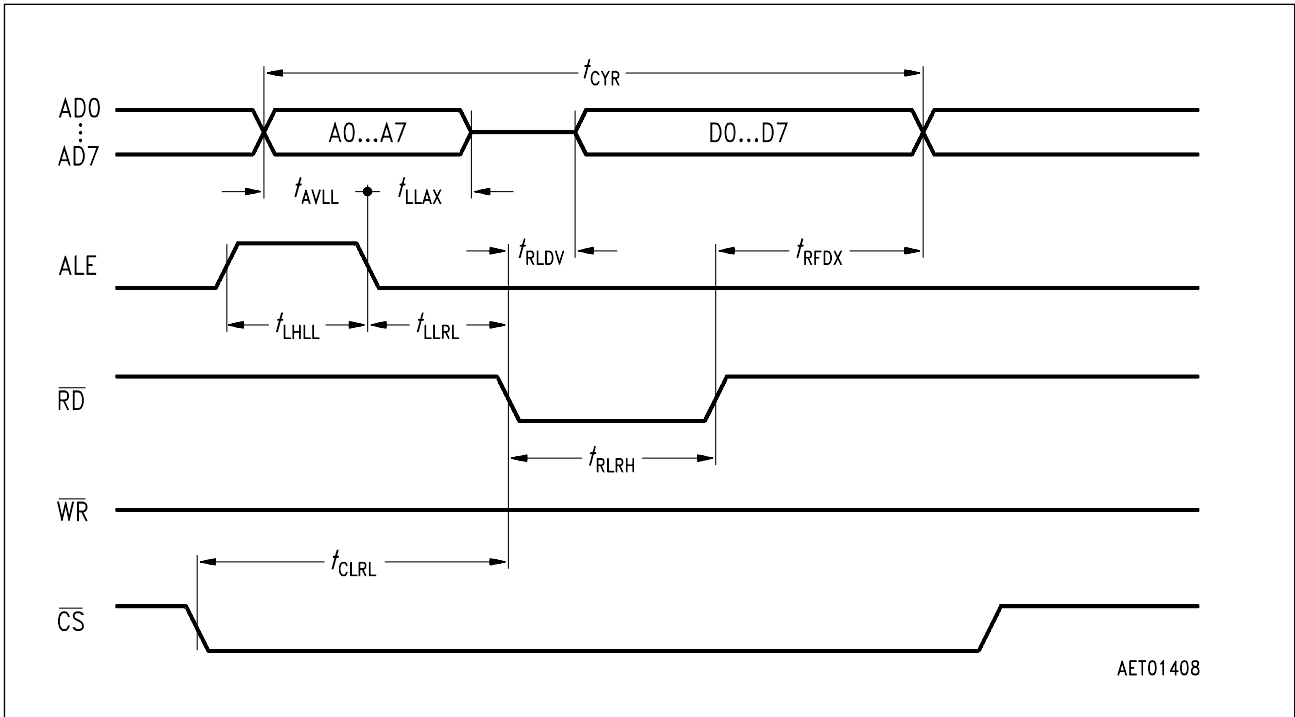
**Figure 12**  
**SI-Write-Timing**

**AC Characteristics (PI Timing)**

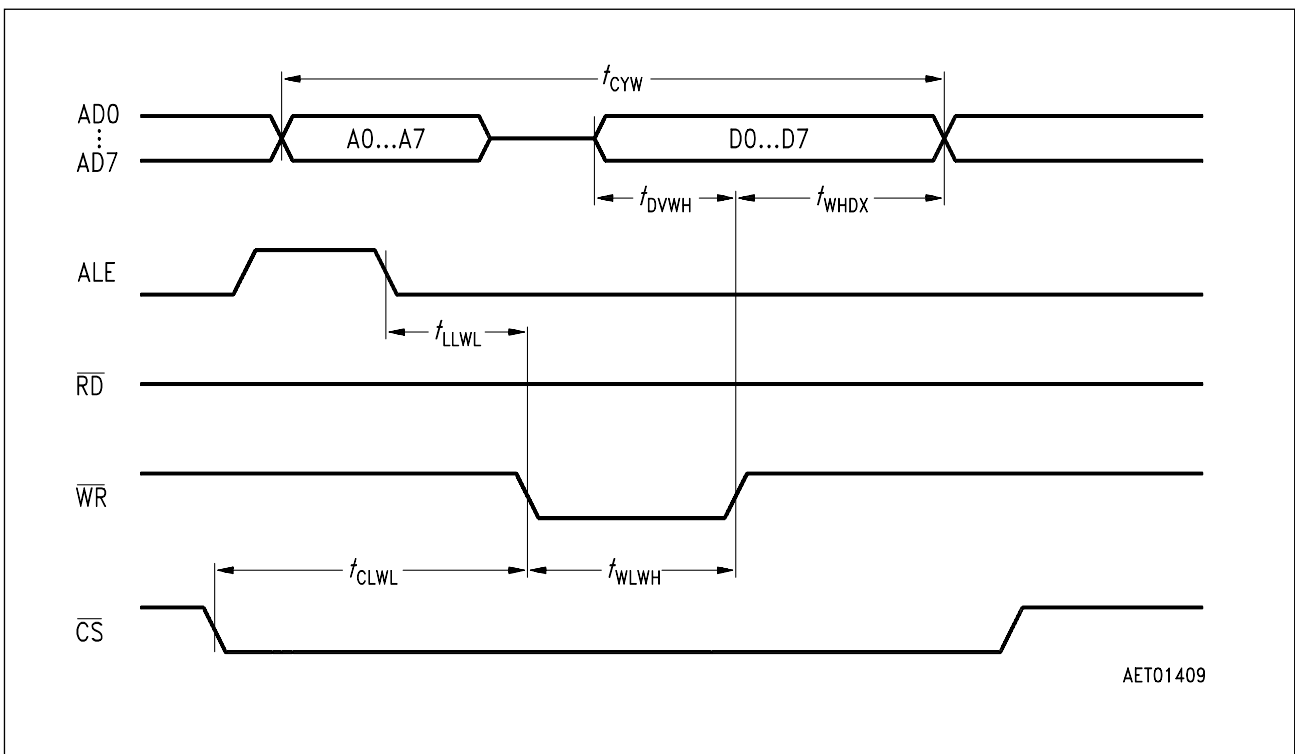
$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$   
 $T_A = -40\text{ to }+110\text{ }^\circ\text{C}$ ;  $C_L = 50\text{ pF}$

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Read-Cycle time	$t_{CYR}$ DC	$4 t_{OSC}$		ns	
Write-Cycle time	$t_{CYW}$ DC	$4 t_{OSC}$		ns	
ALE pulse width	$t_{LHLL}$ DC	30		ns	
Address setup to ALE low	$t_{AVLL}$ SR	10		ns	
Address hold after ALE low	$t_{LLAX}$ SR	10		ns	
RD pulse width	$t_{RLRH}$ SR	$2 t_{OSC} + 30$		ns	
WR pulse width	$t_{WLWH}$ SR	$2 t_{OSC} + 30$		ns	
ALE low to WR active	$t_{LLWL}$ SR	20		ns	
ALE low to RD active	$t_{LLRL}$ SR	20		ns	
Data float after RD high	$t_{RFDX}$ DC	0	20	ns	
RD low to data valid	$t_{RLDV}$ DC		$2 t_{OSC} + 20$	ns	
Data setup before WR high	$t_{DVWH}$ SR	10		ns	
Data hold after WR high	$t_{WHDX}$ SR	5		ns	
CS low to RD low	$t_{CLRL}$ / SR	0		ns	
CS low to WR low	$t_{CLWL}$ SR	0		ns	
WR high to next ALE low	$t_{WHLL}$ SR	$1.5 t_{OSC}$		ns	





**Figure 13**  
PI Timing: Read-Cycle-Timing



**Figure 14**  
PI Timing: Write-Cycle-Timing