Advanced CAN Bit Timing
Version 1.0

Application Note AN-AND-2-135

Author(s) Bruce Emaus
Restrictions Restricted Membership Document
Abstract This Application Note provides a detailed discussion of CAN Bit Timing for the application engineer.

Table of Contents

Contact Information: www.vector-cantech.com or 1-248-449-9290

1 Overview

CAUTION – This heavy-industrial strength application note based on the contributions of Karl Overberg and Klaus Dietmayer drops the reader directly into the atomic structure of CAN bit timing. The contents are not recommended for the typical CAN developer. Do not operate heavy machinery while reading. This material is presented for potential use in non-automotive system-level applications as well as academic purposes.

For basic CAN experimentation with short wiring, almost any set of bit timing values (that follow the rules) will usually work, however, if the goal is “not to be just adequate but to aim at near perfect”, then, the systems architect responsible for the CAN-based physical layer will find it necessary to journey deep into the mountainous terrain of CAN bit timing in order to build a robust distributed embedded system with “optimized bit timing parameters”.

This application note covers the complex relationships between CAN bit timing parameters, CAN controller clock accuracy, physical layer propagation delay, and the optimization of a system’s CAN bit timing parameters. A companion spreadsheet is available to help in the analysis. See section Additional Resources.

1.1 Background

The CAN protocol is asynchronous and nodes synchronize to each other’s messages at the Start-of-Frame (SOF) bit and then adjust synchronization on subsequent bit edges throughout the remainder of the message. As each node uses its own clock reference, the ability to synchronize is dependent on the maximum allowable oscillator drift between any two nodes in the system. Additionally, the use of bit-wise arbitration and in-frame acknowledgment requires that the propagation delay between network nodes must be limited to guarantee synchronization.

For a given CAN network, the maximum allowable oscillator drift and network propagation delay are not independent parameters and both influence the maximum allowed bus length. Both parameters must be optimized together in order to guarantee proper synchronization and message transfer over all expected normal communication and error conditions.

2 CAN Bit Timing Elements

Elements of the CAN bit time are defined by using the time segment names found in the Bosch CAN 2.0 Specification or the by using the typical time segment names implemented by the silicon suppliers as shown in Figure 1. Because the silicon implementation names are somewhat easier to use, this application note is based on this naming convention.

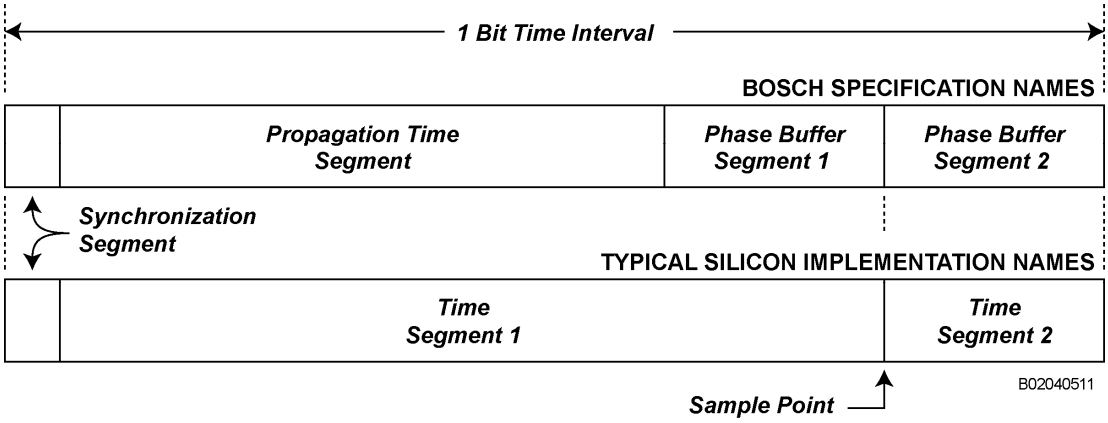

Figure 1 – CAN Bit Time Segment Names

Elements of the CAN bit time interval are shown in Figure 2.

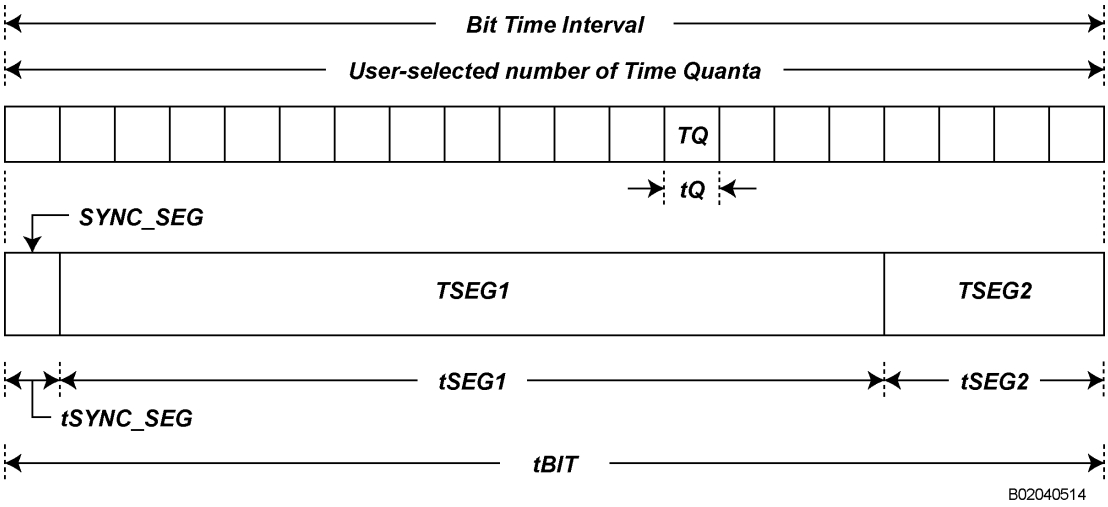

Figure 2 – CAN Bit Timing Elements

2.1 Time Quantum or TQ

The reference for all implemented CAN bit timing is the Time Quantum (TQ). The time duration t_Q of the Time Quantum is derived from the CAN controller clock oscillator and the adjustable clock divider (prescaler). The Bit Time Interval, tBIT, is always set to a fixed, user-selected number of Time Quanta (N*TQ).

2.2 Bit Time Interval

With time duration of t_{BIT}, the bit time interval consists of three time segments each specified as an exact integral number of TQ -

- Synchronization Segment or SYNC_SEG
- Time Segment 1 or TSEG1
- Time Segment 2 or TSEG2

2.3 Synchronization Segment or SYNC_SEG

The Synchronization Segment or SYNC_SEG time interval with time t_{SYNC_SEG} is used to synchronize all the nodes across the network. If all nodes are fully synchronized, then all bit edges will occur during this interval. With time duration of t_{SEG1}, the SYNC_SEG time interval has a fixed period of one TQ.

2.4 Time Segment 1 or TSEG1

With time duration of t_{SEG1}, Time Segment 1 or TSEG1 is the time interval used to compensate for both positive phase errors in synchronization between nodes on the network and propagation delay between network nodes. If a synchronization edge occurs during this interval, TSEG1 is lengthened to compensate for synchronization differences with other nodes on the CAN network.

2.5 Time Segment 2 or TSEG2

With time duration of t_{SEG2}, Time Segment 2 or TSEG2 is the time interval used to compensate for negative phase errors in synchronization between nodes. If a synchronization edge occurs during this interval, TSEG2 is shortened to compensate for synchronization differences with other nodes on the CAN network.

2.6 Re-synchronization Jump Width or SJW

The Re-synchronization Jump Width or SJW is an additional time segment, which is not directly a segment of the bit time but is used to dynamically adjust TSEG1 or TSEG2. With time duration of t_{SJW}, the Re-synchronization Jump Width is the maximum amount of time by which TSEG1 may be lengthened or TSEG2 shortened to compensate for synchronization differences between nodes on the CAN network. The SJW value is dependent upon the synchronization time differences between network nodes.

2.7 Sample Mode

During the bit time interval, each node will sample (read) the bit to determine the bit state.

As shown in Figure 3, most CAN controller implementations provide a choice between two sampling modes –

- Single sample mode
- 2 of 3 sample mode

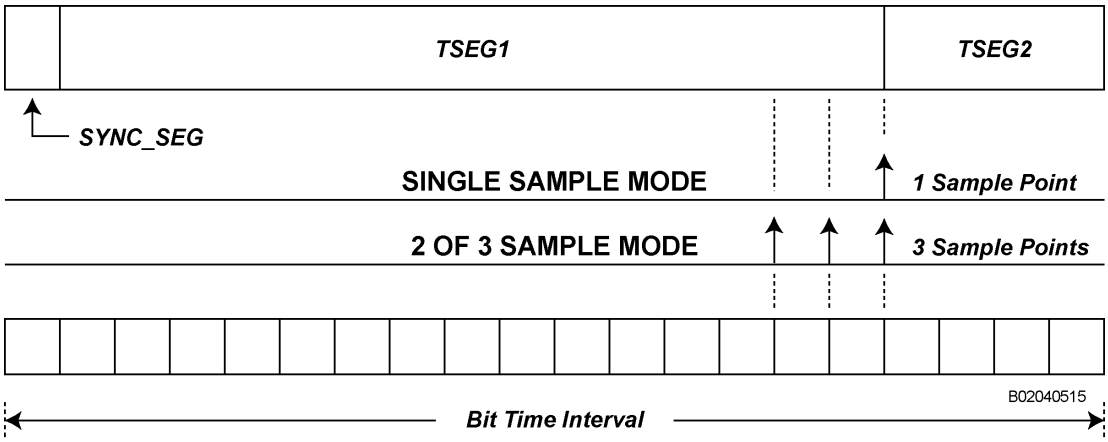

Figure 3 – Two Choices for CAN Bit Sampling

For the single sample mode the receive bit is read once at the end of the TSEG1 interval. In the three sample mode, two of three voting logic is used to determine the bit value with each sample spaced one TQ apart and just prior to the end of the TSEG1 interval. Note – some CAN controllers do not support the 2 of 3 sample mode.

As this application note travels deeper into the differences between the two sampling modes, it is appropriate to mention at this point that most high speed networks use the single sample mode because it allows the system to handle higher levels of system propagation delay.

2.8 Bit Timing Equations

The time duration of the CAN bit, t_{BIT}, is equal to the sum of the SYNC_SEG, TSEG1 and TSEG2 time intervals as follows -

$$1)\ t_{BIT} = t_{SYNC\_SEG} + t_{SEG1} + t_{SEG2}$$

The nominal bit time can also be defined in terms of the TIME_QUANTUM, t_Q, and the integer number of TIME_QUANTA in a given bit period, N_Q, as follows -

$$2)\ t_{BIT} = N_Q\, t_Q$$

2.8.1 Bit Timing Example

For example, beginning with a 20 MHz clock oscillator and a required CAN bit rate of 500K BPS (Bits Per Second), a divide by 2 value for the CAN clock divider will need 20 Time Quanta to establish the 2 micro-second bit time interval.

Oscillator	= 20 MHz
Clock Divider	= 2
t_Q	= 100 nsec
Bit Rate	= 500K BPS
t_{BIT}	= 2 microseconds
Number of Required TQs (N_Q)	= 2 microseconds / 100 nsec
N_Q	= 20

To continue, we could arbitrarily set the values of Time Segment 1 and Time Segment 2. This is accomplished by using the following relationship –

Number of Required TQs	= Synchronization Segment + Time Segment 1 + Time Segment 2
N_Q	= 1 + Time Segment 1 + Time Segment 2

What values should be selected for Time Segment 1 and Time Segment 2 are based on the desired location of the “sample point”. We could choose a sample point at 80 percent by using the following segment values -

Time Segment 1	= 15
Time Segment 2	= 4
Sample Point	= (1 + Time Segment 1) / (1 + Time Segment 1 + Time Segment 2)
	= (1 + 15) / (1 + 15 + 4)
	= 0.80 = 80%

While this example is adequate for basic CAN experimentation, the optimal bit timing parameters require considerable more analysis.

3 Bit Timing Characteristics

Several key system-level and module-level characteristics influence bit timing including –

- CAN Reference Timing Tolerance
- Non-integer Number of Time Quanta
- Clock Skew Between Nodes
- System Propagation Delay
- CAN Synchronization Process
- Re-synchronization Jump Width
- Sample Mode

3.1 CAN Reference Timing Tolerance

With rare exception, each CAN node across the network uses its own local timing reference to establish bit timing. Each typical implementation uses a different clock oscillator frequency and is based on quartz crystals, ceramic resonators, or other electronic means.

From the systems point of view, all CAN clocks -

- Are asynchronous
- Are not exact
- Will drift in time
- Will drift over temperature

For example, quartz crystals, ceramic resonators and Phase Locked Loop derived oscillator references have typical drift tolerances of 0.1%, 1.5% and 0.5%, respectively.

It is important to understand that at the systems level the CAN bit timing reference is effectively distributed. Communications is only possible because all receiving CAN nodes synchronize to the transmitter's "recessive-to-dominant" bit edges.

Because the differences in oscillator frequency between nodes may prevent communications, the bit time analysis must take into account the case that two different network nodes may have local timing references that are at opposite extremes of their tolerance limits.

3.2 Non-Integer Number of Time Quanta

The CAN controller oscillator frequency and the selected clock divider (prescaler) must be able to resolve the nominal network bit period into an exact integer number of Time Quanta. If not, then an effective oscillator tolerance must be computed for that node and used for subsequent analysis.

The effective oscillator tolerance is defined in terms of the oscillator tolerance, Δf, and the deviation of a given node's programmed bit time from the network bit time as follows:

$$3) \quad \Delta f_{effective(node)} = \Delta f_{oscillator(node)} + \left| \frac{t_{BIT(network)} - N_{Q(node)} t_{Q(node)}}{t_{BIT(network)}} \right|$$

For example, let's assume we need to use a 18.432 MHz crystal because of a serial 9600 baud UART-based communication requirement in addition to CAN at 500K BPS.

Using worksheet prescaler_calculator from the Vector Excel Spreadsheet SP-AND-2-135 "Advanced CAN Bit Timing Calculator" -

Prescaler Calculator

Target Bit Timing Values		value	units	notes
Bit Rate (Baud Rate)	=	500	K BPS	**Enter** the desired bit rate in K Bits Per Second. For 250,000 BPS, enter 250.
Bit Period	=	2.00E-06	seconds	The bit period is computed.
Allowable CAN Controller Bit Timing Values				These values are specific to the selected CAN controller. NOTE - this example used values from the Philips SJA1000
Baud Rate Prescaler value (min)	=	1		Typical for most CAN Controller silicon
Baud Rate Prescaler value (max)	=	64		Typical for most CAN Controller silicon
Maximum number of Time Quanta	=	25		This is the maximum value of subdivisions that can be selected. [chip limit]
Minimum number of Time Quanta	=	4		Enter the smallest number of subdivisions which may be selected.
Show only matches to desired bit rate in the Bit Rate Table (yes, no)	=	no	yes, no	Enter **yes** to see only available choices for the bit rate and oscillator. Enter **no** to see all resulting bit rates Enter yes or no in lower case with no embedded blanks
Prescaler Calculator		**value**	**units**	**notes**
CAN Controller Oscillator Frequency	=	18432000	Hz	**Enter** the oscillator frequency that is used for the CAN controller
CAN Controller Prescaler Value	=	2		**Enter** the clock divider or bit rate prescaler value for the CAN controller. Use the Bit Rate Table to the lower right for additional choices.
Divisor Frequency	=	9216000	Hz	
tQ	=	1.08507E-07	seconds	
Number of Time Quanta	=	18.43200		This value must be an integer - if not, then compute the Node Additional Offset Tolerance
Integer Number of Time Quanta ?	=	NO		
Selected Number of Time Quanta	=	18		
Computed Bit Period	=	1.95313E-06	seconds	
Node Additional Offset Tolerance	=	0.02344		
Node Additional Offset Tolerance	=	2.34375	%	in percent

Table 1 – CAN Controller Clock Prescaler Calculator

In this example, the effective oscillator tolerance must be increased by 2.34 percent and, without getting farther into the details, this node using a 18.432 MHz crystal will not provide CAN communications at 500K BPS.

In most cases at high-speed bit rates, a deviation from the nominal network bit period, due to the inability of a given node to resolve the network bit period into an integer number of TQ, will render it incapable of communicating with other nodes.

3.3 Time Quantum Skew

An additional effect, which must be considered, is the Time Quantum Skew between network nodes.

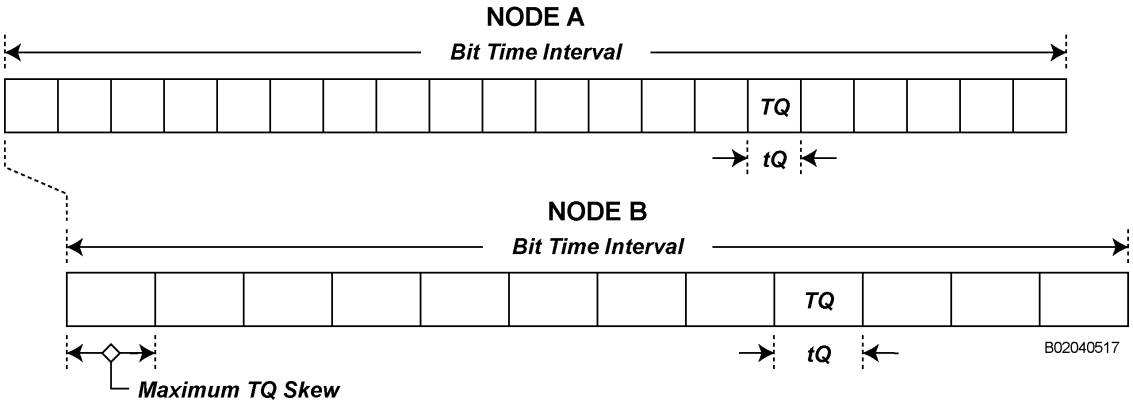

Figure 4 – Time Quantum Skew

Across all of the network membership the various implemented Time Quantum values do not use the same time period. While two nodes might momentarily be closely synchronized such that the Time Quantum skew could be zero, during other times the maximum Time Quantum Skew is near the time duration of that system node which has the longest Time Quantum duration as shown in Figure 4.

Even when all nodes use identical oscillator frequencies, the fact that individual node's timing references are asynchronous will result in up to one TQ of bit clock skew in bit timing between any two nodes.

The skew is not a fixed value – the skew oscillates between zero and a maximum value similar to the beat frequencies encountered when listening to two near frequency tones.

3.4 System Propagation Delay

Contrary to Ethernet and other CSMA-type protocols that do not use in-frame response, propagation delay in a CAN system is a very important characteristic. This is because the CAN protocol utilizes two in-message processes during designated bit time intervals in which multiple nodes may participate.

- These bit-level multiple node participation processes are –
- the non-destructive bit-wise arbitration process during the arbitration field
- the acknowledgment process that occurs during the acknowledgment field

During the arbitration process, multiple transmitters may simultaneously drive their identifier bits onto the bus attempting to win arbitration (gain complete access to the bus). Each contending transmitter must be able to read the state of the bus in order to know if its transmitted bus state has won arbitration or whether another transmitter has won arbitration. Once a transmitter knows that it has lost arbitration, it will stop any subsequent bit transfers and re-attempt to send the message after the current transmission is finished.

During the acknowledgment process, all receivers may simultaneously drive the Ack Slot dominant for a single bit period to indicate to the transmitter their acceptance of the message.

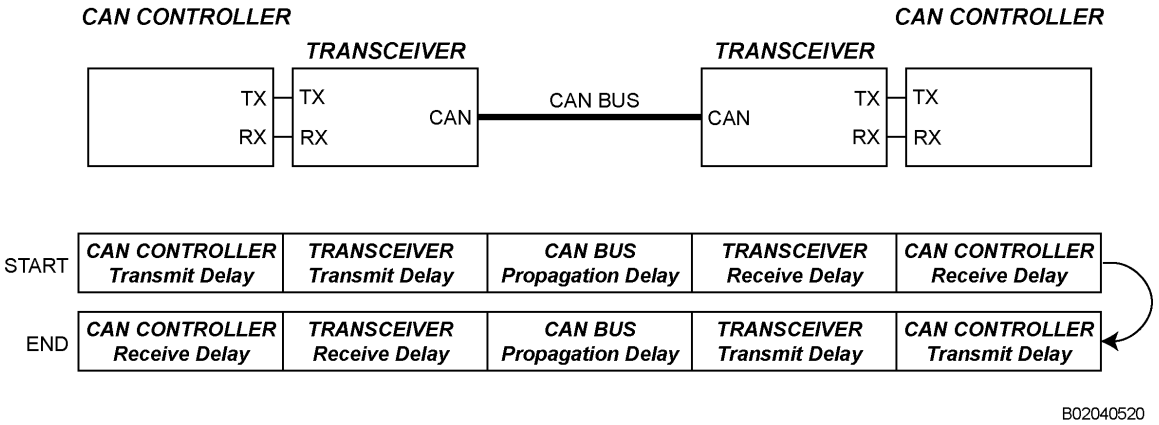

Figure 5 – Basic Components of System Propagation Delay

It is the propagation delay of the CAN network limit the maximum network bus length for a given bit rate.

In Figure 5, a simple model of a two-node CAN bus system shows the typical components of propagation delay.

Assuming that each node in a given network has the same propagation delay components, the total round trip system propagation delay, defined as t_{PROP}, can be defined mathematically in terms of the typical propagation delay components t_{TX}, t_{RX}, t_{LOGIC} and t_{BUS} as follows -

$$4) \quad t_{PROP} = 2(t_{TX} + t_{RX} + t_{LOGIC} + t_{BUS})$$

The parameter t_{TX} represents the transceiver transmit delay, which is the time from the TX logic output of the CAN controller to the signal driven onto the bus by the CAN physical layer transceiver, and the parameter $t_{RX,}$ represents the transceiver receive delay from the signal on the bus through the receive comparator circuit to the logical RX signal at the input to the CAN controller.

The parameter t_{LOGIC} represents the combined transmit and receive delays through the CAN controller's logic and the potential jitter associated with synchronizing the received asynchronous signals.

The parameter t_{BUS} represents the time for an electrical signal to travel from one end of the bus to the other. For example, the signal propagation delay for an unshielded, twisted wire pair is approximately 5nS/Meter; for a 40 meter bus, this implies a propagation delay in one direction of approximately 200nS.

If additional physical layer interface components, such as

- opto-couplers
- common mode chokes
- filter devices
- or repeaters

are used, the propagation delays of each device must be added to the composite propagation delay.

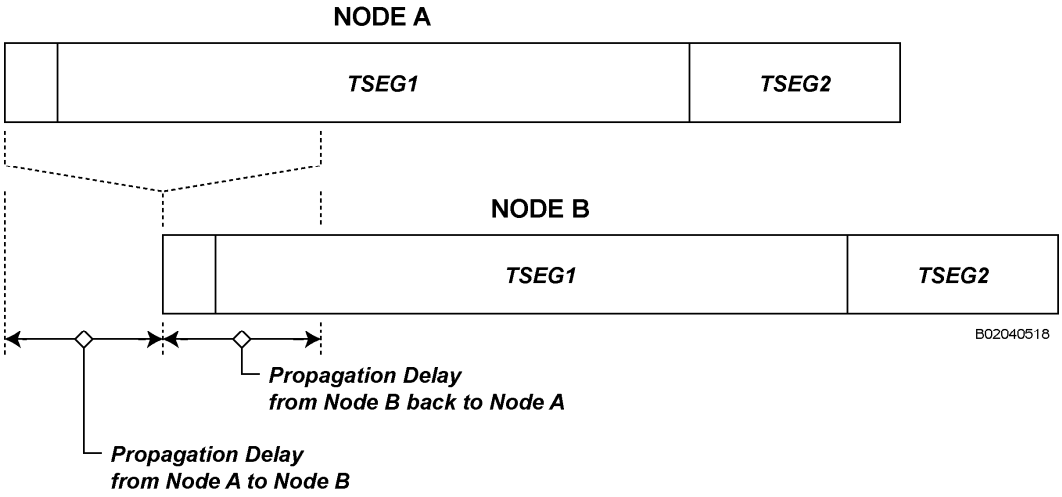

Figure 6 – CAN Bus Propagation Delay Time

The relationship between system propagation delay and CAN bus timing is shown in Figure 6 and illustrates the doubling effect which arbitration has on propagation delay.

A synchronization mechanism is defined in the CAN protocol to help compensate for oscillator reference differences and propagation delays between network nodes.

3.5 CAN Synchronization Process

After the idle bus condition is detected (at least 11 bit times of recessive), CAN protocol synchronization begins with first “recessive-to-dominant” edge.

This event, the beginning of the Start of Frame (SOF) bit, is used by all receiving node to establish synchronization with the transmitter. For the receiving node, all subsequent occurrences of a detected “recessive to dominant” edge transition will result in re-synchronization.

To provide synchronization, the CAN protocol uses a bit stuffing technique to ensure sufficient edge transmissions during the transfer of the message. Under normal communication conditions, the CAN protocol guarantees the longest time between “recessive to dominant” resynchronization edges to be a 10 bit time interval.

Automatically handled by silicon (i.e. no software intervention is required), a single stuff bit is inserted after any transmission of five consecutive bits with the same data value. The added stuff bit is a non-data bit that is the opposite polarity of the previous five data bits and after transmission is automatically removed from the data stream when decoded by a CAN receiver.

The error control mechanisms of the CAN protocol result in even longer periods between synchronization edges under specific conditions, which will be examined in upcoming sections.

A receiving node maintains normal re-synchronization to a transmitted message if each “recessive to dominant” edge is received during the receiving node’s Synchronization Segment (SYNC_SEG), which is the first TQ of each bit period. However, if the “recessive to dominant” edge is detected outside of the receiving node’s Synchronization Segment, then the CAN protocol will either shortened TSEG2 or lengthen TSEG1 to adjust the timing in the appropriate direction as shown in Figure 7.

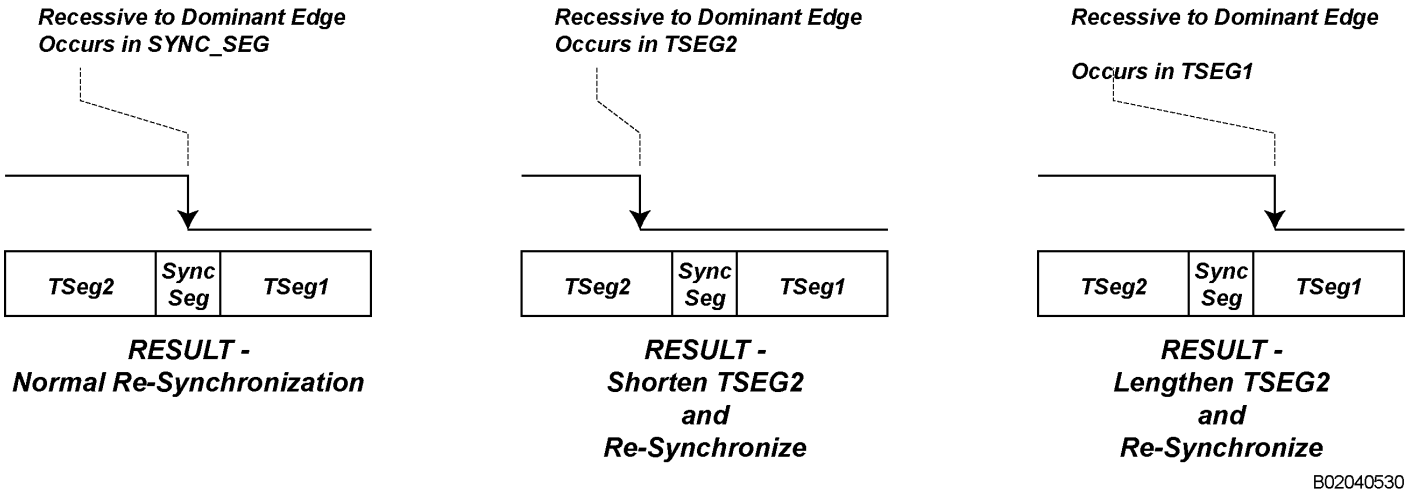

Figure 7 – Re-Synchronization Processing

This process uses the Re-synchronization Jump Width or SJW value as a part of the time segment adjustment.

In the case of a receiver synchronizing to a faster transmitter, a recessive to dominant edge is received during the TSEG2 interval. The receiving node shortens its TSEG2 interval by an amount up to the specified maximum SJW interval, t_{SJW}, to re-synchronize to the faster transmitter. In this manner, the receiver re-aligns with the data transmitted by the faster node to prevent potential false sampling of data during an incorrect bit time.

When a receiver is synchronizing to a slower transmitter and a recessive to dominant edge is received during the TSEG1 interval, the receiving node lengthens its TSEG1 interval by an amount up to the specified maximum SJW interval, t_{SJW}, to re-synchronize to the slower transmitter.

3.6 Re-Synchronization Jump Width

Within some limit, resynchronization compensates for oscillator drift and phase differences between nodes. It is the user-designated SJW value that determines how much a bit period may be lengthened or shortened. In general, the larger the SJW value, the system will be more tolerant to oscillator drift, however, the SJW interval cannot simply be programmed to its largest possible value, because each TSEG1 and TSEG2 interval must be as large or larger than the maximum SJW value to guarantee resynchronization.

In addition, TSEG1 and TSEG2 values directly influence the sample point location, which determines how much system propagation delay can be tolerated before synchronization is lost. The later the sample point occurs in the bit period, the more tolerant the system is to propagation delay.

When the SJW value is set to its maximum value, (with the TSEG2 interval being at least as large), the sample point may not be set as late in the bit period as possible. Thus, increasing SJW to provide more resynchronization will reduce the system tolerance for propagation delay.

Some balance in the selection of bit timing parameters is required to guarantee synchronization at the extremes of the specified oscillator drift while simultaneously maximizing the amount of propagation delay which can be tolerated for the system.

3.7 Sample Point

An additional consideration in optimization of the CAN bit timing parameters is the number of sample points in the bit period.

The CAN protocol allows either single sample mode or three sample mode (a majority 2 out of 3 vote determines the sample value). Because additional TQ periods during the TEG1 are required, the three sample mode effectively reduces the amount of propagation delay which the system can tolerate.

The application note considers both sampling modes and derives the appropriate equations for each considered case.

4 Determining Worst Case Bit Timing Scenarios

In order to optimize the CAN bit timing parameters, it is important to determine worst case scenarios and this requires examination of all protocol bit-level timing processes for both -

- Normal error-free communications
- Error mode communications

Depending on the system configuration, the noise conditions may occur at a negligible rate, however, if the bit timing parameters are not properly configured, loss of synchronization and message corruption may occur due to oscillator drift and propagation delay even with no external noise corruption.

5 Bit Timing Requirements for Normal Communication

5.1 Worst Case Synchronization

In general, the maximum allowable oscillator drift is limited by the maximum amount of user-selected resynchronization (set by the SJW value) and the worst case synchronization time (maximum time between synchronization edges). Because the CAN protocol uses bit stuffing to ensure sufficient edge transmissions during the transfer of the message and because resynchronization only occurs on “recessive to dominant” bit transitions, as shown in Figure 8, the worst case recurring bit pattern is 0000011111. This means that the worst case synchronization time (under normal communication conditions) is 10 bit periods.

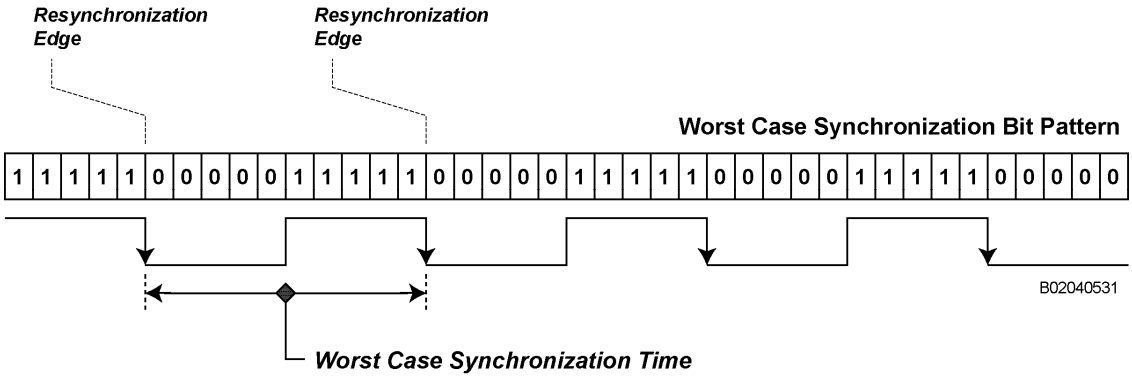

Figure 8 – Worst-Case CAN Communication Bit Pattern

As shown in Figure 9, timing differences between different nodes can easily cause problems.

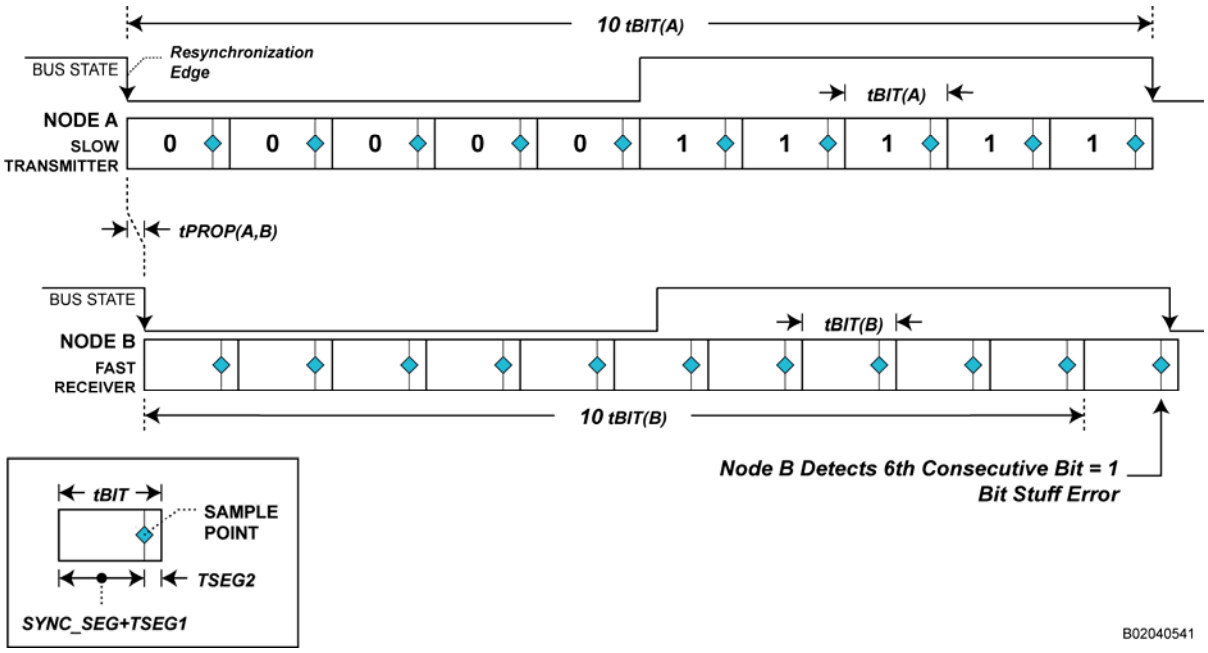

Figure 9 – Fast Receiver Samples Incoming Slow Transmitter Bit Pattern and Detects an Error

5.2 Accumulated Phase Error

If the time difference or phase shift between network nodes exceeds the maximum SJW interval, receiving nodes are not able to fully compensate for the phase shift. This results in a residual phase shift error that may accumulate.

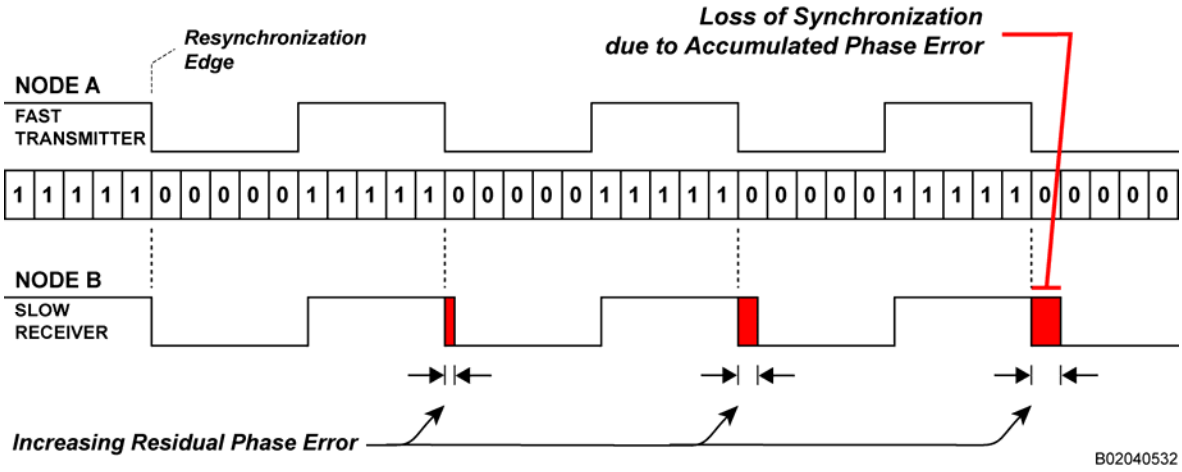

Figure 10 – Loss of Synchronization Due to Accumulating Phase Error

In this example, Node B, a slow receiver, is re-synchronizing to Node A, a fast transmitter at each “recessive to dominant” edge. Within a given bit interval, any phase error greater than the SJW interval does not necessarily cause a synchronization error when the time difference is not sufficiently large to cause an invalid sample.

However, if phase errors occur over successive bit periods and are larger than t_{SJW}, the residual error, which remains after resynchronization can accumulate potentially causing invalid data sampling. This may result in a loss of synchronization, as shown in Figure 10. In order to study the loss of synchronization due to accumulating phase

error, it is necessary to determine the maximum possible number of successive ten bit intervals with a single synchronization edge. While this case seems highly unlikely, it is statistically possible and discussed in Section 9.

However, if we establish the following bit timing condition - the phase error over a given ten bit interval cannot exceed the size of the SJW interval - then only slightly more conservative oscillator tolerances are necessary.

With this simplifying bit timing assumption, the following five scenarios can be evaluated –

- Scenario 1 - Synchronizing a Fast Receiver to a Slow Transmitter
- Scenario 2 - Synchronizing a Slow Receiver to a Fast Transmitter
- Scenario 3 - Synchronizing a Fast Transmitter with a Slow Transmitter
- Scenario 4 - Synchronizing a Slow Transmitter with a Fast Transmitter
- Scenario 5 - Slow Transmitter Wins Arbitration Over a Fast Transmitter

5.2.1 Scenario 1 – Synchronizing a Fast Receiver to a Slow Transmitter

Figure 11 depicts the scenario in which a fast receiver synchronizes to a slow transmitter bit pattern.

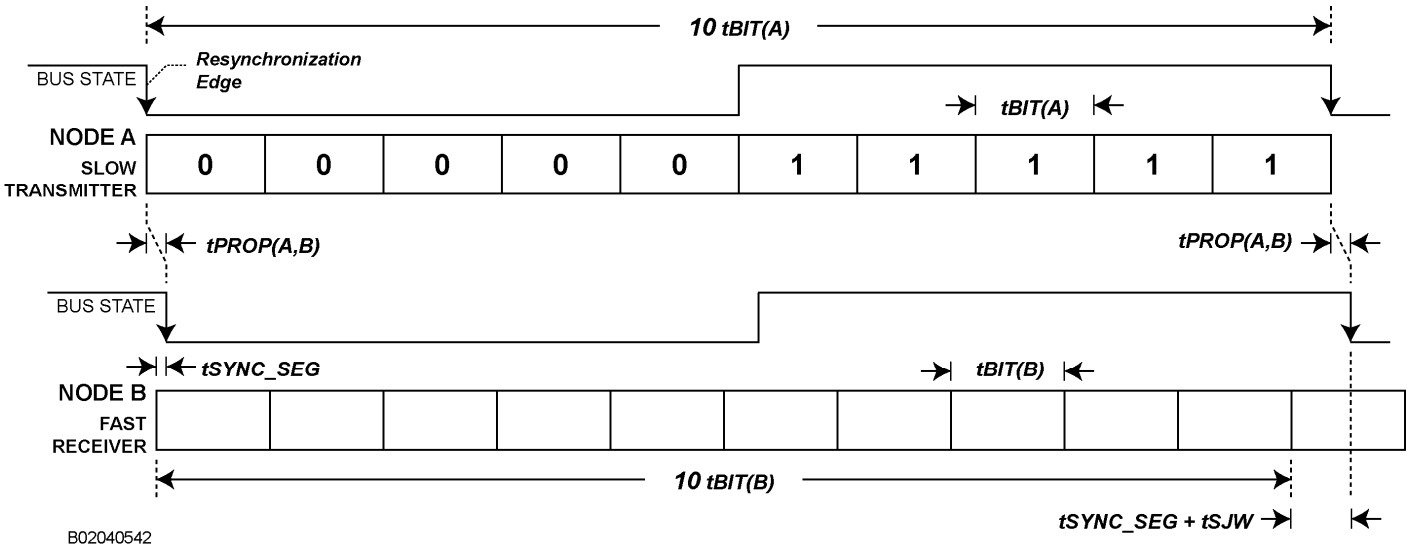

Figure 11 – Fast Receiver Synchronized to a Slow Transmitter

In this instance, we assume one full TQ of bit clock skew between the transmitter and the receiver. Notice that the receiver will assume that it is already synchronized if the incoming recessive to dominant edge occurs during the SYNC_SEG interval.

In order for the faster receiver to maintain synchronization to the incoming bit pattern, the next re-synchronization edge must be detected by the receiver before the receiver's 11th bit time interval exceeds the combined time of the SYNC_SEG interval and the programmed SJW interval.

This case is independent of sample mode and may be described mathematically as follows -

$$5)\quad 10t_{BIT(A)} + t_{PROP(A,B)} \leq t_{PROP(A,B)} - t_{Q(B)} + 10t_{BIT(B)} + t_{Q(B)} + t_{SJW(B)} \qquad \textbf{(1 or 3 Sample Mode)}$$

On inspection, the propagation delay from Node A to Node B cancels from both sides and the SYNC_SEG times (shown as tQ) cancel on the right side. Equation 5 can also be rewritten in terms of the nominal bit time intervals and the oscillator drift tolerance, Δf, using the following approximations –

6) $t_{INTERVAL(A)} \cong t_{INTERVAL\_NOM}(1+\Delta f)$

7) $t_{INTERVAL(B)} \cong t_{INTERVAL\_NOM}(1-\Delta f)$

Based on typical CAN controller references that use quartz crystals or ceramic resonators, we can assume that Δf <<1 and as a result, this approximation is reasonable. Using the relationships described in Equations 6 and 7, Equation 5 can be written in terms of the nominal interval lengths and Δf as follows -

8) $10t_{BIT}(1+\Delta f) \leq 10t_{BIT}(1-\Delta f) + t_{SJW}(1-\Delta f)$

Solving Equation 8 in terms of the SJW interval yields -

9) $t_{SJW} \geq \dfrac{20t_{BIT}\Delta f}{1-\Delta f}$ **(1 or 3 Sample Mode)**

Equation 9 indicates that as one increases the tolerance of the bit timing reference, the required SJW interval must increase.

5.2.2 Scenario 2 – Synchronizing a Slow Receiver to a Fast Transmitter

The case of a slow receiver synchronized to a fast transmitter is shown in Figure 12.

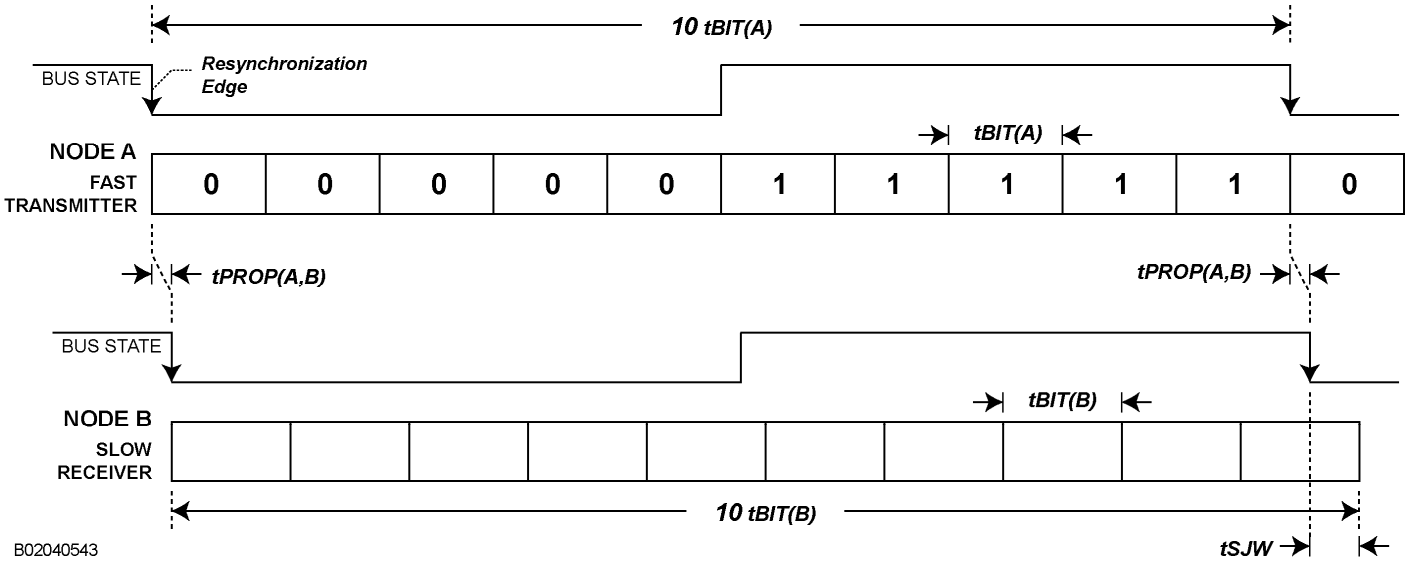

Figure 12 – Slow Receiver Synchronized to a Fast Transmitter

For this case timing, we assume that the receiver immediately synchronizes to the incoming recessive to dominant edge.

In order for the slow receiver to maintain synchronization to the incoming bit pattern, the next re-synchronization edge must be detected during the receiver's 10th bit time interval (during the TSEG2 segment) within the programmed SJW interval immediately before the upcoming SYNC_SEG interval of the next bit.

This case is also independent of sample mode and may be described mathematically as follows

10) $10t_{BIT(A)} + t_{PROP(A,B)} \geq 10t_{BIT(B)} + t_{PROP(A,B)} - t_{SJW(B)}$ **(1 or 3 sample Mode)**

Similar to the previous scenario, the propagation delay from Node A to Node B cancels from both sides of Equation 10 and solving in terms of the SJW interval, results in –

$$11) \quad t_{SJW} \geq \frac{20t_{BIT}\Delta f}{1+\Delta f} \quad \textbf{(1 or 3 sample Mode)}$$

5.2.3 Scenario 3 – Synchronizing a Fast Transmitter with a Slow Transmitter

The third scenario shown in Figure 13 considers two transmitters arbitrating with each other such that a fast transmitter is synchronized to a slow transmitter. For this analysis both nodes are assumed to be transmitting identical bit patterns such that neither node should lose arbitration and stop transmitting over this interval.

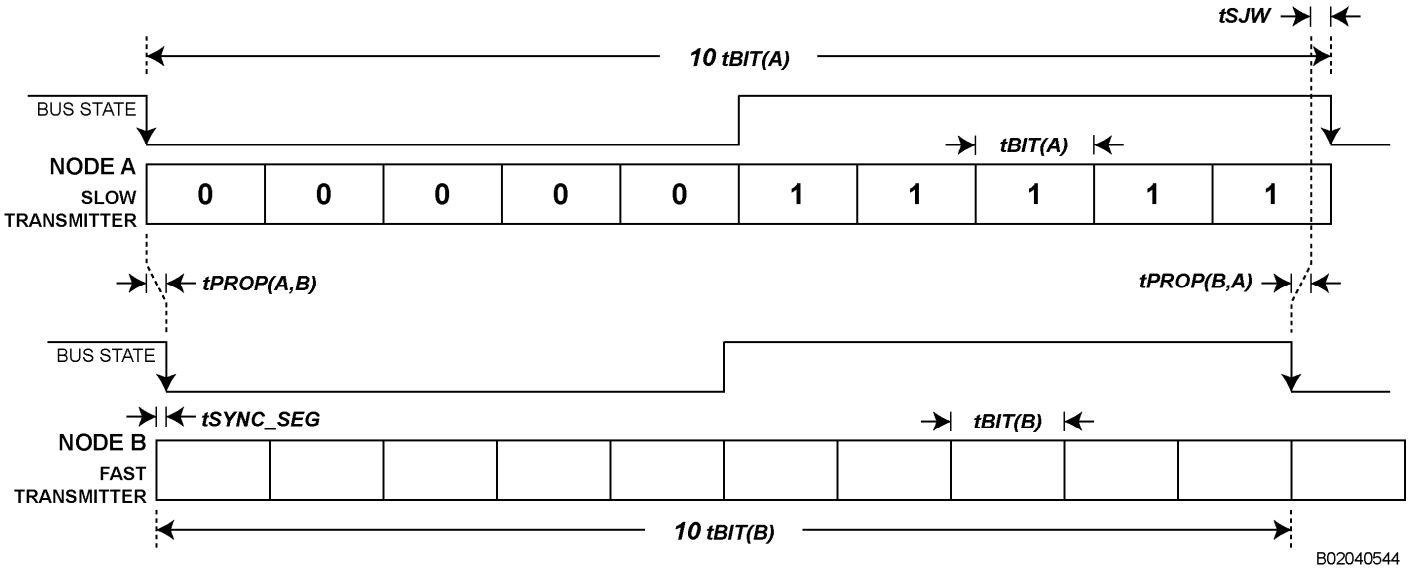

Figure 13 – Synchronizing a Slow Transmitter with a Fast Transmitter

In this instance, we assume one full TQ of bit clock skew between both transmitters.

The requirement for this condition for proper synchronization is that the recessive to dominant edge transmitted by Node B must be received by Node A within a SJW interval of its upcoming SYNC_SEG. The equation describing this condition, which is independent of sample mode, is -

$$12) \quad 10t_{BIT(A)} - t_{SJW(A)} \leq 10t_{BIT(B)} + t_{PROP(A,B)} + t_{PROP(B,A)} - t_{Q(B)} \quad \textbf{(1 or 3 Sample Mode)}$$

The sum of the propagation delay from Node A to Node B and from Node B to Node A is equal to t_{PROP}. Substituting into Equation 12 in terms of nominal bit interval times and CAN clock tolerance, the requirement for the minimum SJW interval, independent of sample mode, is given in Equation 13 as follows -

$$13) \quad t_{SJW} \geq \frac{\Delta f\,(20t_{BIT} - t_Q) + t_Q - t_{PROP\min}}{1 + \Delta f} \quad \textbf{(1 or 3 Sample Mode)}$$

For this case, the dependency worth noting is that the worst case, minimum SJW interval requirement occurs at the *minimum* propagation delay between nodes, i.e. nodes, which are physically close.

5.2.4 Scenario 4 – Synchronizing a Slow Transmitter with a Fast Transmitter

The timing requirements for synchronizing a slow transmitter with a fast transmitter, shown in Figure 14, are the same as for Scenario 2 and result in an identical requirement for the minimum SJW interval size as described by Equation 11, again for both the single sample and three sample modes.

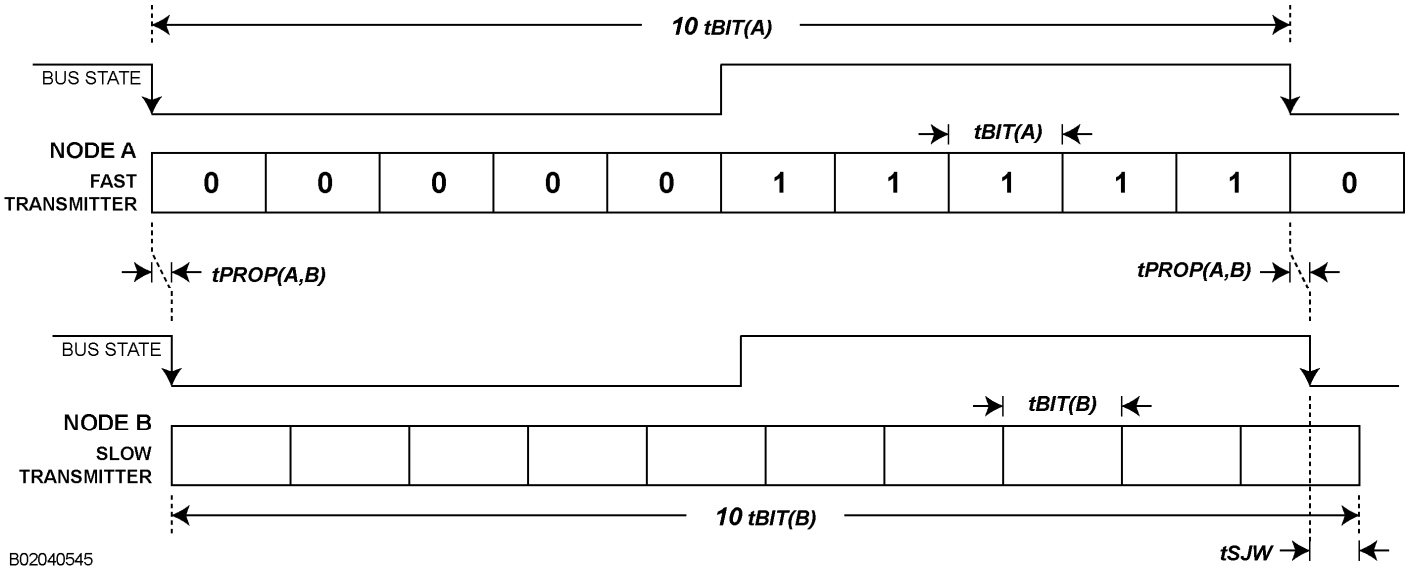

Figure 14 – Synchronizing A Slow Transmitter With A Fast Transmitter

5.2.5 Scenario 5 – Slow Transmitter Wins Arbitration Over a Fast Transmitter

Scenario 5, shown in Figure 15, analyzes the condition of a fast transmitter and a slow transmitter transmitting the identifier at the same time during which the slower node wins arbitration.

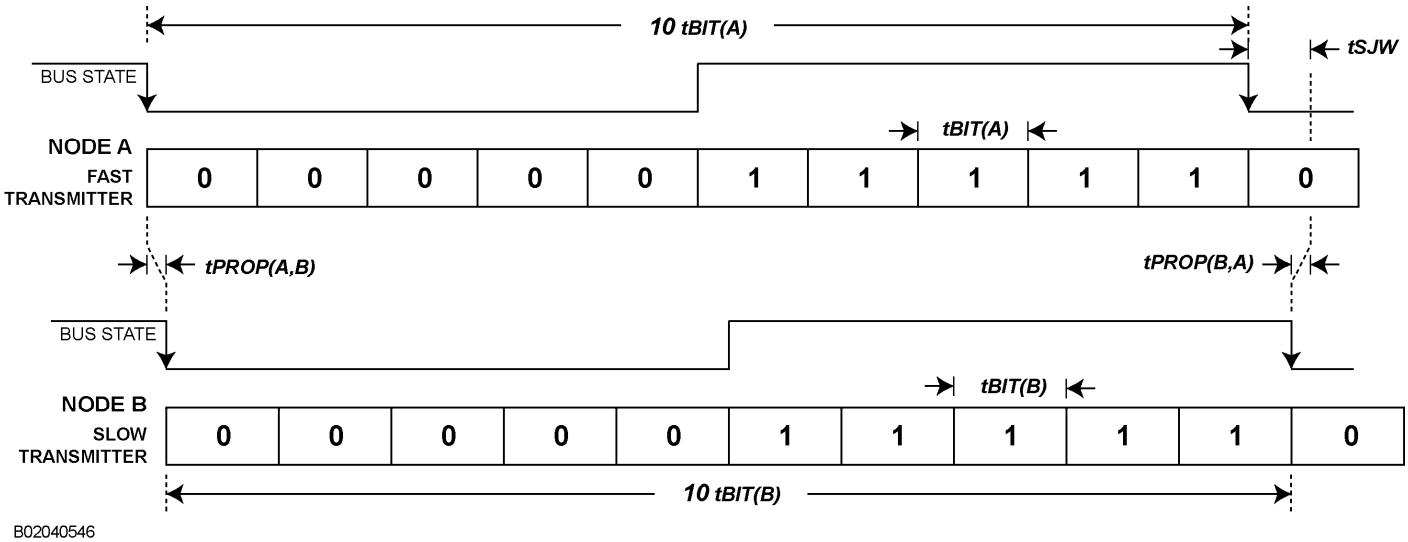

Figure 15 – Slow Transmitter Wins Arbitration Over a Fast Transmitter

Using the maximum phase error simplification assumption, Equation 14 describes the limiting condition for the phase error not to exceed the size of the SJW interval -

14) $t_{PROP(A,B)} + 10t_{BIT(B)} + t_{PROP(B,A)} \leq 10t_{BIT(A)} + t_{SJW(A)} + t_{Q(A)}$ **(1 or 3 Sample Mode)**

Solving for the minimum required t_{SJW} in terms of the oscillator drift tolerance results in Equation 15 -

15) $t_{SJW} \geq \frac{t_{PROP\max} - t_Q + \Delta f\,(20t_{BIT} + t_Q)}{1 - \Delta f}$ **(1 or 3 Sample Mode)**

5.2.6 Overall Timing Requirements

Result 1

Because the typical CAN controller implementation limits the selectable SJW values to be at most 25% of the bit period, it is the relative size to the network bit period of the system propagation delay that establishes whether resynchronization is guaranteed. Based on equation 15 –

- The minimum SJW interval must be equal to or greater than the maximum system propagation delay.

Result 2

To guarantee that uncompensated phase error does not accumulate over multiple bit periods and result in an invalid sample and loss of synchronization, as shown in Figure 11-

- The phase error in any given bit period must not exceed the SJW interval.

Result 3

To guarantee both that the uncompensated phase error will not accumulate and that the phase error will not exceed the TSEG2 interval width,

- The TSEG2 period is either equal or greater than the SJW interval

Result 4

- The phase error in a given bit period can only be corrected by an amount up to the SJW interval.

Therefore, when the phase error is greater than the SJW interval, the uncompensated phase error accumulates in the subsequent bit period.

As shown in Figure 15, it is possible that the positive phase error may exceed the SJW interval in a given bit period by a relatively large amount without loss of synchronization. Since the faster transmitter loses arbitration and reverts to a receiver, this condition will not recur over multiple bit periods. Once only one transmitter remains on the network, the phase error for all receivers is reduced by a maximum SJW interval at each recessive to dominant edge until the phase error is reduced to zero. For these reasons, Scenario 5 is not considered a limiting condition for minimum SJW interval size. However, it does impose a restriction on the maximum TSEG2 interval size to guarantee that an invalid sample is not taken.

Result 5

The equations derived for the first four scenarios described above which define the minimum required SJW interval size give approximately the same result, with a slightly larger interval size required for cases for which the one TIME_QUANTUM skew is worst case. Equations 9 and 13 represent the most conservative requirements for the SJW interval. Which equation is most appropriate is dependant on the selected t_Q value and the minimum propagation delay, $t_{PROPmin}$.

5.3 Maximum TSEG2 Interval for Propagation Delay Tolerance

The ability to maintain synchronization for a specific system propagation delay depends on the bit period and the location of the sample point in the bit period. In general, the slower the bit rate and later the sample point occurs in the period, the greater the allowable system propagation delay.

The worst case condition (Scenario 5) for propagation delay in normal communication results from a fast transmitter and a slow transmitter simultaneously arbitrating, with the faster node losing arbitration to the slower node and becoming a receiver.

With respect to propagation delay, the timing requirement which must be satisfied is that the 11th bit transmitted by the slow transmitter must not arrive at the fast transmitter so late as to be sampled incorrectly.

The mathematical requirement for this condition is sample mode dependent. For the single sample mode, the requirement is expressed -

16) $$11t_{BIT(A)} - t_{SEG2(A)} \geq t_{PROP(A,B)} + 10t_{BIT(B)} + t_{PROP(B,A)}$$ **(1 Sample Mode)**

In terms of the TSEG2 interval size, the requirement becomes -

17) $$t_{SEG2} \leq \frac{t_{BIT} - t_{PROP\max} - 21t_{BIT}\Delta f}{1 - \Delta f}$$ **(1 Sample Mode)**

For networks using the three sample mode, Equation 16 is adjusted for the additional two samples -

18) $$11t_{BIT(A)} - t_{SEG2(A)} - 2t_{Q(A)} \geq t_{PROP(A,B)} + 10t_{BIT(B)} + t_{PROP(B,A)}$$ **(3 Sample Mode)**

In terms of the TSEG2, the requirement becomes -

19) $$t_{SEG2} \leq \frac{t_{BIT} - t_{PROP\max} - 2t_Q - \Delta f(21t_{BIT} - 2t_Q)}{1 - \Delta f}$$ **(3 Sample Mode)**

The three sample mode reduces the amount of allowable system propagation delay by approximately two t_Q and results in the following heuristic –

- Use the single sample mode for high speed networks

Equations 17 and 19 demonstrate the relationship between system propagation delay and the location of the sample and establish an upper limit on the TSEG2 interval size under normal communication conditions. As the maximum system propagation delay increases, the size of the TSEG2 interval must decrease, which corresponds to moving the sample point towards the end of the bit period.

To guarantee synchronization –

- The propagation delay cannot exceed the bit period

6 Bit Timing Requirements for Error Mode Conditions

In contrast to normal communication conditions, the bit timing requirements for error mode conditions are equally important.

6.1 CAN Protocol Error Handling Background

Based on protocol rules, once any transmitting or receiving node detects an error, that node must indicate the error by transmitting an Error Flag such that the entire network membership will be notified of the error condition. Since

the Error Flag is represented by six identical bits which is immediately detected by the CAN receiver logic as a Bit Stuff error, the ability to notify the network is relatively simple and quick.

CAN controllers implement individual transmit and receive error counters, which essentially –

- are incremented as errors are detected
- are decremented as successful transfer occur

Depending on the occurrence of both local and global noise conditions and the severity of such error conditions, the basic CAN protocol fault confinement strategy causes nodes to individually react based on their respective local error counters.

When a transmitting CAN controller's transmit error counter exceeds a specific limit, that transmitter will immediately transition to a "bus off" state and effectively remove itself from the bus.

When a receiving CAN controller's receive error counter exceeds a specific limit, that receiver will transition to a "error passive" state and effectively remove its ability to actively "kill" messages.

Excessive oscillator drift or propagation delay may influence a node to misinterpret error conditions - causing the node to reach a given error state after too few or too many errors.

If proper error control operation, that is correct interpretation of whether the error was a global or local error and correct advancement of the error counters, is required, various scenarios which may occur are considered which also impact the limits on oscillator tolerance and propagation delay.

6.2 Maximum TSEG2 Interval to Guarantee Proper Error Control

The first investigated scenario is a global error condition in which noise on the bus corrupts the reception of an expected recessive bit that follows 5 dominant bits as shown in Figure 16.

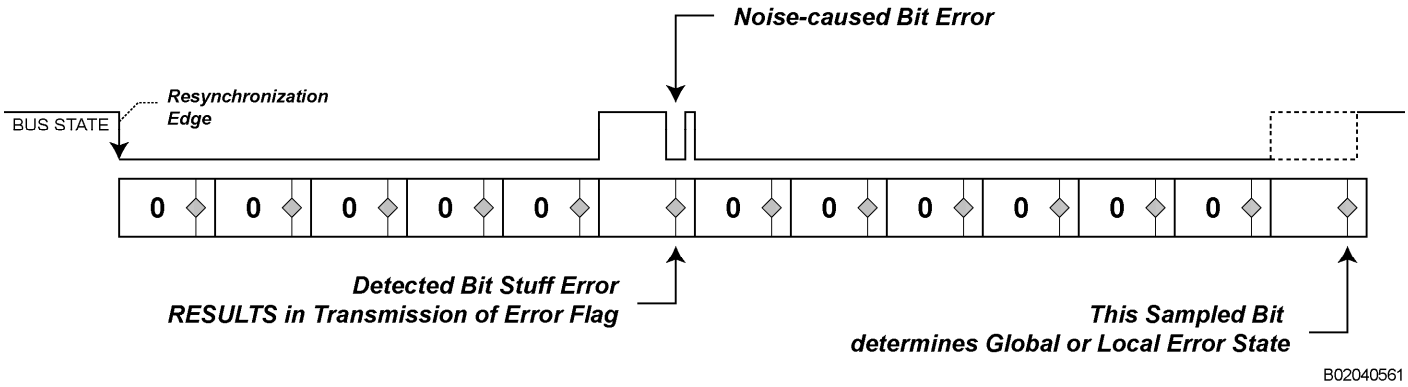

Figure 16 – Noise Induced Error Impacting Bit Timing

A transmitter has transmitted 5 dominant bits which should be immediately followed by a recessive bit, however, this bit is corrupted such that all receivers as well as the transmitter detect a dominant bit – interpreted as a bit stuff error.

Upon detection of this error, each CAN controller immediately transmits an Error Flag beginning in the next subsequent bit period.

And upon completion of sending the Error Flag in the next subsequent bit period, each node samples the bus –

- if this bit is recessive, a global error is interpreted
- if this bit is dominant, a local error is interpreted

Depending on whether the error is interpreted as local or global, a node will advance its error counters differently. Propagation delay and oscillator drift between nodes on the network may cause nodes to improperly sample a dominant bit.

In comparison to normal communication, this error mode condition establishes a more conservative maximum allowable TSEG2 interval width because a longer time period is involved.

Two cases associated with this scenario are considered -

- Circumstance 1 - Fast Transmitter and Slow Receiver
- Circumstance 2 - Fast Transmitter with Close Fast Receiver and Far Slow Receiver

Whether the following two cases are relevant or important to a specific CAN communication system is outside the scope of this discussion and should be determine by the appropriate distributed embedded system engineering activity.

6.2.1 Circumstance 1 – Fast Transmitter and Opposite End Slow Receiver

Considering just two nodes for this case, the worst case condition is that of a fast transmitter and a slow receiver as shown in Figure 17.

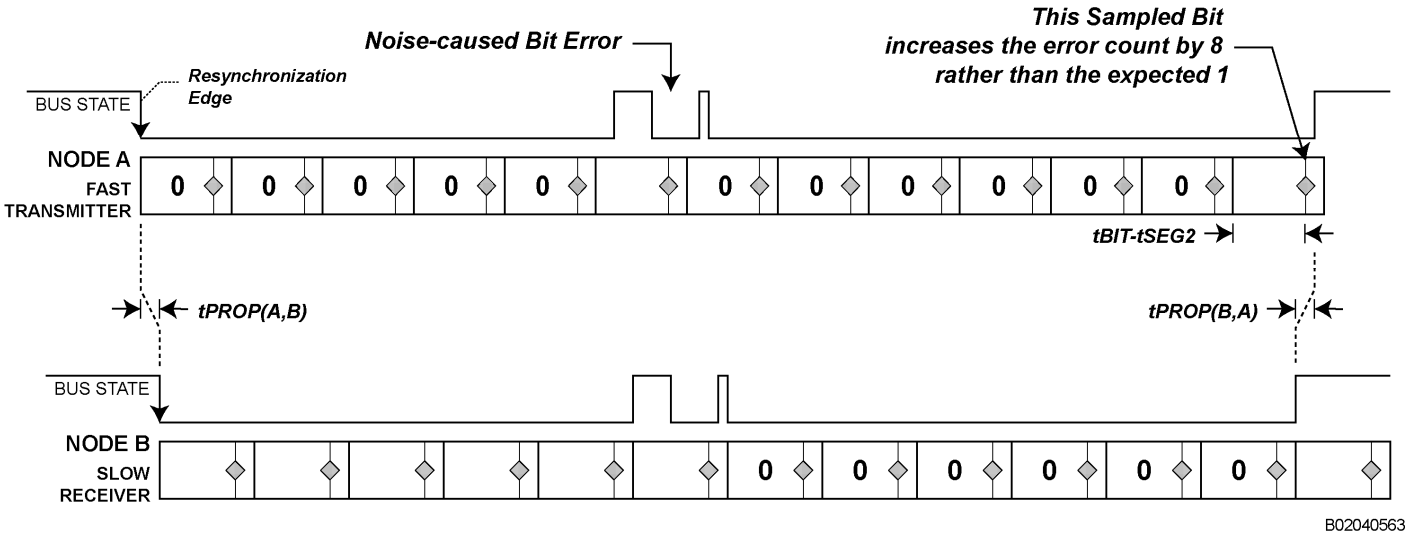

Figure 17 – Incorrect Transmit Error Counting Due to Timing Mismatch

When the last bit period of the slow receiver's Error Flag extends to the sample point of the fast transmitter's 13th bit period, the transmitter's error counter will be incorrectly advanced by 8 instead of by 1 with the limiting condition being sample mode dependent.

For the single sample mode, the timing requirement for proper error counter advancement is -

$$20)\quad t_{PROP(A,B)} + 12t_{BIT(B)} + t_{PROP(B,A)} \le 13t_{BIT(A)} - t_{SEG2(A)} \qquad \textbf{(1 Sample Mode)}$$

To guarantee proper error control, the maximum TSEG2 interval width is effectively equal to the maximum system propagation delay as follows -

$$21)\quad t_{SEG2} \le \frac{t_{BIT} - t_{PROP\max} - 25t_{BIT}\Delta f}{1 - \Delta f} \qquad \textbf{(1 Sample Mode)}$$

In the three sample mode, the requirement for proper error mode control is that the last bit of the Error Flag transmitted by slow receiver must not extend beyond the first sample of fast transmitter's 13th bit period as follows -

22) $t_{PROP(A,B)} + 12t_{BIT(B)} + t_{PROP(B,A)} \leq 13t_{BIT(A)} - t_{SEG2(A)} - 2t_{Q(A)}$ **(3 Sample Mode)**

Solving for the worst case maximum TSEG2 interval width results in -

23) $t_{SEG2} \leq \frac{t_{BIT} - 2t_Q - t_{PROP\max} - \Delta f(25t_{BIT} - 2t_Q)}{1 - \Delta f}$ **(3 Sample Mode)**

6.2.2 Circumstance 2 – Fast Transmitter with Close Fast Receiver and Far Slow Receiver

The same global noise error condition involving three nodes results in a slightly more conservative limit on the maximum TSEG2 interval. This case involves three nodes - a fast transmitter and a close fast receiver with minimum propagation delay between each plus a distant slow receiver with maximum propagation delay between it and the fast transmitter as shown in Figure 18.

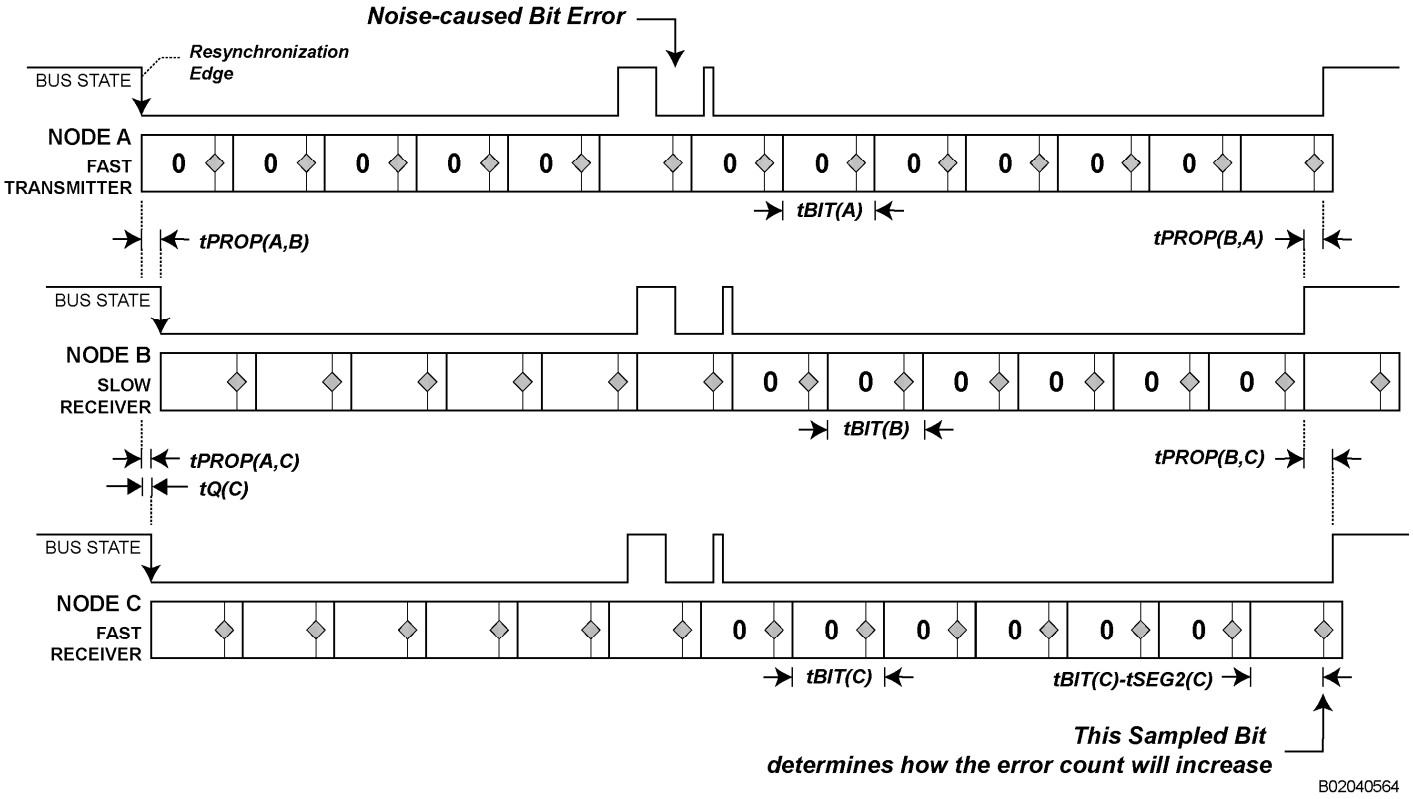

Figure 18 – Incorrect Receive Error Counting Due to Timing Mismatch

With the fast receiver assumed to be synchronized "early" to the fast transmitter, the timing requirement for proper advancement of the fast receiver's error counter is that the last dominant bit of the Error Flag transmitted by the slow receiver must not be detected when the fast receiver samples the 13th bit – a recessive bit is expected.

For the single sample mode, this is expressed as -

24) $t_{PROP(A,B)} + 12t_{BIT(B)} + t_{PROP(B,C)} \leq t_{PROP(A,C)} + 13t_{BIT(C)} - t_{Q(C)} - t_{SEG2(C)}$ **(1 Sample Mode)**

In this case, the following assumptions regarding the propagation delays are made as follows;

25) $t_{PROP(A,B)} + t_{PROP(B,C)} = t_{PROP\max}$

26) $t_{PROP(A,C)} = \frac{t_{PROP\min}}{2}$

While the term $t_{PROPmax}$ represents the maximum allowable propagation delay in the system between nodes at opposite ends of the bus, the term $t_{PROPmin}/2$ represents the minimum possible propagation delay, in one direction, for two nodes that are physically adjacent. Based on these assumptions and solving for t_{SEG2} produces -

27) $t_{SEG2} \leq \frac{t_{BIT} - t_{PROP\max} - t_Q - \Delta f\,(25t_{BIT} - t_Q) + \frac{t_{PROP\min}}{2}}{1 - \Delta f}$ **(1 Sample Mode)**

For the three sample mode, the timing requirement for proper receive error counter advancement is –

28) $t_{PROP(A,B)} + 12t_{BIT(B)} + t_{PROP(B,C)} \leq t_{PROP(A,C)} + 13t_{BIT(C)} - 3t_{Q(C)} - t_{SEG2(C)}$ **(3 Sample Mode)**

Solving for the maximum TSEG2 interval for the three sample mode results in -

29) $t_{SEG2} \leq \frac{t_{BIT} - t_{PROP\max} - 3t_Q - \Delta f\,(25t_{BIT} - 3t_Q) + \frac{t_{PROP\min}}{2}}{1 - \Delta f}$ **(3 Sample Mode)**

7 Resultant Bit Timing Parameters

Based on CAN controller oscillator tolerance and propagation delay, Table 1 summaries the full range of bit timing requirements necessary for proper operation along with the corresponding assumptions and conditions that apply to each derived equation.

Eqn #	Equation	Conditions/Assumptions
9)	$t_{SJW} \geq \frac{20t_{BIT}\Delta f}{1-\Delta f}$	* Normal Communication * Single or Three Sample Mode * Synchronization Over a 10 Bit Interval * Fast Receiver Synchronized to a Slow Transmitter
13)	$t_{SJW} \geq \frac{\Delta f(20t_{BIT}-t_Q)+t_Q-t_{PROP\min}}{1+\Delta f}$	* Normal Communication * Single or Three Sample Mode * Arbitration Over a 10 Bit Interval * Fast Transmitter Synchronized to a Slow Transmitter * Minimum Propagation Delay Between Adjacent Nodes
17)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-21t_{BIT}\Delta f}{1-\Delta f}$	* Normal Communication * Single Sample Mode * Arbitration Over a 10 Bit Interval * Fast Transmitter Synchronized to a Slow Transmitter
19)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-2t_Q-\Delta f(21t_{BIT}-2t_Q)}{1-\Delta f}$	* Normal Communication * Three Sample Mode * Arbitration Over a 10 Bit Interval * Fast Transmitter Synchronized to a Slow Transmitter
21)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-25t_{BIT}\Delta f}{1-\Delta f}$	* Error Frame, Global Noise Scenario * Single Sample Mode * Slow Receiver Synchronized to a Fast Transmitter
23)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-2t_Q-\Delta f(25t_{BIT}-2t_Q)}{1-\Delta f}$	* Error Frame, Global Noise Scenario * Three Sample Mode * Slow Receiver Synchronized to a Fast Transmitter
27)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-t_Q-\Delta f(25t_{BIT}-t_Q)+\frac{t_{PROP\min}}{2}}{1-\Delta f}$	* Error Frame, Global Noise Scenario * Single Sample Mode * Fast Transmitter, 2 Slow Receivers (1 Close, 1 Far)
29)	$t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\max}-3t_Q-\Delta f(25t_{BIT}-3t_Q)+\frac{t_{PROP\min}}{2}}{1-\Delta f}$	* Error Frame, Global Noise Scenario * Three Sample Mode * Fast Transmitter, 2 Slow Receivers (1 Close, 1 Far)
i.	$t_{SJW} \leq \min(t_{SEG1}, t_{SEG2})$	* Normal Communication * Single or Three Sample Mode * Guarantees Resync Logic Works Correctly
ii.	$t_{SEG1} \geq 2t_Q$	* Normal Communication * Single Sample Mode * Guarantees Resync Logic Works Correctly
iii.	$t_{SEG1} \geq 4t_Q$	* Normal Communication * Three Sample Mode * Guarantees Resync Logic Works Correctly
iv.	$t_{SEG2} \geq 2t_Q$	* Normal Communication * Single Sample or Three Sample Mode * Guarantees Resync Logic Works Correctly

Table 2 – Summary of CAN Bit Timing Requirements

8 Determining Optimal Bit Timing Parameters

8.1 TIME_QUANTUM Period, t_q

Choose a value for the TIME_QUANTUM as small as practical - the smaller the TIME_QUANTUM the better the resolution in selecting the location of the sample point in the bit period as well as the size of the SJW interval. With a selected TIME_QUANTUM value, the number of TIME_QUANTA in a single bit time period, N_Q, can easily be determined using Equation 2 -

$$2)\quad N_Q = \frac{t_{BIT(network)}}{t_Q}$$

For all practical implementations, the number of TIME_QUANTA in a bit period must be an integer value. When the value calculated for N_Q is not exactly an integer value, the effective oscillator drift for that node in the system is increased and in most cases makes the communication system unusable.

Since the maximum size of the SJW interval and the SYNC_SEG interval are typically fixed by the selection of the TIME_QUANTUM, the choice of a TIME_QUANTUM value too small a may not provide a sufficiently large SJW interval to guarantee synchronization for a given oscillator tolerance. Sizing of the TIME_QUANTUM may have to be determined iteratively.

8.2 CAN Controller Clock Oscillator Tolerance

From a systems engineering point of view, select the system-level oscillator tolerance for the entire CAN communication system. Based on choice of using either a crystal or a ceramic resonator, this decision is relatively simple – until low cost, high precision ceramic resonators are available, use crystals.

8.3 Minimum Required SJW Interval

For the selected clock oscillator tolerance and either single sample and three sample mode, evaluate both Equations 9 and 13 to determine the minimum required size of the SJW interval.

$$9)\quad t_{SJW} \geq \frac{20t_{BIT}\Delta f_{effective}}{1-\Delta f_{\text{effective}}}$$

$$13)\quad t_{SJW} \geq \frac{\Delta f_{effective}\,(20t_{BIT}-t_Q)+t_Q-t_{PROP\,\min}}{1+\Delta f_{\text{effective}}}$$

Because the minimum propagation delay possible between two nodes on a given system may be difficult to determine when evaluating Equation 13, a reasonable assumption may be to use a $t_{PROPmin}$ value equal to zero.

8.4 Maximum TSEG2 Interval

For single sample mode, the maximum allowable TSEG2 interval width is -

$$17)\quad t_{SEG2} \leq \frac{t_{BIT}-t_{PROP\,\max}-21t_{BIT}\Delta f}{1-\Delta f}$$ **(1 Sample Mode)**

If proper error control is required under all possible conditions, then the more stringent Equations 21 and 27 should be used.

While the selected TSEG2 interval must be at least 2 TQ or equal the size of the SJW interval, to provide the maximum noise margin against noise events, the largest possible value should be chosen for the TSEG2 interval.

With the computed minimum value of t_{SJW}, the integer number of TIME_QUANTA required may be determined for programming the CAN controller. The maximum number of TIME_QUANTA which can be allocated to the SJW interval is 4, as defined by the CAN protocol specification. If the chosen TIME_QUANTUM value is too small to allow the CAN controller to be programmed to satisfy the requirements of Equations 9 and 13, the TIME_QUANTUM size must be increased.

8.5 Example Calculations

As a system development example, consider the design of a 500K BPS network using single sample mode. A crystal derived clock reference is assumed, with a nominal frequency of 16MHz with a tolerance of 0.5%. These assumptions along with assumptions regarding the transceiver characteristics, the logic delays, and the bus signal propagation delay are summarized in Table 3.

Parameter	Description	Minimum	Nominal	Maximum
f_{OSC}	Nominal Oscillator Frequency	-	16.0 MHz	-
Δf	Oscillator Tolerance	-	-	0.5%
t_{TX}	Transmitter Propagation Delay	50nS		340nS
t_{RX}	Receiver Propagation Delay	10nS		50nS
t_{LOGIC}	Logic Propagation Delay	10nS		60nS
t_{BUS}/meter	Bus Propagation Delay	5nS/meter		5.5nS/meter
L_{INTER_NODE}	Node to Node Bus Length	1 meter		40 meters

Table 3 – Example System Tolerances

The steps for this example are as follows -

1. Determine the Nominal Bit Time and the Size of the TIME_QUANTUM, t_Q.

At 500K BPS, the nominal bit time is 2.0uS. A reasonable first assumption for TIME_QUANTUM value in this case is 125ns, which implies 16 TIME_QUANTA in the nominal bit period. The number of TIME_QUANTA is typically chosen between 8 and 25 per bit period.

2. Determine the Effective Oscillator Tolerance.

Since the nominal bit time is exactly an integer number of nominal TIME_QUANTA (N_Q=16), no adjustment to the specified oscillator tolerance needs to be made, and therefore $\Delta f_{effective}$ is 0.5% for this example.

3. Determine the Minimum Required SJW Interval Size.

The minimum required SJW interval must be evaluated from both Equations 9 and 13. Using Equation 9 -

$$t_{SJW} \geq \frac{0.005(20 \text{ x } 2\text{uS})}{1 - 0.005} = 201\text{nS}$$

Determining the minimum propagation delay is necessary before evaluating Equation 13. From Equation 4, $t_{PROPmin}$ is given as follows -

$$t_{PROP\,\min} = 2(t_{TX} + t_{RX} + t_{LOGIC} + t_{BUS})_{\min}$$

Assuming a minimum bus length of 1 meter and the maximum bus propagation velocity of 5 nsec per meter, the minimum propagation delay for this example is given by -

$$t_{PROP\min} = 2(50ns + 10ns + 10ns + (5ns/m) \text{ x } 1\text{meter})$$
$$= 150\text{nS}$$

Evaluating the minimum required SJW interval from Equation 13) yields;

$$t_{SJW} \geq \frac{0.005(20 \text{ x } 2\text{uS} - 125\text{nS}) + 125\text{nS} - 150\text{nS}}{1 - 0.005} = \quad 175\text{nS}$$

The more conservative condition for this case is determined by using Equation 9, and so the actual SJW interval is 250nS, which is the 201nS calculated from Equation 9, rounded up to the next integer number of TIME_QUANTA.

4. Determine the Maximum TSEG2 Interval

Assuming the requirement for this system is proper error counter control under all conditions, with single sample mode, only Equations 21 and 27 need be considered. In order to evaluate these equations, the maximum system propagation delay needs to be determined. Using Equation 4, the maximum system propagation delay is calculated as follows -

$$t_{PROP\max} = 2(t_{TX} + t_{RX} + t_{LOGIC} + t_{BUS})_{\max}$$

Assuming a maximum bus length of 40 meters and the minimum bus propagation velocity, the maximum propagation delay for this example is given by;

$$t_{PROP\max} = 2(340ns + 50ns + 60ns + (5.5ns/m) \text{ x } 40\text{meters})$$
$$= 1340\text{nS}$$

Calculating the maximum TSEG2 interval size from Equation 21 -

$$t_{SEG2} \leq \frac{2uS - 1.34uS - 25(2uS)(0.005)}{1 - 0.005}$$
$$t_{SEG2} \leq 0.412\text{us} = 412\text{nS}$$

Calculating the maximum TSEG2 interval size from Equation 27 -

$$t_{SEG2} \leq \frac{2uS - 1.34us - 125ns - 0.005(25\text{x}2\text{uS} - 125\text{ns}) + 152nS/2}{1 - 0.005}$$
$$t_{SEG2} \leq 0.363\text{us} = 363\text{nS}$$

The most conservative requirement on the maximum TSEG2 interval size is 363nS, as dictated by Equation 27. From the requirements in step 3, the minimum TSEG2 interval size for a TIME_QUANTUM of 125nS was calculated to be 2 TIME_QUANTA, or 250nS. From these requirements, the TSEG2 interval size must be less than 363nS. In this case, the required value must be rounded down to an integer number of TIME_QUANTA. This results in a requirement that the TSEG2 interval can be no larger than 2 TIME_QUANTA, or 250nS. For this example there is no latitude in the selection of the TSEG2 interval size, i.e. TSEG2 = 2 TQ = 250nS is the only possible solution for this example.

The final bit timing parameters for this example are

TQ = 125nS

TSEG1 = 13TQ =1.625uS

TSEG2 = 2TQ = 250nS

SJW = 2TQ = 250nS

The total bit time is TSEG1 + TSEG2 + TQ = 2uS

The single sample point is at 87.5%.

9 Loss of Synchronization Due to Accumulated Phase Error

While the assumption made in deriving the earlier bit timing equations is that the phase error will not exceed the SJW interval, the following analysis further examines the accumulation of phase error.

The effect of accumulated phase error between a slow receiver to a fast transmitter was previously shown in Figure 10 using recurring ten bit pattern intervals with only one recessive to dominant edge for resynchronization. In this situation, at each recessive to dominant edge the slower receiver must shorten its TSEG2 segment by the maximum amount, t_{SJW} in order to synchronize to the faster transmitter. Refer to Figure 12 for additional detail.

When the phase error over a ten bit interval is larger than t_{SJW}, the residual phase error accumulates and when the total cumulative phase error in any given ten bit interval equals t_{SEG2}, incorrect sampling of the transmitted data and loss of synchronization occurs.

To examine the effect of cumulative phase error, first define the phase error of any ***ith*** ten bit interval as $\Theta_{ERROR(i)}$,then the phase error for the first ten bit interval is -

$$\text{A1)}\quad \Theta_{ERROR(1)} = 10 \bullet t_{BIT(B)} - 10 \bullet t_{BIT(A)}$$

If the phase error is larger than t_{SJW}, the phase error is reduced by the maximum synchronization jump width. During the second ten bit interval the phase error increases by the previous uncompensated phase error as follows -

$$\text{A2)}\quad \Theta_{ERROR(2)} = 10 \bullet t_{BIT(B)} - 10 \bullet t_{BIT(A)} + \Theta_{ERROR(1)} - t_{SJW(B)}$$

Substituting for $\Theta_{ERROR(1)}$ from equation A1, the phase error in the second ten bit error is written as -

$$\text{A3)}\quad \Theta_{ERROR(2)} = 2 \bullet 10 \bullet (t_{BIT(B)} - t_{BIT(A)}) - t_{SJW(B)}$$

As each successive ten bit interval is included, the following relationship arises -

$$\Theta_{ERROR(3)} = 3 \bullet 10 \bullet (t_{BIT(B)} - t_{BIT(A)}) - 2 \bullet t_{SJW(B)}$$

$$\Theta_{ERROR(4)} = 4 \bullet 10 \bullet (t_{BIT(B)} - t_{BIT(A)}) - 3 \bullet t_{SJW(B)}$$

-
-
-

$$\text{A4)}\quad \Theta_{ERROR(i)} = i \bullet 10 \bullet (t_{BIT(B)} - t_{BIT(A)}) - (i-1) \bullet t_{SJW(B)}$$

To maintain synchronization, the phase error for any ten bit interval has to be less than the $t_{SEG2(B)}$ as follows -

$$\text{A5)}\quad \Theta_{ERROR(i)} \le t_{SEG2(B)}$$

Bit time intervals may be expressed in terms of the clock oscillator tolerance, Δf, as -

A6) $t_{INTERVAL(A)} = t_{INTERVAL(NOM)} \bullet (1 - \Delta f)$

A7) $t_{INTERVAL(B)} = t_{INTERVAL(NOM)} \bullet (1 + \Delta f)$

Substituting Equation A4 into Equation A5 and using the bit time intervals expressed in terms of the clock oscillator tolerance, Equation A5 can be solved for the oscillator tolerance -

$$A8) \quad \Delta f \leq \frac{(i-1) \bullet t_{SJW} + t_{TSEG2}}{20 \bullet i \bullet t_{BIT} - (i-1) \bullet t_{SJW} - t_{TSEG2}}$$

As the value of i approaches infinity, the oscillator drift limit approaches a limit which is determined by the size of the synchronization jump width relative to the bit period, expressed as -

$$A9) \quad \Delta f_{\text{LIM } i \to \infty} \leq \frac{t_{SJW}}{20 \bullet t_{BIT} - t_{SJW}}$$

Equation A9 represents the upper limit on maximum allowable oscillator drift as the number of successive 10 bit intervals approaches infinity. While this represents an impossible, infinite length CAN message, a reasonable value for i will be determined later.

A requirement for proper synchronization is that the synchronization jump width be less than or equal to the smaller of t_{SEG1} and t_{SEG2}. For the case where the TSEG2 interval width is equal to the SJW interval, Equation A8 becomes -

$$A10) \quad \Delta f_{t_{TSEG2}=t_{SJW}} \leq \frac{t_{SJW}}{20 \bullet t_{BIT} - t_{SJW}}$$

It is important to note at this point that Equation A8 reduces to the same equations, A9 and A10, either for the assumption that $i \to \infty$ or for the assumption that $t_{SJW}=t_{SEG2}$. For the case where i is finite and $t_{SJW} < t_{SEG2}$, Equation A8 is more accurate than Equations A9 and A10. However, Equation A8 is only slightly more conservative than Equations A9 and A10 for typical values of the bit timing parameters. For optimum system performance with respect to propagation delay, the optimum setting for t_{SEG2} is that it be made equal to t_{SJW}, so in most cases, Equation A10 is a simplification to Equation A8 which can generally be applied and accurately predicts the maximum allowable oscillator tolerance.

Because extended 29 bit CAN messages have a finite maximum message length of 128 bits (without including stuff bits), the largest possible value for "***i***" is equal to the number of bytes in the maximum length message -

$$A11) \quad i_{\max} = 128 \text{ Bits} \bullet \left(\frac{1 \text{ Resynchronization Interval}}{8 \text{ Bits}} \right)$$

$$i_{\max} = 16$$

However, not all the bits of a CAN message participate in bit stuffing; some protocol bits are fixed as dominant or recessive and cannot participate in a "1111100000..." repeating pattern. Considering only the transmitted data bytes, the lower limit of "i" is -

$$A12) \quad i_{\min} = 8 \text{ Bytes} \bullet \left(\frac{1 \text{ Resynchronization Interval}}{\text{Byte}} \right)$$

$$i_{\min} = 8$$

Even though the entire maximum length message cannot be forced to become 16 back to back ten bit time intervals with only one resynchronization edge per interval, some of the non-data message fields can be forced to cause residual phase error, and so a reasonable estimate for “i” for maximum length CAN messages is simply the arithmetic mean of the upper and lower limits, i.e. i = 12.

Substituting in Equation A8, this results in the following equation for estimation of the maximum allowable oscillator drift -

$$\text{A13)} \quad \Delta f \leq \frac{11 \bullet t_{SJW} + t_{SEG2}}{240 \bullet t_{BIT} - 11 \bullet t_{SJW} - t_{SEG2}}$$

Equation A13 allows more oscillator tolerance, for the case where $t_{SEG2} > t_{SJW}$, than the equations derived which assume that the maximum allowable phase shift between two nodes must not be greater than the SJW interval.

10 Questions and Answers

Q1 – Have car companies used this bit timing analysis ?

A1 – Automotive companies include DaimlerChrysler, Ford, and GM.

Q2 – Are any spreadsheets available that cover these CAN bit timing details?

A2 – Yes. Contact Vector directly – ask for Bruce Emaus

11 Additional Resources

The following material may provide further useful information -

VECTOR APPLICATION NOTES

AN-AND-1-105	Understanding Dominant/Recessive Logic and Bitwise Arbitration
AN-AND-1-106	Basic CAN Bit Timing

SAE DOCUMENTS (at sae.org)

J2284 Recommended Practice

SAE technical paper #970295 "CAN Bit Timing Requirements" by Karl Overberg & Klaus Dietmayer

OTHER RESOURCES

Philips application note AN97046 "Determination of Bit Timing Parameters for SJA 1000 CAN Controller" by Egon Johnk & Klaus Dietmayer

12 Contacts

Germany
and all countries not named below:
Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart
GERMANY
Phone: +49 711-80670-0
Fax: +49 711-80670-111
Email: info@vector-informatik.de

France, Belgium, Luxemburg:

Vector France SAS
168 Boulevard Camélinat
92240 Malakoff
FRANCE
Phone: +33 1 42 31 40 00
Fax: +33 1 42 31 40 09
Email: information@vector-france.fr

Sweden, Denmark, Norway, Finland, Iceland:
VecScan AB
Theres Svenssons Gata 9
41755 Göteborg
SWEDEN
Phone: +46 31 764 76 00
Fax: +46 31 764 76 19
Email: info@vecscan.com

United Kingdom, Ireland:
Vector GB Ltd.
Rhodium
Central Boulevard
Blythe Valley Park
Solihull, Birmingham
West Midlands B90 8AS
UNITED KINGDOM
Phone: +44 121 50681-50
Email: info@vector-gb.co.uk

USA, Canada, Mexico:
Vector CANtech, Inc.
39500 Orchard Hill Pl., Ste 550
Novi, MI 48375
USA
Phone: +1 248 449 9290
Fax: +1 248 449 9704
Email: info@vector-cantech.com

Japan:
Vector Japan Co. Ltd.
Seafort Square Center Bld. 18F
2-3-12, Higashi-shinagawa,
Shinagawa-ku
Tokyo 140-0002
JAPAN
Phone: +81 3 5769 7800
Fax: +81 3 5769 6975
Email: info@vector-japan.co.jp

Korea:
Vector Korea IT Inc.
Daerung Post Tower III, 508
182-4 Guro-dong, Guro-gu
Seoul, 152-790
REPUBLIC OF KOREA
Phone: +82 2 2028 0600
Fax: +82 2 2028 0604
Email: info@vector-korea.com