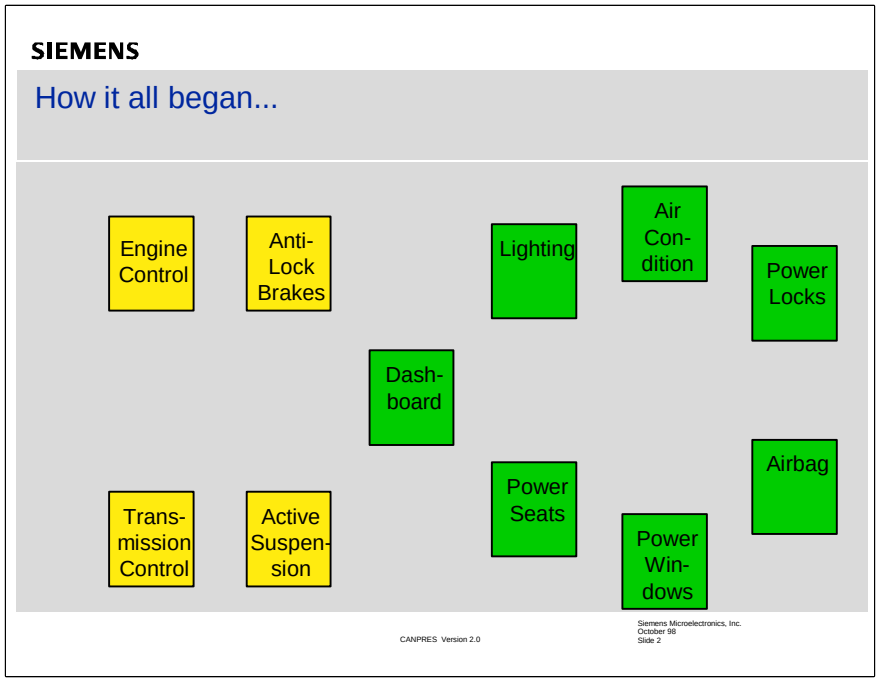
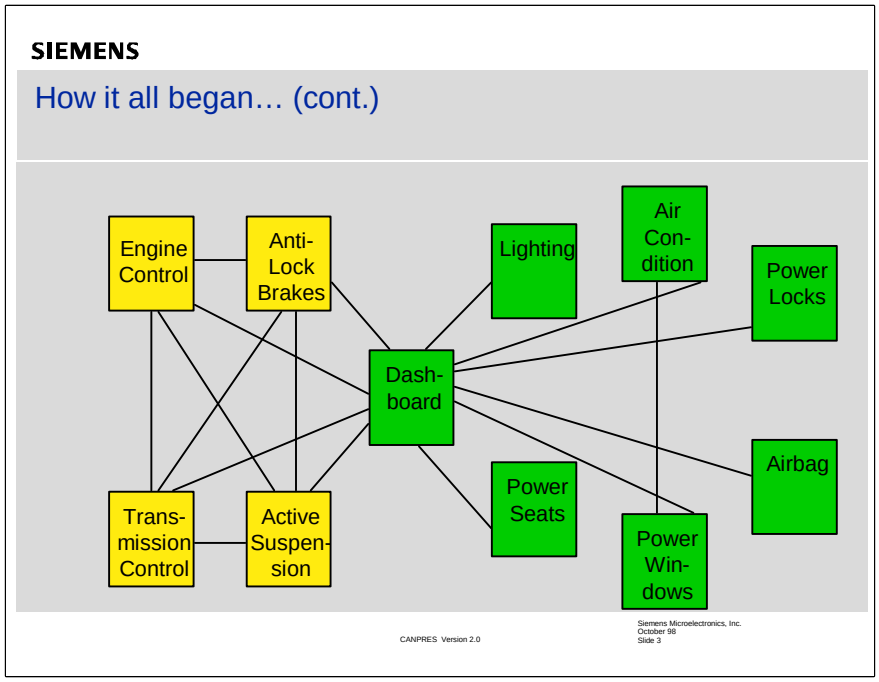
SIEMENS

Controller Area Network

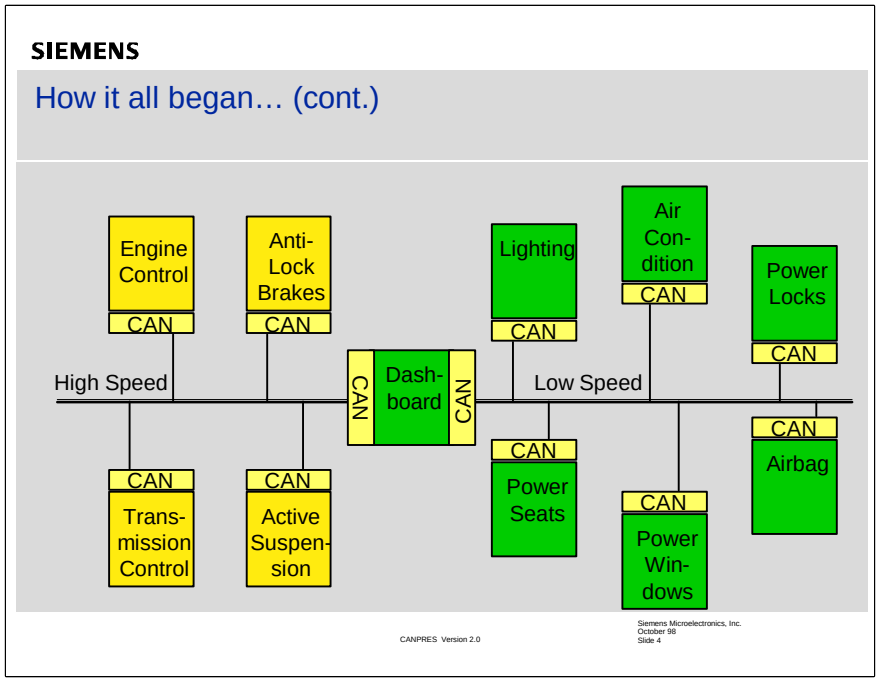
- Fundamentals
  - How it all began...
  - Overview
  - Basic Concepts
- Data Link Layer
  - Bus Access Arbitration
  - Frame Formats
  - Error Detection
  - Error Handling
  - Protocol Versions (Standard /Extended)
- Physical Layer
  - Message Coding
  - Synchronization
  - Bit Timing / Bit Construction
  - CAN Bus Lines
- CAN Implementations
- User Benefits
- Application Examples
- Siemens CAN Devices



•The development of CAN began when more and more electronic devices were implemented into modern motor vehicles. Examples of such devices include engine management systems, active suspension, ABS, gear control, lighting control, air conditioning, airbags and central locking. All this means more safety and more comfort for the driver and of course a reduction of fuel consumption and exhaust emissions.

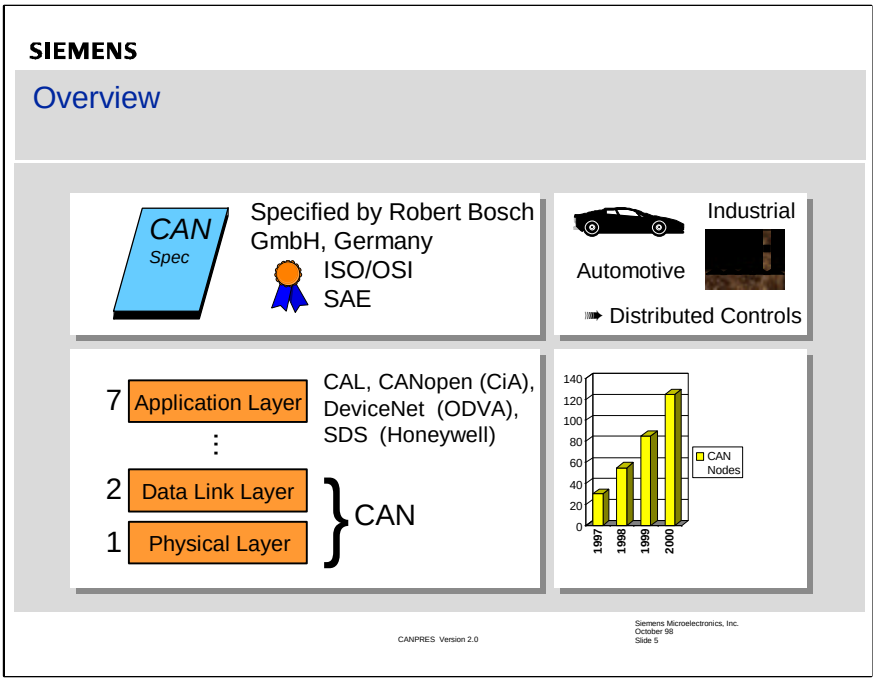


•To improve the behavior of the vehicle even further, it was necessary for the different control systems (and their sensors) to exchange information. This was usually done by discrete interconnection of the different systems (i.e. point to point wiring). The requirement for information exchange has then grown to such an extent that a cable network with a length of up to several miles and many connectors was required. This produced growing problems concerning material cost, production time and reliability.



•The solution to this problem was the connection of the control systems via a serial bus system. This bus had to fulfill some special requirements due to its usage in a vehicle.

•With the use of CAN, point-to-point wiring is replaced by one serial bus connecting all control systems. This is accomplished by adding some CAN-specific hardware to each control unit that provides the "rules" or the protocol for transmitting and receiving information via the bus.



•CAN or Controller Area Network is an advanced serial bus system that efficiently supports distributed control systems. It was initially developed for the use in motor vehicles by Robert Bosch GmbH, Germany, in the late 1980s, also holding the CAN license.

•CAN is internationally standardized by the International Standardization Organization (ISO) and the Society of Automotive Engineers (SAE).

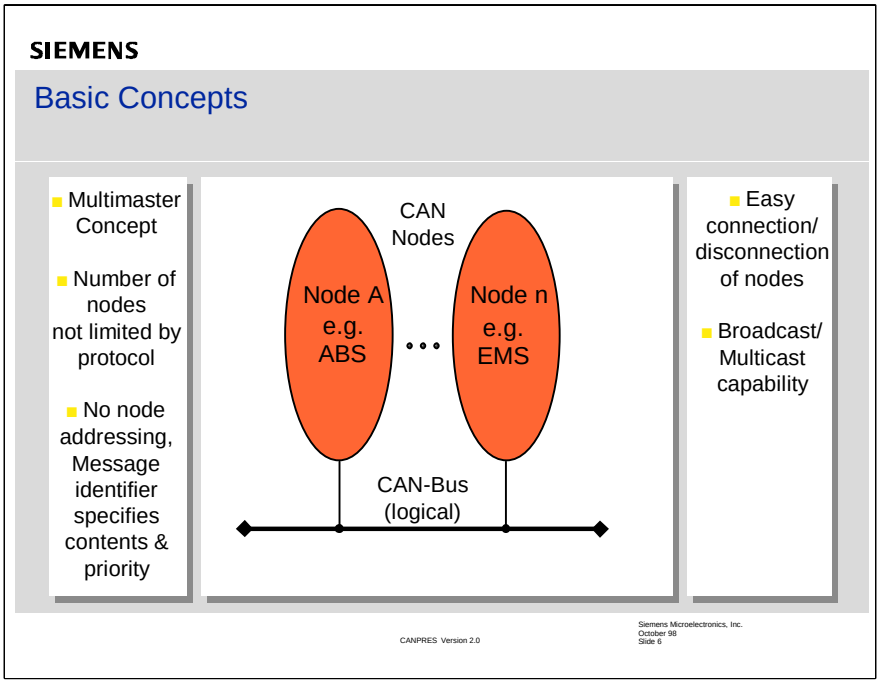
•The CAN protocol uses the Data Link Layer and the Physical Layer in the ISO - OSI model. There are also a number of higher level protocols available for CAN.

•CAN is most widely used in the automotive and industrial market segments. Typical applications for CAN are motor vehicles, utility vehicles, and industrial automation. Other applications for CAN are trains, medical equipment, building automation, household appliances, and office automation. Due to the high volume production in the automotive and industrial markets, low cost protocol devices are available.

•There are about 20 million CAN nodes in use worldwide. By the year 2000 the number of nodes is estimated to be 140 million.

•Examples of vehicle bus systems, other than CAN, are A-BUS from Volkswagen, VAN or Vehicle Area Network, from Peugeot and Renault, and J1850 from Chrysler, General Motors and Ford.

•CAN is clearly the leading vehicle bus protocol in Europe.

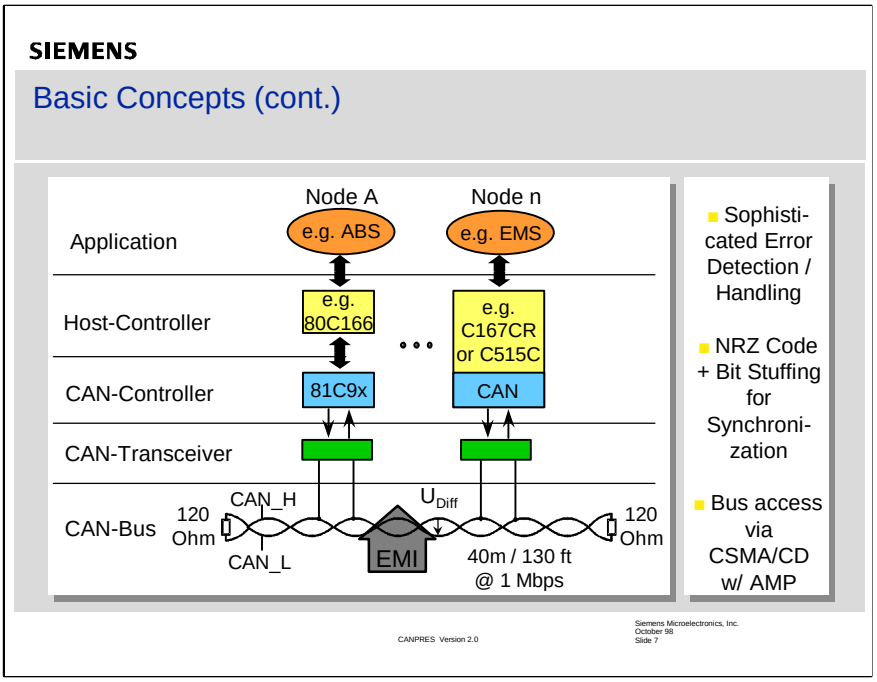


•CAN is a multi-master bus with an open, linear structure with one logic bus line and equal nodes. The number of nodes is not limited by the protocol.

•In the CAN protocol, the bus nodes do not have a specific address. Instead, the address information is contained in the identifiers of the transmitted messages, indicating the message content and the priority of the message.

•The number of nodes may be changed dynamically without disturbing the communication of the other nodes.

•Multicasting and Broadcasting is supported by CAN.



•CAN provides sophisticated error-detection and error handling mechanisms such as CRC check, and high immunity against electromagnetic interference. Erroneous messages are automatically retransmitted. Temporary errors are recovered. Permanent errors are followed by automatic switch-off of defective nodes. There is guaranteed system-wide data consistency.

•The CAN protocol uses Non-Return-to-Zero or NRZ bit coding. For synchronization purposes, Bit Stuffing is used.

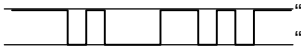
•There is a high data transfer rate of 1000 kilobits per second at a maximum bus length of 40 meters or 130 feet when using a twisted wire pair which is the most common bus medium used for CAN. Message length is short with a maximum of 8 data bytes per message and there is a low latency between transmission request and start of transmission.

•The bus access is handled via the advanced serial communications protocol Carrier Sense Multiple Access/Collision Detection with Non-Destructive Arbitration. This means that collision of messages is avoided by bitwise arbitration without loss of time.

SIEMENS

Basic Concepts - CAN Bus Characteristics - Wired-AND

Two logic states possible on the bus:
“1” = recessive
“0” = dominant



A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

As soon as one node nodes transmits a dominant bit (zero):
Bus is in the dominant state.

Only if all nodes transmit recessive bits (ones):
Bus is in the recessive state.

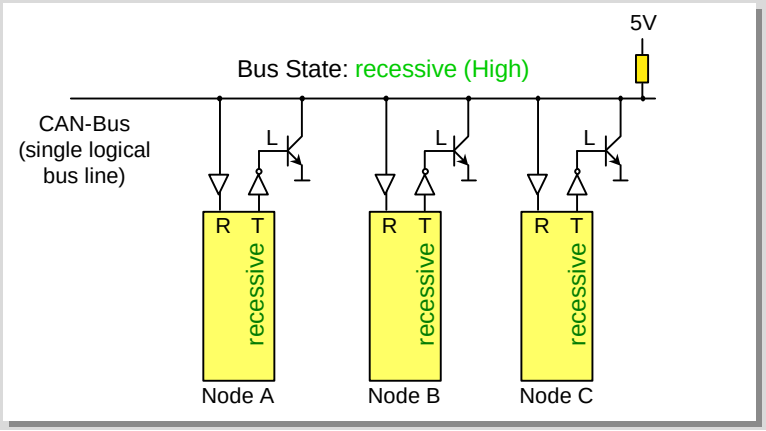
CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 8

- There are two bus states, called "dominant" and "recessive".
- The bus logic uses a "Wired-AND" mechanism, that is, "dominant bits" (equivalent to the logic level "Zero") overwrite the "recessive" bits (equivalent to the logic level "One").

SIEMENS

Basic Concepts - CAN Bus Characteristics - Wired-AND (cont.)



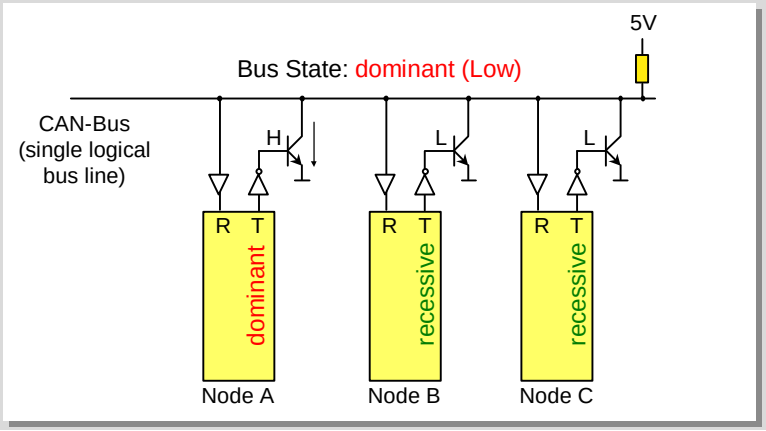
CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 9

•Only if all nodes transmit recessive bits (ones), the Bus is in the recessive state.

SIEMENS

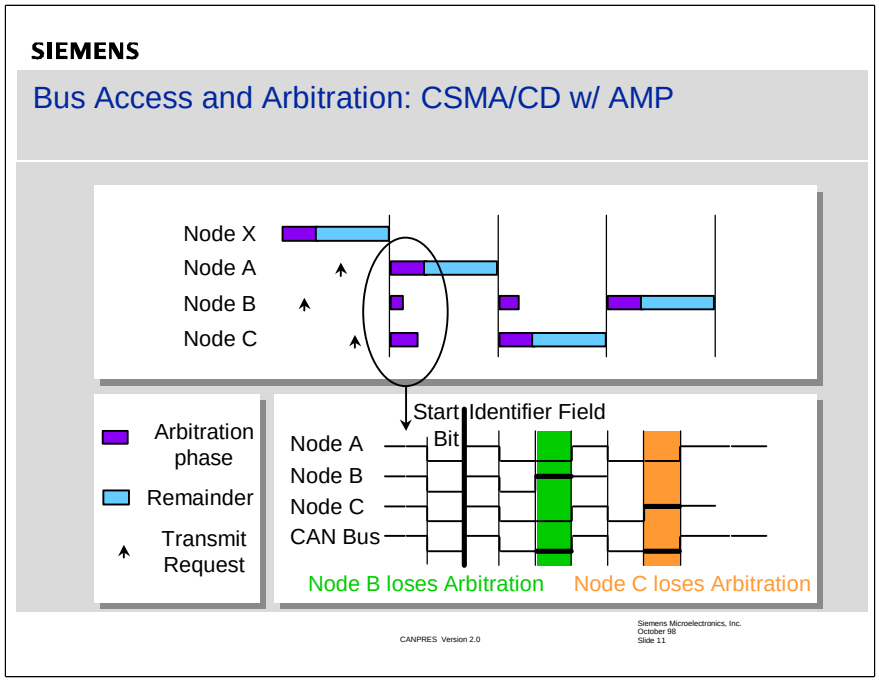
Basic Concepts -
CAN Bus Characteristics - Wired-AND (cont.)



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 10

•As soon as one node transmits a dominant bit (zero), the bus is in the dominant state.



•The CAN protocol handles bus accesses according to the concept called “Carrier Sense Multiple Access with Arbitration on Message Priority”. This arbitration concept avoids collisions of messages whose transmission was started by more than one node simultaneously and makes sure the most important message is sent first without time loss. In the picture above you see the trace of the transmit pins of three bus nodes called A, B and C, and the resulting bus state according to the wired-AND principle.

•If two or more bus nodes start their transmission at the same time after having found the bus to be idle, collision of the messages is avoided by bitwise arbitration. Each node sends the bits of its message identifier and monitors the bus level.

•At a certain time nodes A and C send a dominant identifier bit. Node B sends a recessive identifier bit but reads back a dominant one. Node B loses bus arbitration and switches to receive mode. Some bits later node C loses arbitration against node A. This means that the message identifier of node A has a lower binary value and therefore a higher priority than the messages of nodes B and C. In this way, the bus node with the highest priority message wins arbitration without losing time by having to repeat the message.

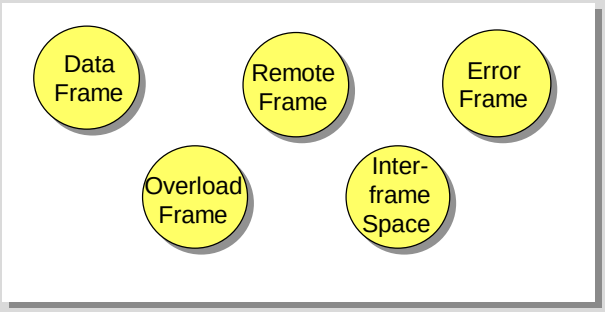
•Nodes B and C automatically try to repeat their transmission once the bus returns to the idle state. Node B loses against node C, so the message of node C is transmitted next, followed by node B’s message.

•It is not permitted for different nodes to send messages with the same identifier as arbitration could fail leading to collisions and errors.

SIEMENS

Frame Formats - Overview

- **Existing Frame Formats:**



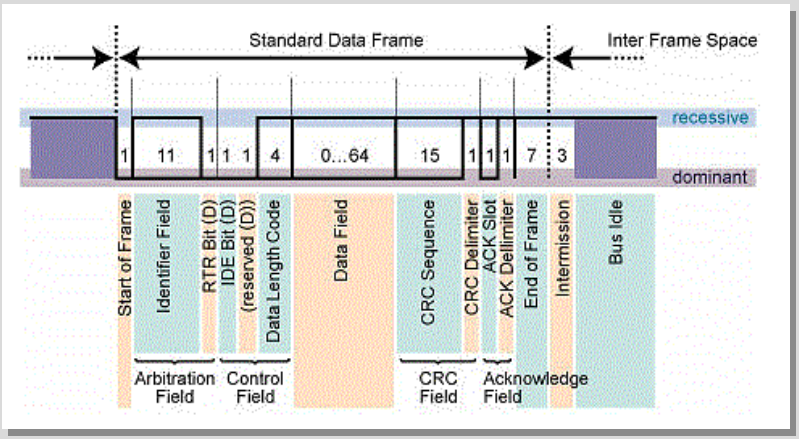
CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 12

•These are the existing Frame formats which are now discussed in the following slides.

SIEMENS

Frame Formats - Data Frame



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 13

•A "Data Frame" is generated by a CAN node when the node wishes to transmit data. The Standard CAN Data Frame is shown above. The frame begins with a dominant Start Of Frame bit for hard synchronization of all nodes.

•The Start of Frame bit is followed by the Arbitration Field consisting of 12 bits: The 11-bit Identifier, which reflects the contents and priority of the message, and the Remote Transmission Request bit. The Remote transmission request bit is used to distinguish a Data Frame (RTR = dominant) from a Remote Frame (RTR = recessive).

•The next field is the Control Field, consisting of 6 bits. The first bit of this field is called the IDE bit (Identifier Extension) and is at dominant state to specify that the frame is a Standard Frame. The following bit is reserved and defined as a dominant bit. The remaining 4 bits of the Control Field are the Data Length Code (DLC) and specify the number of bytes of data contained in the message (0 - 8 bytes).

•The data being sent follows in the Data Field which is of the length defined by the DLC above (0, 8, 16, ..., 56 or 64 bits).

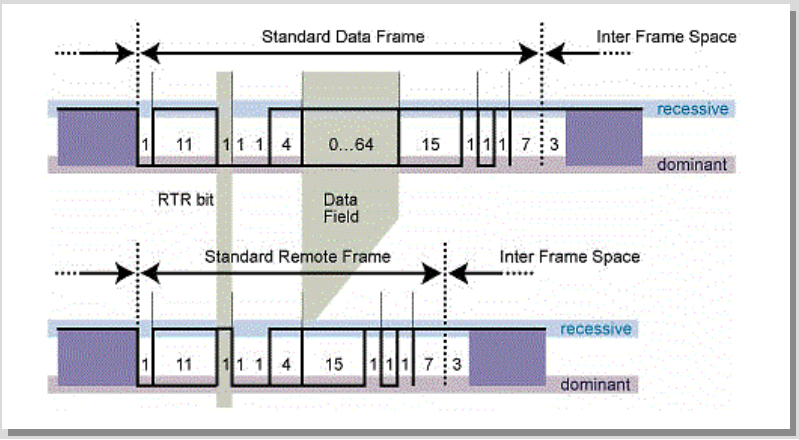
•The Cyclic Redundancy Field (CRC field) follows and is used to detect possible transmission errors. The CRC Field consists of a 15 bit CRC sequence, completed by the recessive CRC Delimiter bit.

•The next field is the Acknowledge Field. During the ACK Slot bit the transmitting node sends out a recessive bit. Any node that has received an error free frame acknowledges the correct reception of the frame by sending back a dominant bit (regardless of whether the node is configured to accept that specific message or not). From this it can be seen that CAN belongs to the "in-bit-response" group of protocols. The recessive Acknowledge Delimiter completes the Acknowledge Slot and may not be overwritten by a dominant bit.

•Seven recessive bits (End of Frame) end the Data Frame.

SIEMENS

Frame Formats - Remote Frame



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 14

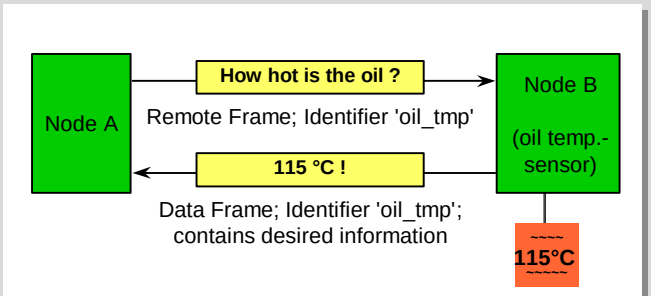
•Generally data transmission is performed on an autonomous basis with the data source node (e.g. a sensor) sending out a Data Frame. It is also possible, however, for a destination node to request the data from the source by sending a Remote Frame.

•There are 2 differences between a Data Frame and a Remote Frame. Firstly the RTR-bit is transmitted as a dominant bit in the Data Frame and secondly in the Remote Frame there is no Data Field. In the very unlikely event of a Data Frame and a Remote Frame with the same identifier being transmitted at the same time, the Data Frame wins arbitration due to the dominant RTR bit following the identifier. In this way, the node that transmitted the Remote Frame receives the desired data immediately.

SIEMENS

Frame Formats - Remote Frame (cont.)

- **Remote Frame scenario**



CANPRES Version 2.0

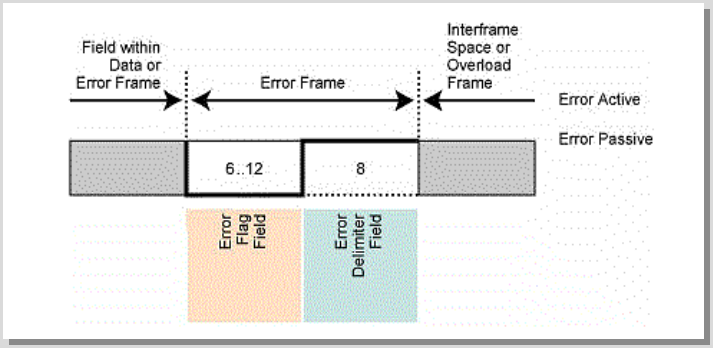
Siemens Microelectronics, Inc.
October 98
Slide 15

•If a node wishes to request the data from the source, it sends a Remote Frame with an identifier that matches the identifier of the required Data Frame. The appropriate data source node will then send a Data Frame as a response to this remote request.

SIEMENS

Frame Formats - Error Frame

- **Active Error Frame: Used for error signaling**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 16

•An *Error Frame* is generated by any node that detects a bus error. The Error Frame consists of 2 fields, an Error Flag field followed by an Error Delimiter field. The Error Delimiter consists of 8 recessive bits and allows the bus nodes to restart bus communications cleanly after an error. There are, however, two forms of Error Flag fields. The form of the Error Flag field depends on the "error status" of the node that detects the error.

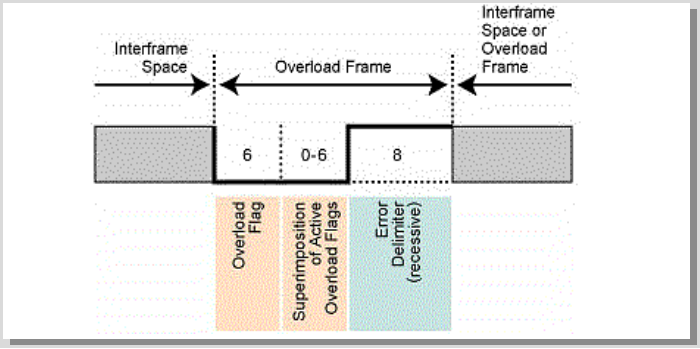
•If an "error-active" node detects a bus error then the node interrupts transmission of the current message by generating an "active error flag". The "active error flag" is composed of six consecutive dominant bits. This bit sequence actively violates the bit stuffing rule. All other stations recognize the resulting bit stuffing error and in turn generate Error Frames themselves. The Error Flag field therefore consists of between six and twelve consecutive dominant bits (generated by one or more nodes). The Error Delimiter field completes the Error Frame. After completion of the Error Frame bus activity returns to normal and the interrupted node attempts to resend the aborted message.

•If an "error passive" node detects a bus error then the node transmits an "passive Error Flag" followed, again, by the Error Delimiter field. The "passive Error Flag" consists of six consecutive recessive bits, and therefore the Error Frame (for an "error passive" node) consists of 14 recessive bits (i.e. no dominant bits). From this it follows that, unless the bus error is detected by the node that is actually transmitting (i.e. is the bus master), the transmission of an Error Frame by an "error passive" node will not affect any other node on the network. If the bus master node generates an "error passive flag" then this may cause other nodes to generate error frames due to the resulting bit stuffing violation.

SIEMENS

Frame Formats - Overload Frame

- **Overload Frame: Used to delay next CAN message**



CANPRES Version 2.0

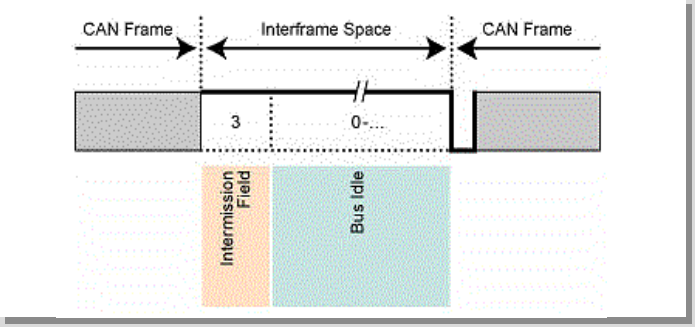
Siemens Microelectronics, Inc.
October 98
Slide 17

•An Overload Frame has the same format as an “active” Error Frame. An Overload Frame, however can only be generated during Interframe Space. This is the way then an Overload Frame can be differentiated from an Error Frame (an Error Frame is sent during the transmission of a message). The Overload Frame consists of 2 fields, an Overload Flag followed by an Overload Delimiter. The Overload Flag consists of six dominant bits followed by Overload Flags generated by other nodes (as for “active error flag”, again giving a maximum of twelve dominant bits). The Overload Delimiter consists of eight recessive bits. An Overload Frame can be generated by a node if due to internal conditions the node is not yet able to start reception of the next message. A node may generate a maximum of 2 sequential Overload Frames to delay the start of the next message.

SIEMENS

Frame Formats - Interframe Space

- **Interframe Space: Separates a frame (of whatever type) from a following Data or Remote Frame.**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 18

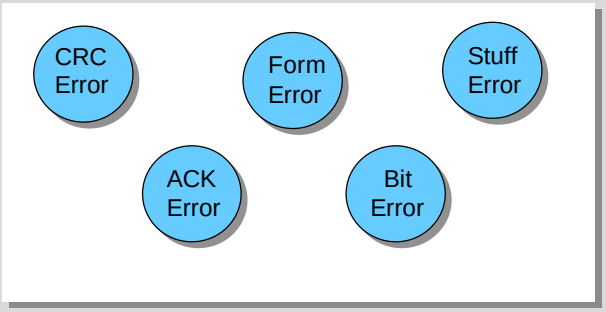
•Interframe Space separates a preceeding frame (of whatever type) from a following Data or Remote Frame. Interframe space is composed of at least 3 recessive bits, these bits are termed the Intermission. This time is provided to allow nodes time for internal processing before the start of the next message frame. After the Intermission, for error active CAN nodes the bus line remains in the recessive state (Bus Idle) until the next transmission starts.

•The Interframe Space has a slightly different format for error passive CAN nodes which were the transmitter of the previous message. In this case, these nodes have to wait another eight recessive bits called Suspend Transmission before the bus turns into bus idle for them after Intermission and they are allowed to send again. Due to this mechanism error active nodes have the chance to transmit their messages before the error passive nodes are allowed to start a transmission.

SIEMENS

Error Detection - Overview

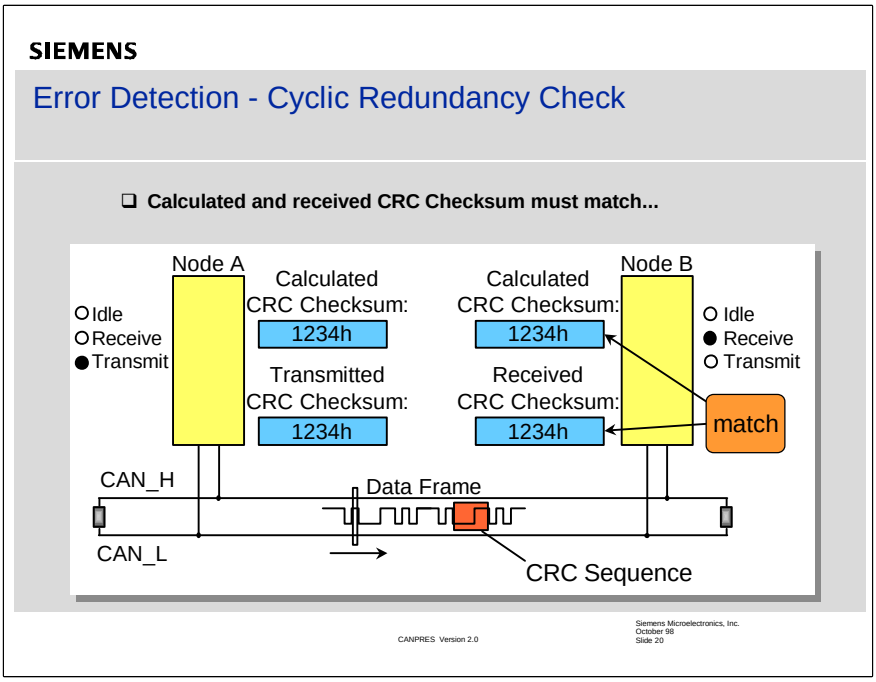
- **Detected Errors:**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 19

•The CAN protocol provides sophisticated error detection mechanisms discussed in the following slides.



•With the Cyclic Redundancy Check, the transmitter calculates a check sum for the bit sequence from the start of frame bit until the end of the Data Field.

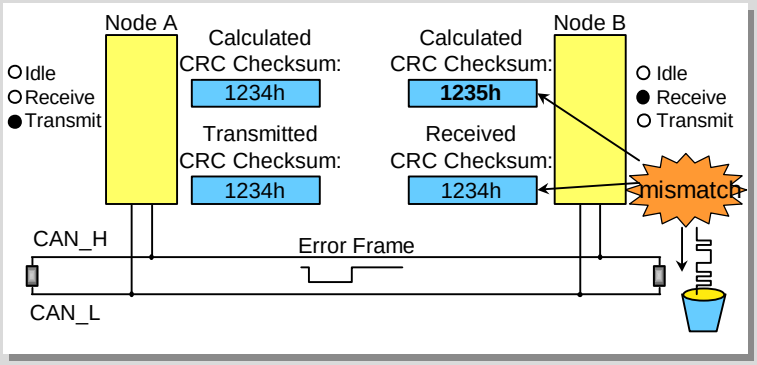
•This CRC sequence is transmitted in the CRC Field of the CAN frame.

•The receiving node also calculates the CRC sequence using the same formula and performs a comparison to the received sequence.

SIEMENS

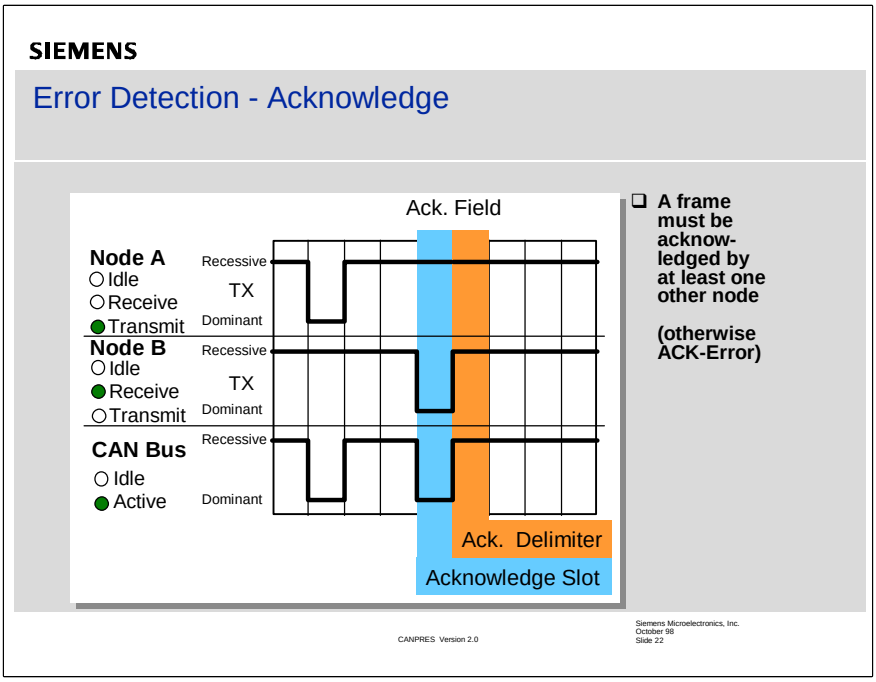
Error Detection - Cyclic Redundancy Check (cont.)

- … otherwise Frame was not received correctly (CRC Error)



•If node B detects a mismatch between the calculated and the received CRC sequence , then a CRC error has occurred.

•Node B discards the message and transmits an Error Frame to request retransmission of the garbled frame.



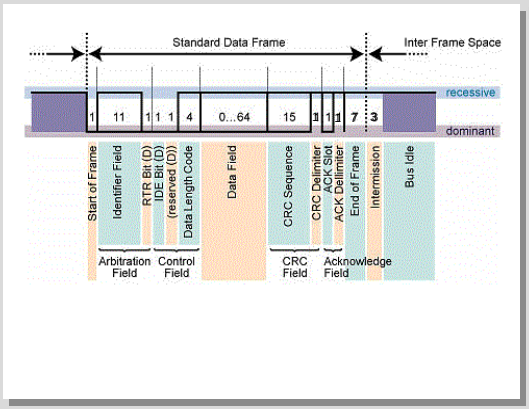
•With the Acknowledge Check the transmitter checks in the Acknowledge Field of a message to determine if the Acknowledge Slot, which is sent out as a recessive bit, contains a dominant bit.

•If this is the case, at least one other node, (here node B) has received the frame correctly.

•If not, an Acknowledge Error has occured and the message has to be repeated. No Error Frame is generated, though.

SIEMENS

Error Detection - Frame Check



- **No dominant bits allowed in CRC Delimiter, ACK Delimiter, End of Frame, Intermission**

  (Form Error)

CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 23

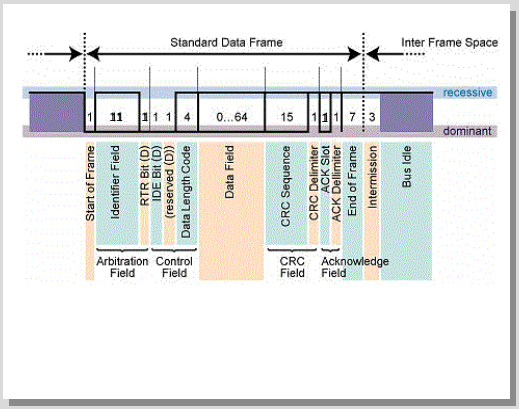
•Another error detection mechanism is the Frame Check. If a transmitter detects a dominant bit in one of the four segments:

CRC Delimiter,
Acknowledge Delimiter,
End of Frame or
Interframe Space

then a Form Error has occurred and an Error Frame is generated. The message will then be repeated.

SIEMENS

Error Detection - Bit Monitoring



- **A transmitted bit must be correctly read back from the CAN bus (otherwise Bit Error)**
- **Dominant bits may overwrite recessive bits only in the Arbitration Field and in the Acknowledge Slot**

CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 24

•All nodes perform Bit Monitoring: A Bit Error occurs if a transmitter

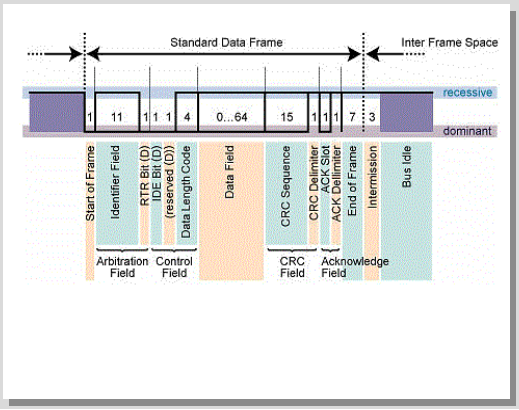
sends a dominant bit but detects a recessive bit on the bus line or, sends a recessive bit but detects a dominant bit on the bus line.

•An Error Frame is generated and the message is repeated.

•When a dominant bit is detected instead of a recessive bit, no error occurs during the Arbitration Field or the Acknowledge Slot because these fields must be able to be overwritten by a dominant bit in order to achieve arbitration and acknowledge functionality.

SIEMENS

Error Detection - Bit Stuffing Check



- **6 consecutive bits with same polarity are not allowed between Start Of Frame and CRC Delimiter (otherwise Bit Stuffing Error)**

CANPRES Version 2.0

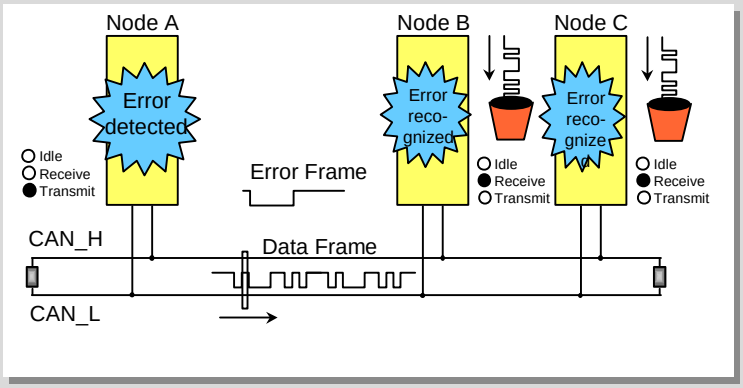
Siemens Microelectronics, Inc.
October 98
Slide 25

•If six consecutive bits with the same polarity are detected between Start of Frame and the CRC Delimiter, the bit stuffing rule has been violated.

•A stuff error occurs and an Error Frame is generated. The message is then repeated.

SIEMENS

Error Handling

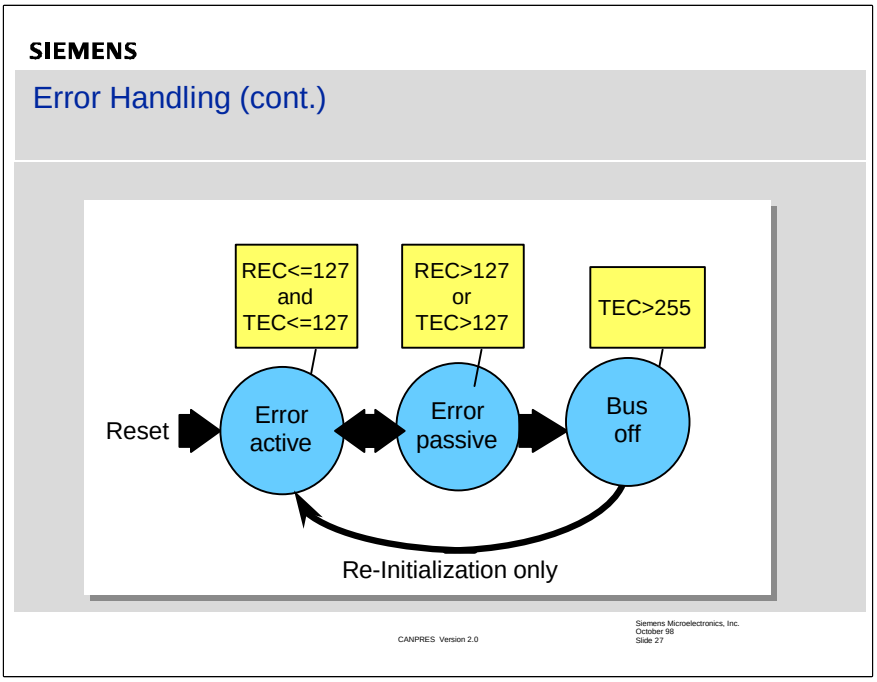


CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 26

•Detected errors are made public to all other nodes via Error Frames.

•The transmission of the erroneous message is aborted and the frame is repeated as soon as possible.



•Each CAN node is in one of three error states "error active", "error passive" or "bus off" according to the value of their internal error counters.

•The error-active state is the usual state after reset. The bus node can then receive and transmit messages and transmit active Error Frames (made of dominant bits) without any restrictions. During CAN communication, the error counters are updated according to quite complex rules. For each error on reception or transmission, the error counters are incremented by a certain value. For each successful transaction, the error counters are decremented by a certain value. The error active state is valid as long as both error counters are smaller than or equal to 127.

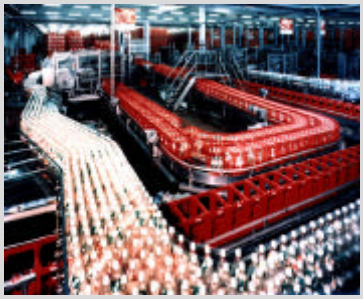
•If either the receive or the transmit error counter has reached the value of 128, the node switches to the error-passive state. In the error-passive state, messages can still be received and transmitted, although, after transmission of a message the node must suspend transmission. It must wait 8 bit times longer than error-active nodes before it may transmit another message. In terms of error signaling, only passive Error Frames (made of recessive bits) may be transmitted by an error-passive node.

•If both error counters go below 128 again due to successful bus communication, the node switches back to the error-active state.

•One feature of the CAN protocol is that faulty nodes withdraw from the bus automatically. The bus-off state is entered if the transmit error counter exceeds the value of 255. All bus activities are stopped which makes it temporarily impossible for the station to participate in the bus communication. During this state, messages can be neither received nor transmitted. To return to the error active state and to reset the error counter values, the CAN node has to be reinitialized.

SIEMENS

Undetected Errors - an example



- 2000 h/year
- 500 kbps
- 25% bus load

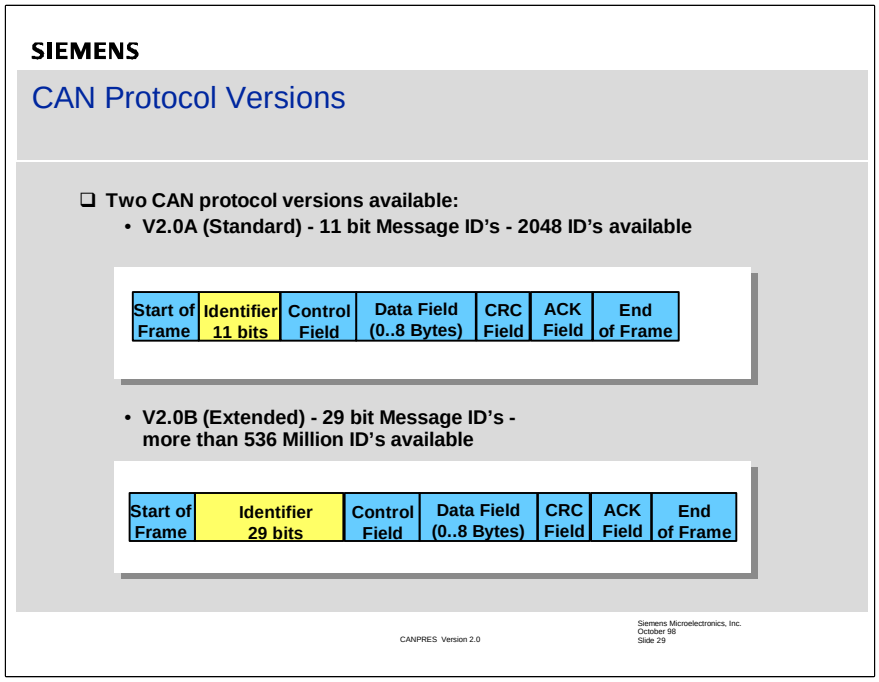
- 1 undetected error every 1000 years

CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 28

•To understand the error detection capabilities of CAN, imagine a vehicle equipped with CAN running 2000 hours per year at a CAN bus speed of 500 kbps with 25% bus load.

•This will result in 1 undetected error every 1000 years.



•The original CAN specifications (Versions 1.0, 1.2 and 2.0A) specify an 11 bit message identifier. This is known as "Standard CAN".

•Those Data Frames and Remote Frames, which contain an 11-bit identifier are therefore called Standard Frames.

•With these frames, 211 (=2048) different messages can be identified (identifiers 0-2047).

•However, the 16 messages with the lowest priority (2032-2047) are reserved.

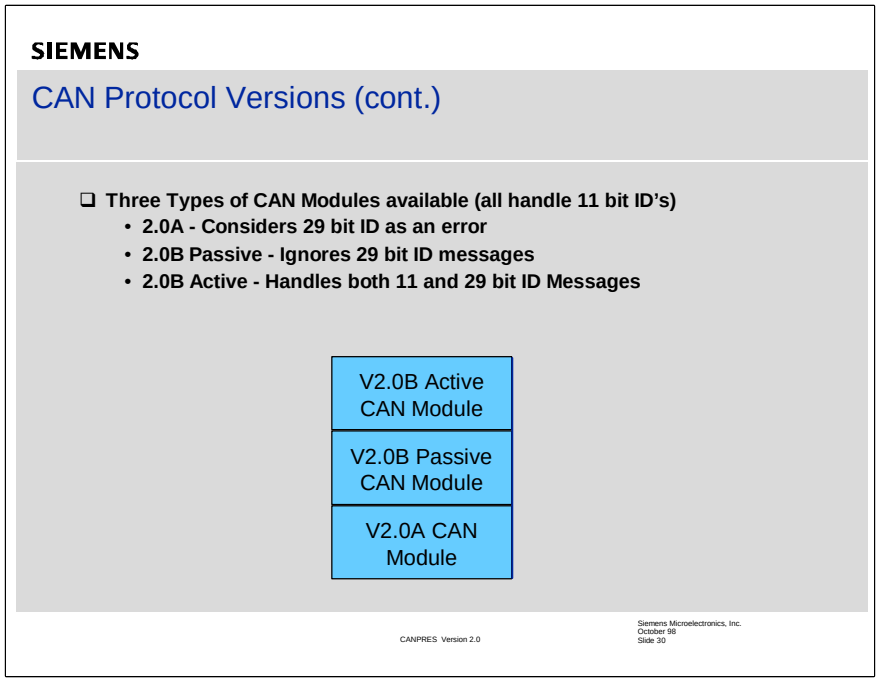
•Specification V2.0A has since been updated (to version 2.0B) to remove this possible message number limitation and meet the SAE J1939 standard for the use of CAN in trucks.

•Version 2.0B CAN is referred to as "Extended CAN".

•Extended Frames, according to CAN specification V2.0 part B, contain a 29-bit identifier which allows 229 (over 536 Million) message identifiers.

•The 29-bit identifier is made up of the 11-bit identifier ("Base ID") and the 18-bit Extended Identifier ("ID Extension").

•CAN specification Version 2.0B still allows message identifier lengths of 11 bits to be used.

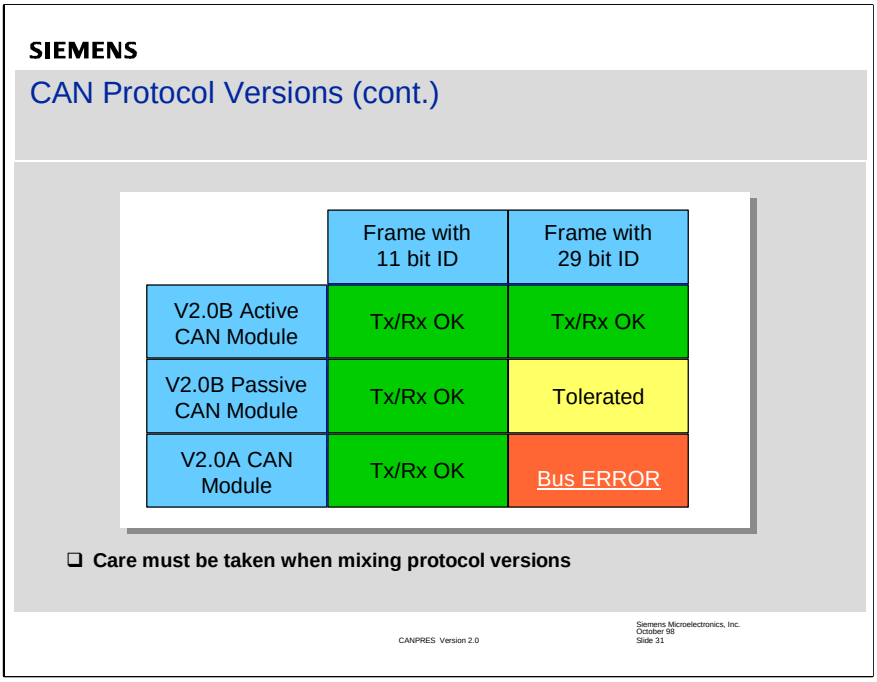


•There are three different types of CAN modules available.

2.0A - Considers 29 bit ID as an error

2.0B Passive - Ignores 29 bit ID messages

2.0B Active - Handles both 11 and 29 bit ID Messages



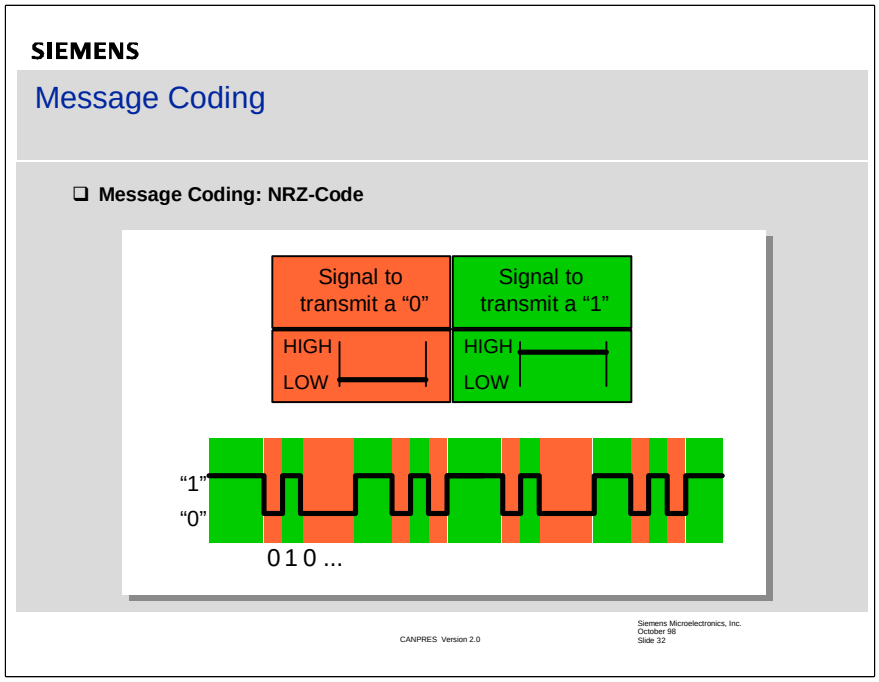
•CAN modules specified after CAN V2.0 part A are only able to transmit and receive Standard Frames according to the Standard CAN protocol.

•Messages using the 29-bit identifier sent to a Standard CAN module cause errors.

•If a device is specified after CAN V2.0 part B, there's one more distinction. Modules named "V2.0B Passive" can only transmit and receive Standard Frames but tolerate Extended Frames without generating Error Frames.

•"V2.0B Active" devices are able to transmit and receive both Standard and Extended Frames.

•Siemens offers V2.0B Active and V2.0B Passive devices.



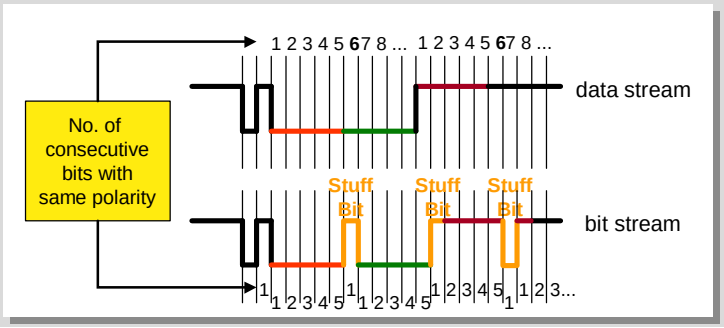
•The CAN protocol uses Non-Return-to-Zero or NRZ bit coding. This means that the signal is constant for one whole bit time and only one time segment is needed to represent one bit.

•Usually, but not always, a "zero" corresponds to a dominant bit, placing the bus in the dominant state, and a "one" corresponds to a recessive bit, placing the bus in the recessive state.

SIEMENS

Bit Stuffing

- **Bit Stuffing ensures sufficient Recessive to Dominant edges**
  - **Stuff Bit inserted after 5 consecutive bits at the same state**
  - **Stuff Bit is inverse of previous bit**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 33

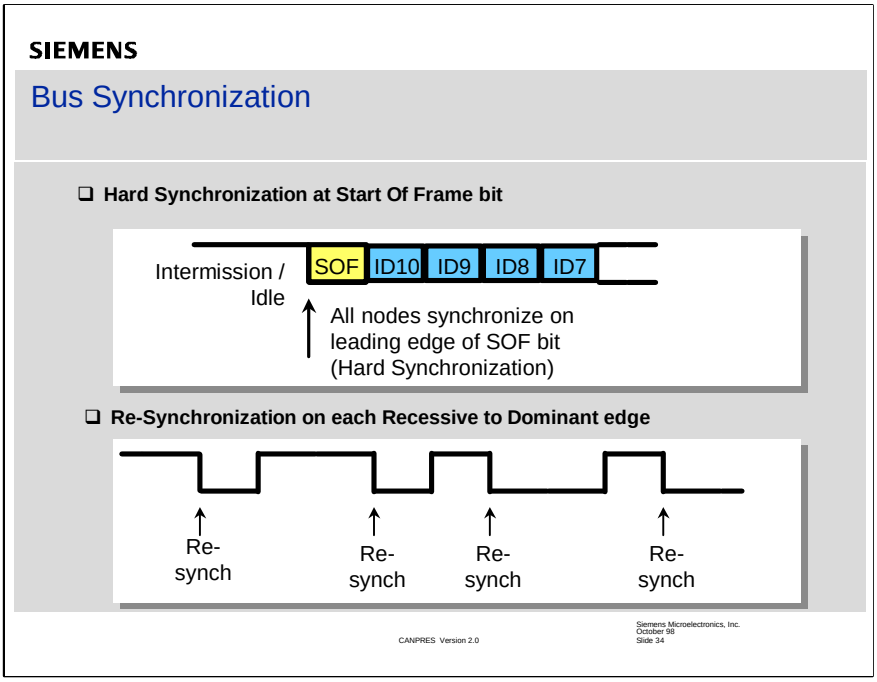
•One characteristic of Non-Return-to-Zero code is that the signal provides no edges that can be used for resynchronization when transmitting a large number of consecutive bits with the same polarity.

•Therefore Bit stuffing is used to ensure synchronization of all bus nodes.

•This means that during the transmission of a message, a maximum of five consecutive bits may have the same polarity.

•Whenever five consecutive bits of the same polarity have been transmitted, the transmitter will insert one additional bit of the opposite polarity into the bit stream before transmitting further bits.

•The receiver also checks the number of bits with the same polarity and removes the stuff bits again from the bit stream. This is called "destuffing".



•In contrast to many other field buses, CAN handles message transfers synchronously.

•All nodes are synchronized at the beginning of each message with the first falling edge of a frame which belongs to the Start Of Frame bit.

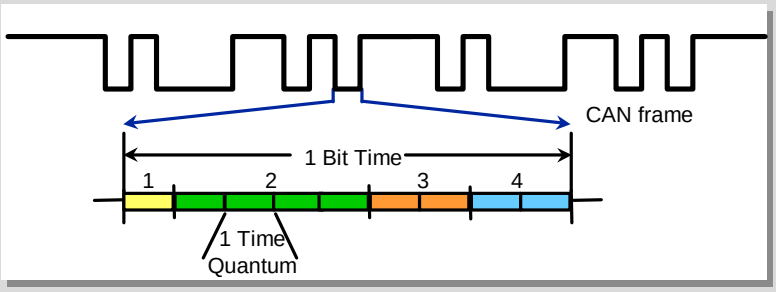
•This is called Hard Synchronization.

•To ensure correct sampling up to the last bit, the CAN nodes need to re-synchronize throughout the entire frame. This is done on each recessive to dominant edge.

SIEMENS

Bit Construction

- 4 Segments, 8-25 Time Quanta (TQ) per bit time
- Time Quanta generated by programmable divide of Oscillator
- CAN Baud Rate (= 1 / Bit Time) programmed by selection of appropriate TQ length + appropriate number of TQ per bit



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 35

•One CAN bit time (or one high or low pulse of the NRZ code) is specified as four non-overlapping time segments.

•Each segment is constructed from an integer multiple of the Time Quantum.

•The Time Quantum or TQ is the smallest discrete timing resolution used by a CAN node.

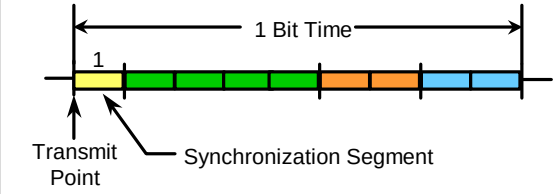
•Its length is generated by a programmable divide of the CAN node's oscillator frequency.

•There is a minimum of 8 and a maximum of 25 Time Quanta per bit.

•The bit time, and therefore the bit rate, is selected by programming the width of the Time Quantum and the number of Time Quanta in the various segments.

SIEMENS

Synchronization Segment



- Next bit is output at start of this segment (Transmit only)
- Bus State change (if any) expected to occur within this segment by the receivers
- Fixed length of 1 Time Quantum

•The first segment within a CAN bit is called the Synchronization Segment and is used to synchronize the various bus nodes.

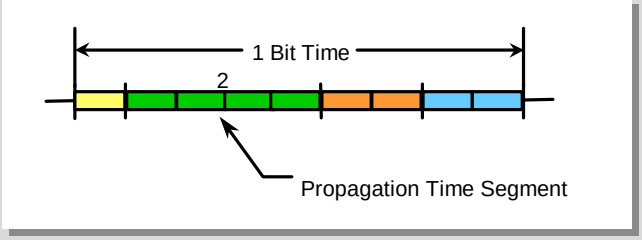
•On transmission, at the start of this segment, the current bit level is output.

•If there is a bit state change between the previous bit and the current bit, then the bus state change is expected to occur within this segment by the receiving nodes.

•The length of this segment is always 1 Time Quantum.

SIEMENS

Propagation Segment



- Allows for signal propagation (across network and through nodes)
- Programmable length (1..8 Time Quanta)

CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 37

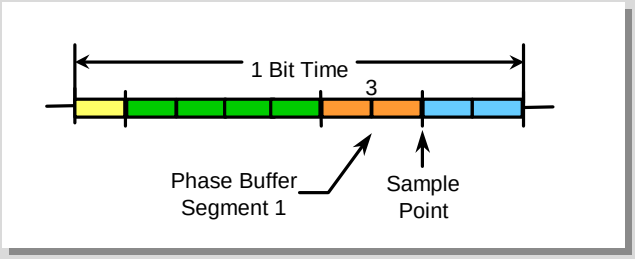
•The Propagation Time Segment is used to compensate for signal delays across the network.

•This is necessary to compensate for signal propagation delays on the bus line and through the electronic interface circuits of the bus nodes.

•This segment may be 1 to 8 Time Quanta long.

SIEMENS

Phase Buffer Segment 1



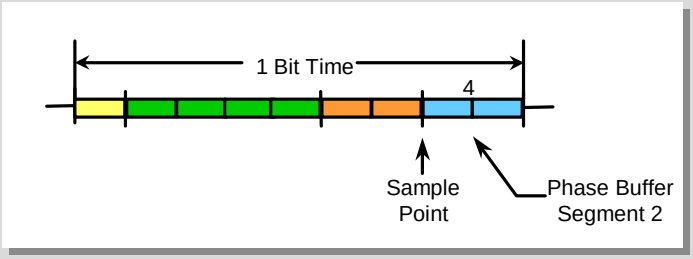
- Allows for lengthening of bit during Re-synchronization
- Bus state is sampled at the end of this segment
- Programmable length (1..8 Time Quanta)

•Phase Buffer Segment 1 is used to compensate for edge phase errors. This segment may be between 1 to 8 Time Quanta long and may be lengthened during resynchronization.

•The sample point is the point of time at which the bus level is read and interpreted as the value of the respective bit. Its location is at the end of Phase Buffer Segment 1 (between the two Phase Buffer Segments).

SIEMENS

Phase Buffer Segment 2



- Allows for shortening of bit during Re-synchronization
- Programmable length (1..8 Time Quanta)

CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 39

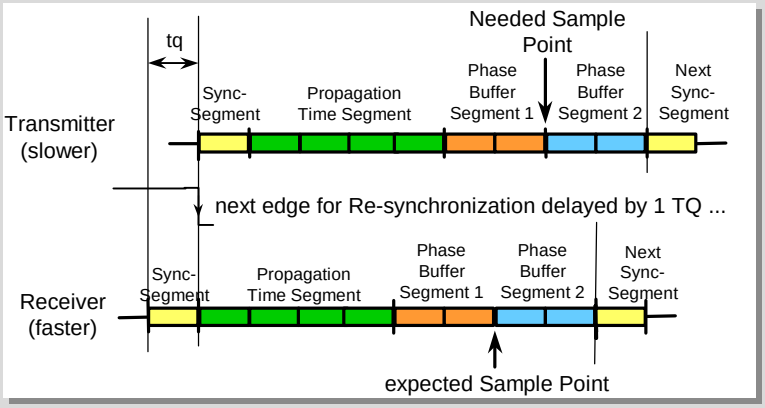
•Phase Buffer Segment 2 is also used to compensate for edge phase errors. This segment may be shortened during resynchronization.

•Phase Buffer Segment 2 may be between 1 to 8 Time Quanta long, but the length has to be at least as long as the information processing time (see below) and may not be more than the length of Phase Buffer Segment 1.

•The information processing time begins with the sample point and is reserved for calculation of the subsequent bit level. It is less than or equal to two Time Quanta long.

SIEMENS

Bit Lengthening



CANPRES Version 2.0

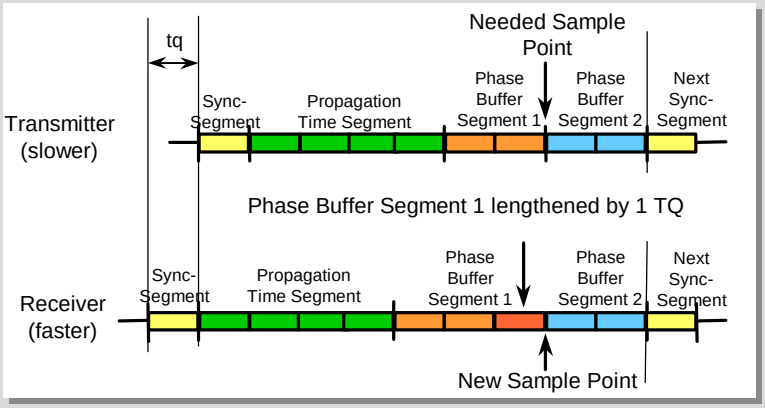
Siemens Microelectronics, Inc.
October 98
Slide 40

•As a result of resynchronization, Phase Buffer Segment 1 may be lengthened or Phase Buffer Segment 2 may be shortened to compensate for oscillator tolerances within the different CAN nodes.

•If, for example, the transmitter oscillator is slower than the receiver oscillator, the next falling edge used for resynchronization may be delayed. So Phase Buffer Segment 1 is lengthened...

SIEMENS

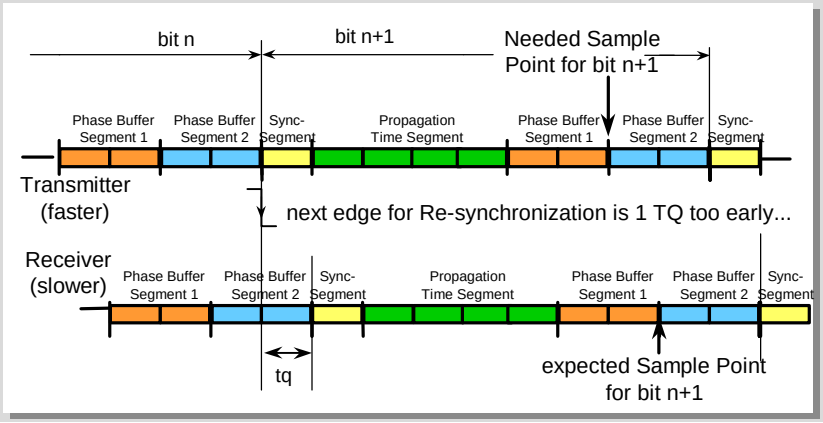
Bit Lengthening (cont.)



•… in order to adjust the sample point and the end of the bit time.

SIEMENS

Bit Shortening



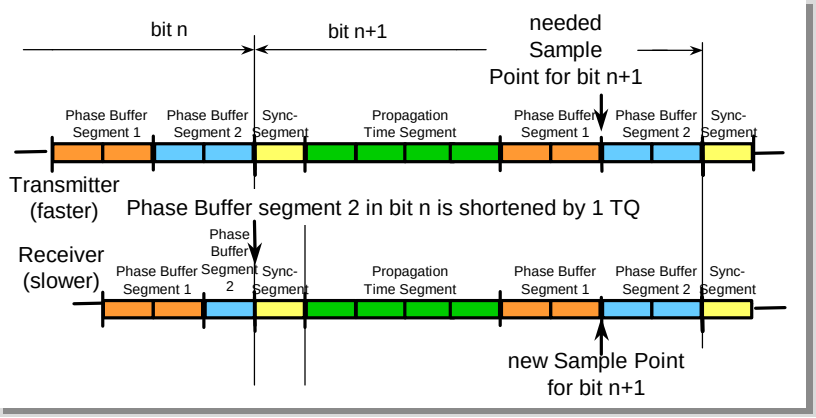
CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 42

•If, on the other hand, the transmitter oscillator is faster than the receiver oscillator, the next falling edge used for resynchronization may be too early. So Phase Buffer Segment 2 in bit N is shortened...

SIEMENS

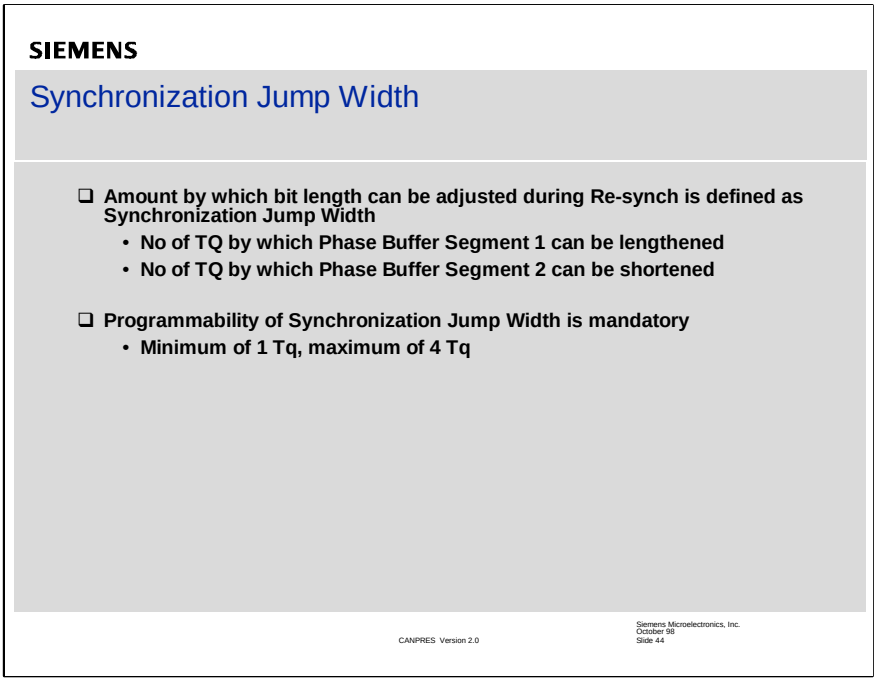
Bit Shortening (cont.)



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 43

•… in order to adjust the sample point for bit N+1 and the end of the bit time



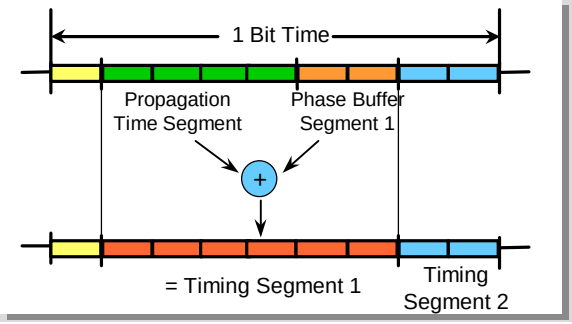
•The limit to the amount of lengthening or shortening of the phase buffer segments is set by the Resynchronization Jump Width.

•The Resynchronization Jump Width may be between 1 and 4 Time Quanta, but it may not be longer than Phase Buffer Segment 2.

SIEMENS

Bit Timing

- **For ease of programming many CAN Modules combine Propagation Time Segment and Phase Buffer Segment 1 (i.e. they only use 3 segments)**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 45

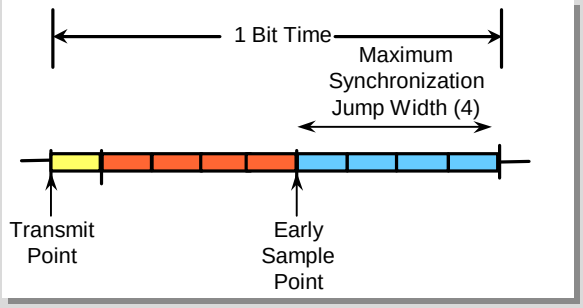
•For many CAN module implementations, the Propagation Time Segment and Phase Buffer Segment 1 are combined, for ease of programming, into one segment often called Timing Segment 1.

•Phase Buffer Segment 2 is then known as Timing Segment 2.

SIEMENS

Why Program the Sample Position ?

- Early sampling decreases the sensitivity to oscillator tolerances
- Lower cost oscillators (e.g. ceramic resonators) can be used



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 46

•Programming of the sample point allows "tuning" of the characteristics to suit the bus.

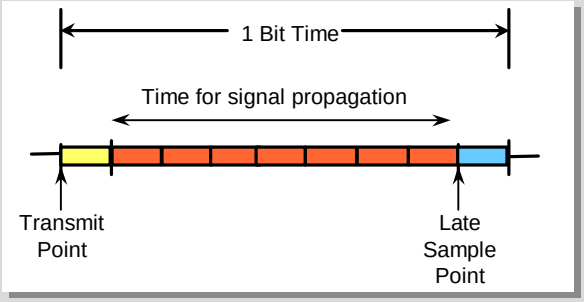
•Early sampling allows more Time Quanta in the Phase Buffer Segment 2 so the Synchronization Jump Width can be programmed to its maximum of 4 Time Quanta.

•This maximum capacity to shorten or lengthen the bit time decreases the sensitivity to node oscillator tolerances, so that lower cost oscillators such as ceramic resonators may be used.

SIEMENS

Why Program the Sample Position ? (cont.)

- ❑ **Late sampling allows maximum signal propagation time**
- ❑ **Maximum bus length / Poor bus topologies can be handled**



CANPRES Version 2.0

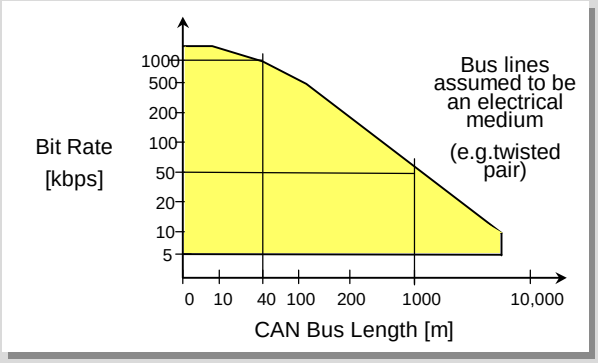
Siemens Microelectronics, Inc.
October 98
Slide 47

•Late sampling allows more Time Quanta in the Propagation Time Segment which allows a poorer bus topology and maximum bus length.

SIEMENS

Relation between Baud Rate and Bus Length

- **Up to 1Mbit / sec @40m bus length (130 feet)**



CANPRES Version 2.0

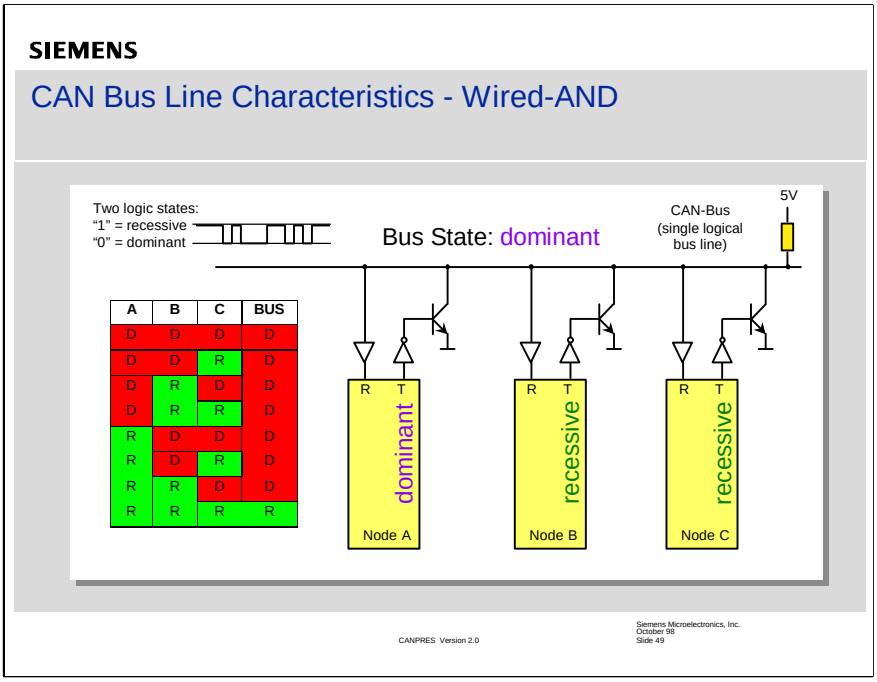
Siemens Microelectronics, Inc.
October 98
Slide 48

•The maximum CAN bus speed is 1 MBaud, which can be achieved with a bus length of up to 40 meters when using a twisted wire pair.

•For bus lengths longer than 40 meters the bus speed must be reduced.

•A 1000 meter bus can still be realised with a 50 KBaud bus speed.

•For a bus length above 1000 meters special drivers should be used.



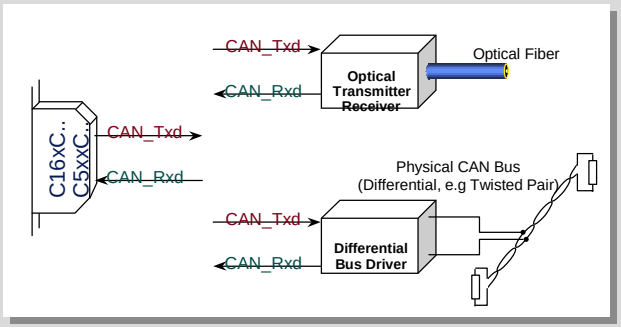
•CAN is serial bus system with one logical bus line. It has an open, linear bus structure with equal bus nodes. The number of nodes on the bus is not restricted by the protocol and may be changed dynamically without disturbing the communication of other nodes. This allows easy connection and disconnection of bus nodes, e.g. for addition of system function, error recovery or bus monitoring.

•The CAN bus line has two logic states: a "recessive" state and a "dominant" state. The actual bus state is "wire-AND" of all node states. This means, that recessive bits (mostly, but not necessarily equivalent to the logic level "1") are overwritten by dominant bits (mostly logic level "0"). As long as no bus node is sending a dominant bit, the bus line is in the recessive state, but a dominant bit from any bus node generates the dominant bus state.

SIEMENS

ISO Physical Layer

- **Usual ISO Physical Layer :-**
  - **Bus wires twisted pair, 120R Termination at each end**
  - **2 wires driven with differential signal (CAN_H, CAN_L)**



CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 50

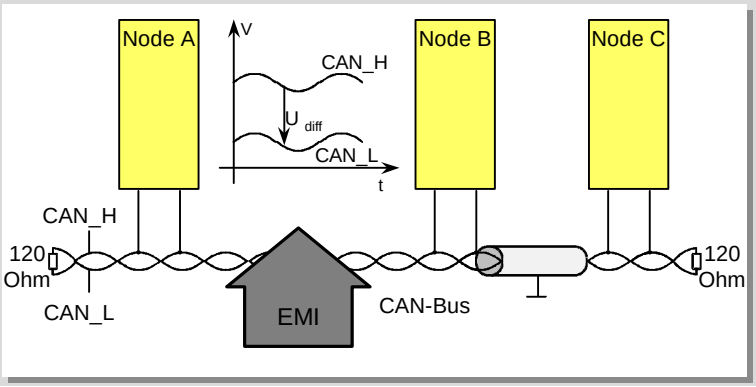
•Therefore, for the CAN bus line, a medium must be chosen that is able to transmit the two possible bit states “dominant” and “recessive”. One of the most common and cheapest ways is to use a twisted wire pair. The bus lines are then called "CAN_H" and "CAN_L". The two bus lines CAN_H and CAN_L are driven by the nodes with a differential signal. The twisted wire pair is terminated by terminating resistors at each end of the bus line, typically 120 Ohms.

•But also an optical medium would be possible for CAN. In this case, the recessive state would be represented by the signal “light off”, the dominant state by the signal “light on”.

SIEMENS

CAN and EMI

- **CAN is insensitive to electromagnetic interference**

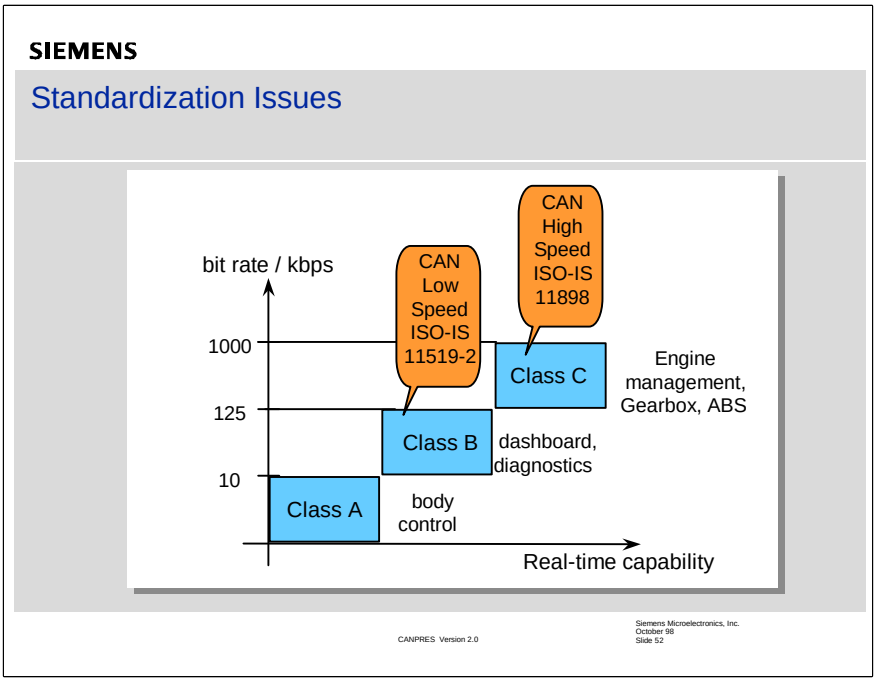


CANPRES Version 2.0

Siemens Microelectronics, Inc.
October 98
Slide 51

•Due to the differential nature of transmission CAN is insensitive to electromagnetic interference, because both bus lines are affected in the same way which leaves the differential signal unaffected.

•To reduce the sensitivity against electromagnetic interference even more, the bus lines can additionally be shielded. This also reduces the electromagnetic emission of the bus itself, especially at high baudrates.



•Vehicle bus system applications can be separated in three different categories according to their real-time capabilities.

•Class A for a low speed bus with bit rates up to 10 kbps, e.g for body control applications,

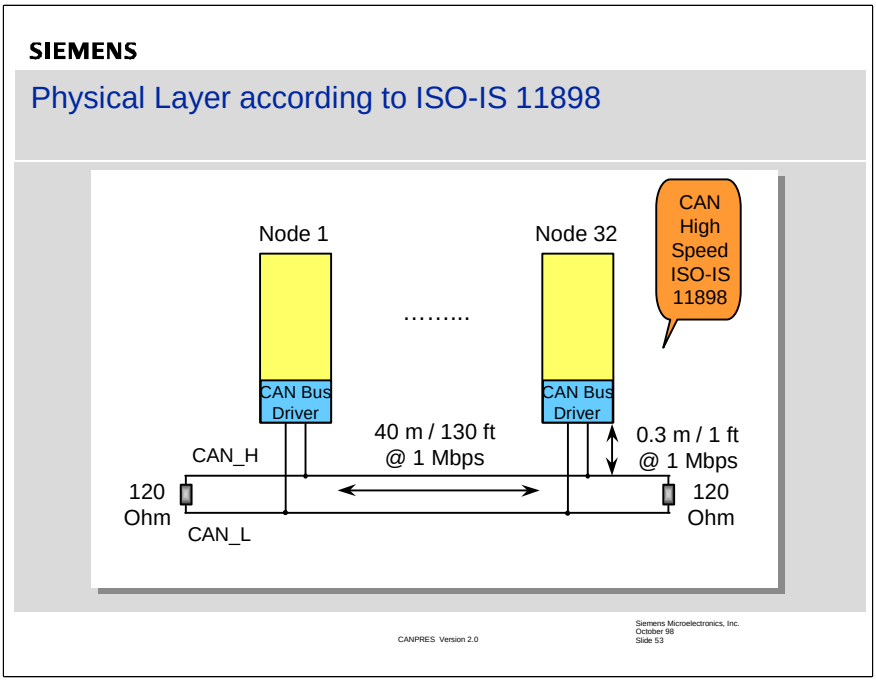
•Class B for a low speed bus with bit rates from 10 kbps to 125 kbps, e.g. for dashboard and diagnostics,

•Class C for a high speed bus with bit rates from 125 kbps to 1 Mbps for real time applications like engine management, Gearbox, ABS etc.

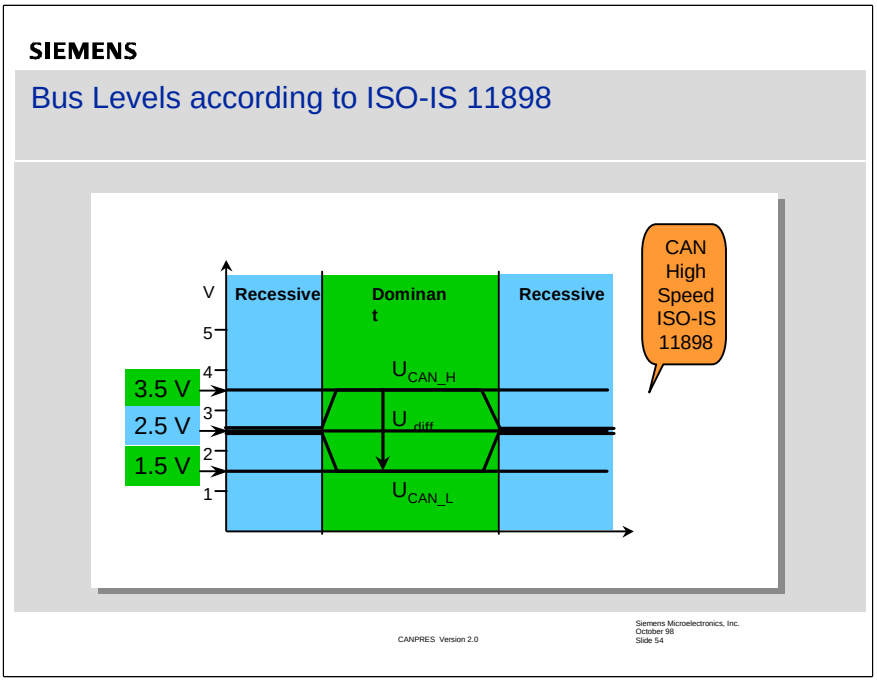
•For the use of CAN in vehicles two standards have been defined for the bus interface:

•CAN High Speed according to ISO-IS 11898 for bit rates between 125 kbps and 1 Mbps

•CAN Low Speed according to ISO-IS 11519-2 for bit rates up to 125 kbps

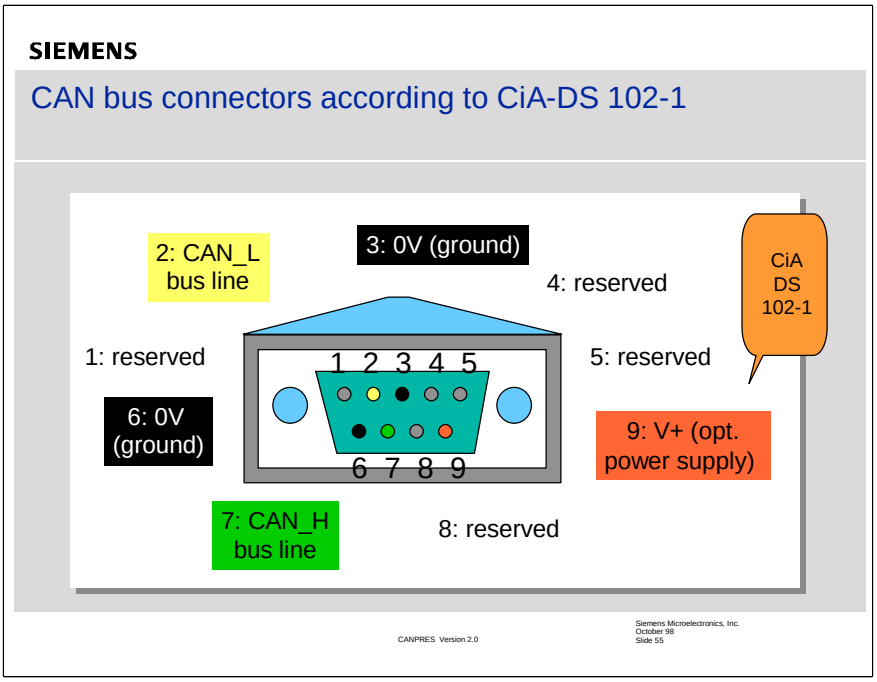


•This is the structure of a Controller Area Network according to ISO-IS 11898. The bus lines may be up to 40 m (130 ft) long at the maximum speed of 1 Mbaud and are terminated by termination resistors of 120 Ohms. The bus lines may be longer when decreasing the baud rate. Up to 30 nodes can be connected with CAN drivers according to this standard. For the connection of more nodes, stronger drivers or repeaters have to be used. To avoid reflexions the connection from the bus lines to the nodes should not exceed 0.3 m (1 ft) at 1 Mbps.



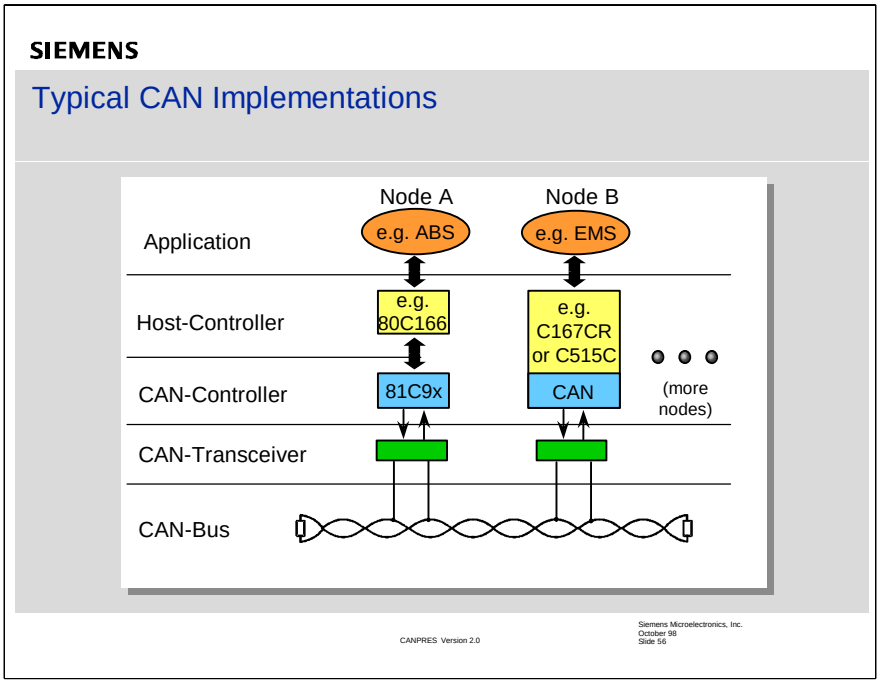
•These are the bus levels according to ISO-IS 11898. A recessive bit is represented by both CAN bus lines driven to a level of about 2.5 V so that the differential voltage between CAN_H and CAN_L is around 0 V.

•A dominant bit is represented by CAN_H going to about 3.5 V and CAN_L going to about 1.5 V. This results in a differential voltage for a dominant bit of about 2V.



•To be able to use CAN as a industrial field bus in an open system the CAN in Automation user's group CiA created a standard called CiA DS 102-1 which is based on the 11898-standard. One important issue in this standard is the proposal to use a 9-pole SUB-D connector for the connection of nodes to the CAN bus lines.

•The bus signals CAN_H and CAN_L are available on pins 7 and 2. The other pins serve as power or ground wires or are reserved for future extensions of the standard.



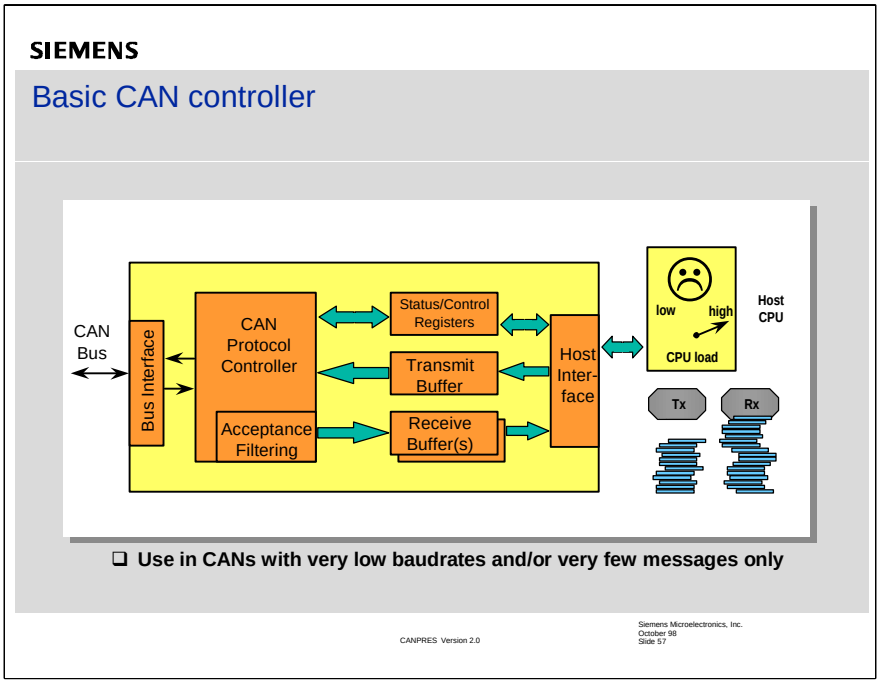
•A typical CAN node used to control a certain application consists of different devices.

•The application itself is controlled by a microcontroller, e.g. the Siemens SAB 80C166.

•To be able to participate in the CAN communication, the microcontroller has to be connected to a CAN protocol controller, e.g. the Siemens stand-alone Full-CAN controller 81C90/91.

•To meet the requirements of e.g. the ISO 11898 CAN standard, a CAN transceiver chip is used to connect the node to the CAN bus lines.

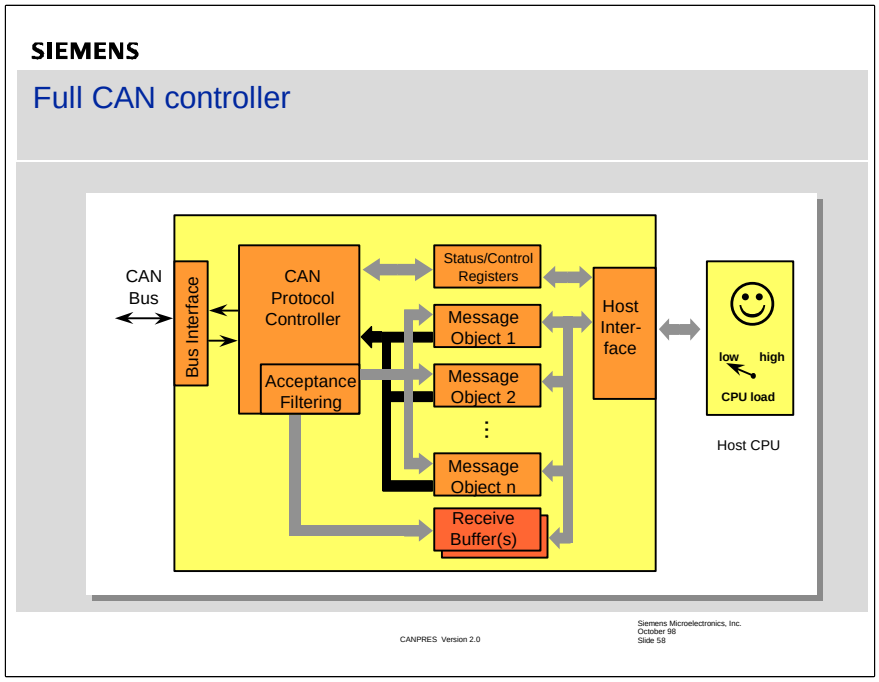
•A more sophisticated way is to use a microcontroller which already has a CAN protocol controller on-chip, e.g. one of the Siemens 8051-compatible 8-bit microcontrollers from the C500 family like the C505C or the C515C. For applications which need higher performance, one of the 16-bit C166-family members with integrated CAN module could be used, e.g. the C164CI or the C167CR. This saves costs as the printed circuit board space is used more efficiently and the user does not have to worry about setting up the communication between the microcontroller and the CAN controller.



•There is one more CAN characteristic concerning the interface between the CAN protocol controller and the host CPU, dividing CAN chips into "Basic-CAN" and "Full-CAN" devices. This has nothing to do with the used protocol Version though, which makes it possible to use both Basic-CAN and Full-CAN devices in the same network.

•In the Basic-CAN devices, only basic functions concerning the filtering and management of CAN messages are implemented in hardware. A Basic-CAN controller typically provides one transmit buffer for outgoing messages and one or two receive buffers for incoming messages. In the receive path, an acceptance filtering is available which allows that only certain CAN identifiers are stored in the receive buffer.

•Because there are only two buffers for the reception of messages the host controller is quite busy reading and storing the incoming messages before they get overwritten by the following ones which results in a quite high CPU load. Also the answering of Remote Frames with the corresponding Data Frame has to be handled by the host controller. Therefore Basic-CAN devices should only be used at low baudrates and low bus loads with only a few different messages.

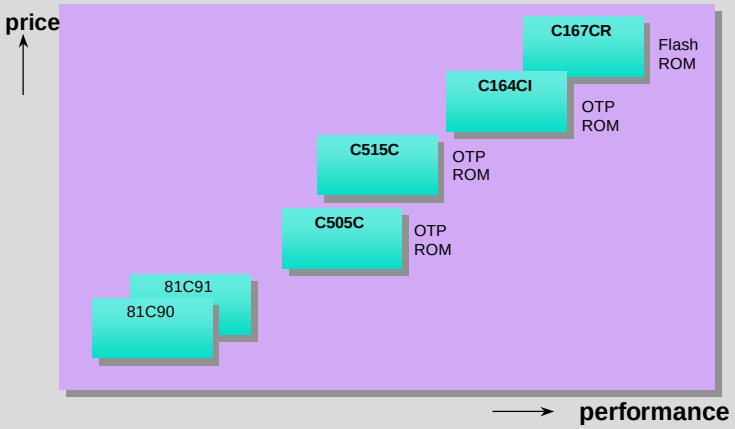


•Full-CAN devices provide the whole hardware for convenient acceptance filtering and message management. For each message to be transmitted or received these devices contain one so called message object in which all information regarding the message (e.g. identifier, data bytes etc.) are stored. During the initialisation of the device, the host CPU defines which messages are to be sent and which are to be received. Only if the CAN controller receives a message whose identifier matches with one of the identifiers of the programmed (receive-) message objects the message is stored and the host CPU is informed by interrupt. Another advantage is that incoming Remote Frames can be answered automatically by the Full-CAN controller with the corresponding Data Frame. In this way, the CPU load is strongly reduced compared to the Basic-CAN solution. Using Full CAN devices, high baudrates and high bus loads with many messages can be handled.

•Many Full-CAN controller provide a "Basic-CAN-Feature": One of their message objects behaves like a Basic-CAN Receive Buffer, i.e. it can be programmed in a way that every message is stored there that does not match with one of the other message objects. This can be very helpful in applications where the number of message objects is not enough to receive all desired messages.

SIEMENS

Complete CAN Capability – Products at Each Price-Performance Point



SIEMENS

User Benefits

- **CAN is low cost**
  - **Serial bus with two wires: good price/performance ratio**
  - **Low cost protocol devices available driven by high volume production in the automotive and industrial markets**
- **CAN is reliable**
  - **Sophisticated error detection and error handling mechanisms results in high reliability transmission**
  - **Example: 500 kbit/s, 25% bus load, 2000 hours per year: One undetected error every 1000 years**
  - **Erroneous messages are detected and repeated**
  - **Every bus node is informed about an error**
  - **High immunity to Electromagnetic Interference**

SIEMENS

User Benefits (cont.)

- CAN means real-time
  - Short message length (0 to 8 data bytes / message)
  - Low latency between transmission request and actual start of transmission
  - Inherent Arbitration on Message Priority (AMP)
  - Multi Master using CSMA/CD + AMP method
- CAN is flexible
  - CAN Nodes can be easily connected / disconnected (i.e. plug & play)
  - Number of nodes not limited by the protocol
- CAN is fast
  - maximum data rate is 1 MBit/s @ 40m bus length (still about 40 kBit/s @ 1000m bus length)

SIEMENS

User Benefits (cont.)

- CAN allows Multi-Master Operation
  - Each CAN node is able to access the bus
  - Bus communication is not disturbed by faulty nodes
  - Faulty nodes self swith-off from bus communication
- CAN means Broadcast Capability
  - Messages can be sent to single/multiple nodes
  - All nodes simultaneously receive common data
- CAN is standardized
  - ISO-DIS 11898 (high speed applications)
  - ISO-DIS 11519-2 (low speed applications)

SIEMENS

Application Examples

- CAN in motor vehicles (cars, trucks, buses)
  - Enables communication between ECUs like engine management system, anti-skid braking, gear control, active suspension ... (power train)
  - Used to control units like dashboard, lighting, air conditioning, windows, central locking, airbag, seat beltsetc. (body control)
- CAN in utility vehicles
  - e.g. construction vehicles, forklifts, tractors etc.
  - CAN used for power train and hydraulic control
- CAN in trains
  - High need of data exchange between the different electronic subsystem control units (i.e. acceleration, braking, door control, error messages etc. as well as diagnostics).

SIEMENS

Application Examples (cont.)

- CAN in industrial automation
  - Excellent way of connecting all kinds of automation equipment (control units, sensors and actuators)
  - Used for initialization, program and parameter up-/download, exchange of rated values / actual values, diagnosisetc.
  - Machine control (printing machines, paper- and textile machines etc.): Connection of the different intelligent subsystems
  - Transport systems
- CAN in medical equipment
  - Computer tomographs, X-ray machines, dentist & wheel chairs
- CAN in building automation
  - Heating, air conditioning, lighting, surveillanceetc.
  - Elevator and escalator control

SIEMENS

Application Examples (cont.)

- CAN in household appliances
  - Dishwashers, washing machines, even coffee machines...
- CAN in office automation
  - photocopier - interface to document handler, paper feeding systems, sorter
  - communicates status, allows on-site / field connection or "hot swapping"
  - DocuText Systems, i.e. automatic print, sort and bind on demand

SIEMENS

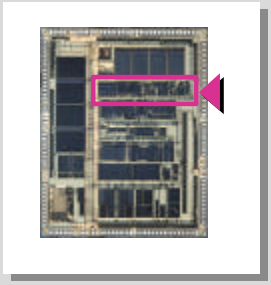
Siemens CAN Modules

- **Two different CAN modules can be found in the Module Library of Siemens Microcontrollers:**
  - 1) Full-CAN V2.0 Part B passive
    - Developed by Siemens
    - Offered as stand-alone SAE 81C90/91 (Peripheral)
  - 2) Full-CAN V2.0 Part B active
    - developed by Bosch
    - Integrated in the 8-Bit µC C505C and C515C
    - Integrated in the 16-Bit µC C167CR and C164CI

SIEMENS

Features of the CAN Module on C167CR / C164CI / C515C / C505C

- **Functionality corresponds to AN 82527**
- **Complies with CAN spec V2.0B active (Standard- und Extended-CAN)**
- **Maximum CAN transfer rate (1 MBit/s)**
- **Full CAN Device**
  - 15 message objects with their own identifier and their own status - and control bits
  - Each message object can be defined as transmit - or receive object



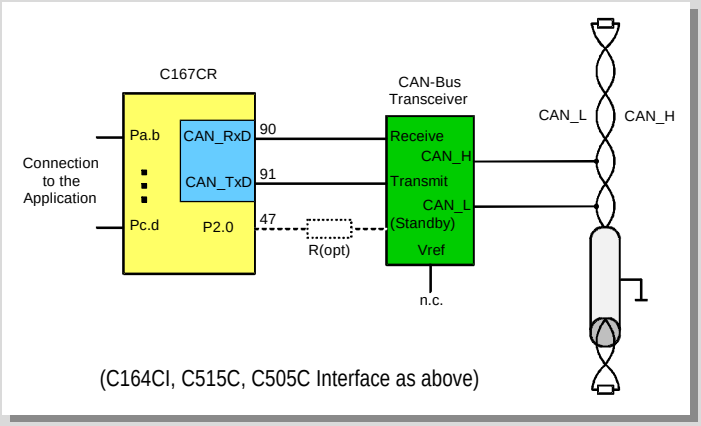
SIEMENS

Features of the CAN Module on C167CR / C164CI / C515C / C505C (cont.)

- **Programmable Mask Registers for Acceptance Filtering**
  - Global mask for incoming messages (Full-CAN-Objects)
  - Additional mask for message object 15 (Basic-CAN-Feature)
- **Basic CAN Feature (Message Object 15)**
  - Equipped with two receive buffers
  - Dedicated global mask register for acceptance filtering
- **Connection to the Host CPU (C166-Core)**
  - Module access via chip-internal XBUS (16-bitdemultiplexed mode)
  - Interrupt connection to the CPU; Flexible interrupt event control
- **To connect to CAN only a CAN transceiver is needed**

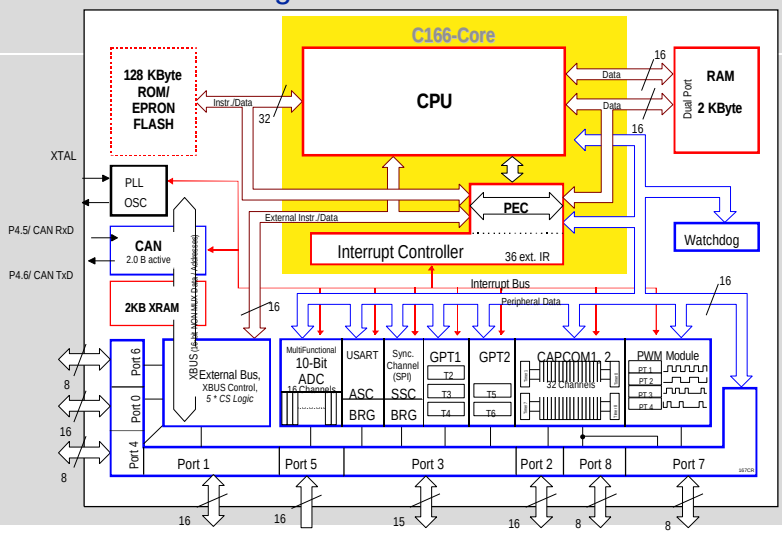
SIEMENS

Connecting the C167CR to CAN



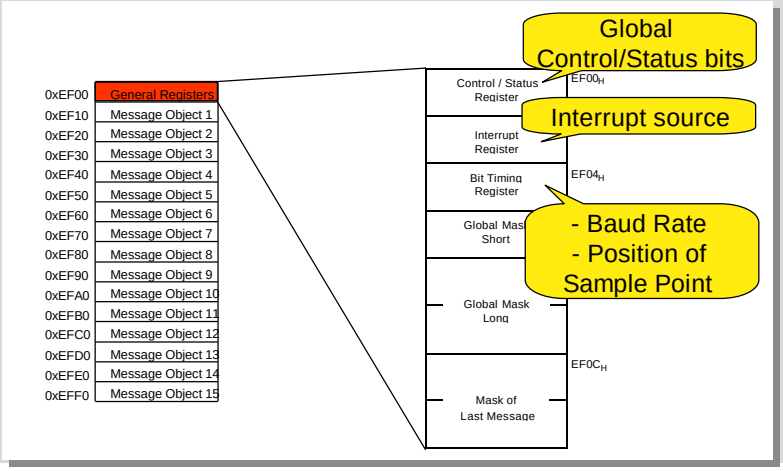
SIEMENS

C167CR Block Diagram



SIEMENS

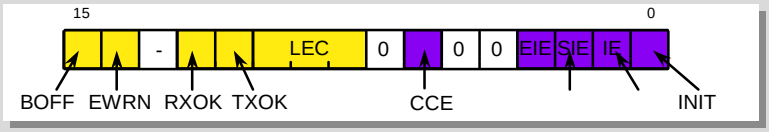
C167CR Full CAN Module - General Registers I



SIEMENS

C167CR - Control / Status Register

- **Contains Global Status and Control bits**

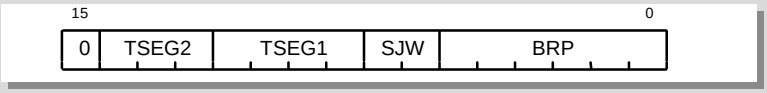


- **Setting INIT starts initialization (set CCE to change bit timing)**
- **IE, SIE, EIE: Interrupt Control| (Global Interrupt Enable, Status Interrupts, Error Interrupts)**
- **LEC contains error code of last error that occurred on the bus**
- **TXOK / RXOK are set on successful frame Transmission / Reception**
- **EWRN indicates Error Counter level of 96, BOFF indicates Bus Off**

SIEMENS

C167CR - Definition of Bit Characteristics

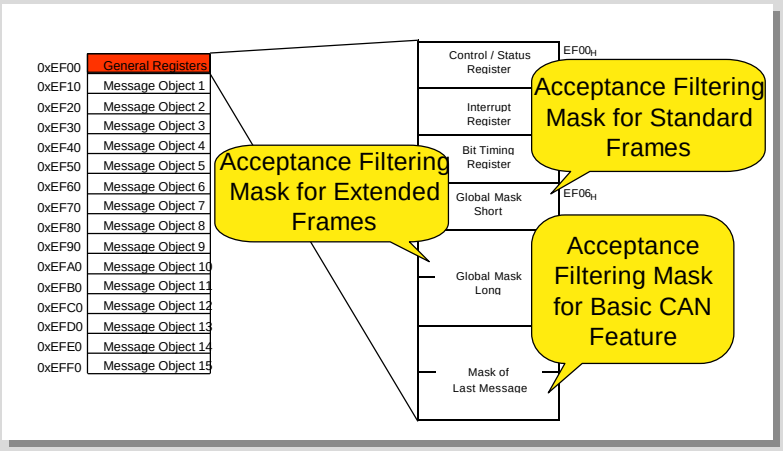
- **Characteristics are defined via Bit Fields in the Bit Timing Register :-**



  - TSEG2 sets the number of Time Quanta in Tseg2
  - TSEG1 sets the number of Time Quanta in Tseg1
  - SJW sets the maximum allowed Synchronization Jump Width
  - BRP sets the divide ratio to generate Tq from the CPU Clock
- Note INIT and CCE bits (in Control / Status Register) must be set to allow write access to the Bit Timing Register

SIEMENS

C167CR Full CAN Module - General Registers II



SIEMENS

C167CR - Global Mask Registers

- **CAN Module allows Rx ID bits to be selected as "don't care"**
  - Allows receipt of a group of messages into 1 Message Object
- **Mask Register defines which ID bits are significant**
- **On match, Data and ID Bits are transferred into Message Object**

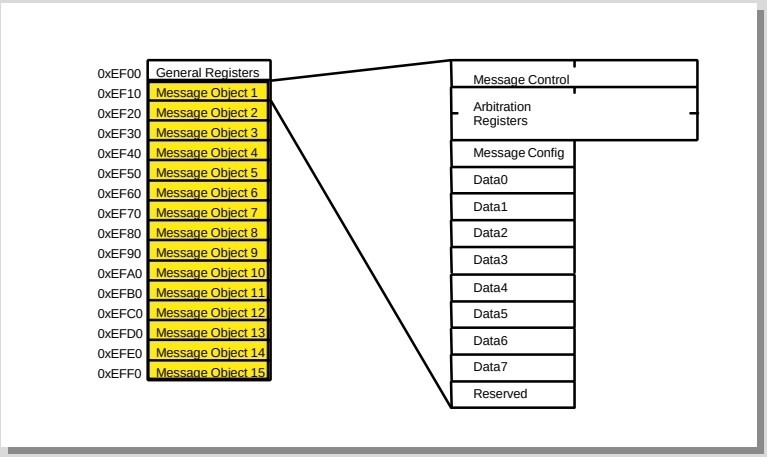
SIEMENS

C167CR - Global Mask Registers (cont.)

	10									0	
Mask Register (std ID)	1	1	1	1	1	1	1	0	1	1	0	(= 0x7f6)
Message Object Arbitration Register	1	0	0	1	0	0	1	0	0	0	0	(= 0x490)
Resulting ID matches (X = don't care)	1	0	0	1	0	0	1	X	0	0	X	
ID's received:	1	0	0	1	0	0	1	0	0	0	0	(= 0x490)
	1	0	0	1	0	0	1	0	0	0	1	(= 0x491)
	1	0	0	1	0	0	1	1	0	0	0	(= 0x498)
	1	0	0	1	0	0	1	1	0	0	1	(= 0x499)

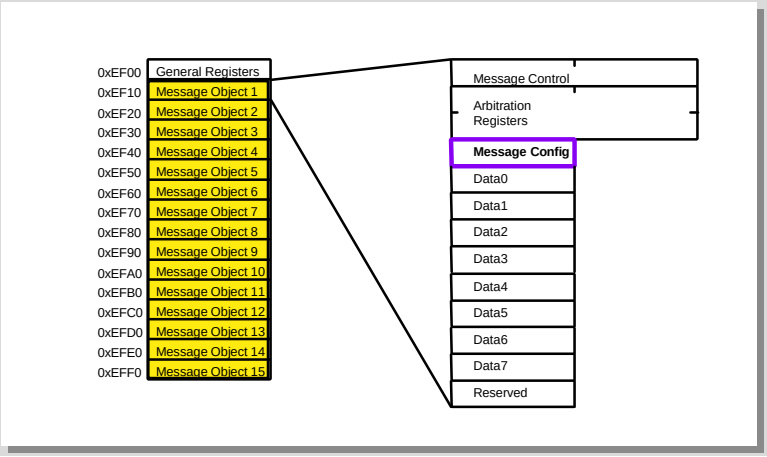
SIEMENS

C167CR Full CAN Module - Registers



SIEMENS

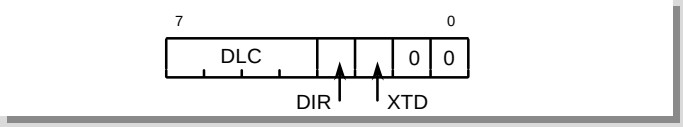
C167CR Full CAN Module - Message Configuration Register



SIEMENS

C167CR - Message Configuration Register

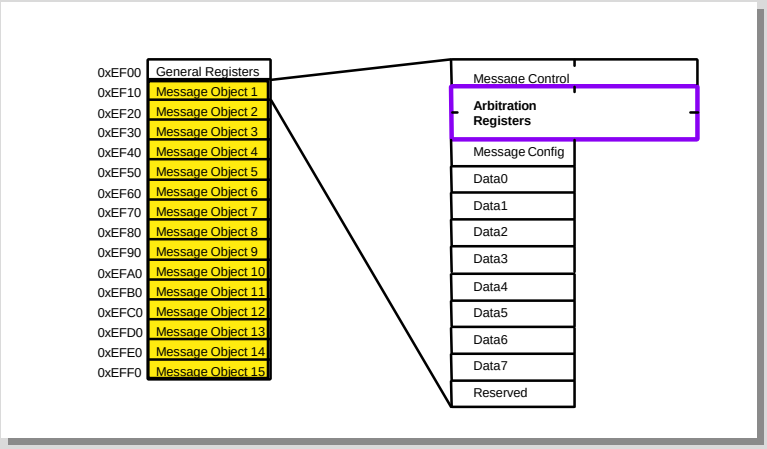
- **Contains important information about the Message Object**



- XTD selects 11 bit (XTD=0) or 29 bit (XTD=1) Identifier
- DIR selects direction of Message Object:
  - 0 = Receive Data Frames with matching ID (generate Remote Frames on Transmit Request)
  - 1 = Transmit Data Frames on Transmit Request (receive Remote Frames with matching ID and answer with corresponding Data Frame)
- DLC configures number of Data Bytes in Message Object

SIEMENS

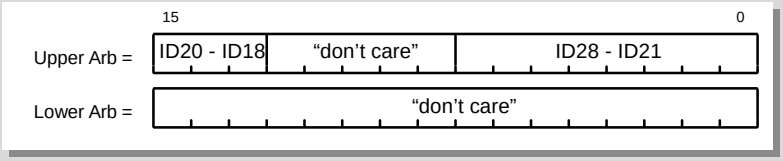
C167CR Full CAN Module - Arbitration Registers



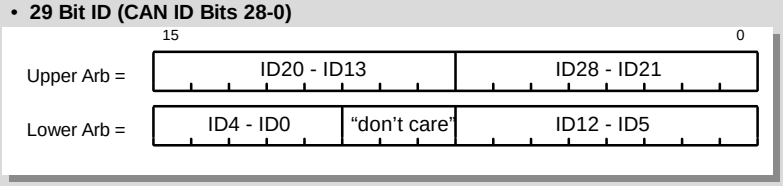
SIEMENS

C167CR - Message Arbitration Registers

- **2 * 16 bit registers (Upper and Lower)**
  - **11 Bit ID (CAN ID Bits 28-18)**

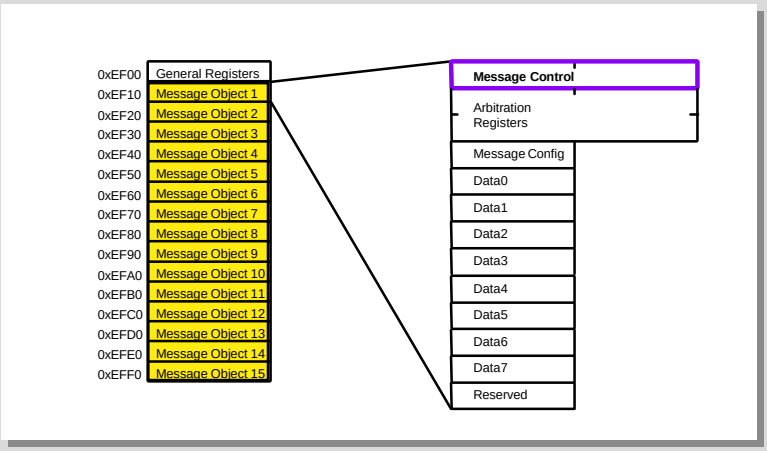


  - **29 Bit ID (CAN ID Bits 28-0)**



SIEMENS

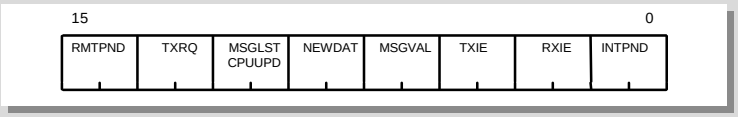
C167CR Full CAN Module - Message Control Register



SIEMENS

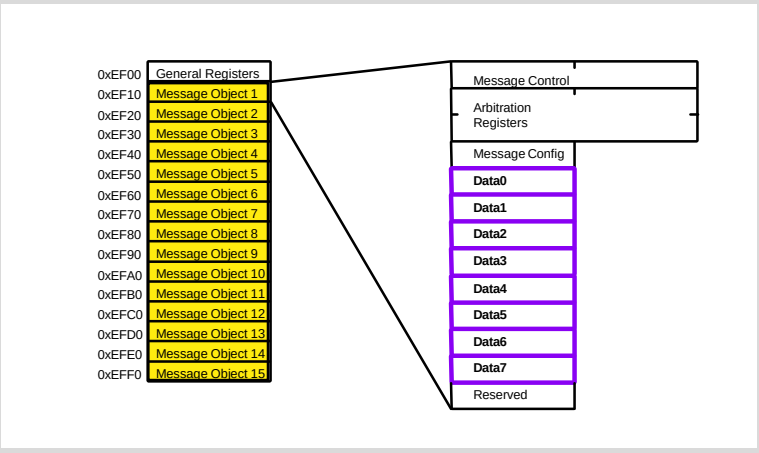
C167CR - Message Control Register

- 16 Bit Register, 2 (register) bits per Status Bit
  - Any combination can be tested in a single And/Or operation
    - Read 01 = Element is clear
    - Read 10 = Element is set
  - Any bit change can be implemented in a single write
    - Write 11 = Leave element unchanged
    - Write 01 = Clear element
    - Write 10 = Set element



SIEMENS

C167CR Full CAN Module - Data Registers



SIEMENS

C167CR - Message Data Registers

- 8 off Byte Wide Registers
  - For Rx object data from message is stored in these registers
  - For Tx object data to be transmitted is taken from these registers
- Registers are used in numerical order
  - If message frame contains 1 byte then Data0 is used
  - If message frame contains 4 bytes then Data0 to Data3 are used