TEMIC
Semiconductors

TSS461C

Vehicle Area Network Data Link Controller

1. Description

Cost optimization in car manufacturing is of extreme importance today. Solutions to this problem often implies the use of more advanced and intelligent electronic circuits.

The TSS461C is such a circuit, which allows the transfer of all the status information needed in a car or truck over a single low-cost wire pair, thereby minimizing the electrical wire usage.

It can be used to interconnect powerful functions (ABS, dashboard, power train control) and to control and interface car body electronics (lights, wipers, power window...).

The TSS461C is fully compliant with the ISO standard ISO/11519-3. This standard supports a wide range of applications such as low-cost remote controlled switches, typically used for lamp control, up to complex, highly autonomous, distributed systems like engine controls, which require fast and secure data transfers.

The TSS461C is a microprocessor interfaced line controller for mid to high complexity bus-masters and listeners like injection/ignition control calculators, dashboard controllers and car stereo or mobile telephone CPUs.

The microprocessor interface consists of a 256 byte RAM and register area divided into 11 control registers, 14 channel register sets and 128 bytes of general purpose RAM, used as a message storage area, and a 6-source maskable interrupt.

The circuit operates in the RAM using DMA techniques, controlled by the channel and control registers. This allows virtually any microprocessor to interface with ease to the TSS461C, and to use the free RAM as a scratch pad.

Messages are encoded in enhanced Manchester code, and an optional pulsed code for use with an optical or radio link, at a maximum bit rate of 1 Mbit/s. The TSS461C analyzes the messages received or transmitted according to 6 different criterias including some higher level checks.

In addition the bus interface has three separate inputs with automatic source diagnosis and selection, allowing for multibus listening or the automatic selection of the most reliable source at any time if several line receivers are connected to the same bus.

2. Features

- Fully compliant to VAN specification ISO/11519.3.
- Handles all specified module types.
- Handles all specified message types.
- Handles retransmission of frames on contention and errors.
- 3 separate line inputs with automatic diagnosis and selection.
- 1 Mbit/s maximum transfer rate.
- Normal or pulsed (optical and radio mode) coding.
- INTEL, NEC, TI and MOTOROLA compatible 8-bit microprocessor interface.
- Multiplexed address and data bus.
- Idle and sleep modes.
- 128 bytes of general purpose RAM.
- DMA capabilities for message handling.
- 14 identifier registers with all bits individually maskable.
- 6-source maskable interrupt including an interrupt-on-reset to detect glitches on the reset pin.
- Integrated crystal or resonator oscillator with internal baud rate generator and buffered clock output.
- Single +5V power supply.
- 0.8 µm CMOS technology.
- SOP 24 packaging.

Preliminary Information

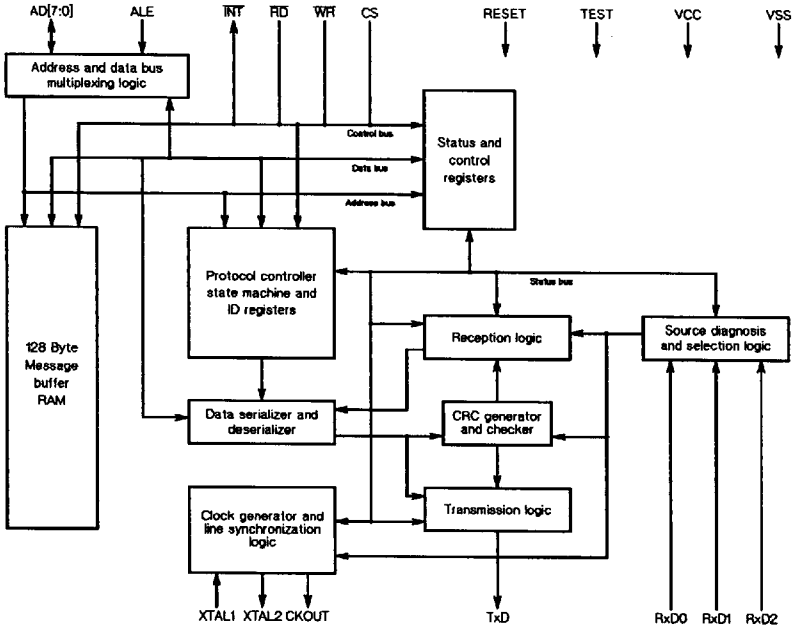
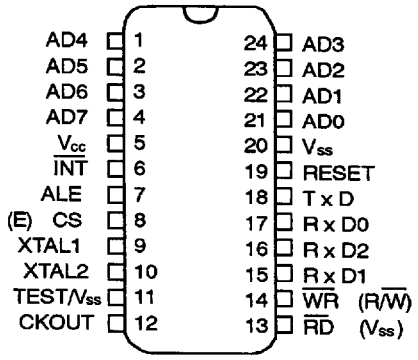


Figure 1. Block Diagram



24 pin SOP

I/O Type	Pin Name	Pin No	Pin Function
I/O TTL	AD0	21	Multiplexed address and data bus. The address is latched on the falling edge of ALE.
	AD1	22	
	AD2	23	
	AD3	24	
	AD4	1	
	AD5	2	
	AD6	3	
	AD7	4	
I trigger TTL	ALE	7	Address Latch Enable
	$\overline{RD}$ (Vss)	13	Read command
	$\overline{WR}$ (R/$\overline{W}$)	14	Write command
	CS (E)	8	Chip select (active high)
Open drain	$\overline{INT}$	6	Interrupt
I trigger CMOS pulldown	RESET	19	Asynchronous general reset - glitch filtered ($\simeq$ 12ns)
I CMOS Pull down	RxD0	17	VAN bus inputs
	RxD1	15	
	RxD2	16	
3–state	TxD	18	VAN bus output
I	XTAL1	9	Crystal oscillator or clock input pins
O	XTAL2	10	
O	CKOUT	12	Buffered clock output
Ground	TEST/Vss	11	Oscillator ground
Power	VCC	5	+ 5 V power supply
Ground	vss	20	

Figure 2. Pinout

3. Operation

The TSS461C is a microprocessor controlled line controller for the VAN bus. It can interface to virtually any microprocessor, but the I/O signals of the circuit have been optimized for use with the TEMIC TSC51/TSC251 series of microcontrollers.

This means that it features a multiplexed address and data bus, controlled by an address strobe pin ALE and separated read $\overline{RD}$ and write $\overline{WR}$ command pins. The address is latched on the falling edge of ALE.

The circuit also features one single interrupt pin. This pin can be treated as level or edge sensitive, i.e. if there is a pending interrupt inside the circuit when another interrupt is reset the $\overline{INT}$ pin will emit a high pulse with the same pulse width as the internal write strobe (typically 20 ns).

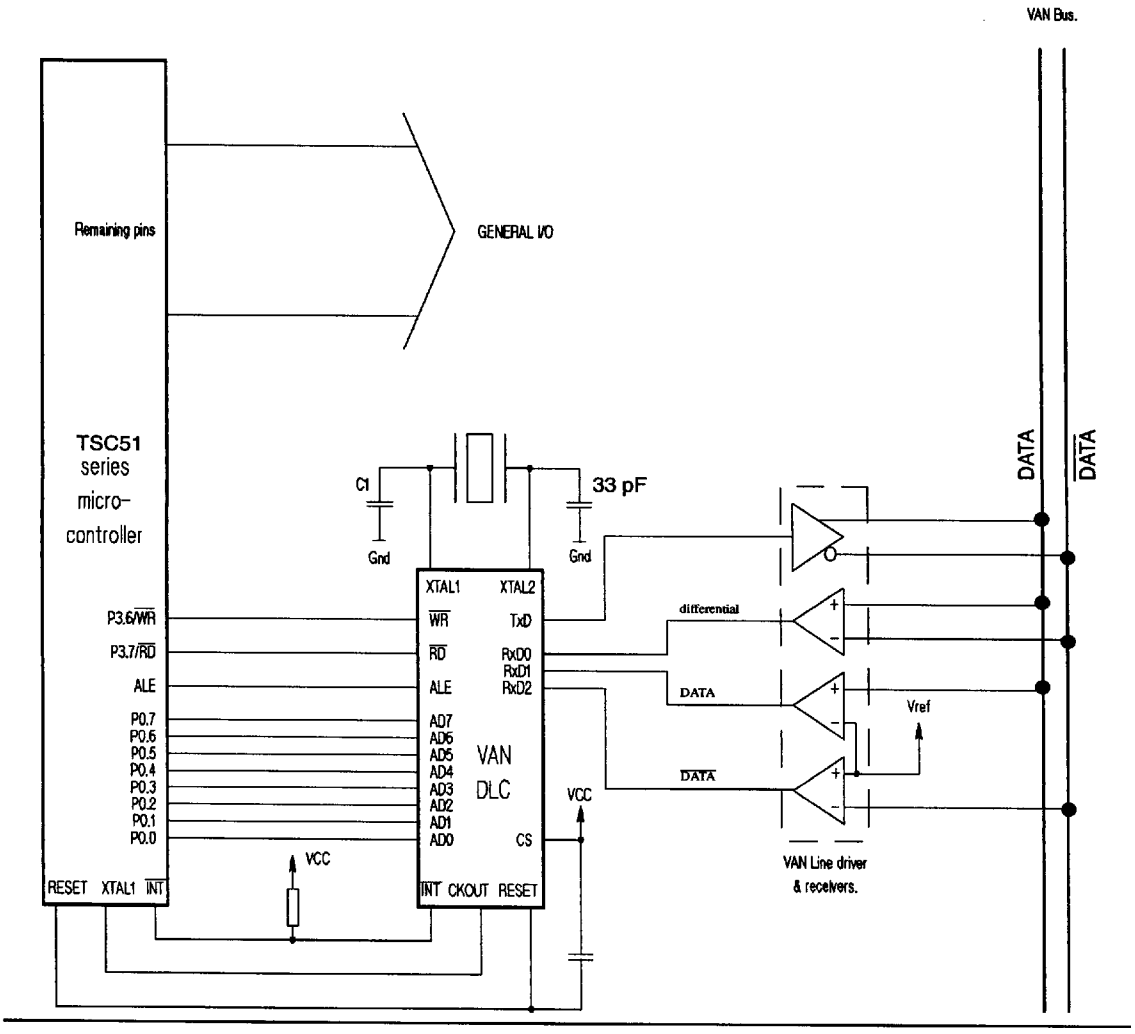


Figure 3. Typical Application

4. Pinout

The TSS461C is available in SOP 24 package. Figure 2. shows the pinout. The names in parenthesis refer to the functionalities in MOTOROLA mode.

Possibilities exist to supply the TSS461C in a PDIL 24 package.

5. Microprocessor Interface

The processor controls the TSS461C by reading and writing the internal registers of the circuit. These registers appear to the processor as regular memory locations.

5.1. Interface Modes

The TSS461C must be plugged in an INTEL or MOTOROLA environment with an 8–bit address/data bus multiplexed.

Table 1. Access Mode Logic

CS (E)	$\overline{RD}$	$\overline{WR}$ (R/$\overline{W}$)	Operation Mode
0			No operation
1	0	0	Write operation in MOTOROLA mode
1	0	1	Read operation in both mode
1	1	0	Write operation in INTEL mode
1	1	1	No operation

In INTEL environment, access operations need CS active, a read one with $\overline{RD}$ active, a write one with $\overline{WR}$ active. If TSS461C is the single peripheral in the processor space, CS can be hired to Vcc.

In MOTOROLA environment, the $\overline{RD}$ pin is hired to VSS and the access operations are driven by CS (E). Contrrary to INTEL mode, CS (E) must never be hired to Vcc even if the TSS461C is alone.

To switch on the fly from one mode to the other, CS must be inactive.

5.1.1. INTEL Mode

The INTEL mode interface consists of 13 pins. 8 pins are the multiplexed address and data bus, and the rest are the address strobe, the read and write commands, the chip select and the interrupt request pins.
To access the memory locations in INTEL mode, the processor must first assert a valid address on the multiplexed address and data bus and drive the address strobe pin high. When the required set-up time has passed the processor must drive the address strobe low, and keep the address valid for the required hold time.
The processor must then either assert the data to be written on the address and data bus, if a write is intended, or float the data bus for a read. The next step is to drive either the write or read command pins low, according to the function required, and at the same time drive the chip select pin high.
The TSS461C access cycle is then terminated by driving the chip select and command pins low.

Note, that the chip select pin may be driven high for the entire access cycle, and may also remain high during and after the termination of the cycle.

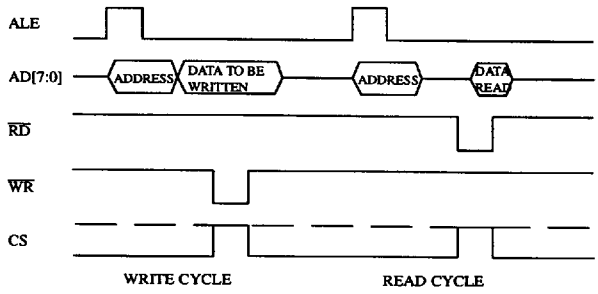


Figure 4. INTEL Read and Write Cycles.

5.1.2. MOTOROLA Mode

In MOTOROLA mode the $\overline{WR}$ pin becomes the R/$\overline{W}$ command, the $\overline{RD}$ pin must be connected to ground and the CS pin becomes the E strobe. This means that there is no separate chip select input, i.e. if some external decoder is to be used, this decoder should not drive the E input high unless the processors E output is high as well.
Please refer to Figure 5. for the MOTOROLA read and write cycles. The main difference between INTEL and MOTOROLA mode is that the timing in INTEL mode is referenced to the command signals ($\overline{RD}$ and $\overline{WR}$), but in MOTOROLA mode the reference is the E signal.

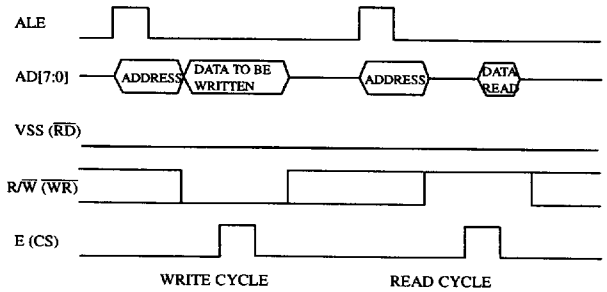


Figure 5. MOTOROLA Read and Write Cycles.

5.2. Interrupts

If an event occurs in the TSS461C, that needs the attention of the processor, this will be signalled on the active low, open drain interrupt request pin. Which events that create such a request is controlled by the internal registers.
Every time the microprocessor accesses any of the interrupt registers (addresses 0x08 to 0x0B) the $\overline{INT}$ pin will be released momentarily. This enables the TSS461C to work with processors that have either edge or level sensitive interrupt inputs.

5.3. Reset

The reset is applied asynchronously regarding XTAL clock. It can be done either by the RESET pin or by software.
The RESET pin is a CMOS trigger input with a pull-down resistor ($\simeq$ 110Ω). An external 1 µF capacitor to Vcc provides to RESET pin an efficient behavior.
The software reset is made through the GRES command bit of the Command Register (0x03).
The two resets are ored, filtered and gauging. Then the internal reset, always asserted asynchronously, enables the internal oscillator. Then it waits for eight clock periods the oscillator stability.
The different blocks of the TSS461C need to be turned on synchronously. So the release of the internal reset is synchronous and a loose of clock can let the TSS461C in permanent reset after applying Reset.

6. Oscillator

An oscillator is integrated in the TSS461C, and consists of an inverting amplifier of which the input is XTAL1 and the output XTAL2.
A parallel resonance quartz crystal or ceramic resonator must be connected to these pins. As can be seen from Figure 3., two capacitors have to be connected from the crystal pins to ground. The values of C1 depend on the frequency chosen and can be selected using the nomograph given in Figure 34.
If the oscillator is not used, then a clock signal must be fed to the circuit via the XTAL1 input.
Note, that this pin will behave as a CMOS level compatible Schmitt trigger input.
In this case the XTAL2 output should be left unconnected. The oscillator also features a buffered clock output pin CKOUT. The signal on this pin is directly buffered from the XTAL1 input, without inversion.

There is one more pin used for the oscillator. The TEST/VSS pin is in fact its ground, and unless this pin is firmly connected to ground, with decoupling capacitors, the oscillator will not operate correctly.
The test mode itself, i.e. when the TEST/VSS pin is held high, is only intended for factory use, and the functionality of this mode is not specified in any way.
Furthermore, it is subject to change without notice, the only exception beeing for incoming inspection tests using the TEMIC test program.
The clock signal is then fed to the clock generator that generates all the necessary timing signals for the operation of the circuit. The clock generator is controlled by a 4-bit code called the clock divider.

$$f(\text{TSCLK}) = \frac{f(\text{XTAL1})}{n \times 16}$$

Table 2. Clock Divider.

Clock Divider	Divide by	32 MHz		24 MHz		16 MHz	
		Ktimes lot /s	Kbits /s	Ktimes lot /s	Kbits /s	Ktimes lot /s	Kbits /s
0000	1					1000	800
0001	2	1000	800	750	600	500	400
0010	4	500	400	375	300	250	200
0011	8	250	200	187.5	150	125	100
0100	16	125	100	93.75	75	62.50	50
0101	32	62.5	50	46.875	37.5	31.25	25
0110	64	31.25	25	23.438	18.75	15.625	12.5
0111	128	15.625	12.5	11.718	9.375	7.813	6.25
1000	1.5	1333	1067	1000	800	666.667	533.333
1001	3	666.667	533.333	500	400	333.333	266.666
1010	6	333.333	266.666	250	200	166.666	133.333
1011	12	166.666	133.333	125	100	83.333	66.666
1100	24	83.333	66.666	62.5	50	41.666	33.333
1101	48	41.666	33.333	31.25	25	20.833	16.666
1110	96	20.833	16.666	15.625	12.5	10.416	8.333
1111	192	10.416	8.333	7.813	6.125	5.208	4.166

Table 3. Clock Divider (continue).

Clock Divider	Divide by	12 MHz		8 MHz		4 MHz	
		Ktimes lot /s	Kbits /s	Ktimes lot /s	Kbits /s	Ktimes lot /s	Kbits /s
0000	1	750	600	500	400	250	200
0001	2	375	300	250	200	125	100
0010	4	187.50	150	125	100	62.50	50
0011	8	93.75	75	62.5	50	31.25	25
0100	16	46.875	37.5	31.25	25	15.625	12.5
0101	32	23.438	18.75	15.625	12.5	7.813	6.25
0110	64	11.718	9.375	7.813	6.25	3.906	3.125
0111	128	5.859	4.688	3.906	3.125	1.953	1.562
1000	1.5	500	400	333.333	266.666	166.666	133.333
1001	3	250	200	166.666	133.333	83.333	66.666
1010	6	125	100	83.333	66.666	41.666	33.333
1011	12	62.50	50	41.666	33.333	20.833	16.666
1100	24	31.25	25	20.833	16.666	10.416	8.333
1101	48	15.625	12.50	10.416	8.333	5.208	4.166
1110	96	7.813	6.25	5.208	4.166	2.604	2.083
1111	192	3.906	3.125	2.604	2.083	1.302	1.042

7. VAN Protocol

7.1. Line Interface

There are three line inputs and one line output available on the TSS461C. Which of the three inputs to use is either programmable by software or automatically selected by a diagnosis system.
The diagnosis system continuously monitors the data received through the three inputs, and compares it with each other and the selected bitrate. It then chooses the most reliable input according to the results.

The data on the line is encoded according to the VAN standard ISO/11519-3. This means that the TSS461C is using a two level signal having a recessive (1) and a dominant (0) state. Furthermore, due to the simple medium used, all data transmitted on the bus is also received simultaneously.
The VAN protocol is hence a CSMA/CD (Carrier Sense Multiple Access/Collision Detection) protocol, allowing for continuous bitwise arbitration of the bus, and non-destructive (for the higher priority message) collision detection.

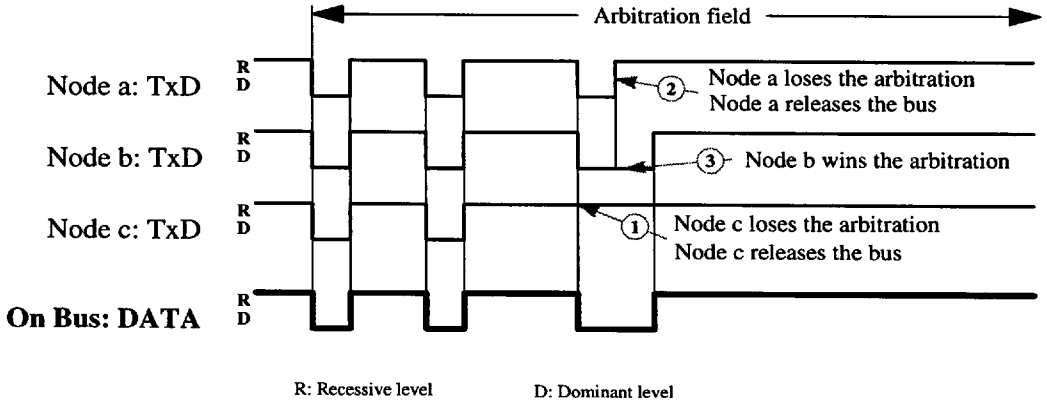


Figure 6. CSMA/CD Arbitration

In addition to the VAN specification there is also a pulsed coding of the dominant and recessive states. This mode is intended to be used with an optical or radio link. In this mode the dominant state for the transmitter is a low pulse, (2x prescaled clocks at the beginning of the bit) and the recessive state is just a high level.
When receiving in this mode it is not the state of the signal itself which is decoded, but the edges. Also, reception is imposed on the RxD0 input, and the diagnosis system does not operate correctly.
In addition in this mode there is an internal loopback in the circuit since optical transceivers are not able to receive the signal that they themselves transmit.

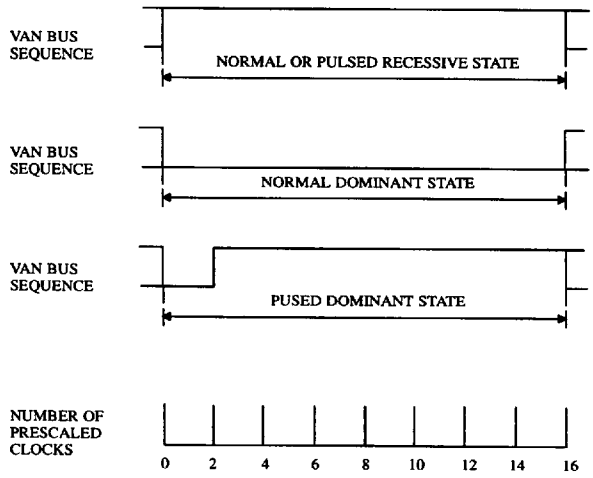


Figure 7. State Encoding.

In Figure 7. the pulsed waveforms are shown. In Figure 10. through Figure 16. the low "timeslots" (i.e. blocks of 16 prescaled clocks) should be replaced by the dominant waveform showed in Figure 7. , if the correct representations for pulsed coding are to be seen.

7.2. VAN Frame

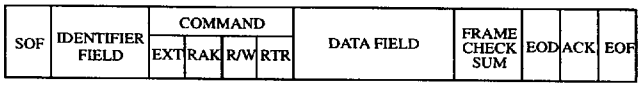


Figure 8. Van Bus Frame.

The VAN bus supports three different module (unit) types:

- First, the *Autonomous* module, which is a bus master. It can transmit Start Of Frame (SOF) sequences, it can initiate data transfers and can receive messages.
- Second, the *Synchronous access* module. It cannot transmit SOF sequences, but it can initiate data transfers and can receive messages.
- And finally, the *Slave* module, which can only transmit using an in-frame mechanism and can receive messages.

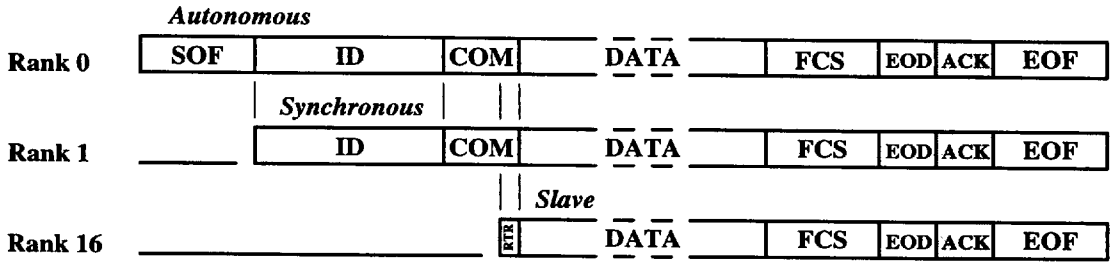


Figure 9. Hierarchical Access Methods

Figure 8. shows a normal VAN bus frame. It is initiated with a Start Of Frame (SOF) sequence shown in Figure 10. The SOF can only be transmitted by an autonomous module. During the preamble the TSS461C will synchronize its bit rate clock to the data received.

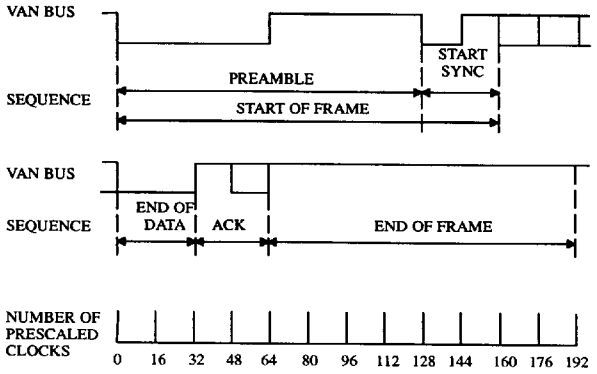


Figure 10. Framing Sequences.

When the complete SOF sequence has been transmitted or received, the circuit will start the transmission or reception of the identifier field.
All data on the VAN bus, including the identifier and Frame Check Sum (FCS), are transmitted using enhanced Manchester code.
In enhanced Manchester code three NRZ bits are transmitted first followed by one Manchester bit, then three more NRZ bits followed by one Manchester bit and so on.
Since the high state is recessive and the low state is dominant, the bus arbitration can be done. If a module wants access to the bus, it must first listen to the bus during one full End Of Frame (EOF) and one full Inter Frame Spacing (IFS) period, to determine whether the bus is free or not (i.e. no dominant states received).

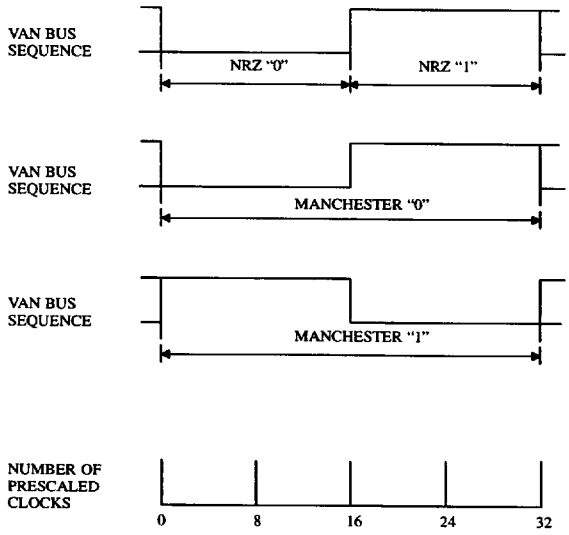


Figure 11. Data Encoding.

The IFS is defined to be a minimum of 64 prescaled clocks periods. The TSS461C, accepts an IFS of zero prescaled clocks for the reception only of a SOF sequence.
Once the bus has been determined as being free, the module must now, if it is an autonomous module, emit a SOF sequence or, if it is a synchronous access module, wait until it detects a preamble sequence.
Up till this point there can be several modules transmitting on the bus, and there is no possibility of knowing if this is the case or not. Therefore the first field in wich arbitration can be performed is the identifier field. Since the logical zeroes on the bus are dominant, and all data is transmitted with the most significant bit (MSB) first, the first module to transmit a logical zero on the bus will be the prioritized module, i.e. the message that is tagged with the lowest identifier will have priority over the other messages.
It is, however, conceivable that two messages transmitted on the bus will have the same identifier. The TSS461C therefore continues the arbitration of the bus throughout the whole frame. More, if the identifier in transmission has been programmed for reception as well, it transmits and receives messages simultaneously, right up till the Frame Check Sequence (FCS). Only then, if the TSS461C has transmitted the whole message, does it discard the message received. Arbitration loss in the FCS field is considered as a CRC error during transmission.
This feature is called full data field arbitration, and it enables the user to extend the identifier. For instance it can be used to transmit the emitting modules address in the first bytes of the data field, thus enabling the identifier to specify the contents of the frame and the data field to specify the source of the information.

The identifier field of the VAN bus frame is always 12 bits long, and it is always followed immediately by the 4-bit command field:

- The first bit of the command is the extension bit (EXT). This bit is defined by the user on transmission and is received and retained by the TSS461C. To conform with the standard it should be set to 1 (recessive) by the user, else the frame is ignored without any IT generation.
- The second bit is the request ACKnowledge bit (RAK). If this bit is a logical one, the receiving module must acknowledge the transfer with an in-frame acknowledgement in the ACK field. If it is set to logical zero, then the ACK field must contain an acknowledge absent sequence.
- Third we have the Read/Write bit (R/W). This bit indicates the direction of the data in a frame.
  - If set to zero it is a "write" message, i.e. data transmitted by one module to be received by another module.
  - If it is set to one it implies a "read" message, i.e. a request that another module should transmit data to be received by the one that requested the data (reply request message).
- Last in the command field is the Remote Transmission Request bit (RTR). This bit is a logical zero if the frame contains data and a logical one if the frame does not contain data. In order to conform with the standard a received frame included the comination R/W. RTR = 01 is ignored without any IT generation.

All the bits in the command field are automatically handled by the TSS461C, so the user need not to be concerned for the encoding and decoding of these. The command bits transmitted on the VAN bus are calculated from the current status of the active message.
After the command field comes the data field. This is just a sequence of bytes transmitted MSB first. In the VAN standard the maximum message length is set to 28 bytes, but the TSS461C handles messages up to 30 bytes.
The next field is the FCS field. This field is a 15 bit CRC checksum defined by the following generator polynomial g(x) of order 15:

$$g(x) = x^{15}+x^{11}+x^{10}+x^{9}+x^{8}+x^{7}+x^{4}+x^{3}+x^{2}+1$$

The division is done with a rest initialized to 0x7FFF, and an inversion of the CRC bits is performed before transmission.
However, since the CRC is calculated automatically from the identifier, command and data fields by the TSS461C, it need not concern the user of the circuit. When the frame check sequence has been transmitted, the transmitting module must transmit an End Of Data (EOD) sequence, followed by the ACKnowledge field (ACK) and the End Of Frame sequence (EOF) to terminate the transfer.

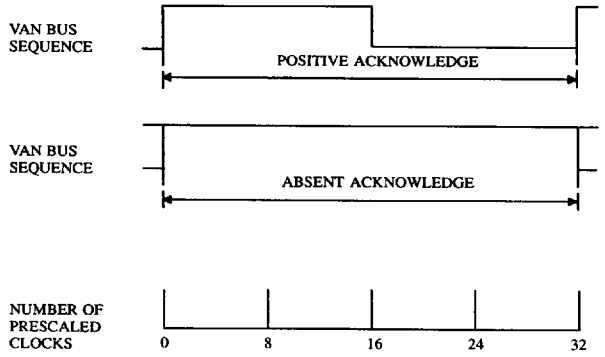


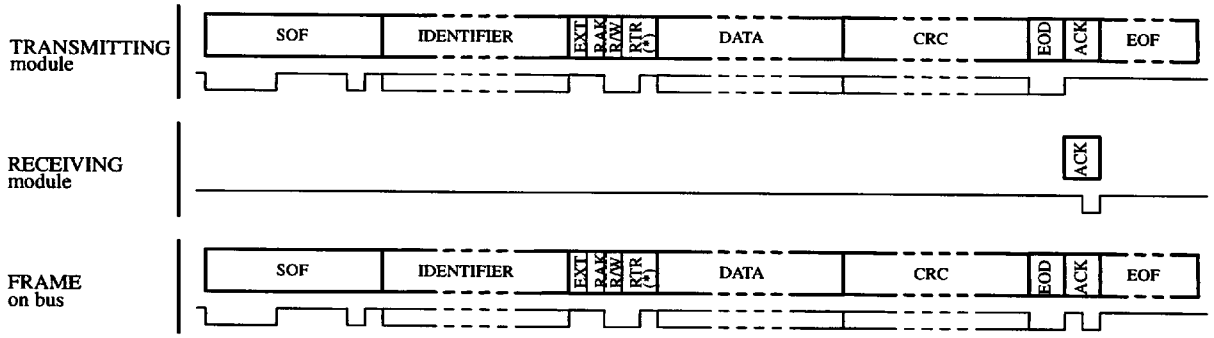
Figure 12. Acknowledge Sequences.

7.3. Frame Examples

The frames transmitted on the VAN bus are generated by several modules, each supplying different parts of the message. Figure 13. through Figure 16. show the four frame types specified in the VAN standard, and what module is generating the different fields.

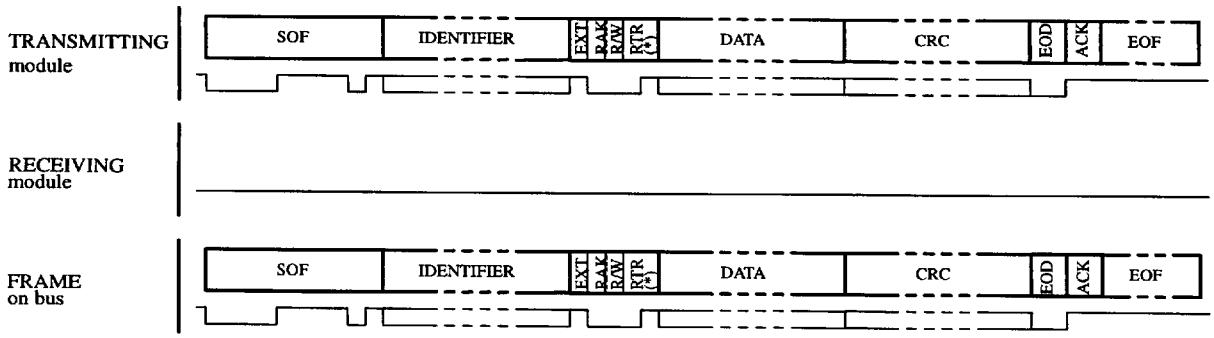
- The most straightforward frame is the normal data frame in Figure 13. Like all other frames it is initiated with a SOF sequence. This sequence is generated by a bus master (not shown in figure).
  During this frame there is basically only one module transmitting with the only exception beeing the acknowledgement, generated by the receiving module if requested in the RAK bit.
- The reply request frame with immediate reply in Figure 14. is the only frame in which a slave module can transmit data by filling it into the appropriate field.
  The only difference for the frame on the bus is that the R/W bit has changed state compared to the normal frame.
  This is a higly interactive frame where a bus master generates the SOF and the initiator generates the identifier, the three first bits of the command, and the acknowledge. The RTR bit, the data field, the frame check, the EOD and the EOF are all generated by the replying module.
- The reply request frame with deferred reply in Figure 15. is basically the same frame as the reply request frame with immediate reply, but since the requested module does not generate the RTR bit the requesting module will continue with the frame check, the EOD and the EOF.
  During this frame the requested module will only generate the acknowledge, and only if this was requested by the initiator through the RAK bit.
- Finally the deferred reply frame in Figure 16. which is sent when a module has prepared a reply for a reply request that has been received earlier.
  This frame very closely mimicks the normal data frame with the only exception beeing the R/W bit that has changed state.

With acknowlegment



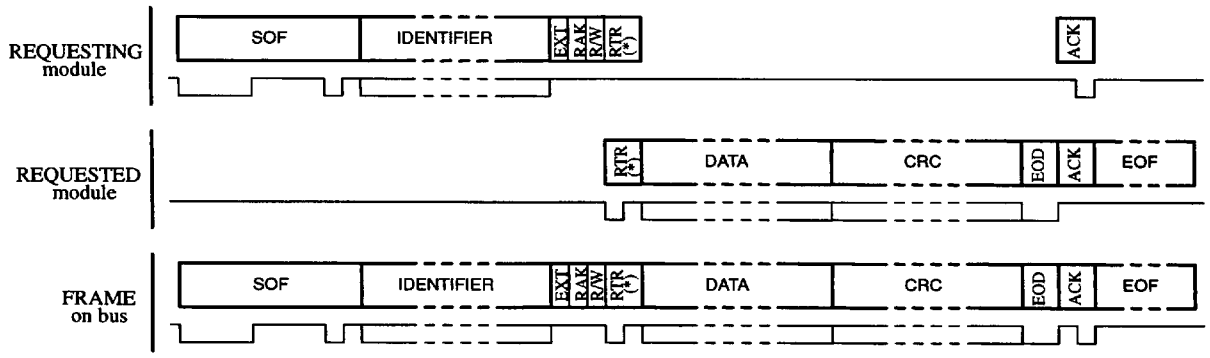
EXT : Recessive from Transmitter
RAK : Recessive for acknowledge from Transmitter
R/W : Dominant from Transmitter
RTR : Dominant from Transmitter (*) Manchester bit
ACK : Positive from Receiver because RAK is Recessive

Without acknowlegment



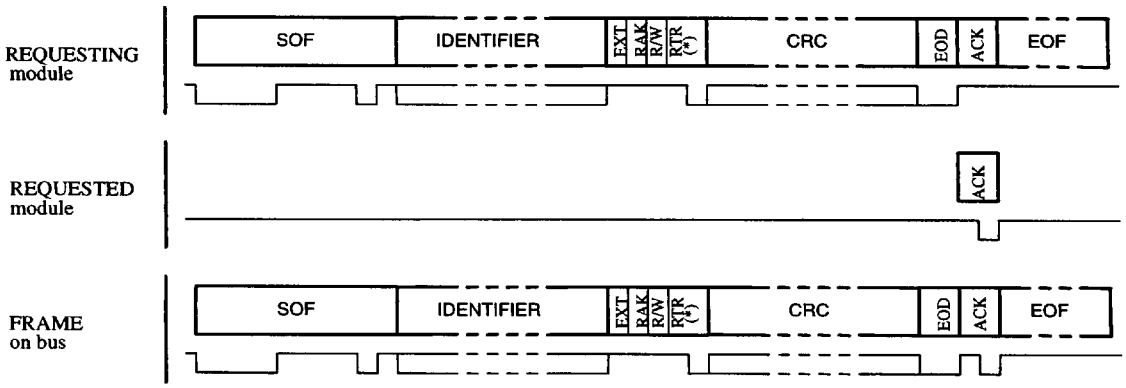
EXT : Recessive from Transmitter
RAK : Dominant for no acknowledge from Transmitter
R/W : Dominant from Transmitter
RTR : Dominant from Transmitter (*) Manchester bit
ACK : Absent from Transmitter and from Receiver because RAK is Dominant

Figure 13. Normal Data Frame



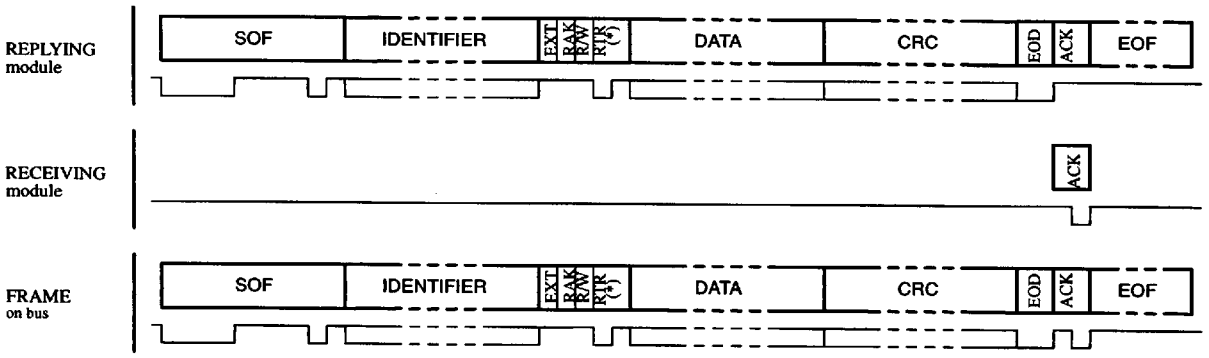
EXT : Recessive from Requestor
RAK : Recessive for acknowledge from Requestor
R/W : Recessive from Requestor
RTR : Recessive from Requestor and Dominant from Requestee(*) Manchester bit
ACK : Absent from Requestee and Positive from Requestor because RAK is Recessive

Figure 14. Reply Request Frame with Immediate Reply



EXT : Recessive from Requestor
RAK : Recessive for acknowledge from Requestor
R/W : Recessive from Requestor
RTR : Recessive from Requestor – (*) Manchester bit
ACK : Absent from Requestor and Positive from Requestee because RAK is Recessive

Figure 15. Reply Request Frame with Deffered Reply



EXT : Recessive from Replyer
RAK : Recessive for acknowledge from Replyer
R/W : Recessive from Replyer
RTR : Dominant from Replyer – (*) Manchester bit
ACK : Absent from Replyer and Positive from Receiver because RAK is Recessive

Figure 16. Deffered Reply Frame

8. Diagnosis System

The purpose of the diagnosis system is to detect any short or open circuits on either the DATA or $\overline{DATA}$ lines and to permit, if it is possible, to carry the communications on the non-defective line.

The diagnosis system is based on the assumption that three separate line receivers are connected to the VAN bus (c.f. Figure 3.):

- One of the line receivers is connected in differential mode, sensing both DATA and $\overline{DATA}$ signals, and is connected to the RxD0 input.
- The other two line receivers are operating in single wire mode and are sensing only one of the two VAN bus signals:
  - the line receiver sensing DATA is connected to RxD1,
  - the line receiver sensing $\overline{DATA}$ is connected to RxD2.

The diagnosis system analyses and compares the data sent over both VAN lines. So, the diagnosis system executes a digital filtering and transition analyses. In order to perform its investigation, three internal signals are generated, RI (*Return to Idle*), SDC (*Synchronous Diagnosis Clock*) and TIP (*Transmission In Progress*).

One of four operating modes can be chosen to manage the results of the diagnosis system.

8.1. Diagnosis States

If the diagnosis system finds a failure on either of the VAN bus signals, it changes from nominal to degraded mode, and connects the line receiver not coupled to the failing signal to the reception logic.
When the diagnosis system finds that the failing signal is working again, it returns to nominal mode and re-connects the differential line receiver to the reception logic.
A major error occurs when both the VAN bus signals are failed.

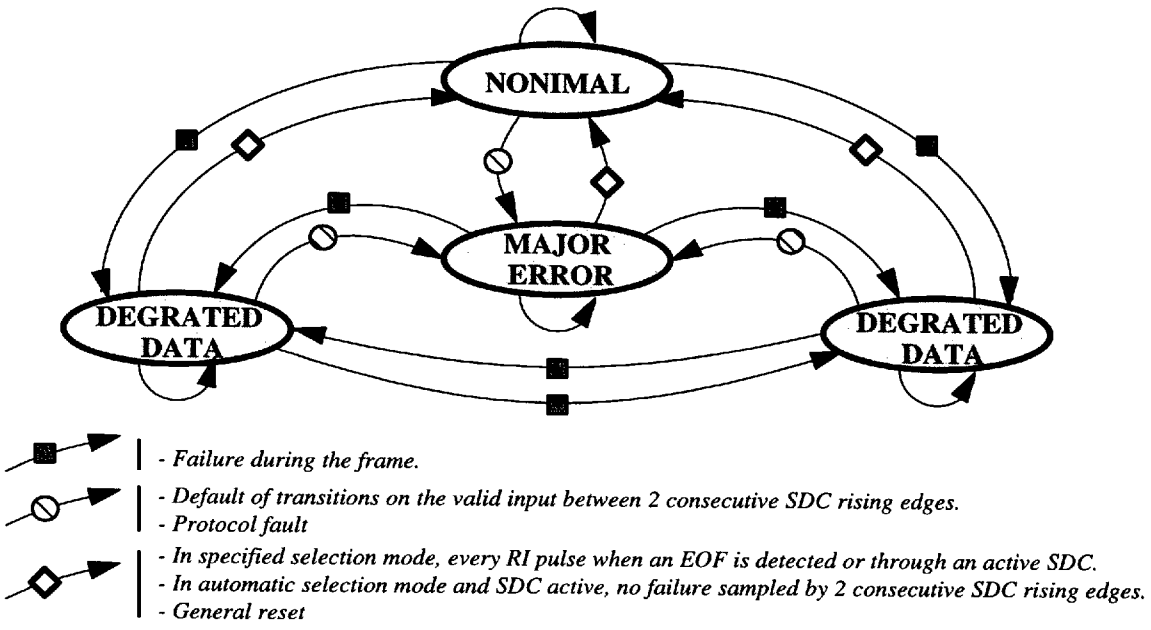


Figure 17. Diagnosis States

Status bits give permanent information on the diagnosis performed, whatever the programmed operating mode. This is encoded over three bits: Sa, Sb and Sc.

– Sa and Sb bits indicate the four possible states of the VAN bus

Table 1. Status bits: Sa & Sb

Sa	Sb		Communication
0	0	Mode	nominal
		Fault	no fault on VAN bus
		Status	differential communication on DATA and $\overline{DATA}$
0	1	Mode	degraded on DATA
		Fault	fault on $\overline{DATA}$
		Status	communication on DATA
1	0	Mode	degraded on $\overline{DATA}$
		Fault	fault on DATA
		Status	communication on $\overline{DATA}$
1	1	Mode	major error
		Fault	fault on DATA and $\overline{DATA}$
		Status	no communication on DATA and $\overline{DATA}$ (attempt to communicate alternatively on DATA then $\overline{DATA}$ every SDC period)

– When set to 1, Sc bit indicates a difference between the three line receiver comparators. This bit is memorized and will be reseted at the end of the next frame (if not error occured during this frame) or by a general reset.

8.2. Internal Operations

8.2.1. Digital Filtering

If several spurious pulses occur during one bit, the diagnosis for defective conductor may be corrupted. To avoid such errors, digital filters are implemented.
Filtering operation is based on sampling of the comparator output signals. A transition is taken into account only if it is observed over five samples ($1/16^{th}$ of timeslot).

8.2.2. Transition Analyses

These analyses are continuously done on the effective edges on comparators after digital filtering.

- **Asynchronous diagnosis**
  The asynchronous diagnosis is done by comparing the number of edges on DATA and $\overline{DATA}$.
  If four edges are detected on one input and no edges on the other during the same peiod, the second input is considered faulty and the diagnosis mode will change to one of the degraded modes.

- **Synchronous diagnosis**
  The synchronous diagnosis counts the number of edges on the data input connected to the reception logic during one SDC period.
  If there are less than four edges during one SDC period, the diagnosis mode will change to the major error mode.
- **Transmission diagnosis**
  The transmission compares RxD1 and RxD2 inputs (through the input comparators and the filters) with the data

transmitted on TxD output.
At a time when the transmission logic generates a dominant - recessive transition, the inputs can give different values. Taking into account the filtering delay, the bus line seen as dominant is assumed to be correct, the other one, recessive, is considered faulty. The diagnosis mode is changed to reflect that.

- **Protocol fault**
  The protocol fault is detected by counting the number of consecutive dominant timeslots.
  If eight consecutive timeslots are dominant, the diagnosis mode will change to the major error mode.

8.3. Generation of Internal Signals

8.3.1. RI Signal (Return to Idle)

This signal is used to return to nominal mode in the three specified selection modes (see sections 8.1. and 8.4.). The RI signal is disabled in automatic selection mode.

The RI signal is a pulse generated when an EOF is detected. So, at the end of each frame, the user, regarding the diagnosis status bit Sa, Sb & Sc, can make its own choice.

8.3.2. SDC Signal (Synchronous Diagnosis Clock)

This time base is used by diagnosis system in automatic selection mode (see section 8.4.) when no event is recorded on the bus.

The SDC is generated either by a special SDC divider connected to the timeslot clock, either manually. The SDC clock period must be long compared to the timeslot duration.

A typical SDC period should be greater than the maximum frame length appearing on the VAN network.

8.3.3. TIP Signal (Transmission In Progress)

This signal must be enabled to allow the transmission diagnosis (see section 8.2.2.).

The TIP turns on synchronously with the beginning of the transmission:
- for asynchronous bus access, the beginning of SOF,
- for synchronous bus access, the beginning of the identifier field,
- for a requestee of in frame reply, the RTR bit of the command field.

The TIP turns off synchronously with the end of the transmission:
- after EOF
- after a losing of arbitration or a code violation detection
- for a requestor of in frame reply, when the arbitration is lost on RTR the bit.

This signal is not generated when the transmission logic only sends an ACK.

8.4. Programming Modes

Four programming modes determine the way to use the three different inputs and the diagnosis system.
- 3 specified selection modes
- 1 automatic selection mode

Table 2. Programming modes

Ma	Mb	Operating mode
0	0	Differential communication
0	1	Degraded communication on RxD2 ($\overline{\text{DATA}}$)
1	0	Degraded communication on RxD1 (DATA)
1	1	Automatic selection according the diagnosis status

9. Registers

The TSS461C memory map consists of three different areas, the Control & Status registers, the Channel registers and the Message data (or Mailbox).

9.1. Mapping

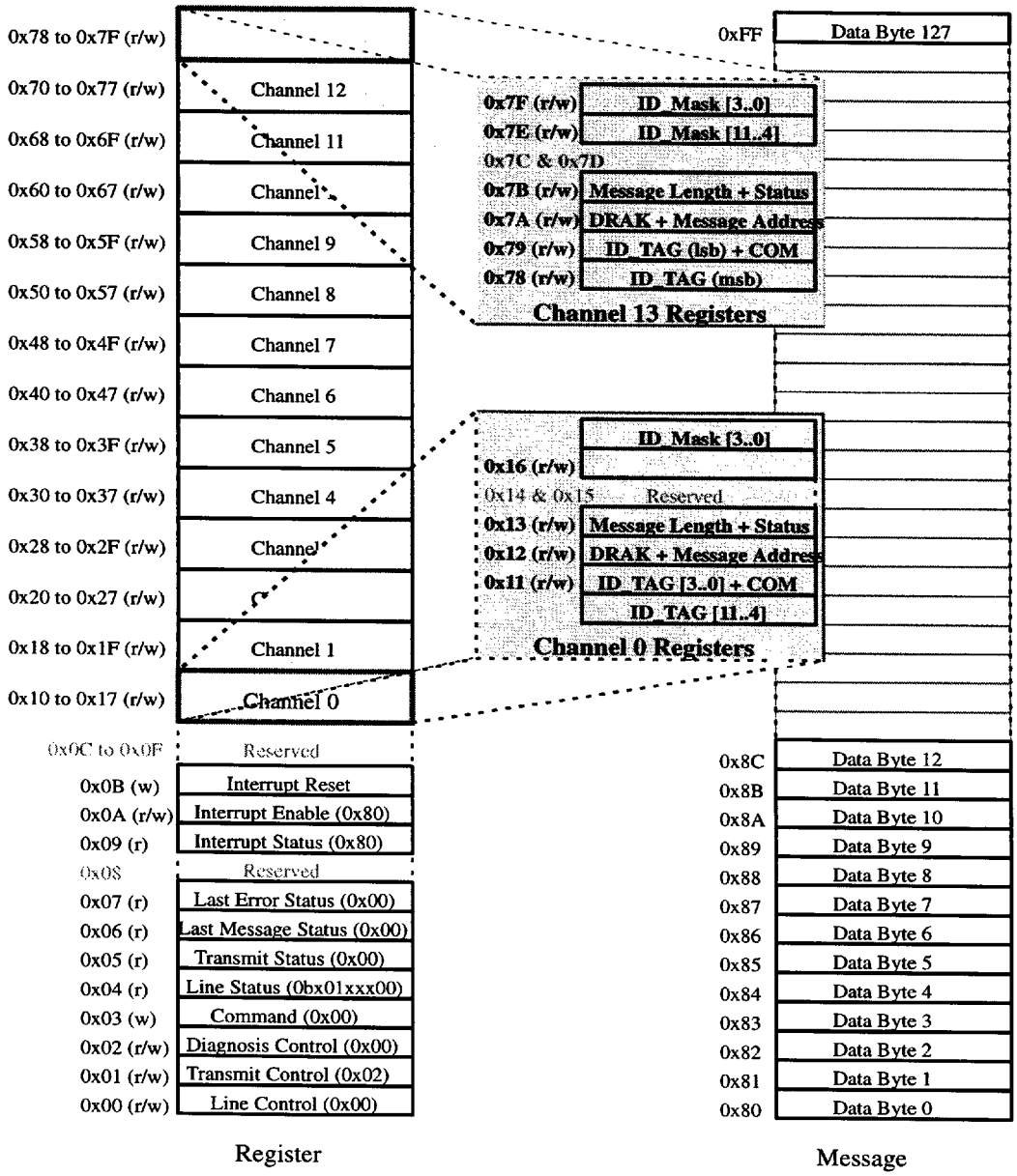


Figure 16. Memory Map.

Note: All the non specified addresses between 0x00 and 0x7F are considered as absent.
(r) means read only register.
(w) means write only register.
(r/w) means read/write register.
Value after RESET is found after register name. If no value is given, the register is not initialized at RESET.

9.2. Control and Status Registers

9.2.1. Line Control Register (0x00) :

7	6	5	4	3	2	1	0
CD3	CD2	CD1	CD0	PC	0	IVTX	IVRX

- Read/write register.
- Default value after reset : 0x00
- reserved : Bit 2, this bit must not be set by the user ; a 0 must always be written to this bit.

CD[3:0] : Clock Divider. They control the VAN Bus rate through a Baud Rate generator according to the formula below :

$$f(TSCLK) = \frac{f(XTAL1)}{n \times 16}$$

PC : Pulsed Code

One : The TSS461C will transmit and receive data using the pulsed coding mode (i.e optical or radio link mode). The use of this mode implies communication via the RXD0 input and the non-functionality of the diagnosis system.
Zero : (default at reset) The TSS461C will transmit and receive data using the Enhanced Manchester code. (RXD0, RXD1, RXD2 used).

IVTX : Invert TXD output
IVRX : Invert RXD inputs

The user can invert the logical levels used on either the TXD output or the RXD inputs in order to adept to different line drivers and receivers.

One : A one on either of these bits will invert the respective signals.
Zero : (default at reset) The TSS461C will set TXD to recessive state in Idle mode and consider the bus free (recessive states on RXD inputs).

9.2.2. Transmit Control Register (0x01) :

7	6	5	4	3	2	1	0
MR3	MR2	MR1	MR0	VER2	VER1	VER0	MT

- Read/Write register.
- Default value after reset : 0x02

MR[3:0] : Maximum Retries.

These bits allow the user to control the amount of retries the circuit will perform if any errors occurred during transmission.

Table 4. Retries

MR [3:0]	Max. Nb of retry	Max. Nb of transmis
0000	0	1
0001	1	2
0010	2	3
0011	3	4
0100	4	5
0101	5	6
0110	6	7
0111	7	8
1000	8	9
1001	9	10
1010	10	11
1011	11	12
1100	12	13
1101	13	14
1110	14	15
1111	15	16

Note : Bus contention is not regarded as an error and that an infinite number of transmission attempts will be performed if bus contention occurs continuously.

VER[2:0]: DLC Version after reset.

– 000 : TSS461A & B

– 001 : TSS461C

These bits must not be set by user; 001 must always be written to these bits.

MT: Module type

The three different module types are supported (see section 7.2.):

One: The TSS461C is at once an autonomous module (Rank 0), an synchronous access module (Rank 1) or a slave module (Rank 16).

Zero: The TSS461C is at once an synchronous access module (Rank 1) or a slave module (Rank 16).

9.2.3. Diagnosis Control Register (0x02) :

7	6	5	4	3	2	1	0
SDC3	SDC2	SDC1	SDC0	Ma	Mb	ETIP	ESDC

- Read/Write register
- Default value after reset : 0x00.

The diagnosis is discussed in greater detail in section 8. of this chapter.

- In its four high order bits the user can program the SDC rate SDC [3:0],
- In its two medium order bits the diagnosis system mode is controlled : M1, M0.
- In the two low order bits, the user controls if the SDC and TIP are to be generated automatically ETIP, ESDC.

SDC [3:0] : SDC divider The input clock is the times lot clock.

Table 5. System Diagnosis Clock Divider

SDC DIVIDER SDC [3:0]	Divide by
0000	64
0001	128
0010	256
0011	512
0100	1024
0101	2048
0110	4096
0111	8192
1000	16384
1001	32768
1010	65536
1011	131072
1100	262144
1101	524288
1110	1048576
1111	2097152

Ma, Mb : Operating mode command bits

Table 6. Diagnosis System Command Bits

Ma	Mb	
0	0	Forces the Communication on RXD0 (differential)
0	1	Forces the Communication on RXD2 ($\overline{DATA}$)
1	0	Forces the Communication on RXD1 (DATA)
1	1	Automatic selection

ETIP : Enable Transmission In Progress

One : Enable TIP generation
Zero : Disable TIP generation.

–The Transmission In Progress (TIP), tells the diagnosis system to enable transmission diagnosis.

ESDC : Enable System Diagnosis Clock

One : Enable SDC divider.
Zero : Disable SDC divider.

–The Synchronous Diagnosis Clock (SDC), controls the cycle time of the synchronous diagnosis.

9.2.4. Command Register (0x03) :

7	6	5	4	3	2	1	0
GRES	SLEEP	IDLE	ACTI	REAR	0	0	MSDC

- Write only register.
- Reserved : Bit 1, 2 these bit must not be set by the user ; a zero must always be written to these bit.
- If the circuit is operating at low birates there might be a considerable delay between the writing of this register and the performing of the actual command (worst case 6 timeslots). The user is therefore recommanded to verify, by reading the Line Status Register (0x04) that the commands have been performed.

GRES : General Reset.

The Reset circuit command bit performs, if set, exactly as if the external reset pin was asserted. This command bit has its own auto-reset circuitry.

One : Reset active
Zero : Reset inactive

SLEEP : Sleep command.

If the user sets the Sleep bit, the circuit will enter sleep mode. When the circuit is in sleep mode, all non-user registers are setup to minimize power consumption and the oscillator is stopped. To exit from this mode the user must set either the idle or activate commands.

One : Sleep active
Zero : Sleep inactive

IDLE : Idle command.

If the user sets the Idle bit, the circuit will enter idle mode. In idle mode the oscillator will operate, but the TSS461C will not transmit or receive anything on the bus, and the TXD output will be in three state

One : Idle active
Zero : Idle inactive

ACTI : Activate command.

The Activate command will put the circuit in the active mode, i.e it will transmit and receive normally on the bus. When the circuit is in activate mode the TXD three-state output is enabled.

One : Activate active
Zero : Activate inactive

REAR : Re-Arbitrate command.

This command will, after the current attempt, reset the retry counter and re-arbitrate the messages to be transmitted in order to find the highest priority message to transmit.

One : Re-arbitrate active
Zero : Re-arbitrate inactive

MSDC : Manual System Diagnosis Clock.

Rather than using the SDC divider described in section 9.2.3., the user can use the manual SDC command to generate a SDC pulse for the diagnosis system.

This MSDC pulse should be high at least two timeslot clock.

9.2.5. Line Status Register (0x04) :

7	6	5	4	3	2	1	0
X	SPG	IDG	Sc	Sb	Sa	TXG	RXG

- Read only register.
- Default value after reset : 0bx01xxx00.
- This register reports the operation mode of the TSS461C in the Sleep an Idle bits (Command Register located at address 0x03) as well as the diagnosis system status bits S2 to S0 discussed in section 8.

SPG : Sleeping

IDG : Idling. Default mode at reset

Sa, Sb and Sc : Diagnosis system status bits

- **Sa and Sb**

Table 7. Diagnosis System Status Bits

Sb	Sa	COMMUNICATION INDICATION
0	0	Nominal mode, differential communication
0	1	Degraded over $\overline{DATA}$, fault on DATA
1	0	Degraded over DATA, fault on $\overline{DATA}$
1	1	Major error, fault on DATA and $\overline{DATA}$

- **Sc** : As soon as one of the three inputs (RXD2, RXD1, RXD0) differs from the others in the input comparison analysis performs by the diagnosis system, S2 is set.
  The only ways to reset this status bit are through the RI signal or a general reset.

TXG : Transmitting. If this status bit is active, it indicates that the TSS461C has choosen an identifier to transmit, and it will continue to make transmission attempt for this message until it succeeds or the retry count is exceeded.

RXG : Receiving. The receiving indicates that there is activity on the bus.
NOTE : For safe modification of active channel registers both bits should be inactive (except "abort" command).

9.2.6. Transmission Status Register (0x05) :

7	6	5	4	3	2	1	0
NRT3	NRT2	NRT1	NRT0	IDT3	IDT2	IDT1	IDT0

- Read only register.
- Default value after reset : 0x00.
- The transmission Status register contains the number of retries made up-to-date, according to the Table 4. , and the channel currently in transmission.

NRT [3:0] : Number of retries done in transmission.

IDT [3:0] : Channel number currently in transmission.

9.2.7. Last Message Status Register (0x06) :

7	6	5	4	3	2	1	0
NRTR3	NRTR2	NRTR1	NRTR0	IDTR3	IDTR2	IDTR1	IDTR0

- Read only register.
- Default value after reset : 0x00.
- This register is basically the same as the transmission status register. It contains the last identifier number that was successfully transmitted, received or exceeded its retry count.
  If it was a successful transmission, the number of retries performed can be seen in this register as well.

NRTR [3:0] : Number of retries done successfully in transmission. In case of reception NRTR[3:0] is undefined.

IDTR [3:0] : Channel number that was successfully transmitted, received or exceeded its retry count.

9.2.8. Last Error Status Register (0x07) :

7	6	5	4	3	2	1	0
X	BOC	BOV	X	FCSE	ACKE	CV	FV

- Read only register.
- Default value after reset : 0x00.
- The Last Error Status Register contains the error code for the last transmission or reception attempt. It is updated after each attempt, i.e. several error codes can be reported during one single transmission (with several retries).

BOC : Buffer occupied.

- when one channel configured in "Reply request" mode has its "received" bit set when it attempts to transmit its request.
- BOC with the link capability between two channels sharing the same received buffer, is set when one channel has already set its "received" bit in its "Message length and status Channel register" and a receive is attempt on the other one.

BOV : Buffer overflow.

BOV indicates that the buffer length setup in the Channel Status Register was shorter than the number of bytes received plus 1, and thus, some data was lost.

One : BOV active
Zero : BOV inactive

FCSE : Framing Check Sequence Error.

FCSE indicates a mismatch between the FCS received and the FCS calculated

One : FCSE active
Zero : FCSE inactive

ACKE : Acknowledge Error.

ACKE indicates a physical violation or collision on ACK field of the frame when the TSS461C is producer.

One : ACKE active
Zero : ACKE inactive

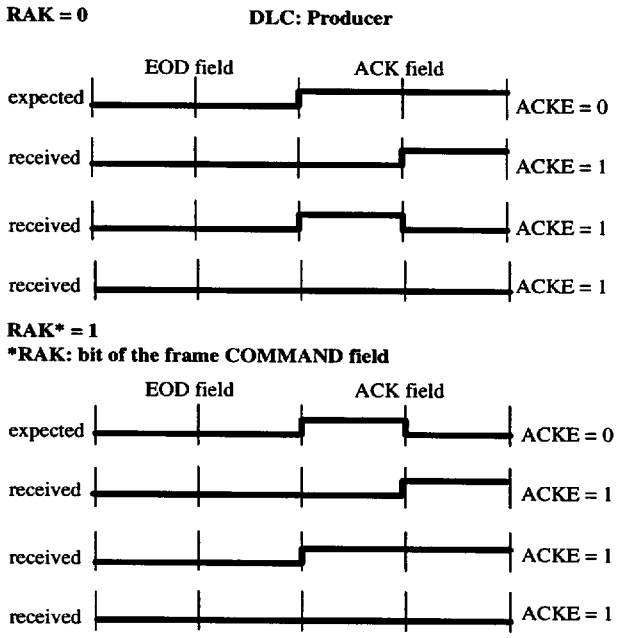


Figure 18. ACKE Status bit

CV : Code Violation.

CV indicates:

- either a Manchester code violation (2 identical TS on Manchester bit), or a physical violation (transmitted bit "dominant", received bit "recessive"), on fields ID, COM, DATA and CRC.
- either a physical violation or collision on field "preamble" and the "recessive" bit of the "Star Sync" field.

One : CV active
Zero : CV inactive

FV : Frame Violation.

FV indicates a physical violation or collision on ACK field of the frame when the TSS461C is consumer.

One : FV active
Zero : FV inactive

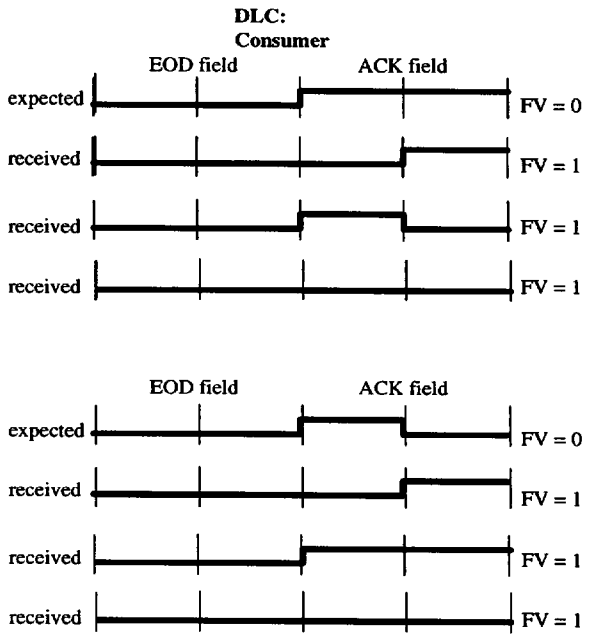


Figure 19. FV Status bit

9.2.9. Interrupt Status Register (0x09) :

7	6	5	4	3	2	1	0
RST	X	X	TE	TOK	RE	ROK	RNOK

- Read only register.
- Default value after reset : 0x80

RST : Reset interrupt.

RE indicates that the circuit has detected a valid reset command via the RESET pin or the reset command bit GRES.

This interrupt cannot be disabled, since its enable bit is set when a reset is detected.

TE : Transmit error interrupt (or exceeded retry).

This flag is set only when the Max number of transmission (1+MR [3:0]) is reached with error of transmission.

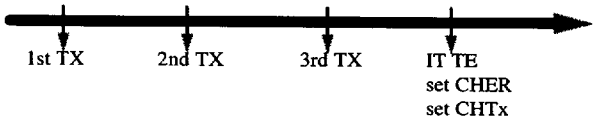


Figure 20. Exceeded retry with MR[3..0] = 3

TOK : Transmit OK interrupt.

RE : Receive error interrupt.

ROK : Receive "with RAK (RAK=1)" OK interrupt.

One : TE, RE, TOK, ROK actives
Zero : TE, RE, TOK, ROK inactives

RNOK : Receive "with no RAK (RAK=0)" OK interrupt.

One : Interrupt actived
Zero : No Interrupt

9.2.10. Interrupt Enable Register (0x0A) :

7	6	5	4	3	2	1	0
1	0	0	TEE	TOKE	REE	ROKE	RNOKE

- Read/write register.
- Default value reset : 0x80

NOTE : On reset the Reset Interrupt Enable bit is set to 1 instead of 0, as is the general rule.

TEE : Transmit Error Enable

TOKE : Transmission OK Enable.

REE : Reception Error Enable.

ROKE : Reception "with RAK" OK enable.

RNOKE : Reception "with no RAK" OK enable.

One : IT enabled.
Zero : IT disabled.

9.2.11. Interrupt Reset Register (0x0B) :

7	6	5	4	3	2	1	0
RSTR	0	0	TER	TOKR	RER	ROKR	RNOKR

- Write only register.
- Reserved bit : 5 and 6. This bit must not be set by user; a zero must always be written to this bit.

RSTR : Reset Interrupt Reset.

TER : Transmit Error Interrupt Reset.

TOKR : Transmit OK Interrupt reset.

RER : Receive Error Interrupt Reset.

ROKR : Receive "with RAK" OK reset.

RNOKR : Receive "with no RAK" OK reset.

One : IT reset.
Zero : IT unchanged.

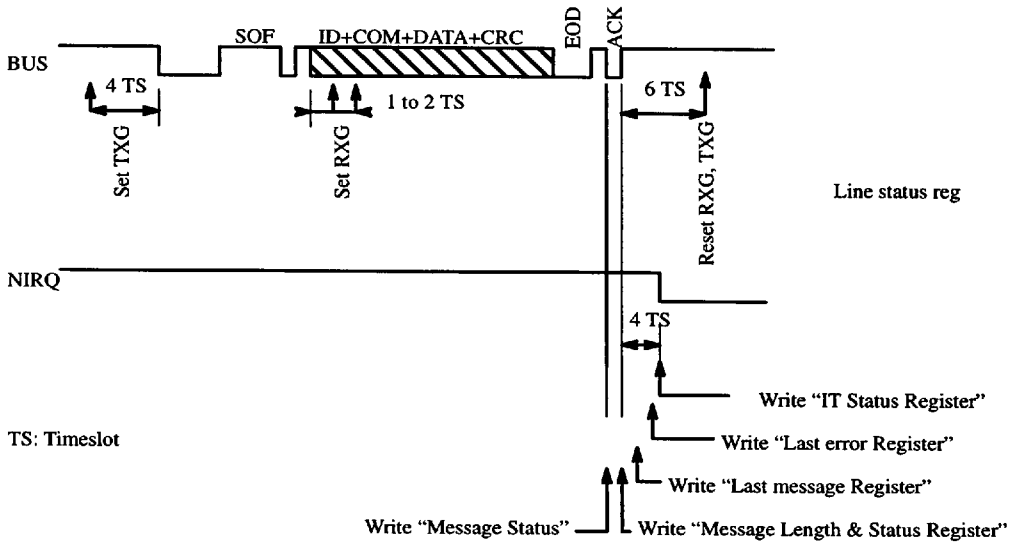


Figure 21. Update of the Status Register

9.3. Channel Registers

There is a total of 14 channel register sets, each occupying 8 bytes for addressing simplicity, integrated into the circuit. Each set contains two 2x8-bit registers for the indentifier tag, indentifier mask and command fields plus two 1x8-bit registers for DMA pointers and message status.
The base_address of each set is: ***(0x10 + (0x08 * channel_number))***.
When the TSS461C is reseted either via the external reset pin or the general reset command, the channel registers are not affected. That is, on power-up of the circuit, all the channel registers start with random values.
Due to this fact, the user should take care to initialize all the channel registers before exiting from idle mode. The easiest way to disable an channel register is to set the received and transmitted bits to 1 in the Message Length & Status Register.

Table 3: Channel Register Sets Map

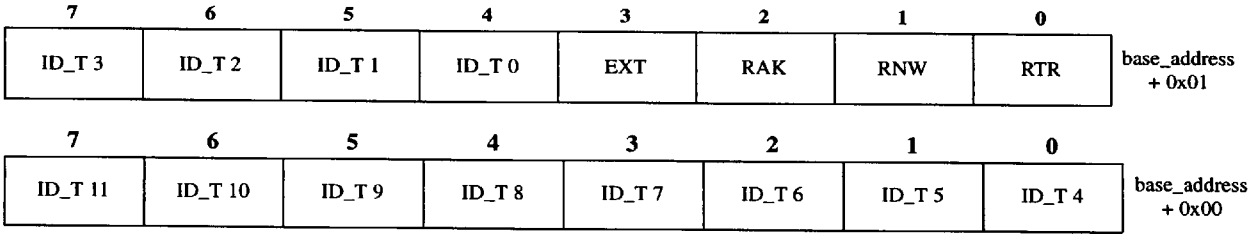
Channel Number	From	To	Channel Number	From	To
6	0x40	0x47	**13**	0x78	0x7F
5	0x38	0x3F	**12**	0x70	0x77
4	0x30	0x37	**11**	0x68	0x6F
3	0x28	0x2F	**10**	0x60	0x67
2	0x20	0x27	**9**	0x58	0x5F
1	0x18	0x1F	**8**	0x50	0x57
0	0x10	0x17	**7**	0x48	0x4F

Table 4: Channel Register Set Structure

Reg. Name	Offset	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ID_MASK	0x07	ID_M [3:0]				X	X	X	X
ID_MASK	0x06	ID_M [11:4]							
(no register)	0x05	X	X	X	X	X	X	X	X
(no register)	0x04	X	X	X	X	X	X	X	X
MESS_L / STA	0x03	M_L [4:0]					CHER	CHTx	CHRx
MESS_PTR	0x02	DRACK	M_P [6:0]						
ID_TAG / CMD	0x01	ID_T [3: 0]				EXT	RAK	RNW	RTR
ID_TAG	0x00	ID_T [11:4]							

9.3.1. Identifier Tag and Command Registers:

The identifier tag and command registers is located at the base_address and base_address + 1. It allows the user to specify the full 12-bit identifier field of the ISO standard and the 4-bit command.



- Read / Write registers.

ID_T [11:0]: Identifier Tag

Upon a reception hit (i.e, a good comparison between the identifier received and an identifier specified, taking the comparison mask into account, as well as a status and command indicating a message to be received), the identifier tag bits value will be rewritten with the identifier bits actually received.

EXT, RAK, RNW & RTR: (See section 11.)

No comparison will be done on the command bits, excepted on EXT bit. The RAK, RNW and RTR bits will be written into the first byte of the Message upon a reception hit.
The RNW and RTR bits, as well as the status bits in the length and status register, must be in a valid position for reception or transmission. If not, the message corresponding to this identifier is considered as inactive or invalid.
The way of knowing if an acknowledge sequence was requested or not is to check the first byte of the Message.

9.3.2. Message Pointer Register:

The message pointer register at address (base_address + 0x02) is 8 bits wide. It indicates where in the Message DATA RAM area the message buffer is located.

7	6	5	4	3	2	1	0	
DRAK	M_P 6	M_P 5	M_P 4	M_P 3	M_P 2	M_P 1	M_P 0	base_address + 0x02

- Read / Write register.

DRAK: Disable RAK (used in 'spy mode')

In reception: whatever is the RAK bit of the incoming valid frame, no ACK answer will be set. If the message was successfully received, an IT is set (ROK or RNOK).

In transmission: no action.
One: disable active, 'spy' mode.
Zero: disable inactive, normal operation.

M_P [6:0]: Message pointer

Since the Message DATA RAM area base address is 0x80, the value in this register is the offset from that address. If the message buffer length value is illegal (i.e. zero), this register is redefined as being a link pointer, thus containing the channel number of the channel that contains the actual message pointer, message length and received status. However, the identifier, mask, error and transmitted status used will be that of the originally matched channel. In any case, if a link is intended, the three high bits of M_P [6:0] should be set to 0.
This allows several channels to use the same actual reception buffer in Message DATA RAM, thus diminishing the memory usage.
Note that only 1 level of link is supported.

9.3.3. Message Length And Status Register:

The message length and status register at address (base_address + 0x03) is also 8 bits wide. It indicates the length of reseved for the message in the Message DATA RAM area.

7	6	5	4	3	2	1	0	
M_L 4	M_L 3	M_L 2	M_L 1	M_L 0	CHER	CHTx	CHRx	base_address + 0x03

- Read / Write register.

M_L [4:0]: Message Length

The 5 high bits of this register allows the user to specify either the length of the message to be transmitted, or the maximum length of a message receivable in the pointed reception buffer.
Note, that the first byte in this register does not contain data, but the length of the message received. This implies that the length value has to be equal to or greater than the maximum length of a message to be received in this buffer (or the length of a message to be transmitted) plus 1, thus allowing a maximum length of 30 bytes and a minimum length of 0 byte.
If the value of this field is "illegal" (i.e 0x00) then this message pointer is defined as being a link
(see Message pointer & register and section 15.).

M_L [4:0] = 0x00	Linked channel
M_L [4:0] = 0x01	Frame with no DATA field (*)
M_L [4:0] = 0x02	Frame with 1 DATA byte
-------	----------
M_L [4:0] = 0x1D	Frame with 28 DATA bytes
M_L [4:0] = 0x1E	Frame with 29 DATA bytes
M_L [4:0] = 0x1F	Frame with 30 DATA bytes

(*) Different of a reply request frame with no in-frame reply (deffered reply)..

CHER: Channel error status and abort command

As status, this bit is set by the TSS461C when error occurs in transmission or on a received frame. The user must reset it.

To abort the transmission defined in the channel, this bit can bit set to 1 by the user (see section 13. and 13.3.)

CHTx: Channel transmitted and $\overline{\text{transmit enable command}}$

CHRx: Channel received and $\overline{\text{receive enable command}}$

The two low order bits of this register contains the message status. Together with the RNW and RTR bits of the command register (base_address + 0x01), they define the message type of this channel (see section 11.). As a general rule (see section 13.3.), the status bits are only set by the TSS461C, so the user must reset them to perform a transmission (CHTx) or/and a reception (CHRx). The received and transmitted bits are only set if the corresponding frame is without errors or if the retry count has been exceeded.

9.3.4. Identifier Mask Registers:

The Identifier Mask registers (base_address + 0x06 and base_address + 0x07) allow bitwise masking of the comparison between the identifier received and the identifier specified.

7	6	5	4	3	2	1	0
ID_M 3	ID_M 2	ID_M 1	ID_M 0	0	0	0	0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID_M 11	ID_M 10	ID_M 9	ID_M 8	ID_M 7	ID_M 6	ID_M 5	ID_M 4

- Read / Write registers.

ID_M [11:0]: Identifier Mask

A value of 1 indicates comparison enabled.
A value of 0 indicates comparison disabled.

10. Mailbox

The mailbox contents all the messages received or to be transmitted. Each messages is link to a channel. The Mailbox RAM area has 128 bytes and is mapped from 0x80 to 0xFF (see section 9.1.).

The message (or message buffer) is composed of:

- 1 byte of message status (only used in receiving),
- n bytes of data. These data are the bytes of the DATA field of the frame with the same organization.

The message is pointed by the Message Pointer Register of the channel, the length of the message is given by the Message Length & Status Register of the channel (sections 9.3.2. and 9.3.3.). This area is a pure RAM, it contents a random value after reset.

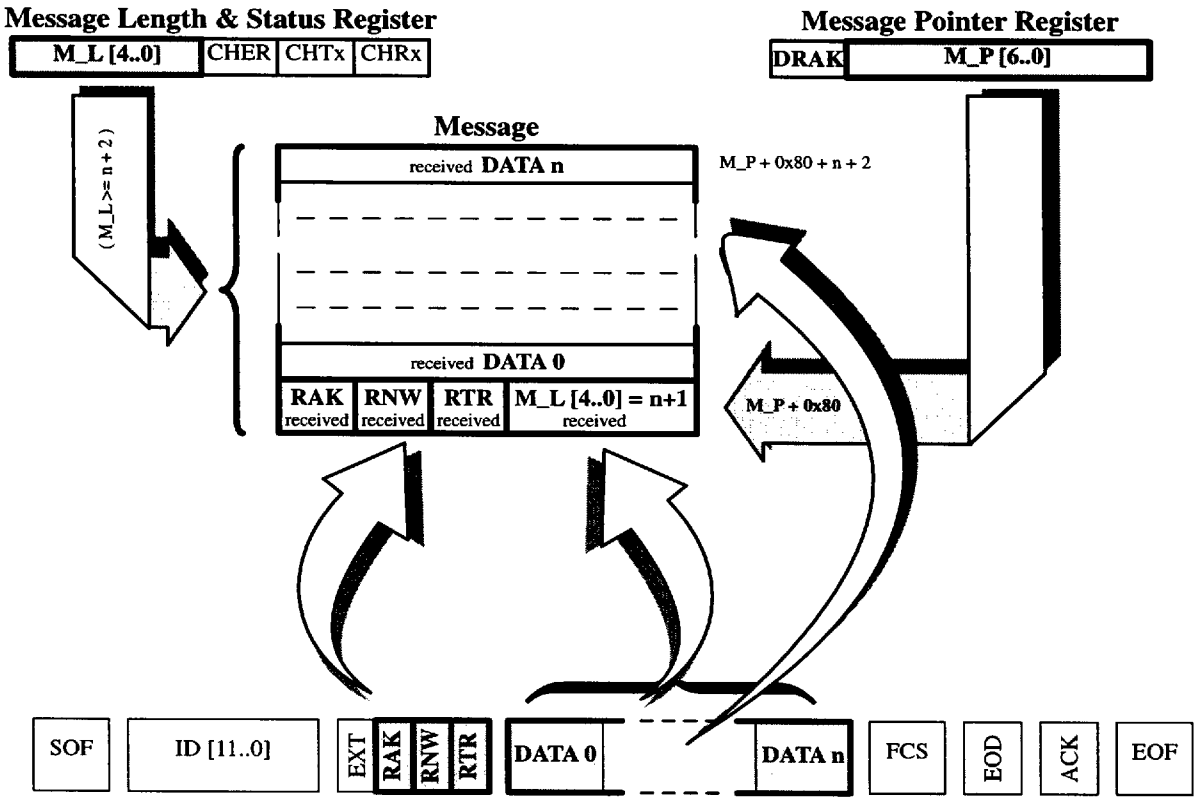


Figure 22. Message buffer structure for reception

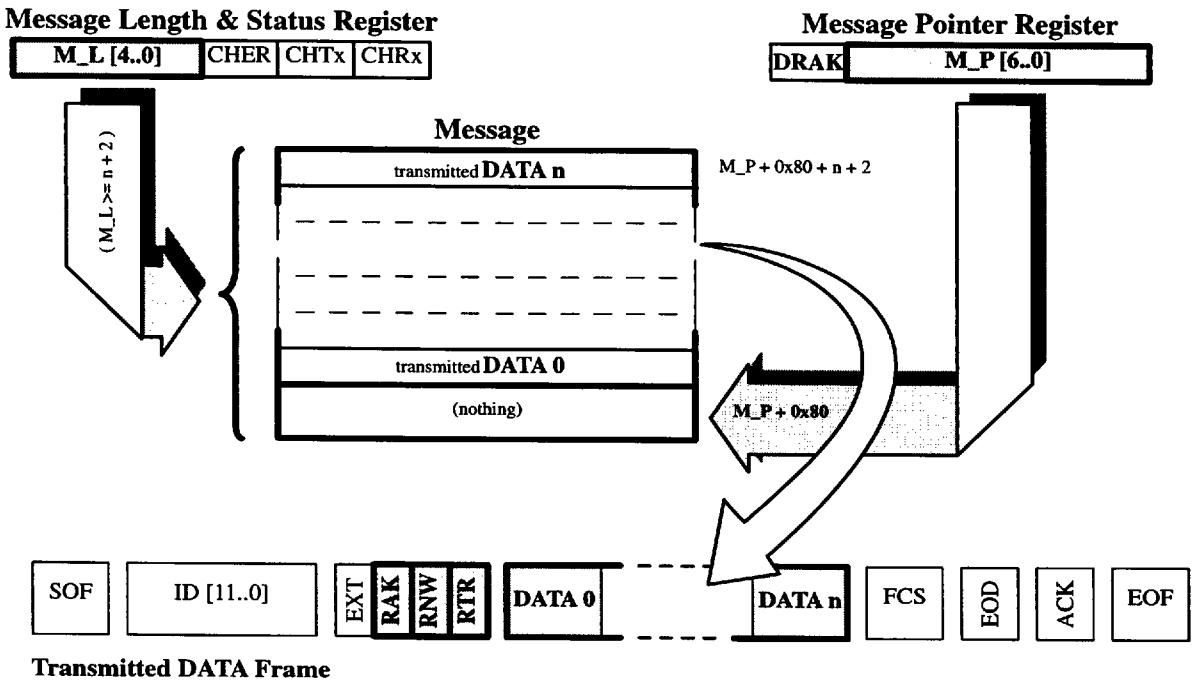


Figure 23. Message buffer structure for transmission

10.1. Message Status (pointed by: *Message Pointer Register*)

7	6	5	4	3	2	1	0
RRAK	RRNW	RRTR	RM_L4	RM_L3	RM_L2	RM_L1	RM_L0

- (no significant value in case of message to be transmitted)

RRAK: Received RAK bit. This bit is the RAK bit coming from the COM field of the received frame.

RRNW: Received RNW bit. This bit is the RNW bit coming from the COM field of the received frame.

RRTR: Received RTR bit. This bit is the RTR bit coming from the COM field of the received frame.

RM_L[4:0]: Message length of the received frame. If the DATA field of the received frame included DATA0 to DATAn, RM_L[4:0] = n+1, even if the reserved length (Message Length & Status Register) is larger.

Frame Type	Node x	Commu-nication	Node A	Message Status on Node A after IT(*)			
Data Frame	I, P		C	RAK	RNW	RTR	length
Immediate Reply	I, C		P	RAK	RNW	RTR	*previous value*
Deferred Reply	I, C		P	RAK	RNW	RTR	*previous value*
Data Frame	C		I, P	*previous values*			
Immediate Reply	P		I, C	RAK	RNW	RTR	length
Deferred Reply	P		I, C	RAK	RNW	RTR	length

P: Producer **I: Initiator** **C: Consumer**

(*) After IT ROK or RNOK. In case of IT RE, the values can be erroneous.

Figure 24. Message Status updating

10.2. Message Data (string pointed by: *Message Pointer Register + 1*)

7	6	5	4	3	2	1	0
DATAn							
-	-	-	-	-	-	-	-
DATA0							

DATA0 is the first received (or transmitted) byte, DATAn is the last one.

Note 1: If the length reserved (in the message length & status register) for an incoming frame is 2 bytes greater or more, the TSS461C will write the 2 bytes of the CRC field in the message string just after DATAn.
Because the VAN frame does not content a message length, the only way for the component to know the length of the DATA field is either the message length register value, either the EOD field detection. When the reserved length is too large, at the moment when it detects the EOD, the TSS461C has already written the 2 bytes of the CRC field, considering these bytes as normal DATA.

Note 2: The Mailbox RAM area is a circular buffer. The next location after 0xFF is 0x80.

11. Messages Types

There are 5 basic message types defined in the TSS461C. Two of them (transmit and receive message types) correspond to the normal frame, and the rest correspond to the different versions of reply frames.

Transmit Message				
	RNW	RTR	Transmitted	Received
Initial setup	0	0	0	Dont care
After transmission	0	0	1	Unchanged

To transmit a normal data frame on the VAN bus, the user must program an identifier as a Transmit Message. The TSS461C will then transmit this message on the bus until it has succeeded or the retry count is exceeded.

Receive Message				
	RNW	RTR	Transmitted	Received
Initial setup	0	1	Dont care	0
After transmission	0	1	Unchanged	1

The opposite of the transmit message type is the Receive Message type. This message type will not generate any frames on the bus. Instead it will listen to the bus until a frame passes that matches its identifier, with the mask taken into account, and then receive the data in that frame.

The data received will be stored in the message buffer and the length of the message received is stored in the first byte of the message buffer.

The actual identifier received is stored in the identifier register itself. This identifier may differ from the identifier specified in the register due to the effect of the mask register.

Normally this should not interfere with the next identifier comparison since the bits that may differ are masked via the mask register.

Reply Request Message				
	RNW	RTR	Transmitted	Received
Initial setup	1	1	0	0
After transmission (Waiting for reply)	1	1	1	0
After reception (of reply)	1	1	1	1

The Reply Request Message type is a demand to transmit on the VAN bus a reply request. When this message type is programmed, three things can happen.

In the first case no other modules on the bus responded with an in-frame reply, and in this case the TSS461C will set the message type to the after transmission state. When this message type is programmed, the TSS461C will listen on the bus for a deferred reply frame matching this identifier, without transmitting the reply request.

The second case is that another module on the bus replies with an in-frame reply. In this case the message type will pass immediatly into the after reception state, without passing the after transmission state.

Reply Request Message without transmission				
	RNW	RTR	Transmitted	Received
Initial setup	1	1	Dont care	0
After reception	1	1	Unchanged	1

In the third case the TSS461C has not yet started to transmit the reply request, when another module either requests a reply, and gets it, or transmits a deferred reply. Warning ! This should be avoided as it may result in an illegal message type (Illegal reply Request).

Immediate Reply Message				
	RNW	RTR	Transmitted	Received
Initial setup	1	0	0	0
After transmission	1	0	1	1

The immediate Reply Message will attempt to transmit an in-frame reply, using the data in the message buffer.

Deferred Reply Message				
	RNW	RTR	Transmitted	Received
Initial setup	1	0	0	1
After reception (of reply request)	1	0	1	1

Above a deferred Reply Message is shown. This message type will immediatly transmit a deferred reply frame.

Reply Request Detection Message				
	RNW	RTR	Transmitted	Received
Initial setup	1	0	1	0
After reception	1	0	1	1

Finally there is the Reply Request Detector Message type. Its purpose is to receive a reply request frame and notify the processor, without transmitting an in-frame reply.

Inactive Message				
	RNW	RTR	Transmitted	Received
Recommanded	Dont care	Dont care	1	1
After transmission	0	0	1	Dont care
After reception	0	1	Dont care	1
Illegal reply request	1	1	0	1

The table above shows all inactive messages types. The last combination will transmit a reply request, but will not receive the reply since its buffer is tagged as occupied.

12. Priority among the different channels

The priority handling on the VAN bus itself is already explained in the Line interface section. The priorities for the messages in the TSS461C is however slightly different.

For instance it's possible that an identifier matches two or more of the identifiers programmed into the registers. In this case, it is the lowest identifier number that has priority. i.e. if both identifier 5 and 10 match the identifier received, it is the identifier 5 that will receive the message.

However, since the identifier 5 will become an inactive message when it has received the frame, the next time the same identifier is seen on the bus, the corresponding data will be received by identifier 10.

The same is valid for messages to be transmitted, i.e. if two or more messages are ready to be transmitted, it is the one with the lowest identifier number that will get priority.

13. Retries, rearbitrate and abort

Retries and rearbitrate commands are located, respectively, in the Transmit Control Register and in the Command Register. An abort command is located in each channel register set, in the Message Length & Status Register (base_address + 0x03). These three commands are available only when the TSS461C is producer.

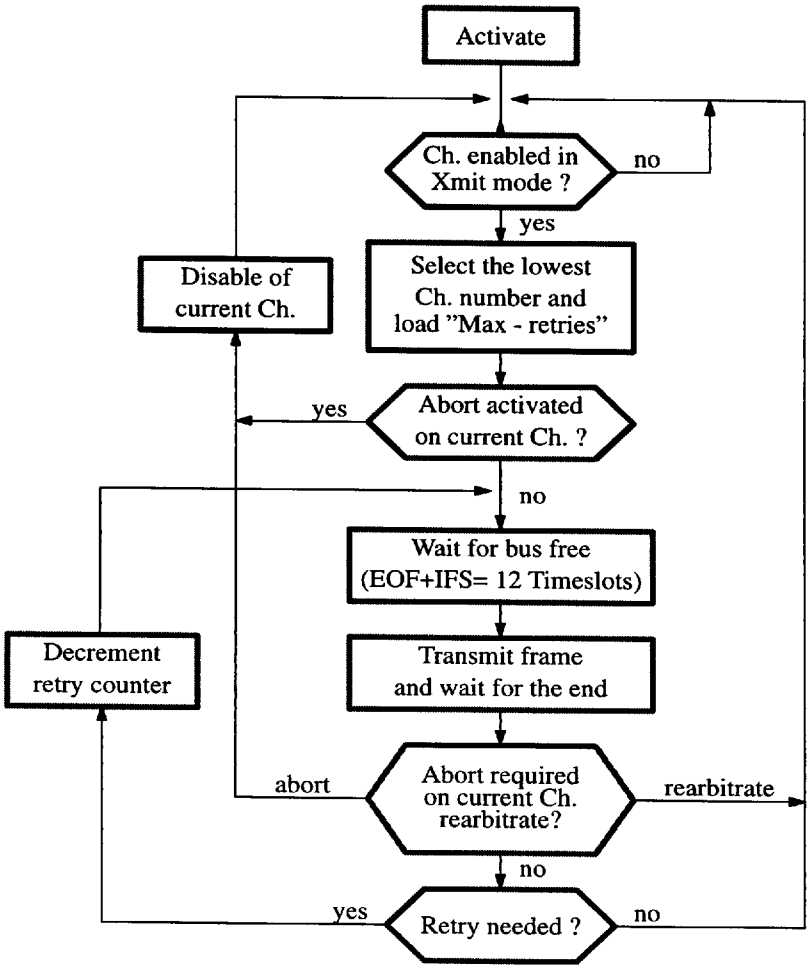


Figure 25. Transmit function

13.1. Retries

The purpose of retries feature is to provide, for the user, the capability of retrying a transmit request in case of failure, when a node tries to reach another node, either on normal DATA frame or on REPLY REQUEST frame.

The maximum of retries is programmable through MR[3:0] of the Transmit Control Register (0x01). When a channel is enable - bit CHTx= 0 of Message Length & Status Register, a 4-bit counter is loaded with MR[3:0]. At each attempt, this counter will be count-down. To 0, an IT TE is set in the Interrupt Status Register (0x09), and the transmission is stopped.

MR[3:0]=1 indicates 1 retry, hence 2 transmission attempts will be performed (see Table 4.). The number of retries performed, as well as the current channel number associated, can be read in the Transmission Status Register (0x05).

The Last Error Status Register (0x07) informs about the trouble encounted:

- Failure cases: - Code viol (CV error bit)
  - Acknoledge error (ACKE error bit)
  - CRC error (FCSE error bit)
- It should be noticed that contention is considered as normal CSMA/CD protocol and, therefore, is not taken into account in failure cases. So, an 'infinite' number of attempts can be performed if bus contention occurs continuously.

There is only one retries counter for all channels. When the user writes the Max_Retries value, all channels start their transmission with this parameter.

13.2. Rearbitrate

The purpose of rearbitrate feature is to postpone a channel already in transmission in order to autorize an higher priority (see section 12.) message to be transmit.

13.2.1. Typical example

- Max_retries = 1 (2 transmissions attempts).
- If Ch 8 is in a the retry loop and the user wants to transmit the Ch 5 without waiting the end of the loop, the user can use the rearbitrate command.
- Then, the TSS461C will wait the end of the current transmission, reload the retries counter and enable the Ch 5 to transmit.
- At the end of this transmission Ch5, either when the attempt is successful or either when the exceeded retry count is reached, the retries counter is reloaded and the transmission is activated for the Ch 8 again.

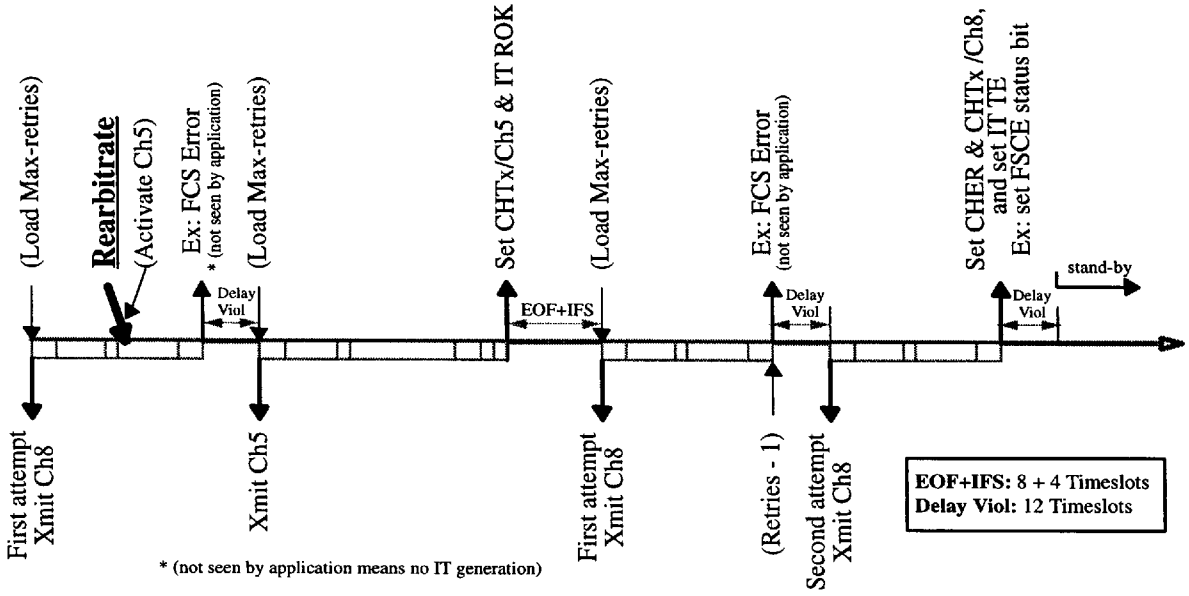


Figure 26. Rearbitrate Example

(same example section 13.2.1.).

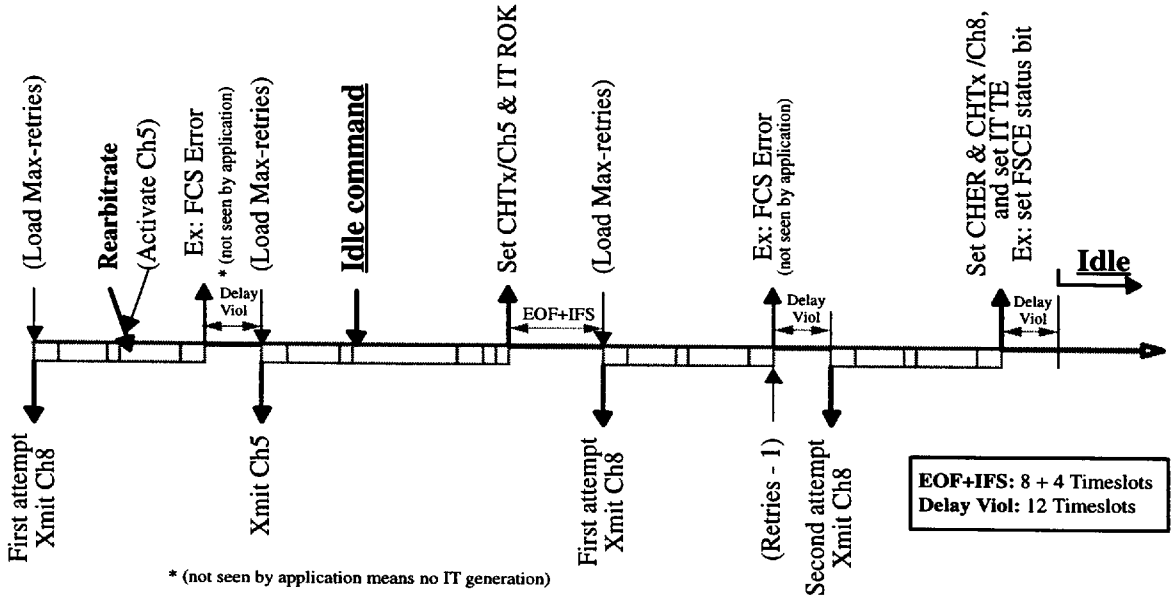


Figure 27. Idle and rearbitrate example

If the user sets the idle bit anywhere (after rearbitrate), the idle mode is entered only at the end of all the transmit attempts (for more information about idle command, see section 14.).

13.2.2. Disable channel after rearbitrate

(same example section 13.2.1.).

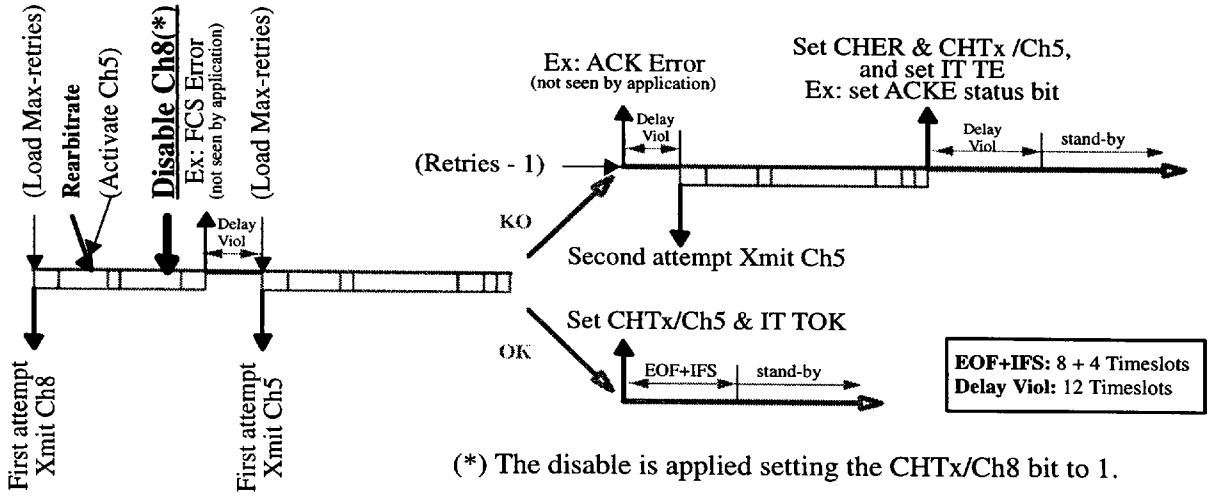


Figure 28. Disable channel after rearbitrate example

In this case, the TSS461C completes the current attempt (Ch8) and let the transmission go on the new channel (Ch5 if validated), otherwise it stops all attempts on the current channel.

13.3. Abort

An abort command is dedicated to channels already enabled in transmission or in in-frame response. For example, this command can be used to break the retry procedure on one channel.

Abort channel is done by setting the Error bit (CHER) in the Message Length & Status Register (base_address + 0x02). This command is taken into account if the channel aborted is not transmitted. When this abort command is really done, the TSS461C set to 1 the Transmitted bit (CHTx) of the Message Length & Status Register.

The abort mechanism is integrated into the transmit function. This mainly means, abort, priority and retries live together in the transmit function.

Example: Ch0, Ch4, Ch6 & Ch13 set in Xmit ACK mode, Max-retry=2 (3 attempts).

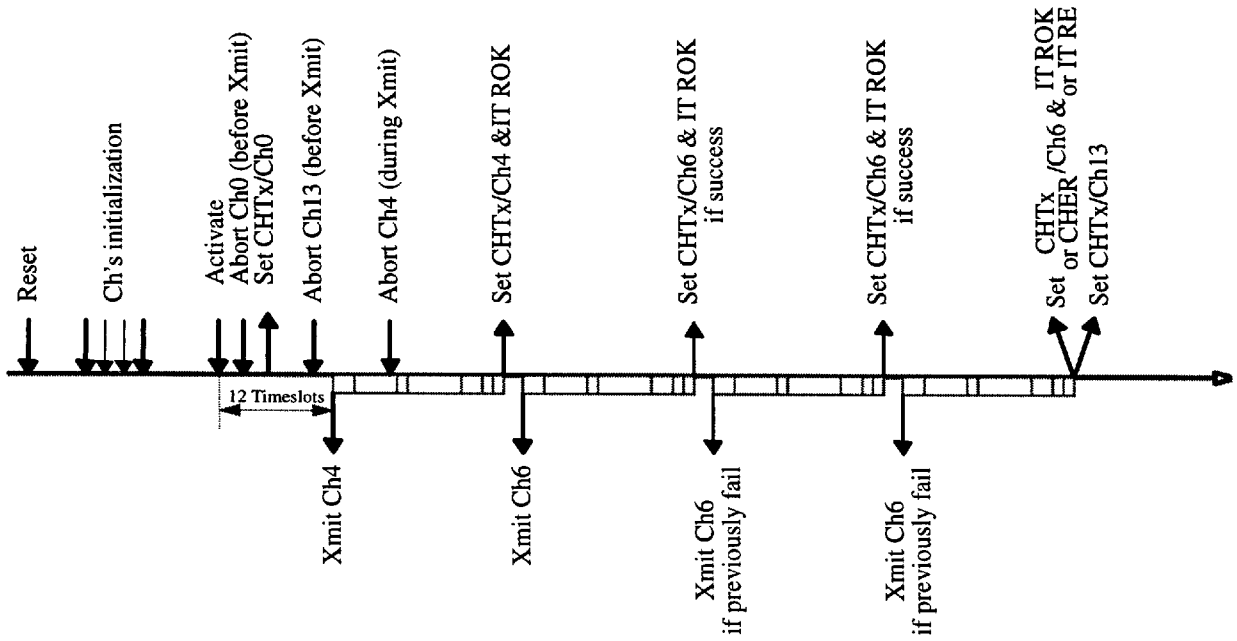


Figure 29. Abort example

14. Activate, idle and sleep modes

Sleep, idle and activate commands are located in the Command Register (0x03). These three commands are general commands for the TSS461C.

14.1. Idle and activate commands

After reset, the TSS461C starts in idle mode. In this mode, the oscillator operates (CKOUT pin active) but the circuit cannot transmit or receive anything on the VAN bus. The TxD output (pin 18) is in three state mode, a pull-up resistor must be be provided externally or by the line driver to avoid floating state on the VAN bus.

To activate the TSS461C, the user must set the activate bit (ACTI) and reset the idle bit (IDLE).

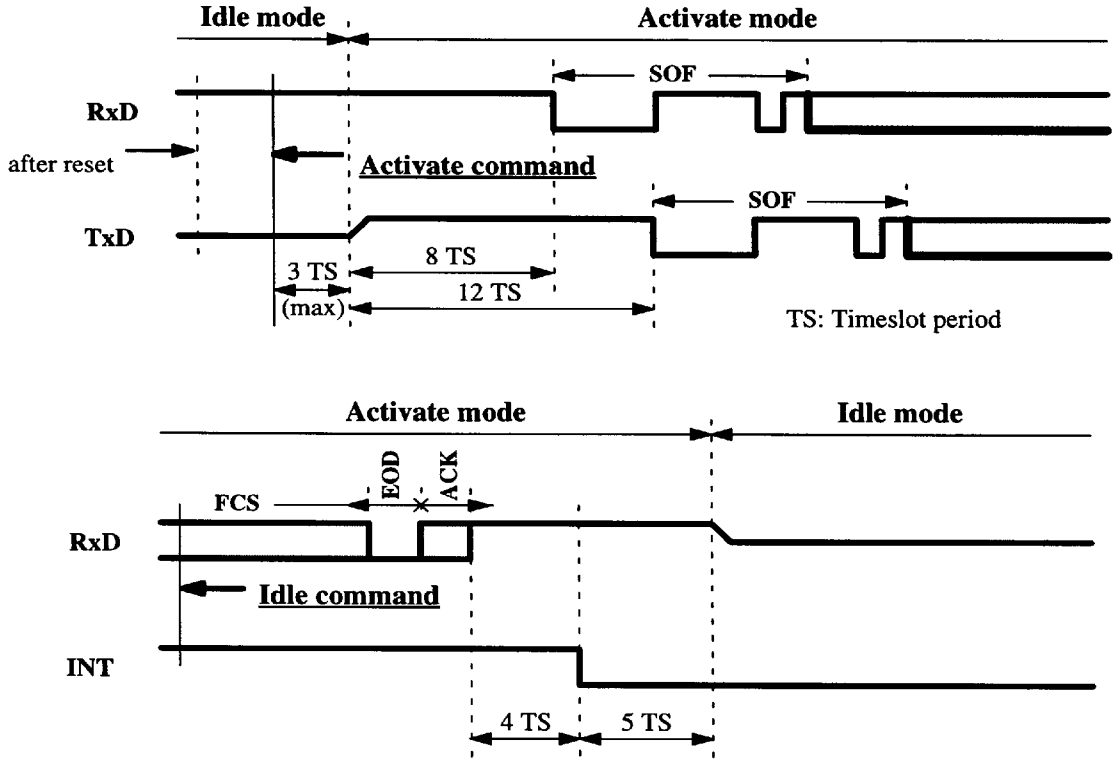


Figure 30. Idle and activate timings

In both cases, the idle state can be verified reading the Line Status register (0x04).

14.2. Sleep command

If the user sets the sleep bit (SLEEP), the TSS461C enters in sleep mode, whatever are the values of activate and idle bits. It means that, all non-user registers are set-up to reduce the power consumption and the internal oscillator is immediately stopped. However, all user registers (accessible by mP bus) are always available by the user
To exit from this mode, the user must set either the idle bit or either the activate bit.

In an application (i.e. typical application Figure 3.) using the CKOUT feature (pin 12), if the TSS461C is put in sleep mode, the clock provided to the microcontroller is stopped. So, the system does not run and the only way to awake this application is an external reset.

15. Linked channels

The linkage feature allows to channels to share the same Message area, the message pointer and the message length assumes this property:

- Zero value as message length (**M_L [4:0]** - *base_address + 0x03*) declares the channel linked to another channel.
- The number of this other channel is defined in the message pointer field (**M_P [6:0]** - *base_address + 0x02*).
- The pointer and the length values for the Message area are defined only once time, in the register set of this other Channel.

Only one level of linkage can be created. This means, (see Figure 31.) a Channel k can be linked to the Channel i but not to Channel j, already defined as linked to Channel i.

All the others can be different between the two channels, for example the ID_Tag.

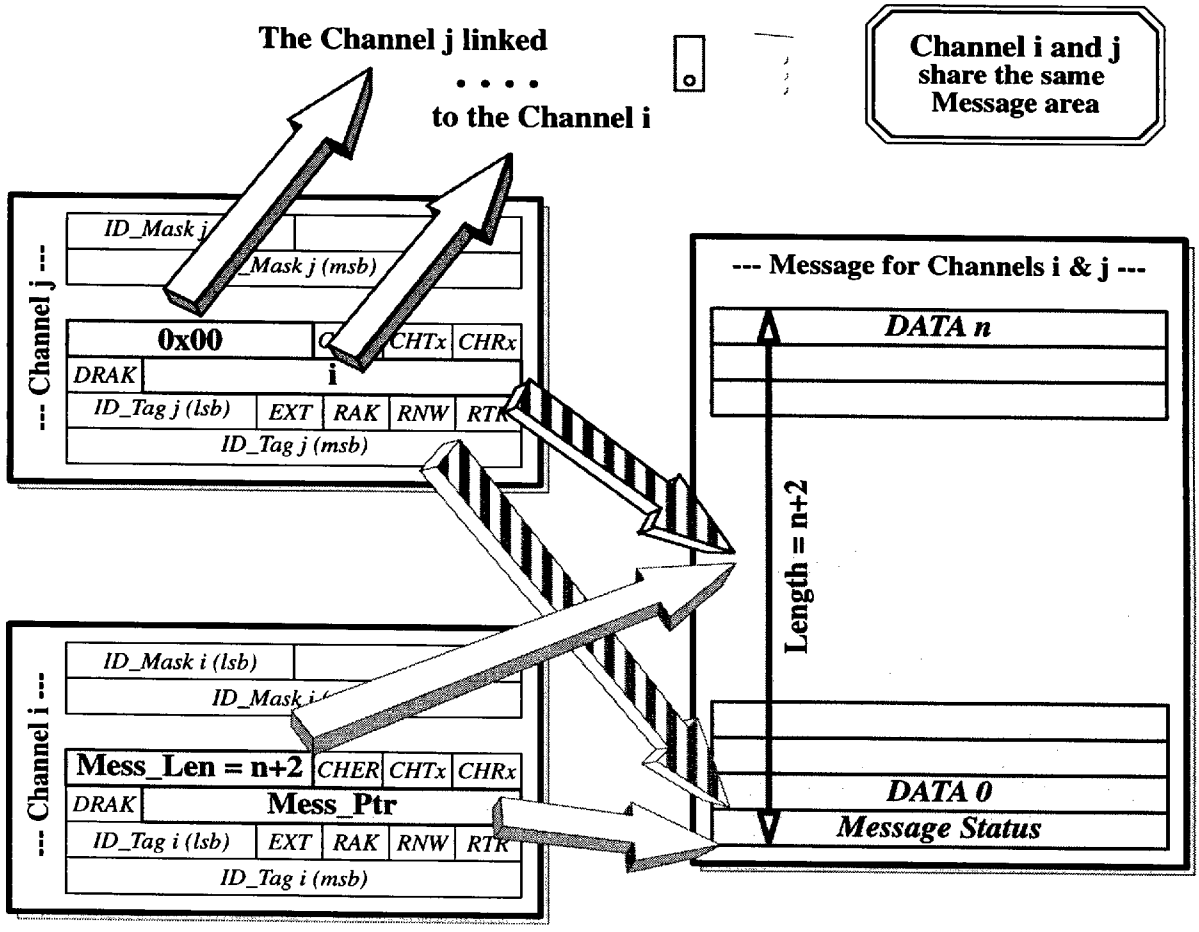


Figure 31. Linkage mechanism

This Message area sharing permits either to optimize the allocation of the 128 bytes of DATA, either to perform some special communications between the different nodes of the network.

16. Absolute Maximum Ratings*

Ambient temperature under bias :

A = Automotive ..... –40°C to 125°C

Storage Temperature ..... –65°C to 150°C

Voltage on V_{CC} to V_{SS} ..... –0.5 to +7.0 V

Voltage on any pin to V_{SS} ..... –0.5 V to V_{CC} + 0.5 V

* NOTICE

Stresses at or 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 exceeding those indicated in the operational sections of this specification is not implied. Expolure to absolute maximum rating conditions may affect device reliability.

17. DC Characteristics

TA = –40°C to 125°C ; VCC = 5 V ± 10 % ; VSS = 0 V

Symbol	Parameter	Min	Max	Type	Test Conditions
V_{IL}	Input Low Voltage (except RESET and XTAL1)	–0.5	0.8	V	
V_{IH}	Input High Voltage (except RESET and XTAL1)	2.0	V_{CC}+0.5	V	
V_{IL1}	Input Low Voltage (RESET and XTAL1)	–0.5	0.3·V_{CC}	V	
V_{IH1}	Input High Voltage (RESET and XTAL1)	0.7·V_{CC}	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 3.2 mA, Vcc min
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –3.2 mA, Vcc min
I_L	Input Leakage Current		±5	µA	$0 < V_{IN} < V_{CC}$
R_{PD}	Input pulldown resistor	110		kΩ	Note 4
C_{IO}	I/O Buffer Capacitance		10	pF	Not tested
I_{CCSB}	Power Supply Current Sleep mode		50	µA	(Note 1)
I_{CCOP}	Power Supply Current Idle or Active mode		15	µA µA	(Notes 2,4) (Notes 2,3)

Notes :
1. Sleep Mode I_{CCSB} is measured according to Figure 32. , with a V_{SS} Clock Signal.
2. Active mode I_{CCOP} is measured at 1 MHz clock, 62.5 KTS/s.
3. Active mode I_{CCOP} is measured at 20 MHz clock, 250 KTS/s.
4. I_{CC} is a function of the Clock Frequency. In Figure 34. is displayed a graph showing I_{CC} versus Clock frequency.
5. RESET, RXD0, RXD1, RXD2 inputs.

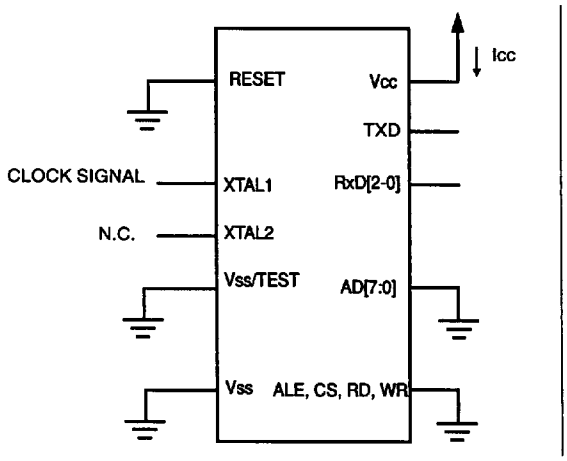


Figure 32. I_{CC}

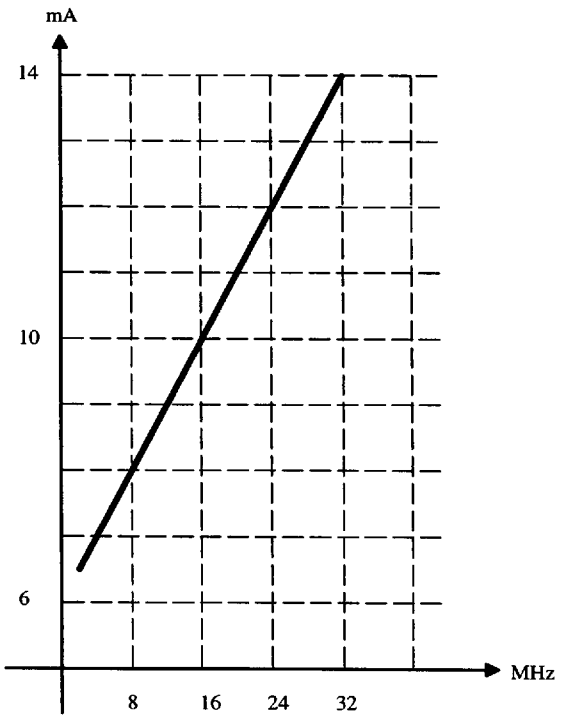


Figure 33. I_{CC} Versus Clock Frequency at 250 KTimeslot/s

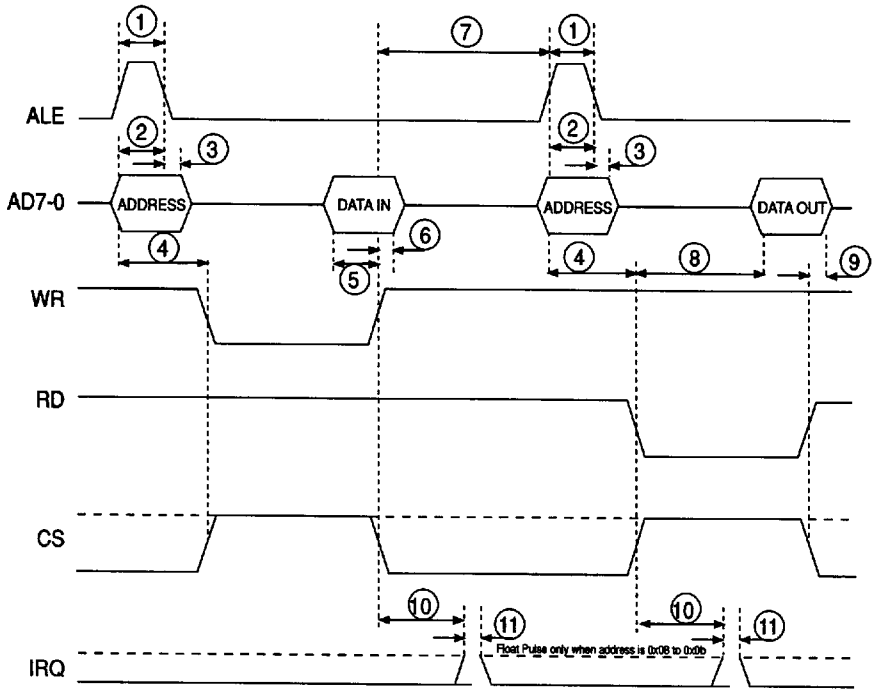
18. AC Characteristics

T_A = –40°C to 125°C ; V_{CC} = 5V ± 10% ; V_{SS} = 0V

18.1. Microprocessor Interface

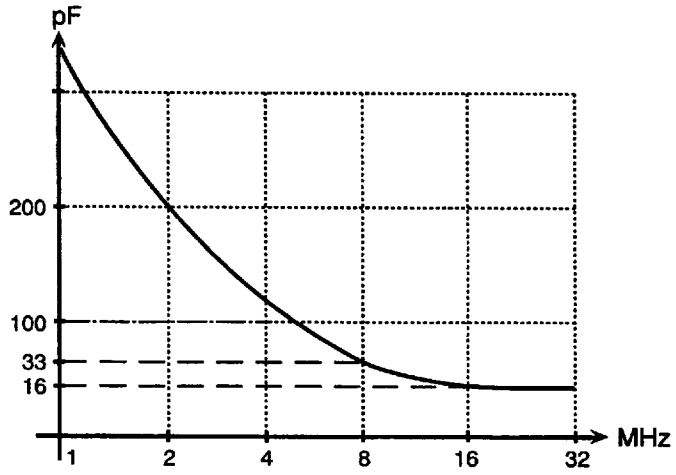
CLOAD = 30pF

	Symbol	Parameter	Min	Max	Unit
	T_{RESET}	RESET High Pulse Width (For Power–up Reset)	15		ns
1	T_{LHLL}	ALE High Pulse Width	10		ns
2	T_{AVLL}	Address Valid to ALE Low Setup Time	10		ns
3	T_{LLAX}	ALE Low to Address Invalid Hold Time	10		ns
4	T_{AVWL}	Address Valid to Command Active Time	20		ns
5	T_{DVWH}	Data Valid to Write Inactive Setup Time	10		ns
6	T_{WHDX}	Write Inactive to Data Invalid Hold Time	TBD		ns
7	T_{WHLH}	Write Inactive to ALE High Recovery Time	20		ns
8	T_{RLDV}	Read Active to Data Valid Access Time		90	ns
9	T_{RHDZ}	Read Inactive to Data Float Time		20	ns
10	T_{WHRLIZ}	Write Inactive or Read Active to IRQ Float Time		90	ns
11	T_{IZIL}	IRQ Float Pulse Width	2	20	ns



18.1.1. Oscillator Characteristics

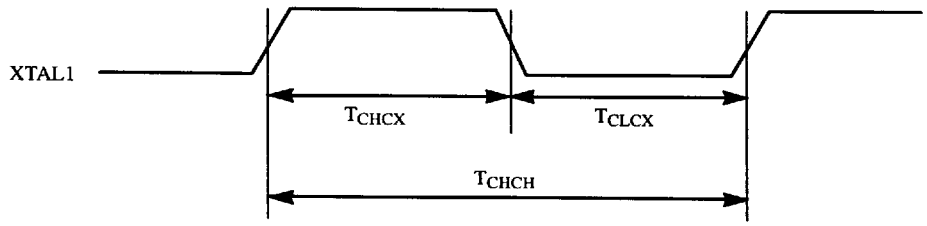
Figure 34. C2 Versus Frequency.



C1 = Crystal load (no capacitance needed)

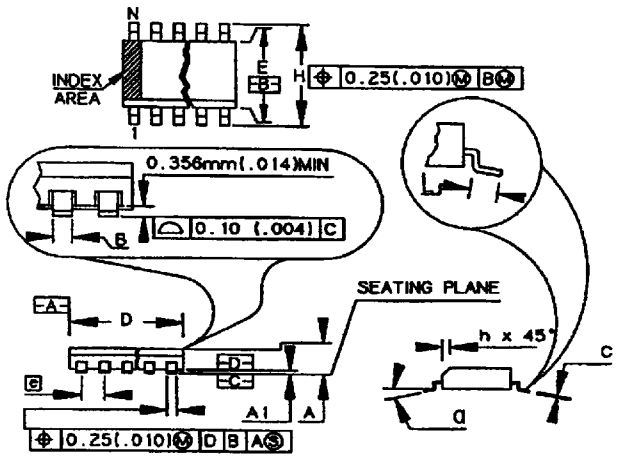
18.2. External Clock drive characteristics (XTAL1)

Symbol	Parameter	Min	Max	Unit
T_{CHCH}	Oscillator period	30		ns
T_{CHCX}	High Time at 32 MHz	10		ns
T_{CLCX}	Low Time at 32 MHz	10		ns



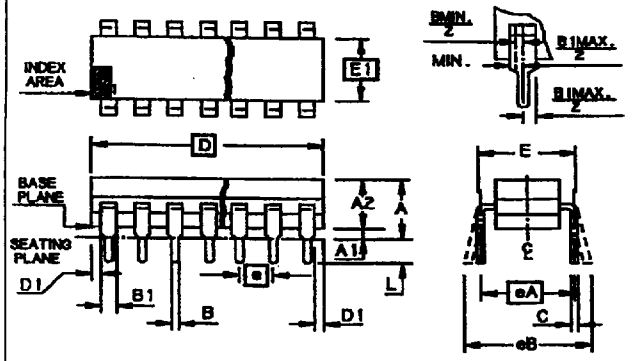
19. Packaging

SO 24



SO	MM		INCH	
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
B	0.35	0.49	0.014	0.019
C	0.23	0.32	0.009	0.013
D	15.20	15.60	0.599	0.614
E	7.40	7.60	0.291	0.299
e	1.27	BSC	0.050	BSC
H	10.00	10.65	0.394	0.419
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
N	24		24	
a	0°		0°	

PDIL 24



PDIL	MM		INCH	
A	–	5.33	–	0.210
A1	0.39	–	0.015	–
A2	2.92	4.95	0.115	0.195
B	0.36	–	0.014	–
B1	1.14	1.78	0.045	0.070
C	0.20	0.38	0.008	0.015
D	28.60	32.30	1.125	1.275
E	7.62	8.25	0.300	0.325
E1	6.10	7.11	0.240	0.280
e	2.54	BSC	0.100	BSC
eA	7.62	BSC	0.300	BSC
eB	–	10.92	–	0.430
L	2.92	4.06	0.115	0.160
D1	0.13	–	0.005	–

20. Ordering Information